Financial Econometrics – Lecture Notes

Rauli Susmel
Dept of Finance
Bauer College of Business
University of Houston

Lecture 1 – Review of Statistics and Linear Algebra (NOT Covered)

Random Variable

In probability and statistics, a *random variable* (RV), or *stochastic variable* is described informally as a variable whose values depend on outcomes of a random experiment (phenomenon).

Notation:

Ω the sample space –the set of possible outcomes from an experiment.
An event A is a set containing outcomes from the sample space.
Σ is the collection of all possible events involving outcomes chosen from Ω. (Formally: Σ is a σ -algebra of subsets of the sample space.)
P is a probability measure over Σ. P assigns a number between [0,1] to each event in Σ.
P is the probability measure over the sample space, Ω, and P_X is the probability measure over χ, the range of the random variable.

Remarks:

- A random variable is a convenient way to express the elements of Ω as numbers rather than abstract elements of sets.
- A random variable ***X*** is a function.
- It is a numerical quantity whose value is determined by a random experiment.
- It takes single elements in outcome set Ω, which can be abstract elements, and maps them to points in R.

Example: Two fair coins are tossed.
The sample space is Ω = {HH; HT; TH; TT}.
Possible events:
- The coins have different sides showing: {HT; TH}.
- At least one head: {HH; HT; TH}.
- First coin shows heads or second coin shows tails: {HH, HT, TT}

The collection of all possible events is Σ = [Φ, {H,H}, {HT}, {TH}, {TT}, {HH,HT}, {HH,TH}, {HT,TH}, {TH,TT}, {HH,HT,TH}, {HH,HT,TT}, {HH,TH,TT}, {HT,TH,TT}, {HH,HT,TH,TT}]
Let the random variable **X** be *"number of heads."* Recall that **X** takes Ω into χ and induces P_X from P. In this example, χ = {0; 1; 2} and A = {Φ; {0}; {1}; {2}; {0;1}; {0;2}; {1;2}; {0;1;2}}.
Prob. of 0 heads = P_X[0] = P[{TT}] = 1/4
Prob. of 1 heads = P_X[1] = P[{HT; TH}] = 1/2
Prob. of 2 heads = P_X[2] = P[{HH}] = ¼
Prob. of 0 or 1 heads = P_X[{0; 1}] = P[{TT; TH; HT}] = 3/4
Prob. of 0 or 2 heads = P_X[{0; 2}] = P[{TT; HH}] = 1/2

Prob. of 1 or 2 heads = Px[{1; 2}] = P[{TH; HT; HH}] = 3/4
Prob. of 1, 2, or 3 heads = Px[{0; 1; 2}] = P[{HH; TH; HT; TT}] = 1
Prob. of "nothing" = Px[Φ] = P[Φ] = 0
The empty set is simply needed to complete the σ-algebra (a technical point). Its interpretation is not important since P[Φ] = 0 for any reasonable P.

Probability Function & CDF

Definition – The *probability function*, $p(x)$, of a RV, X.
For any random variable, X, and any real number, x, we define

$$p(x) = P[X = x] = P[\{X = x\}]$$

where $\{X = x\}$ = the set of all outcomes (event) with $X = x$.

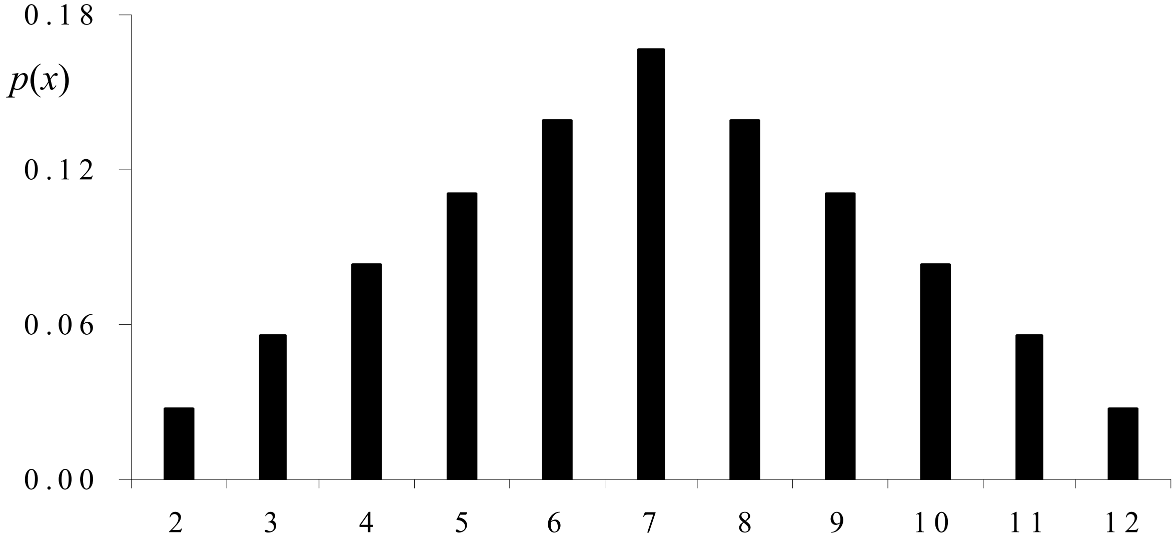
Definition – The *cumulative distribution function* (CDF), $F(x)$, of a RV, X.
For any random variable, X, and any real number, x, we define

$$F(x) = P[X \le x] = P[\{X \le x\}]$$

where $\{X \le x\}$ = the set of all outcomes (event) with $X \le x$.

Example: Two dice are rolled and X is the sum of the two upward faces. Sample space S = { 2:(1,1), 3:(1,2; 2,1), 4:(1,3; 3,1; 2,2), 5:(1,4; 2,3; 3,2; 4,1), 6, 7, 8, 9, 10, 11, 12}.

Graph: Probability function:



$$p(2) = P[X = 2] = P[\{(1,1)\}] = \frac{1}{36}$$
$$p(3) = P[X = 3] = P[\{(1,2),(2,1)\}] = \frac{2}{36}$$
$$p(4) = P[X = 4] = P[\{(1,3),(2,2),(3,1)\}] = \frac{3}{36}$$

$$p(5)=\frac{4}{36}, p(6)=\frac{5}{36}, p(7)=\frac{6}{36}, p(8)=\frac{5}{36}, p(9)=\frac{4}{36}$$

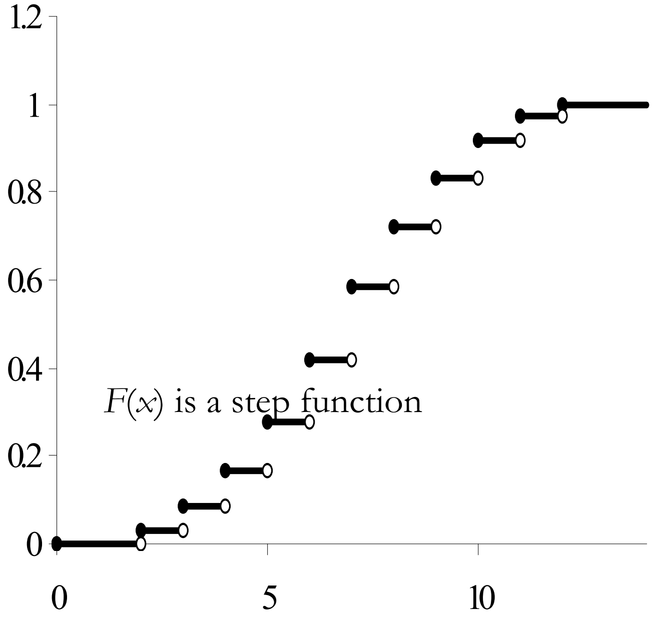
$$p(10)=\frac{3}{36}, p(11)=\frac{2}{36}, p(12)=\frac{1}{36}$$

and $p(x)=0$ for all other x

Note: $\{X=x\}=\phi$ for all other x

Graph: CDF

$$F(x)=\begin{cases} 0 & x<2 \\ \frac{1}{36} & 2\le x<3 \\ \frac{3}{36} & 3\le x<4 \\ \frac{6}{36} & 4\le x<5 \\ \frac{10}{36} & 5\le x<6 \\ \frac{15}{36} & 6\le x<7 \\ \frac{21}{36} & 7\le x<8 \\ \frac{26}{36} & 8\le x<9 \\ \frac{30}{36} & 9\le x<10 \\ \frac{33}{36} & 10\le x<11 \\ \frac{35}{36} & 11\le x<12 \\ 1 & 12\le x \end{cases}$$



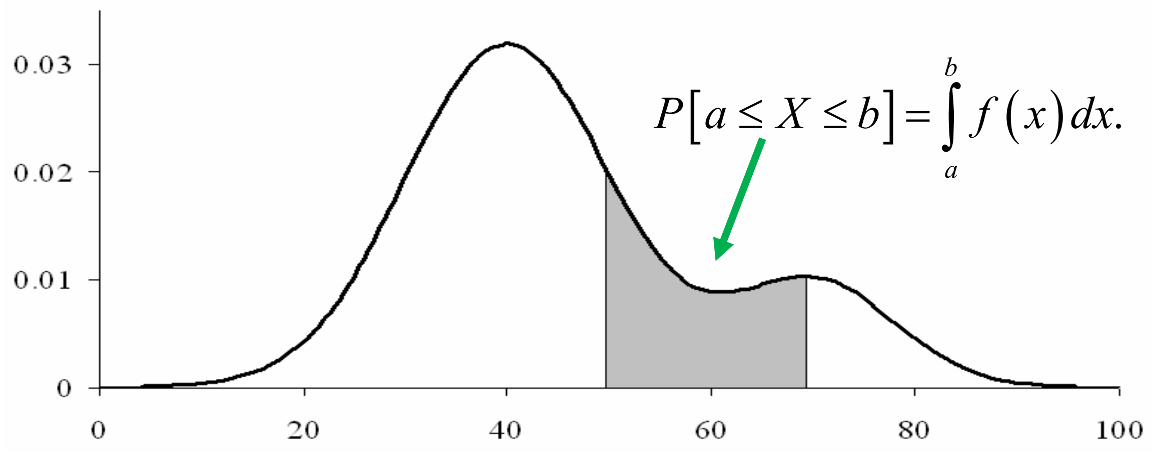
PDF for a Continuous RV

Definition: Suppose that X is a random variable. Let $f(x)$ denote a function defined for $-\infty < x < \infty$ with the following properties:

1. $f(x) \ge 0$
2. $\int_{-\infty}^{\infty} f(x)\,dx = 1.$
3. $P[a \le X \le b] = \int_a^b f(x)\,dx.$

Then, $f(x)$ is called the *probability density function* (pdf) of X. The random variable X is called *continuous*.

- PDF

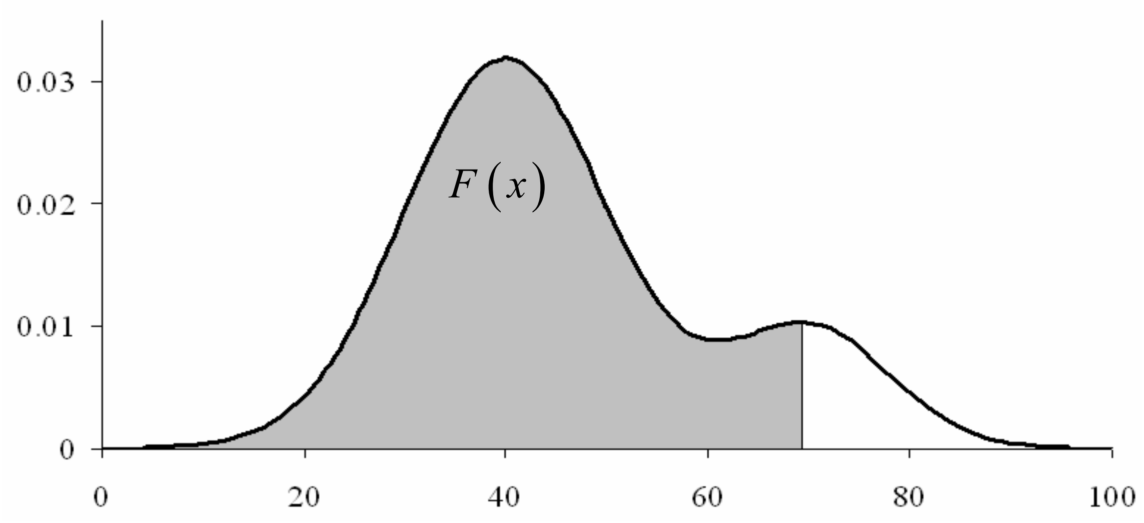


$$P[a \le X \le b] = \int_a^b f(x)\,dx.$$

- If *X* is a continuous random variable with probability density function, *f*(*x*), the *cumulative distribution function* of *X* is given by:

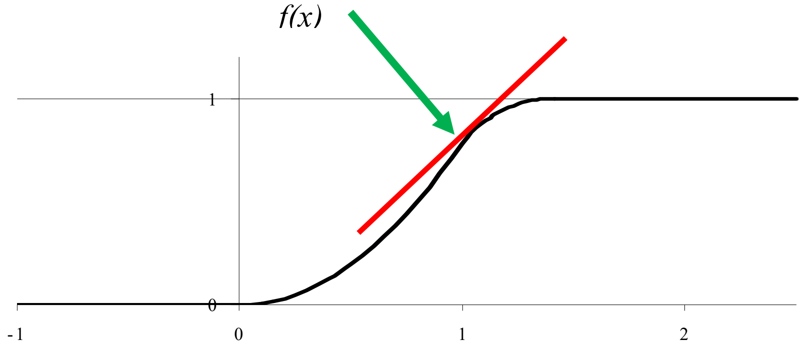
$$F(x) = P[X \le x] = \int_{-\infty}^{x} f(t)\,dt.$$

- CDF



- Also because of the FTC (*fundamental theorem of calculus*):

$$F'(x) = \frac{dF(x)}{dx} = f(x)$$



PDF for a Discrete RV

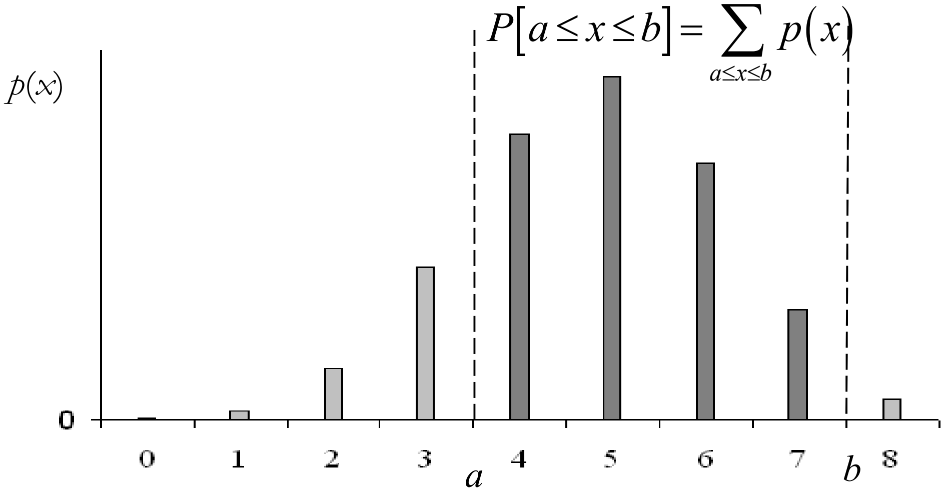
A random variable X is called *discrete* if

$$\sum_{x} p(x) = \sum_{i=1}^{\infty} p(x_i) = 1$$

All the probability is accounted for by values, x, such that $p(x) > 0$.

• For a discrete random variable X the probability distribution is described by the probability function $p(x)$, which has the following properties:

1. $0 \le p(x) \le 1$
2. $\sum_{x} p(x) = \sum_{i=1}^{\infty} p(x_i) = 1$
3. $P[a \le x \le b] = \sum_{a \le x \le b} p(x)$



Bernouille and Binomial Distributions

Suppose that we have a *Bernoulli trial* (an experiment) that has 2 results:

1. Success (S)
2. Failure (F)

Suppose that p is the probability of success (S) and $q = 1 - p$ is the probability of failure (F). Then, the probability distribution with probability function:

$$p(x) = P[X = x] = \begin{cases} q & x = 0 \\ p & x = 1 \end{cases}$$

is called the *Bernoulli distribution.*

• We observe an independent Bernoulli trial (**S, F)** n times. Let X be the number of successes in the n trials. Then, X has a *binomial distribution*:

$$p(x) = P[X = x] = \binom{n}{x} p^x q^{n-x} \quad x = 0, 1, 2, \ldots, n$$

where

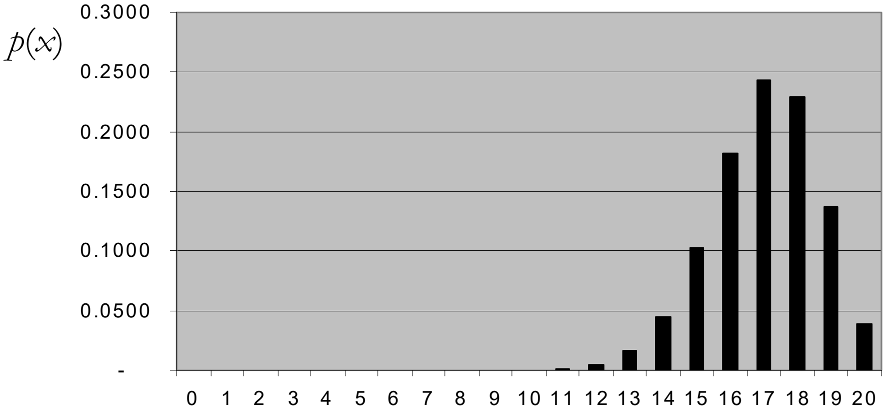
1. p = the probability of success (**S**), and
2. $q = 1 - p$ = the probability of failure (**F**)
3.

Example: If a firm announces profits and they are "surprising," the chance of a stock price, P, increase is 85%. Assume there are n=20 (independent) announcements.

Let X be the number of increases in the stock price following *surprising announcements* in the n = 20 trials.

$$p(x) = P[X = x] = \binom{n}{x} p^x q^{n-x} \quad x = 0, 1, 2, \ldots, n$$
$$= \binom{20}{x} (.85)^x (.15)^{20-x} \quad x = 0, 1, 2, \ldots, 20$$

x	0	1	2	3	4	5
$p(x)$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x	6	7	8	9	10	11
$p(x)$	0.0000	0.0000	0.0000	0.0000	0.0002	0.0011
x	12	13	14	15	16	17
$p(x)$	0.0046	0.0160	0.0454	0.1028	0.1821	0.2428
x	18	19	20			
$p(x)$	0.2293	0.1368	0.0388			



The Poisson Distribution

Suppose events are occurring randomly and uniformly in time.

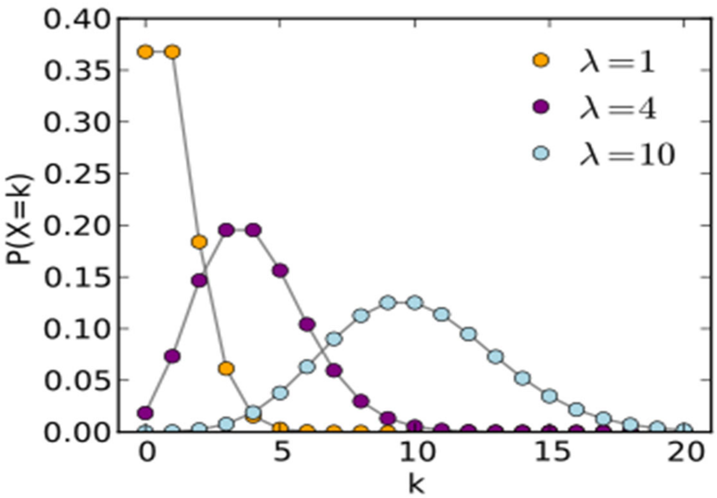
- The events occur with a known average.
- Let X be the number of events occurring (arrivals) in a fixed period of time (time-interval of given length).
- Typical example: X = Number of crime cases coming before a criminal court per year (original Poisson's application in 1838.)
- Then, X will have a *Poisson distribution* with parameter λ:

$$p(x) = \frac{\lambda^x}{x!} e^{-\lambda} \qquad x = 0, 1, 2, 3, 4, \ldots$$

- The parameter λ represents the expected number of occurrences in a fixed period of time. The parameter λ is a positive real number.

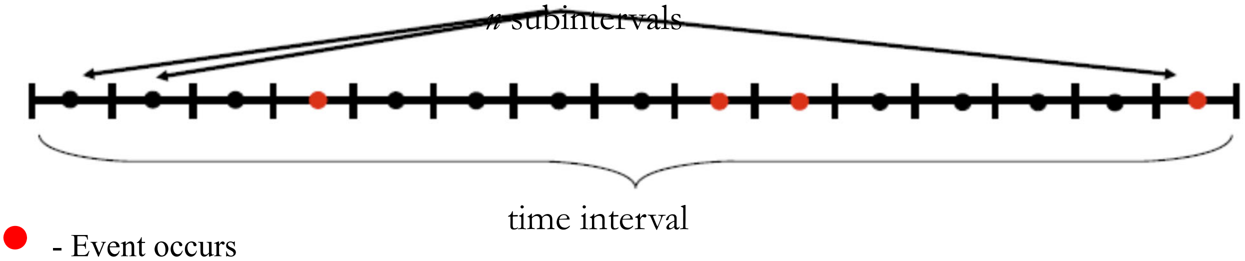
Example: On average, a trade occurs every 15 seconds. Suppose trades are independent. We are interested in the probability of observing 10 trades in a minute (X=10). A Poisson distribution can be used with λ = 4 (4 trades per minute).

- Poisson probability function



Poisson Distribution: Illustration

Suppose a time interval is divided into n equal parts and that one event may or may not occur in each subinterval.



● - Event occurs

● - Event does not occur

X = # of events is $Bin(n,p)$
As $n \rightarrow \infty$, events can occur over the continuous time interval

X = # of events is $Poisson(\lambda)$

Poisson Distribution: Comments

• The Poisson distribution arises in connection with Poisson processes - a stochastic process in which events occur continuously and independently of one another.
• It occurs most easily for time-events; such as the number of calls passing through a call center per minute, or the number of visitors passing through a turnstile per hour. However, it can apply to any process in which the mean can be shown to be constant.
• It is used in *finance* (number of jumps in an asset price in a given interval); *market microstructure* (number of trades per unit of time in a stock market); *sports economics* (number of goals in sports involving two competing teams); *insurance* (number of a given disaster - volcano eruptions/hurricanes/floods- per year); etc.

Example: The number of named storms over a period of a year in the Caribbean is known to have a Poisson distribution with λ = 13.1
Determine the probability function of X.
Compute the probability that X is at most 8.
Compute the probability that X is at least 10.
Given that at least 10 hurricanes occur, what is the probability that X is at most 15?
Solution:

$$p(x) = \frac{\lambda^x}{x!}e^{-\lambda} \qquad x = 0,1,2,3,4,\ldots$$

$$= \frac{13.1^x}{x!}e^{-13.1} \qquad x = 0,1,2,3,4,\ldots$$

x	$p(x)$	x	$p(x)$
0	0.000002	10	0.083887
1	0.000027	11	0.099901
2	0.000175	12	0.109059
3	0.000766	13	0.109898
4	0.002510	14	0.102833
5	0.006575	15	0.089807
6	0.014356	16	0.073530
7	0.026866	17	0.056661
8	0.043994	18	0.041237
9	0.064036	19	0.028432

$$P[\text{at most } 8] = P[X \le 8]$$
$$= p(0) + p(1) + \cdots + p(8) = .09527$$

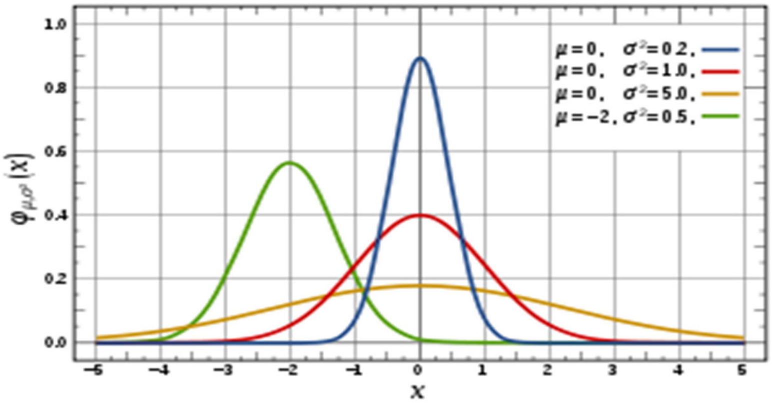
$$P[\text{at least } 10] = P[X \ge 10] = 1 - P[X \le 9]$$
$$= 1 - \left(p(0) + p(1) + \cdots + p(9)\right) = .8400$$

$$P\left[\text{at most } 15 \middle| \text{at least } 10\right] = P\left[X \le 15 \middle| X \ge 10\right]$$
$$= \frac{P\left[\{X \le 15\} \cap \{X \ge 10\}\right]}{P[X \ge 10]} = \frac{P[10 \le X \le 15]}{P[X \ge 10]}$$
$$= \frac{p(10) + p(11) + \cdots + p(15)}{.8400} = 0.708$$

The Normal distribution

A random variable, *X*, is said to have a *normal distribution* with mean *m* and standard deviation *s* if *X* is a continuous random variable with probability density function *f*(*x*):

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Normal distribution: Properties

1. Indexed by two parameters: *m (central parameter) & s (spread parameter).*

2. Symmetric around *m,* which is the location of the maximum of *f*(*x*).

Check:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$f'(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left[-\tfrac{1}{\sigma^2}(x-\mu)\right] = 0$$

This equality holds when $m = x$. Thus, m is an extremum point of $f(x)$. Since $f(x)$ is a pdf, it is the mode.

3. The inflection points of $f(x)$ are $m - s$, $m + s$. (Check: set $f''(x) = 0$ and solve for x.)

Normal distribution: Comments

- The normal distribution is often used to describe or approximate any variable that tends to cluster around the mean. It is the most assumed distribution in economics and finance: rates of return, growth rates, IQ scores, observational errors, etc.
- The central limit theorem (CLT) provides a justification for the normality assumption when n is large.

Notation: PDF: $x \sim \mathrm{N}(\mu,\sigma^2)$
CDF: $\Phi(x)$

The Expectation of X: E(X)

The expectation operator defines the mean (or population average) of a random variable or expression.

Definition

Let X denote a *discrete* RV with probability function $p(x)$ (probability density function $f(x)$ if X is *continuous*) then the expected value of X, $E(X)$ is defined to be:

$$E(X) = \sum_x xp(x) = \sum_i x_i p(x_i)$$

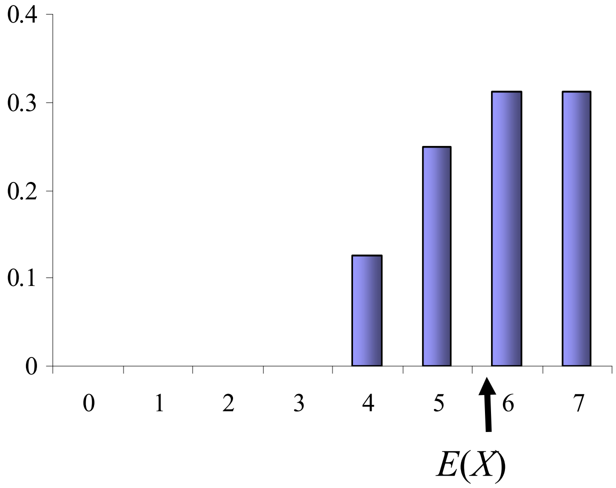
and if X is *continuous* with probability density function $f(x)$

$$E(X) = \int_{-\infty}^{\infty} xf(x)\,dx$$

Sometimes we use E[.] as E_X[.] to indicate that the expectation is being taken over $f_X(x)\,dx$

Interpretation of E(X)

1. The expected value of X, $E(X)$, is the center of gravity of the probability distribution of X.
2. The expected value of X, $E(X)$, is the *long-run average value* of X. (To be discussed later: *Law of Large Numbers*)



E[X]: The Normal Distribution

Suppose *X* has a Normal distribution with parameters *m* and *s.*
Then, E[X] = *m.*

Proof:

$$f\left(x\right)=\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$E\left(X\right)=\int_{-\infty}^{\infty}xf\left(x\right)dx=\int_{-\infty}^{\infty}x\frac{1}{\sqrt{2\pi}\,\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}dx$$

Making the substitution:

$$z=\frac{x-\mu}{\sigma}\quad\Rightarrow\qquad dz=\frac{1}{\sigma}dx\quad\text{and}\quad x=\mu+z\sigma$$

Then,

$$E\left(X\right)=\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}\left(\mu+z\sigma\right)e^{-\frac{z^2}{2}}dz$$
$$=\mu\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}dz+\frac{\sigma}{\sqrt{2\pi}}\int_{-\infty}^{\infty}ze^{-\frac{z^2}{2}}dz$$

Using the following results:

$$\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}dz=1\quad\text{and}\quad\int_{-\infty}^{\infty}ze^{-\frac{z^2}{2}}dz=0$$

Thus $E\left(X\right)=\mu$

Expectation of a function of a RV

Let *X* denote a *discrete RV* with probability function *p*(*x*), then the expected value of *g*(*X*), *E*[*g*(*X*)], is defined to be:

$$E\left[g\left(X\right)\right]=\sum_{x}g\left(x\right)p\left(x\right)=\sum_{i}g\left(x_i\right)p\left(x_i\right)$$

and if X is *continuous* with probability density function $f(x)$

$$E\left[g(X)\right] = \int_{-\infty}^{\infty} g(x)\, f(x)\, dx$$

Examples: $g(x) = (x - \mu)^2 \Rightarrow \mathrm{E}[g(x)] = \mathrm{E}[(x - \mu)^2]$

$g(x) = (x - \mu)^k \Rightarrow \mathrm{E}[g(x)] = \mathrm{E}[(x - \mu)^k]$

Example: Suppose X has a uniform distribution from 0 to b. Then:

$$f(x) = \begin{cases} \frac{1}{b} & 0 \le x \le b \\ 0 & x < 0, x > b \end{cases}$$

Find the *expected value* of $A = X^2$.

If X is the length of a side of a square (chosen at random from 0 to b) then A is the area of the square

$$E\left(X^2\right) = \int_{-\infty}^{\infty} x^2 f(x)\, dx = \int_{a}^{b} x^2 \tfrac{1}{b-a}\, dx = \left[\tfrac{1}{b}\frac{x^3}{3}\right]_0^b = \frac{b^3 - 0^3}{3(b)} = \frac{b^2}{3}$$

= 1/3 the maximum area of the square

Median: Another central measure

A median is the numeric value separating the higher half of a sample, a population, or a probability distribution, from the lower half.

Definition: Median

The *median* of a random variable $\boldsymbol{X}$ is the unique number m that satisfies the following inequalities:

$\mathrm{P}(\boldsymbol{X} \le m) \ge ½$ and $\mathrm{P}(\boldsymbol{X} \ge m) \ge ½$.

For a continuous distribution, we have that m solves:

$$\int_{-\infty}^{m} f_X(x)\, dx = \int_{m}^{\infty} f_X(x)\, dx = 1/2$$

Note: If the **mean** > **median** > **mode** (= most popular observation), the distribution will be skewed to the right. If the **mean** < **median** < **mode**, the distribution will be skewed to the left.

• Calculation of medians is a popular technique in summary statistics and summarizing statistical data, since it is simple to understand and easy to calculate, while also giving a measure that is more robust in the presence of outlier values than is the mean.

An optimality property

A median is also a central point which minimizes the average of the absolute deviations. That is, a value of c that minimizes

$\mathrm{E}(|\mathbf{X} - c|)$

is the median of the probability distribution of the random variable $\boldsymbol{X}$.

Example: Let X have an exponential distribution with parameter l. The probability density function of X is:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The median *m* solves the following integral of *X*:

$$\int_{m}^{\infty} f_X(x)dx = 1/2$$

$$\int_{m}^{\infty} \lambda\, e^{-\lambda x} dx = \lambda \int_{m}^{\infty} e^{-\lambda x} dx = -e^{-\lambda x}\, |_m^{\infty} = e^{-\lambda m} = 1/2$$

That is, $m = \ln(2)/\lambda$.

Moments of Random Variables

The moments of a random variable *X* are used to describe the behavior of the RV (discrete or continuous).

Definition: *K*th Moment
Let *X* be a RV (discrete or continuous), then the *kth moment of X* is:

$$\mu_k = E(X^k) \ = \begin{cases} \sum_x x^k p(x) & if X \ is \ discrete \\ \int_{-\infty}^{\infty} x^k f(x)dx & if X \ is \ continuous \end{cases}$$

Definition: Central Moments
Let *X* be a RV (discrete or continuous). Then, the *kth central moment* of *X* is defined to be:

$$\mu_k^0 = E[(X-\mu)^k] \ = \begin{cases} \sum_x (x-\mu)^k p(x) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty} (x-\mu)^k f(x)dx & \text{if } X \text{ is continuous} \end{cases}$$

where $m = m_1 = E(X)$ = the first moment of *X.*

• The central moments describe how the probability distribution is distributed about the center of gravity, *m.*

• The first central moments is given by:

$$\mu_1^0 = E[X-\mu]$$

• The second central moment depends on the *spread* of the probability distribution of *X* about *m.* It is called the variance of *X* and is denoted by the symbol σ^2 = var(*X*):

$$\mu_2^0 = E[(X-\mu)^2] = \text{var(X)} = \ \sigma^2$$

The square root of var(X) is called the *standard deviation* of *X* and is denoted by the symbol *s* = SD(X). We also refer to it as *volatility*:

$$\sqrt{\mu_2^0} = \sqrt{E[(X-\mu)^2]} = \sigma$$

Moments of a RV: Skewness

The third central moment:

$$\mu_3^0 = E[(X-\mu)^3]$$

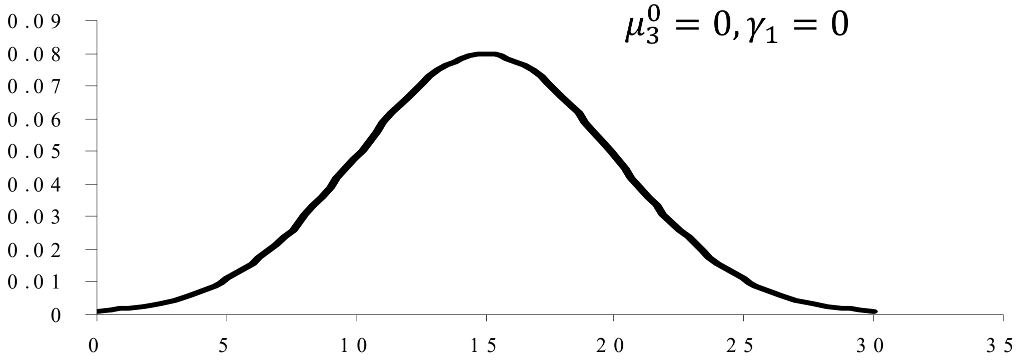
μ_3^0 contains information about the *skewness* of a distribution.

- A popular measure of skewness:

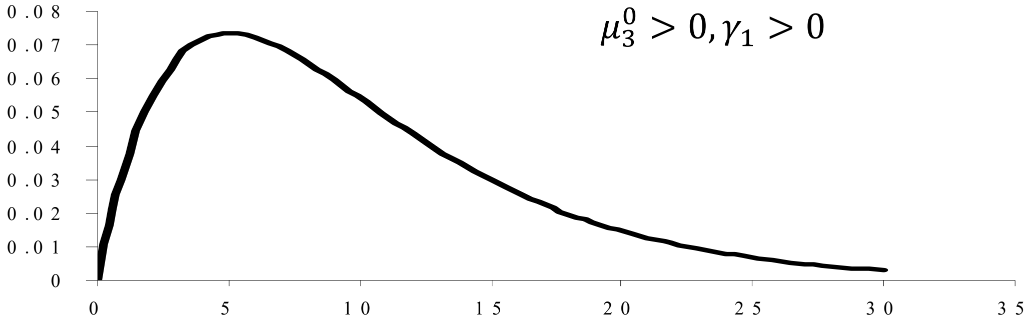
$$\gamma_1 = \frac{\mu_3^0}{\sigma^3} = \frac{\mu_3^0}{(\mu_2^0)^{\frac{3}{2}}}$$

- Distribution according to skewness:

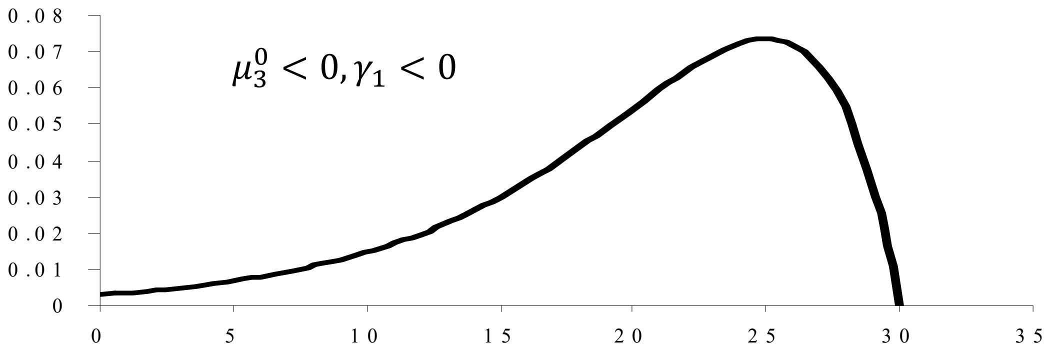
1) Symmetric distribution



2) Positively (right-) skewed distribution (with mode < median < mean)



3) Negatively (left-) skewed distribution (with mode > median > mean)



- Skewness and Economics
- Zero skew means symmetrical gains and losses.
- Positive skew suggests many small losses and few rich returns.
- Negative skew indicates lots of minor wins offset by rare major losses.

- In financial markets, stock returns at the firm level show positive skewness, but at the aggregate (index) level show negative skewness.

- From horse race betting and from U.S. state lotteries there is evidence supporting the contention that gamblers are not necessarily risk-lovers but skewness-lovers: Long shots are overbet (positive skewness loved!).

Moments of a RV: Kurtosis

The fourth central moment:

$$\mu_4^0 = E[(X-\mu)^4]$$

It contains information about the *shape* of a distribution. The property of shape that is measured by this moment is called *kurtosis,* usually estimated by

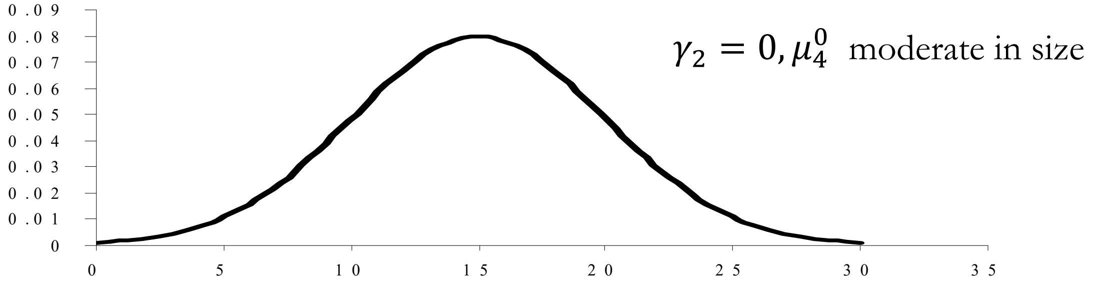
$$\gamma_2 = \frac{\mu_4^0}{\sigma^4}.$$

- The *measure of (excess) kurtosis***:**

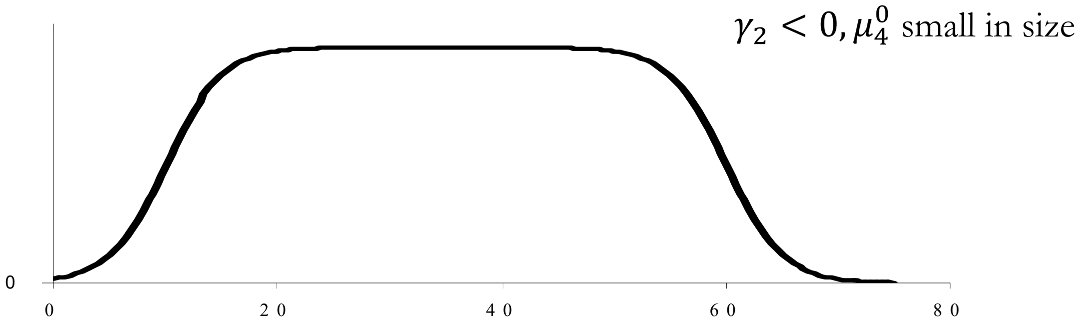
$$\gamma_2 = \frac{\mu_4^0}{\sigma^4} - 3 = \frac{\mu_4^0}{(\mu_2^0)^2} - 3$$

- Distributions:

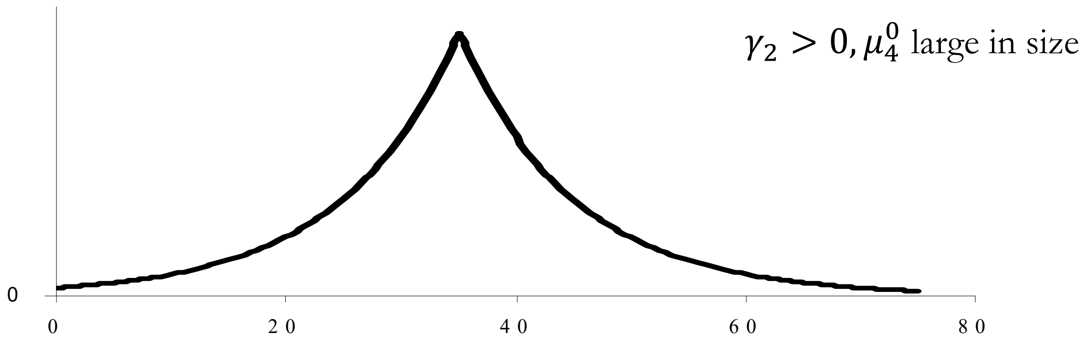
1) Mesokurtic distribution



2) Platykurtic distribution


$\gamma_2 < 0, \mu_4^0$ small in size

3) Leptokurtic distribution (usual shape for asset returns)


$\gamma_2 > 0, \mu_4^0$ large in size

Moments and Expected Values

Note that moments are defined by expected values. We define the expected value of a function of a continuous RV X, $g(X)$, as

$$E\left[g\left(X\right)\right] = \int_{-\infty}^{\infty} g\left(x\right) f\left(x\right) dx$$

- If X is *discrete* with probability function $p(x)$

$$E\left[g\left(X\right)\right] = \sum_x g\left(x\right) p\left(x\right) = \sum_i g\left(x_i\right) p\left(x_i\right)$$

Examples: $g(x) = (x - \mu)^2 \Rightarrow \text{E}[g(x)] = \text{E}[(x - \mu)^2]$

$$g(x) = (x - \mu)^k \;\Rightarrow \mathrm{E}[g(x)] = \mathrm{E}[(x - \mu)^k]$$

• We estimate expected values with sample averages. The Law of Large Numbers (LLN) tells us they are *consistent* estimators of expected values.

Estimating Moments

We estimate expected values with sample averages. For example, the first moment, the mean, and the second central moment, the variance, are estimated by:

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

$$s^2 = \frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N-1} \qquad (N\text{-1 adjustment needed for } \mathrm{E}[s^2] = \sigma^2)$$

• Besides consistent, they are both are *unbiased* estimators of their respective population moments (unbiased = "on average, I get the population parameter"). That is,

$\mathrm{E}[\bar{X}] = \mu$ "population parameter"

$\mathrm{E}[s^2] = \sigma^2$

The Law of Large Numbers (LLN)

Long history: Gerolamo Cardano (1501-1576) stated it without proof. Jacob Bernoulli published a rigorous proof in 1713.

Theorem (Weak LLN)

Let $X_1, \ldots, X_n$ be n mutually independent random variables each having mean m and a finite s -i.e, the sequence $\{X_n\}$ is *i.i.d.*

Let $\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$.

Then for any $d > 0$ (no matter how small)

$$P\left[\left|\bar{X} - \mu\right| < \delta\right] = P\left[\mu - \delta < \bar{X} < \mu + \delta\right] \rightarrow 1 \;\text{ as }\; n \rightarrow \infty$$

• There are many variations of the LLN. It is a general result: A sample average as the sample size goes to infinite tends to its expected value. Also written as $\bar{X}_{\text{n}} \overset{p}{\rightarrow} \mu$. *(convergence in probability*)

Central Limit Theorem (CLT)

Let $X_1, X_2, \ldots, X_n$ be a sequence of *i.i.d.* RVs with finite mean m, and finite variance s^2. Then as n increases, $\bar{X}_{\text{n}}$, the sample mean, approaches the normal distribution with mean μ and variance s^2/n.

This theorem is sometimes stated as $\frac{\sqrt{N}(\bar{X}-\mu)}{\sigma} \overset{d}{\rightarrow} N(0,1)$

where $\overset{d}{\rightarrow}$ means "the limiting distribution (asymptotic distribution) is" (or *convergence in distribution*).

• Many version of the CLT. This one is the *Lindeberg-Lévy CLT*.

• The CLT gives only an asymptotic distribution. We usually take it as an approximation for a finite number of observations. In these cases, the notation goes from $\overset{d}{\rightarrow}$ to $\overset{a}{\rightarrow}$.

Sampling Distributions

All statistics, T(X), are functions of RVs and, thus, they have a distribution. Depending on the sample, we can observe different values for T(X), thus, the finite sample distribution of T(X) is called the *sampling distribution*.

For the sample mean $\bar{X}$, if the X_i's are normally distributed, then the sampling distribution is normal with mean μ and variance σ^2/N. Or

$$\bar{X} \sim N(\mu, \sigma^2/N).$$

Note: If the data is not normal, the CLT is used to approximate the sampling distribution by the asymptotic one, usually after some manipulations. Again, in those cases, the notation goes from $\overset{d}{\rightarrow}$ to $\overset{a}{\rightarrow}$.

• The SD of the sampling distribution is called the *standard error* (SE). Then, SE($\bar{X}$) = σ/sqrt(N).

• For the sample variance σ^2, if the X_i's are normally distributed, then the sampling distribution is derived from this result:

$$(N-1)\, s^2/\sigma^2 \sim \chi^2_{N-1}.$$

It can be shown that the χ^2_{N-1} has a variance equal to 2 times the degrees of freedom (=2*(N – 1)), that is,

$$\text{Var}[(N-1)\, s^2/\sigma^2\,] = 2 * (N-1) \qquad \Rightarrow \text{Var}[s^2] = 2 * \sigma^4 /(N-1)$$

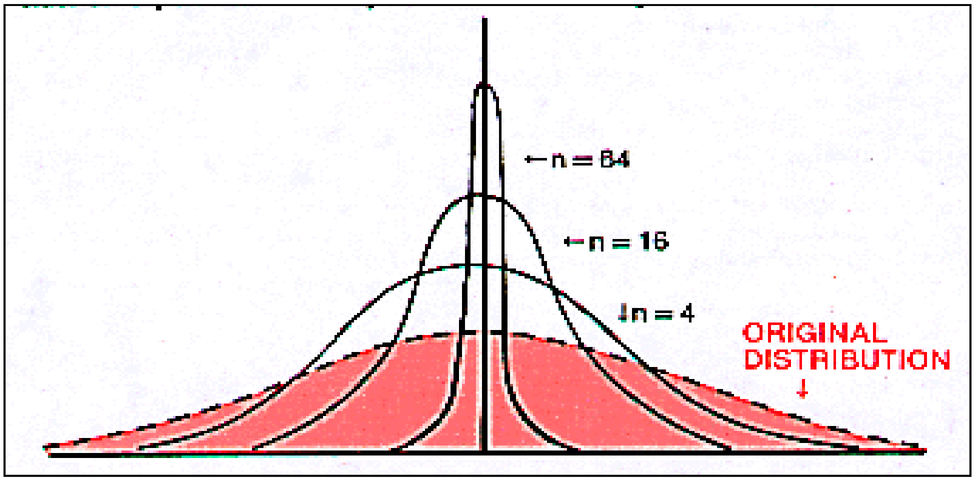
Then, SE(s^2) = SD(s^2) = σ^2 * sqrt[2/(N – 1)].

Note: If the data is not normal (& N is large), the CLT can be used to approximate the sampling distribution by the asymptotic one:

$$s^2 \overset{a}{\rightarrow} N(\sigma^2, 2*\sigma^4/(N-1))$$

• Sampling Distribution for the Sample Mean of a normal population:

$$\bar{X} \sim N(m, \sigma^2/n)$$



Note: As $n \to \infty$, $\bar{X} \to \mu$ –i.e., the distribution becomes a spike at μ!

The Central Limit Theorem

The Central Limit Theorem (CLT) states conditions for the sequence of RV $\{x_n\}$ under which the mean or a sum of a sufficiently large number of x_i's will be approximately normally distributed.

CLT: Under some conditions, $z = n^{1/2}\,(\bar{X} - \mu)/\sigma \xrightarrow{d} N(0,1)$.

• It is a general result. When sums of random variables are involved, eventually (sometimes after transformations) the CLT can be applied.

• The *Berry–Esseen theorem* (*Berry–Esseen inequality*) attempts to quantify the rate at which the convergence to normality takes place.

$$| F_n(x) - \Phi(x) | \leq \frac{C\rho}{\sigma^3 n^{1/2}}$$

where $\rho = E(|X|) < \infty$ and C is a constant (best current C=0.7056).

Two popular versions used in economics and finance:
Lindeberg-Levy: $\{x_n\}$ are *i.i.d.*, with finite μ and finite σ^2.

Lindeberg-Feller: $\{x_n\}$ are independent, with finite μ_i, $\sigma_i^2 < \infty$, $S_n = \Sigma_i\, x_i$, $s_n^2 = \Sigma_i\, \sigma_i^2$ and for $\varepsilon > 0$,

$$\lim_{n\to\infty} \frac{1}{s_n^2} \sum_{i=1}^{n} \int_{|x_i-\mu_i|>\varepsilon S_n} (x_i - \mu_i)^2 f(x_i) dx = 0$$

Note:
Lindeberg-Levy assumes random sampling –observations are *i.i.d.*, with the same mean and same variance.

Lindeberg-Feller allows for heterogeneity in the drawing of the observations --through different variances. The cost of this more general case: More assumptions about how the $\{x_n\}$ vary.

Asymptotic Distribution

An asymptotic distribution is a hypothetical distribution that is the *limiting* distribution of a sequence of distributions.

We will use the asymptotic distribution as a finite sample *approximation* to the true distribution of a RV when *n* -i.e., the sample size- is *large*.

Practical question: When is *n* large?

Hypothesis Testing

A *statistical hypothesis test* is a method of making decisions using experimental data. A result is called *statistically significant* if it is unlikely to have occurred by chance.

• These decisions are made using (null) hypothesis tests. A hypothesis can specify a particular value for a population parameter, say $q=q_0$. Then, the test can be used to answer a question like:

Assuming q_0 is true, what is the probability of observing a value for the (test) statistic used that is at least as big as the value that was actually observed?

• Uses of hypothesis testing:
- Check the validity of theories or models.
- Check if new data can cast doubt on established facts.

• In general, there are two kinds of hypotheses:
(1) About the form of the probability distribution
Example: Is the random variable normally distributed?

(2) About the parameters of a distribution function
Example: Is the mean of a distribution equal to 0?

• The second class is the traditional material of econometrics. We may test whether the effect of income on consumption is greater than one, or whether the size coefficient on a CAPM regression is equal to zero.

• Hypothesis testing involves the comparison between two competing hypothesis (sometimes, they represent partitions of the world).
- The null hypothesis, denoted H_0, is sometimes referred to as the maintained hypothesis.
- The alternative hypothesis, denoted H_1, is the hypothesis that will be considered if the null hypothesis is “rejected.”

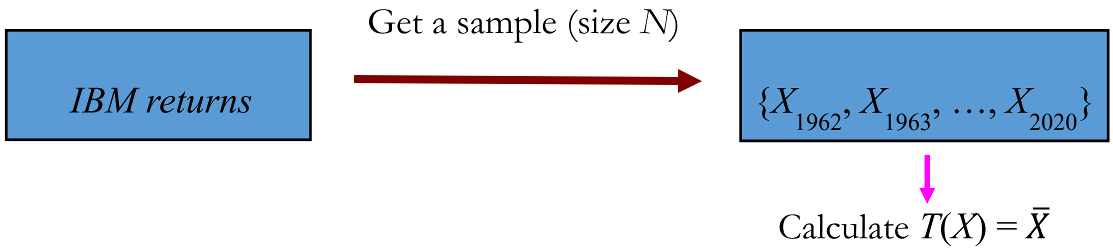
Idea: We collect a sample of data $X = \{X_1, \ldots, X_n\}$. We construct a statistic $T(X) = f(X)$, called the *test statistic*. Now we have a decision rule:

- If $T(X)$ is contained in space R, we reject H_0 (& we learn).
- If $T(X)$ is in the complement of R (R^C), we fail to reject H_0.

Note: $T(X)$, like any other statistic, is a RV. It has a distribution.

Example: Suppose we want to test if the mean of IBM annual returns, μ_{IBM}, is 10%. That is, H_0: $\mu_{IBM} = 10\%$.

From the population, we get a sample: $\{X_{1962}, X_{1963}, \ldots, X_{n=2020}\}$, with N=59. We use $T(X) = \bar{X}$, which is unbiased, consistent, and, assuming X is normally distributed, we know its distribution, $\bar{X} \sim N(\mu, \sigma^2/n)$.



Get a sample (size N)

IBM returns

$\{X_{1962}, X_{1963}, \ldots, X_{2020}\}$

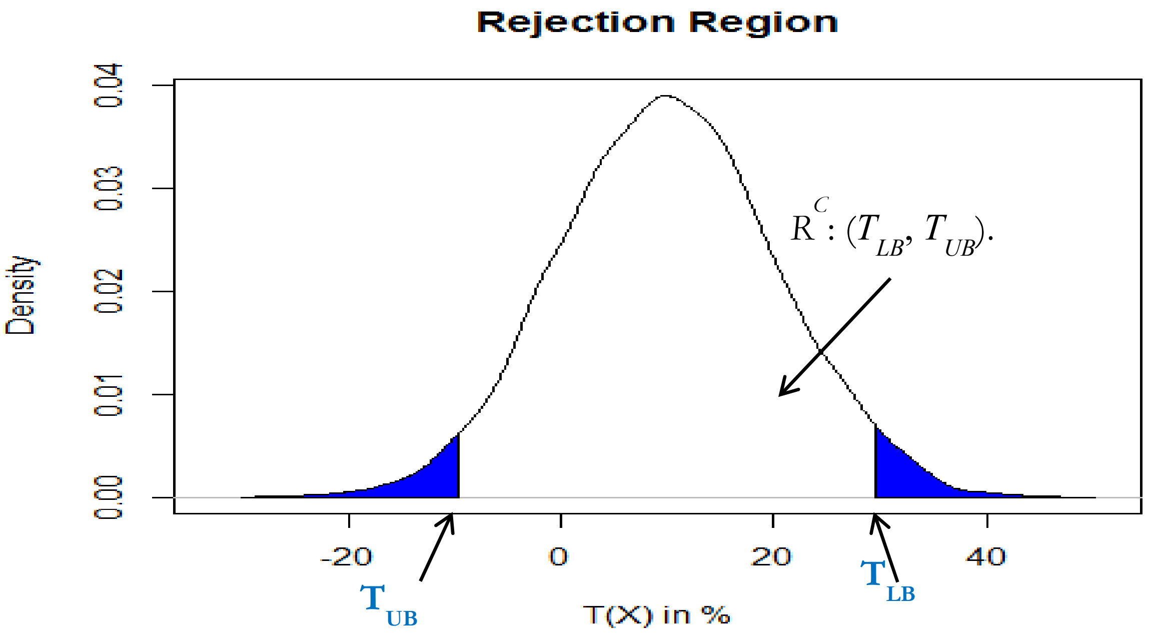
Calculate $T(X) = \bar{X}$

Now, we need to determine the rejection region, R, such that if

$$T(X) = \bar{X} \notin [T_{LB}, T_{UB}] \qquad \Rightarrow \text{Reject } H_0\text{: } \mu_{IBM} = 10\%.$$

That is,

$$R = [\bar{X} < T_{LB}, T_{UB} > \bar{X}]$$



Hypothesis Testing: Steps

We present the *classical approach*, a synthesized approach, known as *significance testing*. It relies on Fisher's *p-value*: the probability, of observing a result at least as extreme as the test statistic, under H_0.
We follow these steps:
1. Identify H_0 & decide on a *significance level* (α%) to compare your test results.
2. Determine the appropriate test statistic $T(X)$ and its distribution under the assumption that H_0 is true.
3. Calculate $T(X)$ from the data.
4. Rule: Reject H_0 if the *p-value* is sufficiently small, that is, we consider $T(X)$ in R (we learn). Otherwise, we reach no conclusion (no learning).

- Q: What *p-value* is "sufficiently small" as to warrant rejection of H_0?

Rule: If *p-value* $< \alpha$ (say, 5%) $\Longrightarrow$ test result is *significant:* Reject H_0.
If the results are "*not significant*," no conclusions are reached (no learning here). Go back gather more data or modify model.

- The father of this approach, Ronald Fisher, favored 5% or 1%.

Example: From the U.S. Jury System
H_0: The defendant is not guilty
H_1: The defendant is guilty

In statistics we learn when we reject. In this case, we learn a defendant is guilty when the jury finds the defendant guilty, by rejecting H_0.

Example: From the U.S. Jury System
1. Identify H_0 & decide on a *significance level* (α%)
H_0: The defendant is not guilty
H_1: The defendant is guilty
Significance level α = "*beyond reasonable doubt*," presumably small level.

2. After judge instructions, each juror forms an "innocent index" $T(X)_i$.

3. Through deliberations, jury reaches a conclusion $T(X) = \sum_{i=1}^{12} T(X)_i$.

4. Rule: If *p-value* of $T(X) < \alpha \Rightarrow$ Reject H_0. That is, guilty!
If *p-value* of $T(X) > \alpha \Rightarrow$ Fail to reject H_0. That is, non-guilty.
Alternatively, we build a rejection region around H_0.

Note: Mistakes are made. We want to quantify these mistakes.

- Failure to reject H_0 does not necessarily mean that the defendant is not guilty, or rejecting H_0 does not mean necessarily the defendant is guilty. *Type I error* and *Type II error* give us an idea of both mistakes.

Definition: Type I and Type II errors
A *Type I error* is the error of rejecting H_0 when it is true. A *Type II error* is the error of "accepting" H_0 when it is false (that is, when H_1 is true).

Notation: Probability of Type I error: a = $P[X \in R|H_0]$
Probability of Type II error: b = $P[X \in R^C|H_1]$

	State of World	
Decision	H_0 true	H_1 true (H_0 false)
Cannot reject ("accept") H_0	Correct decision	Type II error
Reject H_0	Type I error	*Correct decision*

Need to control both types of error:
$\alpha = P(\text{rejecting } H_o|H_o)$
$\beta = P(\text{not rejecting } H_o|H_1)$

Example: From the U.S. Jury System

Type I error is the error of finding an innocent defendant guilty.
Type II error is the error of finding a guilty defendant not guilty.

• In general, we think *Type I error* is the worst of the two errors, we try to minimize the error of sending to jail an innocent person.

Actually, we would like *Type I error* to be zero. However, the only way to do this (100% of innocent defendants are found not guilty) is to never reject H_0. Then, we maximize *Type II error*.

• There is a clear trade-off between both errors. Traditional view: Set *Type I error* equal to a small number (defined in the U.S. court system as "*beyond reasonable doubt*") and design a test that minimizes *Type II error*.

The usual tests (*t-tests*, *F-tests*, Likelihood Ratio tests) incorporate this traditional view.

Example: We want to test if the mean is equal to μ_0. Then,

1. H_0: $\mu = \mu_0$.

   H_1: $\mu \neq \mu_0$.

2. Appropriate T(X): *t-test* (based on σ unknown and estimated by *s*).
   Determine distribution of T(X) under H_0. Sampling distribution of $\bar{X}$, under H_0:

$$\bar{X} \sim N(\mu_0, \sigma^2/n).$$

Then, distribution of T(X) under H_0:

$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \sim t_n$$ –when $n > 30$, $t_n \sim N(0, 1)$.

3. Compute t, $\hat{t}$, using $\bar{X}$, μ_0, s, and N. Get *p-value*($\hat{t}$).

4. Rule: Set an α level. If *p-value*($\hat{t}$) < α ⇒ Reject H_0: $\mu = \mu_0$.

Alternatively, if $|\hat{t}| > t_{n,\alpha/2}$ (=**1.96**, if α=.05) ⇒ Reject H_0: $\mu = \mu_0$.

Technical Note 1: In step 2, the distribution of the t-test, t, is exact if {X} follows a normal distribution, otherwise, the distribution is asymptotic (for this we need a large n); that is

$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \xrightarrow{d} N(0, 1).$$

Technical Note 2: In step 2, we determine the distribution of t, by using the sampling distribution of $\bar{X}$ under H_0. If H_0 is not true, say $\mu = \mu_1$, then

$$\bar{X} \sim N(\mu_1, \sigma^2/n),$$

and, thus, t is distributed N(0, 1) only under H_0, since only under H_0 the $E[\bar{X} - \mu_0] = 0$.

Lecture 1 – Review of Linear Algebra

A Matrix

A matrix is a set of elements, organized into rows and columns

$$\begin{array}{cc} & \text{rows} \longrightarrow \\ \text{columns} \downarrow & \begin{bmatrix} a & b \\ c & d \end{bmatrix} \end{array}$$

- *a* and *d* are the diagonal elements.
- *b* and *c* are the off-diagonal elements.

- Matrices are like plain numbers in many ways: they can be added, subtracted, and, in some cases, multiplied and inverted (divided).

Example:

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}; \qquad b = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$$

• Dimensions of a matrix: numbers of rows by numbers of columns. The Matrix **A** is a 2x2 matrix, **b** is a 1x3 matrix.

• A matrix with only 1 column or only 1 row is called a *vector*.

• If a matrix has an equal numbers of rows and columns, it is called a *square* matrix. Matrix **A**, above, is a square matrix.

Usual Notation:	Upper case letters	⇒ matrices
	Lower case	⇒ vectors

Matrices - Information

• Information is described by data. A tool to organize the data is a list, which we call a vector. Lists of lists are called matrices. That is, we organize the data using matrices.

• We think of the elements of **X** as data points ("data entries", "observations"), in economics, we usually have numerical data.

• We store the data in rows. In a *T*x*k* matrix, **X**, over time we build a database:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$$

- Once the data is organized in matrices it can be easily manipulated: multiplied, added, etc. (this is what Excel does).

- In econometrics, we have a model $y = f(x_1, x_2, \ldots, x_k)$, which we want to estimate. We collect data, say T (or N) observations, on a dependent variable, **y**, and on k explanatory variables, **X**.

- Under the usual notation, vectors will be column vectors: **y** and $\boldsymbol{x_k}$ are Tx1 vectors:

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{y_1} \\ \vdots \\ \boldsymbol{y_T} \end{bmatrix} \quad \& \quad \boldsymbol{x}_j = \begin{bmatrix} \boldsymbol{x_{j1}} \\ \vdots \\ \boldsymbol{x_{jT}} \end{bmatrix} \quad j = 1,\ldots, k$$

X is a Txk matrix: $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$

Its columns are the k Tx1 vectors $\boldsymbol{x_j}$. It is common to treat $\boldsymbol{x_1}$ as vector of ones, **ί.**

Special Matrices – Identity and Null

- *Identity Matrix:* A square matrix with 1's along the diagonal and 0's everywhere else. Similar to scalar "1."

$$\boldsymbol{I} = \begin{bmatrix} \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix}$$

- *Null matrix:* A matrix in which all elements are 0's. Similar to scalar "0."

$$\mathbf{0} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

- Both are *diagonal* matrices ⇒ off-diagonal elements are zero.

Note: Both are examples of *symmetric* and *idempotent* matrices. As we will see later:

- Symmetric: $A = A^T$
- Idempotent: $A = A^2 = A^3 = \ldots$

Elementary Row Operations

Elementary row operations:

– Switching: Swap the positions of two rows
– Multiplication: Multiply a row by a non-zero scalar
– Addition: Add to one row a scalar multiple of another.

- An *elementary matrix* is a matrix which differs from the identity matrix by one single elementary row operation.
- If the matrix subject to elementary row operations is associated to a system of linear equations, then these operations do not change the solution set. Row operations can make the problem easier.

• Elementary row operations are used in Gaussian elimination to reduce a matrix to row echelon form.

Matrix multiplication: Details

Multiplication of matrices requires a *conformability condition*

• The conformability condition for multiplication is that the column dimensions of the lead matrix **A** must be equal to the row dimension of the lag matrix **B**.

• If **A** is an (*m*x*n*) and **B** an (*n*x*p*) matrix (**A** has the same number of columns as **B** has rows), then we define the product of **AB**. **AB** is (*m*x*p*) matrix with its ij-th element is

• What are the dimensions of the vector, matrix, and result?

$$aB = \begin{bmatrix} a_{11} a_{12} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & {}_{22} & b_{23} \end{bmatrix} = c = \begin{bmatrix} c_{11} & c_{12} & c_{13} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \end{bmatrix}$$

Dimensions: a(1x2), B(2x3) $\Rightarrow$ c(1x3)

Transpose Matrix

The transpose of a matrix **A** is another matrix $\mathbf{A}^{\mathrm{T}}$ (also written **A′**) created by any one of the following equivalent actions:

–write the rows (columns) of **A** as the columns (rows) of $\mathbf{A}^{\mathrm{T}}$

–reflect **A** by its main diagonal to obtain $\mathbf{A}^{\mathrm{T}}$

Formally, the (*i*,*j*) element of $\mathbf{A}^{\mathrm{T}}$ is the (*j*,*i*) element of **A**:

$[\boldsymbol{A}^{\mathrm{T}}]_{ij} = [\boldsymbol{A}]_{ji}$

If **A** is a $m \times n$ matrix $\Rightarrow$ $\mathbf{A}^{\mathrm{T}}$ is a $n \times m$ matrix.

(**A**')' = **A**

Conformability changes unless the matrix is square.

$$\text{Example}: \quad A = \begin{bmatrix} 3 & 8 & -9 \\ 1 & 0 & 4 \end{bmatrix} \quad => A' = \begin{bmatrix} 3 & 1 \\ 8 & 0 \\ -9 & 4 \end{bmatrix}$$

Example: In econometrics, an important matrix is **X'X**. Recall **X**:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix} \qquad \text{a } (T\text{x}k) \text{ matrix}$$

Then,

$$\boldsymbol{X'} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{1T}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{k1}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix} \qquad \text{a } (k\text{x}T) \text{ matrix}$$

Basic Operations

Addition, Subtraction, Multiplication

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

Just add elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

Just subtract elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

Multiply each row by each column and add

$$k\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$$

Multiply each row by the scalar

Example:

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 7 & 11 \end{bmatrix}$$

Addition

$$A_{2x2} + B_{2x2} = C_{2x2}$$

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 6 \end{bmatrix}$$

Subraction

$$\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} \text{x} \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 26 & 27 \end{bmatrix}$$

Multiplication

$$A_{2x2} \text{x} B_{2x2} = C_{2x2}$$

$$\frac{1}{8}\begin{bmatrix} 2 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 \\ 3/4 & 1/8 \end{bmatrix}$$

Scalar Multiplication

Basic Matrix Operations: X′X

A special matrix in econometrics, **X′X** (a *k*x*k* matrix):

Recall **X** (*T*x*k*): $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$ & $\boldsymbol{X}' = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{1T}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{k1}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$

$$X'X = \begin{bmatrix} \sum_{i=1}^{T} x_{1i}^2 & \cdots & \sum_{i=1}^{T} x_{1i}x_{ki} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{T} x_{ki}x_{1i} & \cdots & \sum_{i=1}^{T} x_{ki}^2 \end{bmatrix} =$$

$$= \sum_{i=1}^{T} \begin{bmatrix} x_{1i}^2 & \cdots & x_{1i}x_{ki} \\ \vdots & \ddots & \vdots \\ x_{ki}x_{1i} & \cdots & x_{ki}^2 \end{bmatrix} =$$

$$= \sum_{i=1}^{T} \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ki} \end{bmatrix} [x_{1i} \quad \cdots \quad x_{ki}] = \sum_{i=1}^{T} x_i x_i{}'$$

Basic Matrix Operations: ί′X

Recall ί is a column vector of ones (in this case, a *T*x1 vector):

$$ί = \begin{bmatrix} 1 \\ 1 \\ \ldots \\ 1 \end{bmatrix}$$

Given X (*T*x*k*), then **ί' X** is a 1x*k* vector:

$$ί'X = [1 \quad \ldots \quad 1] \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{1T} & \cdots & x_{kT} \end{bmatrix} = [\sum_{t=1}^{T} x_{1t} \quad \ldots \quad \sum_{t=1}^{T} x_{kt}]$$

Note: If x_1 is a vector of ones (representing a constant in the linear classical model), then:

ί' $\mathbf{x_1} = \sum_{t=1}^{T} x_{1t} = \sum_{t=1}^{T} 1 = T$(“*dot product*”)

Inverse of a Matrix

Identity matrix: **AI = A**

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Notation: $\mathbf{I_j}$ is a *j*x*j* identity matrix.

Given **A** (*m*x*n*), the matrix **B** (*n*x*m*) is a *right-inverse* for **A** iff

AB = $\mathbf{I_m}$

Given **A** (*m*x*n*), the matrix **C** (*m*x*n*) is a *left-inverse* for **A** iff

CA = $\mathbf{I_n}$

Theorem: If **A** (*m*x*n*), has both a *right-inverse* **B** and a *left-inverse* **C**, then **C** = **B**.

Proof:

We have **AB** = $\mathbf{I_m}$ and **CA** = $\mathbf{I_n}$.

Thus,

C(**AB**) = **C** $\mathbf{I_m}$ = **C** and **C**(**AB**) = (**CA**)**B** = $\mathbf{I_n}$**B** = **B**

⇒ **C**(*n*x*m*)= **B**(*m*x*n*)

Note:

- This matrix is unique. (Suppose there is another left-inverse **D**, then **D** = **B** by the theorem, so **D** = **C**).
- If **A** has both a right and a left inverse, it is a square matrix. It is usually called *invertible*. We say “the matrix **A** is *non-singular*.”

• Inversion is tricky:
$(\mathbf{ABC})^{-1} = \mathbf{C}^{-1}\,\mathbf{B}^{-1}\,\mathbf{A}^{-1}$

Theorem: If **A** (mxn) and **B** (nxp) have inverses, then **AB** is invertible and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

Proof:

We have $\mathbf{AA}^{-1} = \mathbf{I_m}$ and $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I_n}$
$\mathbf{BB}^{-1} = \mathbf{I_n}$ and $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I_p}$

Thus,

$\mathbf{B}^{-1}\mathbf{A}^{-1}(\mathbf{AB}) = \mathbf{B}^{-1}\,(\mathbf{A}^{-1}\mathbf{A})\,\mathbf{B} = \mathbf{B}^{-1}\,\mathbf{I_n}\,\mathbf{B} = \mathbf{B}^{-1}\,\mathbf{B} = \mathbf{I_p}$
$(\mathbf{AB})\,\mathbf{B}^{-1}\mathbf{A}^{-1} = \mathbf{A}\,(\mathbf{BB}^{-1})\,\mathbf{A}^{-1} = \mathbf{A}\,\mathbf{I_n}\,\mathbf{A}^{-1} = \mathbf{A}\,\mathbf{A}^{-1} = \mathbf{I_m}$
$\Rightarrow$ **AB** is invertible and $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$

Note: It is not possible to divide one matrix by another. That is, we can not write A/B. For two matrices **A** and **B**, the quotient can be written as $\mathbf{AB}^{-1}$ or $\mathbf{B}^{-1}\mathbf{A}$.

• In general, in matrix algebra $\mathbf{AB}^{-1} \neq \mathbf{B}^{-1}\mathbf{A}$.

Thus, writing A/B does not clearly identify whether it represents **AB**-1 or **B**-1**A**.
We'll say $\mathbf{B}^{-1}$ post-multiplies **A** (for $\mathbf{AB}^{-1}$) and $\mathbf{B}^{-1}$ pre-multiplies **A** (for $\mathbf{B}^{-1}\mathbf{A}$)

Transpose and Inverse Matrix

$(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$

If $\mathbf{A}' = \mathbf{A}$, then **A** is called a *symmetric* matrix.

Theorems:

- Given two conformable matrices **A** and **B**, then $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$
- If **A** is invertible, then $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$ (and **A**' is also invertible).

Properties of Symmetric Matrices

Definition:

If $\mathbf{A}' = \mathbf{A}$, then **A** is called a *symmetric* matrix.

• In many applications, matrices are often symmetric. For example, in statistics the *correlation matrix* and *the variance covariance matrix*.

• Symmetric matrices play the same role as real numbers do among the complex numbers.

• We can do calculations with symmetric matrices like with numbers: for example, we can solve $\mathbf{B}^2 = \mathbf{A}$ for **B** if **A** is symmetric matrix (& **B** is square root of A.) This is not possible in general.

Theorems:

- If **A** and **B** are nxn symmetric matrices, then $(\mathbf{AB})' = \mathbf{BA}$
- If **A** and **B** are nxn symmetric matrices, then $(\mathbf{A}+\mathbf{B})' = \mathbf{B}+\mathbf{A}$
- If **C** is any nxn matrix, then $\mathbf{B} = \mathbf{C}'\mathbf{C}$ is symmetric.
- (*Spectral decomposition*) If **A** is nxn symmetric matrix, then it can be diagonalized as $\mathbf{B} = \mathbf{X}^{-1}\mathbf{AX}$, with an orthogonal **X**.

• Useful symmetric matrices:

$\mathbf{V} = \mathbf{X'X}$
$\mathbf{P} = \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'}$ **P**: Projection matrix
$\mathbf{M} = \mathbf{I} - \mathbf{P} = \mathbf{I} - \mathbf{X}(\mathbf{X'X})^{-1}\mathbf{X'}$ **M**: Residual maker
$\text{Var}[\mathbf{b}] = \boldsymbol{\sigma}^2\,(\mathbf{X'X})^{-1}$ OLS Variance of **b**

Application 1: Linear System

There is a functional form relating a dependent variable, y, and *k* explanatory variables, **X**. The functional form is linear, but it depends on *k* unknown parameters, **β**. The relation between y and **X** is not exact. There is an error, **ε**. We have *T* observations of y and **X**.

- Then, the data is generated according to:

$$y_i = \Sigma_{j=1,..k}\, x_{k,i}\, \beta_k + \varepsilon_i \qquad i = 1, 2, ..., T.$$

Or using matrix notation:

$$\mathbf{y} = \mathbf{X}\, \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where **y** & **ε** are (Tx1); **X** is (Txk); and **β** is (kx1).

- We will call this relation *data generating process* (DGP).
- The goal of econometrics is to estimate the unknown vector **β**.

- Assume an economic model as system of linear equations with:

a_{ij} parameters, where i = 1,.., m rows, j = 1,.., n columns
x_i endogenous variables (n),
d_i exogenous variables and constants (m).

$$\begin{cases} a_{11}\, x_1 + a_{12}\, x_2 + \dots + a_{1n}\, x_n = d_1 \\ a_{21}\, x_1 + a_{22}\, x_2 + \dots + a_{2n}\, x_n = d_2 \\ \dots . \quad .. .. \quad . \dots \quad \dots \\ a_{m1}\, x_1 + a_{m2}\, x_2 + \dots + amn\, x_n = d_m \end{cases}$$

We can write this system using linear algebra notation: **A x** = ***d***

$$\begin{bmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{m1}} & \cdots & \boldsymbol{a_{mn}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1} \\ \dots \\ \boldsymbol{x_n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{d_1} \\ \dots \\ \boldsymbol{d_m} \end{bmatrix}$$

d = column vector

A = (mxn) matrix

x = column vector

- Summary: System of linear equations:

$$\mathbf{Ax} = \boldsymbol{d}$$

where

A = (mxn) matrix of parameters
x = column vector of endogenous variables (nx1)
d = column vector of exogenous variables and constants (mx1)

Solve for x*.

- Questions:

- For what combinations of **A** and ***d*** there will zero, one, many or an infinite number of solutions?
- How do we compute (characterize) those sets of solutions?

Theorem: Given **A** (mxn) invertible. Then, the equation **Ax** = ***d*** has one and only one solution for every ***d*** (mx1).

Linear dependence and Rank: Example

A set of vectors is *linearly dependent* if any one of them can be expressed as a linear combination of the remaining vectors; otherwise, it is linearly independent.

- Formal definition: Linear independence (LI)

The set $\{u_1, ..., u_k\}$ is called a *linearly independent* set of vectors iff

$$c_1\, u_1 + ... + c_k\, u_k = \theta \;\Rightarrow\; c_1 = c_2 = ... = c_{k,} = 0.$$

Notes:

- Dependence prevents solving a system of equations. More unknowns than independent equations.
- The number of linearly independent rows or columns in a matrix is the *rank* of a matrix (rank(**A**)).

Examples:

$$v_1' = \begin{bmatrix} 5 & 12 \end{bmatrix}$$

$$v_2' = \begin{bmatrix} 10 & 24 \end{bmatrix}$$

$$A = \begin{bmatrix} 5 & 10 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$$

$$2v_1' - v_2' = 0' \qquad \Rightarrow rank\ (A) = 1$$

$$v_1 = \begin{bmatrix} 2 \\ 7 \end{bmatrix}; v_2 = \begin{bmatrix} 1 \\ 8 \end{bmatrix}; v_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}; \qquad A = \begin{bmatrix} 2 & 1 & 4 \\ 7 & 8 & 5 \end{bmatrix}$$

$$3v_1 - 2v_2$$

$$= \begin{bmatrix} 6 & 21 \end{bmatrix} - \begin{bmatrix} 2 & 16 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 5 \end{bmatrix} = v_3$$

$$3v_1 - 2v_2 - v_3 = 0 \quad \Rightarrow rank(A) = 2$$

A matrix **A** has *full row* rank when each of the rows of the matrix are linearly independent and *full column rank* when each of the columns of the matrix are linearly independent. For a square matrix these two concepts are equivalent and we say matrix **A** has full rank.

Determinant Test

We can check if a matrix square matrix A has full rank, that is, all its rows/columns are linearly independent by computing the determinant. If a square matrix **A** has full rank, it is invertible. That is, the *determinant* of a square matrix **A** detects whether **A** is invertible:

If det(**A**)=0 then **A** is not invertible (equivalently, the rows/columns of **A** are linearly dependent).

Lecture 2 - Introduction: Review, Returns and Data

• All the information and material is on my webpage:
https://www.bauer.uh.edu/rsusmel/4397/4397.htm

• Textbook:
Required: **Introductory Econometrics for Finance**, Cambridge University Press; 4th edition or older, by Chris Brooks.

Recommended: **R Guide for Introductory Econometrics for Finance,** written by Chris Brooks. You can download it from my homepage (pdf format). It's also available for free through Amazon (kindle format).

• Install R in your machine. We will run programs and do some simple programing.

• Two midterms and a final (optional paper for MBA/MS class). There is a project in between midterms.

• Three homework: Two before first Midterm and one before second Midterm.

This Class

This is an applied technical class, with some econometric theory and concepts, followed by related financial applications.

• We will review many math and statistical topics.

• Some technical material may be new to you, for example Linear Algebra. The new material is introduced to simplify exposition of main concepts. You will not be required to have a deep understanding of the new material, but you should be able to follow the intuition behind it.

• This is not a programming class, but we will use R to estimate models. I will cover some of the basics in class. But, the more you know, the more comfortable you will be running the programs.

• For some students, the class will be dry ("*He fries my brain*," a student said last semester.)

• Main Topics

- How do we measure returns and risks of financial assets?
- Can we estimate expected returns with precision? What about the variance of returns?
- Can we explain asset returns?
- How can one explain variations in stock returns across various stocks?
- Are asset returns predictable? In the short run? In the long run?
- Does the risk of an asset vary with time? What are the implications? How can one model time-varying risk?
- Is the equity premium (excess returns of stocks over bonds) really that high?

• Topics Not Covered

This course provides an introduction to the basics of financial econometrics, focusing on estimation of linear models and analysis of time series. There are many more topics in financial econometrics that we will not cover, among them:

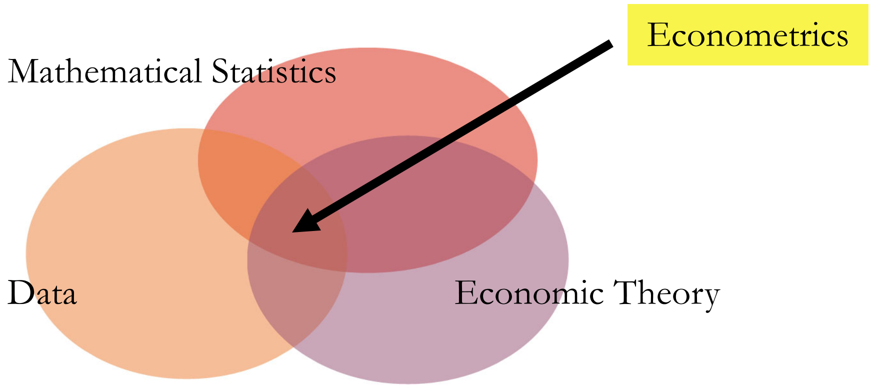
- Credit risk management and probability of default
- Interest rate models and term structure models
- Analyzing high-frequency data and modeling market microstructure
- Estimating models for options
- Multivariate time series models
- Technical methods such as state-space models and the Kalman filter, Markov processes, copulae, nonparametric methods, etc.

What is Econometrics?

Ragnar Frisch, Econometrica Vol.1 No. 1 (1933) revisited

"Experience has shown that each of these three view-points, that of *statistics*, *economic theory*, and *mathematics*, is a necessary, but not by itself a sufficient, condition for a real understanding of the quantitative relations in modern economic life.

It is the unification of all three aspects that is powerful. And it is this unification that constitutes econometrics."



Financial Econometrics is applied econometrics to financial data. That is, we study the statistical tools that are needed to analyze and address the specific types of questions and modeling challenges that appear in analyzing financial data.

Finance is about the trade-off between risk and return.

– How do we measure risk and return?
– Can we predict them?
– How do we measure the trade-off?
– How much should I be compensated for taking a given risk?

Thus, we will be concerned with quantifying rewards and risks associated with uncertain outcomes.

Review – Population and Sample

Definition: Population

A population is the totality of the elements under study. We are interested in learning something about this population.

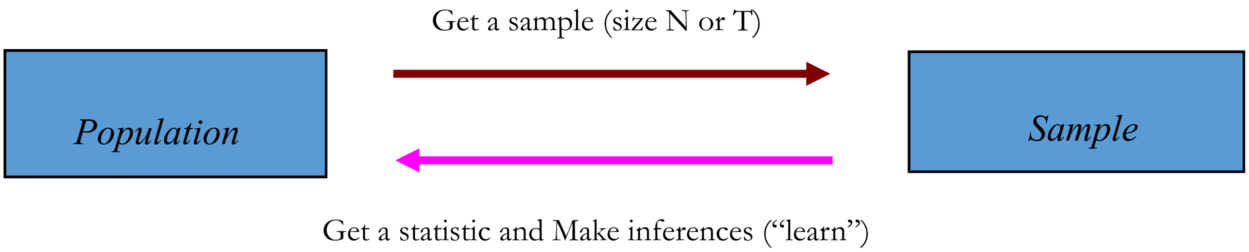
Examples: Number of alligators in Texas, percentage of unemployed workers in cities in the U.S., the total return of all stocks in the U.S., the 10-year Japanese government bond yield from 1960-2021. ¶

A Random Variable (RV) *X* defined over a population is called the population RV. The population RV generates the data. We call the population RV the "*Data Generating Process*," or DGP.

Usually, the population is large, making a complete enumeration of all the values in the population impractical or impossible. Thus, the descriptive statistics describing the population – i.e., the *population parameters*– will be considered unknown.

Typical situation in statistics: we want to make inferences about an unknown population parameter θ using a sample –i.e., a small collection of observations from the general population- $\{X_1, \dots, X_n\}$.

We summarize the information in the sample with a *statistic*, which is a function of the sample. That is, any statistic summarizes the data, or reduces the information in the sample to a single number. To make inferences, we use the information in the statistic instead of the entire sample.


Get a sample (size N or T)
Population
Sample
Get a statistic and Make inferences ("learn")

Definition: Sample
The *sample* is a (manageable) subset of elements of the population.

Example: The total returns of the stocks on the S&P 500 index. ¶

Samples are collected to learn about the population. The process of collecting information from a sample is referred to as *sampling*.

Definition: Random Sample
A *random sample* is a sample where the probability that any individual member from the population being selected as part of the sample is exactly the same as any other individual member of the population.

Example: The total returns of the stocks on the S&P 500 index is *not* a random sample. ¶

In mathematical terms, given a random variable *X* with distribution *F*, a *random sample* of length *n* is a set of *n* independent, identically distributed (*i.i.d.*) random variables with distribution *F*.

Review – Samples and Types of Data

The samples we collect to learn about the population by computing sample statistics are classified in three groups:

- Time Series Data: Collected over time on one or more variables, with a particular *frequency* of observation. For example, monthly S&P 500 returns, or 10' IBM returns.

- Cross-sectional Data: Collected on one or more variables collected at a single point in time. For example, today we record all closing returns for the member of the S&P 500 index.

- Panel Data: Cross-sectional Data collected over time. For example, the CRSP database collects all stocks traded in U.S. markets at the daily frequency since 1962.

Review – Sample Statistic

A *statistic* (singular) is a single measure of some attribute of a sample (for example, its arithmetic mean value). It is calculated by applying a function (statistical algorithm) to the values of the items comprising the sample, which are known together as a set of data.

Definition: Statistic
A *statistic* is a function of the observable random variable(s), which does not contain any unknown parameters.

Examples: Sample mean ($\bar{X}$), sample variance (s^2), minimum, median, (x_1+x_n)/2, etc. ¶

Note: A statistic is distinct from a population parameter. A statistic will be used to estimate a population parameter. In this case, the statistic is called an *estimator*.

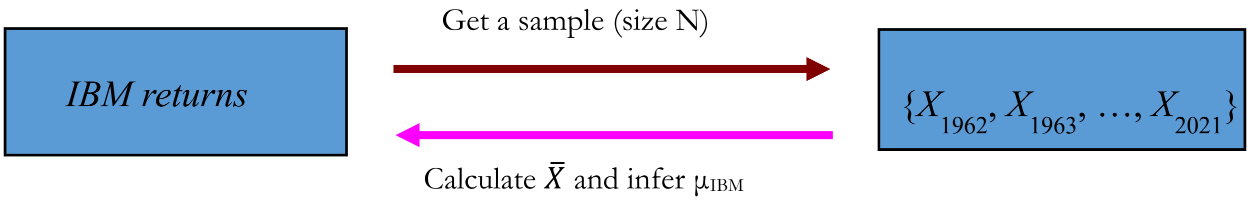
Review – Population and Sample

Sample Statistics are used to estimate population parameters.

Example: $\bar{X}$ is an estimate of the population mean, μ. ¶

Notation: Population parameters: Greek letters (μ, σ, θ, etc.)
Estimators: A hat over the Greek letter ($\hat{\theta}$).

Suppose we want to learn about the mean of IBM annual returns, μ_{IBM}. From the population, we get a sample: $\{X_{1962}, X_{1963}, \ldots, X_{n=2021}\}$. Then, we compute a statistic, $\bar{X}$. As we will see later, on average $\bar{X}$ is a good estimator of μ.


IBM returns

Get a sample (size N)

$\{X_{1962}, X_{1963}, \ldots, X_{2021}\}$

Calculate $\bar{X}$ and infer μ_{IBM}

The definition of a sample statistic is very general. For example, $(x_1+x_n)/2$ is by definition a statistic; we could claim that it estimates the population mean of the variable X. However, this is probably not a good estimate.

We would like our estimators to have certain desirable properties.

Review – Sample Statistic

Some simple properties for estimators:

- An estimator $\hat{\theta}$ is *unbiased* estimator of θ if $E[\hat{\theta}] = \theta$.
- An estimator is *most efficient* if the variance of the estimator is minimized.
- An estimator is BUE, or Best Unbiased Estimate, if it is the estimator with the smallest variance among all unbiased estimates.
- An estimator is *consistent* if as the sample size, n, increases to ∞, $\hat{\theta}_n$ converges to θ. We write

$$\hat{\theta}_n \xrightarrow{p} \theta. \qquad \text{(A LLN is behind this result.)}$$

- An estimator is *asymptotically normal* if as the sample size, n, increases to ∞, $\hat{\theta}_n$, often standardized or transformed, converges in distribution to a Normal distribution. We write

$$\hat{\theta}_n \xrightarrow{d} N(\theta, Var(\hat{\theta}_n)). \qquad \text{(A CLT is behind this result.)}$$

Review – PDF for a Continuous RV

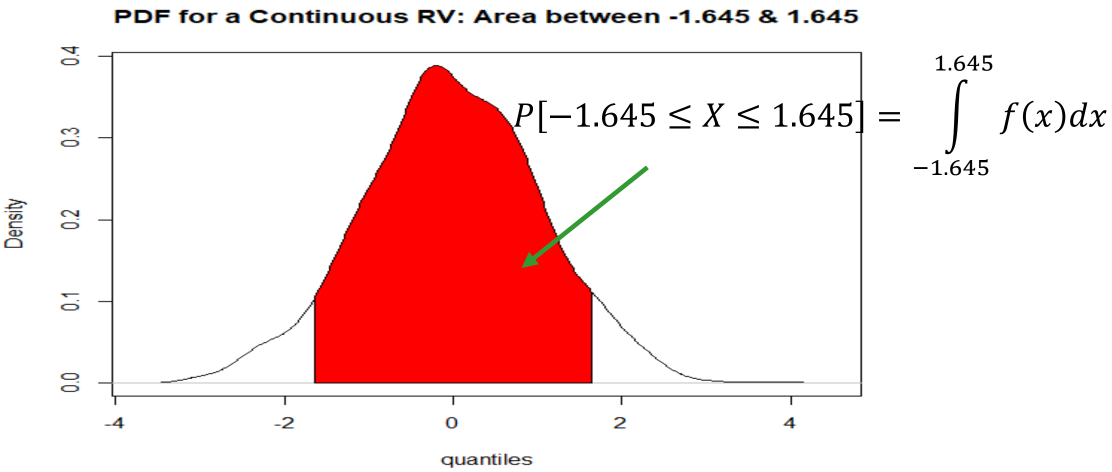
Definition: Suppose that X is a random variable. Let $f(x)$ denote a function defined for $-\infty < x < \infty$ with the following properties:

1. $f(x) \geq 0$
2. $\int_{-\infty}^{\infty} f(x)dx = 1.$
3. $P[a \leq X \leq b] = \int_a^b f(x)dx$

Then, $f(x)$ is called the *probability density function* (pdf) of X. The RV X is called *continuous*. We use the pdf to describe the behavior of X.

Analogous definition applies for a discrete RV, where the notation uses $p(x)$ instead.

The pdf is non-negative and integrates to $\int_{-\infty}^{\infty} f(x)dx = 1.$



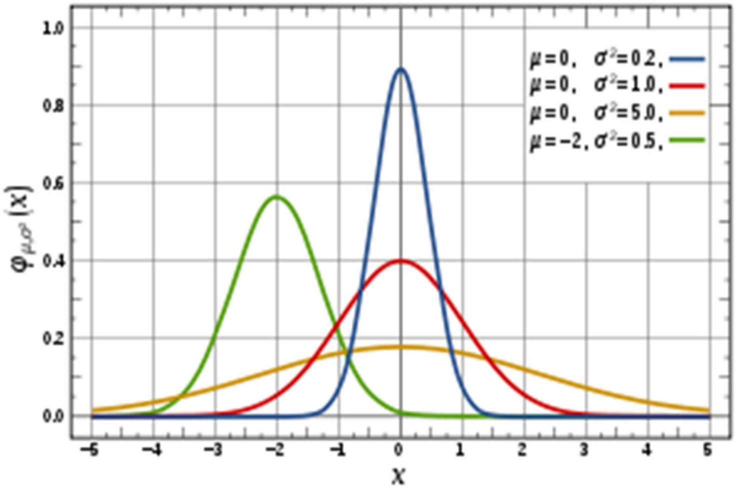
$$P[-1.645 \leq X \leq 1.645] = \int_{-1.645}^{1.645} f(x)dx$$

Remark: We use the pdf to describe the behavior of the RV (discrete or continuous).

Review – Popular PDFs: Normal Distribution

A RV *X* is said to have a *normal distribution* with parameters □ *(mean)* and □□ *(variance)* if *X* is a continuous RV with pdf *f*(*x*):

$$f\left(x\right)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Note: Described by two parameters: *m* and s^2. We write *X* ~ N(μ, σ^2)

The normal distribution is often used to describe or approximate any variable that tends to cluster around the mean. It is the most assumed distribution in economics and finance: rates of return, growth rates, IQ scores, observational errors, etc.

The central limit theorem (CLT) provides a justification for the normality assumption when the sample size, *n,* is large.

Notation: PDF: *X* ~ N(μ, σ^2)
CDF: Φ(*x*)

Review – Popular PDFs: Gamma Distribution

Let the continuous RV X have density function):

$$f(x) = \begin{cases} \dfrac{\lambda^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

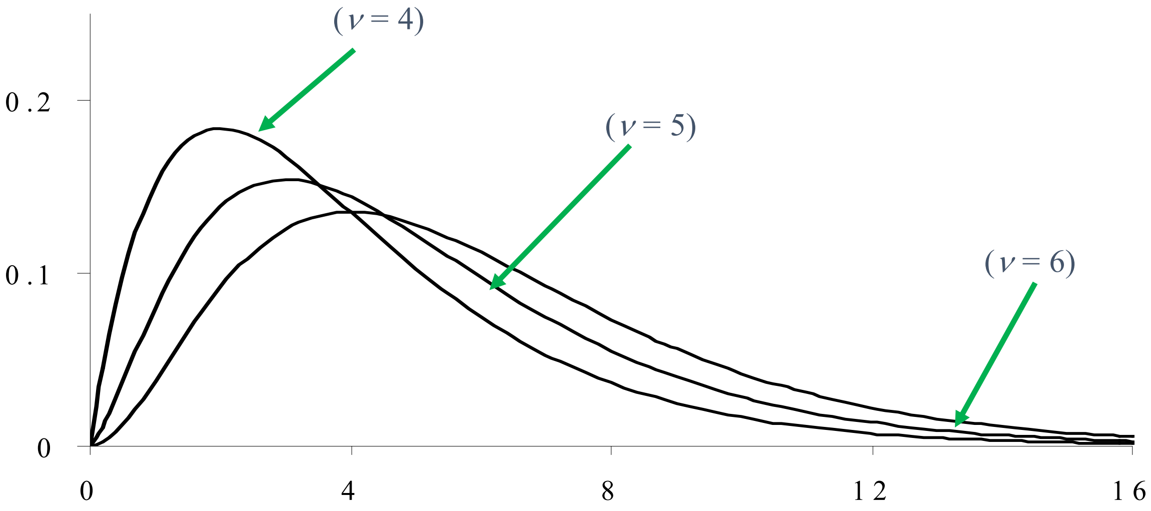
where $a, \lambda > 0$ and Γ(a) is the gamma function evaluated at a.

Then, X is said to have a *Gamma distribution* with parameters a and λ, denoted as $X \sim$ Gamma(a, λ) or $\Gamma(a, \lambda)$.

It is a family of distributions, with special cases:

- Exponential Distribution, or Exp(λ): $a = 1$.
- Chi-square Distribution, or χ^2_ν: $a = n/2$ and $\lambda = ½$.

Below we plot the Chi-square distribution with parameter □, which we refer as degrees of freedom:

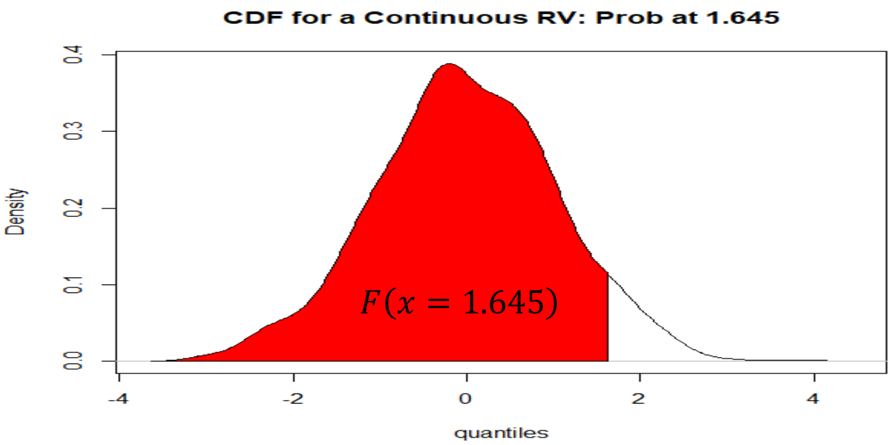


Note: When n is large, the χ^2_ν converges to a N(n, 2n).

Review – CDF for a Continuous RV

If X is a continuous random variable with probability density function, $f(x)$, the *cumulative distribution function* (CDF) of X is given by:

$$F(x) = P[X \leq x] = \int_{-\infty}^{x} f(t)dt$$



Note: The FTC (*fundamental theorem of calculus*) implies:

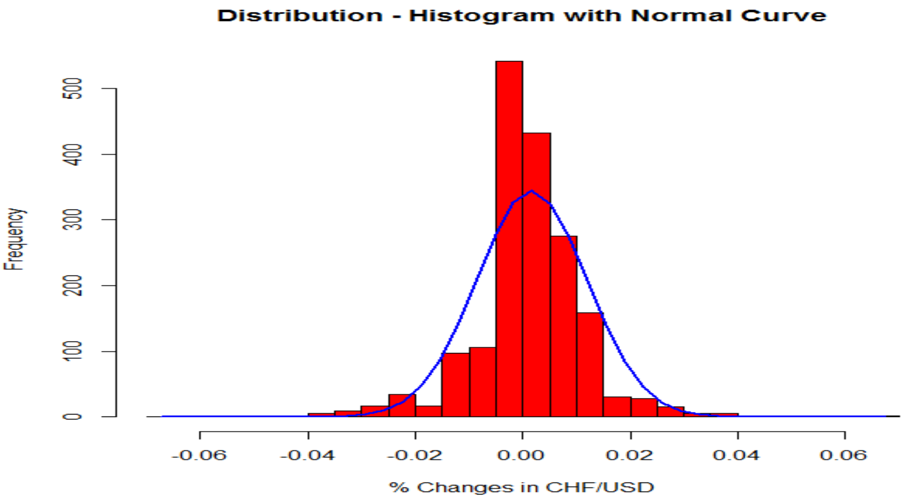
$$F'(x) = \frac{dF(x)}{dx} = f(x)$$

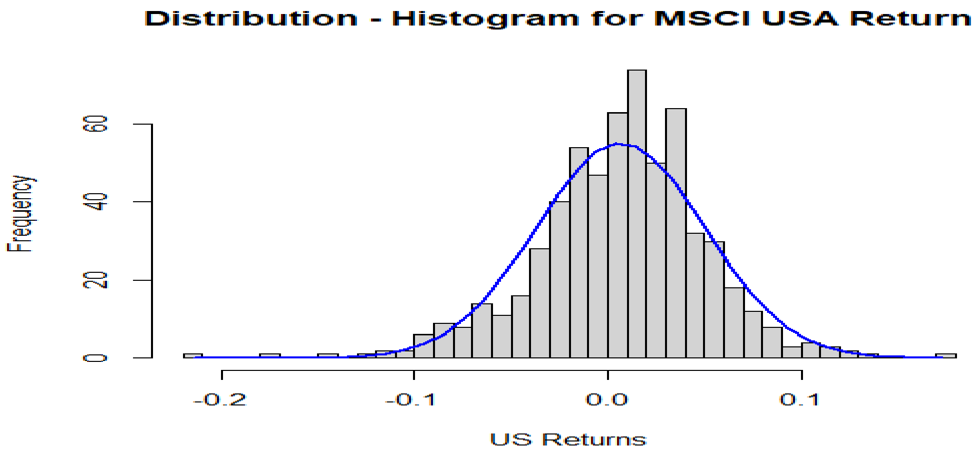
Review – Histogram of a RV

Recall that a *histogram* is an approximate representation of the distribution of numerical data.

Example: We use a histogram to estimate the distribution of a RV. Let X = Percentage changes in the **CHF/USD exchange rate** = e_f and for **MSCI USA Index returns**.

Data: Monthly from January 1971 to June 2020 (N=595 observations) for CHF/USD and January 1970 to June 2020 (N=607).





Note: We overlay a Normal density (blue line) over the histogram. ¶

Review – Moments of Random Variables

The moments of a random variable *X* are used to describe the behavior of the RV (discrete or continuous).

Definition: K^{th} Moment
Let *X* be a RV (discrete or continuous), then the k^{th} *moment of X* is:

$$\mu_k = E(X^k) \ = \begin{cases} \sum_x x^k p(x) & if X\ is\ discrete \\ \int_{-\infty}^{\infty} x^k f(x)dx & if X\ is\ continuous \end{cases}$$

Definition: Central Moments
Let *X* be a RV (discrete or continuous). Then, the k^{th} *central moment* of *X* is defined to be:

$$\mu_k^0 = E[(X-\mu)^k] \ = \begin{cases} \sum_x (x-\mu)^k p(x) & \text{if } X \text{ is discrete} \\ \int_{-\infty}^{\infty} (x-\mu)^k f(x)dx & \text{if } X \text{ is continuous} \end{cases}$$

where $m = m_1 = E(X)$ = the first moment of *X*.

The central moments describe how the probability distribution is distributed about the center of gravity, *m*.

The first central moments is given by:

$$\mu_1^0 = E[X-\mu]$$

The second central moment depends on the *spread* of the probability distribution of *X* about *m*. It is called the variance of *X* and is denoted by the symbol σ^2 = var(*X*):

$$\mu_2^0 = E[(X-\mu)^2] = \text{var(X)} = \sigma^2$$

The square root of var(X) is called the *standard deviation* of *X* and is denoted by the symbol *s* = SD(X). We also refer to it as *volatility*:

$$\sqrt{\mu_2^0} = \sqrt{E[(X-\mu)^2]} = \sigma$$

Review – Moments of a RV: Skewness

The third central moment:

$$\mu_3^0 = E[(X-\mu)^3]$$

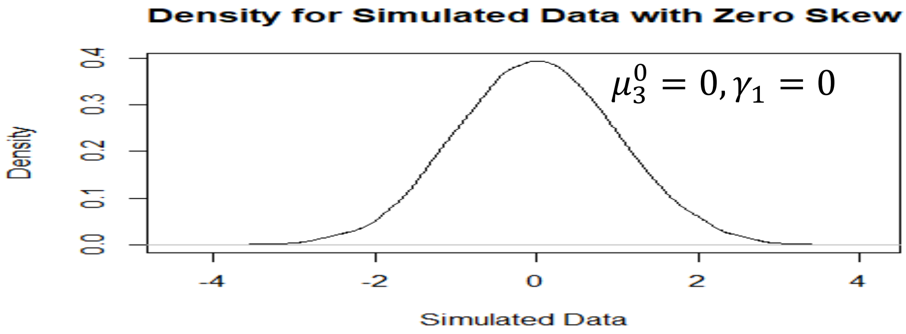
μ_3^0 contains information about the *skewness* of a distribution.

A popular measure of skewness:

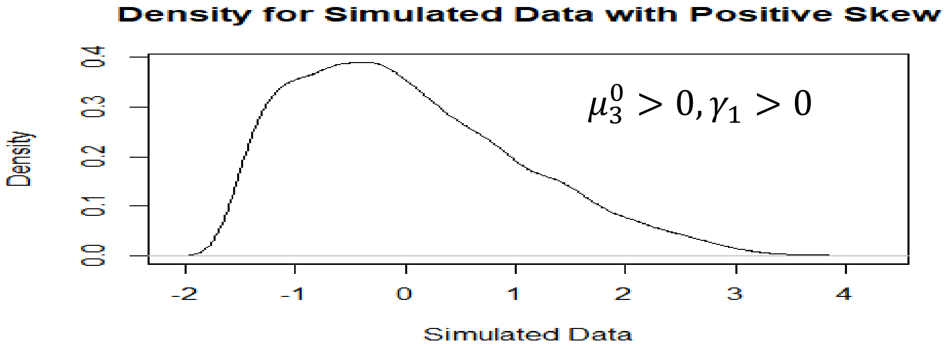
$$\gamma_1 = \frac{\mu_3^0}{\sigma^3} = \frac{\mu_3^0}{(\mu_2^0)^{\frac{3}{2}}}$$

• Distribution according to skewness:

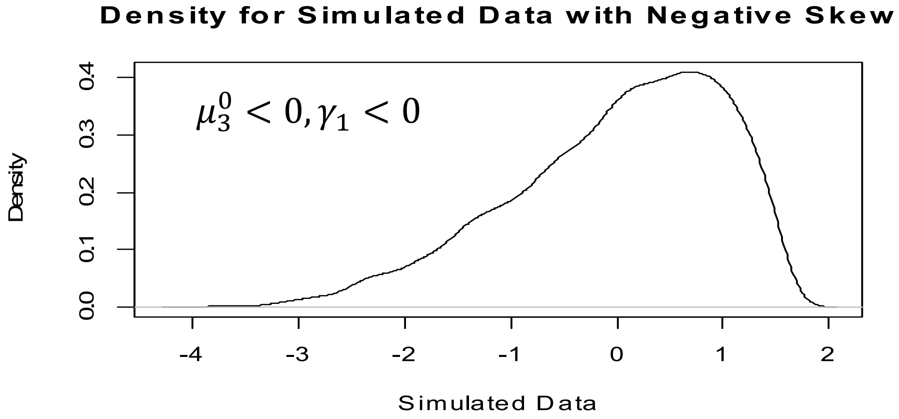
1) Symmetric distribution



2) Positively (right-) skewed distribution (with mode < median < mean)



3) Negatively (left-) skewed distribution (with mode > median > mean)



• Skewness and Economics

- Zero skew means symmetrical gains and losses –i.e., extreme values tend to occur on both sides of the curve on similar proportions.
- Positive skew suggests many small losses and few rich returns –i.e., extreme values tend to occur in the right tail
- Negative skew indicates a lot of minor wins offset by rare major losses –i.e., extreme values tend to occur in the left tail.

In financial markets, stock returns at the firm level show positive skewness, but at the aggregate (index) level show negative skewness.

From horse race betting and from U.S. state lotteries there is evidence supporting the contention that gamblers are not necessarily risk-lovers but skewness-lovers: Long shots are overbet (positive skewness loved!).

Review – Moments of a RV: Kurtosis

The fourth central moment:

$$\mu_4^0 = E[(X-\mu)^4]$$

It contains information about the *shape* of a distribution. The property of shape that is measured by this moment is called *kurtosis,* usually estimated by

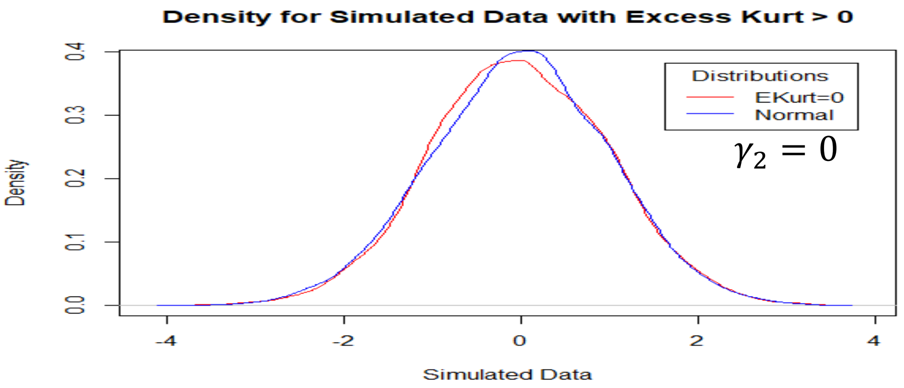
$$\gamma_2 = \frac{\mu_4^0}{\sigma^4}.$$

Kurtosis measures how much weight there is in the tails of the distribution relative to the middle (we call this a measure of the "*fatness*" of the tails). We usually compare the kurtosis of a series relative to the kurtosis of a normal distribution, which is equal to 3. We measure the "excess" fatness of the tail over the normal curve. That is, the *measure of (excess) kurtosis*:

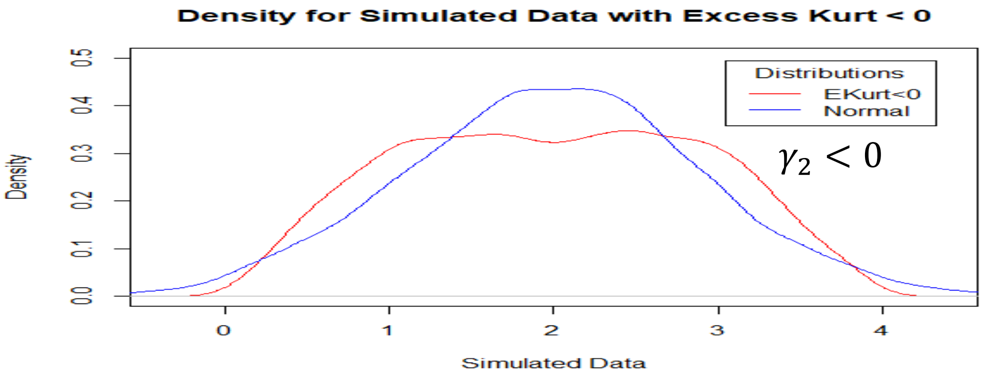
$$\gamma_2 = \frac{\mu_4^0}{\sigma^4} - 3 = \frac{\mu_4^0}{(\mu_2^0)^2} - 3$$

• Distributions:

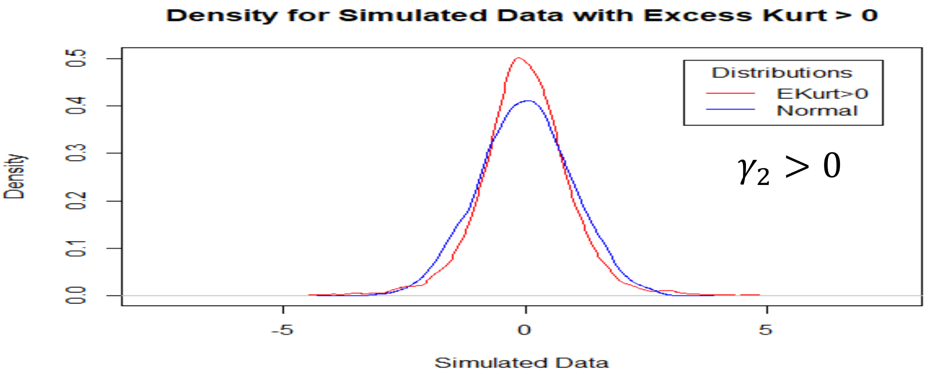
1) Mesokurtic distribution



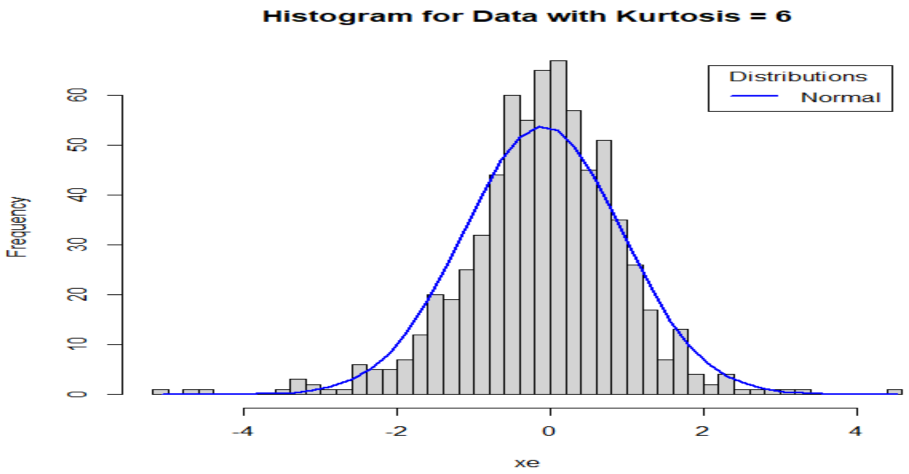
2) Platykurtic distribution



3) Leptokurtic distribution (usual shape for asset returns)



- Positive excess kurtosis, $\gamma_2 > 0$, is the norm for financial returns. Below I simulate a series with μ=0, σ=1, zero skewness & kurtosis = 6 (γ_2=3), overlaid with a standard normal distribution. Fat tails are seen on both sides of the distribution.



Review – Moments and Expected Values

Note that moments are defined by expected values. We define the expected value of a function of a continuous RV X, $g(X)$, as

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

If X is *discrete* with probability function $p(x)$

$$E[g(X)] = \sum_x g(x)p(x) = \sum_i g(x_i)p(x_i)$$

Examples: $g(x) = (x - \mu)^2 \Rightarrow \text{E}[g(x)] = \text{E}[(x - \mu)^2]$

$g(x) = (x - \mu)^k \Rightarrow \text{E}[g(x)] = \text{E}[(x - \mu)^k]$. ¶

We estimate expected values with sample averages. The Law of Large Numbers (LLN) tells us they are *consistent* estimators of expected values.

Review – Estimating Moments

We estimate expected values with sample averages. For example, the first moment, the mean, and the second central moment, the variance, are estimated by:

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$$

$$s^2 = \frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N-1}$$ (N-1 adjustment needed for $\text{E}[s^2] = \sigma^2$)

Besides consistent, they are both are *unbiased* estimators of their respective population moments (unbiased = "on average, I get the population parameter"). That is,

$\text{E}[\bar{X}] = \mu$ "population parameter"

$\text{E}[s^2] = \sigma^2$

Review – Law of Large Numbers (LLN)

Long history: Gerolamo Cardano (1501-1576) stated it without proof. Jacob Bernoulli published a rigorous proof in 1713.

Theorem (Weak LLN)
Let $X_1, \ldots , X_n$ be n mutually independent random variables each having mean m and a finite s - i.e, the sequence $\{X_n\}$ is *i.i.d.*
Let $\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N}$.

Then for any $d > 0$ (no matter how small)

$$P\left[\left|\bar{X}-\mu\right|<\delta\right]=P\left[\mu-\delta<\bar{X}<\mu+\delta\right]\to 1 \text{ as } n\to\infty$$

There are many variations of the LLN. It is a general result: A sample average as the sample size goes to infinite tends to its expected value. Also written as $\bar{X}_{\text{n}} \overset{p}{\rightarrow} \mu$. *(convergence in probability*)

Review – Central Limit Theorem (CLT)

Let $X_1, X_2, \ldots, X_n$ be a sequence of *i.i.d.* RVs with finite mean m, and finite variance s^2. Then as n increases, $\bar{X}_{\text{n}}$, the sample mean, approaches the normal distribution with mean μ and variance s^2/n.
This theorem is sometimes stated as

$$\frac{\sqrt{N}(\bar{X}-\mu)}{\sigma} \overset{d}{\rightarrow} N(0,1)$$

where $\overset{d}{\rightarrow}$ means "the limiting distribution (asymptotic distribution) is" (or *convergence in distribution*).

Many versions of the CLT. This one is the *Lindeberg-Lévy CLT*.

The CLT gives only an asymptotic distribution. We usually take it as an approximation for a finite number of observations. In these cases, the notation goes from $\overset{d}{\rightarrow}$ to $\overset{a}{\rightarrow}$.

Review – Sampling Distributions

All statistics, T(X), are functions of RVs and, thus, they have a distribution. Depending on the sample, we can observe different values for T(X), thus, the finite sample distribution of T(X) is called the *sampling distribution*.

For the sample mean, $\bar{X}$, if the X_{i}'s are normally distributed, then the sampling distribution is normal with mean μ and variance σ^2/N. Or

$$\bar{X} \sim \text{N}(\mu, \sigma^2/N).$$

Note: If the data is not normal, the CLT is used to approximate the sampling distribution by the asymptotic one, usually after some manipulations. Again, in those cases, the notation goes from $\overset{d}{\rightarrow}$ to $\overset{a}{\rightarrow}$.

The SD of the sampling distribution is called the *standard error* (SE). Then, SE($\bar{X}$) = σ/sqrt(N).

For the sample variance σ^2, if the X_i's are normally distributed, then the sampling distribution is derived from this result:

$$(N-1)\, s^2/\sigma^2 \sim \chi^2_{N\text{-}1}.$$

It can be shown that the $\chi^2_{N\text{-}1}$ has a variance equal to 2 times the degrees of freedom (=2*(N – 1)), that is,

$$\text{Var}[(N\text{-}1)\, s^2/\sigma^2\,] = 2 * (N-1) \qquad \Rightarrow \text{Var}[s^2] = 2 * \sigma^4 /(N-1)$$

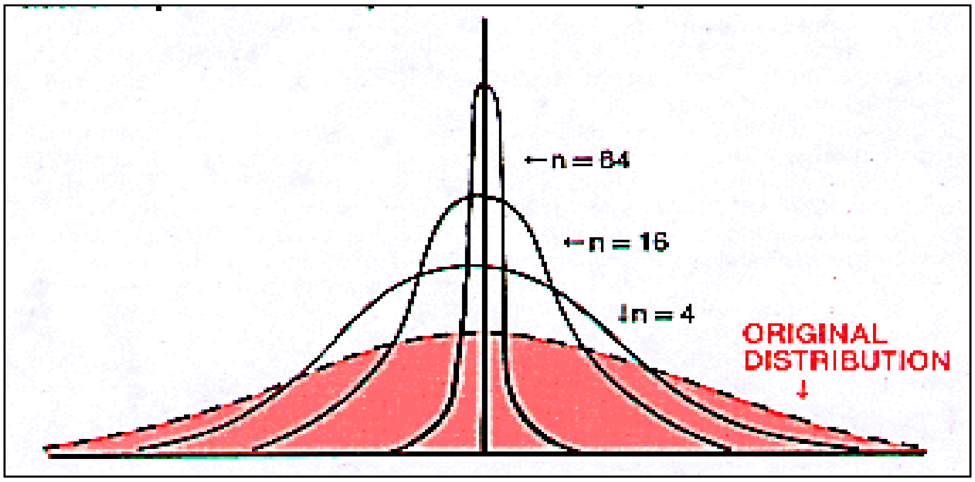
Then, SE(s^2) = SD(s^2) = σ^2 * sqrt[2/(N – 1)].

Note: If the data is not normal (& N is large), the CLT can be used to approximate the sampling distribution by the asymptotic one:

$$s^2 \overset{a}{\rightarrow} \text{N}(\sigma^2, 2*\sigma^4/(\text{N}-1))$$

Sampling Distribution for the Sample Mean of a normal population:

$$\bar{X} \sim \text{N}(\text{m}, \sigma^2/n)$$



Note: As $n\rightarrow\infty$, $\bar{X}\rightarrow \mu$ –i.e., the distribution becomes a spike at μ!

Review – Estimating Moments in R

We estimate sample averages for e_f = **log returns for the CHF/USD**.

- First, we need to import the data. In R, we use the **read** function, usually followed by the type of data we are importing. Below, we import a comma separated values (csv) file with monthly CPIs and exchange rates for 20 different countries, then we use the **read.csv** function:

```
PPP_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/ppp_2020_m.csv",head=TRUE,sep=",")
```

To check the names of the variables we imported, we use the **names()** function. It describes the headers of the file imported (41 headers):

```
> names(PPP_da)
 [1] "Date"      "BG_CPI"    "IT_CPI"    "GER_CPI"   "UK_CPI"
 [6] "SWED_CPI"  "DEN_CPI"   "NOR_CPI"   "IND_CPI"   "JAP_CPI"
[11] "KOR_CPI"   "THAI_CPI"  "SING_CPI"  "MAL_CPI"   "KUW_CPI"
[16] "SUAD_CPI"  "CAN_CPI"   "MEX_CPI"   "US_CPI"    "EGY_CPI“
[...]
```

The **summary()** function provides some stats of variables imported:

```
> summary(PPP_da)
        Date        BG_CPI           IT_CPI          GER_CPI
 1/15/1971:  1   Min.   : 19.77   Min.   :  5.90   Min.   : 31.20
 1/15/1972:  1   1st Qu.: 49.32   1st Qu.: 32.25   1st Qu.: 57.17
 1/15/1973:  1   Median : 69.91   Median : 67.30   Median : 75.30
 1/15/1974:  1   Mean   : 67.92   Mean   : 60.14   Mean   : 72.29
 1/15/1975:  1   3rd Qu.: 89.40   3rd Qu.: 89.65   3rd Qu.: 91.17
 1/15/1976:  1   Max.   :109.71   Max.   :103.50   Max.   :106.60
 (Other)  :588
```

- Second, we extract a variable from the imported data, PPP_da, using the name of file followed by $ and the header of variable. That is, we extract from PPP_da, the column corresponding to the CHF/USD exchange rate:

```
x_chf <- PPP_da$CHF_USD
```

- Now, we define e_f = **log returns (≈ % changes) for the CHF/USD**.

```
T <- length(x_chf)                        # Size of series read (T or N notation is OK)
e_chf <- log(x_chf[-1]/x_chf[-T])         # Log returns
```

- Then, we estimate the sample moments for.

```
x <- e_chf                                # Series to be analyzed
n <- length(x)                            # Number of observations
m1 <- sum(x)/n                            # Mean (X̄)
m2 <- sum((x-m1)^2)/n                     # Used in denominator of both
m3 <- sum((x-m1)^3)/n                     # For numerator of S
```

```
m4 <- sum((x-m1)^4)/n                          # For numerator of K
b1 <- m3/m2^(3/2)                              # Sample Skewness (γ1)
b2 <- (m4/m2^2)                                # Sample Kurtosis (γ2)
s2 <- sum((x-m1)^2)/(n-1)                      # Sample Variance (s2)
sd_s <- sqrt(s2)                               # Sample SD (s)
```

- R output:

```
> m1
[1] -0.002550636
> s2
[1] 0.001115257
> sd_s
[1] 0.03339546
> b1                                           # Sample Skewness (γ1)
[1] -0.06733514
> b2                                           # Sample Kurtosis (γ2)
[1] 4.621602
```

Example: Summary of moments of e_f = % changes in the CHF/USD exchange rate (1971:Jan – 2020:Jun):

Statistic	e_f
Mean	**-0.002551**
Median	-0.001431
Maximum	0.145542
Minimum	-0.145639
Std. Dev.	**0.033395**
Skewness	-0.067335
Kurtosis	4.621602

Small mean (-**0.25%**), slight negative skewness, kurtosis greater than 3, pointing out to non-normality of data ("*fatter tails*"):

$$\Rightarrow \gamma_2 = \frac{\mu_4^0}{\sigma^4} - 3 = 1.62. \text{ ¶}$$

Returns

Returns have better statistical properties than prices. Models tend to focus on returns.

- Gross Return, R_t:

$$R_t = \frac{P_t + D_t}{P_{t-1}}$$

where P_t = Stock price or Value of investment at time t
D_t = Dividend or payout of investment at time t

• Net or simple return, r_t:

$$r_t = R_t - 1 = \frac{(P_t - P_{t-1}) + D_t}{P_{t-1}} = \text{ Capital gain + Dividend yield}$$

Note: This is the return from time t-1 to time t. To be very explicit we can write this as $r_{t-1,t}$.

• Log returns, r_t (or *continuously compounded* returns):

$$r_t = \log(R_t) = \log(P_t + D_t) - \log(P_{t-1}) \approx R_t - 1$$

Derivation:

Recall: $\ln(1) = 0$, $\quad \Rightarrow \frac{\delta \ln(x)}{\delta x} = \frac{1}{x}$.

Now do a 1st-order Taylor expansion around x_0 to get

$$\ln(x) \approx \ln(x_0) + \frac{\delta \ln(x)}{\delta x}|_{x_0}(x - x_0) \quad = \ln(x_0) + \frac{1}{x_0}(x - x_0)$$

Thus, expanding around $x_0 = 1$, we have for $x \approx 1$:

$$\ln(x) \approx 0 + \frac{1}{1}(x - 1) = x - 1$$

Set $x = R_t$ to get result.

When returns are small, say for daily or weekly data, the numerical differences between simple and compounded returns are very small.

Portfolio Returns

For portfolios, the simple rate of return has an advantage: The rate of return on a portfolio is the portfolio of the rates of return.

Let $V_{P,t}$ be the value of a portfolio at time t.

$$V_{P,t} = \sum_{i=1}^{N} N_i P_{i,t}$$

where N_i is the investment in asset i, which has a value $P_{i,t}$ at time t.

$$r_{P,t} = \frac{V_{P,t} - V_{P,t-1}}{V_{P,t-1}} = \frac{\sum_{i=1}^{N} N_i P_{i,t} - \sum_{i=1}^{N} N_i P_{i,t-1}}{\sum_{i=1}^{N} N_i P_{i,t-1}} = \sum_{i=1}^{N} w_i r_{i,t}$$

where w_i is the portfolio weight in asset i.

This relationship does not hold for log returns because the log of a sum is not the sum of the logs.

Multi-period holding return

To simplify notation, include dividends into prices. That is,

$$P_{d,t+1} = P_{t+1} + D_t$$

The two-period holding return is:

$$r_{t,t+2} = \frac{P_{d,t+2}}{P_{d,t}} - 1 = \frac{P_{d,t+2}}{P_{d,t+1}}\frac{P_{d,t+1}}{P_{d,t}} - 1 = (1 + r_{t+1,t+2})(1 + r_{t,t+1}) - 1$$

Or

$$1 + r_{t,t+2} = (1 + r_{t+1,t+2})\ (1 + r_{t,t+1})$$

For small returns, we can use the log approximation:

$$r_{t,t+2} \approx r_{t,t+1} + r_{t+1,t+2}$$

The k-period gross holding return $(1 + r_{t,t+k}) = \prod_{j=0}^{k-1}(1 + r_{t+j,t+j+1})$

Or with the log approximation: $r_{t,t+k} = \sum_{j=0}^{k-1} r_{t+j,t+j+1}$

If the (expected) returns are equal, that is, $r_{t+j,t+j+1} = r$. Then, the log approximation produces:

$$r_{t,t+k} = \sum_{j=0}^{k-1} r_{t+j,t+j+1} = k * r$$

If the returns, $r_{t+j,t+j+1}$, are independent (covariance is 0) and with a constant variance equal to σ_r^2(a constant), then under the log approximation

$$\text{Var}(r_{t,t+k}) = \sum_{j=0}^{k-1} var(r_{t+j,t+j+1}) = k * \sigma_r^2$$

Then, the SD is equal to

$$\text{SD}(r_{t,t+k}) = \sqrt{k * \sigma_r^2} = \sqrt{k} * \sigma_r$$

Real returns

We will deflate values by a Price Index, for example the CPI. Then,

$$\text{Real Price}_t = P_t^{real} = \frac{P_t}{CPI_t}$$

Then, the real return becomes:

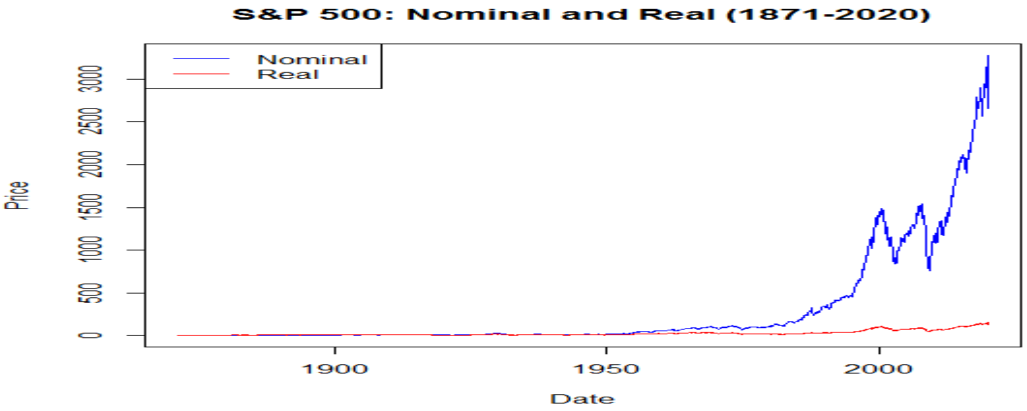
$$r_t^{real} = \frac{P_t^{real}}{P_{t-1}^{real}} - 1 = \frac{\frac{P_t}{CPI_t}}{\frac{P_{t-1}}{CPI_{t-1}}} - 1 = \frac{P_t}{P_{t-1}}\frac{CPI_{t-1}}{CPI_t} - 1 = \frac{(1+r_t)}{(1+\pi_t)} - 1$$

where π_t is the inflation rate at time t.

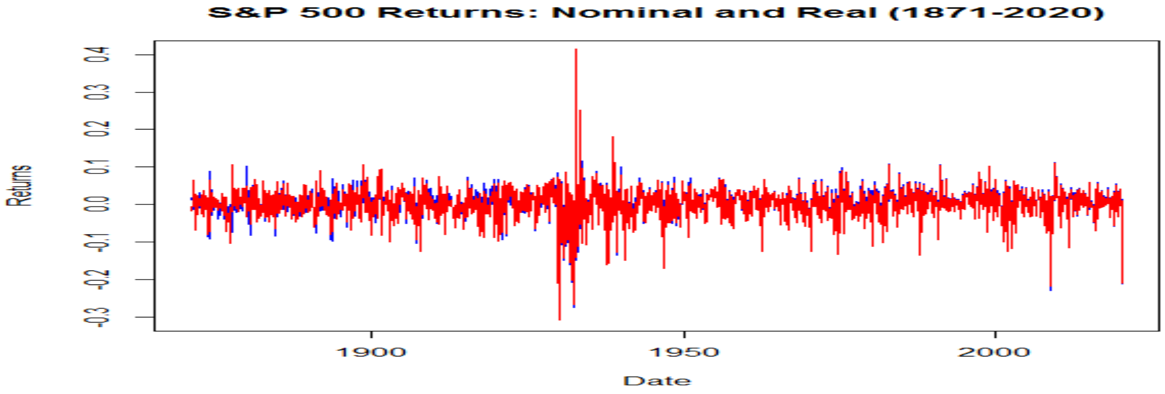
The log approximation (for small returns) produces

$$r_t^{real} \approx r_t - \pi_t$$

Example: Long-run S&P 500 monthly data, from Robert Shiller's website (1871:Jan -2020: Mar).



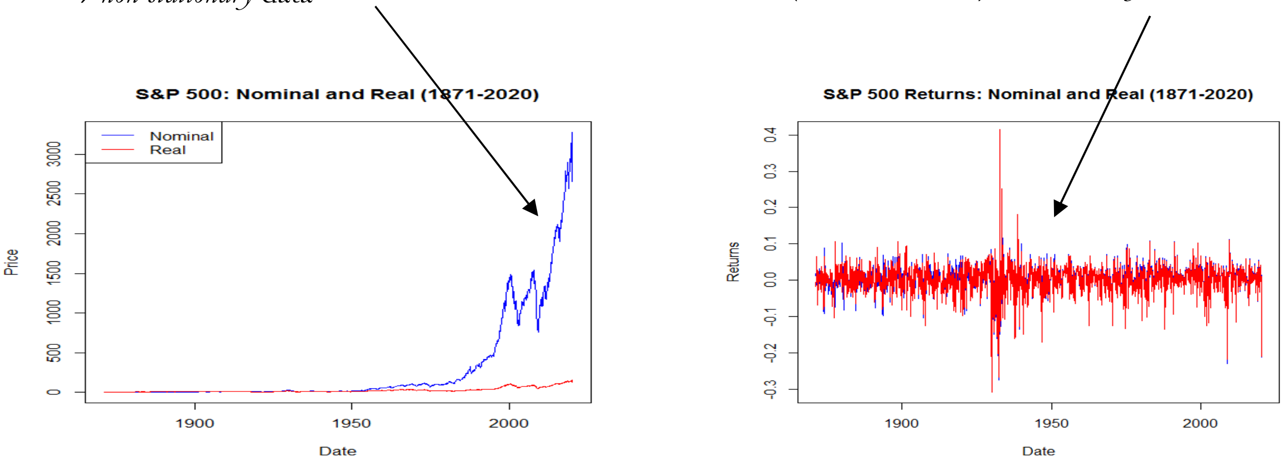
Long-run S&P 500 monthly log returns, from Robert Shiller's website (1871:Jan -2020: Mar; $N = 1790$).



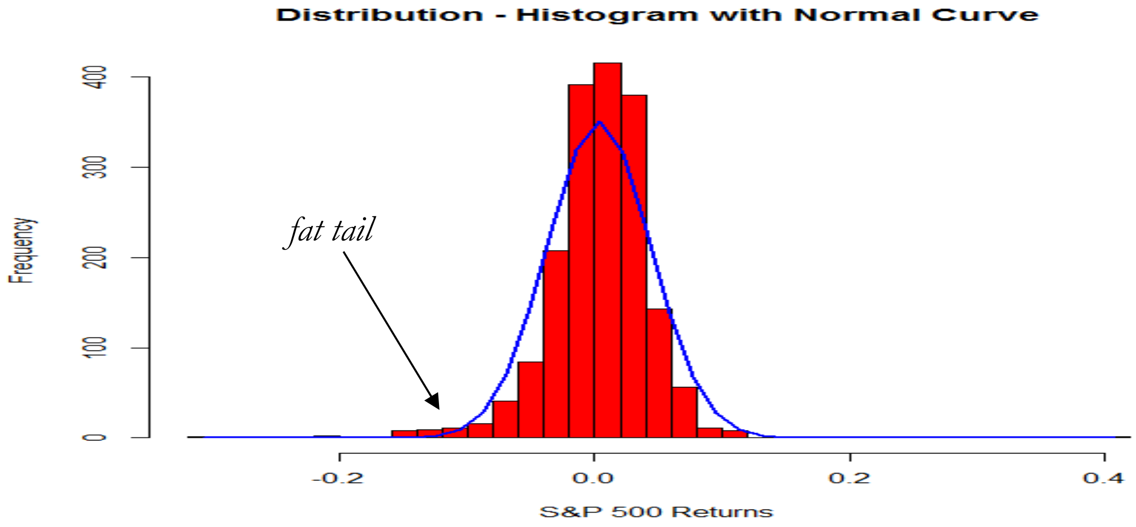
Prices have a clear trend, returns do not. In statistics, we prefer to work with data with no trends, like returns; they have better properties, for example, a well defined long-run mean (expected value).

mean changes with time (& variance too)
⇒ *non-stationary* data

mean seems constant over time
(& variance too) ⇒ *stationary* data



Distribution of **S&P 500 monthly log returns**



Example: Univariate stats for monthly **S&P 500 returns** and **U.S. inflation**

	Return	Inflation	Real Return
Mean	**0.003571**	**0.001693**	**0.001878**
Median	0.006489	0.001399	0.005546
Maximum	0.407459	0.067362	0.414844
Minimum	-0.307528	-0.067216	-0.307524
Std. Deviation (SD)	**0.040699**	0.010364	0.040936
Skewness	**-0.507281**	-0.142742	-0.399958
Kurtosis	**14.38916**	9.689002	14.13552
Jarque-Bera	**9751.2**	3343.1	9296
P-value (JB)	2.2e-16	2.2e-16	2.2e-16

Monthly returns are slightly (left-) skewed (& median > mean), and with "*fat tails*"–i.e., kurtosis is higher than 3.

• Check some results from log approximation:

(1) $r_t^{real} \approx r_t - \pi_t$ (for small % changes)

0.001878 ≈ **0.003571** − **0.001693** = **0.001878**

(2) Multi-period returns: k = **12** –i.e., we go from monthly returns to annual returns

- annual return = **0.003571** * **12** = **0.043852** (4.39%)
- annual SD = **0.040699** * sqrt(**12**) = 0.1409855 (14.10%)

Note: Compounding the monthly return: (1 + **0.003571**)^**12** = 0.043704. (very close to **0.043852**). ¶

Returns: Sample Moments – Changing Frequency

Assuming independence of returns and constant moments, we can use the log returns to easily change frequencies for the mean and variance of returns.

Suppose we have compounded data in base frequency *b (*say, monthly), but we are interested in compounded data in frequency *q* (say, annual). The approximation formulas for mean and standard deviation (SD) are:

q-frequency mean = b-freq mean * q/n

q-freq SD = b-freq SD * $\sqrt{(q/b)}$ $\Rightarrow$ q-freq Variance = sqrt(q-freq SD)

Example: Using the data from the previous table we calculate the weekly mean and standard deviation for returns (b=30, q=7).

- weekly return = **0.003571** * (7/30) = **0.0008332** (0.083%)
- weekly SD = **0.040699** * sqrt(7/30) = 0.019659 (1.97%)

Note: de-compounding the return: (1 + **0.003571**)^(7/30) = 0.000832. ¶

Returns: Sampling Distribution

Recall that the sampling distribution of the sample mean is:

$$\bar{X} \sim \text{N}(\mu, \sigma^2/N)$$

Example: We estimate the monthly S&P 500 mean return as **0.003571**, while the estimate of the variance of the monthly mean is

Estimated var($\bar{X}$) = **0.040699**^2/1790 = 9.253679e-07.

Taking the square root we get the SD of the monthly mean (also called the SE):

S.E.($\bar{X}$) = sqrt(9.253679e-07) = **0.0009619605** (or 0.1%).

Compared to returns, the sample mean is more precisely estimated (**0.1%** vs **4.07%**). Not surprised, the sampling distribution of the mean shrinks towards the sample mean as N increases. ¶

Yields

Consider an $n-$period discount bond. Time is measured in years. Today is t . Bond (asset) pays F_{t+n} dollars n years from now, at $t + n$.

F_{t+n} = Face value.

P_t = Market price of the bond.

$r_{n,t}$ = Yield to maturity (YTM)

n = Maturity of bond

$$\mathrm{P_t} = \frac{F_{t+n}}{(1+r_{n,t})^n}$$

Interpretation: P_t dollars invested at interest rate $r_{n,t}$ for n years compounded annually pays off F_{t+n}.

YTM, $r_{n,t}$, is a raw number. 4% at an annual rate is 0.04.

Continuous Compounding

n is years, m is number of times return (yield) is compounded, for example, $m = 4$ for quarterly, $m = 12$ for monthly, $m = \infty$ for continuous compounding. Then,

$$\mathrm{P_t} = \frac{F_{t+n}}{(1+\frac{r_t}{m})^{mn}}$$

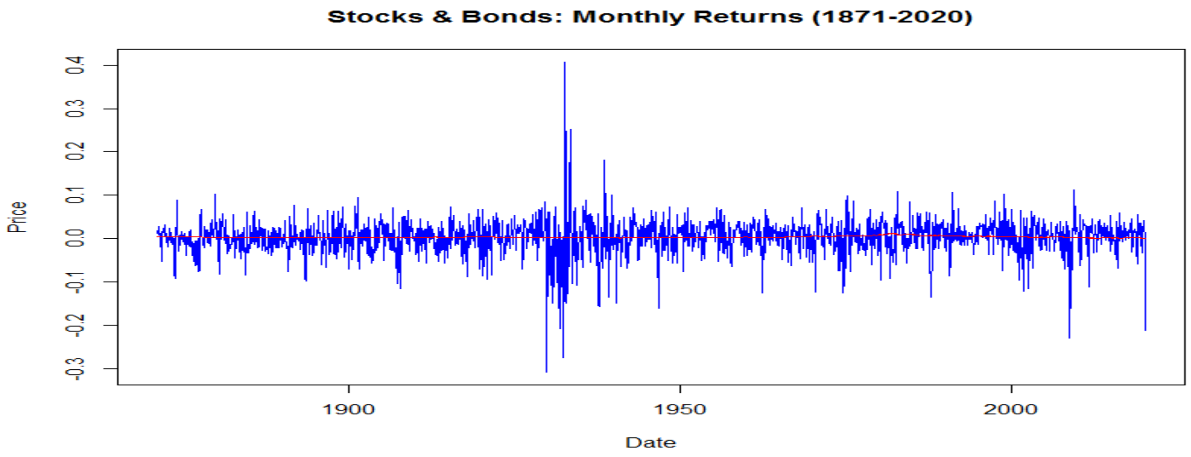
As $m \to \infty$, $\quad F_{t+n} = \mathrm{P_t}\,(1+\frac{r_t}{m})^{mn} \to \mathrm{P_t}\, e^{rn}$

where we used $\quad \lim_{m\to\infty}\left(1+\frac{x}{m}\right)^{mn} = e^{xn}$

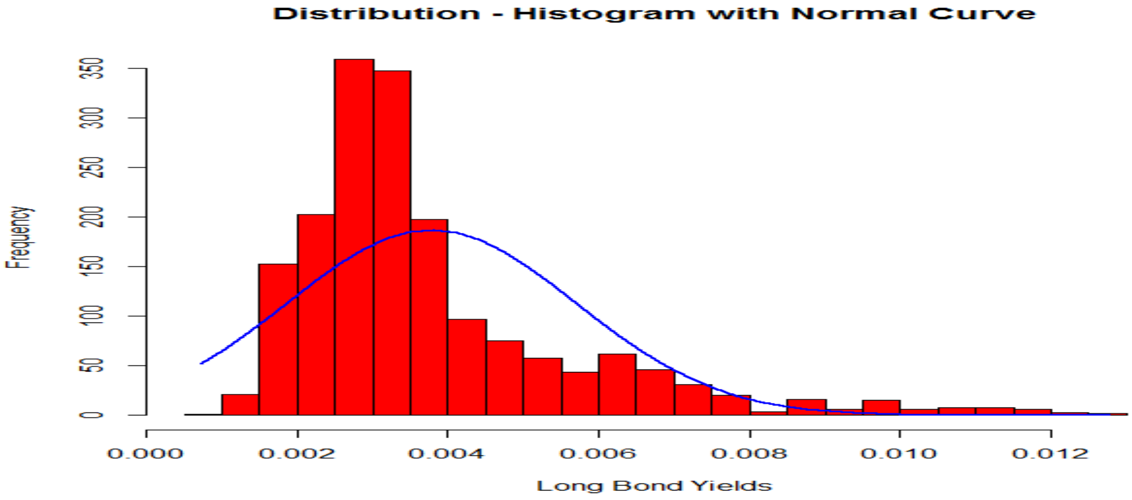
The *effective annual interest rate, r^a*, is simple the annual rate of return:

$$(1+r^a)^n = \left(1+\frac{r}{m}\right)^{mn} \qquad \Rightarrow \qquad r^a = \left(1+\frac{r}{m}\right)^m - 1$$

Example: We plot monthly **long-run S&P 500 (blue) returns** and **Bond (red) yields**, from Robert Shiller's website.



