# MSAN 604: Final Project Report

Deena Liz John
Guoqiang Liang
Mathew Shaw
Maria Vasilenko

## 1 INTRODUCTION

Since the 1980s Canada has struggled with a high bankruptcy rate. In an eight year period Canada's national debt tripled and its debt-to-GDP ratio soared to above 70%. At that time Canada's situation was officially labeled as a "debt crisis". In spite of government efforts since then, Canada's debt-to-GDP level has now grown to more than 220%.

The goal of this report is to forecast Canada's bankruptcy rate by month, for the next two years. Accurate predictions of national bankruptcy rates will allow for better planning from national banks, insurance companies, credit-lenders, and the government. In addition, it will provide better insights to decision-makers to predict bankruptcy events such as wrong investment decisions, poor investment environments, low cash flows, etc.

The data used is 23 years of temporal data of Canada's bankruptcy rate as well as additional influential factors (unemployment rate, population, housing price index). We used this data to build several models which accurately forecast bankruptcy rates for the next two years.

## 2 MODELING APPROACHES

There are a number of approaches to modeling a time series. Our approach to model selection was to look at the characteristics of the observed data, i.e. if trend and seasonality are observed, and if external variables are used for forecasting. Figure 1 shows the monthly data for bankruptcy rate, the variable to be forecasted, and external factors - unemployment rate, population and housing price index - considered to help prediction.

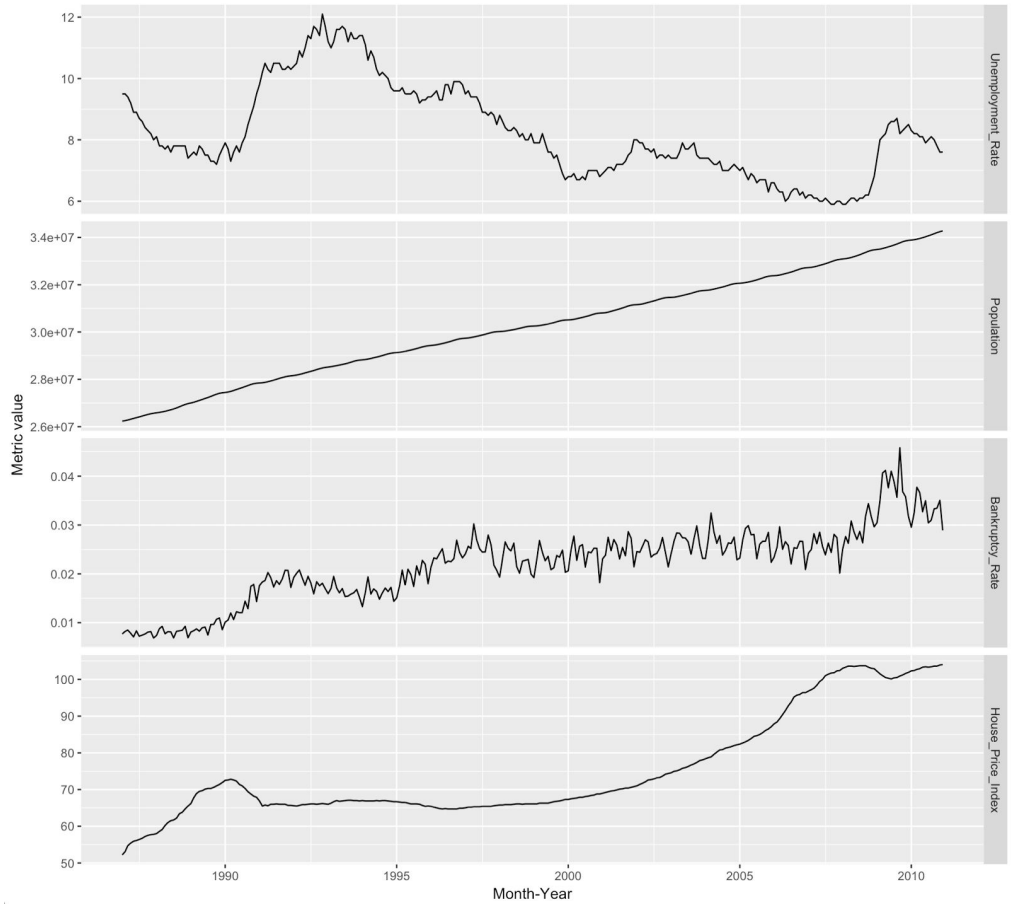



Figure 1: Variables trend across time

Bankruptcy rate shows a clear rising trend over time, and the same is observed for population and housing price index. Furthermore, the constant fluctuations in bankruptcy rate indicate possible seasonality which could be correlated to unemployment rate. From Figure 2(a), we observe bankruptcy rate to be highly correlated to population and housing price index. It is interesting to note from Figure 2(b) that considering the subsetted data starting from 1993 significantly improves the correlation of bankruptcy rate with external variables. Hence, the analysis will focus on both time frames for model fitting.

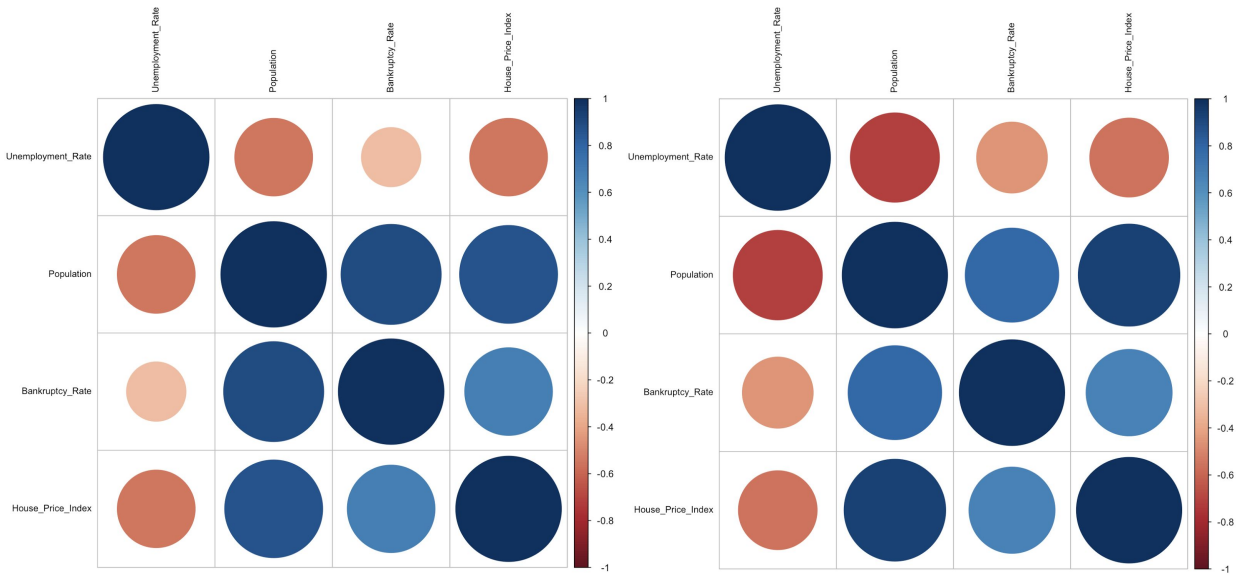



Figure 2: (a) Correlation matrix for 1987-2010
(b) Correlation matrix for 1993-2010

