Yield Curve Modeling with Principal Component Analysis for Market Risk Assessment

Andrea Ranzato

Id: 4709494

Supervisor: Prof. Alessia Pini

Scienze bancarie, finanziarie e assicurative - Economia

M.Sc. Statistical and actuarial sciences

Profile: Data analytics for business and economics

Università Cattolica del Sacro Cuore di Milano

Yield Curve Modeling with Principal Component Analysis for Market Risk Assessment

Andrea Ranzato

April 2020

Abstract

This dissertation illustrates how *principal components* computed on a time series of US yield curves identify the most common kind of movements occurring in interest rates of different maturities. In addition, it is shown that the significant PCs can be employed to build a *risk factor model* for an interest rate sensitive portfolio comprised of US Government bonds which might be used to achive interest risk immunization. Conversely to traditional price sensitivity measures of bond securities, such as *duration*, these models take into account *non parallel shifts* of the yield curve, usually known as *tilt* and *curvature*. Furthermore, we investigate the empirical distribution of the eigenvalues and eigenvectors resulting from the spectral decomposition of the sample covariance matrix of interest rates changes using the *bootstrap*.

Dedication

To my beloved Mum and Dad.

Declaration

If I should identify the main reason which led me to tackle this topic is neither finance, nor statistics per se. On the other hand, even if I am interested in those subjects, I found the drive to write this dissertation because I knew it would have given me the opportunity to deepen my understanding of Linear Algebra which was mainly limited to the computational part. What I was lacking was the intuition behind the concepts, such as eigenvalues and eigenvectors. In doing this, I was facilitated by the precious lectures of Professor Gilbert Strang on the `MIT OpenCourseWare` platform and his best-seller *Introduction to Linear Algebra*. In addition, I found the videos authored by Stanford graduate Grant Sanderson on his `3Blue1Brown` YouTube channel enlightening with his wonderful animations. He literally makes mathematics alive!

I knew that a deeper of study of principal component analysis would have inevitably led me at the heart of Linear Algebra with eigenvalue and eigenvectors and spectral theorem. In fact, principal components can be recovered either by the diagonalization of a symmetric sample covariance matrix, or by singular value decomposition of a sample data matrix. I am happy to have done this choice because I feel that my general understanding of multivariate statistics has also improved a bit.

As a second motivation, I liked the idea to face a topic which I consider to be at the intersection between my bachelor in economics and my master in data analysis, even if I did not have the chance to study finance systematically during my first three years. Thus, this was the opportunity to gain a better idea of the complexity surrounding the world of finance too.

Moreover, during these months, I discovered that trying to combine different fields of knowledge is not easy, yet, enjoyable because it gives the chance to spot how they relate with each other. However, in this process, I witnessed that mathematics is often key to most of them. The more you know it, the more access you have to different subjects. For this reason, I decided to devote part of the time to study linear algebra and include the topics relevant for a better understanding of principal component analysis in Chapter 2, even if they might have taken for granted.

In producing this writing, I tried, hopefully with some coherence, to combine

some math, statistics, finance and economics and a bit of programming, functional-based in R, and object-oriented in Python. I wish I mastered all of these field, but I do not!

In addition, since most of the topics addressed were new to me, in writing this dissertation, I have inevitably relied on some resources more than others. In particular I found extremely useful Vol. II of the *Market risk analysis* series written by Alexander Carol which discusses the use of principal component analysis on the term structure of interest rates. The factor model representation in Chapter 4 is adapted from this resource.

Overall, I would say that this writing shows the process that led me to have a better understanding of principal component analysis as a starting point for potential applications in market risk analysis. In this respect, I will try to mention in the conclusion further directions and improvements which could have taken into consideration within the analysis. For instance, I believe that the sampling function that originated the bootstrap samples (see Appendix B) should have taken into consideration also the autocorrelation and cross-correlation of the time series of interest rates, in order to achieve a distribution of the eigenvalues and loadings valid in the short run.

Acknowledgements

Above all, I want to thank my supervisor Prof. Alessia Pini for the independence she let me to pursue my own interests and guidance which turned crucial when she suggested to apply PCA on the curve estimated with Svensson's model, instead of relying on cash flows mapping which was particularly complex to achieve given my current capabilities in R. Nevertheless, I leave it as challenging programming exercise for the future.

In addition, I want to thank Prof. Alessandro Sbuelz to have borne with my faulty financial knowledge and to have shown me the seminal paper of Litterman and Scheinkman (1991) which surprisingly was not cited in Alexander (2008a) (actually, the unique flaw I found in that book which provided so much help).

Now, it is time to mention the non academic supporting staff!

First of all, my grandmother Giovanna to have always been by my side.

My dear friend Lino to have supported me even in the hardest moments. Much of the memories of these five academic years will be tied to you. Thank you to be a loyal friend.

My lifelong friend Filippo with whom I've spent the most enjoyable and fun moments of my life.

My closer friends: Andrea, Rita, Bruno, Matilde, Agostino and Lorenzo.

Professor Toni, who helped me to overcome my conflictual relationship with mathematics during the first years of high school. More importantly, he showed me that mathematics can be much more than dry calculations, which are also important, but more revealing if tight with good visual intuition of the concepts (where possible). Since I got his help, I started considering mathematics differently.

My whole "enlarged"family. You all have supported and borne with me during these years. In particular, Mariateresa who took care of me as I was her child, and Bianca who provided last minute corrections to the text. So many *s* were missing!

Last but not least my "almost"brothers Pietro and Andrea.

I am truly grateful you are part of my family, and I love you all.

I would like to remember also the sanitary personnel working in the current state of emergency due to *Covid-19*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The first ones ever to apply principal component analysis to the U.S. interest rates term structure were Litterman and Scheinkman (1991) with their seminal paper *Common factors affecting bonds returns* that marked the beginning of a field of research with the aim of identifying "the common factors that affect the returns on U.S. government bonds", in order to account for non-parallel shifts of the yield curve. In fact, traditional risk measures such as duration and convexity are limited to quantify the variation in the prices of bond securities caused only by parallel movements of the yield curve. However, it is well known, and outlined specifically in Jones (1991), that the yield curve observed between two points in time might exhibit fluctuations which are different from a parallel one. Conversely, besides the *parallel shift*, the yield curve might manifest movements originally identified with the following terms: *steepness*, and *curvature*. The former specifies movements characterized by greater absolute changes of interest rates at shorter maturities compared to the longer ones, resulting in an overall declining shape, whereas the latter refers to changes having an inverted bump shape ("U"shape).

It turns out that such movements reflect, in order of importance, the shapes assumed by the first eigenvectors resulting from the *spectral decomposition* of the sample covariance (or correlation) matrix of interest rates changes, whilst the corresponding eigenvalues signal their importance relative to the others.

Given the informational value inherent in the eigenvectors on the kind of movements impacting the yield curve, and since the principal components are computed multiplying the original p interest rates with the associated eigenvector, it is not impossible to realize that the resulting transformed variables will identify with increasing or decreasing trend the type of interest rates change occurring between consecutive time points. Moreover, the principal components have the nice prop-

erty of being *uncorrelated* between each others.

In conclusion, the original p interest rates variables can be approximated with high accuracy using a reduced set k of principal components which are orthogonal and in the following numerical relationship with the original ones: $k << p$.

After Litterman and Scheinkman (1991), later studies applied principal components on the term structure of various countries having the aim of quantifying the number of factors and related variance explained in the system of interest rates changes. A summary of the results accomplished by these empirical analysis is provided by Fabozzi (2012).

Differently from most of the other works on the topic which employ principal component analysis mainly as a *descriptive* tool, Barber and Copper (2012) investigate the persistence over time of the amount of variance explained by the first principal components, providing proper measures of uncertainty associated to the eigenvalues. In particular, they find out that "although the first two components explain 93% of the sample variation within a 90% confidence interval, the remaining components make statistically significant contribution to the covariance matrix".

It is worth mentioning that before the contribution of Anderson, 1963, principal component analysis was restricted to being purely a descriptive methodology (see Pearson, 1901). Nevertheless, the paper *Asymptotic theory for Principal Component Analysis* provided the methodological framework to consider the quantities generated by PCA as realizations from underlying probability distributions. We will try to approximate the theoretical distribution of those quantities using the bootstrap in Chapter 4.

This dissertation is structured as follows. Chapter 2 introduces briefly the intuitions underlying linear algebra concepts which are instrumental in understanding principal component analysis (PCA) from a theoretical point of view. Particular attention is devoted to emphasize the meaning of eigenvalues and eigenvectors resulting from the spectral decomposition of a sample covariance matrix.

The third chapter discusses the issue of interest risk from a broad perspective and shows how variations in the required yield affect the prices of coupon bonds. In addition, it introduces the concept of the yield curve which is used by investors to gauge investment opportunities and to properly discount future cash flows. Furthermore, it illustrates price sensitivity measures such as *duration* and PV01. Finally, it shows how PCA is applied to the US term structure of interest rates in order to attain an accurate *principal component representation* of the interest rates changes using only the first three components.

Chapter 4 is devoted to investigate the persistence of the typical structure of the eigenvectors found in Chapter 3 throughout different time windows. Then, it shows how the principal component representation of the interest rates changes is used to

achieve a *linear risk factor model* which can be used for risk management purposes.

The last chapter summarizes the main findings and suggests potential future refinements of the analysis.

1.2 "Beyond Econometrics"

This section will not have further developments in the following chapters. Nevertheless, it aims to just mention what are the latest advancements in the financial research which adopt a combination of statistical and computing techniques, proper of *machine learning*, in a variety of activities of the investment process. One of the leading researcher in this domain is Marcos Lopez de Prado, professor at the Cornell University[1]. In particular, he points out in de Prado (2018), that innovation has been driven in this sector mainly by three interrelated factors: (1) the availability of a variety of data in forms different from traditional market data, (2) the surge in computational power and, (3) the need to account for non linearities hardly catched by traditional econometrics tools such as principal component analysis (*codependence* measures).

First of all: "the unstructured nature of [these previously unseen data], along with the complexity of the phenomena they measure, means that many of these datasets are beyond the grasp of econometric analysis"(de Prado, 2019, p. 2). Hence, machine learning techniques are required in order to account for the requested flexibility and numerical power. Yet, his concern is directed to build awareness of the fact that applications of machine learning algorithms to finance deserve a stand alone treatment due to the complexity and specificity of this domain. For this reason, and given the impact that finance has on society as whole, financial institutions should implement ad hoc teams with specific training in the field in order to replicate the successful results that machine learning have proven outside finance.

[1]Marcos Lopez de Prado's website with publications: http://www.quantresearch.org and De Prado discussing the use of *AI* at Bloomberg, https://www.bloomberg.com/news/videos/2019-12-17/cornell-s-lopez-de-prado-sees-financial-reckoning-with-ai-video

Chapter 2

Principal Component Analysis (PCA)

2.1 Introduction

Applications of statistical analysis in a variety of disciplines, from natural sciences to finance and economics, require manipulation of data in the form of arrays with rows and columns. Each row records measurements performed on distinct *observational units* across different features, also called *variables*. This particular data structure is represented by a well defined mathematical entity known as *matrix*. Matrices stores data in a tabular fashion making the description of multivariate statistical models convenient. In fact, "[Tabular] arrangement emphasizes the relationships of the data both *within* an observational unit or row and *within* a variable or column. Simple operations on the data matrix may reveal relationships *among* observational units or *among* variables"(Gentle, 2017, p. 331).

Furthermore, matrices have close digital counterparts in computer memories. Hence, not only are they a tool to represent statistical models on paper in a compact way, but also they are the fundamental data structure to store effectively large amounts of data on a personal computer. As a result, this fact, along with the rise in the computational power, has unlocked the ability to run numerical algorithms on large-scale sample datasets, in order to draw meaningful conclusions and predictions[1]. For example, notwithstanding the optimism of Karl Pearson in his seminal paper regarding principal component analysis on the possibility to compute principal components on more than three or four variables using pen and paper, it could have been hardly imaginable in 1900 to run PCA on a multivariate time-series com-

[1] *Computer Age Statistical Inference* is the title of the book of Efron and Hastie, 2016 in which the impact of the computational power on statistical inference is discussed. We will employ a simple but powerful computer intensive techniques in Chapter 4, known as *Bootstrap*, elaborated by Efron himself, in order to get an estimate of the eigenvalues and loadings distribution of PCA.

prising eleven treasury interest rates of eleven different maturities, over a 14-year time frame. Surprisingly, nowadays, this is not even a load of work for a regular PC.

Interestingly, Karl Pearson was also responsible of introducing the use of matrix notation within statistical analysis with his chi-square paper in 1900 (Efron and Hastie, 2016). Since then, matrix theory has surged in importance, in particular, in contexts where huge amount of data needs to be manipulated efficiently. For this reason, according to professor Gilbert Strang[2], understanding matrices and their language is essential in every application which makes use of data.

This language is *linear algebra* which provides the tools to examine and manipulate matrices. For instance, the principal components of a set of variables can be elegantly found, among other algebraic methods, through the spectral decomposition of a covariance matrix, or directly on the sample data X^TX, using singular value decomposition. Both factorizations build on the concepts of eigenvalues and eigenvectors of a matrix which will be presented in the following sections.

Even if mathematics and statistics require rigour, the following sections do not pretend to be a formal exposition of linear algebra concepts. In fact, many concepts such as subspaces, and basis will not be mentioned even if they will be present silently. Instead, the focus will be mainly on ideas and intuitions instrumental in building the geometrical intuitions behind principal component analysis, whose conception was built thanks to the free Linear Algebra course on `MIT OpenCourseWare`[3] and Strang (2016). Moreover, Grant Sanderson provided enlightening animations of linear algebra concepts in his `3Blue1Brown`[4] YouTube channel. Each of these resources were extremely useful in writing the following sections.

This chapter is structured as follows. Section 2.2 sets the notation and introduces the concept of the variance-covariance matrix. Next, section 2.3 introduces the multiplication $\boldsymbol{Ax}$, viewed as the *linear transformation* operated by matrix $\boldsymbol{A}$ on vector $\boldsymbol{x}$. In addition, it demonstrates the particular case in which vector $\boldsymbol{x}$ is not rotated, but just lengthened or shortened by $\boldsymbol{A}$. In that case matrix multiplication is equivalent to scalar multiplication, and it turns out that the scaling factor is an eigenvalue of the matrix A, whereas x is the associated eigenvector. Section 2.4 presents the diagonalization of a square matrix, whose counterpart for square and symmetric matrices is known as *spectral decomposition*, which is the subject of Section 2.5. Subsequently, Section 2.6 examines *positive definite* matrices. This property makes symmetric square matrices even more powerful in statistical applications. Just to

[2]Gilbert Strang has been Professor of the world-famous course `18.06 Linear algebra` at the Massachusetts Institute of Technology. He is an advocate of the need to expose students to more Linear algebra in their curriculum as stated in his pamphlet *Too much calculus*. He has recently published a new book entitled *Linear algebra and learning from data*.

[3]https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/

[4]https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AJtAw/featured

mention, variance-covariance matrices are happily *all* square, symmetric, and positive (semi) definite. Finally Section 2.7 uses the concepts introduced in the previous sections to illustrate *principal components analysis* from a theoretical point of view. Later on, Chapter 3 will show the application of PCA on US Treasury interest rates, with the objective to extract their most common movements using past data. Then, the principal components with higher explanatory power will be employed to build a reduced model able to approximate the interest rates dynamics. That model will then be used to make considerations on the risk factors underlying an interest rates sensitive portfolio.

2.2 Basic Notation

The building blocks of matrix theory and data organization are *vectors* which are vertical arrays of n numbers. In particular, within statistical analysis, a vector stores a sample of n realizations from a random variable X. For instance, a vector might contain 240 daily observations of the 3-month spot interest rate.

$$\underset{(n\times 1)}{\boldsymbol{x}} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2.1}$$

The *transpose* of a vector have dimension $1 \times n$.

$$\boldsymbol{x^T} = \begin{pmatrix} x_1, & x_2, & \dots & , x_n \end{pmatrix} \tag{2.2}$$

More generally, we are interested in n realizations generated from p-dimensional random vector $\boldsymbol{X^T} = (\, X_1, X_2, \dots, X_p \,)$. The n observations across p variables are stored in rectangular $n \times p$ *matrix*. Each entry represents the *i-th* observation on the *k-th* variable.

$$\underset{(n\times p)}{\boldsymbol{X}} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1k} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2k} & \dots & x_{2p} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ik} & \dots & x_{ip} \\ \vdots & \vdots & & \vdots & & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nk} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} | & | & & | & & | \\ \boldsymbol{x_1} & \boldsymbol{x_2} & \dots & \boldsymbol{x_k} & \dots & \boldsymbol{x_p} \\ | & | & & | & & | \end{pmatrix} \tag{2.3}$$

The transpose of a matrix has dimension $p \times n$.

$$\boldsymbol{X^T} = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{pmatrix} = \begin{pmatrix} — & \boldsymbol{x_1^T} & — \\ — & \boldsymbol{x_2^T} & — \\ & \vdots & \\ — & \boldsymbol{x_p^T} & — \end{pmatrix} \tag{2.4}$$

If $n = p$, the matrix is *square*. Eigenvalues and eigenvectors are computed only on square matrices, as we will see in the next section. In addition, if a square matrix has off-diagonals entries equal to each other, then the matrix is said to be *symmetric*.

The *identity matrix* is a particular kind of symmetric square matrix which has diagonal entries equal to one and zero on the others. non c'entra con la frase sopra. Si può aggiungere che l'identità è una matrice quadrata simmetrica particolare, con uno sulla diagonale e zero altrove For instance, a 3×3 identity matrix is as follows:

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

Later on, we will constantly deal with a particular type of square symmetric matrix: the variance-covariance matrix of a p-dimensional random vector $\boldsymbol{x}$.

$$\underset{(p\times p)}{\boldsymbol{\Sigma}} = \mathrm{Cov}(\boldsymbol{x}) = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_p) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_p, X_1) & \mathrm{Cov}(X_p, X_2) & \dots & \mathrm{Var}(X_p) \end{pmatrix} \tag{2.6}$$

This matrix will be at the center of out investigation since it incorporates *linear* "relationships"between the p components of the random vector $\boldsymbol{X}$. Once a sample of data is collected, we shall replace population variance and covariance with their respective *sample estimates*. The sample variance defines a measure of spread around the sample mean $\overline{\boldsymbol{x}}$, whilst sample covariance is a measure of linear association between the measurements made on the k-th and p-th variables.

Another useful result for our purposes is the variance of a linear combinations of random variables.

$$\mathrm{Var}(aX_1 + bX_2) = a^2\mathrm{Var}(X_1) + b^2\mathrm{Var}(X_2) + 2ab\,\mathrm{Cov}(X_1, X_2) \tag{2.7}$$

$$= a^2\sigma_{11} + b^2\sigma_{22} + 2ab\,\sigma_{12} \tag{2.8}$$

The relation above can be rewritten using vector notation as follows, where $\boldsymbol{c^T} =$

$(\,a,\,b\,)$, and $X = (\,X_1,\,X_2\,)$:

$$aX_1 + bX_2 = \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \boldsymbol{c^T X} \tag{2.9}$$

Then, if the random vector has variance-covariance matrix $\boldsymbol{\Sigma}$, equation 2.8 becomes:

$$\text{Var}(aX_1 + bX_2) = \text{Var}(\boldsymbol{c^T X}) = \boldsymbol{c^T \Sigma c} \tag{2.10}$$

given that

$$\boldsymbol{c^T \Sigma c} = \begin{pmatrix} a, & b \end{pmatrix} \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = a^2\sigma_{11} + 2ab\,\sigma_{12} + b^2\sigma_{22}$$

We refer to Johnson and Wichern, 2014 or Strang, 2016 for the presentation of basic matrix operations and representation. However, it is worth highlighting a major difference between them. The former adopts a statistical approach, hence the distinction between *sample vector* and *random vector* is made clear since the beginning. The latter, on the other hand, deals with deterministic matrices. However, vectors made of random variables are introduced in Chapter 12, in which the author shows the usefulness of linear algebra to multivariate statistics. In the next chapter, $\boldsymbol{X}$ will represent a realization of a multivariate time serie with dimension $T \times p$, where, $t = 1, \ldots, T$ represent time periods, and $i = 1, \ldots, p$ a set of interest rates associated to different maturities. Part of the future analysis will be a cross-sectional study devoted to quantifying the amount of correlation existing among those interest rates.

2.3 Eigenvalues and Eigenvectors of a Matrix

The intuition underpinning eigenvalues and eigenvectors is straightforward after having appreciated the geometrical meaning of matrix multiplication. In fact, $\boldsymbol{Ax}$ is truly the matrix A performing a *linear transformation* on vector x. In other words, a matrix $\boldsymbol{A}$ moves a vector $\boldsymbol{x}$ to vector $\boldsymbol{b}$. This fact lies at the heart of linear algebra and it can be represented as:

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{2.11}$$

In Strang (2016), the left term of equation 2.11 is seen as a combination of the columns of A. For instance, multiplying vector $\boldsymbol{x} = \left(\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\right)$ by matrix $\boldsymbol{A} = \left(\begin{smallmatrix} 2 & -3 \\ 4 & -4 \end{smallmatrix}\right)$

moves vector x to vector $\boldsymbol{b} = \left(\begin{smallmatrix}1\\4\end{smallmatrix}\right)$. In matrix notation:

$$\begin{pmatrix} 2 & -3 \\ 4 & -4 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

is equivalent to:

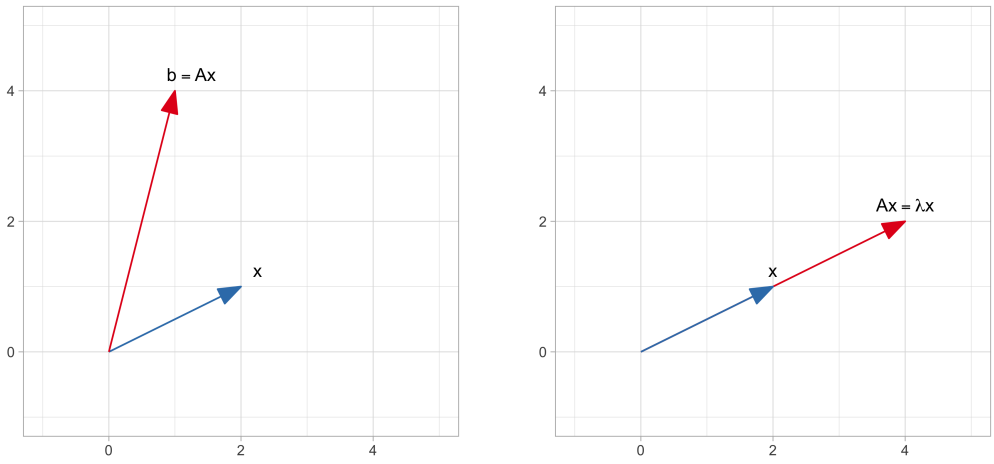
$$2 \begin{pmatrix} 2 \\ 4 \end{pmatrix} + 1 \begin{pmatrix} -3 \\ -4 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

Figure 2.1a shows the intuition behind matrix multiplication of the example above, in which the resulting vector $b = Ax$ lie in a distinct direction compared to x. This is what happens most of the time when a matrix multiplies a vector.

Conversely, figure 2.1b shows a more interesting case in which $\boldsymbol{x}$ is unchanged, but $\boldsymbol{A} = \left(\begin{smallmatrix}4 & -4\\2 & -2\end{smallmatrix}\right)$. In this instance, we note that vector Ax lies in the same direction as x. Hence, matrix A transforms x in the same way as the scalar $\lambda = 2$ would do. In matrix notation we have the following representation:

$$\begin{pmatrix} 4 & -4 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2 \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

If this is the case, then x is a special vector called *eigenvector* of A. In other words,



(a) $b = Ax$ (b) $Ax = \lambda x$

Figure 2.1: Matrix Multiplication

if x is an eigenvector of A, then multiplying x by either the matrix A or the scalar λ is equivalent. Thus, the scalar λ is an *eigenvalue* of matrix A, whose associated eigenvector is x. Generally speaking we have:

$$\boldsymbol{Ax} = \lambda \boldsymbol{x} \tag{2.12}$$

The eigenvalue λ regulates the magnitude of the linear transformation carried out by A on the eigenvector x.

Finding the eigenvalues and eigenvectors of a matrix requires solving a linear system of equations arising from equation 2.12. If the eigenvalue λ is known, its related eigenvector x is searched in the "nullspace"of $\boldsymbol{A} - \lambda \boldsymbol{I}$, in the following way:

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \tag{2.13}$$

In other words, recalling the geometrical intuition of matrix-vector multiplication, equation 2.12 asks to find the *non-zero* vector x that "lands"at the origin when multiplied by matrix $\boldsymbol{A} - \lambda \boldsymbol{I}$. This situation is verified only if $\boldsymbol{A} - \lambda \boldsymbol{I}$ does not have an inverse (singular matrix).

Given λ, x is an eigenvector of A, if and only if, $\boldsymbol{A} - \lambda \boldsymbol{I}$ is a singular matrix.

Then, $\boldsymbol{A} - \lambda \boldsymbol{I}$ is singular (i.e. does not have an inverse) if its determinant is zero:

$$det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0 \tag{2.14}$$

Equation 2.14 is a polynomial of degree n in λ called "*characteristic polynomial*". Its roots are the eigenvalues of matrix A. When A is $n \times n$, equation 2.14 has degree n. Then, A has *potentially* a number n of eigenvalues with n corresponding eigenvectors.

To recap, the procedure to find eigenvalues and eigenvectors of a matrix A is the following:

1. Matrix A is $n \times n$ matrix with real entries.
2. Find the roots of the characteristic polynomial $det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0$. The solutions are the eigenvalues of A.
3. For each eigenvalue λ, solve $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$ to find its corresponding eigenvector $\boldsymbol{x}$.

Lately, we have found the eigenvalue $\lambda = 2$ for $\boldsymbol{A} = \left(\begin{smallmatrix} 4 & -4 \\ 2 & -2 \end{smallmatrix}\right)$. Now, for illustration purposes we search for *all* the eigenvalues of A using the procedure mentioned above.

- *Step 1.*

$$\boldsymbol{A} - \lambda \boldsymbol{I} = \begin{pmatrix} 4 - \lambda & -4 \\ 2 & -2 - \lambda \end{pmatrix}$$

- *Step 2.*

$$\begin{aligned}\det(\boldsymbol{A}-\lambda\boldsymbol{I}) &= \det\begin{pmatrix}4-\lambda & -4\\ 2 & -2-\lambda\end{pmatrix}\\ &= (4-\lambda)(-2-\lambda)-(-4)(2)\\ &= \lambda^2-2\lambda\end{aligned}$$

- *Step 3.*

$$\lambda^2-2\lambda=0$$

Therefore, the eigenvalues of $\boldsymbol{A}$ are $\lambda_1=0$ and $\lambda_2=2$.

Next, the eigenvectors are the solution of $(\boldsymbol{A}-\lambda\boldsymbol{I})\boldsymbol{x}=\boldsymbol{0}$ evaluated first on $\lambda_1=0$ and then in $\lambda_2=2$.

- $\lambda_1=0$.

$$(\boldsymbol{A}-0\boldsymbol{I})\boldsymbol{x}_1=\begin{pmatrix}4 & -4\\ 2 & -2\end{pmatrix}\begin{pmatrix}y\\ z\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix}$$

- $\lambda_2=2$.

$$(\boldsymbol{A}-2\boldsymbol{I})\boldsymbol{x}_2=\begin{pmatrix}2 & -4\\ 2 & -4\end{pmatrix}\begin{pmatrix}y\\ z\end{pmatrix}=\begin{pmatrix}0\\ 0\end{pmatrix}$$

Then, the eigenvectors of $\boldsymbol{A}$ are $\boldsymbol{x}_1=\left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)$ for $\lambda_1=0$ and $\boldsymbol{x}_2=\left(\begin{smallmatrix}2\\1\end{smallmatrix}\right)$ for $\lambda_2=2$.

