Lecture 10
Volatility Models

Brooks (4th edition): Chapter 9

1

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:

$$y_t = X_t\beta + \varepsilon_t, \qquad \varepsilon_t \sim i.i.d.\ D(0, \sigma^2)$$

• 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.

• 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.

Linear and Non-linear Models

• 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 series:

$$y_t = E[y_t \mid I_t] + \varepsilon_t, \qquad \varepsilon_t \sim D(0,\sigma^2)$$

Suppose we have an AR(1) process:

$$y_t = \alpha + \phi\, y_{t-1} + \varepsilon_t.$$

Then, the conditional mean, conditioning on information set at time t, I_t, is:

$$E_t[y_{t+1} \mid I_t] = \alpha + \phi\, y_t$$

• Recall the distinction between conditional moments and unconditional ones. The unconditional mean and variance are:

$$E[y_t] = \alpha/(1-\phi) = \text{constant}$$
$$Var[y_t] = \sigma^2/(1-\phi^2) = \text{constant}$$

The conditional mean is time varying; the unconditional mean is not!

ARCH Models

• Similar idea for the variance. Let's focus on the conditional variance, conditioning on:

Conditional variance:

$$\text{Var}[y_{t+1} \,|I_t] = E_t[(y_{t+1} - E_t[y_{t+1} \,|I_t])^2] = E_t[\varepsilon_{t+1}^2]$$

Unconditional variance:

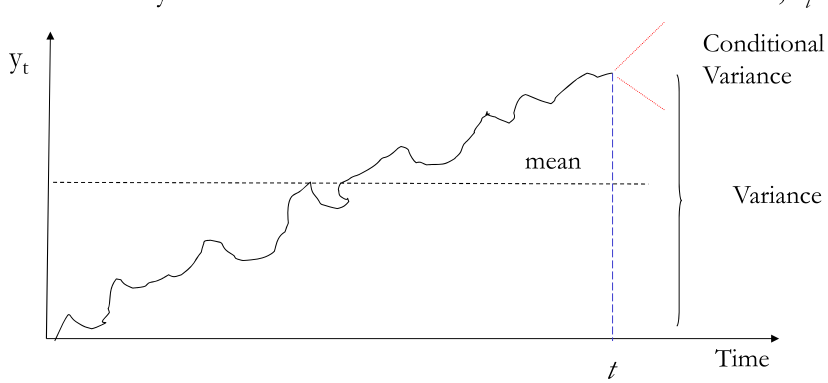
$$\text{Var}[y_{t+1}] = E[(y_{t+1} - E[y_{t+1}])^2] = \sigma^2/(1 - \phi^2)$$

Remark: Conditional moments are time varying; unconditional moments are not!

ARCH Models

• The unconditional variance measures the 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, $\text{Var}[y_t \,|I_t]$, is a better measure of uncertainty at time t. It is a function of information at time t, I_t.



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.

ARCH Models: Stylized Facts of Asset Returns

(6) *Volatility and serial correlation*: Inverse relationship between the two.

(7) *Co-movements in volatility*: Volatility is positively correlated across markets/assets.

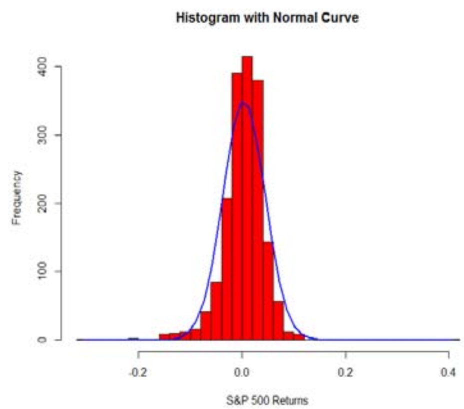
• We need a model that accommodates all these (non-linear) facts.

• Stylized facts (1) and (2) form the basis of Volatility (ARCH) Models.

ARCH Models: Stylized Facts of Asset Returns

• Easy to check leptokurtosis (Stylized Fact #1)

Figure: Descriptive Statistics and Distribution for Monthly S&P500 Returns

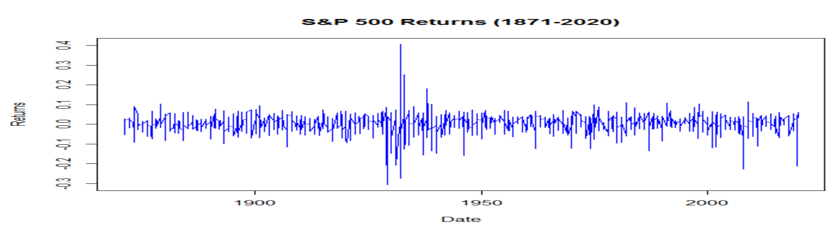


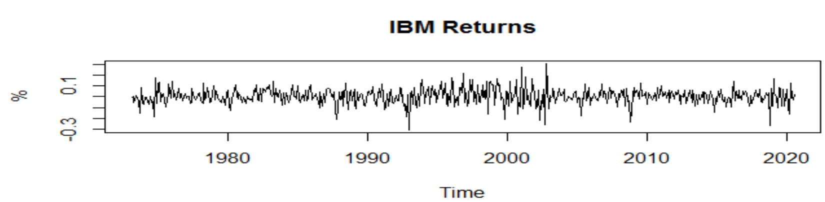
	Statistic
Mean (%)	0. 0585 (p-value: 0.0004)
Standard Dev (%)	0.0449
Skewness	-0.7294
Excess Kurtosis	2. 6406
Jarque-Bera	216.15 (*p-value*: <0.000001)

• Heavy tails: Excess kurtosis greater than 0!

ARCH Models: Stylized Facts of Asset Returns

• 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'**) ${\sigma_t}^2 = Var_{t-1}[\varepsilon_t] = E_{t-1}[\varepsilon_t^2] = \omega + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_q \varepsilon_{t-q}^2$

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

$\nu_t \equiv \varepsilon_t^2 - \sigma_t^2,$ -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.

ARCH Models: Engle (1982)

• 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 $\alpha_i > 0$ for all i.

