Lecture 9-b
ARIMA – Estimation & Diagnostic Testing

Brooks (4th edition): Chapter 6

Review: ARMA Models – ACF & PACF

• We use correlations to select a proper model (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 $p > 0$ & $q > 0$.

	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.

Review: ARMA Model: Identification with 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 log-likelihood function plus a penalty factor, that depends on $k\ (= p + q)$. Many IC's

- Most popular:

- **Akaike Information Criterion (AIC)**
  AIC = -2 * (ln $L - k$) = -2 ln L + 2 * k
  $\Rightarrow$ if normality AIC = T * ln($\boldsymbol{e}'\boldsymbol{e}/T$) + 2* k (+constants)

- **Bayes-Schwarz Information Criterion (BIC or SBIC)**
  BIC = -2 * ln L – ln(T) * k
  $\Rightarrow$ if normality AIC = T * ln($\boldsymbol{e}'\boldsymbol{e}/T$) + ln($T$) * k (+constants)

Review: ARMA Model: Identification with IC

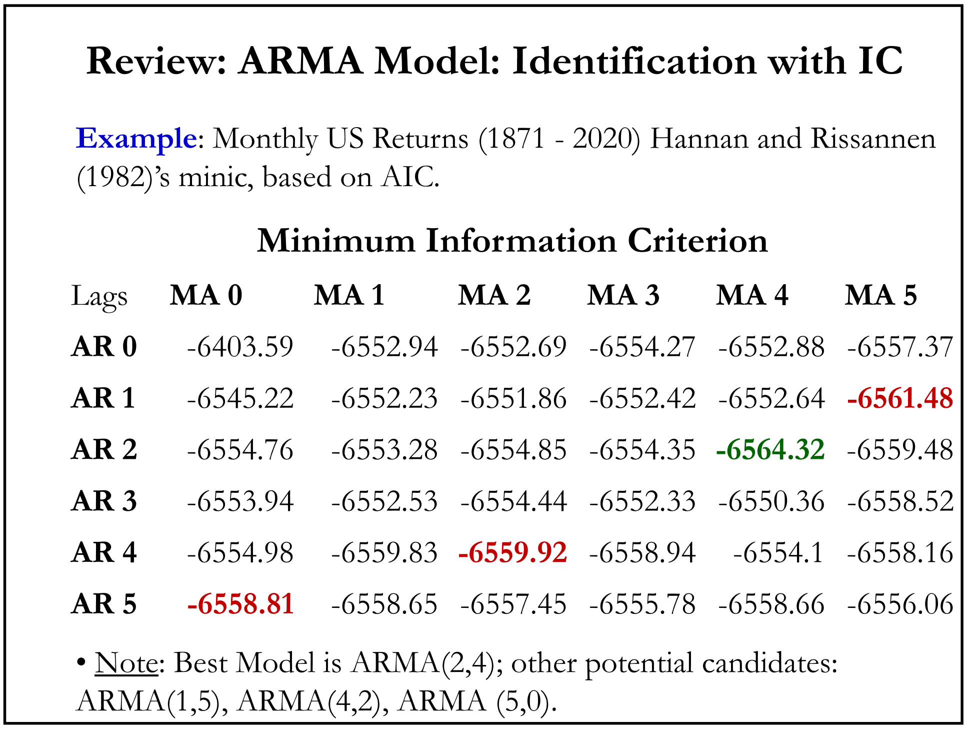
- There are many modifications of the above mentioned IC and there are IC that are specific to the popular AR(p) models.

Small sample correction, like $AICc$, are common:

$$AICc = T\ ln\hat{\sigma}^2 + \frac{2k(k+1)}{T-k-1}$$

- 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.

Minic can also be used with other IC, for example, AIC.



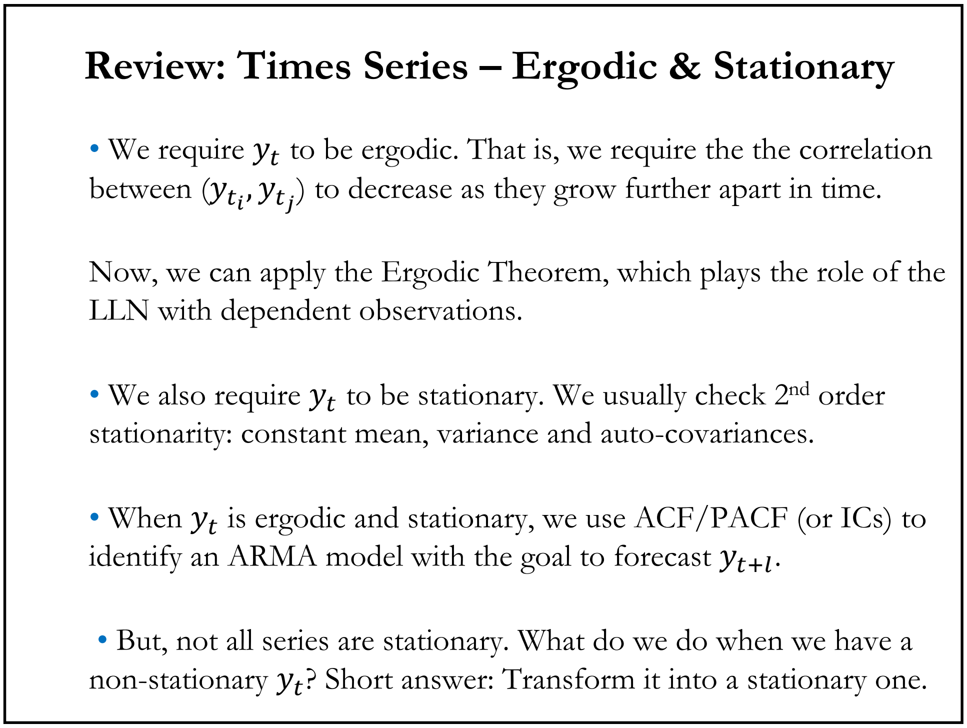
Review: ARMA Model: Identification with IC

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).



Review: Times Series – Ergodic & Stationary

- We require y_t to be ergodic. That is, we require the the correlation between (y_{t_i}, y_{t_j}) to decrease as they grow further apart in time.

Now, we can apply the Ergodic Theorem, which plays the role of the LLN with dependent observations.

- We also require y_t to be stationary. We usually check 2nd order stationarity: constant mean, variance and auto-covariances.

- When y_t is ergodic and stationary, we use ACF/PACF (or ICs) to identify an ARMA model with the goal to forecast y_{t+l}.

- But, not all series are stationary. What do we do when we have a non-stationary y_t? Short answer: Transform it into a stationary one.

Review: Non-Stationary Time Series Models

- Rough indicator of a trend: A slow decay in ACF, which suggests a **stochastic trend** (**unit root process**), or a **trend stationary process**.

- A series with a trend is not stationary. To build a forecasting model, we need to remove the trend from the series. The models we consider:

(1) Deterministic trend: y_t is a function of t. For example,

$$y_t = \alpha + \beta\, t + \varepsilon_t$$

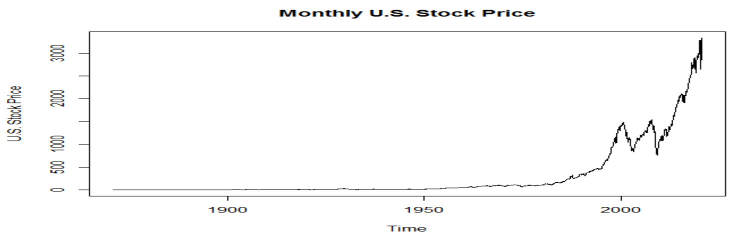
(2) Stochastic trend: y_t is a function of aggregated errors, ε_t, over time. For example,

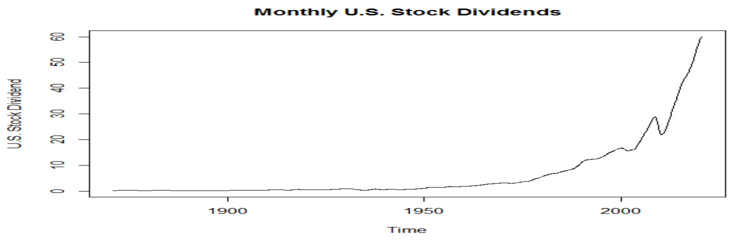
$$y_t = \mu + y_{t-1} + \varepsilon_t = y_0 + t\,\mu + \sum_{j=0}^{t} \varepsilon_{t-j}$$

- The process to remove the trend depends on the structure of the DGP of y_t.

Review: Non-Stationary Time Series Models

Example: Plot of US Monthly Prices and Dividends (1871 – 2020)





Review: Non-Stationarity – Deterministic Trend

- Suppose we have the following model, with a determinist trend:

$$y_t = \alpha + \beta\, t + \varepsilon_t.$$

- $\{y_t\}$ will show only temporary departures from trend line $\alpha + \beta\, t$. This type of model is called a **trend stationary** (**TS**) model.

- Note that trivially, by definition, ε_t is WN. Then, removing $\alpha + \beta\, t$ from y_t creates a WN series –i.e., the influence of t from y_t is gone:

$$\varepsilon_t = y_t - \alpha - \beta\, t$$

- When we replace α & β by their OLS estimates, we **detrend** y_t. The residual from the OLS is called **detrended** y_t.

$$e_t = y_t - \hat{\alpha} - \hat{\beta}\, t \qquad (e_t = \textbf{detrended } y_t \text{ series})$$

Review: Non-Stationarity – Deterministic Trend

- We can detrend in more complicated models. For example, suppose we have a stationary AR(p) model with linear and quadratic trends:

$$y_t = \alpha + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \beta_1 t + \beta_2 t^2 + \varepsilon_t.$$

- Note that removing from y_t a constant, a linear trend and a quadratic trend creates w_t:

$$w_t = \varepsilon_t + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} = y_t - \alpha - \beta_1 t - \beta_2 t^2$$

- w_t is a stationary series: No dependence on t**.** We will work with the residual from a regression of y_t agains a constant, t and t^2:

$$\hat{w}_t = y_t - \hat{\alpha} - \hat{\beta}_1\, t - \hat{\beta}_2 t^2 \qquad (\hat{w}_t = \textbf{detrended } y_t).$$

Remark: We do not necessarily get stationary series by detrending.

Review: Non-Stationarity – Deterministic Trend

- Many economic series exhibit "**exponential trend/growth**":

$$y_t = e^{\alpha + \beta t + \varepsilon_t}$$

- In these cases, we use logs to remove the trend:

$$\ln(y_t) = \alpha + \beta\, t + \varepsilon_t. \qquad (\Rightarrow y_t = e^{\alpha + \beta t + \varepsilon_t})$$

Implication. The average growth rate: $\mathrm{E}[\Delta\ln(y_t)] = \beta$

Comparison with a linear trend: $\mathrm{E}[\Delta y_t] = \beta$ (constant change.)

