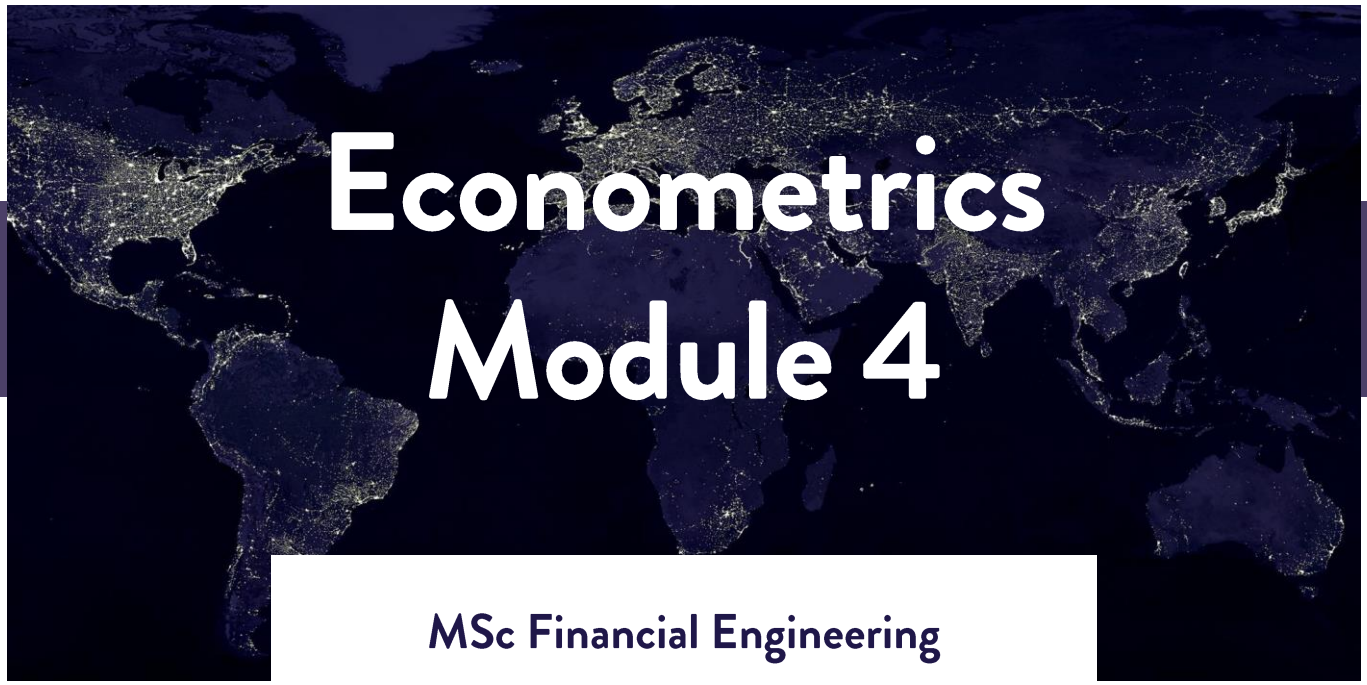
WORLDQUANT UNIVERSITY



Econometrics Module 4

MSc Financial Engineering

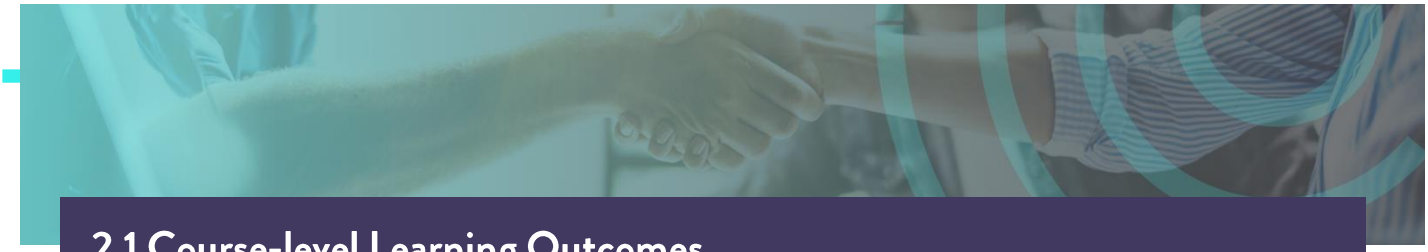
Table of Contents

1. Brief

This document contains the core content for Module 4 of Econometrics, entitled Univariate Volatility Modeling. It contains five sets of notes and one lecture transcript.

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, you will apply statistical techniques to the analysis of econometric data. The course starts with an introduction to the R statistical programming languages that you will use to build econometric models, including multiple linear regression models, time series models, and stochastic volatility models. You will learn to develop programs using the R language, solve statistical problems, and understand value distributions in modeling extreme portfolio and basic algorithmic trading strategies. The course concludes with a review on applied econometrics in finance and algorithmic trading.



