FORECASTING THE EX-ANTE TRACKING ERROR FOR GLOBAL FIXED INCOME PORTFOLIOS: EMPHASIS ON THE ESTIMATION OF THE VARIANCE-COVARIANCE MATRIX

Patricia Vázquez Hinojosa

Trabajo de investigación 013/016

Master en Banca y Finanzas Cuantitativas

Tutores: Dr. Alfonso Novales

Universidad Complutense de Madrid

Universidad del País Vasco

Universidad de Valencia

Universidad de Castilla-La Mancha

www.finanzascuantitativas.com

Estimating the ex-ante tracking error: emphasis on the forecast of the variance and covariance matrix

PATRICIA VÁZQUEZ HINOJOSA

June 12, 2013

Master in Banking an Quantitative Finance

Tutor: Alfonso Novales Cinca
Universidad Complutense de Madrid

Universidad Complutense de Madrid
Universidad de Valencia
Universidad del País Vasco
Universidad de Castilla-La Mancha

www.finanzascuantitativas.com

Estimating the ex-ante tracking error: emphasis on the forecast of the variance and covariance matrix

PATRICIA VÁZQUEZ HINOJOSA[1]

June 12, 2013

Tutor: Alfonso Novales Cinca
Universidad Complutense de Madrid

Abstract

Due to the recent financial crisis risk management has gained much importance and has been put at the core of investment processes. One of the concerns that has raised in the asset management industry is the accuracy of the forecasted risk and the existing deviation between the ex-post or realised tracking error and the forecasted or ex-ante racking error. In this research a model for the forecast of the tracking error is proposed and tested with different volatility forecasting models.

[1] I would like to thank Credit Suisse AG, specially the Global Fixed Income Team, for giving me the opportunity to work on this investigation during the stage .

CONTENTS

1. **Introduction** ... 4

2. **Theoretical concepts** ... 6

2.1 Proposed ex-ante tracking error model ... 7

2.1.1 Model components – Variance and covariance matrix ... 8

2.1.1 Model components – Relative exposures to the risk factors ... 10

2.1.1 Model components – Relative sensitivity of the active returns towards the risk factors ... 10

2.2 Volatility forecasting models ... 11

2.2.1 The historical moving average model ... 12

2.2.2 The EWMA model ... 14

2.2.3 The DCC-GARCH(1,1) model ... 17

3. **Data specifications** ... 21

3.1 Factors description ... 21

3.2 Portfolio description ... 22

3.3 Assumptions made for the ex-ante and ex-post tracking error computations. 23

4. **Estimation of the active return's sensitivities towards the risk factors** ... 26

5. **Results** ... 33

5.1 Parameters of the volatility forecasting models ... 33

5.2 Backtesting – Mean Squared Error ... 36

6. **Conclusions** ... 41

7. **References** ... 43

8. **Appendix** ... 44

1. Introduction

The asset management industry has been changed by the economic crisis. As a matter of fact, risk management has gained much importance and has been put at the core of investment processes. For active portfolio management just as investment performance is naturally contrasted to a benchmark, so should the associated risk of a portfolio be. For this purpose, in this paper, a widely used statistic for risk measurement and risk allocation has been chosen, the ex-ante tracking error. We will define it as the forecasted deviation of the active return, which is the difference between the return obtained by the portfolio of the asset manager and the return generated by a benchmark. Expressed in other words, it provides an estimation of the future relative performance of a portfolio against a benchmark index and gives an idea about the risk profile of a managed portfolio as it measures the relative risk that has been taken against a specified benchmark.

Besides the ex-ante tracking error another widely used risk measure is the ex-post tracking error. The main difference between these two measures is that whilst the former has a forward-looking approach and tries to identify the standard deviation of future active returns based on the decisions made today on a portfolio, the latter one is backward-looking and reflects the decisions taken by a portfolio manager, with respect to the factors affecting the tracking error, for a specified past observation period. If a portfolio manager takes new decisions the ex-post tracking error would not give reliable information of future risk in the portfolio as it is computed using data from past periods.

Due to the strong volatility swings in the past years, the investment industry has become more concerned about the accuracy of the forecasted risk and the existing deviation between the ex-post or realised tracking error and the forecasted or ex-ante racking error. In this research we will thus propose a model for measuring the ex-ante tracking error, which will require risk forecasting. The aim of the investigation will be to contrast different volatility forecasting models and then choose the one delivering most adequate and accurate results by comparing the obtained ex-ante tracking errors with the realized tracking error, i.e. the ex-post tracking error corresponding to the forecast period. The study will focus specifically on actively managed Global Fixed Income Portfolios. This type of portfolios invest in developed countries all over the globe, even though we will focus on a limited number of markets, and mainly on sovereign debt, even though exposure to corporate fixed income securities is allowed.

In the next section we are going to review the theoretical concepts that are required to perform this investigation. It includes the already existing models, but also presents the model suggested in this research to estimate the ex-ante tracking error. In section 3 we specify the data that is used, this concerns the factor selection and portfolio data, and the assumptions that are made. Section 4 is dedicated to the estimation of sensitivities that will be required in the model for the tracking error forecast. In section 5 we will explain the obtained results. Finally, in section 6 we will make a concluding summary and outline related lines of research.

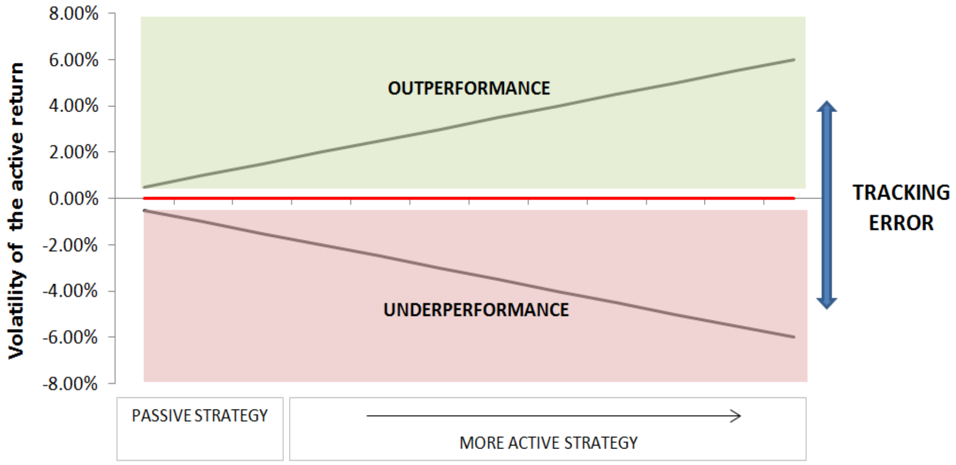
2. Theoretical concepts

In the asset management industry there are different kinds of fixed income portfolio managers, it is relevant to have a look at this different types of management as a different result for the ex-ante tracking error is expected for each of those. In order to measure a portfolio manager's performance, a benchmark is chosen. In the fixed income industry the portfolio is benchmarked against liabilities or a bond market index. There are three principal portfolio management strategies. The first one, is the pure bond index matching strategy which tries to replicate the performance of the benchmark as exactly as possible and involves hence, the least risk of underperformance. The second one, the enhanced indexing strategy, attempts to achieve the least deviation from benchmark by matching the main risk factors influencing its performance but without acquiring each of the issues in the selected bond market index. Finally, there is the active management strategy, which is the one I will focus on throughout the paper. The purpose of this one is to outperform the benchmark by constructing a portfolio that will differ much more from the benchmark that the enhanced index strategy and will take bets on the different factors affecting the performance.[2] By bets we mean the decision of an active portfolio manager to divert to some degree from the factors explaining the benchmark returns. These decisions are based on an exhaustive analysis of how future return determinants may develop. The decisions taken can either lead to outperform the benchmark, which is the goal of actually taking these bets, or to an underperformance. The factors that are hence influenced by the portfolio manager are indeed risk factors.

Having this in mind it is simple to see how the ex-ante tracking error is related to this kind of management strategies. The higher the value of the tracking error the more active is the management strategy pursued by the portfolio manager:

[2] For further details see Focardi, Sergio M. And Fabozzi, Frank J. (2004), *The Mathematics of Financial Modelling & Investment Management*, John Wiley & Sons, Inc.

Graph 1: Tracking Error depending on management style



The ex-ante tracking error is a function of the portfolio and benchmark weights, the volatility of the assets and the correlation across assets. Based on these determinants, the ex-ante tracking error would be computed as follows:

$$EATE = \sigma_p^2 + \sigma_B^2 - 2\rho\sigma_P\sigma_B \qquad [1]$$

σ_p^2: Portfolio variance / σ_P = portfolio standard deviation
σ_B^2: Benchmark variance / σ_B = benchmark standard deviation
ρ: Correlation between portfolio and benchmark returns

However, as this study focuses on predicting the ex-ante tracking error of actively managed portfolios, where portfolio managers don't replicate the benchmark by keeping their portfolios on the same risk factor levels, but try to outperform it by taking bets on some of these risk factors, hence deviating from the risk factor levels, we will make the standard deviation of the active returns depend on those factors on which active portfolio managers actually take these bets. Indeed, returns of a portfolio can be explained by the returns of each of its securities or by a set of factors explaining the evolution of returns, achieving herewith a reduction of dimension of the problem.

2.1 Proposed ex-ante tracking error model

Keeping in mind that we will be working under the approach of explaining active returns and thus the tracking error by considering different risks for which the active portfolio managers take bets, we can move on to explaining the model we are going to implement to estimate the ex-ante tracking error. The model proposed to compute the ex-ante tracking error is shown in the following equation:

$$EA\,TE = ((\Omega - ones) \circ \Theta) \times \textstyle\sum_t \times ((\Omega - ones) \circ \Theta)' \qquad [2]$$

Ω:	Relative exposure between portfolio and benchmark to the risk factors with dimension 1×N, where N = number of risk factors considered
Θ:	Weighted sensitivity vector of the active returns to the risk factors with dimension 1×N
ones:	Vector of ones with dimension 1×N
$\sum_t$:	Variance – covariance matrix in period t of the risk factors composing the multi factor model with dimension N×N
∘:	Multiplication symbol indicating the multiplication of vectors element by element

2.1.1 Model components – Variance and covariance matrix

The accuracy of the resulting ex-ante tracking error will mostly (in the literature there is a general agreement on the ex-ante tracking error being underestimated, several other explanations for its underestimation, such as the stochastic nature of the portfolio weights, have been proved) depend on the estimation of the volatility and correlations across the chosen risk factors. It is hence on these estimations where we will focus to obtain an accurate ex-ante tracking error.

Because the centre of attention of this study is put on the forecast of the variance-covariance matrix we will quickly have a look at its basic properties and its composition. It is a square, symmetric matrix of variances and covariances, in this case of a set of risk factors, given by:

$$\text{Var - Covar} = \begin{pmatrix} \sigma_1^2 & \cdots & \sigma_{1N} \\ \sigma_{21} & \cdots & \sigma_{2N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & \sigma_N^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \cdots & \rho_{1N}\sigma_1\sigma_N \\ \rho_{21}\sigma_2\sigma_1 & \cdots & \rho_{2N}\sigma_2\sigma_N \\ \vdots & \ddots & \vdots \\ \rho_{N1}\sigma_N\sigma_1 & \cdots & \sigma_N^2 \end{pmatrix} \qquad [3]$$

The variance – covariance matrix can hence be expressed as:

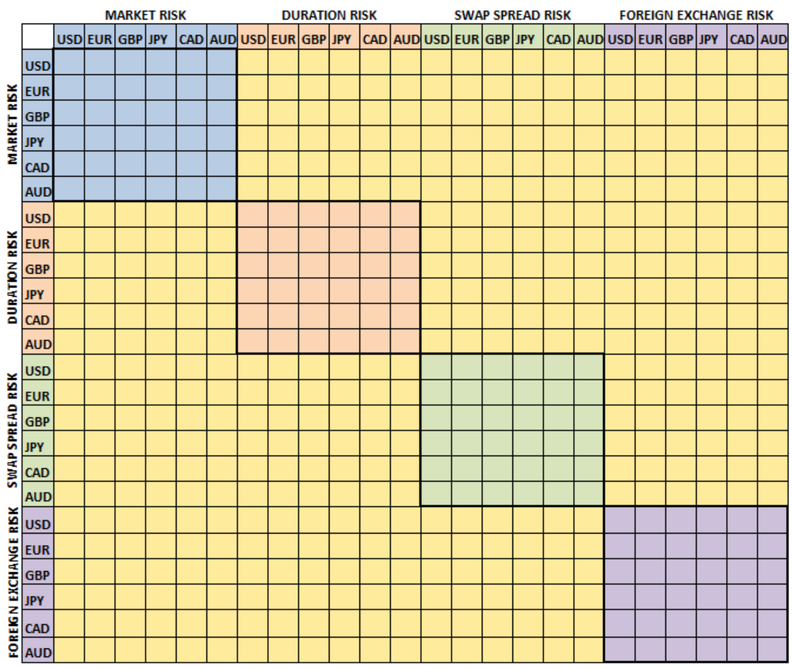
$$\begin{pmatrix} \sigma_1^2 & \cdots & \sigma_{1N} \\ \sigma_{21} & \cdots & \sigma_{2N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & \sigma_N^2 \end{pmatrix} = \begin{pmatrix} \sigma_1 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_N \end{pmatrix} \times \begin{pmatrix} 1 & \cdots & \rho_{1N} \\ \rho_{12} & \cdots & \rho_{2N} \\ \vdots & \ddots & \vdots \\ \rho_{1N} & \cdots & 1 \end{pmatrix} \times \begin{pmatrix} \sigma_1 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_N \end{pmatrix} \quad [4]$$

As we have just seen, the variance and covariance matrix is simply a mathematically convenient way to express a factor's volatilities and correlations.

We need to be aware that volatilities and correlations of financial and economic variables are parameters that are not observable and can therefore only be forecasted in the context of a model. Different methodologies will be tested to obtain the variance-covariance matrix, in particular we will test the historical moving average model, the EWMA model and the DCC-GARCH(1,1) model.

Now that we have had a look at the main characteristics of the variance and covariance matrix, it is interesting to have a look at its composition for our concrete case. The variance and covariance matrix will contain the risks into which a portfolio manager will incur when taking bets. The matrix will have a structure in blocks. If we wanted to know the contributions of a specific risk factor to the total ex-ante tracking error we would need to sum up the components of the block in the diagonal corresponding to the specific risk factor and the correlation it has with the other risk factors. To understand this notion better it is depicted graphically:

Graph 2: Decomposition of the variance and covariance matrix



2.1.1 Model components – Relative exposures to the risk factors

The relative exposures between portfolio and benchmark to each risk factor are obtained dividing the share of the portfolio exposed to a concrete risk factor by the share of the benchmark exposed to this same risk factor. If the results of all relative exposures are equal to one, the benchmark is being exactly replicated and the ex-ante tracking error should hence be equal to zero. If we left the exposure vector as it is and supposed that the benchmark is perfectly replicated the ex-ante tracking error would however be positive, concretely it would be the sum of all the components in the variance and covariance matrix. In order to correct this effect we will subtract a matrix of ones from the relative exposures vector, leaving like this only the percentage deviation of the portfolio towards the exposure of the benchmark against the risk factors.

2.1.1 Model components – Relative sensitivity of the active returns towards the risk factors

It is plausible to think, that taking a bet in one concrete risk factor, will not have the same strength of impact in the active return and hence on the tracking error. Therefore, as a new approach, the ex-ante tracking error has been made dependent on the sensitivities of the active return to the different risk sources. To estimate these sensitivities, a multiple lineal regression model[3] will be used:

$$r_{P,t} - r_{B,t} = \alpha + \beta_1 F_1 + \cdots + \beta_N F_N + \varepsilon_t \qquad [5]$$

$r_{P,t}$:	Portfolio returns in period t
$r_{B,t}$:	Benchmark returns in period t
α:	Constant variable
$\beta_{1,\ldots,N}$:	Sensitivities of the active returns to the selected risk factors
ε_t:	Error term in period t

However, what we want to take into consideration in the estimation of the ex-ante tracking error are not the sensitivities themselves. This would distortion the scale of the tracking error in an adequate manner. Instead, we are going to weight the sensitivities to

[3] For more information on these type of models see: Novales, A. (2000), *Econometría*, Segunda Edición, McGraw-Hill.

the factors according to the level each parameter is above or below the mean value of all the coefficients, in order for the sensitivities to be centred around one. If our sensitivity vector is defined as:

$$\Theta = (\beta_{1,\dots,30}, \gamma_{1,\dots,6}, \delta_{1,\dots,6}, \theta_{1,\dots,5})$$

we will perform the following transformation:

$$\tilde{\Theta} = \frac{\Theta}{mean(\Theta)} \qquad [6]$$

$\tilde{\Theta}$: Transformed sensitivity vector

2.2 Volatility forecasting models

In this section we are going to describe in detail the three models we are going to use to forecast the variance and covariance matrix. We will look at the analytical expressions and comment their properties, advantages and disadvantages.

