COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

Modelling the term structure of Slovak government bond yields before and after the euro adoption

MASTER THESIS

2017 **Bc. Peter ČÁRSKY**

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

Modelling the term structure of Slovak government bond yields before and after the euro adoption

MASTER THESIS

Study programme:	Mathematical Economics, Finance and Modelling
Branch of study:	9.1.9 Applied mathematics
Departement:	Department of Applied Mathematics and Statistics
Supervisor:	Pavol Povala, PhD.

Bratislava 2017 **Bc. Peter ČÁRSKY**

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

24171166

THESIS ASSIGNMENT

Name and Surname:	Bc. Peter Čársky
Study programme:	Mathematical Economics, Finance and Modelling (Single degree study, master II. deg., full time form)
Field of Study:	Applied Mathematics
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Modelling the term structure of Slovak government bond yields before and after the euro adoption

Aim: Evaluation of statistical models of the yield curve in presence of structural changes like redomination of bonds after the euro adoption.

Supervisor:	Pavol Povala, PhD.
Department:	FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department:	prof. RNDr. Daniel Ševčovič, DrSc.
Assigned:	21.01.2016
Approved:	25.01.2016

prof. RNDr. Daniel Ševčovič, DrSc.
Guarantor of Study Programme

Student

Supervisor

24171166

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Peter Čársky
Študijný program: ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Modelling the term structure of Slovak government bond yields before and after the euro adoption
Modelovanie Slovenskej výnosovej krivky pred a po zavedení Eura

Cieľ: Evaluácia štatistických modelov výnosovej krivky v prítomnosti štrukturálnych zmien akou je napríklad redenominácia dlhopisov po vstupe do eurozóny.

Vedúci: Pavol Povala, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, DrSc.

Dátum zadania: 21.01.2016

Dátum schválenia: 25.01.2016

prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

študent

vedúci práce

Acknowledgement

I would like to express my grattitude to my supervisor Pavol Povala, PhD. for his guidance and support throughout the elaboration of this thesis.

Abstrakt

ČÁRSKY, Peter: *Modelovanie Slovenskej výnosovej krivky pred a po zavedení Eura.* [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: Pavol Povala, PhD., Bratislava, 2017, 91 s.

Slovenská výnosová krivka je podstatný ekonomický ukazovateľ, špeciálne pre investorov. Predikovanie pohybu tejto krivky s dostatočnou presnosťou je preto jedna zo základných úloh. V našej práci predstavíme viacero predikčných modelov založených na teórii časových radov. Podstatnou súčasťou našej práce je obsiahnutie informácie o štrukturálnej zmene výnosovej krivky, ktorá sa udiala počas prijatia Eura. Navrhneme dve možné riešenia tohto problému. Nakoniec vyhodnotíme kvalitu predikcií produkovaných každým modelom spolu s popisom ich silných a slabých stránok.

Kľúčové slová: Výnosová krivka • Časové rady • autoregresia • PCA • Diebold-Li.

Abstract

ČÁRSKY, Peter: *Modelling the term structure of Slovak government bond yields before and after the euro adoption.* [Master thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Pavol Povala, PhD., Bratislava: FMFI UK, 2017, 91 p.

The yield curve is an important economic indicator especially for investors. To predict the movement of yield curve to sufficient extent is then key task. In our thesis we introduce several predictive models based on theory of time series. Essential task in our thesis is to find a way to use the information of structural change of yield curve caused by euro adoption. We propose two different solutions to this problem in our thesis. Finally we evaluate quality of forecasts provided by each technique as well as their overall strengths and weaknesses.

Keywords: Yield curve • Time series • Autoregression • PCA • Diebold-Li.

Contents

Introduction

The yield curve is a graphed line that expresses yields as a function of time to maturity. Fixed income investors monitor the yield curve closely because significant interest rate changes, affecting financing costs and therefore expenditure decisions of businesses across all market sectors, are a major driving force of the market and the economy. Modelling the yield curve is therefore an important task in every economy. Two popular approaches to term structure modelling are no-arbitrage models and equilibrium models. The no-arbitrage tradition focuses on perfectly fitting the term structure at a point in time to ensure that no arbitrage possibilities exist, which is important for pricing derivatives. The equilibrium tradition focuses on modelling the dynamics of the instantaneous rate, typically using affine models, after which yields at other maturities can be derived under various assumptions about the risk premium. As Diebold and Li state in [4] many theoretical improvements have been made in this area. However the main goal of this thesis is to model the yield curve in time. This is the reason why in our thesis, we introduce several predictive methods based on theory of time series and evaluate them according to their forecasting power. The basic idea is to take time series of yields with different maturities and either use autoregressive techniques for modelling them or at first use various techniques to reduce the dimensionality of the problem and then model this lower number of time series.

We will work with data consisting of zero coupon yields on Slovak government bonds with maturities of 1 to 15 years. The very first thing we evaluate is presence of a structural break caused by euro adoption in January 2009. We gather insights on this change by comparing descriptive statistics for these two time periods in chapter 1. Further insight into this problem is provided in chapter 2 where we look also on the stationarity of univariate time series, formally testing presence of unit root. Decrease of dimensionality is done in chapter 3, where we use principal components analysis to find a 3-dimensional space that covers most of the variation present in the original 15-dimensional space. In chapter 4 we use Nelson-Siegel's strictly defined functional form to model the yield curve in time by performing an autoregressive analysis on factors from this model. In chapter 5 we use a theoretical framework that allows us to explain the structural change by exogenous time series. In chapter 6 a vector autoregression model is defined and two such models are specified and fitted on the data. Final evaluation of predictive power takes place in chapter 7 where we use several evaluation techniques, one of them developed by us specially for this problem.

CHAPTER 1

DESCRIPTIVE STATISTICS

In this chapter, we take a look at properties of time series of yields from Slovak Government Zero Coupon Bonds. This data can be found on *http://www.finance.gov.sk/Default.aspx?CatID=10501* and they cover period from 7.1.2003 to 31.10.2016 for bonds with maturities up to 10 years. We have data for bonds with maturities of 11 to 15 years starting from 11.5.2006 up to 31.10.2016.

1.1 Whole time period

In the figure 1.1 we can observe time series of yields for maturities from 1 to 10 years. Same time series but for maturities from 11 to 15 years can be seen in figure 1.2. We can see that time series of yields from bonds with similar maturities tend to follow approximately same pattern. This can be also seen in table 1.1 of we can see sample correlations between yields from bonds with different maturities. For similar maturities close to diagonal of the variance matrix 1.1, sample correlation is close to 1. Another observation is that especially for short maturities the time series changes drastically during years 2008 and 2009. The most probable cause for this structural break is adoption of euro in Slovakia which became national currency on 1.1.2009.

Table 1.1: Sample correlations between yields with different maturities rounded to two decimal numbers

Maturity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00	0.98	0.94	0.90	0.86	0.83	0.79	0.77	0.74	0.71	0.69	0.67	0.66	0.65	0.64
2	0.98	1.00	0.99	0.97	0.94	0.92	0.89	0.87	0.85	0.83	0.81	0.79	0.78	0.77	0.76
3	0.94	0.99	1.00	0.99	0.98	0.96	0.95	0.93	0.91	0.89	0.88	0.86	0.85	0.84	0.83
4	0.90	0.97	0.99	1.00	1.00	0.99	0.98	0.96	0.95	0.93	0.92	0.91	0.90	0.89	0.88
5	0.86	0.94	0.98	1.00	1.00	1.00	0.99	0.98	0.97	0.96	0.95	0.94	0.93	0.92	0.92
6	0.83	0.92	0.96	0.99	1.00	1.00	1.00	0.99	0.99	0.98	0.97	0.96	0.95	0.95	0.94
7	0.79	0.89	0.95	0.98	0.99	1.00	1.00	1.00	0.99	0.99	0.98	0.98	0.97	0.96	0.96
8	0.77	0.87	0.93	0.96	0.98	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.98	0.97
9	0.74	0.85	0.91	0.95	0.97	0.99	0.99	1.00	1.00	1.00	1.00	0.99	0.99	0.98	0.98
10	0.71	0.83	0.89	0.93	0.96	0.98	0.99	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99
11	0.69	0.81	0.88	0.92	0.95	0.97	0.98	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99
12	0.67	0.79	0.86	0.91	0.94	0.96	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
13	0.66	0.78	0.85	0.90	0.93	0.95	0.97	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00
14	0.65	0.77	0.84	0.89	0.92	0.95	0.96	0.98	0.98	0.99	1.00	1.00	1.00	1.00	1.00
15	0.64	0.76	0.83	0.88	0.92	0.94	0.96	0.97	0.98	0.99	0.99	1.00	1.00	1.00	1.00

In figure 1.3 we can see histograms of yields for all observed maturities. Here we can also see similarities among yields with maturities close to each other. In the case of longer

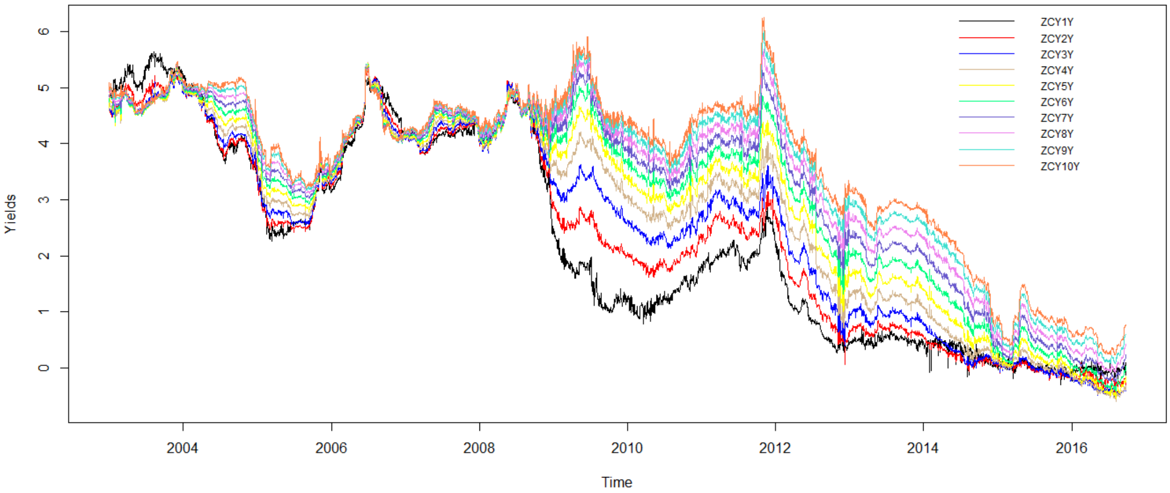

Figure 1.1: Behaviour of yields from Slovak government zero coupon bonds with maturity 1 to 10 years

maturities we can observe a peak at approximately 1 percentage point lower than respective maximum for that time series. For short maturities we can see two peaks, one similar to that observed in case of longer maturities, while second is close to the minimum value of the time series across whole time period. This corresponds to the decline of yields with short maturities during the euro adoption period.

In table 1.2 we can find for each time series some descriptive statistics. We can see that the minimum value is close to zero and is smallest for the maturity of 5 years. The largest value is obtained for the longest maturity of 15 years. All of these values are from recent time period in year 2016 as we can see in figure 1.1 and 1.2. Second statistics computed for each maturity is the maximum. We can see a rising trend which however is not strict for short maturities. The average value follows same rising pattern which is not strict because of maturities of 11 and 12 years. The last statistic is standard deviation. We can see that these values are declining with increasing maturities which means that time series of long maturities are slightly more stable than their short-term counterparts.

Table 1.2: Basic descriptive statistics of yields from Slovak bonds for the whole time period

Maturity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Min	-0.30	-0.45	-0.54	-0.56	-0.53	-0.42	-0.26	-0.09	0.08	0.23	0.36	0.49	0.60	0.70	0.79
Max	5.64	5.34	5.42	5.45	5.45	5.42	5.42	5.70	6.02	6.25	6.41	6.52	6.58	6.64	6.69
Average	2.30	2.45	2.62	2.80	2.98	3.14	3.30	3.45	3.58	3.69	3.57	3.68	3.77	3.85	3.92
SD	1.83	1.74	1.71	1.69	1.65	1.61	1.57	1.53	1.49	1.45	1.52	1.49	1.47	1.45	1.43

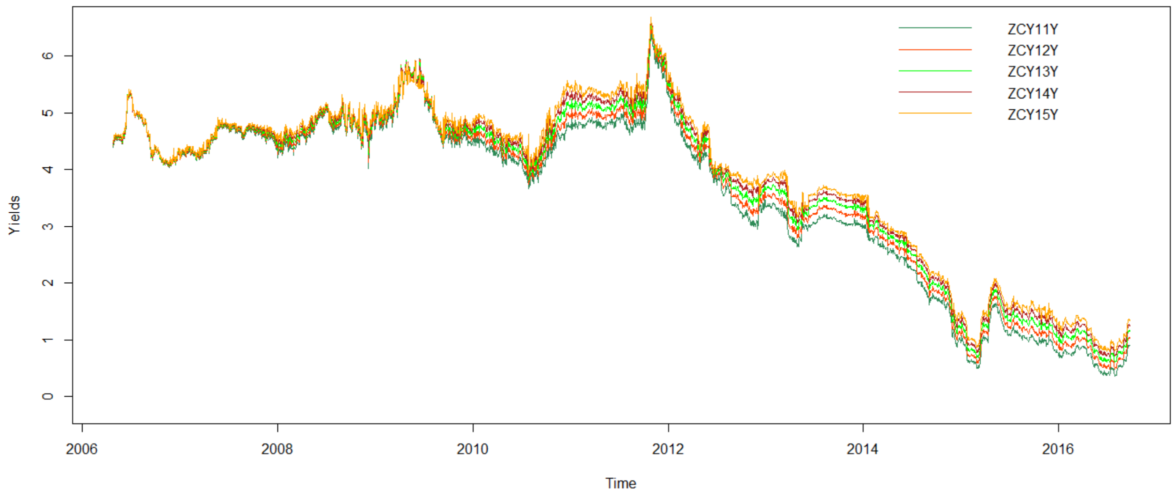

Figure 1.2: Behaviour of yields from Slovak government zero coupon bonds with maturity 11 to 15 years

1.2 The euro adoption

In this section we separate the time series into two parts. The first one covering time period before euro adoption starts at 7.1.2003 for maturities of 1 to 10 years and at 11.5.2006 for maturities of 11 to 15 years. This time period ends at the end of year 2007 for all maturities. We exclude years 2008 and 2009 to eliminate the transition phase of euro adoption. Second time period after euro adoption therefore starts at 1.1.2010 and ends at 31.10.2016. For each time period we compute similar statistics as in section 1.1.

In figures 1.4, 1.5 and 1.6 we can see comparison of histograms of yields for two considered time periods. First difference between time period before euro adoption and after euro adoption is lower overall level of yields in case of period after euro adoption which is significant mostly for short maturities. Another difference is that we can observe usually only one peak in time period before euro adoption while during latter time period there are more peaks. This means that yield distribution during time period before euro adoption is more stable and therefore is easier to model and potentially forecast. This observation is confirmed and further elaborated in section 2.6.1.

Concerning comparisons of basic statistics that can be seen in table 1.3 we can observe main pattern consisting of overall decline of level of yield curve that can be seen particularly in values of minimum and average that are significantly lower for time period after euro adoption. Exception to this pattern is slightly higher values of maximum for time period after euro adoption in case of long maturities. However maximal values are indeed

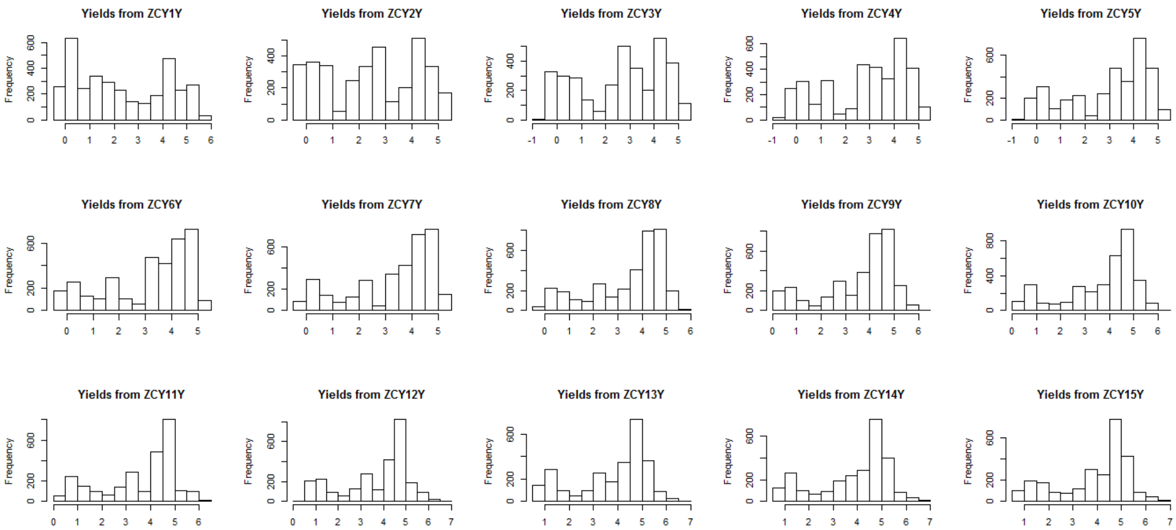

Figure 1.3: Histograms of yields from different maturities

lower for short maturities. This means that with euro adoption the yield curve did not experience just parallel shift but also change in slope. Standard deviations are decreasing with increasing maturities for time period before euro adoption. For time period after euro adoption we can see increase of standard deviation with higher maturities which indicates also structural change in nature of time series. Interesting observation is special case of short maturities where standard deviation for time periods before and also after euro adoption is lower than standard deviation of whole time period. This is mainly due to the significant decrease of level of yields from bonds with these maturities. We can also see overall higher standard deviations in case of time period after euro adoption. This observation was also observed and commented in histograms 1.4, 1.5 and 1.6.

All this evidence in descriptive statistics point out that Slovak yield curve experienced a structural break with euro adoption. We will take this fact into account in our following work and we will aim at finding a good way to overcome this modelling issue.

Table 1.3: Basic descriptive statistics of yields from Slovak bonds where BEA represents time period before euro adoption and AEA represents time period after euro adoption

Period	Statistic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
BEA	Min	2.25	2.42	2.52	2.66	2.79	2.91	3.03	3.12	3.17	3.21	4.04	4.05	4.07	4.08	4.10
AEA	Min	-0.30	-0.45	-0.54	-0.56	-0.53	-0.42	-0.26	-0.09	0.08	0.23	0.36	0.49	0.60	0.70	0.79
BEA	Max	5.64	5.34	5.42	5.45	5.45	5.42	5.42	5.44	5.46	5.47	5.40	5.40	5.40	5.40	5.41
AEA	Max	2.85	3.15	3.61	4.04	4.42	4.86	5.32	5.70	6.02	6.25	6.41	6.52	6.58	6.64	6.69
BEA	Average	4.15	4.10	4.12	4.17	4.23	4.29	4.34	4.39	4.43	4.48	4.53	4.55	4.57	4.59	4.60
AEA	Average	0.75	0.98	1.21	1.45	1.71	1.96	2.20	2.42	2.63	2.81	2.98	3.12	3.25	3.36	3.45
BEA	SD	0.90	0.80	0.74	0.69	0.65	0.62	0.60	0.58	0.57	0.56	0.28	0.28	0.28	0.28	0.28
AEA	SD	0.75	1.00	1.18	1.31	1.40	1.46	1.50	1.53	1.54	1.55	1.56	1.56	1.56	1.56	1.56

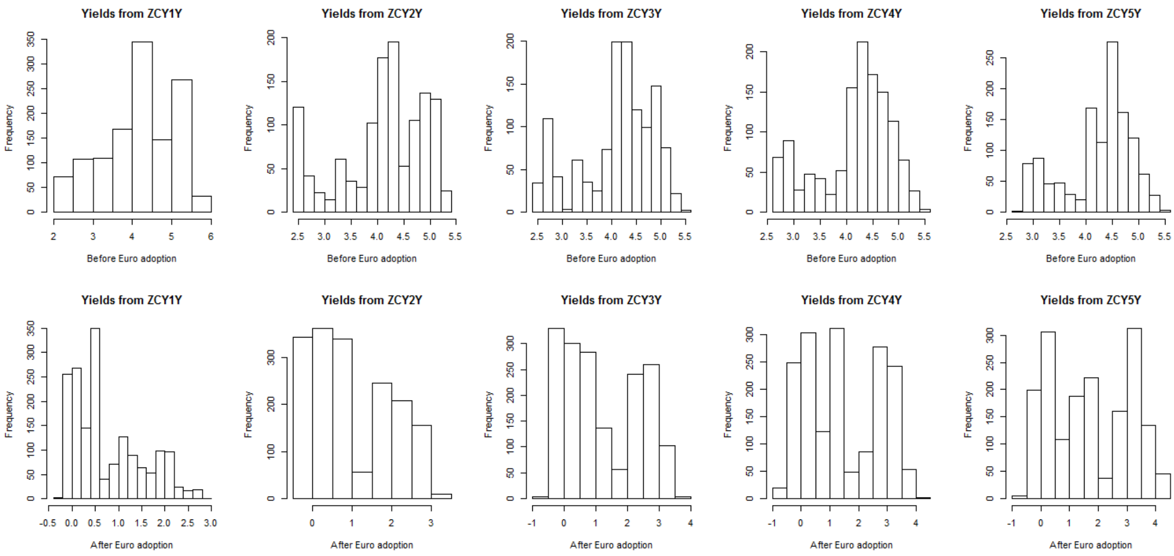

Figure 1.4: Comparison of histograms of yields for maturities of 1 to 5 years for time period before and after euro adoption

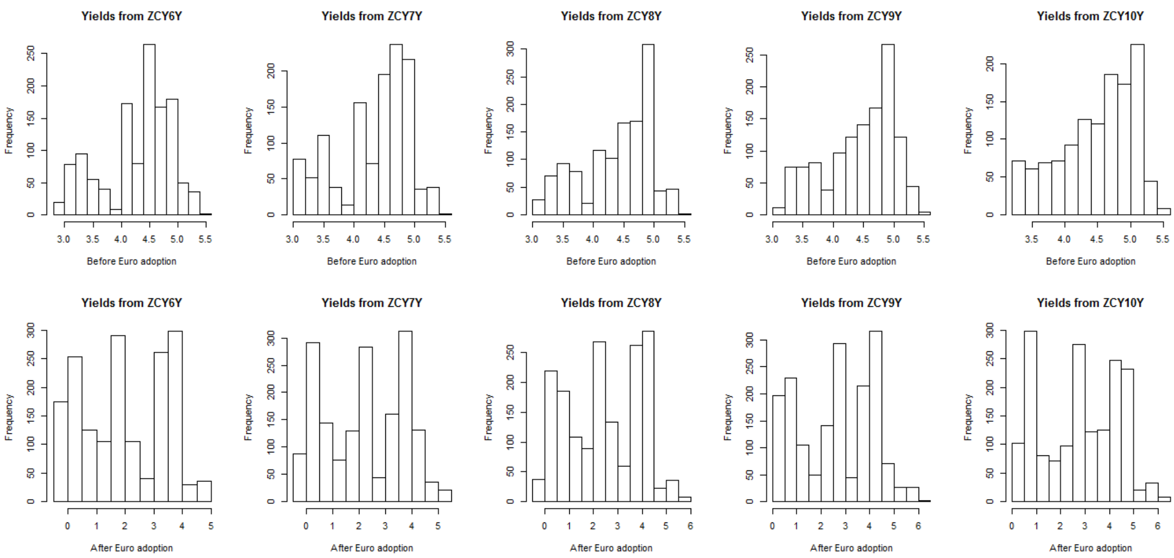

Figure 1.5: Comparison of histograms of yields for maturities of 6 to 10 years for time period before and after euro adoption

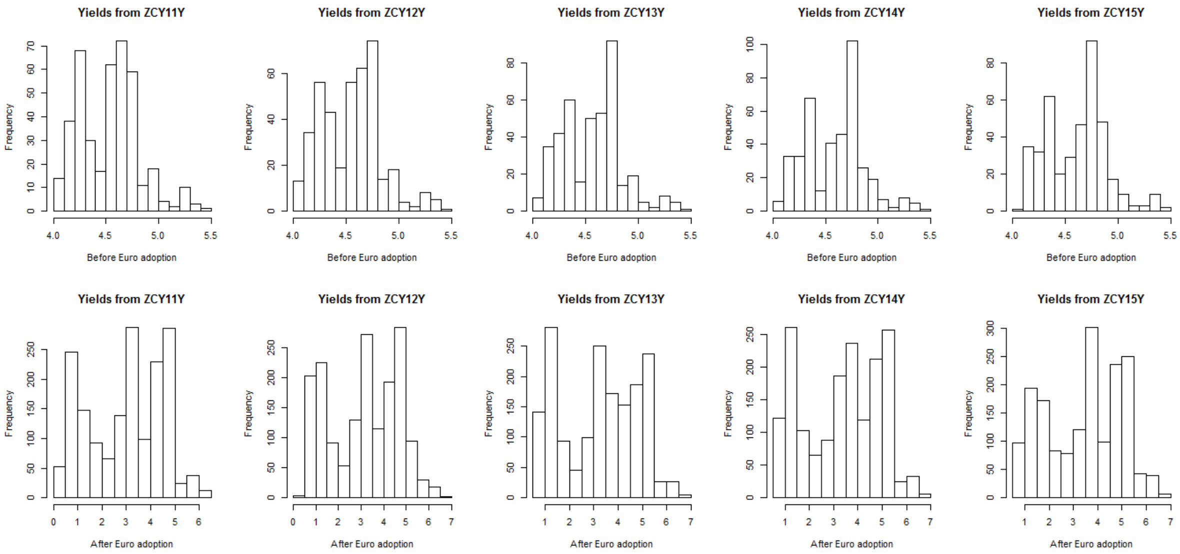

Figure 1.6: Comparison of histograms of yields for maturities of 11 to 15 years for time period before and after euro adoption

UNIVARIATE TIME SERIES APPROACH

In this chapter, we want to look at time series of yields from Slovak Government Zero Coupon Bonds separately for each maturity. Great disadvantage of this approach is that we ignore mutual dependence of yields of different maturities. However we take this fact into account in other modelling techniques so we can evaluate importance of these dependencies on the underlying process.

