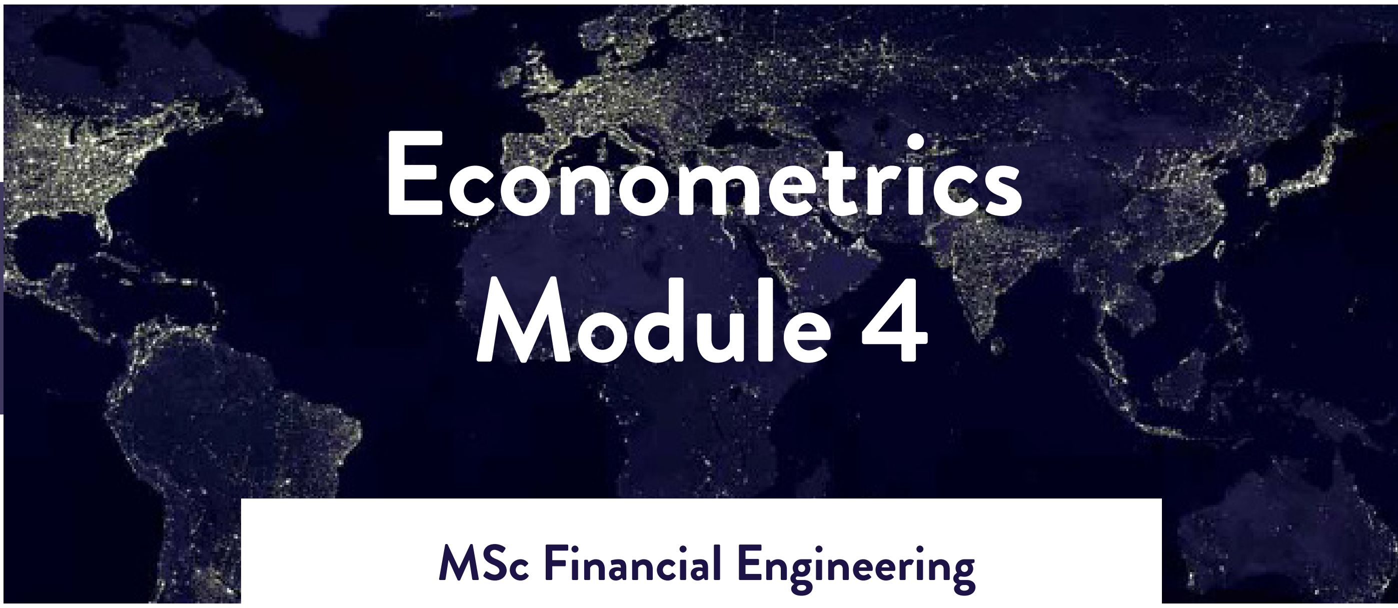


Econometrics Module 4

MSc Financial Engineering

Table of Contents

1. Brief

This document contains the core content for Module 4 of Econometrics, entitled Univariate Volatility Modeling. It consists of three lecture video lecture transcripts, five sets of supplementary notes, and a peer review question.

2. Course Context

Econometrics is the second course presented in the WorldQuant University (WQU) Master of Science in Financial Engineering (MScFE) program. In this course, we apply statistical techniques to the analysis of econometric data. After Module 1's introduction to the R statistical programming language, we then go on to build econometric models, including multiple linear regression models, time series models and stochastic volatility models, before the course concludes with sections on extreme value theory and risk management.

2.1 Course-level Learning Outcomes

Upon completion of the Econometrics course, you will be able to:

1. Write programs using the R language.
2. Use R packages to solve common statistical problems..
3. Formulate a generalized linear model and fit the model to data.
4. Understand the properties of the multivariate normal distribution.
5. Use graphic techniques to visualize multidimensional data.
6. Apply multivariate statistical techniques (PCA, factor analysis, etc.) to analyze multidimesional data.
7. Fit a time series model to data.
8. Fit discrete-time volatility models.
9. Understand and apply filtering techniquesto volatility modeling.
10. Understand the use of extreme value distributions in modeling extreme portfolio returns.
11. Define common risk measures like VAR and Expected Shortfall.
12. Define and use copulas in risk management

2.2 Module Breakdown

The Econometrics course consists of the following one-week modules:

1. Learning R and Stylized Facts of Financial Data
2. Generalized Linear Modules
3. Univariate Time Series Models
4. Univariate Volatility Modeling
5. Multivariate Time Series Analysis
6. Extreme Value Theory
7. Introduction to Risk Management

3. Module 4

Univariate Volatility Modeling

In this module, we turn to a central aspect of interest to the financial engineer — namely, risk. In econometric terms, we link this to the volatility of returns, which we characterize with the statistical concept of variance, or its square root, standard deviation.

3.1 Module-level Learning Outcomes

Upon completion of ths module, you will be able to:

1. Understand the fundamentals of ARCH and GARCH models.
2. Describe the parameter estimation of ARCH and GARCH models.
3. Define filtering and the Kalman filter.

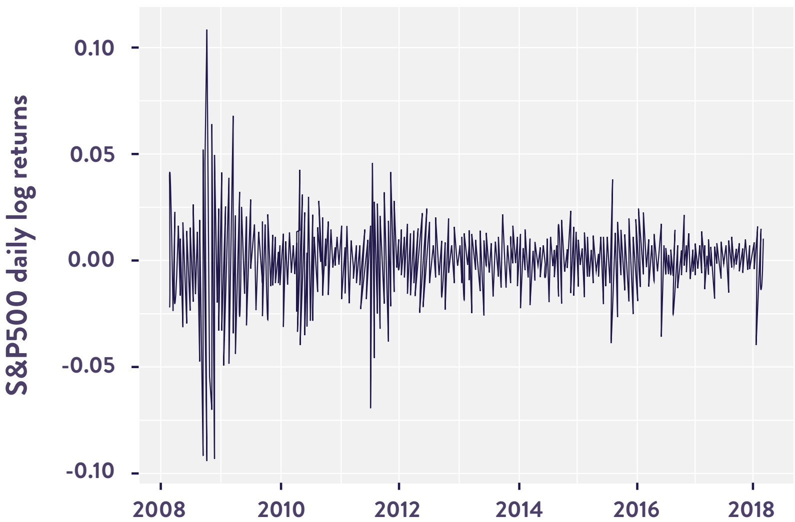
3.2 Transcripts and Notes

3.2.1 Notes: Introduction

In this module we turn to a central aspect of interest to the financial engineer — namely, risk. In econometric terms we link this to the **volatility** of returns, which we characterize with the statistical concept of **variance**, or its square root, **standard deviation**. Importantly, this volatility is not constant over time.

Almost all returns on stocks or stock market indices show periods of relative tranquility followed by periods of high volatility as we observed in the daily log returns on the S&P500 in Module 3:

Figure 1: Daily Log Returns of the S&P500 Index



Source: Calculations on data from fred.stlouisfed.org (2018).

During the Global Financial Crisis of 2008, the daily log returns on the S&P500 were far more volatile than in the rest of the sample. Similarly, there are periods in 2010, 2011, 2015/16 and possibly in 2018 where the volatility was obviously larger than in the rest of the sample. This means that this time series does not have constant variance. Hence, according to our definitions in Module 3, the series does not qualify as weakly stationary. In terms of our definitions in Module 2: Generalized Linear Models, this means that the series is not homoscedastic, but heteroscedastic. If the mean is constant, as it seems to be in the S&P500 log returns, the simple linear method will still be consistent, but not efficient.

In this module, we will build models that simultaneously model the conditional expectation and conditional variance of the data-generating process.

The simplest way to build intuition on this is that we will apply the auto-regressive properties of the conditional expectation that we developed in Module 3 to the conditional variance. We will call the simplest version of these models Auto-Regressive Conditional Heteroscedastic (ARCH) models. That is, while the variance in the population (the unconditional variance) is constant, we allow the data-generating process to have the following property: that, periods of high variance relative to the unconditional variance tend to be followed by periods of high variance, and periods with low variance tend to be followed by periods of similarly low variance in comparison to the unconditional variance. Thus, we allow the conditional variance to be autocorrelated in the same sense as we allowed the conditional expectation to be autocorrelated in Module 3.

This means we allow there to be a mean-reverting, but autocorrelated process that governs the realization of variance of the process. Thus, considering the conditional expectation and conditional variance of a process jointly, we will retain the classification of stationarity: while the variance of the process in a particular window might not be constant, the process that generates the changes in variance in the population is itself stationary. It is constant over time.