But to be able to accomplish the forecast for any of the volatility forecasting methodologies, we need first to model the behaviour of the movements of the percentage changes of the risk factors and the portfolio and benchmark returns. The model needs to consider the temporal dynamics of returns, as well as its distribution at any point in time. A widely used model to characterize the development of price returns, yields and foreign exchange rates is the random walk model[4]. A random walk is the process by which the variable follows a path consisting of a sequence of discrete steps with a fixed length. Each of the steps occurs randomly. The analytical expression is:

$$x_{i,t} = \mu + x_{i,t-1} + \sigma\varepsilon_t \qquad [7]$$

$x_{i,t}$: Percentage change of factor i in time period t
μ: Mean of $x_{i,t}$
$x_{i,t-1}$: Percentage change of factor i in time period t-1

[4] For details about the model see: Hull, John, *Options Futures and Other Derivatives*, Seventh Edition, 2009, Pearson Prentice Hall.

σ:	Standard deviation of $x_{i,t}$
ε_t:	Random variable where $\varepsilon_t \sim iid\ N(0,1)$

As we can observe in the above expression the changes are assumed to have a constant variance, which is not ideal for financial time series data. We are going to relax this assumption and let the variance change with time:

$$x_{i,t} = \mu + x_{i,t-1} + \sigma_t \varepsilon_t \qquad [8]$$

2.2.1 The historical moving average model

We will start to compute the ex-ante tracking error forecasting the variance and covariance matrix with the simplest model that we want to analyze, which is the historical moving average volatility model[5]. To understand this model, let's recall the expression used to compute the standard historical variance:

$$\sigma_{i,T}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(x_{i,t} - \bar{x}_i)^2 \qquad [9]$$

$\sigma_{i,T}^2$:	Historical variance for time period T of factor i
T:	Observation period
$x_{i,t}$:	Percentage change of factor i in time period t
$\bar{x}_i$:	Mean of the percentage change of factor i

For almost any conceivable financial and economic series and across all frequencies, volatility varies over time. This is caused by phenomena such as volatility clustering where periods of high volatility are followed by periods of high volatilities and the other way round. To capture the inconstant nature of volatility in this model, instead of computing the variance for the whole observation period as in the standard historical variance model, a fixed number of observations is chosen, a so called window, for which the variance is

[5] For further details see: Gabudean, Radu and Schuehle, Niels (2011), *Volatility Forecasting: A Unified Approach to Building, Estimating and Testing Models*, Barclays Capital.

computed. In such a way, if we compute the variance in t and the window length is for example 20, the observations considered lie between t-19 and t. The variance in t+1 would be computed with the observations between t-18 and t+1. As we can see, the last observation in the sample length falls out to be replaced by the most recent one. In terms of a forecasted variance, the expression for the historical moving average model is:

$$\hat{\sigma}^2_{i,T+1} = \frac{1}{M-1} \sum_{t=T-M+1}^{T} (x_{i,t} - \bar{x}_{i,S})^2 \qquad [10]$$

$\hat{\sigma}^2_{i,T+1}$:	Forecasted variance for time period T+1 of factor i
M:	Window length
$x_{i,t}$:	Percentage change of factor i in time period t
$\bar{x}_{i,S}$:	Is the mean of the percentage change of factor i comprised in the window, where S = {T-M+1, T}

Even though volatility is not constant over time, it has the feature to not change abruptly. Therefore, its level in the recent past still forecasts the future. The model takes strong advantage of this characteristic known as persistence over time.

For the forecast of covariances the model takes the following form:

$$\hat{\sigma}_{ij,T+1} = \frac{1}{M-1} \sum_{t=T-M+1, i\neq j}^{T} (x_{i,t} - \bar{x}_{i,S})(x_{j,t} - \bar{x}_{j,S}) \qquad [11]$$

$\hat{\sigma}_{ij,T+1}$:	Forecasted covariance for time period T+1 between the factors i and j
M:	Window length
$x_{i,t}$:	Percentage change of factor i in time period t
$x_{j,t}$:	Percentage change of factor j in time period t
$\bar{x}_{i,S}$:	Mean of the percentage changes of factor i comprised in the window, where S = {T-M+1, T}
$\bar{x}_{j,S}$:	Mean of the percentage changes of factor j comprised in the window, where S = {T-M+1, T}

However, this model has some disadvantages like the fact that it reacts to increases in volatility with some delay as we are dealing with the average of volatility levels of the last

M days. It also equally weights the M days selected for the calculus, so for example the presence of one day of high volatility will increase the forecasted volatility and will tend to maintain this high volatility level during M days to reduce drastically afterwards. Another problem that this model entails is the specification of the window length. A shorter sample length assigns more weight to recent data delivering a timelier estimate whilst a longer sample length provides more precision. To determine the optimal value we will use a parametric method, the quasi-maximum likelihood estimation method[6]. We will assume that the parameter has a normal distribution. This assumption can be accepted due to the "Law of Large Numbers" that states that as the sample size is increased the estimated parameter converges to its true value even though if in reality it does not have a normal distribution. The log-likelihood function that we want to optimize looks as follows:

$$L_{QML} = -\frac{1}{2}\sum_{t}(nlog(2\pi) + log|\Sigma_t(\mathrm{M})| + x_t'(\Sigma_t(M))^{-1}x_t) \qquad [12]$$

We can see that the parameter we want to estimate, the window length, is part of the variance-covariance matrix. Of course, volatility is affected by many other factors, like past data, but for this estimation technique the relevant issue is that there is a parameter M that actually will influence the volatility estimate and how it will be affected by it. The interest does not rely on how the effect comes to be. As a result we can observe a clear separation between the model we use to forecast volatility and how the associated parameter is estimated. This parameter estimation method can hence be used for other volatility estimation methods and this is actually what we are going to do. Remember that the variance and covariance matrix is symmetric and semi-definite positive, the estimation of parameters via the quasi-maximum likelihood method is thus computationally possible.

2.2.2 The EWMA model

As we have seen, the historical moving average model confers to each observation the same weight. Any event has the same influence on the volatility from the day it starts being part of the window until it exits the sample again. The Exponentially Weighted Moving Average Model[7] improves this issue by giving higher weight and hence more

[6] For more details see: Gabudean, Radu and Schuehle, Niels (2011), *Volatility Forecasting: A Unified Approach to Building, Estimating and Testing Models*, Barclays Capital.

[7] For more information see: Mina, Jorge and Xiao, Jerry Yi (2001), *Return to RiskMetrics: The Evolution of a* Standard, RiskMetrics Group, Inc.

importance to more recent data. This model defines the weights as the power of a constant parameter between 0 and 1, which makes the weight of the observations decrease exponentially:

$$\sigma_{i,T+1}^2 = (1-\lambda)\sum_{n=0}^{\infty}\lambda^n (x_{i,T-n} - \bar{x}_i)^2 \qquad [13]$$

In practice the above expression needs to be truncated, leading to:

$$\sigma_{i,T+1}^2 = (1-\lambda)\sum_{n=0}^{m-1}\lambda^n (x_{i,T-n} - \bar{x}_{i,S})^2 + \lambda^m \frac{1}{T}\sum_{l=1}^{T}(x_{i,l} - \bar{x}_{i,T})^2 \qquad [14]$$

$\hat{\sigma}_{i,T+1}^2$:	Forecasted variance for time period T+1 of factor i
λ:	Parameter between 0 and 1 that defines the weights of the observations
$x_{i,t}$:	Percentage change of factor i in time period t
$\bar{x}_{i,S}$:	Mean of the percentage changes of factor i comprised in the sample window, where S = {T-M+1, T}

The term $\sum_{N=1}^{T}(x_{i,N} - \bar{x}_{i,T})^2$ is the sample variance of the time series and can be interpreted as the level of long term variance.

Note that the total sum of the weights needs to be equal to one:

$$(1-\lambda)\sum_{n=0}^{M-1}\lambda^n = 1 - \lambda^m \qquad [15]$$

Going back to equation 9, it can be written down as:

$$\sigma^2_{i,T+1} = (1-\lambda)((x_{i,T} - \bar{x}_{i,S})^2 + \lambda(x_{i,T-1} - \bar{x}_{i,S})^2 + \lambda^2(x_{i,T-2} - \bar{x}_{i,S})^2 + \lambda^3(x_{i,T-3} - \bar{x}_{i,S})^2 + \cdots + \lambda^{m-1}(x_{i,T-m-1} - \bar{x}_{i,S})^2) + \lambda^m \sigma^2_{i,T}$$

$$= (1-\lambda)\,(x_{i,T} - \bar{x}_{i,S})^2 + \lambda(1-\lambda)\,(x_{i,T-1} - \bar{x}_{i,S})^2 + \lambda^2(1-\lambda)(x_{i,T-2} - \bar{x}_{i,S})^2 + \cdots + \lambda^{m-1}(1-\lambda)(x_{i,T-m-1} - \bar{x}_{i,S})^2$$

$$= (1-\lambda)\,(x_{i,T} - \bar{x}_{i,S})^2 + \lambda(1-\lambda)\sum_{n=0}^{m-1} \lambda^n (x_{i,T-n} - \bar{x}_{i,S})^2 + \lambda^m \sigma^2_{i,T}$$

$$= (1-\lambda)\,(x_{i,T} - \bar{x}_{i,S})^2 + \lambda(1-\lambda^m)\sigma^2_{i,T} + \lambda^m \sigma^2_{i,T}$$

Introducing the expression 12 we get to the usual EWMA expression:

$$\hat{\sigma}^2_{i,T+1} = (1-\lambda)\,(x_{i,T} - \bar{x}_{i,S})^2 + \lambda\sigma^2_{i,T} \qquad [16]$$

$\hat{\sigma}^2_{i,T+1}$:	Forecasted variance for time period T+1 of factor i
λ:	Parameter between 0 and 1 that defines the weights of the observations
$x_{i,t}$:	Percentage change of factor i in time period t
$\bar{x}_{i,S}$:	Mean of the percentage changes of factor i comprised in the sample window, where S = {T-M+1, T}
$\sigma^2_{i,T}$:	Volatility of factor i for the day T

To forecast covariances the model takes the following expression:

$$\hat{\sigma}_{ij,T+1} = (1-\lambda)(x_{i,T} - \bar{x}_{i,S})(x_{j,T} - \bar{x}_{j,S}) + \lambda\sigma_{ij,T} \qquad [17]$$

Besides the positive characteristic of attributing more weight to recent information, this model has other advantages like the fact that it doesn't need a big amount of data given that the power of λ will converge to 0 with the increase of the time periods considered. In essence the model considers an infinite horizon and thus avoids the problem of having to select a sample length. Another positive aspect is that once the volatility has been computed for a specific day, the formula doesn't require historical input again, the storage of big amounts of data is not necessary. At any point in time, we only need to remember the current estimated variance and the most recent observation. Finally, it only needs the estimation of one single parameter λ. The closer the parameter to 1, the more persistent is the estimated volatility, meaning that the estimator is not strongly affected by recent and current events. The EWMA approach was proposed by the RiskMetrics Team, who uses a λ of 0.94 for daily variance estimations. In this paper we will estimate the parameter via the quasi-maximum likelihood estimation method, as we did for the estimation of the window length in the previous section. The volatility in the log-likelihood function will now depend on the parameter λ:

$$L_{QML} = -\frac{1}{2}\sum_{t}\left(nlog(2\pi) + log|\Sigma_t(\lambda)| + x_t'(\Sigma_t(\lambda))^{-1}x_t\right) \qquad [18]$$

Even though this model improves the historical moving average model it still carries some inconvenience. The main problem is that even if it captures the persistent character of volatility it doesn't incorporate its mean reverting nature. The EWMA model doesn't include a constant that can serve as a reference level for long term volatility.

2.2.3 The DCC-GARCH(1,1) model

Before we explain the DCC-GARCH(1,1) model it is best to have first a look at the more simple univariate GARCH(1,1) model. In contrast to the EWMA model, the Generalized Autoregressive Conditional Heteroskedastic model, proposed by Bollerslev in 1986, includes a weighted long term average variance rate:

$$\hat{\sigma}_{i,T+1}^2 = \gamma V_L + \alpha(x_{i,T} - \bar{x}_{i,S})^2 + \beta\sigma_{i,T}^2 \qquad [19]$$

$\hat{\sigma}_{i,T+1}^2$:	Forecasted variance for time period T+1 of factor i
γ, α, β:	Weight parameters where $\gamma + \alpha + \beta = 1$
V_L:	Long term or unconditional variance; γV_L is often expressed as ω
$x_{i,t}$:	Percentage change of factor i in time period t
$\bar{x}_{i,S}$:	Mean of the percentage changes of factor i comprised in the sample window, where S = {T-M+1, T}
$\sigma_{i,T}^2$:	Volatility of factor i for the day T

Note that when $\gamma = 0$, $\alpha = 1 - \lambda$ and $\beta = \lambda$, the model becomes the EWMA model which is essentially a particular case of the GARCH(1,1) model. Having a closer look at the parameters we can say that α reflects the "reaction" of the volatility to market changes, the higher its value the more sensitive is the volatility to those changes. The β contemplates the persistence of volatility, if it has a high value it signals that shock in the conditional variance take a long time to disappear. The parameter γ that we can write as $1 - \alpha - \beta$ is the rate of reversion to the long run variance. The higher the sensitiveness to shocks and the higher the persistence, the lower will be the rate of reversion. In the EWMA model the persistence is equal to one, so the model expects that any shock in volatility will persist forever and hence, any increase in volatility will raise the estimate of all future. For the variance and covariance matrix generated by the GARCH(1,1) model to be semi-definite positive the following conditions are sufficient: $\omega \geq 0$, $\alpha \geq 0$, $\beta \geq 0$.

In 2001 Engle and Sheppard introduce the Dynamic Conditional Correlation model[8]. It is a multivariate GARCH model designed to estimate large time varying covariance matrices. The model aims to simplify the estimation of multivariate conditional variance by estimating univariate GARCH models for each factor and using the resulting transformed residuals to estimate a conditional correlation estimator. It is thus a model that is estimated in two steps.

To apply this model we need to work with the standardized values of the percentage changes $x_{i,t}$, which we assume again to be normally distributed. Writing $x_{i,t}$ as the conditional standard deviation times the standard disturbance, which has mean zero and variance one for each series, we have:

$$x_{i,t} = \sqrt{\sigma_{i,t}^2}\varepsilon_{i,t} \rightarrow \varepsilon_{i,t} = \frac{x_{i,t}}{\sigma_{i,t}} \qquad [20]$$

$\varepsilon_{i,t}$: Standardized disturbance with mean zero and variance one

In the DDC-GARCH model the covariance matrix can be decomposed as follows:

$$H_t = D_t R_t D_t \qquad [21]$$

[8] Engle, Robert and Sheppard, Kevin (2001), *Theoretical and Empirical Properties of Dynamic Conditional Correlation Multivariate GARCH*, National Bureau of Economic Research.

where D_t is a diagonal matrix containing the time varying standard deviations of the factors that are computed in the univariate GARCH model:

$$D_t = \begin{bmatrix} \sigma_{1,t} & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{2,t} & 0 & \cdots & 0 \\ 0 & 0 & \sigma_{3,t} & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{N,t} \end{bmatrix} \qquad [22]$$

and R_t represents the conditional correlation matrix of the standardized disturbances $\varepsilon_{i,t}$:

$$R_t = \begin{bmatrix} 1 & q_{12,t} & q_{13,t} & \cdots & q_{1N,t} \\ q_{21,t} & 1 & q_{23,t} & \cdots & q_{2N,t} \\ q_{31,t} & q_{32,t} & 1 & \cdots & q_{3N,t} \\ \vdots & \vdots & & \ddots & \vdots \\ q_{N1,t} & q_{N2,t} & q_{N3,t} & \cdots & 1 \end{bmatrix} \qquad [23]$$

We can see that the conditional correlation is the conditional covariance between the standardized disturbances:

$$\varepsilon_t = D_t^{-1} x_t \sim N(0, R_t) \rightarrow E_{t-1}(\varepsilon_t \varepsilon_t') = D_t^{-1} H_t D_t^{-1} = R_t \qquad [24]$$

As the variance and covariance matrix needs to be semi-definite positive, to accomplish this R_t, also needs to be semi-definite positive. Moreover, being R_t a correlation matrix all its components must be equal or less than one. There are two most common ways to parameterize the conditional correlation matrix. The most simple is via the exponential smoother, which is a geometrically weighted average of standardized residuals. We are going to use its alternative, the GARCH(1,1) model:

$$\hat{q}_{ij,T+1} = \bar{\rho}_{ij} + \alpha\left(\varepsilon_{i,T}\varepsilon_{j,T} - \bar{\rho}_{ij}\right) + \beta\left(q_{ij,T} - \bar{\rho}_{ij}\right)$$

$$= \bar{\rho}_{ij}\left(\frac{1-\alpha-\beta}{1-\beta}\right) + \alpha \sum_{s=1,\infty} \beta^s \, \varepsilon_{i,T-s}\varepsilon_{j,T-s} \qquad [25]$$

where the correlation estimator has the following expression:

$$\rho_{ij,t} = \frac{q_{ij,t}}{\sqrt{q_{ii,t}q_{jj,t}}} \qquad [26]$$

$\hat{q}_{ij,T+1}$:	Estimated covariance of the standardized disturbances
$\bar{\rho}_{ij}$:	Unconditional expectation of the cross product, for variances $\bar{\rho}_{ii} = 1$. Each term in the denominator has an expected value equal to one.
α, β:	Weight parameters
$\varepsilon_{i,T}$:	Standardized disturbance of factor i at time T
$\varepsilon_{j,T}$:	Standardized disturbance of factor j at time T

Because the covariance matrix $Q_t = |q_{ij,t}|$ is a weighted average of a positive definite and a semi-positive definite matrix, the correlation matrix will be positive definite as well.