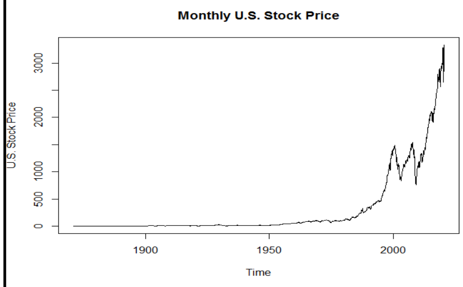
- We use the same process to remove trends in more general models:

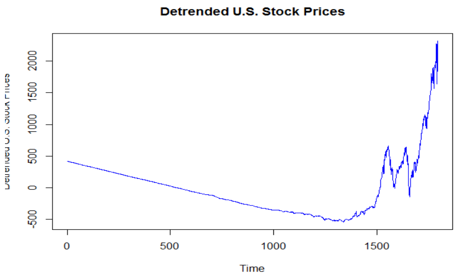
$$y_t = e^{\alpha + \beta_1 t + \beta_2 t^2 + \ldots + \beta_k t^k + \varepsilon_t}$$

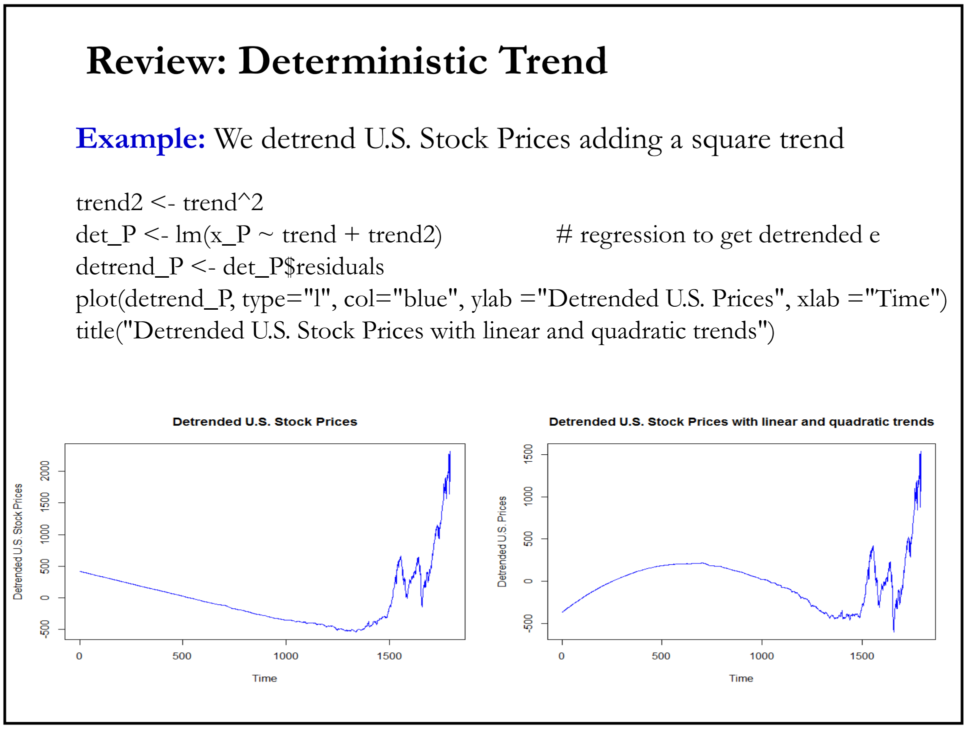
Review: Deterministic Trend

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")
```





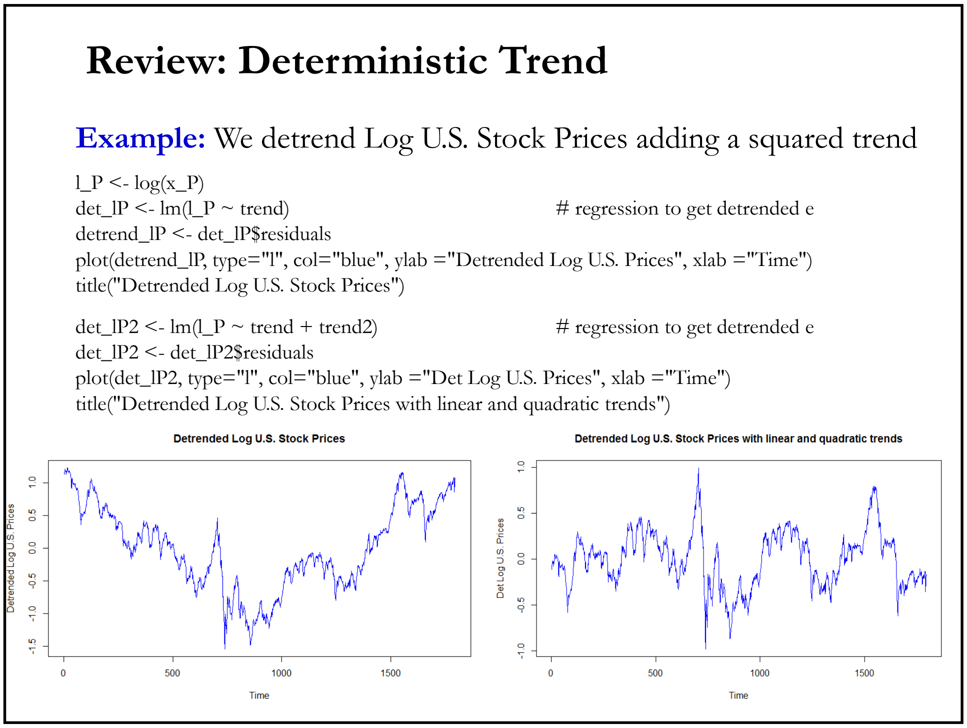


Review: Deterministic Trend

Example: 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")
```

Detrended U.S. Stock Prices
Detrended U.S. Stock Prices
-500
0
500
1000
1500
2000
0
500
1000
1500
Time
Detrended U.S. Stock Prices with linear and quadratic trends
Detrended U.S. Prices
-500
0
500
1000
1500
0
500
1000
1500
Time



Review: Deterministic Trend

Example: We detrend Log U.S. Stock Prices adding a squared 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_lP2$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")
```

Detrended Log U.S. Stock Prices
Detrended Log U.S. Prices
-1.5
-1.0
-0.5
0.0
0.5
1.0
0
500
1000
1500
Time
Detrended Log U.S. Stock Prices with linear and quadratic trends
Det Log U.S. Prices
-1.0
-0.5
0.0
0.5
1.0
0
500
1000
1500
Time

Review: Non-Stationarity – Deterministic Trend

- Estimation of AR(p) with a trend component: **Frish-Waugh method** (a 2-step method).

Steps:

(1) Detrend y_t: Regress y_t against a constant, $t, t^2, \ldots, t^k$.

⇒ get the residuals (=y_t without the influence of t).

$$\hat{w}_t = y_t - \hat{\alpha} - \hat{\beta}_1\, t - \hat{\beta}_2 t^2 - \ldots - \hat{\beta}_k t^k$$

(2) Estimate AR(p): Use residuals, $\hat{w}_t$, to estimate AR(p) model.

Review: Stochastic Trend

- Modern approach: The trend is "variable," it changes in an unpredictable way. Therefore, it is considered a **stochastic trend** (**ST**).

- The ST appears in the special case of AR(1) model, with $\phi_1 = 1$ (**unit root**, non-stationary, case):

$$y_t = \mu + y_{t-1} + \varepsilon_t$$

Q: Where is the (stochastic) trend? After backward substitution:

$$\begin{aligned} y_t &= \mu + y_{t-1} + \varepsilon_t \\ &= \mu + (\mu + y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t \\ &\ldots \\ &= y_0 + \underset{\substack{\downarrow \\ \boxed{\text{Deterministic trend}}}}{t\,\mu} + \textstyle\sum_{j=0}^{t} \varepsilon_{t-j} \end{aligned}$$

Review: Stochastic Trend

- A unit root generates a trend:

$$y_t = y_0 + \underset{\substack{\downarrow \\ \boxed{\text{Deterministic trend}}}}{t\,\mu} + \textstyle\sum_{j=0}^{t} \varepsilon_{t-j}$$

- This process is a "**random walk with drift**": y_t grows with t.

- 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.

Remark: A shock at time $t-j$, ε_{t-j}, affects y_t forever.

- We remove the trend by **differencing** y_t

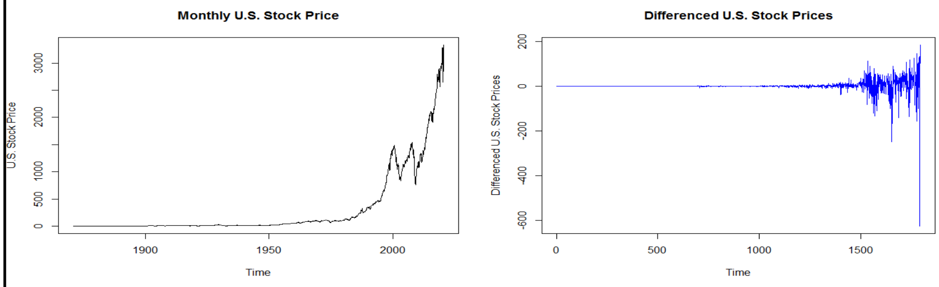
$$\Rightarrow \Delta y_t = (1-L)\, y_t = \mu + \varepsilon_t$$

Note: Applying the $(1-L)$ operator to a time series is called *differencing*

Review: Stochastic Trend

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")
```



Review: Stochastic Trend – ARIMA model

- When y_t has a stochastic trend, we use **Autoregressive Integrated Moving Average (ARIMA)** models.

- Q: 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: $y_{t+1} = \mu + \beta\,(t+1) + \varepsilon_{t+1}$ $\Rightarrow \varepsilon_t$ does not affect y_{t+1}.

– ST: $y_{t+1} = \mu + y_t + \varepsilon_{t+1} = \mu + [\mu + y_{t-1} + \varepsilon_t] + \varepsilon_{t+1}$
$= 2 * \mu + y_{t-1} + \varepsilon_t + \varepsilon_{t+1}$ $\Rightarrow \varepsilon_t$ affects y_{t+1}.

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,

$$\phi(L)\,(1-L)^d\, y_t = \theta(L)\,\varepsilon_t \qquad \text{is ARMA}(p, q).$$

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). Usual cases in finance:

$d = 1$. A time series with **unit root** is I(1), typical of asset prices.

$d = 0$. A stationary time series is I(0), typical of asset returns.

Examples:

Example 1: RW: $y_t = y_{t-1} + \varepsilon_t$.

y_t is non-stationary, but

$$w_t = (1-L)\; y_t = \varepsilon_t \qquad \Rightarrow w_t \sim \text{WN!}$$

Now, $y_t \sim$ ARIMA(0, 1, 0). ($d = 1$)

ARIMA(p, d, q) Models

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

$$w_t = (1 - L)\, y_t$$
$$= \mu + \delta\, t + \phi_1\, y_{t-1} + \varepsilon_t - [\mu + \delta\, (t - 1) + \phi_1\; y_{t-2} + \varepsilon_{t-1}].$$
$$= \delta + \phi_1\, w_{t-1} + \varepsilon_t - \varepsilon_{t-1} \qquad \Rightarrow w_t \sim \text{ARMA}(1, 1).$$

Now, $y_t \sim$ ARIMA(1, 1, 1).

- First differencing made both process stationary. However:

– Example 1: Differencing a series with a unit root in the AR part of the model reduces the AR order. **Differencing** is right in these cases.

– Example 2: Differencing introduced an extra MA structure (and non-invertibility ($\theta_1 = 1$)). This happens when we difference a TS series. **Detrending** should be used in these cases.

ARIMA(p, d, q) Models

- 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.

Note: There are many formal tests for unit roots. Most popular tests: ADF (Augmented Dickey-Fuller) and PP (Phillips-Perron).

ARIMA Models: Unit Roots 1?

