MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research

Title**:**

COMPARATIVE ANALYSIS BETWEEN STANDARD RISK MEASUREMENTS AND BEHAVIORAL SIMULATIONS

Author: **Francisco Urbano García**

Advisor: **Pilar Muñoz Gracia**

Department: **Statistics and Operation Research**

University**: Universitat Politècnica de Catalunya**

Date: January 2012

UNIVERSITAT DE BARCELONA

Table of Contents

1 INTRODUCTION

This project explores behavioral driven simulations as an alternative to the existing classical methods to calculate the most common risk measurements for financial time series, that is, VaR (value at risk) and Expected Shortfalls.

1.1 Technology and Procedures

To perform verifications, analysis of data quality, graphics and general statistical calculations the statistical packages **R**[1](v.2.14.1), and **TETRAD**[2](v.4.3) were used, while for the simulation software **Ruby**[3](v.1.9.3) was chosen as programming language. Ruby was chosen for several important reasons:

- **Object Oriented.** Ruby is an entirely object-oriented language which facilitates the design of software as well as the maintenance and development of a **UML**[4] documentation to describe the project as it grows.
- **Interpreted language**. Languages like Ruby, Perl or Python allow a rapid and functional development. Since eventually a final stage is to recode the project into a lower level language such as C or Assembler for efficiency reasons, interpreted languages provide an experimental platform that reduces development time and offers more guarantees for their operational structure before the recoding is done.
- **Continuity**. The project might eventually face more ambitious goals and functionality for which high-level programming and general purpose languages are ideal. In complex scenarios general purpose languages handle more efficiently a wider range of structures than specialized languages such as R, Matlab ... etc.

For the simulator it has been develop a number of random number generators the quality of which is guaranteed by the **DIEHARD**[5] and the **DIEHARDER**[6] packages which includes other new tests based on the development of the state of the art on the quality of a Random Number Generator (RNG).

1 http://cran.r-project.org/
2 http://www.phil.cmu.edu/projects/tetrad/
3 http://www.ruby-lang.org/en/
4 http://www.uml.org/
5 http://stat.fsu.edu/pub/diehard/
6 http://www.phy.duke.edu/~rgb/General/dieharder.php

1.2 Financial Time Series

In finance the ideal investment model would be the one that predicts exactly the price of financial time series. If we had such model it would seem to imply that all of us could be rich in no time but, is that model even possible for it to exist? Actually it is, let's imagine for a moment a market where all the investors use exactly the same deterministic model to decide the price of a stock for the next day, if that model would give a results, let's say, of 5% increase in the price, all stock holders will believe that is actually the right price because, after all, their method is perfect, right? And because everyone will sell and buy at that 5% increase, and the price is set based on a bidding process, we will have a self-fulfilled prophecy where the investors make the model perfect. In fact, **any model will give a perfect forecast of the price as long as the all investor believe it does**.

Although the model is perfect, yet, we cannot become rich overnight because everyone knows the same information at the same time. In order to become millionaires we need something else than a perfect model; we need either the perfect model to be shared just by a few, after all, the only way to be richer in the stock market is if someone gets poorer, or have the information before everybody else.

There is no way to handle information before anyone else does, at least legally, but once everyone has the information those acting faster behave in effect as if they had the information before than anyone else, and that is perfectly legal. As a matter of fact, investment firms spend hundreds of millions in computerized systems to be able to send buy or sell orders before anyone else. This make them earn money by being faster, but how about smarter?

Since the stock market behaves over ally in a way that for someone to gain someone else has to lose, we can conclude that there exists no perfect mathematical model which is both widely publicized and useful to gain money, or expressed in a mathematical form where Δ indicates the increase in the parameter.:

$$\Delta Publicity \cdot \Delta Gain = Constant$$

A perfect model that is not known by everyone cannot be self-fulfilling; let's imagine for a second that such a perfect forecast model exists, since the price in the market is set by a bidding process the existence of such model would imply that it accounts for the results of such bidding, so this perfect model is simply the outcome of a simulation of this bidding process with all the possible bidding strategies of every possible investor.

In fact this is not difficult to calculate if we knew what bidding strategies are into play but which we can only guess, and even if we guessed right, the moment the investors without this happy insight start losing money they would consider their strategies at fault and they would change them, thus, making in turn our perfect model useless.

We nonetheless could try to account for the new models that investors will use next, thus making our model still perfect, but then we can only expect that eventually they will try to do the same guessing process and they will try to simulate what outcome of our simulation is, once we reach this point the “perfect” model is made public and the market is reduced to a guessing game.

In effect, this process is very similar to the rock-paper-scissors game; we try to figure out the pattern the other player is following, if any, but if we guess his pattern the player will change it after losing three of four times in a row, making thus our insight less than useless because now is the other player the one that knows our pattern.

The end result of this guessing game is that **financial series behave in a random walk fashion way with no real chances for investors to consistently know what really is going to happen next.**

But let's see in practice how results the intent to fit an Auto Regressive Integrated Moving Average model (ARIMA) in financial series, and to show this we will use data from the Standard & Poor's 500 which is a free-float capitalization-weighted index published since 1957 of the prices of 500 large-cap common stocks actively traded in the United States. This is how the series looks like from January 3, 1950 until November 13, 2011.

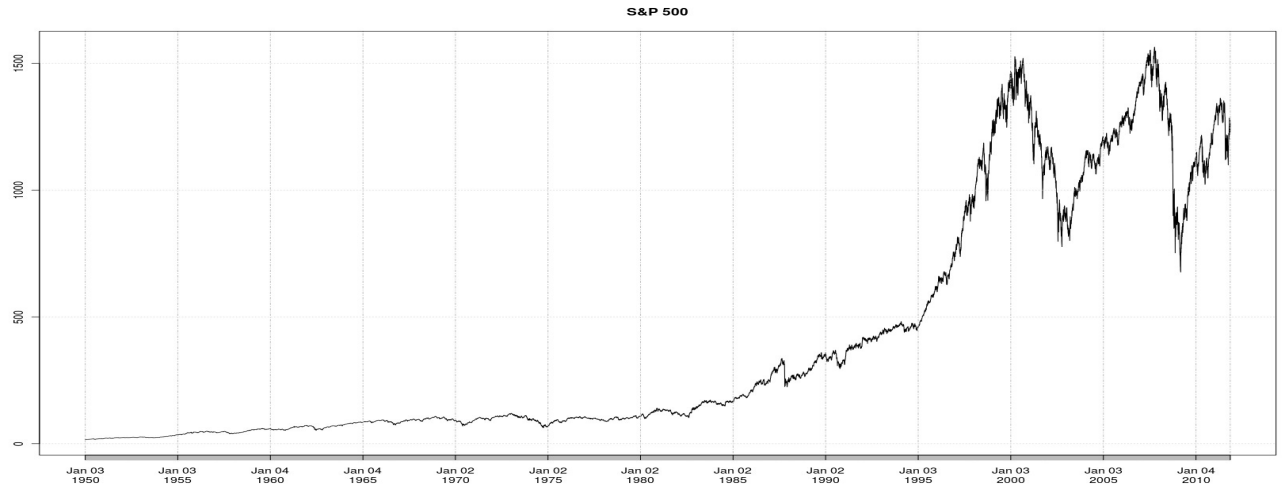


Figure 1.1: S&P 500 Time Series

If we calculate the Autocorrelated Function (ACF) and Partial Autocorrelated Function (PACF) for this time series we obtain the following:

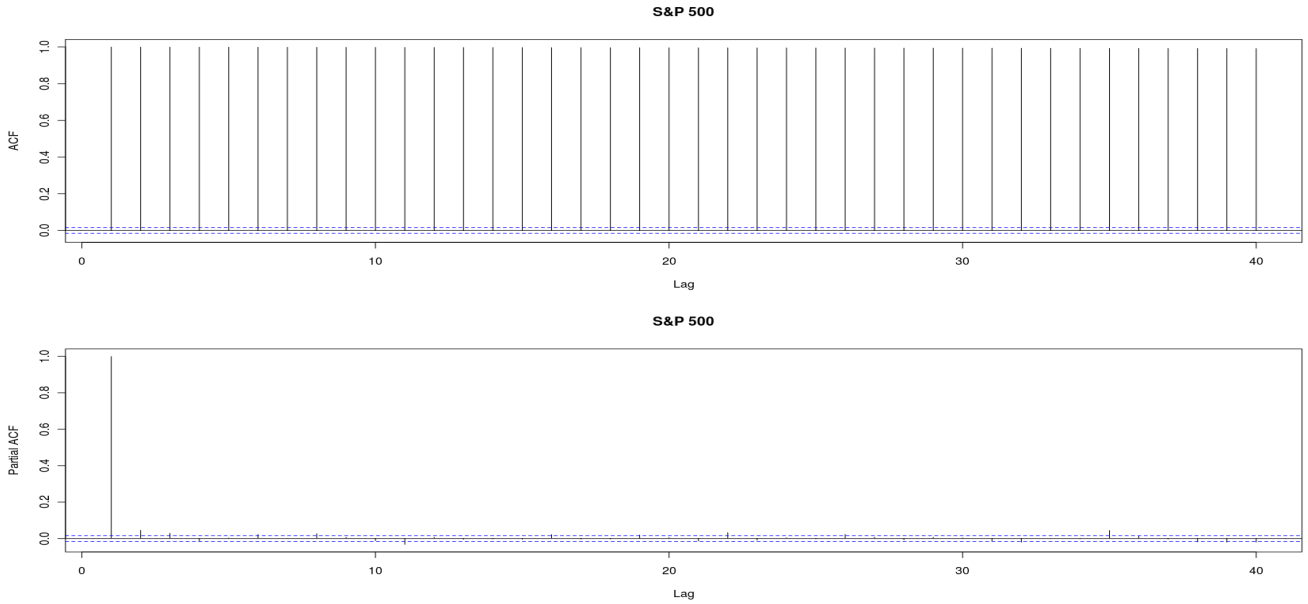


Figure 1.2: ACF and PACF for the S&P 500

The ACF and PACF indicates as a distinct possibility an Autoregressive Model of order one AR(1) or two AR(2), nonetheless the slow decline in the ACF shows that possibly we are dealing with an unit root[7] which implies this process is non-stationary and needs to be transformed prior to adjust any AR model, we can use a Dickey-Fuller test to check if this is the case:

Augmented Dickey-Fuller Test
Test Results:
PARAMETER:
Lag Order: 1
STATISTIC:
Dickey-Fuller: 0.9015
P VALUE:
0.9015

Table 1.1: Dickey-Fuller unit root test for S&P 500

7 A linear stochastic process has a unit root if one is a root of the process's characteristic equation

As we can see, the p-value is way above the 0.05 significance and therefore we cannot reject the null hypothesis that there is no unit root. Now we can apply logarithms to correct the heterocedasticity associated with financial series and differentiate the series to account for the unit root to obtain the log returns (which besides approximates the real returns values), we can also multiply by 100 the results to obtain the percent values. This is the time series we obtain after these transformations which from now on we will refer simply as *returns*:

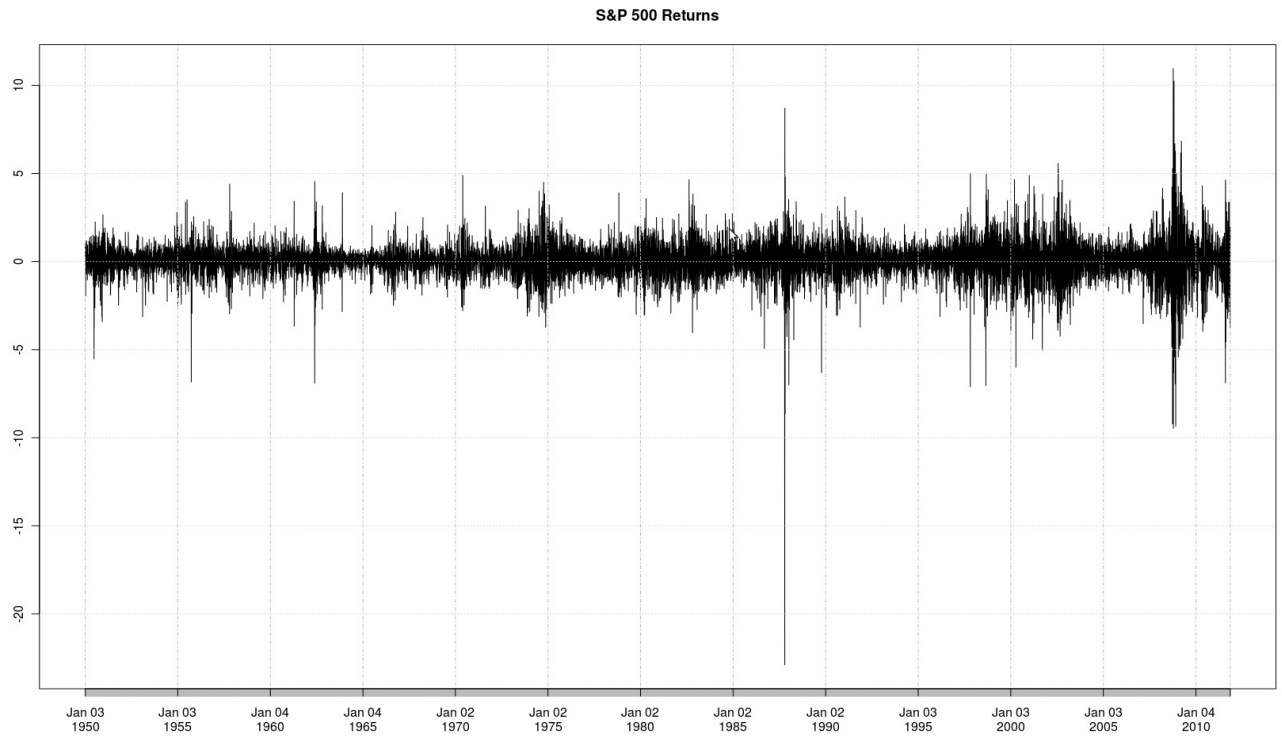


Figure 1.3: Returns for S&P 500

We can observe that the variance is not constant overtime, that is a problem for ARIMA models because for them to make a good fitting stationarity is required and this time series shows clearly that variance is not constant due to clusters of volatility throughout time and there is no simple transformation that will turn it constant.

Another requisite for ARIMA models to offer a good fitting is that the residuals must follow a Gaussian distribution which is not the case as we can see in the QQ-plot shown in Figure 1.4 where the residuals show very heavy tails and the Jarque-Bera Normality Test confirms the non normality behavior of the residuals with an asymptotic p-value: < 2.2e-16.

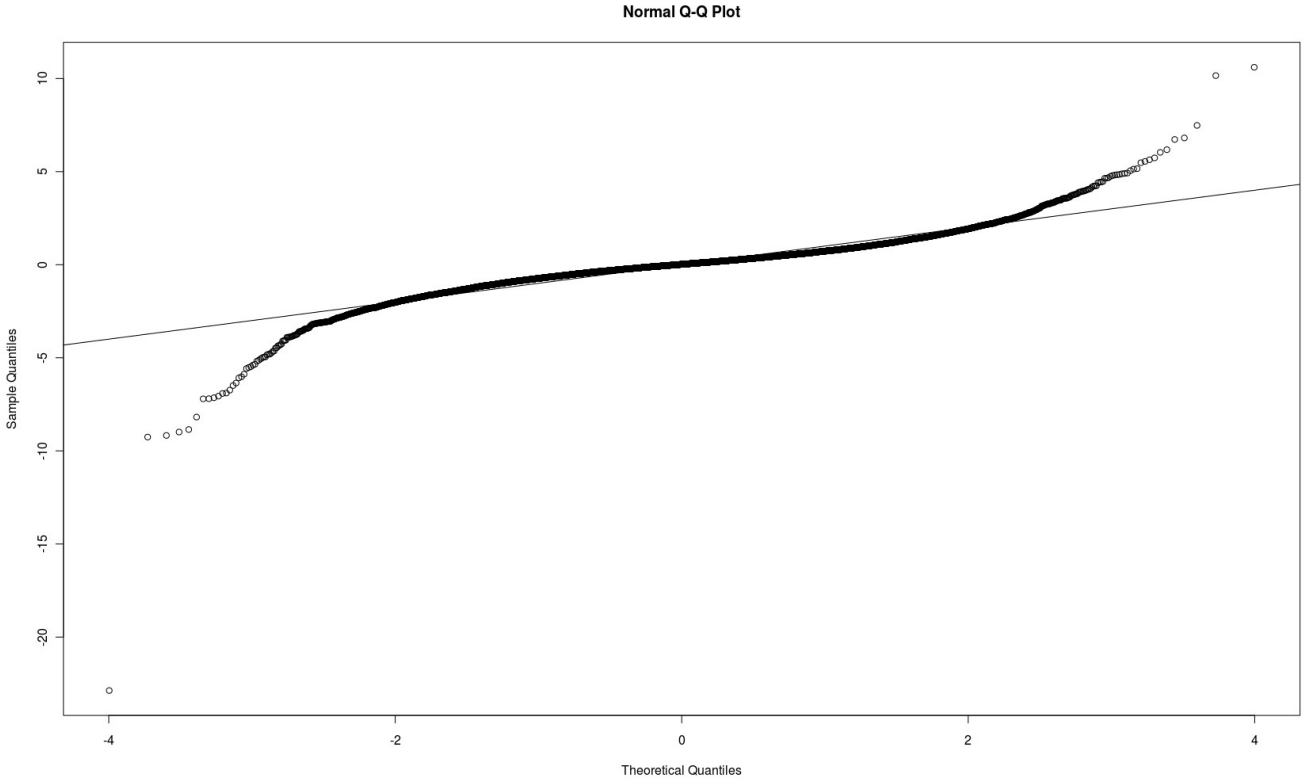


Figure 1.4: QQ-Plot of the residuals for the AR(2) model fitted for the S&P 500 returns

Even if we ignore these two requirements for obtaining a meaningful model and we fit an AR(2) model we would obtain the following statistical significant values with standard errors of 0.008 for both autoregressive parameters:

$$\epsilon_t \sim N(0, \sigma^2 = 0.9593)$$
$$r_t = 0.0319 \cdot r_{t-1} - 0.0444 \cdot r_{t-2} + \epsilon_t$$

Given the much higher magnitude of the variance of the residuals compared to the parameters in the auto-regressive model, this model will basically behave like an standard normal distribution, making all the study useless to predict if the next movement of the time series is going to be up or down.

So far it seems like a dead end where only good guessing and intuition is going to lead investment in the stock market. Financial times series show two distinct features though: Non constant variance in the returns and heavy tails in its residuals distribution. But there is yet a third property that will give us some hope; **the squared values of the residuals are highly auto-correlated.**

1.3 Volatility

In finance, volatility measures variation of price of a financial time series over time. **Volatility is basically the standard deviation of the returns residuals**.

We have seen that there are many reason not to be optimistic when it comes to predicting the price of a financial time series and also that the returns of this kind of series show three main features:

1. They show clusters of high and low volatility.
2. The residuals distribution for ARIMA models show a non normality with heavy tails.
3. The squared values of the residuals are highly auto-correlated (see Figure 1.5)

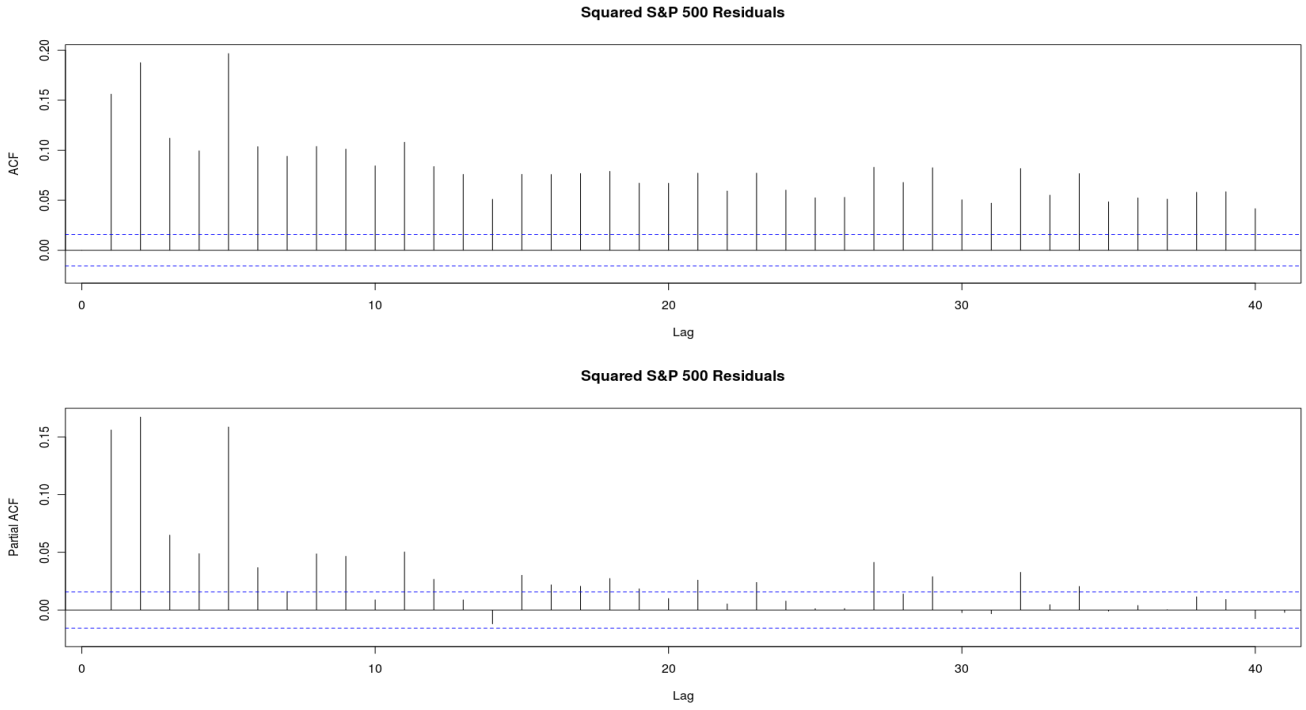


Figure 1.5: Squared S&P 500 returns

So though we do not have useful correlations for the first order magnitudes of the returns it turns out that we have high correlations for the second order, that is, the variance/volatility. Of course, even if we could predict exactly what the volatility is going to be, that still tells us nothing about whether the price is going up or down, but **it will allow us to do risk measurements** about a portfolio which is what forecasting volatility is all about as we will discuss in the next chapter.

Let's next talk about risk and different ways to analyze volatility to obtain sensible risk measurements.

2 RISK

Modeling the volatility does not tells us generally anything about whether the time series is going to increase or decrease its value, but it does not mean these modeling techniques are useless since they allow us to manage the risk we take in our investments; investing in high volatility markets would give us a higher risk to lose money but also better prospects of high benefits, on the contrary, low volatility markets would give us less benefits but also less chances to incurre in heavy loses.

Every time we take a decision we are taking chances, sometimes it is easy to measure the risk when we have a precise measure of the phenomenon in question, but often this is not the case and decisions have to be taken based on professional knowledge besides statistical measurements.

There are many techniques to handle risk in a timely fashion that integrates expert opinions and statistical calculations, for example, **Delphi**[8] is a structured communication technique, originally developed as a systematic, interactive forecasting method which relies on a panel of experts. In its standard version, the experts answer questionnaires in two or more rounds.

After each round, a facilitator provides an anonymous summary of the experts' forecasts from the previous round as well as the reasons they provided for their judgments. Thus, experts are encouraged to revise their earlier answers in light of the replies of other members of their panel.

It is believed that during this process the range of the answers will decrease and the group will converge towards the "correct" answer.

Finally, the process is stopped after a predefined stop criterion (e.g. number of rounds, achievement of consensus, stability of results) and the mean or median scores of the final rounds determine the results.

Next we can see a diagram depicting the Delphi process:

8 Harold A. Linstone, Murray Turoff (1975)

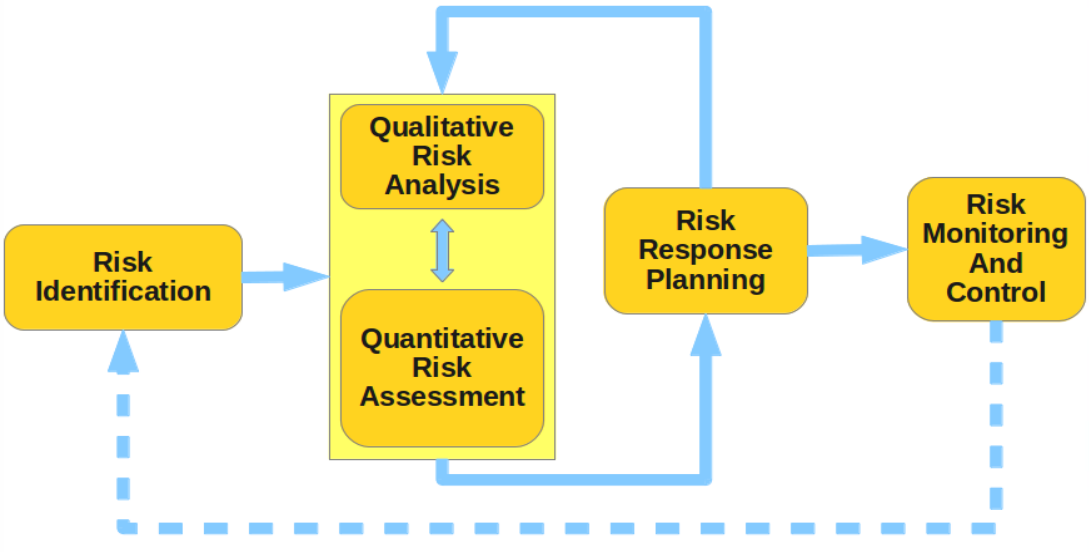


Figure 2.1: Delphi Risk management Decision process

But whatever the method we use for handling risk in financial markets we will always have to make a quantitative assessment of the risk, that assessment will have to answer at least three questions:

1. What are the chances to lose money.
2. How much money we can lose.
3. In what period of time is this risk considered.

In other words, when it comes to risk we are taking for the money we are investing we are highly interested in knowing above all what is the **value at risk**.

Classical techniques analyze the behavior of the volatility in a financial times series in order to calculate risk measurements, these methods pay no attention at why the volatility occurs and simply focus on fitting the volatility to models that emulate their returns statistical features.

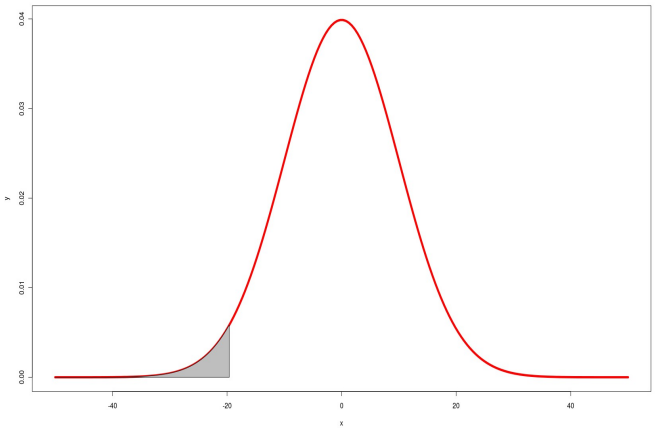
The simulation in this project will attempt to understand what causes volatility in order to further refine risk analysis. Since some data required for the simulation is only available aggregated by weeks all analyses for now on will be done on the weekly aggregated S&P 500 time series from January 3, 2004 until November 13, 2011.

2.1 Value at Risk (VaR) and Expected Shortfall

The VaR is a widely used risk measure of the risk of loss on a specific portfolio of financial assets. It basically accounts for what is the minimum amount of money that might be lost in a given period for a given probability. The Expected Shortfall risk measure complements the VaR by accounting for the money that will be lost on average in the same circumstances.

There are many classical ways to measure such quantities, among those methods the most popular are:

- Econometric Modeling[9]
- Empirical Quantile
- Extreme Value Theory[10] (EVT)
- Peaks Over Threshold[11] (POT)



Before it is described how the simulation in this project will tackle this problem let's first see how these methods behave when measuring the VaR for a period of one year (that is 2012) in the S&P 500 index.

2.2 Econometric Modeling

2.2.1 Volatility Models

In order to account for the correlations we found in the returns we need to immediately discard ARIMA models since the squared values of a random non Gaussian distribution behaves nothing like a Normal distribution, besides there are some characteristics in the volatility financial time series that need to be tailor-made addressed.

9 Tsay R.S. (2005). pag 342.
10 Gumbel, E.J. (1935)
11 Tsay R.S. (2005). pag 359.

ARCH[12] stands for Auto-regressive Conditional Heterocedasticity, and it is a model which describes statistically the basic behavior that can be observed in the returns of financial time series. The first feature of returns time series is its lack of auto-correlation for the its first order values, in other words, we can not predict its value.

This can be expressed with a multiplicative model:

$$a_t = \sigma_t \cdot \epsilon_t$$

where a_t is the residuals of the returns also named *impact*, σ_t is the the latent variable we are trying to estimate, that is, the volatility and ϵ_t is a distribution like t-Student or the Standard Normal distribution. The second feature shown int the time series returns are auto-correlations in the second order values, and the simplest way to express this is with a linear model:

$$\sigma_t^2 = \omega + \sum_{i=1}^{s} \alpha_i \cdot a_{t-i}^2$$

This model is an ARCH model, but there is a third feature that can be observed in the returns of Figure 1.3; whatever the volatility is, it seems it tends to persist, meaning that if we have high volatility the most likely scenario for the next step is high volatility too, and the same goes when we have low volatility. This persistence feature can be expressed mathematically estimating a new set of parameters as follow:

$$\sigma_t^2 = \omega + \sum_{i=1}^{s} \alpha_i \cdot a_{t-i}^2 + \sum_{j=1}^{m} \beta_i \cdot \sigma_{t-j}^2$$

And this expression defines a Generalized ARCH model or GARCH[13]. The ARCH family keeps growing when we consider more features of the returns series like its asymmetry. Asymmetry can be tackle with models like the **APARCH**[14] which comprises the ARCH and GARCH models and there are many other statistical techniques to deal with returns, and among those we can find **TAR**[15] models, **Switching Markov Chains**[16] models, and **Neural Networks**[17].

12 Engle, R. F. (1982).
13 Bollerslev, Tim (1986).
14 Würtz, D. and Chalabi, Y. and Luksan (2006)
15 Chan, K. S. and Tsay, R. S. (1998).
16 Lux, Thomas (2008).
17 Cheng, B. and Titterington, D. M. (1994)

2.2.2 RiskMetrics

This method was first established in 1989, when Sir Dennis Weatherstone, the new chairman of J.P. Morgan, asked for a daily report measuring for explaining the risks of his firm. Nearly four years later in 1992, J.P. Morgan launched the RiskMetrics methodology to the marketplace, making the substantive research and analysis that satisfied Sir Dennis Weatherstone's request freely available to all market participants.

This method is the fastest and simplest of the econometric methods to calculate the VaR and the expected shortfall for **very short positions**. Since it is very efficient, this method is suitable for daily analysis of a large number of time series for short periods.

The method assumes a conditional normal distribution for the next period returns and the volatility follows an **IGARCH**[18] model, a simplify version of a GARCH model where α and β sums exactly one. This means, among other things, that the volatility will grow to infinity as the period for analysis does.

If we fit the IGARCH model for the S&P 500 from 2004 to 2011 we obtain the following results:

$$a_t = \sigma_t \epsilon_t$$
$$\sigma_t^2 = (1 - 0.89) a_{t-1}^2 + 0.89\, \sigma_{t-1}^2$$

which in this case results in the following parameters estimations:

$$a_{t-1} = -0.9596079$$
$$\hat{\sigma_{t-1}} = 2.492783$$
$$\hat{\sigma_t}^2 = 5.606962$$

The VaR for k number of periods is then calculated as follows where *qnorm* is the quantile normal function:

$$VaR[k] = qnorm(p) \cdot \sigma_{t+1} \cdot \sqrt{k}$$

We can see that VaR in this case depends heavily in the value of the previous volatility since we have a IGARCH model with a high persistence parameter. The period considered is determined by the parameter k which, if going to infinity, it will take the VaR to infinity as well, this is the reason why this method cannot be used for long positions. If we now calculate the VaR and the expected shortfall for the next year period of 2012 we obtain the following results:

18 Mikosch, T. and Starica, C. (2004)

P	VaR[52] %	ES[52] %
1%	39.72 %	45.51 %
5%	28.09 %	35.22 %
10%	21.88 %	29.97 %

Table 2.1 Risk Measurements for RiskMetrics

Which we can read as that in a long position of one year fro 2012 we have a 1% chance to lose at least 39.71% of our investment and, on average, 45.51%. We read it similarly for and similarly for 5% and 10%. We can appreciate very large values for the VaR and the ES, this is expected since RiksMetrics is not a good method for evaluating long positions since the risk tends to infinity as we evaluate longer and longer positions.

