Lecture Notes: part 1

Financial Econometrics

2024-2025

Francisco Blasques
and
Paolo Gorgi

Contents

Chapter 1

Introduction

From an econometric methodology perspective, this course is essentially devoted to the art of specifying time-varying parameter models, and using them to conduct inference, probabilistic analysis, policy analysis and forecasting. As we shall see, time-varying parameter models can be divided into two broad categories: *observation-driven* models and *parameter-driven* models. Both classes of models are capable of describing the temporal dynamics of time-series featuring time-varying conditional volatilities, time-varying tail probabilities, time-varying regression coefficients, time-varying conditional moments of higher-order, and much more!

While *observation-driven* models and *parameter-driven* models can be used to describe similar features in the data, they actually approach the data in very different ways and require distinct statistical tools and techniques.

Part 1 of these lecture notes is devoted to the study of *observation-driven* models for conditional volatility. Part 2 is devoted to *parameter-driven* models for volatility and to other extensions. The focus is on the practical implementation and analysis of these models. In the remainder of this introductory chapter we shall use time-series of *financial returns* as a motivation for the use of time-varying parameter models. In particular, financial return data clarifies the need to go beyond models of the conditional mean (like linear regression and ARMA models) and make use of models that can describe time-variation in conditional volatilities. Below, we provide first a quick recap of linear regression and ARMA models. Next, we show the limitations of using these models for analyzing financial returns.

1.1 Models for the conditional mean

In this section, we shall revisit some basic models for the conditional mean: the linear regression model and the autoregressive model.

1.1.1 Linear regression

In your introductory econometrics courses, you have most likely learned about regression. In particular, by now, you should be familiar with the *linear regression model*

$$y_t = \alpha + \beta x_t + \varepsilon_t$$

where y_t is the *dependent* or *endogenous* variable, x_t is the *independent* or *explanatory* variable, ε_t is the *error term* or *innovation*, and α and β are the fixed *unknown parameters* typically called *intercept* and *slope* respectively.

You should also remember that when the error term ε_t satisfies the assumption that $\mathbb{E}(\varepsilon_t|x_t) = 0$, then the linear regression model is a model of the conditional expectation of y_t given x_t. In other words, we have

$$
\begin{aligned}
\mathbb{E}(y_t|x_t) &= \mathbb{E}(\alpha + \beta x_t + \varepsilon_t|x_t) \\
&= \alpha + \beta \underbrace{\mathbb{E}(x_t|x_t)}_{=x_t} + \underbrace{\mathbb{E}(\varepsilon_t|x_t)}_{=0} \\
&= \alpha + \beta x_t.
\end{aligned}
$$

The errors ε_t account for the fact that the relation between y_t and x_t holds only *"on average"*. The linear regression model states essentially that, *on average*, the *dependent variable* y_t is linearly related to the *explanatory variable* x_t.
The parameter β measures the expected change in y_t given a unit change in x_t, and the parameter α measures the expected value of y_t when $x_t = 0$. If the parameters α and β were known, then the average relation between y_t and x_t would also be known, and econometricians would not be needed! Fortunately however, α and β are unknown, and hence, they must be estimated from the data!

1.1.2 AR(1) model

In your introductory econometrics courses you surely learned about time-series models. Most probably, you studied linear time-series models like the *autoregressive model of order 1*, also called the AR(1) model. A sequence $\{x_t\}_{t\in\mathbb{Z}}$ is said to follow an AR(1) process if

$$
x_t = \phi x_{t-1} + \varepsilon_t \qquad \forall\ t \in \mathbb{Z}
$$

where $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ is a *white noise* sequence with $\mathbb{E}(\varepsilon_t|x_{t-1}) = 0$. A *white noise* sequence is a sequence that is serially uncorrelated, has mean zero $\mathbb{E}(\epsilon_t) = 0$, and finite unconditional variance $\mathbb{V}\mathrm{ar}(\epsilon_t) = \sigma^2$.
Linear autoregressive models like the AR(1) are very useful in modeling the temporal dependence that we usually observe in economic and financial time-series. In the AR(1) model, this dependence can be well understood by noting that the conditional expectation of x_t depends on the value of x_{t-1}. In particular, the conditional expectation of x_t given x_{t-1} is $\mathbb{E}(x_t|x_{t-1}) = \phi x_{t-1}$.
You may also remember from your introductory courses that a time series $\{x_t\}_{t\in\mathbb{Z}}$ is weakly stationary if its mean, variance and autocovariance function are invariant in time. Figure 1.1 shows a typical path of a time-series generated by an AR(1) model with time-varying *conditional distribution*, but the multiple paths reveal the time-invariant *unconditional distribution*.

Definition 1.1. (Weak Stationarity) *A time-series $\{x_t\}_{t\in\mathbb{Z}}$ is said to be weakly stationary if the mean $\mathbb{E}(x_t)$ and the variance $\mathbb{V}ar(x_t)$ are constant in t and the autocovariance function $\mathbb{C}ov(x_t, x_{t-h})$ is constant in t, for each h.*

You also learned that the linear Gaussian AR(1) model is weakly stationary as long as $|\phi| < 1$. In other words, the Gaussian AR(1) is stationary as long as it does not exhibit 'too much' temporal dependence.

Theorem 1.1. *Let $\{x_t\}_{t\in\mathbb{Z}}$ be a time-series generated by the linear Gaussian AR(1) model*

$$
x_t = \phi x_{t-1} + \varepsilon_t \qquad \forall\ t \in \mathbb{Z}
$$

with $|\phi| < 1$ and innovations $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ that are white noise. Then $\{x_t\}_{t\in\mathbb{Z}}$ is weakly stationary.

This stationarity property of the time-series is important for understanding the properties of estimators because it allows us to make use of laws of large numbers and central limit theorems.

As you have most likely learned in your introductory time series course, the AR(1) model presented above can be generalized to the class of ARMA models. The AR(1) and ARMA models are useful for the conditional mean in time series.

1.2 Properties of Financial Returns

Models for the conditional mean are useful in many empirical applications, but here we will argue that they are not very useful for modeling financial returns. We will see that stock returns are basically unpredictable

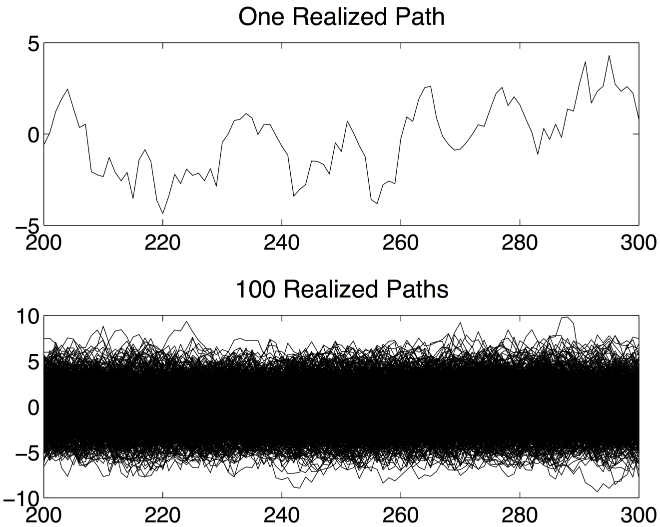

Figure 1.1: Single path [above] shows time-varying conditional mean. Multiple paths [below] show invariance of the distribution (mean and variance are clearly constant over time).

in mean and therefore ARMA models are not of great use. Instead, the variance (or volatility) of stock returns can be predicted. This justifies the need of models that go beyond the conditional mean and account for time-variation in the conditional variance.

1.2.1 Random walk of stock prices

In this section we turn our attention to time-series of stock prices. In particular, we look at the stock prices of the companies listed in the Standard and Poor's top 100 companies in the US; commonly known as S&P100. We denote with p_t the price of a certain stock at time t.

We will argue that the time series $\{p_t\}_{t\in\mathbb{Z}}$ of each individual stock price seems to behave essentially like a *random walk*. We shall say that a time series $\{p_t\}_{t\in\mathbb{Z}}$ follows a *random walk* if we have

$$p_t = p_{t-1} + \epsilon_t,$$

where $\{\epsilon_t\}_{t\in\mathbb{Z}}$ is a *white noise* sequence with $\mathbb{E}(\epsilon_t|p_{t-1}) = 0$. The random walk dynamics imply that stock prices are essentially impossible to forecast. In other words, the best forecast $\hat{p}_{t+1}$ for the price at time $t+1$ conditional on the data until time t is simply given by the last observed price $\hat{p}_{t+1} = p_t$. This is easy to show since

$$\begin{aligned}\hat{p}_{t+1} &= \mathbb{E}(p_{t+1}|p_t)\\ &= \mathbb{E}(p_t + \epsilon_{t+1}|p_t)\\ &= \mathbb{E}(p_t|p_t) + \mathbb{E}(\epsilon_{t+1}|p_t)\\ &= p_t + 0 = p_t.\end{aligned}$$

Below we provide empirical evidence that stock prices essentially behave like random walks by studying stock prices of several stocks. Naturally, if stock prices behave like *random walks*, then we should be able to find evidence that they are unit-root non-stationary. Furthermore, we should also find that variation of stock prices (referred as returns or log-returns) are not only stationary but *white noise*.

First we investigate whether stock prices are unit root non-stationary. Figure 1.2 plots the daily stock prices of Apple and Intel over a period of 10 years, starting in 2006. The figure indeed suggests that stock prices are non-stationary since their mean is not constant over time. The non-stationarity assumption can also be formally tested using a Augmented Dickey Fuller (ADF) unit-root test. Table 1.1 reports the p-value

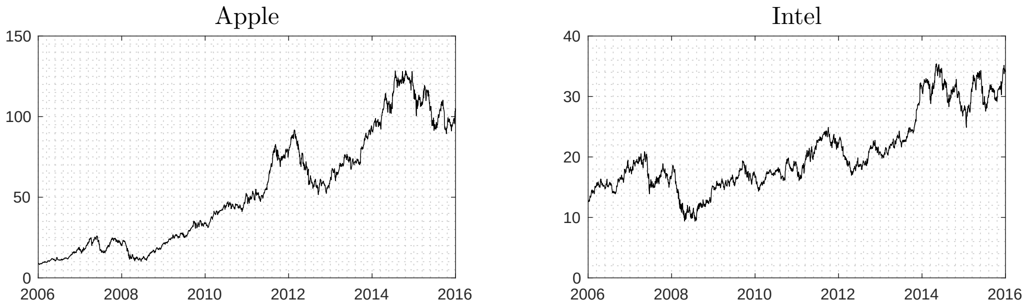

Figure 1.2: Daily stock prices of Apple and Intel from August 2006 to August 2016

of the Augmented Dickey Fuller (ADF) unit-root test applied to the daily, weekly and monthly stock prices of Apple and Intel. We can see that the test suggests that stock prices are indeed non-stationary. Table 1.2 below reports the fraction of times that the null hypothesis of a unit-root is rejected for the prices of all stocks in the S&P100 index. There is overwhelming evidence of non-stationarity in stock prices. The results for each stock in the S&P100 index are can be found in Tables A.1 and A.2 in Appendix A.

Table 1.1: P-values of ADF test for Apple and Intel stock prices

	daily	weekly	monthly
Apple	0.239	0.188	0.230
Intel	0.313	0.356	0.115

Table 1.2: Fraction of H_0 rejections for ADF test over all S&P100 stock prices

daily	weekly	monthly
0.00	0.00	0.00

We now focus our attention on returns (or log returns), i.e. price variations. Studying the properties of returns (or log-returns) is of great interest because we are typically interested in how risky or remunerative a certain investment is. Therefore we are actually more interested in the price variation more than the price level itself. Furthermore, if stock prices are *random walks* then we should find that returns (or log-returns) are *white noise*. In practice we focus on log-returns instead of returns.

Log-returns are defined as first differences of log-prices. In particular, log-returns $\{y_t\}_{t\in\mathbb{Z}}$ are obtained as

$$y_t = \log(p_t) - \log(p_{t-1}) = \log\left(\frac{p_t}{p_{t-1}}\right).$$

We work with first differences of log-prices instead of prices because they have some appealing properties. For instance, log-returns are a good approximation for returns rates

$$y_t = \log\left(\frac{p_t}{p_{t-1}}\right) \approx \frac{p_t - p_{t-1}}{p_{t-1}}.$$

Therefore, if $y_t = 0.01$ we can say that the price from time $t-1$ to t increased of about 1%. Throughout these notes we will always work with log-returns and sometimes for convenience we shall refer to them as simply returns.

Figure 1.3 plots the daily log-returns of Apple and Intel. The figure suggests that the log-returns of these stocks are stationary. In particular, we can see that the mean seems to be constant over time. Table 1.3

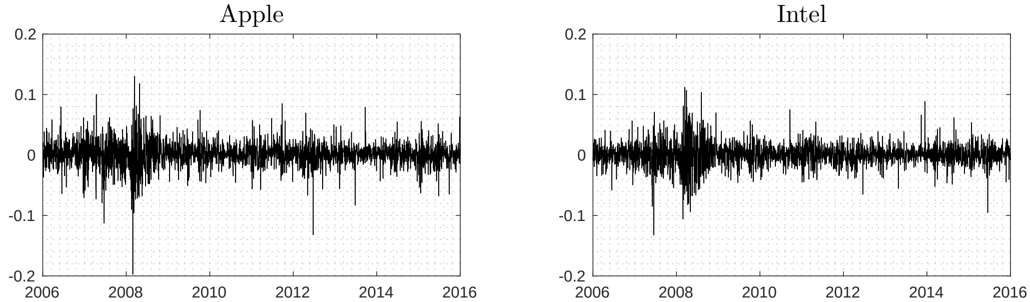

Figure 1.3: Daily log-returns of Apple and Intel from August 2006 to August 2016

reports the results of the ADF unit-root test applied to the daily, weekly and monthly log-returns of Apple and Intel. Shaded values are significant at the 95% confidence level. We can see that the test suggests that log-returns are stationary. Table 1.4 below reports the fraction of times the null hypothesis of a unit-root is rejected for the log-returns of all stocks in the S&P100 index. The results further confirm the stationarity of log-returns.

Table 1.3: P-values of ADF test for Apple and Intel log-returns

	daily	weekly	monthly
Apple	0.001	0.001	0.001
Intel	0.001	0.001	0.001

Table 1.4: Fraction of H_0 rejections for ADF test over all S&P100 stocks

daily	weekly	monthly
1.00	0.99	0.98

Having established the non-stationarity of *stock prices* and the stationarity of *stock returns*, we now move further and investigate whether log-returns are *white noise*. If log-returns are *white noise* we should find that they are uncorrelated. Figure 1.4 shows the estimated autocorrelation function for the daily stock returns of Apple and Intel, ranging over 25 lags. The significance bounds (in red) reveal that there is little evidence autocorrelation in the stock returns of Apple and Intel at the daily frequency. The evidence for temporal dependence is even weaker at lower frequencies. Figure 1.5 plots the sample autocorrelation for weekly returns. There is no evidence of autocorrelation of weekly log-returns.

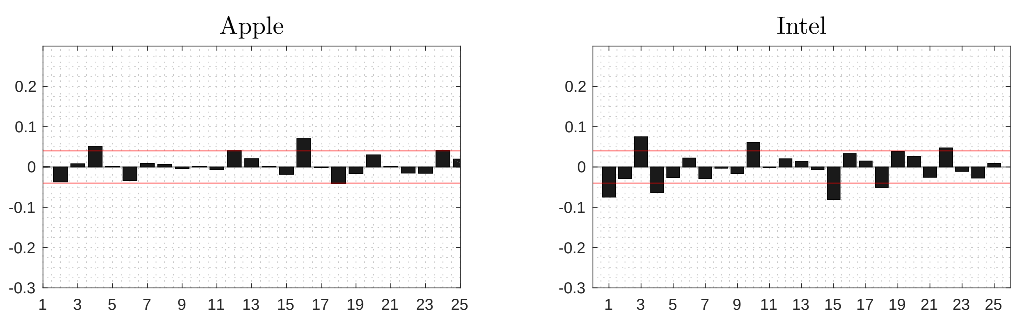

Figure 1.4: Sample ACF for daily log-returns of Apple and Intel

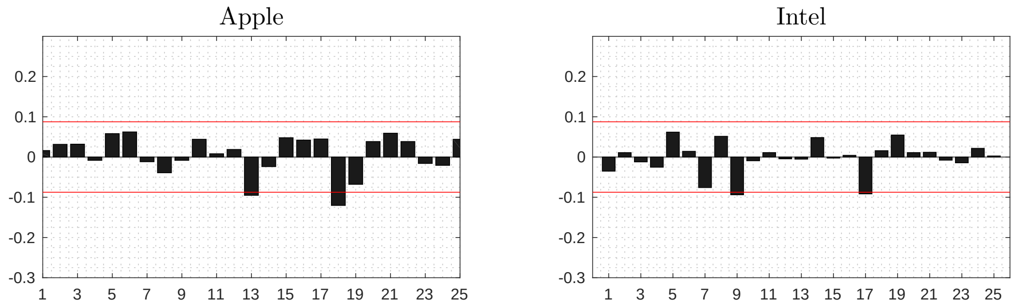

Figure 1.5: Sample ACF for weekly log-returns of Apple and Intel

Table 1.5 reports the estimated coefficient of an MA(1) model and the estimated coefficient of an AR(1) model for Apple and Intel log-returns, measured at the daily, weekly, and monthly frequencies. Shaded values are significant at the 5% confidence level. We can see that only the coefficients for daily log-returns are significantly different from zero. Table 1.6 further confirms that the presence of autocorrelation in stock returns depends to a large extent on the frequency at which returns are observed. In particular, this table reports the frequency with which the MA and AR coefficients are found to be statistically significant at the 5% level, over all the stocks in the S&P100 index. The results for each individual stock in the S&P100 index are reported in Tables A.3 and A.4 in Appendix A.

Table 1.5: Estimates of MA(1) and AR(1) coefficients for Apple and Intel log-returns

	MA(1) daily	AR(1) daily	MA(1) weekly	AR(1) weekly	MA(1) monthly	AR(1) monthly
Apple	-0.026	-0.026	0.040	0.037	0.035	0.036
Intel	-0.044	-0.042	-0.040	-0.038	-0.038	-0.049

Table 1.6: Fraction of significant coefficients (5% level) over all the S&P100 log-returns

MA(1) daily	AR(1) daily	MA(1) weekly	AR(1) weekly	MA(1) monthly	AR(1) monthly
0.6337	0.6238	0.4356	0.4059	0.1584	0.1287

Overall, it seems fair to say that there is evidence of significant but weak autocorrelation in log-returns at the daily frequency. So daily stock log-returns are not exactly white noise but the autocorrelation is basically negligible and therefore we can essentially consider daily log-returns as white noise. At the weekly and monthly frequencies there is not much evidence of autocorrelation in log-returns and therefore the white noise assumption is well suited in these cases.

1.2.2 Volatility clustering

From the discussion in the previous paragraph, we can conclude that ARMA models, and more in general models for the conditional mean, are not very useful to describe log-returns. Does this mean that log-returns cannot be predicted? Well, the mean seems to be unpredictable but if we a closer look at Figure 1.3 we can notice that there are time periods where the variability of the log-returns his higher and periods where it is lower. For instance, we can see that around 2008 the volatility (variability) of log-returns is higher. This changes of variability are well known as volatility clusters. In the following we see that past log-returns can be useful to predict the volatility of future log-returns. Predicting volatility is of key importance since volatility is one of the most important measures of financial risk.

Now instead of analyzing log-returns we consider squared log-returns, i.e. the square of log-returns. Given that log-returns have a mean of approximately zero, squared log-returns offer a natural indicator of scale. As such, the clusters of volatility may reveal themselves through autocorrelation in squared log-returns. Figure 1.6 plots the daily squared log-returns of Apple and Intel. The figure shows that squared log-returns tend

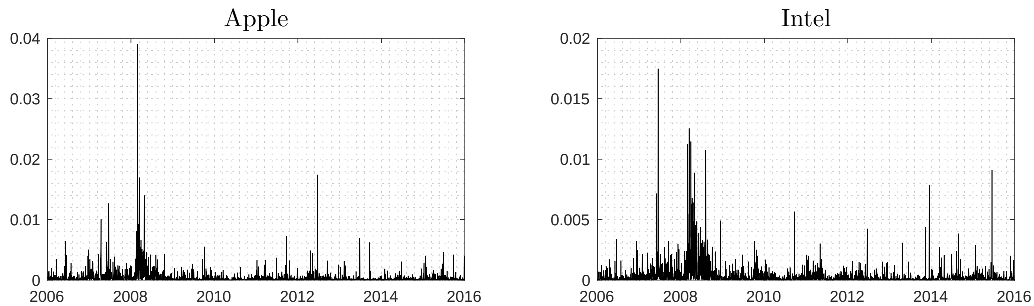

Figure 1.6: Daily squared log-returns of Apple and Intel stocks.

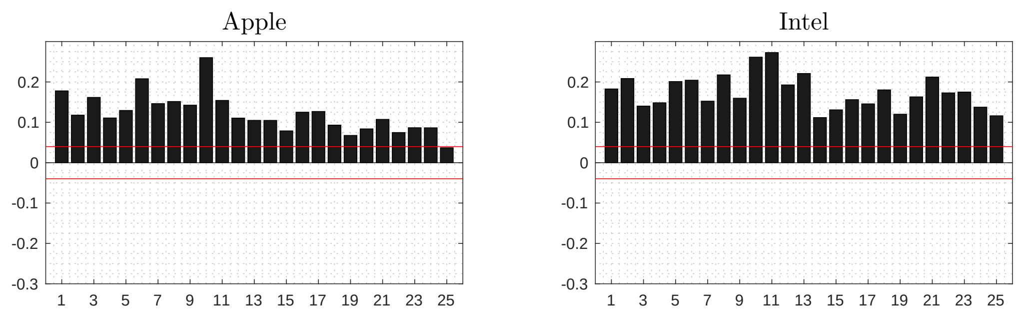

Figure 1.7: Sample ACF for the squared daily log-returns of Apple and Intel.

to be higher during the financial crisis in 2008. Figure 1.7 reports the autocorrelation function of the squared log-returns. The figure provides strong evidence of autocorrelation in squared log-returns for both Apple and Intel. The temporal dependence of squared log-returns is also made clear in Table 1.7. The table reports the MA(1) and AR(1) coefficient estimates for the daily squared log-returns of Apple and Intel. Similar results are obtained for other stocks in the S&P100 index as reported in tables A.7 and A.8 in Appendix A. These results indicate the need of developing models that are able to describe the volatility clustering and capture autocorrelation of financial squared log-returns.

Table 1.7: Estimates of MA(1) and AR(1) coefficients for daily squared log-returns

	MA(1)	AR(1)
Apple	0.184	0.178
Intel	0.189	0.183

Finally, we investigate other features of log-returns. Table 1.8 below suggests also that we will need to use models capable of explaining non-Gaussian fat-tailed data. Large sample kurtosis (larger than 4) suggests that the density of stock returns has tails that are fatter than those of a normal density. The Jarque-Bera test statistic also suggests the rejection of the null hypothesis of Gaussian returns at a 5% significance level. The evidence for the non-Gaussianity of stock returns is stronger at higher frequencies (e.g. for daily returns) and weaker at lower frequencies (e.g. for weekly and monthly returns). Tables A.5 and A.6 provide similar results for each of the stocks in the S&P100 index.

Table 1.8: Estimated moments and p-value of the Jarque-Bera test for Intel and Apple log-returns

Stock	Mean	Var	Skew	Kurt	JB
Apple	0.006	0.036	-4.979	47.712	0.001
Intel	-0.003	0.017	-1.873	10.394	0.001

The evidence for strong temporal dependence in squared stock returns will force us to go beyond models of the conditional expectation. The evidence for fat-tailed data also requires us to consider alternative models. In general, the linear-Gaussian regression models that were useful to us in the past are no longer up to the task. The ARMA models, autoregressive distributed lag (ADL) models, or error correction models (ECM) that you studied in your introductory time-series course, are not suitable to address the problems at hand. In the coming chapters we shall explore uncharted territory! Together, we will design and study models that can explain changes in volatilities, correlations, and other features in the data!

Part I

Observation Driven Models

Chapter 2

Autoregressive Conditional Heteroskedasticity Models

In this first part of the course, we study a class of models that are capable of describing time-series that are uncorrelated over time but exhibit time-varying conditional volatility. These models are especially well suited to describe the dynamics of financial returns.

In Chapter 2, we introduce the *Autoregressive Conditional Heteroskedaticity* (ARCH) model. As you may recall from your introductory econometrics courses, the term *heteroskedaticity* refers to the variance not being constant. In contrast, a time-series that is *homoeskedastic* is a time-series with constant fixed variance. In this section, we will often talk about time-series with time-varying *volatility* rather than time-varying *variance* since the *variance* may not always exist.[1]

In Chapter 3 we also look at a more general model called *Generalized Autoregressive Conditional Heteroschedaticity* (GARCH) model. Finally, in Chapter 6 we extend our models to the multivariate setting. Multivariate models are capable of explaining not only the time-varying conditional volatilities, but also, the time-varying conditional correlations of multiple financial returns.

These models, both univariate and multivariate, are often called *time-varying parameter* models. Furthermore, in this chapter, all models belong to the class of *observation-driven* models. As we shall see, these time-varying parameter models are said to be *observation-driven* since past observations are used to update the values of the unobserved time-varying parameter.

2.1 The ARCH(1) model

Consider a sequence of financial returns $\{y_1, y_2, y_3, ...\}$. The *Autoregressive Conditional Heteroschedaticity* (ARCH) model describes the dynamics of returns as

$$y_t = \sigma_t \varepsilon_t \tag{2.1}$$

where σ_t is the conditional volatility at time t, and ε_t is an independent and identically distributed sequence of shocks with mean zero and unit variance $\{\varepsilon_t\}_{t\in\mathbb{Z}} \sim \text{NID}(0,1)$.

In econometrics, equation (2.1) is called the *observation-equation.* This equation tells us how each *observed* financial return y_t is obtained from the *unobserved* conditional volatility σ_t and the *unobserved* shocks ε_t.

In addition to the observation equation stated above, we need an equation that tells us how the conditional volatility σ_t evolves over time. In the case of the first-order ARCH model, labeled ARCH(1), the parameter *updating equation* is given by

$$\sigma_t^2 = \omega + \alpha_1 y_{t-1}^2, \qquad \forall\ t \in \mathbb{Z} \tag{2.2}$$

[1]Recall from your introductory probability courses that a random variable with fat tails (e.g. a Cauchy random variable, or a student-t random variable with 2 degrees of freedom) may not have a finite variance.

where $\omega > 0$ and $\alpha_1 \geq 0$ are parameters that determine the behavior of the conditional volatility.

The idea behind the ARCH(1) updating equation is to capture time variation in the variance and, in this way, describe the "volatility clustering" that is typically observed in stock returns. The squared observation y_{t-1}^2 can be seen as an estimate of the variance at time $t-1$. When y_{t-1}^2 is large, then σ_t^2 also tends to be large (for positive α_1). Therefore, through the observation equation, y_t^2 is more likely to be large as well. As a result, large (small) values of the variance at time $t-1$ are likely to produce large (small) values of the variance at time t. This exactly reflects the "volatility clustering" mentioned above. As we shall see in the following, the variance is time varying only conditional on the past (for this reason ARCH models are said to be "Conditional Heteroschedastic"). The marginal variance of the ARCH model is not time varying.

Definition 2.1. ARCH(1) Model: *The ARCH(1) model is given by*

$$y_t = \sigma_t \varepsilon_t, \quad \sigma_t^2 = \omega + \alpha_1 y_{t-1}^2, \qquad \forall\ t \in \mathbb{Z} \tag{2.3}$$

where $\omega > 0$ and $\alpha_1 \geq 0$ are parameters to be estimated, $\{\varepsilon_t\}_{t \in \mathbb{Z}}$ is an exogenous $NID(0,1)$ sequence.

Remark 2.1. *The parameters ω and α_1 are constrained to be non-negative to ensure that σ_t^2 is positive. Furthermore, as we shall see later, ω is strictly positive to guarantee that the unconditional variance is non-zero.*

Let us now analyze carefully the properties of this model. In particular, we are interested in verifying if the ARCH(1) is capable of describing the main features of financial returns. Theorem 2.1 shows that, conditional on past returns $Y^{t-1} = \{y_{t-1}, y_{t-2}, y_{t-3}, ...\}$, the distribution of y_t is Gaussian with mean zero and variance σ_t^2. As such, y_t has a conditional variance σ_t^2 that is time-varying.

Theorem 2.1. *The conditional distribution of y_t given the past Y^{t-1} is normal with mean $\mathbb{E}(y_t|Y^{t-1}) = 0$ and variance $\mathbb{V}ar(y_t|Y^{t-1}) = \sigma_t^2$, namely $y_t|Y^{t-1} \sim N(0, \sigma_t^2)$.*

Proof. The conditional mean is obtained as

$$\mathbb{E}(y_t|Y^{t-1}) = \mathbb{E}(\sigma_t \varepsilon_t|Y^{t-1}) = \sigma_t \mathbb{E}(\varepsilon_t|Y^{t-1}) = \sigma_t \mathbb{E}(\varepsilon_t) = \sigma_t \cdot 0 = 0,$$

where the second equality follows because σ_t is a constant conditional on Y^{t-1} and the third equality follows from the independence of ε_t and the past Y^{t-1}. Similarly, the conditional variance is obtained as

$$\mathbb{V}ar(y_t|Y^{t-1}) = \mathbb{V}ar(\sigma_t \epsilon_t|Y^{t-1}) = \mathbb{E}(\sigma_t^2 \epsilon_t^2|Y^{t-1}) = \sigma_t^2 \mathbb{E}(\epsilon_t^2|Y^{t-1}) = \sigma_t^2 \mathbb{E}(\epsilon_t^2) = \sigma_t^2 \cdot 1 = \sigma_t^2,$$

where the third equality follows because σ_t^2 is a constant conditional on Y^{t-1} and the fourth equality follows from the independence of ε_t and the past Y^{t-1}. Finally, we have to show that the conditional distribution of $y_t = \sigma_t \varepsilon_t$ given Y^{t-1} is normally distributed. This can be noted from the fact that, conditional on Y^{t-1}, the factor σ_t is a constant and $\varepsilon_t \sim N(0,1)$. Therefore, since a Normal random variable multiplied by a constant is normal as well, we can immediately conclude that $y_t|Y^{t-1}$ is normal. ■

Theorem 2.1 is important because it tells us the conditional distribution of y_t given the past. Therefore we can use this result to calculate the probability of extreme events conditional on the recent behavior of the market.