At first, we model yields from Zero Coupon bond with maturity of 1 year describing all the steps. Next, we use the same procedure to model also yields with longer maturities. Theoretical backround of framework used to analyse this time series can be found in 2.1.

2.1 Theory of time series

In this section, we briefly introduce the framework we use for analysing the time series.

2.1.1 ARIMA

$ARMA$ modelling, defined in 2.2 is used to analyse and model behaviour of so called stationary time series. $ARIMA(p, k, q)$ modelling is used, when we work with differences of order k from the original time series. It is necessary to take differences if the process is not stationary or there is a unit root present. We define stationary process same way as it is defined in [8].

Definition 2.1 (White noise)**.** *Random process u_t is called white noise if and only if:*

(1) $E[u_t] = 0 \ \forall t$

(2) $Var[u_t] = \sigma^2 \ \forall t$

(3) $Cov[u_t, u_s] = 0 \ \forall t \neq s$

Definition 2.2 (ARMA model)**.** *Let u_t be white noise. Then an ARMA(p,q) process is a process that contains p autoregressive components and q moving average components. This process can be expressed as following:* $x_t = \delta + \alpha_1 x_{t-1} + ... + \alpha_p x_{t-p} + u_t - \beta_1 u_{t-1} - ... - \beta_q u_{t-q}$.

Definition 2.3 (Stationary process, from [8])**.** *Process x_t is stationary, if and only if it fulfills following conditions.*

(1) $E[x_t] = \mu \ \forall t$

(2) $Cov[x_t, x_s] = \gamma(|t-s|) \ \forall t, s$

Statistical test 2.1 (Ljung-Box test)**.** *Ljung-Box test is used to test the hypothesis, whether all autocorrelations up to lag* m *are simultaneously equal to zero.*

The test statistic is defined as $Q = T(T+2)\sum_{j=1}^{m} \frac{\rho(j)^2}{T-j} \underset{T\to\infty}{\sim} \chi^2_m$. *If this test is used for testing autocorrelation of residuals in an* $ARIMA(p,i,q)$ *model, degrees of freedom of the Chi-squared distribution is set to* $m-p-q$ *instead of* m.

Definition 2.4 (Spectrum of a time series)**.** *For a stationary time series* Y *with autocovariances* $\{\gamma_j\}_{j=-\infty}^{\infty}$ *, we define its spectrum as* $s_Y(\omega) = \frac{1}{2\pi}\sum_{j=-\infty}^{\infty} \gamma_j e^{-i\omega j}$, *where* i *is imaginary unit.*

2.2 Unit root

For finding a good model, it is necesarry to check, whether there is unit root in our time series or the process is stationary[1]. In the figure 2.1 we can see behaviour of the process in time.

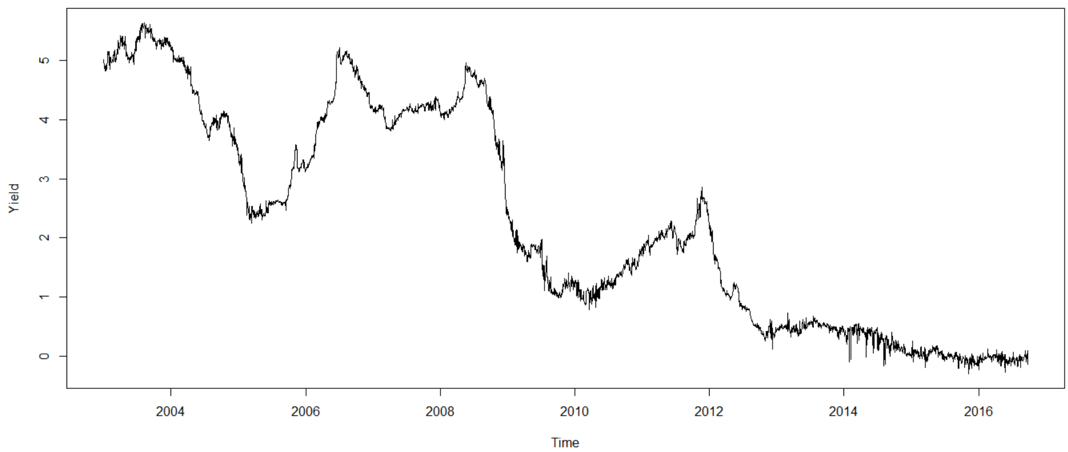

Figure 2.1: Time series of yields from zero coupon bond with maturity 1 year

To test the hypothesis that we have no unit root in the time series, we use three statistical tests as done in [3].

The first one is Adjusted Dickey-Fuller test, for further explanation of how this test works, please see [9] or [6]. The null hypothesis of this test is that there is a unit root in

[1] By stationarity we mean so called weak stationarity defined in 2.3.

Table 2.1: Results of ADF and KPSS tests applied on time series of yields from zero coupon bonds with maturity 1 year consisting of values of test statistics for each test as well as three critical values

Type of test	Test statistic	1% crit. v.	5% crit. v.	10% crit. v.
ADF with drift	-0.9411	-3.43	-2.86	-2.57
ADF with drift and trend	-1.4348	-3.96	-3.41	-3.12
KPSS with drift	29.0651	0.347	0.463	0.739
KPSS with drift and trend	1.1224	0.119	0.146	0.216

the observed process. As input parameter for this test, we have to specify, whether there is a drift, drift and trend or none of them. From the figure 2.1 we can not say, whether there really is a trend or just couple of jumps that change behaviour of the process significantly. However, presence of a drift is evident, therefore we try running the test twice with two different input options. Results of these tests can be seen in table 2.1. With the trend option we get test statistic equal to -1.4348, while 10 percent critical value is equal to -3.12. If we change the input option to drift, we get test statistic equal to -0.9411, while 10 percent critical value is equal to -2.57. We reject the null hypothesis if the test statistic value is lower that chosen critical value. In both cases, test statistic is higher than 10 percent critical value, so we don't refuse the null hypothesis.

Second test used is Kwiatkowski-Phillips-Schmidt-Shin test further explained in [11]. In this method, two models can be considered. One with a drift and one with a linear trend. The null hypothesis is stationarity of observed process in time. Similarly to ADF test, we consider both options observable in table 2.1. Model with drift has test statistic of value 29.0651 and the most strict quoted 1 percent critical value is 0.739. Specifying model with trend, we get value of test statistic 1.1113. 1 percent critical value is equal to 0.216. Conclusion is that we refuse the null hypothesis and therefore stationarity of the process. This result is in accord with the ADF test.

The last method we use is Phillips-Perron unit root test, which is non-parametric and robust technique introduced in [7]. Null hypothesis in this case is presence of unit root. P-value is equal to 0.695, so we have no reason to not refuse the null hypothesis. Again, we have obtained same result that there is not enough evidence to declare that there is no unit root in the observed time series.

The final result of this analysis is that we have strong evidence that there is a unit root in the time series and that time series is not stationary. We run same tests for the first differences of original time series. Results are in all cases opposite to the results of tests used on original time series as you can see in table 2.2 where we chose type of test as none since there is no significant drift nor trend present. P-value of Phillips-Perron test in this case was equal to 0.01. According to these results, we should use first differences for further analysis.

Table 2.2: Results of ADF and KPSS tests applied on first differences of time series of yields from zero coupon bonds with maturity 1 year consisting of values of test statistics for each test as well as three critical values

Type of test	Test statistic	1% crit. v.	5% crit. v.	10% crit. v.
ADF without drift	-43.6256	-2.58	-1.95	-1.62
KPSS without drift	0.0928	0.347	0.463	0.739

2.3 ARIMA modelling

We want to find a suitable ARIMA model for the observed time series. The theoretical background can be found in 2.1.1.

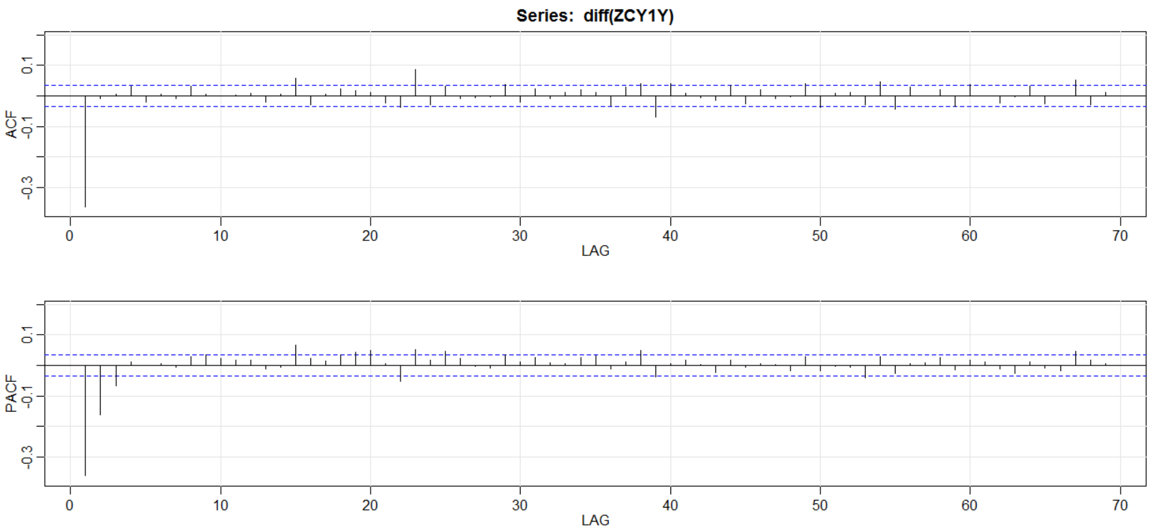

Figure 2.2: Sample autocorrelation and partial autocorrelation functions

In the figure 2.2 we can see autocorrelation function of the differenced time series, further explained in [1] and partial autocorrelation function of the differenced time series, further explained in [13]. If we look at PACF, we might suggest an AR(3) model because of the first three lags being significantly non-null. But the ACF lags don't look like they are converging to 0, having values of quite large lags significantly non-null. If we look at first at ACF, we see that the first lag is significantly bigger than all the other lags, wich are more or less close to zero. This observation might lead to choosing MA(1) model, but as we can see, also PACF doesn't look like converging to zero with bigger lags.

The fact, that ACF and PACF terms of bigger lags are significantly different from zero might make finding appropriate model difficult. We tried all combinations of $ARIMA(p, 1, q)$ models up to $p < 15$ and $q < 15$. We looked at quality of residuals of fitted models, more precisely to their sample autocorrelations. We used Ljung-Box test, further described in 2.1, testing whether all autocorrelations up to certain lag are different from zero. We also looked at ACF of residuals compared to approximate 95% interval of certainty $\pm 2/\sqrt{T}$. The last criterium was BIC. Using this approach, we have chosen $ARIMA(3, 1, 0)$ model.

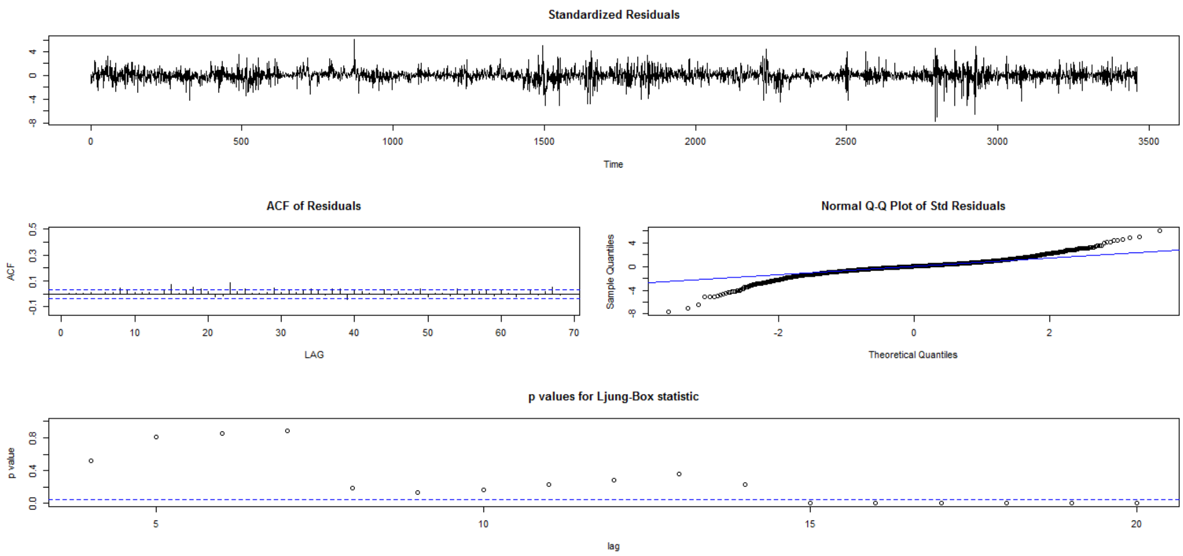

Figure 2.3: Behaviour of residuals in time, autocorrelation of residuals, QQ plot and p-values of Ljung-Box test for $ARIMA(3, 1, 0)$ model

As we can see in figure 2.3, even the best model doesn't describe behaviour of the time series perfectly. The fitted model is: $\triangle x_t = -0.0014 - 0.4334 \triangle x_{t-1} - 0.1907 \triangle x_{t-2} - 0.0661 \triangle x_{t-3}$, where $\triangle x_t = x_{t+1} - x_t$.

In figure 2.4 we can see that modules of all roots of characteristic polynome $1+0.4334L+0.1907L^2 + 0.0661L^3$ are bigger than 1, so the process is stationary.

Final conclusion may be that we can find a model that is not that bad, but it just can not describe the whole complex movements of yields in time. The reason can be that the estimated parameters are constant in time so the model can not describe when the process changes nature at some point. This idea is used for analysis in section 2.6.

2.4 Spectral analysis

Spectral analysis searches for hidden periodicity in a time series. From the spectrum, defined in 2.4 we can see, which frequencies have the biggest impact on variance of the pro-

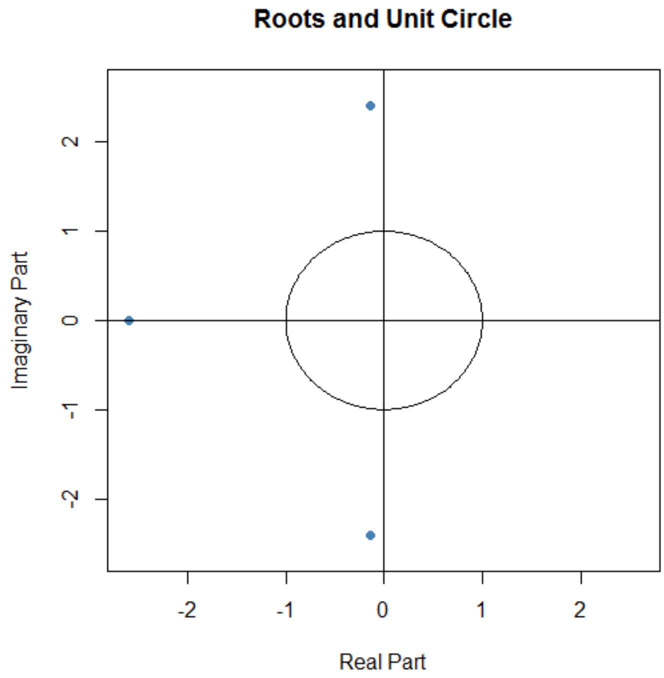

Figure 2.4: Roots of $ARIMA(3,1,0)$ process together with unit root

cess. In figure 2.5 we can see that there is a peak around $frequency = 4$. Since two previous frequencies and one consecutive frequency have values of spectrum not significantly lower than the maximum one, we compute weighted average of four frequencies with maximal values of spectrum. We compute final frequency as $freq = w_2 freq_2 + w_3 freq_3 + w_4 freq_4 + w_5 freq_5$, where $w_i = \frac{spectrum_i}{\sum_{j=2}^{5} spectrum_j}$. This way, we get $freq = 0.2447651$. Final step is to compute the period: $P = 1/freq = 1/0.2447651 = 4.08555$. We see, that spectral analysis has found a periodic movement in the yields of 1 year zero coupon bonds with period of approximately 4 years.

2.5 Yields with longer maturities

We perform same analysis on time series of yields from Slovak government zero coupon bonds with higher maturities. At first, we try to find unit root using Phillips-Perron test. P-values of this test can be found in table 2.3. All p-values are bigger than 0.5 so we don't refuse the null hypothesis of unit root in any of these cases. This means that we work with at least first differences. Applying same test to first differences results in p-values equal to 0.01 so we refuse the null hypothesis. First differences are then used for further analysis.

We try to find the best model for time series of yields for each maturity. Detailed steps

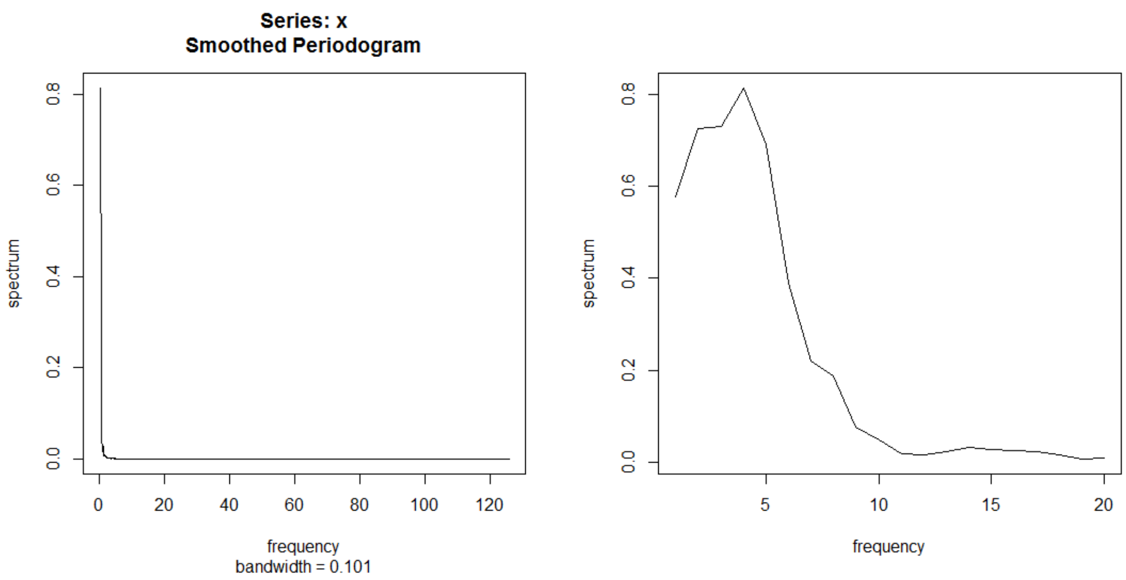

Figure 2.5: Full sample spectrum and first 20 frequencies of sample spectrum of yields from Slovak government bonds with maturity of 1 year

Table 2.3: P-values of Phillips-Perron test

Maturity of bond	1	2	3	4	5
P-value	0.6848862	0.6458403	0.6348452	0.6692528	0.7081263
Maturity of bond	6	7	8	9	10
P-value	0.7381668	0.7598119	0.7773977	0.7923337	0.8005964
Maturity of bond	11	12	13	14	15
P-value	0.5739533	0.6027349	0.6236036	0.6337487	0.6311513

can be found in 7. Final orders of estimated models can be found in table 2.4. All the models have been tested for stationarity and invertability.

2.6 Split time series

2.6.1 Period before the euro adoption

We have seen that constancy of parameters in ARIMA model is a great disadvantage, because yields from government bonds are influenced by number of external factors. Influence of such factor can be complete change of nature of process in time, or even substantial change of the whole yield curve. One such event can be euro adoption in Slovakia on 1.1.2009. This can be seen in figure 2.6 as the biggest decline in observed period.

This observation leads us to splitting the time series into two parts: before euro adoption and after euro adoption. At first we take time window up to the start of year 2008 so we minimize the influence of euro adoption on the time series. Behaviour of time series

Table 2.4: Models that fit the best for time series of yields from bonds with each maturity. First number contains number of autoregressive components, second number describes differences of which order has been taken and the last number for each maturity is number of moving average components.

Maturity of bond	1	2	3	4	5
ARIMA process	$(3,1,0)$	$(8,1,1)$	$(2,1,3)$	$(0,1,3)$	$(0,1,3)$
Maturity of bond	6	7	8	9	10
ARIMA process	$(3,1,3)$	$(0,1,7)$	$(0,1,7)$	$(0,1,8)$	$(0,1,9)$
Maturity of bond	11	12	13	14	15
ARIMA process	$(0,1,8)$	$(0,1,8)$	$(0,1,8)$	$(0,1,1)$	$(5,1,2)$

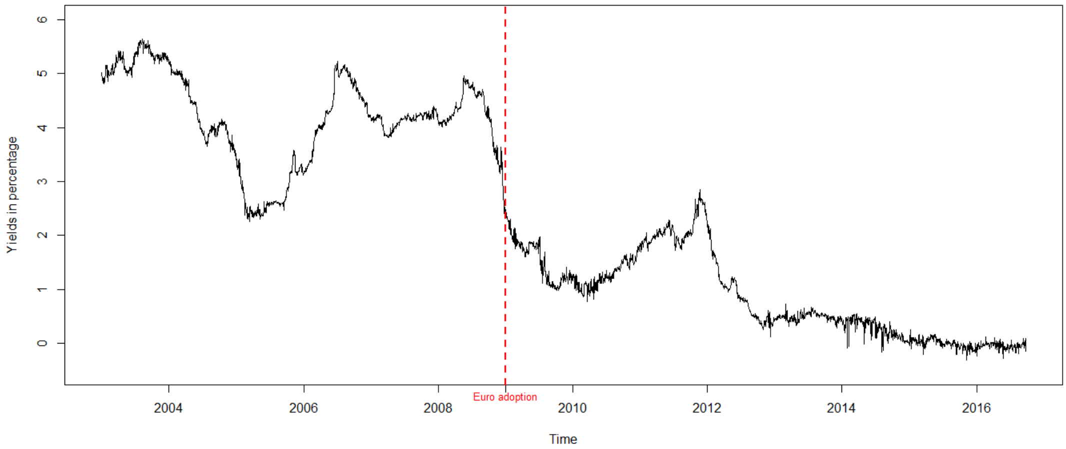

Figure 2.6: Behaviour of yields from Slovak government zero coupon bonds with maturity of 1 year and with highlighted date of euro adoption

during this period can be seen in figure 2.7.

We run same three tests for unit root as in section 2.2. All three tests provide strong evidence of unit root in the time series so we have to take differences. Testing differenced time series, we get same result as with the whole time period, that there is no unit root present.

We have found $ARIMA(5,1,3)$ stationary and invertible model with substantially better residuals than in case of the best model for the whole time series. This fact is illustrated in figures 2.8 and 2.9 where we can see p-values for Ljung-Box test on residuals. In the first case, p-values for lags 15 and higher are bellow critical line of 5%. In the case of shorter observed period, we can see that all p-values are significantly above 5%. We can conclude that we have found better model that describes well movement of yields in the period before euro adoption than the whole period. The reason behind this may be that the underlying process did not change nature during the observed period and therefore it is possible to

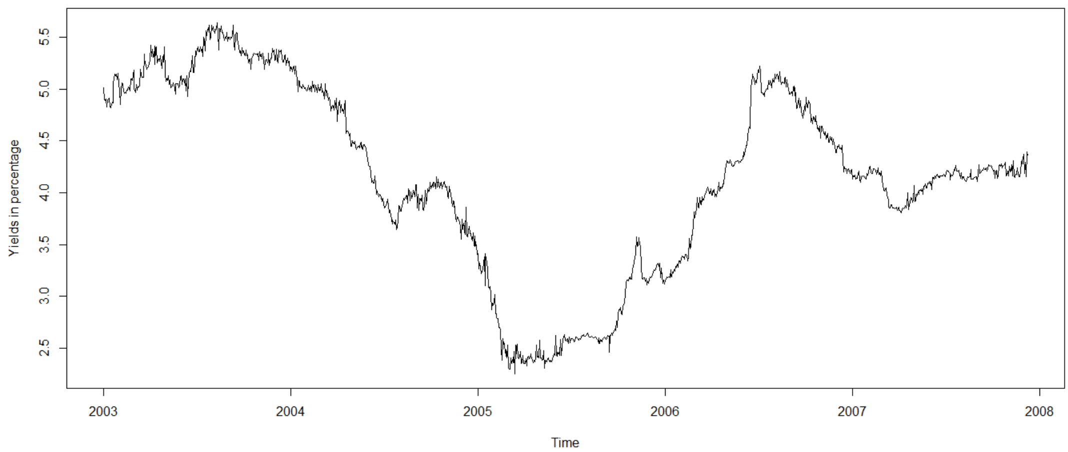

Figure 2.7: Behaviour of yields from Slovak government zero coupon bonds with maturity of 1 year before euro adoption

model it using autoregressive approach.