We plot the histogram for **bond yields**, with a normal curve (in blue) for comparison purposes



Review – Hypothesis Testing

A *statistical hypothesis test* is a method of making decisions using experimental data. A result is called *statistically significant* if it is unlikely to have occurred by chance.

These decisions are made using (null) hypothesis tests. A hypothesis can specify a particular value for a population parameter, say $q=q_0$. Then, the test can be used to answer a question like:

Assuming q_0 is true, what is the probability of observing a value for the (test) statistic used that is at least as big as the value that was actually observed?

Uses of hypothesis testing:

- Check the validity of theories or models.
- Check if new data can cast doubt on established facts.

Testing involves the comparison between two competing hypothesis (sometimes, they represent partitions of the world).

- The null hypothesis, denoted H_0, is sometimes referred to as the maintained hypothesis.
- The alternative hypothesis, denoted H_1, is the hypothesis that will be considered if the null hypothesis is "rejected."

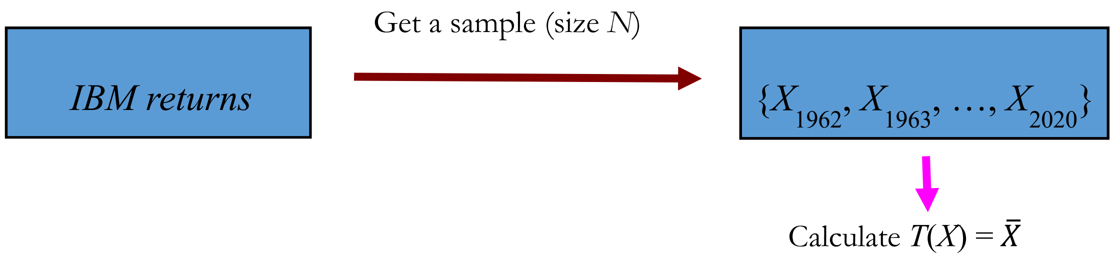
Idea: We collect a sample of data $X = \{X_1, \ldots, X_n\}$. We construct a statistic $T(X) = f(X)$, called the *test statistic*. Now we have a decision rule:

- If $T(X)$ is contained in space R, we reject H_0 (& we learn).
- If $T(X)$ is in the complement of R (R^C), we fail to reject H_0.

Note: $T(X)$, like any other statistic, is a RV. It has a distribution.

Example: Suppose we want to test if the mean of IBM annual returns, μ_{IBM}, is 10%. That is, H_0: $\mu_{IBM} = 10\%$.

From the population, we get a sample: $\{X_{1962}, X_{1963}, \ldots, X_{n=2020}\}$, with N=59. We use $T(X) = \bar{X}$, which is unbiased, consistent, and, assuming X is normally distributed, we know its distribution, $\bar{X} \sim N(\mu, \sigma^2/n)$.

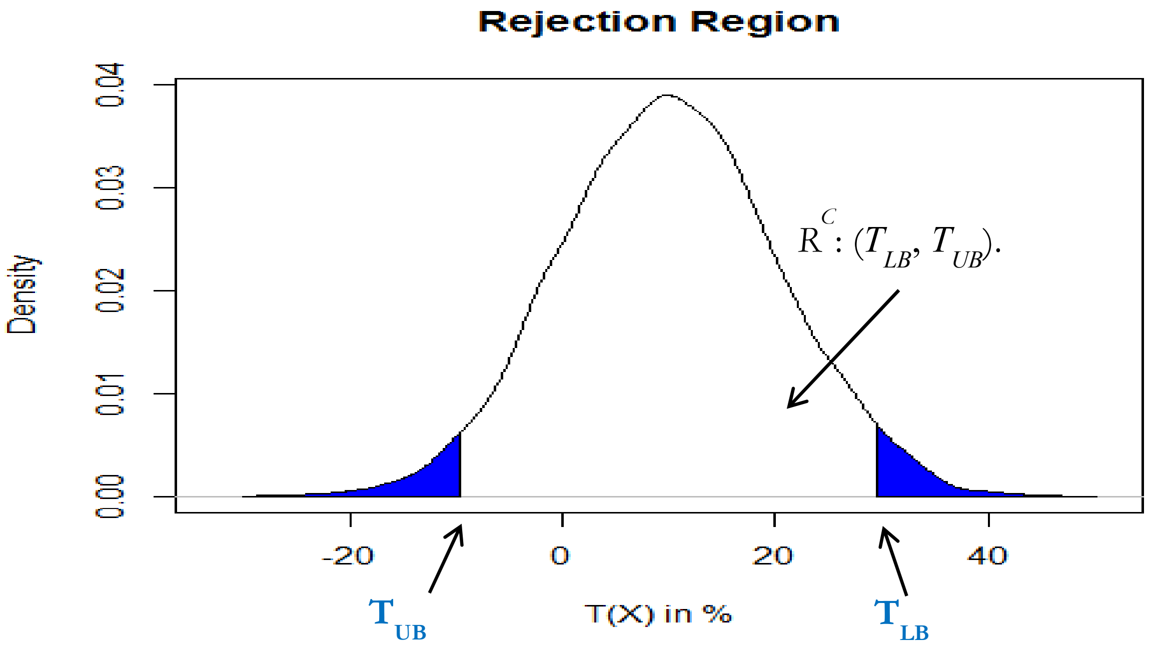

IBM returns
Get a sample (size N)
$\{X_{1962}, X_{1963}, \ldots, X_{2020}\}$
Calculate $T(X) = \bar{X}$

Now, we need to determine the rejection region, R, such that if

$$T(X) = \bar{X} \notin [T_{LB}, T_{UB}] \qquad \Rightarrow \text{Reject } H_0\text{: } \mu_{IBM} = 10\%.$$

That is,

$$R = [\bar{X} < T_{LB}, T_{UB} > \bar{X}]. \text{ ¶}$$



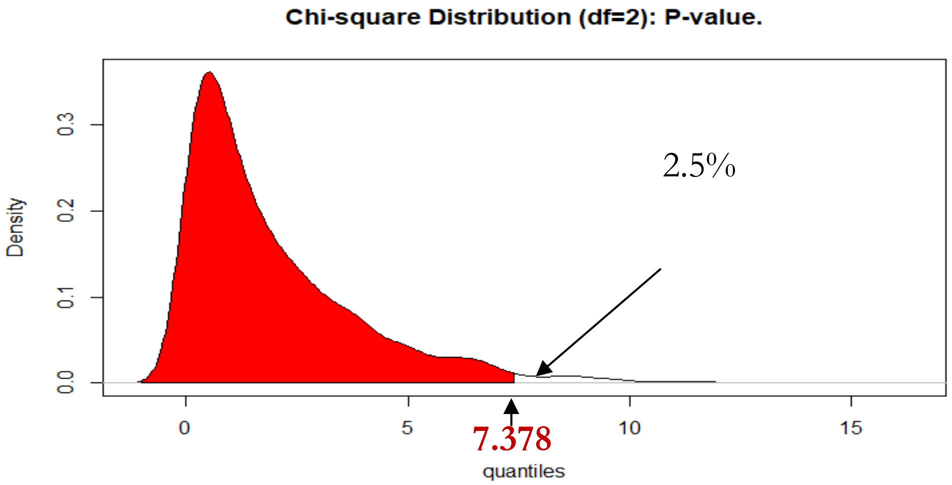
Q: How do we determine T_{LB} and T_{UB} and, thus, make a decision?

Review – Hypothesis Testing: p-value and steps

We present the *classical approach*, a synthesized approach, known as *significance testing*. It relies on Fisher's *p-value*:
p-value is the probability of observing a result at least as extreme as the test statistic, under H_0.

Example: Suppose $T(X) \sim \chi^2_2$. We compute $\widehat{T(X)} = \mathbf{7.378}$. Then,

$p\text{-}value(\widehat{T(X)} = \mathbf{7.378}) = 1 - \text{Prob}[T(X) < \mathbf{7.378}] = 0.025$



Note: In R, we calculate the *p-value* using pchisq(q ,df), which computes the CDF at value q of a Chi-square distribution with df degrees of freedom. Then,

```
> pchisq(q = 7.378, df = 2)                # Prob[T(X) < 7.378]
0.975003
p_val <- 1 - pchisq(q=7.378, df=2)         # p-value(T(X) = 7.378) = 1 - Prob[T(X) < 7.378]
> p_val
[1] 0.02499699. ¶
```

Using the distribution of the test statistic T(X) under the null hypothesis, Fisher's *significance testing* approach determines a rejection region, based on the significance level (α%).

We follow these steps:
1. Identify H_0 & decide on a *significance level* (α%) to compare your test results.
2. Determine the appropriate test statistic T(X) and its distribution under the assumption that H_0 is true.
3. Calculate T(X) from the data.
4. Rule: Reject H_0 if the *p-value* is sufficiently small, that is, we consider T(X) in R (we learn). Otherwise, we reach no conclusion (no learning).
Note: In Step 4, setting α% is equivalent to setting R.

• Q: What *p-value* is "sufficiently small" as to warrant rejection of H_0?
Rule: If *p-value* < α (say, 5%) ⟹ test result is *significant:* Reject H_0.

If the results are "*not significant*," no conclusions are reached (no learning here). Go back gather more data or modify model.
The father of this approach, Ronald Fisher, favored 5% or 1%.

Example: From the U.S. Jury System
H_0: The defendant is not guilty.
H_1: The defendant is guilty. ¶
In statistics, we learn when we reject. In this case, we learn a defendant is guilty when the jury finds the defendant guilty, by rejecting H_0.

Example: From the U.S. Jury System
1. Identify H_0 & decide on a *significance level* (α%)
H_0: The defendant is not guilty
H_1: The defendant is guilty
Significance level α = "*beyond reasonable doubt*," presumably small level.

2. After judge instructions, each juror forms an "innocent index" $\text{T}(X)_i$.

3. Through deliberations, jury reaches a conclusion $\text{T}(X) = \sum_{i=1}^{12} \text{T}(X)_i$.

4. Rule: If *p-value* of $\text{T}(X) < \alpha$ ⇒ Reject H_0. That is, guilty!
If *p-value* of $\text{T}(X) > \alpha$ ⇒ Fail to reject H_0. That is, non-guilty.
Alternatively, we build a rejection region around H_0. ¶

Note: Mistakes are made. We want to quantify these mistakes.

Failure to reject H_0 does not necessarily mean that the defendant is not guilty, or rejecting H_0 does not mean necessarily the defendant is guilty. *Type I error* and *Type II error* give us an idea of both mistakes.

Definition: Type I and Type II errors
A *Type I error* is the error of rejecting H_0 when it is true. A *Type II error* is the error of "accepting" H_0 when it is false (that is, when H_1 is true).

Notation: Probability of Type I error: $\alpha = \text{P}[\text{X} \in R|\text{H}_0]$
Probability of Type II error: $\beta = \text{P}[\text{X} \in R^C|\text{H}_1]$
We call 1 – β the power of the test –i.e., the probability of rejecting a false null hypothesis.

Example: From the U.S. Jury System
Type I error is the error of finding an innocent defendant guilty.
Type II error is the error of finding a guilty defendant not guilty.

	State of World

Decision	H_0 true ("not guilty")	H_1 true ("guilty")
Cannot reject H_0	*Correct decision*	Type II error
Reject H_0	Type I error	*Correct decision*

Note: We usually think that we learn when we reject H_0. Note that some "learning" comes from Type I error –i.e., from *false positives*. ¶

In general, we think *Type I error* is the worst of the two errors: We try to minimize the error of sending to jail an innocent person.

Actually, we would like *Type I error* to be zero. However, the only way to do this (100% of innocent defendants are found not guilty) is to never reject H_0. Then, we maximize *Type II error*.

There is a clear trade-off between both errors. Traditional view: Set *Type I error* equal to a small number (defined in the U.S. court system as "*beyond reasonable doubt*") and design a test that minimizes *Type II error*.

The usual tests (*t-tests*, *F-tests*, Likelihood Ratio tests) incorporate this traditional view.

Example: We want to test if the mean is equal to μ_0. Then,

1. H_0: $\mu = \mu_0$.
H_1: $\mu \neq \mu_0$.

2. Appropriate T(X): *t-test* (based on σ unknown and estimated by s).
Determine distribution of T(X) under H_0. Sampling distribution of $\bar{X}$, under H_0:

$$\bar{X} \sim N(\mu_0, \sigma^2/n).$$

Then, distribution of T(X) under H_0:

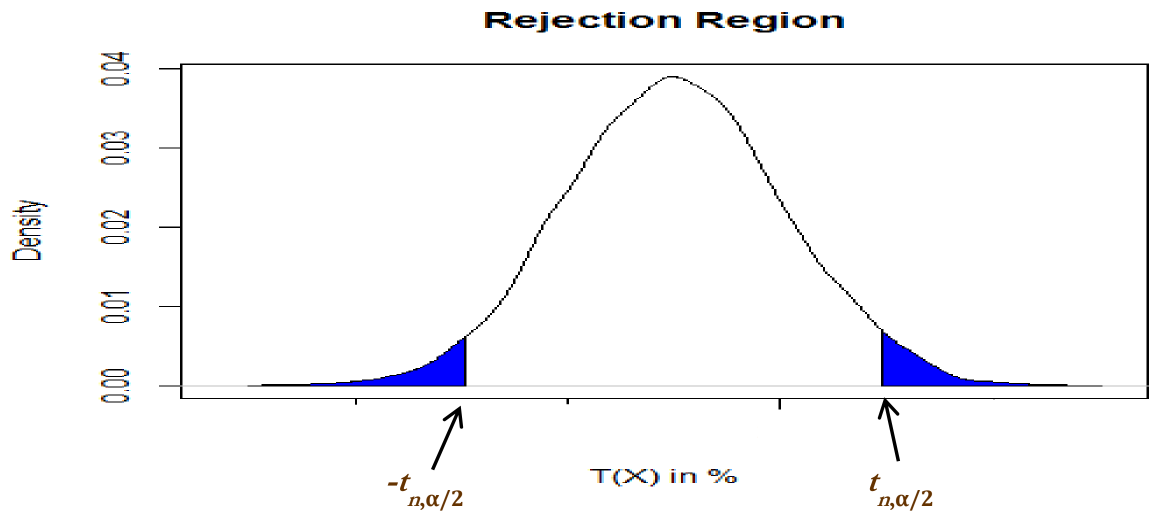
$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \sim t_{n-1} \qquad \text{–when } n > 30,\ t_n \sim N(0, 1).$$

3. Compute t, $\hat{t}$, using $\bar{X}$, μ_0, s, and N. Get *p-value*($\hat{t}$).

4. Rule: Set α level. If *p-value*($\hat{t}$) < α $\Rightarrow$ Reject H_0: $\mu = \mu_0$.
Alternatively, if $|\hat{t}| > t_{n-1,\alpha/2}$ (=**1.96**, if α=.05) $\Rightarrow$ Reject H_0: $\mu = \mu_0$. ¶

Notice the alternative Rule, where we set a Rejection region:

$$R = [|\hat{t}| > t_{n,\alpha/2}]$$



If α = 5% and $n > 30$, then $t_{n>30,\ .025} = \mathbf{1.96}$ (≈ **2**).

Technical Note 1: In step 2, the distribution of the t-test, t, is exact if {X} follows a normal distribution, otherwise, the distribution is asymptotic (for this we need a large *n*); that is

$$t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}} \overset{d}{\rightarrow} N(0, 1).$$

Technical Note 2: In step 2, we determine the distribution of t, by using the sampling distribution of $\bar{X}$ under H$_0$. If H$_0$ is not true, say $\mu = \mu_1$, then

$$\bar{X} \sim N(\mu_1, \sigma^2/n),$$

thus, t is distributed N(0, 1) only under H$_0$, since only under H$_0$ the E$[\bar{X} - \mu_0] = 0$.

Review – Hypothesis Testing: Examples

Example 1: We want to test if the monthly return of the S&P 500 is equal to zero using α = .05. We use the **S&P 500 returns** (1871-2020) with the following mean and variance: $\bar{X}$ = **0.003571**, s = **0.04070**, N = 1790.

1. H$_0$: $\mu = 0$.

H$_1$: $\mu \neq 0$.

2. $t = \frac{\bar{X} - \mu_0}{s/\sqrt{n}}$

3. $\hat{t} = \frac{0.003571}{0.04070/\sqrt{1790}} = \mathbf{3.7121}$ & *p-value*($\hat{t}$) = **0.0001**

4. Rule: *p-value*($\hat{t}$) = **0.0001** < α = .05 ⇒ Reject H$_0$: $\mu = 0$.
Alternatively, $|\hat{t}$ = **3.7121**$| > t_{1789,.025}$ = **1.96** ⇒ Reject H$_0$: $\mu = 0$.

Conclusion: S&P 500 monthly mean returns are not equal to zero.

Note: In R, we find the $t_{1789,.025}$ using qt (p, df) which gives the quantile of the t-distribution with df degrees of freedom. That is,

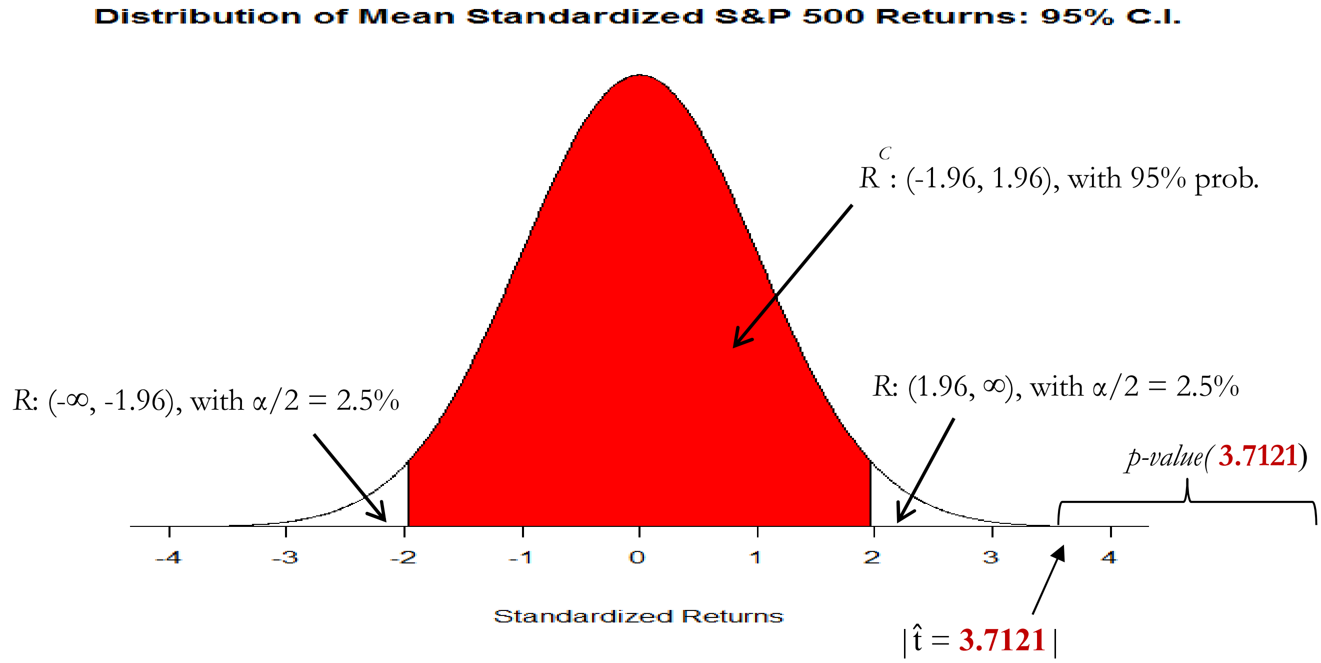
```
> qt(.975, 1789)                                # = (-1)*qt(.025, 1789) by symmetry.
[1] 1.961291

# Check result with pt (q, df)
> pt (q=1.96, df=1789)
[1] 0.9749246. ¶
```

Q: How do we calculate the *p-value*? Recall, it is the probability of observing a result at least as extreme as the test statistic, under H_0.

In this case, we know that under H$_0$: $\mu = 0$, the t-stat is well approximated by a N(0,1) distribution (since N>30). Then, we use the R function *pnorm* to calculate the cumulative standard normal value up to **3.7121**, and then subtract it from 1:

```
p_val <- 1 - pnorm(3.7121)                      # p-value of t_test
> p_val
[1] 0.0001027734
```

The observed t ($\hat{t}$ = **3.7121**) is outside the non-rejection region (R^C) built around H$_0$: (-1.96, 1.96). ¶



Example 2: We want to test if **monthly S&P 500 returns** (1871-2020) have a normal distribution using α = .05. If the distribution is normal, skewness is zero and kurtosis is equal to

3 (or excess kurtosis equals 0). The estimated moments are: $\hat{\gamma}_1 = \mathbf{-0.5072814}$, $\hat{\gamma}_2 = (14.38916 - 3) = 11.38916$, & $N = 1790$.

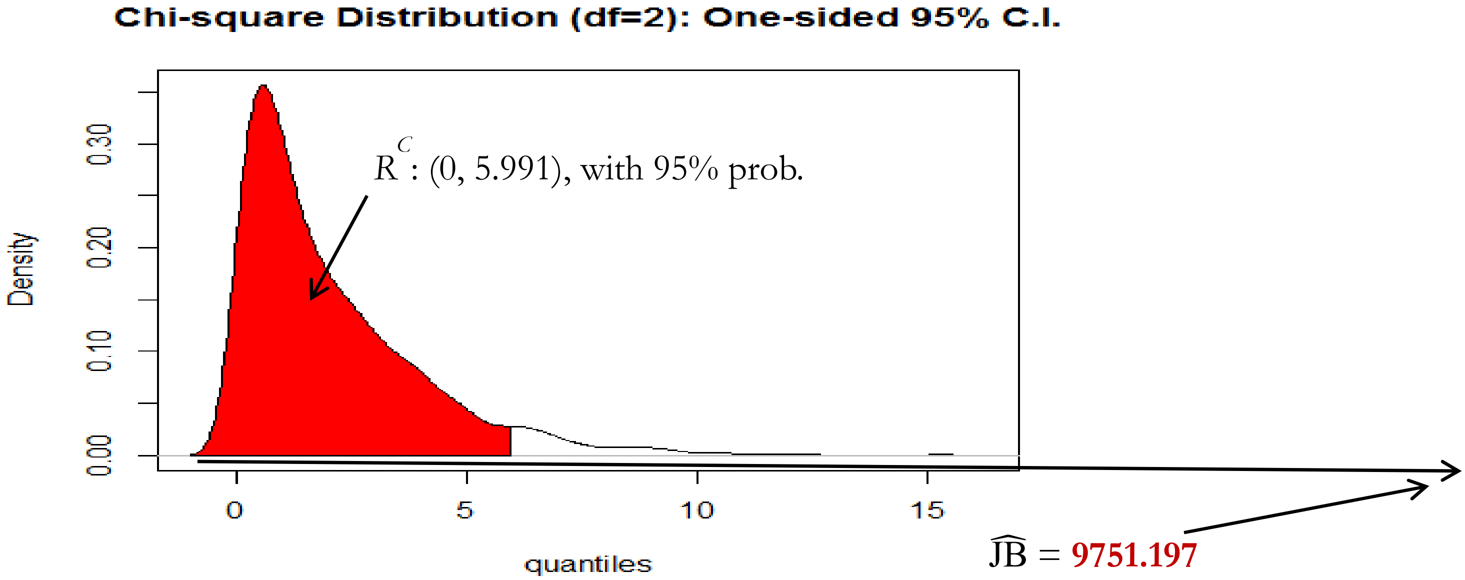
1. H₀ (Data is normal): $\gamma_1 = \frac{\mu_3^0}{\sigma^3} = 0$ and $\gamma_2 = \frac{\mu_4^0}{\sigma^4} - 3 = 0$.
H₁ (Data is not normal): $\gamma_1 \neq 0$ and/or $\gamma_2 \neq 0$.

2. Appropriate T(X): the *Jarque-Bera test* (JB), $\quad \text{JB} = \frac{N}{6} * (\gamma_1^2 + \frac{\gamma_2{}^2}{4})$
Under H₀, $\text{JB} \xrightarrow{d} \chi_2^2$ (chi-square distribution with 2 *degrees of freedom*)

3. $\widehat{\text{JB}} = \frac{1790}{6} * [(\mathbf{-0.5072814})^2 + \frac{(11.38916)^2}{4}] = \mathbf{9751.197}$

4. Rule: *p-value*($\widehat{\text{JB}} = \mathbf{9751.197}$) ≈ 0 < α = .05 ⇒ Reject H₀.
Alternatively, compare $\widehat{\text{JB}}$ to the $\chi^2_{2,.95}$ value ($\chi^2_{2,.95} = \mathbf{5.991}$). That is,

$\widehat{\text{JB}} > \chi^2_{2,.95}$ ⇒ Reject H₀. (A strong rejection!)



R^C: (0, 5.991), with 95% prob.

$\widehat{\text{JB}} = \mathbf{9751.197}$

Conclusion: Monthly S&P 500 returns are not normally distributed. ¶

Review – Confidence Intervals (C.I.)

When we estimate parameters with an estimator, $\hat{\theta}$, we get a a point estimate for θ, meaning that $\hat{\theta}$ is a single value in R^k. For example, in the previous example, we get $\bar{X} = 0.003571$.

Broader concept: Estimate a set C_n, a collection of values in R^k. For example, for $\bar{X}$ = {0.00155, 0.00554}.

It is common to focus on intervals $C_n = [L_n; U_n]$, called an *interval estimate* for θ. The goal of C_n is to contain the true population value, θ. We want to see $\theta \in C_n$, with high probability.

Technical detail: Since C_n is a function of the data, it is a RV and, thus, it has a pdf associated with it. The *coverage probability* of the interval $C_n= [L_n; U_n]$ is $\text{Prob}[\theta \in C_n]$.

Intervals estimates C_n provide an idea of the uncertainty in the estimation of θ: The wider the interval C_n, the more uncertain we are about our estimate, $\hat{\theta}$.

Interval estimates C_n are called *confidence intervals* (C.I.) as the goal is to set the coverage probability to equal a pre-specified target, usually 90% or 95%. C_n is called a $(1-\alpha)\%$ C.I.

When we know the distribution for the point estimate, it is straightforward to construct a C.I. For example, if the distribution of $\hat{\theta}$ is normal, then a $(1-\alpha)\%$ C.I. is given by:

$$C_n= [\hat{\theta} - z_{\alpha/2} * \text{Estimated SE}(\hat{\theta}),\ \hat{\theta} + z_{\alpha/2} * \text{Estimated SE}(\hat{\theta})]$$

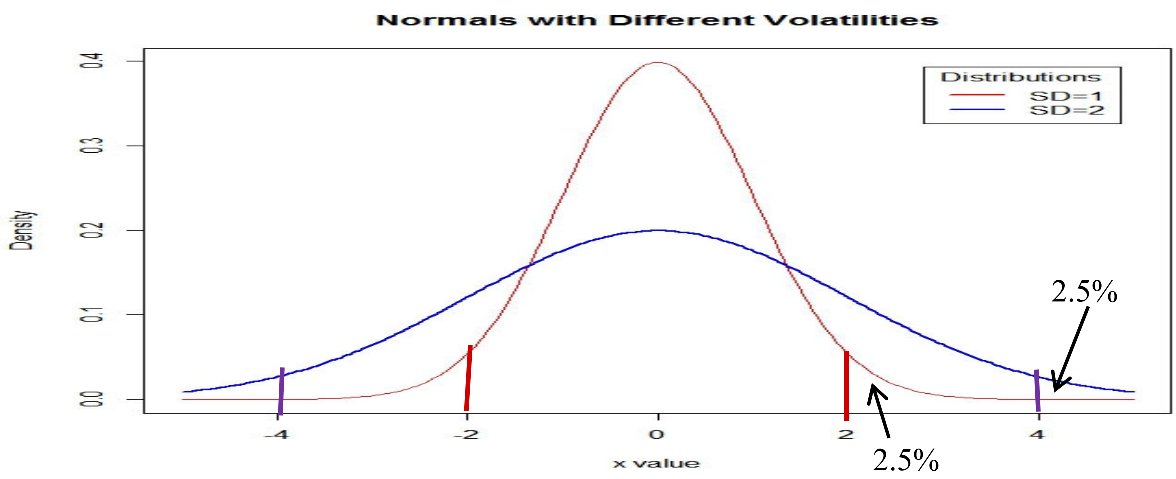
This C.I. is symmetric around $\hat{\theta}$. Its length is proportional to $\text{SE}(\hat{\theta})$.

If the data follows a Normal distribution, then for the sample mean a $(1-\alpha)\%$ C.I. is given by:

$$C_n= [\bar{X} - z_{\alpha/2} * \text{SD}(\bar{X}),\ \bar{X} - z_{\alpha/2} * \text{SD}(\bar{X})]$$

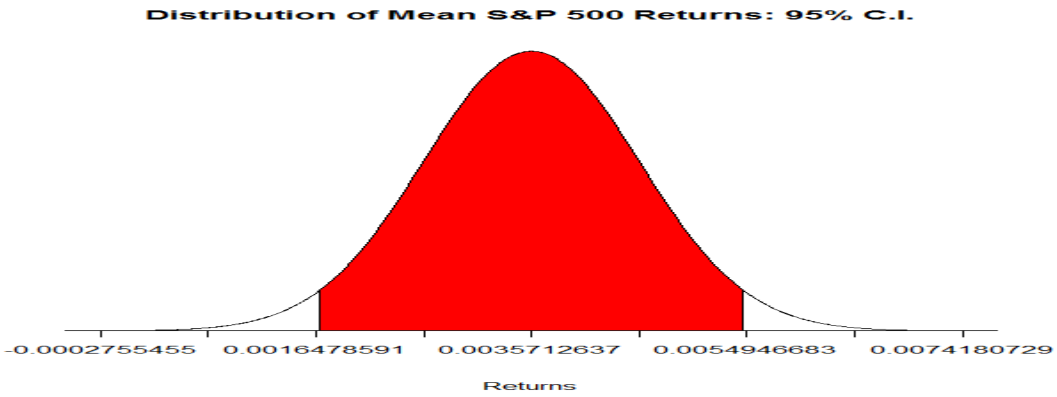
The size of the symmetric C.I. depends on the SD (=SE). The higher SD, the wider the C.I.

Example: Two 95% C.I. for the mean, with two different SD (=1, 2), are plotted below.



Then,

$$C_n= [0.003571 - 1.96 * (0.04070/\sqrt{1790}),\ 0.003571 + 1.96 * (0.04070/\sqrt{1790}\ )]$$
$$= [0.001686, 0.005456] = [0.17\%, 0.55\%].$$



By looking at the 95% C.I., we can reject than monthly S&P 500 returns are 0, since 0% is outside the 95% C.I. But, the C.I is wide, even after 130 years of data.
Conclusion: Reject H_0: $\mu = 0$, since 0 is outside the observed 95% C.I. ¶

Example: We want to estimate a 95% C.I. for the variance of monthly mean return of the S&P 500. Assuming normality, the sample variance is distributed: $(N\text{-}1)\, s^2/\sigma^2 \sim \chi^2_{N-1}$.
To derive a (1 - α)% C.I. for the variance, we rewrite the standard confidence interval for a chi-squared variable:

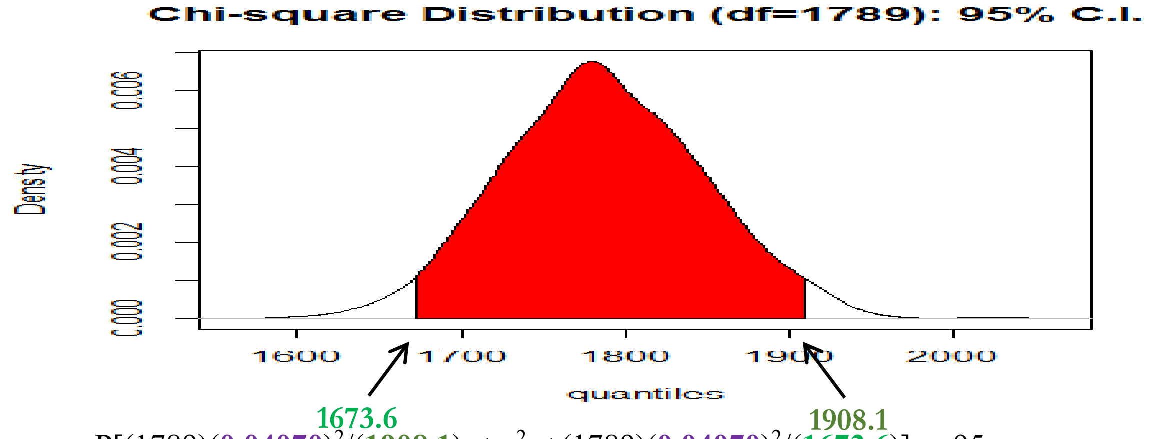
$$P(\chi^2_{\upsilon,\alpha/2} < \chi^2_{\nu} < \chi^2_{\upsilon,1-\alpha/2}) = P(\chi^2_{\upsilon,\alpha/2} < (N\text{-}1)\, s^2/\sigma^2 < \chi^2_{\upsilon,1-\alpha/2}) = 1 - \alpha$$

After some easy algebra, we derive:

$$P[(N - 1)\, s^2/\chi^2_{\upsilon,1-\alpha/2} < \sigma^2 < (N - 1)\, s^2/\chi^2_{\upsilon,\alpha/2}] = 1 - \alpha.$$

Note: This C.I. is not symmetric. But, as the degrees of freedom get large, the χ^2_{N-1} starts to look like the normal distribution and, thus, CIs will look more symmetric.

From the χ^2_{1789} distribution, we get: $\chi^2_{1789,.025}$ = **1673.6** & $\chi^2_{1789,.975}$ = **1908.1**.



$$P[(1789)(\mathbf{0.04070})^2/(\mathbf{1908.1}) < \sigma^2 < (1789)(\mathbf{0.04070})^2/(\mathbf{1673.6})] = .95$$

$$P[0.001553 < \sigma^2 < 0.00177071] = .95$$

Taking square root above delivers a 95% C.I. for σ:

⇒ 95% C.I. for σ is given by (**3.941%**, **4.208%**).

The C.I. is quite compact around the sample point estimate. Compared to the sample mean estimate, σ is measured with accuracy.

Note: Usually N is large (N>30). We can use the normal approximation to calculate CIs for the population σ. For the S&P data, we estimate the S.E. for the sample SD:

$$\text{SE}(s) = s/\sqrt{2 * (N-1)} = \mathbf{0.04070}/\text{sqrt}(2*1789) = 0.00068 \text{ (or .068\%)}.$$

A 95% CI for σ is given by (**3.937%**, **4.203%**). (Good approximation!) ¶

C.I. Application: Using the ED – The Bootstrap

In the previous examples, we assumed that we knew the distribution of the data: Stock returns follow a normal distribution. What happens when the data follows an unknown distribution, *F*?

We still can use the sample mean, $\bar{X}$, or the sample variance, s^2, as estimates of μ and σ^2, since the LLN tell us that they are both consistent estimators. If we have a "large" dataset –i.e., large *N*– we can use the CLT to justify a C.I based on a normal distribution.

But, when we have an unknown distribution *F* and we do not have a large enough *N* or we suspect the normal approximation is not a good one, we still can build a C.I. for any statistic using a new method: a *bootstrap*.

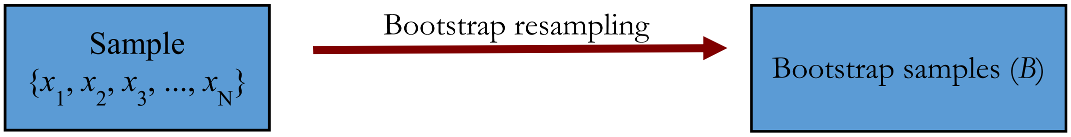
Bootstrapping is the practice of estimating the properties of an estimator -say, its variance- by measuring those properties when sampling from an approximating distribution (the *bootstrap DGP*).

That is, it is necessary to estimate a bootstrap DGP from which to draw the simulated samples. The DGP that generated the original data is unknown, and so it cannot be used to generate simulated data.

⇒ The bootstrap DGP estimates the unknown true DGP.

Idea: We use the data at hand -the empirical distribution (ED)- to estimate the variation of statistics that are themselves computed from the same data. Recall that, for large samples, the ED approximates the CDF very well.

The *empirical bootstrap* is a statistical technique, easy to implement, that takes advantage of today's modern computers, by resampling from the ED. Bootstrapping uses the ED –i.e., sample- as if it were the true CDF.



• Suppose we have a sample with *N* observations drawn from *F(x)*:

$$\{x_1, x_2, x_3, \ldots, x_N\}$$

From the ED, *F**, we sample ("*resample*") with replacement *N* observations:

$$\{x_1^*=x_2,\ x_2^*= x_4,\ x_3^* = x_4, .., x_N^*= x_{N-8}\}$$

This is an *empirical bootstrap sample*, which is a resample of the same size N as the original data, drawn from F^*.

For any statistic θ computed from the original sample data, we can define a statistic θ^* by the same formula, but computed instead using the resampled data. Then,

$$\{x_1^*=x_2,\ x_2^*=x_4,\ x_3^*=x_4,\ ..,\ x_N^*=x_{N-8}\} \quad \Rightarrow \hat{\theta}_1^*$$

θ^* is computed by resampling the original data; we can compute many θ^* by resampling many times from F^*. Say, we resample θ^* B times.

$$\{x_1^*=x_2,\ x_2^*=x_4,\ x_3^*=x_4,\ \ldots,\ x_N^*=x_{N-8}\} \Rightarrow \hat{\theta}_1^*$$
$$\{x_1^*=x_3,\ x_2^*=x_3,\ x_3^*=x_3,\ \ldots,\ x_N^*=x_{N-8}\} \quad \Rightarrow \hat{\theta}_2^*$$
$$\vdots$$
$$\{x_1^*=x_1,\ x_2^*=x_1,\ x_3^*=x_5,\ \ldots,\ x_N^*=x_N\} \quad \Rightarrow \hat{\theta}_B^*$$

C.I. Application: The Empirical Bootstrap

We have a collection of estimated θ^*:

$$\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, \ldots, \hat{\theta}_B^*\}.$$

From this collection of $\hat{\theta}^*$'s, we can compute the mean, the variance, skewness, draw a histogram, etc., and confidence intervals. From this collection of $\hat{\theta}^*$'s, we learn about the behavior of statistic θ.

• Bootstrap Steps:
1. From the original sample, draw random sample with size N**.**
2. Compute statistic θ from the resample in 1: $\hat{\theta}_1^*$.
3. Repeat steps 1 & 2 B times ⇒ Get B statistics: $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, \ldots, \hat{\theta}_B^*\}$
4. Compute moments; draw histograms; etc. for these B statistics.

With a large enough B, the LLN allows us to use the $\hat{\theta}^*$'s to estimate the distribution of θ, $F(\theta)$.

• Results (bootstrap principle):
1. With a large enough B, the LLN allows us to use the $\hat{\theta}^*$'s to estimate the distribution of $\hat{\theta}$, $F(\hat{\theta})$.
2. The variation in $\hat{\theta}$ is well approximated by the variation in $\hat{\theta}^*$.

Result **2** is the one that we use to estimate the size of a C.I.

To build a C.I. for θ, we use $\hat{\theta}$, computed from the original sample. As in the previous C.I.'s, we want to know how far is $\hat{\theta}$ from θ. For this, we would like to know the distribution of

$$q = \hat{\theta} - \theta.$$

If we knew the distribution of $q = \hat{\theta} - \theta$, we build a $(1 - \alpha)\%$ C.I., by finding the critical values $q_{\alpha/2}$ & $q_{(1-\alpha/2)}$ to have:

$$\Pr\,(q_{\alpha/2} \leq \hat{\theta} - \theta \leq q_{(1-\alpha/2)}\,|\theta) = 1 - \alpha$$

Or, after some manipulations:

$$\Pr\,(\hat{\theta} - q_{\alpha/2} \geq \theta \geq \hat{\theta} - q_{(1-\alpha/2)}\ |\theta) = 1 - \alpha,$$

which gives a (1 – α)% C.I.:

$$C_n = [\hat{\theta} - q_{(1-\alpha/2)}, \hat{\theta} - q_{\alpha/2}]$$

We do not know the distribution of q, but we can use the bootstrap to estimate it with

$$q^* = \hat{\theta}^* - \hat{\theta}.$$

and then to get $q^*_{\alpha/2}$ & $q^*_{(1-\alpha)/2}$:

$$C_n = [\hat{\theta} - q^*_{(1-\alpha)/2}, \hat{\theta} - q^*_{\alpha/2}]$$

This C.I. is called the *pivotal* C.I.

Intuition: The distribution of $\hat{\theta}$ is 'centered' at θ, while the distribution of θ^* is centered at $\hat{\theta}$. If there is a significant separation between $\hat{\theta}$ and θ, these two distributions will also differ significantly.

On the other hand, the distribution of $q = \hat{\theta} - \theta$ describes the variation of $\hat{\theta}$ about its center. Similarly, the distribution of $q^* = \theta^* - \hat{\theta}$ describes the variation of θ^* about its center.

Then, even if the centers are quite different, the two variations about the centers can be approximately equal.

Example: We want to estimate a 95% C.I. for the variance of monthly returns of the S&P 500. (You need to install R package ***resample***, using the **install.packages()** function.)

```
Sh_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Shiller_2020data.csv", head=TRUE,
sep=",")
SP <- Sh_da$P
T <- length(SP)
lr <- log(SP[-1]/SP[-T])
lr_var <- var(lr)
T_s <- length (lr)
sim_size <- 1000                                             # B = size of bootstrap

library(resample)                                            # call library resample
data_star <- sample(lr,T_s*sim_size, replace=TRUE)           # create B resamples of size T_s
boot_sample <- matrix(data_star, nrow=T_s, ncol=sim_size)  # organize resamples in matrix

boots_vars <- colVars(boot_sample)                    # variance for each bootstrap sample
q_star <- boots_vars - lr_var                         # Compute q* for each bootstrap sample
q <- quantile(q_star, c(0.025, 0.975))                # Find the 0.025 & 0.975 quantile for q*
ci <- lr_var -c(q[2], q[1])                           # Calculate the 95% C.I. for the variance.
cat("Confidence interval: ",ci, "\n")                 # Print C.I using cat
```

```
> lr_var
[1] 0.001656445
> ci
      97.5%        2.5%
0.001376664 0.001909769
```

```
> cat("Confidence interval: ",ci, "\n")
Confidence interval:  0.001376664 0.001909769
>
```

Or for σ, the 95% CI is given by (**3.71%**, **4.37%**). ¶

Example: Now, we construct the same 95% C.I. for the variance of monthly S&P 500 returns but using the R package ***boot***. You need to install package first, using the **install.packages()** function.

```
library(boot)
# function to obtain the variance from the data
var_p <- function(data, i) {
          d <-data[i]
return(var(d))
}

boot.samps <- boot(data=lr, statistic=var_p,  R=sim_size)  # resampling and θ* estimation
boot.ci(boot.samps, type = "basic")                        #  boot  computes  the  CI  with
type=basic.

> boot.ci(boot.samps, type = "basic")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.samps, type = "basic")

Intervals :
Level      Basic
95%   ( 0.0014,  0.0019 )
Calculations and Intervals on Original Scale
```

- Check results using previous step-by-step process:

```
q_star <- boot.samps$t - lr_var                            # q* = θ* − θ̂
q_ad <- sort(q_star)                                       # sort q*
> lr_var - q_ad[975]                                       # CI's Lower Bound
[1] 0.001357793
> lr_var - q_ad[25]                                        # CI's Upper Bound
[1] 0.001914674
```

We can transform this CI for the variance into a CI for the SD:

```
> sqrt(lr_var - q_ad[975])
[1] 0.03684825
> sqrt(lr_var - q_ad[25])
[1] 0.04375699
```

A 95% CI for σ is given by (**3.68%**, **4.38%**), wider than the CI assuming a Normal distribution for returns. ¶

Note that we can also gauge the uncertainty of the estimation of θ by computing the sample standard error, SE($\hat{\theta}$*). (Recall we call the standard deviation of an estimator its standard error.):

- Steps

1. Computing the sample variance:

$$\text{Var}(\hat{\theta}*) = \frac{1}{B-1}\sum_{i=i}^{B}(\hat{\theta}_i^* - \bar{\theta}*)^2,$$

where $\bar{\theta}* = \frac{1}{B}\sum_{i=i}^{B}\hat{\theta}_i^*$.

2. Estimate the S.E. of $\hat{\theta}$*: SE($\hat{\theta}$*) = sqrt[Var($\hat{\theta}$*)].

C.I. Application: The Bootstrap Percentile Method

There are many ways to construct a C.I. using bootstrapping. An easy one: use the distribution of the $\hat{\theta}$*'s to compute directly a C.I. –i.e., without computing $q^* = \theta^* - \hat{\theta}$. This is the *bootstrap percentile method*.

The percentile method uses the distribution of $\hat{\theta}$* as an approximation to the distribution of $\hat{\theta}$. It is very simple, but not as appealing, since comparing differences is sounder.

Example: We construct a 95% C.I. for the variance of S&P 500 returns (continuation of previous example). Using the boot.ci function, with type=**perc**, from boot library:

```
boot.ci(boot.samps, type = "perc")
> boot.ci(boot.samps, type = "perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.samps, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.0014,  0.0020 )
Calculations and Intervals on Original Scale
```

- Check results by sorting boot.samps$t.

```
> new <- sort(boot.samps$t)
> new[25]                                    # CI's Lower Bound
[1] 0.001398215
> new[975]                                   # CI's Upper Bound
[1] 0.001955096
```

Or for σ, the 95% CI is given by (**3.74%**, **4.42%**).

```
> new <- sort(boot.samps$t)
> new[25]                                    # CI's Lower Bound
```

```
[1] 0.001398215
> new[975]                                          # CI's Upper Bound
[1] 0.001955096
```

Or for σ, the 95% CI is given by (**3.74%**, **4.42%**). ¶

C.I. Application: The Parametric Bootstrap Method

If we assume the data is from a parametric model (say, from a Normal or a Gamma distribution), we can use the parametric bootstrap to access the uncertainty (variance, C.I.) of the estimated parameter. In a parametric bootstrap, we generate bootstrap samples from the assumed distribution, based on moments computed from the sample. We do not use the ED.

Suppose we have a sample with N observations drawn from $F(x;\, \theta)$:

$$\{x_1, x_2, x_3, \ldots, x_N\}$$

In the parametric bootstrap, we know $F(x;\, \theta)$, the distribution of x, but we do not know its parameters. Suppose there is only one unknown parameter, θ (say, the variance). From the sample, we compute $\hat{\theta}$, the estimator of θ. Then, we bootstrap from $F(x;\, \hat{\theta})$ and proceed as before to form a C.I..

- Steps:

1. Draw B samples of size N from $F(x;\, \hat{\theta})$.
2. For each bootstrap sample, $\{x_1^*, x_2^*, x_3^*, \ldots, x_N^*\}$, calculate $\hat{\theta}^*$. ⇒ Get B $\hat{\theta}^*$.
3. Estimate a C.I. using the previous methods.

Example: Suppose **S&P 500 monthly returns** follow a N(0, σ^2). We estimate σ^2 with $s^2 = \mathbf{0.04070}^2$.

```
> lr_var
[1] 0.001656445

x <- rnorm(T_s*sim_size, mean=0, sd=lr_sd)                  # generate normal data
boot_sample <- matrix(x, nrow=T_s, ncol=sim_size)           # organize simulated data
boots_vars  <- colVars(boot_sample)                         # compute variances
q_star <- boots_vars - lr_var
q <- quantile(q_star, c(0.025, 0.975))
ci <- lr_var - c(q[2], q[1])
> ci
      97.5%        2.5%
0.001547382 0.001760286
```

Or for σ, the 95% CI is given by (**3.94%**, **4.20%**). Very close to the C.I.'s we obtained before assuming a Normal distribution for returns. Not a surprise! ¶

Note: In the previous example, to gauge the uncertainty of the estimation of s^2, we can also compute the sample standard error, SE(s^2).

- Steps

1. Draw B samples of size N from a N(0, s^2) ⇒ Get B s^{2*}.
2. Estimate the variability of s^2 by computing the sample variance

$$\text{Var}(s^2) = \frac{1}{B-1}\sum_{i=i}^{B}(s_i^{2*} - s_B^2)^2,$$

where $s_B^2 = \frac{1}{B}\sum_{i=i}^{B} s_i^{2*}$.

3. Estimate the S.E. of s^2**:** SE(s^2) = sqrt[Var(s^2)].

Remark**:** An important difference between the nonparametric and parametric bootstrap procedures is that in the nonparametric procedure, only values of the original sample appear in the bootstrap samples. In the parametric bootstrap, the range of values in the bootstrap sample is the entire support of $F(x;\ \theta)$. In the parametric bootstrap of the above example, the values in the bootstrap sample could be any value between negative and positive infinity.

C.I. Application: Bootstrapping – Why?

Question: Why do we need a bootstrap?

- Sample sizes are "small" and asymptotic assumptions do not apply
- DGP assumptions are violated.
- Distributions are complicated.

Usually, we would not use a bootstrap to compute C.I.'s for the mean; in general, the normal distribution works well, as long as N is large enough. The bootstrap is used to generate standard errors for estimates of other statistics where the normal distribution is not a good approximation. A typical example is the median, where for non-normal underlying distributions the SE of the median is complicated to compute.

Efron (1979) is the seminal paper. But, the related literature is older. It became popular in the 1980's due to the explosion of computer power.

Disadvantages and Advantages:
- Disadvantage: Only *consistent* results, no finite sample results.
- Advantage: Simplicity.

C.I. Application: Value-at-Risk

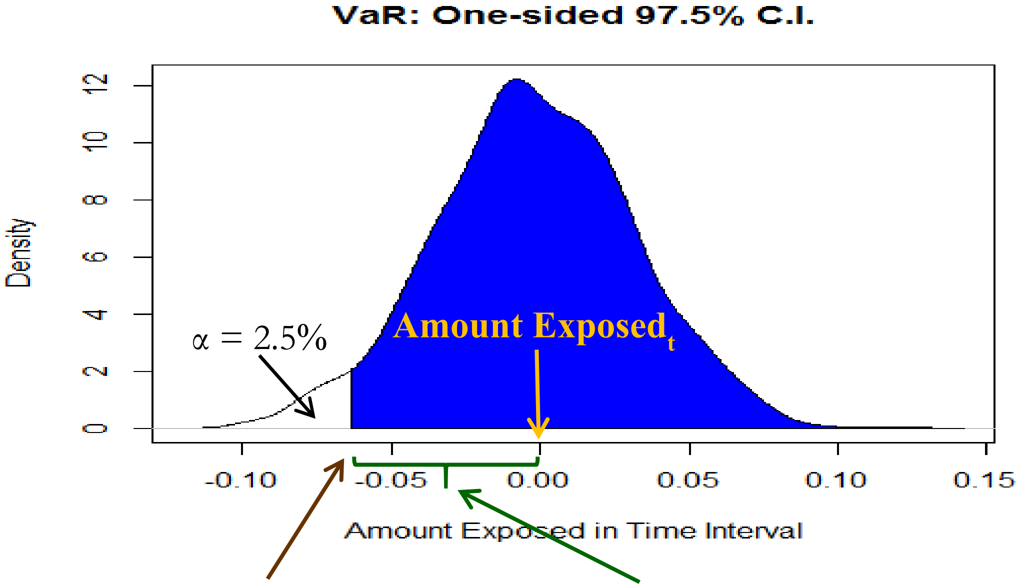
We usually measure risk of an asset/investment with its volatility.
Volatility is calculated including positive (right tail) and negative (left tail) returns. Investors, however, love the right tail of the returns distribution, but dislike the other tail. *Value-at-Risk* (VaR) focuses on the left tail.
VaR gives a formal definition of "worst case scenario" for an asset.
VaR: *Maximum expected amount (loss)* in a given time interval within a (*one-sided*) (1 - α)% C.I.:

$$\text{VaR}(1 - \alpha) = \text{Amount exposed} * (1 + \text{worst \% change scenario in C.I.})$$

It is common to express the "*expected loss*" relative to today's expected value of asset/investment:

$$\text{VaR-mean}(1 - \alpha) = \text{VaR} – \text{E[Amount exposed]}$$

Example: Let $\alpha = .05$
VaR = Amount exposed * (1 + worst change scenario in 97.5% C.I.).
VaR-mean(97.5%) = VaR – E[Amount exposed]. ¶



VaR(97.5%): Minimum Amount within C.I. VaR-mean(97.5%). ¶

When a company is involved with transactions denominated in foreign currency (FC), it is exposed to *currency risk*. *Transaction exposure* (TE) provides a simple measure of this exposure:

$$TE_t = \text{Value of a fixed future transaction in FC} * S_t$$

where S_t is the exchange rate expressed as units of domestic currency (USD for us) per unit of FC (say, EUR).

Example: A Swiss company, Swiss Cruises, sells packages in USD.
Amount = **USD 1 million**.
Payment: 30 days.
S_t = **0.92 CHF/USD**

$$\Rightarrow \mathbf{TE_t = USD\ 1M} * \mathbf{0.92\ CHF/USD} = \mathbf{CHF\ 0.92M}.$$

If S_t is described by a Random Walk, then TE_t is a forecast of the value of the transaction in 30 days (TE_{t+30}). ¶

C.I. Application: Range Estimates and VaR

Swiss Cruises wants a measure of the uncertainty related to the amount to receive in CHF in 30 days, since S_{t+30} is unknown.
We can use a range to quantify this uncertainty; we want to say

$$TE_{t+30} \in [TE_{LB}, TE_{UB}]. \qquad \text{with high probability.}$$

To determine this range for TE, we assume that (log) changes in S_t, $e_{f,t}$, are normally distributed: $e_{f,t} \sim N(\mu, \sigma^2)$.
Then, we build a (1-α)% interval around the mean: $[\mu \pm z_{\alpha/2}\, \sigma]$.
Usual α's in interval calculations: $\alpha = .05 \Rightarrow z_{.025} = 1.96\ (\approx 2)$
$\alpha = .02 \Rightarrow z_{.01} = 2.33$
As usual, we estimate (μ, σ) using ($\bar{X}$, s). ¶

Example: Range estimate based on a Normal distribution.
Assume Swiss Cruises believes that CHF/USD monthly changes ($e_{f,t}$) follow a normal distribution. Swiss Cruises estimates the mean and variance using the last 15 years of data:
$\bar{X}$ = Monthly mean = **-0.00152** ≈ **-0.15%**
s^2 = Monthly variance = 0.001014 (⇒ s = **0.03184**, or **3.18%**)
$e_{f,t} \sim N(\mathbf{-0.00152}, \mathbf{0.03184}^2)$ $e_{f,t}$ = CHF/USD monthly changes.

Swiss Cruises constructs a 95% CI for CHF/USD monthly changes.
Recall that a 95% C.I. for $e_{f,t+30}$ (which applies to any t) is given by:

$$e_{f,t} \in [\mathbf{-0.00152} \pm 1.96 * \mathbf{0.03184}] = [\mathbf{-0.06393}; \mathbf{0.06089}].$$

Based on this range for $e_{f,t}$, we build a 95% C.I. for S_{t+30} and, then, for TE_{t+30} (= **USD 1M** * S_{t+30}).

$$e_{f,t+30} \in [\mathbf{-0.00152} \pm 1.96 * \mathbf{0.03184}] = [\mathbf{-0.06393}; \mathbf{0.06089}].$$

Now, we derive a range for S_{t+30}:
(A) Upper bound
$S_{t+30,UB} = S_t * (1 + \mathbf{e_{f,t,UB}})$ = **0.92 CHF/USD** * (1 + **0.06089**) = **0.97602 CHF/USD**

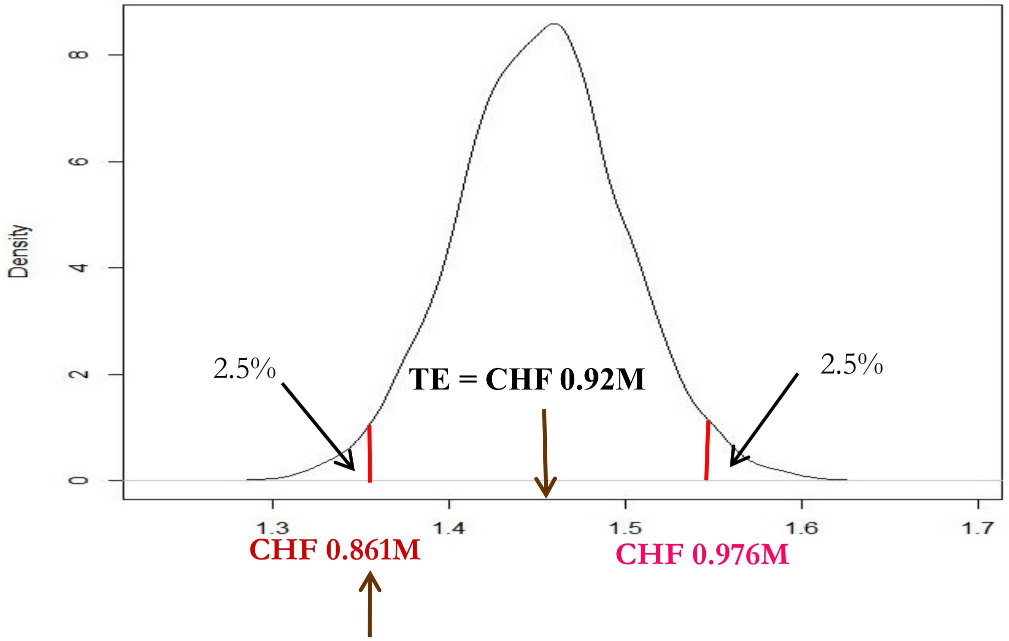
(B) Lower bound
$S_{t+30,LB} = S_t * (1 + \mathbf{e_{f,t,LB}})$ = **0.92 CHF/USD** * (1 + (**-0.06393**))] = **0.86118 CHF/USD**
⇒ $S_{t+30} \in$ [= **0.86118 CHF/USD**; **0.97602 CHF/USD**].

Finally, we derive the bounds for the TE:
(A) Upper bound (with $S_{t+30,UB}$ = **0.97602 CHF/USD**)
TE_{UB}: **USD 1M** * [**0.97602 CHF/USD**] = **CHF 976,019**.

(B) Lower bound (with $S_{t+30,LB}$ = **0.86118 CHF/USD**)
TE_{LB}: **USD 1M** * [**0.86118 CHF/USD**] = **CHF 861,184**.
⇒ $TE_{t+30} \in$ [**CHF 0.861M**; **CHF 0.976M**]. ¶

- The lower bound, for a receivable, represents the worst case scenario within the interval in 30 days. That is, this is the VaR:

VaR = Amount exposed * (1+ worst % change scenario in C.I.)
= $\mathbf{TE_t} * (1 + e_{f,LB})$



VaR(97.5%): Minimum revenue within a 97.5% C.I.

It is common to express the "*expected loss*" relative to today's expected value of transaction (or asset):

$$\text{VaR-mean} = \text{VaR} - \mathbf{TE_t} = \mathbf{TE_t} * (1 + e_{f,LB}) - \mathbf{TE_t}$$
$$= \mathbf{TE_t} * \mathbf{e_{f,LB}}$$

Or just

VaR-mean = Amount exposed * worst case scenario

The minimum revenue to be received by SC in the next 30 days, within a 97.5% CI.

VaR(97.5%) = **CHF 0.92M** * [1+ (**-0.00152** – 1.96 * **0.03184**)]
= **CHF 0.8612M**.

Interpretation of VaR: If SC expects to cover expenses with this USD 1M inflow, the maximum amount in CHF to cover, within a 97.5% one-sided CI, should be **CHF 0.8612M**.

Relative to today's valuation (or *expected valuation*, according to RWM), the maximum expected loss in 30 days within a 97.5% one-sided C.I. is:

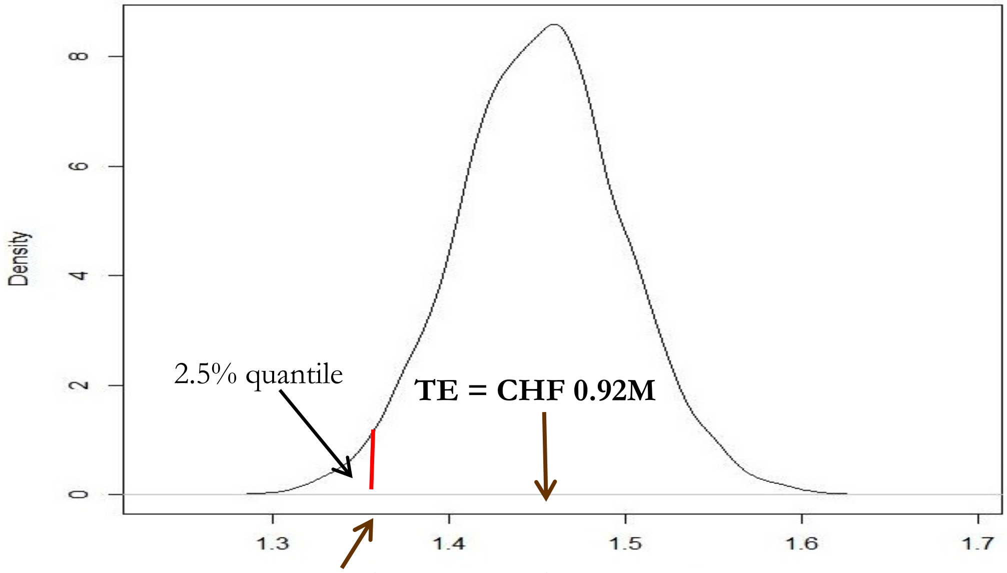
VaR-mean(.975)= **CHF 0.8612M** – **CHF 0.92M** = **CHF -0.0927M**.

Note that we can also compute the VaR-mean as:

VaR-mean(.975)= **CHF 1.45M** * (**-0.00152** – 1.96 * **0.03184**)
= **CHF -0.0588M**. ¶

Technically speaking, the VaR is a *quantile*, where a quantile is the fraction of observations that lie below a given value (in this case the VaR).

Example: In the previous example, the 0.025 quantile (or 2.5% quantile) for expected loses is **CHF -0.0588M**.



VaR (97.5%) = **0.8612M**

VaR(97.5%)-mean = **CHF -0.0588M**

Note: We could have used a different quantile –i.e. a different significant level- to calculate the VaR, for example 1% (⇒ $z_{.01} = 2.33$). Then,

VaR(99%) = **CHF 0.92M** * [1+ (**-0.00152** – 2.33 * **0.03184**)]
= **CHF 0.8503M** (A more *conservative* bound.)
⇒ VaR-mean (.99) = **CHF 0.8503M** – **CHF 0.92M** = **CHF -0.0697M**.

Interpretation of VaR-mean: Relative to today's valuation (or *expected valuation*, according to RWM), the maximum *expected loss* with a 99% "chance" is **CHF -0.0697M**.

Note: As the C.I. gets wider, Swiss Cruises can spend less CHF on account of the **USD 1M** receivable. ¶

C.I. Application: Range Estimates and VaR - Bootstrap

VaR is a statistic –a function of the data. We can do an empirical bootstrap to calculate the mean, SE (=SD), C.I., etc.

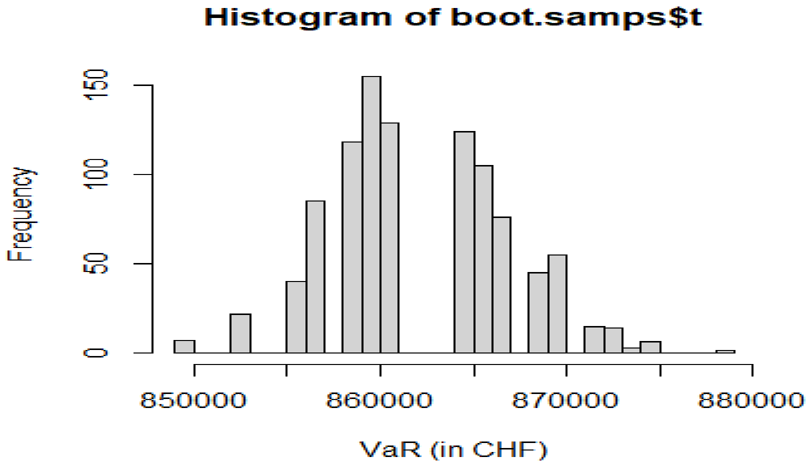
Example: We want to calculate the average VaR(97.5%) and its S.E., using all CHF/USD data from 1990:Jan - 2020:Sep. Then,

```
chfusd <- read.csv("http://www.bauer.uh.edu/rsusmel/4386/chfusd.csv",sep=",")  # Data
S <- chfusd$CHF_USD                                  # Extract CHF_USD column of the data
T <- length(S)                                       # Check total T (1971:1 to 2017:1)
Tstart <- 229                                        # Start of sample period: 1990:1
```

```
Val <- 1000000                                    # Value of transaction in FC (in M)
S_0 <- S[T]                                       # Today's S_t
e <- log(SP[-1]/SP[-T])
T_s <- length(e_f)
alpha <- .05                                      # Specify alpha level for VaR
T_s_low <- round(T_s*alpha/2)                     # Obs corresponding to alpha/2*T_s
TE_o <- Val*S_0*(1+e_f)                           # calculate Original TE values
STE_o <- sort(TE_o)                               # sort Original TE
VaR_o <- STE_o[T_s_low]                           # Original VaR
> VaR_o
[1] 860293

# function to obtain VaR from the data
varisk <- function(data, i) {
        d <-data[i]
TE <- Val*S_0*(1+d)                               # calculate R TE values
STE <- sort(TE)                                   # sort TE
VaR <- STE[T_s_low]
return(VaR)
}

boot.samps <- boot(data=e_f, statistic=varisk, R=sim_size)
> boot.samps

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = e_f, statistic = varisk, R = sim_size)

Bootstrap Statistics :
   original   bias    std. error
t1*   860293 1929.305    4870.733

> boot.ci(boot.samps, type = "basic")             # boot computes the CI.
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.samps, type = "basic")

Intervals :
Level      Basic
95%   (849352, 867587 )

> mean(boot.samps$t)
[1] 862222.3
```

```
> sd(boot.samps$t)
[1] 4870.733
```

```
hist(boot.samps$t, xlab="VaR (in CHF)", breaks=30)
```



Bootstrap estimated VaR(97.5%) = **CHF 0.8622M**. ¶

C.I. Application: Performance Evaluation

In the 1990s, Bankers Trust evaluated traders based on a risk-adjusted performance measure called RAROC: Risk-adjusted return on capital.

RAROC = Profits/Capital-at-Risk

RAROC adjusts profits taking into account the exposure of the bank, called *capital-at-risk*. BT defined this exposure as the amount of capital needed to cover 99% of the maximum expected loss over a year.

That is, capital-at-risk is the worst loss within one-sided 99% C.I. We called this VaR-mean(99%).

The rationale for this measure: BT needs to hold enough cash to cover 99% of possible losses.

Example: Ranking two traders I and II, dealing in different markets.

	Segment	Profits (in USD, annualized)	Position (in USD)	Volatility (annualized)
Trader I	Futures stock indices	**3.3 M**	**45 M**	**21%**
Trader II	FX Market	**3.0 M**	**58 M**	**14%**

To calculate RAROC, we calculate the VaR-mean(99%) –i.e., worst possible loss in a 99% CI.

Assuming normality for profits with mean equal to zero (not important, since all traders are evaluated using the same mean). Then, (since α =.01 ⇒ $z_{.01}$ = **2.33**):

VaR-mean(99%) = Amount exposed * worst case scenario
= Position * $z_{.01}$ * Volatility

Since α =.01 ⇒ $z_{.01}$ = **2.33**.

(1) Calculate VaR-mean (99%) for each trader (under normal distribution)

Trader I: **USD 45M** * **2.33** * **0.21** = USD 22,018,500.
Trader II: **USD 58M** * **2.33** * **0.14** = USD 18,919,600.

(2) Calculate RAROC:

Trader I: RAROC = **USD 3.3M** /USD 22,018,500 = .1499.
Trader II: RAROC = **USD 3.0M** /USD 18,919,600 = .1586.

Conclusion: Once adjusted for risk, Trader II provided a better return. ¶

Lecture 3 – Least Squares

So far, we focused on one RV only, say returns and learning about its distribution, for example, using descriptive statistics. Now, we focus on relations between variables, say between y & x.

The usual notation applies here: y is the dependent variable and x is a set of independent variables

Example: According to the CAPM, excess returns for IBM (y) are proportional to excess returns for the "market" (x). We can express this relation as:

$$y_i = \alpha + x_i\,\beta + \varepsilon_i, \qquad i = 1, 2, \ldots., T.$$

where ε_i is an "error" term that follows a $N(0, \sigma^2)$. The term ε_i represents the effects of individual variation that have not been controlled for with x_i and α & β are parameters. ¶

In the example, we have that IBM excess returns are only related ("explained") by the market. This is a *one variable model*.

But, we could have used more variables, for example the 3 factors in the standard Fama-French model: Market, SMB (size factor), and HML (book-to-market factor). This represents a *multivariate model* for IBM returns:

$$y_i = \alpha + x_{1,i}\,\beta_1 + x_{2,i}\,\beta_2 + x_{3,i}\,\beta_3 + \varepsilon_i$$

- Though, not necessary correct, we usually think of y as the *endogenous* variable and x as the *exogenous* or *predetermined* variable.

This lecture: Estimation of population parameters α & β and the properties of estimators.

- Old method: Gauss (1795, 1801) used it in astronomy.

Idea: There is a functional form relating a dependent variable Y and *k* variables X. This function depends on unknown parameters, θ. The relation between Y and X is not exact. There is an error, ε. We have *T* observations of Y and X.

We will assume that the functional form is known:

$$y_i = f(x_i, \theta) + \varepsilon_i \qquad i = 1, 2, \ldots., T.$$

We will estimate the parameters θ by minimizing a sum of squared errors:

$$\min_\theta \{S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2\}$$

The estimator obtained by this minimization is called the *Least Squares* (LS) estimator.

The functional form, $f(x_i, \boldsymbol{\theta})$, is dictated by theory or experience. When the function $f(x_i, \boldsymbol{\theta})$ is linear, $f(x_i, \boldsymbol{\theta}) = \alpha + x_{1,i}\,\beta_1 + x_{2,i}\,\beta_2 + x_{3,i}\,\beta_3 + \ldots + x_{k,i}\,\beta_k$,

we call the estimator the *Ordinary Least Squares* (OLS) estimator.

Notation: In lecture 2, we used ^ over the estimator of the parameter of interest. For example, $\hat{\boldsymbol{\theta}}$ is the estimator of the parameter $\boldsymbol{\theta}$. Sometimes, to emphasize the method of estimation, we add to the estimated parameter the initials of the method used, say $\hat{\theta}_{LS}$.

For historical reasons, in the linear model, **b** is popularly used to denote the OLS estimator of $\boldsymbol{\beta}$.

Example 1: We want to study the effect of the tech boom (**x**) on the San Francisco housing market (**y**). We rely on a simple linear model, with only one explanatory variable, the tech boom variable. That is,

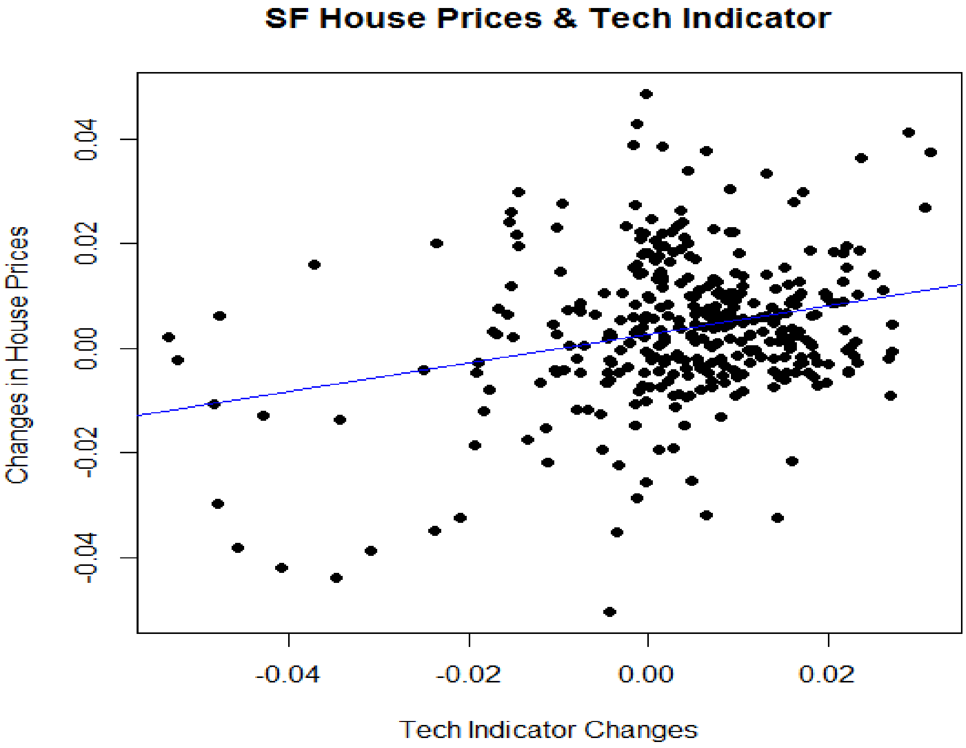
$$y_i = \alpha + x_i \beta + \varepsilon_i$$

In this model, we are interested in estimating β, our parameter of interest. β measures the *marginal effect* of x on y. We can use the estimate of β to check if the high tech boom has a positive effect on SF housing prices. In this case we test:

H_0 (No or Negative effect): $\beta \leq 0$.

H_1 (Positive effect): $\beta > 0$.

We have monthly data on SF Housing Prices and a Tech Indicator, developed by the Federal Reserve. We transform the data in percentage changes. Below, we plot the data: SF House Prices vs Tech Indicator (both in % changes).



Example 2: We want to study the effect of a CEO's network (**x**) on a firm's CEO's compensation (**y**). We build a CEO's compensation model including a CEO's network (**x**) and

other "*control variables*" (**W**: education, experience, etc.), controlling for other features that make one CEO's compensation different from another. That is,

$$y_i = f(x_i, \mathbf{W}_i, \boldsymbol{\theta}) + \varepsilon_i \qquad i = 1, 2, ..., T.$$

The term ε_i represents the effects of individual variation that have not been controlled for with $\mathbf{W}_i$ or x_i and $\boldsymbol{\theta}$ is a vector of parameters.

Usually, $f(x, \theta)$ is linear. Then, the compensation model becomes:

$$y_i = \alpha + x_i\,\beta + W_{1,i}\,\gamma_1 + W_{2,i}\,\gamma_2 + ... + \varepsilon_i$$

Again, in this model, we are interested in the estimation of β, our parameter of interest, which measures the effect of a CEO's network on a CEO's compensation. ¶

- Objective function:

$$S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2 = \Sigma_i\, [y_i - f(x_i, \theta)]^2$$

- We want to minimize w.r.t to θ. That is,

$$\min_\theta \{S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2 = \Sigma_i\, [y_i - f(x_i, \theta)]^2\}$$

$$\Rightarrow \frac{\partial S(x_i,\theta)}{\partial \theta} = -2\sum_i^T\{y_i - f(x_i,\theta)\}f'(x_i,\theta)\}$$

$$\text{f.o.c.} \quad \Rightarrow -2\sum_i^T\{y_i - f(x_i,\hat{\theta}_{LS})\}f'(x_i,\hat{\theta}_{LS})\} = 0$$

$$\Rightarrow \sum_i^T\{y_i - f(x_i,\hat{\theta}_{LS})\}f'(x_i,\hat{\theta}_{LS})\} = 0$$

The f.o.c. deliver the *normal equations*. The solution to the normal equation, $\hat{\theta}_{LS}$, is the Least Squares estimator.

The estimator $\hat{\theta}_{LS}$ is a function of the data (y_i, x_i).

LS Estimation – One Variable Linear Model

One explanatory variable in a linear model:

$$f(x_i, \theta) = \beta_1 + \beta_2\, x_i$$

Linear Model: $\quad y_i = \beta_1 + \beta_2\, x_i + \varepsilon_i$

Objective function: $\quad S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2 = \Sigma_i\, (y_i - \beta_1 - \beta_2\, x_i)^2$

First, we take first derivatives:

(β_1): $\;2\,\Sigma_i\,(y_i - \beta_1 - \beta_2\, x_i)\,(-1)$

(β_2): $\;2\,\Sigma_i\,(y_i - \beta_1 - \beta_2\, x_i)\,(-x_i)$

Second, we get the f.o.c. (2 equations, 2 unknowns):

(β_1): $\;2\,\Sigma_i\,(y_i - b_1 - b_2\, x_i)\,(-1) = 0 \qquad \Rightarrow \Sigma_i\,(y_i - b_1 - b_2\, x_i) = 0 \qquad (1)$

(β_2): $\;2\,\Sigma_i\,(y_i - b_1 - b_2\, x_i)\,(-x_i) = 0 \qquad \Rightarrow \Sigma_i\,(y_i\, x_i - b_1 x_i - b_2\, x_i^2) = 0 \; (2)$

Now, we solve for b_1 & b_2, the OLS estimators.

From (1): $\quad \Sigma_i\, y_i - \Sigma_i\, b_1 - b_2\,\Sigma_i\, x_i = 0 \qquad \Rightarrow b_1 = \bar{y} - b_2\,\bar{x}$

From (2): $\Sigma_i\, y_i\, x_i - (\bar{y} - b_2\, \bar{x})\, \Sigma_i\, x_i - b_2\, \Sigma_i\, x_i^2 = 0 \Rightarrow b_2 = \frac{\Sigma_i (y_i - \bar{y}) x_i}{\Sigma_i (x_i - \bar{x}) x_i}$

or, more elegantly,

$$b_2 = \frac{\Sigma_i (y_i - \bar{y})(x_i - \bar{x})}{\Sigma_i (x_i - \bar{x})^2} = \frac{cov(y_i, x_i)}{var(x_i)}$$

Note that we need $var(x_i) \neq 0$ to get b_2.

• Suppose y_i represents IBM excess returns (r_{IBM} - r_f) at time *t* and x_i represents Market excess returns (say, r_{Mkt} - r_f) at time *t*. Then, b_2 estimates the stock's beta in the CAPM.

Then, b_2 estimates the stock's beta in the CAPM:

$$b_2 = \hat{\beta}_{CAPM} = \frac{cov(r_{IBM} - rf,\ r_{Mkt} - rf)}{var(r_{Mkt} - rf)}$$

That is, the CAPM β is the ratio of a covariance over a variance.

• Interpretation of coefficients

- b_1 estimates the *constant* of the regression, the value of y, when x equals to 0. In the case of the CAPM, it should be 0.
- b_2 estimates the *slope* of the regression, the marginal effect –i.e., the first derivative of y_i with respect to x_i:

$$\frac{\delta y_i}{\delta x_i} = \beta_2$$

That is, if x increases by one unit (say, %), then y increased by b_2 units (say, b_2%). In the case of the CAPM, it measures the *systematic exposure* (or volatility relative to the market) of a stock.

• Conditional Prediction

Suppose analysts estimate that x_i will be 10%, then you estimate (or predict, given the 10% value of x_i):

Predicted $y_i|x_i$=.10 = $b_1 + b_2 * .10$.

We will call the Predicted y_i = $\hat{y}_i$ = fitted value.

Example: We estimate the CAPM for **IBM returns** using *lm* function in R.

• Import data with read function

```
SFX_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",
head=TRUE, sep=",")
```

• Extract variables from imported data

```
x_ibm <- SFX_da$IBM                  # extract IBM price data
x_Mkt_RF <- SFX_da$Mkt_RF            # extract Market excess returns (in %)
x_RF <- SFX_da$RF                    # extract Risk-free rate (in %)
```

• Define log returns & adjust size of variables accordingly

```
T <- length(x_ibm)                   # sample size
lr_ibm <- log(x_ibm[-1]/x_ibm[-T])   # create IBM log returns (in decimal returns)
Mkt_RF <- x_Mkt_RF[-1]/100           # Adjust sample size to ( T-1) by removing 1st obs
RF <- x_RF[-1]/100                   # Adjust sample size and use decimal returns.
```

• Define excess returns and estimate CAPM with lm function:

```
ibm_x <- lr_ibm – RF                              # IBM excess returns
fit_ibm_capm <- lm(ibm_x ~ Mkt_RF)                # lm (=linear model) package in R
summary(fit_ibm_capm)                             # print lm results
```

```
> summary(fit_ibm_capm)
Call:
lm(formula = ibm_x ~ Mkt_RF)

Residuals:
     Min        1Q    Median        3Q       Max
-0.314401 -0.031692 -0.000537  0.031447  0.248201

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.005791   0.002487  -2.329   0.0202 *       b1 = -0.005791
xMkt_RF          0.895774   0.053867  16.629   <2e-16 ***   b2 = 0.895774
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.05887 on 567 degrees of freedom
Multiple R-squared:  0.3278,    Adjusted R-squared:  0.3266
F-statistic: 276.5 on 1 and 567 DF,  p-value: < 2.2e-16
```

Interpretation of b_1 and b_2:

b_1 = constant. If gives the additional IBM return, after excess market returns are incorporated. Under the CAPM, b_1 should be close to 0.

b_2 = slope. It is the marginal effect. If market excess returns increase by 1%, IBM excess returns increase by b_2%. In this case, by .90%. The estimate of the CAPM $\beta < 1$, implying that IBM is less volatile (“safer”) than the market.

Conditional prediction of IBM excess returns:

Suppose market excess returns increase are 10%, then we predict IBM excess returns = **-0.005791** + **0.895774** * .10 = **0.08378** (8.38%). ¶

LS Estimation – Application: Hedging

In the linear model, we can estimate the optimal hedge ratio using a regression. To see this, we derive the optimal hedge ratio for a position in foreign currency (FC).

Notation:

S_t: Exchange rate at time t. We use direct quotations, that is, DC units per unit of FC, say S_t = 1.30 USD/GBP.

$F_{t,T}$: Forward/Futures price at time t with a T maturity.

n_s: Number of units of foreign currency held.

n_f: Number of futures foreign exchange units held (opposite position).

$\pi_{h,t}$: (Uncertain) profit of the hedger at time t.

ΔX: Change in X (= $X_t - X_{t-1}$)

$\boldsymbol{h}$ = hedge ratio =(n_f/n_s)= Number of futures per spot in position.

We want to calculate $\boldsymbol{h^*}$ (*optimal h*): We minimize the variability of $\pi_{h,t}$.

$\pi_{h,T} = \Delta S_t\, n_s + \Delta F_{t,T}\, n_f$ (Or, $\pi_{h,T}/n_s = \Delta S_t + \boldsymbol{h}\, \Delta F_{t,T}$.)

We want to select h to minimize:

$$\text{Var}(\pi_{h,T}/n_s) = \text{Var}(\Delta S_T) + \boldsymbol{h}^2\, \text{Var}(\Delta F_{t,T}) + 2\, \boldsymbol{h}\, \text{Covar}(\Delta S_T, \Delta F_{t,T})$$
$$= \sigma_S^2 + \boldsymbol{h}^2 \sigma_F^2 + 2\, \boldsymbol{h}\, \sigma_{SF}$$

f.o.c.

$$2\, \boldsymbol{h^*}\, \sigma_F^2 + 2 * \sigma_{SF} = 0$$
$$\Rightarrow \boldsymbol{h^*} = -\sigma_{SF}/\sigma_F^2$$

<u>Note</u>: A covariance over a variance. It can be estimated by LS:

$\Delta S_t = \beta_1 + \beta_2\, \Delta F_{t,T} + \varepsilon_t$ $\Rightarrow$ b_2 estimates $\boldsymbol{h^*}$.

Example: In March, we are long a GBP 1M position. We are uncertain about S_t in the next 90 days. We hedge this position using June GBP futures (size of contract = GBP 62,500). We want to determine $\boldsymbol{h^*}$.

Get Data (S_t and $F_{t,90\text{-days}}$, for say 10 years). Do a regression.

$$\Delta S_t = \beta_1 + \beta_2\, \Delta F_{t,T} + \varepsilon_t$$

Suppose we estimate this regression:

$$\Delta S_t = .001 + \mathbf{.82}\, \Delta F_{t,T},$$
$$\Rightarrow h^* = -.82.$$

Now, we determine the number of June GBP futures contracts:

$\Rightarrow n_f$/size of the contract = h * n_s / size of the contract =

= -.82 * 1,000,000 / 62,500 = -13.12 ≈ 13 contracts sold! ¶

LS Estimation – OLS

LS is a general estimation method. It allows any functional form for the relation between y_i and x_i. In this lecture, we cover the case where $f(x_i, \theta)$ is linear. When the relation is linear, we do OLS estimation.

We assume a linear system with k independent variables and T observations. That is,

$$y_i = \beta_1 x_{1i} + \beta_2\, x_{2i} + \ldots + \beta_k\, x_{ki} + \varepsilon_i, \quad i= 1, 2, \ldots., T$$

The whole system (for all i) is:

$$y_1 = \beta_1 x_{11} + \beta_2\, x_{12} + \ldots + \beta_k\, x_{k1} + \varepsilon_1$$
$$y_2 = \beta_1 x_{12} + \beta_2 x_{22} + \ldots + \beta_k x_{k2} + \varepsilon_2$$
$$\ldots . \qquad .. .. \qquad . \ldots \qquad \ldots$$
$$y_T = \beta_1 x_{1T} + \beta_2 x_{2T} + \ldots + \beta_k x_{kT} + \varepsilon_T$$

It is cumbersome and complicated to write the whole system. Using linear algebra, we can rewrite the system in a more compact and simplify derivations.

For example, after some definitions, we can write the whole system as:

$$\mathbf{y} = f(\mathbf{X}, \theta) + \boldsymbol{\varepsilon} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Linear Algebra: Brief Review – Matrix

Life (& notation) becomes easier with linear Algebra. Concepts:

- Matrix.

A matrix is a set of elements, organized into rows and columns

$$\begin{array}{cc} & \text{Columns} \\ \text{Rows} & \begin{bmatrix} a & b \\ c & d \end{bmatrix} \end{array}$$

- *a* & *d* are the diagonal elements.
- *b* & *c* are the off-diagonal elements.

Matrices are like plain numbers in many ways: they can be added, subtracted, and, in some cases, multiplied and inverted (divided).

Linear Algebra: Matrices and Vectors

Examples:

$$A = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}; \qquad b = [b_1 \quad b_2 \quad b_3].$$

Dimensions of a matrix: numbers of rows by numbers of columns. The Matrix **A** is a 2x2 matrix, **b** is a 1x3 matrix.

A matrix with only 1 column or only 1 row is called a *vector*.

If a matrix has an equal numbers of rows and columns, it is called a *square* matrix. Matrix **A**, above, is a square matrix.

Usual Notation:	Upper case letters	⇒ matrices
	Lower case	⇒ vectors

Linear Algebra: Matrices – Information

Information is described by data. A tool to organize the data is a list, which we call a vector. Lists of lists are called matrices. That is, we organize the data using matrices.

We think of the elements of **X** as data points ("data entries", "observations"), in economics, we usually have numerical data.

We store the data in rows. In a *T*x*k* matrix, **X**, over time we build a database:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix}$$

Once the data is organized in matrices it can be easily manipulated: multiplied, added, etc. (This is what Excel does).

Linear Algebra: Matrices in Econometrics

In econometrics, we have a model $y = f(x_1, x_2, ..., x_k)$, which we want to estimate. We collect data, say T (or N) observations, on a dependent variable, **y**, and on k explanatory variables, **X**.

Under the usual notation, vectors will be column vectors: **y** and $\mathbf{x_k}$ are Tx1 vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \qquad \& \qquad \boldsymbol{x_j} = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jT} \end{bmatrix} \qquad j = 1,..., k$$

X is a Txk matrix: $\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix}$

Its columns are the k Tx1 vectors $\mathbf{x_j}$. It is common to treat $\mathbf{x_1}$ as vector of ones, **ί.**

In general, we import matrices (information) to our programs.

Example: In R, we use the **read** function, usually followed by the type of data we are importing. Below, we import a comma separated values (csv) file with monthly CPIs and exchange rates for 20 different countries, then we use the **read.csv** function:

```
PPP_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/ppp_2020_m.csv",head=TRUE,sep=",")
```

The **names()** function describes the headers of the file imported (41 headers):

```
>names(PPP_da)
 [1] "Date"      "BG_CPI"    "IT_CPI"    "GER_CPI"   "UK_CPI"
 [6] "SWED_CPI"  "DEN_CPI"   "NOR_CPI"   "IND_CPI"   "JAP_CPI"
[11] "KOR_CPI"   "THAI_CPI"  "SING_CPI"  "MAL_CPI"   "KUW_CPI"
[16] "SUAD_CPI"  "CAN_CPI"   "MEX_CPI"   "US_CPI"    "EGY_CPI“ [...]
```

The **summary()** function provides some stats of variables imported:

```
>summary(PPP_da)
       Date        BG_CPI           IT_CPI          GER_CPI
 1/15/1971:  1   Min.   : 19.77   Min.   :  5.90   Min.   : 31.20
 1/15/1972:  1   1st Qu.: 49.32   1st Qu.: 32.25   1st Qu.: 57.17
 1/15/1973:  1   Median : 69.91   Median : 67.30   Median : 75.30
 1/15/1974:  1   Mean   : 67.92   Mean   : 60.14   Mean   : 72.29
 1/15/1975:  1   3rd Qu.: 89.40   3rd Qu.: 89.65   3rd Qu.: 91.17
 1/15/1976:  1   Max.   :109.71   Max.   :103.50   Max.   :106.60
 (Other)  :588
```

We extract a variable from the matrix by the name of file followed by **$** and the header of variable:

```
>x_chf <- PPP_da$CHF_USD               # extract CHF/USD exchange rate data
```

We can transform the vector x_chf. For example, for % changes:

```
T <- length(x_chf)                     # length of CHF/USD exchange rate data
lr_chf <- log(x_chf[-1]/x_chf[-T])     # create log returns (changes) for the CHF/USD. ¶
```

Linear Algebra: Special Matrices

• *Identity Matrix,* **I**: A square matrix with 1's along the diagonal and 0's everywhere else. Similar to scalar "1." **A*I** = **A**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• *Null matrix,* **0**: A matrix in which all elements are 0's. Similar to scalar "0." **A*0** = **0**

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Both are *diagonal* matrices ⇒ off-diagonal elements are zero.
Both are examples of *symmetric* and *idempotent* matrices. As we will see later:

- Symmetric: $\mathbf{A} = \mathbf{A}^T$
- Idempotent: $\mathbf{A} = \mathbf{A}^2 = \mathbf{A}^3 = \ldots$

Linear Algebra: Multiplication

We want to multiply two matrices: **A*B**. But, multiplication of matrices requires a *conformability condition.*

Conformability condition: The column dimensions of the lead matrix **A** must be equal to the row dimension of the lag matrix **B**.

If **A** is an (mx$\boldsymbol{n}$) and **B** an ($\boldsymbol{n}$xp) matrix (**A** has the same number of columns as **B** has rows), then we define the product of **AB**. **AB** is (mxp) matrix with its ik-th element is $\sum_{\boldsymbol{j}=1}^{\boldsymbol{n}} a_{i\boldsymbol{j}}\, b_{\boldsymbol{j}k}$

Q: What are the dimensions of the vector, matrix, and result?

$$aB = [a_{11} a_{12}] \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = c = [c_{11} \quad c_{12} \quad c_{13}]$$
$$= [a_{11}b_{11} + a_{12}b_{21} \quad a_{11}b_{12} + a_{12}b_{22} \quad a_{11}b_{13} + a_{12}b_{23}]$$

Dimensions: a(1x**2**), B(**2**x3) ⇒ c(1x3)

Example: We want to multiply **A** (2x**2**) and **B** (**2**x2), where **A** has elements $a_{i\boldsymbol{j}}$ and **B** has elements $b_{\boldsymbol{j}k}$. Recall the ik^{th} element is $\sum_{\boldsymbol{j}=1}^{\boldsymbol{n=2}} a_{i\boldsymbol{j}}\, b_{\boldsymbol{j}k}$

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} \mathbf{4} = 2*1+1*2 & \mathbf{3} = 2*0+1*3 \\ \mathbf{26} = 7*1+9*2 & \mathbf{27} = 7*0+9*3 \end{bmatrix}$$

$C_{2x2} = A_{2\,x2} * \; B_{2x2}$
Dimensions: A(2x**2**), B(**2**x2) ⇒ C(2x2), a square matrix. ¶

Linear Algebra: Transpose

The transpose of a matrix **A** is another matrix $\mathbf{A}^{\mathrm{T}}$ (also written **A′**) created by any one of the following equivalent actions:

–write the rows (columns) of **A** as the columns (rows) of $\mathbf{A}^{\mathrm{T}}$
–reflect **A** by its main diagonal to obtain $\mathbf{A}^{\mathrm{T}}$

Example: $A = \begin{bmatrix} 3 & 8 & -9 \\ 1 & 0 & 4 \end{bmatrix} \quad \Rightarrow A' = \begin{bmatrix} 3 & 1 \\ 8 & 0 \\ -9 & 4 \end{bmatrix}$. ¶

Some transpose results:
- If **A** is a $m \times n$ matrix ⇒ $\mathbf{A}^{\mathrm{T}}$ is a $n \times m$ matrix.
- (**A**')' = **A**
- Conformability changes unless the matrix is square.
- (**AB**)' = **B**'**A**'

Example: In econometrics, an important matrix is **X'X**. Recall **X** (usually, the matrix of *k* independent variables):

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix} \quad \text{a } (T\text{x}k) \text{ matrix}$$

Then,

$$\boldsymbol{X}' = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kT} \end{bmatrix} \qquad \text{a } (k\text{x}T) \text{ matrix}$$

Linear Algebra: Math Operations

Addition, Subtraction, Multiplication
- Addition: Just add elements

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

- Subtraction: Just subtract element

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

- Multiplication: Multiply each row by each column and add

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

- Scalar Multiplication: Multiply each element by the scalar, *k*

$$k \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$$

Examples:

Addition: $\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} + \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 7 & 11 \end{bmatrix}$

$A_{2x2} \; + \; B_{2x2} \; = \; C_{2x2}$

Subtraction: $\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 5 & 6 \end{bmatrix}$

Multiplication: $\begin{bmatrix} 2 & 1 \\ 7 & 9 \end{bmatrix} \text{x} \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 26 & 27 \end{bmatrix}$

$A_{2\,x2} \;\; \text{x} \;\; B_{2x2} \;\; = C_{2x2}$

Scalar Multiplication: $\frac{1}{8}\begin{bmatrix} 2 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 \\ 3/4 & 1/8 \end{bmatrix}$. ¶

Linear Algebra: Math Operations – X'X

A special matrix in econometrics, **X′X** (a *k*x*k* matrix):

$$\boldsymbol{X}\,(T\text{x}2) = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ \vdots & \vdots \\ x_{1T} & x_{2T} \end{bmatrix} \;\&\; \boldsymbol{X'} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \end{bmatrix}$$

$$\boldsymbol{X''X}\,(2\text{x}2) = \begin{bmatrix} \sum_{i=1}^{T} x_{1i}^2 & \sum_{i=1}^{T} x_{2i}x_{1i} \\ \sum_{i=1}^{T} x_{1i}x_{2i} & \sum_{i=1}^{T} x_{2i}^2 \end{bmatrix} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{T}} \begin{bmatrix} x_{1i}^2 & x_{2i}x_{1i} \\ x_{2i}x_{1i} & x_{2i}^2 \end{bmatrix}$$

$$= \sum_{\boldsymbol{i=1}}^{\boldsymbol{T}} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} \begin{bmatrix} x_{1i} & x_{2i} \end{bmatrix}$$

$$= \sum_{\boldsymbol{i=1}}^{\boldsymbol{T}} \boldsymbol{x}_i \boldsymbol{x}_i{}'$$

For the general case, with *k* explanatory variables, we have **X′X** (a *k*x*k* matrix):

$$\boldsymbol{X}\,(T\text{x}k) = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1T} & x_{2T} & \cdots & x_{kT} \end{bmatrix} \;\&\; X' = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1T} \\ x_{21} & x_{22} & \cdots & x_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k1} & x_{k2} & \cdots & x_{kT} \end{bmatrix}$$

$$\boldsymbol{X''X}\,(k\text{x}k) = \begin{bmatrix} \sum_{i=1}^{T} x_{1i}^2 & \sum_{i=1}^{T} x_{1i}x_{2i} & \cdots & \sum_{i=1}^{T} x_{1i}x_{ki} \\ \sum_{i=1}^{T} x_{2i}x_{1i} & \sum_{i=1}^{T} x_{2i}^2 & \cdots & \sum_{i=1}^{T} x_{2i}x_{ki} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{T} x_{ki}x_{1i} & \sum_{i=1}^{T} x_{ki}x_{2i} & \cdots & \sum_{i=1}^{T} x_{ki}^2 \end{bmatrix} =$$

$$= \sum_{i=1}^{T} \begin{bmatrix} x_{1i}^2 & \cdots & x_{1i}x_{ki} \\ \vdots & \ddots & \vdots \\ x_{ki}x_{1i} & \cdots & x_{ki}^2 \end{bmatrix} = \sum_{i=1}^{T} \begin{bmatrix} x_{1i} \\ \vdots \\ x_{ki} \end{bmatrix} \begin{bmatrix} x_{1i} & \cdots & x_{ki} \end{bmatrix}$$

$$= \sum_{i=1}^{T} \boldsymbol{x}_i \boldsymbol{x}_i{}'$$

Linear Algebra: Math Operations – ί'X

Recall **ί** is a column vector of ones (in this case, a *T*x1 vector):

$$\boldsymbol{\acute{\imath}} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Given **X** (*T*x*k*), then **ί′ X** is a 1x*k* vector:

$$\boldsymbol{\acute{\imath}'X} = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & \ddots & \vdots \\ x_{1T} & \cdots & x_{kT} \end{bmatrix} = \begin{bmatrix} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \boldsymbol{x_{1t}} & \ldots & \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} \boldsymbol{x_{kt}} \end{bmatrix}$$

<u>Note</u>: If $\mathbf{x}_1$ is a vector of ones (representing a constant in the linear classical model), then:

$$\boldsymbol{\acute{\imath}}'\,\mathbf{x}_1 = \sum_{t=1}^{T} x_{1t} = \sum_{t=1}^{T} 1 = T \quad (\textit{dot product, “}\bullet\textit{”})$$

Linear Algebra: Inverse of a Matrix

Identity matrix: **AI** = **A**, where $I_j = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$

Notation: $\mathbf{I_j}$ is a *j*x*j* identity matrix.

- Given **A** (*m*x*n*), the matrix **B** (*n*x*m*) is a *right-inverse* for **A** iff **AB** =$\mathbf{I_m}$
- Given **A** (*m*x*n*), the matrix **C** (*m*x*n*) is a *left-inverse* for **A** iff **CA** = $\mathbf{I_n}$

- **Theorem**: If **A** (*m*x*n*), has both a *right-inverse* **B** and a *left-inverse* **C**, then **C** = **B** = $\mathbf{A}^{-1}$

Note:

- If **A** has both a right and a left inverse, it is a square matrix. It is usually called *invertible*. We say “the matrix **A** is *non-singular*.”
- This matrix, $\mathbf{A}^{-1}$, is unique.
- If det(**A**) ≠ 0 ⇒ **A** is non-singular.

Linear Algebra: Symmetric Matrices

Definition:

If **A'** = **A**, then **A** is called a *symmetric* matrix.

In many applications, matrices are often symmetric. For example, in statistics the *correlation matrix* and *the variance covariance matrix*.

Symmetric matrices play the same role as real numbers do among the complex numbers.

We can do calculations with symmetric matrices like with numbers: for example, we can solve $\mathbf{B}^2 = \mathbf{A}$ for **B** if **A** is symmetric matrix (& **B** is square root of **A**.) This is not possible in general. **X’X** is symmetric. It plays a very important role in econometrics.

Linear Algebra: Operations in R

Many ways to create a vector (c, 2:7, seq, rep, etc) or a matrix (c, cbind, rbind). We use **c()**, the **combine function**:

```
> v1 <- c(1, 3, 8)              # a (3x1) vector (vectors are usually treated as a column list)
> v1
[1] 1 3 8

> A <- matrix(c(1, 2, 3, 7, 8, 9), ncol = 3)     # a (2x3) matrix
> A
     [,1] [,2] [,3]
[1,]    1    3    8
[2,]    2    7    9
```

```
> B <- matrix(c(1, 3, 1, 1, 2, 0), nrow = 3)
> B
        [,1] [,2]
[1,]    1    1
[2,]    3    2
[3,]    1    0
```

• Now, we use **rbind** to create A and **cbind** to create B

```
> v1 <- c(1, 3, 8)                # a (3x1) vector
> v2 <- c(2, 7, 9)
> A <- rbind(v1, v2)
> A                               # a (2x3) matrix
  [,1] [,2] [,3]
v1    1    3    8
v2    2    7    9

> v3 <- c(1, 3, 1)
> v4 <- c(1, 2, 0)
> B <- cbind(v3,v4)
> B                               # a (3x2) matrix
   v3 v4
[1,]  1  1
[2,]  3  2
[3,]  1  0
```

• Matrix addition/subtraction: +/-   –element by element.

• Matrix multiplication: **%*%**

```
> C <- A%*%B                      #A is 2x3; B is 3x2     => C is 2x2
> C
        [,1] [,2]
[1,]   18    7
[2,]   32   16
```

• Scalar multiplication: *****

```
> 2*C                    # elementwise multiplication of C by scalar 2
    [,1] [,2]
[1,]   36   14
[2,]   64   32
```

• <u>Note</u>: Usually, matrices will be data –i.e., read as input.

• Dot product "•" is a function that takes pairs of vectors and produces a number. For vectors **c** & **z**, it is defined as:

$$\boldsymbol{c} \bullet \boldsymbol{z} = c_1 * z_1 + c_2 * z_2 + \ldots + c_n * z_n = \sum_{i=1}^{n} c_i z_i$$

• Dot product with 2 vectors: v1 • v2: sum of the elementwise multiplied elements of both vectors

```
> t(v1) %*% v2                    # v1 <- c(1, 3, 8) & v2 <- c(2, 7, 9)
        [,1]
```

```
[1,]   95
```

- Dot product with a vector itself: v1 • v1: Sum of the square elements of vector

```
> t(v1) %*% v1
     [,1]
[1,]   74
```

- Dot product with **ί** (a vector of ones): sum of elements of vector

```
> i <- c(1,1,1);
> t(i) %*% v1                    # v1 <- c(1, 3, 8)
          [,1]
[1,]    12
```

- Product of 2 vectors: v1 & t(v2): A (3x3) matrix.

```
> v1%*%t(v2)                     # v1 <- c(1, 3, 8)  --a (3x1) vector x (1x3) vector
     [,1] [,2] [,3]
[1,]    2    7    9
[2,]    6   21   27
[3,]   16   56   72
```

- <u>Property of dot product</u>: If the dot product of two vectors is equal to zero, then the vectors are *orthogonal* (perpendicular *or* "⊥") vectors. We interpret this result as "the vectors are *uncorrelated*."

- Matrix transpose: **t**

```
> t(B)                           #B is 3x2  => t(B) is 2x3
     [,1] [,2] [,3]
[1,]    1    3    1
[2,]    1    2    0
```

- **X'X**   (a symmetric matrix)

```
> t(B)%*%B                       # command crossprod(B) is more efficient
     [,1] [,2]
[1,]   11    7
[2,]    7    5
```

- Determinant: **det**   (a symmetric matrix)

```
> det(t(B)%*%B)                  # Matrix has to be square. If det(A)=0 => A non-invertible
[1] 6
```

- $(X'X)^{-1}$: Inverse: **solve**

```
>solve(t(B)%*%B)                 #Matrix inside solve() has to be square
              [,1]      [,2]
[1,]  0.8333333 -1.166667
[2,] -1.1666667  1.833333
```

- Take the diagonal elements of a matrix A: **diag()**

```
 > diag(solve(t(B)%*%B))
[1] 0.8333333 1.833333
```

- Square root of (positive) elements of a matrix A: **sqrt()**

```
> sqrt(diag(solve(t(B)%*%B)))
        v3        v4
0.9128709 1.3540064
```

Linear Algebra: Examples

Example 1 – Linear DGP

There is a functional form relating a dependent variable, y, and k explanatory variables, **X**. The functional form is linear, but it depends on k unknown parameters, **β**. The relation between y and **X** is not exact. There is an error, **ε**. We have T observations of y and **X**.

- Then, the data is generated according to:

$$y_i = \Sigma_{j=1,..k}\, x_{k,i}\, \beta_k + \varepsilon_i \qquad\qquad i = 1, 2, ..., T.$$

Or using matrix notation:

$$\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where **y** & **ε** are (Tx1); **X** is (Txk); and **β** is (kx1).

- We will call this relation *data generating process* (DGP).

- The goal of econometrics is to estimate the unknown vector **β**. ¶

Example 2 – Linear System

Assume an economic model as system of linear equations with:

a_{ij} parameters, where i = 1,.., m rows, j = 1,.., n columns
x_i endogenous variables (n),
d_i exogenous variables and constants (m).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = d_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = d_2 \\ \ldots . \quad \ldots \quad \ldots \quad \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + amn\,x_n = d_m \end{cases}$$

- Using linear algebra, we have a system of linear equations: **Ax** = ***d***

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ \ldots \\ d_m \end{bmatrix}$$

where

A = (mxn) matrix of parameters
x = column vector of endogenous variables (nx1)
d = column vector of exogenous variables and constants (mx1). ¶

We want to solve for the solution of **Ax** = ***d***, ⇒ x*.

Theorem: Given A (*m*x*n*) invertible. Then, the equation Ax = *d* has one and only one solution for every *d* (*m*x1). That is, there is a unique x*.

$$\Rightarrow \mathbf{x}^* = \mathbf{A}^{-1}\, \boldsymbol{d}$$

Example: In practice, we avoid computing A^{-1}, we solve a system.

```
A <- matrix(c(1, 1, 5, 7, 9, 11, 10, 10, 14), ncol = 3) # check det(A) for singularity (det(A)=-72)
d <- c(2, 5, 2)
> solve(A,d)
[1] -0.7222222  1.5000000 -0.7777778
```

Linear Algebra: Linear Dependence and Rank

A set of vectors is *linearly dependent* if any one of them can be expressed as a linear combination of the remaining vectors; otherwise, it is linearly independent.

Formal definition: Linear independence (LI)

The set $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_k\}$ is called a *linearly independent* set of vectors iff

$$c_1\, \boldsymbol{u}_1 + ... + c_k\, \boldsymbol{u}_k = \mathbf{0} \qquad \Rightarrow \; c_1 = c_2 = ... = c_{k,} = 0.$$

Notes:

- Dependence prevents solving a system of equations (**A** is not invertible). More unknowns than independent equations.
- The number of linearly independent rows or columns in a matrix is the *rank* of a matrix (rank(**A**)).

Examples:

(1) $v_1' = [5 \quad 12]$

$v_2' = [10 \quad 24]$

$$\boldsymbol{A} = \begin{bmatrix} 5 & 10 \\ 12 & 24 \end{bmatrix} = \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}$$

$$2v_1' - v_2' = \mathbf{0} \qquad \Rightarrow rank(\boldsymbol{A}) = 1$$

(2) $v_1 = \begin{bmatrix} 2 \\ 7 \end{bmatrix}; v_2 = \begin{bmatrix} 1 \\ 8 \end{bmatrix}; v_3 = \begin{bmatrix} 4 \\ 5 \end{bmatrix};$

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 7 & 8 & 5 \end{bmatrix}$$

$$3{v_1}' - 2{v_2}' = [6 \quad 21] - [2 \quad 16]$$
$$= [4 \quad 5] = {v_3}'$$

$$3v_1' - 2v_2' - {v_3}' = \mathbf{0} \qquad \Rightarrow rank(\boldsymbol{A}) = 2. ¶$$

Least Squares Estimation with Linear Algebra

Let's assume a linear system with *k* independent variables and T observations. That is,

$y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_k x_{ki} + \varepsilon_i$,   i= 1, 2, ..., T

The whole system (for all *i*) is:

$$\begin{aligned}
\boldsymbol{y_1} &= \boldsymbol{\beta_1 x_{11} + \beta_2 x_{21} + ... + \beta_k x_{k1} + \varepsilon_1}\\
\boldsymbol{y_2} &= \boldsymbol{\beta_1 x_{12} + \beta_2 x_{22} + ... + \beta_k x_{k2} + \varepsilon_2}\\
\boldsymbol{....} & \quad \boldsymbol{.. ..} \qquad \boldsymbol{. ...} \qquad \boldsymbol{...}\\
\boldsymbol{y_T} &= \boldsymbol{\beta_1 x_{1T} + \beta_2 x_{2T} + ... + \beta_k x_{kT} + \varepsilon_2}
\end{aligned}$$

Using linear algebra we can rewrite the system as:

$$\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Vectors will be column vectors: **y**, **x**$_\mathbf{j}$, and **ε** are *T*x1 vectors:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_T \end{bmatrix} \qquad \Rightarrow \qquad \mathbf{y'} = [y_1\ y_2\ ....\ y_T]$$

$$\boldsymbol{x_j} = \begin{bmatrix} x_{j1} \\ \vdots \\ x_{jT} \end{bmatrix} \qquad \Rightarrow \qquad \mathbf{x_j'} = [x_{j1}\ x_{j2}\ ....\ x_{jT}]$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_T \end{bmatrix} \qquad \Rightarrow \qquad \boldsymbol{\varepsilon'} = [\varepsilon_1\ \varepsilon_2\ ....\ \varepsilon_T]$$

X is a *T*x*k* matrix.   ⇒ $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ ....\ \mathbf{x}_k]$

$$\Rightarrow \qquad X = \begin{bmatrix} \boldsymbol{x_{11}} & \boldsymbol{x_{21}} & \cdots & \boldsymbol{x_{k1}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x_{1T}} & \boldsymbol{x_{2T}} & \cdots & \boldsymbol{x_{kT}} \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

- The whole system (for all *i*) is:

$$\begin{aligned}
y_1 &= \beta_1 x_{11} + \beta_2 x_{21} + ... + \beta_k x_{k1} + \varepsilon_1\\
y_2 &= \beta_1 x_{12} + \beta_2 x_{22} + ... + \beta_k x_{k2} + \varepsilon_2\\
.... & \quad .. .. \qquad . ... \qquad ...\\
y_T &= \beta_1 x_{1T} + \beta_2 x_{2T} + ... + \beta_k x_{kT} + \varepsilon_2
\end{aligned}$$

- With the linear assumption: $f(\mathbf{X}, \theta) = \mathbf{X}\,\boldsymbol{\beta}$, we can write the objective function as:

$$S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

We want to minimize $S(x_i, \theta)$

.

First derivative w.r.t. β: $-2\,(\mathbf{y} - \mathbf{X}\,\boldsymbol{\beta})'\,\mathbf{X}$

F.o.c. (normal equations): $-2\,(\mathbf{y} - \mathbf{X}\mathbf{b})'\,\mathbf{X} = \mathbf{y}'\,\mathbf{X} - \mathbf{b}'\mathbf{X}'\,\mathbf{X} = \mathbf{0}$

Taking transpose $\Rightarrow \mathbf{X}'\mathbf{y} - (\mathbf{X}'\mathbf{X})\,\mathbf{b} = \mathbf{0} \ \ \Rightarrow (\mathbf{X}'\mathbf{X})\,\mathbf{b} = \mathbf{X}'\mathbf{y}$

Assuming (**X**′**X**) is non-singular –i.e., invertible-, we solve for **b**: $\Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y}$

Note: **b** is called the Ordinary Least Squares (OLS) estimator. (*Ordinary* = $f(\mathbf{X}, \theta)$ is linear.)

X is a Txk matrix. Its columns are the k Tx1 vectors $\mathbf{x_k}$. It is common to treat $\mathbf{x_1}$ as vector of ones:

$$x_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1T} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix} \quad \Rightarrow \quad x_1' = [1\ 1 \ldots.\ 1] = \iota'$$

This vector of ones represent the usual constant in the model.

Note: Recall the dot product: Post-multiplying a vector (1xT) $\boldsymbol{x_k}$ by $\boldsymbol{\iota}$ (or $\boldsymbol{\iota}'\,\boldsymbol{x_k}$) produces a scalar, the sum of all the elements of vector $\boldsymbol{x_k}$:

$$\boldsymbol{x_k}'\,\boldsymbol{\iota} = \boldsymbol{\iota}'\,\boldsymbol{x_k} = x_{k1} + x_{k2} + \ldots. + x_{kT} = \sum_j x_{kj}$$

Example: CAPM Model for IBM returns:

```
SFX_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",
head=TRUE, sep=",")

x_ibm <- SFX_da$IBM
x_Mkt_RF <- SFX_da$Mkt_RF
x_RF <- SFX_da$RF

T <- length(x_ibm)
lr_ibm <- log(x_ibm[-1]/x_ibm[-T])
Mkt_RF <- x_Mkt_RF[-1]/100
RF <- x_RF[-1]/100

ibm_x <- lr_ibm - RF
T <- length(ibm_x)
x0 <- matrix(1,T,1)
x <- cbind(x0, Mkt_RF)

>b <- solve(t(x)%*% x)%*% t(x)%*%y              # b = (X′X)^-1X′ y  (OLS regression)
> b
                    [,1]
              -0.005791039
Mkt_RF         0.895773564
```

Note: We got these coefficient before, using the lm() function. Use summary to print results:

```
fit_ibm <- lm(x_ibm~ Mkt_RF)
summary(fit_ibm). ¶
```

OLS – Assumptions

Typical OLS Assumptions

(**1**) DGP: $y_i = \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_k x_{ki} + \varepsilon_i$, i= 1, 2, ..., T

⇒ functional form known, but $\boldsymbol{\beta}$ is unknown.

(**2**) $E[\varepsilon_i] = 0$. ⇒ expected value of the errors is 0.

(**3**) Explanatory variables $X_1, X_2, \ldots, X_k$, are given (& non random)

⇒ no correlation with ε ($\text{Cov}(\varepsilon_i, X_j) = 0$.)

(**4**) The *k* explanatory variables are independent.

(**5**) $\text{Var}[\varepsilon_i] = E[\varepsilon_i^2] = \sigma^2 < \infty$ (homoscedasticity = same variance)

(**6**) $\text{Cov}(\varepsilon_i, \varepsilon_j) = E[\varepsilon_i\, \varepsilon_j] = 0$. (no serial/cross correlation)

- These are the assumptions behind the *classical linear regression model* (CLM).

Least Squares – Assumptions with Linear Algebra Notation

We can rewrite the assumptions, conditioning on **X**, which allows **X** to be a random variable (though, once we condition, **X** becomes a matrix of numbers). Using linear algebra:

(**A1**) DGP: $\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$
(**A3**) $\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\, \mathbf{I}_T$
(**A4**) **X** has full column rank –rank(**X**)=*k*–, where $T \geq k$.

- Assumption (**A1**) is called *correct specification*. We know how the data is generated. We call

$\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}$ the Data Generating Process (DGP).

Note: The errors, **ε**, are called *disturbances*. They are not something we add to $f(\mathbf{X}, \boldsymbol{\theta})$ because we don't know precisely $f(\mathbf{X}, \boldsymbol{\theta})$. No. The errors are part of the DGP.

- Assumption (**A2**) is called *regression*.

From Assumption (**A2**) we get:

(i) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$ ⇒ $E[\mathbf{y}|\mathbf{X}] = E[f(\mathbf{X}, \theta)|\mathbf{X}] + E[\boldsymbol{\varepsilon}|\mathbf{X}] = f(\mathbf{X}, \theta)$

That is, the observed **y** will equal $E[\mathbf{y}|\mathbf{X}]$ + random variation.

(ii) Using rules of expectations, we get two results:

(1) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$ ⇒ $E[\boldsymbol{\varepsilon}] = 0$

⇒ The conditional expectation = unconditional expectation

(2) $\text{Cov}(\boldsymbol{\varepsilon}, \mathbf{X}) = E[(\boldsymbol{\varepsilon} - 0)(\mathbf{X} - \boldsymbol{\mu}_X)] = E[\boldsymbol{\varepsilon}\mathbf{X} + \boldsymbol{\varepsilon}\, \boldsymbol{\mu}_X]$

$= E[\boldsymbol{\varepsilon}\mathbf{X}] + \boldsymbol{\mu}_X\, E[\boldsymbol{\varepsilon}] = E[\boldsymbol{\varepsilon}\mathbf{X}] = 0$

$\Rightarrow$ That is, $E[\boldsymbol{\varepsilon}\mathbf{X}] = 0 \quad \Rightarrow \boldsymbol{\varepsilon} \perp \mathbf{X}$.
There is no information about **ε** in **X** and viceversa.

• Assumption (**A3**) gives the model a constant variance for all errors and no relation between the errors at different measurements/times. That is, we have a diagonal variance-covariance matrix:

$$\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma^2 \end{bmatrix} = \sigma^2 \boldsymbol{I}_T \quad (kxk) \text{ matrix}$$

This assumption implies

(i) *homoscedasticity* $\Rightarrow E[\varepsilon_i^2|\mathbf{X}] = \sigma^2$ for all i.

(ii) *no serial/cross correlation* $\Rightarrow E[\varepsilon_i\, \varepsilon_j\, |\mathbf{X}] = 0$ for $i \neq j$.

It can be shown using the law of total variance that

$\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\mathbf{I}_T \quad \Rightarrow \text{Var}[\boldsymbol{\varepsilon}] = \sigma^2\mathbf{I}_T$

• From Assumption (**A4**) $\Rightarrow$ the *k* independent variables in **X** are linearly independent. Then, the *kxk* matrix **X′X** will also have full rank –i.e., rank(**X′X**) = *k*.

Thus, **X′X** is invertible. We will need this result to solve a system of equations given by the 1st-order conditions of Least Squares Estimation.

Note: To get asymptotic results we will need more assumptions about **X**.

• We assume a linear functional form for $f(x, \theta) = \mathbf{X}\,\boldsymbol{\beta}$:
(**A1**') DGP: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

CLM: OLS – Summary

Classical linear regression model (CLM) - Assumptions:

(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$
(**A3**) $\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\, \mathbf{I}_T$
(**A4**) **X** has full column rank –rank(**X**) = *k*, where $T \geq k$.

Objective function: $S(x_i, \boldsymbol{\theta}) = \Sigma_i\, \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\beta)'\,(\mathbf{y} - \mathbf{X}\beta)$
Normal equations $-2\,(\mathbf{y} - \mathbf{X}\mathbf{b})'\,\mathbf{X} = 0$
$\mathbf{y}'\,\mathbf{X} - \mathbf{b}'\,\mathbf{X}'\mathbf{X} = 0$
Solving for **b** $\Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y}$ (*k*x1) vector

OLS Estimation: Second Order Condition

$$\frac{\partial^2 e'e}{\partial b \partial b'} = 2\boldsymbol{X'X} = 2\begin{bmatrix} \Sigma_{i=1}^{n} x_{i1}^2 & \Sigma_{i=1}^{n} x_{i1}x_{i2} & \dots & \Sigma_{i=1}^{n} x_{i1}x_{iK} \\ \Sigma_{i=1}^{n} x_{i2}x_{i1} & \Sigma_{i=1}^{n} x_{i2}^2 & \dots & \Sigma_{i=1}^{n} x_{i2}x_{iK} \\ \dots & \dots & \dots & \dots \\ \Sigma_{i=1}^{n} x_{iK}x_{i1} & \Sigma_{i=1}^{n} x_{iK}x_{i2} & \dots & \Sigma_{i=1}^{n} x_{iK}^2 \end{bmatrix}$$

If there were a single **b**, we would require this to be positive, which it would be:

$$2\mathbf{x'x} = 2\sum_{i=1}^{n} x_i^2 > 0.$$

The matrix counterpart of a positive number is a positive definite (pd) matrix. We need **X′X** to be pd.

A square matrix (mxm) **A** "takes the sign" of the quadratic form, **z′A z**, where **z** is a mx1 vector. Then, **z′A z** is a scalar.

$$\boldsymbol{X'X} = \begin{bmatrix} \Sigma_{i=1}^{n} x_{i1}^2 & \Sigma_{i=1}^{n} x_{i1}x_{i2} & \dots & \Sigma_{i=1}^{n} x_{i1}x_{iK} \\ \Sigma_{i=1}^{n} x_{i2}x_{i1} & \Sigma_{i=1}^{n} x_{i2}^2 & \dots & \Sigma_{i=1}^{n} x_{i2}x_{iK} \\ \dots & \dots & \dots & \dots \\ \Sigma_{i=1}^{n} x_{iK}x_{i1} & \Sigma_{i=1}^{n} x_{iK}x_{i2} & \dots & \Sigma_{i=1}^{n} x_{iK}^2 \end{bmatrix}$$

Definition: Positive definite matrix
A matrix **A** is *positive definite* (pd) if **z′A z** >0 for *any* **z** (a *k*x1 vector).

For some matrices, it is easy to check. Let **A** = **X′X** (a *k*x*k* matrix).
Then, **z′A z** = **z′X′X z** = **v′v** = ¤*i*=*1*□*k*□□*v*□*i*□*2*□□ > 0.
⇒ **X′X** is pd ⇒ **b** is a min!

Technical note: In general, we need eigenvalues of **A** to check this. If all the eigenvalues are positive, then **A** is pd.

OLS Estimation – Properties of b

The OLS estimator of β in the CLM is

$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'}\,\mathbf{y}$ ⇒ **b** is a (linear) function of the data (y_i ,x_i).

$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'}\,\mathbf{y} = (\mathbf{X'X})^{-1}\,\mathbf{X'}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon}$

□ $\mathbf{b} - \boldsymbol{\beta} = (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon}$

Under the typical assumptions, we can establish properties for **b**.

1) $\text{E}[\mathbf{b}|\mathbf{X}] = \text{E}[\boldsymbol{\beta}|\mathbf{X}] + \text{E}[(\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon}|\mathbf{X}]$
$= \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'}\,\text{E}[\boldsymbol{\varepsilon}|\mathbf{X}] = \boldsymbol{\beta}$ (**b** is *unbiased.*)

2) $\text{Var}[\mathbf{b}|\mathbf{X}] = \text{E}[(\mathbf{b} - \boldsymbol{\beta})\,(\mathbf{b} - \boldsymbol{\beta})'|\mathbf{X}] = \text{E}[(\mathbf{X'X})^{-1}\,\mathbf{X'}\,\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon}'\,\mathbf{X}(\mathbf{X'X})^{-1}]$
$= (\mathbf{X'X})^{-1}\,\mathbf{X'}\,\text{E}[\boldsymbol{\varepsilon}$□$'|\mathbf{X}]\,\mathbf{X}(\mathbf{X'X})^{-1}$
$= (\mathbf{X'X})^{-1}\,\mathbf{X'}\,\{\sigma^2\,\mathbf{I}_T\}\,\mathbf{X}(\mathbf{X'X})^{-1} = \sigma^2\,(\mathbf{X'X})^{-1}\,\mathbf{X'X}\,(\mathbf{X'X})^{-1}$
$= \sigma^2\,(\mathbf{X'X})^{-1}$ (*k*x*k*) matrix

3) **Gauss-Markov Theorem**: **b** is BLUE (*Best Linear Unbiased Estimator*). No other linear & unbiased estimator has a lower variance.

Proof**:**

Let $\mathbf{b}^* = \mathbf{Cy}$ (linear in $\mathbf{y}$)

$E[\mathbf{b}^*|\mathbf{X}] = E[\mathbf{Cy}|\mathbf{X}] = E[\mathbf{C}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon})|\mathbf{X}] = \boldsymbol{\beta}$ (unbiased if $\mathbf{CX}=\mathbf{I}$)

$\text{Var}[\mathbf{b}^*|\mathbf{X}] = E[(\mathbf{b}^* - \boldsymbol{\beta})(\mathbf{b}^* - \boldsymbol{\beta})'|\mathbf{X}] = E[\mathbf{C}\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon}'\mathbf{C}\,'|\mathbf{X}] = \sigma^2\,\mathbf{CC}'$

Now, we relate $\text{Var}[\mathbf{b}|\mathbf{X}]$ to $\text{Var}[\mathbf{b}^*|\mathbf{X}]$.

Let $\mathbf{D} = \mathbf{C} - (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'$ (note $\mathbf{DX} = 0$)

Then,

$$\begin{aligned}\text{Var}[\mathbf{b}^*|\mathbf{X}] &= \sigma^2\,(\mathbf{D} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\,(\mathbf{D}' + \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1})\\ &= \sigma^2\,\mathbf{DD}' + \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1} = \text{Var}[\mathbf{b}|\mathbf{X}] + \sigma^2\,\mathbf{DD}'.\end{aligned}$$

Since $\mathbf{DD}'$ is positive definite $\Rightarrow \text{Var}[\mathbf{b}^*|\mathbf{X}] > \text{Var}[\mathbf{b}|\mathbf{X}]$ ▪

4) If we make an additional assumption:

(**A5**) $\boldsymbol{\varepsilon}|\mathbf{X} \sim$ *i.i.d.* $N(\mathbf{0}, \sigma^2\mathbf{I}_T)$

we can derive the distribution of **b**.

Since $\mathbf{b} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}$, we have that **b** is a linear combination of normal variables $\Rightarrow \mathbf{b}|\mathbf{X} \sim$ *i.i.d.* $N(\boldsymbol{\beta}, \sigma^2\,(\mathbf{X}'\,\mathbf{X})^{-1})$

then, $SD[\mathbf{b}|\mathbf{X}] = \text{sqrt}(\text{diagonal elements of } \sigma^2\,(\mathbf{X}'\,\mathbf{X})^{-1})$

Note: The marginal distribution of a multivariate normal distribution is also normal, then

$b_k|\mathbf{X} \sim N(\beta_k, v_k^2)$

$\text{Std Dev}\,[b_k|\mathbf{X}] = \text{sqrt}\{[\sigma^2(\mathbf{X}'\mathbf{X})^{-1}]_{kk}\} = v_k$

Remark: With (**A5**) we can do tests of hypothesis.

5) If (**A5**) is not assumed, we still can obtain a (limiting) distribution for **b**. Under additional assumptions –mainly, the matrix $\mathbf{X}'\mathbf{X}$ does not explode as T becomes large–, as $T \rightarrow \infty$

(i) $\mathbf{b} \overset{p}{\rightarrow} \boldsymbol{\beta}$ (**b** is consistent)

(ii) $\mathbf{b} \overset{a}{\rightarrow} N(\boldsymbol{\beta}, \sigma^2\,(\mathbf{X}'\,\mathbf{X})^{-1})$ (**b** is asymptotically normal)

- Properties (1)-(4) are called *finite* (or *small)* sample properties, they hold for every sample size.

- Properties (i) and (ii) in (5) are called *asymptotic* properties, they only hold when T is large (actually, as T tends to ∞). Property (ii) is very important: When the errors are not normally distributed we still can do testing about $\boldsymbol{\beta}$, but we rely on an “approximate distribution.”

OLS Estimation – Fitted Values and Residuals

OLS estimates **β** with **b.** Now, we define *fitted values* as:

$$\hat{\boldsymbol{y}} = \mathbf{X}\ \mathbf{b}$$

Now we define the estimated error, **e**:

$\mathbf{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$**e** represents the unexplained part of **y**, what the regression cannot explain. They are usually called *residuals.*

Note that **e** is uncorrelated (orthogonal) with **X** ⇒ **ε** ⊥ **X**

$\mathbf{e} = \mathbf{y} - \mathbf{Xb} \quad \Rightarrow \mathbf{X}'\mathbf{e} = \mathbf{X}'\,(\mathbf{y} - \mathbf{Xb}) = \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}\,(\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y} = \mathbf{0}$

Using **e**, we can define a measure of unexplained variation:

Residual Sum of Squares (RSS) = $\mathbf{e'e} = \sum_i e_i{}^2$

OLS Estimation – Var[b|X]

We use the variance to measure precision of estimates. For OLS:

$$\text{Var}[\mathbf{b}|\mathbf{X}] = \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1}$$

Example: One explanatory variable model.

(**A1**') DGP: $\mathbf{y} = \beta_1 + \beta_2\,\mathbf{x} + \boldsymbol{\varepsilon}$

$$\text{Var}[\mathbf{b}|\mathbf{X}] = \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1} = \sigma^2 \begin{bmatrix} \sum_i 1 & \sum_i 1x_i \\ \sum_i 1x_i & \sum_i x_i^2 \end{bmatrix}^{-1} = \sigma^2 \begin{bmatrix} T & T\bar{x} \\ T\bar{x} & \sum_i x_i^2 \end{bmatrix}^{-1}$$

$$= \sigma^2 \frac{1}{T(\sum_i x_i^2 - T\bar{x}^2)} \begin{bmatrix} \sum_i x_i^2 & -T\bar{x} \\ -T\bar{x} & T \end{bmatrix}$$

$$\text{Var}[b_1|\mathbf{X}] = \sigma^2\,\frac{\sum_i x_i^2}{T(\sum_i x_i^2 - T\bar{x}^2)} = \sigma^2\,\frac{\sum_i x_i^2/T}{\sum_i (x_i - \bar{x})^2}$$

$$\text{Var}[b_2|\mathbf{X}] = \sigma^2\,\frac{1}{(\sum_i x_i^2 - T\bar{x}^2)} = \sigma^2\,\frac{1}{\sum_i (x_i - \bar{x})^2}$$

$$\text{Covar}[b_1, b_2|\mathbf{X}] = \sigma^2\,\frac{-\bar{x}}{\sum_i (x_i - \bar{x})^2}.\ ¶$$

• In general, we do not know σ^2. It needs to be estimated. We estimate σ^2 using the residual sum of squares (RSS):

$$\text{RSS} = \sum_i e_i{}^2$$

The natural estimator of σ^2 is $\hat{\sigma}2$ = RSS/*T*. Given the LLN, this is a consistent estimator of σ^2. However, this not unbiased.

• The unbiased estimator is s^2

$$s^2 = \text{RSS}/(T\text{-}k) = \sum_i e_i{}^2\,/(T\text{-}k) = \boldsymbol{e}'\boldsymbol{e}/(T\text{-}k)$$

To get E[s^2], we use a property of RSS (& χ_v^2 distribution):

$$\text{E}[\mathbf{e}'\mathbf{e}/[(T-k)\,\sigma^2]|\mathbf{X}] = (T-k) \qquad \Rightarrow \text{E}[s^2|\mathbf{X}] = \sigma^2$$

<u>Note</u>: (*T-k*) is referred as a *degrees of freedom* correction.

• Then, the estimator of Var[**b**|**X**] = $s^2\,(\mathbf{X}'\mathbf{X})^{-1}$

This estimator gives us the *standard errors* (SE) of the individual coefficients. For example, for the b_k coefficient:

$$SE[b_k|\mathbf{X}] = \text{sqrt}[s^2(\mathbf{X}'\mathbf{X})^{-1}]_{kk} = s_{b,k}$$

OLS Estimation – Testing Only One Parameter

We are interested in testing a hypothesis about one parameter in our linear model: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

1. Set H_0 and H_1 (about only one parameter): H_0: $\beta_k = \beta_k^0$
H_1: $\beta_k \neq \beta_k^0$.

2. Appropriate T(*X*): *t-statistic*. To derive the distribution of the test under H_0, we will rely on assumption (**A5**) $\boldsymbol{\varepsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_T)$ (otherwise, results are only asymptotic).

Let b_k = OLS estimator of β_k
$SE[b_k|\mathbf{X}] = \text{sqrt}\{[s^2(\mathbf{X'X})^{-1}]_{kk}\} = s_{b,k}$

From assumption (**A5**), we know that
$b_k|\mathbf{X} \sim N(\beta_k, v_k^2)$ $\Rightarrow$ Under H_0: $b_k|\mathbf{X} \sim N(\beta_k^0, s_{b,k}^2)$.
$\Rightarrow$ Under H_0: $t_b = (b_k - \beta_k^0)/s_{b,k}|\mathbf{X} \sim t_{T-k}$.

- We measure distance in standard error units:

$$t_b = (b_k - \beta^0{}_k)/s_{b,k}$$

Note: t_b is an example of the *Wald* (normalized) *distance measure*. Most tests statistics in econometrics will use this measure.

3. Compute t_b, $\hat{t}$, using b_k, $\beta^0{}_k$, s, and $(\mathbf{X'X})^{-1}$. Get *p-value*($\hat{t}$).
4. Rule: Set an α level. If *p-value*($\hat{t}$) < α $\Rightarrow$ Reject H_0: $\beta_k = \beta_k^0$
Alternatively, if $|\hat{t}| > t_{T-k,\alpha/2}$ $\Rightarrow$ Reject H_0: $\beta_k = \beta_k^0$.

- Special case: H_0: $\beta_k = 0$
H_1: $\beta_k \neq 0$.

Then,

$$t_k = b_k/\text{sqrt}\{s^2(\mathbf{X}'\,\mathbf{X})_{kk}^{-1}\} = b_k/SE[b_k] \Rightarrow t_k \sim t_{T-k}.$$

This special case of t_k is called the *t-value* or *t-ratio* (also refer as the "t-stats"). That is, the t-value is the ratio of the estimated coefficient and its SE.

- The t-value is routinely reported in all regression packages. In the lm() function, it is reported in the third column of numbers.

- Usually, α = 5%, then if $|t_k| > 1.96 \approx 2$, we say the coefficient b_k is "*significant*."

Example: For the 3-Factor Fama-French Model (continuation) for IBM returns we want to test if the stock's beta is equal to one, that is, if IBM bears the same risk as the market:

```
SFX_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",head=TRUE,sep=",")
```

```
x_ibm <- SFX_da$IBM                         # Read IBM price data (Mkt_RF Factor)
x_Mkt_RF <- SFX_da$Mkt_RF                   # Read Factor data -Mkt_RF Factor (in %)
x_SMB <- SFX_da$SMB
x_HML <- SFX_da$HML
x_RF <- SFX_da$RF
T <- length(x_ibm)                          # Sample size
 lr_ibm <- log(x_ibm[-1]/x_ibm[-T])         #  Log returns for IBM (lost one observation)
 Mkt_RF <- x_Mkt_RF[-1]/100                  # Adjust size (take one observation out )
 SMB <- x_SMB[-1]/100
 HML <- x_HML[-1]/100
 RF <- x_RF[-1]/100
y <- ibm_x                                  # Define y (IBM excess returns)
x1 <- Mkt_RF                                # Regressor 1 (Mkt_RF)
x2 <- SMB                                   # Regressor 2 (SMB)
x3 <- HML                                   # Regressor 3 (HML)
T <- length(x1)                             # New sample size (Original – 1 observation)
x0 <- matrix(1,T,1)                         # Define vector of ones (the constant in X)
x <- cbind(x0,x1,x2,x3)                     # Matrix X
k <- ncol(x)                                # Number of regressors (=rank(X)=k)
b <- solve(t(x)%*% x)%*% t(x)%*%y           # b = (X′X)^-1X′ y  (OLS regression)
e <- y - x%*%b                              # regression residuals, e
n <- ncol(x)                                # number or regressors, k
RSS <- as.numeric(t(e)%*%e)                 # RSS
Sigma2 <- as.numeric(RSS/(T-k))                     # Estimated σ^2 = s^2
SE_reg <- sqrt(Sigma2)                      # Estimated σ – Regression stand error
Var_b <- Sigma2*solve(t(x)%*% x)            # Estimated Var[b|X] = s^2 (X′X)^-1
SE_b <- sqrt(diag(Var_b))                   # SE[b|X]
t_b <- b/SE_b                               # t-values
y_hat <- x%*%b                              # fitted values
```

```
> t(b)
                    Mkt_RF        SMB        HML
[1,] -0.005088944 0.9082989 -0.2124596 -0.1715002
> t(SE_b)
                    Mkt_RF        SMB        HML
[1,] 0.002487509 0.05672206 0.08411188 0.08468165
> t(t_b)
                     Mkt_RF       SMB       HML
[1,] -2.045799 16.01315 -2.525917 -2.025235    ⇒ all coefficients are significant (|t|>2).
```

- Q: Is the market beta (β_1) equal to 1? That is,

  H0: $\beta_1 = 1$ vs.

  H1: $\beta_1 \neq 1$

  ⇒ $t_k = (b_k - \beta_k^0)/\text{Est. SE}(b_k)$

  $t_1 = (0.9082989 - 1)/\ 0.05672206 = -1.616674$

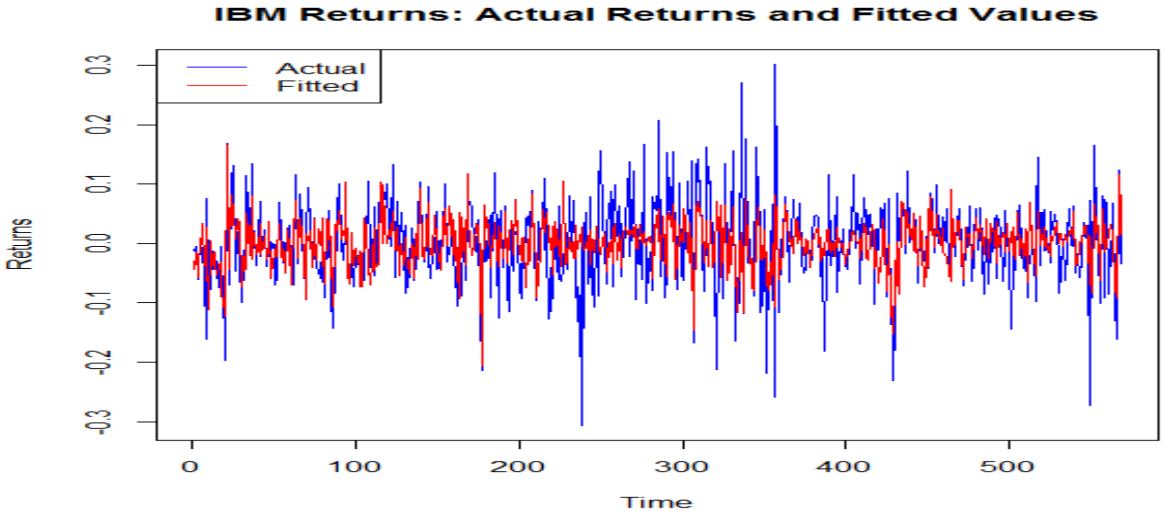
  ⇒ $|t_1| < 1.96$ ⇒ Cannot reject H_0 at 5% level.

Note: You should get the same numbers using *lm* (use summary(.) to print results) and extracting information from lm:

```
fit_ibm <- lm(ibm_x ~ Mkt_RF + SMB + HML)
summary(fit_ibm)
b_ibm <- fit_ibm$coefficients                # Extract from lm function OLS coefficients
SE_ibm <- sqrt(vcov(fit_ibm))                # SE from fit_ibm (also a kx1 vector)
t_beta1 <- (b_ibm[2] – 1)/SE_ibm[2]          # t-stat for H_0: Beta_1 - 1
p_val <- 1- pnorm(abs(t_beta1))              # pvalue for t_beta
p_val
```

```
> summary(fit_ibm)                           # print lm results
Call:
lm(formula = ibm_x ~ Mkt_RF + SMB + HML)

Residuals:
     Min        1Q            Median       3Q       Max
-0.307488 -0.030388 -0.000861  0.034350  0.252667

Coefficients:
                 Estimate        Std. Error t value Pr(>|t|)
(Intercept)      -0.005089       0.002488  -2.046   0.0412 *
Mkt_RF           0.908299        0.056722  16.013   <2e-16 ***
SMB              -0.212460       0.084112  -2.526   0.0118 *
HML              -0.171500       0.084682  -2.025   0.0433 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05848 on 565 degrees of freedom
Multiple R-squared:  0.3389,    Adjusted R-squared:  0.3354
F-statistic: 96.55 on 3 and 565 DF,  p-value: < 2.2e-16
```

```
plot(y, type = "l", col = "blue",                                  # Plot IBM returns
main = "IBM Returns: Actual Returns and Fitted Values", ylab = "Returns", xlab = "Time")
lines(y_hat, type = "l", col = "red")                              # Add fitted values to plot
legend("topleft",                                                         # Add legend to plot
     legend = c("Actual", "Fitted"), col = c("blue", "red"), lty = 1)
```



OLS Estimation – Linear Algebra Interpretation

- Disturbances and Residuals

In the population: $E[\mathbf{X}'\,\boldsymbol{\varepsilon}] = 0$.

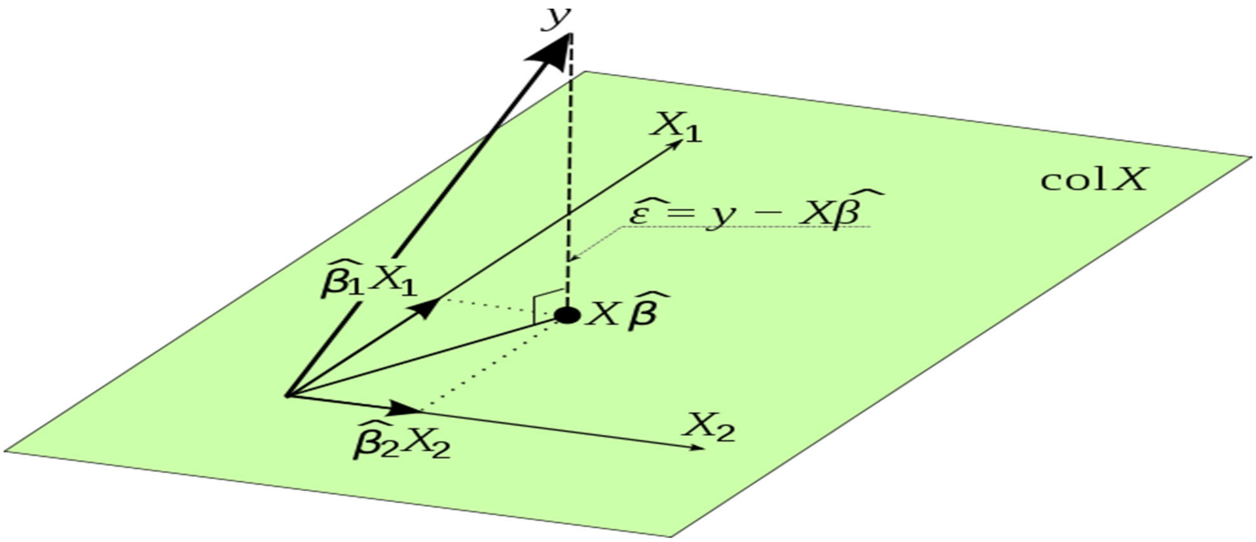
In the sample: $\mathbf{X}'\mathbf{e} = \mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{b}) = \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{y}$

$= 1/\mathrm{T}(\mathbf{X}'\,\mathbf{e}) = 0$.