The Econometric Modeling methodology described by Tsay[19] works exactly like this one but doing a full ARIMA and GARCH analysis of the time series making it more statistically sound for short positions but also a lot more complex to calculate.

2.3 Empirical Quantile

This method is the simplest and more straight forward of all, it calculates the VaR and the expected shortfall based on the empirical distribution of the data we have from the time series. That makes this technique suitable for long position analysis but poor for short ones, on the positive side though empirical quantile accounts for non Gaussian behaviors. Let's now calculate it for the S&P 500 time series:

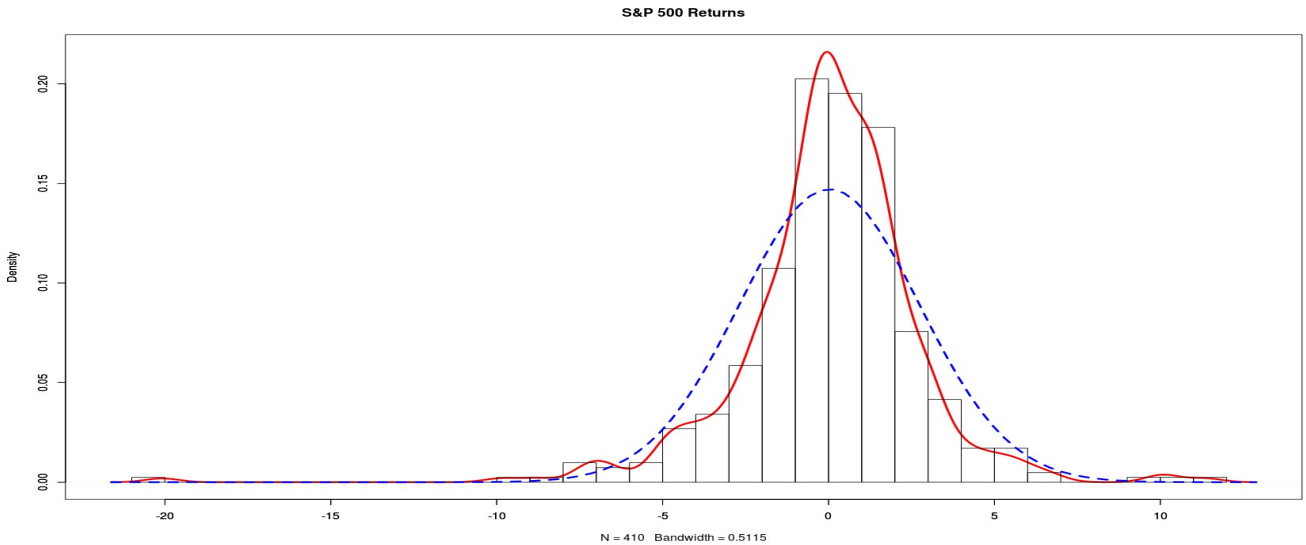


Figure 2.2: S&P 500 Returns Historgram from 2004 until 2011

19 Tsay R.S. (2005).

The blue-dotted line shows what would be the expected Gaussian behavior for the S&P 500 time series, the red line is the actual empirical behavior. As we can see they behave very differently, specially in the negative side of the distribution which is the one we are interested in when calculating the VaR and the Expected Shortfall.

One useful feature for the VaR calculations is that we can keep working with the returns time series since VaR = Value x VaR(Returns), So for now on we will work with the VaR(Returns). if we now calculate the value at risk an the expected shortfall for this distribution we obtain the following results:

P	VaR %	ES %
1%	7.28 %	10.69 %
5%	4.53 %	6.72 %
10%	2.84 %	5.21 %

Table 2.2Risk Measurements for Empirical Quantile

Which means that in a long position (in this case years) we have a 1% chance to lose at least 7.28% of our investment and, on average, 10.69%, and similarly for the 5% and 10%.

2.4 Extreme Value Theory (EVT)

Extreme value theory is a branch of statistics dealing with the extreme deviations from the median of probability distributions. The general theory sets out to assess the type of probability distributions generated by processes. Extreme value theory is important for assessing risk for highly unusual events, such as 100-year floods but it can also been used effectively to calculate VaR.

The theory says that when the period chosen goes to infinity the distribution for the maximum value in that period converge to one of the EVT distribution families. This makes this method not suitable for very short positions because they don't fit accurately to the theory.

Depending on the value for the parameter ξ in the Extremae Value Distribution we have three different families of distributions: Frechet, Gumbel and Weibull.

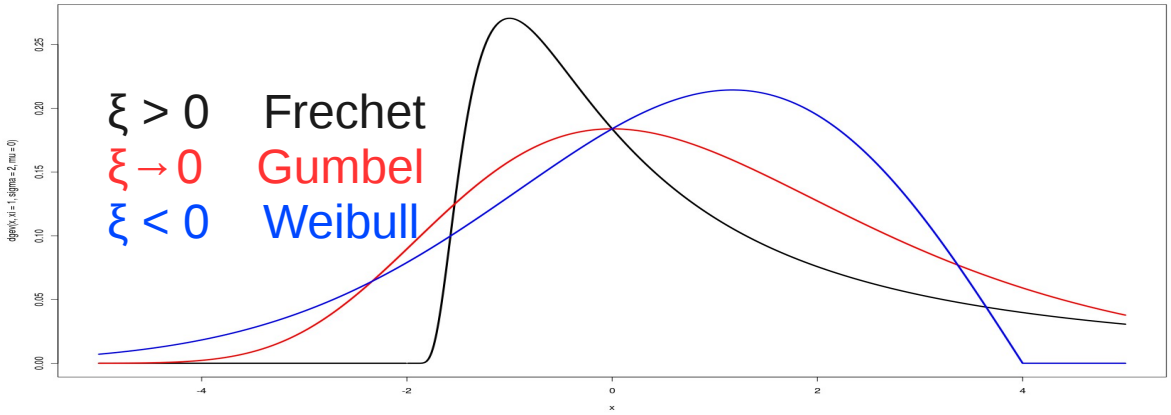


Figure 2.3: Extreme Value Distribution Family

When fitting this distribution for financial time series we will usually obtain a Frechet distribution, if we fit the S&P 500 time series we obtain the following parameter, which, as expected, shows a positive value for the ξ :

$$\xi = 0.8505214$$
$$\sigma = 1.6233564$$
$$\mu = 4.0204596$$

Next we can see the empirical distribution for 8 periods of one year vs the Frechet theoretical one for the S&P 500 time series :

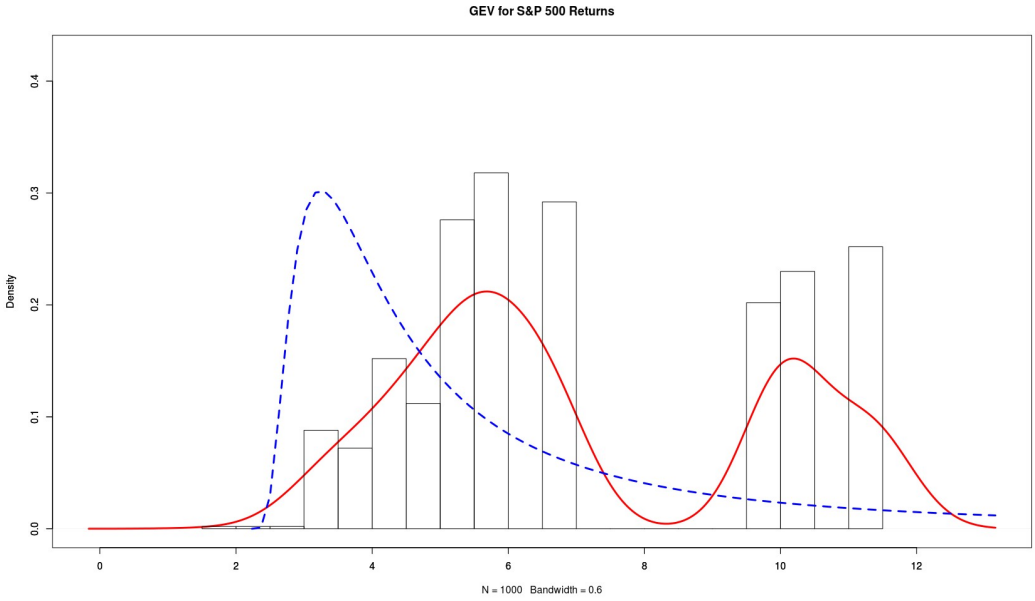


Figure 2.4: Theoretical Frechet vs Sample Simulated Distribution

As we can see the approximation is not close due to the few samples we have for the size for the period, if analyze the residuals we obtain the following values and fitting:

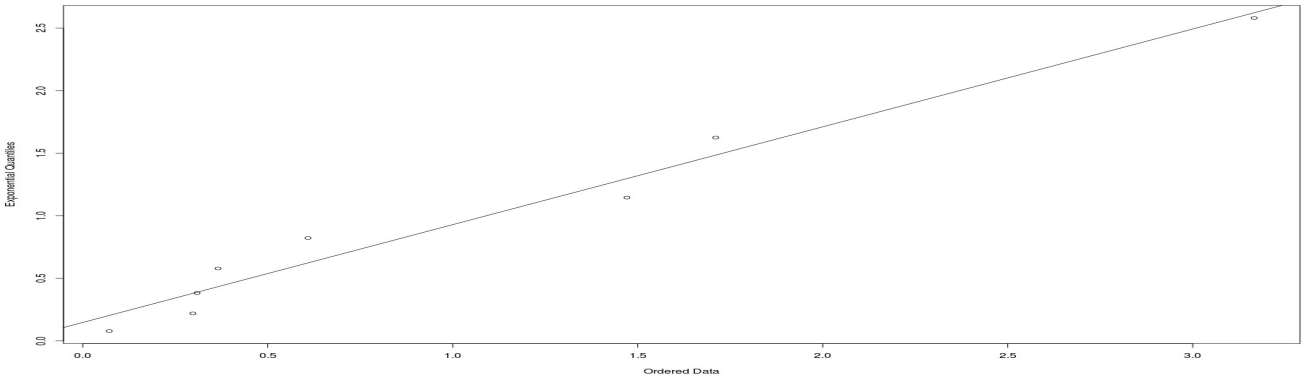


Figure 2.5: Residuals for the Extreme Value Distribution in the S&P 500 fitting

As we can appreciate, besides a quite acceptable fitting for the residuals, we obtain a positive value for the ξ, something typical in financial series. If we now estimate the VaR and expected shortfall for the next 52 periods we obtain the following values.

P	VaR %	ES % (est)
1%	5.43 %	19.88 %
5%	2.94 %	6.69 %
10%	2.56 %	4.33 %

Table 2.3Risk Measurements for EVT

We can see how the ES for the 1% evaluation is quite high compared with the VaR, this is due to the difficulties to estimate the parameters when very few periods are available to do so. In this case the extreme values from the 2007/2008 crisis kick in and heavily affects the 1% estimation for the ES.

Another risk measurement that can be calculated easily with the EVT method is the **Return Level** which accounts for the amount of money that will be lost on average once within the period. In this case for a period of one year we have the following percentage: 7 periods 8 weeks (~1 year)

Return Levels for about 1 Year (95%)

4.25% **5.44%** 7.85%

Which means that, **on average**, once a year the investment will incurre in a 5.44% loss within a confidence interval between 4.25% and 7.85%.

2.5 Peaks Over Threshold (POT)

The approach to VaR calculation using the extreme value theory shows some problems. First, the choice of period length is not clearly defined. Second, the approach is unconditional and, hence, does not take into consideration effects of other explanatory variables .

A different approach to treat extreme values is offered by the POT method, in this case we set a threshold in the time series and we measure, for a given period, how many times the value goes over the threshold and, when doing so, how big are the distances from the threshold. In these conditions we have a Poisson Distribution for the first case and a Generalized Pareto Distribution for the second as the following figure shows:

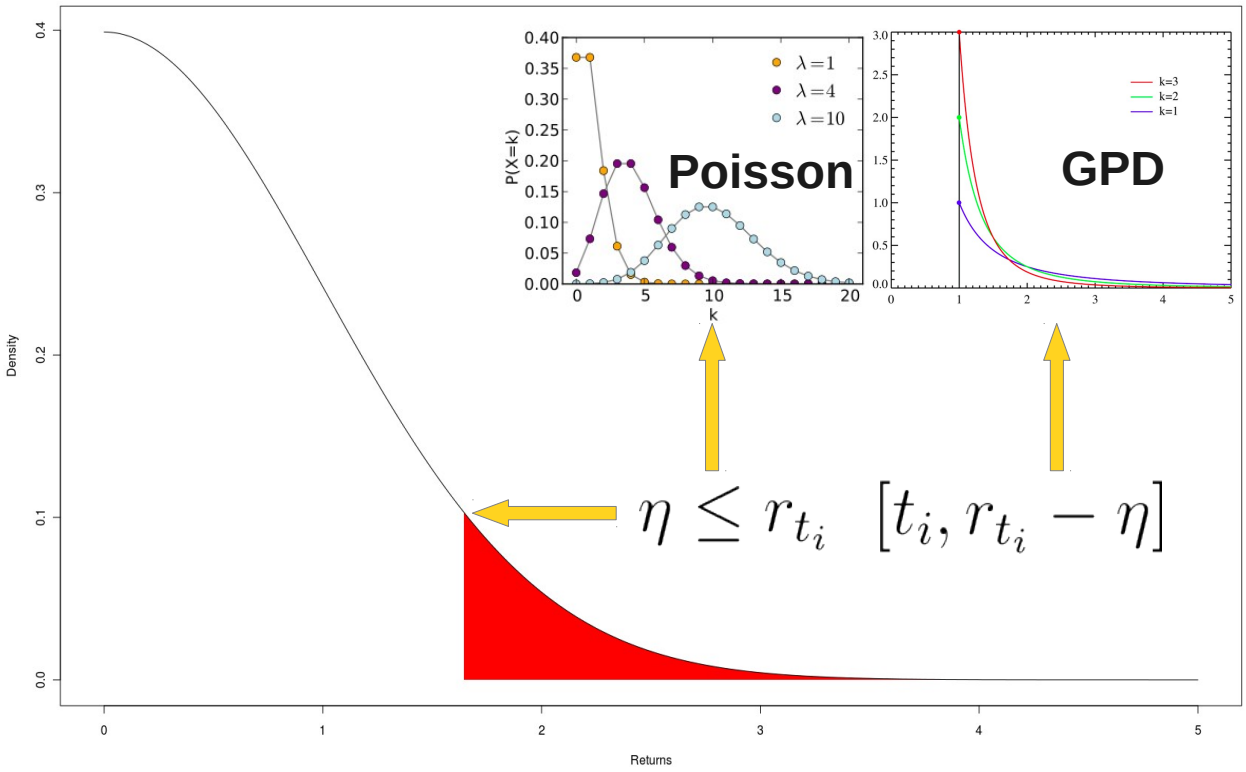


Figure 2.6: Peak Over Threshold Distributions

This method has the advantage over the EVT that it can be use for short as well as for long positions, the only problem is to choose the right threshold.

In choosing a threshold we have to keep first in mind that the threshold follows the theory behind POT, that is, **that particular statistics follow a straight line.**

In the Figure 2.7 we have the Mean Excess, the Shape and the Scale. All these three measurement

should follow a straight line when moving the threshold, therefore, when this is not so we have reached the maximum value for threshold that still makes the theory sound. For instance, for the S&P 500 time series thresholds beyond 6% would not be sustained by the distributions behind the POT theoretical framework since no straight line is followed anymore by any of the mentioned statistics.

But this is not all, we still have to decide if we want to set a threshold for long positions or for short positions, and depending on the market we are investing we might need to set that threshold higher or lower depending on how big is the volatility in that market.

For the S&P 500 time series, and considering the theory and literature sustaining POT, we can set a thresholds to evaluate short and long positions at 2.5% and 4%.

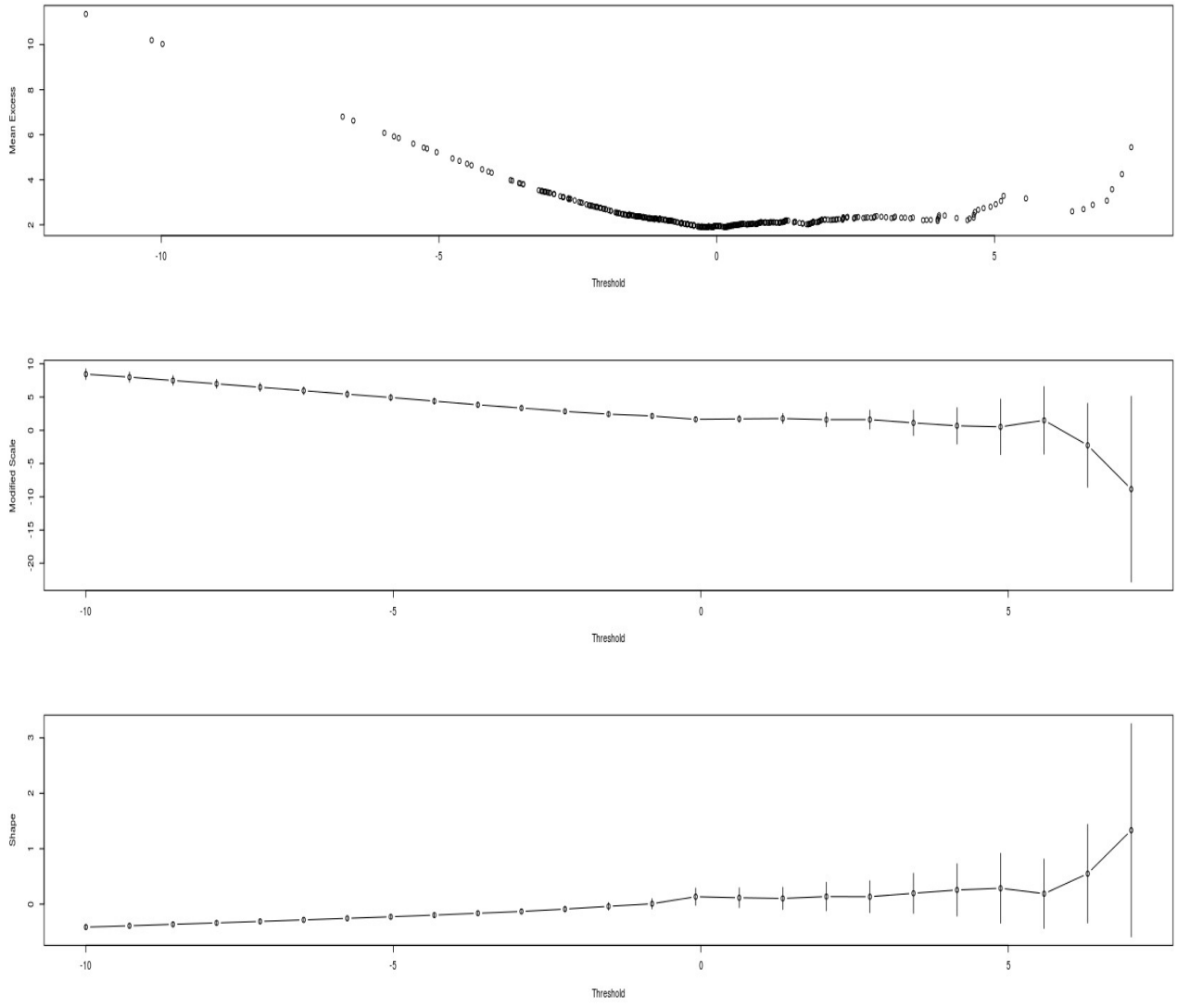


Figure 2.7: Mean Excess, Shape and Scale by Threshold (POT)

And if we now calculate the VaR and the Expected Shortfall for a short and long position we obtain the following results:

2.5 % Threshold			4.0 % Threshold		
P	VaR %	ES %	P	VaR %	ES %
1%	8.32 %	11.41 %	1%	7.97 %	11.60 %
5%	4.35 %	6.89 %	5%	4.35 %	6.79 %
10%	2.86 %	5.20 %	10%	3.19 %	5.24 %

Table 2.4 Risk Measurements for POT

The VaR of a long position must be always bigger than the one for a short position. This is due to the fact that the longer the position the more chances to incurre in loses. In this case we can observe how depending on what probability risk we choose the long position will be determined for one threshold or the other, for instance, at 1% the long position risk is described by the 2.5% threshold whereas for a 10% risk the long position is determined by the 4% threshold.

2.6 The Human Factor Risk

Models from the ARCH family and others are greatly favored by investors to make risk assessment in finance yet, they are all limited by its stochastic nature. They all assume an underlying random behavior and try to measure it. The stock market is in reality a completely deterministic phenomenon **which complexity is beyond our technology to simulate in detail** and, therefore, statistical models handle our ignorance and limitations as if it was a true random process; once the model is fitted and the residuals show the expected distribution behavior, we have finished, **we cannot go beyond with statistical models**.

These models also ignore the fact that the underlying risk factor in their formulas is human. Humans are the ones that take the decision to buy or sell at a specific price, or use an specific strategy or algorithm to take that decision for them. Human emotions, feelings, problems, beliefs... they all affect greatly the decisions we take in any aspect of our lives, but these classical models only care about **how** the time series behaves and pay no attention to **why** they behave the way they do.

When Sir Isaac Newton described the way gravity works he did not explained why beyond saying it

was God's will, but when Albert Einstein gave an explanation to why gravity behaves the way it does this knowledge brought better predictions about the movement of the planets, among many other things. This teaches us that understanding a process goes beyond mere curiosity and the pleasure for knowledge, but it might bring, literally, tangible benefits, especially when applied to finance.

Thus, trying to go beyond a mere description of how volatility behaves and understanding why, implies simulations of the human processes involved. But there is no classical statistical way to approach this problem.

If we could simulate how investors behave we could have a more detailed picture of why volatilities increase or decrease, and this way we will be able to better assess risk in a financial framework.

So the next method to calculate VaR and the Expected Shortfalls will be based on a simulation performed with a simulator developed for this project. Next it will be described the architecture and capabilities of the simulator to continue with the details of the simulations itself. We will then calculate the VaR and Expected Shortfall for the S&P 500 time series and, since a simulation will return as much data as we require, the methodology to calculate VaR will be the empirical quantile applied to the results of the simulation for the period we are interested in analyzing risk.

Finally we will discuss the result of the simulation analysis with the results given by the classical methods to calculate risk measurement explained before.

3 THE SIMULATOR

3.1 Random Number Generator (RNG)

Strictly speaking we cannot say that a RNG is truly a random number generators, to simplify the nomenclature, and within the context of the project, we will understand as random samples those that posses features compatible with randomness.

Next we can see an **Unified Modeling Language**[20] (UML) class diagram depicting the structure of the RNG develop for this project, despite the fact that only a couple of distributions are used within the simulator it has been developed a whole collection of distributions so that the simulator can be easily expanded to deal with more complex simulations.

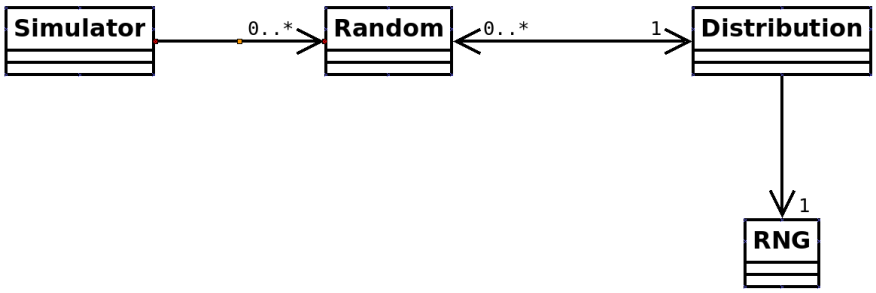


Figure 3.1 UML Diagram for the RNG

3.1.1 Basic Description of Classes

The basic design contains a class name *Simulator* which is in charge to manage every simulation and accesses the class *Random* in order to obtain random samples when the are required. The class *Random* handles the methods and parameters that setup the the generations of random samples, in particular its distributions and the RNG that generates it. The cla*ss Distribution* contains several methods used to convert uniform random samples coming from the class *RNG* into any other random distribution required. Finally the class *RNG* implements several methods to generate uniform random samples and manage the parameters that sets them up.

20 http://uml.org/

3.1.2 Basic Description of Classes Interactions

The class *Simulator* can create as many instances of the class *Random* as required, on the other hand the class *Random* will be subscribed to a unique class *Distribution* that will define uniquely the kind of distribution that the random sample will follow. The class *Distribution* will be also uniquely subscribed to t*he* class *RNG*, this way every distribution uses a unique *RNG* for generation. Nonetheless the class *Distribution* will also be able to use as many instances of the class *Random* as required in order *to cr*eate complex random numbers composed by several other random distributions. The class *RNG* is se*lf*-contained and simply offers its functionality to the class *Distribution*.

3.1.3 Methods per Class

Next we can find a brief functional description of the methods available in each class and its setup options:

3.1.3.1 Simulator

The *Simulator* class contains the main code for executing the simulations, all the methods in the class are oriented for the setup and management of simulations, fittings and forecasting, the methods available for these tasks are:

3.1.3.1.1 read_events

This method read external events like depression and mood levels, volume of transactions in the market or atypical impacts caused by news.

3.1.3.1.2 simulation

This method engages the main process of simulation

3.1.3.1.3 define_target

This method does a statistical analysis of a times series and set the results as the target to be achieved by the genetic algorithm optimizer.

3.1.3.1.4 print_chromosomes

This method prints the chromosomes from the genetic algorithm into a file for ulterior statistical

analysis.

3.1.3.1.5 print_forecast

This method prints the forecast from the simulations for ulterior statistical analysis.

3.1.3.1.6 fit_model

This method uses a genetic algorithm in order to find the parameters that better fit the time series statistical analysis that has been set as target.

3.1.3.1.7 simulate

This method sets a particular set of parameters for a simulation before it is executed.

3.1.3.2 Random

Instances for the class *Random* are initialized with the statistical distribution that the generated random samples are going to follow and what *RNG* will generate them, if no parameters are entered it will use the default configuration and will return a standard uniform distribution generated by the Mersenne Twister[21] m19937 algorithm. The purpose for the class is to offer a collection of methods manage random numbers, the methods available are:

3.1.3.2.1 rnd

This methods has no parameters and simply returns a random number every time that it is called. The numbers returned shows the properties determined in the initialization of the class that contains it.

3.1.3.2.2 dh_ascii

DIEHARD is a batch of test to verify the quality of a RNG through the analysis of series of numbers generated by it. In order to execute this batch of test the numbers must have a defined format and size, this method returns by default 3.000.000 numbers of 32bits in the format required by *DIEHARD*[22].

3.1.3.2.3 dhr_ascii

This method is equivalent to dh_ascii but for *DIEHARD,* but in this case it returns a serie of numbers formated to be batched analyzed by the *DIEHARDER* test. The test *in DIEHARDER* are more intensive

21 Matsumoto, M.; Nishimura, T. (1998).
22 The minimum number advised by G. Marsaglia is 2.9 millions

and they require more time for it execution that those in *DIEHARD* (a full test analysis for a file without cycles required can take more than a day to finish, these test also require much larger samples upon the selected test to be executed.

If the sample is not big enough *DIEHARDER* reuse the sample which weakens its results. By default the method generates a file with 10.000.000 numbers though this number is not enough to avoid reuse in all the test.

3.1.3.3 Distribution

The *Distribution* class is also initialized determining what distribution must be followed by the random samples, and what *RNG* must generate them. The distributions available are:

- **Standard Uniform:** Real numbers within the interval **[0,1)** (Default Distribution)
- **Discrete Uniform:** Integers between the parameters **a** y **b**.
- **Exponential:** parameter **lambda**.
- **Erlang:** parameters **k** and **lambda**.
- **Weibull:** parameters **alpha** and **beta**.
- **Triangular:** parameters **a**, **b**, **c**.
- **Geometric:** parameter **p**.
- **Normal:** parameters **mu** and **sigma2**. (Box & Muller – Classic).
- **Normal2:** parameters **mu** and **sigma2**. (Box & Muller – Monte Carlo. R. Knop).
- **Normal3:** parameters **mu** and **sigma2**. (Monte Carlo / Exponential Majority).
- **Normal4:** parameters **mu** and **sigma2**. (Polynomial).
- **Lognormal:** parameters **mu** and **sigma2**.
- **Generalized Pareto:** parameters **mu**, **sigma** and **epsilon**.

Besides the initialization methods and management of distributions there are two more methods:

3.1.3.3.1 rnd

Just like in the class *Random*, this method has no parameters and simply returns a random number every time it is called.

3.1.3.3.2 zeroneize

This method turns an integer into a real number within the range [0,1)

3.1.3.4 RNG

This class is the heart to generate random numbers, here are implemented all the algorithms and strategies to return list of integers showing features compatible with randomness. When instantiating a class an algorithm has to be chosen, if no algorithm is found it will use by default the RNG Mersenne Twister mt19937.

The available algorithms are:

- **Mersenne Twister**[23]: mt19937 version.
- **Multiply With Carry**[24]: Designed by George Marsaglia.
- **Linear Congruential Generator**[25]: Implemented in languages like C/C++.
- **CTR_AES**[26]: Cryptographically strong.

3.1.3.4.1 rnd

Again, and just like in the classes *Random* and *Distribution*, this method has no parameters and simply returns a random number when called.

3.1.3.4.2 seed

This method its initialize when the class is and its functionality is to aid the RNG's to find an appropriate seed when there is none specified in the parameters. The algorithm to generate seeds is based in a Linear Congruent Generator.

23 Matsumoto, M.; Nishimura, T. (1998).
24 Marsaglia, G.; Zaman, A. (1991).
25 S.K. Park and K.W. Miller (1988).
26 Joan Daemen, Vincent Rijmen (2002).

4 THE SIMULATION

Many investors believe that emotions have nothing to do with the market and that Efficient-Market Hypothesis holds[27], on the other hand, there are studies that show that phenomena like the Halloween Indicator[28] might have some self-fulfilled psychological basis.

Around 70% of the stock market in the United States of America is controlled by algorithmic trading, and some of these algorithms try to use sentiment analysis[29] to interpret in real-time what's the human perception about news[30] and how this will affect the market, there are even companies like **Opfine**[31] selling sentiment analysis over a portfolio of companies.

Beyond the effect that psychology might have in the price of the stock market, this project will mainly be concerned with the effect that human psychology have in volatility, because, after all, this is a magnitude which information can be foretasted in a statistical way.

The simulation of a financial time series in this project will try to be as deterministic as possible, only introducing stochastic behavior when no theory or deterministic model can explain a phenomenon. Since we are exploring the effects of human psychology on the volatility we will need a tool to find out how investors feel in a particular value, but that is not an easy or even possible task. Much easier is to find out about how a whole nation feels, and try to relate that to the volatility of a index comprised of a large number of companies within the country, that is why in this project we use the S&P 500 index.

Thus, the first step will be to find out a sensible and mensurable relationship between human emotions and volatility.

4.1 The Model

The first approach to create a behavioral model of volatility for this project was related to sentiment analysis techniques. The basic idea was to find correlations between the mood of the general population

27 This hypothesis asserts that financial markets are "information efficient". That is, one cannot consistently achieve returns in excess of average market returns on a risk-adjusted basis, given the information available at the time the investment is made.

28 This indicator is a variant of the stock adage "*Sell in May and go away*"; the belief that the period from November to April inclusive has significantly stronger growth on average than the other months.

29 Sentiment analysis or opinion mining refers to the application of natural language processing, computational linguistics, and text analysis to identify and extract subjective information in source materials.

