MODELLING THE TERM STRUCTURE OF INTEREST RATES FOR THREE EUROPEAN COUNTRIES

A Dissertation presented in partial fulfilment of the Requirements for the Degree of Master in Finance

Igor das Neves Mendes

Supervisor:

Prof. José Joaquim Dias Curto, Associate Professor, ISCTE Business School, Quantitative Methods Departament

Co-Supervisor

Dr. Marco Vidal, Partner at Nimble Consulting

September 2018

Acknowledgments

I wish to thank my supervisor, Professor Dias Curto, for being not only a useful guidance throughout this thesis, but also an extraordinary and caring professor during the courses I had the pleasure to be taught by him.

Secondly, I would also like to thank my colleagues from MsC in Finance, specially Fred, Marisa, Miguel and Nathalie for being like a chosen family throughout this journey in Lisbon.

I also wish to thank my oldest and dearest friend Francisco, for being always kind and a brotherly figure, even when he had problems of his own. Good luck for your Master's Degree!

To my friend Diana, for always having a positive thinking and for being a company in this long journey that is writing a thesis. You will be a great pharmacist.

Finally, to Mom, Dad, Edgar and "Avó Zé", for being always a reference and a support in the hardest times and without whom I would not get this far. Thank you for being there for me since I can remember.

To anyone who, somehow, helped me during the writing of this thesis, a sincere and heartfelt thanks.

Abstract

It is extremely important to understand the inner workings of debt markets, since they involve interesting and complex financial concepts. Moreover, the concepts related to government debt issued in form of bonds have a series of characteristics that might impact the average person reality, for which it should be completely understood.

Having the previous motivations in mind, in this thesis, concepts related to the correct estimation of the zero-coupon yield curves (or term structure of interest rates) will be deeply analysed. In order to do so, several relevant models, from the one of Nelson and Siegel (which was the basis for several other more complete models), to the widely currently applied Adjusted Svensson model proposed by De Pooter will be discussed and applied.

In the end of all the applications of the different exposed models, a comparison between the criteria resulting from such applications is performed, in order to assess which model fits in a more accurate way into the dataset used in this thesis.

Finally, the relationship between risk and return will also be approached in a bond market context, as well as the connection between this relationship and the different risk profiles that an investor may present.

Keywords: Bonds; Credit Rating; Nelson and Siegel; Term Structure of Interest Rates

JEL Classification: C, G.

Resumo

É extremamente importante compreender o funcionamento dos mercados de dívida, uma vez que esses mercados envolvem conceitos financeiros interessantes e complexos. Para além disso, os conceitos associados à dívida pública emitida sob a forma de obrigações compreendem um conjunto de caraterísticas que podem ter impacto na realidade de cada pessoa no geral, pelo o qual devem ser totalmente compreendidos.

Tendo estas motivações em mente, no decorrer desta tese, conceitos relacionados com a estimação correta da estrutura temporal das taxas de juro serão profundamente analisados, através da apresentação e aplicação de um conjunto de modelos relevantes para o efeito, desde o modelo proposto por Nelson and Siegel (sendo este a base para uma série de outros modelos mais completos), até ao modelo ajustado de Svensson, proposto por De Pooter, que é vastamente aplicado na atualidade.

Após a aplicação dos diferentes modelos expostos, é feita uma comparação entre os resultados de cada aplicação, com vista a determinar qual o modelo que se adequa de uma forma mais precisa nos dados utilizados.

Por fim, a relação entre risco e retorno será também abordada num contexto de mercado de obrigações, assim como a ligação entre essa relação com os diferentes perfis de risco que um investidor pode apresentar.

Keywords: Obrigações; Avaliação de Crédito; Nelson e Siegel; Estrutura Temporal das Taxas de Juro;

JEL Classification: C, G.

Table of Contents

1. Introduction 1

2. Literature Review 3

2.1. Bonds 3

2.2. Curves 4

2.2.1. Forward rate curve 4

2.2.2. Spot rate curve 5

2.3. Bootstrapping 5

2.4. Models Description 6

2.4.1. Nelson and Siegel (1987) 6

2.4.2. Svensson (1994) 7

2.4.3. Diebold and Li (2006) 8

2.4.4. Adjusted Svensson (2007) 11

2.4.5. McCulloch Splines 12

2.5. Selection Criteria 13

2.5.1. Average Absolute Mean Error 13

2.5.2. Root Mean Square Error 14

2.6. Previous Studies 15

3. Methodology 16

3.1. Software 16

3.2. Data 17

4. Results 19

4.1. Outliers 19

4.1.2. Germany 19

4.1.2. Spain 21

4.1.3. Poland 22

4.2. Germany 23
4.2.1. Nelson and Siegel Model Application 23
4.2.2. Svensson and Adjusted Svensson Models Application 25
4.2.3. Diebold and Li Model Application 26
4.2.4. McCulloch Model Aplication 28
4.3. Spain 31
4.3.1. Nelson and Siegel Model Application 31
4.3.2. Svensson and Adjusted Svensson Models Application 33
4.3.3. Diebold and Li Model Application 35
4.3.4 McCulloch Model Application 36
4.4. Poland 38
4.4.1. Nelson and Siegel Model Application 38
4.4.2. Svensson and Adjusted Svensson Models Application 40
4.4.3. Diebold and Li Model Application 41
4.4.4. McCulloch Model Application 43
5. Discussion of Results 45
6. Conclusion 49
7. Bibliography 50
8. Appendix 52
8.1. Appendix A - Plots 52
8.2. Appendix B – Data 58
8.3. Appendix C – R/RStudio Scripts 66

List of Tables

Table 1 - Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the German sub-sample. 20

Table 2 - Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the Spanish sub-sample 21

Table 3 - Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the Polish sub-sample. 23

Table 4 - Criteria resulting from the Application of the Regular (left) and Edited (right) functions of NS model to the German data. 23

Table 5 - Estimated parameters for the application of the Edited NS function to the German data. 24

Table 6 - Criteria resulting from the Application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the Application of the Regular (lower left) and Edited (lower right) functions of ASV to the German data. 25

Table 7 - Estimated parameters for the application of the Regular ASV function to the German data. 25

Table 8 - Criteria resulting from the Application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the German sample. 27

Table 9 - Estimated parameters for the application of the Edited DL function to the German data. 28

Table 10 - Knot Points for the application of McC methodology to the German data. 29

Table 11 - Estimated parameters for the application of McC model to the German data and the respective t-test values. 30

Table 12 - Criteria resulting from the application of the McC methodology to the German sample. 31

Table 13 - Criteria resulting from the Application of the Regular (left) and Edited (right) functions of NS model to the Spanish data. 31

Table 14 - Estimated parameters for the application of the Edited NS function to the Spanish data. 33

Table 15 - Criteria resulting from the Application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the Application of the Regular (lower left) and Edited (lower right) functions of ASV to the Spanish data. 34

Table 16 - Estimated parameters for the application of the Regular SV function to the Spanish data. 34
Table 17- Criteria resulting from the application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the Spanish sample. 35
Table 18 - Estimated parameters for the application of the Edited DL function to the Spanish data. 36
Table 19 - Estimated parameters for the application of McC model to the Spanish data and the respective t-test values 38
Table 20 - Criteria resulting from the application of the McC methodology to the Spanish sample. 38
Table 21 - Criteria resulting from the application of the Regular (left) and Edited (right) functions of NS model to the Polish data. 38
Table 22 - Estimated parameters for the application of the Regular (left) and Edited(right) NS function to the Polish data. 39
Table 23 - Criteria resulting from the application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the application of the Regular (lower left) and Edited (lower right) functions of ASV to the Polish data. 40
Table 24 - Estimated parameters for the application of the regular SV function to the Polish sample. 41
Table 25 - Criteria resulting from the application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the Polish sample. 42
Table 26 - Estimated parameters for the application of the Edited DL function to the Polish data. 42
Table 27- Estimated parameters for the application of McC model to the Polish data and the respective t-test values 44
Table 28 -Criteria resulting from the application of the McC methodology to the Polish sample. 44
Table 29 - Summary of the criteria resulting from all the different estimations. 45
Table 30 - Moody's Rating Scale (Souce: Moody's website) 47
Table 31 - Interest Rates for different maturities determined by applying the optimal model for each country. 48
Table 32 - German bonds data 58

Table 33 - German bonds data (cont.) 59
Table 34 - German bonds data (cont.) 59
Table 35 - German bonds data (cont.) 60
Table 36 - German bonds data (cont.) 60
Table 37 - German bonds data (cont.) 61
Table 38 - Spanish bonds data 61
Table 39 - Spanish bonds data (cont.)... 62
Table 40 - Spanish bonds data (cont.)... 62
Table 41 - Spanish bonds data (cont.)... 63
Table 42 - Spanish bonds data (cont.)... 63
Table 43 - Polish bonds data... 64
Table 44 - Polish bonds data (cont.) 64
Table 45- Polish bonds data (cont.) 65

List of Plots

Plot 1 - Yield Errors resulting from the NS model application to the German data 20
Plot 2 - Yield Errors resulting from the NS model application to the Spanish data 21
Plot 3 - Yield Errors resulting from the NS model application to the Polish data......... 22
Plot 4 - Comparison between the curves resulting from the application of the Regular (upper) and the Edited (lower) NS functions to the Spanish data. 32
Plot 5 - Graphical Representation of the curve resulting from the application of the Edited NS function to the German sample. 52
Plot 6 - Graphical Representation of the curve resulting from the application of the Regular ASV function to the German sample. 52
Plot 7 - Graphical Representation of the curve resulting from the application of the Edited DL function with a λ= 0,0417 to the German sample. 53
Plot 8 - Graphical Representation of the curve resulting from the application of the CS methodology to the German sample. 53
Plot 9 - Graphical Representation of the curve resulting from the application of the Edited NS function to the Spanish sample. 54
Plot 10 - Graphical Representation of the curve resulting from the application of the Regular SV function to the Spanish sample. 54
Plot 11- Graphical Representation of the curve resulting from the application of the Edited DL function with a λ = 0,0298 to the Spanish sample. 55
Plot 12 - Graphical Representation of the curve resulting from the application of the CS methodology to the Spanish sample. 55
Plot 13 - Graphical Representation of the curve resulting from the application of the Regular NS function to the Polish sample. 56
Plot 14 - Graphical Representation of the curve resulting from the application of the Regular SV function to the Polish sample. 56
Plot 15 - Graphical Representation of the curve resulting from the application of the Edited DL function with a λ = 0,05 to the Polish sample 57
Plot 16 - Graphical Representation of the curve resulting from the application of the CS methodology to the Polish sample. 57

List of Abbreviations

AABSE – Average Absolute Mean Error

ASV – Adjusted Svensson

CF – Cash Flow

CS – Cubic Splines

DL – Diebold and Li

FV – Face Value

IR – Interest Rate

ISIN – International Securities Identification Number

MAE – Mean Absolute Error

McC – McCulloch

NS – Nelson and Siegel

PV – Present Value

RMSE – Root Mean Square Error

SV – Svensson

TSIR – Term Structure of Interest Rates

YTM – Yield to Maturity

ZCYC – Zero-coupon Yield Curve

1. Introduction

It was quite high the impact on the public opinion when, in July 2011, Moody's Rating Agency decided to downgrade the Portuguese sovereign debt to the so-called "junk" category. In fact, and despite not knowing exactly the reasons and implications of such decision, the average Portuguese citizen developed an extremely negative opinion towards Moody's, rather than trying to understand the motives that led the agency to perform such downgrade.

In order to understand the implications presented above, it is crucial to understand that investors' main objective, as rational and informed individuals, is to maximize their wealth (and therefore the return on their investments) and, at the same time, minimize the risk inherent to those same investments. Nevertheless, different investors may have different perspectives regarding the risk associated with different class of assets. Risk averse investors prefer safer investments, even though that preference may imply a lower return while, on the other hand, risk taker investors are willing to invest in more risky assets, looking for a higher return.

When investing in bonds or any other kind of debt, the most important associated risk is related to the eventual incapacity of the issuer to meet its financial obligations on time. Therefore, on July 2011, Moody's had gathered enough evidence that indicated a substantial increase of the probability that the Portuguese Republic would default towards the buyers of their government bonds which, consequently, provoked the downgrade itself.

Since issuing debt is commonly practiced by governments to finance their current expenses, Moody's decision caused several political consequences, which had an ominous impact on the average Portuguese citizen.

The aforementioned case emphasizes the importance of a deep understanding regarding debt matters or, more precisely, government debt issued as bonds, once these matters may have a direct impact on everyone's surrounding environment.

Therefore, the present thesis will focus firstly on how zero-coupon yield curves are estimated. In order to do so, several models will be presented, as well as a brief discussion regarding their advantages and disadvantages. Such models will later be applied to a dataset that includes observations from three distinct European countries. Moreover, and in order to find the optimal model to meet this purpose, a reliable and unbiased comparison will be performed among the outcomes resulting from the application of each model to the observations from each country.

Secondly, the relationship described before between risk and return of an investment will also be approached. From the obtained results, I intend to check whether this relationship currently exists, by assessing if it holds for the used data. By doing so, it will be possible to conclude if government bonds carrying a theoretical higher risk also carry a theoretical higher return. For this purpose, I decided to apply Moody's ratings for the three countries in spite of any other rating agency.

In the end, I intend to, not only deepen my knowledge regarding the aforementioned matters, but also to achieve significant and relevant results.

2. Literature Review

2.1. Bonds

A bond is a fixed-income security in which the issuer/borrower (usually a company or a sovereign government) borrows funds from an investor/lender for a determined period of time, normally being obliged to repay the face value (FV) of the bond in the end of such period. Being a security, every bond is identified by a specific code composed by 12 characters – the International Securities Identification Number (ISIN), which provides an easier international identification.

Since bonds are a form of debt, they usually are rated based on the risk that the issuer does not meet their payment obligations – credit risk. This rating is attributed by credit risk agencies (such as Moody's and Standard & Poor's, among others) that analyse the likelihood of default of the bond issuer. The higher is the attributed rating, the better is the probability that the issuer meets every obligation.

When investors buy a given bond, they look for a rate of return that meets their expectations. This rate of return corresponds to the Yield to Maturity (YTM), which can be seen as the expected return in a bond investment, if the investor decides to keep the bond until it matures. The YTM is calculated by applying the bond pricing formula described later to the observed market prices of any bond.

There are two important dates to take into consideration when analysing bonds: the issue date – date at which the bond is issued and thus can start to be traded in the market- and the maturity date – date at which the issuer must repay the principal borrowed from the investor.

Furthermore, bonds can be segmented in two different groups: coupon bonds and zero-coupon bonds. The first group contains all the bonds that, during their lifetime, pay periodic (usually yearly or half-yearly) amounts to the investor, i.e. coupons. Such payments can be fixed throughout time (following a fixed coupon rate, expressed as a percentage of the bond's face value) or they can be attached to a given benchmark (floating-coupon rate), thus changing its value during the bond's lifetime. On the other hand, zero-coupon bonds do not pay any kind of interest until the maturity date, when the principal redemption occurs.

Like any other security, bonds are priced in order to be traded in the market. The fair value formula of a given a bond, maturing at M, with a YTM equal to y, that pays a coupon C throughout its lifetime and with a face value equal to FV is given by[1]:

$$P = \sum_{m=1}^{M} C\, e^{-(m \times y)} + FVe^{-(M \times y)} \tag{1}$$

Another relevant feature regarding only coupon bonds is the concept of accrued interest. When purchasing a bond, the buyer is obtaining the right to receive all future coupon payments until the maturity date. However, if such purchase occurs between two coupon payment dates, the bond seller will not have the right to receive the next coupon payment, even though the bond was in their possession for a fraction of such period. The accrued interest corresponds to this amount of interest that the bond seller earned but will not collect. The value for the accrued interests can be calculated by using the following expression:

$$a = \frac{\text{number of days after the last coupon payment}}{365} \times c \tag{2}$$

where c corresponds to the value of the coupon payment. The price that the buyer will pay to the seller, i.e. the dirty price, is obtained by summing the observed market price (or clean price) and the accrued interest, so that the seller can receive the portion of interest associated with the period in which the bond was in their possession.

2.2. Curves

2.2.1. Forward rate curve

A forward rate, f (t, t*), is the interest rate (IR) underlying a future investment starting t periods in the future and maturing t* periods after the starting date (with t* > t). A forward rate curve is a curve that relates the forward rates as a function of maturity.

An important concept regarding forward rates is the concept of instantaneous forward rate. The instantaneous forward rate corresponds to a forward rate that has a maturity close to zero, i.e. when $t^* \rightarrow t$. Formalizing, the instantaneous forward rate, f(t), is given by:

[1] I will always assume a continuous compounding throughout this thesis.

$$f(t) = \lim_{t^* \to t} f(t, t^*) \tag{3}$$

This concept is the starting point for several models that will be discussed later in this thesis.

2.2.2. Spot rate curve

The term structure of interest rates (TSIR) depicts the relationship between interest rates and time to maturity. This relationship is measured by means of the zero-coupon yield curve (ZCYC), also known as spot rate curve.

The spot rate curve provides the YTM for a bond that only has a single CF at a given maturity, i.e. zero-coupon bond. This curve is constructed by observing the prices and plotting the yields from all the zero-coupon bonds traded on the market.

However, in the vast majority of cases, the zero-coupon bond market is quite illiquid (specially for longer maturities) which complicates the direct observation and consequent estimation of the spot rate curve. In order to overcome this complication, there are several methods that allow to determine the (theoretical) ZCYC by using coupon bonds.

2.3. Bootstrapping

Bootstrapping is a procedure firstly presented by Efron(1979) and widely applied to bonds by Fama and Bliss (1987) that allows the conversion of the YTM corresponding to a given coupon bond into zero-coupon yields. This method consists in decomposing coupon bonds with long maturities in several one-year zero-coupon instruments. In order to fully explain the method, consider a hypothetical example where there are two bonds being traded: bond A, maturing in one year, that does not pay any coupon until then, with a price of 95 and bond B, maturing in two years, with a coupon rate of 5% and a price of 102, both with a FV equal to 100.

From the first bond and having in mind the pricing formula of any bond, it is easy to calculate the zero-coupon yield, as it follows: $95 = 100\ e^{-(1 \times y_{1y})} \Leftrightarrow y_{1y} = 5{,}129\%$. Once the one-year YTM for bond A is estimated, the next step consists on decomposing bond B into two hypothetical zero-coupon instruments. The first instrument, maturing in one year, has only one CF equal to the first CF paid by bond B (100 x 5% = 5), while the second instrument, maturing in two years, has also only one CF, equal to 105 (principal redemption + coupon).

Since the one-year zero-coupon yield is already estimated, the CF of the first instrument can be discounted at such rate, obtaining the Present Value: PV (1^{st} instrument) = 5 $e^{-(1 \times 5{,}129\%)}$= 4,75. Given the PV of the first zero-coupon instrument, the price of the second zero-coupon instrument is given by simply subtracting the former value from the observed price for the 2-year coupon bond, resulting in: PV(2^{nd} instrument) = 102 – 4,75 = 97,25. Finally, since both the PV and the FV of the second hypothetical zero-coupon instrument are determined, it is possible to determine the theoretical 2-year zero-coupon yield, as it follows: $97{,}25=105e^{-(2 \times y_{2y})} \Leftrightarrow y_{2y}=3{,}834\%$.

Although this example was a simple demonstration of the bootstrapping methodology, it can be used for every maturity, as long as there are coupon bonds for those maturities. The following methodologies here explained can be applied either to zero-coupon bonds and coupon bonds.

2.4. Models Description

2.4.1. Nelson and Siegel (1987)

It is perfectly observable that the yield curves for IR usually tend to be monotic, humped or even S-shaped. This is one of the factors that led Nelson and Siegel (NS) (1987) to base their work on the solutions to differential equations once they can generate any of these three types of curves.

The other reason is related to the expectations theory, which states that long-term IR hold a reliable forecast for short-term IR in the future. Therefore, if spot rates are generated based on differential equations, forward rates can be expressed as the solution to those differential equations, since they express the current expectation of the spot rates in the future.

Formalizing, the instantaneous forward rate for the maturity m, represented by f(m,β), can be given as the solution to a second-order differential equation, resulting in

$$f(m,\beta)=\beta_0+\beta_1 \exp\left(-\frac{m}{\tau}\right)+\beta_2\left[\left(\frac{m}{\tau}\right) \exp\left(-\frac{m}{\tau}\right)\right] \quad (4)$$

where β represents the parameter vector $\beta = (\beta_0,\beta_1,\beta_2,\tau)$. In order to reach an expression for the spot rates, $s(m,\beta)$, and given the fact that spot rates can be seen as the equally weighted average of the instantaneous forward rates

$$s(m,\beta)=\frac{1}{m}\int_0^m f(m,\beta)\ dm \tag{5}$$

the following expression for the spot rates was proposed by the authors:

$$s(m,\beta)=\beta_0+\beta_1\left[\frac{1-\exp\left(-\frac{m}{\tau}\right)}{\left(\frac{m}{\tau}\right)}\right]+\beta_2\left[\frac{1-\exp\left(-\frac{m}{\tau}\right)}{\left(\frac{m}{\tau}\right)}-\exp\left(-\frac{m}{\tau}\right)\right] \tag{6}$$

Another interesting feature of NS model is that all the four parameters estimated for the previous equation hold important information and thus can be interpreted:

- By determining the limit of the spot rate function, $\lim_{m\to\infty} s(m,\beta)$, it is easy to conclude that β_0 is the asymptotic value. In other words, when the maturity approaches a very significant value, the spot rate assumes the value of β_0. For that reason, β_0 can be classified as the long-term IR. It is assumed that IR are positive and therefore that $\beta_0 > 0$. However, currently this may not be true, since it has been possible to observe negative IR.
- The next parameter, β_1, provides information regarding the speed at which the function approaches its limit (β_0) – rate of convergence. Furthermore, if the value for β_1 is positive, the slope of the function towards its limit will be negative and vice versa. Moreover, by determining the limit of the spot rate function when the maturity approaches 0, $\lim_{m\to 0} s(m,\beta)$, the result is equal to $\beta_0 + \beta_1$. This value can be interpreted as the instantaneous short rate.
- The last two parameters are related to the hump or the U-shape of the spot rate curve. If β_2 (also known as the curvature factor) is positive, the curve will describe a hump. However, if β_2 is negative, the spot rate function will assume a U-shape. The value of β_2 will also affect the size of the hump/U-shape. On the other hand, τ corresponds to the point in which the hump/U-shape is located.

2.4.2. Svensson (1994)

The model developed by Svensson (SV) (1994) also starts by defining the expression for the instantaneous forward rate at maturity m, as it follows:

$$f(m,\beta)=\beta_0+\beta_1\exp\left(-\frac{m}{\tau_1}\right)+\beta_2\left[\left(\frac{m}{\tau_1}\right)\exp\left(-\frac{m}{\tau_1}\right)\right]+\beta_3\left[\left(\frac{m}{\tau_2}\right)\exp\left(-\frac{m}{\tau_2}\right)\right] \quad (7)$$

where $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \tau_1, \tau_2)$ is the set parameters to be estimated. Recalling the expression (insert nr of the expression) for the instantaneous forward rate, we may conclude that the expression proposed by SV is an extension of the one proposed by NS, by adding the fourth term to the previous ones. The author found out that by adding a second hump-shape (or U-shape) to the NS's expression, the flexibility of the model could be increased, and thus its accuracy.

Finally, and considering spot rates as the average of the instantaneous forward rates, by applying the expression (5) the following expression is determined:

$$\begin{aligned} s(m,\beta)=\beta_0+\beta_1\left[\frac{1-\exp\left(-\frac{m}{\tau_1}\right)}{\left(\frac{m}{\tau_1}\right)}\right]+\beta_2\left[\frac{1-\exp\left(-\frac{m}{\tau_1}\right)}{\left(\frac{m}{\tau_1}\right)}-\exp\left(-\frac{m}{\tau_1}\right)\right] \\ +\beta_3\left[\frac{1-\exp\left(-\frac{m}{\tau_2}\right)}{\left(\frac{m}{\tau_2}\right)}-\exp\left(-\frac{m}{\tau_2}\right)\right] \end{aligned} \quad (8)$$

Since it is an extension of the NS model, the interpretations given to the parameters before still hold. Moreover, the new two parameters introduced by SV, β_3 and τ_2, can also be interpreted. Similarly to β_2, the new parameter β_3 determines the size and the form of the second shape (if $\beta_3 > 0$ it produces a hump while if $\beta_3 < 0$ it produces a U-shape). On the other hand, while τ_1 provides information regarding the position of the first shape, τ_2 specifies the location of the second one.

It is proven that, in the vast majority of cases, introducing this fourth component in the model increases its performance in modelling the TSIR. However, this "upgrade" may have a crucial drawback while estimating the parameters – a multicollinearity problem.

2.4.3. Diebold and Li (2006)

Similarly to SV, Diebold and Li (DL) (2006) base their model on the NS's one. However, the authors noticed that estimating the vector of parameters on the original model, $\beta = (\beta_0, \beta_1, \beta_2, \tau)$, could be challenging, due to the high number of numerical optimizations to be made. Therefore, the model they propose was developed in order to simplify these operations and thus simplify the estimation itself.

The expression for the forward rate curve proposed by DL is not very different from the one proposed by an extension of the original NS, developed by Siegel and Nelson (1988):

$$f(m,\beta)=\beta_0+\beta_1 \exp(-\lambda m)+\beta_2 \lambda\exp(-\lambda m) \tag{9}$$

which, by repeating the process and apply the expression (5), results in the following expression for the yield curve

$$s(m,\beta)=\beta_0+\beta_1\left[\frac{1 - \exp(-\lambda m)}{\lambda m}\right]+\beta_2\left[\frac{1 - \exp(-\lambda m)}{\lambda m}-\exp(-\lambda m)\right] \tag{10}$$

By comparing the last expression with expression (6) one notices that the original τ was replaced by λ in a relation that can be expressed as $\lambda=\frac{1}{\tau}$. DL define this new parameter as the "exponential decay rate". When λ assumes small values, it produces a slow decay and, therefore, provides a better fit at longer maturities. On the other hand, for large values of λ, the decay happens in a faster rate, benefiting the fit of the curve at short maturities.

Nevertheless, DL still agree with NS in the fact that λ also determines the point where the loading on β_2, $\left[\frac{1 - \exp(-\lambda m)}{\lambda m}-\exp(-\lambda m)\right]$, achieves its maximum (NS defended that τ corresponds to the location of the hump)

The other significant difference between the models proposed by DL and NS is also related with τ and λ. While in the original model τ is a parameter present in the parameter vector β and therefore is estimated alongside with β_0, β_1 and β_2, this does not happen for λ in DL's model.

In fact, in this model, λ assumes a prespecified value, which is defined before the estimation of the remaining parameters. By doing this, β_0, β_1 and β_2 can now be estimated using the ordinary least squares (in the NS case, nonlinear estimation methods were required). This was the most significant contribution by this model, since instead of having four parameters to estimate, there are now only three and the estimation procedures are far easier than in the original case.

The next question that arises is to choose an acceptable value for λ. There is not a specific method to determine this value, but it must be a substantiated and rational choice.

As defended both by DL and NS, λ corresponds to the maturity at which the hump or U-shape is located (or, in other words, the maturity at which the loading on β_2 reaches its maximum). Being so, DL considered that this loading achieves its maximum point usually between two-year or three-years maturities. In consequence, by doing a simple average of this time period and obtaining the value of 30 months, the authors prespecified that $\lambda = 0{,}0609$ in their study (in other terms, when λ=0,0609, the loading on β_2 reaches its maximum point).

Finally, DL also provided an alternative interpretation for the loadings on β_0, β_1 and β_2 as three latent dynamic factors. This different insight is firstly made as a perspective of short-term, medium-term and long-term factors and then as a perspective of level, slope and curve:

- The loading on β_0 is equal to 1. Being a constant, this loading does not increase or decay in the limit. Thus, it can be interpreted as the long-term factor (this interpretation is not very different from the one provided by NS, who considered β_0 the long-term IR). By increasing β_0, all yields (regardless of their maturity) increase equally, which will change the level of the yield curve. For that reason, DL also interpreted the long-term factor as the level factor;
- The next loading, $\left[\frac{1 - \exp(-\lambda m)}{\lambda m}\right]$, is associated to β_1. It is a function that starts at its maximum (which is 1 when the maturity is very close to 0) and decays quickly towards 0. For that reason, this second loading can be viewed as a short-term factor. This factor also governs the slope of the yield curve. In order to consolidate this idea, it is clearly seen that an increase on the value of β_1 increases short-term yields in a stronger way than it increases long-term ones. By increasing short-term yields, it changes also the slope of the yield curve.
- Finally, the loading on β_2 is given by $\left[\frac{1 - \exp(-\lambda m)}{\lambda m} - \exp(-\lambda m)\right]$. For maturities very close to 0, this loading equally assumes the value of 0. However, it starts increasing when the maturity also increases until it reaches its maximum (which is determined by the value of λ, as it was explained before). It then starts decreasing towards 0 for longer maturities. For this set of reasons, DL entitled this loading as the medium-term factor or curvature factor. Performing the same reasoning as before, an increase on the value of β_2 will not have any significant impact on very short-term or very long-term yields, creating, in fact, an increase on medium-term ones. This change in medium-term yields will positively impact the yield curve curvature.

2.4.4. Adjusted Svensson (2007)

A multicollinearity problem happens when two or more independent variables in a multiple regression model are highly correlated between them. In the SV model particular case, a multicollinearity problem may occur between τ_1 and τ_2 if these two parameters assume similar values.

If this happens, the SV model is no longer a four-factor base model, but a three-factor one. Actually, it becomes very similar to the NS model, differing only in the curvature factor: while in the NS model the curvature factor is equal to β_2, if there is a multicollinearity problem in the SV model there is only one curvature factor equal to $\beta_2 + \beta_3$ (the individual parameters can no longer be estimated efficiently, only their sum).