- We have two ways to look at **y**:

y = E[**y**|X] + ε = Conditional mean + disturbance

y = **Xb** + e = Projection + residual



OLS Estimation – Important Matrices: M

Important Matrices

(1) "Residual maker"

$$\mathbf{M} = \mathbf{I}_T - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \qquad (T\text{x}T \text{ matrix})$$

$$\mathbf{My} = \mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{y} - \mathbf{Xb} = \mathbf{e} \quad \text{(residuals)}$$

$$\mathbf{MX} = (\mathbf{I}_T - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\,\mathbf{X} = \mathbf{0}$$

- **M** is symmetric – $\mathbf{M} = \mathbf{M}'$
- **M** is idempotent – $\mathbf{M}^*\mathbf{M} = \mathbf{M}$
- **M** is singular – $\mathbf{M}^{-1}$ does not exist. $\Rightarrow$ rank(**M**) = $T - k$

- Special case: $\mathbf{X} = \boldsymbol{\mathfrak{i}}$

$$\mathbf{M}^0 = \mathbf{I} - \mathfrak{i}(\mathfrak{i}'\,\mathfrak{i})^{-1}\,\mathfrak{i}' = \mathbf{I} - \mathfrak{i}\,\mathfrak{i}'/T \qquad \text{- since } \mathfrak{i}'\,\mathfrak{i} = T$$

$$\mathbf{M}^0\,\mathbf{y} = \mathbf{y} - \mathfrak{i}(\mathfrak{i}'\,\mathfrak{i})^{-1}\,\mathfrak{i}'\,\mathbf{y} = \mathbf{y} - \mathfrak{i}\,\bar{\mathbf{y}} \qquad \text{- since } \mathfrak{i}'\,\mathbf{y}/T = \bar{y}$$

$$\mathbf{M}^0\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \bar{y} = \begin{bmatrix} y_1 - \bar{y} \\ y_2 - \bar{y} \\ \vdots \\ y_T - \bar{y} \end{bmatrix}$$

Interpretation of M^0: De-meaning matrix.

(2) "Projection matrix"

$$\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \qquad (T\text{x}T \text{ matrix})$$

$$\mathbf{Py} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{Xb} = \hat{\mathbf{y}} \qquad \text{(fitted values)}$$

Py = Projection of **y** into the *column space* (dimension *k)* of **X**.

$$\mathbf{PX} = (\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\,\mathbf{X} = \mathbf{X}$$

PX = Projection of **X** into **X** = **X**.

$$\mathbf{PM} = \mathbf{MP} = \mathbf{0}$$

Note: $\mathbf{M} = \mathbf{I}_T - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \mathbf{I}_T - \mathbf{P}$

- **P** is symmetric – $\mathbf{P} = \mathbf{P}'$
- **P** is idempotent – $\mathbf{P}^*\mathbf{P} = \mathbf{P}$
- **P** is singular – $\mathbf{P}^{-1}$ does not exist. $\Rightarrow$ rank(**P**) = k

Results when X Contains a Constant Term

Let the first column of **X** be a column of ones. That is

$$\mathbf{X} = [\mathfrak{i}, \mathbf{x}_2, \ldots, \mathbf{x}_K]$$

• Then,

(1) Since $\mathbf{X}'\mathbf{e} = \mathbf{0}$ $\Rightarrow \mathbf{x}_1'\mathbf{e} = 0$ –the residuals sum to zero.

(2) Since $\mathbf{y} = \mathbf{Xb} + \mathbf{e}$ $\Rightarrow \mathfrak{i}'\mathbf{y} = \mathfrak{i}'\mathbf{Xb} + \mathfrak{i}'\,\mathbf{e} = \mathfrak{i}'\mathbf{Xb}$

$\Rightarrow \bar{y} = \bar{x}b$

That is, the regression line passes through the means.

Note: These results are only true if **X** contains a constant term!

Goodness of Fit of the Regression

After estimating the model (**A1**), we would like to judge the adequacy of the model. There are two ways to do this:

- Visual: Plots of fitted values and residuals, histograms of residuals.
- Numerical measures: R^2, adjusted R^2, AIC, BIC, etc.

Numerical measures. In general, they are simple and easy to compute. We call them *goodness-of-fit* measures. Most popular: R^2.

<u>Definition</u>: Variation

In the context of a model, we consider the *variation* of a variable as the movement of the variable, usually associated with movement of another variable.

Total variation = Total sum of squares (TSS) = $\sum_i (y_i - \bar{y})^2$

We want to decompose TSS in two parts: one explained by the regression and one unexplained by the regression.

$$\begin{aligned}\text{TSS} &= \textstyle\sum_i (y_i - \bar{y})^2 = \sum_i (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 \\ &= \textstyle\sum_i (y_i - \hat{y}_i)^2 + \sum_i (\hat{y}_i - \bar{y})^2 + 2\sum_i (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) \\ &= \textstyle\sum_i e_i{}^2 + \sum_i (\hat{y}_i - \bar{y})^2\end{aligned}$$

Since

$$\textstyle\sum_i (y_i - \hat{y}_i)(\hat{y}_i - \bar{y}) = \sum_i e_i(\hat{y}_i - \bar{y}) = 0$$

Or $\qquad$ TSS $\;$ = RSS + SSR

RSS: Residual Sum of Squares (also called SSE: SS of errors)
SSR: Regression Sum of Squares (also called ESS: *explained* SS)

Goodness of Fit of the Regression – Linear Algebra

Recall that we can use the de-meaning matrix $\mathbf{M^0}$ to write

$\mathbf{y} - \boldsymbol{\iota}\bar{y} = \mathbf{M^0 y}$ (Tx1 vector) $\quad$ where $\mathbf{M^0} = \mathbf{I} - \boldsymbol{\iota}(\boldsymbol{\iota}'\,\boldsymbol{\iota})^{-1}\,\boldsymbol{\iota}'$

Using linear algebra we also get the decomposition of TSS. Now,

TSS = $\sum_i (y_i - \bar{y})^2 = \mathbf{y'M^{0\prime}\,M^0y} = \mathbf{y'M^0M^0y} = \mathbf{y'M^0y}$.

We want to decompose the total variation of $\mathbf{y}$ (assume $\mathbf{X_1} = \boldsymbol{\iota}$ –a constant.)

$\mathbf{y} = \mathbf{Xb} + \mathbf{e}$,

then,

$$\begin{aligned}\mathbf{M^0y} &= \mathbf{M^0Xb} + \mathbf{M^0e} = \mathbf{M^0Xb} + \mathbf{e} && \text{(deviations from means)} \\ \mathbf{y'M^0y} &= \mathbf{b'(X'\,M^0)(M^0X)b} + \mathbf{e'e} && \text{(sum of squared deviations from means)} \\ &= \mathbf{b'X'\,M^0Xb} + \mathbf{e'e.} && (\mathbf{M^0} \text{ is idempotent \& } \mathbf{e'\,M^0X} = \mathbf{0}) \\ \text{TSS} &= \text{SSR} + \text{RSS}\end{aligned}$$

A Goodness of Fit Measure

We want to have a measure that describes the fit of a regression. Simplest measure: the standard error of the regression (SER)

SER = sqrt{RSS/(T-*k)}* ⇒ SER depends on units. Not good!

- R-squared (R^2)

$$1 = \text{SSR/TSS} + \text{RSS/TSS}$$

$$R^2 = \text{SSR/TSS} = \text{Regression variation/Total variation}$$

$$R^2 = \mathbf{b}'\mathbf{X}'\mathbf{M}^0\mathbf{X}\mathbf{b}/\mathbf{y}'\mathbf{M}^0\mathbf{y} = 1 - \mathbf{e}'\mathbf{e}/\mathbf{y}'\mathbf{M}^0\mathbf{y}$$

$$= (\hat{\mathbf{y}} - \acute{\imath}\,\bar{y})'\,(\hat{\mathbf{y}} - \acute{\imath}\,\bar{y})/(\mathbf{y} - \acute{\imath}\bar{y})'\,(\mathbf{y} - \acute{\imath}\bar{y}) = [\,\hat{\mathbf{y}}'\hat{\mathbf{y}} - T\,\bar{y}^2]/[\mathbf{y}'\mathbf{y} - T\bar{y}^2]$$

As introduced here, R^2 lies between 0 and 1 (& it is independent of units of measurement!). It measures how much of total variation (TSS) is explained by regression (SSR): the higher R^2, the better.

<u>Note</u>: R^2 is bounded by zero and one only if:

(a) There is a constant term in **X** –we need **e' M**$^{\mathbf{0}}$**X**=**0**!

(b) The line is computed by linear least squares.

- Main problem with R^2: Adding regressors

It can be shown that R^2 never falls when regressors (say **z**) are added to the regression. This occurs because RSS decreases with more information.

<u>Problem</u>: Judging a model based on R^2 tends to over-fitting.

- Comparing Regressions

- Make sure the denominator in R^2 is the same - i.e., same left hand side variable. For example, linear vs. loglinear. Loglinear will almost always appear to fit better because taking logs reduces variation.

- Linear Transformation of data does not change R^2.

- Based on **X**, $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

Suppose we work with $\mathbf{X}^* = c\mathbf{X}$, instead (*c* is a constant).

$$\mathbf{P}^*\mathbf{y} = \mathbf{X}^*\mathbf{b}^* = c\mathbf{X}\,(c\mathbf{X}'\,c\mathbf{X})^{-1}c\mathbf{X}'\mathbf{y}$$

$$= c\mathbf{X}\,(c^2\,\mathbf{X}'\mathbf{X})^{-1}\,c\mathbf{X}'\mathbf{y}$$

$$= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{X}'\mathbf{y} = \mathbf{P}\mathbf{y}$$

⇒ same fit, same residuals, same R^2!

Adjusted R-squared

R^2 is modified with a penalty for number of parameters: *Adjusted-*R^2

$$\overline{R}^2 = 1 - \frac{(T-1)}{(T-k)}(1 - R^2) = 1 - \frac{(T-1)}{(T-k)}\frac{\text{RSS}}{\text{TSS}} = 1 - \frac{s^2}{\text{TSS}/(T-1)}$$

⇒ maximizing $\overline{R}^2$ <=> minimizing $[\text{RSS}/(T-k)] = s^2$

- *Degrees of freedom* –i.e., $(T-k)$– adjustment assumes something about “unbiasedness.”

$\overline{R}^2$ includes a penalty for variables that do not add much fit. Can fall when a variable is added to the equation.

It will rise when a variable, say **z**, is added to the regression if and only if the t-ratio on **z** is larger than one in absolute value.

Theil (1957) shows that, under certain assumptions (an important one: the true model is being considered), if we consider two linear models

M_1: $\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}_1$
M_2: $\mathbf{y} = \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}_2$

and choose the model with smaller s^2 (or, larger Adjusted R^2), we will select the true model, M_1, on average.

In this sense, we say that "maximizing Adjusted R^2" is an *unbiased* model-selection criterion.

Other Goodness of Fit Measures

There are other goodness-of-fit measures that also incorporate penalties for number of parameters (degrees of freedom). We minimize these measures.

Information Criteria

- *Amemiya*: $[\mathbf{e}'\mathbf{e}/(T-k)] * (1 + k/T) = s^2 * (1 + k/T)$

- *Akaike Information Criterion* (AIC)

$\text{AIC} = -2/T(\ln L - k)$ *L*: *Likelihood*

⇒ if normality $\text{AIC} = \ln(\mathbf{e}'\mathbf{e}/T) + (2/T)\,k$ (+constants)

- *Bayes-Schwarz Information Criterion* (BIC)

$\text{BIC} = -(2/T \ln L - [\ln(T)/T]\,k)$

⇒ if normality $\text{AIC} = \ln(\mathbf{e}'\mathbf{e}/T) + [\ln(T)/T]\,k$ (+constants)

Example: 3 Factor F-F Model (continuation) for IBM returns:

```
b <- solve(t(x)%*% x)%*% t(x)%*%y                    # b = (X'X)^-1 X' y  (OLS regression)
e <- y - x%*%b                                       # regression residuals, e
RSS <- as.numeric(t(e)%*%e)                          # RSS
R2 <- 1 - as.numeric(RSS)/as.numeric(t(y)%*%y)                # R-squared
Adj_R2 <- 1 - (T-1)/(T-k)*(1-R2)                     # Adjusted R-squared
AIC <- log(RSS/T) + 2*k/T                            # AIC under N(.,.) –i.e.,  under (A5)
```

```
> R2
[1] 0.338985              ⇒ The 3 factors explain 34% of the variability of IBM returns.
> Adj_R2
[1] 0.3354752
> AIC
[1] -5.671036. ¶
```

Maximum Likelihood Estimation

Idea: Assume a particular distribution with unknown parameters. Maximum likelihood (ML) estimation chooses the set of parameters that maximize the likelihood of drawing a particular sample.

Consider a sample (X_1, ... , X_n) which is drawn from a pdf $f(\boldsymbol{X}|\theta)$ where θ are parameters. If the X_i's are independent with pdf $f(X_i|\theta)$ the joint probability of the whole sample is:

$$L(X \mid \theta) = f(X_1 \ldots X_n \mid \theta) = \prod_{i=1}^{n} f(X_i \mid \theta)$$

The function $L(X|\ \theta)$ –also written as L(X; θ)– is called the *likelihood function*. This function can be maximized with respect to θ to produce maximum likelihood estimates: $\hat{\theta}_{MLE}$.

It is often convenient to work with the *Log of the likelihood* function. That is,

$\ln L(X|\theta) = \Sigma_i \ln f(X_i|\ \theta).$

The ML estimation approach is very general. We need a model and a pdf for the errors to apply ML. Now, if the model is not correctly specified, the estimates are sensitive to misspecification.

A lot of applications in finance and economics: Time series, volatility (GARCH and stochastic volatility) models, factor models of the term structure, switching models, option pricing, logistic models (mergers and acquisitions, default, etc.), trading models, etc.

In general, we rely on numerical optimization to get MLEs.

Maximum Likelihood Estimation: Properties

ML estimators (MLE) have very appealing properties:

(1) *Efficiency.* Under general conditions, they achieve lowest possible variance for an estimator.

(2) *Consistency*. As the sample size increases, the MLE converges to the population parameter it is estimating:

$$\hat{\theta}_{MLE} \overset{p}{\to} \theta$$

(3) *Asymptotic Normality:* As the sample size increases, the distribution of the MLE converges to the normal distribution.

$$\hat{\theta}_{MLE} \xrightarrow{a} N(\theta, [n\mathbf{I}(\theta)]^{-1})$$

where $\mathbf{I}(\theta)$ is the information matrix:

$$E\left[\left(\frac{\partial \log L}{\partial \theta}\right)\left(\frac{\partial \log L}{\partial \theta}\right)^{\mathrm{T}}\right] = I(\theta) \ \ (k\text{x}k \text{ matrix})$$

(4) *Invariance.* The ML estimate is invariant under functional transformations. That is, if $\hat{\theta}_{MLE}$ is the MLE of θ and if $g(\theta)$ is a function of θ, then $g(\hat{\theta}_{MLE})$ is the MLE of $g(\theta)$.

(5) *Sufficiency*. If a single sufficient statistic exists for θ, the MLE of θ must be a function of it. That is, $\hat{\theta}_{MLE}$ depends on the sample observations only through the value of a sufficient statistic.

Maximum Likelihood Estimation: Example

Let the sample be $\boldsymbol{X}$ = {5, 6, 7, 8, 9, 10} drawn from a Normal(μ, 1). The probability of each of these points based on the unknown mean, μ, can be written as:

$$f(5|\mu) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(5-\mu)^2}{2}\right]$$
$$f(6|\mu) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(6-\mu)^2}{2}\right]$$
$$\vdots$$
$$f(10|\mu) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(10-\mu)^2}{2}\right]$$

Assume that the sample is independent. Then, the joint pdf function can be written as:

$$L(X|\mu) = f(5|\mu) * f(6|\mu) * \ldots * f(10|\mu)$$
$$= \frac{1}{(2\pi)^{6/2}} exp\left[-\frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \cdots - \frac{(10-\mu)^2}{2}\right]$$

The value of m that maximizes the likelihood function of the sample can then be defined by

$$\max_{\mu} L(X|\mu).$$

It easier, however, to maximize the Log likelihood, ln L(X|μ). That is,

$$\max_{\mu} ln\big(L(X|\mu)\big) = -\, {}^{6}\!/_{2}\, ln(2\pi) + \left[-\frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \cdots - \frac{(10-\mu)^2}{2}\right]$$

1st-derivative $\Rightarrow \frac{\partial}{\partial\mu}\left[K - \frac{(5-\mu)^2}{2} - \frac{(6-\mu)^2}{2} - \cdots - \frac{(10-\mu)^2}{2}\right]$

f.o.c. $\Rightarrow$ $(5-\hat{\mu}_{MLE}) + (6-\hat{\mu}_{MLE}) + \cdots + (10-\hat{\mu}_{MLE}) = 0$

Then, the first order conditions:

$$(5-\hat{\mu}_{MLE}) + (6-\hat{\mu}_{MLE}) + \cdots + (10-\hat{\mu}_{MLE}) = 0$$

Solving for $\hat{\mu}_{MLE}$:

$$\hat{\mu}_{MLE} = \frac{5+6+7+8+9+10}{6} = 7.5 = \bar{x}$$

That is, the MLE estimator $\hat{\mu}_{MLE}$ is equal to the sample mean. This is good for the sample mean: MLE has very good properties!

Example: For $\boldsymbol{X}$ = {5, 6, 7, 8, 9, 10} ~ N(μ, 1), code to get $\hat{\mu}_{MLE}$.

```
mu <- 0                                    # assumed mean (initial value)
```

```
x_6 <- c(5, 6, 7, 8, 9, 10)             # data
dnorm(5, mu, sd=1)                      # probability of observing a 5, assuming a N(mu=0, sd=1)
dnorm(x_6)                              # probability of observing each element in x_6
l_f <- prod(dnorm(x_6))                 # Likelihood function
log(l_f)                                        # Log likelihood function
sum(log(dnorm(x_6)))                            # Alternative calculation of Log likelihood function

# Step 1 - Create Likelihood function
likelihood_n <- function(mu){                   # Create a prob function with mu as an argument
sum(log(dnorm(x_6, mu, sd=1)))
}
> likelihood_n(mu)                      # print likelihood
[1] -183.0136

negative_likelihood_n <- function(mu){   # R uses a minimization algorithm, change sign
sum(log(dnorm(x_6, mu, sd=1))) * (-1)
}
> negative_likelihood_n(mu)
[1] 183.0136


# Step 2 - Maximize (or Minimize negative Likelihood function)
results_n <- nlm(negative_likelihood_n, mu, stepmax=2)    # nlm minimizes the function
> results_n                             # Show nlm results
$minimum
[1] 14.26363                    <=  The minimized value of function (-14.26363 is the max)
$estimate
[1] 7.5                         <= The MLE for μ (=$\hat{\mu}_{MLE}$).
$gradient
[1] -4.736952e-12               <= Should be very close to zero if we're at a minimum

mu_max <- results_n$estimate                    # Extract estimates
> mu_max                                # Should be equal to mean
[1] 7.5
> likelihood_n(mu_max)                  # Check max value at mu_max
[1] -14.26363. ¶
```

• Now, we generalize the previous example to an *i.i.d.* sample ***X*** = {X_1, X_2,..., X_T} drawn from a Normal(μ, σ^2). Then, the joint pdf function is:

$$L(X|\mu) = \frac{1}{(2\pi\sigma^2)^{-T/2}} \exp\left[-\frac{(x_1-\mu)^2}{2\sigma^2} - \frac{(x_2-\mu)^2}{2\sigma^2} - \cdots - \frac{(x_T-\mu)^2}{2\sigma^2}\right]$$

Then, taking logs, we have:

$$L = -\frac{T}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{T}(X_i-\mu)^2 = -\frac{T}{2}\ln 2\,\pi - \frac{T}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\boldsymbol{X}-\mu)'(\boldsymbol{X}-\mu)$$

Then, taking logs, we have:

$$L = -\frac{T}{2}\ln(2\pi\sigma^2) - \frac{\sum_{i=1}^{T}(x_i-\mu)^2}{2\sigma^2} = -\frac{T}{2}\ln 2\,\pi - \frac{T}{2}\ln\sigma^2 - \frac{(X-\mu)'(X-\mu)}{2\sigma^2}$$

Taking first derivatives:

$$\frac{\partial L}{\partial\mu} = -\frac{\sum_{i=1}^{T}2(x_i-\mu)(-1)}{2\sigma^2} = \frac{\sum_{i=1}^{T}(x_i-\mu)}{\sigma^2}$$

$$\frac{\partial \ln L}{\partial\sigma^2} = -\frac{T}{2\sigma^2} + \frac{\sum_{i=1}^{T}(x_i-\mu)^2}{2\sigma^4}$$

We can write the first derivatives as a vector, the *gradient,* whose length is the number of unknown parameters in the likelihood –i.e., size of θ. In this case, a 2x2 vector:

$$\frac{\partial L}{\partial\theta} = \begin{bmatrix}\frac{\partial L}{\partial\mu} \\ \frac{\partial \ln L}{\partial\sigma^2}\end{bmatrix} = \begin{bmatrix}\frac{\sum_{i=1}^{T}(x_i-\mu)}{\sigma^2} \\ -\frac{T}{2\sigma^2} + \frac{\sum_{i=1}^{T}(x_i-\mu)^2}{2\sigma^4}\end{bmatrix}$$

In the case of a log likelihood function, the vector of first derivatives is called the *Score.*

When we set the Score equal to **0**, we have the set of first order conditions (f.o.c.).Then, we have the f.o.c. and jointly solve for the ML estimators:

$$(1)\ \ \frac{\partial L}{\partial\mu} = \frac{1}{\hat{\sigma}^2_{MLE}}\sum_{i=1}^{T}(X_i-\hat{\mu}_{MLE}) = 0 \quad \Rightarrow \hat{\mu}_{MLE} = \frac{1}{T}\sum_{i=1}^{T}X_i = \bar{X}$$

Note: The MLE of μ is the sample mean. Therefore, it is unbiased.

$$(2)\ \ \frac{\partial \ln L}{\partial\sigma^2} = -\frac{T}{2\hat{\sigma}^2_{MLE}} + \frac{1}{2\hat{\sigma}^4_{MLE}}\sum_{i=1}^{T}(X_i-\hat{\mu}_{MLE})^2 = 0$$

$$\Rightarrow \hat{\sigma}^2_{MLE} = \frac{1}{T}\sum_{i=1}^{T}(X_i-\bar{X})^2$$

Note: The MLE of σ^2 is not s^2. Therefore, it is biased! But, it is consistent.

Example: Using $\boldsymbol{X}$ = {5, 6, 7, 8, 9, 10}, now drawn from a Normal(μ, σ^2).

$$\hat{\mu}_{MLE} = \bar{X} = 7.5$$

$$\hat{\sigma}^2_{MLE} = \frac{\sum_{i=1}^{6}(x_i-7.5)^2}{6} = \frac{17.5}{6} = \mathbf{2.916667}$$

$$\hat{\sigma}_{MLE} = \text{sqrt}(\mathbf{2.916667}) = \mathbf{1.707825}$$

Note 1: $s^{2} = \frac{17.5}{(6-1)} = \mathbf{3.5}$

Note 2: The computation of MLE for the mean parameter $\hat{\mu}_{MLE}$ is independent of the computation of the MLE for the variance $\hat{\sigma}^2_{MLE}$. ¶

- To obtain the variance of $\hat{\theta}_{MLE} = [\hat{\mu}_{MLE}, \hat{\sigma}^2_{MLE}]$ we invert the information matrix for the whole sample $\mathbf{I}(\theta|X)$. Recall,

$$\hat{\theta}_{MLE} \xrightarrow{a} N(\theta,\ \mathbf{I}(\theta|X)^{-1})$$

where $\mathbf{I}(\theta|X)$ is the *Information matrix* for the whole sample. It is generally calculated as:

$$E\left[-\left(\frac{\partial^2 \log L(\theta|X)}{\partial\theta\partial\theta\prime}\right)\right] = I(\theta|X), \qquad (kxk \text{ matrix})$$

where the matrix of second derivatives is the Hessian matrix, **H**:

$$\frac{\partial^2 \log L(\theta|X)}{\partial\theta\partial\theta\prime} = \mathbf{H}$$

In practice, we use numerical optimization packages (say, *nlm* in R), which minimize a function. Then, we *minimize* the *negative* log $L(\theta|X)$ and, thus, to get Var[$\hat{\theta}_{MLE}$] we do not need to multiply **H** by **(-1)**.

Example: For $\boldsymbol{X}$ = {5, 6, 7, 8, 9, 10} ~ N(μ, σ^2), code to get MLEs.

```
mu <- 0                                  # assumed mean (initial value)
sig <- 1                                         # assumed sd (initial value)
x_6 <- c(5, 6, 7, 8, 9, 10)
# Step 1 - Create Likelihood function
likelihood_lf <- function(x){            # Create a prob function with mu & sig as arguments
mu <- x[1]
sig <- x[2]
sum(log(dnorm(x_6, mu, sd=sig)))
}
negative_likelihood_lf <- function(x){   # R uses a minimization algorithm, change sign
mu <- x[1]
sig <- x[2]
sum(log(dnorm(x_6, mu, sd=sig))) * (-1)
}
negative_likelihood_lf(x)

# Step 2 - Maximize Log Likelihood function (or Minimize negative Likelihood function)
results_lf <- nlm(negative_likelihood_lf, x, stepmax=4)     # nlm minimizes the function
> results_lf                                     # displays nlm results
$minimum
[1] 11.72496                             <=  Minimized value of function
$estimate
[1] 7.500000 1.707825                            <= MLEs for μ & σ^2 (=μ̂_MLE & σ̂²_MLE)
$gradient
```

```
[1] -1.846772e-07 -7.986103e-08      <= ≈  0 if we're at a minimum
$code
[1] 1                                <= = 1 if we program  stopped at a minimum
$iterations
[1] 34                               <= Number of iterations

par_max <- results_lf$estimate                # Extract estimates
> par_max                                     # Should be equal to sample mean
[1] 7.500000 1.707825
> likelihood_lf(par_max)                      # Check max value of likelihood function
[1] -11.72496

# Step 3 – Standard Errors (by inverting the Hessian)
results_lf <- nlm(negative_likelihood_lf, x, stepmax=4 , hessian=TRUE)
par_hess <- results_lf$hessian                # Extract Hessian
> par_hess                                    # Show Hessian
             [,1]          [,2]
[1,]  2.0571428731 -0.0009030531
[2,] -0.0009030531  4.1122292411
cov_lf <- solve(coeff_hess)                   # invert Hessian to get cov(MLEs)
> cov_lf                                      # Show covariance matrix
             [,1]         [,2]
[1,] 0.4861111542 0.0001067509
[2,] 0.0001067509 0.2431771280
se_lf <- sqrt(diag(cov_lf))                   # Compute standard errors of MLEs
>se_lf
[1] 0.6972167 0.4931299

# t-tests
> par_max[1]/se_lf[1]                         # t-ratio for mu
[1] 10.75706
par_max[2]/se_lf[2]                           # t-ratio for sigma2
[1] 3.463236. ¶
```

Maximum Likelihood Estimation: Linear Model Example

We will work the previous example with matrix notation. Suppose we assume:

$$y_i = \boldsymbol{x}_i{}'\boldsymbol{\beta} + \varepsilon_i, \qquad \varepsilon_i \sim N(0,\sigma^2)$$

or $$y = X\beta + \varepsilon, \qquad \varepsilon \sim N(0,\sigma^2 I_T)$$

where $\boldsymbol{x}_i$ is a kx1 vector of exogenous numbers and $\boldsymbol{\beta}$ is a kx1 vector of unknown parameters. Then, the joint likelihood function becomes:

$$L = \prod_{i=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right) = (2\pi\sigma^2)^{-T/2} \prod_{i=1}^{T} exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right)$$

Taking logs, we have the log likelihood function:

$$\ln L = -\frac{T}{2}\ln 2\,\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{T}\varepsilon_i^2 = -\frac{T}{2}\ln 2\,\pi\sigma^2 - \frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X\beta})'(\boldsymbol{y}-\boldsymbol{X\beta})$$

The joint likelihood function becomes:

$$\ln L = -\frac{T}{2}\ln 2\,\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{T}\varepsilon_i^2 =$$
$$= -\frac{T}{2}\ln(2\pi) - \frac{T}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)'(y - X\beta)$$

We take first derivatives of the log likelihood w.r.t. $\boldsymbol{\beta}$ and σ^2:

$$\frac{\partial \ln L}{\partial \beta} = -\frac{1}{2}\sum_{i=1}^{T} 2\varepsilon_i\, x_i'/\sigma^2 = -\frac{1}{\sigma^2}\boldsymbol{X'}\varepsilon$$
$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{T}{2\sigma^2} - (-\frac{1}{2\sigma^4})\sum_{i=1}^{T}\varepsilon_i^2 = (\frac{1}{2\sigma^2})[\frac{\varepsilon'\varepsilon}{\sigma^2} - T]$$

Using the f.o.c., we jointly estimate $\boldsymbol{\beta}$ and σ^2:

$$\frac{\partial \ln L}{\partial \beta} = -\frac{1}{\sigma^2}\boldsymbol{X'\varepsilon} = \frac{1}{\sigma^2}\boldsymbol{X}'(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}_{MLE}) = 0 \quad \Rightarrow \widehat{\boldsymbol{\beta}}_{MLE} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$
$$\frac{\partial \ln L}{\partial \sigma^2} = (\frac{1}{2\hat{\sigma}^2_{MLE}})[\frac{\boldsymbol{e'e}}{\hat{\sigma}^2_{MLE}} - T] = 0 \quad \Rightarrow \hat{\sigma}^2_{MLE} = \frac{\boldsymbol{e'e}}{T} = \sum_{i=1}^{T}\frac{(y_i - \boldsymbol{X}_i\widehat{\boldsymbol{\beta}}_{MLE})^2}{T}$$

Under (**A5**) –i.e., normality for the errors–, we have that $\widehat{\boldsymbol{\beta}}_{MLE} = \boldsymbol{b}$.
This is a good result for OLS **b**. ML estimators have very good properties: Efficiency, consistency, asymptotic normality and invariance.

$\hat{\sigma}^2_{MLE}$ is biased, but given that it is an ML estimator, it is efficient, consistent and asymptotically normally distributed.

Example: We estimate the 3 F-F factor model for IBM.

```
SFX_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/Stocks_FX_1973.csv",head=TRUE,sep=",")
x_ibm <- SFX_da$IBM
x_Mkt_RF<- SFX_da$Mkt_RF
x_SMB <- SFX_da$SMB
```

```
x_HML <- SFX_da$HML
x_RF <- SFX_da$RF
T <- length(x_ibm)
lr_ibm <- log(x_pfe[-1]/x_pfe[-T])
x0 <- matrix(1,T-1,1)
 Mkt_RF <- x_Mkt_RF[-1]/100
 SMB <- x_SMB[-1]/100
 HML <- x_HML[-1]/100
 RF <- x_RF[-1]/100
ibm_x <- lr_ibm - RF
X <- cbind(x0, Mkt_RF, SMB, HML)

# Step 1 - Negative Likelihood function
likelihood_lf <- function(theta, y ,X) {
N <- nrow(X)
k <- ncol(X)
beta <- theta[1:k]
sigma2 <- theta[k+1]^2
e <- y - X%*%beta
logl <- -.5*N*log(2*pi)-.5*N*log(sigma2)- ((t(e)%*%e)/(2*sigma2))
return(-logl)
}
theta <- c(0,1,1,1,.1)                              # initial values
likelihood_lf(theta,ibm_x,X)
          [,1]
[1,] -599.0825

# Step 2 - Maximize (or Minimize negative Likelihood function)
results_lf <- nlm(likelihood_lf, theta, hessian=TRUE, y=ibm_x, X=X)      # nlm minimizes l_f
par_max <- results_lf$estimate                               # Extract estimates
> par_max                                                    # Should be equal to OLS results
[1] -0.0005907974  0.8676052091 -0.6815947799 -0.2284249895  0.0557422421
> likelihood_lf(par_max,ibm_x,X)                             # Check max value of likelihood function
 [,1]
[1,] -835.3316
# Compare with OLS results
```

```
> fit_capm <- lm(ibm_x ~ Mkt_RF + SMB + HML)
> summary(fit_capm)
Coefficients:
              Estimate     Std. Error   t value   Pr(>|t|)
(Intercept)   -0.0005903   0.0023793    -0.248    0.80416
Mkt_RF        0.8676042    0.0542554    15.991    < 2e-16 ***
SMB           -0.6815950   0.0804542    -8.472    < 2e-16 ***
HML           -0.2284263   0.0809992    -2.820    0.00497 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Maximum Likelihood Estimation: Score and Information Matrix

Definition: Score (or efficient score)

$$S(X;\theta) = \frac{\delta\, log(\, L(X|\theta))}{\delta\theta} = \sum_{i=1}^{n} \frac{\delta\, log(\, f(x_i|\theta))}{\delta\theta}$$

$S(X;\theta)$ is called the *score* of the sample. It is the vector of partial derivatives (the gradient), with respect to the parameter θ. If we have k parameters, the score will have a kx1 dimension.

Definition: *Fisher information* for a single parameter for observation i:

$$E\left[\left(\frac{\partial \log(f(x_i|\theta))}{\partial\theta}\right)^2\right] = I(\theta)$$

$I(\theta)$ is sometimes just called *information*. It measures the shape of the log $f(X|\theta)$.

The concept of information can be generalized for the k-parameter case. In this case, for the whole sample:

$$E\left[\left(\frac{\partial\, log\, L}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial\, log\, L}{\partial\boldsymbol{\theta}}\right)^T\right] = \boldsymbol{I}(\theta)$$

This is kxk matrix.

If L is twice differentiable with respect to θ, and under certain regularity conditions, then the information may also be written as

$$E\left[\left(\frac{\partial \log L}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial \log L}{\partial\boldsymbol{\theta}}\right)^{\mathrm{T}}\right] = E\left[-\left(\frac{\delta^2 \log(L(X \mid \boldsymbol{\theta}\,))}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right)\right] = \mathbf{I}(\boldsymbol{\theta})$$

$\mathbf{I}(\theta)$ is called the *information matrix* (negative Hessian). It measures the shape of the likelihood function.

• The inverse of the information matrix for the whole sample is the Variance of $\hat{\theta}_{MLE}$. That is,

$$\text{Var}(\hat{\theta}_{MLE}) = I(\theta)^{-1}$$

Sometimes, the notation for the information matrix for the whole sample is $I(\theta|X)$.

Remark: In practice, we use the inverse of the Hessian, evaluated at $\hat{\theta}_{MLE}$, as the estimator of the variance. R calculates the Hessian in all optimization packages (for example, *nlm* or *optim*). In the previous example, we extracted the Hessian from the *nlm* function with

```
        coeff_hess <- results_lf$hessian            # Extract Hessian
```

Example: We assume:

$$y_i = X_i\boldsymbol{\beta} + \varepsilon_i,\ \varepsilon_i \sim N(0,\sigma^2)$$

or

$$y = X\beta + \varepsilon, \quad \varepsilon \sim N(0,\sigma^2 I_T)$$

Taking logs, we have the log likelihood function:

$$\ln L = -\frac{T}{2}\ln 2\pi\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{T}\varepsilon_i^2 = -\frac{T}{2}\ln 2\pi - \frac{T}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{X\beta})'(\boldsymbol{y}-\boldsymbol{X\beta})$$

The score function is –first derivatives of log L w.r.t. **θ** = (**β**, σ^2):

$$\frac{\partial \ln L}{\partial \beta} = -\frac{1}{2}\sum_{i=1}^{T} 2\varepsilon_i\, \boldsymbol{x}_i'/\sigma^2 = -\frac{1}{\sigma^2}\boldsymbol{X'\varepsilon}$$

$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{T}{2\sigma^2} - (-\frac{1}{2\sigma^4})\sum_{i=1}^{T}\varepsilon_i^2 = (\frac{1}{2\sigma^2})[\frac{\boldsymbol{\varepsilon'\varepsilon}}{\sigma^2} - T]$$

Then, we take second derivatives to calculate I(θ):

$$\frac{\partial^2 \ln L}{\partial\beta\partial\beta'} = -\sum_{i=1}^{T}\boldsymbol{x}_i\boldsymbol{x}_i{}'/\sigma^2 = -\frac{1}{\sigma^2}X'X \qquad \text{a } k\text{x}k \text{ matrix.}$$

$$\frac{\partial^2 \ln L}{\partial\beta\partial\sigma^2{}'} = -\frac{1}{\sigma^4}\sum_{i=1}^{T}\varepsilon_i x_i{}'$$

$$\frac{\partial^2 \ln L}{\partial\sigma^2\partial\sigma^2{}'} = -\frac{1}{2\sigma^4}[\frac{\varepsilon'\varepsilon}{\sigma^2} - T] + (\frac{1}{2\sigma^2})(-\frac{\varepsilon'\varepsilon}{\sigma^4}) = -\frac{1}{2\sigma^4}[2\frac{\varepsilon'\varepsilon}{\sigma^2} - T] \qquad \text{a scalar.}$$

Using linear algebra notation:

$$I(\theta) = E[-\frac{\partial \ln L}{\partial\theta\partial\theta'}] = \begin{bmatrix} (\frac{1}{\sigma^2}X'X) & 0 \\ 0 & \frac{T}{2\sigma^4} \end{bmatrix} \qquad \text{a } (k+1)\text{x}(k+1) \text{ matrix.} \ ¶$$

Example: We continue the previous IBM example, computing MLE SEs for linear model

```
# Step 3 - Compute S.E. by inverting Hessian
par_hess <- results_lf$hessian                    # Extract Hessian
> par_hess                                        # Show Hessian matrix
           [,1]         [,2]          [,3]          [,4]          [,5]
[1,] 183123.2131 1034.3403801 300.5280632 452.9161743 -3.243494e+02
[2,]   1034.3404  390.1995683  71.3131499 -55.6126338 -6.913297e-01
[3,]    300.5281   71.3131499 170.5839168 -26.9486009 -3.023956e-01
[4,]    452.9162  -55.6126338 -26.9486009 165.2938181 -2.928687e-01
[5,]   -324.3494   -0.6913297  -0.3023956  -0.2928687  3.629895e+05
```

```
cov_lf <- solve(par_hess)                    # invert Hessian to get covariance
se_lf <- sqrt(diag(cov_lf))                  # Compute standard errors (compare with OLS SE)
> se_lf
[1] 0.002370939 0.054063912 0.080170161 0.080713227 0.001659791

# We can do testing. For example, H0: Beta = 1.
> (par_max[2] -1)/se_lf[2]                   # t-test for H0: beta=1
[1] -2.448857
```

• Summary: OLS vs MLE

	OLS		**Bootstrap**	
	Coefficients	S.E.	Coefficients	S.E.
Intercept	**-0.00509**	**0.00238**	**-0.00509**	**0.00237**
Mkt_RF	**0.86761**	**0.05425**	**0.86761**	**0.05406**
SMB	**-0.68159**	**0.08045**	**-0.68159**	**0.08017**
HML	**-0.22842**	**0.08100**	**-0.22842**	**0.08071**

Data Problems

"*If the data were perfect, collected from well-designed randomized experiments, there would hardly be room for a separate field of econometrics.*" Zvi Griliches (1986, **Handbook of Econometrics**)

Three important data problems:
(1) Missing Data – very common, especially in cross sections and long panels.
(2) Outliers - unusually high/low observations.
(3) Multicollinearity - there is perfect or high correlation in the explanatory variables.

• In general, data problems are exogenous to the researcher. We cannot change the data or collect more data.

Missing Data

General Setup

We have an indicator variable, s_i. If $s_i = 1$, we observe Y_i, and if $s_i = 0$ we do not observe Y_i.

Note: We always observe the missing data indicator s_i.

Suppose we are interested in the population mean $\theta = \text{E}[Y_i]$.

With a lot of information -large T-, we can learn $p = \text{E}[s_i]$ and $\mu_1 = \text{E}[Y_i| s_i = 1]$, but nothing about $\mu_0 = \text{E}[Y_i|s_i = 0]$.

We can write: $\theta = p \cdot \mu_1 + (1 - p) \cdot \mu_0$.

Problem: Since even in large samples we learn nothing about μ_0, it follows that without additional information/assumptions there is no limit on the range of possible values for θ.

Now, suppose the variable of interest is binary: $Y_i \in \{0, 1\}$. We also have an explanatory variable of Y_i, say W_i.

Then, the natural (not data-informed) lower and upper bounds for μ_0 are 0 and 1 respectively.

This implies bounds on θ:

$$\theta \in [\theta_{\text{LB}}, \theta_{\text{UB}}] = [p \cdot \mu_1, p \cdot \mu_1 + (1 - p)].$$

These bounds are *sharp*, in the sense that without additional information we cannot improve on them.

If from the variable W_i we can infer something about the missing values, these bounds can be improved.

Missing Data – CLM

Now, suppose we have the CLM: $y_i = \mathbf{x}_i \beta + \varepsilon_i$

We use the selection indicator, s_i , where $s_i = 1$ if we can use observation i. Then,

$$\mathbf{b} = \beta + (\Sigma_i\, s_i\, \mathbf{x}_i'\mathbf{x}_i /T)^{-1} (\Sigma_i\, s_i\, \mathbf{x}_i'\varepsilon_i /T)$$

• For unbiased (and consistent) results, we need $\text{E}[s_i\, \mathbf{x}_i'\varepsilon_i\,] = 0$,
implied by $\text{E}[\varepsilon_i|s_i\, \mathbf{x}_{ii}] = 0$ (*)

In general, we find that when $s_i = h(\mathbf{x}_i)$, that is, the selection is a function of $\mathbf{x}_i$, we have an inconsistent OLS **b**. This situation is called *selection bias*.

Example of Selection Bias: Determinants of Hedging.
A researcher only observes companies that hedge. Estimating the determinants of hedging from this population will bias the results! ¶

If missing observations are randomly (exogenously) “selected,” it is likely safe to ignore problem. Rubin (1976) calls this assumption “*missing completely at random*” (or MCAR).

In general, MCAR is rare. In general, it is more common to see “*missing at random,”* where missing data depends on observables (say, education, sex) but one item for individual *i* is NA (Not Available).

If in the regression we “control” for the observables that influence missing data (not easy), it is OK to delete the whole observation for *i*.

Missing Data – Usual Solutions

Otherwise, we can:
a. Fill in the blanks –i.e., *impute* values to the missing data- with averages, interpolations, or values derived from a model.

b. Use (inverse) probability weighted estimation. Here, we inflate or “over-weight” unrepresented subjects or observations.

c. Heckman selection correction. We build a model for the selection function, $h(x_i)$.

Outliers

Many definitions: Atypical observations, extreme values, conditional unusual values, observations outside the expected relation, etc.

In general, we call an *outlier* an observation that is numerically different from the data. But, is this observation a “mistake,” say a result of measurement error, or part of the (heavy-tailed) distribution?

In the case of normally distributed data, roughly 1 in 370 data points will deviate from the mean by 3*SD. Suppose T=1,000 and we see **9** data points deviating from the mean by more than 3*SD indicates outliers... Which of the **9** observations can be classified as an outlier?

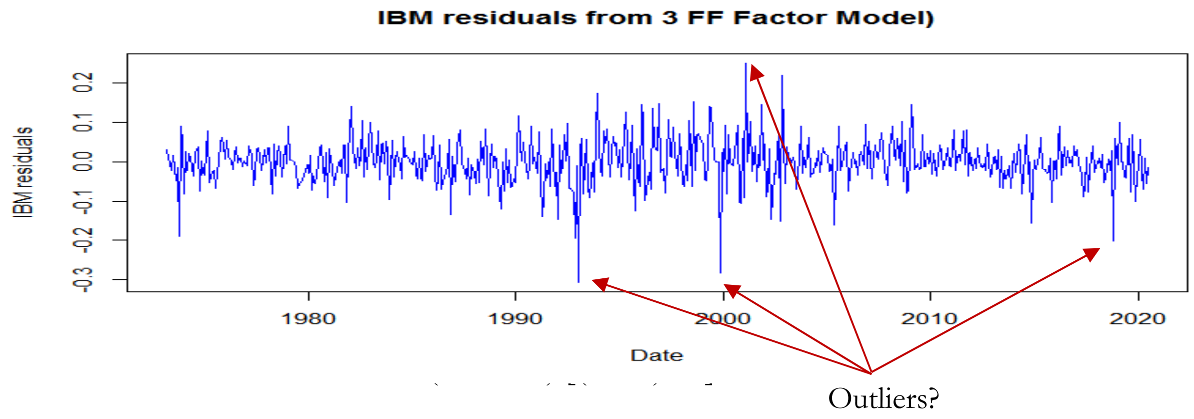
Problem with outliers: They can affect estimates. For example, with small data sets, one big outlier can seriously affect OLS estimates.

Outliers: Identification

• Informal identification method:
- *Eyeball*: Look at the observations away from a scatter plot.

Example: Plot residuals for the 3 FF factor model for IBM returns

```
x_resid <- residuals(fit_capm)
plot(x_resid, typ ="l", col="blue", main ="IBM Residuals from 3 FF Factor Model",
xlab="Date", ylab="IBM residuals")
```

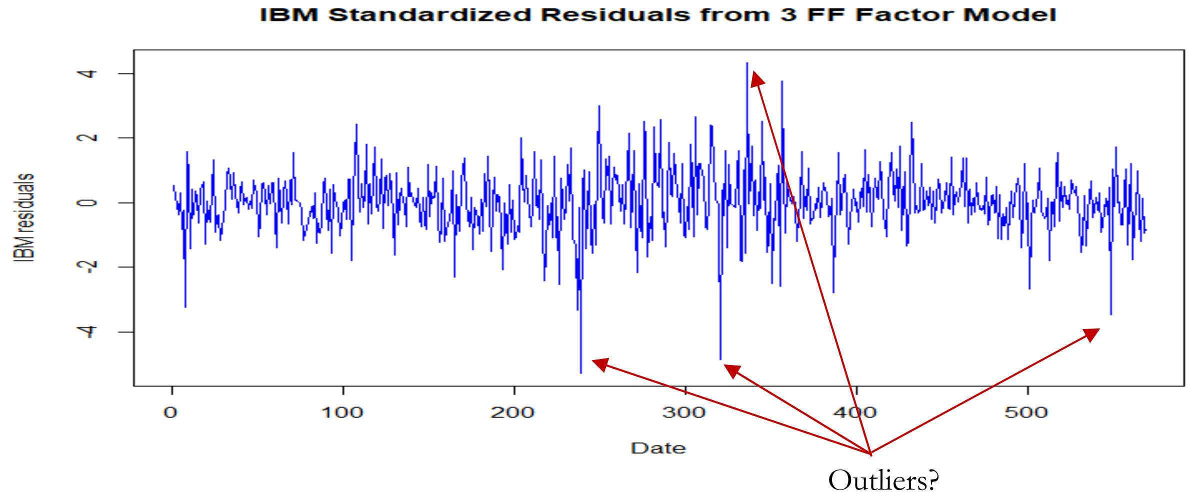


• Formal identifications methods:

- *Standardized residuals*, $e_i/SD(e_i)$: Check for errors that are 2*SD (or more) away from the expected value.

Example: Plot standardized residuals for IBM residuals

x_stand_resid <- x_resid/sd(x_resid) # standardized residuals
plot(x_stand_resid, typ ="l", col="blue", main ="IBM Standardized Residuals from 3 FF Factor Model", xlab="Date", ylab="IBM residuals")



- *Leverage statistic*: It measures the difference of an independent data point from its mean. High leverage observations can be potential outliers. Leverage is measured by the diagonal values of the **P** matrix:

$$h_t = 1/T + (x_t - \bar{x})/[(T-1)s_x^2].$$

But, an observation can have high leverage, but no *influence*.

- *Influence statistic: Dif beta*. It measures how much an observation influences a parameter estimate, say b_j. Dif beta is calculated by removing an observation, say *i*, recalculating b_j, say $b_j(-i)$, taking the difference in betas and standardizing it. Then,

$$\textit{Dif beta}_{j(-i)} = [b_j - b_j(-i)]/\mathrm{SE}[b_j].$$

- *Influence statistic: Distance D (as in Cook's D*). It measures the effect of deleting an observation on the fitted values, say $\hat{y}_j$.

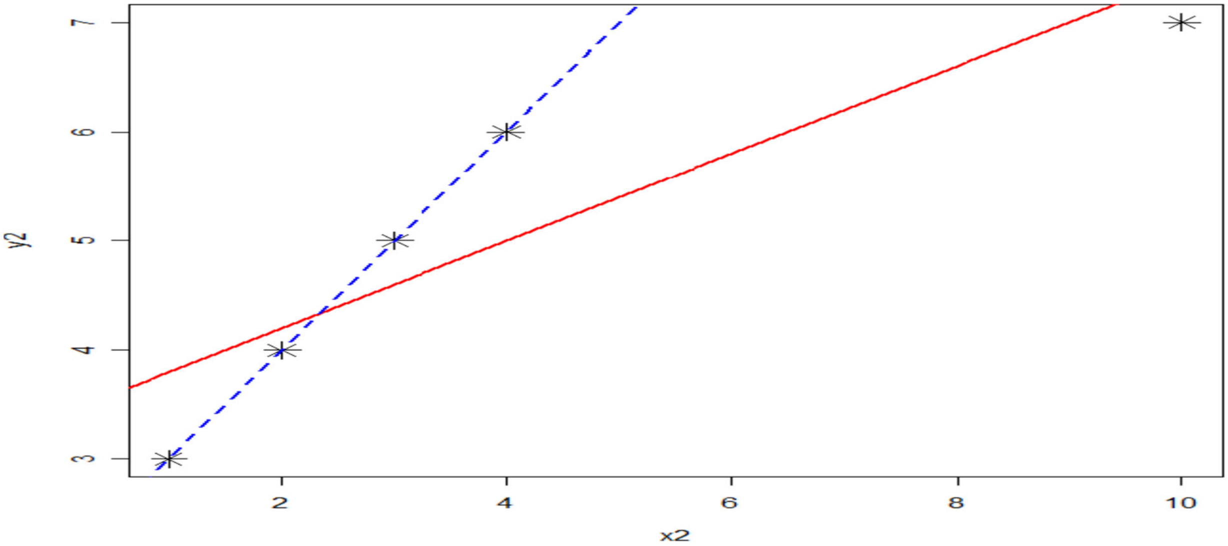
$$D_j = \Sigma_j\, [\hat{y}_j - \hat{y}_j(-i)]/[k * \mathrm{MSE}],$$

where *k* is the number of parameters in the model and MSE is mean square error of the regression model.

The identification statistics are usually compared to some ad-hoc cut-off values. For example, for Cook's D, if $D_i > 4/T \Rightarrow$ observation *i* is considered a (potential) highly influential point.

The analysis can also be carried out for groups of observations. In this case, we would be looking for blocks of highly influential observations.

Outlier Identification: Leverage & Influence



Deleting the observation in the upper right corner has a clear effect on the regression line. This observation has *leverage* and *influence.*

Outliers: Summary of Rules of Thumb

General rules of thumb (ad-hoc thresholds) used to identify outliers:

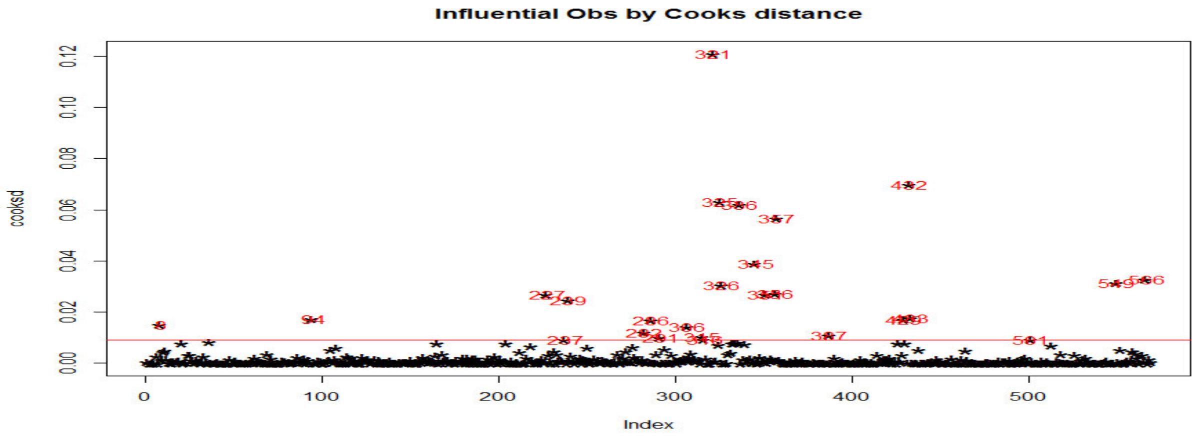
Measure	**Value**
abs(stand resid)	> 2
leverage	> (2*k*+2)/*T*

abs(*Dif Beta*) > 2/sqrt(*T*)
Cook's D > 4/*T*

In general, if we have 5% or less observations exceeding the ad-hoc thresholds, we tend to think that the data is OK.

Example: Cook's D for IBM returns using the 3 FF Factor Model

```
y <- ibm_x
x <- cbind(x0, Mkt_RF, SMB, HML)
dat_xy <- data.frame(y, x)
fit_capm <- lm(y ~ x - 1)
cooksd <- cooks.distance(fit_capm)
# plot cook's distance
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")
# add cutoff line
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # add cutoff line
# add labels
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd, na.rm=T),
names(cooksd),""), col="red")  # add labels

 # influential row numbers
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T))])
 # print first 10 influential observations.
head(dat_xy[influential, ], n=10L)
```



```
> # print first 10 influential observations.
>head(dat_xy[influential, ],n=10L)

            V1       Mkt_RF   SMB     HML
8   -0.16095068  1  0.0475  0.0294  0.0219
94   0.01266444  1  0.0959 -0.0345 -0.0835
227 -0.04237227  1  0.1084 -0.0224 -0.0403
237 -0.19083575  1  0.0102  0.0205 -0.0210
239 -0.30648638  1  0.0153  0.0164  0.0252
282  0.07787100  1 -0.0597 -0.0383  0.0445
```

```
286  0.20734626  1  0.0625 -0.0389  0.0117
291  0.15218986  1  0.0404 -0.0565 -0.0006
306  0.13928315  1 -0.0246 -0.0512 -0.0096
315  0.16196934  1  0.0433  0.0400  0.0253
```

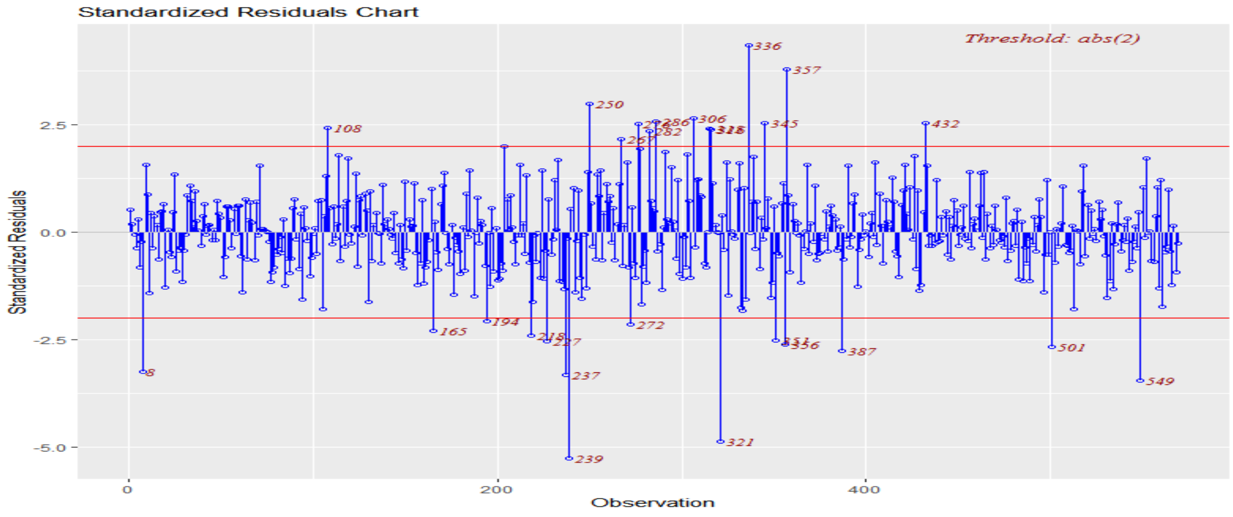
Note: There are easier ways to plot Cook’s D and identify the suspect outliers. The package *olsrr* can be used for this purpose too. ¶

Example: Different tools to check for outliers for residual in the FF model for IBM returns. We will use the package *olsrr* --install it with **install.packages()**.
install.packages(“olsrr”)

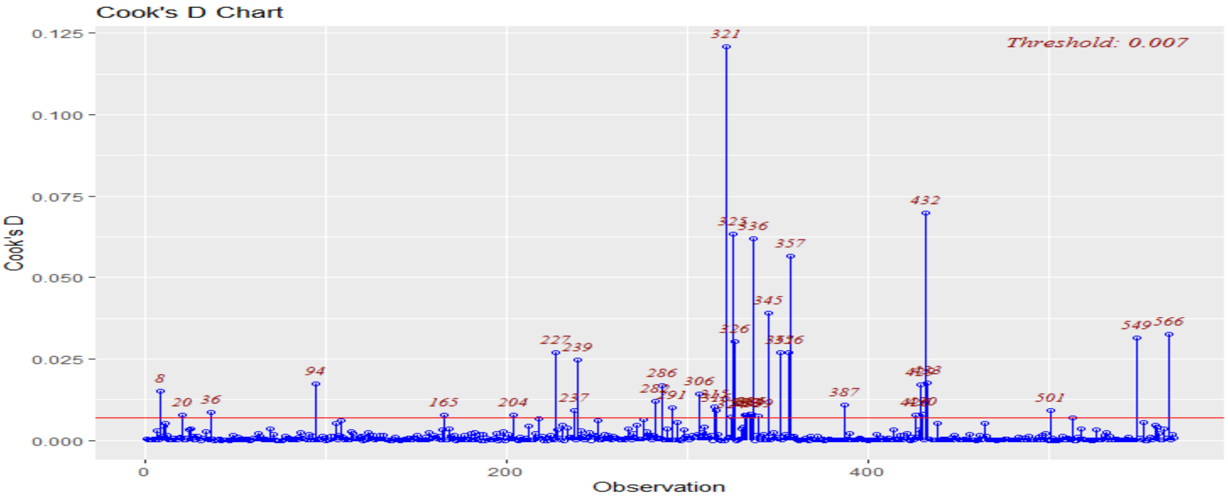
```
library(olsrr)                              # need to install package olsrr
x_resid <- residuals(fit_capm)
x_stand_resid <- x_resid/sd(x_resid)        # standardized residuals
sum(x_stand_resid > 2)                      # Rule of thumb count (5% count is OK)
x_lev <- ols_leverage(fit_capm)                     # leverage residuals
sum(x_lev > (2*k+2)/T)                      # Rule of thumb count (5% count is OK)
sum(cooksd > 4/T)                           # Rule of thumb count (5% count is OK)
ols_plot_resid_stand(fit_capm)              # Plot standardized residuals
ols_plot_cooksd_bar(fit_capm)               # Plot Cook’s D measure
ols_plot_dfbetas(fit_capm)                  # Plot Difference in betas
```

```
> sum(x_stand_resid > 2)
[1] 13                    # 5%? = 13/569 = 0.0228
> sum(x_lev > (2*k+2)/T)
[1] 32                    # 5%? = 32/569 = 0.0562
> sum(cooksd > 4/T)
[1] 38                    # 5%? = 38/569 = 0.0668
```

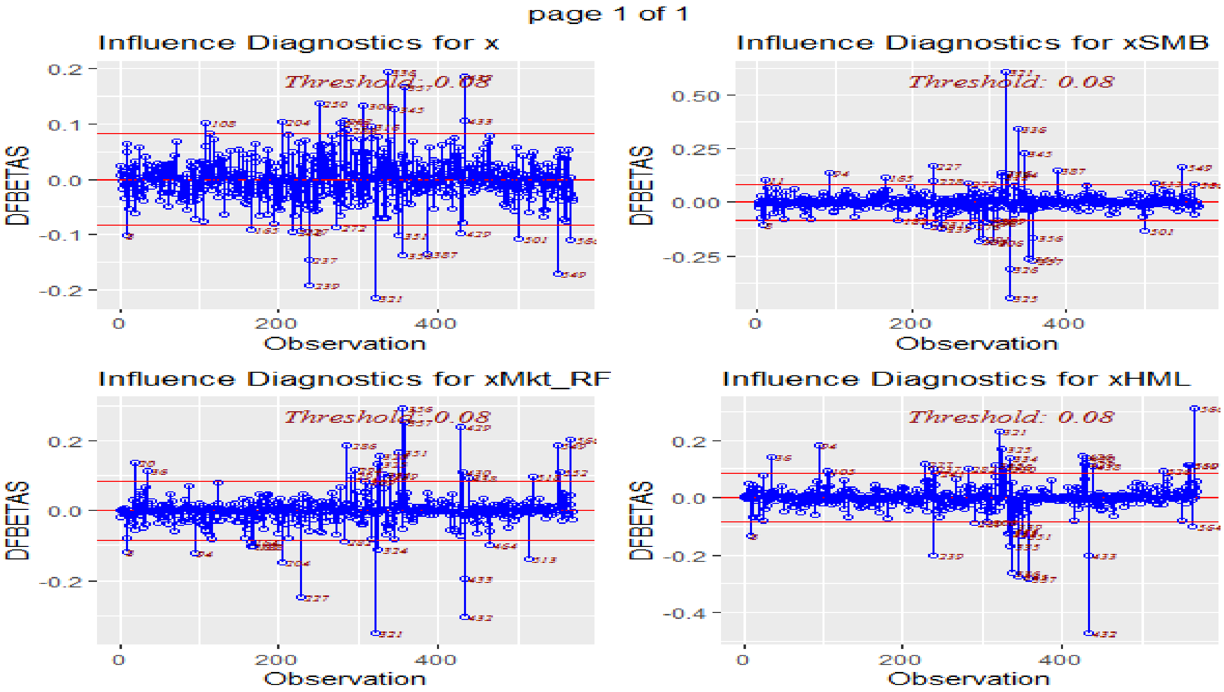
```
>ols_plot_resid_stand(fit_capm)                     # Plot Standardize residuals
```



```
>ols_plot_cooksd_bar(fit_capm)              # Plot Cook’s D measure
```



>ols_plot_dfbetas(**fit_capm**)



Outliers: What to Do?

Typical solutions:

- Use a non-linear formulation or apply a transformation (log, square root, etc.) to the data.
- Remove suspected observations. (Sometimes, there are theoretical reasons to remove suspect observations. Typical procedure in finance: remove public utilities or financial firms from the analysis.)
- Winsorization of the data (cut an α% of the highest and lowest observations of the sample).
- Use dummy variables.
- Use LAD (quantile) regressions, which are less sensitive to outliers.
- Weight observations by size of residuals or variance (robust estimation).

General rule: Present results with or without outliers.

Multicollinearity

The **X** matrix is *singular* (perfect collinearity) or *near singular (multicollinearity*).

- *Perfect collinearity*

Not much we can do. OLS will not work ⇒ **X'X** cannot be inverted. The model needs to be reformulated.

- *Multicollinearity*.

OLS will work. β is still unbiased. The problem is in $(\mathbf{X'X})^{-1}$; that is, in the Var[**b**|**X**]. Let's see the effect on the variance of particular coefficient, b_k.

Recall the estimated Var[b_k|**X**] is the *k*th diagonal element of $\sigma^2(\mathbf{X'X})^{-1}$.

Let define $R^2_{k.}$ as the R^2 in the regression of $\mathbf{x}_k$ on the other regressors, $\mathbf{X}_{(-k)}$. Then, we can show the estimated Var[b_k|**X**] is

$$\text{Var}[b_k|\mathbf{X}] = \frac{s^2}{[(1-R^2_{k.})\sum_{i=1}^{n}(x_{ik}-\bar{x}_k)^2]}.$$

⇒ the higher $R^2_{k.}$ –i.e., the fit between $\mathbf{x}_k$ and the rest of the regressors–, the higher Var[b_k|**X**].

Multicollinearity: Signs

Signs of Multicollinearity:

- Small changes in **X** produce wild swings in **b**.
- High R^2, but **b** has low t-stats –i.e., high standard errors
- "Wrong signs" or difficult to believe magnitudes in **b**.

There is no *cure* for collinearity. Estimating something else is not helpful (transforming regressors, principal components, etc.).

There are "measures" of multicollinearity, such as the

- *K# = Condition number* = max(singular value)/min(singular value)
- *Variance inflation factor* = $\text{VIF}_k = 1/(1 - R^2_{k.})$.

Rule of thumb for Condition number: If *K#* > 30 such matrix cannot be inverted reliably. Thus, **X** shows severe multicollinearity.

Multicollinearity: VIF and Condition Index

Belsley (1991) proposes to calculate the VIF and the condition number, using R_X, the correlation matrix of the standardized regressors:

$\text{VIF}_k = \text{diag}(R_X^{-1})_k$

Condition Index = $\kappa_k = \text{sqrt}(\lambda_1/\lambda_k)$

where $\lambda_1 > \lambda_2 > ... > \lambda_p > ...$ are the ordered eigenvalues of R_X.

Belsley's (1991) rules of thumb for κ_k:

- below 10	⇒ good
- from 10 to 30	⇒ concern
- greater than 30	⇒ trouble
- greater than 100	⇒ disaster.

Another common rule of thumb: If $\text{VIF}_k > 5$, concern.

Best approach: Recognize the problem and understand its implications for estimation.

Note: Unless we are very lucky, some degree of multicollinearity will always exist in the data. The issue is: when does it become a problem?

Multicollinearity: Example

Example: Check for multicollinearity for IBM returns 3-factor model

```
library(olsrr)
ols_vif_tol(fit_capm)
ols_eigen_cindex(fit_capm )
```

```
> ols_vif_tol(fit_capm)
  Variables     Tolerance      VIF
1   xMkt_RF     0.8901229 1.123440
2   xSMB        0.9147320 1.093216
3   xHML        0.9349904 1.069530
```

```
> ols_eigen_cindex(fit_capm)
  Eigenvalue Condition Index  intercept      xMkt_RF     xSMB     xHML
1  1.4506645      1.000000     0.01557614    0.24313961    0.212001760    0.1518949
2  1.0692689      1.164770     0.66799183    0.01432250    0.001789253    0.2129328
3  0.7967889      1.349310     0.16184731    0.01239755    0.576432492    0.4107435
4  0.6832777      1.457085     0.15458473    0.73014033    0.209776495    0.2244287
```

Note: Multicollinearity does not seem to be a problem. ¶

Lecture 4 - OLS: Sampling, and Bootstrapping

OLS Estimation - Assumptions

CLM Assumptions

(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$ is correctly specified.

(**A2**) $E[\varepsilon|\mathbf{X}] = 0$

(**A3**) $Var[\varepsilon|\mathbf{X}] = \sigma^2\,\mathbf{I}_T$

(**A4**) $\mathbf{X}$ has full column rank – rank($\mathbf{X}$)=k-, where $T \geq k$.

• From assumptions (**A1**), (**A2**), and (**A4**)

$$\Rightarrow \mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\,\mathbf{y}$$

We define $\mathbf{e} = (\mathbf{y} - \mathbf{Xb})$ $\quad\Rightarrow \mathbf{X}'\mathbf{e} = \mathbf{X}'\,(\mathbf{y} - \mathbf{Xb})\ \mathbf{X}'\mathbf{y} - \mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{0}$

• Now, we will study the properties of **b**.

Sampling Distribution of b

Small sample = For *all* sample sizes –i.e., for all values of *T* (or *N*).

$$\mathbf{b} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} \quad \Rightarrow \ \mathbf{b} \text{ is a vector of random variables.}$$

• Properties

(1) $E[\mathbf{b}\,|\,\mathbf{X}] = \boldsymbol{\beta}$

(2) $Var[\mathbf{b}\,|\,\mathbf{X}] = E[(\mathbf{b} - \boldsymbol{\beta})\,(\mathbf{b} - \boldsymbol{\beta})'\,|\,\mathbf{X}] = \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1}$

(3) Gauss-Markov Theorem: **b** is BLUE (MVLUE).

(4) If (**A5**) $\boldsymbol{\varepsilon}\,|\,\mathbf{X} \sim N(\mathbf{0}, \sigma^2\,\mathbf{I}_T)$ $\quad\Rightarrow \mathbf{b}\,|\,\mathbf{X} \sim N(\boldsymbol{\beta}, \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1})$

$$\Rightarrow b_k|\ \mathbf{X} \ \sim N(\beta_k, \sigma^2\,(\mathbf{X}'\mathbf{X})_{kk}{}^{-1})$$

(Note: the last implication is derived from the fact that the marginal distributions of a multivariate normal are also normal.)

Note: Under (**A5**), **b** is also the MLE. Thus, it has all the nice MLE properties: efficiency, consistency, sufficiency and invariance!

Sampling Distribution of b

Recall that a sample statistic like **b** is a function of RVs. Then, it has a statistical distribution.

In general, in finance, we observe only *one* sample mean (actually, our only sample). But, *many* sample means are possible from the DGP.

• A *sampling distribution* is a distribution of a statistic over all possible samples.

Let's generate some y_i's using a DGP and, then, some **b**'s. Using:

$$\mathbf{b} = \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} = \boldsymbol{\beta} + \Sigma_i\, \mathbf{v}_i'\varepsilon_i$$

Set $\boldsymbol{\beta}$ = .4; then, the DGP is:

$$\mathbf{y} = (.4)\,\mathbf{X} + \boldsymbol{\varepsilon}$$

(1) Generate **X** (to be treated as numbers). Say **X** ~ N(2,4)

$\Rightarrow x_1$=**3.22**, x_2=**2.18**, x_3=-0.37, ..., x_T =1.71

(2) Generate $\boldsymbol{\varepsilon}$ ~ N(0,1)

$\Rightarrow$ draws ε_1=**0.52**, ε_2=**-1.23**, ε_3=1.09, ..., ε_T=-0.09

(3) Generate **y** = .4 **X** + $\boldsymbol{\varepsilon}$

$\Rightarrow y_1 = .4 * 3.22 + \mathbf{0.52} = 1.808$

$y_2 = .4 * \mathbf{2.18} + (\mathbf{-1.23}) = -0.358$

$y_3 = .4 * (-0.37) + 1.09 = 0.942$

... $y_T = .4 * 1.71 + (-0.09) = 0.594$

(4) Generate $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y} = \Sigma_i\,(x_i - \bar{x})\,(y_i - \bar{y})/\,\Sigma_i\,(x_i - \bar{x})^2$

- We want to generate many **b**'s. Steps

(1) Generate **X** (to be treated as numbers). Say **X** ~ N(2,4)

(2) Generate $\boldsymbol{\varepsilon}$ ~ N(0,1)

(3) Generate **y** = .4 **X** + $\boldsymbol{\varepsilon}$

(4) Generate $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y} = \Sigma_i\,(x_i - \bar{x}\,)\,(x_i - \bar{y})/\,\Sigma_i\,(x_i - \bar{x})^2$

Conditioning on step (1), we can repeat (2)-(4) B times, say 1,000 times. Then, we are able to generate a sampling distribution for **b**.

We can, obviously, play with *T*; say *T*=100; 1,000; 10,000.

We can check: $E[\mathbf{b}|\mathbf{X}] = (1/B)\,\Sigma_i\, b_i = \boldsymbol{\beta}$?

We can calculate the variance of Var[**b**|**X**].

Sampling Distribution of b – Code in R

Steps (1)-(4) in R to generate **b**, with a sample of size *T*=100:

```
> T <- 100                                # sample size
> x <- rnorm(T,2,2)                       # generate x from a N(2, 2^2).
> ep <- rnorm(T,0,1)                      # generate errors from a N(0, 1).
> y <- .4*x + ep                          # generate y
> b <-solve(t(x)%*% x)%*% t(x)%*%y        # OLS regression
```

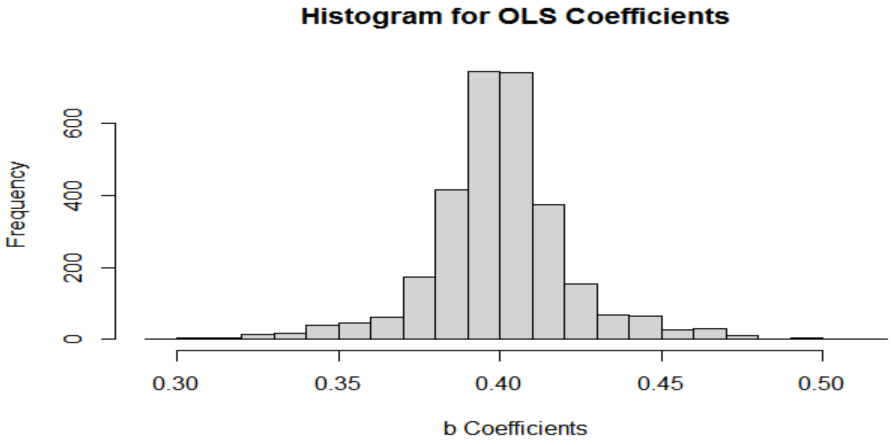
We run these commands B (say, B=1,000) times to get the sampling distribution of **b**. Then, we can calculate means, variances, skewness and kurtosis coefficients, etc.

- Script to generate the sampling distribution for B=1,000 & T=100:

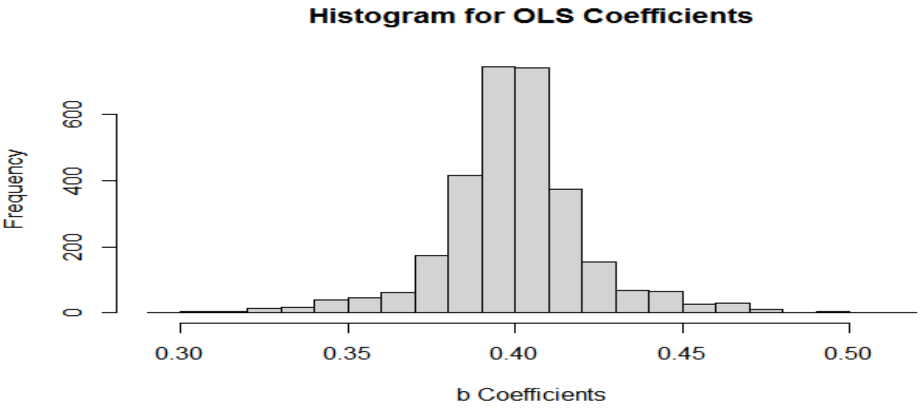
```
Allbs=NULL                                #Initialize vector that collects the b
T <- 100
x <- rnorm(T,2,2)                         # generate x
reps=1000                                 # number of repetitions (B)
```

```
for (i in seq(1,reps,1)){                          # “for” loop starts
ep <-  rnorm(T,0,1)                       # generate errors, ep
y <-  .4*x+ep                             #generate y
b <- solve(t(x)%*% x)%*% t(x)%*%y         #OLS regression
Allbs=rbind(Allbs,b)                      #accumulate b as rows
}                                         # loop ends
mb <- mean(Allbs)
varb <- var(Allbs)
hist(Allbs[,1],main="Histogram for OLS Coefficients",  xlab="b Coefficients")
```

For T=100
B = 1,000
Mean[b] = **0.3995132**
SD[b] = **0.02613134**



For T=1,000
B = 1,000
Mean[b] = **0.3999375**
SD[b] = **0.022086**



Bootstrapping (Again!)

Bootstrapping is the practice of estimating the properties of an estimator -say, its variance- by measuring those properties when sampling from an approximating distribution (the *bootstrap DGP*).

<u>Idea</u>: We use the data at hand -the empirical distribution (ED)- to estimate the variation of statistics that are themselves computed from the same data. Recall that, for large samples drawn from *F*, the ED approximates the CDF of *F* very well.

Thus, an easy choice for an approximating distribution is the ED of the observed data. That is, the ED becomes a "*fake population*."

John Fox (2005, UCLA): *"The population is to the sample as the sample is to the bootstrap samples."*

Bootstrapping: Empirical Bootstrap

Suppose we have a dataset with *N i.i.d.* observations drawn from *F*:

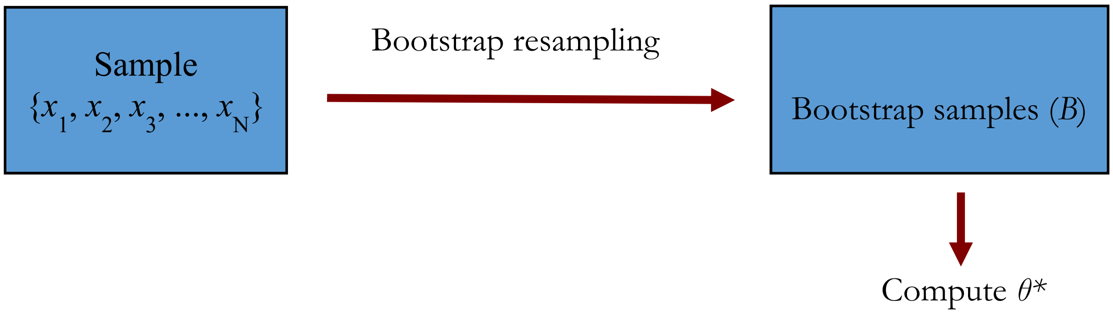
$\{x_1, x_2, x_3, ..., x_N\}$ -"*fake population*."

From the ED, *F**, we sample with replacement *N* observations:

$\{x_1^*, x_2^*, x_3^*, ..., x_N^*\}$ - a bootstrap sample

This is an *empirical bootstrap sample*, which is a resample of the same size *N* as the original data, drawn from *F**.

For any statistic θ computed from the original sample data, we can define a statistic θ^* by the same formula, but computed instead using the resampled data.



• θ^* is computed by resampling the original data; we can compute many θ^* by resampling many times from *F**. Say, we resample θ^* *B* times:

$$\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, ..., \hat{\theta}_B^*\}.$$

From this collection of $\hat{\theta}^*$'s, we can compute moments, C.I.'s, etc.

Bootstrap Steps:

1. From the original sample, draw random sample with size *N*.
2. Compute statistic θ from the resample in 1: $\hat{\theta}_1^*$.
3. Repeat steps 1 & 2 *B* times ⇒ Get B statistics: $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, ..., \hat{\theta}_B^*\}$
4. Compute moments, draw histograms, etc. for these B statistics.

• Results:
1. With a large enough *B*, the LLN allows us to use the $\hat{\theta}*$'s to estimate the distribution of $\hat{\theta}$, $F(\hat{\theta})$.
2. The variation in $\hat{\theta}$ is well approximated by the variation in $\hat{\theta}*$.
Result **2** is the one we used in Lecture 2 to estimate the size of a C.I.

Bootstrapping: Variations

If the ED is used for the draws, the method is usually called the *nonparametric bootstrap*. If a distribution is assumed, say a t-distribution, and we draw from this distribution, the method is called the *parametric bootstrap.*

• If the y's and the x's are sampled together, this method is sometimes called the *paired bootstrap* –for example, in a regression.

• If blocks of data are sample together, the method is called *block bootstrap* –for example, in the presence of correlated data, typical of time series or spatial data.

Bootstrapping: Why?

Question: Why do we need a bootstrap?
- *N* is "small," asymptotic assumptions do not apply.
- DGP assumptions are violated.
- Distributions are complicated.

The main appeal is its simplicity and its *consistent* results.

Bootstrapping in Econometrics

Bootstrapping provides a very general method to estimate a wide variety of statistics. It is most useful when:
(1) Reliance on "formulas" is problematic because the formula's assumptions are dubious.
(2) A formula holds only as $T \to \infty$*, but our sample is not very big.*
(3) A formula is complicated or it has not even been worked out yet.

The most common econometric applications are situations where you have a consistent estimator of a parameter of interest, but it is hard or impossible to calculate its standard error or its C.I.

Technical note: Bootstrapping is easiest to implement if the estimator is "smooth," $\sqrt{T}$-consistent, and based on an *i.i.d.* sample. In other situations, it is more complicated.

Bootstrapping in Econometrics: Example

You are interested in the relation between CEO's education (**X**) and firm's long-term performance (**y**). You have 1,500 observations on both variables. You estimate the correlation coefficient, ρ, with its sample counterpart, *r.* You find the correlation to be very low.

Q: How reliable is this result? The distribution of *r* is complicated. You decide to use a bootstrap to study the distribution of *r*.

Randomly construct a sequence of *B* samples (all with *T*= 1,500). Say,

$B_1 = \{(x_1,y_1), (x_3,y_3), (x_6,y_6), (x_6,y_6), ..., (x_{1458},y_{1458})\} \Rightarrow r_1$

$B_2 = \{(x_5,y_5), (x_7,y_7), (x_{11},y_{11}), (x_{12},y_{12}), ..., (x_{1486},y_{1486})\} \Rightarrow r_2$

. . . .

$B_B = \{(x_2,y_2), (x_2,y_2), (x_2,y_2), (x_3,y_3), ..., (x_{1499},y_{1499})\} \Rightarrow r_B$

We rely on the observed data. We take it as our "fake population" and we sample from it *B* times. We have a collection of *bootstrap subsamples*.

The sample size of each bootstrap subsample is the same (*T*). Thus, some elements are repeated.

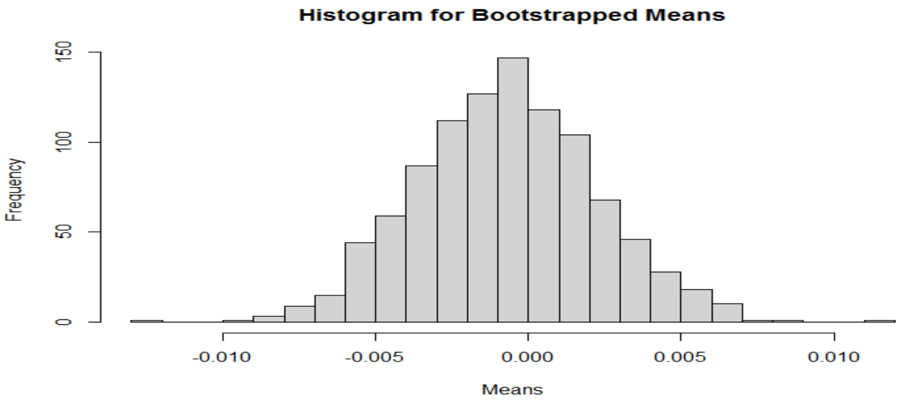
Now, we have a collection of estimators of ρ_i's: $\{r_1, r_2, r_3, ..., r_B\}$. We can do a histogram and get an approximation of the probability distribution. We can calculate its mean, variance, kurtosis, confidence intervals, etc.

Bootstrapping in Econometrics: Estimating the mean

Example: We bootstrap the mean returns of IBM, using monthly data 1973-2020, with ***B*** = **1,000.** (You need to install R package *boot*.)

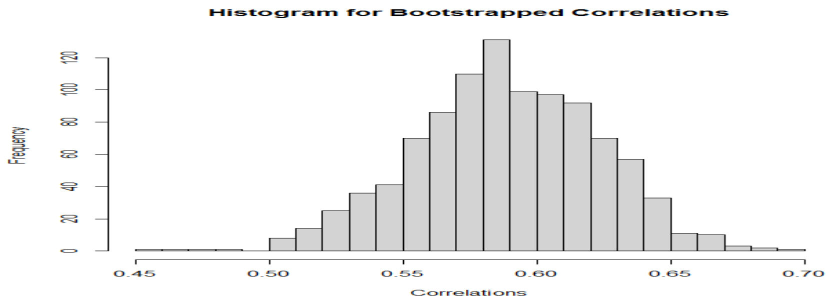
```
sim_size = 1000

library(boot)
# function to obtain the mean from the data
mean_p <- function(data, i) {
        d <-data[i]
return(mean(d))
}

# bootstrapping with sim_size replications
boot.samps <- boot(data=ibm_x, statistic=mean_p,   R=sim_size)

# view stored bootstrap samples and compute mean
boot.samps                          # Print original mean, bias and SE of bootstraps
mean(boot.samps$t)                  # The bootstrapped estimate of the mean
sd(boot.samps$t)                    # SD of the mean estimate
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = ibm_x, statistic = mean_p, R = sim_size)

Bootstrap Statistics :
        original          bias     std. error
t1* -0.0006990633 5.021474e-07 0.002964358

> boot.samps$t[1:10]                          # Show first 10 bootstrapped mean
 [1] -0.0066684274  0.0011648002 -0.0010053505 -0.0024989738 -0.0025442486
 [6] 0.0007935133 -0.0039867127  0.0030962313 -0.0017929592 -0.0023480292

> mean(boot.samps$t)                          # The estimate of the bootstrapped mean
[1] -0.0006985612
> sd(boot.samps$t)                            #SD of the bootstrapped mean
[1] 0.002964358

> # Elegant histogram
> hist(boot.samps$t,main="Histogram for Bootstrapped Means",
+      xlab="Means", breaks=20)
```



Example: We bootstrap the correlation between the returns of IBM & the S&P 500, using monthly data 1973-2020, with ***B* = 1,000**.

```
sim_size = 1000
x_sp <- SFX_da$SP500
lr_sp <- log(x_sp[-1]/x_sp[-T])
dat_spibm <- data.frame(lr_sp, lr_ibm)

library(boot)
# function to obtain the correlation coefficient from the data
cor_xy <- function(data, i) {
        d <-data[i,]
        return(cor(d$lr_sp,d$lr_ibm))
}
# bootstrapping with sim_size replications
boot.samps <- boot(data=dat_spibm, statistic=cor_xy,   R=sim_size)

# view stored bootstrap samples and compute mean
boot.samps                              # Print original ρ, bias and SE of bootstraps
mean(boot.samps$t)                      # our estimate of the correlation
sd(boot.samps$t)                        # SD of the correlation estimate

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = dat_spibm, statistic = cor_xy, R = sim_size)

Bootstrap Statistics :
    original      bias     std. error
t1* 0.5894632 -0.001523914  0.03406313


> boot.samps$t[1:10]                    #show first 10 bootstrapped correlations coeff
 [1] 0.5863186 0.5898572 0.6473122 0.6473249 0.5311525 0.5734280 0.6241236 0.5790740
 [9] 0.5790095 0.5932918
> mean(boot.samps$t)                    #our estimate of the correlation
[1] 0.5879392
> sd(boot.samps$t)                      #SD of the correlation estimate
[1] 0.03406313

> # Elegant histogram
> hist(boot.samps$t,main="Histogram for Bootstrapped Correlations",
+     xlab="Correlations", breaks=20)
```



- Simple 95% **percentile method** C.I.

```
> new <- sort(boot.samps$t)
> new[25]
```

```
[1] 0.5151807
> new[975]
[1] 0.6495722
```

Note: You get same results using
boot.ci(boot.samps, type = "**perc**")

- **Empirical boostrap method** C.I.

```
> boot.ci(boot.samps, type="basic")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.samps, type = "basic")

Intervals :
Level     Percentile
95%   ( 0.5293, 0.6637 )
Calculations and Intervals on Original Scale. ¶
```

Bootstrapping: How many bootstraps?

It is not clear. There are many theorems on asymptotic convergence, but there are no clear rules regarding *B*. There are some suggestions.

Efron and Tibsharani's (1994) textbook recommends *B*=200 as enough. (Good results with *B* as low as 25!)

Davidson and Mackinnon's (2001) textbook suggests steps to select *B*. In the D&M simulations, on average, *B* is between 300 and 2,400.

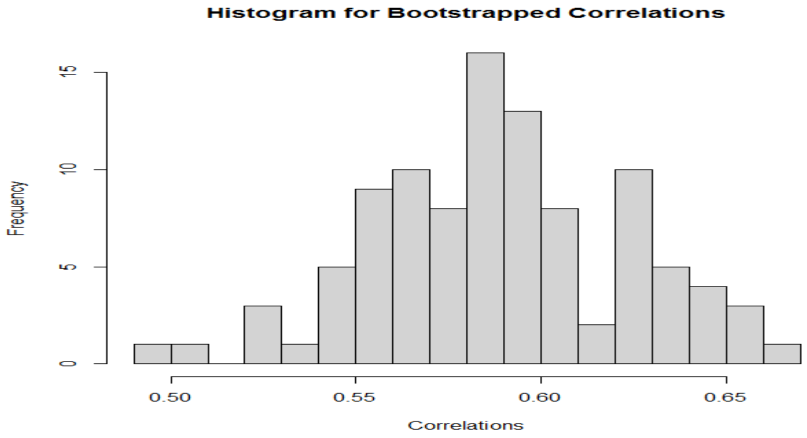
Wilcox's (2010) textbook recommends "599 [...] for general use."

Rule of thumb: Start with *B*=100, then, try *B*=1,000, and see if your answers have changed by much. Increase bootstraps until you get stability in your answers.

Example: We bootstrap the correlation between IBM returns and S&P 500 returns, using ***B* = 100**.

```
> # view bootstrap results
> boot.samps
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
```

```
boot(data = dat_spibm, statistic = cor_xy, R = sim_size)

Bootstrap Statistics :
     original      bias    std. error
t1* 0.5898636 -0.00115623  0.03449216

> mean(boot.samps$t)
[1] 0.5887074
> sd(boot.samps$t)
[1] 0.02885868. ¶
```



Example: We bootstrap the correlation between IBM returns and S&P 500 returns, using ***B* = 25**.

```
> # view bootstrap results
> boot.samps
ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = dat_spibm, statistic = cor_xy, R = sim_size)

Bootstrap Statistics :
     original      bias                 std. error
t1* 0.5898636 -0.00115623  0.03449216

> mean(boot.samps$t)
[1] 0.5847676
> sd(boot.samps$t)
[1] 0.03449216
```

Conclusion: Results do not change that much. ¶

Bootstrapping: Bias

You can estimate the bias of the bootstrap of a parameter, say **b**:

$$Bias(\mathbf{b}) = (1/B)\Sigma_r\, \mathbf{b}(r) - \mathbf{b}$$

Note: In the OLS case, **b** is an unbiased estimator, but as an estimate, the bias can be non-zero. This estimate must be analyzed along the SE's.

Example: In the previous bootstrapping correlations exercise, R displays the bias:

```
Bootstrap Statistics :
          original         bias        std. error
t1*      0.5898636     -0.001244376   0.03455582.   ¶
```

Bootstrapping: Linear Model - Var[b]

Some assumptions in the CLM are not reasonable –for example, normality. If we assuming normality (A5), we also assume the sampling distribution of **b**. But if data is not normal, results are only asymptotic.

We can use a bootstrap to estimate the sampling distribution of **b**. It can give us a better idea of the small sample distribution. Then, we can estimate the Var[**b**].

Monte Carlo (MC=repeated sampling) method:

1. Estimate model using full sample (of size T) ⇒ we get **b**
2. Repeat B times:
   - Draw T observations from the sample, *with replacement*
   - Estimate $\boldsymbol{\beta}$ with **b**(r).
3. Estimate variance with

$$\mathbf{V}_{\mathbf{boot}} = (1/B)\,[\mathbf{b}(r) - \mathbf{b}][\mathbf{b}(r) - \mathbf{b}]'$$

• In the case of one parameter, say $\mathbf{b}_1$: Estimate variance with

$$\text{Var}_{\mathbf{boot}}[\mathbf{b}_1] = (1/B)\Sigma_r\, [\mathbf{b}_1(r) - \mathbf{b}_1]^2$$

You can also estimate Var[$\mathbf{b}_1$] as the variance of $\mathbf{b}_1$ in the bootstrap

$$\text{Var}_{\mathbf{boot}}[\mathbf{b}_1] = (1/B)\Sigma_r\, [\mathbf{b}_1(r) - \text{mean}(\mathbf{b}_{1\text{-}r})]^2;$$

$$\text{mean}(\mathbf{b}_{1\text{-}r}) = (1/B)\Sigma_r\, \mathbf{b}_1$$

Note: Obviously, this method for obtaining standard errors of parameters is most useful when no formula has been worked out for the standard error (SE), or the formula is complicated –for example, in some 2-step estimation procedures.

Bootstrapping: Linear Model - Estimating Var[b]

Example: We bootstrap the SE for **b** for IBM returns using the 3 FF Factor Model. We use the R package *lmboot*, which needs to be installed with the **install.packages()** function.

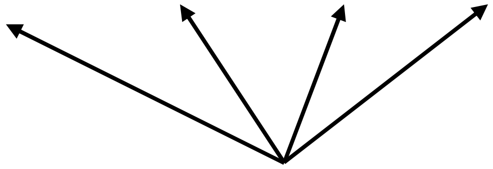
```
library(lmboot)                                         # need to run before
install.packages(“lmboot”)
y <- ibm_x
x <- cbind(x0, Mkt_RF, SMB, HML)
dat_yx <- data.frame(y, x)                    # lmboot needs an R data frame. We make one.
ff3_b <- paired.boot(y ~ x-1, data=dat_yx, B = sim_size)
ff3_b$origEstParam                            # print OLS results (“original estimates”)
> ff3_b$origEstParam
            [,1]
x           -0.005088944
xMkt_RF      0.908298898
xSMB        -0.212459588
xHML        -0.171500223

# Mean values for b
mean(ff3_b$bootEstParam[,1])       # print mean of bootstrap samples for constant
mean(ff3_b$bootEstParam[,2])       # print mean of bootstrap samples for Mkt_RF
mean(ff3_b$bootEstParam[,3])       # print mean of bootstrap samples for SMB
mean(ff3_b$bootEstParam[,4])       # print mean of bootstrap samples for HML

# Statistics for sampling distribution of b
summary(ff3_b$bootEstParam)         # distribution of b

# SD of parameter vector b
sd(ff3_b$bootEstParam[,1])
sd(ff3_b$bootEstParam[,2])
sd(ff3_b$bootEstParam[,3])
sd(ff3_b$bootEstParam[,4])

# bootstrap bias
ff3_b$origEstParam[1] - mean(ff3_b$bootEstParam[,1])
ff3_b$origEstParam[2] - mean(ff3_b$bootEstParam[,2])
ff3_b$origEstParam[3] - mean(ff3_b$bootEstParam[,3])
ff3_b$origEstParam[4] - mean(ff3_b$bootEstParam[,4])

> summary(ff3_b$bootEstParam)
             x                     xMkt_RF         xSMB                    xHML
 Min.      :-0.012159     Min.    :0.7115     Min.     :-0.5175     Min.     :-0.4699
 1st Qu.   :-0.006731     1st Qu. :0.8669     1st Qu.  :-0.2890     1st Qu.  :-0.2362
 Median    :-0.005074     Median  :0.9087     Median   :-0.2185     Median   :-0.1690
 Mean      :-0.005008     Mean    :0.9068     Mean     :-0.2125     Mean     :-0.1710
 3rd Qu.   :-0.003273     3rd Qu. :0.9492     3rd Qu.  :-0.1415     3rd Qu.  :-0.1086
 Max.      : 0.002293     Max.    :1.0854     Max.     : 0.1909     Max.     : 0.2477

> sd(ff3_b$bootEstParam[,1])
[1] 0.002493708
```

```
> sd(ff3_b$bootEstParam[,2])
[1] 0.06132218
> sd(ff3_b$bootEstParam[,3])
[1] 0.1108
> sd(ff3_b$bootEstParam[,4])
[1] 0.09729972
```

- Comparing OLS and Bootstrap

	OLS		**Bootstrap**		**Bias** (2)-(1)
	Coeff. (1)	S.E.	Coeff. (2)	S.E.	
x	-0.00509	0.00249	-0.00501	0.00249	8.0765e-05
xMkt_RF	**0.90829**	0.05672	**0.90684**	**0.06132**	-0.0014571
xSMB	**-0.21246**	0.08411	**-0.21245**	**0.11080**	1.9914e-06
xHML	-0.17150	0.08468	**-0.17099**	**0.09730**	0.0005133

```
> ff3_b$bootEstParam[1:10,]          # print the first 10 of B=1,000 bootstrap samples
               x      xMkt_RF        xSMB        xHML
  [1,] -6.109007e-03 0.9186830 -0.1299534100 -0.163421636
  [2,] -1.757503e-03 0.8333006 -0.2067565390 -0.147604991
  [3,] -3.907573e-03 0.9746878 -0.2870744815 -0.169189619
  [4,]  1.596103e-03 0.9185157 -0.2937731120 -0.296972497
  [5,] -8.409239e-03 0.7309406 -0.0681714313 -0.149883639
  [6,] -1.998929e-03 0.9133751 -0.3001713380 -0.315913280
  [7,] -6.289286e-03 0.9441856 -0.2276894034 -0.058924929
  [8,] -5.533354e-03 0.8210057 -0.2221866298 -0.078512341
  [9,] -6.152301e-03 1.0389917 -0.2592958758 -0.237930809
 [10,] -3.778058e-03 0.9544829 -0.1859554067 -0.217702583
```



From the B samples, we compute variances and SD as usual. ¶

Bootstrapping: Some Remarks

Question: How reliable is bootstrapping?

- There is still no consensus on how far it can be applied, but for now nobody is going to dismiss your results for using it.
- There is a general agreement that for normal and close to normal (and symmetric) distributions it works well.
- Bootstrapping is more problematic for skewed distributions.
- It can be unreliable for situations where there are not a lot of observations. Typical example in finance: estimation of quantiles in the tails of returns distributions.

Note: We presented two simple examples. There are many bootstraps variations. We will not cover them.

Lecture 5 - Testing in the CLM

Review – OLS Assumptions

CLM Assumptions

(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) $E[\varepsilon|\mathbf{X}] = 0$
(**A3**) $Var[\varepsilon|\mathbf{X}] = \sigma^2\,\mathbf{I}_T$
(**A4**) **X** has full column rank –rank(**X**)=*k*-, where $T \geq k$.

Issues for this lecture:
Q: What happens when we impose restrictions to the DGP (**A1**)?

Q: How do we test restrictions in the context of OLS estimation?

OLS Subject to Linear Restrictions

Restrictions: Theory imposes certain restrictions on parameters and provide the foundation of several tests. In this Lecture, we only consider linear restrictions, written as $\mathbf{R}\beta = \mathbf{q}$.
The dimension of **R** is *J*x*k*, where *J* is the number of restrictions, and *k* is the number of parameters. β, as usual, is a *k*x1 column vector. Then, **q** is a *J*x1 column vector.

Examples:

(1) Dropping variables from the equation. That is, certain coefficients in **b** forced to equal 0. For example, in the 3 Fama-French factor model, are variables $\mathbf{x}_3$=*SMB* and $\mathbf{x}_4$=*HML* significant?. Using the above notation:

$$\mathbf{R}\beta = \mathbf{q} \qquad \Rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \beta_1 \\ \beta_{Mkt} \\ \beta_{SMB} \\ \beta_{HML} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

We have two restrictions (*J*=2): $\beta_{SMB} = 0$ & $\beta_{HML} = 0$. We have *k=4* parameters.
⇒ **R** is a 2x4 matrix, β is a 4x1 vector, and **q** is a 2x1 vector.

Note: The restrcitions make the FF model into the traditional CAPM.

(2) Adding up conditions: Sums of certain coefficients must equal fixed values. Adding up conditions in demand systems. In a CAPM setting, the sum of all cross-sectional β_i's should be equal to 1. For example, in the 3 Fama-French factor model, we force $\boldsymbol{\beta_{SMB}} + \boldsymbol{\beta_{HML}} = 1$.

$$\mathbf{R}\beta = \mathbf{q} \qquad \Rightarrow [0 \quad 0 \quad 1 \quad 1] * \begin{bmatrix} \beta_1 \\ \beta_{Mkt} \\ \beta_{SMB} \\ \beta_{HML} \end{bmatrix} = 1$$

We have one restrictions (*J*=1): $\beta_{SMB} + \beta_{HML} = 1$. We have *k=4* parameters.
⇒ **R** is a 1x4 matrix (a row vector), β is a 4x1 vector, and **q** is a scalar.

(3) Equality restrictions: Certain coefficients must equal other coefficients. Using real vs. nominal variables in equations. For example, in the 3 Fama-French factor model, we force $\boldsymbol{\beta}_{\boldsymbol{SMB}} = \boldsymbol{\beta}_{\boldsymbol{HML}}$.

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{q} \qquad \Rightarrow [0 \quad 0 \quad 1 \quad -1] * \begin{bmatrix} \beta_1 \\ \beta_{Mkt} \\ \beta_{SMB} \\ \beta_{HML} \end{bmatrix} = 0.$$

We have one restrictions (J=1): $\beta_{SMB} + \beta_{HML} = 1$. We have k=4 parameters.

⇒ **R** is a 1x4 matrix (a row vector), **β** is a 4x1 vector, and **q** is a scalar. ¶

• Common formulation: We minimize the error sum of squares, subject to the linear restrictions. That is,

$$\text{Min}_{\mathbf{b}} \; \{S(x_i, \theta) = \Sigma_i\, \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\,(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\} \qquad \text{s.t. } \mathbf{R}\boldsymbol{\beta} = \mathbf{q}$$

In practice, restrictions can usually be imposed by solving them out. Suppose we have a model:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

(1) Dropping variables –i.e., force a coefficient to equal zero, say β_3.

Problem: $\text{Min}_\beta \; \sum_{i=1}^{n} \left(y_i - \beta_1 x_{i1} - \beta_2 x_{i2} - \beta_3 x_{i3}\right)^2 \quad s.t. \;\; \beta_3 = 0$

$\text{Min}_\beta \; \sum_{i=1}^{n} \left(y_i - \beta_1 x_{i1} - \beta_2 x_{i2}\right)^2$

(2) Adding up. Suppose we impose: $\beta_1 + \beta_2 + \beta_3 = 1$. Then, $\beta_3 = 1 - \beta_1 - \beta_2$.
Substituting in model:

$$(\mathbf{y} - \mathbf{x}_3) = \beta_1 (\mathbf{x}_1 - \mathbf{x}_3) + \beta_2 (\mathbf{x}_2 - \mathbf{x}_3) + \mathbf{e}.$$

Problem: $\text{Min}_\beta \sum_{i=1}^{n} \left((y_i - x_{i3}) - \beta_1 (x_{i1} - x_{i3}) - \beta_2 (x_{i2} - x_{i3})\right)^2$

(3) Equality. Suppose we impose: $\beta_2 = \beta_3$.
Substituting in model:

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \beta_2 \mathbf{x}_3 + \mathbf{e} = \beta_1 \mathbf{x}_1 + \beta_2 (\mathbf{x}_2 + \mathbf{x}_3) + \mathbf{e}$$

Problem: $\text{Min}_\beta \; \sum_{i=1}^{n} \left(y_i - \beta_1 x_{i1} - \beta_2 x_{i2} - \beta_3 x_{i3}\right)^2 \quad s.t. \;\; \beta_2 = \beta_3$

$\text{Min}_\beta \sum_{i=1}^{n} \left(y_i - \beta_1 x_{i1} - \beta_2 (x_{i2} + x_{i3})\right)^2$

• Before setting the general restricted LS problem, we look at the simplest case: one explanatory variable (x) and one restriction ($r\beta = q$).

Then, the Lagrangean (recall values of Lagrange multiplier', λ, play no role):

$$Min_{\beta,\lambda} \quad L(\beta, \lambda \mid y, x) = \sum_{t=1}^{T} (y_t - x_t \beta)^2 + 2\lambda(r\beta - q)$$

Then, the f.o.c. are:

$$\frac{\partial L}{\partial \beta} = \sum\nolimits_{t=1}^{T} 2(y_t - x_t b^*)(-x_t) + 2\lambda r = 0 \qquad \Rightarrow -\sum\nolimits_{t=1}^{T} (y_t x_t - x_t^2 b^*) + \lambda r = 0$$

$$\frac{\partial L}{\partial \lambda} = 2(rb^* - q) = 0 \qquad \Rightarrow (rb^* - q) = 0$$

From the 1st equation

$$-(x'y - x'xb^*) + \lambda r = 0 \qquad \Rightarrow \; b^* = (x'x)^{-1} x'y - (x'x)^{-1} \lambda r$$
$$= b - (x'x)^{-1} \lambda r$$

$b^* = b - r\,(\mathbf{x'x})^{-1}\lambda$ ⇒ Restricted OLS = OLS + "*correction*"

Premultiplying both sides by r and then subtract q:

$$rb^* - q = rb - r^2\,(\mathbf{x'x})^{-1}\lambda - q$$
$$0 = -\,r^2\,(\mathbf{x'x})^{-1}\lambda + (rb - q)$$

Solving for λ ⇒ $\lambda = [r^2\,(\mathbf{x'x})^{-1}]^{-1}\,(rb - q)$

Substituting in b* ⇒ $b^* = b - (\mathbf{x'x})^{-1} r\,[r^2\,(\mathbf{x'x})^{-1}]^{-1}\,(rb - q)$

This is the Restricted OLS estimator.

- Properties of Restricted OLS.

Property 1. Taking expectations of b*:

$$E[b^*|X] = E[b|X] - (\mathbf{x'x})^{-1} r\,[r^2\,(\mathbf{x'x})^{-1}]^{-1}\,E[(rb - q)|X]$$
$$= \beta - (\mathbf{x'x})^{-1} r\,[r^2\,(\mathbf{x'x})^{-1}]^{-1}\,(r\beta - q)$$

Implications:

If the restriction is *true* –i.e., $(r\beta = q)$ ⇒ $E[b^*|X] = \beta$

If the restriction is *not true* –i.e., $(r\beta \neq q)$ ⇒ $E[b^*|X] \neq \beta$

Then, if theory imposes a correct restriction, then, b* is *unbiased:*

$$E[b^*|X] = \beta$$

In practice, if restriction is true, the restricted and unrestricted estimators should be similar.

Note: If theory is correct, the expected shadow price is 0!

$$E[\lambda|X] = [r^2\,(\mathbf{x'x})^{-1}]^{-1}\,E[(rb - q)|X] = 0$$

That is, you would pay nothing to release the restriction, $r\beta = q$.

Property 1. We compute the Var[b*]. It can be shown that

$$\text{Var}[b^*|X] = \text{Var}[b|X] - \sigma^2\,(\mathbf{x'x})^{-1}\,r\,[r^2\,(\mathbf{x'x})^{-1}]^{-1}\,r\,(\mathbf{x'x})^{-1}$$
$$\Rightarrow \text{Var}[b|X] - \text{Var}[b^*|X] = \sigma^2\,(\mathbf{x'x})^{-1}\,r\,[r^2\,(\mathbf{x'x})^{-1}]^{-1}\,r\,(\mathbf{x'x})^{-1} > 0.$$

⇒ The restricted OLS estimator is more efficient!

Remark from Properties 1 and 2: It is common to select an estimator based on the MSE (=RSS/*T*). The one with the lowest MSE is said to be more "*precise*."

We can decompose the MSE of an estimator, $\hat{\theta}$, as:

$$\text{MSE}[\hat{\theta}] = \text{Variance}[\hat{\theta}] + \text{Squared bias}[\hat{\theta}]$$

For an unbiased estimator, like **b** ⇒ MSE [**b**] = Var[**b**|**X**]

• Back to **b***. Suppose the theory is incorrect ⇒ **b*** is biased.

There may be situations (small bias, but much lower variance) where **b*** is more "precise" (lower MSE) than **b**. It is possible that a practitioner may prefer imposing a wrong H_0 to get a better MSE.

• For the general case, with *k* explanatory variables and J restrictions, which we write as:

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{q},$$

we have a programming problem:

$$\text{Minimize wrt } \boldsymbol{\beta} \qquad L^* = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad \text{s.t. } \mathbf{R}\boldsymbol{\beta} = \mathbf{q}$$

Quadratic programming problem: Minimize a quadratic criterion subject to a set of linear restrictions. We solve this minimizations problem using a Lagrangean formulation.

The Lagrangean approach (the 2 is for convenience, since the value of λ is irrelevant for extrema):

$$\text{Min}_{\mathbf{b},\lambda} \qquad L^* = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\,\boldsymbol{\lambda}\,(\mathbf{R}\boldsymbol{\beta} - \mathbf{q})$$

f.o.c.:

$$\partial L^*/\partial \mathbf{b}' = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{b}^*) + 2\mathbf{R}'\boldsymbol{\lambda} = \mathbf{0} \Rightarrow -\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{b}^*) + \mathbf{R}'\boldsymbol{\lambda} = \mathbf{0}$$

$$\partial L^*/\partial \lambda = 2(\mathbf{R}\mathbf{b}^* - \mathbf{q}) = \mathbf{0} \qquad \Rightarrow (\mathbf{R}\mathbf{b}^* - \mathbf{q}) = \mathbf{0}$$

where **b*** is the restricted OLS estimator.

f.o.c.:

$$-\mathbf{X}'(\mathbf{y} - \mathbf{X}\mathbf{b}^*) + \mathbf{R}'\boldsymbol{\lambda} = \mathbf{0} \qquad (1)$$

$$(\mathbf{R}\mathbf{b}^* - \mathbf{q}) = \mathbf{0} \qquad (2)$$

where **b*** is the restricted OLS estimator.

Then, from the 1st equation (and assuming full rank for **X**):

$$-\mathbf{X}'\mathbf{y} + \mathbf{X}'\mathbf{X}\mathbf{b}^* + \mathbf{R}'\boldsymbol{\lambda} = \mathbf{0} \Rightarrow \mathbf{b}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\boldsymbol{\lambda}$$

$$= \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\boldsymbol{\lambda}$$

Premultiply both sides by **R** and then subtract **q**

$$\mathbf{R}\mathbf{b}^* = \mathbf{R}\mathbf{b} - \mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\boldsymbol{\lambda}$$

$$\mathbf{R}\mathbf{b}^* - \mathbf{q} = \mathbf{R}\mathbf{b} - \mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\boldsymbol{\lambda} - \mathbf{q}$$

$$\mathbf{0} \quad = -\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\lambda + (\mathbf{Rb} - \mathbf{q})$$

Solving for λ $\Rightarrow$ $\lambda = [\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1} (\mathbf{Rb} - \mathbf{q})$

Substituting in **b*** $\Rightarrow$ $\mathbf{b}^* = \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{R\,b} - \mathbf{q})$

Note: Restricted OLS = Unrestricted OLS + "correction"

Restricted Least Squares

Question: How do linear restrictions affect the properties of the least squares estimator?

Restricted LS estimator: $\mathbf{b}^* = \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{Rb} - \mathbf{q})$

• Properties:

1. Unbiased? Yes, if Theory is correct!

$$E[\mathbf{b}^*|\mathbf{X}] = \beta \; - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1} E[(\mathbf{Rb} - \mathbf{q})|\mathbf{X}] = \beta$$

But, if Theory is incorrect: $E[(\mathbf{Rb} - \mathbf{q})|\mathbf{X}] \neq \mathbf{0}$ $\Rightarrow E[\mathbf{b}^*|\mathbf{X}] \neq \beta$.

2. Efficiency?

$$\text{Var}[\mathbf{b}^*|\mathbf{X}] = \sigma^2(\mathbf{X}'\mathbf{X})^{-1} - \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1} \mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}$$

$$\text{Var}[\mathbf{b}^*|\mathbf{X}] = \text{Var}[\mathbf{b}|\mathbf{X}] - \text{a nonnegative definite matrix} < \quad \text{Var}[\mathbf{b}|\mathbf{X}]$$

3. **b*** may be more "precise," where precision is measured by the MSE (=RSS/*T*).

We can decompose the MSE of an estimator, $\hat{\theta}$, as:

$$\text{MSE}[\hat{\theta}] = \text{Variance}[\hat{\theta}] + \text{Squared bias}[\hat{\theta}]$$

For an unbiased estimator, say **b**, then, MSE [**b**] = Var[**b**|**X**]

Suppose the theory is incorrect. Then, **b*** is biased. There may be situations (small bias, but much lower variance) where **b*** is more "precise" (lower MSE) than **b**. A practitioner may prefer imposing a wrong H_0 to get a better MSE.

Restricted Least Squares - Interpretation

1. $\mathbf{b}^* = \mathbf{b} - \mathbf{Cm}$, $\mathbf{m}$ = the "discrepancy vector" $\mathbf{Rb} - \mathbf{q}$.

Note: If $\mathbf{m} = \mathbf{0}$ $\Rightarrow \mathbf{b}^* = \mathbf{b}$. (Q: What does $\mathbf{m} = \mathbf{0}$ mean?)

2. $\lambda = [\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{Rb} - \mathbf{q}) = [\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}\mathbf{m}$

When does $\lambda = \mathbf{0}$? We usually think of λ as a "shadow price."

3. Combining results: $\mathbf{b}^* = \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\lambda$

4. We can show that RSS never decreases with restrictions:

$\mathbf{e}'\mathbf{e} = (\mathbf{y} - \mathbf{Xb})'(\mathbf{y} - \mathbf{Xb}) \leq \mathbf{e}^{*\prime}\mathbf{e}^* = (\mathbf{y} - \mathbf{Xb}^*)'(\mathbf{y} - \mathbf{Xb}^*)$

$\Rightarrow$ Restrictions cannot increase R^2 $\Rightarrow R^2 \geq R^{2*}$

- Two cases
  - Case 1: Theory is correct: $\mathbf{R}\boldsymbol{\beta} - \mathbf{q} = \mathbf{0}$ (restrictions hold).
    b* is unbiased & $\text{Var}[\mathbf{b}^*|\mathbf{X}] \leq \text{Var}[\mathbf{b}|\mathbf{X}]$

  - Case 2: Theory is incorrect: $\mathbf{R}\boldsymbol{\beta} - \mathbf{q} \neq \mathbf{0}$ (restrictions do not hold).
    b* is biased & $\text{Var}[\mathbf{b}^*|\mathbf{X}] \leq \text{Var}[\mathbf{b}|\mathbf{X}]$.

- Interpretation

- The theory gives us information.
  Bad information produces bias (away from “the truth.”)
  Any information, good or bad, makes us more certain of our answer. In this context, *any* information reduces variance.

Testing: Parameter vs Diagnostic

So far, the tests discussed in Lectures 3 & 4, involved parameters. We call these types of testing *parameter tests.*

When we tests the assumptions behind the CLM, for example, (A5), we perform a *diagnostic tests*.

• *Parameter testing:* We test economic H_0's.

Example: Test $\beta_k = 0$ -say, there is no size effect on the expected return equation. ¶

• *Diagnostic testing:* We test assumptions behind the model. In our case, assumptions (A1)-(A5) in the CLM.

Example: Test $E[\varepsilon|X] = 0$ -i.e., the residuals are zero-mean, white noise distributed errors. ¶

Review – Significance Testing

Fisher's *significance testing* procedure relies on the *p-value*: the probability of observing a result at least as extreme as the test statistic, under H_0.

• Fisher's Idea

1) Form H_0 & decide on a *significance level* (α%) to compare your test results.

2) Find $T(X)$. Know (or derive) the distribution of $T(X)$ under H_0.

3) Collect a sample of data $X = \{X_1, \ldots, X_n\}$.
Compute the test-statistics $T(X)$ used to test $H_0 \Rightarrow$ Report its *p-value*.

4) Rule: If *p-value* < α (say, 5%) ⟹ test result is *significant:* Reject H_0.
If the results are “*not significant*,” no conclusions are reached (no learning here).
⟹ Go back gather more data or modify model.

Review – Testing Only One Parameter

We are interested in testing a hypothesis about one parameter in our linear model: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$

1. Set H_0 and H_1 (about only one parameter): H_0: $\beta_k = \beta_k^0$

H_1: $\beta_k \neq \beta_k^0$.

2. Appropriate T(*X*): *t-statistic*. To derive the distribution of the test under H_0, we will rely on assumption (**A5**) $\varepsilon|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_T)$ (otherwise, results are only asymptotic).

Let b_k = OLS estimator of β_k

$SE[b_k|\mathbf{X}] = \text{sqrt}\{[s^2(\mathbf{X'X})^{-1}]_{kk}\} = s_{b,k}$

From assumption (**A5**), we know that

$b_k|\mathbf{X} \sim N(\beta_k, v_k^2)$ $\Rightarrow$ Under H_0: $b_k|\mathbf{X} \sim N(\beta_k^0, s_{b,k}^2)$.

$\Rightarrow$ Under H_0: $t_b = (b_k - \beta_k^0)/s_{b,k}|\mathbf{X} \sim t_{T-k}$.

3. Compute t_b, $\hat{t}$, using b_k, $\beta^0{}_k$, s, and $(\mathbf{X'X})^{-1}$. Get *p-value*($\hat{t}$).

4. Rule: Set an α level. If *p-value*(¤t⍰) $< \alpha$ $\Rightarrow$ Reject H_0: $\beta_k = \beta_k^0$

Alternatively, if $|\hat{t}| > t_{T-k,\alpha/2}$ $\Rightarrow$ Reject H_0: $\beta_k = \beta_k^0$.

Review – Testing Only One Parameter: *t-value*

Special case: H_0: $\beta_k = 0$

H_1: $\beta_k \neq 0$.

Then,

$t_k = b_k/\text{sqrt}\{s^2(\mathbf{X}'\,\mathbf{X})_{kk}^{-1}\} = b_k/SE[b_k] \Rightarrow t_k \sim t_{T-k}$.

In this case, we call t_k the *t-value* or *t-ratio.*

Usually, α = 5%, then if $|\hat{t}_k| > 1.96 \approx 2$, we say the coefficient b_k is "*significant*."

Review – Confidence Intervals

The goal of the *confidence intervals* (C.I.) is to set the coverage probability to equal a $(1 - \alpha)\%$ pre-specified target.

When we know the distribution of point estimate, it is easy to construct a C.I. Under the usual assumptions for b_k we have:

$C_n = [b_k - t_{T-k,\alpha/2}$ * Estimated SE(b_k), $b_k + t_{T-k,\alpha/2}$ * Estimated SE(b_k)]

This C.I. is symmetric around b_k: length is proportional to SE(b_k).

Usual α levels and $t_{T-k,\alpha/2}$ –when N > 30, (usual case) $t_{T-k,\alpha/2} \approx z_{\alpha/2}$)

α = 5%, then $z_{\alpha/2} = 1.96$.

α = 2%, then $z_{\alpha/2} = 2.33$.

α = 1%, then $z_{\alpha/2} = 2.58$.

Testing: The Expectation Hypothesis (EH)

Example: EH states that forward/futures prices are good predictors of future spot rates:

$$E_t[S_{t+T}] = F_{t,T}.$$

Implication of EH: $S_{t+T} - F_{t,T}$ = unpredictable.

That is, $E_t[S_{t+T} - F_{t,T}] = E_t[\varepsilon_t] = 0$!

Empirical tests of the EH are based on a regression:

$$(S_{t+T} - F_{t,T})/S_t = \alpha + \beta\, Z_t + \varepsilon_t, \qquad \text{(where } E[\varepsilon_t]=0)$$

where Z_t represents any economic variable that might have power to explain S_t, for example, $(i_d - i_f)$.

Then, under EH, H_0: $\alpha = 0$ and $\beta = 0$.
vs H_1: $\alpha \neq 0$ and/or $\beta \neq 0$.

- We will informally test EH using exchange rates (USD/GBP), 3-mo forward rates and 3-mo interest rates.

```
SF_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/SpFor_prices.csv", head=TRUE,
sep=",")
summary(SF_da)
x_date <- SF_da$Date
x_S <- SF_da$GBPSP
x_F3m <- SF_da$GBP3M
i_us3 <- SF_da$Dep_USD3M
i_uk3 <- SF_da$Dep_UKP3M
T <- length(x_S)
prem <- (x_S[-1] - x_F3m[-T])/x_S[-1]
int_dif <- (i_us3 - i_uk3)/100
y <- prem
x <- int_dif[-T]
fit_eh <- lm( y ~ x)
```

- We do two individual t-tests on α & β.

```
> summary(fit_eh)
Call:
lm(formula = y ~ x)

Residuals:
 Min       1Q    Median       3Q       Max
-0.125672 -0.014576 -0.000439  0.017356  0.094283

Coefficients:
```

```
              Estimate    Std. Error  t value  Pr(>|t|)
(Intercept)  -0.0001854  0.0016219  -0.114   0.90906      ⇒ constant not significant (|t|<2)
x            -0.2157540  0.0731553  -2.949   0.00339 **      ⇒ slope is significant (|t|>2). ⇒
Reject H0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02661 on 361 degrees of freedom
Multiple R-squared:  0.02353,   Adjusted R-squared:  0.02082
F-statistic: 8.698 on 1 and 361 DF,  p-value: 0.003393
```

- 95% C.I. for b:

$$C_n = [\ b_k - t_{k,\alpha/2} * \text{Estimated SE}(b_k),\ \ b_k + t_{k,\alpha/2} * \text{Estimated SE}(b_k)]$$

Then,

$$C_n = [\mathbf{-0.215754} - 1.96 * \mathbf{0.0731553}, \mathbf{-0.215754} + 1.96 * \mathbf{0.0731553}]$$
$$= [\mathbf{-0.3591384}, -0.07236961]$$

Since $\boldsymbol{\beta} = \mathbf{0}$ is not in C_n with 95% confidence ⇒ Reject H_0: $\beta_1 = 0$ at 5% level.

Note: The EH is a joint hypothesis, it should be tested with a joint test! ¶

The General Linear Hypothesis: Wald Statistic

Most of our test statistics, including joint tests, are Wald statistics.

Wald = normalized distance measure.

- One parameter:

$$t_b = (b_k - \beta^0{}_k)/s_{b,k} = \text{distance/unit}$$

- More than one parameter.

Let $\mathbf{z}$ = (random vector – hypothesized value) be the distance

$$W = \mathbf{z}' \, [\text{Var}(\mathbf{z})]^{-1} \, \mathbf{z} \qquad \text{(a quadratic form)}$$

- Distribution of W? We have a quadratic form.
  - – If $\mathbf{z}$ is normal and σ^2 known, $W \sim \chi^2_{\text{rank}(\text{Var}(\mathbf{z}))}$
  - – If $\mathbf{z}$ is normal and σ^2 unknown, $W \sim F$
  - – If $\mathbf{z}$ is not normal and σ^2 unknown, we rely on asymptotic theory, $W \xrightarrow{d} \chi^2_{\text{rank}(\text{Var}(\mathbf{z}))}$

The General Linear Hypothesis: H_0: $\mathbf{R}\beta - \mathbf{q} = \mathbf{0}$

- Suppose we are interested in testing J joint hypotheses.

Example: We want to test that in the 3 FF factor model that the SMB and HML factors have the same coefficients, $\beta_{SMB} = \beta_{HML} = \beta^0$.

We can write linear restrictions as H₀: $\mathbf{R}\beta - \mathbf{q} = \mathbf{0}$, where $\mathbf{R}$ is a Jxk matrix and $\mathbf{q}$ a Jx1 vector.

In the above example (J=2), we write:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \beta_1 \\ \beta_{Mkt} \\ \beta_{SMB} \\ \beta_{HML} \end{bmatrix} = \begin{bmatrix} \beta^0 \\ \beta^0 \end{bmatrix}$$

• Question: Is $\mathbf{Rb} - \mathbf{q}$ close to $\mathbf{0}$?
There are two different approaches to this question. Both have in common the property of unbiasedness for **b**.

(1) We base the answer on the discrepancy vector:

$$\mathbf{m} = \mathbf{Rb} - \mathbf{q}.$$

Then, we construct a Wald statistic:

$$W = \mathbf{m}' \,(\text{Var}[\mathbf{m}|\mathbf{X}])^{-1}\, \mathbf{m}$$

to test if **m** is different from **0**.

(2) We base the answer on a model loss of fit when restrictions are imposed: RSS must increase and R^2 must go down. Then, we construct an F test to check if the unrestricted RSS (RSS_U) is different from the restricted RSS (RSS_R).
• To test H₀, we calculate the discrepancy vector:

$$\mathbf{m} = \mathbf{Rb} - \mathbf{q}.$$

Then, we compute the Wald statistic:

$$W = \mathbf{m}' \,(\text{Var}[\mathbf{m}|\mathbf{X}])^{-1}\, \mathbf{m}$$

It can be shown that $\text{Var}[\mathbf{m}|\mathbf{X}] = \mathbf{R}[\sigma^2(\mathbf{X}'\mathbf{X})^{-1}]\mathbf{R}'$. Then,

$$W = (\mathbf{Rb} - \mathbf{q})' \,\{\mathbf{R}[\sigma^2(\mathbf{X}'\mathbf{X})^{-1}]\mathbf{R}'\}^{-1}\, (\mathbf{Rb} - \mathbf{q})$$

Under H₀ and assuming (**A5**) & estimating σ^2 with $s^2 = \mathbf{e}'\mathbf{e}/(T\text{-}k)$:

$$W^* = (\mathbf{Rb} - \mathbf{q})' \,\{\mathbf{R}[s^2(\mathbf{X}'\mathbf{X})^{-1}]\mathbf{R}\}^{-1}\, (\mathbf{Rb} - \mathbf{q})$$

$$\text{F} = W^*/J \sim F_{J,T\text{-}k}.$$

If (**A5**) is not assumed, the results are only asymptotic: $J * F \xrightarrow{d} \chi_J^2$

Example: We want to test that in the 3 FF factor model (T=569)

1. H₀: $\beta_{SMB} = 0.2$ and $\beta_{HML} = 0.6$.
   H₁: $\beta_{SMB} \neq 0.2$ and/or $\beta_{HML} \neq 0.6$. $\Rightarrow J = 2$

We define **R** (2x4) below and write $\mathbf{m} = \mathbf{R}\beta - \mathbf{q} = \mathbf{0}$:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \beta_1 \\ \beta_{Mkt} \\ \beta_{SMB} \\ \beta_{HML} \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.6 \end{bmatrix}$$

2. Test-statistic: $F = W^*/J = (\mathbf{Rb} - \mathbf{q})'\ \{\mathbf{R}[s^2(\mathbf{X}'\mathbf{X})^{-1}]\mathbf{R}'\}^{-1}\ (\mathbf{Rb} - \mathbf{q})$

Distribution under H_0: $F = W^*/2 \sim F_{2,T-2}$ (or asymptotic, $2^*F \xrightarrow{d} \chi^2_2$)

3. Get OLS results, compute F, $\hat{F}$.

4. <u>Decision Rule</u>: $\alpha = 0.05$ level. We reject H_0 if p-value(¤F⍰) < .05.

Or, reject H_0, if $\hat{F} > F_{J=2,T-2,.05}$.

```
J <- 2                                                  # number of restriction
fit_capm <- lm(ibm_x ~ Mkt_RF + SMB + HML)
b <- fit_capm$coefficients                              # Extract OLS coefficients
Var_b <- vcov(fit_capm)                                 # Extract Var[b]
R <- matrix(c(0,0,0,0,1,0,0,1), nrow=2)                 # matrix of restrictions
q <- c(.2, .6)                                          # hypothesized values
m <- R%*%b - q                                          # m = Estimated R*Beta - q
Var_m <- R %*% Var_b %*% t(R)                           # Variance of m
det(Var_m)                                              # check for non-singularity
W <- t(m)%*%solve(Var_m)%*%m
F_t <- as.numeric(W/J        )                                # F-test statistic

qf(.95, df1=J, df2=(T - k))                             # exact distribution (F-dist) if errors normal
p_val <- 1 - pf(F_t, df1=J, df2=(T - k))                # p-value(F_t) under errors normal
p_val

F_t_asym <- J*F                                         # Asymptotic F test (a Chi-square test)

qchisq(.95, df=J)                                       # asymptotic distribution (chi-square)
p_val <- 1 - pchisq(F_t_asym, df=J)                           # p-value(F_t) under asymptotic
distribution
p_val
```

```
> F_t
49.217
>
> qf(.95, df1=J, df2=(T - k))                           # exact distribution (F-dist) if errors normal
[1] 3.011644                             F_t > 3.011644⇒ reject H0 at 5% level
p_val <- 1 - pf(F_t, df1=J, df2=(T - k))                # p-value(F_t) under errors normal
> p_val
[1] < 2.2e-16   `                       ⇒ reject H0 at 5% level.

> F_t_asym
98.433
>
> qchisq(.95, df=J)                                     # asymptotic distribution (chi-square)
```

```
[1] 5.991465                    F_t > 5.991465 ⇒ reject H0 at 5% level
> p_val <- 1 - pchisq(F_t_asym, df=J)              # p-value(F_t) under asymptotic distribution
> p_val
 [1] < 2.2e-16                  ⇒ so low it is almost zero. Extremely low chance H0 is true.
```

• You can use the R package *car* to test linear restrictions (linear H_0).

```
install.packages("car")
library(car)
linearHypothesis(fit_capm, c("SMB = 0.2","HML = 0.6"), test="F")             # Exact F test

Linear hypothesis test

Hypothesis:
SMB = 0.2
HML = 0.6

Model 1: restricted model
Model 2: ibm_x ~ Mkt_RF + SMB + HML

Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    567 2.2691
2    565 1.9324  2   0.33667 49.217 < 2.2e-16 **            ⇒ reject H0 at 5% level
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


linearHypothesis(fit_capm, c("SMB = 0.2","HML = 0.6"), test="Chisq")   # Asymptotic F

Linear hypothesis test

Hypothesis:
SMB = 0.2
HML = 0.6

Model 1: restricted model
Model 2: ibm_x ~ Mkt_RF + SMB + HML

Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    567 2.2691
2    565 1.9324  2   0.33667 98.433  < 2.2e-16 ***          ⇒ reject H0 at 5% level
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Example: Now, we do a joint test of the EH. H_0: $\alpha = 0$ and $\beta = 0$.
Using the previous program but with:

```
J <- 2                                   # number of restriction
R <- matrix(c(1,0,0,1), nrow=2)          # matrix of restrictions
q <- c(0,0)                              # hypothesized values
> F_t
4.1024
>
> qf(.95, df1=J, df2=(T - k))            # exact distribution (F-dist) if errors normal
[1] 3.020661                        F_t > 3.020661 ⇒ reject H0 at 5% level
p_val <- 1 - pf(F_t, df1=J, df2=(T - k))           # p-value(F_t) under errors normal
> p_val
[1] 0.01731     `                   ⇒ reject H0 at 5% level.

> F_t_asym
8.2047
>
> qchisq(.95, df=J)                      # asymptotic distribution (chi-square)
[1] 5.991465                       F_t > 5.991465 ⇒ reject H0 at 5% level
> p_val <- 1 - pchisq(F_t_asym, df=J)              # p-value(F_t) under asymptotic distribution
> p_val
 [1] 0.01653     `                  ⇒ reject H0 at 5% level. ¶
```

The F Test: H_0: $R\beta - q = 0$

(2) We know that imposing the restrictions leads to a loss of fit. R^2 must go down. Does it go down a lot? –i.e., significantly?

Recall (i) $\mathbf{e^* = y - Xb^* = e - X(b^* - b)}$
(ii) $\mathbf{b^* = b - (X'X)^{-1}R'[R(X'X)^{-1}R']^{-1}(Rb - q)}$

$$\Rightarrow \qquad \mathbf{e^{*\prime}e^* = e'e + (b^* - b)'\, X'X\, (b^* - b)}$$

Replacing $(\mathbf{b^* - b})$ from (ii) in the above formula, we get:

$$\mathbf{e^{*\prime}e^* - e'e = (Rb - q)'[R(X'X)^{-1}R']^{-1}(Rb - q)}$$

Note: $\mathbf{e^{*\prime}e^* - e'e}$ is a quadratic form, then we can use a lot of results to derive its asymptotic distribution

• The F-distribution is a ratio of two independent $\chi^2_J$and χ^2_T RV divided by their degrees of freedom

$$F = \frac{\chi^2_J \,/\, J}{\chi^2_T \,/\, T} \sim F_{J,T}$$

Then, to get to the F-test, we rely on two results:

– $W = (\mathbf{Rb} - \mathbf{q})'\{\mathbf{R}[\sigma^2(\mathbf{X}'\mathbf{X})^{-1}]\mathbf{R}'\}^{-1}(\mathbf{Rb} - \mathbf{q}) \sim \chi^2_J$ (if σ^2 is known)

– $\mathbf{e}'\mathbf{e}/\sigma^2 \sim \chi^2_{T-k}$.

$\Rightarrow F = (\mathbf{e}^{*\prime}\mathbf{e}^* - \mathbf{e}'\mathbf{e})/J \,/\, [\mathbf{e}'\mathbf{e}/(T-k)] \sim F_{J,T-K}$.

• We can write the F-test in terms of R^2's. Let

R^2 = unrestricted model = 1 – RSS/TSS

R^{*2} = restricted model fit = 1 – RSS*/TSS

Then, dividing and multiplying F by TSS we get

$F = ((1 - R^{*2}) - (1 - R^2))/J \,/\, [(1 - R^2)/(T-k)] \sim F_{J,T-K}$

or

$F = \{ (R^2 - R^{*2})/J \} \,/\, [(1 - R^2)/(T-k)] \sim F_{J,T-K}$.

The F Test: H$_0$: F-test of Goodness of Fit

In the linear model

$\mathbf{y} = \mathbf{X}\,\beta + \varepsilon = \mathbf{X}_1\,\beta_1 + \mathbf{X}_2\,\beta_2 + \ldots + \mathbf{X}_k\,\beta_k + \varepsilon$

• We want to test if the slopes $\mathbf{X}_2, \ldots, \mathbf{X}_k$ are equal to zero. That is,

H_0: $\beta_2 = \cdots = \beta_k = 0$

H_1: at least one $\beta \neq 0$ $\Rightarrow J = k - 1$

We can write H$_0$: $\mathbf{R}\beta - \mathbf{q} = \mathbf{0}$ $\Rightarrow \begin{bmatrix} \mathbf{0} & \mathbf{1} & \ldots & \mathbf{0} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta_1} \\ \boldsymbol{\beta_2} \\ \ldots \\ \boldsymbol{\beta_k} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \ldots \\ \mathbf{0} \end{bmatrix}$

• We have $J = k - 1$. Then,

$F = \{ (R^2 - R^{*2})/(k - 1) \} \,/\, [(1 - R^2)/(T-k)] \sim F_{k-1,T-K}$.

For the restricted model, $R^{*2} = 0$. Then,

$F = \{ R^2/(k-1) \}/[(1 - R^2)/(T-k)] \sim F_{k-1,T-K}$

Recall ESS/TSS is the definition of R^2. RSS/TSS is equal to $(1 - R^2)$.

$$F(k-1, n-k) = \frac{R^2/(k-1)}{(1-R^2)/(T-k)} = \frac{\frac{ESS}{TSS}/(k-1)}{\frac{RSS}{TSS}/(T-k)} = \frac{ESS/(k-1)}{RSS/(T-k)}$$

This test statistic is called the *F-test of goodness of fit.*

Example: We want to test if all the FF factors (Market, SMB, HML) are significant, using monthly data 1973 – 2020 (T=569).

```
y <- ibm_x
T <- length(x)
```

```
x0 <- matrix(1,T,1)
x <- cbind(x0,Mkt_RF, SMB, HML)
k<- ncol(x)
b <- solve(t(x)%*% x)%*% t(x)%*%y              #OLS regression
e <- y - x%*%b
RSS <- as.numeric(t(e)%*%e)
R2 <- 1 - as.numeric(RSS)/as.numeric(t(y)%*%y)   #R-squared
> R2
[1] 0.338985

F_goodfit <- (R2/(k-1))/((1-R2)/(T-k))          #F-test of goodness of fit.
> F_goodfit
[1] 96.58204
```

⇒ F_goodfit > $F_{2,565,.05}$ = **2.387708** ⇒ Reject H_0. ¶

The F Test: General Case – Example

In the linear model

$$\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \boldsymbol{\beta}_1 + \mathbf{X}_2\,\boldsymbol{\beta}_2 + \mathbf{X}_3\,\boldsymbol{\beta}_3 + \mathbf{X}_4\,\boldsymbol{\beta}_4 + \boldsymbol{\varepsilon}$$

We want to test if the slopes $\mathbf{X}_3$, $\mathbf{X}_4$ are equal to zero. That is,

H_0: $\boldsymbol{\beta}_3 = \boldsymbol{\beta}_4 = \mathbf{0}$

H_1: $\boldsymbol{\beta}_3 \neq \mathbf{0}$ or $\boldsymbol{\beta}_4 \neq \mathbf{0}$ or both $\boldsymbol{\beta}_3$ and $\boldsymbol{\beta}_4 \neq \mathbf{0}$

We can use, $F = (\mathbf{e}^{*\prime}\mathbf{e}^{*} - \mathbf{e}'\mathbf{e})/J \,/\, [\mathbf{e}'\mathbf{e}/(T-k)] \sim F_{J,T-K}$.

Define $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon} = \boldsymbol{\beta}_1 + \mathbf{X}_2\,\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}$ (Restricted RSS = RSS_R, with k_R parameters)

$\mathbf{y} = \boldsymbol{\beta}_1 + \mathbf{X}_2\,\boldsymbol{\beta}_2 + \mathbf{X}_3\,\boldsymbol{\beta}_3 + \mathbf{X}_4\,\boldsymbol{\beta}_4 + \boldsymbol{\varepsilon}$ (Unrestricted RSS = RSS_U, with k_U parameters)

Then,

$$F(k-1, n-k) = \frac{\dfrac{RSS_R - RSS_U}{(k_U - k_R)}}{\dfrac{RSS_U}{(T - k_U)}}$$

The F Test: Are SMB and HML Priced Factors?

Example: We want to test if the additional FF factors (SMB, HML) are significant, using monthly data 1973 – 2020 (T=569).

Unrestricted Model:

(U) $IBM_{Ret} - r_f = \beta_0 + \beta_1\,(Mkt_{Ret} - r_f) + \beta_2\,SMB + \beta_3\,HML + \varepsilon$

Hypothesis: H_0: $\beta_2 = \beta_3 = 0$

H_1: $\beta_2 \neq 0$ and/or $\beta_3 \neq 0$

Then, the Restricted Model:

(R) $IBM_{Ret} - r_f = \beta_0 + \beta_1\,(Mkt_{Ret} - r_f) + \varepsilon$

Test: $F = \frac{(RSS_R - RSS_U)/J}{RSS_U/(T-k_u)} \sim F_{J,T\text{-}K}$. with $J = k_U - k_R = 4 - 2 = 2$

- The unrestricted model was already estimated in Lecture 3. For the restricted model:

```
y <- ibm_x
x0 <- matrix(1,T,1)
x_r <- cbind(x0,Mkt_RF)                          # Restricted X vector
k <- ncol(x)
T <- nrow(x)
k2 <- ncol(x)

b2 <- solve(t(x_r)%*% x_r)%*% t(x_r)%*%y         # Restricted OLS regression
e2 <- y – x_r%*%b2
RSS2 <- as.numeric(t(e2)%*%e2)

> RSS = 1.932442                                 # RSSU
> RSS2 = 1.964844                                # RSSR
> J <- k - k2                                    # J = degrees of freedom of numerator
> F_test <- ((RSS2 - RSS)/J)/(RSS/(T-k))
> F_test
[1] 4.736834
> qf(.95, df1=J, df2=(T-k))                      # F2,565,.05 value (≈ 3)
[1] 3.011672                                ⇒ Reject H0.
> p_val <- 1 - pf(F_test, df1=J, df2=(T-k))      # p-value of F_test
> p_val
[1] 0.009117494                                  ⇒ p-value is small ⇒ Reject H0.
```

- There is package in R, *lmtest,* that performs this test, *waldtest,* (and many others, used in this class). You need to install it first: install.packages("lmtest").

Note: The models need to be nested. For the *waldtest*, the default reports the *F-test* with the F distribution.

Example: We test if the additional FF factors (SMB, HML) are significant, using monthly data 1973 – 2020 (T=569).

```
library(lmtest)
fit_wU <- lm (y ~ Mkt_RF + SMB + HML)
fit_wR <- lm (y ~ Mkt_RF)
waldtest(fit_wU, fit_wR)
Wald test

Model 1: y ~ Mkt_RF + SMB + HML
Model 2: y ~ Mkt_RF
  Res.Df Df      F   Pr(>F)
1    565
```

```
2    567 -2 4.7368 0.009117 **                        ⇒ p-value is small: Reject H0. ¶
```

Trilogy of Asymptotic Tests: LR, Wald, and LM

In practice, we tend to rely on the asymptotic distribution of the Wald test. That is, $W \xrightarrow{d} \chi_J^2$.

There are two other popular tests that are asymptotically equivalent –i.e., they have the same asymptotic distribution: the Likelihood Ratio (LR) and the Lagrange Multiplier (LM) tests.

• The LR is based on the (log) Likelihood. It needs two ML estimations: the unrestricted estimation, producing $\hat{\theta}_{ML}$, and the restricted estimation, producing $\hat{\theta}^R$. Below we define the LR test:

$$LR = 2[\log(L(\hat{\theta}_{ML})) - \log(L(\hat{\theta}^R))] \xrightarrow{\text{d}} \chi_J^2$$

Note: MLE requires assuming a distribution, usually, a normal.