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 Use graphic techniques to visualize multidimensional data.

5 Apply multivariate statistical techniques (PCA, factor analysis, etc.) to analyze multidimensional data.

6 Fit a time series model to data.

7 Fit discrete-time volatility models.

8 Understand and apply filtering techniques to volatility modeling.

9 Understand the use of extreme value distributions in modeling extreme portfolio returns.

10 Define some common risk measures like VaR and Expected Shortfall.

11 Define and use copulas in risk management.

12 Implement basic algorithmic trading strategies

2.2 Module Breakdown

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

1. Basic Statistics
2. Linear Models
3. Univariate Time Series Models
4. Univariate Volatility Modeling
5. Multivariate Time Series Analysis
6. Introduction to Risk Management
7. Algorithmic Trading

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

After completing this 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.
4. Describe other volatility models.

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.

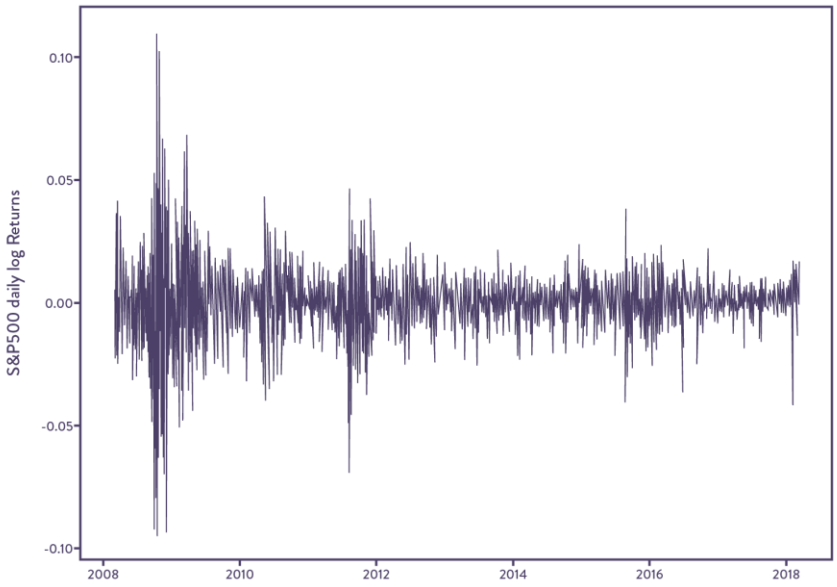


Figure 1

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, 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, 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.

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).

3.2.2 Notes: ARCH and GARCH Models

The simplest ARCH Model

The simplest case of an autoregressive 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 (that is, 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. Note that nothing would change if we assumed a more complicated time series process for the conditional mean, but all the important aspects of this choice remains 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:

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

[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.

We allow the conditional variance $\mathbb{E}_t(u_t^2) = \sigma_t^2$ to be auto-correlated. 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 leads to the different types of ARCH models. A large number of different models have been developed for different applications.

The simplest version which we will call an ARCH(1). 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:

$$\begin{aligned}
\mathbb{E}_{t-1}(u_t^2) &= \mathbb{E}_{t-1}\left(\varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2}\right)^2 \\
&= \sigma_\varepsilon^2 \left(\omega + \alpha_1 \mathbb{E}_{t-1}(u_{t-1}^2)\right) \\
&= \omega + \alpha_1 u_{t-1}^2.
\end{aligned}$$

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:

$$\begin{aligned}
\mathbb{E}(u_t^2) &= \mathbb{E}\left(\varepsilon_t \sqrt{\omega + \alpha_1 u_{t-1}^2}\right)^2 \\
&= \omega + \alpha_1 \mathbb{E}(u_{t-1}^2) \\
&= \frac{\omega}{1 - \alpha_1},
\end{aligned}$$

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:

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

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

$$u_t = \varepsilon_t \sqrt{h_t}$$
$$h_t = \omega + \sum_{i=1}^{p} \alpha_i u_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j}$$

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 to the ARCH model

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).

1 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 for 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.

2 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.

3 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).

4 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 is built to allow this feedback.

5 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.

3.2.3 Notes: Maximum Likelihood Estimation of GARCH Models

Many software solutions have built-in packages that allow the implementation of Maximum Likelihood Estimation (or MLE) using the click of a button or one code line. The details regarding MLE are provided in the next paragraph.

MLE in a linear regression

$\{\varepsilon_t\}$ – drawn from a normal distribution having a zero mean and a constant variance σ^2.
The likelihood of any realization of $\{\varepsilon_t\}$ is:

$$L_t = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) exp\left(\frac{-\varepsilon_t^2}{2\sigma^2}\right)$$

L_t – likelihood of ε_t
$\{\varepsilon_t\}$ – independent ⟹the likelihood of the joint realizations is the product of the individual likelihoods.

