THIAGO WINKLER ALVES

FORECASTING DAILY VOLATILITY USING HIGH FREQUENCY FINANCIAL DATA
THIAGO WINKLER ALVES

FORECASTING DAILY VOLATILITY USING HIGH FREQUENCY FINANCIAL DATA

Dissertation presented to the Professional Master Program from the São Paulo School of Economics, part of Fundação Getulio Vargas, in partial fulfillment of the requirements for the degree of Master of Economics, with a major in Quantitative Finance.

Supervisor:
Prof. Dr. Juan Carlos Ruilova Terán

SÃO PAULO
2014
Winkler Alves, Thiago.
Forecasting Daily Volatility Using High Frequency Financial Data / Thiago
77 f.

Orientador: Prof. Dr. Juan Carlos Ruilova Terán.
Dissertação (MPFE) – Escola de Economia de São Paulo.

3. Mercado futuro. I. Ruilova Terán, Juan Carlos. II. Dissertação (MPFE) – Escola
de Economia de São Paulo. III. Título.

CDU 336
FORECASTING DAILY VOLATILITY USING HIGH FREQUENCY FINANCIAL DATA

Dissertation presented to the Professional Master Program from the São Paulo School of Economics, part of Fundação Getulio Vargas, in partial fulfillment of the requirements for the degree of Master of Economics, with a major in Quantitative Finance.

Date of Approval: August 6th, 2014

Examining Committee:

Prof. Dr. Juan Carlos Ruilova Terán
(Supervisor)
Fundação Getulio Vargas

Prof. Dr. Alessandro Martim Marques
Fundação Getulio Vargas

Prof. Dr. Hellinton Hatsuo Takada
Insper
To my parents, Jaqueline and Ary,
and to my little sister, Marina.
ACKNOWLEDGEMENTS

To my grandmothers and parents, who taught me and my sister how important studying is. She, even six years younger, keeps inspiring me with her daily conquests.

To Vicki, because revising the English in my writing was the smallest contribution she ever made me. Thank you for all the support, even when it meant I would be away.

To all my Professors, but especially to Prof. Dr. Juan Carlos Ruílova Terán and Prof. Dr. Alessandro Martim Marques, for all the knowledge and precious advice given.

To all my friends and colleagues, particularly to Wilson Freitas and Thársis Souza, who are always available to sit at a bar table and talk finance while drinking beer.

Finally, it is also important to thank Itaú Unibanco for the financial support. Without it, this degree would not be possible.
“Wahrlich es ist nicht das Wissen, sondern das Lernen,
nicht das Besitzen, sondern das Erwerben,
nicht das Da-Seyn, sondern das Hinkommen,
was den größten Genuss gewährt.”
(Johann Carl Friedrich Gauß)
ABSTRACT

Aiming at empirical findings, this work focuses on applying the HEAVY model for daily volatility with financial data from the Brazilian market. Quite similar to GARCH, this model seeks to harness high frequency data in order to achieve its objectives. Four variations of it were then implemented and their fit compared to GARCH equivalents, using metrics present in the literature. Results suggest that, in such a market, HEAVY does seem to specify daily volatility better, but not necessarily produces better predictions for it, what is, normally, the ultimate goal.

The dataset used in this work consists of intraday trades of U.S. Dollar and Ibovespa future contracts from BM&FBovespa.

**Keywords**: financial engineering. volatility forecast. high frequency financial data. futures market.
RESUMO

Objetivando resultados empíricos, este trabalho tem foco na aplicação do modelo HEAVY para volatilidade diária com dados financeiros do mercado Brasileiro. Muito similar ao GARCH, este modelo busca explorar dados em alta frequência para atingir seus objetivos. Quatro variações dele foram então implementadas e seus ajustes comparados a equivalentes GARCH, utilizando métricas presentes na literatura. Os resultados sugerem que, neste mercado, o HEAVY realmente parece especificar melhor a volatilidade diária, mas não necessariamente produz melhores previsões, o que, normalmente, é o objetivo final.

A base de dados utilizada neste trabalho consite de negociações intradiárias de contratos futuros de dólares americanos e Ibovespa da BM&FBovespa.

**Palavras-chave:** engenharia financeira. previsão de volatilidade. dados financeiros em alta frequência. mercado de futuros.
List of Figures

Figure 1 – Weak-sense stationarity example ................................. 20
Figure 2 – Random walk example ............................................. 22
Figure 3 – HEAVY (in black) vs GARCH (in gray) adjustment to volatility changes 42
List of Tables

Table 1 – Time series of AAPL prices ........................................ 18
Table 2 – Description of the dataset ........................................... 47
Table 3 – Summary statistics for the dataset ................................. 48
Table 4 – \( \chi^2(1) \) significance levels ...................................... 49
Table 5 – Parameters comparison: GARCH x GARCHX x HEAVY-r ...... 52
Table 6 – Models’ log-likelihood and likelihood-ratio test versus GARCHX .... 53
Table 7 – In-sample: HEAVY x GARCH .......................................... 54
Table 8 – In-sample with direct estimation: HEAVY x GARCH ............. 55
Table 9 – Out-of-sample: HEAVY x GARCH .................................... 55
Table 10 – Out-of-sample with direct estimation: HEAVY x GARCH ...... 56
Table 11 – Parameters and log-likelihood comparison: IGARCH x Int-HEAVY .. 56
Table 12 – In-sample: Int-HEAVY x IGARCH ................................... 57
Table 13 – Out-of-sample: Int-HEAVY x IGARCH ............................... 57
Table 14 – Parameters and log-likelihood comparison: GJR-GARCH x "GJR-HEAVY" 58
Table 15 – In-sample: "GJR-HEAVY" x GJR-GARCH .......................... 58
Table 16 – In-sample with direct estimation: "GJR-HEAVY" x GJR-GARCH .... 59
Table 17 – Out-of-sample: "GJR-HEAVY" x GJR-GARCH ...................... 59
Table 18 – Out-of-sample with direct estimation: "GJR-HEAVY" x GJR-GARCH . 59
Table 19 – Parameters and log-likelihood comparison: GJR-GARCH x Ext-HEAVY 60
Table 20 – In-sample: Ext-HEAVY x GJR-GARCH .......................... 60
Table 21 – In-sample with direct estimation: Ext-HEAVY x GJR-GARCH ...... 61
Table 22 – Out-of-sample: Ext-HEAVY x GJR-GARCH ....................... 61
Table 23 – Out-of-sample with direct estimation: Ext-HEAVY x GJR-GARCH ... 61
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>AutoRegressive</td>
</tr>
<tr>
<td>ARCH</td>
<td>AutoRegressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>ARIMA</td>
<td>AutoRegressive Integrated Moving Average</td>
</tr>
<tr>
<td>ARIMAX</td>
<td>ARIMA with Explanatory Variables</td>
</tr>
<tr>
<td>ARMA</td>
<td>AutoRegressive Moving Average</td>
</tr>
<tr>
<td>DOL</td>
<td>BM&amp;FBovespa’s U.S. Dollar Future Contract</td>
</tr>
<tr>
<td>EWMA</td>
<td>Exponentially Weighted Moving Average</td>
</tr>
<tr>
<td>Ext-HEAVY</td>
<td>Extended High-frequency-based Volatility</td>
</tr>
<tr>
<td>GARCH</td>
<td>Generalized AutoRegressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>GARCHX</td>
<td>GARCH with Explanatory Variables</td>
</tr>
<tr>
<td>GJR-GARCH</td>
<td>Glosten-Jagannathan-Runkle-GARCH</td>
</tr>
<tr>
<td>HEAVY</td>
<td>High-frequency-based Volatility</td>
</tr>
<tr>
<td>HFT</td>
<td>High Frequency Trading</td>
</tr>
<tr>
<td>IGARCH</td>
<td>Integrated Generalized AutoRegressive Conditional Heteroskedasticity</td>
</tr>
<tr>
<td>IND</td>
<td>BM&amp;FBovespa’s Ibovespa Future Contract</td>
</tr>
<tr>
<td>Int-HEAVY</td>
<td>Integrated High-frequency-based Volatility</td>
</tr>
<tr>
<td>MA</td>
<td>Moving Average</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum-Likelihood Estimation</td>
</tr>
<tr>
<td>MSRV</td>
<td>MultiScale Realized Variance</td>
</tr>
<tr>
<td>OLS</td>
<td>Ordinary Least Squares</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>RM</td>
<td>Realized Measure</td>
</tr>
<tr>
<td>RS</td>
<td>Realized Semivariance</td>
</tr>
<tr>
<td>RV</td>
<td>Realized Variance</td>
</tr>
<tr>
<td>WSS</td>
<td>Weak-Sense Stationary</td>
</tr>
</tbody>
</table>
Introduction

Early exchange houses (bourses) first appeared in the XIII century, and evolved into what we now know as the securities exchange market. Advances in computer technology in the last decades have taken these central pieces of world economy from places where one could find yelling traders to complex electronic trading systems. Evolution then was natural: investors want their orders to be executed before prices change; and the more trades happen during a day, the more bourses earn with fees. The competition between exchanges also raises the pace of such technological evolution, and an ever decreasing latency time became the objective of most of them.

Automated and algorithmic trading systems take a big role in today’s global economy. This move from manual trades allows investors to grow and diversify their portfolios more easily, and with faster and cheaper technology available, even more market agents will follow this trend. The necessary speed for making decisions requires accurate methods, justifying the study of high frequency trading (HFT) techniques and their effects.

The study of this new environment, created by financial markets that operate at a much higher speed, and are much more interconnected, is in the edge of research in finance\(^1\). It is also of great importance, not only because of all the capacity it has for generating profit, but mainly because of the understanding it can provide of the microstructure of the market, allowing both regulators and market players to keep it healthy and profitable (SCHMIDT, 2011).

Along with these technological advances, the amount of data that becomes available to researchers and market players significantly increases and econometric models that try to harness all this new information also emerge. Most of these models seek to uncover the dynamics of the prices of assets, and less attention is given to volatility models.

Different from prices, volatility is not observable, really difficult to precisely measure, and, as a consequence, not easy to model. But being able to do so is of great value, because it has direct application in things such as trading strategies and, more importantly, risk management.

When dealing only with daily information to model volatility, the first major recognized family - a family, because the number of parameters may vary - of models is ARCH (ENGL\(^E\)E, 1982), followed four years later by GARCH (BOLLERSLEV, 1986), which, along with its variations, is up to this day the most known and used family of daily volatility models.

\(^1\) [http://kolmogorov.math.stevens.edu/conference2013]
A recent volatility-focused proposal is the HEAVY family of univariate (SHEPPARD, 2010) and multivariate (NOURELDIN; SHEPHARD; SHEPPARD, 2012) models. One can look at it as a generalized version of GARCH which uses realized measures from each day to leverage information about daily volatility. This should not be confused with models such as HARCH (DACOROGNA et al., 1997), which use similar data to model volatility within a day (intraday), or even with the direct application of GARCH models on high frequency financial data.

This piece of work focused on implementing different versions of the univariate HEAVY family of models and compared it with GARCH models which offer similar characteristics. These were evaluated with the two most liquid future contracts from the São Paulo Stock Exchange (BM&FBovespa²) and an analysis of the quality of the results is presented.

Objectives

The main objective of this work is to apply a recent and non-standard approach to model daily volatility, one that uses high frequency financial data to leverage information, on data of the Brazilian market. This kind of data is not often explored in studies of such a market, mainly because it is less liquid than other international markets, where this kind of research is usually employed.

The goal is to check whether the results presented in the original paper, where the model is compared to more traditional approaches, are the same in a less friendly environment, as well as to encourage researchers to further explore developing markets. Besides that, some extensions to the model, which were proposed, but not tested by the original authors, are also deployed.

With these objectives in mind, the intent is to produce a text that anyone with basic knowledge in statistics and finance can understand. Some assumptions are that the reader knows what a random variable is and can recognize the functional form of a normal distribution. Also, knowledge of basic financial jargons and security classes, like futures, is necessary. Apart from these, all the underlying econometrics background will be covered in part I, which the more experienced reader can skip.

Algorithms and (properly commented) source codes will also be available, either as a form of annex or in GitHub³, so that everything is reproducible. Python, as opposed to any specialized econometrics commercial package, was the language of choice, both because it is free and because the author believes it has a simple syntax that anyone with basic programming skills can get around with.

² The dataset is available in <ftp://ftp.bmf.com.br/MarketData> and will be detailed in chapter 7.
³ <https://github.com/twalves/dissertation>
Organization

The remainder of this work is organized as follows:

a) Part I contains the first part of a literature review, focused in general econometrics:
   • chapter 1 presents basic terms and concepts that will be used in the entire text that follows;
   • chapter 2 gives an introduction to mean models, which will be essential for the less experienced reader to understand Part II;
   • chapter 3 finishes Part I explaining the basic aspects and methods of estimation theory.

b) Part II follows Part I as a second part of a literature review, focused in volatility models:
   • chapter 4 presents the more traditional family of GARCH models;
   • chapter 5 introduces the realized measures that the models presented in chapter 6 will take advantage of;
   • chapter 6 finishes part II by finally presenting the main models used in this work: the HEAVY models.

c) Part III presents the implementation of the models from part II:
   • chapter 7 gives details of the dataset;
   • chapter 8 introduces the evaluation measures that will be used in chapter 9;
   • chapter 9 presents experiments and results. Comparisons are made in order to evaluate the meaning of each result.

d) The conclusions of this work are presented.
Part I

Financial Series Modeling
1 General Definitions

In order for the reader to fully understand the work that follows, it is important that some terms and concepts are explained. This is the purpose of this first chapter.

1.1 Time Series

Let \( Y_t \) be a stochastic process, with values \( y_t \) occurring every any given interval of time (a month, a week, a day, or even just 5 minutes). The ordered sequence of data points, \( y_1, y_2, \ldots, y_{T-1}, y_T \), is called a time series, and the interval between each observation, \( \Delta t \), may or may not be always the same. It is important to notice, though, that if different \( \Delta t \)'s are used, each observed point cannot be treated equally.

Not surprisingly, the most studied stochastic process in finance is the price of assets. They are, often, measured daily (with equal intervals of time), using the value of the last trade of each day as the observations. This approach allows the study of less liquid securities, and also facilitates data retrieval, since more detailed financial information is sometimes difficult to gather (especially without paying for it). Daily data is even freely available through services like Yahoo! Finance.

To illustrate, the time series of Apple Inc. stock prices (in U.S. dollars), during the first two weeks of June, 2014, is shown in table 1 below:

<table>
<thead>
<tr>
<th>NASDAQ:AAPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014-06-02 89.81</td>
</tr>
<tr>
<td>2014-06-03 91.08</td>
</tr>
<tr>
<td>2014-06-04 92.12</td>
</tr>
<tr>
<td>2014-06-05 92.48</td>
</tr>
<tr>
<td>2014-06-06 92.22</td>
</tr>
<tr>
<td>2014-06-09 93.70</td>
</tr>
<tr>
<td>2014-06-10 94.25</td>
</tr>
<tr>
<td>2014-06-11 93.86</td>
</tr>
<tr>
<td>2014-06-12 92.29</td>
</tr>
<tr>
<td>2014-06-13 91.28</td>
</tr>
</tbody>
</table>

Table 1 – Time series of AAPL prices

1 Funny fact: when this happens, \( \Delta t \) itself is a time series.
2 <http://finance.yahoo.com>
1.2 Stationarity

When modeling either the price or the volatility dynamics of an asset for trading purposes, the main goal is trying to predict what is going to happen next, so that the strategy that amounts for the biggest return in the future is taken in the present. If the distribution of the underlying process is known, forecasting becomes easy, since it is just a matter of assuming that the statistical properties of the series will remain the same.