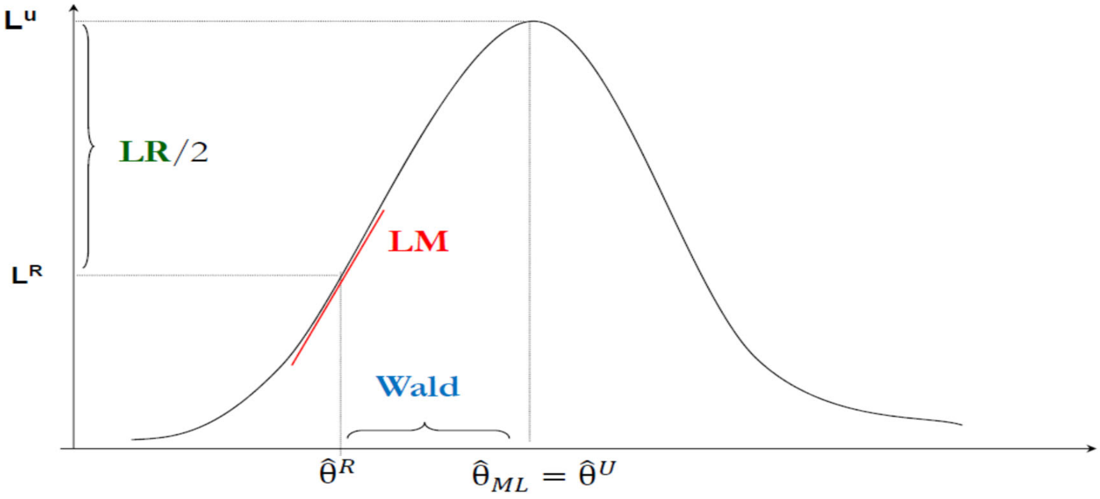
Technical note: The LR test is a *consistent test*. An asymptotic test which rejects H0 with probability one when the H1 is true is called a *consistent test.* That is, a consistent test has asymptotic power of 1. The LR test is a consistent test.

```
library(lmtest)
fit_wU <- lm (y ~ Mkt_RF + SMB + HML)
fit_wR <- lm (y ~ Mkt_RF)
waldtest(fit_wU, fit_wR)
Wald test

Model 1: y ~ Mkt_RF + SMB + HML
Model 2: y ~ Mkt_RF
  Res.Df Df      F   Pr(>F)
1    565
2    567 -2 4.7368 0.009117 **                        ⇒ p-value is small: Reject H0. ¶
```

• The LM test needs only one estimation: the restricted estimation, producing $\hat{\theta}^R$. If the restriction is true, then the slope of the objective function (say, the Likelihood) at $\hat{\theta}^R$should be zero. The slope is called the Score, S($\hat{\theta}^R$). The LM test is based on a Wald test on S($\hat{\theta}^R$) = 0.

$$LM = S(\hat{\theta}^R)'[Var(S(\hat{\theta}^R)]^{-1}S(\hat{\theta}^R) \xrightarrow{\text{d}} \chi_J^2$$

It turns out that there is a much simpler formulation for the LM test, based on the residuals of the restricted model. We will present this version of the test in Lecture 6.



If the likelihood function were quadratic then LR = LM = W. In general, however, W > LR > LM.

Testing Remarks: Pre-testing

A special case of omitted variables.

- First, a researcher starts with an unrestricted model (U):

$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. (U)

- Then, based on ("preliminary") tests –say, an *F-test*- a researcher decides to use restricted estimator, **b***. That is,

$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. s.t. $\mathbf{R}\boldsymbol{\beta} = \mathbf{q}$ (R)

- We can think of the estimator we get from estimating R as:

$\mathbf{b}_{PT} = I_{\{0,c\}}(F)\,\mathbf{b}^* + I_{\{c,\infty\}}(F)\,\mathbf{b}$,

where $I_{\{0,c\}}$ is an indicator function:

$I_{\{c,\infty\}}(F) = 0$, if *F-stat* in *R (rejection region)* –say, $F > c$,

$I_{\{0,c\}}(F) = 1$ if *F-stat* in R^C –say, $F < c$.

c : critical value chosen for testing H0: $\mathbf{R}\boldsymbol{\beta} = \mathbf{q}$ **,** using the *F-stat.*

• The *pre-test estimator* is a rule, which chooses between the restricted estimator, **b***, or the OLS estimator, **b**:

$\mathbf{b}_{PT} = I_{\{0,c\}}(F)\,\mathbf{b}^* + I_{\{c,\infty\}}(F)\,\mathbf{b}$,

where $\mathbf{b}^* = \mathbf{b} - (\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'[\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}']^{-1}(\mathbf{R}\mathbf{b} - \mathbf{q})$

• Two "negative" situations:

(1) H_0: $\mathbf{R}\boldsymbol{\beta} = \mathbf{q}$ is true. The *F-test* will incorrectly reject H_0 α% of the time. That is, in α% of the repeated samples, we have "irrelevant variables"

⇒ OLS **b:** No bias, but inefficient estimator.

(2) H_0: $\mathbf{R}\boldsymbol{\beta} = \mathbf{q}$ is false. The *F-test* will correctly reject H_0 a % of times equal to the power π of the test. That is, (100 – π)% of the time, $\mathbf{R}\boldsymbol{\beta}$=$\mathbf{q}$ will be incorrectly imposed, we have "omitted variables:"

⇒ OLS **b***: bias, but small variance!

The failure of the OLS estimator to have the properties under correct specification is called *pre-test bias*.

Pre-testing (also called *sequential estimation*, *data mining*) is common in practice. In general, it is ignored –and not even acknowledged.

Main argument to ignore pre-testing: We need some assumptions to decide which variables are included in a model. Is the probability that pre-testing yields an incorrect set of $\boldsymbol{X}$ greater than the probability of selecting the "correct" assumption?

David Hendry, a well known thinker of these methodological issues, does not see pre-testing in the discovery stage as a problem. For him, pre-testing at that stage is part of the *process of discovery*.

Practical advise: Be aware of the problem. Do not rely solely on stats to select a model –use economic theory as well.

- Do not use same sample evidence to generate an H_0 and to test it!

Example: The Fama-French factors have been "discovered" using the CRSP/Compustast database for a long, long time. Thus, testing the Fama-French factors using the CRSP/Compustat is not advisable!

(You can test them with another dataset, for example, get international data.) ¶

Testing Remarks: Significance level, α

So far, we have assumed that the distribution of the test statistic –say the *F*-statistic– under H_0 is known exactly, so that we have what is called an *exact test***.**

Technically, the *size of a test* is the supremum of the rejection probability over all DGPs that satisfy H_0. For an exact test, the size equals the *nominal level*, α –i.e., the Prob[Type I error] = α.

Usually, the distribution of a test is known only approximately *(asymptotically*). In this case, we need to draw a distinction between the nominal level, α (*nominal size*), of the test & the actual *rejection probability (empirical size*), which may differ greatly from the nominal level.

Simulations are needed to gauge the empirical size of tests.

Testing Remarks: A word about α

Ronald Fisher, before computers, tabulated distributions. He used a .10, .05, and .01 percentiles. These tables were easy to use and, thus, those percentile became the de-facto standard α for testing H_0.

"It is usual and convenient for experimenters to take 5% as a standard level of significance." – Fisher (1934).

Given that computers are powerful and common, why is $p = 0.051$ unacceptable, but $p = 0.049$ is great? There is no published work that provides a theoretical basis for the standard thresholds.

Rosnow and Rosenthal (1989): " ... surely God loves .06 nearly as much as .05."

Practical advise: In the usual Fisher's null hypothesis (significance) testing, significance levels, α, are arbitrary. Make sure you pick one, say 5%, and stick to it throughout your analysis or paper.

• Report *p-values*, along with CI's. Search for *economic significance*.

Questions: .10, .05, or .01 significance?
Many tables will show *, **, and *** to show .10, .05, and .01 significance levels –for example, lm() in R. Throughout the paper, the authors will point out the different significance levels. In these papers, it is not clear what α is the paper using for inference.

We can think of these stars (or *p-values*) as ways of giving weights to H_0 relative to H_1.

Testing Remarks: A word about H_0

In applied work, we only learn when we reject H_0; say, when the *p*-value < α. But, rejections are of two types:
- Correct ones, driven by the power of the test
- Incorrect ones, driven by Type I Error ("*statistical accident*," luck).

It is important to realize that, however small the *p*-value, there is always a finite chance that the result is a pure accident. At the 5% level, there is 1 in 20 chances that the rejection of H_0 is just luck.

Since negative results are difficult to publish (*publication bias*), there is an unknown but possibly large number of false claims taken as truths.

Example: If $\alpha = 0.05$, proportion of false H_0=10%, and $\pi = .50$, **47.4%** of rejections are true H_0 -i.e., "false positives." ¶

Testing Remarks: Mass significance

We have a model. We perform *k* different tests, say *k t-tests,* each with a *nominal significance level* of α:

α = Prob (Rejecting for a given test |H_0 for this test is true)

The *overall significance* of the test procedure is, however, given by
α^* = Prob (Rejecting at least one test | all H_0 are true).

The probability of rejecting at least one H_0 is obviously greater than of rejecting a specific test. This is the problem of *mass significance*.

• Two cases
(1) Independent tests $\alpha^* = 1 - (1 - \alpha)^k$ & $\alpha = 1 - (1 - \alpha^*)^{1/k}$

(2) Dependent tests: $\alpha^* \leq k\alpha$ & $\alpha \geq \alpha^*/k$
$\Rightarrow$ close to the "independent" values for small α, but can differ for large α.

Example:	α =0.05 and k=5	$\Rightarrow$ α^*(Indep) =.23 & α^*(Dep)=.25
	α =0.05 and k=20	$\Rightarrow$ α^*(Indep) =.64 & α^*(Dep) = 1
	α^* =0.05 and k=5	$\Rightarrow$ α(Indep) =.0102 & α(Dep) =.01
	α^* =0.20 and k=5	$\Rightarrow$ α(Indep) =.044 & α(Dep) =.04
	α^* =0.20 and k=20	$\Rightarrow$ α(Indep) =.011 & α(Dep) =.01. ¶

• David Hendry's suggestions:
In repeated *parametric testing* (overall level 5%):
- Only accept variables as important when their *p-values* are less than 0.001, preferably smaller
- Maybe look for other ways of choosing variables, say AIC.

In repeated *diagnostic testing* (overall level 20%), we should only accept there is no misspecification if
- All *p-values* are greater than 0.05, *or*
- Most *p-values* are greater than 0.10 with a few in the range 0.02 to 0.10

Non-nested Models and Tests

So far, all our tests (t-, F- & Wald tests) have been based on nested models, where the R model is a restricted version of the U model.

Example:

Model U $\mathbf{Y = X\beta + W\delta + \varepsilon}$ (Unrestricted)
Model R $\mathbf{Y = X\beta + \xi}$ (Restricted)

Model U becomes Model R under H_0: $\boldsymbol{\delta} = \mathbf{0}$.¶

• Sometimes, we have two rival models to choose between, where neither can be nested within the other -i.e., neither is a restricted version of the other.

Example:

Model 1 $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\delta} + \boldsymbol{\varepsilon}$

Model 2 $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\xi}$**.** ¶

If the dependent variable is the same in both models (as is the case here), we can simply use Adjusted-R^2 to rank the models and select the one with the largest Adjusted-R^2.
We can also use AIC and/or BIC.

• Alternative approach: Encompassing
(1) Form a composite or *encompassing* model that nests both rival models −say, Model 1 and Model 2. This is the unrestricted Model (ME).
(2) Test the relevant restrictions of each rival model against ME. We do two F-tests, where the restricted models are Model 1 and Model 2.

Assuming the restrictions cannot be rejected, we prefer the model with the lower F statistic for the test of restrictions.

Note: We test a hybrid model. Also, multicollinearity may appear.

Example: We have:

Model 1 $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\delta} + \boldsymbol{\varepsilon}$

Model 2 $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\xi}$

Then, the Encompassing Model (ME) is:

ME: $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{W}\boldsymbol{\delta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$

Now test, separately, the hypotheses (1) $\boldsymbol{\delta} = \mathbf{0}$ and (2) $\boldsymbol{\gamma} = \mathbf{0}$. That is,

F-test for H_0: $\boldsymbol{\gamma} = \mathbf{0}$: ME (U Model) vs Model 1 (R Model).
F-test for H_0: $\boldsymbol{\delta} = \mathbf{0}$: ME (U Model) vs Model 2 (R Model). ¶

Non-nested Models and Tests: IFE or PPP?

Example: What drives log changes in exchange rates for the USD/GBP (e): $(i_d - i_f)$ or $(I_d - I_f)$?

Model 1 (IFE): $\mathbf{e} = \alpha^1 + \beta^1\, (\mathbf{i_d} - \mathbf{i_f}) + \boldsymbol{\varepsilon}^1$

Model 2 (PPP): $\mathbf{e} = \alpha^2 + \beta^2\, (\mathbf{I_d} - \mathbf{I_f}) + \boldsymbol{\varepsilon}^2$

```
SF_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/SpFor_prices.csv", head=TRUE,
sep=",")
x_date <- SF_da$Date
x_S <- SF_da$GBPSP
x_F3m <- SF_da$GBP3M
i_us3 <- SF_da$Dep_USD3M
i_uk3 <- SF_da$Dep_UKP3M
cpi_uk <- SF_da$UK_CPI
cpi_us <- SF_da$US_CPI
T <- length(x_S)
```

```
int_dif <- (i_us3[-1] - i_uk3[-1])/100
lr_usdgbp <- log(x_S[-1]/x_S[-T])
I_us <- log(cpi_us[-1]/cpi_us[-T])
I_uk <- log(cpi_uk[-1]/cpi_uk[-T])
inf_dif <- (I_us - I_uk)
```

Encompassing Model (ME or "Unrestricted Model")

$$\mathbf{e = \alpha + \beta_1\,(i_d - i_f) + \beta_2\,(I_d - I_f) + \varepsilon^1}$$

```
# Encompassing Model and Test
fit_me <- lm(lr_usdgbp ~ int_dif + inf_dif)                    # ME estimation
> summary(fit_me)

Coefficients:
             Estimate   Std. Error  t value  Pr(>|t|)
(Intercept)  -0.0009633 0.0016210   -0.594   0.5527
int_dif      -0.0278510 0.0741189   -0.376   0.7073          ⇒ cannot reject H0: β1 = 0.
inf_dif       0.7444711 0.3429106   2.171    0.0306 *        ⇒ reject H0: β2 = 0.
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.02662 on 360 degrees of freedom
Multiple R-squared:  0.01316,   Adjusted R-squared:  0.007673
F-statistic: 2.399 on 2 and 360 DF,  p-value: 0.09221
```

Note: Two F-tests are needed, but for the one variable case, the t-tests are equivalent.

- The package in R, *lmtest,* performs this test, *encomptest*. Recall you need to install it first: install.packages("lmtest").

Note: The test reported is an *F*-test ~ $F_{1,T-k}$, which, in this case with only one variable in each Model, is equal to $(t_{T-k})^2$

```
> library(lmtest)
> fit_m1 <- lm(lr_usdgbp ~ int_dif)                  # Restricted Model 1
> fit_m2 <- lm(lr_usdgbp ~ inf_dif)                  # Restricted Model 2
> encomptest(fit_m1, fit_m2)
 1: lr_usdgbp ~ int_dif
Model 2: lr_usdgbp ~ inf_dif
Model E: lr_usdgbp ~ int_dif + inf_dif
          Res.Df Df     F  Pr(>F)
M1 vs. ME    360 -1 4.7134 0.03058 *           ⇒ reject H0: β2 = 0. Check: (2.171)² = 4.713
M2 vs. ME    360 -1 0.1412 0.70732.          ¶
```

Non-nested Models: *J*-test

We present the most popular test for non-nested models, the Davidson-MacKinnon (1981)'s *J*-test.

We start with two non-nested models. Say,

Model 1: $\mathbf{Y = X\beta} + \varepsilon$

Model 2: $\mathbf{Y} = \mathbf{Z}\gamma + \xi$

<u>Idea</u>: If Model 2 is true, then the fitted values from the Model 1, when added to the 2nd equation, should be insignificant.

• Steps:
(1) Estimate **Model 1** ⇒ obtain fitted values: $\mathbf{Xb}$.
(2) Add $\mathbf{Xb}$ to the list of regressors in Model 2

$$\Rightarrow \mathbf{Y} = \mathbf{Z}\gamma + \lambda\mathbf{Xb} + \xi$$

(3) Do a *t-test* on λ. A significant *t*-value would be evidence against Model 2, in favour of **Model 1**.
(4) Repeat the procedure for the models the other way round.
(4.1) Estimate **Model 2** ⇒ obtain fitted values: $\mathbf{Zc}$.
(4.2) Add $\mathbf{Zc}$ to the list of regressors in Model 1:

$$\Rightarrow \mathbf{Y} = \mathbf{X}\beta + \lambda\, \mathbf{Zc} + \varepsilon$$

(4.3) Do a *t-test* on λ. A significant *t*-value would be evidence against **Model 1** and in favour of **Model 2**.

(5) Rank the models on the basis of this test.

• It is possible that we cannot reject both models. This is possible in small samples, even if one model, say Model 2, is true.

It is also possible that both *t-tests* reject H_0 ($\lambda \neq 0$ & $\lambda \neq 0$). This is not unusual. McAleer's (1995), in a survey, reports that out of 120 applications all models were rejected 43 times.

<u>Technical Note</u>: As some of the regressors in step (3) are stochastic, Davidson and MacKinnon (1981) show that the *t-test* is *asymptotically* valid.

• One would also want to examine the diagnostic test results when choosing between two models.

Non-nested Models: *J*-test – IFE or PPP?

Example: Now, we test Model 1 vs Model 2, using the *J*-test.

Model 1 (IFE): $\mathbf{e = \alpha^1 + \beta^1 (i_d - i_f) + \varepsilon^1}$

Model 2 (PPP): $\mathbf{e = \alpha^2 + \beta^2 (I_d - I_f) + \varepsilon^2}$

```
y <- lr_usdgbp
fit_m1 <- lm( y ~ int_dif)
summary(fit_m1)
y_hat1  <- fitted(fit_m1)
fit_J1 <- lm( y ~ inf_dif + y_hat1)
summary(fit_J1)

fit_m2 <- lm( y ~ inf_dif)
summary(fit_m2)
y_hat2  <- fitted(fit_m2)
fit_J2 <- lm( y ~ int_dif + y_hat2)
summary(fit_J2)
```

```
> fit_m1 <- lm( y ~ int_dif)
> y_hat1  <- fitted(fit_m1)
> fit_J1 <- lm(formula = y ~ inf_dif + y_hat1)
> summary(fit_J1)

Residuals:
      Min        1Q    Median       3Q       Max
-0.136310 -0.014168  0.000351  0.017227  0.092421
Coefficients:
              Estimate   Std. Error  t value Pr(>|t|)
(Intercept)  0.0001497   0.0025556   0.059   0.9533
inf_dif      0.7444711   0.3429106   2.171   0.0306 *
y_hat1       1.2853298   3.4206106   0.376   0.7073
```

⇒ cannot reject H_0: λ=0. (Good for Model 2)

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02662 on 360 degrees of freedom
Multiple R-squared:  0.01316,   Adjusted R-squared:  0.007673
F-statistic: 2.399 on 2 and 360 DF,  p-value: 0.09221


> fit_m2 <- lm( y ~ inf_dif)
> y_hat2  <- fitted(fit_m2)
> fit_J2 <- lm(formula = y ~ int_dif + y_hat2)
> summary(fit_J2)
Residuals:
      Min        1Q    Median       3Q       Max
-0.136310 -0.014168  0.000351  0.017227  0.092421
Coefficients:
             Estimate  Std. Error       t value    Pr(>|t|)
(Intercept) -0.000304  0.0016409        -0.186     0.8529
int_dif     -0.027851  0.0741189        -0.376     0.7073
y_hat2      1.0066945  0.4636932        2.171      0.0306 *
```

⇒ *R*eject H_0: λ=0. (Again, good for Model 2)

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02662 on 360 degrees of freedom
Multiple R-squared:  0.01316,   Adjusted R-squared:  0.007673
F-statistic: 2.399 on 2 and 360 DF,  p-value: 0.09221
```

- The *lmtest* package also performs this test. Recall that you need to install it first: install.packages(“lmtest”).

```
library(lmtest)
fit_m1 <- lm(lr_usdgbp ~ int_dif)
fit_m2 <- lm(lr_usdgbp ~ inf_dif)

> jtest(fit_m1, fit_m2)
J test
```

```
Model 1: lr_usdgbp ~ int_dif
Model 2: lr_usdgbp ~ inf_dif
                  Estimate Std. Error t value  Pr(>|t|)
M1 + fitted(M2)   1.0067    0.4637     2.1710   0.03058 *            ⇒ Reject H0:
λ=0. (Model 2 selected)
M2 + fitted(M1)   1.2853    3.4206     0.3758   0.70732.      ¶
```

Non-nested Models: *J*-test – Application

We want to test

$\mathbf{H_0}$: $\boldsymbol{y = X\beta + \varepsilon_0}$ (additive) vs

$\mathbf{H_1}$: $\ln \boldsymbol{y} = (\ln \boldsymbol{X})\, \boldsymbol{\gamma} + \boldsymbol{\varepsilon}_1$ (multiplicative)

We look at the *J*-test

- Step 1: OLS on $\mathbf{H_1}$: get $\hat{\boldsymbol{\gamma}}$
  OLS $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\lambda_1} \exp\{\ln(\mathbf{X})\, \hat{\boldsymbol{\gamma}}\} + \boldsymbol{\varepsilon}$ ⇒ *t-test* on $\boldsymbol{\lambda_1}$
- Step 2: OLS on $\mathbf{H_0}$: get $\mathbf{b}$
  OLS $\ln \mathbf{y} = (\ln \boldsymbol{X})\, \boldsymbol{\gamma} + \boldsymbol{\lambda_0}\, \mathbf{Xb} + \boldsymbol{\varepsilon}$ ⇒ *t-test* on $\boldsymbol{\lambda_0}$

Situations:
(1) Both OK: $\lambda_1 = 0$ and $\lambda_0 = 0$ ⇒ get more data
(2) Only 1 is OK: $\lambda_1 \neq 0$ and $\lambda_0 = 0$ (multiplicative is OK);
$\lambda_0 \neq 0$ and $\lambda_1 = 0$ (additive is OK)
(3) Both rejected: $\lambda_1 \neq 0$ and $\lambda_0 \neq 0$ ⇒ new model is needed.

Non-nested Models: *J*-test – Remarks

The *J*-test was designed to test non-nested models (one model is the true model, the other is the false model), not for choosing competing models –the usual use of the test.

The *J*-test is likely to over reject the true (model) hypothesis when one or more of the following features is present:
i) A poor fit of the true model
ii) A low/moderate correlation between the regressors of the 2 models
iii) The false model includes more regressors than the correct model.

Davidson and MacKinnon (2004) state that the *J*-test will over-reject, *often quite severely* in finite samples when the sample size is small or where conditions (i) or (iii) above are obtained.

Lecture 6 – Specification, Forecasting & Model Selection

OLS Estimation - Assumptions

Brief Review of CLM Assumptions
(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) $E[\varepsilon|\mathbf{X}] = 0$
(**A3**) $Var[\varepsilon|\mathbf{X}] = \sigma^2\, \mathbf{I}_T$
(**A4**) $\mathbf{X}$ has full column rank –rank($\mathbf{X}$)=k-, where $T \geq k$.

Question: What happens when (**A1**) is not correctly specified?

First, we look at (**A1**), in the context of linearity. Are we omitting a relevant regressor? Are we including an irrelevant variable? What happens when we impose restrictions in the DGP?

Second, in (**A1**), we allow some non-linearities in its functional form.

Specification Errors: Omitted Variables

Omitting relevant variables: Suppose the correct model (DGP) is
$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}$ –the "long regression," with $\mathbf{X}_1$ & $\mathbf{X}_2$.

But, we compute OLS omitting $\mathbf{X}_2$. That is,
$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}$ –the "short regression."

We have two *nested* models: one model becomes the other, once a restriction is imposed. In the above case, the true model becomes "the short regression" by imposing the restriction $\boldsymbol{\beta}_2 = 0$.

Question: What are the implications of using the wrong model, with omitted variables?
We already know the answer, we are imposing a wrong restriction: the restricted estimator, **b***, is biased, but it is more efficient.

Specification Errors: Omitted Variables

Some easily proved results:
$E[\mathbf{b}_1|\mathbf{X}] = E\,[(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\,\mathbf{y}] = E\,[(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\,(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon})]$
$= \square_1 + (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2\boldsymbol{\beta}_2 \neq \boldsymbol{\beta}_1.$

Thus, unless $\mathbf{X}_1'\mathbf{X}_2 = 0$, $\mathbf{b}_1$ is *biased*. The bias can be **huge**. It can reverse the sign of a price coefficient in a "demand equation."

(2) $Var[\mathbf{b}_1|\mathbf{X}] \leq Var[\mathbf{b}_{1.2}|\mathbf{X}]$, where $\mathbf{b}_{1.2}$ is the OLS estimator of $\boldsymbol{\beta}_1$ in the long regression (the true model).

Thus, we get a smaller variance when we omit $\mathbf{X}_2$.

Interpretation: Omitting $\mathbf{X}_2$ amounts to using extra information –i.e., $\boldsymbol{\beta}_2 = \mathbf{0}$. Even if the information is wrong, it reduces the variance.

(3) Mean Squared Error (MSE = RSS/T)

If we use MSE as precision criteria for selecting an estimator, $\mathbf{b}_1$ may be more "precise."
Precision = Mean squared error (MSE)
= Variance + Squared bias.
Smaller variance but positive bias. If bias is small, a practitioner may still favor the short regression.

Note: Suppose $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$. Then the bias goes away. Interpretation, the information is not "right," it is irrelevant: $\mathbf{b}_1$ is the same as $\mathbf{b}_{1.2}$.

Example: We fit an ad-hoc model for U.S. short-term interest rates ($i_{US,t}$) that includes inflation rate ($I_{US,t}$), changes in the USD/EUR (e_t), money growth rate ($m_{US,t}$), and unemployment ($u_{US,t}$), using monthly data from 1975:Jan-2020: Jul. That is,

$$i_{US,t} = \beta_0 + \beta_1 I_{US,t} + \beta_2 e_t + \beta_3 m_{US,t} + \beta_4 u_{US,t} + \varepsilon_i$$

```
Fger_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/FX_USA_GER.csv", head=TRUE,
sep=",")
us_CPI <- Fger_da$US_CPI
us_M1 <- Fger_da$US_M1
us_i <- Fger_da$US_I3M
us_GDP <- Fger_da$US_GDP
ger_CPI <- Fger_da$GER_CPI
us_u <- Fger_da$US_UN
S_ger <- Fger_da$USD_EUR

T <- length(us_CPI)
us_I <- log(us_CPI[-1]/us_CPI[-T])  # US Inflation: (Log) Changes in CPI
us_mg <- log(us_M1[-1]/us_M1[-T])           # US Money Growth: (Log) Changes in M1
e_ger <- log(S_ger[-1]/S_ger[-T])           #  (Log) Changes in USD/EUR
us_i_1 <- us_i[-1]                          # Adjust sample size of untransformed data
us_u_1 <- us_u[-1]                          # Adjust sample size of untransformed data
us_i_0 <- us_i[-T]                          # lagged interest rates, by removing T observation
xx_i <- cbind(us_I ,e_ger, us_mg, us_u_1)   # X matrix
fit_i <- lm(us_i_1 ~ xx_i)
>  summary(fit_i)
```

```
Coefficients:
             Estimate   Std. Error t value   Pr(>|t|)
(Intercept)  2.12516    0.52177    4.073 5.34e-05 ***
xx_i_us_I    410.03733  37.17344   11.030   < 2e-16 ***
xx_i_e_ger   8.90564    4.59915    1.936  0.053343 .
xx_i_us_mg  -50.07811   15.04907  -3.3280.000935 ***      ⇒ significant.
```

```
xx_i_us_u_10.22673      0.08346   2.717  0.006805 **       ⇒ significant.
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.113 on 542 degrees of freedom
Multiple R-squared:  0.2276,    Adjusted R-squared:  0.2219
F-statistic: 39.93 on 4 and 542 DF,  p-value: < 2.2e-16
```

- Now, we include lagged interest rates

```
xx_i <- cbind(us_I ,e_ger, us_mg, us_u_1, us_i_0)   # X matrix with lagged interest rates
fit_i <- lm(us_i_1 ~ xx_i)
summary(fit_i)

Coefficients:
              Estimate    Std. Error  t value Pr(>|t|)
(Intercept)   0.101007    0.079458   1.271  0.20420
xx_ius_I      16.367138 6.144709   2.664  0.00796 **
xx_ie_ger     3.112901    0.691673  4.501  8.3e-06 ***
xx_ius_mg     1.231633    2.284528  0.539  0.59003        ⇒ now, not significant.
xx_ius_u_1   -0.015444    0.012632  -1.2230.22199        ⇒ now, not significant.
xx_i_us_i_0 0.22673      0.08346      2.717  0.00681 **     ⇒ significant effect on other coeff.
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.113 on 542 degrees of freedom
Multiple R-squared:  0.2276,    Adjusted R-squared:  0.2219
```

Note: Lagged i_{US} ($i_{US,\,t-1}$) is very significant & changes significance of other variables. It may point out to a general misspecification in (**A1**). ¶

Omitted Variables Example: Gasoline Demand

We have a linear model for the demand for gasoline (G) as function of price (PG) and income (Y):

$$\mathbf{G} = PG\,\beta_1 + Y\,\beta_2 + \varepsilon,$$

Q: What happens when you wrongly exclude Income (Y)?

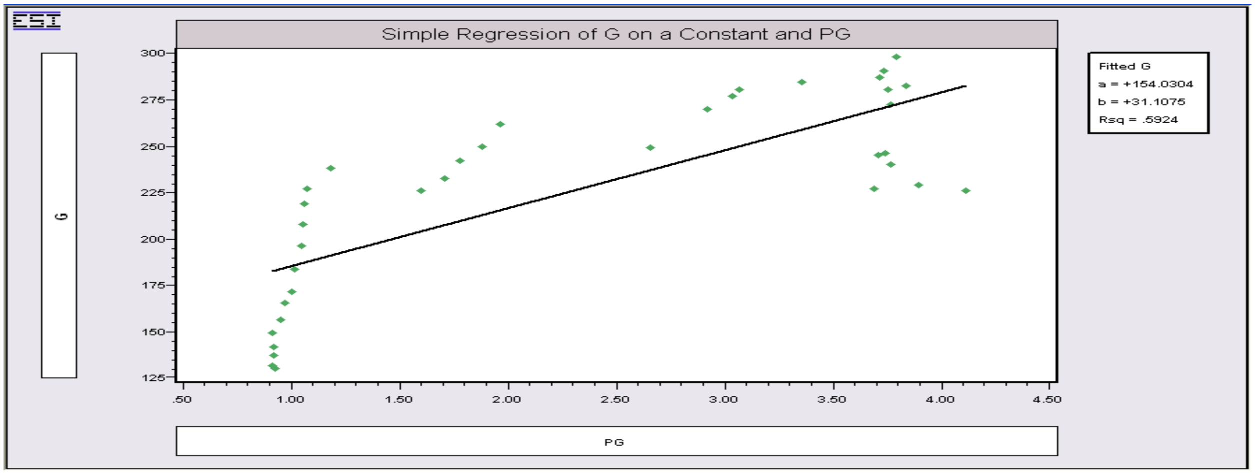
$$E[b_1|\mathbf{X}] = \beta_1 + \qquad\qquad\qquad \beta_2$$

In time series data, $\beta_1 < 0,\ \beta_2 > 0$ (usually)
Cov[*Price*, *Income*] > 0 in time series data.

⇒ The short regression will overestimate the price coefficient.

In a simple regression of G (demand) on a constant and PG, the Price Coefficient (β_1) should be negative.

Example: Estimation of a 'Demand' Equation: Shouldn't the Price Coefficient be Negative? Taken from Green's graduate Econometrics textbook



- If a multiple regression is done, incorporating income, Y, theory works!

```
Ordinary      least squares regression ............
LHS=G         Mean                 =      226.09444
              Standard deviation   =       50.59182
              Number of observs.   =             36
Model size    Parameters           =              3
              Degrees of freedom   =             33
Residuals     Sum of squares       =     1472.79834
              Standard error of e  =        6.68059
Fit           R-squared            =         .98356
              Adjusted R-squared   =         .98256
Model test    F[  2,    33] (prob) =    987.1(.0000)
--------+---------------------------------------------------------
Variable| Coefficient    Standard Error  t-ratio  P[|T|>t]
--------+---------------------------------------------------------
Constant|   -79.7535***      8.67255       -9.196   .0000
       Y|     .03692***       .00132       28.022   .0000
      PG|   -15.1224***      1.88034       -8.042   .0000
--------+---------------------------------------------------------
```

Note: Income is helping us to identify a demand equation –i.e., with a negative slope for the price variable. ¶

Specification Errors: Irrelevant Variables

Irrelevant variables. Suppose the correct model is

$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\varepsilon}$ –the "short regression," with $\mathbf{X}_1$

But, we estimate

$\mathbf{y} = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\varepsilon}$ –the "long regression."

Some easily proved results: Including irrelevant variables just reverse the omitted variables results: It increases variance -the cost of not using information-; but does not create biases.

⇒ Since the variables in $\mathbf{X}_2$ are truly irrelevant, then $\boldsymbol{\beta}_2 = \mathbf{0}$,
so $E[\mathbf{b}_{1.2}|\mathbf{X}] = \boldsymbol{\beta}_1$.

• A simple example
Suppose the correct model is: $\mathbf{y} = \beta_1 + \beta_2\,\mathbf{X}_2 + \boldsymbol{\varepsilon}$
But, we estimate: $\mathbf{y} = \beta_1 + \beta_2\,\mathbf{X}_2 + \beta_3\,\mathbf{X}_3 + \boldsymbol{\varepsilon}$

• Results:
- Unbiased: Given that $\beta_3 = 0$ ⇒ $E[b_2|X] = \beta_2$
- Efficiency:

$$Var[b_2|X] = \frac{\sigma^2}{\sum(X_{2i} - \bar{X}_2)^2} \times \frac{1}{1 - r^2_{X_2,X_3}} > \frac{\sigma^2}{\sum(X_{2i} - \bar{X}_2)^2}$$

where r_{X_2,X_3} is the correlation coefficient between X_2 and X_3.

Note: These are the results in general. Note that if X_2 and X_3 are uncorrelated, there will be no loss of efficiency after all.

Testing Model Specification: Nested Models

In both previous cases, we have two nested models, one is the restricted version of the other. For example, in the case of omitted variables:

(U) $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta}_1 + \mathbf{Z}\,\boldsymbol{\beta}_2 + \varepsilon$ –the "long regression,"
(R) $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta}_1 + \varepsilon$ –the "short regression."

To test H_0 (No omitted variables): $\boldsymbol{\beta}_2 = 0$, we can use the F-test:

$$F = \frac{(RSS_R - RSS_U)/J}{RSS_U/(T-k)} \sim F_{J,T-K}.$$

Example: In the previous Lecture, we performed this F-test to test if in the 3-factor FF model for IBM returns, SMB and HML were significant, which they were. That is, we showed that the usual CAPM formulation for IBM returns had omitted variables: SMB and HML.

Testing Model Specification with an LM Test

Note that the F-test requires two estimations: the Unrestricted model and the Restricted model.

There is another test of H_0: $\boldsymbol{\beta}_2 = 0$, that only uses the restricted model as the basis for testing: The Lagrange Multiplier (LM) test, which we introduced in Lecture 5.

In this lecture, we present the simpler formulation of the LM test, which is based on the residuals of the restricted model.

Simple intuition. Everything that is omitted from (& belongs to!) a model is in the residuals ($\mathbf{e}_R$). The LM test is based on $\mathbf{e}_R$: We check if the omitted variables, **Z**, show up as drivers of e_R. We use a simple regression of e_R against **Z** to check for the misspecification.

• LM test steps:
(1) Run restricted model ($\mathbf{y} = \mathbf{X}\,\beta_1 + \varepsilon$). Get restricted residuals, $\mathbf{e}_R$.

(2) (Auxiliary Regression). Run the regression of e_R on all the *m* omitted *m* variables, **Z**, and the *k* included variables, **X**. In our case:

$$e_{R,i} = \alpha_0 + \alpha_1\, x_{i,1} + ...+ \alpha_k\, x_{i,k} + \gamma_1\, z_{i,1} + ... + \gamma_m\, z_{i,m} + v_i$$

⇒ Keep the R^2 from this regression, R^2_{eR}.
(3) Compute LM-statistic:

$$\text{LM} = T * R^2_{eR} \xrightarrow{d} \chi^2_m.$$

Technical Note: We include the original variables in (2), **X**, in the auxiliary regression to get the convenient form for the LM-test, as shown by Engle (1982).

The LM Test is very general. It can be used in many settings, for example, to test for nonlinearities, interactions among variables, autocorrelation or heteroscedasticity (discussed later).

Asymptotically speaking, the LM Test, the LR Test and the Wald Test are equivalent –i.e, they have the same limiting distribution, χ^2_m. In small *T,* they can have different conclusions. In general, however, we find: W > LR > LM. That is, the LM test is more conservative (cannot reject more often) and the Wald test is more aggressive.

Example: We use an LM test to check if the standard CAPM for IBM returns omits SMB and HML.

```
fit_r <- lm (ibm_x ~ Mkt_RF)
resid_r <- fit_r$residuals                # get residuals from R model
fit_lm <- lm (resid_r ~ Mkt_RF + SMB + HML)     # auxiliary regression
> summary(fit_lm)

Coefficients:
              Estimate      Std. Error t value Pr(>|t|)
(Intercept)   0.0007021     0.0024875   0.282   0.7779
Mkt_RF        0.0125253     0.0567221   0.221   0.8253
SMB          -0.2124596     0.0841119  -2.526   0.0118 *
HML          -0.1715002     0.0846817  -2.025   0.0433 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.05848 on 565 degrees of freedom
Multiple R-squared:  0.01649,   Adjusted R-squared:  0.01127
```

```
F-statistic: 3.158 on 3 and 565 DF,  p-value: 0.02438

R2_r <- summary(fit_lm)$r.squared   # extracting  R2 from fit_lm
> R2_r
[1] 0.01649104

LM_test <- R2_r * T
> LM_test
[1] 9.383402    ⇒ LM_test > qchisq (.95,df=2) ⇒ Reject H0.

qchisq(.95, df = 2)              # chi-squared (df=2) value at 5% level
p_val <- 1 - pchisq(LM_test, df = 2)             # p-value of LM_test
> p_val
[1] 0.009171071                  ⇒ p-value is small ⇒ Reject H0.      ¶
```

Note: In Lecture 5 we performed the same test with the Wald test (using the F distribution), the p-value was **0.0091175**. (This almost exact coincidence is not always the case.)

Functional Form: Linearity in Parameters

Linear in variables and parameters:

$$\boldsymbol{y} = \beta_1 + \beta_2\boldsymbol{X}_2 + \beta_3\boldsymbol{X}_3 + \beta_4 X_4 + \boldsymbol{\varepsilon}.$$

So far, this is the linear model we have used. OLS estimates all parameters: β_1, β_2, β_3, & β_4.

Non-linear in variables, but linear in parameters –i.e., *intrinsic linear*:

$$\boldsymbol{y} = \beta_1 + \beta_2\boldsymbol{X}_2^2 + \beta_3\sqrt{\boldsymbol{X}_3} + \beta_4 \log \boldsymbol{X}_4 + \boldsymbol{\varepsilon}$$

Define: $\boldsymbol{Z}_2 = \boldsymbol{X}_2^2$, $\quad \boldsymbol{Z}_3 = \sqrt{\boldsymbol{X}_3}$, $\quad$ & $\quad \boldsymbol{Z}_4 = \log \boldsymbol{X}_4$

Then, the non-linear model becomes a linear model:

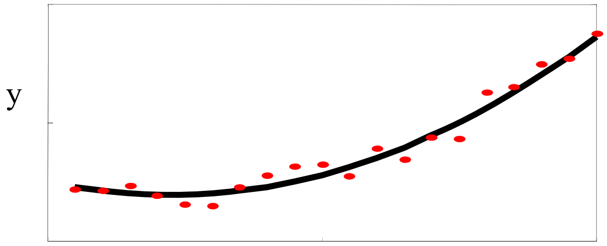
$$\boldsymbol{y} = \beta_1 + \beta_2\boldsymbol{Z}_2 + \beta_3\boldsymbol{Z}_3 + \beta_4 Z_4 + \boldsymbol{\varepsilon}$$

Again, OLS can be used to estimate all β_1, β_2, β_3, & β_4.

Suppose we have:

$$\boldsymbol{y} = \beta_1 + \beta_2\boldsymbol{X}_2 + \beta_3\boldsymbol{X}_2^2 + \boldsymbol{\varepsilon}$$

The model allows for a quadratic relation between **y** and **X₂**:



$\mathbf{X}_2$

Let $\boldsymbol{X}_3 = \boldsymbol{X}_2^2$, then, the model is intrinsic linear:

$$\boldsymbol{y} = \beta_1 + \beta_2 \boldsymbol{X}_2 + \beta_3 \boldsymbol{X}_3 + \varepsilon$$

Example: We want to test if a measure of market risk $(\text{Mkt}_{\text{Ret}} - r_f)^2$ is significant in the 3 FF factors (SMB, HML) for IBM returns. The model is non-linear in $(\text{Mkt}_{\text{Ret}} - r_f)$, but still intrinsic linear:

$$\text{IBM}_{\text{Ret}} - r_f = \beta_0 + \beta_1 (\text{Mkt}_{\text{Ret}} - r_f) + \beta_2 \,\text{SMB} + \beta_3 \,\text{HML} + \beta_4 (\text{Mkt}_{\text{Ret}} - r_f)^2 + \boldsymbol{\varepsilon}$$

We can do OLS, by redefining the variables: Let $\boldsymbol{X}_1 = (\text{Mkt}_{\text{Ret}} - r_f)$; $\boldsymbol{X}_2 = \text{SMB}$; $X_3 = \text{HML}$; $X_4 = X_1^2$. Then,

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{X}_1 + \beta_2 \boldsymbol{X}_2 + \beta_3 X_3 + \beta_4 \boldsymbol{X}_1^2 + \boldsymbol{\varepsilon}$$

```
Mkt_RF2 <- Mkt_RF^2
fit_capm_2 <- lm (ibm_x ~ Mkt_RF + SMB + HML + Mkt_RF2)
summary(fit_capm_2)

Coefficients:
            Estimate   Std. Error  t value Pr(>|t|)
(Intercept) -0.004765  0.002854    -1.670  0.0955 .
Mkt_RF      0.906527   0.057281    15.826 <2e-16 ***
SMB         -0.215128  0.084965    -2.532  0.0116 *
HML         -0.173160  0.085054    -2.036  0.0422 *
Mkt_RF2     -0.143191  0.617314    -0.232  0.8167          ⇒ Not significant!
```

- Now, we also check with an LM test if all variables squares ($(\text{Mkt}_{\text{Ret}} - r_f)^2$, SMB^2, and HML^2) are omitted from the 3-factor FF model for IBM returns.

```
Mkt_RF2 <- Mkt_RF^2
SMB2 <- SMB^2
HML2 <- HML^2
fit_r <- lm (ibm_x ~ Mkt_RF + SMB + HML)
resid_r <- fit_r$residuals
fit_lm <- lm (resid_r ~ Mkt_RF + SMB + HML + Mkt_RF2 + SMB2 + HML2)
R2_r <- summary(fit_lm)$r.squared
LM_test <- R2_r * T
> LM_test
[1] 2.453822
p_val <- 1 - pchisq(LM_test, df = 3)  # p-value of LM_test
> p_val
[1] 0.4836944           ⇒ p-value is higher than standard levels ⇒ Cannot Reject H0. ¶
```

- Nonlinear in parameters:

$$\boldsymbol{y} = \beta_1 + \beta_2 \boldsymbol{X}_2 + \beta_3 X_3 + \beta_2 \beta_3 \boldsymbol{X}_4 + \boldsymbol{\varepsilon}$$

This model is nonlinear in parameters since the coefficient of X_4 is the product of the coefficients of X_2 and X_3. OLS cannot be used to estimate all parameters.

- Some nonlinearities in parameters can be linearized by appropriate transformations, but not this one. This is not an intrinsic linear model. Different estimation techniques should be used in these cases.

Intrinsic linear models can be estimated using OLS. Sometimes, transformations are needed. Suppose we start with a power function:

$$\boldsymbol{y} = \beta_1 \boldsymbol{X}^{\beta_2} \boldsymbol{\varepsilon}$$

- The errors enter in multiplicative form. Then, using logs:

$$\log \boldsymbol{y} = \log \beta_1 \, \boldsymbol{X}^{\beta_2} \boldsymbol{\varepsilon} = \log \beta_1 + \beta_2 \log \boldsymbol{X} + \log \boldsymbol{\varepsilon},$$

or

$$\boldsymbol{y}' = \beta_1' + \beta_2 \boldsymbol{X}' + \boldsymbol{\varepsilon}',$$

where $\boldsymbol{y}' = \log \boldsymbol{y}\,, \boldsymbol{X}' = \log \boldsymbol{X}, \beta_1' = \log \beta_1\,, \boldsymbol{\varepsilon}' = \log \boldsymbol{\varepsilon}$

Now, we have an intrinsic linear model: OLS can be used to estimate all the parameters.

Similar intrinsic model can be obtained if $\boldsymbol{y} = \boldsymbol{e}^{\beta_1 + \beta_2 \boldsymbol{X} + \boldsymbol{\varepsilon}}$

Note: Recall that we can only use logs when $\boldsymbol{y}$ has positive values. In general, we use logs when we believe the independent variable has an exponential or power formulation, typical behavior for nominal variables, like sales, revenue or prices.

- Not all models are intrinsic linear. For example:

$$\boldsymbol{y} = \beta_1 \boldsymbol{X}^{\beta_2} + \boldsymbol{\varepsilon}$$
$$\log \boldsymbol{y} = \log(\beta_1 \boldsymbol{X}^{\beta_2} + \boldsymbol{\varepsilon})$$

We cannot linearize the model by taking logarithms. There is no way of simplifying $\log(\beta_1 X^{\beta_2} + \varepsilon)$.

We will have to use some nonlinear estimation technique for these situations. (ML can estimate this model.)

Functional Form: Linear vs Log specifications

Two popular models, especially in Corporate Finance: linear or log?

Model 1 - Linear model: $\boldsymbol{y} = \beta_1 + \beta_2 \boldsymbol{X} + \varepsilon$

Model 2 - (Semi-) Log model: $\log \boldsymbol{y} = \beta_1 + \beta_2 \boldsymbol{X} + \varepsilon$

Box–Cox transformation: $\frac{Y^{\lambda}-1}{\lambda} = \beta_1 + \beta_2 \boldsymbol{X} + \varepsilon$

$\frac{Y^{\lambda}-1}{\lambda} = Y - 1$ when $\lambda = 1$

$\frac{Y^{\lambda}-1}{\lambda} = \log(Y)$ when $\lambda \to 0$

Putting $\lambda = 0$ gives the (semi–)log model (think about the limit of λ tends to zero.). The Box-Cox transformation is flexible. We can estimate λ to test if λ is equal to 0 or 1. It is possible that it is neither!

Functional Form: Ramsey's RESET Test

To test the specification of the functional form, Ramsey designed a simple test. We start with the fitted values from our (**A1**) model:

$$\hat{\mathbf{y}} = \mathbf{Xb}.$$

Then, we add $\hat{\mathbf{y}}^2$ to the regression specification:

$$\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \hat{\mathbf{y}}^2\,\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

If $\hat{\mathbf{y}}^2$ is added to the regression specification, it should pick up quadratic and interactive nonlinearity, if present, without necessarily being highly correlated with any of the $\boldsymbol{X}$ variables.

We test H_0 (linear functional form): $\boldsymbol{\gamma} = 0$

H_1 (non linear functional form): $\boldsymbol{\gamma} \neq 0$

$\Rightarrow$ *t-test* on the OLS estimator of $\boldsymbol{\gamma}$.

If the *t-statistic* for $\hat{\mathbf{y}}^2$ is significant $\Rightarrow$ evidence of nonlinearity.
The RESET test is intended to detect nonlinearity, but not be specific about the most appropriate nonlinear model (no specific functional form is specified in H_1).

Example: We want to test the functional form of the 3 FF Factor Model for IBM returns, using monthly data 1973-2020.

```
fit <- lm(ibm_x ~ Mkt_RF + SMB + HML)
y_hat <- fitted(fit)
y_hat2 <- y_hat^2
fit_ramsey <- lm(ibm_x ~ Mkt_RF + SMB + HML + y_hat2)
> summary(fit_ramsey)
```

```
Coefficients:
               Estimate     Std. Error    t value      Pr(>|t|)
(Intercept)   -0.004547     0.002871      -1.584  0.1137
Mkt_RF         0.903783     0.058003      15.582  <2e-16 ***
SMB           -0.217268     0.085128      -2.552  0.0110 *
HML           -0.173276     0.084875      -2.042  0.0417 *
y_hat2        -0.289197     0.763526      -0.379  0.7050        ⇒ Not significant!
```

There is package in R, *lmtest,* that performs this test, *resettest,* (and many others, used in this class, encompassing, jtest, waldtest, etc). You need to install it first: install.packages("lmtest"), then call the library(lmtest).

Note: The test reported is an *F*-test ~ $F_{1,T-k}$, which is equal to $(t_{T-k})^2$. The *p-values* should be the same.

```
>library(lmtest)
> resettest(y ~ Mkt_RF + SMB + HML, power=2, type="fitted")
        RESET test
data:  y ~ Mkt_RF + SMB + HML
RESET = 0.14346, df1 = 1, df2 = 564, p-value = 0.705
```

⇒ cannot reject H_0. Check: $(-0.379)^2 = 0.1434$. ¶

Qualitative Variables and Functional Form

Suppose that you want to model CEO compensation. You have data on annual total CEO compensation (*Comp*), annual returns, annual sales, CEO's age, CEO's previous experience, and the CEO's last degree (education). We have qualitative data.

We can run individual regressions for each last degree –i.e., BA/BS; MS/MA/MBA; Doctoral-, but we will have three small samples:

Undergrad degree $\qquad Comp_i = \beta_{0\text{-u}} + \boldsymbol{\beta}_{\mathbf{1\text{-u}}}'\mathbf{z}_\mathbf{i} + \varepsilon_{u,i}$

Masters degree $\qquad Comp_i = \beta_{0\text{-m}} + \boldsymbol{\beta}_{\mathbf{1\text{-m}}}'\mathbf{z}_\mathbf{i} + \varepsilon_{m,i}$

Doctoral degree $\qquad Comp_i = \beta_{0\text{-d}} + \boldsymbol{\beta}_{\mathbf{1\text{-d}}}'\mathbf{z}_\mathbf{i} + \varepsilon_{d,i}$

where the $\mathbf{z}_\mathbf{i}$ is a vector of the CEO *i's* age and previous experience and his/her firm's *annual* returns and annual sales.

Alternatively, we can combine the regressions in one. We can use a variable (a *dummy or indicator variable*) that points whether an observation belongs to a category or class or not. For example:

$D_{C,i} = 1 \qquad$ if observation *i* belongs to category C (say, male.)
$\qquad = 0 \qquad$ otherwise.

Simple process: First, we define dummy/indicator variables for Masters & doctoral degrees:

$D_m = 1 \qquad$ if at least Masters degree
$\qquad = 0 \qquad$ otherwise.
$D_d = 1 \qquad$ if doctoral degree
$\qquad = 0 \qquad$ otherwise.

Then, we introduce the dummy/indicator variables in the model:

$$Comp_i = \beta_0 + \boldsymbol{\beta}_\mathbf{1}'\mathbf{z}_i + \beta_2 D_{m,i} + \beta_3 D_{d,i} + \boldsymbol{\gamma}_\mathbf{1}'\mathbf{z}_i\, D_{m,i} + \boldsymbol{\gamma}_\mathbf{2}'\mathbf{z}_i\, D_{d,i} + \varepsilon_i$$

This model uses all the sample to estimate the parameters. It is flexible:

- Model for undergrads only ($D_{m,i} = 0$ & $D_{d,i} = 0$):

$$Comp_i = \beta_0 + \boldsymbol{\beta}_\mathbf{1}'\mathbf{z}_i + \varepsilon_i$$

- Model for Masters degree only ($D_{m,i} = 1$ & $D_{d,i} = 0$):

$$Comp_i = (\beta_0 + \beta_2) + (\boldsymbol{\beta}_\mathbf{1} + \boldsymbol{\gamma}_\mathbf{1})'\mathbf{z}_i + \varepsilon_i$$

- Model for Doctoral degree only ($D_{m,i} = 1$ & $D_{d,i} = 2$):

$$Comp_i = (\beta_0 + \beta_2 + \beta_3) + (\boldsymbol{\beta}_\mathbf{1} + \boldsymbol{\gamma}_\mathbf{1} + \boldsymbol{\gamma}_\mathbf{2})'\mathbf{z}_i + \varepsilon_i$$

The parameters for the different categories are:

- Constant:

Constant for undergrad degree: β_0

Constant for Masters degree: $\beta_0 + \beta_2$

Constant for Doctoral degree: $\beta_0 + \beta_2 + \beta_3$

- Slopes:

Slopes for Masters degree: $\boldsymbol{\beta}_1 + \boldsymbol{\gamma}_1$

Slopes for Doctoral degree: $\boldsymbol{\beta}_1 + \boldsymbol{\gamma}_1 + \boldsymbol{\gamma}_2$

We can test the effect of education on CEO compensation:

(1) H_0: No effect of grad degree: $\beta_3 = \beta_2 = 0$ & $\boldsymbol{\gamma}_1 = \boldsymbol{\gamma}_2 = \mathbf{0}$ ⇒ *F-test.*

(2) H_0: No effect of Masters degree on constant: $\beta_2 = 0$ ⇒ *t-test.*

(3) H_0: No effect of doctoral degree: $\beta_3 = 0$ & $\boldsymbol{\gamma}_2 = \mathbf{0}$ ⇒ *F-test.*

(4) H_0: No effect of Dr degree on marginal effect: $\boldsymbol{\gamma}_2 = \mathbf{0}$ ⇒ *F-test.*

• We may have more than one qualitative category (last degree above) in our data that we may want to introduce in our model.

Example: Suppose we also have data for CEO graduate school. Now, we can create another qualitative category, "quality of school", defined as Top 20 school, to test if a Top 20 school provides "more value." To do this, we use D_{T20} to define if any schooling is in the Top 20.

$D_{T20} = 1$ if school is a Top 20 school
$\quad\;\; = 0$ otherwise.

The model becomes:

$$Comp_i = \beta_0 + \boldsymbol{\beta}_1'\mathbf{z}_i + \beta_2 D_{m,i} + \beta_3 D_{d,i} + \beta_4 D_{T20,i} + \boldsymbol{\gamma}_1'\mathbf{z}_i\, D_{m,i} + \boldsymbol{\gamma}_2'\mathbf{z}_i\, D_{d,i} + \boldsymbol{\gamma}_3'\mathbf{z}_i\, D_{T20,i} + \varepsilon_i$$

In this setting, we can test the effect of a Top20 education on CEO compensation:

(1) H_0: No effect of Top20 degree: $\beta_4 = 0$ and $\boldsymbol{\gamma}_3 = \mathbf{0}$ ⇒ *F-test.* ¶

• The omitted category is the reference or control category.

- In our first example, with only educational degrees, the reference category is undergraduate degree. - In the second example, with educational degrees and quality of school (Top20 dummy), the reference category is undergraduate degree with no Top 20 education.

• *Dummy trap*. If there is a constant, the numbers of dummy variables per qualitative variable should be equal to the number of categories minus 1. If you put the number of dummies variables equals the number of categories, you will create perfect multicollinearity.

Dummy Variables as Seasonal Factors

A popular use of dummy variables is in estimating seasonal effects. We may be interested in studying the January effect in stock returns or if the returns of oil companies (say, Exxon or BP) are affected by the seasons, since in the winter people drive less and in the summer more.

In this case, we define dummy/indicator variables for Summer, Fall and Winter (the base case is, thus, Spring):

$D_{Sum,i} = 1$ if observation i occurs in Summer
$= 0$ otherwise.
$D_{Fall,i} = 1$ if observation i occurs in Fall
$= 0$ otherwise.
$D_{Win,i} = 1$ if observation i occurs in Winter
$= 0$ otherwise.

Then, letting **Z** be the three FF factors, we have:

$$XOM_i = \beta_0 + \boldsymbol{\beta}_1'\mathbf{z}_i + \beta_2 D_{Sum,i} + \beta_3 D_{Fall,i} + \beta_4 D_{Win,i} + \varepsilon_i$$

Example: In the context of the 3-factor FF model, we test if Exxon (XOM) is affected by seasonal (quarters) factors:

$$XOM_i = \beta_0 + \boldsymbol{\beta}_1'\mathbf{z}_i + \beta_2 D_{Sum,i} + \beta_3 D_{Fall,i} + \beta_4 D_{Win,i} + \varepsilon_i$$

```
x_xom <- SFX_da$XOM                                    # Extract XOM prices
T <- length(x_xom)
lr_xom <- log(x_xom[-1]/x_xom[-T])
xom_x <- lr_xom – RF

T <- length(xom_x)
Summ <- rep(c(0,0,0,0,0,0,1,1,1,0,0,0), round(T/12)+1)      # Create Summer dummy
Fall <- rep(c(0,0,0,0,0,0,0,0,0,1,1,1), round(T/12)+1)           # Create Fall dummy
Wint <- rep(c(1,1,1,0,0,0,0,0,0,0,0,0), round(T/12)+1)      # Create Winter dummy
T1 <- T+1
Fall_1 <- Fall[2:T1]                                   # Adjusting sample (starts in Feb)
Wint_1 <- Wint[2:T1]
Summ_1 <- Summ[2:T1]
fit_xom_s <- lm(xom_x ~ Mkt_RF + SMB + HML + Fall_1 + Wint_1 + Summ_1)
summary(fit_xom_s)
> summary(fit_xom_s)

Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept) 0.002445   0.003485  0.702   0.4832      ⇒ constant for reference category
(Spring)≈0.
Mkt_RF     0.761816   0.040602  18.763  < 2e-16 ***
SMB        -0.261925  0.060575  -4.324  1.81e-05 ***
HML        0.370623   0.060049  6.172   1.29e-09 ***
Fall_1     -0.006609  0.004947  -1.336   0.1822
Wint_1     -0.011283  0.004928  -2.290   0.0224 *   ⇒ significant. Reject H0: No Winter effect.
Summ_1     -0.007100  0.004944  -1.436   0.1515
```

<u>Interpretation</u>: In the Winter quarter, Exxon excess returns decrease, relative to the Spring, by 1.13%. But since Spring's (& Fall's & Winter's) effect is non-significant, the decrease is in absolute terms.

- We can test if all quarters jointly matter. That is, H_0: $\beta_2 = \beta_3 = \beta_4 = 0$.

We do an F-test:

```
fit_u <- lm (xom_x ~ Mkt_RF + SMB + HML + Fall_1 + Wint_1 + Summ_1)
fit_r <- lm (xom_x ~ Mkt_RF + SMB + HML)
resid_u <- fit_u$residuals
RSS_u <- sum((resid_u)^2)
resid_r <- fit_r$residuals
RSS_r <- sum((resid_r)^2)
f_test <- ((RSS_r - RSS_u)/2)/(RSS_u/(T-4))
> f_test
[1] 2.706574
>
p_val <- 1 - pf(f_test,df1=3, df2=T-3)          # p-value of F-test
>  p_val
[1] 0.05504357       ⇒ p-value is "marginal." Cannot reject H0: No significant joint seasonal
effect.
```

• Suppose we are also interested in checking if the slopes –i.e., the marginal effects- are affected by the Winter quarter. Then, we fit:

$$XOM_i = \beta_0 + \boldsymbol{\beta_1}'\mathbf{z}_i + \beta_2 D_{Sum,i} + \beta_3 D_{Fall,i} + \beta_4 D_{Win,i} + \boldsymbol{\gamma_1}'\mathbf{z}_i D_{Win,i} + \varepsilon_i$$

```
Mkt_W <- Mkt_RF*Wint_1
SMB_W <- SMB*Wint_1
HML_W <- HML*Wint_1
fit_xom_s2 <- lm(xom_x ~ Mkt_RF + SMB + HML + Mkt_W + SMB_W + HML_W + Fall_1
+ Wint_1 + Summ_1)
summary(fit_xom_s2)
> summary(fit_xom_s2)

Coefficients:
              Estimate   Std. Error t value Pr(>|t|)
(Intercept)  0.003127  0.003478  0.899   0.368962
Mkt_RF       0.695762  0.048202  14.434 < 2e-16 ***
SMB         -0.291199  0.075197  -3.872  0.000120 ***
HML          0.270262  0.077416  3.491   0.000519 ***
Mkt_W                  0.208912  0.091972    2.271   0.023497 *          ⇒ significant effect
on Mkt's slope
SMB_W                  0.064753  0.126138   0.513    0.607911
HML_W        0.198753  0.124261   1.599  0.110278
Fall_1      -0.006795  0.004934  -1.377  0.169038
Wint_1      -0.013747  0.005000  -2.750  0.006159 **             ⇒ significant effect on
constant.
Summ_1      -0.007492  0.004928  -1.520  0.129012
```

Interpretation: The only factor interacting significantly with Winter is the Market factor. Then, we have two significantly different slopes:

- In the Winter, the Market slope is: **0.695762** + **0.208912** = **0.903674**

- In all other quarters, the Market is: **0.695762**

It looks like in the Winter, XOM behaves closer to the Market, while in all other quarters, it is significantly less risky than the market.

- Now, a joint interacting Winter effect is not significant (but, significant at the 10% level):

```
> f_test
[1] 6.505231
p_val <- 1 - pf(f_test, df`1= 3, df2=T-7)                       # p-value of F-test
>  p_val
[1] 0.0007923967
```

⇒ *p-value* < .05, then, we reject H_0 (joint Winter interactive effect): $\gamma_1 = 0$. ¶

Dummy Variables: Is There a January Effect?

Example: We want to test the January effect on IBM stock returns, where because of tax reasons/window dressing, stocks go down in December and recover in January. The test can be done by adding a dummy variable to the 3-factor FF model:

$$D_{J,i} = 1 \qquad \text{if observation } i \text{ occurs in January}$$
$$= 0 \qquad \text{otherwise.}$$

Then, we estimate the expanded model:

$$(IBM_{Ret} - r_f)_i = \beta_0 + \beta_1\,(Mkt_{Ret} - r_f)_i + \beta_2\,SMB_i + \beta_3\,HML_i + \beta_4\,D_{J,i} + \varepsilon_i$$

We test H_0(No January effect): $\beta_4 = 0$ ⇒ *t-test*.

Alternatively, we can estimate do an LM test on the residuals of the 3-factor FF model and check if D_J is significant.

```
T <- length(ibm_x)
Jan <- rep(c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (round(T)/12+1))       # Create January dummy
T2 <- T+1
Jan_1 <- Jan[2:T2]          # Adjust sample
fit_r <- lm (ibm_x ~ Mkt_RF + SMB +  HML)                    # Restricted Regression
resid_r <- fit_r$residuals  # Keep residuals (eR)
fit_Jan <- lm (resid_r ~ Mkt_RF + SMB + HML + Jan_1)   # Auxiliary Regression
> summary(fit_Jan)

Coefficients:
             Estimate   Std. Error  t value Pr(>|t|)
(Intercept) -0.002111  0.002561   -0.824  0.41027
Mkt_RF       -0.005198
             0.056405   -0.092  0.92661
SMB          -0.026306  0.084063   -0.313  0.75445
HML          -0.014914  0.083606   -0.178  0.85848
Jan_1        0.026966   0.008906   3.028  0.00258 **
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.058 on 565 degrees of freedom
Multiple R-squared:  0.01597,   Adjusted R-squared:  0.009
F-statistic: 2.292 on 4 and 565 DF,  p-value: 0.05841

R2_r  <-  summary(fit_Jan)$r.squared                  # Keep R^2 from Auxiliary Regression
> R2_r
[1] 0.01596528
LM_test <- R2_r * T
> LM_test
[1] 9.084247
p_val <- 1 - pchisq(LM_test, df = 1)                   # p-value of LM_test
> p_val
[1] 0.002578207   ⇒ p-value is small ⇒ Reject H0.
```

Given this result, we modify the 3-factor FF and add the January Dummy to the FF model:

```
fit_ibm_new <- lm (ibm_x ~ Mkt_RF + SMB + HML + Jan_1)
> summary(fit_ibm_new)

Coefficients:
             Estimate   Std. Error  t value Pr(>|t|)
(Intercept) -0.007302  0.002561   -2.851  0.00452 **
Mkt_RF       0.905182  0.056405   16.048  < 2e-16 ***
SMB         -0.247691  0.084063   -2.946  0.00335 **
HML         -0.154093  0.083606   -1.843  0.06584 .
Jan_1        0.026966  0.008906    3.028  0.00258 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.058 on 565 degrees of freedom
Multiple R-squared:  0.3499,    Adjusted R-squared:  0.3453
F-statistic: 76.01 on 4 and 565 DF,  p-value: < 2.2e-16
```

Interpretation: We have two constants (excess return, Jensen’s alpha):
Feb - Dec: **-0.7302%** (significant).
January: **-0.7302%** + **2.6966%** = **1.9664%** (significant).

When the January dummy was not in the model, we had: **-0.005191**, which is close to an average of the constants (= **-0.007302** *11 + **0.019664**)/12 = -0.00505).

Interpretation: During January IBM has an additional **2.6966%** excess returns. This is a big number. Today, the evidence for the January effect is much weaker than in this case. ¶

Note: In the FF model we expect the constant to be very small (≈0). In this case, it is not zero. Maybe we have a misspecified (**A1**).

Dummy Variable for One Observation

We can use a dummy variable to isolate a single observation.

$$D_J \quad = 1 \quad \text{for observation } j.$$
$$\quad\;\; = 0 \quad \text{otherwise.}$$

Define **d** to be the dummy variable in question.

$\mathbf{Z}$ = all other regressors. $\mathbf{X} = [\mathbf{Z}, \mathbf{D}_J]$

Multiple regression of **y** on **X**. We know that

$\mathbf{X}'\mathbf{e} = \mathbf{0}$ where **e** = the column vector of residuals.

$\Rightarrow \mathbf{D}_J'\mathbf{e} = 0 \quad \Rightarrow e_j = 0$ (perfect fit for observation j).

This approach can be used to deal with (eliminate) *outliers*.

Example: In Dec 1992, IBM reported record losses and gave a very bleak picture of its future. The stock tumbled -30.64% that month. We check the effect of that extreme observation, a potential outlier, on the 3-factor FF model + January dummy:

```
dec_1992 <- rep(0,T)                                    # Define Dec 1992 dummy
dec_1992[239] <- 1                                      # Define Dec 1992 dummy (=1 if Dec 1992)
fit_d92 <- lm (ibm_x ~ Mkt_RF + SMB + HML + Jan_1 + dec_1992)
> summary(fit_d92)

Coefficients:
             Estimate  Std. Error  t value Pr(>|t|)
(Intercept)  -0.0067720.002502  -2.707  0.00699 **
Mkt_RF       0.908775 0.055054  16.507  < 2e-16 ***
SMB          -0.2392130.082059  -2.915  0.00370 **
HML          -0.1386290.081647  -1.698  0.09008 .
Jan_1        0.026163 0.008694  3.009  0.00273 **
dec_1992     -0.3062020.056710  -5.399 9.86e-08 ***        (same value of observation)
```

Note: Potential "Outlier" has no major effect on coefficients. ¶

Chow Test: Testing the effect of Categories on a Model

It is common to have a qualitative variable with two categories, say education (MS/MBA or not). Before modelling the data, we can check if only one regression ("pooling") model applies to both categories.

We use the Chow Test (an F-test) –Chow (1960, *Econometrica*).

Steps:

(1) Run OLS with all the data, with no distinction between schools (Pooled regression or Restricted regression). Keep RSS_R.

(2) Run two separate OLS, one for each school (Unrestricted regression). Keep RSS_1 and RSS_2 $\Rightarrow$ $RSS_U = RSS_1 + RSS_2$.
(Alternative, we can run just one regression with the dummy variable).

(3) Run a standard F-test (testing Restricted vs. Unrestricted models):

$$F = \frac{(RSS_R - RSS_U)/(k_U - k_R)}{(RSS_U)/(T - k_U)} = \frac{(RSS_R - [RSS_1 + RSS_2])/k}{(RSS_1 + RSS_2)/(T - 2k)}$$

Example: Who visits doctors more: Men or Women?
Data: German Health Care Usage Data, with 7,293 Individuals.
Time Periods: Varying Number.
Variables in the file are:
Data downloaded from Journal of Applied Econometrics Archive. This is an unbalanced panel with 7,293 individuals. There are altogether **27,326** observations. The number of observations ranges from 1 to 7 per family. (Frequencies are: 1=1525, 2=2158, 3=825, 4=926, 5=1051, 6=1000, 7=987). The dependent variable of interest is
DOCVIS = number of visits to the doctor in the observation period
HHNINC = household nominal monthly net income in German marks / 10000.
(4 observations with income=0 were dropped)
HHKIDS = children under age 16 in the household = 1; otherwise = 0
EDUC = years of schooling
AGE = age in years
MARRIED= marital status (1 = if married)
WHITEC = 1 if has "white collar" job

- OLS Estimation for **Men** only. Keep RSS_M = **379.8470**

```
+----------------------------------------------------+
| Ordinary    least squares regression               |
| LHS=HHNINC   Mean                  =   .3590541    |
|              Standard deviation    =   .1735639    |
|              Number of observs.    =      14243    |
| Model size   Parameters            =          5    |
|              Degrees of freedom    =      14238    |
| Residuals    Sum of squares        =   379.8470    |
|              Standard error of e   =   .1633352    |
| Fit          R-squared             =   .1146423    |
|              Adjusted R-squared    =   .1143936    |
+----------------------------------------------------+
+--------+--------------+----------------+--------+--------+--+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+--+
|Constant|     .04169***        .00894        4.662    .0000
```

```
|AGE     |      .00086***        .00013         6.654   .0000
|EDUC    |      .02044***        .00058        35.528   .0000
|MARRIED |      .03825***        .00341        11.203   .0000
|WHITEC  |      .03969***        .00305        13.002   .0000
+--------+------------------------------------------------------+
```

- OLS Estimation for **Women** only. Keep RSS$_W$ = **363.8789**

```
+-----------------------------------------------------------+
| Ordinary    least squares regression                      |
| LHS=HHNINC   Mean                      =   .3444951       |
|              Standard deviation        =   .1801790       |
|              Number of observs.        =      13083       |
| Model size   Parameters                =          5       |
|              Degrees of freedom        =      13078       |
| Residuals    Sum of squares            =   363.8789       |
|              Standard error of e       =   .1668045       |
| Fit          R-squared                 =   .1432098       |
|              Adjusted R-squared        =   .1429477       |
+-----------------------------------------------------------+
+--------+--------------+----------------+--------+--------+--+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+--+
|Constant|      .01191           .01158         1.029   .3036
|AGE     |      .00026*          .00014         1.875   .0608
|EDUC    |      .01941***        .00072        26.803   .0000
|MARRIED |      .12081***        .00343        35.227   .0000
|WHITEC  |      .06445***        .00334        19.310   .0000
+--------+------------------------------------------------------+
```

- OLS Estimation for **ALL**. Keep$_{ALL}$ = **752.4767**

```
+-----------------------------------------------------------+
| Ordinary    least squares regression                      |
| LHS=HHNINC   Mean                      =   .3520836       |
|              Standard deviation        =   .1769083       |
|              Number of observs.        =      27326       |
| Model size   Parameters                =          5       |
|              Degrees of freedom        =      27321       |
| Residuals    Sum of squares            =   752.4767       | All
| Residuals    Sum of squares            =   379.8470       | Men
| Residuals    Sum of squares            =   363.8789       | Women
+-----------------------------------------------------------+
+--------+--------------+----------------+--------+--------+--+
|Variable| Coefficient  | Standard Error |b/St.Er.|P[|Z|>z]|
+--------+--------------+----------------+--------+--------+--+
|Constant|      .04186***        .00704         5.949   .0000
|AGE     |      .00030***     .919581D-04       3.209   .0013
```

```
|EDUC    |      .01967***         .00045        44.180    .0000
|MARRIED |      .07947***         .00239        33.192    .0000
|WHITEC  |      .04819***         .00225        21.465    .0000
+--------+------------------------------------------------------+
```

Chow Test = F = [(**752.4767** – (**379.847** + **363.8789**))/5] / [(**379.847** + **363.8789**)/(**27,326** – 10)]
= = **64.281**
F(5, 27311) = **2.214100** ⇒ reject H_0. ¶

Functional Form: Structural Change

Suppose there is an event that we think had a big effect on the behaviour of our model. Suppose the event occurred at time T_{SB}. We think that the before and after behaviour of the model is significantly different. For example, the parameters are different before and after T_{SB}. That is,

$$y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \beta_3 X_{3,i} + \varepsilon_i \qquad \text{for } i \leq T_{SB}$$
$$y_i = \gamma_0 + \gamma_1 X_{1,i} + \gamma_2 X_{2,i} + \gamma_3 X_{3,i} + \varepsilon_i \qquad \text{for } i > T_{SB}$$

The event caused structural change in the model.

What events may have this effect on a model? A financial crisis, a big recession, an oil shock, Covid-19, etc.

Under the H_0 (No *structural change*), we can pool the data into one model. That is, the parameters are the same under both regimes. We fit the same model for all *i*, for example:

$$y_i = \beta_0 + \boldsymbol{\beta_1}'\mathbf{x}_i + \varepsilon_i$$

If the Chow test rejects H_0, we need to reformulate the model. A typical reformulation includes a dummy variable ($D_{SB,i}$). For example:

$$y_i = \beta_0 + \boldsymbol{\beta_1}'\mathbf{x}_i + \beta_2 D_{SB,i} + \boldsymbol{\gamma_1}'\mathbf{x}_i D_{SB,i} + \varepsilon_i$$

where

$$D_{SB,i} = 1 \qquad \text{if observation } i \text{ occurred after } T_{SB}$$
$$= 0 \qquad \text{otherwise.}$$

.

Example: We are interested in the effect of the October 1973 oil shock in GDP growth rates. We can include a dummy variable in the model, say D_{73}:

$$D_{73,i} = 1 \qquad \text{if observation } i \text{ occurred after October 1973}$$
$$= 0 \qquad \text{otherwise.}$$
$$y_i = \beta_0 + \boldsymbol{\beta_1}'\mathbf{x}_i + \beta_2 D_{73,i} + \boldsymbol{\gamma_1}'\mathbf{x}_i D_{73,i} + \varepsilon_i$$

In the model, the oil shock affected the constant and the slopes.
- Constant:
  Before oil shock ($D_{73} = 0$): β_0
  After oil shock ($D_{73} = 1$): $\beta_0 + \beta_2$
- Slopes:
  Before oil shock ($D_{73} = 0$): $\boldsymbol{\beta_1}$
  After oil shock ($D_{73} = 1$): $\boldsymbol{\beta_1} + \boldsymbol{\gamma_1}$

We can estimate the above model and do an F-test to test if H_0 (No *structural change*): $\beta_2 = 0$ & $\gamma_1 = 0$.

Alternatively, we can also use a Chow test to test H_0 (No *structural change*), before fitting the model with the $D_{73,i}$ dummy variable.

• Steps for Chow (Structural Change) Test:
(1) Run OLS with all the data, with no distinction between regimes (Restricted or pooled model): Keep RSS_R.

(2) Run two separate OLS, one for each regime (Unrestricted model):
Before October 1973. Keep RSS_1.
After October 1973. Keep RSS_2. ⇒ $RSS_U = RSS_1 + RSS_2$.

(3) Run a standard F-test (testing Restricted vs. Unrestricted models):

$$F = \frac{(RSS_R - RSS_U)/(k_U - k_R)}{(RSS_U)/(T - k_U)} = \frac{(RSS_R - [RSS_1 + RSS_2])/k}{(RSS_1 + RSS_2)/(T - 2k)}$$

If the Chow test rejects H_0, we need to reformulate the model. A typical reformulation includes a dummy variable ($D_{73,i}$). ¶

Testing for structural change is the more popular use of the Chow test.

Chow tests have many interpretations: tests for structural breaks, pooling groups, parameter stability, predictive power, etc.

One important consideration: T may not be large enough. For example, we may think that Covid-19 had a structural effect on the behaviour of tech companies. We may not have enough data to run an F-test.
Example: 3 Factor Fama-French Model for IBM (continuation)
Q: Did the dot.com bubble (end of 2001) affect the structure of the FF Model? Sample: Jan 1973 – June 2020 (T = 569).
Pooled RSS = **1.9324**
Jan 1973 – Dec 2001 RSS = RSS_1 = **1.3307** (T = 342)
Jan 2002 – June 2020 RSS = RSS_2 = **0.5791** (T = 227)

	Constant	Mkt – rf	SMB	HML	RSS	T
1973-2020	-0.0051	0.9083	-0.2125	-0.1715	**1.9324**	569
1973-2001	-0.0038	0.8092	-0.2230	-0.1970	**1.3307**	342

2002 – 2020	-0.0073	1.0874	-0.1955	-0.3329	**0.5791**	227

$$F = \frac{[RSS_R - (RSS_1 + RSS_2)]/k}{(RSS_1 + RSS_2)/(T-k)} = \frac{[\mathbf{1.9324} - (\mathbf{1.3307} + 0\mathbf{.5792})]/4}{(\mathbf{1.3307} + \mathbf{.05791})/(569 - 2*4)} = \mathbf{1.6627}$$

⇒ Since $F_{4,565,.05}$ = **2.39**, we cannot reject H_0. ¶

Chow Test: Structural Change – Unknown Break

The previous example, computes the Chow test assuming that we know exactly when the break occurred –say, October 73 or Dec 2001.

That is, the results are *conditional* on the assumed breaking point.

In general, breaking points are unknown, we need to estimate them.

One quick approach is to do a rolling Chow test –that is we run the Chow test for all dates in the sample– and pick the date that maximizes the F-test.

This test was proposed by Quandt (1958):

$$QLR_T = \max_{\tau \varepsilon \{\tau_{\min}, \ldots, \tau_{\max}\}} F_T(\tau)$$

The max (supremum) is taken over all potential breaks in ($\tau_{\min}$, $\tau_{\max}$). For example, $\tau_{\min}$= T*.15; $\tau_{\max}$ = T*.85; that is we trim 30% of the observations (π_0 = 15% in each side) to run the test.

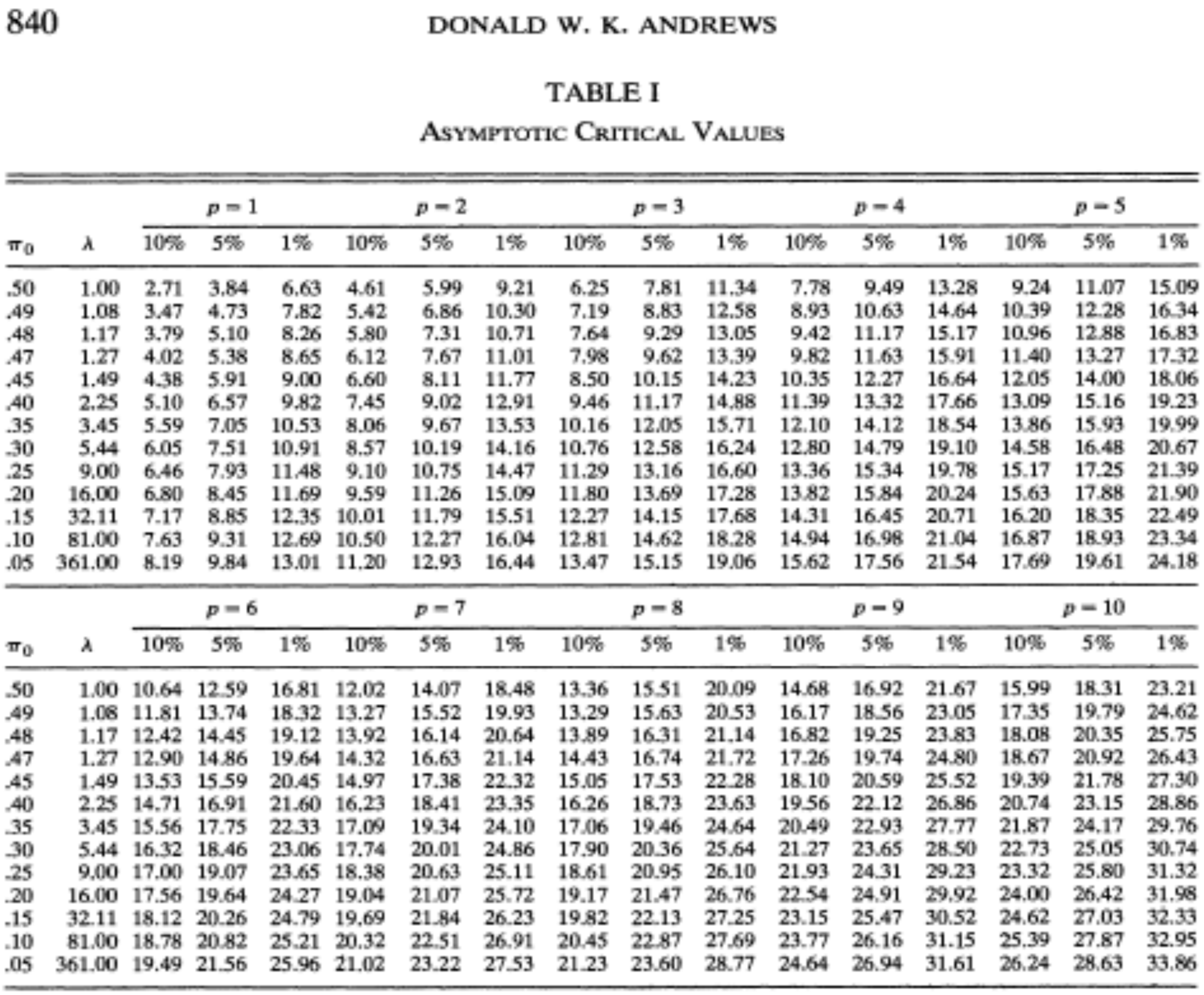
The problem with this approach is that the technical conditions under which the asymptotic distribution is derived are not met in this setting (the F-test are correlated, they are not independent).

Andrews (1993) showed that under appropriate conditions, the QLR statistic, also known as SupLR statistic, has a *non-standard limiting distribution* ("*non-standard"* = no existing table; needs a new one).

Andrews (1993) tabulated the non-standard distribution for different number of parameters in model (*k),* trimming values (π_0), & significance level (α). Andrews' table is in the next slide.

For example, for k=4, π_0= $\tau_{\min}/T$=(1-$\tau_{\max}/T$)=.15, & α=.05, the critical value is = **16.45**.

Critical values of the QLR test Distribution, taken from Andrews (1993). Note: p = # of parameters (*k),* π_0 = trimming value. (Ignore λ.)



840 DONALD W. K. ANDREWS

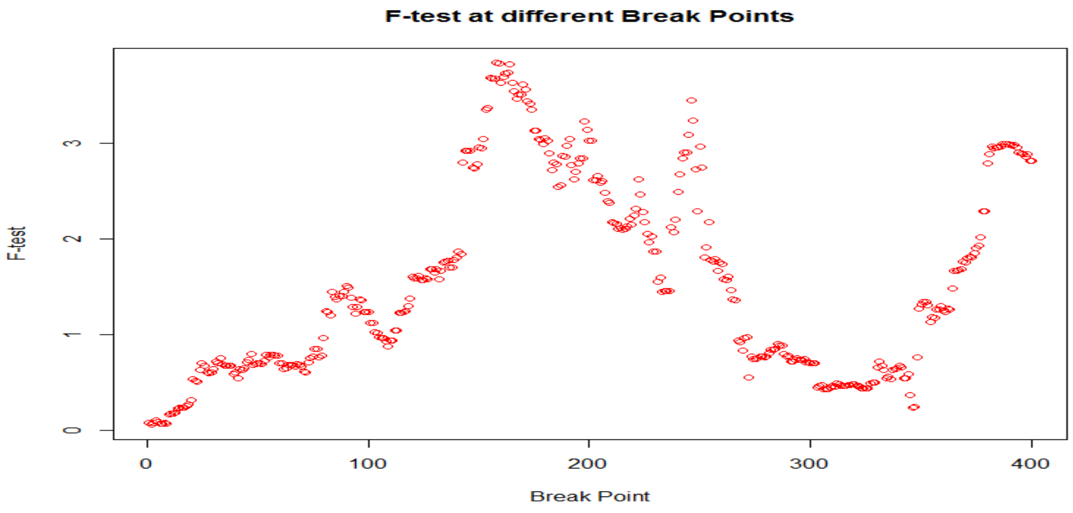
TABLE I

ASYMPTOTIC CRITICAL VALUES

		$p=1$			$p=2$			$p=3$			$p=4$			$p=5$		
π_0	λ	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%
.50	1.00	2.71	3.84	6.63	4.61	5.99	9.21	6.25	7.81	11.34	7.78	9.49	13.28	9.24	11.07	15.09
.49	1.08	3.47	4.73	7.82	5.42	6.86	10.30	7.19	8.83	12.58	8.93	10.63	14.64	10.39	12.28	16.34
.48	1.17	3.79	5.10	8.26	5.80	7.31	10.71	7.64	9.29	13.05	9.42	11.17	15.17	10.96	12.88	16.83
.47	1.27	4.02	5.38	8.65	6.12	7.67	11.01	7.98	9.62	13.39	9.82	11.63	15.91	11.40	13.27	17.32
.45	1.49	4.38	5.91	9.00	6.60	8.11	11.77	8.50	10.15	14.23	10.35	12.27	16.64	12.05	14.00	18.06
.40	2.25	5.10	6.57	9.82	7.45	9.02	12.91	9.46	11.17	14.88	11.39	13.32	17.66	13.09	15.16	19.23
.35	3.45	5.59	7.05	10.53	8.06	9.67	13.53	10.16	12.05	15.71	12.10	14.12	18.54	13.86	15.93	19.99
.30	5.44	6.05	7.51	10.91	8.57	10.19	14.16	10.76	12.58	16.24	12.80	14.79	19.10	14.58	16.48	20.67
.25	9.00	6.46	7.93	11.48	9.10	10.75	14.47	11.29	13.16	16.60	13.36	15.34	19.78	15.17	17.25	21.39
.20	16.00	6.80	8.45	11.69	9.59	11.26	15.09	11.80	13.69	17.28	13.82	15.84	20.24	15.63	17.88	21.90
.15	32.11	7.17	8.85	12.35	10.01	11.79	15.51	12.27	14.15	17.68	14.31	16.45	20.71	16.20	18.35	22.49
.10	81.00	7.63	9.31	12.69	10.50	12.27	16.04	12.81	14.62	18.28	14.94	16.98	21.04	16.87	18.93	23.34
.05	361.00	8.19	9.84	13.01	11.20	12.93	16.44	13.47	15.15	19.06	15.62	17.56	21.54	17.69	19.61	24.18

		$p=6$			$p=7$			$p=8$			$p=9$			$p=10$		
π_0	λ	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%	10%	5%	1%
.50	1.00	10.64	12.59	16.81	12.02	14.07	18.48	13.36	15.51	20.09	14.68	16.92	21.67	15.99	18.31	23.21
.49	1.08	11.81	13.74	18.32	13.27	15.52	19.93	13.29	15.63	20.53	16.17	18.56	23.05	17.35	19.79	24.62
.48	1.17	12.42	14.45	19.12	13.92	16.14	20.64	13.89	16.31	21.14	16.82	19.25	23.83	18.08	20.35	25.75
.47	1.27	12.90	14.86	19.64	14.32	16.63	21.14	14.43	16.74	21.72	17.26	19.74	24.80	18.67	20.92	26.43
.45	1.49	13.53	15.59	20.45	14.97	17.38	22.32	15.05	17.53	22.28	18.10	20.59	25.52	19.39	21.78	27.30
.40	2.25	14.71	16.91	21.60	16.23	18.41	23.35	16.26	18.73	23.63	19.56	22.12	26.86	20.74	23.15	28.86
.35	3.45	15.56	17.75	22.33	17.09	19.34	24.10	17.06	19.46	24.64	20.49	22.93	27.77	21.87	24.17	29.76
.30	5.44	16.32	18.46	23.06	17.74	20.01	24.86	17.90	20.36	25.64	21.27	23.65	28.50	22.73	25.05	30.74
.25	9.00	17.00	19.07	23.65	18.38	20.63	25.11	18.61	20.95	26.10	21.93	24.31	29.23	23.32	25.80	31.32
.20	16.00	17.56	19.64	24.27	19.04	21.07	25.72	19.17	21.47	26.76	22.54	24.91	29.92	24.00	26.42	31.98
.15	32.11	18.12	20.26	24.79	19.69	21.84	26.23	19.82	22.13	27.25	23.15	25.47	30.52	24.62	27.03	32.33
.10	81.00	18.78	20.82	25.21	20.32	22.51	26.91	20.45	22.87	27.69	23.77	26.16	31.15	25.39	27.87	32.95
.05	361.00	19.49	21.56	25.96	21.02	23.22	27.53	21.23	23.60	28.77	24.64	26.94	31.61	26.24	28.63	33.86

Example: We search for breaking points for IBM returns in the 3-factor FF model. Below, we plot all starting at *T**15:



Maximum F is **3.83** occurs in May 1993 (observation #243). Then, $\widehat{QLR}$ = **3.83** < **16.45** ⇒ cannot reject H_0 at 5% level.

Chow Test: Structural Change – Script in R

Chow Test for different breaking points, starting at T1.

```
y <- ibm_x;
x1 <-  Mkt_RF
x2 <-  SMB
x3 <- HML
T <- length(x1)
x0 <- matrix(1,T,1)
x <- cbind(x0,x1,x2,x3)
k <- ncol(x)
b <- solve(t(x)%*% x)%*% t(x)%*%y                 # b = (X'X)-1 X' y  (OLS regression)
e <- y - x%*%b                                    # regression residuals, e
RSS_R <- as.numeric(t(e)%*%e)                     # RSS for Restricted (no structural change)

T1 <- round(T * 1/5)                              # Trim  .20 of data
t <- T1                                           # t will be the counter for loop. Starts at T1.
T2 <- round(T * 4/5)                              # Trim  .20 of data
T_sam <- T2 - T1
All_F <- matrix(0,T_sam,1)                        # Matrix to accumulate the (T2-T1) F-tests
while (t <= T2) {                                 # Start while loop with counter t
y_1 <- y[1:t]
x_u1 <- x[1:t,]

b_1 <- solve(t(x_u1)%*% x_u1)%*% t(x_u1)%*%y_1          # b = (X'X)-1 X' y  (OLS regression)
e1 <- y_1 - x_u1%*%b_1                                  # regression residuals, e
RSS1 <- as.numeric(t(e1)%*%e1)                          # RSS for regime 1
kk = t+1
y_2 <- y[kk:T]
x_u2 <- x[kk:T,]
b_2 <- solve(t(x_u2)%*% x_u2)%*% t(x_u2)%*%y_2          # b = (X'X)-1 X' y  (OLS regression)
e2 <- y_2 - x_u2%*%b_2                                  # regression residuals, e
RSS2 <- as.numeric(t(e2)%*%e2)                          # RSS for regime 2
F <- ((RSS_R - (RSS1+RSS2))/k)/((RSS1+RSS2)/(T - 2*k))
kt <- t - T1 +1                                         # kt is an index that start at 1
All_F[kt] <- F                                          # add F-test to All_F according to kt
t = t+1
}
plot(All_F, col="red",ylab ="F-test", xlab ="Break Point")
title("F-test at different Break Points")
F_max <- max(All_F)                                      # Find the maximum F-test (QLR)
```

Chow Test: Structural Change –Remarks

The results are *conditional* on the breaking point –say, October 73 or Dec 2001.

The breaking point is usually unknown. It needs to be estimated.

It can deal only with one structural break –i.e., two categories!

The number of breaks is also unknown.

Characteristics of the data (heteroscedasticity –for example, regimes in the variance- and unit roots (high persistence) complicate the test.

In general, only asymptotic (consistent) results are available.

There are many modern tests that take care of these issues, but usually also with *non-standard* distributions.

Forecasting and Prediction

Objective: Forecast

Distinction: Ex post vs. Ex ante forecasting

– Ex post: RHS data are observed

– Ex ante (true forecasting): RHS data must be forecasted

Prediction and Forecast

- Prediction: Explaining an outcome, which could be a future outcome.
- Forecast: A particular prediction, focusing in a future outcome.

Example: Prediction: Given $\mathbf{x}^0$ $\Rightarrow$ predict $\mathbf{y}^0$.

Forecast: Given $\mathbf{x}^0_{t+1}$ $\Rightarrow$ predict $\mathbf{y}_{t+1}$. ¶

• Two types of predictions:

- In sample (prediction): The expected value of **y** (in-sample), given the estimates of the parameters.
- Out of sample (forecasting): The value of a future **y** that is not observed by the sample.

<u>Notation</u>:

- Prediction for T made at T: $\hat{Y}_T$.
- Forecast for $T+l$ made at T: $\hat{Y}_{T+l}$, $\hat{Y}_{T+l|T}$, $\hat{Y}_T(l)$.

where T is the forecast origin and l is the forecast horizon. Then,

$\hat{Y}_T(l)$: *l-step ahead* forecast = Forecasted value Y_{T+l} at time T.

• Any prediction or forecast needs an information set, I_T. This includes data, models and/or assumptions available at time T. The predictions and forecasts will be conditional on I_T.

For example, in-sample, $I_T = \{\mathbf{x}^0\}$ to predict $\mathbf{y}^0$.

Or in a time series context, $I_T = \{\mathbf{x}^0_{T-1}, \mathbf{x}^0_{T-2}, ..., \mathbf{x}^0_{T-q}\}$ to predict $\mathbf{y}_{t+l}$.

Then, the forecast is just the conditional expectation of Y_{T+l}, given the observed sample:

$$\hat{Y}_{T+l} = E[Y_{T+l}|X_T, X_{T-1}, \ \ldots, X_1]$$

Example: If $X_T = Y_T$, then, the one-step ahead forecast is*:*

$$\hat{Y}_{T+1} = E[Y_{T+1}|Y_T, Y_{T-1}, \ \ldots, Y_1]. \P$$

• The conditional expectation of Y_{T+l} is, in general, based on a model, the experience of the forecaster or a combination of both.

Example: We base the conditional expectation on the 3 FF factor model:

$$\hat{Y}_{T+l} = \text{E}[(\beta_0 + \beta_1 (\text{Mkt}_{\text{Ret}} - \text{r}_\text{f})_{T+l} + \beta_2 \text{SMB}_{T+l} + \beta_3 \text{HML}_{T+l})|\text{I}_\text{T}]$$

Note: The forecast of Y_{T+l} also needs a forecast for the driving variables in the model. We need a forecast for $\text{E}[(\text{Mkt}_{\text{Ret}} - \text{r}_\text{f})_{T+l} |\text{I}_\text{T}]$; $\text{E}[\text{SMB}_{T+l} |\text{I}_\text{T}]$; & $\text{E}[\text{HML}_{T+l} |\text{I}_\text{T}]$. ¶

In general, we will need a model for $\hat{X}_{T+l}$. Things can get complicated very quickly.

Keep in mind that the forecasts are a random variable. Technically speaking, they can be fully characterized by a pdf.

In general, it is difficult to get the pdf for the forecast. In practice, we get a point estimate (the forecast) and a C.I.

Later in this class, when we cover time series (Brooks Chapter 6), we go deeper into forecasting.

Forecasting and Prediction – Model Validation

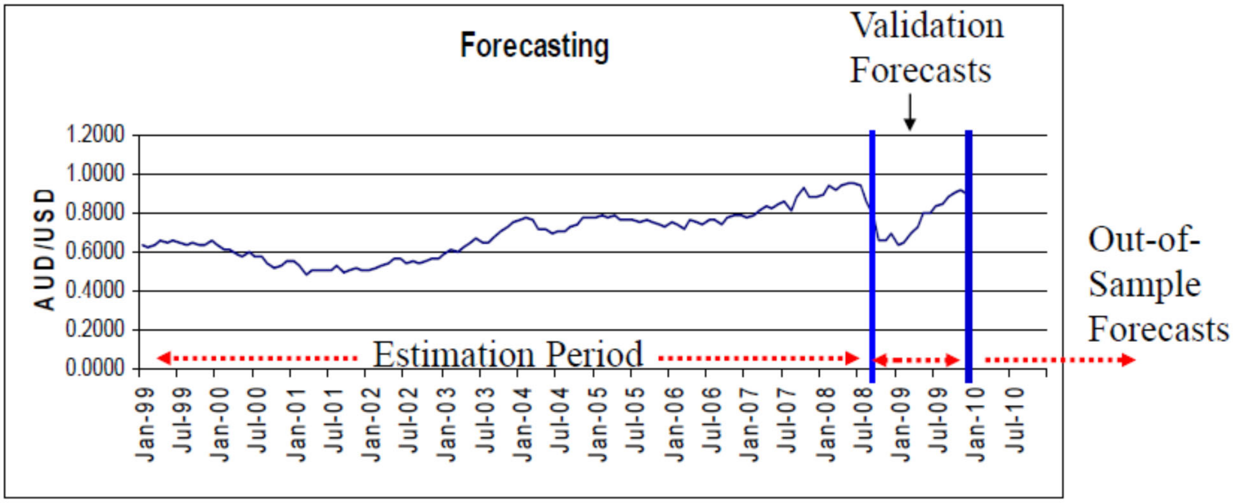
Prediction & model validation.

– Within sample prediction: Using the whole sample (T observations), we predict $\mathbf{y}$ as usual, with $\hat{y}$.

– Hold out sample: We estimate the model only using a part of the sample (say, up to time T_1). The rest of the sample ($T- \quad T_1$ observations) are used to check the predictive power of the model –i.e., the accuracy of predictions, by comparing $\mathbf{y}^0$ with actual $\mathbf{y}$.

Model validation refers to establishing the statistical adequacy of the assumptions behind the model –i.e., (**A1**)-(**A5**) in this lecture. Predictive power can be used to do model validation.

Prediction Intervals: Model Validation



Steps to measure forecast accuracy:
1) Select a (long) part of the sample (*estimation period*) to estimate the parameters of the model. (Get in-sample forecasts, $\hat{y}$.)
2) Keep a (short) part of the sample to check the model's forecasting skills. This is the *validation step*. You can calculate true MSE or MAE
3) If happy with Step 2), proceed to do out-of-sample forecasts.

Prediction Intervals: Point Estimate

Prediction: Given $\mathbf{x}^0 \Rightarrow$ predict $\mathbf{y}^0$.

Given the CLM, we have:

Expectation: $E[\mathbf{y}|\mathbf{X}, \mathbf{x^0}] = \boldsymbol{\beta}'\mathbf{x^0}$;
Predictor: $\hat{y}^0 = \mathbf{b}'\mathbf{x}^0$
Realization: $y^0 = \boldsymbol{\beta}'\mathbf{x^0} + \varepsilon^0$

Note: The predictor includes an estimate of ε^0:
$\hat{y}^0 = \mathbf{b}'\mathbf{x}^0 +$ estimate of ε^0. (Estimate of $\varepsilon^0=0$, but with variance.)

• Associated with the prediction (a point estimate), there is a forecast error (& a variance):

$$\hat{y}^0 - y^0 = \mathbf{b}'\mathbf{x^0} - \boldsymbol{\beta}'\mathbf{x^0} - \varepsilon^0 = (\mathbf{b} - \boldsymbol{\beta})'\mathbf{x^0} - \varepsilon^0$$

$$\Rightarrow \text{Var}[(\hat{y}^0 - y^0)|\mathbf{x^0}] = E[(\hat{y}^0 - y^0)'(\hat{y}^0 - y^0)|\mathbf{x^0}]$$
$$= \mathbf{x^0}'\text{Var}[(\mathbf{b} - \boldsymbol{\beta})|\mathbf{x^0}]\mathbf{x^0} + \sigma^2$$

Example: We have already estimated the 3 Factor Fama-French Model for IBM returns:

```
             Estimate     Std. Error   t value Pr(>|t|)
(Intercept)  -0.005089    0.002488   -2.046   0.0412 *
Mkt_RF        0.908299    0.056722   16.013   <2e-16 ***
SMB          -0.212460    0.084112   -2.526   0.0118 *
```

```
HML          -0.171500    0.084682   -2.025   0.0433 *
```

Suppose we are given $\mathbf{x}^0$ = [1.0000 -0.0189 -0.0142 -0.0027]
Then,

$\hat{y}^0$ = -0.005089 + 0.908299 * (-0.0189) -0.212460 * -0.0142 - 0.171500 * (-0.0027) =
= **-0.01877582**

Suppose we observe **y^0 = 0.1555214**. Then, the forecast error is

$\hat{y}^0 - y^0$ = **-0.01877582** - **0.1555214** = -0.1742973

- In R:

```
x_0 <- rbind(1.0000, -0.0189, -0.0142, -0.0027)
y_0 <- 0.1555214
y_f0 <-  t(b)%*% x_00
> y_f0
           [,1]
[1,] -0.01877582
ef_0 <- y_f0 - y_0
> ef_0
          [,1]
[1,] -0.1742973
```

Prediction Intervals: Confidence Intervals

How do we estimate the uncertainty behind the forecast? Form a confidence interval.

Two cases:
(1) If $\mathbf{x}^0$ is given –i.e., constants. Then,

$\text{Var}[\hat{y}^0 - y^0|\mathbf{x}^0] = \mathbf{x}^{0\prime}\, \text{Var}[\mathbf{b}|\mathbf{x}^0]\, \mathbf{x}^0 + \sigma^2$

$\Rightarrow$ Form confidence interval as usual.

Note: In out-of-sample forecasting, $\mathbf{x}^0$ is unknown, it has to be estimated.

(2) If $\mathbf{x}^0$ has to be estimated, then we use a random variable. What is the variance of the product? One possibility: Use bootstrapping.

- Assuming $\mathbf{x}^0$ is known, the variance of the forecast error is

$\sigma^2 + \mathbf{x}^{0\prime}\, \text{Var}[\mathbf{b}|\mathbf{x}^0]\mathbf{x}^0 = \sigma^2 + \sigma^2[\mathbf{x}^{0\prime}\, (\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}^0]$

If the model contains a constant term, this is

$$\text{Var}[e^0] = \sigma^2\left[1 + \frac{1}{N} + \sum_{j=1}^{K-1}\sum_{k=1}^{K-1}(x_j^0 - \bar{x}_j)(x_k^0 - \bar{x}_k)(Z'M^0Z)^{jk}\right]$$

(where **Z** is **X** without $\mathbf{x}_1 = \mathbf{i}$). In terms squares and cross products of deviations from means.

Note: Large σ^2, small N, and large deviations from the means, decrease the precision of the forecasting error.

Interpretation: Forecast variance is smallest in the middle of our "experience" and increases as we move outside it.

Then, the $(1 - \alpha)\%$ C.I. is given by: $[\hat{y}^0 \pm t_{T\text{-}k,\alpha/2} * \text{sqrt}(\text{Var}[e^0])]$

As $\mathbf{x}^0$ moves away from its mean, the C.I increases, this is known as the "*butterfly effect*."

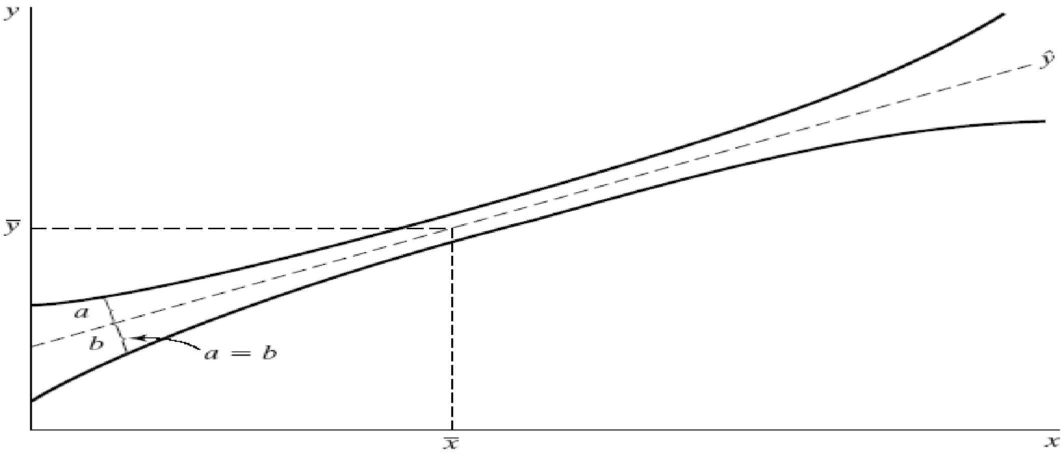


FIGURE 6.1 Prediction Intervals.

Example (continuation): We want to calculate the variance of the forecast error: for thee given $\mathbf{x}^0 = [1.0000\ \text{-}0.0189\ \text{-}0.0142\ \text{-}0.0027]$
Recall we got $\hat{y}^0 = \mathbf{b}'\mathbf{x}^0 =$ **-0.01877587**

Then,

$$\text{Estimated Var}[\hat{y}^0 - y^0|\mathbf{x}^0] = \mathbf{x}^{0\prime}\ \text{Var}[\mathbf{b}|\mathbf{x}^0]\ \mathbf{x}^0 + s^2 = \mathbf{0.003429632}$$

```
var_ef_0 <- t(x_0)%*% Var_b%*% x_0 + Sigma2
> var_ef_0
          [,1]
[1,] 0.003429632
> sqrt(var_ef_0)
         [,1]
[1,] 0.05856306
```

Check: What is the forecast error if $\mathbf{x}^0$ = colMeans(x)?

```
# (1-alpha)% C.I. for prediction         (alpha = .05)
CI_lb <- y_f0 – 1.96 * sqrt(var_ef_0)
> CI_lb
>[1] -0.1335594
```

```
CI_ub <- y_f0 + 1.96 * sqrt(var_ef_0)
>CI_ub
>[1] 0.09600778
```

That is, CI for prediction: [**-0.13356**; **0.09601**] with 95% confidence. A wide interval, which makes clear the uncertainty surrounding the point forecast: $\hat{y}^0$ = **-0.01877587**. ¶

Evaluation of Forecasts: Measures of Accuracy

Summary measures of out-of-sample forecast accuracy, after *m* forecasts:

Mean Error = $\frac{1}{m}\sum_{i=T+1}^{T+m}(\hat{y}_i - y_i) = \frac{1}{m}\sum_{i=T+1}^{T+m} e_i$

Mean Absolute Error (MAE) = $\frac{1}{m}\sum_{i=T+1}^{T+m} |\hat{y}_i - y_i| = \frac{1}{m}\sum_{i=T+1}^{T+m} |e_i|$

Mean Squared Error (MSE) = $\frac{1}{m}\sum_{i=T+1}^{T+m}(\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2$

Root Mean Square Error (RMSE) = $\sqrt{\frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2}$

Theil's U-stat = $\frac{\sqrt{\frac{1}{m}\sum_{i=T+1}^{T+m} {e_i}^2}}{\sqrt{\frac{1}{T}\sum_{i=1}^{T} {y_i}^2}}$

Theil's U statistics has the interpretation of an R^2. But, it is not restricted to be smaller than 1.

The lower the above criteria, say MSE, the better the forecasting ability of our model.

Question: We have two competing forecasting models. How do we know the MSE for model 1 is significantly better than the MSE for model 2? We need a test.

Example: We want to check the forecast accuracy of the 3 FF Factor Model for IBM returns. We estimate the model using only 1973 to 2017 data (T=539), leaving 2018-2020 (*m*=30 observations) for validation of predictions.

```
T0 <- 1
T1 <- 539                          # End of Estimation Period
T2 <- T1+1                         # Start of Validation Period
y1 <- y[T0:T1]
x1 <- x[T0:T1, ]

 fit2 <- lm(y1~ x1 - 1)            # Estimation Period Regression from T0 to T1
 b1 <- fit2$coefficients                  # Extract OLS coefficients from regression
> summary(fit2)

             Estimate   Std. Error   t value Pr(>|t|)
x1           -0.003848  0.002571   -1.497  0.13510
x1Mkt_RF     0.865579    0.059386   14.575  < 2e-16 ***
x1SMB        -0.224914  0.085505   -2.630  0.00877 **
```

```
x1HML         -0.230838   0.090251    -2.558  0.01081 *
```

We condition on the observed data (no model to predict FF factors used) from 2018: Jan to 2020: Jun.

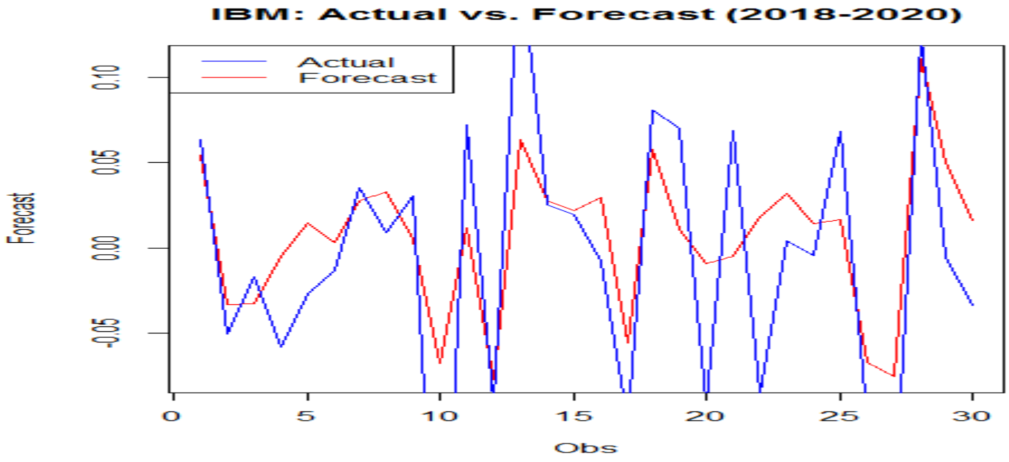
```
x_0 <- x[T2:T,]                                   # Validation data
 y_0 <- y[T2:T]                                   # Validation data
 y_f0 <-  x_0%*% b1                         # Forecast
 ef_0 <- y_f0 - y_0                         # Forecasat error
 mse_ef_0 <- sum(ef_0^2)/nrow(x_0)                # MSE
> mse_ef_0
[1] 0.003703207
> mae_ef_0 <- sum(abs(ef_0))/nrow(x_0)      # MAE
> mae_ef_0
[1] 0.04518326
```

That is,MSE = **0.003703207**

MAE = **0.04518326**

- Plot of actual IBM returns and forecasts.

```
plot(y_f0, type="l", col="red", main = "IBM: Actual vs. Forecast (2018-2020)", xlab = "Obs",
ylab = "Forecast")
lines(y_0, type = "l", col = "blue")
legend("topleft",  legend = c("Actual", "Forecast"),  col = c("blue", "red"),  lty = 1)
```



Evaluation of forecasts: Testing Accuracy

Suppose two competing forecasting procedures produce a vector of errors: $e^{(1)}$ & $e^{(2)}$. Then, if expected MSE is the criterion used, the procedure with the lower MSE will be judged superior.

- We want to test H_0: MSE(1) = MSE(2)
  H_1: MSE(1) ≠ MSE(2).

Assumptions: forecast errors are unbiased, normal, and uncorrelated. If forecasts are unbiased, then MSE = Variance.

Consider, the pair of RVs: $(e^{(1)} + e^{(2)})$ & $(e^{(1)} - e^{(2)})$. Now,

$$E[(e^{(1)}+e^{(2)})(e^{(1)}-e^{(2)})]=\sigma_1^2-\sigma_2^2$$

That is, we test H_0 by testing that the two RVs are not correlated! Under H_0,

$$E[(e^{(1)}+e^{(2)})(e^{(1)}-e^{(2)})]= 0.$$

This idea is due to Morgan, Granger and Newbold (MGN, 1977).

- There is a simpler way to do the MGN test. Let,

$$z_t = e_t^{(1)} + e_t^{(2)}$$
$$x_t = e_t^{(1)} - e_t^{(2)}$$

(1) Do a regression: $z_t = \beta\, x_t + \varepsilon_t$

(2) Test H_0: $\beta = 0$ ⇒ a simple *t-test*.

The MGN test statistic is exactly the same as that for testing the null hypothesis that $\beta = 0$ in this regression (recall: $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y}$). This is the approach taken by Harvey, Leybourne and Newbold (1997).

If the assumptions are violated, these tests have problems.

A non-parametric HLN variation: Spearman's rank test for zero correlation between x_t and z_t.

Example: We produce IBM returns one-step-ahead forecasts for 2018-2020 using the 3 FF Factor Model for IBM returns:

$$(\text{IBM}_{\text{Ret}} - r_f)_t = \beta_0 + \beta_1\,(\text{Mkt}_{\text{Ret}} - r_f)_t + \beta_2\,\text{SMB}_t + \beta_3\,\text{HML}_t + \varepsilon_t$$

Taking expectations at time $t+1$, conditioning on time t information set, $I_t=\{(\text{Mkt}_{\text{Ret}} - r_f)_t, \text{SMB}_t, \text{HML}_t\}$

$$E[(\text{IBM}_{\text{Ret}} - r_f)_{t+1}|I_t] = \beta_0 + \beta_1\, E[(\text{Mkt}_{\text{Ret}} - r_f)_{t+1}|I_t] + \beta_2\, E[\text{SMB}_{t+1}|I_t] + \beta_3\, E[\text{HML}_{t+1}|I_t]$$

In order to produce forecast, we will make a naive assumption: The best forecast for the FF factors is the previous observation. Then,

$$E[(\text{IBM}_{\text{Ret}} - r_f)_{t+1}|I_t] = \beta_0 + \beta_1\,(\text{Mkt}_{\text{Ret}} - r_f)_t + \beta_2\,\text{SMB}_t + \beta_3\,\text{HML}_t.$$

Now, replacing the β by the estimated **b**, we have our one-step-ahead forecasts. We produce one forecast at a time.

We compare the forecast accuracy relative to a random walk model for IBM returns. That is,

$$E[(\text{IBM}_{\text{Ret}} - r_f)_{t+1}|I_t] = (\text{IBM}_{\text{Ret}} - r_f)_t$$

Using R, we create the forecasting errors for both models and MSE:

```
x_01 <- x[T1:(T-1),]                      # By assumption on the X, it starts at T1.
y_0 <- y[T2:T]
y_f0 <-  x_01%*% b1                       # b1 coefficients from previous regresssion
ef_0 <- y_f0 - y_0                                  # e_t^(2)
mse_ef_0 <- sum(ef_0^2)/nrow(x_0)
> mse_ef_0                                # MSE(2)
[1] 0.01106811
```

```
ef_rw_0 <- y[T1:(T-1)] - y_0                    # e_t^(1)
mse_ef_rw_0 <- sum(ef_rw_0^2)/nrow(x_0)
> mse_ef_rw_0                                   # MSE(1)      <= (1) is the higher MSE.
[1] 0.02031009
```

- Now, we create

$$z_t = e_t^{(1)} + e_t^{(2)}, \;\&\; x_t = e_t^{(1)} - e_t^{(2)}.$$

Then, regress:

$$z_t = \beta\, x_t + \varepsilon_t \qquad \text{and test } H_0\text{: } \beta = 0.$$

```
z_mgn <- ef_rw_0 + ef_0
x_mgn <- ef_rw_0 - ef_0
fit_mgn <- lm(z_mgn ~ x_mgn)
> summary(fit_mgn)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05688    0.03512   1.619    0.117
x_mgn        2.77770    0.58332   4.762 5.32e-05 ***
```

Conclusion: We reject that both MSE are equal ⇒ MSE of RW is higher.

Evaluation of forecasts: MSE/MAE?

MSE and MAE are very popular criteria to judge the forecasting power of a model. However, it may not be the best measure for everybody.

Richard Levich's textbook compares forecasting services to the freely available forward rate. He finds that forecasting services may have some ability to predict direction (appreciation or depreciation).

For some investors, the direction is what really matters, since direction determines potential profits, not the error.

Example: Two forecasts: Forward Rate ($F_{t,T}$) and Forecasting Service (FS)

$F_{t,T=1\text{-month}}$ = **.7335** USD/CAD

$E_{FS,t}\,[S_{t+1\text{-month}}]$= **.7342** USD/CAD.

(Investor's stragey: buy CAD forward if FS forecasts CAD appreciation.)

Based on the FS forecast, Ms. Sternin decides to buy CAD forward at $F_{t,1\text{-m}}$.

(A) Suppose that the CAD appreciates to S_{t+1} = .7390 USD/CAD.

$\text{MAE}_{FS} = \left|.7390 - \mathbf{.7342}\right|$ = .0052 USD/CAD.

Investor makes a profit of .7390 - **.7335** = **USD .055** USD.

(B) Suppose that the CAD depreciates to S_{t+1} = .7315 USD/CAD.

$\text{MAE}_{\text{FS}} = \left| .7315 - \mathbf{.7342} \right|$ = .0027 USD/CAD. ⇒ smaller MAE!

Investor takes a loss of .7315 - **.7335** = **USD -.0020**. ¶

Forecasting Application: Fundamental Approach

There are two pure approaches to forecasting. Based on how we select the "driving" variables X_t, we have:

- Fundamental (based on data considered fundamental)
- Technical analysis (based on data that incorporates only past prices)

• Fundamental Approach to Forecast Exchange Rates, S_t (USD/JPY)

Based on an economic model, we generate $E_t[S_{t+T}] = E_t[f(X_{t+T})] = g(X_t)$, where X_t is a dataset regarded as *fundamental* economic variables:

- GNP growth rate,
- Current Account,
- Interest rates,
- Inflation rates, etc.

The economic model usually incorporates:

- Statistical characteristics of data (seasonality, autocorrelation, etc.)
- Experience of the forecaster (what information to use, lags, etc.)

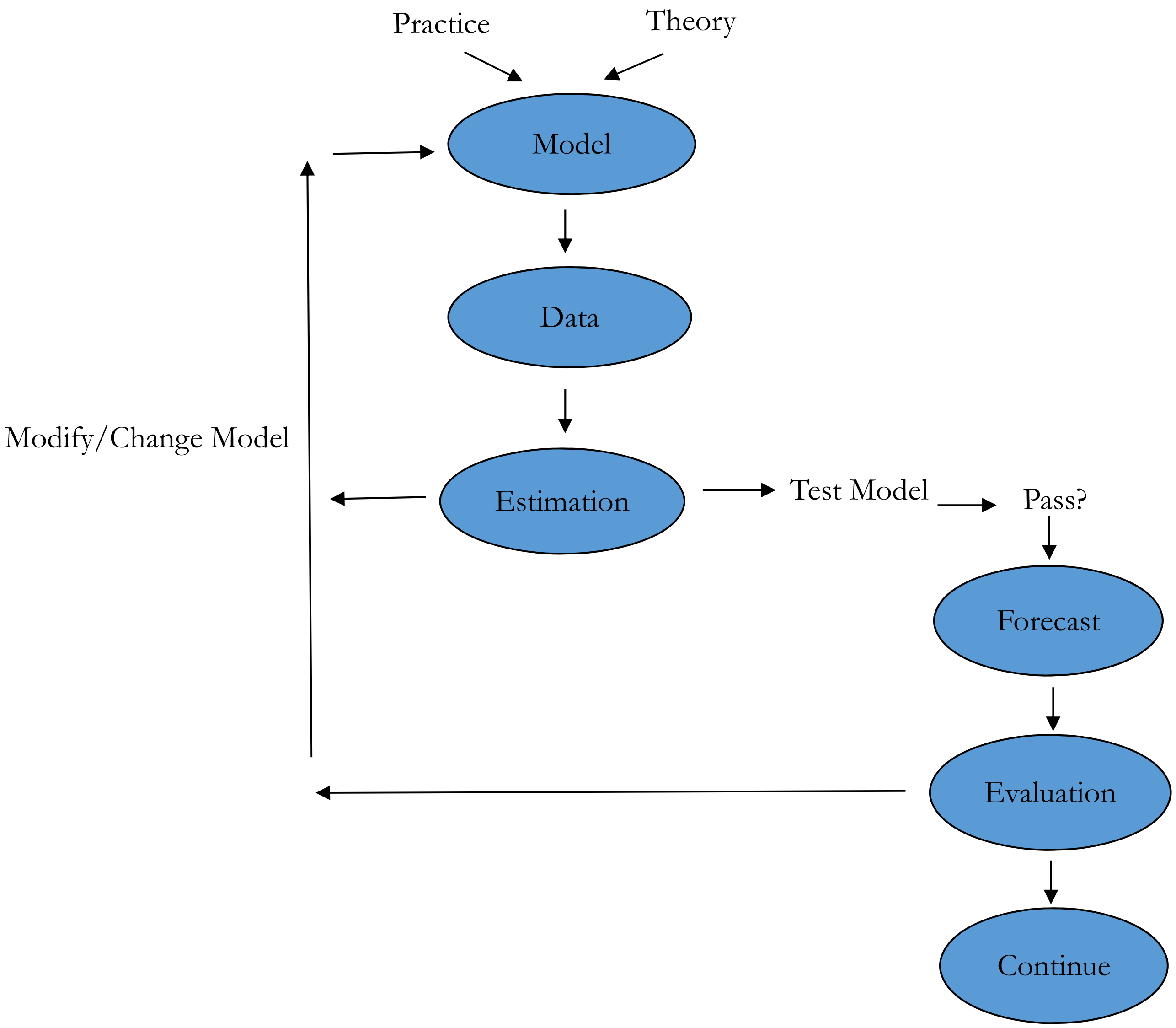
⇒ Mixture of art and science.

The economic model provides the structure for the forecasts (also called *structural model).*

• We compare the economic model's performance with the performance of a simpler model, the Random Walk (RW model), which is found to be very good model for S_t in the short-run. The forecasts for the RW are given by:

$$E_t[S_{t+1}] = S_t$$

- Steps



- Fundamental Forecasting: Steps (example: S_t = USD/JPY)

(1) Select a Model: Based on Theory (IFE, & Asset Approach)

$$e_t = \beta_0 + \beta_1 (i_{US,t} - i_{JAP,t}) + \beta_2 (y_{US,t} - y_{JAP,t}) + \beta_3 (m_{US,t} - m_{JAP,t}) + \varepsilon_t$$

$$E_t[e_{t+1}] = \beta_0 + \beta_1 E_t[i_{US,t} - i_{JAP,t}] + \beta_2 E[y_{US,t} - y_{JAP,t}] + \beta_3 E[m_{US,t} - m_{JAP,t}]$$

$$\Rightarrow E_t[S_{t+1}] = S^F_{t+1} = S_t * (1 + E_t[e_{t+1}])$$

(2) Collect data: S_t, $\mathbf{X}_t$ (Interest rates (i), GDP growth rates (y) and money growth (m) data needed.)

(3) Estimation of Model (using *estimation period*): OLS ⇒ get **b**.

(4) Generate forecasts. Assumptions about X_t are needed.

$$E_t[\mathbf{X}_{t+1}] = \delta_1 + \delta_2 (\mathbf{X}_t) \quad \text{-an AR(1) model.}$$

(5) Evaluation of Forecasts: MSE (& compare with RW's MSE).

Example: (1) & **(2)** Based on model I collect quarterly data (FX_USA_JAP.csv) from 1978:II – 2020:II. I read the data and transform it to estimate model:

```
 FX_da  <-  read.csv("http://www.bauer.uh.edu/rsusmel/4397/FX_USA_JAP.csv",  head=TRUE,
sep=",")
us_I <- FX_da$US_INF                       # Extract US Money growth (mUS) data from FX_da
us_i <- FX_da$US_I3M                       # Extract US 3-mo Interest rate (iUS) data
us_y <- FX_da$US_GDP_g                     # Extract US GDP growth (yUS) data
us_tb <- FX_da$US_CA_c                     # Extract US Current account change (tbUS) data
jp_I <- FX_da$JAP_INF                      # Extract Japan Inflation (IUS) data
jp_mg <- FX_da$JAP_MI_c                    # Extract Japan Money growth (mJP) data
jp_i <- FX_da$JAP_I3M                      # Read Japan 3-mo Interest rate (iJP) data
jp_y <- FX_da$JAP_GDP_g                    # Extract Japan GDP growth yJP) data
jp_tb <- FX_da$JAP_CA_c                    # Extract Japan Current account change (tbJP) data
e_f <- FX_da$JPY.USD_c                     # Extract changes in JPY/USD (e)

inf_dif <- us_I - jp_I                     # Define inflation rate differential (inf_dif)
int_dif <- us_i - jp_i                     # Define interest rate differential (int_dif)
mg_dif <- us_mg - jp_mg                    # Define money growth rate differential (mg_dif)
y_dif <- us_y - jp_y                       # Define income growth rate differential (y_dif)
tb_dif <- us_tb - jp_tb                    # Define Trade balance differential (tb_dif)

xx <- cbind(int_dif, y_dif, mg_dif)
T <- length(e_f)
T_est <- 161                     # Define final observation for estimation period.
e_f1 <- e_f[1:T_est]             # Adjust sample size to T_est
xx_1 <- xx[1:T_est,]             # Adjust sample size to T_est
```

(3) Estimation of model(using only *estimation period* (T=161): Get **b**.

```
fit_ef <- lm(e_f1 ~ xx_1)
> summary(fit_ef)
Call:
lm(formula = e_f1 ~ xx_1)

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.7246    0.6971   2.474   0.0144 *
xx_1int_dif   -0.5281   0.2478  -2.131   0.0346 *
xx_1y_dif     -0.2034   0.4538  -0.448   0.6546
xx_1mg_dif     0.1104   0.1912   0.577   0.5647
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.293 on 157 degrees of freedom
Multiple R-squared:  0.04673,   Adjusted R-squared:  0.02851
F-statistic: 2.565 on 3 and 157 DF,  p-value: 0.05661
```

(4) Generate Forecasts. Need first to estimate model for **X** variables. (using *estimation period* data only)

- AR(1) for ($i_{US,t} – i_{JAP,t}$)

```
int_dif_lag1 <- int_dif[1:T_est-1]                # Lag (iUS,t – iJAP,t)
int_dif_lag0 <- int_dif[2:T_est]                  # Adjust sample size (lost one observation
above)
fit_int <- lm(int_dif_lag0 ~ int_dif_lag1)        # Fit AR(1) model
> summary(fit_int)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.22774    0.11074   2.057   0.0414 *
int_dif_lag1  0.87537    0.03772  23.210   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.045 on 158 degrees of freedom
Multiple R-squared:  0.7732,    Adjusted R-squared:  0.7718
F-statistic: 538.7 on 1 and 158 DF,  p-value: < 2.2e-16
```

- AR(1) for ($m_{US,t} – m_{JAP,t}$)

```
mg_dif_lag1 <- mg_dif[1:T_est-1]                  # Lag (mUS,t – mJAP,t)
mg_dif_lag0 <- mg_dif[2:T_est]                    # Adjust sample size (lost one observation)
fit_mg <- lm(mg_dif_lag0 ~ mg_dif_lag1)           # Fit AR(1) model
> summary(fit_mg)

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.008708   0.216621  -0.040 0.967986
mg_dif_lag1  0.296597   0.076124   3.896 0.000144 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.74 on 158 degrees of freedom
Multiple R-squared:  0.08766,   Adjusted R-squared:  0.08188
F-statistic: 15.18 on 1 and 158 DF,  p-value: 0.000144
```

- AR(1) for ($y_{US,t} – y_{JAP,t}$)

```
y_dif_lag1 <- y_dif[1:T_est-1]                    # Lag (yUS,t – yJAP,t)
y_dif_lag0 <- y_dif[2:T_est]              # Adjust sample size (lost one observation above)
fit_y <- lm(y_dif_lag0 ~ y_dif_lag1)      # Fit AR(1) model
> summary(fit_y)
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.166258   0.086575   1.920   0.0566 .
y_dif_lag1     -0.008828   0.077255  -0.114   0.9092
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.08 on 158 degrees of freedom
Multiple R-squared:  8.263e-05, Adjusted R-squared:  -0.006246
F-statistic: 0.01306 on 1 and 158 DF,  p-value: 0.9092
```

- Now, we can do *one-step-ahead* forecast for the **X** variables:

```
T_val <- T_est+1                                          # start of Validation period
xx_cons <- rep(1,T-T_val+1)                               # create the constant vector
int_dif_0 <- cbind(xx_cons,xx[T_val:T,1]) %*% fit_int$coeff      # 8 forecasts for (iUS,t – iJAP,t)
mg_dif_0 <- cbind(xx_cons,xx[T_val:T,2]) %*% fit_mg$coeff    # 8 forecasts for (mUS,t – mJAP,t)
y_dif_0 <- cbind(xx_cons,xx[T_val:T,3]) %*% fit_y$coeff      # 8 forecasts for (yUS,t – yJAP,t)
```

- Finally, we compute the *one-step-ahead* forecast for **e** and MSE:

```
e_Mod_0 <- cbind(xx_cons,int_dif_0,mg_dif_0,y_dif_0)%*%fit_ef$coeff # Model’s forecast
f_e_Mod <- e_f[T_val:T] - e_Mod_0                              # Model’s forecast error
mse_e_f <- sum(f_e_Mod^2)/(T-T_val+1)                     # Model’s MSE
> mse_e_f
[1] 3.598407
```

- Compute the *one-step-ahead* forecast for RW Model and MSE **e**:

```
e_f_RW_0 <- rep(0,T-T_val+1)                    # RW forecast = 0 (always 0, for all t+T!)
f_e_RW <- e_f[T_val:T] - e_f_RW_0                    # RW’s forecast error
mse_e_RW <- sum(f_e_RW^2)/(T-T_val+1)                # RW’s MSE
> mse_e_RW
[1] 3.381597
```

⇒ Lower MSE than Model. Not good for Model.

- Compare MSEs: The RW model has a better MSE (usual finding).

- A MGN test is usually done. But, we have only *m*=8 observations, we can do the test, but the results are very likely not to be taken seriously.

(5) Evaluation of Forecasts

- MGN/HLN test:

```
z_mgn <- e_Mod + e_RW
x_mgn <- e_Mod - e_RW
fit_mgn <- lm(z_mgn ~ x_mgn)
```

```
> summary(fit_mgn)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.8361     1.9665   0.425    0.686
x_mgn       1.1327     1.3330   0.850    0.428          ⇒ not significant, but unreliable (a very small sample).

Residual standard error: 3.026 on 6 degrees of freedom      ⇒very small number for df to make inferences.
Multiple R-squared:  0.05322,   Adjusted R-squared:  -0.1046
F-statistic: 0.3373 on 1 and 6 DF,  p-value: 0.5826
```

• Suppose you are happy with the Model, you believe the difference in MSEs is not significant), now you generate out-of-sample forecasts.

(6) Out-of-sample one-step-ahead forward forecast for S_t:

$$E_{t=2020:II}[S_{t+1=2020:III}] = S_{t=2020:II}\,(1 + E_{t=2020:II}[e_{t+1=2020:II}])$$

We observe S_t today (2020:II): $S_{2020:II}$ = 100.77 JPY/USD, which we invert since we work with direct quotes: $S_{2020:II}$ = 0.009279 USD/JPY.

We need to forecast the independent variables, based on AR(1) results,

$$\mathbf{X}_t = \{(i_{US,t} - i_{JAP,t}),\ (y_{US,t} - y_{JAP,t}),\ (m_{US,t} - m_{JAP,t})\}$$

• Forecasting $(i_{US,t+1} - i_{JAP,t+1})$: $E_{t=2020:II}[(i_{US,t} - i_{JAP})_{t+1=2020:III}]$

```
int_dif_p1 <- cbind(1,int_dif[T]) %*% fit_int$coeff# int_dif_p1 = E_{t=2020:II}[(i_{US,t} – i_{JAP})_{t+1=2020:III}]
> int_dif_p1
          [,1]
[1,] 0.4684645
```

• Forecasting $(m_{US,t} - m_{JAP,t})$: $E_{t=2020:II}[(m_{US,t} - m_{JAP})_{t+1=2020:III}]$

```
mg_dif_p1 <- cbind(1,m_dif[T]) %*% fit_m$coeff  #    mg_dif_p1    =    E_{t=2020:II}[(m_{US,t} – m_{JAP})_{t+1=2020:III}]
> mg_dif_p1
         [,1]
[1,] 4.921977
```

• Forecasting $(y_{US,t} - y_{JAP,t})$: $E_{t=2020:II}[(y_{US,t} - y_{JAP})_{t+1=2020:III}]$

```
y_dif_p1 <- cbind(1,y_dif[T]) %*% fit_y$coeff       # y_dif_p1 = E_{t=2020:II}[(y_{US,t} – y_{JAP})_{t+1=2020:III}]
> y_dif_p1
         [,1]
[1,] 0.176617
```

• Forecasting $E_{t=2020:II}[S_{t+1=2020:III}]$

```
S <- 0.009279                                   # Today's value of S_{t=2020:II}
```

```
e_f_p1 <- cbind(1,int_dif_p1,mg_dif_p1,y_dif_p1)%*%fit_ef$coeff        # Today's forecast for $e_{t=2020:III}$
> e_f_p1                                            # Print forecast for $e_{t=2020:III}$
       [,1]
[1,] 0.4955111                      ⇒ 0.50% depreciation of USD against JPY in 3rd Quarter.

S_p1 <- S*(1+e_f_p1/100)                            # Today's forecast for $S_{t=2020:III}$

> S_p1 <- S*(1+e_f_p1/100)                  # e is in %, we divide by 100 to put it decimal from
> S_p1                                      # Print forecast for $S_{t=2020:III}$
       [,1]
[1,] 0.009324999                    ⇒ Model's forecast for $S_{t+1=2020:III}$ =
```

⇒ Model's forecast for $S_{t+1=2020:III} = E_{t=2020:II}[S_{t+1=2020:III}] =$ **0.009324999 USD/JPY**.
(using the indirect quote, $E_{t=2020:II}[S_{t+1=2020:III}] =$ 107.2386 JPY/USD).

• We can use the one-step-ahead forecasts to generate *two-step-ahead* forecasts. That is, we forecast $E_{t=2020:II}[S_{t+1=2020:IV}]$ (=S_p2 below)

```
S1 <- S_p1                                          # Today's forecast for $S_{t+1=2020:III}$
int_dif_p2 <- cbind(1,int_dif_p1)%*%fit_int$coeff   # Today's forecast for $(i_{US} – i_{JP})_{t+2}$
mg_dif_p2 <- cbind(1,mg_dif_p1)%*%fit_mg$coeff      # Today's forecast for $(m_{US} – m_{JP})_{t+2}$
y_dif_p2 <- cbind(1,y_dif_p1)%*%fit_y$coeff         # Today's forecast for $(y_{US} – y_{JP})_{t+2}$
e_f_p2 <- cbind(1,int_dif_p2,mg_dif_p2,y_dif_p2)%*%fit_ef$coeff # Today's forecast for $e_{t=2020:IV}$
> e_f_p2
       [,1]
[1,] 1.110734               ⇒ 1.11% depreciation of USD against JPY in 4th Quarter.
S_p2 <- S1*(1+e_f_p2/100)
> S_p2
          [,1]
[1,] 0.009382085
```

⇒ $E_{t=2020:II}[S_{t+1=2020:III}] =$ **0.009382085 USD/JPY.**

• We can use the two-step-ahead forecast to generate *three-step-ahead* forecasts. Obviously, we can continue this process to generate *l-step-ahead* forecasts for S_t (a simple do loop will do it).

Eventually, we will collect *m* of out-of-sample forecasts (*m* one-step-ahead forecasts, *m* two-step-ahead forecasts, *m* three-step-ahead forecasts, etc.) to get an MSE and run a MGN/HLN test on them.

It is possible that one model is the best in the short-term (say, up to 3 steps ahead); other is better in the medium-term (say, from 4 to 6 steps ahead); and another is best for longer-term. For example, the RW model is very good ("*unbeatable*") up to 3 months ahead. Then, other models start to produce better forecasts, especially after 6 months.

Forecasting Application: Fundamental Approach

Practical Issues in Fundamental Forecasting

- Are we using the "right model?"
- Estimation of the model (OLS, MLE, other methods).
- Some explanatory variables (X_{t+T}) are contemporaneous.

⇒ We also need a model to forecast the X_{t+T} variables.

• Does Forecasting Work?

For exchange rates, in the short-run, RW models beat structural (and other) models: Lower MSE, MAE.

Many argue that the structural models used are not the "right model."

Model Selection Strategies

Specifying the DGP in (**A1**) is the most important step in applied work. We have assumed "correct specification," which, in practice, is an unrealistic assumption, since we do not really observed the true DGP.

A bad model can create a lot of problems: biases, wrong inferences, bad forecasts, etc.

So far, we have implicitly used a simple strategy:
(1) We started with a DGP, which we assumed to be true.
(2) We tested some H_0 (from economic theory).
(3) We used the model (restricted, if needed) for prediction & forecasting.

Question: How do we propose and select a model (a DGP)?
Potentially, we have a huge number of possible models (different functional form, *f(.)*, and explanatory variables, **X**). Say, we have

Model 1 $\quad \mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$
Model 2 $\quad \mathbf{Y} = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\xi}$
Model 3 $\quad \mathbf{Y} = (\mathbf{W}\boldsymbol{\gamma})^{\lambda} + \boldsymbol{\eta}$
Model 4 $\quad \mathbf{Y} = \exp(\mathbf{Z}\,\mathbf{D}\,\boldsymbol{\delta}) + \boldsymbol{\epsilon}$

We want to select the best model, the one that is closest to the true and unobserved DGP. In practice, we aim for a good model.

Model Selection Strategies: Views

A model is a simplification. Many approaches:

- "Pre-eminence of theory." Economic theory should drive a model. Data is only used to quantify theory. Econometric methods offer sophisticated ways 'to bring data into line' with a particular theory.
- Purely data driven models. Success of ARIMA models (late 60s – early 70s), discussed in Lecture 6:
No theory, only exploiting the time-series characteristics of the data to build models.
- Modern (LSE) view. A compromise: theory and the characteristics of the data are used to build a model.

• Theory and practice play a role in deriving a good model. David Hendry (2009) emphasizes:

"This implication is not a tract for mindless modeling of data in the absence of economic analysis, but instead suggests formulating more general initial models that embed the available economic theory as a special case, consistent with our knowledge of the institutional framework, historical record, and the data properties."

"Applied econometrics cannot be conducted without an economic theoretical framework to guide its endeavours and help interpret its findings. Nevertheless, since economic theory is not complete, correct, and immutable, and never will be, one also cannot justify an insistence on deriving empirical models from theory alone."

Model Selection Strategies: A Good Model

According to David Hendry, a good model should be:
- Data admissible -i.e., modeled and observed **y** should have the same properties.
- Theory consistent -our model should "make sense"
- Predictive valid -we should expect out-of-sample validation
- Data coherent -all information should be in the model. Nothing left in the errors (*white noise errors*).
- Encompassing -our model should explain earlier models.

That is, we are searching for a statistical model that can generate the observed data (**y**, **X**), this is usually referred as *statistical adequacy*, makes theoretical sense and can explain other findings.

Model Selection Strategies: FAQ

FAQ in practice:
- Should I include all the variables in the database in my model?
- How many explanatory variables do I need in my model?
- How many models do I need to estimate?
- What functional form should I be using?
- Should the model allow for structural breaks?
- Should I include dummies & interactive dummies?
- Which regression model will work best and how do I arrive at it?

Model Selection Strategies: Important Concepts

Diagnostic testing: We test assumptions behind the model. In our case, assumptions (**A1**)-(**A5**) in the CLM.
Example: Test $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$ -i.e., the residuals are zero-mean, uncorrelated with anything (that is, white noise distributed errors).

In selecting a model, this is a very important step. We run a lot of test to check the residuals are acceptable or the model is not misspecified: Ramsey's reset test, tests for autocorrelation, etc.

Parameter testing: We test economic H_0's.
Example: Test $\boldsymbol{\beta}_k = 0$ -say, there is no size effect on the expected return equation.

Model Selection Strategies: Two Methods

There are several *model-selection methods*. We will consider two:
- *Specific to General*
- *General to Specific*

- Specific to General. Start with a small "restricted model," do some testing and make model bigger model in the direction indicated by the tests (for example, add variable x_k when test reject H_0: β_k=0).

- General to Specific. Start with a big "general unrestricted model," do some testing and reduce model in the direction indicated by the tests (for example, eliminate variable x_k when test cannot reject H_0: β_k=0).

Model Selection Strategies: Specific to General

Steps:
(1) Begin with a small theoretical model – for example, the CAPM
$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.
(2) Estimate the model – say, using OLS.
(3) Do some diagnostic testing – are residuals white noise?
If the assumptions do not hold, then use:
- More advanced econometrics – GLS instead of OLS?
- A more general model – More regressors? Lags?
(4) Test economic H_0 on the parameters – Is size significant?
(5) Modify model in (1) in the direction of rejections of H_0.

• This strategy is known as *specific to general*.

Example: Specific-to-general strategy to model IBM returns:
(1) We start with the 3-factor FF model for IBM:

$$(IBM_{Ret} - r_f)_t = \beta_0 + \beta_1 (Mkt_{Ret} - r_f)_t + \beta_2 SMB_t + \beta_3 HML_t + \varepsilon_t$$

(2) Estimate the 3-factor FF model for IBM:

```
fit_r <- lm (ibm_x ~ Mkt_RF + SMB + HML)
```

```
> summary(fit_r)

Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  -0.005191  0.002482  -2.091   0.0369 *
Mkt_RF       0.910379   0.056784  16.032    <2e-16 ***
SMB          -0.221386  0.084214  -2.629   0.0088 **
HML          -0.139179  0.084060  -1.656   0.0983 .
---

Residual standard error: 0.05842 on 566 degrees of freedom
Multiple R-squared:  0.3393,    Adjusted R-squared:  0.3358
F-statistic:  96.9 on 3 and 566 DF,  p-value: < 2.2e-16
```

(3) Diagnostic tests: Check t-values & R^2, F-test goodness of fit, etc.

(4) LM Test to test if there is a January Effect (H_0: No January effect):

```
> LM_test
[1] 9.084247              ⇒ LM_test > 3.84 ⇒ Reject H0.
```

(5) Given this result, we modify the 3-factor FF and add the January Dummy to the FF model:

```
fit_new <- lm (ibm_x ~ Mkt_RF + SMB + HML + Jan_1)
> summary(fit_new)
Coefficients:
             Estimate  Std. Error t value Pr(>|t|)
(Intercept)  -0.007302 0.002561  -2.851  0.00452 **
Mkt_RF       0.905182  0.056405  16.048  < 2e-16 ***
SMB          -0.247691 0.084063  -2.946  0.00335 **
HML          -0.154093 0.083606  -1.843  0.06584 .
Jan_1        0.026966  0.008906   3.028  0.00258 **
```

• Some remarks based on the previous example:

• The specific-to-general method makes assumptions along the way.
(1) Very likely the starting model is based on theory and experience (HML is not significant at the usual 5% level). Not clear how to proceed from there to a more general model.