There are many ways of formalizing the nature of this conditional heteroscedasticity. In this lecture we will explore three different characterizations: the simplest is the ARCH model, where we only allow an auto-regressive process in the variance. Next, we will consider generalized ARCH (GARCH) models where we allow the variance to have both auto-regressive (AR) and moving average (MA) properties. Thus, the ARCH model extends the AR models of Module 3 to the variance, where GARCH models extend ARMA models to the variance. We will also consider a different approach, called Stochastic Volatility Modeling, where we take an agnostic stance on the process that drives the heteroscedasticity of the process.

As you might expect, the theoretical underpinnings of these models are more complex, which means there are many more formal probabilistic and statistical issues related to what is permissible, or not, for our empirical models to be reliable. These are too deep to cover here, so you should familiarize yourself with the necessary theoretical features once you understand the core intuition of the methods we will cover in this module. As usual, the packages developed to operationalize these models in R recognize these issues and have built-in tests to make sure that you model these issues appropriately.

An accessible treatment of applications and the generic theoretical issues is available in Enders (2014), Chapter 3. A condensed treatment of the main concerns for the financial engineer is available in Tsay (2010), Chapter 3, and a deep treatment of the theoretical issues is available in Hamilton (1992).

With regard to R packages: in this module we will use the following packages (and their dependencies which are automatically installed with each): `tidyverse, readxl, stats, forecast and fGarch.`

The Simplest ARCH model

The simplest case of an auto-regressive conditional heteroscedastic model is set up as follows: We assume that the conditional expectation of the process is modeled as an ARMA process, but now with errors that are not white noise (i.e. they do not have constant variance). For expositional simplicity, we assume that an AR(1) process is sufficient to capture the time series properties of the mean of the process (nothing would change if we assumed a more complicated time series process for the conditional mean, but all the important aspects of this choice remain the same as in the previous module):

$$y_t = a_0 + a_1 y_{t-1} + u_t.$$

Now, however, we assume that the error process u_t has the following structure.

Firstly, the expectation (conditional and unconditional[1]) of the error remains zero (so only its variance deviates from white noise):

$$\mathbb{E}(u_t) = \mathbb{E}_{t-1}(u_t) = 0.$$

Secondly, for the model of the conditional mean to be consistent, we still require the level of the errors to be uncorrelated over time:

$$\mathbb{E}(u_t u_{t-s}) = 0 \; \forall \; s > 0.$$

We allow the conditional variance $\mathbb{E}_t({u_t}^2) = {\sigma_t}^2$ to be autocorrelated. The subscript t now denotes that the time t conditional variance might depend on the moment of observation t. Unfortunately, this variance is not observable, so we have to build a model that we can use to estimate its time path. The various assumptions we make on this model lead to the different types of ARCH models. A large number of different models have been developed for different applications.

[1]The notation $\mathbb{E}_{t-1}(z_t)$ means $\mathbb{E}(z_t|F_{t-1})$ where F_{t-1} refers to all information known up to period t.

The simplest version we will call an ARCH(1), where, as in Module 3, we specify the lag order of the model of interest. We thus assume that the following process generates the innovations and their conditional variance (this specification is due to Engle (1982)):

$$u_t = \varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2}$$

Where ε_t is a white noise as defined in Module 3 that is independent of u_t with unit variance $\sigma_\varepsilon{}^2 = 1$.

Note what this means for the conditional and unconditional variance. The period t - 1 conditional variance of u_t is:

$$\mathbb{E}_{t-1}(u_t^2) = \mathbb{E}_{t-1}\left(\varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2}\right)^2$$

$$= \sigma_\varepsilon^2(\omega + \alpha_1 \mathbb{E}_{t-1}(u_{t-1}^2))$$

$$= \omega + \alpha_1 u_{t-1}^2.$$

This is an $AR(1)$ process in $u_t{}^2$, i.e. it is a stationary, mean reverting process (if $\alpha_1 < 1$). The unconditional variance is constant:

$$\mathbb{E}(u_t^2) = \mathbb{E}\left(\varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2}\right)^2$$

$$= \omega + \alpha_1 \mathbb{E}(u_{t-1}^2)$$

$$= \frac{\omega}{1 - \alpha_1}$$

where the last step used the definition of a stationarity: $\mathbb{E}(u_t^2) = \mathbb{E}(u_{t-1}^2)$. Note, for this to be positive, we will require $\alpha_1 < 1$. In practice, stronger restrictions are necessary for large sample results to hold. You can read up on this in Tsay (2010).

You should check that this assumed process indeed satisfies all the requirements of an unconditional white noise process: $\mathbb{E}(u_t) = \mathbb{E}_{t-1}(u_t) = 0$ and $\mathbb{E}(u_t u_{t-s}) = 0 \; \forall \; s > 0$.

Extensions to the ARCH model

We can easily extend this to an ARCH(p) process if we assume the conditional variance follows an AR(p) process:

$$u_t = \varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2 + \ldots + \alpha_p u_{t-p}^2}$$

or a GARCH(p,q) process if we assume that the conditional variance follows an ARMA(p,q) process (Bollerslev, 1986):

$$u_t = \varepsilon_t \sqrt{h_t}$$

$$h_t = \omega + \sum_{i=1}^{p} \alpha_i u_{t-i}^2 + \sum_{j=1}^{q} \beta_i h_{t-j}.$$

The conditions for stationarity, and for large sample theory to hold, are available in the texts mentioned in the introduction and in the original papers cited.

Limitations

There are a number of limitations of the ARCH model, some of which are addressed by the extensions to the ARCH model we consider (see Tsay (2010)).

- The model assumes symmetric effects: whether a shock to the process is positive or negative, the impact on the conditional variance is the same. This is not representative of financial series – negative shocks tend to be more destabilizing to the variability of the returns on a financial asset than positive shocks. Consider what happened in the Global Financial Crisis of 2008. The EGARCH model mentioned below is a generalization built to deal with this weakness.

- ARCH models allowing only AR properties in the conditional variance process tend to predict slow mean reversion in the conditional variance. In practice, there can be sharp spikes in volatility that vanish quickly. Other models, like regime shifting models or the stochastic volatility model we briefly touched on in the last section, attempt to address this weakness.

- The ARCH model struggles to deal with excess kurtosis (relative to normal distributions) due to fundamental mathematical constraints that must hold for the derivations to be well defined. This can be rectified somewhat by using distributions other than the normal ones (with fatter tails and more kurtosis) as the assumed fundamental shock distribution. We will see this problem at the end of the practical application to follow.

- The ARCH model allows only the shock to the conditional expectation to feed to persistence in the conditional variance, but does not allow changes in the conditional variance to affect the conditional expectation. In practice, we may expect the expected return on an asset to be depressed in periods of high volatility. The GARCH-M process mentioned in the last section of this module is built to allow this feedback.

- The model imposes a specific, mechanical process that generates the conditional variance time path. It does not explain why the conditional variance follows this process. For the skeptical financial engineer, this is not too troublesome if the interest is in near-future projections rather than fundamental analysis. For the long run-investor, however, this is not desirable. This same weakness is present in all the univariate volatility models in this module, which is why we will move to multivariate models in the next module that have more hope of capturing the fundamental causes of periods of excess volatility. In the bigger scheme of things, this is a fundamental issue of **economic causality**, which requires a good economic model (theoretical and empirical) to overcome with any confidence.

3.2.3 Transcript: Practical Application (Part 1)

How can you tell when it is necessary to use an ARCH or GARCH model? This video covers the steps necessary to make this call formally by applying the model-building process to the daily log returns on the S&P500.

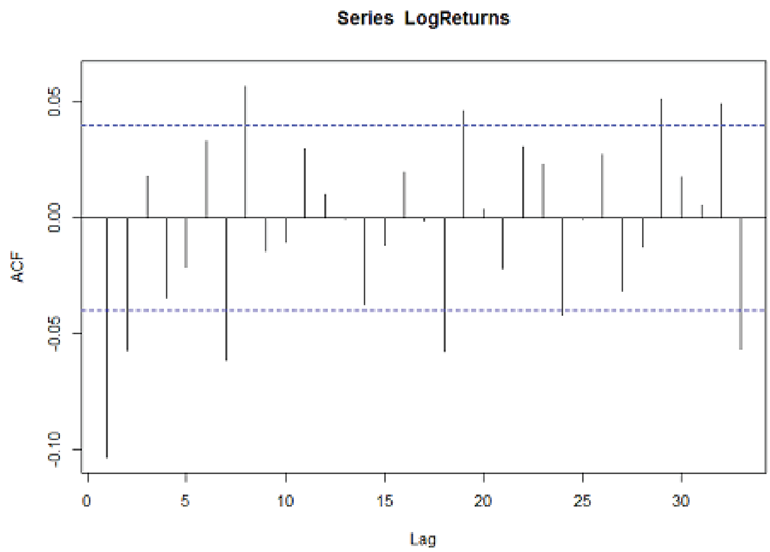
For this illustration, we will illustrate the very basics of the estimation approach by omitting many aspects of importance that, in practice, must be considered before you perform any of the estimations. In the next section we will examine some of these considerations and cover where to learn more about them for full-scale, practical applications.