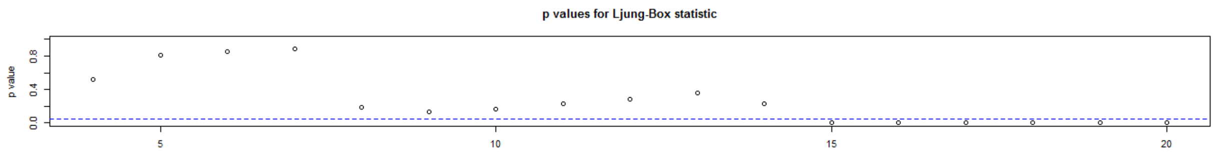

Figure 2.8: Ljung-Box test for maximum of 20 lags for the original time series

Concerning other maturities, we worked with first differences because p-values of Phillips-Perron test were higher than 0.6. We have found substantially better models for the period before euro adoption than for the whole period as you can see in results from code in the Attachments. This may be due to a simple fact that we have less data to be fitted so it is expected to model it more easily, however we think that the main reason is that during period before euro adoption there was not an influential event that would change the nature of process. We conclude that using ARIMA modelling on this period is a good way of modelling the process and that we can describe behaviour of yields in time quite well using this approach.

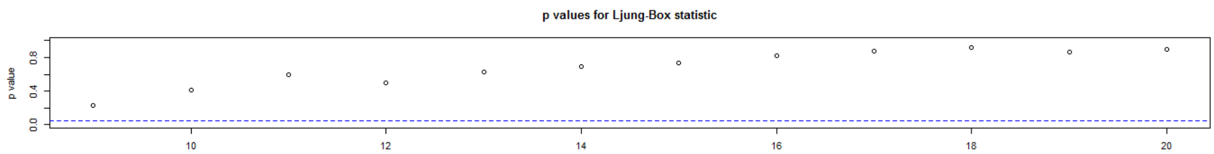

Figure 2.9: Ljung-Box test for maximum of 20 lags for the time series before euro adoption

We tried to find some periodic movements in time series of yields using spectral analysis but as expected, the time period was too small to detect any such movement.

2.6.2 Period after the euro adoption

In the figure 2.10 we can see behaviour of yields from 1 year zero coupon bond. Observed period starts at 1.1.2010 so one year after euro adoption.

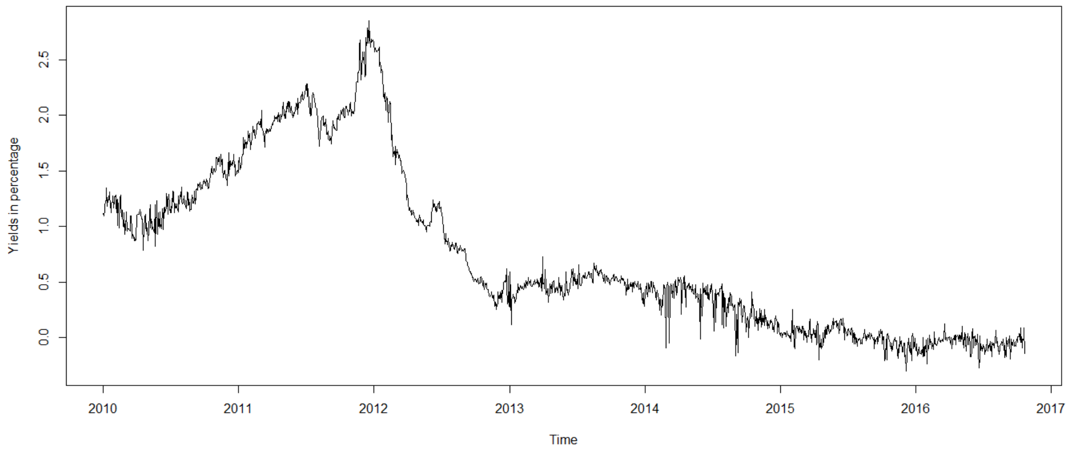

Figure 2.10: Behaviour of yields from Slovak government zero coupon bonds with maturity of 1 year after euro adoption

We use same steps as in the case of period before euro adoption. All of these steps can be found in the Attachments. At first, we use Phillips-Perron test to find out, whether we should work with differences. As all p-values are higher than 0.35 we conclude that differencing is necessary. Applying same test to first differences gives us p-values of 0.01 for all maturities so we proceed with first differences to ARIMA modelling.

Concerning residuals, we have found better models than for the whole time period, but globally worse than in the case of period before euro adoption. This may be due to the fact, that in this period, there was debt crisis in eurozone which influenced to some extent behaviour of yields in the whole Europe. As the debt crisis lasted for a longer period of time which is not clearly defined, we can not split this time series to get better results. Also the time horizon would be quite small so fitted models might be noisy.

Spectral analysis has not found any periodic pattern in the time series. This result is in accord with previous results in this field.

2.7 Conclusion

There is a wide discussion in the literature concerning stationarity of interest rates. In [14], authors suggest that not refusing the null hypothesis of unit root by ADF and PP tests might be caused by small sample size. This is probably not our case, because we have daily data from multiple years and KPSS test refused the null hypothesis so there is enough evidence that sample size is not an issue here. Another possible cause of unit root is suggested in [12] where they find significant influence of structural breaks or regime switches concluding into unit root. Several regime switches has been observed in the time series of yields from Slovak government bonds so this could be the reason for not refusing the unit root hypothesis. However we obtained same results after splitting the time series. More structural breaks can be present during the observed period that we did not include in our analysis that could possibly cause the non-stationarity of the time series. As we can not perfectly distinguish such events, we can not conclude that this is the main reason. Also after further splitting we could have issues with sample size and the tests being not powerful enough. In [12], authors suggest that reason why interest rates are not stationary might be volatility of inflation which is due to exogenous shocks in the economy that have a long memory. Therefore mean-reversion of nominal interest rates is not observed even on a longer time span. This might be also reason why we observe presence of unit root in the time series as there were multiple such shocks that could influence inflation and therefore also behaviour of yields on a larger scale.

Conclusion from ARIMA modelling is that for the whole time period, it is very hard to model behaviour of yields using parameters constant in time. Main reason behind this is that government bonds are influenced by many factors like monetary policy, which are time-varying in general. Autoregressive approach could capture this behaviour if these input factors follow same pattern. What can not be captured are structural changes in nature of time series. One such change occured with euro adoption. Other is probably connected with eurozone debt crisis. This means that if we want to model time period across these changes we should allow models with time-varying parameters or at least using some dummy explanatory variables that would indicate structural switch in yield curve.

By using spectral analysis, we have found that yields from zero coupon bond with maturity of 1 year are periodic with period of approximately 4 years. No such result was found for longer maturities or for split time series. This result may be due to the fact that the short end of yield curve is controlled mainly by cyclical variation of monetary policy and inflation trend. Long end of yield curve is composed mainly by inflation trend and risk premium. Spectral analysis captured this cyclical variation which is not present for maturities of 2 years or more.

PRINCIPAL COMPONENTS ANALYSIS

3.1 Theory of principal components

We briefly introduce definition of principal components. For further explanation of theory behind principal components analysis, please see [2].

Definition 3.1 (Principal components, from [2])**.** *If X is any random vector having finite variance, let*

$$Var(X) = ODO^T$$

be the spectral decomposition of its variance matrix. Consider the random vector

$$Y = O^T X.$$

(We are using a linear transformation that is derived from the spectral decomposition, O being the same matrix in both places.) Then

$$Var(Y) = O^T Var(X) O = O^T ODO^T O = D,$$

where we used the defining property of orthogonal matrices

$$(O^T = O^{-1}).$$

Thus Y has a diagonal variance matrix. Hence, since the off-diagonal elements of the variance matrix are covariances, the components of Y are uncorrelated. And, since the diagonal elements of the variance matrix are variances and the diagonal elements of D are the eigenvalues of $Var(X)$, the variances of the components of Y are the eigenvalues of the variance matrix of X. The components of Y are called the principal components of X. Since an orthogonal matrix is invertible, we also have

$$X = OY.$$

This expresses an arbitrary random vector X as a linear combination of uncorrelated random variables (its principal components).

3.2 Application of PCA

When estimating ARIMA models for time series of yields from bonds with different maturities, we noticed that a model which suits the best for a certain maturity offered often quite good fit for time series of yields from bonds with maturities close to the original one. This means that yields from bonds with similar maturities tend to follow the same pattern. This observation can be seen in figure 1.1 but even better in figure 1.2. However we can see that with euro adoption, the whole yield curve changed its shape with short-term yields decreasing significantly and long-term yields even rising slightly. From this moment, we can see bigger spread between maturities than before. This spread stays stable for relatively long period, but lowers around year 2015 getting into similar form as in year 2005 but with paralel decline of the whole curve. In table 1.1 we can see sample correlations between yields from bonds with different maturities. For similar maturities close to diagonal of the variance matrix 1.1, sample correlation is close to 1. This observation supports our hypothesis about similarity of time series of yields with maturities close to each other.

It is therefore useful to take advantage of similarities between yields from bonds with different maturities to capture movement of the yield curve as whole. To detect by how many hidden factors we can model the behaviour of yields with all 15 maturities, we use principal components analysis. The key part of theory used can be found in 3.1. For more detailed theoretical approach, please see [2]. One disadvantage of PCA is time invariance so if we randomly shuffle the index of yields, output of PCA remains the same.

We proceed with taking same three time periods into account as in time series analysis: whole time period, period before euro adoption and period after euro adoption. Script used for this analysis can be found in the Attachments.

Proportion of explained variance by principal components for three time periods can be found in table 3.1. We can see that almost 100% of variance of yields can be explained with 5 first principal components in case of the whole time period and the time period after euro adoption. Almost 100% of variance for time period before euro adoption is explained with 6 principal components. More than 99% of variance in cases of all three time periods can be explained using just two principal components. These values can be seen graphically displayed in figures 3.1, 3.2 and 3.3 in a cumulative way.

The ability to capture main movements of yields curve using only small number of dimensions can be seen in amount of information that is lost when considering only the space given by some of the first principal components. We can do a transformation of

Table 3.1: Proportion of variance explained by principal components for three considered time windows

Principal component	Whole period	Before euro	After euro
1	0.9304	0.8538	0.9878
2	0.06545	0.1411	0.00935
3	0.00345	0.00290	0.00209
4	0.00068	0.00209	0.0007
5	0.00004	0.00015	0.00005
6	0.00000	0.000010	0.00000
7	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000
12	0.00000	0.00000	0.00000
13	0.00000	0.00000	0.00000
14	0.00000	0.00000	0.00000
15	0.00000	0.00000	0.00000

original yield data into base given by 1, 2 and 3 principal components in following way.

$$Y_i = O_i^T (X - \mu) \qquad i = 1, 2, 3,$$

where Y_i is a matrix with i rows and n columns representing transformed yields into base given by i first principal components. O_i is a $i \times 15$ transformation matrix with i eigenvectors belonging to i largest eigenvalues of $VAR(X)$ as rows. X and μ are $15 \times n$ matrices where X contains original yield data for 15 maturities and n days. μ is formed by average yield across time for each maturity, therefore contains n identical columns. Thanks to this operation, we have centered data around 0.

In the next step, we transform these yields back into original base, loosing some portion of information in the process.

$$X_i = O_i \times Y_i + \mu.$$

The last step is to plot these data together with the original time series to graphically see if we can obtain satisfying results using reduced number of dimensions.

In the figures 3.4, 3.5, 3.6 we can see that a great portion of yield movements can be described using small number of principal components. Especially using 3 principal components transformation which is plotted in green often coincides with original time series plotted in black color. For yields from bonds with longer maturities it may seem that only two first principal components can explain a lot of information, but are too inaccurate in case of yields with maturity of 1 year. First principal component describes suprisingly well

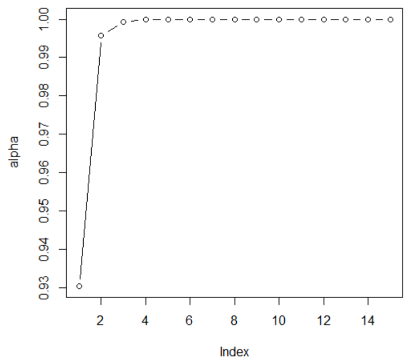

Figure 3.1: Cumulative explanatory power of principal components for the whole time period

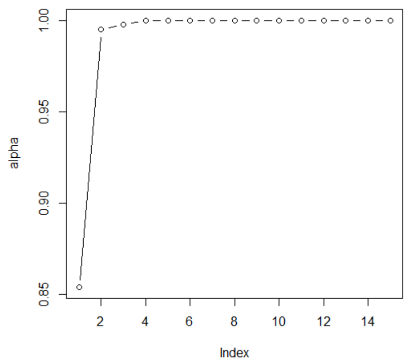

Figure 3.2: Cumulative explanatory power of principal components for time period before euro adoption

yields movements in case of longer maturities but the shorter the maturity gets, it lacks variability seen in the original time series.

We can also take a look at eigenvectors corresponding to the largest three eigenvalues to get an idea what movements they capture. This information can be seen in table 3.2. First principal component has very similar values for each maturity therefore we can say that it depends equally on yield with each maturity. This may be interpreted as level of yield curve. Second principal component depends positively on yields with short maturities the components of eigenvector declines monotonically with larger maturities becoming negative from maturity of 7 years. This monotonical dependance can be explained as slope of yield curve controlling how flat the yield curve is. Third principal component depends negatively on short and long maturities, on the other side it depends positively on mid-term maturities with maximum for 5 years. The interpretation of this observation can

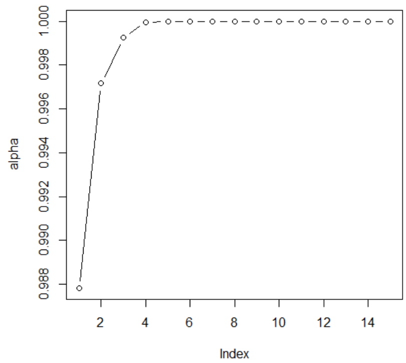

Figure 3.3: Cumulative explanatory power of principal components for time period after euro adoption

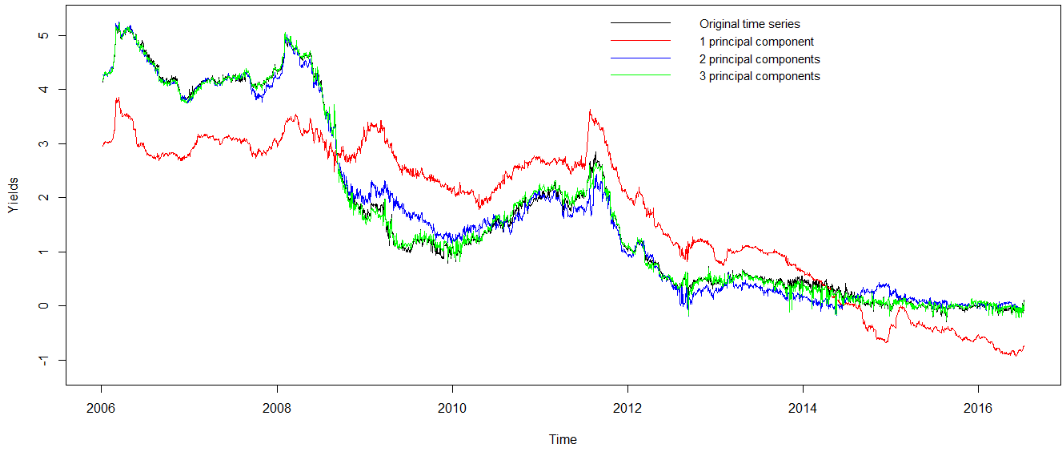

Figure 3.4: Original time series of yields from bonds with maturity of 1 year in black color together with transformed yields into base given by 1,2 and 3 principal components and backwards.

be that third principal component governs curvature of the yield curve.

3.3 Conclusion

We can conclude that using the transformation into base given by first three principal components, we loose only little amount of information which can be considered as noise. In addition to that we can interpret first three components quite intuitively as level, slope and curvature. This idea of reducing dimension of yields, then modelling this small amount of time series and finally transforming it back into original base can be used to predict future movements of yield curve. We use this approach and evaluate it in chapter 7. The

Figure 3.5: Original time series of yields from bonds with maturity of 7 years in black color together with transformed yields into base given by 1,2 and 3 principal components and backwards.

second conclusion from this chapter is that we can describe yield curve in time by using only three factors. Such a factor model is also Diebold-Li model, introduced in chapter 4.

Figure 3.6: Original time series of yields from bonds with maturity of 13 years in black color together with transformed yields into base given by 1,2 and 3 principal components and backwards.

Table 3.2: Eigenvectors corresponding to first three principal components

Yields	PC1	PC2	PC3
ZCY1Y	-0.2308411	0.57837043	-0.58014036
ZCY2Y	-0.2568372	0.42742090	-0.10414491
ZCY3Y	-0.2733832	0.29562798	0.17600945
ZCY4Y	-0.2812168	0.18527438	0.29856641
ZCY5Y	-0.2827637	0.09336351	0.32249771
ZCY6Y	-0.2802764	0.01682650	0.28915806
ZCY7Y	-0.2753683	-0.04670181	0.22490058
ZCY8Y	-0.2692292	-0.09923263	0.14605878
ZCY9Y	-0.2626083	-0.14233037	0.06277998
ZCY10Y	-0.2559710	-0.17728665	-0.01882720
ZCY11Y	-0.2496335	-0.20524295	-0.09518049
ZCY12Y	-0.2437205	-0.22709972	-0.16430207
ZCY13Y	-0.2383532	-0.24375814	-0.22540855
ZCY14Y	-0.2335250	-0.25588037	-0.27824721
ZCY15Y	-0.2292519	-0.26413677	-0.32304544

CHAPTER 4

DIEBOLD-LI

In this chapter, we at first introduce main technical points from article [4] and then we use their approach to model our data. We also higlight theoretical parts which can be done differently and we suggest some improvements, apply and evaluate them in chapter 7.

4.1 Theory

Main motivation for Diebold and Li to introduce their model is, as they state, that surprisingly little attention has been paid to the key practical problem of yield curve forecasting. The goal of this model is therefore to forecast yield curve movements with sufficient precision and time ahead.

As input data we have for each time moment yields for different maturities, in our case 15 yields with maturities from 1 to 15 years. Basic model to fit yield curve in one point of time is Nelson-Siegel model, which is three-component exponential approximation. Corresponding functional form can be seen in 4.1.

$$y_t(\tau) = \beta_{1t} + \beta_{2t}\Big(\frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau}\Big) + \beta_{3t}\Big(\frac{1-e^{-\lambda_t \tau}}{\lambda_t \tau} - e^{-\lambda_t \tau}\Big) \tag{4.1}$$

Parameter λ_t governs the exponential decay rate. Smaller the λ_t is, slower the decay rate gets. λ_t also governs where the loading on β_{3t} achieves its maximum. Traditional interpretation of parameters β is that β_{1t} may be viewed as long-term factor since the loading is equal to 1 which is a constant that does not decay with bigger τ. The loading on β_{2t} is a function that starts at 1 but decays monotonically and quickly to 0, hence it may be viewed as a short-term factor. The loading on β_{3t} starts at 0 (and is thus not short-term), increases, and then decays to zero (and thus is not long-term), hence it may be viewed as a medium-term factor.

However Diebold and Li propose another interpretation of factors β as level, slope and curvature. The long-term factor β_{1t} governs the level of yield curve. The argument for this interpretation is that increase in β_{1t} increases all yields equally and therefore the level of yield curve rises. The short-term factor β_{2t} can be interpreted as yield curve slope. This can be seen in the fact that the loading on β_{2t} decays with τ which means that the loading

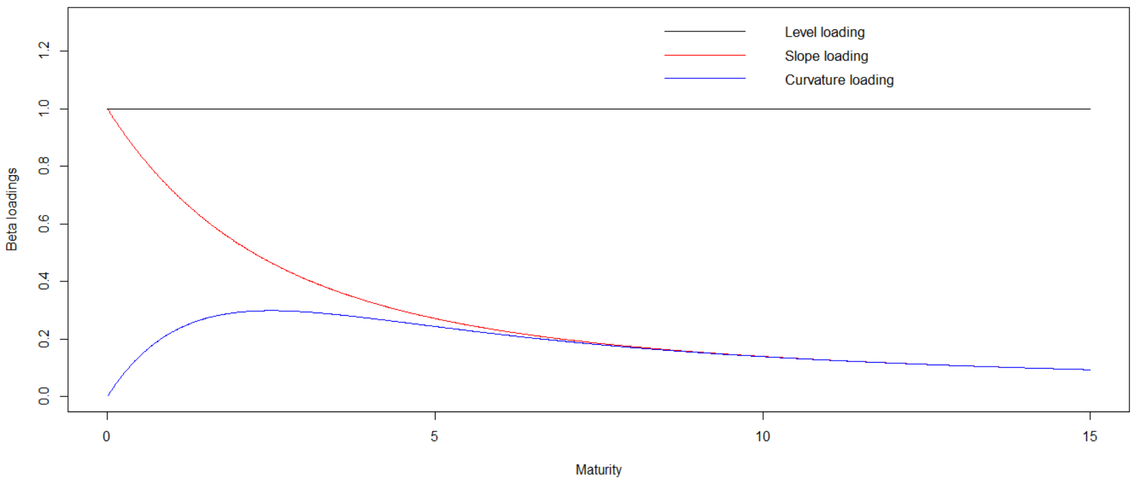

Figure 4.1: Loadings on parameters β in Nelson-Siegel model for $\lambda_t = 0.7173231$ and maturity given in years.

is the biggest for $\tau = 0$ and close to 0 for τ big enough. Parameter β_{3t} can be seen as yield curve curvature because its loading at first increases, then achieves a maximum for some $\widehat{\tau}$, and declines to 0. Therefore the factor β_{3t} affects mainly yields with maturities close to $\widehat{\tau}$.

To model fitted yield curve by Nelson-Siegel in time we can either perform nonlinear least squares estimation of parameters $\{\beta_{1t}, \beta_{2t}, \beta_{3t}, \lambda_t\}$ for each point in time or we can fix λ_t in time and perform ordinary least squares estimation of parameters $\{\beta_{1t}, \beta_{2t}, \beta_{3t}\}$. Diebold and Li suggest second approach because of its numerical triviality and simplicity of modelling. The question arises, of course, as to an appropriate value for λ_t. As this value directly influences the maximum of the third factor loading, they suggest to fix it on a value that maximizes the factor loading for maturity of 30 months. Then they come up with an estimate of $\widehat{\lambda_t} = 0.0609$, where maturity is given in months. However our results for this optimization problem are slightly different with $\widehat{\lambda_t} = 0.0598$. In our future work, we use our estimate of $\widehat{\lambda_t}$ instead of the value given by Diebold and Li.

As the choice of λ_t is to large extent heuristic, we suggest our own approach where we at first perform nonlinear least squares estimation of $\{\beta_{1t}, \beta_{2t}, \beta_{3t}, \lambda_t\}$, calculate average value of λ_t across time. We fix this value as $\widehat{\lambda_t}$ and perform linear least squares estimation of $\{\beta_{1t}, \beta_{2t}, \beta_{3t}\}$.

Next step is to model the behaviour of factors β in time. Diebold and Li uses basic AR(1) process. The yield forecasts in this case are:

$$\widehat{y}_{t+h/t}(\tau) = \widehat{\beta}_{1,t+h/t} + \widehat{\beta}_{2,t+h/t}\Big(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\Big) + \widehat{\beta}_{3,t+h/t}\Big(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\Big),$$

where

$$\widehat{\beta}_{i,t+h/t} = \widehat{c}_i + \widehat{\gamma}_i\widehat{\beta}_{it}, \quad i = 1, 2, 3.$$

We can see some space for different approach in this step as well. As authors suggest, vector autoregression model can be used instead of AR(1) processes for capturing mutual dependence of factors. However the authors suggest that this approach does not show better results mainly because of two reasons. Firstly, correlations between factors are not really significant and secondly in practice, forecasting with VAR often suffer from overfitting and estimation errors due to large number of parameters.

We suggest our own approach by going through same steps as in chapter 2 in order to find suitable $ARIMA(p, i, q)$ process to model the time series.

4.2 Application

We use original approach from the article [4] together with our changes, both illustrated in section 4.1 to model yield curve in time in order to get predictions. As we already pointed out, choice of λ_t is quite subjective so we not only use value of λ_t gained by maximization of loading on β_{3t} but we at first estimate independently parameters $\{\beta_{1t}, \beta_{2t}, \beta_{3t}, \lambda_t\}$ for each time and then calculate its average. The distribution of λ_t obtained can be seen in histogram 4.2.

Apparently the distribution is not centered around value $\lambda_t = 0.7173231$, which is value of λ_t given by Diebold-Li approach for maturity expressed in years. The average value is equal to 0.3504465 so more than two times lower than the original one. This is significantly different and therefore we also use this average value of λ_t in the next steps because it may provide us with different results.

After we choose the value of fixed λ we proceed to at first estimate parameters $\{\beta_{1t}, \beta_{2t}, \beta_{3t}\}$. Second step is to model these parameters by AR(1) model. We get theoretically poor quality of residuals because of very significant autocorrelation for the first lags. Since we evaluate the quality of models in out-of-sample forecasting results, we can not refuse this model only because of quality of residuals.