A time series is said to be stationary if its joint probability distribution, along with the distribution parameters (such as mean and variance), is constant over time. That is:

\[ f(y_{t_1}, \ldots, y_{t_k}) = f(y_{t_1+\tau}, \ldots, y_{t_k+\tau}), \quad \forall t_1, \ldots, t_k, \quad \forall k, \tau. \] (1.1)

A more relaxed definition is that of weak-sense stationary (WSS) processes. For \( Y_t \) to be WSS, the only requirements are:

- \( \mathbb{E}(|y_t|^2) < \infty, \quad \forall t; \)
- \( \mathbb{E}(y_t) = \mu, \quad \forall t; \)
- \( \text{Cov}(y_{t_j}, y_{t_k}) = \text{Cov}(y_{t_j+\tau}, y_{t_k+\tau}), \quad \forall t_j, t_k, \tau. \)

That is, the covariance of the process is not a function of time, but otherwise it is a function of the lag between each observation. Note that if \( t_j = t_k \):

\[ \text{Var}(y_{t_k}) = \text{Cov}(y_{t_k}, y_{t_k}) = \text{Cov}(y_{t_k+\tau}, y_{t_k+\tau}) = \text{Var}(y_{t_k+\tau}), \quad \forall t_k, \tau. \] (1.2)

Therefore, in a weak-sense stationary process, the expected value and the variance of the observations are always constant. These characteristics will be fundamental when finding the correct parameters to the models presented in the rest of this work. It is important to notice, however, that strictly stationarity does not imply in weak-sense stationarity.

Most raw financial data are far from being stationary, but it is possible to apply simple transformations to them in order to obtain WSS time series, the most common being the use of returns instead of prices. As will be seen in section 2.5, it is also useful to favor log returns over 'standard' returns.

Figure 1 shows plots for (a) Apple Inc. stock prices for the years of 2012 and 2013\(^3\) and (b) their log returns, the first being non-stationary and the second being WSS.

\(^3\) Already adjusted to the 7:1 split that occurred in 2014-06-09.
Chapter 1. General Definitions

20

(a) AAPL prices

(b) AAPL log returns

Figure 1 – Weak-sense stationarity example

1.3 Linear Regression

Econometrics is a field of economics which seeks to find relationships between economic variables by employing methods from statistics, mathematics and computer science. Most of these relationships are studied in the form of linear regressions, because estimating their parameters is relatively easy and yet they are powerful enough to describe how most financial variables relate to each other. All the models presented in this work are linear regressions.

A linear regression models how the value of independent variables affects the value of a dependent variable, having the following general form:

\[ y_t = \alpha + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \ldots + \beta_{n-1} x_{n-1,t} + \beta_n x_{n,t} + \epsilon_t, \tag{1.3} \]

or, simply:

\[ y_t = \alpha + \sum_{i=0}^{n} \beta_i x_{i,t} + \epsilon_t, \tag{1.4} \]

where:

- \( t \) represents the \( t^{th} \) observation of a variable;
- \( y \) is the dependent variable;
- \( x_i \) are the independent variables;
- \( \alpha \) is a parameter representing the intercept;
- \( \beta_i \) are parameters representing the weight that the values of each \( x_i \) have in the value of \( y \);
- \( \epsilon \) is the error, that is \( \epsilon_t = y_t - \alpha - \sum_{i=0}^{n} \beta_i x_{i,t} \).
A well known risk measure in the market is the $\beta$ of an investment (a single security or a basket), which relates its returns to the returns of a benchmark, often the market itself. This measure is calculated using a simple linear regression:

$$r_t = \alpha + \beta r_{b,t} + \epsilon_t,$$

where $r$ is the return of the investment, and $r_b$ is the return of the used benchmark. The noise $\epsilon$ may here be interpreted as an unexplained return, and $\alpha$ is the so called active-return of an investment. The investment follows the return of the benchmark by a factor of $\beta$, meaning that if the absolute value of this parameter is smaller than one, the investment is less volatile than the benchmark.

A good proxy for the market performance may be given by an index, such as the well known S&P 500. This particular index was used to find a $\beta$ value of approximately 0.98 for Apple Inc. stocks, using data for the years of 2012 and 2013. Details on how to estimate parameters of linear regressions will be briefly discussed in chapter 3.

### 1.4 High Frequency Financial Data

With the rise of systems capable of rapid performing automated and algorithmic trading strategies, popularly known as high frequency trading, the amount of data generated during a single trading day is enormous, sometimes in a ratio of a trade every millisecond, for each asset. This quantity of available high frequency intraday information allowed researchers to develop models as the HEAVY model that is studied in this work, because before technology allowed, there were not enough per day data to take advantage from. One of the first major publications in the subject is (GENÇAY et al., 2001).

Sometimes, the expression high frequency financial data refers to all the information found in an exchange’s order book, including not only effective trades, but all the bids and asks that occurred during a day. These may also appear in the literature as ultra high frequency financial data and can have many practical applications, but they are not the focus here. A model for the dynamics of the order book applied in the context of HFT was the subject of a dissertation presented by a previous master student from Fundação Getulio Vargas (NUNES, 2013) (in Portuguese).

As for this work, high frequency financial data accounts only for intraday trades. In contrast to daily data, these offer treatment problems of their own (GOODHART; O’HARA, 1997) and different statistical properties, as shown in (CONT, 2001). Every intraday trade will be accounted as being of high frequency, while daily returns (or anything with a smaller granularity) are considered to be of low frequency.
2 Mean Models

Even though volatility models may be applied completely separated from mean models, a good understanding of the latter is a valuable asset when learning them. This chapter aims to give the reader a review on the subject.

2.1 Random Walk

The most basic way to model the mean is through a random walk. It is a stochastic process where the next value of a variable $y_t$ is given by its previous value $y_{t-1}$ plus a random perturbation. That is:

$$y_t = y_{t-1} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} N(0, 1),$$

(2.1)

where $\epsilon_t$ is an independent and identically normally distributed random variable, with mean 0 and variance 1. An error of this form is also known in the literature as a white noise.

This process is a Markov chain, since it has no memory of older values, other than the last one. It is also a martingale\(^1\), i.e. $\mathbb{E}(y_{t+s} | \mathcal{F}_t) = y_t, \forall t, s$, so forecasting is impossible, since the expected value of any future iteration of $y_t$ will always be the current value. Figure 2 shows 30 simulations of a random walk process, with $y_0 = 0$, as if it had been observed daily during the years of 2012 and 2013.

\[\text{Figure 2 – Random walk example}\]

\(^1\) $\mathcal{F}_t$ is the symbol for $\sigma$-algebra, and here it means "all the information available until moment $t".
### 2.2 Autoregressive Models

As its name should already suggest, an autoregressive (AR) model of order $p$ is one of the form:

$$y_t = \omega + \alpha_1 y_{t-1} + \alpha_2 y_{t-2} + \ldots + \alpha_{p-1} y_{t-(p-1)} + \alpha_p y_{t-p} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, 1), \quad (2.2)$$

or, simply:

$$y_t = \omega + \sum_{i=1}^{p} \alpha_i y_{t-i} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, 1). \quad (2.3)$$

An AR$(p)$ is then a model where there is an autoregression of the dependent variable $y_t$ by its $p$ past values, $y_{t-1}, y_{t-2}, \ldots, y_{t-(p-1)}, y_{t-p}$, the independent variables of the model, with parameters $\omega, \alpha_1, \alpha_2, \ldots, \alpha_{p-1}, \alpha_p$.

If the process being modeled is stationary, some conclusions about the possible values of the parameters $\alpha_i$ may rise. For simplicity, consider an AR(1) model:

$$y_t = \omega + \alpha_1 y_{t-1} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0, 1). \quad (2.4)$$

If the expectation is taken:

$$\mathbb{E}(y_t) = \mathbb{E}(\omega + \alpha_1 y_{t-1} + \epsilon_t)$$
$$= \mathbb{E}(\omega) + \mathbb{E}(\alpha_1 y_{t-1}) + \mathbb{E}(\epsilon_t)$$
$$= \mathbb{E}(\omega) + \mathbb{E}(\alpha_1)\mathbb{E}(y_{t-1}) + \mathbb{E}(\epsilon_t)$$
$$= \omega + \alpha_1 \mathbb{E}(y_{t-1}). \quad (2.5)$$

For the process to be stationary, $\mathbb{E}(y_t)$ must be constant, $\forall t$:

$$\mathbb{E}(y_t) = \mathbb{E}(y_{t-1}) = \mu, \quad (2.6)$$

hence,

$$\mathbb{E}(y_t) - \alpha_1 \mathbb{E}(y_{t-1}) = \mu(1 - \alpha_1) = \omega, \quad (2.7)$$

and

$$\mu = \frac{\omega}{1 - \alpha_1}. \quad (2.8)$$

That is, $\alpha_1 \neq 1$ and, from (2.7), $\omega = \mu(1 - \alpha_1)$:

$$y_t = \omega + \alpha_1 y_{t-1} + \epsilon_t$$
$$= \mu(1 - \alpha_1) + \alpha_1 y_{t-1} + \epsilon_t$$
$$= \mu - \alpha_1 \mu + \alpha_1 y_{t-1} + \epsilon_t, \quad (2.9)$$

and, finally,

$$y_t - \mu = \alpha_1(y_{t-1} - \mu) + \epsilon_t. \quad (2.10)$$
Taking the variance from (2.10):

\[
\text{Var}(y_t) = \alpha_1^2 \text{Var}(y_{t-1}) + 1. \tag{2.11}
\]

\(\text{Var}(y_t)\) should also be constant, \(\forall t\), such that:

\[
\text{Var}(y_t) = \text{Var}(y_{t-1}) = \sigma^2, \tag{2.12}
\]

hence,

\[
\text{Var}(y_t) - \alpha_1^2 \text{Var}(y_{t-1}) = \sigma^2 (1 - \alpha_1^2) = 1, \tag{2.13}
\]

and

\[
\sigma^2 = \frac{1}{1 - \alpha_1^2}. \tag{2.14}
\]

Since variance is a positive quantity, \(|\alpha_1| < 1\).

### 2.2.1 Forecasting

When using AR\((p)\) models for forecasting, all one needs to do is to calculate the expected value of \(y_t\) \(s\)-steps ahead, \(s\) being the horizon of prediction, given the information available until the moment. The name \textit{mean model} comes exactly from this seek for the expected value, or the \textit{mean}.

Forecasting \(y_{t+3}\) using an AR\((2)\) model would be:

\[
\mathbb{E}(y_{t+3} \mid \mathcal{F}_t) = \mathbb{E}(\omega + \alpha_1 y_{t+2} + \alpha_2 y_{t+1} + \epsilon_{t+3} \mid \mathcal{F}_t)
\]

\[
= \omega + \alpha_1 \mathbb{E}(y_{t+2} \mid \mathcal{F}_t) + \alpha_2 \mathbb{E}(y_{t+1} \mid \mathcal{F}_t)
\]

\[
= \omega + \alpha_1 \mathbb{E}(\omega + \alpha_1 y_{t+1} + \alpha_2 y_t + \epsilon_{t+2} \mid \mathcal{F}_t) + \alpha_2 \mathbb{E}(\omega + \alpha_1 y_t + \alpha_2 y_{t-1} + \epsilon_{t+1} \mid \mathcal{F}_t)
\]

\[
= \omega + \alpha_1 \omega + \alpha_1^2 \mathbb{E}(y_{t+1} \mid \mathcal{F}_t) + \alpha_1 \alpha_2 y_t + \alpha_2 \omega + \alpha_2 \alpha_1 y_t + \alpha_2^2 y_{t-1}
\]

\[
= \omega (1 + \alpha_1 + \alpha_2) + 2 \alpha_1 \alpha_2 y_t + \alpha_2^2 y_{t-1} + \alpha_1^2 \mathbb{E}(y_{t+1} \mid \mathcal{F}_t)
\]

\[
= \omega (1 + \alpha_1 + \alpha_2) + 2 \alpha_1 \alpha_2 y_t + \alpha_2^2 y_{t-1} + \alpha_1^2 \mathbb{E}(\omega + \alpha_1 y_t + \alpha_2 y_{t-1} + \epsilon_{t+1} \mid \mathcal{F}_t)
\]

\[
= \omega (1 + \alpha_1 + \alpha_2) + 2 \alpha_1 \alpha_2 y_t + \alpha_2^2 y_{t-1} + \alpha_1^2 \omega + \alpha_1^3 y_t + \alpha_2^2 y_{t-1}
\]

\[
= \omega (1 + \alpha_1 + \alpha_2 + \alpha_1^2) + 2 \alpha_1 \alpha_2 y_t + \alpha_1^2 y_t + \alpha_2^2 y_{t-1} + \alpha_2^2 y_{t-1}. \tag{2.15}
\]
A possible Python implementation of 2.15 is:

```python
import numpy

def ar2_forecast(omega, alphas, y_past, steps=1):
    #
    # Returns the 'steps'-ahead forecast for the AR(2) model.
    #
    y = numpy.zeros(steps + 2)
    y[0] = y_past[0]
    y[1] = y_past[1]

    for i in range(2, steps + 2):
        y[i] = omega + alphas[0] * y[i-1] + alphas[1] * y[i-2]

    return numpy.copy(y[2:(steps + 2)])
```

2.3 Moving Average Models

A moving average (MA) model has a similar structure to an AR model, but instead of using the past values of the dependent variable as the independent variables, it uses the past perturbations. A MA\(q\) model is of the form:

\[
y_t = \omega + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + \ldots + \beta_{q-1} \epsilon_{t-(q-1)} + \beta_q \epsilon_{t-q} + \epsilon_t, \quad \epsilon_t \sim_{i.i.d.} N(0, 1),
\]

or, simply:

\[
y_t = \omega + \sum_{j=1}^{q} \beta_j \epsilon_{t-j} + \epsilon_t, \quad \epsilon_t \sim_{i.i.d.} N(0, 1).
\]

2.4 Autoregressive Moving Average Models

An autoregressive moving average (ARMA) model is a union of the two previously presented models. An ARMA\((p,q)\) has the following form:

\[
y_t = \omega + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \beta_j \epsilon_{t-j} + \epsilon_t, \quad \epsilon_t \sim_{i.i.d.} N(0, 1).
\]
2.5 Autoregressive Integrated Moving Average Models

When a stochastic process \( Y_t \) is not WSS, one possible way to transform it in one is to differentiate it. That is, instead of modeling \( y_t \), \( \Delta^1 y_t \) is modeled:

\[
\Delta^1 y_1 = y_t - y_{t-1}.
\]  

(2.19)

Two direct implications of this are:

- if \( \Delta^1 y_t \) is stationary and completely random, \( \Delta^1 y_t = \epsilon_t \), then \( y_t \) is a random walk;
- if \( y_t \) is the log price of an asset, \( \Delta^1 y_t \) represents its log return (as mentioned in section 1.2).

An ARIMA\((p, d, q)\) is a model in which the time series is differentiated \( d \)-times before the proper ARMA\((p, q)\) is used. As a consequence, modeling with an ARIMA\((p, 0, q)\) is exactly the same as direct applying an ARMA\((p, q)\).

\[
\Delta^d y_t = \omega + \sum_{i=1}^{p} \alpha_i \Delta^d y_{t-i} + \sum_{j=1}^{q} \beta_j \epsilon_{t-j} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0,1). \tag{2.20}
\]

The name autoregressive integrated moving average (ARIMA) comes from the fact that if one has a time series generated by this model, it is necessary to 'integrate' it \( d \)-times in order to obtain a series in the same unit as \( y_t \).

2.6 ARIMA with Explanatory Variables Models

Finally, an ARIMA with explanatory variables (ARIMAX) is an ARIMA with extra independent variables, which are neither autoregressives nor moving averages:

\[
y_t = \omega + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \beta_j \epsilon_{t-j} + \sum_{k=1}^{n} \gamma x_{k,t} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{2.21}
\]

where \( x_k \) are exogenous variables at instant \( t \), which can be any other variable that are known to somehow relate to \( y \).

Note that in the equation above \( y_t \) was used instead of \( \Delta^d y_t \). In favor of notation simplicity, from now on in this text, it will be assumed that any necessary differentiation is applied to the time series before modeling it \((d = 0)\), since, in practice, it makes no difference.
3 Estimation

Linear regression models were presented in both previous chapters, but until now no word was said about how to find the right values for each of the model’s parameters. The focus of this chapter is to give an introduction to two of the most used estimation methods: ordinary least squares and maximum likelihood. A simple AR(1) will be used to exemplify the application of both of them, but extending the idea to a more complex model should not be a problem.