Robert F. Engle (1942, USA)

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

• Even though the errors may be serially uncorrelated they are not independent: There will be volatility clustering, which produces fat tails. We define standardized errors:

$$z_t = \varepsilon_t / \sigma_t$$

• They have conditional mean zero and a time invariant conditional variance equal to 1. That is, $z_t \sim D(0, 1)$. If z_t is assumed to follow a N(0, 1), with a finite fourth moment (use Jensen's inequality). Then:

$$E[\varepsilon_t^4] = E[z_t^4]E[\sigma_t^4] > E[z_t^4]E[\sigma_t^2]^2 = E[z_t^4]E[\varepsilon_t^2]^2$$
$$= 3\, E[\varepsilon_t^2]^2$$
$$\kappa(\varepsilon_t) = E[\varepsilon_t^4]/E[\varepsilon_t^2]^2 > 3.$$

Technical point: It can be shown that for an ARCH(1), the 4th moment for an ARCH(1):

$$\kappa(\varepsilon_t) = \frac{3(1-\alpha^2)}{1-3\alpha^2} \qquad \text{if } 3\alpha^2 < 1.$$

ARCH Models: Alternative Representation

• 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.

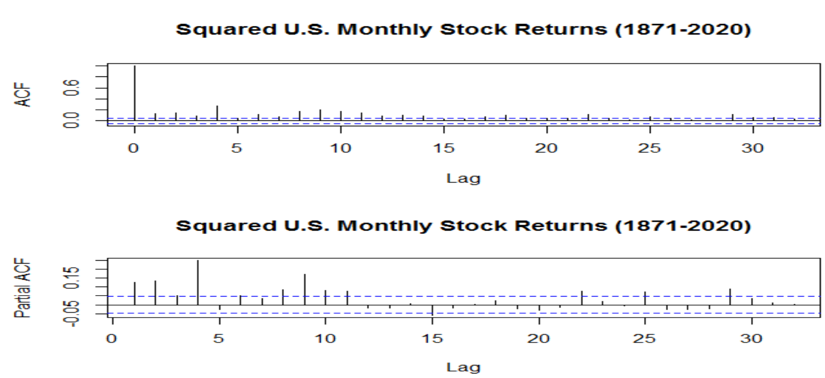
Also, if we assume υ_t is normally distributed, then

$$\varepsilon_t \sim N(0, \sigma_t^2).$$

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).



GARCH Model: Bollerslev (1986)

• 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.

GARCH Model: GARCH(1,1)

• 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] = $\sigma^2 = \omega / (1 - \alpha_1 - \beta_1)$.

⇒ 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

$$\alpha_1 + \beta_1 < 1.$$

GARCH Model: Determination of Order

• 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:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \delta\ f(X_{t-1}, \theta)$$

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:

$$f(y_1, y_2, \ldots, y_T; \boldsymbol{\theta}) = f(y_1|\mathbf{x_1}; \boldsymbol{\theta}) f(y_2|y_1, \mathbf{x}_2, \mathbf{x}_1; \boldsymbol{\theta}) f(y_3|y_2, y_1, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1; \boldsymbol{\theta}) \ldots$$
$$\ldots f(y_T|y_{T-1}, \ldots, y_1, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_1; \boldsymbol{\theta}).$$

Taking logs:

$$L = \log(f(y_1, y_2, \ldots, y_T; \boldsymbol{\theta}))) = \log(f(y_1|\mathbf{x_1}; \boldsymbol{\theta})) + \log(f(y_2|y_1, \mathbf{x}_2, \mathbf{x}_1; \boldsymbol{\theta}))$$
$$+ \ldots + \log(f(y_T|y_{T-1}, \ldots, y_1, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_1; \boldsymbol{\theta}))$$
$$= \sum_{t=1}^{T} \log(f(y_t|Y_{t-1}, X_t; \boldsymbol{\theta}))$$

We maximize this function with respect to the *k* mean parameters (γ) and the *m* variance parameters (ω, α, β).

ARCH Estimation: MLE – ARCH(1)

• **Example**: ARCH(1) model.

Mean equation: $y_t = \boldsymbol{X}_t \boldsymbol{\gamma} + \varepsilon_t$, $\quad \varepsilon_t \sim \text{N}(0, \sigma_t^2)$

Variance equation: $\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2$

We write the pdf for the normal distribution,

$$f(\varepsilon_t|\gamma, \omega, \alpha_1) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left[-\frac{\varepsilon_t^2}{2\sigma_t^2}\right] = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left[-\frac{(y_t - X_t\boldsymbol{\gamma})^2}{2\sigma_t^2}\right]$$

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 $\boldsymbol{\theta} = (\gamma, \omega, \alpha_1)$ the function L.

ARCH Estimation: MLE – ARCH(1)

Example (continuation): ARCH(1) model.

$$L = -\frac{T}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\log(\omega + \alpha_1\varepsilon_{t-1}^2) - \frac{1}{2}\sum_{t=1}^{T}\varepsilon^2{}_t/(\omega + \alpha_1\varepsilon_{t-1}^2)$$

Taking derivatives with respect to $\boldsymbol{\theta} = (\omega, \alpha_1, \boldsymbol{\gamma})$, where $\boldsymbol{\gamma}$ is a vector of k mean parameters:

$$\frac{\partial L}{\partial \omega} = -\sum_{t=1}^{T} 1/(\omega + \alpha_1\varepsilon_{t-1}^2) - (-1/2)\sum_{t=1}^{T}\varepsilon_t^2/(\omega + \alpha_1\varepsilon_{t-1}^2)^2$$

$$\frac{\partial L}{\partial \alpha_1} = -\sum_{t=1}^{T}\varepsilon_{t-1}^2/(\omega + \alpha_1\varepsilon_{t-1}^2) - (-1/2)\sum_{t=1}^{T}\varepsilon_t^2\varepsilon_{t-1}^2/(\omega + \alpha_1\varepsilon_{t-1}^2)^2$$

$$\frac{\partial L}{\partial \boldsymbol{\gamma}} = -\sum_{t=1}^{T}\boldsymbol{X'}_t\varepsilon_t/\sigma_t^2 \qquad (k\text{x1 vector of derivatives})$$

ARCH Estimation: MLE – ARCH(1)

Example (continuation): 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:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = S(y_t, \boldsymbol{\theta}) = 0 \qquad \text{-a } (k+2) \text{ system of equations.}$$

The vector of first derivatives is called the score vector, $S(y_t, \boldsymbol{\theta})$.