Figure 2.2, shows the following facts for the example shown above: (a) any vector along the directions (z,z) and $(2y,z)$ are eigenvectors for matrix A; (b) A gives a set of linearly independent eigenvectors because each one contributes to identify distinct directions. Thus, each of them adds "new"information to the system. As we will see in the next section, independent eigenvectors for $\boldsymbol{A}$ are the evidence that the matrix is "diagonalizable"; (c) the eigenvectors are displayed as *unit vectors*; (d) the eigenvectors, even if they are independent, they are not orthogonal between each other.

Eigenvectors and eigenvalues can be found easily using R or Python. In R, the function to compute eigenvalues and eigenvectors of a square matrix is `eigen()` which is provided in the base environment. Instead, Python requires loading one of the following libraries for scientific computations: `NumPy` or `SciPy`. Both libraries

have dedicated modules called `numpy.linalg` and `scipy.linalg`. However, the latter offers more elaborated functions. R, as well as Python, divides the eigenvectors by its length to obtain unit vectors. One additional result useful to check if we got

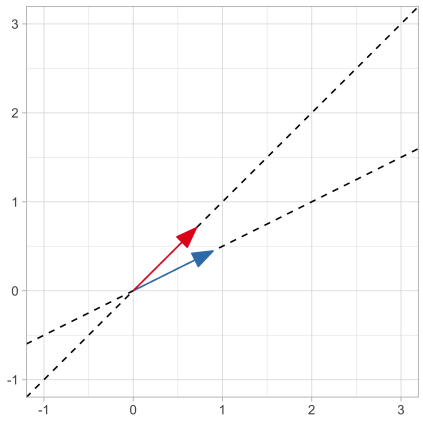


Figure 2.2: Matrix A has a set of independent eigenvectors

the eigenvalues correctly is to compare the *trace* and determinant of the matrix A with the sum and product of the eigenvalues respectively. In fact:

> The product of the n eigenvalues equals the determinant. The sum of the n eigenvalues equals the sum of the n diagonal entries.

2.4 Matrix Diagonalization

Diagonalization brings eigenvalues and eigenvectors of a matrix A together by means of equation $\boldsymbol{AX} = \boldsymbol{X\Lambda}$, in which the columns are $\boldsymbol{Ax}_k = \lambda_k \boldsymbol{x}$. The following definition of diagonalization is taken from Strang (2016).

> Suppose the $p \times p$ matrix A has p linearly independent eigenvectors
> $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$.
> Put them into the columns of an *eigenvector matrix* $\boldsymbol{X}$.
> Then $X^{-1}AX$ is the *eigenvalue matrix* $\boldsymbol{\Lambda}$.

$$\boldsymbol{X}^{-1}\boldsymbol{AX} = \boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{pmatrix} \tag{2.15}$$

For instance, since matrix $\boldsymbol{A} = \left(\begin{smallmatrix} 4 & -4 \\ 2 & -2 \end{smallmatrix}\right)$, has two linearly independent eigenvectors (see fig. 2.2), then it is diagonalisable. Using R, we can verify relationship 2.2 for this matrix. The function `solve()` is used to find the inverse of the eigenvector

matrix $\boldsymbol{X}$.

$$\begin{pmatrix} 2.24 & -2.24 \\ -1.41 & 2.83 \end{pmatrix} \begin{pmatrix} 4 & -4 \\ 2 & -2 \end{pmatrix} \begin{pmatrix} 0.89 & 0.71 \\ 0.45 & 0.71 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \tag{2.16}$$

Equation 2.15 can be rewritten also into the following way which turns out to be useful within the context of spectral decomposition in the following section.

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1} \tag{2.17}$$

To conclude, we remark the following facts:

1. If the eigenvalues of a matrix A are all distinct, then the eigenvectors are independent. As a consequence the eigenvector matrix will be *invertible*. If this is valid, then matrix A can be diagonalized. Otherwise, if A has duplicated eigenvalues (the technical term here is "algebraic multiplicity"), then it has linearly dependent eigenvectors, hence it cannot be diagonalized.

2. The order of the columns of the eigenvectors matrix determines the order of the eigenvalues in Λ.

2.5 Spectral decomposition

Diagonalization of symmetric matrices is a particular type of matrix factorization known as *spectral decomposition* which turns out to be one of the viable strategies to compute principal components, our ultimate objective. Within the context of principal component analysis, spectral decomposition involves the diagonalization of covariance or correlation matrices of a set of variables. These two matrices have the nice property, among the others, of being symmetric.

A symmetric matrix have special features regarding its eigenvalues and eigenvectors:

1. Symmetric matrices have only real eigenvalues.

2. The eigenvectors are orthogonal.

As a consequence, *every symmetric matrix can be diagonalized*, since its eigenvectors are linearly independent and orthogonal. The diagonalization of a symmetric matrix is: $S = X\Lambda X^{-1}$. However, since the eigenvector matrix is the orthogonal Q for which $Q^T = Q^{-1}$ is true, then, the diagonalization can be rewritten as $S = Q\Lambda Q^T$. In Strang (2016), the definition of symmetric diagonalization, also known as spectral decomposition, is as follow:

Every symmetric matrix S has the factorization $S = Q\Lambda Q^T$ with real eigenvalues in Λ and orthonormal eigenvectors in the columns of Q:

$$\boldsymbol{S = Q\Lambda Q^{-1} = Q\Lambda Q^T} = \lambda_1 \boldsymbol{q_1 q_1}^{\boldsymbol{T}} + \ldots + \lambda_n \boldsymbol{q_n q_n^T} \qquad (2.18)$$

with $Q^{-1} = Q^T$.

For a 2×2 symmetric matrix, the spectral spectral decomposition $S = Q\Lambda Q^T$ has a nice geometrical representation which is depicted in Figure 2.3. In this case, matrix $\boldsymbol{S} = \left(\begin{smallmatrix} 2.50 & 2.44 \\ 2.44 & 2.43 \end{smallmatrix}\right)$ is a symmetric covariance matrix resulting from a sample of two highly correlated variables. The first multiplication $Q\Lambda$ results in "stretching"the orthonormal eigenvectors contained in Q, by the corresponding eigenvalues in Λ. Figure 2.3b shows that the effect of eigenvalue $\lambda_1 = 4.90$ overwhelms $\lambda_2 = 0.03$. Finally, $(Q\Lambda)Q^T$ rotates back to the original matrix S.

$$\begin{aligned} \begin{pmatrix} 2.50 & 2.44 \\ 2.44 & 2.43 \end{pmatrix} &= \begin{pmatrix} -0.71 & 0.70 \\ -0.70 & -0.71 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} -0.71 & -0.70 \\ 0.70 & -0.71 \end{pmatrix} \\ &= \begin{pmatrix} -3.5 & 0.02 \\ -3.44 & -0.02 \end{pmatrix} \begin{pmatrix} -0.71 & -0.70 \\ 0.70 & -0.71 \end{pmatrix} \end{aligned} \qquad (2.19)$$

To recap, a 2×2 symmetric matrix can be factorized as:

$$\boldsymbol{S = Q\Lambda Q^{-1}} = \begin{pmatrix} \boldsymbol{q_1} & \boldsymbol{q_2} \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{q_1^T} \\ \boldsymbol{q_2^T} \end{pmatrix} \qquad (2.20)$$

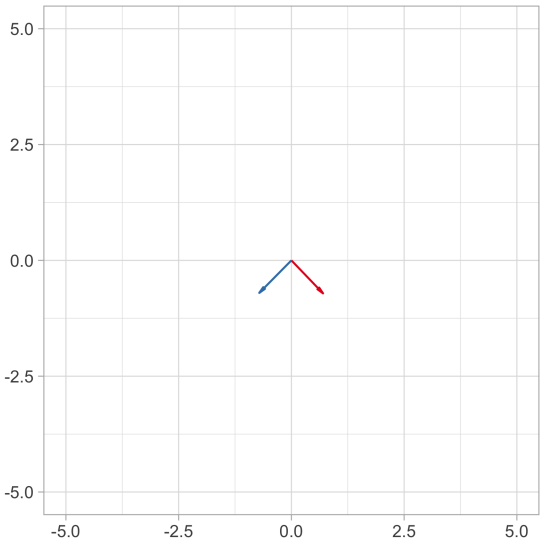
in which λ_1, λ_2 are the eigenvalues of $\boldsymbol{S}$ and $\boldsymbol{q_1}$, $\boldsymbol{q_2}$ are the associated orthonormal eigenvectors. Therefore, $\boldsymbol{q_1^T q_1 = 1}$ and $\boldsymbol{q_1^T q_2 = 0}$.

2.6 Positive Definite Matrices

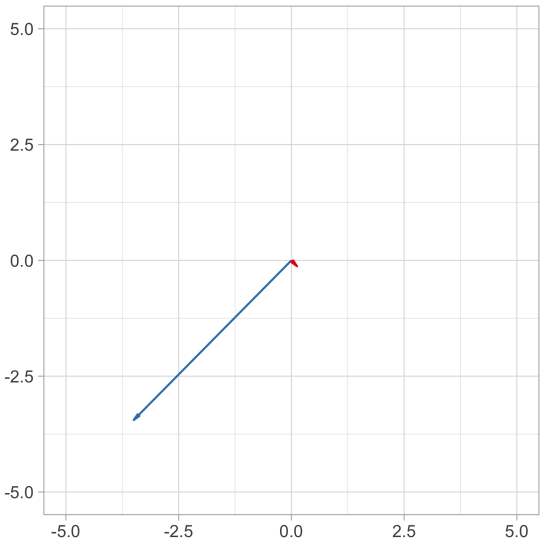
The first section introduced square matrices from a general point of view and dealt with their eigenvalues and eigenvectors computation. Subsequently, we showed that symmetric square matrices have real eigenvalues.

This section makes one further step by adding the feature of positive definiteness to symmetric square matrices. These matrices have *all positive* eigenvalues. As we see, the more features a matrix has, the more properties it presents.

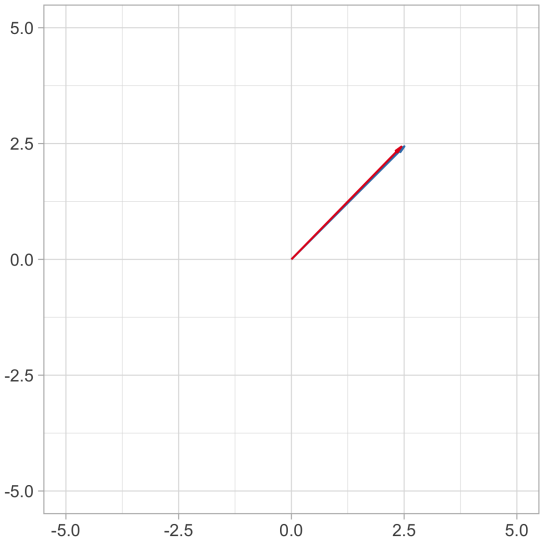
In order to test if a matrix symmetric matrix S is positive definite we study the sign of its associated *quadratic form* $Q(x, y) = \boldsymbol{x^T S x}$. From Strang (2016):



(a) Rotation



(b) Stretch



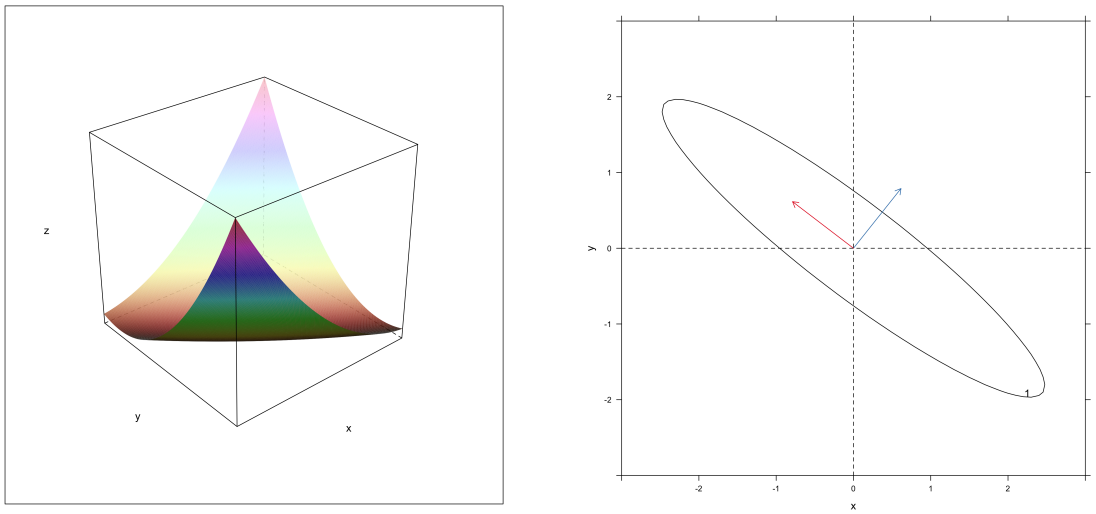
(c) Rotate back

Figure 2.3: Geometry behind spectral decomposition

S is positive definite if $\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{x} > 0$ for *every nonzero* vector $\boldsymbol{x}$:

$$\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{x} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} a & b \\ b & c \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = ax^2 + 2bxy + cy^2 > 0 \quad (2.21)$$

For instance, let be $S_1 = \left(\begin{smallmatrix} 1.09 & 1.26 \\ 1.26 & 1.72 \end{smallmatrix}\right)$ a sample covariance matrix of two variables. Then, the associated quadratic form $Q(x, y) = 1.09x^2 + 2.52xy + 1.72y^2$ is a surface illustrated in Figure 2.4a. As we can see, for each (x, y), $Q(x, y)$ is positive. Thus, S_1 is positive definite. As we should have expected its eigenvalues are not only real but also all positive: $\lambda_1 = 2.7$ and $\lambda_2 = 0.10$. In addition, Figure 2.4b shows the contour plot $Q(x, y) = 1$ of the associated quadratic form and its orthonormal eigenvectors identifying a new coordinate system.

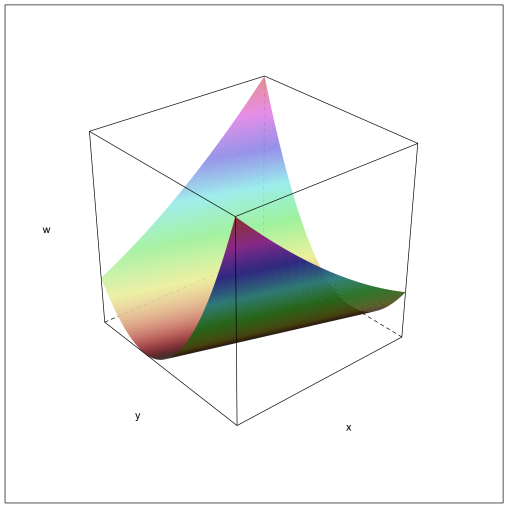


(a) $Q(x, y) = 1.09x^2 + 2.52xy + 1.72y^2$ (b) $1.09x^2 + 2.52xy + 1.72y^2 = 1$.

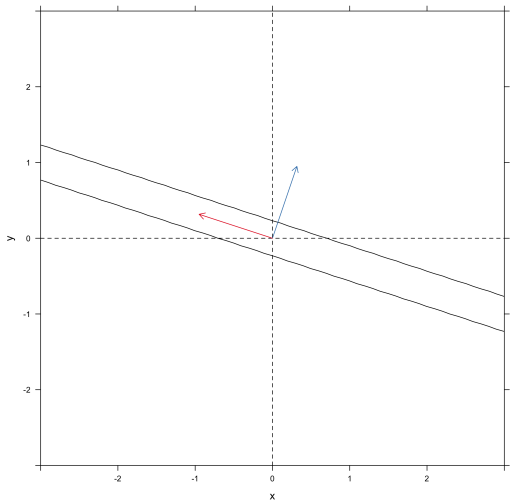
Figure 2.4: Quadratic form surface and contour plot associated to the positive definite matrix S_1.

Conversely, when $Q(x, y) = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{S}\boldsymbol{x} \geq 0$, for *all* (x, y), S is positive *semidefinite*. For instance, let be $S_2 = \left(\begin{smallmatrix} 2 & 6 \\ 6 & 18 \end{smallmatrix}\right)$ a population covariance matrix. This matrix is positive semidefinite, in fact there exists an entire set of points (a subspace in $\mathbb{R}^2$), – identified by the eigenvector in red (see, Figure 2.5b), for which $Q(x, y) = 0$. If S_2 were the covariance matrix of a multivariate normal model, then our observed sample would be closely tight to that subspace with correlation almost equal to one. In other words, if the process generating the data were to have the matrix S_2 to generate the data, then the two variables would be perfectly dependent. This fact is confirmed by the fact that the correlation matrix of S_2 is $C = \left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right)$.

To conclude, variance-covariance matrices are generally positive definite. However, when the experiments are *dependent*, i.e. perfectly correlated, then the covariance matrix is positive *semidefinite*. Further explanation can be found in Chapter



(a) $Q(x, y) = 2x^2 + 12xy + 18y^2$



(b) $2x^2 + 12xy + 18y^2 = 1$.

Figure 2.5: Quadratic form surface and contour plot associated to the positive *semidefinite* matrix S_2.

12 of Strang (2016). In the following section, spectral decomposition will be applied to positive definite covariance matrices, in order to figure out principal components.

2.7 PCA

Principal component analysis is a widely adopted technique in multivariate statistical investigation. Its objective is to reproduce a substantial portion of the variance-covariance structure of a large set of p variables using a smaller number k ($k \ll p$) of *uncorrelated* new variables, known as *principal components*, which are linear combinations of the original ones. "Geometrically, these linear combinations represent the selection of a new coordinate system, (...) [whose] axes represent the directions with maximum variability and provide a simpler and more parsimonious description of the covariance structure"(Johnson and Wichern, 2014, p. 430-431).

For instance, Figure 2.6 illustrates a scatterplot of observations when $p = 2$. The two arrows represent the orthonormal eigenvectors obtained through the spectral decomposition of the sample covariance matrix of two features `MAT1YR` and `MAT1MO`. Both vectors identify the directions of maximum variability having equations:

$$Z_1 = 0.712 \times \texttt{MAT1MO} + 0.701 \times \texttt{MAT1YR}$$

$$Z_2 = 0.701 \times \texttt{MAT1MO} - 0.712 \times \texttt{MAT1YR}$$

As we can see, the eigenvector in blue identifies the most informative component in terms of variability. As a consequence, the component in red might be discarded

without losing much explanatory power. Needless to say, principal component analysis extends easily to higher dimensions when $p > 2$. For example, if $p = 3$, PCA would have identified three planes in $\mathbb{R}^3$ instead of lines.

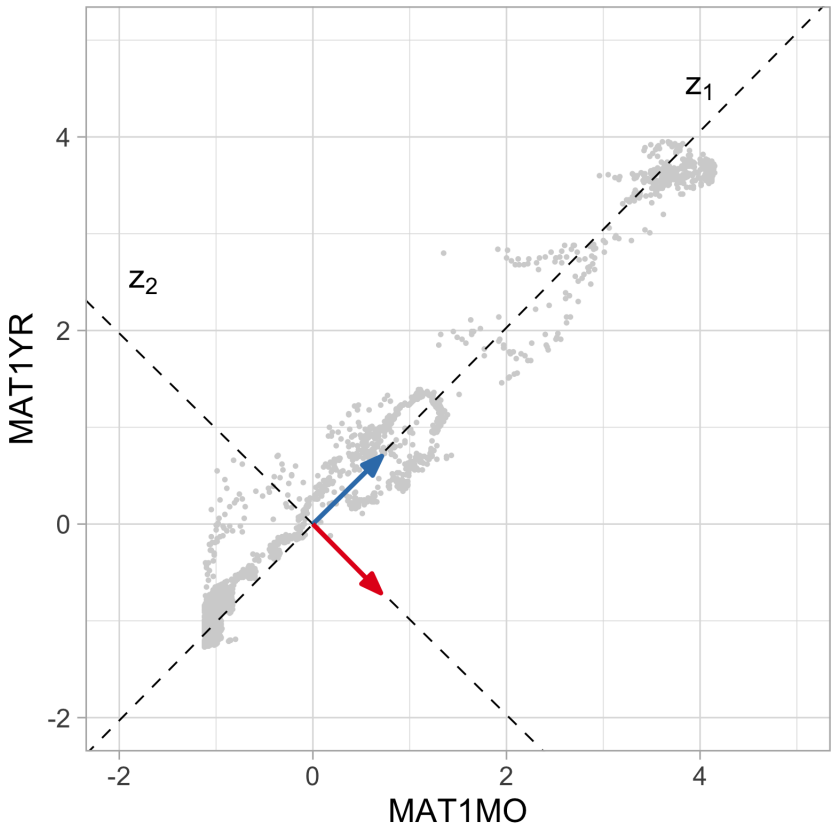


Figure 2.6: Eigenvectors shows direction of maximum variability

Jolliffe (2002)[5] gives a brief overview of the history of PCA highlighting the different approaches that have been adopted to derive principal components. The first method employed was geometrical in nature, and it is attributable to Karl Pearson with his article entitled *Lines and Planes of Closest Fit to Systems of Points in Space* published in 1901. Conversely, Hotelling in 1933 adopted a strategy based on algebraic arguments based on Lagrange multipliers "ending up with an eigenvalue/eigenvector problem"of a *correlation* matrix. Subsequently, in 1936 Girshick adopted a statistical approach to the problem and he "introduced the idea that sample PCs were maximum likelihood estimates of underlying population PCs". The latter approach results to be of interest when the study is directed on drawing conclusions on the distribution of certain parameters rather than descriptive ones. We

[5]Furthermore, he identifies the following four remarkable contributions to PCA literature. Firstly, the writing by Anderson in 1963 *Asymptotic theory for principal component analysis*. Secondly, Rao's paper of 1964 *The use and interpretation of principal component analysis in applied research*. Next, Gower's work *Some distance properties of latent root and vector methods used in multivariate analysis* published in 1966. Finally, *Two case studies in the application of principal component analysis* due to Jeffers in 1967.

will try to approximate theoretical PCs distributions of the eigenvalues and loadings for our problem using the Bootstrap, in Chapter 4.

An alternative method to determine principal components relies in the application of the singular value decomposition of a *rectangular* sample data matrix X. This technique is presented in Strang (2016), and provides advantages in terms of computational efficiency and geometrical insights of what PCA does.

Suppose the random vector of p elements $\boldsymbol{x^T} = (X_1, X_2, \ldots, X_p)$ has covariance matrix $\boldsymbol{\Sigma}$ (see, 2.6), and $\boldsymbol{a_k^T} = (a_{1k}, a_{2k}, \ldots, a_{pk})$ is a vector of p constants with $k = 1, 2, \ldots, p$. Further, consider the random vector $\boldsymbol{z^T} = (\boldsymbol{x^T a_1}, \boldsymbol{x^T a_2}, \ldots, \boldsymbol{x^T a_p}) = (Z_1, Z_2, \ldots, Z_p)$ whose elements are weighted sums of vector $\boldsymbol{x}^T$. Then, compute the variances and covariances of random variables in $\boldsymbol{z}^T$, using properties of a linear combination of random variables, in the following way:

$$\mathrm{Var}(Z_k) = \boldsymbol{a_k^T \Sigma a_k} \qquad k = 1, \ldots, p \tag{2.22}$$

$$\mathrm{Cov}(Z_i, Z_k) = \boldsymbol{a_i^T \Sigma a_k} \qquad i, j = 1, \ldots k \tag{2.23}$$

which can be collected into the variance-covariance matrix:

$$\mathrm{Cov}(\boldsymbol{z}) = \begin{pmatrix} \mathrm{Var}(Z_1) & \mathrm{Cov}(Z_1, Z_2) & \ldots & \mathrm{Cov}(Z_1, Z_p) \\ \mathrm{Cov}(Z_2, Z_1) & \mathrm{Var}(Z_2) & \ldots & \mathrm{Cov}(Z_2, Z_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(Z_p, Z_1) & \mathrm{Cov}(Z_p, Z_2) & \ldots & \mathrm{Var}(Z_p) \end{pmatrix} \tag{2.24}$$

The objective of PCA is to find principal components which are *uncorrelated* linear combinations $Z_1, Z_2, \ldots, Z_p$ whose variances in (2.25) are as large as possible, under the constraints $\boldsymbol{a_k^T a_k} = 1$, in order to keep the variance finite. Thus, roughly speaking, we want to find those vectors $\boldsymbol{a_k}$, for $k = 1, \ldots, p$, such that (2.25) and (2.26) are maximized and zero respectively.

In Tsay (2010), the problem of finding the principal components $\boldsymbol{z}^T$, is articulated as follows:

1. The first principal component of $\boldsymbol{x}$ is the linear combination $Z_1 = \boldsymbol{x^T a_1}$ that maximizes $\mathrm{Var}(Z_1)$ subject to constraint $\boldsymbol{a_1^T a_1} = 1$.

2. The second principal component of $\boldsymbol{x}$ is the linear combination $Z_2 = \boldsymbol{x^T a_2}$ that maximizes $\mathrm{Var}(Z_2)$ subject to the constraints $\boldsymbol{a_2^T a_2} = 1$ and $\mathrm{Cov}(Z_2, Z_1) = 0$.

3. The *k-th* principal component of $\boldsymbol{x}$ is the linear combination $Z_k = \boldsymbol{x^T a_k}$ that maximizes $\mathrm{Var}(Z_k)$ subject to the constraints $\boldsymbol{a_k^T a_k} = 1$ and $\mathrm{Cov}(Z_i, Z_k) = 0$ for $k \neq i$.

As stated previously, a variety of strategies have been adopted to solve it. Within

our case, the spectral decomposition of the covariance matrix $\boldsymbol{\Sigma}$ is the method of choice to work out the PCs.

Thus, from (2.6) we recall that $(\lambda_1, \boldsymbol{q_1}), (\lambda_2, \boldsymbol{q_2}), \ldots, (\lambda_p, \boldsymbol{q_p})$ are the eigenvalue-eigenvectors pairs resulting from the spectral decomposition of the symmetric positive definite $\boldsymbol{\Sigma}$. Further, suppose the eigenvalues are arranged in decreasing order $\lambda_1, \lambda_2, \ldots, \lambda_p \geq 0$. Then we have the following result[6]:

> The *k-th* principal component of $\boldsymbol{x}$ is the random variable $Z_k = \boldsymbol{x^T q_k}$ for $k = 1, \ldots, p$. Furthermore,
>
> $$\mathrm{Var}(Z_k) = \boldsymbol{q_k^T \Sigma q_k} \qquad k = 1, \ldots, p \tag{2.25}$$
> $$\mathrm{Cov}(Z_i, Z_k) = \boldsymbol{q_i^T \Sigma q_k} \qquad i \neq k \tag{2.26}$$
>
> Moreover,
>
> $$\sum_{i=1}^{p} \mathrm{Var}(X_i) = \mathrm{trace}(\boldsymbol{\Sigma}) = \sum_{i=1}^{p} \lambda_i = \sum_{i=1}^{p} \mathrm{Var}(Z_i) \tag{2.27}$$
>
> Thus,
>
> $$\frac{\mathrm{Var}(Z_i)}{\sum_{i=1}^{p} \mathrm{Var}(X_i)} = \frac{\lambda_i}{\lambda_1 + \cdots + \lambda_p} \tag{2.28}$$

As a consequence, the total covariation of the random vector $\boldsymbol{x}$ is given by the sum of the eigenvalues of $\boldsymbol{\Sigma}$ which also represent the individual variances of each principal component Z_i. "A major aim of PCA is to use only a reduced set of principal components to represent the original variables $[(X_1, \ldots, X_p)]$"(Alexander, 2008a, p.50). Hence, when p is large, we wish PCA to output a set of p transformed variables, i.e. the principal components, in which the first k ($k << p$) of them are able to explain most of the covariation of the original system.