(2) We tested for a January effect and then added to the model. However, we could have tested for a Dot.com effect or for an interactive Dot.com/January effect with the 3 FF factors. Not clear when to stop the search.

(3) Select used a p-value to add variables to the model. In this case, we use the standard 5% for the tests.

Model Selection Strategies: General to Specific

Begin with a *general unrestricted model* (GUM), which nests restricted models and, thus, allows any restrictions to be tested. Say:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \mathbf{W}^{\lambda}\boldsymbol{\delta} + (\mathbf{X} * \mathbf{W})\boldsymbol{\zeta} + (\mathbf{Z} * \mathbf{D})\boldsymbol{\psi} + \boldsymbol{\varepsilon}.$$

Then, reduction of the GUM starts. Mainly using *t-tests*, and *F-tests*, we move from the GUM to a smaller, more parsimonious, specific model. If competing models are selected, encompassing tests or information criteria (AIC, BIC) can be used to select a final model. This is the *discovery stage.* After this reduction, we keep a final (restricted GUM) model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Creativity is needed for the specification of a GUM. Theory and empirical evidence play a role in designing a GUM.

• Steps:

Step 1 - First ensure that the GUM does not suffer from any diagnostic problems. Check residuals in the GUM to ensure that they possess acceptable properties. (For example, test for white noise in residuals, incorrect functional form, autocorrelation, etc.).

Step 2 - Test the restrictions implied by the specific model against the general model – either by exclusion tests or other tests of linear restrictions.

Step 3 - If the restricted model is accepted, test its residuals to ensure that this more specific model is still acceptable on diagnostic grounds.

• This strategy is called *general to specifics* ("*gets*"), *LSE*, *TTT* (Test, test, test). It was pioneered by Sargan (1964). The properties of gets are discussed in Hendy and Krolzig (2005, Economic Journal).

• The role of diagnostic testing is two-fold.

- In the *discovery steps* (Steps 1 & 2), the tests are being used as design criteria. Testing plays the role of checking that the original GUM was a good starting point after the GUM has been simplified.

- In the context of model evaluation (Step 3), the role of testing is clear cut. Suppose you use the model to produce forecasts. These forecasts can be evaluated with a test. This is the critical evaluation of the model.

Example: General-to-specific strategy to model IBM returns:

Step 1 - Start with a GUM: the 3-factor FF model for IBM + January Dummy + Dot.com Dummy + non-linear & interactive effects:

$$\begin{aligned}(\mathrm{IBM}_{\mathrm{Ret}} - r_f)_t = {} & \beta_0 + \beta_1\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t + \beta_2\,\mathrm{SMB}_t + \beta_3\,\mathrm{HML}_t + \beta_4\,\mathrm{January}_t + \beta_5\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t^2 \\ & + \beta_6\,\mathrm{SMB}_t^2 + \beta_7\,\mathrm{HML}_t^2 + \beta_8\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t{*}\mathrm{SMB}_t + \beta_9\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t{*}\mathrm{HML}_t + \\ & + \beta_{10}\,\mathrm{Dot.com}_t + \beta_{11}\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t * \mathrm{January}_t + \beta_{12}\,\mathrm{HML}_t * \mathrm{January}_t \\ & + \beta_{13}\,(\mathrm{Mkt}_{\mathrm{Ret}} - r_f)_t{*}\,\mathrm{Dot.com}_t + \beta_{14}\,\mathrm{HML}_t * \mathrm{Dot.com} + \beta_{15}\,\mathrm{SMB}_t * \mathrm{Dot.com} + \varepsilon_t\end{aligned}$$

Estimate GUM:

```
t_sb <- 342                              # Structural break date (End of 1st-regime)
T_s_1 <- T - t_sb
d_0 <- matrix(0, t_sb, 1)                # Dot.com dummy = 0 before t_sb
d_1 <- matrix(1, T_s_1, 1)               # Dot.com dummy = 1 after t_sb
Dot_com <- rbind(d_0,d_1)                # Dot.com dummy (join rows d_0 & d_1)
Mkt_Jan <- Mkt_RF * Jan_1
HML_Jan <- HML * Jan_1
Mkt_Dot <- Mkt_RF * Dot_com
HML_Dot <- HML * Dot_com
SMB_Dot <- SMB * Dot_com
```

```
fit_gum <- lm (ibm_x ~ Mkt_RF + SMB + HML + Jan_1 + Mkt_RF_2 + SMB_2 + HML_2 +
Mkt_HML + Mkt_SMB + SMB_HML + Mkt_Jan + HML_Jan + Mkt_Dot + HML_Dot +
SMB_Dot)
> summary(fit_gum)

Coefficients:
              Estimate  Std. Error t value Pr(>|t|)
(Intercept)  -0.007836  0.003063  -2.559 0.010772 *
Mkt_RF        0.791866  0.090474   8.752  < 2e-16 ***
SMB          -0.295790  0.110655  -2.673 0.007738 **
HML          -0.233942  0.135146  -1.731 0.084004 ⇒  practice says “keep it.” Judgement call.
Jan_1         0.031769  0.009349   3.398 0.000727 ***
Mkt_RF_2     -0.433762  0.850899  -0.510 0.610417
SMB_2        -0.927271  1.470645  -0.631 0.528615
HML_2         2.707992  1.670366   1.621 0.105545 ⇒ almost 10%, I keep it. Judgement call.
Mkt_HML       0.628721  1.557090   0.404 0.686531
Mkt_SMB       0.791625  1.746939   0.453 0.650618
SMB_HML      -1.044806  2.029091  -0.515 0.606819
Mkt_Jan      -0.069413  0.189309  -0.367 0.714008
HML_Jan      -0.259697  0.255484  -1.016 0.309841
Mkt_Dot       0.323382  0.130645   2.475 0.013612 *
HML_Dot       0.059742  0.208277   0.287 0.774342
SMB_Dot       0.076998  0.198964   0.387 0.698910
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05788 on 553 degrees of freedom
Multiple R-squared:  0.3663, Adjusted R-squared:  0.3491
F-statistic: 21.31 on 15 and 553 DF,  p-value: < 2.2e-16
```

Step 1 – Check GUM residuals for departures of (**A2**)-(**A3**). A Ramsey’s reset test can be done (using the *resettest* in the *lmtest* library).

```
> resettest(fit_gum, type="fitted")
        RESET test
data:  fit_gumHomework 1 Review <br>
RESET = 1.2645, df1 = 2, df2 = 552, p-value = 0.2832
```

Step 2 – Reduce Model with t-test and F-tests. Say, we keep all the variables with a p-value close to 10% (we still keep HML, using previous experience). We estimate a restricted GUM:

```
fit_gum_r <- lm (ibm_x ~ Mkt_RF + SMB + HML + Jan_1 + HML_2 + Mkt_Dot)
> summary(fit_gum_r)

Coefficients:
               Estimate   Std. Error t value Pr(>|t|)
(Intercept)   -0.008696   0.002788  -3.119  0.00191 **
Mkt_RF         0.779336   0.072453  10.756  < 2e-16 ***
SMB           -0.280018   0.083891  -3.338  0.00090 ***
HML           -0.250480   0.088504  -2.830  0.00482 **
Jan_1          0.028499   0.008937   3.189  0.00151 **
HML_2          1.676011   1.331161   1.259  0.20853
Mkt_Dot        0.344030   0.116685   2.948  0.00333 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05761 on 562 degrees of freedom
Multiple R-squared:  0.3618,  Adjusted R-squared:  0.355
F-statistic: 53.11 on 6 and 562 DF,  p-value: < 2.2e-16
```

Step 2 – Test the restrictions implied by the specific model against the general model. Using an F-test, we test *J*=9 restrictions:

H_0: $\beta_5 = \beta_6 = \beta_8 = \beta_9 = \beta_{10} = \beta_{11} = \beta_{12} = \beta_{14} = \beta_{15}$.

```
e_u <- fit_gum$residuals                                  GUM residuals
RSS_u <- t(e_u)%*%e_u
e_r <- fit_gum_r$residuals            pf                 # Restricted GUM residuals
RSS_r <- t(e_r)%*%e_r
f_test_gum <- ((RSS_r - RSS_u)/9)/(RSS_u/(T-16))         # F-test
> f_test_gum
        [,1]
[1,] 0.4299497            ⇒ we cannot reject H0 (f_test_gum < qchisq(.95,9, 553) = 1.896801)
> qf(.95, df1=9, df2=T-16)
[1] 1.896801
p_val <- 1 - pf(f_test_gum, df = 9 , df2=T-16)           # p-value of F-test
>  p_val
[1,] 0.919105             ⇒ p-value is very high. No evidence for H0.
```

Step 2 – Further specification checks of Restricted GUM, for example, perform a Ramsey's reset test (using the *resettest* in the lmtest library).

```
> resettest(fit_gum_r, type="fitted")

	RESET test

data:  fit_gum_r
RESET = 1.1361, df1 = 2, df2 = 561, p-value = 0.3218
```

Step 3 - Test if Restricted GUM residuals are acceptable –i.e., do diagnostic tests (mainly, make sure they are white noise). If Restricted GUM passes all the diagnostic tests, it becomes the "final model."

Note: With the final model, we use it to justify/explain financial theory and features, and do forecasting.

• Some remarks based on the previous example:

The general-to-specific method makes assumptions along the way.

(1) Select a p-value for the tests of significance in the discovery stage (we use **10%**). Given that we performed **15** *t-tests*, we should not be surprised we rejected the GUM, since we had an overall significance, $\alpha^* = .79$ [= 1 – (1 - **.10**)^**15**]. Mass significance is an issue.

(2) Judgement calls are also made.

(3) The reduction of the GUM involves "pre-testing" –i.e., data mining. We are likely rejecting a true H_0 (false positives) and not rejecting a true H_1, (false negatives) along the way. This increases the probability that the final model is not a good approximation. It is common to ignore (or not even acknowledge) pre-testing issues.

Model Selection Strategies: Properties

A modeling strategy is *consistent* if its probability of finding the true model tends to 1 as T -the sample size- increases.

• Properties for strategies

(1) Specific to General
- It is not consistent if the original model is incorrect.
- It need not be predictive valid, data coherent, & encompassing.
- No clear stopping point for an unordered search.

(2) General to Specific
- It is consistent under some circumstances. But, it needs a large T.
- It uses data mining, which can lead to incorrect models for small T.
- The significance levels are incorrect. This is the problem of *mass significance*.

Lecture 7 - Departures from CLM Assumptions & the Generalized Regression Model

Review of CLM Results

Recall the CLM Assumptions
(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$
(**A3**) $Var[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\,\mathbf{I}_T$
(**A4**) $\mathbf{X}$ has full column rank –rank($\mathbf{X}$) = k–, where $T \geq k$.

• OLS estimation:

$\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\,\mathbf{y}$
$Var[\mathbf{b}|\mathbf{X}] = \sigma^2\,(\mathbf{X}'\mathbf{X})^{-1}$
$\Rightarrow$ **b** unbiased and efficient (MVUE)

• If (**A5**) $\boldsymbol{\varepsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_T)$ $\Rightarrow \mathbf{b}|\mathbf{X} \sim N(\beta, \sigma^2(\mathbf{X}'\,\mathbf{X})^{-1})$
Under (**A5**), **b** is also the MLE (consistency, efficiency, invariance, etc). (**A5**) gives us *finite sample* results for **b** (and for tests: *t-test*, *F-test*, Wald tests).

CLM: Departures from the Assumptions

So far, we have discussed some violations of CLM Assumptions:
(1) (**A1**) – OLS can easily deal with some non-linearities in the DGP.
$\Rightarrow$ as long as we have intrinsic linearity, **b** keeps its nice properties.
– Wald, F, & LM tests to check for misspecification

(2) (**A4**) – Multicollinearity is a potential problem. In general, exogenous to the researcher. We need to be aware of this problem.

• In this lecture, we examine assumptions (**A2**), (**A3**) and (**A5**). That is, we check
(i) **X** is stochastic. That is, it has a distribution.
(ii) $Var[\boldsymbol{\varepsilon}|\mathbf{X}] \neq \sigma^2\,\mathbf{I}_T$
(iii) $\boldsymbol{\varepsilon}|\mathbf{X}$ is not $N(\mathbf{0}, \sigma^2\mathbf{I}_T)$

CLM: Departures from (A2)

The traditional derivation of the CLM assumes **X** as non-stochastic. In our derivation, however, we allowed **X** to be stochastic, but we conditioned on observing its realizations (an elegant trick, but not very realistic).

With stochastic **X** we need additional assumptions to get unbiasedness and consistency for the OLS **b**.
– We need independence between **X** & **ε**: $\{x_i, \varepsilon_i\}$ i=1, 2, ..., T is a sequence of independent observations.

– We require that **X** have finite means and variances. Similar requirement for ε, but we also require E[ε] = **0**.
Then,

$$E[\mathbf{b}] = \boldsymbol{\beta} + E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\,\boldsymbol{\varepsilon}] = \boldsymbol{\beta} + E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\,E[\boldsymbol{\varepsilon}] = \boldsymbol{\beta}$$

<u>Technical Note</u>: To get consistency (& asymptotic normality) for **b**, we need an additional (asymptotic) assumption regarding **X**:

$$\mathbf{X}'\mathbf{X}/T \xrightarrow{p} \mathbf{Q} \qquad (\mathbf{Q} \text{ a pd } (k\text{x}k) \text{ matrix of finite elements})$$

or $$\text{plim}\,(\mathbf{X}'\mathbf{X}/T) = \mathbf{Q}$$

Question: Why do we need this assumption in terms of a ratio divided by *T*?
Each element of **X′X** matrix is a sum of *T* numbers. As $T \rightarrow \infty$, these sums will become large. We divide by *T* so that the sums will not be too large.

<u>Note</u>: This assumption is not a difficult one to make since the LLN suggests that the each component of **X′X**/*T* goes to the mean values of **X′X**. We require that these values are finite.
– Implicitly, we assume that there is not too much dependence in **X**.

CLM: Departures from (A2) – Endogeneity

If there is dependence between **X** & ε, OLS **b** is no longer unbiased or consistent. Easy to see the biased result: we cannot longer separate $E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\,\boldsymbol{\varepsilon}]$ into a product of two expectations:

$$E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\,\boldsymbol{\varepsilon}] \neq E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\,E[\boldsymbol{\varepsilon}]$$

Dependence between **X** & ε occurs when **X** is also an endogenous variable, like **y**. This is common, especially in Corporate Finance. For example, we study CEO compensation as function of size of firm, and Board composition. Board Composition and size of firm are endogenous –i.e., determined by the firm, dependent on CEO's decisions.

Inconsistency is a fatal flaw in an estimator. In these situations, we use different estimation methods. The most popular is Instrumental Variable (IV) estimation.

CLM: Departures from (A2) – Asymptotics

Now, we have a new set of assumptions in the CLM:
(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$.
(**A2'**) **X** stochastic, but E[**X**′ □] = 0 and E[ε] = **0.**
(**A3**) $\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\,\mathbf{I}_T$
(**A4'**) plim (**X′X**/*T)* = **Q** (p.d. matrix with finite elements, rank= *k*)

With these new assumptions and using properties of plims and the CLT, we can show the following asymptotic results:
1. **b** and s^2 are consistent.

2. $\sqrt{T}\,(\mathbf{b} - \boldsymbol{\beta}) \xrightarrow{d} \text{N}(\mathbf{0}, \sigma^2\mathbf{Q}^{-1})$ $\Rightarrow \mathbf{b} \xrightarrow{a} \text{N}(\boldsymbol{\beta}, (\sigma^2/T)\mathbf{Q}^{-1})$

3. *test-t* $\xrightarrow{d}$ N(0,1)

F-tests & Wald tests $\xrightarrow{d} \chi^2_J$

CLM: Departures from (A5)

Notice that asymptotic results 2 and 3 state the asymptotic distribution of **b** and the *t*-, *F*- and Wald test. All derived from the new set of assumptions and the CLT. (**A5**) was not used.

That is, we relax (**A5**), but, now, we require *large samples* ($T \rightarrow \infty$).

Note: In practice, we use the asymptotic distribution as an approximation to the finite sample – i.e., for any *T*– distribution. This is why we used the $\xrightarrow{a}$ notation in:

$$\mathbf{b} \xrightarrow{a} \text{N}(\boldsymbol{\beta}, (\sigma^2/T)\mathbf{Q}^{-1})$$

We should be aware that this approximation may not be accurate in many situations.

• Two observations regarding relaxing (**A5**) $\boldsymbol{\varepsilon}|\mathbf{X} \sim i.i.d.\ \text{N}(\mathbf{0}, \sigma^2\mathbf{I}_T)$:

– Throwing away the normality for **ε**|**X** is not bad. In many econometric situations, normality is not a realistic assumption (daily, weekly, or monthly stock returns do not follow a normal).

– Removing the *i.i.d.* assumption for **ε**|**X** is also not bad. In many econometric situations, identical distributions are not realistic, since different means and variances are common.

Questions:
- Do we need to throw away normality for **ε**|**X**?
Not necessarily. We can test for normality on the residuals using a Jarque-Bera test.

- Why are we interested in large sample properties, like consistency, when in practice we have finite samples?
As a first approximation, the answer is that if we can show that an estimator has good large sample properties, then we may be optimistic about its finite sample properties. For example, if an estimator is inconsistent, we know that for finite samples it will definitely be biased.

CLM: Departures from (A3)

Now, we relax (**A3**). The CLM assumes that errors are uncorrelated and all are drawn from a distribution with the same variance, σ^2.
(**A3**) $\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\mathbf{I}_T$

Instead, we will assume:
(**A3’**) $\text{Var}[\boldsymbol{\varepsilon}|\mathbf{X}] = \boldsymbol{\Sigma}$ (sometimes written= $\sigma^2\boldsymbol{\Omega}$, where $\boldsymbol{\Omega} \neq \mathbf{I}_T$)

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1T} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2T} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{T1} & \sigma_{T2} & \cdots & \sigma_T^2 \end{bmatrix}$$

- Two Leading Cases:
  - Pure heteroscedasticity: We model only the diagonal elements.
  - Pure autocorrelation: We model only the off-diagonal elements.

CLM: Departures from (A3) – Heteroscedasticity

Pure heteroscedasticity: $E[\boldsymbol{\varepsilon}_i'\boldsymbol{\varepsilon}_j|\mathbf{X}] = \sigma_{ij} \quad = \sigma_i^2 \quad$ if i=j

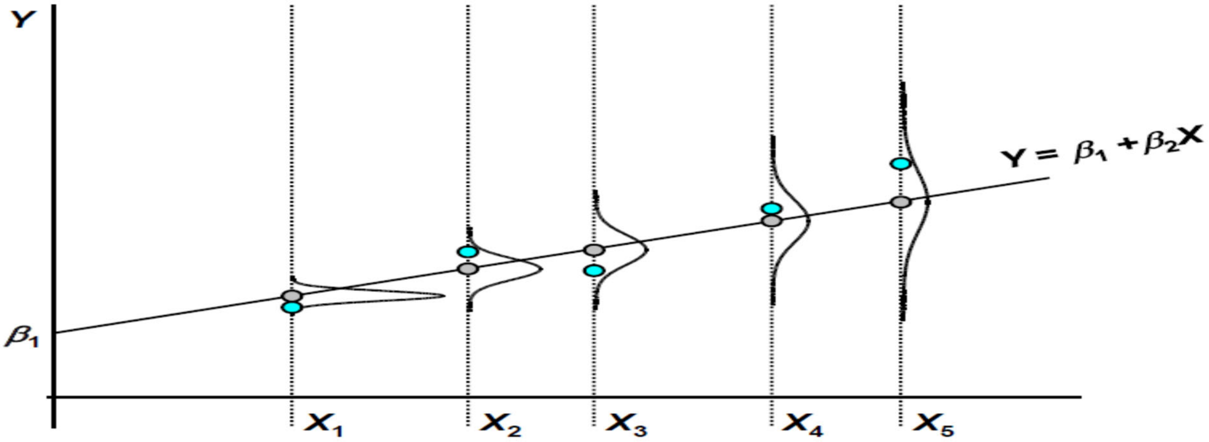
$= 0 \qquad$ if $i \neq j$

$\Rightarrow \text{Var}[\boldsymbol{\varepsilon}_i|\mathbf{X}] = \sigma_i^2$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_T^2 \end{bmatrix}$$

This type of variance-covariance structure is common in time series, where we observe the variance of the errors changing over time or subject to different regimes (say, bear and bull regimes).

Relative to pure heteroscedasticity, LS gives each observation a weight of 1/*T*. But, if the variances are not equal, then some observations (low variance ones) are more informative than others.



CLM: Departures from (A3) – Cross-correlation

Pure cross/auto-correlation: $E[\boldsymbol{\varepsilon}_i'\boldsymbol{\varepsilon}_j|\mathbf{X}] = \sigma_{ij} \qquad$ if $i \neq j$

$= \sigma^2 \qquad$ if i=j

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2 & \sigma_{12} & \cdots & \sigma_{1T} \\ \sigma_{21} & \sigma^2 & \cdots & \sigma_{2T} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{T1} & \sigma_{T2} & \cdots & \sigma^2 \end{bmatrix}$$

This type of variance-covariance structure is common in cross sections, where errors can show strong correlations, for example, when we model returns, the errors of two firms in the same industry can be subject to common (industry) shocks. Also common in time series, where we observe clustering of shocks over time.

Relative to pure cross/auto-correlation, LS is based on simple sums, so the information that one observation (today's) might provide about another (tomorrow's) is never used.

Note: Heteroscedasticity and autocorrelation are different problems and generally occur with different types of data. But, the implications for OLS are the same.

CLM: Departures from (A3) – Implications

OLS **b** is still *unbiased* and *consistent.* (Proofs do not rely on (**A3**).

OLS **b** still follows an *asymptotic normal distribution*. It is
– Easy to show this result for the pure heteroscedasticity case using a version of the CLT that assumes only independence ;
– More complicated derivation –i.e., with new assumptions- for the auto-correlation case.

Note: We used (**A3**) to derive our test statistics. A revision is needed!

Finding Heteroscedasticity

There are several theoretical reasons why the σ_i^2 may be related to some variables z_i and/or z_i^2:

1. Following the *error-learning models*, as people learn, their errors of behavior become smaller over time. Then, σ_i^2 is expected to decrease.
2. As data collecting techniques improve, σ_i^2 is likely to decrease. Companies with sophisticated data processing techniques are likely to commit *fewer errors* in forecasting customer's orders.
3. As incomes grow, people have more *discretionary income* and, thus, more choice about how to spend their income. Hence, σ_i^2 is likely to increase with income.
4. Similarly, companies with larger profits are expected to show greater variability in their dividend/buyback policies than companies with lower profits.

Heteroscedasticity can also be the result of *outliers* (either very small or very large). The inclusion/exclusion of an outlier, especially if *T* is small, can affect the results of regressions.

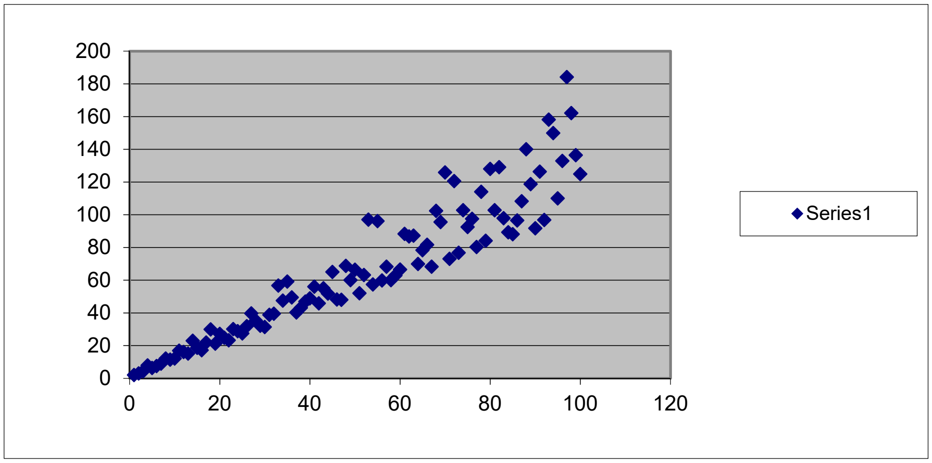
Violations of (**A1**) *–model is correctly specified–*, can produce heteroscedasticity, due to omitted variables from the model or incorrect functional form (e.g., linear vs log–linear models).

Skewness in the distribution of one or more regressors included in the model can induce heteroscedasticity. Examples are economic variables such as income, wealth, and education.

Heteroscedasticity is usually modeled using one the following specifications:
– H1 : σ_t^2 is a function of past ε_t^2 and past σ_t^2 (ARCH models).
– H2 : σ_t^2 increases monotonically with one (or several) exogenous variable(s) ($z_{1,}, \ldots, z_T$).
– H3 : σ_t^2 increases monotonically with $E(y_t)$.
– H4 : σ_t^2 is the same within p subsets of the data but differs across the subsets (*grouped heteroscedasticity*). This specification allows for structural breaks.

These are the usual alternatives hypothesis (H_1) in the heteroscedasticity tests.

• **Visual test**
In a plot of residuals against dependent variable or other variable will often produce a fan shape.



Testing for Heteroscedasticity

Question: Why do we want to test for heteroscedasticity if **b** is unbiased?
OLS is no longer efficient. There is an estimator with lower asymptotic variance (the GLS/FGLS estimator).

We want to test: H_0: $E(\varepsilon^2|x_1, x_2,\ldots, x_k) = E(\varepsilon^2) = \sigma^2$

H_1 and the structure of the test depend on what we consider the drivers of ¤σ▯*i*▯*2*▯ – i.e., in the previous examples: H1, H2, H3, H4, etc.

The key is whether $E[\varepsilon^2] = \sigma_i^2$ is related to **x** and/or x_i^2. Suppose we suspect a particular independent variable, say $\mathbf{X}_j$, is driving σ_i^2.

Then, a simple test: Check the RSS for large values of $\mathbf{X}_j$, and the RSS for small values of $\mathbf{X}_j$. This is the Goldfeld-Quandt (GQ) test.

Testing for Heteroscedasticity: GQ Test

GQ tests H_0: $\sigma_i^2 = s^2$

H_1: $\sigma_i^2 = f(\mathbf{X}_j)$

- Easy to compute:

– **Step 1**. Arrange the data from small to large values of the independent variable suspected of causing heteroscedasticity, $\mathbf{X}_j$.

– **Step 2**. Run two separate regressions, one for small values of $\mathbf{X}_j$ and one for large values of $\mathbf{X}_j$, omitting d middle observations (≈ 20%). Get the RSS for each regression: RSS_1 for small values of $\mathbf{X}_j$ and RSS_2 for large $\mathbf{X}_j$'s.

– **Step 3**. Calculate the F ratio
$GQ = RSS_2/RSS_1$, $\sim F_{df,df}$ with $df = [(T - d) - 2(k+1)]/2$ (**A5** holds).

If (**A5**) does not hold, the $F_{df,df}$ distribution becomes an approximation and other tests may be preferred.

Note: When we suspect more than one variable is driving σ^2_i, the GQ test is not very useful.

- But, the GQ test is a popular test for structural breaks (two regimes) in variance. For these tests, we rewrite **step 3** to allow for a different sample size in the sub-samples 1 and 2, since the breaking point does not have to be in the middle of the sample.

– **Step 3**. Calculate the F-test ratio
$GQ = [RSS_2/ (T_2 - k)]/[RSS_1/ (T_1 - k)]$

Note: The package *lmtest* computes this test *gqtest*. It splits the sample in the middle. You need to specify the d of middle observations not included in test. Recall, you need to install the package before using it: install.packages("lmtest").

Example: We test if the 3-factor FF model for IBM and GE returns shows heteroscedasticity with a GQ test, using *gqtest* in package *lmtest.*

- IBM returns

```
> library(lmtest)
> gqtest(ibm_x ~ Mkt_RF + SMB + HML, fraction = .20)
      Goldfeld-Quandt test

data:  ibm_x ~ Mkt_RF + SMB + HML
GQ = 1.1006, df1 = 224, df2 = 223, p-value = 0.2371            ⇒ cannot reject H0 at 5%
level.
alternative hypothesis: variance increases from segment 1 to 2
```

- GE returns

```
 gqtest(ge_x ~ Mkt_RF + SMB + HML, fraction = .20)
       Goldfeld-Quandt test

data:  ge_x ~ Mkt_RF + SMB + HML
GQ = 2.744, df1 = 281, df2 = 281, p-value < 2.2e-16            ⇒ reject H0 at 5% level.
alternative hypothesis: variance increases from segment 1 to 2
```

Testing for Heteroscedasticity: LM Tests

Popular heteroscedasticity LM tests:
- Breusch and Pagan (1979)'s LM test (BP).
- White (1980)'s general test.

Both tests are based on OLS residuals, **e**, and calculated under H_0 (No heteroscedasticity): s^2. The squared residuals are used to estimate σ_i^2.

• The BP test is an LM test, derived under normality –i.e., (**A5**). It is a general tests designed to detect any linear forms of heteroscedasticity, driven by some variables, **z**. That is, the BP tests:

H_0: $\sigma_i^2 = s^2$
H_1: $\sigma_i^2 = f(\mathbf{z}_i)$

• The White test is an asymptotic Wald-type test, where normality is not needed. It allows for nonlinearities by using squares and cross-products of all the x's in the auxiliary regression –i.e., as the drivers of σ_i^2. That is, the White tests:

H_0: $\sigma_i^2 = s^2$
H_1: $\sigma_i^2 = f(x_1^2, x_2^2, ..., x_J^2, x_1x_2, x_1x_3, x_2x_3, ...)$

Testing for Heteroscedasticity: BP Test

The derivation of the BP test is complicated, it relies on the likelihood function, which is constructed under normality, and its first derivative, the score. However, the implementation of the BP test is simple, based on the squared OLS residuals, e_i^2.

• Calculation of the Breusch-Pagan test

- **Step 1**. Run OLS on DGP:

$\mathbf{y} = \mathbf{X}\,\beta + \varepsilon$. –Keep e_i and compute $\sigma_R^2 = RSS/T$

- **Step 2**. (Auxiliary Regression). Run the regression of e_i^2/σ_R^2 on the m explanatory variables, **z**. In our example,

$$e_i^2/\sigma_R^2 = \alpha_0 + z_{i,1}'\,\alpha_1 + ... + z_{i,m}'\,\alpha_m + v_i$$

- **Step 3**. Keep the RSS from this regression. et's call it RSS_e. Compute

$$\text{LM} = RSS_e/2 \overset{d}{\rightarrow} \chi_m^2.$$

• There is version of the BP, which is robust to departures from normality. It is the "*studentized*" version of Koenker (1981). The BP test is asymptotically equivalent to a T*R^2 test, where R^2 is calculated from a regression of e_i^2/σ_R^2 on the variables **Z**. (Omitting σ_R^2 from the denominator is OK.)

• We have different Steps 2 & 3:

- **Step 2**. (Auxiliary Regression). Run the regression of e_i^2 on the m explanatory variables, **z**. In our example,

$$e_i^2 = \alpha_0 + z_{i,1}'\,\alpha_1 + ... + z_{i,m}'\,\alpha_m + v_i \quad \text{–Keep } R^2.$$

- **Step 3**. Using the R^2 from Step 2. Let's call it R_{e2}^2. Compute

$$\text{LM} = T\,R_{e2}^2 \xrightarrow{d} \chi_m^2.$$

Example: We suspect that squared Mkt_RF (x1) –a measure of the overall market's variance- drives heteroscedasticity. We do a studentized LM-BP test for **IBM** in the 3-factor FF model:

```
fit <- lm (ibm_x ~ Mkt_RF + SMB + HML)          # Step 1 – OLS in DGP (3-factor FF model)
e <- fit$residuals                               # Step 1 – keep residuals
e2 <- e^2                                # Step 1 – squared residuals
Mkt_RF_2 <- Mkt_RF^2
fit <- lm(e2 ~ Mkt_RF_2)                 # Step 2 – Auxiliary regression
Re_2 <- summary(fit_BP)$r.squared        # Step 2 – keep R^2
LM_BP_test <- Re2 * T
> LM_BP_test                             # Step 3 – Compute LM-BP test: R^2 * T
[1] 0.25038
p_val <- 1 - pchisq(LM_BP_test, df = 1)  # p-value of LM_test
>  p_val
[1] 0.6168019
```

LM-BP Test: **0.25028** ⇒ cannot reject H_0 at 5% level ($\chi^2_{[1],.05} \approx$ **3.84**); with a *p-value*= **.6168**.

• The *bptest* in the *lmtest* package performs a studentized LM-BP test for the same variables used in the model (Mkt, SMB and HML). For IBM in the 3-factor FF model:

```
> bptest(ibm_x ~ Mkt_RF + SMB + HML)   #bptest only allows to test H1:σ_i^2= f(x_i=model
variables
        studentized Breusch-Pagan test

data:  ibm_x ~ Mkt_RF + SMB + HML
BP = 4.1385, df = 3, p-value = 0.2469
```

LM-BP Test: **4.1385** ⇒ cannot reject H_0 at 5% level ($\chi^2_{[3],.05} \approx$ **7.815**); with a *p-value* = **0.2469**.

Note: Heteroscedasticity in financial time series is very common. In general, it is driven by squared market returns or squared past errors.

Example: We suspect that squared Market returns drive heteroscedasticity. We do an LM-BP (studentized) test for **Disney**:

```
lr_dis <- log(x_dis[-1]/x_dis[-T])              # Log returns for DIS
dis_x <- lr_dis – RF                            # Disney excess returns
fit <- lm (dis_x ~ Mkt_RF + SMB + HML)          # Step 1 – OLS in DGP (3-factor FF model)
e <- fit_r$residuals                            # Step 1 – keep residuals
e2 <- e^2                                       # Step 2 – squared residuals
fit <- lm (e2 ~ Mkt_RF_2)                       # Step 2 – Auxiliary regression
Re_2 <- summary(fit_BP)$r.squared               # Step 2 – Keep R^2 from Auxiliary reg
LM_BP_test <- Re_2 * T                          # Step 3 – Compute LM Test: R^2 * T
> LM_BP_test
[1] 14.15224
>  p_val <- 1 - pchisq(LM_BP_test, df = 1)      # p-value of LM_test
>  p_val
[1] 0.0001685967
```

LM-BP Test: **14.15** ⇒ reject H$_0$ at 5% level ($\chi^2_{[1],.05} \approx$ **3.84**); with a *p-value* = **.0001**.

- We do the same test but with SMB squared for Disney:

```
fit <- lm (dis_x ~ Mkt_RF + SMB + HML)
e <- fit_r$residuals
e2 <- e^2
SMB_2 <- SMB^2
fit <- lm (e2 ~ SMB_2)
Re_2 <- summary(fit_BP)$r.squared
LM_BP_test <- Re_2 * T
> LM_BP_test
[1] 7.564692
p_val <- 1 - pchisq(LM_BP_test, df = 1)  # p-value of LM_test
>  p_val
[1] 0.005952284
```

LM-BP Test: **7.56** ⇒ reject H$_0$ at 5% level ($\chi^2_{[1],.05} \approx$ **3.84**); with a *p-value*= **.006.**

- If we do use the lmtest package, we get:

```
> bptest(dis_x ~ Mkt_RF + SMB + HML)

        studentized Breusch-Pagan test

data:  dis_x ~ Mkt_RF + SMB + HML
```

BP = **6.9935**, df = 3, p-value = **0.07211**

LM-BP Test: **6.99** ⇒ cannot reject H_0 at 5% level ($\chi^2_{[3],.05} \approx$ **7.815**); with a p-value = **.07211**.

Note: In general, you need squared values when model heteroscedasticity in financial assets.

Example: We suspect that squared interest rate differentials drive heteroscedasticity for residuals in encompassing (IFE + PPP) model for changes in the **USD/GBP**. We do an LM-BP (studentized) test:

```
y <- lr_usdgbp
fit <- lm (y ~ inf_dif + int_dif)
e <- fit$residuals
e2 <- e^2
int_dif_2 <- int_dif^2
fit_BP <- lm (e2 ~ int_dif_2)
Re_2 <- summary(fit_BP)$r.squared
LM_BP_test <- Re_2 * T
> LM_BP_test
[1] 21.11134
p_val <- 1 - pchisq(LM_BP_test, df = 1)          # p-value of LM_test
>  p_val
[1] 4.333567e-06
```

LM-BP Test: **21.11134** ⇒ reject H_0 at 5% level (*p-value* < **.00001**).

Testing for Heteroscedasticity: White Test

The White test derivation is also complicated, but, the usual calculation of the White test is a known one for us:

– **Step 1**. (Same as BP's Step 1). Run OLS on DGP:

$$\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}. \quad \text{Keep residuals, } e_i.$$

– **Step 2**. (Auxiliary Regression). Regress e^2 on all the explanatory variables (X_j), their squares (X_j^2), and all their cross products.

For example, when the model contains $k = 2$ explanatory variables, the test is based on:

$$e_i^2 = \beta_0 + \beta_1\, x_{1,i} + \beta_2\, x_{2,i} + \beta_3\, x_{1,i}^2 + \beta_4\, x_{2,i}^2 + \beta_5\, x_1 x_{2,i} + v_i$$

Let m be the number of regressors in auxiliary regression (in the above example, m=5). Keep R^2, say R^2_{e2}.

– **Step 3**. Compute the statistic: $LM = T\, R^2_{e2} \xrightarrow{d} \chi^2_m$.

Example: White Test for the 3 factor F-F model for IBM returns (T=569):

$$IBM_{Ret} - r_f = \beta_0 + \beta_1\,(Mkt_{Ret} - r_f) + \beta_2\, SMB + \beta_3\, HML + \varepsilon$$

```
b <- solve(t(x)%*% x)%*% t(x)%*%y                # OLS regression (can use lm package too)
e <- y - x%*%b
e2 <- e^2
xx2 <- cbind(x1^2,x2^2,x3^2,x1*x2,x1*x3,x2*x3)  # Not including original variables is OK
fit2 <- lm(e2~xx2)
r2_e2 <- summary(fit2)$r.squared                 # Keep R^2 from Auxiliary regression
> r2_e2
[1] 0.0166492
lm_t <- T*r2_e2                                   # Compute LM test: R^2 * sample size (T)
> lm_t
[1] 10.93483
> ncol(xx2)
[1] 6
```

• Now, we do a White Test for the 3 factor F-F model for **DIS** and **GE** returns (*T*=569).

- For **DIS**, we get:

```
fit <- lm (dis_x ~ Mkt_RF + SMB + HML)
e <- fit$residuals
e2 <- e^2
Mkt_RF_2 <- Mkt_RF^2;  SMB_2 <- SMB^2;  HML_2 <- HML^2;
Mkt_HML <- Mkt_RF*HML;  Mkt_SMB <- Mkt_RF*SMB;  SMB_HML <- SMB*HML
fit_W <- lm (e2 ~ Mkt_RF_2 + SMB_2 + HML_2 + Mkt_HML + Mkt_SMB + SMB_HML)
Re_2W <- summary(fit_W)$r.squared
LM_W_test <- Re_2W * T
> LM_W_test
```

[1] **25.00148** ⇒ reject H_0 at 5% level ($\chi^2_{[6],05}$≈12.59).

```
p_val <- 1 - pchisq(LM_W_test, df = 6)            # p-value of LM_test
>  p_val
[1] 0.0003412389
```

- For **GE**, we get:

LM-White Test: **20.15** (*p-value* = **0.0026**) ⇒ reject H_0 at 5% level.

Example: We do a White Test for the residuals in the encompassing (IFE + PPP) model for changes in the **USD/GBP** (*T*=363):

```
fit_gbp <- lm(lr_usdgbp ~ inf_dif + int_dif)
e_gbp <- fit_gbp$residuals
e_gbp2 <- e_gbp^2
int_dif2 <- int_dif^2;  inf_dif2 <- inf_dif^2;  int_inf_dif <- int_dif*inf_dif
fit_W <- lm (e_gbp2 ~ int_dif2 + inf_dif2+ int_inf_dif)
Re_2W <- summary(fit_W)$r.squared
LM_W_test <- Re_2W * T
p_val <- 1 - pchisq(LM_W_test, df = 3)                    # p-value of LM_test
```

```
> LM_W_test
[1] 15.46692
> p_val
[1] 0.001458139                         ⇒ reject H0 at 5% level
```

Testing for Heteroscedasticity: LR Test

We define the likelihood function, assuming normality –i.e. (**A5**)–, for a general case, where we have g different variances:

$$\ln L = -\frac{T}{2}\ln 2\pi - \sum_{i=1}^{g}\frac{T_i}{2}\ln\sigma_i^2 - \frac{1}{2}\sum_{i=1}^{g}\frac{1}{\sigma_i^2}(y_i - X_i\beta)'(y_i - X_i\beta)$$

We have two models:
(R) Restricted under H_0: $\sigma_i^2 = \sigma^2$. From this model, we calculate ln L

$$\ln L_R = -\frac{T}{2}[\ln(2\pi)+1] - \frac{T}{2}\ln(\hat{\sigma}^2)$$

(U) Unrestricted. From this model, we calculate the log likelihood.

$$\ln L_U = -\frac{T}{2}[\ln(2\pi)+1] - \sum_{i=1}^{g}\frac{T_i}{2}\ln\hat{\sigma}_i^2; \quad \hat{\sigma}_i^2 = \tfrac{1}{T_i}(y_i - X_i b)'(y_i - X_i b)$$

- Now, we can estimate the Likelihood Ratio (LR) test:

$$LR = 2(\ln L_U - \ln L_R) = T\ln\hat{\sigma}^2 - \sum_{i=1}^{g} T_i \ln\hat{\sigma}_i^2 \xrightarrow{a} \chi^2_{g-1}$$

Under the usual regularity conditions, LR is approximated by a χ^2_{g-1}.

Testing for Heteroscedasticity: Remarks

Drawbacks of the Breusch-Pagan test:
- It is sensitive to violations of the normality assumption. The studentized version of Koenker is more robust and, then, more used.

Drawbacks of the White test
- If a model has several regressors, the test can consume a lot of df's.

- In cases where the White test statistic is statistically significant, heteroscedasticity may not necessarily be the cause, but model specification errors.

- It is general. It does not give us a clue about how to model heteroscedasticity to do FGLS. The BP test points us in a direction.

- In simulations, it does not perform well relative to others, especially, for time-varying heteroscedasticity, typical of financial time series.

Finding Auto-correlation

In general, we find autocorrelation (or serial correlation) in time series, shocks are persistent over time: It takes time to absorb a shock.
The shocks can also be correlated over the cross-section, causing cross-correlation. For example, if an unexpected new tax is imposed on the technology sector, all the companies in the sector are going to share this shock.

Usually, we model autocorrelation using two model: autoregressive (AR) and moving averages (MA).

In an AR model, the errors, ε_t, show a correlation over time. In an MA model, the errors, ε_t, are a function (similar to a weighted average) of previous errors, now denoted u_t's.

Examples:

- First-order autoregressive autocorrelation: AR(1)

$$\varepsilon_t = r_1\varepsilon_{t-1} + u_t$$

- Fifth-order autoregressive autocorrelation: AR(p)

$$\varepsilon_t = r_1\varepsilon_{t-1} + r_2\varepsilon_{t-2} + \cdots + r_p\varepsilon_{t-p} + u_t$$

- Third-order moving average autocorrelation: MA(3)

$$\varepsilon_t = u_t + \lambda_1 u_{t-1} + \lambda_2 u_{t-2} + \lambda 3 u_{t-3}$$

Note: The last example is described as third-order moving average autocorrelation, denoted MA(3), because it depends on the three previous innovations as well as the current one.

Finding Auto-correlation – Visual Check

Plot data, usually residuals from a regression, to see if there is a pattern:
- Positive autocorrelation: A positive (negative) observation tends to be followed by a positive (negative) observation. We tend to see continuation in the series.

- Negative autocorrelation: A positive (negative) observation tends to be followed by a negative (positive) observation. We tend to see reversals.

- No autocorrelation: A positive (negative) observation has the same probability of being followed by a negative or positive (positive or negative) observation. We tend to no pattern.

Example: I simulate a y_t series, with N(0,1) u_t errors:

$$y_t = \rho_1 y_{t-1} + u_t$$

Three cases:
(1) Positive autocorrelation: $\rho_1 = .70$

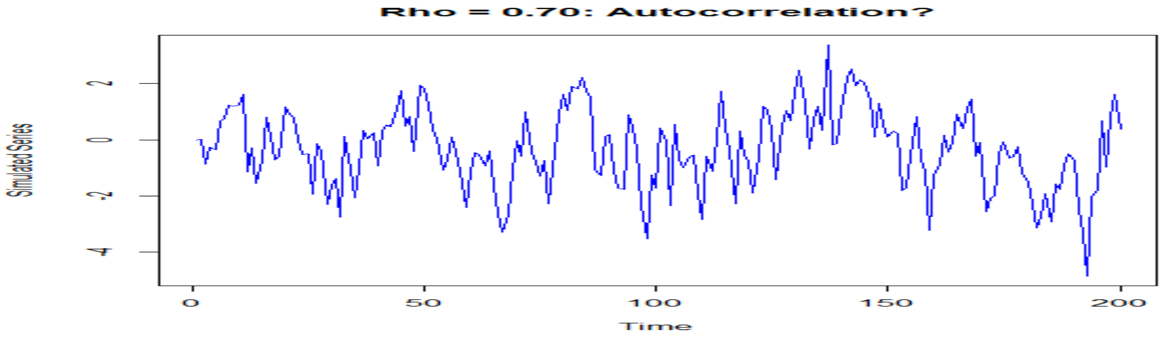
(2) Negative autocorrelation: $\rho_1 = -.70$

(3) No correlation: $\rho_1 = -0$

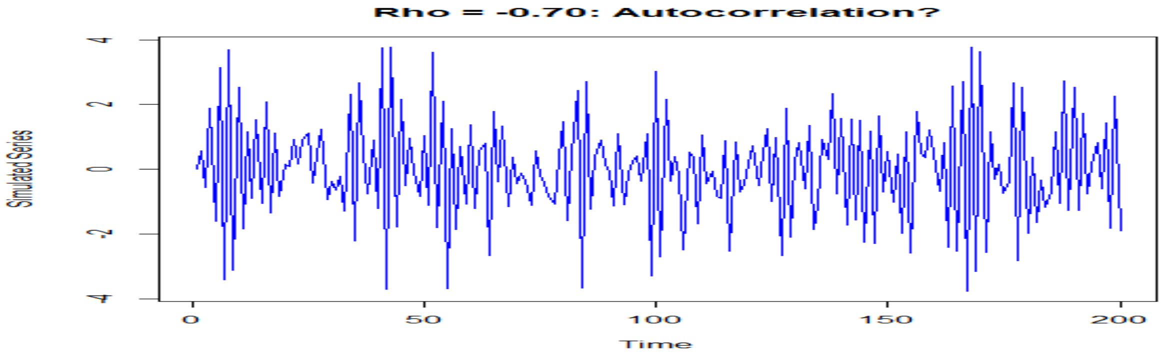
• R code for simulation:

```
T_sim <- 200
u <- rnorm(200)                                                    # Draw T_sim normally
distributed errors
y_sim <- matrix(0,T_sim,1)
rho <- .7                                                  # Change to create different
correlation patterns
a <- 2                                                     # Time index for observations
while (a <= T_sim) {
                    y_sim[a] = rho * y_sim[a-1] + u[a]      # y_sim simulated autocorrelated
values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Visual Test: Autocorrelation?")
```

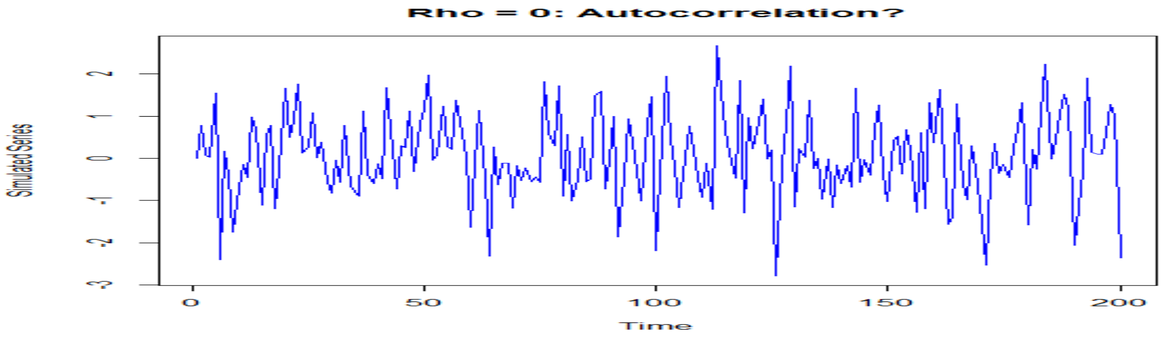
(1) Positive autocorrelation $r_1 = .70$



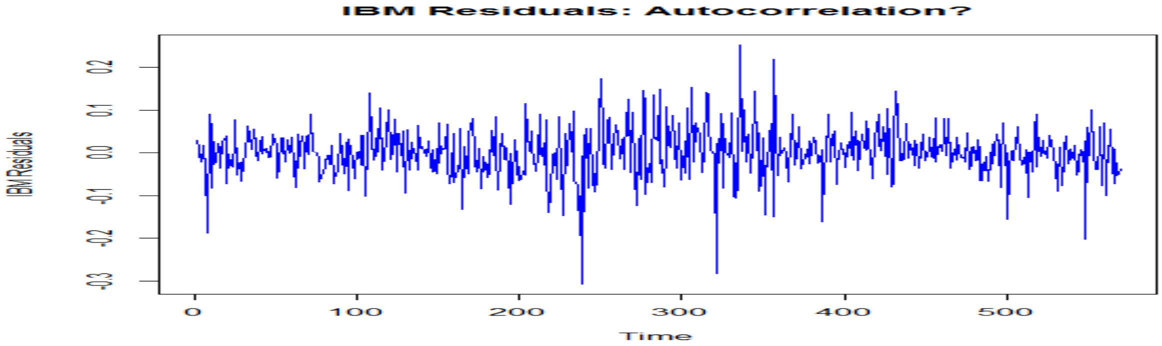
(2) Negative autocorrelation $r_1 = -.70$

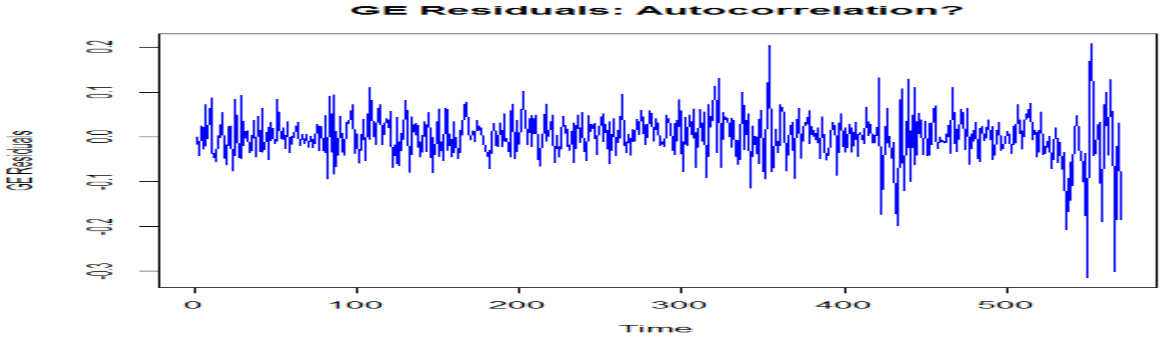


(3) No autocorrelation: $r_1 = 0$



Example: Residual plot for the 3 factor F-F model for **IBM returns** and **GE returns**:





Conclusion: It looks like a small ρ_1, but not very clear pattern from the graphs.

Testing for Autocorrelation: LM Test

There are several autocorrelation tests. Under the null hypothesis of no autocorrelation of order p, we have H_0: $\rho_1 = ... = \rho_p = 0$.

Under H_0, we can use OLS residuals.

- Breusch–Godfrey (1978) LM test. Similar to the BP test:

- **Step 1**. (Auxiliary Regression). Run the regression of e_i on all the explanatory variables, **X**. In our example,

$$e_t = \mathbf{X}_t'\, \boldsymbol{\beta} + \alpha_1\, e_{t-1} + ... + \alpha_p\, e_{t-p} + v_t$$

- **Step 2**. Keep the R^2 from this regression. Let's call it R_e^2. Then, calculate:

$$\text{LM} = (T\text{-}\ p)\ R_e^2 \overset{d}{\to}\ \chi_p^2.$$

Example: LM-AR Test for the 3 factor F-F model for IBM returns (p=**12 lags**):

```
fit_ibm<- lm(ibm_x ~ Mkt_RF + SMB + HML)      # OLS regression
e <- fit_ibm$residuals                        # OLS residuals
p_lag <- 12                              # Select # of lags for test (set p)
e_lag <- matrix(0,T-p_lag,p_lag)              # Matrix to collect lagged residuals
a <- 1
while (a<=p_lag) {                       # Do loop creates matrix (e_lag) with lagged e
  za <- e[a:(T-p_lag+a-1)]
   e_lag[,a] <- za
a <- a+1
}

Mkt_RF_p <- Mkt_RF[(p_lag+1):T]               # Adjust for new sample size: T – p_lag
SMB_p <- SMB[(p_lag+1):T]
HML_p <- HML[(p_lag+1):T]
fit1 <- lm(e[(p_lag+1):T] ~ e_lag + Mkt_RF_p + SMB_p + HML_p)        # Auxiliary Regression
r2_e1 <- summary(fit1)$r.squared              # get R^2 from Auxiliary Regression
lm_t <- (T-p_lag )* r2_e1                     # LM-test wih p lags
lm_t                                          # print lm_t
df <- ncol(e_lag)                             # degrees of freedom of test
1 - pchisq(lm_t,df)                           # p-value of lm_t
r2_e1 <- summary(fit1)$r.squared
> r2_e1
[1] 0.0303721
> (T-p_lag)
[1] 557
lm_t <- (T - p_lag) * r2_e1
> lm_t
[1] 16.91726
df <- ncol(e_lag)                             # degrees of freedom for the LM Test
> 1-pchisq(lm_t,df)
[1] 0.1560063
```

LM-AR(**12**) Test: **16.91726** ⇒ cannot reject H_0 at 5% level (*p-value* > .05).

• If I run the test with p=**4 lags**, I get

LM-AR(**4**) Test: **2.9747** (*p-value* = 0.56) ⇒ cannot reject H_0 at 5% level (*p-value* > .05).

• The package *lmtest,* performs this test, *bgtest,* (and many others, used in this class, encompassing, jtest, waldtest, etc). You need to install it first: install.packages(“lmtest”), then call the library(lmtest).

```
library(lmtest)
> bgtest(ibm_x ~ Mkt_RF + SMB + HML, order=12)

        Breusch-Godfrey test for serial correlation of order up to
        12

data:  lr_ibm ~ Mkt_RF + SMB + HML
LM test = 16.259, df = 12, p-value = 0.1797  (minor difference with the previous test, likely due
                                              to multiplication by T. Results do not change much)
```

Note: If you do not include in the Auxiliary Regression the original regressors (Mkt_RF, SMB, HML) the test do not change much. You get
LM-AR(**12**) Test: **16.83253** ⇒ very similar. Not entirely correct, but it works well.

• Autocorrelation is very common. If I run the test for **Disney**, **CNP**, or **GE**, instead, we get significant test results.

- For **DIS**:

```
lr_dis <- log(x_dis[-1]/x_dis[-T])
dis_x <- lr_dis – RF

> bgtest(dis_x ~ Mkt_RF + SMB + HML, order=4)
        Breusch-Godfrey test for serial correlation of order up to  4

data:  dis_x ~ Mkt_RF + SMB + HML
LM test = 8.6382, df = 4, p-value = 0.07081  ⇒ cannot reject H0 at 5% level (p-value >.05)

> bgtest(dis_x ~ Mkt_RF + SMB + HML, order=12)
        Breusch-Godfrey test for serial correlation of order up to 12

data:  dis_x ~ Mkt_RF + SMB + HML
LM test = 30.068, df = 12, p-value = 0.002728        ⇒ reject H0 at 5% level (p-value < .05)
```

- For **GE** (with **12 lags**):

```
lr_ge <- log(x_ge[-1]/x_ge[-T]); ge_x <- lr_ge – RF
lr_cnp <- log(x_cnp[-1]/x_cnp[-T]); cnp_x <- lr_cnp – RF

> bgtest(ge_x ~ Mkt_RF + SMB + HML, order=4)
        Breusch-Godfrey test for serial correlation of order up to 4

data:  ge_x ~ Mkt_RF + SMB + HML
```

```
LM test = 28.257, df = 4, p-value = 0.005073         ⇒ cannot reject H0 at 5% level (p-value
>.05)
```

- For **CNP** (with **12 lags**):

```
> bgtest(cnp_x ~ Mkt_RF + SMB + HML, order=12)
      Breusch-Godfrey test for serial correlation of order up to 12

data:  cnp_x ~ Mkt_RF + SMB + HML
LM test = 31.718, df = 12, p-value = 0.00153         ⇒ reject H0 at 5% level (p-value < .05)
```

• Question: How many lags are needed in the test?
Enough to make sure there is no auto-correlation left in the residuals. There are some popular rule of thumbs: for daily data, 5 or 20 lags; for weekly, 4 or 12 lags; for monthly data, 12 lags; for quarterly data, 4 lags.

Testing for Autocorrelation: Durbin-Watson

The Durbin-Watson (1950) (DW) test for AR(1) autocorrelation: H_0: $\rho_1 = 0$ against H_1: $\rho_1 \neq 0$. Based on simple correlations of **e**.

$$d = \frac{\sum_{t=2}^{T} (e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2}$$

It is easy to show that when $T \to \infty$, $d \approx 2(1 - \rho_1)$.

ρ_1 is estimated by the sample correlation r.

Under H_0, $\rho_1=0$. Then, d should be distributed randomly around 2.

Small values (close to 0) or Big values (close to 4) of d lead to rejection of H_0. The distribution depends on **X**. Durbin-Watson derived bounds for the test. Since there are better tests, in practice, the DW is used "visually," that is, without checking the bounds.

Example: DW Test for the 3 factor F-F model for **IBM returns**

```
fit_dw <- lm(ibm_x ~ Mkt_RF + SMB + HML)      # OLS regression
e <- fit_dw$residuals                         # OLS residuals
RSS <- t(e)%*%e                               # RSS
DW <- sum((e[1:(T-1)]-e[2:T])^2)/RSS          # DW stat
> DW
[1] 2.042728                  ⇒ DW statistic ≈2  ⇒ No evidence for autocorrelation of order 1.
> 2*(1-cor(e[1:(T-1)],e[2:T]))                           # approximate DW stat
```

```
[1] 2.048281
```

• Similar finding for Disney returns:

```
> DW
        [,1]
[1,] 2.1609                    ⇒ DW statistic ≈2  ⇒  But, DIS suffers from autocorrelation!
```

⇒ This is why DW are not that informative. They only test for AR(1) in residuals.

• The package *lmtest* performs this test too, *dwtest*:

```
> dwtest(y ~ Mkt_RF + SMB + HML)
DW = 2.0427, p-value = 0.7087
```

Example: DW Test for the residuals of the encompassing model (IFE + PPP) for changes in USD/GBP:

```
fit_gbp <- lm(lr_usdgbp ~ inf_dif + int_dif)
e_gbp <- fit_gbp$residuals
> dwtest(fit_gbp)

        Durbin-Watson test

data:  fit_gbp
DW = 1.8588, p-value = 0.08037                ⇒not significant at 5% level.
alternative hypothesis: true autocorrelation is greater than 0
```

Testing for Autocorrelation: Portmanteau tests

We present two the Box-Pierce (1970) test and its modification, the Ljung-Box (1978) test.

- Box-Pierce (1970) test (Q test).

It tests H_0: $\rho_1 = ... = \rho_p = 0$ using the sample correlation r_j:

$$r_j: = \frac{\sum_{j=1}^{T-j} e_t\, e_{t-j}}{\sum_{j=1}^{T} e_j^2}$$

Then, under H_0: $$Q = T \sum_{j=1}^{p} r_j^2 \xrightarrow{d} \chi_p^2.$$

- Ljung-Box (1978) test (LB test).

A variation of the Box-Pierce test. It has a small sample correction.

$$LB = T * (T-2) * \sum_{j=1}^{p} \frac{r_j^2}{T-j} \xrightarrow{d} \chi_p^2.$$

Note: The LB statistic is widely used. But, the Breusch–Godfrey (1978) LM tests conditions on **X**. Thus, it is more powerful.

Example: Q and LB tests with **p** = **12 lags** for the residuals in the 3-factor FF model for **IBM returns**:

```
RSS <- sum(e^2)
r_sum <- 0
lb_sum <- 0
p_lag <- 12
a <- 1
while (a <= p_lag) {
            za <- as.numeric(t(e[(p_lag+1):T]) %*% e[a:(T-p_lag+a-1)])
  r_sum <- r_sum + (za/RSS)^2                              #sum cor(e[(p_lag+1):T], e[a:(T-
p_lag+a-1)])^2
  lb_sum <- lb_sum + (za/RSS)^2/(T-a)                      # sum with LB correction
a <- a + 1
}
Q <- T*r_sum
LB <- T*(T-2)*lb_sum
> Q
[1] 16.39559   (p-value = 0.1737815)              ⇒ cannot reject H0 at 5% level.
> LB
[1] 16.46854   (p-value = 0.1707059)              ⇒ cannot reject H0 at 5% level.
```

• The *Box.test* function computes Q & LB:

- Q test

```
> Box.test(e, lag = 12, type="Box-Pierce")

        Box-Pierce test

data:  e
X-squared = 16.304, df = 12, p-value = 0.1777
```

- LB test

```
> Box.test(e, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e
X-squared = 16.61, df = 12, p-value = 0.1649
```

Note: There is a minor difference between the previous code and the code in Box.test. They are based on how the correlations of e are computed (centered around the mean, or assumed zero mean).

• Same tests (p = **12 lags**) & same model: for Disney & GE.

- For **DIS** (dis_x), we get:

```
> Q
[1] 28.76842   (p-value = 0.004264043)       ⇒  reject H0 at 5% level.
```

```
> LB
[1] 29.05072   (p-value = 0.003872236)        ⇒  reject H0 at 5% level.
```

- For **GE** (ge_x), we get

```
> Q
[1] 24.20958   (p-value = 0.01904602)                ⇒  reject H0 at 5% level.
> LB
[1] 24.33922   (p-value = 0.01828389)                ⇒  reject H0 at 5% level.
```

• Autocorrelation in financial asset returns is a usual finding in monthly, weekly and daily data.

Example: Same Q and LB tests (p=12 lags) for the **USD/GBP** residuals in the encompassing (PPP + IFE) model:

```
> fit_gbp <- lm(lr_usdgbp ~ inf_dif + int_dif)
> e_gbp <- fit_gbp$residuals
>  Box.test(e_gbp, lag = 12, type="Box-Pierce")

        Box-Pierce test

data:  e_gbp
X-squared = 19.587, df = 12, p-value = 0.0753          ⇒ cannot reject H0 at 5% level, but close.

>  Box.test(e_gbp, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e_gbp
X-squared = 20.032, df = 12, p-value = 0.06649         ⇒ cannot reject H0 at 5% level, but close.
```

• Time-varying volatility is very common in financial time series. We can use the tests for autocorrelation to check for autocorrelation in squared returns, ¤e⍰𝑖⍰2⍰, which based on White's idea, we use to estimate ¤σ⍰𝑖⍰2⍰.

Testing for Autocorrelation: Heteroscedasticity

We can use a Portmanteu test on the squared residuals to check for this particular kind of heteroscedasticity.

H_0: $\sigma_i^2 = s^2$
H_0: $\sigma_i^2 = f(e_{i-1}^2\ , e_{i-2}^2, ..., e_{i-p}^2)$

• Of course, an LM-BP test can also be used, using lagged squared residuals as the drivers of heteroscedasticity (more on this topic in Lecture 10).

Example: Q and LB tests with p=12 lags for the squared residuals in the 3-factor FF model for IBM returns:

```
> e_ibm <- fit_ibm$residuals
> e_ibm2 <- e_ibm^2
```

- Q test

```
> Box.test(e_ibm2, lag = 12, type="Box-Pierce")

        Box-Pierce test

data:  e_ibm2
X-squared = 37.741, df = 12, p-value = 0.0001693
```

- LB test

```
> Box.test(e_ibm2, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e_ibm2
X-squared = 38.435, df = 12, p-value = 0.0001304
```

- Q and LB tests with p=12 lags for the squared residuals in the 3-factor FF model for **DIS** & **GE** returns:

- For **DIS** (dis_x), we get

```
> Box.test(e_dis2, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e_dis2
X-squared = 73.798, df = 12, p-value = 6.195e-11
```

- For **GE** (ge_x), we get

```
> Box.test(e_ge2, lag = 12, type="Ljung-Box")

        Box-Ljung test

data:  e_ge2
X-squared = 115.9, df = 12, p-value < 2.2e-16
```

- Strong evidence for time-varying heteroscedasticity in the residuals.

Generalized Regression Model (GRM)

Now, we go back to the CLM Assumptions:
(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is correctly specified.
(**A2**) or (**A2’**)
(**A3’**) $\text{Var}[\varepsilon|\mathbf{X}] = \boldsymbol{\Sigma}$ (sometimes written $\text{Var}[\varepsilon|\mathbf{X}] = \sigma^2\boldsymbol{\Omega}$)
(**A4**) or (**A4’**)

This is the generalized regression model (GRM), which allows the variances to differ across observations and allows correlation across observations.

OLS is still unbiased. Can we still use OLS?

GR Model: True Variance for b

- We will not be estimating **Σ**. Impossible with *T* data points.

- We will estimate $\mathbf{X}'\mathbf{\Sigma}\mathbf{X} = \Sigma_i\Sigma_j\, \sigma_{ij}\, \mathbf{x}_i\, \mathbf{x}_j'$, a (*k*x*k*) matrix. That is, we are estimating [*k**(*k*+1)]/2 elements.

- This distinction is very important in modern applied econometrics:
  – The White estimator
  – The Newey-West estimator

- Both estimators produce a *consistent* estimator of $\text{Var}_T[\mathbf{b}|\mathbf{X}]$.

$$\text{Var}_T[\mathbf{b}|\mathbf{X}] = (\mathbf{X'X})^{-1}\, \mathbf{X}'\mathbf{\Sigma}\mathbf{X}\, (\mathbf{X'X})^{-1}$$

Since **b** consistently estimates **β**, the OLS residuals, **e**, are also consistent estimators of **ε**. We use **e** to consistently estimate $\mathbf{X}'\mathbf{\Sigma}\mathbf{X}$.

GR Model: Robust Covariance Matrix

We will not be estimating **Σ**. Impossible with *T* data points.

We will estimate $\mathbf{X}'\mathbf{\Sigma}\mathbf{X} = \Sigma_i\Sigma_j\, \sigma_{ij}\, \mathbf{x}_i\, \mathbf{x}_j'$, a (*k*x*k*) matrix. That is, we are estimating [*k**(*k*+1)]/2 elements.

This distinction is very important in modern applied econometrics:
– The White estimator
– The Newey-West estimator

Both estimators produce a *consistent* estimator of $\text{Var}_T[\mathbf{b}|\mathbf{X}]$.

$$\text{Var}_T[\mathbf{b}|\mathbf{X}] = (\mathbf{X'X})^{-1}\, \mathbf{X}'\mathbf{\Sigma}\mathbf{X}\, (\mathbf{X'X})^{-1}$$

Since **b** consistently estimates **β**, the OLS residuals, **e**, are also consistent estimators of **ε**. We use **e** to consistently estimate $\mathbf{X}'\mathbf{\Sigma}\mathbf{X}$.

Covariance Matrix: The White Estimator

The White estimator simplifies the estimation since it only assumes heteroscedasticity. Then, **Σ** is a diagonal matrix, with elements σ_i^2.

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_T^2 \end{bmatrix}$$

Thus, we need to estimate:

$$\mathbf{Q}^* = (1/T)\ \mathbf{X}'\boldsymbol{\Sigma}\mathbf{X}$$

where

$$\boldsymbol{X'}\,\boldsymbol{\Sigma}\,\boldsymbol{X} = \begin{bmatrix} \sum_{i=1}^{T} x_{1i}^2\,\sigma_i^2 & \cdots & \sum_{i=1}^{T} x_{1i}x_{ki}\sigma_i^2 \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{T} x_{ki}x_{1i}\,\sigma_i^2 & \cdots & \sum_{i=1}^{T} x_{ki}^2\sigma_i^2 \end{bmatrix} = \Sigma_i\ \sigma_i^2\ \mathbf{x}_i\ \mathbf{x}_i'$$

Question: How do we estimate σ_i^2?

We need to estimate: $\mathbf{Q}^* = (1/T)\ \mathbf{X}'\boldsymbol{\Sigma}\mathbf{X} = (1/T)\ \Sigma_i\ \sigma_i^2\ \mathbf{x}_i\ \mathbf{x}_i'$

The OLS residuals, **e**, are consistent estimators of $\boldsymbol{\varepsilon}$. This suggests using e_i^2 to estimate σ_i^2. That is, we estimate

$$(1/T)\ \mathbf{X}'\boldsymbol{\Sigma}\mathbf{X} \text{ with } \mathbf{S_0} = (1/T)\ \Sigma_i\ e_i^2\ \mathbf{x}_i\ \mathbf{x}_i'.$$

White (1980) shows that a consistent estimator of Var[**b**|**X**] is obtained if e_i^2 is used as an estimator of σ_i^2. Taking the square root, we get a *heteroscedasticity-consistent* (HC) standard error.

Note: The estimator is also called the *sandwich estimator* or the *White estimator* (also known as *Eiker-Huber-White estimator).*

(**A3'**) was not specified. That is, the White estimator is *robust* to a potential misspecifications of heteroscedasticity in (**A3'**).

The White estimator allows us to make inferences using the OLS estimator **b** in situations where heteroscedasticity is suspected, but we do not know enough to identify its nature.

• Remarks:

(1) Since there are many refinements of the White estimator, the White estimator is usually referred as HC0 (or just "HC"):

$$\text{HC0} = (\mathbf{X'X})^{-1}\ \mathbf{X'}\ \text{Diag}[e_i^2]\ \mathbf{X}\ (\mathbf{X'X})^{-1}$$

(2) In large samples, SEs, *t*-tests and *F*-tests are asymptotically valid.

(3) The OLS estimator remains inefficient. But inferences are asymptotically correct.

(4) The HC standard errors can be larger or smaller than the OLS ones. It can make a difference to the tests.

(5) It is used, along the Newey-West estimator, in almost all finance papers. Included in all the packaged software programs

(6) In R, you can use the library "*sandwich,"* to calculate White SEs. They are easy to program:

```
# White SE in R
White_f <- function(y,X,b) {
T <- length(y); k <- length(b);
yhat <- X%*%b                                   # fitted values
e <- y-yhat                                     # residuals
hhat <- t(X)*as.vector(t(e))                    # xi ei
G <- matrix(0,k,k)
 za <- hhat[,1:k]%*%t(hhat[,1:k])                         # X’ diag[e_i] X
 G <- G + za                                              # X’ diag[e_i] X
 F <- t(X)%*%X
 V <- solve(F)%*%G%*%solve(F)                             # S_0
 white_se <- sqrt(diag(V))
 ols_se <- sqrt(diag(solve(F)*drop((t(e)%*%e))/(T-k)))
l_se = list(white_se,olse_se)
return(l_se) }
```

Example 1: We estimate t-values using OLS and White SE, for the 3 factor F-F model for **IBM returns**:

$$IBM_{Ret} - r_f = \beta_0 + \beta_1 (Mkt_{Ret} - r_f) + \beta_2\, SMB + \beta_4\, HML + \varepsilon$$

```
fit_ibm <- lm(ibm_x ~ Mkt_RF + SMB + HML)                 # OLS Regression with lm
b_i <-fit_ibm$coefficients                                # Extract OLS coefficients
SE_OLS <- sqrt(diag(vcov(fit_ibm)))                       # Extract OLS SE from fit_ibm
t_OLS <- b_i/SE_OLS                                       # Calculate  OLS t-values

> b_i
 (Intercept)      Mkt_RF         SMB         HML
-0.005191356  0.910379487 -0.221385575 -0.139179020
> SE_OLS
(Intercept)      Mkt_RF         SMB         HML
0.002482305 0.056784474 0.084213761 0.084060299
> t_OLS
(Intercept)      Mkt_RF         SMB         HML
  -2.091345   16.032190   -2.628853   -1.655705

> library(sandwich)
White <- vcovHC(fit_ibm, type = "HC0")
SE_White <- sqrt(diag(White))                             # White SE HC0
t_White <- b_i/SE_White

> SE_White
(Intercept)      Mkt_RF         SMB         HML
0.002505978 0.062481080 0.105645459 0.096087035
> t_White
(Intercept)      Mkt_RF         SMB         HML
  -2.071589   14.570482   -2.095552   -1.448468

White <- vcovHC(fit_ibm, type = "HC3")                    # White SE HC3 (refinement)
SE_White <- sqrt(diag(White))# White SE HC0
t_White <- b_i/SE_White
> SE_White
(Intercept)      Mkt_RF         SMB         HML
```

```
0.002533461 0.063818378 0.108316056 0.098800721
> t_White
(Intercept)    Mkt_RF        SMB        HML
  -2.049116   14.265162   -2.043885   -1.408684
```

Example 2: We estimate Mexican interest rates (**i**$_{MX}$) with a linear model including US interest rates, changes in exchange rates (MXN/USD), Mexican inflation and Mexican GDP growth, using quarterly data 1978:II – 2020:II (T=166):

$$i_{MX,t} = \beta_0 + \beta_1\, i_{US,t} + \beta_2\, e_t + \beta_3\, mx\_I_t + \beta_4\, mx\_y_t + \varepsilon_t$$

```
FMX_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/FX_USA_MX.csv", head=TRUE,
sep=",")

us_i <- FMX_da$US_int                                  # US short-term interest rates (iUS)
mx_CPI <- FMX_da$MX_CPI                                # Mexican CPI
mx_M1 <- FMX_da$MX_M1                                  # Mexican Money Supply (M1)
mx_i <- FMX_da$MX_int                                  # Mexican short-term int rates (iMX)
mx_GDP <- FMX_da$MX_GDP                                # Mexican GDP
S_mx <- FMX_da$MXN_USD                                 # St = exchange rates (MXN/USD)
T <- length(mx_CPI)
mx_I <- log(mx_CPI[-1]/mx_CPI[-T])                     # Mexican Inflation: Log  changes in
CPI
mx_y <- log(mx_GDP[-1]/mx_GDP[-T])               # Mexican growth: Log changes in GDP
mx_mg <- log(mx_M1[-1]/mx_M1[-T])                      # Money growth: Log  changes in
M1
e_mx <- log(S_mx[-1]/S_mx[-T])                         # Log changes in St.
us_i_1 <- us_i[-1]/100          # Adjust sample size.
mx_i_1 <- mx_i[-1]/100
mx_i_0 <- mx_i[-T]/100
fit_i <- lm(mx_i_1 ~ us_i_1 + e_mx + mx_I + mx_y)
> summary(fit_i)

Coefficients:
             Estimate  Std. Error t value Pr(>|t|)
(Intercept)  0.04022   0.01506   2.671  0.00834 **
us_i_1       0.85886   0.31211   2.752  0.00661 **
e_mx        -0.01064   0.02130  -0.499  0.61812
mx_I         3.34581   0.19439  17.212  < 2e-16 ***
mx_y        -0.49851   0.73717  -0.676  0.49985
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

White <- vcovHC(fit_i, type = "HC0")
SE_White <- sqrt(diag(White))                    # White SE HC0
t_White <- b_i/SE_White
```

```
> SE_White
(Intercept)     us_i_1       e_mx       mx_I       mx_y
0.009665759 0.480130221 0.026362820 0.523925226 1.217901733
> t_White
(Intercept)     us_i_1       e_mx       mx_I       mx_y
  4.1613603   1.7888018  -0.4035554   6.3860367  -0.4093221
```

⟹ $i_{US,t}$ not longer significant at 5% level.

```
White3 <- vcovHC(fit_i, type = "HC3")                    # Using popular refinement HC3
SE_White3 <- sqrt(diag(White3))                          # White SE HC3
t_White <- b_i/SE_White3
> t_White3
(Intercept)     us_i_1       e_mx       mx_I       mx_y
  3.6338983   1.5589936  -0.2117600   5.4554986  -0.3519886
```

⟹ $i_{US,t}$ not longer significant at 10% level

Newey-West Estimator

Now, we also have autocorrelation. We need to estimate

$$\mathbf{Q}^* = (1/T)\ \mathbf{X'\Sigma X}\ = (1/T) \sum_{i=1}^{T}\sum_{j=1}^{T} \sigma_{ij}\ \mathbf{x}_i\ \mathbf{x}_j'$$

Newey and West (1987) follow White (1980) to produce a HAC (*Heteroscedasticity and Autocorrelation Consistent*) estimator of $\mathbf{Q}^*$, also referred as *long-run variance* (LRV): Use $e_i e_j$ to estimate σ_{ij}

$$\Rightarrow \text{natural estimator of } \mathbf{Q}^*\text{: } (1/T)\ \Sigma_i\Sigma_j\ \mathbf{x}_i e_i\ e_j \mathbf{x}_j'$$

Or using time series notation, estimator of $\mathbf{Q}^*$: $(1/T)\ \Sigma_t\Sigma_s\ \mathbf{x}_t e_t\ e_s \mathbf{x}_s'$

There are some restrictions that need to be imposed: $\mathbf{Q}^*$ needs to be a pd matrix (use a quadratic form) and the double sum cannot explode (use decaying waits to cut the sum short).

- Two components for the NW HAC estimator:

(1) Start with Heteroscedasticity Component:

$$\mathbf{S_0} = (1/T) \sum_{i=1}^{T} e_i^2\ \mathbf{x}_i\ \mathbf{x}_i' \qquad \text{– the White estimator.}$$

(2) Add the Autocorrelation Component

$$\mathbf{S}_T = \mathbf{S}_0 + (1/T) \sum_{l=1}^{L} k(l) \sum_{t=l+1}^{T} (\mathbf{x}_{t-l} e_{t-l}\ e_t \mathbf{x}_t' + \mathbf{x}_t e_t\ e_{t-l} \mathbf{x}_{t-l}')$$

where

$$k(\tfrac{j}{L(T)}) = \frac{L+1-|j|}{L+1} \qquad \text{–decaying weights (\textit{Bartlett kernel})}$$

L is the cut-off lag, which is a function of *T*. (More data, longer *L*).

The weights are linearly decaying, suppose *L*=30. Then,
$k(1) = 30/31 = 0.9677419$

$k(2) = 29/31 = 0.9354839$
$k(3) = 28/31 = 0.9032258$

$$\mathbf{S}_T = \mathbf{S}_0 + (1/T) \sum_{l=1}^{l} k(l) \sum_{t=l+1}^{T} (\mathbf{x}_{t-l} e_{t-l}\, e_t \mathbf{x}_t' + \mathbf{x}_t e_t\, e_{t-l} \mathbf{x}_{t-l}')$$

Then,

$$\text{Est. Var}[\mathbf{b}] = (1/T)\, (\mathbf{X'X}/T)^{-1}\, \mathbf{S}_T\, (\mathbf{X'X}/T)^{-1} \quad \text{–NW's HAC Var.}$$

Under suitable conditions, as L, $T \rightarrow \infty$, and $L/T \rightarrow 0$, $\mathbf{S}_T \rightarrow \mathbf{Q}^*$. Asymptotic inferences can be based on OLS **b**, with *t-tests* and *Wald tests* using N(0,1) and χ^2 critical values, respectively.

There are many refinements of the NW estimators. Today, all HAC estimators are usually referred as NW estimators, regardless of the weights (*kernel*) used if they produce a positive (semi-) definite covariance matrix.

• All econometric packages (SAS, SPSS, Eviews, etc.) calculate NW SE. In R, you can use the library "*sandwich*," to calculate NW SEs:

```
library(sandwich)
> NeweyWest(x, lag = NULL, order.by = NULL, prewhite = TRUE, adjust = FALSE,
diagnostics = FALSE, sandwich = TRUE, ar.method = "ols", data = list(), verbose = FALSE)
```

You need to install the package *sandwich* and then call the library(sandwich).

Example:

```
## fit the 3 factor Fama French Model for IBM returns:
fit_ibm <- lm(ibm_x ~ Mkt_RF + SMB + HML)
## NeweyWest computes the NW SEs. It requires lags=L & suppression of prewhitening
NeweyWest(fit_ibm, lag = 4, prewhite = FALSE)
```

Note: It is usually found that the NW SEs are downward biased.

• You can also program the NW SEs yourself. In R:

```
NW_f <- function(y, X, b, lag)
{
T <- length(y);
k <- length(b);
yhat <- X%*%b
e <- y - yhat
hhat <- t(X)*as.vector(t(e))
G <- matrix(0,k,k)
a <- 0
w <- numeric(T)
while (a <= lag) {
  Ta <- T - a
  ga <- matrix(0,k,k)
  w[lag+1+a] <- (lag+1-a)/(lag+1)
  za <- hhat[,(a+1):T] %*% t(hhat[,1:Ta])
  ga <- ga + za
  G <- G + w[lag+1+a]*ga
a <- a+1
}
F <- t(X)%*%X
V <- solve(F)%*%G%*%solve(F)
 nw_se <- sqrt(diag(V))
 ols_se <- sqrt(diag(solve(F)*drop((t(e)%*%e))/(T-k)))
l_se = list(nw_se,ols_se)
return(l_se)
}
NW_f(y,X,b,lag=4)
```

Example 1: We estimate the 3 factor F-F model for **IBM returns** with NW SE with **4 lags**:

```
> t_OLS
(Intercept)     Mkt_RF       SMB       HML
  -2.091345   16.032190   -2.628853   -1.655705       ⇒ with SE_OLS: SMB significant at 1% level
NW <- NeweyWest(fit_ibm, lag = 4, prewhite = FALSE)
SE_NW <- diag(sqrt(abs(NW)))
> t_NW <- b_i/SE_NW
> SE_NW
(Intercept)     Mkt_RF       SMB       HML
0.002527425 0.069918706 0.114355320 0.104112705
> t_NW
(Intercept)     Mkt_RF       SMB       HML
  -2.054010   13.020543   -1.935945   -1.336811       ⇒ SMB close to significant at 5% level
```

- If we add more lags in the NW function (**lag** = **8**)

```
NW <- NeweyWest(fit_ibm, lag = 8, prewhite = FALSE)
SE_NW <- diag(sqrt(abs(NW)))
t_NW <- b_i/SE_NW
> t_NW
(Intercept)     Mkt_RF       SMB       HML
  -2.033648   12.779060   -1.895993   -1.312649      ⇒ not very different results.
```

Example 2: Mexican short-term interest rates with NW SE with **4 lags**:

```
NW <- NeweyWest(fit_i, lag = 4, prewhite = FALSE)
SE_NW <- diag(sqrt(abs(NW)))
t_NW <- b_i/SE_NW
> SE_NW
(Intercept)     us_i_1       e_mx       mx_I       mx_y
 0.01107069  0.55810758  0.01472961  0.51675771  0.93960295
> t_NW
```

```
(Intercept)     us_i_1        e_mx        mx_I        mx_y
  3.6332593   1.5388750  -0.7222770   6.4746121  -0.5305582
```

⇒ $i_{US,t}$ not longer significant at 10% level

• If we add more lags in the text (**lag** = **8**)

```
NW <- NeweyWest(fit_i, lag = 8, prewhite = FALSE)
SE_NW <- diag(sqrt(abs(NW)))
t_NW <- b_i/SE_NW
> t_NW
(Intercept)     us_i_1        e_mx        mx_I        mx_y
  3.0174983   1.4318654  -0.8279016   6.5897816  -0.5825521
```

⇒ similar results.

• There are many estimators of $\mathbf{Q}^*$ based on a specific parametric model for $\mathbf{\Sigma}$. Thus, they are not *robust* to misspecification of (**A3'**). This is the appeal of White & NW.

NW SEs are used almost universally in academia. However:
- NW SEs perform poorly in Monte Carlo simulations:
- NW SEs tend to be downward biased.
- The finite-sample performance of tests using NW SE is not well approximated by the asymptotic theory.
- Tests have size distortions.

Question: What happens if we know the specific form of (**A3'**)? We can do much better than using OLS with NW SEs. In this case, we can do Generalized LS (GLS), a method that delivers the most efficient estimators.

Generalized Least Squares (GLS)

GRM: Assumptions (**A1**), (**A2**), (**A3'**) & (**A4**) hold. That is,

(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \varepsilon$ is correctly specified.

(**A2**) $E[\varepsilon|\mathbf{X}] = 0$

(**A3'**) $Var[\varepsilon|\mathbf{X}] = \mathbf{\Sigma} = \sigma^2\,\mathbf{\Omega}$ ($\mathbf{\Omega}$ is symmetric $\Rightarrow$ $\mathbf{T'T}$= □)

(**A4**) **X** has full column rank –i.e., rank(**X**)=k–, where $T \geq k$.

Question: What happens if we know the specific form of (**A3'**)?
We can use this information to gain efficiency.

When we know (**A3'**), we transform the **y** and **X** in such a way, that we can do again OLS with the transformed data.

To do this transformation, we exploit a property of symmetric matrices, like the variance-covariance matrix, $\mathbf{\Omega}$:

$$\mathbf{\Omega} \text{ is symmetric} \qquad \Rightarrow \text{exists } \mathbf{T} \ni \quad \mathbf{T'T} = \mathbf{\Omega} \qquad \Rightarrow \mathbf{T'}^{-1}\,\mathbf{\Omega}\,\mathbf{T}^{-1} = \mathbf{I}$$

<u>Note</u>: $\mathbf{\Omega}$ can be decomposed as

$$\mathbf{\Omega} = \mathbf{T'T} \qquad (\text{think of } \mathbf{T} \text{ as } \mathbf{\Omega}^{1/2}) \qquad \Rightarrow \mathbf{T'}^{-1}\,\mathbf{\Omega}\,\mathbf{T}^{-1} = \mathbf{I}$$

We transform the linear model in (**A1**) using $\mathbf{P} = \mathbf{\Omega}^{-1/2}$.

$$\mathbf{P} \;=\; \mathbf{\Omega}^{-1/2} \qquad\qquad \Rightarrow \mathbf{P'P} = \mathbf{\Omega}^{-1}$$

$$\mathbf{Py} = \mathbf{PX}\boldsymbol{\beta} + \mathbf{P}\boldsymbol{\varepsilon} \text{ or}$$

$$\mathbf{y^*} \;= \mathbf{X^*}\boldsymbol{\beta} + \boldsymbol{\varepsilon}^*.$$

$$\text{E}[\boldsymbol{\varepsilon}^*\boldsymbol{\varepsilon}^{*\prime}|\mathbf{X^*}] = \mathbf{P}\,\text{E}[\boldsymbol{\varepsilon\varepsilon}'|\mathbf{X^*}\,]\mathbf{P'} = \mathbf{P}\,\text{E}[\boldsymbol{\varepsilon\varepsilon}'|\mathbf{X}]\;\mathbf{P'} = \sigma^2\mathbf{P}□\;\mathbf{P'}$$

$$= \sigma^2\,\mathbf{\Omega}^{-1/2}\,\mathbf{\Omega}\,□^{-1/2} = \;\sigma^2\mathbf{I}_T \qquad \Rightarrow \text{back to } (\mathbf{A3})$$

The transformed model is homoscedastic: We have the CLM framework back □ we can use OLS!

$$\mathbf{b}_{GLS} = \mathbf{b^*} = (\mathbf{X^{*\prime}X^*})^{-1}\,\mathbf{X^{*\prime}y^*}$$

$$= (\mathbf{X'P'PX})^{-1}\,\mathbf{X'P'Py}$$

$$= (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}y}$$

Remarks:

– The transformed model is homoscedastic:

$$\text{Var}[\boldsymbol{\varepsilon}^*|\mathbf{X^*}] = \text{E}[\boldsymbol{\varepsilon}^*\boldsymbol{\varepsilon}^{*\prime}|\mathbf{X^*}]= \mathbf{P}\text{E}[\boldsymbol{\varepsilon\varepsilon}'|\mathbf{X^*}]\mathbf{P'} = \sigma^2\mathbf{P}□\;\mathbf{P'} = \sigma^2\mathbf{I}_T$$

– We have the CLM framework back: We do OLS with the transformed model, we call this OLS estimator, the GLS estimator:

$$\mathbf{b}_{GLS} = \mathbf{b^*} = (\mathbf{X^{*\prime}X^*})^{-1}\,\mathbf{X^{*\prime}y^*} = (\mathbf{X'P'PX})^{-1}\,\mathbf{X'P'Py}$$

$$= (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}y}$$

– Key assumption: $\mathbf{\Omega}$ is known, and, thus, **P** is also known; otherwise we cannot transformed the model.

Big Question: Is $\mathbf{\Omega}$ known?

The GLS estimator is:

$$\mathbf{b}_{GLS} = (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}y}$$

Note I: $\mathbf{b}_{GLS} \neq \mathbf{b}$. $\mathbf{b}_{GLS}$ is BLUE by construction, **b** is not.

• Check unbiasedness:

$$\mathbf{b}_{GLS} \;= (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}y} = (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}}\,(\mathbf{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon})$$

$$= \boldsymbol{\beta} + (\mathbf{X'\Omega^{-1}X})^{-1}\,\mathbf{X'\Omega^{-1}}\,\boldsymbol{\varepsilon}$$

$$\Rightarrow \text{E}[\mathbf{b}_{GLS}\,|\mathbf{X}] = \boldsymbol{\beta}$$

• Efficient Variance

$\mathbf{b}_{GLS}$ is BLUE. The "best" variance can be derived from

$$\text{Var}[\mathbf{b}_{GLS}|\mathbf{X}] = \sigma^2(\mathbf{X^{*\prime}X^*})^{-1} = \sigma^2(\mathbf{X'\Omega^{-1}X})^{-1}$$

Then, the usual OLS variance for **b** is biased and inefficient!

Note II: Both unbiased and consistent. In practice, both estimators will be different, but not that different. If they are very different, something is not kosher.

• Steps for GLS:

Step 1. Find transformation matrix **P** = $\mathbf{\Omega}^{-1/2}$ (in the case of heteroscedasticity, **P** is a diagonal matrix).

Step 2. Transform the model: **X*** = **PX** & **y*** = **Py**.

Step 3. Do GLS; that is, OLS with the transformed variables.

Key step to do GLS: **Step 1**, getting the transformation matrix:

$$\mathbf{P} = \mathbf{\Omega}^{-1/2}.$$

Technical detail: If we relax the CLM assumptions (**A2**) and (**A4**), as we did in Lecture 7-a, we only have asymptotic properties for GLS:

- Consistency - "well behaved data."
- Asymptotic distribution under usual assumptions.
  (easy for heteroscedasticity, complicated for autocorrelation.)
- Wald tests and *F*-tests with usual asymptotic χ^2 distributions.

(Weighted) GLS: Pure Heteroscedasticity

$$(\mathbf{A3'})\;\; Var[\varepsilon] = \mathbf{\Sigma} = \sigma^2\mathbf{\Omega} = \sigma^2\begin{bmatrix} \omega_1 & 0 & \dots & 0 \\ 0 & \omega_2 & \dots & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & \dots & \omega_T \end{bmatrix}$$

$$\mathbf{\Omega}^{-1/2} = \mathbf{P} = \begin{bmatrix} 1/\sqrt{\omega_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\omega_2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1/\sqrt{\omega_T} \end{bmatrix}$$

• Now, transform **y** & **X**:

$$\boldsymbol{y}^* = \mathbf{Py} = \begin{bmatrix} 1/\sqrt{\omega_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\omega_2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1/\sqrt{\omega_T} \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} y_1/\sqrt{\omega_1} \\ y_2/\sqrt{\omega_2} \\ \vdots \\ y_T/\sqrt{\omega_T} \end{bmatrix}$$

• Each observation of y, y_i, is divided by $\sqrt{\omega_i}$. Similar transformation occurs with **X**:

$$\boldsymbol{X}^* = \mathbf{PX} = \begin{bmatrix} 1/\sqrt{\omega_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{\omega_2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1/\sqrt{\omega_T} \end{bmatrix} * \begin{bmatrix} 1 & x_{21} & \cdots & x_{k1} \\ 1 & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{2T} & \cdots & x_{kT} \end{bmatrix} =$$

$$= \begin{bmatrix} 1/\sqrt{\omega_1} & x_{21}/\sqrt{\omega_1} & \dots & x_{k1}/\sqrt{\omega_1} \\ 1/\sqrt{\omega_2} & x_{22}/\sqrt{\omega_2} & \dots & x_{k2}/\sqrt{\omega_2} \\ \vdots & \vdots & \dots & \vdots \\ 1/\sqrt{\omega_T} & x_{2T}/\sqrt{\omega_T} & \dots & x_{kT}/\sqrt{\omega_n} \end{bmatrix}$$

Now, we can do OLS with the transformed variables:

$$\mathbf{b}_{\text{GLS}} = \mathbf{b^*} = (\mathbf{X^{*\prime}X^*})^{-1}\, \mathbf{X^{*\prime}y^*} = (\mathbf{X'\Omega^{-1}X})^{-1}\, \mathbf{X'\Omega^{-1}y}$$

In the case of heteroscedasticity, GLS is also called *Weighted Least Squares* (WLS): Think of $[\omega_i]^{-1/2}$ as weights. The GLS estimator is:

$$\mathbf{b}_{\text{GLS}} = (X'\Omega^{-1}X)^{-1}(X'\Omega^{-1}y) = \left(\sum_{i=1}^{T} \frac{1}{\omega_i} x_i x_i'\right)^{-1} \left(\sum_{i=1}^{T} \frac{1}{\omega_i} x_i y_i\right)$$

Observations with lower (bigger) variances –i.e., lower (bigger) ω_i– are given higher (lower) weights in the sums: More precise observations, more weight!

The GLS variance is given by:

$$\hat{\sigma}_{GLS}^2 = \frac{\sum_{i=1}^{T} \left(\frac{y_i - x_i'\,\mathbf{b}_{\text{GLS}}}{\omega_i}\right)^2}{T-K}$$

Example: Last Lecture, we found that squared market returns (Mkt_RF^2) influence the heteroscedasticity in DIS returns. Suppose we assume: **(A3')** $\sigma_i^2 = (\text{Mkt\_RT}_i)^2$.
Steps for GLS:

Step 1. Find transformation matrix, **P**, with *i*th diagonal element: $1/\sqrt{\sigma_i^2}$

Step 2. Transform model: Each y_i and x_i is divided ("weighted") by
$\sigma_i =$ sqrt[(Mkt_RT$_i$)^2].
Step 3. Do GLS, that is, OLS with transformed variables.

```
T <- length(dis_x)
Mkt_RF2 <- Mkt_RF^2                                 # (A3')
y_w <- dis_x/sqrt(Mkt_RF2)                          # transformed y = y*
x0 <- matrix(1,T,1)
xx_w <- cbind(x0, Mkt_RF, SMB, HML)/sqrt(Mkt_RF2)   # transformed X = X*
fit_dis_wls <- lm(y_w ~ xx_w)                       # GLS
> summary(fit_dis_wls)
Call:
lm(formula = y_w ~ xx_w)
Residuals:
   Min      1Q  Median      3Q     Max
-59.399  -0.891   0.316   1.503  77.434
Coefficients:
              Estimate   Std. Error t value Pr(>|t|)
xx_w          -0.006607  0.001586  -4.165 3.59e-05 ***
xx_wMkt_RF  1.588057   0.334771   4.744 2.66e-06 ***           ⇒ OLS b: 1.26056
xx_wSMB      -0.200423  0.067498  -2.969  0.00311 **            ⇒ OLS b:  -0.028993
```

```
xx_wHML     -0.042032   0.072821  -0.577  0.56404               ⇒ OLS b:  0.174545
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 7.984 on 566 degrees of  freedom
Multiple R-squared:  0.09078,   Adjusted R-squared:  0.08435
F-statistic: 14.13 on 4 and 566 DF,  p-value: 5.366e-11
```

GLS: First-order Autocorrelation Case

We assume an AR(1) process for the ε_t:

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t, \qquad u_t\text{: non-autocorrelated error} \sim D(0, \sigma_u^2)$$

- Steps for GLS:

Step 1. To find the transformation matrix **P**, we need to derive the implied (**A3'**) based on the AR(1) process for ε_t:

(1) Find diagonal elements of **Ω**: $\text{Var}[\varepsilon_t] = \sigma_{ii} = \sigma_\varepsilon^2$

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t \quad \text{(the autoregressive form)}$$

$\Rightarrow \quad \text{Var}[\varepsilon_t] = \rho^2\,\text{Var}[\varepsilon_{t-1}] + \text{Var}[u_t] \qquad (\text{Var}[\varepsilon_t]=\text{Var}[\varepsilon_{t-1}]= \sigma_\varepsilon^2)$

$\Rightarrow \quad \sigma_\varepsilon^2 = \sigma_u^2/(1-\rho^2)$ $\qquad$ –we need to assume $|\rho|< 1$.

(2) Find the off-diagonal elements of **Ω**: $\sigma_{ij}= \gamma_{l=j-i}$:

$\Rightarrow \quad \sigma_{ij} = \gamma_l = \text{Cov}[\varepsilon_i, \varepsilon_j] = E[\varepsilon_i\, \varepsilon_j] \qquad l = j - i$

$$\begin{aligned}\gamma_1 = Cov[\varepsilon_t, \varepsilon_{t-1}] &= E[(\rho\varepsilon_{t-1} + u_t)\, \varepsilon_{t-1}] \\ &= \rho\, E[\varepsilon_{t-1}\, \varepsilon_{t-1}] + E[u_t\, \varepsilon_{t-1}] \\ &= \rho\, Var[\varepsilon_{t-1}] = \rho\, \sigma_\varepsilon^2 \\ &= \frac{\rho\sigma_u^2}{(1-\rho^2)}\end{aligned}$$

$$\begin{aligned}\gamma_2 = Cov[\varepsilon_t, \varepsilon_{t-2}] &= E[(\rho\varepsilon_{t-1} + u_t)\, \varepsilon_{t-2}] \\ &= \rho\, E[\varepsilon_{t-1}\, \varepsilon_{t-2}] + E[u_t\, \varepsilon_{t-2}] \\ &= \rho\, Cov[\varepsilon_t, \varepsilon_{t-1}] = \rho\, \gamma_1\end{aligned}$$

$$\begin{aligned}\gamma_3 = Cov[\varepsilon_t, \varepsilon_{t-3}] &= E[(\rho\varepsilon_{t-1} + u_t)\, \varepsilon_{t-3}] \\ &= \rho\, E[\varepsilon_{t-1}\, \varepsilon_{t-3}] + E[u_t\, \varepsilon_{t-3}] \\ &= \rho\, Cov[\varepsilon_t,\ \varepsilon_{t-2}] = \rho\, \gamma_2 \\ &= \rho^2\gamma_1 = \frac{\rho^3\sigma_u^2}{(1-\rho^2)}\end{aligned}$$

$$\vdots$$

$$\gamma_l = Cov[\varepsilon_t, \varepsilon_{t-l}] = \rho^{l-1}\,\gamma_1$$

- If we define $\gamma_0 = \sigma_\varepsilon^2 = \sigma_u^2/(1 - \rho^2)$, then we generalize the *autocovariance function*, γ_l:

$$\gamma_l = \rho^l\, \gamma_0$$

- Now, we get (**A3'**) $\mathbf{\Sigma} = \sigma^2\,\mathbf{\Omega}$.

(A3') $$\boldsymbol{\sigma^2\Omega} = \left(\frac{\sigma_u^2}{1-\rho^2}\right)\begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{T-1} \\ \rho & 1 & \rho & \cdots & \rho^{T-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{T-1} & \rho^{T-2} & \rho^{T-3} & \cdots & 1 \end{bmatrix}$$

$$\boldsymbol{\Omega^{-1/2}} = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & 0 & \dots & 0 \\ -\rho & 1 & 0 & \dots & 0 \\ 0 & -\rho & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & -\rho & 0 \end{bmatrix}$$

Step 2. With **P** = $\boldsymbol{\Omega}^{-1/2}$, we transform the data (**y** & **X**) to do GLS:

$$\mathbf{P} = \boldsymbol{\Omega^{-1/2}} = \begin{bmatrix} \sqrt{1-\rho^2} & 0 & 0 & \dots & 0 \\ -\rho & 1 & 0 & \dots & 0 \\ 0 & -\rho & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & -\rho & 0 \end{bmatrix}$$

$$\mathbf{y}^* = \mathbf{P}\,\boldsymbol{y} = \begin{pmatrix} \left(\sqrt{1-\rho^2}\right) y_1 \\ y_2 - \rho y_1 \\ y_3 - \rho y_2 \\ \dots \\ y_T - \rho_{T-1} \end{pmatrix}$$

$$\boldsymbol{x}_k^* = \mathbf{P}\,\boldsymbol{x}_k = \begin{pmatrix} \left(\sqrt{1-\rho^2}\right) x_{k1} \\ x_{k2} - \rho\, x_{k1} \\ x_{k3} - \rho\, x_{k2} \\ \dots \\ x_T - \rho\, x_{T-1} \end{pmatrix}$$

Step 3. GLS is done with transformed data. In (**A3**') we assume ρ known. In practice, we need to estimate it.

GLS: The Autoregressive Transformation

With AR models, sometimes it is easier to transform the data by taking *pseudo differences.*

- For the AR(1) model, we multiply the DGP by ρ and subtract it from it. That is,

$$y_t \;=\; \boldsymbol{x}_t{}'\boldsymbol{\beta} + \varepsilon_t, \qquad\qquad \varepsilon_t = \rho\varepsilon_{t-1} + u_t$$

$$\rho y_{t-1} = \rho\boldsymbol{x}_{t-1}{}'\boldsymbol{\beta} + \rho\varepsilon_{t-1}$$

$$- - - - - - - - - - - - - - - -$$

$$y_t - \rho y_{t-1} = (\boldsymbol{x}_t - \rho\boldsymbol{x}_{t-1})'\boldsymbol{\beta} + (\varepsilon_t - \rho\varepsilon_{t-1})$$

$$y_t^* = \boldsymbol{x}_t^*{}'\boldsymbol{\beta} + u_t$$

Now, we have the errors, u_t, which are uncorrelated. We can do OLS with the pseudo differences.

<u>Note</u>: $y_t^* = y_t - \rho y_{t-1}$ & $\boldsymbol{x}_t^* = \boldsymbol{x}_t - \rho\boldsymbol{x}_{t-1}$ are *pseudo differences*.

FGLS: Unknown Ω

The problem with GLS is that **Ω** is unknown. For example, in the AR(1) case, ρ is unknown.

<u>Solution</u>: Estimate **Ω**. □ *Feasible GLS* (FGLS).

In general, there are two approaches for GLS

(1) Two-step, or *Feasible estimation*:- First, estimate **Ω** first.
- Second, do GLS.

Similar logic to HAC procedures: We do not need to estimate **Ω**, difficult with *T* observations. We estimate (1/*T*)**X′Ω**$^{-1}$**X**.

—Nice asymptotic properties for FGLS estimator. Not longer BLUE

(2) ML estimation of β, σ^2, and **Ω** at the same time (joint estimation of all parameters). With some exceptions, rare in practice.

FGLS: Specification of Ω

• **Ω** must be specified first.

• **Ω** is generally specified (modeled) in terms of a few parameters. Thus, **Ω** = **Ω**(**θ**) for some small parameter vector **θ**. Then, we need to estimate **θ**.

Examples:

(1) Var[ε_i|**X**] = σ^2 f(γ′**z**$_i$). Variance a function of γ and some variable **z**$_i$ (say, market volatility, firm size, country dummy, etc). In general, f is an exponential to make sure the variance is positive.

(2) ε_i with AR(1) process. We have already derived σ^2 **Ω** as a function of □.

<u>Technical note</u>: To achieve full efficiency, we do not need an *efficient* estimate of the parameters in **Ω**, only a *consistent* one.

FGLS: Estimation – Steps

Steps for FGLS:

Step 1. Estimate the model proposed in (**A3'**). Get $\hat{\sigma}_i^2$ & $\hat{\sigma}_{ij}$

Step 2. Find transformation matrix, **P**, using the estimated $\hat{\sigma}_i^2$ & $\hat{\sigma}_{ij}$.

Step 3. Using **P** from Step 2, transform model: **X***= **PX** and **y***= **Py**.

Step 4. Do FGLS, that is, OLS with **X*** & **y***.

Example: In the pure heteroscedasticity case (**P** is diagonal):

1. Estimate the model proposed in (**A3'**). Get $\hat{\sigma}_i^2$.
2. Find transformation matrix, **P**, with i^{th} diagonal element: $1/\hat{\sigma}_i$
3. Transform model: Each y_i and x_i is divided ("weighted") by $\hat{\sigma}_i$.
4. Do FGLS, that is, OLS with transformed variables.

FGLS Estimation: Heteroscesdasticity

Example: Last lecture, we found that Mkt_RF^2 and SMB^2 are drivers of the heteroscedasticity in DIS returns: Suppose we assume: (**A3'**) $\sigma_i^2 = \gamma_0 + \gamma_1\, (Mkt\_RT_i)^2 + \gamma_3\, (SMB_i)^2$

- Steps for FGLS:

1. Use OLS squared residuals to estimate (**A3'**):

```
fit_dis <- lm(dis_x ~ Mkt_RF + SMB + HML)
e_dis <- fit_dis$residuals
e_dis2 <- e_dis^2
fit_dis2 <- lm(e_dis2 ~ Mkt_RF2 + SMB2)
summary(fit_dis2)
var_dis2 <- fit_dis2$fitted                     # Estimated variance vector, with elements
```

$\hat{\sigma}_i^2$.

2. Find transformation matrix, **P**, with i^{th} diagonal element: $1/\hat{\sigma}_i$

```
w_fgls <- sqrt(var_dis2)                        # 1/σ̂_i
```

3. Transform model: Each y_i and x_i is "weighted" by $1/\hat{\sigma}_i$.

```
y_fw <- dis_x/w_fgls                            # transformed y
xx_fw <- cbind(x0, Mkt_RF, SMB, HML)/w_fgls          # transformed X
```

4. Do GLS, that is, OLS with transformed variables.

```
fit_dis_fgls <- lm(y_fw ~ xx_fw - 1)
> summary(fit_dis_fgls)
Coefficients:
               Estimate   Std. Error t value Pr(>|t|)
xx_fw          -0.003097  0.002696  -1.149    0.251
xx_fwMkt_RF     1.208067  0.073344  16.471   <2e-16 ***
xx_fwSMB       -0.043761  0.105280  -0.416    0.678
xx_fwHML        0.125125  0.100853   1.241    0.215     ⇒ not longer significant at 10%.
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.9998 on 566 degrees of freedom
Multiple R-squared:  0.3413,    Adjusted R-squared:  0.3366
F-statistic: 73.31 on 4 and 566 DF,  p-value: < 2.2e-16
```

• Comparinig OLS, GLS & FGLS results:

	$\mathbf{b}_{OLS}$	**SE**	$\mathbf{b}_{GLS}$	**SE**	$\mathbf{b}_{FGLS}$	**SE**
Intercept	0.00417	0.00279	-0.00661	0.00159	-0.00310	0.00270
Mkt_RF	**1.26056**	0.06380	**1.58806**	0.33477	**1.20807**	0.07334
SMB	-0.02899	0.09461	-0.20042	0.06750	-0.04376	0.10528
HML	0.17455	0.09444	-0.04203	0.07282	0.12513	0.10085

• Comments:

- The GLS estimates are quite different than OLS estimates (remember OLS is unbiased and consistent). Very likely the assumed functional form in (**A3'**) was not a good one.
- The FGLS results are similar to the OLS, as expected, if model is OK. FGLS is likely a more precise estimator (HML is not longer significant at 10%.

FGLS Estimation: AR(1) Case – Cochrane-Orcutt

In the AR(1) case, it is easier to estimate the model in *pseudo differences*:

$$\mathbf{y}_t^* = \mathbf{X}_t^* \, \beta + u_t$$

$$\mathbf{y}_t - \rho \mathbf{y}_{t-1} = (\mathbf{X}_t - \rho \mathbf{X}_{t-1})' \, \beta + \varepsilon_t - \rho \varepsilon_{t-1}$$

$$\Rightarrow \mathbf{y}_t = \rho \mathbf{y}_{t-1} + \mathbf{X}_t' \, \beta - \mathbf{X}_{t-1}' \, \rho\beta + u_t$$

We have a linear model, but it is nonlinear in parameters. OLS is not possible, but non-linear estimation is possible.

Before today's computer power, Cochrane–Orcutt's (1949) iterative procedure was an ingenious way to do this estimation.

• Steps for Cocharne-Orcutt:

(0) Do OLS in (**A1**) model: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$. Get residuals, **e**, & RSS.

(1) Estimate ρ with a regression of $\mathbf{e}_t$ against $\mathbf{e}_{t-1}$ ⇒ get *r* (the estimator of ρ).

(2) FGLS Step. Use *r* transform the model to get **y*** and **X***.

Do OLS with **y*** and **X*** ⇒ get **b** to estimate β.

Get residuals, $\mathbf{e}^* = \mathbf{y} - \mathbf{X}\,\mathbf{b}$, and new RSS. Go back to (1).

(3) Iterate until convergence, usually achieved when the difference in RSS of two consecutive iterations is lower than some tolerance level, say .0001. Then, stop when $RSS_i – RSS_{i-1} < .0001$.

Example: Cochrane-Orcutt in R

```
# C.O. function requires Y, X (with constant), OLS b.
c.o.proc <- function(Y,X,b_0,tol){
  T <- length(Y)
  e <- Y - X%*%b_0                           # OLS residuals
rss <- sum(e^2)                                        # Initial RSS of model, RSS9
rss_1 <- rss                                 # RSS_1 will be used to reset RSS after each
iteration
d_rss = rss                                  # initialize d_rss: difference between RSSi & RSSi-1
  e2 <- e[-1]                                # adjust sample size for et
          e3 <- e[-T]                        # adjust sample size for et-1
  ols_e0 <- lm(e2 ~ e3 - 1)                  # OLS to estimate rho
  rho <- ols_e0$coeff[1]                               # initial value for rho, ρ0
 i<-1
while (d_rss > tol) {                        # tolerance of do loop. Stop when diff in RSS < tol
rss <- rss_1      # RSS at iter (i-1)
   YY <- Y[2:T] - rho * Y[1:(T-1)]                     # pseudo-diff Y
   XX <- X[2:T, ] - rho * X[1:(T-1), ]                 # pseudo-diff X
ols_yx <- lm(YY ~ XX - 1)                    # adjust if constant included in X
b <- ols_yx$coef                             # updated OLS b at iteration i
#  b[1] <- b[1]/(1-rho)                      # If constant not pseudo-differenced remove tag #
e1 <- Y - X%*%b                              # updated residuals at iteration i
           e2 <- e1[-1]                      # adjust sample size for updated et
   e3 <- e1[-T]                              # adjust sample size for updated e_t-1 (lagged et)
   ols_e1 <- lm(e2~e3-1)                               # updated regression to value for rho at
iteration i
   rho <- ols_e1$coeff[1]                              # updated value of rho at iteration i, ρi
 rss_1 <- sum(e1^2)                          # updated value of RSS at iteration i, RSSi
 d_rss <- abs(rss_1 - rss)                             # diff in RSS (RSSi - RSSi-1)
i <- i+1
   }
 result <-list()
 result$Cochrane-Orc.Proc <- summary(ols_yx)
 result$rho.regression <- summary(ols_e1)
 #  result$Corrected.b_1 <- b[1]
 result$Iterations < -i-1
 return(result)
}
```

Example: In the model for **Mexican interest rates** (i_{MX}), we suspect an AR(1) in the residuals:

$$i_{MX,t} = \beta_0 + \beta_1\, i_{US,t} + \beta_2\, e_t + \beta_3\, mx\_I_t + \beta_4\, mx\_y_t + \varepsilon_t$$

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t$$

- Cochrane-Orcutt estimation.

```
y <- mx_i_1
T_mx <- length(mx_i_1)
xx_i <- cbind(us_i_1, e_mx, mx_I, mx_y)
x0 <- matrix(1,T_mx,1)
X <- cbind(x0,xx_i)                                   # X matrix
fit_i <- lm(mx_i_1 ~ us_i_1 + e_mx + mx_I + mx_y)
b_i <-fit_i$coefficients                                  # extract coefficients from lm
> summary(fit_i)

Coefficients:
            Estimate   Std. Error  t value Pr(>|t|)
(Intercept) 0.04022    0.01506     2.671   0.00834 **
us_i_1      0.85886    0.31211     2.752   0.00661 **
e_mx        -0.01064   0.02130     -0.499  0.61812
mx_I        3.34581    0.19439     17.212  < 2e-16 ***
mx_y        -0.49851   0.73717     -0.676  0.49985


> c.o.proc(y,X,b,.0001)
$Cochrane.Orcutt.Proc
Call:
lm(formula = YY ~ XX - 1)
Residuals:
     Min       1Q   Median       3Q      Max
-0.69251 -0.02118 -0.01099  0.00538  0.49403
Coefficients:
            Estimate   Std. Error  t value Pr(>|t|)
XX          0.16639    0.07289     2.283   0.0238 *
XXus_i_1    1.23038    0.76520     1.608   0.1098      ⇒ not longer significant at 5% level.
XXe_mx      -0.00535   0.01073     -0.499   0.6187
XXmx_I      0.41608    0.27260     1.526   0.1289      ⇒ not longer significant at 5% level.
XXmx_y      -0.44990   0.53096     -0.847   0.3981
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.09678 on 160 degrees of freedom
Multiple R-squared:  0.1082,    Adjusted R-squared:  0.08038
F-statistic: 3.884 on 5 and 160 DF,  p-value: 0.002381

$rho
       e3
0.8830857                     ⇒ very high autocorrelation.

$Corrected.b_1
       XX
```

```
0.1663884                    ⇒ Constant corrected if X does not include a constant

$Number.Iteractions
[1] 10                       ⇒ algorithm converged in 10 iterations.
```

GLS: General Remarks

GLS is great (BLUE) if we know **Ω**. Very rare case.

It needs the specification of **Ω** –i.e., the functional form of autocorrelation and heteroscedasticity.

If the specification is bad ⇒ estimates are biased.

In general, GLS is used for larger samples, because more parameters need to be estimated.

Feasible GLS is not BLUE (unlike GLS); but, it is consistent and asymptotically more efficient than OLS.

We use GLS for inference and/or efficiency. OLS is still unbiased and consistent.

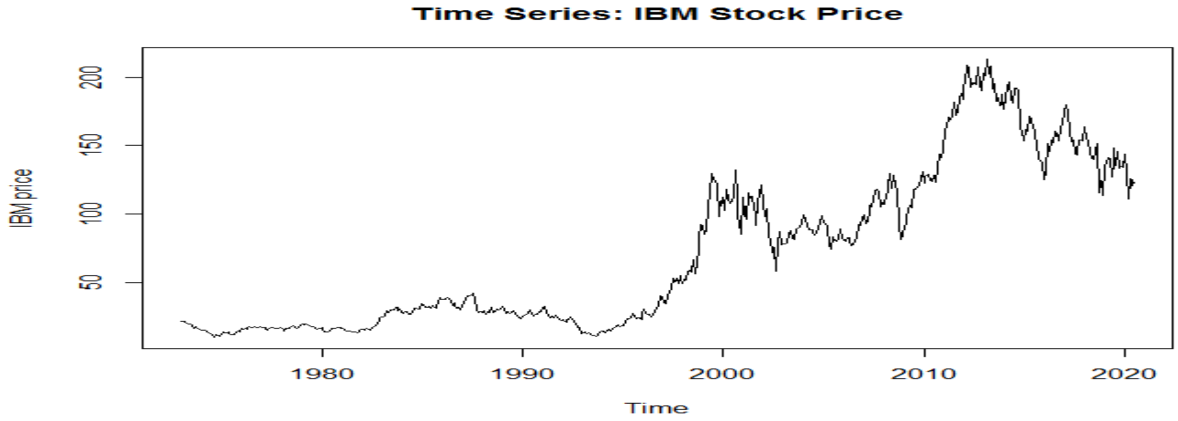
OLS and GLS estimates will be different due to sampling error. But, if they are very different, then it is likely that some other CLM assumption is violated.

Lecture 8 - Time Series

Time Series: Introduction

• A time series y_t is a process observed in sequence over time,
$t = 1, ..., T \quad \Rightarrow \mathbf{Y}_t = \{y_1, y_2, y_3, ..., y_T\}$.

Examples: IBM monthly stock prices from 1973:January to 2020:September (plot below); or USD/GBP daily exchange rates from February 15, 1923 to March 19, 1938.



• Different ways to do the plot in R:
- Using plot.ts, creating a timeseries object in R:

```
ts_ibm <- ts(x_ibm,start=c(1973,1),frequency=12)
# the function ts creates a timeseries object, start = starting year,

plot.ts(ts_ibm,xlab="Time",ylab="IBM price", main="Time Series: IBM Stock Price")
```

- Using R package ggplot2

```
library(ggplot2)
ggplot(data= SFX_da, aes(x = x_date1, y = x_ibm)) +
  geom_line() +
  labs(x = "Date",
    y = "IBM price",
    title = "Time Series: IBM ",
    subtitle = "Monthly: 1973-2020")
```

Time Series: Introduction – Types

Usually, time series models are separated into two categories:
– Univariate ($y_t \in R$, it is a scalar)
Example: We are interested in the behavior of IBM stock prices as function of its past.
$\Rightarrow$ Primary model: Autoregressions (ARs).

– Multivariate ($y_t \in R^m$, it is a vector-valued)

Example: We are interested in the joint behavior of IBM stock and bond prices as function of their joint past.

⇒ Primary model: Vector autoregressions (VARs).

Time Series: Introduction – Dependence

Given the sequential nature of Y_t, we expect y_t & y_{t-1} to be dependent. This is the main feature of time series: dependence. It creates statistical problems.

In classical statistics, we usually assume we observe several *i.i.d.* realizations of Y_t. We use $\bar{Y}$ to estimate the mean.

With several independent realizations we are able to sample over the entire probability space and obtain a "good" –i.e., consistent or close to the population mean– estimator of the mean.

But, if the samples are highly dependent, then it is likely that Y_t is concentrated over a small part of the probability space. Then, the sample mean will not converge to the mean as the sample size grows.

Technical note: With dependent observations, the classical results (based on LLN & CLT) are not to valid. New assumptions and tools are needed: stationarity, ergodicity, CLT for martingale difference sequences (MDS CLT).

Roughly speaking, stationarity requires constant moments for Y_t; ergodicity requires that the dependence is short-lived, eventually y_t has only a small influence on y_{t+k}, when k is relatively large.

The amount of dependence in Y_t determines the 'quality' of the estimator. There are several ways to measure the dependence. The most common measure: Covariance.

$$\text{Cov}(Y_t, Y_{t+k}) = E[(Y_t - \mu)(Y_{t+k} - \mu)]$$

Note: When μ =0, then $\text{Cov}(Yt, Y_{t+k}) = E[Y_t \, Y_{t+k}]$

Time Series: Introduction – Forecasting

In a time series model, we describe how y_t depends on past y_t's. That is, the information set is $I_t = \{y_{t-1}, y_{t-2}, y_{t-3}, \ldots.\}$

The purpose of building a time series model: Forecasting.

We estimate time series models to forecast out-of-sample. For example, the *l-step ahead* forecast:

$$\hat{y}_{T+l} = E_t[y_{t+l} | I_t].$$

In the 1970s it was found that very simple time series models out-forecasted very sophisticated (big) economic models. This finding represented a big shock to the big multivariate models that were very popular then. It forced a re-evaluation of these big models.

Time Series: Introduction – White Noise

In general, we assume the error term, ε_t, is uncorrelated with everything, with mean 0 and constant variance, σ^2. We call a process like this a *white noise* (WN) *process*.

We denote a WN process as

$$\varepsilon_t \sim WN(0, \sigma^2)$$

White noise is the basic building block of all time series. It can be written as:

$$z_t = \sigma\, u_t, \qquad u_t \sim i.i.d\,(0, 1) \qquad \Rightarrow z_t \sim WN(0, \sigma^2)$$

The z_t's are random shocks, with no dependence over time, representing unpredictable events. It represents a model of news.

Time Series: Introduction – Conditionality

We make a key distinction: *Conditional* vs *Unconditional* moments. In time series we model the *conditional mean* as a function of its past, for example in an AR(1) process, we have:

$$y_t = \alpha + \beta\, y_{t-1} + \varepsilon_t.$$

Then, the conditional mean forecast at time *t*, conditioning on information at time I_{t-1}, is:

$$E_t[\, y_t|I_{t-1}] = E_t[\, y_t] = \alpha + \beta\, y_{t-1}$$

Notice that the unconditional mean is given by:

$$E[y_t] = \alpha + \beta\, E[y_{t-1}] = \alpha/(1 - \beta) = \text{constant}$$

The conditional mean is time varying; the unconditional mean is not!

Key distinction: Conditional vs. Unconditional moments.

Time Series: Introduction – AR and MA models

Two popular models for $E[y_t|I_{t-1}]$:

– An autoregressive (AR) process models $E[y_t|I_{t-1}]$ with lagged dependent variables:

$$E[y_t|I_{t-1}] = f(y_{t-1}, y_{t-2}, y_{t-3}, ...)$$

Example: AR(1) process, $y_t = \alpha + \beta\, y_{t-1} + \varepsilon_t.$

– A moving average (MA) process models $E[y_t|I_{t-1}]$ with lagged errors, ε_t:

$$E[y_t|I_{t-1}] = f(\varepsilon_{t-1}, \varepsilon_{t-2}, \varepsilon_{t-3}, ...)$$

Example: MA(1) process, $y_t = \mu + \theta_1\, \varepsilon_{t-1} + \varepsilon_t$

– There is a third model, ARMA, that combines lagged dependent variables and lagged errors.

• We want to select an appropriate time series model to forecast y_t. In this class, we will use linear model, with choices: AR(p), MA(q) or ARMA(p, q).

• Steps for forecasting:
(1) Identify the appropriate model. That is, determine p, q.
(2) Estimate the model.
(3) Test the model.
(4) Forecast.

CLM Revisited: Time Series

With autocorrelated data, we get dependent observations. For example, with autocorrelated errors:

$$\varepsilon_t = \rho\varepsilon_{t-1} + u_t,$$

the independence assumption (A2) is violated. The LLN and the CLT cannot be easily applied in this context. We need new tools.

We introduce the concepts of *stationarity* and *ergodicity.* The ergodic theorem will give us a counterpart to the LLN.

To get asymptotic distributions, we also need a CLT for dependent variables, using new technical concepts: mixing and stationarity. Or we can rely on a new CLT: The *martingale difference sequence CLT.*

• We will not cover these technical points in detail.

Time Series – Stationarity

Consider the joint probability distribution of the collection of RVs:

$$F(y_{t_1}, y_{t_2}, \dots, y_{t_T}) = F(Y_{t_1} \le y_{t_1}, Y_{t_2} \le y_{t_2}, \dots, Y_{t_T} \le y_{t_T})$$

To do statistical analysis with dependent observations, we need some extra assumptions. We need some form of invariance on the structure of the time series.

If the distribution F is changing with every observation, estimation and inference become very difficult.

Stationarity is an invariant property: the statistical characteristics of the time series do not change over time.

There different definitions of stationarity, they differ in how strong is the invariance of the distribution over time.

We say that a process is stationary of
1st order if $\quad F(y_{t_1}) = F(y_{t_1+k}) \quad$ for any t_1, k

2nd order if $F(y_{t_1}, y_{t_2}) = F(y_{t_{1+k}}, y_{t_{2+k}})$ for any t_1, t_2, k
Nth-order if $F(y_{t_1}, \ldots, y_{t_T}) = F(y_{t_{1+k}}, \ldots, y_{t_{T+k}})$ for any t_1, ..., t_T, k

Nth-order stationarity is a strong assumption (& difficult to verify in practice). *2nd order* stationarity is weaker: only consider mean and covariance (easier to verify in practice).

Moments describe a distribution. We calculate moments as usual:

$$\mathrm{E}[Y_t] = \mu$$
$$\mathrm{Var}(Y_t) = \sigma^2 = E[(Y_t - \mu)^2]$$
$$\mathrm{Cov}(Y_{t_1}, Y_{t_2}) = E[(Y_{t_1} - \mu)(Y_{t_2} - \mu)] = \gamma(t_1 - t_2)$$

Time Series – Stationarity, Autocovariances & Autocorrelations

$\mathrm{Cov}(Y_{t_1}, Y_{t_2}) = \gamma(t_1 - t_2)$ is called the *auto-covariance function.*

Notes: $\gamma(t_1 - t_2)$ is a function of $k = t_1 - t_2$.
$\gamma(0)$ is the variance.

The autocovariance function is symmetric. That is,

$$\gamma(t_1 - t_2) = \mathrm{Cov}(Y_{t_1}, Y_{t_2}) = \mathrm{Cov}(Y_{t_2}, Y_{t_1}) = \gamma(t_2 - t_1)$$

Autocovariances are unit dependent. We will have different values if we calculate the autocovariance for IBM returns in % terms or in decimal terms.

Remark: The autocovariance measures the (linear) dependence between the Y_t's separated by k periods.

From the autocovariances, we derive the autocorrelations:

$$\mathrm{Corr}(Y_{t_1}, Y_{t_2}) = \rho(Y_{t_1}, Y_{t_2}) = \frac{\gamma(t_1 - t_2)}{\sigma_{t_1}\sigma_{t_2}} = \frac{\gamma(t_1 - t_2)}{\gamma(0)}$$

the last step takes assumes: $\sigma_{t_1} = \sigma_{t_2} = \sqrt{\gamma(0)}$

$\mathrm{Corr}(Y_{t_1}, Y_{t_2}) = \rho(Y_{t_1}, Y_{t_2})$ is called the *auto-correlation function* (ACF), –think of it as a function of $k = t_1 - t_2$. The ACF is also symmetric.

Unlike autocovoriances, autocorrelations are not unit dependent. It is easier to compare dependencies across different time series.

Stationarity requires all these moments to be independent of time. If the moments are time dependent, we say the series is *non-stationary.*

For strictly stationary process (constant moments), we need:

$$\mu_t = \mu$$
$$\sigma_t = \sigma$$

because $F(y_{t_1}) = F(y_{t_{1+k}}) \Rightarrow \mu_{t_1} = \mu_{t_{1+k}} = \mu$

$$\sigma_{t_1} = \sigma_{t_{1+k}} = \sigma$$

Then,

$$F(y_{t_1}, y_{t_2}) = F(y_{t_{1+k}}, y_{t_{2+k}}) \Rightarrow \text{Cov}(y_{t_1}, y_{t_2}) = \text{Cov}(y_{t_{1+k}}, y_{t_{2+k}})$$
$$\Rightarrow \rho(t_1, t_2) = \rho(t_1 + k, t_2 + k)$$

Let $t_1 = t - k$ & $t_2 = t$

$$\Rightarrow \rho(t_1, t_2) = \rho(t - k, t) = \rho(t, t - k) = \rho(k) = \rho_k$$

The correlation between any two RVs depends on the time difference. Given the symmetry, we have $\rho(k) = \rho(-k)$.

Time Series – Weak Stationarity

A *Covariance stationary* process (or *2nd -order weakly stationary*) has:

- constant mean
- constant variance
- covariance function depends on time difference between RVs.

That is, Z_t is covariance stationary if:

$\text{E}(Z_t) = \text{constant} = \mu$

$\text{Var}(Z_t) = \text{constant} = \sigma$

$\text{Cov}(Z_{t_1}, Z_{t_2}) = \text{E}[(Z_{t_1} - \mu_{t_1})(Z_{t_2} - \mu_{t_2})] = \gamma(t_1 - t_2) = f(t_1 - t_2)$

Remark: Covariance stationarity is only concerned with the covariance of a process, only the mean, variance and covariance are time-invariant. N^{th}*-order* stationarity is stronger and assumes that the whole distribution is invariant over time.

Example: Stationary time series. Assume $\varepsilon_t \sim WN(0, \sigma^2)$.

$y_t = \phi\, y_{t-1} + \varepsilon_t$ (AR(1) process)

• Mean

Taking expectations on both side:

$E[y_t] = \phi\, E[y_{t-1}] + E[\varepsilon_t]$

$\mu = \phi\, \mu + 0$

$\Rightarrow E[y_t] = \mu = 0$ (assuming $\phi \neq 1$)

• Variance

Applying the variance on both side:

$\text{Var}[y_t] = \gamma(0) = \phi^2\, \text{Var}[y_{t-1}] + \text{Var}[\varepsilon_t]$

$\gamma(0) = \sigma^2/(1 - \phi^2)$ (assuming $|\phi| < 1$)

• Autocovariances

$\gamma(1) = \text{Cov}[y_t, y_{t-1}] = E[y_t\, y_{t-1}] = E[(\phi\, y_{t-1} + \varepsilon_t) y_{t-1}]$

$= \phi\, E[y_{t-1}^2] = \phi\, \text{Var}[y_{t-1}^2] = \phi\, \gamma(0)$

$= \phi\, [\sigma^2/(1 - \phi^2)]$

$\gamma(2)$ = Cov[y_t, y_{t-2}] = E[y_t y_{t-2}] = E[(ϕ y_{t-1} + ε_t)y_{t-2}]

= ϕ E[y_{t-1}, y_{t-2}] = ϕ Cov[y_{t-1}, y_{t-2}] = ϕ ϕ $\gamma(0)$ = ϕ^2 $\gamma(0)$

= ϕ^2 [σ^2/(1- ϕ ^{2})]

…

$\gamma(k)$ = Cov[y_t, y_{t-k}] = ϕ^k $\gamma(0)$

$\Rightarrow$ If |ϕ |< 1, the process is covariance stationary: mean, variance and covariance are constant.

Note: From the autocovariance function, we can derive the auto-correlation function:

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{\phi^k\,\gamma(0)}{\gamma(0)} = \phi^k$$

Example: Non-stationary time series. Assume $\varepsilon_t \sim WN(0, \sigma^2)$.

$y_t = \mu + y_{t-1} + \varepsilon_t$ (Random Walk with drift process)

Doing backward substitution:

$y_t = \mu + (\mu + y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t$

$= 2*\mu + y_{t-2} + \varepsilon_t + \varepsilon_{t-1}$

$= 2*\mu + (\mu + y_{t-3} + \varepsilon_{t-2}) + \varepsilon_t + \varepsilon_{t-1}$

$= 3*\mu + y_{t-3} + \varepsilon_t + \varepsilon_{t-1} + \varepsilon_{t-2}$

$\Rightarrow y_t = \mu\, t + \sum_{j=0}^{t-1} \varepsilon_{t-j} + y_0$

• **Mean** & **Variance**

E[y_t] = μ t + y_0

Var[y_t] = $\gamma(0) = \sum_{j=0}^{t-1} \sigma^2 = \sigma^2$ t

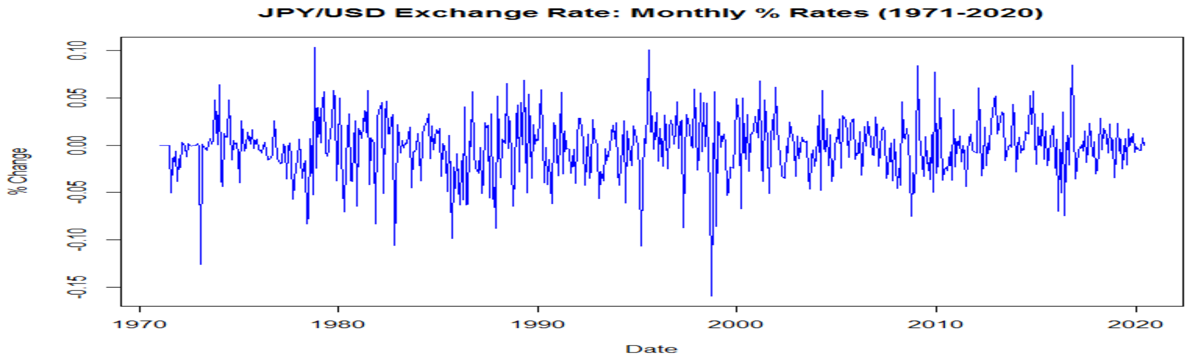
$\Rightarrow$ the process is non-stationary; that is, moments are time dependent.

Stationary Series – Examples

Examples: Assume $\varepsilon_t \sim WN(0, \sigma^2)$.

$y_t = 0.08 + \varepsilon_t + 0.4\, \varepsilon_{t-1}$ - MA(1) process

$y_t = 0.13\, y_{t-1} + \varepsilon_t$ - AR(1) process

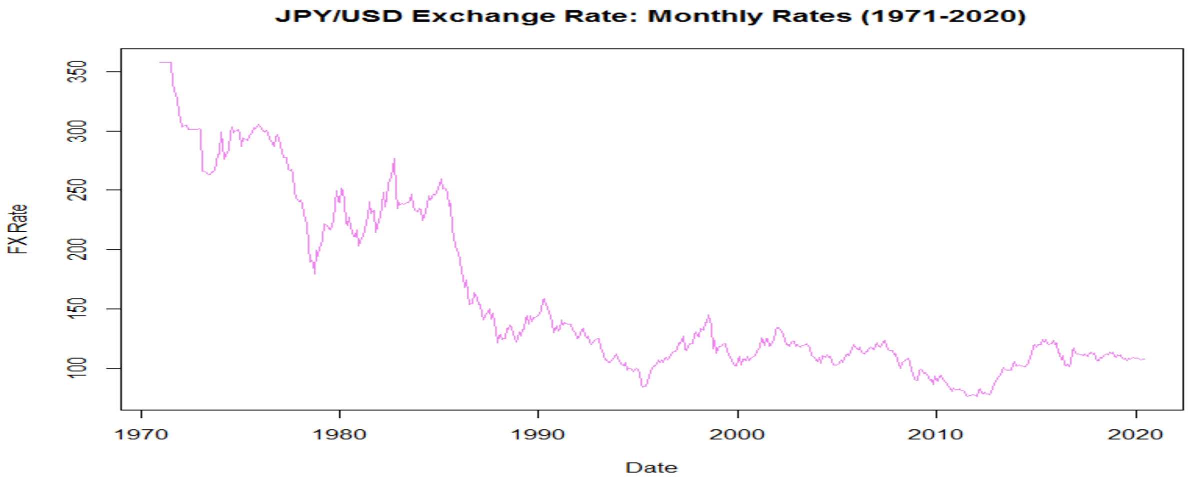


Non-Stationary Series – Examples

Examples: Assume $\varepsilon_t \sim$ WN(0, σ^2).

$y_t = \mu\ t + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \varepsilon_t$ - AR(2) with deterministic trend

$y_t = \mu + y_{t-1} + \varepsilon_t$ - Random Walk with drift



Time Series – Stationarity: Remarks

The main characteristic of time series is that observations are dependent.

To analyze time series, however, we need to assume that some features of the series are not changing. If we have non-stationary series (say, mean or variance are changing with each observation), it is not possible to make inferences.

Stationarity is an invariant property: the statistical characteristics of the time series do not vary over time.

If IBM is weak stationary, then, the returns of IBM may change month to month or year to year, but the average return and the variance in two equal lengths time intervals will be more or less the same.

In the long run, say 100-200 years, the stationarity assumption may not be realistic. After all, technological change has affected the return of IBM over the long run. But, in the short-run, stationarity seems likely to hold.

In general, time series analysis is done under the stationarity assumption.

Time Series – Ergodicity

We want to estimate the mean of the process {Z_t}, $\mu(Z_t)$. But, we need to distinguishing between *ensemble average* (with *m* observations) and *time average* (with *T* observations):

- Ensemble Average: $\bar{\bar{z}} = \frac{\sum_{i=1}^{m} Z_i}{m}$

- Time Series Average: $\overline{z} = \frac{\sum_{t=1}^{T} z_t}{T}$

Question: Which estimator is the most appropriate?
A: Ensemble Average. But, it is impossible to calculate. We only observe one Z_t, with dependent observations.

Question: Under which circumstances we can use the time average (with only one realization of $\{Z_t\}$)? Is the time average an unbiased and consistent estimator of the mean?
The *Ergodic Theorem* gives us the answer.

• Intuition behind Ergodicity:

We go to a casino to play a game with 20% return, but on average, one gambler out of 100 goes bankrupt. If 100 gamblers play the game, there is a 99% chance of winning and getting a 20% return. This is the *ensemble scenario.* Suppose that gambler 35 is the one that goes bankrupt. Gambler 36 is not affected by the bankruptcy of gamble 35.

Suppose now that instead of 100 gamblers you play the game 100 times. This is the *time series* scenario. You win 20% every day until day 35 when you go bankrupt. There is no day 36 for you (dependence at work!).

<u>Result</u>: The probability of success from the group (ensemble scenario) does not apply to one person (time series scenario).

Ergodicity describes a situation where the ensemble scenario outcome applies to the time series scenario.

• With dependent observation, we cannot use the LLN used before. The *ergodicity theorem* plays the role of the LLN with dependent observations.

The formal definition of ergodicity is complex and is seldom used in time series analysis. One consequence of ergodicity is the ergodic theorem, which is extremely useful in time series.

It states that if Z_t is an ergodic stochastic process then

$$\frac{1}{T}\sum_{t=1} g(Z_t) \overset{a.s}{\rightarrow} \text{E}[g(Z)]$$

for any function g(.). And, for any time shift k

$$\frac{1}{T}\sum_{t=1} g(Z_{t_1+k}, Z_{t_2+k}, \dots, Z_{t_\tau+k}) \overset{a.s}{\rightarrow} \text{E}[g(Z_{t_1}, Z_{t_2}, \dots, Z_{t_\tau}))]$$

where a.s. means *almost sure convergence*, a strong form of convergence.

Definition: A covariance-stationary process is *ergodic* for the **mean** if

$$\bar{z} \overset{p}{\rightarrow} \text{E}[Z_t] = \mu$$

This result needs the variance of $\bar{z}$ to collapse to 0. It can be shown that the var$[\overline{z}]$ can be written as a function of the autocorrelations, ρ_k:

$$\text{var}[\,\bar{z}\,] = \text{var}[\,(z_1 + z_2 + \cdots + z_T)/T] = \tfrac{\gamma_0}{T}\textstyle\sum_k (1 - \tfrac{|k|}{T})\rho_k$$

Theorem: A sufficient condition for ergodicity for the mean is that the autocorrelations ρ_k between two observations, say (y_{t_i}, y_{t_j}), $\rho\big(t_i, t_j\big) = \rho_{t_i - t_j}$, go to zero as t_i & t_j grow further apart.

Condition for ergodicity: $\rho_k \rightarrow 0$, as $k \rightarrow \infty$

Time Series – Lag Operator

Define the operator L as

$$L^k\, z_t = z_{t-k}.$$

It is usually called *Lag operator.* But it can produce lagged or forward variables (for negative values of k). For example:

$$L^{-3}\, z_t = z_{t+3}.$$

Also note that if c is a constant $\Rightarrow L\, c = c.$

Sometimes the notation for L when working as a lag operator is B *(backshift operator),* and when working as a forward operator is F.

Important application: Differencing

$$\Delta\, z_t = (1 - L)\, z_t = z_t - z_{t-1}.$$
$$\Delta^d\, z_t = (1 - L)^d\, z_t = z_{t-k}.$$

Time Series – Useful Result: Geometric Series

The function $f(x) = (1 - x)^{-1}$ can be written as an infinite geometric series (use a Maclaurin series around c=0):

$$f(x) = \tfrac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \ldots = \textstyle\sum_{n=0}^{\infty} x^n$$

If we multiply $f(x)$ by a constant, a:

$$\textstyle\sum_{n=0}^{\infty} a x^n = \tfrac{a}{1-x} \rightarrow \quad \sum_{n=1}^{\infty} a x^n = a\left(\tfrac{1}{1-x} - 1\right)$$

We will use this result when, under certain conditions, we invert a lag polynomial (say, $\theta(L)$) to convert an AR (MA) process into an infinite MA (AR) process.

Example: Suppose we have an MA(1) process:

$$y_t = \mu + \theta_1\, \varepsilon_{t-1} + \varepsilon_t = \mu + \theta(L)\, \varepsilon_t,$$

with

$$\theta(L) = (1 + \theta_1\, L)$$

Recall,

$$f(x) = \frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \ldots = \sum_{n=0}^{\infty} x^n$$

Let $x = -\theta_1 L$. Then,

$$\theta(L)^{-1} = \frac{1}{1-(-\theta_1 L)} = 1 + (-\theta_1 L) + (-\theta_1 L)^2 + (-\theta_1 L)^3 + (-\theta_1 L)^4 + \ldots$$

$$= \sum_{n=0}^{\infty} (-\theta_1 L)^n = 1 - \theta_1 L + \theta_1^2 L^2 - \theta_1^3 L^3 + \theta_1^4 L^4 + \cdots$$

That is, we get an AR(∞), by multiplying both sides by $\theta(L)^{-1}$:

$$\theta(L)^{-1}\, y_t = \theta(L)^{-1}\, \mu + \varepsilon_t = \mu^* + \varepsilon_t.$$

Or

$$\theta(L)^{-1}\, y_t = y_t - \theta_1 y_{t-1} + \theta_1^2 y_{t-}2 - \theta_1^3 y_{t-}3 + \theta_1^4 y_{t-}4 + \cdots = \mu^* + \varepsilon_t.$$

Solving for y_t:

$$y_t = \mu^* + \theta_1 y_{t-1} - \theta_1^2 y_{t-}2 + \theta_1^3 y_{t-}3 - \theta_1^4 y_{t-}4 + \cdots + \varepsilon_t.$$

Moving Average Process

An MA process models $E[y_t|I_{t-1}]$ with lagged error terms. An MA(q) model involves q lags. We keep the white noise assumption for ε_t: $\varepsilon_t \sim WN(0, \sigma^2)$

Example: A linear MA(q) model:

$$y_t = \mu + \theta_1\, \varepsilon_{t-1} + \theta_2\, \varepsilon_{t-2} + \ldots + \theta_q\, \varepsilon_{t-q} + \varepsilon_t = \mu + \theta(L)\, \varepsilon_t,$$

where

$$\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \theta_3 L^3 + \ldots + \theta_q L^q$$

In time series, the constant does not affect the properties of AR and MA process. It is usually removed (think of the data analyze as demeaned). Thus, in this situation we say "without loss of generalization", we assume μ=0.