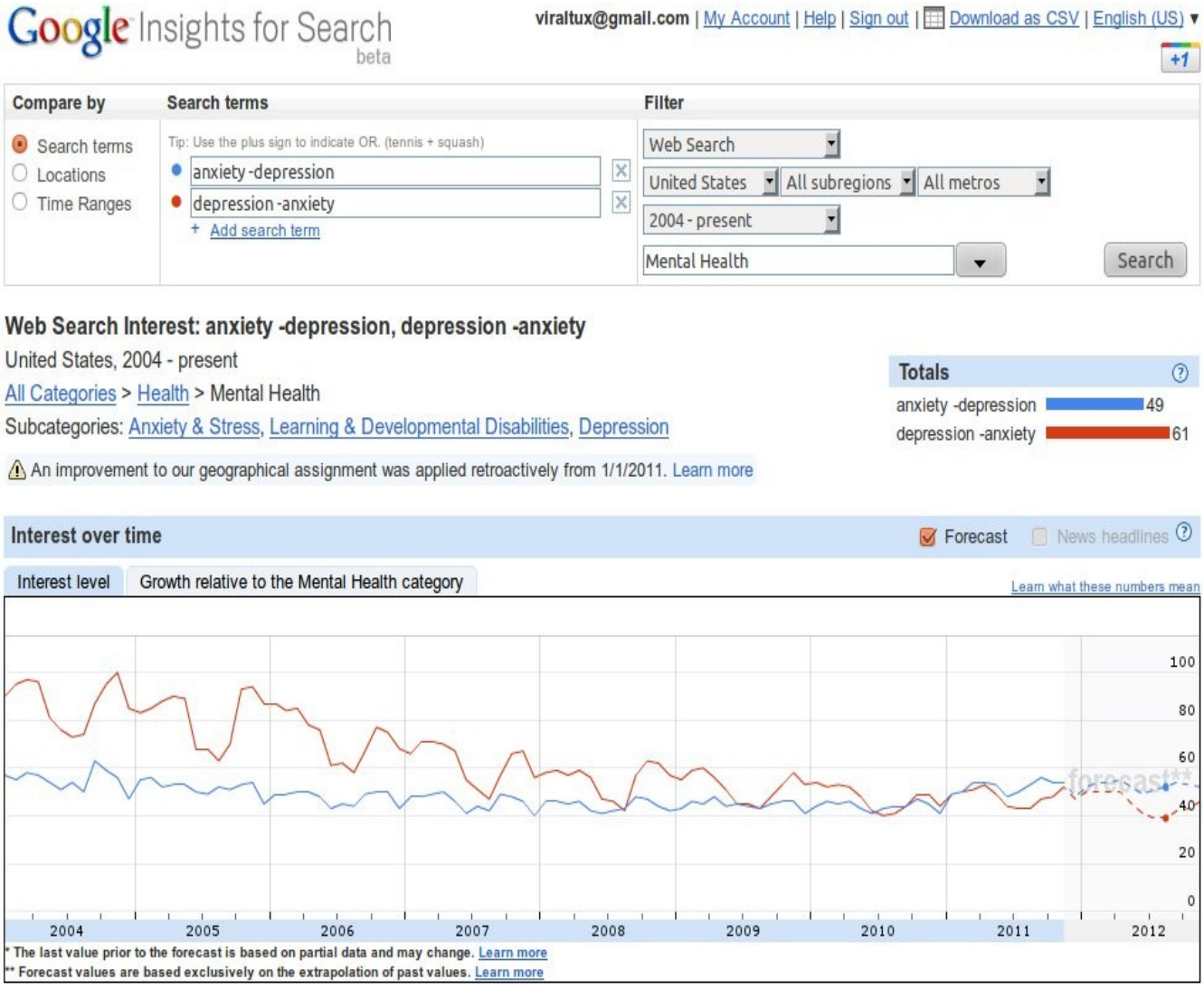
30 Engle, R.F.; Ng, V.K. (1991)

31 http://opfine.com/

in key aspect of finance and society in the USA and the volatility in the S&P 500. Unfortunately the free resources available would not allow us to text-mine historical data and the non-free resources were beyond the budget for this project.

Nonetheless, there is a relatively new tool deployed by Google corporation named ***Google Insights for Search***[32] which facilitates just enough information to perform a sentiment analysis for the purpose of this project.

Google Insights allow us to compare search volume patterns across specific regions, categories, time frames and properties. For this project it has been decided to seek data about the keywords **depression** and **anxiety** within the USA in the category of Mental Health from the year 2004 until 2012.



Frigure 4.1: Google Insight results for anxiety and depression within the U.S.A.

32 www.google.com/insights

The year 2004 is the minimum year available for search in *Google Insights* and the results are weekly aggregated. The numbers on the graph reflect how many searches have been done for a particular term, relative to the total number of searches done on Google over time. They do not represent absolute search volume numbers, because the data is normalized and presented on a scale from 0-100. Each point on the graph is divided by the highest point, or 100. When there is no enough data, 0 is shown.

Let's plot now depression and anxiety separately to have a better look at its behavior:

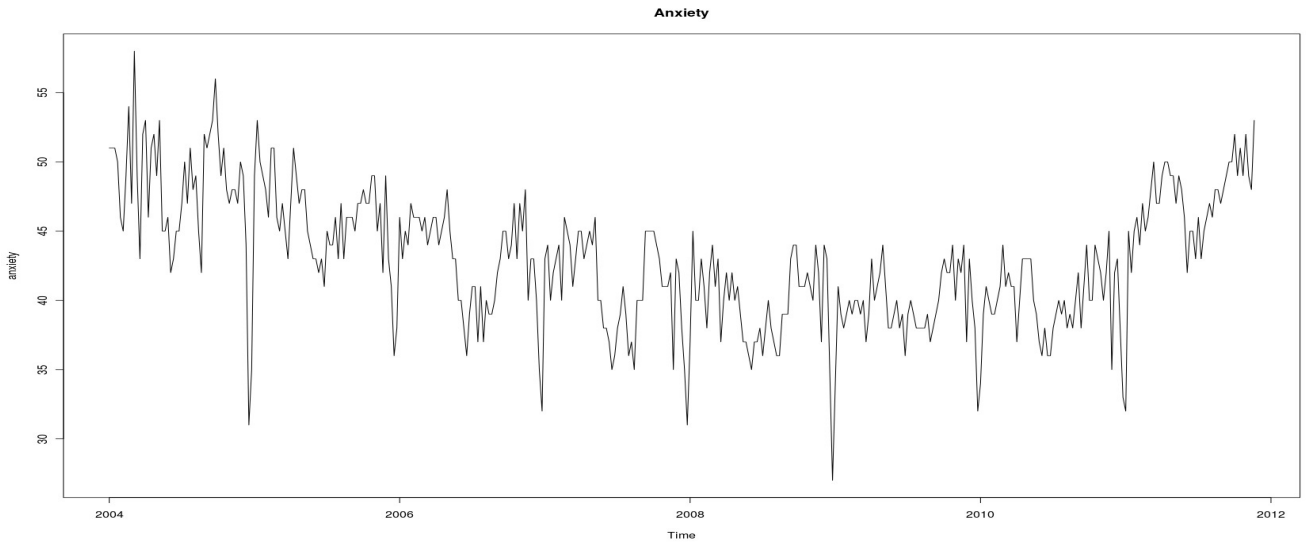


Figure 4.2: Anxiety volume search at Google since 2004

We can see much clearly now how anxiety shows a change in its trend around the time the financial crisis began with a period of high volatility.

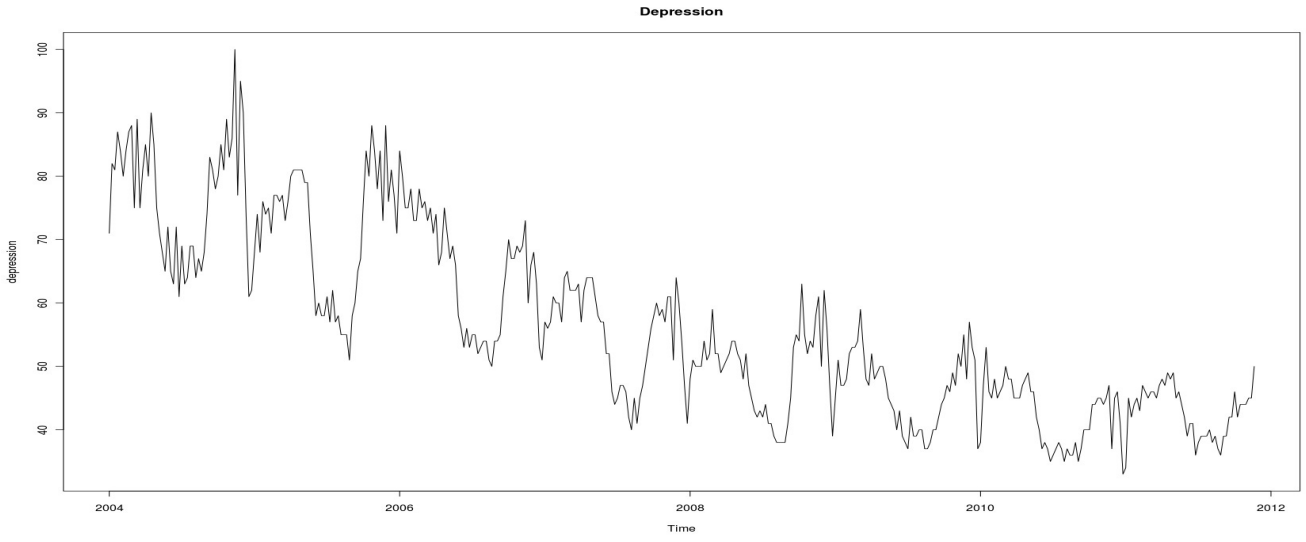


Figure 4.3: Depression volume search at Google since 2004

Depression, on the other hand, steady decreases yet it seems to slow that trend after 2009, that is right after the financial crisis.

The reason why anxiety and depression were chosen as keywords was because one purpose of this project is to study states of mind that might affect differently the volatility in the market. Expectations were that depression will affect volatility by reducing it and anxiety would affect volatility by increasing it.

But before we go any further, we need to analyze again the S&P 500 time series this time considering the data weekly and from 2004 until 2011 so that we can compare results with the sentiment data gathered from Google Insights.

If we calculate the ACF and PACF for the returns in these conditions we have the following plots:

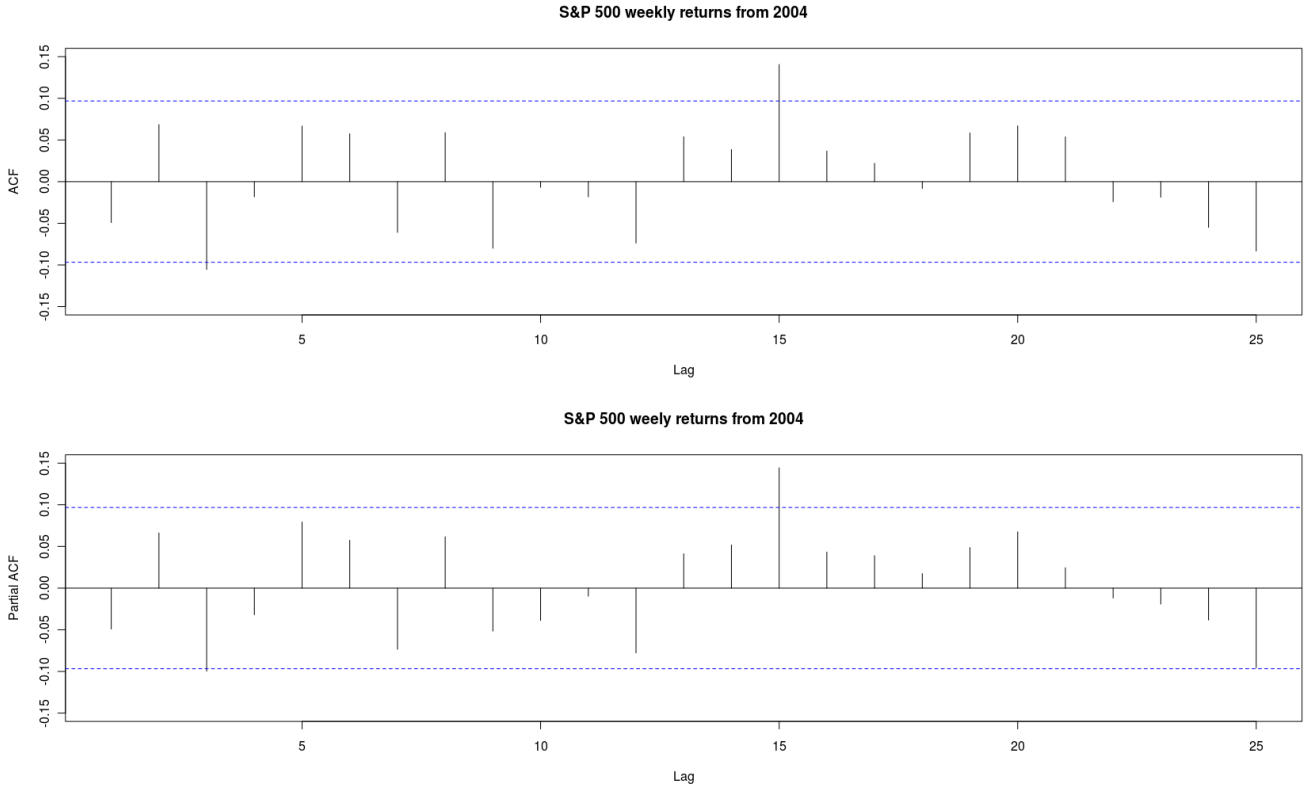


Figure 4.4: ACF and PACF for S&P 500 weekly returns since 2004

These plots show no auto-correlated behavior whatsoever for the returns which implies we can consider a model where returns and the impact (residuals) are the same.

If we now calculate the correlation matrix for anxiety, depression, volume for S&P 500, returns and squared returns for S&P 500 we obtain the following.

	anxiety	depression	volume	returns	returns^2
anxiety		**0.57**	**-0.39**	0.01	-0.08
depression			**-0.65**	-0.01	-0.07
volume				**-0.11**	**0.37**
returns					**-0.29**
returns^2					

Table 4.1: Correlation Matrix for anxiety, depression and volume, returns and squared returns for the S&P 500.

There are many interesting things to comment in the matrix in Table 4.1, first we see that the returns correlates to nothing except, of course, to the squared returns, this is totally expected since, as we have already seen the returns, or the price, is not something we can not easily predict.

We can also see a very noticeable positive correlation of 0.37 between the volume and the squared returns, this is an indication that the higher the trading in the market the higher the volatility might be.

But the most interesting correlation of all is the one for volume and depression; a whooping negative correlation of -0.65. Now would be the time to chant the old statistician adage of “correlation does not mean causation” but in this project we assume that there is a psychological effect in the market and we are merely trying to measure it.

On the other hand we also find a significant negative correlation between volume and anxiety though not so important as with depression. This seems to be bad news for the theory that anxiety is causing volatility to increase and that we should keep looking for other keywords, but let's have a look first at the cross-plots of all these parameters.

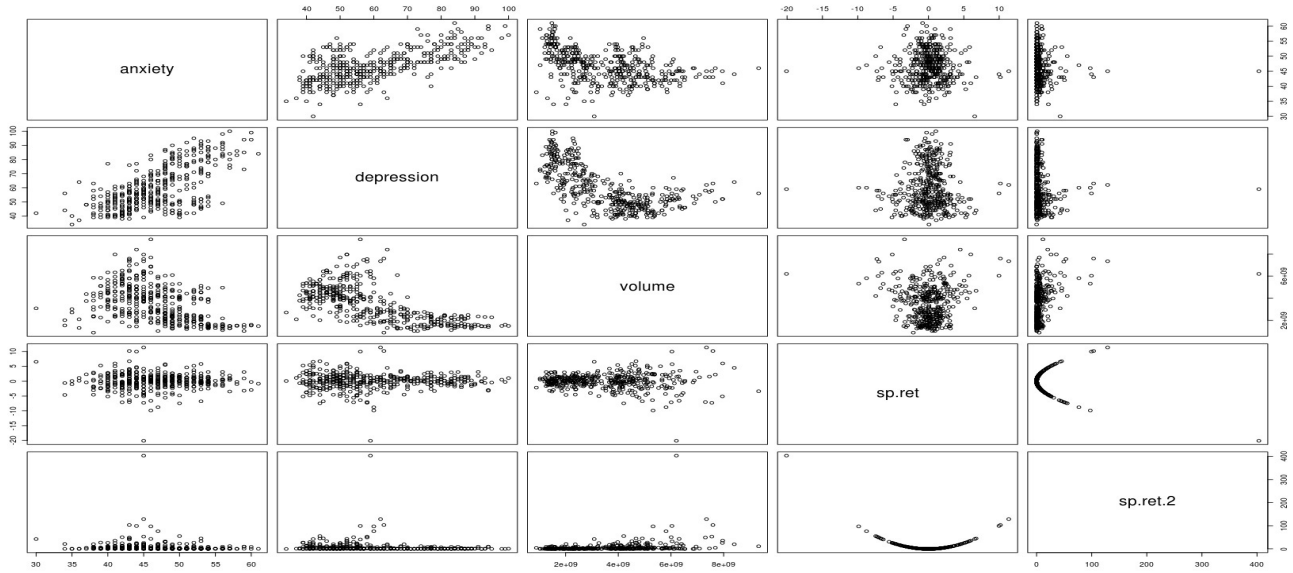


Figure 4.5: Plot anxiety, depression, S&P 500 volume, S&P 500 returns and squared returns

In the plots we cans see the striking relationship between depression and volume and also how anxiety seems not to offer anything better than depression does. We can nonetheless combine anxiety and depression into a new behavioral parameter that can name **mood**.

This new behavioral parameter will be the result of making the operation anxiety – depression. This operation makes sense since both parameters are measure in the same scale. When mood equals 0 it means anxiety and depression and leveled, therefore the higher the anxiety the higher the value of mood and the lower the mood the higher will be depression. This way this new psychological parameters shows the level of anxiety accounting somehow for the depression associated with it.

With this new parameter we obtain the following variance-covariance matrix:

	mood	depression	volume	returns	returns^2
mood		**-0.95**	**0.61**	0.01	0.05
depression			**-0.65**	-0.01	-0.07
volume				**-0.11**	**0.37**
returns					**-0.29**
Returns^2					

Table 4.2: Variance-Covariance Matrix for mood, depression and volume, returns and squared returns for the S&P 500.

We can see the parameter mood has a high negative correlation with depression and a high positive one with volume. If we calculate the new cross-plots for this parameters we have the following:

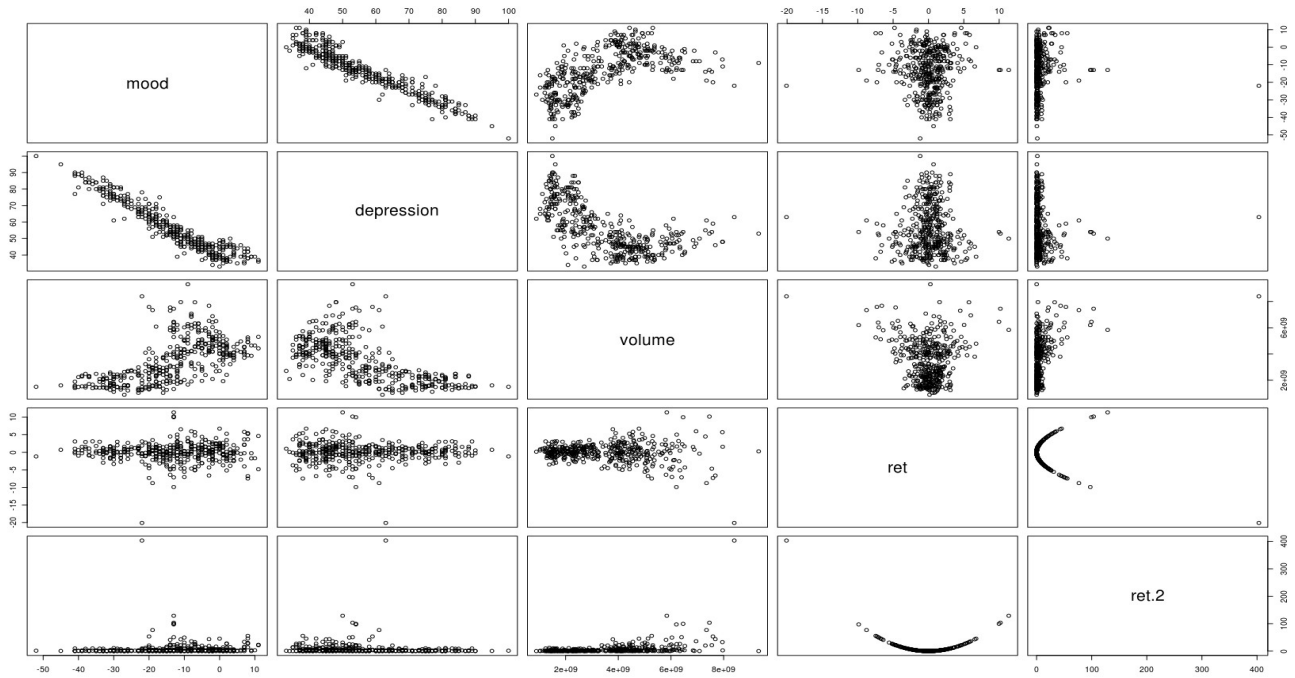


Figure 4.6: Plot anxiety, depression, S&P 500 volume, S&P 500 returns and squared returns

Now the we can appreciate a linear relationship between mood and depression with higher heterocedasticity when the mood increases in value, this is due to the fact that the higher the mood the lower is the weight of depression in its value and the higher is the one of anxiety.

Since volume has a clear effect on the squared returns and, at the same time, volume has a negative correlation with depression and a positive correlation with mood, now we have reasons to believe we might have two psychological parameters that affect volatility positively and negatively.

In the introduction we saw that the volatility is expressed in a multiplicative model and in the S&P 500 data weekly data since 2004 until 2011 we have that the returns equals the impact, therefore $r_t = \sigma_t \cdot \epsilon_t$ which means that $r_t^2 = \sigma_t^2 \cdot \epsilon_t^2$ and consequently $\log(r_t^2) = \log(\sigma_t^2 \cdot \epsilon_t^2) = \log(\sigma_t^2) + \log(\epsilon_t^2)$
So by applying logarithms to the squared returns we have a linear expression of the logarithm of the volatility, and if we calculate again the correlation matrix with the logarithm of the squared returns we have:

	mood	depression	volume	returns^2	log.ret^2.
mood		**-0.95**	**0.61**	0.05	**0.17**
depression			**-0.65**	-0.07	**-0.18**
volume				**0.37**	**0.36**
returns^2					**0.41**
log.ret^2.					

Table 4.3: Correlation Matrix for mood, depression and volume, squared returns and log squared returns for the S&P 500.

Once we have a linear expression for the logarithm of the volatility we can observe a noticeable positive and negative correlation with the two psychological parameters depression and anxiety when before we had none, this is one more sign of possible psychological effects on volatility.

At this point we might consider use only the parameter mood an discard depression altogether, nonetheless using the parameter depression on its own will give us a better picture of how these parameters affect the later simulations separately and it will give us a deeper insight about wether both parameter are needed or we can safely use only the mood psychological parameter as a drive for the volatility in the time series.

In the next figure we can observe a graphical comparison between the new parameter mood and the S&P 500 time series.

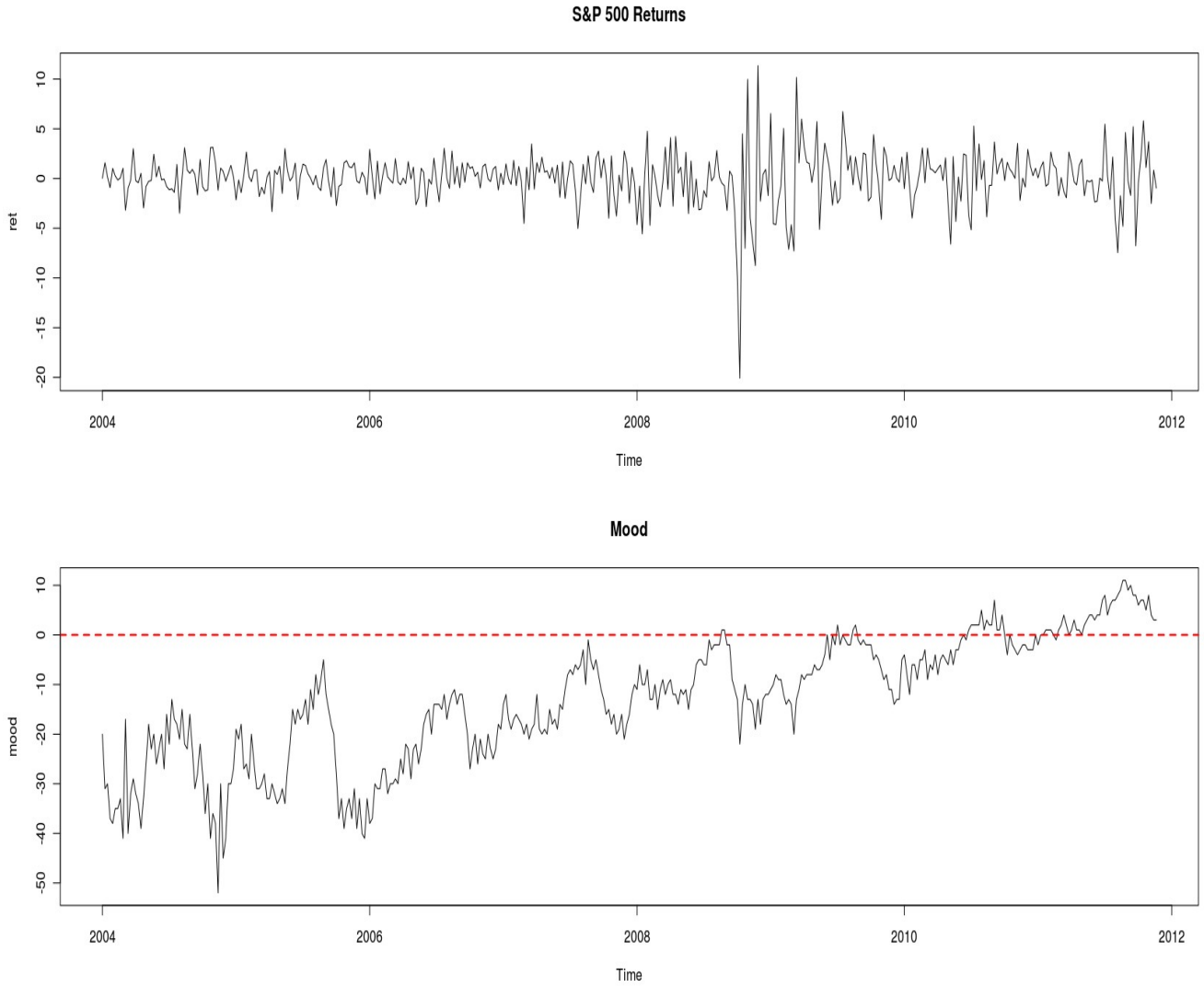


Figure 4.7: S&P 500 Returns vs Mood

We can appreciate that only when the mood is well below zero we have a low volatility period in the returns, once the mood approaches zero we start to observe an increase in the volatility and just about when the mood reaches for the first time positive values at the end of 2008 we have a full blown out financial crisis showed by a huge volatility in the markets.

We can also observe periods of high a low volatility after 2008 that roughly matches the periods of high a low mood, these observations joined with all we have seen so far seems to support the idea of psychological influence in the markets volatility.

Now is about time to gather all these correlations into an structured model and validate its design with the data available.

4.2 Structural Equation Modeling

Structural Equation Modeling[33] (SEM) is a statistical technique for testing and estimating causal relations using a combination of statistical data and qualitative causal assumptions. This technique is suited for both confirmatory and exploratory modeling, and therefore for theory testing and theory development.

The primary purpose for this technique in this project is to show that the latent variable of volatility is affected by psychological factors. Making use of the the linear properties for the logarithm of the squared returns given by $\log(r_t^2) = \log(\sigma_t^2 \cdot \epsilon_t^2) = \log(\sigma_t^2) + \log(\epsilon_t^2)$ we will try to fit the following structured equation model:

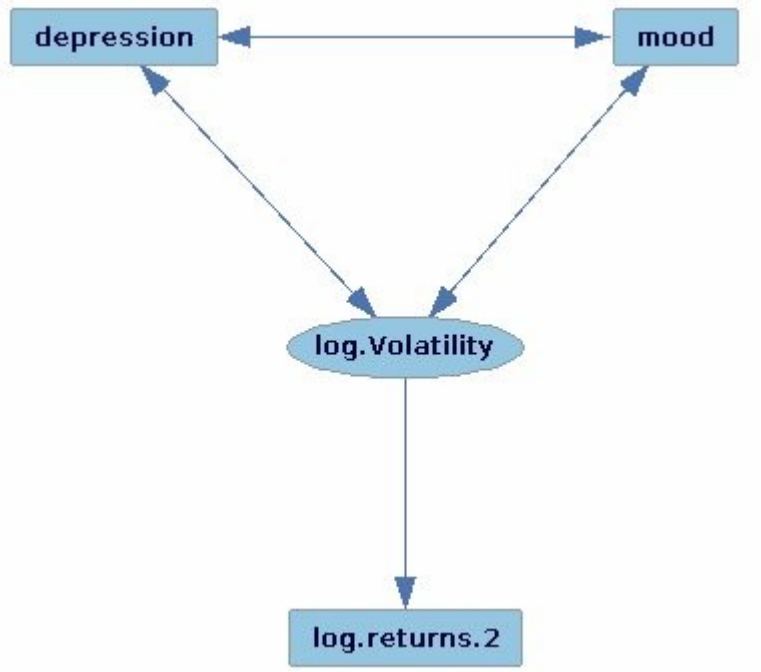


Figure 4.8: Structural Equation Model for psychological factors and volatility

In this model we presume that depression, mood and the log volatility interact and affects each others behavior. The log volatility is a latent variable and so it is as well in the structural model which, following the linear equation shown previously equals the logarithm of the squared returns plus an error. The following plots shows the errors implied by the model:

33 Haavelmo, T. (1943)

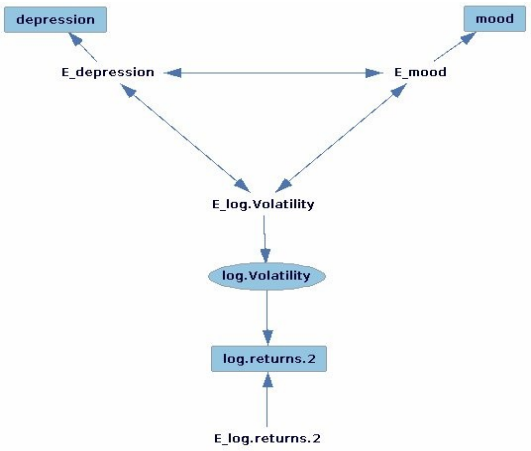


Figure 4.9: Structural Equation Model for psychological factors and volatility plus errors

Now, we have -2 degrees of freedom and Structural Equation Models with negative degrees of freedom are under-identified, and other model statistics are meaningless. We need at least 0 degrees of freedom and that can only be achieved by either increasing the number of variables or fixing some parameters in the model.

Two obvious candidates are the variance between depression and mood and the linear coefficient between the log volatility and the logarithm of the squared returns. These to parameters would give us back to more degrees of freedom having total of zero degrees of freedom to estimate the remaining parameters in the model. The only problem no having positive degrees of freedom is that the results are uninformative and a p-value on the model cannot be establish, nonetheless, the results might serve as clues to fix more parameters in later models. After fitting the model with zero degrees of freedom this is the result:

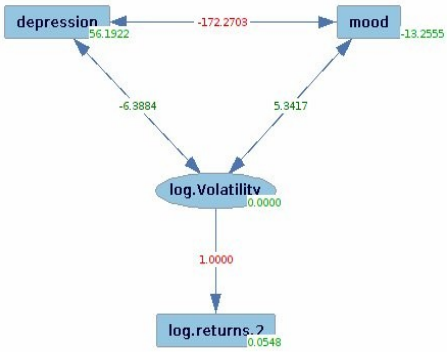


Figure 4.10: SEM estimation with two parameters fixed

The figures in red are the fixed parameters expressing the variance between the depression and mood and the theoretical value between the log volatility and the logarithm of the squared returns.

We can observe how the estimation offers a positive covariance between mood and log volatility and a negatively one for depression and mood, this result shows that depression would have a decreasing on the volatility, whereas mood would a have a increasing one.