However we also use our classic approach on ARIMA modelling. At first we take parameters $\{\beta_{1t}, \beta_{2t}, \beta_{3t}\}$ estimated with $\lambda_t = 0.7173231$ and we look at the stationarity of these

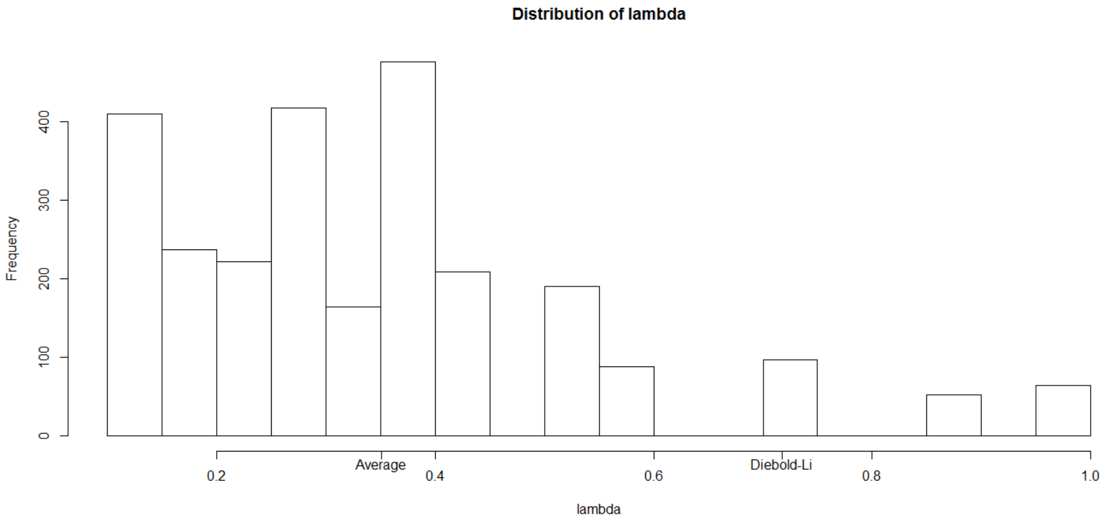

Figure 4.2: Distribution of λ_t calculated by nonlinear least squares optimization across time.

time series. Using same three tests as in chapter 2, we don't refuse the null hypothesis of unit root using Adjusted Dickey-Fuller test and Phillips-Perron test for all three time series besides the β_{3t} where we get $p - value = 0.01$. We refuse the null hypothesis of stationarity using KPSS test in all cases. Because of simplicity and better results of differentiated time series we choose to get first differences of the time series and perform same tests on this data. In this case we refuse the hypothesis of unit root and don't refuse the stationarity hypothesis. The significance level for all these tests on all time series was $\alpha = 0.05$.

The next step is to find the best possible ARIMA(p,1,q) model by looking especially at residuals. The chosen models are for β_{1t} ARIMA(0,1,1), for β_{2t} ARIMA(0,1,1) and for β_{3t} ARIMA(2,1,4). We also examine fitted processes for stationarity and invertibility with positive results.

For the fixed value of $\lambda_t = 0.3504465$ we get very similar results. Concerning refusal of hypotheses at significance level $\alpha = 0.05$ we get even exactly same results. We therefore work with first differences. Fitted models are in this case ARIMA(2,1,6) for β_{1t}, ARIMA(7,1,2) for β_{2t} and ARIMA(3,1,3) for β_{3t}. All these processes were tested positively on stationarity and invertibility.

One argument for modelling univariate time series rather than using vector autoregression was that the factors β_{it} are not significantlly dependent. Their respective correlation matrices for both fixations of parameter λ_t can be found in tables 4.1 and 4.2.

We can see that their respective correlation matrices are quite similar and that only β_{1t} is correlated somehow significantly with other parameters. Interesting result is negative

Table 4.1: Correlation matrix of $\beta_{it}, i = 1, 2, 3$, where $\lambda_t = 0.7173231$.

Parameter	β_{1t}	β_{2t}	β_{3t}
β_{1t}	1.0000000	-0.67187469	0.25330191
β_{2t}	-0.6718747	1.00000000	0.01104823
β_{3t}	0.2533019	0.01104823	1.00000000

Table 4.2: Correlation matrix of $\beta_{it}, i = 1, 2, 3$, where $\lambda_t = 0.3504465$.

Parameter	β_{1t}	β_{2t}	β_{3t}
β_{1t}	1.0000000	-0.57534245	0.27301238
β_{2t}	-0.5753425	1.00000000	0.04519802
β_{3t}	0.2730124	0.04519802	1.00000000

sign of correlation between factors of level and slope which means that when level of yield curve rises, in most cases slope of yield curve decreases. In other words with parallel shift of all yields upwards, the yield curve usually flattens and vice versa.

In chapter 3, we have interpreted first three principal components in similar way as Diebold and Li. One may ask, whether the time series of parameters β_{it} are similar to transformed yields into base given by first three principal components. As a first insight we can calculate their respective correlations. This can be seen in tables 4.3, 4.4 and 4.5.

Table 4.3: Correlation matrix between different level factors

Parameter	β_{1t}^{mean}	β_{1t}^{DL}	PC1
β_{1t}^{mean}	1.0000000	0.9447136	0.6782265
β_{1t}^{DL}	0.9447136	1.0000000	0.8538158
PC1	0.6782265	0.8538158	1.0000000

What one could expect is the fact that parameters $\beta_{it}, i = 1, 2, 3$ are strongly correlated as they are calculated in the same manner but with different parameter λ_t. Concerning correlations between parameters β_{it} and transformed yields using principal components we can see that in all cases the correlations are positive, somehow smaller in the case of level factor and λ_t given by average than in the case of its fixation at value given by Diebold-Li approach. However for the slope factor the correlation between parameters β_{it} and the second principal component is even higher than between themselves. Sample correlations for the curvature factor are around 0.6. This can be explained by the fact that the third component captured only little portion of variance so it can be capturing random noise to some extent and therefore we can not expect it to correspond perfectly to a certain curvature factor given by Diebold-Li approach.

Another way to look at the similarities between different essential factor approaches is to normalize the time series of transformed yields by principal components. This can be

Table 4.4: Correlation matrix between different slope factors

Parameter	β_{2t}^{mean}	β_{2t}^{DL}	PC2
β_{2t}^{mean}	1.0000000	0.8533396	0.9968706
β_{2t}^{DL}	0.8533396	1.0000000	0.8849532
PC2	0.9968706	0.8849532	1.0000000

Table 4.5: Correlation matrix between different curvature factors

Parameter	β_{3t}^{mean}	β_{3t}^{DL}	PC3
β_{3t}^{mean}	1.0000000	0.8653481	0.6549803
β_{3t}^{DL}	0.8653481	1.0000000	0.5712361
PC3	0.6549803	0.5712361	1.0000000

done by basic operations assuring that the transformed yields has same average as average of time series of parameters β_{it} calculated by using two different λ_t parameters. Also we transform the time series so it has the same variance as average variance of parameters β_{it}.

Then we plot these three triples in the figures 4.3, 4.4 and 4.5. We can see that behaviour of these time series is indeed similar for each factor. We can conclude that modelling these three factors is essential for the problem of yield curve modelling as we found it in two different approaches.

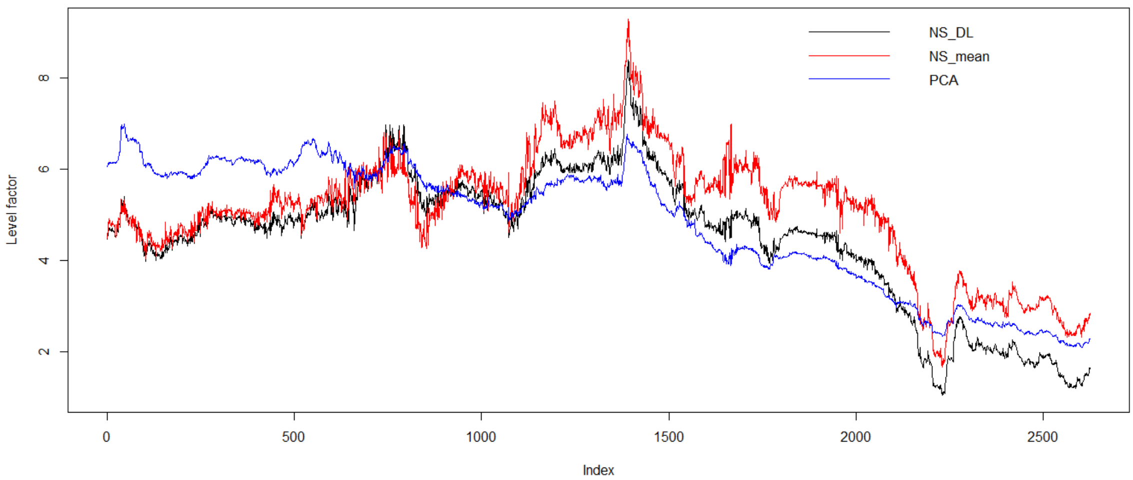

Figure 4.3: Time series of level factors calculated by different approaches.

4.3 Conclusion

We have seen that Diebold-Li approach shows some similarities with our PCA based transformation approach. However the biggest difference lies in the functional Nelson-Siegel

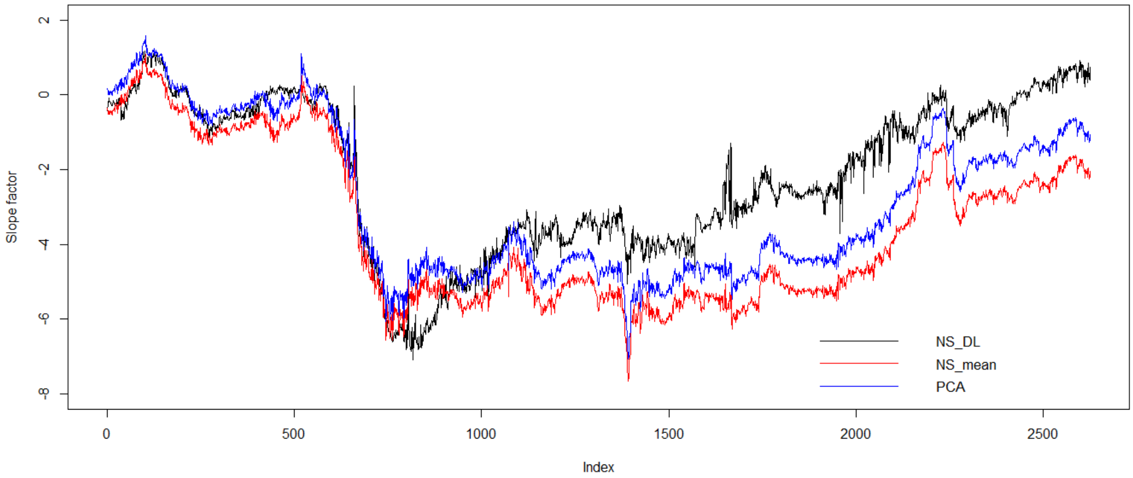

Figure 4.4: Time series of slope factors calculated by different approaches.

description of yield curve. This can be a great advantage if actual Slovak yield curve can be succesfully described by this function because we get value of yield to maturity for each chosen τ. On the other hand if Slovak yield curve has more complicated form that can not be modelled using this type of function we can end up with very bad estimates. In chapter 7 we will see that latter is the case probably because of complicated nature of Slovak yield curve during observed and predicted time period.

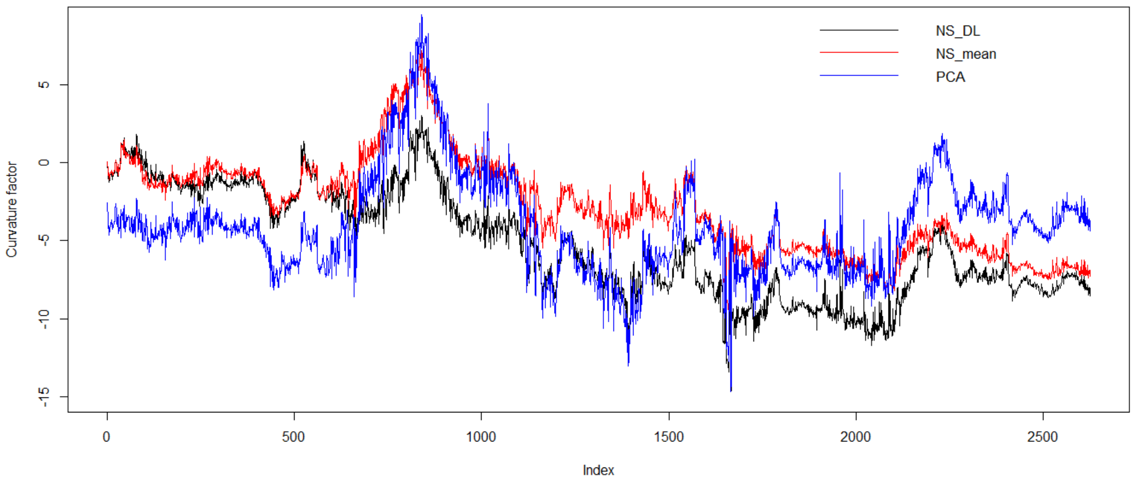

Figure 4.5: Time series of curvature factors calculated by different approaches.

REGRESSION WITH ARMA ERRORS

As we may have seen in chapter 2, adoption of euro changed substantially the nature of Slovak yield curve. We can take this observation into account using extention of ARIMA modelling with another time series that has impact on the time series of yields. Such extention is provided by adding a regressor term to the original ARIMA process. This model is defined in 5.1 in similar way as in [5].

Definition 5.1 (Regression with ARMA errors)**.** *Regression model with ARMA errors is defined by following equalities:*

$$y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_k x_{k,t} + n_t,$$

$$n_t = \alpha_1 n_{t-1} + ... + \alpha_p n_{t-p} + u_t - \gamma_1 u_{t-1} - ... - \gamma_q u_{t-q},$$

where y_t is time series we want to model, x_t is time series that has non-zero correlation with y_t, n_t is an ARMA process without intercept and u_t is white noise.

In practice, there is often confusion between proper use of regression models with ARMA errors and so called ARMAX models. In ARMAX models, the explanatory time series is added directly into ARMA equation so we get $y_t = \beta_0 + \beta_1 x_{1,t} + ... + \beta_k x_{k,t} + \alpha_1 y_{t-1} + ... + \alpha_p y_{t-p} + u_t - \gamma_1 u_{t-1} - ... - \gamma_q u_{t-q}$, where u_t is white noise. Using this kind of definition, it is harder to interpret the fitted parameters as they may capture some effects only partially and for example parameters β can not be intrepreted as direct effects of x_t on y_t since we have also lagged values of y_t and lagged values of u_t that model the value of y_t.

This is the reason we choose the regression model with ARMA errors for modelling univariate time series of yields. The question arises how to choose a suitable time series x_t. We want this time series to capture the effect of euro adoption which became Slovak currency on 1.1.2008. First idea is to create a dummy variable that has value 0 for dates before euro adoption and 1 for dates after euro adoption. Disadvantage of this approach is the observation that especially for yields with lower maturities, the change of time series did not come in an instant but was observed during longer period of time which can be seen in figure 2.6.

Therefore we can create another time series that has value 0 during time period before year 2007, increases linearly during years 2007 and 2008, reaching value 1 on 1.1.2009 and staying there for the time period afterwards. This approach captures the non-instantaneous nature of shock caused by euro adoption.

We described two possible choices of time series x_t. Now we must choose also orders of ARIMA(p,i,q) process for modelling residuals. Our first choice is closely related to univariate ARIMA modelling. If we want to eliminate the factor of chosen orders (p, i, q) on evaluation, we have to choose same orders for each maturity as in the case of ARIMA models without external regressor. As we modelled first differences, we take first differences $\triangle x_t = x_{t+1} - x_t$ and also first differences $\triangle y_t = y_{t+1} - y_t$ so it is in accord with the definition 5.1. Second choice of orders of ARIMA modelling for original univariate time series is basic AR(1) model because we will evaluate also this the most basic approach to model the original time series. However regression with ARMA errors can be used also in case of PCA and Diebold-Li approach. In these cases we model three time series of transformed yields for PCA and factors for Diebold-Li. We have seen in chapter 1 that euro adoption had impact mainly on the slope of the yield curve, which is specifically modelled using second principal component and second factor. This is the reason why we add exogenous time series to model these time series. Evaluation of these models can be seen in chapter 7.

VECTOR AUTOREGRESSION

In chapter 2 we modelled each time series separately. If we want to take advantage of knowledge of mutual dependences between yields with different maturities, we can use so called vector autoregression. This approach consists of modelling whole vector in time. Mathematically we define this process in 6.1.

Definition 6.1 (Vector autoregression)**.** $A\, n-dimensional$ *vector* y_t *follows a VAR(p) process described as follows.*

$$y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + CD_t + u_t,$$

where $A_i, i \in \{1, 2, ..., p\}$ *are* $n \times n$ *matrices of coefficients describing influence of* $i - th$ *lagged state* y_{t-i} *on the actual state of* y_t. $n \times k$ *coefficient matrix* C *describes influence of contamporenous exogenous variable* $k \times 1$ *matrix* D_t *on* y_t. $n \times 1$ *vector* u_t *assigns a spherical disturbance.*

We consider two possible vector autoregression models in order to model the yield curve. In both cases we choose $p = 1$ as the number of parameters to be fitted rises with each added lag by factor of $n \times n$ which in our case is equal to 225. In both cases we consider an intercept as part of descriptive equation and in one case we add also exogenous variable decribing euro adoption. We choose same approach as in chapter 5, where we defined the explanatory time series that was equal to 0 during time period before year 2007, increased linearly during years 2007 and 2008, reaching value 1 on 1.1.2009 and staying there for the time period afterwards.

In table 6.1 we can see fitted $\widehat{A}_1$ matrix. Since we have strongly autocorrelated time series one could expect that diagonal components would dominate others which is not really the case here. This observation might mean that we overfitted the model but it may also mean that yields with other maturities have higher impact on time series of yields with some maturity than the time series itself. This idea is supported by the fact that individual time series are strongly cross-correlated. However we can see that most of the parameters are not statistically significant which supports the idea of overfitting. In table 6.2 we can see fitted intercept vector $\widehat{C}$. In most cases, elements of $\widehat{C}$ are significantly different from zero so adding also the intercept term seems to be good choice.

In table 6.3 we can see fitted $\widehat{A}_1$ matrix of VAR(1) model with explanatory time series

Table 6.1: Matrix $\widehat{A}_1$ of estimated parameters from VAR(1) model

τ	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	11Y	12Y	13Y	14Y	15Y
1Y	0.93	0.16	-0.44	0.95	-0.91	0.41	-0.22	0.16	0.05	0.96	-2.44	0.50	1.39	0.12	-0.63
2Y	0.03	1.47	-2.06	2.19	0.79	-1.46	-1.63	1.76	-0.35	1.67	-1.75	0.38	0.28	-0.88	0.56
3Y	0.01	0.28	-0.13	1.31	-0.14	-0.11	-1.20	0.95	-0.12	0.85	-0.71	-0.01	0.17	-0.55	0.37
4Y	0.05	-0.34	0.87	0.42	-0.67	1.10	-0.61	0.15	0.14	-0.01	-0.08	-0.21	0.16	0.14	-0.12
5Y	0.07	-0.54	1.17	-0.13	-0.77	1.40	-0.14	-0.08	0.29	-0.44	0.15	-0.27	0.31	0.20	-0.22
6Y	0.07	-0.56	1.14	-0.36	-0.68	1.33	-0.17	0.04	0.64	-0.70	0.27	-0.16	0.20	0.03	-0.08
7Y	0.06	-0.46	0.85	-0.19	-0.56	0.92	-0.37	0.42	0.70	-0.53	0.30	-0.12	0.14	-0.32	0.18
8Y	0.05	-0.32	0.45	0.03	-0.23	0.55	-1.01	0.76	0.93	-0.30	0.29	0.10	-0.18	-0.58	0.45
9Y	0.04	-0.19	0.11	0.20	0.11	0.15	-1.30	0.93	0.90	-0.14	0.40	0.23	-0.25	-0.85	0.66
10Y	0.03	-0.01	-0.36	0.54	0.32	-0.07	-1.59	0.91	0.83	0.17	0.47	0.27	-0.48	-0.74	0.72
11Y	0.01	0.13	-0.68	0.65	0.50	-0.04	-1.80	0.82	0.62	0.24	0.73	0.32	-0.61	-0.72	0.81
12Y	-0.01	0.28	-0.97	0.77	0.64	-0.12	-1.68	0.63	0.45	0.37	0.60	0.39	-0.68	-0.42	0.77
13Y	-0.04	0.43	-1.17	0.67	0.86	-0.04	-1.68	0.51	0.37	0.23	0.55	0.42	-0.78	-0.07	0.74
14Y	-0.06	0.52	-1.23	0.54	0.80	0.27	-1.49	0.18	0.21	0.34	0.51	0.10	-0.76	0.26	0.79
15Y	-0.09	0.64	-1.34	0.44	0.83	0.37	-1.31	0.06	0.14	0.55	0.09	-0.11	-0.78	0.58	0.92

Table 6.2: Matrix $\widehat{C}$ of estimated intercept parameters from VAR(1) model

τ	Intercept
1Y	0.01
2Y	0.01
3Y	-0.01
4Y	-0.02
5Y	-0.03
6Y	-0.04
7Y	-0.03
8Y	-0.03
9Y	-0.02
10Y	-0.01
11Y	0.00
12Y	0.01
13Y	0.02
14Y	0.03
15Y	0.04

for euro adoption. This matrix is similar to that for classic VAR(1) model in table 6.1. Main similarity consists in the fact that $\widehat{A}_1$ is not diagonally dominant nor estimated parameters are significant. Adding intercept and exogenous term signalling euro adoption seems to be good step because of their significance especially for equations describing short maturities. Estimated matrix $\widehat{C}$ can be seen in table 6.4. However in overall we might also suffer from overfitting as we have too many parameters to estimate.

Table 6.3: Matrix $\widehat{A}_1$ of estimated parameters from VAR(1) model with exogenous variable signalling euro adoption

τ	1Y	2Y	3Y	4Y	5Y	6Y	7Y	8Y	9Y	10Y	11Y	12Y	13Y	14Y	15Y
1Y	0.90	0.26	-0.62	1.11	-0.74	-0.07	-0.18	0.41	0.16	1.22	-2.71	0.51	1.04	0.02	-0.31
2Y	0.05	1.38	-1.90	2.06	0.65	-1.05	-1.67	1.55	-0.45	1.44	-1.52	0.37	0.59	-0.79	0.28
3Y	0.06	0.14	0.12	1.09	-0.37	0.57	-1.26	0.60	-0.28	0.47	-0.32	-0.02	0.68	-0.40	-0.09
4Y	0.10	-0.49	1.13	0.18	-0.92	1.81	-0.67	-0.21	-0.02	-0.41	0.32	-0.22	0.70	0.29	-0.60
5Y	0.11	-0.67	1.39	-0.33	-0.98	2.01	-0.19	-0.39	0.15	-0.78	0.50	-0.28	0.77	0.33	-0.64
6Y	0.10	-0.65	1.31	-0.52	-0.84	1.79	-0.21	-0.20	0.53	-0.96	0.53	-0.17	0.55	0.13	-0.40
7Y	0.08	-0.52	0.95	-0.29	-0.66	1.20	-0.40	0.27	0.63	-0.69	0.47	-0.12	0.35	-0.26	-0.02
8Y	0.06	-0.35	0.50	-0.01	-0.27	0.68	-1.02	0.69	0.90	-0.37	0.36	0.10	-0.09	-0.55	0.37
9Y	0.04	-0.18	0.10	0.21	0.11	0.13	-1.30	0.95	0.91	-0.13	0.39	0.23	-0.27	-0.85	0.67
10Y	0.02	0.01	-0.41	0.58	0.36	-0.20	-1.58	0.98	0.86	0.24	0.40	0.27	-0.58	-0.77	0.81
11Y	0.00	0.18	-0.75	0.72	0.57	-0.23	-1.78	0.92	0.67	0.35	0.63	0.32	-0.75	-0.76	0.94
12Y	-0.02	0.33	-1.05	0.84	0.70	-0.32	-1.66	0.73	0.49	0.48	0.48	0.39	-0.83	-0.47	0.90
13Y	-0.05	0.46	-1.23	0.72	0.92	-0.20	-1.67	0.59	0.41	0.32	0.46	0.42	-0.89	-0.10	0.84
14Y	-0.06	0.53	-1.25	0.56	0.82	0.21	-1.48	0.21	0.23	0.38	0.48	0.10	-0.80	0.25	0.83
15Y	-0.08	0.62	-1.30	0.42	0.80	0.46	-1.32	0.02	0.12	0.50	0.14	-0.11	-0.71	0.60	0.86

Table 6.4: Matrix $\widehat{C}$ of estimated intercept and explanatory time series for euro adoption parameters from VAR(1) model with exogenous variable signalling euro adoption

τ	Intercept	euro adoption
1Y	-0.04	0.05
2Y	0.05	-0.04
3Y	0.06	-0.07
4Y	0.05	-0.08
5Y	0.03	-0.07
6Y	0.01	-0.05
7Y	0.00	-0.03
8Y	-0.01	-0.01
9Y	-0.02	0.00
10Y	-0.02	0.01
11Y	-0.02	0.02
12Y	-0.01	0.02
13Y	0.00	0.02
14Y	0.02	0.01
15Y	0.05	-0.01

PREDICTIONS

In this chapter we evaluate fitted models by looking at the quality of their forecasts. First model we used in chapter 2 consisted of finding suitable ARIMA(p,i,q) process to model yields of each maturity separatelly. Since the data for maturities 1-10 are available from 7.1.2003 but data for maturities 11-15 are available only from 11.5.2006, in order to ensure that input data are the same for each model so there is no assymetric information present, we trained new ARIMA processes for maturities 1-10 with starting point of 11.5.2006. However the orders (p,i,q) were very close or even the same as for the processes for longer time period. ARIMA models for maturities 11-15 are the same as in chapter 2 because of identical time window. Fitted models together with the whole procedure of finding suitable models can be found in the results of code in the Attachments.