Such type of non-stationary time series can be modeled by univariate techniques such as ARIMA and Holt-Winters, or multivariate techniques such as SARIMAX, VAR and VARX. Further we will briefly describe each of these approaches and why a technique might seem appropriate. For details about checking for non-stationarity of bankruptcy rate time series, please refer to the appendix.

### 2.1 Univariate Forecasting Techniques

Univariate techniques are the more direct approaches that forecast solely using available historical data, 23 years in this analysis. Given the non-stationarity of the time series, two main univariate methods are available as described below.

- Holt-Winters Multiplicative Triple Exponential Smoothing

  Holt-Winters is the simplest of possible approaches in terms of interpretation as it lacks mathematical complexity. This method captures non-stationarity by considering the forecast equation and three smoothing equations to account for level, trend and seasonality. Further, the increasing peak to trough seasonal variations over time indicate a multiplicative model to be appropriate.

- Seasonal AutoRegressive Integrated Moving Average (SARIMA)

  SARIMA is a generalization of an autoregressive moving average (ARMA) model with differencing steps applied to eliminate non-stationarity. Further, the time series is

regressed on its own prior values, (corresponds to AR), and regression error is represented as linear combination of previous errors (corresponds to MA). This approach could give good results with appropriate parameter tuning as it accounts for nearly all variations in the data.

### 2.2 Multivariate Forecasting techniques

Multivariate techniques consider not only the single response variable that we want to predict but also any other external variables that could potentially have an effect on the response variable. In the context of modeling and predicting bankruptcy rate, apart from the other three external variables --- unemployment rate, population and housing price index, we can also regard “month” as an additional variable to capture the seasonality of our models. Depending on the relation between the external variables and the response variable, there are three main categories of methods.

- Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX)

  SARIMAX shares the same base idea of SARIMA, but with the ability to account for the effects of external variables. By using this model, it is assumed that the external variables influence the response but the response does not influence them.

- Vector Autoregression (VAR)

  In the vector autoregression framework, all variables are treated symmetrically since we think of them as being endogenous. In other words, the external variables and the response variable are mutually dependent.

- Vector Autoregression eXogenous (VARX)

  VARX can be seen as the combination of SARIMAX and VAR. It provides the flexibility to take some variables as endogenous and others as exogenous.

## 3 FINAL MODEL AND RESULTS

We use an ensembling modeling approach to predict the bankruptcy rate. In particular, we average the predictions of the five individual models listed above to get our final predictions.

To build an ensemble, we chose the best performing model within each model class based on test RMSE (root mean squared error). We assume that individual models have low correlation with each other. This could be justified by *what* models we ensemble and *how* we approach the modeling task. First, we blend together different time series modeling frameworks (Box- Jenkins, Holt-Winters, multivariate models like VAR). Second, we choose the best model within each class based on the independent work of each of the researcher in our group, which allows to bring some additional randomness to models.

Overall, ensembling allows to smooth the effects of over- or underfitting of each individual model and reduce generalization error (i.e. how accurately we can predict target variable values for previously unseen data).

We split the data set into the training and test sets. The test set spans the last 24 months (2009-2010), thus matching the size of the target prediction (2011-2012). Based on the exploratory analysis of the data, we decided to use two training sets: the first one covering the period from 1987 to 2008, and the second one spanning the period from 1993 to 2008.

Apart from testing different models, we also implemented different data transformations.
Below are some of the techniques we tried:
- Making log-transformation of a bankruptcy rate (which helps to reduce variation)
- Considering a variable's changes rather than levels to account for the assumption that it is dynamics of changes that might influence the target variable rather than just levels of certain variables. For example, in VAR and VARX models we included percentage change of population and change of house price index instead of the population and House Price Index levels.

Finally, our ensemble model consists of the following 5 models:

**Univariate models:**

1. Holt-Winters (Triple Exponential Smoothing, multiplicative seasonality)
2. SARIMA (2,1,4)(2,1,2)[12]

**Multivariate models:**

3. VAR(p=8)
4. VARX(p=10), endogenous variables: [bankruptcy rate, population growth, unemployment rate], exogenous variables: [house price index growth]
5. SARIMAX (0,1,13)(2,1,2)[12], exogenous variables: [unemployment rate]

Below is the table summarizing each model's characteristics and goodness of fit.

| Model | Test RMSE |
|---|---|
| Holt-Winters (Triple Exponential Smoothing, multiplicative seasonality) | 0.008507 |
| SARIMA (2,1,4)(2,1,2)[12] | 0.003825 |
| VAR (p= 8) | 0.003426 |
| VARX (p = 10), endogenous variables: [bankruptcy rate, population growth, unemployment rate], exogenous variables: [house price index growth] | 0.003198 |
| SARIMAX (0,1,13)(2,1,2)[12], exogenous variables: [unemployment rate] | 0.002530 |
| **ENSEMBLE** | **0.0034** |

Table 1. Summary of ensemble models

Averaging test predictions generated by these five models results in an RMSE of 0.0034.

Finally, we produce the forecasts for bankruptcy rate for a target period of January 2011 - December 2012.