As long as α + β < 1 the model will be mean reverting to the unconditional expectation $\bar{\rho}$, with β standing for the persistence of the volatility and α reflecting the reaction of the volatility to changes in the market.

The parameters will again be estimated via the quasi-maximum model, expressed as:

$$L_{QML} = -\frac{1}{2}\sum_t (nlog(2\pi) + log|H_t| + x_t' H_t^{-1} x_t)$$

$$= -\frac{1}{2}\sum_t (nlog(2\pi) + log|D_t\, R_t D_t| + x_t' D_t^{-1} R_t^{-1} D_t^{-1} x_t)$$

$$= -\frac{1}{2}\sum_t (nlog(2\pi) + 2log\,|D_t| + log|R_t| + \varepsilon_t' R_t^{-1} \varepsilon_t \qquad [27]$$

3. Data specifications

In this section we are going to specify the risk factors we are going to use, based on the type of securities we are treating. We are also going to give details about the data used to perform the research, as well as the assumptions that are made in the different stages of the analysis.

3.1 Factors description

As we are focusing on fixed income portfolios we need to select factors that have an influence on the returns of this concrete type of securities. The factors influencing fixed income returns have been widely studied and there is a general consensus on those being mainly influenced by changes in the yield curve such as shift, twist and butterfly effects, which have an imminent impact on prices of fixed income securities, as well as credit risk. These return determinants need to be included in the model as risk factors as we aim to capture the risk, or in other words the deviation of active returns i.e. the ex-ante tracking error, taken by active portfolio managers, by making decisions on the degree of deviation from the factor levels explaining the benchmark returns. In this research they will be represented as market risk, duration risk and swap spread risk. We need to recall that we are dealing with international portfolios; we have therefore selected six different developed countries' markets: the Euro-zone, UK, the US, Japan, Australia and Canada. Their currencies are: USD, EUR, GBP, JPY, CAD and AUD. For each of these markets we will need to analyse the different risk factors as they vary from market to market. The domestic market will be the United States and hence the domestic currency will be the USD.

The first factor, the market risk, will capture the effects of risk due to changes in the yield curve level, slope and curvature. To reproduce this risk the LIBOR rate for each specific market has been taken. As we want to reflect the risk throughout the whole yield curve, in each market the 3 months and the 2, 5, 10 and 30 years LIBOR rate series have been selected. The next risk factor, the duration risk, will capture the sensitivity of securities to changes in the level of interest rates. To account for duration risk, the time series of the Macaulay duration and the yield to maturity of the benchmark have been taken. Both the Macaulay duration and the yield to maturity are available by country of risk, so the data has been taken for the countries subject to study. Out of this data the modified duration will be calculated, which is the data that is finally used to pattern duration risk. And last but not least, the swap spread, measuring the spread between points on the swap curve and points on the sovereign curve in a concrete market, will be used as a proxy to explain credit risk[9]. The concrete point selected is the 10 year swap spread. Because we are dealing specifically with global portfolios, an extra factor influencing returns and adding

[9] See Lang et al. 1998 and the Barra Risk Model Handbook.

risk to the portfolio will be needed to be taken into account, the foreign exchange risk. This involves five additional risk sources, one for each investment outside the domestic country. The foreign exchange rates of the EUR, GBP, JPY, CAD and AUD against the USD will therefore be included to capture this risk.

3.2 Portfolio description

To make this investigation, two real international portfolios will be used. They will be called portfolio 1 and portfolio 2. They have been taken at four different points in time; one just prior to this century's financial crisis on the 31.12.2007, the next in the middle of the crises on the 31.08.2008, another one on the 30.06.2009 and one last point in time after the big momentum of the crisis, concretely on the 30.06.2011. The idea behind taking these concrete moments in time is to see whether the macroeconomic environment influences the selection of the model computing the ex-ante tracking error.

All the selected portfolios benchmark against the JPMorgan Global Traded Index. Bond market indices are a method to measure the value of a specific section of the bond market. It is usually computed as a weighted average of market capitalization from the prices of the bonds eligible for the index. The JPMorgan Global Traded Index considers government bonds valued in local currencies for the following developed markets: the Euro-zone, UK, the US, Japan, Australia, Canada, Denmark and Sweden. However, over 98% of its total share is invested in the markets we are treating, therefore, it fits our purposes quite well. The securities issued outside the markets we are analyzing won't be considered in the analysis.

Concerning the data range and frequency for each portfolio return and risk factor series and each moment in time in which we are going to analyze the different portfolios, we will always work with daily data for the last 5 years excluding Saturdays and Sundays. Thus, if we are working with portfolio 1 in time period 31.12.2007, the data used in the analysis will range from 01.01.2002 to 31.12.2007. Finally, the described time series will be converted into the percentage changes of the variables from one period to the next before working with them. This means, if we have an observation on day t and an observation on day t-1 of factor i, the percentage change in t of this factor will be:

$$x_{i,t} = \frac{u_{i,t} - u_{i,t-1}}{u_{i,t-1}} \qquad [28]$$

All the required series have been downloaded from Bloomberg and all the programming will be done in R.

3.3 Assumptions made for the ex-ante and ex-post tracking error computations

For each of the selected points in time where we have taken the status of the portfolios subject to study, we will always compute three forecasting horizons; two weeks, one month and one year. To obtain the variance and covariance matrix on T+10 and T+20 (recall that we consider a 5-day week), where T is the evaluation date, we forecast the volatility on a daily basis, meaning T+1, T+2 until T+10 and T+20. The variances and covariances in the matrix are daily estimations. So once we have forecasted the matrix in T+10 and T+20 we need to multiply the values by the square root of 10 and 20 respectively to convert the components into biweekly and monthly variances and covariances. To obtain the ex-ante tracking error in one year we will simply annualize the variance and covariance matrix obtained for T+20 by multiplying it by the square root of twelve.

For each of these forecasting horizons and each forecasting methodology, the ex-ante tracking error will be computed 100’000 times by giving the bets a random structure. This means that the relative exposures will change randomly. However, the source of randomness will only be the exposure of the portfolio to the risk factors as it is the only exposure that a portfolio manager can influence, the exposure of the benchmark will hence be kept constant as it is found on the period of time in which we are making the analysis. There will however be some restrictions on the random character of the portfolio weights. Each time exactly five factor exposures will be allowed to change. Also, if we increase or decrease the exposure to some factor, the exact opposite will be done with a factor of the same category. For example, if the exposure to AUD is incremented by 5%, the exposure to another currency will be needed to be reduced by another 5%. In total 10 risk factor weights will be changed in each iteration. The reason behind this restriction is the fact, that at the time a decision is taken, we only have a concrete and limited amount of assets available.

We will consequently obtain 100’000 different ex-ante tracking errors for each volatility model and period of time of analysis. On these results we will finally perform back testing against the realized tracking errors, i.e. the ex-post tracking errors. This risk measure will be obtained by calculating the volatility of the difference of the total portfolio’s and benchmark’s return, with the historical volatility model:

$$\sigma_{i,T}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(x_{i,t} - \bar{x}_i)^2 \qquad [29]$$

$\sigma_{i,T}^2$: Historical variance for time period T of factor i

T: Observation period
$x_{i,t}$: Percentage change of factor i in time period t
$\bar{x}_i$: Mean of the percentage change of factor i

The analytical expression for the ex-post tracking error[10] is:

$$EP\ TE = \sigma(r_{P,t} - r_{B,t}) \qquad [30]$$

$r_{P,t}$: Portfolio returns for a given time period
$r_{B,t}$: Benchmark returns for a given time period

To make a proper comparison with the different forecast horizons, we need to compute the two weeks, one month and one year backward looking tracking errors for each portfolio and period of analysis. As we use daily data, with equation 7 we will obtain daily volatility estimates. We need therefore to transform the results into the correct volatility horizon as we do for the estimated variance and covariance matrix. If T is the period of analysis of the portfolio and we want to compute the monthly ex-post tracking error, we obtain first the daily volatility in T+20 and multiply the results by the square root of 20. Consequently, for the biweekly and annual ex-post tracking errors the daily volatilities in T+10 and T+250 need to be multiplied respectively by the square root of 10 and 250. As there is no standard about the data window that needs to be used to compute the standard deviation, we will use different sample windows and have a look at how this affects the results. Concretely the used windows will contain 30, 60, 120 and 250 observations respectively.

If a portfolio manager takes different bets, different securities or amounts of securities will be needed to achieve the level of risk on a specific risk factor. The portfolio returns depend thereby on the bets taken. That being so, the 100'000 vectors of relative exposures used in the computation of the ex-ante tracking error will be stored and used to weight the different securities contained in the portfolio in such a way, that the amount of risks taken are met. The way this is done is by defining matching conditions between the risk in which we incur and the securities. For example, if the bet is to increase 10 year euro bonds by some percentage, those securities trading in Euros and maturing in 10 years will be filtered out and incremented in the same proportion. The sum of the increments in the

[10] For further details see: Focardi, Sergio M. And Fabozzi, Frank J. (2004), *The Mathematics of Financial Modeling & Investment Management*, John Wiley & Sons, Inc.

filtered bonds will be equal to the total percentage of the bet taken. At the end we will also have 100’000 ex-post tracking errors for each portfolio and analysed time period.

On these two risk measures we will perform backtesting to draw conclusions on which volatility model gives the best results under which circumstance and forecasting horizon.

4. Estimation of the active return's sensitivities towards the risk factors

To estimate the sensitivities of the active return towards the different factors in section 2.1.1 we propose a multiple linear regression model. With the described risk factors the model has the following expression:

$$r_{P,t} - r_{B,t} = \alpha + \beta_1 M_{USD,t}^{3M} + \cdots + \beta_6 M_{USD,t}^{30Y} + \cdots + \beta_{25} M_{AUD,t}^{3M} + \cdots + \beta_{30} M_{AUD,t}^{30Y} + \gamma_1 D_{USD,t} + \cdots + \gamma_6 D_{AUD,t} + \delta_1 SS_{USD,t} + \cdots + \delta_6 \delta_1 SS_{AUD,t} + \cdots + \theta_1 FX_{\frac{EUR}{USD},t} + \theta_2 FX_{\frac{GBP}{USD},t} + \ldots + \theta_5 FX_{\frac{AUD}{USD},t} + \varepsilon_t \quad [31]$$

$r_{P,t}$:	Portfolio returns in period t
$r_{B,t}$:	Benchmark returns in period t
α:	Constant variable
$\beta_{1,\ldots,30}$:	Sensitivities of the active returns to market risk in USD, EUR, GBP, JPY, CAD and AUD for the selected term structure
$\gamma_{1,\ldots,6}$:	Sensitivities of the active returns to duration risk in USD, EUR, GBP, JPY, CAD and AUD
$\delta_{1,\ldots,6}$:	Sensitivities of the active returns to swap spread risk in USD, EUR, GBP, JPY, CAD and AUD
$\theta_{1,\ldots,5}$:	Sensitivities of the active return to foreign exchange risk in EUR, GBP, JPY, CAD and AUD
$M_{currency,t}^{maturity}$:	Market risk for the respective currencies and maturities in period t
$D_{currency,t}$:	Duration risk for the respective currencies and maturities in period t
$SS_{currency,t}$:	Swap spread risk for the respective currencies and maturities in period t
$FX_{\frac{\text{currency}}{\text{USD}},\text{t}}$:	Foreign exchange risk being the USD the domestic currency in period t
ε_{t}:	Error term in period t

There are some main assumptions that underlie this type of models. First of all, that there is a linear relationship between the dependent and the explanatory variables. The error term included in the model should, on one hand, have a homoscedastic variance and on the other hand, have no serial correlation. Finally, the distribution of the error term should be normal. Let's remark that if we wanted to forecast the ex-ante tracking error for only one specific market, the factors duration and swap spread could also be divided into different maturities to obtain more precise results and capture the risk better. If we would do that for an international portfolio we would obtain even more factors to consider and thus, estimating the sensitivities would be a too complicated task.

We will proceed by estimating the multiple regression model using the well known Ordinary Least Squares method. Successively, we will contrast whether the independent variables selected are of significance for the explanation of the dependent variable and will check whether the assumptions are actually met.

Because of the relatively high amount of independent variables and the common information that these variables most surely carry, many coefficients show to be statistically insignificant. However, the fact that their parameters show statistical insignificance does not mean that the variable itself does not carry relevant information for the model. A much more reliable way in this situation than using indicators such as the t-statistic to contrast whether a variable provides relevant information and should hence be kept in the model or not, is to check whether withdrawing the variables increments the variance of the residuals. This is done for each of the variables and we come to the conclusion that we can remove the 3 month LIBOR rates series for all the markets from the model. The rest of the proposed risk factors are kept in the model. We need to mention that in a few cases, the variance of the residuals did not increment, or at least not substantially for the factors we leave. Nonetheless, as it did not happen for a type of variable in all the markets (by type we mean for example the 5 year Libor rate independently of which country), but most of the cases in only one market, because we need to have the same risk factors in each market to be able to make consistent bets, the variable was kept.

Keeping only the factors that changed the variance of the residuals in a significant way, we estimate via OLS the definitive multiple regression model, which is:

$$r_{P,t} - r_{B,t} = \alpha + \beta_1 M_{USD,t}^{2Y} + \cdots + \beta_4 M_{USD,t}^{30Y} + \cdots + \beta_{21} M_{AUD,t}^{2Y} + \cdots + \beta_{24} M_{AUD,t}^{30Y} + \gamma_1 D_{USD,t} + \cdots + \gamma_6 D_{AUD,t} + \delta_1 SS_{USD,t} + \cdots + \delta_6 \delta_1 SS_{AUD,t} + \cdots + \theta_1 FX_{\frac{EUR}{USD},t} + \theta_2 FX_{\frac{GBP}{USD},t} + \ldots + \theta_5 FX_{\frac{AUD}{USD},t} + \varepsilon_t \qquad [32]$$

$r_{P,t}$:	Portfolio returns in period t
$r_{B,t}$:	Benchmark returns in period t
α:	Constant variable
$\beta_{1,\ldots,24}$:	Sensitivities of the active returns to market risk in USD, EUR, GBP, JPY, CAD and AUD for the selected term structure
$\gamma_{1,\ldots,6}$:	Sensitivities of the active returns to duration risk in USD, EUR, GBP, JPY, CAD and AUD
$\delta_{1,\ldots,6}$:	Sensitivities of the active returns to swap spread risk in USD, EUR, GBP, JPY, CAD and AUD

$\theta_{1,\ldots,5}$: Sensitivities of the active return to foreign exchange risk in EUR, GBP, JPY, CAD and AUD

$M_{currency,t}^{maturity}$: Market risk for the respective currencies and maturities in period t

$D_{currency,t}$: Duration risk for the respective currencies and maturities in period t

$SS_{currency,t}$: Swap spread risk for the respective currencies and maturities in period t

$FX_{\frac{\text{currency}}{\text{USD}},t}$: Foreign exchange risk being the USD the domestic currency in period t

ε_t: Error term in period t

Checking now for the assumptions, the results show uncorrelated, heteroskedastic and not normally distributed residuals. In order to provide an overview of the estimation outputs, in the next table we show some statistics and indices for each of the treated portfolios and estimation periods:

Table 1: Estimation Output

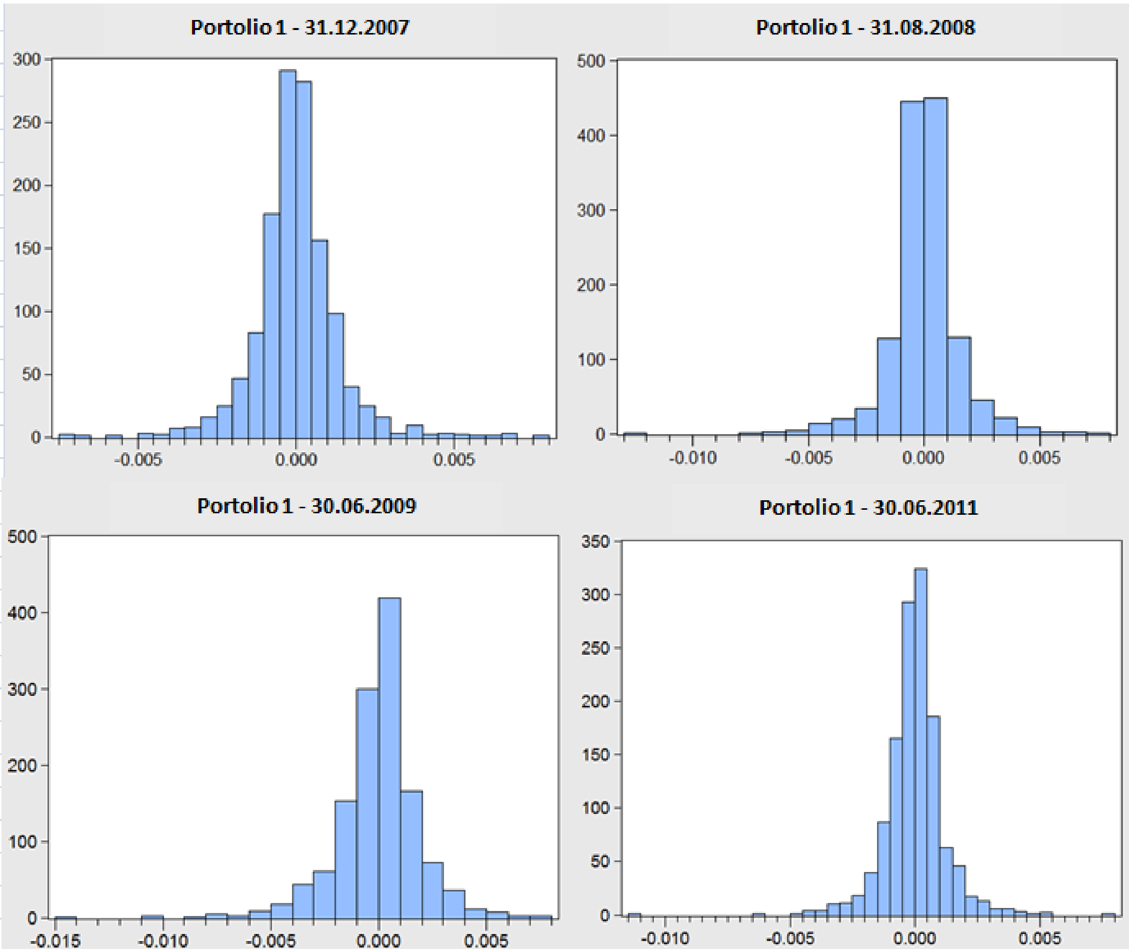
	R-squared	F-statistic	Durbin-Watson stat	Breusch-Pagan-Godfrey Test	Harvey Test	Glesjer Test	Jarque Bera
Portf 1 - 31.12.2007	0.73	83.22139 (0)	2.146772	2.470210 (0)	1.283031 (0.1106)	2.078763 (0.0001)	1967.075 (0)
Portf 1 - 31.08.2008	0.77535	106.3188 (0)	2.122813	2.580382 (0)	1.559674 (0.0141)	1.944176 (0.0004)	3121.090 (0)
Portf 1 - 30.06.2009	0.744363	89.62634 (0)	2.045261	1.323168 (0.0849)	1.415969 (0.0440)	1.721120 (0.0034)	1935.104 (0)
Portf 1 - 30.06.2011	0.812012	132.8508 (0)	2.216594	1.307267 (0.0944)	2.146788 (0)	0.962899 (0.5386)	6006.789 (0)
Portf 2 - 31.12.2007	0.606685	47.47854 (0)	2.105671	1.640635 (0.0070)	1.436299 (0.0378)	1.965361 (0.0003)	2422.768 (0)
Portf 2 - 31.08.2008	0.725457	81.39945 (0)	2.144948	2.472118 (0)	1.394768 (0.0514)	1.961181 (0.0003)	2829.228 (0)
Portf 2 - 30.06.2009	0.743005	88.99018 (0)	2.103105	1.657512 (0.0061)	0.915611 (0.6243)	1.531475 (0.0179)	1297.245 (0)
Portf 2 - 30.06.2011	0.756501	95.55303 (0)	2.273803	1.502456 (0.0226)	0.744865 (0.8815)	1.025386 (0.4277)	3937.125 (0)

* Recall that for each time period we work with daily data excluding weekends for a data range of 5 years.

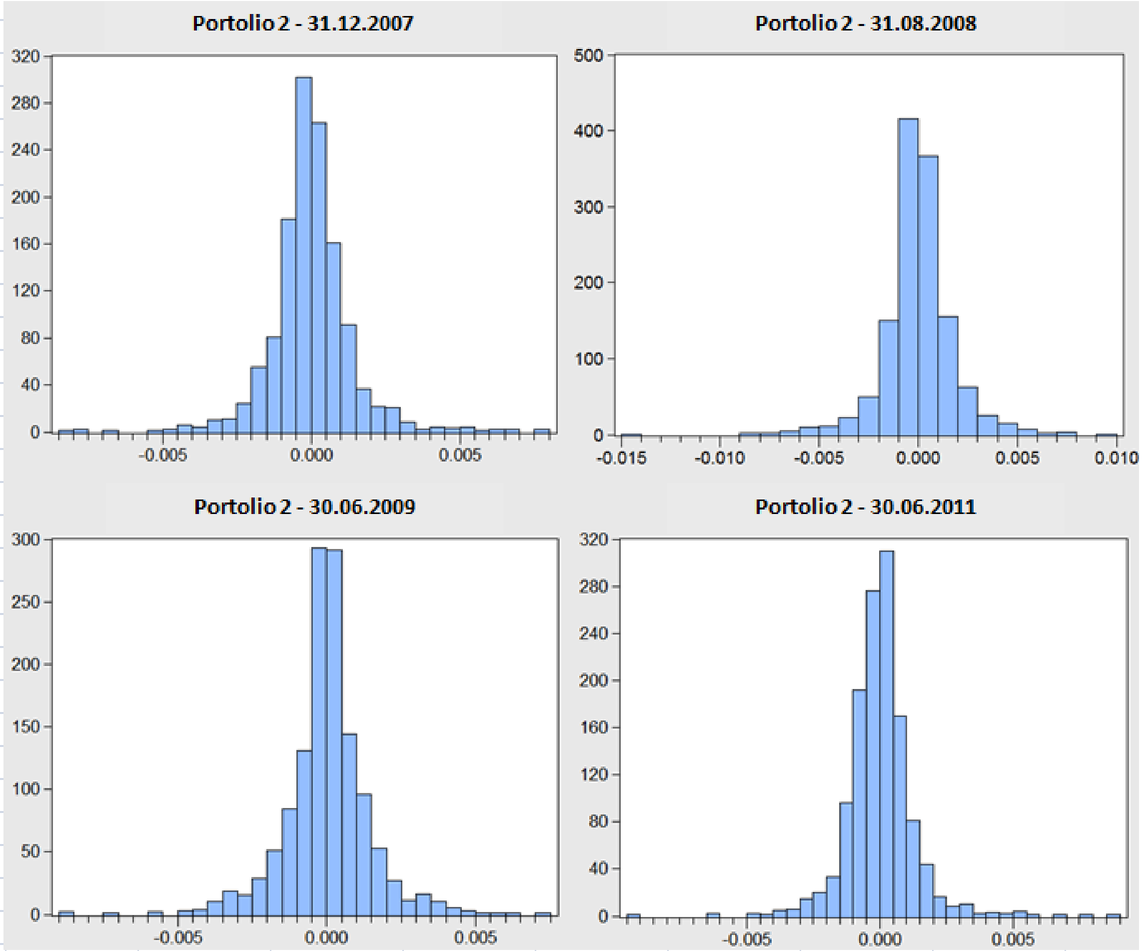
The R^2 represents the fraction of the dependent variable's variance that is explained by the model, it is thus a measure of fit. As we can see in the table, the R^2 values range from over 0.6 to around 0.81, meaning that we have generally obtained a relatively good fit. Further, as we can read from the table, we have F-statistics that clearly reject the null hypothesis which states that the independent variables don't jointly explain the dependent variable. The model has therefore explanatory power. The next statistic is the Durbin-Watson diagnostic statistic which helps to determine whether the residuals are correlated or not. As the values are all above 2 or close to it, we can say that the residuals are independently distributed, which is, together with the linear relationship between the variables, the most important assumption of the linear regression model. The next three tests in the table, the

Brausch-Pagan-Godfrey Test, the Harvey Test and the Glesjer Test, contrast the null hypothesis that the variance of the residuals is homoscedatic. As we can see, the general result for all portfolio and points in time in which they are analyzed, the different heteroskedasticity tests agree on rejecting homoscedatic residuals. Finally, the table shows the results for the Jarque-Bera test which contrasts the normal distribution of residuals. We can clearly see that normality is rejected in all portfolios as was the case when doing the simple linear regressions. The reason is, as already mentioned, the excess of kurtosis, meaning there are several extreme events present in the data. In order to show that we plot the histograms of the residuals, where the outliers are clearly visible:

Graph 3: Histogram of residuals – portfolio 1

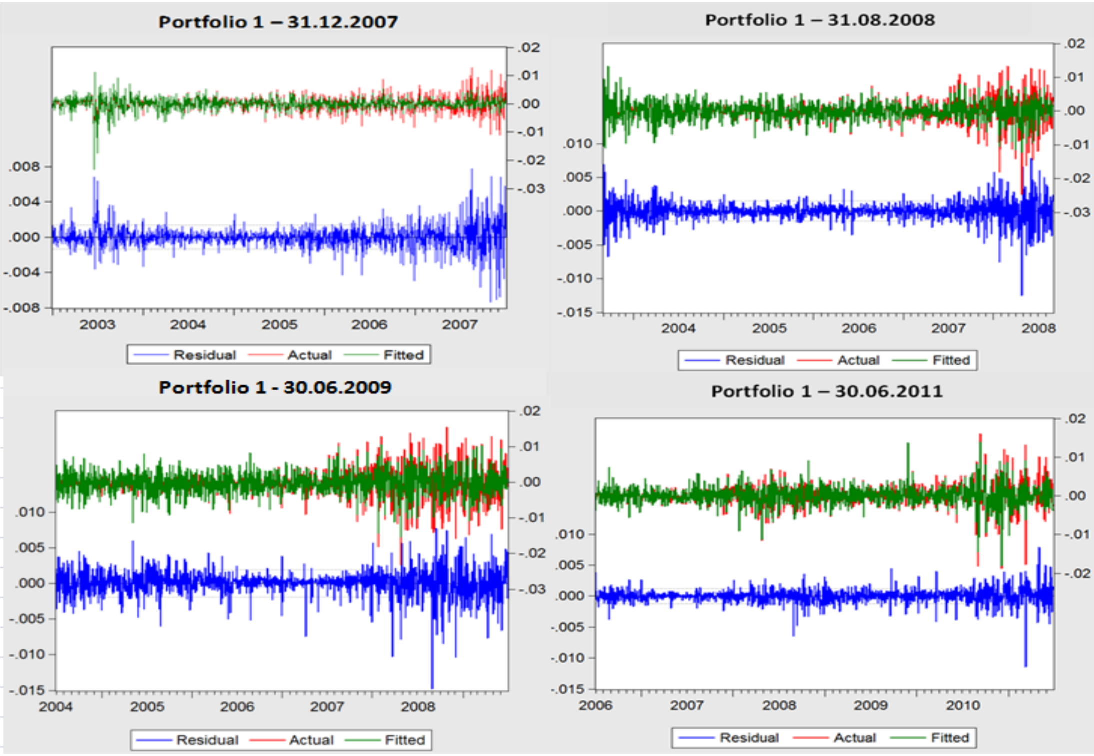


Graph 4: Histogram of residuals – portfolio 2

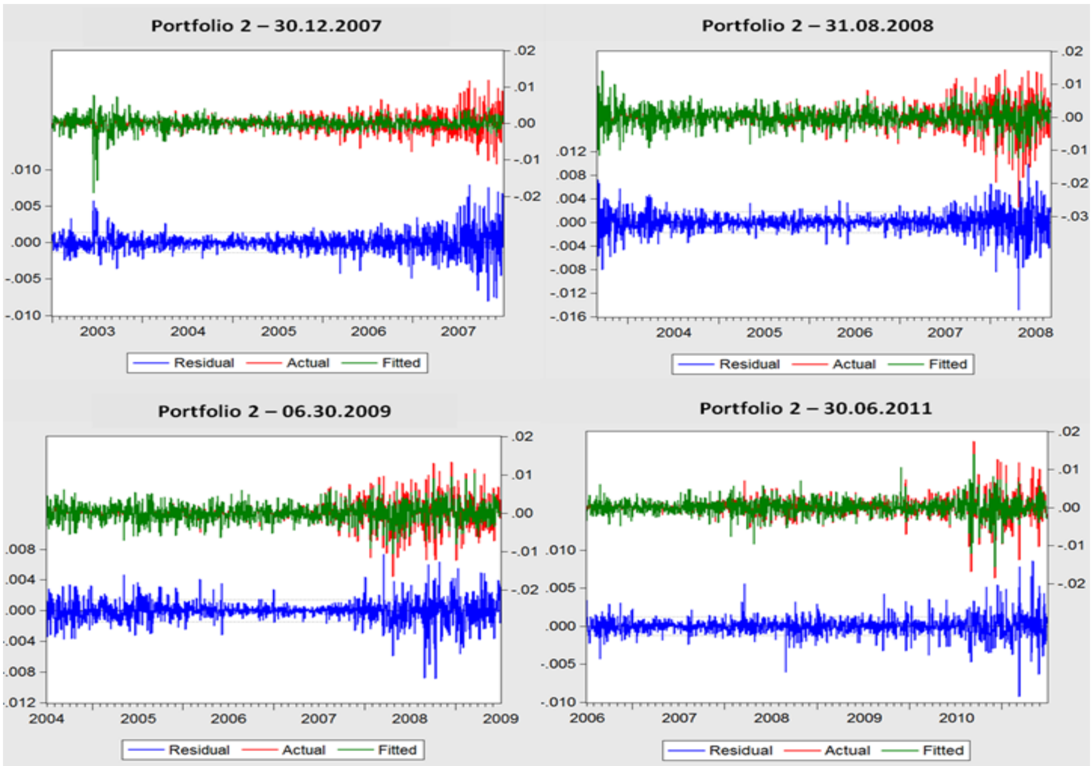


In the next charts we show the actual and fitted values of the dependent variable, along with the residuals, in order to get an additional feel for the fit of the model.

Graph 5: Dependent variable and residuals – portfolio 1



Graph 6: Dependent variable and residuals – portfolio 2



For both portfolios we can observe that the fitted graph adapts acceptable when there aren't outliers in the actual graph. Consequently we have higher absolute residuals when extreme events take place.

As we could see along the analysis the model does present some good features. However, the fact that we have problems such as heteroskedasticity and non normality of residuals, to get reliable and not distorted results, we would require a further development of the model. Nonetheless, as this is not the main focus of this research and the aim of estimating the coefficients of the independent variable is simply to capture the fact that a change in a concrete risk factor will not have the same strength of impact in the active return, we will not continue to develop this model further, but we will use the weighted sensitivities as described in section 2.1.1.. To see the estimated and transformed sensitivity values please check the appendix.

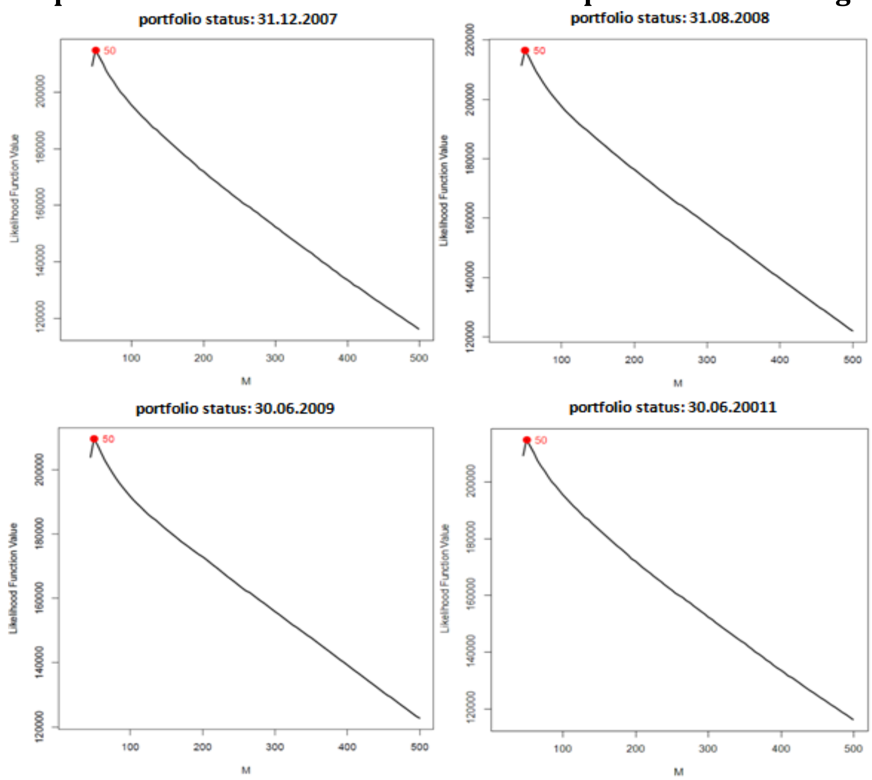
5. Results

In this part we will show on one hand the results obtained for the different parameters that are needed in the different volatility forecasting models, on the other hand, we will explain the performed backtest and show its results.