Now, to be able to calculate the p-value for a model we need at least one degree of freedom, so next we are going to fix the value estimated for the variance between mood and log volatility and we will estimate the model again, these are the results:

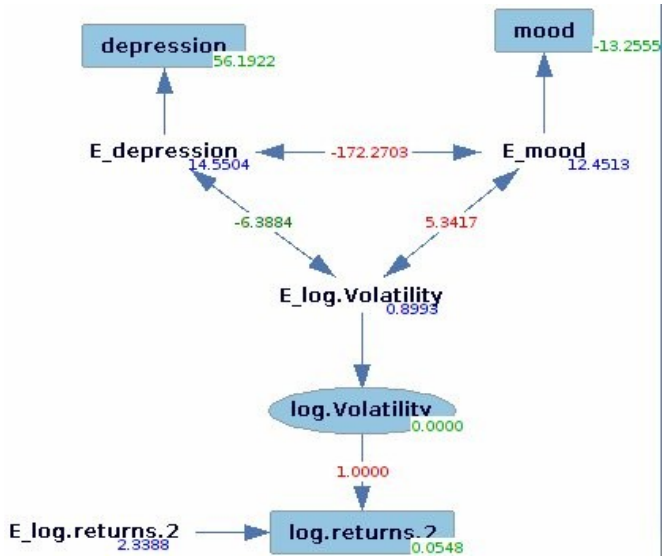


Figure 4.11: SEM estimated with one degree of freedom

In red we can see the fixed parameters and in green the estimated ones and in blue the estimated variance for the errors, after fixing the variance for the mood and the log volatility the model offers the following results:

Estimated degrees of Freedom = 1
Chi Square = 0.0012
P Value = 0.9729
BIC Score = -6.0174

Table 4.4 SEM Chi Square, p value and BIC Score

The above chi square test assumes that the maximum likelihood function over the measured variables has been minimized. Under that assumption, the null hypothesis for the test is that the population covariance matrix over all of the measured variables is equal to the estimated covariance matrix over all of the measured variables written as a function of the free model parameters--that is, the unfixed parameters for each directed edge (the linear coefficient for that edge), each exogenous variable (the variance for the error term for that variable), and each bi-directed edge (the covariance for the exogenous variables it connects). The model is explained in Bollen, Structural Equations with Latent Variable, 110. Degrees of freedom are calculated as m (m + 1) / 2 - d, where d is the number of linear coefficients, variance terms, and error covariance terms that are not fixed in the model.

With a p-value of 0.9729 we have an excellent fitting for this model. This of course does not mean that this model it is true, only that the data available agree with it, nonetheless, the whole purpose of these analysis have been to support with data the theoretical stand that psychological factors do have an interaction with volatility in financial markets.

With all the data an insights discussed in the previous chapters we now can move on and design a suitable simulator for the volatility, and the first step will be to make a few considerations about how the price in the market behaves since, after all, it is the volatility of the price what we are trying to simulate.

4.3 The Price

The simulation is going to be focused on the volatility but since the volatility is simply the variance of the price and since the simulation pretends to be as deterministic as possible that means that it has to try to be deterministic as possible calculating the price.

In fact, as it was mentioned in the introduction, the perfect strategy for investment exists and its based in simply trying to figure out what everybody else strategy is and simulate it, now let's see how easy would that be.

We can find a wide range of investment strategies in the stock market some of the most popular are among the following:

- Algorithmic trading
- Buy and hold

- CANSLIM
- Contrarian
- Liability-driven investment strategy
- Market timing
- Trading strategy
- Trend following

And as it was mentioned before just the Algorithmic trading strategy takes around 70% of all the transactions in the USA. Now, since we are going to focus in the S&P 500 Algorithmic trading becomes very important for the simulation. Algorithmic trading has a subfamily of strategies on its own and among those we can find:

- Trend Following
- Pair Trading
- Delta Neutral Strategies
- Arbitrage
- Conditions for arbitrage
- Mean Reversion
- Scalping
- Transaction cost reduction
- Strategies that only pertain to dark pools

And companies like Nanex[34] have been able to identify through reverse-engineering techniques some of the algorithms used within these strategies, here is shown a basic list of them offered by Nanex:

34 http://www.nanex.net

-TILT-	Bot Wars	Flag Repeater	Orange Marmalade	The Spartan
2-step	Botastic	The Flood	The Outer Limits	Spastic BAT
2200 BTU's	BOTvsBOT	Flutter	Pacific Rim	Street Lamps
4-Wheel Drive	The Bridge	Focus	The Palace	Stubby Triangles
60-Step	Bristles	The Follower	Penny Pincher	Sunshowers
The Abyss	Broken BAT	Fred	The Pepsi Challenge	T1 Killer
Algo Mountains	Broken Highway	Frog Pond	Periscopes	Take Two
Almost Human	Broken SKY	From Above	Petting Zoo	Tank Tracks
Apollo	Broken Zanti	From Below	Pinger	Tesla's Cathedral
Asimov's Nightmare	Buckaroo Banzai	Full Moon Rising	Plate Shift	Test Pattern
The Awakening	The Bug	Fuzzy Orange	Platform Drilling	Them
Back to School	The Bunker	Gold Finger	The Port	tHigh EQ
The Bagman	CancelBot	Gone Fishing	Power Line	The Thin Blue Line
Banker's Ball	CancelBot Jr.	Good Luck Human	Power Tower	Thin Blue Line
Bankers Blitz	Cancelled Check	The Green Flash	Puzzle Pieces	Things that make you go 'hmmmm'
BAT Cave	Cannons	The Green Hornet	The Quota	The Tickler
BAT Code	Cannons 2	Ground Strike	Quota Catcher	To The Moon, Alice!
BAT Discovery	The Carnival	Hairline	Quota Machine	Twilight
BAT Dribble	Castle Wall	Heart Attack	The Raceway	Wading Pool
BAT Fence	Changing Tide	High EQ	Racing Stripe	Wake Up Call
BAT Hats	Cherokee Nation	High Tide	Railway	Warp 15
BAT Horizon	The Circus Comes to Town	I'm A PC	The Ramp	Waste Pool
BAT Lego	City Of BATS	Inner Chart	Red Sky at Night	When the Levee Breaks
Bat Pig	City Under Siege	Jump Shot	Red Tide	Wild Thing
Batastic	The Click	Junior	Redline	Wild Thing Edge
Batsicles	Clockwork Orange	Just Ask	Repeater Wars	Yellow Picket Fence
BBOBomber	Clogged Artery	The Knife	Robot Fight	Yellow Snow
The Beach	Continental Crust	Landmine	Robot Hunting	You Don't Know Jack
Beyond the Blue Wall	Control Tower	Life and Death	Rock Star	Zanti Mahem
Bid Stuffer	Crazy Eyes	Lightning Strike	Rollerball	The Zanti Misfit
The Bird	The Crown	Living On The Edge	Rougue Wave	Zapata
Blast This	Danger Will Robinson	Local Dump	The Rover	Zappa Street
Blockhead	Day Trippin	Low Tide	Runaway	Zapper Clone
Blotter	The Dead Pool	Made in America	S.O.S.	Zero to Sixty
Blue Bandsaw	The Deep	Mainframe	Scissors	
The Blue Bidder	The Deer Hunter	Mannie, Moe and Jack	Scofflaw	
Blue Blaster	Deer vs. Bat	Marco Polo	Sea Level	
Blue Blind	Depth Ping	Market Share	Sea of BATS	
Blue Blocker	Detox	Master Blaster	Sea of BATS Star	
Blue Flicker	Dinosaur Hunt	Maxy-Zapper	The Search	
Blue Ice	Dirty Glaciers	Meteors	Search Bots	
The Blue Pig	Don't Tread On Me	The Monster	The Seekers	
Blue Stubble	Double Dip	Monster Mash	Seen Too Much	
Blue Thicket	Double Pole, Double Throw	Morning Zanti	Seizure	
Blue Wave	The Drowning	The Morphing	Shades of Blue	
Blue Zinger	Early Discovery	NARA Zapper	The Shredder	
Bluegrass	Early Riser	No Joy	Simple BAT	
Boston Buck'r	Enchanted Forest	No Reason	Single Track	
Boston Shuffle	EPIC Zapper	Obstructus Maximus	Social Butterfly	
Boston Zapper	Eraser Head	One Ping Only	Solar Flare	
Bot Town	Faster Zapper	Orange Crush	Soylent Blue	

Table 4.5: List of names of some algorithms used in algorithmic trading

At this point we can imagine that the complexity to simulate all these strategies is huge, but if we consider that every strategy and algorithm have its own parameters and that they might change forth and back depending on how they are performing the task to predict a price based on this becomes

impossible in practical terms. Yet, this gigantic rock-paper-scissors that the market is with its thousands of different strategies could be simulated and adjusted to analyze the volatility, but the complexity of such simulation is beyond the ambitions of this project.

Nonetheless, we will approximate the outcome of such interaction with deterministic models which parameters will be adjusted considering that time series to emulate.

4.4 Steps

Next it will be explained the steps that a standard simulation will take and, by doing so, it will be described the classes and methods involved:

4.4.1 Simulation parameters

The first step will be to set up the parameters for the simulation, these parameters are the following:

- **:start_date** Initial date for the simulation
- **:gap** Time length between transaction events in the simulation
- **:end_date** Final date when the simulation stops
- **:output** If true prints a file the the results for statistical analysis
- **:ninvestors** Number of investors involved in the simulation
- **:price_list** Initial prices for the simulation
- **:weight_shape** Determines the influence of investors in the market
- **:top_weight** Fixes the maximum influence investors have in the market
- **:var_bet_price** Variance of the investors analysis of the market price
- **:strategy_hash** Strategies used by the investors
- **:buy_percent** Percentage at which investors decides to buy
- **:sell_percent** Percentage at which investors decided to sell
- **:init_var_bet_price** Initial variance of prices for the strategies analysis

- **:depression_factor** Influence of depression in the volatility
- **:mood_factor** Influence of mood in the volatility
- **:volume_factor** Influence of volume in the volatility
- **:news_factor** Influence of news in the volatility

4.4.2 Initializing Investors

Once we have set the parameters the next step is to create as many investors as the parameter :ninvestors defines.

Every investor identifies a class of investors and thus, each might have a different influence in the market. For instance we might be in a situation were very few investors have a huge influence in the market or in a situation where every investor has the same influence. In the first case we can expect big jumps in the volatility whereas in the second case every impact will be very smooth.

For each investors the influence is set as follows:

$$1 + ws_0 \cdot (1:N)^{ws_1}$$

Where 1:N is a list of numbers from 1 to N number of investors and the parameter :weight_shape has two values (ws_0 and ws_1), if the first value is 0 we have a constant and then every investors has the same influence, if the first value is positive and the second is zero we have an increasing lineal influence in the markets, and if the second parameter is bigger than one we have an exponential influence in the investors, that is, very few investors will have most of the impact in the market.

The parameters :top_weight determines what is the maximum impact that any investor may have in the market, when this parameters is close to one the impacts will be very important in the volatility and the lower the value the less volatility will be expected.

In Figure 4.12 we can appreciate different weight influence distributions for the values [0,1] (constant), [1,1] (line), [1,2] smooth parabola, [1,5] parabola.

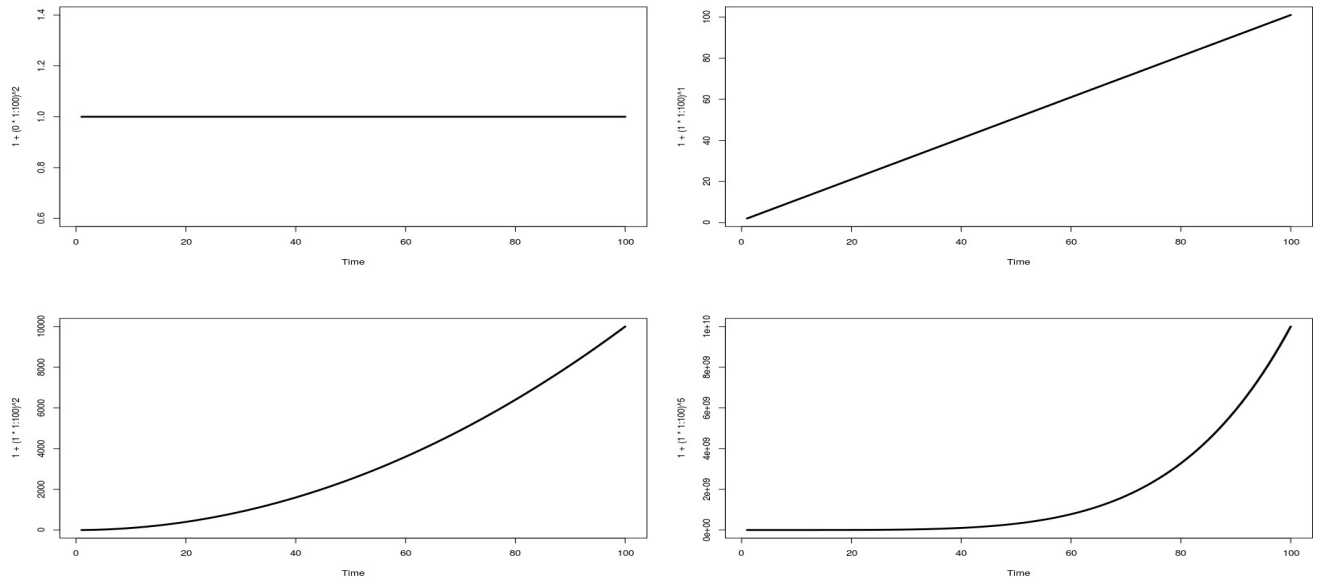


Figure 4.12: Possibles market weight distributions for the investors

Right after determining the weight for each investors is turn for their strategies to be set, as we have seen there are hundreds if not thousands of possible strategies for each investor, to keep the simulation simple enough to be managed every investor will have the same strategy which works as follows:

Every investor makes an assessment of the real value of the stock, in his assessment is above a determined percentage then the investor will buy, and if the assessment is below another determined percentage then the investor will sell. Thus, considering the weights and the prices, every time there is transaction the new price for the stock market is calculated as follows:

*new_stock_price = buyer_weight * buyer_price/2 +*

*seller_weight * seller_price/2 +*

*(1-buyer_weight/2-seller_weight/2) * stock_price*

Though the percentages to buy and sell could be consider different for each investor, again, to simplify the model every investor will have the same purchase and sell percentages.

The initial assessment of the price for each investor could be itself a parameter but to keep the number of parameters low this initial values will follow a normal distribution with variance determined by the parameter :var_bet_price and its mean determined by the initial values of the series introduced by the parameter :price_list. The parameter :var_bet_price affects the initial volatility of the simulation since the further apart the initial assessment from investors the bigger will be the impacts in the price.

4.4.2.1 Investor Strategies

The investor strategies determine how investors update the evaluation for the real price of stock. We saw previously that there are potentially thousands of different strategies for investment. In the project it has been implemented two strategies: **delta** and **var**.

The delta strategy is the simplest and only used for testing purposes, this strategies updates the evaluation every investor have on the price with the following formula:

$$\Delta\ price = price_t - price_{t-1}$$
$$bet.price_{t+1} = bet.price_t + \Delta\ price$$

Where price is the price in the stock and bet.price the evaluation of the real price that the investor does. This is an extremely simple strategy where the investors simple update the value of their best guess with the increase or decrease in the price of the stock.

The var strategies is the one used by all the investors, that again and as mention previously, all investors will follow to simplify the simulation. The var strategy updates the the bet.price with the following formula:

$$s1 = s0 + s.inc$$
$$bet.price_{t+1} = s1/s0 \cdot bet.price + price_t \cdot (1 - s1/s0) + \Delta\ price$$
$$s0 = s1$$

The rational of this process is as follows; we searche for a simple investment strategy that would allow us to manipulate the volatility in the market by simply adjusting it, this is important since effects on the volatility coming from external events like news or the mood of the investors can hardly be simulated. In this circumstances we can only measure the impact more in a classical fashion than anything else.

Since we force as condition that the initial price evaluation of the investors follow a normal distribution, and since the variance of that distribution has a directly proportional relationship with the volatility in the markets, we only need a way to increase the variance of those evaluations in real time, this is how is done.

Since the evaluation follows a Gaussian distribution, and since this kind of distributions belong to the location and scale family, if we want to update its standard deviation from σ_0 to σ_1 we only need to consider the following transformations steps:

$$bet.price \sim N(price, \sigma_0)$$
$$\frac{bet.price - price}{\sigma_0} \sim N(0,1)$$
$$\sigma_1 \cdot \left(\frac{bet.price - price}{\sigma_0}\right) + price \sim N(price, \sigma_1)$$
$$\frac{\sigma_1}{\sigma_0} bet.price + price\left(1 - \frac{\sigma_1}{\sigma_0}\right) \sim N(price, \sigma_1)$$

So now we only need to update **s.inc** with any external phenomena to influence the volatility in the market. Finally the bet.price is update with the increase in the price to keep the distribution of evaluations centered around the last price in the market.

4.4.3 Initializing Events

Once the investors have been initialized is turn for the external events, the values for the depression, mood and volume are loaded into the simulation and, in every step, they will update the s.inc parameter to affect the simulation volatility. There is another source of external events that affect volatility; the news. The difference between depression, mood, volume and the news events is that the news only affect volatility temporarily. Let's see how this is done.

For these three external events the following procedure is followed to update the s.inc value to affect volatility in the market.

$$\Delta event = event_t - event_{t-1}$$
$$s.inc = s.inc + factor \cdot \Delta event$$

so the factor determines in each case how much influence the increase of decrease of the event will have and the sign will determine if the effect is positive or negative. In the case of the news events it works differently, in this case we have

$$s.inc = s.inc + factor \cdot news_t$$

In the fitting process performed in this project it has been used a set of news impacts starting in 2008-11-07[35] with a value of 16 and then decreasing its value by half in sixteen steps.

Introducing news events is important since some behavior in the volatility can only be explained by an atypical event and its appearance affects the resulting simulation as well as the later estimation of parameters when fitting a time series.

35 These dates follow roughly the crisis calendar as describe by the Washington Post at http://www.washingtonpost.com/wp-srv/business/economy-watch/timeline/index.html

4.4.4 Transactions and Prices Updates

Now everything is ready to let the simulation performed transactions. Every event in the simulation is stored and ordered by date. The simulation goes through all this events updating the parameters of the model accordingly, as explained before, every time the simulation finds an event depression, mood, volume or news it proceeds to update the volatility of the transaction process.

The main to events that drive the simulation are prices updates and investors transactions. Once the initial prices are set the simulation proceeds to select two investors.

The selection of the two investors is done as follows:

1. Create a list with potential buyers among those whose evaluation of the market price is above the :buy_percent parameter.
2. Create a list with potential sellers among those whose evaluation of the market price is below the :sell_percent parameter.
3. Select from the lists one buyer and seller randomly

In case no buyers or sellers are found due to extreme values two investors are chosen randomly to accommodate to the new price in the market.

Once we have the two investors the negotiate the transaction according to the following formula:

$$transaction.price = (buyer.bet.price - seller.bet.price)/2.0 + seller.bet.price$$

That is, the reach an agreement on the mid point between their two estimations for the real price. This new transaction is pondered in the market as explained previously.

Once the new price in the market is updated all investors are informed and they recalculate the new real value of the stock upon their personal strategies. Once every investors have updated the estimation a new bidding process begins and this process continues until the end date of the simulation.

In Figure 4.13 we can see a diagram describing succinctly all the steps described previously

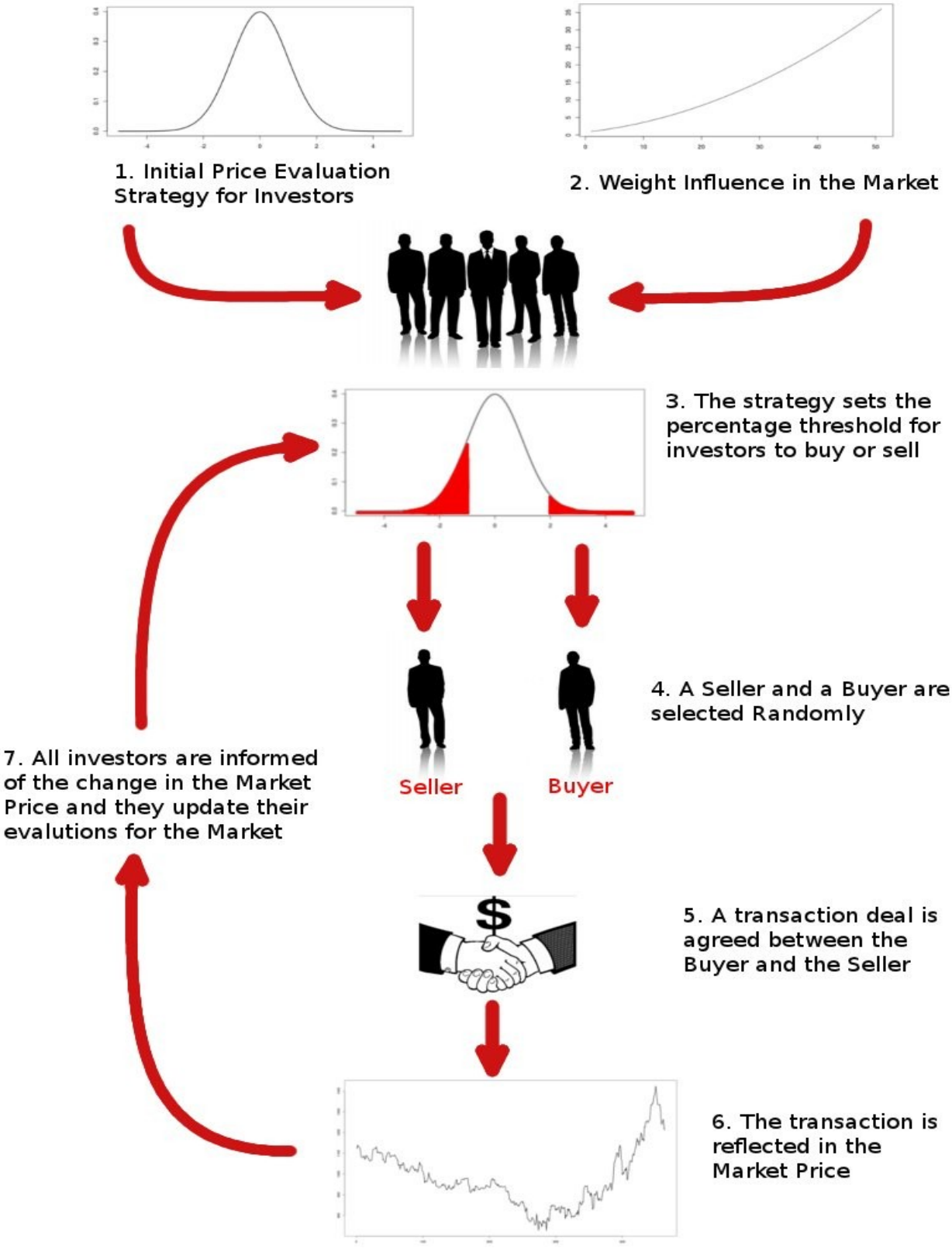


Figure 4.13: Diagram for the Basic Steps of a Simulation

4.5 Architecture

In the following figure it can be seen an UML class diagram showing the basic architecture of the simulator that it has been develop for this project.

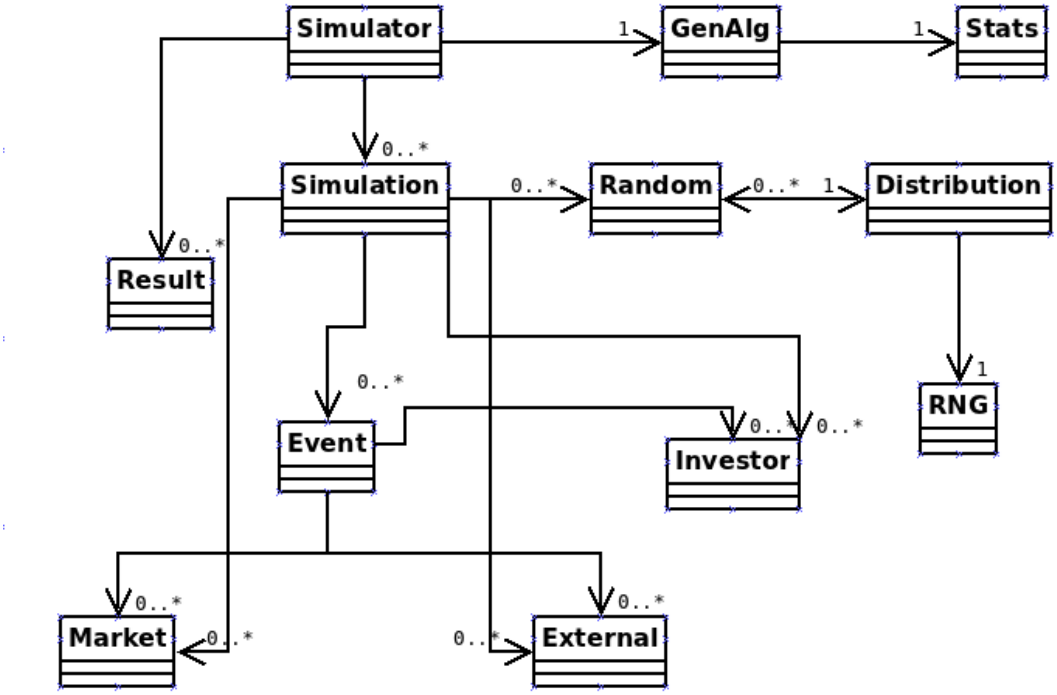


Figure 4.14: Class Diagram for the project Simulation

This simulator class is the one controlling all the processes from the simulation itself to *GenAlg* class which is a genetic algorithm used to fit the simulation parameters to a particular time series. The class *Stats* calculate basic statistical operations related to confidence intervals. The class *Result* is in charge to analyze statistically the results coming from every simulation. The classes *Market*, *External* and *Investor* are the actors that take actions within the simulation via events and, finally, all events are represented by the class *Event* and controlled by the class *Simulation*.

5 FITTING PARAMETERS

Once we have a working simulator we can use it to characterize a particular time series, in this project we are working with the S&P 500 time series so the next step is to figure out what values we should give to the parameters discussed previously so that the simulation volatility behaves the closest to the volatility shown in the S&P 500.

Despite the simplicity and relaxations of the model we need nonetheless to set 14 parameters for the simulation to characterize a time series. That is computationally challenging not just for the large number of parameters to be estimated but because every simulation takes several seconds to be executed and, since we are working with a stochastic phenomenon, every simulation with the same parameters will not return the same results. This means that in order to calculate a fitting value for a particular set of parameter we need to execute several simulations per set of parameters to estimate their values within a confidence interval and compare those estimations in the fitting process.

There is no classical mathematical method to properly optimize the fitting of the parameters of this kind of simulation. Methods like Nelder-Mead o Pseudo-Newtonian need a fix value for a fix point which is not the case for this problem; for each point we have range. Besides those methods only guarantee the optimum locally whereas in these simulation is feasible to have different points for the same optimum value.

Fortunately there is one way to tackle this situations; Genetics Algorithms. These algorithms are search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

These algorithms store a family of points (chromosomes) for which a fitting value is calculated, then inheritance, mutations and crossover happen randomly on those chromosomes discarding in each generation the less fit chromosomes and moving forward the fittest.

This algorithm is particularly suited for this project because not only guarantees the optimum value in the long run, but a family of values which is something more than useful in a problem where different and not necessarily similar points (chromosomes) might be equally optimal.

The fitting procedure has the following steps:

1. Finding an appropriate initial chromosome and range for its values
2. Scan rapidly optimal values with a small number of simulations to estimate the value of the chromosome (In the simulation were used 5)
3. Use the optimal in step 2 as a starting chromosome for a new search with a larger number of simulations per chromosome to achieve a further refine results with smaller confidence intervals. (In the simulation were used 16)

Given the nature of this problem the more simulations the better since we can never guarantee we have the optimal chromosome, just that eventually we will get it. That is why the decision to stop the fitting process for his project is based on not achieving any further improvement in the chromosome fittings after a few thousands of simulations.

The fitting function for the simulation is the following:

$$q1 + q3 + var + (lm.intercept + lm.trend + ar2 + ar3 + ar4) \cdot 0.1$$

where each parameter is a scaled difference between the statistic calculated in the S&P 500 time series and the time series generated by the simulator. The parameters represent the following scale differences:

- **q1,3:** First and third quantile for the returns
- **var:** Variance of the returns
- **lm.intercept:** Intercept from the linear regression of the time series.
- **lm.trend:** Trend from the linear regression of the time series.
- **ar 1,2,3:** First three significant values from the returns ACF.

The parameters q1, q2 and var are given higher weight in order to keep the optimization locked to a good fitting of the variance while improving the other parameters. This is important since the risk is highly associated to the variance whereas the other parameters affect the overall behavior of the time series.

To scale the parameters so that the would have the same influence in the fitting function the following formula was used to compare the statistic calculated from the simulation and the one from the S&P 500 time series:

$$2 \cdot \frac{|a-b|}{|a|+|b|}$$

where **a** and **b** are the two values that we want to compare to see how close are to each other. This way we can use in the same fitting function parameters with very different orders of magnitude.

6 FORECAST

The fittest chromosome is chosen after the Genetic Algorithm cannot find a better chromosome for a number of generations, next we can see a plot with the evolution for the fittest values for all chromosome evaluation.

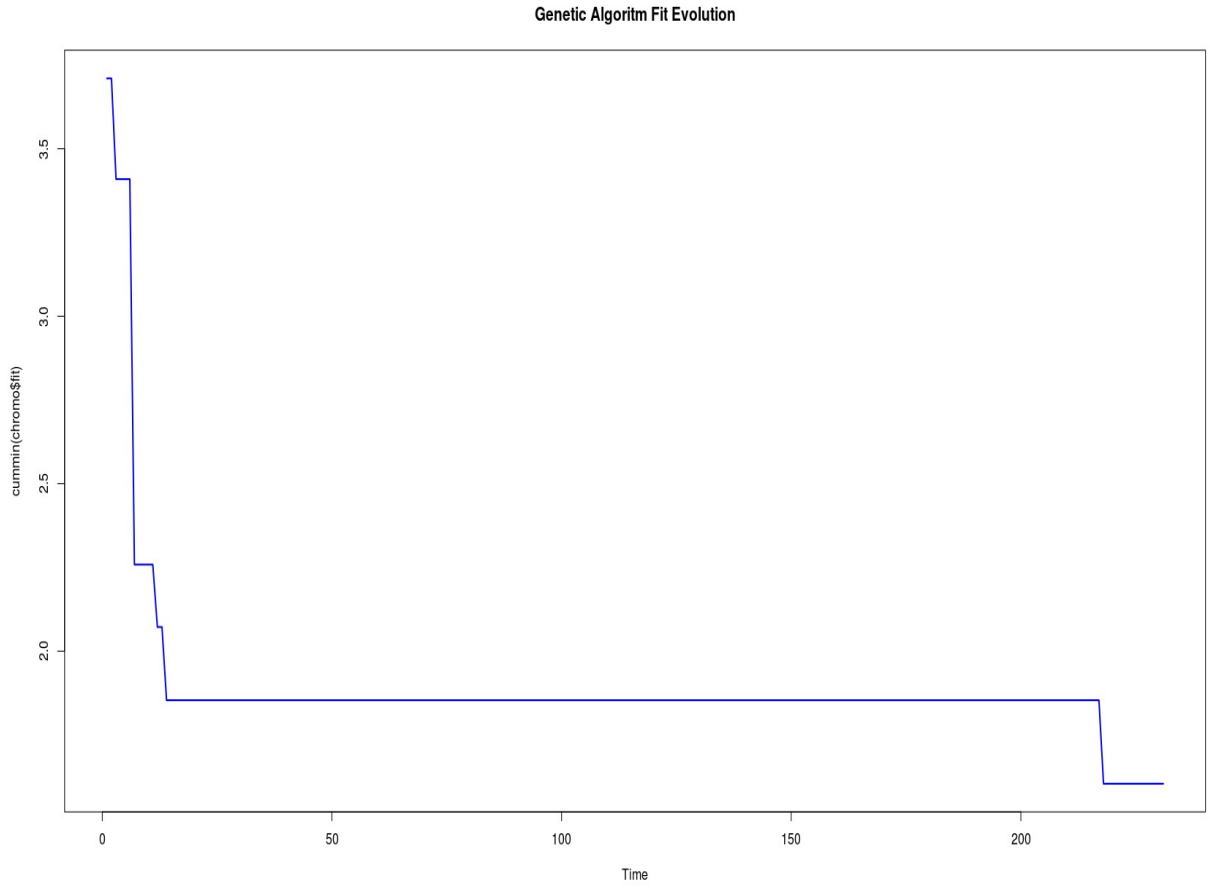


Figure 6.1: Genetic Algorithm Fittest Value Evolution

Once we have a optimal chromosome that fits the time series that we want to analyze is time to do a forecast based on that chromosome. We only need to set the end date for the forecast, set the optimal chromosome in the simulation and simulate as many times as required for the ulterior analysis.

Since it was calculated previously the risk measures for a long position of a year, we will run 200 simulations for an extra year from the last value in the time series S&P 500 until 2012-11-13.