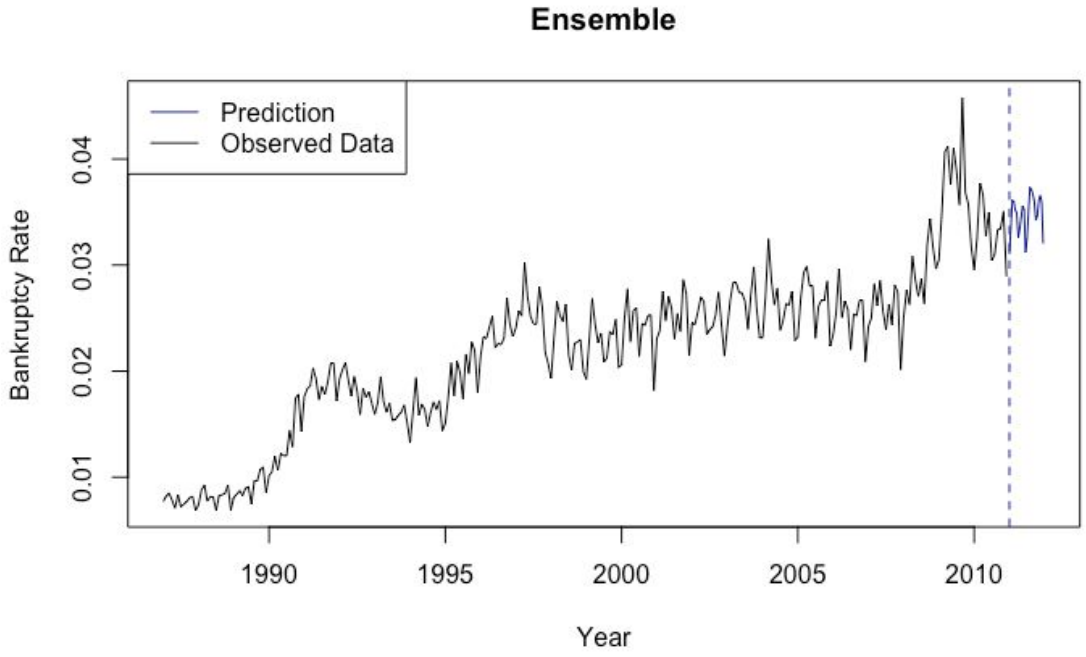



Figure 3: RESULTS ON TEST SET (2009 - 2010)

This visualization shows the ensembled predictions for January 2011 - December 2012. We can see it continues to predict and upward trend, despite the most recent drop in bankruptcy rate.

### 3.1 Model limitations

The main limitations of our ensemble model arise from the drawbacks of each of the model within the ensemble.
First, given that most of the models within the ensemble fail to satisfy all or some of the assumptions about the residuals. Those assumptions include:

1. *Residuals zero mean*, which holds for all models
2. *Constant variance (homoscedasticity)*, which fails to hold in SARIMA and VAR models
3. *No serial correlation in residuals* assumption is not satisfied in VAR and VARX models
4. *Normally distributed residuals* assumption doesn't hold in triple exponential smoothing model.

Since the assumptions do not hold, prediction intervals are not valid. However, the assumptions do not affect the model fit and point predictions. Thus, we can still rely on predictions generated by our model.

## 4 CONCLUSION

Our forecast for bankruptcy rates for January 2011 - December 2012 suggests that bankruptcy rates will trend upward. We can see from the exact predictions from the visualization of the ensembled model. From the plot we can see several spikes which correspond to monthly seasonality. However, what is more interesting is that the model predicts bankruptcy rate to continue to gradually increase, even after the recent severe drop in 2010. Most likely the additional features used in our model are not correlated with this recent spike and drop and therefore continues to predict the steady trend based on the relationship of the additional features we included in each model.

# APPENDIX

## Appendix 1: Notes on ACF, PACF

We can formally check the non-stationarity of bankruptcy rate time series by plotting the autocorrelation function (ACF) and partial autocorrelation (PACF) as in Figure 3. Autocorrelation is the linear dependence of a variable with itself at two points in time. This correlation could be due to a direct correlation between two points at lag h, but it may also be influenced by the intermediate observations. In contrast, PACF measures the same correlation but with the effect of the intermediate lags removed.

The slow exponential decay in ACF indicates a clear trend, and spikes at every twelfth point indicate monthly seasonality.

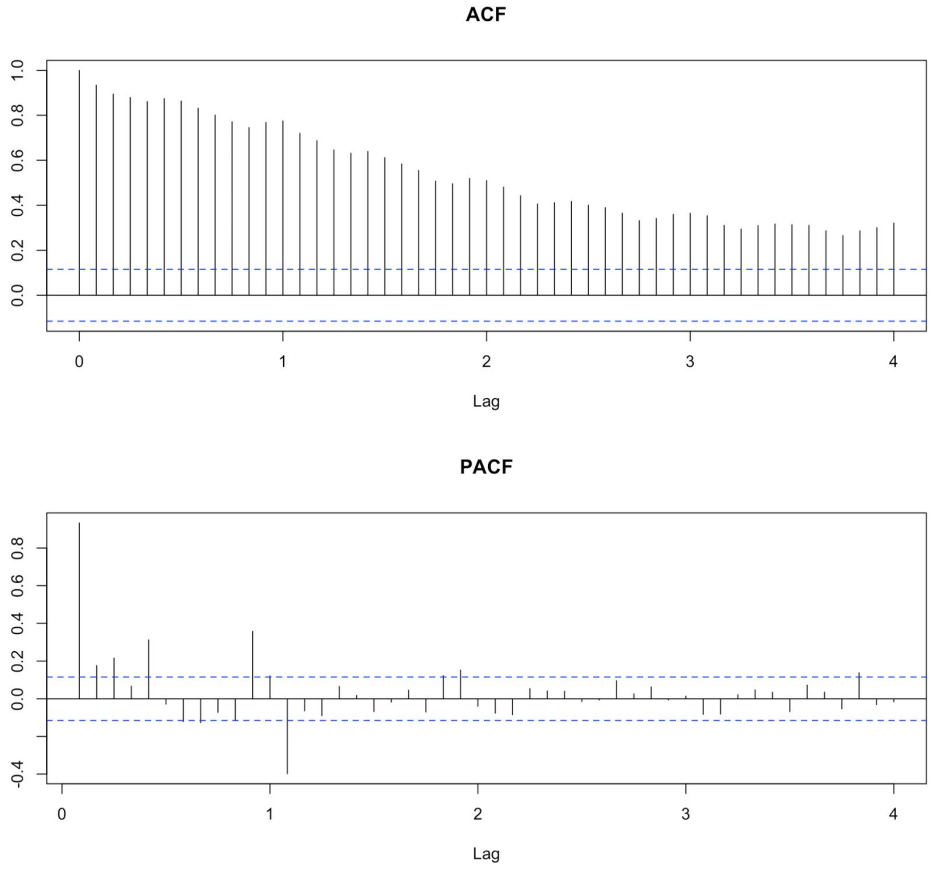



Figure A1: ACF and PACF plots of bankruptcy rate time series

**Appendix 2: Model plots**

Below are forecast plots for the five models considered:

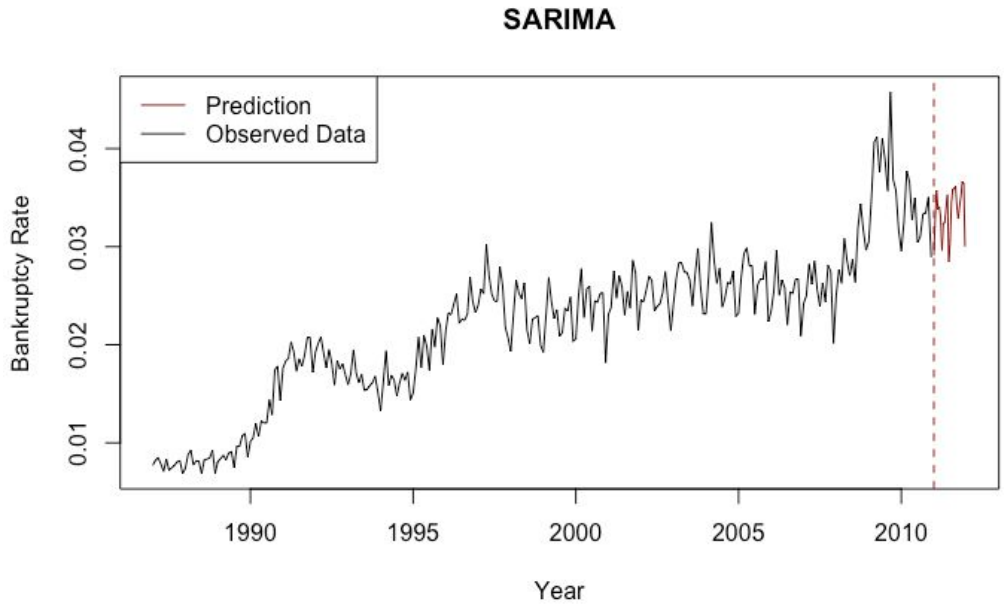



Figure A3: Predictions of SARIMA model

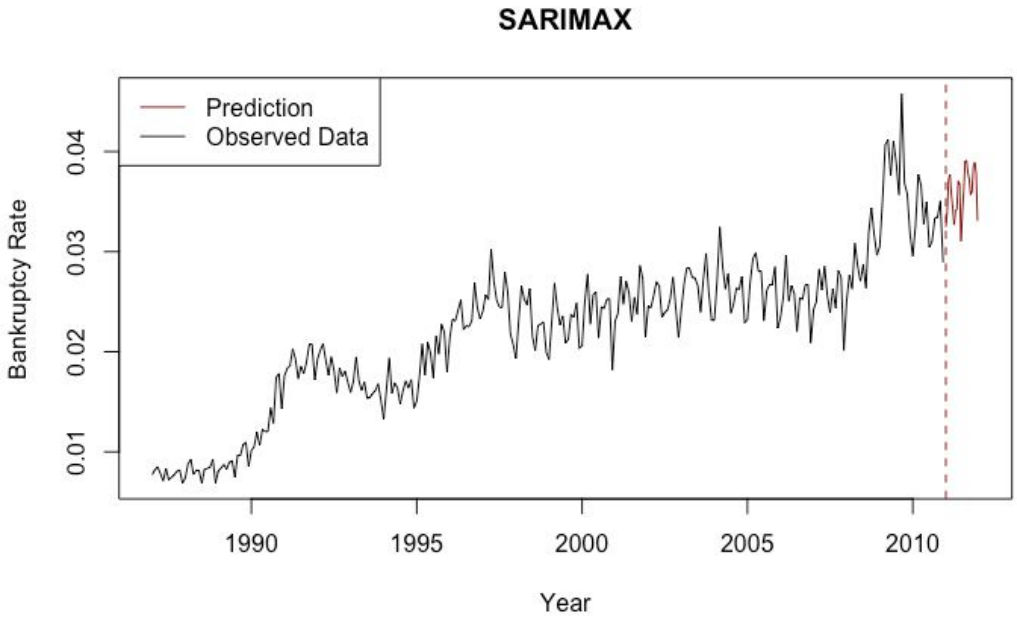



Figure A4: Predictions of SARIMAX model

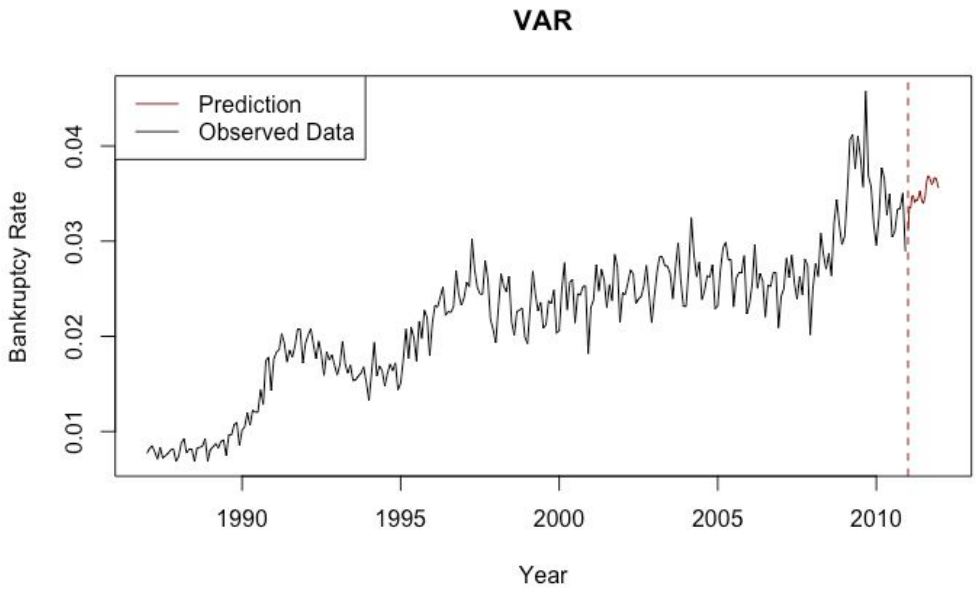



Figure A5: Predictions of VAR model

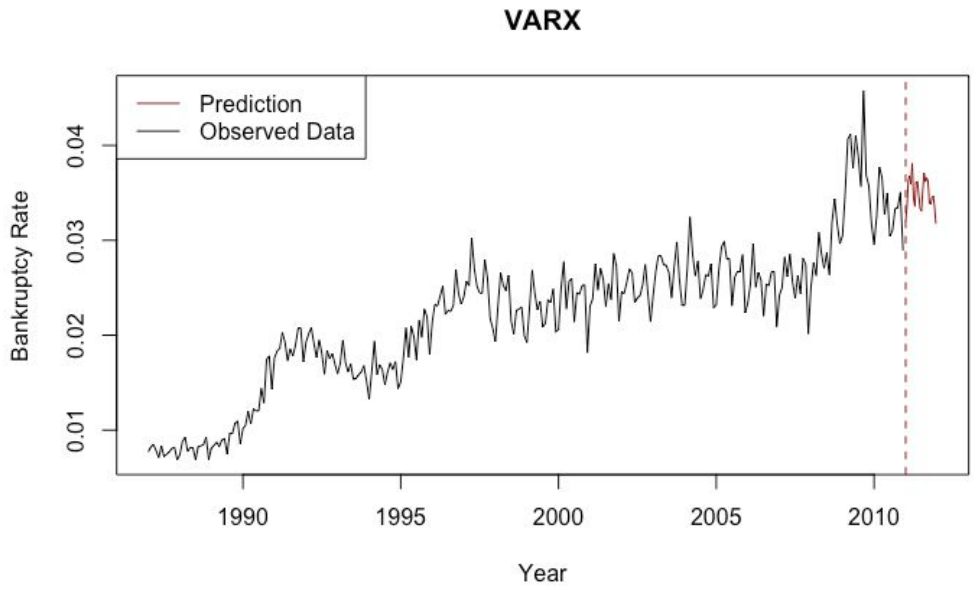



Figure A6: Predictions of VARX model

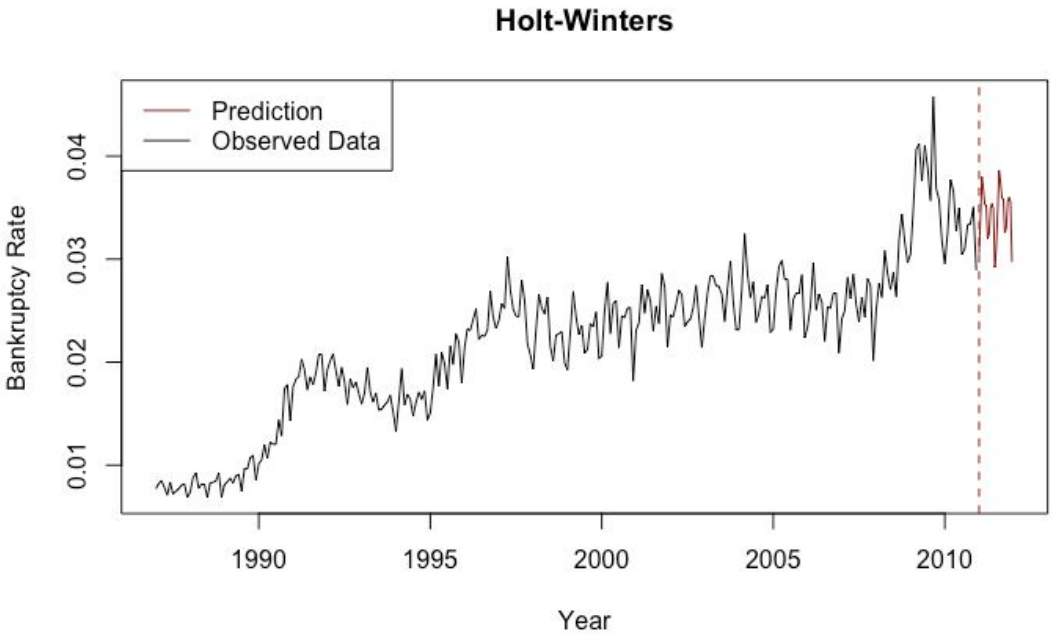



Figure A6: Predictions of Holt-Winters model