Figure 2.1 plots the time-series and the conditional volatility σ_t simulated from an ARCH(1) model for different values of ω and α_1. It is clear that the ARCH(1) model is capable of generating clusters of volatility. Furthermore, larger values of α_1 are responsible for a stronger clustering behavior of the conditional variance. Figure 2.2 shows that α_1 determines the autocorrelation in squared returns y_t^2 but not in the returns themselves.
Theorem 2.2 complements Figure 2.2 by showing that the returns y_t generated by the ARCH(1) model are indeed uncorrelated.

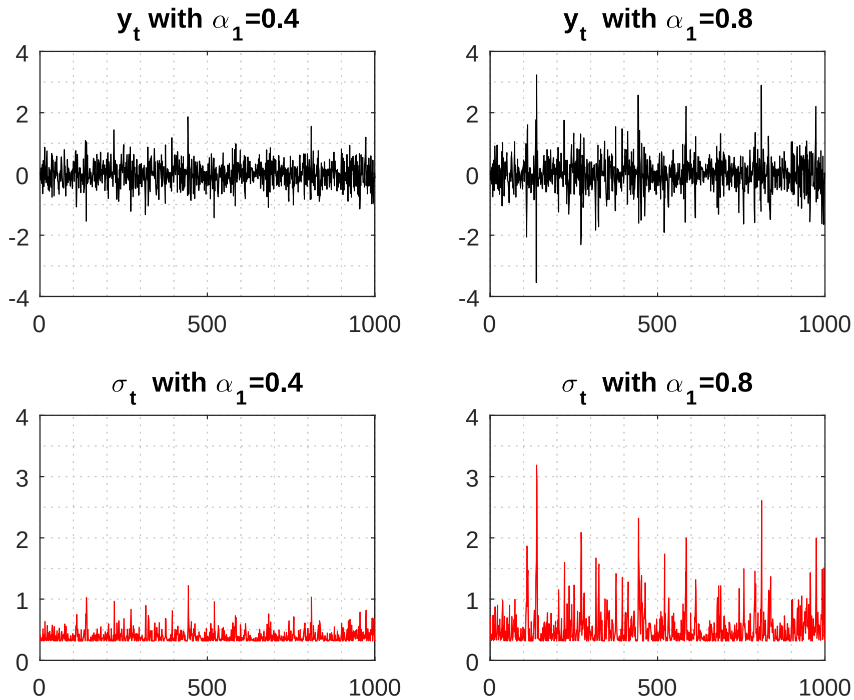

Figure 2.1: Path of returns (black line) and volatility (red lines) simulated from an ARCH(1) model with $(\omega, \alpha_1) = (0.1, 0.4)$ (left graphs) and $(\omega, \alpha_1) = (0.1, 0.8)$ (right graphs).

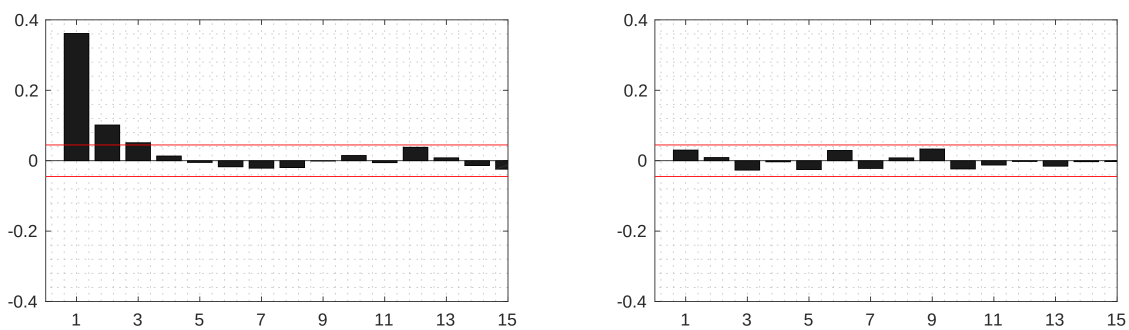

Figure 2.2: Sample ACF for squared returns (left graph) and returns (right graph) obtained from a sample of size $T = 2000$ observations simulated from an ARCH(1) model with $(\omega, \alpha_1) = (0.1, 0.4)$.

Theorem 2.2. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an ARCH(1) model have zero autocovariance at any lag. Hence they are uncorrelated over time.*

Proof. The autocovariance function for any $l > 0$ is given by

$$\mathbb{C}ov(y_t, y_{t-l}) = \mathbb{E}(y_t y_{t-l}) = \mathbb{E}(\mathbb{E}(y_t y_{t-l}|Y^{t-1})) = \mathbb{E}(y_{t-l}\mathbb{E}(y_t|Y^{t-1})) = \mathbb{E}(y_{t-l}\cdot 0) = 0,$$

where the second equality follows by the law of total expectation and the third equality follows from the fact that conditional on Y^{t-1} we have that y_{t-l} is a constant for any $l > 0$. Given that all autocovariances are zero, then the autocorrelation is also zero at any lag. ∎

While the ARCH(1) model is capable of generating a time-varying *conditional* variance, this does not mean that the *unconditional* variance of returns $\{y_t\}_{t\in\mathbb{Z}}$ defined by the ARCH(1) changes over time. In fact, it is possible to show that the unconditional distribution of $\{y_t\}_{t\in\mathbb{Z}}$ is invariant in time when $\alpha_1 < 1$. In what follows we show that the unconditional mean, variance and autocovariances of returns generated by an ARCH(1) model are all finite and time-invariant, so that $\{y_t\}_{t\in\mathbb{Z}}$ is weakly stationary.

We have already established in Theorem 2.2 that the unconditional autocovariances are all equal to zero at any lag. Theorem 2.3 shows that also the unconditional mean is invariant in time and equal to zero.

Theorem 2.3. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an ARCH(1) model with $\alpha_1 < 1$ have unconditional mean zero.*

Proof. We know that the conditional mean $\mathbb{E}(y_t|Y^{t-1})$ is equal to zero. Therefore we obtain that

$$\mathbb{E}(y_t) = \mathbb{E}(\mathbb{E}(y_t|Y^{t-1})) = \mathbb{E}(0) = 0,$$

by an application of the law of total expectation. ■

Finally, we turn to the unconditional variance of the returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an ARCH(1) model. We establish that the unconditional variance is finite and time-invariant in three steps. First, we show that the squared observations y_t^2 of an ARCH(1) model follow an AR(1) model. This is known as the AR(1) representation of the ARCH(1) model. This result is useful to show that when $\alpha_1 < 1$ the unconditional variance of y_t exists and also to find its value.

Theorem 2.4. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(1) model. Then $\{y_t^2\}_{t\in\mathbb{Z}}$ follows an AR(1) model*

$$y_t^2 = \omega + \alpha_1 y_{t-1}^2 + \eta_t$$

where $\{\eta_t\}_{t\in\mathbb{Z}}$ is a white noise sequence.

Proof. Define first a new error term η_t as $\eta_t = y_t^2 - \sigma_t^2$. It can be shown that $\{\eta_t\}_{t\in\mathbb{Z}}$ is a white noise sequence. Using this new definition, we go ahead and substitute σ_t^2 for $y_t^2 - \eta_t$ in the updating equation for σ_t^2. This yields

$$y_t^2 - \eta_t = \omega + \alpha y_{t-1}^2.$$

which is naturally equivalent to

$$y_t^2 = \omega + \alpha_1 y_{t-1}^2 + \eta_t.$$

Therefore we conclude that $\{y_t^2\}_{t\in\mathbb{Z}}$ follows an AR(1) process. ■

Naturally you may ask: *why is the AR representation useful?* Well, the AR representation turns out to be useful for model selection. It is common practice in empirical applications to obtain the autocorrelation functions of the squared observations. The AR(1) representation tells us that the squared observation of an ARCH(1) model should have an exponentially decreasing ACF and only the first lag of the PACF should be different from zero.

Another advantage of the AR representation of squared returns is that you can use your knowledge of time-series econometrics to obtain the unconditional variance of y_t when $\alpha_1 < 1$. This result is shown in Theorem 2.5.

Theorem 2.5. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(1) model. If $\alpha_1 < 1$, then the unconditional variance of y_t is time-invariant and, in particular, given by $\mathbb{V}ar(y_t) = \omega/(1-\alpha_1)$.*

Proof. First, we note that

$$\mathbb{V}\text{ar}(y_t) \;=\; \mathbb{E}(y_t^2).$$

Then, we can unfold the AR(1) representation of Theorem 2.4 and obtain

$$y_t^2 = \sum_{i=0}^{\infty} \alpha_1^i \omega + \sum_{i=0}^{\infty} \alpha_1^i \eta_{t-i} = \omega/(1-\alpha_1) + \sum_{i=0}^{\infty} \alpha_1^i \eta_{t-i}.$$

Therefore, since we know that $\mathbb{E}(\eta_t) = 0$ for any t we can conclude that

$$\mathbb{E}(y_t^2) = \omega/(1-\alpha_1) + \sum_{i=0}^{\infty} \alpha_1^i \mathbb{E}(\eta_{t-i}) = \omega/(1-\alpha_1).$$

■

Having established in Theorems 2.2, 2.3 and 2.5 that the returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an ARCH(1) model have zero unconditional mean, fixed unconditional variance, and are uncorrelated at any lag, we are now in a position to conclude that $\{y_t\}_{t\in\mathbb{Z}}$ is a *weakly stationary white noise* sequence[2].

Corollary 2.1. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(1) model with $\alpha_1 < 1$. Then, $\{y_t\}_{t\in\mathbb{Z}}$ is a weakly stationary white noise sequence.*

Remark 2.2. *The condition $\alpha_1 < 1$ is also sufficient for strict stationarity but not necessary.*

It is important to note that the ARCH(1) model is not only capable of describing the clusters of volatility observed in financial returns. It can also generate the fat tails (e.g. large kurtosis) observed in their unconditional sample distribution. Indeed, while the conditional distribution of y_t given the past Y^{t-1} is Gaussian, the unconditional distribution of y_t is non-Gaussian. Figure 2.3 plots the unconditional distribution of returns y_t generated by an ARCH(1) model. The plots shows that the distribution of y_t has fatter tails than a Normal distribution.

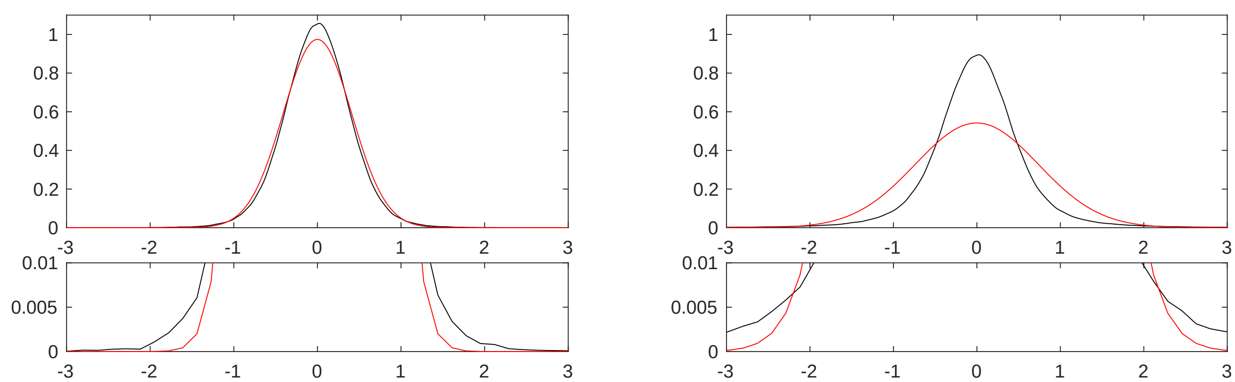

Figure 2.3: Unconditional density of data simulated from an ARCH model with $(\omega, \alpha) = (0.1, 0.4)$ (left graph) and $(\omega, \alpha) = (0.1, 0.8)$ (right graph). The two bottom figures provide a 'zoom in' on the tails of each density.

Curiously there is not a closed form analytic expression for the unconditional probability density function of y_t. Nonetheless, some properties of the unconditional distribution can be derived from higher order moments. Theorem 2.6 derives the Kurtosis is an indicator of how fat are the tails of a probability distribution, namely how likely extreme observations are. The Kurtosis of the unconditional distribution of the ARCH model is larger than 3[3]. This is coherent with what we typically observe in stock returns, namely a sample Kurtosis larger than 3.

[2] In introductory time-series courses you have focused on models of the conditional mean (e.g. ARMA) models. When modeling the conditional mean, a *white noise* sequence is seen essentially as *unstructured noise*. In contrast, when focusing on modeling higher-order moments (like the conditional variance), a *white noise* sequence may still contain structure to be explored. The aim of ARCH models is to exploit this information to predict volatility.

[3] The Kurtosis of the Normal distribution is equal to 3

Theorem 2.6. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(1) model with $\alpha_1 < \frac{1}{\sqrt{3}}$. Then the kurtosis of y_t is given by*

$$k_u = \frac{\mathbb{E}(y_t^4)}{\mathbb{E}(y_t^2)^2} = \frac{3(1-\alpha_1^2)}{1-3\alpha_1^2}.$$

Proof. We will not discuss the details of this result. ■

2.2 The ARCH(q) model

Very often, the conditional variance of stock returns shows strong persistence over time. This can be noted from the autocorrelation functions of the squared observations. Consider, for example, the autocorrelation function of the squared log-returns of the S&P 500 stock index in Figure 2.2. Clearly, an ARCH(1) model will not be appropriate to model this type of dependence since the ACF does not decay exponentially.

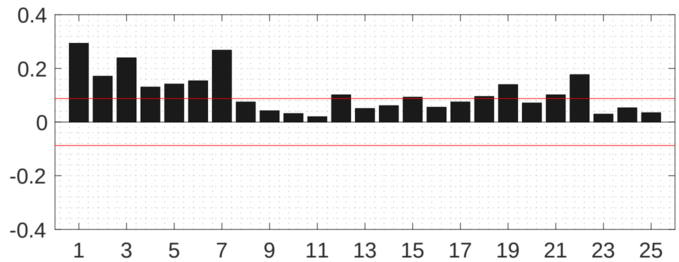

Figure 2.4: ACF with 95% confidence intervals, obtained from daily squared log-returns of the S&P 500 stock index

Luckily, we can take this empirical evidence into account, by including more lags of y_t^2 in the updating equation for the conditional variance σ_t^2. When the updating equation for σ_t^2 depends not only on y_{t-1}^2, but also, on y_{t-2}^2, then the resulting model is called an *ARCH model of order* 2, or ARCH(2). The ARCH(2) model is given by

$$y_t = \sigma_t \epsilon_t \ , \quad \{\varepsilon_t\}_{t\in\mathbb{Z}} \sim \text{NID}(0,1) \ ,$$
$$\sigma_t^2 = \omega + \alpha_1 y_{t-1}^2 + \alpha_2 y_{t-2}^2, \qquad \forall\ t \in \mathbb{Z}$$

where $\omega > 0$, $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$. In general, one may include as many lags of y_t^2 as desired for achieving a correct description of the temporal dynamics of squared returns. When q lags are used, the resulting model is called the *ARCH model of order* q, or ARCH(q) model.

Definition 2.2. ARCH(q) Model: *The ARCH(q) model is given by*

$$y_t = \sigma_t \epsilon_t \ , \quad \{\varepsilon_t\}_{t\in\mathbb{Z}} \sim NID(0,1) \ ,$$
$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2, \qquad \forall\ t \in \mathbb{Z}$$

where $\omega > 0$, $\alpha_1 \geq 0$, ..., $\alpha_p \geq 0$ are strictly positive parameters.

In the same way as for the ARCH(1) model, it can be shown that the conditional distribution of the $y_t|Y^{t-1}$ generated by an ARCH(q) model is normal with mean zero and variance σ_t^2.

Lemma 2.1. *The conditional distribution of y_t given the past Y^{t-1} is normal with mean $\mathbb{E}(y_t|Y^{t-1}) = 0$ and variance $\mathbb{V}ar(y_t|Y^{t-1}) = \sigma_t^2$, namely $y_t|Y^{t-1} \sim N(0, \sigma_t^2)$.*

We have already seen that the ARCH(1) model can be re-written as an AR(1) model for the squared returns y_t^2. As we shall now see, a similar representation holds for the ARCH(q) model. In particular, the squared returns of an ARCH(q) model follow an AR(q) process.

Theorem 2.7. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(q) model. Then $\{y_t^2\}_{t\in\mathbb{Z}}$ follows an AR(q) model*

$$y_t^2 = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \eta_t$$

where $\{\eta_t\}_{t\in\mathbb{Z}}$ is a white noise sequence.

Proof. The proof of this theorem is left as an exercise. ■

The AR(q) representation for the squared returns of an ARCH(q) model, tells us that the ARCH(q) is capable of generating arbitrary structures for the first q lags of the autocorrelation function. The exponential decay of the ACF, with seasonal patterns, occurs only after q lags. For instance, for the S&P 500 series we could choose $q = 4$.

The AR(q) representation for the squared returns of an ARCH(q) also allows us to easily establish conditions for the weak stationarity of the sequence $\{y_t^2\}_{t\in\mathbb{Z}}$. In particular, as you may recall from your introductory time series courses, a necessary and sufficient condition for the weak stationarity of the $\{y_t^2\}_{t\in\mathbb{Z}}$ generated by an AR(q) is that $\sum_{i=1}^{q} \alpha_i < 1$.

Corollary 2.2. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by an ARCH(q) model satisfying $\sum_{i=1}^{q} \alpha_i < 1$. Then $\{y_t^2\}_{t\in\mathbb{Z}}$ is weakly stationary.*

Theorem 2.8. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by the ARCH(q) model satisfying $\sum_{i=1}^{q} \alpha_i < 1$. Then, $\{y_t\}_{t\in\mathbb{Z}}$ is a weakly stationary white noise sequence with $\mathbb{E}(y_t) = 0$, $\mathbb{V}ar(y_t) = \omega/(1 - \sum_{i=1}^{q} \alpha_i)$ and $\mathbb{C}ov(y_t, y_{t-l}) = 0$ for any $l \neq 0$.*

Proof. The derivation of the mean and autocovariances follows the exact same argument as for the ARCH(1) model. The derivations are identical because they involve only the observation equation (which is the same for both models), and the fact that, conditional on Y^{t-1}, the conditional variance σ_t^2 is given; which holds for both ARCH(1) and ARCH(q) models.

The derivation of the unconditional variance uses again the AR representation of the ARCH model. First, we make use of the observation equation to conclude that

$$\mathbb{V}\mathrm{ar}(y_t) \;=\; \mathbb{E}(y_t^2).$$

Next, we use the AR(q) representation of Theorem 2.7 to conclude that

$$\mathbb{E}(y_t^2) = \omega/(1 - \sum_{i=1}^{q} \alpha_i).$$

■

Chapter 3

Generalized ARCH Models

3.1 The GARCH(1,1) model

The specification of ARCH models with several lags is useful for describing the strong dynamics of squared returns. However, in practice, this is not the most parsimonious model specification available. In particular, when squared returns exhibit very strong dependence (or very long memory), then the use of an ARCH(q) model would require a large order q and therefore many parameters to be estimated. One way of substantially increasing the temporal dependence in squared returns requiring only a few additional parameters is to use lags of σ_t^2 in the updating equation. For example, when one lag of σ_t^2 and one lag of y_t^2 is used in the updating equation, we obtain the so-called *generalized autoregressive conditional heteroeskedasticity model* of order (1,1), or GARCH(1,1).

Definition 3.1. GARCH(1,1) Model: *The GARCH(1,1) model is given by*

$$\begin{aligned} y_t &= \sigma_t \epsilon_t \ , \\ \sigma_t^2 &= \omega + \beta_1 \sigma_{t-1}^2 + \alpha_1 y_{t-1}^2 \ , \qquad \forall\ t \in \mathbb{Z} \end{aligned}$$

where $\omega > 0$, $\alpha_1 \geq 0$, $\beta_1 \geq 0$ are parameters, $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ is an $NID(0,1)$ sequence and ε_t is independent of the past Y^{t-1}.

In the same way as discussed for the ARCH(1) model, it can be shown that $y_t|Y^{t-1}$ has a normal distribution with mean zero and variance σ_t^2. The proof of Lemma 3.1 below is equal to that of Theorem 2.1.

Lemma 3.1. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by a GARCH(1,1) model. The conditional distribution of y_t given the past Y^{t-1} is normal with mean $\mathbb{E}(y_t|Y^{t-1}) = 0$ and variance $\mathbb{V}ar(y_t|Y^{t-1}) = \sigma_t^2$, namely $y_t|Y^{t-1} \sim N(0, \sigma_t^2)$.*

Figure 3.1 shows that GARCH models are capable of modeling extreme changes in the conditional volatility of returns, giving rise to strong clusters of volatility.
It is easy to show that the returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by the GARCH(1,1) model are uncorrelated and have mean zero, just as we did for the ARCH(1) and ARCH(q) models. In particular, the proofs are identical because they involve only the observation equation (which is the same for both models), and the fact that, conditional on Y^{t-1}, the conditional variance σ_t^2 is given (which holds for both ARCH and GARCH models).

Lemma 3.2. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(1,1) model have zero autocovariance at any lag and are thus uncorrelated over time.*

Lemma 3.3. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(1,1) model have unconditional mean zero.*

Once again, we can provide an ARMA representation for the squared returns of this model. In particular, the squared observations generated by a GARCH(1,1) model can be shown to admit an *autoregressive moving average* model of order (1,1), or ARMA(1,1), representation.

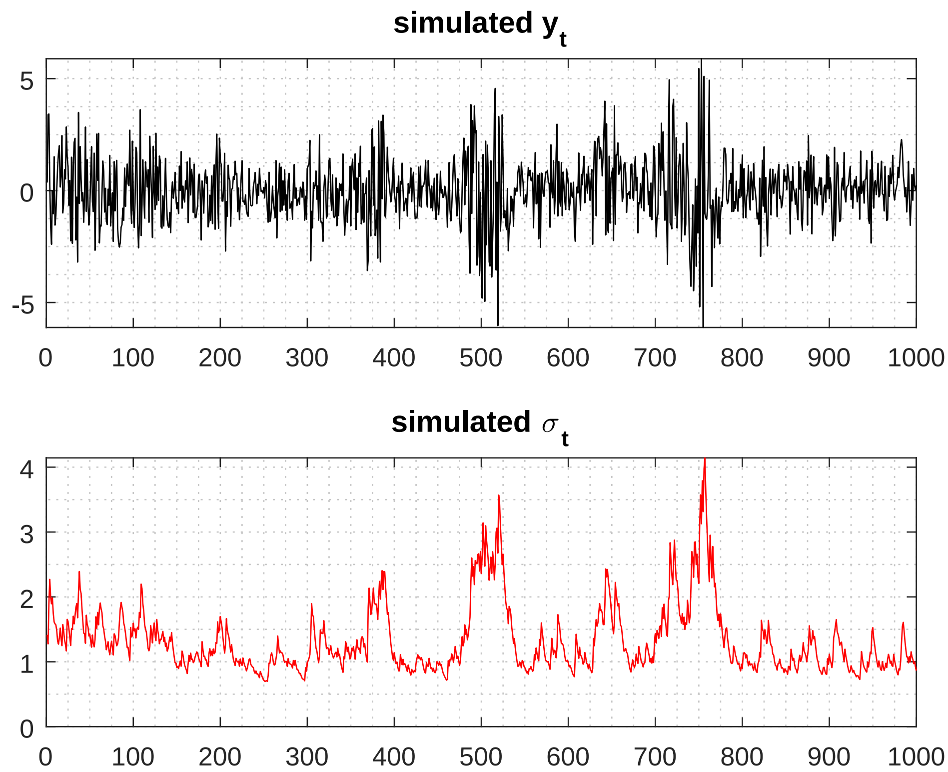

Figure 3.1: Sample path of returns $\{y_t\}_{t=1}^T$ and volatility $\{\sigma_t\}_{t=1}^T$ generated from a GARCH(1,1) model with $(\omega, \beta_1, \alpha_1) = (0.1, 0.75, 0.2)$.

Lemma 3.4. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by a GARCH(1,1) model. Then $\{y_t^2\}_{t\in\mathbb{Z}}$ admits an ARMA(1,1) representation*

$$y_t^2 = \omega + (\alpha_1 + \beta_1)y_{t-1}^2 + \eta_t - \beta_1\eta_{t-1}$$

where $\{\eta_t\}_{t\in\mathbb{Z}}$ is a white noise process.

Proof. Define the white noise sequence $\eta_t = y_t^2 - \sigma_t^2$. Then, plugging in $\sigma_t^2 = y_t^2 - \eta_t$ and $\sigma_{t-1}^2 = y_{t-1}^2 - \eta_{t-1}$ in the updating equation for the GARCH(1,1), we obtain

$$y_t^2 = \omega + (\alpha_1 + \beta_1)y_{t-1}^2 + \eta_t - \beta_1\eta_{t-1},$$

which is indeed an ARMA(1,1) process. ■

The ARMA(1,1) representation of the squared returns generated by a GARCH(1,1) model can be useful to obtain the unconditional variance of y_t. Theorem 3.1 tells us that the returns y_t generated by a GARCH(1,1) model with $\alpha_1 + \beta_1 < 1$ have a time-invariant unconditional variance given by $\omega/(1 - \beta_1 - \alpha_1)$.

Theorem 3.1. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(1,1) model with $\alpha_1 + \beta_1 < 1$ have a time-invariant unconditional variance given by $\mathbb{V}ar(y_t) = \omega/(1 - \beta_1 - \alpha_1)$.*

Proof. We know that

$$\mathbb{V}\text{ar}(y_t) = \mathbb{E}(y_t^2),$$

therefore from the ARMA(1,1) representation of y_t^2 in Lemma 3.4 we immediately obtain

$$\mathbb{E}(y_t^2) = \omega/(1 - \beta_1 - \alpha_1).$$

■

For a GARCH model with $\omega = 0.1$, $\alpha_1 = 0.2$ and $\beta_1 = 0.5$, Figure 3.2 shows that the variance of y_t increases dramatically as β_1 increases from 0.5 to 0.75, and $1 - \beta_1 - \alpha_1$ approaches zero.

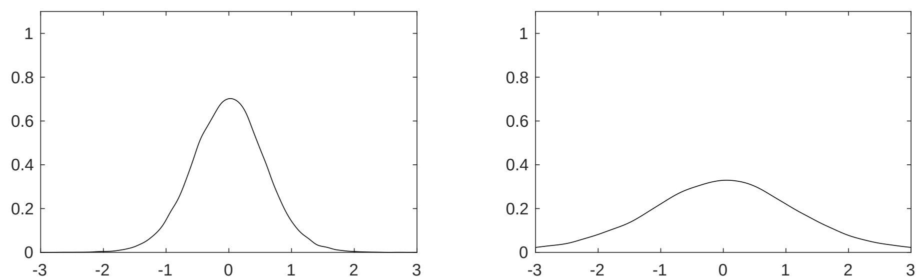

Figure 3.2: Unconditional sample density of y_t generated by a GARCH(1,1) model with parameters $(\omega, \alpha_1, \beta_1) = (0.1, 0.2, 0.5)$ [left figure] and $(\omega, \alpha_1, \beta_1) = (0.1, 0.2, 0.75)$ [right figure].

To better appreciate why the GARCH(1,1) model is able to capture high persistence in the variance, we can rewrite the GARCH(1,1) model as an ARCH(∞) model with some constraints on the parameters. In particular, unfolding the GARCH(1,1) updating equation, the conditional variance σ_t^2 can be expressed as

$$\sigma_t^2 = \frac{\omega}{(1 - \beta_1)} + \alpha_1 \sum_{i=0}^{\infty} \beta_1^i y_{t-1-i}^2.$$

Theorem 3.2. *Let $\{y_t\}_{t \in \mathbb{Z}}$ be a time-series generated by the GARCH(1,1) model and let the following additional parameter restriction hold $\alpha_1 + \beta_1 < 1$. Then, $\{y_t\}_{t \in \mathbb{Z}}$ is a weakly stationary White Noise sequence with $\mathbb{E}(y_t) = 0$, $\mathbb{V}ar(y_t) = \omega/(1 - \beta_1 - \alpha_1)$ and $\mathbb{C}ov(y_t, y_{t-l}) = 0$ for any $l \neq 0$.*

Figure 3.3 shows plots the ACF of y_t^2 and shows that the temporal dependence in squared returns is greatly affected by the value of β_1.

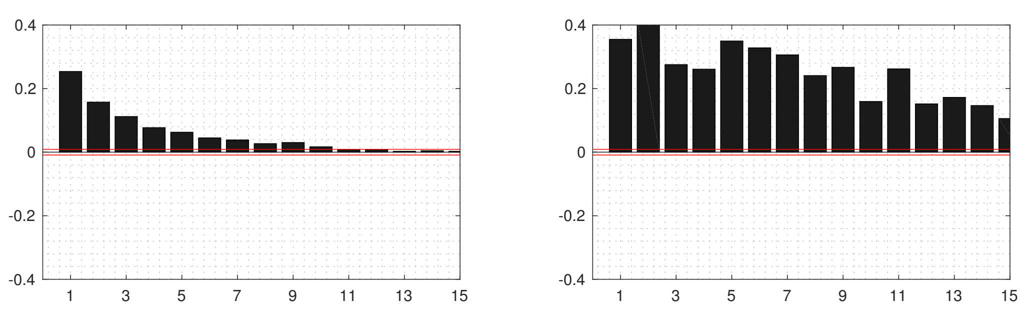

Figure 3.3: Sample ACF of squared returns y_t^2 generated by a GARCH(1,1) model with parameters $(\omega, \alpha_1, \beta_1) = (0.1, 0.2, 0.5)$ [left figure] and $(\omega, \alpha_1, \beta_1) = (0.1, 0.2, 0.78)$ [right figure].

3.2 The GARCH(p,q) model

In order to describe additional temporal dynamics, the GARCH(1,1) model can be further extended to a GARCH(p,q) with general orders p and q.

Definition 3.2. GARCH(p,q) Model: *The GARCH(p,q) model is given by*

$$y_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i y_{t-i}^2, \qquad \forall\ t \in \mathbb{Z}, \tag{3.1}$$

where $\omega > 0$, $\alpha_1 \geq 0$, $\beta_1 \geq 0$ are parameters, $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ is an $NID(0,1)$ sequence and ε_t is independent of the past Y^{t-1}.

As for all previous models, it can be shown that $y_t|Y^{t-1}$ has a normal distribution with mean zero and variance σ_t^2.