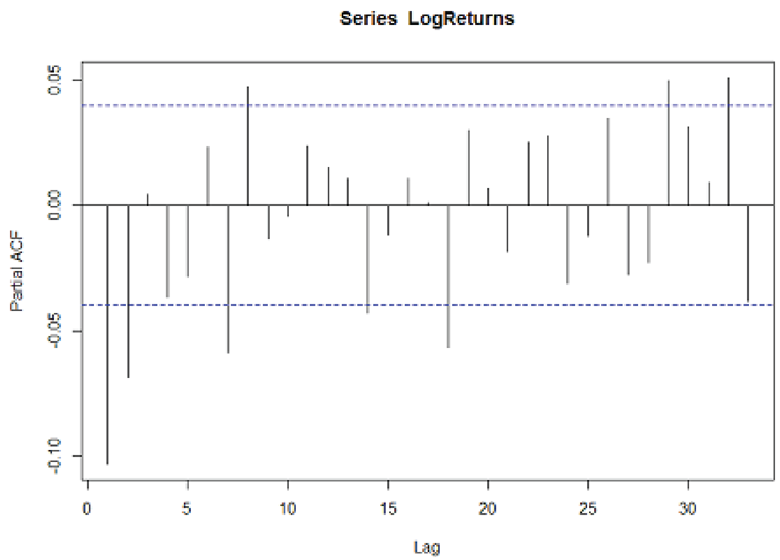
Step 1: Fit an adequate model of the level of the variable

Because the errors must be white noise, we must first ensure that we model all the time series information in the level of the variable by fitting an ARMA(p,q) model of sufficiently high order.

We begin by investigating the autocorrelation and partial autocorrelation functions.

```
# import data
library(tidyverse)
library(readxl)
library(stats)
library(forecast)
library(fGarch)
SnP500 <- read_excel("C:/<your path>/<your file name>.xlsx",
                     col_types = c("date", "numeric", "numeric"))
# store log returns in a variable:
LogReturns = SnP500$LogReturns
Acf(LogReturns)
Pacf(LogReturns)
```





Both the ACF and PACF show the typical patterns of a stationary process. You should confirm this with formal statistical tests.

Next, we fit an ARMA process to account for the auto-correlations.

```
# find order of AR process automatically via the AIC
ar_model <- ar(LogReturns)

# fit and evaluate the automatically chosen AR model:
ar_model_fitted <- arima(LogReturns,order = c(ar_model$order,0,0))
ar_model_fitted

# Output:
Call:
arima(x = LogReturns, order = c(ar_model$order, 0, 0))

Coefficients:
          ar1      ar2      ar3      ar4      ar5     ar6      ar7     ar8  intercept
      -0.1060  -0.0718  -0.0023  -0.0361  -0.0298  0.0202  -0.0535  0.0478      3e-04
s.e.   0.0203   0.0203   0.0204   0.0204   0.0204  0.0204   0.0204  0.0203      2e-04
sigma^2 estimated as 0.0001617:  log likelihood = 7170.65,  aic = -14321.3

# test that errors are white noise
Box.test(ar_model_fitted$residuals, lag = 24, type = 'Ljung')

# Output:
      Box-Ljung test

data:  ar_model_fitted$residuals
X-squared = 26.084, df = 24, p-value = 0.3489
```

The Box-Ljung test we employ here is an improvement on the Box-Pierce test. Numerical studies showed that the Box-Pierce statistic performs poorly in small samples, which led to the development of the Box-Ljung test. The Box-Ljung test, with degrees of freedom k, is a test that all $k$autocorrelations in a time series (of length n) are zero. The empirical statistic is a function of the sum of autocorrelations (autocorrelation of order i denoted as $\hat{\rho}_i$) up to order k, and is given by:

$$Q(k) = n(n+2)\sum_{i=1}^{k}\frac{\hat{\rho}_i^2}{n-i}.$$

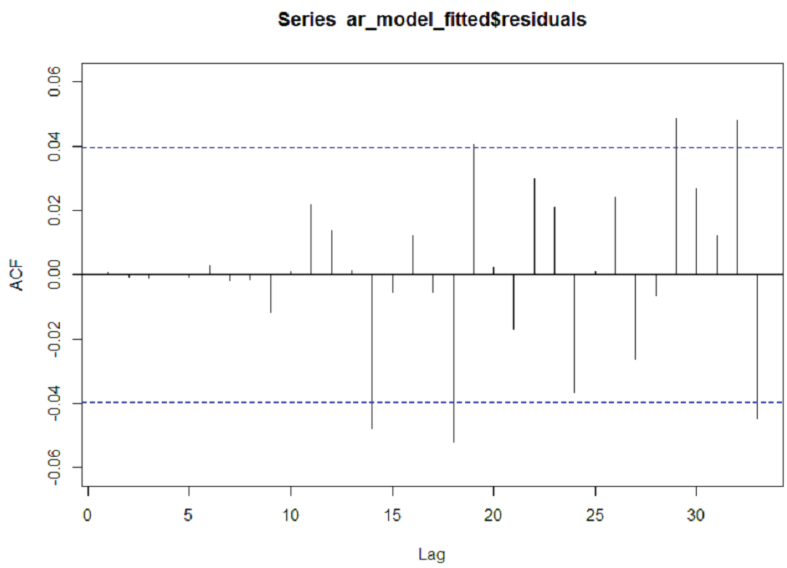
This quantity is chi-squared distributed with k degrees of freedom (often denoted as “df” in code). Thus “lag” and “df” parameters in the R code are different ways of referring to the same idea.

As with any statistical test, if the p-value is larger than the required degree of significance, we do not reject the hypothesis. That is, in some sense, we can argue that the hypothesis has no strong evidence against it.

In this case, the p-value is very large, so we clearly cannot reject the hypothesis that the residuals have no autocorrelation up to 24 lags out. Thus, we have evidence in favor of an encompassing model. The AR(8) model chosen with the AIC is, however, not parsimonious – there are several insignificant coefficients: AR coefficients 3, 4, 5 and 6. We should therefore be concerned that these coefficients are not accurately estimated, which will yield poor forecasting performance.

If we view the ACF of the residuals of this process, (below) we see that while the first view auto-correlations are negligible, there are some small but significant and alternating auto-correlations further out in the lag structure. This is typical of a process with some moving average properties.

```
Acf(ar_model_fitted$residuals)
```



After some experimentation, an ARMA(5,2) model seems a better one for capturing the time series properties of this process parsimoniously:

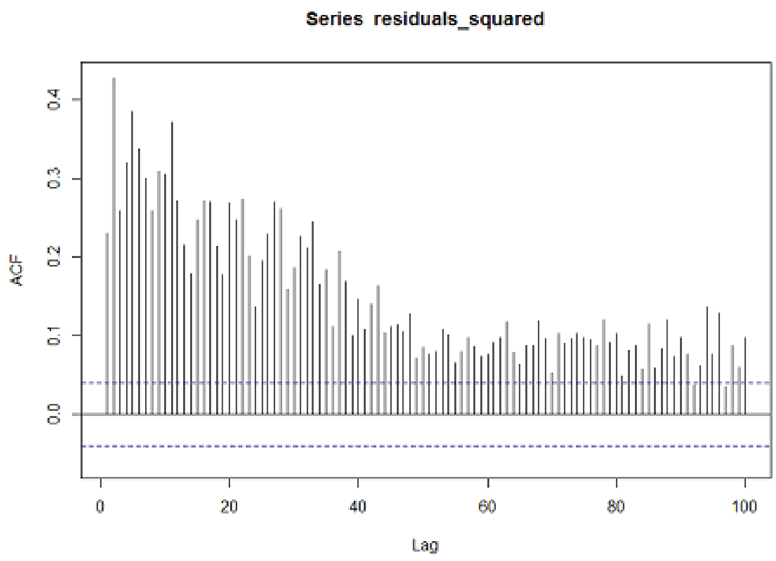
```
# fit better ARMA model:
arma_model_fitted <- arima(LogReturns,order = c(5,0,2))
arma_model_fitted
Call:
arima(x = LogReturns, order = c(5, 0, 2))
Coefficients:
          ar1      ar2      ar3      ar4      ar5     ma1     ma2  intercept
      -1.3924  -0.7613  -0.1516  -0.0744  -0.0631  1.2878  0.5548      3e-04
s.e.   0.1571   0.1483   0.0440   0.0357   0.0233  0.1567  0.1305      2e-04
sigma^2 estimated as 0.0001618:  log likelihood = 7169.89,  aic = -14321.78
```

This model now has only significant coefficients and tests as having white noise errors up to 24 lags, although the ACF still has far lag significant autocorrelation that the simple ARMA approach cannot capture parsimoniously. We will evaluate if the ARCH model helps control for this.