# Appendix 3: Code in R

Reading in the data and converting it to TS object:

```
df <- read.csv(file.path(path,"train.csv"), header = TRUE)

# Remove NAs and check consistency of number of rows
df <- na.omit(df)
nrow(df) == (2010-1986)*12
```

```
## [1] TRUE
```

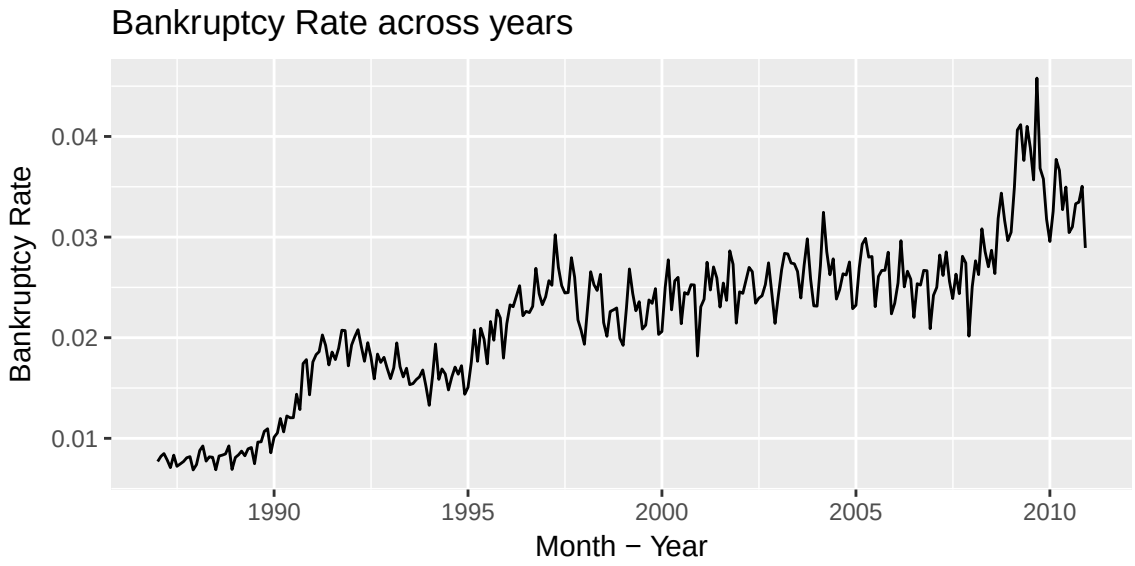
```
# Convert to TS object
BR_ts <- ts(df['Bankruptcy_Rate'], start = c(1987,1), end = c(2010, 12), frequency = 12)
```