The likelihood of the joint realizations is the following:

$$L = \prod_{t=1}^{T}\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) exp\left(\frac{-\varepsilon_t^2}{2\sigma^2}\right)$$

We apply the natural logarithm to both sides as it is easier to work with the sum as compared to the product.

$$lnL = -\frac{T}{2}\ln(2\pi) - \frac{T}{2}ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{t=1}^{T}\varepsilon_t^2 \quad \textbf{(1)}$$

The goal is to select the distributional parameters so as to maximize the likelihood of drawing the observed sample.

$\{\varepsilon_t\}$ is generated by the following model:

$$\varepsilon_t = y_t - \beta x_t \tag{2}$$

We substitute equation (2) into (1). The likelihood function becomes:

$$lnL = -\frac{T}{2}\ln(2\pi) - \frac{T}{2}ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{t=1}^{T}(y_t - \beta x_t)^2 \tag{3}$$

Maximizing the log likelihood function with respect to σ^2 and β:

$$\frac{\partial lnL}{\partial \sigma^2} = -\frac{T}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{t=1}^{T}(y_t - \beta x_t)^2 \tag{4}$$

and

$$\frac{\partial lnL}{\partial \beta} = \frac{1}{\sigma^2}\sum_{t=1}^{T}(y_t x_t - \beta x_t^2). \tag{5}$$

The partial derivatives are equal to zero.

$$\frac{\partial lnL}{\partial \sigma^2} = 0 \tag{6}$$

and

$$\frac{\partial lnL}{\partial \beta} = 0. \tag{7}$$

Solving for the values of σ^2 and β yield the maximum value of lnL

$$\hat{\sigma}^2 = \frac{\sum \varepsilon_t^2}{T} \quad \textbf{(8)}$$

$$\hat{\beta} = \frac{\sum x_t y_t}{\sum x_t^2} \quad \textbf{(9)}$$

The results are similar to the ones provided by OLS estimates. It is easy to find the optimal solutions when there is a linear model.

MLE at GARCH models

Things get complicated in GARCH models as the first-order equations are non-linear. The solutions are obtained using a search algorithm. Equation (2), identified in the previous section, considers an ARCH(1) process.

$$y_t - \beta x_t = \varepsilon_t$$

ε_t is modeled as following:

$$\varepsilon_t = v_t (h_t)^{0.5} \text{ (the conditional variance is not constant)} \quad \textbf{(10)}$$

$$L = \prod_{t=1}^{T} \left(\frac{1}{\sqrt{2\pi h_t}}\right) exp\left(\frac{-\varepsilon_t^2}{2h_t}\right) \quad \textbf{(11)}$$

The log likelihood function is the following:

$$lnL = -\frac{T}{2}\ln(2\pi) - 0.5\sum_{t=1}^{T} ln(h_t) - 0.5\sum_{t=1}^{T}\left(\frac{\varepsilon_t^2}{h_t}\right) \quad \textbf{(12)}$$

$$\varepsilon_t = y_t - \beta x_t$$

The conditional variance for ARCH(1) process is the following:

$$h_t = a_0 + a_1 \varepsilon_{t-1}^2 \quad \textbf{(13)}$$

We substitute for h_t and y_t yields:

$$lnL = -\frac{T-1}{2}\ln(2\pi) - 0.5\sum_{t=2}^{T}\ln(\alpha_0 + \alpha_1\varepsilon_{t-1}^2) - 0.5\sum_{t=2}^{T}\left(\frac{y_t-\beta x_t}{\alpha_0+\alpha_1\varepsilon_{t-1}^2}\right) \quad \textbf{(14)}$$

- The initial observation is lost since ε_0 is outside the sample
- There are no simple solutions to the first-order conditions for a maximum
- Computers are able to select parameter values that maximize log likelihood

Estimating GARCH parameters

Omega, alpha and beta parameters are estimated in Python.

Data: EUR/USD

Frequency: Daily

Source: FRED

Starting date: January 2nd, 2013

End date: November 14th 2014

```
>>> fx2=np.log(fx)
>>> def GARCH11_logL(param, fx2):
...  omega, alpha, beta = param
...  n = len(fx)
...  s = np.ones(n)
...  for i in range(3,n):
...   s[i] = omega + alpha*fx2[i-1]**2 + beta*(s[i-1])
...  logL = -( (-np.log(s) - fx2**2/s).sum() )
...  return logL
...
>>> R = optimize.fmin(GARCH11_logL,np.array([.1,.1,.1]),args=(fx2,),full_output=1)
Optimization terminated successfully.
         Current function value: -699.399278
         Iterations: 98
         Function evaluations: 174
>>> print("omega = %.6f\nbeta  = %.6f\nalpha = %.6f\n") % (R[0][0],R[0][2],R[0][1])
omega = 0.000500
beta  = 0.002339
alpha = 0.991121
```