5.1 Parameters of the volatility forecasting models

In the historical moving average model, before being able to forecast the variance and covariance matrix, we needed first to estimate the optimal window length. Using the quasi-maximum likelihood model for its estimation, we have retrieved the optimal sample length for each of the four estimation periods. Because the extraction of the optimal sample length is computationally intense, we have used a grid of possible M values. We have imposed the window length to start with 20 and have a maximum of 500 observations. The value for M increases on a 5 observations step. With a more powerful data processor, the extraction of M could be performed more precisely. In the following charts we plot the maximum likelihood function values for different sizes of the window length M. The one delivering the highest value for the likelihood function is the optimal window and hence the one used to perform the volatility forecasts. The M values and the likelihood functions for each period are depicted in the following charts:

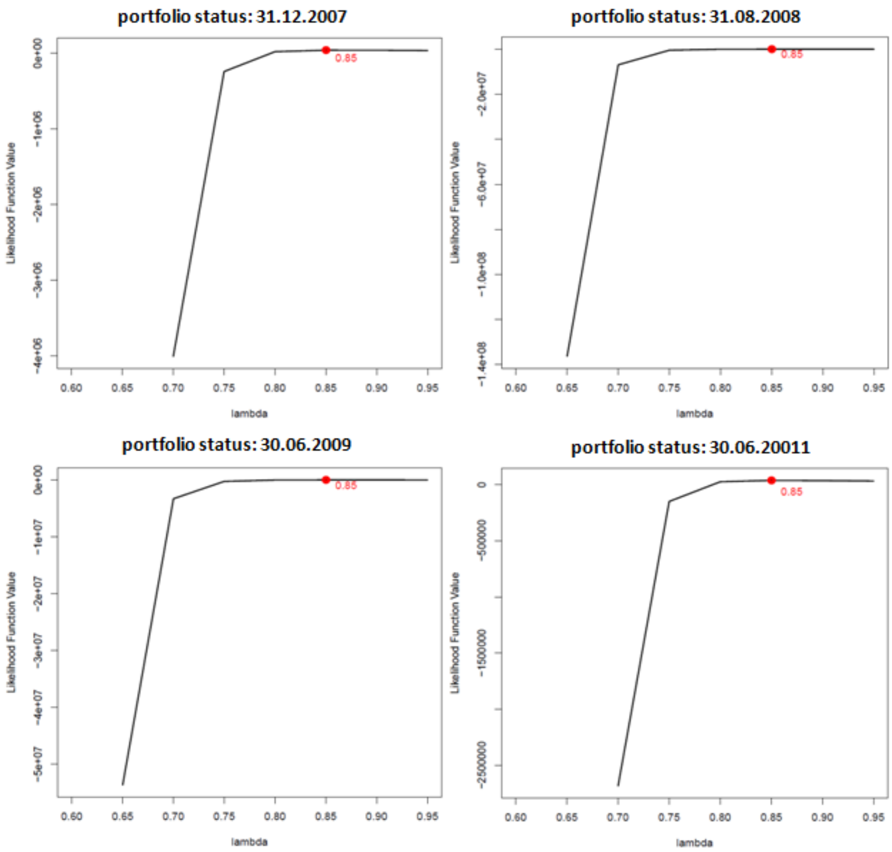
Graph 7: Maximum likelihood functions - Optimal window length



As for both portfolios we are analyzing the same risk factors, the same optimal window lengths are valid for both of them. As we can see in the graphs, the output seems to be consistent. For all periods in time subject to analysis the likelihood function reaches the optimum value at 50 observations.

The next model we defined for volatility forecasting is the EWMA model. This method is supposed to improve the historical moving average model by giving more weight to more recent data by weighting the observation with the power of a constant parameter lambda, between 0 and 1. To obtain an estimation of this parameter we used again the quasi-maximum likelihood model. Because the optimization process is again computationally demanding, a grid was again specified. The parameter is defined to move between 0.6 and 0.95 with steps equal to 0.05. For all periods of analysis the value of the constant parameter that maximizes the likelihood function is equal to 0.85. This can be seen in the following charts:

Graph 8: Maximum likelihood functions - Optimal lambda



For values of the parameter lambda lower than 0.65 the variance and covariance matrix is close to being singular, leading the likelihood function to tend towards infinity.

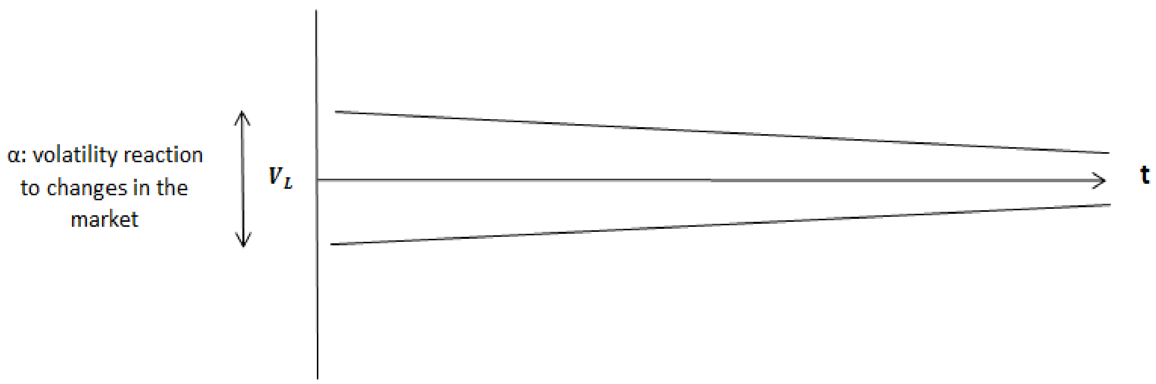
Finally, the parameters we needed to optimize are those contained in the last volatility forecasting model subject to study, the DCC-GARCH(1,1) model. First we had the parameter gamma accounting for the rate of reversion to the long run variance. Next we needed to estimate the parameter alpha that considers the "reaction" of the volatility to market changes. Finally, the estimation of the parameter beta was required, which reflects the persistence of volatility. The quasi-maximum likelihood model has been used for the estimation of the parameters, but this time we have used an optimizer to obtain the estimations included in the R package called fGarch. The obtained results for each data period are shown in the next table 2:

Table 2: Optimal parameter values

γ	α	β
0.000201849	0.088933132	0.910865019
0.000242633	0.086442137	0.91331523
4.2842E-05	0.099269682	0.900687476
4.2842E-05	0.099269682	0.900687476

From the table we can clearly see that for all periods of analysis we have a high persistence in volatility. When a shock in volatility occurs, it will therefore take a long time until the volatility reverts to its mean level. This is consistent with the resulting gamma value that clearly shows that the rate of reversion to the mean is very low. The alpha value also shows that the reaction of volatility to changes in the market won't be pronounced. To better understand these results, we represent these results graphically:

Graph 9: Volatility effects in the DCC-GARCH(1,1) model



5.2 Backtesting – Mean Squared Error

To perform the backtesting we used one of the most important criterions to evaluate the performance of predictors, the Mean Squared Error (MSE)[11]. It has the advantages of being adequate for comparative purposes and is analytically tractable. In fact it is nothing else than a statistical loss function. Adapted to our case the Mean Squared Error is expressed as follows:

$$MSE = \frac{\sum_{s=1}^{S} \sum (EA\ TE_{l,s} - EP\ TE_{h,s})}{S} \qquad [33]$$

where S stands for the number of simulated vectors of relative exposures, with whom we calculated S times the ex-ante and ex-post tracking errors. The subscript l stands for the different types of volatility methods and forecast horizons for which we computed the ex-ante tracking error. In total we have 9 different forward looking tracking errors as we use three different volatility estimation methods and the same number of forecast horizons. Finally, the subscript h accounts for the number of different windows used to compute the ex-post tracking errors. As we take four different windows, we computed for each vector of relative exposures, four different realized tracking errors.

In the tables shown below, we present the obtained Mean Squared Errors. The tables are grouped by period of analysis. To make the interpretation easier, for each of the forecasting periods and portfolios, the Mean Squared Errors with lower values, which represent better approximations to the ex-post values, are green coloured. As the values of the Mean Squared Error become worse they start turning into red.

The first table shows the results for the period of analysis just prior to the financial crisis, concretely for the portfolios as of the 31.12.2007. In general we can say that the ex-ante tracking error that approximates the realized tracking error is the one computed with the volatility forecasting model EWMA. This is valid for all forecasting periods and both portfolios. The volatility forecasting model DCC-GARCH(1,1) seems to deliver the next best results, however it differs only marginally from the results obtained by the historical moving average model. The ex-post tracking errors that are best approximated by all ex-ante tracking errors are the ones computed with a sample window of 250 observations.

[11] For more details see: Novales, A. (2000), *Econometría*, Segunda Edición, McGraw-Hill.

Table 3: Mean Squared Errors – 31.12.2007

	PORTFOLIO 1				PORTFOLIO 2			
	EPTE V30	EPTE V60	EPTE V120	EPTE 250	EPTE V30	EPTE V60	EPTE V120	EPTE 250
	FORECASTING PERIOD T+10				FORECASTING PERIOD T+10			
HIST	0.2512658	0.1168729	0.054515	0.0210867	0.3045006	0.1515858	0.0778413	0.0359666
EWMA	0.2338921	0.1051256	0.0465935	0.0162814	0.2183834	0.092933	0.0378402	0.0110696
DCC_GARCH	0.2571341	0.1208867	0.0572673	0.0228116	0.2742171	0.130459	0.0629331	0.0260941
	FORECASTING PERIOD T+20				FORECASTING PERIOD T+20			
HIST	0.5791249	0.2752388	0.1307949	0.0524258	0.530297	0.2391299	0.1051243	0.0363978
EWMA	0.0320411	0.0043127	0.0502946	0.1269726	0.2518956	0.0690545	0.0096866	0.0014411
DCC_GARCH	0.2575316	0.0736129	0.0118949	0.0008779	0.4605291	0.1931206	0.0754849	0.0200233
	FORECASTING PERIOD T+250				FORECASTING PERIOD T+250			
HIST	1.226764	2.1604608	1.56869	2.5315149	1.0179015	1.8122228	1.2892241	2.2580396
EWMA	0.0057906	0.1510853	0.0315529	0.257397	0.3454951	0.8555428	0.5101699	1.1694696
DCC_GARCH	0.405646	0.9970994	0.6104027	1.2530092	0.8402622	1.5722353	1.0881943	1.9891254

For the period of analysis as of the 31.08.2008, when the global financial crisis is at its worst moment with the threat of total collapse of large financial institutions, the bailout of banks by national governments, and downturns in stock markets around the world, we obtain for all volatility forecasting models the worst results, meaning the highest Mean Squared Error values. While for the periods 31.12.2007, 30.06.2009 and 30.06.2011 the highest Mean Squared Error values in the tables are 0.5791249, 0.5265762 and 1.8665694 respectively, in evaluation period 31.08.2008 the highest value goes up to 21.5811112. Keeping in mind that for turbulent periods none of the ex-ante tracking error estimations are satisfactory, for the forecasting horizons T+10 and T+20 the best results are delivered by the DCC-GARCH(1,1) model, while for the forecasting horizon T+250 the historical moving average model clearly delivers the lowest Mean Squared Error values. In general we can observe that the ex-post tracking error that is best approximated for all forecasting periods is in that case the one computed with the shortest window, meaning 30 observations.

Table 4: Mean Squared Errors – 31.08.2008

	PORTFOLIO 1				PORTFOLIO 2			
	EPTE V30	EPTE V60	EPTE V120	EPTE 250	EPTE V30	EPTE V60	EPTE V120	EPTE 250
	FORECASTING PERIOD T+10				FORECASTING PERIOD T+10			
HIST	0.1539987	0.2513504	0.2945105	0.36434672	0.1671352	0.1983723	0.2128198	0.2450337
EWMA	0.3749468	0.574875	0.548958	0.4386495	0.7442857	0.826503	0.8568712	0.9184724
DCC_GARCH	0.1037422	0.1859392	0.2233572	0.28473498	0.0872564	0.0598462	0.0965412	0.0432586
	FORECASTING PERIOD T+20				FORECASTING PERIOD T+20			
HIST	0.633641	0.9351446	1.0740085	1.25804083	0.8173471	0.9825884	1.0590019	1.1751572
EWMA	2.6584938	2.859646	2.7589357	2.9475975	7.8485393	8.5867076	8.8731879	9.2761705
DCC_GARCH	2.3598826	2.9148281	3.1562029	3.46651462	0.4291474	0.2317935	0.1731304	0.1087128
	FORECASTING PERIOD T+250				FORECASTING PERIOD T+250			
HIST	3.776868	3.3879543	3.4495046	1.94633944	8.3803739	6.1155885	6.3398696	6.2657708
EWMA	4.2748442	4.8648706	4.8438358	5.19463847	21.581112	20.475532	19.909805	18.345369
DCC_GARCH	11.026106	10.355708	10.464225	7.68534694	18.173753	14.771803	15.600233	16.314484

One year after the big turmoil we can see the Mean Squared Error values shrink back to even lower levels than for the period of analysis 31.12.2007. This is plausible as at the end of 2007 the financial crisis had already started to show its first symptoms. For the two shorter forecast horizons T+10 and T+20 the ex-ante tracking error that best approaches the backward looking tracking error is the one computed with the EWMA model. For the forecasting horizon T+250 the best results are delivered by the historical moving average model. Regarding which ex-post tracking error is better approximated depending on the sample length with which it was computed, we cannot see any clear pattern.

Table 5: Mean Squared Errors – 30.06.2009

	PORTFOLIO 1				PORTFOLIO 2			
	EPTE V30	EPTE V60	EPTE V120	EPTE 250	EPTE V30	EPTE V60	EPTE V120	EPTE 250
	FORECASTING PERIOD T+10				FORECASTING PERIOD T+10			
HIST	0.084316	0.0599246	0.16903	0.1859254	0.0167208	0.0398957	0.0506092	0.05048334
EWMA	0.135519	0.1132537	0.2598492	0.280459	0.0267301	0.0085328	0.0045854	0.00467592
DCC_GARCH	0.1808683	0.1601244	0.3297988	0.3526957	0.0384274	0.0155949	0.0100271	0.01014962
	FORECASTING PERIOD T+20				FORECASTING PERIOD T+20			
HIST	0.004578	0.0911901	0.2756412	0.3205081	0.0110105	0.0388845	0.0563998	0.05261557
EWMA	0.0411296	0.0657579	0.1837007	0.2190483	0.0122785	0.0004634	0.0012999	0.001068
DCC_GARCH	0.0207663	0.1915033	0.468494	0.5265762	0.0128283	0.0015681	0.0026602	0.00242238
	FORECASTING PERIOD T+250				FORECASTING PERIOD T+250			
HIST	0.0201389	0.0178116	0.0096602	0.0158341	0.0118346	0.0088258	0.0083972	0.02836103
EWMA	0.1396212	0.1411818	0.1207742	0.0255911	0.189525	0.1241912	0.1282833	0.27860098
DCC_GARCH	0.0837913	0.0767854	0.0867893	0.2341796	0.1893091	0.1245807	0.1286944	0.27694202

The last table showing the results for the evaluation period as per 30.06.2011 shows the generally lowest Mean Squared Error values. For the forecasting horizons T+10 and T+20, for portfolio 1 the best approximations to the ex-post tracking errors are those delivered by the historical moving average model. In contrast, for portfolio 2 the best approximations are attained via the ex-ante tracking error calculated with the EWMA model. For the 1 year forecast horizon however, the historical moving average model is the one delivering the most accurate results for both portfolios. Like for the estimation period 31.12.2007, in that case the ex-ante tracking error that is best approximated is the one computed with a window length of 250 observations.

Table 6: Mean Squared Errors – 30.06.2011

	PORTFOLIO 1				PORTFOLIO 2			
	EPTE V30	EPTE V60	EPTE V120	EPTE 250	EPTE V30	EPTE V60	EPTE V120	EPTE 250
	FORECASTING PERIOD T+10				FORECASTING PERIOD T+10			
HIST	0.0053757	0.0040183	0.0032461	0.0014238	0.0067746	0.0038239	0.0038625	0.0015704
EWMA	0.1602434	0.1521043	0.146826	0.128988	0.0023177	0.0008075	0.0008701	0.0002741
DCC_GARCH	0.0230048	0.0200035	0.0181187	0.0123126	0.0437716	0.0356259	0.035581	0.0272167
	FORECASTING PERIOD T+20				FORECASTING PERIOD T+20			
HIST	0.0650839	0.0570606	0.0543636	0.0392749	0.0324266	0.0232331	0.0232747	0.0140516
EWMA	0.6011712	0.5763181	0.5677571	0.5146453	0.0080182	0.0037789	0.0039075	0.0010908
DCC_GARCH	0.5414785	0.5180363	0.5098726	0.4593505	0.2557276	0.2287564	0.2285108	0.1961669
	FORECASTING PERIOD T+250				FORECASTING PERIOD T+250			
HIST	0.006628	0.0117276	0.0380225	0.0429306	0.0403268	0.0140534	1.5674599	0.6531313
EWMA	0.9636416	1.1129505	1.3327152	1.3515879	0.1227406	0.0723101	1.8665694	0.8418223
DCC_GARCH	0.8246446	0.9627265	1.1680018	1.1857655	0.2154162	0.2758274	0.9649613	0.446593

From these observations we can make some general deductions. For periods with high instability in the markets, the results for the forecast of the tracking error are generally poor. However, if a volatility model needs to be selected the best choice seems to be the historical moving average model, at least when estimating for a 1-year horizon. For the forecasting horizons T+10 and T+20 the DCC-GARCH(1,1) model could also be considered. Under negative economic environments the ex-post tracking error that is best approximated is one computed with a short window length.

The more stable the economic environment, the more accurate are the ex-ante tracking errors, independently of the volatility estimation model that has been used. The volatility model delivering the best results under these economic circumstances seems to be for the forecasting period T+250 the historical moving average model. For the two shorter forecasting periods it is unclear whether to favour the historical moving average or the EWMA model. It would be interesting to see whether the EWMA would definitely outperform the historical moving average model in an environment of more economic stability than being the case for the year 2011, where the financial crisis is still patent. Finally, the last pattern that can be read out is that for more stable markets the ex-post tracking error that is best approximated is the one computed with a longer sample length which in our case is 250 observations.

6. Conclusions

Triggered by the recent financial crisis, growing concerns about forecasted risk have come up. One of those concerns regards the existing deviation between the ex-post or realised tracking error, and the forecasted or ex-ante racking error. The tracking error has been defined as the volatility of active returns, which is the difference between portfolio and benchmark returns.