In order to solve the multicollinearity problem that may arise in the "original" SV model, De Pooter (2007) developed and proposed an "Adjusted Svensson" (ASV) model. The author's solution was based on ensuring that the two medium-term components present in the SV model were different, even though $\tau_1 \approx \tau_2$. In order to do so, De Pooter changed the forward rate curve expression (and consequently the spot rate curve one) as it is shown above:

$$\begin{aligned} f(m,\beta)=\beta_0+\beta_1 \exp\left(-\frac{m}{\tau_1}\right)+\beta_2\left[\left(\frac{m}{\tau_1}\right)\exp\left(-\frac{m}{\tau_1}\right)\right] \\ +\beta_3\left[\exp\left(-\frac{m}{\tau_2}\right)+\left(\frac{2m}{\tau_2}-1\right)\exp\left(-\frac{2\tau}{\tau_2}\right)\right] \end{aligned} \tag{11}$$

Once again, applying expression (5), the following expression for the spot rate curve is obtained:

$$\begin{aligned} s(m,\beta)=\beta_0+\beta_1\left[\frac{1-\exp\left(-\frac{m}{\tau_1}\right)}{\left(\frac{m}{\tau_1}\right)}\right]+\beta_2\left[\frac{1-\exp\left(-\frac{m}{\tau_1}\right)}{\left(\frac{m}{\tau_1}\right)}-\exp\left(-\frac{m}{\tau_1}\right)\right] \\ +\beta_3\left[\frac{1-\exp\left(-\frac{m}{\tau_2}\right)}{\left(\frac{m}{\tau_2}\right)}-\exp\left(-\frac{2m}{\tau_2}\right)\right] \end{aligned} \tag{12}$$

By comparing this final expression for the spot rate curve proposed by De Pooter with the one proposed by SV, the only existing difference is a simple change in the second medium-term component – $\exp\left(-\frac{m}{\tau_2}\right)$ in the original SV model becomes $\exp\left(-\frac{2m}{\tau_2}\right)$ in the adjusted one.

Although this modification seems simple, it ensures that even if $\tau_1 \approx \tau_2$ the two medium-term components are different and thus it is possible to estimate efficiently the two parameters individually – the multicollinearity problem is solved. Furthermore, the adjusted component is still a medium-term one, since it continues to start at zero, increases for medium maturities and returns finally back to zero (even though, due to this adjustment, this increase and decrease happen at a faster rate).

2.4.5. McCulloch Splines

The NS model and its extensions are not the only way to estimate the TSIR. In fact, McCulloch (McC) (1971,1975) proposed a very different procedure in order to satisfy the same purposes, based on cubic splines (CS).

The first main difference between NS-based models and the one proposed by McC is that while the first ones work directly with forward rate curves and later with yield curves, the later works firstly with discount curves.

By definition, a spline function consists of polynomial pieces – basis functions – on sub-intervals joined together with certain continuity conditions through a series of knot points. McC suggested that this type of functions, more specifically cubic polynomial ones, have all the properties required to ensure the continuity and smoothness of the discount function. The author starts by defining the discount factors as

$$\delta(m,\beta)=1+\sum_{j=1}^{n}\beta_j\, g^j(m)\,, \tag{13}$$

being $g^j(m)$ ($j = 1,\ldots, n$) the set of cubic basis functions. In order to ensure the continuity and smoothness mentioned before, all the basis functions must be twice-differentiable at each point. Furthermore, the condition that $g^j(0) = 0$ must also be verified (the discount factor for any payment maturing now is always equal to 1, which forces the value of g^j to be 0). β^j represents an unknown vector parameter.

The first step in this methodology is to find out the proper number of basis functions and knot points to be used. Considering that the bonds are arranged in ascending order by their maturities m ($m_1 < m_2 < \ldots < m_k$) for a number of bonds equal to k, McC defines the number of basis functions as

$$n = \mathrm{INT}\left[\sqrt{k}+0.5\right] \tag{14}$$

which results in n -1 knot points (q_j). In order to define the different knot points, the author proposes the following expression:

$$q_j = m_h + \theta(m_{h+1} - m_h), \tag{15}$$

with h = INT $\left[\frac{(j-1)k}{n-2}\right]$ and $\theta = \frac{(j-1)k}{n-2} - h$. The previous expression only holds for $1 < j < n - 1$, since the first knot point must be equal to 0 ($q_j = 0$) and the last one must be equal to the maximum maturity present in the set of bonds ($q_{n-1} = m_k$).

Having the knot points defined, McC proposes the following set of expressions for the definition of the basis functions:

$$g^j(m) = \begin{cases} 0 & m < q_{j-1} \\ \dfrac{\left(m - q_{j-1}\right)^3}{6\left(q_j - q_{j-1}\right)} & q_{j-1} \leq m < q_j \\ \dfrac{\left(q_j - q_{j-1}\right)^2}{6} + \dfrac{\left(q_j - q_{j-1}\right)\left(m - q_j\right)}{2} + \dfrac{\left(m - q_j\right)^2}{2} - \dfrac{\left(m - q_j\right)^3}{6\left(q_{j+1} - q_j\right)} & q_j \leq m < q_{j+1} \\ \left(q_{j+1} - q_{j-1}\right)\left[\dfrac{2q_{j+1} - q_j - q_{j-1}}{6} + \dfrac{m - q_{j+1}}{2}\right] & q_{j+1} \leq m \end{cases} \tag{16}$$

There are two specific cases regarding the basis functions. Firstly, when j = n, the basis function becomes $g^j(m) = m$. Furthermore, for j = 1, it is defined that $q_{j-1} = q_j = 0$.

Finally, it is possible to convert the discount factors to spot rates. This conversion is ensured by the following expression (assuming continuous compounding):

$$s(m,\beta) = \frac{-\ln[\delta\,(m,\beta)]}{m} \tag{17}$$

2.5. Selection Criteria

Given all models presented previously, I intend to assess which model produces the best results for the observed data. Therefore, from all the criteria that can be used to compare the accuracy of different estimations, I will use two of the most applied criteria in this type of estimations.

2.5.1. Average Absolute Mean Error

Firstly, the Average Absolute Mean Error (AABSE) or Mean Absolute Error (MAE) is a metric used to assess the accuracy of a given estimation. It consists on the equally weighted average of the absolute differences between the observed (y_i) and the observed ($\hat{y}_i$) values, i.e. errors, given by:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{18}$$

where n represents the sample size.

2.5.2. Root Mean Square Error

The other criteria that will be estimated and analysed is the Root Mean Square Error (RMSE). It is the square root of the averaged squared errors, given by:

$$\text{RMSE}= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{19}$$

Regardless being two completely different criteria, both AABSE and RMSE have certain aspects in common:

- Negatively-oriented measures – the lower the errors resulting from an estimation, the better the estimation accuracy is. Therefore, since both metrics average the difference between the forecasted and the observed values, lower values for AABSE and RMSE can be interpreted as a better model accuracy. The values for both criteria can range between 0 and + ∞, where 0 would be the optimal case (the averaged errors would be equal to 0, which can be interpreted as a perfect fit of the observed values into the estimated model);
- Direction of the errors – both criteria ignore the positive or negative signs of the errors, either by using the absolute values in the AABSE case or by squaring the errors in the RMSE case;

On the other hand, the main difference between these two metrics is related to the size of the errors. By squaring the errors, the RMSE will present significantly higher values than the AABSE if there is any error of a considerable magnitude. In fact, if the value for the RMSE is abnormally higher than the value for AABSE, it might indicate the presence of, at least, one error with a very significant value, i.e. an outlier.

2.6. Previous Studies

Leañez and Moreno (2009) applied the NS, SV and Vasicek (1977) models to 10 distinct coupon bonds issued by the Venezuelan government in order to estimate the Venezuelan TSIR. The main goal of the this study was to determine which of the three applied models was more appropriate to model the Venezuelan TSIR, based on the comparison criteria resulting from each estimation.

The authors concluded that, given the collected data, the SV methodology was the one that resulted in the smallest values for the comparison criteria and thus, the best model to meet their purposes. Furthermore, and although it will not be approached, Vasicek methodology produced the worst results which is comprehensible, given its simplicity (it is a one-factor model while NS and SV are a three-factor and four-factor models, respectively).

In order to conclude regarding the adequacy of the estimation of provisions in insurance companies, Chan et al (2015) approached the Liability Adequacy Test (LAT) imposed by the Brazilian regulation. This test consists on verifying if the net carrying amount[2] is enough to support the current estimation of the eventual cash flows associated to the insurance contracts, which is discounted by the TSIR. If, when subtracting the current estimation to the net carrying amount, the difference is positive, the provisions constituted by the insurance company might not be enough to cover the eventual cash flows that may result from the insurance contracts. With the goal of estimating the most appropriate TSIR to apply in the LAT, the authors chose the Vasicek, SV and CS methodologies.

The most contributing result of this study is related to the fact that the LAT is very sensitive to the elected methodology to estimate the TSIR. In fact, when applied to the same empirical evidence, the LAT values, when the SV and CS methodologies were used to determine the current estimates, were negative (which indicates that the provisions are accurately estimated). On the other hand, for the same data, the Vasicek methodology resulted in positive values for the LAT, meaning that the value allocated to the built provisions is not enough to cover all the future expected cash flows. In the end, the choice of the most adequate model might influence the LAT results and the conclusions resulting from such test.

[2] Technical provisions deducted from deferred acquisitions expenses and related intangible assets.

3. Methodology

3.1. Software

In order to achieve the goal of this thesis, the used methodology will consist on an empirical application of the models explained and discussed in the Literature Review. Being so, for the data corresponding to each selected country, I will apply the NS, SV, ASV, DL and CS methodologies, so that it is possible to extract contributing results.

From a more practical perspective, I decided to use the R/RStudio software, once I am familiar with the program itself and it is quite an user-friendly application. After looking into several R/RStudio packages, the package "*termstrc*" was the one more eligible to meet my purposes, since it offers wide variety of functions which are very useful to the required estimation procedures.

The most valuable functions (and thus, the most applied ones) present in this package are the following:

- "*estim_nss*" – computation of the parameters for the NS class models (NS, SV, ASV and DL). The used inputs for this function were the created dataset, the country for which the parameters are being estimated, the estimation method to apply and the maturity range for the estimation, which was 30 years in every case;
- "*estim_cs*" – computation of the parameters for the MCS method. The used inputs are similar to the ones presented above (with the exception that it is not required to specify the estimation method to be applied, since the MCS is the only one available with this function);
- "*summary*" – it returns the criteria required to compare the accuracy of the models (more specifically, the AABSE and RMSE). This function requires as inputs any objected created by any of the previous two functions;
- "*plot*" – it returns the graphical representations of the curves computed. Similarly to the "*summary*" function, it requires as input any object created with one of the two first functions;

For the NS, SV, ASV and DL methodologies, the software uses a grid search[3] to determine the optimal start parameters required for the estimation. Once the start

[3] Procedure that, through cross-validating different parameters, allows to chose the optimal combination.

parameters are defined, the software adjusts such values in order to find the optimal final parameters to apply in each model.

On the other hand, for the CS methodologies, after determining the values for the knot points and defining the basis functions, the package "*termstrc*" applies a simple OLS methodology to find the optimal β vector estimates.

However, there were some alterations to be made. The NS class models have one assumption that might be incorrect for same cases, since they restrict the instantaneous short rate, i.e. the sum of β_0 and β_1, to be positive. Thus, I have created a new function, "*estim_nss.couponbonds_edit*", in which I altered the source code of the original "*estim_nss*" in order to remove that restriction.

Moreover, and since the creators of the package assumed that the ZCYC could only assume positive values, the "*plot*" functions become obsolete when the previous modification is applied. Therefore, I also changed the source code for the "*plot*" functions, allowing them to have a graphical representation even when the curve assumes negative values, and thus creating the new function "*plot.termstrc_nss_edit*".

3.2. Data

As it was stated before, my main goal is to model the TSIR of three different European countries (Germany, Spain and Poland).

For that purpose, and having in mind the required inputs for the package "*termstrc*", I collected the following information regarding 33 German bonds, 29 Spanish bonds and 18 Polish bonds:

- International Securities Identifying Number (ISIN);
- Maturity Date;
- Issue Date;
- Coupon Rate;
- Price;
- Accrued Interest;
- Observation Date;
- Cash flows for every bond until maturity;
- Maturity dates of every cash flow;

All the bond-related data was extracted from the Frankfurt Stock Exchange website (**http://en.boerse-frankfurt.de/**) on 11/06/2018, which corresponds to the observation date. The collected data is purely composed by government bonds with a fixed yearly coupon rate. All the information regarding such dataset was organized in an Excel spreadsheet and it can be consulted on Appendix B.

Before transposing the collected data into R/RStudio, I had to perform a slight transformation in the maturity, issue, cashflows and observation dates, since the two programs have different origin dates (while the origin date for Excel is 01/01/1900, the origin date for R/RStudio is 01/01/1970). Script X on Appendix C contains the commands inherent to the conversion process for every date present in the sample.

After the dates conversion, the data is prepared to be transposed into R/RStudio. The script that allows R/RStudio to read all the information required by the package "*termstrc*" can be consulted in Appendix C.

4. Results

4.1. Outliers

After some estimations, I noticed that the obtained results were showing large errors. In order to assess the possibility of the existence of outliers – observations that far from the remaining observations – I plotted the errors resulting from the NS yield estimation for the three countries. By consulting these plots, it is possible to conclude if there is any bond that could be considered an outlier.

If the existence of any outlier is found, package "*termstrc*" provides a function, "*rm_bond*" that removes all the information regarding one specific bond from the dataset. This function requires only the ISIN of the desired bond and it eases the process described below.

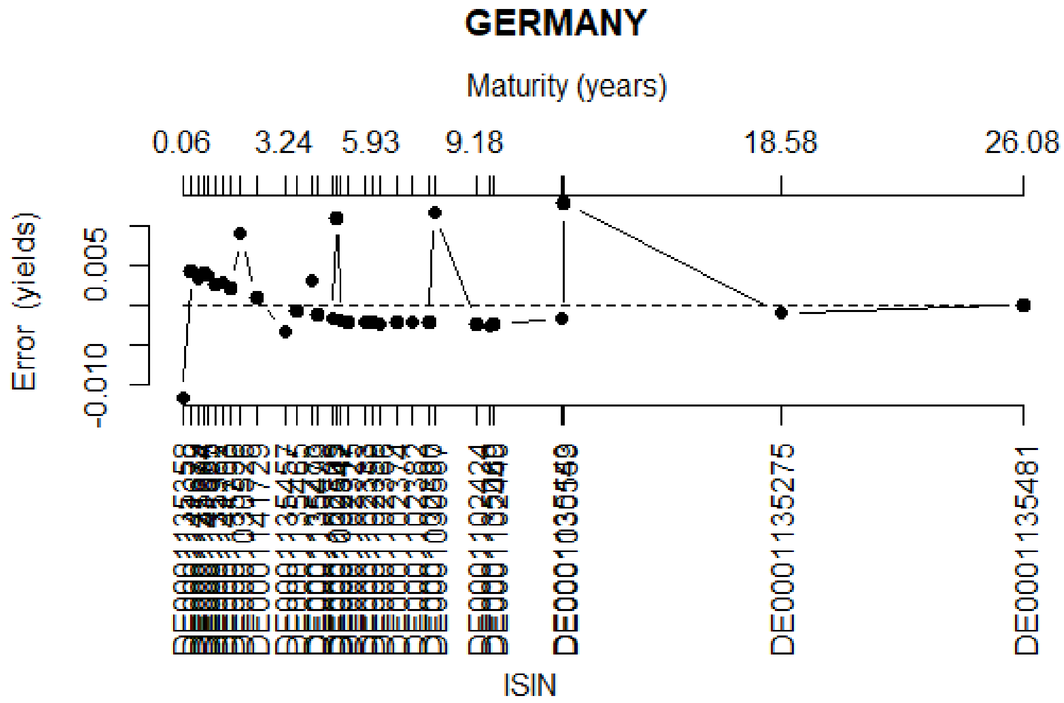
4.1.2. Germany

Plot 1 shows the errors plot for the German sub-dataset which are associated with every bond, as represented by their corresponding ISIN.

After analysing the plot, I decided to exclude from the German sub-dataset the bonds with the ISIN given by: "DE0001135358", "DE0001030559", "DE0001030567" and "DE0001030542", since they reveal the highest associated Yield errors for the NS estimation procedure.

Even though I do not include the plot for the errors associated with the remaining models, the results were similar – the four bonds with the highest associated errors were the ones excluded for the NS model.

In order to compare the results before and after the removal of the previous outlier bonds, **Table 1** shows the regular NS estimation results for the original dataset and the new one that does not contain the four bonds mentioned above.



Plot 1- Yield Errors resulting from the NS model application to the German data

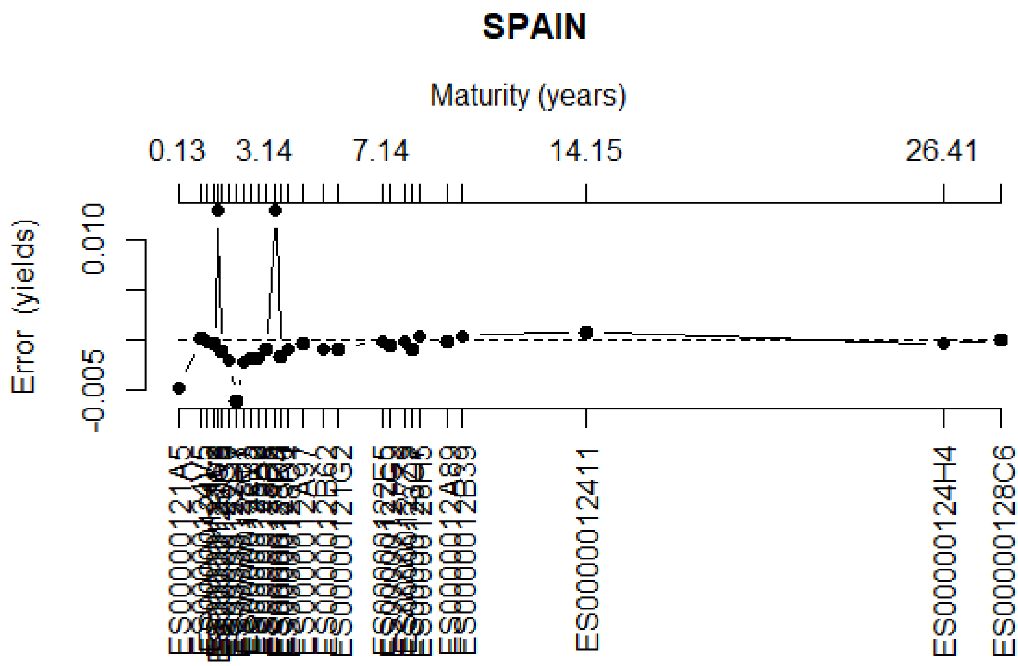
As it was expected, by removing simply four bonds from the German sub-dataset, the errors decreased drastically. Taking into consideration the errors related to the Yields, the RMSE decreased from 0,500635% to 0,177809% (which can be seen as a decrease of approximately 64,48% of the original RMSE), while the AABSE decreased from 0,370402% to 0,104588% (a decrease of approximately 71,76%).

```
> summary(NelsonSiegelGERM)                > summary(NelsonSiegelGERMOutlier)
---------------------------------------    ---------------------------------------
Goodness of fit:                           Goodness of fit:
---------------------------------------    ---------------------------------------

                    GERMANY                                    GERMANY
RMSE-Prices         3.704203               RMSE-Prices         0.514982
AABSE-Prices        2.019856               AABSE-Prices        0.346653
RMSE-Yields (in %)  0.500635               RMSE-Yields (in %)  0.177809
AABSE-Yields (in %) 0.370402               AABSE-Yields (in %) 0.104588
```

Table 1- Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the German sub-sample.

4.1.2. Spain

Repeating the outlier exclusion process performed in the previous case, **Plot 2** displays the Yield errors for the Spanish NS estimation.



Plot 2- Yield Errors resulting from the NS model application to the Spanish data

Taking into consideration that the Spanish sub-dataset has less observation when compared to the German sub-dataset, I decided to exclude only the three bonds with the highest associated errors, which have the following ISIN's: "ES000000126W8", "ES00000127H7" and "ES00000128D4".

Table 2 show the criteria resulting from the NS parameters estimation for Spain. Similarly to the German case, the accuracy of the model improved substantially after removing the three chosen outliers.

```
> summary(NelsonSiegelSPAIN)
---------------------------------------
Goodness of fit:
---------------------------------------

                     SPAIN
RMSE-Prices          1.093953
AABSE-Prices         0.618369
RMSE-Yields (in %)   0.388372
AABSE-Yields (in %)  0.197642
```

```
> summary(NelsonSiegelSPAINOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     SPAIN
RMSE-Prices          0.3693258
AABSE-Prices         0.2486826
RMSE-Yields (in %)   0.1073841
AABSE-Yields (in %)  0.0636488
```

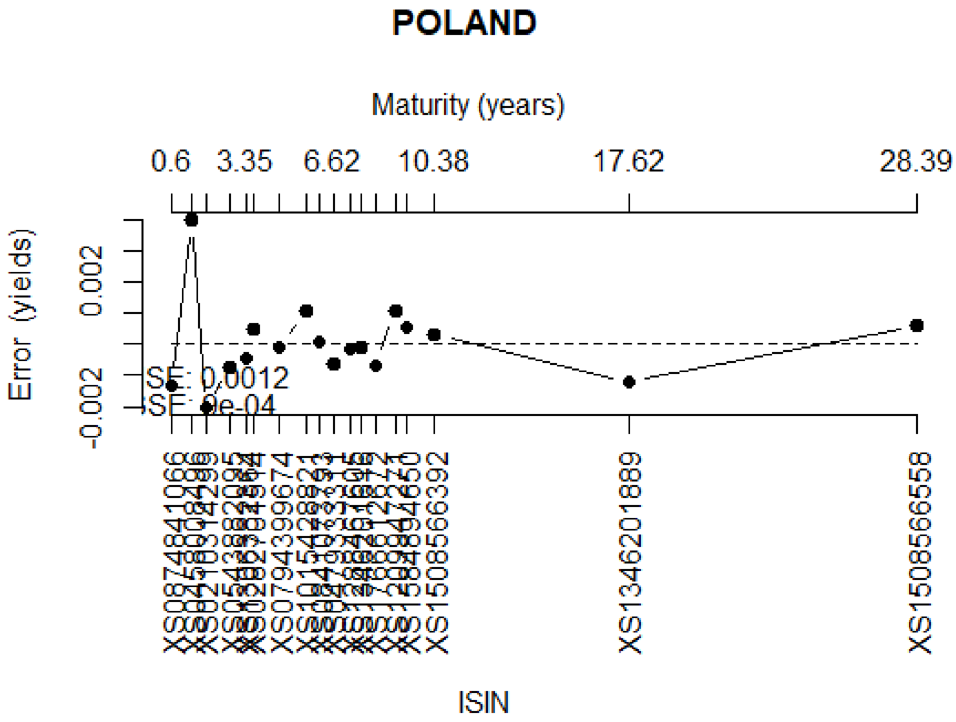
Table 2- Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the Spanish sub-sample

Analysing the criteria related to the Yields, the RMSE decreased from 0,388372% to 0,1073841% (a decrease equal to 72,35% of the original RMSE) and the AABSE decreased from 0,197642% to 0,0636488% (67,8% of the original AABSE).

4.1.3. Poland

Finally, the same procedure is applied to the Polish sub-dataset. However, and as it is possible to conclude by looking at **Plot 3**, the observations collected regarding Poland are the ones that present lower values for the yield errors associated with the NS estimation.

Given the relatively short dimension of the Polish sub-dataset and given the fact that the errors associated with the NS estimation present lower values when compared to the ones regarding Germany and Spain, I decided to exclude only the bond with the highest associated error, which is identified by "XS0458008496".



Plot 3 - Yield Errors resulting from the NS model application to the Polish data.

Regarding **Table 3**, it is curious to conclude that the Polish estimation for the NS parameters is the one that produces the best results (lowest values for the criteria), even though it is the one which contains the fewest observations. In fact, the Polish NS estimation before removing the chosen outlier produces better results than the German NS estimation after removing the four chosen outliers.

Nevertheless, by removing the bond identified by "XS0458008496", the RMSE decreases around 43,77% of its original value (from 0,1249608% to 0,0702709%) while the AABSE decreases approximately 32,87% (from 0,0864879% to 0,0580572%).

```
> summary(NelsonSiegelPOLAND)
---------------------------------------
Goodness of fit:
---------------------------------------

                     POLAND
RMSE-Prices          0.6660981
AABSE-Prices         0.4698795
RMSE-Yields (in %)   0.1249608
AABSE-Yields (in %)  0.0864879
```

```
> summary(NelsonSiegelPOLANDOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     POLAND
RMSE-Prices          0.6709467
AABSE-Prices         0.4457237
RMSE-Yields (in %)   0.0702709
AABSE-Yields (in %)  0.0580572
```

Table 3 - Comparison of the Criteria resulting from NS application before (left) and after (right) the exclusion of outliers from the Polish sub-sample.

In general, the assessment of the existence of outliers was paramount since it improved tremendously the accuracy of the estimation procedures for the different models. Thus, the results that follow are related to the new dataset in which the outliers were excluded. However, the script that allows to compute the results for the original dataset, i.e. the dataset that still contains the information regarding the outlier bonds, is also available on Appendix C.

4.2. Germany

4.2.1. Nelson and Siegel Model Application

As it was said before, for Nelson and Siegel, Svensson, Adjusted Svensson and Diebold and Li models, I used both the regular function provided by package “termstrc” and the edited one, which allows that $\beta_0 + \beta_1$ can be negative.

```
> summary(NelsonSiegelGERMOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.514982
AABSE-Prices         0.346653
RMSE-Yields (in %)   0.177809
AABSE-Yields (in %)  0.104588
```

```
> summary(NelsonSiegelGERMOutlier_edit)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.4833290
AABSE-Prices         0.2844180
RMSE-Yields (in %)   0.1636731
AABSE-Yields (in %)  0.0851209
```

Table 4- Criteria resulting from the Application of the Regular (left) and Edited (right) functions of NS model to the German data.

Table 4 shows some metrics that evaluate the accuracy of the model. In this case, since RMSE-Yields$_{\text{Regular}}$ = 0,177809% is greater than RMSE-Yields$_{\text{Edited}}$ = 0,1636731 % and AABSE-Yields$_{\text{Regular}}$ = 0,104588% is greater than AABSE-Yields$_{\text{Edited}}$ = 0,0851209% , I conclude that the edited function models the German government bond yields more accuratelly than the regular one. Thus, the Nelson and Siegel parameters that will be interpreted are the ones computed for the edited function, which are displayed in **Table 5.**

As I mentioned above, the estimate for β_0 corresponds to the long-term IR, which in this case is approximately equal to 1,65%. In other words, when m $\rightarrow \infty$, the spot rate will be equal to 1,65%.

```
> NelsonSiegelGERMOutlier_edit
---------------------------------------------------
Estimated Nelson/Siegel parameters:
---------------------------------------------------

        GERMANY
beta_0   1.64768
beta_1  -2.17407
beta_2  -3.65410
tau_1    1.99872
```

Table 5- Estimated parameters for the application of the Edited NS function to the German data.

From the negative value of β_1 estimate (-2,17%), it is possible to conclude that the slope of the function towards its limit (β_0) will be positive. Furthermore, it is also possible to determine the instantaneous short rate by summing the estimates of β_0 and β_1, which results approximately in -0,53%. The previous value indicates that, when m$\rightarrow$ 0, the spot rate is equal to -0,53%. This is a direct result of the modification of the regular function available in the package since it would not be possible to reach a negative value for the instantaneous short rate without such modification.

The last two parameters describe the behaviour of the function's shape. Given that β_2 is negative, the function will have a U-shape, which will occur approximately at m = 2, given that $\tau \approx 2$. Therefore, the function will decrease until m = 2 (where the U-shape is located) and will then start increase, reaching a maximum value of 1,65% (the value of β_0).

Given all the parameters, the following expression, attending to expression (6) models the TSIR for the German bonds:

$$s(m,\beta)=1{,}648 - 2{,}174\left[\frac{1-\exp\left(-\frac{m}{1{,}999}\right)}{\left(\frac{m}{1{,}999}\right)}\right] - 3{,}654\left[\frac{1-\exp\left(-\frac{m}{1{,}999}\right)}{\left(\frac{m}{1{,}999}\right)} - \exp\left(-\frac{m}{1{,}999}\right)\right]$$

4.2.2. Svensson and Adjusted Svensson Models Application

As I mentioned before, the Adjusted Svensson model is very similar to the original one, with the difference that it is prepared for the presence of multicollinearity. This being the case, the ASV model performs better than the SV model if multicollinearity problems may arise.

```
> summary(SvenssonGERMOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.4772284
AABSE-Prices         0.2704256
RMSE-Yields (in %)   0.1615744
AABSE-Yields (in %)  0.0853793
```

```
> summary(SvenssonGERMOutlier_edit)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.4772066
AABSE-Prices         0.2704237
RMSE-Yields (in %)   0.1616350
AABSE-Yields (in %)  0.0854414
```

```
> summary(AdjSvenssonGERMOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.4772931
AABSE-Prices         0.2699646
RMSE-Yields (in %)   0.1604473
AABSE-Yields (in %)  0.0842777
```

```
> summary(AdjSvenssonGERMOutlier_edit)
---------------------------------------
Goodness of fit:
---------------------------------------

                     GERMANY
RMSE-Prices          0.4757400
AABSE-Prices         0.2664332
RMSE-Yields (in %)   0.1620800
AABSE-Yields (in %)  0.0857496
```

Table 6- Criteria resulting from the Application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the Application of the Regular (lower left) and Edited (lower right) functions of ASV to the German data.

Having this in mind, I assume that multicollinearity problems may arise in the original SV model and that is why the ASV performs better than the original SV.

```
> AdjSvenssonGERMOutlier
---------------------------------------------------
Estimated Adj. Svensson parameters:
---------------------------------------------------

        GERMANY
beta_0    1.68369
beta_1   -1.68369
beta_2   29.01536
tau_1     1.28653
beta_3  -20.64904
tau_2     2.29961
```

Table 7- Estimated parameters for the application of the Regular ASV function to the German data.

According to the results obtained, displayed on **Table 7**, with the ASV methodology, the long-term IR will be equal to 1,68639%. Considering that $\beta_0 + \beta_1$ gives the estimated value of the instantaneous short rate, it will be approximately equal to 0, due to the symmetry of both values. It is still possible to affirm, based on the negative value of the β_1 estimate, that the slope of the function towards its limit (β_0) will be positive, just like in the NS methodology.