Moving Average Process – Stationarity

To check if an MA(q) process is stationary, we check the moments (assume $\mu = 0$).

$$y_t = \varepsilon_t + \theta_1\, \varepsilon_{t-1} + \theta_2\, \varepsilon_{t-2} + \theta_3\, \varepsilon_{t-3} + \ldots + \theta_q\, \varepsilon_{t-q}$$

$$y_{t-1} = \varepsilon_{t-1} + \theta_1\, \varepsilon_{t-2} + \theta_2\, \varepsilon_{t-3} + \ldots + \theta_{q-1}\, \varepsilon_{t-q} + \theta_q\, \varepsilon_{t-(q+1)}$$

- **Mean**

$$E[y_t] = \mu + \theta_1\, E[\varepsilon_{t-1}] + \theta_2\, E[\varepsilon_{t-2}] + \ldots + \theta_q\, E[\varepsilon_{t-q}] + E[\varepsilon_t] = \mu = 0$$

- **Variance**

$$Var[y_t] = \theta_1^2\, Var[\varepsilon_{t-1}] + \theta_2^2\, Var[\varepsilon_{t-2}] + \ldots + \theta_q^2\, Var[\varepsilon_{t-q}] + Var[\varepsilon_t]$$

$$= (1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2)\, \sigma^2.$$

To get a positive variance, we require $(1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2) > 0$.

- **Autocovariances**

$$y_t = \varepsilon_t + \theta_1\, \varepsilon_{t-1} + \theta_2\, \varepsilon_{t-2} + \theta_3\, \varepsilon_{t-3} + \ldots + \theta_q\, \varepsilon_{t-q}$$

$$y_{t-1} = \varepsilon_{t-1} + \theta_1\, \varepsilon_{t-2} + \theta_2\, \varepsilon_{t-3} + \ldots + \theta_{q-1}\, \varepsilon_{t-q} + \theta_q\, \varepsilon_{t-(q+1)}$$

$$\gamma(1) = Cov[y_t, y_{t-1}] = E[y_t\, y_{t-1}]$$

$$= E[(\varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \ldots + \theta_q\varepsilon_{t-q}) * (\varepsilon_{t-1} + \theta_1\varepsilon_{t-2} + \theta_2\varepsilon_{t-3} + \ldots + \theta_q\varepsilon_{t-(q+1)})]$$

$$= E[\varepsilon_t\, \varepsilon_{t-1}] + \theta_1\, E[\varepsilon_t\, \varepsilon_{t-2}] + \ldots. + \theta_1\, E[\varepsilon_{t-1}\, \varepsilon_{t-1}] + \theta_1^2\, E[\varepsilon_{t-1}\, \varepsilon_{t-1}] + \theta_2\, E[\varepsilon_{t-2}\, \varepsilon_{t-1}] + \theta_1\theta_2\, E[\varepsilon_{t-2}\, \varepsilon_{t-2}]$$

$$+ \theta_q E[\varepsilon_{t-q}\, \varepsilon_{t-1}] + \theta_q\theta_1 E[\varepsilon_{t-q}\, \varepsilon_{t-2}] + \ldots + \theta_q\, \theta_{q-1} E[\varepsilon_{t-q}\, \varepsilon_{t-q}] + \theta_q^2 E[\varepsilon_{t-q}\, \varepsilon_{t-(q+1)}]$$

$$= E[y_t\, \varepsilon_{t-1}] + \theta_1 E[y_t\, \varepsilon_{t-2}] + \theta_2 E[y_t\, \varepsilon_{t-3}] + \ldots + \theta_q E[y_t\, \varepsilon_{t-q-1}]$$

$$= \theta_1\, \sigma^2 + \theta_2\, \theta_1\, \sigma^2 + \theta_3\, \theta_2\, \sigma^2 + \ldots + \theta_q\, \theta_{q-1}\, \sigma^2 + 0$$

$$= \sigma^2 \sum_{j=1}^{q} \theta_j\, \theta_{j-1} \qquad \text{(where } \theta_0=1)$$

We continue with the derivations of the $\gamma(k)$ function. It is easier to derive it by rewriting y_t & y_{t-2} :

$$y_t = \varepsilon_t + \theta_1\, \varepsilon_{t-1} + \theta_2\, \varepsilon_{t-2} + \theta_3\, \varepsilon_{t-3} + \ldots + \theta_q\, \varepsilon_{t-q}$$

$$y_{t-2} = \varepsilon_{t-2} + \theta_1\, \varepsilon_{t-3} + \theta_2\, \varepsilon_{t-4} + \theta_3\, \varepsilon_{t-5} + \ldots + \theta_q\, \varepsilon_{t-(q+2)}$$

$$\gamma(2) = \text{Cov}[y_t, y_{t-2}] = E[y_t\, y_{t-2}]$$

$$= E[(\varepsilon_t+\theta_1\varepsilon_{t-1}+\theta_2\varepsilon_{t-2}+\ldots+\theta_q\varepsilon_{t-q}+)*(\varepsilon_{t-2}+\theta_1\varepsilon_{t-3}+\theta_2\varepsilon_{t-4}+\ldots+\theta_q\varepsilon_{t-q-2})]$$

$$= E[y_t\, \varepsilon_{t-2}] + \theta_1 E[y_t\, \varepsilon_{t-3}] + \theta_2 E[y_t\, \varepsilon_{t-4}] + \ldots + \theta_q E[y_t\, \varepsilon_{t-q-2}]$$

$$= \theta_2\, \sigma^2 + \theta_3\, \theta_1\, \sigma^2 + \theta_4\, \theta_2\, \sigma^2 + \ldots + \theta_q\, \theta_{q-2}\, \sigma^2 + 0$$

$$= \sigma^2 \sum_{j=2}^{q} \theta_j\, \theta_{j-2} \qquad \text{(where } \theta_0=1)$$

⋮

$$\gamma(q) = E[y_t\, y_{t-q}] =$$

$$= E[\varepsilon_t y_{t-q}] + \theta_1 E[\varepsilon_{t-1} y_{t-q}] + \theta_2 E[\varepsilon_{t-2} y_{t-q}] + \ldots + \theta_q E[\varepsilon_{t-q} y_{t-q}]$$

$$= \theta_q\, \sigma^2$$

$$= \sigma^2 \sum_{j=q}^{q} \theta_j\theta_{j-q} \qquad \text{(where } \theta_0=1)$$

In general, for the *k autocovariance:*

$$\gamma(k) = \sigma^2 \sum_{j=k}^{q} \theta_j\theta_{j-k} \qquad \text{for } |k| \leq q$$

$$\gamma(k) = 0 \qquad \text{for } |k| > q$$

Remark: After lag q , the autocovariance are 0.

• Autocorrelations

From the autocovariances, we define the autocorrelations, by dividing the autocorrelations by $\gamma(0)$:

$$\rho(q) = \frac{\sigma^2 \sum_{j=q}^{q} \theta_j\theta_{j-q}}{(1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2)\, \sigma^2} = \frac{\sum_{j=q}^{q} \theta_j\theta_{j-q}}{(1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2)} \qquad (\theta_0=1)$$

In general, for the *k autocorrelation function* (ACF):

$$\rho(k) = \frac{\sum_{j=q}^{q} \theta_j\theta_{j-q}}{(1 + \theta_1^2 + \theta_2^2 + \ldots + \theta_q^2)} \qquad \text{for } |k| \leq q$$

$$\rho(k) = 0 \qquad \text{for } |k| > q$$

Remark: After lag q , the ACF are 0.

- It can be shown that for ε_t with same distribution (say, normal) the ACF are non-unique. For example, for the MA(1) processes:

$$y_t = \varepsilon_t + 0.5\, \varepsilon_{t-1} \Rightarrow \rho(1) = \theta_1/(1+\theta_1^2) = 0.4$$

$$y_t = \varepsilon_t + 2\, \varepsilon_{t-1} \Rightarrow \rho(1) = \theta_1/(1+\theta_1^2) = 0.4$$

- It is easy to verify that the sums $\sum_{j=k}^{q} \theta_j\theta_{j-k}$ are finite. Then, mean, variance and covariance are constant.

$\Rightarrow$ MA(q) is always stationary.

Moving Average Process – Invertibility

As mentioned above, it is possible that different time-series processes produce the same ACF.

Example: The two MA(1) produce the same $\gamma(k)$:

$y_t = \varepsilon_t + .20\, \varepsilon_{t-1}$, $\quad \varepsilon_t \sim i.i.d.\ N(0, 25)$
$z_t = \upsilon_t + 5\, \upsilon_{t-1}$, $\quad \upsilon_t \sim i.i.d.\ N(0; 1)$

We only observe the time series, y_t or z_t , and not the noise, ε_t or υ, thus, we cannot distinguish between the models. Thus, we select only one of them. Which one? We select the model with an AR(∞) representation.
Assuming $\theta(L) \neq 1$, we can invert $\theta(L)$. Then, by inverting $\theta(L)$, an MA(*q*) process generates an AR process:

$$y_t = \mu + \theta(L)\, \varepsilon_t \qquad \Rightarrow \theta(L)^{-1}\, y_t = \Pi(L)\, y_t = \mu^* + \varepsilon_t,$$

Then, we have an infinite sum polynomial on θL. (Recall the geometric series result.) That is, we convert an MA(*q*) into an AR(∞):

$$\sum_{j=0}^{\infty} \pi_j(L) y_t = \mu^* + \varepsilon_t$$

We need to make sure that $\Pi(L) = \theta(L)^{-1}$ is defined: We require $\theta(L) \neq 0$. When this condition is met, we can write ε_t as a causal function of y_t. We say the MA is *invertible.* For this to hold, we require:

$$\sum_{j=0}^{\infty} |\pi_j(L)| < \infty$$

Technical note: An invertible MA(q) is typically required to have roots of the lag polinomial equation $\theta(z) = 0$ greater than one in absolute value ("*outside the unit circle*"). In the MA(1) case, we require$|\theta_1|< 1$.
In the previous example, we select the model with $\theta_1 = .20$.

Moving Average Process – MA(1)

Example: $\quad y_t = \theta_1\, \varepsilon_{t-1} + \varepsilon_t = \mu + \theta(L)\, \varepsilon_t$, $\quad$ with $\theta(L) = (1+ \theta_1 L)$

• Moments

$$E[y_t] = 0$$
$$Var[y_t] = \gamma(0) = \sigma^2 + \theta_1^2\, \sigma^2 = \sigma^2\, (1+ \theta_1^2)$$
$$Cov[y_t, y_{t-1}] = \gamma(1) = E[y_t\, y_{t-1}] = E[(\theta_1\, \varepsilon_{t-1} + \varepsilon_t)(\theta_1\, \varepsilon_{t-2} + \varepsilon_{t-1})]$$
$$= \theta_1\, \sigma^2$$

⋮

$$\gamma(k) = E[y_t\, y_{t-k}] = E[(\theta_1\, \varepsilon_{t-1} + \varepsilon_t)(\theta_1\, \varepsilon_{t-(k+1)} + \varepsilon_{t-k})] = 0 \qquad \text{(for } k>1)$$

That is, for $|k| > 1$, $\quad \gamma(k) = 0$.

To get the ACF, we divide the autocovariances by $\gamma(0)$. Then, the autocorrelation function (ACF):

$$\rho(1) = \gamma(1)/\gamma(0) = \theta_1\, \sigma^2 /\sigma^2\, (1+ \theta_1^2) = \theta_1 / (1+ \theta_1^2)$$

⋮

$$\rho(k) = \gamma(k)/\gamma(0) = 0 \qquad \text{(for } k > 1)$$

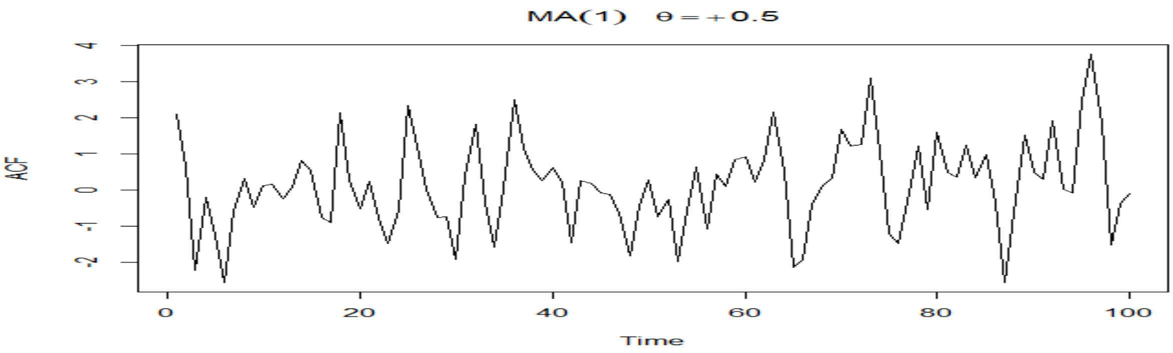
Remark: The autocovariance function is zero after lag 1. Similarly, the ACF is also zero after lag 1.
Note that $|\rho(1)| \leq 0.5$.
When $\theta_1 = 0.5 \qquad \Rightarrow \rho(1) = 0.4$.
$\theta_1 = -0.9 \qquad \Rightarrow \rho(1) = -0.497238$.
$\theta_1 = -2 \qquad \Rightarrow \rho(1) = -0.4$. $\qquad$ (same $\rho(1)$ for θ_1 & $1/\theta_1$)

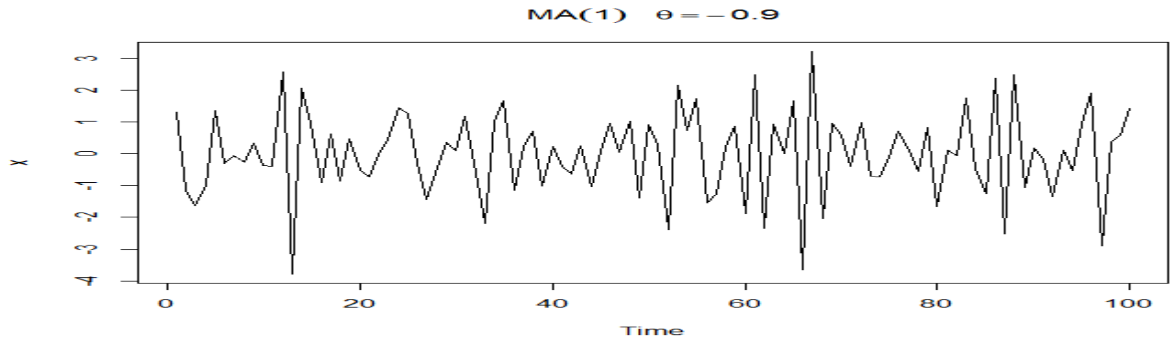
- We simulate and plot three MA(1) processes, with standard normal ε_t -i.e., $\sigma=1$:

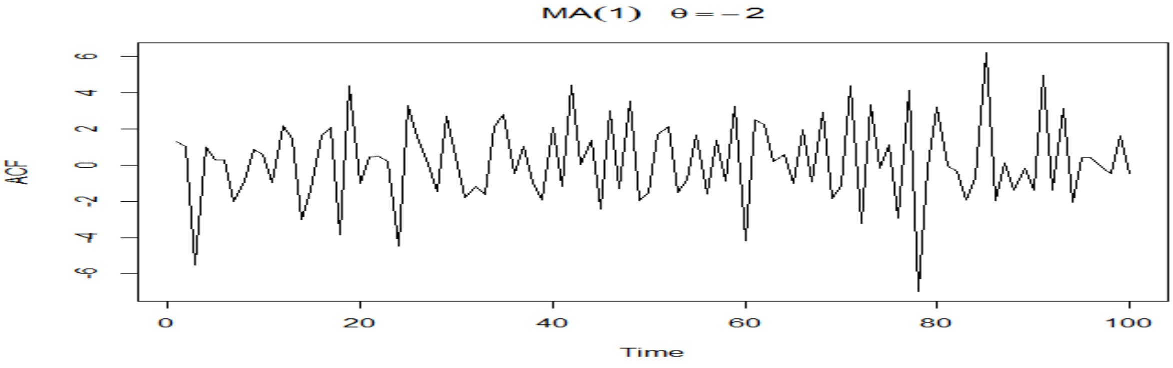
$$y_t = \varepsilon_t + 0.5\, \varepsilon_{t-1}$$
$$y_t = \varepsilon_t - 0.9\, \varepsilon_{t-1}$$
$$y_t = \varepsilon_t - 2\, \varepsilon_{t-1}$$

R script to plot $y_t = \varepsilon_t + \mathbf{0.5}\, \varepsilon_{t-1}$ with 100 simulations

```
> plot(arima.sim(list(order=c(0,0,1), ma=0.5), n=100), ylab="ACF",
main=(expression(MA(1)~~~theta==+.5)))
```







Note: The process $\theta_1 > 0$ is smoother than the ones with $\theta_1 < 0$.

- Invertibility: If $|\theta_1|<1$, we can write $(1+ \theta_1 L)^{-1}\, y_t + \mu^* = \varepsilon_t$

Or expanding

$$(1 - \theta_1 L + \theta_1{}^2 L^2 + \ldots + \theta_1{}^j L^j + \ldots)\, y_t + \mu^* = \mu^* + \sum_{i=1}^{\infty} \pi_i(L)\, y_t = \varepsilon_t$$

That is, $\pi_i = (-\theta_1)^i$.
The simulated process with $\theta_1 = -2$ is non-invertible, the infinite sum of π_i would explode. We would select the MA(1) with $\theta_1 = -.5$.

Moving Average Process – MA(2)

Example: $y_t = \mu + \theta_2\, \varepsilon_{t-2} + \theta_1\, \varepsilon_{t-1} + \varepsilon_t = \mu + \theta(L)\, \varepsilon_t$,
with

$\theta(L) = (1 + \theta_1 L + \theta_2 L^2)$.

• Moments

$$E\left(Y_t\right) = \mu$$

$$\gamma_k = \begin{cases} \sigma^2\left(1 + \theta_1^2 + \theta_2^2\right), & k = 0 \\ -\theta_1\sigma^2\left(1 - \theta_2\right), & |k| = 1 \\ -\theta_2\sigma^2, & |k| = 2 \\ 0, & |k| > 2 \end{cases}$$

Remark: The autocovariance function is zero after lag 2. Similarly, the ACF is also zero after lag 2.

– Invertibility: The roots of $\lambda^2 - \theta_1 \lambda - \theta_2 = 0$ all lie inside the unit circle. It can be shown the invertibility condition for an MA(2) process is:

$\theta_1 + \theta_2 < 1$
$\theta_1 - \theta_2 < 1$
$-1 < \theta_2 < 1$.

Moving Average Process – Estimation

MA processes are more complicated to estimate. In particular, there are nonlinearities. Consider an MA(1):

$$y_t = \varepsilon_t + \theta\, \varepsilon_{t-1}$$

We cannot do OLS, since we do not observe ε_{t-1}. But, based on the ACF, we can do the estimation.

• The auto-correlation is $\rho_1 = \theta/(1+\theta^2)$. Then, we can use the method of moments (MM), which uses the estimated ρ_1, r_1, to estimate of θ:

$$r_1 = \frac{\hat{\theta}}{(1+\hat{\theta}^2)} \quad \Rightarrow \quad \hat{\theta} = \frac{1 \pm \sqrt{1-4r_1^2}}{2r_1}$$

A nonlinear solution and difficult to solve.

• Alternatively, if $|\theta| < 1$, we can invert the MA(1) process. Then, based on the AR representation, we can try finding $a \in (-1; 1)$,

$$\varepsilon_t\,(a) = y_t + a y_{t-1} + a^2\, y_{t-2} + \ \ldots$$

and look (numerically) for the least-square estimator

$$\hat{\theta} = \arg_a \min\{S_T(a) = \sum_{t=1}^{T} \varepsilon_t^2(a)\}$$

The Wold Decomposition

Theorem - Wold (1938).

Any covariance stationary $\{y_t\}$ has infinite order, moving-average representation:

$$y_t = S_t + \kappa_t,$$

where

κ_t is a deterministic term –i.e., completely predictable. For example, κ_t= μ or a linear combination of past (known) values of κ_t.

$S_t = \sum_{j=0}^{\infty} \psi_j \, \varepsilon_{t-j}$ (= $\psi(L)\varepsilon_t$, with $\psi(L)$ = infinite lag polynomial)

$\sum_{j=0}^{\infty} \psi_j^2 < \infty$ (assumption for the stability of polynomial, "*square summability*")

ψ_j only depend on j (weights of innovations are not time dependent)

$\psi_0 = 1$ (a convenient assumption)

$\varepsilon_t \sim \text{WN}(0, \sigma^2)$ (ε_t independent and uncorrelated with S_t)

Thus, y_t is a linear combination of innovations over time plus a deterministic part.

- A stationary process can be decomposed into a sum of two parts, one represented as an MA(∞) and the other a deterministic "trend."

Example: Let $x_t = y_t - \kappa_t$. (x_t = MA(∞) part) Then, check moments:

$$E[x_t] = E[y_t - \kappa_t] = \sum_{j=0}^{\infty} \psi_j E[\varepsilon_{t-j}] = 0.$$

$$E[x_t^2] = \sum_{j=0}^{\infty} \psi_j^2 E[\varepsilon_{t-j}^2] = \sigma^2 \sum_{j=0}^{\infty} \psi_j^2 < \infty.$$

$$E[x_t, x_{t-j}] = E[(\varepsilon_t + \psi_1 \varepsilon_{t-1} + \psi_2 \varepsilon_{t-2} + \ldots)(\varepsilon_{t-j} + \psi_1 \varepsilon_{t-j-1} + \psi_2 \varepsilon_{t-j-2} + \ldots)$$

$$= \sigma^2 (\psi_j + \psi_1 \psi_{j+1} + \psi_2 \psi_{j+2} + \ldots) = \sigma^2 \sum_{k=0}^{\infty} \psi_k \psi_{k+2}$$

X_t is a covariance stationary process.

Remark: This old theorem is the backbone of time series analysis. We will approximate the Wold infinite lag polynomial $\psi(L)$ with a ratio of two finite lag polynomials. This approximation is the basis of ARMA modeling.

Autoregressive (AR) Process

We model the conditional expectation of y_t, $E[y_t|I_{t-1}]$, as a function of its past history. We assume ε_t follows a WN(0, σ^2).

The most common models are AR models. An AR(1) model involves a single lag, while an AR(*p*) model involves *p* lags. Then, the AR(*p*) process is given by:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

Using the lag operator we write the AR(*p*) process:

$$\phi(L)\, y_t = \varepsilon_t$$

with

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \ldots - \phi_p L^p$$

- We can look at an AR(*p*) process:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \varepsilon_t,$$

as a *stochastic (linear) difference equation* (SDE). With difference equations we try to get a solution –i.e., given some initial conditions/history, we know the value of y_t for any t– and, then, we study its characteristics (stability, long-run value, etc.).

The solution to a SDE can be written as a sum of two solutions:

1) the homogeneous equation (the part that only depends on the y_t's):

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} \qquad (\text{set } \mu + \varepsilon_t = 0)$$

2) A particular solution to the equation.

Once we get a solution, we study its stability. We want a stable one.

• We get a solution to the simple case, the AR(1) process.

$$y_t = \mu + \phi_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

We use the backward substitution method:

$$\begin{aligned} y_t &= \mu + \phi_1 (\mu + \phi_1 y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &= \mu (1 + \phi_1) + \phi_1^2 y_{t-2} + \varepsilon_t + \phi_1 \varepsilon_{t-1} \\ &= \mu (1 + \phi_1) + \phi_1^2 (\mu + \phi_1 y_{t-3} + \varepsilon_{t-2}) + \varepsilon_t + \phi_1 \varepsilon_{t-1} \\ &= \mu (1 + \phi_1 + \phi_1^2) + \phi_1^3 y_{t-3} + \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} \end{aligned}$$

⋮

$$\Rightarrow y_t = \mu (1 + \phi_1 + \phi_1^2 + ... + \phi_1^{t-1}) + \sum_{j=0}^{t-1} \phi_1^j \varepsilon_{t-j} + \phi_1^t y_0$$

The solution is a function of t, the whole sequence ε_t, ε_{t-1}, ..., ε_1 and the initial condition y_0. The effect of y_0 "dies out" if $|\phi_1|< 1$.

• The stability of the solution is crucial. With a stable solution, Y_t does not explode. This is good: We need well defined moments.

It turns out that the stability of the equation depends on the solution to the homogenous equation. In the AR(1) case:

$$y_t = \phi_1 y_{t-1}$$

with solution $y_t = \phi_1^t y_0$

If$|\phi_1|< 1$, y_t never explodes, as as t → ∞. In this case, in the solution to the AR(1) process, the effect of y_0 "dies out" as t → ∞.

• We can analyze the stability from the point of view of the roots of the lag polynomial. For the AR(1) process

$$\phi(z) = 1 - \phi_1 z = 0 \qquad \Rightarrow |z| = 1/|\phi_1| > 1$$

That is, the AR(1) process is stable if the root of $\phi(z)$ is greater than one (also said as "*the roots lie outside the unit circle*").

This result generalizes to AR(p) process. For example, for the AR(2) process

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t,$$

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 \qquad \Rightarrow \text{both roots, } z_1 \text{ \& } z_2\text{, should lie outside the } \textit{unit circle}.$$

For an AR*(p),* we need the roots of $\phi(z)$ to be outside the unit circle.

• For the AR(2), $y_t = \phi_1 y_{t-1} - \phi_2 y_{t-2}$

We need the roots of $\phi(z)$ to be outside the unit circle.

The characteristic polynomial of the AR(2) can be written as:

$$\phi(z) = 1 - (\lambda_1 + \lambda_2) z - \lambda_1 \lambda_2 z^2 = (1 - \lambda_1 z)(1 - \lambda_2 z) = 0$$

where $\phi_1 = \lambda_1 + \lambda_2$, and $\phi_2 = \lambda_1 \lambda_2$. ($\lambda_1$ & λ_2 are *eigenvalues*.)

If $|\lambda_1|<1$, and $|\lambda_2|<1$, the roots lie *outside the unit root* ⇒ stationary.

Then, some implications for ϕ_1 & ϕ_2:

$$|\lambda_1 + \lambda_2| < 2 \qquad \Rightarrow |\phi_1| < 2$$

$$|\lambda_1 \lambda_2| < 1 \qquad \Rightarrow |\phi_2| < 1$$

• Summary:

We say the process is globally (asymptotically) stable if the solution of the associated homogenous equation tends to 0, as t → ∞.

Theorem

A necessary and sufficient condition for global asymptotical stability of a p^{th} order deterministic difference equation with constant coefficients is that all roots of the associated lag polynomial equation $\phi(z)$=0 have *moduli* strictly more than 1.

For the case of real roots, moduli means "absolute values."

AR(1) Process – Stationarity & ACF

An AR(1) model:

$$y_t = \phi_1 \, y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

Recall that in a previous example, under the stationarity condition $|\phi_1| < 1$, we derived the moments:

$E[y_t] = \mu = 0$ (assuming $\phi_1 \neq 1$)

$Var[y_t] = \gamma(0) = \sigma^2/(1 - \phi_1^2)$ (assuming $|\phi_1| < 1$)

$\gamma(1) = E[y_t \, y_{t-2}] = E[(\phi_1 \, y_{t-1} + \varepsilon_t) * y_{t-1}] = \phi_1 \, \gamma(0)$

$\gamma(2) = E[y_t \, y_{t-2}] = E[(\phi_1 \, y_{t-1} + \varepsilon_t) * y_{t-2}]$

$\qquad = \phi_1 \, E[y_{t-1} \, y_{t-2}] = \phi_1 \, \gamma(1) = \phi_1^2 \, \gamma(0)$

$\gamma(3) = E[y_t \, y_{t-3}] = E[(\phi_1 \, y_{t-1} + \varepsilon_t) * y_{t-3}]$

$\qquad = \phi_1 \, E[y_{t-1} \, y_{t-3}] = \phi_1 \, \gamma(2) = \phi_1^3 \, \gamma(0)$

⋮

$\gamma(k) = \phi_1 \, \gamma(k-1) = \phi_1^k \, \gamma(0)$

Now, we derive the autocorrelation: $\rho(t_1, t_2) = \frac{\gamma(t_1 - t_2)}{\sigma_{t_1}\sigma_{t_2}}$

If the process is stationary ($\sigma_t = \sigma_{t-1} = \sqrt{\gamma(0)}$)

$\rho(1) = \rho(t,\, t-1) = \frac{\gamma(1)}{\sigma_t \sigma_{t-1}} = \frac{\gamma(1)}{\gamma(0)} = \phi_1$

$\rho(2) = \frac{\gamma(2)}{\gamma(0)} = \phi_1^2$

⋮

$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \phi_1^k$

Remark: The ACF decays with *k*.

When we plot $\rho(k)$ against *k*, we plot also $\rho(0)$ which is 1.

Note that when $\phi_1 = 1$, the AR(1) is non-stationary, $\rho(k) = 1$, for all *k*. The present and the past are always correlated!

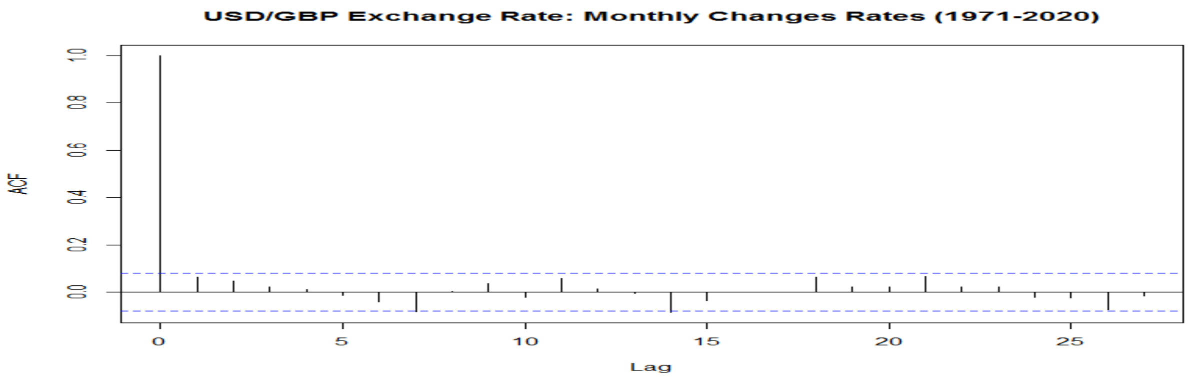
• Again, when $|\phi_1| < 1$, the autocorrelations do not explode as *k* increases. There is an exponential decay towards zero.

Note:

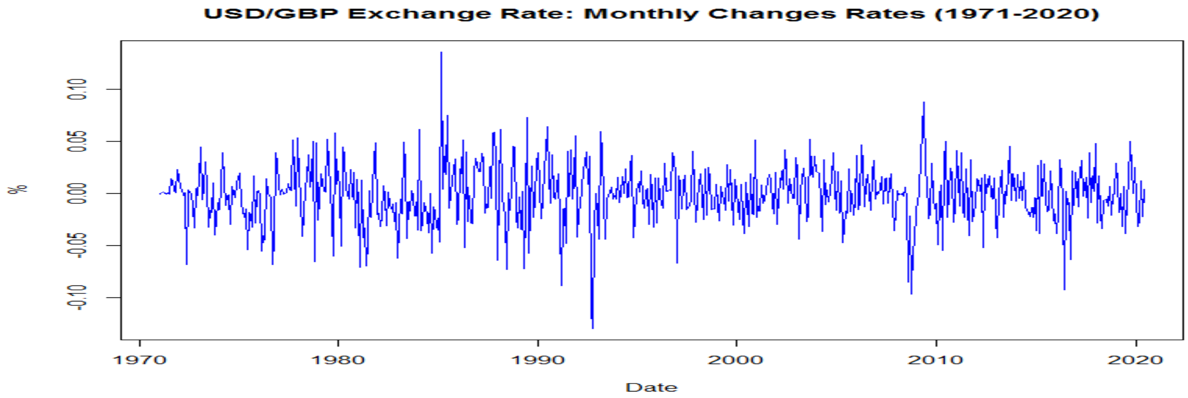
– when $0 < \phi_1 < 1$ ⇒ All autocorrelations are positive.

– when $-1 < \phi_1 < 0$ ⇒ The sign of $\rho(k)$ shows an alternating pattern beginning a negative value.

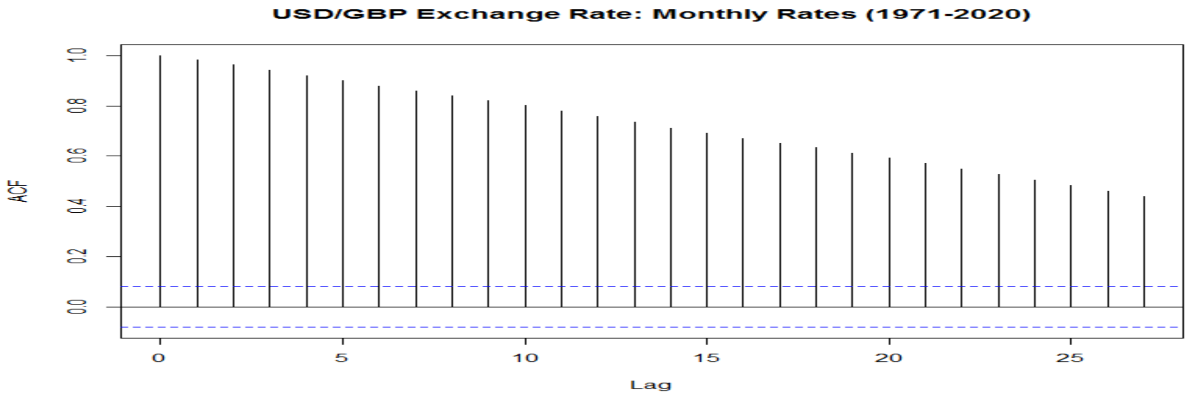
Example: A process with $|\phi_1| < 1$ (actually, 0.065) is the monthly changes in the USD/GBP exchange rate. Below we plot its corresponding ACF:



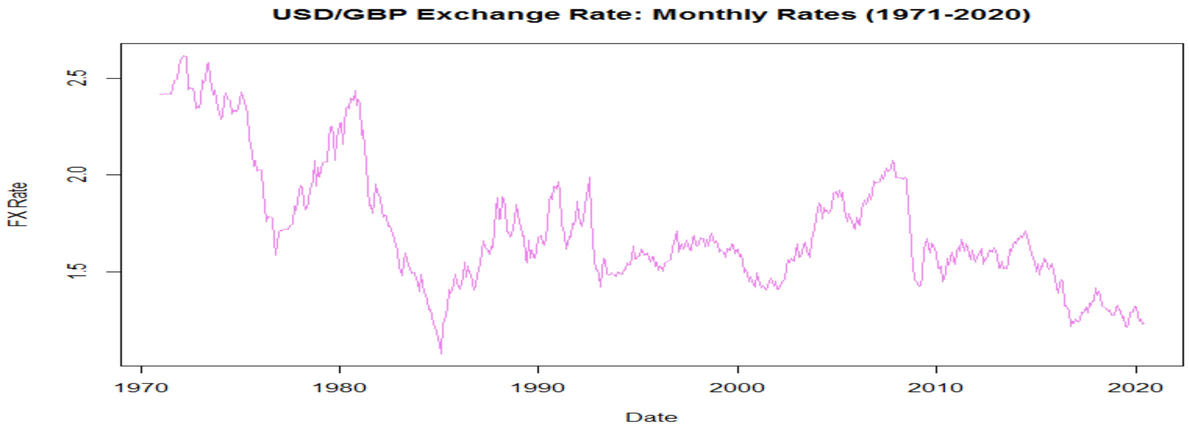
Below we plot the monthly changes in the USD/GBP exchange rate. Stationary series do not look smooth:



Example: A process with $\phi_1 \approx 1$ (actually, 0.99) is the nominal USD/GBP exchange rate. Below, we plot the ACF, it is not 1 all the time, but its decay is very slow (after 30 months, it is still .40 correlated!):



Below we plot the nominal USD/GBP exchange rate. Stationary series look smooth, smooth enough that you can clearly spot trends:



AR(1) Process – Stationarity & ACF

An AR(2) model:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

- **Moments:**

$$E[y_t] = \mu / /(1 - \phi_1 - \phi_2) = 0 \quad (\text{assuming } \phi_1 + \phi_2 \neq 1)$$
$$Var[y_t] = \sigma^2/(1 - \phi_1^2 - \phi_2^2) \quad (\text{assuming } \phi_1^2 + \phi_2^2 < 1)$$

- **Autocovariance function**

$$\gamma(k) = Cov[y_t, y_{t-k}] = E[(\phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t) y_{t-k}]$$
$$= \phi_1 E[y_{t-1} y_{t-k}] + \phi_2 E[y_{t-2} y_{t-k}] + E[\varepsilon_t y_{t-k}]$$
$$= \phi_1 \gamma(k-1) + \phi_2 \gamma(k-2) + E[\varepsilon_t y_{t-k}]$$

- We have a recursive formula:

$(k=0)$ $\gamma(0) = \phi_1 \gamma(-1) + \phi_2 \gamma(-2) + E[\varepsilon_t y_t]$
$= \phi_1 \gamma(1) + \phi_2 \gamma(2) + \sigma^2$

$(k=1)$ $\gamma(1) = \phi_1 \gamma(0) + \phi_2 \gamma(1) + E[\varepsilon_t y_{t-1}]$
$= \phi_1 \gamma(0) + \phi_2 \gamma(1)$
$\Rightarrow \gamma(1) = [\phi_1/(1 - \phi_2)]\, \gamma(0)$

$(k=2)$ $\gamma(2) = \phi_1 \gamma(1) + \phi_2 \gamma(0) + E[\varepsilon_t y_{t-2}]$
$= \phi_1 \gamma(1) + \phi_2 \gamma(0)$
$\Rightarrow \gamma(2) = [\phi_1^2 \gamma(0)/(1 - \phi_2)] + \phi_2 \gamma(0)$
$= [\phi_1^2/(1 - \phi_2) + \phi_2]\, \gamma(0)$

Replacing $\gamma(1)$ and $\gamma(2)$ back to $\gamma(0)$:

$$\gamma(0) = [\phi_1^2/(1 - \phi_2)]\, \gamma(0) + [\phi_2\, \phi_1^2/(1 - \phi_2) + \phi_2^2]\, \gamma(0) + \sigma^2$$

$$= \frac{\sigma^2(1 - \phi_2)}{(1 - \phi_2) - \phi_1^2(1 + \phi_2) + \phi_2^2(1 - \phi_2)} \qquad \Rightarrow |\phi_2|<1$$

- Dividing the previous formulas by $\gamma(0)$, we get the ACF:

$$\rho(k) = \gamma(k)/\gamma(0) = \phi_1 \rho(k-1) + \phi_2 \rho(k-2) + E[\varepsilon_t y_{t-k}]/\gamma(0)$$

$(k=0)$ $\rho(0) = 1$

$(k=1)$ $\rho(1) = \phi_1 /(1 - \phi_2)$

$(k=2)$ $\rho(2) = \phi_1 \rho(1) + \phi_2 \rho(0) = \phi_1^2/(1 - \phi_2) + \phi_2$

$(k=3)$ $\rho(3) = \phi_1 \rho(2) + \phi_2 \rho(1) =$

$= \phi_1^3/(1 - \phi_2) + \phi_1 \phi_2 + \phi_2\phi_1 /(1 - \phi_2)$

Remark: Again, we see exponential decay in the ACF.

From the work above, we need: $\phi_1 + \phi_2 \neq 1.$
$\phi_1^2 + \phi_2^2 < 1.$
$|\phi_2| < 1.$

AR(p) Process – VAR(1) Representation

With AR process with more lags than the AR(1) process, it is complicated to determine stationarity by looking at the ϕ_i's coefficients.

Stationarity conditions can be derived in a simplified way by rewriting an AR(p) as AR(1) process. For example, the AR(2) process:

$$y_t = \mu + \phi_1\, y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t \quad \Rightarrow (1 - \phi_1\, L - \phi_2 L^2) y_t = \mu + \varepsilon_t$$

can be written in matrix form as an AR(1):

$$\begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} \mu \\ 0 \end{bmatrix} + \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix} \;\Rightarrow\; \tilde{y}_t = \tilde{\mu} + A\tilde{y}_{t-1} + \tilde{\varepsilon}_t$$

- The AR(2) in matrix AR(1) form is called *Vector AR(*1*)* or VAR(1).
- We can derived a matrix lag polynomial A(L):

$$\tilde{y}_t = \tilde{\mu} + A\tilde{y}_{t-1} + \tilde{\varepsilon}_t \qquad \Rightarrow A(L)\tilde{y}_t = [I - AL]\, \tilde{y}_t = \tilde{\varepsilon}_t.$$

AR(2) Process – VAR(1) & Stationarity

If A(L) is invertible we can write an MA(∞) representation:

$$\tilde{y}_t = \tilde{\mu} + A\tilde{y}_{t-1} + \tilde{\varepsilon}_t \qquad \Rightarrow \tilde{y}_t = [I - AL]^{-1}\tilde{\varepsilon}_t$$

Note: Recall the expansion:

Checking that [**I** – **A***L*] is not singular, same as checking that $\mathbf{A}^j$ does not explode. The stability of the system (solution) can be determined by the *eigenvalues* of **A**. That is, get the λ_i's and check if $|\lambda_i|<1$ for all *i.*

$$\boldsymbol{A} = \begin{bmatrix} \phi_1 & \phi 2 \\ 1 & 0 \end{bmatrix} \Rightarrow |\boldsymbol{A} - \lambda I| = \det \begin{bmatrix} \phi 1 - \lambda & \phi 2 \\ 1 & -\lambda \end{bmatrix} = -(\phi_1 - \lambda)\lambda - \phi_2$$
$$= \phi_2 - \phi_1\, \lambda + \lambda^2$$

- Solution to quadratic equation: $\lambda_i = \frac{\phi_1 \pm \sqrt{\phi_1{}^2 - 4\phi_2}}{2}$

Stability and stationary: $|\lambda_i| < 1.$ ⇒ roots of $\phi(z)$ *outside unit circle.*

For the AR(2) process, we have already derived some relations between λ_i's and ϕ_i's:

$$\lambda_1\lambda_2 = \phi_2 \qquad \Rightarrow |\lambda_1\lambda_2| = |\phi 2| < 1$$
$$\lambda_1 + \lambda_2 = \phi_1 \quad \Rightarrow |\lambda_1 + \lambda_2| = |\phi 1\,| < 2$$

- We derived autocovariance function, γ(*k*), before, getting a recursive formula. Let's write the first autocovariances:

(*k*=0) $\gamma(0) = \phi_1\, \gamma(1) + \phi_2\, \gamma(2) + \sigma^2$
(*k*=1) $\gamma(1) = [\phi_1/(1 - \phi_2)]\, \gamma(0)$
(*k*=2) $\gamma(2) = [\phi_1^2/(1 - \phi_2) + \phi_2]\, \gamma(0)$

With$|\phi_2| < 1$, we get well defined γ(1), γ(2) & γ(0).

The VAR(1) has a nice property: The VAR(1) is Markov -i.e., forecasts depend only on today's data.

It looks complicated, but it is straightforward to apply the VAR formulation to any AR(*p)* processes. We can also use the same eigenvalue conditions to check the stationarity of AR(*p)* processes.

AR Process –Stationarity & Ergodicity

Theorem: The linear AR(p) process is strictly stationary and ergodic if and only if the roots of $\phi(L)$ are $|r_j|>1$ for all j, where $|r_j|$ is the modulus of the complex number r_j.

Note: If one of the r_j's equals 1, $\phi(L)$ (& y_t) has a unit root –i.e., $\phi(1)=0$. This is a special case of *non-stationarity*.

Recall $\phi(L)^{-1}$ produces an infinite sum on the ε_{t-j}'s. If this sum does not explode, we say the process is *stable*.

AR Process – Dynamic Multiplier & IRF

If the process is stable, we can calculate $\delta y_t/\delta\varepsilon_{t-j}$.

$\delta y_t/\delta\varepsilon_{t-j}$ = How much y_t is affected today by an innovation (ε) $t-j$ periods ago. When expressed as a function of *j,* we call this *dynamic multiplier*. Accumulated over time it is the *impulse response function (IRF).*

The *dynamic multiplier* measurers the effect of an innovation, ε_t, (economist like to call the ε_t 's, "*shocks*") on subsequent values of y_t: That is, the first derivative on the Wold representation:

$$\delta y_{t+j}/\delta\varepsilon_t = \delta y_j/\delta\varepsilon_0 = \psi_j.$$

For an AR(1) process:

$$\delta y_{t+j}/\delta\varepsilon_t = \delta y_j/\delta\varepsilon_0 = \phi^j.$$

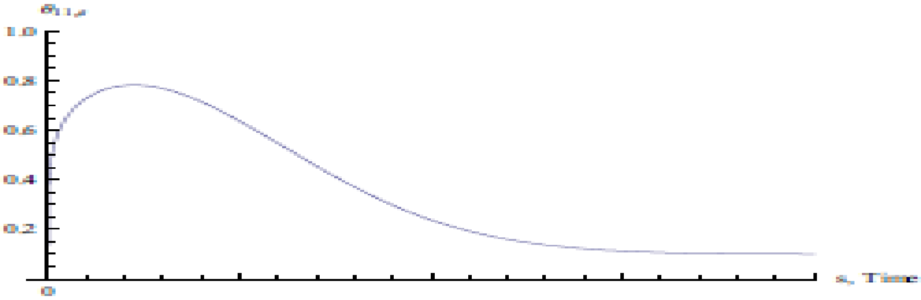
That is, the dynamic multiplier for any linear stochastic difference equation (SDE) depends only on the length of time j, not on time t.

- The *impulse-response function* (IRF) is an accumulation of the sequence of dynamic multipliers, as a function of time from the one time change in the innovation, ε_t.

Usually, IRFs are represented with a graph, that measures the effect of the innovation, ε_t, on y_t over time:

$$\delta y_{t+j}/\delta\varepsilon_t + \delta y_{t+j+1}/\delta\varepsilon_t + \delta y_{t+j+21}/\delta\varepsilon_t + ... = \psi_j + \psi_{j+1} + \psi_{j+2} + ...$$

- Once we estimate the AR, MA or ARMA coefficients, we draw an IRF.



Example: AR(1) process:

$$y_t = \mu + \phi_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

The AR(1) is stable if $|\phi_1|<1$ ⇒ stationarity condition.

We invert the AR(1) to get an MA(∞): $1/(1-\phi_1) = \sum_{j=0}^{\infty} \phi_1{}^j$

Then,

$$y_t = \mu^* + \phi_1\varepsilon_{t-1} + \phi_1{}^2\varepsilon_{t\text{-}}2 + \phi_1{}^3\varepsilon_{t\text{-}}3 + \phi_1{}^4\varepsilon_{t\text{-}}4 + \cdots + \varepsilon_t.$$

Under the stationarity condition, we calculate the dynamic multiplier:

$\delta y_{t+1}/\delta\varepsilon_{t-j} = \phi_1{}^{j}$

Accumulated over time, after J periods, the effect of shock ε_t at t+J is:

IRF(at t+J) = $\sum_{j=0}^{J-1} \phi_1{}^{j}$

Suppose $\phi_1 = 0.40$. Then,

$$\delta y_t/\delta\varepsilon_{t-1} = \phi_1 = 0.40$$
$$\delta y_t/\delta\varepsilon_{t-2} = \phi_1 = 0.40^2$$

⋮

$$\delta y_t/\delta\varepsilon_{t-J} = \phi_1 = 0.40^J$$

After 5 periods, the accumulated effect of a shock today is:

IRF(at t+5) = $0.40 + 0.40^2 + 0.40^3 + 0.40^4 + 0.40^5 = 0.65984$

AR Process – Causality

The AR(*p)* model:

Then, $y_t = \phi(L)^{-1}(\mu + \varepsilon_t)$, $\Rightarrow$ an MA(∞) process!

But, we need to make sure that we can invert the polynomial ϕ(L).

When ϕ(L) ≠ 0, we say the process y_t is *causal* (strictly speaking, a *causal function of* {ε_t}).

<u>Definition</u>: A linear process {y_t} is *causal* if there is a

<u>Definition</u>: A linear process {y_t} is *causal* if there is a

$$\psi(L) = 1 + \psi_1 L + \psi_2 L^2 + \cdots$$
$$\text{with} \quad \sum_{j=0}^{\infty} \left|\psi_j(L)\right| < \infty$$
$$\text{with} \quad y_t = \psi(L)\varepsilon_t.$$

Example: AR(1) process:

$$\phi(L)y_t = \mu + \varepsilon_t, \qquad \text{where}\,\phi(L) = 1 - \phi_1 L$$

Then, y_t is causal if and only if:

$|\phi_1| < 1$ (same condition as stationarity)

or

the root r_1 of the polynomial $\phi(z) = 1 - \phi_1 z$ satisfies $|r_1|>1$.

Question: How do we calculate the Ψ‘s coefficients for an AR(*p*)?

A: Matching coefficients (μ =0):

$$Y_t = \frac{1}{(1-\phi 1 L)}\varepsilon_t \overset{|\phi_1|<1}{\overset{\frown}{=}} \textstyle\sum_{i=0}^{\infty} \phi 1^i L^i \varepsilon_t$$
$$= (1 + \phi 1 L + \phi 1^2 L^2 + \cdots)\,\varepsilon_t \qquad \Rightarrow \Psi_i = \phi_1{}^{i}, \quad i \geq 0$$

AR Process – Estimation and Properties

We go back to the general AR*(p)*. Define

$$\boldsymbol{x}_t = \left(1\; y_{t-1}\; y_{t-2} \ldots.\, y_{t-p}\right)$$
$$\boldsymbol{\beta} = (\mu\; \phi_1\; \phi_2\; \ldots.\; \phi_p)$$

Then the model can be written as

$$y_t = \boldsymbol{x}_t{}'\boldsymbol{\beta} + \varepsilon_t$$

The OLS estimator is

$$\mathbf{b} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

• Properties:
– Using the Ergodic Theorem, OLS estimator is consistent.
– Using the MDS CLT, OLS estimator is asymptotically normal.
$\Rightarrow$ asymptotic inference is the same.

The asymptotic covariance matrix is estimated just as in the cross-section case: The sandwich estimator.

ARMA Process

A combination of AR(p) and MA(q) processes produces an ARMA(p, q) process:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots . + \phi_p y_{t-p} + \varepsilon_t - \theta_1\varepsilon_{t-1} - \theta_2\varepsilon_{t-2} - \cdots - \theta_q\varepsilon_{t-q}$$
$$= \mu + \sum_{i=1}^{p} \phi_i\, y_{t-i} - \sum_{i=1}^{q} \theta_i L^i \varepsilon_t + \varepsilon_t$$
$$\Rightarrow \phi(L) y_t = \mu + \theta(L)\varepsilon_t$$

Usually, we insist that ϕ(L) ≠ 0, $\theta(L)$ ≠ 0 & that the polynomials ϕ(L), $\theta(L)$ have no *common factors*. This implies it is not a lower order ARMA model.

ARMA Process – Common Factors

It is possible to reduce the order of an ARMA structure if the ϕ(L) and $\theta(L)$ lag polynomials have *common factors*.

Example: Suppose we have the following ARMA(2, 3) model

$$\phi(L) y_t = \theta(L)\varepsilon_t$$

with

$$\phi(L) = 1 - .6L + .3L^2$$
$$\theta(L) = 1 - 1.4L + .9L^2 - .3L^3 = (1 - .6L + .3L^2)(1 - L)$$

This model simplifies to: $y_t = (1 - L)\varepsilon_t$ $\Rightarrow$ an MA(1) process.

• Pure AR Representation: $\Pi(L)(y_t - \mu) = a_t \Rightarrow \Pi(L) = \frac{\phi_p(L)}{\theta_q(L)}$

• Pure MA Representation: $(y_t - \mu) = \Psi(L)a_t \Rightarrow \Psi(L) = \frac{\theta_q(L)}{\phi_p(L)}$

• Special ARMA(p, q) cases: – p = 0: MA(q)
– q = 0: AR(p).

ARMA: Stationarity, Causality and Invertibility

Theorem: If ϕ(L) and $\theta(L)$ have no common factors, a (unique) *stationary* solution to $\phi(L)y_t = \theta(L)\varepsilon_t$ exists if and only if

$$|z| \leq 1 \Rightarrow \phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p \neq 0.$$

This ARMA(p, q) model is causal if and only if

$$|z| \leq 1 \Rightarrow \phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \ldots - \phi_p z^p \neq 0.$$

This ARMA(p, q) model is invertible if and only if

$$|z| \leq 1 \Rightarrow \theta(z) = 1 + \theta_1 z - \theta_2 z^2 + \ldots + \theta_p z^p \neq 0.$$

Note: Real data cannot be *exactly* modeled using a finite number of parameters. We choose *p, q* to create a good approximated model.

Lecture 9 – ARIMA Models – Identification & Estimation

ARMA Process

We defined the ARMA(*p, q)* model:

$$\phi(L)(y_t - \mu) = \theta(L)\varepsilon_t$$

The mean does not affect the order of the ARMA. Then, if $\mu \neq 0$, we demean the data: $x_t = y_t - \mu$.

Then, $\phi(L)x_t = \theta(L)\varepsilon_t$ ⇒ x_t is a *demeaned* ARMA process.

• In this lecture, we will study:
- Identification of *p, q*.
- Estimation of ARMA(*p, q*)
- Non-stationarity of x_t.
- Differentiation issues – ARIMA(*p, d, q*)
- Seasonal behavior – SARIMA(*p, d, q*)s

Autocovariance Function (Again)

We define the autocovariance function: $\gamma(t-j) = E[y_t\, y_{t-j}]$

For an AR(*p*) process, WLOG with μ=0 (or demeaned y_t), we get:

$$\gamma(t-j) = E[(\phi_1 y_{t-1} y_{t-j} + \phi_2 y_{t-2} y_{t-j} + \ldots. + \phi_p y_{t-p} y_{t-j} + \varepsilon_t y_{t-j})]$$
$$= \phi_1\gamma(j-1) + \phi_2\gamma(j-2) + \ldots. + \phi_p\gamma(j-p)$$

Notation: γ(*k*) or γ_k are commonly used. Sometimes, γ(*k*) is referred as "*covariance at lag k.*"

The γ(*t-j*) determine a system of equations:

$$\gamma(0) = E[y_t, y_t] = \phi_1\gamma(1) + \phi_2\gamma(2) + \phi_3\gamma(3) + \cdots. + \phi_p\gamma(p) + \sigma^2$$
$$\gamma(1) = E[y_t, y_{t-1}] = \phi_1\gamma(0) + \phi_2\gamma(1) + \phi_3\gamma(2) + \cdots. + \phi_p\gamma(p-1)$$
$$\gamma(2) = E[y_t, y_{t-2}] = \phi_1\gamma(1) + \phi_2\gamma(0) + \phi_3\gamma(1) + \ldots. + \phi_p\gamma(p-2) \vdots \qquad \vdots$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

This a *pxp* system of equations. Using linear algebra, we can write the system as:

$$\boldsymbol{\Gamma}\, \boldsymbol{\phi} = \boldsymbol{\gamma}$$

where **Γ** is a *pxp* matrix of autocovariances, with $\gamma(0)$ on the diagonal; $\boldsymbol{\phi}$ is the *p*x1 vector of AR(p) coefficients; and **γ** is the *p*x1 vector of $\gamma(k)$ autocovariances

Example: AR(1) model:

$$y_t = \phi_1\, y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim WN.$$

Then, the autocovariance function is:

$$\gamma(0) = E[y_t\, y_t] = Var[y_{t-1}] = \sigma^2/(1 - \phi_1^2)$$

$\gamma(1) = E[y_t\, y_{t-1}] = E[(\phi_1\, y_{t-1} + \varepsilon_t) * y_{t-1}] = \phi_1\, \gamma(0)$
$\gamma(2) = E[y_t\, y_{t-2}] = E[(\phi_1\, y_{t-1} + \varepsilon_t) * y_{t-2}] = \phi_1\, \gamma(1) = \phi_1^2\, \gamma(0)$

$\gamma(3) = E[y_t\, y_{t-3}] = E[(\phi\; y_{t-1} + \varepsilon_t) * y_{t-3}] = \phi_1\, E[y_{t-1}\, y_{t-3}] = \phi_1^3\, \gamma(0)$

....

$\gamma(k) = \phi_1\, \gamma(k-1) = \phi_1^k\, \gamma(0) \;\; \Rightarrow$ If $|\phi_1|<1$, exponential decay.

Under stationarity, moments are constant. That is,

$\text{Var}[y_t] = \text{Var}[y_{t-1}] = \sqrt{\gamma(0)}$. ¶

Example: MA(1) process:

$y_t = \theta_1\, \varepsilon_{t-1} + \varepsilon_t, \qquad\qquad \varepsilon_t \sim \textit{WN}.$

Then, the autocovariance function is:

$\gamma(0) = \sigma^2 + \theta_1^2\, \sigma^2 = \sigma^2\, (1+ \theta_1^2)$
$\gamma(1) = E[y_t\, y_{t-1}] = E[(\theta_1\, \varepsilon_{t-1} + \varepsilon_t)(\theta_1\, \varepsilon_{t-2} + \varepsilon_{t-1})] = \theta_1\, \sigma^2$

.....

$\gamma(k) = E[y_t\, y_{t-k}] = E[(\theta_1\, \varepsilon_{t-1} + \varepsilon_t)(\theta_1\, \varepsilon_{t-(k+1)} + \varepsilon_{t-k})] = 0 \qquad\qquad$ (for $k>1$)

That is, for $|k| > 1$, $\quad \gamma(k) = 0$. ¶

Example: ARMA(1,1) process:

$y_t \; = \phi_1\, y_{t-1} + \theta_1\, \varepsilon_{t-1} + \varepsilon_t, \qquad\qquad \varepsilon_t \sim \textit{WN}.$

$$\begin{aligned}\gamma(k) = E[y_t\; y_{t-k}] &= E[\{\phi_1 y_{t-1} + \theta_1 \varepsilon_{t-1}\} y_{t-k}] \\ &= \phi_1 E[y_{t-1}\; y_{t-k}] + E[\varepsilon_t y_{t-k}] + \theta_1 E[\varepsilon_{t-1} y_{t-k}] \\ &= \phi_1 \gamma(k-1) + E[\varepsilon_t y_{t-k}] + \theta_1 E[\varepsilon_{t-1} y_{t-k}]\end{aligned}$$

$$\gamma(0) = \phi_1 \gamma(-1) + \underbrace{E[\varepsilon_t y_t]}_{\sigma^2} + \theta_1 E\left[\varepsilon_{t-1} \qquad \underbrace{y_t}_{\phi_1 y_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}}\right]$$

$$= \phi_1 \gamma(1) + \sigma^2 + \theta_1 E\left[\varepsilon_{t-1}[\phi_1 \qquad \underbrace{y_{t-1}}_{\phi_1 y_{t-2} + \varepsilon_{t-1} + \theta_1 \varepsilon_{t-2}} \qquad + \varepsilon_t + \theta_1 \varepsilon_{t-1}]\right]$$

$$= \phi_1 \gamma(1) + \sigma^2 + \theta_1\left(\phi_1 \sigma^2 + \theta_1 \sigma^2\right)$$

• Similarly:

$$\gamma(1) = \phi_1\, \gamma(0) + \underbrace{E[\varepsilon_t y_{t-1}]}_{0} + \theta_1 E\left[\varepsilon_{t-1} \qquad \underbrace{y_{t-1}}_{\phi 1 y_{t-2} + \varepsilon_{t-1} - \theta_1 \varepsilon_{t-2}}\right]$$

$$= \phi_1\, \gamma(0) + \theta_1 \sigma^2$$

• Two equations for $\gamma(0)$ and $\gamma(1)$. Solving for $\gamma(0)$:

$$\gamma(0) = \phi_1\left(\phi 1 \gamma(0) + \theta_1 \sigma^2\right) + \sigma^2\left(1 + \phi 1\, \theta_1 + \theta_1^2\right)$$

$$\gamma(0) = \sigma^2\, \frac{1 + \theta_1^2 + 2\phi 1\, \theta_1}{1 - \phi 1^{\,2}}$$

$$\begin{aligned}\gamma(1) &= \phi_1\,\gamma(0) + \theta_1\sigma^2 \\ &= \phi_1\,\sigma^2\,\frac{1+\theta_1{}^2+2\phi 1\,\theta_1}{1-\phi 1^{\,2}} + \theta_1\sigma^2 \\ &= \sigma^2\,\frac{(1+\phi 1\,\theta_1)*(\phi 1+\,\theta_1)}{1-\phi 1^{\,2}}\end{aligned}$$

Continuing the process:

$$\begin{aligned}\gamma(2) &= E[y_t\,y_{t-2}] \\ &= E[\{\phi_1 y_{t-1} - \theta_1\varepsilon_{t-1} + \varepsilon_t\}y_{t-2}] \\ &= \phi_1 E[y_{t-1}\,y_{t-2}] + \theta_1 E[\varepsilon_{t-1}y_{t-2}] + E[\varepsilon_t y_{t-2}] \\ &= \phi_1\gamma(1)\end{aligned}$$

In general:

$$\gamma(k) = \phi_1\gamma(k-1) = \phi_1{}^{k-1}\gamma(1), \quad k > 1$$

⇒ If $|\phi_1|$<1, exponential decay. ¶

Note: If stationary, ARMA(1,1) and AR(1) show exponential decay. Difficult to distinguish one from the other by looking at the autocovariance functions.

Autocorrelation Function (ACF)

Now, we define the autocorrelation function (ACF):

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \frac{\text{covariance at lag } k}{\text{variance}}$$

The ACF lies between -1 and +1, with $\rho(0) = 1$.

Dividing the autocovariance system by γ(0), we get:

$$\begin{bmatrix}\rho(0) & \rho(1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0)\end{bmatrix}\begin{bmatrix}\phi_1 \\ \phi_2 \\ \vdots \\ \phi_p\end{bmatrix} = \begin{bmatrix}\rho(1) \\ \rho(2) \\ \vdots \\ \rho(p)\end{bmatrix}$$

Or using linear algebra:

$$\mathbf{P}\,\boldsymbol{\phi} = \boldsymbol{\rho}$$

These are “Yule-Walker” equations, which can be solved numerically.

Autocorrelation Function (ACF) – Estimation & Correlogram

- **Estimation**:

Easy: Use sample moments to estimate γ(*k)* and plug in formula:

$$r_k = \hat{\rho}_k = \frac{\sum(Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum(Y_t - \bar{Y})^2}$$

Then, we plug the $\hat{\rho}_k$ in the Yule-Walker equations and solve for $\boldsymbol{\phi}$:

$$\widehat{\mathbf{P}}\,\boldsymbol{\phi} = \widehat{\boldsymbol{\rho}}$$

The sample *correlogram* is the plot of the ACF against *k*. As the ACF lies between -1 and +1, the correlogram also lies between these values.

• **Distribution**:
For a linear, stationary process, with large *T*, the distribution of the sample ACF, $r_k = \hat{\rho}_k$ is approximately normal with:

$$\mathbf{r} \xrightarrow{d} N(\boldsymbol{\rho}, \mathbf{V}/T), \qquad \mathbf{V} \text{ is the covariance matrix.}$$

Under H$_0$: $\rho_k = 0$

$$\mathbf{r} \xrightarrow{d} N(\mathbf{0}, \mathbf{I}/T) \qquad \Rightarrow \text{Var}[r(k)] = 1/T.$$

Under H$_0$: $\rho_k = 0$, the SE = $1/\sqrt{T}$ ⇒ 95% C.I.: $0 \pm 1.96 * 1/\sqrt{T}$

Then, for a white noise sequence, approximately 95% of the sample ACFs should be within the above C.I. limits.

Note: The SE = $1/\sqrt{T}$ are sometimes referred as *Bartlett's SE*.

Example: Sample ACF for an AR(1) process:
Under stationarity (Var[y_t]= Var[y_{t-1}]=$\sqrt{\gamma(0)}$):

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)} = \phi_1^k \qquad k = 0, 1, 2, \ldots$$

If | ϕ_1 |< 1, the ACF will show exponential decay.

Suppose $\phi_1 = 0.4$. Then,

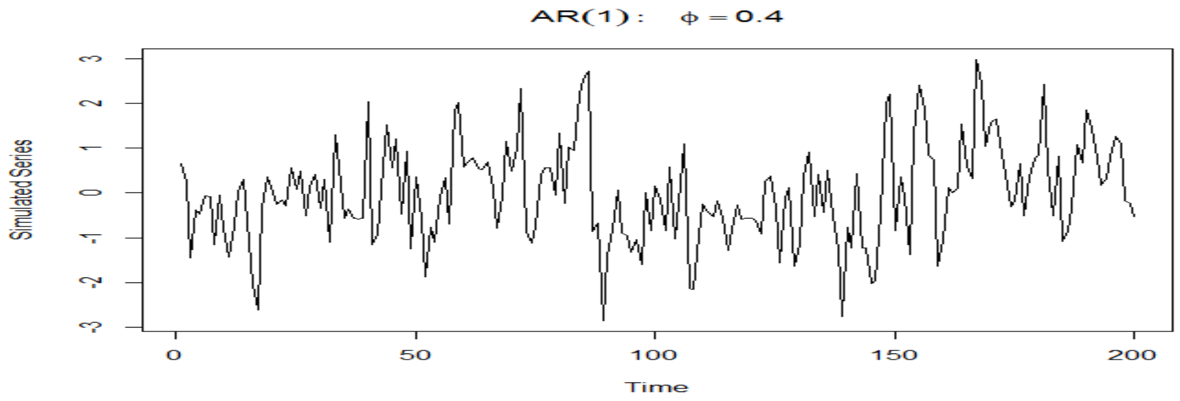
$$\rho(0) = 1$$
$$\rho(1) = 0.4$$
$$\rho(2) = 0.4^2 = 0.16$$
$$\rho(3) = 0.4^3 = 0.064$$
$$\rho(4) = 0.4^4 = 0.0256$$

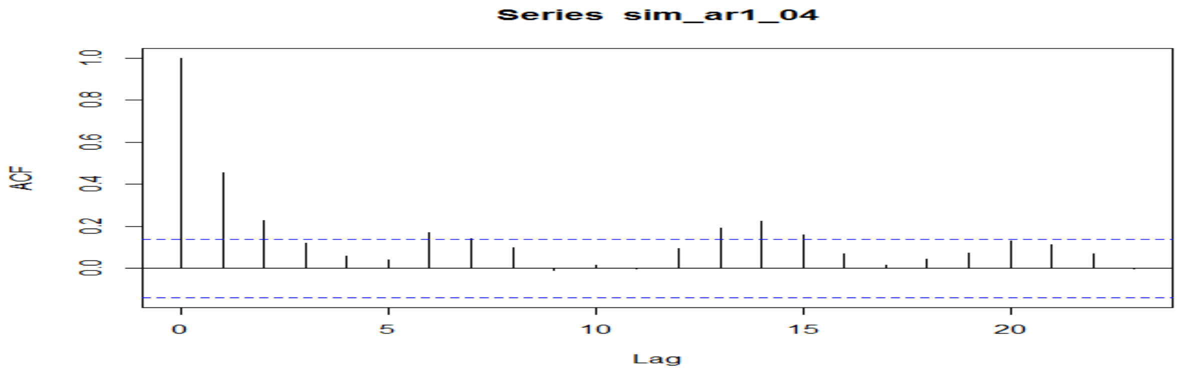
⋮

$$\rho(k) = = 0.4^k$$

• We simulate an AR(1) series with with ϕ_1= 0.4, using the R function *arima.sim*.

```
sim_ar1_04 <- arima.sim(list(order=c(1,0,0), ar=0.4), n=200)        #simulate AR(1) series
plot(sim_ar1_04, ylab="Simulated Series", main=(expression(AR(1):~~~phi==0.4)))
acf(sim_ar1_04)                                                      #plot ACF for sim series
```





Example: Sample ACF for an MA(1) process:

$\rho(0) = 1$

$\rho(k) = \theta_1/(1+ \theta_1^2)$, for $k = 1, -1$

$\rho(k) = 0$ for $|k| > 1$.

After $k = 1$ –i.e., one lag– the ACF dies out.

Suppose $\theta_1 = 0.5$. Then,

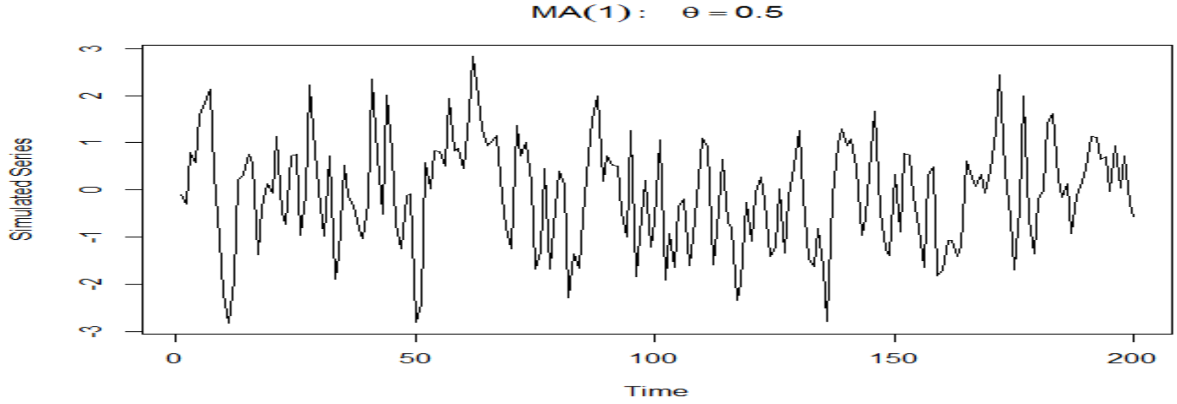
$\rho(0) = 1$

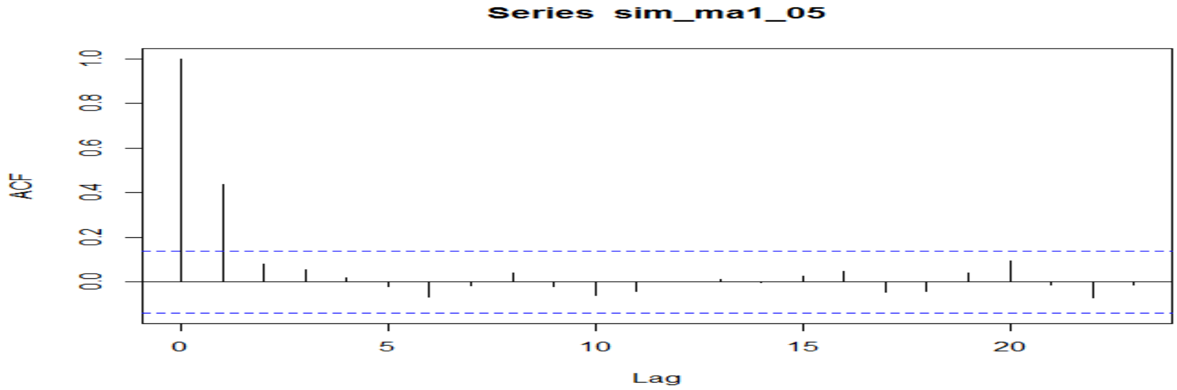
$\rho(1) = 0.4$

$\rho(k) = 0$ for $|k| > 1$.

- We simulate an MA(1) series with ϕ_1=0.4

```
sim_ma1_05 <- arima.sim(list(order=c(0,0,1), ma=0.5), n=200)      #simulate MA(1) series
plot(sim_ma1_05, ylab="Simulated Series", main=(expression(MA(1):~~~theta==0.5)))
acf(sim_ma1_05)                                                   #plot ACF for sim series
```





Example: Sample ACF for an ARMA(1,1) process:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t + \theta_1 \varepsilon_{t-1}$$

From the autocovariances, we get

$$\gamma(0) = \sigma^2 \frac{1+\theta_1{}^2+2\phi 1\, \theta_1}{1-\phi 1^{\,2}}$$

$$\gamma(1) = \sigma^2 \frac{(1+\phi 1\, \theta_1)*(\phi 1+ \theta_1)}{1-\phi 1^{\,2}}$$

$$\gamma(k) = \phi_1 \gamma(k-1) = \phi_1{}^{k-1} \sigma^2 \frac{(1+\phi 1\, \theta_1)*(\phi 1+ \theta_1)}{1-\phi 1^{\,2}}$$

Then,

$$\rho(k) = \phi_1{}^{k-1} \frac{(1+\phi 1\; \theta 1)*(\phi 1+\theta 1)}{1+\theta 1^2+2\phi 1\theta 1}$$

⇒ If $|\phi_1|<1$, exponential decay. Similar pattern to AR(1).

- The ACF for an ARMA(1,1):

$$\rho(k) = \phi_1{}^{k-1} \frac{(1+\phi 1\; \theta 1)*(\phi 1+\theta 1)}{1+\theta 1^2+2\phi 1\theta 1}$$

- Suppose $\phi_1 = 0.4$, $\theta_1 = 0.5$. Then,

$$\rho(0) = 1$$

$$\rho(1) = \frac{(1+0.4*0.5)*(0.4+0.5)}{1+0.5^2+2*0.4*0.5} = 0.6545$$

$$\rho(2) = 0.4 * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5} = 0.2618$$

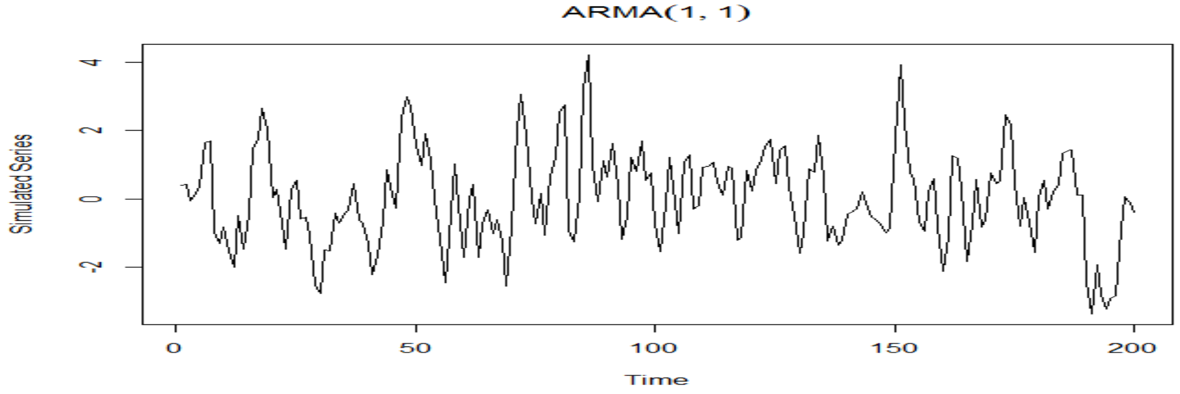
$$\rho(3) = 0.4^2 * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5} = 0.0233$$

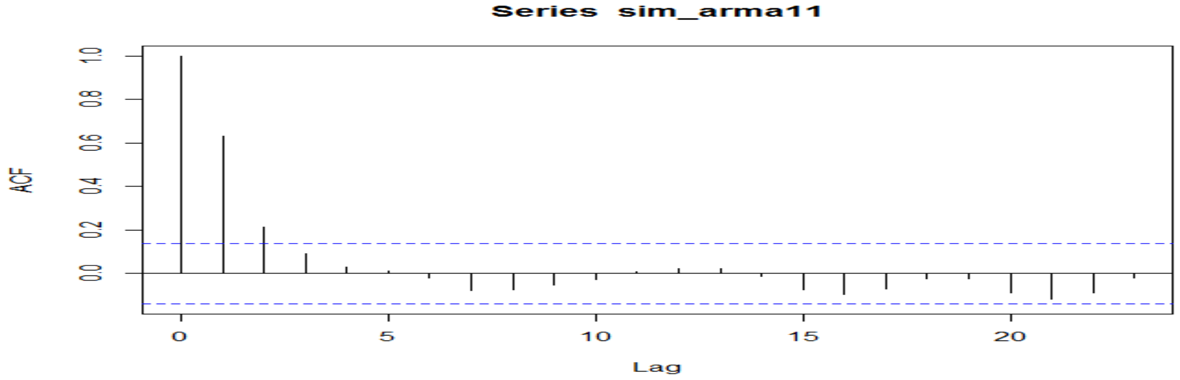
$\vdots$

$$\rho(k) = 0.4^{k\text{-}1} * \frac{(1 + 0.4 * 0.5) * (0.4 + 0.5)}{1 + 0.5^2 + 2*0.4*0.5}$$

Plot of simulated series ARMA (1,1) and ACF

```
> sim_arma11 <- arima.sim(list(order=c(1,0,1), ar=0.4, ma=0.5), n=200)   # sim ARMA(1,1)
```



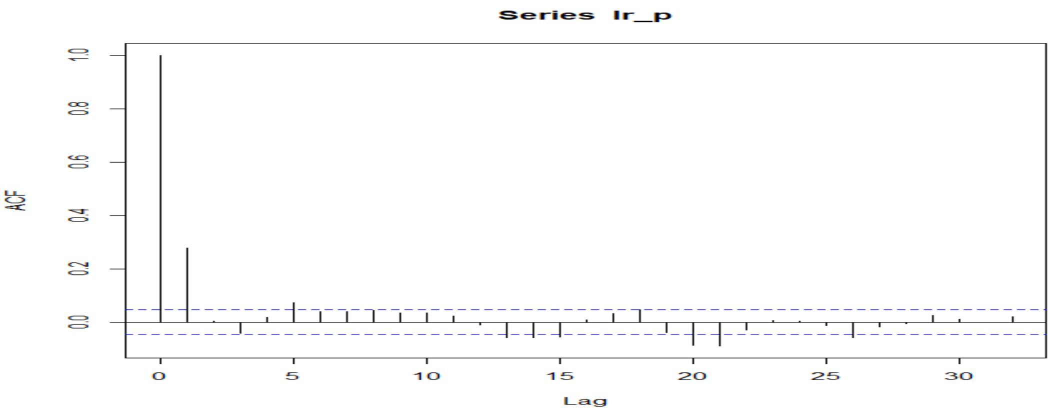


Example: US Monthly Returns (1871 – 2020, T=1,795)

```
Sh_da <- read.csv("C://Financial Econometrics/Shiller_2020data.csv", head=TRUE, sep=",")
x_P <- Sh_da$P
x_D <- Sh_da$D
T <- length(x_P)
lr_p <- log(x_P[-1]/x_P[-T])
lr_d <- log(x_D[-1]/x_D[-T])
acf_p <- acf(lr_p)                          # acf: R function that estimates the ACF
> acf_p
Autocorrelations of series ‘lr_p’, by lag
0        1      2     3      4      5      6      7      8      9      10     11
1.000  0.279  0.004  -0.043  0.017  0.074  0.039  0.039  0.044  0.035  0.034  0.022
   12     13     14     15     16     17     18     19     20     21     22     23
-0.010 -0.059 -0.058 -0.056  0.009  0.033  0.047 -0.040 -0.087 -0.090 -0.029  0.005
```

```
   24    25      26     27      28      29     30      31     32
 0.003 -0.013 -0.058 -0.018 -0.005  0.026  0.011  0.000  0.020
```

SE(r_k) = 1/sqrt(T) = 1/sqrt(1,795) = .0236. ⇒ 95% C.I.: ± 2* 0.0236

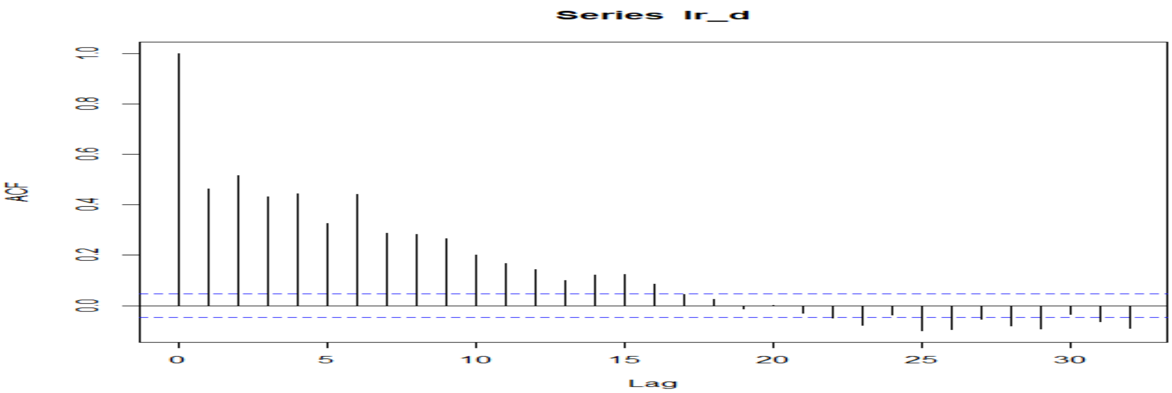


Note: With the exception of first correlation, correlations are small. However, many are significant, not strange result when T is large. ¶

Example: US Monthly Stock Dividends (1871 – 2020, T=1,795)

```
> acf_d
Autocorrelations of series ‘lr_d’, by lag
0         1        2       3        4       5        6        7       8       9        10     11
1.000  0.462  0.516  0.432  0.444  0.326  0.442  0.288  0.283  0.265  0.202  0.168
   12    13      14     15      16     17      18      19     20     21      22     23
0.142  0.100  0.122  0.123  0.085  0.045  0.026 -0.013  0.001 -0.029 -0.049 -0.077
   24     25      26     27      28      29     30      31     32
-0.038 -0.100 -0.095 -0.055 -0.081 -0.092 -0.034 -0.063 -0.089
```

High correlations and significant even after 32 months!



Note: Correlations are positive for almost 1.5 years, then correlations become negative. ¶

ACF – Joint Significance Tests

Recall the Ljung-Box (LB) statistic as:

$$LB = T(T+2)\sum_{k=1}^{m}\left(\frac{\hat{\rho}_k^2}{(T-k)}\right)$$

The LB test can be used to determine if the first m sample ACFs are jointly equal to zero. Under H_0: $\rho_1 = \rho_2 = \ldots = \rho_m = 0$, the LB has an asymptotic $\chi^2(m)$ distribution.

Example: LB test with 20 lags for US Monthly Returns and Changes in Dividends (1871 – 2020)

```
> Box.test(lr_p, lag=20, type= "Ljung-Box")

        Box-Ljung test

data:  lr_p
X-squared = 208.02, df = 20, p-value < 2.2e-16        ⇒ Reject H0 at 5% level. Joint significant
first 20 correlations.

> Box.test(lr_d, lag=20, type= "Ljung-Box")

        Box-Ljung test

data:  lr_d
X-squared = 2762.7, df = 20, p-value < 2.2e-16       ⇒ Reject H0 at 5% level. Joint significant
            first 20 correlations. ¶
```

Partial ACF (PACF)

The ACF gives us a lot of information about the order of the dependence when the series we analyze follows a MA process: The ACF is zero after q lags for an MA(q) process.

If the series we analyze, however, follows an ARMA or AR, the ACF alone tells us little about the orders of dependence: We only observe an exponential decay.

We introduce a new function that behaves like the ACF of MA models, but for AR models, namely, the partial autocorrelation function (PACF).

The PACF is similar to the ACF. It measures correlation between observations that are k time periods apart, after controlling for correlations at intermediate lags.

Intuition: Suppose we have an AR(1):

$$y_t = \phi_1 y_{t-1} + \varepsilon_t.$$

Then,

$$\gamma(2) = \phi_1^2\,\gamma(0)$$

The correlation between y_t and y_{t-2} is not zero, as it would be for an MA(1), because y_t is dependent on y_{t-2} through y_{t-1}.

Suppose we break this chain of dependence by removing ("partialing out") the effect y_{t-1}. Then, we consider the correlation between $[y_t–\phi_1 y_{t-1}]$ and $[y_{t-2} – \phi_1 y_{t-1}]$ –i.e, the correlation between y_t and y_{t-2} with the linear dependence of each on y_{t-1} removed:

$$\gamma(2) = \text{Cov}(y_t – \phi_1 y_{t-1}, y_{t-2} – \phi_1 y_{t-1}) = \text{Cov}(\varepsilon_t, y_{t-2} – \phi_1 y_{t-1}) = 0.$$

Similarly,

$$\gamma(k) = \text{Cov}(\varepsilon_t, y_{t-k} – \phi_1 y_{t-1}) = 0 \qquad \text{for all } k > 1.$$

Definition: The PACF of a stationary time series $\{y_t\}$ is

$\phi_{11} = \text{Corr}(y_t, y_{t-1}) = \rho(1)$

$\phi_{hh} = \text{Corr}(y_t – E[y_t|I_{t-1}], y_{t-h} – E[y_{t-h}|I_{t-1}])$ for $h = 2, 3, ...$

This removes the linear effects of y_{t-1}, y_{t-2} , ... y_{t-h-1} .

The PACF Φ_{hh} is also the last coefficient in the best linear prediction of y_t given $y_{t-1}, y_{t-2}, ..., y_{t-h}$.

Estimation by Yule-Walker equation, using sample estimates:

$$\hat{\boldsymbol{\phi}}_h = [\hat{\boldsymbol{R}}]^{-1}\hat{\gamma}(k) \qquad \Rightarrow \text{a recursive system,}$$

where $\boldsymbol{\phi}_h = (\phi_{h1}, \phi_{h2} , ..., \phi_{hh})$ and $\boldsymbol{R}$ is the (hxh) correlation matrix.

A recursive algorithm, Durbin-Levinson, can be used. Also OLS can be used.

Partial ACF – AR(p)

Example: AR(p) process:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ ... + \ \phi_p y_{t-p} + \varepsilon_t$$
$$E[y_t|I_{t-1}] = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ ... + \phi_p y_{t-h-1}$$
$$E[y_{t-h}|I_{t-1}] = \mu + \phi_1 y_{t-h-1} + \phi_2 y_{t-h-2} + \ ... + \phi_p y_{t-1}$$

Then,

$$\phi_{hh} \quad = \phi_h \quad \text{if } 1 \leq h \leq p$$
$$\quad = 0 \quad \text{otherwise}$$

⇒ After the p^{th} *PACF,* all remaining PACF are 0 for AR(p) processes. ¶

The plot of the PACF is called the *partial correlogram*.

Example: We simulate an AR(2) process:

$$y_t = \mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t$$

```
sim_ar2 <- arima.sim(list(order=c(1,0,0), ar=c(0.5, 0.3)), n=200)    #simulate AR(2) series
plot(sim_ar2, ylab="Simulated Series", main=(expression(AR(2))))
pacf_ar2 <- pacf(sim_ar2)
```

• Print PACF

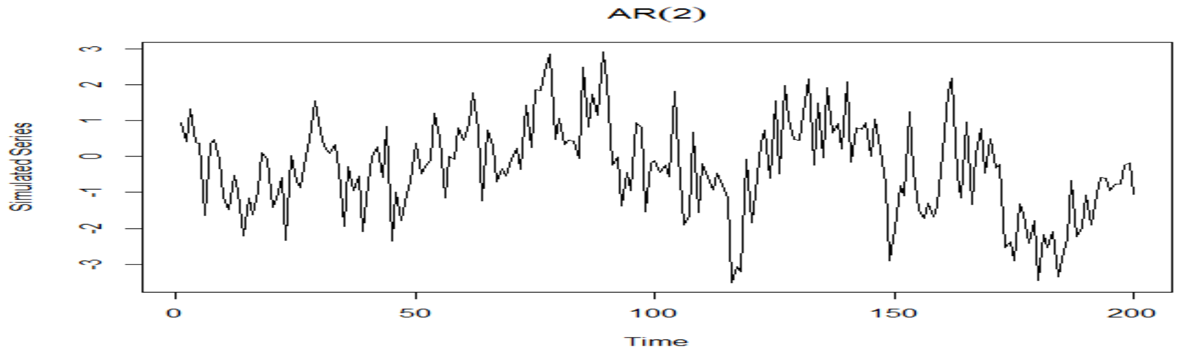
```
> pacf_ar2
  1      2      3      4      5      6      7      8      9     10     11
0.558  0.286  0.038  0.103 -0.010  0.009  0.111  0.060 -0.021 -0.076  0.016
```

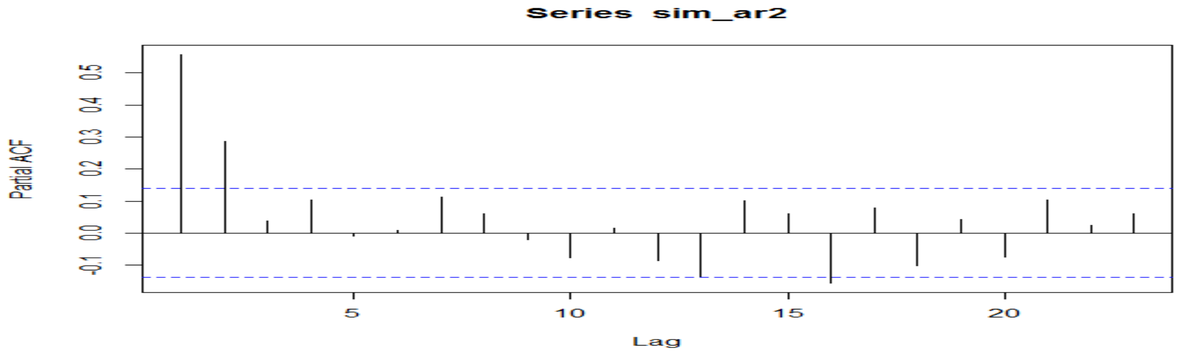
```
    12    13       14     15       16     17       18       19     20      21       22      23
-0.086 -0.139  0.100  0.061 -0.156  0.078 -0.103  0.043 -0.075  0.104  0.024  0.061
   24     25       26     27       28       29      30       31      32
-0.038 -0.100 -0.095 -0.055 -0.081 -0.092 -0.034 -0.063 -0.089
```

SE(r_k) ≈ 1/sqrt(200) = .0707. ⇒ 95% C.I.: ± 2* 0.0707

Plot of simulated series and PACF

```
> plot(sim_ar2, ylab="Simulated Series", main=(expression(AR(2))))
>pacf_ar2 <- pacf(sim_ar2)
```





<u>Note</u>: The PACF can be calculated by *h* regressions, each one with *h* lags. The *hh* coefficient is the h^{th} order PACF.

```
> ar(sim_ar2, order.max=1, method = "ols")
Coefficients:
     1
0.5586
Intercept: -0.008403 (0.0761)
Order selected 1  sigma^2 estimated as  1.152

> ar(sim_ar2, order.max=2, method = "ols")
Coefficients:
     1       2
```

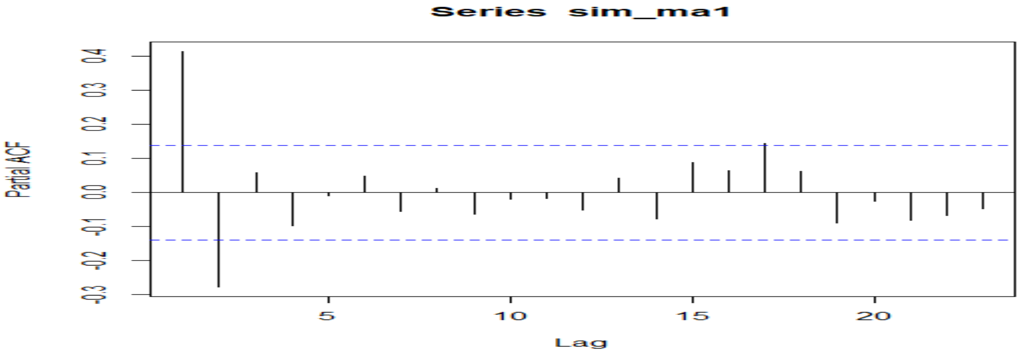
```
0.3974  0.2869
Intercept: -0.009847 (0.07326)
Order selected 2  sigma^2 estimated as  1.063.
```

Partial ACF – MA(q)

Following the analogy that PACF for AR processes behaves like an ACF for MA processes, we will see exponential decay ("*tails off*") in the partial correlogram for MA process. Similar pattern will also occur for ARMA(p, q) process.

Example: We simulate an MA(1) process with $\theta_1 = 0.5$.

```
sim_ma1 <- arima.sim(list(order=c(0,0,1), ma=0.5), n=200)
> pacf(sim_ma1)
```

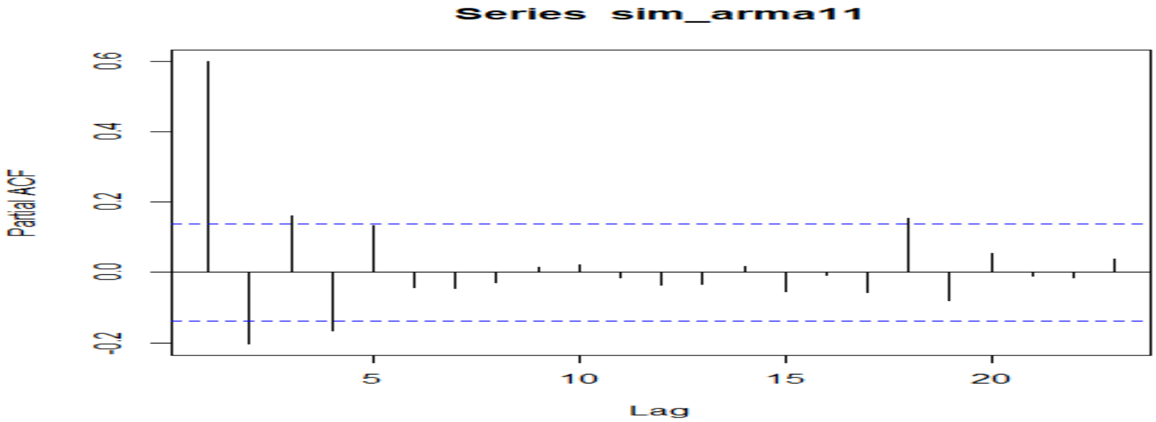


Partial ACF – ARMA(p,q)

For an ARMA processes, we will see exponential decay ("*tails off*") in the partial correlogram.

Example: We simulate an ARMA(1) process with $\phi_1 = 0.4$ & $\theta_1 = 0.5$.

```
sim_arma11 <- arima.sim(list(order=c(1,0,1), ar=0.4, ma=0.5), n=200)
> pacf(sim_arma11)
```



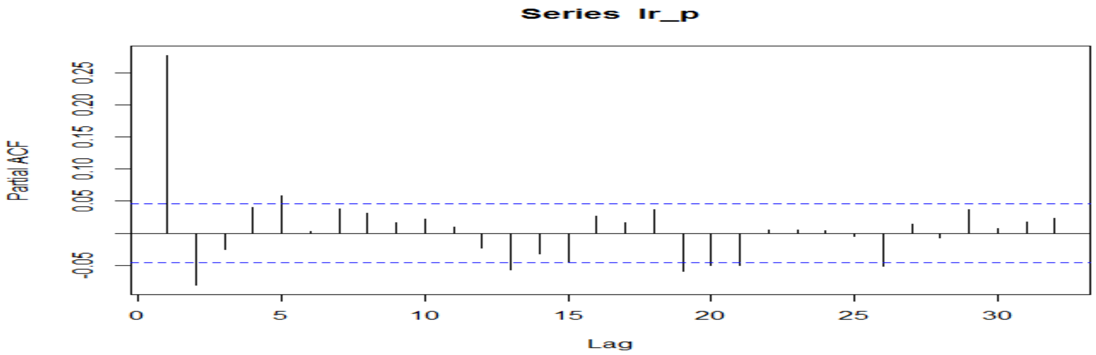
Partial ACF – Examples

Example: US Monthly Returns (1871 – 2019, *T*=1,795)

```
pacf_p <- acf(lr_p)                          # pacf: R function that estimates the PACF
> pacf_p

Partial autocorrelations of series ‘lr_p’, by lag
  1      2      3      4      5      6      7      8      9     10     11
 0.278 -0.081 -0.026  0.041  0.058  0.002  0.038  0.032  0.016  0.022  0.009
 12     13     14     15     16     17     18     19     20     21     22
-0.023 -0.057 -0.032 -0.045  0.027  0.017  0.037 -0.059 -0.051 -0.050  0.005
   23     24     25     26     27     28     29     30     31     32
0.006  0.004 -0.005 -0.051  0.014 -0.007  0.037  0.008  0.018  0.023
```

SE(r_k) = 1/sqrt(1,795) = .0236.   ⇒ 95% C.I.: ± 2* 0.0236

```
> pacf(lr_p)
```



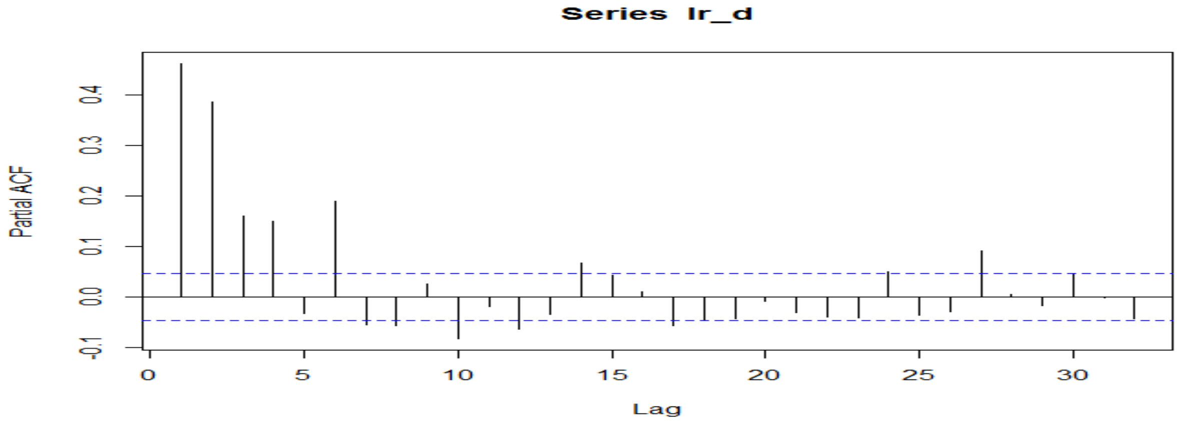
Note: With the exception of the first partial correlation, partial correlations are small, though, again, some are significant. ¶

Example: US Monthly Stock Dividends (1871 – 2020, T=1,795)

```
pacf_d <- pacf(lr_d)
> pacf_d

Partial autocorrelations of series ‘lr_d’, by lag
  1      2      3      4      5      6      7      8      9     10     11
0.462  0.385  0.160  0.150 -0.033  0.189 -0.054 -0.056  0.027 -0.082 -0.019
 12     13     14     15     16     17     18     19     20     21     22
-0.023 -0.057 -0.032 -0.045  0.027  0.017  0.037 -0.059 -0.051 -0.050  0.005
   23     24     25     26     27     28     29     30     31     32
-0.041  0.050 -0.036 -0.030  0.091  0.006 -0.017  0.044 -0.002 -0.042
```

Higher partial correlations than for stock returns.

```
> pacf(lr_d)
```



Note: Partial correlations are positive for almost 6 lags, then become small. ¶

Non-Stationary Time Series Models

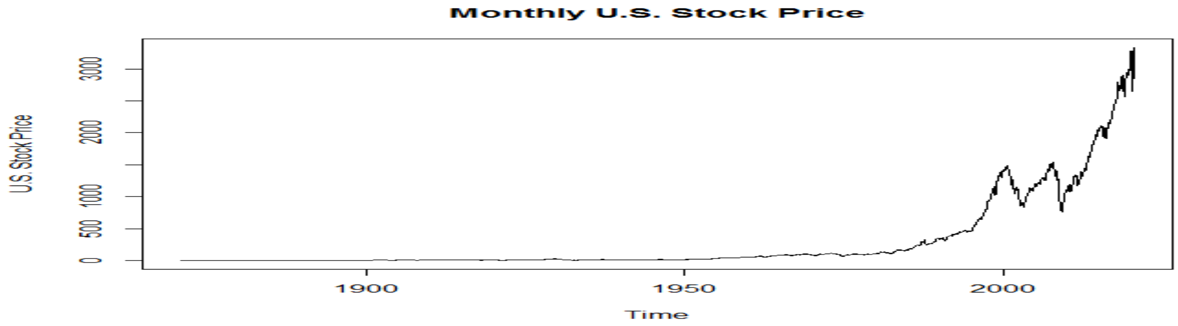
The ACF is as a rough indicator of whether a trend is present in a series. A slow decay in ACF is indicative of highly correlated data, which suggests a true unit root process, or a trend stationary process.

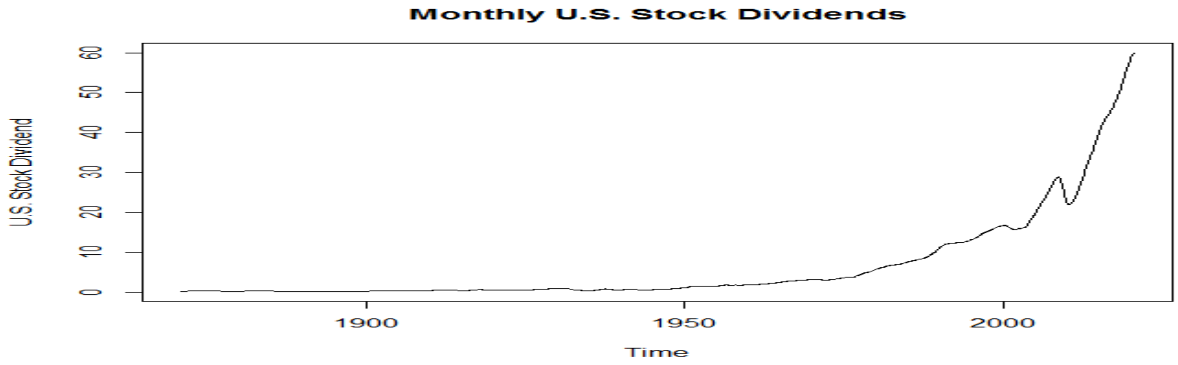
Formal tests can help to determine whether a system contains a trend and whether the trend is deterministic or stochastic (unit root).

We will analyze two situations faced in ARMA models:
(1) Deterministic trend – Simple model: $y_t = \alpha + \beta\, t + \varepsilon_t$.
– Solution: *Detrending* –i.e., regress y_t on *t*. Then, keep residuals for further modeling.

(2) Stochastic trend – Simple model: $y_t = c + y_{t-1} + \varepsilon_t$.
– Solution: *Differencing* –i.e., apply $\Delta = (1 - L)$ operator to y_t . Then, use Δy_t for further modeling.

Example: Plot of US Monthly Prices and Dividends (1871 – 2020)





Non-Stationary Time Series Models – Deterministic Trend

Suppose we have the following model:

$$y_t = \alpha + \beta\, t + \varepsilon_t. \qquad \Rightarrow \Delta y_t = y_t - y_{t-1}$$
$$= \beta\, t - \beta\,(t-1) + \varepsilon_t - \varepsilon_{t-1}$$
$$= \beta + \varepsilon_t - \varepsilon_{t-1}$$
$$\Rightarrow E[\Delta y_t] = \beta$$

$\{y_t\}$ will show only temporary departures from the trend line $\alpha+\beta\, t$. This type of model is called a trend stationary (TS) model.

If a series has a deterministic time trend, then we simply regress y_t on an intercept and a time trend ($t = 1, 2, \ldots, T$) and save the residuals. The residuals are the *detrended* y_t series ($=y_t$ without the influence of t).

If y_t is stochastic, we do not necessarily get stationary series, by detrending.

Many economic series exhibit "exponential trend/growth". They grow over time like an exponential function over time instead of a linear function. In this cases, it is common to work with logs

$$\ln(y_t) = \alpha + \beta\, t + \varepsilon_t.$$
$\Rightarrow$ The average growth rate is: $E[\Delta \ln(y_t)] = \beta$

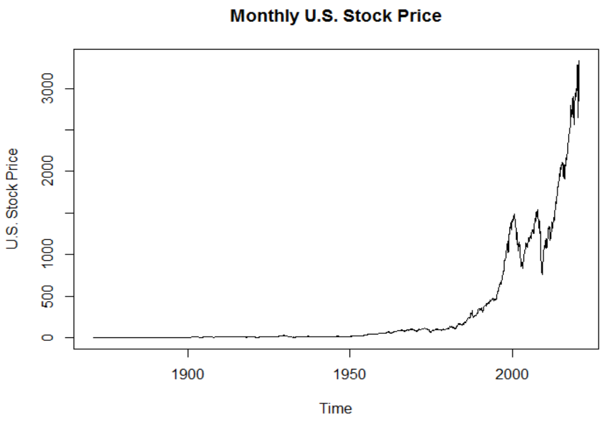
We can have a more general model:

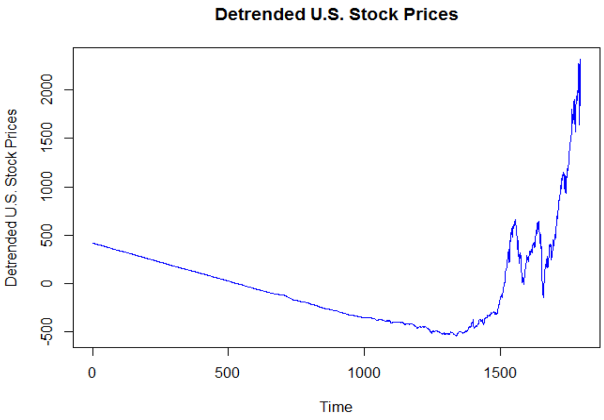
Estimation:
- OLS.
- Frish-Waugh method (a 2-step method):
  (1) Detrend y_t, get the residuals ($=y_t$ without the influence of t)
  (2) Use residuals to estimate the AR(p) model.

Example: We detrend **U.S. Stock Prices**

```
T <- length(x_P)                         # length of series
trend <- c(1:T)                          # create trend
det_P <- lm(x_P ~ trend)                 # regression to get detrended e
```

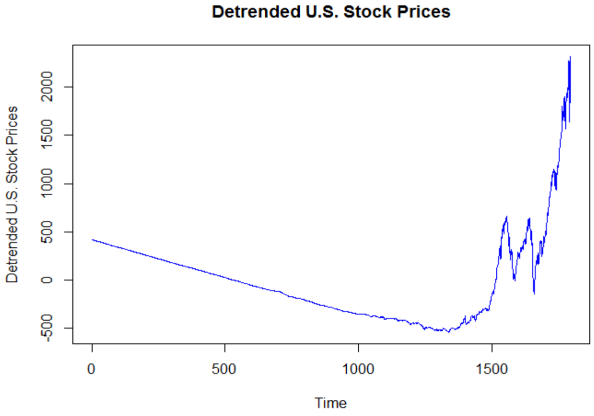
```
detrend_P <- det_P$residuals
plot(detrend_P, type="l", col="blue", ylab ="Detrended U.S. Prices", xlab ="Time")
title("Detrended U.S. Stock Prices")
```

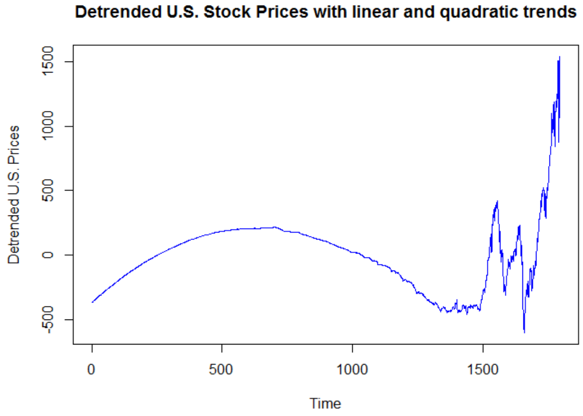




- Not very appealing series. We still see trends. Now, we detrend U.S. Stock Prices adding a square trend

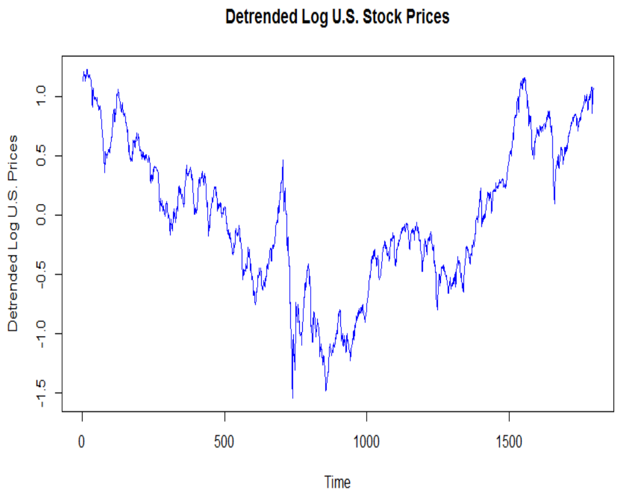
```
trend2 <- trend^2
det_P <- lm(x_P ~ trend + trend2)              # regression to get detrended e
detrend_P <- det_P$residuals
plot(detrend_P, type="l", col="blue", ylab ="Detrended U.S. Prices", xlab ="Time")
title("Detrended U.S. Stock Prices with linear and quadratic trends")
```

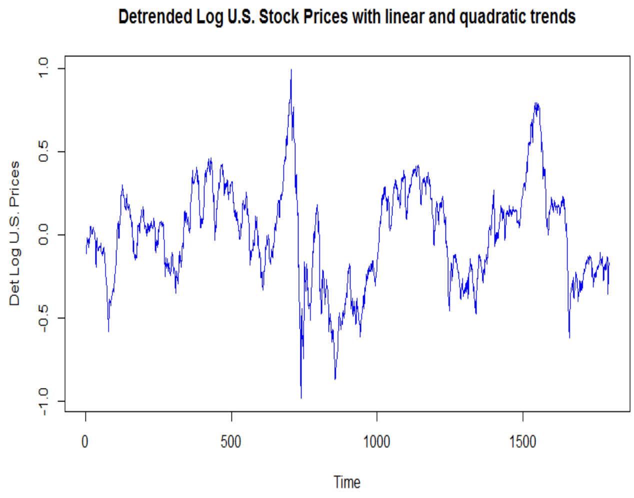




- Still, trends are still observed. We detrend **Log U.S. Stock Prices** adding a square trend

```
l_P <- log(x_P)
det_lP <- lm(l_P ~ trend)                      # regression to get detrended e
detrend_lP <- det_lP$residuals
plot(detrend_lP, type="l", col="blue", ylab ="Detrended Log U.S. Prices", xlab ="Time")
title("Detrended Log U.S. Stock Prices")
det_lP2 <- lm(l_P ~ trend + trend2)            # regression to get detrended e
det_lP2 <- det_lP$residuals
plot(det_lP2, type="l", col="blue", ylab ="Det Log U.S. Prices", xlab ="Time")
title("Detrended Log U.S. Stock Prices with linear and quadratic trends")
```





Remark: The second detrended series, with linear and quadratic trends looks better, but we still see trends in the graph. ¶

Non-Stationary Time Series Models – Stochastic Trend

The more modern approach is to consider trends in time series as a variable.

A variable trend exists when a trend changes in an unpredictable way. Therefore, it is considered as *stochastic*.

Recall the AR(1) model: $y_t = c + \phi_1 y_{t-1} + \varepsilon_t$.

As long as $|\phi| < 1$, everything is fine: OLS is consistent, t-stats are asymptotically normal, etc.

Now consider the extreme case where $\phi_1 = 1$, $\Rightarrow y_t = c + y_{t-1} + \varepsilon_t$.

Where is the (stochastic) trend? No *t* term.

Let us replace recursively the lag of y_t on the right-hand side:

$$
\begin{aligned}
y_t &= \mu + y_{t-1} + \varepsilon_t \\
&= \mu + (\mu + y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\
&\ldots \\
&= y_0 + t\,\mu + \textstyle\sum_{j=0}^{t} \varepsilon_{t-j}
\end{aligned}
$$

A *constant* (y_0), a *determinist trend* ($t\,\mu$) and an accumulation of errors over time ($\sum_{j=0}^{t} \varepsilon_{t-j}$) appear in the recursive formulation. This is what we call a "*random walk with drift*". The series grows with *t*.

Each ε_t shock represents a shift in the intercept. All values of $\{\varepsilon_t\}$ have a 1 as coefficient

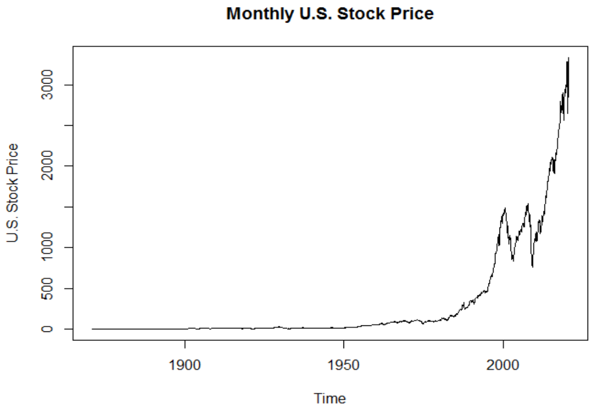
$\Rightarrow$ each shock never vanishes (permanent).

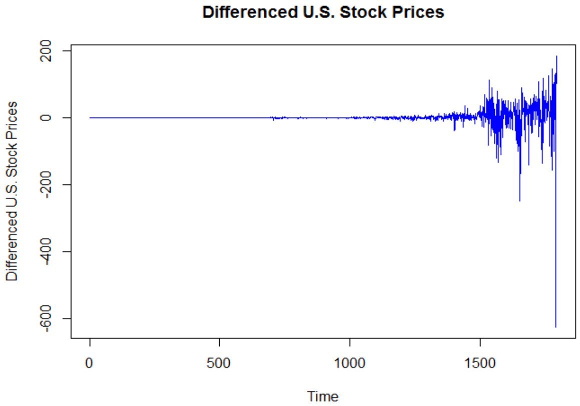
We remove the trend by differencing $y_t \Rightarrow \Delta y_t = (1 - L)\, y_t = \mu + \varepsilon_t$

<u>Note</u>: Applying the $(1 - L)$ operator to a time series is called *differencing*

Example: We difference **U.S. Stock Prices**, using the *diff* R function:

```
diff_P <- diff(x_P)
> plot(diff_P,type="l", col="blue", ylab ="Differenced U.S. Stock Prices", xlab ="Time")
> title("Differenced U.S. Stock Prices")
```





<u>Remark</u>: The trend is gone from the graph. ¶

• y_t is said to have a *stochastic trend* (ST), since each ε_t shock gives a permanent and random change in the conditional mean of the series.

For these situations, we use *Autoregressive Integrated Moving Average (ARIMA)* models.

Question: Deterministic or Stochastic Trend?
They appear similar: Both lead to growth over time. The difference is how we think of ε_t. Should a shock today affect y_{t+1}?

– TS: $y_{t+1} = c + \beta\,(t+1) + \varepsilon_{t+1}$ ⇒ ε_t does not affect y_{t+1}.

– ST: $y_{t+1} = c + y_t + \varepsilon_{t+1} = c + [c + y_{t-1} + \varepsilon_t] + \varepsilon_{t+1}$ ⇒ ε_t affects y_{t+1}. (In fact, the shock will have a permanent impact.)

ARIMA(p,d,q) Models

For $p, d, q \geq 0$, we say that a time series $\{y_t\}$ is an *ARIMA (p,d,q) process* if $w_t = \Delta^d y_t = (1 - L)^d\, y_t$ is ARMA(p,q). That is,

Applying the $(1 - L)$ operator to a time series is called *differencing*.

Notation: If y_t is non-stationary, but $\Delta^d y_t$ is stationary, then y_t is integrated of order *d*, or I*(d)*. A time series with *unit root* is I(1). A stationary time series is I(0).

Examples:

Example 1: RW: $y_t = y_{t-1} + \varepsilon_t$.
y_t is non-stationary, but

$$(1 - L)\, y_t = \varepsilon_t \quad \Rightarrow \text{white noise!}$$

Now, $y_t \sim$ ARIMA(0,1,0).

Example 2: AR(1) with time trend: $y_t = \mu + \delta\, t + \phi_1\, y_{t-1} + \varepsilon_t$.
y_t is non-stationary, but

$$w_t = (1 - L)\, y_t = \mu + \delta\, t + \phi_1\, y_{t-1} + \varepsilon_t. - (\mu + \delta\, (t\text{-}1) + \phi_1\, y_{t-2} + \varepsilon_{t-1}.$$
$$= \delta + \phi_1\, w_{t-1} + \varepsilon_t - \varepsilon_{t-1}$$

Now, $y_t \sim$ ARIMA(1,1,1).

We call both process first difference stationary. ¶

Note:
– Example 1: Differencing a series with a unit root in the AR part of the model reduces the AR order.
– Example 2: Differencing can introduce an extra MA structure. We introduced non-invertibility. This happens when we difference a TS series. Detrending should be used in these cases.

• In practice:

A root near 1 of the AR polynomial	⇒ differencing
A root near 1 of the MA polynomial	⇒ over-differencing

• In general, we have the following results:
- Too little differencing: not stationary.
- Too much differencing: extra dependence introduced.

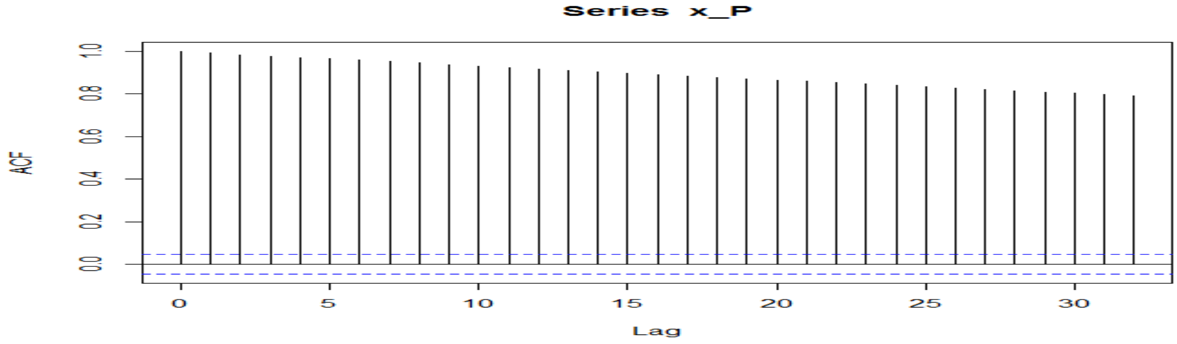
• Finding the right *d* is crucial. For identifying preliminary values of *d*:
- Use a time plot.
- Check for slowly decaying (persistent) ACF/PACF.

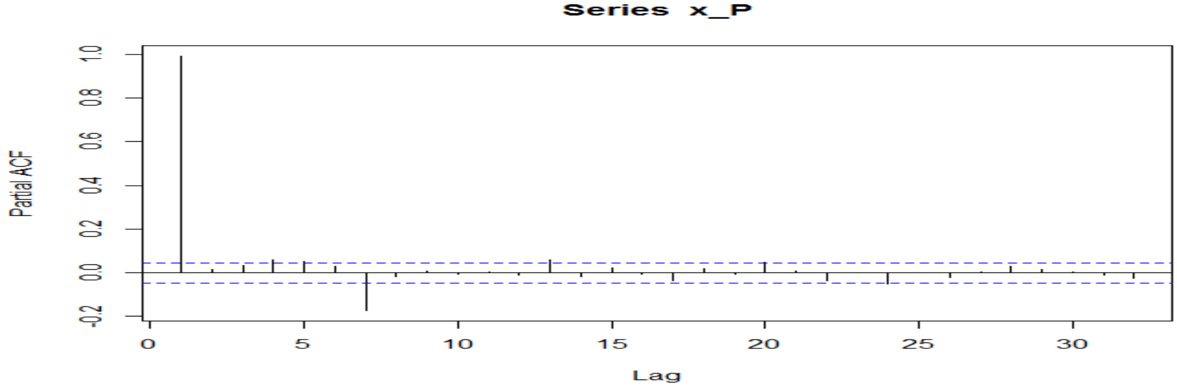
ARIMA Models: Unit Roots 1?

Example 1: **Monthly Stock Price levels** (1871-2020)

```
acf_P <- acf(x_P)
> acf_P
Autocorrelations of series ‘x_p’, by lag

0      1      2      3      4      5      6      7      8      9      10     11
1.000  0.992  0.984  0.977  0.971  0.966  0.961  0.954  0.946  0.938  0.931  0.924
```

```
 12     13     14    15     16    17     18     19    20    21     22    23
0.917  0.911  0.904  0.897  0.891  0.884  0.877  0.871  0.865  0.860  0.854  0.848
  24    25     26    27     28     29     30     31    32
0.841  0.834  0.827  0.821  0.815  0.809  0.803  0.797  0.790
```

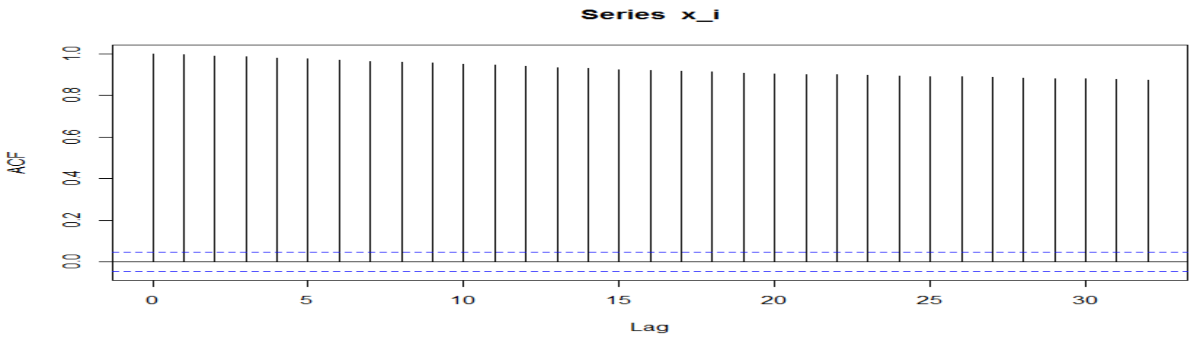


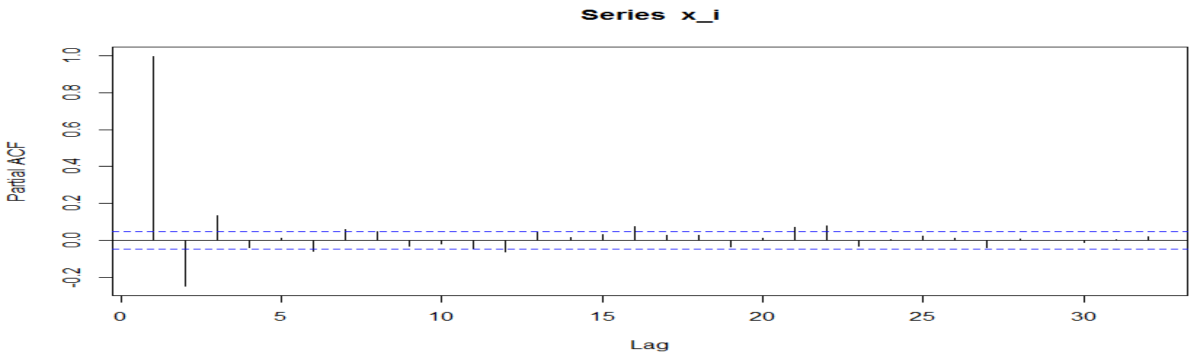


Very high autocorrelations. Looks like $\phi_1 \approx 1$. ¶

Example 2: **Monthly Interest Rates** (1871-2020)

```
acf_i <- acf(x_i)
> acf_i
Autocorrelations of series ‘x_i’, by lag
0        1      2     3      4     5      6      7     8     9      10    11
1.000  0.996  0.990  0.985  0.980  0.975  0.970  0.965  0.960  0.956  0.951  0.946
 12     13     14    15     16    17     18     19    20    21     22    23
0.940  0.934  0.929  0.924  0.919  0.915  0.912  0.908  0.904  0.901  0.899  0.896
 24     25     26    27     28     29     30     31    32
0.894  0.891  0.889  0.887  0.884  0.882  0.879  0.877  0.874
```





Very high autocorrelations. Looks like $\phi_1 \approx 1$. ¶

ARIMA Models – Random Walk

A *random walk* (RW) is defined as a process where the current value of a variable is composed of the past value plus an error term defined as a white noise (a normal variable with zero mean and variance one).

A Random Walk is an ARIMA(0,1,0) process

Popular model. Used to explain the behavior of financial assets, unpredictable movements (Brownian motions, drunk persons).

It is a special case (limiting) of an AR(1) process: a *unit-root* process.

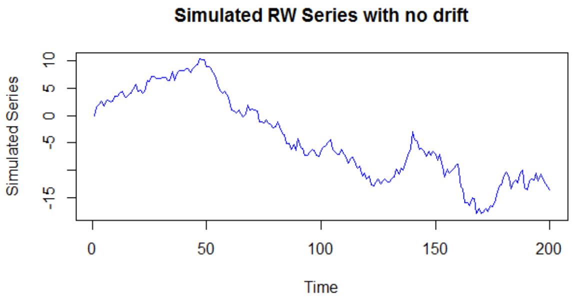
Implication: $E[y_{t+1}|I_t] = y_t$ $\Rightarrow \Delta y_t$ is absolutely random.

Thus, a RW is nonstationary, and its variance increases with *t*.

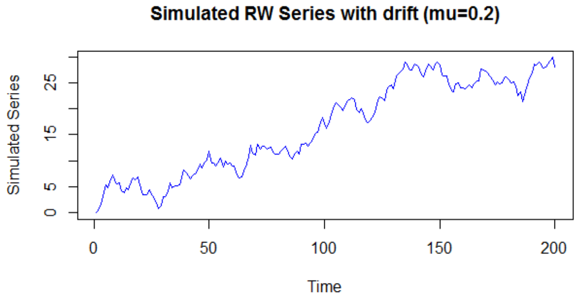
Examples: Two simulated RW

```
T_sim <- 200                    # Sample size for simulation
u <- rnorm(200)                 # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)      # Vector to collect simulated data
phi <- 1                        # Change to create different correlation patterns
a <- 2                          # Time index for observations
```

```
mu <- 0                                    # RW Drift (mu = 0, no drift)
while (a <= T_sim) {
  y_sim[a] = mu + rho * y_sim[a-1] + u[a]  # y_sim simulated autocorrelated values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Simulated RW Series with no drift")
```



Simulated RW Series with no drift



Simulated RW Series with drift (mu=0.2)

Remark: The (stochastic trends) are clear in both graphs. ¶

ARIMA Models – Random Walk with Drift

Change in y_t is partially deterministic (μ) and partially stochastic.

$$y_t - y_{t-1} = \Delta y_t = \mu + \varepsilon_t$$

It can also be written as

$$y_t = y_0 + t\,\mu \; + \sum_{j=0}^{t} \varepsilon_{t-j}$$

$\Rightarrow$ ε_t has a permanent effect on the mean of y_t .

Recall the difference between conditional and unconditional forecasts:

$E[y_t] = y_0 + t\,\mu$ (Unconditional forecast)

$E[y_{t+s}\,|y_t] = y_t + s\,\mu$ (Conditional forecast)

ARIMA Models: Box-Jenkins

An effective procedure for building empirical time series models is the Box-Jenkins approach, which consists of three stages:

(1) Model specification or identification (of ARIMA order)
(2) Estimation
(3) Diagnostics testing.

Two main approaches to (1) Identification.
- *Correlation approach*, mainly based on ACF & PACF.

- *Information criteria*, based on the *maximized likelihood* (x2) plus a *penalty function.* For example, model selection based on the AIC.

Question: We have a family of ARIMA models, indexed by *p, q,* and *d.* How do we select one?
Box-Jenkins Approach provides a method to answer the question.
1) Make sure data is stationary –check a time plot. If not, differentiate.

2) Using ACF & PACF, guess small values for *p* & *q*.
3) Estimate order *p, q*.
4) Run diagnostic tests on residuals.

⇒ Are they white noise? If not, add lags (*p* or *q*, or both).

If order choice not clear, use AIC, AIC Corrected (AICc), BIC, or HQC (Hannan and Quinn (1979)).

Value parsimony. When in doubt, keep it simple (KISS).

ARIMA Models: Identification – Correlations

Correlation approach.

Basic tools: sample ACF and sample PACF.

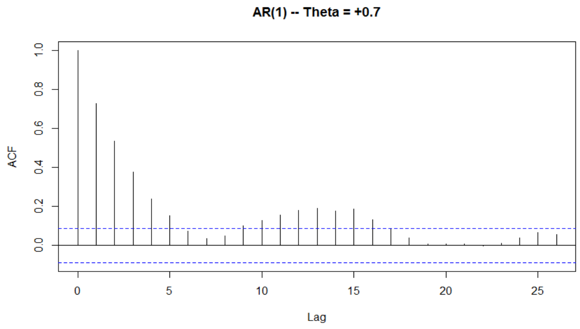
- ACF identifies order of MA: Non-zero at lag *q;* zero for lags > *q*.
- PACF identifies order of AR: Non-zero at lag *p;* zero for lags >*p.*
- All other cases, try ARMA(*p, q*) with *p* > 0 and *q* > 0.

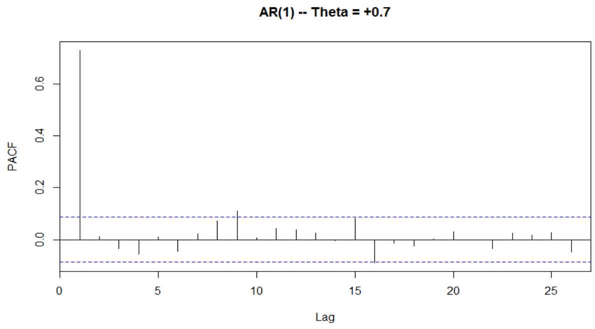
Summary: For *p*>0 and *q*>0.

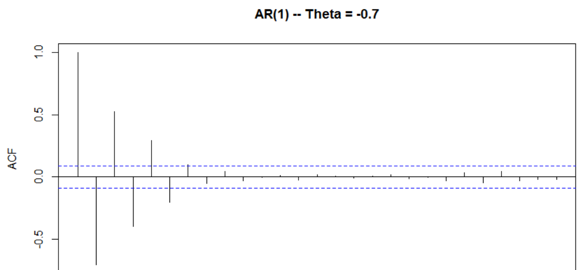
	AR(p)	MA(q)	ARMA(*p*, *q*)
ACF	Tails off	0 after lag q	Tails off
PACF	0 after lag p	Tails off	Tails off

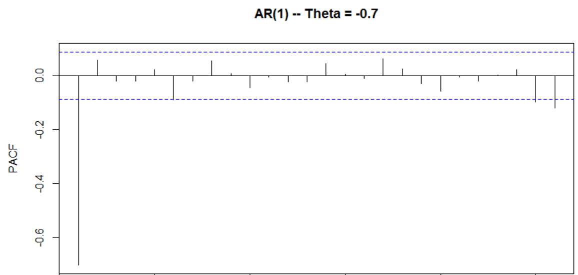
Note: Ideally, “Tails off” is exponential decay. In practice, in these cases, we may see a lot of non-zero values for the ACF and PACF.

ARIMA Models: Identification – AR(1)



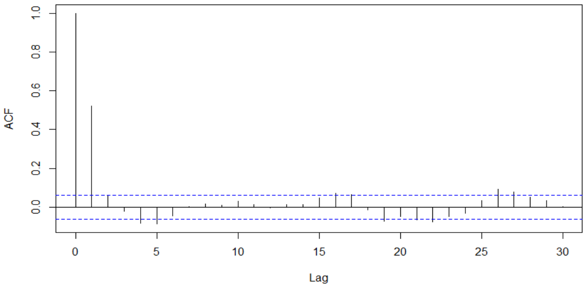




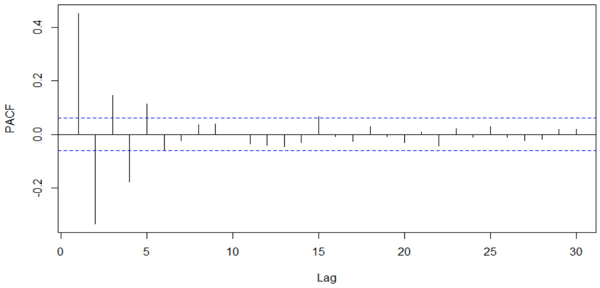


ARIMA Models: Identification – MA(1)

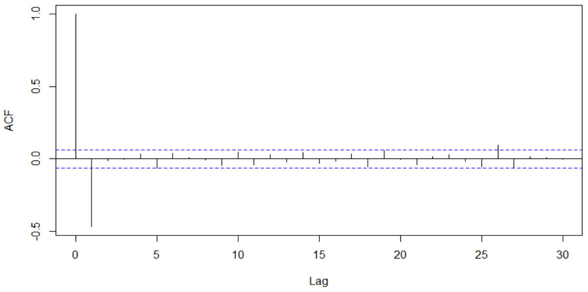
MA(1) -- Theta = +0.7



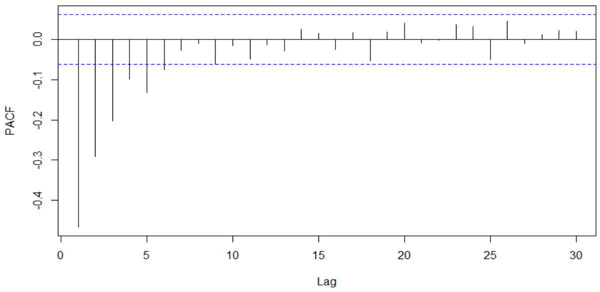
MA(1) -- Theta = +0.7



MA(1) -- Theta = -0.7

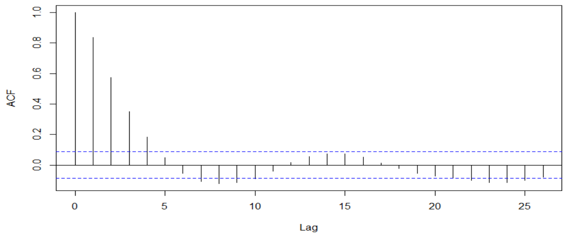


MA(1) -- Theta = -0.7

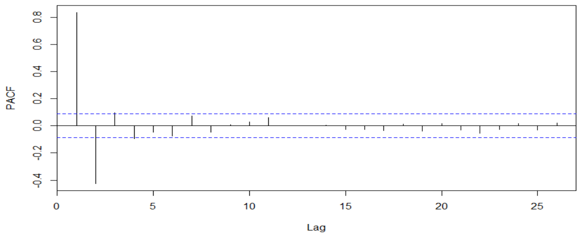


ARIMA Models: Identification – ARMA(1,1)

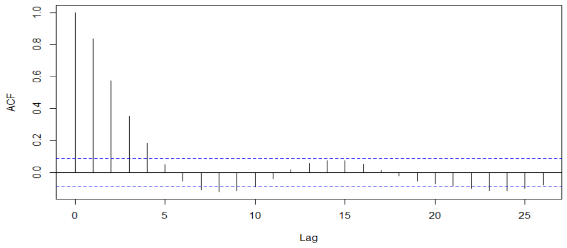
AR(1) -- Phi = 0.7; Theta = 0.5



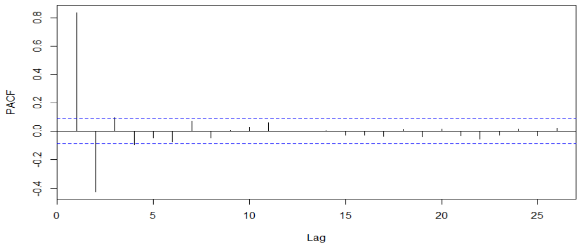
AR(1) -- Phi = 0.7; Theta = 0.5



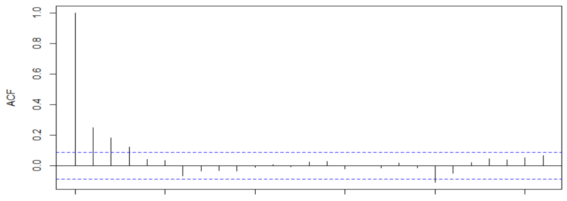
AR(1) -- Phi = -0.7; Theta = -0.5



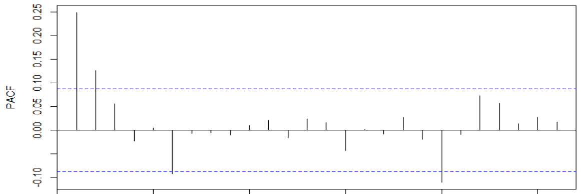
AR(1) -- Phi = -0.7; Theta = -0.5

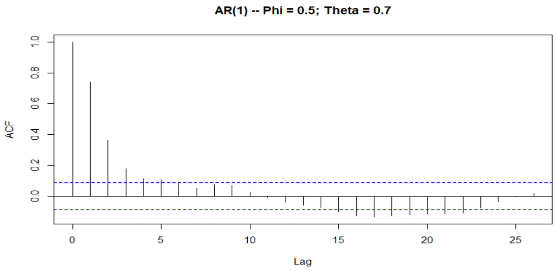


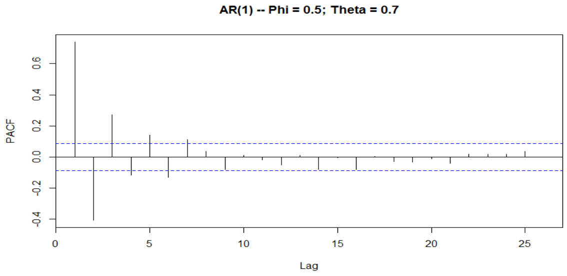
AR(1) -- Phi = 0.7; Theta = -0.5

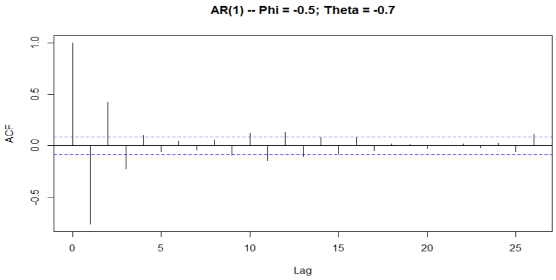


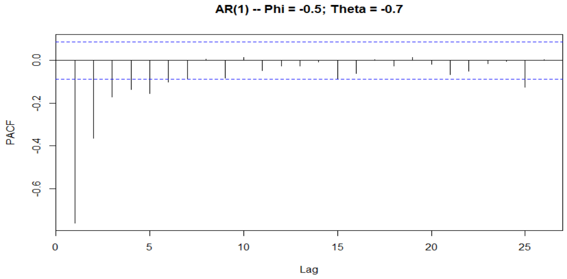
AR(1) -- Phi = 0.7; Theta = -0.5

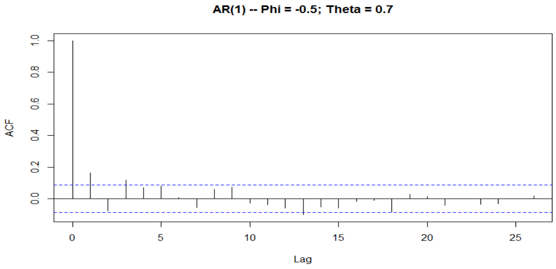


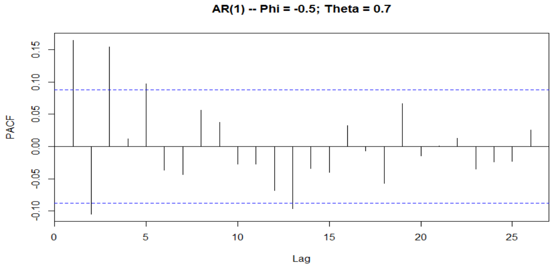






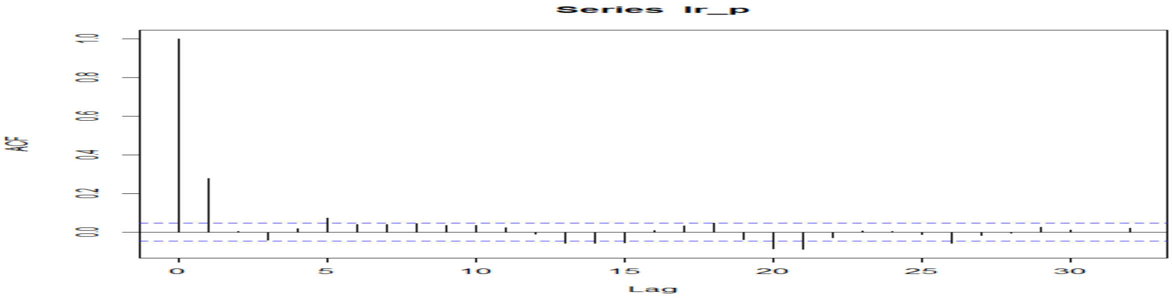


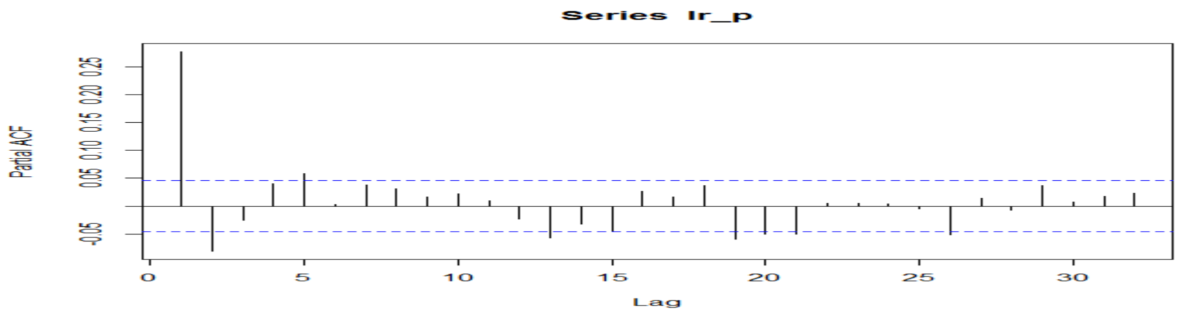




ARIMA Models: Identification – Examples

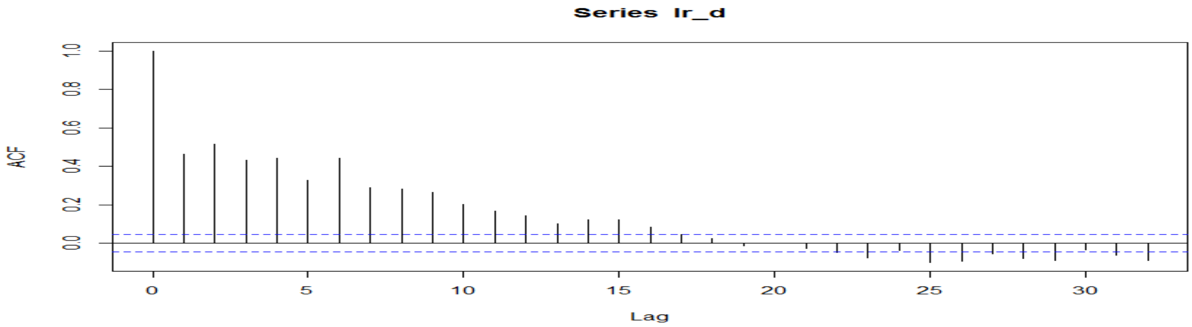
Example 1: **Monthly US Returns** (1871 - 2020).