## Exploratory Data Analysis

Plotting Dependent variable: Bankruptcy Rate

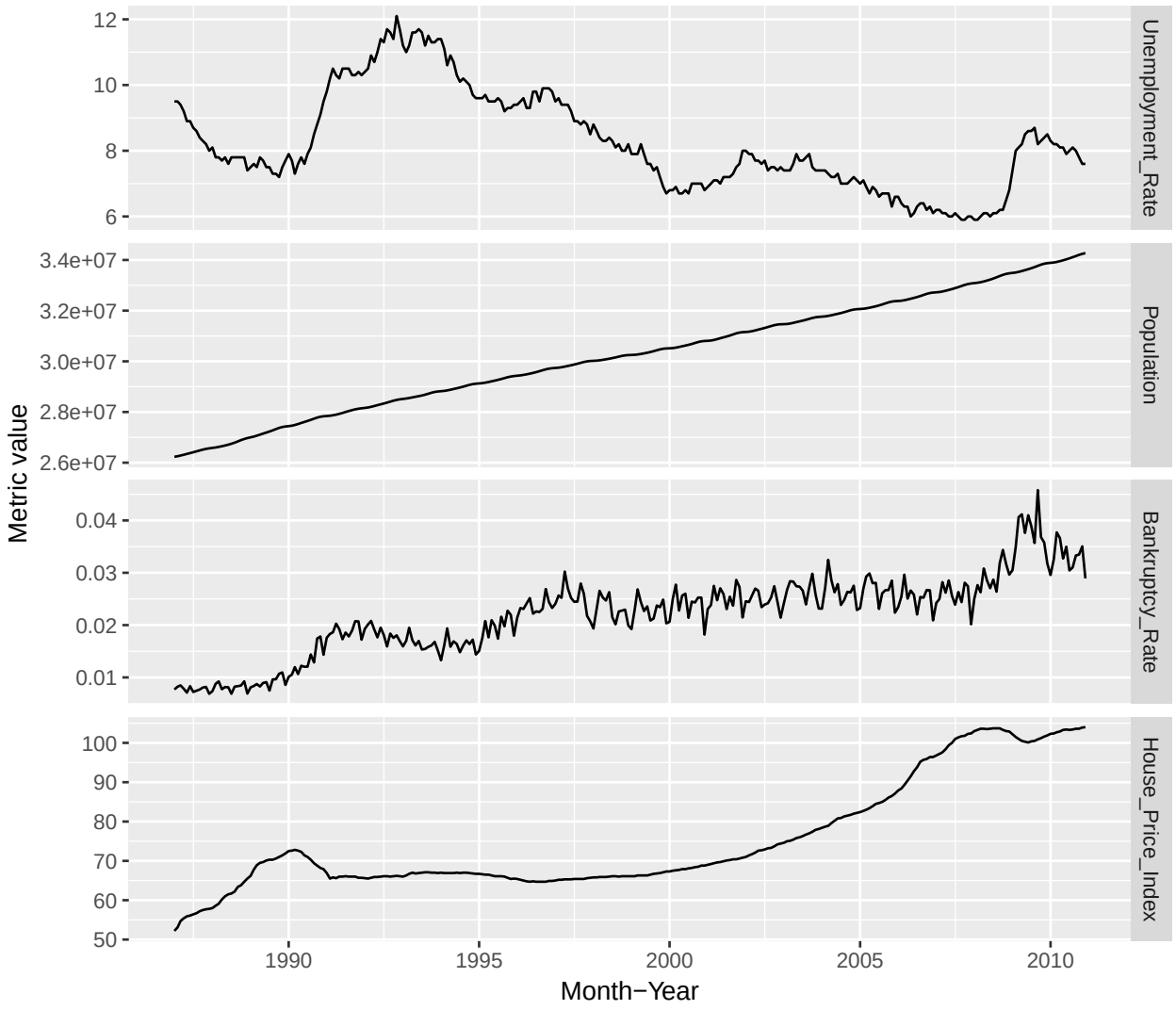
```
# Plotting TS data
ggplot() + geom_line(data=data.frame(BR_ts), aes(as.Date(as.yearmon(time(BR_ts))),BR_ts)) +
  labs(y="Bankruptcy Rate") +
  labs(x="Month - Year") +
  labs(title="Bankruptcy Rate across years")
```

```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




Plotting all predictors across time:

```
df2 <- df
df2[1] <- seq(as.Date("1987/01/01"), by = "month", length.out = 288)
df2_melt <- melt(df2, id.vars=c("Month"))
ggplot() + geom_line(data=df2_melt, aes(x=Month, y=value)) + facet_grid(variable ~ .,  scales = "free")
```




## Model Building

Split into train and test:

```
train_1987 <- window(BR_ts, start=c(1987,1), end=c(2008,12))
train_1993 <- window(BR_ts, start=c(1993,1), end=c(2008,12))

test <- window(BR_ts, start=c(2009,1), end=c(2010,12))
```

### SARIMA (2,1,4)(2,1,2)[12]

```
m_sarima <- arima(log(train_1993), order = c(2,1,4), seasonal = list(order = c(2,1,2), period = 12), me
sarima_fc <- forecast(m_sarima, h = 24)
rmse.sarima <- sqrt(mean((exp(sarima_fc$mean) - test)^2))
rmse.sarima
```

```
## [1] 0.003842615
```

### Multiplicative Holt-Winters model

```
m_hw_mult <- HoltWinters(train_1993, seasonal = "mult")
hw_mult_fc <- forecast(m_hw_mult, 24, prediction.interval = TRUE)
rmse_hw_mult<-  sqrt(mean((hw_mult_fc$mean - test)^2))
rmse_hw_mult
```

```
## [1] 0.00850725
```

### SARIMAX

```
m_sarimax <- arima(log(train_1993), order = c(0,1,13), seasonal = list(order = c(2,1,2), period = 12), x
sarimax_fc <-  predict(m_sarimax, n.ahead = 24, newxreg = df2[df2['Month'] >= '2010-01-01' & df2['Month
sqrt(mean((exp(sarimax_fc$pred) - test)^2))
```

```
## [1] 0.002575159
```

### VARX

```
data_ts <- ts(df ,frequency = 12, start = c(1987,1,19), end = c(2010,12) )
b_rate<- ts(df$Bankruptcy_Rate, frequency = 12, start = c(1987,1), end = c(2010,12))
unemp_rate<-ts(df$Unemployment_Rate, frequency = 12, start = c(1987,1))
data_ts <- ts(df,frequency = 12, start = c(1987,1,19), end = c(2010,12) )
population <- ts(df$Population, frequency = 12, start = c(1987,1,19))
house_price_ind <- ts(df$House_Price_Index, frequency = 12, start = c(1987,1,19))

# Training set 1
train1.data <- window(data_ts, frequency = 12, start = c(1987,1), end = c(2008,12))
train1.b_rate <- window(b_rate, frequency = 12, start = c(1987,1), end = c(2008,12))
train1.unemp_rate <- window(unemp_rate, frequency = 12, start = c(1987,1), end = c(2008,12))
train1.pop <- window(population, frequency = 12, start = c(1987,1), end = c(2008,12))
train1.hpi <- window(house_price_ind, frequency = 12, start = c(1987,1), end = c(2008,12))