Take the last f.o.c., the kx1 vector, $\frac{\partial L}{\partial \boldsymbol{\gamma}} = 0$:

$$\frac{\partial L}{\partial \boldsymbol{\gamma}} = -\sum_{t=1}^{T}\boldsymbol{X'}_t\varepsilon_t/\sigma_{t,MLE}^2 = \sum_{t=1}^{T}\boldsymbol{X'}_t(y_t - \boldsymbol{X}_t\boldsymbol{\gamma}_{MLE})/\sigma_{t,MLE}^2 = 0$$

$$= \sum_{t=1}^{T}\frac{\boldsymbol{X'}_t}{\sigma_{t,MLE}}\left(\frac{y_t}{\sigma_{t,MLE}} - \frac{\boldsymbol{X}_t}{\sigma_{t,MLE}}\boldsymbol{\gamma}_{MLE}\right) = 0$$

The last equation shows that MLE is GLS for the mean parameters, $\boldsymbol{\gamma}$, each observation is weighted by the inverse of $\sigma_{t,MLE}$.

ARCH Estimation: MLE

Example (continuation): 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).

ARCH Estimation: MLE – Standard Errors

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

$$T^{1/2}(\hat{\theta} - \theta_0) \to N(0, A_0^{-1}), \quad where A_0 = T^{-1}\sum_{t=1}^{T}\frac{\partial S_t(y_t, \theta_0)}{\partial \theta}$$

• 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, $\boldsymbol{\theta}_0$.
- 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

ARCH Estimation: Numerical Optimization

• Initial values:

- Numerical optimization needs initial values for $\boldsymbol{\theta}$, say $\boldsymbol{\theta}_0$. 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:

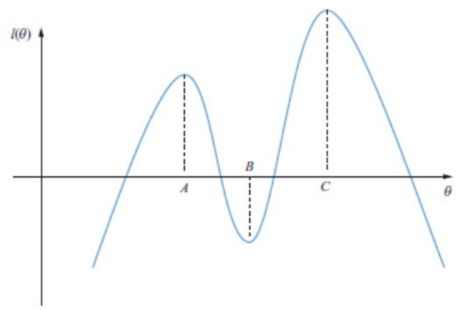


Figure 9.2 The problem of local optima in maximum likelihood estimation

ARCH Estimation: Numerical Optimization

• Initial values (continuation):

- 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 ε_0, ..., ε_q (and ε_0, ε_1 , ..., ε_p if we assume an AR(p) process for the mean).

<u>Usual assumptions</u>: σ_0 = unconditional SD; $\varepsilon_0 = \varepsilon_1 = ... = \varepsilon_p = 0$.

- Alternatively, we can take σ_0 (and ε_0, ε_1, ..., ε_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
```

ARCH Estimation: MLE – Example (in R)

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 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
```

ARCH Estimation: MLE – Example (in R)

Example 1 (continuation):

```
> 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
```

ARCH Estimation: MLE – Example (in R)

Example 1 (continuation):

```
> 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
```

ARCH Estimation: MLE – Example (in R)

Example 1 (continuation): Summary for CHF/USD changes

$e_{f,t} = [\log(S_t) - \log(S_{t-1})] = a_0 + a_1\, e_{f,t-1} + \varepsilon_t, \qquad \varepsilon_t \,|\, I_{t-1} \sim N(0, \sigma_t^2)$

$\sigma_t^2 = \omega + \alpha_1\, \varepsilon_{t-1}^2 + \beta_1\, \sigma_{t-1}^2.$

• T: 562 (January 1971 - July 2020, monthly).

The estimated model for s_t is given by:

$e_{f,t} =$ **-0.00211** + **0.02600** $e_{f,t-1}$,
(.0012) (0.044)

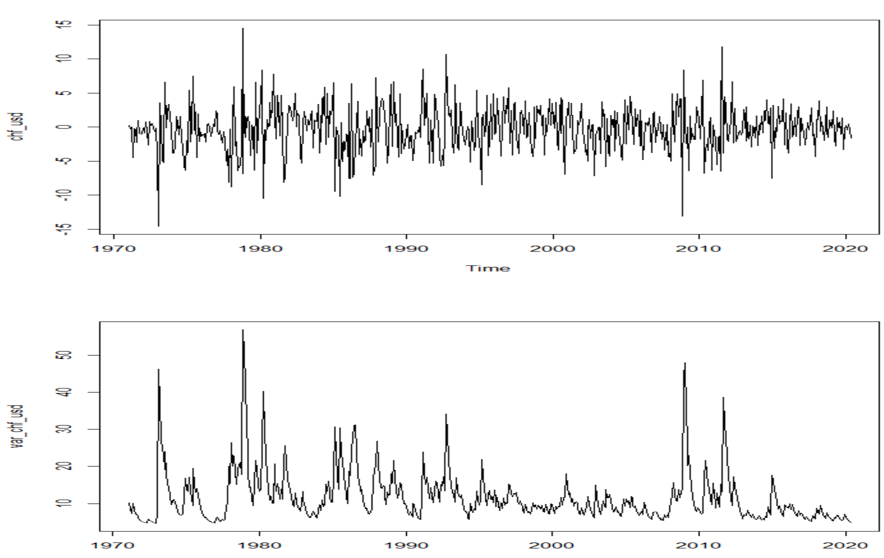
$\sigma_t^2 =$ **0.00012** + **0.19003** ε_{t-1}^2 + **0.71007** σ_{t-1}^2.
(0.00096)* (0.050)* (0.040)*

Unconditional σ^2 = **0.00012** /(1- **0.19003** - **0.71007**) = 0.001201201

Log likelihood: 1745.197

Note: $\alpha_1 + \beta_1 =$ **.90** < 1. (Persistent.)