In this research we have proposed a model for the estimation of the ex-ante tracking error specifically for actively managed global fixed income portfolios. Instead of making the active returns depend on securities, we have chosen variables standing for the risks on which active fixed income portfolio managers take bets to try to outperform the benchmark.

The main focus has been put on testing the model with different volatility forecasting methodologies to figure out which one delivers the best approximation of the estimated ex-ante tracking error to the realized tracking error under different economic environments. For this purpose two international fixed income portfolios, benchmarking both against the JPMorgan Global Traded Index, have been selected for analysis, at four different points in time around the financial crisis.

The volatility forecasting methodologies that have been used for the analysis are the historical moving average model, the Exponentially Weighted Moving Average (EWMA) model and the Dynamic Conditional Correlation (DCC-GARCH(1,1)) model. Each of those models differs in the way the different observations are weighted. The last model also tries to capture characteristics of return volatilities such as volatility clustering and mean reversion. The different parameters contained in the mentioned models have been estimated using the quasi-maximum likelihood model.

As a new approach, the model that is proposed for the forecast of the tracking error tries to account for the fact that the active returns are not influenced in the same proportion by all the risk factors that have been selected. To include this effect, the sensitivities of the active returns towards the risk factors have been estimated applying Ordinary Least Squares to a multiple linear regression model. The sensitivities are however not included in the model as is, but are transformed in order to be centred around one.

To perform backtesting and be able to make a comparison between the results obtained with the different volatility forecasting models, for each portfolio and period of analysis 100'000 bet strategies have been simulated. Each single strategy has been applied to the portfolios afterwards, to obtain the according ex-post tracking errors.

The comparison of the obtained results has been done using the Mean Squared Error, which is suitable for the assessment of the performance of predictors. The values of the Mean Squared Errors showed that for unstable economic environment the model performs badly for all volatility forecasting models. The best result is however delivered by the historical moving average model, at least for the 1-year horizon. For more stable economic environments for the forecasting periods T+10 and T+20, it is unclear whether to favour the historical moving average or the EWMA model. For T+250 the historical moving average model seems again to be the better choice.

To improve the forecast of the tracking error the investigation could be developed further on different grounds. The model to estimate the sensitivities of the active returns towards the risk factors could for example be strongly worked out. The analysis could also be centred on portfolios trading in one single market eliminating in that way the additional foreign exchange risks and in exchange, more factors could be introduced to replicate the unavoidable risks. It would also be interesting to incorporate periods with more economic stability to gain a better insight on whether the EWMA does outperform the historical moving average under such economic conditions. Of course, different volatility forecasting models, such as for example stochastic volatility models could be tested out as well.

7. References

- Focardi, Sergio M. And Fabozzi, Frank J. (2004), *The Mathematics of Financial Modeling & Investment Management*, John Wiley & Sons, Inc.
- MSCI Barra (2007), *Barra Risk Model Handbook*, MSCI Barra.
- Mina, Jorge and Xiao, Jerry Yi (2001), *Return to RiskMetrics: The Evolution of a* Standard, RiskMetrics Group, Inc.
- Gabudean, Radu and Schuehle, Niels (2011), Volatility Forecasting: A Unified Approach to Building, Estimating and Testing Models, Barclays Capital.
- Jessop, David (2012), *Forecasting Volatility and Tracking Errors*, UBS Investment Research.
- Novales, A. (2000), *Econometría*, Segunda Edición, McGraw-Hill.
- Amman, Manuel and Tobler, Jürg (2000), Measurement and Decomposition of Tracking Error Variance, Swiss Institute of Banking and Finance, University of St. Gallen.
- Poon, Ser-Huang and Granger, Clive W.J. (2003), Forecasting Volatility in Financial Markets: A Review, Journal of Economic Literature, Vol. XLI, pp. 478-539.
- Pope, Peter F. and Yadav, Pradeep K. (1994), Discovering Errors in Tracking Error, The Journal of Portfolio Management.
- Hull, John, Options Futures and Other Derivatives (2009), Seventh Edition, Pearson Prentice Hall.
- Normand, John (2007), Introduction to Portfolio Management, JPMorgan.
- Minkah, Richard (2007), Forecasting Volatility, Uppsala University.
- Engle, Robert (2000), Dynamic Conditional Correlation – A Simple Class of Multivariate GARCH Models, research supported by NSF grant SBR-9730062 and NBER.
- Engle, Robert and Sheppard, Kevin (2001), Theoretical and Empirical Properties of Dynamic Conditional Correlation Multivariate GARCH, National Bureau of Economic Research.

8. Appendix

Table 7: Sensitivities of the active returns towards the risk factors – Portfolio 1

Dez 07		Aug 08		Jun 09		Jun 11	
Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients
0.013625	3.376879794	-0.006229	-1.688286791	0.004783	1.675937883	-0.002544	-1.392182163
-0.01562	-3.871329349	0.025897	7.019034039	-0.00233	-0.816419667	0.001033	0.565300383
0.0094	2.329737252	6.24E-05	0.016912682	0.015528	5.440929007	-0.001096	-0.599776592
-0.028394	-7.037293568	-0.019962	-5.410431999	0.008068	2.826984495	-0.009399	-5.14352207
-0.004813	-1.192875042	-0.010021	-2.716057462	-0.010687	-3.744668232	-0.002321	-1.270147327
0.017313	4.290929899	-0.01827	-4.951838124	0.011333	3.971023212	-0.004579	-2.505818444
-0.019225	-4.76480837	-0.021654	-5.869025875	0.032561	11.40920205	0.015452	8.455974361
-0.011659	-2.889617726	0.001766	0.478650582	0.00116	0.40645786	0.012344	6.75514804
-0.00663	-1.643208296	0.007576	2.05337305	0.003328	1.166113584	-0.001779	-0.973542479
0.004499	1.115051904	-0.020168	-5.466265533	0.00781	2.73658266	-0.001745	-0.954936271
0.029316	7.265806095	0.030887	8.37150652	0.004544	1.592193548	0.014564	7.970023984
-0.033513	-8.306008993	-0.037775	-10.23840641	-0.002113	-0.740384016	0.004249	2.325228777
-0.000699	-0.173243228	0.006344	1.719455997	0.012553	4.398504755	-0.008559	-4.68383928
-0.01708	-4.233182156	-0.021055	-5.70667497	-0.069219	-24.25397121	-0.071965	-39.38222851
-0.059602	-14.77202125	-0.144965	-39.29081629	-0.087606	-30.69667869	0.024389	13.3466709
0.013451	3.33375487	0.078422	21.25522985	0.01624	5.690410039	-0.026406	-14.45045683
-0.003055	-0.757164607	0.003244	0.879242631	-0.007825	-2.741838581	-0.00063	-0.344762092
-0.003759	-0.931647057	-0.018525	-5.020952449	0.003877	1.358480278	0.011098	6.073285236
-0.024897	-6.170581742	-0.023408	-6.344424018	-0.010063	-3.526021935	-0.016561	-9.062865092
0.036359	9.011374123	0.05774	15.64965152	0.016963	5.943745412	0.006379	3.490852993
-0.012011	-2.976858951	-0.061713	-16.72647981	-0.003644	-1.276838312	-0.001353	-0.740417636
0.032889	8.151354095	0.049577	13.43718	-0.003987	-1.397023696	-0.001819	-0.995432136
-0.063917	-15.84146978	0.022346	6.056583181	-0.024278	-8.506882692	-0.002222	-1.215970427
0.009873	2.446967648	-0.038785	-10.51215335	0.006534	2.289479014	-0.00361	-1.975541512
-0.081295	-20.14850957	-0.082926	-22.47597856	-0.128002	-44.85122326	-0.053625	-29.34582094
-0.012878	-3.191740036	-0.026641	-7.220685247	-0.034379	-12.04621963	-0.005455	-2.985201925
-0.019536	-4.841887975	-0.011262	-3.052413845	-0.00591	-2.070832717	-6.56E-05	-0.035899037
0.079092	19.60250838	0.085608	23.20289864	0.085373	29.91424731	0.047607	26.0525221
0.001814	0.449589721	0.015578	4.222207679	0.00246	0.861970979	-5.45E-03	-2.9835602
0.006451	1.59884415	0.016228	4.398381449	0.023298	8.163495879	0.011572	6.332677667
0.001547	0.383415269	0.001946	0.527437164	0.001788	0.626505736	-0.000121	-0.066216211
-0.001506	-0.373253649	0.0043	1.165457249	0.000956	0.33497734	-6.41E-05	-0.035078175
-1.46E-06	-0.000361853	-0.000555	-0.150425296	0.001139	0.399099571	-3.20E-05	-0.017511725
-5.92E-06	-0.001467239	-2.55E-05	-0.006911433	-1.99E-05	-0.006972855	2.47E-06	0.001351686
0.000627	0.155398432	-0.001784	-0.48352924	-0.003939	-1.38020475	0.003355	1.835994951
-0.001142	-0.283038292	-0.001594	-0.432032292	0.000213	0.074634073	-0.001035	-0.566394866
0.000944	0.233965103	-0.007136	-1.93411696	-0.018388	-6.443057869	-0.015164	-8.298368834
-0.016366	-4.056221263	0.001557	0.422003939	0.021465	7.521222382	-0.007614	-4.166696142
0.012428	3.080210061	-0.006659	-1.804832516	-0.021251	-7.446237915	0.01749	9.571252367
-0.014325	-3.550370866	0.015108	4.094820491	0.006273	2.198025996	0.016384	8.966003361
0.016875	4.182374057	0.005655	1.5327118	0.028383	9.945252964	-0.015624	-8.550099885

Table 8: Sensitivities of the active returns towards the risk factors - Portfolio 2

Dez 07		Aug 08		Jun 09		Jun 11	
Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients	Estimated coefficients	Transformed coefficients
0.015745	4.195438021	-0.012039	-3.105060202	-0.007873	-2.582746909	0.0000214	0.011834575
-0.013341	-3.554864315	0.033356	8.603072355	0.002867	0.940522722	-0.002141	-1.18401051
-0.00003	-0.007993848	3.50E-04	0.090270876	-0.008087	-2.65294986	-0.009913	-5.482062674
-0.021776	-5.80246798	-0.023453	-6.048922411	-0.021158	-6.940906781	0.006625	3.663741069
-0.000941	-0.250740373	-0.00782	-2.016909276	-0.005461	-1.791487472	-0.002311	-1.278023488
0.005362	1.428767143	-0.017158	-4.425336235	-0.00394	-1.292521633	-0.006045	-3.342990908
-0.002908	-0.774870357	-0.04169	-10.75255086	0.010386	3.407139514	0.02772	15.32964565
-0.026206	-6.982892905	0.005372	1.385528981	-0.010526	-3.453066678	-0.00237	-1.310651522
-0.007599	-2.024841761	0.00775	1.998855101	-0.001077	-0.353311117	-0.002193	-1.212767421
-0.00259	-0.690135565	-0.025588	-6.599574752	0.015812	5.187145195	-0.003839	-2.123034259
0.026416	7.038849843	0.038784	10.00304468	-0.01558	-5.111037322	0.012171	6.730776234
-0.032725	-8.719956129	-0.035214	-9.082281746	0.008438	2.768095823	0.00357	1.974272546
0.001684	0.448721348	0.007814	2.015361775	0.011236	3.685983013	0.000342	0.189131992
-0.000311	-0.08286956	-0.011284	-2.910333027	-0.04797	-15.73661491	-0.066506	-36.77898318
-0.06486	-17.28269991	-0.175927	-45.37452663	-0.068431	-22.44887002	0.021162	11.70295676
0.007611	2.0280393	0.094963	24.49255187	0.00523	1.71570765	-0.04135	-22.86727445
-0.008875	-2.364846773	0.001933	0.49855315	-0.007633	-2.504014626	0.002571	1.421808044
0.003969	1.057586123	-0.01587	-4.093139413	0.006164	2.022107449	0.00531	2.93652303
-0.011046	-2.943334924	-0.019628	-5.0623907	-0.005068	-1.66256336	-0.01009	-5.579946775
0.025127	6.69538083	0.062276	16.06202584	0.01136	3.726661359	0.014719	8.139864875
-0.013422	-3.576447705	-0.076483	-19.72624964	-0.016847	-5.526678162	-0.002823	-1.561168459
0.035631	9.494293562	0.054505	14.05775449	0.012352	4.052088126	0.014189	7.846765589
-0.061667	-16.43188799	0.040203	10.3690286	-0.008666	-2.842891491	-0.010441	-5.774055925
0.005886	1.568393026	-0.048541	-12.51953877	-0.003248	-1.06551022	0.002032	1.1237316
-0.074936	-19.96756707	-0.100427	-25.90180919	-0.035585	-11.6737011	-0.038222	-21.13743564
-0.008788	-2.341664613	-0.036263	-9.352836456	-0.001956	-0.6416681	-0.016815	-9.298989597
-0.015571	-4.149073701	-0.011022	-2.842758829	-0.004089	-1.341401259	-3.61E-04	-0.199639325
0.073735	19.64754668	0.103496	26.69335581	0.03421	11.22263073	0.041107	22.73289119
0.002779	0.740496809	0.019667	5.072449455	0.011907	3.906105352	4.83E-03	2.672180656
0.004299	1.145518454	0.017499	4.51328586	0.00882	2.893411372	-0.004242	-2.345900319
0.001687	0.449520733	0.002059	0.531050665	0.001659	0.544236901	0.000417	0.230608306
-0.000791	-0.210771132	0.004388	1.131738862	0.001929	0.632810718	-1.22E-05	-0.006746814
-2.63E-04	-0.070079403	-0.000783	-0.201948844	0.00072	0.236196847	-1.80E-05	-0.009954315
-3.32E-06	-0.000884653	-3.40E-05	-0.008769171	-7.95E-05	-0.026080068	1.61E-05	0.008903582
0.000388	0.103387104	-0.001267	-0.326780569	-0.002846	-0.933633647	0.002829	1.564486564
-0.001493	-0.397827181	-0.00169	-0.43587937	0.000321	0.105304427	-0.001	-0.55301752
0.005401	1.439159146	-0.006485	-1.672590365	-0.02467	-8.093022512	-0.011677	-6.457585579
-0.011086	-2.953993389	0.004707	1.214014318	0.02686	8.811454587	-0.008326	-4.60442387
0.007324	1.951564819	-0.011118	-2.867518841	-0.02209	-7.246650477	0.010551	5.834887852
-0.014685	-3.912988717	0.016268	4.195790295	0.013775	4.518904949	0.010547	5.832675782
0.019001	5.063037018	0.005428	1.399972321	0.013854	4.544820992	-0.014175	-7.839023344