The first curvature of the function is a U-shape, given the positive estimate of β_2 and it occurs when m ≈ 1,29. Moreover, as it was expected, the second curvature is the opposite of the first one, i.e. a hump (which is verifiable by looking at the positive estimate for β_3) and it occurs approximately one year after the U-shape ($\tau_2 \approx 2,30$)

The expression for the ASV function, according to expression (12), that allows to model the term structure of German interest rates is given by:

$$s(m,\beta)=1,684 -1,684\left[\frac{1 - \exp\left(-\frac{m}{1,287}\right)}{\left(\frac{m}{1,287}\right)}\right]+29,015\left[\frac{1 - \exp\left(-\frac{m}{1,287}\right)}{\left(\frac{m}{1,287}\right)}- \exp\left(-\frac{m}{1,287}\right)\right]$$

$$- 20,649\left[\frac{1\text{-}\exp\left(-\frac{m}{2,2997}\right)}{\left(\frac{m}{2,2997}\right)}- \exp\left(-\frac{2m}{2,2997}\right)\right]$$

4.2.3. Diebold and Li Model Application

Once the NS model parameters are estimated (more specifically τ), DL model does not offer any substantial improvement regarding estimation accuracy, since the values resulting from the NS estimation are theoretically the optimal ones.

In order to assess if the τ determined in the NS estimation is, in fact, the optimal one, I have defined the following values for λ:

- The default value, $\lambda_{Default} = 0,0609$ which is the one prespecified by the authors in their paper. It corresponds to a value for $\tau_{Default}= \frac{1}{0,0609\times 12}=1,368$
- $\lambda_{NS/Regular}=\frac{1}{1,66311 \times 12} \approx 0,0501$ and $\lambda_{NS/Edited}= \frac{1}{1,99872 \times 12}\approx 0,0417$, where 1,66311 and 1,99872 correspond to $\tau_{NS/Regular}$ and $\tau_{NS/Edited}$, respectively. Both values for τ were converted into a monthly basis given the fact that the default value for λ was also computed on such basis by DL;

- A hypothetical value just for comparison, $\lambda_{\text{Hypothetical}} = 0{,}0285$. This value for λ corresponds to a value for $\tau_{\text{Hypothetical}} = \frac{1}{0{,}0285 \times 12} \approx 2{,}924$;

After defining all these values for λ and estimating the remaining parameters (both for the Regular and the Edited functions), the criteria for each estimation is displayed on **Table 8**.

```
> summary(DieboldLiGERMOutlierDefault)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.634795
AABSE-Prices          0.468739
RMSE-Yields (in %)    0.175590
AABSE-Yields (in %)   0.119219
```

```
> summary(DieboldLiGERMOutlierDefault_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.634799
AABSE-Prices          0.468738
RMSE-Yields (in %)    0.175593
AABSE-Yields (in %)   0.119221
```

```
> summary(DieboldLiGERMOutlier0.0501)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.514936
AABSE-Prices          0.346601
RMSE-Yields (in %)    0.177828
AABSE-Yields (in %)   0.104600
```

```
> summary(DieboldLiGERMOutlier0.0417_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.4833420
AABSE-Prices          0.2844801
RMSE-Yields (in %)    0.1636690
AABSE-Yields (in %)   0.0851299
```

```
> summary(DieboldLiGERMOutlier0.0285)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.841712
AABSE-Prices          0.599431
RMSE-Yields (in %)    0.302648
AABSE-Yields (in %)   0.202256
```

```
> summary(DieboldLiGERMOutlier0.0285_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      GERMANY
RMSE-Prices           0.519860
AABSE-Prices          0.278138
RMSE-Yields (in %)    0.190242
AABSE-Yields (in %)   0.102262
```

Table 8- Criteria resulting from the Application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the German sample.

As it was expected, the lowest criteria (RMSE-Yields = 0,1636690% and AABSE-Yields = 0,0851299%) was obtained when $\lambda = 0{,}0417$, which was the value for the λ when $\tau_{\text{NS/Edited}}$. In other words, NS models really provide the optimal value for τ, and therefore, the optimal value for λ in DL model.

Nonetheless, it is also interesting to verify that the default value for λ performs better than the hypothetical value in both the regular and the edited functions. This is due to the fact that the default λ and its corresponding τ are closer than the one estimated by the NS model than the hypothetical λ and τ. **Table 9** shows the remaining parameters estimated for DL model when $\lambda = 0{,}0417$.

```
> DieboldLiGERMOutlier0.0417_edit
--------------------------------------------------
Estimated Diebold/Li parameters:
--------------------------------------------------

        GERMANY
beta_0  1.64762
beta_1 -2.17388
beta_2 -3.65466
```

Table 9- Estimated parameters for the application of the Edited DL function to the German data.

The estimated parameters are practically equal to the ones estimated in the NS model (which is comprehensible given the fact that they share the same τ in the NS case and λ in the DL one). Thus, the interpretation of such factors is identical to the one performed for the NS ones.

4.2.4. McCulloch Model Aplication

The first step regarding the McCulloch Splines (MCS) method is to find out the number of basis functions, as well as knot points to use. In order to do so, expression (14) must be applied. Considering a number of bonds k = 28, the number of basis functions to be estimated is given by $n = INT\left[\sqrt{28}+0{,}5\right] = 5$. Given that the last knot point is given by q_{n-1}, there are four knot points to be estimated.

Both the first and the last knot points are easy to estimate. As it was mentioned above, $q_1 = 0$ and q_4 is equal to the maximum maturity available in the dataset. In this case, and as it is possible to see in Appendix B, the bond represented by the ISIN "DE0001135481" is the one with the greatest maturity, which is approximately 26,08 years, i.e., $q_4 = 26{,}08$.

Regarding q_2, the values of h and θ must be computed in the first place. Recalling their expressions:

- $h_2 = INT\left[\frac{(j-1)k}{n-2}\right] = INT\left[\frac{(2-1)\times 28}{5-2}\right] = 9;$
- $\theta_2 = \frac{(j-1)k}{n-2} - h = \frac{(2-1)\times 28}{5-2} - 9 = 0{,}3(3);$

Once both values are computed, it is easy to apply them into expression (15) resulting in:

- $q_2 = m_h + \theta(m_{h+1} - m_h) = m_9 + 0{,}333(m_{10} - m_9) = 2{,}3507 + 0{,}333(3{,}2356 - 2{,}3507) \approx 2{,}6457$

Both m_9 and m_{10} correspond to the maturities of the 9th and 10th bonds, since they are presented in ascending order in Appendix B. For the third knot point, the procedure is very similar:

- $h_3 = \text{INT}\left[\frac{(j-1)k}{n-2}\right] = \text{INT}\left[\frac{(3-1)\times 28}{5-2}\right] = 18;$
- $\theta_3 = \frac{(j-1)k}{n-2} - h = \frac{(3-1)\times 28}{5-2} - 18 = 0{,}6(6);$
- $q_3 = m_h + \theta(m_{h+1} - m_h) = m_{18} + 0{,}667(m_{19} - m_{18}) = 5{,}9315 + 0{,}667(6{,}1836 - 5{,}9315) \approx 6{,}0995$

In order to assess if the knot points computations are correct, it is also possible to extract that information from the software, as **Table 10** shows.

```
> CubicSplinesGERMOutlier$knotpoints
[[1]]
                DE0001141729 DE0001102358
    0.000000        2.645662     6.099543     26.082192
```

Table 10- Knot Points for the application of McC methodology to the German data.

Once all the four knot points are well defined, the next step in this procedure is to define the basis functions. Theoretically, and according to what was previously said, there are five basis functions to determine which, according do expression (16), can be defined as:

$$g^1(m) = \begin{cases} \frac{m^2}{2} - \frac{m^3}{15{,}8742} & 0 \le m < 2{,}6457 \\ 2{,}6457\left[5{,}2914 + \frac{m - 2{,}6457}{2}\right] & 2{,}6457 \le m \end{cases}$$

$$g^2(m) = \begin{cases} 0 & m < 0 \\ \frac{(m)^3}{15{,}8742} & 0 \le m < 2{,}6457 \\ 1{,}1666 + \frac{2{,}6457(m - 2{,}6457)}{2} + \frac{(m - 2{,}6457)^2}{2} - \frac{(m - 2{,}6457)^3}{20{,}7228} & 2{,}6457 \le m < 6{,}0995 \\ 6{,}0995\left[1{,}5922 + \frac{m - 6{,}0995}{2}\right] & 6{,}0995 \le m \end{cases}$$

$$g^3(m)=\begin{cases} 0 & m < 2{,}6457 \\ \dfrac{(m - 2{,}6457)^3}{20{,}7228} & 2{,}6457 \leq m < 6{,}0995 \\ 1{,}9881 + \dfrac{3{,}4538(m - 6{,}0995)}{2} + \dfrac{(m - 6{,}0995)^2}{2} - \dfrac{(m - 6{,}0995)^3}{119{,}883} & 6{,}0995 \leq m < 26{,}08 \\ 23{,}4343\left[7{,}2358 + \dfrac{m - 26{,}08}{2}\right] & 26{,}08 \leq m \end{cases}$$

$$g^4(m)=\begin{cases} 0 & m < 6{,}0995 \\ \dfrac{(m - 6{,}0995)^3}{119{,}883} & 6{,}0995 \leq m < 26{,}08 \end{cases}$$

$g^5(m) = m, \forall m$

The five previously defined expressions, in conjunction with expressions (13) and (17), allow, theoretically speaking, to compute the spot rate for any desired maturity. I would like to highlight that the expression corresponding to $g^1(m)$ only has two sub-expressions (instead of the typical four expressions defined by McC) due to the fact that it is impossible to define the first two sub-expressions contemplated in the model (given the impossibility to divide by 0).

On the other hand, $g^4(m)$ is also composed by only two sub-expressions once, attending to expression (16), a hypothetical definition of a third and fourth sub-expressions for $g^4(m)$ would require the definition of a fifth knot point that would have a higher value than the longest maturity available, which is impossible.

Table 11 shows the values for the β estimates and the results of the application of a t-test, so that it is possible to see if all the estimates are statistically significant.

```
> CubicSplinesGERMOutlier
-----------------------------------------------------
Estimated parameters and robust standard errors:
-----------------------------------------------------
[1] "GERMANY:"

t test of coefficients:

            Estimate   Std. Error t value  Pr(>|t|)
alpha 1  0.00243981  0.00522008  0.4674 0.6446197
alpha 2 -0.00592534  0.00133500 -4.4385 0.0001887 ***
alpha 3 -0.00116110  0.00030282 -3.8343 0.0008482 ***
alpha 4  0.00131414  0.00044649  2.9432 0.0072985 **
alpha 5  0.00684187  0.00412426  1.6589 0.1107060
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘
```

Table 11 - Estimated parameters for the application of McC model to the German data and the respective t-test values.

By interpreting the significance codes provided by R/RStudio, it is possible to conclude that there are two β estimates that are not statistically significant, considering a significance level of 5%. More precisely, β_1 and β_5 are not statistically significant at a significance level of 5%, which results in their exclusion from the estimation of the discount function (and thus in the estimation of the spot rates) and, consequently, the exclusion of $g^1(m)$ and $g^5(m)$.

The criteria that will be used later for the determination of the methodology more suitable to model the TSIR for the German case is displayed on **Table 12**.

```
> summary(CubicSplinesGERMOutlier)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                       GERMANY
RMSE-Prices            0.4782171
AABSE-Prices           0.2591658
RMSE-Yields (in %)     0.1687261
AABSE-Yields (in %)    0.0923868
```

Table 12- Criteria resulting from the application of the McC methodology to the German sample.

4.3. Spain

4.3.1. Nelson and Siegel Model Application

Similarly to the German case, **Table 13** shows the criteria resulting from both the regular and the edited version of the NS function.

```
> summary(NelsonSiegelSPAINOutlier)
---------------------------------------
Goodness of fit:
---------------------------------------

                     SPAIN
RMSE-Prices          0.3693258
AABSE-Prices         0.2486826
RMSE-Yields (in %)   0.1073841
AABSE-Yields (in %)  0.0636488
```

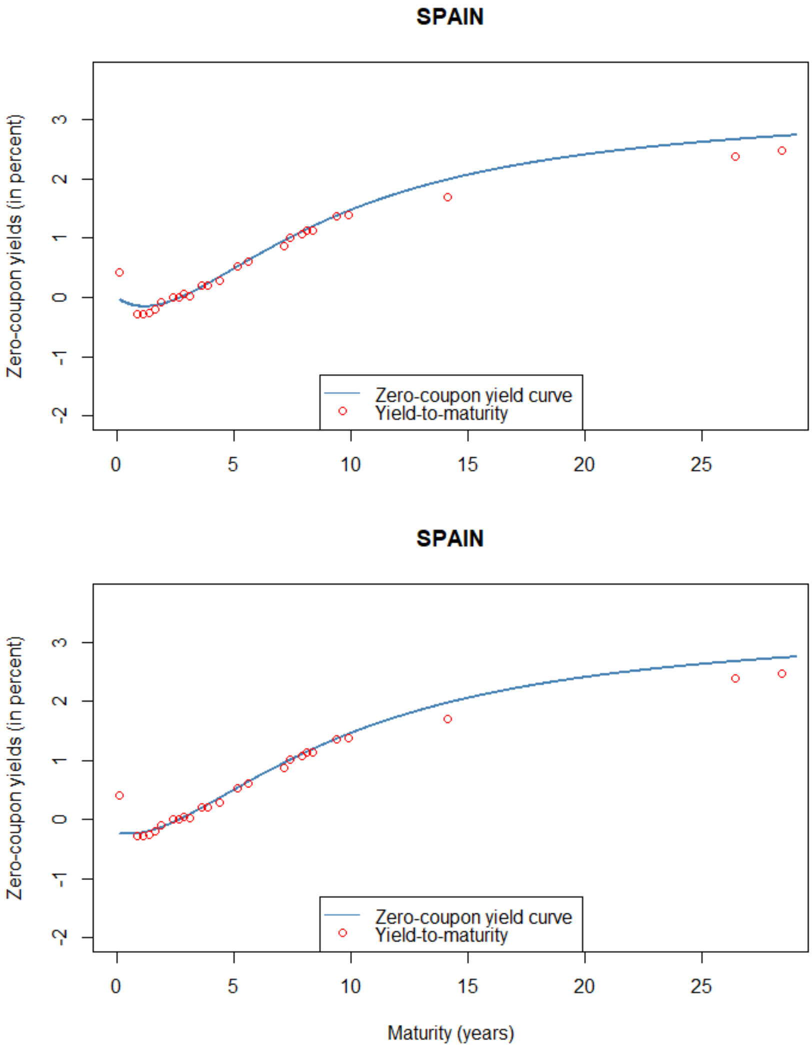
```
> summary(NelsonSiegelSPAINOutlier_edit)
---------------------------------------
Goodness of fit:
---------------------------------------

                     SPAIN
RMSE-Prices          0.3475580
AABSE-Prices         0.2338075
RMSE-Yields (in %)   0.1369770
AABSE-Yields (in %)  0.0624221
```

Table 13- Criteria resulting from the Application of the Regular (left) and Edited (right) functions of NS model to the Spanish data.

While the AABSE-Yields$_{\text{Regular}}$ = 0,0636488% is greater than the AABSE-Yields$_{\text{Edited}}$ = 0,0624221% (suggesting that the edited function of the model is better than the regular one), the RMSE-Yields$_{\text{Edited}}$ = 0,1369770% is greater than the RMSE-Yields$_{\text{Regular}}$ = 0,1073841% (suggesting, on the other hand, that the regular function is better than the edited one).

It was previously stated that the main difference between the RMSE and the AABSE is that the first one attributes a higher weight to large errors than the later, as it is possible to conclude by a brief comparison between expression (18) and expression (19). In order to verify if there is any large error contributing to this difference of results, the plots of both estimations are displayed in **Plot 4**.

As it was expected, there is a bond yield causing this disagreement between the two metrics. The first bond, which could be considered an outlier and thus be excluded from the model, has a positive YTM while the immediately following ones are negative.



Plot 4- Comparison between the curves resulting from the application of the Regular (upper) and the Edited (lower) NS functions to the Spanish data.

Since the regular function restricts the value of the spot rate to be positive or equal to 0 when m → 0, it forces the ZCYC to start exactly at 0 %, which reduces the error corresponding to the first bond when compared to the edited function. Given the fact that the error has a substantial value, this difference is the reason why there is a discrepancy between the RMSE and the AABSE.

However, I decide to choose the edited function for the NS model for a simple reason: it allows the instantaneous short rate to be negative, which is consistent with the current economic and financial reality.

```
> NelsonSiegelSPAINOutlier_edit
---------------------------------------------------
Estimated Nelson/Siegel parameters:
---------------------------------------------------

          SPAIN
beta_0  3.52552
beta_1 -3.74884
beta_2 -4.27965
tau_1   2.79850
```

Table 14- Estimated parameters for the application of the Edited NS function to the Spanish data.

According to **Table 14**, the long-term IR for Spain, which corresponds to the β_0 estimate, is approximately equal to 3,53%. On the other hand, and having in mind its definition, the value for the instantaneous short rate is equal to -0,22332%. Furthermore, and just like in the German case, the value of β_1 estimate is negative, which indicates a positive slope of the function towards its limit.

Moreover, the Spanish ZCYC will also describe a U-shape (due to the negative β_2 estimate) occurring when m = 2,7985 (approximately 2 years and 9 months). Considering all the parameters, the expression of the NS function for the Spanish case, based on expression (6), is given by:

$$s(m,\beta)= 3{,}526 - 3{,}749\left[\frac{1 - \exp\left(-\frac{m}{2{,}799}\right)}{\left(\frac{m}{2{,}799}\right)}\right] -4{,}28\left[\frac{1 - \exp\left(-\frac{m}{2{,}799}\right)}{\left(\frac{m}{2{,}799}\right)} -\exp\left(-\frac{m}{2{,}799}\right)\right]$$

4.3.2. Svensson and Adjusted Svensson Models Application

Table 15 show the criteria related to the regular and the edited estimation of both the SV and ASV models.

By analysing the RMSE-Yields and the AABSE-Yields of the four obtained outputs, the regular estimation for the SV is the one which performs better for the Spanish sub-dataset, since it is the one that produces the lowest values for the presented criteria. Thus, **Table 16** shows the estimated SV parameters for this set of Spanish bonds.

```
> summary(SvenssonSPAINOutlier)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                      SPAIN
RMSE-Prices           0.3158726
AABSE-Prices          0.2339675
RMSE-Yields (in %)    0.0931929
AABSE-Yields (in %)   0.0580431
```

```
> summary(SvenssonSPAIN_editOutlier)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                      SPAIN
RMSE-Prices           0.3164471
AABSE-Prices          0.2342109
RMSE-Yields (in %)    0.0935875
AABSE-Yields (in %)   0.0581422
```

```
> summary(AdjSvenssonSPAINOutlier)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                      SPAIN
RMSE-Prices           0.3248387
AABSE-Prices          0.2369366
RMSE-Yields (in %)    0.0988934
AABSE-Yields (in %)   0.0593164
```

```
> summary(AdjSvenssonSPAINOutlier_edit)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                      SPAIN
RMSE-Prices           0.3216703
AABSE-Prices          0.2353550
RMSE-Yields (in %)    0.1003769
AABSE-Yields (in %)   0.0597156
```

Table 15- Criteria resulting from the Application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the Application of the Regular (lower left) and Edited (lower right) functions of ASV to the Spanish data.

According to the estimated parameters, the theoretical long-term IR for Spain, given by β_0, is equal to 0,967546%, while the instantaneous short rate is equal to 0,094897%.

```
> SvenssonSPAINOutlier
---------------------------------------------------
Estimated Svensson parameters:
---------------------------------------------------

          SPAIN
beta_0   0.967546
beta_1  -0.872649
beta_2  -3.114583
tau_1    1.595908
beta_3   7.341866
tau_2   25.551031
```

Table 16 - Estimated parameters for the application of the Regular SV function to the Spanish data.

The location of the two different shapes is given by the estimated values for τ_1 and τ_2, which are approximately 1,596(one year and seven months) and 25,551(twenty-five years and six months), respectively. Furthermore, the first shape, located when m ≈ 1,596, will be a U-shape (given that $\beta_2 < 0$), while the second one, occurring when m ≈ 25,551, will be a hump.

The ZCYC for Spain, given the parameters achieved with the SV regular estimation and applying expression (8), is defined as:

$$s(m,\beta)=0{,}966 - 0{,}873\left[\frac{1 - \exp\left(-\frac{m}{1{,}596}\right)}{\left(\frac{m}{1{,}596}\right)}\right] - 3{,}115\left[\frac{1 - \exp\left(-\frac{m}{1{,}596}\right)}{\left(\frac{m}{1{,}596}\right)} - \exp\left(-\frac{m}{1{,}596}\right)\right]$$
$$+ 7{,}342\left[\frac{1\text{-}\exp\left(-\frac{m}{25{,}551}\right)}{\left(\frac{m}{25{,}551}\right)} - \exp\left(-\frac{m}{25{,}551}\right)\right]$$

4.3.3. Diebold and Li Model Application

Similarly to the German case, I have predefined three values for λ: the default value (λ_{Default} = 0,0609), the hypothetical value ($\lambda_{\text{Hypothetical}}$ = 0,0285) and a new value which is linked to the τ obtained in both the regular and edited NS model for the Spanish case ($\lambda_{\text{NS/Regular}}=\frac{1}{2{,}53435 \times 12} \approx 0{,}0329$ and $\lambda_{\text{NS/Edited}}= \frac{1}{2{,}7985 \times 12} \approx 0{,}0298$). The criteria resulting from the DL estimation for all the previous λ is shown in **Table 17**.

```
> summary(DieboldLiSPAINOutlierDefault)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           1.132085
AABSE-Prices          0.830615
RMSE-Yields (in %)    0.292902
AABSE-Yields (in %)   0.193043
```

```
> summary(DieboldLiSPAINOutlierDefault_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           1.131933
AABSE-Prices          0.830578
RMSE-Yields (in %)    0.292977
AABSE-Yields (in %)   0.193072
```

```
> summary(DieboldLiSPAINOutlier0.0329)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           0.3695596
AABSE-Prices          0.2487033
RMSE-Yields (in %)    0.1073552
AABSE-Yields (in %)   0.0636344
```

```
> summary(DieboldLiSPAINOutlier0.0298_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           0.3476699
AABSE-Prices          0.2338644
RMSE-Yields (in %)    0.1367989
AABSE-Yields (in %)   0.0624204
```

```
> summary(DieboldLiSPAINOutlier0.0285)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           0.3948815
AABSE-Prices          0.2845237
RMSE-Yields (in %)    0.1244025
AABSE-Yields (in %)   0.0807084
```

```
> summary(DieboldLiSPAINOutlier0.0285_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                      SPAIN
RMSE-Prices           0.3427508
AABSE-Prices          0.2304593
RMSE-Yields (in %)    0.1472569
AABSE-Yields (in %)   0.0624811
```

Table 17- Criteria resulting from the application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the Spanish sample.

Not surprisingly, the criteria demonstrate that the DL also performs better in the Spanish case when the value for λ is equivalent to the value for τ estimated in the NS procedure, which leads to very similar parameters compared to the ones computed in the NS case. However, and unlike in the German case, the hypothetical value for λ performs better when compared to the default one, once it is closer to the value obtained in the NS estimation.

Assuming a value for λ = 0,0298, the estimated parameters for the DL procedure are shown in **Table 18**.

```
> DieboldLiSPAINOutlier0.0298_edit
---------------------------------------------------
Estimated Diebold/Li parameters:
---------------------------------------------------

          SPAIN
beta_0   3.52506
beta_1  -3.74725
beta_2  -4.28376
```

Table 18- Estimated parameters for the application of the Edited DL function to the Spanish data.

Given the similarity of these estimated parameters to the ones shown in exhibit Y, the interpretation performed before still holds for this application.

4.3.4 McCulloch Model Application

As it was shown before, for the the 28 bonds that compose the German sub-dataset, it was necessary to define 5 basis functions. Once the Spanish sample has a similar number of bonds (k = 25), and applying once again expression (14), there are $n=\text{INT}\left[\sqrt{25}+0{,}5\right] = 5$ basis functions to be defined and thus, 4 knot points.

The definition of the first and the last knot points is, once again, straightforward. Similarly to the German case, the first knot point, q_1, is equal to 0, while the last corresponds to the longest maturity available in the dataset which, in the Spanish case, is equal to the maturity of the 25th bond, m_{25} = 28,408 years. As before, the application of expression (15) for the determination of the two remaining knot points, requires the determination of the auxiliary values for h and θ, this time taking into account the sample size for Spain:

- $h_2 = \text{INT}\left[\frac{(l-1)k}{n-2}\right] = \text{INT}\left[\frac{(2-1)\times 25}{5-2}\right] = 8;$
- $\theta_2 = \frac{(l-1)k}{n-2} - h = \frac{(2-1)\times 25}{5-2} - 8 = 0{,}3(3);$
- $q_2 = m_h + \theta(m_{h+1} - m_h) = m_8 + 0{,}333(m_9 - m_8) = 2{,}6438 + 0{,}333(2{,}8877 - 2{,}6438) \approx 2{,}7251;$

Repeating the previous procedure for the third knot point, the results are the following:

- $h_3 = \text{INT}\left[\frac{(l-1)k}{n-2}\right] = \text{INT}\left[\frac{(3-1)\times 25}{5-2}\right] = 16$;
- $\theta_3 = \frac{(l-1)k}{n-2} - h = \frac{(3-1)\times 25}{5-2} - 16 = 0{,}6(6)$;
- $q_3 = m_h + \theta(m_{h+1} - m_h) = m_{16} + 0{,}667(m_{17} - m_{16}) = 7{,}1397 + 0{,}666(7{,}3945 - 7{,}1397) \approx 7{,}3096$;

With the definition of the four knot points, and attending to expression (16), it is now possible to define the basis functions as:

$$g^1(m) = \begin{cases} \frac{m^2}{2} - \frac{m^3}{16{,}3506} & 0 \leq m < 2{,}7251 \\ 2{,}7251\left[5{,}4502 + \frac{m - 2{,}7251}{2}\right] & 2{,}7251 \leq m \end{cases}$$

$$g^2(m) = \begin{cases} 0 & m < 0 \\ \frac{m^3}{16{,}3506} & 0 \leq m < 2{,}7251 \\ 1{,}2377 + \frac{2{,}7251(m - 2{,}7251)}{2} + \frac{(m - 2{,}7251)^2}{2} - \frac{(m - 2{,}7251)^3}{27{,}507} & 2{,}7251 \leq m < 7{,}3096 \\ 7{,}3096\left[1{,}9824 + \frac{m - 7{,}3096}{2}\right] & 7{,}3096 \leq m \end{cases}$$

$$g^3(m) = \begin{cases} 0 & m < 2{,}7251 \\ \frac{(m - 2{,}7251)^3}{27{,}507} & 2{,}7251 \leq m < 7{,}3096 \\ 3{,}5029 + \frac{4{,}5845(m - 7{,}3096)}{2} + \frac{(m - 7{,}3096)^2}{2} - \frac{(m - 7{,}3096)^3}{126{,}5904} & 7{,}3096 \leq m < 28{,}408 \\ 25{,}6829\left[46{,}7813 + \frac{m - 28{,}408}{2}\right] & 28{,}408 \leq m \end{cases}$$

$$g^4(m) = \begin{cases} 0 & m < 7{,}3096 \\ \frac{(m - 7{,}3096)^3}{126{,}5904} & 7{,}3096 \leq m < 28{,}408 \end{cases}$$

$g^5(m) = m, \forall\, m$

However, and similar to the results regarding the German application of McC methodology, **Table 19** shows the β estimates (as well as the results of the t-test) where it may be concluded that the first and the last β estimates, β_1 and β_5, are not statistically significant at a 5% significance level. Thus, the application of the first and the last basis functions, $g^1(m)$ and $g^5(m)$, is irrelevant if we consider such significance level, which results in the exclusion of those two functions when applying this model to the Spanish sub-dataset.

```
> CubicSplinesSPAINOutliers
---------------------------------------------------
Estimated parameters and robust standard errors:
---------------------------------------------------
[1] "SPAIN:"

t test of coefficients:

           Estimate   Std. Error  t value  Pr(>|t|)
alpha 1  0.00174584  0.00274648   0.6357    0.5322
alpha 2 -0.00649873  0.00058255 -11.1557 4.879e-10 ***
alpha 3 -0.00101272  0.00015657  -6.4680 2.629e-06 ***
alpha 4  0.00205050  0.00027125   7.5594 2.765e-07 ***
alpha 5  0.00152348  0.00228163   0.6677    0.5119
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 19- Estimated parameters for the application of McC model to the Spanish data and the respective t-test values

The criteria regarding this estimation of the McC methodology is shown in **Table 20**, which will be useful for a comparison between all the results from all the estimations later on this thesis.

```
> summary(CubicSplinesSPAINOutliers)
---------------------------------------------------
Goodness of fit:
---------------------------------------------------

                      SPAIN
RMSE-Prices           0.2465712
AABSE-Prices          0.1898545
RMSE-Yields (in %)    0.1259986
AABSE-Yields (in %)   0.0637313
```

Table 20 - Criteria resulting from the application of the McC methodology to the Spanish sample.

4.4. Poland

4.4.1. Nelson and Siegel Model Application

Repeating the procedure used for the previous two cases, **Table 21** shows the criteria for the estimation of the parameters of NS model for the Polish case.

```
> summary(NelsonSiegelPOLANDOutlier)
-------------------------------------------
Goodness of fit:
-------------------------------------------

                    POLAND
RMSE-Prices         0.6709467
AABSE-Prices        0.4457237
RMSE-Yields (in %)  0.0702709
AABSE-Yields (in %) 0.0580572
```

```
> summary(NelsonSiegelPOLANDOutlier_edit)
-------------------------------------------
Goodness of fit:
-------------------------------------------

                    POLAND
RMSE-Prices         0.6709551
AABSE-Prices        0.4457272
RMSE-Yields (in %)  0.0702700
AABSE-Yields (in %) 0.0580564
```

Table 21- Criteria resulting from the application of the Regular (left) and Edited (right) functions of NS model to the Polish data.

Unlikely the German and the Spanish examples, the estimation of the NS parameters for Polish bonds is the first scenario where both the regular and the edited functions present similar results. By looking at the criteria displayed above for the two functions, it is possible to conclude that the RMSE and AABSE values for both functions are practically equal (for example, the difference between the RMSE-Yields$_{\text{Regular}}$ and the RMSE-Yields$_{\text{Edited}}$ is equal to 0,0000009 %, which is insignificant).

The reason why both the criteria shown in **Table 21** and the parameters estimates shown in **Table 22** are not exactly equal is related to the fact that, by changing the regular function into the edited one, the start parameters (the parameters at which the estimation starts in order to achieve the optimal result) also change. By changing the start parameters, the final optimal results may not be exactly equally, even though they are very approximate.