Now consider a sample of n observations for each of the p variables in vector $\boldsymbol{x}$ included in matrix $\boldsymbol{X}$ with dimension $n \times p$. The transformed variables are contained in the columns of $\boldsymbol{Z}$ and are the result of the following matrix multiplication:

$$\underset{(n\times p)}{\boldsymbol{Z}} = \underset{(n\times p)(p\times p)}{\boldsymbol{X}\,\boldsymbol{A}} \tag{2.29}$$

[6]The formulas concerning PCA are taken from Tsay (2010) which discusses PCA within the context of financial time series. Conversely to Johnson and Wichern (2014) or Hastie et al. (2013), it adopts the approach based on the spectral decomposition of the covariance matrix, whereas the other two give a proof based on constrained optimization.

$$\begin{pmatrix} | & & | & & | \\ \boldsymbol{z_1} & \dots & \boldsymbol{z_k} & \dots & \boldsymbol{z_p} \\ | & & | & & | \end{pmatrix} = \begin{pmatrix} | & & | & & | \\ \boldsymbol{x_1} & \dots & \boldsymbol{x_k} & \dots & \boldsymbol{x_p} \\ | & & | & & | \end{pmatrix} \begin{pmatrix} | & & | \\ \boldsymbol{a_1} & \dots & \boldsymbol{a_p} \\ | & & | \end{pmatrix}$$

Each *k-th* principal component in Z, is produced by "weighting" *all* the columns in X with the coefficients in the *k-th* column vector of A. For this reason, each of them is *linear combination* of the original variables in X.

The role of principal components analysis consists in finding those weights (the columns of A), also known as *loadings*, that lead the derived variables in Z to have desirable properties: maximum variance as possible and zero correlation between each other. It turns out that this is achieved in the case $\boldsymbol{A} = \boldsymbol{Q}$, where Q is the orthogonal matrix of eigenvectors generated by the spectral decomposition of $\boldsymbol{\Sigma}$.

To conclude, we provide a toy example of how (2.29) works in practice. This is only one of many schemes available to view matrix multiplication. However, we consider this one to provide a better intuition of the problem. The objective is to calculate the columns of the rectangular matrix Z, given X, – our "sample data"–, and the square matrix of weights A. Matrix Z has the same dimension as X. In fact, we can think as Z being the result of the transformation performed by A on X.

Suppose we have the following matrices:

$$\boldsymbol{X} = \begin{pmatrix} -4.67 & 8.37 \\ 1.33 & -4.67 \\ 3.33 & -3.67 \end{pmatrix} \tag{2.30}$$

$$\boldsymbol{A} = \begin{pmatrix} -0.49 & -0.87 \\ 0.87 & -0.49 \end{pmatrix} \tag{2.31}$$

Then we populate the columns of matrix Z as follows,

$$\boldsymbol{z_1} = \begin{pmatrix} 9.55 \\ -4.72 \\ -4.83 \end{pmatrix} = -0.49 \begin{pmatrix} -4.67 \\ 1.33 \\ 3.33 \end{pmatrix} + 0.87 \begin{pmatrix} 8.37 \\ -4.67 \\ -3.67 \end{pmatrix} \tag{2.32}$$

$$\boldsymbol{z_2} = \begin{pmatrix} -0.08 \\ 1.12 \\ -1.11 \end{pmatrix} = -0.87 \begin{pmatrix} -4.67 \\ 1.33 \\ 3.33 \end{pmatrix} - 0.49 \begin{pmatrix} 8.37 \\ -4.67 \\ -3.67 \end{pmatrix} \tag{2.33}$$

In the next section, principal component analysis will be performed on the time series of the daily US yield curve. Since interest rates of different maturities show

high correlation we will be able to represent a good amount of variability of the system using only a few principal components.

Chapter 3

PCA in Market Risk Analysis

3.1 Introduction

Identifying the major sources of risk (risk factors) resulting from the movements of the yield curve is one of the principal concerns of financial institutions managing fixed-income portfolios. In fact, the yield curve, which has an impact on bond prices, exhibits a variety of movements which cannot be always traced back to standard *parallel shifts*. In other words, it is not always the case that interest rates of different maturities rise or decline by the same amount along the entire spectrum of maturities. As a result, managing interest rate risk can be regarded as a "multidimensional"problem which should be tackled having consideration of the multiple types of fluctuations in the term structure. In a few words, we might say that fixed-income portfolios are exposed to *yield curve risk*.

This problem is hardly achieved by traditional price volatility measures, such as *duration*, which account only for parallel shifts of the yield curve. Hence, it is in the interest of risk analysts to have in their toolkit more sophisticated models which might suggest, on the basis of past observations of the yield curve, what are the most common movements of the yields occurring over two consecutive points in time. As we will see later on, the fluctuations that occur more frequently are successfully identified by the eigenvectors resulting from the spectral decomposition of the sample covariance matrix of interest rates changes. This procedure is part of the theoretical framework known as *statistical factor modelling* in which the factors are estimated by means of principal component analysis.

The chapter is structured as follows. Section 3.2 introduces the concept of *interest rate risk* of fixed-income instruments from a broad perspective. This concept is specified subsequently in Section 3.3 which illustrates how changes in interest rates affect the price of coupon bonds, – a particular class of fixed-income instruments – in order to provide a grounded justification of the reasons behind the need of

"multidimensional"risk models.

Section 3.4 discusses the term structure of interest rates and its related properties. In particular, it looks at past daily realizations of US yield curve in order to identify two main stylized facts which can be recognized also in the term structure of other countries: the high correlation existing between absolute changes in yields of different maturities, and its most frequent movements, *level*, *slope*, and *curvature*, as classified originally by Litterman and Scheinkman (1991). Finally, Section 3.6 discusses the estimation procedure and properties of statistical factor models on interest rates data. In the next chapter the analysis performed on the interest rates will be used to build the profit and loss of a bond portfolio.

3.2 Interest Rate Risk

In order to justify the use of principal component analysis as a tool for managing interest rate risk, it seems valuable introducing first the concept of interest rate risk from a high level perspective, in order to have a conceptual framework of the problem that this technique tries to address. This section is adapted from Chapter 6.1 and 6.4 of Fabrizi (2016).

Using a first approximation, risk associated with fixed-income instruments can be defined as the uncertainty regarding the result of the investment made on such a security. More specifically, Fabrizzi outlines two notions of risk concerning bonds, on the basis of two distinct perspectives: the *risk ex-ante* and *risk ex-post*. The first refers to the *expected* result, the second to the *realized* result.

Risk ex-ante, is defined in terms of variability of potential results one *might* obtain from the investment, as a consequence of diverse random factors that can affect the investment itself, such as the solvency of the issuer or the dynamics of market interest rates. On the other hand, the risk ex-post refers to the variability of past performances precisely quantified, either by comparing the difference between the attained results and the expected ones, or by assessing the variability of the observed values around the mean value, during a certain time framework. In the latter case, it is necessary to employ statistical estimators, including the standard deviation or the variance.

Even if acting on the basis of the risk ex-ante would be desirable, the difficulties related to its rigorous measurements lead instead to assess risk ex-post using the statistical estimators mentioned above, or using measurements that *approximate* the risk itself, such as duration and convexity. Furthermore, one possible refinement to those measurements is represented by statistical factor models estimated by means of principal component analysis which can be regarded as a "multidimensional "risk measure, given that they account for variations of interest rates of different maturi-

ties.

Interest rate risk is not the only type of risk affecting fixed-income instruments. Instead, generally speaking, risk affecting bonds takes a variety of forms, including:

1. **Market risk**
    - market risk in broad sense
        - counterpart risk
        - regulatory risk
        - liquidity risk
        - monetary risk
        - country risk
    - market risk in strict sense
        - **interest rate risk**
        - currency risk
2. **Credit Risk**

Hence, it is worth stressing, that models and measures discussed in this dissertation are valid exclusively to handle *interest rate risk*. For instance, credit risk modelling would deserve a completely stand alone discussion, since it makes use of its own models.

Having said that, we define interest rate risk as the effects of variations in the term structure of interest rates (see, Section 3.4) either on the price of a financial instrument, and on the income we can get from the reinvestment of the intermediate coupons. Thus, assuming the ordinary circumstance in which an investor buys a coupon bond and disinvest from it before its maturity, interest rate risk occurs in two different manners:

1. **Volatility risk (or Price risk)**
2. **Reinvestment risk**

Volatility risk concerns the variations of bond prices to variations of interest rates. In general, prices and interest rates are related by an inverse relationship: when interest rates increase, risk has a negative impact, since bond prices decrease. On the other hand, when interest rates decline, prices rise. More specifically, the impact of volatility risk might be quantified by the instantaneous percentage change of the bond price caused by a variation of market interest rates, where these rates are considered to be the yields of similar bonds with the same maturity.

Conversely, reinvestment risk is concerned with variations of the conditions upon which it is possible to reinvest the intermediate coupons. It manifests itself in the opposite way as volatility risk does: when interest rates grow, the coupons, – if reinvested – might provide higher returns. Instead, when rates sink, a lower return from the coupons is realized.

Furthermore, interest rate risk, – in its two different forms – depends upon two more factors, one concerning the *type* and the *technical characteristics* of the bond itself; the other the behaviour adopted by the investors relative to the *holding period*. Specifically, the former considers the following characteristics: whether a bond has zero coupons, the particular kind of interest rates (fixed or variable), the entity of the coupons and their frequency and, finally, their length (maturity). Instead, the latter shows the subsequent relationship with interest rate risk: the shorter is the investment horizon, the greater is volatility risk and the lesser is the reinvestment risk. The inverse relationships are also true, namely the longer the investment horizon, the bigger the reinvestment risk, and the smaller is price risk.

In synthesis, the effects of interest rates changes have opposite impact on volatility risk and reinvestment risk. For this reason, investors, choosing the appropriate investment allocation, might have the opportunity to balance the divergent manifestation of the two components in order to contain, or at most, offset interest rate risk in both aspects.

3.3 Fixed-income Securities

A bond is a financial instrument traded in the *fixed-income capital market* or *debt market*, which is populated by investors and issuers, typically companies or governments. Investors are willing to pay a price to the issuers in order to obtain a security whose value is known as *face value* (*or par value*). Holding such securities, guarantee a stream of future cash flows, called *coupon payments*, until *maturity*, when the capital originally invested is given back. The amount of the coupons is decided as a percentage of face value. This percentage is the *coupon rate* of the bond.

Bonds can be purchased at the time of issue on the primary market where the price is usually set at par or below par by the issuer. Once the bond is purchased on the primary market, it can be resold in the secondary one, where its price fluctuates freely around par value, according to the economic laws of supply and demand.

Although the coupons and the payback of the principal at maturity are guaranteed (unless the issuer is solvent), it is important to stress, that the *realized rate of return* of a bond is not; primarily due to *interest rate risk*. Hence, neither the coupon rate, nor the yield to maturity, which will be defined later in the section, are indicative of the performance of a bond. Basically, fluctuations of the market interest rates

affect the price of the bond, which in turn might affect its overall rate of return. For this reason, it is crucial for investors to quantify, at least approximately, the volatility of bond prices to variation of the interest rates with appropriate measures such as *duration* and *convexity*. However, we will see later on that these measures provide a partial representation of the interest risk of a bond. As a consequence, principal component analysis is implemented in order to provide a solution to the drawbacks of those standard measures.

"Because a bond's coupon and principal repayments all occur months or years in the future, the price an investor would be willing to pay for a claim to those payments depends on the value of dollars to be received in the future compares to dollars in hand today"(Bodie et al., 2014, p. 78). Therefore, in order to compare the amount of money at a different point in time, we need the concept of *present value* which is based on the assumption that a dollar today is more valuable than a dollar paid in one year. This is true, for the reason that you could deposit one dollar today in a bank account that secures interest and have more than a dollar in one year.

The present value of a sum of money in a future time t is computed by means of a *discounting factor*, which is determined in turn by interest rate R. The higher the interest rate, the lower a future amount of money is valued.

$$\text{discounting factor} = \frac{1}{(1+R)^t} \tag{3.1}$$

For more details regarding basic notions of financial mathematics see Stefani et al. (2011), and Fabozzi (2013) for extensive bond analysis and concepts.

For illustration purposes, we present the following example inspired by Bodie et al. (2014). Assume an investor buys a bond issued on the primary market, with the following characteristics:

Table 3.1: Coupon Bond Example

Features	
`Price`	\$850
`Face Value`	\$1000
`Coupon rate`	0.08
`Rate type`	Fixed
`Maturity`	30 years
`Coupon frequency`	Yearly
`Amortization`	No
`Options`	No

Since both the coupons and price are known, the investor can calculate the *yield to maturity* or *internal rate of return* (I.R.R.), to get an estimate *ex-ante* of

the *expected* return of the bond. From a mathematical point of view, the yield to maturity is defined as the interest rate that makes the present value of a bond's payments equal to its price at $t = 0$.

$$\begin{aligned} P &= \frac{C^{(1)}}{1+R} + \frac{C^{(2)}}{(1+R)^2} + \frac{C^{(3)}}{(1+R)^3} + \ldots + \frac{C^{(n)}}{(1+R)^n} + \frac{FV}{(1+R)^n} \\ &= \sum_{t=1}^{n} \frac{C^{(t)}}{(1+R)^t} + \frac{FV}{(1+R)^n} \end{aligned} \tag{3.2}$$

In equation 3.2, the C's are the yearly coupons paid each year starting from $t = 1$ until $t = n$. The last period includes both the payback of the capital FV, and the last coupon $C^{(n)}$. More specifically, in our example, we can set the following equation[1] using the specifics provided in Table 3.4:

$$\begin{aligned} \$850 &= \frac{\$80}{1+R} + \frac{\$80}{(1+R)^2} + \frac{\$80}{(1+R)^3} + \ldots + \frac{\$80}{(1+R)^{30}} + \frac{\$1000}{(1+R)^{30}} \\ &= \sum_{t=1}^{30} \frac{\$80}{(1+R)^t} + \frac{\$1000}{(1+R)^{30}} \end{aligned} \tag{3.3}$$

The resulting yield to maturity of the bond, calculated with Python script in Appendix B, is 9.5%.

As mentioned above, the yield to maturity is a measure of *potential* profitability of the investment on the bond. In fact, it can be interpreted as the evaluation *ex-ante* of the average return of the bond. However, the yield calculated at the time of purchase according to (3.3) is only a *promised* yield which will be verified only under the following assumptions: (1) the intermediate coupons are reinvested at the annual yield to maturity and (2) the holding-period coincides with the maturity of the bond. As a consequence, if the above conditions do not hold the *realized* yield to maturity would be different.

In addition, the yield to maturity coincides with the rate of return that – if realized – would make the investment "fair". In other words, ceteris paribus, it is the proper return that should be granted to the investor as a reward for having renounced to his capital for the entire life time of the bond.

Table 3.2 shows the decomposition of the computation performed in equation 3.3. The third column, "Present Value at t = 0", lists the present value at time $t = 0$ of each future coupon payment. Notice that in year 30 the present value is given by the sum of both the final coupon and the amount of capital invested. The sum of the third column gives back \$850 which is the bond price. Thus, the principle of

Table 3.2: Life time of coupon bond: P. $850, F.V. $1000, Mat. 30 years, 8% C.R.

Year	**Coupon Value ($)**	**PV at t=0 ($)**
1	80.00	73.04
2	80.00	66.69
3	80.00	60.88
4	80.00	55.59
5	80.00	50.75
6	80.00	46.33
7	80.00	42.30
8	80.00	38.62
9	80.00	35.26
10	80.00	32.19
11	80.00	29.40
12	80.00	26.84
13	80.00	24.50
14	80.00	22.37
15	80.00	20.42
16	80.00	18.65
17	80.00	17.02
18	80.00	15.54
19	80.00	14.19
20	80.00	12.96
21	80.00	11.83
22	80.00	10.80
23	80.00	9.86
24	80.00	9.00
25	80.00	8.23
26	80.00	7.50
27	80.00	6.85
28	80.00	6.25
29	80.00	5.71
30	1080.00	70.38

time value of money is respected.

Equation 3.3 explained how to compute the yield to maturity given the trading price and the promised cash flows. However, it is interesting to see it from a different

[1]Equation 3.3 has degree n, hence it cannot be solved analytically. As a consequence, we rely on an numerical approximation in order to solve it; see Appendix B where is presented a Python class object to search for the yield of a bond provided its technical features. The numerical algorithm implemented is the "bisection search". For details on simple numerical methods implemented in Python, see Guttag (2017). As stated in Choudhry (2004, p.8), "the process [of finding the yield-to-maturity] involves estimating a value for the yield and calculating the price associated with estimated yield. If the calculated price is higher than the price of the bond at the time, the yield estimate is lower than the actual yield, and so it must be adjusted until it converges to the level that corresponds with the bond price".

point of view, where the unknown becomes P, rather than R. In this way, the equation sees bond pricing as the process of computing the present value of the promised coupons:

$$\begin{aligned} P &= \frac{\$80}{1+R} + \frac{\$80}{(1+R)^2} + \ldots + \frac{\$80}{(1+R)^{30}} + \frac{\$1000}{(1+R)^{30}} \\ &= \sum_{t=1}^{30} \frac{\$80}{(1+R)^t} + \frac{\$1000}{(1+R)^{30}} \end{aligned} \tag{3.4}$$

From this perspective, equation 3.4 makes clear one fundamental property of bonds: *trading prices move in the opposite direction with yield*. The higher the yield the lower will be the trading price. The mathematical intuition of the relationship

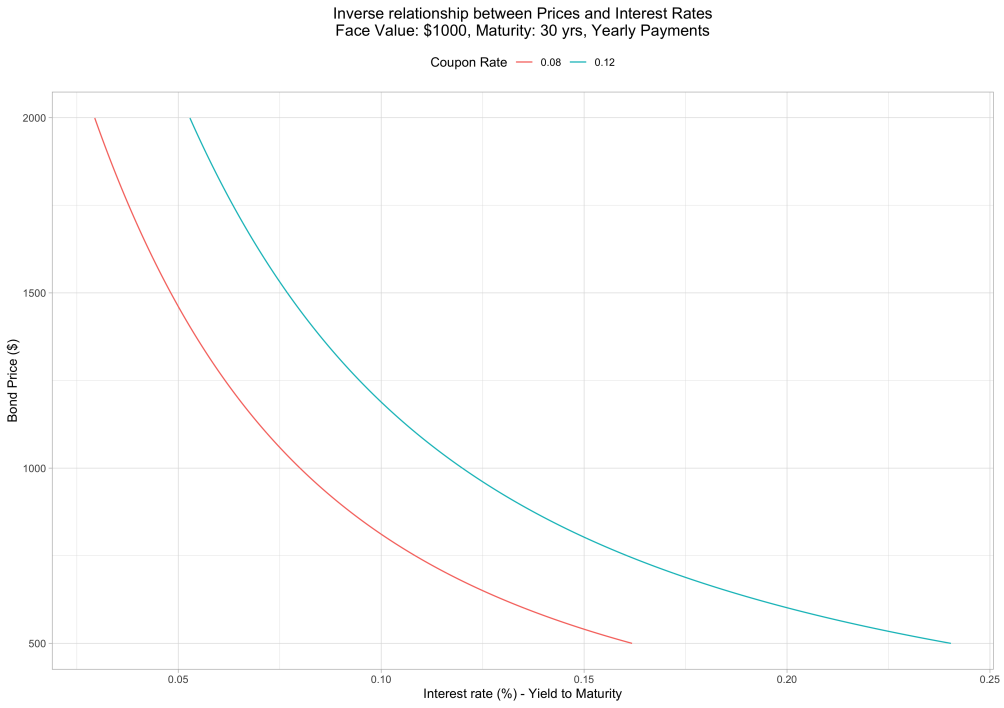


Figure 3.1: Inverse relationship between Price and Yield to maturity

underlying bond prices and yield to maturity is straightforward, in fact, if we let the interest rate increase, each coupon will be discounted more heavily, hence, the final price will be lower.

Instead, the economic intuition is more subtle: since the coupon rate is fixed, when market interest rates rise compared to the coupon rate, "the coupon payments alone will not provide investors as high return as they could earn elsewhere in the market. To receive a competitive return on such investment, investors also need some price appreciation on their bonds. The bonds, therefore, must sell below par value to provide a built in capital gain on the investment"Bodie et al. (2014, p.463).

To make this statement clearer, we should recall that the source of income of

a bond can be decomposed into three components which are summarized by the yield: (1) total coupon payments, (2) interest-on-interest, (3) capital gain. The composition of the three sources of income for the bond is illustrated in Table 3.3.

Table 3.3: Source of Income decomposition of bond: P. \$850, F.V. \$1000, Mat. 30 year, 8% C.R.

Total coupon payment	**Interest-on-interest**	**Capital gain**
\$2400	\$9640.46	\$150

The first component is given by the sum of the periodic coupons received during the lifetime of the bond, whereas the second is the interest generate from reinvestment of the periodic cash flows. Finally, the capital gain (or loss) is the difference between the face value and the trading price. Interest-on-interest has been computed with formula[2] (4.4) which assumes the annual reinvestment rate r to coincide with the yield to maturity 9.5%.

$$\text{interest on interest} = C\left[\frac{(1+r)^n - 1}{r}\right] - nC \tag{3.5}$$

We notice that interest-on-interest is a substantial part of the source of income of a bond. However, it is important to stress that it is only a potential gain as the reinvestment rate will hardly coincides with the yield-to-maturity for the entire holding period. For this reason, the yield to maturity and interest-on-interest, are both *expected* quantities evaluated *ex-ante* that will hardly be verified *ex-post* due to two main reason: (1) the investor might want to sell the security before its maturity at current market interest rates, and (2) market interest rates are not constant during the life of the bond. In other words, in order to achieve the promised return of 9.5% at the time of purchase of the instrument, the investor should produce an interest on interest of \$9640.46 by reinvesting the coupons. On the other hand, if the assumptions hold, then the investor expect to receive at the end of the 30 years the following amounts: the payback on the capital (\$850), the interest-on-interest (\$9640.46), the capital gain (\$150), and the coupons payment (\$2400).

Still, even with its flaws, the yield-to-maturity is a measure of profitability which is useful for investors in order to evaluate what rate of return they *might* earn on average by holding the bond until maturity compared to other instruments.

At this point, giving a concrete example of how reinvestment risk and volatility risk occurs is much easier. First of all, reinvestment risk is immediately appreciated by assuming a lower interest rate on (4.4) which will immediately affect the interest on interest component and, in turn, the realized yield. On the other hand, volatility

[2]Interest on interest is computed as it were an annuity, see Chapter 2 of Fabozzi (2013)

risk can be illustrated with a simple example. Suppose we are considering buying at the time of issue the bond illustrated in Table 3.4. The promised yield is 9.5%. However, immediately after the purchase, market interest rates rise by 200 basis point affecting the bond yield which consequently rise to 11.5%. Then, the price of the bond, calculated with (3.4), becomes \$706. Thus, the relative percentage change of the price of the bond is $\frac{\$706-\$850}{\$850} \approx -0.17\%$. This result can be appreciated qualitatively also by looking at Figure 3.1 which illustrates the inverse relationship between yield and bond prices. Furthermore, it shows that volatility risk is not symmetric, due to convexity. In fact, assuming an equal but opposite variation of the yield would have led to a price of \$1060 with a relative percentage increase of 24%. This property is well known by investors, who might be willing to pay higher price for a bond with higher convexity, ceteris paribus.

More generally, in secondary markets, prices of fixed-income securities, such as bonds, are determined by market interest rates which are in turn the result of macroeconomic factors responsible of shifting the supply and demand of securities. Mishkin and Eakins (2018) provides a basic framework to understand those determinants. In particular, macro factors altering the demand of assets are: wealth, risk, expected return and liquidity. On the other hand, supply of assets is influenced by: expected profitability of business opportunities, expected inflation and government budget. However, the details of this particular topic are outside of the scope of the current treatment.

To conclude, the following table summarizes three possible scenarios involving bonds trading. Bonds are said to trade *at discount* when the selling price equals the face value. On the other hand, bond trades *at premium* in the opposite case. Finally, when selling price equals the face value, the bond is sold *at par*.

Bond selling at:	**Relationship**
Par	Coupon rate = yield to maturity
Discount	Coupon rate < yield to maturity
Premium	Coupon rate > yield to maturity

3.4 The Yield Curve

The yield curve might seem an easy topic at first sight, however a whole field of research is devoted in estimating and modelling it, as well as providing economic theories[3] pretending to explain the reasons behind its different shapes (see Choudhry, 2004). In Figure 3.2a, we illustrate some stylized patterns extrapolated from the US

[3]We mention the *expectation hypothesis* and *liquidity preference* theory.

spot term structure which will be examined more deeply in the following sections. With the term *spot* we refer to the interest rate that is valid from now, time zero, until time t in the future.

We shall introduce the concept of the *spot yield curve* because the assumption introduced in Section 3.3 of a flat rate used to discount cash flows at different maturity is not realistic. In reality, one can observe patterns in the required yields as a function of the maturity of the bond. Usually, the higher the maturity of a bond, the higher is the expected return reflected by the yield (see Figure 3.2a). However, in certain circumstances this might not be case as illustrated in Figure 3.2b or Figure 3.2c.

For us, it will be relevant not much the particular shape assumed by the yield curve on a certain day, instead, it will be of interest understanding its absolute *change* between two consecutive points in time. The notion of *yield curve change* is discussed extensively by Jones (1991) who discusses how the knowledge of the most common kind of movements having occurred historically proves useful not only for interest risk analysis, but also to figurate out portfolio allocations that might capitalize on such movements.

3.5 Duration and PV01

This section is devoted to briefly introduce measures of bond price volatility, such as *duration* and *price value of a basis point* (PV01) which will turn useful in the next chapter[4].