Lemma 3.5. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by a GARCH(p,q) model. The conditional distribution of y_t given the past Y^{t-1} is normal with mean $\mathbb{E}(y_t|Y^{t-1}) = 0$ and variance $\mathbb{V}ar(y_t|Y^{t-1}) = \sigma_t^2$, namely $y_t|Y^{t-1} \sim N(0, \sigma_t^2)$.*

The proof that the unconditional mean and autocovariances of the returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by the GARCH(p,q) process are zero, is once again identical to that of the simple ARCH(1) model.

Lemma 3.6. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(p,q) model have zero autocovariance at any lag, and are thus uncorrelated over time.*

Lemma 3.7. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(p,q) model have unconditional mean zero.*

An ARMA(q^*,p) representation is also available for the squared returns of the GARCH(p,q) model, where $q^* = \max\{q, p\}$. Notice that the number p of MA lags in the ARMA representation of the GARCH(p, q) is determined exclusively by the number of y_t^2 in the GARCH updating equation. In contrast, the number of AR lags in the ARMA representation depends on the number of lags of both y_t^2 and σ_t^2 featured in the GARCH updating equation.

Lemma 3.8. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by a GARCH(p,q) model. Then $\{y_t^2\}_{t\in\mathbb{Z}}$ admits an ARMA($\max\{q,p\}$,p) representation*

$$y_t^2 = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{i=1}^{p} \beta_i y_{t-i}^2 + \eta_t - \sum_{i=1}^{p} \beta_i \eta_{t-i}$$

where $\{\eta_t\}_{t\in\mathbb{Z}}$ is a white noise process.

Proof. Define the white noise sequence $\eta_t = y_t^2 - \sigma_t^2$. Then, plugging in $\sigma_t^2 = y_t^2 - \eta_t$ and $\sigma_{t-i}^2 = y_{t-i}^2 - \eta_{t-i}$ in the updating equation for the GARCH(p,q), we obtain

$$y_t^2 = \omega + \sum_{i=1}^{q} \alpha_i y_{t-i}^2 + \sum_{i=1}^{p} \beta_i y_{t-i}^2 + \eta_t - \sum_{i=1}^{p} \beta_i \eta_{t-i},$$

which is indeed an ARMA($\max\{q,p\}$,p) process. ■

Once again, the ARMA representation of the squared returns generated is useful to obtain the unconditional variance of y_t

Theorem 3.3. *The returns $\{y_t\}_{t\in\mathbb{Z}}$ generated by an GARCH(p,q), model with $\sum_{i=1}^{q} \alpha_i + \sum_{i=1}^{p} \beta_j < 1$ have a time-invariant unconditional variance given by $\mathbb{V}ar(y_t) = \omega/(1 - \sum_{i=1}^{q} \alpha_i - \sum_{i=1}^{p} \beta_j)$.*

We also obtain that a sequence $\{y_t\}_{t\in\mathbb{Z}}$ generated by a $GARCH$(p,q) is white noise

Corollary 3.1. *Let $\{y_t\}_{t\in\mathbb{Z}}$ be generated by a $GARCH(p,q)$ model satisfying $\sum_{i=1}^{q} \alpha_i + \sum_{i=1}^{p} \beta_j < 1$. Then $\{y_t\}_{t\in\mathbb{Z}}$ is weakly stationary white noise sequence.*

3.3 Simulating GARCH models with R

As we shall now see, it is very easy to simulate data from a GARCH(1,1) model using R. The code discussed below can be found in the R file `Simulate_GARCH.R`. First, we define the sample size n, and the parameter values for ω, α_1 and β_1.

```
n <- 1000
omega <- 0.1
alpha <- 0.2
beta <-  0.75
```

Next, we generate n Gaussian $\mathrm{N}(0,1)$ random innovations, labeled `epsilon`, by calling the function *rnorm*.

```
epsilon <- rnorm(n)
```

Then, we define 2 vectors of length n full of zeros to contain our simulated conditional variance $\{\sigma_t^2\}_{t=1}^n$ and simulated observations $\{y_t\}_{t=1}^n$.

```
sig2 <- rep(0,n)
x <- rep(0,n)
```

Finally, we simulate data from our GARCH(1,1) model using a *for loop* that runs from $t = 2$ to $t = n$. Notice that before starting the *for loop* we must first set the initial value σ_1^2. A reasonable option is to set it to the unconditional variance of y_t, which is given by $\omega/(1 - \alpha_1 - \beta_1)$.

```
sig2[1] <- omega/(1-alpha-beta)

x[1] <- sqrt(sig2[1]) * epsilon[1]

for(t in 2:n){

  sig2[t] <- omega + alpha * x[t-1]^2 + beta * sig2[t-1]

  x[t] <- sqrt(sig2[t]) * epsilon[t]

}
```

Stacking the entire code together we obtain the following R script for simulating a time series of length $n = 1000$ from a GARCH(1,1) model.

```
n <- 1000
omega <- 0.1
alpha <- 0.2
beta <-  0.75

epsilon <- rnorm(n)

sig2 <- rep(0,n)
x <- rep(0,n)

sig2[1] <- omega/(1-alpha-beta)
x[1] <- sqrt(sig2[1]) * epsilon[1]
```

```
for(t in 2:n){
  sig2[t] <- omega + alpha * x[t-1]^2 + beta * sig2[t-1]
  x[t] <- sqrt(sig2[t]) * epsilon[t]
}
```

Chapter 4

Parameter Estimation

In practice, given an observed sequence of T stock returns $\{y_1, y_2, ..., y_T\}$, generated by a GARCH(p, q) model with $\theta_0 = (\omega, \alpha_1, ..., \alpha_q, \beta_1, ..., \beta_p)$, we do not know what are the parameter values that correctly describe the dynamics of the time-varying conditional volatility. The problem we face is that of trying to find the values of the *true parameter vector* θ_0 of the GARCH(p, q) model from which the data $\{y_1, y_2, ..., y_T\}$ was generated. We thus need to find a way of *estimating* and *conducting inference* on the parameter vector θ_0.

A standard method used for the estimation of GARCH parameters is the method of *Maximum Likelihood*. In this section, we shall see how to write down the *Likelihood Function* for any given GARCH(p, q) model, and how to obtain the *Maximum Likelihood Estimator* (MLE) $\hat{\theta}_T$ of the unknown parameter vector θ_0.

4.1 Deriving the likelihood function

As you already know from your introductory probability courses, the joint density function $f(x, y)$ of two random variables x and y, can *always* be factorized into the product of a a conditional density $f(x|y)$ and a marginal density $f(y)$,

$$f(x, y) = f(x|y) \times f(y).$$

In your introductory time series courses you have also noted that this factorization can be very useful in the context of maximum likelihood! In particular, the joint density function of the data $\{y_1, ..., y_T\}$ can be factorized as the product of several conditional densities $f(y_t|y_{t-1})$ and a marginal density $f(y_1)$.

For concreteness, let $p(y_1, \ldots, y_t; \theta)$ denote the joint probability density function of the vector of random returns $\{y_1, \ldots, y_T\}$ generated by a GARCH model. Note that the joint density of the data depends on the vector of parameters $\theta = (\omega, \alpha_1, ..., \alpha_q, \beta_1, ..., \beta_p)$ since, as we verified in the previous chapter, these parameters determine the distributional properties of the data. Recall from your introductory statistics courses that, the *likelihood function* is exactly the same as the *joint density function* $p(y_1, \ldots, y_t; \theta)$. The only difference is that that the likelihood function takes the data $\{y_1, ..., y_T\}$ as given and analyses $p(y_1, \ldots, y_t; \theta)$ as being a function of the parameter vector θ, whereas the *joint density function* takes the parameter vector θ as given (it is fixed) and analyses $p(y_1, \ldots, y_t; \theta)$ as a function of the data $\{y_1, ..., y_T\}$.

In any case, you can naturally factorize the joint density function as follows

$$p(y_1, \ldots, y_t; \theta) = p(y_2, \ldots, y_t|y_1; \theta) \times p(y_1; \theta).$$

Furthermore, we can also factorize the joint density $p(y_2, \ldots, y_t|y_1; \theta)$ as

$$p(y_2, \ldots, y_t|y_1; \theta) = p(y_3, \ldots, y_t|y_1; \theta) \times p(y_2|y_1; \theta)$$

which implies that

$$p(y_1, \ldots, y_t; \theta) = p(y_3, \ldots, y_t|y_2, y_1; \theta) \times p(y_2|y_1; \theta) \times p(y_1; \theta).$$

Repeating this procedure, we obtain the desired factorization of the joint density function

$$
\begin{aligned}
p(y_1, \ldots, y_T; \theta) &= p(y_1; \theta) \times p(y_2|y_1; \theta) p(y_3|y_2, y_1; \theta) \times \cdots \times p(y_T|y_{T-1}, \ldots, y_1; \theta) \\
&= p(y_1; \theta) \prod_{t=2}^{T} p(y_t|y_{t-1}, \ldots, y_1; \theta), \qquad (4.1)
\end{aligned}
$$

where $p(y_1; \theta)$ denotes the marginal density of y_1 and $p(y_t|y_{t-1}, \ldots, y_1; \theta)$ denotes the conditional density of y_t given all the previous elements $\{y_{t-1}, \ldots, y_1\}$.

Naturally, you may ask: *why is this factorization useful?* Well, the answer is that while the joint density of the data $p(y_1, \ldots, y_T; \theta)$ is very complicated and potentially even intractable, each of the conditional densities $p(y_t|y_{t-1}, \ldots, y_1; \theta)$ is perfectly simple. Indeed, as we have seen in Chapters 2 and 3, for any GARCH model, the distribution of y_t conditional on its past $\{y_{t-1}, y_{t-2}, ...\}$, is Gaussian with mean zero and variance σ_t^2,

$$
y_t|y_{t-1}, y_{t-2}, ... \sim N(0, \sigma_t^2).
$$

The reason for this is that, conditional on the past data $\{y_{t-1}, y_{t-2}, ...\}$, the conditional variance σ_t^2 is given (i.e. it is known!). The probability density function of a normal random variable y_t with mean zero and variance σ_t^2 is simply given by

$$
p(y_t|y_{t-1}, y_{t-2}, ...; \theta) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{y_t^2}{2\sigma_t^2}\right\}.
$$

As you may also recall, we often work with the logarithm of the likelihood function, the so called *log-likelihood* function. In the following, we discuss how to obtain the log-likelihood function for ARCH and GARCH models.

Log-likelihood function of the ARCH(1) model

In the case $\{y_t\}_{t=1}^T$ is an observed sample of size T from an ARCH(1) model, we have that $p(y_t|y_{t-1}, \ldots, y_1; \theta) = p(y_t|y_{t-1}; \theta)$ because y_t depends only on the previous observation y_{t-1}. Furthermore, as discussed in the previous section $p(y_t|y_{t-1}; \theta)$, is a Normal density with mean zero and variance $\sigma_t^2 = \omega + \alpha_1 y_{t-1}^2$. More specifically,

$$
p(y_t|y_{t-1}; \theta) = \frac{1}{\sqrt{2\pi(\omega + \alpha_1 y_{t-1}^2)}} \exp\left\{-\frac{y_t^2}{2(\omega + \alpha_1 y_{t-1}^2)}\right\}.
$$

As a result, for the ARCH(1) model, we can write the log-likelihood function as

$$
\begin{aligned}
\log\left(p(y_1, \ldots, y_T; \theta)\right) &= \log\left(p(y_1; \theta)\right) + \sum_{t=2}^{T} \log\left(p(y_t|y_{t-1}; \theta)\right) \\
&= \log\left(p(y_1; \theta)\right) - \frac{1}{2}\sum_{t=2}^{T}\left(\log(2\pi) + \log\left(\omega + \alpha_1 y_{t-1}^2\right) + \frac{y_t^2}{\omega + \alpha_1 y_{t-1}^2}\right). \qquad (4.2)
\end{aligned}
$$

In practice, the marginal density $p(y_1; \theta)$ has an unknown functional form and therefore it is common practice to use a conditional log-likelihood function where y_1 is treated as given. Therefore the term $\log\left(p(y_1; \theta)\right)$ in (4.2) can be simply dropped and we consider the following log-likelihood function

$$
L(y_1, ..., y_T, \theta) = \sum_{t=2}^{T} l_t(\theta), \qquad (4.3)
$$

where each term $l_t(\theta)$, which denotes the contribution to the likelihood of observation t, is given by

$$
l_t(\theta) = -\frac{1}{2}\left(\log\left(\omega + \alpha_1 y_{t-1}^2\right) + \frac{y_t^2}{\omega + \alpha_1 y_{t-1}^2}\right).
$$

Note that in the log-likelihood in (4.3) we also dropped the term $\log(2\pi)$ as it is an additive constant and therefore irrelevant. Additive constants can be always dropped when writing log-likelihood functions because the vector θ that maximizes $L(y_1, ..., y_T, \theta)$ is the same that maximizes $L(y_1, ..., y_T, \theta) + c$, where c is any given constant not depending on θ. Furthermore, derivatives of $L(y_1, ..., y_T, \theta)$ are the same as derivatives of $L(y_1, ..., y_T, \theta) + c$. This can be also noted from Figure 4.1. The first plot shows the log-likelihood function in (4.3) for different values of α_1 using Apple stock returns. The second plot instead shows the log-likelihood with the additional term $\log(2\pi)$ as in (4.2). As you can see the likelihood function is exactly the same, it is only shifted along the vertical axis. Therefore, maximizing this two functions leads to the same result. Finally, the third plot of Figure 4.1 shows the likelihood function instead of the log-likelihood. This shows that both function, though different, are always maximized at the same point. This is the reason why we can use the log-likelihood instead of the likelihood.

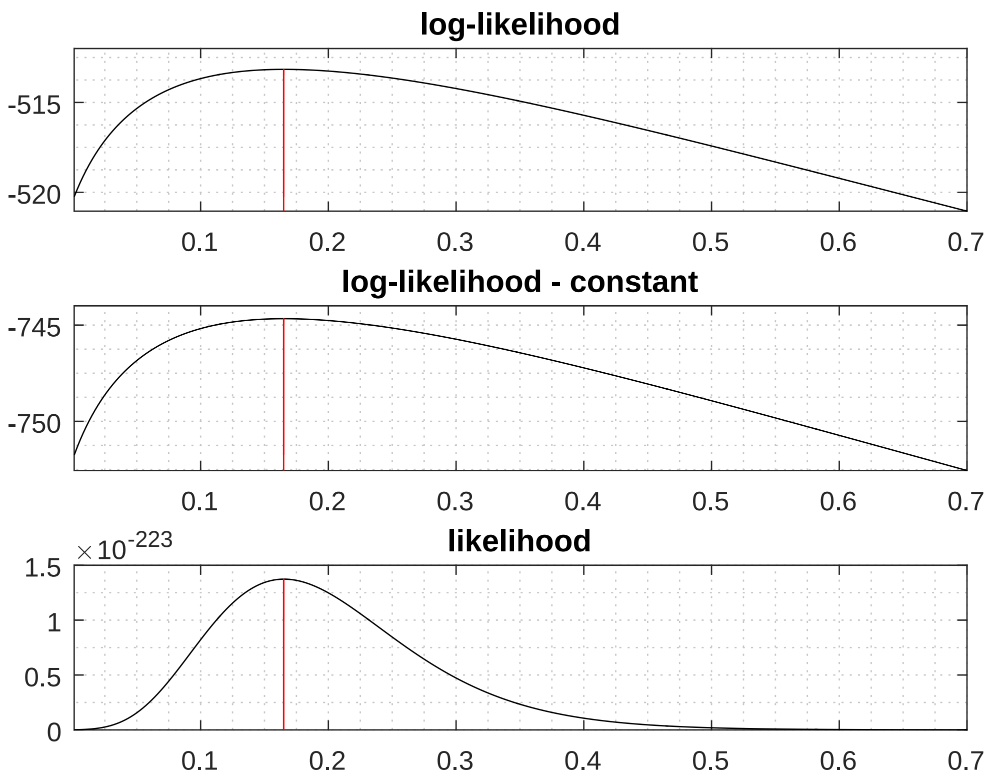

Figure 4.1: Likelihood functions for Apple daily log-returns in 2015 for different values of α_1 with fixed $\omega = 2$. The red lines denote the values α_1 that maximize the functions.

Likelihood function of the ARCH(q) model

Using a similar argument as discussed for the ARCH(1) model, it can be shown that the conditional log-likelihood function of an ARCH(q) model is given by

$$L(y_1, ..., y_T, \theta) = \sum_{t=q}^{T} l_t(\theta),$$

where $l_t(\theta)$ is given by

$$l_t(\theta) = -\frac{1}{2}\left(\log\left(\omega + \alpha_1 y_{t-1}^2 + \cdots + \alpha_q y_{t-q}^2\right) + \frac{y_t^2}{\omega + \alpha_1 y_{t-1}^2 + \cdots + \alpha_q y_{t-q}^2}\right).$$

Note that in this case the first q observations, $y_1, \ldots, y_q$, are treated as given constants and the sum starts from $q + 1$. This because we need q past observations, $\{y_{t-q}, \ldots, y_{t-1}\}$, to obtain the conditional variance σ_t^2 and the first observation is available at time $t = 1$.

Likelihood function of the GARCH(1,1) model

When the sample of data is generated by a GARCH(1,1) model, then we can simply use the general formula for the conditional density

$$p(y_t|y_{t-1};\theta) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{y_t^2}{2\sigma_t^2}\right\}.$$

where σ_t^2 is naturally obtained as

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

As a result, for the GARCH(1,1) model, we can write the log-likelihood function as

$$\begin{aligned} \log\left(p(y_1, \ldots, y_T;\theta)\right) &= \log\left(p(y_1;\theta)\right) + \sum_{t=2}^{T} \log\left(p(y_t|y_{t-1};\theta)\right) \\ &= \log\left(p(y_1;\theta)\right) - \frac{1}{2}\sum_{t=2}^{T}\left(\log(2\pi) + \log\sigma_t^2 + \frac{y_t^2}{\sigma_t^2}\right). \qquad (4.4) \end{aligned}$$

As argued before, the constant terms and marginal density of y_1 are typically ignored, giving rise to a simplified log-likelihood function of the form

$$L(y_1, ..., y_T, \theta) = \sum_{t=2}^{T} l_t(\theta) = \sum_{t=2}^{T} -\frac{1}{2}\left(\log\sigma_t^2 + \frac{y_t^2}{\sigma_t^2}\right). \qquad (4.5)$$

Note that since we are using the recursion

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

at time $t = 2$ we have that $\sigma_2^2 = \omega + \alpha_1 y_1^2 + \beta_1 \sigma_1^2$. Indeed y_1 is observed but what about σ_1^2? In practice σ_1^2 is fixed to be equal to a given constant (often the sample variance of the observations is used).

Likelihood function of the GARCH(p, q) model

For the GARCH(p, q) model the expression of the log-likelihood is the same as the one for the GARCH(1,1) model given in (4.5). The only difference lies in the way the σ_t^2 is obtained through the updating equation GARCH(p, q) model.

4.2 Maximum Likelihood Estimator and Asymptotic properties

Once we have obtained the conditional log-likelihood function $L(y_1, ..., y_T, \theta)$, the *maximum likelihood estimator* (MLE) $\hat{\theta}_T$ obtained from the sample of data $\{y_1, ..., y_T\}$, is simply defined as the argument that maximizes the log-likelihood function $L(y_1, ..., y_T, \theta)$ over the parameter space Θ containing all possible values of the parameters. In particular,

$$\hat{\theta}_T = \arg\max_{\theta \in \Theta} L(y_1, ..., y_T, \theta).$$

It is important to note that the log-likelihood function $L(y_1, ..., y_T, \theta)$ is a random variable. Indeed, for every new realization of the random sample $\{y_1, ..., y_T\}$, we obtain a new log-likelihood function to be maximized over θ. As such, the maximum likelihood estimator $\hat{\theta}_T$ is also a random variable. In particular, for every new sample of data $\{y_1, ..., y_T\}$ there is a new value $\hat{\theta}_T$ that maximizes the log-likelihood. We call this value the *point estimate* of the *true parameter* θ_0.

While the MLE $\hat{\theta}_T$ is a continuous random variable, and hence, it will be almost surely different from θ_0, it does have important properties. In particular, under standard regularity conditions, it can be shown that $\hat{\theta}_T$ is *consistent and asymptotically normal* for θ_0.

Recall from introduction to econometrics that an estimator $\hat{\theta}_T$ is said to be a *consistent* estimator of the true unknown θ_0 if $\hat{\theta}_T$ converges in probability to θ_0 as the sample size T diverges to infinity. Furthermore, $\hat{\theta}_T$ is said to be asymptotically normal for θ_0 if the random variable $\sqrt{T}(\hat{\theta}_T - \theta_0)$ converges in distribution to a multivariate normal random variable $N(\mathbf{0}, \Omega)$, where $\mathbf{0}$ is a vector of zeros and Ω is a variance covariance matrix called the *asymptotic variance* of $\hat{\theta}_T$.

Lemma 4.1. *Under appropriate regularity conditions $\hat{\theta}_T$ converges in probability to θ_0 as the sample size diverges*

$$\hat{\theta}_T \xrightarrow{p} \theta_0 \quad as \quad T \to \infty.$$

Lemma 4.2. *Under appropriate regularity conditions $\hat{\theta}_T$ is asymptotically normal for θ_0,*

$$\sqrt{T}(\hat{\theta}_T - \theta_0) \xrightarrow{d} N(\mathbf{0}, \Omega) \quad as \quad T \to \infty$$

where $\Omega = \mathcal{I}(\theta_0)^{-1}$ is the inverse Fisher Information matrix

$$\mathcal{I}(\theta_0) = -\mathbb{E}\left(\frac{\partial^2 l_t(\theta)}{\partial\theta\partial\theta^\top}\right) = \frac{1}{2}\mathbb{E}\left(\frac{1}{\sigma_t^4}\frac{\partial\sigma_t^2}{\partial\theta}\frac{\partial\sigma_t^2}{\partial\theta^\top}\right).$$

At this point, it is important to note that:

(a) the inverse Fisher Information matrix $\mathcal{I}(\theta_0)$ must be inverted for us to obtain the asymptotic variance Ω of the MLE;

(b) the Fisher Information matrix depends on the derivative process $\frac{\partial\sigma_t^2}{\partial\theta}$.

In practice, point (a) means that we must be careful in scaling the log-likelihood function since any computer program may have trouble in finding the inverse of a matrix that is not properly scaled.

Point (b) is important since it shows that the random sequence $\frac{\partial\sigma_t^2}{\partial\theta}$ plays a role in obtaining the asymptotic variance of the maximum likelihood estimator. Note that $\{\frac{\partial\sigma_t^2}{\partial\theta}\}$ is a vector time-series determined by its own updating equation. Take the GARCH(1,1) model as an example, then we have

$$\frac{\partial\sigma_t^2}{\partial\theta} = \left[\begin{array}{ccc} \frac{\partial\sigma_t^2}{\partial\omega} & \frac{\partial\sigma_t^2}{\partial\alpha} & \frac{\partial\sigma_t^2}{\partial\beta} \end{array}\right]$$

where the elements of this vector are obtained through the following updating equations

$$\begin{aligned}\frac{\partial\sigma_t^2}{\partial\omega} &= \frac{\partial\omega}{\partial\omega} + \frac{\partial\alpha y_{t-1}^2}{\partial\omega} + \frac{\partial\beta\sigma_{t-1}^2}{\partial\omega} \\ &= 1 \;+\; 0 \;+\; \beta\frac{\partial\sigma_{t-1}^2}{\partial\omega} \quad ,\end{aligned}$$

$$\begin{aligned}\frac{\partial\sigma_t^2}{\partial\alpha} &= \frac{\partial\omega}{\partial\alpha} + \frac{\partial\alpha y_{t-1}^2}{\partial\alpha} + \frac{\partial\beta\sigma_{t-1}^2}{\partial\alpha} \\ &= 0 \;+\; y_{t-1}^2 \;+\; \beta\frac{\partial\sigma_{t-1}^2}{\partial\alpha} \quad ,\end{aligned}$$

$$
\begin{aligned}
\frac{\partial \sigma_t^2}{\partial \beta} &= \frac{\partial \omega}{\partial \beta} + \frac{\partial \alpha y_{t-1}^2}{\partial \beta} + \frac{\partial \beta \sigma_{t-1}^2}{\partial \beta} \\
&= \; 0 \; + \; 0 \; + \; \sigma_{t-1}^2 \; + \; \beta \frac{\partial \sigma_{t-1}^2}{\partial \alpha} \; .
\end{aligned}
$$

Taking all equations together we obtain the following lemma.

Lemma 4.3. *The conditional volatility derivative process $\{\partial \sigma_t^2 / \partial \theta\}$ of the GARCH(1,1) model satisfies the following updating equation*

$$
\frac{\partial \sigma_t^2}{\partial \theta} = \begin{bmatrix} 1 \\ y_{t-1}^2 \\ \sigma_{t-1}^2 \end{bmatrix} + \beta \frac{\partial \sigma_{t-1}^2}{\partial \theta}.
$$

4.3 Statistical Inference

The asymptotic normality result stated in Lemma 4.2 above shows that, for large T, the maximum likelihood estimator $\hat{\theta}_T$ is a random variable that is approximately Gaussian, centered at the unknown θ_0, and with a variance that vanishes to zero as $T \to \infty$. Indeed, the asymptotic normality of the MLE tells us that

$$
\sqrt{T}\Big(\hat{\theta}_T - \theta_0\Big) \overset{app}{\sim} N\Big(0 \; , \; \mathcal{I}(\theta_0)^{-1} \Big)
$$

where $\overset{app}{\sim}$ denotes an 'approximate' distribution. This means naturally, that

$$
\hat{\theta}_T - \theta_0 \overset{app}{\sim} N\Big(0 \; , \; \frac{1}{T} \, \mathcal{I}(\theta_0)^{-1} \Big)
$$

and hence that

$$
\hat{\theta}_T \overset{app}{\sim} N\Big(\theta_0 \; , \; \frac{1}{T} \, \mathcal{I}(\theta_0)^{-1} \Big).
$$

Notice how the variance $\frac{1}{T} \, \mathcal{I}(\theta_0)^{-1}$ os the MLE vanishes to zero as $T \to \infty$. This means that the distribution of the maximum likelihood estimator collapses to the true parameter θ_0 and becomes becomes increasingly accurate as $T \to \infty$.

Figure 4.2 below shows the density of the MLE for the parameters ω, α_1 and β_1 in the context of a GARCH(1,1) model. These densities were obtained through a Monte Carlo exercise. In particular, we first simulate data $\{y_1, ..., y_T\}$ from a GARCH(1,1) model with parameter vector $\theta_0 = (\omega, \alpha_1, \beta_1) = (0.1, 0.2, 0.75)$, and then use the simulated data to obtain a point estimate $\hat{\theta}_T$ pretending that we do not know θ_0. By repeating this procedure many times, we obtain a new point estimate $\hat{\theta}_T$ for each new time series that we simulate from the GARCH(1,1) model. Figure 4.2 shows the density of the MLE that we obtained by simulating $N = 1000$ time series with sample size $T = 500$, $T = 1000$ and $T = 5000$. This figure shows that the distribution of the MLE is indeed approximately normal in large samples, and furthermore, that it is collapsing to the true parameter θ_0.

Figure 4.3 uses the same Monte Carlo procedure to obtain the density of the quantity $\sqrt{T}(\hat{\theta}_T - \theta_0)$. The figure shows that $\sqrt{T}(\hat{\theta}_T - \theta_0)$ is indeed approximately normally distributed.

In practice, the Fisher information matrix $\mathcal{I}(\theta_0) = -\mathbb{E}(\partial^2 l_t(\theta)/\partial\theta\partial\theta^\top)$ is unknown since it depends on the true unknown parameter θ_0, and also, because the expectation $\mathbb{E}$ is unknown. We can however approximate $\mathcal{I}(\theta_0)$ by its plug-in estimator

$$
\mathcal{I}(\theta_0) \;=\; -\mathbb{E}\left(\frac{\partial^2 l_t(\theta)}{\partial\theta\partial\theta^\top}\right) \;\approx\; -\frac{1}{T}\sum_{t=1}^{T} \frac{\partial^2 l_t(\hat{\theta}_T)}{\partial\theta\partial\theta^\top}.
$$

Notice that, in the expression above, we have replaced the expectation $\mathbb{E}$ with the sample average $1/T \sum_{t=1}^{T}$ and we have substituted the unknown true parameter θ_0 by the sample estimate $\hat{\theta}_T$.

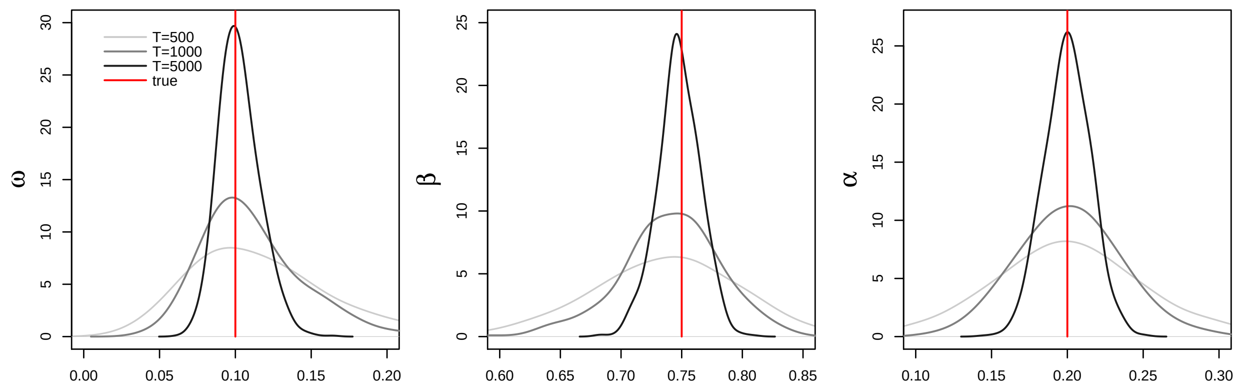

Figure 4.2: Distribution of the ML estimator $\hat{\theta}_T = (\hat{\omega}_T, \hat{\alpha}_T, \hat{\beta}_T)$ for different sample sizes T. The red line denotes the true value θ_0.