3.1 Ordinary Least Squares

Given an AR(1) model:

\[ y_t = \omega + \alpha_1 y_{t-1} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} N(0, 1), \quad (3.1) \]

and reorganizing the equation:

\[ \epsilon_t = y_t - \omega - \alpha_1 y_{t-1}. \quad (3.2) \]

The goal of the Ordinary Least Squares (OLS) method is to find the values \( \hat{\omega} \) and \( \hat{\alpha}_1 \) for the parameters \( \omega \) and \( \alpha_1 \), respectively, that minimize the squared errors, given a sample of size \( T \). That is:

\[
\arg\min_{\omega, \alpha_1} \sum_{t=2}^{T} \epsilon_t^2 = \arg\min_{\omega, \alpha_1} \sum_{t=2}^{T} (y_t - \omega - \alpha_1 y_{t-1})^2 = \arg\min_{\omega, \alpha_1} f(\omega, \alpha_1). \quad (3.3)
\]

Note that the sum starts by the second element, since values for both \( y_t \) and \( y_{t-1} \) are needed. The estimated parameters \( \hat{\omega} \) and \( \hat{\alpha}_1 \) will be the solution of the following system of equations:

\[
\frac{\delta f}{\delta \omega} = -2 \sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) = 0; \quad (3.4)
\]

\[
\frac{\delta f}{\delta \alpha_1} = -2 \sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) y_{t-1} = 0. \quad (3.5)
\]

Solving (3.3) for \( \hat{\omega} \):

\[
-2 \sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) = 0
\]

\[
\sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) = 0
\]

\[
\sum_{t=2}^{T} y_t = \sum_{t=2}^{T} \hat{\omega} + \sum_{t=2}^{T} \hat{\alpha}_1 y_{t-1}. \quad (3.6)
\]
Chapter 3. Estimation

Rewriting:
\[ \sum_{t=2}^{T} y_t = \hat{\omega} \sum_{t=2}^{T} 1 + \hat{\alpha}_1 \sum_{t=2}^{T} y_{t-1}. \]  (3.7)

Now, dividing both sides by \((T - 1)\):
\[ \sum_{t=2}^{T} \frac{y_t}{T - 1} = \frac{\hat{\omega}}{T - 1} \sum_{t=2}^{T} 1 + \frac{\hat{\alpha}_1}{T - 1} \sum_{t=2}^{T} y_{t-1}. \]  (3.8)

Being \(\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}\) (the sample mean), it is possible to simplify (3.8):
\[ \bar{y}_t = \hat{\omega} + \hat{\alpha}_1 \bar{y}_{t-1}, \]  (3.9)
and finally find the right value for \(\hat{\omega}\):
\[ \hat{\omega} = \bar{y}_t - \hat{\alpha}_1 \bar{y}_{t-1}. \]  (3.10)

Now, substitute \(\hat{\omega}\) in (3.3):
\[
\begin{align*}
\arg\min_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} & [y_t - (\bar{y}_t - \hat{\alpha}_1 \bar{y}_{t-1} - \alpha_1 y_{t-1})]^2 \\
\arg\min_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} & [y_t - \bar{y}_t + \hat{\alpha}_1 \bar{y}_{t-1} - \alpha_1 y_{t-1}]^2 \\
\arg\min_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} & [(y_t - \bar{y}_t) - \hat{\alpha}_1(y_{t-1} - \bar{y}_{t-1})]^2.
\end{align*}
\]  (3.11)

Differentiating and solving for \(\hat{\alpha}_1\):
\[
\begin{align*}
-2 \sum_{t=2}^{T} (y_t - \bar{y}_t) - \hat{\alpha}_1(y_{t-1} - \bar{y}_{t-1})(y_{t-1} - \bar{y}_{t-1}) &= 0 \\
\sum_{t=2}^{T} (y_t - \bar{y}_t) - \hat{\alpha}_1(y_{t-1} - \bar{y}_{t-1})(y_{t-1} - \bar{y}_{t-1}) &= 0 \\
\sum_{t=2}^{T} (y_t - \bar{y}_t)(y_{t-1} - \bar{y}_{t-1}) - \hat{\alpha}_1(y_{t-1} - \bar{y}_{t-1})^2 &= 0 \\
\hat{\alpha}_1 &= \frac{\sum_{t=2}^{T}(y_t - \bar{y}_t)(y_{t-1} - \bar{y}_{t-1})}{(y_{t-1} - \bar{y}_{t-1})^2} = \frac{\text{Cov}(y_t, y_{t-1})}{\text{Var}(y_{t-1})}.
\end{align*}
\]  (3.12)

In section 1.3 the \(\beta\) measure was presented. To estimate its value using the OLS method, it is as simple as doing:
\[ \beta = \frac{\text{Cov}(r, r_b)}{\text{Var}(r_b)}. \]  (3.13)
3.2 Maximum-Likelihood

Given an AR(1) model:

\[ y_t = \omega + \alpha_1 y_{t-1} + \epsilon_t, \quad \epsilon_t \overset{i.i.d.}{\sim} N(0, 1), \quad (3.14) \]

and a sample \( y_1, y_2, \ldots, y_T \) of size \( T \). If the sample observations are assumed to be independent and identically distributed, it is known that the joint probability distribution of them is equal to the product of the their individual distributions, given the parameters \( \omega \) and \( \alpha_1 \):

\[ f(y_2, \ldots, y_{T-1}, y_T | \omega, \alpha_1) = f(y_2 | \omega, \alpha_1) \times \cdots \times f(y_{T-1} | \omega, \alpha_1) \times f(y_T | \omega, \alpha_1). \quad (3.15) \]

Since the presented models assume a distribution for the perturbation term (which is actually unknown), the objective of the maximum-likelihood estimation (MLE) is to find parameter values that make the sample distribution match the assumed one. For this, the likelihood function is defined as:

\[ \ell(\hat{\omega}, \hat{\alpha_1}; y_2, \ldots, y_T), \quad (3.16) \]

where \( \hat{\omega} \) and \( \hat{\alpha_1} \) are variable parameters and \( y_2, \ldots, y_T \) are fixed parameters. The goal is to find values for \( \hat{\omega} \) and \( \hat{\alpha_1} \) to which:

\[ \ell(\hat{\omega}, \hat{\alpha_1}; y_2, \ldots, y_T) = f(y_2, \ldots, y_T | \omega, \alpha_1) = \prod_{t=2}^{T} f(y_t | \omega, \alpha_1). \quad (3.17) \]

In order to do that, we maximize \( \ell \), and that is the reason the method is called maximum-likelihood. Given that all the sample is available, \( \forall t \geq 1 \):

\[ \mathbb{E}(y_t | \mathcal{F}_{t-1}) = \mathbb{E}(\omega + \alpha_1 y_{t-1} + \epsilon_t | \mathcal{F}_{t-1}) = \omega + \alpha_1 y_{t-1}, \quad (3.18) \]

and

\[ \text{Var}(y_t | \mathcal{F}_{t-1}) = \text{Var}(\omega + \alpha_1 y_{t-1} + \epsilon_t | \mathcal{F}_{t-1}) = 1. \quad (3.19) \]

Assuming that \( \epsilon_t \) is a white noise:

\[ \ell(\hat{\omega}, \hat{\alpha_1}; y_2, \ldots, y_T) = \prod_{t=2}^{T} f(y_t | \hat{\omega}, \hat{\alpha_1}) = \prod_{t=2}^{T} \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(y_t - \hat{\omega} - \hat{\alpha_1} y_{t-1})^2}{2} \right]. \quad (3.20) \]

The MLE estimates \( \hat{\omega} \) and \( \hat{\alpha_1} \) will be given by:

\[ \arg\max_{\hat{\omega}, \hat{\alpha_1}} \ell(\hat{\omega}, \hat{\alpha_1}; y_2, \ldots, y_T). \quad (3.21) \]
Since the maximum of a product may be very difficult to solve by hand, and may cause floating point problems in a computer, it is usual to calculate the maximum of the log-likelihood function. The result is the same, since log is a strictly monotonically increasing function.

\[
\arg\max_{\hat{\omega}, \hat{\alpha}_1} \ln \ell(\hat{\omega}, \hat{\alpha}_1 ; y_2, ..., y_T) = \arg\max_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} \ln \left\{ \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1})^2}{2} \right] \right\}
\]

\[
= \arg\max_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} \ln \left( \frac{1}{\sqrt{2\pi}} \right) - \frac{(y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1})^2}{2}
\]

\[
= \arg\max_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} \ln 1 - \ln \sqrt{2\pi} - \frac{(y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1})^2}{2}
\]

\[
= \arg\max_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} \frac{1}{2} \ln 2\pi - \frac{(y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1})^2}{2}
\]

\[
= \arg\max_{\hat{\omega}, \hat{\alpha}_1} \sum_{t=2}^{T} \frac{1}{2} [\ln 2\pi - (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1})^2].
\] (3.22)

The estimated parameters \( \hat{\omega} \) and \( \hat{\alpha}_1 \) will be the solution of the following system of equations:

\[
\frac{\delta f}{\delta \hat{\omega}} = \sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) = 0;
\] (3.23)

\[
\frac{\delta f}{\delta \hat{\alpha}_1} = \sum_{t=2}^{T} (y_t - \hat{\omega} - \hat{\alpha}_1 y_{t-1}) y_{t-1} = 0.
\] (3.24)

Which, in this case, yield exactly the same results as the ones of the OLS method, (3.10) and (3.12).

It is important to mention, though, that in both methods, after finding the values for \( \hat{\omega} \) and \( \hat{\alpha}_1 \), one would still need to show that they are in fact a minimum point, in case of OLS, or a maximum point, in case of MLE, of their respective objective functions. This can be accomplished by simply differentiating the function again and will not be shown here.
Part II

Volatility Models
4 GARCH Family

Following the study of mean models, part II focuses completely on volatility models, and this chapter is responsible for introducing the most known and used family of daily volatility models in the market (when one seeks for something more robust than just standard deviations). Afterwards, GARCH models will be used as benchmarks in the tests of chapter 9.

Up until this point, it was assumed that the variable being modeled could be any one. From now on, the text will implicitly assume that the variable of interest is the volatility of returns of a given asset.

4.1 ARCH

Introduced in (ENGLE, 1982), the autoregressive conditional heteroskedasticity (ARCH) model attempts to give a dynamics to the perturbations of the mean models, other than just assuming they are all white noises. A proper definition of an ARCH of order $q$ is of the following form:

$$
\epsilon_t = \sigma_t \eta_t, \quad \eta_t \overset{i.i.d.}{\sim} N(0, 1), \quad (4.1)
$$

$$
\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \alpha_2 \epsilon_{t-2}^2 + \cdots + \alpha_q \epsilon_{t-(q-1)}^2 + \alpha_q \epsilon_{t-q}^2, \quad (4.2)
$$

In order to find the right parameters to this model, though, it has to be combined with a mean model, like the ones from chapter 2, so that finding values for $\epsilon_t$, $\forall t$, becomes feasible. If it is the desire of the user to use it standalone, as it was in this work, one needs to assume the log prices of the underlying asset to be random walks, so that their log returns are given only by:

$$
r_t = \epsilon_t. \quad (4.3)
$$

This way, ARCH may be simplified to:

$$
r_t = \sigma_t \eta_t, \quad \eta_t \overset{i.i.d.}{\sim} N(0, 1), \quad (4.4)
$$

$$
\sigma_t^2 = \omega + \sum_{j=1}^{q} \alpha_j r_{t-j}^2, \quad (4.5)
$$

and anyone can use it to model and forecast volatility without properly modeling $r_t$.

If historical price information is available, it is possible to ignore (4.1) completely, using returns to directly model their volatility with (4.5). The sections that follow will assume this is the only goal, and no attention will be given to the possible dynamics $r_t$ may have: GARCH and its variants will be presented only with the equation for $\sigma_t^2$. A more formal definition would require one to explicitly write the equation of $\epsilon_t$ as well.
4.2 GARCH

Although ARCH was the first model of this family, the generalized autoregressive conditional heteroskedasticity (GARCH) (BOLLERSLEV, 1986) is the most known one. The main difference from its predecessor is that it employs an AR-like dynamics to the model. A GARCH(p, q) may be specified as:

\[
\sigma_t^2 = \omega_G + \alpha_{G,1}r_{t-1}^2 + \alpha_{G,2}r_{t-2}^2 + \ldots + \alpha_{G,q-1}r_{t-(q-1)}^2 + \alpha_{G,q}r_{t-q}^2 \\
+ \beta_{G,1}\sigma_{t-1}^2 + \beta_{G,2}\sigma_{t-2}^2 + \ldots + \beta_{G,p-1}\sigma_{t-(p-1)}^2 + \beta_{G,p}\sigma_{t-p}^2, \tag{4.6}
\]

or, simply:

\[
\sigma_t^2 = \omega_G + \sum_{j=1}^{q} \alpha_{G,j}r_{t-j}^2 + \sum_{i=1}^{p} \beta_{G,i}\sigma_{t-i}^2. \tag{4.7}
\]

Therefore, in this model, \(\sigma_t^2\) is a regression of the q past values of \(r^2\) plus an autoregression (AR) of its own p past values, with parameters \(\omega_G, \alpha_{G,1}, \ldots, \alpha_{G,q}, \beta_{G,1}, \ldots, \beta_{G,p}\). The subscripted \(G\) will be important later when differentiating the estimated parameters in the experiments, since the Greek letters \(\omega, \alpha, \beta\) are used across all models which will be tested. Choosing different letters would probably just cause more confusion, so the subscript will be adopted from now on.

Like it was done before with the AR model, if \(R_t\) is supposed to be WSS, it is possible to find restrictions to the possible values of \(\alpha_{G,j}, \beta_{G,i}\) as well. Consider a GARCH(1, 1) model:

\[
\sigma_t^2 = \omega_G + \alpha_{G,1}r_{t-1}^2 + \beta_{G,1}\sigma_{t-1}^2. \tag{4.8}
\]

Taking the expectation of \(\sigma_t^2\) is the same as taking the variance of \(r\):

\[
\mathbb{E}(\sigma_t^2) = \mathbb{E}(\omega_G + \alpha_{G,1}r_{t-1}^2 + \beta_{G,1}\sigma_{t-1}^2) \\
= \mathbb{E}(\omega_G) + \mathbb{E}(\alpha_{G,1}r_{t-1}^2) + \mathbb{E}(\beta_{G,1}\sigma_{t-1}^2) \\
= \mathbb{E}(\omega_G) + \mathbb{E}(\alpha_{G,1})\mathbb{E}(r_{t-1}^2) + \mathbb{E}(\beta_{G,1})\mathbb{E}(\sigma_{t-1}^2) \\
= \omega_G + \alpha_{G,1}\mathbb{E}(r_{t-1}^2) + \beta_{G,1}\mathbb{E}(\sigma_{t-1}^2). \tag{4.9}
\]

From (4.4), it is known that \(\mathbb{E}(r_t^2) = \mathbb{E}(\sigma_t^2)\) and, for the process to be stationary, \(\mathbb{E}(\sigma_t^2)\) must be a constant, \(\forall t:\)

\[
\mathbb{E}(\sigma_t^2) = \mathbb{E}(\sigma_{t-1}^2) = \mathbb{E}(r_{t-1}^2) = \sigma^2, \tag{4.10}
\]

hence,

\[
\mathbb{E}(\sigma_t^2) - \alpha_{G,1}\mathbb{E}(r_{t-1}^2) - \beta_{G,1}\mathbb{E}(\sigma_{t-1}^2) = \sigma^2(1 - \alpha_{G,1} - \beta_{G,1}) = \omega_G, \tag{4.11}
\]

and

\[
\sigma^2 = \frac{\omega_G}{1 - \alpha_{G,1} - \beta_{G,1}}. \tag{4.12}
\]

Since variance is a positive quantity, \(\omega_G, \alpha_{G,1}, \beta_{G,1} > 0\) and \((\alpha_{G,1} + \beta_{G,1}) < 1\).
Chapter 4. GARCH Family

4.2.1 Estimating

For all the models that are going to be used in the experiments of chapter 9, the MLE was the method of choice to estimate the parameters. Since GARCH will be used in its GARCH(1, 1) form, this will be the example here.