Programming code in R:

```
# This is the final script where all the calculations of my master's thesis "Estimating the ex-ante tracking error:
emphasis on the forecast of the variance and covariance matrix" are included.
# The structure of this script is as follows: On the first part all the functions that have been created for this
purpose are defined; after defining all the needed functions, all the calculations are done.
#
# Author: Patricia
#
################################################################################
###

################################################################################
##
#                       Definition of functions
################################################################################
##

setwd("C:/Users/Patricia/workspace/TesinaQFB/")
library(fGarch)
library(plyr)

# Function for calculating the dynamics of the variance covariance matrix with the historical volatility model
getHistoricalVolatilities = function(.data, .M)
{
	j = T - .M
	ls_Sigma = vector("list", j)
	for(i in 1:j)
	{
		window = .data[i:(.M+i-1), ]
		ls_Sigma[[i]] = cov(window)
	}
	return(ls_Sigma)
}

# Function for calculating the dynamics of the variance and covariance matrix with the EWMA model
getVolatilities_EWMA = function(.data, .lambda)
{

	Sigma0 = cov(.data)
	vec_histMean = apply(.data, 2, mean)
	numObs = nrow(.data) #Number of observations
	numFact = ncol(.data) #Number of risk factors
	ls_Sigma_EWMA = vector("list", (numObs+1))
	ls_Sigma_EWMA[[1]] = Sigma0

	for(i in 2:(numObs+1))
	{
		Sigma_t = matrix(0, ncol = numFact, nrow = numFact)
		x_t_prev = .data[(i-1), ]
		Sigma_t_prev = ls_Sigma_EWMA[[(i-1)]]
		for(j in 1:numFact)
		{
			for(k in 1:numFact)
			{
				if(k<j)
				{
					Sigma_t[j, k] = Sigma_t[k, j]
```

```
                    } else {

                              Sigma_t[j, k] = as.numeric((1-.lambda) * (x_t_prev[j] -
vec_histMean[j]) * (x_t_prev[k] - vec_histMean[k]) + .lambda * Sigma_t_prev[j, k])
                    }

               }
          }
          ls_Sigma_EWMA[[i]] = Sigma_t
     }
     return(ls_Sigma_EWMA)
}

# Function for calculating the dynamics of the volatility of each risk factor with an univariate GARCH(1,1)
getVolatilities_GARCH = function(.data, .alpha, .beta)
{
     #Get the volatility series for each risk factor via a GARCH(1,1)
     Sigma0 = diag(cov(.data))
     vec_histMean = apply(.data, 2, mean)
     numObs = nrow(.data) #Number of observations
     numFact = ncol(.data) #Number of risk factors
     ma_Sigma_GARCH = matrix(0, ncol = numFact, nrow = (numObs + 1))
     ma_Sigma_GARCH[1, ] = Sigma0
     for(i in 2:(numObs+1))
     {
          ma_Sigma_GARCH[i, ] = (1 - .alpha - .beta) * Sigma0 + .alpha * (as.numeric(.data[(i-1), ]) -
vec_histMean)^2 + .beta * ma_Sigma_GARCH[(i-1), ]
     }
     return(ma_Sigma_GARCH)
}

# Function for obtaining the dynamics of the matrix R, which will be needed for building the DCCGARCH
getMatrixR = function(.E, .alpha, .beta, .ma_Hist_Correl_E = NA)
{
     if(all(is.na(.ma_Hist_Correl_E)))
     {
          ma_histCorrel = cor(.E)
          E = .E
     } else {
          ma_histCorrel = .ma_Hist_Correl_E
          E = t(as.matrix(.E))
     }
     numObs = nrow(E)
     numFact = ncol(E)
     ls_R = vector("list", (numObs+1))
     ls_R[[1]] = ma_histCorrel
     for(i in 2:(numObs+1))
     {
          R_t = diag(numFact)
          R_previous_t = ls_R[[(i-1)]]
          epsilon_t = E[(i-1), ]

          for(j in 1:numFact)
          {
               for(k in 1:numFact)
               {
                    if(j == k)
                    {
                    }
```

```
					else
					{
						if(j>k){R_t[j, k] = R_t[k, j]}
						else {R_t[j, k] = as.numeric(ma_histCorrel[j, k] + .alpha *
(epsilon_t[j] * epsilon_t[k] - ma_histCorrel[j, k]) + .beta * (R_previous_t[j, k] - ma_histCorrel[j, k]))}
					}
				}
			}
			ls_R[[i]] = R_t
		}
		return(ls_R)
}

# Function for calculating the dynamics of the variance and covariance matrix with the DCC GARCH model
getVolatility_DCCGARCH = function(.lsD, .lsR)
{
		D.length = length(.lsD)
		R.length = length(.lsR)
		if(D.length != R.length)
		{
			stop("The input lists must have the same length")
		} else {

			ls_H = vector("list", D.length)
			for(i in 1:D.length)
			{
				ls_H[[i]] = .lsD[[i]] %*% .lsR[[i]] %*% .lsD[[i]]
			}
		}
		return(ls_H)
}

#Function which gives, for a specific value of M, the value of the likelihood function
OptimizeM = function(.M)
{
		j = T - .M #Number of observations in output
		ls_Sigma = getHistoricalVolatilities(df_riskFactorSeries, .M)
		QML = 0 #Initial value of the likelihood function
		checkfield = numeric(T)
		for (i in 1:j)
		{
			Sigma_t = data.matrix(ls_Sigma[[i]])
			x_t = data.matrix(df_riskFactorSeries[i, ])
			if(is.infinite(QML))
			{

			} else {
				if(det(Sigma_t) < 1.120882e-296)
				{
					value_t = Inf
				} else {
					value_t = as.numeric(N * log(2 * pi) + log(det(Sigma_t)) + x_t %*%
solve(Sigma_t) %*% t(x_t))
				}}
			QML = QML + value_t
			checkfield[i] = i
		}
		return(-0.5 * QML)
}
```

```
# Function which gives, for a specific value of the parameter lambda, the value of the likelihood function
OptimizeLambda = function(.lambda)
{
	j = 200 #Number of observations in output
	N = 41
	ls_Sigma = getVolatilities_EWMA(df_riskFactorSeries_cut, .lambda)
	QML = 0 #Initial value of the likelihood function
	checkfield = numeric(j)
	for (i in 1:j)
	{
		Sigma_t = data.matrix(ls_Sigma[[i+1]])
		x_t = data.matrix(df_riskFactorSeries_cut[i, ])
		if(is.infinite(QML))
		{

		} else {
			if(det(Sigma_t) < 1.120882e-294)
			{
				value_t = Inf
			} else {
				value_t = as.numeric(N * log(2 * pi) + log(det(Sigma_t)) + x_t %*%
solve(Sigma_t) %*% t(x_t))
		}}
		QML = QML + value_t
		checkfield[i]=i
	}
	return(-0.5 * QML)
}

# Function for predicting the variance-covariance matrix with the historical model
PredictReturnsAndVolatilities = function(.data, .lastWin, .Sigma_T, .M, n)
{
	ls_PredictedSigma = vector("list", n)
	vec_histMean = apply(.data, 2, mean)
	for(i in 1:n)
	{
		if(i == 1) {Sigma_new = .Sigma_T}
		else {Sigma_new = ls_PredictedSigma[[(i-1)]]}
		x_next_period = numeric(N)
		for(j in 1:N)
		{
			x_next_period[j] = vec_histMean[j] + .data[(T + i - 1), j] + sqrt(Sigma_new[j, j]) *
rnorm(1) #Random walk
		}
		.data[(T+i), ] = x_next_period
		window = .data[(nrow(.data)-.M+1):nrow(.data), ]
		ls_PredictedSigma[[i]] = cov(window)
	}
	return(ls_PredictedSigma)
}

# Function for predicting the variance and covariance matrix with the EWMA model
PredictReturnsAndVolatilities_EWMA = function(.data, .Sigma_EWMA_T, .lambda, n)
{
	ls_PredictedSigma_EWMA = vector("list", n)
	vec_histMean = apply(.data, 2, mean)
	for(i in 1:n)
	{
		if(i == 1)
		{
```

```
				Sigma_last_period = .Sigma_EWMA_T
			} else {
				Sigma_last_period = ls_PredictedSigma_EWMA[[(i-1)]]
			}
			x_next_period = numeric(N)
			Sigma_t = matrix(0, ncol = N, nrow = N)
			for(j in 1:N)
			{
				x_next_period[j] = vec_histMean[j] + .data[(T + i - 1), j] +
sqrt(Sigma_last_period[j, j]) * rnorm(1) #Random walk
				for(k in 1:N)
				{
					if(k<j)
					{
						Sigma_t[j, k] = Sigma_t[k, j]

					} else {

						Sigma_t[j, k] = as.numeric((1-.lambda) * (x_next_period[j] -
vec_histMean[j]) * (x_next_period[k] - vec_histMean[k]) + .lambda * Sigma_last_period[j, k])
					}

				}
			}
			.data[(T+i), ] = x_next_period
			ls_PredictedSigma_EWMA[[i]] = Sigma_t
		}
		return(ls_PredictedSigma_EWMA)
}

# Function for predicting the variance and covariance matrix with the DCC GARCH model
PredictReturnsAndVolatilities_DCCGARCH = function(.data, .ma_VolatilitiesGARCH, .ma_Hist_Correl_E,
.Sigma_DCCGARCH_T, .params, n)
{
		ls_PredictedSigma_DCCGARCH = vector("list", n)
		vec_histMean = apply(.data, 2, mean)
		Sigma0 = apply(.data,2,var)
		ma_Sigma_GARCH = matrix(0, ncol = N, nrow = (n+1))
		ma_Sigma_GARCH[1, ] = Sigma0
		for(i in 1:n)
		{
			if(i == 1)
			{
				Sigma_last_period = .Sigma_DCCGARCH_T
			} else {
				Sigma_last_period = ls_PredictedSigma_DCCGARCH[[(i-1)]]
			}
			x_next_period = numeric(N)
			for(j in 1:N)
			{
				x_next_period[j] = vec_histMean[j] + .data[(T + i - 1), j] +
sqrt(Sigma_last_period[j, j]) * rnorm(1) #Random walk
				ma_Sigma_GARCH[(i+1), j] = (1 - .params[1] - .params[2]) * Sigma0[j] +
.params[1] * (as.numeric(x_next_period[j]) - vec_histMean[j])^2 + .params[2] * ma_Sigma_GARCH[i, j]
#
			}
			.data[(T+i), ] = x_next_period
			E_i = x_next_period / sqrt(ma_Sigma_GARCH[(i+1), ])
			diag_i = sqrt(ma_Sigma_GARCH[(i+1), ])
			D_i = diag(diag_i)
```

```
			R_i = getMatrixR(E_i, .params[1], .params[2], .ma_Hist_Correl_E)[[2]]
			H_i = getVolatility_DCCGARCH(list(D_i), list(R_i))[[1]]
			ls_PredictedSigma_DCCGARCH[[i]] = H_i
		}
		return(ls_PredictedSigma_DCCGARCH)
}

# Function for simulating relative weights (bets) on the different risk factors
SimulateWeights = function(.originalWeights, .benchmarkWeights, .numWeightsToSimulate,
.numSimulations)
{
		numRisks = length(.originalWeights)
		ma_WeightSimulations = matrix(0, ncol = numRisks, nrow = .numSimulations)
		for(i in 1:.numSimulations)
		{
			posToSimulate1 = sample(1:24, .numWeightsToSimulate[1])
			posToSimulate2 = sample(25:30, .numWeightsToSimulate[2])
			posToSimulate3 = sample(31:36, .numWeightsToSimulate[3])
			posToSimulate4 = sample(37:41, .numWeightsToSimulate[4])
			weights = .originalWeights
			sumSelectedWeights_original_1 = sum(weights[posToSimulate1])
			sumSelectedWeights_original_2 = sum(weights[posToSimulate2])
			sumSelectedWeights_original_3 = sum(weights[posToSimulate3])
			sumSelectedWeights_original_4 = sum(weights[posToSimulate4])

			weights[posToSimulate1] = weights[posToSimulate1] * runif(.numWeightsToSimulate[1],
min = 0.8, max = 1.2)
			weights[posToSimulate2] = weights[posToSimulate2] * runif(.numWeightsToSimulate[2],
min = 0.8, max = 1.2)
			weights[posToSimulate3] = weights[posToSimulate3] * runif(.numWeightsToSimulate[3],
min = 0.8, max = 1.2)
			weights[posToSimulate4] = weights[posToSimulate4] * runif(.numWeightsToSimulate[4],
min = 0.8, max = 1.2)

			sumSelectedWeights_simulated_1 = sum(weights[posToSimulate1])
			sumSelectedWeights_simulated_2 = sum(weights[posToSimulate2])
			sumSelectedWeights_simulated_3 = sum(weights[posToSimulate3])
			sumSelectedWeights_simulated_4 = sum(weights[posToSimulate4])

			weights[posToSimulate1] = weights[posToSimulate1] * (sumSelectedWeights_original_1 /
sumSelectedWeights_simulated_1)
			weights[posToSimulate2] = weights[posToSimulate2] * (sumSelectedWeights_original_2 /
sumSelectedWeights_simulated_2)
			weights[posToSimulate3] = weights[posToSimulate3] * (sumSelectedWeights_original_3 /
sumSelectedWeights_simulated_3)
			weights[posToSimulate4] = weights[posToSimulate4] * (sumSelectedWeights_original_4 /
sumSelectedWeights_simulated_4)
			ma_WeightSimulations[i, ] = weights / .benchmarkWeights
			ma_WeightSimulations[i, ][is.na(ma_WeightSimulations[i, ])] = 1

		}
		return(ma_WeightSimulations)
}

# Function for calculating the ex-ante tracking error using a particular series of variance and covariance
matrices and relative exposures
getEATE = function(.ls_PredictedSigma, .sensit, .ma_SimulatedWeights, .T)
{
		EATE = matrix(0, ncol = length(.T), nrow = nrow(.ma_SimulatedWeights))
		Theta = t(as.matrix(.sensit))
```

```
	ones = matrix(rep(1,41), ncol = 41)
	for(i in .T)
	{
		Sigma = .ls_PredictedSigma[[i]]
		for(j in 1:nrow(.ma_SimulatedWeights))
		{
			Omega = as.matrix(.ma_SimulatedWeights[j, ])
			EATE[j, which(.T == i)] = ((Omega - ones) * Theta) %*% Sigma %*% t(((Omega -
ones) * Theta)) * sqrt(i)
		}
	}
	return(EATE)
}

# Function for calculating the ex-post tracking error (EPTE)
getEPTE = function(.securities, .relativeExposures, .originalWeights_pf, .originalWeights_bm, .indicators,
.shares, .returnBenchmark, .additional, .T)
{
	vec_BenchmarkReturns = .returnBenchmark
	additional = .additional
	numSecurities = ncol(.securities)
	originalExposures = .originalWeights_pf / .originalWeights_bm
	originalExposures[is.na(originalExposures)] = 1

	EPTE_V30 = matrix(0, ncol = 3, nrow = nrow(.relativeExposures))
	EPTE_V60 = matrix(0, ncol = 3, nrow = nrow(.relativeExposures))
	EPTE_V120 = matrix(0, ncol = 3, nrow = nrow(.relativeExposures))
	EPTE_V250 = matrix(0, ncol = 3, nrow = nrow(.relativeExposures))
	ls_EPTE = vector("list", 4)
	for(i in 1:nrow(.relativeExposures))
	{
#		print(i)
		relativeExposures_i = .relativeExposures[i, ]
		indicators_i = as.matrix(.indicators[, originalExposures != relativeExposures_i])
		relativeDeviations_i = ((originalExposures - relativeExposures_i) /
relativeExposures_i)[originalExposures != relativeExposures_i]

		shares = .shares
		if(ncol(indicators_i) > 0)
		{

			for(j in 1:ncol(indicators_i))
			{
				combination_j = indicators_i[, j]
				comb_j_curr = as.vector(combination_j[1])
				comb_j_factor = as.vector(combination_j[2])
				positions = NULL

				for(k in 1:numSecurities)
				{
					curr = as.vector(additional[1, k])
					factors = as.vector(additional[2:5, k])
					if(comb_j_curr == curr && comb_j_factor %in% factors)
					{
						positions = c(positions, k)
					}
				}

				sharesToChange = shares[positions]
				sharesToChange.length = length(sharesToChange)
```

```
                                        newShares = sharesToChange + relativeDeviations_i[j] /
sharesToChange.length
                                        shares[positions] = newShares

                              }
                    }
                    vec_finalWeights = shares
                    securities = .securities
                    for(l in 1:length(vec_finalWeights))
                    {
                              securities[, l] = securities[, l] * as.numeric(vec_finalWeights[l])
                    }
                    vec_PortfolioReturns = apply(securities, 1, sum, na.rm = TRUE)
                    vec_ActiveReturns = vec_PortfolioReturns - vec_BenchmarkReturns
                    EPTE_V30[i, 1] = sd(vec_ActiveReturns[(.T + 10 - 29):(.T + 10), ]) * sqrt(10)
                    EPTE_V30[i, 2] = sd(vec_ActiveReturns[(.T + 20 - 29):(.T + 20), ]) * sqrt(20)
                    EPTE_V30[i, 3] = sd(vec_ActiveReturns[(.T + 250 - 29):(.T + 250), ]) * sqrt(250)
                    EPTE_V60[i, 1] = sd(vec_ActiveReturns[(.T + 10 - 59):(.T + 10), ]) * sqrt(10)
                    EPTE_V60[i, 2] = sd(vec_ActiveReturns[(.T + 20 - 59):(.T + 20), ]) * sqrt(20)
                    EPTE_V60[i, 3] = sd(vec_ActiveReturns[(.T + 250 - 59):(.T + 250), ]) * sqrt(250)
                    EPTE_V120[i, 1] = sd(vec_ActiveReturns[(.T + 10 - 119):(.T + 10), ]) * sqrt(10)
                    EPTE_V120[i, 2] = sd(vec_ActiveReturns[(.T + 20 - 119):(.T + 20), ]) * sqrt(20)
                    EPTE_V120[i, 3] = sd(vec_ActiveReturns[(.T + 250 - 119):(.T + 250), ]) * sqrt(250)
                    EPTE_V250[i, 1] = sd(vec_ActiveReturns[(.T + 10 - 249):(.T + 10), ]) * sqrt(10)
                    EPTE_V250[i, 2] = sd(vec_ActiveReturns[(.T + 20 - 249):(.T + 20), ]) * sqrt(20)
                    EPTE_V250[i, 3] = sd(vec_ActiveReturns[(.T + 250 - 249):(.T + 250), ]) * sqrt(250)

          }
          ls_EPTE[[1]] = EPTE_V30
          ls_EPTE[[2]] = EPTE_V60
          ls_EPTE[[3]] = EPTE_V120