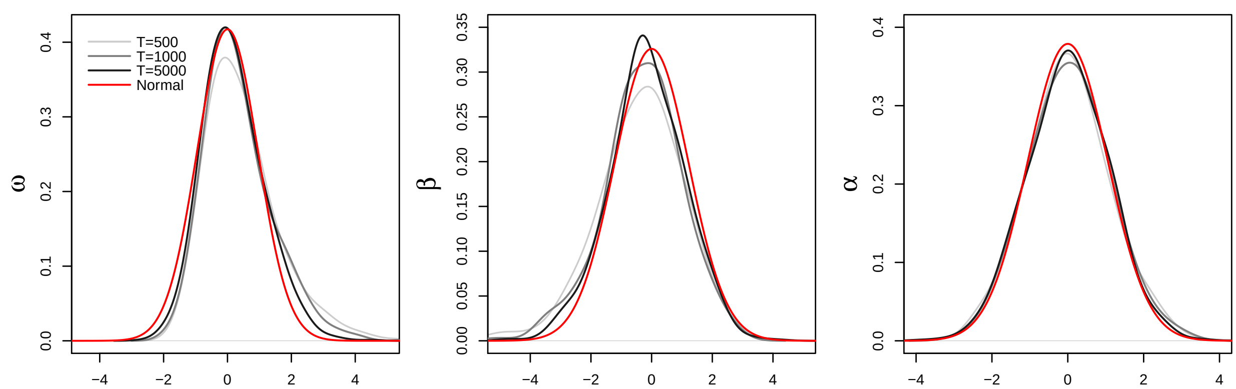

Figure 4.3: Distribution of $\sqrt{T}(\hat{\theta}_T - \theta_0)$ the for different sample sizes T. The red line denotes the normal density function.

We can now invert this estimate of the Fisher information matrix to obtain an estimate of the asymptotic variance-covariance matrix of the MLE $\hat{\theta}_T$

$$\hat{\Omega} = \left(-\frac{1}{T} \sum_{t=1}^{T} \frac{\partial^2 l_t(\hat{\theta}_T)}{\partial \theta \partial \theta^\top} \right)^{-1}.$$

With $\hat{\Omega}$ we are now in a position to report standard errors for our point estimates, construct confidence intervals for θ_0 and produce p-values. All of these are done in the same way as you have done in introductory statistics courses.

In particular, the standard error of the ith element of the vector θ, can be obtained by taking the square root of the ith diagonal element of $\hat{\Omega}_{ii}$,

$$\mathrm{SE}(\hat{\theta}_T^i) = \sqrt{\frac{1}{T}\hat{\Omega}_{ii}}$$

where $\hat{\theta}_T^i$ denotes the ith element of the vector $\hat{\theta}_T$ and $\hat{\Omega}_{ii}$ the element in the ith row ad ith column of the variance-covariance matrix $\hat{\Omega}$.

An approximate 95% confidence interval for the ith element of θ_0, labeled θ_0^i, can be obtained by taking an interval around the point estimate $\hat{\theta}_T^i$ that spans ± 1.96 standard errors

$$\left[\hat{\theta}_T^i - 1.96 \times \sqrt{\frac{1}{T}\hat{\Omega}_{ii}} \quad , \quad \hat{\theta}_T^i + 1.96 \times \sqrt{\frac{1}{T}\hat{\Omega}_{ii}}\right]$$

4.4 Numerical Optimization of the log-Likelihood Function

As you may recall, in introductory econometrics, we were always able to derive an expression for the maximum likelihood estimator, by taking a derivative of the log-likelihood, setting it to zero, and solving for $\hat{\theta}_T$. The idea was, of course, that if $\hat{\theta}_T$ is the maximizer of the log-likelihood function $L(y_1, ..., y_T, \theta)$, then it must satisfy

$$\sum_{t=1}^{T} \frac{\partial l_t(\hat{\theta}_T)}{\partial \theta} = \mathbf{0}.$$

In other words, the derivative of the likelihood must be zero at $\hat{\theta}_T$. This is how we obtained the expression for the MLE

$$\hat{\beta} = \frac{\sum_{t=1}^{T} y_t x_t}{\sum_{t=1}^{T} x_t^2}$$

in the context of the simple linear Gaussian regression model

$$y_t = \beta x_t + \epsilon_t.$$

Similarly, you may recall that this was the way in which we obtained the expression for the MLE

$$\hat{\rho} = \frac{\sum_{t=1}^{T} x_t x_{t-1}}{\sum_{t=1}^{T} x_{t-1}^2}$$

in the context of the Gaussian AR(1) model

$$x_t = \rho x_{t-1} + \epsilon_t.$$

Unfortunately, in the context of a GARCH model it is not possible for us to obtain estimates of the parameter vector in the same way. The problem is too complicated, and trying to set the derivative of the log-likelihood to zero does not deliver a closed form expression for $\hat{\theta}_T$.

Therefore, a naive way to proceed would be simply to evaluate the log-likelihood function for several values of θ and pick the one that maximizes it. For instance, Table 4.1 reports the log-likelihood values of a GARCH(1,1) model using Apple returns for different values of $\theta = (\omega, \alpha_1, \beta_1)$. In this case we could pick $\theta = (0.20, 0.07, 0.85)$ as it gives the maximum log-likelihood. This approach is naive and we can do much better by using numerical Algorithms.

Table 4.1: Log-likelihood function of the GARCH(1,1) model for daily Apple log-returns (from 2010 to 2016) evaluated at different values of θ.

parameter value $\theta = (\omega, \alpha_1, \beta_1)$	log-lik value
$(0.30, 0.10, 0.70)$	-2962.5
$(0.20, 0.10, 0.70)$	-3090.0
$(0.20, 0.07, 0.70)$	-3211.4
$(0.20, 0.07, 0.80)$	-2941.6
$(0.20, 0.07, 0.85)$	-2882.3

One simple algorithm to maximize the log-likelihood $L(y_1, ..., y_T, \theta)$ is the *Newton-Raphson* algorithm.

Remark 4.1. *Starting from an initial value $\theta^{(1)}$, the Newton-Raphson algorithm finds the maximum of the log-likelihood function $L_T(\theta) = L(y_1, ..., y_T, \theta)$ by updating the parameter as follows*

$$\theta^{(k+1)} = \theta^{(k)} - \nabla L_T(\theta^{(k)})\Big(\nabla^2 L_T(\theta^{k})\Big)^{-1}$$

where $\nabla L_T(\theta^{(k)})$ and $\nabla^2 L_T(\theta^{(k)})$ denote the Jacobian and Hessian matrix respectively

$$\nabla L_T(\theta^{(k)}) = \frac{\partial L_T(\theta^{(k)})}{\partial\theta} \quad \text{and} \quad \nabla^2 L_T(\theta^{(k)}) = \frac{\partial^2 L_T(\theta^{(k)})}{\partial\theta\partial\theta^\top}.$$

Lemma 4.4. *Under appropriate regularity conditions, the Newthon-Raphson algorithm converges to the MLE as the number of iterations k go to infinity; i.e. $\theta^{(k)} \to \hat{\theta}_T$ as $k \to \infty$.*

4.5 Estimating GARCH models with R

We will now make use of R to evaluate and maximize the log-likelihood function of a GARCH(1,1) model.

Write the likelihood function

We first create an R *function* to evaluate the log-likelihood function. We call this function `llik_fun_GARCH`. The function takes as input a vector of data labeled `x` and a parameter vector labeled `par`, and returns as output the log-likelihood. You can find this function as an R file labeled `llik_fun_GARCH.R`.

The inputs and outputs of the functions are defined by starting the script with the following line

```
llik_fun_GARCH <- function(par,x){
```

First, we define n to be the number of observations in the vector of data `x`. Then, we define the parameter values `omega`, `alpha` and `beta` from the input vector `par`. We consider some link functions to impose some restriction on `omega`, `alpha` and `beta`, which are useful to avoid numerical problems. In particular, we consider an exponential link function exp() to ensure $\omega > 0$. In this way, `par[1]` will be allowed to take any value in the optimization (even negative) but still ensuring $\omega > 0$. Similarly, we impose $0 < \alpha < 1$ and $0 < \beta < 1$ by using the logistic link function $\text{logistic}() = \exp()/(1 + \exp())$.

```
n <- length(x)
omega <- exp(par[1])
alpha <- exp(par[2])/(1+exp(par[2]))
beta <- exp(par[3])/(1+exp(par[3]))
```

We then use the data `x` and the parameters to *filter* the conditional variance, which we label `sig2` in the code. In particular, we use a *for loop* to obtain the sequence $\{\sigma_t^2\}_{t=1}^n$ recursively from $t = 2$ to $t = n$. We set σ_1^2 equal to the sample variance of the data `var(x)` to initialize the updating equation.

```
sig2 <- rep(0,n)
sig2[1] <- var(x)

for(t in 2:n){

    sig2[t] <- omega + alpha*x[t-1]^2 + beta*sig2[t-1]

}
```

Finally, we use the filtered sequence $\{\sigma_t^2\}_{t=1}^n$ to calculate the contribution to the log-likelihood function of each observation y_t. Below `l` is a vector that contains the log-likelihood values from $t = 1$ to n. Finally, we calculate the average log-likelihood value `llik` as the average of all contributions and give it as output using `return()`.

```
l <- -(1/2)*log(2*pi) - (1/2)*log(sig2) - (1/2)*x^2/sig2

llik <- mean(l)
return(llik)
```

Stacking all the code together, we obtain the following script to evaluate the log-likelihood function of the GARCH model.

```
llik_fun_GARCH <- function(par,x){
  n <- length(x)
  omega <- exp(par[1])
  alpha <- exp(par[2])/(1+exp(par[2]))
  beta <- exp(par[3])/(1+exp(par[3]))

  sig2 <- rep(0,n)
  sig2[1] <- var(x)

  for(t in 2:n){
    sig2[t] <- omega + alpha*x[t-1]^2 + beta*sig2[t-1]
  }

  l <- -(1/2)*log(2*pi) - (1/2)*log(sig2) - (1/2)*x^2/sig2

  llik <- mean(l)
  return(llik)
}
```

Optimizing the likelihood function

We are now in a position to optimize the log-likelihood function and obtain the maximum likelihood estimator $\hat{\theta}_T$. The code for estimating the parameters of the ARCH model through the MLE is available on the R file `Estimate_ML_GARCH.R` . First, we load the likelihood function using `source()`. Note that the file "`llik_fun_GARCH.R`" has to saved on the working directory. You can visualize the current working directory using the command `getwd()`. You can also change the working directory using `setwd("C:/Users/R code")`, where `C:/Users/R code` will be replaced with the path of the folder that you want to use as working directory.

```
source("llik_fun_GARCH.R")
```

Then, we load the series of observed data of interest to us. In this case, our data is saved in the file `stock_returns.txt`, and hence, we can load the data by writing

```
x <- scan("stock_returns.txt")
```

Next, we define the initial value for ω, α_1 and β_1. These are the values that the numerical algorithm will use to start the iteration towards the maximum of the log-likelihood. However, since the input `par` of the function `llik_fun_GARCH.R` is transformed using link functions (see above), we set the initialization `par_ini` by using the inverse of the link functions. The inverse of $\exp(p)$ is $\log(p)$ and the inverse of logistic(p) $= \exp(p)/(1 + \exp(p))$ is $\log(p/(1 - p))$.

```
a <- 0.2
b <- 0.6
omega <- var(x)*(1-a-b)

par_ini <- c(log(omega),log(a/(1-a)),log(b/(1-b)))
```

Finally, we obtain the point estimate $\hat{\theta}_T$ by optimizing the log-likelihood function using the R function `optim()`. Since the `optim()` function tries to find the *minimum* of any function, we must give it the *negative* of the log-likelihood function

```
optim(fn=function(par) - llik_fun_GARCH(par,x))
```

In order for `optim()` to work properly, we also need to provide the initial values of the parameter and select the numerical algorithm. We give it the starting value for the iterations `par_ini` and select the algorithm `BFGS`. We run the optimizer and store the results in `est`.

```
est <- optim(par=par_ini,fn=function(par)-llik_fun_GARCH(par,x), method = "BFGS")
```

Finally, we note that the output of the `optim()` function includes:

1. The point estimates $\hat{\theta}_T$, which will be contained in `est$par`;
2. The value of the negative average log-likelihood evaluated at $\hat{\theta}_T$, which will be contained in `est$value`;
3. An exit flag which will confirm if convergence was obtained or, in contrast, if the algorithm found an obstacle and the optimization was aborted. When the variable `est$convergence` is zero, then the optimization has been successful.

Standard errors and confidence intervals

The covariance matrix estimate $\hat{\Omega}$ can be obtained from the hessian matrix of the average negative log-likelihood (second derivative with respect to the parameters). The function `Hess_fun_GARCH()` gives the average log-likelihood but without transforming the parameter input with the link functions (see the R file `Hess_fun_GARCH.R`). This is needed because we want the hessian with respect to the original parameters not through the link functions. We obtain the hessian matrix using the R function `optimHess()` as follows:

```
hessian <- optimHess(par=theta_hat, fn=function(par)-Hess_fun_GARCH(par,x))
```

Next, we obtain the covariance matrix estimate $\hat{\Omega}$ by simply inverting the `hessian` matrix. We label the matrix $\hat{\Omega}$ as `SIGMA`. This is done through the following code

```
SIGMA <- solve(hessian)
```

Finally, we can use the covariance matrix `SIGMA` to obtain a 95% confidence interval for β. Note that the estimate of β is the third element of the vector `theta_hat` and its corresponding variance is the element in position (3,3) of the matrix `SIGMA`.

```
lb_beta <- theta_hat[3]-1.96*sqrt(SIGMA[3,3])/sqrt(length(x))
ub_beta <- theta_hat[3]+1.96*sqrt(SIGMA[3,3])/sqrt(length(x))

ci_beta <- c(lb_beta, ub_beta)
```

The resulting confidence interval is labeled `ci_beta`. This code can be found in the file `Estimate_ML_GARCH.R`.

Chapter 5

Financial Analysis of ARCH and GARCH Models

In this chapter, we turn to the econometric analysis of ARCH and GARCH models. In particular, we use estimated ARCH and GARCH models to produce measures of risk that are useful for the economic and financial analysis of data.

5.1 Estimation of the conditional volatility

Once the parameter vector θ_0 has been estimated we usually are also interested in obtaining an estimate of sequence of conditional variances $\{\sigma_t^2\}_{t=1}^T$. We denote this estimated sequence with $\{\hat{\sigma}_t^2\}_{t=1}^T$ and we call it the filtered volatility. How is $\{\hat{\sigma}_t^2\}_{t=1}^T$ obtained? Well, simply through model updating equation and plugging in the estimated parameter vector $\hat{\theta}_T$. For example, for the GARCH(1,1) model our parameter estimate will be $\hat{\theta}_T = (\hat{\omega}, \hat{\alpha}_1, \hat{\beta}_1)$. Therefore, $\{\hat{\sigma}_t^2\}_{t=1}^T$ is obtained as

$$\hat{\sigma}_t^2 = \hat{\omega} + \hat{\beta}_1 \hat{\sigma}_{t-1}^2 + \hat{\alpha}_1 y_{t-1}^2, \text{ for } t = 2, \dots, T.$$

The value $\hat{\sigma}_1^2$ can be set equal to the sample variance of the observations.

Deriving the estimated conditional volatility can be useful to understand how the risk of a certain financial asset (or market) evolved over time.

Conditional volatility with R

Using R, we can obtain the estimated conditional variance of a GARCH(1,1) as follows. The R code for this chapter is in the file `analysis_GARCH.R`. First note that the maximum likelihood estimate of θ_0, as obtained in the previous chapter, are given by `omega_hat`, `alpha_hat` and `beta_hat`. The time series is labeled `x`. We define a vector `sigma2` that will contain our estimated volatility. We also initialize the conditional volatility using the sample variance.

```
n <- length(x)
sigma2 <- rep(0,n)
sigma2[1] <- var(x)
```

Now we are ready to obtain recursively the estimated conditional volatility using the GARCH(1,1) updating equation and using `theta_hat` as parameter values.

```
for(t in 2:n){

  sigma2[t] <- omega_hat + alpha_hat*x[t-1]^2 + beta_hat*sigma2[t-1]

}
```

The estimated conditional volatility, together with the data series `stock_returns.txt`, is plotted in Figure 5.1.

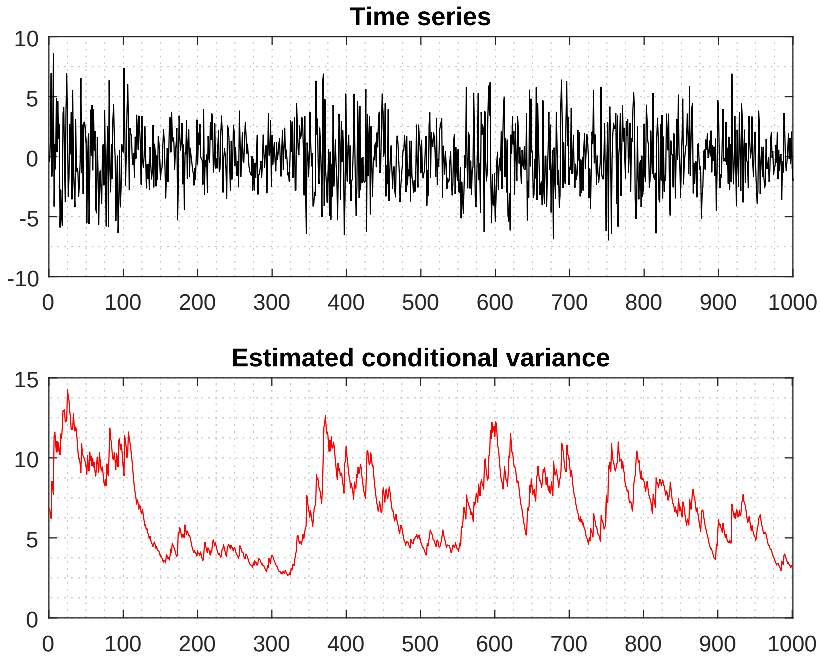

Figure 5.1: Time series (first plot) and estimated filtered variance (second plot).

5.2 Diagnostic tests

One main focus of concern for the economic and financial evaluation of any econometric model is naturally whether the model describes appropriately the dynamics of the data. In other words, it is important to test if the model is correctly specified. If the model seems to be well specified, we can proceed with confidence and extract valuable information from our model. In contrast, if there is strong evidence of model misspecification, then we should either search for a better model or, at the very minimum, we should be very careful about the conclusions that we derive from the model.

In the context of ARCH and GARCH models, we can use the residuals $u_t = y_t/\hat{\sigma}_t$ to test for correct model specification. The specification tests will fall into two main categories: (i) *Homoscedasticity tests*; and (ii) *Normality tests*.

Homoscedasticity tests focus on the fact that the error term ϵ_t is assumed to have fixed variance over time. Hence, the residuals $u_t = y_t/\hat{\sigma}_t \approx \epsilon_t$ should also exhibit this feature, at least approximately. A simple way of testing the homoscedasticity assumption, is to plot the autocorrelation function of the squared residuals $\{u_t^2\}_{t=1}^T$ and analyzing the Q-statistic associated to the ACF at each lag. In R we can obtain a vector, which we label `u`, that contains the standardized residuals using the following code

```
u=x/sqrt(sigma2)
```

where `x` is the data vector and `sigma2` contains the estimated conditional variance. An ACF for the squared residuals can be obtained writing the command

```
acf(u^2,main="")
```

Figure 5.2 shows the squared residuals and the corresponding ACF obtained from the dataset `stock_returns.mat`.

The normality tests focus on the fact that the error term ϵ_t is assumed to be Gaussian. Hence, the residuals $u_t = y_t/\hat{\sigma}_t \approx \epsilon_t$ should also be normal, at least approximately. The Jarque-Bera test and the qq-plot are two simple ways of testing for normality of the residual term $\{u_t\}_{t=1}^T$.

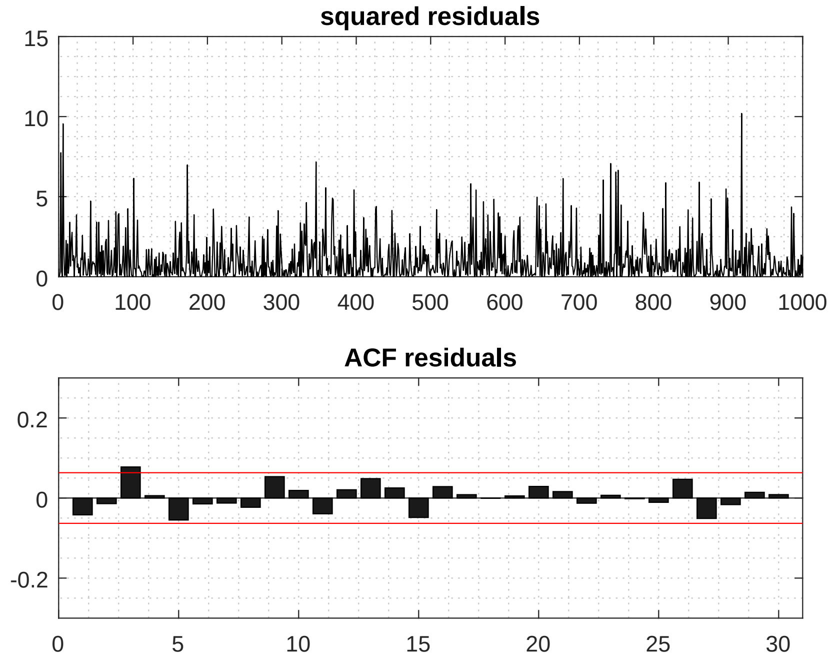

Figure 5.2: Squared residuals (first plot) and autocorrelation function of squared residuals (second plot).

The null hypothesis H_0 of the Jarque-Bera test is that the residuals are normal against the alternative H_1 that the residuals are not normal. The Jarque-Bera test statistic is given by

$$\text{JB} = \frac{T+1}{6}\big(\hat{\mu}_3^2 + (\hat{\mu}_4 - 3)^2\big),$$

where $\hat{\mu}_3$ denotes the sample skewness of the residuals $\{u_t\}_{t=1}^T$ and $\hat{\mu}_4$ denotes the sample Kurtosis. Under the null hypothesis H_0 of normal residuals the Jarque-Bera statistics JB has a chi-square distribution with 2 degrees of freedom χ_2^2. In R, a Jarque-Bera normality test for the residuals $\{u_t\}_{t=1}^T$ can be obtained using the command

```
jarque.bera.test(u)
```

The output gives several information including the p-value of the test.

5.3 Model Selection

Of course, in practice, it can happen that several competing models seem to be well specified, especially when they are nested. In such cases, it is important to find ways of selecting the best model among the set of alternative competing models.

Since we have learned how to estimate ARCH and GARCH models using the method of maximum likelihood, it may seem natural to select the model that achieves the best log-likelihood for the given data set at hand. However, as you may recall from your introductory econometrics courses, performing model selection by comparing log-likelihoods leads to incorrect results. The reason for this is simple: nested models with a larger number of parameters can fit the data better, hence they always achieve a higher log-likelihood in finite samples! It is important to highlight that a higher *sample* log-likelihood may be achieved *not* because the model is better, but simply because it is able to *overfit* the data. In other words, it can fit better the observed sample of data, but not the true dynamics.

One way of avoiding the problem of *overfitting* and incorrectly selecting larger models is to penalize the number of parameters in the model. This is the main idea behind the majority of the so-called *information criteria*. Two well known examples are the *Akaike's information criterion* (AIC), and the *Bayesian Information Criterion* (BIC). These information criteria take the value of the log-likelihood of the model $\log L(y_1, ..., y_T; \hat{\theta}_T)$, evaluated at the point estimate $\hat{\theta}_T$, and add a penalty for the number of parameters in the model $k = \text{dimension}(\theta)$,

$$\text{AIC} = 2k - 2\log L(y_1, ..., y_T; \hat{\theta}_T) \ ,$$

$$\text{BIC} = \log(T)k - 2\log L(y_1, ..., y_T; \hat{\theta}_T) \ .$$

Note that both the AIC and BIC are based on a negative log-likelihood, so that the lower the value of the information criterion, the better the model seems to be.

Comparing the AIC and BIC values of several models constitutes a reasonable basis for model selection.

5.4 Value-at-Risk

In finance, the *Value-at-Risk* (VaR) is a popular risk measure. Specifically, the daily α-VaR is the minimum amount the investor stands to loose with probability α over a period of one day. For example, if a portfolio has a daily 10%-VaR of 1 million euros, this means that there is a 10% probability that the value of the portfolio will fall by more than 1 million euros in one day.

The VaR can also stated in *percentage loss*. For example, if a portfolio has a daily 5%-VaR of 17%, then there is a 5% probability that the value of the portfolio will fall by more than 17% of its value in one day. Mathematically, given a portfolio value p_t at time t, and a random percentual return $y_t = (p_t - p_{t-1})/p_{t-1}$ on the portfolio, the 5%-VaR *in percentage loss* is defined as the value c that satisfies

$$P(y_t \leq c) = 0.05.$$

Note that the VaR expressed in percentage loss can immediately be turned into the VaR in monetary loss by multiplying c loss by the value of the stock at that time.

Definition 5.1. (Value-at-Risk - VaR) *Let $\{p_t\}_{t\in\mathbb{Z}}$ be a random sequence of portfolio values, and $y_t = (p_t - p_{t-1})/p_{t-1}$ denote the return sequence (i.e. the percentage changes) on the portfolio. The estimated α-VaR in percentage loss at time t is defined as the percentage value c that satisfies $P(y_t \leq c) = \alpha$. Furthermore, the estimated α-VaR in monetary loss at time $t \in \mathbb{Z}$ is given by $c \times p_t$.*

Conditional volatility models like the GARCH, can easily provide us with an estimate of the time-varying VaR of a stock at any time, conditional on past information. Indeed, since stock returns (typically expressed in percentage changes or log-differences) are modeled to satisfy

$$y_t = \sigma_t \varepsilon_t$$

the distribution of y_{t+1} conditional on the past and present Y^t is easily tractable for any t. For example, when the innovation sequence $\{\varepsilon_t\}$ is $NID(0,1)$, then $y_{t+1}|Y^t \sim N(0, \sigma_t^2)$. As a result, the VaR is obtained immediately through application of the Gaussian quantile function $\mathcal{Q}$ (the inverse of the Gaussian cumulative distribution function $\Phi(\cdot)$)

$$\mathcal{Q}(\alpha) = \inf\{y \in \mathbb{R} : \alpha \leq \Phi(y)\}.$$

In other words the α-VaR at time $t+1$ is the value q that satisfies

$$P(y_{t+1} \leq q|Y^t) = \alpha,$$

that in the Gaussian case is

$$\alpha\text{-VaR}_{t+1} = z_\alpha \sigma_{t+1}^2,$$

where z_α is the quantile of level α of a standard Normal distribution. Obviously, the same reasoning applies if ε_t has other distributions! Note that Monte Carlo simulations may be necessary for calculating a multiple step-ahead VaR.

Conditional VaR with R

The R file `analysis_GARCH.R` also illustrates the calculation of the α-VaR for $\alpha = 0.1$, 0.05, and 0.01, by applying a GARCH filter to a time series of (percentage) stock returns. The first part of the code, as we have seen in the previous section, provides the estimated conditional variance in the vector `sigma2`.

We obtain 3 vectors of length `n`, `VaR10`, `VaR05` and `VaR01`, that contain the conditional VaR at each t for $\alpha = 0.1$, 0.05, and 0.01. We use R quantile function for the normal `qnorm()` as follows:

```
VaR10 <- qnorm(0.1,0,sqrt(sigma2))
VaR05 <- qnorm(0.05,0,sqrt(sigma2))
VaR01 <- qnorm(0.01,0,sqrt(sigma2))
```

The output is shown in Figure 5.3 which plots the data (in black) and the respective conditional VaRs on the bottom graph.

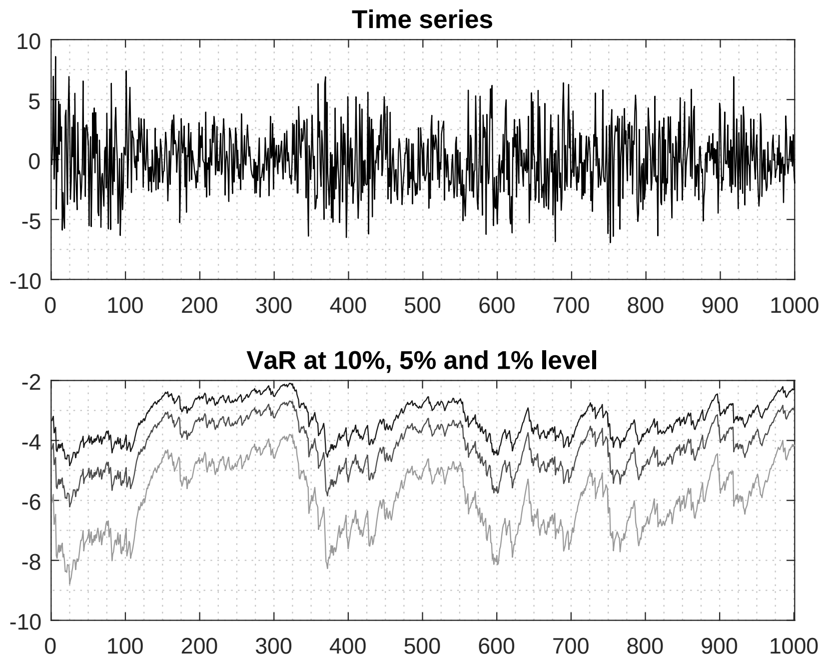

Figure 5.3: Conditional α-VaR for $\alpha = 0.1$, $\alpha = 0.05$, and $\alpha = 0.01$, estimated from a GARCH(1,1) model.

5.5 Forecasting conditional volatility

Often we are interested in forecasting the risk of financial assets in the next time period. Here we will see in practice how to obtain forecasts of the volatility of stock returns h-steps ahead.

In particular, assume we are at time T and we want to predict the variance of y_{T+h} for $h = \{1, 2, \dots\}$. Indeed the idea is to consider the variance of y_{T+h} conditional on the past and present Y^T as forecast. This variance forecast $\sigma^2_T(h)$ is given by

$$\sigma^2_T(h) = \mathbb{V}ar(y^2_{T+h}|Y^T) = \mathbb{E}(y^2_{T+h}|Y^T) = \mathbb{E}(\sigma^2_{T+h}|Y^T).$$

Note that it is immediate to see that when $h = 1$ we have $\sigma^2_T(1) = \sigma^2_{T+1}$ as the expectation of σ^2_{T+1} conditional on Y^T is simply σ^2_{T+1}. However, this is not the case for $h > 1$ and we have $\sigma^2_T(h) \neq \sigma^2_{T+h}$. We will now show how to obtain $\sigma^2_T(h)$ for ARCH and GARCH models.

For an ARCH(q) model $\sigma^2_T(h)$ for $h > 1$ is given by

$$\sigma^2_T(h) = \omega + \sum_{i=1}^{q} \alpha_i \sigma^2_T(h-i),$$

where $\sigma^2_T(h-i) = y^2_{T+h-i}$ if $h - i \leq 0$.