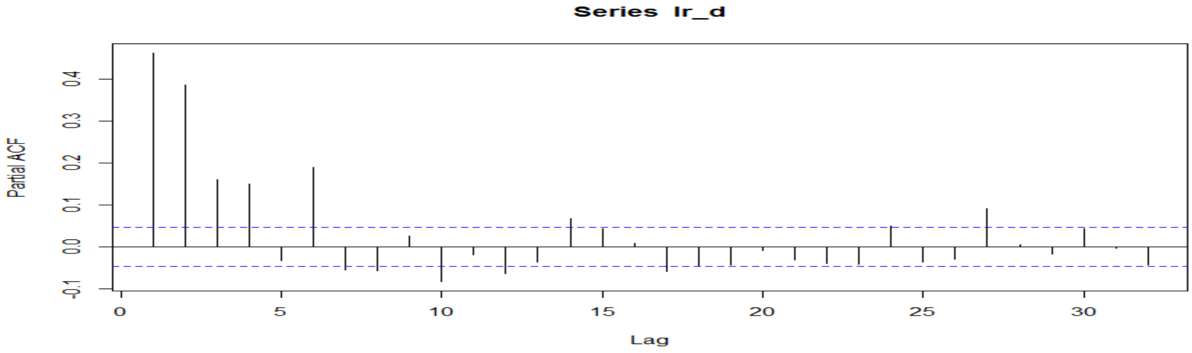




Note: ARMA(1,1), MA(1), AR(2)?

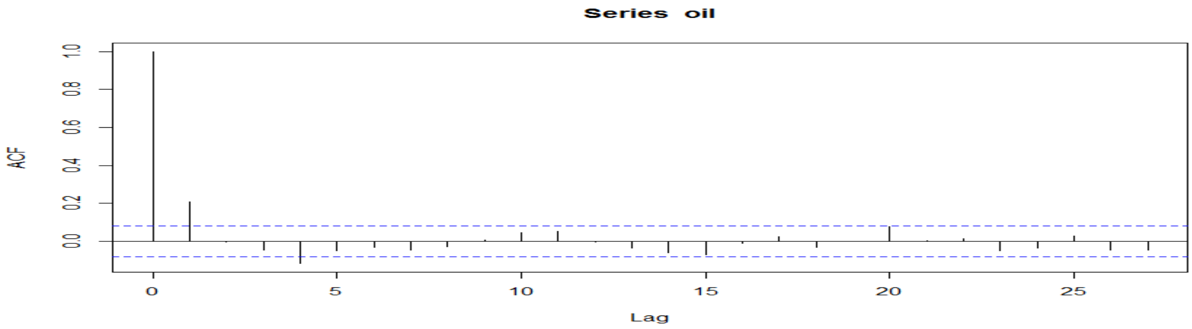
Example 2: **Monthly US Dividend Changes** (1871 - 2020).

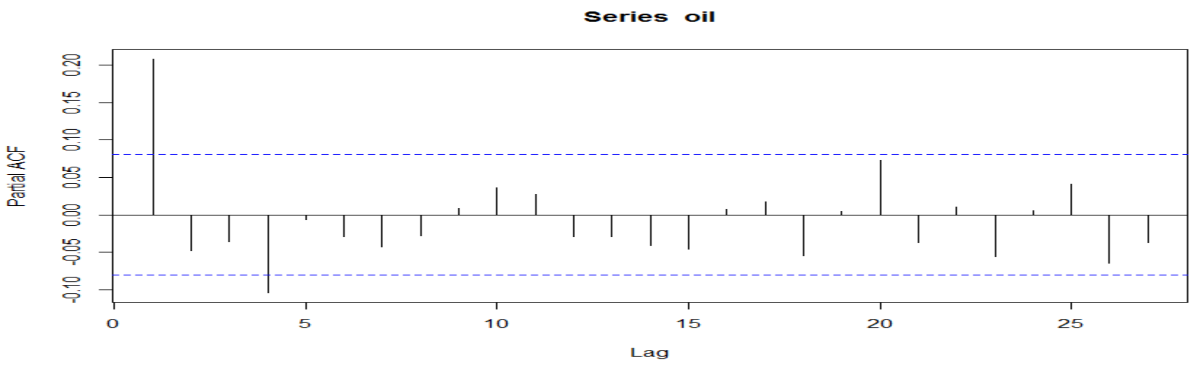




Note: Not clear: Maybe long a ARMA(p,q) or needs differencing? ¶

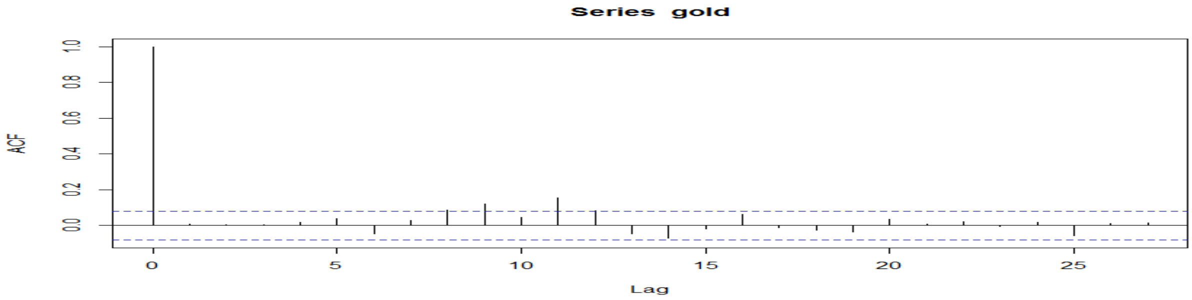
Example 3: **Monthly Log Changes in Oil Prices** (1973 - 2020).

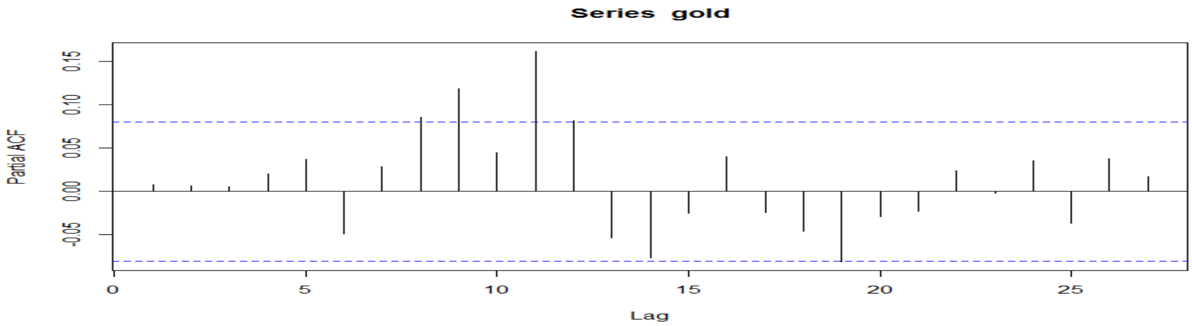




Note: MA(1), AR(4)? ¶

Example 4: **Monthly Log Changes in Gold** (1973 - 2020).





Note: No clear ARMA structure. ¶

ARIMA Model: Identification – IC

It is difficult to identify an ARMA model using the ACF and PACF. It is common to rely on information criteria (IC).

IC's are equal to the estimated variance or the log-likelihood function plus a penalty factor, that depends on k. Many IC's:

- Akaike Information Criterion (AIC)

$$AIC = -2 * (\ln L - k) = -2 \ln L + 2 * k$$

$$\Rightarrow \text{if normality } AIC = T * \ln(\mathbf{e}'\mathbf{e}/T) + 2 * k \qquad (\text{+constants})$$

- Bayes-Schwarz Information Criterion (BIC or SBIC)

BIC = -2 * ln L – ln(T) * k

⇒ if normality AIC = T * ln(**e'e**/T) + ln(T) * k (+constants)

- Hannan-Quinn (HQIC)

HQIC = -2*(ln L – k [ln(ln(T))]

⇒ if normality AIC = T * ln(**e'e**/T) + 2 k [ln(ln(T))] (+constants)

It is very common to compute the IC's under normality (it is the default setting in R and almost all other packages). Recall that under normality, we write the Likelihood function as:

$$\ln L = -\frac{T}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\underbrace{S(\phi p, \theta_q, \mu)}_{Errors\,SS} = -\frac{T}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}T\sigma^2\,\widehat{\ln L} =$$

$$= -\frac{T}{2}\ln\widehat{\sigma}^2 - \underbrace{\frac{T}{2}(1 + \ln 2\,\pi)}_{\text{constant}}$$

Since we compare different ARIMA models, using the same data, the constants play no role in our decision. They can be ignored. Then,

- AIC = T * $\ln(\widehat{\sigma}^2)$ + 2 * k
- BIC = T * $\ln(\widehat{\sigma}^2)$ + ln(T) * k
- HQIC = T * $\ln(\widehat{\sigma}^2)$ + 2 * k * [ln(ln(T))]

The goal of these criteria is to provide us with an easy way of comparing alternative model specifications, by ranking them.

General Rule: The lower the IC, the better the model. For the previous IC's, then choose model to AIC_J, BIC_J, or HQIC.

ARIMA Model: Identification – Remarks

Some remarks about IC's:

- IC's are not test statistics. They do not test a model.
- They are used for ranking. The raw value tends to be ignored.
- They have two components: a *goodness of fit* component –based on *lnL*– and a model complexity component –the penalty based on *k*.
- Different penalties, different IC's.
- Some authors scale the IC's by T. Since raw values tend to be irrelevant, this is not an issue.

We would like these statistics –i.e., the IC's– to have good properties. For example, if the true model is being considered among many, we want the IC to select it. This can be done on average (unbiased) or as T increases (consistent).

Some results regarding AIC and BIC.

- AIC and Adjusted R^2 are not consistent.
- AIC is conservative –i.e., it tends to over-fit: k_{AIC} too large models.
- In time series, AIC selects the model that minimizes the out-of-sample one-step ahead forecast MSE.
- BIC is more parsimonious than AIC. It penalizes the inclusion of parameters more ($k_{BIC} \leq k_{AIC}$).

- BIC is consistent in autoregressive models.
- No agreement which criteria is better.

ARIMA Model: Identification – Small Sample Modifications

• There are modifications of IC to get better finite sample behavior, a popular one is AIC corrected, AICc, statistic:

$$AICc = T\ln\widehat{\sigma}^2 + \frac{2k(k+1)}{T-k-1}$$

AICc converges to AIC as *T* gets large. Using AICc is not a bad idea.

For AR(*p*) models, other AR-specific criteria are possible: Akaike's final prediction error (FPE), Akaike's BIC, Parzen's CAT.

Hannan and Rissannen's (1982) *minic (=Min*imum *IC)*: Calculate the BIC for different *p's* (estimated first) and different *q's*. Select the best model –i.e., lowest BIC.

Note: Box, Jenkins, and Reinsel (1994) proposed using the AIC above.

ARIMA Model: Identification – In practice

Example: Monthly US Returns (1871 - 2020) Hannan and Rissannen (1982)'s minic, based on AIC.

Minimum Information Criterion

Lags	**MA 0**	**MA 1**	**MA 2**	**MA 3**	**MA 4**	**MA 5**
AR 0	-6403.59	-6552.94	-6552.69	-6554.27	-6552.88	-6557.37
AR 1	-6545.22	-6552.23	-6551.86	-6552.42	-6552.64	**-6561.48**
AR 2	-6554.76	-6553.28	-6554.85	-6554.35	**-6564.32**	-6559.48
AR 3	-6553.94	-6552.53	-6554.44	-6552.33	-6550.36	-6558.52
AR 4	-6554.98	-6559.83	**-6559.92**	-6558.94	-6554.1	-6558.16
AR 5	**-6558.81**	-6558.65	-6557.45	-6555.78	-6558.66	-6556.06

Note: Best Model is ARMA(2,4); other potential candidates: ARMA(1,5), ARMA(4,2), ARMA (5,0).

R has a couple of functions that select automatically the “best” ARIMA model: *armaselect* (using package *caschrono*) minimizes BIC and *auto.arima* (using package *forecast*) minimizes AIC, AICc (default) or BIC.

```
> armaselect(lr_p)                              # shows the best 10 models according to BIC
      p q       sbc
 [1,] 2 0 -11644.79
 [2,] 1 0 -11641.53
 [3,] 3 0 -11637.71
 [4,] 4 0 -11632.43
 [5,] 5 0 -11629.95
 [6,] 2 1 -11627.42
 [7,] 6 0 -11621.70
 [8,] 1 3 -11620.18
 [9,] 3 1 -11619.93
[10,] 2 2 -11619.44
```

```
> auto.arima(lr_p, ic="bic", trace=TRUE)                          # ic=“BIC”.
function approximates models.

Fitting models using approximations to speed things up...

 ARIMA(2,0,2) with non-zero mean : -6519.957
 ARIMA(0,0,0) with non-zero mean : -6392.599
 ARIMA(1,0,0) with non-zero mean : -6527.879
 ARIMA(0,0,1) with non-zero mean : -6536.548
 ARIMA(0,0,0) with zero mean     : -6385.246
 ARIMA(1,0,1) with non-zero mean : -6529.358
 ARIMA(0,0,2) with non-zero mean : -6530.806
 ARIMA(1,0,2) with non-zero mean : -6523.415
 ARIMA(0,0,1) with zero mean     : -6534.284

 Now re-fitting the best model(s) without approximations...

 ARIMA(0,0,1) with non-zero mean : -6536.463
```

```
> auto.arima(lr_p, ic="bic", max.p=5, max.q = 5, trace=TRUE)       # approximates models.
Series: lr_p
ARIMA(0,0,1) with non-zero mean
Coefficients:
         ma1    mean
      0.2880  0.0037
s.e.  0.0218  0.0012
sigma^2 estimated as 0.001523:  log likelihood=3279.47
AIC=-6552.94   AICc=-6552.93   BIC=-6536.46
```

Note: The function auto.arima does not try a lot of models, it tries to keep the $p+q \leq 5$. ¶

Remark: Do not take the results from auto.arima or armaselect or minic as the final model. We still need to check the residuals are WN.

• Script in R to select model using *arima* function.

```
p <- 6                                          # set max order for AR part: p-1
q <- 6                                          # set max order for Ma part: q-1
npq <- p*q
aic_m <- matrix(0,nrow = npq, ncol=3)                  # matrix collects p, q, AIC: AIC in last
column
j <- 0
k <- 1
while (j < p) {
i <- 0
while (i < q) {
mod_j <- arima(lr_p, order=c(i,0,j))            # fit arima(p,0,q) process
aic_m[k,] <- cbind(i, j, mod_j$aic)             # extract aic from arima fit model
i <- i + 1
k <- k + 1
}
j <- j + 1
}
aic_m                                           # Print all the results AR(i), MA(j), AIC
min_aic <- min(aic_m[,3])                       # Minimum AIC
min_aic                                                # Print Minimum
which(aic_m == min_aic, arr.ind=TRUE)           # Prints the row
```

ARIMA Model: Identification – Final Remarks

There is no agreement on which criteria is best. The AIC is the most popular, but others are also used.

Asymptotically, the BIC is consistent –i.e., it selects the true model if, among other assumptions, the true model is among the candidate models considered.

The AIC is not consistent, generally producing too large a model, but is more efficient –i.e., when the true model is not in the candidate model set the AIC asymptotically chooses whichever model minimizes the MSE/MSPE.

ARIMA Process – Estimation

We assume:

- The model order (*d*, *p* and *q*) is known. Make sure y_t is I(0).
- The data has zero mean (μ=0). If this is not reasonable, demean y .

Fit a zero-mean ARMA model to the demeaned y_t:

$$\phi(L)(y_t - \bar{y}) = \theta(L)\varepsilon_t$$

Several ways to estimate an ARMA(*p, q*) model:

1) *Maximun Likelihood Esimation* (MLE). Assume a distribution, usually a normal distribution, and, then, do ML.

2) *Yule-Walker for* ARMA*(p,q)*. Method of moments. Not efficient.

3) *Innovations algorithm for MA(q).*

4) *Hannan-Rissanen algorithm for* ARMA(*p, q*).

ARIMA Process – Estimation: MLE

Steps:

1) Assume a distribution for the errors. Typically, *.i.i.d.* normal, say:

$\varepsilon_t \sim$ *i.i.d.* N(0,σ^2)

$\Rightarrow$ pdf: $f(\varepsilon_t) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\varepsilon_t^2}{2\sigma^2}\right]$

2) Write down the joint pdf for $\boldsymbol{\varepsilon}$: $f(\varepsilon_1, ..., \varepsilon_T) = f(\varepsilon_1) ... f(\varepsilon_T)$

Note: We are not writing the joint pdf in terms of the y_t's, as a multiplication of the marginal pdfs because of the dependency in y_t.

3) Get ε_t. For the general stationary ARMA(*p,q*) model:

$$\varepsilon_t = y_t - \phi_1 y_{t-1} - \cdots - \phi p y_{t-p} - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q}$$

(if $\mu \neq 0$, demean y_t.)

4) The joint pdf for {ε_1. ..., ε_T) is:

$$\boldsymbol{\mathcal{L}} \;=\; f\left(\varepsilon_1, \cdots, \varepsilon_T \middle| \mu, \phi, \theta, \sigma^2\right) = \left(2\pi\sigma^2\right)^{-T/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{t=1}^{n} \varepsilon_t^2\right\}$$

5) Let $Y = (y_1, y_2, ..., y_T)$. With an AR(*p, q*) model, we need *p* and *q* initial lags for y_t and ε_t. We assume that initial conditions $Y_* = (y_0, y_{-1}, \dots, y_{-p+1})'$ and $\varepsilon_* = (\varepsilon_0, \varepsilon_{-1}, \dots, \varepsilon_{-q+1})'$ are known.

6) The conditional log-likelihood function is given by

$$\mathcal{L} = \ln \mathrm{L}\left(\mu, \phi, \theta, \sigma^2\right) = -\frac{T}{2}\ln\left(2\pi\sigma^2\right) - \frac{S_*(\mu, \phi, \theta)}{2\sigma^2}$$

where $S_*(\mu, \phi, \theta) = \sum_{t=1}^{n} \varepsilon_t^2(\mu, \phi, \theta | Y, Y_*, \varepsilon_*)$ is the conditional sum of squares (SS).

Note: Usual Initial conditions: $y_* = \bar{y}$ and $\varepsilon_* = E[\varepsilon_t] = 0$.

- Numerical optimization problem, where initial values (**y**$_*$) matter.

Example: AR(1) process:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \overset{i.i.d.}{\sim} N\left(0, \sigma^2\right).$$

- Write down the joint likelihood for $\boldsymbol{\varepsilon}_t$

$$\boldsymbol{\mathcal{L}} \;=\; f(\varepsilon_1, \cdots, \varepsilon_n) = (2\pi\sigma)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{t=1}^{n} \varepsilon_t^2\right\}$$

First, we need to solve for ε_t:

$$Y_1 = \phi_1 Y_0 + \varepsilon_1 \quad \rightarrow \text{Let's take } Y_0 = 0$$
$$Y_2 = \phi_1 Y_1 + \varepsilon_2 \quad \Rightarrow \varepsilon_2 = Y_2 - \phi_1 Y_1$$
$$Y_3 = \phi_1 Y_2 + \varepsilon_3 \quad \Rightarrow \varepsilon_3 = Y_3 - \phi_1 Y_2$$

⋮

$$Y_n = \phi_1 Y_{n-1} + \varepsilon_n \Rightarrow \varepsilon_n = Y_n - \phi_1 Y_{n-1}$$

Technical note: The joint likelihood is in terms of ε_t. We want to change the joint from $\boldsymbol{\varepsilon}_t$ to $\mathbf{y}_t$, for this, we need the Jacobian |J|.

$$|J| = \begin{vmatrix} \frac{\partial \varepsilon_2}{\partial Y_2} & \frac{\partial \varepsilon_2}{\partial Y_3} & \cdots & \frac{\partial \varepsilon_2}{\partial Y_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varepsilon_n}{\partial Y_2} & \frac{\partial \varepsilon_n}{\partial Y_3} & \cdots & \frac{\partial \varepsilon_n}{\partial Y_n} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \cdots & 0 \\ -\phi_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix} = 1$$

Then,

$$f(Y_2, \cdots, Y_n | Y_1) = f(\varepsilon_2, \cdots, \varepsilon_n)|J| = f(\varepsilon_2, \cdots, \varepsilon_n)$$

- Then, the likelihood function can be written as

$$\mathcal{L}(\phi_1, \sigma_a^2) = f(Y_1, \cdots, Y_n) = f(Y_1) f(Y_2, \cdots, Y_n | Y_1) = f(Y_1) f(\varepsilon_2, \cdots, \varepsilon_n)$$

$$= \left(\frac{1}{2\pi\gamma_0}\right)^{1/2} e^{-\frac{(Y_1 - 0)^2}{2\gamma_0}} \left(\frac{1}{2\pi\sigma^2}\right)^{(T-1)/2} e^{-\frac{1}{2\sigma^2} \sum_{t=2}^{T} (Y_t - \phi_1 Y_{t-1})^2},$$

where $Y_1 \sim N\left(0, \gamma_0 = \dfrac{\sigma^2}{1 - {\phi_1}^2}\right)$.

Then,

$$\mathcal{L}(\phi_1, \sigma^2) = \frac{\sqrt{1 - {\phi_1}^2}}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{t=2}^{n} (Y_t - \phi_1 Y_{t-1})^2 + (1 - {\phi_1}^2) Y_1^2\right]\right\}$$

- Then, the log likelihood function:

$$L = \ln \mathcal{L}(\phi_1, \sigma^2) = -\tfrac{n}{2} \ln 2\pi - \tfrac{n}{2} \ln \sigma^2 - \tfrac{1}{2} \ln(1 - {\phi_1}^2) -$$

$$- \tfrac{1}{2\sigma^2} \underbrace{\left[\underbrace{\textstyle\sum_{t=2}^{n} (Y_t - \phi_1 Y_{t-1})^2}_{S_*(\phi_1)} + (1 - {\phi_1}^2) Y_1^2\right]}_{S(\phi_1)}$$

where $S^*(\phi_1)$ is the conditional SS and $S(\phi_1)$ is the unconditional SS.

• F.o.c.'s:

$$\frac{\partial L(\phi_1, \sigma^2)}{\partial \phi_1} = 0$$

$$\frac{\partial L(\phi_1, \sigma^2)}{\partial \sigma} = 0$$

Note:

- If we neglect $ln(1 - {\phi_1}^2)$, then MLE = Conditional LSE.

$$\max_{\phi} L(\phi_1, \sigma^2) = \min S\,(\phi_1).$$

- If we neglect both $ln(1 - {\phi_1}^2)$ and $(1 - {\phi_1}^2) Y_1^2$, then

$$\max_{\phi} L(\phi_1, \sigma^2) = \min S\,({\phi_1}_*).$$ ¶

ARIMA Process – Estimation: Yule-Walker

Yule-Walker for AR(p): Regress y_t against y_{t-1}, y_{t-2} ,. . . , y_{t-p}

- *Yule-Walker for* ARMA*(p, q):* Method of moments. Not efficient.

Example: For an AR(*p*), we the Yule-Walker equations are

$$\begin{bmatrix} \rho(0) & \rho(1) & \cdots & \rho(p-1) \\ \rho(1) & \rho(0) & \cdots & \rho(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & \rho(0) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{bmatrix}$$

Method of Moments (MM) Estimation: Equate sample moments to population moments, and solve the equation. In this case, we use:

$$E(Y_t) = \frac{1}{T}\sum_{t=1}^{T} Y_t \Rightarrow \mu = \bar{Y}$$

$$E[(Y_t - \mu)(Y_{t-k} - \mu)) = \frac{1}{T}\sum_{t=1}^{T}(Y_t - \mu)(Y_{t-k} - \mu)) \quad \Rightarrow \gamma_k = \hat{\gamma}_k \quad (\& \, \rho_k = \hat{\rho}_k)$$

• Then, the Yule-Walker estimator for ϕ is given by solving

$$\begin{bmatrix} 1 & \hat{\rho}(1) & \cdots & \hat{\rho}(p-1) \\ \hat{\rho}(1) & 1 & \cdots & \hat{\rho}(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ \hat{\rho}(p-1) & \hat{\rho}(p-2) & \cdots & 1 \end{bmatrix} \begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} \hat{\rho}(1) \\ \hat{\rho}(2) \\ \vdots \\ \hat{\rho}(p) \end{bmatrix}$$

$$\Rightarrow \hat{\phi} = \hat{R}_{\mathrm{p}}{}^{-1}\hat{\rho}_p \text{ ¶}$$

Note: If $\hat{\gamma}_0 > 0,$ then, $\hat{\Gamma}_m$ is nonsingular.

• If $\{Y_t\}$ is an AR(p) process,

$$\hat{\phi} \xrightarrow{d} N\left(\phi, \frac{\sigma^2}{T}\Gamma_p^{-1}\right)$$

$$\hat{\phi}_{kk} \xrightarrow{d} N\left(0, \frac{1}{T}\right) \text{ for k > p.}$$

• Thus, we can use the sample PACF to test for AR order, and we can calculate approximated C.I. for ϕ.

• Distribution:

If y_t is an AR(p) process, and T is large,

$$\sqrt{T}(\hat{\phi} - \phi) \overset{approx.}{\sim} N(0, \hat{\sigma}^2 \hat{\Gamma}_p^{-1})$$

100(1−α)% approximate C.I. for ϕ_j is

$$\hat{\phi}_j \pm z_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{T}}\left(\hat{\Gamma}_p^{-1}\right)_{jj}^{1/2}$$

Note: The Yule-Walker algorithm requires $\mathbf{\Gamma}^{-1}$.

• For AR(p). The *Levinson-Durbin (LD) algorithm* avoids $\mathbf{\Gamma}^{-1}$. It is a recursive linear algebra prediction algorithm. It takes advantage that $\mathbf{\Gamma}$ is a symmetric matrix, with a constant diagonal (Toeplitz matrix). Use LD replacing γ with $\hat{\gamma}$.

Side effect of LD: automatic calculation of PACF and MSPE.

Example 1: AR(1) (MM) estimation:

$$y_t = \phi_1 y_{t-1} + \varepsilon_t$$

It is known that $\rho_1 = \phi_1$. Then, the MME of ϕ_1 is

$$\Rightarrow \rho_1 = \hat{\rho}_1$$

$$\widehat{\phi_1} = \hat{\rho}_1 = \frac{\sum_{t=1}^{T}(Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_{t=1}^{T}(Y_t - \bar{Y})^2}$$

• Also, σ^2 is unknown: $\gamma_0 = \frac{\sigma^2}{(1-\phi_1{}^2)} \Rightarrow \hat{\sigma}^2 = \hat{\gamma}_0\left(1 - \widehat{\phi_1}^2\right)$. ¶

Example 2: Suppose we suspect an AR(3). We have estimated $\hat{\rho}_1$, $\hat{\rho}_2$, and $\hat{\rho}_3$. Then,

$$\begin{bmatrix} 1 & \hat{\rho}_1 & \hat{\rho}_2 \\ \hat{\rho}_1 & 1 & \hat{\rho}_1 \\ \hat{\rho}_2 & \hat{\rho}_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} \hat{\rho}_1 \\ \hat{\rho}_2 \\ \hat{\rho}_3 \end{bmatrix}$$

Suppose we get: $\hat{\rho}_1$= 0.5, $\hat{\rho}_2$ = 0.4, and $\hat{\rho}_3$ = -0.3. Then, solving for $\boldsymbol{\phi}$:

$$\begin{bmatrix} \hat{\phi}_1 \\ \hat{\phi}_2 \\ \hat{\phi}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.4 \\ 0.5 & 1 & 0.5 \\ 0.4 & 0.5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5 \\ 0.4 \\ -0.3 \end{bmatrix} = \begin{bmatrix} 0.555 \\ 0.511 \\ -0.777 \end{bmatrix}$$

• Solving system with R:

```
Rho <- matrix(c(1, 0.5, 0.4, 0.5, 1, 0.5, 0.4, 0.5, 1), nrow=3)
r <- c(.5, 0.4, -0.3)
solve(Rho)%*%r. ¶
```

Example: MA(1) process with MM estimation:

$$y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

Again using the autocorrelation of the series at lag 1,

$$\rho_1 = -\frac{\theta_1}{(1+\theta_1{}^2)} = \hat{\rho}_1$$

$$\theta_1{}^2 \hat{\rho}_1 + \theta_1 + \hat{\rho}_1 = 0$$

$$\hat{\theta}_{1,2} = \frac{-1 \pm \sqrt{1-4\hat{\rho}_1^2}}{2\hat{\rho}_1}$$

• Choose the root satisfying the invertibility condition. For real roots:

$$1 - 4\hat{\rho}_1^2 \geq 0 \quad \Rightarrow 0.25 \geq \hat{\rho}_1^2 \qquad \Rightarrow -0.5 \leq \hat{\rho}_1 \leq 0.5$$

If $\hat{\rho}_1 = \pm 0.5$, unique real roots but non-invertible.

If $|\hat{\rho}_1| < 0.5$, unique real roots and invertible. ⇒ We keep this one. ¶

• Remarks

- The MMEs for MA and ARMA models are complicated.
- In general, regardless of AR, MA or ARMA models, the MMEs are sensitive to rounding errors. They are usually used to provide initial estimates needed for a more efficient nonlinear estimation method.
- The moment estimators are not recommended for final estimation results and should not be used if the process is close to being nonstationary or noninvertible.

ARIMA Process – Estimation: Yule-Walker – Remarks

The MM estimations for MA and ARMA models are complicated.

In general, regardless of AR, MA or ARMA models, the MMEs are sensitive to rounding errors. They are usually used to provide initial estimates needed for a more efficient nonlinear estimation method.

The moment estimators are not recommended for final estimation results and should not be used if the process is close to being nonstationary or noninvertible.

ARIMA Process – Estimation: Hannan-Rissanen

Hannan-Rissanen algorithm for ARMA(*p,q*)

Steps:

1. Estimate high-order AR.
2. Use Step (1) to estimate (unobserved) noise ε_t

3. Regress y_t against y_{t-1}, y_{t-2}, ..., y_{t-p}, $\hat{\varepsilon}_{t-1}$, ... , $\hat{\varepsilon}_{t-q}$

4. Get new estimates of ε_t. Repeat Step (3).

Example: We estimate a ARIMA(0,0,1) model for **S&P 500 historical returns**, using the *arima* function, part of the R forecast package.

```
> arima(lr_p, order=c(0,0,1), method="ML")                 #ML estimation method
Call:
arima(x = lr_p, order = c(0, 0, 1), method = "ML")
Coefficients:
       ma1  intercept
    0.2880     0.0037
s.e.  0.0218     0.0012
sigma^2 estimated as 0.001522:  log likelihood = 3279.47,  aic = -6552.94. ¶
```

Note: Model was selected by ACF/PACF and confirmed with *auto.arima* function. Not a lot of structure in stock returns.

Example: We use auto.arima function to estimate a model for **DIS, GE**, and **IBM** returns.

```
> auto.arima(lr_dis)
Coefficients:
       ar1    mean
    0.0538  0.0072
s.e.  0.0419  0.0038
sigma^2 estimated as 0.007462:  log likelihood=588.13
AIC=-1170.25   AICc=-1170.21   BIC=-1157.22
> auto.arima(lr_ge)
Coefficients:
       ar1      ma1
    0.0592  -0.9848
s.e.  0.0428   0.0096
sigma^2 estimated as 0.005591:  log likelihood=667.5
```

Note: Very low AR(1) coefficient, and not significant.

```
> auto.arima(lr_ibm)
Series: lr_ibm
ARIMA(0,0,0) with zero mean
sigma^2 estimated as 0.005126:  log likelihood=694.13
AIC=-1386.26   AICc=-1386.25   BIC=-1381.91
sigma^2 estimated as 0.001522:  log likelihood = 3279.47,  aic = -6552.94.
```

Note: Unpredictable! In general, we do not find a lot of structure in stock returns; autocorrelations die out very quickly. This result is expected, given the Efficient Markets Hypothesis. ¶

Example: We use auto.arima function to estimate a model for changes in **oil prices**.

```
> auto.arima(lr_oil)
Series: lr_oil
ARIMA(4,0,0) with zero mean
Coefficients:
       ar1      ar2      ar3      ar4
    0.2950  -0.1024  -0.0570  -0.0984
```

```
s.e.  0.0521   0.0543   0.0551   0.0539

sigma^2 estimated as 0.008913:  log likelihood=344.52
AIC=-679.04   AICc=-678.87   BIC=-659.55
```

Note: AR(4) ⇒ significant autocorrelation in changes in oil prices, but mainly decaying at .30.

Example: We use auto.arima function to estimate a model for monthly **U.S. interest long rates** (1871 – 2020).

```
> auto.arima(x_i)
Series: x_i
ARIMA(0,1,2)
Coefficients:
        ma1      ma2
     0.4012  -0.0957
s.e.  0.0236   0.0238
sigma^2 estimated as 0.02719:  log likelihood=690.02
AIC=-1374.04   AICc=-1374.03   BIC=-1357.56
```

Note: We need to differentiate interest rates to get a stationary MA(2) model. ¶

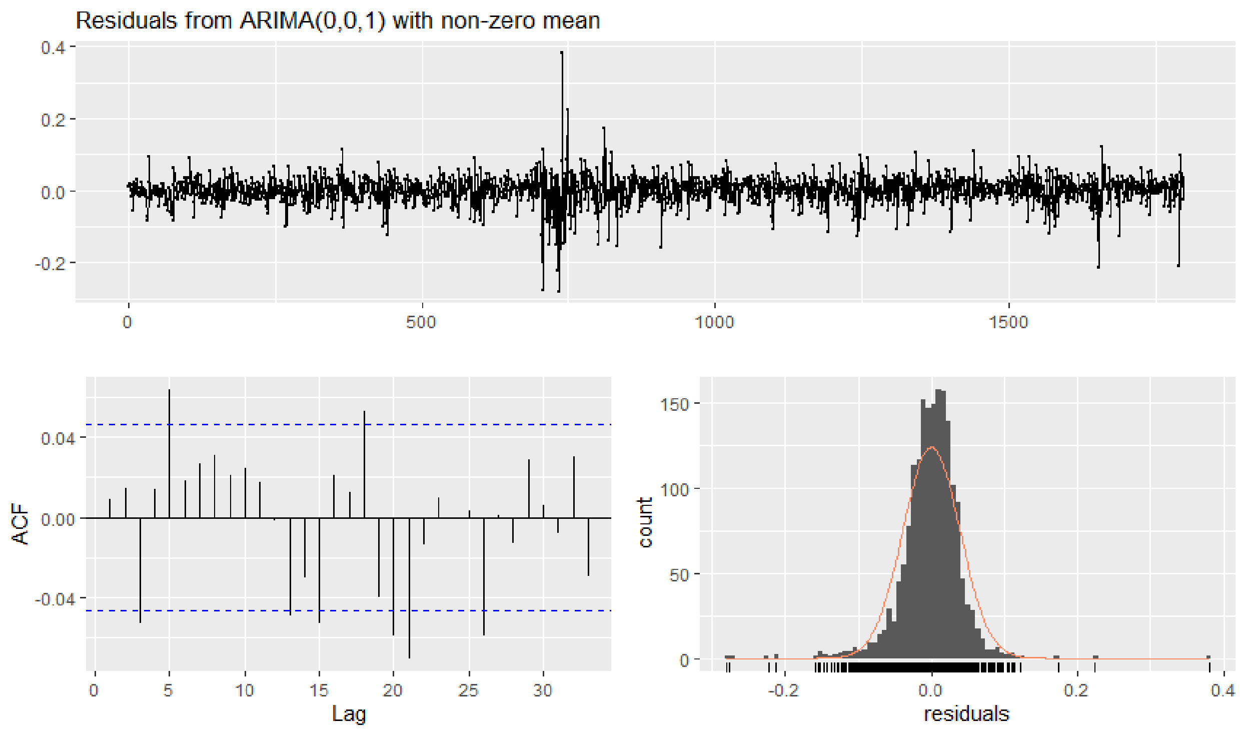
ARIMA Process – Diagnostic Tests

Once the model is estimated, we run diagnostic tests. Usually, we check for extra-AR structure in the mean. We check visual plots of residuals, ACFs, and the distribution of residuals. More formally, we compute the LB test on the residuals. If we find extra-AR structure, we increase p and/or q.

If we use *arima*() or *auto.arima*() functions, we can use the function *checkresiduals*() to do the plots and testing for us.

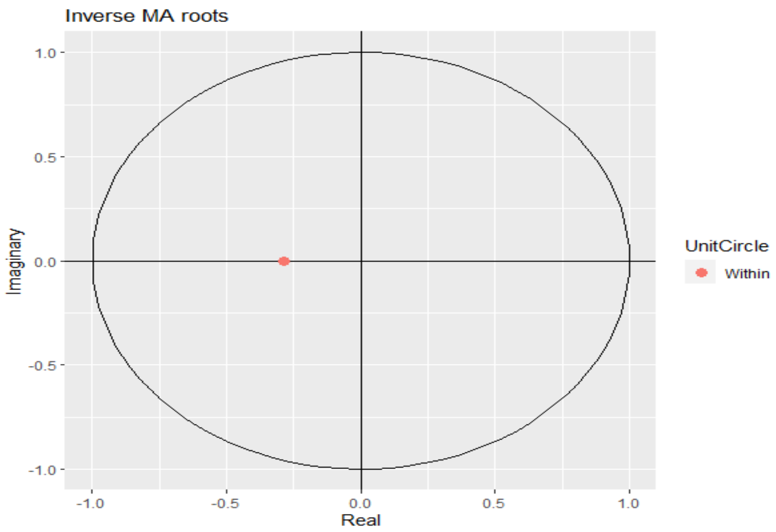
Example: We check the MA(1) model for **U.S. historical stock returns**

```
> fit_arima_lr_p <- arima(lr_p, order=c(0,0,1), method="ML")
> checkresiduals(fit_arima_lr_p)

        Ljung-Box test

data:  Residuals from ARIMA(0,0,1) with non-zero mean
Q* = 18.579, df = 8, p-value = 0.01728                ⇒ There seems to be more AR structure

Model df: 2.   Total lags used: 10
```



We check stationarity/invertibility too -i.e., if the roots are inside the unit circle. In this case, an MA model, stationarity is not an issue (MA are stationary), but invertibility is.

```
> autoplot(fit_arima_lr_p)
```



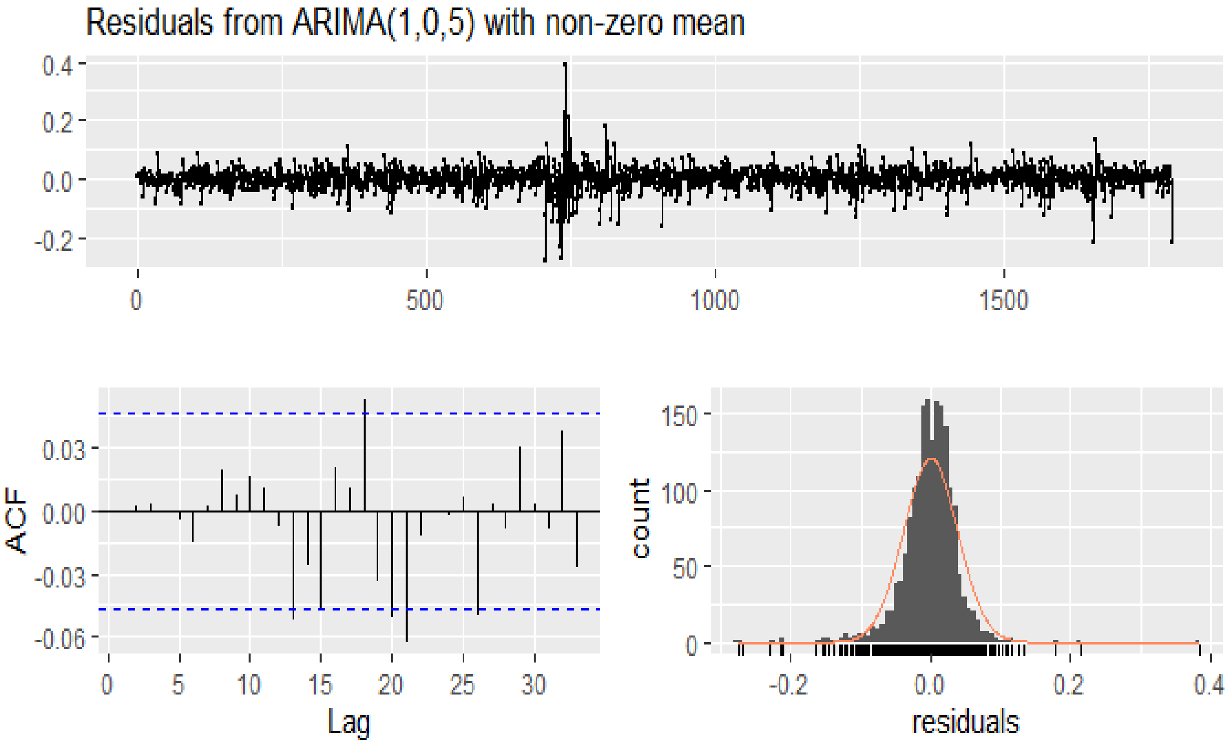
Note: All roots are inside the unit circle and are real: invertible MA(1). ¶

Example: We change the model for **U.S. stock returns**. We estimate an ARIMA(1,0,5).

```
> fit_arima_lr_p15 <- arima(lr_p, order=c(1,0,5))
> fit_arima_lr_p15
```

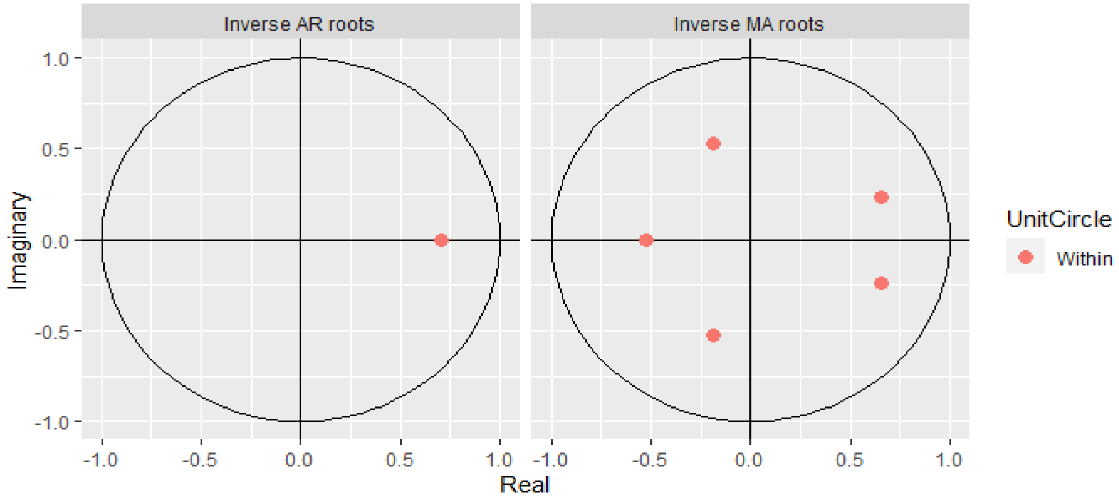
Coefficients:

```
        ar1      ma1       ma2       ma3      ma4      ma5       intercept
     0.7077  -0.4071  -0.1965  -0.0671  0.0338  0.0807     0.0035
s.e.  0.1039   0.1058   0.0392   0.0263  0.0256  0.0250     0.0014

sigma^2 estimated as 0.001502:  log likelihood = 3278.2,  aic = -6540.4

> checkresiduals(fit_arima_lr_p15)
        Ljung-Box test

data:  Residuals from ARIMA(1,0,5) with non-zero mean
Q* = 1.7047, df = 3, p-value = 0.6359                    ⇒ The joint 10 lag autocorrelation not
significant.

Model df: 7.   Total lags used: 10
```



<u>Note</u>: We still see some small autocorrelations different from 0.

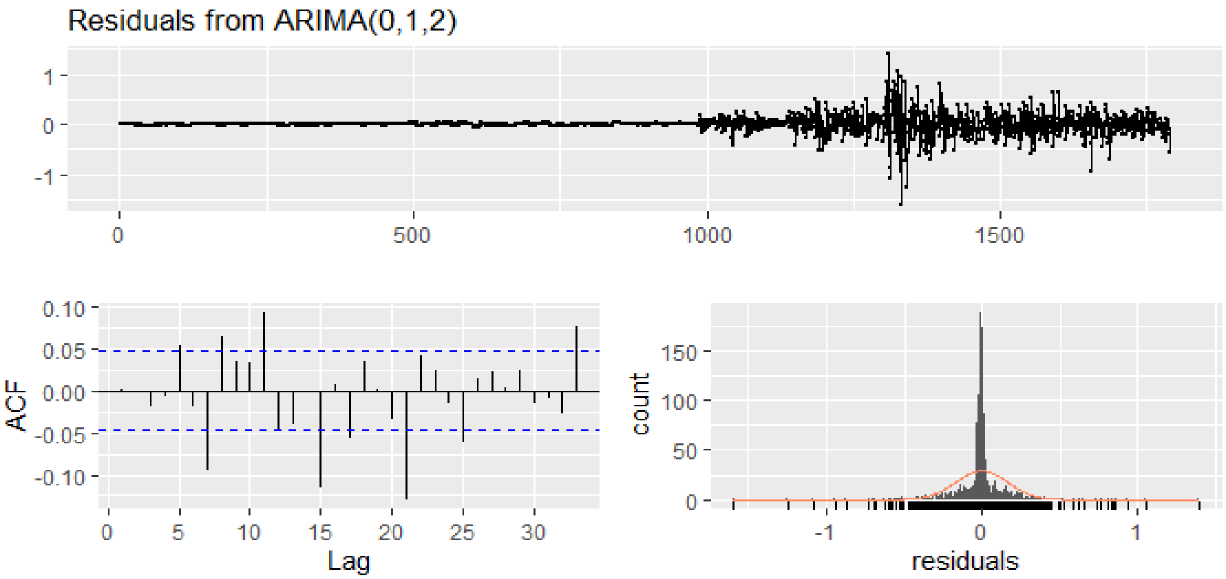
We check the stationarity and invertibility of ARIMA(1,0,5) model
> autoplot(fit_arima_lr_p15)



Note: All roots inside the unit circle: stationary and invertible. Notice that we have some roots on the MA part that are imaginary. ¶

Example: We check the fit of the ARIMA model for **U.S. long interest rates**

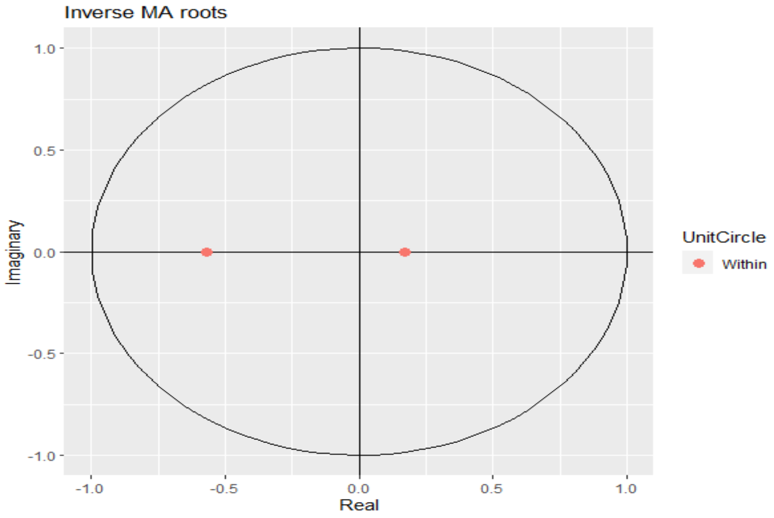
```
> fit_arima_i <- auto.arima(x_i)
ARIMA(0,1,2)
Coefficients:
        ma1      ma2
     0.4012  -0.0957
s.e.  0.0236   0.0238

sigma^2 estimated as 0.02719:  log likelihood=690.02
AIC=-1374.04   AICc=-1374.03   BIC=-1357.56

> checkresiduals(fit_arima_i)

        Ljung-Box test
data:  Residuals from ARIMA(0,1,2)
Q* = 34.029, df = 8, p-value = 4.014e-05      ⇒ Again, more AR or MA structure needed

Model df: 2.   Total lags used: 10
```



Note: We still see some large autocorrelations. ⇒ change model (usually, increase *p* and/or *q)*. But, we may in the presence of a series with regime change. We may need to focus on 2nd regime (post 1950s).

We check the invertibility of ARIMA(0,1,2) model

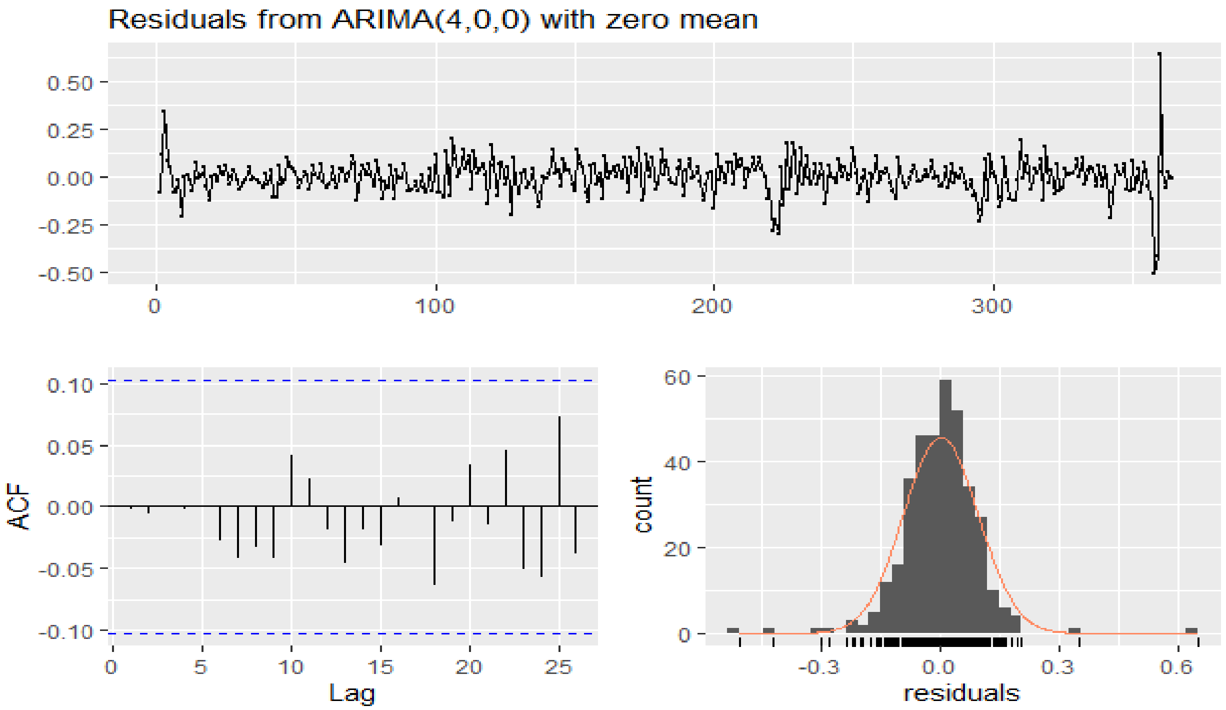
```
> autoplot(fit_arima_i)
```



Note: All roots are inside the unit circle. MA process is invertible. Notice that all roots are real. ¶

Example: We check the fit of the ARIMA(4,0,0) model selected by auto.arima for changes in **Oil Prices.**

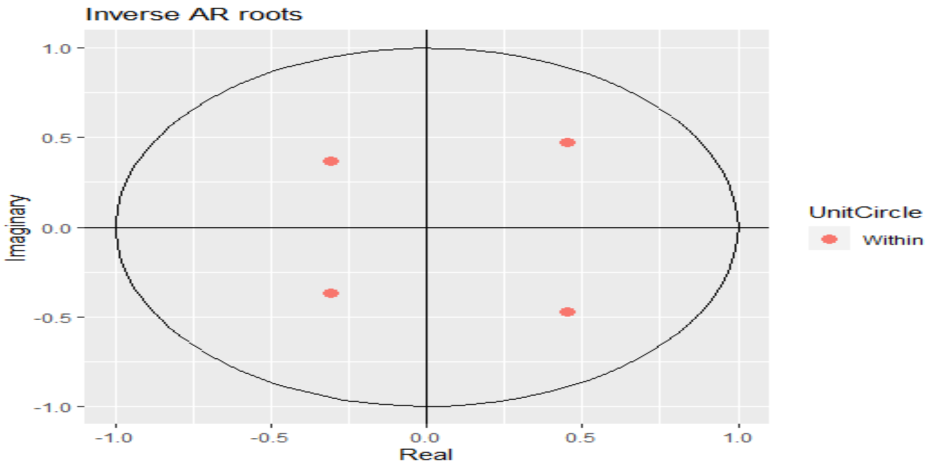
```
fit_arima_oil<- auto.arima(lr_oil)
> fit_arima_oil
Series: lr_oil
ARIMA(4,0,0) with zero mean

Coefficients:
        ar1     ar2     ar3     ar4
      0.295  -0.102  -0.057  -0.098
s.e.  0.052   0.054   0.055   0.054

sigma^2 estimated as 0.00891:  log likelihood=344.52
AIC=-679.04   AICc=-678.87   BIC=-659.55

> checkresiduals(fit_arima_oil)

        Ljung-Box test

data:  Residuals from ARIMA(4,0,0) with zero mean
Q* = 2.72, df = 9, p-value = 0.84                    ⇒ No significant joint AR structure

Model df: 4.   Total lags used: 10
```



Note: Nothing significant. Happy with fit. Ready to forecast.

We check the stationarity of AR(4) model

```
> autoplot(fit_arima_oil)
```



Note: All roots inside the unit circle –we have imaginary roots. ¶

Non-Stationarity in Variance

Stationarity in mean does not imply stationarity in variance. However, non-stationarity in mean implies non-stationarity in variance.

If the mean function is time dependent:
1. The variance, Var(y_t) is time dependent.
2. Var[y_t] is unbounded as $t\rightarrow$ □.
3. Autocovariance functions and ACFs are also time dependent.
4. If t is large with respect to the initial value y_0, then □k □ 1.

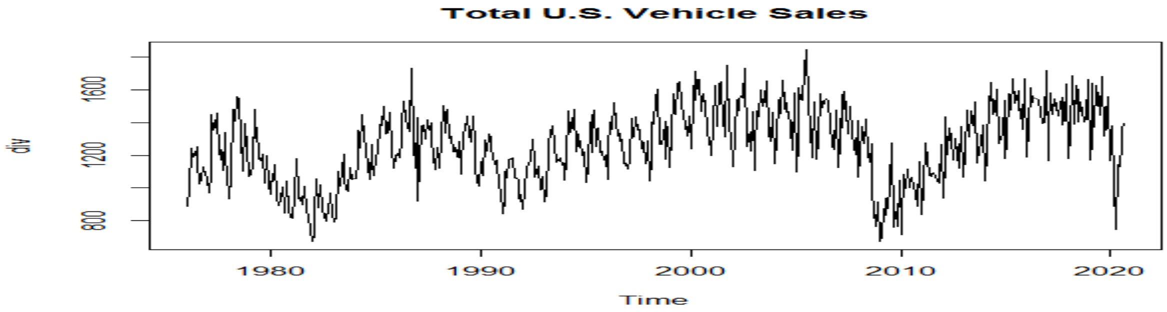
• It is common to use *variance stabilizing* transformations: Find a function $G(.)$ so that the transformed series $G(y_t)$ has a constant variance. Very popular transformation:

1) Log transformation:

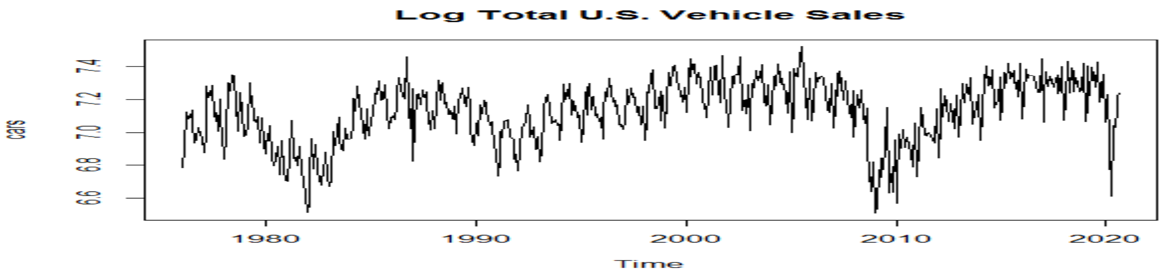
$$G(Y_t) = \log(Y_t)$$

Example: We log transform the monthly variable Total U.S. Vehicle Sales data (1976: Jan – 2020: Sep):

```
ts_car <- ts(x_car,start=c(1976,1),frequency=12)
plot.ts(ts_car,xlab="Time",ylab="div", main="Total U.S. Vehicle Sales")
```



```
l_car <- log(ts_car)
> plot.ts(l_car,xlab="Time",ylab="div", main="Log Total U.S. Vehicle Sales")library(tseries)
```



Note: The volatility is significantly reduced by the log transformation. ¶

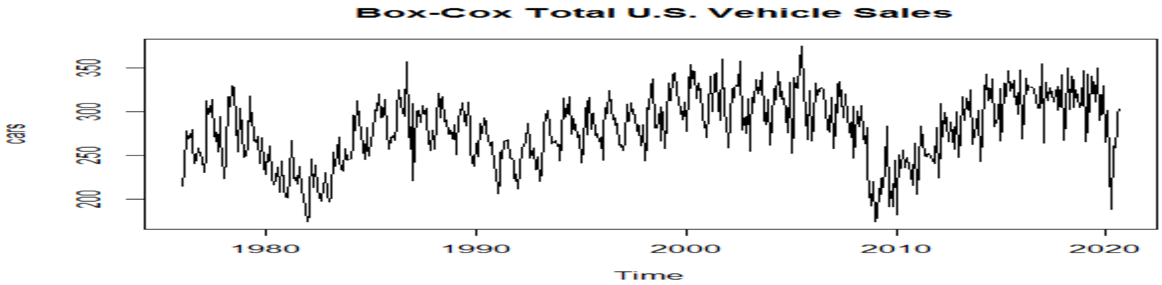
2) Box-Cox transformation:

$$G(Y_t) = \frac{Y_t^{\lambda} - 1}{\lambda}$$

where $\lambda > 0$, usually between 0 and 2 (it can be estimated too). When λ=1, we have a linear *¤y⍰t⍰*; when $\lambda \rightarrow 0$, a log transformation for Y_t.

Example: We do a Box-Cox transformation of the monthly variable Total U.S. Vehicle Sales data (1976: Jan – 2020: Sep), setting λ= 0.75:

```
lambda <- 0.75
b_cox_car <- (ts_car^lambda - 1)/lambda
> plot.ts(b_cox_car, xlab="Time",ylab="cars", main=" Box-Cox Total U.S. Vehicle Sales")
```



Note: Again, we see a reduced volatility. But, different λs will have a different impact on volatility. ¶

- Remarks

Variance stabilizing transformation is only done for positive series, usually for nominal series (say, in USD total retail sales or units, like Total U.S. vehicle sales). If a series has negative values, then we need to add each value with a positive number so that all the values in the series are positive.

Then, we can search for any need for transformation.

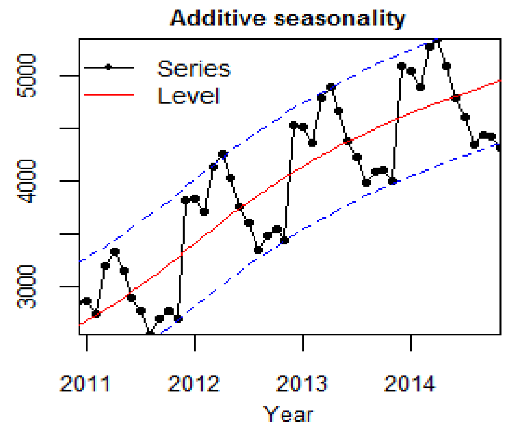
It should be performed before any other analysis, such as differencing.

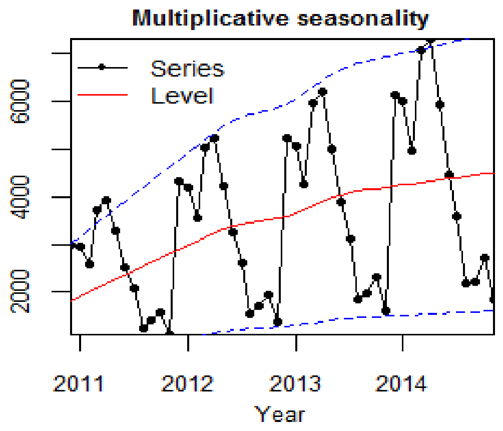
Not only stabilize the variance, but we tend to find that it also improves the approximation of the distribution by Normal distribution.

Seasonal Time Series

In time series, seasonal patterns ("*seasonalities*") can show up in two forms: additive and multiplicative.

- Additive: The seasonal variation is independent of the level.
- Multiplicative: The seasonal variation is a function of the level.

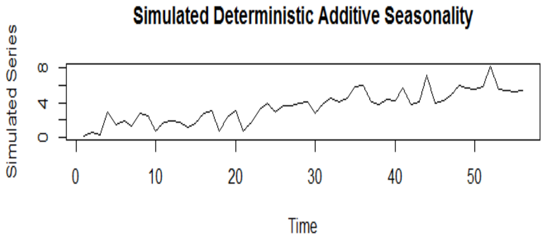


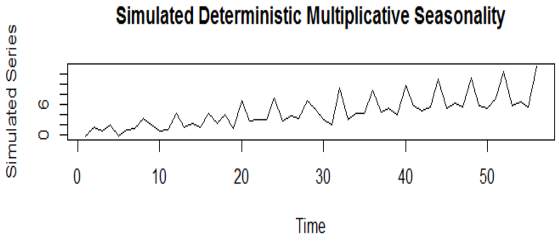


Note: In the multiplicative case, the amplitude of the seasonal pattern is changing over time, while in the additive the amplitude is constant.

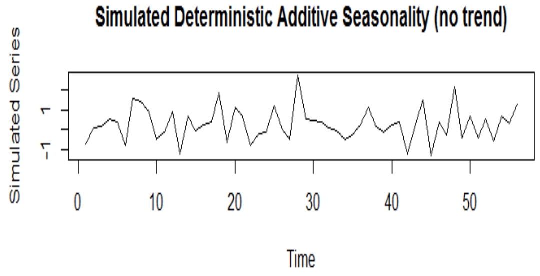
Examples: We simulate the two seasonal patterns, additive and multiplicative, with trend and no trend.

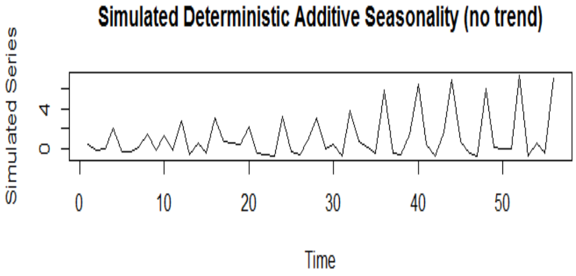
A. With trend





B. With no trend





• In the presence of seasonal patterns, we proceed to do seasonal adjustments to remove these predictable influences, which can blur both the true underlying movement in the series, as well as certain non-seasonal characteristics which may be of interest to analysts.

The type of adjustment depends on how we view the seasonal pattern: Deterministic or Stochastic.

Similar to the situation where the series had a trend, once we determine the nature of the seasonal pattern, we filter the series –i.e., we remove the seasonal patter- to conduct further ARIMA modeling.

When we work with a nominal series (not changes, say, USD total retail sales or total units sold), it is common to first apply a variance stabilizing transformation to the data, usually using logs.

Seasonal Time Series – Types

Two types of seasonal behavior:

- **Deterministic** – Usual treatment: Build a deterministic function,

$$f(t) = f(t + k \times s), \qquad k = 0, \pm 1, \pm 2, \cdots$$

We can include seasonal (means) dummies, for example, monthly or quarterly dummies. (This is the approach in Brooks' Chapter 10).

Instead of dummies, trigonometric functions (sum of cosine curves) can be used. A linear time trend is often included in both cases.

-**Stochastic** – Usual treatment: SARIMA model. For example:

$$y_t = \theta_0 + \Phi_1 \, y_{t-s} + \varepsilon_t + \Theta_1 \varepsilon_{t-s}$$

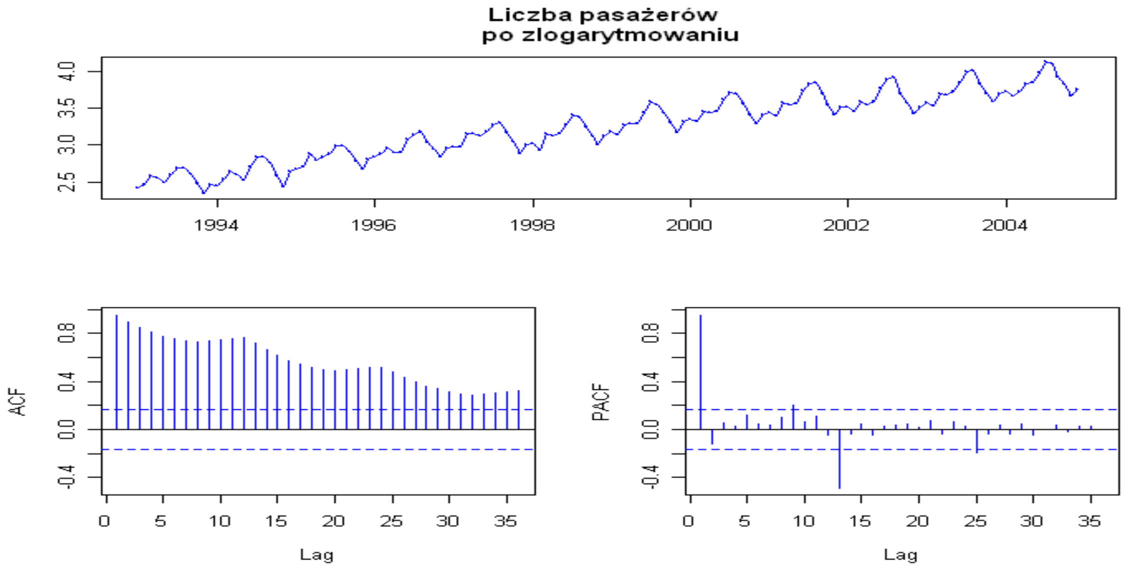
or

$$(1 - \Phi_1 L^s) \, y_t = (1 - \Theta_1 L^s) \, \varepsilon_t$$

where s the seasonal periodicity –associated with the frequency– of y_t. For quarterly data, s = 4; monthly, s= 12; daily, s = 7, etc.

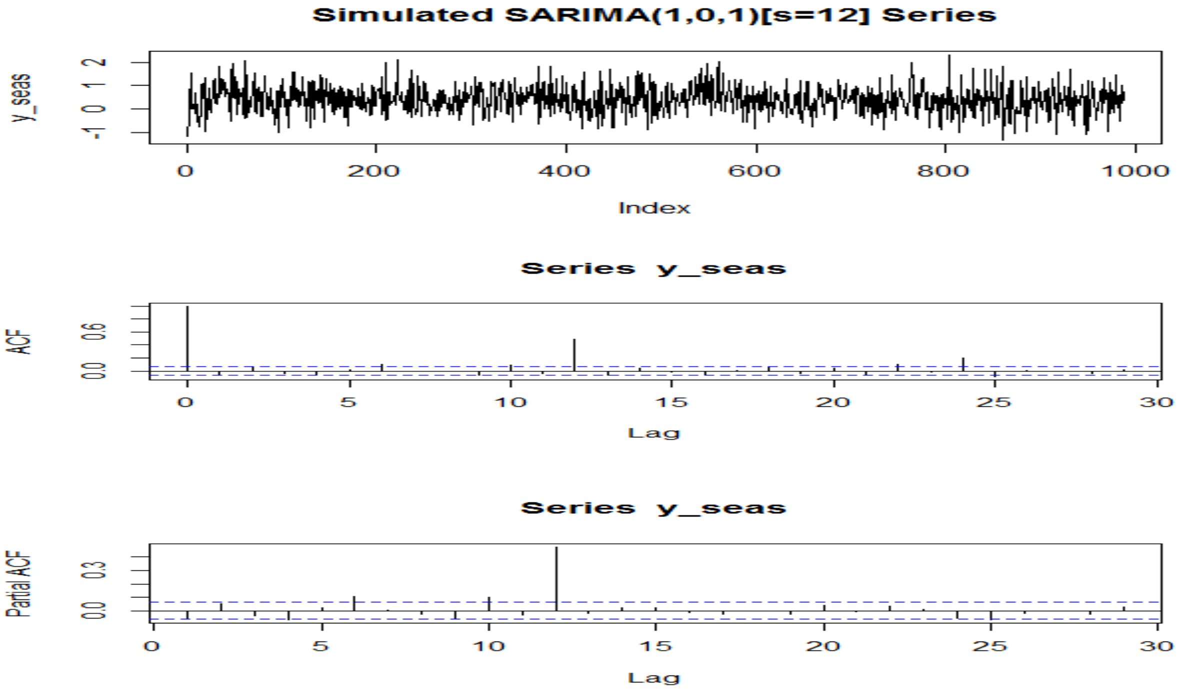
Seasonal Time Series – Finding Seasonality with Visual Patterns

The raw series along with the ACF and PACF can be used to discover seasonal patterns.



<u>Signs</u>: Periodic repetitive wave pattern in ACF, repetition of significant ACFs, PACFs after *s* periods.

- We simulate an ARMA(1,1) with a December seasonal pattern, typical of retail sales with a significant Christmas spike.



Suppose y_t has monthly frequency and we suspect that in every December y_t increases.
– For the additive model, we can regress y_t against a constant and a December dummy, **D**$_t$:

$$y_t = \mu + \boldsymbol{D}_t\boldsymbol{\mu}_s + \varepsilon_t$$

For the multiplicative model, we can regress y_t against a constant and a December dummy, $\mathbf{D}_t$:, interacting with a trend:

$$y_t = \mu + \boldsymbol{D}_t \boldsymbol{\mu}_s * t + \varepsilon_t$$

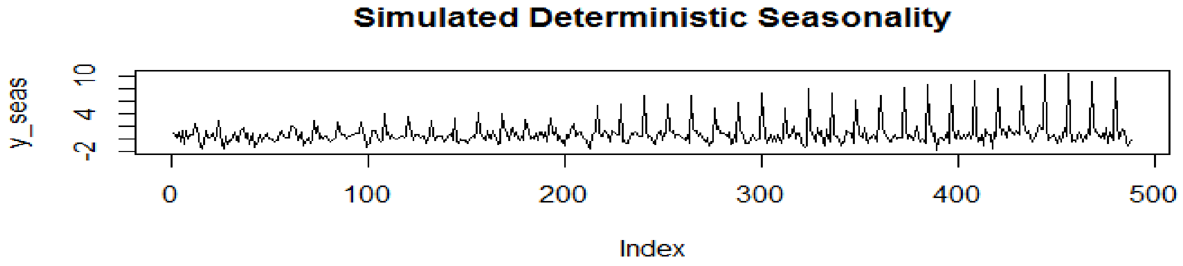
The residuals of this regressions, e_t, –i.e., e_t = filtered y_t, free of "monthly seasonal effects"– are used for further ARMA modeling.

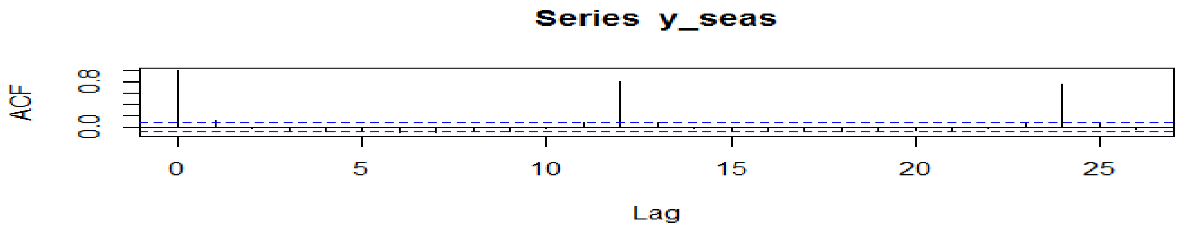
Example: We simulate an AR(1) series, with a multiplicative December seasonal behavior.

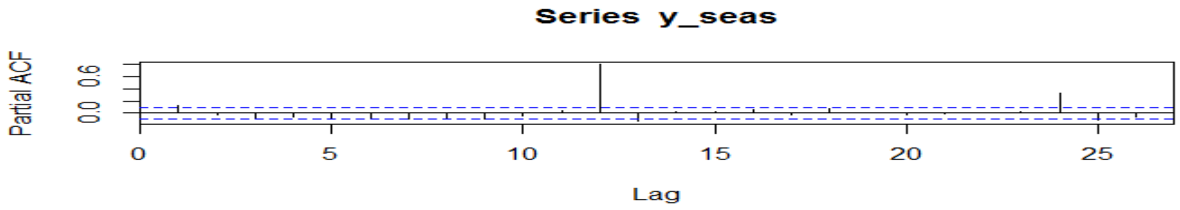
$$y_t = \mu + \phi_1 y_{t-1} + \boldsymbol{D}_t \boldsymbol{\mu}_s * t + \varepsilon_t$$

```
Seas_12 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1), (length(y_sim)/12+1))      # Create Oct dummy
T_sim <- 500
u <- rnorm(T_sim, sd=0.75)                        # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)                        # vector to accumulate simulated data

phi1 <- 0.2                                       # Change to create different correlation
patterns
k <- 12                                           # Seasonal Periodicity
a <- k+1                                          # Time index for observations
mu <- 0.2
mu_s <- .02
while (a <= T_sim) {
        y_sim[a] = mu + phi1 * y_sim[a-1] + Seas_12[a] * mu_s * a + u[a]          # y_sim
simulated autocorrelated values
a <- a + 1
}
y_seas <- y_sim[(k+1):T_sim]
plot(y_seas, type="l", main="Simulated Deterministic Seasonality")
```

We plot simulated series, ACF, & PACF.



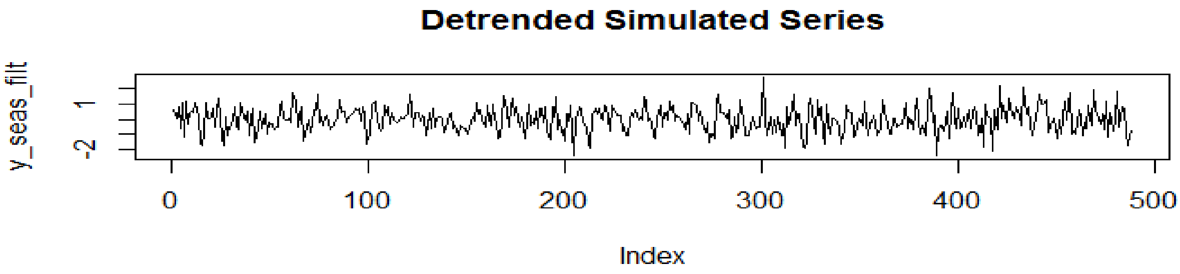


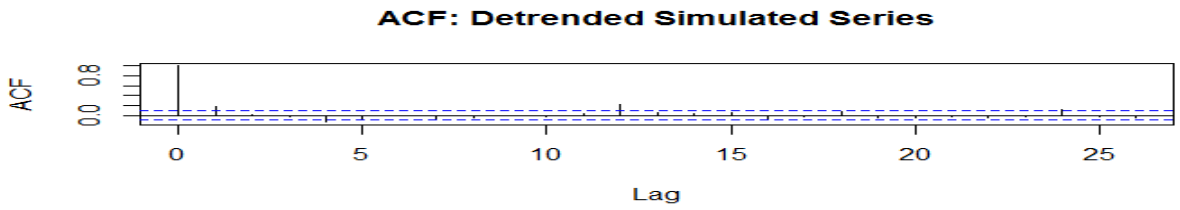


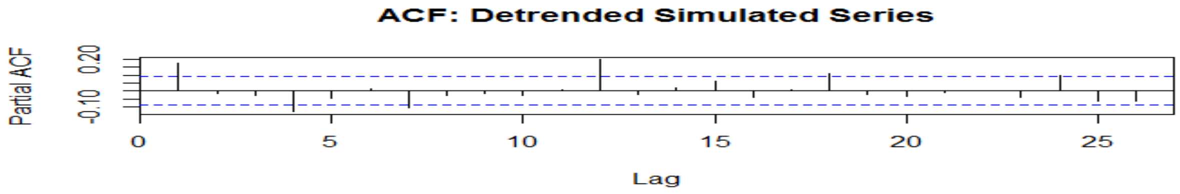
- We detrend ("*filter*" the simulated series).

```
trend <- c(1:T_sim)
trend_sim <- trend[(k+1):T_sim]
sea_trend <- seas_d*trend_sim
fit_seas <- lm(y_seas ~ seas_d + trend_sim + sea_trend)
> summary(fit_seas)

Coefficients:
                Estimate     Std. Error t value Pr(>|t|)
(Intercept)     0.1356538    0.0804474   1.686  0.09239 .
seas_d          0.6929134    0.2859528   2.423  0.01575 *
trend_sim       0.0008504    0.0002749   3.093  0.00209 **
sea_trend       0.0174034    0.0009766  17.821  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8209 on 484 degrees of freedom
Multiple R-squared:  0.7929,    Adjusted R-squared:  0.7917
F-statistic: 617.8 on 3 and 484 DF,  p-value: < 2.2e-16
```

- We plot the detrended simulated series, along with the ACF and PACF.

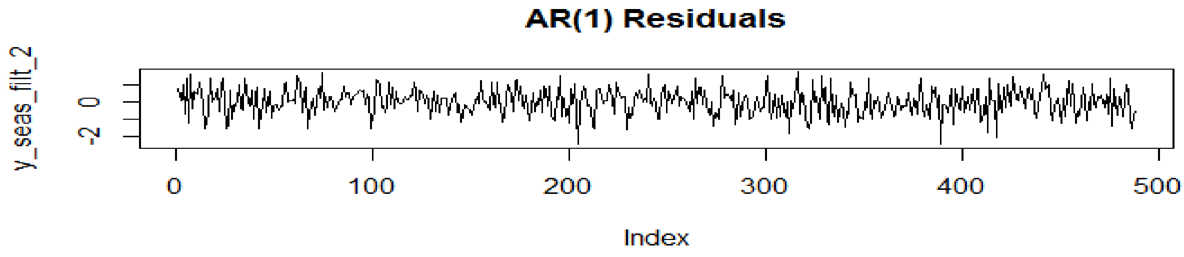


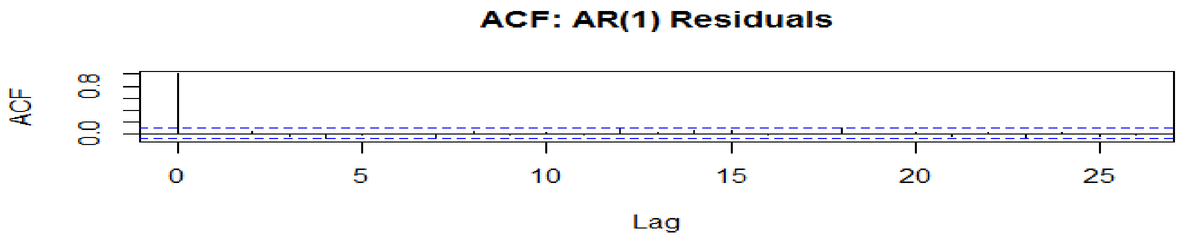


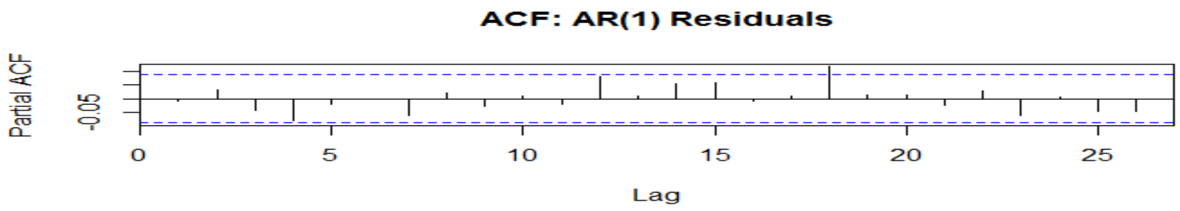


The strong December seasonal pattern is gone from the detrended series. We run an ARIMA(1,0,0):

```
> fit_y_seas_ar1 <- arima(y_seas_filt, order=c(1,0,0))
Call:
arima(x = y_seas_filt, order = c(1, 0, 0))

Coefficients:
        ar1   intercept
     0.1785    -0.0001                  ⇒ Very close to phi1 = 0.20
s.e.  0.0446     0.0443

sigma^2 estimated as 0.6471:  log likelihood = -586.26,  aic = 1178.51

y_seas_filt_2 <- fit_seas_det_ar1$residuals              # Extract Residuals

plot(y_seas_filt_2,type="l", main="AR(1) Residuals")
acf(y_seas_filt_2, main="ACF: AR(1) Residuals")
pacf(y_seas_filt_2, main="ACF: AR(1) Residuals")
```



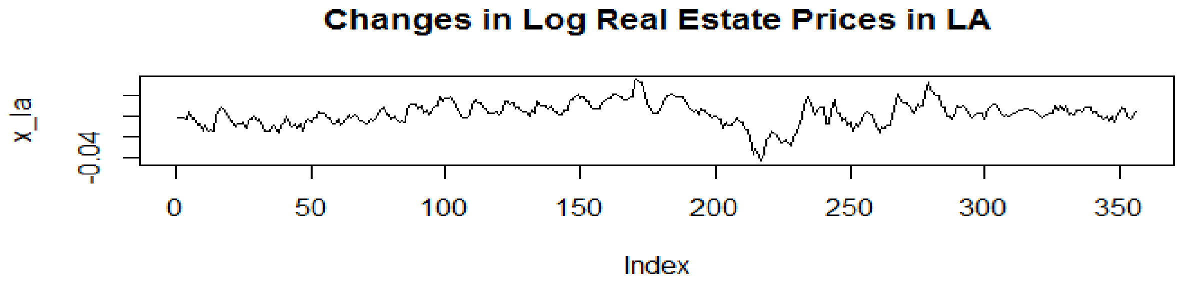




There is no seasonality pattern in the residuals. ¶

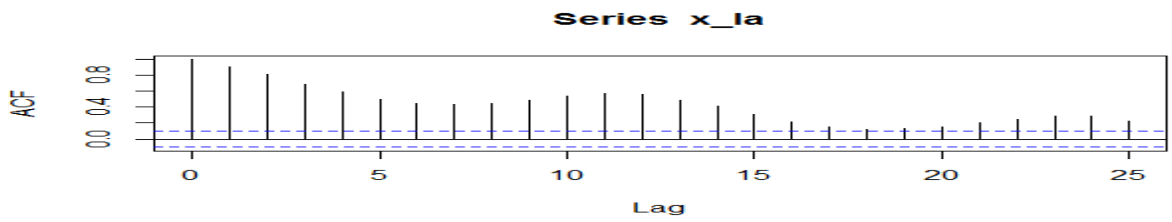
Example: We model **log changes in real estate prices in the LA market**, y_t. First, we run a regression to remove (*filter*) the monthly effects from y_t. Then, we model y_t as an ARMA(*p, q)* process.

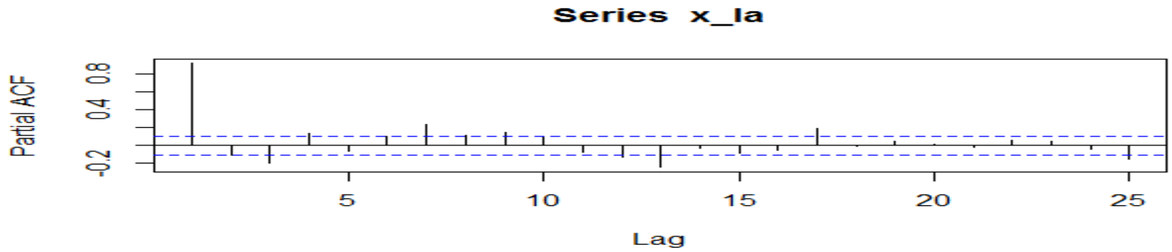
```
RE_da <- read.csv(" http://www.bauer.uh.edu/rsusmel/4397/Real_Estate_2019.csv",
head=TRUE, sep=",")
x_la <- RE_da$LA_c
zz <- x_la
T <- length(zz)
plot(x_la, type="l", main="Changes in Log Real Estate Prices in LA")
```



We look at the ACF & PACF for LA

```
> acf(x_la)
> pacf(x_la)
```





Note: ACF shows highly autocorrelated data, with some seasonal pattern (there is a periodic decreasing wave).

- We define monthly dummies. Then, we regress x_la against the monthly dummies.

```
Feb1 <- rep(c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create January dummy
Mar1 <- rep(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create March dummy
Apr1 <- rep(c(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create April dummy
May1 <- rep(c(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create May dummy
Jun1 <- rep(c(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create June dummy
Jul1 <- rep(c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Jul dummy
Aug1 <- rep(c(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0), (length(zz)/12+1))     # Create Aug dummy
Sep1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0), (length(zz)/12+1))     # Create Sep dummy
Oct1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Nov1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0), (length(zz)/12+1))     # Create Oct dummy
Dec1 <- rep(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0), (length(zz)/12+1))     # Create Oct dummy
seas1 <- cbind(Feb1, Mar1, Apr1, May1, Jun1, Jul1, Aug1, Sep1, Oct1, Nov1, Dec1)
seas <- seas1[1:T,]
x_la_fit_sea <- lm(x_la ~ seas)                    # Regress x_la against constant + seasonal dummies
> summary(x_la_fit_sea)

Coefficients:
                      Estimate        Std. Error t value Pr(>|t|)
(Intercept)     -0.0014063      0.0020125  -0.699 0.485157
seasFeb1         0.0006752      0.0028223   0.239 0.811079
seasMar1         0.0049095      0.0028223   1.740 0.082838 .
seasApr1         0.0090903      0.0028223   3.221 0.001400 **
seasMay1         0.0104159      0.0028223   3.691 0.000260 ***
seasJun1         0.0103464      0.0028223   3.666 0.000285 ***
seasJul1         0.0080593      0.0028223   2.856 0.004557 **
seasAug1         0.0062247      0.0028223   2.206 0.028080 *
seasSep1         0.0032244      0.0028223   1.142 0.254055
seasOct1         0.0011967      0.0028461   0.420 0.674421
seasNov1        -0.0006218      0.0028461  -0.218 0.827181
seasDec1        -0.0009031      0.0028461  -0.317 0.751195
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

<u>Note</u>: Returns –i.e., home prices– are higher from April to August.

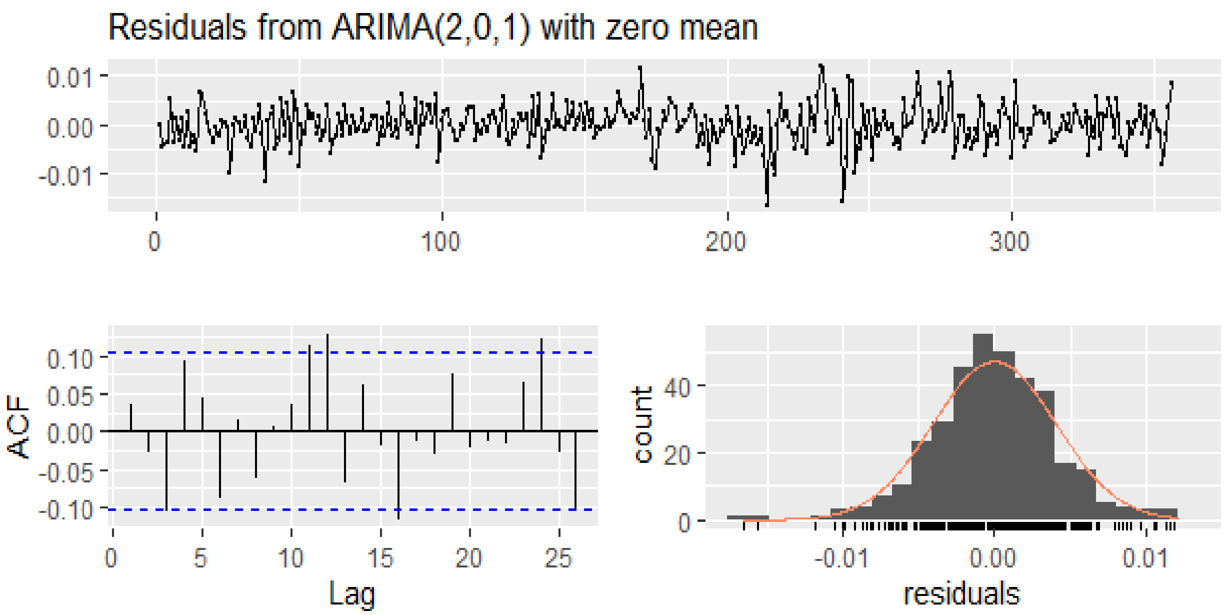
Now, we model e_t, the filtered LA series

```
x_la_filt <- x_la_fit_sea$residuals              # residuals, e_t = filtered x_la series
fit_ar_la_filt <- auto.arima(x_la_filt)          # use auto.arima to look for a good model
> fit_ar_la_filt

Series: x_la_filt
ARIMA(2,0,1) with zero mean

Coefficients:
         ar1     ar2     ma1
      0.0987  0.7737  0.7245
s.e.  0.0963  0.0866  0.1136

sigma^2 estimated as 1.668e-05:  log likelihood=1453.66
AIC=-2899.33   AICc=-2899.21   BIC=-2883.83

> checkresiduals(fit_ar_la_filt)

        Ljung-Box test

data:  Residuals from ARIMA(2,0,1) with zero mean
```

Q* = **13.5**, df = 7, p-value = 0.06083 ⇒ Reject H_0 at 5% lever. But, judgement call is OK.

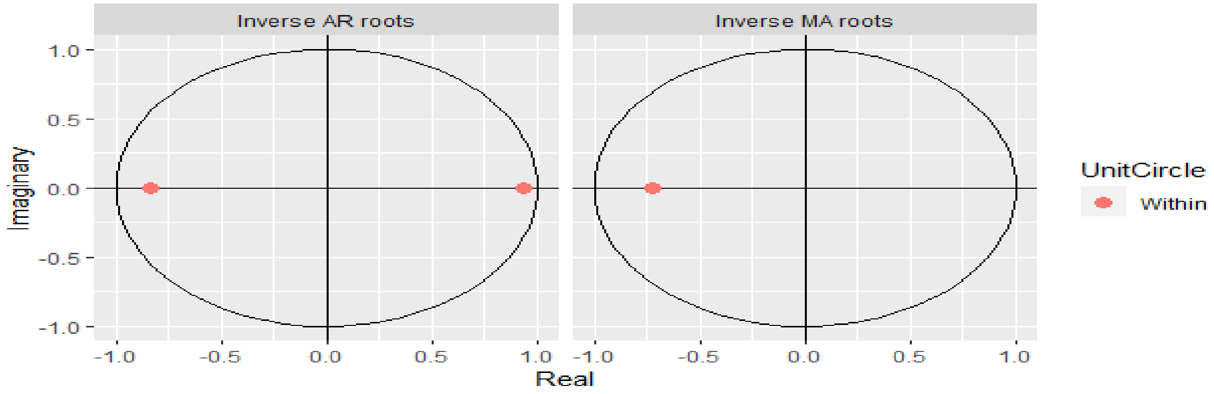
```
Model df: 3.   Total lags used: 10
```

We check residual plots.



Note: ACF shows some small, but significant autocorrelations, but the seasonal (wave) pattern is no longer there.

Finally, we check the stationarity & the invertibility of the ARIMA(2,0,1) process.



Note: All roots inside the unit circle (& real): stationarity and invertibility. ¶

Seasonal Time Series – SARIMA

For stochastic seasonality, we use the Seasonal ARIMA model. In general, we have the SARIMA*(P, D, Q)ₛ*:

$$\Phi_P(L^s)(1-L^s)^D y_t = \theta_0 + \Theta_Q(L^s)\varepsilon_t$$

where θ_0 is constant and

$$\Phi_P(L^s) = 1 - \Phi_1 L^s - \Phi_2 L^{2s} - \cdots - \Phi_P L^{sP}$$
$$\Theta_Q(L^s) = 1 + \Theta_1 L^s + \Theta_2 L^{2s} + \cdots + \Theta_Q L^{sQ}$$

Example 1: SARIMA(0,0,1)$_{12}$= SMA(1)$_{12}$

$$y_t = \theta_0 + \varepsilon_t + \Theta_1 \varepsilon_{t-12}$$

- Invertibility Condition: $|\Theta_1|$< 1.

– E[y_t] = θ$_0$.

– $Var(y_t) = (1 + \Theta_1{}^2)\sigma^2$

– $ACF: \rho_k = \begin{cases} \frac{\Theta_1}{1 + \Theta_1{}^2}, & |k| = 12 \\ 0, & \text{otherwise} \end{cases}$ ⇒ ACF non-zero at seasonal lags 12, 24,...

Example 2: SARIMA(1,0,0)$_{12}$= SAR(1)$_{12}$

$$(1 - \Phi_1 L^{12})y_t = \theta_0 + \varepsilon_t$$

The process is

$$y_t = \theta_0 + \Phi_1 y_{t-12} + \varepsilon_t$$

- This is a simple seasonal AR model.
- Stationarity Condition: $|\Phi_1|$ < 1.

– $E[Y_t] = \frac{\theta_0}{1-\Phi_1}$

– $Var(Y_t) = \frac{\sigma^2}{1-\Phi_1{}^2}$

– ACF: $\rho_{12k} = \Phi_1{}^k, \qquad k = 0, \ \pm 1, \ \pm 2, \cdots$

When Φ_1 = 1, the series is non-stationary. ¶

• Now, we put together the seasonal behavior and the ARMA behavior. That is, we have the multiplicative SARIMA model *(p,d,q)* x *(P,D,Q)$_s$*

Example 1: ARIMA(0,0,1) x (0,0,1)$_{12}$ (usually, with monthly data):

$$y_t = (1 + \theta_1 L)(1 + \Theta L^{12})\varepsilon_t$$

Then, the process is

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \varepsilon_{t-12} + \theta_1 \Theta \varepsilon_{t-12}. ¶$$

Example 2: Suppose *p* = *Q* = 1 and *P* = *q* = 0, with *s*=4, then, we have an ARIMA(1,0,0) x (0,0,1)$_4$ (usually, with quarterly data):

$$(1 - \phi_1 L)\, y_t = (1 + \Theta L^4)\varepsilon_t$$

Then, the process is

$$y_t = \phi_1 y_{t-1} + \varepsilon_t + \Theta \varepsilon_{t-4}. ¶$$

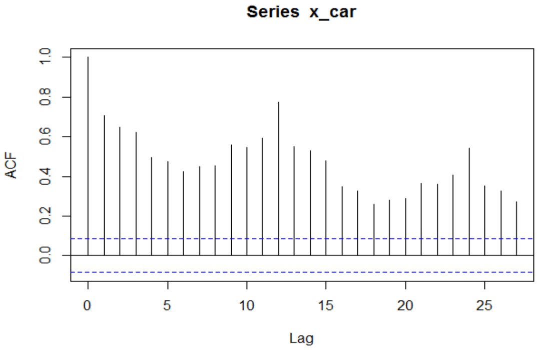
In general, we the multiplicative SARIMA model *(p,d,q)* x *(P,D,Q)$_s$* is written as:

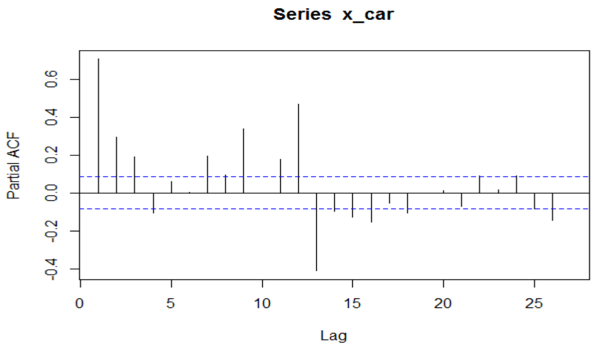
$$\Phi(L)\phi(L)y_t = \theta(L)\Theta(L)\varepsilon_t$$

where $\phi(L)$ is the AR lag polynomial, $\theta(L)$ is the MA lag polynomial, $\Phi(L)$ is the seasonal AR lag polynomial, and $\Theta(L)$ is the seasonal MA lag polynomial.

Example: We model with a SARIMA model for **U.S. vehicle sales**. First, we look at the raw data:

```
Car_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/TOTALNSA.csv", head=TRUE,
sep=",")
x_car <- Car_da$TOTALNSA
> acf(x_car)
```

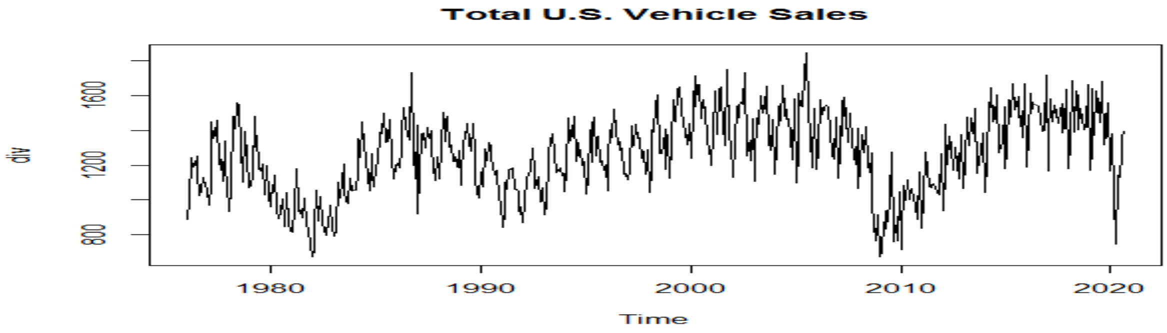




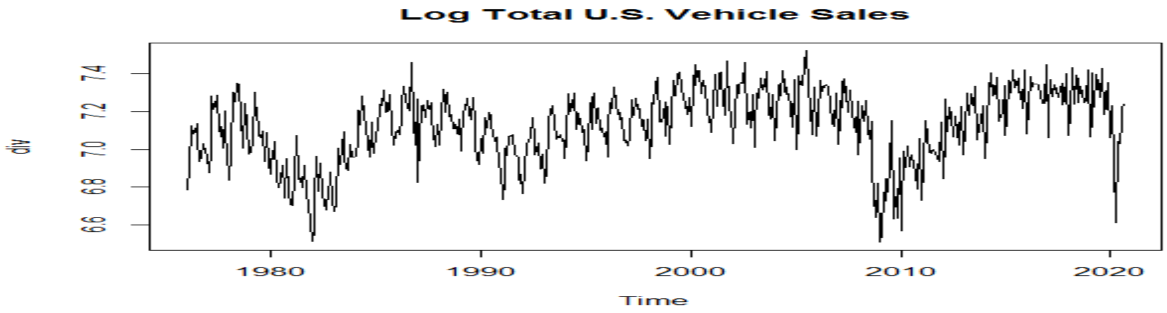
Note: ACF shows a highly autocorrelated data, with some clear seasonal wave pattern.

- Then, we log transform the data:

```
ts_car <- ts(x_car,start=c(1976,1),frequency=12)
plot.ts(ts_car,xlab="Time",ylab="div", main="Total U.S. Vehicle Sales")
```

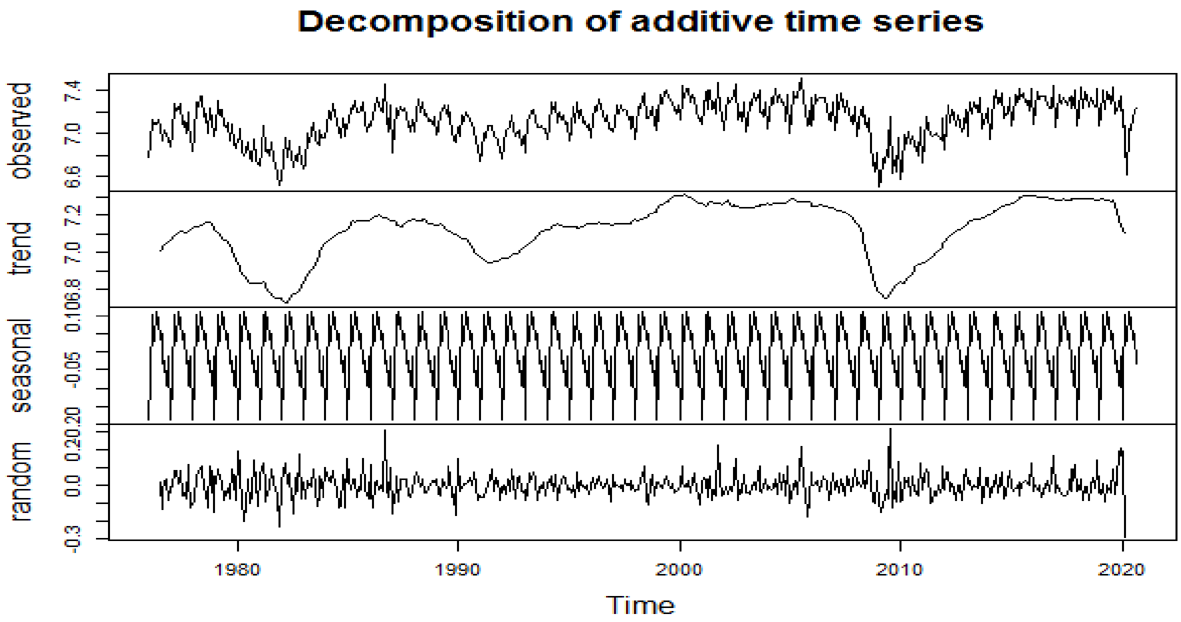


```
l_car <- log(ts_car)
> plot.ts(l_car,xlab="Time",ylab="div", main="Log Total U.S. Vehicle Sales")library(tseries)
```



- R has a function, *decompose*, that decomposes the data in trend, seasonal and random (unexplained):

```
comp_lcar <- decompose(l_car)
> plot(comp_lcar)
```



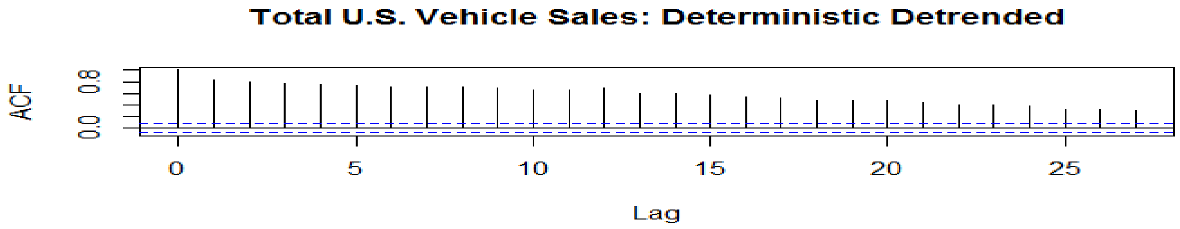
- Question: Should we try deterministic seasonalities?

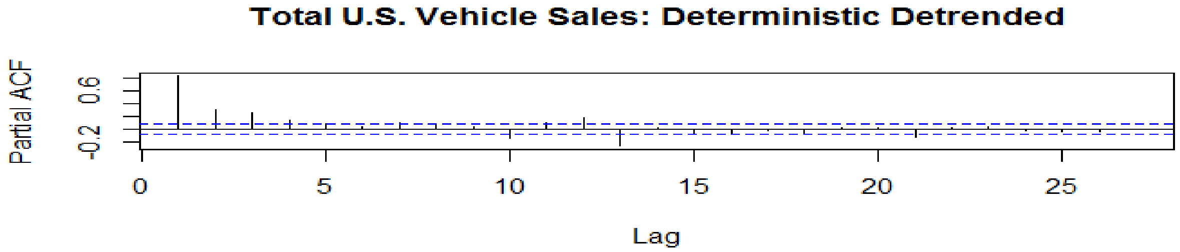
No clear trend in data. We regress l_car against monthly dummies:

```
zz <- l_car
seasd <- cbind(Jan1, Feb1, Mar1, Apr1, May1, Jun1, Jul1, Aug1, Sep1, Oct1, Nov1)
seas_d <- seasd[1:length(zz),]
fit_car_det <- lm(l_car ~ seas_d)
> summary(fit_car_det)

Coefficients:
              EstimateStd. Error t value Pr(>|t|)
(Intercept)   7.03020  0.02564 274.171  < 2e-16 ***
seas_dJan1    0.08235  0.03626   2.271 0.023551 *
seas_dFeb1   -0.09854          0.03606  -2.733 0.006494 **
seas_dMar1    0.01462  0.03606   0.406 0.685259
seas_dApr1    0.19884  0.03606   5.514 5.51e-08 ***
seas_dMay1    0.11396  0.03606   3.160 0.001668 **
seas_dJun1    0.20192  0.03606   5.599 3.47e-08 ***
seas_dJul1    0.17824  0.03606   4.943 1.04e-06 ***
seas_dAug1    0.12804  0.03606   3.551 0.000419 ***
seas_dSep1    0.14824  0.03606   4.111 4.57e-05 ***
seas_dOct1    0.06599  0.03606   1.830 0.067813 .
seas_dNov1    0.07014  0.03626   1.934 0.053638 .
```

- Check ACF and PACF

```
res_car_det <- fit_car_det$residuals
acf(res_car_det)
pacf(res_car_det)
```





- Now, we use auto.arima to check for best SARIMA model:

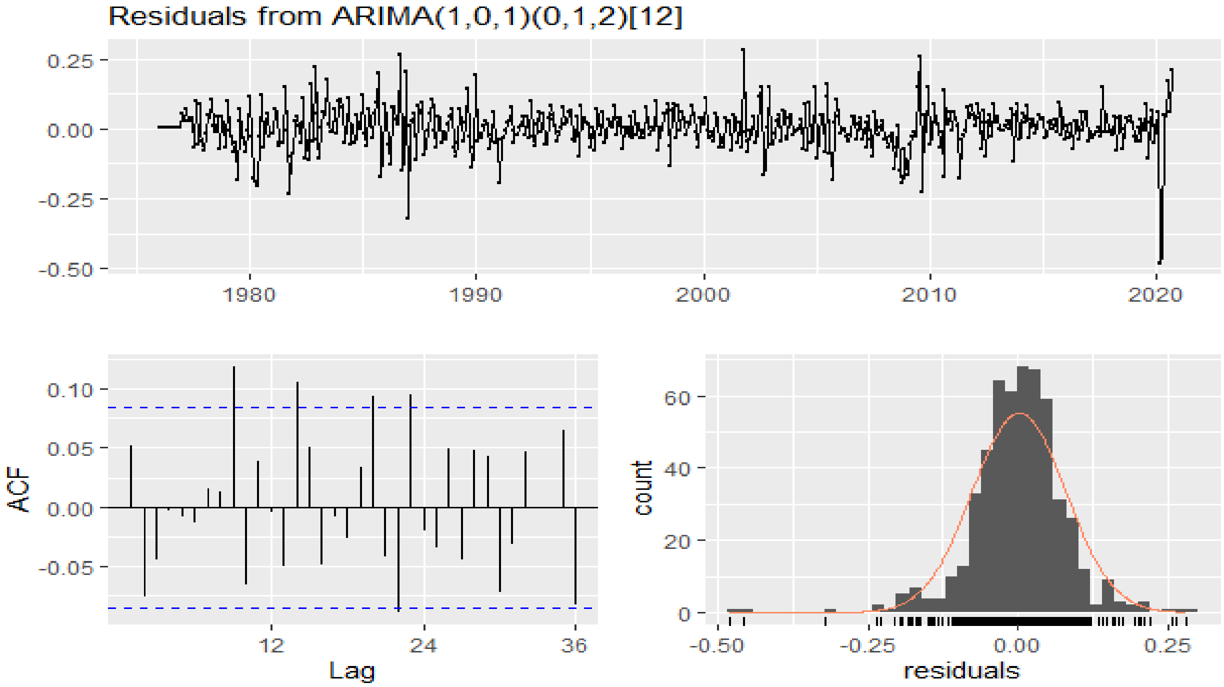
```
> fit_lcar <- auto.arima(l_car, trace=TRUE, ic="bic")

 Fitting models using approximations to speed things up...

 ARIMA(2,0,2)(1,1,1)[12] with drift         : -1049.585
 ARIMA(0,0,0)(0,1,0)[12] with drift         : -609.8308
 ARIMA(1,0,0)(1,1,0)[12] with drift         : -928.3348
 ARIMA(0,0,1)(0,1,1)[12] with drift         : -780.978
...
 ARIMA(2,0,2)(0,1,2)[12] with drift         : -1072.605
 ARIMA(2,0,2)(1,1,2)[12] with drift         : -1055.059
 ARIMA(1,0,2)(0,1,2)[12] with drift         : -1080.563
 ARIMA(0,0,2)(0,1,2)[12] with drift         : -905.0785
 ARIMA(1,0,1)(0,1,2)[12] with drift         : -1081.598
Now re-fitting the best model(s) without approximations...

 ARIMA(1,0,1)(0,1,2)[12]                    : -1132.208

 Best model: ARIMA(1,0,1)(0,1,2)[12]
```

- Check estimated best SARIMA model and check its residuals:

```
> fit_lcar
Series: l_car
ARIMA(1,0,1)(0,1,2)[12]

Coefficients:
         ar1      ma1      sma1      sma2
```

```
     0.9539  -0.5113  -0.5921  -0.2099
s.e.  0.0163   0.0509   0.0464   0.0442

sigma^2 estimated as 0.006296:  log likelihood=581.76
AIC=-1153.52   AICc=-1153.41   BIC=-1132.21

> checkresiduals(fit_lcar)

        Ljung-Box test

data:  Residuals from ARIMA(1,0,1)(0,1,2)[12]
Q* = 44.006, df = 20, p-value = 0.001502

Model df: 4.   Total lags used: 24
```

- Finally, we check residuals, ACF and distribution.



Note: ACF shows small and significant autocorrelation, but the seasonal pattern is gone. More lags maybe needed. ¶

Forecasting

One of the most important objectives in time series analysis is to forecast its future values. It is the primary objective of ARIMA modeling.

Two types of forecasts.

- In sample (prediction): The expected value of the RV (in-sample), given the estimates of the parameters.

- Out of sample (forecasting): The value of a future RV that is not observed by the sample.

To evaluate forecasts, we can use in-sample estimation to learn about the order of the ARMA(*p,q*) model and then use the model to forecast. We do the in-sample estimation keeping a hold-out sample. We use the hold-out sample to validate the selected ARMA model.

Any forecasts needs an information set, *I*T. This includes data, models and/or assumptions available at time *T*. The forecasts will be conditional on *I*T.

The variable to forecast $Y_{T+\ell}$ is a RV. It can be fully characterized by a pdf.

In general, it is difficult to get the pdf for the forecast. In practice, we get a point estimate (the forecast) and a C.I.

<u>Notation</u>:

- Forecast for *T*+ℓ made at *T*: $\hat{Y}_{T+\ell}$, $\hat{Y}_{T+\ell|T}$, $\hat{Y}_T(\ell)$.
- *T*+ℓ forecast error: $e_{T+\ell} = e_T(\ell) = Y_{T+\ell} - \hat{Y}_{T+\ell}$
- Mean squared error (MSE): $MSE(e_{T+\ell}) = E[Y_{T+\ell} - \hat{Y}_{T+\ell}]^2$

To get a point estimate, *¤¤Y⍰⍰T+ℓ⍰*, we need a cost function to judge various alternatives. This cost function is call *loss function.* Since we are working with forecast, we work with a expected loss function.

A popular loss functions is the MSE, which is quadratic and symmetric. We can use asymmetric functions, for example, functions that penalize positive errors more than negative errors.

If we use the MSE as the loss function, we look for $\hat{Y}_{T+l}$, which minimizes it. That is,

$$\min E\,[e^2_{T+\ell}] = E[(Y_{T+\ell} - \hat{Y}_{T+\ell})^2] = E[{Y_{T+\ell}}^2 - 2Y_{T+\ell}\hat{Y}_{T+\ell} + {\hat{Y}_{T+\ell}}^2]$$

Then, f.o.c. implies:

$$E[-2Y_{T+\ell} + 2\hat{Y}_{T+\ell}] = 0 \qquad \Rightarrow E[Y_{T+\ell}] = \hat{Y}_{T+\ell}.$$

The optimal point forecast under MSE is the (conditional) mean:

$$\hat{Y}_{T+\ell} = \mathrm{E}[Y_{T+\ell}|I_T]$$

Different loss functions lead to different optimal forecast. For example, for the MAE, the optimal point forecast is the median.

The computation of E[$Y_{T+\ell}$ |*I*T] depends on the distribution of $\{\varepsilon_t\}$. If $\{\varepsilon_t\}$ ~ WN, then E[$\varepsilon_{T+\ell}$|*I*T] = 0, which greatly simplifies computations, especially in the linear model.

Then, for an ARMA(*p, q)* stationary process (with a Wold representation), the minimum MSE linear forecast (best linear predictor) of $Y_{T+\ell},$ conditioning on IT is:

$$Y_{T+\ell} = \theta_0 + \Psi_l\varepsilon_{T+\ell} + \Psi_{l+1}\varepsilon_{T+\ell-1} + \cdots$$

Forecasting Steps for ARMA Models

The usual process has the following steps:

- ARIMA model: $Y_t = \phi Y_{t-1} + \varepsilon_t$
- Estimation $\widehat{\phi}$ (Estimate of ϕ) $\Rightarrow \hat{Y}_t = \widehat{\phi} Y_{t-1}$ (Prediction)
    (Evaluation in-sample)
- Forecast $\widehat{\mathrm{Y}}_{\mathrm{t+1}} = \widehat{\phi}\widehat{Y}_t$(Forecast)
    (Evaluation out-of-sample)

We observe the time series: $I_T = \{Y_1, Y_2, \ldots, Y_T\}$.

- At time T, we want to forecast: $Y_{T+1}, Y_{T+2}, \ldots, Y_{T+\ell}$.
- T: The forecast origin.
- ℓ: Forecast horizon
- $\hat{Y}_T(\ell)$: *ℓ-step ahead* forecast = Forecasted value $Y_{T+\ell}$

Use the conditional expectation of $Y_{T+\ell}$, given the observed sample.

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell}|Y_T, Y_{T-1}, \ \ldots, Y_1]$$

Example: One-step ahead forecast: $\hat{Y}_{T+1} = E[Y_{T+1}|Y_T, Y_{T-1}, \ \ldots, Y_1]$. ¶

Forecast accuracy to be measured by MSE

$\Rightarrow$ conditional expectation, best forecast.

Forecasting From MA(q) Models

The stationary MA(q) model for Y_t is

$$Y_t = \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q\varepsilon_{t-q}$$

Then, assuming we have the data up to time T ($Y_1, Y_2, \ldots, Y_T, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$) and parameter constancy, we produce at time T *l-step ahead* forecasts using:

$$Y_{T+1} = \mu + \varepsilon_{T+1} + \theta_1\varepsilon_T + \cdots + \theta_q\varepsilon_{T-q+1}$$
$$Y_{T+2} = \mu + \varepsilon_{T+2} + \theta_1\varepsilon_{T+1} + \cdots + \theta_q\varepsilon_{T-q+2}$$
$$\vdots$$
$$Y_{T+l} = \mu + \varepsilon_{T+l} + \theta_1\varepsilon_{T+l-1} + \cdots + \theta_q\varepsilon_{T+l-q}$$

Now, we take conditional expectations:

$$\hat{Y}_{T+l} = E[Y_{T+l}|I_T] = \mu + \mathrm{E}[\varepsilon_{T+l}|I_T] + \theta_1\mathrm{E}[\varepsilon_{T+l-1}|I_T] + \cdots + \theta_q\mathrm{E}\left[\varepsilon_{T+l-q}|I_T\right]$$

Note the forecasts are a linear combination of errors.

Some of the errors are know at time T: $\varepsilon_1 = \hat{\varepsilon}_1$, $\varepsilon_2 = \hat{\varepsilon}_2$, ..., $\varepsilon_T = \hat{\varepsilon}_T$, the rest are unknown. Thus,

$$\mathrm{E}[\varepsilon_{T+j}] = 0 \qquad \text{for } j > 1.$$

Example: For an MA(2) we have:

$$\hat{Y}_{T+1} = \mu + \mathrm{E}[\varepsilon_{T+1}|I_T] + \theta_1\mathrm{E}[\varepsilon_T|I_T] + \theta_2\mathrm{E}[\varepsilon_{T-1}|I_T]$$
$$\hat{Y}_{T+2} = \mu + \mathrm{E}[\varepsilon_{T+2}|I_T] + \theta_1\mathrm{E}[\varepsilon_{T+1}|I_T] + \theta_2\mathrm{E}[\varepsilon_T|I_T]$$

$$\hat{Y}_{T+3} = \mu + \mathrm{E}[\varepsilon_{T+3}|I_T] + \theta_1 \mathrm{E}[\varepsilon_{T+2}|I_T] + \theta_2 \mathrm{E}[\varepsilon_{T+1}|I_T]$$

At time *T*=*t*, we know ε_t and ε_{t-1}. Set $\mathrm{E}\big[\varepsilon_{t+j}|I_t\big]$=0 for *j* > 1. Then,

$$\hat{Y}_{t+1} = \mu + \theta_1 \mathrm{E}[\varepsilon_t|I_t] + \theta_2 \mathrm{E}[\varepsilon_{t-1}|It] = \mu + \theta_1 \hat{\varepsilon}_t + \theta_2 \hat{\varepsilon}_{t-1}$$
$$\hat{Y}_{t+2} = \mu + \theta_2 \mathrm{E}[\varepsilon_t|It] = \mu + \theta_2 \hat{\varepsilon}_t$$
$$\hat{Y}_{t+3} = \mu$$
$$\hat{Y}_{t+l} = \mu \text{for } l > 2. \qquad \Rightarrow \text{MA(2) memory of 2 periods. ¶}$$

The example generalizes: An MA(*q*) process has a memory of only *q* periods. All forecasts beyond *q* revert to the unconditional mean, *μ*.

Example: We fit an MA(1) to the U.S. stock returns (T=1,975):

```
library(tseries)
library(forecast)
fit_p_ts <- arima(lr_p, order=c(0,0,1))                #fit an MA(1) model
fcast_p <- forecast(fit_p_ts, h=4)                      #produce 4-step ahead forecasts
> fit_p_ts
> fcast_p
Coefficients:
       ma1  intercept
     0.2888     0.0037
s.e.  0.0218     0.0012

sigma^2 estimated as 0.001522:  log likelihood = 3275.83,  aic = -6545.67

> fcast_p
    Point Forecast       Lo 80       Hi 80       Lo 95      Hi 95
1796    0.012570813 -0.03742238 0.06256401 -0.06388718 0.08902881
1797    0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
1798    0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
1799    0.003689524 -0.04834634 0.05572539 -0.07589247 0.08327152
```

Remark: After the first forecast, the MA(1) process generates constant forecasts. ¶

Forecasting From AR(p) Models

The stationary AR(*p*) model for Y_t is

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

Then, assuming we have the data up to time *T* ($Y_1, Y_2, \ldots, Y_T$) and parameter constancy, we produce at time *T l-step ahead* forecasts using:

$$Y_{T+1} = \mu + \phi_1 Y_T + \phi_2 Y_{T-1} + \cdots + \phi_p Y_{T-p+1} + \varepsilon_{T+1}$$
$$Y_{T+2} = \mu + \phi_1 Y_{T+1} + \phi_2 Y_T + \cdots + \phi_p Y_{T-p+2} + \varepsilon_{T+2}$$

⋮

$$Y_{T+l} = \mu + \phi_1 Y_{T+l-1} + \phi_2 Y_{T+l-2} + \cdots + \phi_p Y_{T+l-p} + \varepsilon_{t+l}$$

Now, we take conditional expectations:

$$\hat{Y}_{T+l} = E[Y_{T+l}|I_T] = \mu + \phi_1 \mathrm{E}[Y_{T+l-1}|I_T] + \phi_2 \mathrm{E}[Y_{T+l-2}|I_T] + \cdots + \phi_p \mathrm{E}[Y_{T+l-p}|I_T]$$

Note that $\mathrm{E}[Y_{T+l-j}|I_T]$ are also forecasts. The forecasts $\hat{Y}_{T+l}$ is a linear combination of past forecast.

Example: AR(2) model for Y_{t+l} is

$$Y_{t+l} = \mu + \phi_1 Y_{t+l-1} + \phi_2 Y_{t+l-2} + \varepsilon_{t+l}$$

Then, taking conditional expectations at time $T=t$, we get the forecasts:

$$\hat{Y}_{t+1} \quad \hat{Y}_{t+1} = \mu + \phi_1 Y_t + \phi_2 Y_{t-1}$$
$$\hat{Y}_{t+2} = \mu + \phi_1 \hat{Y}_{t+1} + \phi_2 Y_t$$
$$\hat{Y}_{t+3} = \mu + \phi_1 \hat{Y}_{t+2} + \phi_2 \hat{Y}_{t+1}$$

⋮

$$\hat{Y}_{t+l} = \mu + \phi_1 \hat{Y}_{t+l-1} + \phi_2 \hat{Y}_{T+l-1}$$

AR-based forecasts are autocorrelated, they have long memory! ¶

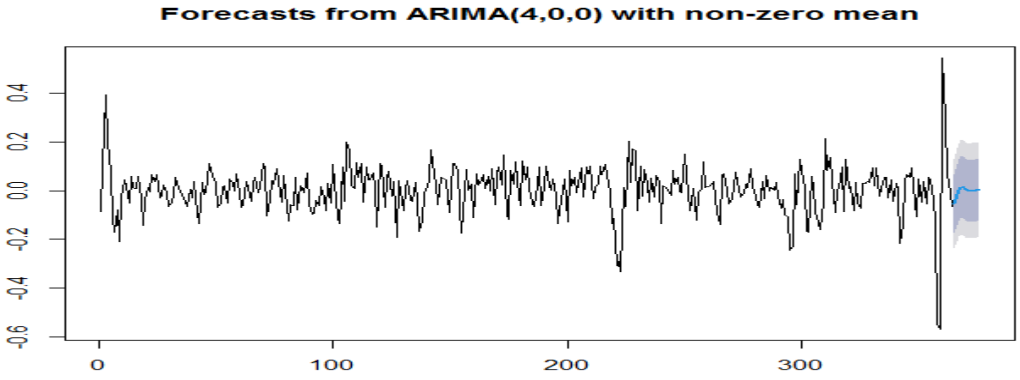
Example: We fit an AR(4) to the changes in Oil Prices (T=346):

```
fit_oil_ts <- arima(lr_oil, order=c(4,0,0))
fcast_oil <- forecast(fit_oil_ts, h=12)
> fit_oil_ts

Coefficients:
        ar1      ar2      ar3      ar4     intercept
      0.2946  -0.1027  -0.0571  -0.0983     0.0017
s.e.  0.0521   0.0543   0.0551   0.0539     0.0051

sigma^2 estimated as 0.008812:  log likelihood = 344.57,  aic = -677.14

> fcast_oil
   Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
365  -5.425015e-02 -0.1745546 0.0660543 -0.2382399 0.1297396
366  -1.578754e-02 -0.1412048 0.1096297 -0.2075966 0.1760216
367   2.455760e-03 -0.1229760 0.1278875 -0.1893755 0.1942871
368   1.356917e-02 -0.1123501 0.1394884 -0.1790077 0.2061460
369   1.160479e-02 -0.1154462 0.1386558 -0.1827029 0.2059125
370   5.060891e-03 -0.1221954 0.1323172 -0.1895608 0.1996826
371   9.059104e-04 -0.1263511 0.1281629 -0.1937169 0.1955287
```

Note: You can extract the point forecasts from the forecast function using $mean. That is, fcast_oil$mean extracts the whole vector of forecasts.

- We plot the 12 forecasts:

```
> plot(fcast_oil)
```



Remark: Different from the MA(1) forecasts, the AR(1) process generates non-constant forecasts. ¶

Forecasting From ARMA(p,q) Models

The stationary ARMA model for Y_t is

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

Assume that we have data $Y_1, Y_2, \ldots, Y_T$; $\varepsilon_1 = \hat{\varepsilon}_1$, $\varepsilon_2 = \hat{\varepsilon}_2$, ..., $\varepsilon_T = \hat{\varepsilon}_T$. We want to forecast $Y_{T+\ell}$. Then,

$$Y_{T+\ell} = \theta_0 + \phi_1 Y_{T+\ell-1} + \cdots + \phi_p Y_{T+\ell-p} + \varepsilon_{T+\ell} + \theta_1 \varepsilon_{T+\ell-1} + \cdots + \theta_q \varepsilon_{T+\ell-q}$$

Taking expectations:

$$\hat{Y}_{T+l} = \theta_0 + \phi_1 \hat{Y}_{T+l-1} + \cdots + \phi_p \hat{Y}_{T+l-p} + E[\varepsilon_{T+\ell}|I_T] + \theta_1 \mathrm{E}[\varepsilon_{T+\ell-1}|I_T] + \cdots + \theta_q \ \mathrm{E}[\varepsilon_{T+\ell-q}|I_T]$$

Remark: An ARMA forecasting is a combination of past $\hat{Y}_{T+l-i}$ forecasts and observed past $\hat{\varepsilon}_{t+l-i}$.

Alternatively, considering the Wold representation:

$$Y_{T+\ell} = \mu + \Psi(B)\varepsilon_t = \theta_0 + \frac{\theta_q(B)}{\phi_p(B)}\varepsilon_t = \mu + \varepsilon_{T+\ell} + \Psi_1 \varepsilon_{T+\ell-1} + \Psi_2 \varepsilon_{T+\ell-2} + \cdots + \Psi_\ell \varepsilon_T + \cdots$$

Taking the expectation of $Y_{T+\ell}$, we have

$$\hat{Y}_{T+\ell} = E(Y_{T+\ell}|Y_T, Y_{T-1}, \cdots, Y_1) = \mu \ + \ \Psi_\ell \varepsilon_T + \Psi_{\ell+1}\varepsilon_{T-1} + \cdots$$

where

$$E\left(\varepsilon_{T+j}|Y_T, \cdots, Y_1\right) = \begin{cases} 0, & j > 0 \\ \varepsilon_{T+j}, & j \le 0 \end{cases}$$

Then, we define the forecast error:

$$\begin{aligned} e_T(\ell) = Y_{T+\ell} - \hat{Y}_{T+\ell} &= \varepsilon_{T+\ell} + \Psi_1 \varepsilon_{T+\ell-1} + \cdots + \Psi_{\ell-1}\varepsilon_{T+1} \\ &= \sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i} \end{aligned}$$

The forecast error is: $e_T(\ell) = \sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i}$

Note: The expectation of the forecast error: E[$e_T(\ell)$] = 0

⇒ we say the forecast is *unbiased.*

- The variance of the forecast error:

$$Var\big(e_T(\ell)\big) = Var\left(\sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i}\right) = \sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2$$

Example 1: One-step ahead forecast (l = 1).

$$Y_{T+1} = \mu + \varepsilon_{T+1} + \Psi_1 \varepsilon_T + \Psi_2 \varepsilon_{T-1} + \cdots$$
$$\hat{Y}_{T+1} = \mu + \Psi_1 \varepsilon_T + \Psi_2 \varepsilon_{T-1} + \cdots$$
$$e_T(1) = Y_{T+1} - \hat{Y}_{T+1} = \varepsilon_{T+1}$$
$$Var\big(e_T(1)\big) = \sigma^2\ .\ ¶$$

Example 2: One-step ahead forecast (ℓ = 2).

$$Y_{T+2} = \mu + \varepsilon_{T+2} + \Psi_1 \varepsilon_{T+1} + \Psi_2 \varepsilon_T + \cdots \hat{Y}_{T+2}$$
$$= \mu + \Psi_2 \varepsilon_T + \cdots e_T(2)$$
$$= Y_{T+2} - \hat{Y}_{T+2} = \varepsilon_{T+2} + \Psi_1 \varepsilon_{T+1} Var\big(e_T(2)\big) = \sigma^2 * (1 + \Psi_1^2)$$

Note:

$$\underset{\ell\to\infty}{\widehat{\lim Y_T}}\ (\ell) = \mu$$
$$\lim_{\ell\to\infty} Var[e_T(\ell)] = \gamma_0 < \infty$$

As we forecast into the future, the forecasts are not very interesting (unconditional forecasts!). That is why ARMA (or ARIMA) forecasting is useful only for short-term forecasting. ¶

A 100(1- α)% prediction interval for $Y_{T+\ell}$ (ℓ-steps ahead) is

Example: 95% C.I. for the 2-step-ahead forecast:

$$\hat{Y}_T(2)\ \pm\ 1.96\,\sigma\sqrt{1 + \Psi_1^2}$$

When computing prediction intervals from data, we substitute estimates for parameters, giving approximate prediction intervals. ¶

Note: Since Ψ_i's are RV, MSE[$\varepsilon_{T+\ell}$] = MSE[$e_{T+\ell}$] = $\sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2$

Example: We fit an ARMA(4, 5), as selected by the function *auto.arima*, to changes in monthly U.S. earnings (1871 – 2020):

```
x_E <- Sh_da$E
T <- length(x_E)
lr_e <- log(x_E[-1]/x_E[-T])
fit_e <- auto.arima(lr_e)
> auto.arima(lr_e)
```

```
Series: lr_e
ARIMA(4,0,5) with non-zero mean

Coefficients:
        ar1     ar2          ar3      ar4     ma1      ma2      ma3     ma4
     0.3541  0.9786  0.2530  -0.6381  0.2943  -0.6794  -0.5720  0.1787
s.e.  0.0414  0.0466  0.0414   0.0363  0.0455   0.0400   0.0465  0.0362
         ma5    mean
     -0.1498  0.0032
s.e.   0.0286  0.0008

sigma^2 estimated as 0.0005759:  log likelihood=4140.46
AIC=-8258.91   AICc=-8258.76   BIC=-8198.52
```

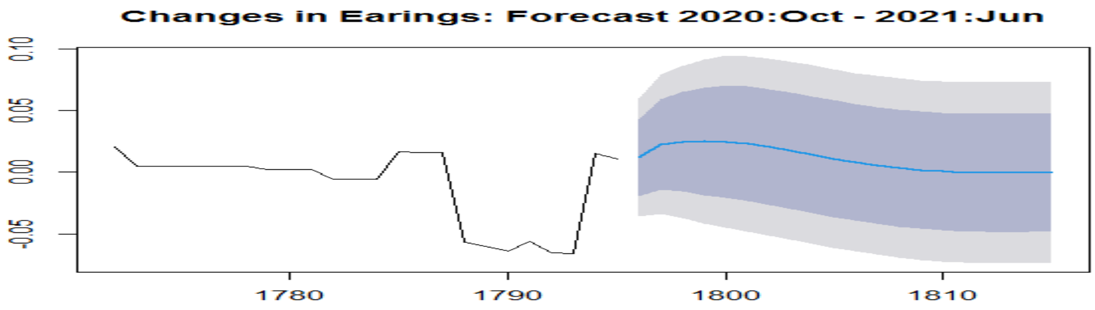
• We forecast 20 periods ahead

```
> fcast_e <- forecast(fit_e, h=20)                  # h=number of step-ahead forecasts
> fcast_e
     Point Forecast       Lo 80         Hi 80              Lo 95        Hi 95
1791   -0.054521445 -0.08527728 -0.023765608 -0.10155844 -0.007484451
1792   -0.048064225 -0.08471860 -0.011409845 -0.10412226  0.007993811
1793   -0.032702992 -0.07280271  0.007396723 -0.09403021  0.028624230
1794   -0.030680456 -0.07365723  0.012296320 -0.09640776  0.035046851
1795   -0.017583413 -0.06228564  0.027118816 -0.08594957  0.050782746
1796   -0.013681751 -0.05882105  0.031457550 -0.08271635  0.055352853
1797   -0.008775187 -0.05458154  0.037031165 -0.07882996  0.061279583
1798   -0.001197077 -0.04705319  0.044659034 -0.07132795  0.068933794
1799   -0.001083388 -0.04698821  0.044821436 -0.07128876  0.069121982
1800    0.005124015 -0.04078796  0.051035988 -0.06509229  0.075340318
1801    0.006219195 -0.03973961  0.052178005 -0.06406874  0.076507130
1802    0.007874051 -0.03809120  0.053839304 -0.06242374  0.078171840
1803    0.011029600 -0.03506469  0.057123889 -0.05946553  0.081524732
1804    0.010082045 -0.03611076  0.056274848 -0.06056375  0.080727841
```

Note: You can extract the point forecasts from the forecast function using $mean. That is, fcast_e$mean extracts the whole vector of forecasts.

• We plot the forecast and the C.I.

```
> plot(fcast_e, type="l", include = 24, main = "Changes in Earings: Forecast 2020:Oct -
2021:Jun")                  #We include the last 24 observations along the forecast.
```



Forecasting From ARMA(p,q) Models - Updating

Suppose we have *T* observations at time *t=T*. We have a good ARMA model for Y_t. We obtain the forecast for Y_{T+1}, Y_{T+2}, etc.

- At $t = T + 1$, we observe Y_{T+1}. Now, we update our forecasts using the original value of Y_{T+1} and the forecasted value of it.

The forecast error is: $e_T(\ell) = Y_{T+\ell} - \hat{Y}_T(\ell) = \sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i}$

The forecast error associated with $\hat{Y}_{T-1}(\ell+1)$ is:

$$\begin{aligned} e_{T-1}(\ell+1) &= Y_{T-1+\ell+1} - \hat{Y}_{T-1}(\ell+1) \\ &= \sum_{i=0}^{\ell} \Psi_i \varepsilon_{T-1+\ell+1-i} = \sum_{i=0}^{\ell} \Psi_i \varepsilon_{T+\ell-i} \\ &= \sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i} + \Psi_\ell \varepsilon_T = e_T(\ell) + \Psi_\ell \varepsilon_T \end{aligned}$$

- Then,

$$e_{T-1}(\ell+1) = Y_{T+\ell} - \hat{Y}_{T-1}(\ell+1) = Y_{T+\ell} - \hat{Y}_T(\ell) + \Psi_\ell \varepsilon_T$$
$$\hat{Y}_T(\ell) = \hat{Y}_{T-1}(\ell+1) + \Psi_\ell \varepsilon_T$$
$$\hat{Y}_T(\ell) = \hat{Y}_{T-1}(\ell+1) + \Psi_\ell \{Y_T - \hat{Y}_{T-1}(1)\}$$
$$\hat{Y}_{T+1}(\ell) = \hat{Y}_T(\ell+1) + \Psi_\ell \{Y_{T+1} - \hat{Y}_T(1)\}$$

Example: $\ell = 1$, $T = 100$.

$$\hat{Y}_{101}(1) = \hat{Y}_{100}(2) + \Psi_1 \{Y_{101} - \hat{Y}_{100}(1)\}\ .\ ¶$$

Forecasting From ARMA(p,q) Models - Remarks

In general, we need a large *T*. Better estimates and it is possible to check for model stability and check forecasting ability of model by withholding data.

Seasonal patterns also need large *T*. Usually, you need 4 to 5 seasons to get reasonable estimates.

Parsimonious models are very important. Easier to compute and interpret models and forecasts. Forecasts are less sensitive to deviations between parameters and estimates.

Forecasting From Simple Models: ES

Industrial companies, with a lot of inputs and outputs, want quick and inexpensive forecasts. Easy to fully automate.

Exponential Smoothing Models (ES) fulfill these requirements.

In general, these models are limited and not optimal, especially compared with Box-Jenkins methods.

Goal of these models: Suppress the short-run fluctuation by smoothing the series. For this purpose, a weighted average of all previous values works well.
There are many ES models. We will go over the Simple Exponential Smoothing (SES) and Holt-Winter's Exponential Smoothing (HW ES).

Simple Exponential Smoothing: SES

We "*smooth*" the series Y_t to produce a quick forecast, S_{t+1} also referred as the "*level's* forecast*"*,. Smooth? The graph of S_t is less jagged than the graph of original series Y_t.

Observed time series: $Y_1, Y_2, \ldots, Y_T$

The equation for the model is:

$$S_t = \alpha Y_{t-1} + (1-\alpha)S_{t-1}$$

where

- α: The smoothing parameter, 0 ≤ α ≤ 1.
- Y_t: Value of the observation at time t.
- S_t: Value of the smoothed observation at time t –i.e., the forecast.

The equation can also be written as an updating equation:

$$S_t = S_{t-1} + \alpha(Y_{t-1} - S_{t-1}) = S_{t-1} + \alpha(\text{forecast error})$$

SES: Forecast and Updating

From the updating equation for S_t*:*

$$S_t = S_{t-1} + \alpha(Y_{t-1} - S_{t-1})$$

we compute the forecast:

$$S_{t+1} = \alpha Y_t + (1-\alpha)S_t = S_t + \alpha(Y_t - S_t)$$

That is, a simple updating forecast: last period forecast + adjustment.

For the next period, we have:

$$S_{t+2} = \alpha Y_{t+1} + (1-\alpha)S_{t+1} = \alpha S_{t+1} + (1-\alpha)S_{t+1} = S_{t+1}$$

Then the ℓ-step ahead forecast is:

$$S_{t+\ell} = S_{t+1} \quad \Rightarrow \text{A naive forecast!}$$

Note: ES forecasts are not very interesting after $\ell > 1$.

SES: Exponential?

Question: Why Exponential?
For the observed time series $\{Y_1, Y_2, \ldots, Y_T, Y_{T+1}\}$, using backward substitution, $S_{t+1} = \hat{Y}_t(1)$ can be expressed as a weighted sum of previous observations:

$$\begin{aligned} S_{t+1} &= \alpha Y_t + (1-\alpha)S_t = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)S_{t-1}] \\ &= \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + (1-\alpha)^2 S_{t-1} \\ &\Rightarrow \hat{Y}_t(1) = c_0 Y_t + c_1 Y_{t-1} + c_2 Y_{t-2} + \cdots \end{aligned}$$

where c_i's are the weights, with

$$c_i = \alpha(1-\alpha)^i;\ i = 0,\ 1,\ \ldots;\ 0 \le \alpha \le 1.$$

We have decreasing weights, by a constant ratio for every unit increase in lag.

Then,

$$\hat{Y}_t(1) = \alpha(1-\alpha)^0 Y_t + \alpha(1-\alpha)^1 Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \cdots$$

$$\hat{Y}_t(1) = \alpha Y_t + (1-\alpha)\hat{Y}_{t-1}(1) \quad \Rightarrow S_{t+1} = \alpha Y_t + S_t$$

- Let's look at the weights:

$$c_i = \alpha\,(1-\alpha)^i; \qquad i = 0,\ 1,\ \ldots;\ 0 \le \alpha \le 1.$$

$\boldsymbol{c_i = \alpha(1-\alpha)^i}$	**α = 0.25**	**α = 0.75**
c_0	0.25	0.75
c_1	0.25 * 0.75 = 0.1875	0.75 * 0.25 = 0.1875
c_2	.25 * 0.75^2 = 0.140625	0.75 * 0.25^2 = 0.046875
c_3	.25 * 0.75^3 = 0.1054688	0.75 * 0.25^3 = 0.01171875
c_4	.25 * 0.75^4 = 0.07910156	0.75 * 0.25^4 = 0.002929688
⋮		
c_{12}	.25 * 0.75^{12} = 0.007919088	0.75 * 0.25^{12} = 4.470348e-08

Decaying weights. Faster decay with greater α, associated with faster learning: we give more weight to more recent observations.

We do not know α; we need to estimate it.

SES: Selecting α

Choose α between 0 and 1.

- If α = 1, it becomes a naive model; if α ≈ 1, more weights are put on recent values. The model fully utilizes forecast errors.
- If α is close to 0, distant values are given weights comparable to recent values. Set α ≈ 0 when there are big random variations in Y_t.
- α is often selected as to minimize the MSE.

In empirical work, 0.05 ≤ α ≤ 0.3 are used (α ≈ 1 is used rarely).

Numerical Minimization Process:
- Take different α values ranging between 0 and 1.
- Calculate 1-step-ahead forecast errors for each α.
- Calculate MSE for each case.

Choose α which has the min MSE: $e_t = Y_t - S_t \Rightarrow \min \sum_{t=1}^{n} e_t^2 \Rightarrow \alpha$

Example:

Time	Y_t	S_{t+1} ($\alpha = 0.10$)	$(Y_t – S_t)^2$
1	**5**	-	-
2	**7**	(0.1)**5** +(0.9)5 = **5**	4
3	6	(0.1)**7** + (0.9)**5** = **5.2**	0.64
4	3	(0.1)6 + (0.9)**5.2** = **5.28**	5.1984
5	4	(0.1)3 + (0.9)**5.28** = 5.052	1.107
TOTAL			**10.945**

$$MSE = \frac{SSE}{n-1} = 2.74$$

Calculate this for α = 0.2, 0.3,…, 0.9, 1 and compare the MSEs. Choose α with minimum MSE.

Note: $Y_{t=1}$ = **5** is set as the initial value for the recursive equation. ¶

SES: Initial Values

We have a recursive equation, we need initial values, S_1 (or Y_0).

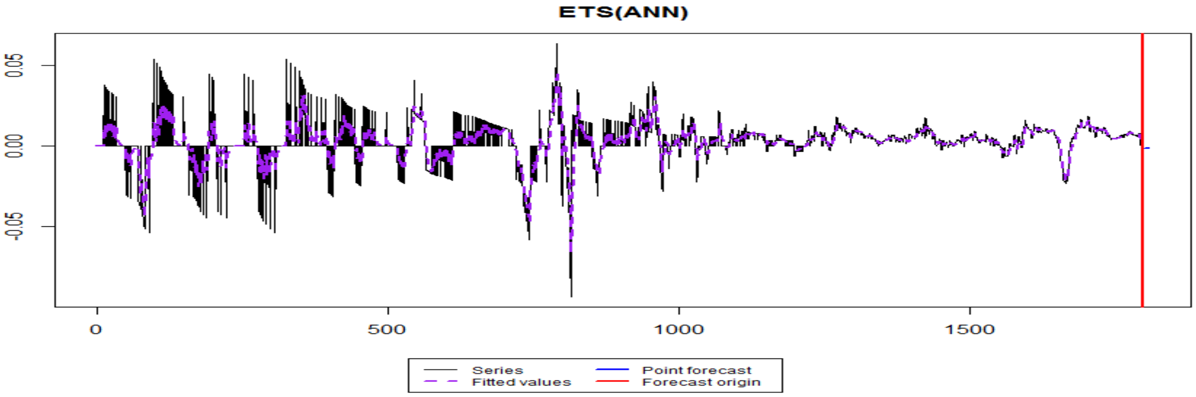
Approaches:
– Set S_1 to Y_1 is one method of initialization. Then, $S_2 = Y_1$.
– Also, take the average of, say first 4 or 5 observations. Use this average as an initial value.

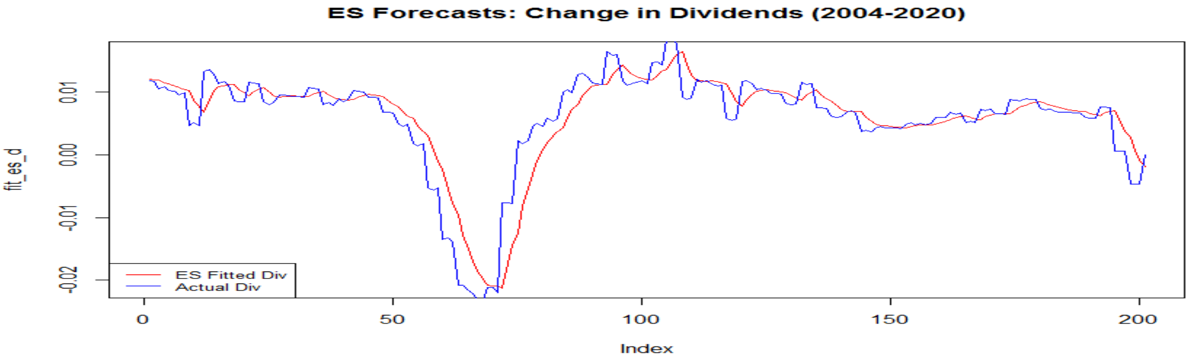
– Estimate S_1 (similar to the estimation of α.)

SES: Forecasting Examples

Example 1: We want to forecast log changes in U.S. monthly dividends (T=1796) using SES. First, we estimate the model using the R function *HoltWinters*(), which has as a special case SES: set beta=FALSE, gamma=FALSE. We use estimation period *T*=1750.