We only need one more thing before we do the forecast, since the volatility is driven not just by the parameters of the model, but for the value of the external events depression, mood, volume and news, we need to make a forecast of those values to sensibly make a forecast with the simulator. Fortunately the depression, mood and volume can be foreseen using ARIMA models, the news can somehow be foreseen, fore example, we might know the date a Central Bank is going to issue interest rates, that would allow us to treat different scenarios upon the nature of those news. Nonetheless for this project only the ARIMA fittings have been done to drive the forecast volatility.

Once the forecasting is done, we will be able to place the new forecasting data with the files containing the actual data in order to perform simulations beyond the current date and gather the results for ulterior statistical analysis.

We can see next the ARIMA fitting and validation process for each external parameter:

6.1 Depression

Let's have first a preview of the depression time series and its ACF and PACF:

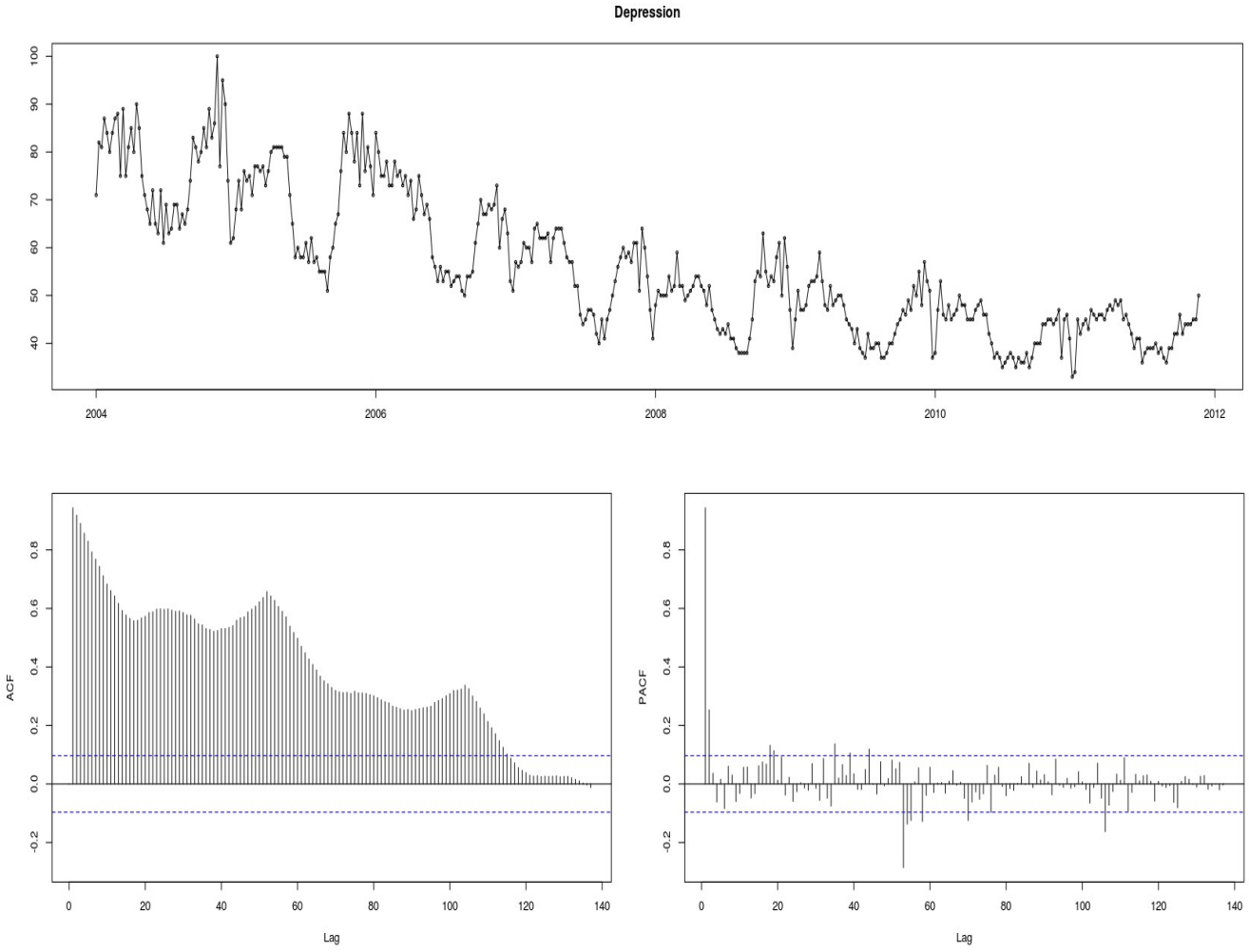


Figure 6.2: Depression Time Series with ACF and PACF

The slow decay in the ACF shows a likely unit root in the time series, but in case no unit root was found the ACF and PACF suggest an underlaying AR(2) model.

The depression time series also seems to show some seasonality behavior, to test this we can analyze a plot of the time series grouped seasonally, in this case the seasonality period will be of 52 since we have the depression data group weekly.

In the next figure we can see the plot of the time series grouped seasonally:

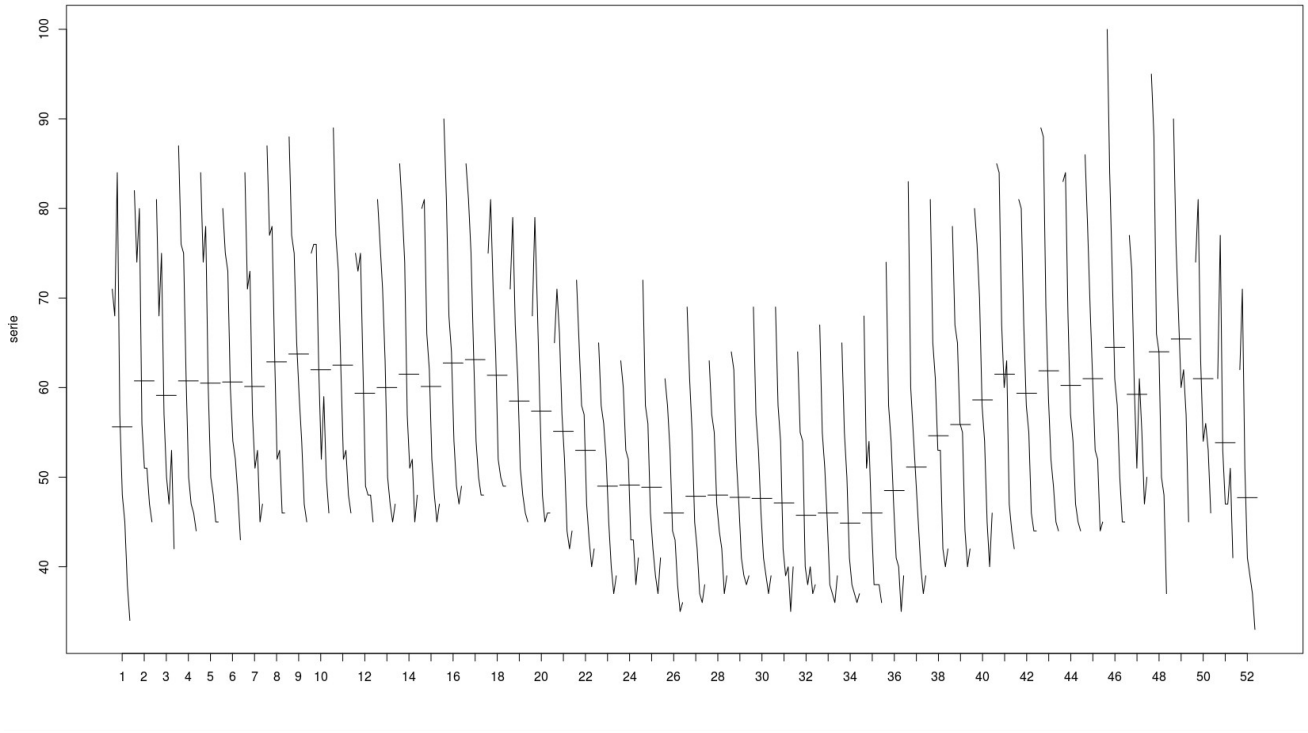


FIgure 6.3: Seasonally grouped Depression time series

We can clearly see that in the in the weeks around summer time the level of depression falls clearly respect the remaining weeks, there is also a clear low level in the peaks previous to New Year.

With these results in mind we apply a Box-Cox transformation to the time series to stabilize the variance and make the time series more normal distribution-like, as well as to improve the Pearson correlations. If we now check the transformed time series for unit roots we obtain the following results:

Augmented Dickey-Fuller Test
Test Results:
PARAMETER:
Lag Order: 1
STATISTIC:
Dickey-Fuller: -0.396
P VALUE:
0.4901

Table 6.1 Augmented Dickey-Fuller Test for the BoxCox Depression time series

Which clearly shows we cannot reject the hypothesis that there is an unit root as the underlaying cause for the time series behavior.

Since we have seen that the time series also shows a clear seasonal pattern we can apply first a differentiation of order 52 to solve the seasonality behavior and check if this differentiation also solves the unit root problem. Next we can see the time series after applying the Box-Cox transformation and a differentiation of order 52:

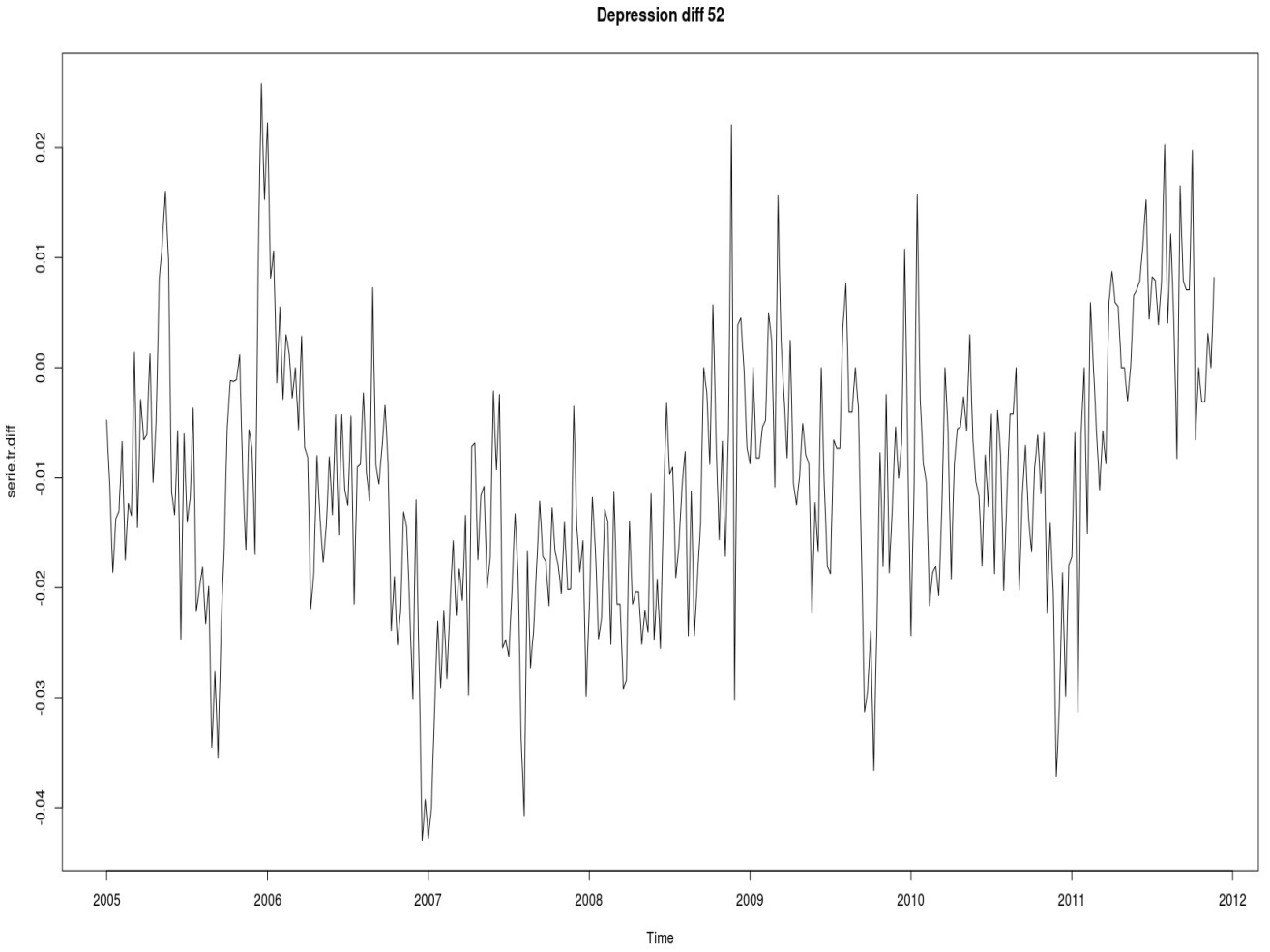


Figure 6.4: Box-Cox Transformation and differentiation of order 52 for Depression

For this case turns out that continuing with a regular differentiation, despite reducing the variance in the transform time series, leads to ARIMA models that offer poor forecasting when considering one year windowed time series, the extra differentiation also caused the model to be less stable when recalculating it with the full data due to the fact that one parameter was lost after differentiating.

Therefore no more differentiating is needed and we can proceed to analyze the ACF and PACF of the transformed time series to search for candidates models. In the next figure we see can such ACF and PACF.

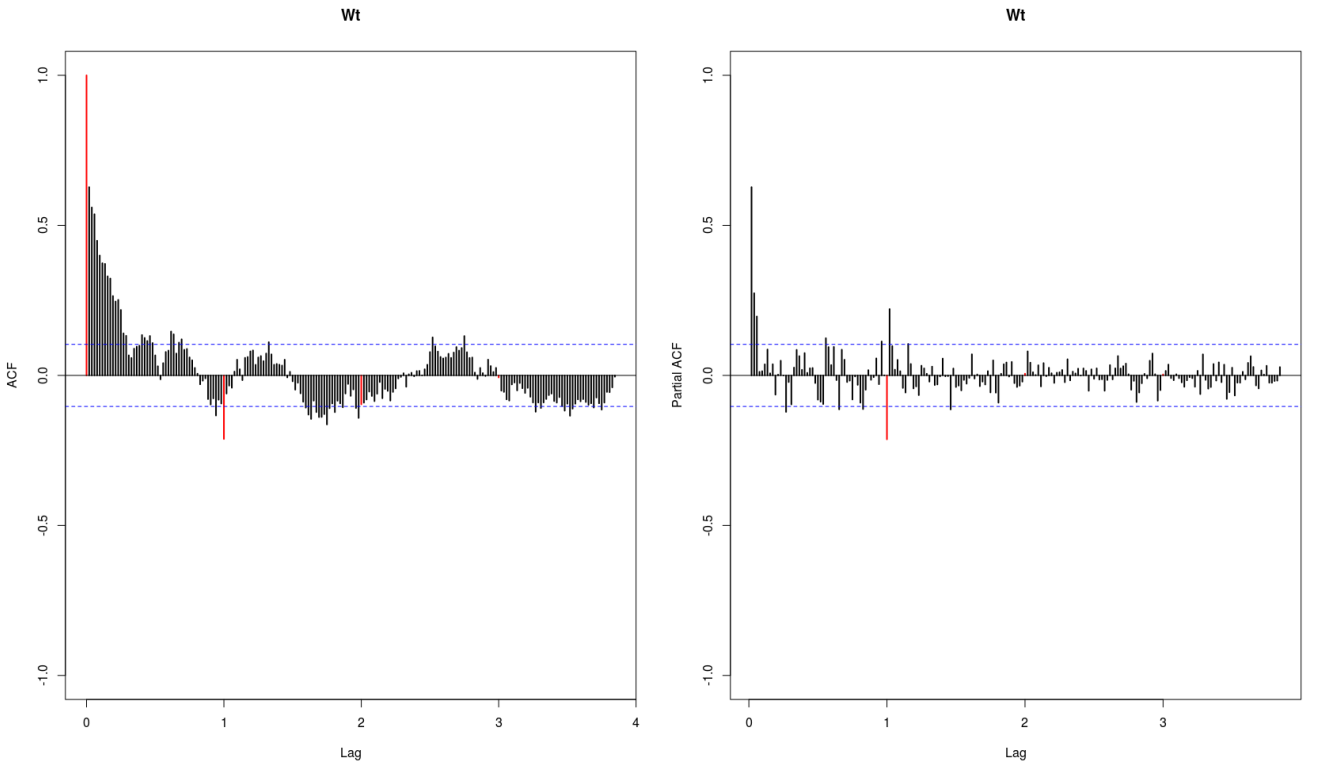


Figure 6.5: ACF and PACF for transformed Depression time series

We can observe an exponential decreasing in the regular part of the ACF plus three significant values in the PACF which hints towards and AR(3) model. In the seasonal part of the ACF and PACF we can consider either a AR(1) or an MA(1). After the comparison among many models, the model with better fittings and forecasting properties is an ARIMA(3,0,0)(1,0,0)[52] with the following parameters:

ARIMA(3,0,0)(1,1,0)[52]

Coefficients:

	ar1	ar2	ar3	sar1
	0.5096	0.2728	0.2177	-0.4040
s.e.	0.0158	0.0164	0.0135	0.0123

sigma^2 estimated as 6.744e-05: log likelihood=1209.3

AIC=-2408.61 AICc=-2408.44 BIC=-2389.19

Table 6.2 ARIMA model for the Depression

If we now use the Ljung-Box test to validate the residuals of the previous model for unaccounted correlations. We can see that there are no correlations unaccounted for and the fitting is good:

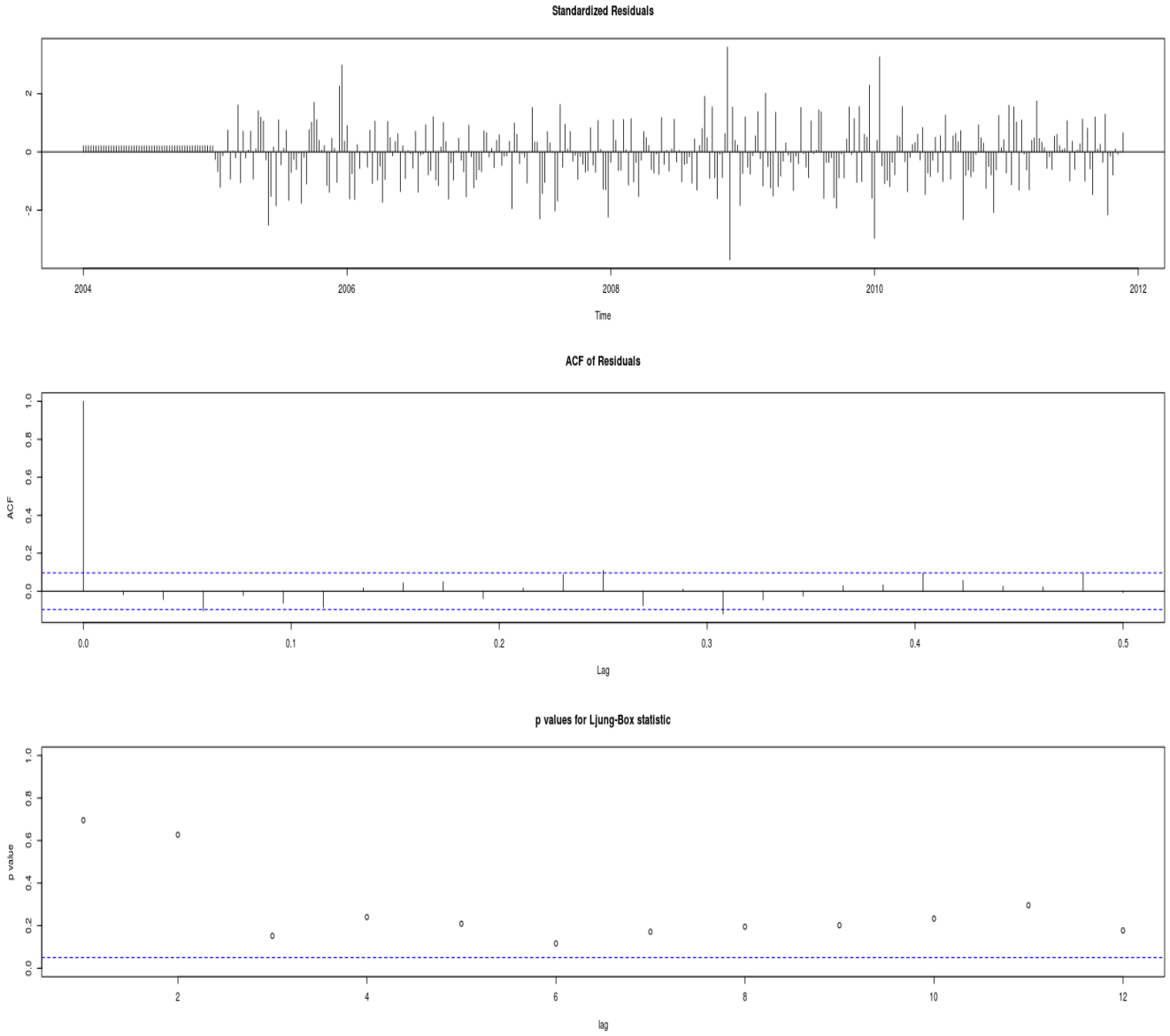


Figure 6.6: LJung-Box test for the Depression ARIMA model

Now we can finally check the stability of the model by calculating a forecasting on the same data by trimming the time series by one year, refitting the ARIMA model with the windowed data and comparing the forecasting of the model with the real data, the results of such comparison can be seen in the next figure:

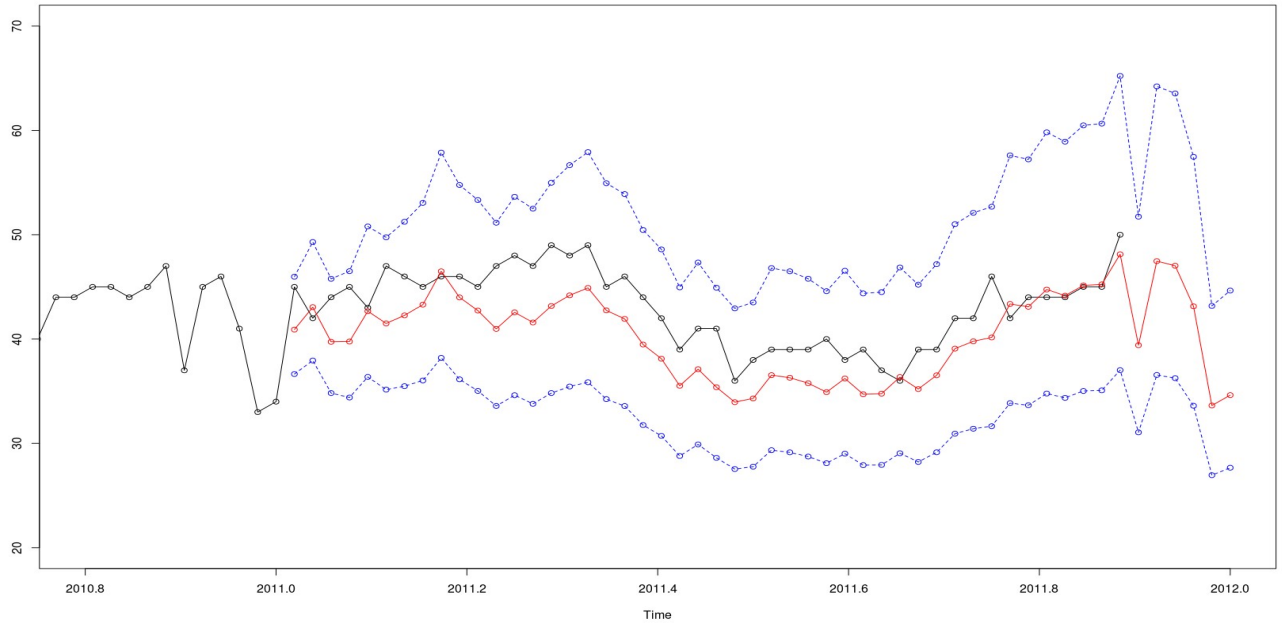


Figure 6.7: Forecasting for a one year windowed Depression time series

The blue lines are the confidence intervals at 95% whereas the red line is the forecast done with the model. We can appreciate an excellent fitting which can make us feel confident about the stability of the model for future forecast, In the next figure we can see the forecast for the next year with all the data available from the depression time series.

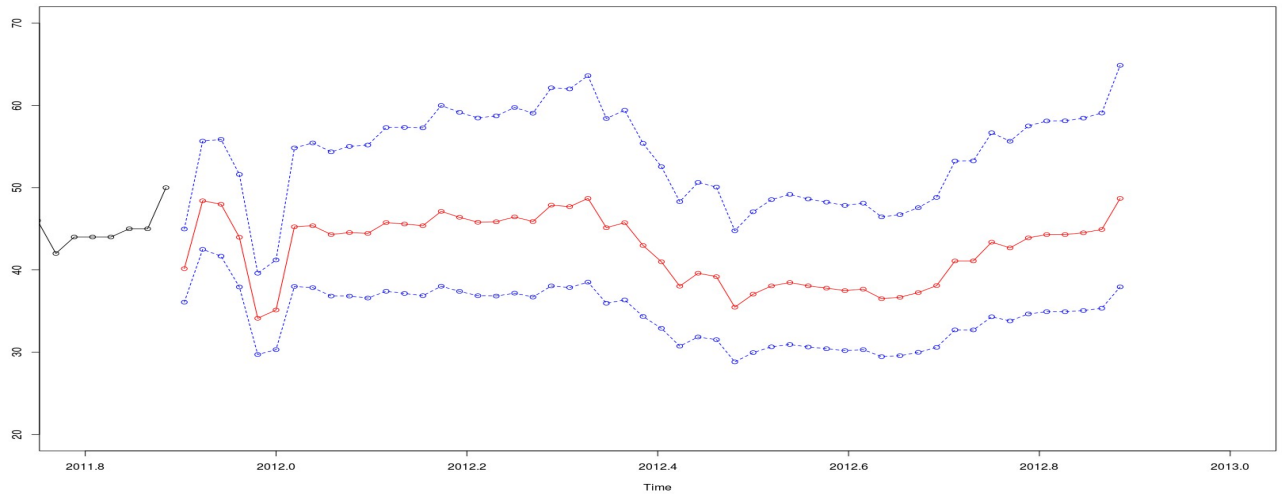


Figure 6.8: Forecasting for Depression for the next year

We can appreciate that for the next year forecasting in 20120 depression values will oscillate but show no signs of increasing or decreasing keeping overall the same values.

6.2 Mood

Let's now see the results for the ARIMA fitting of the time series Mood.

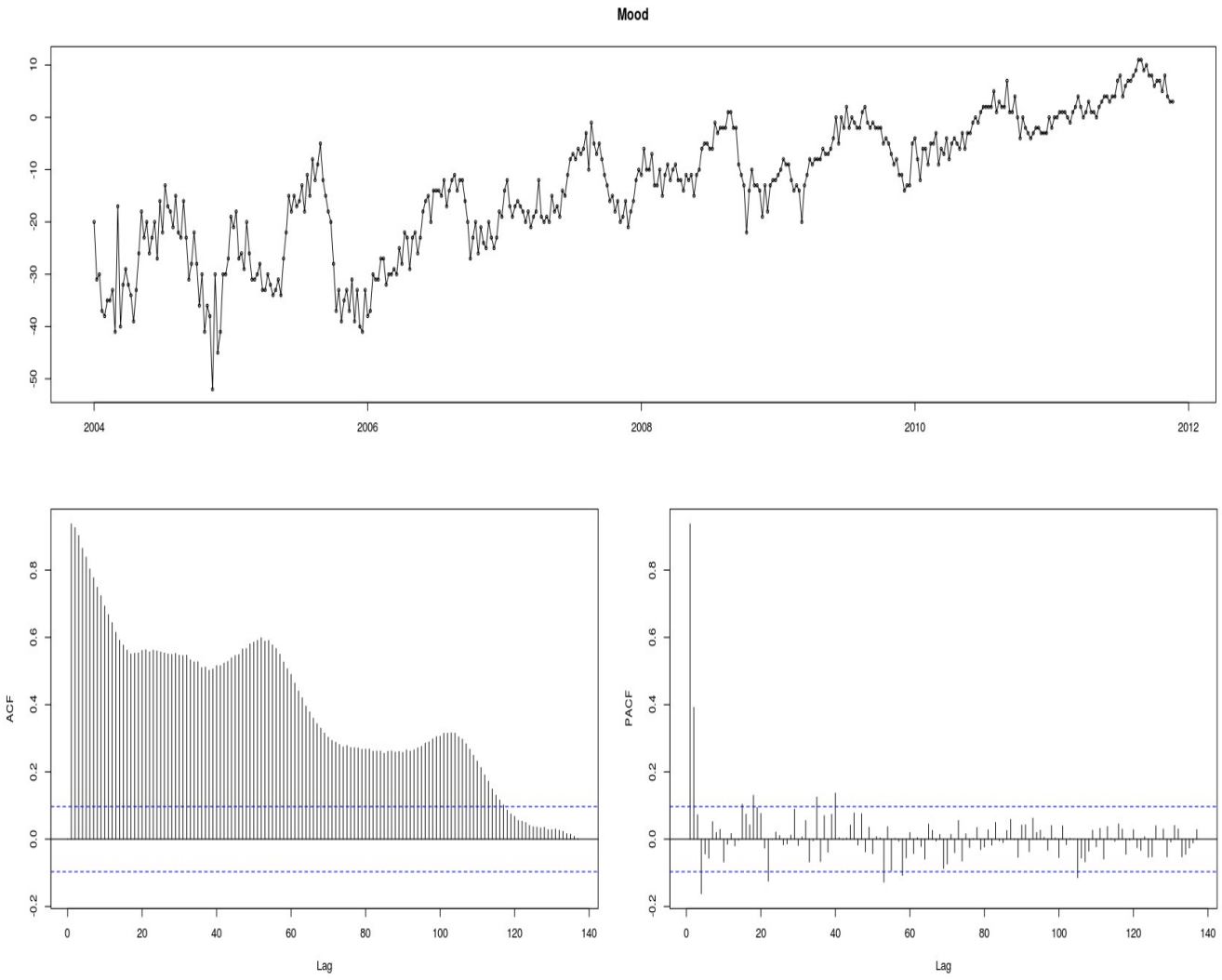


Figure 6.9: Mood Time Series with ACF and PACF

We can appreciate again a slow decay in the ACF showing a likely unit root in the time series, the ACF and PACF also suggest an underlaying AR(2) model.

The Mood time series also seems to show some seasonality behavior, and again, to test this we can analyze a plot of the time series grouped seasonally, in this case as, for any time series in this project since we have the data group weekly, the period will be of 52.

In the next figure we can see the plot of the time series grouped seasonally:

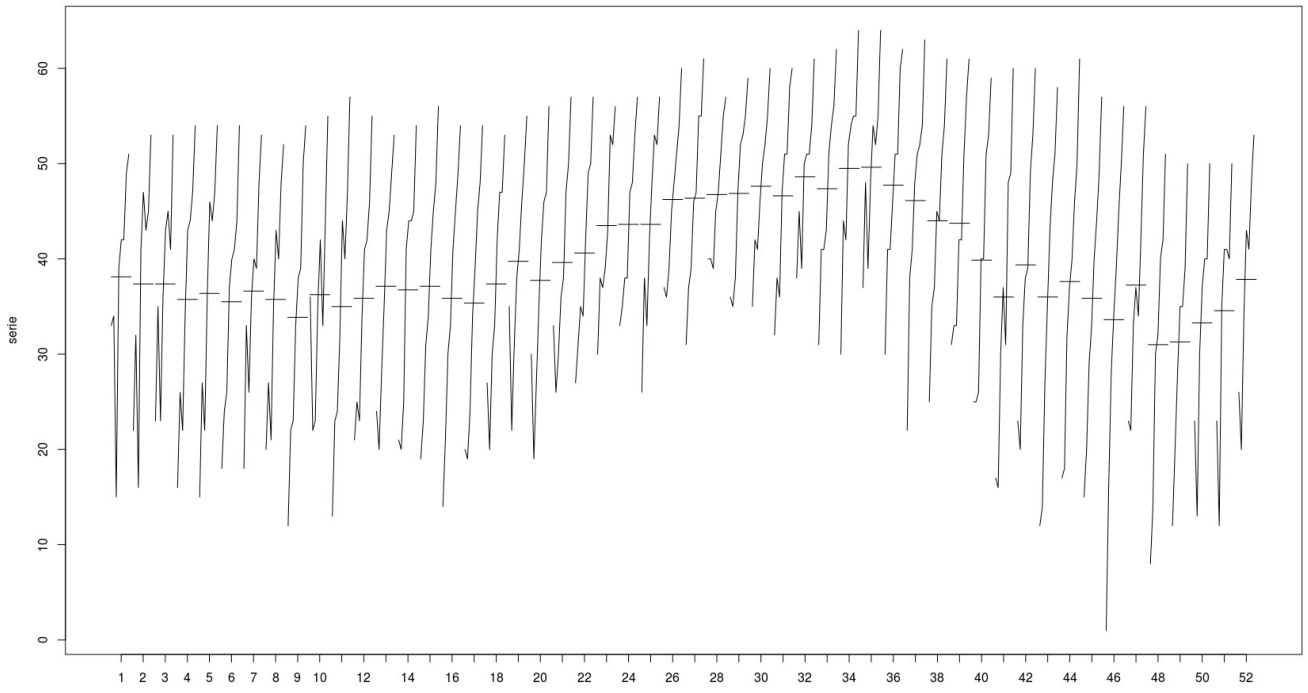


Figure 6.10: Seasonally grouped Mood time series

We can again see clearly that in the in the weeks around summer time the level of mood raises clearly respect the remaining weeks, and there is also a clear raised level in the peaks previous to New Year.