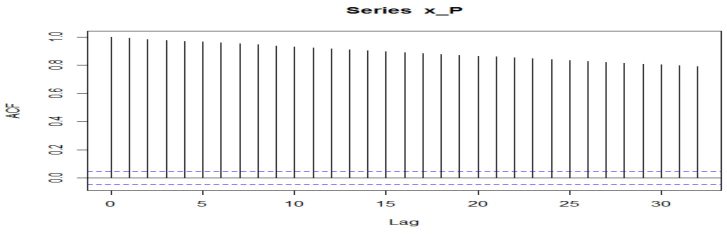
Example 1: Monthly Stock Price levels (1871-2020)

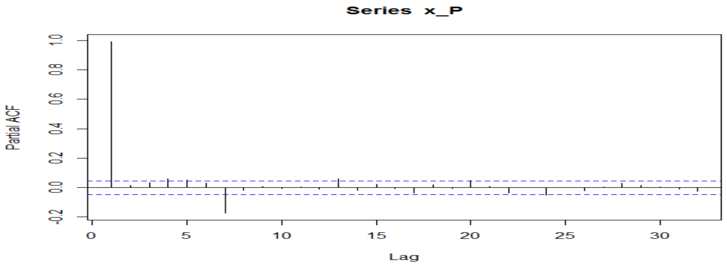
```
acf_P <- acf(x_P)
> acf_P
Autocorrelations of  series ‘x_p’, by lag

    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 $\phi_1 \approx 1$.

ARIMA Models – Unit Roots 1: ACF & PACF

Example 1: Monthly Stock Price levels (1871-2020)





ARIMA Models: Unit Roots 2?

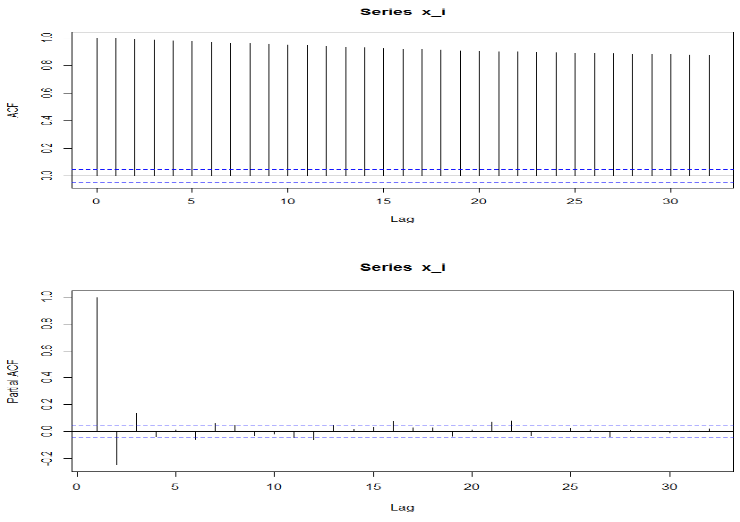
Example 2: Monthly Interest Rates (1871-2020)

```
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 – Unit Roots 2: ACF & PACF

Example 2: Monthly Interest Rates (1871-2020)



ARIMA Models – Random Walk

- **Random walk** (**RW**): A process where the current value of a variable is composed of the past value plus a WN error:

$$y_t = y_{t-1} + \varepsilon_t$$

- Implication: E[$y_{t+1} | I_t$] = y_t ⇒ Δy_t is absolutely random.

- Popular model. Used to explain the behavior of financial assets, unpredictable movements (Brownian motions, drunk persons).

Note: RW is a special case of an AR(1) process: a **unit-root** process.

- RW is an ARIMA(0,1,0) process:

$$\Delta y_t = (1 - L) y_t = \varepsilon_t, \qquad \varepsilon_t \sim WN(0, \sigma^2).$$

- A RW is nonstationary: ts variance increases with t.

ARIMA Models – Random Walk with Drift

- **Random walk with a drift**: We add a constant to the process:

$$y_t = \mu + \; y_{t-1} + \varepsilon_t$$

⇒ The drift creates a trend in y_t. Recall that y_t can also be written as:

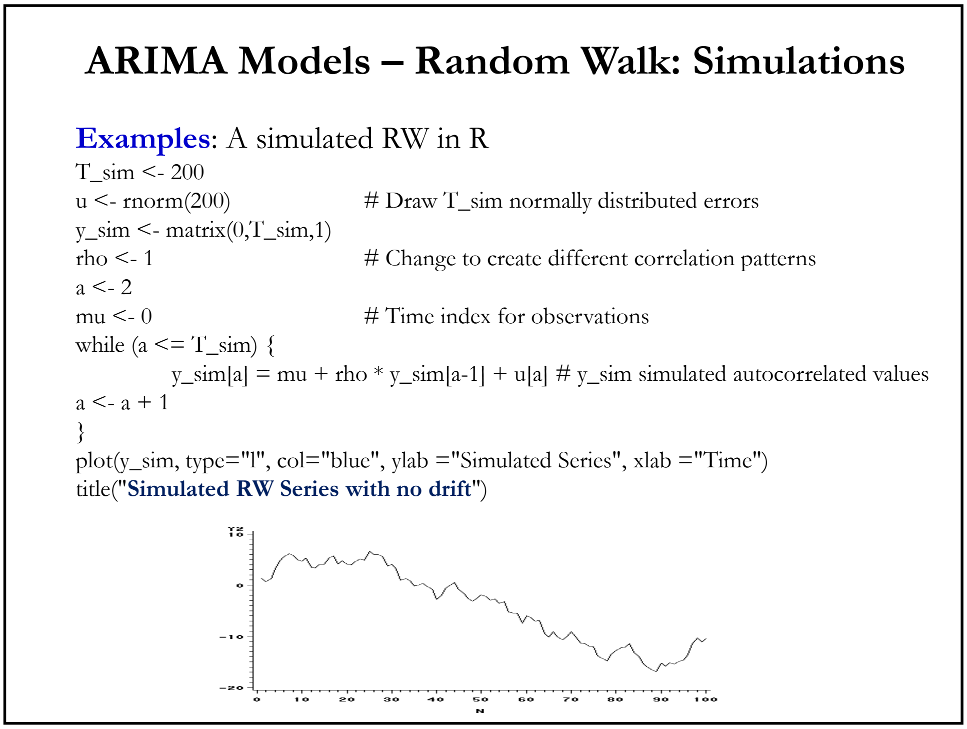
$$y_t = y_0 + t\,\mu \; + \sum_{j=0}^{t} \varepsilon_{t-j}$$

- Change in y_t is partially deterministic (μ) and partially stochastic.

$$\Delta y_t = \; y_t - y_{t-1} = \mu + \varepsilon_t$$

- Recall the difference between conditional and unconditional forecasts:

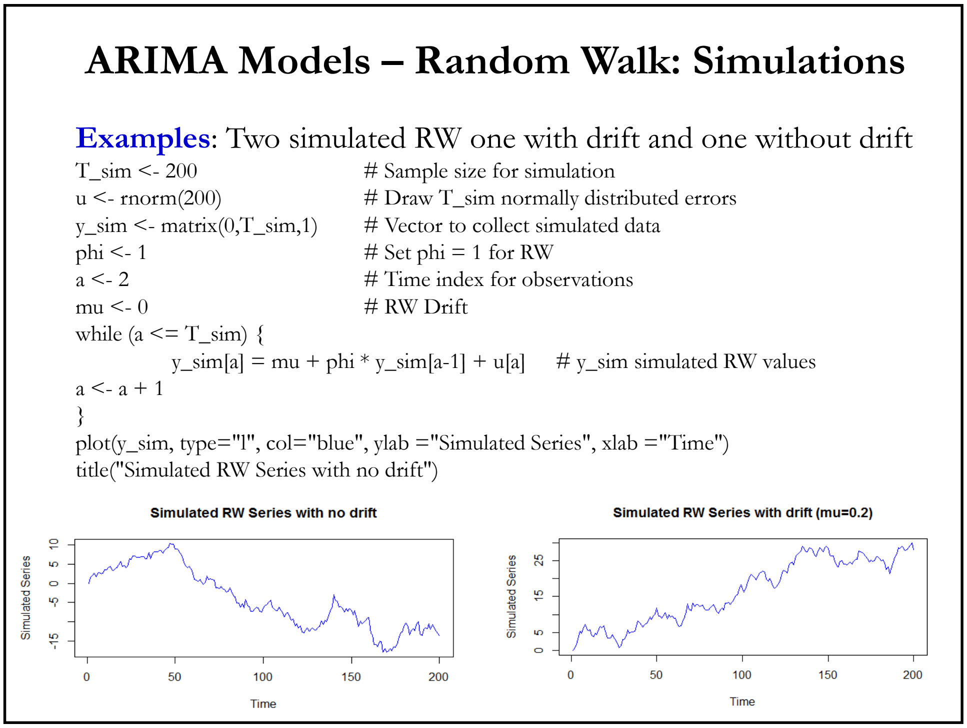
E[y_{t+s}] = $y_0 + (t + s)\,\mu$ (Unconditional forecast)
E[$y_{t+s} | y_t$] = $y_t + s\,\mu$ (Conditional forecast)



ARIMA Models – Random Walk: Simulations

Examples: A simulated RW in R

```
T_sim <- 200
u <- rnorm(200)                # Draw T_sim normally distributed errors
y_sim <- matrix(0,T_sim,1)
rho <- 1                       # Change to create different correlation patterns
a <- 2
mu <- 0                        # Time index for observations
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")
```



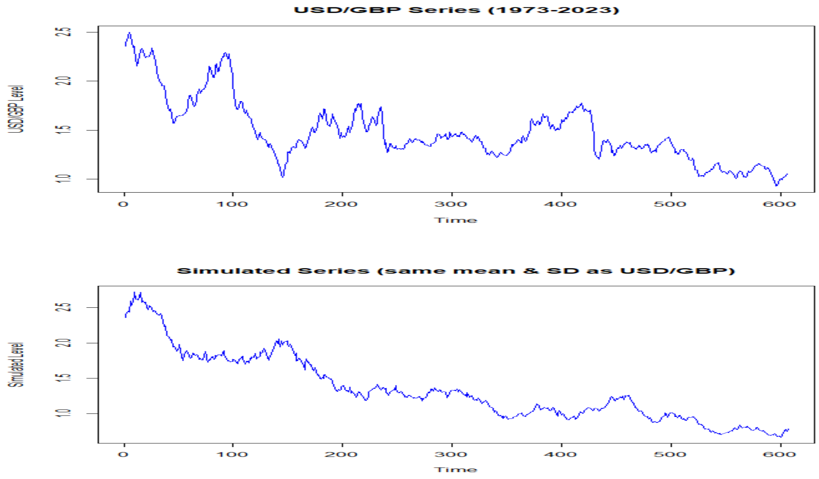
ARIMA Models – Random Walk: Simulations

Examples: Two simulated RW one with drift and one without drift

```
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                       # Set phi = 1 for RW
a <- 2                         # Time index for observations
mu <- 0                        # RW Drift
while (a <= T_sim) {
        y_sim[a] = mu + phi * y_sim[a-1] + u[a]     # y_sim simulated RW values
a <- a + 1
}
plot(y_sim, type="l", col="blue", ylab ="Simulated Series", xlab ="Time")
title("Simulated RW Series with no drift")
```

ARIMA Models – RW with Drift

- Two series: 1) True USD/GBP 1973-2023 series; 2) A simulated RW (same drift and variance). Very similar pattern!



ARIMA Models: Box-Jenkins

- We have a family of ARIMA models, indexed by p, q, and d.
Q: How do we select one?

An effective procedure for building empirical time series models is the Box-Jenkins approach, which consists of three stages:

(1) **Identification** or Model specification (order of ARIMA)

(2) **Estimation** of order p, q.

(3) **Diagnostics testing** on residuals:
$\Rightarrow$ Are they white noise? If not, add lags (p, q, or both).

If we are happy with model, then we proceed to **forecasting**.

ARIMA Models: Identification

• Recall the two main approaches to (1) Identification.

- **Correlation approach**: Based on ACF & PACF.

1) Make sure data is stationary –check a time plot. If not, differentiate.
2) Using ACF & PACF, guess small values for p & q.

- **Information criteria**: Very common situation: The order choice not clear from looking at ACF & PACF. Then, use AIC (or $AICc$), BIC, or HQIC (Hannan and Quinn (1979)).

This is the usual (& easier) approach.

R Note: The R function auto.arima uses $AICc$ to select p, q; d is selected using a formal unit root test (KPSS).

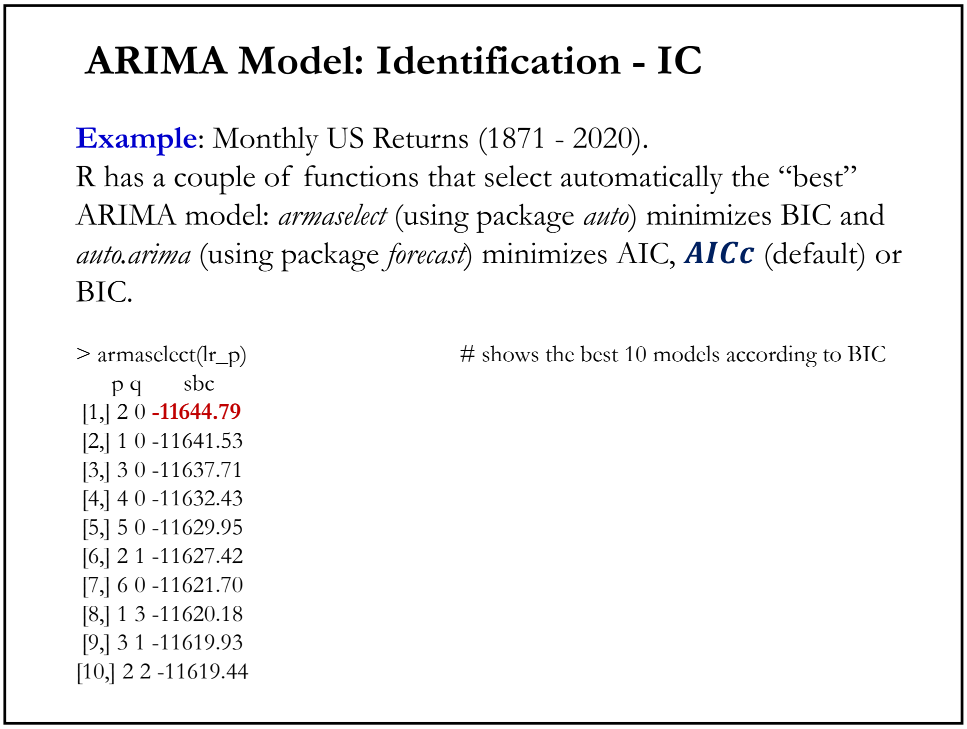
• Value parsimony. When in doubt, keep it simple (KISS).