Remember that, the objective of this method is to find values for \( \hat{\omega}_G, \hat{\alpha}_{G,1}, \) and \( \hat{\beta}_{G,1} \) to which:

\[
\ell(\hat{\omega}_G, \hat{\alpha}_{G,1}, \hat{\beta}_{G,1} ; r_2, ..., r_T) = f(r_2, ..., r_T | \omega_G, \alpha_{G,1}, \beta_{G,1})
\]

\[
= \prod_{t=2}^{T} f(r_t | \omega_G, \alpha_{G,1}, \beta_{G,1}). \tag{4.13}
\]

Given that all the sample is available, \( \forall t \geq 1 \):

\[
\mathbb{E}(r_t | \mathcal{F}_{t-1}) = \mathbb{E}(\sigma_t \eta_t | \mathcal{F}_{t-1}) = 0, \tag{4.14}
\]

and

\[
\text{Var}(r_t | \mathcal{F}_{t-1}) = \text{Var}(\sigma_t \eta_t | \mathcal{F}_{t-1})
\]

\[
= \mathbb{E}(\sigma_t^2 | \mathcal{F}_{t-1})
\]

\[
= \mathbb{E}(\omega + \alpha_{G,1} r_{t-1}^2 + \beta_{G,1} \sigma_{t-1}^2 | \mathcal{F}_{t-1})
\]

\[
= \omega_G + \alpha_{G,1} r_{t-1}^2 + \beta_{G,1} \sigma_{t-1}^2. \tag{4.15}
\]

Assuming that \( \eta_t \) is a white noise:

\[
\ell(\hat{\omega}_G, \hat{\alpha}_{G,1}, \hat{\beta}_{G,1} ; r_2, ..., r_T) = \prod_{t=2}^{T} f(r_t | \hat{\omega}_G, \hat{\alpha}_{G,1}, \hat{\beta}_{G,1}), \tag{4.16}
\]

which is equal to:

\[
\prod_{t=2}^{T} \frac{1}{\sqrt{2\pi(\hat{\omega}_G + \hat{\alpha}_{G,1} r_{t-1}^2 + \hat{\beta}_{G,1} \sigma_{t-1}^2)}} \exp \left[ -\frac{r_t^2}{2(\hat{\omega}_G + \hat{\alpha}_{G,1} r_{t-1}^2 + \hat{\beta}_{G,1} \sigma_{t-1}^2)} \right]. \tag{4.17}
\]

The MLE estimates \( \hat{\omega}_G, \hat{\alpha}_{G,1}, \) and \( \hat{\beta}_{G,1} \) will be given by:

\[
\argmax_{\omega_G, \alpha_{G,1}, \beta_{G,1}} \ell(\hat{\omega}_G, \hat{\alpha}_{G,1}, \hat{\beta}_{G,1} ; r_2, ..., r_T). \tag{4.18}
\]

The last part of the procedure is quite similar the one presented in section 3.2. In a computer, one may use an optimizer to find the maximum-likelihood estimator for each of the parameters. This will be exemplified in subsection 6.1.1 with the HEAVY model, but the same method is easily adapted to work with GARCH and the source code is available in annex A.
4.2.2 Forecasting

If one wants to know the probable value $\sigma_t^2$ will have $s$-steps ahead in the future, as with the AR model, all that is necessary is to calculate its expected value.

Forecasting $\sigma_{t+3}^2$ using a GARCH(1, 1) model then is:

$$
E(\sigma_{t+3}^2 | \mathcal{F}_t) = E(\omega_G + \alpha_{G,1}r_{t+2}^2 + \beta_{G,1}\sigma_{t+2}^2 | \mathcal{F}_t)
$$

$$
= \omega_G + \alpha_{G,1}E(r_{t+2}^2 | \mathcal{F}_t) + \beta_{G,1}E(\sigma_{t+2}^2 | \mathcal{F}_t)
$$

$$
= \omega_G + \alpha_{G,1}E(\sigma_{t+2}^2 | \mathcal{F}_t) + \beta_{G,1}E(\sigma_{t+2}^2 | \mathcal{F}_t)
$$

$$
= \omega_G + (\alpha_{G,1} + \beta_{G,1})E(\sigma_{t+2}^2 | \mathcal{F}_t)
$$

$$
= \omega_G + (\alpha_{G,1} + \beta_{G,1})[\omega_G + \alpha_{G,1}E(r_{t+1}^2 | \mathcal{F}_t) + \beta_{G,1}E(\sigma_{t+1}^2 | \mathcal{F}_t)]
$$

$$
= \omega_G + (\alpha_{G,1} + \beta_{G,1})[\omega_G + \alpha_{G,1}E(\sigma_{t+1}^2 | \mathcal{F}_t) + \beta_{G,1}E(\sigma_{t+1}^2 | \mathcal{F}_t)]
$$

$$
= \omega_G + (\alpha_{G,1} + \beta_{G,1})[\omega_G + (\alpha_{G,1} + \beta_{G,1})E(\sigma_{t+1}^2 | \mathcal{F}_t)]
$$

$$
= \omega_G + \omega_G(\alpha_{G,1} + \beta_{G,1}) + (\alpha_{G,1} + \beta_{G,1})^2E(\sigma_{t+1}^2 | \mathcal{F}_t)
$$

$$
= \omega_G(1 + \alpha_{G,1} + \beta_{G,1}) + (\alpha_{G,1} + \beta_{G,1})^2E(\sigma_{t+1}^2 | \mathcal{F}_t)
$$

$$
= \omega_G(1 + \alpha_{G,1} + \beta_{G,1}) + (\alpha_{G,1} + \beta_{G,1})^2(\omega_G + \alpha_{G,1}r_{t+1}^2 + \beta_{G,1}\sigma_{t+1}^2 | \mathcal{F}_t)
$$

$$
= \omega_G(1 + \alpha_{G,1} + \beta_{G,1}) + (\alpha_{G,1} + \beta_{G,1})^2(\omega_G + \alpha_{G,1}r_{t+1}^2 + \beta_{G,1}\sigma_{t+1}^2) . \quad (4.19)
$$

Since $(\alpha_{G,1} + \beta_{G,1}) < 1$, $\sigma_t^2$ will slowly, but surely, mean revert. The algorithm to compute such forecast will also be left to be exemplified in subsection 6.1.2, for the HEAVY case. Again, the same method is easily adapted to work with GARCH and the source code is available in annex A.

4.3 IGARCH

The Integrated Generalized Autoregressive Conditional Heteroskedasticity (IGARCH) looks exactly like a standard GARCH:

$$
\sigma_t^2 = \omega_{IG} + \sum_{j=1}^{q} \alpha_{IG,j}r_{t-j}^2 + \sum_{i=1}^{p} \beta_{IG,i}\sigma_{t-i}^2 , \quad (4.20)
$$

but it is in fact a restricted version of it, where parameters $\alpha_{IG,j}$ and $\beta_{IG,i}$ sum up to one:

$$
\sum_{j=1}^{q} \alpha_{IG,j} + \sum_{i=1}^{p} \beta_{IG,i} = 1 . \quad (4.21)
$$

The motivation behind it is to persist the model shocks in the conditional variance, so that they remain important while forecasting (BOLLERSLEV; ENGLE, 1993). As should be concluded from (4.12), with IGARCH, the stochastic process $R_t$ (for which
$r_t$, $\forall t > 0$, are the realizations) is not WSS. (NELSON, 1990) shows, however, that for a GARCH(1,1)-generated process to be strictly stationary, the necessary condition is $E[\ln(\alpha_G \eta_t^2 + \beta_G)] < 0$, a weaker condition than $E(\alpha_G \eta_t^2 + \beta_G) < 1$.

Given (4.21), when $q = p = 1$, it is possible to rewrite:

$$\sigma_t^2 = \omega_G + \alpha_G r_{t-1}^2 + (1 - \alpha_G)\sigma_{t-1}^2.$$  \hspace{1cm} (4.22)

The letters $IG$ will be used to designate IGARCH parameters.

### 4.4 GARCH with Explanatory Variables

A GARCH with Explanatory Variables (GARCHX) is to GARCH as ARIMAX is to ARIMA: a GARCH with extra independent variables, which are neither past values of $r_t$ nor of $\sigma_t^2$:

$$\sigma_t^2 = \omega_GX + \sum_{j=1}^{q} \alpha_GX_r r_{t-j}^2 + \sum_{i=1}^{p} \beta_GX \sigma_{t-i}^2 + \sum_{k=1}^{n} \gamma_GX \xi_{k,t}.$$  \hspace{1cm} (4.23)

As it will be further seen in chapter 9, this model will be specially useful to produce a joint GARCH-HEAVY model, so that one is able to evaluate which of the two is more descriptive of the daily volatility of the security in question. The letters $GX$ will be used to designate GARCHX parameters.

### 4.5 GJR-GARCH

The idea behind the GARCH extension proposed in (GLOSTEN; JAGANNATHAN; RUNKLE, 1993) is that future increases in the volatility of returns are associated with present falls in asset prices. To capture this statistical leverage effect, the three authors which gave their initials to the name of the GJR-GARCH model propose the following:

$$\sigma_t^2 = \omega_{GJR} + \sum_{j=1}^{q} \alpha_{GJR} r_{t-j}^2 + \sum_{k=1}^{a} \gamma_{GJR} \psi_{t-k}^2 I_{t-k} + \sum_{i=1}^{p} \beta_{GJR} \sigma_{t-i}^2,$$  \hspace{1cm} (4.24)

with $I_t = 1$ if $r_t < 1$, and $I_t = 0$ otherwise, $\forall t$. Adding a new term to the equation every time a negative return occurred in the past, heightening the effect squared returns have in the resulted volatility.

Since it is not possible to predict whether a negative return will happen, when forecasting with an horizon of size $s$, $s > 1$, the evaluations from chapter 9 will assume:

$$E(I_{t+s} | \mathcal{F}_t) = E(I_{t+s}) \approx \frac{1}{2},$$  \hspace{1cm} (4.25)

causing the constraints to this models’ parameters to be slightly different from the ones from GARCH. In the GJR-GARCH(1,1,1) case: $\left(\alpha_{GJR,1} + \frac{\gamma_{GJR}}{2} + \beta_{GJR,1}\right) < 1$. 

5 Realized Measures

After presenting the most known GARCH models, the concept of realized measures (RM) need to be explained before continuing, because they are in the heart of the HEAVY models, which are the focus of this work and will be presented in the next chapter.

As GARCH used the daily returns of an asset, \( r_t \), as its main source of information, the respective realized measures, \( RM_t \), are what HEAVY utilizes. They are nonparametric-based estimators of the variance of a security in a day, which, as opposed to \( r_t^2 \), ignore overnight effects, and sometimes even the variation of the first moments of a given day, since these may be regarded as more noisy than the rest of the trading session.

Both (BARNDORFF-NIELSEN; SHEPHARD, 2006) and (ANDERSEN; BOLLERSLEV; DIEBOLD, 2002) give a background on the subject, and the idea here is only to present the realized measures that were used during the tests with the HEAVY family of models in this work.

5.1 Realized Variance

The most simply of the realized measures, realized variance (RV), is the given by the sum of intraday squared returns. If \( p_\tau \) is the log price of an asset at instant \( \tau \), then:

\[
RM_t = RV_t = \sum_{\tau=1}^{n_t} (p_\tau - p_{\tau-1})^2,
\]

where \( n_t \) is the number of used intraday prices within day \( t \). As already pointed out in section 1.1, \( \Delta \tau \), the interval between each observation, needs to be a constant to allow equal treatment for each \( p_\tau \). In this work, a \( \Delta \tau \) of five minutes (300s) was used. The source code for the Python implementation of \( RV_t \), given a intraday time series \( ts \), follows:

```python
import numpy
import pandas

def realized_variance(ts, delta=300, base=0):
    ts = ts.resample(rule=(str(delta) + 's'), how='last', fill_method='pad', base=base)
    ts = numpy.diff(ts)
    ts = numpy.power(ts, 2.0)
    return sum(ts)
```
5.2 Multiscale Realized Variance

The microstructure of the market may be very noisy in practice, and there are several studies available that propose estimators to mitigate this effect. Some examples are: pre-averaging (JACOD et al., 2009); realized kernels (BARNDORFF-NIELSEN et al., 2008); two scale realized variance (ZHANG; MYKLAND; AÏT-SAHALIA, 2005); and its successor, the multiscale realized variance (MSRV), presented in (ZHANG et al., 2006). The latter was chosen given its simplicity of implementation.

Given realized variances of different scales, \( K \):

\[
RV^K_t = \frac{1}{K} \sum_{\tau=K+1}^{n_t} (p_\tau - p_{\tau-K})^2.
\]

The MSRV is defined as:

\[
RM_t = \text{MSRV}_t = \sum_{i=1}^{M_t} \alpha_i RV^i_t,
\]

where \( M_t \) is the quantity of averaged scales, with its optimal value for a sample of size \( n_t \) being in the order of \( O(\sqrt{n_t}) \). The weights \( \alpha_i \) are defined as:

\[
\alpha_i = \frac{12i}{M_t^2} \left( \frac{i}{M_t} - \frac{1}{2} - \frac{1}{2M_t} \right),
\]

and the MSRV as whole may be implemented as follows:

```python
from numpy.core.umath import floor, subtract
import numpy
import pandas

def multiscale_realized_variance(ts, delta=300, base=0):
    result = 0.0
    ts = ts.resample(rule=(str(delta) + 's'), how='last', fill_method='pad',
                     base=base)
    m = int(floor(numpy.sqrt(len(ts))))
    m2 = numpy.power(m, 2.0)
    for i in range(1, m + 1):
        scaled_ts = subtract(ts[i:len(x)], x[0:len(ts)−i])
        scaled_ts = numpy.power(scaled_ts, 2.0)
        result += ((sum(scaled_ts) / i) * 12.0 * (i / m2)
                    * (((i / m) − 0.5 − (1.0 / (2.0 * m)))
                        / (1.0 − (1.0 / m2))))
    return result
```
5.3 "Realized EWMA"

Normally used to measure daily volatility, the exponentially weighted moving average (EWMA) is a special case of IGARCH(1, 1), where \( \omega_{IG} = 0 \), and has the form:

\[
\sigma_t^2 = (1 - \lambda)\sigma_{t-1}^2 + \lambda \sigma_{t-1}^2.
\] (5.5)

However, the name EWMA is commonly associated in the market to a particular implementation of this model, the one from RiskMetrics\textsuperscript{TM} (MORGAN, 1996), which sets \( \lambda = 0.94 \).

Usually, when employed in high frequency financial data, EWMA serves as an estimator to volatility within a day, as opposed to the other realized measures presented until this point, which measure the open-to-close variance of intraday prices. The idea of a 'Realized EWMA' is to measure a RV with weights, hoping that the last returns during a day should have more importance to the overall dynamics. As it will be later presented in chapter 9, this measure, even being quite simple to implement, is very representative. In a similar notation to the previous measures:

\[
RM_t = EWMA_t = (n_t - 1) \sum_{\tau=1}^{n_t} (0.06)(0.94)^{n_t-\tau}(p_\tau - p_{\tau-1})^2.
\] (5.6)

Without the factor \((n_t - 1)\), this would be the normally employed version of EWMA. With this factor, the measure is now in the same magnitude of \( r^2 \). For this to work properly, though, the weights should also be normalized, as it is shown in the algorithm below:

```python
import numpy
import pandas
def realized_ewma(ts, delta=300, base=0):
    ts = ts.resample(rule=('{}s'.format(delta)), how='last', fill_method='pad', base=base)
    ts = numpy.diff(ts)
    ts = numpy.power(ts, 2.0)
    weights = 0.06 * numpy.power(0.94, range(len(ts) - 1, -1, -1))
    weights /= sum(weights)
    return sum(weights * ts * (len(ts) - 1.0))
```

Here, the Python function `range` is used 'backwards', creating a list with values \([n_t, n_t - 1, \ldots, 1, 0]\) (the last parameter is the `step`).
5.4 Subsampling

Another way to diminish the effect of microstructure noise is to average across subsamples. This is achieved by simply changing the start point of the sampling process: instead of counting intervals of size $\Delta \tau$ from the beginning of the trading day, these intervals begin after a delay of fifteen or thirty seconds (or any other amount of time).