To evaluate the quality of predictions we use sum of squared errors defined as follows:

$$SSE_{\tau} = \sum_{i=1}^{N} (y_{t_i,\tau} - \widehat{y}_{t_i,\tau})^2,$$

where $\tau \in \{1, 2, ..., 15\}$ represents maturity of yield, N is number of observations in predicted period, $y_{t_i,\tau}$ stands for true value of yield and $\widehat{y}_{t_i,\tau}$ is predicted value of yield.

In table 7.1 you can find sum of SSE_{τ} for each considered model and for $N = 10, 20, ..., 70$. We have 6 different models each with two possible approaches to ARIMA time series modelling. The first one consisting of series of steps further explained in chapter 2 resulting in certain ARIMA(p,i,q) process with desired properties. Using the second approach, we model each time series only by simple AR(1) process without any differentiations. In addition to these models, we specified another five models that either combine some features of already mentioned 6 models or are not based on ARIMA modelling but on VAR modelling.

Similar information as in table 7.1 can be found also in figure 7.1 where on the horizontal axis we have different choices of N and on the vertical axis there is cumulative sum of errors for each model. In this case, the arbitrarily chosen ARIMA(p,i,q) models are considered. In figure 7.2 you can find the same information as in 7.1 but for time series modelled with AR(1) processes. Cumulative sum of errors is plotted also for the last 5 models in figure 7.3.

Table 7.1: Cumulative errors for every 10th observation, where DL with $\lambda = 0.7173231$ represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$. DL with $\lambda = 0.3504465$ represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time. PCA model represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC. Univariate model represents approach where we model each 1-dimensional time series separatelly. Univariate dummy reg represents regression model with ARMA errors where the external time series regressors is dummy variable having values 0 and 1. Univariate linear reg represents similar approach but with explanatory time series of zeros for time period before year 2008, linearly increasing during years 2008 and 2009 and finally stays on value 1 for the time period after year 2009. Optimized ARIMA stands for the approach where we have chosen orders (p,i,q) of ARIMA process. AR(1) stands for the approach where the time series is modelled by AR(1) process. VAR(1) means vector autoregression process of order $p = 1$ while VAR(1) with linear reg stands for the vector autoregression model of order $p = 1$ and exogenous non-dummy time series signalling euro adoption.

Number of steps ahead considered	10	20	30	40	50	60	70
DL with $\lambda = 0.7173231$ and optimized ARIMA	4.68	15.21	36.27	57.52	77.94	105.69	145.96
DL with $\lambda = 0.7173231$ and AR(1)	3.75	9.37	18.74	33.71	55.24	80.94	109.43
DL with $\lambda = 0.3504465$ and optimized ARIMA	2.31	12.01	33.40	54.74	74.89	103.26	145.20
DL with $\lambda = 0.3504465$ and AR(1)	6.05	38.05	111.65	261.33	495.32	792.45	1137.81
PCA with optimized ARIMA	1.39	7.61	22.93	38.01	51.74	71.93	103.30
PCA with AR(1)	1.39	7.52	21.89	34.96	45.71	60.75	83.82
Univariate optimized ARIMA	1.42	8.44	25.88	43.37	59.76	83.57	119.84
Univariate AR(1)	1.06	5.70	16.63	25.74	32.20	41.18	55.20
Univariate dummy reg with optimized ARIMA	1.23	7.06	21.48	35.16	46.94	63.82	89.66
Univariate dummy reg with AR(1)	1.06	5.67	16.53	25.54	31.89	40.73	54.55
Univariate linear reg with optimized ARIMA	1.24	7.10	21.56	35.29	47.10	64.03	89.95
Univariate linear reg with AR(1)	0.78	3.91	11.18	16.86	20.65	25.23	31.96
DL with $\lambda = 0.7173231$, AR(1) and linear reg	3.79	9.66	19.70	36.43	61.05	91.12	125.06
DL with $\lambda = 0.3504465$, opt. ARIMA and lin. reg	2.27	11.74	32.61	53.28	72.60	99.79	140.02
PCA with AR(1) and linear reg	1.38	7.45	21.66	34.51	45.00	59.63	82.12
VAR(1)	1.77	9.88	28.00	45.16	60.23	81.33	112.93
VAR(1) with linear reg	1.65	8.92	24.13	37.53	47.90	61.30	80.85

Very interesting observation is that besides Diebold-Li model with λ chosen as average, all other models show better forecasting power in the case of simple AR(1) processes. This may be caused by various factors but we assume that the main one is that in the training dataset, we have data for 2625 days. During this quite long period, several shocks occured and changed behaviour of time series. Modelling all these changes with autoregressive approach and limited number of AR and MA terms is very hard task. As we can see, probably better approach is to model only the main autoregressive trend with only one autoregressive term because adding more terms might just model some terminal parts of history with no impact on distant future.

Concerning different models, we see that the worst approach seems to be the Diebold-Li model with both choices of λ, especially for $\lambda = 0.3504465$ and AR(1) process. This may

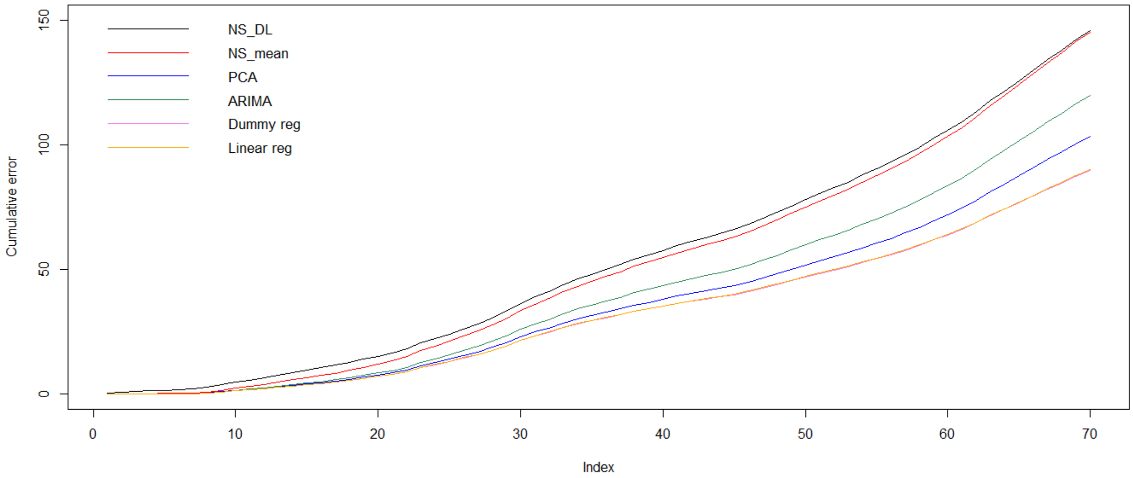

Figure 7.1: Cumulative errors of different approaches modelled by optimized ARIMA processes, where NS_DL in black represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$. NS_mean in red color represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time. PCA in blue represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC. ARIMA in seagreen represents approach where we model each 1-dimensional time series separatelly. Dummy reg in violet represents regression model with ARMA errors where the external time series regressors is dummy variable having values 0 and 1. Linear reg in orange represents similar approach but with explanatory time series of zeros for time period before year 2008, linearly increasing during years 2008 and 2009 and finally stays on value 1 for the time period after year 2009.

be caused by incapability of Nelson-Siegel framework to capture the shape of Slovak yield curve with parameter λ constant in time. In the case of AR(1) processes, the PCA approach provided better results than Diebold-Li but worse than univariate time series approaches. Advantage of PCA over Diebold-Li is that it does not impose functional form on the yield curve but it tries to capture main directions of variance and then linearly transformes the original data into this space. The reason why PCA got weaker results than univariate time series may be our choice of number of principal components taken into account. If we took more principal components than 3, the results might be better. Another result we can see here is that the reduction of dimensions was not really succesful in terms of elimination random noises present in 1-dimensional data and capturing only main movements. This means that the 15-dimensional time series is more complex with some behaviour specific for each maturity. Interesting comparison is between the univariate time series approach and approach consisting of regression with ARMA errors. We can see that only by adding one external regressor, we can improve the quality of predictions. However we can not

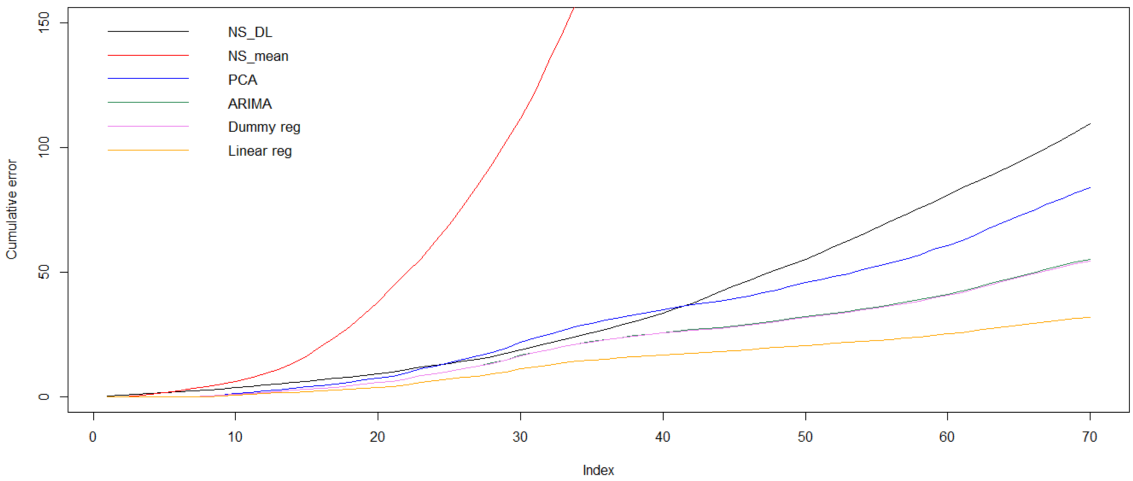

Figure 7.2: Cumulative errors of different approaches modelled by AR(1) processes, where NS_DL in black represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$. NS_mean in red color represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time. PCA in blue represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC. ARIMA in seagreen represents approach where we model each 1-dimensional time series separatelly. Dummy reg in violet represents regression model with ARMA errors where the external time series regressors is dummy variable having values 0 and 1. Linear reg in orange represents similar approach but with explanatory time series of zeros for time period before year 2008, linearly increasing during years 2008 and 2009 and finally stays on value 1 for the time period after year 2009.

forget the fact that if we want to use explanatory time series, we have to predict them as well, which can bring us another source of forecasting errors. Fortunately in this case it is quite simple to predict whether euro will be the currency in Slovakia for the upcoming time period. As we can see in the case of ARIMA(p,i,q) approach there is very little difference between our two proposed time series. However in the case of AR(1) modelling, approach with linear change between zero and one during the period around euro adoption performs the best. It is actually the best model according to sum of SSE_τ. When we compare VAR(1) model with univariate AR(1) models for each time series, we can see that the latter perfoms better. This supports our hypothesis of overfitting VAR(1) model because of large number of parameters. Both approaches get better results after introducing exogenous euro adoption time series but also in this case the AR(1) models deliver lower sum of SSE_τ. Adding explanatory time series when modelling slope factors or second principal component does not significantly change results of concerned models.

In table 7.2 we can take a look from another perspective onto SSE_τ of different models.

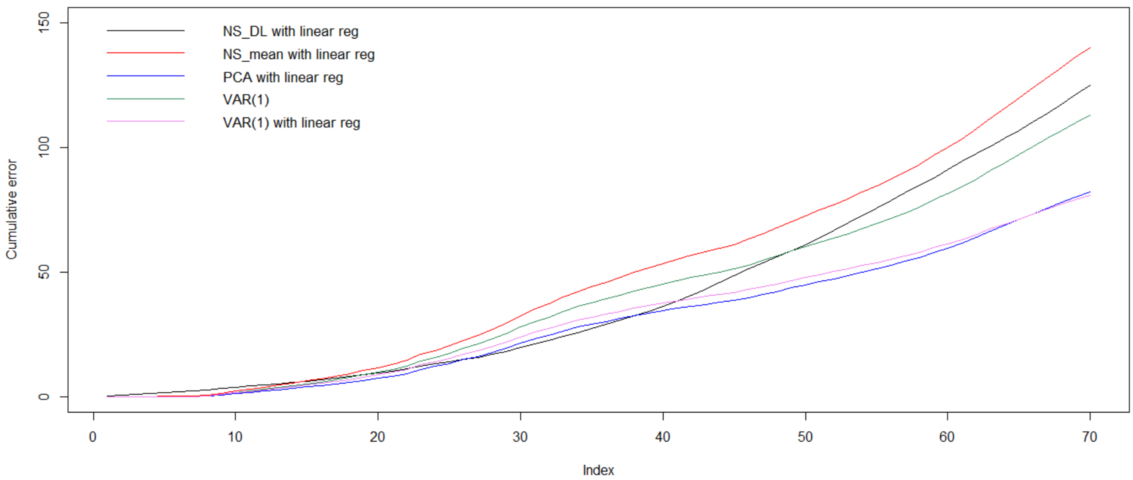

Figure 7.3: Cumulative errors of different approaches modelled by different models, where NS_DL with linear reg in black represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$, factors modelled in time by AR(1) processes with the extension of exogenous time series for euro adoption in case of the slope factor. NS_mean with linear reg in red color represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time, factors modelled by optimized ARIMA processes with the extension of exogenous time series for euro adoption in case of the slope factor. PCA with linear reg in blue represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC modelled by AR(1) processes in time with addition of an exogenous time series for euro adoption in case of second PC. VAR(1) in sea-green represents vector autoregression model with order $p = 1$. VAR(1) with linear reg in violet represents vector autoregression model with order $p = 1$ and exogenous time series signalling euro adoption.

We have SSE_τ for each model and each method of ARIMA modelling for $N = 70$ and for $\tau = 1, 3, ..., 15$. This can provide us with insight on which maturities affected the sum of SSE_τ of each model. We can see that in overall, the optimized ARIMA processes were able to predict movement of yields with short maturities approximately just as well or even better than their AR(1) counterparts. Where AR(1) processes achieved better results were long maturities. The reason why after summing all the SSE_τ the AR(1) processes had significantly better results may be that the errors in case of long maturities were in overall higher than in case of short maturities so the long maturities had higher impact on the total sum of SSE_τ resulting in weaker overall results of optimized ARIMA processes. To evaluate impact of this observation, we should use some evaluation methods that are more robust. One such approach is to take median value of SSE_τ for each model instead of their sum. However we also define our own evaluation method, further described in 7.1. Great advantage of this method is its robustness to increase of absolute values of yields in case of

Table 7.2: Cumulative errors for every second maturity starting with maturity of 1 year, where DL with $\lambda = 0.7173231$ represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$. DL with $\lambda = 0.3504465$ represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time. PCA model represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC. Univariate model represents approach where we model each 1-dimensional time series separatelly. Univariate dummy reg represents regression model with ARMA errors where the external time series regressors is dummy variable having values 0 and 1. Univariate linear reg represents similar approach but with explanatory time series of zeros for time period before year 2008, linearly increasing during years 2008 and 2009 and finally stays on value 1 for the time period after year 2009. Optimized ARIMA stands for the approach where we have chosen orders (p,i,q) of ARIMA process. AR(1) stands for the approach where the time series is modelled by AR(1) process. VAR(1) means vector autoregression process of order $p = 1$ while VAR(1) with linear reg stands for the vector autoregression model of order $p = 1$ and exogenous non-dummy time series signalling euro adoption.

Maturity	1	3	5	7	9	11	13	15
DL with $\lambda = 0.7173231$ and optimized ARIMA	0.8	2.1	0.3	1.7	6.6	14.4	22.9	30.7
DL with $\lambda = 0.7173231$ and AR(1)	6.9	11.6	19.7	10.6	2.7	0.6	2.0	4.9
DL with $\lambda = 0.3504465$ and optimized ARIMA	0.8	0.2	1.1	7.1	14.1	17.8	18.3	17.3
DL with $\lambda = 0.3504465$ and AR(1)	61.3	124.4	125.0	89.3	63.2	49.1	41.8	38.0
PCA with optimized ARIMA	0.5	0.5	0.3	4.8	10.4	13.0	13.1	12.3
PCA with AR(1)	1.3	0.3	0.3	4.1	8.7	10.5	10.3	9.3
Univariate optimized ARIMA	1.0	0.1	1.2	5.6	10.1	13.7	15.5	17.6
Univariate AR(1)	0.5	1.3	0.2	2.1	4.5	6.7	7.3	6.8
Univariate dummy reg with optimized ARIMA	0.5	0.6	0.3	3.3	7.2	10.5	12.2	14.0
Univariate dummy reg with AR(1)	0.5	1.3	0.2	2.2	4.4	6.5	7.1	6.7
Univariate linear reg with optimized ARIMA	0.5	0.5	0.3	3.3	7.2	10.5	12.2	14.0
Univariate linear reg with AR(1)	0.7	2.3	1.1	0.4	1.7	3.4	4.1	3.4
DL with $\lambda = 0.7173231$, AR(1) and linear reg	10.4	14.0	21.8	11.7	3.1	0.6	1.9	4.7
DL with $\lambda = 0.3504465$, opt. ARIMA and lin. reg	0.6	0.2	0.9	6.7	13.6	17.3	17.9	16.9
PCA with AR(1) and linear reg	1.6	0.2	0.3	4.0	8.5	10.2	10.0	9.0
VAR(1)	1.1	0.1	0.7	5.9	11.6	13.9	13.8	12.9
VAR(1) with linear reg	1.7	1.4	0.2	2.6	7.6	10.3	10.5	9.3

longer maturities which causes also bigger errors.

Definition 7.1 (Sum of ranked errors)**.** *We define sum of ranked errors*[1] *as value, assigned to a forecasting method, describing predictive power of that method. The computation of SRE is done in following steps.*

(1) Compute SSE_τ for each prediction method and $\tau \in \{1, 2, ..., 15\}$.

(2) For each τ, arrange SSE_τ in ascending order.

(3) Add rank to each predictive method and fixed τ starting with 1,2,...,15. This way, method with the smallest SSE_τ gets rank 1 and so on.

(4) If a tie[2] *occurs, add average rank to each of concerning models.*

[1] further described with abbreviation SRE

[2] Tie is defined as specific case where $SSE_{\tau,i} = SSE_{\tau,j}$, where i and j represents different predictive meth-

(5) *For each method, compute sum of their ranks. We describe this number as SRE, sum of ranked errors.*

Perhaps better insight is provided when we divide SRE by number of maturities, which is in our case equal to 15, obtaining average rank per method. All four mentioned evaluation methods with $N = 70$ can be seen in table 7.3.

Table 7.3: Cumulative errors for every second maturity starting with maturity of 1 year, where DL with $\lambda = 0.7173231$ represents Diebold-Li approach with the value of λ chosen as argmax of the third factor loading for $\tau = 2.5$. DL with $\lambda = 0.3504465$ represents Diebold-Li approach with value of λ chosen as the average value of non-dependent fits of Nelson-Siegel models for each point of time. PCA model represents approach with transformation of 15-dimensional time series of yields into 3-dimensional base given by first three PC. Univariate model represents approach where we model each 1-dimensional time series separatelly. Univariate dummy reg represents regression model with ARMA errors where the external time series regressors is dummy variable having values 0 and 1. Univariate linear reg represents similar approach but with explanatory time series of zeros for time period before year 2008, linearly increasing during years 2008 and 2009 and finally stays on value 1 for the time period after year 2009. Optimized ARIMA stands for the approach where we have chosen orders (p,i,q) of ARIMA process. AR(1) stands for the approach where the time series is modelled by AR(1) process. VAR(1) means vector autoregression process of order $p = 1$ while VAR(1) with linear reg stands for the vector autoregression model of order $p = 1$ and exogenous non-dummy time series signalling euro adoption.

Evaluation method	$\sum_{\tau=1}^{15} SSE_\tau$	$median(SSE_\tau)$	SRE	ARE
DL with $\lambda = 0.7173231$ and optimized ARIMA	145.96	5.79	154	10.27
DL with $\lambda = 0.7173231$ and AR(1)	109.43	4.95	125	8.33
DL with $\lambda = 0.3504465$ and optimized ARIMA	145.20	10.9	198	13.20
DL with $\lambda = 0.3504465$ and AR(1)	1137.81	63.18	255	17.00
PCA with optimized ARIMA	103.30	7.86	140	9.33
PCA with AR(1)	83.82	6.64	118	7.87
Univariate optimized ARIMA	119.84	8.12	158	10.53
Univariate AR(1)	55.20	3.38	80	5.33
Univariate dummy reg with optimized ARIMA	89.66	5.4	110	7.33
Univariate dummy reg with AR(1)	54.55	3.39	79	5.27
Univariate linear reg with optimized ARIMA	89.95	5.42	117	7.80
Univariate linear reg with AR(1)	31.96	2.2	78	5.20
DL with $\lambda = 0.7173231$, AR(1) and linear reg	125.06	6.43	134	8.93
DL with $\lambda = 0.3504465$, opt. ARIMA and lin. reg	140.02	10.41	175	11.67
PCA with AR(1) and linear reg	82.12	6.54	110	7.33
VAR(1)	112.93	9.07	152	10.13
VAR(1) with linear reg	80.85	5.22	112	7.47

We can see that quality of predictions evaluated by more robust techniques is similar to previously used sum of SSE_τ. This observation supports our previously stated arguments and denies the hypothesis about abnormal impact of higher errors in case of long maturities. However we can see that the smallest ARE is higher than 5 so we can not say that one particular method really dominates the others. It means that if we combine sev-

ods.

eral prediction methods in particular way, we could obtain substantially better forecasting power. However this is just a theoretical option since we have no evident indication of how to combine the best models according to our evaluation methods.

Conclusion

In our work we introduced several models all based on autoregression theory in order to find the best way to predict future movements of yield curve. We can say that our main goal was achieved since we have found a suitable way to include the information of structural break caused by the euro adoption. In overall models that included this exogenous time series performed better regarding quality of forecasting. Perhaps surprising observation was better predictive power of simple AR(1) processes than of optimized arbitrarily chosen processes ARIMA(p,i,q). We contributed the evaluation methodology by introducing our own robust evaluation metrics SRE and ARE. Another interesting finding of our thesis was the one-to-one similarity of factors from Diebold-Li model and first three principal components. All variations of Diebold-Li approach disappointed comparing their predictive power to other methods. This is probably caused by incapability of Nelson-Siegel framework to describe whole yield curve. However this property defines yield to maturity for any given τ while every other modelling technique can model only given 15 points of actual yield curve. If we need to estimate some other yield besides these 15 points, question arises how to perform this task. One idea could be to make simple linear interpolation which should work for long maturities where there is no significant difference between two yield values. However for short maturities this method could cause significant deviations. For τ higher than 15 or lower than 1 this method even can not calculate any value. In these cases well known Nelson-Siegel functional form can be fitted to solve this problem but as we already mentioned, Nelson-Siegel framework may not be capable of capturing the form of yield curve to satisfying extent. In this case we propose Nelson-Siegel-Svensson framework that is similar to Nelson-Siegel's but with additional curvature factor. It defines yield curve in following functional form.

$$y(\tau) = \beta_1 + \beta_2\Big(\frac{1 - e^{-\lambda_1\tau}}{\lambda_1\tau}\Big) + \beta_3\Big(\frac{1 - e^{-\lambda_1\tau}}{\lambda_1\tau} - e^{-\lambda_1\tau}\Big) + \beta_4\Big(\frac{1 - e^{-\lambda_2\tau}}{\lambda_2\tau} - e^{-\lambda_2\tau}\Big)$$

This way we eliminate the great disadvantage of modelling only some points of yield curve and in the end, we get yield to maturity for any desired τ.

One of further improvements of our work could be also based on Diebold-Li idea to model factors of a term structure model in time. Nelson-Siegel-Svensson is such model that could better fit on Slovak yield curve but in this case, question of how to fix parameter λ would be even harder because we have two of these parameters present here. Another option is to use Bayesian techniques such as Kalman filter to model state-space system in time.

Another improvement might be derived from our observation of significance of explanatory time series for euro adoption by introducing more such time series capturing main movements of yield curve. However by this approach we relocate part of predictive uncertainty to another process. Essential property of such exogenous process thus should be its easy predictability to same extent we want to predict movement of yield curve. To find such time series unfortunately is not that easy so suitability of this approach is questionable.

When we know that underlying process is unstable because of several shocks or regime shifts, an interesting idea is to take only the last portion of time series that is enough stable. We tried this idea taking only ARIMA processes fitted on time period after euro adoption but with weaker results than in case of taking whole time period with explanatory time series for euro adoption. This result emphasis the importance of use of every data available because it has some influence on the process even in distant future.

Other improvement could be done also in case of evaluation of predictions from different models. We could use stepwise evaluation where we at first specify length of time period T we want our models to forecast. Then we separate the data into test set containing $T \times n$ observations where T is length of predicted time period and n is the number of times we want to evaluate quality of predictions. In the first step we train considered models on remaining data and calculate predictions on time period of length T. Computation of specific evaluation metrics follows and is saved for each model. In next step we add previously predicted values into training set and fit parameters of models on these data. Another computation of evaluation metrics follows and this procedure is repeated together n times. Final examination of results is then derived from all of these partial results. Advantage of such procedure is elimination of random noise effects which would implicate higher credibility of our conclusions.