Step 2: Use the residuals of the ARMA model to test for ARCH effects

Next, we check if there is any autocorrelation in the squares of the residuals. This serves as a diagnostic – if we find strong time series properties in the squared residuals, this implies that the conditional variance of the original process is not constant:

```
# test for autocorrelation in the squared residuals
residuals_squared = ar_model_fitted$residuals^2
Acf(residuals_squared, lag.max = 100)
Box.test(residuals_squared,lag=24, type = 'Ljung')

     Box-Ljung test
data:  residuals_squared
X-squared = 2963.4, df = 24,
p-value < 2.2e-16
```



From both the ACF and the Box-Ljung test of the squared residuals we conclude that there is strong autocorrelation in the squared residuals. This means that the ARMA model does not fully capture the time series properties of the daily log returns of the S&P500. This means we have to fit an ARCH or GARCH model. Note as well that the autocorrelations of the squared residuals are very slow to fade. This means that the distribution has fatter tails than a normal distribution and may require a different type of model which we will discuss in the extensions later in this module.

In the next video we will fit and evaluate an ARCH model.

3.2.4 Transcript: Practical Application (Part 2)

In this video we will fit and evaluate an ARCH model and fit and evaluate a GARCH model.

Step 3: Fit and evaluate an ARCH model

We now augment the ARMA(5,2) model for the conditional expectation with an ARCH(2) model for the conditional variance, hence we fit an ARMA(6,2)-ARCH(2) model using the `fGarch` package in R. We estimate an ARMA(6,2) in this step as the ARMA(5,2) had a convergence problem in this estimation - we will turn to this later, but given the complicated nature of performing the estimation in practice, sometimes models do not converge and thus do not give interpretable output.

```
> ARCH_model <- garchFit(formula = ~ arma(6,2) + garch(2,0), data =
LogReturns, trace = F)
> summary(ARCH_model)
Call:
 garchFit(formula = ~arma(6, 2) + garch(2, 0), data = LogReturns, trace = F)
Conditional Distribution:
 norm
         Estimate  Std. Error  t value Pr(>|t|)
mu      2.331e-03   6.041e-04    3.858 0.000114 ***
ar1    -1.000e+00   2.397e-01   -4.171 3.03e-05 ***
ar2    -3.523e-01   2.800e-01   -1.258 0.208322
ar3    -9.591e-02   3.895e-02   -2.462 0.013813 *
ar4    -1.431e-01   3.240e-02   -4.417 9.99e-06 ***
ar5    -1.311e-01   3.504e-02   -3.743 0.000182 ***
ar6    -5.582e-02   2.765e-02   -2.019 0.043509 *
ma1     9.066e-01   2.393e-01    3.789 0.000151 ***
ma2     2.520e-01   2.575e-01    0.979 0.327801
omega   4.606e-05   2.911e-06   15.825  < 2e-16 ***
alpha1  3.293e-01   3.816e-02    8.629  < 2e-16 ***
alpha2  5.204e-01   5.082e-02   10.240  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Log Likelihood:
 7686.628    normalized:  3.158023
```

```
Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1097.079  0
 Shapiro-Wilk Test  R    W      0.9540607 0
 Ljung-Box Test     R    Q(10)  17.74626  0.05939766
 Ljung-Box Test     R    Q(15)  21.66579  0.1168714
 Ljung-Box Test     R    Q(20)  23.33152  0.2728111
 Ljung-Box Test     R^2  Q(10)  134.7288  0
 Ljung-Box Test     R^2  Q(15)  194.6373  0
 Ljung-Box Test     R^2  Q(20)  239.7193  0
 LM Arch Test       R    TR^2   189.3394  0
```

The parameter estimates in the table have the estimated mean (called `mu`), the AR and MA coefficients as in the ARMA model. The coefficient `omega` is (as in our notation) the constant in the conditional variance equation and would correspond to the constant variance if we had no ARCH effects. The coefficients `alpha 1` and `alpha2` (as in our notation)are the two lags of the conditional variance that we allow in the ARCH specification. They are both strongly significant, confirming our initial analysis that there are ARCH effects in the residuals. Is there evidence that we have statistically captured these ARCH effects sufficiently? We will use the Standardized Residual Tests to come to this conclusion.

The first two tests, the Jarque-Bera and Shapiro-Wilk tests, are tests of error normality. Error normality is clearly rejected. This is problematic, as the estimation algorithm uses the normal distribution to obtain the fit. If the errors are not normal, then the estimation is not going to be reliable. For now, we will ignore this and move on to the other tests.

The Ljung-Box Tests are the same as the ones we used before (for 10, 15 and 20 lags respectively). The first three tests test for autocorrelation in the level of the residuals. As in our ARMA model, these do not reject the hypothesis of no-autocorrelation, so this is in favor of our estimation being encompassing.

The next three test for autocorrelation in the squared-residuals, as we did in our initial analysis to determine whether there are ARCH effects probable. These are all rejected, meaning that the ARCH(2) specification is not sufficient to capture the time series properties of the conditional variance. Similarly, the Lagrange Multiplier (LM) test for ARCH effects also rejects this.

Thus, we know we must use a more general specification if we wish to have a good model of the conditional variance. Next, we will add a GARCH effect.

Step 4: Fit and evaluate a GARCH model

We now augment the ARMA(5,2) model for the conditional expectation with a GARCH(2,1) model for the conditional variance, hence we fit an ARMA(5,2)+GARCH(2,1) model using the `fGarch` package in R.

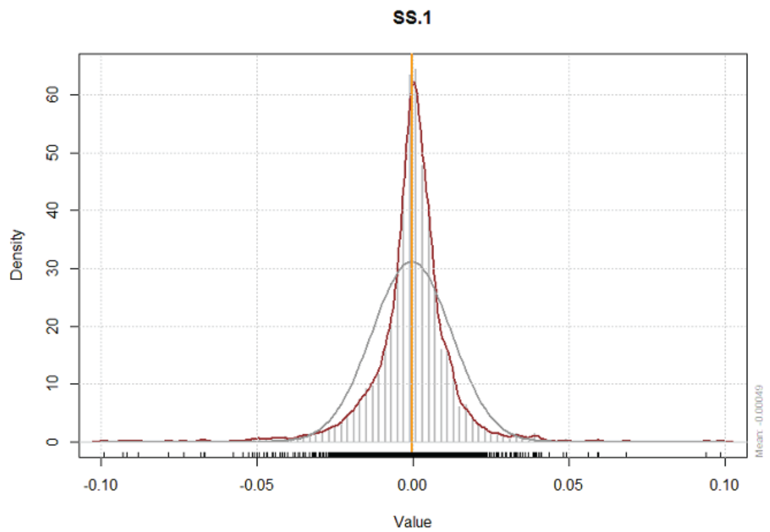
```
> ARCH_model <- garchFit(formula = ~ arma(5,2) + garch(2,1), data =
LogReturns, trace = F)
> summary(ARCH_model)

Call:
 garchFit(formula = ~arma(5, 2) + garch(2, 1), data = LogReturns, trace = F)


Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu       1.083e-03   2.616e-04    4.138 3.50e-05 ***
ar1      3.081e-01   1.114e-01    2.765 0.005687 **
ar2     -7.782e-01   1.212e-01   -6.419 1.37e-10 ***
ar3     -8.203e-02   2.970e-02   -2.762 0.005739 **
ar4     -2.340e-02   2.327e-02   -1.006 0.314525
ar5     -2.687e-02   2.585e-02   -1.039 0.298604
ma1     -3.761e-01   1.097e-01   -3.428 0.000609 ***
ma2      8.005e-01   1.172e-01    6.829 8.54e-12 ***
omega    2.976e-06   5.190e-07    5.735 9.73e-09 ***
alpha1   9.532e-02   2.150e-02    4.434 9.25e-06 ***
alpha2   7.640e-02   2.765e-02    2.763 0.005720 **
beta1    8.084e-01   1.985e-02   40.727  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  422.3005  0
 Shapiro-Wilk Test  R    W      0.975357  0
 Ljung-Box Test     R    Q(10)  7.339824  0.693021
 Ljung-Box Test     R    Q(15)  13.68987  0.5491683
 Ljung-Box Test     R    Q(20)  15.23322  0.7629107
 Ljung-Box Test     R^2  Q(10)  5.067725  0.8866082
 Ljung-Box Test     R^2  Q(15)  10.62788  0.7784997
 Ljung-Box Test     R^2  Q(20)  17.65104  0.610384
 LM Arch Test       R    TR^2   8.810223  0.7190488
```

The new parameter, `beta1`, refers to the moving average part of the GARCH process. Barring the `ar4` and `ar5` coefficients, all the coefficients are now significant. To get a parsimonious model, it should be tested whether these can be omitted from the model without losing the encompassing properties of the estimated model. We will set this issue aside for now to focus on the matter of interest for this module.