Subsampling is theoretically always beneficial (HANSEN; LUNDE, 2006) and, like the authors of HEAVY did with their RV estimator, it was applied in the three realized measures presented in this chapter. This was done using starting points every thirty seconds, up to four minutes and a half, creating ten different subsamples which were then averaged. The values presented for these measures in part III always consider subsampling.

Implementing it is quite easy, as it may be seen from the code below. Here, `fun` may be the name of any function that calculates a RM, given that they all have the same header.

```python
def subsampling(ts, fun, bins=10, bin_size=30):
    result = 0.0
    delta = bins * bin_size
    for i in range(bins):
        result += fun(ts, delta=delta, base=(i * bin_size))
    return result / bins
```

5.5 Realized Semivariance

As the GJR-GARCH, presented in section 4.5, tries to measure the effect of statistical leverage, section 6.4 will present an extended version of the HEAVY model which utilizes realized semivariances (RS) with the same intent. These are realized variances that only make use of negative returns (BARNDORFF-NIELSEN; KINNEBROCK; SHEPHARD, 2008), that is:

$$RS_t = \sum_{\tau=1}^{n_t} (p_\tau - p_\tau-1)^2 I_\tau,$$

with $I_\tau = 1$ if $(p_\tau - p_\tau-1) < 1$, and $I_\tau = 0$ otherwise, $\forall \tau^1$. Applying the same in the multiscale case would be more tricky, since the weights would have to follow a different equation. This was not attempted here.

---

1. Implementing it in both realized variance and 'realized EWMA' is quite easy: one just needs to substitute the line of code `ts = numpy.power(ts, 2.0)` by `ts = numpy.power(ts[ts < 0.0], 2.0)` in both functions.
6 HEAVY Family

This chapter is, finally, responsible for presenting the main subject of this work: the HEAVY family of models. The objective is to measure the efficiency of this family proposed in (SHEPHARD; SHEPPARD, 2010) as an alternative to GARCH models in the Brazilian market.

The first two sections that follow present the models experimented in the original paper, while the last two sections are dedicated to models that were proposed by the original authors as possible extensions to the standard HEAVY. In chapter 9 all four variants will be tested against their GARCH counterparts to check whether they do or do not stand out in this particular market, as well as if the extensions are really worth applying.

6.1 HEAVY

In section 4.2, the GARCH(1, 1) was presented as:

\[ \sigma_t^2 = \omega_G + \alpha_G r_{t-1}^2 + \beta_G \sigma_{t-1}^2. \]  

(6.1)

If all the information until moment \( t \) is available, it is known that:

\[ \text{Var}(r_{t+1} | \mathcal{F}_t^{LF}) = \sigma_{t+1}^2 = \omega_G + \alpha_G r_t^2 + \beta_G \sigma_t^2, \quad (\alpha_G + \beta_G) < 1, \]  

(6.2)

where the superscript \( LF \) stands for low frequency, that is, GARCH only makes use of daily information of security prices variation.

The high-frequency-based volatility (HEAVY) models are specified as:

\[ \text{Var}(r_{t+1} | \mathcal{F}_t^{HF}) = h_{t+1} = \omega_H + \alpha_H R_{Mt} + \beta_H h_t, \quad \beta_H < 1; \]  

(6.3)

\[ \text{E}(R_{Mt+1} | \mathcal{F}_t^{HF}) = \mu_{t+1} = \omega_{RM} + \alpha_{RM} R_{Mt} + \beta_{RM} \mu_t, \quad (\alpha_{RM} + \beta_{RM}) < 1, \]  

(6.4)

where (6.3) is named HEAVY-r, which models the close-to-close conditional variance (as GARCH does), and (6.4) is named HEAVY-RM, modeling the open-to-close variation.

Note that, now, the superscript \( HF \) is used, from high frequency, since this family of models is designed to harness high frequency financial data to predict daily asset return volatility. This is accomplished by using one of the realized measures presented in chapter 5 as the main source of information, as opposed to square returns, like GARCH.

\(^1\) For the rest of the text, the index in each parameter is going to be suppressed, since models of higher order will not be referenced anymore.
Another important remark is that if one only needs to do 1-step ahead forecasts, only HEAVY-r is needed, since HEAVY-RM exists only as a companion model for when multistep-ahead predictions are necessary.

As it will be seen in chapter 9, when compared to GARCH, HEAVY models differ especially in:

- they typically estimate smaller values for $\beta$, with $\omega$ being really close to zero, so they tend to be a weighted sum of recent realized measures (having more momentum), while GARCH models normally present longer memory;
- they also tend to adjust faster to changes in the level of volatility, like it can be seen in figure 3 (already using Brazilian data for U.S. Dollar future contracts - HEAVY in black, GARCH in gray).

![Figure 3 – HEAVY (in black) vs GARCH (in gray) adjustment to volatility changes](image)

When explaining estimation and forecasting in the HEAVY case, the approach will be different to that used with GARCH or AR, where mathematical demonstrations were used. Since the methodology employed in the HEAVY model is quite similar the one from its counterpart, here the focus will be in its algorithmic implementation. The `heavyr` function that appear in the two following subsections has this form:

```python
def heavyr(omega, alpha, beta, rm_last, h_last):
    h = omega + alpha * rm_last + beta * h_last
    return h
```
Chapter 6. HEAVY Family

6.1.1 Estimating

As suggested in the original paper, both HEAVY-r and HEAVY-RM models are estimated separately. The MLE is the method of choice, with the only difference being the parameters for the probability density function (PDF).

HEAVY-r utilizes a PDF with mean 0 and variance $h_t$, with $r_t$ as the realizations:

$$ f(r_t | 0, h_t), \quad (6.5) $$

while HEAVY-RM utilizes a PDF with mean 0 and variance $\mu_t$, with $RM_t^{\frac{1}{2}}$ as the realizations:

$$ f(RM_t^{\frac{1}{2}} | 0, \mu_t). \quad (6.6) $$

Then, it is necessary to use an optimizer with a function like this one:

```python
import loglikelihood
import numpy

def heavyr_likelihood(parameters, r2_data, rm_data, h0, lag=1, tstudent_df=0.0, estimating=True):
    t = len(r2_data)
    h = numpy.repeat(h0, t)
    for i in range(lag, t):
        h[i] = heavyr(parameters[0], parameters[1], parameters[2],
                       rm_data[i - lag], h[i - lag])
    if estimating:
        return loglikelihood.generic_loglikelihood(r2_data, h,
                                                    df=tstudent_df, sign=-1.0)
    else:
        return loglikelihood.generic_loglikelihood(r2_data, h,
                                                    df=tstudent_df, sign=1.0), numpy.copy(h)
```

The parameter $lag$ is used for direct estimation (explained in section 8.3); the boolean $estimating$ signalizes whether the log-likelihood should be multiplied by $-1$ (since optimizers normally minimize); and the function `generic_loglikelihood` works with both a normal or a Student’s t-distribution (when $df \neq 0$).

Both `generic_loglikelihood` and the function that calls the optimizer (with proper bounds and constraints definitions) are available in annex A.
6.1.2 Forecasting

Forecasting is as easy as employing the algorithm below, where parameter \texttt{steps} determines the prediction horizon, and the variable \texttt{lag} (determined by the length of the available past information) is used for direct forecasting (accompanying direct estimation).

```python
import numpy

def heavy_forecast(h_parameters, mu_parameters, rm_past, h_past, mu_past, 
                   steps=1, full_output=False):
    lag = len(rm_past)
    h = numpy.zeros(steps)
    mu = numpy.zeros(steps)

    for i in range(0, steps):
        if i < lag:
            h[i] = heavyr(h_parameters[0], h_parameters[1], 
                           h_parameters[2], rm_past[i], h_past[i])
            mu[i] = heavyrm(mu_parameters[0], mu_parameters[1], 
                             mu_parameters[2], rm_past[i], mu_past[i])
        else:
            h[i] = heavyr(h_parameters[0], h_parameters[1], 
                           h_parameters[2], mu[i - lag], h[i - lag])
            mu[i] = heavyrm(mu_parameters[0], mu_parameters[1], 
                             mu_parameters[2], mu[i - lag], mu[i - lag])

    if full_output:
        return numpy.copy(h), numpy.copy(mu)
    else:
        return numpy.copy(h)
```
6.2 Int-HEAVY

The HEAVY family alternative for IGARCH has the following form:

\[ h_t = \omega_{IH} + \alpha_{IH} RM_{t-1} + \beta_{IH} h_{t-1}, \quad \beta_{IH} < 1; \]  
\[ \mu_t = \alpha_{IRM} RM_{t-1} + (1 - \alpha_{IRM}) \mu_{t-1}, \quad \alpha_{IRM} < 1, \]  

(6.7) (6.8)

With (6.7) being exactly equal to (6.3) and changes only happening to HEAVY-RM. Notice that in this model, not only is \( \alpha_{RM} + \beta_{RM} = 1 \), but the trend parameter \( \omega_{RM} \) is also completely removed.

6.3 "GJR-HEAVY"

"GJR-HEAVY" is obviously not how the authors named this model, but on account of its similarity to GJR-GARCH, this is the name used during the tests to refer to it:

\[ h_t = \omega_{GH} + \alpha_{GH} RM_{t-1} + \gamma_{GH} RM_{t-1} I_{t-1} + \beta_{GH} h_{t-1}, \quad \beta_{IH} < 1; \]  
\[ \mu_t = \omega_{GRM} + \alpha_{GRM} RM_{t-1} + \gamma_{GRM} RM_{t-1} I_{t-1} + \beta_{GRM} \mu_{t-1}, \]  
\[ (\alpha_{GRM} + \frac{\gamma_{GRM}}{2} + \beta_{GRM}) < 1, \]  

(6.9) (6.10)

with \( I_{t-1} = 1 \) if \( r_{t-1} < 1 \), and \( I_{t-1} = 0 \) otherwise, \( \forall t \), and the same assumption adopted with GJR-HEAVY:

\[ \mathbb{E}(I_{t+s} | \mathcal{F}_t) = \mathbb{E}(I_{t+s}) \approx \frac{1}{2}, \]  

(6.11)

6.4 Extended HEAVY

This extended version of HEAVY will be referred to as Ext-HEAVY during the experiments from chapter 9, following the nomenclature choice of the original authors for the Int-HEAVY model. It uses realized semivariances to accomplish the same statistic leverage effects sought by GJR-GARCH, and that is the reason it will be compared with this model in the tests (along with "GJR-HEAVY"). It presents the following form:

\[ h_t = \omega_{XH} + \alpha_{XH} RM_{t-1} + \gamma_{XH} RS_{t-1} + \beta_{XH} h_{t-1}, \quad \beta_{XH} < 1; \]  
\[ \mu_t = \omega_{XRM} + \alpha_{XRM} RM_{t-1} + \beta_{XRM} \mu_{t-1}, \quad (\alpha_{XRM} + \beta_{XRM}) < 1; \]  
\[ \mu^*_t = \omega_{XRS} + \alpha_{XRS} RS_{t-1} + \beta_{XRS} \mu^*_{t-1}, \quad (\alpha_{XRS} + \beta_{XRS}) < 1, \]  

(6.12) (6.13) (6.14)

with (6.14) modeling the open-to-close semivariation.
Part III

Application


## 7 Dataset Description

For the experiments that will follow, the two most liquid future contracts from BM&FBovespa were chosen. They are:

- BRL/USD, which has the official ticker in the format DOLMYY\(^1\) and will be represented in the tests as DOL;

- BRLxIbovespa, where Ibovespa is the main stock index of the Exchange. This contract has its official ticker in the format INDMYY\(^1\) and will be represented as IND in the reports of this work.

The database was constructed entirely with market data available for free\(^2\) at <ftp://ftp.bmf.com.br/MarketData>, and the procedure to do so is explained in appendix A. Table 2 gives a digest of the dataset after it was cleaned:

<table>
<thead>
<tr>
<th>Asset</th>
<th>Ave dur</th>
<th>Start date</th>
<th>End date</th>
<th>T</th>
<th>Intraday</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRL/USD</td>
<td>4.673s</td>
<td>2012-01-03</td>
<td>2013-12-30</td>
<td>489(^3)</td>
<td>3385698</td>
<td>6923.717</td>
</tr>
<tr>
<td>BRLxIbovespa</td>
<td>4.335s</td>
<td>2012-01-03</td>
<td>2013-12-30</td>
<td>489(^3)</td>
<td>3573470</td>
<td>7307.709</td>
</tr>
</tbody>
</table>

Table 2 – Description of the dataset

With each column of the table representing:

- Ave dur: the average duration between intraday trades;

- T: the total number of days in the sample;

- Intraday: the total number of intraday trades;

- Average: the average number of trades within each day of the sample.

For each of the two future contracts, the database holds information about daily returns, squared returns, realized variance, multiscale realized variance, "realized EWMA", realized semivariance, and "realized semiEWMA". For all the realized measures listed, the first half an hour of each day was ignored, since these tend to be more noisy. The duration of thirty minutes was chosen using the resulted log-likelihood in the estimation of the

\(^1\) M is a letter representing the month of maturity of the contract, and YY is a 2-digit number representing the year of maturity.

\(^2\) It is limited to the last two years of historical data.

\(^3\) Four days were removed from the time series. More details are available in appendix A.
models as the metric: without these observations, the function had a higher value in most cases. They were all also subsampled.

Table 3 shows the summary statistics for each of the measures calculated in the the dataset:

<table>
<thead>
<tr>
<th></th>
<th>BRL/USD</th>
<th></th>
<th>BRLxIbovespa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avol</td>
<td>SD</td>
<td>ACF&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td>( r^2 )</td>
<td>12.746</td>
<td>1.350</td>
<td>0.087</td>
</tr>
<tr>
<td>RV</td>
<td>9.764</td>
<td>0.349</td>
<td>0.481</td>
</tr>
<tr>
<td>MSRV</td>
<td>9.065</td>
<td>0.406</td>
<td>0.472</td>
</tr>
<tr>
<td>EWMA</td>
<td>8.535</td>
<td>0.326</td>
<td>0.281</td>
</tr>
<tr>
<td>RS</td>
<td>6.949</td>
<td>0.211</td>
<td>0.389</td>
</tr>
<tr>
<td>SemiEWMA</td>
<td>6.384</td>
<td>0.183</td>
<td>0.277</td>
</tr>
</tbody>
</table>

Table 3 – Summary statistics for the dataset

Here, Avol stands for annualized volatility, and is calculated using the following:

\[
\text{Avol} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} 252 x_i},
\]

for which the squared returns present higher values. This is probably because they account for overnight returns, while the other measures do not.

Realized measures, in general, have a much smaller standard deviation (SD), and have a higher serial correlation<sup>4</sup>. Note that EWMA, however, stands out from its cousins, since it presents even smaller Avol and SD values, although showing a smaller ACF<sub>1</sub>. The smaller values for the semivariances are expected.

Most of these results are in line with the ones available in the original HEAVY study, except that, in their findings, currencies used to have no difference in the Avol values for squared returns and realized measures. Presumably, the overnight effects were slighter for them. Also, the Avol values for indices were a bit more distant in their difference.

Table 3 was computed in its totality using daily and intraday returns multiplied by 100, giving a rough idea of percent change. The experiments in chapter 9 use the same kind of data, to smooth the work of the optimizer. Except for \( \omega \), which almost always tend to zero, all other parameters should not be affected by applying this technique.