# Training set 2
train2.data <- window(data_ts, frequency = 12, start = c(1993,1), end = c(2008,12))
train2.b_rate <- window(b_rate, frequency = 12, start = c(1993,1), end = c(2008,12))
train2.unemp_rate <- window(unemp_rate, frequency = 12, start = c(1993,1), end = c(2008,12))
train2.pop <- window(population, frequency = 12, start = c(1993,1), end = c(2008,12))
train2.hpi <- window(house_price_ind, frequency = 12, start = c(1993,1), end = c(2008,12))

# Test set
test.data <- window(data_ts, frequency = 12, start = c(2009,1), end = c(2010,12))
test.b_rate <- window(b_rate, frequency = 12, start = c(2009,1), end = c(2010,12))
test.unemp_rate <- window(unemp_rate, frequency = 12, start = c(2009,1), end = c(2010,12))
test.pop <- window(population, frequency = 12, start = c(2009,1), end = c(2010,12))
test.hpi <- window(house_price_ind, frequency = 12, start = c(2009,1), end = c(2010,12))
```

Creating the growth change variables:

```
#Find population growth
pop_growth <- population/lag(population,-1) -1
train1.pop_growth <- window(pop_growth, start = c(1987,2), end = c(2008,12))
```

```
train2.pop_growth <- window(pop_growth, start = c(1993,2), end = c(2008,12))
test.pop_growth <- window(pop_growth, start = c(2009,1), end = c(2010,12))

# Find HPI change
hpi_growth <- house_price_ind/lag(house_price_ind,-1) -1
train1.hpi_growth <- window(hpi_growth, start = c(1987,2), end = c(2008,12))
train2.hpi_growth <- window(hpi_growth, start = c(1993,2), end = c(2008,12))
test.hpi_growth <- window(hpi_growth, start = c(2009,1), end = c(2010,12))

# create a data frame
y <- data.frame(window(train2.b_rate, start = c(1993,2), end = c(2008,12)),
                window(train2.unemp_rate/100, start = c(1993,2), end = c(2008,12)),
                train2.hpi_growth, train2.pop_growth)

colnames(y)<- c("train2.b_rate", "train2.unemp_rate", "train2.hpi_growth", "train2.pop_growth")
```

```
fit_var_8 <- VAR(y, p = 8, type = "both")
pred.var_8 <- predict(fit_var_8, n.ahead = 24, ci = 0.95)
sqrt(mean((pred.var_8$fcst$train2.b_rate[,1] - test.b_rate)^2))
```

```
## [1] 0.003426352
```

### VAR

```
m.varx <- VAR(data.frame(y$train2.b_rate,y$train2.unemp_rate,y$train2.pop_growth), exogen = data.frame(

#Make predictions on a test set

pred.varx <- predict(m.varx, n.ahead = 24, ci = 0.95, dumvar = data.frame( X1 = test.hpi_growth))
rmse.varx <- sqrt(mean((pred.varx$fcst$y.train2.b_rate[,1] - test.b_rate)^2))
rmse.varx
```

```
## [1] 0.003198469
```

## Testing model assumptions

Below model assumptions are checked for the SARIMAX model. The same has been repeated on other 4 models as well.

### (i) Test for Zero-Mean: t test

```
e <- m_sarimax$residuals
t.test(e)
```

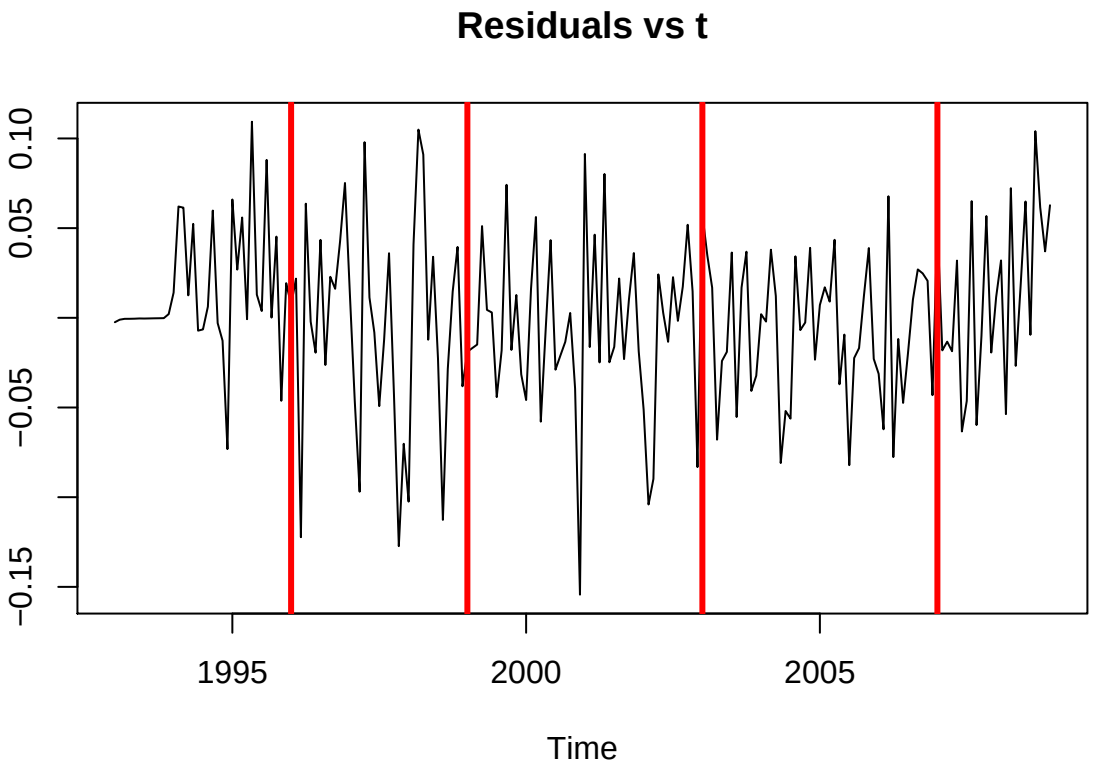
```

    One Sample t-test

data:  e
t = 0.082274, df = 191, p-value = 0.9345
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
```

```
 -0.006359743  0.006913381
sample estimates:
   mean of x
0.0002768191
```

Clearly p-value is high considering a significance level of 0.05. Hence we cannot reject the null hypothesis and the residuals have zero mean. This assumption is satisfied.

### (ii) Homoscedasticity

```
par(mfrow=c(1,1))
plot(e, main="Residuals vs t", ylab="")
abline(v=c(1996,1999,2003,2007), lwd=3, col="red")
```