Note that while the normality tests still reject, all the ARCH tests do not reject the hypothesis that we have removed all ARCH effects from the residuals at the horizons of the tests.

Another way to see the nature of the deviation from normality is a kernel density estimate. Loosely defined, this is a smoothed version of a histogram that imposes the features required of a density function.

```
> GARCH_residuals <- ARCH_model@residuals
> GARCH_residuals <- timeSeries(GARCH_residuals)
> densityPlot(GARCH_residuals)
```



In this plot, the grey lines are a normalized histogram, the red curve the kernel density estimate, and the overlaid grey curve a normal distribution with the same mean and standard deviation as the data involved.

From this we can clearly observe that the closest density that represents residuals have far greater kurtosis than that of a normal distribution. It also seems to be slightly skewed to the right. The extensions to the GARCH model are built to deal with some of these features.

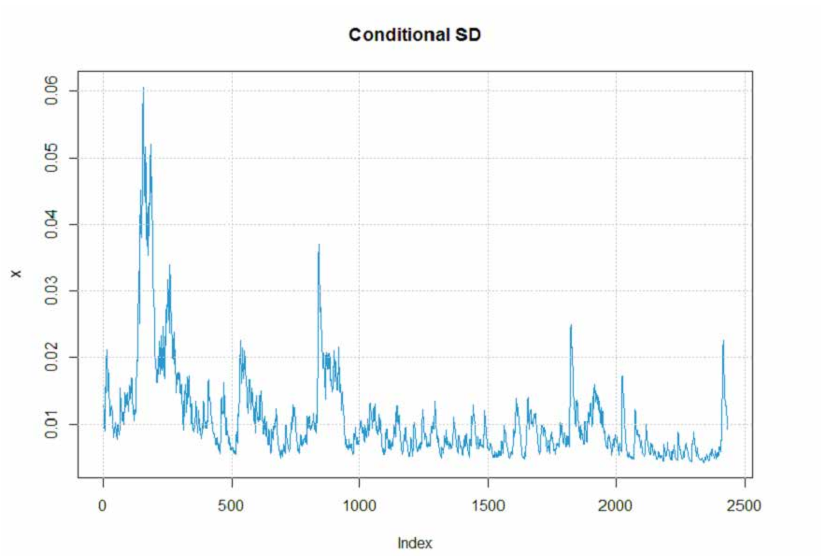
3.2.5 Notes: Forecasts

The level of the daily log returns of the S&P500 is largely unpredictable, and beyond that, it has a reasonably constant mean.

The conditional volatility is important to a financial engineer. Will tomorrow be more volatile or less volatile than today? We will not explore this here in detail, but a first pass of this would be to get the estimated path of the conditional variance of the process. This can be considered an ARMA process that may be forecasted in the ways that we covered in Module 3. There are limitations to what we can do with these models, which will be discussed in the next section of this module.

For now, we merely consider the nature of the fitted conditional volatility as estimated by our last model. In this case, the conditional standard deviation (i.e. the square root of the conditional variance):

```
> plot(ARCH_model,which=2)
```



This seems like a clearly autocorrelated process that may be evaluated by the methods of Module 3. Compare this to the plot of the daily log returns with which we started this module. The spikes in the conditional standard deviation clearly map the periods of excess volatility.

3.2.6 Notes: Maximum Likelihood Estimation

Unlike the linear models in Modules 2 and 3, the models in this module are not linear in parameters – there are complicated functional forms that relate all the parameters to each other simultaneously. As such, we cannot find a closed-form solution for the estimator as we could with the simple linear model.[2]

Therefore, we need a different estimation paradigm. The most important one that we review here is maximum likelihood.

The idea behind maximum likelihood is that, usually, we assume that there is some fixed data-generating process of which we wish to uncover the population coefficients. Given randomness, we know that any particular sample will occur with some probability given by the level of the joint density function of the stochastic data-generating process at the observed sample.

When we do maximum likelihood estimation, we turn this process around as a practical way of searching for the "best coefficients". That is, given the *fixed sample*, we ask what set of coefficient values would have made it most likely to have observed the fixed sample.

In simple cases, such as ordinary least squares with normal errors, we can show that the OLS estimator we derived is the same as the maximum likelihood estimator. In a situation as non-linear as a GARCH estimation, we need a different approach.

The optimization (the search for the maximum of the natural logarithm of the likelihood function[3]) is usually done by sophisticated numerical optimization methods. These are quite delicate for computational reasons far removed from the economic issues we study. You should not try to write your own optimizers until you have deeply studied the issues involved.

[2]Technically, a model with MA components is also not linear in parameters as there are recursive elements that require the same new paradigm.

[3]We use the logarithm of the likelihood function as the liklihood function is the product of many terms,where the logarithm of that product is a summation, which is much simpler to maximize. This is without loss of generality since the maximum of a function is also a maximum of any strictly increasing transformation of that function.

In the AR(1)-ARCH(1) model with normal errors:

$$\varepsilon_t \underset{iid}{\sim} N(0,1)$$
$$u_t = \varepsilon_t \sqrt{h_t}$$
$$y_t = a_0 + a_1 y_{t-1} + u_t$$
$$h_t = \omega + \alpha_1 u_{t-1}^2.$$

The conditional[4] log likelihood function is given by:

$$\ln L = -\frac{T-1}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=2}^{T}\ln(\omega + \alpha_1 u_{t-1}^2) - \sum_{t=2}^{2}\frac{(y_t - a_0 - a_1 y_{t-1})^2}{(\omega + \alpha_1 u_{t-1}^2)}.$$

Thus, the "estimation process" used by the packages we employed is a search with some algorithm for the values of ω, α_1, α_0 and α_1 that gives the largest value of $\ln L$ given our sample of observations $\{y_t\}_{t=1}^{T}$

[4]Conditional on the first error which we cannot observe

3.2.7 Transcript: Simulated GARCH Process and Estimation

In this module we have learned that it is often not possible for a GARCH process to fit perfectly to real-world data.

It is therefore useful to test the estimation methods on simulated data – that is, data that we have constructed to exactly fit the model of interest. This allows us to judge how well the estimation techniques actually work in the ideal situation.

For this purpose we will use the garchSim method from the fGarch package to create the data from the following AR(3)-GARCH(2,2) process:

$$y_t = 0.5y_{t-1} + 0.2y_{t-2} - 0.1y_{t-2} + u_t$$

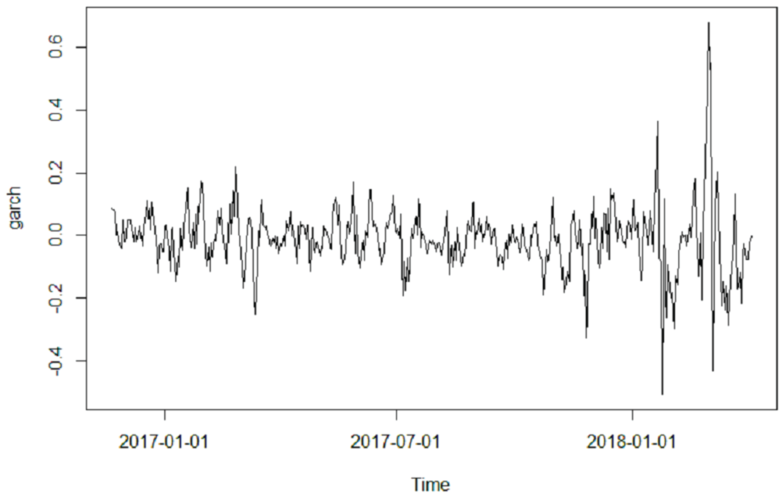
$$u_t = \varepsilon_t\sqrt{h_t}$$

$$\varepsilon_t \sim N(0, 1)$$

$$h_t = 0.001 + 0.3{u_{t-1}}^2 + 0.2{u_{t-2}}^2 + 0.2h_{t-1} + 0.1h_{t-2}.$$

The code to generate 500 observations of this process is:

```
library(fGarch)
library(forecast)
# generate a simulated process
spec = garchSpec(model = list(omega = 0.001, ar = c(0.5,0.2,-0.1), alpha =
c(0.3, 0.2), beta = c(0.2,0.1)))
process = garchSim(spec, n = 500)
plot(process)
```



The scale of the process and the time-scale is arbitrary.