ARCH Estimation: MLE – Example (in R)



ARCH Estimation: MLE – Example (in R)

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:

$r_t = [\log(P_t) - \log(P_{t-1})] = a_0 + a_1 r_{t-1} + \varepsilon_t, \qquad \varepsilon_t \mid I_{t-1} \sim N(0, \sigma_t^2)$

$\sigma_t^2 = \omega + \alpha_1\, \varepsilon_{t-1}^2 + \beta_1\, \sigma_{t-1}^2$

The estimated model for s_t is given by:

$r_t = 0.338 + 0.278\, r_{t-1},$
(.08)* (0.025)*

$\sigma_t^2 = 0.756 + 0.126\, \varepsilon_{t-1}^2 + 0.826\, \sigma_{t-1}^2.$
(0.151)* (0.017)* (0.021)*

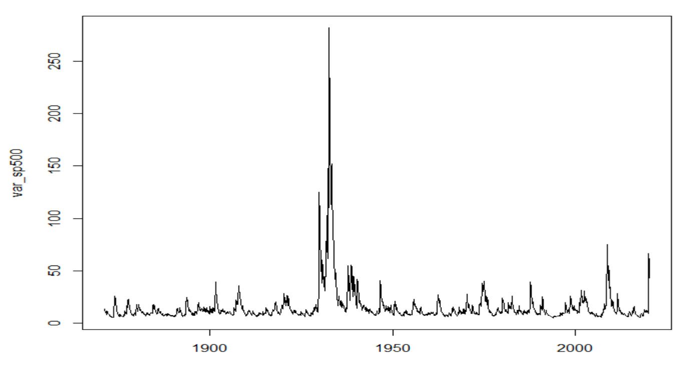
Unconditional $\sigma^2 = 0.756 / (1 - 0.126 - 0.826) = 15.4630$

Log likelihood: 4795.08

Note: $\alpha_1 + \beta_1 = .952 < 1.$ (Very persistent.)

ARCH Estimation: MLE – Example (in R)

Example 2: Below, 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:

$$E_t[\sigma_{t+j}^2] = \sigma_t^2 + j\omega$$

GARCH: Forecasting and Persistence

Example 1: We want to forecast next month (September 2020) variance for CHF/USD changes. Recall we estimated σ_t^2:

$$\sigma_t^2 = 0.00012 + 0.19003\, \varepsilon_{t-1}^2 + 0.71007\, \sigma_{t-1}^2.$$

getting $\sigma_{2020:9}^2 = 0.003672220$ $(=\sigma_{2020:9} = \text{sqrt}(0.00367) = 6.1\%)$

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, $(\alpha_1 + \beta_1) = 0.190 + 0.710 = 0.900$

$$E_{2020:9}[\sigma_{2020:10}^2] = 0.00012 + 0.00367 * (0.9) = 0.003423$$

We also forecast $\sigma_{2020:12}^2$

$$E_{2020:9}[\sigma_{2020:12}^2] = 0.00012 * \{1 + (0.9) + (0.9)^2\} + 0.00367 * (0.9)^3$$
$$= 0.00300063$$

GARCH: Forecasting and Persistence

Example 1 (continuation):
We forecast volatility for March 2021:

$$E_{2020:6}[\sigma^2_{2021:03}] = \mathbf{0.00012} * \{1 + (\mathbf{0.9}) + (\mathbf{0.9})^2 + \ldots + (\mathbf{0.9})^5\} +$$

$$+ \mathbf{0.00367} * (\mathbf{0.9})^6 = 0.002512659$$

<u>Remark</u>: We observe that as the forecast horizon increases (j ⟶ ∞), the forecast reverts to the unconditional variance:

$$\omega/(1 - \alpha_1 - \beta_1) = \mathbf{0.00012}/(1 - \mathbf{0.9}) = 0.0012$$

$$\Rightarrow \sigma = \text{sqrt}(0.0012) = 0.0346$$ (3.46% ≈ close to sample SD = 3.36%)

GARCH: Forecasting and Persistence

Example 2: On August 2020, we forecast the December's variance for the S&P500 changes. Recall we estimated σ_t^2:

$$\sigma_t^2 = \mathbf{0.756} + 0.125\, \varepsilon_{t-1}^2 + 0.826\, \sigma_{t-1}^2.$$

getting $\sigma^2_{2020:8} = \mathbf{43.037841}$

We based the $\sigma^2_{2020:12}$ forecast on:

$$E_t[\sigma^2_{t+j}] = \omega * [\textstyle\sum_{i=0}^{j-1}(\alpha_1 + \beta_1)^i] + \sigma_t^2\, (\alpha_1 + \beta_1)^j$$

Then, since $(\alpha_1 + \beta_1) = \mathbf{0.952}$

$$E_{2020:8}[\sigma^2_{2020:12}] = \mathbf{0.756} * \{1 + (\mathbf{0.952}) + (\mathbf{0.952})^2 + (\mathbf{0.952})^3\} +$$