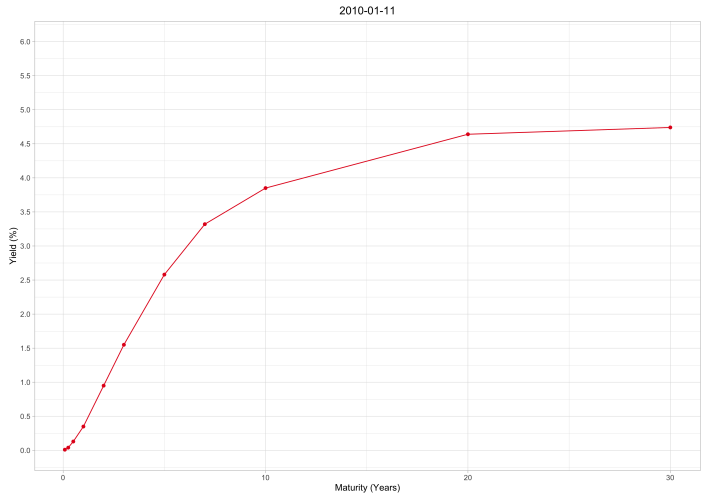
Equation 3.6 is referred to as *Maculay duration* and it can be recovered from the derivative of (3.2), with respect to the yield, as shown in (3.7). As stated in (Alexander, 2008b, p. 23), "the Maculay duration represents the future point in time where the loss from the the fall in the bond price is just offset by the income gained from an increased interest on the coupon payments".

$$\text{Maculay duration} = \frac{\sum_{t=1}^{n} \frac{tC^{(t)}}{(1+y)^t} + \frac{nFV}{(1+y)^n}}{P} \tag{3.6}$$

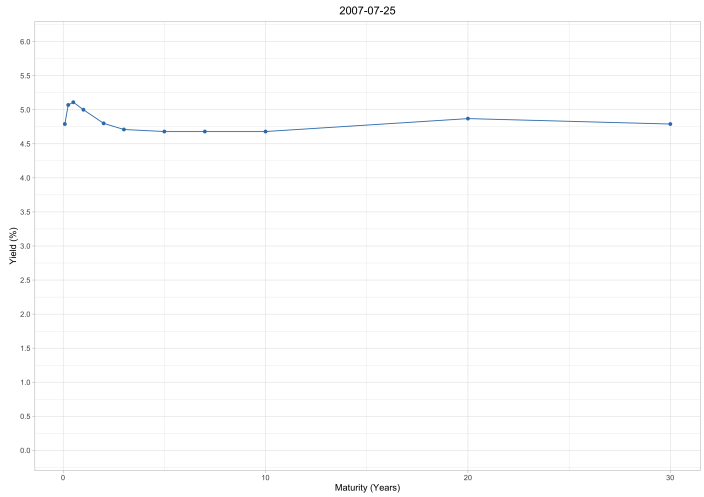
$$\frac{dP}{dy}\frac{1}{P} = -\frac{1}{1+y} \times \text{Maculay duration} \tag{3.7}$$

Further, Maculay duration can be used to identify *modified duration* which is informative of the *approximate* percentage price change for a given change in the

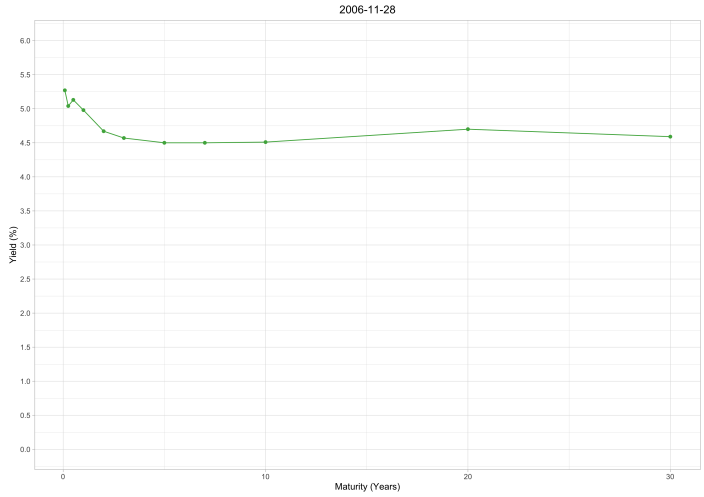
[4]The formulas regarding duration are drawn from Fabozzi (2013), whereas those on PV01 are taken from Alexander (2008b).



(a) Rising Yield Curve



(b) Flat Yield Curve



(c) Inverted Yield Curve

Figure 3.2: US Treasury Yield Curves
Source: US Treausury

yield, usually measured in 100 basis points (0.01 or 1%).

$$\text{modified duration} = \frac{\text{Maculay duration}}{1+y} \tag{3.8}$$

$$\frac{dP}{P} = -\text{modified duration} \times dy \tag{3.9}$$

As we can see the drawback of volatility price measures based on duration consists in the fact that they consider only *parallel* shifts of the spot yield curve. As stated previously, principal component analysis provides a more effective instrument for bond portfolio immunization, given that provides coverage against to the most common movements in the yield curve, using past data.

Conversely, the present value of a basis point move (PV01) measures the absolute variation in the present value of a series of cash flows when the yield curve decreases by 0.01%. Suppose that a sequence of cash flows with their associated spot rates are denoted as:

$$\boldsymbol{c}^{\boldsymbol{T}} = (C^{(1)}, C^{(2)}, \ldots, C^{(n)}) \tag{3.10}$$

$$\boldsymbol{r}^{\boldsymbol{T}} = (R^{(1)}, R^{(2)}, \ldots, R^{(n)}) \tag{3.11}$$

Let be the present value of the cash flows to be $PV(\boldsymbol{c}, \boldsymbol{r})$, and $\boldsymbol{r}^{-} = (R^{(1)} - 0.01, R^{(2)} - 0.01, \ldots, R^{(n)} - 0.01)$. Then we denote with:

$$PV01(\boldsymbol{c}, \boldsymbol{r}) = PV(\boldsymbol{c}, \boldsymbol{r}^{-}) - PV(\boldsymbol{c}, \boldsymbol{r}) \tag{3.12}$$

the sensitivity of the cash flows present value, if each of the associated rates were to decrease by one basis point. Further, the PV01 of a generic cash flow $C^{(i)}$ is computed as:

$$PV01^{(i)} = C^{(i)} \times \delta 01^{(i)}, \tag{3.13}$$

and the total PV01 of the entire sequence as:

$$PV01 = \sum_{t=1}^{n} C^{(t)} \times \delta 01^{(t)} \tag{3.14}$$

where n is the number of cash flows.

Consider, as an example, a selected pool of cash flows from the bond portfolio which will be introduced later in Section 4.4.

Table 3.4: PV01 of sequence of cash flows.

Yrs to Mat	**C ($)**	**R**	**PV**	$\boldsymbol{R - 0.01\%}$	**PV**	**PV01**
1.47	18.687	0.46	18.5619	0.45	18.5646	0.0027
3.02	18.687	0.58	18.3616	0.57	18.3671	0.0055
5.60	18.687	0.66	18.0110	0.65	18.0210	0.0100
24.20	108.688	1.21	81.2337	1.20	81.4282	0.1945
			136.16		136.38	0.22

In this example, the present value of the sequence of cash flows will increase by an amount of \$0.22 if $\boldsymbol{r}$ were to decrease by 0.01% at each maturity.

3.6 PCA on the Interest Rates Term Structure

As we have seen in the previous section, fixed-income portfolios are exposed to interest rate risk. In particular, financial institutions are willing to quantify the exposure of bond portfolios to *unequal* fluctuations in interest rates of *different* maturities. However, standard measures of bond price volatility[5], such as duration and convexity, do not provide a good estimate of such changes. In fact, their explanation power is limited to: (1) variations of the required yield, disregarding the existence of interest rates of different maturities, and (2) *parallel* shifts of the yield. In other words, these measures rely on the simplifying assumption of a *flat* yield curve (Fabozzi, 2013).

For instance, "consider the situation of a U.S. government bond trader. The trader's portfolio is likely to consist of many bonds with different maturities. [Hence], there is an exposure to movements in the one-year rate, the two-year rate, the three-year rate, and so on. [...] He or she must be concerned with all the different ways in which the U.S. Treasury yield curve changes its shape through time"(Hull, 2018, p.185). As a result, bonds portfolio are sensitive to several sources of risk, potentially as many interest rates impact its value. For this reason, there is the need to reduce the number comprising the set of risk factors to a manageable one, usually three or four.

Hence, a statistical approach[6] requires to collect historical observations (see Fig. 3.3) of interest rates at different maturities to infer what the future changes

[5]Standard measures and models to manage interest rate risk such as the *Duration Gap* are discussed by Sironi and Resti, 2007. Particular attention is devoted to the asset-liabilities mismatch.

[6]The random process generating the behaviour charachterizing the yield curve is also modelled by means of stochastic differential equations. Those (advanced, – at least for who is writing) models are reviewed in Choudhry (2004). In contrast, the approach adopted here is statistical in nature. The difference between the two frameworks can be well appreciated in Redfern and McLean (2004).

might be on the basis of those observed in the past[7]. This would imply to estimate a model which we require to achieve the following objectives:

1. Identifying the fundamental yield curve movements.

2. Being able to replicate as faithfully as possible the covariation of the overall system of interest rates in a compact way.

The first task is accomplished by *principal component analysis*, whereas the second by a *linear factor model*[8], whose factors are selected among the PCs with higher explanatory power. Hence, those principal components represent a reduced set of "basis"that combined linearly with the eigenvectors are able to replicate with accuracy the past behaviour of the interest rates in a compact manner, and more importantly to identify those common movements.

3.6.1 Spectral analysis of the term structure

To illustrate how PCA can be used as a dimension reduction tool for interest risk analysis, we consider the multivariate time series of daily U.S. spot rates[9] for a period of 3.492 trading days, from February 02, 2006 to January 01, 2020 (see, Figure 3.3). The interest rates decline sharply after the 2008 as a consequence of the policies implemented by the FED in order to stimulate the economy in response to the financial crisis. More importantly for our purposes, we observe that the term structure is highly correlated, in particular, among rates of close maturity. The figures in Table 3.5 confirm this fact, since the correlation between yields is higher around the main diagonal, meaning that interest rates of similar maturity move closely. Conversely, the correlation is weaker between interest rates of different maturity. As a result, figures in Table 3.5 demonstrate that "treasury yields do not move around in a completely uncorrelated fashion. If they did, it would be impossible to analyze the interest rates risk of a bond portfolio in any meaningful

[7]Yet, this approach is subject to another kind of risk known as *historical bias*. It might not be the case that the past will repeat itself in the future. In trying to reduce the historical bias risk, one might plug into the model subjective information adopting a *bayesian* approach. *Model risk* should also be taken into account.

[8]*Multi-factor models* are extensively used in finance. Their aim is to explain the covariance structure of portfolios' asset returns, as a function of p underlying factors. In the literature, one can identify three types of factors models depending, on the estimation strategy adopted. A comparison between their explanatory power on U.S. equities is provided by Connor (1995). On the other hand, the estimation process of the three models, – *macroeconomic*, *fundamental* and *statistical* – is discussed in Embrechts et al. (2015) and, more extensively, in Tsay (2010). Generally speaking, both the macroeconomic and fundamental require the researcher to provide the observations on a factor, for example in the form of a economic indicator or a financial index, whereas the latter allow the analyst to estimate directly from the data the main factors that might drive the overall risk of the portfolio.

[9]https://www.treasury.gov/resource-center/data-chart-center/interest-rates/Pages/TextView.aspx?data=yield

Table 3.5: Correlation matrix of the absolute interest rates changes measured in Bps.

Mat.	1MO	3MO	6MO	1YR	2YR	3MO	5YR	7YR	10YR	20YR	30YR
1MO	1.00	0.67	0.50	0.46	0.25	0.21	0.16	0.14	0.11	0.09	0.09
3MO	0.67	1.00	0.76	0.62	0.34	0.30	0.24	0.20	0.16	0.13	0.12
6MO	0.50	0.76	1.00	0.82	0.51	0.48	0.39	0.34	0.30	0.24	0.23
1YR	0.46	0.62	0.82	1.00	0.74	0.70	0.60	0.53	0.47	0.40	0.38
2YR	0.25	0.34	0.51	0.74	1.00	0.94	0.87	0.79	0.72	0.61	0.58
3YR	0.21	0.30	0.48	0.70	0.94	1.00	0.94	0.88	0.82	0.72	0.68
5YR	0.16	0.24	0.39	0.60	0.87	0.94	1.00	0.97	0.93	0.84	0.80
7YR	0.14	0.20	0.34	0.53	0.79	0.88	0.97	1.00	0.97	0.91	0.87
10YR	0.11	0.16	0.30	0.47	0.72	0.82	0.93	0.97	1.00	0.96	0.93
20YR	0.09	0.13	0.24	0.40	0.61	0.72	0.84	0.91	0.96	1.00	0.98
30YR	0.09	0.12	0.23	0.38	0.58	0.68	0.80	0.87	0.93	0.98	1.00

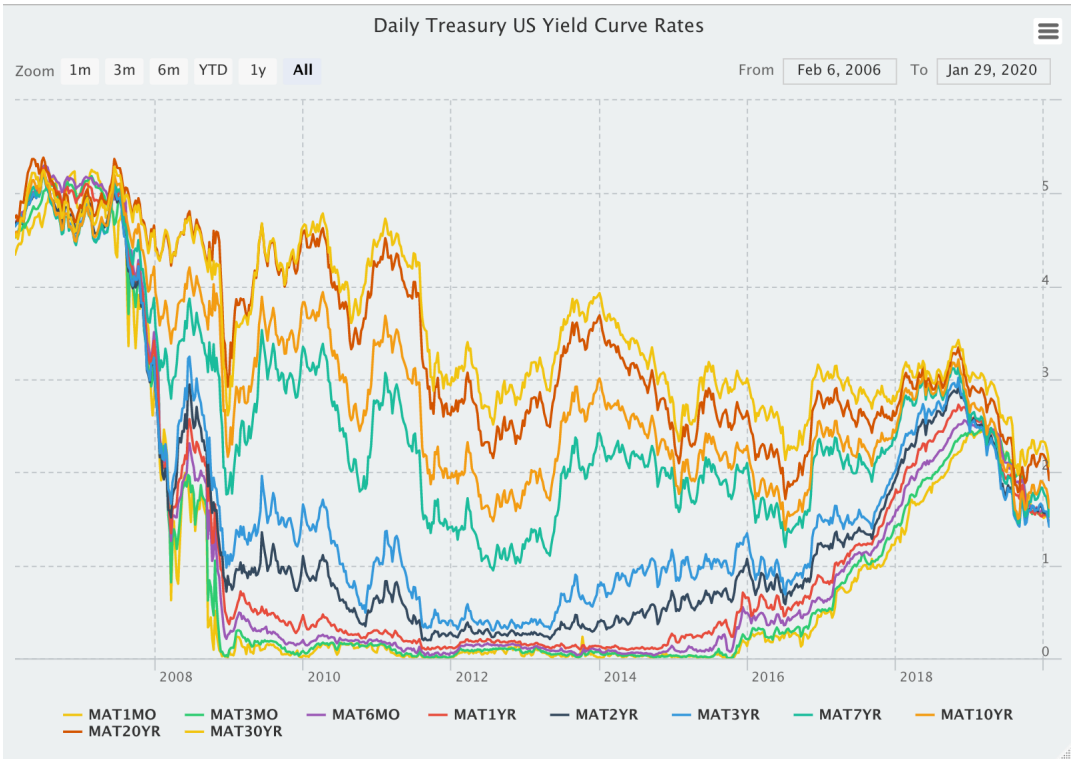


Figure 3.3: U.S. Interest Rates Term Structure.
Source: U.S. Treasury.

way; even the notion of portfolio duration would be meaningless"(Fabozzi, 2012, p. 797). Hence, to assess the yield curve risk affecting a fixed-income portfolio, it would be useful to understand, at least historically, what are the most common types of shifts that have occurred in the yield curve.

As stated previously, principal component analysis performed on the *interest rates changes* is capable of detecting them, in the form of principal components.

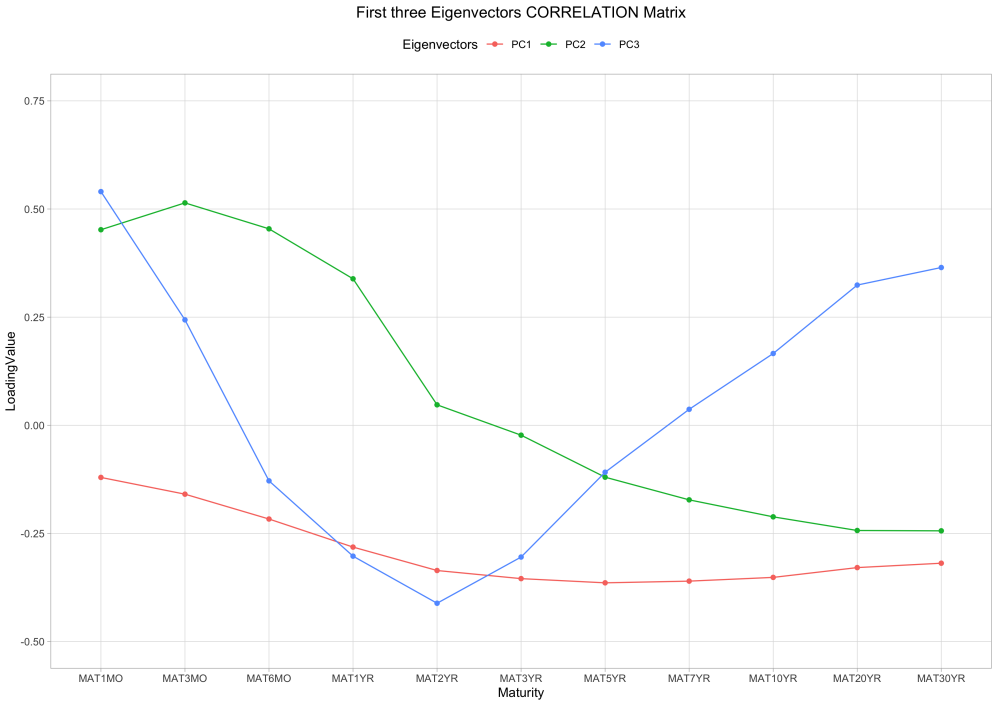
Usually[10], the first principal component records an almost *parallel shift* of the yield curve, the second one a change in the slope (*tilt*), and the third one a change located in the middle of the term structure (*curvature* or *convexity*). The first degree of intuition for this representation is provided by Figure 3.4a which illustrates the first three *eigenvectors* resulting from the spectral decomposition of the correlation matrix of interest rates changes. In red, the first eigenvector is approximately a parallel line since it takes similar values across the entire spectrum of maturities. For this reason, it captures parallel movements of the yield curve. Subsequently, the eigenvector in green is almost decreasing, hence it explains movements which are upward in nature on early maturities and downward on later ones. Ultimately, the third eigenvector in blue, is decreasing at the beginning and increasing at the end. Therefore, it describes inverted "bumps"of the yield curve (Alexander, 2008a)[11].

Table 3.6 shows the actual values of the eigenvectors, also known as *loadings*, whereas Table 3.7 illustrates the corresponding eigenvalues in decreasing order of explanatory power. Recalling the notions introduced in Section 2.6, we know that the correlation matrix is positive definite, hence the eigenvalues are all positive. In addition, according to formulas (2.27) and (2.28), we know that each principal component contributes to explain an amount of variance corresponding to its associated eigenvalue. For instance, `PC1` explains, by itself, about 62% of the total covariation between changes in the US interest rates. Instead, the first three PCs, considered jointly, explain about 91% of the total covariation in the system.

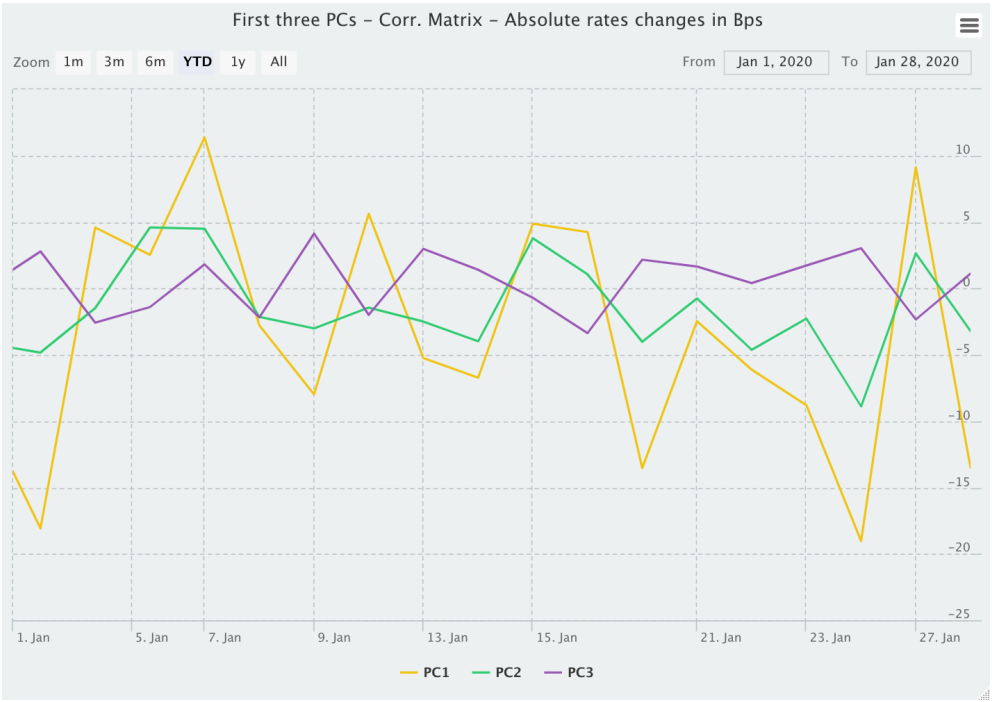
Once, the eigenvalue and eigenvector are found using spectral decomposition, the principal components are computed using equation (2.29). Figure 3.4 shows the first

[10]Fabozzi (2012, p. 1109) reports the academic studies which have applied PCA on the term structure of different countries. In particular: Robert Litterman and Jose Scheinkman, "Common Factors Affecting Bond Returns", Journal of Fixed Income (September 1991), pp. 54-61; Alfred Bühler and Heinz Zimmermann, "A Statistical Analysis of the Term Structure of Interest Rates in Switzerland and Germany", Journal of Fixed Income (December 1996), pp. 55-67; Joel R. Barber and Mark L. Copper, "Immunization Using Principal Component Analysis", Journal of Portfolio Management (Fall 1996), pp. 99-105; Rita L. D'Ecclesia and Stavros Zenios, "Risk Factor Analysis and Portfolio Immunization in the Italian Bond Market", Journal of Fixed Income (September 1994), pp. 51-58; Bennett W. Golub and Leo M. Tilman, "Measuring Yield Curve Risk Using Principal Components Analysis, Value at Risk, and Key Rate Durations", Journal of Portfolio Management (Summer 1997), pp. 72-84; Lionel Martellini and Philippe Priaulet, Fixed-Income Securities: Dynamic Methods for Interest Rate Risk Pricing and Hedging (Chichester, England: Wiley, 2000); Sandrine Lardic, Philippe Priaulet, and Stephane Priaulet, "PCA of Yield Curve Dynamics: Questions of Methodologies", Journal of Bond Trading and Management (April 2003), pp. 327-349. Fabozzi (2012) reports also the number of factors (PCs) employed and the percentage of variance explained. In all of the cases it is greater than 80%.

[11]It is worth noting that, within this case, the eigenvectors, and consequently the principal components, have straightforward interpretation since the system of interest rates is *ordered* and comprised of variables of the same kind. Differently, the principal components would be the result of linear combinations of variables having independent meaning. Therefore, they would be still valid from a mathematical point of view, yet their interpretation would be more obscure. These isses are discussed by Rao (1964) in a very technical paper, and more accessibly by Jolliffe (2002) in Chapter 11.



(a) Eigenvectors



(b) PCs

Figure 3.4: First three eigenvectors and principal components resulting from the correlation matrix.

Table 3.6: Eigenvector loadings of the correlation matrix of interest rates changes measured in Bps.

Maturity	$\boldsymbol{w_1}$	$\boldsymbol{w_2}$	$\boldsymbol{w_3}$	$\boldsymbol{w_4}$	$\boldsymbol{w_5}$
MAT1MO	−0.120	0.452	0.540	0.622	0.294
MAT3MO	−0.159	0.514	0.244	−0.216	−0.720
MAT6MO	−0.217	0.454	−0.128	−0.497	0.216
MAT1YR	−0.281	0.339	−0.302	−0.142	0.482
MAT2YR	−0.336	0.047	−0.411	0.318	−0.086
MAT3YR	−0.354	−0.023	−0.305	0.258	−0.152
MAT5YR	−0.364	−0.120	−0.108	0.154	−0.163
MAT7YR	−0.360	−0.172	0.037	0.052	−0.121
MAT10YR	−0.352	−0.212	0.166	−0.059	−0.032
MAT20YR	−0.329	−0.243	0.324	−0.203	0.123
MAT30YR	−0.319	−0.244	0.365	−0.247	0.166

Table 3.7: Eigenvalues of the correlation matrix of interest rates changes measured in Bps.

Component	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
λ	6.89	2.38	0.74	0.45	0.23	0.12	0.09
Var. Exp. (%)	62.62	21.60	6.76	4.12	2.13	1.12	0.80
Cumulative (%)	62.62	84.22	90.98	95.10	97.23	98.35	99.15

three PCs which are the result of the weighted sum of the interest rates changes with the loadings of $\boldsymbol{w_1}$, $\boldsymbol{w_2}$ and $\boldsymbol{w_3}$ as weights (see Table 3.6). Therefore, we might say informally that PC1 "incorporates"the "information"of the first eigenvector which captures a parallel shift of the yield. The same applies for PC2 and PC3, associating the corresponding eigenvector.

Furthermore, it is useful to recall that the PCs are *orthogonal*, thus their correlation is zero. This property is particularly useful in market risk models because it allows to handle *uncorrelated* risk factors.

In conclusion: "the first principal component captures a *common trend* [...] [in] interest rates [changes]. That is, if the first principal component changes at a time when the other components are fixed, then [the interest rates] all move by roughly the same amount. For this reason we often called the first component the *trend component*. [...] Then the second principal component usually captures a change in slope of the term structure. [...] For this reason we often called the third component the *curvature* or *convexity* component"(Alexander, 2008a, p. 51-52).

To clarify the role of the principal components, it seems useful to introduce a toy example which assumes a multivariate time series of interest rates changes with pattern illustrated in Table 3.8. Assume, moreover, it repeats itself cyclically for

n days. Yield curve changes as such presents only perfectly upward or downward parallel shifts. Then, we shall ask ourselves how the correlation matrix, eigenvectors and eigenvalues would be like in this particular case.

According to the exposition made above, we should expect, firstly, the correlation matrix to be $(n \times n)$ filled by ones in each entry, since there is perfect correlation between yields of different maturities; secondly, the loadings of the first eigenvector $\boldsymbol{w_1}$ to be constant across the entire spectrum of maturities, and, eventually, the eigenvalues to be all zeros, except for the one associated to $\boldsymbol{w_1}$.

Using R we obtain the following results: $\boldsymbol{w_1^T} = (0.302, 0.302, \ldots, 0.302)$ and $\boldsymbol{\lambda}^T = (11, 0, \ldots, 0)$. On the other hand, the principal components of this toy example are illustrated in Figure 3.5. As we can see, `PC1` in yellow shows a positive trend in correspondence of positive changes in the interest rates, whereas it presents a negative trend when the changes are negative. Essentially, in this artificial example we isolate the effect of the first eigenvector on the first principal component by zeroing the effect of the others.

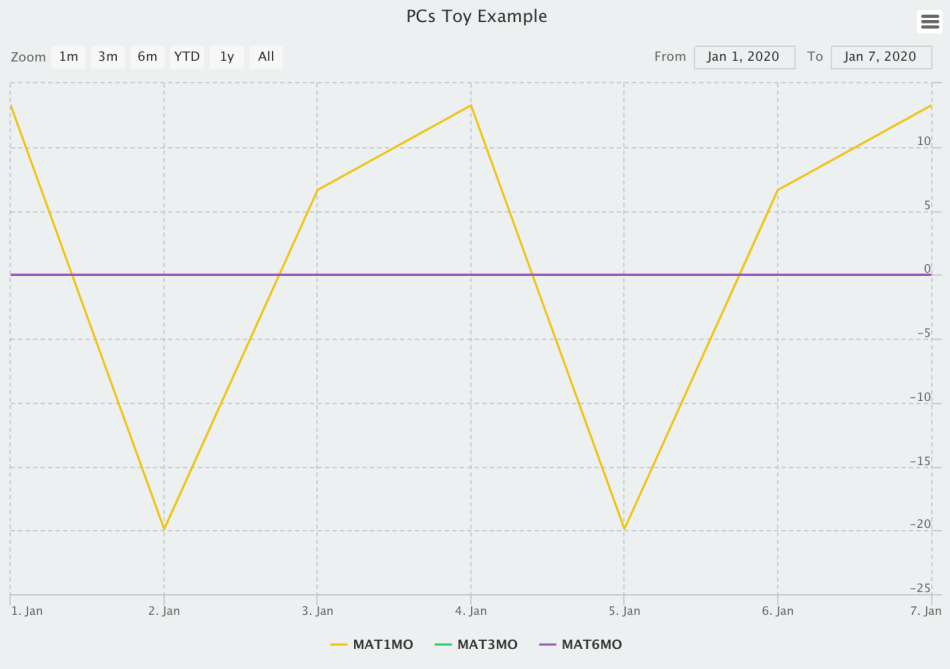


Figure 3.5: First three PCs of the correlation matrix in the case of only parallel shifts.