ARIMA Model: Identification - IC

• We would like the IC 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.

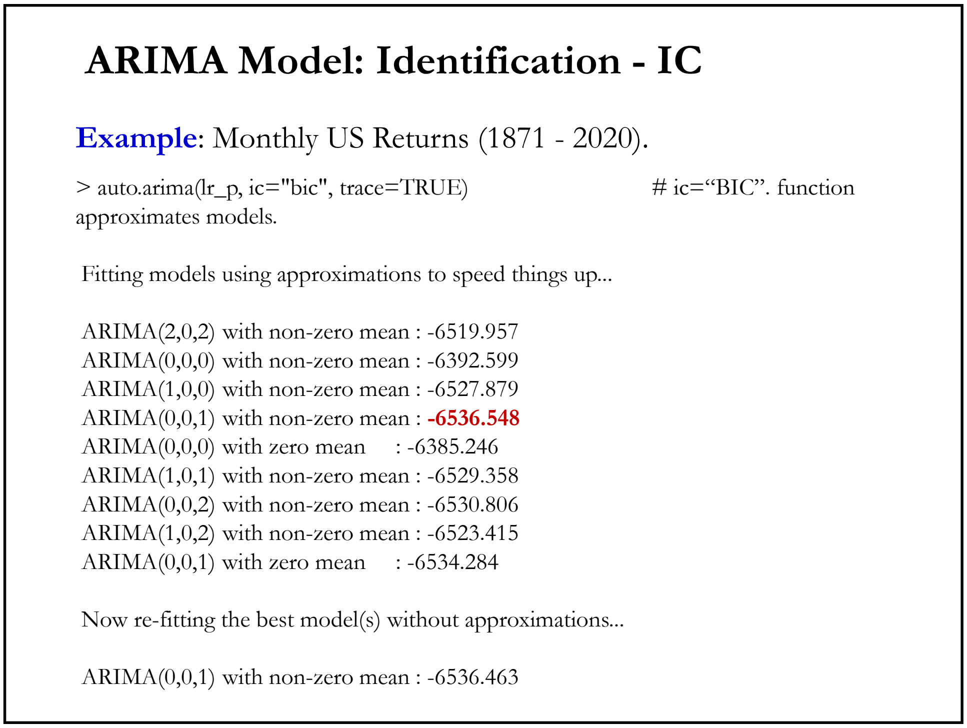
- AIC and Adjusted R^2 are **not consistent**.
- AIC is conservative –i.e., it tends to over-fit: k_{AIC} too large models.
- In time series, AIC selects the model that minimizes the out-of-sample one-step ahead forecast MSE.
- BIC is **more parsimonious** than AIC. It penalizes the inclusion of parameters more ($k_{BIC} \leq k_{AIC}$).
- BIC is **consistent** in autoregressive models.
- No agreement which criteria is better.



ARIMA Model: Identification - IC

Example: Monthly US Returns (1871 - 2020).
R has a couple of functions that select automatically the "best" ARIMA model: *armaselect* (using package *auto*) 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
```



ARIMA Model: Identification - IC

Example: Monthly US Returns (1871 - 2020).

```
> 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
```

ARIMA Model: Identification - IC

Example (continuation): Monthly US Returns (1871 - 2020).

```
> 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
```

- auto.arima does not try a lot of models, 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.

ARIMA Model: Identification - IC

- 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 – IC - 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_t.

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) **OLS for AR(p)**.

4) ***Innovations algorithm for* MA(q)**.

5) ***Hannan-Rissanen algorithm for* ARMA(p, q)**.

ARIMA Process – Estimation Hannan-Rissanen

5) ***Hannan-Rissanen algorithm for*** **ARMA(p, q)**

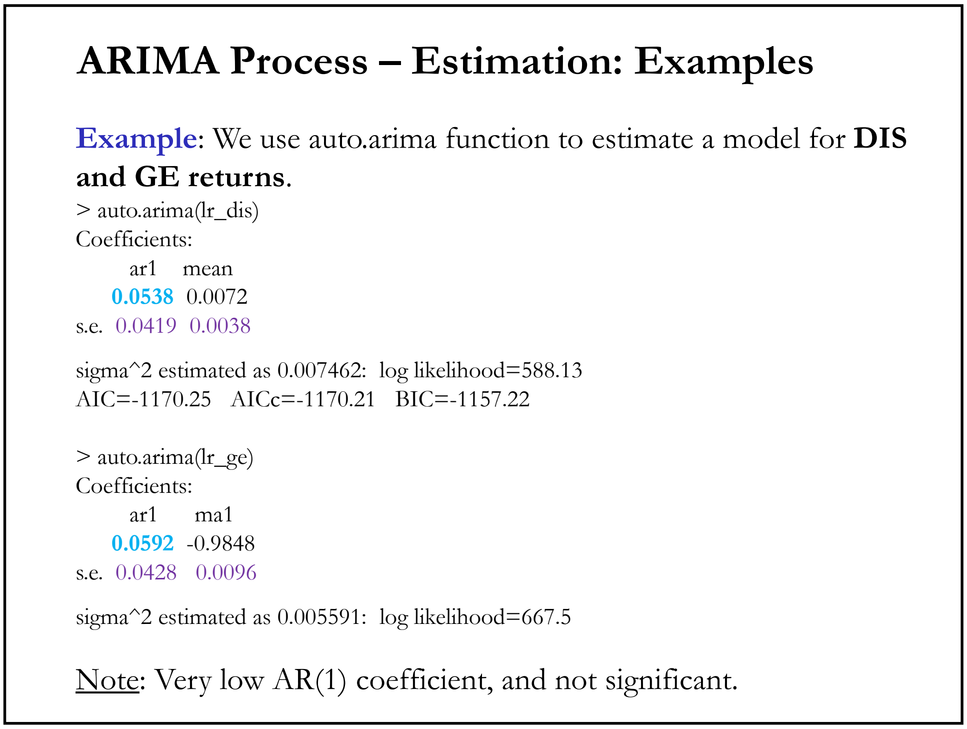
Steps:
1. Estimate high-order AR.
2. Use Step (1) to estimate (unobserved) noise ε_t
3. Regress y_t against $y_{t-1}, y_{t-2}, \ldots, y_{t-p}, \hat{\varepsilon}_{t-1}, \ldots, \hat{\varepsilon}_{t-q}$
4. Get new estimates of ε_t. Repeat Step (3).

ARIMA Process – Estimation: Examples

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.

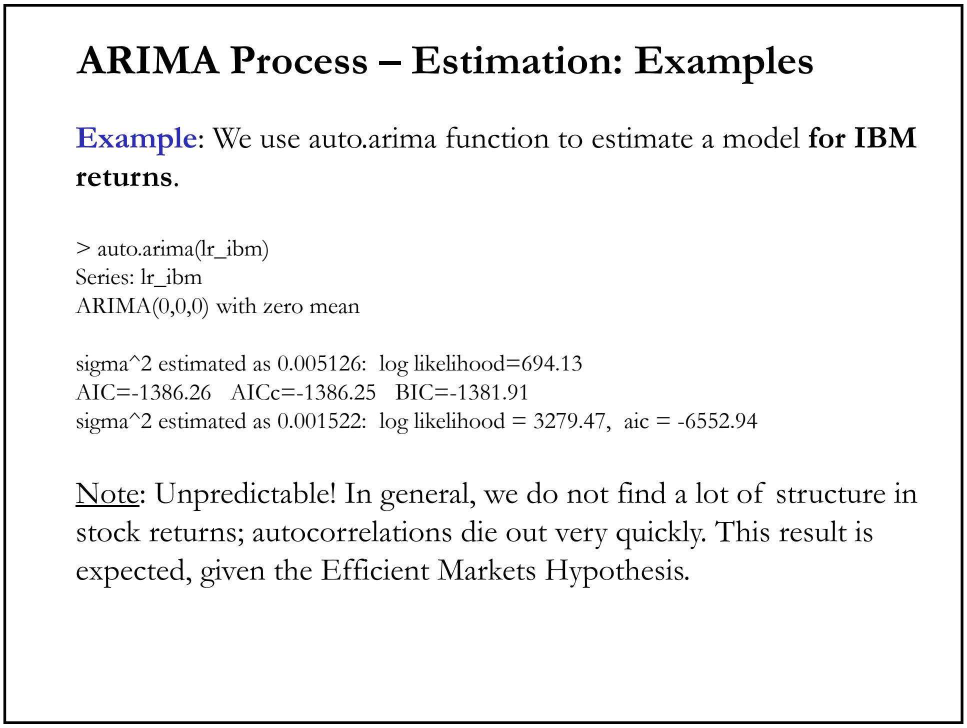


ARIMA Process – Estimation: Examples

Example: We use auto.arima function to estimate a model for **DIS and GE 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.

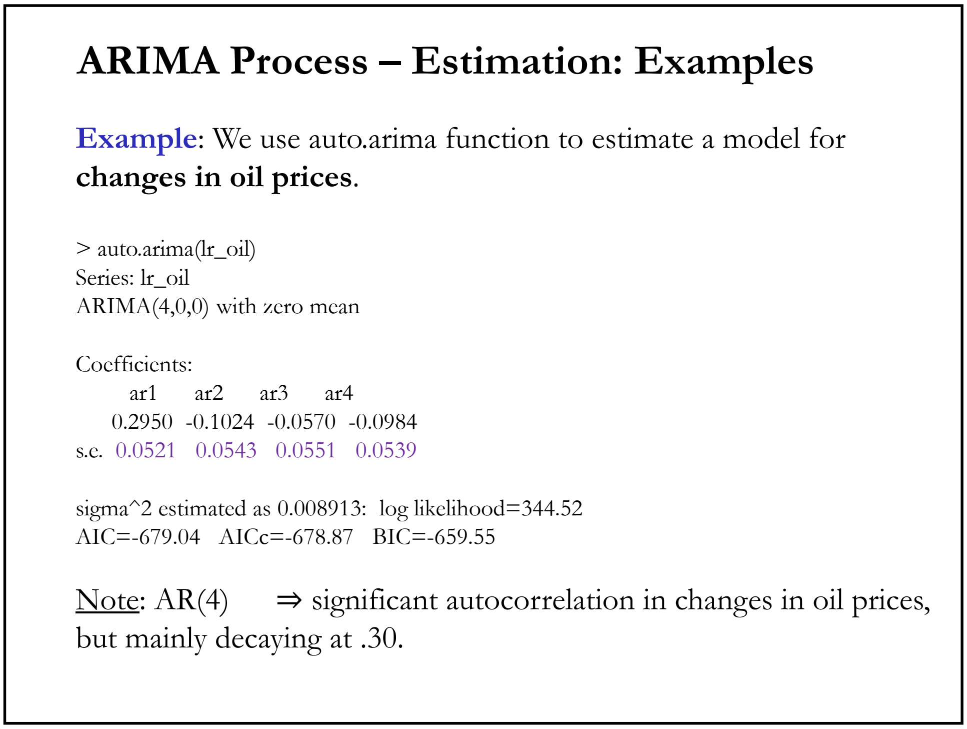


ARIMA Process – Estimation: Examples

Example: We use auto.arima function to estimate a model **for IBM returns**.

```
> 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.