Source: Pawel Lachowicz from QuantatRisk

According to Python, the GARCH(1,1) estimated parameters are:

Omega=0.000500

Beta=0.002339

Alpha=0.991121

3.2.4 Transcript: Simulated GARCH Process and Estimation

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

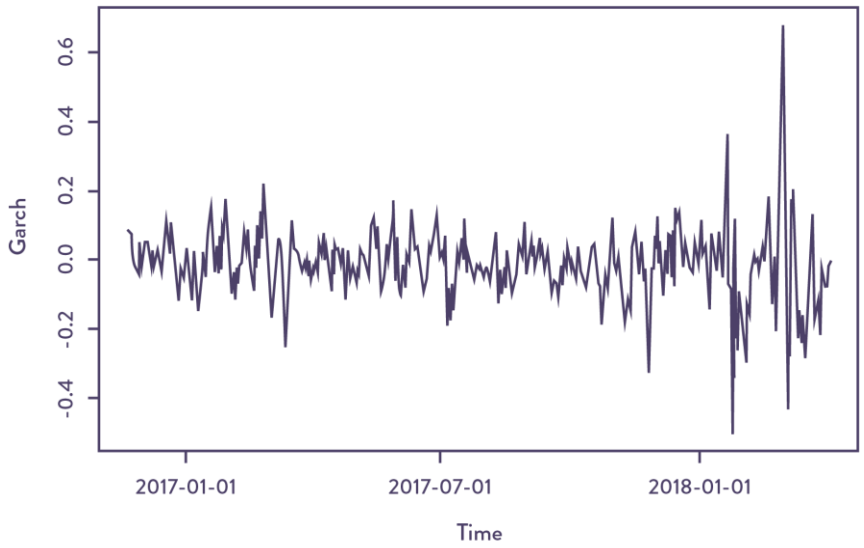
It is therefore useful to test the estimation methods on simulated data – that is, data that we have constructed to exactly fit the model of interest. This allows us to judge how well the estimation techniques actually work in the ideal situation.

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:

$$y_t = 0.5y_{t-1} + 0.2y_{t-2} - 0.1y_{t-2} + u_t$$
$$u_t = \varepsilon_t\sqrt{h_t}$$
$$\varepsilon_t \sim N(0,1)$$
$$h_t = 0.001 + 0.3u_{t-1}^2 + 0.2u_{t-2}^2 + 0.2h_{t-1} + 0.1h_{t-2}$$

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.

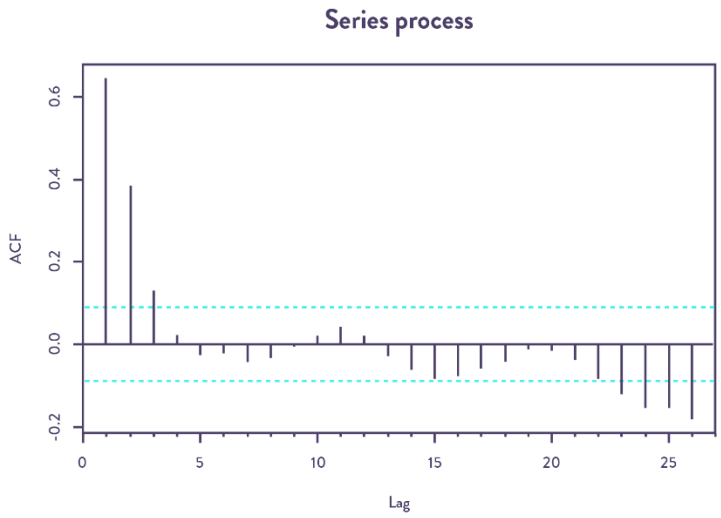
Your graph will look different as you will generate a series of unique random errors. The standard approach in a text like this would have been to provide you with the "seed" – this is the "initial condition" that the pseudo random number generator used by the code uses to produce the "random draws". This would have allowed you to reproduce the same graphs *exactly* as in these notes.

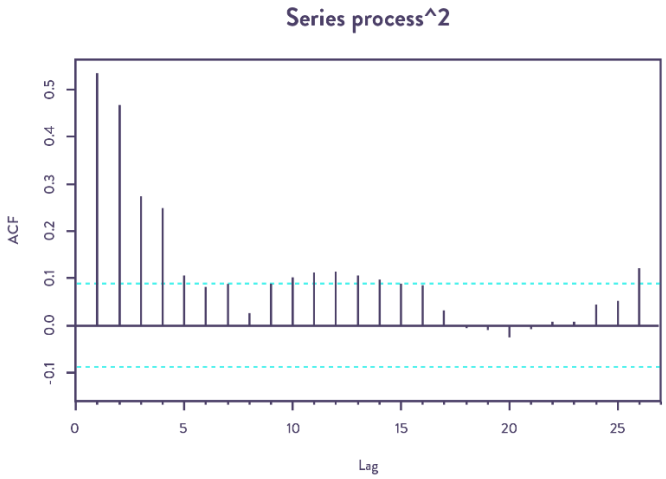
We avoid this convention intentionally in this course: it is important for the empirical financial engineer to have a good understanding of how strong or weak "sampling variation" can be in the conclusion a statistical test reaches. For any "random" example you are presented in this master's program, 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 auto-correlations 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 auto-correlation 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 auto-correlated. 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 auto-correlation 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.