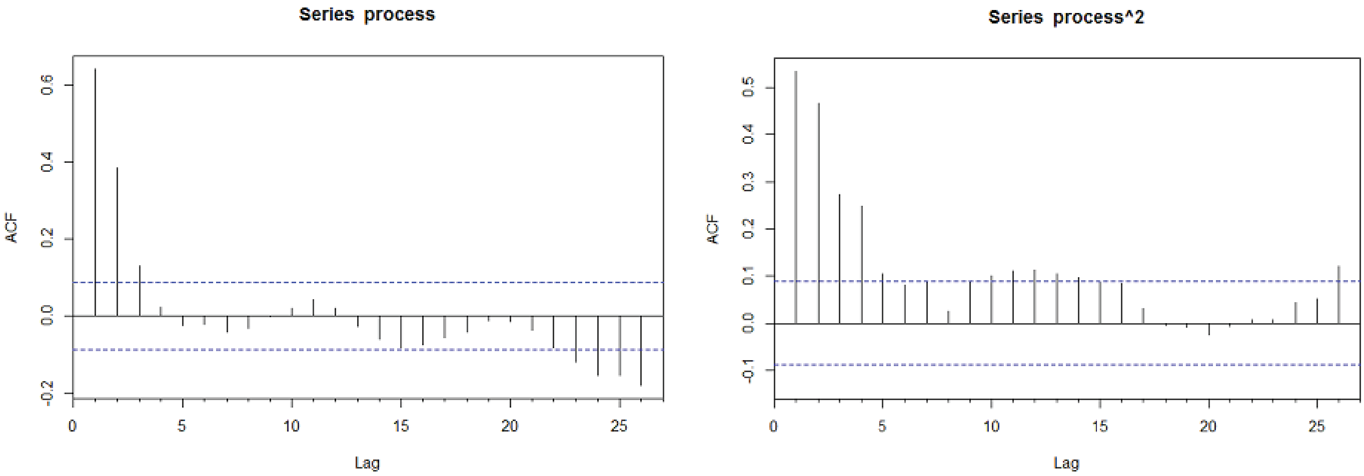
Your graph will look different as you will generate a series of unique random errors.

The standard approach in a text like this would have been to provide you with the "seed" – this is the "initial condition" that the pseudo-random-number-generator used by the code uses to produce the "random draws". This would have allowed you to reproduce the same graphs exactly as in these notes. We avoid this convention intentionally in this course. It is important for the empirical financial engineer to have a good understanding of how strong or weak "sampling variation" can be in the conclusion a statistical test reaches. For any "random" example you are presented in this Master's program, you should repeat "statistically equivalent" versions of it many times to establish how reliable the "standard" is under the conditions you view as representative.

Notice that it has many of the features of the examples in the main text of this module:

- There is significant persistence in the level of the series, although it is clearly not trending and wandering arbitrarily.
- There are periods of high volatility followed by periods of tranquility.

We can observe this in the ACF of the level and square of this process:



Both show significant autocorrelations for several lags, although the square of the process is nowhere near as persistent as in the real-world data we studied in the model. You should also observe the oscillations in the autocorrelation of the process.

Fitting models

Let's fit a few models: ones we know are wrong, and ones we know must be encompassing as we know exactly what the process is.

Fit an ARCH(3) model

```
> model1 <- garchFit(formula = ~ garch(3,0) , data = process, trace = F)
> summary(model1)
Title:
 GARCH Modelling
Error Analysis:
         Estimate   Std. Error  t value Pr(>|t|)
mu      -0.0037423   0.0034820   -1.075  0.28248
omega    0.0014521   0.0002492    5.827 5.64e-09 ***
alpha1   0.6019698   0.1000166    6.019 1.76e-09 ***
alpha2   0.2292895   0.0702393    3.264  0.00110 **
alpha3   0.1211404   0.0401046    3.021  0.00252 **
---
Standardised Residuals Tests:
                               Statistic p-Value
 Jarque-Bera Test   R    Chi^2  1.177431  0.5550397
 Shapiro-Wilk Test  R    W      0.9961873 0.2748569
 Ljung-Box Test     R    Q(10)  206.8405  0
 Ljung-Box Test     R    Q(15)  214.9172  0
 Ljung-Box Test     R    Q(20)  217.6125  0
 Ljung-Box Test     R^2  Q(10)  8.057226  0.6232471
 Ljung-Box Test     R^2  Q(15)  9.292568  0.8617298
 Ljung-Box Test     R^2  Q(20)  10.23874  0.9636334

 LM Arch Test       R    TR^2   9.059357  0.6978523
```

Note that while this is sufficient to capture the time series properties of the squared residuals (i.e. the GARCH effects), the levels of the residuals are still autocorrelated. The residuals test as normal with no uncertainty.

Fit the correct AR(3)-GARCH(2,2) model

```
> model2 <- garchFit(formula = ~ arma(3,0) + garch(2,2) , data = process, trace = F)
> summary(model2)

Error Analysis:
         Estimate  Std. Error  t value Pr(>|t|)
mu     -1.261e-03   2.144e-03   -0.588 0.556414
ar1     5.645e-01   4.659e-02   12.116  < 2e-16 ***
ar2     1.818e-01   5.654e-02    3.216 0.001299 **
ar3    -1.490e-01   4.517e-02   -3.298 0.000975 ***
omega   1.012e-03   2.256e-04    4.484 7.33e-06 ***
alpha1  3.411e-01   8.318e-02    4.101 4.11e-05 ***
alpha2  3.514e-01   9.158e-02    3.837 0.000125 ***
beta1   1.000e-08   1.627e-01    0.000 1.000000
beta2   1.242e-01   1.129e-01    1.100 0.271285
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Standardised Residuals Tests:
                                Statistic p-Value
 Jarque-Bera Test   R    Chi^2  3.441567  0.1789259
 Shapiro-Wilk Test  R    W      0.9962484 0.2879572
 Ljung-Box Test     R    Q(10)  8.876449  0.5438679
 Ljung-Box Test     R    Q(15)  14.10072  0.5179047
 Ljung-Box Test     R    Q(20)  16.11481  0.7094768
 Ljung-Box Test     R^2  Q(10)  6.775412  0.7464636
 Ljung-Box Test     R^2  Q(15)  9.848108  0.829184
 Ljung-Box Test     R^2  Q(20)  11.31691  0.9375862
 LM Arch Test       R    TR^2   8.010693  0.7842942

Information Criterion Statistics:
      AIC       BIC       SIC      HQIC

-2.785105 -2.709242 -2.785738 -2.755336
```

Now, all the diagnostic tests on the residuals look good. They are normal and without autocorrelation in either level or square. Thus, the model seems encompassing.

Note that the coefficients of the AR process are significant and almost within two standard deviations of their true values. However, the coefficients on the ARCH effects, alpha1 and alpha2, are significant but they are also statistically significantly different from their true values and the GARCH effect coefficients, beta1 and beta2, are nowhere near significant.

This often happens in situations where there are both AR and MA effects in either the conditional expectation or conditional variance – even though the process is generated as GARCH(2,2), it may be that a simpler empirical specification is capable of capturing the same time series properties more parsimoniously. This is due to the fact that there may exist many different but equivalent representations of the same process.

You should check that when you estimate an AR(3)-ARCH(2) model, you obtain an encompassing and parsimonious model.

The GARCH-M model

The GARCH-in-mean, or GARCH-M, model addresses the limitations of standard ARCH/GARCH models which do not allow the realization of the conditional variance process to affect the conditional expectation. It can be established that during a financial crisis the returns on many assets are lower than in more tranquil times. Alternatively, an asset that is systematically riskier than another must offer a higher return to be invested in.