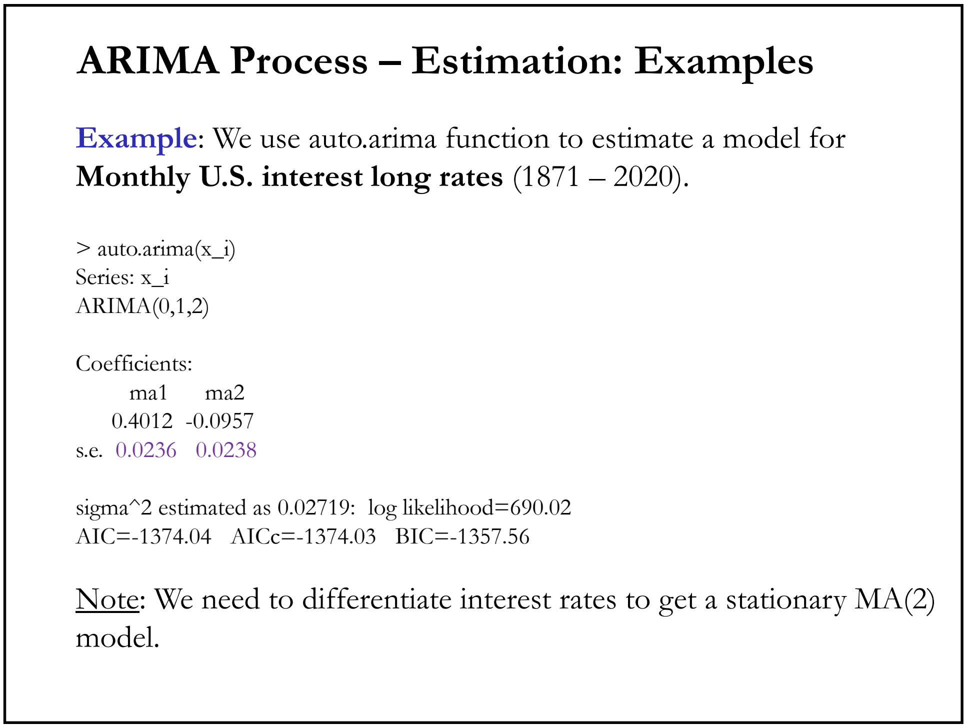


ARIMA Process – Estimation: Examples

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.



ARIMA Process – Estimation: Examples

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.

- Check for extra-AR structure in the mean.
- Check visual plots of residuals, ACFs, and the distribution of residuals.
- 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.

- We can also use the function *autoplot*() to check the stability of the roots. *Autoplot* graphs the *inverse roots*, not the roots. Thus we have the reverse stationarity result: If the inverse roots are inside the unit circle, the process is stationary.

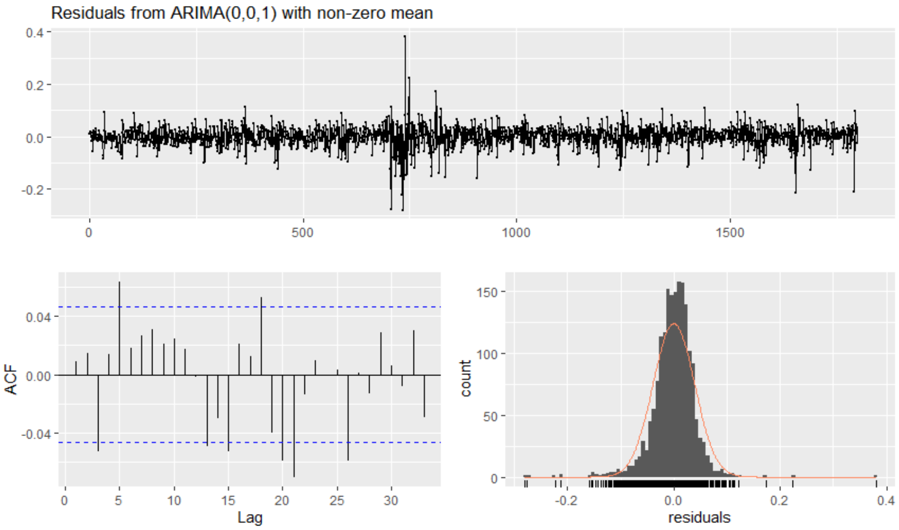
ARIMA Process – Diagnostic Tests

Example: We check the MA(1) model for **U.S. long returns**

```
> 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


fit_arima_lr_p <- arima(lr_p, order=c(0,0,1), method="ML")
> checkresiduals(fit_arima_lr_p)                    # Check if there is extra AR structure

       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
```

ARIMA Process – Diagnostic Tests

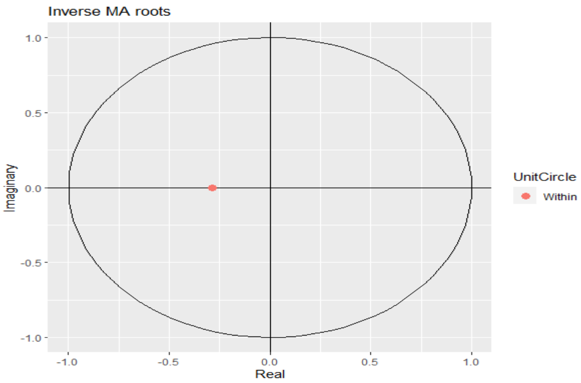
Example (continuation):



ARIMA Process – Diagnostic Tests

Example (continuation): We check stationarity/invertibility too -i.e., if the roots are inside the unit circle.

```
> autoplot(fit_arima_lr_p)                    # Check if inverse roots inside unit circle
```



Note: All inverse roots are inside unit circle & real: invertible MA(1).

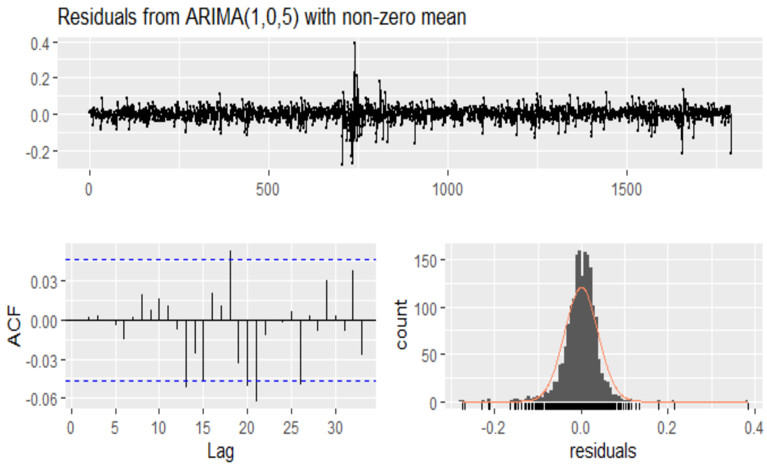
ARIMA Process – Diagnostic Tests

Example: We change the model for **U.S. long 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
```

ARIMA Process – Diagnostic Tests

Example (continuation):

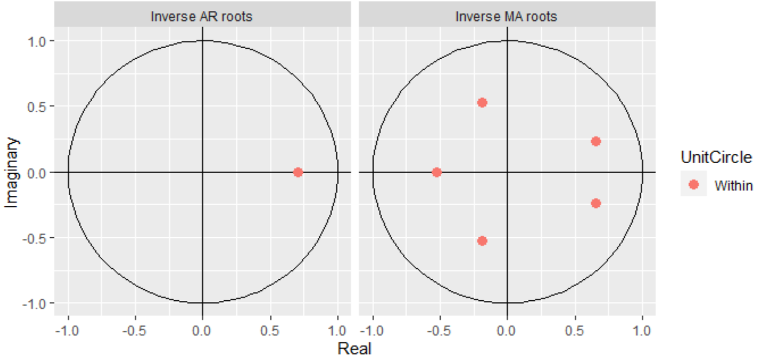


Note: We still see some small autocorrelations different from 0.

ARIMA Process – Diagnostic Tests

Example (continuation): We check the stationarity and invertibility of ARIMA(1,0,5) model