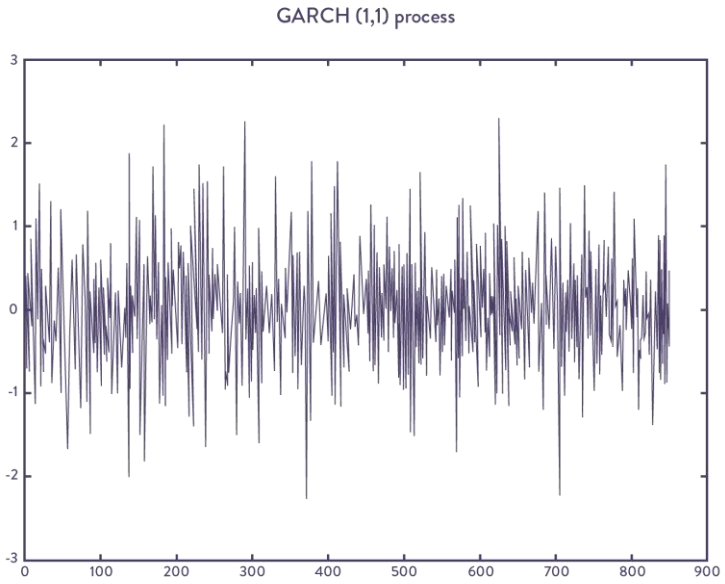
3.2.4.1 Post-script

You may also experiment with simulating a GARCH(1,1) process in Python as follows:

Simulating a GARCH(1,1) process in Python:

In [ ]:

```
import scipy as sp

from matplotlib.pyplot import *

 sp.random.seed(12345)

v=850 # v is the number of observations

v1=150 # we need to drop the first several observations

v2=v+v1 # sum of two numbers

alpha=(0.15,0.25) # GARCH (1,1) coefficients alpha0 and alpha1, see Equation (3)

beta=0.2

errors=sp.random.normal(0,1,v2)

t1=sp.zeros(v2)

t1[0]=sp.random.normal(0,sp.sqrt(alpha[0]/(1-alpha[1])),1)

for i in range(1,v2-1):

...  t1[i]=errors[i]*sp.sqrt(alpha[0]+alpha[1]*errors[i-1]**2+beta*t1[i-1]**2)

y=t1[v1-1:-1] # drop the first n1 observations

title('GARCH (1,1) process')

x=range(v)

plot(x,y)
```



3.2.5 Notes: GARCH in Algorithmic Trading

Forecasting financial assets represents a challenge for quantitative analysts as variables are influenced by a wide range of factors (macroeconomic evolutions, liquidity, market psychology or regional conflicts). Knowing the future path of financial assets is crucial in trading decisions. The following points present how GARCH models can be used in quantitative trading.

- GARCH models are popular quantitative models for projecting high-frequency indicators that do not have constant volatility like the Foreign Exchange (also referred to as "FX"), short-term interest rates, commodities, stock prices, cryptocurrencies or stock market indexes. Most GARCH research papers focus on stock and FX, forecasting, and other types of assets being less popular.
- Univariate Stochastic Volatility models like GARCH are generally used to predict next period evolution $(t + 1)$ of asset returns or prices. The predictions are used to generate short-term trading signals (buy/sell).
- It is generally easier to predict a trend than to predict a certain level of price / return.
- The models are used to capture the so-called volatility clustering phenomenon in financial markets. Volatility clustering means that "large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes" (Mandelbrot, 1963). Consequently, high volatility is usually followed by a high-volatility period, while low volatility is usually followed by a low-volatility period in the financial markets.
- "Turbulent behavior" is artificially introduced in the model by generating random numbers from a normal or t-distribution (Mandelbrot, 2004).
- It denies the existence of long dependence (excepting FIGARCH models).
- It represents a combination of elements long familiar to statisticians.
- Algorithmic trading companies have started to combine forecasts from different models and algorithms in order to obtain trading signals (hybrid

approach). So GARCH predictions are used as inputs in more sophisticated algorithms and strategies.

- GARCH does not incorporate finance theory. It identifies data patterns. A combination of text mining and GARCH approach can be used in more sophisticated approaches.