For the GARCH(1,1) model $\sigma^2_T(h)$ for $h > 1$ is given by

$$\sigma^2_T(h) = \omega + (\alpha_1 + \beta_1)\sigma^2_T(h-1),$$

where the recursion is initialized at $\sigma^2_T(1) = \sigma^2_{T+1}$.

In all cases it can be shown that $\sigma^2_T(h)$ converges to the unconditional variance as h increases, provided that the weak stationarity condition of the model is satisfied. For instance for the GARCH(1,1) model we have that $\sigma^2_T(h)$ converges to $\omega/(1-\beta_1-\alpha_1)$. Therefore, in practical situations, when h is large we can approximate our forecast $\sigma^2_T(h)$ using the unconditional variance.

Finally, we note that $\sigma^2_T(h)$ in practice cannot be directly obtained because the true parameter vector θ_0 is unknown. Therefore what we can do is to use the estimate $\hat{\sigma}^2_T(h)$ obtained plugging in the maximum likelihood estimate in place of θ_0.

5.6 Forecasting VaR and conditional density

Chapters 2 and 3 showed that ARCH and GARCH models provide a description of the conditional density of y_t given its past $Y^{t-1} = \{y_{t-1}, y_{t-2}, ...\}$. In particular, we have noted that, since σ^2_t is given when we condition on the past Y^{t-1}, we obtain

$$y_t | Y^{t-1} \;\sim\; N(0, \sigma^2_t) \; .$$

Now, given a sample of data $\{y_1, ..., y_T\}$, one may naturally ask what is the conditional density of the future return y_{T+1}. This is useful if, for example, we wish to calculate the VaR for the next trading day.

Luckily, since ARCH and GARCH models are *observation-driven*, the value of σ^2_{T+1} is known, conditional on the data $\{y_1, ..., y_T\}$. As such, the conditional density of y_{T+1} is known conditional on $y_1, ..., y_T$, and it is given by

$$y_{T+1} | y_1, ..., y_T \;\sim\; N(0, \sigma^2_{T+1}) \; .$$

Conditional densities for h-step ahead returns, y_{T+h}, can also be obtained through simulations. Here, we shall focus however on the one-step-ahead conditional density only. In particular, we note that the one-step-ahead VaR can be obtained immediately through application of the Gaussian quantile function for a $N(0, \sigma^2_{T+1})$ random variable.

5.7 News Impact Curve

Finally, one interesting piece of information obtained immediately upon estimating the parameters of an ARCH or GARCH model is the so-called *news impact curve* (NIC). Here, we shall the define the NIC as the updating function that maps values of y_t to values of σ^2_{t+1}. In essence, the NIC fixes σ^2_t to some value $\sigma^2_t = c$, and looks at the GARCH update as a function of y_t.

Figure 5.4 below shows the NIC for a GARCH(1,1) model. The curve is obtained for $\omega = 0.1$, $\beta = 0.8$, and three different values of $\alpha = 0.05,\ 0.1$, and 0.2. Moreover, it sets $\sigma^2_t = 1$. Hence, the plot corresponds to the following function,

$$\sigma^2_t(y_t) = 0.9 + \alpha y^2_t \quad \text{for} \quad \alpha = 0.05,\ 0.1,\ \text{and}\ 0.2.$$

The NIC shows how small absolute value of y_t will lead to a decrease in the conditional volatility parameter ($\sigma^2_t > \sigma^2_{t+1}$), and large values of y_t will lead to an increase in the conditional volatility ($\sigma^2_t < \sigma^2_{t+1}$).

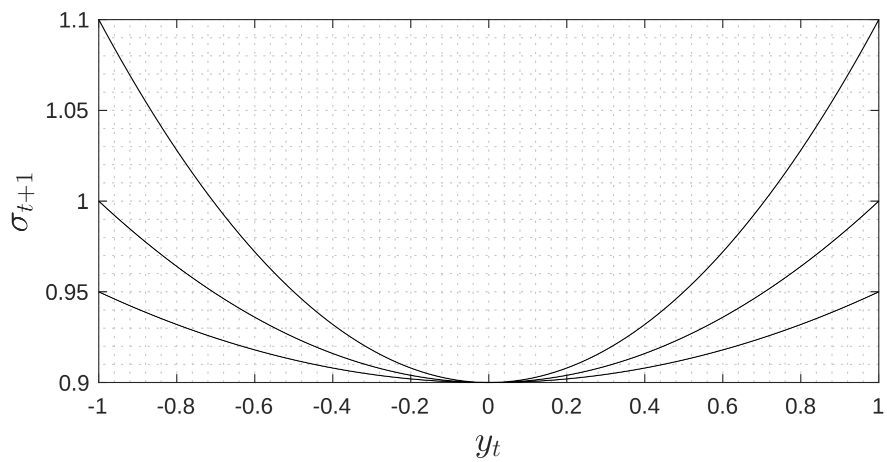

Figure 5.4: News impact curve for GARCH model.

Chapter 6

Multivariate GARCH models

In practical applications very often we deal with problems that involve multiple time series. For instance, we may have several financial assets and be interested in choosing which assets to buy and which assets to sell. Assume that we have n financial assets and the return of each asset i at time t is denoted by $y_{i,t}$, for $i = 1, \ldots, n$. We can represent our data as a multivariate time series $\{\boldsymbol{y}_t\}_{t=1}^{T}$, where $\boldsymbol{y}_t = (y_{1,t}, \cdots, y_{n,t})^\top$. In the previous chapter we have seen univariate GARCH models, so models for a single element $y_{i,t}$ of our multivariate time series. A possible approach could be to use n univariate models and deal with each component $y_{i,t}$ separately. This approach would be appropriate if the series are independent as, in that case, there would not be any loss of information due to the independence. However, empirical evidence suggests that different financial assets are not independent. More specifically, it is well known that financial assets are typically positively correlated. For example this means that when the price of Microsoft increases (decreases) the price of IBM tends to increase (decrease) as well. Figure 6.1 provides us the correlation matrix for all the stock returns in the S&P 100 index. As we can see, most stocks are positively correlated with a correlation coefficient between 0.2 and 0.8. This further highlights the the importance of using multivariate models that can take this dependence among different financial into account.

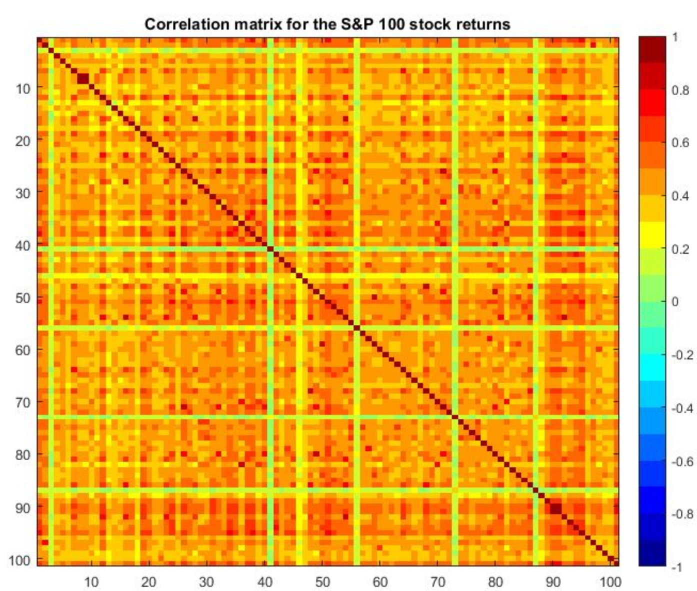

Figure 6.1: Sample correlation matrix of all daily stock log-returns in the S&P 100 index.

As we have discussed in the introduction of the course, past observations contain no information (or very little) to predict the mean of financial assets. However, there is strong evidence of time variation in the variance. As you have seen in previous econometric courses, the variance of a multivariate random variable is a matrix, which is often

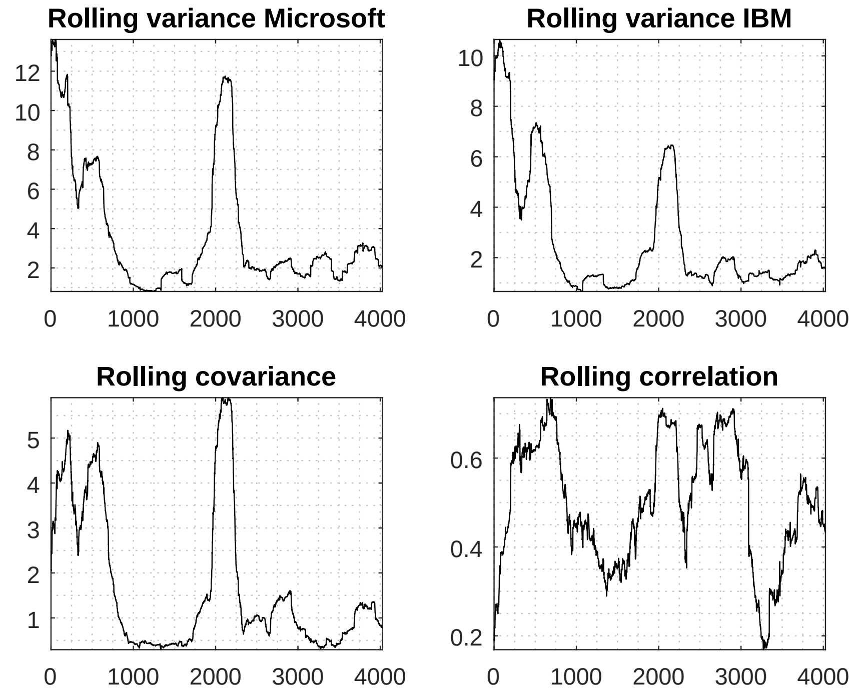

Figure 6.2: Rolling covariance matrix and correlation estimated on a rolling window of length 250 using daily log-returns of Microsoft and IBM.

called the *covariance matrix*. The *covariance matrix* contains the variance of each of the univariate variables in the diagonal and the covariance between each pair of univariate variables outside the diagonal. At this point, the question we want to ask is whether this covariance matrix is time varying and whether past observations contain information to predict the covariance matrix. The answer is: yes, stock returns and more in general financial returns exhibit time varying variances and covariances. To illustrate this fact in practical situations we can estimate the covariance matrix on a *rolling window*. More specifically, given a multivariate sample $\{\boldsymbol{y}_t\}_{t=1}^T$ of length T we can fix a window length h and for each $t > h$ estimate the covariance matrix as follows

$$\hat{\Sigma}_{t-h,t} = \sum_{s=t-h}^{t} \boldsymbol{y}_s \boldsymbol{y}_s^\top .$$

In this way, we have sample estimates of the covariance matrix for $T - h$ time windows and we can check whether there is time variation. Figure 6.2 shows the covariance matrix and the correlation estimated on a rolling window with $h = 250$ using daily log-returns of Microsoft and IBM. The figures suggest (in a qualitative way) that there is indeed evidence of time variation. We can see that the estimated variance level changes dramatically over time. The same happens for the covariance and the correlation. Note also that the correlation is always positive. This is also coherent with the fact discussed before, namely, stock returns are positively correlated.

The time varying variance and covariance between stock returns has to be taken into account in order to properly model the dynamics of multiple financial assets. Therefore we need appropriate multivariate GARCH models that allows us to have time variation in the covariance matrix. In general, the idea is to consider multivariate models of the form

$$\boldsymbol{y}_t = \boldsymbol{\Sigma}_t^{1/2} \boldsymbol{\varepsilon}_t,$$

where $\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \ldots, \varepsilon_{nt})^\top$ is an n-dimensional error vector that is assumed to be i.i.d. with zero mean $\mathbb{E}(\boldsymbol{\varepsilon}_t) = \boldsymbol{0}_n$ and variance equal to the identity matrix $\mathbb{V}ar(\boldsymbol{\varepsilon}_t) = \boldsymbol{I}_n$. More specifically, we will assume that $\boldsymbol{\varepsilon}_t$ has a multivariate normal distribution, namely $\boldsymbol{\varepsilon}_t \sim N_n(\boldsymbol{0}_n, \boldsymbol{I}_n)$. The $n \times n$-dimensional matrix $\boldsymbol{\Sigma}_t$ depends only on the past $Y^{t-1} = \{\boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-2}, \ldots\}$. Similarly as for the univariate case, it can be easily shown that $\boldsymbol{\Sigma}_t$ is the conditional covariance matrix of $\boldsymbol{y}_t$ given the past, i.e. $\boldsymbol{\Sigma}_t = \mathbb{C}ov(\boldsymbol{y}_t|Y^{t-1}) = \mathbb{E}(\boldsymbol{y}_t\boldsymbol{y}_t^\top|Y^{t-1})$. Furthermore, the conditional distribution of $\boldsymbol{y}_t$ given Y^{t-1} is multivariate normal with zero mean and covariance matrix $\boldsymbol{\Sigma}_t$, namely $\boldsymbol{y}_t|Y^{t-1} \sim N_n(\mathbf{0}_n, \boldsymbol{\Sigma}_t)$.

The conditional covariance matrix $\boldsymbol{\Sigma}_t$ has the following form

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \sigma_{1t}^2 & \sigma_{12t} & \cdots & \sigma_{1nt} \\ \sigma_{12t} & \sigma_{2t}^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{(n-1)nt} \\ \sigma_{1nt} & \cdots & \sigma_{(n-1)nt} & \sigma_{nt}^2 \end{bmatrix}.$$

where each diagonal element of the matrix σ_{it}^2 denotes the conditional variance of y_{it}, $i = 1, \ldots, n$, and each other element σ_{ijt} denotes the conditional covariance between y_{it} and y_{jt}, $i, j = 1, \ldots, n$, $i \neq j$.

In the bivariate case $n = 2$ the conditional covariance matrix is given by

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \sigma_{1t}^2 & \sigma_{12t} \\ \sigma_{12t} & \sigma_{2t}^2 \end{bmatrix}.$$

Note the matrix $\boldsymbol{\Sigma}_t$ is symmetric and also positive definite.

In the rest of this section, we will see several multivariate GARCH specifications. These GARCH specifications differ on how the conditional covariance matrix $\boldsymbol{\Sigma}_t$ is specified. Extending univariate GARCH models to a multivariate setting leads to some issues such as the curse of dimensionality, i.e. too many parameters to be estimated. These issues need to be taken into account when specifying multivariate GARCH models.

6.1 The VECH model

The most natural way to extend univariate GARCH models to the multivariate case is given by the VECH model. The specification of the VECH model is based on the half vectorization operator vech$(\cdot)$.

The vech$(\cdot)$ operator stacks the lower triangular elements of a squared matrix into a vector. For instance consider the 3×3 matrix $\boldsymbol{A}$ given by

$$\boldsymbol{A} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix},$$

then

$$\text{vech}(\boldsymbol{A}) = \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \\ \alpha_{22} \\ \alpha_{32} \\ \alpha_{33} \end{bmatrix}.$$

In general, for an $n \times n$ matrix $\boldsymbol{A}$ the operator vech$(\cdot)$ produces a vector of length $\tilde{n} = n(n+1)/2$ that contains all the lower triangular elements of the $n \times n$ matrix.

The idea is to represent the covariance matrix $\boldsymbol{\Sigma}_t$ in vector form using the vech$(\cdot)$ operator. Note that we only need to consider the lower triangular part of $\boldsymbol{\Sigma}_t$ because $\boldsymbol{\Sigma}_t$ is symmetric.

In the bivariate case, i.e. $n = 2$, the 2×2 covariance matrix $\boldsymbol{\Sigma}_t$ of a VECH(1,1) model is specified as

$$\text{vech}(\boldsymbol{\Sigma}_t) = \begin{bmatrix} \sigma_{1,t}^2 \\ \sigma_{12,t} \\ \sigma_{2,t}^2 \end{bmatrix} = \begin{bmatrix} \tilde{\omega}_1 \\ \tilde{\omega}_2 \\ \tilde{\omega}_3 \end{bmatrix} + \begin{bmatrix} \tilde{\beta}_{11} & \tilde{\beta}_{12} & \tilde{\beta}_{12} \\ \tilde{\beta}_{21} & \tilde{\beta}_{22} & \tilde{\beta}_{23} \\ \tilde{\beta}_{31} & \tilde{\beta}_{32} & \tilde{\beta}_{33} \end{bmatrix} \begin{bmatrix} \sigma_{1,t-1}^2 \\ \sigma_{12,t-1} \\ \sigma_{2,t-1}^2 \end{bmatrix} + \begin{bmatrix} \tilde{\alpha}_{11} & \tilde{\alpha}_{12} & \tilde{\alpha}_{13} \\ \tilde{\alpha}_{21} & \tilde{\alpha}_{22} & \tilde{\alpha}_{23} \\ \tilde{\alpha}_{31} & \tilde{\alpha}_{32} & \tilde{\alpha}_{33} \end{bmatrix} \begin{bmatrix} y_{1,t-1}^2 \\ y_{1,t-1}y_{2,t-1} \\ y_{2,t-1}^2 \end{bmatrix} \tag{6.1}$$

The conditional covariance matrix $\mathbf{\Sigma}_t$ depends on past squared observations $y_{1,t-1}^2$ and $y_{2,t-1}^2$ and on the product $y_{1,t-1}y_{2,t-1}$. For the univariate GARCH we have discussed how past squared observations are a natural way to update the time varying variance because a squared observation $y_{1,t-1}^2$ can be seen as an estimate at time $t-1$ of the variance of $y_{1,t-1}^2$. A very similar idea applies for the covariance. In particular, the product $y_{1,t-1}y_{2,t-1}$ can be seen as estimate of the covariance between $y_{1,t-1}$ and $y_{2,t-1}$. This justifies from an intuitive point of view the VECH specification in (6.1).

We also note that the specification of the conditional covariance matrix in (6.1) is very general. For instance, the conditional variance of y_{1t} is given by

$$\sigma_{1t}^2 = \tilde{\beta}_{11}\sigma_{1,t-1}^2 + \tilde{\beta}_{12}\sigma_{12,t-1} + \tilde{\beta}_{12}\sigma_{2,t-1}^2 + \tilde{\alpha}_{11}y_{1,t-1}^2 + \tilde{\alpha}_{12}y_{1,t-1}y_{2,t-1} + \tilde{\alpha}_{13}y_{2,t-1}^2,$$

therefore σ_{1t}^2 depends on all lagged vales of the conditional covariance matrix and also on lagged squared observations $y_{1,t-1}^2$ and $y_{2,t-1}^2$ and on the product $y_{1,t-1}y_{2,t-1}$. However we can see that this specification requires a lot of parameters. Even for this first order and bivariate case we already have 21 parameters to estimate!

In the general multivariate case, the conditional covariance matrix of the VECH(p,q) model with general order p and q is given by

$$\text{vech}(\mathbf{\Sigma}_t) = \tilde{\boldsymbol{W}} + \sum_{i=1}^{q}\tilde{\boldsymbol{A}}_i\text{vech}(\boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top) + \sum_{i=1}^{p}\tilde{\boldsymbol{B}}_i\text{vech}(\mathbf{\Sigma}_{t-i}), \tag{6.2}$$

where $\tilde{\boldsymbol{W}}$ is an $\tilde{n}$-dimensional vector of parameters and $\tilde{\boldsymbol{B}}_i$ and $\tilde{\boldsymbol{A}}_i$ are square matrices of parameters of dimension $\tilde{n}\times\tilde{n}$, where $\tilde{n} = n(n+1)/2$.

Although the conditional covariance matrix $\mathbf{\Sigma}_t$ is time varying, the unconditional covariance matrix of a VECH(p,q) is constant.

Remark 6.1. *The unconditional covariance matrix $\mathbf{\Sigma} = \mathbb{V}ar\,(\boldsymbol{y}_t)$ of the VECH(p,q) model, when it exists, is given by*

$$vech(\mathbf{\Sigma}) = \left(\boldsymbol{I}_{\tilde{n}} - \sum_{i=1}^{q}\tilde{\boldsymbol{A}}_i - \sum_{i=1}^{p}\tilde{\boldsymbol{B}}_i\right)^{-1}\tilde{\boldsymbol{W}}.$$

The VECH model has 2 main drawbacks that typically prevent its use in practical applications. The first is the so called "curse of dimensionality". The problem is that as the dimension n of the vector $\boldsymbol{y}_t$ increases, the number of parameters is of order $O(n^4)$. In particular, the number of parameters to estimate in the VECH(p,q) model is $\tilde{n} + (p+q)\tilde{n}^2$. In practice, even when we have just a few financial assets to model the number of parameters becomes huge. Just to have an idea of the problem, assume that we have $n = 4$ financial assets and we consider $p = q = 1$, then the number of parameters to estimate is 210. If we have $n = 10$ stocks, then we have 6105 parameters. This is completely infeasible in practical applications. The second problem of the VECH model is that it is not clear what type of restrictions we should impose on the parameter matrices $\tilde{\boldsymbol{W}}$, $\tilde{\boldsymbol{B}}_i$ and $\tilde{\boldsymbol{A}}_i$ to ensure that $\mathbf{\Sigma}_t$ is a positive definite matrix for any t.

6.2 The DVECH model

A possible solution to solve the "curse of dimensionality" problem of the VECH model is to impose that the matrices $\tilde{\boldsymbol{B}}_i$ and $\tilde{\boldsymbol{A}}_i$ are diagonal. This leads to the diagonal VECH model, which is often called the DVECH model. The DVECH model is a special case of the VECH model.

For the Bivariate case, i.e. $n = 2$, the conditional covariance matrix of the DVECH(1,1) model is given by

$$\begin{bmatrix}\sigma_{1,t}^2\\ \sigma_{21,t}\\ \sigma_{2,t}^2\end{bmatrix} = \begin{bmatrix}\tilde{\omega}_1\\ \tilde{\omega}_2\\ \tilde{\omega}_3\end{bmatrix} + \begin{bmatrix}\tilde{\beta}_{11} & 0 & 0\\ 0 & \tilde{\beta}_{22} & 0\\ 0 & 0 & \tilde{\beta}_{33}\end{bmatrix}\begin{bmatrix}\sigma_{1,t-1}^2\\ \sigma_{21,t-1}\\ \sigma_{2,t-1}^2\end{bmatrix} + \begin{bmatrix}\tilde{\alpha}_{11} & 0 & 0\\ 0 & \tilde{\alpha}_{22} & 0\\ 0 & 0 & \tilde{\alpha}_{33}\end{bmatrix}\begin{bmatrix}y_{1,t-1}^2\\ y_{1,t-1}y_{2,t-1}\\ y_{2,t-1}^2\end{bmatrix} \tag{6.3}$$

Compared to bivariate VECH(1,1) model, the DVECH specification in (6.3) reduces the number of parameters from 21 to 9. The DVECH allows us to attenuate the "curse of dimensionality" problem, however, this comes at a price. The DVECH model does not allow for causality in variance. This can be noted from the expressions of the conditional variances $\sigma^2_{1,t}$, $\sigma^2_{2,t}$ and the covariance $\sigma_{12,t}$ that are given by

$$\begin{aligned}
\sigma^2_{1,t} &= \omega_{11} + \beta_{11}\sigma^2_{1,t-1} + \alpha_{11} y^2_{1,t-1},\\
\sigma^2_{2,t} &= \omega_{22} + \beta_{22}\sigma^2_{2,t-1} + \alpha_{22} y^2_{2,t-1},\\
\sigma_{21,t} &= \omega_{21} + \beta_{21}\sigma^2_{21,t-1} + \alpha_{21} y_{1,t-1} y_{2,t-1}
\end{aligned}$$

The variances $\sigma^2_{1,t}$ and $\sigma^2_{2,t}$ depend only on their past values $y^2_{1,t-1}$ and $y^2_{2,t-1}$ respectively. Therefore, large values of $y^2_{1,t-1}$ does not have any effect on $y^2_{2,t}$.

The DVECH model in (6.3) can be also rewritten in matrix form by relying on the Hadamard matrix product.

The Hadamard product $\odot$ is a matrix operation that given 2 matrices of the same dimension produces a matrix where each element ij is the product of the elements ij of the two original matrices. For instance, given two 3×3 matrices $\boldsymbol{A} = (\alpha_{ij})$ and $\boldsymbol{B} = (\beta_{ij})$, we have that

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} \odot \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \\ \beta_{31} & \beta_{32} & \beta_{33} \end{bmatrix} = \begin{bmatrix} \alpha_{11}\beta_{11} & \alpha_{12}\beta_{12} & \alpha_{13}\beta_{13} \\ \alpha_{21}\beta_{21} & \alpha_{22}\beta_{22} & \alpha_{23}\beta_{23} \\ \alpha_{31}\beta_{31} & \alpha_{32}\beta_{32} & \alpha_{33}\beta_{33} \end{bmatrix}.$$

The bivariate DVECH(1,1) conditional covariance matrix in (6.3) can be rewritten in matrix form as

$$\begin{bmatrix} \sigma^2_{1,t} & \sigma_{21,t} \\ \sigma_{21,t} & \sigma^2_{2,t} \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{21} \\ \omega_{21} & \omega_{22} \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{21} \\ \beta_{21} & \beta_{22} \end{bmatrix} \odot \begin{bmatrix} \sigma^2_{1,t-1} & \sigma_{21,t-1} \\ \sigma_{21,t-1} & \sigma^2_{2,t-1} \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{21} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \odot \begin{bmatrix} y^2_{1,t-1} & y_{1,t-1}y_{2,t-1} \\ y_{1,t-1}y_{2,t-1} & y^2_{2,t-1} \end{bmatrix}$$

Note that all the the matrices in the above formulation are symmetric because the covariance matrix must be symmetric.

Figure 6.3 shows the conditional variances, covariance and correlation generated from a bivariate DVECH model.

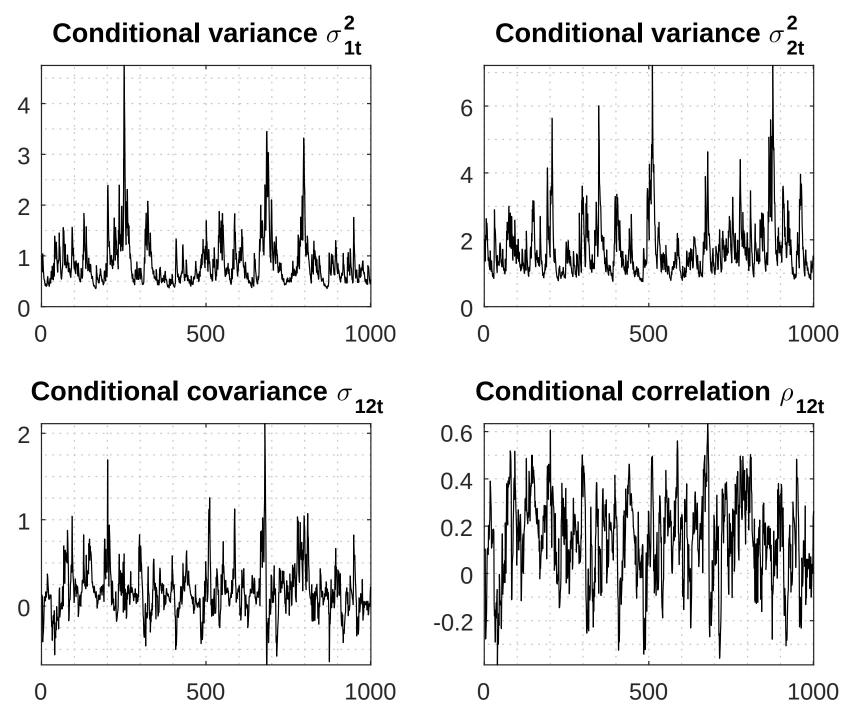

Figure 6.3: Conditional variances, covariance and correlation generated from a DVECH model.

A general multivariate DVECH(1,1) model can thus be written as follows

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \boldsymbol{A}_1 \odot (\boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top) + \boldsymbol{B}_1 \odot \boldsymbol{\Sigma}_{t-1} \ , \tag{6.4}$$

where $\boldsymbol{W}$, $\boldsymbol{B}_1$ and $\boldsymbol{A}_1$ are symmetric $n \times n$ matrices containing the parameters to be estimated. A natural extension to the VECH(1,1) model is the VECH(p,q) model. The conditional covariance matrix of a n-variate DVECH(p,q) model is given by

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \sum_{i=1}^{q} \boldsymbol{A}_i \odot (\boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top) + \sum_{i=1}^{p} \boldsymbol{B}_i \odot \boldsymbol{\Sigma}_{t-i}, \tag{6.5}$$

where $\boldsymbol{W}$, $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ are symmetric $n \times n$ matrices containing the parameters to be estimated.

The DVECH model is in fact a special case of the VECH model where the matrices of parameters are imposed to be diagonal. This constraint attenuates the "curse of dimensionality". However, a limitation of this model is that it does not ensure that the conditional covariance $\boldsymbol{\Sigma}_t$ be positive definite.

6.3 The scalar DVECH model

A very simple version of the DVECH model is the scalar DVECH (sDVECH) model. The sDVECH imposes that the parameters in each of the matrices $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ are the same. The conditional covariance matrix of an n-dimensional sDVECH(1,1) can be written as follows

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \alpha_1 \, \boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top + \beta_1 \, \boldsymbol{\Sigma}_{t-1} \ , \tag{6.6}$$

where $\boldsymbol{W}$ is a symmetric $n \times n$ matrix, and α_1 and β_1 are scalar parameters to be estimated. The name "scalar" is due to the fact that the matrices $\boldsymbol{B}_i$ and $\boldsymbol{A}_i$ are replaced by scalars. The sDVECH(1,1) model in (6.6) is a special case of the DVECH(1,1) model in (6.4) with matrices $\boldsymbol{B}_1$ and $\boldsymbol{A}_1$ given by

$$\boldsymbol{B}_1 = \begin{bmatrix} \beta_1 & \dots & \beta_1 \\ \vdots & \ddots & \vdots \\ \beta_1 & \dots & \beta_1 \end{bmatrix}, \qquad \boldsymbol{A}_1 = \begin{bmatrix} \alpha_1 & \dots & \alpha_1 \\ \vdots & \ddots & \vdots \\ \alpha_1 & \dots & \alpha_1 \end{bmatrix}.$$

The sDVECH model is widely used in practice when the number of stocks n is large. The sDVECH model strongly attenuates the curse of dimensionality. The number of parameters of the sDVECH(1,1) in (6.6) is $n(n+1)/2 + 2$.

The extension from the sDVECH(1,1) to the sDVECH(p,q) is straightforward. In particular, the conditional covariance matrix of the sDVECH(p,q) is given by

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \sum_{i=1}^{q} \alpha_i \, \boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top + \sum_{i=1}^{p} \beta_i \, \boldsymbol{\Sigma}_{t-i}. \tag{6.7}$$

where $\boldsymbol{W}$ is a symmetric $n \times n$ matrix, and $\{\alpha_1, \dots, \alpha_q\}$ and $\{\beta_1, \dots, \beta_p\}$ are scalar parameters to be estimated.