Abstracting from ARMA properties of some return series , a simple version, the GARCH(1,1)-M is given by the data-generating process:

$$r_t = \mu + c\sigma_{t-1}^2 + u_t$$
$$u_t = \varepsilon_t \sqrt{h_t}$$
$$h_t = \omega + \alpha_1 u_{t-1}^2 + \beta_1 h_{t-1}.$$

If c is significantly positive, it can be considered a "risk premium" for the asset in times when it is more volatile. Other alternatives are to use just the conditional standard deviation or the log of the variance in the mean equation. The most appropriate data-generating process will depend on the economics behind the data-generating process you are trying to model.

Models with asymmetry: The TGARCH and EGARCH models

Enders (2014) argues as follows for asymmetric effects of volatility via leverage effects: a piece of "bad news", or a negative return (i.e. a fall in the stock price) reduces the value of equity relative to the fixed value of debt of that firm. This increase in leverage, in turn, makes the stock riskier, i.e., we expect the volatility of the returns to increase. The opposite effect will also be present – an increase in the stock price reduces leverage and risk.

To model this, we can use the threshold-GARCH model (TGARCH) where we add a dummy variable d_{t-1} which is zero when $\varepsilon_{t-1} > 0$ and equal to 1 when $\varepsilon_{t-1} < 0$.

The simplest case, considering only the variance equation in h_t, is the TGARCH(1,1) model:

$$h_t = \omega + \alpha_1 u_{t-1}^2 + \gamma_1 d_{t-1} u_{t-1}^2 + \beta_1 h_{t-1}.$$

Thus, if $\varepsilon_{t-1} > 0$ (so $u_{t-1} > 0$), the impact of u_{t-1}^2 on h_t is α_1. If $\varepsilon_{t-1} < 0$, the impact of u_{t-1}^2 on h_t is $(\alpha_1 + \gamma_1)$. If is positive and statistically significant, it means that negative shocks have larger (positive) impacts on volatility than positive shocks.

The EGARCH model employs a logarithmic transformation to h_t to ensure positive variances - in all the other models, all the estimated coefficients need to be positive, which is not the case here - as well as working in the *levels* of the residuals (not their squares) and, additionally, standardizing by their standard deviation $\sqrt{h_t}$.

Specifically, we assume for the EGARCH(1,1) model:

$$\ln(h_t) = \omega + \alpha_1 \frac{u_{t-1}}{\sqrt{h_{t-1}}} + \gamma_1 \left| \frac{u_{t-1}}{\sqrt{h_{t-1}}} \right| + \beta_1 \ln(h_{t-1}).$$

Again, we have leverage effects: if $u_{t-1} > 0$ its impact on $\ln(h_t)$ is $\alpha_1 + \gamma_1$. if $u_{t-1} < 0$ its impact on is $\ln(h_t)$ is $-\alpha_1 + \gamma_1$.

Enders (2014) warns, however, that it may be very difficult to forecast with this model.

The IGARCH Model

Above, we augmented our stationary ARMA(p,q) processes by allowing the conditional variance to have some ARMA-like persistence, while maintaining stationarity of the process. Therefore, the unconditional expectations and variances were constant.

If we were to augment and integrate, i.e. ARIMA(p,d,q) with d>0, we would obtain an integrated GARCH, or IGARCH, model. This is a model where a shock to the conditional variance (and/or the conditional expectation) never fades out.

We have not explored this yet, but theoretically even a simple unit root process does not have a well-defined unconditional variance. This property carries over to the IGARCH process as well. This is intuitive: since a unit root process, even without drift, may wander arbitrarily up or down for arbitrarily long periods, there is no way to describe, with a single number, what variability we may expect over all time. In formal terms, starting from some point in time, the process will have a well-defined conditional variance.

If we attempt to take the limit of this to find the unconditional variance, we end up adding an infinite sequence of positive conditional variances, implying that the unconditional variance "equals" infinity. This is what we mean by "not well defined". In practice, it is hard to think of examples of interest to the financial engineer that fall into this category, although Tsay (2010) gives an application to value at risk.

The stochastic volatility model

The ARCH/GARCH models previously explained all assume that there is only one fundamental source of randomness – the white noise error process , which gives the innovation to the conditional expectation equation. Via the equations that characterize the data-generating process, we convert this into a time series process for both the conditional expectation and the conditional variance.

The stochastic volatility model assumes that there are two independent innovations:

$$\varepsilon_t \underset{iid}{\sim} N(0,1)$$
$$\nu_t \underset{iid}{\sim} N(0,\sigma_\nu^2).$$

Assuming again for expositional purposes that an AR(1) process is sufficient to capture the time series properties of the conditional mean, the full data-generating process is then:

$$y_t = a_0 + a_1 y_{t-1} + u_t$$
$$u_t = \varepsilon_t \sqrt{h_t}$$
$$\ln(h_t) = \omega + \alpha_1 \ln(h_{t-1} + ... + \alpha_m \ln(h_{t-m}) + \nu_t$$

where we again use the natural logarithm of the conditional variance to ensure that it is always positive in levels.

Since we now have two independent error processes, but only one observed series (the level of our variable of interest), we need more sophisticated estimation methods to uncover the parameters of this model. The two options are the **Kalman filter** (an extension of maximum likelihood) or **Markov Chain Monte Carlo** (MCMC) methods (which is a stochastic search algorithm that uses simulation methods to find the best set of parameters).

The details of the Kalman filter are beyond the scope of these notes, so only a brief discussion is provided here. The Kalman filter is an extension to the maximum likelihood approach, particularly built to provide an estimation/filtering technique for what is known as state space models.

These models always have some form of the following structure:

- **An observation equation**. That is a model equation that states how an observed endogenous variable depends on other observed variables and some unobserved state variables. In the stochastic volatility model, the observation equation is the equation for the observed mean of the process, while the state variables are the unobserved errors and the current value of the conditional variance equation.

- **A state equation**. This is a model equation that describes how the state of the system (in this case, the conditional volatility) evolves over time.

The Kalman filter uses a probabilistic model of the evolution and disturbances of the observed variables and the fundamentally unobserved state of the system to construct period-by-period best estimates of the value of the state. That is, we use the best estimates of the parameters of the system of the equation combined with our distributional assumption (e.g. that the innovations are normal) to construct an estimate of the current state of the system. Then, when a new observation arrives, we use the new information to update our best guess of the parameters, and the previously estimated state variable, to provide our best estimates of the new current level of the state variable. This is, in turn is used to improve the forecast of the observed variables.

Different versions of the Kalman filter exist, some employing only forward filtering, others applying both forward and backward filtering (i.e. after going from the first to last observation, starting from the last observation and asking what would have been the best guess of the previous state variable level). In this sense, the Kalman filter provides a smoothed estimate of the true state variable process in any sample.

Tsay (2010) states that while the stochastic volatility models show some ability to improve in sample fit, its out-of-sample forecasting ability relative to other models has yielded mixed results.

References

Bollerslev, T. (1986) ‘Generalized Autoregressive Conditional Heteroskedasticity’, *Journal of Econometrics*, 31(3), pp. 307–327. doi: 10.1016/0304-4076(86)90063-1.

Enders, W. (2014) *Applied Econometric Time Series*. 4th edn. Wiley.

Engle, R. F. (1982) ‘Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation’, *Econometrica*, 50(4), pp. 987–1007.

Fred.stlouisfed.org (2018) ‘Data: S&P500 daily index’.

Hamilton, J. D. (1992) *Time Series Analysis*. Princeton: Princeton University Press.

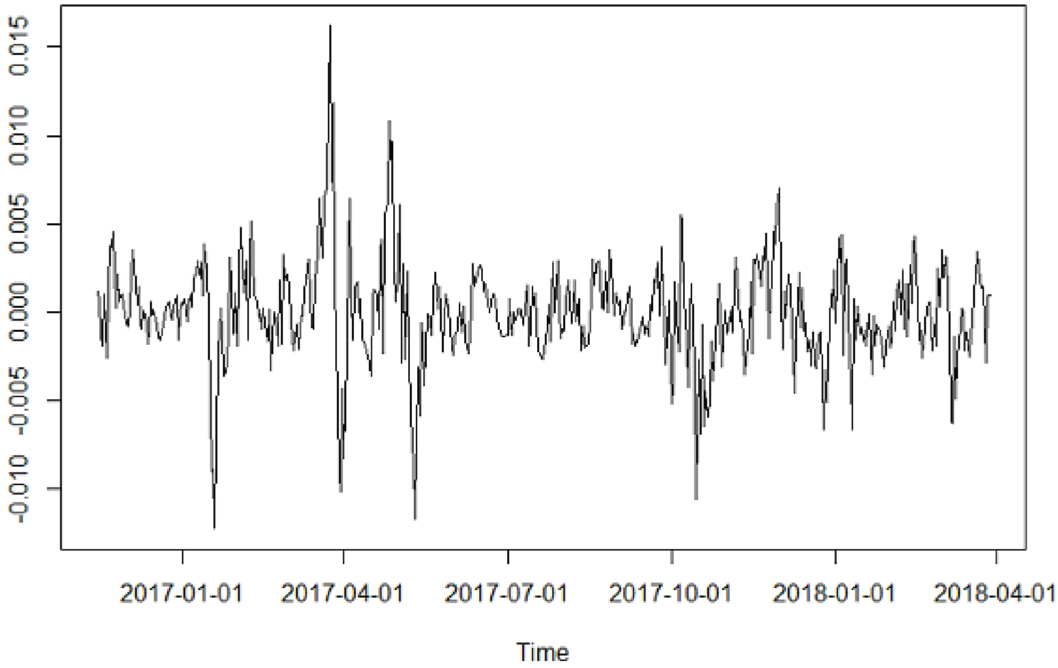
Tsay, R. S. (2010) *Analysis of Financial Time Series*. Wiley.