Bibliography

[1] Bartlett, P.: *Introduction to Time Series Analysis. Lecture 3.*, available on the internet (26.12.2016): *http://www.stat.berkeley.edu/ bartlett/courses/153-fall2010/lectures/3.pdf*

[2] Charles, J., Geyer, C., J.: *Principal Components Theory Notes*, available on the internet (26.12.2016): *http://www.stat.umn.edu/geyer/5601/notes/spect.pdf*

[3] Charpentier A.: *Unit root tests in R*, available on the internet (26.12.2016): *https://www.r-bloggers.com/unit-root-tests/*

[4] Diebold, F., X., Li, C.: *Forecasting the term structure of government bond yields*, Journal of Econometrics 130 (2006) 337-364, available on the internet (26.12.2016): *http://www.ssc.upenn.edu/ fdiebold/papers/paper49/Diebold-Li.pdf*

[5] Hyndman, R., J., Athanasopoulos, G.: *Forecasting: principles and practice*, available on the internet (26.12.2016): *https://www.otexts.org/fpp/9/1*

[6] MacKinnon, J. G.: *Critical Values for Cointegration Tests, Department Working Paper No. 1227, 2010*, Queen's University, available on the internet (26.12.2016): *http://qed.econ.queensu.ca/working_ papers/papers/qed_ wp_ 1227.pdf*

[7] Phillips, P., Perron, P.: *Testing for a unit root in time series regression*, Biometrika, 1988, available on the internet (26.12.2016): *http://www.ssc.wisc.edu/ bhansen/718/PhillipsPerron1988.pdf*

[8] Stehlikova, B.: *Introduction to time series*, lectures FMFI UK (2016), available on the internet (26.12.2016): *http://www.iam.fmph.uniba.sk/institute/stehlikova/cr16/prednasky/01-uvod.pdf*

[9] Stehlikova, B.: *Unit root*, lectures FMFI UK (2016), available on the internet (26.12.2016): *http://www.iam.fmph.uniba.sk/institute/stehlikova/cr16/prednasky/05-unit-root.pdf*

[10] Stehlikova, B.: *Spectral analysis*, lectures FMFI UK (2016), available on the internet (26.12.2016): *http://www.iam.fmph.uniba.sk/institute/stehlikova/cr16/prednasky/09_spektrum.pdf*

[11] *Unit Root Tests*, available on the internet (26.12.2016): *https://faculty.washington.edu/ezivot/econ584/notes/unitroot.pdf*

[12] Van den End, J., W.: *Statistical evidence on the mean reversion of interest rates*, De Nederlandsche Bank NV, Working Paper No. 284 (2011), available on the internet (26.12.2016): *https://www.dnb.nl/en/binaries/working%20paper%20284_tcm47-252978.pdf*

[13] Westfalishe Wilhelms-Universitat Munster: *Partial autocorrelation function*, available on the internet (26.12.2016): *http://www.wiwi.uni-muenster.de/05/download/studium/timeseries/WS1415/chapter_ 5.pdf*

[14] Zapata, H., O., Maradiaga, D., I., Pujula, A., L., Dicks, M., R.: *Recent Developments in Unit Root Tests and Historical Crop Yields*, available on the internet (26.12.2016): *http://ageconsearch.umn.edu/bitstream/103871/1/Zapata%20et%20al%20_ Recent%20Developments%20in%20Unit%20Root_ AAEA2011.pdf*

Attachments

Codes used for time series analysis

In this section we attach codes used for our analysis. All of them are written in R 3.3.1 and should be run separately if not stated otherwise.

Descriptive statistics

```
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")

#Creating time series of yields for whole time period
yields <- mydata[1:3461,1:18]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(min(yields$YYYY),st))

#Creating time series of yields for time period before Euro adoption
mydata_aea = mydata[mydata$YYYY > 2009,4:18][1:1716,]
yields_aea <- ts(mydata_aea, frequency=252, start=c(2010,1,1))

#Creating time series of yields for time period after Euro adoption
yields_bea <- mydata[mydata$YYYY < 2008,1:18]
frequencies <- rep(0,length(unique(yields_bea$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields_bea$YYYY[which(yields_bea$YYYY==(unique(yields_bea$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields_bea <- ts(yields_bea[,4:18], frequency=freq, start=c(min(yields_bea$YYYY),st))

#Histograms for each maturity
par(mfrow=c(3,5))

for(i in 1:15){
        hist(yields[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="")
        }

#Descriptive statistics for whole time period
round(c(apply(yields[,1:10],FUN=min,2),apply(na.omit(yields[,11:15]),FUN=min,2)),2)
round(c(apply(yields[,1:10],FUN=max,2),apply(na.omit(yields[,11:15]),FUN=max,2)),2)
```

```
round(c(apply(yields[,1:10],FUN=mean,2),apply(na.omit(yields[,11:15]),FUN=mean,2)),2)
round(c(apply(yields[,1:10],FUN=sd,2),apply(na.omit(yields[,11:15]),FUN=sd,2)),2)


#Comparing data before Euro adoption and after Euro adoption
#Histograms for each maturity
par(mfrow=c(2,5))

for(i in 1:5){
        hist(yields_bea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="Before Euro adoption")
        }

for(i in 1:5){
        hist(yields_aea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="After Euro adoption")
        }

for(i in 6:10){
        hist(yields_bea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="Before Euro adoption")
        }

for(i in 6:10){
        hist(yields_aea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="After Euro adoption")
        }

for(i in 11:15){
        hist(yields_bea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="Before Euro adoption")
        }

for(i in 11:15){
        hist(yields_aea[,i],main=paste("Yields from ZCY", i, "Y", sep=""),xlab="After Euro adoption")
        }

#Descriptive statistics for time periods before and after Euro adoption
round(c(apply(yields_bea[,1:10],FUN=min,2),apply(na.omit(yields_bea[,11:15]),FUN=min,2)),2)
round(apply(yields_aea,FUN=min,2),2)
round(c(apply(yields_bea[,1:10],FUN=max,2),apply(na.omit(yields_bea[,11:15]),FUN=max,2)),2)
round(apply(yields_aea,FUN=max,2),2)
round(c(apply(yields_bea[,1:10],FUN=mean,2),apply(na.omit(yields_bea[,11:15]),FUN=mean,2)),2)
round(apply(yields_aea,FUN=mean,2),2)
round(c(apply(yields_bea[,1:10],FUN=sd,2),apply(na.omit(yields_bea[,11:15]),FUN=sd,2)),2)
round(apply(yields_aea,FUN=sd,2),2)
```

ARIMA

```
#set the path to the input file
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")[1:3461,]
attach(mydata)

#reading in libraries
library(astsa)
library(fArma)
library(WDI)
library(urca)

#creating time series from our dataset
yields <- mydata[1:3461,1:18]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(min(yields$YYYY),st))
```

```
#ZCY1Y
#is there unit root in the time series?
plot(yields[,1],type = "l",ylab="Yield")
#Adjusted Dickey-Fuller test
summary(ur.df(ZCY1Y,type="drift",lags=20,selectlags="BIC"))
summary(ur.df(ZCY1Y,type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(ZCY1Y,type="mu"))
summary(ur.kpss(ZCY1Y,type="tau"))
#Phillips-Perron test
PP.test(ZCY1Y)

#There is an unit root in the original time series, we need to take differences
plot(diff(ZCY1Y),type = "l")
mean(diff(ZCY1Y))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(ZCY1Y),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(ZCY1Y),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(ZCY1Y))

#we work with first differences of the original time series

acf2(diff(ZCY1Y))
#We choose ARIMA(3,1,0) process
#It is impossible to find really good model for this time series,
#because there is significant autocorrelation for quite large lags(more than 20)
sarima(ZCY1Y,3,1,0,details = FALSE)

#testing stationarity of the process
armaRoots(c(-0.4334, -0.1907, -0.0661))
#process is stacionary

#spectral analysis
sp = spectrum(yields[,1], kernel("daniell"), log="no")
plot(sp$spec[1:20],type = "l",xlab="frequency",ylab="spectrum")
#we take weighted average of 4 frequencies as the peak value is not significantly bigger than other three values
weights = sp$spec[(which.max(sp$spec)-2):(which.max(sp$spec)+1)]
weights = weights/sum(weights)
frequency = t(weights)%*%sp$freq[(which.max(sp$spec)-2):(which.max(sp$spec)+1)]
period = 1/frequency
period
#period/length(mydata[YYYY==2004,1])
#period of approximately 4 years in 1 year ZC bond




#testing unit root for all maturities
p_values <- rep(100,15)
for(i in 1:15){
        p_values[i] <- PP.test(na.omit(yields[,i]))$p.value}
#in all cases p-value > 50% => we take differences and test them

p_values_diff <- rep(100,15)
for(i in 1:15){
        p_values_diff[i] <- PP.test(na.omit(diff(yields[,i])))$p.value}
#p-values = 0.01 => we work with 1st differences



#ZCY2Y
acf2(diff(ZCY2Y))
sarima(ZCY2Y,8,1,1,details = FALSE)
```

```
armaRoots(c(0.6822,  0.1689,  0.0613, 0.0448, 0.0207, −0.0598, −0.0005, 0.0438))
#process is stationary
armaRoots(c(−0.9501))
#process is invertible



#ZCY3Y
acf2(diff(ZCY3Y))
sarima(ZCY3Y,2,1,3,details = FALSE)

armaRoots(c(0.5384,−0.8043))
#process is stationary
armaRoots(c(0.8002,−0.9223,0.1615))
#process is invertible


#ZCY4Y
acf2(diff(ZCY4Y))
sarima(ZCY4Y,0,1,3,details = FALSE)

armaRoots(c(0.2657,0.0540,−0.0816))
#process is invertible


#ZCY5Y
acf2(diff(ZCY5Y))
sarima(ZCY5Y,0,1,3,details = FALSE)

armaRoots(c(0.2586,0.0750,−0.1045))
#process is invertible


#ZCY6Y
acf2(diff(ZCY6Y))
sarima(ZCY6Y,3,1,3,details = FALSE)

armaRoots(c(0.2614,0.5763,0.0785))
#process is stationary
armaRoots(c(0.5174,0.5759,−0.1820))
#process is invertible


#ZCY7Y
acf2(diff(ZCY7Y))
sarima(ZCY7Y,0,1,7,details = FALSE)

armaRoots(c(0.2578,0.0390,−0.0769,0.0200,−0.0268,−0.0018,−0.0560))
#process is invertible


#ZCY8Y
acf2(diff(ZCY8Y))
sarima(ZCY8Y,0,1,7,details = FALSE)

armaRoots(c(0.2678,0.0088,−0.0472,0.0121,−0.0207,−0.0066,−0.0572))
#process is invertible


#ZCY9Y
acf2(diff(ZCY9Y))
sarima(ZCY9Y,0,1,8,details = FALSE)

armaRoots(c(0.2823,−0.0123,−0.0184,0.0005,−0.0155,−0.0102,−0.0401,−0.0240))
#process is invertible
```

```
#ZCY10Y
acf2(diff(ZCY10Y))
sarima(ZCY10Y,0,1,9,details = FALSE)

armaRoots(c(0.3042,-0.0203,0.0091,-0.0129,-0.0122,-0.0131,-0.0197,-0.0397,0.0084))
#process is invertible


#ZCY11Y
acf2(diff(na.omit(ZCY11Y)))
sarima(na.omit(ZCY11Y),0,1,8,details = FALSE)

armaRoots(c(0.2938,-0.0233,-0.0008,-0.0117,-0.0024,0.0038,-0.0294,-0.0560))
#process is invertible


#ZCY12Y
acf2(diff(na.omit(ZCY12Y)))
sarima(na.omit(ZCY12Y),0,1,8,details = FALSE)

armaRoots(c(0.2940,-0.0209,0.0017,-0.0111,0.0001,0.0085,-0.0199,-0.0647))
#process is invertible


#ZCY13Y
acf2(diff(na.omit(ZCY13Y)))
sarima(na.omit(ZCY13Y),0,1,8,details = FALSE)

armaRoots(c(0.2974,-0.0114,-0.0001,-0.0047,0.0043,0.0114,-0.0144,-0.0636))
#process is invertible


#ZCY14Y
acf2(diff(na.omit(ZCY14Y)))
sarima(na.omit(ZCY14Y),0,1,1,details = FALSE)

armaRoots(c(0.3060))
#process is invertible


#ZCY15Y
acf2(diff(na.omit(ZCY15Y)))
sarima(na.omit(ZCY15Y),5,1,2,details = FALSE)

armaRoots(c(1.1847,-0.5909,-0.1642,-0.0926,-0.0257))
#process is stationary
armaRoots(c(1.5202,-0.9712))
#process is invertible



#spectral analysis for all maturities
period = rep(0,15)

for(i in 1:15){
        sp = spectrum(na.omit(yields[,i]), kernel("daniell"), log="no")
        frequency = sp$freq[which.max(sp$spec)]
        period[i] = 1/frequency
        }
#no significant periodic movements found for longer maturities
```

ARIMA applied to period before euro adoption

```
#set the path to the input file
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")
attach(mydata)
```

```
#reading in libraries
library(astsa)
library(fArma)
library(WDI)
library(urca)

yields <- mydata[1:2625,1:18]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(min(yields$YYYY),st))

#graph with label for Euro adoption
plot(yields[,1],type = "l",ylab="Yields in percentage",ylim=c(-0.5,6))
xtick<-2009
axis(side=1, at=xtick, labels = FALSE)
text(x=xtick, par("usr")[3],
     labels = "Euro adoption",col="red", pos = 1, xpd = TRUE, cex=0.85)
abline(v=2009, col="red",lty=2,lwd=2)

#creating time series of yields from time period before Euro adoption
yields_bea <- mydata[mydata$YYYY < 2008,1:18]
frequencies <- rep(0,length(unique(yields_bea$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields_bea$YYYY[which(yields_bea$YYYY==(unique(yields_bea$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields_bea <- ts(yields_bea[,4:18], frequency=freq, start=c(min(yields_bea$YYYY),st))


plot(yields_bea[,1],type = "l",ylab="Yields in percentage")

#yields of 1Y ZC Bond before year 2008 (euro adoption in 2009)
bea1y <- mydata[YYYY<2008,4]

#Adjusted Dickey-Fuller test
summary(ur.df(bea1y,type="drift",lags=20,selectlags="BIC"))
summary(ur.df(bea1y,type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(bea1y,type="mu"))
summary(ur.kpss(bea1y,type="tau"))
#Phillips-Perron test
PP.test(bea1y)


#Adjusted Dickey-Fuller test
summary(ur.df(diff(bea1y),type="none",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(diff(bea1y),type="mu"))
#Phillips-Perron test
PP.test(diff(bea1y))

acf2(diff(bea1y))

sarima(bea1y,5,1,3,details = FALSE)

armaRoots(c(-0.1670,-0.0895,0.6875,0.3155,0.1435))
#process is stationary

armaRoots(c(0.2357,-0.0625,0.6766))
#process is invertible
```

```
#testing unit root for all maturities
p_values <- rep(100,15)
for(i in 1:15){
        p_values[i] <- PP.test(na.omit(yields_bea[,i]))$p.value}
#in all cases p-value > 60% => we take differences and test them

p_values_diff <- rep(100,15)
for(i in 1:15){
        p_values_diff[i] <- PP.test(na.omit(diff(yields_bea[,i])))$p.value}
#p-values = 0.01 => we work with 1st differences



#ZCY2Y
acf2(diff(yields_bea[,2]))
sarima(yields_bea[,2],8,1,1,details = FALSE)

armaRoots(c(0.7573,0.1757,0.0626,0.0398,-0.0177,-0.1073,0.0732,-0.0133))
#process is stationary
armaRoots(c(-0.9495))
#process is invertible


#ZCY3Y
acf2(diff(yields_bea[,3]))
sarima(yields_bea[,3],5,1,7,details = FALSE)

armaRoots(c(0.9882, -1.253, 1.1358, -0.6662, 0.7414))
#process is stationary
armaRoots(c(1.1379, -1.4359, 1.3316, -0.9426, 0.9481, -0.1801, 0.0347))
#process is invertible


#ZCY4Y
acf2(diff(yields_bea[,4]))
sarima(yields_bea[,4],0,1,8,details = FALSE)

armaRoots(c(0.0792, -0.0027, -0.0642, -0.1368, 0.0108, 0.0531, -0.0655, -0.0795))
#process is invertible


#ZCY5Y
acf2(diff(yields_bea[,5]))
sarima(yields_bea[,5],7,1,1,details = FALSE)

armaRoots(c(0.3889, 0.0000, 0.0773, 0.1130, -0.0594, -0.0388, 0.0924))
#process is stationary
armaRoots(c(0.4220))
#process is invertible


#ZCY6Y
acf2(diff(yields_bea[,6]))
sarima(yields_bea[,6],1,1,8,details = FALSE)

armaRoots(c(0.1688))
#process is stationary
armaRoots(c(0.1945, -0.0166, -0.0701, -0.1320, 0.0320, 0.0376, -0.1081, -0.0488))
#process is invertible


#ZCY7Y
acf2(diff(yields_bea[,7]))
sarima(yields_bea[,7],8,1,0,details = FALSE)
```

```
armaRoots(c(-0.0454, 0.0539, 0.0569, 0.1285, -0.0030, -0.0305, 0.0764, 0.0501))
#process is stationary


#ZCY8Y
acf2(diff(yields_bea[,8]))
sarima(yields_bea[,8],7,1,1,details = FALSE)

armaRoots(c(0.4871, 0.1374, -0.0026, 0.0766, -0.0502, -0.0007, 0.0486))
#process is stationary
armaRoots(c(0.5925))
#process is invertible


#ZCY9Y
acf2(diff(yields_bea[,9]))
sarima(yields_bea[,9],4,1,1,details = FALSE)

armaRoots(c(0.5654, 0.1959, -0.0253, 0.0648))
#process is stationary
armaRoots(c(0.7732))
#process is invertible


#ZCY10Y
acf2(diff(yields_bea[,10]))
sarima(yields_bea[,10],8,1,0,details = FALSE)

armaRoots(c(-0.3113, -0.0533, -0.0559, 0.0426, 0.0467, 0.0674, 0.0070, 0.0306))
#process is stationary


#ZCY11Y
acf2(diff(na.omit(yields_bea[,11])))
sarima(na.omit(yields_bea[,11]),2,1,6,details = FALSE)

armaRoots(c(1.3806, -0.6886))
#process is stationary
armaRoots(c(1.6693, -1.1390, 0.2648, -0.2015, 0.3202, -0.2496))
#process is invertible


#ZCY12Y
acf2(diff(na.omit(yields_bea[,12])))
sarima(na.omit(yields_bea[,12]),2,1,6,details = FALSE)

armaRoots(c(1.3767, -0.6780))
#process is stationary
armaRoots(c(1.7027, -1.1787, 0.2935, -0.1984, 0.3022, -0.2339))
#process is invertible


#ZCY13Y
acf2(diff(na.omit(yields_bea[,13])))
sarima(na.omit(yields_bea[,13]),2,1,6,details = FALSE)

armaRoots(c(1.3677, -0.6616))
#process is stationary
armaRoots(c(1.7223, -1.2001, 0.3180, -0.2005, 0.2870, -0.2190))
#process is invertible


#ZCY14Y
acf2(diff(na.omit(yields_bea[,14])))
sarima(na.omit(yields_bea[,14]),2,1,6,details = FALSE)

armaRoots(c(1.3561, -0.6411))
#process is stationary
```

```
armaRoots(c(1.7346, -1.2090, 0.3321, -0.1944, 0.2658, -0.2024))
#process is invertible


#ZCY15Y
acf2(diff(na.omit(yields_bea[,15])))
sarima(na.omit(yields_bea[,15]),2,1,6,details = FALSE)

armaRoots(c(1.3355, -0.6182))
#process is stationary
armaRoots(c(1.7378, -1.2114, 0.3403, -0.1820, 0.2418, -0.1868))
#process is invertible



#spectral analysis for all maturities
period = rep(0,15)

for(i in 1:15){
        sp = spectrum(na.omit(yields_bea[,i]), kernel("daniell"), log="no")
        frequency = sp$freq[which.max(sp$spec)]
        period[i] = 1/frequency
        }
#(length of period = length of the whole observed time window) => no significant periodic movements found
```

ARIMA applied to period after euro adoption

```
#set the path to the input file
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")
mydata_aea = mydata[mydata$YYYY > 2009,4:18][1:1716,]


#reading in libraries
library(astsa)
library(fArma)
library(WDI)
library(urca)

#creating time series of yields from time period before Euro adoption
yields_aea <- ts(mydata_aea, frequency=252, start=c(2010,1,1))


#yields of 1Y ZC Bond after year 2009 (euro adoption in 2009)
plot(yields_aea[,1],type="l",ylab="Yields in percentage")
summary(ur.df(yields_aea[,1],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(yields_aea[,1],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(yields_aea[,1],type="mu"))
summary(ur.kpss(yields_aea[,1],type="tau"))
#Phillips-Perron test
PP.test(yields_aea[,1])

#testing unit root for all maturities
p_values <- rep(100,15)
for(i in 1:15){
        p_values[i] <- PP.test(na.omit(yields_aea[,i]))$p.value}
#in all cases p-value > 35% => we take differences and test them

p_values_diff <- rep(100,15)
for(i in 1:15){
        p_values_diff[i] <- PP.test(na.omit(diff(yields_aea[,i])))$p.value}
#p-values = 0.01 => we work with 1st differences

#ZCY1Y
acf2(diff(yields_aea[,1]))
sarima(yields_aea[,1],15,1,1,details = FALSE)
```

```
armaRoots(c(0.3433, 0.1507, 0.0983, 0.0950, −0.0546, 0.0523, 0.0165,
 0.0331, 0.0381, −0.0683, 0.0127, 0.0210, −0.0254, 0.0502, 0.0642 ))
#process is stationary
armaRoots(c(0.8899))
#process is invertible

#ZCY2Y
acf2(diff(yields_aea[,2]))
sarima(yields_aea[,2],1,1,8,details = FALSE)

armaRoots(c(−0.4713))
#process is stationary
armaRoots(c(−0.0835, 0.2319, 0.0414, 0.0068, −0.0308, 0.0291, 0.0384, −0.0589))
#process is invertible

#ZCY3Y
acf2(diff(yields_aea[,3]))
sarima(yields_aea[,3],0,1,8,details = FALSE)

armaRoots(c(0.4047, 0.0481, −0.0295, 0.0094, 0.0058, 0.0530, 0.0176, −0.0762))
#process is invertible

#ZCY4Y
acf2(diff(yields_aea[,4]))
sarima(yields_aea[,4],1,1,3,details = FALSE)

armaRoots(c(−0.4778))
#process is stationary
armaRoots(c(−0.0677, 0.2579, −0.0450))
#process is invertible

#ZCY5Y
acf2(diff(yields_aea[,5]))
sarima(yields_aea[,5],2,1,2,details = FALSE)

armaRoots(c(−0.6563, −0.1728))
#process is stationary
armaRoots(c(−0.2678, 0.1524))
#process is invertible

#ZCY6Y
acf2(diff(yields_aea[,6]))
sarima(yields_aea[,6],1,1,5,details = FALSE)

armaRoots(c(0.7160))
#process is stationary
armaRoots(c(1.0704, −0.1829, −0.1719, 0.1499, −0.0837))
#process is invertible

#ZCY7Y
acf2(diff(yields_aea[,7]))
sarima(yields_aea[,7],11,1,1,details = FALSE)

armaRoots(c(−1.1354, −0.4156, −0.0931, 0.0166, 0.0343, 0.0764,
  0.1160, 0.0942, 0.0405, 0.0842, 0.0849))
#process is stationary
armaRoots(c(−0.8129))
#process is invertible

#ZCY8Y
acf2(diff(yields_aea[,8]))
sarima(yields_aea[,8],10,1,0,details = FALSE)

armaRoots(c(−0.3063, −0.1156, 0.0243, 0.0135, 0.0414,
  0.0491, 0.0877, 0.0389, 0.0196, 0.0548))
#process is stationary
```

```
#ZCY9Y
acf2(diff(yields_aea[,9]))
sarima(yields_aea[,9],10,1,0,details = FALSE)

armaRoots(c(-0.2952, -0.0838, 0.0147, 0.0353, 0.0416,
  0.0480, 0.0843, 0.0493, 0.0268, 0.0417))
#process is stationary

#ZCY10Y
acf2(diff(yields_aea[,10]))
sarima(yields_aea[,10],10,1,0,details = FALSE)

armaRoots(c(-0.2843, -0.0592, 0.0104, 0.0539, 0.0442,
  0.0434, 0.0753, 0.0603, 0.0306, 0.0282))
#process is stationary

#ZCY11Y
acf2(diff(na.omit(yields_aea[,11])))
sarima(na.omit(yields_aea[,11]),8,1,1,details = FALSE)

armaRoots(c(0.1727, 0.0772, 0.0316, 0.0583, 0.0184, 0.0152, 0.0467, 0.0439))
#process is stationary
armaRoots(c(0.4431))
#process is invertible

#ZCY12Y
acf2(diff(na.omit(yields_aea[,12])))
sarima(na.omit(yields_aea[,12]),11,1,0,details = FALSE)

armaRoots(c(-0.2534, -0.0377, 0.0183, 0.0621, 0.0451,
  0.0259, 0.0489, 0.0779, 0.0242, 0.023, 0.0373))
#process is stationary

#ZCY13Y
acf2(diff(na.omit(yields_aea[,13])))
sarima(na.omit(yields_aea[,13]),0,1,8,details = FALSE)

armaRoots(c(0.2405, -0.0078, -0.0317, -0.0354, -0.0210, -0.0103, -0.0391, -0.0605))
#process is invertible

#ZCY14Y
acf2(diff(na.omit(yields_aea[,14])))
sarima(na.omit(yields_aea[,14]),14,1,0,details = FALSE)

armaRoots(c(-0.2396, -0.0672, 0.0318, 0.0294, 0.0234, 0.0118,
  0.0465, 0.0672, 0.0135, 0.0370, 0.0561, 0.0009, -0.0239, -0.0997))
#process is stationary

#ZCY15Y
acf2(diff(na.omit(yields_aea[,15])))
sarima(na.omit(yields_aea[,15]),14,1,0,details = FALSE)

armaRoots(c(-0.2583, -0.0952, 0.0165, 0.0072, 0.0002, -0.0030,
  0.0479, 0.0486, 0.0073, 0.0351, 0.0576, 0.0118, -0.0151, -0.1082))
#process is stationary



#spectral analysis for all maturities
period = rep(0,15)

for(i in 1:15){
        sp = spectrum(na.omit(yields_aea[,i]), kernel("daniell"), log="no")
        frequency = sp$freq[which.max(sp$spec)]
        period[i] = 1/frequency
        }
#(length of period = length of the whole observed time window) => no significant periodic movements found
```