```
> NelsonSiegelPOLANDOutlier
------------------------------------------
Estimated Nelson/Siegel parameters:
------------------------------------------

        POLAND
beta_0  2.62252
beta_1 -2.25589
beta_2 -5.22906
tau_1   1.66475
```

```
> NelsonSiegelPOLANDOutlier_edit
------------------------------------------
Estimated Nelson/Siegel parameters:
------------------------------------------

        POLAND
beta_0  2.62253
beta_1 -2.25593
beta_2 -5.22900
tau_1   1.66479
```

Table 22- Estimated parameters for the application of the Regular (left) and Edited(right) NS function to the Polish data.

Since the results are very close, the following interpretation holds for both functions. Therefore, for the Polish case, the long-term IR, given by β_0 estimate, is approximately equal to 2,62%. Moreover, and unlike the two previous NS estimations for the Germany and Spain, the instantaneous short rate will be positive ($\beta_0 + \beta_1 > 0$) and approximately equal to 0,366%. From the estimated value for β_1, it is also possible to conclude that the spot rate function will increase i.e. positive slope, towards its limit.

Regarding the remaining two parameters, and in line with the results for the other two countries, the shape created by this function will be a U-shape, resulting from the negative estimate for β_2. Moreover, and according to the estimate for $\tau \approx 1{,}665$, this U-shape occurs when the maturity is approximately 1 year and 8 months. Finally, substituting the parameters in expression (6), the NS spot rate function for Poland is equal to:

$$s(m,\beta)=2{,}622 - 2{,}556\left[\frac{1 - \exp\left(-\frac{m}{1{,}665}\right)}{\left(\frac{m}{1{,}665}\right)}\right] - 5{,}229\left[\frac{1 - \exp\left(-\frac{m}{1{,}665}\right)}{\left(\frac{m}{1{,}665}\right)} - \exp\left(-\frac{m}{1{,}665}\right)\right]$$

4.4.2. Svensson and Adjusted Svensson Models Application

From the results presented before in the NS estimation, it was shown that there is not any Polish bond that forces the instantaneous short rate to be negative, resulting in an indifference regarding the edited and the regular NS functions. Therefore, it is expected that this indifference also occurs in both the SV and ASV estimations. **Table 23** shows the criteria for the estimation using both the regular and the edited SV and ASV functions.

```
> summary(SvenssonPOLANDOutlier)          > summary(SvenssonPOLANDOutlier_edit)
----------------------------------------- -----------------------------------------------
Goodness of fit:                          Goodness of fit:
----------------------------------------- -----------------------------------------------

                     POLAND                                    POLAND
RMSE-Prices          0.4315658            RMSE-Prices          0.4319176
AABSE-Prices         0.3407207            AABSE-Prices         0.3412148
RMSE-Yields (in %)   0.0584741            RMSE-Yields (in %)   0.0585707
AABSE-Yields (in %)  0.0486658            AABSE-Yields (in %)  0.0488069
> summary(AdjSvenssonPOLANDOutlier)       > summary(AdjSvenssonPOLANDOutlier_edit)
----------------------------------------- -----------------------------------------------
Goodness of fit:                          Goodness of fit:
----------------------------------------- -----------------------------------------------

                     POLAND                                    POLAND
RMSE-Prices          0.4372262            RMSE-Prices          0.4352831
AABSE-Prices         0.3463222            AABSE-Prices         0.3439109
RMSE-Yields (in %)   0.0586345            RMSE-Yields (in %)   0.0585242
AABSE-Yields (in %)  0.0490108            AABSE-Yields (in %)  0.0488546
```

Table 23- Criteria resulting from the application of the Regular (upper left) and Edited (upper right) functions of SV model, as well as from the application of the Regular (lower left) and Edited (lower right) functions of ASV to the Polish data.

Firstly, comparing the criteria related to both the regular SV and ASV estimations with the criteria resulting with the edited estimations, the conclusion is similar to the one regarding the NS case: the edited function does not improve the accuracy of the estimation.

Furthermore, by comparing the criteria among the two models (SV and ASV), one concludes that they also present very similar values: using the ASV does not improve the accuracy of the model, which indicates that there is no existence of multicollinearity in this sample.

However, the regular function for the SV estimation was the one with the lowest $RMSE_{Yields}$ and $AABSE_{Yields}$ (even though the difference towards the other criteria is very small) and thus, Table 24 provides the results for that estimation.

Even though those parameters produce a curve that accurately models the TSIR for the Polish bonds (attending to the low RMSE and AABSE of the estimation), some of them cannot be reliably interpreted as they were for the previous cases.

```
> SvenssonPOLANDOutlier
---------------------------------------------------
Estimated Svensson parameters:
---------------------------------------------------

        POLAND
beta_0   0.000000120318
beta_1   0.150055588827
beta_2  -2.753924822124
tau_1    1.685144315939
beta_3   7.722983814893
tau_2   12.507615818088
```

Table 24- Estimated parameters for the application of the regular SV function to the Polish sample.

For example, and attending to **Table 24**, interpreting the estimate for β_0 as the long-term IR would mean that, in the long term, the spot rate for Poland would be approximately equal to 0, which does not make sense. In fact, according to the estimated parameters, the instantaneous short rate ($\beta_0 + \beta_1 \approx 0{,}15\%$) is higher than the long-term IR, which, once again, is incorrect.

Nonetheless, and having in mind the low criteria results obtained with this estimation as well as expression (8), I will use these parameters to model the ZCYC, resulting in the following expression:

$$s(m,\beta) = 0{,}15\left[\frac{1 - \exp\left(-\frac{m}{1{,}685}\right)}{\left(\frac{m}{1{,}685}\right)}\right] - 2{,}754\left[\frac{1 - \exp\left(-\frac{m}{1{,}685}\right)}{\left(\frac{m}{1{,}685}\right)} - \exp\left(-\frac{m}{1{,}685}\right)\right]$$

$$+ 7{,}723\left[\frac{1\text{-}\exp\left(-\frac{m}{12{,}508}\right)}{\left(\frac{m}{12{,}508}\right)} - \exp\left(-\frac{m}{12{,}508}\right)\right]$$

4.4.3. Diebold and Li Model Application

Once again, I have defined a group of values for λ, being two of them equal to the German and the Spanish sub-datasets (the default value, λ_{Default} = 0,0609 and the hypothetical value, $\lambda_{\text{Hypothetical}}$ = 0,0285).

The remaining two values were, once again, defined based on the value for τ obtained in the NS estimation. Since both values are very close to each other ($\tau_{\text{NS/Regular}}$ = 1,66475 and $\tau_{NS/Edited}$ = 1,66479), I defined just one more value for $\lambda = \frac{1}{1{,}66475 \times 12} \approx 0{,}05$, which I used for both the regular and the edited DL estimations.

The criteria associated with the parameter estimations using the DL model for the Polish bonds are shown in **Table 25**.

```
> summary(DieboldLiPOLANDOutlierDefault)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.6709611
AABSE-Prices         0.4573956
RMSE-Yields (in %)   0.1151188
AABSE-Yields (in %)  0.0806713
```

```
> summary(DieboldLiPOLANDOutlierDefault_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.6709838
AABSE-Prices         0.4573878
RMSE-Yields (in %)   0.1150815
AABSE-Yields (in %)  0.0806615
```

```
> summary(DieboldLiPOLANDOutlier0.05)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.6712056
AABSE-Prices         0.4457709
RMSE-Yields (in %)   0.0701205
AABSE-Yields (in %)  0.0579530
```

```
> summary(DieboldLiPOLANDOutlier0.05_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.6712030
AABSE-Prices         0.4457675
RMSE-Yields (in %)   0.0701216
AABSE-Yields (in %)  0.0579531
```

```
> summary(DieboldLiPOLANDOutlier0.0285)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          1.078315
AABSE-Prices         0.773380
RMSE-Yields (in %)   0.131451
AABSE-Yields (in %)  0.114218
```

```
> summary(DieboldLiPOLANDOutlier0.0285_edit)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.8236750
AABSE-Prices         0.5660077
RMSE-Yields (in %)   0.1419937
AABSE-Yields (in %)  0.0978112
```

Table 25- Criteria resulting from the application of the Regular (left side) and Edited (right side) functions of DL model, attending to the previously defined λ values, for the Polish sample.

Not surprisingly, and in line with what happened in the previous cases, the DL estimations with the best results are the ones in which the λ value is equivalent to the τ value from the NS estimation. Furthermore, and attending to the remaining four results, since the hypothetical value is closer to the optimal value than the default DL value, the estimation using λ = 0,0285 is more suitable to model the TSIR than the one using the default value.

Table 26 show the estimated parameters assuming that λ = 0,05. As it was expected, such results are extremely similar to the ones achieved in the NS estimation procedure applied previously.

```
> DieboldLiPOLANDOutlier0.05
---------------------------------------------
Estimated Diebold/Li parameters:
---------------------------------------------

        POLAND
beta_0   2.62296
beta_1  -2.25995
beta_2  -5.22041
```

Table 26- Estimated parameters for the application of the Edited DL function to the Polish data.

Like the previous cases, the interpretation performed in the NS estimation still holds for the DL obtained parameters.

4.4.4. McCulloch Model Application

Similarly to the two previous CS methodology applications, the determination of the number of basis functions to define constitutes the first step. Therefore, and attending to the Polish sample size of 17 bonds, k = 27, based on expression (14), $n=\text{INT}\left[\sqrt{17}+0,5\right]=4$ basis functions need to be defined (which is different from the previous two cases, where, in both them, due to the higher sample size, it was necessary to define 5 basis functions). Logically, by having 4 basis functions to be defined, there is the need to define 3 knot points.

The first knot point will, according to the conditions imposed by the model, be equal to 0 ($q_1 = 0$), whereas the third and last one will assume the value of the longest maturity present in the Polish sub-dataset which, attending to Appendix B, corresponds to m_{17} = 28,3918 years = q_3. The remaining knot point is defined according to expression (15) as:

- $h_2 = \text{INT}\left[\frac{(1-1)k}{4-2}\right] = \text{INT}\left[\frac{(2-1)\times 17}{4-2}\right] = 8;$
- $\theta_2 = \frac{(1-1)k}{n-2} - h = \frac{(2-1)\times 17}{4-2} - 8 = 0,5;$
- $q_2 = m_h + \theta(m_{h+1} - m_h) = m_8 + 0,5(m_9 - m_8) = 6,0822 + 0,5(6,6164 - 6,0822) \approx 6,3493;$

Once the three knot points are defined, the definition of the four basis functions applicable to the Polish sub-dataset, and according to expression (16), is given by:

$$g^1(m)=\begin{cases} \frac{m^2}{2} - \frac{m^3}{38,0958} & 0 \leq m < 6,3493 \\ 6,3493\left[12,6986 + \frac{m-6,3493}{2}\right] & 6,3493 \leq m \end{cases}$$

$$g^2(m)=\begin{cases} 0 & m < 0 \\ \frac{m^3}{38,0958} & 0 \leq m < 6,3493 \\ 6,7189 + \frac{6,3493(m-6,3493)}{2} + \frac{(m-6,3493)^2}{2} - \frac{(m-6,3493)^3}{132,255} & 6,3493 \leq m < 28,3918 \\ 28,3918\left[8,4057 + \frac{m-28,3918}{2}\right] & 28,3918 \leq m \end{cases}$$

$$g^3(m)=\begin{cases} 0 & m < 6{,}3493 \\ \dfrac{(m - 6{,}3493)^3}{132{,}255} & 6{,}3493 \leq m < 28{,}3918 \end{cases}$$

$g^4(m) = m, \forall\ m$

For the previously defined basis functions, the results regarding the β estimates are displayed on **Table 27**.

```
> CubicSplinesPOLANDOutlier
----------------------------------------------------
Estimated parameters and robust standard errors:
----------------------------------------------------
[1] "POLAND:"

t test of coefficients:

           Estimate   Std. Error t value  Pr(>|t|)
alpha 1 -0.00770280  0.00085928 -8.9642 6.299e-07 ***
alpha 2 -0.00134591  0.00022001 -6.1176 3.674e-05 ***
alpha 3  0.00301791  0.00035003  8.6219 9.761e-07 ***
alpha 4  0.00967957  0.00160167  6.0434 4.143e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Table 27- Estimated parameters for the application of McC model to the Polish data and the respective t-test values

As it seems, the application of McC model to the Polish bonds is the first case where all the β estimates are statistically significant for a 5% significance level. Thus, and unlikely to the procedure applied in the previous two cases, since all the estimates are statistically significant, it is not necessary to exclude any of such estimates and therefore, none of the basis functions defined before.

The criteria related to this estimation is displayed on **Table 28**.

```
> summary(CubicSplinesPOLANDOutlier)
---------------------------------------------
Goodness of fit:
---------------------------------------------

                     POLAND
RMSE-Prices          0.3886941
AABSE-Prices         0.3039050
RMSE-Yields (in %)   0.1690640
AABSE-Yields (in %)  0.0897713
```

Table 30-Criteria resulting from the application of the McC methodology to the Polish sample.

5. Discussion of Results

Given all the estimations performed in the previous section, I intend to assess which of the presented models describes better the behaviour of the three countries bond yields and thus it is more suitable to model the TSIR of those countries. Having this in mind, **Table 1** gathers the RMSE and AABSE (in %) of every estimation performed before[4].

Table 29- Summary of the criteria resulting from all the different estimations.

			Germany	Spain	Poland
Nelson and Siegel	Regular Function	RMSE	0,177809	0,1073841	0,0702709
		AABSE	0,104588	0,0636488	0,0580572
	Edited Function	RMSE	0,1636731	0,1369770	0,0702700
		AABSE	0,0851209	0,0624221	0,0580564
Svensson	Regular Function	RMSE	0,1615744	0,0931929	0,0584741
		AABSE	0,0853793	0,0580431	0,0486658
	Edited Function	RMSE	0,161635	0,0935875	0,0585707
		AABSE	0,0854414	0,0581422	0,0488069
Adjusted Svensson	Regular Function	RMSE	0,1604473	0,0988934	0,0586345
		AABSE	0,0842777	0,0593164	0,0490108
	Edited Function	RMSE	0,1620800	0,1003769	0,0585242
		AABSE	0,0857496	0,0597156	0,0488546
Cubic Splines	Regular Function	RMSE	0,1687261	0,1259986	0,1690640
		AABSE	0,0923868	0,0637313	0,0897713

For Germany, the regular estimation for the ASV model is the one showing the lowest deviation (RMSE = 0,1604473% and AABSE = 0,0842777%) and thus the best one to describe the behaviour of the term structure of German IR while, on the other hand, the regular NS was the worse model to model to meet the same purposes, with a RMSE =0,177809% and an AABSE = 0,104588%.

In the Spanish case, the best results were achieved with the regular SV estimation, which led to a RMSE = 0,0931929% and an AABSE = 0,0580431%). Depending on the criteria

[4] I did not include DL results since, having computed the parameters for NS model, they do not add any benefit apart from making the estimation procedure easier (by defining a fix value for λ)

to be analysed, there are two models that produce the two worst comparison criteria: i) the edited estimation for the NS model produces the highest value for the RMSE = 0,1369770% and ii) the McCulloch splines model, which produces the highest value for AABSE = 0,0637313%).

Finally, and similarly to the previous case, the regular SV estimation for the Polish bonds achieved the lowest criteria (RMSE = 0,0584741% and AABSE = 0,0486658%) while the McCulloch splines methodology produced the highest and thus the worst criteria (RMSE = 0,1690640% and AABSE = 0,0897713%).

According to the information provided by the European Central Bank (ECB), the three countries approached in this thesis use SV models while estimating the TSIR. Thus, the achieved results are in line with the current practice in those three countries. Nonetheless, I would like to add some comments regarding the model estimations themselves.

Firstly, it is curious to state that the edited function only performed better than the regular function for the NS estimations. For SV and ASV, due to the increase of flexibility caused by the addition of the last term (allowing the curve to start at positive values when $m \rightarrow 0$), the regular function has shown better results than the edited one.

Furthermore, it is possible to state that, by comparing the estimated parameters for Germany and Spain in the NS and the SV/ASV estimations, the values for τ in the NS model tend to be higher than the values for τ_1 in the SV/ASV models (in the Polish case, both values are extremely similar). Otherwise speaking, the addition of the fourth term proposed in the SV/ASV models provides enough flexibility to the spot rate curve so that the first hump/U-shape occurs for shorter maturities than in the NS case, since the modelling of the longer maturity yields is assured by the second curvature.

Secondly, I would like to emphasise, once more, the results achieved with the DL estimations for the three countries. As I stated before, once the optimal value for τ is estimated in the NS estimations, DL model does not bring any significant improvement in estimating more suitable parameters. However, by defining a-priori a fixed value for λ, the estimation procedure becomes much easier and faster than in the NS case. In fact, while the edited estimation for the NS model could take several hours, the estimation for the edited DL function was instantaneous. In addition, the DL can be particularly useful for a more advanced analysis of the TSIR, since it allows to draw conclusions regarding the behaviour of the ZCYC for different λ values.

Finally, I would like to stress the results achieved with the MCS methodology, which have two great drawbacks: i) the RMSE and AABSE values resulting from the MCS estimation for the three countries were always high, even though when the MCS methodology was not the model with the worst criteria, and ii) the parameters estimated with this procedure cannot be financially interpreted, unlike the ones estimated with the remaining four models. Nonetheless, historically speaking, the development of the MCS model in 1971 was a breakthrough in modelling the TSIR, for which it should always be considered as a benchmark.

Once I have concluded regarding the most appropriate models to model the TSIR, I will now draw conclusions about the relationship between the presented bonds' credit ratings and the corresponding return. For that purpose, I have decided to use Moody's rating system.

When investing in any security, there is always a clear relationship between risk and return. The higher the risk associated with any investment, the higher will be the return.

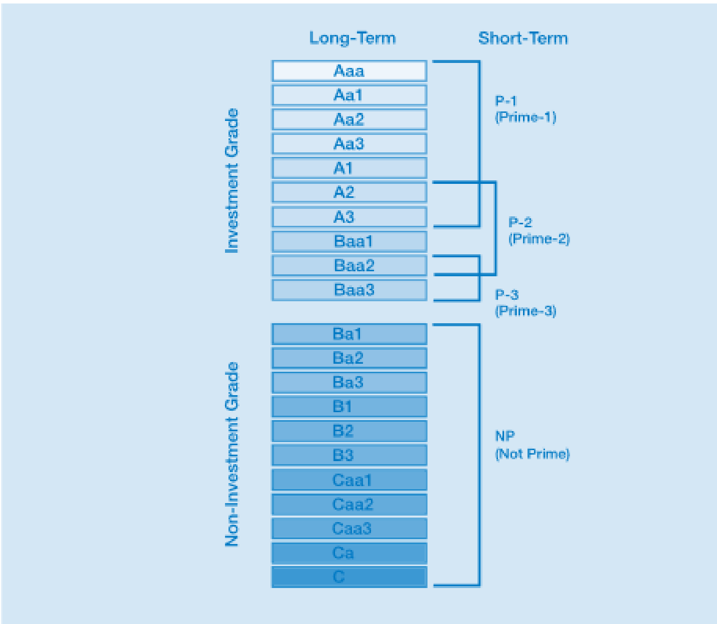


Table 30 33- Moody's Rating Scale (Souce: Moody's website)

Moody's rating system classifies German debt securities as Aaa (the highest grade in Moody's scale and thus, the most secure investment), Spanish as Baa1 and Polish as A2. Consulting the scale on **Table 29**, it is possible to assess that, according to Moody's, German debt securities correspond to the investment with the lowest associated credit risk, while, on the other hand, Spanish bonds carry the highest associated risk. Summing up, Credit Risk$_{Germany}$ < Credit Risk$_{Poland}$ < Credit Risk$_{Spain}$.

In order to verify if the relationship depicted before between risk and return also holds for the obtained results, I determined the 3, 15 and 30-Years IR for the three approached countries, by using expressions resulting from the ASV estimation for Germany, the SV estimation for Spain and the SV estimation for Poland, which are shown on **Table 2.**

Table 31- Interest Rates for different maturities determined by applying the optimal model for each country.

	3 Years	15 Years	30 Years
Germany	-0,564%	0,869%	1,276%
Spain	0,044%	2,015%	2,806%
Poland	0,039%	1,879%	2,079%

The computed IR for the three countries show a clear relationship between them. Germany presents always the lowest value for the IR, followed by Poland and lastly by Spain, regardless the maturity. Furthermore, as the maturity increases, so does the provided return of these instruments once, for longer maturities, the uncertainty and thus risk are also higher.

Recalling that German bonds have the highest classification in Moody's grading scale, it was expected that the return associated with them was the lowest of the three countries. Therefore, and relating with the risk profiles presented in the Introduction, a risk averse investor would prefer to invest in German bonds, rather than any of the remaining two.

Furthermore, and doing the opposite reasoning for Spain, it was also expected that the IR resulting from the Spanish bonds were the highest, given that Spain has the lowest rating, and thus, the highest associated credit risk. This higher level of risk would attract risk taker investors to invest in government bonds issued by Spain.

Finally, and given that Poland's classification is in the middle of the other two countries' classification, it is normal that the return associated to Polish bonds also assume intermediate levels. Being on an intermediate level, Polish bonds could be attractive to either risk averse and risk taker investors, or even to an investor who has a risk neutral position.

6. Conclusion

Interest rates are widely applied in the financial environment once, for example, several asset valuation methodologies use them as a reference to discount future cash flows. Being so and having in mind that the value for interest rates is not directly observable, it is crucial that the estimation of such values is performed in the most accurately and correct manner possible.

The difficult access to information regarding government bonds constituted a limitation to the achievement of the goal of this thesis, once with access to more information, the results presented would be more accurate. Nonetheless, and given the sample sizes of the elected date, the results presented were enough to extract contributing conclusions regarding the approached concepts.

Furthermore, being a topic that is not frequently chosen for academic purposes, the lack of information regarding procedures to adopt in this type of studies also constitutes a limitation. However, through an extensive searching for relevant and appropriate information, I was able to tackle this limitation and collect the amount of knowledge required to the correct application of the concepts.

In spite of the aforementioned limitations, different models used to estimate the TSIR were applied to three different contexts to determine which of them provided the best results for each of the three European countries. The results point out to the undeniable strength of the Svensson and Adjusted Svensson models, which are the most applied by central banks. Furthermore, being the original framework to the development of the other models, the application of the Nelson and Siegel methodology results in satisfactory results. Finally, the Cubic Spline methodology proposed by McCulloch gives a different framework when compared to the remaining models, even though it produces the poorest results.

Regarding the well-know risk-return relationship, it was possible to conclude the existence of such relationship: i) the German TSIR provided a lower return but, on the other hand, has a lower credit risk and ii) the Spanish TSIR provided the highest return of the three countries but, on the other hand, has the highest credit risk.

7. Bibliography

Andritzky, J.R. 2012***, Government Bonds and Their Investors: What Are the Facts and Do They Matter?***, Working paper no. 12/158, International Monetary Fund (IMF).

Attinasi, M., Checherita, C. and Nickel, C. 2009***. What Explains the Surge in Euro Area Sovereign Spreads During the Financial Crisis of 2007-09***, Working Paper no. 1131, European Central Bank.

Fama, E.F. and Bliss, R.R. 1987, ***The Information in Long-Maturity Forward Rates, The American Economic Review*** 77: 680-692.

Borio, C. and R. McCauley.1996. ***The Economics of Recent Bond Yield Volatility***, BIS Economic Papers No. 45.

Chan, B.L., Duarte, A.A., Oliveira, L.V., Silva, A.F. and Weffort, E.F.J. 2015. The Term Structure of Interes Rates and its Impact on the Liability Adequacy Test for Insurance Companies in Brazil. ***Revista de Contabilidade & Finanças – USP***, vol. 26 :223 – 236.

Chirinos, A.M.A. and Moreno, M. 2009. ***Estimation of the Term Structure of Interest Rates: The Venezuelan Case***. Unpublished working paper.

Curto, J.J.D. 2016. ***Forecasting Methods Handouts***, Class Handouts, ISCTE-IUL, Lisbon.

Curto, J.J.D. 2016. ***Quantitative Methods for Finance Handouts***, Class Handouts, ISCTE-IUL, Lisbon.

De Pooter, M. 2007. ***Examining the Nelson-Siegel Class of Term Structure Models***, Tinbergen Institute Discussion Paper, Tinbergen.

Deboor, C.E. 1978. ***A Practical Guide to Splines***, New York: Sprienger-Verlag.

Dette, H. and Ziggel, D. 2006. ***Discount curve estimation by monotonizing McCulloch Splines***. Ruhr University Bochum, Germany.

Diebold, F.X. and Li, C. 2006. Forecasting the term structure of government bond yields, ***Journal of Econometrics,*** 130: 337-364.

Dobson, S.W. 1978. Estimating term structure equations with individual bond data, ***Journal of Finance*** 33, 75-92.

Duffie, D. and Kan, R. 1996. A Yield-Factor Model of Interest Rates, ***Mathematical Finance***, 6: 379-406.

Favero, C. and Missale, A. 2012. Sovereign spreads in the Eurozone: Which prospects for a Eurobond?, ***Economic Policy*** 70: 231-273.

Ferstl, R. and Hayden, J. 2010. Zero-Coupon Yield Curve Estimation with the Package termstrc, ***Journal of Statistical Software***, vol. 36

Filipovic, D. 1999. A Note on the Nelson-Siegel Family, ***Mathematical Finance*** 9: 349-359.

Fong, H.G. and Vasicek, O.A. 1982. Term Structure Modelling Using Exponential Splines, ***Journal of Finance*** 37: 339-356.

Huang, Y. and Su, W. 2010. ***Comparison of Multivariate GARCH Models with Application to Zero-Coupon Bond Volatility***, Unpublished Master Thesis, Lund University, Sweden.

Leañez, A.M.A.C and Moreno. M, ***Estimation of the Term Structure of Interest Rates: The Venezuelan Case***, Unpublised Paper, Venezuela

Lutz, F.A. 1940. The Structure of Interest Rates, ***Quarterly Journal of Economics***, 55: 36-63.

McCulloch J.H. 1975. The Tax-Adjusted Yield Curve***, The Journal of Finance***, 30: 811-830.

Nelson, C.R. and Siegel, A.F. 1987. Parsimonious Modelling of Yield Curves, ***Journal of Business,*** 60: 473-489.

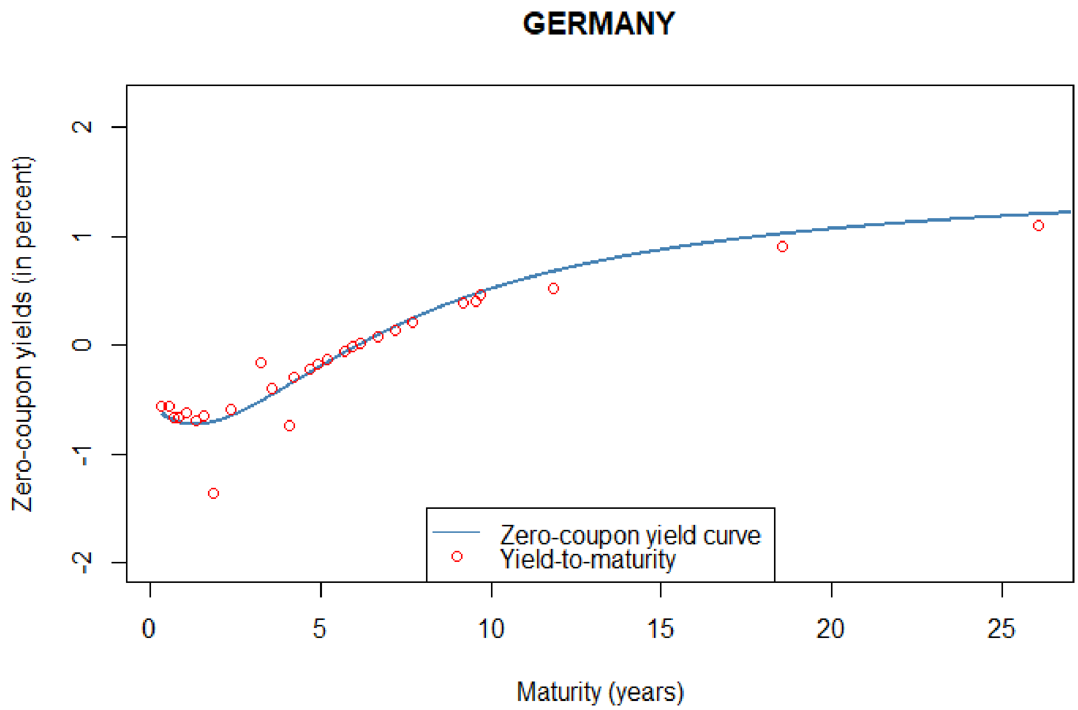
Schaefer, S.M. 1973. ***On Measuring the Term Structure of Interest Rates***, Discussion Paper, London Business School.

Shea, G.S. 1985. Interest Rate Term Structure Estimation with Exponential Splines: A Note, ***Journal of Finance***, 40: 319-325.

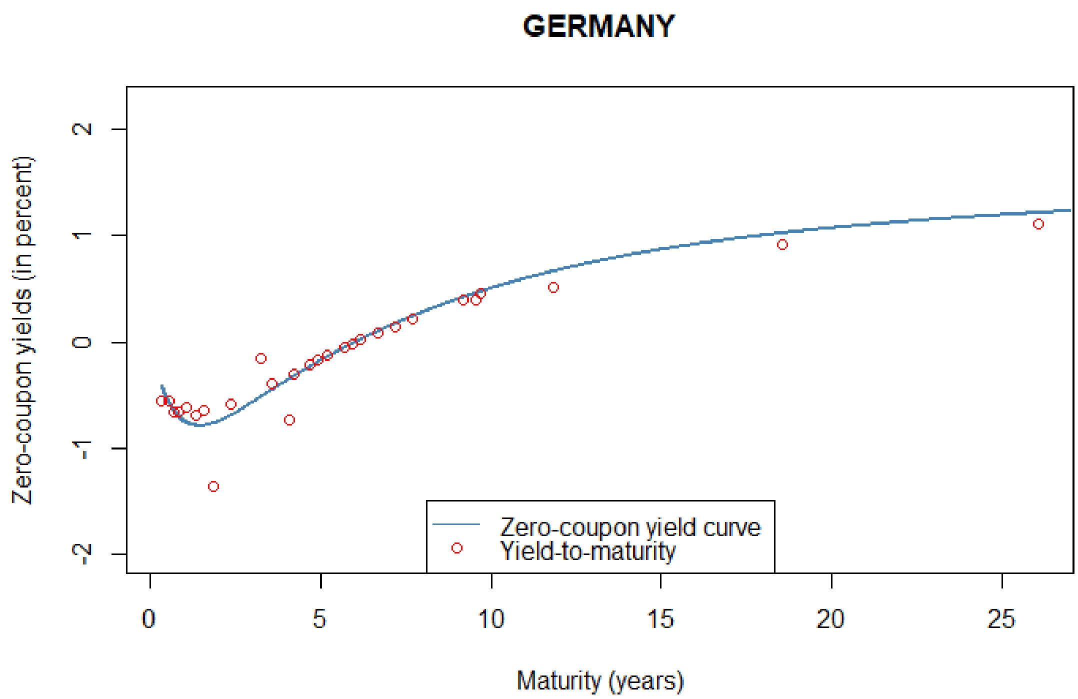
Svensson, L.E.O. 1994. ***Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994***, Working Paper No. 4871, National Bureau of Economic Research.