6.4 The BEKK model

In order to solve the issue of having a positive definite conditional covariance matrix $\boldsymbol{\Sigma}_t$, the BEKK model has been proposed. Also the BEKK is a special case of the more general VECH model. In the bivariate case, the conditional covariance matrix of the BEKK(1,1) model is given by

$$\begin{aligned} \begin{bmatrix} \sigma_{1,t}^2 & \sigma_{21,t} \\ \sigma_{21,t} & \sigma_{2,t}^2 \end{bmatrix} = & \begin{bmatrix} \omega_{11} & 0 \\ \omega_{21} & \omega_{22} \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{21} \\ 0 & \omega_{22} \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} \sigma_{1,t-1}^2 & \sigma_{21,t-1} \\ \sigma_{21,t-1} & \sigma_{2,t-1}^2 \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{21} \\ \beta_{12} & \beta_{22} \end{bmatrix} + \\ & + \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} y_{1,t-1}^2 & y_{1,t-1}y_{2,t-1} \\ y_{1,t-1}y_{2,t-1} & y_{2,t-1}^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{21} \\ \alpha_{12} & \alpha_{22} \end{bmatrix} \end{aligned} \tag{6.8}$$

This specification allows us to easily obtain a positive definite covariance matrix $\boldsymbol{\Sigma}_t$, see Remark 6.2 for conditions in a more general case. The updating equation for the conditional covariance matrix of a multivariate BEKK(1,1) model is thus given by

$$\boldsymbol{\Sigma}_t = \boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{A}_1(\boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top)\boldsymbol{A}_1^\top + \boldsymbol{B}_1\boldsymbol{\Sigma}_{t-1}\boldsymbol{B}_1^\top, \tag{6.9}$$

The extension from the BEKK(1,1) to the BEKK(p,q) model is also trivial. In particular, the conditional covariance matrix of a BEKK(p,q) model is updated as follows,

$$\boldsymbol{\Sigma}_t = \boldsymbol{W}\boldsymbol{W}^\top + \sum_{i=1}^{q}\boldsymbol{A}_i(\boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top)\boldsymbol{A}_i^\top + \sum_{i=1}^{p}\boldsymbol{B}_i\boldsymbol{\Sigma}_{t-i}\boldsymbol{B}_i^\top, \tag{6.10}$$

where $\boldsymbol{W}$ is a lower triangular $n \times n$ matrix and $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are $n \times n$ matrices. The conditions to ensure that $\boldsymbol{\Sigma}_t$ is positive definite are easy to be met for the BEKK model.

Remark 6.2. *The BEKK(p,q) model in (6.10) has a positive definite conditional covariance matrix $\boldsymbol{\Sigma}_t$ for any $t \in \mathbb{N}$ if $\boldsymbol{\Sigma}_0, \dots, \boldsymbol{\Sigma}_{-p-1}$ are positive definite and $\boldsymbol{W}$ or any $\boldsymbol{B}_i$ is a full rank matrix.*

As already discussed, the advantage of the BEKK model is that we can ensure the covariance matrix to be positive definite. One of the disadvantages of the BEKK formulation is that the parameters are difficult to interpret.

6.5 The CCC model

Another approach to deal with the curse of dimensionality is provided by the Constant Conditional Correlation (CCC) model. As the name suggests, the peculiarity of this model is that the conditional correlation matrix is constant and the time variation in the conditional covariance matrix $\boldsymbol{\Sigma}_t$ is only provided by dynamic variances. This model, unlike the DVECH and the BEKK, is not a special case of the very general VECH model.

The conditional covariance matrix of a bivariate CCC model is given by

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \sigma_{1t}^2 & \sigma_{12t} \\ \sigma_{12t} & \sigma_{2t}^2 \end{bmatrix} = \begin{bmatrix} \sigma_{1t} & 0 \\ 0 & \sigma_{2t} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix} \begin{bmatrix} \sigma_{1t} & 0 \\ 0 & \sigma_{2t} \end{bmatrix}, \tag{6.11}$$

where the conditional variances $\sigma_{1,t}^2$ and $\sigma_{2,t}^2$ are specified as

$$\sigma_{1,t}^2 = \omega_1 + \beta_1\sigma_{1,t-1}^2 + \alpha_1 y_{1,t}^2, \tag{6.12}$$

$$\sigma_{2,t}^2 = \omega_2 + \beta_2\sigma_{2,t-1}^2 + \alpha_2 y_{2,t}^2. \tag{6.13}$$

This model is called CCC because the conditional correlation matrix $\boldsymbol{R}$ is constant

$$\boldsymbol{R} = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{bmatrix}.$$

In particular, the conditional correlation between y_{1t} and y_{2t} is given by $\mathbb{C}orr(y_{1t}, y_{2t}|Y^{t-1}) = \rho_{12}$. This does not mean that the conditional covariance is constant. In fact, the conditional covariance is time varying $\mathbb{C}ov(y_{1t}, y_{2t}|Y^{t-1}) = \sigma_{12t} = \sigma_{1t}\sigma_{2t}\rho_{12}$. Figure 6.4 below shows the conditional variances $\sigma_{1,t}^2$ and $\sigma_{2,t}^2$, the conditional covariance σ_{12t} and the conditional correlation ρ_{12} generated from a CCC model. The pictures clearly illustrate how the conditional covariance is time varying even though the conditional variance is constant.

The bivariate CCC model can be easily extended to a general multivariate case of order n. The conditional covariance matrix of an n-variate CCC model is given by

$$\boldsymbol{\Sigma}_t = \boldsymbol{D}_t\boldsymbol{R}\boldsymbol{D}_t, \tag{6.14}$$

where $\boldsymbol{R}$ is an $n \times n$ correlation matrix and $\boldsymbol{D}_t = \mathrm{diag}(\sigma_{1,t}, \dots, \sigma_{n,t})$ is an $n \times n$ diagonal matrix containing the conditional standard deviation of each component. The variance of each component $\sigma_{i,t}^2$, $i = 1, \dots, n$, follows a GARCH(1,1) dynamic, namely

$$\sigma_{i,t}^2 = \omega_i + \beta_i\sigma_{i,t-1}^2 + \alpha_i y_{i,t}^2.$$

The CCC model is very appealing because it handles both the curse of dimensionality and the positive definiteness of $\boldsymbol{\Sigma}_t$. However, the assumption that the conditional correlation matrix is constant can be very restrictive. In practical applications, we often see evidence of changing conditional correlation.

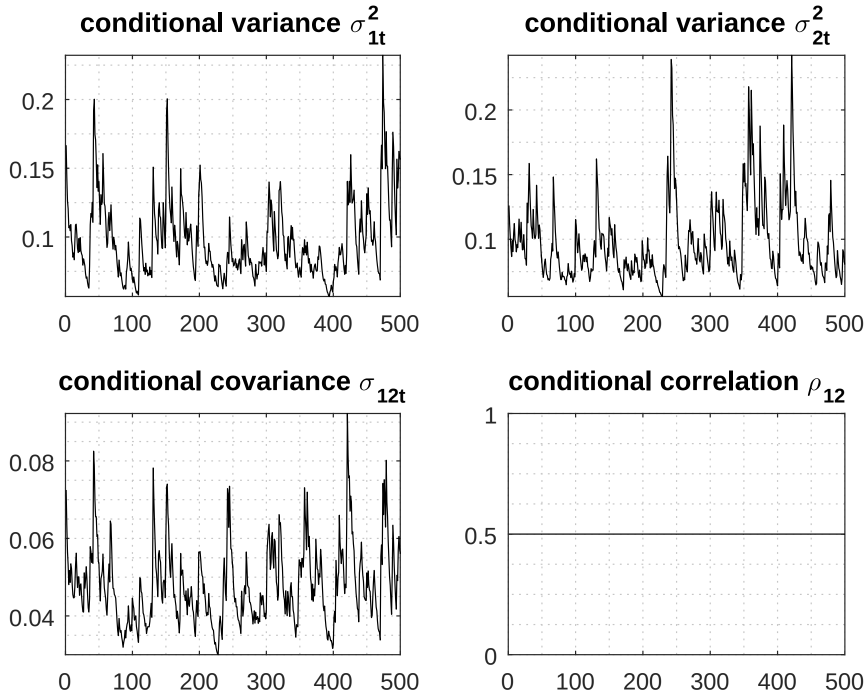

Figure 6.4: Conditional variances, covariance and correlation generated from a CCC model.

6.6 The DCC model

The assumption of constant conditional correlation is often too restrictive. For this reason, the Dynamic Conditional Correlation (DCC) model has been introduced. The specification of the DCC is similar to the CCC the difference lies on the fact that the conditional correlation matrix $\boldsymbol{R}$ varies over time.

The conditional covariance matrix of the bivariate DCC model is given by

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} \sigma^2_{1t} & \sigma_{12t} \\ \sigma_{12t} & \sigma^2_{2t} \end{bmatrix} = \begin{bmatrix} \sigma_{1t} & 0 \\ 0 & \sigma_{2t} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12t} \\ \rho_{12t} & 1 \end{bmatrix} \begin{bmatrix} \sigma_{1t} & 0 \\ 0 & \sigma_{2t} \end{bmatrix}, \tag{6.15}$$

where the conditional variances $\sigma^2_{1,t}$ and $\sigma^2_{2,t}$ are specified as

$$\sigma^2_{1,t} = \omega_1 + \beta_1 \sigma^2_{1,t-1} + \alpha_1 y^2_{1,t},$$
$$\sigma^2_{2,t} = \omega_2 + \beta_2 \sigma^2_{2,t-1} + \alpha_2 y^2_{2,t}.$$

So far the specification is the same as the CCC model but we are now left with the specification of the dynamic conditional correlation ρ_{12t}. First, we need to define the standardized observations $v_{1t} = y_{1t}/\sigma_{1t}$ and $v_{2t} = y_{2t}/\sigma_{2t}$. The conditional variances of v_{1t} and v_{2t} are equal to 1, i.e. $\mathbb{V}ar(v_{1t}|Y^{t-1}) = \mathbb{V}ar(v_{2t}|Y^{t-1}) = 1$ and the conditional covariance (correlation) between v_{1t} and v_{2t} is given by $\mathbb{E}(v_{1t}v_{2t}|Y^{t-1}) = \rho_{12t}$. Note that v_{1t} and v_{2t} are different from the errors ε_{1t} and ε_{2t}. Given the definition of v_{1t} and v_{2t}, we are now ready to specify the conditional correlation as

$$\rho_{12t} = \frac{q_{12t}}{\sqrt{q_{11t}}\sqrt{q_{22t}}}, \tag{6.16}$$

where

$$q_{11t} = \omega_q + \beta_q q_{11t-1} + \alpha_q v^2_{1,t-1},$$
$$q_{22t} = \omega_q + \beta_q q_{22t-1} + \alpha_q v^2_{2,t-1},$$
$$q_{12t} = \omega_q + \beta_q q_{12t-1} + \alpha_q v_{1,t-1} v_{2,t-1}.$$

The formulation in (6.16) for the conditional correlation is needed to ensure that ρ_{12t} is between -1 and 1. Note that each equation for q_{ijt} has the same static parameters ω_q, β_q and α_q.

6.7 Other extensions

All the multivariate GARCH models presented in this section can be extended to include a mean different from zero or also a time varying conditional mean. For instance, including a constant mean the multivariate GARCH model becomes

$$\boldsymbol{y}_t = \boldsymbol{\mu} + \boldsymbol{\Sigma}_t^{1/2} \boldsymbol{\varepsilon}_t, \tag{6.17}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$ is an n-dimensional vector containing the means of each element $y_{i,t}$. The specification of $\boldsymbol{\Sigma}_t$ can then be one of those discussed in the previous section.

Finally, we mention that there are several different specifications that we have not discussed in this course such as the Factor GARCH model and Multivariate GAS models.

6.8 Simulate from a bivariate DVECH(1,1) with R

This section describes how to simulate from a bivariate DVECH model with $p = q = 1$ using R. The code can be found on the R file `generate_DVECH.R`. The first step is to set the sample size T, which is labeled `n`, and choose the parameter values. For simplicity, we use the notation that the parameters ω_{11}, ω_{22} and ω_{12} are labeled `w11`, `w22`, `w12`. The same is true for the β_{ij} and α_{ij} parameters where we use `bij` and `aij`, for $i, j = 1, 2$.

```
n <- 1000

w11 <- 0.1
w22 <- 0.2
w12 <- 0.02

b11 <- 0.7
b22 <- 0.7
b12 <- 0.7

a11 <- 0.2
a22 <- 0.2
a12 <- 0.15
```

The second step is to define the matrices `x` and `VECHt` that will contain the generated series $\{\boldsymbol{y}_t\}_{t=1}^T$ and the generated conditional covariance matrix $\{\text{vech}(\boldsymbol{\Sigma}_t)\}_{t=1}^T$. The initial value $\text{vech}(\boldsymbol{\Sigma}_1)$ of the conditional covariance matrix is set to be equal to the unconditional covariance matrix.

```
x <- matrix(0,nrow = n, ncol = 2)
VECHt <- matrix(0,nrow=n,ncol=3)

VECHt[1,1] <- w11/(1-b11-a11)
VECHt[1,3] <- w22/(1-b22-a22)
VECHt[1,2] <- w12/(1-b12-a12)
```

Next, we generate the first observation y_1. The R function `mvrnorm()` is used to generate from a multivariate normal distribution. This function `mvrnorm()` is part of the package `MASS`. Once we have $\boldsymbol{\Sigma}_1$, we can generate y_1 from a bivariate normal with mean $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}_1$. The first argument of `mvnrnd()` is the number of observations to generate (1 in our case), the second argument is the mean (a vector of zeros) and the third argument is the covariance matrix. Therefore, we first need to get $\text{vech}(\boldsymbol{\Sigma}_1)$ in matrix form. For this reason we define a 2 matrix labeled `SIGMAt` where we put in matrix form $\text{vech}(\boldsymbol{\Sigma}_1)$.

```
SIGMAt <- cbind(c(VECHt[1,1],VECHt[1,2]),c(VECHt[1,2],VECHt[1,3]))
x[1,] <- mvrnorm(1,rep(0,2),SIGMAt)
```

Finally, we are ready to use a *for loop* to obtain the generated series. We iterate the equation of vech($\mathbf{\Sigma}_t$) of the bivariate DVECH model with the observation equation for t from 2 to T.

```
for(t in 2:n){
  VECHt[t,1] <- w11 + b11*VECHt[t-1,1] + a11*x[t-1,1]^2
  VECHt[t,3] <- w22 + b22*VECHt[t-1,3] + a22*x[t-1,2]^2
  VECHt[t,2] <- w12 + b12*VECHt[t-1,2] + a12*x[t-1,1]*x[t-1,2]

  SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))
  x[t,] <- mvrnorm(1,rep(0,2),SIGMAt)
}
```

Figure 6.5 shows a generated series from a bivariate DVECH model.

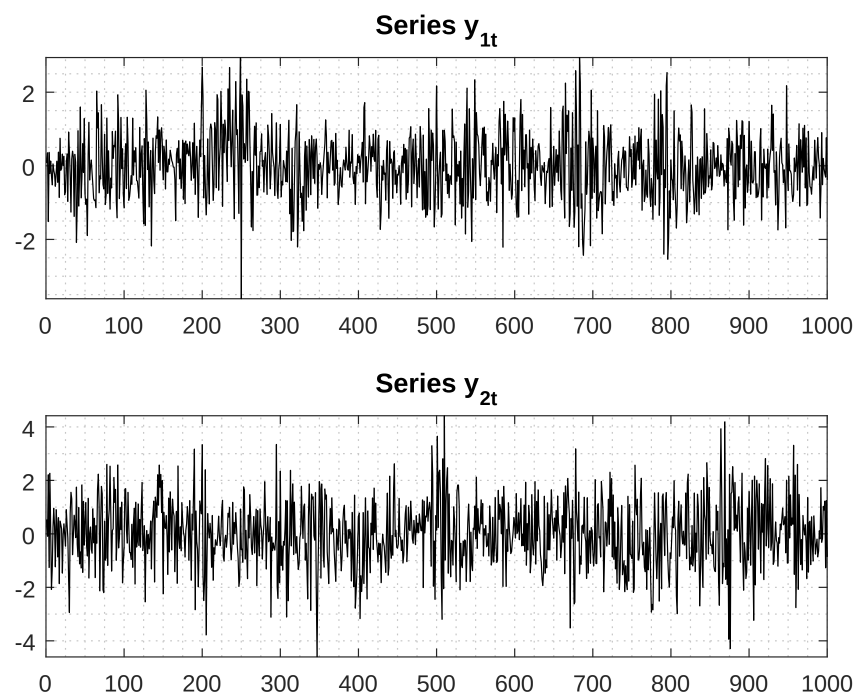

Figure 6.5: Series generated from a bivariate DVECH model.

Chapter 7

Estimation of multivariate GARCH models

7.1 Maximum likelihood estimation

Multivariate GARCH models can be estimated by maximum likelihood. The log-likelihood function is given by

$$L(y_1, ..., y_T, \theta) = -\frac{1}{2}\sum_{t=1}^{T}\Big(\log|\boldsymbol{\Sigma}_t| + \boldsymbol{y}_t^\top \boldsymbol{\Sigma}_t^{-1}\boldsymbol{y}_t\Big)\,.$$

The covariance matrix $\boldsymbol{\Sigma}_t$ is obtained recursively using the the observed data and the updating equation of the specific multivariate GARCH we are estimating. For instance, for a bivariate VECH model with $p = q = 1$ we can use the following updating equations to obtain $\boldsymbol{\Sigma}_t$

$$\begin{aligned}
\sigma^2_{1,t} &= \omega_{11} + \beta_{11}\sigma^2_{1,t-1} + \alpha_{11}y^2_{1,t-1},\\
\sigma^2_{2,t} &= \omega_{22} + \beta_{22}\sigma^2_{2,t-1} + \alpha_{22}y^2_{2,t-1},\\
\sigma_{21,t} &= \omega_{21} + \beta_{21}\sigma^2_{21,t-1} + \alpha_{21}y_{1,t-1}y_{2,t-1},
\end{aligned}$$

for $t = 1, \dots, T$. As for the univariate case, the updating equations need to be initialized. A practical way is to set the initial condition $\boldsymbol{\Sigma}_1$ equal to the sample covariance matrix.

Once the log-likelihood function is obtained, the estimation of a multivariate GARCH is equivalent to the estimation of a univariate GARCH. In particular, the ML estimator $\hat{\theta}_T$ is the maximizer of the log-likelihood function

$$\hat{\theta}_T = \arg\max_{\theta\in\Theta} L(y_1, ..., y_T, \theta).$$

Furthermore, in large samples, the following approximation for the distribution of $\hat{\theta}_T$ holds true

$$\sqrt{T}\Big(\hat{\theta}_T - \theta_0\Big) \overset{app}{\sim} N\Big(0\ ,\ \mathcal{I}(\theta_0)^{-1}\Big).$$

See the section on the estimation of univariate GARCH models for more details and how to estimate the Fisher information matrix $\mathcal{I}(\theta_0)$.

7.2 Estimating a bivariate scalar DVECH with R

In this section, we will see how to estimate a bivariate sDVECH model by maximum likelihood using R. More specifically, we are only going to discuss how to write the log-likelihood function because the optimization is then equivalent to the univariate GARCH case (see Section 4.5 see how to optimize the likelihood). The R file `estimation_sDVECH.R` contains the code to optimize the log-likelihood function of the sDVECH model. The log-likelihood function is labeled `llik_fun_sDVECH(,)` and it is contained in the file `llik_fun_sDVECH.R`. We need to create an R function that takes as argument the observed time series, labeled `x`, and a parameter vector, labeled `par`, and gives as output the average log-likelihood value.

The first line of code defines the name of the function and the input.

```
llik_fun_sDVECH <- function(par,x){
```

Then, the time series length is obtained and each parameter value is set equal to an element of the input parameter vector `par` using some appropriate link functions. Furthermore, the matrix `VECHt` that will contain the conditional covariance matrix is defined.

```
  w11 <- exp(par[1])
  w12 <- par[2]
  w22 <- exp(par[3])
  a <- exp(par[4])/(1+exp(par[4]))
  b <- exp(par[5])/(1+exp(par[5]))

  d <- dim(x)
  n <- d[1]

  VECHt <- matrix(0,nrow=n,ncol=3)
```

Now, the conditional variance is initialized using the sample covariance matrix. The average log-likelihood output `llik` is defined and set to zero.

```
  llik <- 0

  C <- cov(x)
  VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])
```

Finally, a *for loop* allows us to obtain recursively the conditional covariance matrix using the updating equation of the sDVECH model. Furthermore, the average log-likelihood is summed up at each iteration of the loop (see last line of code in the *for loop*). Note that in R matrix product is obtained with the operator `%*%`. The average log-likelihood is then returned as output of the function.

```
   for(t in 2:n){

    VECHt[t,1] <- w11+b*VECHt[t-1,1]+a*x[t-1,1]^2
    VECHt[t,3] <- w22+b*VECHt[t-1,3]+a*x[t-1,2]^2
    VECHt[t,2] <- w12+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]

    SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))
    llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
  }

  return(llik)
}
```

The full code to obtain the log-likelihood function is given below.

```
llik_fun_sDVECH <- function(par,x){

  w11 <- exp(par[1])
  w12 <- par[2]
  w22 <- exp(par[3])
```

```
  a <- exp(par[4])/(1+exp(par[4]))
  b <- exp(par[5])/(1+exp(par[5]))

  d <- dim(x)
  n <- d[1]

  VECHt <- matrix(0,nrow=n,ncol=3)
  llik <- 0

  C <- cov(x)
  VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])

  for(t in 2:n){

    VECHt[t,1] <- w11+b*VECHt[t-1,1]+a*x[t-1,1]^2
    VECHt[t,3] <- w22+b*VECHt[t-1,3]+a*x[t-1,2]^2
    VECHt[t,2] <- w12+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]

    SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))
    llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
  }

  return(llik)
}
```

Figure 7.1 provides a plot of conditional volatility and conditional correlations between the stock returns of Google and IBM.

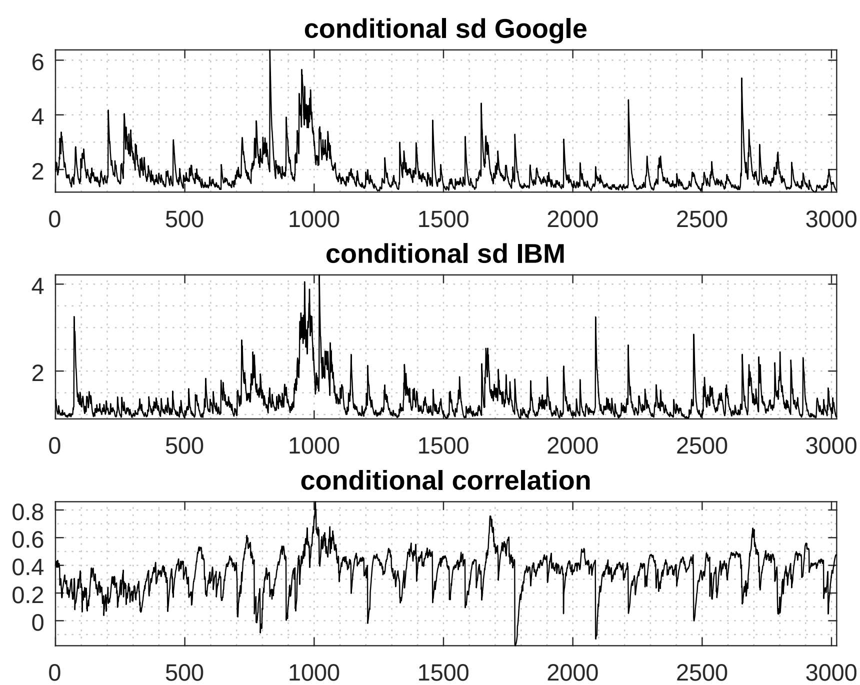

Figure 7.1: Conditional variances and correlations estimated from IBM and Google log-returns.

7.3 Estimation of the sDVECH model with covariance targeting

One of the main issues in estimating multivariate GARCH models is that the large number of parameters can cause numerical problems when optimizing the log-likelihood function. **Covariance targeting** is a 2-steps method that allows us to reduce the number of parameters in the likelihood optimization. In particular, in a first step, the unconditional covariance $\boldsymbol{\Sigma}$ is estimated using the sample covariance of the log-returns. Then, in a second step, the estimated unconditional covariance is plugged-in into the log-likelihood function and the remaining parameters are estimated by optimizing the resulting log-likelihood. In the following, we focus on covariance targeting for the sDVECH model, but we note that covariance targeting can applied to VECH models in general.

The first step to perform covariance targeting is to reparametrize the model in terms of the unconditional covariance. As discussed before, the conditional covariance matrix $\boldsymbol{\Sigma}_t = \mathbb{V}ar(\boldsymbol{y}_t|Y^{t-1})$ of the sDVECH(p,q) is given by

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \sum_{i=1}^{q} \alpha_i\, \boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top + \sum_{i=1}^{p} \beta_i\, \boldsymbol{\Sigma}_{t-i}. \tag{7.1}$$

Furthermore, it can be shown that the unconditional covariance matrix $\boldsymbol{\Sigma} = \mathbb{V}ar(\boldsymbol{y}_t) = \mathbb{E}(\boldsymbol{y}_t\boldsymbol{y}_t^\top)$ of the sDVECH(p,q) model is given by

$$\boldsymbol{\Sigma} = \left(1 - \sum_{i=1}^{q} \alpha_i - \sum_{i=1}^{p} \beta_i\right)^{-1} \boldsymbol{W}.$$

Therefore, the matrix $\boldsymbol{W}$ in (7.1) can be expressed as

$$\boldsymbol{W} = \left(1 - \sum_{i=1}^{q} \alpha_i - \sum_{i=1}^{p} \beta_i\right) \boldsymbol{\Sigma}.$$

The unconditional variance $\boldsymbol{\Sigma}$ can be estimated using the observed data as follows: $\hat{\boldsymbol{\Sigma}} = T^{-1}\sum_{t=1}^{T} \boldsymbol{y}_t\boldsymbol{y}_t^\top$. Finally, we can plug-in the expression of $\boldsymbol{W}$ with the estimated $\hat{\boldsymbol{\Sigma}}$ into expression (7.1) and obtain

$$\boldsymbol{\Sigma}_t = \left(1 - \sum_{i=1}^{q} \alpha_i - \sum_{i=1}^{p} \beta_i\right) \hat{\boldsymbol{\Sigma}} + \sum_{i=1}^{q} \alpha_i\, \boldsymbol{y}_{t-i}\boldsymbol{y}_{t-i}^\top + \sum_{i=1}^{p} \beta_i\, \boldsymbol{\Sigma}_{t-i}. \tag{7.2}$$

The updating equation in (7.2) is used to obtain the log-likelihood function. The parameters $\{\alpha_1, \ldots, \alpha_q\}$ and $\{\beta_1, \ldots, \beta_p\}$ can be estimated by maximum likelihood as discussed in Section 7.1. In this way, we are left with only $p + q$ parameters to be estimated in the likelihood optimization. This is very appealing because the number of parameters in the optimization does not depend on the number of stocks. This makes estimation much more reliable and fast.

The covariance targeting approach is summarized by the following steps:

1. Estimate the unconditional covariance $\boldsymbol{\Sigma}$ from the log-returns $\hat{\boldsymbol{\Sigma}} = T^{-1}\sum_{t=1}^{T} \boldsymbol{y}_t\boldsymbol{y}_t^\top$, where $\boldsymbol{y}_t = (y_{1t}, \ldots, y_{nt})^\top$.
2. Obtain the log-likelihood function using the updating equation given in (7.2).
3. Maximize the log-likelihood function to estimate the parameters $\{\alpha_1, \ldots, \alpha_q\}$ and $\{\beta_1, \ldots, \beta_p\}$.

7.4 Estimating a sDVECH model with CT in R

In the following, we illustrate how to estimate a sDVECH(1,1) model by covariance targeting with R. We are going to see how the likelihood function can be obtained by incorporating the estimated covariance $\hat{\boldsymbol{\Sigma}}$. The resulting likelihood function can then be optimized with respect to the parameters β_1 and α_1 (see Section 4.5 on how to optimize the likelihood). The R file `CT_estimation_sDVECH.R` contains the code to optimize the log-likelihood function of the sDVECH model with covariance targeting. The log-likelihood function with covariance targeting is labeled `llik_CT_sDVECH(,)` and it is contained in the file `llik_CT_sDVECH.R`. We need to create an R function that takes as argument the observed time series, labeled `x`, and a parameter vector that contains β_1 and α_1, labeled `par`, and gives as output the average log-likelihood value.

The first line of code defines the name of the function and the input.

```
llik_CT_sDVECH <- function(par,x){
```

Then, the time series length is obtained and each parameter value is set equal to an element of the input parameter vector `par` using appropriate link functions. Note that the only parameters that enter into the likelihood are β_1 and α_1 since we are using the covariance targeting approach. Furthermore, the matrix `VECHt` that will contain the conditional covariance matrix is defined and the average log-likelihood output `llik` is defined and set to zero.

```
  a <- exp(par[1])/(1+exp(par[1]))
  b <- exp(par[2])/(1+exp(par[2]))

  d <- dim(x)
  n <- d[1]

  VECHt <- matrix(0,nrow=n,ncol=3)
  llik <- 0
```

Now, the sample covariance of the observed data is obtained. Furthermore, the conditional covarince matrix is initialized using the sample covariance matrix.

```
  C <- cov(x)
  VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])
```

Finally, a *for loop* allows us to obtain recursively the conditional covariance matrix using the updating equation of the sDVECH model. Note that $\boldsymbol{W}$ is replaced by $\hat{\boldsymbol{\Sigma}}(1-\beta_1-\alpha_1)$ as described before in the application of the covariance targeting approach. Furthermore, the average log-likelihood is summed up at each iteration of the loop (see last line of code in the *for loop*).

```
  for(t in 2:n){

    VECHt[t,1] <- C[1,1]*(1-a-b)+b*VECHt[t-1,1]+a*x[t-1,1]^2
    VECHt[t,3] <- C[2,2]*(1-a-b)+b*VECHt[t-1,3]+a*x[t-1,2]^2
    VECHt[t,2] <- C[1,2]*(1-a-b)+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]

    SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))

    llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
  }

  return(llik)
```

The full code to obtain the log-likelihood function with covariance targeting is given below.

```
llik_CT_sDVECH <- function(par,x){

  a <- exp(par[1])/(1+exp(par[1]))
  b <- exp(par[2])/(1+exp(par[2]))

  d <- dim(x)
  n <- d[1]

  VECHt <- matrix(0,nrow=n,ncol=3)
  llik <- 0

  C <- cov(x)
  VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])
```

```
  for(t in 2:n){

    VECHt[t,1] <- C[1,1]*(1-a-b)+b*VECHt[t-1,1]+a*x[t-1,1]^2
    VECHt[t,3] <- C[2,2]*(1-a-b)+b*VECHt[t-1,3]+a*x[t-1,2]^2
    VECHt[t,2] <- C[1,2]*(1-a-b)+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]

    SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))

    llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
  }

  return(llik)
}
```

7.5 Estimation of the CCC model equation by equation

The CCC model can be also estimated through equation by equation approach. This approach involves only the estimation of univariate GARCH models. Then the constant correlation matrix $\boldsymbol{R}$ is estimated using the standardized residuals obtained from the univariate GARCH. This method is appealing because there is no need to optimize the log-likelihood function over the full parameter vector of the model. When the dimension of the parameter vector is large, numerical optimization methods become time-consuming and we are also more likely to encounter numerical problems.