3.6.2 Linear Factor Model

Factor models are conceptually independent from principal component analysis and they constitute an independent field of research in statistics. Nevertheless, PCA provides a feasible estimation strategy for their "factors". This should not be a surprise, given that "analysis of principal components are more of a means to an

Table 3.8: Interest rates changes of a toy example.

Mat.	1MO	3MO	6MO	1YR	2YR	3YR	5YR	7YR	10YR	20YR	30YR
2020/01/01	4	4	4	4	4	4	4	4	4	4	4
2020/01/02	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6
2020/01/03	2	2	2	2	2	2	2	2	2	2	2
2020/01/04	4	4	4	4	4	4	4	4	4	4	4
2020/01/05	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6
2020/01/06	2	2	2	2	2	2	2	2	2	2	2
2020/01/07	4	4	4	4	4	4	4	4	4	4	4

end rather than an end in themselves, because they frequently serve as intermediate steps in much larger investigations"(Johnson and Wichern, 2014, p.430).

We require from a linear factor model[12] to approximate with accuracy the covariation of the observed interest rates changes, using a small set of risk factors.

In particular, we approximate the random vector of interest rates changes

$$(\Delta R^{(m1)}, \Delta R^{(m3)}, \ldots, \Delta R^{(y30)})$$

with three risk factors given by the first three principal components $\boldsymbol{z_1}, \boldsymbol{z_2}, \boldsymbol{z_3}$, and factor weights the corresponding eigenvectors $\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}$. We call *principal component representation* at time t the daily interest rates changes provided by the following linear factor model:

$$\begin{aligned}
\Delta R_t^{(m1)} &= r_t^{(m1)} - r_{t-1}^{(m1)} \approx w_1^{(m1)} z_t^{(1)} + w_2^{(m1)} z_t^{(2)} + w_3^{(m1)} z_t^{(3)} \\
\Delta R_t^{(m3)} &= r_t^{(m3)} - r_{t-1}^{(m3)} \approx w_1^{(m3)} z_t^{(1)} + w_2^{(m3)} z_t^{(2)} + w_3^{(m3)} z_t^{(3)} \\
&\vdots \\
\Delta R_t^{(y30)} &= r_t^{(y30)} - r_{t-1}^{(y30)} \approx w_1^{(y30)} z_t^{(1)} + w_2^{(y30)} z_t^{(2)} + w_3^{(y30)} z_t^{(3)}
\end{aligned}$$

In matrix notation the principal component representation is as follows:

$$\underset{(n\times 11)}{\boldsymbol{\Delta R}} \approx \underset{(n\times 3)(3\times 11)}{\boldsymbol{Z}\ \boldsymbol{W^T}} \tag{3.15}$$

$$\begin{pmatrix} | & & | & & | \\ \boldsymbol{\Delta R_{m1}} & \ldots & \boldsymbol{\Delta R_{y3}} & \ldots & \boldsymbol{\Delta R_{y30}} \\ | & & | & & | \end{pmatrix} = \begin{pmatrix} | & | & | \\ \boldsymbol{z_1} & \boldsymbol{z_2} & \boldsymbol{z_3} \\ | & | & | \end{pmatrix} \begin{pmatrix} — & \boldsymbol{w_1^T} & — \\ — & \boldsymbol{w_2^T} & — \\ — & \boldsymbol{w_3^T} & — \end{pmatrix}$$

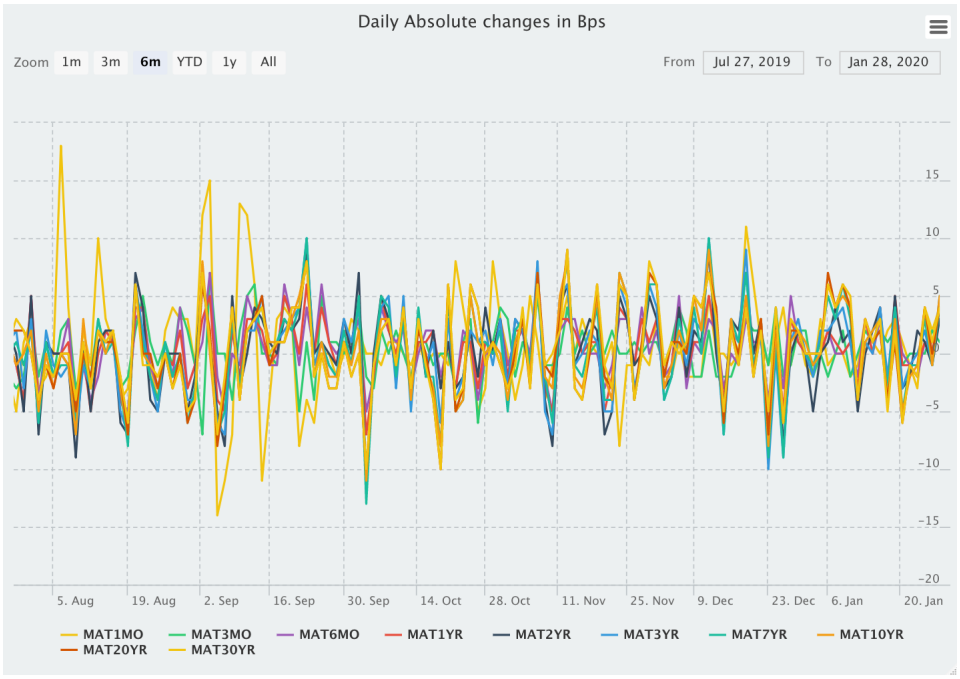
We shall see matrix multiplication of equation 3.15 as we did in Chapter 2: each

[12]The general formulation would be $\boldsymbol{R} = \boldsymbol{XW} + \boldsymbol{\Psi}$

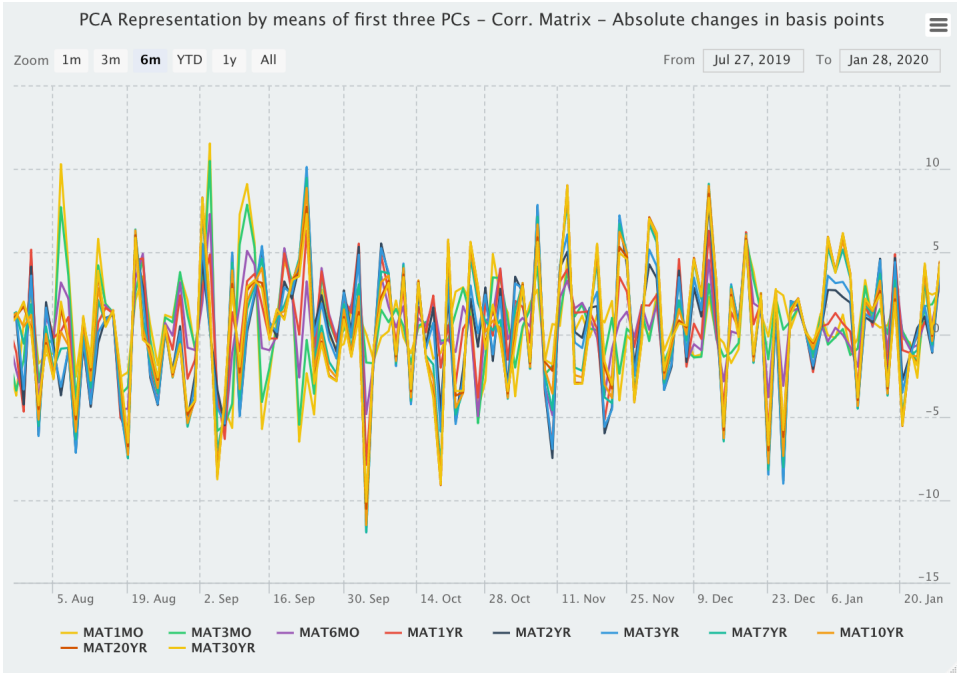
m-th column of $\boldsymbol{\Delta R}$ is computed by "weighting" *all* the three PCs in $\boldsymbol{Z}$, with the coefficients of the *m-th column* of $\boldsymbol{W^T}$.

The linear factor model is able to explain about 90% of the total covariation of the interest rates changes given that we have used only the first three components. A better approximation can be achieved adding more PCs. However, the model demonstrates the power of principal component analysis, since we are capable of explaining almost the entirety of the variance in the system exploiting only three components, instead of the original eleven interest rates. This model will be employed in the next chapter to describe the *profit and loss* of a portfolio composed of U.S. Government bonds.

Figure 3.6a shows the actual interest rates changes, whereas Figure 3.6 the principal component approximation. As we can see the linear factor model estimated by means of principal component makes a good job in replicating the overall covariation of the system.



(a) U.S. Daily Treasury Interest Rates Changes



(b) PCA representation of Daily Interest Rates Changes - Cov. Mat.

Figure 3.6: Comparison between actual interest rates dynamics and "approximated"PCA representation by means of first three PCs.

Chapter 4

Empirical Analysis

4.1 Introduction

The following section accomplishes three tasks. Firstly, we want to investigate the behaviour of the eigenvectors throughout successive equal time windows, in order to see whether the usual structure of parallel shift, tilt and curvature is maintained on shorter time periods. Secondly, we extend this exercise on a larger scale adopting a resampling technique known as *bootstrap*, in order to approximate the theoretical distribution of the eigenvalues and eigenvectors. Finally, we apply the principal component representation obtained in the last chapter, to derive the profit and loss of a fixed-income portfolio employed in traditional risk models.

4.2 PCA on different time windows

Figures 4.1 and 4.2 should be compared with Figure 3.4a. In particular, the following set of illustrations show the eigenvectors obtained from the application of the spectral decomposition on the covariance matrix on a rolling basis, along six consecutive time windows made of 582 daily observations each. The objective consists in investigating the stability of the loadings structure on shorter time frames. What emerges is that the distinctive structure made clear in Figure 3.4a, and originally identified by Litterman and Scheinkman (1991), breaks when principal component analysis is performed on shorter time windows. Unless stated otherwise in this section the principal components are computed on the covariance matrix, rather than the correlation one.

Conversely, we notice that, on a shorter time frame, the first eigenvector assumes an alternative typical shape which is persistent in the first four time window. In particular, it is almost flat for shorter maturities, increasing for intermediate ones, and slightly decreasing for longer maturities. Recalling, the inverse relationship

Table 4.1: Cumulative Variance Explained (%) by the first three components in each time window.

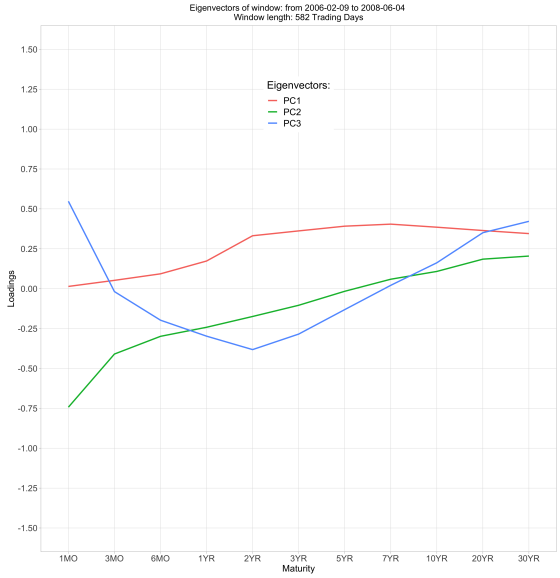
Window	**1**	**2**	**3**	**4**	**5**	**6**
Cumulative (%)	65.50	66.53	70.55	72.22	67.64	77.60

which underlies bond prices and yields introduced in Section 3.3, this might mean that, between two consecutive days, there is a higher probability that prices of *zero coupon* bonds with intermediate maturity decline in price, compared to those with a shorter maturity. In this respect, bonds with shorter maturities are in higher demand compared to those with a halfway maturity. In addition, we notice that the eigenvector is slightly decreasing on its tail. The reason for such phenomenon might be traced back to the fact that long term bonds are highly required among institutional investors, thus their demand exceeds the supply, having the effect of lowering their prices (Choudhry, 2004). Instead, more recently starting from Figure 4.2b, this paradigm seems to be inverted as the eigenvector takes higher values on earlier maturities than longer ones.

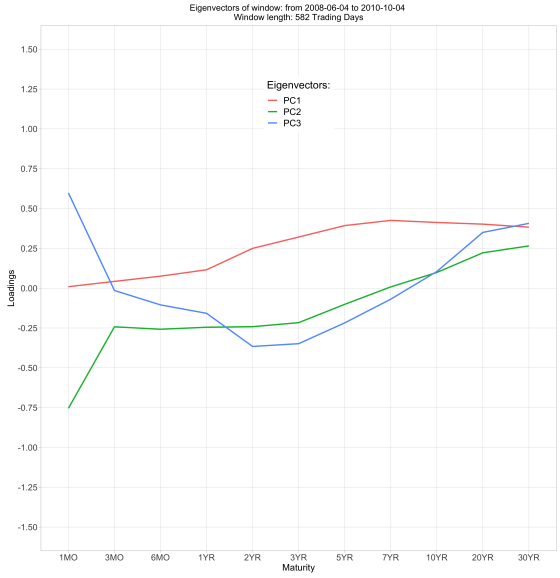
The second eigenvector has a higher variability compared to the first one. Nevertheless, it is still possible to identify at least two typical patterns with some degree of persistence. Firstly, it is increasing in Figure 4.1a and 4.1b. Secondly, it shows a bump either in Figure 4.1c and 4.2a.

Besides the first and the second, the third eigenvector exhibits the typical curvature form in four out of six of the different time frames.

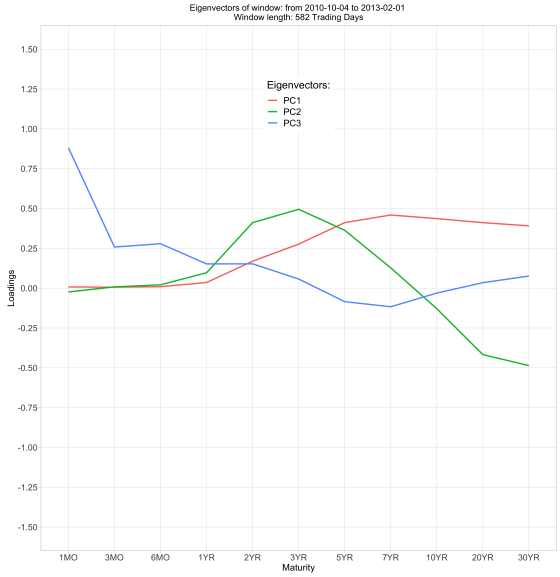
We can also evaluate the effect of considering a shorter time frame on the amount of variance explained by the first three components in each time window. The figures in Table 4.1 are not satisfactory as those obtained on the entire set of observations (see, Table 3.7). Thus, we should conclude that in order to achieve higher explanatory power we should consider longer time frames.



(a) Window 1

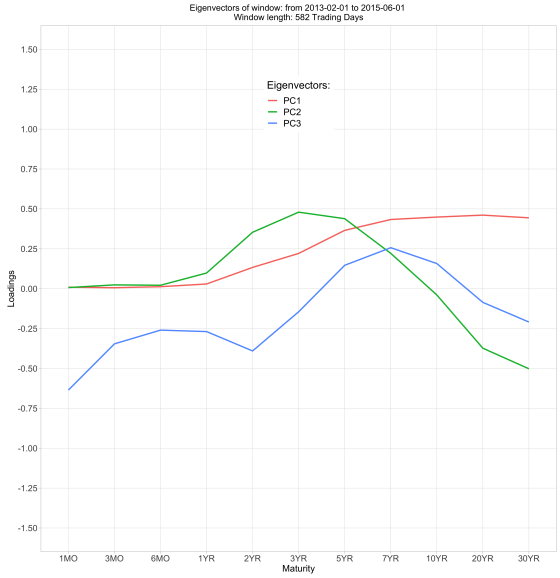


(b) Window 2



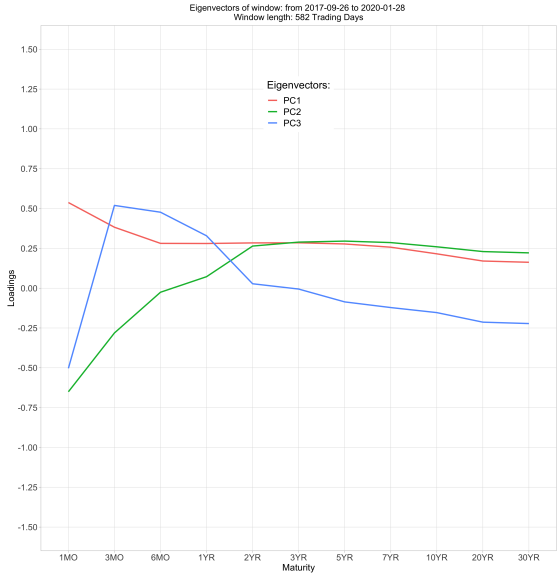
(c) Window 3

Figure 4.1: Eigenvectors of the US daily spot rate covariance matrix on different time windows (1)



(a) Window 4

(b) Window 5



(c) Window 6

Figure 4.2: Eigenvectors of the US daily spot rate covariance matrix on different times windows (2)

4.3 Bootstrap: Eigenvalues and Eigenvectors

In this section, we apply *bootstrap* to approximate the theoretical distribution of the eigenvalues and eigenvector loadings. In other words, we would like to have a *probabilistic* representation of the results obtained in Chapter 4. However, an important remark is needed from the very first. The following figures should be interpreted taking in consideration that each bootstrap sample is truly a *random sample*[1], whereas the figures obtained in the previous section are affected by either the *autocorrelation* existing within a single interest rate and the *cross-correlation* subsisting among them. In particular, the proportion of variance explained by the first three components obtained with the bootstrap will substantially outperform those obtained considering consecutive time frames, even if each bootstrap sample will have the same number of observations of the time windows presented in the previous section ($n = 582$). In fact, in the latter case, the estimated eigenvalues and eigenvectors are inevitably affected by the temporal dependence that underlies two consecutive observations of interest rates changes. For this reason, this application highlights the benefits of handling samples comprised of independent and identically distributed observations, compared with time series samples which might not be stationary in nature.

A potential future analysis of the bootstrap to this particular application should incorporate the estimated autocorrelation and cross-correlation of the interest rates changes, in order to obtain more representative "artificial samples". Consequently, the estimated eigenvectors and eigenvalues on each sample will be more accurate.

To conclude the premise, we might reasonably state that the results deriving from the non parametric bootstrap performed in this particular case should be considered significant either in case of stationary interest rates or *asymptotically*, meaning that these results would be attained only if we had an infinite amount of past observations such that any temporal dependency is zeroed.

After this needed specification, we briefly introduce the bootstrap.

The bootstrap[2] is a computer intensive resampling technique based on the simple but powerful idea of *repeated sampling* from a collection of available observations, with the objective of evaluating the uncertainty surrounding a parameter of interest. In our case, the original data set is made up of 3.492 daily observation across eleven interest rates. Then, the procedure is conducted as follows. We ask R to compose 10.000 distinct cross-sectional samples made of 582 observations each, by drawing

[1]The *bootstrap* produces approximately samples with i.i.d. observations, notwithstanding technicalities attributable to the process with which a computer generates random numbers.

[2]This technique was conceived by renowned statistician Bradley Efron as an improvement over the *jackniffe*.

randomly[3] from the actual data set. Subsequently, the eigenvectors and eigenvalues are computed on each of the ten thousand samples. As a result, we are capable of obtaining an estimate of their associated variability, under the *i.i.d.* assumption.

The results of such process are illustrated in the following figures and tables.

Figure 4.3a and 4.3b illustrates the empirical distributions and box plots of the first three eigenvalues resulting from 10.000 bootstrap samples. As we expected, the first eigenvalue consistently attains higher values compared to the second and third one, meaning that, under stationary conditions of interest rates changes, the first component contributes significantly to explain most of the covariation in the system, confirming, with the specifications mentioned above, the figures obtained in Section 3.6. Furthermore, we notice a similar degree of variability associated to λ_1 and λ_2 which is significantly higher than the one around λ_3.

Thus, we can conclude, with reasonable confidence, that the third component contributes to a much lesser extent in explaining the overall covariation of the interest rates changes. This is true, because the estimated density is highly concentrated around the measures of central tendencies, whilst the other two exhibit more spread.

After having considered the eigenvalues singularly, we evaluate the variability of the cumulative variance explained by the first three components. The results shown in Figure 4.4 and in Table 4.6 demonstrate that the first three components are capable of explaining an amount of variance equal to 0.926 on average.

For inferential purposes, it is useful to consider the 95% estimated confidence interval for the four statistics obtained from 10.000 bootstrap samples:

Table 4.2: 95% confidence interval for the statistics resulting from 10.000 bootstrap sample

Statistic	`0.025`	`0.0975`
λ_1	154.23	240.31
λ_2	35.48	101.64
λ_3	15.35	28.02
$\frac{\lambda_1+\lambda_2+\lambda_3}{(\lambda_1+\lambda_2+...+\lambda_{11})}$	0.908	0.942

Therefore, if we were to repeat the estimation process of the statistics above one hundred times, we are confident that they would be included in the provided intervals 95 times out of 100.

Finally, Figure 4.5 represents the "probabilistic"counterpart of the first three eigenvectors typical structure which identifies a *parallel shift*, *tilt* and *curvature* of the yield curve, respectively.

The figure should be read from the top to the bottom. Each quadrant illustrates

[3]The random draws should be performed with replacement.

Table 4.3: Descriptive statistics resulting from 10.000 bootstrap samples of λ_1.

Eigenvalue	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max**	**Sd.**
λ_1	124.2	174.6	187.9	190.4	203.6	308.1	22.15

Table 4.4: Descriptive statistics resulting from 10.000 bootstrap samples of λ_2.

Eigenvalue	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max**	**Sd.**
λ_2	19.29	53.66	65.42	66.08	77.74	131.25	17.17

Table 4.5: Descriptive statistics resulting from 10.000 bootstrap samples of λ_3.

Eigenvalue	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max**	**Sd.**
λ_3	11.95	18.20	20.14	20.51	22.38	39.54	3.25

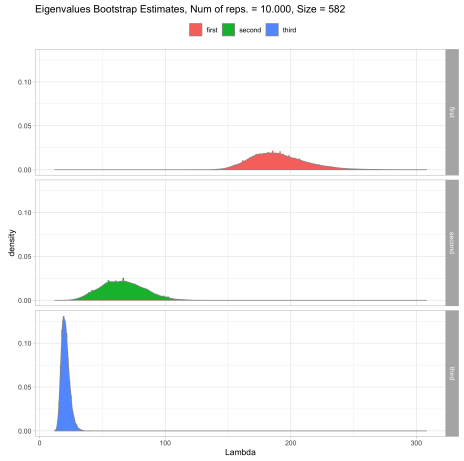
Table 4.6: Descriptive statistics resulting from 10.000 bootstrap samples of the cumulative variance explained by the first three components.

Var. Exp.	**Min.**	**1st Qu.**	**Median**	**Mean**	**3rd Qu.**	**Max**	**Sd.**
$\frac{\lambda_1+\lambda_2+\lambda_3}{(\lambda_1+\lambda_2+...+\lambda_{11})}$	0.891	0.921	0.927	0.926	0.933	0.954	0.009

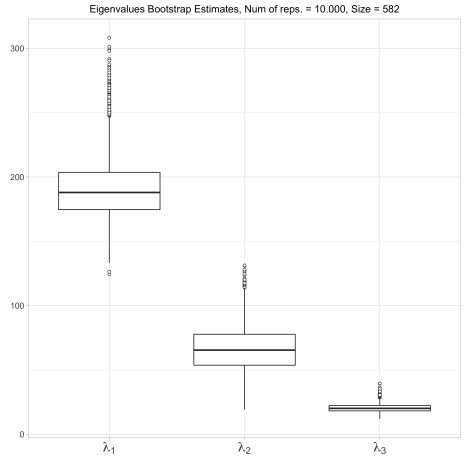
the values taken by the corresponding loading resulting from 10.000 bootstrap samples. The first eigenvector, in red, assumes with higher probability similar values across the eleven maturities, confirming the idea that it captures a parallel shift of the yield curve.

In the same way we can observe that the second eigenvector, in green, most of the times represent a tilt of the yield curve, meaning that yields on shorter maturities witness a positive change, whereas the yields on longer ones change negatively. We might notice also that, with lower probability, the second eigenvector captures an inverted behaviour in which the longer yields maturities have positive changes.

Finally, the third eigenvector almost always shows the typical curvature component.



(a) Estimated density on first three Eigenvalues - 10.000 Bootstrap sample



(b) Boxplots on first three Eigenvalues - 10.000 Bootstrap sample

Figure 4.3: Estimated density and boxplots of the first three Eigenvalues obtained from 10.000 bootstrap estimates.

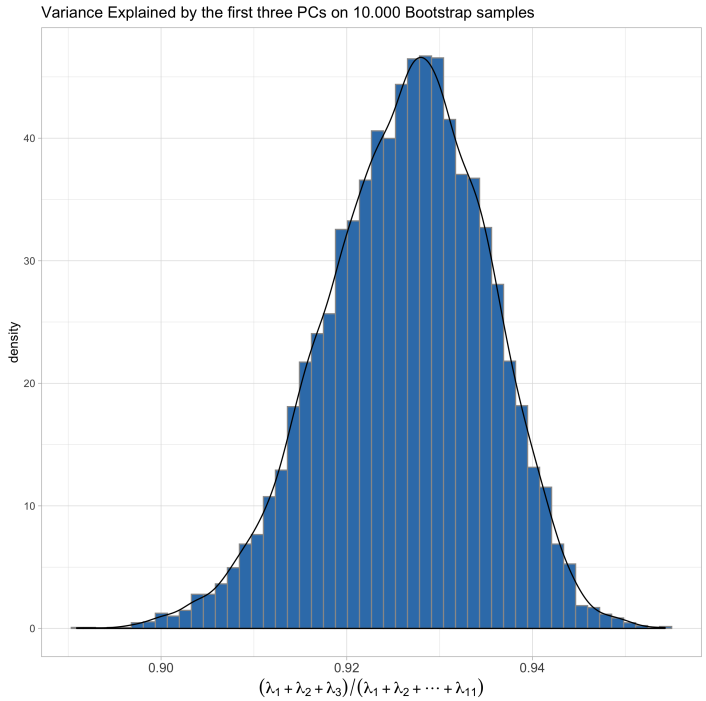


Figure 4.4: Kernel estimated density of the variance explained by the first three Eigenvalues.