Principal components analysis

```
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")
attach(mydata)

#creating time series from our dataset
yields <- mydata[1:3461,1:18]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(min(yields$YYYY),st))

#some graphs

#plotting time series of bonds of maturities 1-10 years
plot(yields[,1],type = "l",ylab="Yields",ylim=c(-0.7,6.2))
lines(yields[,2],type = "l", col = "red")
lines(yields[,3],type = "l", col = "blue")
lines(yields[,4],type = "l", col = "tan")
lines(yields[,5],type = "l", col = "yellow")
lines(yields[,6],type = "l", col = "springgreen")
lines(yields[,7],type = "l", col = "slateblue")
lines(yields[,8],type = "l", col = "violet")
lines(yields[,9],type = "l", col = "turquoise")
lines(yields[,10],type = "l", col = "sienna1")

legend('topright', names(mydata)[4:13] , lty=1,
 col=c('black', 'red', 'blue','tan','yellow','springgreen','slateblue','violet','turquoise','sienna1'),
 bty='n', cex=.75)


#plotting time series of bonds of maturities 11-15 years
plot(na.omit(yields[,11]),type = "l", col = "seagreen",ylab="Yields",ylim=c(-0.2,6.6))
lines(na.omit(yields[,12]),type = "l", col = "orangered")
lines(na.omit(yields[,13]),type = "l", col = "green")
lines(na.omit(yields[,14]),type = "l", col = "firebrick")
lines(na.omit(yields[,15]),type = "l", col = "orange")

legend('topright', names(mydata)[14:18] , lty=1,
 col=c('seagreen','orangered','green','firebrick','orange'),
 bty='n', cex=1)


#PCA for whole observed period
#computing sample variance matrix
yields_nao <- na.omit(mydata[,1:18])[1:2625,]
frequencies <- rep(0,length(unique(yields_nao$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields_nao$YYYY[which(yields_nao$YYYY==(unique(yields_nao$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields_nao <- ts(yields_nao[,4:18], frequency=freq, start=c(min(yields_nao$YYYY),st))

round(cor(yields_nao), digits = 2)
ZC_pca<-prcomp(yields_nao, scale=FALSE)
summary(ZC_pca)
ZC_pca

alpha <- rep(0, 15)
for(i in 1:15) alpha[i] <- sum((ZC_pca$sdev^2)[1:i])/sum(ZC_pca$sdev^2)
plot(alpha, type="b")
```

```
#transformation of original yield time series into 3-dimensional base defined by the first three PC
means <- t(replicate(dim(yields_nao)[1],apply(yields_nao,2,mean)))
tr_yields <- (yields_nao-means)%*%ZC_pca$rotation[,1:3]

tr_yields1 <- (yields_nao-means)%*%ZC_pca$rotation[,1]
tr_yields2 <- (yields_nao-means)%*%ZC_pca$rotation[,1:2]

#Backward transformation
yields_b1 <- tr_yields1%*%t(ZC_pca$rotation[,1]) + means
yields_b2 <- tr_yields2%*%t(ZC_pca$rotation[,1:2]) + means
yields_b3 <- tr_yields%*%t(ZC_pca$rotation[,1:3]) + means

yields_b1ts <- ts(yields_b1, frequency=freq, start=c(min(na.omit(mydata[,1:18])[1:2625,]$YYYY),st))
yields_b2ts <- ts(yields_b2, frequency=freq, start=c(min(na.omit(mydata[,1:18])[1:2625,]$YYYY),st))
yields_b3ts <- ts(yields_b3, frequency=freq, start=c(min(na.omit(mydata[,1:18])[1:2625,]$YYYY),st))

#Graphs with transformed yields with 1,2,3 principal components
plot(yields_nao[,1],type = "l",ylab="Yields",ylim=c(-1.2,5.3))
lines(yields_b1ts[,1], type = "l", col = "red")
lines(yields_b2ts[,1], type = "l", col = "blue")
lines(yields_b3ts[,1], type = "l", col = "green")

legend('topright', c("Original time series","1 principal component",
"2 principal components","3 principal components") , lty=1,
 col=c('black','red','blue','green'),
 bty='n', cex=0.9)


plot(yields_nao[,7],type = "l",ylab="Yields",ylim=c(-0.5,5.5))
lines(yields_b1ts[,7], type = "l", col = "red")
lines(yields_b2ts[,7], type = "l", col = "blue")
lines(yields_b3ts[,7], type = "l", col = "green")

legend('topright', c("Original time series","1 principal component",
"2 principal components","3 principal components") , lty=1,
 col=c('black','red','blue','green'),
 bty='n', cex=0.8)


plot(yields_nao[,13],type = "l",ylab="Yields",ylim=c(0.3,6.7))
lines(yields_b1ts[,13], type = "l", col = "red")
lines(yields_b2ts[,13], type = "l", col = "blue")
lines(yields_b3ts[,13], type = "l", col = "green")

legend('topright', c("Original time series","1 principal component",
"2 principal components","3 principal components") , lty=1,
 col=c('black','red','blue','green'),
 bty='n', cex=0.8)

#ARIMA modelling
library(astsa)
library(fArma)
library(WDI)
library(urca)

#Diebold-Li approach => AR(1) processes
sarima(tr_yields[,1],1,0,0,details = FALSE)
sarima(tr_yields[,2],1,0,0,details = FALSE)
sarima(tr_yields[,3],1,0,0,details = FALSE)

#Our approach => Finding the best ARIMA process

#is there unit root in the time series?
plot(tr_yields[,1],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(tr_yields[,1],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(tr_yields[,1],type="trend",lags=20,selectlags="BIC"))
```

```
#KPSS test
summary(ur.kpss(tr_yields[,1],type="mu"))
summary(ur.kpss(tr_yields[,1],type="tau"))
#Phillips-Perron test
PP.test(tr_yields[,1])

#There is an unit root in the original time series, we need to take differences
plot(diff(tr_yields[,1]),type = "l")
mean(diff(tr_yields[,1]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(tr_yields[,1]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(tr_yields[,1]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(tr_yields[,1]))

#we work with first differences of the original time series

acf2(diff(tr_yields[,1]))
sarima(tr_yields[,1],1,1,2,details = FALSE)

#testing stationarity of the process
armaRoots(c(0.9479))
#process is stationary
#testing invertibility of the process
armaRoots(c(1.1524, -0.2139))
#process is invertible

#is there unit root in the time series?
plot(tr_yields[,2],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(tr_yields[,2],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(tr_yields[,2],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(tr_yields[,2],type="mu"))
summary(ur.kpss(tr_yields[,2],type="tau"))
#Phillips-Perron test
PP.test(tr_yields[,2])

#There is an unit root in the original time series, we need to take differences
plot(diff(tr_yields[,2]),type = "l")
mean(diff(tr_yields[,2]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(tr_yields[,2]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(tr_yields[,2]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(tr_yields[,2]))

#we work with first differences of the original time series

acf2(diff(tr_yields[,2]))
sarima(tr_yields[,2],8,1,0,details = FALSE)

#testing stationarity of the process
armaRoots(c(-0.3454, -0.1256, -0.0794, -0.0425, -0.0244, -0.0350, -0.0137, 0.0625))
#process is stationary


#is there unit root in the time series?
plot(tr_yields[,3],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(tr_yields[,3],type="drift",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(tr_yields[,3],type="mu"))
#Phillips-Perron test
```

```
PP.test(tr_yields[,3])

#There is an unit root in the original time series, we need to take differences
plot(diff(tr_yields[,3]),type = "l")
mean(diff(tr_yields[,3]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(tr_yields[,3]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(tr_yields[,3]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(tr_yields[,3]))

#we work with first differences of the original time series

acf2(tr_yields[,3])
sarima(tr_yields[,3],1,0,6,details = FALSE)

#testing stationarity of the process
armaRoots(c(0.9952))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.5244, 0.119, −0.0704, 0.0692, 0.0537, −0.0233))
#process is invertible

acf2(diff(tr_yields[,3]))
sarima(tr_yields[,3],5,1,1,details = FALSE)

#testing stationarity of the process
armaRoots(c(0.2066, −0.0116, 0.0870, −0.0414, −0.0567 ))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.7326))
#process is invertible




#PCA for time period before Euro adoption
#creating time series of yields from time period before Euro adoption
yields_bea <− ts(mydata[YYYY < 2008,4:18], frequency=252, start=c(2003,1,7))

#computing sample variance matrix
round(cor(na.omit(yields_bea)), digits = 2)
ZC_BEA_pca<−prcomp(na.omit(yields_bea), scale=FALSE)
summary(ZC_BEA_pca)
ZC_BEA_pca

alpha <− rep(0, 15)
for(i in 1:15) alpha[i] <− sum((ZC_BEA_pca$sdev^2)[1:i])/sum(ZC_BEA_pca$sdev^2)
plot(alpha, type="b")


#PCA for time period after Euro adoption
#creating time series of yields from time period after Euro adoption
yields_aea <− ts(mydata[YYYY > 2009,4:18], frequency=252, start=c(2003,1,7))

#computing sample variance matrix
round(cor(na.omit(yields_aea)), digits = 2)
ZC_AEA_pca<−prcomp(na.omit(yields_aea), scale=FALSE)
summary(ZC_AEA_pca)
ZC_AEA_pca

alpha <− rep(0, 15)
for(i in 1:15) alpha[i] <− sum((ZC_AEA_pca$sdev^2)[1:i])/sum(ZC_AEA_pca$sdev^2)
plot(alpha, type="b")
```

ARIMA for forecasting

```
#set the path to the input file
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")

#reading in libraries
library(astsa)
library(fArma)
library(WDI)
library(urca)

#creating time series from our dataset
yields <- na.omit(mydata[,1:18])[1:2625,]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(2006,st))



#ZCY1Y
#is there unit root in the time series?
plot(yields[,1],type = "l",ylab="Yield")
#Adjusted Dickey-Fuller test
summary(ur.df(yields[,1],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(yields[,1],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(yields[,1],type="mu"))
summary(ur.kpss(yields[,1],type="tau"))
#Phillips-Perron test
PP.test(yields[,1])

#There is an unit root in the original time series, we need to take differences
plot(diff(yields[,1]),type = "l")
mean(diff(yields[,1]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(yields[,1]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(yields[,1]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(yields[,1]))

#we work with first differences of the original time series

acf2(diff(yields[,1]))
#We choose ARIMA(3,1,0) process
#It is impossible to find really good model for this time series,
#because there is significant autocorrelation for quite large lags(more than 20)
sarima(yields[,1],3,1,0,details = FALSE)

#testing stationarity of the process
armaRoots(c(-0.4334, -0.2034, -0.0801))
#process is stacionary


#testing unit root for all maturities
p_values <- rep(100,15)
for(i in 1:15){
        p_values[i] <- PP.test(yields[,i])$p.value}
#in all cases p-value > 50% => we take differences and test them

p_values_diff <- rep(100,15)
```

```
for(i in 1:15){
        p_values_diff[i] <- PP.test(diff(yields[,i]))$p.value}
#p-values = 0.01 => we work with 1st differences



#ZCY2Y
acf2(diff(yields[,2]))
sarima(yields[,2],3,1,5,details = FALSE)

armaRoots(c( -0.0029, 0.076, 0.8430))
#process is stationary
armaRoots(c(0.2630, 0.1025, 0.8396, -0.2356, -0.0604))
#process is invertible




#ZCY3Y
acf2(diff(yields[,3]))
sarima(yields[,3],12,1,0,details = FALSE)

armaRoots(c(-0.2713, -0.1093, -0.0100, 0.0122, 0.0203, -0.0277, -0.0354, 0.0493,
0.0486, 0.0208, 0.0393, -0.0478))
#process is stationary


#ZCY4Y
acf2(diff(yields[,4]))
sarima(yields[,4],12,1,0,details = FALSE)

armaRoots(c(-0.2853, -0.1421, 0.0161, 0.0059, 0.0247, -0.0190, -0.0135, 0.0271,
0.0385, 0.0403, 0.0494, -0.0476))
#process is invertible


#ZCY5Y
acf2(diff(yields[,5]))
sarima(yields[,5],2,1,3,details = FALSE)

armaRoots(c(0.1278, 0.7802))
#process is stationary

armaRoots(c(0.4102, 0.8100, -0.3062))
#process is invertible


#ZCY6Y
acf2(diff(yields[,6]))
sarima(yields[,6],2,1,3,details = FALSE)

armaRoots(c(0.1351, 0.7540))
#process is stationary
armaRoots(c(0.4070, 0.7822, -0.2931))
#process is invertible


#ZCY7Y
acf2(diff(yields[,7]))
sarima(yields[,7],0,1,7,details = FALSE)

armaRoots(c(0.2727, 0.0467, -0.0723, 0.0281, -0.0272, -0.0043, -0.0534))
#process is invertible


#ZCY8Y
acf2(diff(yields[,8]))
sarima(yields[,8],0,1,7,details = FALSE)
```

```
armaRoots(c(0.2806, 0.0190, −0.0453, 0.0153, −0.0184, −0.0063, −0.0593))
#process is invertible


#ZCY9Y
acf2(diff(yields[,9]))
sarima(yields[,9],0,1,7,details = FALSE)

armaRoots(c(0.2890, −0.0032, −0.0230, 0.0033, −0.0107, −0.0045, −0.0594))
#process is invertible


#ZCY10Y
acf2(diff(yields[,10]))
sarima(yields[,10],0,1,8,details = FALSE)

armaRoots(c(0.2924, −0.0171, −0.0094, −0.0068, −0.0051, −0.0016, −0.0399, −0.0418))
#process is invertible


#ZCY11Y
acf2(diff(yields[,11]))
sarima(yields[,11],0,1,8,details = FALSE)

armaRoots(c(0.2938, −0.0233, −0.0008, −0.0117, −0.0024, 0.0038, −0.0294, −0.0560))
#process is invertible


#ZCY12Y
acf2(diff(yields[,12]))
sarima(yields[,12],8,1,0,details = FALSE)

armaRoots(c(−0.2915, −0.0619, −0.0122, 0.0030, 0.0000, −0.0090, 0.0215, 0.0729))
#process is stationary


#ZCY13Y
acf2(diff(yields[,13]))
sarima(yields[,13],8,1,1,details = FALSE)

armaRoots(c(0.2076, 0.0757, 0.0216, 0.0029, −0.0071, −0.0123, 0.0233, 0.0696))
#process is stationary
armaRoots(c(0.5054))
#process is invertible


#ZCY14Y
acf2(diff(yields[,14]))
sarima(yields[,14],8,1,1,details = FALSE)

armaRoots(c(0.2718, 0.0827, 0.0316, −0.0043, −0.0141, −0.0137, 0.0246, 0.0615))
#process is stationary
armaRoots(c(0.5798))
#process is invertible


#ZCY15Y
acf2(diff(yields[,15]))
sarima(yields[,15],5,1,2,details = FALSE)

armaRoots(c(1.1847, −0.5909, −0.1642, −0.0926, −0.0257))
#process is stationary
armaRoots(c(1.5202, −0.9712))
#process is invertible
```

Diebold-Li

```
#Run after PCA code
library(YieldCurve)
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
mydata = read.csv("input.csv", na.string = "", sep = ";")
attach(mydata)
yields <- na.omit(mydata[,1:18])[1:2625,]
frequencies <- rep(0,length(unique(yields$YYYY)))

for(i in 1:length(frequencies)){
        frequencies[i] <- length(yields$YYYY[which(yields$YYYY==(unique(yields$YYYY)[i]))])
        }
freq <- mean(frequencies[2:(length(frequencies)-1)])
st <- 13 - frequencies[1]/freq*12
yields <- ts(yields[,4:18], frequency=freq, start=c(2006,st))
maturity_SK <- seq(from = 1, to = 15, by = 1)

#Finding lambda that maximizes mid_term factor loading
mid_term <- function(lambda) {(1-exp(-30/12*lambda))/(30/12*lambda)-exp(-30/12*lambda)}
lambda_opt <- optimize(mid_term, interval=c(0, 1), maximum=TRUE)$maximum

#Plotting 3 loadings in DL model
tau <- seq(from = 0, to = 15, by = 0.01)
plot(tau,rep(1,times=length(tau)),type="l",ylim=c(0,1.3),ylab="Beta loadings",xlab="Maturity")
lines(tau,(1-exp(-lambda_opt*tau))/(lambda_opt*tau),col="red")
lines(tau,(1-exp(-lambda_opt*tau))/(lambda_opt*tau)-exp(-lambda_opt*tau),col="blue")
legend('topright', c("Level loading","Slope loading","Curvature loading") , lty=1,
 col=c('black', 'red', 'blue'),
 bty='n', cex=1.1)




factor_loadings <- function(lambda){
        a <- rep(0,15)
        b <- rep(0,15)
        for(i in 1:15){
                a[i] <- (1-exp(-i*lambda))/(i*lambda)
                b[i] <- (1-exp(-i*lambda))/(i*lambda)-exp(-i*lambda)
        }
        cbind(a,b)
}


NS_opt <- Nelson.Siegel( rate= yields, maturity=maturity_SK )
NS_DL <- matrix(data=0,ncol=3,nrow=dim(yields)[1])
NS_mean <- matrix(data=0,ncol=3,nrow=dim(yields)[1])

#Calculating average lambda of fitted NS models
lambda_mean <- mean(NS_opt[,4])

#Creating histogram of lambdas from fitted NS models
hist(NS_opt[,4],xlab="lambda",main="Distribution of lambda")
axis(side=1, at=lambda_opt, labels = FALSE)
text(x=lambda_opt,  par("usr")[3],
     labels = "Diebold-Li",col="black", pos = 1, xpd = TRUE, cex=1)
axis(side=1, at=lambda_mean, labels = FALSE)
text(x=lambda_mean,  par("usr")[3],
     labels = "Average",col="black", pos = 1, xpd = TRUE, cex=1)


#Fitting linear models with fixed lambda
for (i in 1:dim(NS_DL)[1]){
        NS_DL[i,] <- lm(yields[i,] ~ factor_loadings(lambda_opt))$coeff
        NS_mean[i,] <- lm(yields[i,] ~ factor_loadings(lambda_mean))$coeff
}
```

```
library(astsa)
library(fArma)
library(WDI)
library(urca)

#Diebold-Li approach => AR(1) processes
sarima(NS_DL[,1],1,0,0,details = FALSE)
sarima(NS_DL[,2],1,0,0,details = FALSE)
sarima(NS_DL[,3],1,0,0,details = FALSE)

sarima(NS_mean[,1],1,0,0,details = FALSE)
sarima(NS_mean[,2],1,0,0,details = FALSE)
sarima(NS_mean[,3],1,0,0,details = FALSE)

#Our approach => Finding the best ARIMA process

#is there unit root in the time series?
plot(NS_DL[,1],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_DL[,1],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(NS_DL[,1],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_DL[,1],type="mu"))
summary(ur.kpss(NS_DL[,1],type="tau"))
#Phillips-Perron test
PP.test(NS_DL[,1])

#There is an unit root in the original time series, we need to take differences
plot(diff(NS_DL[,1]),type = "l")
mean(diff(NS_DL[,1]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_DL[,1]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_DL[,1]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(NS_DL[,1]))

#we work with first differences of the original time series

acf2(diff(NS_DL[,1]))
sarima(NS_DL[,1],0,1,1,details = FALSE)

#testing invertibility of the process
armaRoots(c(0.3799))
#process is invertible


#is there unit root in the time series?
plot(NS_DL[,2],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_DL[,2],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(NS_DL[,2],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_DL[,2],type="mu"))
summary(ur.kpss(NS_DL[,2],type="tau"))
#Phillips-Perron test
PP.test(NS_DL[,2])

#There is an unit root in the original time series, we need to take differences
plot(diff(NS_DL[,2]),type = "l")
mean(diff(NS_DL[,2]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_DL[,2]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_DL[,2]),type="mu"))
#we dont refuse the H0 that the process is stationary
```

```
PP.test(diff(NS_DL[,2]))

#we work with first differences of the original time series

acf2(diff(NS_DL[,2]))
sarima(NS_DL[,2],3,1,1,details = FALSE)

#testing stationarity of the process
armaRoots(c(0.2726, 0.0389, 0.0919))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.7472))
#process is invertible


#is there unit root in the time series?
plot(NS_DL[,3],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_DL[,3],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(NS_DL[,3],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_DL[,3],type="mu"))
summary(ur.kpss(NS_DL[,3],type="tau"))
#Phillips-Perron test
PP.test(NS_DL[,3])

#There is an unit root in the original time series, we need to take differences
plot(diff(NS_DL[,3]),type = "l")
mean(diff(NS_DL[,3]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_DL[,3]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_DL[,3]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(NS_DL[,3]))

#we work with first differences of the original time series

acf2(NS_DL[,3])
sarima(NS_DL[,3],1,0,5,details = FALSE)
#testing stationarity of the process
armaRoots(c(0.9984))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.5231, 0.0889, -0.0585, 0.0595, 0.0477))
#process is invertible

acf2(diff(NS_DL[,3]))
sarima(NS_DL[,3],2,1,4,details = FALSE)
#testing stationarity of the process
armaRoots(c(0.9438, -0.6173))
#process is stationary
#testing invertibility of the process
armaRoots(c(1.4678, -1.0269, 0.2072, 0.1205))
#process is invertible


#Lambda = mean approach
#is there unit root in the time series?
plot(NS_mean[,1],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_mean[,1],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(NS_mean[,1],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_mean[,1],type="mu"))
summary(ur.kpss(NS_mean[,1],type="tau"))
#Phillips-Perron test
```

```
PP.test(NS_mean[,1])

#There is an unit root in the original time series, we need to take differences
plot(diff(NS_mean[,1]),type = "l")
mean(diff(NS_mean[,1]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_mean[,1]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_mean[,1]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(NS_mean[,1]))

#we work with first differences of the original time series

acf2(diff(NS_mean[,1]))
sarima(NS_mean[,1],2,1,6,details = FALSE)
#testing stationarity of the process
armaRoots(c(1.2664, -0.7329))
#process is stationary
#testing invertibility of the process
armaRoots(c(1.7263, -1.2431, 0.1996, 0.1707, -0.0697, 0.0047))
#process is invertible


#is there unit root in the time series?
plot(NS_mean[,2],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_mean[,2],type="drift",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_mean[,2],type="mu"))
#Phillips-Perron test
PP.test(NS_mean[,2])

#There is an unit root in the original time series, we need to take differences
plot(diff(NS_mean[,2]),type = "l")
mean(diff(NS_mean[,2]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_mean[,2]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_mean[,2]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(NS_mean[,2]))

#we work with first differences of the original time series

acf2(diff(NS_mean[,2]))
sarima(NS_mean[,2],7,1,2,details = FALSE)
#testing stationarity of the process
armaRoots(c(-1.0038, -0.9751, -0.3831, -0.1829, -0.1129, -0.0975, -0.0794))
#process is stationary
#testing invertibility of the process
armaRoots(c(-0.6550, -0.6184))
#process is invertible


#is there unit root in the time series?
plot(NS_mean[,3],type = "l")
#Adjusted Dickey-Fuller test
summary(ur.df(NS_mean[,3],type="drift",lags=20,selectlags="BIC"))
summary(ur.df(NS_mean[,3],type="trend",lags=20,selectlags="BIC"))
#KPSS test
summary(ur.kpss(NS_mean[,3],type="mu"))
summary(ur.kpss(NS_mean[,3],type="tau"))
#Phillips-Perron test
PP.test(NS_mean[,3])
```

```
#There is an unit root in the original time series, we need to take differences
plot(diff(NS_mean[,3]),type = "l")
mean(diff(NS_mean[,3]))

#mean is close to zero and there is no significant drift so the type of test is none
summary(ur.df(diff(NS_mean[,3]),type="none",lags=20,selectlags="BIC"))
#we refuse the H0 that there is an unit root in differentiated time series
summary(ur.kpss(diff(NS_mean[,3]),type="mu"))
#we dont refuse the H0 that the process is stationary
PP.test(diff(NS_mean[,3]))



acf2(NS_mean[,3])
sarima(NS_mean[,3],1,0,5,details = FALSE)
#testing stationarity of the process
armaRoots(c(0.9982))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.5081, 0.1318, −0.0809, 0.0748, 0.0402))
#process is invertible

acf2(diff(NS_mean[,3]))
sarima(NS_mean[,3],3,1,3,details = FALSE)
#testing stationarity of the process
armaRoots(c(0.1218, −0.3501, 0.2384))
#process is stationary
#testing invertibility of the process
armaRoots(c(0.6324, −0.2825, 0.3280))
#process is invertible


#Correlation matrices for two different choices of lambda
cor(NS_mean)
cor(NS_DL)


#graphs

#At first we create time series of PCA yields
PCA_yields <− matrix(data=0,nrow=dim(tr_yields)[1],ncol=dim(tr_yields)[2])
PCA_yields[,1] <− −1*tr_yields[,1]
PCA_yields[,2] <− tr_yields[,2]
PCA_yields[,3] <− tr_yields[,3]

for(i in 1:3){
        PCA_yields[,i] <− PCA_yields[,i]/stdev(PCA_yields[,i])*(stdev(NS_DL[,i])+stdev(NS_mean[,i]))/2
        PCA_yields[,i] <− PCA_yields[,i]+(rep(mean(NS_DL[,i]),dim(NS_DL)[1])+rep(mean(NS_mean[,i]),dim(NS_mean)[1]))/2
        }


#level factor
cor(cbind(NS_mean[,1],NS_DL[,1],−1*tr_yields[,1]))
plot(NS_DL[,1],type="l",ylab="Level factor",ylim=c(0,9))
lines(NS_mean[,1],col="red")
lines(PCA_yields[,1],col="blue")

legend('topright', c("NS_DL","NS_mean","PCA") , lty=1,
 col=c('black', 'red', 'blue'),
 bty='n', cex=1.1)

#slope factor
cor(cbind(NS_mean[,2],NS_DL[,2],PCA_yields[,2]))
plot(NS_DL[,2],type="l",ylab="Slope factor",ylim=c(−8,2))
lines(NS_mean[,2],col="red")
lines(PCA_yields[,2],col="blue")

legend('bottomright', c("NS_DL","NS_mean","PCA") , lty=1,
```

```
 col=c('black', 'red', 'blue'),
 bty='n', cex=1.1)

#curvature factor
cor(cbind(NS_mean[,3],NS_DL[,3],PCA_yields[,3]))
plot(NS_DL[,3],type="l",ylab="Curvature factor",ylim=c(-15,9))
lines(NS_mean[,3],col="red")
lines(PCA_yields[,3],col="blue")

legend('topright', c("NS_DL","NS_mean","PCA") , lty=1,
 col=c('black', 'red', 'blue'),
 bty='n', cex=1.1)
```