```
> autoplot(fit_arima_lr_p15)
```



Note: All *inverse* roots inside the unit circle: stationary and invertible. Notice that we have some roots on the MA part that are imaginary.

ARIMA: Forecasting

- Forecasting is the primary objective of ARIMA modeling.

- Two types of forecasts.

- **In sample** (prediction): The expected value of the RV (in-sample), the "fitted values," $\hat{Y}_t$.

- **Out of sample** (forecasting): The value of a future RV that is not observed by the sample, $\hat{Y}_{T+\ell}$. This is what we are going to do.

Notation:

- Forecast for $T+\ell$ made at T: $\hat{Y}_{T+\ell}, \hat{Y}_{T+\ell|T}, \hat{Y}_T(\ell)$.

- $T+\ell$ forecast error: $e_{T+\ell} = e_T(\ell) = Y_{T+\ell} - \hat{Y}_{T+\ell}$

- Mean squared error (MSE): $MSE(e_{T+\ell}) = E[Y_{T+\ell} - \hat{Y}_{T+\ell}]^2$

ARIMA: Forecasting – Basic Concepts

- 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 $\mathrm{E}[Y_{T+\ell} \,|\, I_T]$ depends on the distribution of $\{\varepsilon_t\}$. Then, if

$\{\varepsilon_t\} \sim \mathrm{WN} \quad \Rightarrow \mathrm{E}[\varepsilon_{T+\ell} \,|\, I_T] = 0.$

ARIMA: Forecasting Steps for ARMA Models

- Process:

(1) Find ARIMA model
(Use ACF, PACF or Minic)

$$Y_t = \phi\, Y_{t-1} + \varepsilon_t$$
$$\Downarrow$$

(2) Estimation
(& Evaluation in-sample)

$$\hat{\phi} \text{ (Estimate of } \phi)$$
$$\Downarrow$$
$$\hat{Y}_t = \hat{\phi}\, Y_{t-1} \text{ (Prediction)}$$

(3) Forecast
(& Evaluation out-of-sample)

$$\Downarrow$$
$$\hat{Y}_{t+1} = \hat{\phi}\, \hat{Y}_t \text{(Forecast)}$$

ARIMA: Forecasting From ARMA Models

- We observe the time series: $I_T = \{Y_1, Y_2, \ldots, Y_T\}$.

- We determine an ARIMA(p, d, q) model.

- At time T, we want to forecast: $Y_{t+1}, Y_{t+2}, \ldots, Y_{T+\ell}$.

- The information we have is $\{Y_1, Y_2, \ldots, Y_T, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T\}$.

- Use the conditional expectation of $Y_{T+\ell}$, given the information at T:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} | Y_T, Y_{T-1}, \ldots, Y_1]$$

Example: We have an AR(1) model.

$$Y_{T+1} = \mu + \phi_1 Y_T + \varepsilon_{T+1}$$

Then, the one-step ahead forecast:

$$\hat{Y}_{T+1} = E[Y_{T+1} | Y_T, Y_{T-1}, \ldots, Y_1] = \mu + \phi_1 Y_T$$

since $E[\varepsilon_{T+1} | Y_T, Y_{T-1}, \ldots, Y_1] = 0$.

ARIMA: Forecasting From MA(q) Models

- The stationary MA(q) model for Y_t is

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

We produce at time T *l-step ahead* forecasts using:

$$Y_{T+1} = \mu + \varepsilon_{T+1} + \theta_1 \varepsilon_T + \cdots + \theta_q \varepsilon_{T-q+1}$$

$$Y_{T+2} = \mu + \varepsilon_{T+2} + \theta_1 \varepsilon_{T+1} + \cdots + \theta_q \varepsilon_{T-q+2}$$

$$\vdots$$

$$Y_{T+\ell} = \mu + \varepsilon_{T+l} + \theta_1 \varepsilon_{T+l-1} + \cdots + \theta_q \varepsilon_{T+l-q} \qquad (l > 2)$$

Now, we take conditional expectations:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} | I_T] = \mu + \mathrm{E}[\varepsilon_{T+\ell} | I_T] + \theta_1 \mathrm{E}[\varepsilon_{T+\ell-1} | I_T] + \\ + \cdots + \theta_q \mathrm{E}\left[\varepsilon_{T+\ell-q} | I_T\right]$$

Note: Forecasts are a linear combination of errors.

ARIMA: Forecasting From MA(q) Models

- Some of the errors are know at T: $\varepsilon_1 = \hat{\boldsymbol{\varepsilon}}_{\mathbf{1}}$, $\varepsilon_2 = \hat{\boldsymbol{\varepsilon}}_{\mathbf{2}}$, ..., $\varepsilon_T = \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{T}}$, 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 & ε_{t-1}. Set $\mathrm{E}\left[\varepsilon_{t+j} | I_t\right]$=0 for $j > 1$. Then,

$$\hat{Y}_{t+1} = \mu + \theta_1 \mathrm{E}[\varepsilon_t | I_t] + \theta_2 \mathrm{E}[\varepsilon_{t-1} | I_t] = \mu + \theta_1 \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{t}} + \theta_2 \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{t-1}}$$
$$\hat{Y}_{t+2} = \mu + \theta_2 \mathrm{E}[\varepsilon_t | I_t] = \mu + \theta_2 \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{t}}$$
$$\hat{Y}_{t+3} = \mu$$
$$\hat{Y}_{t+\ell} = \mu \qquad \text{for } \ell > 2. \quad \Rightarrow \text{MA(2) memory of 2 periods}$$

ARIMA: Forecasting From MA(q) Models

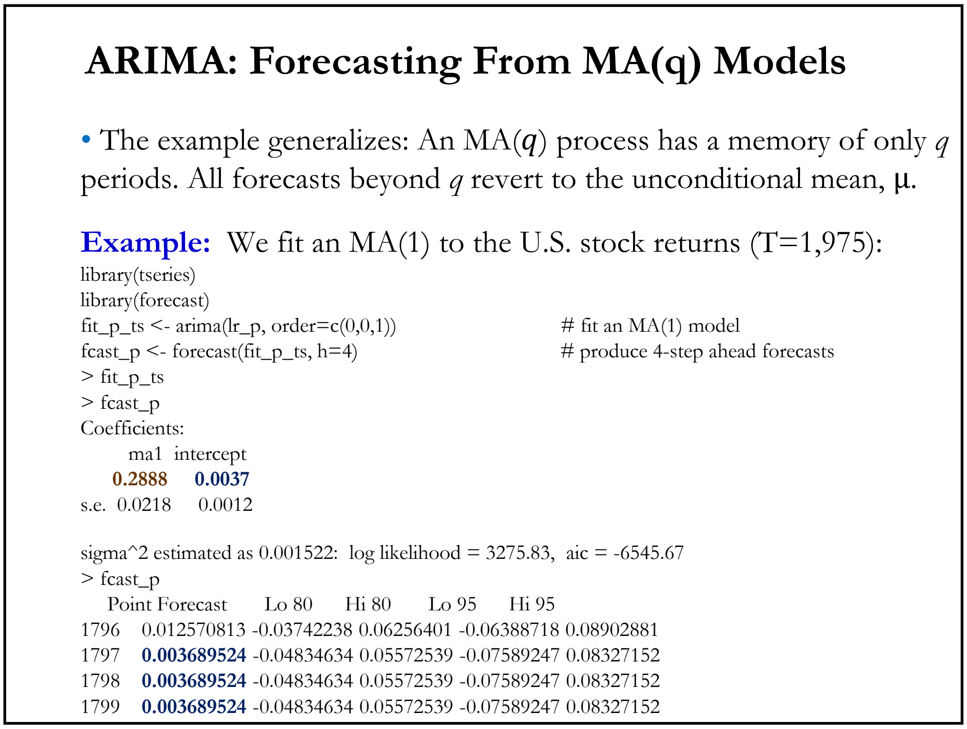
Example (continuation):

- At time t, we estimate the model: $\hat{\mu} = 0.28$, $\hat{\theta}_1 = 0.42$, & $\hat{\theta}_2 = 0.12$.

We also observe $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{t}} = \mathbf{0.45}$ & $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{t-1}} = \mathbf{-0.93}$.

Then, the forecasts are:

$$\hat{Y}_{t+1} = 0.28 + 0.42 * \mathbf{0.45} + 0.12 * \mathbf{-0.93} = 0.3574$$
$$\hat{Y}_{t+2} = 0.28 + 0.12 * \mathbf{0.45} = 0.334$$
$$\hat{Y}_{t+3} = 0.28$$
$$\hat{Y}_{t+\ell} = 0.28 \qquad \text{for } \ell > 2.$$

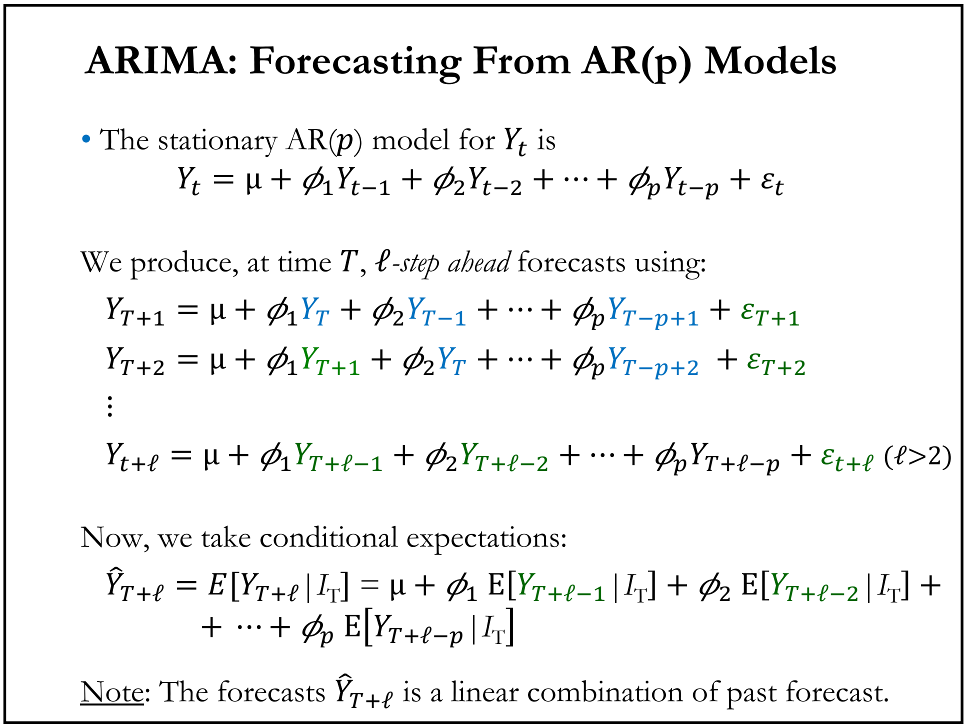


ARIMA: Forecasting From MA(q) Models

- 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
```



ARIMA: Forecasting From AR(p) Models

- The stationary AR(p) model for Y_t is

$$Y_t = \mu + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \varepsilon_t$$

We produce, at time T, ℓ-*step ahead* forecasts using:

$$Y_{T+1} = \mu + \phi_1 Y_T + \phi_2 Y_{T-1} + \cdots + \phi_p Y_{T-p+1} + \varepsilon_{T+1}$$

$$Y_{T+2} = \mu + \phi_1 Y_{T+1} + \phi_2 Y_T + \cdots + \phi_p Y_{T-p+2} + \varepsilon_{T+2}$$

$$\vdots$$

$$Y_{t+\ell} = \mu + \phi_1 Y_{T+\ell-1} + \phi_2 Y_{T+\ell-2} + \cdots + \phi_p Y_{T+\ell-p} + \varepsilon_{t+\ell} \quad (\ell{>}2)$$

Now, we take conditional expectations:

$$\hat{Y}_{T+\ell} = E[Y_{T+\ell} \,|\, I_T] = \mu + \phi_1 \, E[Y_{T+\ell-1} \,|\, I_T] + \phi_2 \, E[Y_{T+\ell-2} \,|\, I_T] + + \cdots + \phi_p \, E[Y_{T+\ell-p} \,|\, I_T]$$

Note: The forecasts $\hat{Y}_{T+\ell}$ is a linear combination of past forecast.

ARIMA: Forecasting From AR(p) Models

Example: AR(2) model for $Y_{t+\ell}$ is

$$Y_{t+\ell} = \mu + \phi_1 Y_{t+\ell-1} + \phi_2 Y_{t+\ell-2} + \varepsilon_{t+\ell}$$

Then, taking conditional expectations at $T = t$, we get the forecasts:

$$\hat{Y}_{t+1} = \mu + \phi_1 Y_t + \phi_2 Y_{t-1}$$
$$\hat{Y}_{t+2} = \mu + \phi_1 \hat{Y}_{t+1} + \phi_2 Y_t$$
$$\hat{Y}_{t+3} = \mu + \phi_1 \hat{Y}_{t+2} + \phi_2 \hat{Y}_{t+1}$$
$$\vdots$$
$$\hat{Y}_{t+\ell} = \mu + \phi_1 \hat{Y}_{t+\ell-1} + \phi_2 \hat{Y}_{T+\ell-1}$$

- AR-based forecasts are autocorrelated, they have long memory!

- At time t, we estimate the model: $\hat{\mu} = 0$, $\hat{\phi}_1 = .803$, & $\hat{\phi}_2 = .682$.

We also observe $Y_t = \mathbf{1.55}$ & $Y_{t-1} = \mathbf{3.03}$. Then,

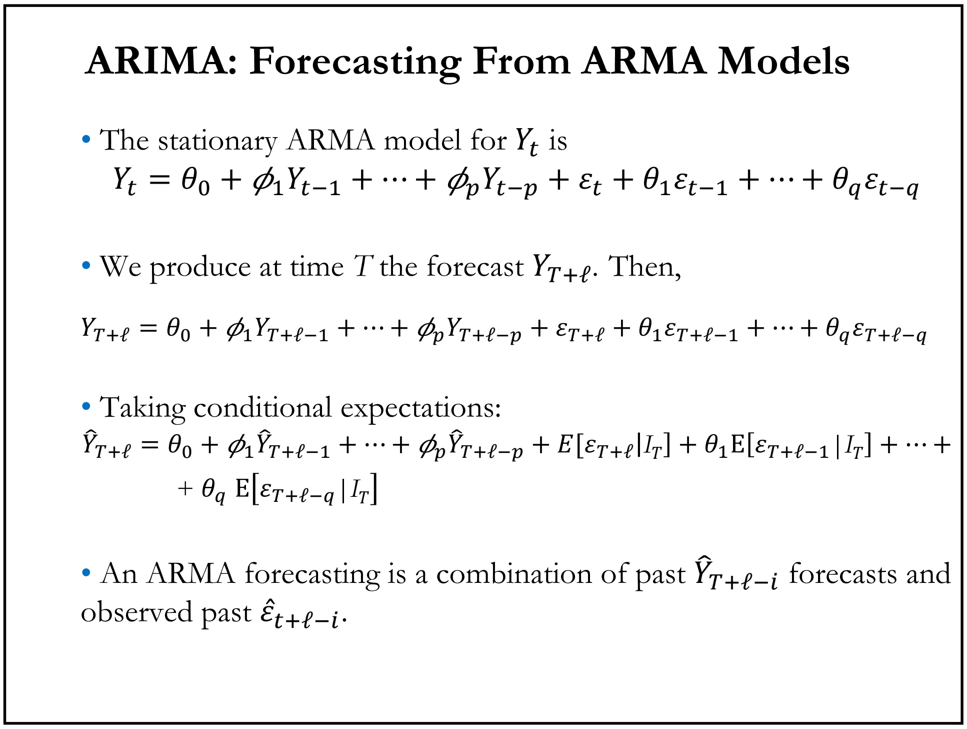
$$\hat{Y}_{t+1} = .803 * \mathbf{1.55} + .682 * \mathbf{3.03} = 3.3111$$
$$\hat{Y}_{t+2} = .803 * 3.3111 + .682 * \mathbf{1.55} = \mathbf{3.715921}$$

ARIMA: Forecasting From AR(p) Models

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.