8. Appendix

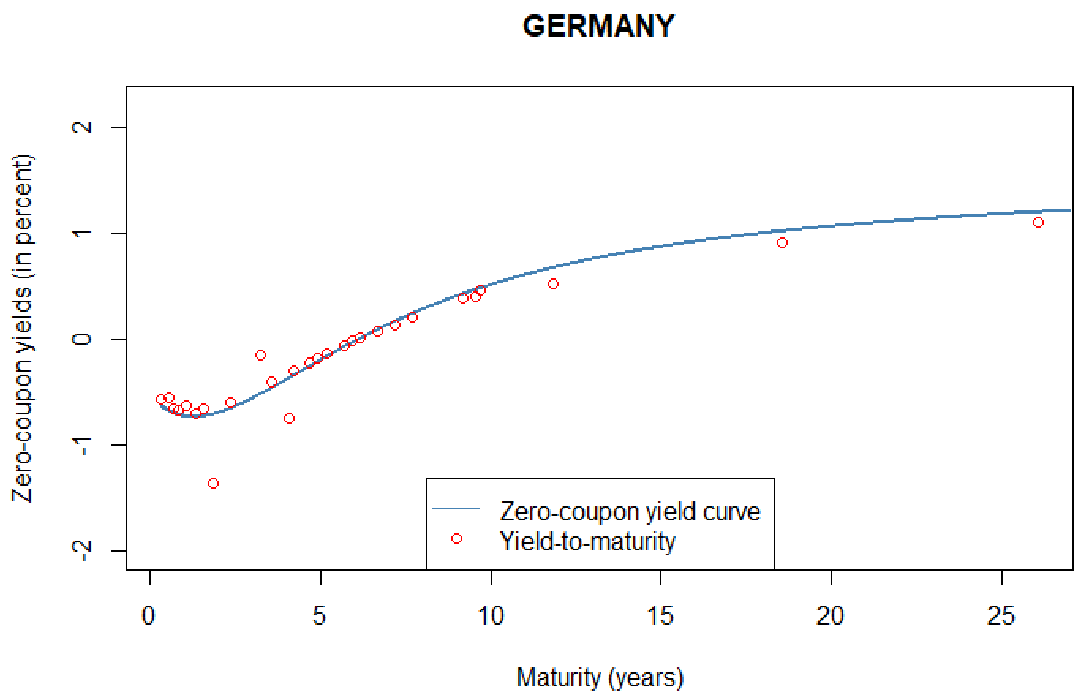
8.1. Appendix A – Plots



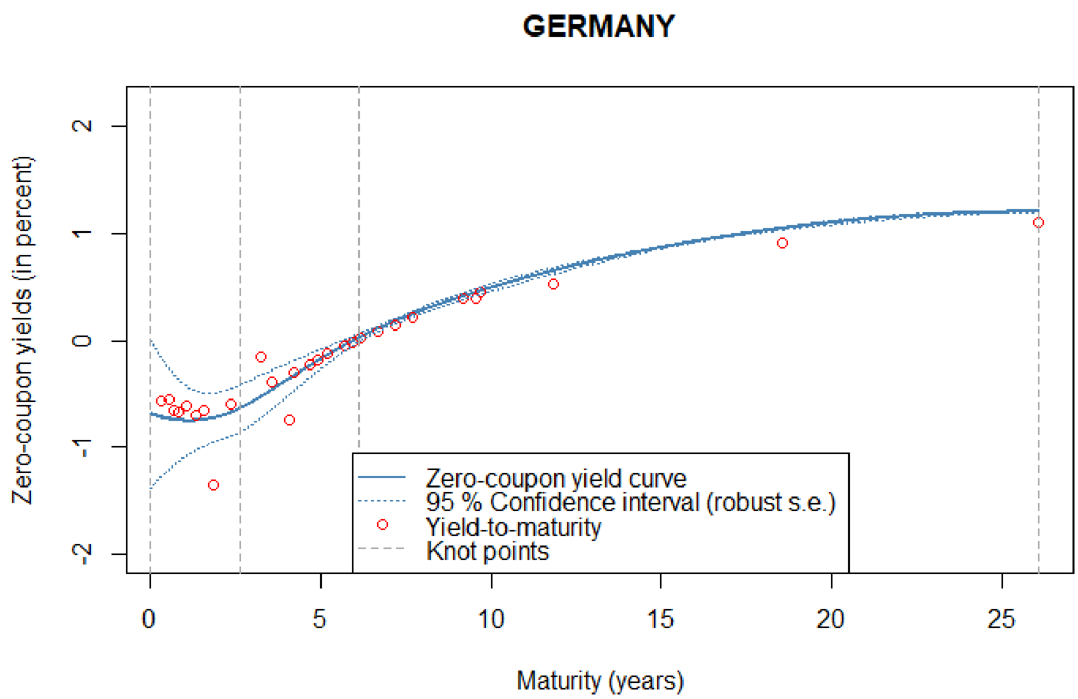
Plot 5 - Graphical Representation of the curve resulting from the application of the Edited NS function to the German sample.



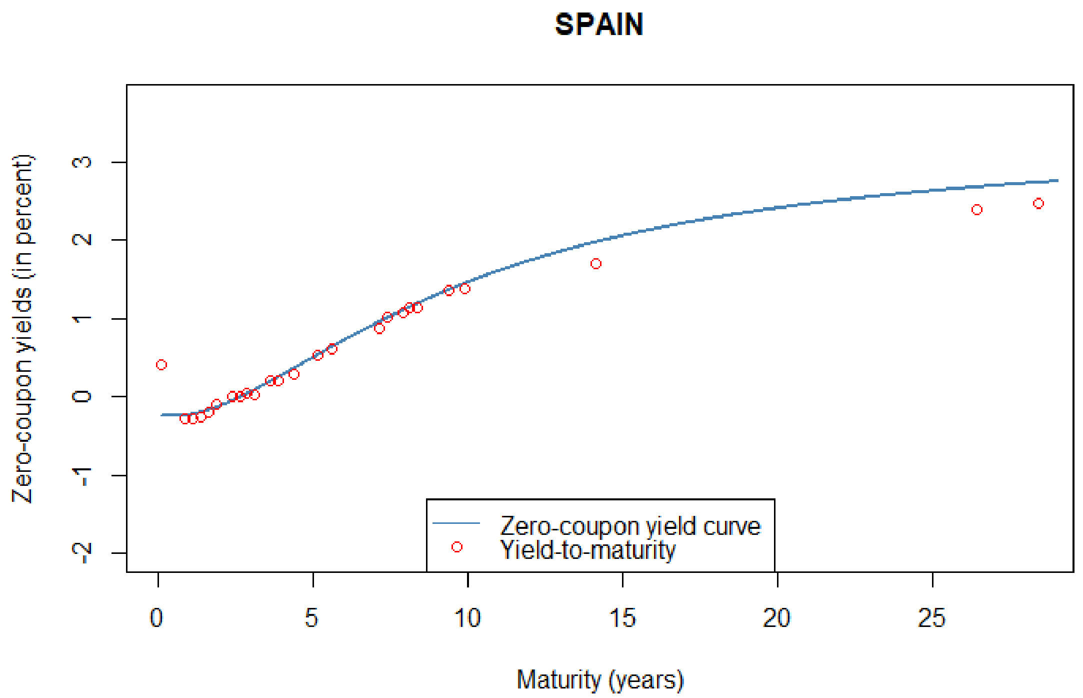
Plot 6 - Graphical Representation of the curve resulting from the application of the Regular ASV function to the German sample.



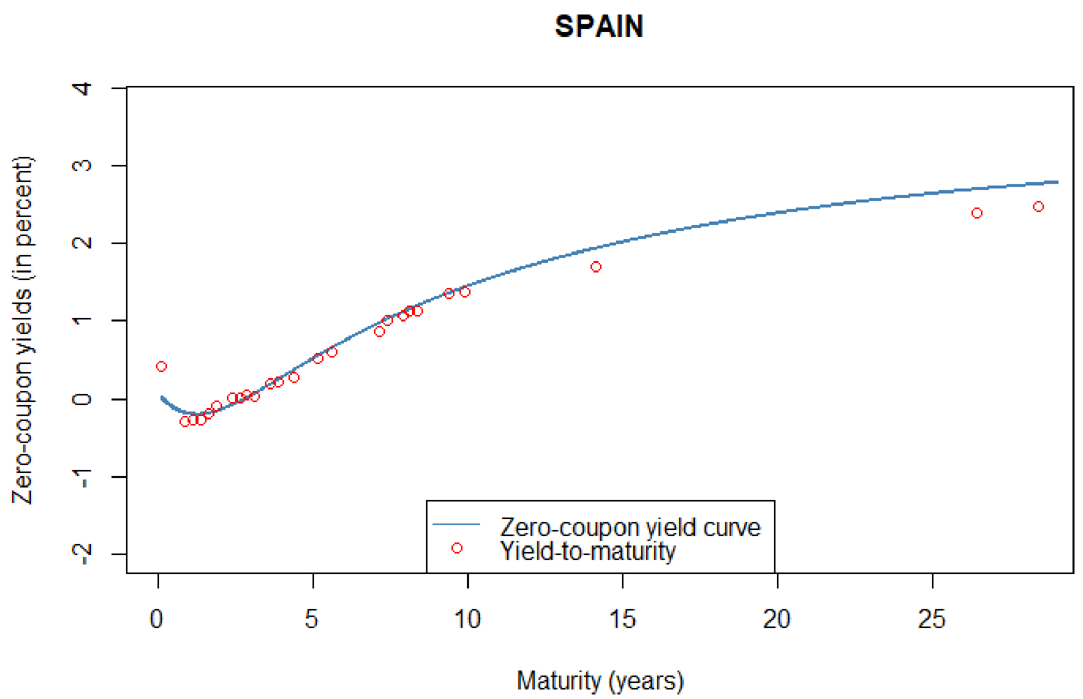
Plot 7 - Graphical Representation of the curve resulting from the application of the Edited DL function with a λ= 0,0417 to the German sample.



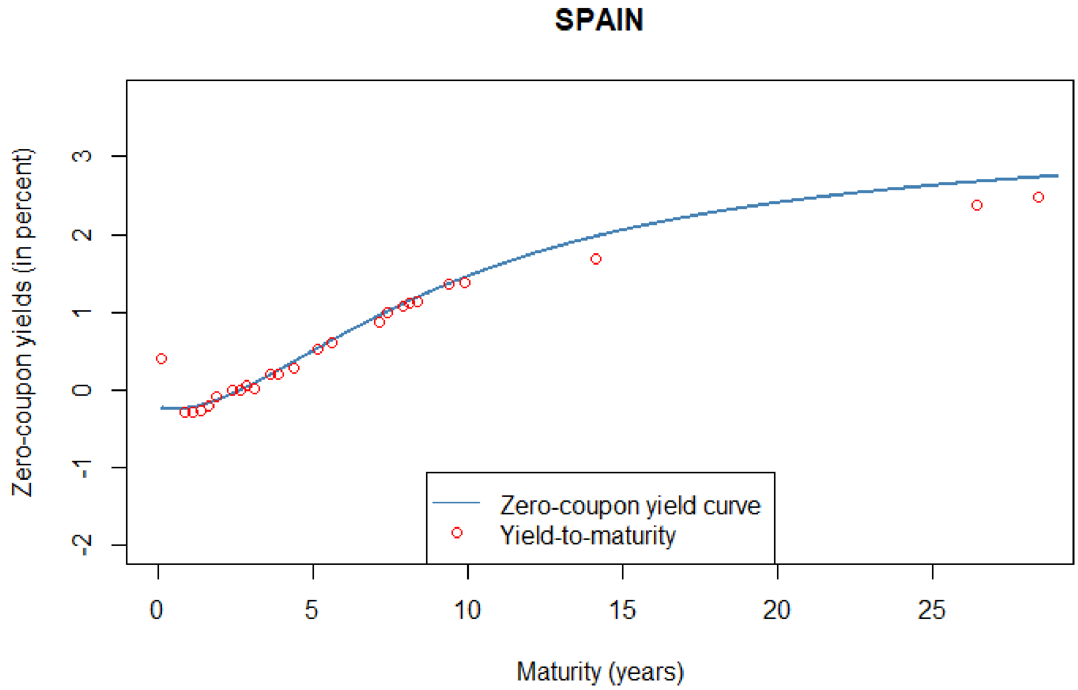
Plot 8- Graphical Representation of the curve resulting from the application of the CS methodology to the German sample.



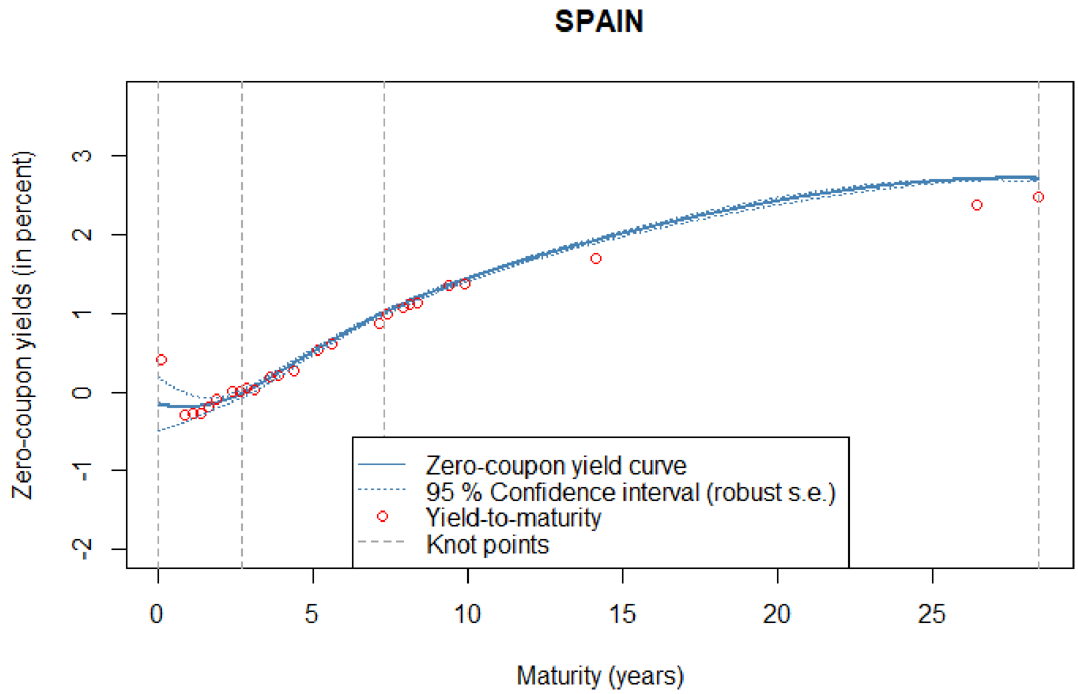
Plot 9 - Graphical Representation of the curve resulting from the application of the Edited NS function to the Spanish sample.



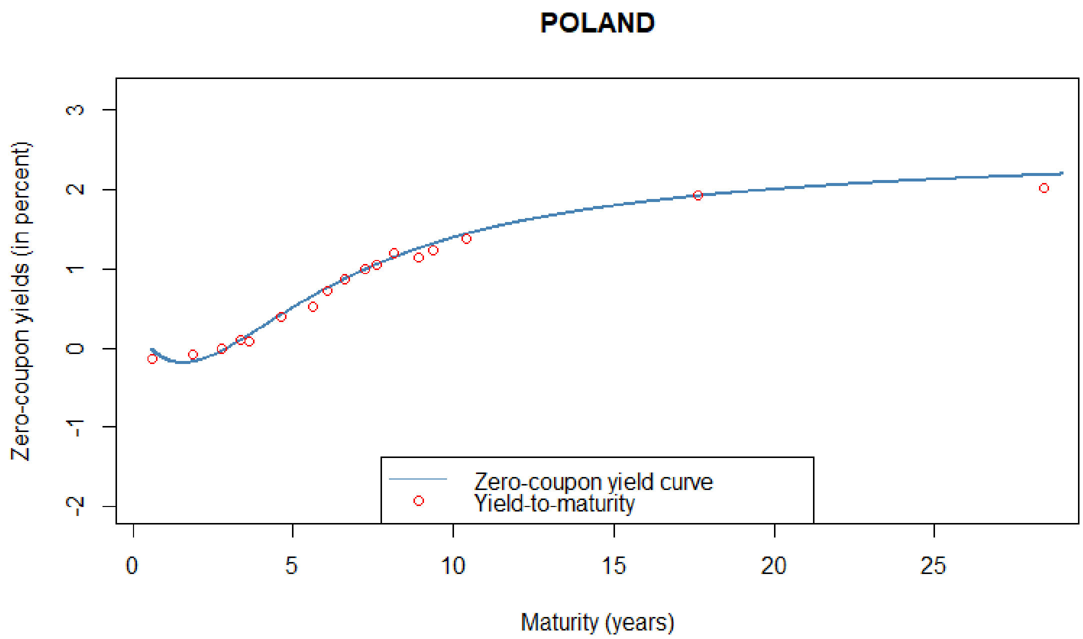
Plot 10 - Graphical Representation of the curve resulting from the application of the Regular SV function to the Spanish sample.



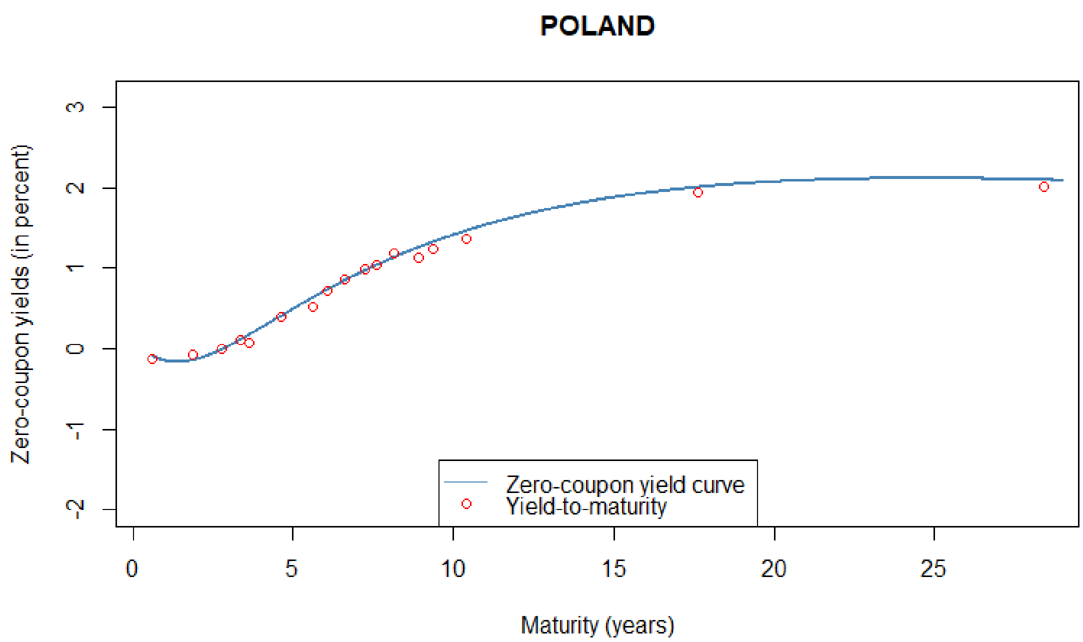
Plot 11- Graphical Representation of the curve resulting from the application of the Edited DL function with a λ = 0,0298 to the Spanish sample.



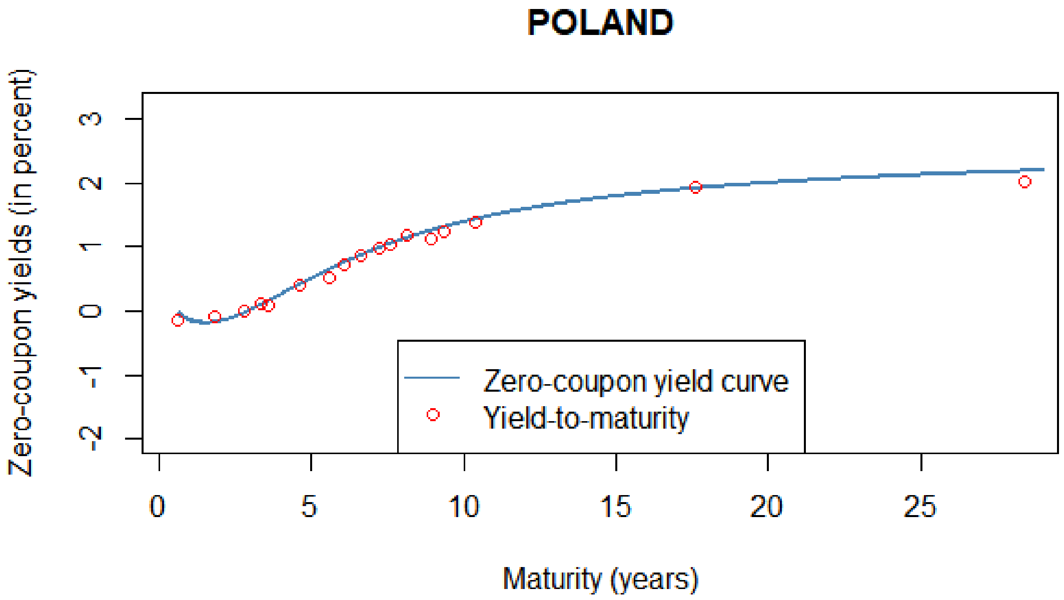
Plot 12 - Graphical Representation of the curve resulting from the application of the CS methodology to the Spanish sample.



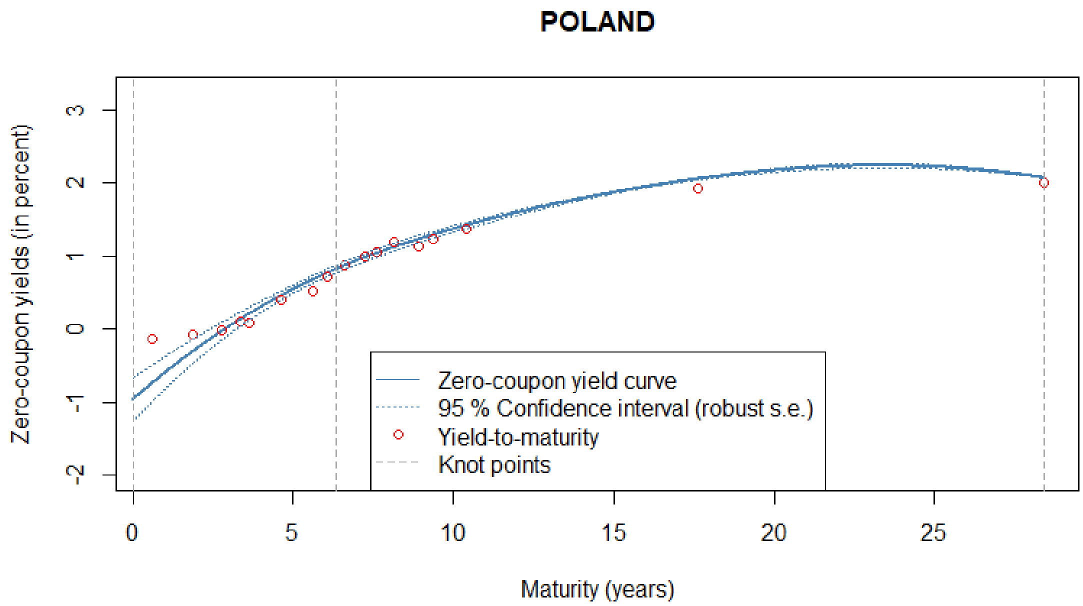
Plot 13- Graphical Representation of the curve resulting from the application of the Regular NS function to the Polish sample.



Plot 14 - Graphical Representation of the curve resulting from the application of the Regular SV function to the Polish sample.



Plot 15- Graphical Representation of the curve resulting from the application of the Edited DL function with a λ = 0,05 to the Polish sample



Plot 16- Graphical Representation of the curve resulting from the application of the CS methodology to the Polish sample.

8.2. Appendix B – Data

Germany

Table 12 - German bonds data

	Bond 1[5]	Bond 2	Bond 3	Bond 4	Bond 5	Bond 6
ISIN	DE0001135358	DE0001141679	DE0001135374	DE0001141687	DE0001141695	DE0001135382
MATURITY DATE	2018-07-04	2018-10-12	2019-01-04	2019-02-22	2019-04-12	2019-07-04
ISSUE DATE	2008-05-30	2013-09-06	2008-11-14	2014-01-17	2014-04-12	2009-05-22
COUPON RATE	0.0425	0.01	0.0375	0.01	0.005	0.035
PRICE	100.285	100.55	102.535	101.19	100.99	104.48
ACCRUED INTEREST	3.889	0.641	1.541	0.2767	0.0712	3.2027
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-07-04	2018-10-12	2019-01-04	2019-02-22	2019-04-12	2018-07-04
MATURITY	0,063013699	0,336986301	0,567123288	0,701369863	0,835616438	1,063013699

[5] The information with a red font corresponds to bonds that were excluded from the sample (either for being considered an outlier or for having a maturity higher than 30 years)

Table 33 - German bonds data (cont.)

	Bond 7	Bond 8	Bond 9	Bond 10	Bond 11	Bond 12
ISIN	DE0001141703	DE0001135390	DE0001030526	DE0001141729	DE0001135457	DE0001135465
MATURITY DATE	2019-10-11	2020-01-04	2020-04-15	2020-10-16	2021-09-04	2022-01-04
ISSUE DATE	2014-09-05	2009-11-13	2009-04-15	2015-07-03	2011-08-26	2011-11-25
COUPON RATE	0.0025	0.0325	0.01945	0.0025	0.025	0.02
PRICE	101.27	106.235	106.26	102	108.85	108.67
ACCRUED INTEREST	0.1609	1.3356	0.2349	0.1575	1.6767	0.8219
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-10-11	2019-01-04	2019-04-15	2018-10-16	2018-09-04	2019-01-04
MATURITY	1,334246575	1,567123288	1,846575342	2,350684932	3,235616438	3,569863014

Table 34 - German bonds data (cont.)

	Bond 13	Bond 14	Bond 15	Bond 16	Bond 17	Bond 18
ISIN	DE0001135473	DE0001135499	DE0001102309	DE0001030542	DE0001102317	DE0001102325
MATURITY DATE	2022-07-04	2022-09-04	2023-02-15	2023-04-15	2023-05-15	2023-08-15
ISSUE DATE	2012-04-13	2012-09-04	2013-01-18	2012-03-23	2013-05-15	2013-08-15
COUPON RATE	0.0175	0.015	0.015	0.00105	0.015	0.02
PRICE	108.58	107.7	108.14	107.89	108.33	111.11
ACCRUED INTEREST	1.6013	1.1178	0.4438	0.0134	0.078	1.6
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-07-04	2018-09-04	2019-02-15	2019-04-15	2019-05-15	2018-08-15
MATURITY	4,065753425	4,235616438	4,684931507	4,846575342	4,928767123	5,180821918

Table 35 - German bonds data (cont.)

	Bond 19	Bond 20	Bond 21	Bond 22	Bond 23	Bond 24
ISIN	DE0001102333	DE0001102358	DE0001102366	DE0001102374	DE0001102382	DE0001102390
MATURITY DATE	2024-02-15	2024-05-15	2024-08-15	2025-02-15	2025-08-15	2026-02-15
ISSUE DATE	2014-01-31	2014-05-15	2014-08-15	2015-01-16	2015-07-17	2016-01-15
COUPON RATE	0.0175	0.015	0.01	0.005	0.01	0.005
PRICE	110.29	109.04	106.09	102.82	106.16	102.22
ACCRUED INTEREST	0.5178	0.078	0.8	0.1479	0.8	0.1479
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-02-15	2019-05-15	2018-08-15	2019-02-15	2018-08-15	2019-02-15
MATURITY	5,684931507	5,931506849	6,183561644	6,687671233	7,183561644	7,687671233

Table 36 - German bonds data (cont.)

	Bond 25	Bond 26	Bond 27	Bond 28	Bond 29	Bond 30
ISIN	DE0001030567	DE0001102424	DE0001135069	DE0001102440	DE0001135143	DE0001030559
MATURITY DATE	2026-04-15	2027-08-15	2028-01-04	2028-02-15	2030-01-04	2030-04-15
ISSUE DATE	2015-03-12	2017-07-14	1998-01-04	2018-01-12	2000-01-04	2014-04-10
COUPON RATE	0.00102	0.005	0.05625	0.005	0.0625	0.00513
PRICE	110.36	100.91	149.074	100.34	164.07	116.8
ACCRUED INTEREST	0.0134	0.4438	2.3116	0.1945	2.5684	0.0671
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-04-15	2018-08-15	2019-01-04	2019-02-15	2019-04-01	2019-04-15
MATURITY	7,849315068	9,183561644	9,57260274	9,687671233	11,57534247	11,85205479

Table 37 - German bonds data (cont.)