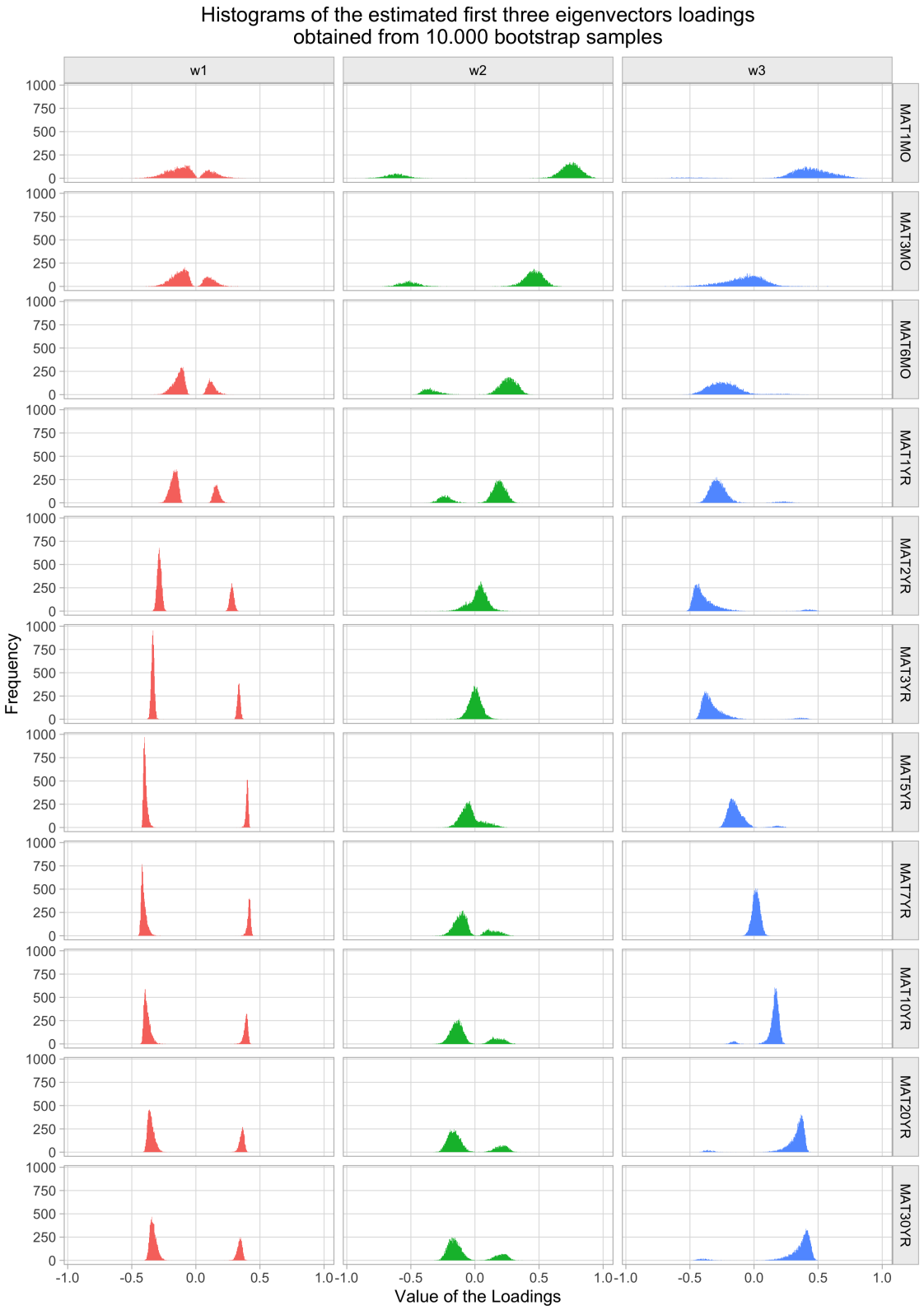


Figure 4.5: Histograms of the estimates of the first three eigenvectors loadings obtained from 10.000 bootstrap samples.

4.4 Portfolio Application

This section demonstrates how to build a principal component factor model for interest rate sensitive portfolios.

The portfolio chosen for this application comprises twenty six U.S. Government bonds expiring in a range of time that goes from a few months to 30 years (see Table 4.11 for all the details). The market prices of the securities refers to March 10, 2020, as provided by *Business Insider*[4]. Bonds data were retrieved by means of a Python script which have semi-automated the process of data collection (see Appendix D). If run on a terminal, the program keeps asking for the bond data which can be copied and pasted from the site. Afterwards, the figures are stored in a dictionary which is then converted into a *csv* file. Subsequently, that file has been read using R in order to perform the analysis (see Appendix A).

For the sake of simplicity, we assume to have acquired from the market one unit of each security at the market price listed on March 10, 2020 which we set to be our reference time point t.

The portfolio cash flow along the term structure is illustrated in Figure 4.9 wherein the dashed lines represent the eleven *vertices* of the risk factors for which historical observations are available (see the well known Figure 3.3). In fact, the portfolio value is sensitive to those constant key rates. However, since most of the projected cash flows at time t occur between those vertices, we have opted to use Svensson's model[5] in order to obtain the *interpolated* spot rate for each of the requested non standard maturity. This can be done because each cash flow can be regarded as a zero-coupon bond whose corresponding yield is the one approximated by means of Svensson's model.

Figure 4.8 shows the interpolation performed by Svensson's model[6] on the yield curve observed on March 10, 2020 (see Table 4.7) at the requested cash flow maturities measured in years. In other words, since the portfolio cash flow is a vector $\boldsymbol{c^T} = (C_1, C_2, \ldots, C_{226})$ with dimension (1×226), Svensson's model adds $226 - 11$ points to the yield curve at the constant desired time points indicated by the cash flow maturities.

Afterwards, the process shown for the daily yield curve of March 10 is repeated for each of the past curves, starting from February 09, 2006 to March 10, 2020, in order to reconstruct the cross-sectional term structure. The final result is shown

[4]https://markets.businessinsider.com/bonds/finder

[5]An alternative strategy relies on *Cash Flow Mapping* which consists in assigning each cash flow falling on non key maturities to the vertices by keeping the main financial characteristics of the portfolio invariant. Since this process would have been quite laborious to carry out in R, we have decide to embrace Svensson's model which is capable of approximating the spot rates at the requested non standard maturities.

[6]In R the *YieldCurve* package implements Nelson-Siegel, Diebold-Li and Svensson models.

in Figure 4.6. In practical terms, applying Svensson's model to the historical yield curves grows the dataset of interest rates from an original dimension of (3521×11) to a size of (3521×226). In other words, we use Svensson's model to reconstruct the past term structure as if we were capable of observing each day the yield curve at the intermediate maturities.

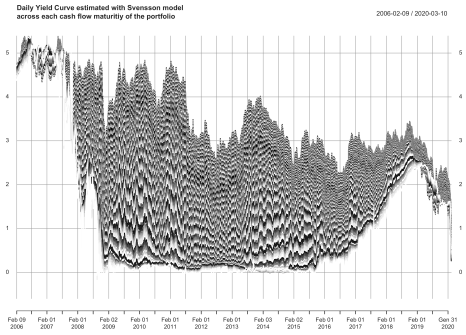
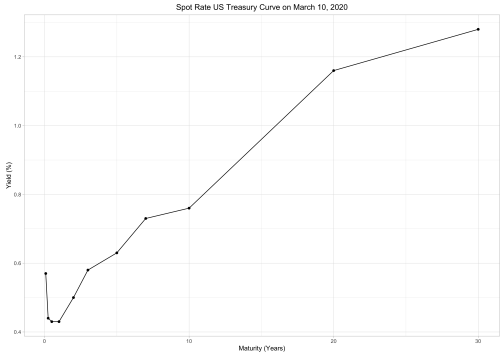


Figure 4.6: Approximated U.S. term structure by means of the Svensson's model.

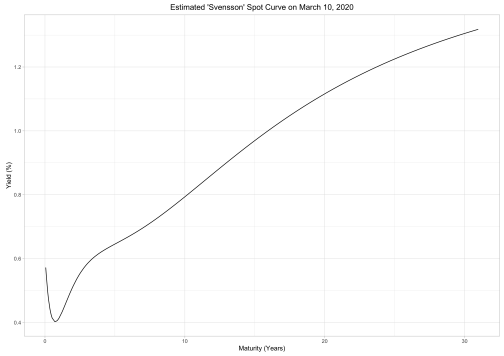
The functional form for the estimated spot yield curve by Svensson's model, implemented in the *YieldCurve* R package is the following:

$$y_t(\tau) = \beta_0 + \beta_1 \frac{1 - e^{(-\frac{\tau}{\lambda_1})}}{\frac{\tau}{\lambda_1}} + \beta_2 \left[\frac{1 - e^{(-\frac{\tau}{\lambda_1})}}{\frac{\tau}{\lambda_1}} - e^{(-\frac{\tau}{\lambda_1})} \right] + \beta_3 \left[\frac{1 - e^{(-\frac{\tau}{\lambda_2})}}{\frac{\tau}{\lambda_2}} - e^{(-\frac{\tau}{\lambda_2})} \right]$$

where τ represents the input value of key interest rates available, whereas the remaining parameters are optimized through a proper algorithm. For instance, Figure 4.8b is generated with estimated parameters in Table 4.8.



(a) Spot Rate US Treasury Curve on March 10, 2020.



(b) Estimated Svensson Spot Curve on March 10, 2020.

Figure 4.7: Actual Yield Curve and estimated one by means of the Svensson's model.

The approximated yield curve on March 10, 2020 is then employed to compute the vector $\boldsymbol{p^T} = (PV01^{(1)}, \ldots, PV01^{(226)})$ at time t of present value of basis point move associated to each of the portfolio cash flow, in the same way did in Section 3.5. As a result we obtain an *exact* representation of what would happen at time t to

Table 4.7: US Treasury Yield curve on March 10, 2020. Source: US Treasury

1MO	3MO	6MO	1YR	2YR	3YR	5YR	7YR	10YR	20YR	30YR
0.57	0.44	0.43	0.43	0.5	0.58	0.63	0.73	0.76	1.16	1.28

Table 4.8: Estimated coefficients of the Svensson's model for the Yield Curve on March 10, 2020.

β_0	β_1	β_2	β_3	λ_1	λ_2
1.73	-1.12	-1.67	-2.66	0.60	4.18

the present value of each $C^{(i)}$ if the yield curve was to move downward by 0.01% at each of the n maturities.

Prior to proceed the analysis, we introduce the *profit and loss* of the portfolio which is defined as the change in value between two consecutive days.

$$\Delta P_t = P_t - P_{t-1}$$

Then, "the [profit and loss] on the portfolio is approximated as a weighted sum of the changes in the interest rate risk factors with weights given by the present values of a basis point at the maturity corresponding to the interest rate"(Alexander, 2008a, p. 62). Therefore, we shall represent the profit and loss at time t in the following way:

$$P_t - P_{t-1} = -\sum_{i=1}^{226} C^{(i)} \delta 01_t (R_t^{(i)} - R_{t-1}^{(i)}) \tag{4.1}$$

$$\Delta P_t = -\sum_{i=1}^{226} PV01^{(i)} \Delta R_t^{(i)} \tag{4.2}$$

$$\Delta P_t = -\boldsymbol{p^T \Delta R_t} \tag{4.3}$$

where the vector $\boldsymbol{\Delta R_t}^{\boldsymbol{T}} = (\Delta R_t^{(1)}, \ldots, \Delta R_t^{(226)})$ contains the *changes* between two consecutive yield curves estimated using Svensson. The minus sign is due to the convention of representing the losses as positive quantities. "The [...] vector [$\boldsymbol{p^T}$] is held fixed at its current value so that we are measuring the interest rate risk of the *current* portfolio"(Alexander, 2008a, p.62).

At this point, if principal component were not known, we would be constrained to model the joint behaviour of the entire set of 226 interest rates changes. This would

imply to consider 226 variances and 25.425 covariances, for a total number of 25.651 parameters which is clearly unfeasible. Conversely, using the principal component approximation derived in Chapter 3, we are able to represent the interest rates changes using only three principal components while being still capable of explaining most of the covariation in the system (see Figure 4.8a which displays the percentage variance explained by each principal component).

Then, the *principal component representation* using the first three components is obtained from the spectral decomposition of the (226×226) sample *covariance* matrix of the daily Svensson yield curve *changes* (see Figure 4.10a). The following equation have been derived in Chapter 3 in Equation 3.15.

$$\Delta R_t^{(i)} \approx w_1^{(i)} z_t^{(1)} + w_2^{(i)} z_t^{(2)} + w_3^{(i)} z_t^{(3)} \tag{4.4}$$

Using matrix notation the PCA approximation for the entire set of observations would be as follows:

$$\underset{(3520\times226)}{\boldsymbol{\Delta R}} \approx \underset{(3520\times3)}{\boldsymbol{Z}} \underset{(3\times226)}{\boldsymbol{W^T}} \tag{4.5}$$

Then, the *j-th* principal component *risk factor sensitivity* of the portfolio is computed as:

$$k_j = -\sum_{i=1}^{226} PV01^{(i)} w_j^{(i)} \tag{4.6}$$

or, equivalently,

$$k_j = -\boldsymbol{p^T w_j} \tag{4.7}$$

which measures the change in the portfolio value when the principal component risk factor changes, keeping all the other principal component risk factors constant (Alexander, 2008c, p. 33).

The entire set of risk factor sensitivities are computed as follows:

$$\underset{(1\times3)}{\boldsymbol{k^T}} = \boldsymbol{p^T W} \tag{4.8}$$

Eventually, *the principal component factor model representation* of the portfolio profit and loss is:

$$\Delta P_t \approx \boldsymbol{k^T z_t} \tag{4.9}$$

where $\boldsymbol{z_t^T} = (z_t^{(1)}, z_t^{(2)}, z_t^{(3)})$ and $\boldsymbol{k^T} = (k_1, k_2, k_3)$ denote the principal component

Table 4.9: Risk Factor Sensitivities.

k_1	k_2	k_3
0.2930	0.1360	-0.1417

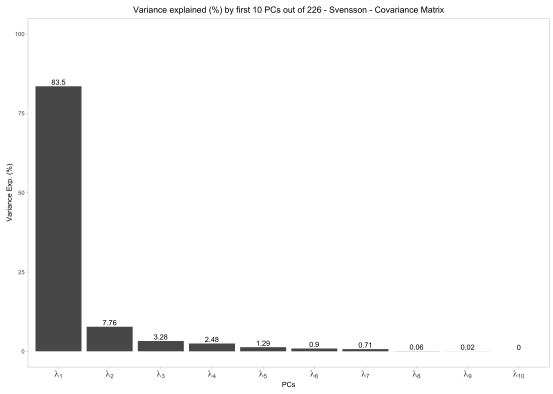
risk factor at time t, and their (constant) risk factor sensitivities.

Using PCA we reduced the number of risk factor from $n = 226$ to $k = 3$.

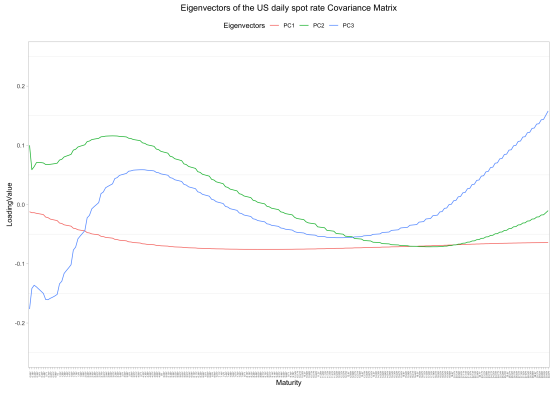
The estimated risk factor sensitivities are shown in Table 4.9. Therefore, the resulting PCA factor model for this example is:

$$-\text{p\&l}_t \approx \$0.2930 \times z_t^{(1)} + \$0.1360 \times z_t^{(2)} - \$0.1417 \times z_t^{(3)} \tag{4.10}$$

which can be used to immunize the portfolio against the most frequent movements of the yield curve.



(a) Percentage variance explained by the first ten principal components estimated on the term structure reconstructed using Svensson's model.



(b) First three eigenvectors.

Figure 4.8: Results of PCA performed on the interpolated interest rates changes.

Table 4.10: US Government Bonds paying semi annually on March 10, 2020. Source: Business Insider

Name	ISN	Price	C.R.	Mat.	Pymnt Day	Cp Left	Days to Cp
US TREASURY 2020	US9128282Q23	99.30	1.5000	2020-08-15	2020-08-15	1	158
US TREASURY 2021	US912828QV50	100.73	0.7135	2021-07-15	2020-07-15	3	127
US TREASURY 2022	US912828V723	102.56	1.8750	2022-01-31	2020-07-31	4	143
US TREASURY 2023	US9128284A52	106.05	2.6250	2023-02-28	2020-08-31	6	174
US TREASURY 2024	US9128283J70	105.62	2.1250	2024-11-30	2020-05-31	10	82
US TREASURY 2025	US9128284F40	108.54	2.6250	2025-03-31	2020-03-30	11	20
US TREASURY 2026 15.02	US912810EW46	127.43	6.0000	2026-02-15	2020-08-15	12	158
US TREASURY 2027 15.08	US912810FA17	138.05	6.3750	2027-08-15	2020-08-15	15	158
US TREASURY 2028	US9128283R96	105.56	0.5215	2028-01-15	2020-07-15	16	127
US TREASURY 2029 15.08	US912810FJ26	149.01	6.1250	2029-08-15	2020-08-15	19	158
US TREASURY 2030 15.5.	US912810FM54	154.21	6.2500	2030-05-15	2020-05-15	21	66
US TREASURY 2031	US912810FP85	148.56	5.3750	2031-02-15	2020-08-15	22	158
US TREASURY 2036	US912810FT08	155.39	4.5000	2036-02-15	2020-08-15	32	158
US TREASURY 2037	US912810PU60	166.94	5.0000	2037-05-15	2020-05-15	35	66
US TREASURY 2038	US912810PX00	160.25	4.5000	2038-05-15	2020-05-15	37	66
US TREASURY 2039	US912810QC53	159.91	4.5000	2039-08-15	2020-08-15	39	158
US TREASURY 2040	US912810QK79	151.81	3.8750	2040-08-15	2020-08-15	41	158
US TREASURY 2041	US912810QQ40	160.27	4.3750	2041-05-15	2020-05-15	43	66
US TREASURY 2042	US912810QX90	132.29	2.7500	2042-08-15	2020-08-15	45	158
US TREASURY 2043	US912810RB61	131.44	2.8750	2043-05-15	2020-05-15	47	66
US TREASURY 2044	US912810RG58	142.83	3.3750	2044-05-15	2020-05-15	49	66
US TREASURY 2045	US912810RK60	124.17	2.5000	2045-02-15	2020-08-15	50	158
US TREASURY 2046	US912810RQ31	124.95	2.5000	2046-02-15	2020-08-15	52	158
US TREASURY 2047	US912810RZ30	137.04	2.7500	2047-11-15	2020-05-15	56	66
US TREASURY 2048	US912810SE91	150.75	3.3750	2048-11-15	2020-05-15	58	66
USA 19/49	US912810SK51	129.01	2.3750	2049-11-15	2020-05-15	60	66

Table 4.11: Portfolio cash flows and $\boldsymbol{p^T}$ vector.

i	**Maturity**	**C ($)**	$R_t^{(i)}$	$R_{t-1}^{(i)}$	**PV ($)**	$R_t^{(i)} - 0.01\%$	**PV ($)**	**PV01**
1	0.055 56	100.750 00	0.572 16	0.542 28	100.718 07	0.005 62	100.718 63	0.000 56
2	0.183 33	14.492 50	0.501 20	0.437 96	14.479 22	0.004 91	14.479 49	0.000 26
3	0.227 78	3.125 00	0.482 42	0.409 41	3.121 58	0.004 72	3.121 65	0.000 07
4	0.352 78	4.937 50	0.442 76	0.346 54	4.929 81	0.004 33	4.929 98	0.000 17
5	0.397 22	2.500 00	0.432 65	0.329 52	2.495 72	0.004 23	2.495 82	0.000 10
6	0.438 89	18.687 50	0.424 79	0.315 73	18.652 77	0.004 15	18.653 58	0.000 82
7	0.483 33	1.187 50	0.417 96	0.303 14	1.185 11	0.004 08	1.185 17	0.000 06
8	0.700 00	14.492 50	0.402 87	0.266 80	14.451 77	0.003 93	14.452 78	0.001 01
9	0.744 44	3.125 00	0.402 66	0.263 41	3.115 67	0.003 93	3.115 90	0.000 23
10	0.869 44	4.937 50	0.405 82	0.259 27	4.920 14	0.003 96	4.920 57	0.000 43
...	...	...	...	...	...	...	...	...
216	27.866 67	1.1875	1.274 00	0.981 93	0.834 49	0.012 64	0.836 79	0.002 30
217	28.338 89	3.0625	1.281 24	0.989 30	2.134 96	0.012 71	2.140 94	0.005 98
218	28.383 33	1.1875	1.281 92	0.989 98	0.827 22	0.012 72	0.829 54	0.002 32
219	28.855 56	103.0625	1.288 93	0.997 15	71.220 69	0.012 79	71.423 88	0.203 20
220	28.900 00	1.1875	1.289 58	0.997 82	0.820 00	0.012 80	0.822 34	0.002 34
221	29.372 22	1.6875	1.296 39	1.004 80	1.155 94	0.012 86	1.159 30	0.003 36
222	29.416 67	1.1875	1.297 02	1.005 44	0.812 83	0.012 87	0.815 19	0.002 36
223	29.888 89	101.6875	1.303 61	1.012 24	69.046 99	0.012 94	69.251 02	0.204 03
224	29.933 33	1.1875	1.304 22	1.012 87	0.805 72	0.012 94	0.808 10	0.002 38
225	30.450 00	1.1875	1.311 21	1.020 10	0.798 66	0.013 01	0.801 07	0.002 40
226	30.966 67	101.1875	1.317 99	1.027 13	67.457 92	0.013 08	67.664 42	0.206 50
					3481.415		3485.724	4.308 82

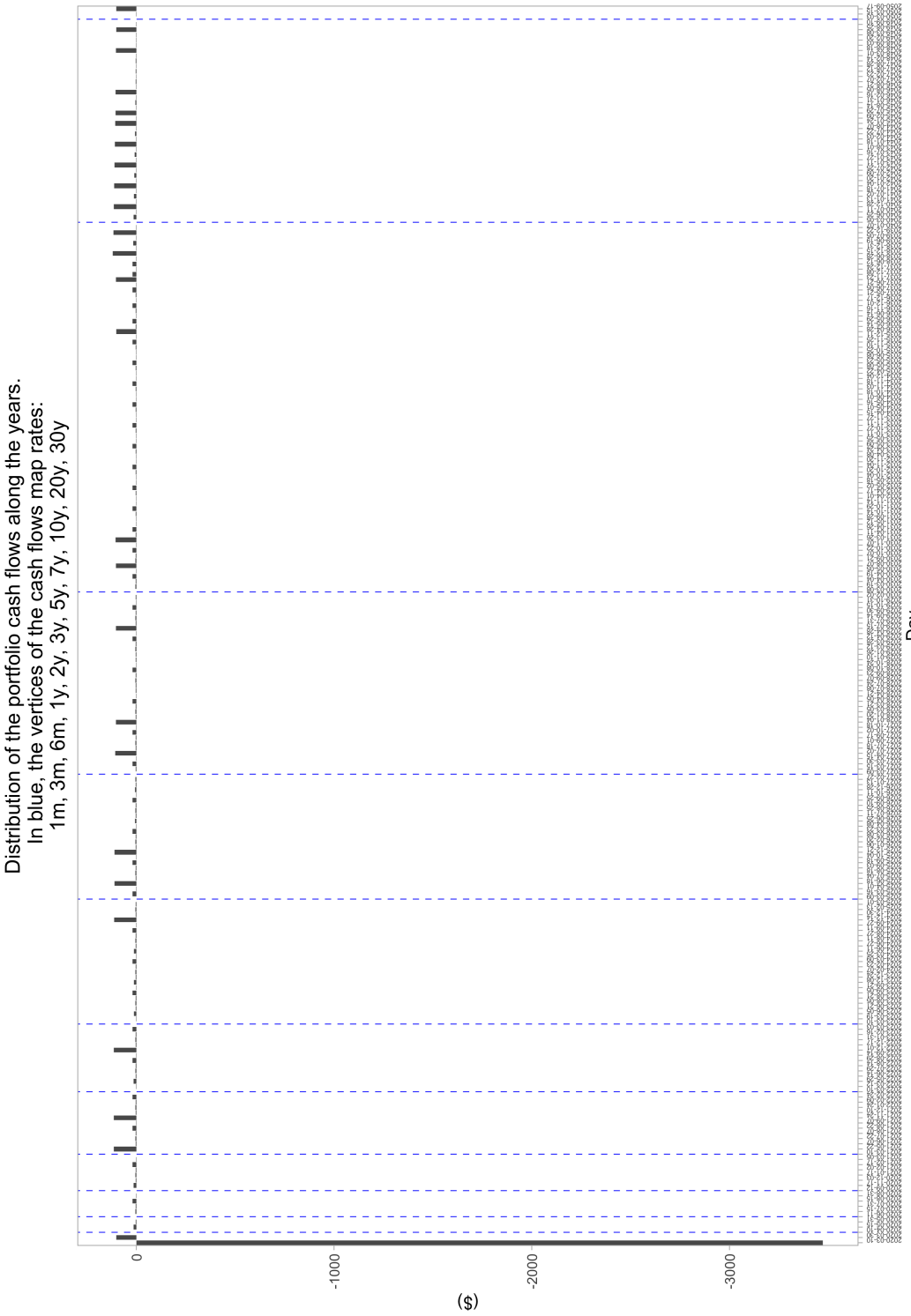
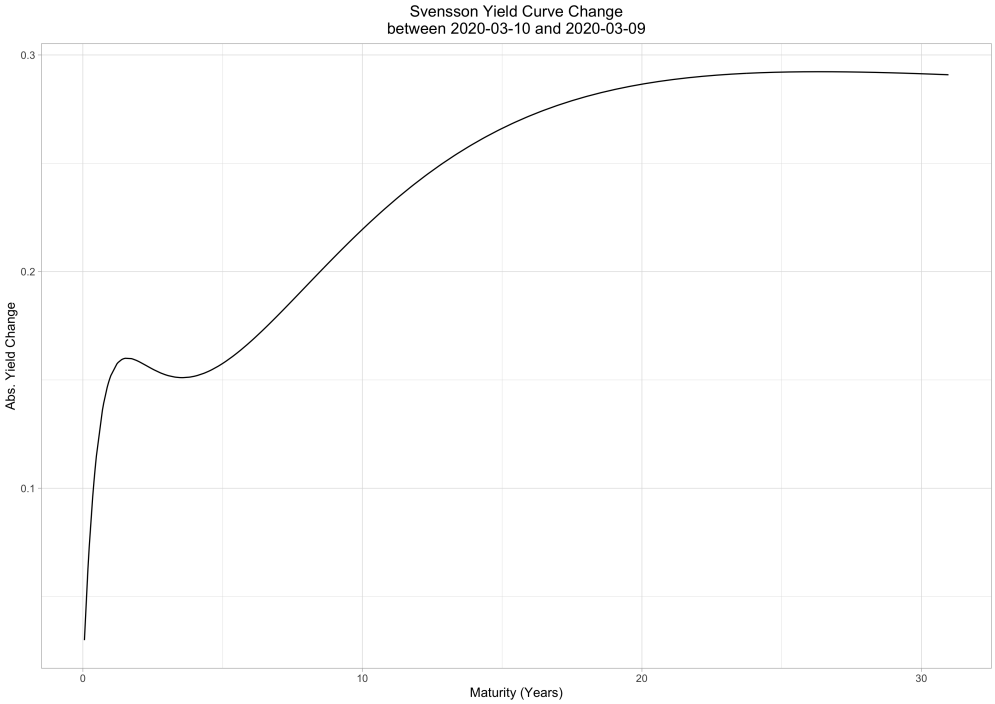
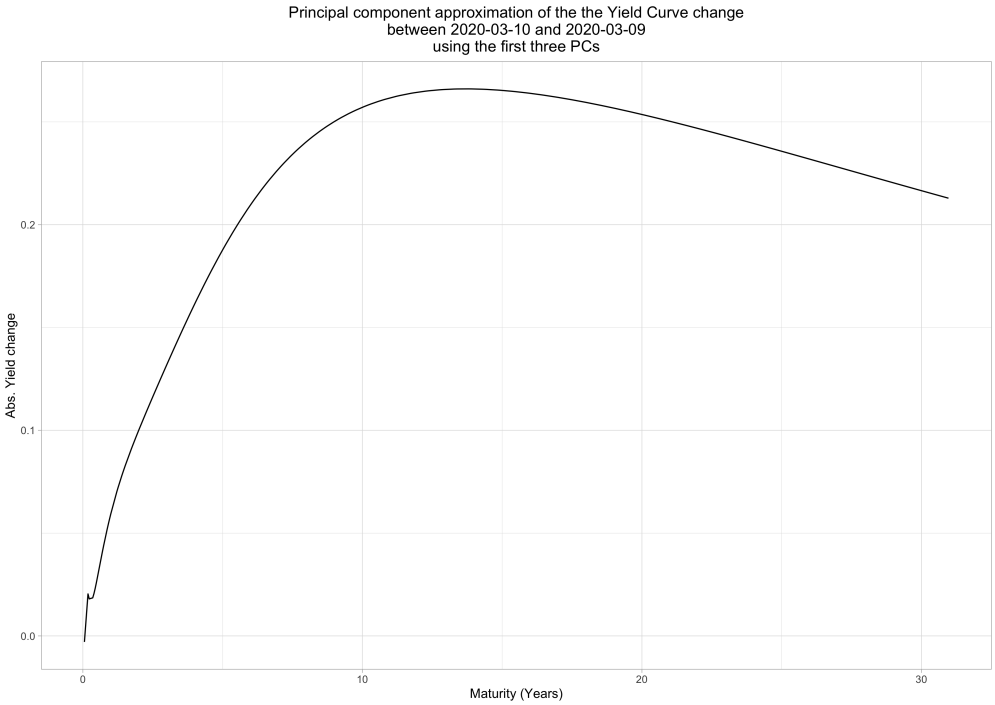


Figure 4.9: Representation of the portfolio cash flows at time t.