With these results in mind we apply a Box-Cox transformation to the time series to stabilize the variance and make the time series more normal distribution-like, as well as to improve the Pearson correlations. If we now check the transformed time series for unit roots we obtain the following results:

Augmented Dickey-Fuller Test
Test Results:
PARAMETER:
Lag Order: 1
STATISTIC:
Dickey-Fuller: -0.0985
P VALUE:
0.585

Table 6.3 ***Augmented Dickey-Fuller Test for the Box-Cox Mood time series***

These results show we cannot reject the hypothesis that there is an unit root as the underlaying cause for the time series behavior.

Since in this case we also have a clear seasonal pattern we can also apply first a differentiation of order 52 to solve the seasonality behavior and check if this differentiation also solves the unit root problem. For this particular case an extra differentiation is needed for a better fit. Next we can see the time series after applying the Box-Cox transformation and a differentiation of order 52 and a regular differentiation:

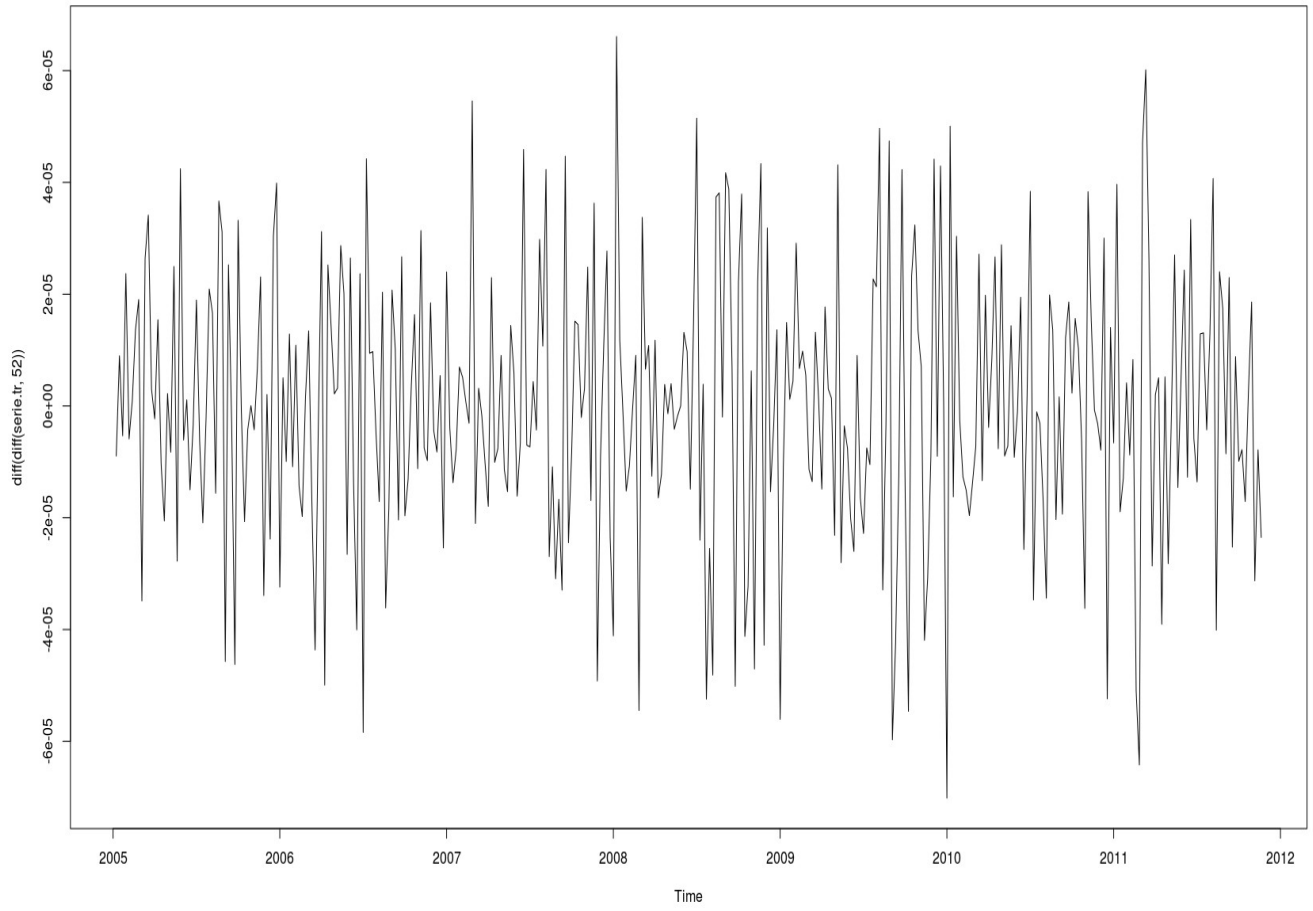


Figure 6.11: Box-Cox Transformation with seasonal and regular differentiation for Mood

After analyzing several ARIMA models the best fit for the Mood time series was given by one seasonal differentiation and one regular differentiation.

Now we can proceed to analyze the ACF and PACF of the transformed time series to search for

candidates models. In the next figure we can such ACF and PACF.

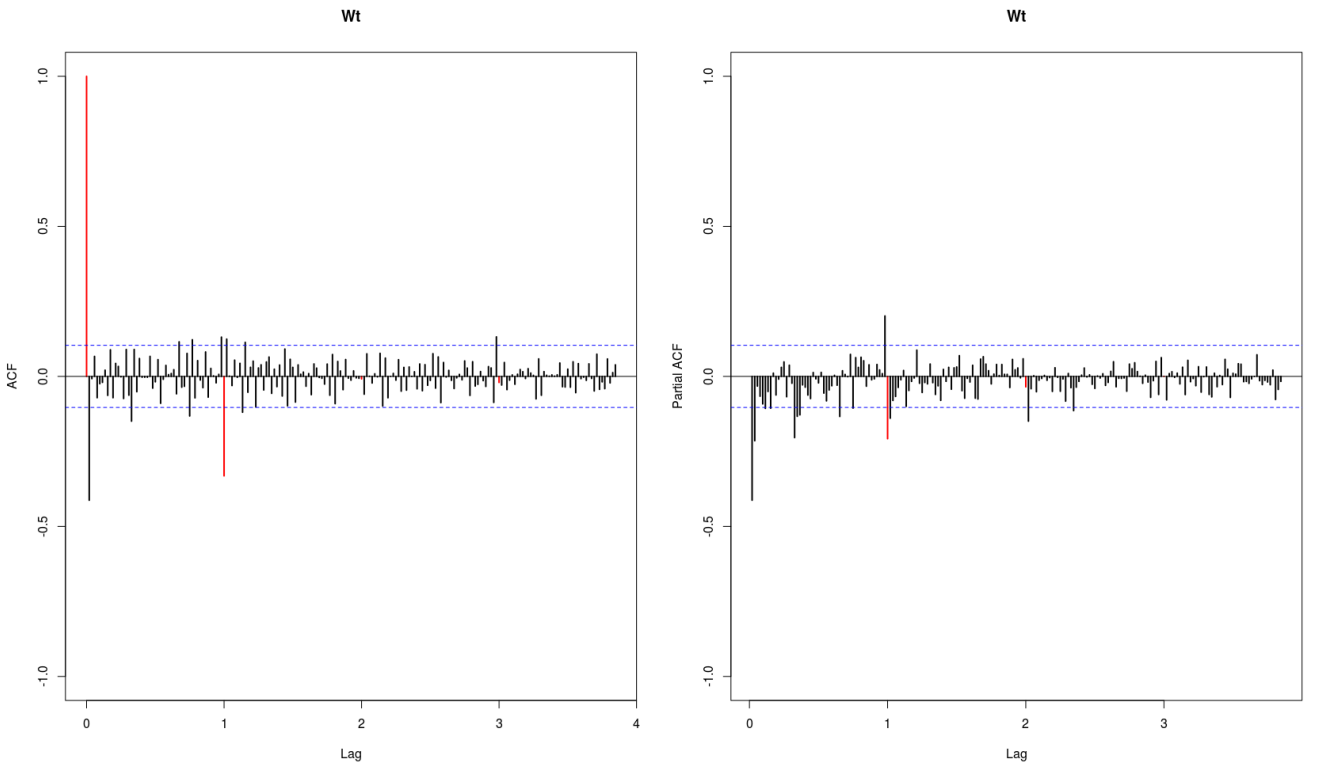


Figure 6.12: ACF and PACF for transformed Mood time series

In this case we can try several ARIMA models to fit the time series, after trying several options the one that gave a better fitting and forecasting behavior was an MA(2) for the regular part and and MA(1) for the seasonal part of the time series. After fitting the ARIMA(0,1,2)(0,1,1)[52] we obtain the following results:

ARIMA(0,1,2)(0,1,1)[52]
Coefficients:
ma1 ma2 sma1
-0.5940 0.0274 -0.6110
s.e. 0.0569 0.0704 0.0709
sigma^2 estimated as 98915: log likelihood=-2249.35
AIC=4506.71 AICc=4506.84 BIC=4521.68

Table 6.4 ARIMA model for the Mood

Now again we use the Ljung-Box test to validate the residuals of the previous model for unaccounted correlations. In this case we also see that there that the fitting is good enough:

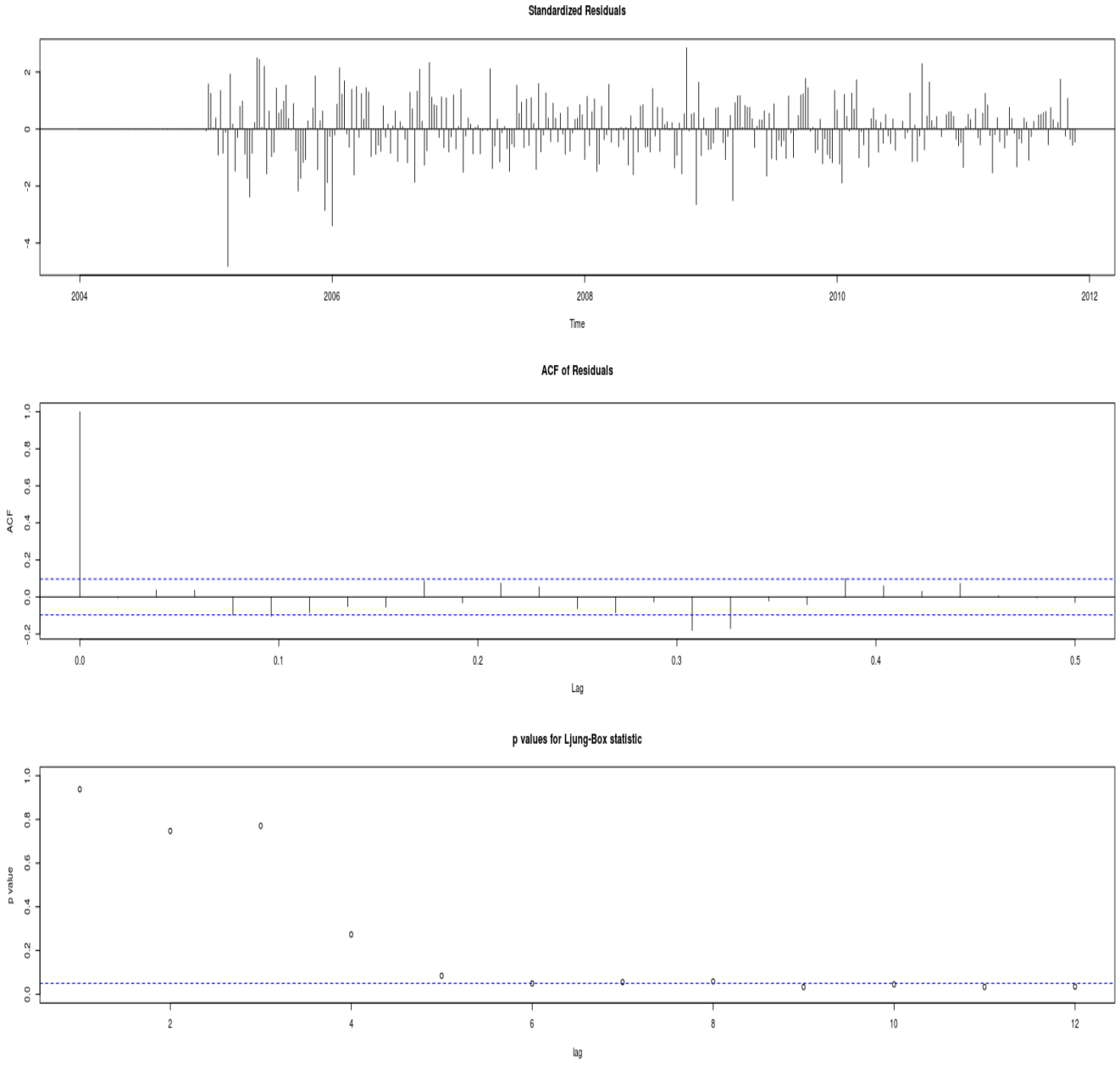


Figure 6.13: LJung-Box test for the Mood ARIMA model

Now, as we did previously with the depression times series, we can check the stability of the model by calculating a forecasting on the same data by trimming the time series by again one year, refitting the

ARIMA model with the windowed data and comparing the forecasting of the model with the real data, the results of such comparison can be seen in the next figure:

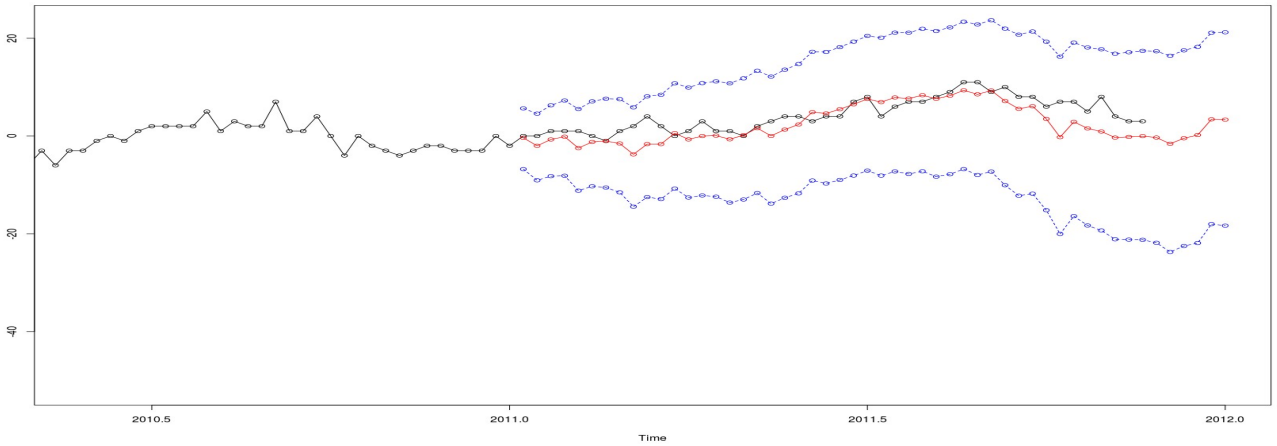


Figure 6.14: Forecasting for a one year windowed Mood time series

The blue lines show again the confidence intervals at 95% whereas the red line is the forecast done with the model. We can appreciate a really good fitting which can make us feel confident about the stability of the model for future forecast, In the next figure we can see the forecast for the next year with all the data available from the mood time series.

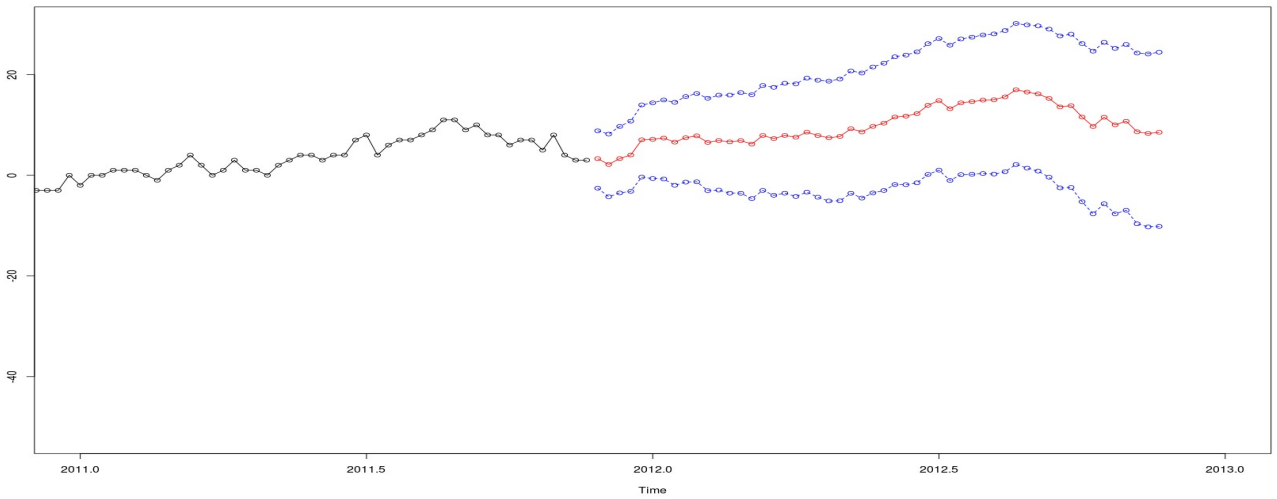


Figure 6.15: Forecasting for Mood for the next year

In this case we can see that for the next year 2012 we might expect an steady increase in the levels of the parameter mood which, considering the stable behavior of the forecast for the depression, this is increase is driven by higher levels of anxiety.

6.3 Volume

And finally let's now the results for the ARIMA fitting of the time series Volume.

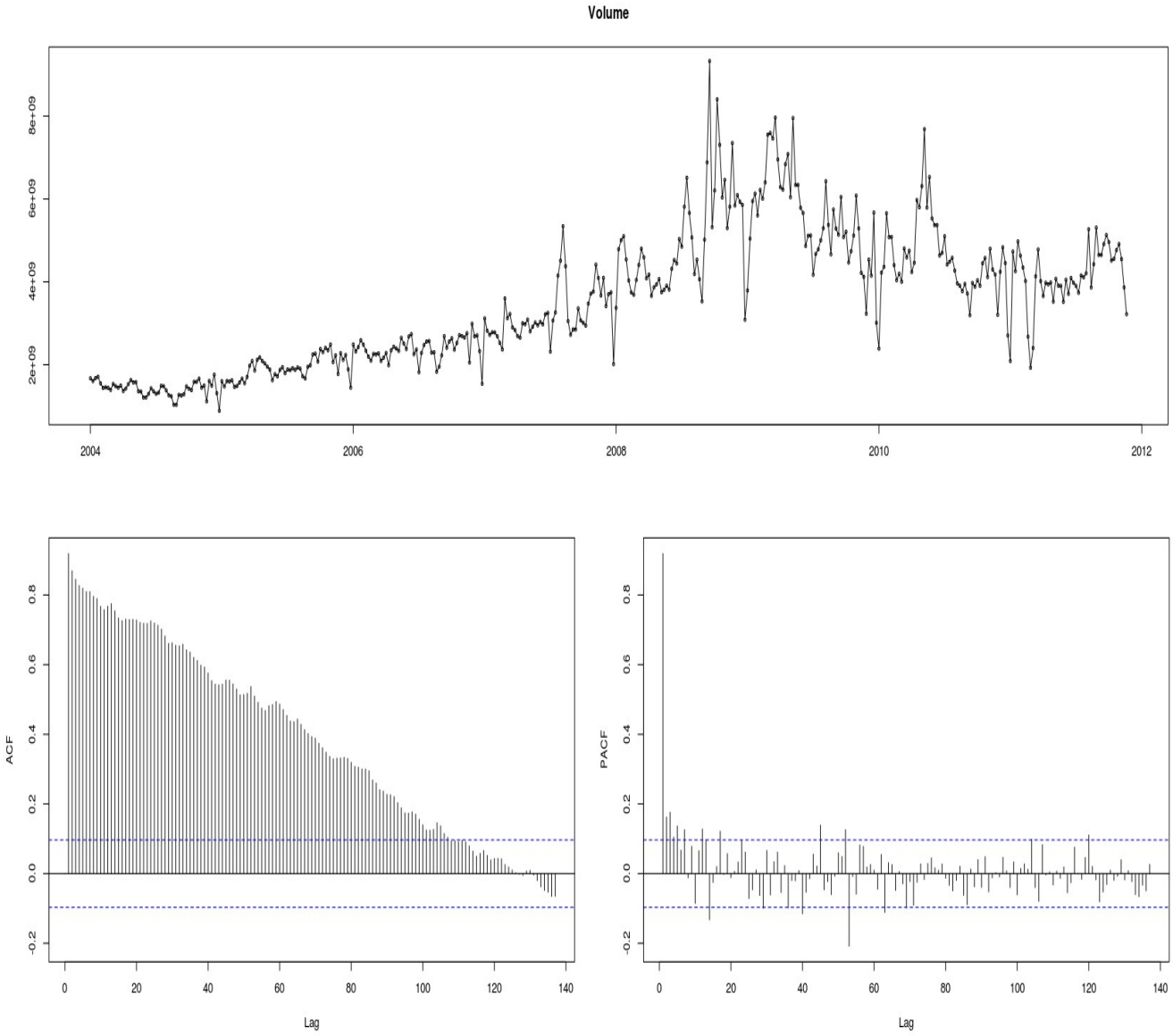


Figure 6.16: Volume Time Series with ACF and PACF

As we the depression and mood time series we can appreciate a slow decay in the ACF showing a likely unit root, the ACF and PACF also suggest an underlaying AR(3) or AR(4) model.

The volume time series also seems to show some seasonality behavior, and again, to test this we can analyze a plot of the time series grouped seasonally, in this case as, for any time series in this project since we have the data group weekly, the period will be of 52.

In the next figure we can see the plot of the time series grouped seasonally:

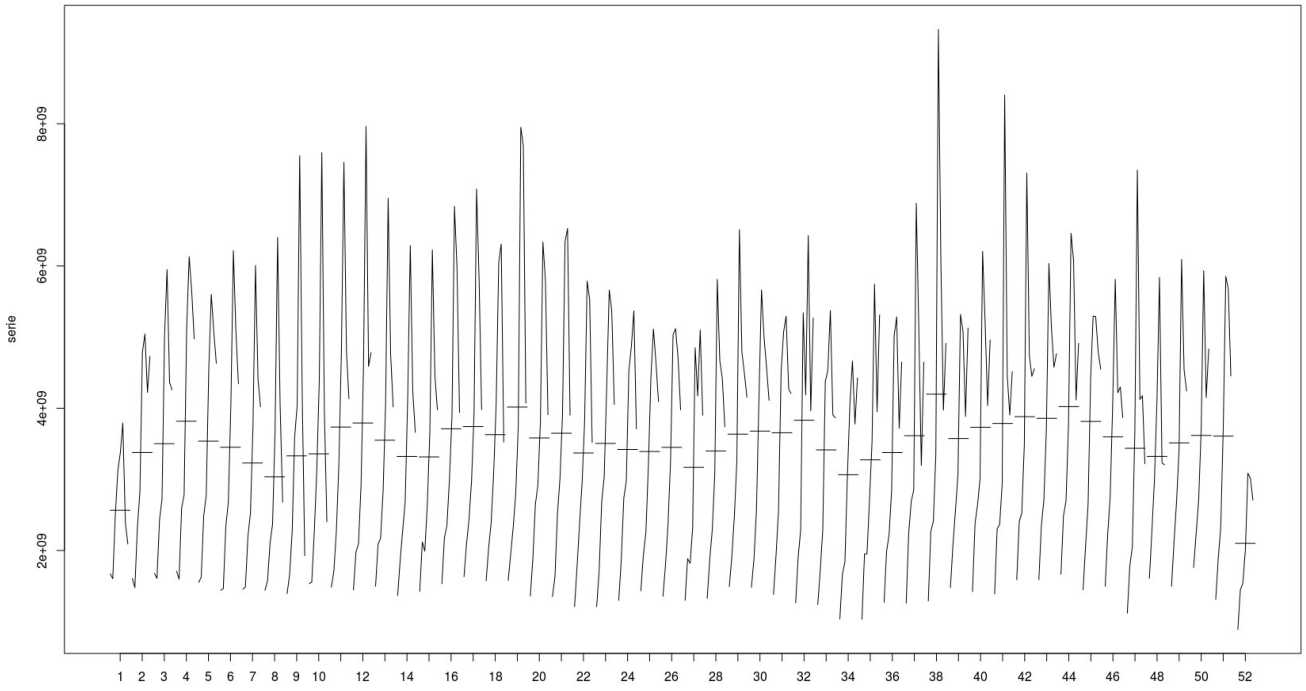


Figure 6.17: Seasonally grouped Volume time series

In this case the seasonality is not so clear and the fittings performed considering it were not better than those ignoring it. So for the volume time series it is better to just treat it as a time series with no seasonality.

Again with these results in mind we apply a Box-Cox transformation to the time series to stabilize the variance and make the time series more normal distribution-like, as well as to improve the Pearson correlations.

If we now check the transformed time series for unit roots we obtain the following results:

Augmented Dickey-Fuller Test
Test Results:
PARAMETER:
Lag Order: 1
STATISTIC:
Dickey-Fuller: 0.2958
P VALUE:
0.7106

Table 6.5Augmented Dickey-Fuller Test for the BoxCox Volume time series

These results show, as with the previous times series, that we cannot reject the hypothesis that there is an unit root as the underlaying cause for the time series behavior.

Since in this case we do not have a clear seasonal pattern there is no need to apply a seasonal differentiation of order 52. So in order to solve the unit root a simple regular differentiation will suffice for this case. Next we can see the time series after applying the Box-Cox transformation with a regular differentiation:

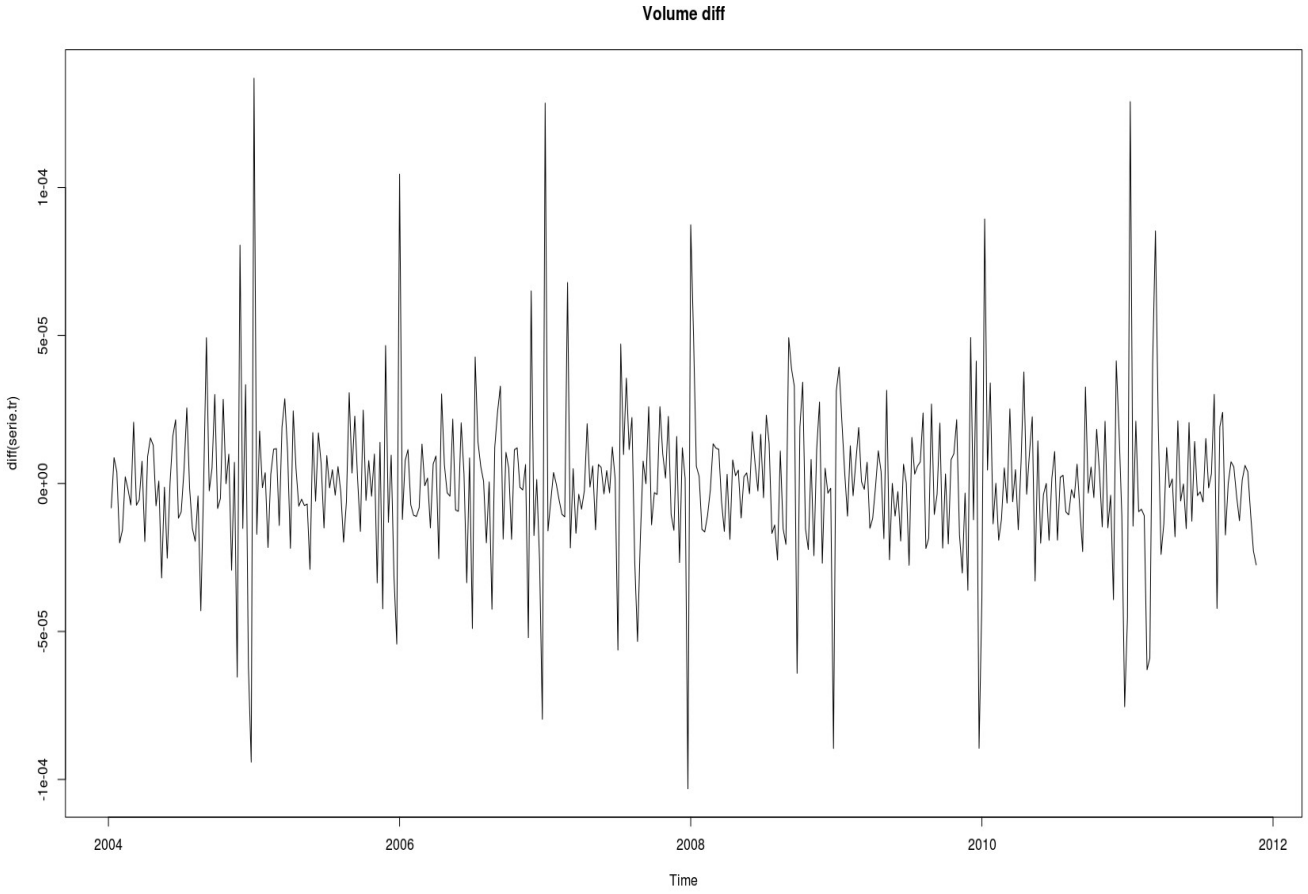


Figure 6.18: Box-Cox Transformation with a regular differentiation for Volume

The several ARIMA models analyzed showed that the best fit for the volume time series was given by a simple regular differentiation, extra differentiations or seasonal differentiations offered no real improvement in the forecasting making in contrast the model less stable when considering the year window forecasting analysis.

We can now finally proceed to analyze the ACF and PACF of the transformed time series to search for candidates models. In the next figure we can see such ACF and PACF.