          ls_EPTE[[4]] = EPTE_V250
          return(ls_EPTE)
}

# Function for calculating the mean squared error out of two series
getMSE = function(vec_EATE, vec_EPTE)
{
          return(mean((vec_EATE - vec_EPTE)^2))
}

# Function for building a table summarizing all MSE values of a portfolio in a specific period
buildMSETable = function(.EATE_HIST, .EATE_EWMA, .EATE_DCCGARCH, .ls_EPTE)
{
          outputTable = matrix(0, ncol = 12, nrow = 3)
          for(i in 1:4)
          {
                    currentEPTEWindow = .ls_EPTE[[i]]
                    for (j in 1:3)
                    {
                              outputTable[1, (i*3-3+j)] = getMSE(.EATE_HIST[, j], currentEPTEWindow[, j])
                              outputTable[2, (i*3-3+j)] = getMSE(.EATE_EWMA[, j], currentEPTEWindow[, j])
                              outputTable[3, (i*3-3+j)] = getMSE(.EATE_DCCGARCH[, j], currentEPTEWindow[,
j])
                    }
          }
          df_MSE = as.data.frame(outputTable)
          colnames(df_MSE) = paste(c(rep("V30", 3), rep("V60", 3), rep("V120", 3), rep("V250", 3)), c("T+10",
"T+20", "T+250"))
          rownames(df_MSE) = c("HIST", "EWMA", "DCC_GARCH")
```

```
	return(df_MSE)
}

#########################################################################################
#####################
#                    Calculations
#########################################################################################
#####################



# Importing the data
data_1_07 = read.csv(file="pf1_p1.csv",head=TRUE,sep=";")
data_1_08 = read.csv(file="pf1_p2.csv",head=TRUE,sep=";")
data_1_09 = read.csv(file="pf1_p3.csv",head=TRUE,sep=";")
data_1_11 = read.csv(file="pf1_p4.csv",head=TRUE,sep=";")
data_2_07 = read.csv(file="pf2_p1.csv",head=TRUE,sep=";")
data_2_08 = read.csv(file="pf2_p2.csv",head=TRUE,sep=";")
data_2_09 = read.csv(file="pf2_p3.csv",head=TRUE,sep=";")
data_2_11 = read.csv(file="pf2_p4.csv",head=TRUE,sep=";")

#adapt manually for each series
df_riskFactorSeries = data_1_07[, -c(1:3, 45)]

T = nrow(df_riskFactorSeries) #Number of observations in time
N = ncol(df_riskFactorSeries) #Number of risk factors

# Obtaining the optimal window size M
intervalForM = seq(20,500,5)
LikelihoodFuntionValue = numeric(max(intervalForM))
for(k in intervalForM)
{
	LikelihoodFuntionValue[k] = OptimizeM(k) #Applies the function OptimizeM for a grid of values for
M
}

# Plotting the results and saving the plot as a pdf file
pdf("plot_M_4.pdf") #adapt manually!
plot(intervalForM, LikelihoodFuntionValue[intervalForM], type = "l", lwd = 2, xlab = "M", ylab = "Likelihood
Function Value")
ymax = max(LikelihoodFuntionValue[is.finite(LikelihoodFuntionValue)],na.rm=TRUE)
xmax = which(LikelihoodFuntionValue==ymax)
points(xmax,ymax, col="red", lwd=5)
x = as.character(xmax)
text((xmax+20),ymax, x, col = "red", lwd=2)
dev.off()

M = 50 # Define the value of the optimal M
ls_Sigma_Historical = getHistoricalVolatilities(df_riskFactorSeries, M) #Obtain historical variance and
covariance matrices

lastWin = df_riskFactorSeries[(nrow(df_riskFactorSeries)-M+1):nrow(df_riskFactorSeries), ]
Sigma_T = ls_Sigma_Historical[[length(ls_Sigma_Historical)]]

numPredictionPeriods = 20

# Predict variance and covariance matrices with the historical method
ls_PredictedSigma = PredictReturnsAndVolatilities(df_riskFactorSeries, lastWin, Sigma_T, M,
numPredictionPeriods)
```

```
#Importing sensitivities and weights

sensit_1_07 = read.csv(file="coefficients1.csv",head=FALSE,sep=";")#Sensitivities portfolio 1, period up to 2007
sensit_1_08 = read.csv(file="coefficients2.csv",head=FALSE,sep=";")#Sensitivities portfolio 1, period up to 2008
sensit_1_09 = read.csv(file="coefficients3.csv",head=FALSE,sep=";")#Sensitivities portfolio 1, period up to 2009
sensit_1_11 = read.csv(file="coefficients4.csv",head=FALSE,sep=";")#Sensitivities portfolio 1, period up to 2011
sensit_2_07 = read.csv(file="coefficients5.csv",head=FALSE,sep=";")#Sensitivities portfolio 2, period up to 2007
sensit_2_08 = read.csv(file="coefficients6.csv",head=FALSE,sep=";")#Sensitivities portfolio 2, period up to 2008
sensit_2_09 = read.csv(file="coefficients7.csv",head=FALSE,sep=";")#Sensitivities portfolio 2, period up to 2009
sensit_2_11 = read.csv(file="coefficients8.csv",head=FALSE,sep=";")#Sensitivities portfolio 2, period up to 2011

weights_pf_1_07 = as.numeric(read.csv(file="weights1.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 1, period up to 2007
weights_bm_07 = as.numeric(read.csv(file="weights1.csv",head=TRUE,sep=";")[2, ])#Weights bechmark, period up to 2007
weights_pf_1_08 = as.numeric(read.csv(file="weights2.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 1, period up to 2008
weights_bm_08 = as.numeric(read.csv(file="weights2.csv",head=TRUE,sep=";")[2, ])#Weights bechmark, period up to 2008
weights_pf_1_09 = as.numeric(read.csv(file="weights3.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 1, period up to 2009
weights_bm_09 = as.numeric(read.csv(file="weights3.csv",head=TRUE,sep=";")[2, ])#Weights bechmark, period up to 2009
weights_pf_1_11 = as.numeric(read.csv(file="weights4.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 1, period up to 2011
weights_bm_11 = as.numeric(read.csv(file="weights4.csv",head=TRUE,sep=";")[2, ])#Weights bechmark, period up to 2011
weights_pf_2_07 = as.numeric(read.csv(file="weights5.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 2, period up to 2007
weights_pf_2_08 = as.numeric(read.csv(file="weights6.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 2, period up to 2008
weights_pf_2_09 = as.numeric(read.csv(file="weights7.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 2, period up to 2009
weights_pf_2_11 = as.numeric(read.csv(file="weights8.csv",head=TRUE,sep=";")[1, ])#Weights portfolio 2, period up to 2011

# Simulating the relative exposures (bets) to the risk factors
numBets = c(4,2,2,2)
numSimulations = 100000 #Number of different vectors of relative exposures to simulate
ma_RelativeExposures_1_07 = SimulateWeights(weights_pf_1_07, weights_bm_07, numBets, numSimulations)#Matrix where each row represents a vector of relative exposures
ma_RelativeExposures_1_08 = SimulateWeights(weights_pf_1_08, weights_bm_08, numBets, numSimulations)
ma_RelativeExposures_1_09 = SimulateWeights(weights_pf_1_09, weights_bm_09, numBets, numSimulations)
ma_RelativeExposures_1_11 = SimulateWeights(weights_pf_1_11, weights_bm_11, numBets, numSimulations)
ma_RelativeExposures_2_07 = SimulateWeights(weights_pf_2_07, weights_bm_07, numBets, numSimulations)
ma_RelativeExposures_2_08 = SimulateWeights(weights_pf_2_08, weights_bm_08, numBets, numSimulations)
ma_RelativeExposures_2_09 = SimulateWeights(weights_pf_2_09, weights_bm_09, numBets, numSimulations)
ma_RelativeExposures_2_11 = SimulateWeights(weights_pf_2_11, weights_bm_11, numBets, numSimulations)

evaluationPeriods = c(10,20) # T+10, T+20
# Calculating the ex-ante tracking error using the historical variance covariance matrices
```

```
EATE = getEATE(ls_PredictedSigma, sensit_2_11, ma_RelativeExposures_2_11, evaluationPeriods) #the inputs of this function have to be adapted manually for each portfolio/period
EATE = as.data.frame(EATE)
EATE$V3 = EATE$V2 * sqrt(12)
#head(sqrt(EATE))

write.csv(sqrt(EATE),"EATE_HIST_1_07.csv") #adapt manually

write.csv(ls_EPTE_1_08[[1]], file = "EPTE_1_08_V30.csv")
write.csv(ls_EPTE_1_08[[2]], file = "EPTE_1_08_V60.csv")
write.csv(ls_EPTE_1_08[[3]], file = "EPTE_1_08_V120.csv")
write.csv(ls_EPTE_1_08[[4]], file = "EPTE_1_08_V250.csv")


# EWMA model

# Finding the optimal value of the parameter lambda
intervalForLambda = seq(0.6,.95,0.05)
LikelihoodFuntionValue = numeric(length(intervalForLambda))
for(k in intervalForLambda)
{
	LikelihoodFuntionValue[which(intervalForLambda == k)] = OptimizeLambda(k)
}

# Plotting the results and saving the plot as pdf
pdf("plot_lambda_1.pdf") #adapt manually
plot(intervalForLambda, LikelihoodFuntionValue, type = "l", lwd = 2, xlab = "lambda", ylab = "Likelihood Function Value")
ymax = max(LikelihoodFuntionValue[is.finite(LikelihoodFuntionValue)],na.rm=TRUE)
xmax = intervalForLambda[which(LikelihoodFuntionValue==ymax)]
points(xmax,ymax, col="red", lwd=5)
x = as.character(xmax)
text((xmax+.02),ymax, x, col = "red", lwd=2)
dev.off()

lambda = .85 # Defining lambda with the optimal value obtained previously
T = nrow(df_riskFactorSeries)

# Obtaining the dynamics for the variance-covariance matrix with the EWMA method
ls_Sigma_EWMA = getVolatilities_EWMA(df_riskFactorSeries, lambda)
Sigma_T_EWMA = ls_Sigma_EWMA[[length(ls_Sigma_EWMA)]]

# Predicting the variance-covariance matrices with the EWMA method
ls_PredictedSigma_EWMA = PredictReturnsAndVolatilities_EWMA(df_riskFactorSeries, Sigma_T_EWMA, lambda, numPredictionPeriods)

# Calculating the ex ante tracking error with the EWMA variance covariance matrices
EATE_EWMA = getEATE(ls_PredictedSigma_EWMA, sensit_1_07, ma_RelativeExposures_1_07, evaluationPeriods)#adapt manually inputs
EATE_EWMA = as.data.frame(EATE_EWMA)
EATE_EWMA$V3 = EATE_EWMA$V2 * sqrt(12)
#head(sqrt(EATE_EWMA))

write.csv(sqrt(EATE_EWMA),"EATE_EWMA_1_07.csv")#adapt manually


# DCC GARCH

# Optimize GARCH Parameters using the package fGarch
params = matrix(0, ncol = 3, nrow = 41)
```

```
for(i in 1:41)
{
	riskSerie_i = df_riskFactorSeries[, i]
	params[i, ] = (garchFit(data = riskSerie_i))@fit$par[2:4]
}

sumParams = apply(params, 1, sum)

for(i in 1:41)
{
	params[i, ] = params[i, ] / sumParams[i]
}

params = apply(params,2,mean)
params=params[2:3]

# Build the volatility series for each risk factor using a GARCH(1,1) model
alpha = params[1]
beta = params[2]

ma_VolatilitiesGARCH = getVolatilities_GARCH(df_riskFactorSeries, alpha, beta)

# Build the additional matries E, R and Dt for the DCC GARCH
E = df_riskFactorSeries / sqrt(ma_VolatilitiesGARCH[2:nrow(ma_VolatilitiesGARCH), ])
ls_D = vector("list", 100)
for(i in 1:100)
{
	diag_i = sqrt(ma_VolatilitiesGARCH[2:101, ][i, ])
	ls_D[[i]] = diag(diag_i)
}

ls_R = getMatrixR(E, alpha, beta)
ls_R = ls_R[-1]

# Calculate the variance and covariance dynamics with the DCC GARCH model
ls_Volatilities_DCCGARCH = getVolatility_DCCGARCH(ls_D, ls_R)

#Predict the variance and covariance matrices using this method
numPredictionPeriods = 20

ma_Hist_Correl_E = cor(E)
sigma_T_DCCGARCH = ls_Volatilities_DCCGARCH[[length(ls_Volatilities_DCCGARCH)]]

ls_PredictedSigma_DCCGARCH = PredictReturnsAndVolatilities_DCCGARCH(df_riskFactorSeries,
ma_VolatilitiesGARCH, ma_Hist_Correl_E, sigma_T_DCCGARCH, params, numPredictionPeriods)

#Obtain the ex-ante tracking error using the DCC GARCH volatilities
EATE_DCCGARCH = getEATE(ls_PredictedSigma_DCCGARCH, sensit_1_07, ma_RelativeExposures_1_07,
evaluationPeriods)#adapt manually
EATE_DCCGARCH = as.data.frame(EATE_DCCGARCH)
EATE_DCCGARCH$V3 = EATE_DCCGARCH$V2 * sqrt(12)
#head(sqrt(EATE_DCCGARCH))
write.csv(sqrt(EATE_DCCGARCH), "EATE_DCCGARCH_1_07.csv") #save the results in a csv file


# Calculate the ex-post tracking error

indicators = read.csv(file="indicators.csv",head=TRUE,sep=";")#Used to match deviations in risk exposures
with weights of each security in the portfolio
# For each portfolio and every period we import the corresponding files
```

```
securities_1_07 = read.csv(file="securities1.csv",head=FALSE,sep=";")
shares_1_07 = read.csv(file="shares1.csv",head=FALSE, sep=";")
return_Benchmark_07 = read.csv(file="rb1.csv",head=TRUE,sep=";")
additional_1_07 = read.csv(file="additional1.csv",head=FALSE,sep=";")

securities_1_08 = read.csv(file="securities2.csv",head=FALSE,sep=";")
shares_1_08 = read.csv(file="shares2.csv",head=FALSE, sep=";")
return_Benchmark_08 = read.csv(file="rb2.csv",head=TRUE,sep=";")
additional_1_08 = read.csv(file="additional2.csv",head=FALSE,sep=";")

securities_1_09 = read.csv(file="securities3.csv",head=FALSE,sep=";")
shares_1_09 = read.csv(file="shares3.csv",head=FALSE, sep=";")
return_Benchmark_09 = read.csv(file="rb3.csv",head=TRUE,sep=";")
additional_1_09 = read.csv(file="additional3.csv",head=FALSE,sep=";")

securities_1_11 = read.csv(file="securities4.csv",head=FALSE,sep=";")
shares_1_11 = read.csv(file="shares4.csv",head=FALSE, sep=";")
return_Benchmark_11 = read.csv(file="rb4.csv",head=TRUE,sep=";")
additional_1_11 = read.csv(file="additional4.csv",head=FALSE,sep=";")

securities_2_07 = read.csv(file="securities51.csv",head=FALSE,sep=";")
shares_2_07 = read.csv(file="shares5.csv",head=FALSE, sep=";")
return_Benchmark_07 = read.csv(file="rb1.csv",head=TRUE,sep=";")
additional_2_07 = read.csv(file="additional5.csv",head=FALSE,sep=";")

securities_2_08 = read.csv(file="securities6.csv",head=FALSE,sep=";")
shares_2_08 = read.csv(file="shares6.csv",head=FALSE, sep=";")
return_Benchmark_08 = read.csv(file="rb2.csv",head=TRUE,sep=";")
additional_2_08 = read.csv(file="additional6.csv",head=FALSE,sep=";")

securities_2_09 = read.csv(file="securities7.csv",head=FALSE,sep=";")
shares_2_09 = read.csv(file="shares7.csv",head=FALSE, sep=";")
return_Benchmark_09 = read.csv(file="rb3.csv",head=TRUE,sep=";")
additional_2_09 = read.csv(file="additional7.csv",head=FALSE,sep=";")

securities_2_11 = read.csv(file="securities8.csv",head=FALSE,sep=";")
shares_2_11 = read.csv(file="shares8.csv",head=FALSE, sep=";")
return_Benchmark_11 = read.csv(file="rb4.csv",head=TRUE,sep=";")
additional_2_11 = read.csv(file="additional8.csv",head=FALSE,sep=";")


T_1_07 = 261 #These values indicate the position at which we will start the predictions
T_1_08 = 260
T_1_09 = 261
T_1_11 = 261
T_2_07 = 261
T_2_08 = 260
T_2_09 = 261
T_2_11 = 261

# Applying the function for calculating the EPTE for each portfolio/period
ls_EPTE_1_07 = getEPTE(securities_1_07, ma_RelativeExposures_1_07, weights_pf_1_07, weights_bm_07,
indicators, shares_1_07, return_Benchmark_07, additional_1_07, T_1_07)
ls_EPTE_1_08 = getEPTE(securities_1_08, ma_RelativeExposures_1_08, weights_pf_1_08, weights_bm_08,
indicators, shares_1_08, return_Benchmark_08, additional_1_08, T_1_08)
ls_EPTE_1_09 = getEPTE(securities_1_09, ma_RelativeExposures_1_09, weights_pf_1_09, weights_bm_09,
indicators, shares_1_09, return_Benchmark_09, additional_1_09, T_1_09)
ls_EPTE_1_11 = getEPTE(securities_1_11, ma_RelativeExposures_1_11, weights_pf_1_11, weights_bm_11,
indicators, shares_1_11, return_Benchmark_11, additional_1_11, T_1_11)
```

```
ls_EPTE_2_07 = getEPTE(securities_2_07, ma_RelativeExposures_2_07, weights_pf_2_07, weights_bm_07,
indicators, shares_2_07, return_Benchmark_07, additional_2_07, T_2_07)
ls_EPTE_2_08 = getEPTE(securities_2_08, ma_RelativeExposures_2_08, weights_pf_2_08, weights_bm_08,
indicators, shares_2_08, return_Benchmark_08, additional_2_08, T_2_08)
ls_EPTE_2_09 = getEPTE(securities_2_09, ma_RelativeExposures_2_09, weights_pf_2_09, weights_bm_09,
indicators, shares_2_09, return_Benchmark_09, additional_2_09, T_2_09)
ls_EPTE_2_11 = getEPTE(securities_2_11, ma_RelativeExposures_2_11, weights_pf_2_11, weights_bm_11,
indicators, shares_2_11, return_Benchmark_11, additional_2_11, T_2_11)

#MSE

# Build the MSE summary table for each case and export it as csv
tableMSE = buildMSETable(EATE_HIST_1_07, EATE_EWMA_1_07, EATE_DCCGARCH_1_07, ls_EPTE_1_07)
#adapt manually

write.csv(tableMSE, file = "tableMSE_2_11.csv")
```