The steps to estimate a CCC model are the following:

1. Estimate a univariate GARCH model for each series $\{y_{it}\}_{t=1}^{T}$, $i = 1, \ldots, n$.
2. Obtain the standardized errors from each of these series $\hat{\varepsilon}_{it} = (y_{it} - \hat{\mu}_i)/\hat{\sigma}_{it}$, $i = 1, \ldots, n$.
3. Estimate the correlation matrix from the residuals $\hat{\boldsymbol{R}} = T^{-1}\sum_{t=1}^{T} \hat{\boldsymbol{\varepsilon}}_t \hat{\boldsymbol{\varepsilon}}_t^{\top}$, where $\hat{\boldsymbol{\varepsilon}}_t = (\hat{\varepsilon}_{1t}, \ldots, \hat{\varepsilon}_{nt})^{\top}$.

7.6 Estimating a CCC model equation by equation with R

In the following, we show how to estimate a bivariate CCC model using R. The code can be found on the R file `estimation_CCC.R`. The bivariate time series is contained in the matrix labeled `x`. As discussed in the previous section, the first step is to estimate univariate GARCH(1,1) models for each of the series. The following lines of code do that. In case you do not remember how to estimate a univariate GARCH model yous can go back to Chapter 4. The parameter estimates for the first series are labeled `omega_hat1`, `alpha_hat1` and `beta_hat1` whereas, the parameter estimates for the second series are labeled `omega_hat2`, `alpha_hat2` and `beta_hat2`.

```
est1 <- optim(par=par_ini,fn=function(par)-llik_fun_GARCH(par,x[,1]), method = "BFGS")

est2 <- optim(par=par_ini,fn=function(par)-llik_fun_GARCH(par,x[,2]), method = "BFGS")
```

The second step is to obtain the standardized residuals. This is done through the following R code. In case you do not remember how to obtain standardized residuals from univariate GARCH models, you can go back to Chapter 5.

```
n <- length(x[,1])
s1 <- rep(0,n)
s2 <- rep(0,n)

s1[1] <- var(x[,1])
s2[1] <- var(x[,2])

for(t in 2:n){
  s1[t] <- omega_hat1 + alpha_hat1*x[t-1,1]^2 + beta_hat1*s1[t-1]
```

```
  s2[t] <- omega_hat2 + alpha_hat2*x[t-1,2]^2 + beta_hat2*s2[t-1]
}

e1 <- x[,1]/sqrt(s1)
e2 <- x[,2]/sqrt(s2)
```

Finally, the last step is to to calculate the correlation between the residuals of the first series and and the residuals of the second series. This can be done using the R function `cor()`.

```
r <- cor(e1,e2)
```

It is easy also to extend this equation by equation method to a larger dimension n instead of the bivariate case with $n = 2$. What we need is just to estimate n univariate GARCH models, get the residuals and compute their correlation matrix.

Chapter 8

Financial Analysis of Multivariate GARCH models

8.1 VaR portfolio prediction

Assume we have a portfolio that contains n assets, where $y_{i,t}$ denotes the return of asset i at time t. The vector $\boldsymbol{y}_t = (y_{1,t}, \ldots, y_{n,t})^\top$ represents the vector of returns at time t. The proportion of our portfolio invested asset i at time t is given by $k_{it} \in [0, 1]$. The quantity k_{it} is called the weight of asset i in the portfolio. Indeed, the portfolio weights have to sum to 1, i.e. $\sum_{i=1}^n k_{it} = 1$. The weights can be stacked into a vector $\boldsymbol{k}_t = (k_{1t}, \ldots, k_{nt})^\top$. We therefore obtain that the return of our portfolio $y_{p,t}$ at time t is given by

$$y_{p,t} = \sum_{i=1}^{n} k_{i,t} y_{i,t} = \boldsymbol{k}_t^\top \boldsymbol{y}_t.$$

We consider a multivariate GARCH model for $\boldsymbol{y}_t$ with conditional mean equal to $\boldsymbol{\mu}_t = (\mu_{1t}, \ldots, \mu_{nt})^\top$ and covariance matrix $\boldsymbol{\Sigma}_t$. Note that so far we have considered that the conditional mean is zero, namely $\boldsymbol{\mu}_t = \boldsymbol{0}_n$. This because stock returns shows no autocorrelation (or very little) and a sample mean close to zero. In general, as an alternative to the zero mean assumption, we can either choose a static conditional mean $\boldsymbol{\mu}_t = \boldsymbol{\mu}$ or choose a dynamic specification for $\boldsymbol{\mu}_t$ (AR model may be an option).

Therefore, as discussed in the previous sections, we have that $\boldsymbol{y}_t | Y^{t-1} \sim N_n(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. We can now find the conditional distribution of our portfolio return at time t, which is

$$y_{p,t} | Y^{t-1} \sim N(\mu_{p,t}, \sigma^2_{p,t}),$$

where the portfolio conditional mean $\mu_{p,t}$ is equal to

$$\mu_{p,t} = \mathbb{E}\left(y_{p,t} | Y^{t-1}\right) = \mathbb{E}\left(\boldsymbol{k}_t^\top \boldsymbol{y}_t | Y^{t-1}\right) = \boldsymbol{k}_t^\top \mathbb{E}\left(\boldsymbol{y}_t | Y^{t-1}\right) = \boldsymbol{k}_t^\top \boldsymbol{\mu}_t,$$

and the conditional variance $\sigma^2_{p,t}$ is given by

$$\sigma^2_{p,t} = \mathbb{V}ar\left(y_{p,t} | Y^{t-1}\right) = \mathbb{V}ar\left(\boldsymbol{k}_t^\top \boldsymbol{y}_t | Y^{t-1}\right) = \boldsymbol{k}_t^\top \mathbb{V}ar\left(\boldsymbol{y}_t | Y^{t-1}\right) \boldsymbol{k}_t = \boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t.$$

In case we only have 2 financial assets, i.e. $n = 2$, the conditional mean $\mu_{p,t}$ is given by

$$\mu_{p,t} = \boldsymbol{k}_t^\top \boldsymbol{\mu}_t = k_{1,t}\mu_{1,t} + k_{2,t}\mu_{2,t},$$

and the conditional variance $\sigma^2_{p,t}$ is given by

$$\sigma^2_{p,t} = \boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t = \begin{bmatrix} k_{1,t} & k_{2,t} \end{bmatrix} \begin{bmatrix} \sigma^2_{1,t} & \sigma_{12,t} \\ \sigma_{12,t} & \sigma^2_{2,t} \end{bmatrix} \begin{bmatrix} k_{1,t} \\ k_{2,t} \end{bmatrix} = k^2_{1,t}\sigma^2_{1,t} + k^2_{2,t}\sigma^2_{2,t} + 2k_{1,t}k_{2,t}\sigma_{12,t}.$$

Finally, we obtain that the conditional α-VaR of the portfolio at time t is given by

$$\alpha\text{-VaR}_t = \mu_{p,t} + z_\alpha \sigma_{p,t},$$

where z_α is the quantile of level α of a standard normal distribution.

8.2 Dynamic portfolio optimization

Another useful application of multivariate GARCH models is dynamic portfolio optimization. We have a portfolio of assets and at each time period we need to choose which assets to sell and buy. Therefore, at time t we have to decide the portfolio weights for time $t+1$. The idea is to choose the weights in such a way to maximize our utility function, i.e. high returns but low volatility. In the following we consider that our objective is to choose portfolio weights, $\boldsymbol{k}_t$, to maximize the so called Shape ratio, which is given by

$$S_{p,t} = \frac{\mu_{p,t}}{\sigma_{p,t}}.$$

The intuition behind maximizing the Shape ratio is that we want high portfolio returns $\mu_{p,t}$ and at the same time low risk (volatility) $\sigma_{p,t}$. The optimization problem can be written as

$$\max_{\boldsymbol{k}_t} \frac{\boldsymbol{k}_t^\top \boldsymbol{\mu}_t}{\sqrt{\boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t}}, \quad \text{s.t.} \sum_{i=1}^{n} k_{i,t} = 1,\ k_{it} \geq 0. \tag{8.1}$$

This problem in general can be solved using numerical routines in R.

In the simplest case where we only have 2 risky assets, i.e. $n = 2$, the optimization problem in (8.1) can be written as

$$\max_{k_{1t}} \frac{k_{1t}\mu_{1t} + (1-k_{1t})\mu_{2t}}{\sqrt{k_{1t}^2\sigma_{1t}^2 + (1-k_{1t})^2\sigma_{2t}^2 + 2k_{1t}(1-k_{1t})\sigma_{12t}}}, \quad \text{s.t. } 0 \leq k_{1t} \leq 1. \tag{8.2}$$

For this special case there is a closed form solution for the optimal weights when the constraint $k_{1t}, k_{2t} \geq 0$ is not imposed. In some situations this is reasonable as we can have a so called *short position* on an asset, namely a negative portfolio weight. The optimal weights are then given by

$$\begin{aligned} k_{1t} &= \frac{\mu_{1t}\sigma_{2t}^2 - \mu_{2t}\sigma_{12t}}{\mu_{1t}\sigma_{2t}^2 + \mu_{2t}\sigma_{1t}^2 - (\mu_{1t}+\mu_{2t})\sigma_{12t}}, \\ k_{2t} &= 1 - k_{1t}. \end{aligned} \tag{8.3}$$

8.3 Dynamic portfolio optimization with R

The R file `portfolio_CCC.R` provides you the code to obtain optimal portfolio weights in terms of maximum Sharpe Ratio. In this file a CCC model is used to obtain the estimate of the conditional covariance matrix $\boldsymbol{\Sigma}_t$. Furthermore, the conditional mean $\boldsymbol{\mu}_t$ is assumed to be constant, i.e. $\boldsymbol{\mu}_t = \boldsymbol{\mu}$, and it is estimated using the sample mean of the log-returns. In general, a different model can be also used and the conditional mean can be also time varying. In the following, we will only discuss how to obtain the optimal weights for a given conditional expectation $\boldsymbol{\mu}_t$ and conditional covariance matrix $\boldsymbol{\Sigma}_t$.

The constant conditional means for the first series and the second series are labeled `mu1` and `mu2` respectively. Instead, the vectors `s1`, `s2` and `s12` contain σ_{1t}^2, σ_{2t}^2 and σ_{12t} respectively, for each time $t = 1, \dots, T$.

First, we create a matrix labeled `kt` that will contain the optimal portfolio weights. Then we use a *for loop* to obtain the portfolio weights at time each time t from $t = 1, \dots, T$. This is achieved through the function `max_SR_portfolio()` which is contained in the R file `max_SR_portfolio.R`. The function requires 2 inputs. The first argument requires the conditional mean vector $\boldsymbol{\mu}_t$ and the second argument requires conditional covariance matrix $\boldsymbol{\Sigma}_t$. Note that `max_SR_portfolio()` gives the weights that maximize the Sharpe Ratio under the constraint that each weight is positive.

```
kt = matrix(0,nrow=n,ncol=2)

mu1 <- mean(x[,1])
mu2 <- mean(x[,2])
mut <- cbind(mu1,mu2)

for(t in 1:n){
  SIGMAt <- cbind(c(s1[t],s12[t]),c(s12[t],s2[t]))
```

```
  kt[t,] <- max_SR_portfolio(mut,SIGMAt)
}
```

Figure 8.1 shows the optimal portfolio weights that maximize the Sharpe Ratio. The series considered are monthly log-returns of Microsoft and IBM.

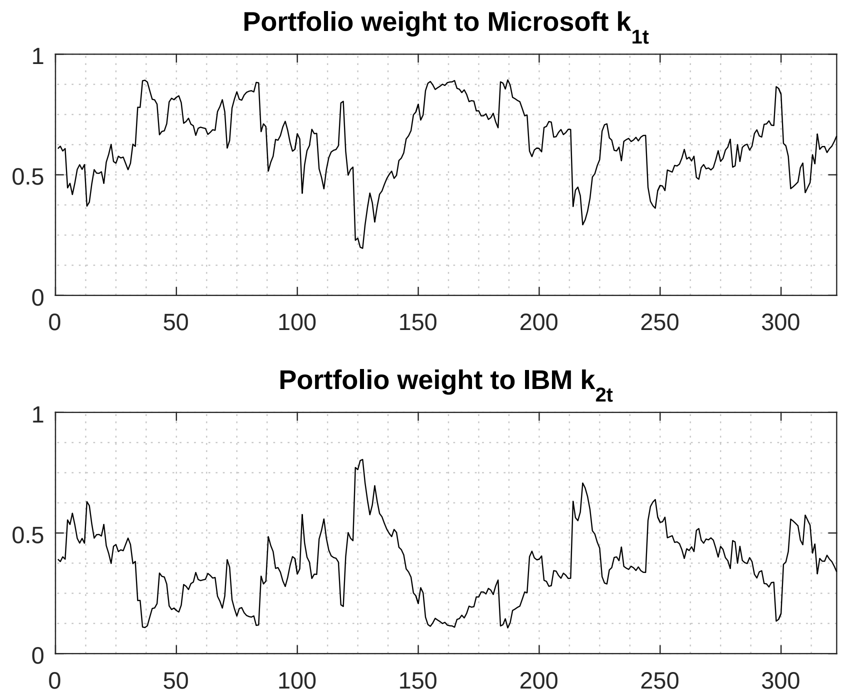

Figure 8.1: Optimal portfolio weights obtained using Microsoft and IBM monthly log-returns

8.4 Out-of-sample evaluation of different portfolio strategies

There are different methods we can use to decide on which assets to invest. For instance, above we have given an example of dynamic portfolio optimization based on the CCC model. However, we could use a different model and this would lead to different portfolio weights. The question that arises now is: **how can we decide which portfolio strategy is best?** The answer is simple: we can consider a sub-sample of the observed data and see how different strategies perform in this sub-sample. In particular, we can take our observed dataset $\{\boldsymbol{y}_t\}$, for $t = 1, \ldots, T$, and split it into two sub-samples: an *in-sample* dataset, for $t = 1, \ldots, T_1$, and an *out-of-sample* dataset, for $t = T_1 + 1, \ldots, T$. We can then use the in-sample dataset to estimate our models and the out-of-sample dataset to evaluate the performance of our portfolio strategies. To evaluate the performance of our portfolios we can calculate their means and variances and obtain their sharpe ratios. We can then choose the portfolio strategy that delivers the highest sharpe ratio.

The out-of-sample portfolio evaluation is as follows:

1. Estimate a multivariate GARCH model using the in-sample dataset, $t = 1, \ldots, T_1$. The estimation can be based on ML, covariance targeting or equation-by-equation, depending on the model we are using.
2. For the out-of-sample dataset, $t = T_1 + 1, \ldots, T$, obtain an estimate of the conditional mean $\boldsymbol{\mu}_t$ and the conditional covariance matrix $\boldsymbol{\Sigma}_t$ of the returns by using the multivariate GARCH model estimated in the previous point.
3. For the out-of-sample dataset, $t = T_1 + 1, \ldots, T$, use the conditional covariance matrix and the conditional mean to obtain the log-returns of the optimal portfolio (see Sections 8.2 and 8.3).

4. Use the log-returns of the optimal portfolio $\{y_{p,t}\}$, for $t = T_1 + 1, \ldots, T$, to obtain an empirical estimate of the sharpe ratio of the portfolio as

$$\hat{S}_p = \frac{\bar{y}_p}{\hat{\sigma}_p},$$

where

$$\bar{y}_p = \frac{1}{T - T_1} \sum_{t=T_1+1}^{T} y_{p,t}, \qquad \hat{\sigma}_p^2 = \frac{1}{T - T_1} \sum_{t=T_1+1}^{T} (y_{p,t} - \bar{y}_p)^2.$$

Appendix A

Stock Return Properties: Empirical Evidence

Table A.1: p-values of ADF test for log returns of S&P100 stocks.

Stock	daily prices	daily returns	weekly prices	weekly returns	monthly prices	monthly returns
AAPL	0.239	0.001	0.188	0.001	0.230	0.001
ABBV	0.615	0.001	0.643	0.001	0.651	0.001
ABT	0.372	0.001	0.606	0.001	0.619	0.001
ACN	0.308	0.001	0.369	0.001	0.379	0.001
AGN	0.184	0.001	0.206	0.001	0.266	0.001
AIG	0.607	0.001	0.573	0.001	0.677	0.001
ALL	0.411	0.001	0.509	0.001	0.484	0.001
AMGN	0.198	0.001	0.296	0.001	0.387	0.001
AMZN	0.054	0.001	0.051	0.001	0.069	0.001
AXP	0.486	0.001	0.431	0.001	0.500	0.001
BA	0.352	0.001	0.490	0.001	0.506	0.001
BAC	0.561	0.001	0.687	0.001	0.768	0.001
BIIB	0.191	0.001	0.177	0.001	0.266	0.001
BK	0.462	0.001	0.574	0.001	0.552	0.001
BLK	0.296	0.001	0.362	0.001	0.398	0.001
BMY	0.197	0.001	0.458	0.001	0.558	0.001
BRK-B	0.417	0.001	0.516	0.001	0.518	0.001
C	0.589	0.001	0.479	0.001	0.563	0.001
CAT	0.421	0.001	0.510	0.001	0.560	0.001
CELG	0.159	0.001	0.144	0.001	0.223	0.001
CL	0.355	0.001	0.554	0.001	0.607	0.001
CMCSA	0.198	0.001	0.336	0.001	0.357	0.001
COF	0.431	0.001	0.247	0.001	0.293	0.001
COP	0.525	0.001	0.583	0.001	0.584	0.001
COST	0.198	0.001	0.287	0.001	0.338	0.001
CSCO	0.344	0.001	0.262	0.001	0.251	0.001
CVS	0.294	0.001	0.386	0.001	0.369	0.001
CVX	0.454	0.001	0.564	0.001	0.588	0.001
DD	0.387	0.001	0.620	0.001	0.660	0.001
DHR	0.202	0.001	0.457	0.001	0.431	0.001
DIS	0.300	0.001	0.346	0.001	0.204	0.001
DOW	0.345	0.001	0.509	0.001	0.557	0.001
DUK	0.366	0.001	0.392	0.001	0.352	0.001
EMR	0.471	0.001	0.659	0.001	0.684	0.001
EXC	0.520	0.001	0.581	0.001	0.558	0.001
F	0.515	0.001	0.598	0.001	0.711	0.001
FB	0.494	0.001	0.512	0.001	0.526	0.001
FDX	0.382	0.001	0.419	0.001	0.457	0.001
FOX	0.459	0.001	0.482	0.001	0.543	0.001
FOXA	0.449	0.001	0.481	0.001	0.549	0.001
GD	0.263	0.001	0.429	0.001	0.462	0.001
GE	0.484	0.001	0.412	0.001	0.599	0.001
GILD	0.274	0.001	0.256	0.001	0.295	0.001
GM	0.691	0.001	0.703	0.001	0.731	0.001
GOOG	0.348	0.001	0.435	0.001	0.473	0.001
GOOGL	0.338	0.001	0.448	0.001	0.474	0.001
GS	0.510	0.001	0.529	0.001	0.542	0.001
HAL	0.357	0.001	0.431	0.001	0.406	0.001
HD	0.147	0.001	0.236	0.001	0.330	0.001
HON	0.253	0.001	0.390	0.001	0.400	0.001

Table A.2: (continued) p-values of ADF test for log returns of S&P100 stocks.

Stock	daily prices	daily returns	weekly prices	weekly returns	monthly prices	monthly returns
IBM	0.488	0.001	0.495	0.001	0.604	0.001
INTC	0.313	0.001	0.356	0.001	0.115	0.001
JNJ	0.311	0.001	0.470	0.001	0.494	0.001
JPM	0.366	0.001	0.521	0.001	0.474	0.001
KHC	0.692	0.001	0.701	0.999	0.676	0.126
KMI	0.714	0.001	0.739	0.001	0.734	0.001
KO	0.473	0.001	0.613	0.001	0.644	0.001
LLY	0.358	0.001	0.498	0.001	0.562	0.001
LMT	0.141	0.001	0.215	0.001	0.241	0.001
LOW	0.159	0.001	0.277	0.001	0.312	0.001
MA	0.433	0.001	0.189	0.001	0.246	0.001
MCD	0.340	0.001	0.406	0.001	0.435	0.001
MDLZ	0.439	0.001	0.602	0.001	0.612	0.001
MDT	0.330	0.001	0.422	0.001	0.466	0.001
MET	0.522	0.001	0.548	0.001	0.617	0.001
MMM	0.274	0.001	0.432	0.001	0.473	0.001
MO	0.126	0.001	0.598	0.001	0.477	0.001
MON	0.435	0.001	0.492	0.001	0.494	0.001
MRK	0.437	0.001	0.562	0.001	0.514	0.001
MS	0.418	0.001	0.428	0.001	0.532	0.001
MSFT	0.249	0.001	0.425	0.001	0.692	0.001
NKE	0.191	0.001	0.542	0.001	0.639	0.001
ORCL	0.332	0.001	0.119	0.001	0.250	0.001
OXY	0.428	0.001	0.475	0.001	0.500	0.001
PCLN	0.188	0.001	0.201	0.001	0.271	0.001
PEP	0.378	0.001	0.477	0.001	0.494	0.001
PFE	0.413	0.001	0.520	0.001	0.470	0.001
PG	0.458	0.001	0.581	0.001	0.581	0.001
PM	0.551	0.001	0.609	0.001	0.608	0.001
PYPL	0.715	0.001	0.711	0.001	0.697	0.017
QCOM	0.346	0.001	0.011	0.001	0.153	0.001
RTN	0.175	0.001	0.306	0.001	0.295	0.001
SBUX	0.160	0.001	0.361	0.001	0.413	0.001
SLB	0.418	0.001	0.518	0.001	0.518	0.001
SO	0.375	0.001	0.536	0.001	0.560	0.001
SPG	0.175	0.001	0.281	0.001	0.321	0.001
T	0.375	0.001	0.631	0.001	0.609	0.001
TGT	0.420	0.001	0.438	0.001	0.497	0.001
TWX	0.195	0.001	0.098	0.001	0.447	0.001
TXN	0.158	0.001	0.141	0.001	0.103	0.001
UNH	0.136	0.001	0.191	0.001	0.221	0.001
UNP	0.295	0.001	0.477	0.001	0.542	0.001
UPS	0.507	0.001	0.586	0.001	0.603	0.001
USB	0.398	0.001	0.473	0.001	0.579	0.001
UTX	0.412	0.001	0.509	0.001	0.582	0.001
V	0.459	0.001	0.494	0.001	0.514	0.001
VZ	0.379	0.001	0.584	0.001	0.577	0.001
WBA	0.329	0.001	0.366	0.001	0.386	0.001
WFC	0.396	0.001	0.530	0.001	0.582	0.001
WMT	0.486	0.001	0.499	0.001	0.526	0.001
XOM	0.488	0.001	0.591	0.001	0.599	0.001

Table A.3: Estimated ACF(1), MA(1) and AR(1) coefficients for log returns of S&P100 stocks.

	Daily			Weekly			Monthly		
	ACF(1)	MA(1)	AR(1)	ACF(1)	MA(1)	AR(1)	ACF(1)	MA(1)	AR(1)
AAPL	-0.026	-0.026	-0.026	0.037	0.040	0.037	0.036	0.035	0.036
ABBV	-0.031	-0.031	-0.031	-0.122	-0.171	-0.122	-0.061	-0.094	-0.061
ABT	-0.010	-0.011	-0.011	-0.030	-0.034	-0.030	-0.059	-0.056	-0.060
ACN	-0.026	-0.031	-0.026	-0.068	-0.072	-0.068	0.043	0.045	0.043
AGN	-0.003	-0.003	-0.003	-0.022	-0.022	-0.023	0.037	0.036	0.039
AIG	0.137	0.125	0.137	-0.023	-0.023	-0.023	0.185	0.208	0.186
ALL	-0.057	-0.062	-0.057	-0.076	-0.065	-0.076	-0.001	-0.001	-0.001
AMGN	-0.043	-0.050	-0.043	-0.045	-0.048	-0.045	-0.042	-0.059	-0.042
AMZN	0.008	0.009	0.008	-0.005	-0.005	-0.005	-0.015	-0.017	-0.015
AXP	-0.057	-0.062	-0.057	-0.009	-0.009	-0.009	0.091	0.138	0.091
BA	0.004	0.005	0.004	-0.066	-0.066	-0.066	-0.009	-0.008	-0.009
BAC	-0.008	-0.008	-0.008	-0.036	-0.036	-0.036	0.111	0.112	0.111
BIIB	0.014	0.016	0.014	-0.054	-0.048	-0.054	0.001	0.001	0.001
BK	-0.118	-0.135	-0.118	-0.072	-0.070	-0.101	-0.134	-0.148	-0.135
BLK	-0.035	-0.038	-0.035	-0.129	-0.118	-0.129	-0.109	-0.092	-0.112
BMY	-0.018	-0.020	-0.018	-0.025	-0.025	-0.025	-0.074	-0.090	-0.074
BRK-B	0.029	0.029	0.029	-0.028	-0.027	-0.028	-0.005	-0.005	-0.005
C	0.056	0.056	0.056	-0.040	-0.040	-0.040	0.082	0.096	0.083
CAT	-0.005	-0.005	-0.005	-0.046	-0.046	-0.046	-0.033	-0.038	-0.033
CELG	-0.001	-0.001	-0.001	0.005	0.005	0.005	-0.052	-0.058	-0.053
CL	-0.009	-0.011	-0.009	-0.060	-0.061	-0.060	-0.154	-0.163	-0.154
CMCSA	-0.063	-0.067	-0.063	-0.066	-0.073	-0.066	-0.073	-0.071	-0.073
COF	-0.031	-0.031	-0.031	-0.069	-0.064	-0.069	0.113	0.116	0.113
COP	-0.038	-0.042	-0.038	-0.063	-0.063	-0.063	-0.029	-0.032	-0.029
COST	-0.015	-0.016	-0.015	-0.125	-0.132	-0.125	0.018	0.024	0.018
CSCO	-0.044	-0.047	-0.044	-0.049	-0.050	-0.050	0.009	0.010	0.009
CVS	-0.057	-0.060	-0.057	-0.054	-0.058	-0.054	-0.028	-0.025	-0.028
CVX	-0.069	-0.069	-0.069	-0.038	-0.042	-0.039	-0.107	-0.109	-0.108
DD	-0.011	-0.011	-0.011	-0.076	-0.070	-0.076	-0.021	-0.022	-0.021
DHR	-0.013	-0.013	-0.013	-0.007	-0.006	-0.007	-0.049	-0.050	-0.049
DIS	-0.028	-0.030	-0.028	-0.024	-0.024	-0.024	0.159	0.140	0.160
DOW	-0.032	-0.032	-0.032	0.038	0.039	0.039	0.071	0.072	0.071
DUK	-0.038	-0.038	-0.038	-0.020	-0.021	-0.020	-0.021	-0.022	-0.021
EMR	-0.063	-0.073	-0.063	-0.064	-0.064	-0.064	-0.073	-0.081	-0.073
EXC	-0.035	-0.037	-0.035	-0.033	-0.034	-0.033	0.114	0.132	0.114
F	0.012	0.012	0.012	-0.028	-0.028	-0.028	-0.087	-0.084	-0.087
FB	0.022	0.021	0.022	0.073	0.075	0.081	0.006	0.007	0.006
FDX	-0.001	-0.001	-0.001	-0.014	-0.014	-0.014	-0.044	-0.042	-0.045
FOX	0.025	0.026	0.025	-0.009	-0.008	-0.009	-0.036	-0.037	-0.036
FOXA	0.006	0.006	0.006	-0.027	-0.027	-0.027	-0.059	-0.060	-0.060
GD	-0.035	-0.035	-0.035	0.034	0.035	0.034	0.010	0.011	0.010
GE	-0.011	-0.011	-0.011	-0.039	-0.039	-0.039	-0.034	-0.039	-0.034
GILD	-0.019	-0.021	-0.019	-0.061	-0.068	-0.062	0.031	0.039	0.032
GM	0.026	0.025	0.026	-0.040	-0.041	-0.041	-0.034	-0.043	-0.035
GOOG	0.010	0.010	0.010	-0.020	-0.020	-0.020	0.098	0.117	0.104
GOOGL	0.010	0.010	0.010	0.001	0.001	0.001	0.098	0.112	0.104
GS	-0.045	-0.049	-0.045	-0.152	-0.133	-0.152	0.083	0.098	0.083
HAL	0.019	0.021	0.019	-0.102	-0.095	-0.102	0.082	0.092	0.082
HD	0.009	0.010	0.009	-0.066	-0.068	-0.066	0.064	0.088	0.064
HON	-0.001	-0.001	-0.001	0.006	0.006	0.006	0.065	0.099	0.066

Table A.4: (continued) Estimated ACF(1), MA(1) and AR(1) coefficients for log returns of S&P100 stocks.