ARIMA: Forecasting From ARMA Models

- The stationary ARMA model for Y_t is

$$Y_t = \theta_0 + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

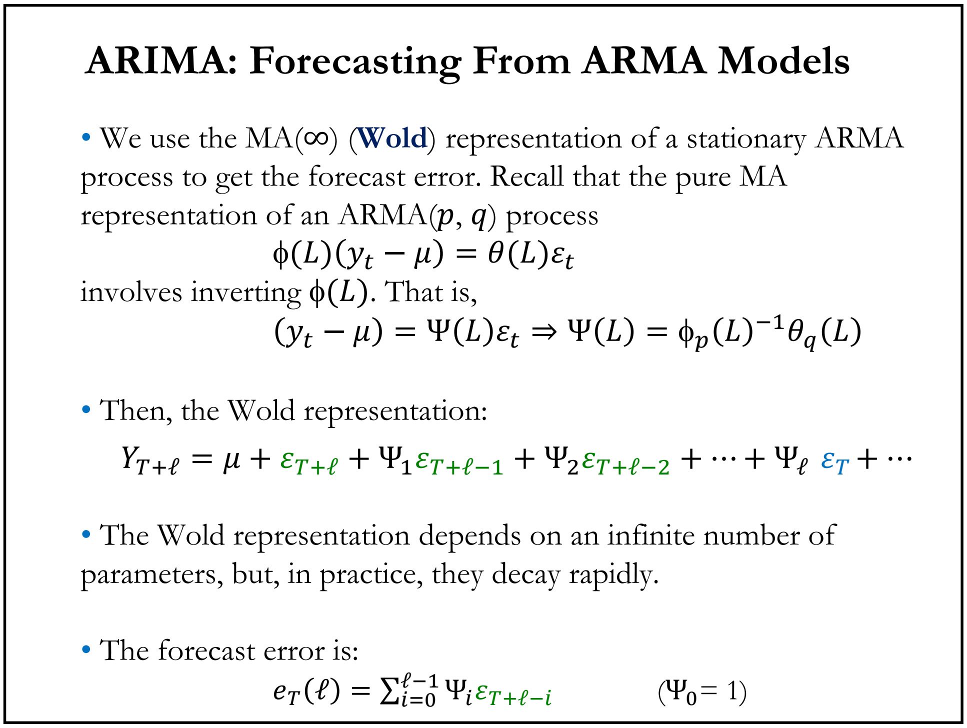
- We produce at time *T* the forecast $Y_{T+\ell}$. Then,

$$Y_{T+\ell} = \theta_0 + \phi_1 Y_{T+\ell-1} + \cdots + \phi_p Y_{T+\ell-p} + \varepsilon_{T+\ell} + \theta_1 \varepsilon_{T+\ell-1} + \cdots + \theta_q \varepsilon_{T+\ell-q}$$

- Taking conditional expectations:

$$\hat{Y}_{T+\ell} = \theta_0 + \phi_1 \hat{Y}_{T+\ell-1} + \cdots + \phi_p \hat{Y}_{T+\ell-p} + E[\varepsilon_{T+\ell}|I_T] + \theta_1 \mathrm{E}[\varepsilon_{T+\ell-1}\,|\,I_T] + \cdots +$$
$$+ \theta_q \, \mathrm{E}\left[\varepsilon_{T+\ell-q}\,|\,I_T\right]$$

- An ARMA forecasting is a combination of past $\hat{Y}_{T+\ell-i}$ forecasts and observed past $\hat{\varepsilon}_{t+\ell-i}$.



ARIMA: Forecasting From ARMA Models

- We use the MA(∞) (**Wold**) representation of a stationary ARMA process to get the forecast error. Recall that the pure MA representation of an ARMA(p, q) process

$$\phi(L)(y_t - \mu) = \theta(L)\varepsilon_t$$

involves inverting $\phi(L)$. That is,

$$(y_t - \mu) = \Psi(L)\varepsilon_t \Rightarrow \Psi(L) = \phi_p(L)^{-1}\theta_q(L)$$

- Then, the Wold representation:

$$Y_{T+\ell} = \mu + \varepsilon_{T+\ell} + \Psi_1 \varepsilon_{T+\ell-1} + \Psi_2 \varepsilon_{T+\ell-2} + \cdots + \Psi_\ell \, \varepsilon_T + \cdots$$

- The Wold representation depends on an infinite number of parameters, but, in practice, they decay rapidly.

- The forecast error is:

$$e_T(\ell) = \textstyle\sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i} \qquad (\Psi_0 = 1)$$

ARIMA: Forecasting From ARMA Models

- The forecast error is:

$$e_T(\ell) = \sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i} \qquad (\Psi_0 = 1)$$

Note: If the expected forecast error is zero, E[$e_T(\ell)$] = 0, we say the forecast is **unbiased**.

- The variance of the forecast error:

$$Var\big(e_T(\ell)\big) = Var\Big(\sum_{i=0}^{\ell-1} \Psi_i \varepsilon_{T+\ell-i}\Big) = \sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2 \qquad (\Psi_0 = 1)$$

Example 1: One-step ahead forecast (ℓ = 1).

$$Y_{T+1} = \mu + \varepsilon_{T+1} + \Psi_1 \varepsilon_T + \Psi_2 \varepsilon_{T-1} + \Psi_3 \varepsilon_{T-2} + \cdots$$

Forecast: $\hat{Y}_{T+1} = \mu + \Psi_1 \varepsilon_T + \Psi_2 \varepsilon_{T-1} + \cdots$

Forecast error: $e_T(1) = Y_{T+1} - \hat{Y}_{T+1} = \varepsilon_{T+1}$

Variance: $Var\big(e_T(1)\big) = \sigma^2$

ARIMA: Forecasting From ARMA Models

Example 2: Two-step ahead forecast (ℓ = 2).

$$Y_{T+2} = \mu + \varepsilon_{T+2} + \Psi_1 \varepsilon_{T+1} + \Psi_2 \varepsilon_T + \Psi_3 \varepsilon_{T-1} + \cdots$$
$$\hat{Y}_{T+2} = \mu + \Psi_2 \varepsilon_T + \Psi_3 \varepsilon_{T-1} + \cdots$$
$$e_T(2) = Y_{T+2} - \hat{Y}_{T+2} = \varepsilon_{T+2} + \Psi_1 \varepsilon_{T+1}$$
$$Var\big(e_T(2)\big) = \sigma^2 * (1 + \Psi_1^2)$$

Similarly,

$$e_T(3) = Y_{T+3} - \hat{Y}_{T+3} = \varepsilon_{T+3} + \Psi_1\, \varepsilon_{T+2} + \Psi_2\, \varepsilon_{T+1}$$
$$Var\big(e_T(3)\big) = \sigma^2 * (1 + \Psi_1^2 + \Psi_2^2)$$

Note: $\lim_{\ell \to \infty} \hat{Y}_T(\ell) = \mu$

- In practice, the Ψ_i's decay rapidly. Then, as we forecast into the future, the forecasts are not very interesting (unconditional forecasts!).
- This is why ARIMA forecasting is useful only for short-term.

Review: Forecasting From ARMA Models: C.I.

- A 100(1 - α)% prediction interval for $Y_{T+\ell}$ (ℓ-steps ahead) is

$$\hat{Y}_T(\ell) \pm z_{\alpha/2} \sqrt{Var(e_T(\ell))}$$

$$\hat{Y}_T(\ell) \pm z_{\alpha/2}\, \sigma \sqrt{\textstyle\sum_{i=0}^{\ell-1} \Psi_i^2} \qquad (\Psi_0 = 1)$$

Example: 95% C.I. for the 1-step and 2-step-ahead forecasts:

$$\hat{Y}_T(1) \pm 1.96\, \sigma$$

$$\hat{Y}_T(2) \pm 1.96\, \sigma \sqrt{1 + \Psi_1^2}$$

- 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[$e_{T+\ell}$] = $\sigma^2 \sum_{i=0}^{\ell-1} \Psi_i^2$

Forecasting From Simple Models: ES

- Industrial companies, with a lot of inputs and outputs, want quick and inexpensive forecasts. Easy to fully automate. In general, we use past Y_t to forecast future Y_t's, usually referred as the **level's forecasts**.

- 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) & 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 called *level's forecast*. Smooth? The graph of S_t is less jagged than the graph of the original series, Y_t.

- We use the observed time series at time t: $Y_1, Y_2, \ldots, Y_t$.

- The equation for the **level**: $S_t = \alpha Y_{t-1} + (1-\alpha) S_{t-1}$

where

- α: The smoothing parameter, $0 \leq \alpha \leq 1$.
- 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**:

$$S_t = S_{t-1} + \alpha(Y_{t-1} - S_{t-1}) = S_{t-1} + \alpha * (\text{past forecast error})$$

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 for next period ($t+1$):

$$S_{t+1} = \alpha Y_t + (1-\alpha) S_t = S_t + \alpha(Y_t - S_t)$$

That is, a simple updating forecast: last period forecast + adjustment.

The forecast for the period $t+2$, 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:

$S_{t+\ell} = S_{t+1}$ ⇒ A naive forecast!

<u>Note</u>: SES forecasts are not very interesting after $\ell > 1$.

SES: Forecast and Updating

Example: An industrial firm uses SES to forecast sales:

$$S_{t+1} = S_t + \alpha \, * \, (Y_t - S_t)$$

The firm estimates α = **0.25**. The firm observes Y_t = **5** and, last period's forecast, S_t = **3**.

Then, the forecast for time $t+1$ is:

$$S_{t+1} = \mathbf{3} + \mathbf{0.25} * (\mathbf{5} - \mathbf{3}) = \mathbf{3.50}$$

The forecast for time $t+1$ (& any period after time $t+1$) is:

$$S_{t+\ell} = S_{t+1} = \mathbf{3.50} \qquad \text{for } \ell > 1.$$

Later, the firm observes: Y_{t+1} = **4.77**, Y_{t+2} = **3.15**, & Y_{t+3} = **1.85**. Then, the MSE:

$$\text{MSE} = \frac{1}{3} * [(\mathbf{4.77} - \mathbf{3.50})^2 + (\mathbf{3.15} - \mathbf{3.50})^2 + (\mathbf{1.85} - \mathbf{3.50})^2] = \mathbf{1.486}.$$

SES: Forecast and Updating

Example (continuation):

Note: If α = **0.75**, then

$$S_{t+1} = \mathbf{3} + \mathbf{0.75} * (\mathbf{5} - \mathbf{3}) = \mathbf{4.50}$$

A bigger α gives more weight to the more recent observation –i.e., Y_t.

Again, the forecast for time $t+1$ and any period after time $t+1$ is:

$$S_{t+\ell} = S_{t+1} = \mathbf{4.50} \qquad \text{for } \ell > 1.$$

SES: Exponential?

- Q: 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:

$$S_{t+1} = \alpha Y_t + (1-\alpha) S_t = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha) S_{t-1}]$$
$$= \alpha Y_t + \alpha(1-\alpha) Y_{t-1} + (1-\alpha)^2 S_{t-1}$$
$$\Rightarrow \hat{Y}_t(1) = S_{t+1} = c_0 Y_t + c_1 Y_{t-1} + c_2 Y_{t-2} + \cdots$$

where c_i's are the weights, with

$$c_i = \alpha(1-\alpha)^i;\ i = 0, 1, \quad \ldots;\ 0 \leq \alpha \leq 1.$$

- We have decreasing weights, by a constant ratio for every unit increase in lag.