3.3 Peer Review Question

In each module, you are required to complete a peer review assignment, in addition to two multiple-choice quizzes. The peer review assignment is designed to test your ability to apply and analyze the knowledge have learned in the module.

Question

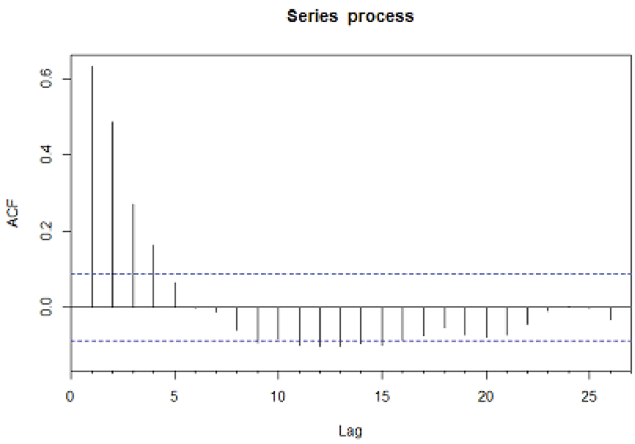
Consider the following plot of a time series process:

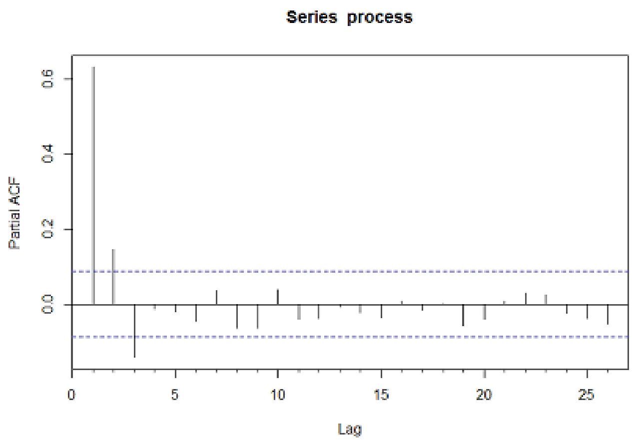


1. Discuss what can be deduced from the time series properties (persistence and volatility) of the series from the time series plot.

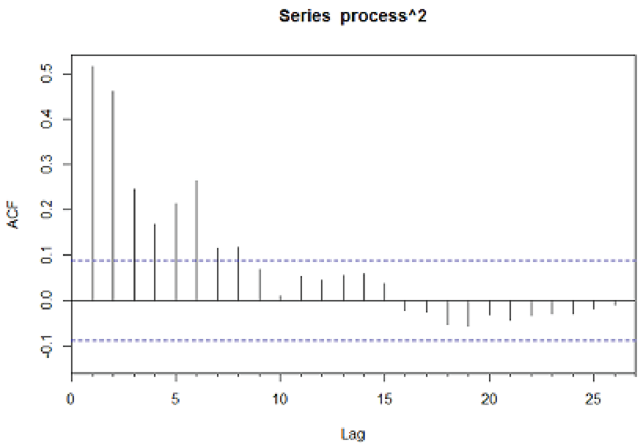
2. Next, consider the ACF and PACF of the process and its square.

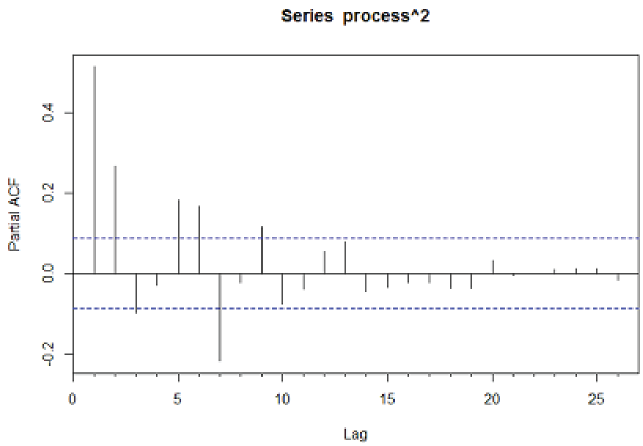
ACF and PACF of the level of the process:





ACF and PACF of the square of the process:





Discuss what can be deduced from the time series properties of the series from these ACF and PACF functions and discuss what this means for the appropriate modeling approach.

3. Consider the output of estimating 6 different ARMA-GARCH models for this process via maximum likelihood assuming normal errors:

	Model 1		Model 2		Model 3		Model 4		Model 5		Model 6	
Coefficient	Estimate	p-value	Estimate	p-value	Estimate	p-value	Estimate	p-value	Estimate	p-value	Estimate	p-value
mu	0.000	0.717	0.000	0.713	0.000	0.925	0.000	0.927	0.000	0.061	0.000	0.100
ar1	0.531	0.000	0.531	0.000	0.514	0.000	0.699	0.000				
ar2	0.188	0.000	0.188	0.000	0.135	0.002						
ar3	-0.104	0.010	-0.104	0.011								
ma1							-0.157	0.012				
omega	0.000	0.006	0.000	0.017	0.000	0.072	0.000	0.120	0.000	0.000	0.000	0.224
alpha1	0.456	0.000	0.448	0.000	0.476	0.000	0.477	0.000	0.615	0.000	0.567	0.000
alpha2			0.003	0.980	0.149	0.628	0.149	0.698	0.287	0.000	0.264	0.678
beta1	0.520	0.000	0.525	0.000	0.000	1.000	0.000	1.000			0.000	1.000
beta2					0.338	0.408	0.342	0.510			0.158	0.770

			Model 1		Model 2		Model 3		Model 4		Model 5		Model 6	
	Test on:	Test Distribution	Statistic	p-value	Statistic	p-value	Statistic	p-value	Statistic	p-value	Statistic	p-value	Statistic	p-value
Jarque-Bera	*residuals*	*Chi^2*	3.135	0.209	3.090	0.213	3.148	0.207	2.697	0.260	2.241	0.326	0.423	0.810
Shapiro-Wilk	*residuals*	*W*	0.996	0.183	0.996	0.183	0.995	0.082	0.995	0.129	0.996	0.234	0.995	0.132
Ljung-Box	*residuals*	*Q(10)*	3.348	0.972	3.379	0.971	4.617	0.915	6.424	0.778	166.590	0.000	177.066	0.000
Ljung-Box	*residuals*	*Q(15)*	5.473	0.987	5.515	0.987	6.774	0.964	8.633	0.896	182.690	0.000	194.670	0.000
Ljung-Box	*residuals*	*Q(20)*	8.571	0.987	8.611	0.987	9.693	0.973	11.450	0.934	193.579	0.000	207.810	0.000
Ljung-Box	*residuals^2*	*Q(10)*	8.468	0.583	8.352	0.595	8.678	0.563	9.317	0.502	27.412	0.002	9.255	0.508
Ljung-Box	*residuals^2*	*Q(15)*	9.250	0.864	9.062	0.874	9.479	0.851	10.422	0.792	32.123	0.006	12.362	0.651
Ljung-Box	*residuals^2*	*Q(20)*	14.195	0.820	14.029	0.829	12.543	0.896	13.230	0.867	37.976	0.009	19.198	0.509

Identify the ARMA-GARCH structure of each of the models.

4. Analyze each of the six models and select the best one. Your analysis should touch on the values and statistical significance of coefficients and tests, as well as come to a conclusion regarding whether the models are encompassing and/or parsimonious. All coefficients need not be discussed individually, only to the extent necessary for giving a complete answer.