$$+ \mathbf{43.037841} * (\mathbf{0.952})^4 = 38.02797$$

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**

GARCH: Forecasting – Application to VaR

Example (continuation):
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**

Remark: 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)
```

GARCH: Rugarch Package

Example (continuation): > 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

GARCH: Rugarch Package

Example (continuation): > ar1_garch11

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

GARCH: Rugarch Package

Example (continuation): > ar1_garch11

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.

d.o.f=2

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!

IGARCH

• 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: $y_t = \boldsymbol{X}_t\boldsymbol{\gamma} + \delta\,\sigma_t^2 + \varepsilon_t,\qquad \varepsilon_t \sim \text{N}(0, \sigma_t^2)$
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 $I_{t-i} = 1$ if $\varepsilon_{t-i} < 0$;
$= 0$ otherwise.

• Using the indicator variable I_{t-i}, this model captures sign (asymmetric) effects in volatility: Negative news ($\varepsilon_{t-i} < 0$) 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 $\varepsilon_{t-i} < 0$;
$= 0$ otherwise.

GARCH: Variations – Asymmetric GJR

• The GARCH(1,1) version:

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \gamma_1 \varepsilon_{t-1}^2 I_{t-1} + \beta_1 \sigma_{t-1}^2$$

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).

<u>Note</u>: The variance depends on both the size and the sign of the variance which helps to capture leverage type (asymmetric) effects.

GARCH: Variations – TARCH

• Threshold ARCH (TARCH) – Rabemananjara & Zakoian (1993)

Large events –i.e., large errors- have a different effect from small events. We use 2 indicator variables, $I(\varepsilon_{t-i} > \kappa)$ & $I(\varepsilon_{t-i} < \kappa)$: one for "large events," $(\varepsilon_{t-i} > \kappa)$, & one for "small events," $(\varepsilon_{t-i} < \kappa)$:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \{\alpha_i^+ I(\varepsilon_{t-i} > \kappa) + \alpha_i^- I(\varepsilon_{t-i} < \kappa)\} \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j {\sigma^2}_{t-j}$$

There are two variances:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i^+ \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j {\sigma^2}_{t-j}, \qquad \text{if } (\varepsilon_{t-i} > \kappa)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i^- \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j {\sigma^2}_{t-j}, \qquad \text{if } (\varepsilon_{t-i} < \kappa)$$

• We can modify the model in many ways. For example, we can allow for the asymmetric effects of negative news.

ARCH Estimation: MLE – Regularity Conditions

Technical Note: The appeal of MLE is the optimal properties of the resulting estimators under ideal conditions. However, this 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.

ARCH Estimation: MLE – Remarks

• 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 *t* likelihood function 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 $\upsilon \to \infty$, 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:
$\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$. 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:
$LM = (T - q)\, R_{e2}^2 \xrightarrow{d} \chi_q^2$.

ARCH: Testing

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
 }
```

ARCH: Testing

Example (continuation):

```
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.

• Ignoring ARCH

- Suppose you suspect y_t has an AR structure:

$$y_t = \gamma_0 + \phi_1 y_{t-1} + \varepsilon_t, \qquad \varepsilon_t | I_{t-1} \sim N(0, \sigma^2_t).$$

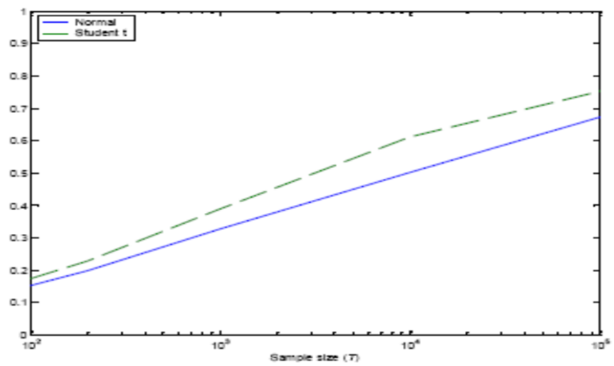
with ARCH structure in the error term, but you ignore it. You fit the AR(1) model using OLS.

• 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.

• White's (1980) SE help. NW SE help less.

ARCH: Testing – Ignoring ARCH

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).
Note: 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:

$$\boldsymbol{s^2} = \frac{1}{(T-1)}\sum_{i=1}^{\boldsymbol{T}}(\boldsymbol{x_i} - \overline{\boldsymbol{x}})^2$$

• Suppose we want to use 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 $\overline{\boldsymbol{x}}$ 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:

$$Variance_t = \sum_{i=1}^{k} r_{t,\boldsymbol{i}}^2$$

RV Models: Intuition

• 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 ($i \to 0$),

$RV_t \to$ 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 ∞.

• 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.

For example, the bid-ask bounce induces serial correlation in intra-day returns, which biases RV_t. The usual solutions is to 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.

RV Models: High Frequency – Practice

• Key choice: The sampling frequency (observations per period). Theory dictates using as many $r_{t,i}$ as possible. Then, use the highest frequency available, say millisecond to millisecond returns.

• But, in practice, market microstructure frictions limit the highest sampling frequency that may be used to reliably estimate RV_t.

• In intra-daily RV estimation, it is common to use 10' intervals. They have good properties. However, there are estimations with 1' intervals.

• Hansen and Lunde (2006) find that for highly 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.

RV Models: High Frequency – TAQ

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 j^{th} 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:

$$RVol_{t=30-min} = \sqrt{RV_{t=30-min}}$$

RV Models: High Frequency – TAQ

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	9:30:00	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

RV Models: High Frequency – TAQ

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 – TAQ

Example (continuation): We read SPY trade data for 2014:Jan.

```
> HF_da <- read.csv("c:/Financial Econometrics/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
```

RV Models: High Frequency – TAQ

Example (continuation): Using the SPY trade data, 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.

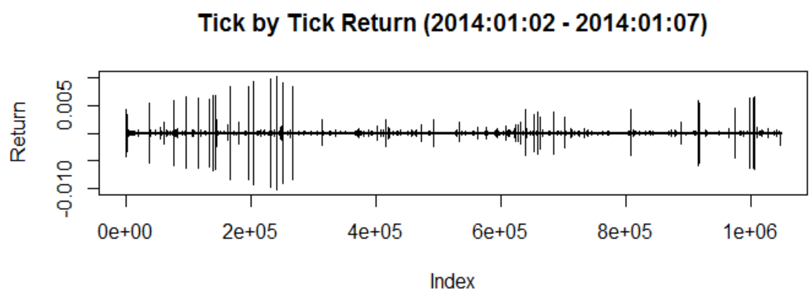
```
HF_da <- read.csv("http://www.bauer.uh.edu//rsusmel//4397//SPY_2014.csv",
head=TRUE, sep=",")
summary(HF_da)
pt <- as.POSIXct(paste(HF_da$DATE, HF_da$TIME), format="%Y%m%d %H:%M:%S")

