Canadian National Bankruptcy Rates Forecasting

Anant Agarwal, Devin Bowers, Fei Liu, Cara Qin

12/8/2017

```
data <- read.csv("/Users/xiaohui/Documents/0_2017_USF/MSAN_604_TS/Final project/train.csv", header = TRU
data <- data[which(is.na(data['Month']) == 0),] #remove blank lines at bottom
test <- read.csv("/Users/xiaohui/Documents/0_2017_USF/MSAN_604_TS/Final project/test.csv") # 2011, 2012
```

Split data to training(1987-2008) and validation(2009 and 2010)

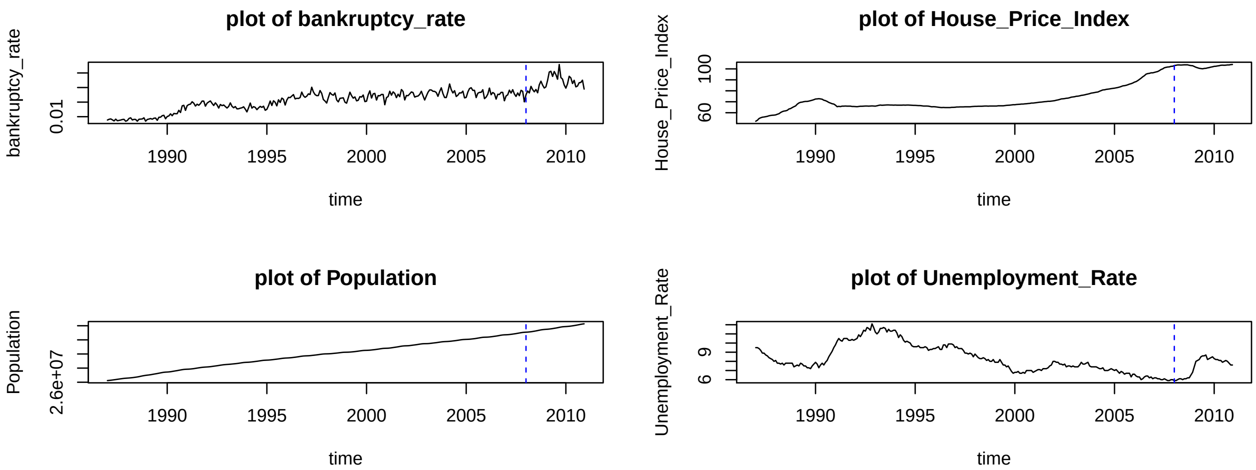
```
n = nrow(data)
train <- data[1:(n-24),] # 1987-2008
valid <- data[(n-23):n,] # 2009-2010
```