Basic FX Trading using GARCH

- The strategy is carried out on a rolling basis. GARCH parameters are estimated based on previous k days EUR/USD daily returns. So k days are used as a window to estimate GARCH parameters. The goal is to forecast next day $(t+1)$ return value.
- If the predicted EUR/USD > 1 -> EUR is expected to appreciate against USD -> Buy EUR and sell USD.
- If the predicted EUR/USD <1 -> EUR is expected to depreciate against USD -> Sell EUR and buy USD.
- If the predicted EUR/USD = 1 -> the algorithm does not trade.
- FX trading is risky and requires high initial investment.
- The quant trader indicates the level of k (it can be 500 or 1,000 for example). If k is too low, the algorithm cannot estimate GARCH parameters. If k is too high, there could be structural breaks in the data not correctly captured by parameters. Therefore, GARCH parameters can vary from one period to another.
- The GARCH model may not converge in certain cases (the solution is not unique). In this case the algorithm can indicate "buy EUR", which is simply a guess.
- The mentioned algorithm is basic and there is room for further improvement. Ways for improving the algorithm include introducing signals generated by:

1. Technical indicators (MA, Bollinger bands, etc.)
2. Text mining
3. Different GARCH models
4. Macroeconomic indicators

5 Other algorithms or models (Different values for k can be introduced in the model.)

Forecast daily EUR/USD return using GARCH(1,1) model

Forecasting the FX represents a challenge for every quantitative analyst as the variable is influenced by different factors (macroeconomic evolutions, market psychology or regional conflicts). Knowing the future path helps FX quant traders in decision taking. We forecast the evolution of EUR/USD using a GARCH(1,1) model. GARCH models are popular quantitative models for projecting high-frequency indicators that do not have constant volatility like the foreign exchange, short-term interest rates, stock prices or stock market indexes. The following example shows how to use the GARCH approach in FX forecasting.

Asset: EUR/USD daily return

Period: January 17, 2018 – January 17, 2019

Source: FRED database

https://fred.stlouisfed.org/series/DEXUSEU

Software: R 3.5.2

Rugarch package may not work in older versions of R as truncnorm package cannot be loaded.

Null values are removed from the analysis

The complete code can be found in *EURUSD_GARCH_Forecast* txt file in Additional files folder.

R code:

```
install.packages("rugarch")
library("rugarch")
plot(fx)

#Specify GARCH(1,1) model using ugarchspec function.
#garchOrder=c(1,1)
```

```
#mean model should be white noise
#armaOrder=c(0,0)

garch11_spec <- ugarchspec(variance.model = list(
  garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(0, 0)))
#Estimate GARCH model
garch11_fit<-ugarchfit(spec=garch11_spec, data=fx)
#Forecast FX
garch11_forecast<-ugarchforecast(garch11_fit, nahead=15)
garch11_forecast
```

Python and Matlab code for GARCH(1,1) can be found on Dr. Pawel Lachowicz's website.

http://www.quantatrisk.com/2014/10/23/garch11-model-in-python/

3.2.6 Notes: Extensions to ARCH Models

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 r_t, 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 σ_t 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

A piece of "bad news", or a negative return, such as a drop in the stock price, reduces the value of equity relative to the fixed value of debt of that firm according to Enders (2014). 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 following 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 γ_1 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 $\ln(h_t)$ 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 up 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.

$\alpha + \beta < 1$ if the returns process is stationary, which is necessary for mean reverting processes.
If $\alpha + \beta = 1$, the return process is a random walk, and the GARCH(1,1) process may be expressed as:

$$\sigma 2t = \omega + (1 - \lambda)\varepsilon 2t - 1 + \lambda\sigma 2t - \lambda = \beta, \qquad 1 \geq \lambda \geq 0$$

IGARCH models are often appropriate in foreign exchange markets.
When $\omega = 0$, the IGARCH model becomes an EWMA (exponentially weighted moving average) model.

NGARCH: non-linear GARCH

$$rt = r - 0.5\sigma 2t + \sigma t\xi t$$

$$\sigma 2t = \omega + \alpha\sigma 2t - 1(\xi t - 1 - \theta - \lambda)2 + \beta\sigma t - 1$$

$$\xi t = \varepsilon t + \lambda$$

NGARCH models are typically not risk-neutral, although local risk-neutrality may exist. Models that are not risk-neutral are incomplete and generally do not have replicating portfolios everywhere available. Hence, hedging may not be possible. Typically, NGARCH models are more complex than the preceding GARCH models in this presentation. For these reasons, NGARCH models are not currently used as extensively in finance as other GARCH models.

High-Frequency GARCH Models

Model frequency is a function of the time interval associated with return observations. So hourly observation is more frequent than daily and daily is more frequent than weekly, and so on.

Most GARCH models in practice assume that unexpected portion of the actual return is conditionally normally distributed. Non-normal models are characterized by fat-tailed distributions for the unexpected portion of the return. tGARCH models develop the unexpected portion of the return according to a student-t distribution. Non-normal models may be developed by modifying the likelihood function used to estimate the model parameters.