	Bond 31	Bond 32	Bond 33
ISIN	DE0001135275	DE0001135481	DE0001102432
MATURITY DATE	2037-01-04	2044-07-04	2048-08-15
ISSUE DATE	2005-01-04	2012-04-27	2017-08-15
COUPON RATE	0.04	0.025	0.0125
PRICE	152.38	131.28	101.51
ACCRUED INTEREST	1.7315	2.3424	0.8972
TODAY	2018-06-11	2018-06-11	2018-06-11
CASHFLOWS (DATE OF NEXT PAYMENT)	2019-01-04	2018-07-04	2018-08-15
MATURITY	18,58082192	26,08219178	30,2

Spain

Table 38 - Spanish bonds data

	Bond 1	Bond 2	Bond 3	Bond 4	Bond 5	Bond 6
ISIN	ES00000121A5	ES00000124V5	ES00000121L2	ES00000121O6	ES00000126W8	ES00000126C0
MATURITY DATE	2018-07-30	2019-04-30	2019-07-30	2019-10-31	2019-11-30	2020-01-31
ISSUE DATE	2008-02-19	2014-01-14	2009-02-10	2009-06-02	2013-11-30	2014-07-08
COUPON RATE	0.041	0.0275	0.046	0.043	0.00561	0.014
PRICE	100.56	102.725	105.6	106.431	103.236	102.643
ACCRUED INTEREST	3.4821	0.2863	3.932	2.5564	0.2847	0.4871
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-07-30	2019-04-30	2018-07-30	2018-10-31	2018-11-30	2019-01-31
MATURITY	0,134246575	0,884931507	1,134246575	1,389041096	1,471232877	1,64109589

Table 392 - Spanish bonds data (cont.)

	Bond 7	Bond 8	Bond 9	Bond 10	Bond 11	Bond 12
ISIN	ES00000122D7	ES00000127H7	ES00000122T3	ES00000128X2	ES00000123B9	ES00000128B8
MATURITY DATE	2020-04-30	2020-07-30	2020-10-31	2021-01-31	2021-04-30	2021-07-30
ISSUE DATE	2010-02-20	2015-06-16	2010-07-13	2017-06-06	2011-01-24	2016-03-08
COUPON RATE	0.04	0.015	0.0485	0.0005	0.055	0.0075
PRICE	107.783	102.725	111.65	100.12	115.755	102.28
ACCRUED INTEREST	0.3945	0.983	2.8834	0.0173	0.5726	0.641
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-04-30	2018-07-30	2018-10-31	2019-01-31	2019-04-30	2018-07-30
MATURITY	1,887671233	2,136986301	2,391780822	2,643835616	2,887671233	3,136986301

Table 40 - Spanish bonds data (cont.)

	Bond 13	Bond 14	Bond 15	Bond 16	Bond 17	Bond 18
ISIN	ES00000128D4	ES00000123K0	ES00000128O1	ES0000012A97	ES0000012B62	ES00000121G2
MATURITY DATE	2021-11-30	2022-01-31	2022-04-30	2022-10-31	2023-07-30	2024-01-31
ISSUE DATE	2015-11-30	2011-11-22	2017-01-24	2017-10-10	2018-05-22	2008-09-16
COUPON RATE	0.00306	0.0585	0.004	0.0045	0.0035	0.048
PRICE	105.677	120.57	100.74	100.75	99.39	123.25
ACCRUED INTEREST	0.1553	2.0034	0.0416	0.27	0.0153	1.6438
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-11-30	2019-01-31	2019-04-30	2018-10-31	2018-07-30	2019-01-31
MATURITY	3,473972603	3,643835616	3,887671233	4,391780822	5,136986301	5,643835616

Table 413 - Spanish bonds data (cont.)

	Bond 19	Bond 20	Bond 21	Bond 22	Bond 23	Bond 24
ISIN	ES00000122E5	ES00000127G9	ES00000127Z9	ES00000123C7	ES00000128H5	ES0000012A89
MATURITY DATE	2025-07-30	2025-10-31	2026-04-30	2026-07-30	2026-10-31	2027-10-31
ISSUE DATE	2010-02-24	2015-06-09	2016-01-19	2011-03-15	2016-07-26	2017-07-04
COUPON RATE	0.0465	0.0215	0.0195	0.059	0.013	0.0145
PRICE	126.09	108.09	106.6	136.89	101.3	100.7
ACCRUED INTEREST	3.9493	1.29	0.203	5.0432	0.78	0.87
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-07-30	2018-10-31	2019-04-30	2018-07-30	2018-10-31	2018-10-31
MATURITY	7,139726027	7,394520548	7,890410959	8,139726027	8,394520548	9,394520548

Table 424 - Spanish bonds data (cont.)

	Bond 25	Bond 26	Bond 27	Bond 28	Bond 29
ISIN	ES0000012B39	ES0000012411	ES00000124H4	ES00000128C6	ES0000012B47
MATURITY DATE	2028-04-30	2032-07-30	2044-10-31	2046-10-31	2048-10-31
ISSUE DATE	2018-01-30	2002-07-30	2013-10-16	2016-03-15	2018-02-27
COUPON RATE	0.014	0.0575	0.0515	0.029	0.027
PRICE	100.07	150.45	152.86	107.85	102.73
ACCRUED INTEREST	0.1457	4.8835	3.1464	1.7717	0.7693
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-04-30	2018-07-30	2018-10-31	2018-10-31	2018-10-31
MATURITY	9,893150685	14,14520548	26,40821918	28,40821918	30,4109589

Poland

Table 435 - Polish bonds data

	Bond 1	Bond 2	Bond 3	Bond 4	Bond 5	Bond 6
ISIN	XS0874841066	XS0458008496	XS0210314299	XS0543882095	XS1306382364	XS0282701514
MATURITY DATE	2019-01-15	2019-10-15	2020-04-15	2021-03-23	2021-10-14	2022-01-18
ISSUE DATE	2013-01-15	2009-10-15	2005-01-18	2010-09-23	2015-10-14	2007-01-18
COUPON RATE	0.01625	0.04675	0.042	0.04	0.00875	0.045
PRICE	101.05	107.324	107.887	111.15	102.55	115.92
ACCRUED INTEREST	0.6544	3.0612	0.6559	0.8767	0.5753	1.7753
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-01-15	2018-10-15	2019-04-15	2019-03-23	2018-10-14	2019-01-18
MATURITY	0,597260274	1,345205479	1,846575342	2,783561644	3,345205479	3,608219178

Table 446 - Polish bonds data (cont.)

	Bond 7	Bond 8	Bond 9	Bond 10	Bond 11	Bond 12
ISIN	XS0794399674	XS1015428821	XS0841073793	XS0479333311	XS1288467605	XS1346201616
MATURITY DATE	2023-01-19	2024-01-15	2024-07-09	2025-01-20	2025-09-09	2026-01-19
ISSUE DATE	2012-06-18	2014-01-15	2012-10-09	2010-01-20	2015-09-09	2016-01-18
COUPON RATE	0.0375	0.03	0.03375	0.0525	0.015	0.015
PRICE	115.3	113.65	115.73	128	103.5	103.25
ACCRUED INTEREST	1.4692	1.2082	3.1161	2.0425	1.1301	0.5877
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2019-01-19	2019-01-15	2018-07-09	2019-01-20	2018-09-09	2019-01-19
MATURITY	4,610958904	5,6	6,082191781	6,616438356	7,252054795	7,61369863

Table 45 - Polish bonds data (cont.)

	Bond 13	Bond 14	Bond 15	Bond 16	Bond 17	Bond 18
ISIN	XS1766612672	XS1209947271	XS1584894650	XS1508566392	XS1346201889	XS1508566558
MATURITY DATE	2026-08-07	2027-05-10	2027-10-22	2028-10-25	2036-01-18	2046-10-25
ISSUE DATE	2018-02-07	2015-04-07	2017-03-23	2016-10-25	2016-01-18	2016-10-25
COUPON RATE	0.0125	0.00875	0.01375	0.01	0.02375	0.02
PRICE	0.0125	97.76	101.18	96.29	106.19	99.31
ACCRUED INTEREST	1.0548	0.0767	0.874	0.6274	0.937	1.2548
TODAY	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11	2018-06-11
DATE OF NEXT PAYMENT	2018-08-07	2019-05-10	2018-10-22	2018-10-25	2019-01-18	2018-10-25
MATURITY	8,161643836	8,917808219	9,369863014	10,38082192	17,61643836	28,39178082

8.3. Appendix C – R/RStudio Scripts

Script used to convert Excel Dates into R/RStudio Dates

```
#Convert Excel Dates into R/Rstudio Dates (GERMANY)

MaturityDatesExcelGERM <- c(43285, 43385, 43469, 43518, 43567, 43650,
43749, 43834, 43936, 44120, 44443, 44565, 44746, 44808, 44972, 45031,
45061, 45153, 45337,45427,  45519,  45703,  45884, 46068,   46127,
46614,  46756,  46798,  47487,  47588,  50044,  52782,  54285)
MaturityDatesRStudioGERM <- as.Date(MaturityDatesExcelGERM,origin =
"1899-12-30")

MaturityDatesRStudioGERMvalue <- as.numeric(MaturityDatesRStudioGERM)
MaturityDatesRStudioGERMvalue

IssueDatesExcelGERM <- c(39598, 41523,  39766,  41656,  41741,  39955,
41887,  40130,  39918,  42188,  40781,  40872,  41012,  41156,  41292,
40991,  41409,  41501,  41670,  41774,  41866,  42020,  42202,  42384,
42075,  42930,  35799,  43112,  36529,  41739,  38356,  41026,  42962)
IssueDatesRStudioGERM <- as.Date(IssueDatesExcelGERM, origin = "1899-
12-30")

IssueDatesRStudioGERMvalue <- as.numeric(IssueDatesRStudioGERM)
IssueDatesRStudioGERMvalue

CFDatesExcelGERM <- c(43285, 43385, 43469, 43518, 43567, 43285, 43650,
43384, 43749, 43469, 43834, 43570, 43936, 43389, 43754, 44120, 43347,
43712, 44078, 44443, 43469, 43834, 44200, 44565, 43650, 44016, 44381,
44746, 43347, 43712, 44078, 44443, 44808, 43511, 43876, 44242, 44607,
44972, 43570, 43936, 44301, 44666, 45031, 43600, 43966, 44331, 44696,
45061, 43327, 43692, 44058, 44423, 44788, 45153, 43511, 43876, 44242,
44607, 44972, 45337, 43600, 43966, 44331, 44696, 45061, 45427, 43327,
43692, 44058, 44423, 44788, 45153, 45519, 43511, 43876, 44242, 44607,
44972, 45337, 45703, 43327, 43692, 44058, 44423, 44788, 45153, 45519,
45884, 43511, 43876, 44242, 44607, 44972, 45337, 45703, 46068, 43570,
43936, 44301, 44666, 45031, 45397, 45762, 46127, 43327, 43692, 44058,
44423, 44788, 45153, 45519, 45884, 46249, 46614, 43469, 43834, 44200,
44565, 44930, 45295, 45661, 46026, 46391, 46756, 43511, 43876, 44242,
44607, 44972, 45337, 45703, 46068, 46433, 46798, 43556, 43922, 44287,
44652, 45017, 45383, 45748, 46113, 46478, 46844, 47209, 47574, 43570,
43936, 44301, 44666, 45031, 45397, 45762, 46127, 46492, 46858, 47223,
47588, 43469, 43834, 44200, 44565, 44930, 45295, 45661, 46026, 46391,
46756, 47122, 47487, 47852, 48217, 48583, 48948, 49313, 49678, 50044,
43285, 43650, 44016, 44381, 44746, 45111, 45477, 45842, 46207, 46572,
46938, 47303, 47668, 48033, 48399, 48764, 49129, 49494, 49860, 50225,
50590, 50955, 51321, 51686, 52051, 52416, 52782, 43327, 43692, 44058,
44423, 44788, 45153, 45519, 45884, 46249, 46614, 46980, 47345, 47710,
48075, 48441, 48806, 49171, 49536, 49902, 50267, 50632, 50997, 51363,
51728, 52093, 52458, 52824, 53189, 53554, 53919, 54285)
CFDatesRStudioGERM <- as.Date(CFDatesExcelGERM,origin = "1899-12-30")

CFDatesRStudioGERMvalue <- as.numeric(CFDatesRStudioGERM)
CFDatesRStudioGERMvalue

TodayDateExcel <- 43262
TodayDateRStudio <- as.Date(TodayDateExcel, origin = "1899-12-30")
TodayDateRStudiovalue <- as.numeric(TodayDateRStudio)
TodayDateRStudiovalue
```

```
#Convert Excel Dates into R/Rstudio Dates (SPAIN)

MaturityDatesExcelSPA <- c(43311, 43585, 43676, 43769, 43799,  43861,
43951, 44042, 44135, 44227, 44316, 44407, 44530, 44592, 44681, 44865,
45137, 45322, 45868, 45961, 46142, 46233, 46326, 46691, 46873, 48425,
52901, 53631, 54362)
MaturityDatesRStudioSPA <- as.Date(MaturityDatesExcelSPA,origin =
"1899-12-30")

MaturityDatesRStudioSPAvalue <- as.numeric(MaturityDatesRStudioSPA)
MaturityDatesRStudioSPAvalue

IssueDatesExcelSPA <- c(39497, 41653, 39854, 39966, 41608, 41828,
40229, 42171, 40372, 42892, 40567, 42437, 42338, 40869, 42759, 43018,
43242, 39707, 40233, 42164, 42388, 40617, 42577, 42920, 43130, 37467,
41563, 42444, 43158)
IssueDatesRStudioSPA <- as.Date(IssueDatesExcelSPA,origin = "1899-12-
30")

IssueDatesRStudioSPAvalue <- as.numeric(IssueDatesRStudioSPA)
IssueDatesRStudioSPAvalue

CFDatesExcelSPA <- c(43311, 43585, 43311, 43676, 43404, 43769, 43434,
43799, 43496, 43861, 43585, 43951, 43311, 43676, 44042, 43404, 43769,
44135, 43496, 43861, 44227, 43585, 43951, 44316, 43311, 43676, 44042,
44407, 43434, 43799, 44165, 44530, 43496, 43861, 44227, 44592, 43585,
43951, 44316, 44681, 43404, 43769, 44135, 44500, 44865, 43311, 43676,
44042, 44407, 44772, 45137, 43496, 43861, 44227, 44592, 44957, 45322,
43311, 43676, 44042, 44407, 44772, 45137, 45503, 45868, 43404, 43769,
44135, 44500, 44865, 45230, 45596, 45961, 43585, 43951, 44316, 44681,
45046, 45412, 45777, 46142, 43311, 43676, 44042, 44407, 44772, 45137,
45503, 45868, 46233, 43404, 43769, 44135, 44500, 44865, 45230, 45596,
45961, 46326, 43404, 43769, 44135, 44500, 44865, 45230, 45596, 45961,
46326, 46691, 43585, 43951, 44316, 44681, 45046, 45412, 45777, 46142,
46507, 46873, 43311, 43676, 44042, 44407, 44772, 45137, 45503, 45868,
46233, 46598, 46964, 47329, 47694, 48059, 48425, 43404, 43769, 44135,
44500, 44865, 45230, 45596, 45961, 46326, 46691, 47057, 47422, 47787,
48152, 48518, 48883, 49248, 49613, 49979, 50344, 50709, 51074, 51440,
51805, 52170, 52535, 52901, 43404, 43769, 44135, 44500, 44865, 45230,
45596, 45961, 46326, 46691, 47057, 47422, 47787, 48152, 48518, 48883,
49248, 49613, 49979, 50344, 50709, 51074, 51440, 51805, 52170, 52535,
52901, 53266, 53631, 43404, 43769, 44135, 44500, 44865, 45230, 45596,
45961, 46326, 46691, 47057, 47422, 47787, 48152, 48518, 48883, 49248,
49613, 49979, 50344, 50709, 51074, 51440, 51805, 52170, 52535, 52901,
53266, 53631, 53996, 54362)
CFDatesRStudioSPA <- as.Date(CFDatesExcelSPA,origin = "1899-12-30")

CFDatesRStudioSPAvalue <- as.numeric(CFDatesRStudioSPA)
CFDatesRStudioSPAvalue
```

```
#Convert Excel Dates into R/Rstudio Dates (POLAND)

MaturityDatesExcelPOL <- c(43480, 43753, 43936, 44278, 44483, 44579,
44945, 45306, 45482, 45677, 45909, 46041, 46241, 46517, 46682, 47051,
49692, 53625)
MaturityDatesRStudioPOL <- as.Date(MaturityDatesExcelPOL,origin =
"1899-12-30")

MaturityDatesRStudioPOLvalue <- as.numeric(MaturityDatesRStudioPOL)
MaturityDatesRStudioPOLvalue

IssueDatesExcelPOL <- c(41289, 40101, 38370, 40444, 42291, 39100,
41078, 41654, 41191, 40198, 42256, 42387, 43138, 42101, 42817, 42668,
42387, 42668)
IssueDatesRStudioPOL <- as.Date(IssueDatesExcelPOL, origin = "1899-12-
30")

IssueDatesRStudioPOLvalue <- as.numeric(IssueDatesRStudioPOL)
IssueDatesRStudioPOLvalue

CFDatesExcelPOL <- c(43480, 43388, 43753, 43570, 43936, 43547, 43913,
44278, 43387, 43752, 44118, 44483, 43483, 43848, 44214, 44579, 43484,
43849, 44215, 44580, 44945, 43480, 43845, 44211, 44576, 44941, 45306,
43290, 43655, 44021, 44386, 44751, 45116, 45482, 43485, 43850, 44216,
44581, 44946, 45311, 45677, 43352, 43717, 44083, 44448, 44813, 45178,
45544, 45909, 43484, 43849, 44215, 44580, 44945, 45310, 45676, 46041,
43319, 43684, 44050, 44415, 44780, 45145, 45511, 45876, 46241, 43595,
43961, 44326, 44691, 45056, 45422, 45787, 46152, 46517, 43395, 43760,
44126, 44491, 44856, 45221, 45587, 45952, 46317, 46682, 43398, 43763,
44129, 44494, 44859, 45224, 45590, 45955, 46320, 46685, 47051, 43483,
43848, 44214, 44579, 44944, 45309, 45675, 46040, 46405, 46770, 47136,
47501, 47866, 48231, 48597, 48962, 49327, 49692, 43398, 43763, 44129,
44494, 44859, 45224, 45590, 45955, 46320, 46685, 47051, 47416, 47781,
48146, 48512, 48877, 49242, 49607, 49973, 50338, 50703, 51068, 51434,
51799, 52164, 52529, 52895, 53260, 53625)
CFDatesRStudioPOL <- as.Date(CFDatesExcelPOL,origin = "1899-12-30")

CFDatesRStudioPOLvalue <- as.numeric(CFDatesRStudioPOL)
CFDatesRStudioPOLvalue
```