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)
```

RV Models: High Frequency – TAQ

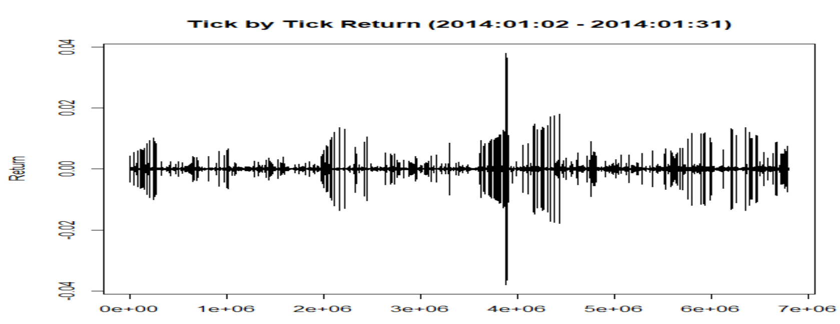
Example (continuation): We plot the tick-by-tick data.



Very noisy data, with lots of "jumps":
Mean tick by tick return: -3.7365e-09
Tick-by-tick SD: 6.3163e-05

RV Models: High Frequency – TAQ

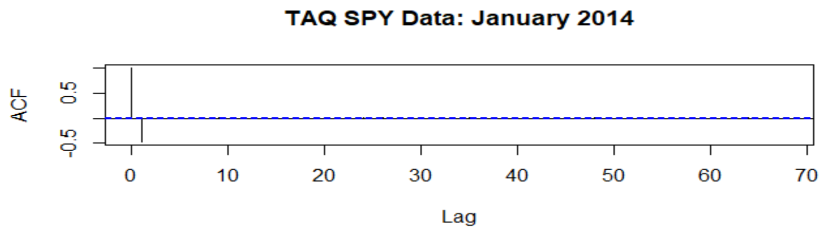
Example (continuation): For the whole month of January 2020:




```
> mean(spy_ret)
[1] -4.796933e-09
> sd(spy_ret)
[1] 7.804991e-05
```

RV Models: High Frequency – TAQ

Example (continuation): 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 1st-order autocorrelation: **-0.459.**

RV Models: High Frequency – TAQ

Example (continuation): 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).

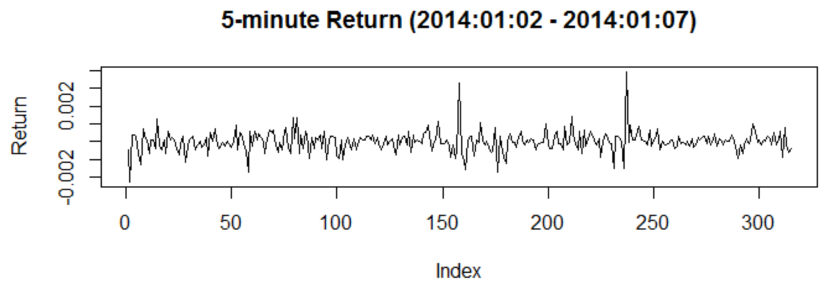
```
library(highfrequency)
spy_5 <- aggregateTrades(
hf_1,
on = "minutes",                # you can use also seconds, days, weeks, etc.
k = 5,                         # number of units in for "on"
marketOpen = "09:30:00",
marketClose = "16:00:00",
tz = "GMT"
)

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)")
```

RV Models: High Frequency – TAQ

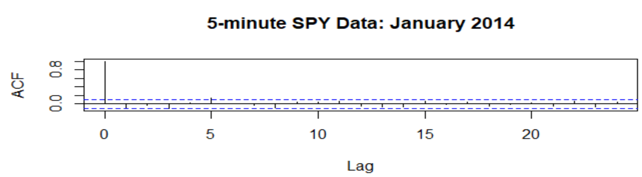
Example (continuation): We plot the 5-minute return data. Smoother, easier to read.



$RVol_{t=2014:01:02} = 0.0053344$
$RVol_{t=2014:01:03} = 0.0043888$
$RVol_{t=2014:01:04} = 0.0059836$
$RVol_{t=2014:01:05} = 0.0052772$

RV Models: High Frequency – TAQ

Example (continuation): We plot the autocorrelogram for the 5' TAQ SPY data:



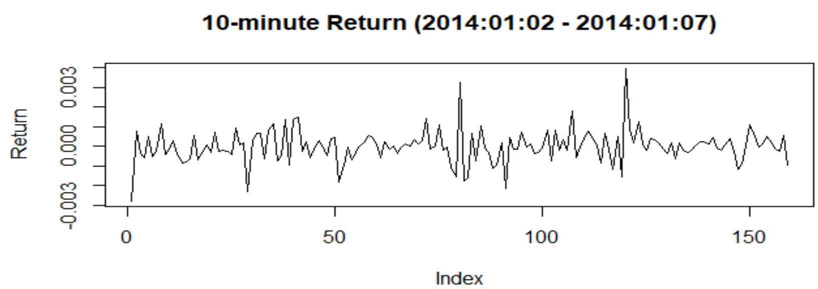

```
> acf_spy_5 <- acf(spy_5_ret, main = "5-minute SPY Data: January 2014")
> acf_spy_5
Autocorrelations of series 'spy_ret', by lag
```

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.

RV Models: High Frequency – TAQ

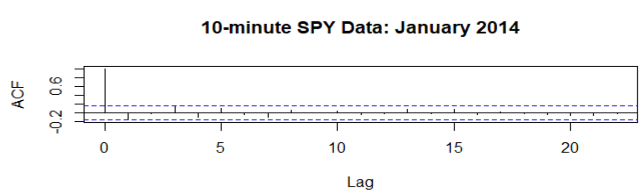
Example (continuation): We plot the 10-minute return data. Smoothing increases.



$RVol_{t=2014:01:02} = 0.005478294$
$RVol_{t=2014:01:03} = 0.004256046$
$RVol_{t=2014:01:04} = 0.006190508$
$RVol_{t=2014:01:05} = 0.005145601$

RV Models: High Frequency – TAQ

Example (continuation): We plot the autocorrelogram for the 10' TAQ SPY data:



Note: Now, none of the autocorrelations is significant. The 10-minute returns look independent.

RV Models: High Frequency – TAQ

• 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.

RV Models: R Script

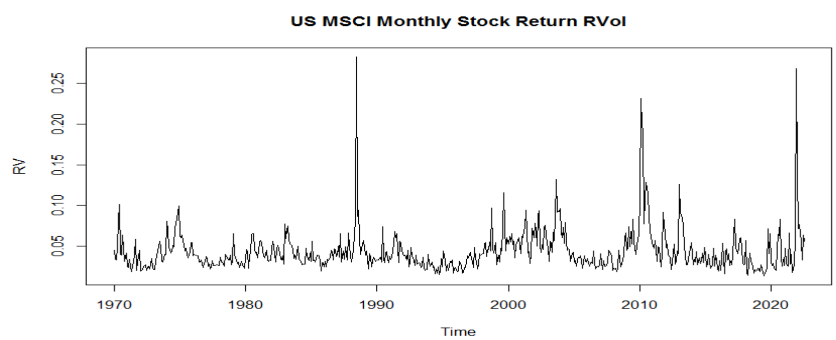
Example: R script to compute realized volatility

```
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
```

RV Models: Monthly RV From Daily Data

Example: Using daily 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)
```

RV Models: Monthly RV From Daily Data

Example (continuation):

Technical computing points:

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) 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.