```
mod1 <- HoltWinters(lr_d[1:1750], beta=FALSE, gamma=FALSE)
> mod1
Holt-Winters exponential smoothing without trend and without seasonal component.
Call:
HoltWinters(x = lr_d[1:1750], beta = FALSE, gamma = FALSE)
Smoothing parameters:
 alpha: 0.289268                              ⇒ Estimated α
 beta : FALSE
 gamma: FALSE
Coefficients:
        [,1]
a 0.004666795                          ⇒ Forecast
```

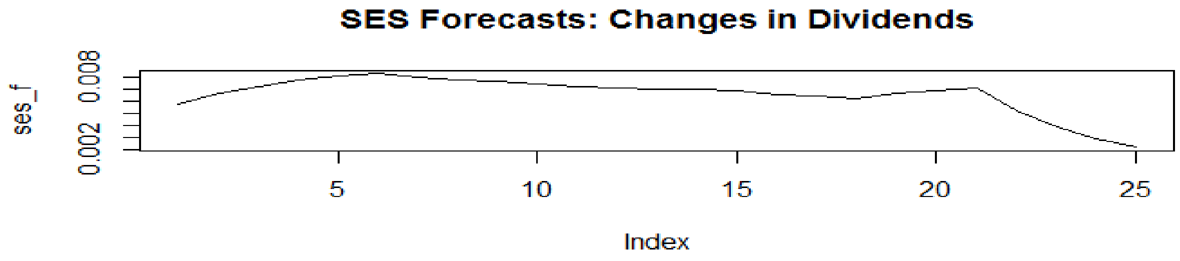


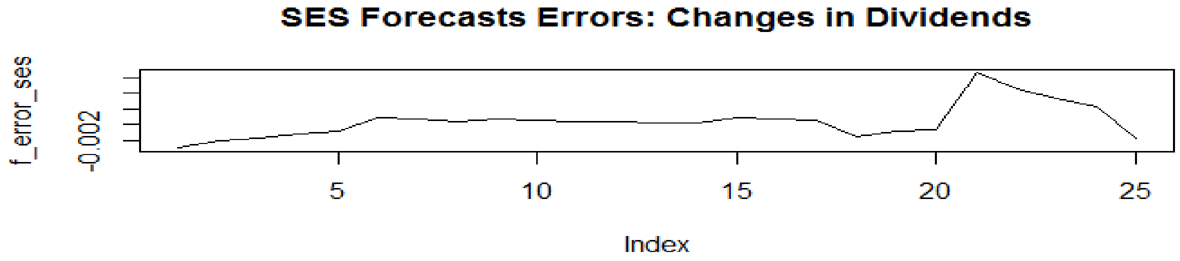


- Now, we forecast one-step ahead forecasts

```
T_last <- nrow(mod1$fitted)                    # number of in-sample forecasts
h <- 25                                        # forecast horizon
ses_f <- matrix(0,h,1)                         # Vector to collect forecasts
alpha <- 0.29
```

```
y <- lr_d
T <- length(lr_d)
sm <- matrix(0,T,1)
T1 <- T – h + 1                                          # Start of forecasts
a <- T1                                          # index for while loop
sm[a-1] <- mod1$fitted[T_last]                           # last in-sample forecast
while (a <= T) {
        sm[a] = alpha * y[a-1] +  (1-alpha) * sm[a-1]
a <- a + 1
}
ses_f <- sm[T1:T]
ses_f
f_error_ses <- sm[T1:T] - y[T1:T]                # forecast errors
MSE_ses <- sum(f_error_ses^2)/h                  # MSE
plot(ses_f, type="l", main ="SES Forecasts: Changes in Dividends")
```





- *h-step-ahead* forecasts

```
> forecast(mod1, h=25, level=.95)
     Point Forecast       Lo 95       Hi 95
1751    0.004666795 -0.01739204 0.02672563
1752    0.004666795 -0.01829640 0.02762999
1753    0.004666795 -0.01916647 0.02850006
1754    0.004666795 -0.02000587 0.02933947
1755    0.004666795 -0.02081765 0.03015124
1756    0.004666795 -0.02160435 0.03093794
1757    0.004666795 -0.02236816 0.03170175
1758    0.004666795 -0.02311098 0.03244457
1759    0.004666795 -0.02383445 0.03316804
1760    0.004666795 -0.02454001 0.03387360
1761    0.004666795 -0.02522891 0.03456250
```

```
1762    0.004666795 -0.02590230 0.03523589
1763    0.004666795 -0.02656117 0.03589476
1764    0.004666795 -0.02720642 0.03654001
...
```

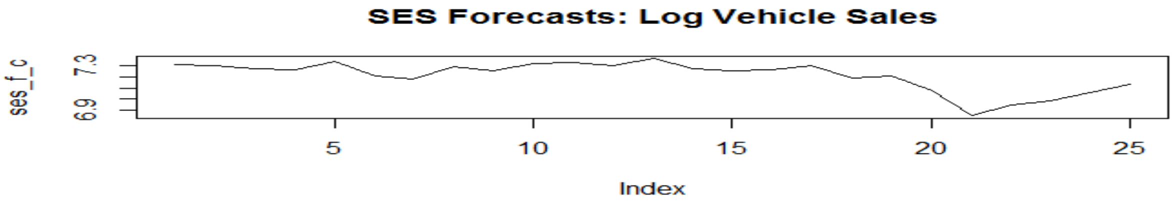
Note: Constant forecasts, but C.I. gets wider (as expected) with h. ¶

Example 2: We want to forecast **log monthly U.S. vehicles** (1976-2020, T=537) using SES.

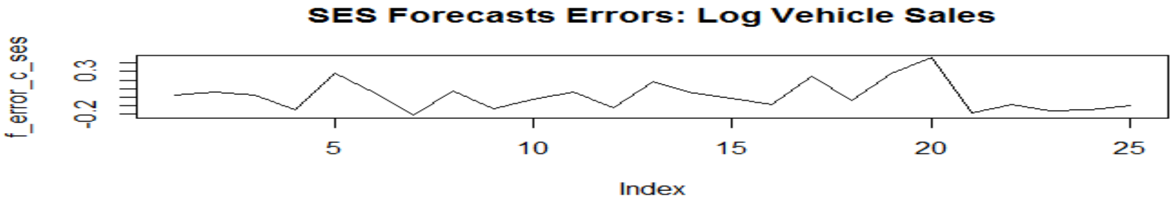
```
mod_car <- HoltWinters(l_car[1:512], beta=FALSE, gamma=FALSE)
> mod_car
Holt-Winters exponential smoothing without trend and without seasonal component.
Call:
HoltWinters(x = l_car[1:512], beta = FALSE, gamma = FALSE)
Smoothing parameters:
 alpha: 0.4888382                    ⇒ Estimated α
 beta : FALSE
 gamma: FALSE
Coefficients:
      [,1]
a 7.315328
```

• Now, we do one-step ahead forecasting

```
ses_f_c <- sm_c[T1:T]
f_error_c_ses <- sm_c[T1:T] - y[T1:T]
> plot(ses_f_c, type="l", main ="SES Forecasts: Log Vehicle Sales")
```



```
> plot(f_error_c_ses, type="l", main ="SES Forecasts Errors: Log Vehicle Sales")
```



```
MSE_ses <- sum(f_error_c_ses^2)/h
> MSE_ses
[1] 0.027889. ¶
```

SES: Remarks

Some computer programs automatically select the optimal α using a line search method or non-linear optimization techniques.

We have a recursive equation, we need initial values for S_1.

This model ignores trends or seasonalities. Not very realistic, especially for manufacturing facilities, retail sector, and warehouses. But, deterministic components, D_t, can be easily incorporated.

The model that incorporates both features is called *Holt-Winter's ES.*

Holt-Winters (HW) ES: Multiplicative Model

Now, we introduce trend (T_t) and seasonality (I_t) factors. We produce smooth forecasts for them too. Both can be included as additively or multiplicatively factors.

Details for multiplicative seasonality –i.e., Y_t/I_t– and additive trend
- The forecast, S_t, is adjusted by the deterministic trend: $S_t + T_t$.
- The trend, T_t, is a weighted average of T_{t-1} and the change in S_t.
- The seasonality is also a weighted average of I_{t-S} and the Y_t/S_t

• Then, the model has three equations:

$$S_t = \alpha \frac{Y_{t-1}}{I_{t-s}} + (1-\alpha)\,(S_{t-1} - T_{t-1})$$

$$T_t = \beta\,(S_t - S_{t-1}) + (1-\beta)\,T_{t-1}$$

$$I_t = \gamma \frac{Y_t}{S_t} + (1-\gamma)\,I_{t-s}$$• We think of (Y_t/S_t) as capturing *seasonal effects*.

We think of (Y_t/S_t) as capturing *seasonal effects*.

s = # of periods in the seasonal cycles

(s = 4, for quarterly data)

We have only three parameters:

α = smoothing parameter

β = trend coefficient

γ = seasonality coefficient

Question: How do we determine these 3 parameters?
- Ad-hoc method: α, β and γ can be chosen as value between 0.02< α, γ, β <0.2
- Optimal method: Minimization of the MSE, as in SES.

Holt-Winters (HW) ES: Forecasting & Initial Values

h-step ahead forecast: $\hat{Y}_t(h) = (S_t + h\,T_t) * I_{t+h-s}$

Note: Seasonal factor is multiplied in the *h*-step ahead forecast

• To calculate initial values for the algorithm, we need at least one complete season of data to determine the initial estimates.

- Initial values for S_0 and T_0:

$$S_0 = \frac{\sum_{t=1}^{s} Y_t}{sT_0}$$

$$\mathrm{T}_0 = \frac{1}{s}\left(\frac{Y_{s+1}-Y_1}{s} + \frac{Y_{s+2}-Y_2}{s} + \cdots + \frac{Y_{s+s}-Y_s}{s}\right)$$

$$\text{or } \mathrm{T}_0 = [\{\textstyle\sum_{t=1}^{s} Y_t/s\} - \{\sum_{t=s+1}^{2s} Y_t/s\}]/s$$

- Initial values for $I_{t\text{-}s}$.:

Assume we have *T* observation and quarterly seasonality *(s*=4):

(1) Compute the averages of each of *T* years.

$$A_t = \textstyle\sum_{i=1}^{4} Y_{t,i}/4\,, \quad t = 1,\, 2, \cdots, 6 \quad \text{(yearly averages)}$$

(2) Divide the observations by the appropriate yearly mean: $Y_{t,i}/A_t$.

(3) I_s is formed by computing the average $Y_{t,i}/A_t$ per year:

$$I_s = \textstyle\sum_{i=1}^{T} Y_{t,s}/A_t \quad s = 1,\, 2,\, 3,\, 4$$

Holt-Winters (HW) ES: Forecasting & Initial Values

We can damp the trend as the forecast horizon increases, using a parameter ϕ:

$$S_t = \alpha\frac{Y_{t-1}}{I_{t-s}} + (1-\alpha)(S_{t-1} - \phi\, T_{t-1})T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}I_t = \gamma\frac{Y_t}{S_t} + (1-\gamma)I_{t-s}$$

h-step ahead forecast:

$$\hat{Y}_t(h) = \{S_t + (1 + \phi + \phi^2 + \cdots + +\phi^{2h-1})T_t\} * I_{t+h-s}$$

This model is based on practice: It seems to work well for industrial outputs. Not a lot of theory or clear justification behind the damped trend.

Holt-Winters (HW) ES: Additive Model

Instead of a multiplicative seasonal pattern, we use an additive one.

Now, the model has the following three equations:

$$S_t = \alpha(Y_{t-1} - I_{t-s}) + (1-\alpha)(S_{t-1} - T_{t-1})$$

$$T_t = \beta(S_t - S_{t-1}) + (1-\beta)T_{t-1}$$

$$I_t = \gamma(S_{t-1} - T_{t-1}) + (1-\gamma)I_{t-s}$$

h-step ahead forecast:

$$\hat{Y}_t(h) = S_t + h\,T_t + I_{t+h-s}$$

HW ES Models – Different Types

We have many variations:
1. No trend and additive seasonal variability.
2. Additive seasonal variability with an additive trend.
3. Multiplicative seasonal variability with an additive trend.
4. Multiplicative seasonal variability with a multiplicative trend.
5. Dampened trend with additive seasonal variability.
6. Multiplicative seasonal variability and dampened trend.

Select the type of model to fit based on the presence of
- Trend – additive or multiplicative, dampened or not
- Seasonal variability – additive or multiplicative

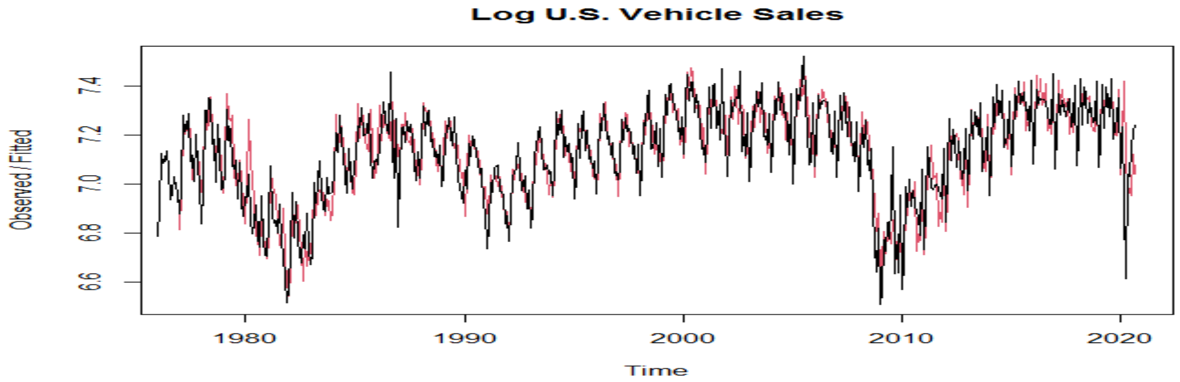
HW ES: Example – Log U.S. Vehicles Sales

Example: We want to forecast log U.S. monthly vehicle sales with HW. We use the R function *HoltWinters*().

```
l_car_18 <- l_car[1:512]
l_car_ts <- ts(l_car_18, start = c(1976, 1), frequency = 12)   # convert lr_d in a ts object
hw_d_car <- HoltWinters(l_car_18, seasonal="additive")
> hw_d_car
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = lr_d_ts, seasonal = "additive")

Smoothing parameters:
 alpha: 0.4355244                   ⇒ Estimated smoothing parameter
 beta : 0.009373815                 ⇒ Estimated trend parameter ≈ 0 (no trend)
 gamma:0.3446495            ⇒ Estimated seasonal parameter

> hw_d_car
Coefficients:
          [,1]
a    7.177857555                    ⇒ forecast for level
b    0.0001100345                   ⇒ forecast for trend
s1  -0.075314457                    ⇒ forecast for seasonal month 1
s2  -0.084468361                    ⇒ forecast for seasonal month 2
s3   0.049447067
s4  -0.273299309
s5  -0.138251757
s6  -0.026603921
s7  -0.144953062
s8   0.079214066
s9   0.037899454
s10  0.020477134
s11  0.089309775
```

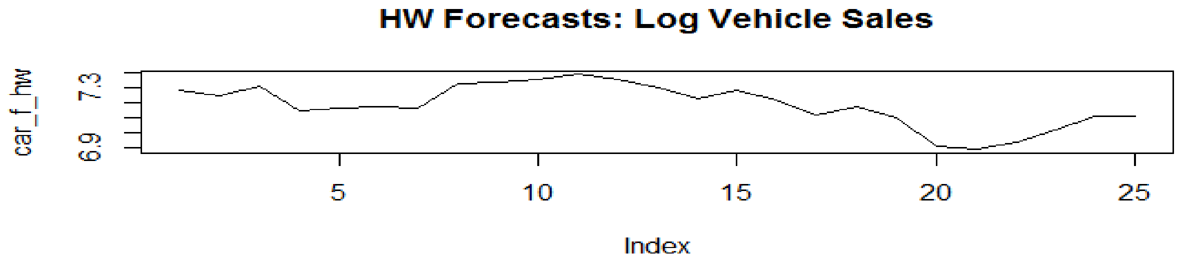
```
s12 -0.012530316
```

```
>plot(hw_d_car)
```



- Now, we forecast one-step ahead forecasts

```
T_last <- nrow(hw_d_car$fitted)
h <- 25
ses_f_hw <- matrix(0,h,1)
 alpha <- 0.4355244
 beta <- 0.009373815
 gamma <- 0.3446495
y <- l_car
T <- length(l_car)
sm <- matrix(0,T,1)
Tr <- matrix(0,T,1)
I <- matrix(0,T,1)
T1 <- T-h+1
a <- T1
sm[a-1] <- 7.177857555
Tr[a-1] <- -0.000309358
I[501:512] <- c(-0.075314457,-0.084468361,0.049447067,-0.273299309,-0.138251757, -
0.026603921, -0.144953062,0.079214066,0.037899454,0.020477134,0.089309775,-
0.012530316)
while (a <= T) {
          sm[a] = alpha * y[a-1] +  (1-alpha) * sm[a-1]
          Tr[a] = beta * (sm[a] - sm[a-1]) + (1 - beta) * Tr[a-1]
          I[a] = gamma * (y[a] - sm[a]) + (1 - gamma) * I[a - 12]
a <- a + 1
}

hh <- c(1:h)
car_f_hw <- sm[T1:T] + hh*Tr[T1:T] + I[T1:T]
car_f_hw
f_error_c_hw<- car_f_hw - y[T1:T]
plot(car_f_hw, type="l", main ="SES Forecasts: Log Vehicle Sales")
```



```
MSE_hw <- sum(f_error_c_hw^2)/h
> MSE_hw
[1] 0.01655964. ¶
```

HW ES: Remarks

If a computer program selects γ = 0 = β, it has a lack of trend or seasonality. It implies a constant (deterministic) component. In this case, an ARIMA model with deterministic trend may be a more appropriate model.

- For HW ES, a seasonal weight near one implies that a non-seasonal model may be more appropriate.

We can model seasonalities as multiplicative or additive:

$\Rightarrow$ Multiplicative seasonality: Forecast$_t$ = S$_t$ * I$_{t-s}$.

$\Rightarrow$ Additive seasonality: Forecast$_t$ = S$_t$ + I$_{t-s}$.

Evaluation of forecasts – Accuracy measures

The mean squared error (*MSE*) and mean absolute error (*MAE*) are the most popular accuracy measures:

$$\text{MSE} = \frac{1}{m}\sum_{i=T+1}^{T+m}(\hat{y}_i - y_i)^2 = \frac{1}{m}\sum_{i=T+1}^{T+m} e_i{}^2$$

$$\text{MAE} = \frac{1}{m}\sum_{i=T+1}^{T+m}|\hat{y}_i - y_i\,| = \frac{1}{m}\sum_{i=T+1}^{T+m}|e_i|$$

where *m* is the number of out-of-sample forecasts.

But other measures are routinely used:

- Mean absolute percentage error (*MAPE*) = $\frac{100}{T-(m-1)}\sum_{i=T+1}^{T+m}|\frac{\hat{y}_i-y_i}{y_i}|$

- Absolute *MAPE (AMAPE)* = $\frac{100}{T-(m-1)}\sum_{i=T+1}^{T+m}|\frac{\hat{y}_i-y_i}{\hat{y}_i+y_i}|$

<u>Remark</u>: There is an asymmetry in MAPE, the level y_i matters.

- % correct sign predictions (PCSP) = $\frac{1}{T-(m-1)}\sum_{i=T+1}^{T+m} z_i$

where z_i = 1 if $(\hat{y}_{i+l} \;*\; y_{i+l}) > 0$
= 0, otherwise.

- % correct direction change predictions (PCDP)= $\frac{1}{T-(m-1)}\sum_{i=T+1}^{T+m} z_i$

where z_i = 1 if $(\hat{y}_{i+l} - y_i) * (y_{i+l} - y_i) > 0$
= 0, otherwise.

Remark: We value forecasts with the right direction (sign) or forecast that can predict turning points. For stock investors, the sign matters!

MSE penalizes large errors more heavily than small errors, the sign prediction criterion, like MAE, does not penalize large errors more.

Example: We compute MSE and the % of correct direction change (PCDC) predictions for the one-step forecasts for U.S. monthly vehicles sales based on the SES and HW ES models.

```
> MSE_ses
[1] 0.027889
> MSE_hw
[1] 0.01655964
```

We calculate PCDC with following script for HW and SES:

```
bb_hw <- (car_f_hw - y[(T1-1):(T-1)]) * (y[T1:T] - y[(T1-1):(T-1)])
indicator_hw <- ifelse(bb_hw > 0,1,0)
pcdc_hw <- sum(indicator_hw)/h
> indicator_hw
 [1] 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 0 0
> pcdc_hw
[1] 0.76
```

```
bb_s <- (ses_f_c - y[(T1-1):(T-1)]) * (y[T1:T] - y[(T1-1):(T-1)])
indicator_s <- ifelse(bb_s > 0,1,0)
pcdc_s <- sum(indicator_s)/h
> indicator_s
 [1] 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0
> pcdc_s
[1] 0.76. ¶
```

Evaluation of forecasts – DM Test

To determine if one model predicts better than another, we define the loss differential between two forecasts:

$$d_t = g(e_t^{M1}) - g(e_t^{M2})$$

where g(.) is the forecasting loss function. M1 and M2 are two competing sets of forecasts – could be from models or something else.

We only need $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$, not the structure of M1 or M2. In this sense, this approach is "*model-free*."

Typical (symmetric) loss functions: $g(e_t) = e_t^2$ & $g(e_t) = |e_t|$.

But other g(.)'s can be used: $g(e_t) = \exp(\lambda e_t^2) - \lambda e_t^2$ $(\lambda>0)$.

Then, we test the null hypotheses of equal predictive accuracy:

H_0: $E[d_t] = 0$
H_1: $E[d_t] = \mu \neq 0$.

- Diebold and Mariano (1995) assume $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$ is covariance stationarity and other regularity conditions (finite Var[d_t], independence of forecasts after ℓ periods) needed to apply CLT. Then,

$$\frac{\bar{d}-\mu}{\sqrt{Var[\bar{d}]/T}} \xrightarrow{d} N(0,1), \qquad \bar{d} = \frac{1}{m}\sum_{i=T+1}^{T+m} d_i$$

• Then, under H_0, the DM test is a simple *z-test*:

$$DM = \frac{\bar{d}}{\sqrt{\hat{V}ar[\bar{d}]/T}} \xrightarrow{d} N(0,1)$$

where $\hat{V}ar[\bar{d}]$ is a consistent estimator of the variance, usually based on sample autocovariances of d_t:

$$\hat{V}ar[\bar{d}] = \gamma(0) + 2\sum_{j=k}^{\ell}\gamma(j)$$

There are some suggestion to calculate small sample modification of the DM test. For example, :

$DM^* = DM/\{[T + 1 - 2\,\ell + \ell\,(\ell - 1)/T]/T\}^{1/2} \sim t_{T\text{-}1}$.

where ℓ-step ahead forecast. If ARCH is suspected, replace ℓ with $[0.5\,\sqrt{(T)}] + \ell$.

Note: If $\{e_t^{M1}\}$ & $\{e_t^{M2}\}$ are perfectly correlated, the numerator and denominator of the DM test are both converging to 0 as $T \to \infty$.

⇒ Avoid DM test when this situation is suspected (say, two nested models.) Though, in small samples, it is OK.

• Code in R

```
dm.test <- function (e1, e2, h = 1, power = 2) {
d <- c(abs(e1))^power - c(abs(e2))^power
  d.cov <- acf(d, na.action = na.omit, lag.max = h - 1, type = "covariance", plot = FALSE)$acf[, , 1]
  d.var <- sum(c(d.cov[1], 2 * d.cov[-1]))/length(d)
  dv <- d.var                          #max(1e-8,d.var)
  if(dv > 0)
    STATISTIC <- mean(d, na.rm = TRUE) / sqrt(dv)
  else if(h==1)
    stop("Variance of DM statistic is zero")
  else
  {
    warning("Variance is negative, using horizon h=1")
    return(dm.test(e1,e2,alternative,h=1,power))
  }
    n <- length(d)
  k <- ((n + 1 - 2*h + (h/n) * (h-1))/n)^(1/2)
  STATISTIC <- STATISTIC * k
  names(STATISTIC) <- "DM"
}
```

Example: We compare the SES and HW forecasts for the log of U.S. monthly vehicle sales. We use the *dm.test* function, part of the forecast package.

```
library(forecast)

> dm.test(f_error_c_ses, f_error_c_hw, power=2)
        Diebold-Mariano Test
data:  f_error_c_sesf_error_c_hw
DM = 1.6756, Forecast horizon = 1, Loss function power = 2, p-value = 0.1068
alternative hypothesis: two.sided

> dm.test(f_error_c_ses,f_error_c_hw, power=1)
        Diebold-Mariano Test
data:  f_error_c_sesf_error_c_hw
DM = 1.94, Forecast horizon = 1, Loss function power = 1, p-value = 0.064
alternative hypothesis: two.sided
```

Note: Cannot reject H_0: $MSE_{SES} = MSE_{HW}$ at 5% level. ¶

Evaluation of forecasts – DM Test: Remarks

The DM tests is routinely used. Its “model-free” approach has appeal. There are model-dependent tests, with more complicated asymptotic distributions.

The loss function does not need to be symmetric (like MSE).

The DM test is based on the notion of unconditional –i.e., on average over the whole sample- expected loss.

Following Morgan, Granger and Newbold (1977), the DM statistic can be calculated by regression of d_t, on an intercept, using NW SE. But, we can also condition on variables that may explain d_t. We move from an unconditional to a conditional expected loss perspective.

Combination of Forecasts

Idea – from Bates & Granger (*Operations Research Quarterly,* 1969):

- We have different forecasts from R models:

$$\hat{Y}_T^{M1}(\ell), \hat{Y}_T^{M2}(\ell), ... \hat{Y}_T^{MR}(\ell)$$

Question: Why not combine them?

$$\hat{Y}_T^{Comb}(\ell) = \omega_{M1}\hat{Y}_T^{M1}(\ell) + \omega_{M2}\hat{Y}_T^{M2}(\ell)+....+\omega_{MR}\hat{Y}_T^{MR}(\ell)$$

Very common practice in economics, finance and politics, reported by the press as "consensus forecast." Usually, as a simple average.

Question: Advantage? Lower forecast variance. Diversification argument.

Intuition: Individual forecasts are each based on partial information sets (say, private information) or models.

The variance of the forecasts is:

$$Var\left[\hat{Y}_T^{Comb}(\ell)\right] = \sum_{j=1}^{R}(\omega_{Mj})^2 Var[\hat{Y}_T^{Mj}(\ell)] + 2\sum_{j=1}^{R}\quad\sum_{i=j+1}^{R}\omega_{Mj}\omega_{Mi}Co\,\text{var}[\,\hat{Y}_T^{Mj}(\ell)\hat{Y}_T^{Mi}(\ell)]$$

Note: Ideally, we would like to have negatively correlated forecasts.

Assuming unbiased forecasts and uncorrelated errors,

$$Var[\hat{Y}_T^{Comb}(\ell)] = \sum_{j=1}^{R}(\omega_{Mj})^2\,\sigma_j^2$$

Example: Simple average: ω_j=1/R. Then,

$$Var[\hat{Y}_T^{Comb}(\ell)] = 1/R^2\sum_{j=1}^{R}\sigma_j^2.\,¶$$

Example: We combine the SES and HW forecast of log US vehicles sales:

```
f_comb <- (ses_f_c + car_f_hw)/2
f_error_comb <- f_comb - y[T1:T]
> var(f_comb)
[1] 0.0178981
> var(car_f_hw)
[1] 0.02042458
> var(ses_f_c)
[1] 0.01823237. ¶
```

Combination of Forecasts – Optimal & Regression Weights

We can derived optimal weights –i,e., ω_j's that minimize the variance of the forecast. Under the uncorrelated assumption:

$$\omega_{Mj} *= \sigma_j^{-2}/\textstyle\sum_{j=1}^{R}\sigma_j^{-2}$$

The ω_j*'s are inversely proportional to their variances.

In general, forecasts are biased and correlated. The correlations will appear in the above formula for the optimal weights. For the two forecasts case:

$$\omega_{Mj} *= (\sigma_1^2-\sigma_{12})/(\sigma_1^2+\sigma_2^2-2\sigma_{12}) = (\sigma_1^2-\rho\sigma_1\sigma_2)/(\sigma_1^2+\sigma_2^2-2\rho\sigma_1\sigma_2)$$

Ideally, we would like to have negatively correlated forecasts.

- Granger and Ramanathan(1984) used a regression method to combine forecasts.

- Regress the actual value on the forecasts. The estimated coefficients are the weights.

$$y_{T+\ell} = \beta_1 \hat{Y}_T^{M1}(\ell) + \beta_2 \hat{Y}_T^{M2}(\ell) + \ldots. + \beta_R \hat{Y}_T^{MR}(\ell) + \varepsilon_{T+\ell}$$

Should use a constrained regression
– Omit the constant
– Enforce non-negative coefficients.
– Constrain coefficients to sum to one

Example: We regress the SES and HW forecasts against the observed car sales to obtain optimal weights. We omit the constant

```
> lm(y[T1:T] ~ ses_f_c + car_f_hw - 1)

Call:
lm(formula = y[T1:T] ~ ses_f_c + car_f_hw - 1)

Coefficients:
 ses_f_c  car_f_hw
 -0.5426    1.5472
```

Note: Coefficients (weights) add up to 1. But, we see negative weights. In general, we use a constrained regression, forcing parameters to be between 0 and 1 (& non-negative). But, h=25 delivers not a lot of observations to do non-linear estimation. ¶

- Remarks:

- To get weights, we do not include a constant. Here, we are assuming unbiased forecasts. If the forecasts are biased, we include a constant.
- To account for potential correlation of errors, we can allow for ARMA residuals or include y_{T+l-1} in the regression.
- Time varying weights are also possible.

Question: Should weights matter? Two views:
- Simple averages outperform more complicated combination techniques.
- Sampling variability may affect weight estimates to the extent that the combination has a larger MSE.

Combination of Forecasts: Final Remarks

- Since Bates and Granger (1969) and Granger and Ramanathan (1984), combination weights have generally been chosen to minimize a symmetric, squared-error loss function.

- But, asymmetric loss functions can also be used. Elliot and Timmermann (2004) allow for general loss functions (and distributions). They find that the optimal weights depend on higher order moments, such a skewness.

- It is also possible to forecast quantiles and combine them. We will not explore these issues in more detail in this class.

Lecture 10 – Volatility Models

Linear and Non-linear Models

So far, we have focused on linear models. We have relied on Assumption (**A1**), where the relation between y_t & X_t is given by:

$$y_t = X_t\beta + \varepsilon_t, \qquad \varepsilon_t \sim i.i.d.\ \mathrm{D}(0, \sigma^2)$$

There are, however, many relationships in finance that are intrinsically non-linear: The payoffs to options are non-linear in some of the input variables, for example, S_t; investors' willingness to trade off returns and risks are also non-linear; CEO compensation that depends on thresholds and with a big option component are also non-linear.

The textbook of Campbell *et al.* (1997) defines a non-linear data generating process as one where the current value of y_t is related non-linearly to current and previous values of the error term, ε_t:

$$y_t = f(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots)$$

where ε_t is *i.i.d.* and *f* is a non-linear function.

A friendlier and slightly more specific definition of a non-linear model is given by the equation

$$y_t = g(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots) + \varepsilon_t\, \sigma^2(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots)$$

where g is a function of past error terms only, and σ^2 can be interpreted as a variance term, since it is multiplied by the current value of the error.

Cases
- *N*on-linear in mean only: $g(\bullet)$ = non-linear & $\sigma^2(\bullet) = \sigma^2$
- *N*on-linear in variance only: $g(\bullet)$ = linear & $\sigma^2(\bullet) \neq$ non-linear $g(\bullet)$
- *N*on-linear in mean and variance: both $g(\bullet)$ & $\sigma^2(\bullet)$ are non-linear.

Most popular non-linear models in finance: The ARCH models, where we model a time-varying variance as a function of past ε_t's.

ARCH Models

Until the early 1980s econometrics had focused almost solely on modeling the conditional means of time series, conditioning on information set at time t, I_t:

$$y_t = \mathrm{E}[y_t|\, I_t] + \varepsilon_t, \qquad \varepsilon_t \sim D(0, \sigma^2)$$

Suppose we have an AR(1) process:

$$y_t = \alpha + \phi\, y_{t-1} + \varepsilon_t.$$

The conditional mean is:

$\mathrm{E}[y_{t+1}|\, I_t] = \mathrm{E}_t[y_{t+1}] = \alpha + \phi\, y_t$

The unconditional mean and variance are:

$$\mathrm{E}[y_t] = \alpha/(1 - \phi) = \text{constant}$$
$$\mathrm{Var}[y_t] = \sigma^2/(1 - \phi^2) = \text{constant}$$

<u>Note</u>: Conditional mean is time varying; unconditional mean is not!

Similar idea for the variance. For the AR(1) process, we have:

- Conditional variance:

$$\text{Var}[y_{t+1}|I_t] = \text{E}_t[(y_{t+1} - \text{E}_t[y_{t+1}|I_t])^2] = \text{E}_t[\varepsilon_{t+1}^2]$$

- Unconditional variance:

$$\text{Var}[y_{t+1}] = \text{E}[(y_{t+1} - \text{E}[y_{t+1}])^2] = \sigma^2/(1 - \phi^2)$$

The unconditional variance measures overall uncertainty. In the AR(1) example, the information available at time *t, I_t*, plays no role:

$$\text{Var}[y_t] = \sigma^2/(1 - \phi^2)\ .$$

The conditional variance, Var[$y_t|I_t$], is a better measure of uncertainty at time *t*. It is a function of information at time *t, I_t*.

Notation: $E_t[Z_{t+1}] = E[Z_{t+1}|\ I_t]$

Summary:
- Unconditional variance measures the overall uncertainty.
- Conditional variance measures uncertainty at time *t*.

Remark: Conditional moments are time varying; unconditional moments are not!

ARCH Models: Stylized Facts of Asset Returns

(1) *Thick tails*: Leptokurtic (thicker tails than Normal).
(2) *Volatility clustering*: "Large changes tend to be followed by large changes of either sign."
(3) *Leverage Effects*: Tendency for changes in stock prices to be negatively correlated with changes in volatility.
(4) *Non-trading Effects, Weekend Effects*: When a market is closed, information accumulates at a different rate to when it is open –for example, the weekend effect, where stock price volatility on Monday is not three times the volatility on Friday.
(5) *Expected events*: Volatility is high at regular times such as news announcements or other expected events, or even at certain times of day –for example, less volatile in the early afternoon.
(6) *Volatility and serial correlation*: Inverse relationship between the two.
(7) *Co-movements in volatility*: Volatility is positively correlated across markets/assets.

We need a model that accommodates all these (non-linear) facts.

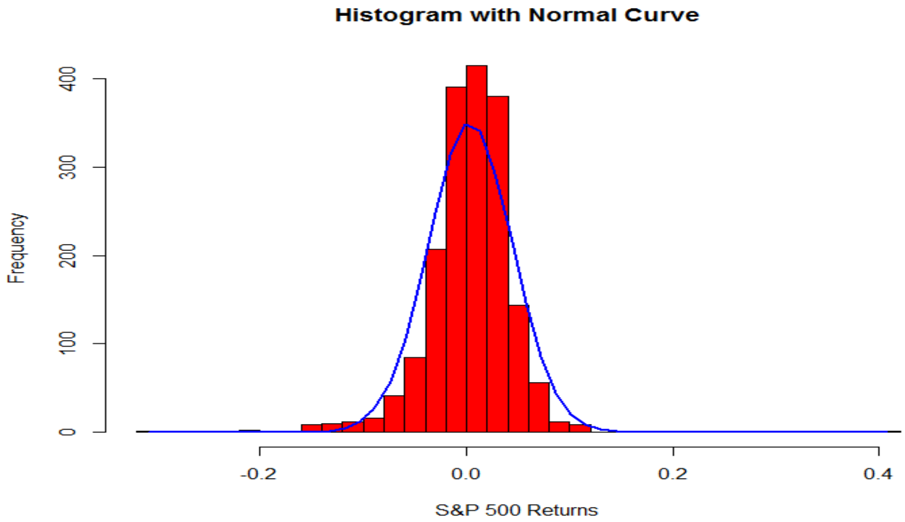
Stylized facts (1) and (2) form the basis of Volatility (ARCH) Models.

• Easy to check leptokurtosis (Stylized Fact #1).

Descriptive Statistics and Distribution for Monthly S&P500 Returns

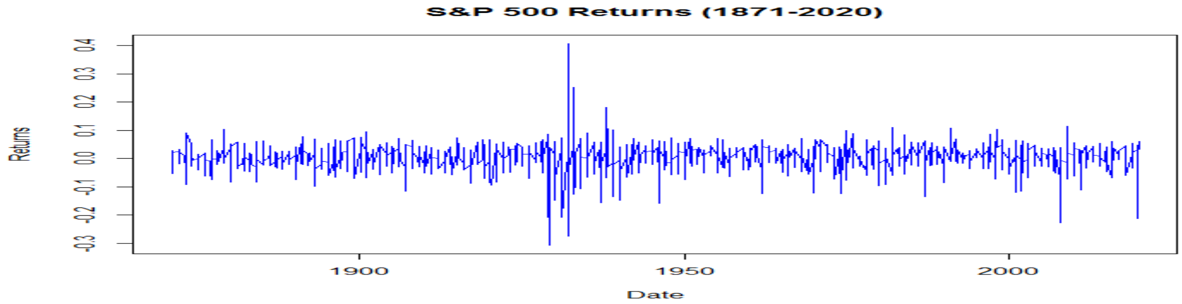
	Statistic

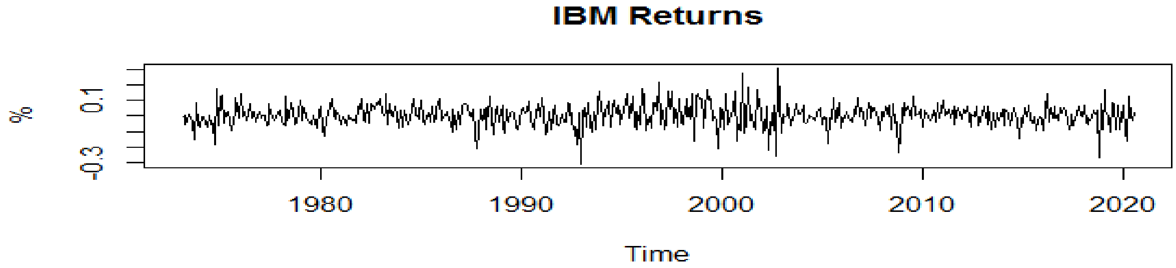
Mean (%)	0. 0585 (p-value: 0.0004)
Standard Dev (%)	0.0449
Skewness	-0.7294
Excess Kurtosis	2. 6406
Jarque-Bera	216.15 (*p-value*: <0.000001)



Note: Excess kurtosis greater than 0! Heavy tails are very common in financial time series.

- Easy to check Volatility Clustering (Stylized Fact #2)





Note: Periods with low changes, usually long, and periods of high changes, usually short. That is, volatility shows autocorrelation.

ARCH Models: Engle (1982)

We start with assumptions (**A1**) to (**A5**), but with a specific (**A3'**):

$$y_t = \gamma X_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, {\sigma_t}^2)$$

(**A3'**) $\quad {\sigma_t}^2 = Var_{t-1}[\varepsilon_t] = E_{t-1}[\varepsilon_t^2] = \omega + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_q \varepsilon_{t-q}^2$

which we can write, using the L operator, as:

$${\sigma_t}^2 = \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 = \omega + \alpha(L)\varepsilon^2$$

We can write the model in terms of an AR(q) for ε_t^2. Define

$\nu_t \equiv \varepsilon_t^2 - \sigma_t^2, \qquad$ -an error term for the variance.

Then,

$$\varepsilon_t^2 = \omega + \alpha(L)\varepsilon_t^2 + \nu_t$$

Correlated ε_t^2's: High (low) past ε_t^2's produce a high (low) ε_t^2 today.

The model

$${\sigma_t}^2 = \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 = \omega + \alpha(L)\varepsilon^2$$

is an AR(q) model for squared innovations, ε_t^2. We have the ARCH model: *Auto-Regressive Conditional Heteroskedasticity*.

The ARCH(q) model estimates the unobservable (*latent*) variance.

Non-negative constraints: Since we are dealing with a variance, we usually impose

$\omega > 0$ and $\alpha_i > 0 \qquad$ for all i.

Notation: $E_{t-1}[\varepsilon_t^2] = E[\varepsilon_t^2||I_{t-1}]$

Useful result: Since $E[\varepsilon_t] = 0$, then $E_{t-1}[\varepsilon_t^2] = \sigma_t^2$

ARCH Models: Unconditional Variance

The unconditional variance is determined by:

$$\sigma^2 = E[{\sigma_t}^2] = \omega + \sum_{i=1}^{q} \alpha_i E[\varepsilon_{t-i}^2] = \omega + \sum_{i=1}^{q} \alpha_i \sigma^2$$

That is,

$$\sigma^2 = \frac{\omega}{1 - \sum_{i=1}^{q} \alpha_i}$$

To obtain a positive σ^2, we impose another restriction: $(1 - \sum_{i=1}^{q} \alpha_i) > 0$

Example: ARCH(1)

$$Y_t = \beta X_t + \varepsilon_t, \qquad \varepsilon_t \sim N(0, {\sigma_t}^2)$$
$${\sigma_t}^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 \qquad \Rightarrow \sigma^2 = \frac{\omega}{1 - \alpha_1}$$

We need to impose restrictions: $\omega > 0$, $\alpha_1 > 0$, & $(1 - \alpha_1) > 0$. ¶

ARCH Models: Leptokurtosis

Errors may be serially uncorrelated, but they are not independent: There will be volatility clustering, which produces fat tails.

We want to calculate the kurtosis of the errors:

$$\kappa(\varepsilon_t) = E[\varepsilon_t^4]/E[\varepsilon_t^2]^2$$

We define standardized errors: $z_t = \frac{\varepsilon_t}{\sigma_t}$

They have conditional mean zero and a time invariant conditional variance equal to 1. That is, $z_t \sim N(0, 1)$.

From the definition of z_t we have: $\varepsilon_t = z_t \sigma_t$

Now, we compute the fourth (also central, since E[ε_t]=0) moment:

$$E[\varepsilon_t^4] = E[z_t^4] E[\sigma_t^4]$$

Then, using Jensen's inequality:

$$E[\varepsilon_t^4] = E[z_t^4]E[\sigma_t^4] > E[z_t^4]E[\sigma_t^2]^2 = E[z_t^4]E[\varepsilon_t^2]^2$$
$$= 3\, E[\varepsilon_t^2]^2$$

$$\kappa(\varepsilon_t) = E[\varepsilon_t^4]/E[\varepsilon_t^2]^2 > 3.$$

where we have used the fact that since E[ε_t] = 0, then E[ε_t^2] = E[σ_t^2].

Technical point: It can be shown that for an ARCH(1), the 4th moment for an ARCH(1):

$$\kappa(\varepsilon_t) = \frac{3(1-\alpha^2)}{1-3\alpha^2} \qquad \text{if } 3\alpha^2 < 1.$$

More convenient, but less intuitive, presentation of the ARCH(1) model:

$$y_t = \gamma X_t + \varepsilon_t$$
$$\varepsilon_t = \sigma_t \upsilon_t, \qquad \upsilon_t \sim D(0,\ 1)$$

that is, υ_t is *i.i.d.* with mean 0, and Var[υ_t]=1. Since υ_t is *i.i.d.,* then:

$$E_{t-1}[\varepsilon_t^2] = E_{t-1}[\sigma_t^2 \upsilon_t^2] = E_{t-1}[\sigma_t^2]\, E_{t-1}[\upsilon_t^2] = \omega + \alpha_1 \varepsilon_{t-1}^2$$

which delivers the AR(1) representation for ε_t^2.

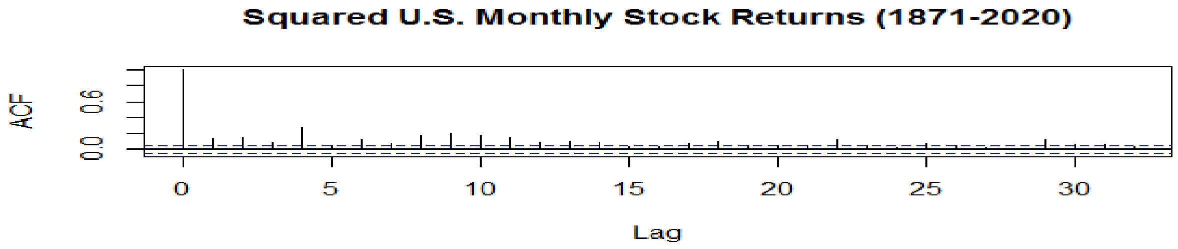
Also, if we assume υ_t is normally distributed, then

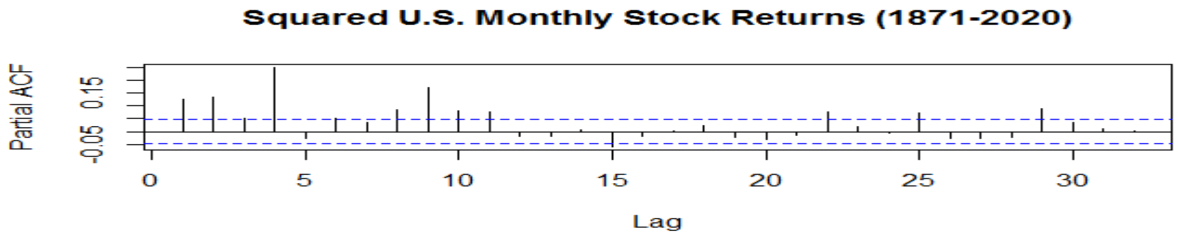
$$\varepsilon_t \sim N(0, \sigma_t^2).$$

GARCH Model: Bollerslev (1986)

An early technique to determine q was to look at the ACF/PACF for squared returns, ε_t^2, which usually determined a very large q.

Example: We calculate the ACF and PACF for the squared of the **U.S. monthly stock returns** (1871-2020).





Note: Highly autocorrelated squared returns. To accommodate the long autocorrelations, we use large q.

This result is not surprising, σ_t^2 is a very persistent process. Persistent processes can be captured with an AR(p), where p is large. This is not efficient.

Following the idea of an ARMA process, we can use a more parsimonious representation of the ARCH model: The Generalized ARCH model or GARCH(q, p):

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-j}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$
$$= \omega + \alpha(L)\varepsilon^2 + \beta(L)\sigma^2$$

which can be shown it is an ARMA(max(p,q), p) model for the squared innovations.

Popular GARCH model: GARCH(1,1):

$$\sigma_{t+1}^2 = \omega + \alpha_1 \varepsilon_t^2 + \beta_1 \sigma_t^2$$

with an unconditional variance: Var[ε_t^2] = $\sigma^2 = \omega / (1 - \alpha_1 - \beta_1)$.

$\Rightarrow$ Restrictions: $\omega > 0$, $\alpha_1 > 0$, $\beta_1 > 0$; $(1 - \alpha_1 - \beta_1) > 0$.

Technical details: This is *covariance stationary* if all the roots of

$$\alpha(L) + \beta(L) = 1$$

lie outside the unit circle. For the GARCH(1,1) this amounts to

$$\alpha_1 + \beta_1 < 1.$$

Question: What should be the order of the GARCH Model?
We should use enough lags to make sure the residuals do not have any more autocorrelation in the square residuals.

If the order of GARCH process is well determined, the ACF/PACF for ε_t^2 should show no significant autocorrelations.

We can add lags until the tests for ARCH structure in the squared residuals, discussed later, are not longer significant.

- A GARCH(1,1) is a very good starting point.

GARCH-X

In the GARCH-X model, exogenous variables are added to the conditional variance equation.

Consider the GARCH(1,1)-X model:

$$\sigma_t^2 = \omega + \alpha_1\varepsilon_t^2 + \beta_1\sigma_{t-1}^2 + \delta\, f(X_{t-1}, \theta),$$

where $f(X_t, \theta)$ is strictly positive for all t. Usually, X_t, is an observed economic variable or indicator, for example, a liquidity index, and $f(.)$ is a non-linear transformation, which should be non-negative.

Examples: We can use 3-mo T-bill rates for modeling stock return volatility, or interest rate differentials between countries to model FX return volatility.

The US congressional budget office uses inflation in an ARCH(1) model for interest rate spreads. ¶

ARCH Estimation: MLE

All of these models can be estimated by maximum likelihood. First we need to construct the sample likelihood.

Since we are dealing with dependent variables, we use the conditioning trick to get the joint distribution:

$$f(y_1, y_2, \dots, y_T; \theta) = f(y_1|x_1; \theta) * f(y_2|y_1, x_2, x_1; \theta) * f(y_3|y_2, y_1, x_3, x_2, x_1; \theta) *$$
$$* \ldots * \ f(y_T|y_{T-1}, \dots, y_1, x_{T-1}, \ \dots, x_1; \theta).$$

Taking logs:

$$\begin{aligned} L = \log(f(y_1, y_2, \dots, y_T; \theta)))) &= \log(f(y_1|x_1; \theta)) + \log(f(y_2|y_1, x_2, x_1; \theta) \\ &\quad + \;\dots\; + \log(f(y_T|y_{T-1}, \dots, y_1, x_{T-1}, \dots, x_1; \theta)) \\ &= \textstyle\sum_{t=1}^{T} \log(f(y_t|Y_{t-1}, X_t; \theta)) \end{aligned}$$

We maximize this function with respect to the *k* mean parameters (**γ**) and the *m* variance parameters (ω, **α**, **β**).

Example: ARCH(1) model.

Mean equation: $y_t = \boldsymbol{X}_t \boldsymbol{\gamma} + \varepsilon_t,$ ε_t ~ N(0, σ_t^2)

Variance equation: $\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2$

We write the pdf for the normal distribution,

$$f(\varepsilon_t|\gamma, \omega, \alpha_1) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left[-\frac{\varepsilon_t^2}{2\sigma_t^2}\right] = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left[-\frac{(y_t - X_t\gamma)^2}{2\sigma_t^2}\right]$$

We form the likelihood $\mathcal{L}$ (the joint pdf):

$$\mathcal{L} = \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right) = (2\pi)^{-T/2} \prod_{t=1}^{T} \frac{1}{\sqrt{\sigma_t^2}} \exp\left(-\frac{\varepsilon_t^2}{2\sigma_t^2}\right)$$

We take logs to form the log likelihood, $L = \log \mathcal{L}$:

$$L = \sum_{t=1}^{T} \log(f_t) = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \log(\sigma_t^2) - \frac{1}{2}\sum_{t=1}^{T} {\varepsilon^2}_t/\sigma_t^2$$

Then, we maximize L with respect to **θ** = (γ, ω, α_1) the function L.

$$L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \log(\omega + \alpha_1\varepsilon_{t-1}^2) - \frac{1}{2}\sum_{t=1}^{T} {\varepsilon^2}_t/(\omega + \alpha_1\varepsilon_{t-1}^2)$$

Taking derivatives with respect to **θ** = (ω, α₁, **γ**), where **γ** is a vector of *k* mean parameters:

$$\frac{\partial L}{\partial \omega} = -\textstyle\sum_{t=1}^{T} 1/(\omega + \alpha_1\varepsilon_{t-1}^2) - (-1/2)\sum_{t=1}^{T} \varepsilon_t^2/\,(\omega + \alpha_1\varepsilon_{t-1}^2)^2$$

$$\frac{\partial L}{\partial \alpha_1} = -\textstyle\sum_{t=1}^{T} \varepsilon_{t-1}^2/(\omega + \alpha_1\varepsilon_{t-1}^2) - (-1/2)\sum_{t=1}^{T} \varepsilon_t^2\varepsilon_{t-1}^2/\,(\omega + \alpha_1\varepsilon_{t-1}^2)^2$$

$$\frac{\partial L}{\partial \boldsymbol{\gamma}} = -\textstyle\sum_{t=1}^{T} \boldsymbol{X'}_t\varepsilon_t\,/\sigma_t^2$$ (*k*x1 vector of derivatives)

We form the f.o.c.; that is, we write the first derivative vectors as $\frac{\partial L}{\partial \boldsymbol{\theta}}$ and, then, set it equal to 0:

$\frac{\partial L}{\partial \boldsymbol{\theta}}$ = S(yₜ, **θ**) = 0 -a (*k*+2) system of equations.

The vector of first derivatives is called the score vector, S(yₜ, **θ**).

Take the last f.o.c., the *k*x1 vector, $\frac{\partial L}{\partial \boldsymbol{\gamma}} = 0$:

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\gamma}} &= -\textstyle\sum_{t=1}^{T} \boldsymbol{X'}_t\varepsilon_t\,/\sigma_{t,MLE}^2 = \sum_{t=1}^{T} \boldsymbol{X'}_t(y_t - \boldsymbol{X}_t\boldsymbol{\gamma}_{MLE})/\sigma_{t,MLE}^2 = 0 \\ &= \textstyle\sum_{t=1}^{T} \frac{\boldsymbol{X'}_t}{\sigma_{t,MLE}}\left(\frac{y_t}{\sigma_{t,MLE}} - \frac{\boldsymbol{X}_t}{\sigma_{t,MLE}}\boldsymbol{\gamma}_{MLE}\right) = 0 \end{aligned}$$

The last equation shows that MLE is GLS for the mean parameters, $\boldsymbol{\gamma}$: each observation is weighted by the inverse of $\sigma_{t,MLE}$.

We have a (*k*+2) system. It is a non-linear system. The system is solved using numerical optimization (usually, with the Newton-Raphson method). ¶

Technical Note: If the conditional density for ε_t is well specified and θ_0 (the true parameter) belongs to the parameter space, Ω, then

$$\boldsymbol{T^{\frac{1}{2}}\left(\hat{\theta}-\theta_0\right) \rightarrow N\left(0, A_0^{-1}\right)}, \quad \text{where } A_0 \;= \boldsymbol{T^{-1}\sum_{t=1}^{T}\frac{\partial S_t(y_t, \theta_0)}{\partial \theta}}$$

A_0 is the matrix of second derivatives of the log likelihood, *L***.** It is called the *Hessian*. In general, it is difficult to numerically compute and make sure it is positive definite (so it can be inverted), especially when the dimensions are big.

• There a lot of computational tricks to compute a Hessian that is invertible, the most popular algorithm is the Broyden–Fletcher–Goldfarb–Shanno, or "BFGS."

ARCH Estimation: MLE – Standard Errors

Under the correct specification assumption, $A_0 = B_0$, where

$$B_0 = T^{-1}\sum_{t=1}^{T} E[S_t(y_t, \theta_0), S_t(y_t, \theta_0)']$$

We estimate A_0 and B_0 by replacing $\boldsymbol{\theta}_0$ by its estimated MLE value, $\boldsymbol{\theta}_{MLE}$.

The estimator B_0 has a computational advantage over A_0.: Only first derivatives are needed. But $A_0 = B_0$ only if the distribution is correctly specified. This is very difficult to know in practice.

Common practice in empirical studies: Assume the necessary regularity conditions are satisfied.

ARCH Estimation: Numerical Optimization

In general, we have a (*k*+*m* x *k*+*m*) system; *k* mean parameters and *m* variance parameters. But, it is a non-linear system. We use *numerical optimization,* which are methods that search over the parameter space looking for the values that maximize the log likelihood function.

In R, the function *optim* does numerical optimization. It minimizes any non-linear function. It needs as inputs:

- Initial values for the parameters, θ_0.
- Function to be minimized (includes the GARCH process).
- Data used.
- Other optional inputs: Choice of method, hessian calculated, etc.

Example: *optim*(theta0, log_lik_garch11, data=z, method="BFGS", hessian=TRUE)
theta0 = initial values
log_lik_garch11 = function to be minimized. ¶

- Initial values:

- Numerical optimization needs initial values for θ, say θ_0. It is very common to find that the optimization is sensitive to the initial values. It is a good practice to try different sets of initial values.

We want to avoid selecting a local maximum:

- Initial values (continuation):

- Numerical optimization needs initial values for θ, say θ_0. It is very common to find that the optimization is sensitive to the initial values. It is a good practice to try different sets of initial values.

We want to avoid selecting a local maximum:

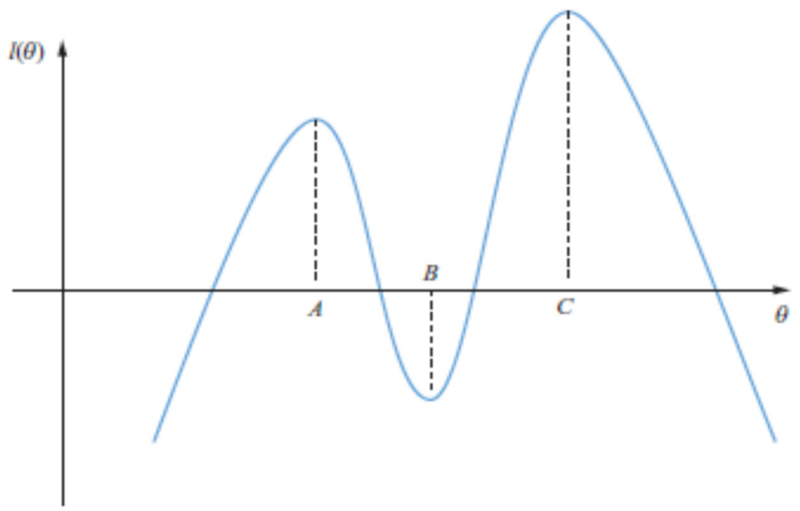


Figure 9.2 The problem of local optima in maximum likelihood estimation

- Given the autoregressive structure in σ_t^2, and sometimes we have AR(p) in the mean, we need to make assumptions about σ_0 and the $\varepsilon_0, ..., \varepsilon_q$ (and $\varepsilon_0, \varepsilon_1, ..., \varepsilon_p$ if we assume an AR(p) process for the mean).

Usual assumptions: σ_0 = unconditional SD; $\varepsilon_0 = \varepsilon_1 = ... = \varepsilon_p = 0$.

- Alternatively, we can take σ_0 (and $\varepsilon_0, \varepsilon_1, ..., \varepsilon_p$) as parameters to be estimated (it can be computationally more intensive and estimation can lose power.)

ARCH Estimation: MLE – Example (in R)

Log likelihood of AR(1)-GARCH(1,1) Model:

```
log_lik_garch11 <- function(theta, data) {
 mu <- theta[1]; rho1 <- theta[2]; omega <- abs(theta[3]); alpha1 <- abs(theta[4]); beta1 <-
abs(theta[5]);
chk0 <- (1 - alpha1 - beta1)
 r <- ts(data)
 n <- length(r)
 u <- vector(length=n);  u <- ts(u)
 u[1] = 0
for (t in 2:n)
 {u[t] = r[t] - mu - rho1*r[t-1]}                          # this setup allows for ARMA in
mean
 h <- vector(length=n);  h <- ts(h)
 h[1] = omega/chk0                                  # set initial value for h[t] series
if (chk0==0) {h[1]=.000001}                                # check to avoid dividing by 0
 for (t in 2:n)
 {h[t] = abs(omega + alpha1*(u[t-1]^2) + beta1*h[t-1])
if (h[t]==0) {h[t]=.00001} }                        # check to avoid log(0)
return(-1*sum(- 0.5 * log(abs(h[2:n])) - 0.5 * (u[2:n]^2)/abs(h[2:n])))
}
# I use optim to minimize a function, to maximize multiply by -1
```

Example 1: GARCH(1,1) model for **changes in CHF/USD**. We will use R function *optim* (*mln* can also be used) to maximize the likelihood function.

```
PPP_da <-
read.csv("http://www.bauer.uh.edu/rsusmel/4397/ppp_2020_m.csv",head=TRUE,sep=",")
x_chf <- PPP_da$CHF_USD                             # CHF/USD 1971-2020 monthly data
T <- length(x_chf)
z <- log(x_chf[-1]/x_chf[-T])
theta0 = c(-0.002,  0.026,   0.001,   0.19,   0.71)           # initial values
ml_2 <- optim(theta0,  log_lik_garch11, data=z, method="BFGS", hessian=TRUE)
logL_g11 <- log_lik_garch11(ml_2$par, z)            # value of log likelihood
logL_g11
ml_2$par                                             # estimated parameters
I_Var_m2 <- ml_2$hessian
eigen(I_Var_m2)                                     # check if Hessian is pd.
sqrt(diag(solve(I_Var_m2)))                         # parameters SE
chf_usd <- ts(z, frequency=12, start=c(1971,1))
plot.ts(chf_usd)                                           # time series plot of data

> logL_g11                                                        # Log likelihood value
[1] –1745.197
> ml_2$par                                                        # Extract from ml_2 function
parameters
```

```
[1] -0.0021051742  0.0260003610  0.00012375  0.1900276519  0.7100718082
> I_Var_m2 <- ml_2$hessian                                    # Extract Hessian (matrix of
2nd derivatives)
> eigen(I_Var_m2)                                             # Check if Hessian is pd to
invert.
eigen() decomposition
$values                                                       # Eigenvalues: if positives
=> Hessian is pd
[1] 1.687400e+08 6.954454e+05 7.200084e+03 5.120984e+02 2.537958e+02
$vectors
             [,1]              [,2]              [,3]              [,4]
                               [,5]
[1,]  4.265907e-05  9.999960e-01 -0.0011397586  0.0018331957 -0.0018541203
[2,] -3.333961e-06 -2.188159e-03 -0.0010048203  0.9769058449 -0.2136566699
[3,]  9.999998e-01 -4.223001e-05 -0.0003544245  0.0001291633  0.0005770707
[4,] -3.599974e-06 -1.702277e-03 -0.8603563865 -0.1097470278 -0.4977344477
[5,] -6.893837e-04  6.416141e-04 -0.5096905472  0.1833226197  0.8405994743

> sqrt(diag(solve(I_Var_m2)))                                      # Invert Hessian:
Parameters Var on diag
[1] 1.203690e-03 4.419049e-02 7.749756e-05 5.014454e-02 3.955411e-02
> t_stats <- ml_2$par/sqrt(diag(solve(I_Var_m2)))
> t_stats
[1] -1.7489333  0.5883701  1.5967743  3.7895984 17.9519078
```

Summary for CHF/USD changes

$e_{f,t}$ = [log(S_t) - log(S_{t-1})] = a_0 + a_1 $e_{f,t-1}$ + ε_t, $\varepsilon_t \mid I_t \sim$ N(0, σ_t^2)

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

- *T*: 562 (January 1971 - July 2020, monthly).

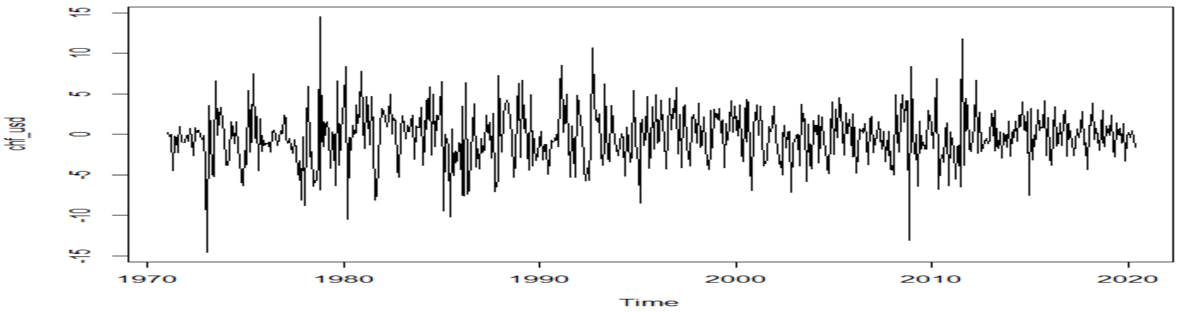
The estimated model for $e_{f,t}$ is given by:

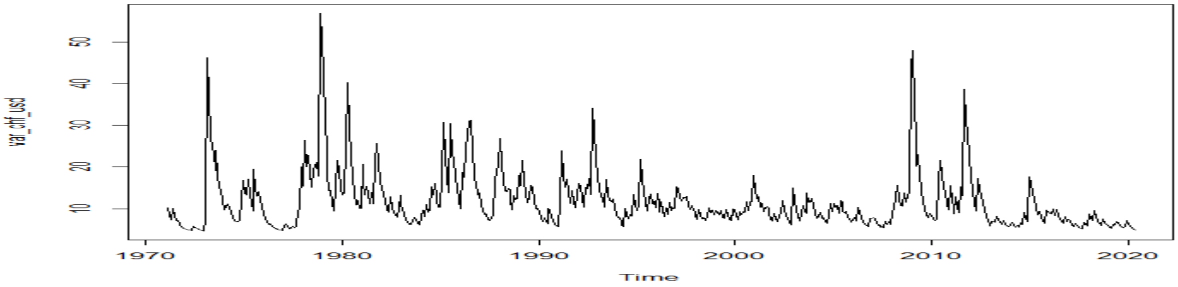
$e_{f,t}$ = **-0.00211** + **0.02600** $e_{f,t-1}$,
(.0012) (0.044)

σ_t^2 = **0.00012** + **0.19003** ε_{t-1}^2 + **0.71007** σ_{t-1}^2.
(0.00096)* (0.050)* (0.040)*

Unconditional σ^2 = **0.00012** /(1- **0.19003** - **0.71007**) = 0.001201201 Log likelihood: 1745.197

Note: $\alpha_1 + ß_1$ = **.90** < 1. (Persistent.) ¶





Example 2: Using Robert Shiller's monthly data set for the S&P 500 (1871:Jan - 2020:Aug, T=1,795), we estimate an AR(1)-GARCH(1,1) model:

$r_t = [\log(P_t) - \log(P_{t-1})] = \mathbf{a_0} + \mathbf{a_1}\, r_{t-1} + \varepsilon_t, \qquad \varepsilon_t \,|\, I_{t-1} \sim N(0, \sigma_t^2)$

$\sigma_t^2 = \boldsymbol{\omega} + \boldsymbol{\alpha_1}\, \varepsilon_{t-1}^2 + \boldsymbol{\beta_1}\, \sigma_{t-1}^2$

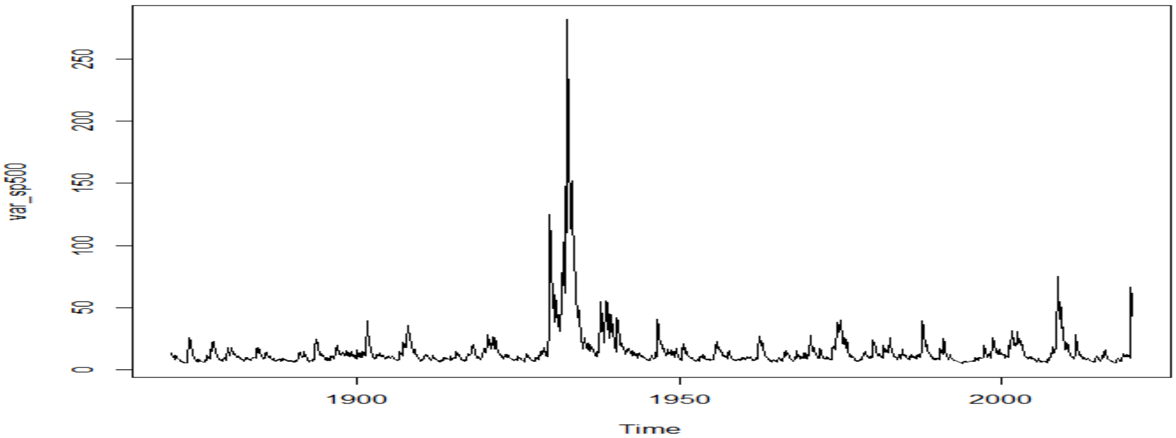
The estimated model for s_t is given by:

$r_t =$ **0.338** + **0.278** r_{t-1},
(.08)* (0.025)*

$\sigma_t^2 =$ **0.756** + **0.126** ε_{t-1}^2 + **0.826** σ_{t-1}^2.
(0.151)* (0.017)* (0.021)*

Unconditional σ^2 = **0.756** /(1 - **0.126** - **0.826**) = 15.4630
Log likelihood: 4795.08

Note: $\alpha_1 + \beta_1$ = **.952** < 1. (Very persistent.)



Above, we plot the time-varying variance. Certain events are clearly different, for example, the 1930 great depression, with a peak variance of 282 (18 times unconditional variance!). The covid-19 volatility similar to the 2008-2009 financial crisis recession. ¶

GARCH: Forecasting and Persistence

Consider the forecast in a GARCH(1,1) model:

$$\sigma_{t+1}^2 = \omega + \alpha_1 \varepsilon_t^2 + \beta_1 \sigma_t^2 = \omega + \sigma_t^2\,(\alpha_1 z_t^2 + \beta_1) \qquad (\varepsilon_t^2 = \sigma_t^2 z_t^2)$$

Taking expectation at time t

$$E_t[\sigma_{t+1}^2] = \omega + \sigma_t^2(\alpha_1 1 + \beta_1)$$

Then, by repeated substitutions:

$$E_t[\sigma_{t+j}^2] = \omega * [\textstyle\sum_{i=0}^{j-1} (\alpha_1 + \beta_1)^i] + \sigma_t^2\,(\alpha_1 + \beta_1)^j$$

Assuming $(\alpha_1 + \beta_1) < 1$, as j → ∞, the forecast reverts to the unconditional variance: $\sigma^2 = \omega/(1 - \alpha_1 - \beta_1)$.

When $\alpha_1 + \beta_1 = 1$, today's volatility affect future forecasts forever:

$$E_t[\sigma_{t+j}^2] = \sigma_t^2 + j\omega$$

Example 1: We want to forecast next month (September 2020) variance for CHF/USD changes. Recall we estimated σ_t^2:

$$\sigma_t^2 = \mathbf{0.00012} + \mathbf{0.19003}\,\varepsilon_{t-1}^2 + \mathbf{0.71007}\,\sigma_{t-1}^2.$$

getting $\sigma_{2020:9}^2$ = **0.003672220** (=$\sigma_{2020:9}$ = sqrt(**0.00367**) = 6.1%)

We based the $\sigma_{2020:10}^2$ forecast on:

$$E_t[\sigma_{t+j}^2] = \omega * [\textstyle\sum_{i=0}^{j-1} (\alpha_1 + \beta_1)^i] + \sigma_t^2(\alpha_1 + \beta_1)^j$$

Then, $(\alpha_1 + \beta_1)$ = **0.190** + **0.710** = **0.900**

$$E_{2020:9}[\sigma_{2020:10}^2] = \mathbf{0.00012} + \mathbf{0.00367} * (\mathbf{0.9}) = \mathbf{0.003423}$$

We also forecast $\sigma_{2020:12}^2$

$E_{2020:9}[\sigma^2_{2020:12}]$ = **0.00012** * {1+ (**0.9**)+ (**0.9**)^2} + **0.00367** * (**0.9**)^3 = 0.00300063

We forecast volatility for March 2021:

$E_{2020:6}[\sigma^2_{2021:03}]$ = **0.00012** * {1 + (**0.9**) + (**0.9**)^2+ … + (**0.9**)^5} + **0.00367** * (**0.9**)^6
= 0.002512659

Remark: We observe that as the forecast horizon increases (j → ∞), the forecast reverts to the unconditional variance:

$\omega/(1 - \alpha_1 - \beta_1)$ = **0.00012**/(1 - **0.9**) = 0.0012
⇒ σ = sqrt(0.0012) = 0.0346 (3.46% ≈ close to sample SD = **3.36%**). ¶

Example 2: On August 2020, we forecast the December's variance for the S&P500 changes. Recall we estimated σ^2_t:

$$\sigma^2_t = \mathbf{0.756} + \mathbf{0.125}\, \varepsilon^2_{t-1} + \mathbf{0.826}\ \sigma^2_{t-1},$$

getting $\sigma^2_{2020:8}$ = **43.037841**

We based the $\sigma^2_{2020:12}$ forecast on:

$$E_t[\sigma^2_{t+j}] = \omega * [\textstyle\sum_{i=0}^{j-1}(\boldsymbol{\alpha_1} + \boldsymbol{\beta_1})^i] + \sigma^2_t\ (\boldsymbol{\alpha_1} + \boldsymbol{\beta_1})^j$$

Then, since $(\boldsymbol{\alpha_1} + \boldsymbol{\beta_1})$ = **0.952**

$E_{2020:8}[\sigma^2_{2020:12}]$ = **0.756** * {1+ (**0.952**) + (**0.952**)^2 + (**0.952**)^3} +
+ **43.037841** * (**0.952**)^4 = 38.02797

Lower variance forecasted for the end of the year, but still far from the unconditional variance of **15.4**. ¶

GARCH: Forecasting – Application to VaR

Example: In September 2020, Swiss Cruises wants to construct a VaR-mean for the USD 1 M receivable in 30 days (October). Data

Receivable: USD 1 M
$S_{t=2020:9}$ = **1.45 CHF/USD**
$e_{f,t=2020:9}$ = **0.01934126**
$TE_{t=2020:9}$ = USD 1M * **1.45 CHF/USD** = **CHF 1.45M**.

$E_{2020:9}[\sigma^2_{2020:10}]$ = **0.003423** ⇒ sqrt(**0.003423**) = 0.05851 (5.85%)
VaR-mean(.99) = **CHF 1.45M** * {$E_{2020:9}[e_{f,t=2020:10}]$ – 2.33 * sqrt($E_{2020:9}[\sigma^2_{2020:10}]$)}

$E_{2020:9}[e_{f,t=2020:10}]$ = **-0.00211** + **0.026** * $e_{f,t=2020:9}$
= **-0.00211** + **0.026** * **0.01934126** = **-0.001607**

VaR-mean(.99) = **CHF 1.45M** * (**-0.001607** – 2.33 * sqrt(**0.003423**))
= **CHF -0.1999941 M**

Interpretation of VaR-mean: Relative to today's valuation (or *expected valuation*, according to RWM), the maximum *expected loss* with a 99% "chance" is **CHF -0.20 M**.

We also derive this value, using the sample mean and sample SD:
sample mean = **-0.00259**
sample SD = **0.033357**

⇒ VaR-mean(.99)= **CHF 1.45M** * * (**-0.00259** - 2.33 * **0.033357**) =
= **CHF - 0.1164491**

<u>Remark</u>: The GARCH forecast reflects the higher than average uncertainty in 2020:9 (Covid-19, presidential elections). ¶

GARCH: Rugarch Package

GARCH estimation requires numerical optimization, which is dependent on initial values. The R package does a good job at estimating ARMA-GARCH models, allowing for different models and performing a lot of specification tests.

You need to specify the model ("*specs*") first, for example, you want to estimate an AR(1)-GARCH(1,1) with a constant in the mean. Then, you estimate the model with the *ugarchfit* command.

Example: We estimate an AR(1)-GARCH(1,1) for the historical U.S. monthly returns (1871 – 2020, T = 1,797).

```
x <- lr_p                                   # SP500 long run monthly returns
library(rugarch)                            # You need to install package first!
mod_gar <- ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
mean.model = list(armaOrder = c(1, 0), include.mean = TRUE))
ar1_garch11 <- ugarchfit(spec=mod_gar, data=lr_p)
```

```
> ar1_garch11
*---------------------------------*
*        GARCH Model Fit          *
*---------------------------------*
Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(1,0,0)
Distribution    : norm
Optimal Parameters
------------------------------------
         Estimate      Std. Error  t value Pr(>|t|)
mu       0.004695   0.001052   4.4651    8e-06
ar1      0.277567   0.025120  11.0496    0e+00
omega    0.000075   0.000015   4.8968    1e-06
alpha1   0.126715   0.017529   7.2289    0e+00
beta1    0.826194   0.020600  40.1061    0e+00
```

```
Robust Standard Errors:
          Estimate  Std. Error  t value Pr(>|t|)
mu      0.004695        0.001145   4.1018 0.000041
ar1     0.277567        0.022948  12.0957 0.000000
omega 0.000075        0.000021   3.6307 0.000283
alpha1 0.126715       0.026943   4.7031 0.000003
beta1   0.826194        0.028409  29.0821 0.000000
LogLikelihood : 3472.361
Information Criteria
------------------------------------

Akaike       -3.8591
Bayes        -3.8438
Shibata      -3.8591
Hannan-Quinn -3.8534

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                              statistic p-value
Lag[1]                       0.3178 0.57294
Lag[2*(p+q)+(p+q)-1][2]       2.5441 0.08393
Lag[4*(p+q)+(p+q)-1][5]       6.9210 0.02072          ⇒ Need to add more lags in mean.
d.o.f=1
H0 : No serial correlation
Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                             statistic p-value
Lag[1]                       0.1915  0.6617
Lag[2*(p+q)+(p+q)-1][5]       1.1353  0.8284
Lag[4*(p+q)+(p+q)-1][9]       1.6161  0.9455          ⇒ No evidence of extra ARCH lags.
```

IGARCH

Recall the technical detail: The standard GARCH model:

$$\sigma_t^2 = \omega + \alpha(L)\varepsilon^2 + \beta(L)\sigma^2$$

is covariance stationary if $\alpha(1) + \beta(1) < 1$.

But strict stationarity does not require such a stringent restriction
In the GARCH(1,1) model, if $\alpha_1 + \beta_1 = 1$, we have the Integrated GARCH (IGARCH) model.
In the IGARCH model, the autoregressive polynomial in the ARMA representation has a unit root: a shock to the conditional variance is "*persistent*."

Variance forecasts are generated with: $E_t[\sigma_{t+j}^2] = \sigma_t^2 + j\omega$

⇒ today's variance remains important for all future forecasts. This is persistence!

Variance forecasts are generated with: $E_t[\sigma_{t+j}^2] = \sigma_t^2 + j\omega$

That is, today's variance remains important for future forecasts of all horizons.

In practice (see previous Example 2 for the S&P 500 data), it is often found that $\alpha_1 + \beta_1$ are close to 1.

GARCH: Variations – GARCH-in-mean

The time-varying variance affects mean returns:
Mean equation: $y_t = X_t\gamma + \delta\,\sigma_t^2 + \varepsilon_t,$ $\varepsilon_t \sim$ N(0, σ_t^2)
Variance equation: $\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2$

We have a dynamic mean-variance relations. It describes a specific form of the risk-return trade-off.

Finance intuition says that δ has to be positive and significant. However, in empirical work, it does not work well: δ is not significant or negative.

GARCH: Variations – Asymmetric GJR

GJR-GARCH model – Glosten, Jagannathan & Runkle (JF, 1993):

$$\sigma_t^2 = \omega + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2 + \sum_{i=1}^{q}\gamma_i\varepsilon_{t-i}^2 * I_{t-i} + \sum_{j=1}^{p}\beta_j\sigma_{t-j}^2$$

where I_{t-1} = 1 if $\varepsilon_{t-1} < 0$;
= 0 otherwise.

Using the indicator variable I_{t-i}, this model captures sign (asymmetric) effects in volatility: Negative news ($\varepsilon_{t-1} < 0$) increase the conditional volatility (*leverage effect*).

The GARCH(1,1) version:

$$\sigma_t^2 = \omega + \alpha_1\varepsilon_{t-1}^2 + \gamma_1\,\varepsilon_{t-1}^2\,I_{t-1} + \beta_1\sigma_{t-1}^2$$

where I_{t-1} = 1 if $\varepsilon_{t-1} < 0$;
= 0 otherwise.

When $\varepsilon_{t-1} < 0$ $\Rightarrow \sigma_t^2 = \omega + (\alpha_1 + \gamma_1)\,\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2$
$\varepsilon_{t-1} > 0$ $\Rightarrow \sigma_t^2 = \omega + \alpha_1\,\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2$

This is a very popular variation of the GARCH models. The leverage effect is significant.

There is another variation, the Exponential GARCH, or EGARCH, that also captures the asymmetric effect of negative news on the conditional variance.

GARCH: Variations – NARCH

Non-linear ARCH model NARCH – Higgins and Bera (1992) and Hentschel (1995).

These models apply the Box-Cox-type transformation to the conditional variance:

$$\sigma_t^{\gamma} = \omega + \sum_{i=1}^{q} \alpha_i |\varepsilon_{t-i} - \kappa|^{\gamma} + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^{\gamma}$$

Special case: $\gamma = 2$ (standard GARCH model).

Note: The variance depends on both the size and the sign of the variance which helps to capture leverage type (asymmetric) effects.

ARCH Estimation: MLE – Regularity Conditions

Technical Note: The appeal of MLE is the optimal properties of the resulting estimators under ideal conditions. However, these ideal conditions, which are called "*regularity conditions*," are difficult to verify for ARCH models

• Block-diagonality

In many applications of ARCH models, the parameters can be partitioned into mean parameters, θ_1, and variance parameters, θ_2. Thus, the Information matrix (≈Hessian) is *block-diagonal.*

Not a bad result:

– Regression can be consistently done with OLS.

– Asymptotically efficient estimates for the ARCH parameters can be obtained on the basis of the OLS residuals.

But:

– Conventional OLS standard errors could be terrible.

– When testing for autocorrelation, in the presence of ARCH, the conventional Bartlett s.e. –$T^{-1/2}$– could seriously underestimate the true standard errors.

ARCH Estimation: Non-Normality

The basic GARCH model allows a certain amount of leptokurtosis. It is often insufficient to explain real world data.

Solution: Assume a distribution, other than the normal, that can produce fatter tails in the distribution.

• *t* Distribution - Bollerslev (1987)

The t distribution has a degrees of freedom parameter which allows greater kurtosis. The likelihood function for observation *t* is:

$$l_t = \ln(\Gamma(0.5(v+1))\Gamma(0.5v)^{-1}(v-2)^{-1/2}(1+z_t(v-2)^{-1})^{-(v+1)/2}) - 0.5\ln(\sigma^2{}_t)$$

where Γ is the gamma function and v is the degrees of freedom. As $\upsilon \rightarrow \infty$, this tends to the normal distribution.

ARCH: Testing

Standard BP test, where we test H_0: $\alpha_1 = \alpha_2 = ... = \alpha_q = 0$.

Steps:

– **Step 1.** (Same as BP's Step 1). Run OLS on DGP:

$y = X\,\beta + \varepsilon.$ Keep residuals, e_t.

– **Step 2.** (Auxiliary Regression). Regress e_t^2 on e_{t-1}^2, ..., e_{t-q}^2

$e_t^2 = \alpha_0 + \alpha_1\, e_{t-1}^2 + ... + \alpha_m\, e_{t-q}^2 + v_t.$ Keep R^2, say R_{e2}^2.

– **Step 3**. Compute the statistic:

$$LM = (T - q)\, R_{e2}^2 \xrightarrow{d} \chi_q^2.$$

Example: We do an ARCH Test with 4 lags, for the AR(1) residuals of log changes in the CHF/USD (T = 593):

```
yyy <- z;
T <- length(yyy)
xx_1 <- z[-T]
 yy <- z[-1]
fit2 <- lm(yy ~ xx_1 -1)
res_d <- fit2$residuals                                  # Step 1: extract residuals

p_lag <- 4
 e2_lag <- matrix(0, T-p_lag, p_lag)                # matrix to put lag e^2
 resid_r2 <- res_d^2
 a <-1
 while (a <= p_lag) {
   e2_lag[,a] <- resid_r2[a:(T-p_lag+a-1)]
 a <- a+1
 }

fit_lm2 <- lm(resid_r2[(p_lag+1):T] ~ e2_lag)            # Step 2: Auxiliary Regression
 r2_e1 <- summary(fit_lm2)$r.squared                     # extract R^2
 lm_t <- (T-p_lag)*r2_e1                                 # LM test: Sample size * R^2

> lm_t
[1] 17.08195
```

⇒ Reject H_0 (No ARCH) with a p-value of **0.001**. ¶

ARCH: Testing – Ignoring ARCH

In ARCH Models, testing as usual: LR, Wald, and LM tests. Suppose ARCH is detected, but ARCH is ignored. What are the consequences of ignoring ARCH?

• Ignoring ARCH

- Suppose y_t has an AR structure:

$$y_t = \gamma_0 + \phi_1\, y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \,|\, I_{t-1} \sim N(0, \sigma_t^2).$$

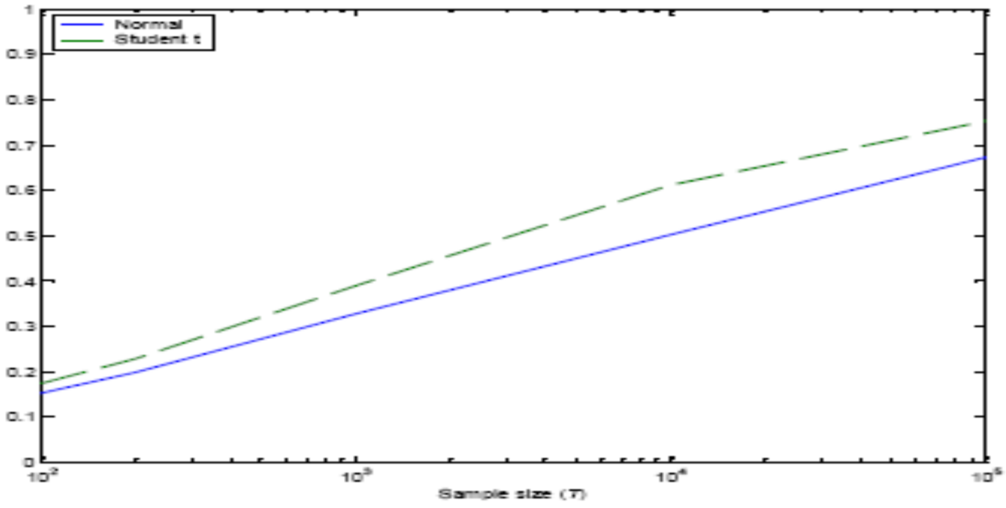
with ARCH structure in the error term, but ARCH is ignored. Then, we fit the AR(1) model using OLS.

- Simulations find that OLS t-test with no correction for ARCH spuriously reject H_0: $\phi_1 = 0$ with arbitrarily high probability for sufficiently large *T*.

- If White's (1980) SE are used, the results are better. NW SE help less.

Figure. From Hamilton (2008). Fraction of samples in which OLS *t-test* leads to rejection of H_0: $\phi_1 = 0$ as a function of *T* for regression with Normal errors (solid blue line) and Student's *t* errors (dashed green line).

<u>Note</u>: H_0 is actually true & the *t-test* is evaluated at the 5% level.



ARCH: Which Model to Use

Questions

1) Lots of ARCH models. Which one to use?

2) Choice of *p* and *q*. How many lags to use?

Hansen and Lunde (2004) compared lots of ARCH models:

- It turns out that the GARCH(1, 1) is a great starting model.
- Add a leverage effect for financial series and it's even better.
- A *t-distribution* is also a good addition.

RV Models: Intuition

The idea of realized volatility is to estimate the latent (unobserved) variance using the realized data, without any modeling. Recall the definition of sample variance:

$$s^2 = \frac{1}{(T-1)}\sum_{i=1}^{T}(x_i - \bar{x})^2$$

Suppose we want to calculate the daily variance for stock returns. We know how to compute it: we use daily information, for *T* days, and apply the above definition.

Alternatively, we use hourly data for the whole day (with *k* hours). Since hourly returns are very small, ignoring $\bar{\boldsymbol{x}}$ seems OK. We use $r_{t,\boldsymbol{i}}^2$ as the *i*^th^ hourly variance on day *t*. Then, we add $r_{t,\boldsymbol{i}}^2$ over the day:

$$Variance_t = \sum_{i=1}^{k} r_{t,i}^2$$

In more general terms, we use higher frequency data to estimate a lower frequency variance:

$$RV_t = \sum_{i=1}^{k} r_{t,\boldsymbol{i}}^2$$

where $r_{t,i}$ is the realized returns in (higher frequency) interval *i* of the (lower frequency) period *t*. We estimate the *t-frequency* variance, using *k i-intervals*. If we have daily returns and we want to estimate the monthly variance, then, *k* is equal to the number of days in a month.

It can be shown that as the interval *i* becomes smaller ($i \rightarrow 0$),

$$RV_t \rightarrow \text{Return Variation } [t-1,\, t].$$

That is, with an increasing number of observations we get an accurate measure of the latent variance.

RV Models: High Frequency

Note that RV is a model-free measure of variation –i.e., no need for ARCH-family specifications. The measure is called *realized variance* (RV). The square root of the realized variance is the *realized volatility* (RVol, RealVol):

$$RVol_t = sqrt(RV_t)$$

Given the previous theoretical result, RV is commonly used with intra-daily data, called *high frequency* (HF) data.

It lead to a revolution in the field of volatility, creating new models and new ways of thinking about volatility and how to model it.

We usually associate realized volatility with an observable proxy of the unobserved volatility.

RV Models: High Frequency – Tick Data

As mentioned above, the theory behind realized variation measures dictates that the sampling frequency, or *k* in the RV_t formula above, goes to ∞. Then, use the highest frequency available, say millisecond to millisecond returns.

Intra-daily data applications are the most common. But, when using intra-daily data, RV calculations are affected by microstructure effects: bid-ask bounce, infrequent trading, calendar effects, etc. $r_{t,i}$ does not look uncorrelated.

Example: The bid-ask bounce induces serial correlation in intra-day returns, which biases RV_t.

The usual solutions:

(1) Filter data using an ARMA model to get rid of the autocorrelations and/or dummy variables to get rid of calendar effects.

Then, used the filtered data to compute RV_t.

(2) Sample at frequencies where the impact of microstructure effects is minimized and/or eliminated.

We will follow solution (2).

RV Models: High Frequency – Practice

In intra-daily RV estimation, it is common to use 10' intervals. They have good properties. However, there are estimations with 1' intervals.

Some studies suggest using an *optimal* frequency, where optimal frequency is the one that minimizes the MSE.

Hansen and Lunde (2006) find that for very liquid assets, such as the S&P 500 index, a **5'** sampling frequency provides a reasonable choice. Thus, to calculate daily RV, we need to add **78** five-minute intervals.

Example: Based on TAQ (*Trade and Quote*) NYSE data, we use 5' realized returns to calculate 30' variances –i.e., we use six 5' intervals. Then, the 30' variance, or $RV_{t=30\text{-min}}$, is:

$$RV_{t=30-min} = \sum_{j=1}^{k=6} r_{t,j}^2, \qquad t = 1,2,\ldots.,T{=}15$$

$r_{t,j}$ is the 5' return during the jth interval on the half hour t. Then, we calculate 30' variances for the whole day –i.e., we calculate 13 variances, since the trading day goes from 9:30 AM to 4:00 PM.

The Realized Volatility, RVol, is:

$$RVol_{t=30-min} = \sqrt{RV_{t=30-min}}$$

Example: Below, we show the first transaction of the **SPY TAQ** (*Trade and Quote*) data (tick-by-tick *trade* data) on **January 2, 2014**.

SYMBOL	DATE	TIME	PRICE	SIZE
SPY	20140102	9:30:00	183.98	500
SPY	20140102	9:30:00	183.98	500
SPY	20140102	9:30:00	183.98	200
SPY	20140102	9:30:00	183.98	500
SPY	20140102	9:30:00	183.98	1000
SPY	20140102	9:0:00	183.98	1000

SPY	20140102	9:30:00	183.98	800
SPY	20140102	9:30:00	183.98	100
SPY	20140102	9:30:00	183.98	100
SPY	20140102	9:30:00	183.97	200
SPY	20140102	9:30:00	183.98	100
SPY	20140102	9:30:00	183.97	200
SPY	20140102	9:30:00	183.98	1000
SPY	20140102	9:30:00	183.97	100
SPY	20140102	9:30:00	183.98	1000
SPY	20140102	9:30:00	183.98	2600
SPY	20140102	9:30:00	183.98	1000
SPY	20140102	9:30:00	183.97	400

Example: Below, we show the first transaction of the **AAPL TAQ** (*Trade and Quote*) data (tick-by-tick *quote* data) on January 2, 2014: 4 AM

SYMBOL	DATE	TIME	BID	OFR	BIDSIZ	OFRSIZ	MODE	EX
AAPL	20140102	4:00:00	455.39	0	1	0	12	T
AAPL	20140102	4:00:00	553.5	558	2	2	12	P
AAPL	20140102	4:00:01	455.39	561.02	1	2	12	T
AAPL	20140102	4:00:45	552.1	558	1	2	12	P
AAPL	20140102	4:00:51	552.1	558.4	1	2	12	P
AAPL	20140102	4:00:51	552.1	558.8	1	2	12	P
AAPL	20140102	4:00:51	552.1	559	1	1	12	P
AAPL	20140102	4:01:14	553	559	1	1	12	P
AAPL	20140102	4:01:30	553.01	561.02	1	2	12	T
AAPL	20140102	4:01:43	553.01	559	1	1	12	T
AAPL	20140102	4:01:44	553.05	559	1	1	12	P
AAPL	20140102	4:01:49	455.39	559	1	1	12	T
AAPL	20140102	4:01:49	553.61	559	1	1	12	T
AAPL	20140102	4:02:02	553.05	559	1	2	12	P
AAPL	20140102	4:02:04	455.39	559	1	1	12	T
AAPL	20140102	4:02:04	548.28	559	1	1	12	T
AAPL	20140102	4:02:33	553.05	558.83	1	2	12	P

AAPL	20140102	4:02:33	555.17	558.83	2	2	12	P
AAPL	20140102	4:03:50	555.2	558.83	5	2	12	P

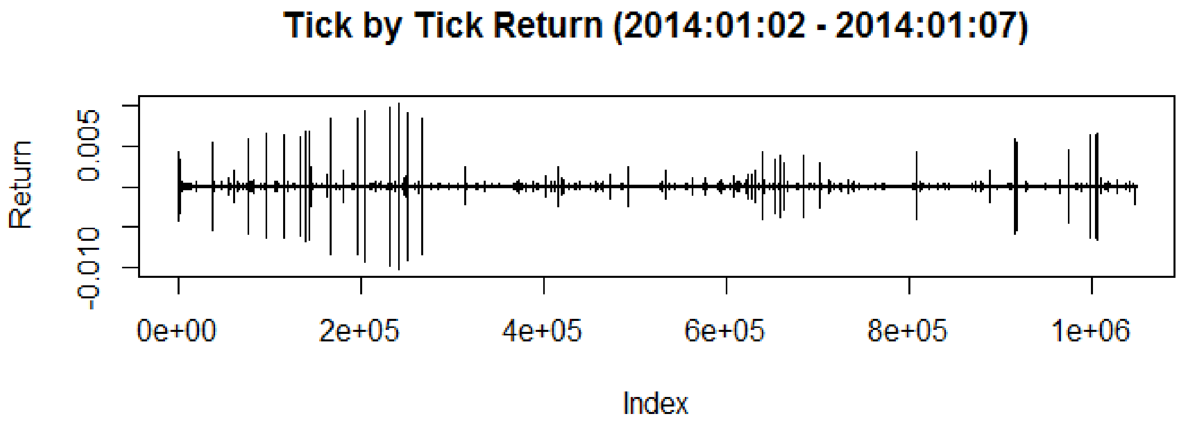
RV Models: High Frequency – Working with Tick Data

Example: We read **SPY trade data** for 2014:Jan.

```
> HF_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/SPY_2014.csv", head=TRUE,
sep=",")
> summary(HF_da)
SYMBOL          DATE              TIME             PRICE             SIZE          G127
SPY:6800865   Min.   :20140102             9:30:00 :  21436        Min.   :176.6     Min.   :      1
                  Min.   :0
                  1st Qu.:20140110  16:00:00:          11352          1st Qu.:178.9     1st Qu.:     100
                  1st Qu.:0
                  Median :20140121             9:30:01 :                5922              Median :182.6
                  Median :      100      Median :0
                  Mean   :20140119 15:59:59:          4090            Mean   :181.4     Mean   :    337
                  Mean   :0
                  3rd Qu.:20140128 15:59:55:          3198            3rd Qu.:183.5     3rd Qu.:    300
3rd Qu.:0
                  Max.   :20140131   15:50:00:          2916            Max.   :189.2     Max.
:4715350       Max.   :0
                                                   (Other) :6751951
     CORR                      COND              EX
 Min.   :0.0e+00         @    :3351783        T      :1649158
 1st Qu.:0.0e+00         F      :2888182        P      :1335135
 Median :0.0e+00                : 524409         Z      :1182126
 Mean   :1.9e-04         O    :   18057        D      :1062382
 3rd Qu.:0.0e+00         4     :    9098         K      : 437900
 Max.   :1.2e+01         6     :    8142         J     : 356539
                               (Other):    1194      (Other): 777625
```

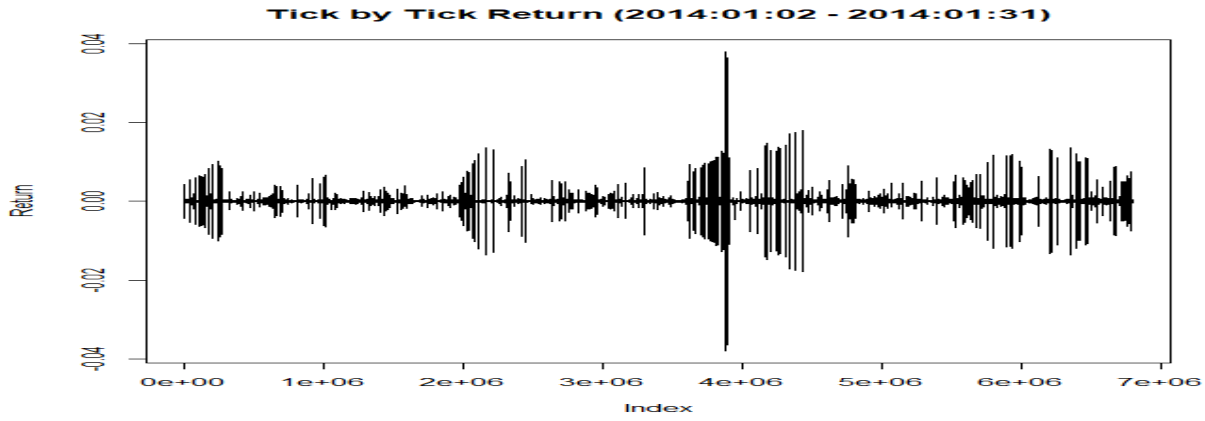
• Now, we calculate using 5'-returns a daily realized volatilitiy for the first 4 days in 2014 (2014:01:02 - 2014:01:07). Originally, we have *T* = 1,048,570.

```
pt <- as.POSIXct(paste(HF_da$DATE, HF_da$TIME), format="%Y%m%d %H:%M:%S")
library(xts)
hf_1 <- xts(x=HF_da, order.by = pt)             # Define a specific time series data set
                                                # pt pastes together DATE and Time.
spy_p <- as.numeric(hf_1$PRICE)                 # Read price data as numeric

T <- length(spy_5_p)
spy_ret <- log(spy_p[-1]/spy_p[-T])
plot(spy_ret, type="l", ylab="Return", main="Tick by Tick Return (2014:01:02 - 2014:01:07)")
mean(spy_ret)
sd(spy_ret)
```



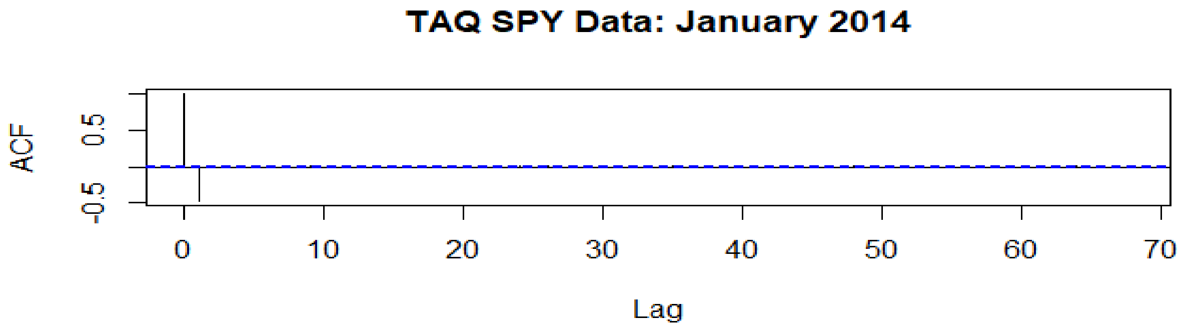
Very noisy data, with lots of "jumps":
Mean tick by tick return: -3.7365e-09
Tick-by-tick SD: 6.3163e-05

- For the whole month of January 2020:



```
> mean(spy_ret)
[1] -4.796933e-09
> sd(spy_ret)
[1] 7.804991e-05
```

- We plot the autocorrelogram for the TAQ SPY data:

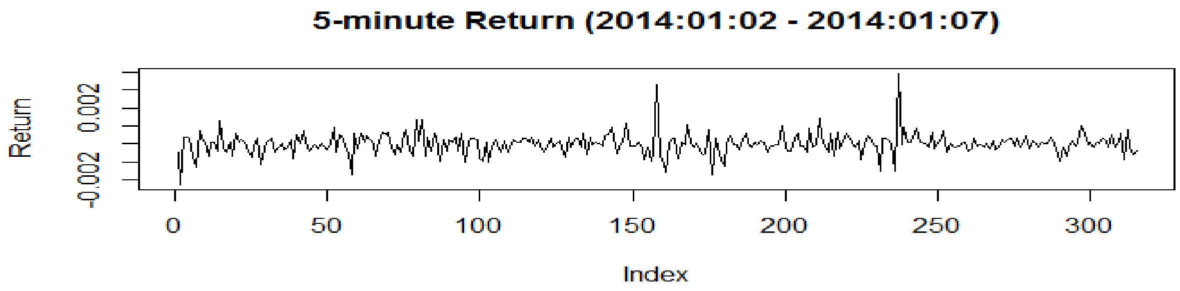


```
Autocorrelations of series ‘spy_ret’, by lag
        0        1        2        3        4        5        6        7        8        9       10
 1.000 -0.469 -0.013 -0.010  0.014 -0.008  0.000 -0.002 -0.001  0.000  0.000
```

Note: We have only a significant autocorrelation, the 1^{st}-order autocorrelation: **-0.459.**

• We aggregate the tick-by-tick data in **5’ intervals** using the function *aggregateTrades* in the R package *highfrequency*. It needs as an input an xts object (hf_1, for us).

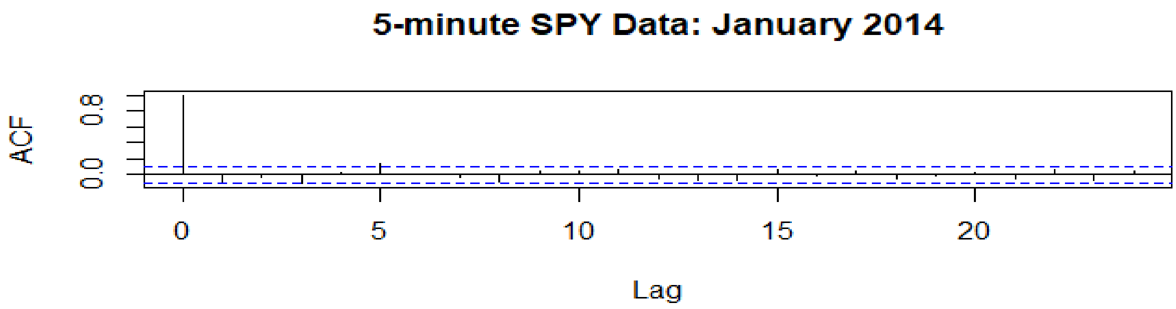
```
library(highfrequency)
spy_5 <- aggregateTrades(
hf_1,
on = "minutes",                          # you can use also seconds, days, weeks, etc.
k = 5,                          # number of units in for “on”
marketOpen = "09:30:00",
marketClose = "16:00:00",
tz = "GMT"
)
spy_5_p <- as.numeric(spy_5$PRICE)
T <- length(spy_5_p)
spy_5_ret <- log(spy_5_p[-1]/spy_5_p[-T])
plot(spy_5_ret, type="l", ylab="Return", main="5-minute Return (2014:01:02 - 2014:01:07)")
```



RVol$_{t=2014:01:02}$ = 0.0053344
RVol$_{t=2014:01:03}$ = 0.0043888
RVol$_{t=2014:01:04}$ = 0.0059836

RVol$_{t=2014:01:05}$ = 0.0052772

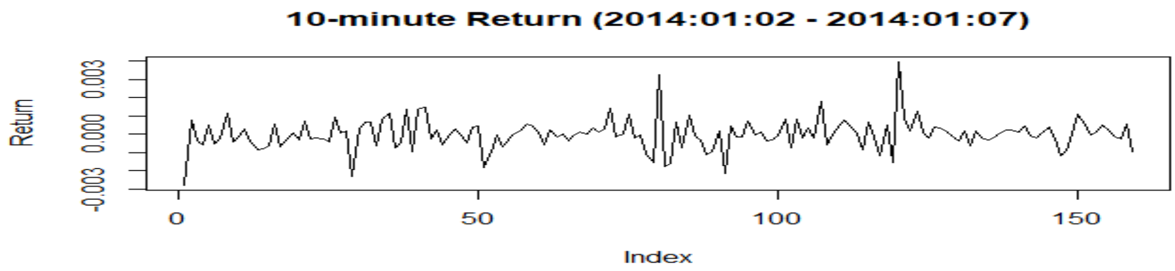
We plot the autocorrelogram for the **5-minute TAQ SPY return** data:



```
> acf_spy_5 <- acf(spy_5_ret, main = "5-minute SPY Data: January 2014")
> acf_spy_5
Autocorrelations of series ‘spy_ret’, by lag
0        1        2      3      4      5      6       7       8      9      10
1.000 -0.105 -0.024 -0.104  0.018  0.147  0.016 -0.024 -0.088  0.048  0.037
```

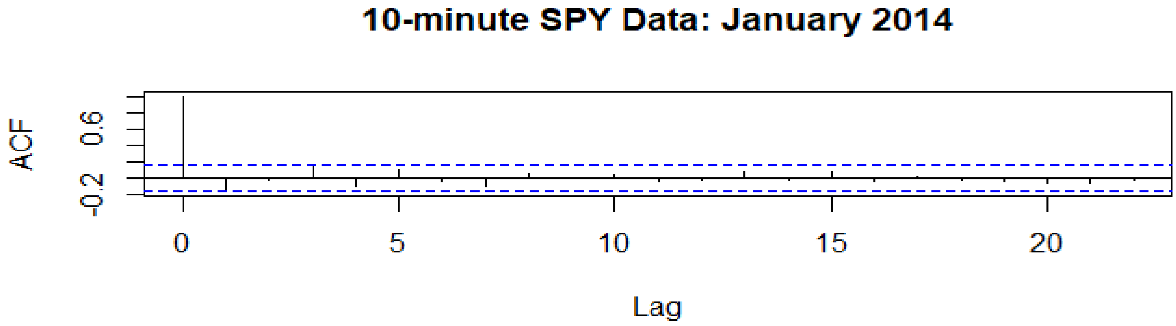
Note: We have a negative 1st-order autocorrelation: **-0.105**, thought not significant. However, the autocorrelation of order 5 is significant.

- We plot the **10-minute TAQ SPY return** data. Smoothing increases.



RVol$_{t=2014:01:02}$ = 0.005478294
RVol$_{t=2014:01:03}$ = 0.004256046
RVol$_{t=2014:01:04}$ = 0.006190508
RVol$_{t=2014:01:05}$ = 0.005145601

We plot the autocorrelogram for the 10’ TAQ SPY return data:



Note: Now, none of the autocorrelations is significant. The **10-minute returns** look independent.
¶

RV Models: High Frequency – TAQ In Practice

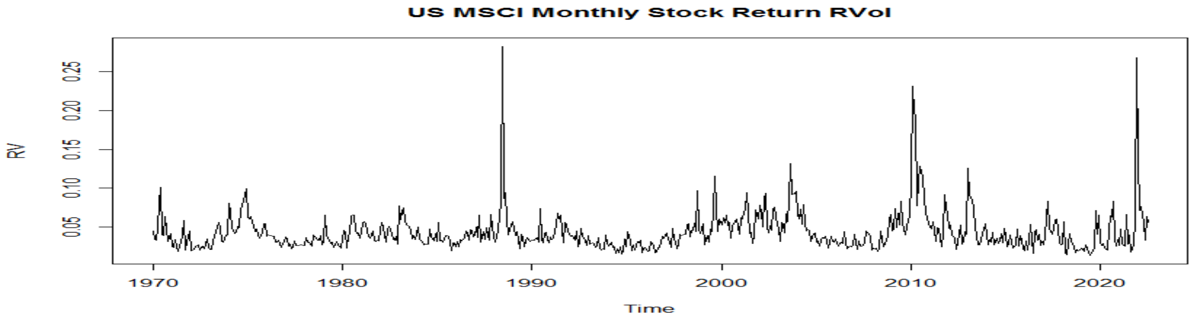
In practice, 10' returns are common. To form a daily measure for RV, we have 39 10-minute returns plus one overnite return (from 16:00 PM to next day 9:30 AM)

We have some technical issues working with tick data:
- Not all days the stock market is open from 9:30 AM to 16:00 PM, NYSE closes early on certain days (Christmas Eve, Thanksgiving).
- For many stocks, we do have lapses in trading. For these stocks, using 5' or 10' intervals may not work well.
- There are many suggested solutions to the problem of infrequent trading. Usual solution: interpolation from quote data.
- We have a lot of (discrete) jumps in the data.

Example: R script to compute *monthly* realized volatility for **MSCI USA daily returns**

```
MSCI_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/MSCI_daily.csv", head=TRUE,
sep=",")
x_us <- MSCI_da$USAT <- length(x_us)
us_r <- log(x_us[-1]/x_us[-T])

x <- us_r                              # US log returns from MSCI USA Index
T <- length(x)
rvs=NULL                               # create vector to fill with RV
i <- 1
k <- 21                                # k: observations per period (78 for 5' data)
while (i < T - k) {
s2 <- sum(x[i:(i+k)]^2)                        # realized variance
i <- k + i
rvs <- rbind(rvs,s2)
}
rvol <- sqrt(rvs)                      # realized volatility
mean(rvol)                             # mean
sd(rvol)                                       # variance
```

Example: Using **daily MSCI USA** data we calculate 1-mo Realized Volatility (*k*=21 days) for log returns for the USA MSCI (1970: Jan – 2020: Oct).



```
> mean(rvol)                                  # average monthly Rvol in the sample
[1] 0.04326531                 ⇒ very close to monthly S&P Volatility: 4.49%
> sd(rvol)                                    # standard deviation of monthly Rvol in the sample
[1] 0.02592653                  ⇒ dividing by sqrt(T) we get the SE = 0.001 (very small). ¶
```

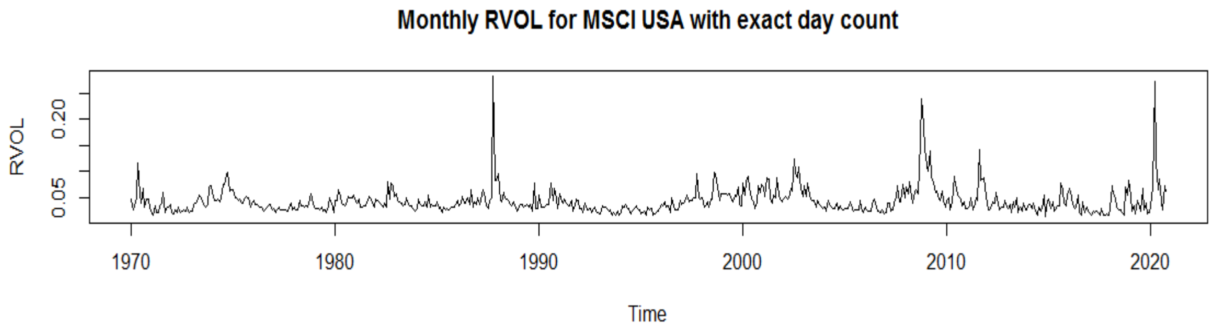
<u>Technical computing points</u>:
We use *k*=21 days, which is an average of the trading days per month. Of course, not all months have the same amount of trading days. In 2019, February had the fewest (19) and October the most (23), but, in 2018, February and September (18) had the fewest and August the most (23). For us, *k*=21 days is an approximation.

To be precise, if we use daily data to calculate a monthly variance, we need to use an exact index of trading days, say, K=[$k_1, k_2, k_3, ... k_J$] where k_i is the exact number of trading days in *month-year i.*

In addition, for daily data, we should not ignore the mean in the computation of RV.

Example: Below, the while loop in R is modified to incorporate the vector K (c1) of exact trading days for each month.

```
MSCI_cd <-
read.table("https://www.bauer.uh.edu/rsusmel/4397/MSCI_d_count_days.txt",header=FALSE)
c1 <- MSCI_cd[,1]                    # Vector with all exact days in a month
n_c1 <- length(c1)                   # Total number of days in sample
rvs=NULL                             # Initialize empty vector to place RVs
t <- 1                               # index for the days for while loop
tj <- 1                              # index for the months for while loop
x_m = mean(x)
while (tj  <= n_c1) {
mj <- c1[tj]                         # reading exact number of days for month tj
xx <- x[t:(t+mj-1)] - x_m                    # daily returns (in deviation from mean) per month
tj
s2 <- sum(xx^2)        # RV for month tj
t <- t + mj
tj <- tj + 1
rvs <- rbind(rvs,s2)                    # add RV for month tj to vector rvs
}

rvol <- sqrt(rvs)                               # realized volatility
> mean(rvol)                            # mean
[1] 0.04285471
> sd(rvol)                              # variance
[1] 0.02622621
> rvs_ts <- ts(rvol,start=c(1970,1),frequency=12)
> plot.ts(rvs_ts,xlab="Time",ylab="RVOL", main="Monthly RVOL for MSCI USA")
```



Note: The results (mean, SD and shape of RV) are very similar, but if used to compare to other monthly volatility estimates, these are the correct monthly RVol estimates. ¶

RV Models: Log Approximation Rules

The log approximations rules for the variance and SD are used to change frequencies for the RV and RVol. For example, suppose we are calculating RV based on frequency j, $RV_{t=j}$; but we are interested in the J-period $RV_{t=J}$. Then, the J-period (with j intervals) realized variance and realized volatility can be calculated as

$$RV_{t=J} = J \, * \, RV_{t=j}$$
$$RVol_{t=J} = sqrt(J) \, * \, RVol_{t=j}$$

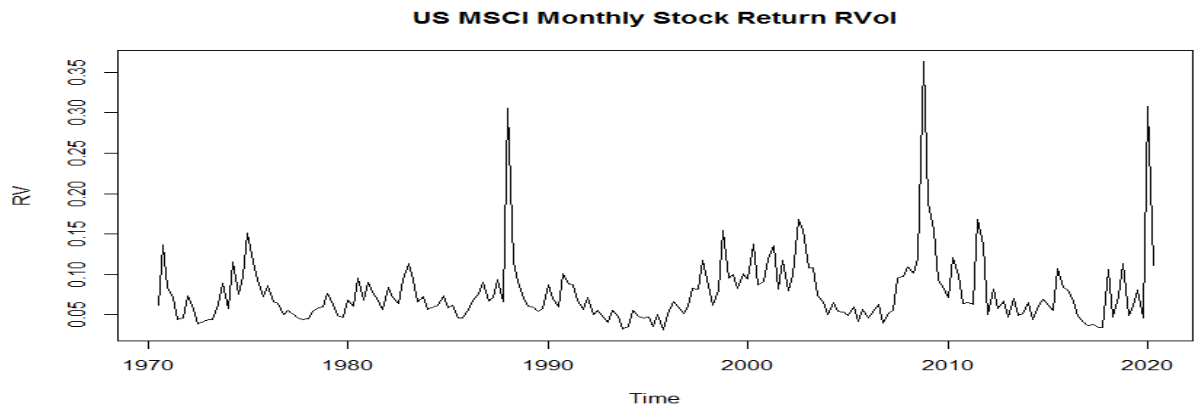
Example: We calculate using 5' data the daily realized variance, $RV_{t=daily}$. Then, the annual variance can be calculated as

$$RV_{t=annual} = 260 \, * \, RV_{t=daily}$$

where 260 is the number of trading days in the year. The annualized RVOL is the squared root of RV_{annual}:

$$RVOL_{t=annual} = sqrt(260) \, * \, RVOL_{t=daily}$$

Example: Using daily data we calculate 3-mo Realized Volatility (*k*=66 days) for log returns for the MSCI (1970: March – 2020: Oct).



```
> mean(rvol)                              # average quarterly Rvol in the sample
[1] 0.07725361               ⇒ log approximation: sqrt(3) * 0.04327 = 0.07495 (close!)
> sd(rvol)                                # standard deviation of quarterly Rvol in the
sample
[1] 0.02592653. ¶
```

RV Models: Properties

Under some conditions (bounded kurtosis and autocorrelation of squared returns less than 1), RV_t is consistent.

Realized volatility is a measure. It has a distribution.

For returns, the distribution of RV is non-normal (as expected). It tends to be skewed right and leptokurtic.

Daily returns standardized by RVol measures are nearly Gaussian.

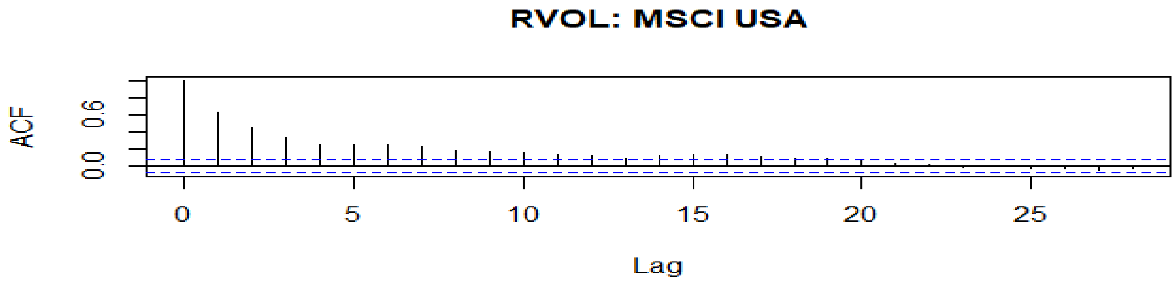
RV is highly persistent. (Check with a LB test.)

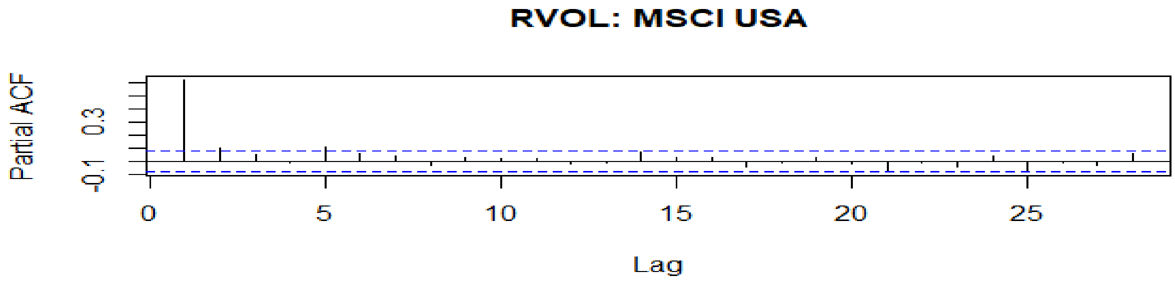
Daily RV calculate with intra-daily data, it is found to be more robust than measures using daily data, like GARCH.

RV Models: ACF and Persistence

Like all volatility measures, RVOL is highly autocorrelated.

Example: We plot the ACF and PACF for the 1-mo Realized Volatility, based on daily data for the monthly USA MSCI data.





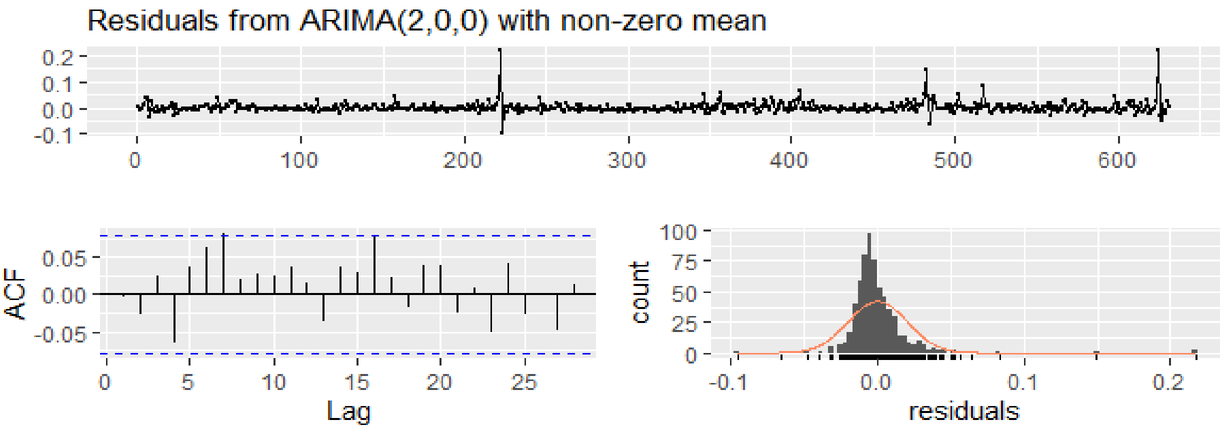
⇒ Model: AR(2)?

RV Models: Forecasting

We can fit ARMA models to the RVOL series to generate forecasts.

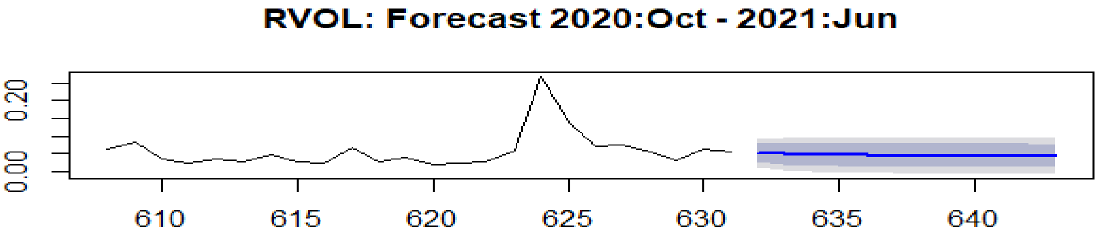
Example: Based on the ACF and PACF, we fit an AR(2) model for the monthly RVOL, calculated from monthly data:

```
> fit_rvol_ar2 <- arima(rvol, order=c(2,0,0))
> fit_rvol_ar2
Call:
arima(x = rvol, order = c(2, 0, 0))

      ar1     ar2        intercept
      0.5631  0.0967     0.0433
s.e.  0.0396  0.0396     0.0023

sigma^2 estimated as 0.0004056:  log likelihood = 1568.46,  aic = -3128.92

> checkresiduals(fit_rvol_ar2)

        Ljung-Box test

data:  Residuals from ARIMA(2,0,0) with non-zero mean
Q* = 12.008, df = 7, p-value = 0.1003

Model df: 3.   Total lags used: 10
```



- AR(2) model seems to pass diagnostic tests. Now, we forecast RVOL.

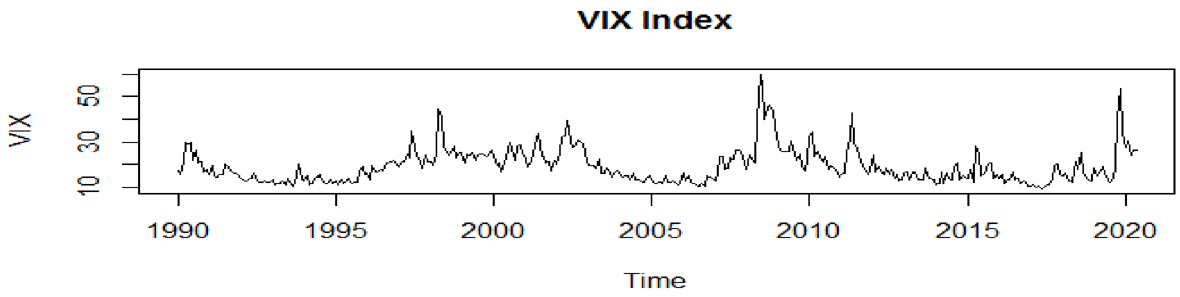
```
fcast_rvol <- forecast(fit_rvol_ar2, h=12, level=.95)    # h=number of step-ahead forecasts
> fcast_rvol
      Point Forecast     Lo 95          Hi 95
632     0.05201688  0.0125419811 0.09149178
633     0.04937852  0.0040761548 0.09468088
634     0.04757422 -0.0005822456 0.09573069
635     0.04630317 -0.0031716903 0.09577804
636     0.04541302 -0.0046992667 0.09552532
637     0.04478891 -0.0056334466 0.09521126
638     0.04435142 -0.0062226287 0.09492546
639     0.04404473 -0.0066036868 0.09469315
640     0.04382975 -0.0068551809 0.09451467
641     0.04367904 -0.0070238175 0.09438190
642     0.04357339 -0.0071382718 0.09428506
643     0.04349934 -0.0072166577 0.09421533. ¶
```



Note: The VIX index ("*fear index*") is a forecast for the next 30-day volatility, derived from S&P 500 options. The VIX on Sep 30, 2020 was 26.37, that is, the volatility at the end of October is expected to be 26.37% annualized or 7.61% monthly, higher than 5.20%, but, well within the 95% C.I. (More on this later.)

RV Models: Forecasting – Using VIX

Empirical work uses the VIX to calculate the implied volatility, IV_t, for the S&P500. The VIX index is based on the S&P500 index options (on a panel of S&P 500 option prices), using the "model-free" approach tailored to replicate the (annualized) risk-neutral volatility of a fixed 30-day maturity.



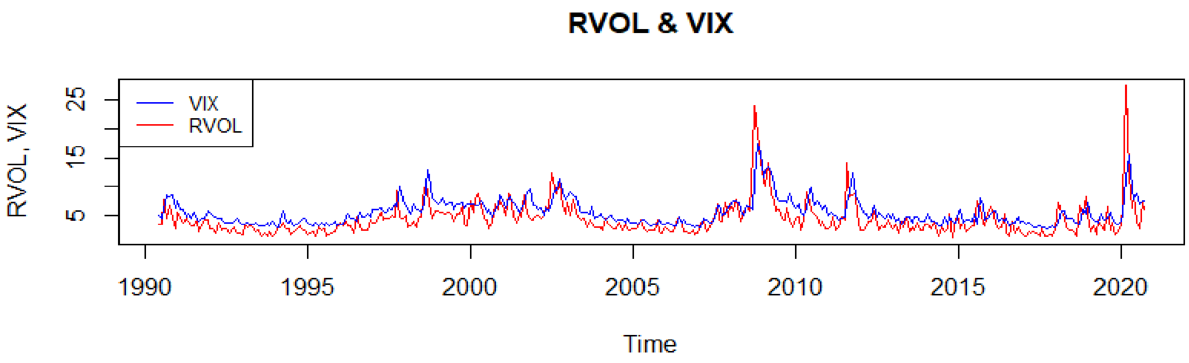
Example: We use **VIX** to forecast monthly RV based on daily data (1990:May - 2020:Sep). We regress

$$RV_{t+1} = \alpha + \beta\ VIX_t + \varepsilon_t.$$

```
Mid_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/Mid1_U_B_data.csv", head=TRUE, sep=",")
v_date <- Mid_da$Code
VIX <- Mid_da$VIX                        # Extract VIX data
T_rv <- length(rvol)                              # End of sample for RVol (2020:Oct)
rvol_90 <- as.numeric(rvol_ts[245:T_rv])*100        # RVol starting in 1990:May in %
rvol_0 <- rvol_90[-1]                    # remove first observation (RVt+1)
VIX_m <- VIX/sqrt(12)                             # VIX in monthly %
lm_rvol_f <- lm(rvol_0 ~ VIX_m)
> summary(lm_rvol_f)
```

```
Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  -0.89301          0.28021          -3.187  0.00156 **
VIX_m         0.94997   0.04641   20.469  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.967 on 363 degrees of freedom
Multiple R-squared:  0.5358,    Adjusted R-squared:  0.5345
F-statistic:   419 on 1 and 363 DF, p-value: < 2.2e-16
```

Note: In sample, a strong positive predictive relation.



Note: There is good match between the two series. RVOL shocks (Financial crisis, Covid) are unexpected by IV.

- We also check the contemporaneous relation between RVOL and VIX.

```
lm_rvol <- lm(rvol_90[-length(rvol_90)] ~ VIX_m)
> summary(lm_rvol)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.88538    0.20458  -9.216   <2e-16 ***
VIX_m         1.12543    0.03388  33.214   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.436 on 363 degrees of freedom
Multiple R-squared:  0.7524,    Adjusted R-squared:  0.7517
F-statistic:  1103 on 1 and 363 DF,  p-value: < 2.2e-16
```

Note: A strong contemporaneous relation. RVOL is highly correlated. ¶

RV Models: Variance Risk Premium (VRP)

The implied volatility of an option, calculated today, or IV_t, is a measure of the ("ex ante") expected variance over the remaining life of the option.

The Black-Scholes (BS) and similar models for option prices produce the same option prices as would be seen under modified probabilities in a world of investors who were indifferent to risk (*risk neutral*).

IV & other parameters extracted from options market prices embed these modified *"risk neutral"* probabilities, that combine investors' objective predictions of the real world returns distribution with their risk preferences.

Under BS assumptions, IV and market volatility are the same. But, BS assumptions do not hold. The VRP uses this disparity.

We define the variance risk premium (VRP) as the difference between the “ex-ante” *risk neutral expectation* at time t of the future return variation over the period $[t, t+1]$ time interval and the ex-post realized return variation over the $[t-1, t]$:

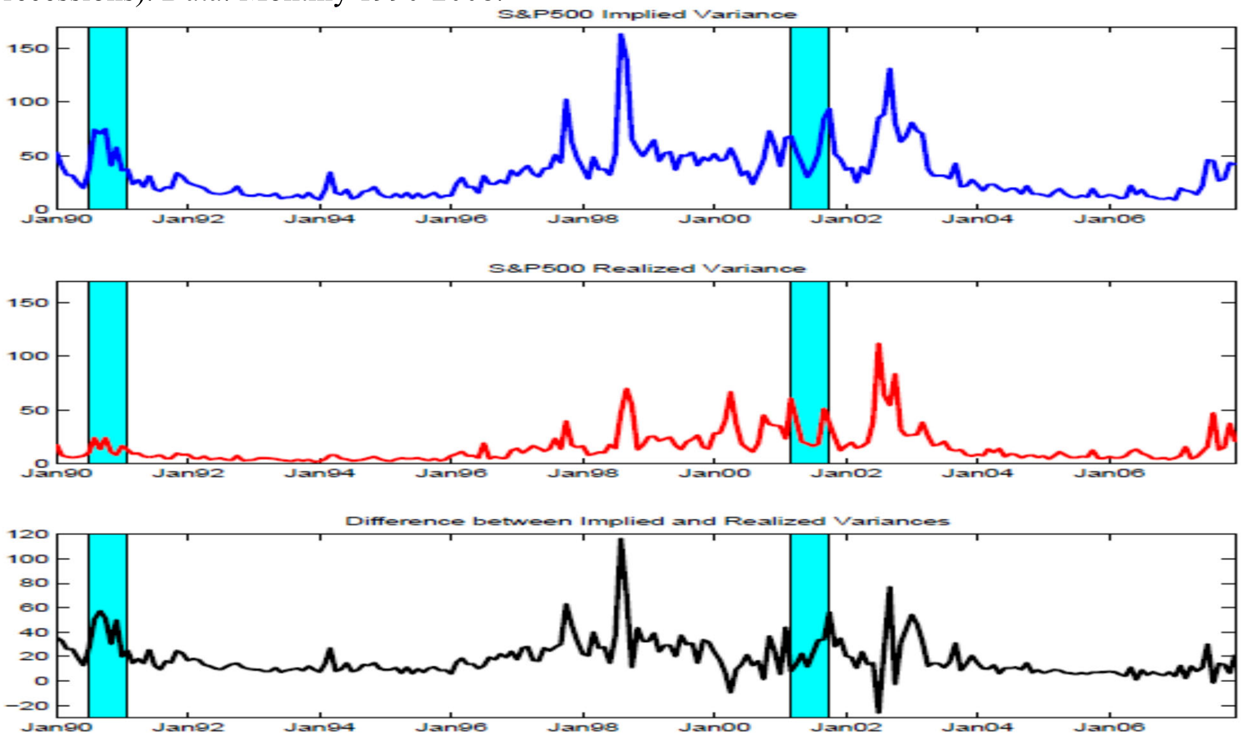
$$\text{VRP}_t = \text{IV}_t - \text{RV}_t.$$

It is an ad-hoc definition, we could have defined VRP_t based on the expectation at time t for RV_{t+1}, in this case $\text{E}_t[\text{RV}_{t+1}]$. The one-step-ahead forecast can be obtained using an ARMA process for RV_t.

In practice, using $\text{E}_t[\text{RV}_{t+1}]$ or RV_t, does not affect VRP_t that much.

The are many ways to calculate IV: based on models, like the BS, or “*model free*,” similar to how we calculated IV, in this case, using changes in option prices for different strike prices and computing an average.

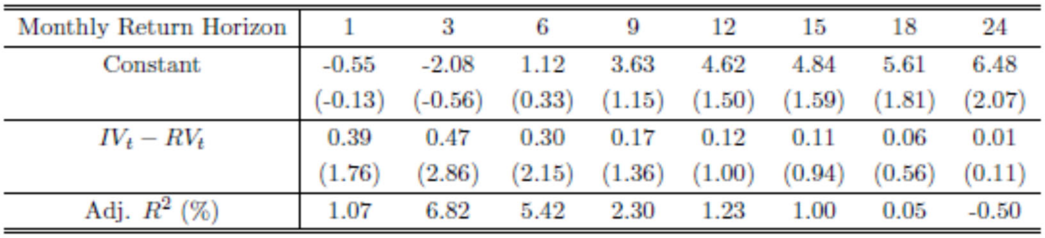
Example: We plot IV_t(=VIX), RV_t & VRP_t for the S&P500 Index (shaded blue area are U.S. recessions). Data: Monthly 1990-2008.



- Bollerslev et al. (2009) use 5’ intervals to calculate RV_t find that VRP_t is a predictor of stock market excess returns at different horizons ($t+h$). That is, they regress:

$$r_{t+h} - r_{f,t+h} = [\log(P_t) - \log(P_{t-1})] = \mu + \delta\ \text{VRP}_t + \varepsilon_t$$

They find that $\hat{\delta}$ is positive and has a *t-stat*=1.76 for monthly data (*h*=1) and a *t-stat* = 2.86 for quarterly data (*h*=3). The R^2 is 1.07% for monthly data and 6.82% for quarterly data. For annual data the t-stat is not significant.



Monthly Return Horizon	1	3	6	9	12	15	18	24
Constant	-0.55	-2.08	1.12	3.63	4.62	4.84	5.61	6.48
	(-0.13)	(-0.56)	(0.33)	(1.15)	(1.50)	(1.59)	(1.81)	(2.07)
$IV_t - RV_t$	0.39	0.47	0.30	0.17	0.12	0.11	0.06	0.01
	(1.76)	(2.86)	(2.15)	(1.36)	(1.00)	(0.94)	(0.56)	(0.11)
Adj. R^2 (%)	1.07	6.82	5.42	2.30	1.23	1.00	0.05	-0.50

Example: We regress excess next-month returns, using the FF Mkt-RF factor as the dependent variable, on today's VRP:

```
FF_da <- read.csv("http://www.bauer.uh.edu/rsusmel/4397/FF_5_factors.csv",header=TRUE)
x_Mkt_RF <- FF_da$Mkt_RF                              # FF excess market returns
T_FF <- length(x_RF)                                  # size of FF_da
Mkt_RF <- x_Mkt_RF[323:T_FF]/100                 # Obs 332: 1990: May
vrp <- VIX_m^2 - rvol_90[-length(rvol_90)]^2     # Variance risk premium
pred_vrp <-lm(Mkt_RF[-1] ~ vrp)                  # Predictive regression
> summary(pred_vrp)

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.655e-03  2.335e-03   2.850  0.00462 **
vrp         1.815e-05  6.210e-05   0.292  0.77029          ⇒ not significant
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04346 on 363 degrees of freedom
Multiple R-squared:  0.0002352, Adjusted R-squared:  -0.002519. ¶
```

Other Models: Parkinson's (1980) Estimator

The Parkinson's (1980) estimator:

$s^2_t = \{\Sigma_t [\ln(H_t) - \ln(L_t)]^2 / (4\ln(2)T)\}$,

where H_t is the highest price and L_t is the lowest price.

There is an RV counterpart, using HF data: Realized Range (RR):

$RR_t = \{\Sigma_j [100 * (\ln(H_{t,j}) - \ln(L_{t,j}))]^2 / (4\ln(2)\}$,

where $H_{t,j}$ and $L_{t,j}$ are the highest and lowest price in the jth interval.

These "range" estimators are very good and very efficient.

These estimators can be applied to intra-daily data. The Realized Range works well with combined with other models.

Stochastic volatility (SV/SVOL) models

Now, instead of a known volatility at time *t*, like ARCH models, we allow for a stochastic shock to σ_t, η_t or υ_t:

$$\sigma_t = \omega + \beta_1 \sigma_{t-1} + \eta_t, \qquad \upsilon_t \sim N(0, \sigma_\eta^2)$$

Or using logs:

$$\log \sigma_t = \omega + \beta_1 \log \sigma_{t-1} + \upsilon_t, \qquad \upsilon_t \sim N(0, \sigma_v^2)$$

The difference with ARCH models: The shocks that govern the volatility are not necessarily the shocks to the mean process, ε_t's.

Usually, the standard model centers log volatility around ω:

$$\log \sigma_t = \omega + \beta_1 (\log \sigma_{t-1} - \omega) + \upsilon_t,$$

Then,

$E[\log(\sigma_t)] = \omega$

$Var[\log(\sigma_t)] = \kappa^2 = \sigma_\upsilon^2/(1 - \beta^2)$.

$\Rightarrow$ Unconditional distribution: $\log(\sigma_t) \sim N(\omega, \kappa^2)$

Like ARCH models, SV models produce returns with kurtosis > 3 (and, also, positive autocorrelations between squared excess returns).

We have 3 SVOL parameters to estimate: $\varphi = (\omega, \beta, \sigma_\upsilon)$.

Estimation: The modern approach uses Bayesian methods (MCMC), which are advanced for this class. Brooks discusses the estimation of SVOL.