Script used to Upload the Data

```
DataSet <- structure(list(GERMANY = structure(list(ISIN =
c("DE0001135358",
                                                  "DE0001141679",
"DE0001135374", "DE0001141687", "DE0001141695",
                                                  "DE0001135382",
"DE0001141703", "DE0001135390", "DE0001030526",
                                                  "DE0001141729",
"DE0001135457", "DE0001135465", "DE0001135473",
                                                  "DE0001135499",
"DE0001102309", "DE0001030542", "DE0001102317",
                                                  "DE0001102325",
"DE0001102333", "DE0001102358", "DE0001102366",
                                                  "DE0001102374",
"DE0001102382", "DE0001102390", "DE0001030567",
                                                  "DE0001102424",
"DE0001135069", "DE0001102440", "DE0001135143",
                                                  "DE0001030559",
"DE0001135275", "DE0001135481", "DE0001102432"), MATURITYDATE =
structure(c(17716, 17816, 17900, 17949, 17998,
                              18081, 18180, 18265, 18367, 18551,
18874, 18996, 19177, 19239,
                              19403, 19462, 19492, 19584, 19768,
19858, 19950, 20134, 20315,
                              20499, 20558, 21045, 21187, 21229,
21918, 22019, 24475, 27213,
                              28716), class = "Date"), ISSUEDATE =
structure(c(14029, 15954,

14197, 16087, 16172, 14386, 16318, 14561, 14349, 16619, 15212,

15303, 15443, 15587, 15723, 15422, 15840, 15932, 16101, 16205,

16297, 16451, 16633, 16815, 16506, 17361, 10230, 17543, 10960,

16170, 12787, 15457, 17393), class = "Date"), COUPONRATE = c(0.0425,

0.01, 0.0375, 0.01, 0.005, 0.035, 0.0025, 0.0325, 0.01945, 0.0025,

0.025, 0.02, 0.0175, 0.015, 0.015, 0.00105, 0.015, 0.02, 0.0175,

0.015, 0.01, 0.005, 0.01, 0.005, 0.00102, 0.005, 0.05625, 0.005,

0.0625, 0.00513, 0.04, 0.025, 0.0125), PRICE = c(100.285, 100.55,

102.535, 101.19, 100.99, 104.48, 101.27, 106.235, 106.26, 102,

108.85, 108.67, 108.58, 107.7, 108.14, 107.89, 108.33, 111.11,

110.29, 109.04, 106.09, 102.82, 106.16, 102.22, 110.36, 100.91,

149.074, 100.34, 164.07, 116.8, 152.38, 131.28, 101.51), ACCRUED =
c(3.889,

0.641, 1.541, 0.2767, 0.0712, 3.2027, 0.1609, 1.3356, 0.2349,

0.1575, 1.6767, 0.8219, 1.6013, 1.1178, 0.4438, 0.0134, 0.078,

1.6, 0.5178, 0.078, 0.8, 0.1479, 0.8, 0.1479, 0.0134, 0.4438,
```

```
2.3116, 0.1945, 2.5684, 0.0671, 1.7315, 2.3424, 0.8972), CASHFLOWS =
structure(list(
ISIN = c("DE0001135358", "DE0001141679", "DE0001135374",
"DE0001141687", "DE0001141695", "DE0001135382", "DE0001135382",
"DE0001141703", "DE0001141703", "DE0001135390", "DE0001135390",
"DE0001030526", "DE0001030526", "DE0001141729", "DE0001141729",
"DE0001141729", "DE0001135457", "DE0001135457", "DE0001135457",
"DE0001135457", "DE0001135465", "DE0001135465", "DE0001135465",
"DE0001135465", "DE0001135473", "DE0001135473", "DE0001135473",
"DE0001135473", "DE0001135499", "DE0001135499", "DE0001135499",
"DE0001135499", "DE0001135499", "DE0001102309", "DE0001102309",
"DE0001102309", "DE0001102309", "DE0001102309", "DE0001030542",
"DE0001030542", "DE0001030542", "DE0001030542", "DE0001030542",
"DE0001102317", "DE0001102317", "DE0001102317", "DE0001102317",
"DE0001102317", "DE0001102325", "DE0001102325", "DE0001102325",
"DE0001102325", "DE0001102325", "DE0001102325", "DE0001102333",
"DE0001102333", "DE0001102333", "DE0001102333", "DE0001102333",
"DE0001102333", "DE0001102358", "DE0001102358", "DE0001102358",
"DE0001102358", "DE0001102358", "DE0001102358", "DE0001102366",
"DE0001102366", "DE0001102366", "DE0001102366", "DE0001102366",
"DE0001102366", "DE0001102366", "DE0001102374", "DE0001102374",
"DE0001102374", "DE0001102374", "DE0001102374", "DE0001102374",
"DE0001102374", "DE0001102382", "DE0001102382", "DE0001102382",
"DE0001102382", "DE0001102382", "DE0001102382", "DE0001102382",
"DE0001102382", "DE0001102390", "DE0001102390", "DE0001102390",
"DE0001102390", "DE0001102390", "DE0001102390", "DE0001102390",
"DE0001102390", "DE0001030567", "DE0001030567", "DE0001030567",
"DE0001030567", "DE0001030567", "DE0001030567", "DE0001030567",
"DE0001030567", "DE0001102424", "DE0001102424", "DE0001102424",
"DE0001102424", "DE0001102424", "DE0001102424", "DE0001102424",
"DE0001102424", "DE0001102424", "DE0001102424", "DE0001135069",
```

```
"DE0001135069", "DE0001135069", "DE0001135069", "DE0001135069",
"DE0001135069", "DE0001135069", "DE0001135069", "DE0001135069",
"DE0001135069", "DE0001102440", "DE0001102440", "DE0001102440",
"DE0001102440", "DE0001102440", "DE0001102440", "DE0001102440",
"DE0001102440", "DE0001102440", "DE0001102440", "DE0001135143",
"DE0001135143", "DE0001135143", "DE0001135143", "DE0001135143",
"DE0001135143", "DE0001135143", "DE0001135143", "DE0001135143",
"DE0001135143", "DE0001135143", "DE0001135143", "DE0001030559",
"DE0001030559", "DE0001030559", "DE0001030559", "DE0001030559",
"DE0001030559", "DE0001030559", "DE0001030559", "DE0001030559",
"DE0001030559", "DE0001030559", "DE0001030559", "DE0001135275",
"DE0001135275", "DE0001135275", "DE0001135275", "DE0001135275",
"DE0001135275", "DE0001135275", "DE0001135275", "DE0001135275",
"DE0001135275", "DE0001135275", "DE0001135275", "DE0001135275",
"DE0001135275", "DE0001135275", "DE0001135275", "DE0001135275",
"DE0001135275", "DE0001135275", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001135481", "DE0001135481", "DE0001135481",
"DE0001135481", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432",
"DE0001102432", "DE0001102432", "DE0001102432", "DE0001102432"
```

```
), CF = c(104.25, 101, 103.75, 101, 100.5, 3.5, 103.5, 0.25,
100.25, 3.25, 103.25, 1.945, 101.945, 0.25, 0.25, 100.25,
2.5, 2.5, 2.5, 102.5, 2, 2, 2, 102, 1.75, 1.75, 1.75, 101.75,
1.5, 1.5, 1.5, 1.5, 101.5, 1.5, 1.5, 1.5, 1.5, 101.5, 0.105,
0.105, 0.105, 0.105, 100.105, 1.5, 1.5, 1.5, 1.5, 101.5,
2, 2, 2, 2, 2, 102, 1.75, 1.75, 1.75, 1.75, 1.75, 101.75,
1.5, 1.5, 1.5, 1.5, 1.5, 101.5, 1, 1, 1, 1, 1, 1, 101, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 100.5, 1, 1, 1, 1, 1, 1, 1, 101,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 100.5, 0.102, 0.102, 0.102,
0.102, 0.102, 0.102, 0.102, 100.102, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 100.5, 5.625, 5.625, 5.625, 5.625,
5.625, 5.625, 5.625, 5.625, 5.625, 105.625, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 100.5, 6.25, 6.25, 6.25, 6.25,
6.25, 6.25, 6.25, 6.25, 6.25, 6.25, 6.25, 106.25, 0.513,
0.513, 0.513, 0.513, 0.513, 0.513, 0.513, 0.513, 0.513, 0.513,
0.513, 100.513, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 104, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5,
2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 2.5,
2.5, 2.5, 2.5, 2.5, 2.5, 2.5, 102.5, 1.25, 1.25, 1.25, 1.25,
1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25,
1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25,
1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 101.25), DATE = structure(c(17716,
17816, 17900, 17949, 17998, 17716, 18081, 17815, 18180, 17900,
18265, 18001, 18367, 17820, 18185, 18551, 17778, 18143, 18509,
18874, 17900, 18265, 18631, 18996, 18081, 18447, 18812, 19177,
17778, 18143, 18509, 18874, 19239, 17942, 18307, 18673, 19038,
19403, 18001, 18367, 18732, 19097, 19462, 18031, 18397, 18762,
19127, 19492, 17758, 18123, 18489, 18854, 19219, 19584, 17942,
18307, 18673, 19038, 19403, 19768, 18031, 18397, 18762, 19127,
19492, 19858, 17758, 18123, 18489, 18854, 19219, 19584, 19950,
```

```
17942, 18307, 18673, 19038, 19403, 19768, 20134, 17758, 18123,
18489, 18854, 19219, 19584, 19950, 20315, 17942, 18307, 18673,
19038, 19403, 19768, 20134, 20499, 18001, 18367, 18732, 19097,
19462, 19828, 20193, 20558, 17758, 18123, 18489, 18854, 19219,
19584, 19950, 20315, 20680, 21045, 17900, 18265, 18631, 18996,
19361, 19726, 20092, 20457, 20822, 21187, 17942, 18307, 18673,
19038, 19403, 19768, 20134, 20499, 20864, 21229, 17987, 18353,
18718, 19083, 19448, 19814, 20179, 20544, 20909, 21275, 21640,
22005, 18001, 18367, 18732, 19097, 19462, 19828, 20193, 20558,
20923, 21289, 21654, 22019, 17900, 18265, 18631, 18996, 19361,
19726, 20092, 20457, 20822, 21187, 21553, 21918, 22283, 22648,
23014, 23379, 23744, 24109, 24475, 17716, 18081, 18447, 18812,
19177, 19542, 19908, 20273, 20638, 21003, 21369, 21734, 22099,
22464, 22830, 23195, 23560, 23925, 24291, 24656, 25021, 25386,
25752, 26117, 26482, 26847, 27213, 17758, 18123, 18489, 18854,
19219, 19584, 19950, 20315, 20680, 21045, 21411, 21776, 22141,
22506, 22872, 23237, 23602, 23967, 24333, 24698, 25063, 25428,
25794, 26159, 26524, 26889, 27255, 27620, 27985, 28350, 28716
), class = "Date")), .Names = c("ISIN", "CF", "DATE")), TODAY =
structure(17693, class = "Date")), .Names = c("ISIN",
"MATURITYDATE", "ISSUEDATE", "COUPONRATE", "PRICE", "ACCRUED",
"CASHFLOWS", "TODAY")), SPAIN = structure(list(ISIN =
c("ES00000121A5",
"ES00000124V5", "ES00000121L2", "ES00000121O6", "ES00000126W8",
"ES00000126C0", "ES00000122D7", "ES00000127H7", "ES00000122T3",
"ES00000128X2", "ES00000123B9", "ES00000128B8", "ES00000128D4",
"ES00000123K0", "ES00000128O1", "ES0000012A97", "ES0000012B62",
"ES00000121G2", "ES00000122E5", "ES00000127G9", "ES00000127Z9",
"ES00000123C7", "ES00000128H5", "ES0000012A89", "ES0000012B39",
"ES0000012411", "ES00000124H4", "ES00000128C6", "ES0000012B47"
), MATURITYDATE = structure(c(17742, 18016, 18107, 18200, 18230,
```

```
18292, 18382, 18473, 18566, 18658, 18747, 18838, 18961, 19023,

19112, 19296, 19568, 19753, 20299, 20392, 20573, 20664, 20757,

21122, 21304, 22856, 27332, 28062, 28793), class = "Date"), ISSUEDATE
= structure(c(13928,

16084, 14285, 14397, 16039, 16259, 14660, 16602, 14803, 17323,

14998, 16868, 16769, 15300, 17190, 17449, 17673, 14138, 14664,

16595, 16819, 15048, 17008, 17351, 17561, 11898, 15994, 16875,

17589), class = "Date"), COUPONRATE = c(0.041, 0.0275, 0.046,

0.043, 0.00561, 0.014, 0.04, 0.015, 0.0485, 5e-04, 0.055, 0.0075,

0.00306, 0.0585, 0.004, 0.0045, 0.0035, 0.048, 0.0465, 0.0215,

0.0195, 0.059, 0.013, 0.0145, 0.014, 0.0575, 0.0515, 0.029, 0.027

), PRICE = c(100.56, 102.725, 105.6, 106.431, 103.236, 102.643,

107.783, 102.725, 111.65, 100.12, 115.755, 102.28, 105.677, 120.57,

100.74, 100.75, 99.39, 123.25, 126.09, 108.09, 106.6, 136.89,

101.3, 100.7, 100.07, 150.45, 152.86, 107.85, 102.73), ACCRUED =
c(3.4821,

0.2863, 3.932, 2.5564, 0.2847, 0.4871, 0.3945, 0.983, 2.8834,

0.0173, 0.5726, 0.641, 0.1553, 2.0034, 0.0416, 0.27, 0.0153,

1.6438, 3.9493, 1.29, 0.203, 5.0432, 0.78, 0.87, 0.1457, 4.8835,

3.1464, 1.7717, 0.7693), CASHFLOWS = structure(list(ISIN =
c("ES00000121A5",

"ES00000124V5", "ES00000121L2", "ES00000121L2", "ES0000012106",

"ES0000012106", "ES00000126W8", "ES00000126W8", "ES00000126C0",

"ES00000126C0", "ES00000122D7", "ES00000122D7", "ES00000127H7",

"ES00000127H7", "ES00000127H7", "ES00000122T3", "ES00000122T3",

"ES00000122T3", "ES00000128X2", "ES00000128X2", "ES00000128X2",

"ES00000123B9", "ES00000123B9", "ES00000123B9", "ES00000128B8",

"ES00000128B8", "ES00000128B8", "ES00000128B8", "ES00000128D4",

"ES00000128D4", "ES00000128D4", "ES00000128D4", "ES00000123K0",

"ES00000123K0", "ES00000123K0", "ES00000123K0", "ES00000128O1",

"ES00000128O1", "ES00000128O1", "ES00000128O1", "ES0000012A97",

"ES0000012A97", "ES0000012A97", "ES0000012A97", "ES0000012A97",
```

```
"ES0000012B62", "ES0000012B62", "ES0000012B62", "ES0000012B62",
"ES0000012B62", "ES0000012B62", "ES00000121G2", "ES00000121G2",
"ES00000121G2", "ES00000121G2", "ES00000121G2", "ES00000121G2",
"ES00000122E5", "ES00000122E5", "ES00000122E5", "ES00000122E5",
"ES00000122E5", "ES00000122E5", "ES00000122E5", "ES00000122E5",
"ES00000127G9", "ES00000127G9", "ES00000127G9", "ES00000127G9",
"ES00000127G9", "ES00000127G9", "ES00000127G9", "ES00000127G9",
"ES00000127Z9", "ES00000127Z9", "ES00000127Z9", "ES00000127Z9",
"ES00000127Z9", "ES00000127Z9", "ES00000127Z9", "ES00000127Z9",
"ES00000123C7", "ES00000123C7", "ES00000123C7", "ES00000123C7",
"ES00000123C7", "ES00000123C7", "ES00000123C7", "ES00000123C7",
"ES00000123C7", "ES00000128H5", "ES00000128H5", "ES00000128H5",
"ES00000128H5", "ES00000128H5", "ES00000128H5", "ES00000128H5",
"ES00000128H5", "ES00000128H5", "ES0000012A89", "ES0000012A89",
"ES0000012A89", "ES0000012A89", "ES0000012A89", "ES0000012A89",
"ES0000012A89", "ES0000012A89", "ES0000012A89", "ES0000012A89",
"ES0000012B39", "ES0000012B39", "ES0000012B39", "ES0000012B39",
"ES0000012B39", "ES0000012B39", "ES0000012B39", "ES0000012B39",
"ES0000012B39", "ES0000012B39", "ES0000012411", "ES0000012411",
"ES0000012411", "ES0000012411", "ES0000012411", "ES0000012411",
"ES0000012411", "ES0000012411", "ES0000012411", "ES0000012411",
"ES0000012411", "ES0000012411", "ES0000012411", "ES0000012411",
"ES0000012411", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000124H4", "ES00000124H4", "ES00000124H4", "ES00000124H4",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
```

```
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES00000128C6", "ES00000128C6", "ES00000128C6",
"ES00000128C6", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47",
"ES0000012B47", "ES0000012B47", "ES0000012B47", "ES0000012B47"
), CF = c(104.1, 102.75, 4.6, 104.6, 4.3, 104.3, 0.561, 100.561,
1.4, 101.4, 4, 104, 1.5, 1.5, 101.5, 4.85, 4.85, 104.85, 0.05,
0.05, 100.05, 5.5, 5.5, 105.5, 0.75, 0.75, 0.75, 100.75, 0.306,
0.306, 0.306, 100.306, 5.85, 5.85, 5.85, 105.85, 0.4, 0.4, 0.4,
100.4, 0.45, 0.45, 0.45, 0.45, 100.45, 0.35, 0.35, 0.35, 0.35,
0.35, 100.35, 4.8, 4.8, 4.8, 4.8, 4.8, 104.8, 4.65, 4.65, 4.65,
4.65, 4.65, 4.65, 4.65, 104.65, 2.15, 2.15, 2.15, 2.15, 2.15,
2.15, 2.15, 102.15, 1.95, 1.95, 1.95, 1.95, 1.95, 1.95, 1.95,
101.95, 5.9, 5.9, 5.9, 5.9, 5.9, 5.9, 5.9, 5.9, 105.9, 1.3, 1.3,
1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 101.3, 1.45, 1.45, 1.45, 1.45,
1.45, 1.45, 1.45, 1.45, 1.45, 101.45, 1.4, 1.4, 1.4, 1.4, 1.4,
1.4, 1.4, 1.4, 1.4, 101.4, 5.75, 5.75, 5.75, 5.75, 5.75, 5.75,
5.75, 5.75, 5.75, 5.75, 5.75, 5.75, 5.75, 5.75, 105.75, 5.15,
5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15,
5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15, 5.15,
5.15, 5.15, 5.15, 105.15, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9,
```

```
2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9,
2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 2.9, 102.9, 2.7, 2.7, 2.7,
2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7,
2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7, 2.7,
2.7, 102.7), DATE = structure(c(17742, 18016, 17742, 18107, 17835,
18200, 17865, 18230, 17927, 18292, 18106, 18382, 17742, 18107,
18473, 17835, 18200, 18566, 17927, 18292, 18658, 18016, 18382,
18747, 17742, 18107, 18473, 18838, 17865, 18230, 18596, 18961,
17927, 18292, 18658, 19023, 18016, 18382, 18747, 19112, 17835,
18200, 18566, 18931, 19296, 17742, 18107, 18473, 18838, 19203,
19568, 17927, 18292, 18658, 19023, 19388, 19753, 17742, 18107,
18473, 18838, 19203, 19568, 19934, 20299, 17835, 18200, 18566,
18931, 19296, 19661, 20027, 20392, 18016, 18382, 18747, 19112,
19477, 19843, 20208, 20573, 17742, 18107, 18473, 18838, 19203,
19568, 19934, 20299, 20664, 17835, 18200, 18566, 18931, 19296,
19661, 20027, 20392, 20757, 17835, 18200, 18566, 18931, 19296,
19661, 20027, 20392, 20757, 21122, 18016, 18382, 18747, 19112,
19477, 19843, 20208, 20573, 20938, 21304, 17742, 18107, 18473,
18838, 19203, 19568, 19934, 20299, 20664, 21029, 21395, 21760,
22125, 22490, 22856, 17835, 18200, 18566, 18931, 19296, 19661,
20027, 20392, 20757, 21122, 21488, 21853, 22218, 22583, 22949,
23314, 23679, 24044, 24410, 24775, 25140, 25505, 25871, 26236,
26601, 26966, 27332, 17835, 18200, 18566, 18931, 19296, 19661,
20027, 20392, 20757, 21122, 21488, 21853, 22218, 22583, 22949,
23314, 23679, 24044, 24410, 24775, 25140, 25505, 25871, 26236,
26601, 26966, 27332, 27697, 28062, 17835, 18200, 18566, 18931,
19296, 19661, 20027, 20392, 20757, 21122, 21488, 21853, 22218,
22583, 22949, 23314, 23679, 24044, 24410, 24775, 25140, 25505,
25871, 26236, 26601, 26966, 27332, 27697, 28062, 28427, 28793
), class = "Date")), .Names = c("ISIN", "CF", "DATE")), TODAY =
structure(17693, class = "Date")), .Names = c("ISIN",
```

```
"MATURITYDATE", "ISSUEDATE", "COUPONRATE", "PRICE", "ACCRUED",

"CASHFLOWS", "TODAY")), POLAND = structure(list(ISIN =
c("XS0874841066", "XS0458008496",

"XS0210314299", "XS0543882095", "XS1306382364", "XS0282701514",

"XS0794399674", "XS1015428821", "XS0841073793", "XS0479333311",

"XS1288467605", "XS1346201616", "XS1766612672", "XS1209947271",

"XS1584894650", "XS1508566392", "XS1346201889", "XS1508566558"),
MATURITYDATE = structure(c(17911,

18184, 18367, 18709, 18914, 19010, 19376, 19737, 19913, 20108,

20340, 20472, 20672, 20948, 21113, 21482, 24123, 28056), class =
"Date"), ISSUEDATE = structure(c(15720,

14532, 12801, 14875, 16722, 13531, 15509, 16085, 15622, 14629,

16687, 16818, 17569, 16532, 17248, 17099, 16818, 17099), class =
"Date"), COUPONRATE = c(0.01625,

0.04675, 0.042,  0.04, 0.00875, 0.045, 0.0375, 0.03, 0.03375, 0.0525,

0.015, 0.015, 0.0125, 0.00875, 0.01375, 0.01, 0.02375, 0.02), PRICE =
c(101.05,

107.324, 107.887, 111.15, 102.55, 115.92, 115.3, 113.65, 115.73, 128,

103.5, 103.25, 100.37, 97.76, 101.18, 96.29, 106.19, 99.31), ACCRUED =
c(0.6544,

3.0612, 0.6559, 0.8767, 0.5753, 1.7753, 1.4692, 1.2082, 3.1161,
2.0425,

1.1301, 0.5877, 1.0548, 0.0767, 0.874, 0.6274, 0.937, 1.2548),
CASHFLOWS = structure(list(

ISIN = c("XS0874841066", "XS0458008496", "XS0458008496",
"XS0210314299",

"XS0210314299", "XS0543882095", "XS0543882095", "XS0543882095",

"XS1306382364", "XS1306382364", "XS1306382364", "XS1306382364",

"XS0282701514", "XS0282701514", "XS0282701514", "XS0282701514",

"XS0794399674", "XS0794399674", "XS0794399674", "XS0794399674",

"XS0794399674", "XS1015428821", "XS1015428821", "XS1015428821",

"XS1015428821", "XS1015428821", "XS1015428821", "XS0841073793",

"XS0841073793", "XS0841073793", "XS0841073793",  "XS0841073793",

"XS0841073793",  "XS0841073793", "XS0479333311", "XS0479333311",

"XS0479333311",  "XS0479333311", "XS0479333311", "XS0479333311",
```

```
"XS0479333311",  "XS1288467605", "XS1288467605", "XS1288467605",
"XS1288467605",  "XS1288467605", "XS1288467605", "XS1288467605",
"XS1288467605",  "XS1346201616", "XS1346201616", "XS1346201616",
"XS1346201616",  "XS1346201616", "XS1346201616", "XS1346201616",
"XS1346201616",  "XS1766612672", "XS1766612672", "XS1766612672",
"XS1766612672",  "XS1766612672", "XS1766612672", "XS1766612672",
"XS1766612672",  "XS1766612672", "XS1209947271", "XS1209947271",
"XS1209947271",  "XS1209947271", "XS1209947271", "XS1209947271",
"XS1209947271",  "XS1209947271", "XS1209947271", "XS1584894650",
"XS1584894650",  "XS1584894650", "XS1584894650", "XS1584894650",
"XS1584894650",  "XS1584894650", "XS1584894650", "XS1584894650",
"XS1584894650",  "XS1508566392", "XS1508566392", "XS1508566392",
"XS1508566392",  "XS1508566392", "XS1508566392", "XS1508566392",
"XS1508566392",  "XS1508566392", "XS1508566392", "XS1508566392",
"XS1346201889",  "XS1346201889", "XS1346201889", "XS1346201889",
"XS1346201889",  "XS1346201889", "XS1346201889", "XS1346201889",
"XS1346201889",  "XS1346201889", "XS1346201889", "XS1346201889",
"XS1346201889",  "XS1346201889", "XS1346201889", "XS1346201889",
"XS1346201889",  "XS1346201889", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558", "XS1508566558",
"XS1508566558",  "XS1508566558", "XS1508566558"), CF = c(101.625,
4.675, 104.675,  4.2, 104.2, 4, 4, 104, 0.875, 0.875, 0.875,
100.875, 4.5, 4.5, 4.5, 104.5, 3.75, 3.75, 3.75, 3.75, 103.75,
3, 3, 3, 3, 3, 103, 3.375, 3.375, 3.375, 3.375, 3.375, 3.375,
103.375, 5.25, 5.25, 5.25, 5.25, 5.25, 5.25, 105.25, 1.5, 1.5,
```

```
1.5, 1.5, 1.5, 1.5, 1.5, 101.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5,

1.5, 101.5, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 1.25, 101.25,

0.875, 0.875, 0.875, 0.875, 0.875, 0.875, 0.875, 0.875, 100.875,

1.375, 1.375, 1.375, 1.375, 1.375, 1.375, 1.375, 1.375, 1.375,

101.375, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 101, 2.375, 2.375, 2.375,

2.375, 2.375, 2.375, 2.375, 2.375, 2.375, 2.375, 2.375, 2.375, 2.375,

2.375, 2.375, 2.375, 2.375, 102.375, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 102),

DATE = structure(c(17911, 17819, 18184, 18001, 18367, 17978,

18344, 18709, 17818, 18183, 18549, 18914, 17914, 18279, 18645,

19010, 17915, 18280, 18646, 19011, 19376, 17911, 18276, 18642,

19007, 19372, 19737, 17721, 18086, 18452, 18817, 19182, 19547,

19913, 17916, 18281, 18647, 19012, 19377, 19742, 20108, 17783,

18148, 18514, 18879, 19244, 19609, 19975, 20340, 17915, 18280,

18646, 19011, 19376, 19741, 20107, 20472, 17750, 18115, 18481,

18846, 19211, 19576, 19942, 20307, 20672, 18026, 18392, 18757,

19122, 19487, 19853, 20218, 20583, 20948, 17826, 18191, 18557,

18922, 19287, 19652, 20018, 20383, 20748, 21113, 17829, 18194,

18560, 18925, 19290, 19655, 20021, 20386, 20751, 21116, 21482,

17914, 18279, 18645, 19010, 19375, 19740, 20106, 20471, 20836,

21201, 21567, 21932, 22297, 22662, 23028, 23393, 23758, 24123,

17829, 18194, 18560, 18925, 19290, 19655, 20021, 20386, 20751,

21116, 21482, 21847, 22212, 22577, 22943, 23308, 23673, 24038,

24404, 24769, 25134, 25499, 25865, 26230, 26595, 26960, 27326,

27691, 28056), class = "Date")), .Names = c("ISIN",

"CF", "DATE")), TODAY = structure(17693, class = "Date")), .Names =
c("ISIN",

"MATURITYDATE", "ISSUEDATE", "COUPONRATE", "PRICE", "ACCRUED",

"CASHFLOWS", "TODAY"))), .Names = c("GERMANY", "SPAIN", "POLAND"

), class = "couponbonds")
```

Script used for the several estimations regarding the original German sample (including outliers)

```
#Nelson and Siegel and Edited Nelson and Siegel estimation of the
#zero-coupon Yield Curve for German Bonds

NelsonSiegelGERM <- estim_nss(DataSet, "GERMANY", matrange =c(0,30),
method="ns")
NelsonSiegelGERM_edit <- estim_nss.couponbonds_edit(DataSet,
"GERMANY", matrange=c(0,30), method="ns")

#See Results
NelsonSiegelGERM
NelsonSiegelGERM_edit

#Good of Fitness
summary(NelsonSiegelGERM)
summary(NelsonSiegelGERM_edit)

#See the Plots
plot.termstrc_nss_edit(NelsonSiegelGERM)
plot.termstrc_nss_edit(NelsonSiegelGERM_edit)

#Svensson and Edited Svensson estimation
#of the zero-coupon Yield Curve for German Bonds
SvenssonGERM <- estim_nss(DataSet, "GERMANY", matrange=c(0,30),
method="sv")
SvenssonGERM_edit <- estim_nss.couponbonds_edit(DataSet, "GERMANY",
matrange=c(0,30), method="sv")

#See Results
SvenssonGERM
SvenssonGERM_edit

#Good of Fitness
summary(SvenssonGERM)
summary(SvenssonGERM_edit)

#See the Plots
plot.termstrc_nss_edit(SvenssonGERM)
plot.termstrc_nss_edit(SvenssonGERM_edit)

#Adjusted Svensson and Edited Adjusted Svensson estimation of the
zero-coupon Yield Curve for German Bonds
AdjSvenssonGERM <- estim_nss(DataSet, "GERMANY", matrange =c(0,30),
method="asv")
AdjSvenssonGERM_edit <- estim_nss.couponbonds_edit(DataSet, "GERMANY",
matrange=c(0,30), method="asv")

#See Results
AdjSvenssonGERM
AdjSvenssonGERM_edit

#Good of Fitness
summary(AdjSvenssonGERM)
summary(AdjSvenssonGERM_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonGERM)
```

```
plot.termstrc_nss_edit(AdjSvenssonGERM_edit)

#Diebold Li and Edited Diebold Li estimation of the
#zero-coupon Yield Curve for German Bonds
DieboldLiGERMDefault <- estim_nss(DataSet, "GERMANY",
matrange=c(0,30), method="dl")
DieboldLiGERM0.0291 <- estim_nss(DataSet, "GERMANY", matrange=c(0,30),
method="dl", lambda=0.0291*12)
DieboldLiGERM0.0285 <- estim_nss(DataSet, "GERMANY", matrange=c(0,30),
method="dl", lambda=0.0285*12)
DieboldLiGERMDefault_edit <- estim_nss.couponbonds_edit(DataSet,
"GERMANY", matrange=c(0,30), method="dl")
DieboldLiGERM0.0136_edit <- estim_nss.couponbonds_edit(DataSet,
"GERMANY", matrange=c(0,30), method="dl", lambda=0.0136*12)
DieboldLiGERM0.0285_edit <- estim_nss.couponbonds_edit(DataSet,
"GERMANY", matrange=c(0,30), method="dl", lambda=0.0285*12)

#See the Results
DieboldLiGERMDefault
DieboldLiGERM0.0291
DieboldLiGERM0.0285
DieboldLiGERMDefault_edit
DieboldLiGERM0.0136_edit
DieboldLiGERM0.0285_edit

#Good of Fitness
summary(DieboldLiGERMDefault)
summary(DieboldLiGERM0.0291)
summary(DieboldLiGERM0.0285)
summary(DieboldLiGERMDefault_edit)
summary(DieboldLiGERM0.0136_edit)
summary(DieboldLiGERM0.0285_edit)

#See the Plots
plot.termstrc_nss_edit(DieboldLiGERMDefault)
plot.termstrc_nss_edit(DieboldLiGERM0.0291)
plot.termstrc_nss_edit(DieboldLiGERM0.0285)
plot.termstrc_nss_edit(DieboldLiGERMDefault_edit)
plot.termstrc_nss_edit(DieboldLiGERM0.0291_edit)
plot.termstrc_nss_edit(DieboldLiGERM0.0285_edit)

#Cubic Splines estimation of the zero-coupon Yield Curve for German
Bonds
CubicSplinesGERM <- estim_cs(DataSet,"GERMANY", matrange=c(0,30))

#See the Results
CubicSplinesGERM

#Good of Fitness
summary(CubicSplinesGERM)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesGERM)


#See the Plots of the Yield Errors
plot(NelsonSiegelGERM, errors ="yield")
```

Script used for the several estimations regarding the original Spanish sample (including outliers)

```
#Nelson and Siegel and Edited Nelson and Siegel estimation
#of the zero-coupon Yield Curve for Spanish Bonds

NelsonSiegelSPAIN <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method= "ns")
NelsonSiegelSPAIN_edit <- estim_nss.couponbonds_edit(DataSet, "SPAIN",
matrange=c(0,30), method= "ns")

#View Results
NelsonSiegelSPAIN
NelsonSiegelSPAIN_edit

#Good of Fitness
summary(NelsonSiegelSPAIN)
summary(NelsonSiegelSPAIN_edit)

#See the plots
plot.termstrc_nss_edit(NelsonSiegelSPAIN)
plot.termstrc_nss_edit(NelsonSiegelSPAIN_edit)

#Svensson and Edited Svensson estimation of the
#zero-coupon Yield Curve for Spanish Bonds
SvenssonSPAIN <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method="sv")
SvenssonSPAIN_edit <- estim_nss.couponbonds_edit(DataSet, "SPAIN",
matrange=c(0,30), method="sv")

#View Results
SvenssonSPAIN
SvenssonSPAIN_edit

#Goodness of Fit
summary(SvenssonSPAIN)
summary(SvenssonSPAIN_edit)

#See the Plots
plot.termstrc_nss_edit(SvenssonSPAIN)
plot.termstrc_nss_edit(SvenssonSPAIN_edit)

#Adj. Svensson and Edited Adj. Svensson estimation of the
#zero-coupon Yield Curve for Spanish Bonds
AdjSvenssonSPAIN <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method="asv")
AdjSvenssonSPAIN_edit <- estim_nss.couponbonds_edit(DataSet, "SPAIN",
matrange=c(0,30), method="asv")

#View Results
AdjSvenssonSPAIN
AdjSvenssonSPAIN_edit

#Goodness of Fit
summary(AdjSvenssonSPAIN)
summary(AdjSvenssonSPAIN_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonSPAIN)
```

```
plot.termstrc_nss_edit(AdjSvenssonSPAIN_edit)

#Diebold and Li and Edited Diebold and Li estimation of
#the zero-coupon Yield Curve for Spanish Bonds
DieboldLiSPAINDefault <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method="dl")
DieboldLiSPAIN0.03775 <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method="dl", lambda= 0.03775*12)
DieboldLiSPAIN0.0285 <- estim_nss(DataSet, "SPAIN", matrange=c(0,30),
method="dl", lambda=0.0285*12)
DieboldLiSPAINDefault_edit <- estim_nss.couponbonds_edit(DataSet,
"SPAIN", matrange =c(0,30), method="dl")
DieboldLiSPAIN0.0362_edit <- estim_nss.couponbonds_edit(DataSet,
"SPAIN", matrange=c(0,30), method="dl", lambda= 0.0362*12)
DieboldLiSPAIN0.0285_edit <- estim_nss.couponbonds_edit(DataSet,
"SPAIN", matrange=c(0,30), method="dl", lambda=0.0285*12)

#View Results
DieboldLiSPAINDefault
DieboldLiSPAIN0.03775
DieboldLiSPAIN0.0285
DieboldLiSPAINDefault_edit
DieboldLiSPAIN0.0362_edit
DieboldLiSPAIN0.0285_edit

#Goodness of Fit
summary(DieboldLiSPAINDefault)
summary(DieboldLiSPAIN0.03775)
summary(DieboldLiSPAIN0.0285)
summary(DieboldLiSPAINDefault_edit)
summary(DieboldLiSPAIN0.0362_edit)
summary(DieboldLiSPAIN0.0285_edit)

#See the Plots
plot.termstrc_nss_edit(DieboldLiSPAINDefault)
plot.termstrc_nss_edit(DieboldLiSPAIN0.03775)
plot.termstrc_nss_edit(DieboldLiSPAIN0.0285)
plot.termstrc_nss_edit(DieboldLiSPAINDefault_edit)
plot.termstrc_nss_edit(DieboldLiSPAIN0.03775_edit)
plot.termstrc_nss_edit(DieboldLiSPAIN0.0285_edit)


#Cubic Splines estimation method of the zero-coupon
#Yield Curve for Spanish Bonds
CubicSplinesSPAIN <- estim_cs(DataSet, "SPAIN", matrange=c(0,30))

#View Results
CubicSplinesSPAIN

#Goodness of Fit
summary(CubicSplinesSPAIN)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesSPAIN)

#See the Plot of the Yield Errors
plot(NelsonSiegelSPAIN, errors="yield")
```

Script used for the several estimations regarding the original Polish sample (including outliers)

```
#Nelson and Siegel and Edited Nelson and Siegel estimation of the
#zero-coupon Yield Curve for Polish Bonds
NelsonSiegelPOLAND <- estim_nss(DataSet, "POLAND", matrange= c(0,30),
method="ns")
NelsonSiegelPOLAND_edit <- estim_nss.couponbonds_edit(DataSet,
"POLAND", matrange=c(0,30), method="ns")

#View Results
NelsonSiegelPOLAND
NelsonSiegelPOLAND_edit

#Goodness of Fit
summary(NelsonSiegelPOLAND)
summary(NelsonSiegelPOLAND_edit)

#See the Plots
plot.termstrc_nss_edit(NelsonSiegelPOLAND)
plot.termstrc_nss_edit(NelsonSiegelPOLAND_edit)

#Svensson and Edited Svensson estimation of the
#zero-coupon Yield Curve for Polish Bonds
SvenssonPOLAND <- estim_nss(DataSet,"POLAND", matrange =c(0,30),
method="sv")
SvenssonPOLAND_edit <- estim_nss.couponbonds_edit(DataSet, "POLAND",
matrange=c(0,30))

#See the Results
SvenssonPOLAND
SvenssonPOLAND_edit

#Goodness of Fit
summary(SvenssonPOLAND)
summary(SvenssonPOLAND_edit)

#See the Plots
plot.termstrc_nss_edit(SvenssonPOLAND)
plot.termstrc_nss_edit(SvenssonPOLAND_edit)

#Adj Svensson and Edited Adj. Svensson estimation of the
#zero-coupon Yield Curve for Polish Bonds
AdjSvenssonPOLAND <- estim_nss(DataSet, "POLAND", matrange=c(0,30),
method="asv")
AdjSvenssonPOLAND_edit <- estim_nss.couponbonds_edit(DataSet,
"POLAND", matrange=c(0,30), method="asv")

#See the Results
AdjSvenssonPOLAND
AdjSvenssonPOLAND_edit

#Goodness of Fit
summary(AdjSvenssonPOLAND)
summary(AdjSvenssonPOLAND_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonPOLAND)
plot.termstrc_nss_edit(AdjSvenssonPOLAND_edit)
```

```
#Diebold and Li and Edited Diebold and Li estimation of the zero-
coupon Yield Curve for Polish Bonds
DieboldLiPOLANDDefault <- estim_nss(DataSet, "POLAND",
matrange=c(0,30), method="dl")
DieboldLiPOLAND0.0349 <- estim_nss(DataSet, "POLAND",
matrange=c(0,30), method="dl", lambda= 0.0349*12)
DieboldLiPOLAND0.0285 <- estim_nss(DataSet, "POLAND",
matrange=c(0,30), method="dl", lambda=0.0285*12)
DieboldLiPOLANDDefault_edit <- estim_nss.couponbonds_edit(DataSet,
"POLAND", matrange =c(0,30), method="dl")
DieboldLiPOLAND0.0351_edit <- estim_nss.couponbonds_edit(DataSet,
"POLAND", matrange=c(0,30), method="dl", lambda= 0.0351*12)
DieboldLiPOLAND0.0285_edit <- estim_nss.couponbonds_edit(DataSet,
"POLAND", matrange=c(0,30), method="dl", lambda=0.0285*12)

#See Results
DieboldLiPOLANDDefault
DieboldLiPOLAND0.0349
DieboldLiPOLAND0.0285
DieboldLiPOLANDDefault_edit
DieboldLiPOLAND0.0351_edit
DieboldLiPOLAND0.0285_edit

#Goodness of Fit
summary(DieboldLiPOLANDDefault)
summary(DieboldLiPOLAND0.0349)
summary(DieboldLiPOLAND0.0285)
summary(DieboldLiPOLANDDefault_edit)
summary(DieboldLiPOLAND0.0351_edit)
summary(DieboldLiPOLAND0.0285_edit)

#See the Plots

plot.termstrc_nss_edit(DieboldLiPOLANDDefault)
plot.termstrc_nss_edit(DieboldLiPOLAND0.0349)
plot.termstrc_nss_edit(DieboldLiPOLAND0.0285)
plot.termstrc_nss_edit(DieboldLiPOLANDDefault_edit)
plot.termstrc_nss_edit(DieboldLiPOLAND0.0349_edit)
plot.termstrc_nss_edit(DieboldLiPOLAND0.0285_edit)

#Cubic Splines estimation method of the zero-coupon Yield Curve for
Polish Bonds
CubicSplinesPOLAND<- estim_cs(DataSet, "POLAND", matrange=c(0,30))

#View Results
CubicSplinesPOLAND

#Goodness of Fit
summary(CubicSplinesPOLAND)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesPOLAND)#Alterar Fun硯 Plot para as CS

#See the Plot of the Yield Errors
plot(NelsonSiegelPOLAND, errors="yield")
```