RV Models: Monthly RV From Daily Data

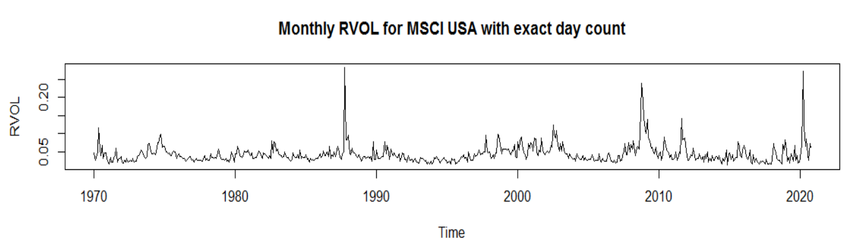
Example (continuation): Below, the while loop in R is modified to incorporate the vector K (c1) of exact trading days for each month.

```
MSCI_cd <-
read.table("https://www.bauer.uh.edu/rsusmel/4397/MSCI_d_count_days.txt",header=FALSE)
c1 <- MSCI_cd[,1]
n_c1 <- length(c1)
rvs=NULL                          #Initialize empty
t <- 1
tj <- 1
x_m = mean(x)
while (tj  <= n_c1) {
mj <- c1[tj]
xx <- x[t:(t+mj-1)] - x_m
s2 <- sum(xx^2)
t <- t + mj
tj <- tj + 1
rvs <- rbind(rvs,s2)
}
```

RV Models: Monthly RV From Daily Data

Example (continuation): Below, we plot the new series

```
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) very similar).

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

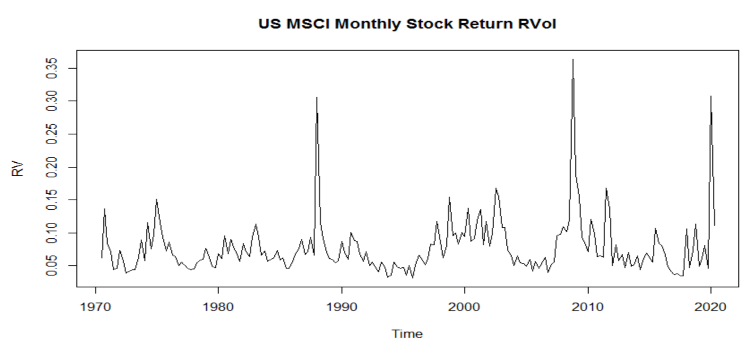
$$RV_{t=annual} = 260 * RV_{t=daily}$$

where 260 is the number of trading days in the year. The annualized RVOL is the squared root of RV_{annual}:

$$RVOL_{t=annual} = sqrt(260) * RVOL_{t=daily}$$

RV Models: Quarterly RV From Daily Data

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 monthly Rvol in the sample
[1] 0.07725361         ⇒ log approximation: sqrt(3) * 0.04326 = 0.07493 (close!)
> sd(rvol)            # standard deviation of monthly 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.

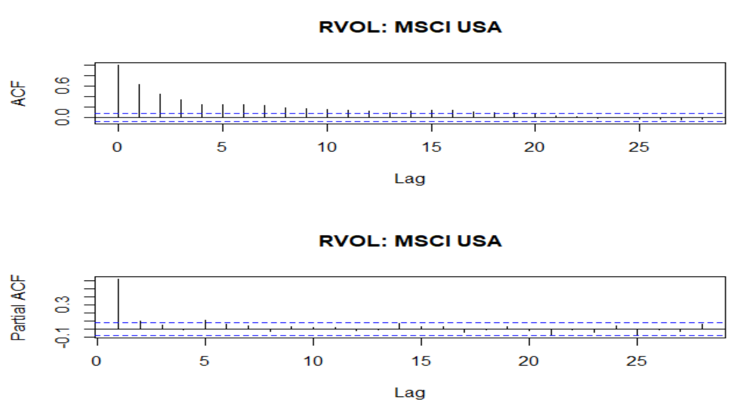
• RV is highly persistent. (Check with a LB test.)

• 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.



⇒ 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(1) 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
```

RV Models: Forecasting

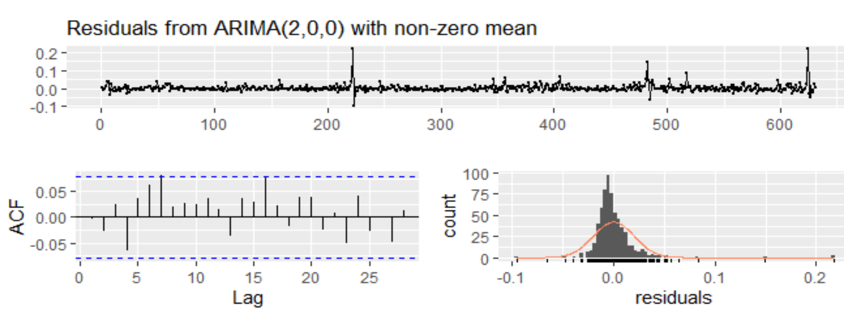
Example (continuation):

```
> 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
```



RV Models: Forecasting

Example (continuation):

fcast_rvol <- forecast(fit_rvol_ar2, h=12, level=.95) # h=number of step-ahead forecasts

> fcast_rvol

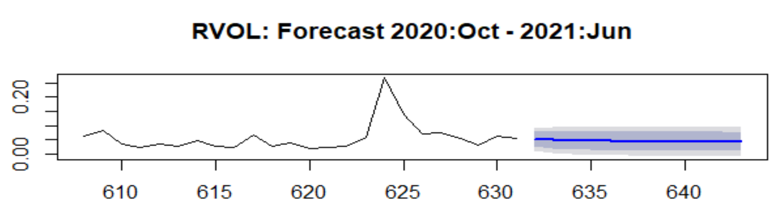
> fcast_rvol

	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

RV Models: Forecasting

Example (continuation):