(a) Svensson Yield curve change between 10-03-20 and 09-03-20



(b) PC approximation of the Svensson Yield Curve Change between 10-03-2020 and 09-03-2020

Figure 4.10: Comparison between Svensson yield curve change and Principal component approximation

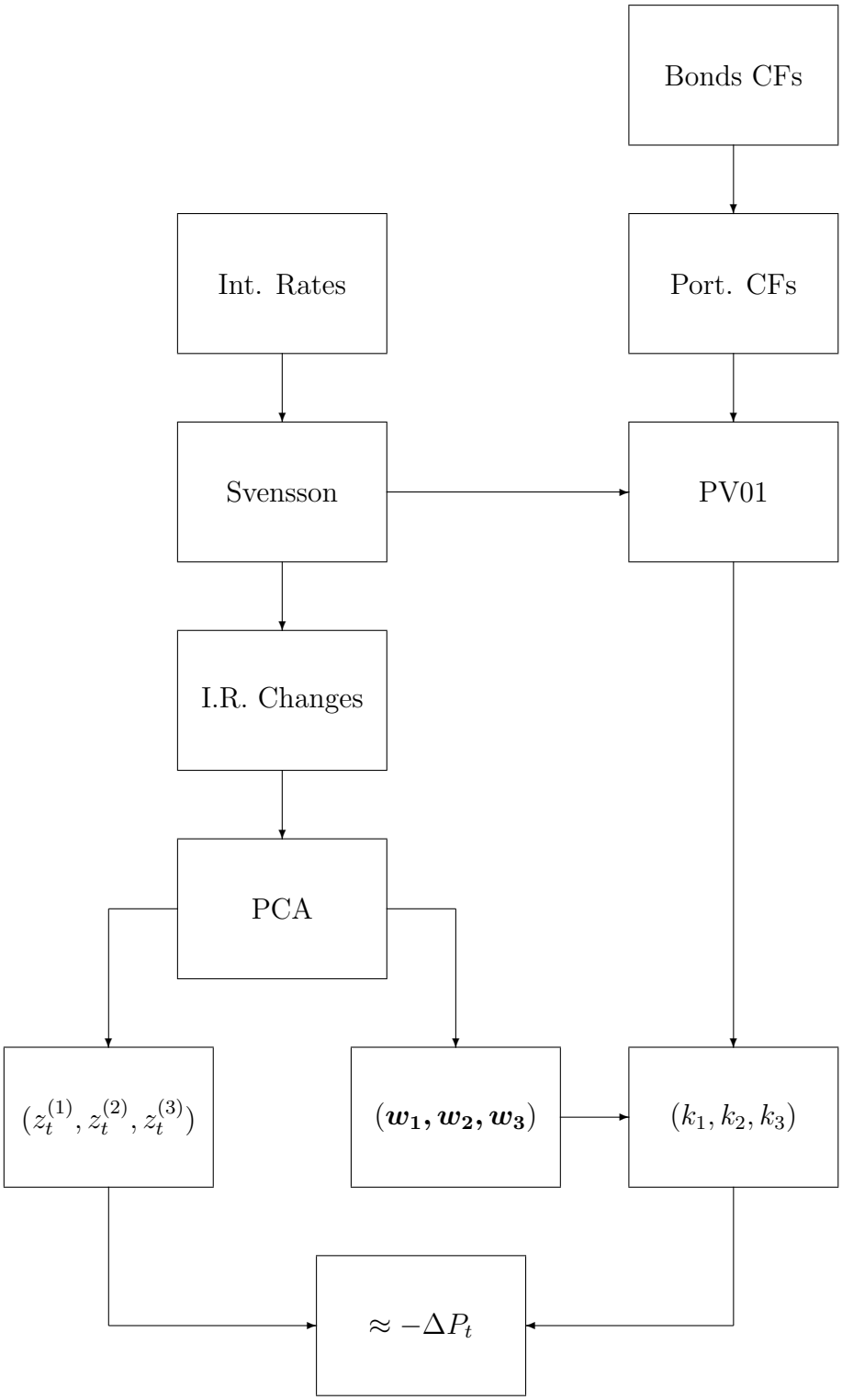


Figure 4.11: Diagram of the analysis.

Chapter 5

Conclusion

This text examined the US spot term structure observed between 2006 and 2020. In particular, the empirical analysis was conducted on a multivariate time series comprised of eleven key interest rates provided by the US Treasury which showed high correlation, in particular between yields of closer maturity. As a consequence, we were able to approximate the dynamics of the interest rates changes with an accuracy of almost 91% using only the first three principal components computed on those interest rates. In addition, we confirmed the existence of the traditional structure for the eigenvectors of the sample correlation matrix identifying a *parallel shift* a *tilt* and *curvature* as main yield curve movements occurring between consecutive time instances. However, as regard to stability, this was not entirely true when the analysis was performed on shorter successive time windows, even if some regularities were still evident, in particular as far as the first eigenvector was concerned. This might suggests that on shorter time windows principal component analysis is affected by short term volatilities of the interest rates.

Further, we showed the approximated empirical distribution associated to the first three eigenvalues and eigenvectors resulting from 10.000 bootstrap samples. We were immediately able to conclude that even if the first two eigenvalues attained consistently higher values compared to the third one, there were a greater amount of uncertainty underlying their distributions. Moreover, we were able to show that the first three components were able to explain about 92% of the variation in the interest rates changes within a 95% confidence interval, under the *i.i.d.* assumption which is not necessarily true in the context of financial time series.

In the last section we included in the analysis a portfolio made of bonds expiring at different maturities. We then succeeded to approximate the historical yield curves using Svensson's model to interpolate the historical daily yield curves, in order to obtain a series of risk factors associated to portfolio cash flows projected from March 10, 2020. Finally we employed the principal component representation to approximate the profit and loss of the portfolio which is of interest in risk management

applications.

To conclude, it seems useful to provide some ideas for potential future analysis:

1. Coding a bootstrap function that might take into consideration the autocorrelation of the interest rates.

2. Investigating more deeply the differences arising from applying spectral decomposition on the correlation matrix instead of the covariance.

3. Applying the technique of cash flow mapping and investigating the possible differences with the results obtained with Svensson's model.

4. Making further steps in the understanding of the uses of the profit and loss obtained by means of PCA.

Appendix A

R Code: PCA on Interest Rate Sensitive Portfolio

This Appendix illustrates the R Code of computations performed in Chapter 4. First of all, we load the data frame containing the specifics of the U.S. Governments bond included in the portfolio. Appendix D shows the Python script employed to automate the data collection process from *Business Insider*.

The main steps performed are: (1) composition of the portfolio cash flow starting from the individual bonds, (2) computation of the vector comprising the $PV01_{(i)}$, (3) acquisition of the eigenvectors, (4) calculation of risk factor weghts.

```
#### INTEREST RATE SENSITIVE PORTFOLIO ####
library(xts)
library(grid)
library(stringr)
library(highcharter)
library(lubridate)
library(tidyverse)
library(jrvFinance)
library(RColorBrewer)
cols = brewer.pal(n = 3,name = "Set1")

bonds <- read.csv("bonds_dataframe.csv")

# COMPUTATIONS ON BOND DATA FRAME
bonds <- bonds %>%
  mutate(PURCHASE_DAY = "2020-03-10") %>%
  mutate_at(c("ISSUE_DATE",
              "COUPON_PYMT_DATE",
```

```
                "MATURITY",
                "STARTCOUPON_DATE",
                "FINALCOUPON_DATE"),
              as.Date, format = "%m/%d/%Y") %>%
  mutate_at(c("ISSUE_DATE",
                "COUPON_PYMT_DATE",
                "MATURITY",
                "STARTCOUPON_DATE",
                "FINALCOUPON_DATE",
                "PURCHASE_DAY"),
              as.Date, format = "%Y-%m-%d") %>%
  mutate(DAYS_TO_COUPON = COUPON_PYMT_DATE - PURCHASE_DAY
     ) %>%
  mutate(YTM_2 = sqrt(1+YTM)-1) %>%
  # Semi annual coupon payment amount
  mutate(SEMI_COUPON_AMOUNT = ((FACE_VAL*(CP_RATE/100))/
     2)) %>%
  mutate(NUMB_PAYMENTS = 2) %>%
  rename(POSTED_YTM = YTM)

# CALCULATE COUPONS TO EXPIRATION
for (i in 1:dim(bonds)[1]){
  bonds[i,"N_CFs"] <- coupons.n(bonds[i,"PURCHASE_DAY"],
                                bonds[i,"MATURITY"],
                                freq = 2)}

# CREATE LIST COMPRISING THE C.F.
cash_flows <- vector(mode = "list", length = dim(bonds)
   [1])
for (cp in 1:dim(bonds)[1]) {
  cash_flows[[cp]] <- c(-bonds[cp,c("MRKT_PRICE")],
                        rep(bonds[cp,c("SEMI_COUPON_
                           AMOUNT")],
                            bonds[cp,c("N_CFs")]-1),
                        bonds[cp,c("SEMI_COUPON_AMOUNT")]
                            +
                          bonds[cp,c("FACE_VAL")])}

# CALCULATE SEMI ANNUAL YTM FOR EACH BOND
```

```
yields_to_mat_semi <- c()
for (i in 1:length(cash_flows)){
  yields_to_mat_semi[i] <- irr(cash_flows[[i]],
                               interval = c(-1,1),
                               cf.freq = 2,
                               comp.freq = 2)}
bonds[,"SEMI_ANN_YTM"] <- yields_to_mat_semi
bonds <- bonds %>%
  mutate(ANNUAL_YTM=((1+SEMI_ANN_YTM)^2)-1)

# FIND DATES OF THE 11 KEY RATES
s <- ymd("2020-03-10")
KEYRATES <- c(s+ddays(31),s+3*ddays(31),
              s+6*ddays(31),s+dyears(1),
              s+2*dyears(1),s+3*dyears(1),
              s+5*dyears(1),s+7*dyears(1),
              s+10*dyears(1),s+20*dyears(1),
              s+30*dyears(1))

# CREATE FACTOR ACCORDING TO: COUPON_PYMT_DATE.
# THIS IS DONE TO FIND PORTFOLIO C.F.
coupon_payment_date <- sort(ymd(bonds[,"COUPON_PYMT_DATE"
  ]))
fct <- factor(coupon_payment_date,
              labels = c("I","II","III","IV","V","VI","
                VII"))
bonds <- bonds %>%
  mutate(GROUP = fct)

# ORDER THE 7 DISTINCT FIRST DATE OF C.F. PYMT
c_pmyt_dates_7 <- sort(unique(ymd(bonds[,"COUPON_PYMT_
  DATE"])))
names(c_pmyt_dates_7) <- levels(fct)

# CREATE 7 SERIES OF DATES TO MATCH THE BONDS C.F.
s1 <- c_pmyt_dates_7[1]
dates_GI <- c(ymd("2020-03-10"),s1)
for(i in 1:60){
  dates_GI[i+2] <- s1+(i*(6*ddays(31)))
```

```
}

s2 <- c_pmyt_dates_7[2]
dates_GII <- c(ymd("2020-03-10"),s2)
for(i in 1:60){
  dates_GII[i+2] <- s2+(i*(6*ddays(31)))
}

s3 <- c_pmyt_dates_7[3]
dates_GIII <- c(ymd("2020-03-10"),s3)
for(i in 1:60){
  dates_GIII[i+2] <- s3+(i*(6*ddays(31)))
}

s4 <- c_pmyt_dates_7[4]
dates_GIV <- c(ymd("2020-03-10"),s4)
for(i in 1:60){
  dates_GIV[i+2] <- s4+(i*(6*ddays(31)))
}

s5 <- c_pmyt_dates_7[5]
dates_GV <- c(ymd("2020-03-10"),s5)
for(i in 1:60){
  dates_GV[i+2] <- s5+(i*(6*ddays(31)))
}

s6 <- c_pmyt_dates_7[6]
dates_GVI <- c(ymd("2020-03-10"),s6)
for(i in 1:60){
  dates_GVI[i+2] <- s6+(i*(6*ddays(31)))
}

s7 <- c_pmyt_dates_7[7]
dates_GVII <- c(ymd("2020-03-10"),s7)
for(i in 1:60){
  dates_GVII[i+2] <- s7+(i*(6*ddays(31)))
}

# MATCH THE THE BONDS C.F. WITH THE FACTORS LEVELS
```

```
cash_flows_df <- lapply(cash_flows,as.data.frame)
names(cash_flows_df) <- bonds[,"GROUP"]

# CREATE PORTFOLIO C.F.
for (i in 1:length(cash_flows)) {
  if (names(cash_flows_df)[i] == names(c_pmyt_dates_7)
     [1]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_GI
       [1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[2]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_
       GII[1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[3]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_
       GIII[1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[4]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_
       GIV[1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[5]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_GV
       [1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[6]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_
       GVI[1:dim(cash_flows_df[[i]])[1]])
  }
  else if (names(cash_flows_df)[i] == names(c_pmyt_dates_
     7)[7]) {
    cash_flows_df[[i]] <- xts(cash_flows_df[[i]],dates_
       GVII[1:dim(cash_flows_df[[i]])[1]])
```

```
  }
}

for(i in 1:length(cash_flows_df)) {
  colnames(cash_flows_df[[i]]) <- c("CF")
}

CFs_all <- merge(cash_flows_df[[1]],
                 cash_flows_df[[2]],
                 cash_flows_df[[3]],
                 cash_flows_df[[4]],
                 cash_flows_df[[5]],
                 cash_flows_df[[6]],
                 cash_flows_df[[7]],
                 cash_flows_df[[8]],
                 cash_flows_df[[9]],
                 cash_flows_df[[10]],
                 cash_flows_df[[11]],
                 cash_flows_df[[12]],
                 cash_flows_df[[13]],
                 cash_flows_df[[14]],
                 cash_flows_df[[15]],
                 cash_flows_df[[16]],
                 cash_flows_df[[17]],
                 cash_flows_df[[18]],
                 cash_flows_df[[19]],
                 cash_flows_df[[20]],
                 cash_flows_df[[21]],
                 cash_flows_df[[22]],
                 cash_flows_df[[23]],
                 cash_flows_df[[24]],
                 cash_flows_df[[25]],
                 cash_flows_df[[26]], all = TRUE)
CFS <- apply(CFs_all,1,sum, na.rm = TRUE)

# CF DATES
dates_CFS <- as.Date(names(CFS))
CFS_df <- as.data.frame(CFS)
CFS_ts <- xts(CFS, dates_CFS)
```

```
# ADD DATES CORRESPONDING TO KEY RATES
dates_cfs_key <- sort(c(ymd(rownames(CFS_df)),KEYRATES))

# CALCULATE SPOT RATES AT INTERMEDIATE MATURITIES
# USING SVENSSON'S MODEL
library(YieldCurve)
maturities.Tres <- c(1/12,3/12,6/12,1,2,3,5,7,10,20,30)
lenghts_from_t0 <- list()
for(i in 1:length(dates_CFS)){
  lenghts_from_t0[[i]] <- dates_CFS[i] - ymd("2020-03-10"
     )
}
maturities.CFS <- unlist(lenghts_from_t0)[-1]/360
```

The interest rates should be loaded in order to have the following representation:

	MAT1MO	MAT3MO	MAT6MO	MAT1YR	MAT2YR	MAT3YR	MAT5YR	MAT7YR	MAT10YR	MAT20YR	MAT30YR
2006-02-09	4.32	4.52	4.67	4.66	4.66	4.62	4.55	4.55	4.54	4.72	4.51
2006-02-10	4.36	4.53	4.70	4.70	4.69	4.67	4.59	4.59	4.59	4.76	4.55
2006-02-13	4.38	4.55	4.71	4.70	4.68	4.66	4.58	4.58	4.58	4.76	4.56
2006-02-14	4.42	4.55	4.72	4.71	4.69	4.68	4.61	4.61	4.62	4.80	4.60
2006-02-15	4.39	4.55	4.70	4.70	4.71	4.68	4.60	4.60	4.61	4.78	4.58
2006-02-16	4.38	4.55	4.69	4.69	4.69	4.67	4.59	4.59	4.59	4.77	4.57

```
# LOAD INTEREST RATES DATA
X <- read.table("/Users/USYIELDS.csv")
X <- xts(X,ymd(rownames(X)))
X <- window(X, start="2006-02-09",end="2020-03-10")

# INTERPOLATE THE HISTORICAL YIELD CURVES WITH
# SVENSSON'S MODEL
# FINAL OBJECTIVE: INTEREST RATES CHANGES IN BASIS POINTS

SvensonParametersAll <- Svensson(rate = X, maturity =
   maturities.Tres)
SvensonParametersAll <- as.data.frame(
   SvensonParametersAll)

# SAVE THE RESULTS OF SVENSSON PARS. IN A CSV FILE
write.table(SvensonParametersAll,"/Users/SvenssonPars.csv
   ",sep=",")
```

```
# LOAD THE SVENSSON PARS.
SvensonParametersAll <- read.csv("/Users/SvenssonPars.csv
   ")
SvensonParametersAll <- xts(SvensonParametersAll,index(X)
   )

# INTERPOLATE SPOT RATES
Svensson.ratesAll <- Srates(SvensonParametersAll,
   maturities.CFS, "Spot")
labelsSvenssonAll <- str_c(round(maturities.CFS,2),"YR")
colnames(Svensson.ratesAll) <- labelsSvenssonAll
Svensson.rateAll <- as.data.frame(Svensson.ratesAll)

# SAVE THE RESULTS OF THE SPOT RATES
write.table(Svensson.ratesAll,"/Users/SvenssonRates.csv",
   sep=",")
Svensson.ratesAll <- read.csv("/Users/SvenssonRates.csv")
colnames(Svensson.ratesAll) <- labelsSvenssonAll
Svensson.ratesAll <- xts(Svensson.ratesAll,index(X))
dim(Svensson.ratesAll)

# ABSOLUTE INTEREST RATES CHANGES IN BPS
SvenssonX <- apply(Svensson.ratesAll,2,diff)*100
SvenssonX <- xts(SvenssonX,index(X)[-1])
SvenssonX[1,]
SvenssonX[3520,]
dim(SvenssonX)

# PCA ON SVENSSON YIELDS CURVES - COV. MATRIX
V <- cov(SvenssonX)
W <- eigen(V)$vectors
Z <- SvenssonX%*%W*(-1)
Z <- xts(Z,index(X)[-1])
colnames(Z) <- str_c("PC",seq(1,dim(Z)[2]))
colnames(W) <- str_c("w",seq(1,dim(Z)[2]))
Mat_labs <- str_c("MAT",round(maturities.CFS,2))
rownames(W) <- Mat_labs

Z3 <- Z[,1:3]
```

```
W3 <- W[,1:3]
head(Z3)
head(W3)
Zt <- Z[dim(Z)[1],1:3]
Zt

# PCA REPRESENTATION
# Xhat contains the approximated daily Yield curves
   changes
# using only the First 3 PCS
Xhat <- Z[,1:3]%*%t(W[,1:3])
colnames(Xhat) <- Mat_labs
Xhat <- xts(Xhat,index(X)[-1])
Xhat["2020-03-10"]

# PV01 OF THE PORTFOLIO CF
options(scipen=999)
options(pillar.sigfig = 6)
t = dim(Svensson.ratesAll)[1]
Svensson.ratesAll[t]
Int_CF <- cbind(CFS[-1],
                maturities.CFS,
                as.vector(Svensson.ratesAll[t]),
                as.vector(Svensson.ratesAll[t-1]))
colnames(Int_CF) <- c("C","Maturity","R10032020","
   R09032020")
head(Int_CF)
Int_CF <- Int_CF %>%
  as_tibble() %>%
  mutate(YTM = R10032020/100) %>%
  mutate(PVal = C/(1+YTM)^Maturity) %>%
  mutate(Yield_PV01 = YTM-0.01/100) %>%
  mutate(PVal001 = C/(1+Yield_PV01)^Maturity) %>%
  mutate(BP01 = (((1+YTM-(0.01/100))^(-Maturity))-((1+YTM
     )^(-Maturity)))) %>%
  mutate(PV01 = C*BP01) %>%
  mutate(DeltaR = R10032020 - R09032020) %>%
  select(Maturity, C, R10032020, R09032020, DeltaR,
     everything()) %>%
```

```
  mutate(PV01dotDeltaR = PV01*DeltaR)

PV01x <- sum(Int_CF[,"PV01"])
# Check with:
PV01 <- sum(Int_CF[,"PVal001"])-sum(Int_CF[,"PVal"])

# PRINCIPAL COMPONENT RISK FACTOR SENSITIVITY
Int_CF <- Int_CF %>%
  mutate(w1 = W[,1]) %>%
  mutate(w2 = W[,2]) %>%
  mutate(w3 = W[,3]) %>%
  mutate(rf1 = PV01*w1) %>%
  mutate(rf2 = PV01*w2) %>%
  mutate(rf3 = PV01*w3)

# RISK FACTOR SENSITIVITIES
RF1 = -sum(Int_CF[,"rf1"])
RF2 = -sum(Int_CF[,"rf2"])
RF3 = -sum(Int_CF[,"rf3"])

# GRAPH FIRST THREE PCS
highchart(type = "stock") %>%
  hc_add_series(Z[,1], name = "PC1") %>%
  hc_add_series(Z[,2], name = "PC2") %>%
  hc_add_series(Z[,3], name = "PC3") %>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_navigator(enabled = FALSE) %>%
  hc_scrollbar(enabled = FALSE) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = TRUE)

# GRAPH FIRST THREE EIGENVECTORS
maturities_labs <- list(rep(round(maturities.CFS,2),3)) #
    *3 for faceting
colnames(W) <- str_c("PC",seq(1,dim(W)[2],1))
W <- as.data.frame(W)
W3 <- W[,c(1:3)]
W3 <- stack(list(PC1 = W3[,1], PC2 = W3[,2], PC3 = W3
   [,3]))
```

```
W3[,"Maturity"] <- rep(str_c(round(maturities.CFS,2),"Z")
   ,3)
colnames(W3) <- c("LoadingValue", "Eigenvectors", "
   Maturity")
ggplot(W3, aes(x = Maturity, y = LoadingValue, colour =
   Eigenvectors, group = Eigenvectors)) +
  geom_line() +
  scale_x_discrete(limits = str_c(round(maturities.CFS,2)
     ,"Z")) +
  theme_light() +
  theme(legend.position="top") +
  ggtitle("Eigenvectors␣of␣the␣US␣daily␣spot␣rate␣
     Covariance␣Matrix") +
  ylim(c(-0.25,0.25)) +
  theme(panel.grid.major = element_blank(),
        axis.text.x = element_text(angle=90,size=rel(0.4)
           ),
        plot.title = element_text(hjust = 0.5))

# BAR PLOT VARIANCE EXP. OF THE PCS
SvenLamdasCov <- eigen(CovXSvensson,symmetric=TRUE)$
   values
SUMSvenLamdasCov <- sum(SvenLamdasCov)
VarExplainedSvenCov <- as.data.frame((SvenLamdasCov/
   SUMSvenLamdasCov)*100)
VarExplainedSvenCov[,"PC"] <- str_c("PC",seq(1,dim(
   XSvensson)[2],1))
colnames(VarExplainedSvenCov)[1] <-  "VarExp"
VarExplainedSvenCov <- VarExplainedSvenCov %>%
  mutate(VarExp = round(VarExp,2))

ggplot(VarExplainedSvenCov, aes(x=PC,y=VarExp)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=VarExp),vjust=-0.3) +
  theme_light() +
  scale_x_discrete(limits = str_c("PC",
                                  seq(1,dim(XSvensson)
                                     [2],1))[1:10],
                   labels = c(expression(lambda[1]),
```

```
                                expression(lambda[2]),
                                expression(lambda[3]),
                                expression(lambda[4]),
                                expression(lambda[5]),
                                expression(lambda[6]),
                                expression(lambda[7]),
                                expression(lambda[8]),
                                expression(lambda[9]),
                                expression(lambda[10]))) +
  scale_y_continuous(limits = c(0,100)) +
  theme(axis.text.x = element_text(size=rel(1.5)),
        plot.title = element_text(hjust = 0.5),
        panel.grid.major.x = element_blank(),
        panel.grid.major.y=element_blank(),
        panel.grid.minor.y=element_blank()) +
  ylab("Variance␣Exp.␣(%)") +
  xlab("PCs")

# TOY EXAMPLE OF PV01
options(pillar.sigfig = 6)
t <- c(1,2,3,4)
cf <- c(6,6,6,106)
int <- c(4.5,4.75,4.85,5)
d <- as.data.frame(cbind(t,cf,int))
d <- d %>%
  as_tibble(d) %>%
  mutate(pval=cf/(1+(int/100))^(t)) %>%
  mutate(int01=int-0.01) %>%
  mutate(pval01=cf/(1+int01/100)^(t)) %>%
  mutate(bp01 = (((1+(int/100)-(0.01/100))^(-t))-(1+int/
     100)^(-t))) %>%
  mutate(pv01=bp01*cf)
PV01 <- sum(d[,"pval01"])-sum(d[,"pval"])
```

Appendix B

R Code: PCA on US Yield Curves and Bootstrap analysis

Interest rates downloaded from the US Treasury are expressed in percentage. The analysis is conducted on *interest rates changes expressed in basis point*. Thus, $0.01\% = 1$Bps and $1\% = 100$Bps.