Display summary statistics, correlation, and plot

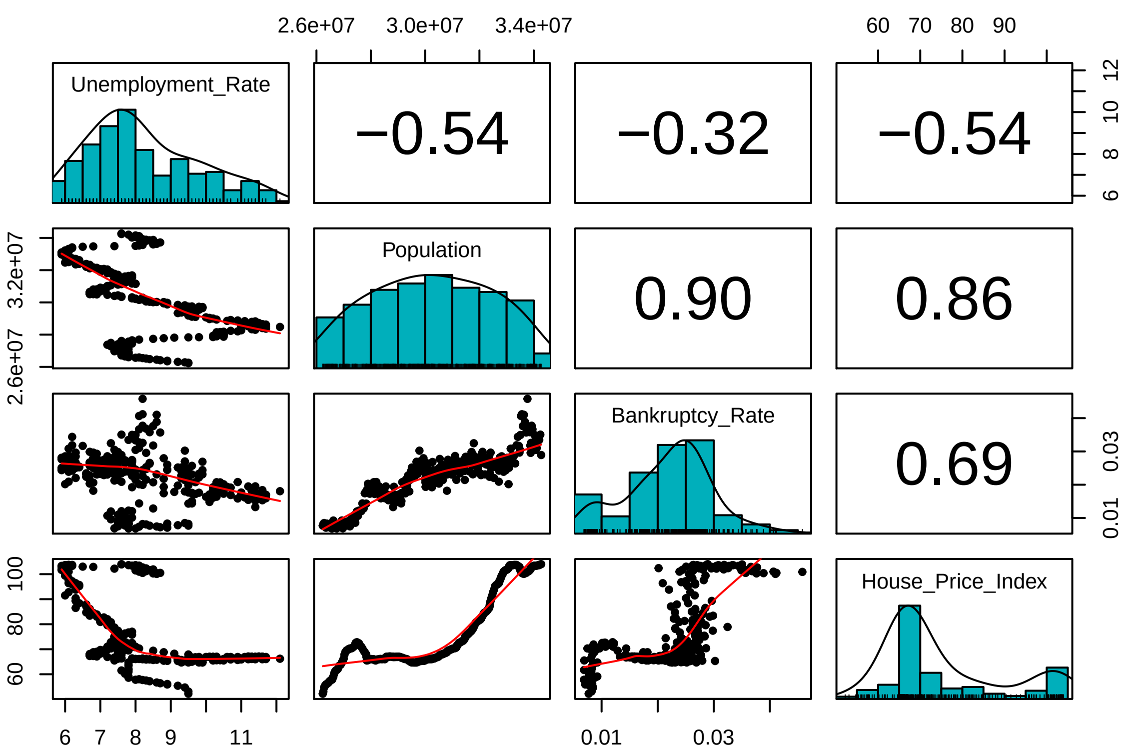
```
summary(data)
```

```
##      Month        Unemployment_Rate   Population        Bankruptcy_Rate
##  Min.   : 11987   Min.   : 5.900    Min.   :26232423   Min.   :0.006862
##  1st Qu.: 39493   1st Qu.: 7.175    1st Qu.:28511929   1st Qu.:0.017277
##  Median : 66998   Median : 7.900    Median :30248741   Median :0.023127
##  Mean   : 66998   Mean   : 8.236    Mean   :30256218   Mean   :0.021904
##  3rd Qu.: 94504   3rd Qu.: 9.400    3rd Qu.:32059937   3rd Qu.:0.026620
##  Max.   :122010   Max.   :12.100    Max.   :34272214   Max.   :0.045798
##  House_Price_Index
##  Min.   : 52.20
##  1st Qu.: 66.00
##  Median : 68.30
##  Mean   : 75.22
##  3rd Qu.: 82.25
##  Max.   :104.00
```

```
par(mfrow=c(2,2))
plot( ts(data$Bankruptcy_Rate, start = c(1987,1), frequency = 12), main = "plot of bankruptcy_rate", yla
abline(v=2008,col='blue',lty=2)
plot( ts(data$House_Price_Index, start = c(1987,1), frequency = 12), main = "plot of House_Price_Index"
abline(v=2008,col='blue',lty=2)
plot( ts(data$Population, start = c(1987,1), frequency = 12), main = "plot of Population", ylab = "Popul
abline(v=2008,col='blue',lty=2)
plot( ts(data$Unemployment_Rate, start = c(1987,1), frequency = 12), main = "plot of Unemployment_Rate"
abline(v=2008,col='blue',lty=2)
```