<sup>4</sup> ACF<sub>1</sub> is the autocorrelation function at lag 1.
8 Evaluation Metrics

This chapter will present a brief description of the comparison metrics used during the experiments from next chapter. The concepts of direct estimation and forecasting will also be discussed.

8.1 Likelihood-ratio Test

To measure whether squared returns or realized measures give more information when describing volatility, the following GARCHX(1, 1, 1) model is built:

$$\sigma_t^2 = \omega_{GX} + \alpha_{GX} r_{t-1}^2 + \beta_{GX} \sigma_{t-1}^2 + \gamma_{GX} R_{Mt-1},$$

and compared with both GARCH and HEAVY using the likelihood-ratio test:

$$D = -2 \ln \left( \frac{\text{likelihood for null model}}{\text{likelihood for alternative model}} \right)$$

$$= -2 \ln(\text{likelihood for null model}) + 2 \ln(\text{likelihood for alternative model}), \quad (8.2)$$

where the alternative model has more parameters than the null model. Since a model with more parameters will always have a likelihood value greater (or at least equal) than a model with less parameters, this quantity is positive. If the null model has $m$ parameters and the alternative model has $n$ parameters, $D$ has a $\chi^2$ probability distribution with $n - m$ degrees of freedom.

In chapter 9 however, the reader will notice that this quantity is being reported as a negative value. This is because the GARCHX model in 8.1 was used as the alternative model in the test to give an idea of loss of representativeness, so the absolute value of $D$, $|D|$, should be considered when calculating the probability of $D$.

The studied scenario will always have $|D| \sim \chi^2(1)$, and its value should be greater than the following values for the test to be accepted with a determined significance level:

<table>
<thead>
<tr>
<th>$\chi^2(1)$</th>
<th>95%</th>
<th>96%</th>
<th>97%</th>
<th>98%</th>
<th>99%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.841</td>
<td>4.217</td>
<td>4.709</td>
<td>5.411</td>
<td>6.634</td>
</tr>
</tbody>
</table>

Table 4 – $\chi^2(1)$ significance levels
Chapter 8. Evaluation Metrics

8.2 Loss Function

The QLIK loss function will be used to measure the s-step ahead forecast quality of each model:

\[
\text{loss}(r_{t+s}^2, \hat{\sigma}_{t+s}^2) = \frac{r_{t+s}^2}{\hat{\sigma}_{t+s}^2} - \ln \left( \frac{r_{t+s}^2}{\hat{\sigma}_{t+s}^2} \right) - 1, \quad \forall s > 0, \quad (8.3)
\]

where \( r_{t+s}^2 \) is the proxy for the variance at time \( t + s \), and \( \hat{\sigma}_{t+s}^2 \) is the prediction made at time \( t \). As exposed by the authors of the HEAVY model, the studies in (PATTON; SHEPPARD, 2009) and (PATTON, 2011) show that this loss function is robust to noise in the proxy. To compare HEAVY to GARCH, the following is used:

\[
L_{t,s} = \text{loss}(r_{t+s}^2, h_{t+s}|t) - \text{loss}(r_{t+s}^2, \sigma^2_{t+s}|t)
\]

\[
= \left[ \frac{r_{t+s}^2}{h_{t+s} + \ln(h_{t+s}|t)} \right] - \left[ \frac{r_{t+s}^2}{\sigma^2_{t+s}|t} + \ln(\sigma^2_{t+s}|t) \right] = -2 \ln \frac{f(r_{t+s} \mid 0, h_{t+s}|t)}{f(r_{t+s} \mid 0, \sigma^2_{t+s}|t)}, \quad (8.4)
\]

with \( f \) the probability density function of the normal distribution. Having this functional form, negative values of \( L_{t,s} \) favor HEAVY in relation to GARCH, in a s-steps ahead forecast.

There is also the cumulative loss function, described as:

\[
\text{loss} \left( \sum_{i=0}^{s} r_{t+i}^2, \sum_{i=0}^{s} \hat{\sigma}_{t+i|^t}^2 \right) = \frac{\sum_{i=0}^{s} r_{t+i}^2}{\sum_{i=0}^{s} \sigma^2_{t+i|^t}} - \ln \left( \frac{\sum_{i=0}^{s} r_{t+i}^2}{\sum_{i=0}^{s} \sigma^2_{t+i|^t}} \right) - 1, \quad \forall s > 0, \quad (8.5)
\]

and the respective comparison form. Note, however, that:

\[
\sum_{i=0}^{s} \text{loss}(r_{t+i}^2, \hat{\sigma}_{t+i|^t}^2) \neq \text{loss} \left( \sum_{i=0}^{s} r_{t+i}^2, \sum_{i=0}^{s} \hat{\sigma}_{t+i|^t}^2 \right). \quad (8.6)
\]

Both pointwise and cumulative functions will be used to compare the forecast performance of each model in the HEAVY family against its counterpart in the GARCH family. Then, a \( t \)-statistic will be computed:

\[
t = \frac{\hat{L}_{t,s}}{\sqrt{S^2/n}}, \quad (8.7)
\]

where \( \hat{L}_{t,s} \) is the sample mean, \( S^2 \) the sample variance, and \( n \) the sample size. The value \( t \) has a Students’ \( t \) probability distribution with \( n - 1 \) degrees of freedom. Since the size \( n \) will be determined by the kind of evaluation being made (for example, if it is in or out of sample), the \( p \)-value for the respective Students’ \( t \)-distribution, with a pre-determined significance level will be reported along with the experiment.
8.3 Direct Estimation and Forecasting

Normally, when estimating, one would use the following equation for the volatility values to be used as input to the PDF in the MLE method (in the HEAVY-r case):

\[
h_t = \omega_H + \alpha_H R M_{t-1} + \beta_H h_{t-1},
\]
and this can be properly used in order to achieve an \( s \)-steps ahead forecast. This may, of course, cause loss of information if some time is spent before the parameters are estimated again.

From time to time, also, the user of the model already knows in advance that the goal is to forecast values for an horizon of size \( s \neq 1 \). When this happens, it is possible to estimate the values of the parameters with such an horizon in mind. This is called direct estimation:

\[
h_t = \omega_H + \alpha_H R M_{t-s} + \beta_H h_{t-s}.
\]

Then, with the right parameter values in hand, direct forecasting can significantly improve the results from the desired prediction:

\[
\mathbb{E}(h_{t+s} | \mathcal{F}_t) = \mathbb{E}(\omega_H + \alpha_H R M_t + \beta_H h_t | \mathcal{F}_t)
= \omega_H + \alpha_H R M_t + \beta_H h_t.
\]

In next chapter, when using the comparison functions presented in section 8.2, both one-step ahead estimation and direct estimation will be used, for different horizons of prediction.
9 Experiments and Results

This final chapter is reserved to present the results from the experiments realized to compare the effectiveness of the HEAVY model against the GARCH model in the Brazilian market scenario. It is important to make it clear, though, that these results are limited in scope and should not overlap the results published by the original authors of the model.

First, an analysis of the representativeness both realized measures and squared returns seem to have in the regression of volatility is shown. Afterwards, each version of the HEAVY model presented in chapter 6 will be compared with its GARCH counterpart from chapter 4, both in-sample and out-of-sample, using the loss functions presented in section 8.2.

9.1 Realized Measures versus Squared Returns

To test whether realized measures or squared returns are more effective in giving information about the historical values of volatility, a GARCHX model was estimated using all three realized measures (each at a time) as exogenous variables. The estimated parameters appear in table 5:

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>GARCH</th>
<th>GARCHX</th>
<th>HEAVY-r</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha_G$</td>
<td>$\beta_G$</td>
<td>$\alpha_{GX}$</td>
</tr>
<tr>
<td>DOL</td>
<td>RV</td>
<td>0.094</td>
<td>0.894</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.000</td>
<td>0.883</td>
<td>0.232</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.000</td>
<td>0.814</td>
<td>0.398</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>0.032</td>
<td>0.932</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.000</td>
<td>0.913</td>
<td>0.083</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.007</td>
<td>0.923</td>
<td>0.079</td>
</tr>
</tbody>
</table>

Table 5 – Parameters comparison: GARCH x GARCHX x HEAVY-r

In this table, as in all others that will follow up in this chapter, the parameter values for the GARCH model are shown in the lines corresponding to the RV measure. This is just to avoid the creation of another line just for properly accommodating $r^2$, given that it would only be used by this model.
Recall from section 8.1 that, here, $\alpha_{GX}$ is the weight for $r^2_{t-1}$ and $\gamma_{GX}$ is the weight for $RM_{t-1}$. It is really interesting to see that, apart from the IND + EWMA scenario, all other $\alpha_{GX}$ values are near zero, and, even in this case, it is quite small. The $\gamma_{GX}$ values, on the other hand, are almost identical the values of $\alpha_H$.

Parameter values, however, do not give the whole picture. The likelihood-ratio test, presented in section 8.1, was chosen in order to perform a better verification of this matter. The results are in table 6:

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>log-likelihood</th>
<th>likelihood-ratio test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>GARCH</td>
<td>GARCHX</td>
</tr>
<tr>
<td>DOL</td>
<td>RV</td>
<td>-550.849</td>
<td>-534.850</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>-534.940</td>
<td>-534.940</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-529.766</td>
<td>-529.766</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>-859.615</td>
<td>-856.538</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>-857.139</td>
<td>-857.139</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-857.466</td>
<td>-857.544</td>
</tr>
</tbody>
</table>

Table 6 – Models’ log-likelihood and likelihood-ratio test versus GARCHX

The columns on the left show the log-likelihood value for each of the models, with different realized measures. They are the maximum values found by the optimizer and are negative because, as explained in chapter 7, the intraday returns used to build the timeseries were multiplied by 100.

The columns on the right show twice the likelihood change for both GARCH and HEAVY when compared to their GARCHX cousins. Note that only the IND + EWMA combination suffers a small loss in the HEAVY case. This is in line with the parameters values in table 5. Nevertheless, when GARCH is compared to GARCHX, the result is completely different, and the loss of information is substantial.

From table 4, in section 8.1, it is possible to affirm that the results from table 6, in the U.S. Dollar case, are statistically significant, with 99% of confidence. In the case of Ibovespa, it is plausible to say, with a significance level of 95%, that the realized measures do provide more information than the squared returns, when modeling volatility.
9.2 HEAVY versus GARCH

From the previous section, it is reasonable for one to believe that the forecast produced by modeling the series of volatility using the HEAVY model would be better than doing so with GARCH. This is true in most of the cases, but, in several experiments, the difference is not significant enough for one to be able to affirm this, as it will be seen in the following sections.

First, though, it is interesting to see that the HEAVY parameters are estimated as promoted. That is, the \( \alpha_H \) values, presented in table 5, are always higher than the \( \alpha_G \) values. Even in the Ibovespa case, where these values are not even 0.1, they are still about three times higher than \( \alpha_G \).

In-Sample

For the in-sample experiments, all the available data points (489) were used for both sampling and forecasting. This means that, for the tests from section 8.2 to be accepted with a level of significance of 95%, the absolute value of the computed \( t \)-statistic should be greater than 1.96 in each scenario.

Table 7 presents the results for both pointwise and cumulative loss functions, when comparing HEAVY to GARCH estimated to perform better at 1-step ahead forecasts. The numbers corresponding to the name of each column represent the forecast horizon that was utilized in the test. Remember, negative values favor HEAVY.

<table>
<thead>
<tr>
<th>Asset</th>
<th>( t &gt; 1.96 )</th>
<th>Pointwise</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RM</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>DOL</td>
<td>RV</td>
<td>-2.83</td>
<td>-2.42</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-3.04</td>
<td>-2.75</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>-1.04</td>
<td>-0.66</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>-0.91</td>
<td>-0.89</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-0.68</td>
<td>-0.23</td>
</tr>
</tbody>
</table>

Table 7 – In-sample: HEAVY x GARCH

One can notice that, for U.S. dollars, the tests indicate that HEAVY really outperforms GARCH. This is only not true for a horizon of size 10 in the pointwise case, with the cumulative result for the same horizon being quite convincing.

It is not possible to affirm the same regarding Ibovespa. Even though HEAVY consistently outperforms GARCH (with the exception of two cases), the difference is never statistically significant.
When direct estimating (see section 8.3), the results follow the same trend as with the 1-step ahead estimation:

\[
\begin{array}{cccccc}
     & t > 1.96 & \text{Pointwise} \\
\hline
\text{Asset} & \text{RM} & 2 & 3 & 5 & 10 \\
\hline
\text{DOL} & \text{RV} & -3.32 & -1.83 & -3.12 & -1.33 \\
          & \text{MSRV} & -2.14 & -1.69 & -2.24 & -0.31 \\
          & \text{EWMA} & -3.14 & -2.86 & -2.65 & -1.21 \\
\hline
\text{IND} & \text{RV} & -0.82 & -0.49 & -1.17 & -0.91 \\
          & \text{MSRV} & -1.06 & -0.15 & -0.60 & -0.03 \\
          & \text{EWMA} & -0.60 & -0.29 & -0.85 & -0.48 \\
\end{array}
\]

Table 8 – In-sample with direct estimation: HEAVY x GARCH

Here the only difference being that GARCH never outperforms HEAVY.

**Out-of-Sample**

For the out-of-sample experiments, a moving window of 451 days (roughly one year and 10 months) was used, so that the other 38 available days could be used as sample for each of the tested horizons of prediction. This means that, for the tests from section 8.2 to be accepted with a level of significance of 95%, the absolute value of the computed \( t \)-statistic should be greater than 2.02 in each scenario.

The likelihood-ratio test from section 9.1 and the in-sample results from previous subsection should indicate that DOL adapts better than IND to the HEAVY models. This is not what the out-of-sample experiments show. Table 9 presents the numbers for 1-step ahead estimations of the models:

\[
\begin{array}{cccccc}
     & t > 2.02 & \text{Pointwise} & \text{Cumulative} \\
\hline
\text{Asset} & \text{RM} & 1 & 2 & 3 & 5 & 10 & 2 & 3 & 5 & 10 \\
\hline
\text{DOL} & \text{RV} & 0.46 & -0.94 & -0.59 & -1.03 & -0.90 & -1.46 & 0.08 & 0.27 & -0.90 \\
          & \text{MSRV} & -0.39 & -0.39 & -0.40 & -0.95 & -0.11 & -0.70 & -0.22 & -0.29 & 0.36 \\
          & \text{EWMA} & -0.45 & -0.21 & 0.44 & -0.33 & -0.12 & -1.08 & -0.23 & -0.11 & 0.15 \\
\hline
\text{IND} & \text{RV} & -0.84 & -1.15 & -1.68 & -0.47 & -1.68 & -1.56 & -2.46 & -2.58 & -3.47 \\
          & \text{MSRV} & -1.52 & -1.96 & -2.23 & -1.31 & -0.26 & -2.46 & -3.19 & -3.42 & -2.89 \\
          & \text{EWMA} & 0.98 & 0.64 & 0.63 & 0.55 & -2.44 & 0.82 & 0.46 & 0.22 & -3.22 \\
\end{array}
\]

Table 9 – Out-of-sample: HEAVY x GARCH

This time, then, the statistically different results appear only for Ibovespa. For both assets, however, it is possible to find horizons where GARCH outperforms HEAVY.
Table 9’s most intriguing combination is, certainly, IND + EWMA: for most horizons, GARCH outperforms HEAVY, but in the only true differences, HEAVY is better. This happens in the 10-step ahead prediction, and the reason for this may be the fact that HEAVY has more momentum than GARCH, which rapidly reverts to the mean.

This conjecture may be further emphasized by table 10: when direct estimation is employed, it becomes impossible to take any conclusions regarding the difference in forecast quality of the models.