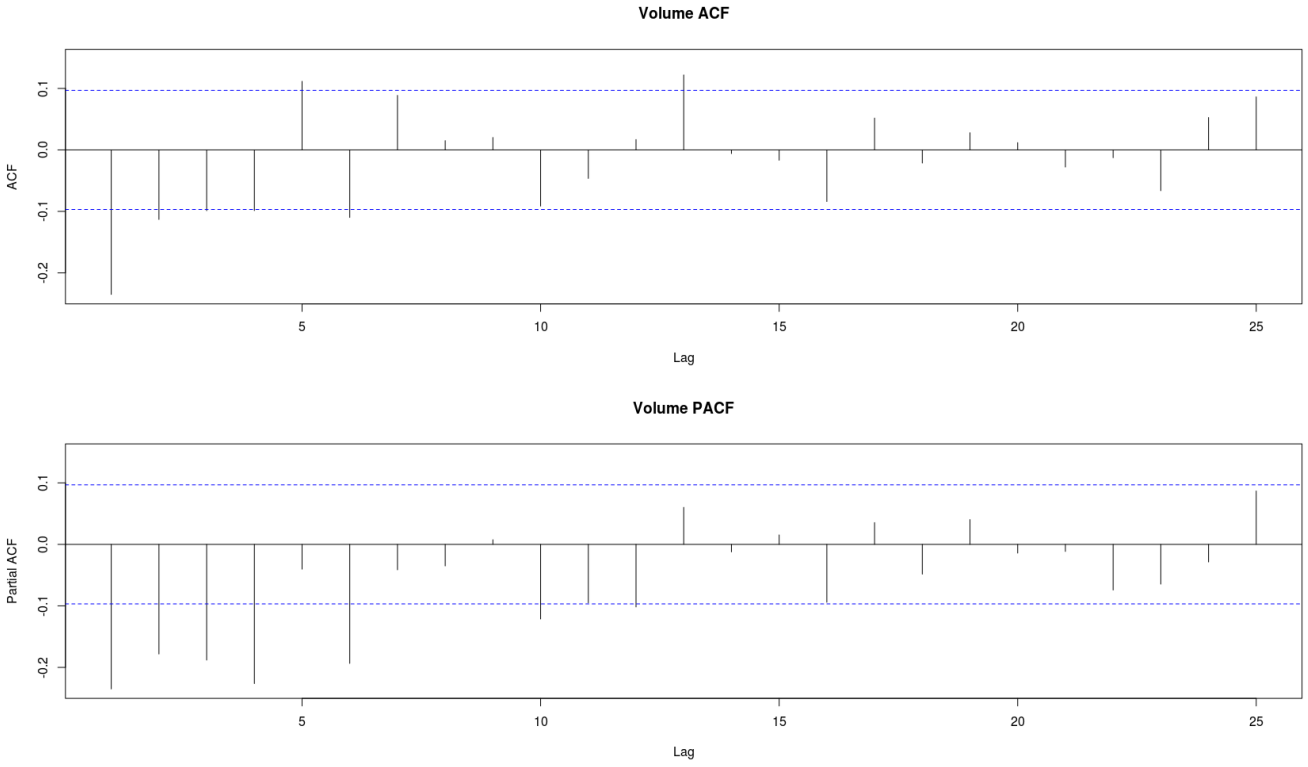


Figure 6.19: ACF and PACF for transformed Volume time series

For this time series volume is not so clear what ARIMA model would fit best just by looking at the ACF and PACF. AR, MA and even ARMA might be candidates. In fact, after trying many different models the model showing a lower criterion AIC was the ARIMA(1,1,1). Next we can see the resulting parameters after the fitting:

ARIMA(1,1,1)

Coefficients:

	ar1	ma1
	0.4438	-0.8775
s.e.	0.0567	0.0269

sigma^2 estimated as 6.004e-10: log likelihood=3770.73

AIC=-7535.47 AICc=-7535.41 BIC=-7523.42

Table 6.6 ARIMA model for the Volume

If we now check the quality of the residuals with the Ljung-Box test to find out if we have accounted sources of correlations we obtain the following plot:

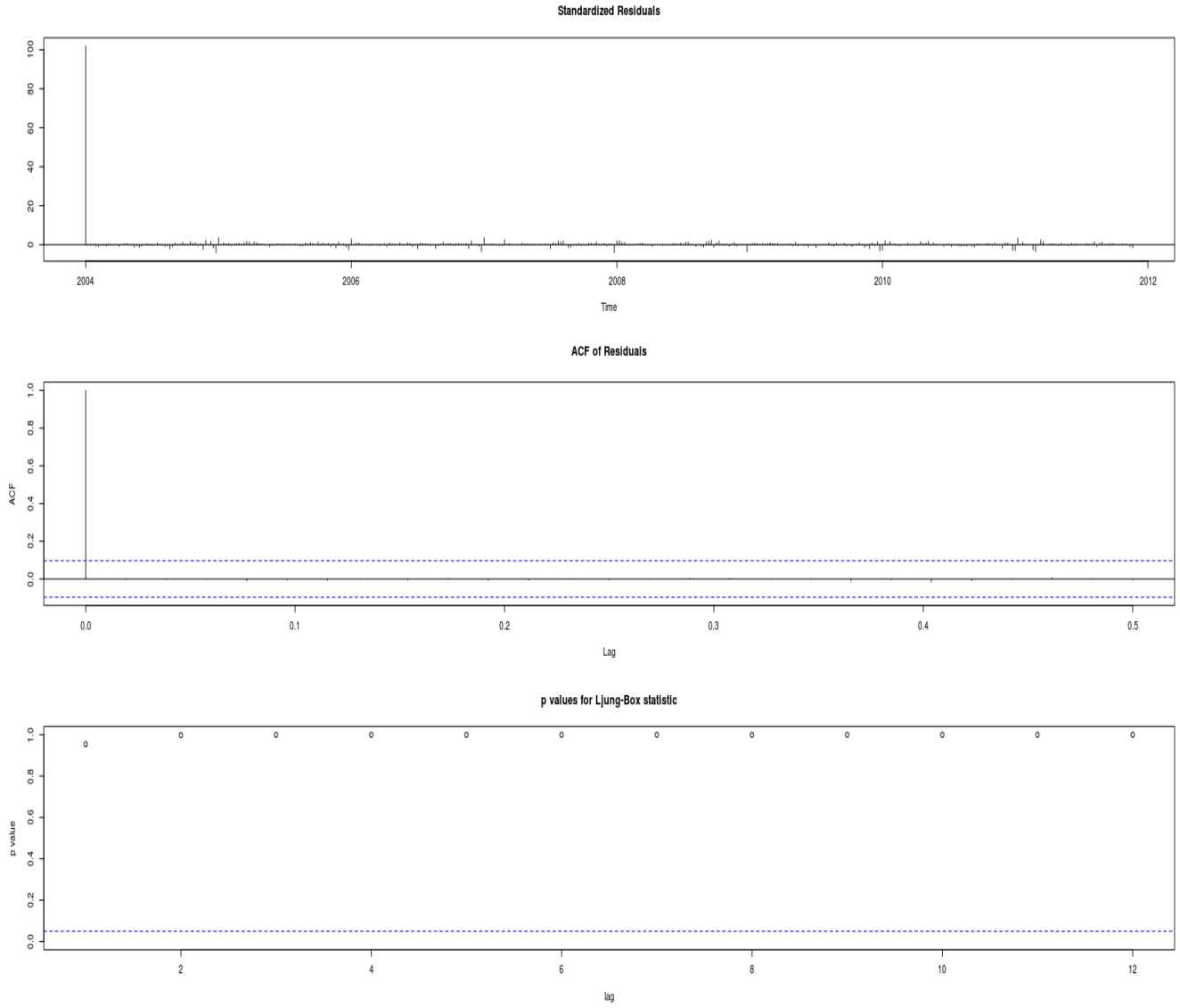


Figure 6.20: LJung-Box test for the Volume ARIMA model

We can observe very well behaved p-values for the Ljung-Box test which guarantees that there is no unaccounted correlations missed by the model.

Finally we can check the stability of the model by calculating a forecasting on the same data by trimming the time series by one year as we did with the previous time series, refitting the ARIMA model with the windowed data and comparing the forecasting of the model with the real data, the results of such comparison can be seen in the next figure:

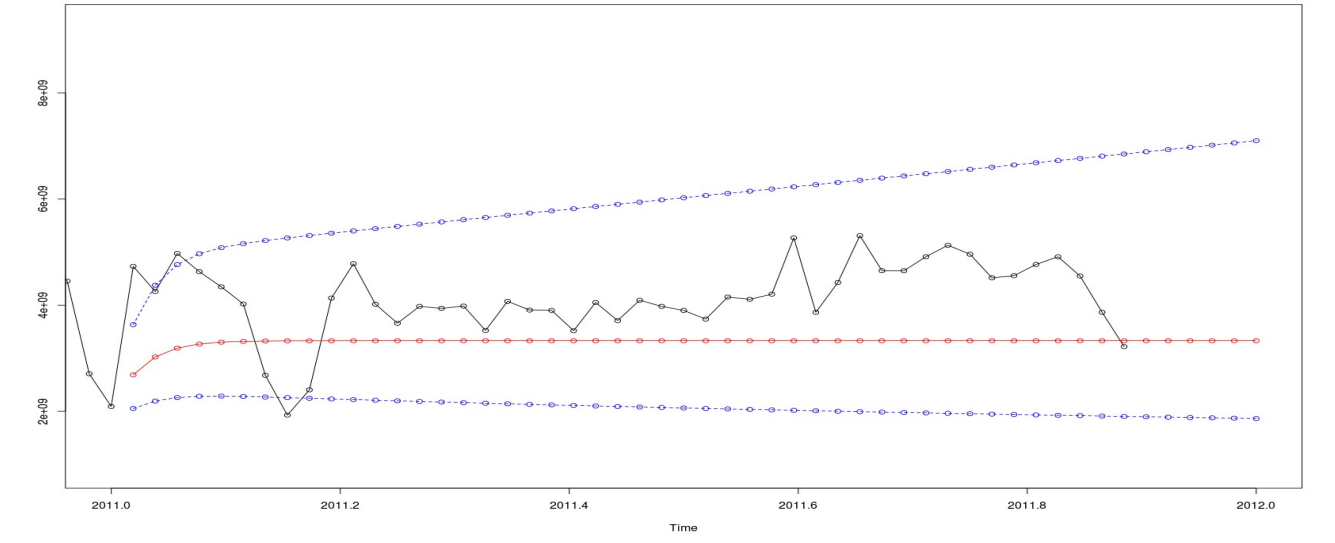


Figure 6.21: Forecasting for a one year windowed Volume time series

As with the previous plots, the blue lines are the confidence intervals at 95% whereas the red line is the forecast done with the model. In this case the fitting is not as perfect as it was with the other cases but very good nonetheless; most values are well withing the confidence interval and the predictions do not go too far from the real data. In the next figure we can see the forecast for the next year with all the data available from the Volume time series.

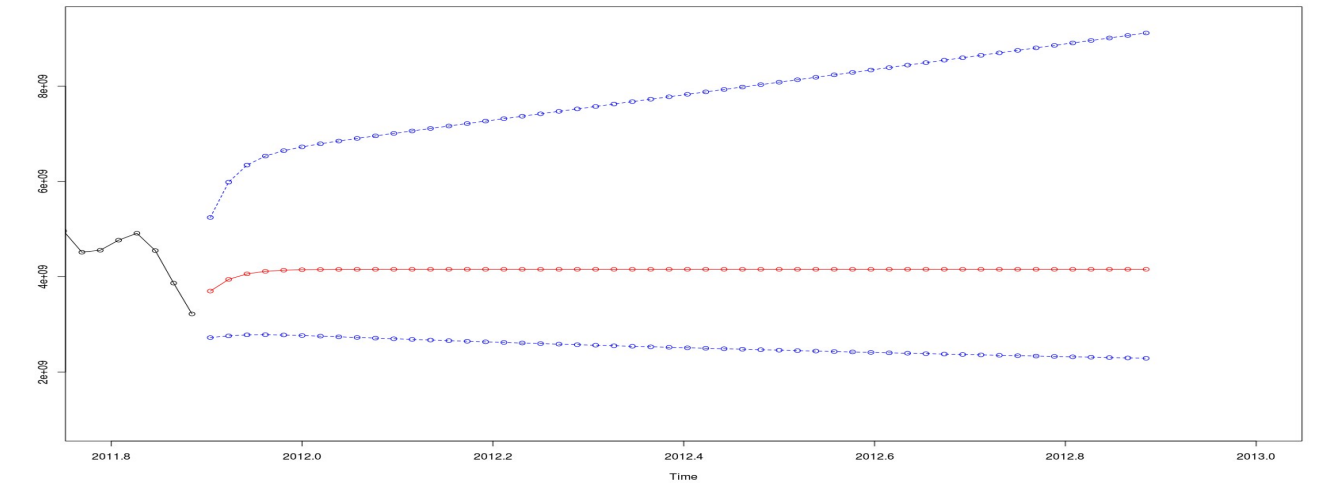


Figure 6.22: Forecasting for Volume for the next year

So we can see that due the stationary non seasonal characteristics of the model the forecasting is constant after the third week forecast. We can also appreciate that the confidence interval is skewed in the upper part meaning that there is more room for growth than decrease in the volume levels.

7 SIMULATION RISK MEASUREMENTS

Finally we have everything in place to calculate the VaR and the Expected Shortfall for the S&P 500 from the simulations. In order to do so we will launch 200 simulations with a final date one year ahead of the data available, that is, until 2012-11-13.

Having 200 simulations will give us around 10000 values for a period of a year which is more than enough to calculate accurately the sample quantile at 1%; since we can expect that 1 out of every 100 values will fall into the 1% quantile that means that we can expect around 100 values within to calculate the VaR and the Expected Shortfall. Obviously for the 5% and 10% quantiles we will have even a more accurate estimation expecting around 500 and 1000 values respectively.

The statistical literature do not offer confidence intervals for risk measurements therefore they will not be calculated in this project since, nonetheless, they are close enough to the estimation for not being relevant for risk management. This is how the distribution of the simulations returns for the S&P 500 looks like After executing the 200 simulations:

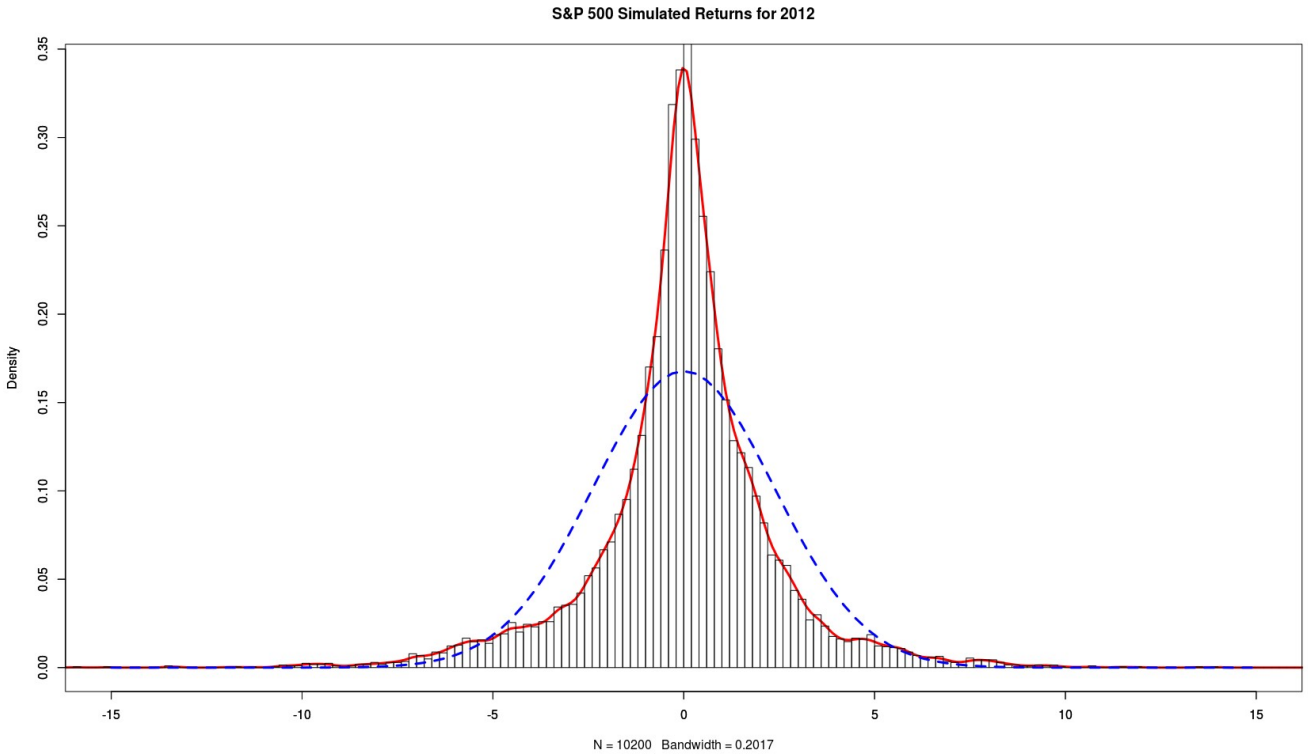


Figure 7.1: S&P 500 200 Simulation Returns for 2012

The red line in the plot correspond to the empirical distribution whereas the blue-dotted line is Normal distribution that fits the data. The distribution is as expected far from Gaussian as we can also see in the following Normal QQ-plot:

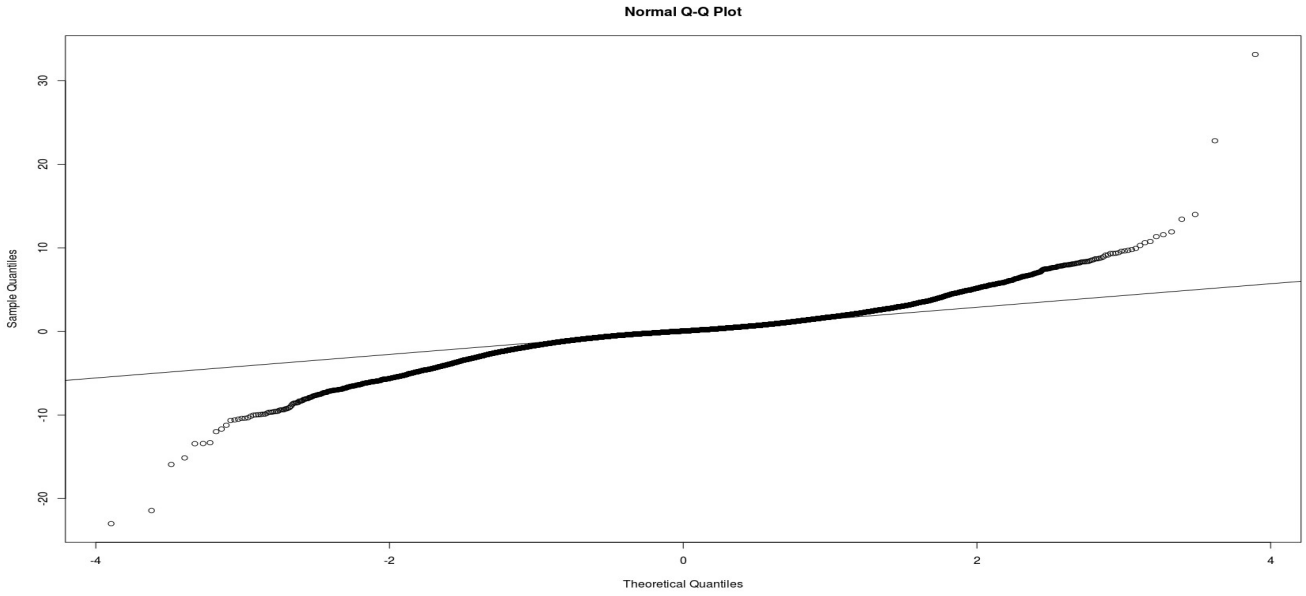


Figure 7.2: Normal QQ-Plot for 200 simulations for the 2012 forecasting

We can also appreciate that simulation shows the heavy tails so characteristic in financial time series. Another characteristic of financial time series that it was introduced previously was the uncorrelated behavior of the first order values of the returns times series as well as the correlated behavior of the second order values of the returns time series; that is the squared returns:In the following plots we can see the ACF for the returns and the squared returns for the 100 simulations of one year forecasting.

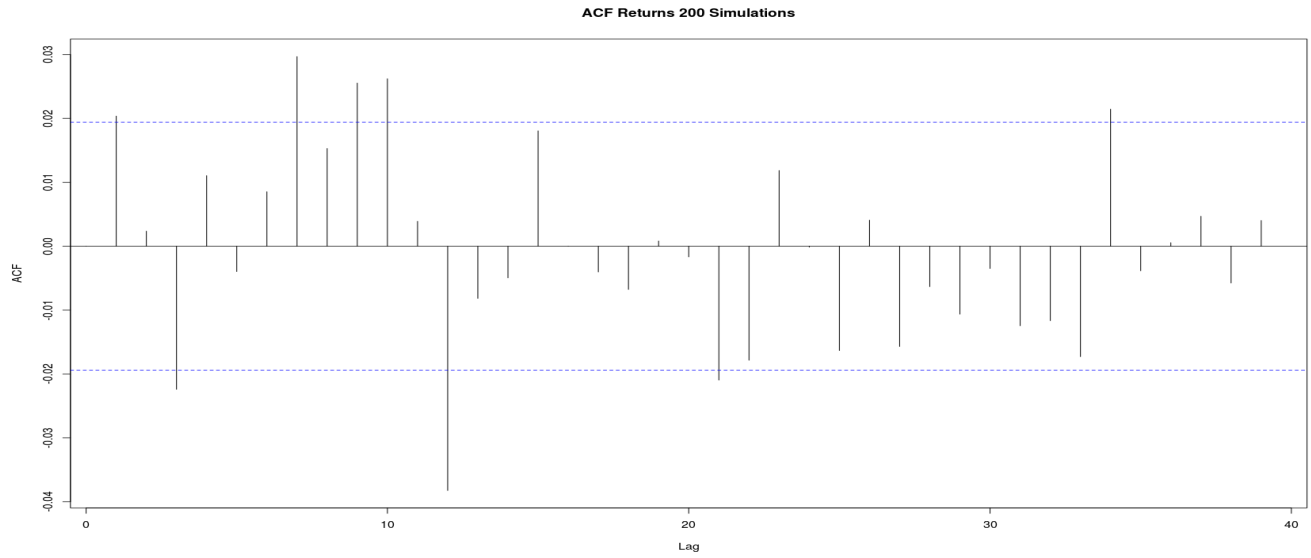


Figure 7.3: ACF for the returns of 100 simulations for a year forecasting

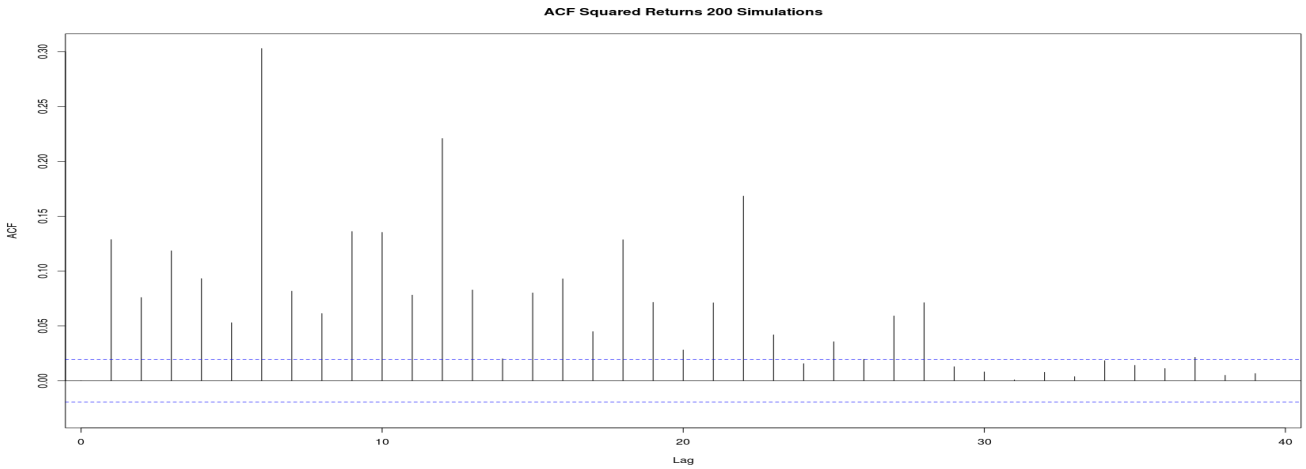


Figure 7.4: ACF for the squared returns of 100 simulations for a year forecasting

We can see how the behavior for the ACF for the returns as squared returns behaves as expected for a financial time series.

Now we can finally calculate the Value at Risk (VaR) and the Expected Returns for the simulation and discuss the results with the risk measurements done previously with classical methods. To calculate the VaR and Expected Shortfall we simple need to calculate the quantiles and the expected values beyond those quantiles just as we do when using the Empirical Quantiles methodology, the results are the following:

P	VaR %	ES %
1%	6.9 %	8.92 %
5%	4.14 %	5.97 %
10%	2.55 %	4.62 %

Table 7.1 Risk Measurements for the Simulations

So the simulation tell us that there is a 1% chance to lose 10.09% or more of our money within the next year when investing in the S&P 500 index. When falling into that 1% scenario the expected amount of money to be lost will be the 14.98% of our investment.

We can also say that there is a 5% chance to lose 5.12% or more with a expected loss of 8.43% of our investment. And finally we also have a 10% chance to lose 3.08% or more with a expected loss of 6.21%.

Finally with all the data available we can next discuss all the results and extract final conclusions for this project.

8 PROJECT RESULTS

We finally have the risk measurements based on behavioral simulations feeding from depression and anxiety data in the general population of the USA for the index S&P 500. After fitting the parameters for the model with thousands of simulations the following optimal values were obtained:

• **:ninvestors**	100	• **:sell_percent**	-1.1395
• **:weight_shape**	[0.4037,1.4575],	• **:depression_factor**	-1.2768
• **:top_weight**	0.4312	• **:mood_factor**	8.936
• **:var_bet_price**	10940.0527	• **:volume_factor**	13.2093
• **:buy_percent**	2.294	• **:news_factor**	1.6385

Table 8.1 Optimal parameters for the fitting process

The final data for the calculation for risk measurements in the year 2012 have been gathered from 200 forecast simulations like the one shown in the following figure:

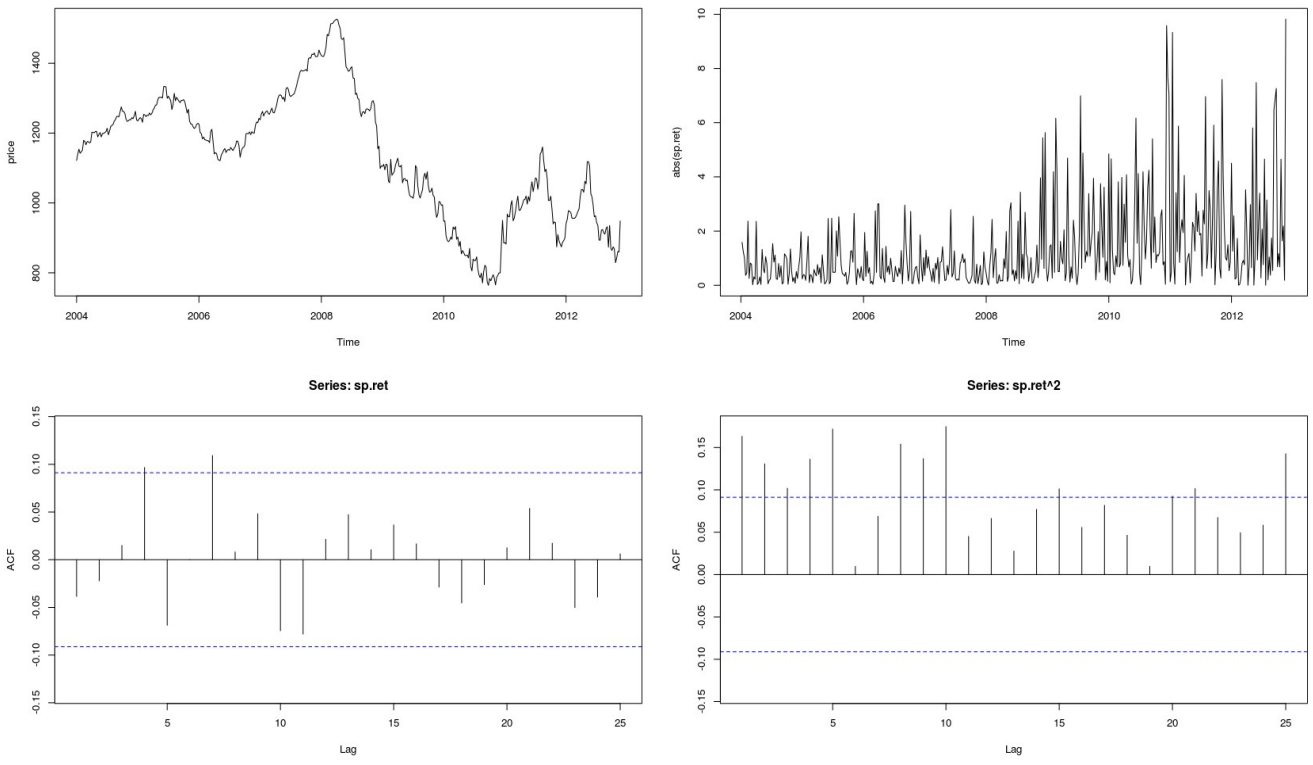


Figure 8.1: Simulation data price, returns absolute value, ACF returns and ACF squared returns.

The price simulated is totally irrelevant for our study since we assume it cannot be predicted in the simulations and we are simply concerned with its volatilities. We can nevertheless appreciate how after the 2007 news event treatment in the simulation, the volatility increases dramatically letting from that moment on the other external events like depression, mood and volume to drive the volatility within the simulation framework.

We can also appreciate how the ACF for the returns and squared returns behave accordingly as expected in a finance time series. The final part of the simulation, the year 2012, is the one considered for the risk measurements.

Since we can execute as many simulations as required, this allow us to gather enough data to apply the empirical quantile methodology to calculate the measurement risk of Value at Riks (VaR) and Expected Shortfall (ES) for the time series Standard & Poor's 500.

In the next table we can see a comparison between all the results of the measurement risk techniques including the behavioral simulations developed for this project.

RiskMetrics		
P	VaR[52] %	ES[52] %
1%	39.72 %	45.51 %
5%	28.09 %	35.22 %
10%	21.88 %	29.97 %

Empirical Quantiles		
P	VaR %	ES %
1%	7.28 %	10.69 %
5%	4.53 %	6.72 %
10%	2.84 %	5.21 %

Extreme Value Theory		
P	VaR %	ES % (est)
1%	5.43 %	19.88 %
5%	2.94 %	6.69 %
10%	2.56 %	4.33 %

Peaks Over Threshold (4.0 %)		
P	VaR %	ES %
1%	7.97 %	11.60 %
5%	4.35 %	6.79 %
10%	3.19 %	5.24 %

Peaks Over Threshold (2.5 %)		
P	VaR %	ES %
1%	8.32 %	11.41 %
5%	4.35 %	6.89 %
10%	2.86 %	5.20 %

Behavioral Simulations		
P	VaR %	ES %
1%	6.9 %	8.92 %
5%	4.14 %	5.97 %
10%	2.55 %	4.62 %

Table 8.2 VaR and Expected Shortfall comparison table for the next year S&P 500 index

9 CONCLUSIONS

At this point a fair question is which of the methodologies discussed for risk measurement is the best, since there is no true VaR available to compare the accuracy of different approaches Tsay[36] recommends to apply several methods to gain insight into the range of VaR.

Nonetheless in this case we can discard the RiskMetrics method for long positions of a year since it clearly gives us an overestimation of the risks involved in the investments for the S&P 500 index.

Empirical Quantiles will most of the time show an optimistic view of the risk for long positions since it is based on a few samples for the length of the position analyzed, nonetheless it is a safe bet and can be consider as a fair minimum value for long positions.

The statistics behind Extreme Value Theory are very sound but they require large amount of data for long positions, for one year position the eight years of data proved to be insufficient for the 1% risk giving us a Expected Shortfall of 19.88% which is likely to be an overestimation due to the scarce data available for the fitting and the influence of the big volatilities found in the period 2007/2008.

Finally Peaks Over Threshold is the best well behaved method among the classical procedures, its statistical theory is very sound and it does not require large sets of data for long positions, upon the risk considered we need to choose between a 2.5% threshold or a 4% threshold but in general it gives us a reliable measurement for the time series studied in this project.

It is important to notice that all classical methods (except for the Econometric Modeling) rely on the premise that the risk in the future will be similar to the risk in the past. Econometric Modeling does not assume this premise since it does forecasting on the volatilities, but the forecasting for such methods only work for very short positions as we have seen in the RiskMetrics case. For large positions there is no classical method based on a forecasting to calculate the VaR and the Expected Shortfall.

The simulation described in this project offers the possibility to calculate such forecast long position risk measurement considering how the time series returns are expected to behave in the future based on how they behaved in the past.

36 Tsay R.S. (2005).

The results the simulations deliver for the VaR and Expected Shortfall are a bit lower than for the Peaks Over Threshold estimations, but that these results makes sense if we consider that the POT might be affected by the extreme values for volatilities found in 2007/2008 whereas the simulation only accounts for the stable predictions we found for the external events volume, mood an depression.

Having a look to the fitted parameters in Table 8.1 we could argue that the mood parameter with a weight of 8.936 explains better the behavioral volatility factor than the depression parameter with just a -1.2 fitting value. This suggest that a simplification of the model might be a good idea and leave the behavioral influence on the simulation just on the mood parameter since, anyway, is the one combining depression and anxiety altogether and, of course, the flexibility of the simulation process always allows us to introduce new external events to drive the volatilities.

The main disadvantage for the behavioral simulation method is the large number of simulations required to estimate parameters which makes this procedure not suitable for analyzing large numbers of time series on very short positions.