```
group <- c(rep(1,40),rep(2,40),rep(3,40),rep(4,40), rep(5,32))
levene.test(exp(e), group)#Levene
```

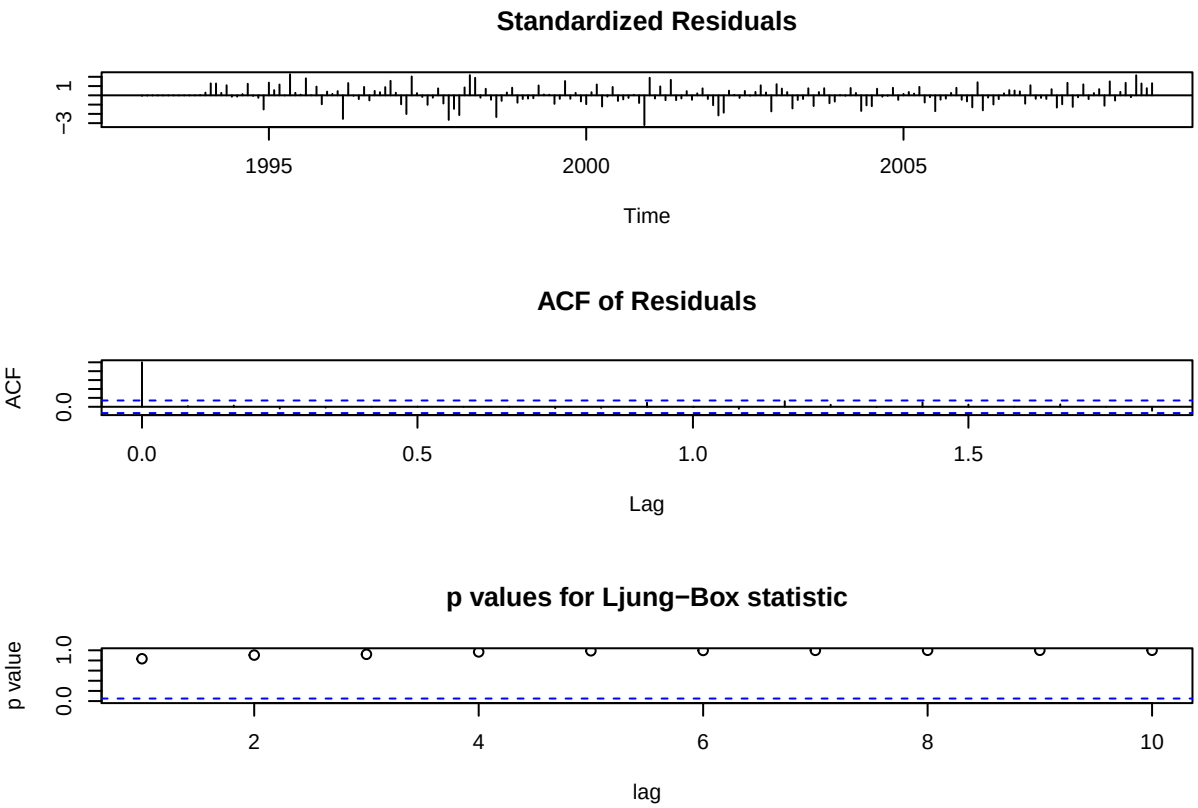
```
Warning: 'levene.test' is deprecated.
Use 'leveneTest' instead.
See help("Deprecated") and help("car-deprecated").
```

```
Warning in leveneTest.default(...): group coerced to factor.
```

```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   4   1.276 0.2809
      187
```

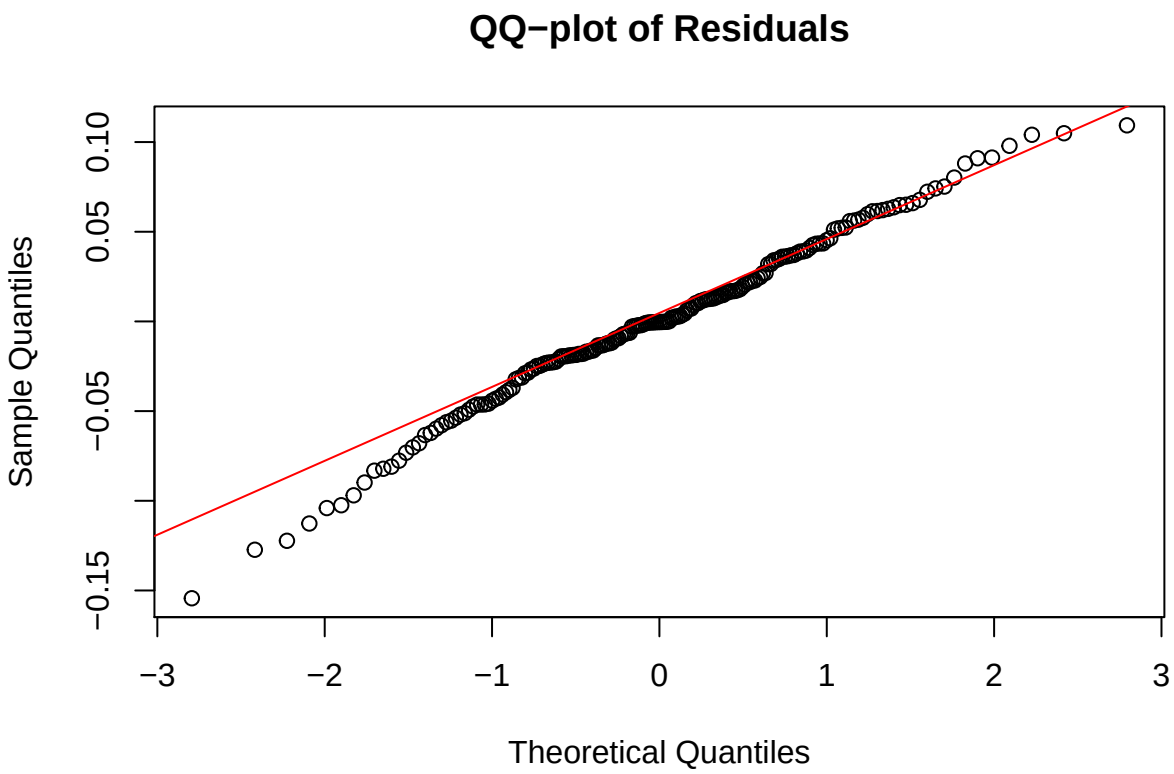
### (iii) Zero-Correlation

```
tsdiag(m_sarimax)
```




### (iv) Normality

```
par(mfrow=c(1,1))
qqnorm(e, main="QQ-plot of Residuals")
qqline(e, col = "red")
```




```
shapiro.test(e) #SW test
```

```
    Shapiro-Wilk normality test

data:  e
W = 0.98833, p-value = 0.1166
```