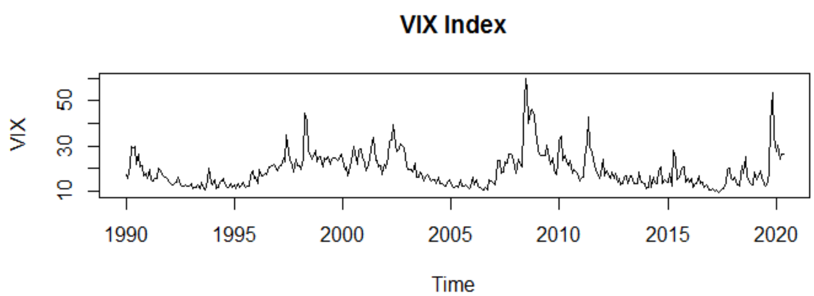
fcast_rvol <- forecast(fit_rvol_ar2, h=12, level=.95) # h=number of step-ahead forecasts



<u>Note</u>: 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 thr 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.



RV Models: Forecasting – Using VIX

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 <- length(rvol)                         # End of sample for RVol (2020:Oct)
rvol_90 <- as.numeric(rvol_ts[245:T])     # RVol starting in 1990:May
rvol_0 <- rvol_90[-1]                     # remove first observation (RV_t+1)

lm_rvol_f <- lm(rvol_0 ~ VIX)
summary(lm_rvol_f)
```

RV Models: Forecasting – Using VIX

Example (continuation):

```
lm_rvol_f <- lm(rvol_0 ~ VIX)
> summary(lm_rvol_f)

Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  0.0259776  0.0035343   7.350 1.32e-12 ***
VIX          0.0009262  0.0001690   5.481 7.94e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02481 on 363 degrees of freedom
Multiple R-squared:  0.07643,   Adjusted R-squared:  0.07388
F-statistic: 30.04 on 1 and 363 DF,  p-value: 7.94e-08
```

Note: In sample, a strong positive predictive relation.

RV Models: Forecasting – Using VIX

Example (continuation): We also check the contemporaneous relation between RVol and VIX.

```
> lm_rvol <- lm(rvol_90[-length(rvol_90)] ~ VIX)
> summary(lm_rvol)

             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  0.0301794  0.0035950   8.395 1.06e-15 ***
VIX          0.0007095  0.0001719   4.128 4.55e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02523 on 363 degrees of freedom
Multiple R-squared:  0.04483,   Adjusted R-squared:  0.0422
F-statistic: 17.04 on 1 and 363 DF,  p-value: 4.551e-05
```

Note: A weaker relation, as expected: IV embeds future expectations.

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.

RV Models: VRP – Definition

• 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*]:

$$VRP_t = IV_t - RV_t.$$

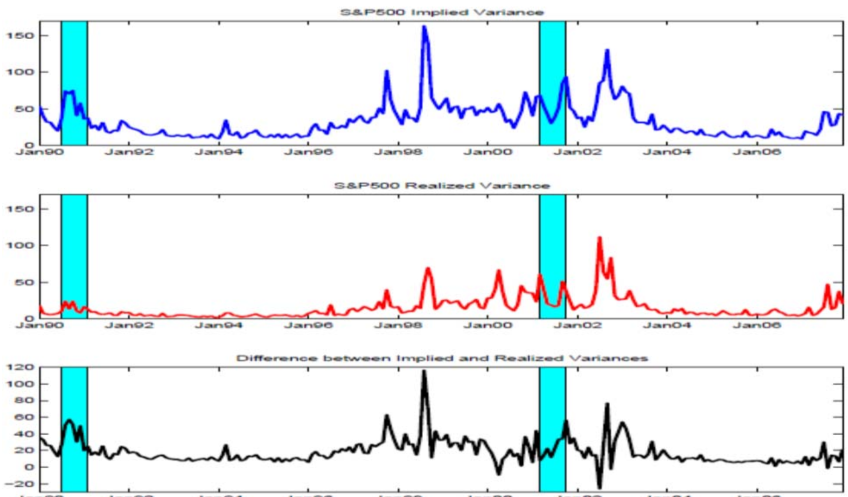
It is an ad-hoc definition, we could have defined VRP_t based on the expectation at time t for RV_{t+1}, in this case $E_t[RV_{t+1}]$. The one-step-ahead forecast can be obtained using an ARMA process for RV_t.

In practice, using $E_t[RV_{t+1}]$ 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.

RV Models: VRP – 1990-2008

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.



RV Models: VRP – Predicting Stock Returns

• 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:

$$r_{t+h} - r_{f,t+h} = [\log(P_t) - \log(P_{t-1})] = \mu + \delta\, VRP_t + \varepsilon_t$$

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

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 j^{th} 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, υ_t:

$$\sigma_t = \omega + \beta\sigma_{t-1} + \eta_t; \qquad \eta_t \sim N(0, \sigma_\eta^2)$$

Or using logs:

$$\log\sigma_t = \omega + \beta\log\sigma_{t-1} + \upsilon_t; \qquad \upsilon_t \sim N(0, \sigma_\upsilon^2)$$

• The difference with ARCH models: The shocks that govern the volatility are not necessarily ε_t's.

• Usually, the standard model centers log volatility around ω:

$$\log\sigma_t = \omega + \beta(\log\sigma_{t-1} - \omega) + \upsilon_t$$

Then,

$$E[\log(\sigma_t)] = \omega$$

$$Var[\log(\sigma_t)] = \varkappa^2 = \sigma_\upsilon^2/(1 - \beta^2).$$

⇒ Unconditional distribution: $\log(\sigma_t) \sim N(\omega, \varkappa^2)$

Stochastic volatility (SV/SVOL) models

• 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: $\varphi = (\omega, \beta, \sigma_v)$.

• Estimation: The modern approach uses Bayesian methods (MCMC), which are advanced for this class. Brooks discusses the estimation of SVOL.