Then, $\hat{Y}_t(1) = \alpha(1-\alpha)^0 Y_t + \alpha(1-\alpha)^1 Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \cdots$

$\hat{Y}_t(1) = \alpha Y_t + (1-\alpha)\hat{Y}_{t-1}(1) \quad \Rightarrow S_{t+1} = \alpha Y_t + S_t$

75

SES: Exponential Weights

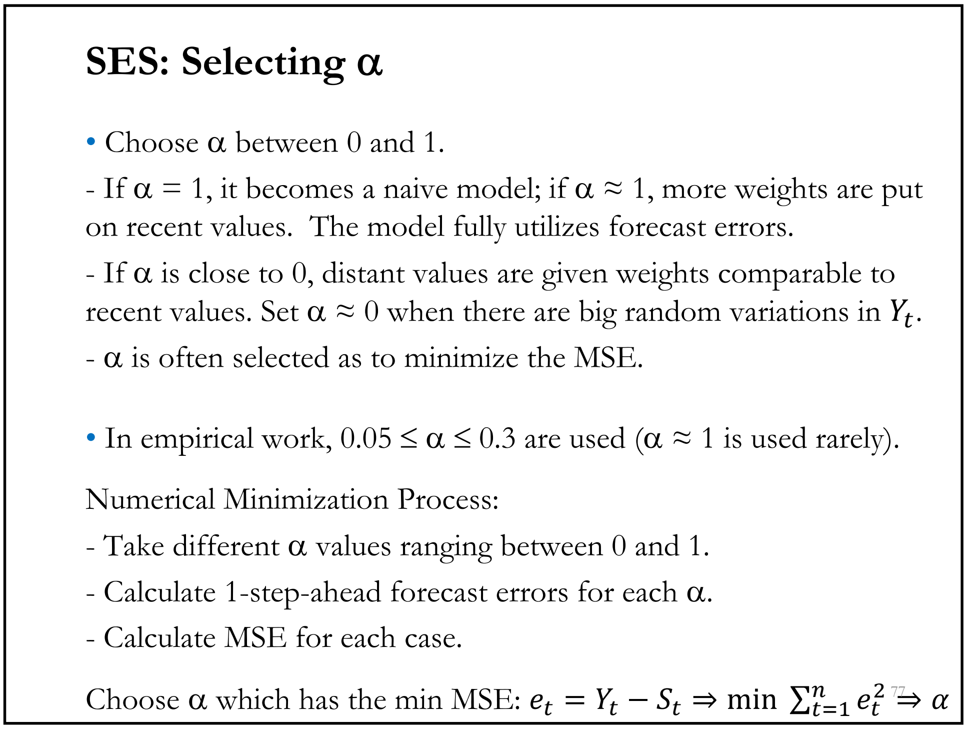
- $c_i = \alpha\,(1-\alpha)^i; \qquad i = 0, 1, \ldots;\ 0 \leq \alpha \leq 1.$

$c_i = \alpha(1-\alpha)^i$	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.

76



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.

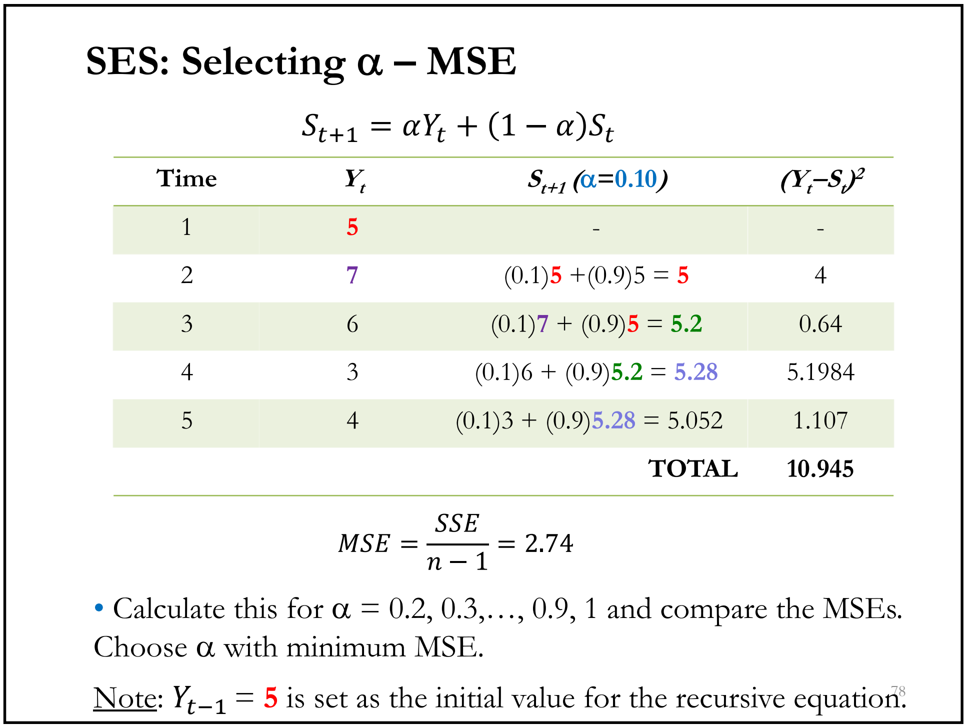
- α is often selected as to minimize the MSE.

- 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: $e_t = Y_t - S_t \Rightarrow \min \sum_{t=1}^{n} e_t^2 \Rightarrow \alpha$



SES: Selecting α – MSE

$$S_{t+1} = \alpha Y_t + (1-\alpha) S_t$$

Time	Y_t	S_{t+1} (α=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 equal to Y_1. Then, $S_2 = Y_1$.

– 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.)

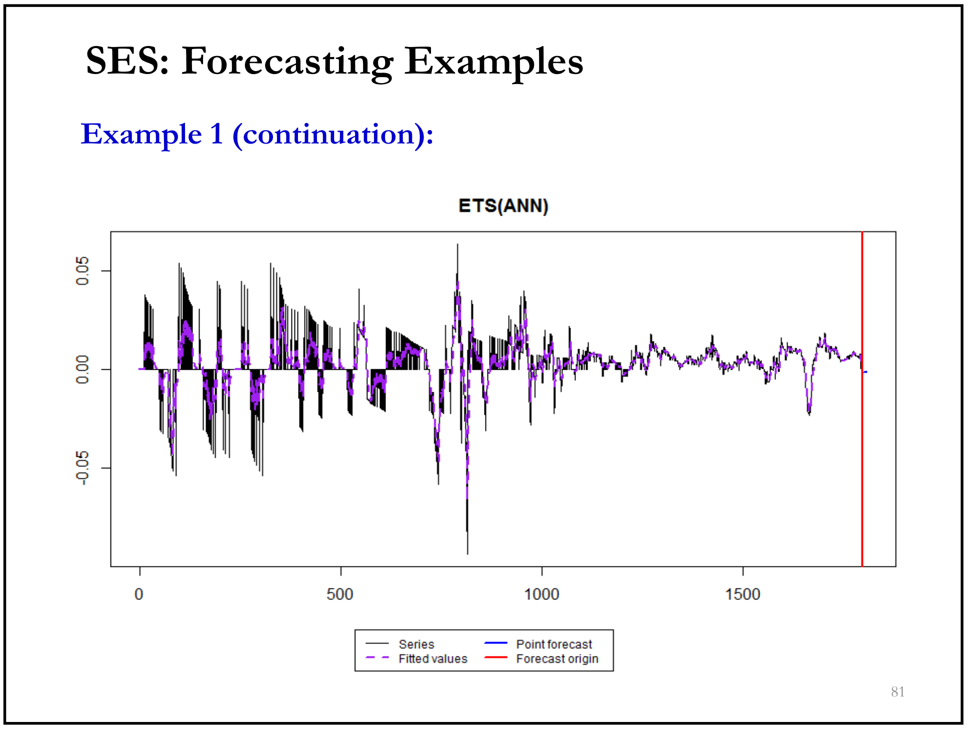
79

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.

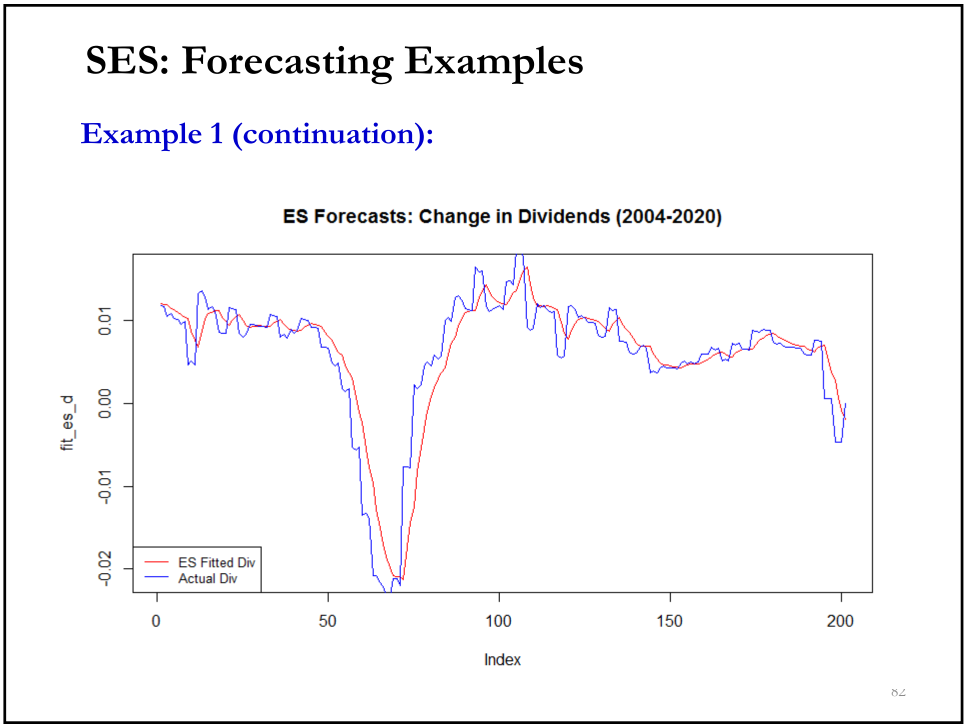
```
mod1 <- HoltWinters(lr_d[1:1750], beta=FALSE, gamma=FALSE)
> mod1
Holt-Winters exponential smoothing without trend and without seasonal component.

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
```

80



SES: Forecasting Examples

Example 1 (continuation):



SES: Forecasting Examples

Example 1 (continuation):

SES: Forecasting Examples

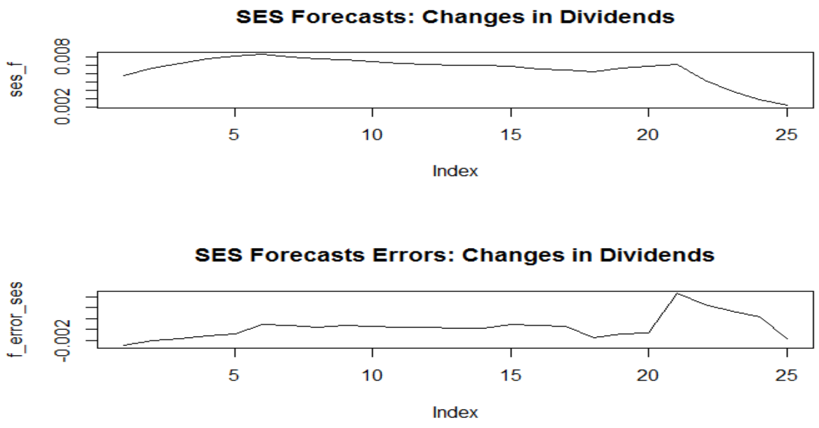
Example 1 (continuation): Now, we do one-step ahead forecasts

```
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")
```

SES: Forecasting Examples

Example 1 (continuation):

```
> ses_f
f_error_ses <- sm[T1:T] - y[T1:T]
> plot(ses_f, type="l", main ="SES Forecasts: Changes in Dividends")
```



SES: Forecasting U.S. Dividends

Example 1 (continuation): *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.

SES: Forecasting Examples

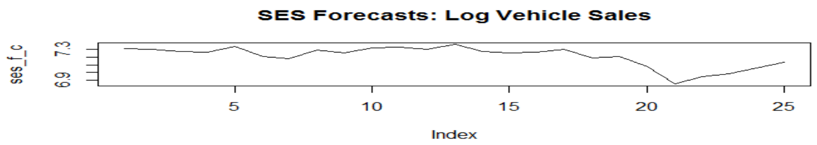
Example 2: We want to forecast **log monthly U.S. vehicles** (1976-2020, T=537) using SES.

```
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
```

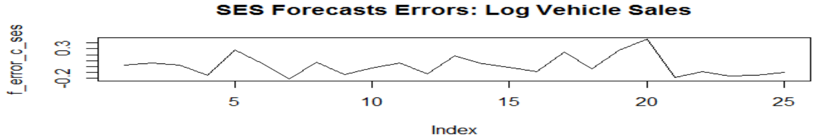
SES: Forecasting Examples

Example 2 (continuation): 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.

- Deterministic components, D_t, can be easily incorporated.

- The model that incorporates both a trend and seasonal features is called *Holt-Winter's ES*.