High frequency models, especially intra-day models, are much more likely to be more volatile and fat-tailed than lower frequency models.

Kalman filter

Kalman filter theory

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:

1. 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.

2. 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, which in turn is used to improve the forecast of the observed variables. Different versions of the Kalman filter exists, some employing only forward filtering, others applying both forward and backward filtering (i.e. after going from 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, their out-of-sample forecasting ability relative to other models has yielded mixed results.

Equations

Kalman Filter model assumes that the state of a system of time t evolves from the prior state at time $t-1$ according to the equation:

$$x_t = F_t \times x_{t-1} + B_t \times u_t + w_t$$

x_t – state vector containing the terms of interest for the system (volatility of returns) at time t

u_t – the vector containing any control inputs (steering angle for example at physics)

f_t – state transition matrix which applies the effect of each system state parameter at time $t-1$ on the system state at time t (example the position and velocity at time $t-1$ both affect the position at time t)

b_t – the control-input matrix which applies the effect of each control input parameter in the vector u_t on the state vector

w_t – the vector containing the process noise terms for each parameter in the state vector

$$w_t \sim N(0, Q_t)$$

An observation (or measurement) z_t of the true state x_t is made according to:

$$z_t = H_t x_t + v_t,$$

where H_t is the observation model which maps the true state space into the observed space and v_t is the observation noise which is assumed to be zero-mean Gaussian white noise with covariance R_t

$$v_t \sim N(0, R_t).$$

Kalman filter in algorithmic trading

Kalman filter can be used in pairs trading. Let's say two assets are cointegrated (there is a strong connection between them). Example: a pair of Exchange Traded-Funds (ETFs).

You can perform a linear regression between the two ETFs. Determining the hedging ratio can be a problem in this situation. The ratio shows how much of each asset has to be bought or sold and it is not constant over time. Therefore, it is crucial to have a mechanism that automatically adjusts it.

The Kalman filter can be used in this situation. The “true” hedging ratio is treated as an unobserved hidden variable in the state space model. The ratio is estimated considering the “noisy” observations.

References

Akcora C. G. et al. (2018). Bitcoin Risk Modeling with Blockchain Graphs.

Bollerslev, T. (1986) 'Generalized autoregressive conditional heteroskedasticity', *Journal of Econometrics*, 31(3), pp. 307–327. doi: 10.1016/0304-4076(86)90063-1.

Chu J. et al. (2017). GARCH Modelling of Cryptocurrencies. Journal of Risk and Financial Management.

Daroczi G. et al. *Introduction to R for Quantitative Finance*. Packt Publishing, Chapter 1 Time Series Analysis, pages 7-26.

Ding J. (2018). Time Series Predictive Analysis of Bitcoin with ARMA-GARCH model in Python and R.

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.

Hultman, H. (2018). Volatility Forecasting.

Jeet P. and Vats P. Learning Quantitative Finance with R. Packt Publishing, Chapter 4: Time Series Modeling, pages 96-125.

Jiang, W. (2012). Using the GARCH model to analyze and predict the different stock markets.

Letra, I. (2016). What Drives Cryptocurrency Value? A Volatility and Predictability Analysis. Master Thesis.

Rossi, E. (2004). Lecture notes on GARCH models. University of Pavia.

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

Tusell, F. (2011). Kalman Filtering in R, Journal of Statistical Software, Volume 39, Issue 2.