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>t &gt; 2.02</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL</td>
<td>RV</td>
<td>−0.71</td>
<td>−0.65</td>
<td>−1.79</td>
<td>−1.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>−0.17</td>
<td>−0.68</td>
<td>−1.87</td>
<td>−1.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>−0.91</td>
<td>−0.89</td>
<td>−1.31</td>
<td>−1.68</td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>−1.26</td>
<td>−1.33</td>
<td>−0.70</td>
<td>−1.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>−1.22</td>
<td>−1.61</td>
<td>−0.05</td>
<td>−0.19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.75</td>
<td>−0.16</td>
<td>0.07</td>
<td>−1.27</td>
<td></td>
</tr>
</tbody>
</table>

Table 10 – Out-of-sample with direct estimation: HEAVY x GARCH

9.3 Int-HEAVY versus IGARCH

Since the Int-HEAVY model resumes to a standard HEAVY model when forecasting is applied to the same horizon that the model was estimated for, the direct estimation comparison between it and IGARCH will not be realized. The estimated parameters and respective log-likelihood values for the in-sample experiments are shown in table 11:

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>IGARCH</th>
<th>Int-HEAVY-r</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\alpha_{IG}$</td>
<td>$\ln \ell$</td>
</tr>
<tr>
<td>DOL</td>
<td>RV</td>
<td>0.104</td>
<td>−551.198</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.232</td>
<td>0.884</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.407</td>
<td>0.811</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>0.043</td>
<td>−862.456</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.095</td>
<td>0.899</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.098</td>
<td>0.917</td>
</tr>
</tbody>
</table>

Table 11 – Parameters and log-likelihood comparison: IGARCH x Int-HEAVY
Notice, though, that since the HEAVY-\(r\) equation (6.3) is unchanged in Int-HEAVY (6.7), the log-likelihood values are the same as in the standard case, which, in turn, continue to be higher than the respective IGARCH values. The results for both in-sample (table 12) and out-of-sample (table 13) experiments are quite similar the ones found in previous section.

### In-Sample

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL RV</td>
<td>-2.29</td>
<td>-1.79</td>
<td>-2.26</td>
<td>-0.50</td>
<td>-3.77</td>
<td>-3.99</td>
<td>-4.53</td>
<td>-4.10</td>
<td></td>
</tr>
<tr>
<td>MSR(\text{V})</td>
<td>-2.73</td>
<td>-2.47</td>
<td>-1.56</td>
<td>0.65</td>
<td>-3.92</td>
<td>-4.32</td>
<td>-4.60</td>
<td>-1.63</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-2.46</td>
<td>-2.20</td>
<td>-3.30</td>
<td>-1.60</td>
<td>-3.78</td>
<td>-4.12</td>
<td>-5.15</td>
<td>-6.47</td>
<td></td>
</tr>
<tr>
<td>IND RV</td>
<td>-1.48</td>
<td>-1.48</td>
<td>-1.15</td>
<td>-0.93</td>
<td>-2.24</td>
<td>-2.63</td>
<td>-3.17</td>
<td>-3.95</td>
<td></td>
</tr>
<tr>
<td>MSR(\text{V})</td>
<td>-1.77</td>
<td>-1.42</td>
<td>-1.04</td>
<td>-0.31</td>
<td>-2.14</td>
<td>-2.53</td>
<td>-3.12</td>
<td>-3.57</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-1.33</td>
<td>-0.95</td>
<td>-0.86</td>
<td>-1.07</td>
<td>-1.77</td>
<td>-1.99</td>
<td>-2.45</td>
<td>-3.58</td>
<td></td>
</tr>
</tbody>
</table>

Table 12 – In-sample: Int-HEAVY x IGARCH

### Out-of-Sample

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL RV</td>
<td>-1.00</td>
<td>-0.43</td>
<td>-0.44</td>
<td>-0.84</td>
<td>-1.60</td>
<td>0.29</td>
<td>0.63</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>MSR(\text{V})</td>
<td>-0.47</td>
<td>-0.39</td>
<td>-1.37</td>
<td>0.34</td>
<td>-0.72</td>
<td>-0.30</td>
<td>-0.56</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.20</td>
<td>0.38</td>
<td>-0.18</td>
<td>-0.03</td>
<td>-1.05</td>
<td>-0.07</td>
<td>0.20</td>
<td>1.12</td>
<td></td>
</tr>
<tr>
<td>IND RV</td>
<td>-1.46</td>
<td>-1.91</td>
<td>-0.81</td>
<td>-2.04</td>
<td>-2.11</td>
<td>-3.05</td>
<td>-3.35</td>
<td>-4.31</td>
<td></td>
</tr>
<tr>
<td>MSR(\text{V})</td>
<td>-2.79</td>
<td>-3.10</td>
<td>-1.93</td>
<td>-1.12</td>
<td>-3.56</td>
<td>-4.27</td>
<td>-4.38</td>
<td>-3.99</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>0.28</td>
<td>0.15</td>
<td>-0.06</td>
<td>-2.12</td>
<td>0.26</td>
<td>-0.26</td>
<td>-0.90</td>
<td>-4.06</td>
<td></td>
</tr>
</tbody>
</table>

Table 13 – Out-of-sample: Int-HEAVY x IGARCH

### 9.4 "GJR-HEAVY" versus GJR-GARCH

Moving to the models that take into account the fact that future increases in the volatility of returns may be associated with present falls in asset prices, "GJR-HEAVY" will be now compared to GJR-GARCH. Table 14 shows the estimated parameter values for each of them.
### Table 14 – Parameters and log-likelihood comparison: GJR-GARCH x 'GJR-HEAVY-r'

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>$\alpha_{GJR}$</th>
<th>$\gamma_{GJR}$</th>
<th>$\beta_{GJR}$</th>
<th>ln $\ell$</th>
<th>$\alpha_{GH}$</th>
<th>$\gamma_{GH}$</th>
<th>$\beta_{GH}$</th>
<th>ln $\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL</td>
<td>RV</td>
<td>0.094</td>
<td>0.000</td>
<td>0.894</td>
<td>-550.849</td>
<td>0.270</td>
<td>0.000</td>
<td>0.835</td>
<td>-534.850</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.232</td>
<td>0.000</td>
<td>0.883</td>
<td>-534.940</td>
<td>0.398</td>
<td>0.000</td>
<td>0.814</td>
<td>-529.766</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.398</td>
<td>0.000</td>
<td>0.814</td>
<td>-529.766</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>0.000</td>
<td>0.067</td>
<td>0.937</td>
<td>-853.995</td>
<td>0.000</td>
<td>0.206</td>
<td>0.906</td>
<td>-852.374</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>0.000</td>
<td>0.177</td>
<td>0.924</td>
<td>-853.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.000</td>
<td>0.206</td>
<td>0.926</td>
<td>-853.544</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is possible to see that these statistical leverage effects are not present in the DOL time series (at least when modeling it this way), since both $\gamma_{GJR}$ and $\gamma_{GH}$ are estimated to be zero. One can also note that not even the log-likelihood values are affected, since they are exactly the same as the ones from table 6.

In the IND time series, however, the exact opposite effect can be noticed. Both $\alpha_{GJR}$ and $\alpha_{GH}$ are estimated to be zero, and $\gamma_{GH}$ has a much higher value than $\alpha_{H}$ had - with $\gamma_{GJR}$ more than doubling in relation to $\alpha_{G}$ (see table 5).

Also in the Ibovespa case, the log-likelihood for both models substantially increased when compared to the standard models, with the difference between HEAVY and GARCH narrowing. These changes are possibly the reason behind the in-sample results for IND not following the pattern from previous findings. The experiments using 1-step ahead estimation here are the first where GARCH truly overcomes HEAVY.

### In-Sample

Table 15 shows the results for forecasts using 1-step ahead estimation, while table 16 presents the results when using direct estimation. As before, the outcomes that diverge from the others (from the same asset) are highlighted in gray.

<table>
<thead>
<tr>
<th>$t &gt; 1.96$</th>
<th>Pointwise</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>RM</td>
<td>1 2 3 5 10</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-3.04 -2.75 -2.58 -3.28</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>-0.58 0.00 -0.11 1.21 1.52</td>
</tr>
<tr>
<td></td>
<td>MSRV</td>
<td>-0.40 -0.13 -0.05 1.20</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>-0.15 0.32 0.29 1.19 1.24</td>
</tr>
</tbody>
</table>

Table 15 – In-sample: "GJR-HEAVY-r" x GJR-GARCH
Chapter 9. Experiments and Results

<table>
<thead>
<tr>
<th>$t &gt; 1.96$</th>
<th>Pointwise</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>RM</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>DOL RV</td>
<td>-3.32</td>
<td>-1.83</td>
<td>-3.12</td>
<td>-1.33</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-2.14</td>
<td>-1.69</td>
<td>-2.24</td>
<td>-0.31</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-3.41</td>
<td>-2.86</td>
<td>-2.65</td>
<td>-1.21</td>
<td></td>
</tr>
<tr>
<td>IND RV</td>
<td>-0.93</td>
<td>-0.47</td>
<td>-0.77</td>
<td>-0.91</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-0.97</td>
<td>-0.50</td>
<td>-0.06</td>
<td>-0.06</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.66</td>
<td>-0.16</td>
<td>-0.35</td>
<td>-0.48</td>
<td></td>
</tr>
</tbody>
</table>

Table 16 – In-sample with direct estimation: "GJR-HEAVY" x GJR-GARCH

Tables 15 and 16 make it possible to verify that $\gamma_{GJR}$ and $\gamma_{GH}$ are so irrelevant in the DOL case, that sometimes their numbers are equal to the ones from section 9.2.

**Out-of-Sample**

Table 17 shows the results for forecasts using 1-step ahead estimation, while table 18 presents the results when using direct estimation. Here, HEAVY backs to win the race against GARCH when considering IND.

<table>
<thead>
<tr>
<th>$t &gt; 2.02$</th>
<th>Pointwise</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Cumulative</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>RM</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>DOL RV</td>
<td>0.35</td>
<td>0.13</td>
<td>0.11</td>
<td>-0.76</td>
<td>-0.56</td>
<td>-0.74</td>
<td>0.07</td>
<td>-0.47</td>
<td>-0.42</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-0.39</td>
<td>-0.39</td>
<td>-0.40</td>
<td>-0.95</td>
<td>-0.11</td>
<td>-0.70</td>
<td>-0.22</td>
<td>-0.29</td>
<td>0.36</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.45</td>
<td>-0.20</td>
<td>0.44</td>
<td>-0.33</td>
<td>-0.12</td>
<td>-1.08</td>
<td>-0.23</td>
<td>-0.11</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>IND RV</td>
<td>-1.38</td>
<td>-1.17</td>
<td>-1.40</td>
<td>-0.93</td>
<td>-2.01</td>
<td>-1.89</td>
<td>-2.30</td>
<td>-3.41</td>
<td>-3.93</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-1.50</td>
<td>-1.42</td>
<td>-1.53</td>
<td>-1.14</td>
<td>-0.28</td>
<td>-2.14</td>
<td>-2.46</td>
<td>-2.96</td>
<td>-1.61</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.69</td>
<td>-0.67</td>
<td>-0.72</td>
<td>-0.58</td>
<td>-1.36</td>
<td>-1.06</td>
<td>-1.34</td>
<td>-1.83</td>
<td>-2.83</td>
<td></td>
</tr>
</tbody>
</table>

Table 17 – Out-of-sample: "GJR-HEAVY" x GJR-GARCH

<table>
<thead>
<tr>
<th>$t &gt; 2.02$</th>
<th>Pointwise</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>RM</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>DOL RV</td>
<td>-0.71</td>
<td>-0.65</td>
<td>-1.79</td>
<td>-1.57</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-0.17</td>
<td>-0.68</td>
<td>-1.87</td>
<td>-1.49</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.91</td>
<td>-0.89</td>
<td>-1.31</td>
<td>-1.68</td>
<td></td>
</tr>
<tr>
<td>IND RV</td>
<td>-0.86</td>
<td>-1.33</td>
<td>-0.82</td>
<td>-1.69</td>
<td></td>
</tr>
<tr>
<td>MSRV</td>
<td>-0.58</td>
<td>-0.88</td>
<td>-0.29</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>EWMA</td>
<td>-0.68</td>
<td>-0.75</td>
<td>-0.13</td>
<td>-1.57</td>
<td></td>
</tr>
</tbody>
</table>

Table 18 – Out-of-sample with direct estimation: "GJR-HEAVY" x GJR-GARCH
9.5 Ext-HEAVY versus GJR-GARCH

The last of the comparisons presented in this work has mixed results. As the reader will easily see, the estimated values for the parameters are quite similar to the ones from section 9.4, except by two things:

- $\gamma_{XH}$ are estimated to be higher than zero in all cases, causing a little positive bump in the value of U.S. Dollar log-likelihood function evaluations, meaning that semivariances do have some impact in its volatility dynamics;

- when looking at Ibovespa, this is the only comparison where HEAVY’s log-likelihoods are smaller than GARCH’s.

<table>
<thead>
<tr>
<th>Asset</th>
<th>RM</th>
<th>$\alpha_{GJR}$</th>
<th>$\gamma_{GJR}$</th>
<th>$\beta_{GJR}$</th>
<th>$\ln \ell$</th>
<th>$\alpha_{XH}$</th>
<th>$\gamma_{XH}$</th>
<th>$\beta_{XH}$</th>
<th>$\ln \ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOL</td>
<td>RV</td>
<td>0.094</td>
<td>0.000</td>
<td>0.894</td>
<td>−550.849</td>
<td>0.232</td>
<td>0.081</td>
<td>0.834</td>
<td>−534.826</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.391</td>
<td>0.023</td>
<td>0.812</td>
<td>−529.764</td>
<td>0.094</td>
<td>0.067</td>
<td>0.937</td>
<td>−853.995</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>0.000</td>
<td>0.067</td>
<td>0.937</td>
<td>−853.995</td>
<td>0.000</td>
<td>0.245</td>
<td>0.870</td>
<td>−855.376</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>0.000</td>
<td>0.196</td>
<td>0.912</td>
<td>−856.944</td>
<td>0.000</td>
<td>0.196</td>
<td>0.912</td>
<td>−856.944</td>
</tr>
</tbody>
</table>

Table 19 – Parameters and log-likelihood comparison: GJR-GARCH x Ext-HEAVY

In-Sample

In-sample results for the Ext-HEAVY versus GJR-GARCH comparison are shown in tables 20 and 21. These values are quite similar to the ones from the previous section, with the results for DOL slightly improving and the ones for IND slightly worsening (when looking at them from a HEAVY perspective).

<table>
<thead>
<tr>
<th>$t &gt; 1.96$</th>
<th>Pointwise</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>RM</td>
<td>1  2  3   5  10</td>
</tr>
<tr>
<td>DOL</td>
<td>RV</td>
<td>−2.83 −2.43 −2.22 −2.71 −1.73</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>−3.05 −2.74 −2.58 −3.28 −2.27</td>
</tr>
<tr>
<td>IND</td>
<td>RV</td>
<td>0.57 0.97 0.78 0.99 1.16</td>
</tr>
<tr>
<td></td>
<td>EWMA</td>
<td>1.06 1.33 1.21 1.29 1.30</td>
</tr>
</tbody>
</table>

Table 20 – In-sample: Ext-HEAVY x GJR-GARCH
Table 21 – In-sample with direct estimation: Ext-HEAVY x GJR-GARCH

Out-of-Sample

Out-of-sample results for the Ext-HEAVY versus GJR-GARCH comparison are presented in tables 22 and 23, and, as before, these do not change much from the findings of the other comparisons.

Table 22 – Out-of-sample: Ext-HEAVY x GJR-GARCH

Table 23 – Out-of-sample with direct estimation: Ext-HEAVY x GJR-GARCH
Conclusion

The main conclusion that this work should lead to, is that the HEAVY family of models are a good alternative to GARCH models. Even though the results with data from the Brazilian futures market were not as astonishing as the ones found by the original study, they do provide sufficient reasons to instigate further investigation.

Results such as the ones from section 9.1 show that, although not forecasting much better than GARCH models, the data may fit better with HEAVY. With the experiments that followed, there are also evidences that BM&FBovespa’s U.S. Dollar future contracts may be even better specified with an Ext-HEAVY model, and Ibovespa future contracts with "GJR-HEAVY".