Another disadvantage for the simulations is the fact that its volatility is driven by behavioral parameters that might be relevant only to big indexes or financial time series heavily affected by the index. Small companies might show a totally uncorrelated behavior versus the depression or mood parameters thus reducing its forecasting power.

On the other hand another great advantage for behavioral simulations, besides the fact of being able to measure risk based on a forecasting, is the fact that the estimations of its parameters allow us to interpret what kind of risk we are considering.

For example, we can give a meaning to the volatility levels in the model by checking if its growth is due to a growth in the volume or a growth or a decrease in the depression levels of investors. If the cause was psychological we might be facing a higher risk than if the cause is purely mechanical and due to the volatility growth since we might interpret it as a simple, and not worrying, market adjustment.

But external factors like mood are not the only ones that might give us clues about the kind of risk we are dealing with. Let's consider again for a moment the optimal values for the fitting procedure in table Table 8.1. It is interesting noticing that the adjustment gave a buy percentage of around 2% and and sell

percentage of around -1%, this means that in this model investors are likely to sell if they feel there is a small risk and they need more confidence when buying, around twice as much as when selling.

The simulation therefore, does not give us just a measurement of the risk, but also a the framework in which this risk have been calculated, making simulations a very useful methodology when it comes to manage risk assessments in the financial market.

Another attractive aspect of having a behavioral driven simulation validated with structural equations is that its claiming about psychological factors affecting the markets follow the Popper's principle of falsifiability allowing us to update the model accordingly we new data is available.

The simulation framework allow more complex interactions and a huge flexibility that, properly validated, has a great potential for further study in the mechanics of financial markets and. though simulations will be never be able to exactly model the financial market, it can certainly become another tool in the statistical toolbox for gaining insights when managing risk in finance markets.

“Essentially, all models are wrong, but some are useful”[37]

George E. P. Box

37 Empirical Model-Building and Response Surfaces (1987), co-authored with Norman R. Draper, p. 424, ISBN 0471810339

APPENDIX

A. *R Code for Basic Statistics, Plots and ARIMA fittings*

```
################################################################################
####  Main file code for basic analysis and plots for the PFM project          ###
####  As well as for the ARIMA forecasting for external events and files       ###
################################################################################

##Fetch Data for S&P 500
library(quantmod)

getSymbols("^GSPC",from="1900-01-01")
sp500 = GSPC$GSPC.Adjusted

# Plot SP500, ACF, PACF
plot(sp500,main="S&P 500")
library(forecast)
tsdisplay(GSPC$GSPC.Adjusted,start=c(1950,3),frequency=1,title="S&P 500")
par(mfrow=c(2,1));Acf(sp500,main="S&P 500");Pacf(sp500,main="S&P 500")

## unit root S&P 500
library(fUnitRoots)
adfTest(sp500)

## Log returns S&P 500 and plots

sp500.ret = 100*diff(log(sp500))
sp500.ret = sp500.ret[2:length(sp500),]
par(mfrow=c(1,1));plot(sp500.ret,main="S&P 500 Returns")
par(mfrow=c(2,1));Acf(sp500.ret,main="S&P 500 Returns");Pacf(sp500.ret,main="S&P 500 Returns")

## Arima for S&P 500
var(sp500.ret))
sp500.ret.diff = diff(sp500.ret)
var(sp500.ret.diff[2:length(sp500.ret.diff)]) #variance increases
```

```
m = arima(sp500.ret,c(3,0,0),include.mean=FALSE) #no mean since the returns are differenced
summary(m)
qqnorm(m$residuals);abline(0,1)
impact = m$residuals

## Squared impact/residuals
par(mfrow=c(2,1));Acf(impact^2,main="Squared S&P 500 Residuals");Pacf(impact^2,main="Squared S&P 500
Residuals")

#Weekly Data extracted from Google Insight and Yahoo Finance
spmood = read.table("~/workspace/university/FME/PFM/sp500.mood.csv",header=T,sep=",")
anxiety = ts(spmood$anxiety, frequency = 52, start = c(2004,1))
depression = ts(spmood$depression, frequency = 52, start = c(2004,1))
mood = ts(spmood$mood, frequency = 52, start = c(2004,1)) #anxiety-depression
volume = ts(spmood$volume , frequency =52, start = c(2004,1))
sp = ts(spmood$close, frequency =52, start = c(2004,1))

##Weekly S&P 500 from 2004
sp.ret = 100*diff(log(sp))
var(sp.ret)
sp.ret.diff = diff(sp.ret)
var(sp.ret.diff[2:length(sp.ret.diff)]) #variance increases
par(mfrow=c(2,1));
Acf(sp.ret,main="S&P 500 weekly returns from 2004");
Pacf(sp.ret,main="S&P 500 weely returns from 2004")
ret = ts(c(mean(sp.ret),sp.ret),frequency=52,c(2004,1))

## Basic Analysis
df1 = data.frame(anxiety,depression,volume,ret,ret^2)
round(cor(df1),2)
plot(df1)

df2 = data.frame(mood,depression,volume,ret,ret^2)
round(cor(df2),2)
plot(df2)

df3 = data.frame(mood, depression,volume,ret^2,log(ret^2))
```

```
round(cor(df3),2)
plot(data.frame(mood, depression,log(ret^2)))

par(mfrow=c(2,1));plot(ret,main="S&P 500 Returns");plot(mood,main="Mood");abline(0,0,lty=2,col="red",lwd=2)

library(forecast)
#ARIMA(p,d,q)(P,D,Q)s model report
tetrad = read.table("~/workspace/university/FME/PFM/data1.txt",header=T,sep="\t")
plot(tetrad$X,tetrad$Y,xlim=c(-4,4),ylim=c(-4,4))
summary(lm(formula = tetrad$X ~ tetrad$Y))

## graficos
xlim = c(2004.00,2012.00)
ylim = c(min(min(anxiety),min(depression)),max(max(anxiety),max(depression)))
par(mfrow=c(1,1))
plot(anxiety ,main="",col='red',xlim=xlim,ylim=ylim)
par(new=TRUE)
plot(depression,main="",col='blue',xlim=xlim,ylim=ylim)
par(new=TRUE)
plot(anxiety-depression,main="",col='green',xlim=xlim,ylim=ylim)

##
par(mfrow=c(2,1))
plot(sp.ret,xlim = c(2007.00,2012.00))
plot(anxiety-depression, xlim = c(2007.00,2012.00))
abline(a=0,b=0)
par(mfrow=c(1,1))

###################### ARIMA for depression, mood and volume ############################
#################################################################################
library(forecast)
library(fUnitRoots)
######################

spmood = read.table("~/workspace/university/FME/PFM/sp500.mood.csv",header=T,sep=",")
#serie = mood + 100 #mood
#serie = depression
```

```
serie = volume
## transform the serie to treat heterocedasticity
lambda = BoxCox.lambda(serie,lower=-2)

tsdisplay(serie,main="Volume")
serie.tr = BoxCox(serie,lambda)
plot(serie.tr)

monthplot(serie)
## estimation of D and d
## estimate D seasonality
s=52
D=0
if (s > 0)
repeat {
  p.value = adfTest(serie.tr)@test$p.value
  serie.tr.ds = diff(serie.tr,s)
  if ( p.value >=  0.05 & var(serie.tr.ds) < var(serie.tr))
     {serie.tr = serie.tr.ds; D=D+1} else break;
}


serie.tr = BoxCox(serie,lambda)
## calculate how many regular diferentations are required

layout(1)
ts.plot(diff(serie.tr),main="Mood diff diff 52")

serie.tr.ds = diff(BoxCox(serie,lambda),s) # when D>0
serie.tr.diff.s = BoxCox(serie,lambda)
d=0
repeat {
  p.value = adfTest(serie.tr.diff.s)@test$p.value
  serie.tr.diff.s.r = diff(serie.tr.diff.s)
  if ( p.value >=  0.05 & var(serie.tr.diff.s.r) < var(serie.tr.diff.s))
     {serie.tr = serie.tr.diff; d=d+1} else break;
}
```

```
serie.tr = BoxCox(serie,lambda)
serie.tr.ds.dr = diff(diff(serie.tr,52)) #diff(diff(serie.tr,52))

s=52
par(mfrow=c(1,2))
acf(serie.tr.ds.dr,ylim=c(-1,1),lag.max=200,col=c(2,rep(1,s-1)),lwd=2,main="Wt")
pacf(serie.tr.ds.dr,ylim=c(-1,1),lag.max=200,col=c(rep(1,s-1),2),lwd=2,main="Wt")

par(mfrow=c(2,1))
Acf(diff(serie.tr),main="Volume ACF")
Pacf(diff(serie.tr),main="Volume PACF")

model = Arima(serie.tr, order = c(3,0,0), seasonal = list(order=c(1,1,0)),include.drift=FALSE) #depression
#model = Arima(serie.tr, order = c(0,1,1), seasonal = list(order=c(0,1,1)),include.drift=FALSE) #depression
model = Arima(serie.tr, order = c(0,1,2), seasonal = list(order=c(0,1,1)),include.drift=FALSE) #mood
#model = Arima(serie.tr, order = c(3,0,0), seasonal = list(order=c(1,1,0)),include.drift=FALSE) #mood
#model = Arima(serie.tr, order = c(1,1,1),include.drift=FALSE);model #volume
model = Arima(serie.tr, order = c(0,1,2), seasonal = list(order=c(0,1,1)),include.drift=FALSE) #volume

tsdiag(model,gof.lag=12)

serie.tr.2<-window(serie.tr,end=c(2011,1))
model2 = Arima(serie.tr.2, order = c(3,0,0), seasonal = list(order=c(1,1,0)),include.drift=FALSE) #depression
#model2 = Arima(serie.tr.2, order = c(0,1,2), seasonal = list(order=c(0,1,1)),include.drift=FALSE) #mood

plot(serie.tr.2)
#pred1<-predict(model,n.ahead=52)
pred1<-predict(model2,n.ahead=52)
pr<-pred1$pred
se<-pred1$se

tl<-pr-1.96*se
tu<-pr+1.96*se

## tl<-InvBoxCox(tl,lambda) - 100 mood
## pr<-InvBoxCox(pr,lambda) - 100
```

```
## tu<-InvBoxCox(tu,lambda) - 100

tl<-InvBoxCox(tl,lambda)
pr<-InvBoxCox(pr,lambda)
tu<-InvBoxCox(tu,lambda)

layout(1)
#ts.plot(depression,tl,tu,pr,lty=c(1,2,2,1),col=c("black","blue","blue","red"),xlim=c(2010.4,2012),type="o")
ts.plot(mood,tl,tu,pr,lty=c(1,2,2,1),col=c("black","blue","blue","red"),xlim=c(2010.4,2012),type="o")
#ts.plot(mood,tl,tu,pr,lty=c(1,2,2,1),col=c("black","blue","blue","red"),xlim=c(2011,2013),type="o")



## ## Fitting  ARIMA model ARIMA(p,d,q)(P,D,Q)s



## seasonality returned by model
##              1,2,3,4,5,6,7
## model$arma = p,q,P,Q,s,d,D
plot(decompose(serie.tr))
s=model$arma[5]
d=model$arma[6]
D=model$arma[7]
if (D>0) for(i in 1:D) {serie.tr = diff(serie.tr,s); tsdisplay(serie.tr)}
if (d>0) for(i in 1:d) {serie.tr = diff(serie.tr); tsdisplay(serie.tr)}

############### Event data files for predictions ##########################
#########################################################################
sp500 = ts(spmood$close, frequency =52, start = c(2004,1))
depression = ts(spmood$depression, frequency =52, start = c(2004,1))
anxiety = ts(spmood$anxiety, frequency =52, start = c(2004,1))
mood = ts(spmood$mood, frequency =52, start = c(2004,1))
volume = ts(spmood$volume, frequency =52, start = c(2004,1))

prediction = InvBoxCox(predict(model,n.ahead=52)$pred,lambda)
volume.pred = c(volume,prediction)
mood.pred = c(mood,prediction-100)
```

```
#prediction
initial_date = 1073174400.0 # 2004-1-04 seconds from 1970
gap =  604800.0             # 7 days in seconds
time = initial_date + c(0,cumsum(rep(gap,length(mood.pred)-1)))

## sp500
## df = data.frame(time,sp500)
## write.table(df,file = 'sp500.pred.dat', quote = FALSE)

## anxiety
## df = data.frame(time,anxiety)
## write.table(df,file = 'anxiety.pred.dat', quote = FALSE)

## depression
df = data.frame(time,depression.pred)
write.table(df,file = 'depression.pred.dat', quote = FALSE)

## mood
df = data.frame(time,mood.pred)
write.table(df,file = 'mood.pred.dat', quote = FALSE)

## volume
df = data.frame(time,volume.pred)
write.table(df,file = 'volume.pred.dat', quote = FALSE)

## news
## initial_date = 1226016000.0  # 2008-11-7 seconds/wk36 from 1970
## period = 52-36
## gap =  604800.0             # 7 days in seconds
## time = initial_date + c(0,cumsum(rep(gap,period-1)))

## news = (period-1):1/(period-1) # 1/x decrease phenomena
## news = c(sum(news),-news)
## df = data.frame(time,news)
## write.table(df,file = 'news.pred.dat', quote = FALSE)
#########################
```

B. R Code for Risk Measurements

```
########################################
#### Risk Measurements Procedures      #####
########################################

library(fUnitRoots)
library(fGarch)
library(forecast)

spt500 = read.table('~/workspace/university/FME/PFM/simulator/data/sp500.dat',header=TRUE)
sp500 = ts(spt500$sp500, frequency =52, start = c(2004,1))

sp500.ret = 100*diff(log(sp500))

plot(sp500.ret, main="S&P 500",col="dark red",lwd="2")
tsdisplay(sp500.ret, main="S&P 500",col="dark red",lwd="2")



################## Empirical Quantile ####################
############################################################

p = 0.05
VaR.pc= quantile(sp500.ret,p); round(VaR.pc,2) #  VaR percent
ES.pc = mean(sp500.ret[sp500.ret <= VaR.pc]); round(ES.pc,2) # Expected Shortfall percent

## Forecast
fore = read.table('~/workspace/university/FME/PFM/simulator/data/forecasts.dat',header=TRUE,row.names =
NULL)

p = 0.10
VaR.pc= quantile(fore$returns,p,na.rm=TRUE); round(VaR.pc,2) #  VaR percent
ES.pc = mean(fore$returns[fore$returns <= VaR.pc],na.rm=TRUE); round(ES.pc,2) # Expected Shortfall percent
layout(1)
dhist(fore$returns)

qqnorm(fore$returns)
qqline(fore$returns)
```

```
Acf(fore$returns, main="ACF Returns 100 Simulations")
Acf(fore$returns^2, main="ACF Squared Returns 100 Simulations")

## Function dhist to plot histogram/empirical density/adjusted normal
dhist = function(data,main="") {
mu = mean(data)
sigma = sqrt(var(data))
den = density(data,na.rm=TRUE)
xlim = c(min(den$x),max(den$x))
ylim = c(0,max(den$y))
plot(den,lwd=3,main="",col="red",xlim=xlim,ylim=ylim)
par(new=T)
h = hist(data,breaks=round(length(data)/10),main=main,xlim=xlim,ylim=ylim,freq=F,xlab="")
curve(dnorm(x,mu,sigma), add=TRUE, col="blue",lwd=3,lty=2,main="",xlim=xlim,ylim=ylim)}

layout(1)
#dhist(sp500.ret,main="S&P 500 Returns")
dhist(fore$returns,main="S&P 500 Returns")

#####################  RiskMetrics  ###################
########################################################
source('~/workspace/university/FME/Análisis de Volatilidad/Week6/mvwindow.R') [38]
source('~/workspace/university/FME/Análisis de Volatilidad/Week9/Igarch.R')
mi = Igarch(as.numeric(sp500.ret),include.mean=F,volcnt=F)
beta = as.numeric(mi$par["beta"])

## GARCH The starting value σ0 is fixed at either zero or the unconditional
## variance of a_t
## In some applications, the sample variance of a_t serves as a good starting value 2
## of σ1 .
## 3.5.2 Forecasting Evaluation
## pag 334 Tsay
mvw = mvwindow(sp500.ret,63)
plot.ts(mvw$sigma.t)
at0 = tail(sp500.ret,1)
sigma0 = tail(mvw$sigma.t,1)
```

38 mwwindor.R and Igarch.R scripts may be found at http://faculty.chicagobooth.edu/ruey.tsay/teaching/bs41202/sp2011/

```
sigmat2 = (1-beta)*at0^2   +  beta*sigma0^2

## VaR[k] = qnorm(p)*sigma_{t+1}*sqrt(k)

rmVaR = function(ret,k,p){
  mi = lgarch(as.numeric(ret),include.mean=F,volcnt=F)
  beta = as.numeric(mi$par["beta"])
  mvw = mvwindow(ret,63)
  at0 = tail(ret,1)
  sigma0 = tail(mvw$sigma.t,1)
  sigmat2 = (1-beta)*at0^2   +  beta*sigma0^2
  -qnorm(p)*sqrt(sigmat2)*sqrt(k)
}

p = 0.1
periods = 52
VaR = -qnorm(p)*sqrt(sigmat2)*sqrt(periods); round(VaR,2) ##percent
# expected shortfall page 333
ES = dnorm(qnorm(p))*sqrt(sigmat2)*sqrt(periods)/p; round(ES,2)

############################## Extreme Value Theory Tsay  page 351/355 #############
#################################################################################

########### fitting a GEV
library(evir)
#library(POT)
source('~/workspace/university/FME/Análisis de Volatilidad/Week9/evtVaR.R')

########## Sample EGV
ev = NULL;
n = 52
for (i in 1:1000) ev = c(ev,max(sample(sp500.ret,n))) #
#for (i in 1:1000) ev = c(ev,max(rnorm(1000)))
hist(ev,breaks=50)
#############

ngev = 52
```

```
m = gev(-sp500.ret,ngev) #n=63/5
p = 0.01
VaR = evtVaR(xi=m$par.ests['xi'],sigma=m$par.ests['sigma'],mu=m$par.ests['mu'],ngev,p)

## ######## Estimation of expecte shortfall by simulation Tsay pag. 355 or evtVaR
rVaRgev = function  (n, xi = 1, mu = 0, sigma = 1, ngev){
  mu - (sigma * (1 - (-ngev*logb(runif(n)))^(-xi)))/xi
}
rnd = rVaRgev(n=100000,xi=m$par.ests['xi'],sigma=m$par.ests['sigma'],mu=m$par.ests['mu'],ngev=ngev)
ES = mean(rnd[rnd>=VaR])

######### Return Levels
## 1 every 7 periods of 8 weeks (+-year) we find this loss or higher
rlevel.gev(m,k.blocks=7)



####### plots
####### dhist for gev
dhist = function(data,main="") {
mu = mean(data)
sigma = sqrt(var(data))
den = density(data)
xlim = c(min(den$x),max(den$x))
ylim = c(0,2*max(den$y))
plot(den,lwd=3,main="",col="red",xlim=xlim,ylim=ylim)
par(new=T)
h = hist(data,breaks=round(length(data)/30),main=main,xlim=xlim,ylim=ylim,freq=F,xlab="")
curve(dgev(x,xi=m$par.ests['xi'],sigma=m$par.ests['sigma'],mu=m$par.ests['mu']), add=TRUE,
col="blue",lwd=3,lty=2,main="",xlim=xlim,ylim=ylim)}

par(mfrow=c(1,1))
dhist(ev,"GEV for S&P 500 Returns")

##GEV Distributions Plot
par(mfrow=c(1,1))
xlim = c(-5,5)
curve(dgev(x,xi=1,sigma=2,mu=0), add=FALSE, col="black",lwd=3,lty=1,main="",xlim=xlim,n=1001) #Frechet
```

```
curve(dgev(x,xi=0.0001,sigma=2,mu=0), add=TRUE, col="red",lwd=3,lty=1,main="",xlim=xlim,n=1001) #Gumbel
curve(dgev(x,xi=-0.5,sigma=2,mu=0), add=TRUE, col="blue",lwd=3,lty=1,main="",xlim=xlim,n=1001) #Weibull
x<-seq(0,5,length=400)
y<-dnorm(x)
#par(mar=c(5,4,2,1))
#plot(x, y2, type="n", xlab=quote(Z==frac(mu[1]-mu[2],sigma/sqrt(n))), ylab="Density")
plot(x, y2, type="n", xlab="Returns", ylab="Density")
p = 0.95
thr = qnorm(p)
polygon(c(thr+x,rev(thr+x)),c(dnorm(thr+x),rep(0,400)),col="red", lty=0)
#polygon(c(0,1,1,0)/10, c(0,0,1,1)/10,col="blue", lty=0)
lines(x, y)
## legend(4.2, .4, fill=c("grey80","grey30"), legend=expression(P(abs(Z)>1.96, H[1])==0.85,
P(abs(Z)>1.96,H[0])==0.05), bty="n")
#text(0, .2, quote(H[0]:~~mu[1]==mu[2]))


################## Peak Over Thresholds  Tsay page 366 ##########################
#################################################################################

library(evir)
library(evd)

################### plots and threshold ############
par(mfrow=c(3,1))
meplot(-sp500.ret)
title(mail="Mean Excess Plot")
tcplot(-sp500.ret,tlim=c(-10,7))

################ fitting a GPD ###################
m = pot(-sp500.ret,2.5) #short position
round(riskmeasures(m,c(0.99,0.95,0.90)),2)
m = pot(-sp500.ret,4) #long position
round(riskmeasures(m,c(0.99,0.95,0.90)),2)
```

C. R Code for Handling and Analyze Simulations

```
## Main file to hanlde the execution fo simulations
## and perform statistical analyses as well as to
## handle external events data files

forecast()
##########

simulate()
##########

## launches 100 forecast simulations
forecast <- function(){
  system('rm ~/workspace/university/FME/PFM/simulator/data/forecasts.dat')
  for( i in 1:100) {system('~/workspace/university/FME/PFM/simulator/lib/simulator.rb')}
  print('**************************************************')
  print(i)
  print('**************************************************')
}

## launches a simulation and shows  plot analysis
simulate <- function(){
  library(forecast)

  stock= NULL; trans = NULL; price = NULL; sp.ret = NULL
  system('~/workspace/university/FME/PFM/simulator/lib/simulator.rb')

  stock = read.table('~/workspace/university/FME/PFM/simulator/data/price_change.events.SP500
weekly.dat',header=TRUE)

  layout(1)
  price =  ts(stock$price, frequency = 52, start = c(2004, 1))
```

```
  sp.ret = 100*(diff(log(price)))
  par(mfrow=c(2,2))

  ts.plot(price)
  ts.plot(abs(sp.ret))
  Acf(sp.ret)
  Acf(sp.ret^2)
}



############### Event data files  ##############################
#############################################################

initial_date = 1073174400.0 # 2004-1-04 seconds from 1970
gap =  604800.0             # 7 days in seconds
time = initial_date + c(0,cumsum(rep(gap,length(spmood$close)-1)))

## sp500
sp500 = ts(spmood$close, frequency =52, start = c(2004,1))
df = data.frame(time,sp500)
write.table(df,file = 'sp500.dat', quote = FALSE)

## anxiety
anxiety = ts(spmood$anxiety, frequency =52, start = c(2004,1))
df = data.frame(time,anxiety)
write.table(df,file = 'anxiety.dat', quote = FALSE)

## depression
depression = ts(spmood$depression, frequency =52, start = c(2004,1))
df = data.frame(time,depression)
write.table(df,file = 'depression.dat', quote = FALSE)
```

```
## mood
mood = ts(spmood$mood, frequency =52, start = c(2004,1))
df = data.frame(time,mood)
write.table(df,file = 'mood.dat', quote = FALSE)

## volume
volume = ts(spmood$volume, frequency =52, start = c(2004,1))
df = data.frame(time,volume)
write.table(df,file = 'volume.dat', quote = FALSE)

## news
initial_date = 1226016000.0  # 2008-11-7 seconds/wk36 from 1970
period = 52-36
gap =  604800.0            # 7 days in seconds
time = initial_date + c(0,cumsum(rep(gap,period-1)))

news = (period-1):1/(period-1) # 1/x decrease phenomena
news = c(sum(news),-news)
df = data.frame(time,news)
write.table(df,file = 'news.dat', quote = FALSE)
```

D. Ruby Code and Structural Equations Models

The core functionality for simulations is implemented in Ruby language and comprehends **2.304** lines of code which is far to big to place it in this documentation, therefore the Ruby code for the project is available to the committee upon request by Email at **viraltux@gmail.com**.

The **TETRAD** model used to validate the structural equation model within the simulation framework is in binary format and cannot be place in a text document, this model is also available over request to the same Email address.

BIBLIOGRAPHY

- Acerbi, C., and Tasche, D. (2002). *Expected Shortfall: a natural coherent alternative to Value at Risk (Wiley Online Library).* Economic Notes 31 (2), 379-88. http://onlinelibrary.wiley.com/doi/10.1111/1468-0300.00091/abstract
- Bak, P., Paczuski, M., and Shubik, M. (1997). *Price variations in a stock market with many agents.* Physica A: Statistical Mechanics and its Applications 246 (3-4), 430-53. http://arxiv.org/pdf/cond-mat/9609144
- Bentler. P. (1995). *EQS structural equations program manual.* Multivariate Software Encino, CA, http://www.econ.upf.edu/~satorra/CourseSEMVienna2010/EQSManual.pdf
- Bollerslev, T. (1986). *Generalized autoregressive conditional heteroskedasticity.* Journal of Econometrics 31: 307–327.
- Bollerslev, T., Engle, R. F., and Nelson, D. B. (1994). *ARCH model.* In R. F. Engle and D. C. McFadden (eds.). Handbook of Econometrics IV , pp. 2959–3038. Elsevier Science, Amsterdam.
- Cao, C. and Tsay, R. S. (1992). *Nonlinear time series analysis of stock volatilities.* Journal of Applied Econometrics 7: s165–s185.
- Chan, K. S. and Tsay, R. S. (1998). *Limiting properties of the conditional least squares estimator of a continuous TAR model*. Biometrika 85: 413–426.
- Chen Y. P. and Goldberg D. (2005). *Convergence Time for the Linkage Learning Genetic Algorithm.* Evolutionary Computation 13(3), 279-302. doi:10.1162/1063656054794806
- Cheng, B. and Titterington, D. M. (1994). *Neural networks: A review from a statistical perspective*. Statistical Science 9: 2–54.
- Cowpertwait and Metcalfe (2009). *Introductory Time Series with R.* Springer, doi:10.1007/978-0-387-88698-5. ASIN: 978-0-387-88697-8
- Cotter J. (2011). *Varying the VaR for Unconditional and Conditional Environments.* http://arxiv.org/abs/1103.5649

- Cotter J. and Dowd K. (2011). *Estimating financial risk measures for futures positions: a non-parametric approach.* http://arxiv.org/abs/1103.5666
- Christoffersen P. and Diebold F. (2000). *How Relevant is Volatility Forecasting for Financial Risk Management?* Review of Economics and Statistics 82 (1), 12-22. doi:10.1162/003465300558597
- Daemen J., Rijmen V. (2002), *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer. ISBN 3-540-42580-2.
- Daniel K., Hirshleifer D., and Subrahmanyam A. (1998). *Investor Psychology and Security Market under- and Overreactions.* The Journal of Finance 53 (6), 1839-85. http://sites.uci.edu/dhirshle/files/2011/02/Investor-Psychology-and-Security-Market-Under-and-Overreactions.pdf
- Diebold, F.X. Schuermann. T, Stroughair, J.D. (1998). *Pitfalls and Opportunities in the use of Extreme Value Theory in Risk Management.* New York University. NYU Stern. http://archive.nyu.edu/fda/bitstream/2451/27079/2/wpa98081.pdf
- Dowd K. and Cotter J. (2011). *Evaluating the Precision of Estimators of Quantile-Based Risk Measures,* Cornell University Library, http://arxiv.org/abs/1103.5665
- Engle, R. F. (1982). *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation*, Econometrica, Vol. 50, No. 4 , pp. 987-1007. Published by: The Econometric Society Stable URL: http://www.jstor.org/stable/1912773
- Engle, R. F.; Ng, V. K. (1991). *Measuring and testing the impact of news on volatility.* Journal of Finance 48 (5): 1749–1778. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=262096
- Haavelmo, T. (1943) *The statistical implications of a system of simultaneous equations*, *Econometrica* **11**:1–2. Reprinted in D.F. Hendry and M.S. Morgan (Eds.), *The Foundations of Econometric Analysis*, Cambridge University Press, 477—490, 1995.
- Hansen P. and Lunde A. (2005). *A forecast comparison of volatility models: does anything beat a GARCH(1,1)?* Journal of Applied Econometrics 20 (7), 873-89 . http://www.finanzaonline.com/forum/attachments/econometria-e-modelli-di-trading-operativo/1258338d1275910853-valutare-modelli-forecasting-volatilita-comparison-volatility-

models.pdf

- Harold A. Linstone, Murray Turoff (1975), *The Delphi Method: Techniques and Applications*, Reading, Mass.: Adison-Wesley, ISBN 9780201042948 (http://is.njit.edu/pubs/delphibook/)
- Heckman, J.J., and E Vytlacil (2005) *Structural Equations, Treatment Effects and Econometric Policy Evaluation***.** National Bureau of Economic Research, 669-738. http://www.nber.org/papers/w11259
- Lach S. (2002). *Existence and Persistence of Price Dispersion: An Empirical Analysis.* Review of Economics and Statistics 84 (3), 433-44. doi:10.1162/003465302320259457
- Lux, Thomas (2008). *The Markov-switching multifractal model of asset returns: GMM estimation and linear forecasting of volatility.* Journal of Business and Economic Statistics 26 (2): 194–210. doi:10.1198/073500107000000403.
- Marsaglia, G.; Zaman, A. (1991). *A new class of random number generators*. Annals of Applied Probability 1 (3): 462–480. doi:10.1214/aoap/1177005878
- Martin R. (2011). *Saddlepoint methods in portfolio theory.* Cornell University Library, http://arxiv.org/abs/1201.0106
- Matsumoto, M.; Nishimura, T. (1998). *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator.* ACM Transactions on Modeling and Computer Simulation 8 (1): 3–30. doi:10.1145/272991.272995
- McNeil, A. (1999). *Extreme value theory for risk managers.* Departement Mathematik ETH Zentrum, http://www.math.ethz.ch/~mcneil/ftp/cad.pdf
- McNeil, A.J., Frey, R. (2000). *Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach.* Journal of empirical finance (Elsevier) 7 (3-4), 271-300 http://www.sciencedirect.com/science/article/pii/S0927539800000128
- Mikosch, T. and Starica, C. (2004). *Nonstationarities in financial time series, the long-range dependence, and the IGARCH effects.* Review of Economics and Statistics,86,1,378-390.MIT Press. http://www.mitpressjournals.org/doi/abs/10.1162/003465304323023886

- Neely C. and Weller P. (2001). *Predicting Exchange Rate Volatility: Genetic Programming vs. GARCH and Risk Metrics*™. Working Paper Series (Federal Reserve Bank of St. Louis), http://research.stlouisfed.org/wp/more/2001-009
- Nelson, D. B. (1991). *Conditional heteroskedasticity in asset returns: A new approach.* Econometrica 59: 347–370.
- Park S. K. and Miller K. W. (1988). *Random Number Generators: Good Ones Are Hard To Find*. Communications of the ACM 31 (10): 1192–1201. doi:10.1145/63039.63042.
- Phelan, M. (1997). *Probability and statistics applied to the practice of financial risk management: The case of JP Morgan's RiskMetrics*™. Journal of Financial Services Research (Springer) 12 (2), 175-200. http://www.springerlink.com/content/g0p352u4360181x2
- Starica C. (2003). *Is Garch(1,1) as Good a Model as the Accolades of the Nobel Prize Would Imply?* Social Science Research Network, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=637322&
- Tsay R. S. (2005). *Analysis of financial time series.* Wiley-Interscience, ASIN: 978-0-470-41435-4
- Turoff M. (1970), *The Design of a Policy Delphi*, *Technological Forecasting and Social Change* 2, 2, (http://is.njit.edu/pubs/delphibook/ch3b1.pdf)
- Tuson A. and Ross P. (1998). *Adapting Operator Settings in Genetic Algorithms.* Evolutionary Computation 6 (2), 161-84. doi:10.1162/evco.1998.6.2.161
- Würtz, D. and Chalabi, Y. and Luksan (2006). *Parameter estimation of ARMA models with GARCH /APARCH errors an R and SPlus software implementation*. Journal of Statistical Software. 28—33, http://www-stat.wharton.upenn.edu/~steele/Courses/956/RResources/GarchAndR/WurtzEtAlGarch.pdf