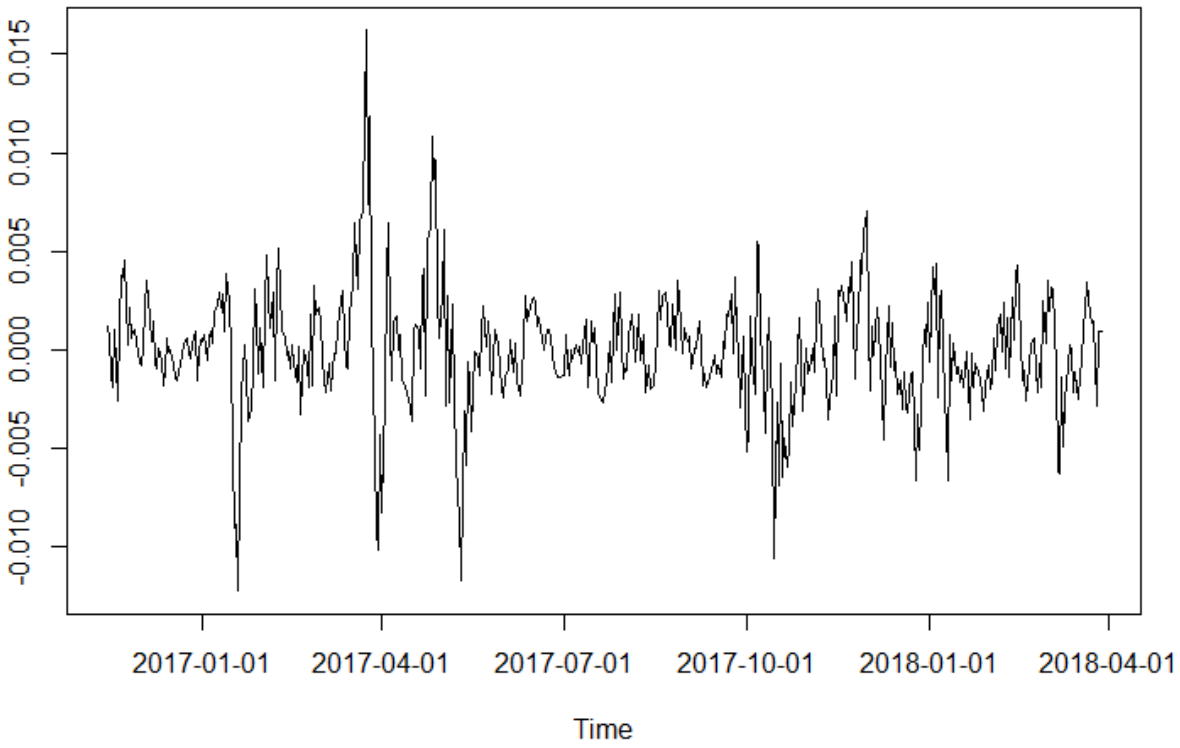
Welch, G. and Bishop G. (2006). An Introduction to Kalman Filter.

Wildi, M. (2013). An Introduction to State Space Models.

3.3 Collaborative Review Task

During each module students will undertake a short case study assignment, which is then marked by their peers according to a grading rubric. The question is provided below.

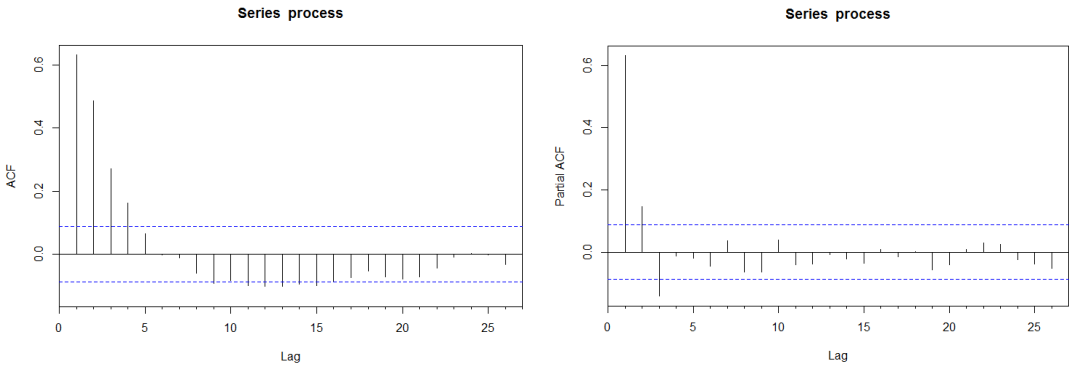
Consider the following plot of a time series process:



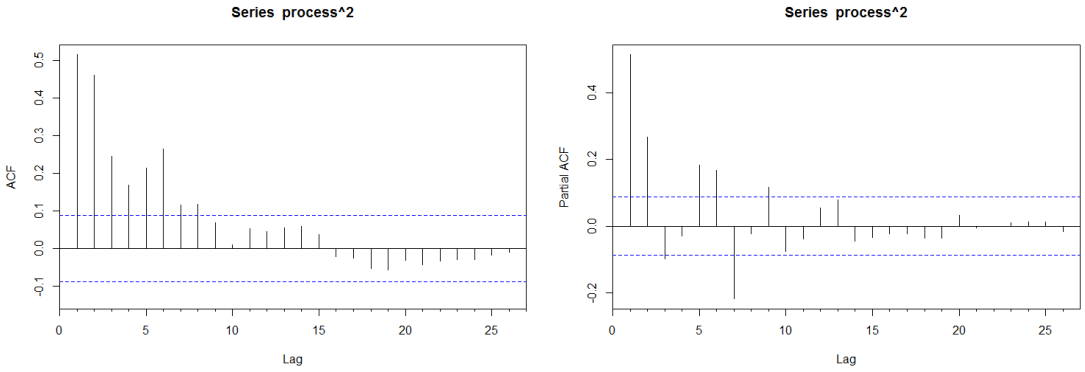
1 Discuss what can be deduced from the time series properties (persistence *and* volatility) of the series from the time series plot.

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:



2 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.

.

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*
	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*

3 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 and come to a conclusion on 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.