Also, in the experiments involving prediction, even though most of the results were not necessarily statistically significant, they did give more advantage to HEAVY. Moreover, the use of a larger sample size could enhance the difference produced by the loss functions. In the original paper, the authors used ten years of data, with moving windows of four years, but just around a fifth of this amount of information was available for this study.

Still regarding forecasting, in the experiments where most of the results favored HEAVY, the difference was higher in the cumulative tests with bigger horizons. This can be related to the fact that HEAVY has more momentum than GARCH, and shows that HEAVY may be more suitable for longer uses, when the time between reestimations of the parameters is considerable.

Finally, one can notice that, during all tests, HEAVY consistently shows better results when using EWMA as the realized measure for DOL, and RV for IND. This is interesting to see, since both estimators are expected to be more noisy than MSRV, and even so they appear to better describe the volatility of these assets.

Future Directions

It was seen that the experiments from chapter 9 were not enough to rigorously affirm that HEAVY models are superior to GARCH models, or the other way around, at least in the studied scenario: the 2-year window of those two assets of BM&FBovespa. A higher number of assets and a larger sample size would probably suffice.

The ideal sample size for estimation could not be determined as well, and this has direct effect in the applicability of any model in a real scenario. One year and 10 months was the amount of choice in the out-of-sample experiments both because the available data did not permit to use more and because earlier tests with smaller samples showed
that the HEAVY parameters estimation was unstable: moving the window one day would cause the parameters to change drastically, while moving again could bring them back to the same level as before.

This was mostly seen with BM&FBovespa’s U.S. Dollar future contract, and, especially, while using realized variances as the realized measure. The multiscale estimator was much less sensible, with EWMA staying somewhere in the middle of the two. This may be due to the presence of intraday noises in the estimators. Again, the use of more securities in the experiments could help answer these questions.

Another point of interest would be comparing the HEAVY models against each other, to check which one is better. As with GARCH, each of them is probably better for given situations, but comparing them would be certainly enlightening.

**Further Comments**

It is the hope of the author, that this work served not only to demonstrate the capabilities of new alternatives when compared to more traditional approaches, but also that it has accomplished its goal of encouraging other researchers to explore the Brazilian market even more. He also hopes that the reading experience was pleasant and that, if this was the first contact of the reader with econometric models, he or she was able to learn the basics with this text.
Bibliography


Appendix
APPENDIX A – Database Construction

BM&FBovespa distributes its market data in .csv files, properly accompanied by description files that explain what information is contained in the files. In the data that were used, from January until August, 2012, there was one file per month, and, afterwards, one file per day. The procedure to transform these files in a useful database (having this work in mind) is explained below, divided in three major steps.

The Python scripts that implement each of these steps are available in GitHub (<https://github.com/twalves/dissertation>), under the following names:

- step 0: init_database.py;
- step 1: init_raw_tables.py;
- step 2: init_work_tables.py;
- step 3: init_work_timeseries.py.

Where step 0 refers solely to the creation of the database and auxiliary tables. If executed in this order, with the proper .csv files in hand, the same database as the one used in this work should be created. In a machine with a 2.9GHz Intel Core i7 processor, 8GB 1600 MHz DDR3 ram memory, and a 5400rpm SATA disk, it takes about 2 hours for the whole process to complete.

A MySQL\textsuperscript{1} database was utilized, but since the Python library SQLAlchemy\textsuperscript{2} was used, the scripts should work properly with any database engine. Other than these, as the code snippets that appeared before, the only required external libraries are the most known NumPy\textsuperscript{3} and pandas\textsuperscript{4}.

The procedures executed in each of the above scripts will now be briefly described.

Step 1

For all the data available for each month (even if in separate files), create a pandas’ DataFrame object with all the columns available in the original data, then filter the data to:

1. remove all the lines for which the instrument symbol represents neither DOL nor IND contracts;

\textsuperscript{1} <http://www.mysql.com>
\textsuperscript{2} <http://www.sqlalchemy.org>
\textsuperscript{3} <http://www.numpy.org>
\textsuperscript{4} <http://pandas.pydata.org>
2. remove all the lines for which the trade time is '00:00:00:000'

3. remove all the lines for which there is no 'trade indicator' number

After that, save all the filtered information in the database, to what was called the 'raw tables', one for each month. When completed, each table should have around 500,000 lines of data (≈ 50MB).

Step 2

Since there are different maturities for each of the future contracts used in this work, the maturity with higher daily volume, per day and per asset, is chosen. Two SQL scripts are used to do the 'dirty' work here, using temporary tables. Then, for the selected data:

1. the weighted average price is calculated when there is more than one trade with exactly the same timestamp, where the volume from each trade is used as the weight;

2. the log prices are computed;

3. the 'instrument symbol' column is divided in three columns: 'asset', 'month', 'year'.

The resulted table is saved in the database as a 'work table', but now only with the columns 'session date', 'trade time', 'asset', 'month', 'year', 'trade price', and 'log trade price'. Again, a table is created for each month, and now each of them has around 250,000 lines of data (≈ 20MB).

Step 3

Only two more tables are created in this step: one for DOL and one for IND. Each of them containing one line per day, with all the required measures to estimate and evaluate the models presented in this work, using the algorithms suggested in chapter 5. During this step, four days were ignored, due to different reasons:

- 2012-02-22 and 2013-02-13, which are half-day holidays in Brazil that occur every Wednesday after Carnival, when trading begins only after 1pm;
- 2012-02-01, which, for some unknown reason, had no data from 1h02pm until 1h43pm;
- 2012-03-12, which also did not present any values until 9h40am.
Annex
ANNEX A – Code Listings

The following source codes constitute a portion of what was produced during the development of this work. These are the ones that received some kind of reference during the text:

- `loglikelihood.py`: log-likelihood functions for both Gaussian distribution and Student’s t-distribution;
- `heavy.py`: HEAVY-r and HEAVY-RM functions for log-likelihood evaluation, estimation and forecasting;
- `garch.py`: GARCH functions for log-likelihood evaluation, estimation and forecasting.

The implementation of all other models used in this work are available in GitHub (<https://github.com/twalves/dissertation>), along with the database creation scripts mentioned in appendix A. The code listed here is also available online, in its properly commented version. Here, comments are ommited in order to keep the length as small as possible.

Most functions are direct translations from the respective mathematical formulae, being quite easy to read and understand. The estimation methods, though, use some assumptions that need to be clarified:

- The value for both $h_0$ and $\sigma_0^2$ is set to be the variance of the $\sqrt{T}$ initial values of the available time series of returns, where $T$ is the size of the sample;
- The value for $\mu_0$ is set to be the average of the $\sqrt{T}$ initial values of the available time series of realized measures, where $T$ is the size of the sample;
- The initial guess for the parameter values, which is an argument of the optimizing function, is built using a brute force algorithm: a grid around the usual values of each parameter is constructed.

Apart from the Python libraries already listed in appendix A, the only different library that appears in the listing below is SciPy\(^1\).

---

\(^1\) <http://scipy.org>
loglikelihood.py:

```python
from numpy.core.umath import log, pi
import scipy.special

def gaussian_loglikelihood(squared_returns, squared_volatilities,
                          sign=1.0):
    terms = -0.5 * (log(2.0 * pi) + log(squared_volatilities)
                    + squared_returns / squared_volatilities)

    result = sign * sum(terms)

    return result

def tstudent_loglikelihood(squared_returns, squared_volatilities, df,
                           sign=1.0):
    terms = -0.5 * (log(df * scipy.special.beta(0.5, 0.5 * df) ** 2)
                     + log(squared_volatilities)
                     + log(1.0 + (squared_returns /
                                (squared_volatilities * df)))
                     * (df + 1.0))

    result = sign * sum(terms)

    return result

def generic_loglikelihood(squared_returns, squared_volatilities, df=0.0,
                          sign=1.0):
    if df <= 0.0:
        return gaussian_loglikelihood(squared_returns,
                                       squared_volatilities, sign)
    else:
        return tstudent_loglikelihood(squared_returns,
                                       squared_volatilities, df, sign)
```
import loglikelihood
import numpy
import time
import scipy.optimize

def heavyr(omega, alpha, beta, rm_last, h_last):
    h = omega + alpha * rm_last + beta * h_last
    return h

def heavyrm(omega, alpha, beta, rm_last, mu_last):
    mu = omega + alpha * rm_last + beta * mu_last
    return mu

def heavy_forecast(h_parameters, mu_parameters, rm_past, h_past, mu_past,
    steps=1, full_output=False):

    lag = len(rm_past)
    h = numpy.zeros(steps)
    mu = numpy.zeros(steps)

    for i in range(0, steps):
        if i < lag:
            h[i] = heavyr(h_parameters[0], h_parameters[1],
            h_parameters[2], rm_past[i], h_past[i])
            mu[i] = heavyrm(mu_parameters[0], mu_parameters[1],
            mu_parameters[2], rm_past[i], mu_past[i])
        else:
            h[i] = heavyr(h_parameters[0], h_parameters[1],
            h_parameters[2], mu[i - lag], h[i - lag])
            mu[i] = heavyrm(mu_parameters[0], mu_parameters[1],
            mu_parameters[2], mu[i - lag], mu[i - lag])

    if full_output:
        return numpy.copy(h), numpy.copy(mu)
    else:
        return numpy.copy(h)
def heavyr_likelihood(parameters, r2_data, rm_data, h0, lag=1, tstudent_df=0.0, estimating=True):
    t = len(r2_data)
    h = numpy.repeat(h0, t)
    for i in range(lag, t):
        h[i] = heavyr(parameters[0], parameters[1], parameters[2], rm_data[i - lag], h[i - lag])
    if estimating:
        return loglikelihood.generic_loglikelihood(r2_data, h, df=tstudent_df, sign=-1.0)
    else:
        return loglikelihood.generic_loglikelihood(r2_data, h, df=tstudent_df, sign=1.0), numpy.copy(h)

def heavyrm_likelihood(parameters, rm_data, mu0, lag=1, tstudent_df=0.0, estimating=True):
    t = len(rm_data)
    mu = numpy.repeat(mu0, t)
    for i in range(lag, t):
        mu[i] = heavyrm(parameters[0], parameters[1], parameters[2], rm_data[i - lag], mu[i - lag])
    if estimating:
        return loglikelihood.generic_loglikelihood(rm_data, mu, df=tstudent_df, sign=-1.0)
    else:
        return loglikelihood.generic_loglikelihood(rm_data, mu, df=tstudent_df, sign=1.0), numpy.copy(mu)
def heavyr_estimate(r2_data, rm_data, lag=1, tstudent_df=0.0, full_output=True):

    before = time.time()

    finfo = numpy.finfo(numpy.float32)

    h0 = 0.0
    t = int(numpy.sqrt(len(r2_data)))
    for i in range(t):
        h0 += r2_data[i]
    h0 /= t

    args = (r2_data, rm_data, h0, lag, tstudent_df)

    ranges = ((finfo.eps, 0.5), (finfo.eps, 0.5), (0.5, 1.0))
    x0 = scipy.optimize.brute(heavyr_likelihood, ranges, args=args, finish=None)

    # Constraint that beta < 1.0
    constraints = ({'type': 'ineq', 'fun': lambda x: 1.0 - x[2]})

    bounds = ((finfo.eps, None), (finfo.eps, None), (finfo.eps, None))
    result = scipy.optimize.minimize(heavyr_likelihood, x0, args=args, method='SLSQP', bounds=bounds, constraints=constraints)

    estimates = result.x

    after = time.time()
    print('HEAVY−R estimation took', (after - before), 'seconds.', end='

    if full_output:
        return estimates, h0
    else:
        return estimates
def heavyrm_estimate(rm_data, lag=1, tstudent_df=0.0, full_output=True):

    before = time.time()

    finfo = numpy.finfo(numpy.float32)

    mu0 = 0.0
    t = int(numpy.sqrt(len(rm_data)))
    for i in range(t):
        mu0 += rm_data[i]
    mu0 /= t

    args = (rm_data, mu0, lag, tstudent_df)

    ranges = ((finfo.eps, 0.5), (finfo.eps, 0.5), (0.5, 1.0))
    x0 = scipy.optimize.brute(heavyrm_likelihood, ranges, args=args, finish=None)

    # Constraint that \( \alpha + \beta < 1.0 \)
    constraints = ([{'type': 'ineq', 'fun': lambda x: 1.0 - x[1] - x[2]})

    bounds = ((finfo.eps, None), (finfo.eps, None), (finfo.eps, None))
    result = scipy.optimize.minimize(heavyrm_likelihood, x0, args=args, method='SLSQP', bounds=bounds, constraints=constraints)

    estimates = result.x

    after = time.time()
    print('HEAVY-RM estimation took', (after - before), 'seconds.', end='

    if full_output:
        return estimates, mu0
    else:
        return estimates
ANNEX A. Code Listings

**garch.py**:

```python
import loglikelihood
import numpy
import time
import scipy.optimize

def garch(omega, alpha, beta, r2_last, sigma2_last):
    sigma2 = omega + alpha * r2_last + beta * sigma2_last
    return sigma2

def garch_forecast(parameters, r2_past, sigma2_past, steps=1):
    lag = len(r2_past)
    sigma2 = numpy.zeros(steps)
    for i in range(0, steps):
        if i < lag:
            sigma2[i] = garch(parameters[0], parameters[1], parameters[2],
                               r2_past[i], sigma2_past[i])
        else:
            sigma2[i] = garch(parameters[0], parameters[1], parameters[2],
                               sigma2[i - lag], sigma2[i - lag])
    return numpy.copy(sigma2)

def garch_likelihood(parameters, r2_data, sigma20, lag=1, tstudent_df=0.0, estimating=True):
    t = len(r2_data)
    sigma2 = numpy.repeat(sigma20, t)
    for i in range(lag, t):
        sigma2[i] = garch(parameters[0], parameters[1], parameters[2],
                           r2_data[i - lag], sigma2[i - lag])
    if estimating:
        return loglikelihood.generic_loglikelihood(r2_data, sigma2,
                                                    df=tstudent_df, sign=-1.0)
    else:
        return loglikelihood.generic_loglikelihood(r2_data, sigma2,
                                                    df=tstudent_df, sign=1.0),
        numpy.copy(sigma2)
```

---

These are Python functions that implement the GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model. The GARCH model is used to model time series data where volatility clustering is observed. The functions `garch`, `garch_forecast`, and `garch_likelihood` are part of the `garch.py` module, which contains the following:

- **garch**: A function that calculates the conditional variance (volatility) of the GARCH model. It takes parameters `omega`, `alpha`, `beta`, the last observed squared return `r2_last`, and the last observed conditional variance `sigma2_last`.
- **garch_forecast**: A function that forecasts the conditional variance for `steps` periods ahead. It uses the last `lag` observed squared returns and conditional variances to make predictions.
- **garch_likelihood**: A function that calculates the likelihood of the observed squared returns given the estimated parameters of the GARCH model. It computes the loglikelihood for both estimating and non-estimating cases.

The `loglikelihood` module contains a generic loglikelihood function which is used in the `garch_likelihood` function to calculate the likelihood of the observed data given the parameters.
def garch_estimate(r2_data, lag=1, tstudent_df=0.0, full_output=True):

    before = time.time()

    finfo = numpy.finfo(numpy.float32)

    sigma20 = 0.0
    t = int(numpy.sqrt(len(r2_data)))
    for i in range(t):
        sigma20 += r2_data[i]
    sigma20 /= t

    args = (r2_data, sigma20, lag, tstudent_df)

    ranges = ((finfo.eps, 0.5), (finfo.eps, 0.5), (0.5, 1.0))
    x0 = scipy.optimize.brute(garch_likelihood, ranges, args=args,
                               finish=None)

    constraints = ({'type': 'ineq', 'fun': lambda x: 1.0 - x[1] - x[2]})
    bounds = ((finfo.eps, None), (finfo.eps, None), (finfo.eps, None))
    result = scipy.optimize.minimize(garch_likelihood, x0, args=args, method='SLSQP', bounds=bounds, constraints=constraints)

    estimates = result.x

    after = time.time()
    print('GARCH estimation took', (after - before), 'seconds.', end='

    if full_output:
        return estimates, sigma20
    else:
        return estimates