Stock	d.ACF(1)	d.MA(1)	d.AR(1)	w.ACF(1)	w.MA(1)	w.AR(1)	m.ACF(1)	m.MA(1)	m.AR(1)
IBM	-0.027	-0.027	-0.027	-0.023	-0.023	-0.023	-0.206	-0.220	-0.208
INTC	-0.042	-0.044	-0.042	-0.038	-0.040	-0.038	-0.048	-0.038	-0.049
JNJ	0.007	0.008	0.007	-0.098	-0.099	-0.098	-0.111	-0.124	-0.111
JPM	-0.063	-0.067	-0.063	-0.088	-0.088	-0.088	-0.061	-0.057	-0.061
KHC	0.055	0.068	0.055	-0.063	-0.392	-0.065	-0.178	-0.366	-0.352
KMI	0.064	0.065	0.064	-0.053	-0.054	-0.054	0.164	0.202	0.165
KO	0.001	0.001	0.001	-0.021	-0.021	-0.021	-0.056	-0.063	-0.056
LLY	-0.022	-0.025	-0.022	-0.040	-0.044	-0.040	-0.140	-0.165	-0.142
LMT	-0.073	-0.073	-0.073	-0.012	-0.012	-0.012	0.115	0.143	0.115
LOW	0.019	0.021	0.019	-0.028	-0.029	-0.028	0.007	0.007	0.007
MA	-0.058	-0.061	-0.058	0.016	0.017	0.017	-0.011	-0.011	-0.011
MCD	-0.018	-0.018	-0.018	-0.042	-0.046	-0.042	0.036	0.033	0.036
MDLZ	-0.055	-0.059	-0.055	-0.018	-0.018	-0.018	-0.104	-0.112	-0.104
MDT	-0.002	-0.002	-0.002	-0.047	-0.047	-0.047	0.002	0.003	0.002
MET	-0.073	-0.073	-0.073	-0.038	-0.042	-0.038	0.027	0.065	0.029
MMM	-0.039	-0.039	-0.039	-0.063	-0.063	-0.063	-0.113	-0.131	-0.113
MO	-0.042	-0.046	-0.042	-0.045	-0.045	-0.046	0.062	0.071	0.062
MON	0.003	0.003	0.003	-0.041	-0.049	-0.041	0.024	0.025	0.024
MRK	0.002	0.002	0.002	-0.042	-0.043	-0.042	-0.084	-0.069	-0.085
MS	0.010	0.011	0.010	-0.173	-0.178	-0.173	0.006	0.009	0.006
MSFT	-0.033	-0.033	-0.033	-0.004	-0.004	-0.004	-0.143	-0.164	-0.144
NKE	-0.010	-0.010	-0.010	-0.024	-0.024	-0.024	-0.069	-0.093	-0.070
ORCL	-0.053	-0.059	-0.053	0.053	0.053	0.053	-0.056	-0.074	-0.057
OXY	-0.051	-0.057	-0.051	-0.037	-0.037	-0.037	-0.022	-0.026	-0.022
PCLN	0.041	0.042	0.041	0.021	0.020	0.021	0.206	0.150	0.220
PEP	-0.070	-0.078	-0.070	-0.058	-0.062	-0.060	-0.030	-0.039	-0.030
PFE	-0.006	-0.007	-0.006	-0.049	-0.051	-0.049	-0.066	-0.064	-0.067
PG	-0.040	-0.040	-0.040	-0.083	-0.083	-0.083	0.081	0.103	0.082
PM	-0.036	-0.044	-0.036	-0.094	-0.101	-0.094	-0.059	-0.075	-0.059
PYPL	0.109	0.110	0.109	-0.141	-0.172	-0.177	-0.161	-0.259	-0.248
QCOM	-0.031	-0.034	-0.031	-0.016	-0.016	-0.016	-0.113	-0.127	-0.113
RTN	0.028	0.029	0.028	-0.024	-0.024	-0.024	0.044	0.044	0.044
SBUX	-0.073	-0.078	-0.073	-0.038	-0.043	-0.038	-0.044	-0.050	-0.044
SLB	-0.031	-0.034	-0.031	-0.101	-0.105	-0.101	-0.015	-0.015	-0.015
SO	-0.055	-0.055	-0.055	-0.127	-0.137	-0.127	-0.147	-0.248	-0.147
SPG	-0.169	-0.173	-0.169	-0.062	-0.073	-0.062	0.003	0.003	0.003
T	-0.026	-0.026	-0.026	-0.096	-0.103	-0.096	-0.017	-0.019	-0.017
TGT	-0.054	-0.062	-0.054	-0.052	-0.053	-0.053	0.104	0.141	0.104
TWX	0.007	0.008	0.007	0.017	0.019	0.017	-0.214	-0.238	-0.216
TXN	0.005	0.006	0.005	-0.119	-0.114	-0.119	-0.005	-0.005	-0.005
UNH	-0.006	-0.006	-0.006	-0.013	-0.014	-0.013	-0.003	-0.003	-0.003
UNP	-0.016	-0.017	-0.016	-0.042	-0.043	-0.042	-0.033	-0.036	-0.033
UPS	-0.039	-0.045	-0.040	-0.081	-0.081	-0.082	-0.040	-0.077	-0.040
USB	-0.010	-0.010	-0.010	-0.117	-0.110	-0.117	-0.063	-0.065	-0.063
UTX	-0.039	-0.039	-0.039	-0.016	-0.016	-0.016	-0.104	-0.119	-0.104
V	-0.093	-0.106	-0.097	0.013	0.013	0.013	-0.031	-0.031	-0.033
VZ	-0.066	-0.070	-0.066	-0.006	-0.006	-0.006	-0.086	-0.093	-0.086
WBA	-0.049	-0.049	-0.049	-0.057	-0.053	-0.057	0.007	0.009	0.008
WFC	-0.097	-0.098	-0.097	-0.125	-0.120	-0.125	-0.075	-0.079	-0.075
WMT	-0.022	-0.025	-0.022	-0.059	-0.060	-0.059	-0.034	-0.040	-0.034
XOM	-0.098	-0.118	-0.098	-0.069	-0.070	-0.069	-0.042	-0.042	-0.042

Table A.5: Moments of Log Returns for S&P100 Stocks

Stock	Mean	Var	Skew	Kurt	JB	Mean	Var	Skew	Kurt	JB	Mean	Var	Skew	Kurt	JB
AAPL	0.000	0.00	14.5	356	0.001	0.009	0.03	22.8	519	0.001	0.040	0.11	10.8	117	0.001
ABBV	0.000	0.00	8.6	110	0.001	0.002	0.00	5.1	38	0.001	0.004	0.00	1.3	4	0.006
ABT	0.000	0.00	10.9	165	0.001	0.002	0.00	22.4	509	0.001	0.006	0.00	10.6	115	0.001
ACN	0.000	0.00	14.1	294	0.001	0.001	0.00	6.7	72	0.001	0.004	0.00	2.0	7	0.001
AGN	0.000	0.00	13.3	297	0.001	0.001	0.00	10.9	160	0.001	0.004	0.00	3.3	19	0.001
AIG	0.002	0.00	28.8	1051	0.001	0.024	0.08	20.6	450	0.001	0.123	0.42	7.3	60	0.001
ALL	0.000	0.00	14.3	258	0.001	0.002	0.00	16.8	332	0.001	0.008	0.00	7.5	64	0.001
AMGN	0.000	0.00	9.6	131	0.001	0.001	0.00	6.6	64	0.001	0.005	0.00	5.1	38	0.001
AMZN	0.001	0.00	13.8	264	0.001	0.003	0.00	8.9	123	0.001	0.011	0.00	5.5	42	0.001
AXP	0.001	0.00	9.2	113	0.001	0.003	0.00	6.4	51	0.001	0.010	0.00	8.5	83	0.001
BA	0.000	0.00	9.5	160	0.001	0.002	0.00	7.5	90	0.001	0.006	0.00	4.6	30	0.001
BAC	0.001	0.00	10.3	134	0.001	0.006	0.00	9.5	102	0.001	0.023	0.00	5.9	44	0.001
BIIB	0.001	0.00	25.5	770	0.001	0.002	0.00	10.2	127	0.001	0.009	0.00	3.3	15	0.001
BK	0.001	0.00	15.9	377	0.001	0.002	0.00	6.3	56	0.001	0.005	0.00	3.8	23	0.001
BLK	0.001	0.00	9.0	115	0.001	0.002	0.00	7.3	78	0.001	0.008	0.00	5.5	40	0.001
BMY	0.000	0.00	6.4	60	0.001	0.001	0.00	5.5	44	0.001	0.004	0.00	2.1	7	0.001
BRK-B	0.000	0.00	20.5	627	0.001	0.029	0.41	22.8	520	0.001	0.120	1.67	10.8	118	0.001
C	0.002	0.00	17.4	408	0.001	0.018	0.05	20.9	459	0.001	0.067	0.20	10.3	109	0.001
CAT	0.000	0.00	8.8	119	0.001	0.002	0.00	5.2	39	0.001	0.010	0.00	5.3	35	0.001
CELG	0.000	0.00	10.9	167	0.001	0.003	0.00	21.5	481	0.001	0.011	0.00	8.6	85	0.001
CL	0.000	0.00	11.0	176	0.001	0.001	0.00	22.3	505	0.001	0.006	0.00	10.7	117	0.001
CMCSA	0.000	0.00	17.9	477	0.001	0.002	0.00	16.5	324	0.001	0.007	0.00	9.8	103	0.001
COF	0.001	0.00	9.5	123	0.001	0.005	0.00	12.4	196	0.001	0.014	0.00	8.1	77	0.001
COP	0.000	0.00	9.6	137	0.001	0.002	0.00	11.4	156	0.001	0.007	0.00	4.9	30	0.001
COST	0.000	0.00	11.9	209	0.001	0.001	0.00	6.5	58	0.001	0.003	0.00	2.8	13	0.001
CSCO	0.000	0.00	11.8	188	0.001	0.002	0.00	6.2	56	0.001	0.006	0.00	2.7	12	0.001
CVS	0.000	0.00	25.3	871	0.001	0.001	0.00	6.6	64	0.001	0.004	0.00	2.8	13	0.001
CVX	0.000	0.00	16.4	385	0.001	0.001	0.00	17.0	343	0.001	0.004	0.00	2.2	8	0.001
DD	0.000	0.00	7.7	82	0.001	0.002	0.00	4.4	27	0.001	0.007	0.00	3.0	12	0.001
DHR	0.000	0.00	8.8	118	0.001	0.002	0.00	22.0	496	0.001	0.008	0.00	10.6	115	0.001
DIS	0.000	0.00	10.3	156	0.001	0.001	0.00	11.6	189	0.001	0.004	0.00	3.1	14	0.001
DOW	0.001	0.00	11.4	212	0.001	0.003	0.00	4.6	28	0.001	0.014	0.00	6.7	54	0.001
DUK	0.000	0.00	16.5	379	0.001	0.003	0.00	20.6	443	0.001	0.014	0.01	10.1	106	0.001
EMR	0.000	0.00	12.5	231	0.001	0.002	0.00	22.4	509	0.001	0.009	0.00	10.3	110	0.001
EXC	0.000	0.00	14.2	292	0.001	0.001	0.00	10.2	149	0.001	0.004	0.00	3.1	13	0.001
F	0.001	0.00	12.9	208	0.001	0.005	0.00	14.5	230	0.001	0.023	0.01	7.0	53	0.001
FB	0.001	0.00	17.0	382	0.001	0.003	0.00	5.5	43	0.001	0.014	0.00	3.5	15	0.001
FDX	0.000	0.00	9.8	176	0.001	0.002	0.00	5.4	43	0.001	0.006	0.00	2.7	11	0.001
FOX	0.000	0.00	12.2	210	0.001	0.002	0.00	7.5	70	0.001	0.007	0.00	3.2	15	0.001
FOXA	0.001	0.00	10.6	157	0.001	0.002	0.00	7.1	65	0.001	0.008	0.00	4.6	29	0.001
GD	0.000	0.00	7.9	102	0.001	0.001	0.00	7.6	78	0.001	0.005	0.00	4.2	23	0.001
GE	0.000	0.00	10.2	157	0.001	0.002	0.00	9.5	128	0.001	0.007	0.00	4.8	30	0.001
GILD	0.000	0.00	11.0	163	0.001	0.003	0.00	15.8	255	0.001	0.014	0.00	7.8	65	0.001
GM	0.000	0.00	7.5	87	0.001	0.002	0.00	3.7	21	0.001	0.007	0.00	3.0	14	0.001
GOOG	0.000	0.00	13.1	243	0.001	0.003	0.00	21.9	494	0.001	0.012	0.00	10.2	109	0.001
GOOGL	0.000	0.00	13.2	247	0.001	0.003	0.00	21.9	491	0.001	0.012	0.00	10.0	106	0.001
GS	0.001	0.00	12.2	194	0.001	0.003	0.00	10.1	124	0.001	0.009	0.00	3.8	22	0.001
HAL	0.001	0.00	10.6	164	0.001	0.003	0.00	13.1	228	0.001	0.012	0.00	5.5	40	0.001
HD	0.000	0.00	9.3	141	0.001	0.002	0.00	8.3	97	0.001	0.004	0.00	2.5	10	0.001
HON	0.000	0.00	6.9	70	0.001	0.001	0.00	7.2	81	0.001	0.005	0.00	5.5	41	0.001

Table A.6: (continued) Moments of Log Returns for S&P100 Stocks

Stock	Mean	Var	Skew	Kurt	JB	Mean	Var	Skew	Kurt	JB	Mean	Var	Skew	Kurt	JB
IBM	0.000	0.00	9.0	134	0.001	0.001	0.00	6.0	49	0.001	0.003	0.00	5.4	41	0.001
INTC	0.000	0.00	8.2	102	0.001	0.001	0.00	5.1	37	0.001	0.005	0.00	3.2	16	0.001
JNJ	0.000	0.00	19.0	526	0.001	0.000	0.00	13.8	239	0.001	0.002	0.00	3.6	20	0.001
JPM	0.001	0.00	10.4	140	0.001	0.003	0.00	9.7	107	0.001	0.008	0.00	2.9	12	0.001
KHC	0.000	0.00	3.7	19	0.001	0.001	0.00	1.9	6	0.001	0.003	0.00	1.0	3	0.079
KMI	0.000	0.00	9.8	140	0.001	0.002	0.00	13.2	200	0.001	0.006	0.00	7.5	59	0.001
KO	0.000	0.00	18.3	481	0.001	0.002	0.00	22.3	505	0.001	0.007	0.00	10.7	117	0.001
LLY	0.000	0.00	13.8	286	0.001	0.001	0.00	17.1	345	0.001	0.003	0.00	6.1	44	0.001
LMT	0.000	0.00	9.3	118	0.001	0.001	0.00	10.3	127	0.001	0.004	0.00	5.8	41	0.001
LOW	0.000	0.00	9.7	155	0.001	0.002	0.00	6.7	64	0.001	0.006	0.00	2.1	8	0.001
MA	0.001	0.00	9.6	139	0.001	0.013	0.06	22.8	520	0.001	0.057	0.28	10.8	118	0.001
MCD	0.000	0.00	9.6	136	0.001	0.001	0.00	6.4	60	0.001	0.002	0.00	2.6	11	0.001
MDLZ	0.000	0.00	7.8	95	0.001	0.001	0.00	20.2	436	0.001	0.005	0.00	9.7	101	0.001
MDT	0.000	0.00	13.7	254	0.001	0.001	0.00	8.8	94	0.001	0.004	0.00	5.3	37	0.001
MET	0.001	0.00	11.1	163	0.001	0.004	0.00	8.6	84	0.001	0.011	0.00	6.4	48	0.001
MMM	0.000	0.00	7.6	81	0.001	0.001	0.00	7.7	91	0.001	0.003	0.00	3.1	13	0.001
MO	0.000	0.00	23.8	723	0.001	0.004	0.00	22.7	518	0.001	0.015	0.02	10.8	117	0.001
MON	0.000	0.00	11.4	181	0.001	0.002	0.00	6.8	66	0.001	0.007	0.00	2.5	9	0.001
MRK	0.000	0.00	12.7	239	0.001	0.001	0.00	6.8	62	0.001	0.004	0.00	3.6	20	0.001
MS	0.001	0.00	31.7	1275	0.001	0.007	0.00	15.7	275	0.001	0.014	0.00	6.9	62	0.001
MSFT	0.000	0.00	13.1	265	0.001	0.001	0.00	6.4	59	0.001	0.005	0.00	2.8	12	0.001
NKE	0.000	0.00	8.5	94	0.001	0.004	0.00	13.0	171	0.001	0.016	0.01	6.2	40	0.001
ORCL	0.000	0.00	8.4	96	0.001	0.001	0.00	4.4	28	0.001	0.005	0.00	3.1	16	0.001
OXY	0.001	0.00	11.8	190	0.001	0.003	0.00	20.2	436	0.001	0.006	0.00	3.8	21	0.001
PCLN	0.001	0.00	10.7	146	0.001	0.003	0.00	5.6	45	0.001	0.014	0.00	2.2	8	0.001
PEP	0.000	0.00	20.8	637	0.001	0.001	0.00	15.0	282	0.001	0.002	0.00	8.6	86	0.001
PFE	0.000	0.00	10.9	167	0.001	0.001	0.00	11.5	169	0.001	0.003	0.00	4.4	28	0.001
PG	0.000	0.00	11.4	200	0.001	0.001	0.00	14.5	274	0.001	0.002	0.00	2.9	13	0.001
PM	0.000	0.00	12.1	204	0.001	0.001	0.00	13.7	238	0.001	0.003	0.00	2.5	11	0.001
PYPL	0.000	0.00	4.0	25	0.001	0.002	0.00	3.2	13	0.001	0.004	0.00	1.7	4	0.011
QCOM	0.000	0.00	11.8	189	0.001	0.002	0.00	5.2	42	0.001	0.006	0.00	2.6	10	0.001
RTN	0.000	0.00	11.0	193	0.001	0.001	0.00	7.3	70	0.001	0.003	0.00	5.4	42	0.001
SBUX	0.000	0.00	9.5	150	0.001	0.003	0.00	19.9	428	0.001	0.010	0.00	8.7	85	0.001
SLB	0.001	0.00	13.0	238	0.001	0.002	0.00	6.8	71	0.001	0.009	0.00	5.6	42	0.001
SO	0.000	0.00	15.5	357	0.001	0.000	0.00	10.2	138	0.001	0.002	0.00	2.3	8	0.001
SPG	0.001	0.00	9.8	132	0.001	0.002	0.00	7.1	62	0.001	0.008	0.00	5.2	31	0.001
T	0.000	0.00	18.0	472	0.001	0.001	0.00	14.3	262	0.001	0.003	0.00	3.6	20	0.001
TGT	0.000	0.00	10.6	181	0.001	0.001	0.00	6.3	50	0.001	0.005	0.00	2.6	10	0.001
TWX	0.000	0.00	18.4	498	0.001	0.003	0.00	22.4	508	0.001	0.013	0.01	10.5	114	0.001
TXN	0.000	0.00	12.3	262	0.001	0.001	0.00	5.4	45	0.001	0.005	0.00	3.1	17	0.001
UNH	0.000	0.00	23.6	747	0.001	0.002	0.00	9.8	112	0.001	0.007	0.00	5.4	36	0.001
UNP	0.000	0.00	9.2	158	0.001	0.003	0.00	16.1	266	0.001	0.011	0.00	7.6	62	0.001
UPS	0.000	0.00	8.2	99	0.001	0.001	0.00	5.9	51	0.001	0.003	0.00	5.8	43	0.001
USB	0.001	0.00	9.9	126	0.001	0.003	0.00	12.4	177	0.001	0.006	0.00	9.7	101	0.001
UTX	0.000	0.00	12.0	226	0.001	0.001	0.00	5.4	43	0.001	0.003	0.00	2.1	8	0.001
V	0.000	0.00	9.6	123	0.001	0.006	0.01	20.8	434	0.001	0.025	0.04	9.9	98	0.001
VZ	0.000	0.00	15.2	370	0.001	0.001	0.00	9.2	109	0.001	0.003	0.00	2.1	7	0.001
WBA	0.000	0.00	14.9	297	0.001	0.001	0.00	6.1	52	0.001	0.006	0.00	2.4	9	0.001
WFC	0.001	0.00	11.9	180	0.001	0.005	0.00	13.3	215	0.001	0.009	0.00	5.5	35	0.001
WMT	0.000	0.00	12.6	216	0.001	0.001	0.00	8.6	109	0.001	0.002	0.00	4.1	24	0.001
XOM	0.000	0.00	14.8	285	0.001	0.001	0.00	15.3	300	0.001	0.002	0.00	2.0	7	0.001

Table A.7: Estimated ACF(1), MA(1) and AR(1) coefficients for squared log returns of S&P100 stocks.

Stock	d.ACF(1)	d.MA(1)	d.AR(1)	w.ACF(1)	w.MA(1)	w.AR(1)	m.ACF(1)	m.MA(1)	m.AR(1)
AAPL	0.178	0.184	0.178	-0.003	-0.003	-0.003	-0.012	-0.012	-0.012
ABBV	0.174	0.180	0.175	0.240	0.256	0.241	0.250	0.551	0.266
ABT	0.129	0.131	0.129	-0.002	-0.002	-0.002	-0.021	-0.022	-0.021
ACN	0.070	0.070	0.070	0.277	0.301	0.277	-0.119	-0.130	-0.119
AGN	0.123	0.125	0.123	0.061	0.062	0.062	0.161	0.244	0.163
AIG	0.234	0.140	0.234	0.040	0.040	0.040	0.208	0.227	0.209
ALL	0.255	0.274	0.255	0.188	0.139	0.188	0.132	0.110	0.132
AMGN	0.161	0.165	0.161	0.011	0.011	0.011	0.004	0.004	0.005
AMZN	0.119	0.121	0.119	0.055	0.055	0.055	0.103	0.120	0.104
AXP	0.207	0.216	0.207	0.331	0.240	0.331	0.021	0.021	0.021
BA	0.211	0.220	0.211	0.110	0.111	0.110	-0.003	-0.003	-0.003
BAC	0.319	0.240	0.319	0.152	0.143	0.152	0.478	0.283	0.479
BIIB	0.010	0.010	0.010	0.020	0.020	0.020	0.014	0.013	0.015
BK	0.357	0.418	0.357	0.315	0.237	0.315	0.209	0.218	0.211
BLK	0.216	0.227	0.216	0.135	0.138	0.136	0.080	0.081	0.081
BMY	0.159	0.163	0.159	0.018	0.018	0.018	0.155	0.159	0.172
BRK-B	0.142	0.145	0.142	-0.002	-0.002	-0.002	-0.009	-0.009	-0.009
C	0.295	0.225	0.295	0.021	0.021	0.021	0.003	0.003	0.003
CAT	0.122	0.124	0.122	0.248	0.265	0.250	0.142	0.183	0.143
CELG	0.152	0.155	0.152	0.007	0.007	0.007	-0.043	-0.045	-0.043
CL	0.227	0.240	0.227	-0.003	-0.003	-0.003	-0.015	-0.015	-0.015
CMCSA	0.205	0.213	0.205	0.062	0.063	0.062	-0.029	-0.031	-0.029
COF	0.262	0.282	0.262	0.308	0.254	0.308	0.161	0.170	0.161
COP	0.367	0.436	0.367	0.105	0.106	0.105	0.039	0.046	0.039
COST	0.126	0.128	0.126	0.245	0.261	0.245	-0.045	-0.045	-0.045
CSCO	0.048	0.048	0.048	0.096	0.096	0.096	-0.109	-0.111	-0.110
CVS	0.140	0.143	0.140	0.112	0.113	0.112	0.106	0.113	0.107
CVX	0.285	0.312	0.285	0.116	0.117	0.116	-0.063	-0.068	-0.064
DD	0.209	0.218	0.209	0.238	0.253	0.238	0.250	0.362	0.251
DHR	0.220	0.232	0.220	-0.001	-0.001	-0.001	-0.011	-0.011	-0.011
DIS	0.190	0.196	0.190	0.268	0.290	0.268	0.157	0.161	0.158
DOW	0.189	0.196	0.189	0.215	0.168	0.216	0.142	0.080	0.142
DUK	0.255	0.274	0.255	-0.004	-0.004	-0.004	-0.012	-0.012	-0.012
EMR	0.169	0.174	0.169	-0.001	-0.001	-0.001	-0.022	-0.023	-0.022
EXC	0.198	0.206	0.198	0.365	0.304	0.365	0.042	0.042	0.042
F	0.269	0.292	0.269	0.134	0.117	0.134	0.059	0.061	0.060
FB	0.032	0.032	0.032	0.125	0.127	0.144	0.010	0.007	0.010
FDX	0.172	0.177	0.172	0.137	0.139	0.137	0.094	0.094	0.094
FOX	0.240	0.255	0.240	0.452	0.516	0.452	0.337	0.165	0.339
FOXA	0.277	0.302	0.277	0.473	0.519	0.473	0.212	0.110	0.213
GD	0.189	0.196	0.189	0.284	0.311	0.284	0.239	0.146	0.239
GE	0.280	0.305	0.280	0.295	0.225	0.295	0.473	0.424	0.475
GILD	0.101	0.102	0.101	-0.008	-0.008	-0.008	-0.033	-0.033	-0.033
GM	0.097	0.098	0.098	0.003	0.003	0.003	0.391	0.319	0.394
GOOG	0.068	0.068	0.068	-0.003	-0.003	-0.003	-0.024	-0.024	-0.024
GOOGL	0.065	0.066	0.065	0.001	0.001	0.001	-0.012	-0.012	-0.012
GS	0.245	0.262	0.245	0.379	0.405	0.379	0.393	0.357	0.396
HAL	0.218	0.229	0.218	0.268	0.256	0.268	0.278	0.315	0.279
HD	0.228	0.242	0.228	0.277	0.323	0.277	-0.079	-0.080	-0.079
HON	0.202	0.210	0.202	0.335	0.382	0.336	0.217	0.155	0.218

Table A.8: (continued) Estimated ACF(1), MA(1) and AR(1) coefficients for squared log returns of S&P100 stocks.

Stock	d.ACF(1)	d.MA(1)	d.AR(1)	w.ACF(1)	w.MA(1)	w.AR(1)	m.ACF(1)	m.MA(1)	m.AR(1)
IBM	0.164	0.168	0.164	0.258	0.278	0.258	0.072	0.075	0.073
INTC	0.183	0.189	0.183	0.052	0.052	0.052	0.025	0.023	0.025
JNJ	0.212	0.221	0.212	0.382	0.461	0.382	-0.039	-0.049	-0.040
JPM	0.349	0.270	0.349	0.467	0.527	0.467	0.184	0.110	0.185
KHC	0.078	0.079	0.079	-0.051	-0.049	-0.051	-0.147	-0.151	-0.274
KMI	0.415	0.528	0.415	0.008	0.008	0.008	0.103	0.103	0.103
KO	0.287	0.314	0.287	-0.001	-0.001	-0.001	-0.013	-0.013	-0.013
LLY	0.260	0.280	0.260	0.191	0.198	0.191	0.112	0.114	0.113
LMT	0.250	0.268	0.250	0.186	0.192	0.186	0.072	0.072	0.072
LOW	0.131	0.134	0.131	0.321	0.361	0.321	0.058	0.058	0.059
MA	0.124	0.126	0.124	-0.002	-0.002	-0.002	-0.010	-0.010	-0.010
MCD	0.173	0.178	0.173	0.064	0.064	0.064	-0.008	-0.008	-0.008
MDLZ	0.104	0.105	0.104	-0.006	-0.006	-0.006	-0.037	-0.040	-0.037
MDT	0.096	0.097	0.096	0.184	0.190	0.184	0.294	0.368	0.295
MET	0.320	0.244	0.320	0.552	0.704	0.553	0.152	0.109	0.152
MMM	0.159	0.164	0.159	0.209	0.218	0.209	0.102	0.103	0.103
MO	0.142	0.145	0.142	-0.001	-0.001	-0.001	-0.004	-0.004	-0.004
MON	0.118	0.120	0.118	0.169	0.174	0.169	0.141	0.144	0.142
MRK	0.168	0.173	0.168	0.150	0.154	0.150	0.007	0.007	0.008
MS	0.248	0.158	0.248	0.425	0.578	0.425	0.167	0.172	0.167
MSFT	0.149	0.152	0.149	0.111	0.113	0.112	0.010	0.010	0.010
NKE	0.156	0.160	0.156	-0.002	-0.002	-0.002	-0.038	-0.041	-0.038
ORCL	0.158	0.162	0.158	0.084	0.084	0.084	0.094	0.094	0.094
OXY	0.230	0.244	0.230	0.020	0.020	0.020	0.189	0.188	0.190
PCLN	0.077	0.077	0.077	0.080	0.080	0.083	0.172	0.147	0.173
PEP	0.336	0.383	0.336	0.155	0.159	0.156	-0.049	-0.050	-0.049
PFE	0.179	0.184	0.179	0.132	0.135	0.133	0.385	0.342	0.387
PG	0.186	0.192	0.188	0.051	0.052	0.052	0.092	0.092	0.093
PM	0.125	0.127	0.125	0.106	0.107	0.106	0.026	0.027	0.026
PYPL	0.109	0.110	0.110	-0.107	-0.119	-0.143	0.475	0.386	0.538
QCOM	0.106	0.107	0.106	0.057	0.057	0.057	0.051	0.051	0.051
RTN	0.178	0.184	0.178	0.154	0.171	0.154	-0.079	-0.079	-0.079
SBUX	0.136	0.139	0.136	0.025	0.025	0.025	-0.017	-0.017	-0.017
SLB	0.183	0.189	0.183	0.216	0.227	0.216	0.154	0.141	0.155
SO	0.312	0.348	0.312	0.063	0.064	0.063	-0.014	-0.014	-0.014
SPG	0.347	0.402	0.347	0.527	0.531	0.527	0.264	0.218	0.265
T	0.139	0.141	0.139	0.307	0.342	0.307	0.016	0.016	0.016
TGT	0.198	0.206	0.198	0.499	0.443	0.500	0.219	0.186	0.220
TWX	0.174	0.179	0.174	0.003	0.003	0.003	0.047	0.048	0.047
TXN	0.101	0.102	0.101	0.139	0.142	0.139	0.009	0.008	0.009
UNH	0.152	0.156	0.152	0.298	0.269	0.298	0.004	0.002	0.004
UNP	0.125	0.127	0.125	-0.006	-0.006	-0.006	-0.008	-0.008	-0.008
UPS	0.160	0.164	0.160	0.191	0.198	0.192	0.003	0.003	0.003
USB	0.282	0.308	0.282	0.643	0.709	0.643	0.001	0.001	0.001
UTX	0.178	0.184	0.178	0.210	0.220	0.211	0.156	0.160	0.158
V	0.173	0.178	0.173	-0.003	-0.003	-0.003	-0.013	-0.013	-0.013
VZ	0.350	0.406	0.350	0.109	0.111	0.109	-0.063	-0.056	-0.063
WBA	0.098	0.099	0.098	0.110	0.114	0.110	-0.025	-0.025	-0.025
WFC	0.205	0.214	0.205	0.129	0.118	0.129	0.502	0.479	0.502
WMT	0.143	0.146	0.143	0.064	0.064	0.064	-0.114	-0.110	-0.115
XOM	0.323	0.364	0.323	0.145	0.148	0.145	-0.047	-0.065	-0.048