Please notice, that the interest rates are neither scaled nor "centered"since they are measured according to the same rule. For further considerations on the technical aspects involving principal component analysis in the context of interest risk analysis, see Redfern and McLean (2004).

```
# PACKAGES
library(xts)
library(ggplot2)
library(lubridate)
library(grid)
library(stringr)
library(highcharter)
library(RColorBrewer)
cols = brewer.pal(n = 3,name = "Set1")

# DATA LOADING
X <- read.table("/Users/USYIELD.csv")
X <- xts(X,ymd(rownames(X)))
X <- window(X, start="2006-02-09",end="2020-01-29")

Mat_labs <- colnames(X) # maturities
PCs_labs <- rep("PC",dim(X)[2]) # PCs
for(i in 1:dim(X)[2]){
```

```
  PCs_labs[i] <- str_c(PCs_labs[i], as.character(i), sep=
     "")
}

# INTEREST RATES CHANGES IN BASIS POINTS
Ts <- apply(X,2,diff)*100
dates <- index(X)[-1]
Ts <- xts(Ts,dates)

# PCA CORRELATION MATRIX
CorTs <- cor(Ts)
W <- eigen(CorTs)$vectors
Z <- Ts%*%W*(-1)
Z <- xts(Z,dates)
colnames(Z) <- PCs_labs

# GRAPH FIRST 3 PCS
highchart(type = "stock") %>%
  hc_title(text = "First␣three␣PCs␣-␣Corr.␣Matrix␣-␣
     Absolute␣rates␣changes␣in␣Bps") %>%
  hc_add_series(Z[, PCs_labs[1]], name = PCs_labs[1]) %>%
  hc_add_series(Z[, PCs_labs[2]], name = PCs_labs[2]) %>%
  hc_add_series(Z[, PCs_labs[3]], name = PCs_labs[3]) %>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_navigator(enabled = FALSE) %>%
  hc_scrollbar(enabled = FALSE) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = TRUE)

# GRAPH EIGENVECTORS CORRELATION MATRIX
colnames(W) <- PCs_labs # prcomp only for labs
W <- as.data.frame(W)
W3 <- W[,c(1:3)]
W3 <- stack(list(PC1 = W3[,1], PC2 = W3[,2], PC3 = W3
   [,3]))
W3[,"Maturity"] <- Mat_labs
colnames(W3) <- c("LoadingValue", "Eigenvectors", "
   Maturity")
```

```
ggplot(W3, aes(x = Maturity,
               y = LoadingValue,
               colour = Eigenvectors,
               group = Eigenvectors)) +
  geom_line() +
  scale_x_discrete(limits = colnames(Ts)) +
  geom_point() +
  theme_light() +
  theme(legend.position="top") +
  ylim(c(-0.5,0.75)) +
  theme(panel.grid.minor = element_blank(),
        plot.title = element_text(hjust = 0.5))

# PCA REPRESENTATION CORRELATION
Z3 <- Z[,1:3]
# Xhat contains in each column the vectors
# of the Perc. int. rate changes at distinct mat.
Xhat2 <- Z3%*%t(W[,1:3])
colnames(Xhat2) <- Mat_labs
Xhat <- xts(Xhat2,dates)
head(Xhat2)
Xhat2 <- xts(Xhat2,dates)
highchart(type = "stock") %>%
  hc_add_series(Xhat2[, Mat_labs[1]], name = Mat_labs[1])
      %>%
  hc_add_series(Xhat2[, Mat_labs[2]], name = Mat_labs[2])
      %>%
  hc_add_series(Xhat2[, Mat_labs[3]], name = Mat_labs[3])
      %>%
  hc_add_series(Xhat2[, Mat_labs[4]], name = Mat_labs[4])
      %>%
  hc_add_series(Xhat2[, Mat_labs[5]], name = Mat_labs[5])
      %>%
  hc_add_series(Xhat2[, Mat_labs[6]], name = Mat_labs[6])
      %>%
  hc_add_series(Xhat2[, Mat_labs[8]], name = Mat_labs[8])
      %>%
  hc_add_series(Xhat2[, Mat_labs[9]], name = Mat_labs[9])
      %>%
```

```
  hc_add_series(Xhat2[, Mat_labs[10]], name = Mat_labs
     [10]) %>%
  hc_add_series(Xhat2[, Mat_labs[11]], name = Mat_labs
     [11]) %>%
  hc_add_theme(hc_theme_flat()) %>%
  hc_navigator(enabled = FALSE) %>%
  hc_scrollbar(enabled = FALSE) %>%
  hc_exporting(enabled = TRUE) %>%
  hc_legend(enabled = TRUE)

# PCA ON SIX CONSECUTIVE TIME WINDOWS
len_window <- dim(Ts)[1]/6 # 582
n_windows = dim(Ts)[1]/len_window
tot_days = dim(Ts)[1]

# Create Indexes to slice Ts into 6 equal windows
points <- c()
for(window in 1:n_windows){
  points <- c(points, window*len_window)
}

# Populate a list with the 6 Time series
TSs <- vector(mode = "list", length = n_windows)
for(p in points){
  TSs[[p/len_window]] <- Ts[(p-len_window):p]
}

# Prcomp on each window
# Prcomp is equivalent to spectral decomposition but use
   SVD
# For each window we have:
# Absolute changes in basis points + EigenVals +
   EigenVecs + PCs
windows_pca <- lapply(TSs, FUN = prcomp, scale = FALSE,
   center = FALSE)

# Select eigenvectors from each ts of windows_pca
selected_eig_vecs = vector(mode = "list", length = n_
   windows)
```

```
for(i in 1:length(selected_eig_vecs)){
  selected_eig_vecs[[i]] <- windows_pca[[i]][[2]][,c("PC1
     ","PC2","PC3")]
}

# Select eigenvalues from each ts of windows_pca
selected_eig_vals = vector(mode = "list", length = n_
   windows)
for(i in 1:length(selected_eig_vals)){
  selected_eig_vals[[i]] <- windows_pca[[i]][[1]]
}
sum_selected_eigneval <- lapply(selected_eig_vals,sum)

var_exp <- vector(mode = "list", length = n_windows)
for(i in 1:length(selected_eig_vecs)){
  var_exp[[i]] <- (selected_eig_vals[[i]]/sum_selected_
     eigneval[[i]])*100
}

var_exp_first_3 <- vector(mode="list",length = n_windows)
for(i in 1:length(var_exp_first_3)){
  var_exp_first_3[[i]] <- sum(var_exp[[i]][1:3])
}

# EIGENVECTORS GRAPHS ON EACH
# We investigate the persistence of the fundamental
   structure:
# parallel shift, tilt, curvature
# Transform Eigenvectors into data.frame for ggplot2
selected_eig_vecs_df <- lapply(selected_eig_vecs,
                               FUN = as.data.frame)

# Create Labels
mat <- row.names(selected_eig_vecs_df[[1]])
mat <- str_replace(mat,"MAT", "")
maturities <- lapply(selected_eig_vecs_df, row.names)
PCs <- lapply(selected_eig_vecs_df,colnames)

# Stack the PCs of the 6 windows for ggplot2 faceting
```

```
stacked_dfs <- lapply(selected_eig_vecs_df, stack)
for(i in 1:length(selected_eig_vecs_df)){
  stacked_dfs[[i]] <- cbind(stacked_dfs[[i]],mat)
}

# Recover start-date and end-date for each window
startdate <- lapply(TSs,first)
endate <- lapply(TSs,last)

startdate <- lapply(startdate, index)
endate <- lapply(endate,index)

plots <- vector(mode="list", length = length(stacked_dfs)
   )
for(i in 1:length(stacked_dfs)){
  plots[[i]] <- ggplot(stacked_dfs[[i]], aes(x = mat,
                                             y = values,
                                             group = ind,
                                             colour = ind
                                                ,
                                             fill=ind)) +
    geom_line(size=1) +
    theme_light() +
    scale_x_discrete(limits = mat) +
    labs(x = "Maturity", y = "Loadings", colour = "
       Eigenvectors:") +
    scale_y_continuous(limits=c(-1.5,1.5), breaks = seq
       (-5,5,0.25)) +
    theme(legend.position = c(0.5,0.85),
          legend.title=element_text(size = 16),
          legend.text=element_text(size = 14),
          axis.text.x=element_text(size = rel(1.5)),
          axis.text.y=element_text(size=rel(1.5)),
          axis.title.x=element_text(size=14),
          axis.title.y=element_text(size=14),
          panel.grid.minor = element_blank(),
          plot.title = element_text(hjust = 0.5))}

# BOOTSTRAP
```

```
# Start from data TS: Absolute changes in basis points
# We create 10.000 bootstrap samples each made of 582 obs
    .
# Ratio 1/6 = 582/3492
# Each sample is stored into the list 'boot_samples'
reps=10000
boot_samples <- vector(mode="list", length = reps)
set.seed(1)
for(rep in 1:reps){
  boot_samples[[rep]] <- Ts[sample(1:dim(Ts)[1], 582,
     replace=TRUE),]
}

covariances <- lapply(boot_samples,cov)
E <- lapply(covariances, eigen, symmetric = TRUE)

# BOOTSTRAP EIGENVALUES
# Isolate the first eigenvalue for each bootstrap sample
# In total 10.000 lambda1
eigenvalues_first = vector(mode = "list", length = reps)
for(i in 1:length(E)){
  eigenvalues_first[[i]] <- E[[i]][[1]][1]
}
eig_first <- unlist(eigenvalues_first)
summary(eig_first)
round(quantile(eig_first,c(0.025,0.975)),2)
round(sd(eig_first),2)

# Isolate the second eigenvalue for each bootstrap sample
# In total 10.000 lambda2
eigenvalues_second = vector(mode = "list", length = reps)
for(i in 1:length(E)){
  eigenvalues_second[[i]] <- E[[i]][[1]][2]
}
eig_second <- unlist(eigenvalues_second)
summary(eig_second)
round(quantile(eig_second,c(0.025,0.975)),2)
round(sd(eig_second),2)
```

```
# Isolate the third eigenvalue for each bootstrap sample
# In total 10.000 lambda3
eigenvalues_third = vector(mode = "list", length = reps)
for(i in 1:length(E)){
  eigenvalues_third[[i]] <- E[[i]][[1]][3]
}
eig_third <- unlist(eigenvalues_third)
summary(eig_third)
round(quantile(eig_third,c(0.025,0.975)),2)
round(sd(eig_third),2)

# We organize the 10.000*lambda1, 10.000*lambda2, 10.000*
   lambda3
# In order to have a stacked data.frame for ggplot2
   faceting

# Create labels "first", "second", "third"
label_first <- rep("first",reps)
label_second <- rep("second",reps)
label_third <- rep("third",reps)

labels <- c(label_first,label_second,label_third)
eigenvalues <- c(eig_first,eig_second,eig_third)

eigenvalues_df <- data.frame(Lambda=as.numeric(
   eigenvalues),
                             Label=as.character(labels))

# Densities
ggplot(eigenvalues_df, aes(x=Lambda,
                           y=..density..,
                           fill = Label)) +
  geom_histogram(bins = 250) +
  geom_density(size=0.5,colour = "grey60") +
  facet_grid(Label ~ .) +
  theme_light() +
  guides(fill=guide_legend(title=NULL)) +
  theme(legend.position="top") +
  ggtitle("Eigenvalues␣Bootstrap␣Estimates")
```

```
# Box plots
ggplot(eigenvalues_df, aes(x=Label, y=Lambda)) +
  geom_boxplot(outlier.size=1.5,outlier.shape=21) +
  theme_light() +
  theme(axis.title.x=element_blank(),
        axis.title.y=element_blank(),
        axis.text.x = element_text(size = rel(2)),
        axis.text.y=element_text(size=rel(1.25)),
        plot.title = element_text(hjust=0.5)) +
  scale_x_discrete(labels=c(expression(lambda[1]),
                            expression(lambda[2]),
                            expression(lambda[3]))) +
  ggtitle("Eigenvalues␣Bootstrap␣Estimates,␣Num␣of␣reps.␣
     =␣10.000,␣Size␣=␣582")

#### Bootstrapped Variance Explained by the first three
   eigenvalues ####
eigenvalues_all = vector(mode = "list", length = reps)
for(i in 1:length(E)){
  eigenvalues_all[[i]] <- E[[i]][[1]]
}

eigenvalue_sum <- lapply(eigenvalues_all, sum)
eig_sum_all <- unlist(eigenvalue_sum)

# Bootstrapped First three Eigenvalues in a 3 column
   dataframe
eigenvalues_df_2 <- as.data.frame(cbind(eig_first,
                                        eig_second,
                                        eig_third),
                                  stringsAsFactors =
                                     FALSE)
dim(eigenvalues_df_2)
colnames(eigenvalues_df_2) <- c("Lambda1","Lambda2","
   Lambda3")

# Sum of the first three eigenvalues on each bootstrap
   sample
```

```
sum_first_eig <- apply(eigenvalues_df_2, 1, sum)

# Ratio between first three eigenvals vs all eigenvals
explained <- data.frame(Var_Explained = sum_first_eig/eig
   _sum_all)
head(explained)
summary(explained)
round(quantile(explained[,"Var_Explained"],c(0.025,0.975)
   ),3)
round(sd(explained[,"Var_Explained"]),3)

# Kernel estimated density of the variance explained
# by the first three eigenvals
ggplot(explained, aes(x=Var_Explained, y=..density..)) +
  geom_histogram(bins = 50, colour = "grey60", fill =
     cols[2]) +
  geom_density() +
  theme_light() +
  xlab(expression((lambda[1]+lambda[2]+lambda[3]) /
                    (lambda[1]+lambda[2]+...+lambda[11]))
                      ) +
  theme(axis.text.x = element_text(size = rel(1.25)),
        axis.title.x = element_text(size = rel(1.25))) +
  ggtitle("Variance␣Explained␣by␣the␣first␣three␣PCs␣on␣
     10.000␣Bootstrap␣samples")

# Bootstrap loadings of the first 3 eigenvectors
# Create a list that contains the first three
   eigenvectors for each bootstrap sample
# We take them from the original list that contained
   everything, named E
# Index traial
E[[1]][[2]][,c(1,2,3)]
loadings_eig = vector(mode = "list", length = reps)
length(loadings_eig)
for(i in 1:length(E)){
  loadings_eig[[i]] <- E[[i]][[2]][,c(1,2,3)]
}
```

```
# Now we create PC1 which is a matrix with:
# 11 columns corresponding to the numb. of loadings of
   the FIRST eigenvector
# 10.000 rows corresponding to the estimated FIRST
   eigenvector on the i-th
# bootstrap sample, with i = 1,...,10.000
# In few, words we organize each bootstrap first
   eigenvector as it was transposed
w1 <- matrix(ncol = 11, nrow = reps)
# Same as above but for the SECOND eigenvector
w2 <- matrix(ncol = 11, nrow = reps)
# Same as above but for the THIRD eigenvector
w3 <- matrix(ncol = 11, nrow = reps)

# Populate PC1, PC2, PC3
for(i in 1:reps){
  w1[i,] <- unlist(loadings_eig[[i]][,1])
  w2[i,] <- unlist(loadings_eig[[i]][,2])
  w3[i,] <- unlist(loadings_eig[[i]][,3])
}

# Now we stack PC1, PC2, PC3 together by row for ggplot2
# The first columns contains the first loading on PC1,
   PC2, PC3,
# The second column contains the second loading on PC1,
   PC2, PC3
# and so on.
PCs_boot <- as.data.frame(rbind(w1,w2,w3))
dim(PCs_boot)
head(PCs_boot)

# We rename the 11 columns of PCs_boot
# in order to identify the loadings of the first three
   eigenvectors
colnames(PCs_boot) <- Mat_labs

# Stack PCs_boot
# The rows of PCs_boots are stacked in column one over
   the other
```

```
A <- as.data.frame(stack(PCs_boot))

# Associate correctly the labels PC1, PC2, PC3 for
   faceting
# In PCs_labs we 330.000 rows which corresponds to 11
# groups of rows with each PC1*10k PC2*10k PC3*10k
PCs_labs_330.000 <- vector(mode="list", length = 11)
for(i in 1:11){
  PCs_labs_330.000[[i]] <- c(rep("w1",10000),
                             rep("w2",10000),
                             rep("w3",10000))
}
LabelsBoot <- unlist(PCs_labs_330.000)

# Associate the labs to each each loading of each
   eigenvector
A <- cbind(A,LabelsBoot)

# Facet graph of bootstrapped loadings of first 3
   eigenvectors
# Estimated using Centered Percentage Changes
# (It might takes some time)
ggplot(A,aes(x=values, fill = LabelsBoot)) +
  geom_histogram(bins=500) +
  facet_grid(ind~LabelsBoot) +
  theme_light() +
  guides(fill=FALSE) +
  theme(panel.grid.minor = element_blank(),
        strip.background = element_rect(fill = "#eeeeee",
                                        colour="grey"),
        strip.text = element_text(colour="black"),
        plot.title = element_text(hjust=0.5)) +
  labs(x="Value␣of␣the␣Loadings", y="Frequency")
```

Appendix C

Yield to Maturity Calculator in Python

```
"""
This a Python Class Object which can be used to instanciate coupon
                                    bonds given the following
                                    parameters:
face value in dollars, maturity in years, annual coupon rate in
                                    decimal points and trading price
                                    in dollars.
It provides methods that will return financial informations
                                    regarding the bond of interest.
                                    In particular, it calculates the
                                    following:
- yield to maturity using the bisection search algorithm
- present value at time t=0 of each coupon paid yearly
- interest-on-interest under the assumption of yearly yield to
                                    maturity as reinvestment rate
- duration of the bond
- capital gain

"""

class multipleCouponBond:
  '''Create a Bond with specifics expressed on annual basis.
    price: selling price on the primary market.
    par_value: nominal face value of the bond. If price = par_value
                                    , bond sold at par.
    mat: number of years until the bond matures.
    coupon_rate: percentage in decimals of the par_value.
    price: valiue paid on the primary market to purchase the bond
                                    '''

  def __init__(self, par_value, mat, coupon_rate, price):
```

```
    self.par_value = par_value
    self.mat = mat
    self.coupon_rate = coupon_rate
    self.price = price
    self.getCoupon()
    self.ytm()
    self.cashFlows()
    self.getActualPrice()
    self.getPeriods()
    self.totalCouponPayments()
    self.duration()
    self.interestOnInterest()
    self.capitalGain()

  def getPeriods(self):
    """It returns a list with the time periods"""
    self.periods = [t for t in range(1,self.mat+1)]
    return self.periods

  def getCoupon(self):
    """It calculates the annual coupon"""
    self.coupon = self.par_value*self.coupon_rate
    return self.coupon

  def ytm(self, r_low = 0.00, r_high = 100.00, epsilon = 0.01):
    """Bisection search is used to find very accurate approximation
                                    of the yield to maturity of the
                                   bond"""
    r = (r_high + r_low)/2
    # Calculate the present value of coupons
    cash_flows = [(self.coupon/((1+r)**t)) for t in range(1, self.
                                   mat+1)]
    pv_c = sum(cash_flows)
    # Calculate the present value of par value
    pv_par = self.par_value/((1+r)**(self.mat))
    # Bond value
    pv = pv_c + pv_par

    while abs(pv - self.price) > epsilon:
      if pv < self.price: # r should decrease
        r_high = r
      else: # r should decrease
        r_low = r
      # update r
      r = (r_high + r_low)/2
      # update pv
```

```
    cash_flows = [(self.coupon/((1+r)**t)) for t in range(1, self
                                .mat+1)]
    pv_c = sum(cash_flows)
    pv_par = self.par_value/((1+r)**(self.mat))
    pv = pv_c + pv_par

  self.results = {'ytm':r, 'pv':pv, 'pv_c':pv_c, 'pv_par':pv_par,
                                 'cash_flows':cash_flows}
  return self.results['ytm']

def getActualPrice(self):
  """It returns the present value the bond taking into account
                                for the last period both the
                                coupon and face value"""
  return round(self.results['pv'],2)

def actualParValue(self):
  """It returns the present value of the final payback on the
                                capital invested"""
  return round(self.results['pv_par'],2)

def cashFlows(self):
  """It returns a list with the present value at time t=0 of the
                                coupons payments"""
  return self.results['cash_flows']

def duration(self):
  """It calculates the duration of the bond"""
  self.d_num = [p*cf for p,cf in zip(self.periods, self.results['
                                cash_flows'])]
  self.d = sum(self.d_num)/self.results['pv']
  return round(self.d,2)

def totalCouponPayments(self):
  """It calculates the total coupon payments of the bond"""
  self.tot_coup_paym = self.mat*self.coupon
  return round(self.tot_coup_paym,2)

def interestOnInterest(self):
  """It calculates the interest on interest of the bond"""
  self.int_on_int = ((self.coupon)* ((((1 + self.results['ytm'])
                                **(self.mat) - 1))/(self.results[
                                'ytm']))) - (self.tot_coup_paym)
  return round(self.int_on_int,2)

def capitalGain(self):
  """It calculates the capital gain of the bond"""
```

```
    self.capital_gain = self.par_value - self.price
    return self.capital_gain


if __name__ == '__main__':

  bond1 = multipleCouponBond(par_value = 1000, mat = 30,
                                  coupon_rate = 0.08, price = 1000)
  print('ytm:', bond1.ytm())
  print('coupon value:', bond1.getCoupon())
  print('periods:', bond1.getPeriods())
  print('duration:', bond1.duration())
  print('coupon cash flows:', bond1.cashFlows())
  print('present value:', bond1.getActualPrice())
  print('present value of par value:', bond1.actualParValue())
  print('total coupon payments:', bond1.totalCouponPayments())
  print('interest on interest:', bond1.interestOnInterest())
```

Appendix D

Python Code: Bonds Database Composition

```
bonds = {
  'name' : list(),
  'mrkt_price' : list(),
  'cp_rate' : list(),
  'ytm' : list(),
  'isn' : list(),
  'issue_price' : list(),
  'issue_date': list(),
  'face_val' : list(),
  'maturity' : list(),
  'coupon_pymt_date' : list(),
  'numb_payments' : list(),
  'startcoupon_date' : list(),
  'finalcoupon_date' : list()
}

active = True

while active:
  message = input("\nDO U WANT TO CONTINUE? (yes/no): ")
  if message == 'no':
    active = False
  else:
    name = str(input("\nBOND NAME: "))
    bonds['name'].append(name)

    price = float(input("\nMARKET PRICE: "))
    bonds['mrkt_price'].append(price)

    coupon_rate = float(input("\nCOUPON RATE (1.75%): "))
```

```
    bonds['cp_rate'].append(coupon_rate)

    ytm = float(input("\nYTM (1.57%): "))
    bonds['ytm'].append(ytm)

    isn = str(input("\nISN: "))
    bonds['isn'].append(isn)

    issue_price = float(input("\nISSUE PRICE: "))
    bonds['issue_price'].append(issue_price)

    issue_date = str(input("\nISSUE DATE: "))
    bonds['issue_date'].append(issue_date)

    face = float(input("\nDENOMINATION FACEVALUE: "))
    bonds['face_val'].append(face)

    maturity = str(input("\nMATURITY date (MM-DD-YR): "))
    bonds['maturity'].append(maturity)

    coupon_pymt_date = str(input("\nCOUPON PAYMENT DATE: "))
    bonds['coupon_pymt_date'].append(coupon_pymt_date)

    numb_payments = str(input("\nNUMB OF PAYMs X YR: "))
    bonds['numb_payments'].append(numb_payments)

    startcoupon_date = str(input("\nCOUPON START DATE (MM-DD-YR): "
                                 ))
    bonds['startcoupon_date'].append(startcoupon_date)

    finalcoupon_date = str(input("\nFINAL COUPON DATE (MM-DD-YR): "
                                 ))
    bonds['finalcoupon_date'].append(finalcoupon_date)


print(bonds)

file = open("bonds_data","w")
file.write(str(bonds))
file.close()


import pandas as pd
with open("bonds_data.txt") as bonds:
    data = bonds.read()
data = eval(data)
data = pd.DataFrame(data)
```

```
bonds.df.rename(columns=str.upper,inplace = True)
bonds.df
```

Bibliography

Alexander, C. (2008a). *Practical financial econometrics* (Vol. 2). Wiley.

Alexander, C. (2008b). *Pricing, hedging and trading financial instruments* (Vol. 3). Wiley.

Alexander, C. (2008c). *Quantitative methods in finance* (Vol. 1). Wiley.

Anderson, T. W. (1963). Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, *34*(1), 122–148.

Barber, J. R., & Copper, M. L. (2012). Principal component analysis of yield curve movements. *Journal of Economics and Finance*, *36*(3), 750–756.

Bodie, Z., Kane, A., & Marcus, A. J. (2014). *Investments* (Tenth). McGraw-Hill.

Choudhry, M. (2004). *Analysing and interpreting the yield curve* (Sixth). Wiley.

Connor, G. (1995). The three types of factor models: A comparison of their explanatory power. *Financial Analysts Journal*, *51*(3), 41–43.

de Prado, M. L. (2018). *Advances in financial machine learning*. Wiley.

de Prado, M. L. (2019). Beyond econometrics: A roadmap towards financial machine learning.

Efron, B., & Hastie, T. (2016). *Computer age statistical inference*. Cambridge University Press.

Embrechts, P., McNeil, A. J., & Frey, R. (2015). *Quantitative risk management: Concepts, techniques and tools*. Princeton University Press.

Fabozzi, F. J. (2012). *The handbook of fixed income securities* (Eighth). McGraw-Hill.

Fabozzi, F. J. (2013). *Bond markets, analysis, and strategies* (Eight). Pearson.

Fabrizi, P. L. (2016). *Economia del mercato mobiliare* (Sixth). Egea.

Gentle, J. E. (2017). *Matrix algebra. theory, computations and applications in statistics* (Second). Springer.

Guttag, J. V. (2017). *Introduction to computation and programming using python* (Second). The MIT Press.

Hastie, T., James, G., Witten, D., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in r*. Springer.

Hull, J. C. (2018). *Risk management and financial institutions* (Fifth). Wiley.

Johnson, R. A., & Wichern, D. W. (2014). *Applied multivariate statistical analysis* (Sixth). Pearson.

Jolliffe, I. (2002). *Principal component analysis* (Second). Springer.

Jones, F. J. (1991). Yield curve strategies. *The Journal of Fixed Income*, *1*(2), 43–48.

Litterman, R., & Scheinkman, J. (1991). Common factors affecting bond returns. *The Journal of Fixed Income*, *1*(1), 54–61.

Mishkin, F. S., & Eakins, S. G. (2018). *Financial markets and institutions* (Ninth). Pearson.

Pearson, K. (1901). On lines and planes of closest fit to system of points in space. *Philosophical Magazine*, *2*(11), 559–572.

Rao, R. (1964). The use and interpretation of principal component analysis in applied research. *The Indian Journal of Statistics*, *26*(4), 329–358.

Redfern, D., & McLean, D. (2004). *Principal component analysis for yield curve modelling* (tech. rep.). Moody's Analytics Research.

Sironi, A., & Resti, A. (2007). *Risk management and shareholders' value in banking*. Wiley.

Stefani, S., Torriero, A., & Zamburno, G. (2011). *Elementi matematica finanziaria e cenni di programmazione lineare* (Fourth). G. Giappichelli Editore.

Strang, G. (2016). *Introduction to linear algebra*. Wellesey - Cambridge Press.

Tsay, R. S. (2010). *Analysis of financial time series* (Third). Wiley.