Predictions

```
#Run after PCA, Diebold-Li and arima_before_forecasting
setwd("C:\\Users\\Peter Carsky\\Documents\\Diplomovka")
yields_all = na.omit(read.csv("input.csv", na.string = "", sep = ";"))

#Number of time periods we want to predict
n_pred <- 70

#Yields before forecasts, common for each dataframe
yields_bf <- yields_all[1:2625,1:18]
#Yields that will be compared with forecasts(true values of time series)
yields_tf <- yields_all[2626:(2625+n_pred),1:18]


######################### DEFINING FORECASTING MODELS ######################
#Forecasts from ARIMA processes specific for each maturity
ARIMA <- matrix(data=0,ncol=15,nrow=n_pred)
ARIMA[,1] <- sarima.for(yields[,1],n_pred,3,1,0)$pred
ARIMA[,2] <- sarima.for(yields[,2],n_pred,8,1,1)$pred
ARIMA[,3] <- sarima.for(yields[,3],n_pred,2,1,3)$pred
ARIMA[,4] <- sarima.for(yields[,4],n_pred,0,1,3)$pred
ARIMA[,5] <- sarima.for(yields[,5],n_pred,0,1,3)$pred
ARIMA[,6] <- sarima.for(yields[,6],n_pred,3,1,3)$pred
ARIMA[,7] <- sarima.for(yields[,7],n_pred,0,1,7)$pred
ARIMA[,8] <- sarima.for(yields[,8],n_pred,0,1,7)$pred
ARIMA[,9] <- sarima.for(yields[,9],n_pred,0,1,8)$pred
ARIMA[,10] <- sarima.for(yields[,10],n_pred,0,1,9)$pred
ARIMA[,11] <- sarima.for(yields[,11],n_pred,0,1,8)$pred
ARIMA[,12] <- sarima.for(yields[,12],n_pred,0,1,8)$pred
ARIMA[,13] <- sarima.for(yields[,13],n_pred,0,1,8)$pred
ARIMA[,14] <- sarima.for(yields[,14],n_pred,0,1,1)$pred
ARIMA[,15] <- sarima.for(yields[,15],n_pred,5,1,2)$pred

#Creating time series with merged historic data with predictions
colnames(ARIMA) <- colnames(yields_bf[,4:18])
ARIMA_ts <- ts(rbind(yields_bf[,4:18],ARIMA), frequency=freq, start=c(2006,st))

#Defining explanatory time series for Euro adoption
#creating dummy regressor
ea <- rep(1,dim(yields)[1])
ea[1:length(YYYY[YYYY<2009])] <- 0

#creating regressor for Euro adoption which slowly turns from 0 to 1
eac <- rep(1,dim(yields)[1])
eac[1:length(YYYY[YYYY<2008])] <- 0
for(i in 1:length(YYYY[YYYY>2007 & YYYY<2010])){
        eac[length(YYYY[YYYY<2008])+i] <- i/(length(YYYY[YYYY>2007 & YYYY<2010])+1)
        }


#Forecasts from ARIMA processes with dummy regression term
library(forecast)

ARIMAXd <- matrix(data=0,ncol=15,nrow=n_pred)
```

```
ARIMAXd[,1] <- forecast(Arima(yields[,1],c(3,1,0),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,2] <- forecast(Arima(yields[,2],c(8,1,1),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,3] <- forecast(Arima(yields[,3],c(2,1,3),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,4] <- forecast(Arima(yields[,4],c(0,1,3),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,5] <- forecast(Arima(yields[,5],c(0,1,3),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,6] <- forecast(Arima(yields[,6],c(3,1,3),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,7] <- forecast(Arima(yields[,7],c(0,1,7),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,8] <- forecast(Arima(yields[,8],c(0,1,7),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,9] <- forecast(Arima(yields[,9],c(0,1,8),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,10] <- forecast(Arima(yields[,10],c(0,1,9),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,11] <- forecast(Arima(yields[,11],c(0,1,8),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,12] <- forecast(Arima(yields[,12],c(0,1,8),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,13] <- forecast(Arima(yields[,13],c(0,1,8),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,14] <- forecast(Arima(yields[,14],c(0,1,1),xreg=ea), xreg=rep(1,n_pred))$mean
ARIMAXd[,15] <- forecast(Arima(yields[,15],c(5,1,2),xreg=ea), xreg=rep(1,n_pred))$mean

#Forecasts from ARIMA processes with linear regression term during years 2008-2009

ARIMAXl <- matrix(data=0,ncol=15,nrow=n_pred)
ARIMAXl[,1] <- forecast(Arima(yields[,1],c(3,1,0),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,2] <- forecast(Arima(yields[,2],c(8,1,1),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,3] <- forecast(Arima(yields[,3],c(2,1,3),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,4] <- forecast(Arima(yields[,4],c(0,1,3),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,5] <- forecast(Arima(yields[,5],c(0,1,3),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,6] <- forecast(Arima(yields[,6],c(3,1,3),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,7] <- forecast(Arima(yields[,7],c(0,1,7),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,8] <- forecast(Arima(yields[,8],c(0,1,7),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,9] <- forecast(Arima(yields[,9],c(0,1,8),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,10] <- forecast(Arima(yields[,10],c(0,1,9),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,11] <- forecast(Arima(yields[,11],c(0,1,8),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,12] <- forecast(Arima(yields[,12],c(0,1,8),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,13] <- forecast(Arima(yields[,13],c(0,1,8),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,14] <- forecast(Arima(yields[,14],c(0,1,1),xreg=eac), xreg=rep(1,n_pred))$mean
ARIMAXl[,15] <- forecast(Arima(yields[,15],c(5,1,2),xreg=eac), xreg=rep(1,n_pred))$mean


#Forecasting first three principal components
PCA_for <- matrix(data=0,ncol=3,nrow=n_pred)
PCA_for[,1] <- sarima.for(tr_yields[,1],n_pred,1,1,2)$pred
PCA_for[,2] <- sarima.for(tr_yields[,2],n_pred,8,1,0)$pred
PCA_for[,3] <- sarima.for(tr_yields[,3],n_pred,5,1,1)$pred

#Transforming predictions from PC base to original base
PCA <- PCA_for%*%t(ZC_pca$rotation[,1:3]) + means[1:n_pred,]


#Forecasting Betas from Nelson-Siegel
NS_DL_for <- matrix(data=0,ncol=3,nrow=n_pred)
NS_DL_for[,1] <- sarima.for(NS_DL[,1],n_pred,0,1,1)$pred
NS_DL_for[,2] <- sarima.for(NS_DL[,2],n_pred,3,1,1)$pred
NS_DL_for[,3] <- sarima.for(NS_DL[,3],n_pred,2,1,4)$pred


NS_mean_for <- matrix(data=0,ncol=3,nrow=n_pred)
NS_mean_for[,1] <- sarima.for(NS_mean[,1],n_pred,2,1,6)$pred
NS_mean_for[,2] <- sarima.for(NS_mean[,2],n_pred,7,1,2)$pred
NS_mean_for[,3] <- sarima.for(NS_mean[,3],n_pred,3,1,3)$pred

#Computing yields implied by forecasted betas
NS_DL_pred <- t(cbind(rep(1,15),factor_loadings(lambda_opt))%*%t(NS_DL_for))
NS_mean_pred <- t(cbind(rep(1,15),factor_loadings(lambda_mean))%*%t(NS_mean_for))



#Modelling all processes only by default AR(1) as suggested by Diebold and Li

#Forecasts from ARIMA processes specific for each maturity
#Forecasts from Regression models with AR(1) errors
```

```
ARIMA1 <- matrix(data=0,ncol=15,nrow=n_pred)
ARIMAXd1 <- matrix(data=0,ncol=15,nrow=n_pred)
ARIMAXl1 <- matrix(data=0,ncol=15,nrow=n_pred)

for(i in 1:15) {
        ARIMA1[,i] <- sarima.for(yields[,i],n_pred,1,0,0)$pred
        ARIMAXd1[,i] <- forecast(Arima(yields[,i],c(1,0,0),xreg=ea), xreg=rep(1,n_pred))$mean
        ARIMAXl1[,i] <- forecast(Arima(yields[,i],c(1,0,0),xreg=eac), xreg=rep(1,n_pred))$mean
        }


#Forecasting first three principal components
PCA1_for <- matrix(data=0,ncol=3,nrow=n_pred)


for(i in 1:3) {
        PCA1_for[,i] <- sarima.for(tr_yields[,i],n_pred,1,0,0)$pred
        }


#Transforming predictions from PC base to original base
PCA1 <- PCA1_for%*%t(ZC_pca$rotation[,1:3]) + means[1:n_pred,]


#Forecasting first three principal components, while second one is modelled also using exogenous time series
PCA1_forx <- matrix(data=0,ncol=3,nrow=n_pred)
PCA1_forx[,1] <- sarima.for(tr_yields[,1],n_pred,1,0,0)$pred
PCA1_forx[,2] <- forecast(Arima(tr_yields[,2],c(1,0,0),xreg=eac), xreg=rep(1,n_pred))$mean
PCA1_forx[,3] <- sarima.for(tr_yields[,3],n_pred,1,0,0)$pred

#Transforming predictions from PC base to original base
PCA1x <- PCA1_forx%*%t(ZC_pca$rotation[,1:3]) + means[1:n_pred,]


#Forecasting Betas from Nelson-Siegel
NS_DL1_for <- matrix(data=0,ncol=3,nrow=n_pred)
NS_mean1_for <- matrix(data=0,ncol=3,nrow=n_pred)

for(i in 1:3) {
        NS_DL1_for[,i] <- sarima.for(NS_DL[,i],n_pred,1,0,0)$pred
        NS_mean1_for[,i] <- sarima.for(NS_mean[,i],n_pred,1,0,0)$pred
        }

#Computing yields implied by forecasted betas
NS_DL1_pred <- t(cbind(rep(1,15),factor_loadings(lambda_opt))%*%t(NS_DL1_for))
NS_mean1_pred <- t(cbind(rep(1,15),factor_loadings(lambda_mean))%*%t(NS_mean1_for))


#Forecasting Betas from Nelson-Siegel with slope modelled by eac regressor
NS_DL1_forx <- matrix(data=0,ncol=3,nrow=n_pred)
NS_DL1_forx[,1] <- sarima.for(NS_DL[,1],n_pred,1,0,0)$pred
NS_DL1_forx[,2] <- forecast(Arima(NS_DL[,2],c(1,0,0),xreg=eac), xreg=rep(1,n_pred))$mean
NS_DL1_forx[,3] <- sarima.for(NS_DL[,3],n_pred,1,0,0)$pred

NS_mean_forx <- matrix(data=0,ncol=3,nrow=n_pred)
NS_mean_forx[,1] <- sarima.for(NS_mean[,1],n_pred,2,1,6)$pred
NS_mean_forx[,2] <- forecast(Arima(NS_mean[,2],c(7,1,2),xreg=eac), xreg=rep(1,n_pred))$mean
NS_mean_forx[,3] <- sarima.for(NS_mean[,3],n_pred,3,1,3)$pred


#Computing yields implied by forecasted betas
NS_DL1x_pred <- t(cbind(rep(1,15),factor_loadings(lambda_opt))%*%t(NS_DL1_forx))
NS_meanx_pred <- t(cbind(rep(1,15),factor_loadings(lambda_mean))%*%t(NS_mean_forx))

#Modelling yield curve by vector autoregression
library(vars)

var <- VAR(yields, p=1, type="const")
```

```
summary(var)
#Creating matrix of estimated coefficients
var_coeff <- matrix(data=0,nrow=15,ncol=16)
var_coeff[1,] <- var$varresult$ZCY1Y$coefficients
var_coeff[2,] <- var$varresult$ZCY2Y$coefficients
var_coeff[3,] <- var$varresult$ZCY3Y$coefficients
var_coeff[4,] <- var$varresult$ZCY4Y$coefficients
var_coeff[5,] <- var$varresult$ZCY5Y$coefficients
var_coeff[6,] <- var$varresult$ZCY6Y$coefficients
var_coeff[7,] <- var$varresult$ZCY7Y$coefficients
var_coeff[8,] <- var$varresult$ZCY8Y$coefficients
var_coeff[9,] <- var$varresult$ZCY9Y$coefficients
var_coeff[10,] <- var$varresult$ZCY10Y$coefficients
var_coeff[11,] <- var$varresult$ZCY11Y$coefficients
var_coeff[12,] <- var$varresult$ZCY12Y$coefficients
var_coeff[13,] <- var$varresult$ZCY13Y$coefficients
var_coeff[14,] <- var$varresult$ZCY14Y$coefficients
var_coeff[15,] <- var$varresult$ZCY15Y$coefficients

names <- rep("",15)
for(i in 1:15){
        names[i] <-  paste(i, "Y", sep="")
}

rownames(var_coeff) <- names
colnames(var_coeff) <- c(names,"Intercept")

round(var_coeff,2)[,1:15]
round(var_coeff,2)[,16]


#Forecasting yield curve by fitted VAR
fcst <- forecast(var,h = n_pred)
var_pred <- matrix(data = 0, nrow = n_pred, ncol = 15)


for(i in 1:15){
        var_pred[,i] <- as.data.frame(fcst)[(1+(i-1)*70):(i*70),3]
        }

#Using VAR with exogenous time series describing Euro adoption
eav <- as.matrix(eac)
colnames(eav) <- "euro_adoption"
varx <- VAR(yields, p=1, type="const", exogen = eav)
summary(varx)

#Creating matrix of estimated coefficients
varx_coeff <- matrix(data=0,nrow=15,ncol=17)
varx_coeff[1,] <- varx$varresult$ZCY1Y$coefficients
varx_coeff[2,] <- varx$varresult$ZCY2Y$coefficients
varx_coeff[3,] <- varx$varresult$ZCY3Y$coefficients
varx_coeff[4,] <- varx$varresult$ZCY4Y$coefficients
varx_coeff[5,] <- varx$varresult$ZCY5Y$coefficients
varx_coeff[6,] <- varx$varresult$ZCY6Y$coefficients
varx_coeff[7,] <- varx$varresult$ZCY7Y$coefficients
varx_coeff[8,] <- varx$varresult$ZCY8Y$coefficients
varx_coeff[9,] <- varx$varresult$ZCY9Y$coefficients
varx_coeff[10,] <- varx$varresult$ZCY10Y$coefficients
varx_coeff[11,] <- varx$varresult$ZCY11Y$coefficients
varx_coeff[12,] <- varx$varresult$ZCY12Y$coefficients
varx_coeff[13,] <- varx$varresult$ZCY13Y$coefficients
varx_coeff[14,] <- varx$varresult$ZCY14Y$coefficients
varx_coeff[15,] <- varx$varresult$ZCY15Y$coefficients

rownames(varx_coeff) <- names
colnames(varx_coeff) <- c(names,"Intercept","Euro adoption")

round(varx_coeff,2)[,1:15]
```

```
round(varx_coeff,2)[,16:17]

#Forecasting yield curve by fitted VARX
fcstx <- forecast(varx,h=n_pred,dumvar=as.matrix(rep(1,n_pred)))

varx_pred <- matrix(data = 0, nrow = n_pred, ncol = 15)

for(i in 1:15){
        varx_pred[,i] <- as.data.frame(fcstx)[(1+(i-1)*70):(i*70),3]
        }



################## EVALUATION #####################################
#Counting cumulative sum of squared errors
cumsum_NS_DL <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_NS_mean <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_PCA <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMA <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMAXd <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMAXl <- matrix(data=0,ncol=15,nrow=n_pred)

cumsum_NS_DL1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_NS_mean1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_PCA1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMA1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMAXd1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_ARIMAXl1 <- matrix(data=0,ncol=15,nrow=n_pred)

cumsum_PCA1x <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_NS_DL1x <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_NS_meanx <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_var1 <- matrix(data=0,ncol=15,nrow=n_pred)
cumsum_var1x <- matrix(data=0,ncol=15,nrow=n_pred)

for(i in 1:n_pred){
        cumsum_NS_DL[i,] <- apply((NS_DL_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_NS_mean[i,] <- apply((NS_mean_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_PCA[i,] <- apply((PCA[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMA[i,] <- apply((ARIMA[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMAXd[i,] <- apply((ARIMAXd[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMAXl[i,] <- apply((ARIMAXl[1:i,]-yields_tf[1:i,4:18])^2,2,sum)

        cumsum_NS_DL1[i,] <- apply((NS_DL1_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_NS_mean1[i,] <- apply((NS_mean1_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_PCA1[i,] <- apply((PCA1[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMA1[i,] <- apply((ARIMA1[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMAXd1[i,] <- apply((ARIMAXd1[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_ARIMAXl1[i,] <- apply((ARIMAXl1[1:i,]-yields_tf[1:i,4:18])^2,2,sum)

        cumsum_NS_DL1x[i,] <- apply((NS_DL1x_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_NS_meanx[i,] <- apply((NS_meanx_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_PCA1x[i,] <- apply((PCA1x[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_var1[i,] <- apply((var_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        cumsum_var1x[i,] <- apply((varx_pred[1:i,]-yields_tf[1:i,4:18])^2,2,sum)
        }

#Counting cumulative errors for each method and different time periods
round(apply(cumsum_NS_DL,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_NS_DL1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_NS_mean,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_NS_mean1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_PCA,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_PCA1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_ARIMA,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_ARIMA1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_ARIMAXd,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_ARIMAXd1,1,sum)[seq(from=10,to=70,by=10)],2)
```

```
round(apply(cumsum_ARIMAXl,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_ARIMAXl1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_NS_DL1x,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_NS_meanx,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_PCA1x,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_var1,1,sum)[seq(from=10,to=70,by=10)],2)
round(apply(cumsum_var1x,1,sum)[seq(from=10,to=70,by=10)],2)

#Counting cumulative errors for each method and different maturities
round(cumsum_NS_DL[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_NS_DL1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_NS_mean[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_NS_mean1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_PCA[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_PCA1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMA[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMA1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMAXd[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMAXd1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMAXl[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_ARIMAXl1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_NS_DL1x[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_NS_meanx[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_PCA1x[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_var1[n_pred,seq(from=1,to=15,by=2)],1)
round(cumsum_var1x[n_pred,seq(from=1,to=15,by=2)],1)

#Counting median sum of squared errors
round(median(cumsum_NS_DL[n_pred,]),2)
round(median(cumsum_NS_DL1[n_pred,]),2)
round(median(cumsum_NS_mean[n_pred,]),2)
round(median(cumsum_NS_mean1[n_pred,]),2)
round(median(cumsum_PCA[n_pred,]),2)
round(median(cumsum_PCA1[n_pred,]),2)
round(median(cumsum_ARIMA[n_pred,]),2)
round(median(cumsum_ARIMA1[n_pred,]),2)
round(median(cumsum_ARIMAXd[n_pred,]),2)
round(median(cumsum_ARIMAXd1[n_pred,]),2)
round(median(cumsum_ARIMAXl[n_pred,]),2)
round(median(cumsum_ARIMAXl1[n_pred,]),2)
round(median(cumsum_NS_DL1x[n_pred,]),2)
round(median(cumsum_NS_meanx[n_pred,]),2)
round(median(cumsum_PCA1x[n_pred,]),2)
round(median(cumsum_var1[n_pred,]),2)
round(median(cumsum_var1x[n_pred,]),2)

#Counting sum of ranked errors
#Creating matrix of SSEs
SSE <- matrix(data=0,ncol=15,nrow=17)

SSE[1,] <- cumsum_NS_DL[n_pred,]
SSE[2,] <- cumsum_NS_DL1[n_pred,]
SSE[3,] <- cumsum_NS_mean[n_pred,]
SSE[4,] <- cumsum_NS_mean1[n_pred,]
SSE[5,] <- cumsum_PCA[n_pred,]
SSE[6,] <- cumsum_PCA1[n_pred,]
SSE[7,] <- cumsum_ARIMA[n_pred,]
SSE[8,] <- cumsum_ARIMA1[n_pred,]
SSE[9,] <- cumsum_ARIMAXd[n_pred,]
SSE[10,] <- cumsum_ARIMAXd1[n_pred,]
SSE[11,] <- cumsum_ARIMAXl[n_pred,]
SSE[12,] <- cumsum_ARIMAXl1[n_pred,]
SSE[13,] <- cumsum_NS_DL1x[n_pred,]
SSE[14,] <- cumsum_NS_meanx[n_pred,]
SSE[15,] <- cumsum_PCA1x[n_pred,]
SSE[16,] <- cumsum_var1[n_pred,]
SSE[17,] <- cumsum_var1x[n_pred,]
```

```
#Creating vector of sum of ranked errors
SRE <- rep(0,17)

for(i in 1:15){
        for(j in 1:17){
                for(k in 1:17){
                        if(SSE[j,i] >= SSE[k,i]){
                                SRE[j] <- SRE[j] + 1
                                }
                        }
                }
        }

#Creating average rank per method
round(SRE/15,2)

########################### GRAPHS ###################################
#Plotting cumulative errors of different forecasting methods
plot(apply(cumsum_NS_DL,1,sum),type="l",ylim=c(0,150),ylab="Cumulative error")
lines(apply(cumsum_NS_mean,1,sum),col="red")
lines(apply(cumsum_PCA,1,sum),col="blue")
lines(apply(cumsum_ARIMA,1,sum),col="seagreen")
lines(apply(cumsum_ARIMAXd,1,sum),col="violet")
lines(apply(cumsum_ARIMAXl,1,sum),col="orange")

legend('topleft', c("NS_DL","NS_mean","PCA","ARIMA","Dummy reg","Linear reg") , lty=1,
 col=c('black', 'red', 'blue',"seagreen","violet","orange"),
 bty='n', cex=1.1)

#Plotting cumulative errors of methods modelled by AR(1) processes
plot(apply(cumsum_NS_DL1,1,sum),type="l",ylim=c(0,150),ylab="Cumulative error")
lines(apply(cumsum_NS_mean1,1,sum),col="red")
lines(apply(cumsum_PCA1,1,sum),col="blue")
lines(apply(cumsum_ARIMA1,1,sum),col="seagreen")
lines(apply(cumsum_ARIMAXd1,1,sum),col="violet")
lines(apply(cumsum_ARIMAXl1,1,sum),col="orange")

legend('topleft', c("NS_DL","NS_mean","PCA","ARIMA","Dummy reg","Linear reg") , lty=1,
 col=c('black', 'red', 'blue',"seagreen","violet","orange"),
 bty='n', cex=1.1)


#Plotting cumulative errors of methods modelled by other methods
plot(apply(cumsum_NS_DL1x,1,sum),type="l",ylim=c(0,150),ylab="Cumulative error")
lines(apply(cumsum_NS_meanx,1,sum),col="red")
lines(apply(cumsum_PCA1x,1,sum),col="blue")
lines(apply(cumsum_var1,1,sum),col="seagreen")
lines(apply(cumsum_var1x,1,sum),col="violet")

legend('topleft', c("NS_DL with linear reg","NS_mean with linear reg","PCA with linear reg","VAR(1)","VAR(1) with linear reg") , lty=1,
 col=c('black', 'red', 'blue',"seagreen","violet"),
 bty='n', cex=1.1)
```