Script used for the several estimations regarding the edited German sample (excluding outliers)

```
#Remove the Outliers
DataSetGERMOutlier <- rm_bond(DataSet, "GERMANY",
c("DE0001135358","DE0001030559","DE0001030567","DE0001030542"))

#Nelson and Siegel and Edited Nelson and Siegel estimation of the
zero-coupon Yield Curve for German Bonds(after removing the outliers)
NelsonSiegelGERMOutlier <- estim_nss(DataSetGERMOutlier, "GERMANY",
matrange =c(0,30), method="ns")
NelsonSiegelGERMOutlier_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="ns")

#See Results
NelsonSiegelGERMOutlier
NelsonSiegelGERMOutlier_edit

#Goodness of Fit
summary(NelsonSiegelGERMOutlier)
summary(NelsonSiegelGERMOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(NelsonSiegelGERMOutlier)
plot.termstrc_nss_edit(NelsonSiegelGERMOutlier_edit)

#Svensson and Edited Svensson estimation of the zero-coupon
#Yield Curve for German Bonds (after removing the outliers)
SvenssonGERMOutlier <- estim_nss(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="sv")
SvenssonGERMOutlier_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="sv")

#See Results
SvenssonGERMOutlier
SvenssonGERMOutlier_edit

#Good of Fitness
summary(SvenssonGERMOutlier)
summary(SvenssonGERMOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(SvenssonGERMOutlier)
plot.termstrc_nss_edit(SvenssonGERMOutlier_edit)

#Adjusted Svensson and Edited Adjusted Svensson estimation of the
#zero-coupon Yield Curve for German Bonds(after removing the outliers)
AdjSvenssonGERMOutlier <- estim_nss(DataSetGERMOutlier, "GERMANY",
matrange =c(0,30), method="asv")
AdjSvenssonGERMOutlier_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="asv")

#See Results
AdjSvenssonGERMOutlier
AdjSvenssonGERMOutlier_edit

#Good of Fitness
```

```
summary(AdjSvenssonGERMOutlier)
summary(AdjSvenssonGERMOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonGERMOutlier)
plot.termstrc_nss_edit(AdjSvenssonGERMOutlier_edit)

#Diebold Li and Edited Diebold Li estimation of the zero-coupon
#Yield Curve for German Bonds (after removing the outliers)
DieboldLiGERMOutlierDefault <- estim_nss(DataSetGERMOutlier,
"GERMANY", matrange=c(0,30), method="dl")
DieboldLiGERMOutlier0.0501 <- estim_nss(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="dl", lambda=0.0501*12)
DieboldLiGERMOutlier0.0285 <- estim_nss(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="dl", lambda=0.0285*12)
DieboldLiGERMOutlierDefault_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="dl")
DieboldLiGERMOutlier0.0417_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="dl", lambda=0.0417*12)
DieboldLiGERMOutlier0.0285_edit <-
estim_nss.couponbonds_edit(DataSetGERMOutlier, "GERMANY",
matrange=c(0,30), method="dl", lambda=0.0285*12)

#See results
DieboldLiGERMOutlierDefault
DieboldLiGERMOutlier0.0501
DieboldLiGERMOutlier0.0285
DieboldLiGERMOutlierDefault_edit
DieboldLiGERMOutlier0.0417_edit
DieboldLiGERMOutlier0.0285_edit

#Good of Fitness
summary(DieboldLiGERMOutlierDefault)
summary(DieboldLiGERMOutlier0.0501)
summary(DieboldLiGERMOutlier0.0285)
summary(DieboldLiGERMOutlierDefault_edit)
summary(DieboldLiGERMOutlier0.0417_edit)
summary(DieboldLiGERMOutlier0.0285_edit)

#See the Plots
plot.termstrc_nss_edit(DieboldLiGERMOutlierDefault)
plot.termstrc_nss_edit(DieboldLiGERMOutlier0.0501)
plot.termstrc_nss_edit(DieboldLiGERMOutlier0.0285)
plot.termstrc_nss_edit(DieboldLiGERMOutlierDefault_edit)
plot.termstrc_nss_edit(DieboldLiGERMOutlier0.0417_edit)
plot.termstrc_nss_edit(DieboldLiGERMOutlier0.0285_edit)

#Cubic Splines estimation of the zero-coupon Yield Curve
CubicSplinesGERMOutlier <- estim_cs(DataSetGERMOutlier,"GERMANY",
matrange=c(0,30))

#See the Results
CubicSplinesGERMOutlier

#Good of Fitness
summary(CubicSplinesGERMOutlier)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesGERMOutlier)
```

Script used for the several estimations regarding the edited Spanish sample (excluding outliers)

```
#Remove the Outliers
DataSetSPAINOutlier <- rm_bond(DataSet, "SPAIN",
c("ES00000126W8","ES00000127H7","ES00000128D4"))

#Nelson and Siegel and Edited Nelson and Siegel estimation of the
zero-coupon Yield Curve for Spanish Bonds (after removing outliers)
NelsonSiegelSPAINOutlier <- estim_nss(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method= "ns")
NelsonSiegelSPAINOutlier_edit <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method= "ns")

#See the Results
NelsonSiegelSPAINOutlier
NelsonSiegelSPAINOutlier_edit

#Good of Fitness
summary(NelsonSiegelSPAINOutlier)
summary(NelsonSiegelSPAINOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(NelsonSiegelSPAINOutlier)
plot.termstrc_nss_edit(NelsonSiegelSPAINOutlier_edit)

#Svensson and Edited Svensson estimation of the zero-coupon
#Yield Curve for Spanish Bonds (after removing the outliers)
SvenssonSPAINOutlier <- estim_nss(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="sv")
SvenssonSPAIN_editOutlier <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="sv")

#View Results
SvenssonSPAINOutlier
SvenssonSPAIN_editOutlier

#Goodness of Fit
summary(SvenssonSPAINOutlier)
summary(SvenssonSPAIN_editOutlier)

#See the Plots
plot.termstrc_nss_edit(SvenssonSPAINOutlier)
plot.termstrc_nss_edit(SvenssonSPAIN_editOutlier)

#Adj. Svensson and Edited Adj. Svensson estimation of the zero-coupon
#Yield Curve for Spanish Bonds (after removing the outliers)
AdjSvenssonSPAINOutlier <- estim_nss(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="asv")
AdjSvenssonSPAINOutlier_edit <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="asv")

#View Results
AdjSvenssonSPAINOutlier
AdjSvenssonSPAINOutlier_edit

#Goodness of Fit
```

```
summary(AdjSvenssonSPAINOutlier)
summary(AdjSvenssonSPAINOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonSPAINOutlier)
plot.termstrc_nss_edit(AdjSvenssonSPAINOutlier_edit)

#Diebold and Li and Edited Diebold and Li estimation of the zero-
coupon Yield Curve for Spanish Bonds (after removing the outliers)
DieboldLiSPAINOutlierDefault <- estim_nss(DataSetSPAINOutlier,
"SPAIN", matrange=c(0,30), method="dl")
DieboldLiSPAINOutlier0.0329 <- estim_nss(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="dl", lambda= 0.0329*12)
DieboldLiSPAINOutlier0.0285 <- estim_nss(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="dl", lambda=0.0285*12)
DieboldLiSPAINOutlierDefault_edit <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN", matrange
=c(0,30), method="dl")
DieboldLiSPAINOutlier0.0298_edit <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="dl", lambda= 0.0298*12)
DieboldLiSPAINOutlier0.0285_edit <-
estim_nss.couponbonds_edit(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30), method="dl", lambda=0.0285*12)

#View Results
DieboldLiSPAINOutlierDefault
DieboldLiSPAINOutlier0.0329
DieboldLiSPAINOutlier0.0285
DieboldLiSPAINOutlierDefault_edit
DieboldLiSPAINOutlier0.0298_edit
DieboldLiSPAINOutlier0.0285_edit

#Goodness of Fit
summary(DieboldLiSPAINOutlierDefault)
summary(DieboldLiSPAINOutlier0.0329)
summary(DieboldLiSPAINOutlier0.0285)
summary(DieboldLiSPAINOutlierDefault_edit)
summary(DieboldLiSPAINOutlier0.0298_edit)
summary(DieboldLiSPAINOutlier0.0285_edit)

#See the Plots
plot.termstrc_nss_edit(DieboldLiSPAINOutlierDefault)
plot.termstrc_nss_edit(DieboldLiSPAINOutlier0.0329)
plot.termstrc_nss_edit(DieboldLiSPAINOutlier0.0285)
plot.termstrc_nss_edit(DieboldLiSPAINOutlierDefault_edit)
plot.termstrc_nss_edit(DieboldLiSPAINOutlier0.0298_edit)
plot.termstrc_nss_edit(DieboldLiSPAINOutlier0.0285_edit)

#Cubic Splines estimation method of the zero-coupon Yield Curve
CubicSplinesSPAINOutliers <- estim_cs(DataSetSPAINOutlier, "SPAIN",
matrange=c(0,30))

#View Results
CubicSplinesSPAINOutliers

#Goodness of Fit
summary(CubicSplinesSPAINOutliers)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesSPAINOutliers)
```

Script used for the several estimations regarding the edited Polishh sample (excluding outliers)

```
#Remove the Outliers
DataSetPOLANDOutlier <- rm_bond(DataSet, "POLAND", c("XS0458008496"))

#Nelson and Siegel and Edited Nelson and Siegel estimation of the
zero-coupon Yield Curve for Polish Bonds(after removing the outliers)
NelsonSiegelPOLANDOutlier <- estim_nss(DataSetPOLANDOutlier, "POLAND",
matrange =c(0,30), method="ns")
NelsonSiegelPOLANDOutlier_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="ns")

#See Results
NelsonSiegelPOLANDOutlier
NelsonSiegelPOLANDOutlier_edit

#Goodness of Fit
summary(NelsonSiegelPOLANDOutlier)
summary(NelsonSiegelPOLANDOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(NelsonSiegelPOLANDOutlier)
plot.termstrc_nss_edit(NelsonSiegelPOLANDOutlier_edit)

#Svensson and Edited Svensson estimation of the zero-coupon
#Yield Curve for Polish Bonds (after removing the outliers)
SvenssonPOLANDOutlier <- estim_nss(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="sv")
SvenssonPOLANDOutlier_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="sv")

#See Results
SvenssonPOLANDOutlier
SvenssonPOLANDOutlier_edit

#Good of Fitness
summary(SvenssonPOLANDOutlier)
summary(SvenssonPOLANDOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(SvenssonPOLANDOutlier)
plot.termstrc_nss_edit(SvenssonPOLANDOutlier_edit)

#Adjusted Svensson and Edited Adjusted Svensson estimation of the
#zero-coupon Yield Curve for Polish Bonds(after removing the outliers)
AdjSvenssonPOLANDOutlier <- estim_nss(DataSetPOLANDOutlier, "POLAND",
matrange =c(0,30), method="asv")
AdjSvenssonPOLANDOutlier_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND ",
matrange=c(0,30), method="asv")

#See Results
AdjSvenssonPOLANDOutlier
AdjSvenssonPOLANDOutlier_edit

#Good of Fitness
summary(AdjSvenssonPOLANDOutlier)
```

```
summary(AdjSvenssonPOLANDOutlier_edit)

#See the Plots
plot.termstrc_nss_edit(AdjSvenssonPOLANDOutlier)
plot.termstrc_nss_edit(AdjSvenssonPOLANDOutlier_edit)

#Diebold Li and Edited Diebold Li estimation of the zero-coupon
#Yield Curve for Polish Bonds (after removing the outliers)
DieboldLiPOLANDOutlierDefault <- estim_nss(DataSetPOLANDOutlier,
"POLAND", matrange=c(0,30), method="dl")
DieboldLiPOLANDOutlier0.05 <- estim_nss(DataSetPOLANDOutlier,
"POLAND", matrange=c(0,30), method="dl", lambda=0.05*12)
DieboldLiPOLANDOutlier0.0285 <- estim_nss(DataSetPOLANDOutlier,
"POLAND", matrange=c(0,30), method="dl", lambda=0.0285*12)
DieboldLiPOLANDOutlierDefault_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="dl")
DieboldLiPOLANDOutlier0.05_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="dl", lambda=0.05*12)
DieboldLiPOLANDOutlier0.0285_edit <-
estim_nss.couponbonds_edit(DataSetPOLANDOutlier, "POLAND",
matrange=c(0,30), method="dl", lambda=0.0285*12)

#See results
DieboldLiPOLANDOutlierDefault
DieboldLiPOLANDOutlier0.05
DieboldLiPOLANDOutlier0.0285
DieboldLiPOLANDOutlierDefault_edit
DieboldLiPOLANDOutlier0.05_edit
DieboldLiPOLANDOutlier0.0285_edit

#Good of Fitness
summary(DieboldLiPOLANDOutlierDefault)
summary(DieboldLiPOLANDOutlier0.05)
summary(DieboldLiPOLANDOutlier0.0285)
summary(DieboldLiPOLANDOutlierDefault_edit)
summary(DieboldLiPOLANDOutlier0.05_edit)
summary(DieboldLiPOLANDOutlier0.0285_edit)

#See the Plots
plot.termstrc_nss_edit(DieboldLiPOLANDOutlierDefault)
plot.termstrc_nss_edit(DieboldLiPOLANDOutlier0.0501)
plot.termstrc_nss_edit(DieboldLiPOLANDOutlier0.0285)
plot.termstrc_nss_edit(DieboldLiPOLANDOutlierDefault_edit)
plot.termstrc_nss_edit(DieboldLiPOLANDOutlier0.0417_edit)
plot.termstrc_nss_edit(DieboldLiPOLANDOutlier0.0285_edit)

#Cubic Splines estimation of the zero-coupon Yield Curve
CubicSplinesPOLANDOutlier <- estim_cs(DataSetPOLANDOutlier,"POLAND",
matrange=c(0,30))

#See the Results
CubicSplinesPOLANDOutlier

#Good of Fitness
summary(CubicSplinesPOLANDOutlier)

#See the Plot
plot.termstrc_cs_edit(CubicSplinesPOLANDOutlier)
```

Scripts used to create the Edited Functions

```
get_constraints_edit <- function(method) {

  ## Diebold/Li

  if (method == "dl") {
    ui <- rbind(c(1,0,0))               # beta0 > 0

    ci <- c(0)
  }

  ## Nelson/Siegel

  if (method == "ns") {
    ui <- rbind(c(1,0,0,0),              # beta0 > 0
                c(0,0,0,1),              # tau1 > 0
                c(0,0,0,-1))             # tau1 < 30
    ci <- c(0,0,-30)
  }

  ## (Adjusted) Svensson

  if (method %in% c("sv","asv")) {
    ui <- rbind(c(1,0,0,0,0,0),         # beta0 > 0
                c(0,0,0,1,0,0),         # tau1 > 0
                c(0,0,0,-1,0,0),        # tau1 < 30
                c(0,0,0,0,0,1),         # tau2 > 0
                c(0,0,0,0,0,-1),        # tau2 < 30
                c(0,0,0,-1,0,1))        # tau2 - tau1 > 0
    ci <- c(0,0,-30,0,-30,0)
  }

  constraints <- list(ui = ui, ci = ci)
  constraints
}


######################################################################
####
### Nelson/Siegel-type yield curve estimation method for 'couponbonds'
###
######################################################################
####

estim_nss_edit <- function(dataset, ...) UseMethod("estim_nss")

estim_nss.couponbonds_edit <- function(dataset,                   #
dataset (static)
                                  group,                     # names
of countries for estimation c("Country 1", "Country 2", ...)
                                  matrange="all",            #
maturity range in years c(min, max)
                                  method="ns",
                                  startparam=NULL,           #
startparameter matrix with columns
c("beta0","beta1","beta2","tau1","beta3","tau2")
                                  # otherwise globally optimal
parameters are searched automatically
                                  lambda=0.0609*12,          # yearly
lambda-value for "Diebold/Li" estimation
```

```
                                    deltatau=0.1,              #
interval for parameter grid
                                    constrOptimOptions = list(control =
list(maxit = 2000), outer.iterations = 200, outer.eps = 1e-04),...
) {

  ## data preprocessing
  prepro <-
prepro_bond(group=group,bonddata=dataset,matrange=matrange)

  n_group=prepro$n_group
  sgroup=prepro$sgroup
  cf=prepro$cf
  cf_p=prepro$cf_p
  m=prepro$m
  m_p=prepro$m_p
  p=prepro$p
  ac=prepro$ac
  y=prepro$y
  duration=prepro$duration

  ## automatically determine globally optimal start parameters
  spsearch <- list()
  length(spsearch) <- n_group

  if(is.null(startparam)){
    startparam <- matrix(ncol = 6, nrow = n_group)

    colnames(startparam) <-
c("beta0","beta1","beta2","tau1","beta3","tau2")

    if (method == "dl") {startparam <- startparam[,1:3, drop=FALSE]}
    if (method == "ns") {startparam <- startparam[,1:4, drop=FALSE]}

    for (k in sgroup){
      print(paste("Searching startparameters for ", group[k]))
      spsearch[[k]] <- findstartparambonds(p[[k]],m[[k]],cf[[k]],
duration[[k]][,3],
                                             method, deltatau)
      startparam[k,] <- spsearch[[k]]$startparam
      print(startparam[k,])
    }
  }

  rownames(startparam) <- group

  ## objective function (weighted price error minimization)
  obj_fct <- function(b) {
    loss_function(p[[k]],

bond_prices(method,b,m[[k]],cf[[k]],lambda)$bond_prices,duration[[k]][
,3])}

  ## calculate optimal parameter vectors
  constraints <- get_constraints_edit(method)
  opt_result <- list()

  for (k in sgroup){
    opt_result[[k]] <- estimatezcyieldcurve(method, startparam[k,],
obj_fct, constraints, constrOptimOptions)
  }
```

```
  ## data post processing
  postpro <-
postpro_bond(opt_result,m,cf,sgroup,n_group,y,p,ac,m_p,method,lambda)

  ## return list of results
  result <- list(group=group,                     # e.g. countries,
rating classes
                 matrange=matrange,               # maturity range of
bonds
                 method=method,                   # estimation method
                 startparam=startparam,           # calculated
startparameters
                 n_group=n_group,                 # number of groups,
                 lambda=lambda,                   # lambda parameter for
dl
                 spsearch = spsearch,             # detailed data from
start param search
                 spot=postpro$zcy_curves,         # zero coupon yield
curves
                 spread=postpro$s_curves,         # spread curves
                 forward=postpro$fwr_curves,      # forward rate curves
                 discount=postpro$df_curves,      # discount factor
curves
                 expoints=postpro$expoints,       # extrapolation points
                 cf=cf,                           # cashflow matrix
                 m=m,                             # maturity matrix
                 duration=duration,               # duration, modified
duration, weights
                 p=p,                             # dirty prices
                 phat=postpro$phat,               # estimated dirty
prices
                 perrors=postpro$perrors,         # price errors
                 ac=ac,                           # accrued interest
                 y=y,                             # maturities and
yields
                 yhat=postpro$yhat,               # estimated yields
                 yerrors=postpro$yerrors,         # yield errors
                 opt_result=opt_result            # optimisation results
  )

  for ( i in 6:length(result)) names(result[[i]]) <- group
  class(result) <- "termstrc_nss"
  result
}

### Estimate zero-coupon yield curve

estimatezcyieldcurve <- function(method, startparam, obj_fct,
constraints, constrOptimOptions) {

  opt_result <- constrOptim(theta = startparam,
                            f = obj_fct,
                            grad = NULL,
                            ui = constraints$ui,
                            ci = constraints$ci,
                            mu = 1e-04,
                            control = constrOptimOptions$control,
                            method = "Nelder-Mead",
                            outer.iterations =
constrOptimOptions$outer.iterations,
```

```
                                          outer.eps = constrOptimOptions$outer.eps)

  opt_result
}

### Start parameter search routine for bond data

findstartparambonds <- function(p,m,cf, weights, method, deltatau =
0.1,
                                              control = list(), outer.iterations =
30, outer.eps = 1e-04) {

  if(method=="dl"){
    startparam = rep(0.01,3)
    tau = NULL
    fmin = NULL
    optind = NULL
  }

  if(method=="ns"){
    tau <- seq(deltatau, max(m), deltatau)
    fmin <- rep(NA, length(tau))
    lsbeta <- matrix(nrow = length(tau), ncol = 4)

    objfct <- function(b) {
loss_function(p,bond_prices("dl",b,m,cf,1/tau[i])$bond_prices,weights)
    }

    ui <- rbind(c(1,0,0))                  # beta0 > 0
                                         # beta0 + beta1 > 0
    ci <- c(0)

    for (i in 1:length(tau)){

      lsparam <- constrOptim(theta = rep(0.01,3), # start parameters
for D/L, objective function is convex
                             f = objfct,
                             grad = NULL,
                             ui = ui,
                             ci = ci,
                             mu = 1e-04,
                             control = control,
                             method = "Nelder-Mead",
                             outer.iterations = outer.iterations,
                             outer.eps = outer.eps)
      beta <- c(lsparam$par,tau[i])
      fmin[i] <- lsparam$value
      lsbeta[i,] <- beta
    }
    optind <- which(fmin == min(fmin, na.rm = TRUE))
    startparam <- lsbeta[optind,]
  }

  if(method %in% c("sv","asv")) {

    objfct <- function(b) {
      bsv <- c(b[1:3],tau1[i],b[4],tau2[j])
loss_function(p,bond_prices(method,bsv,m,cf)$bond_prices,weights)
    }
```

```
    ui <- rbind(c(1,0,0,0))                # beta0 > 0

    ci <- c(0)

    tau1 <- seq(deltatau, max(m),deltatau)
    tau2 <- seq(deltatau, max(m),deltatau)
    tau <- cbind(tau1, tau2)
    fmin <- matrix(nrow = length(tau1), ncol = length(tau2))
    lsbeta <- matrix(nrow = length(tau1)*length(tau2), ncol = 6)
    for (i in 1:length(tau1))
    {
      for (j in 1:length(tau2))
      {
        if(tau1[i] < tau2[j]) {
          lsparam <- constrOptim(theta = rep(0.01,4),
                                 f = objfct,
                                 grad = NULL,
                                 ui = ui,
                                 ci = ci,
                                 mu = 1e-04,
                                 control = control,
                                 method = "Nelder-Mead",
                                 outer.iterations = outer.iterations,
                                 outer.eps = outer.eps)

          beta <- c(lsparam$par[1:3],tau1[i],lsparam$par[4],tau2[j])
          fmin[i,j] <- lsparam$value
          lsbeta[(i-1)*length(tau1)+j,] <- beta
        }
      }
    }

    optind <- which(fmin == min(fmin, na.rm = TRUE),arr.ind=TRUE)
    startparam <- lsbeta[(optind[1]-1)*length(tau1) + optind[2],]
  }
  result <- list(startparam = startparam, tau = tau, fmin = fmin,
optind = optind)
  class(result) <- "spsearch"
  result
}

### Startparameter grid search plots

plot.spsearch <- function(x, rgl = TRUE, ...) {

  if(is.matrix(x$tau)){
    contour(x$tau[,1],x$tau[,2],log(x$fmin),nlevels=10,xlab = "tau_1",
ylab = "tau_2",main = "Log(Objective function)")
    points(x$tau[x$optind[1],1],x$tau[x$optind[2],2],pch = 10, col =
"red")
    if (rgl) {
      open3d()
      persp3d(x$tau[,1], x$tau[,2], log(x$fmin), col = "green3", box =
FALSE,xlab = "tau_1", ylab = "tau_2", zlab = "Log(Objective
function)")

points3d(x$tau[x$optind[1],1],x$tau[x$optind[2],2],min(log(x$fmin),
na.rm = TRUE), col = "red")
    }
    else {
```

```
      par(ask = TRUE)
      persp(x$tau[,1], x$tau[,2], log(x$fmin), col = "green3", box =
TRUE, xlab = "tau_1", ylab = "tau_2", zlab = "Log(Objective
function)", shade = TRUE, ticktype = "detailed", border = NA, cex.lab
= 1, cex.axis = 0.7,  theta = 0, phi = 25, r = sqrt(3), d = 1, scale =
TRUE, expand = 1, ltheta = 135, lphi = 0)
    }
  } else {
    plot(x$tau,x$fmin,xlab = "tau_1", ylab = "Objective function",
type = "l")
    points(x$tau[x$optind],x$fmin[x$optind],pch = 10, col = "red")
  }
}
```

Scripts used to create the Edited Plot functions

```
plot.termstrc_nss_edit <-
  function(x,matrange=c(min(mapply(function(i)
min(x$y[[i]][,1]),seq(x$n_group))),
                        max(mapply(function(i)
max(x$y[[i]][,1]),seq(x$n_group))))
           ,multiple=FALSE, expoints=unlist(x$expoints), ctype="spot",
           errors="none",
           lwd=2,lty=1,type="l",inset=c(0.8,0.1),ask=TRUE,
           ...) {

    # min and max maturity of all bonds in the sample
    samplemat <- c(min(mapply(function(i) min(x$y[[i]][,1]),
seq(x$n_group))),
                   max(mapply(function(i) max(x$y[[i]][,1]),
seq(x$n_group)))


    cdata <- switch(ctype, "spot" = x$spot,
                    "forward" = x$forward,
                    "discount" = x$discount
    )

    cname <- switch(ctype, "spot" = "Zero-coupon yield curve",
                    "forward" = "Forward rate curve",
                    "discount" = "Discount factor curve" )


    # plot all interst rate curves together
    if (multiple) {

      plot(x=cdata,multiple=multiple,
expoints=expoints,lwd=lwd,type=type,...) }

    if (!multiple && ctype %in% c("spot", "forward", "discount")){
      old.par <- par(no.readonly = TRUE)

      if(x$n_group != 1) par(ask=ask)

      # plot each interest rate curve seperately
      for (k in seq(x$n_group)  )
      {

        plot.ir_curve(cdata[[k]], ylim=c(-0.02, max(cdata[[k]][,2]) +
0.01 )*100,
```

```
xlim=c(max(floor(min(x$y[[k]][,1])),matrange[1]),

min(ceiling(max(x$y[[k]][,1])),matrange[2])), lwd=lwd,type=type, ...
        )

        title(names(x$opt_result)[k])

        if(ctype=="spot")
{points(x$y[[k]][,1],x$y[[k]][,2]*100,col="red")
          legend("bottom",legend=c("Zero-coupon yield curve","Yield-
to-maturity"),
                 col=c("steelblue","red"), lty = c(1, -1), pch=c(-
1,21))}
        else     legend("bottom",legend=cname    ,col=c("steelblue"),
lty = lty , pch=(-1))


      }
      on.exit(par(old.par))
    }

    # plot spread curves
    if(ctype == "spread") {plot(x$spread,expoints=expoints,
                                xlim=
c(max(floor(samplemat[1]),matrange[1]),

min(ceiling(samplemat[2]),matrange[2])),lwd=lwd,
                                ...)
    }
    # plot errors
    if(errors %in% c("price", "yield")){

      edata <- switch(errors,"price" = x$perrors, "yield"= x$yerrors )
      if(x$n_group == 1) ask= FALSE
      for(k in seq(x$n_group)){

        plot.error(edata[[k]],ask=ask,main=x$group[k],
                   ylab=paste("Error ",paste(errors,"s)",sep=""),sep="
("),...)

        legend("bottomright", legend=c(paste("  RMSE",
                                             switch(errors,"price" =
round(rmse(x$p[[k]],x$phat[[k]]),4),
                                                    "yield" =
round(rmse(x$y[[k]][,2],x$yhat[[k]][,2]),4)) ,sep=": "),

paste("AABSE",switch(errors,"price" =
round(aabse(x$p[[k]],x$phat[[k]]),4),
                                                            "yield" =
round(aabse(x$y[[k]][,2],x$yhat[[k]][,2]),4)),sep=": ")),bty="n",
inset=inset)

      }

    }



  }
```

```
plot.termstrc_cs_edit <-
  function(x,matrange =c(min(mapply(function(i) min(x$y[[i]][,1]),
seq(x$n_group))),
                        max(mapply(function(i) max(x$y[[i]][,1]),
seq(x$n_group)))),
           multiple=FALSE, ctype="spot",

lwd=2,lty=1,type="l",errors="none",inset=c(0.8,0.1),ask=TRUE, ...) {

    # min and max maturity of all bonds in the sample
    samplemat <- c(min(mapply(function(i) min(x$y[[i]][,1]),
seq(x$n_group))),
                    max(mapply(function(i) max(x$y[[i]][,1]),
seq(x$n_group))))


    cdata <- switch(ctype, "spot" = x$spot,
                    "forward" = x$forward,
                    "discount" = x$discount
    )

    cname <- switch(ctype, "spot" = "Zero-coupon yield curve",
                    "forward" = "Forward rate curve",
                    "discount" = "Discount factor curve" )


    # plot all interst rate curves together
    if (multiple) {

      plot(x=cdata,multiple=multiple,
expoints=NULL,lwd=lwd,type=type,...) }

    if (!multiple && ctype %in% c("spot", "forward", "discount")){
      old.par <- par(no.readonly = TRUE)
      if(x$n_group !=1) par(ask=ask)

      # plot each interest rate curve seperately
      for (k in seq(x$n_group)  )
      {

        plot.ir_curve(cdata[[k]], ylim=c(-0.02, max(cdata[[k]][,2]) +
0.01 )*100,

xlim=c(max(floor(min(x$y[[k]][,1])),matrange[1]),

min(ceiling(max(x$y[[k]][,1])),matrange[2])), lwd=lwd,type=type,...)



        title(x$group[k])

        if(ctype=="spot")
{points(x$y[[k]][,1],x$y[[k]][,2]*100,col="red")
          # lower ci
          lines(cdata[[k]][,1],cdata[[k]][,3]*100, type="l", lty=3,
col="steelblue" )
          # upper ci
          lines(cdata[[k]][,1],cdata[[k]][,4]*100, type="l", lty=3,
col="steelblue")
          # knot points
          abline(v=c(x$knotpoints[[k]]),lty=2, col="darkgrey")
```

```
          legend("bottom",legend=c("Zero-coupon yield curve",
                                   if(x$rse) "95 % Confidence interval
(robust s.e.)" else "95 % Confidence interval" ,"Yield-to-maturity",
"Knot points"),
                 col=c("steelblue","steelblue","red", "darkgrey"),
                 lty = c(1,3,-1,2), pch=c(-1,-1,21,-1))

        } else  legend("bottom",legend=cname,col=c("steelblue"), lty =
lty , pch=(-1))


      }
      on.exit(par(old.par))
    }

    # plot spread curves
    if(ctype == "spread") {plot(x$spread,expoints=NULL,
                            xlim=
c(max(floor(samplemat[1]),matrange[1]),

min(ceiling(samplemat[2]),matrange[2],max(mapply(function(i)

max(x$spread[[i]][,1]),seq(x$spread))))),lwd=lwd ,...)
    }


    # plot errors
    if(errors %in% c("price", "yield")){

      edata <- switch(errors,"price" = x$perrors, "yield"= x$yerrors )

      if(x$n_group == 1) ask= FALSE

      for(k in seq(x$n_group)){
        plot.error(edata[[k]],ask=ask
                   ,main=x$group[k],ylab=paste("Error
",paste(errors,"s)",sep=""),sep=" ("),...)

        legend("bottomright", legend=c(paste("  RMSE",
                                             switch(errors,"price" =
round(rmse(x$p[[k]],x$phat[[k]]),4),
                                                    "yield" =
round(rmse(x$y[[k]][,2],x$yhat[[k]][,2]),4)) ,sep=": "),

paste("AABSE",switch(errors,"price" =
round(aabse(x$p[[k]],x$phat[[k]]),4),
                                                          "yield" =
round(aabse(x$y[[k]][,2],x$yhat[[k]][,2]),4)),sep=": ")),bty="n",
inset=inset)

      }

    }

  }
```