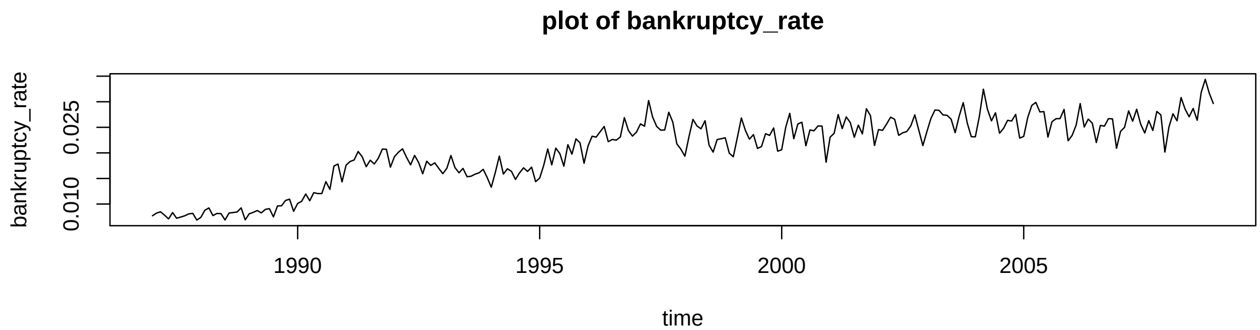


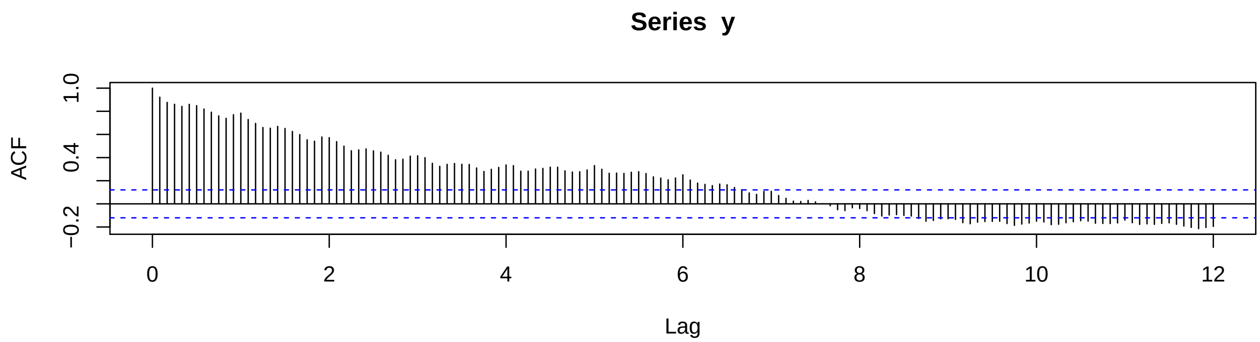

```r
pairs.panels(data[,-1],
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
            #density = TRUE,  # show density plots
             ellipses = FALSE # show correlation ellipses
             )
```



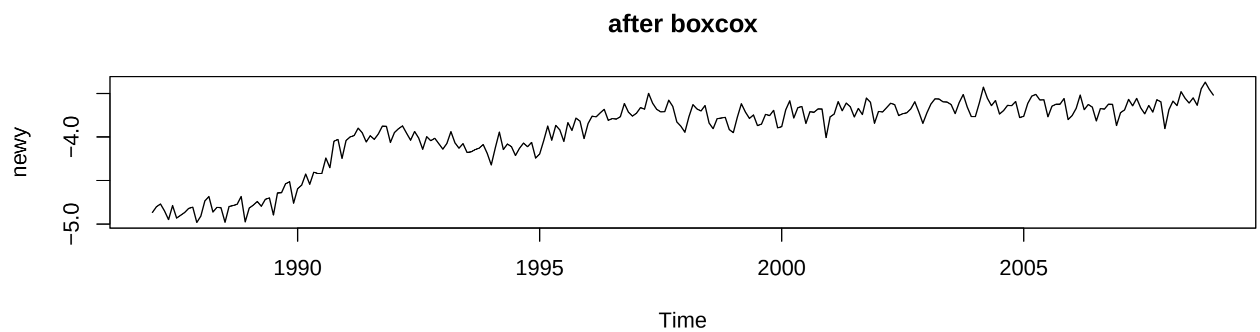
Data Exploration and transformation: take log; take one ordinary differencing and one seasonal differencing

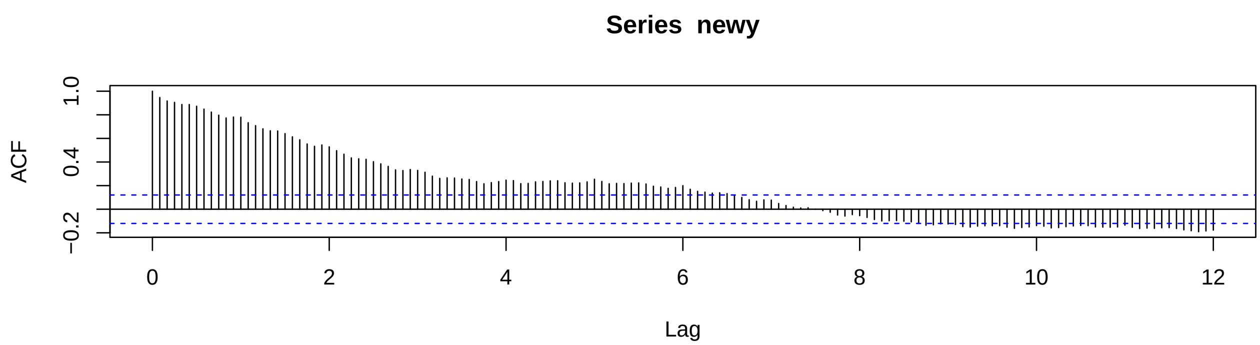
```r
y <- ts(train$Bankruptcy_Rate, start = c(1987,1), frequency = 12)
par(mfrow=c(2,1))
plot(y, main = "plot of bankruptcy_rate", ylab = "bankruptcy_rate", xlab = "time")
acf(y, lag.max = 144)
```






```
#take log of y for having constant variance
newy <- log(y)
plot(newy, main="after boxcox")
acf(newy, lag.max = 144)
```





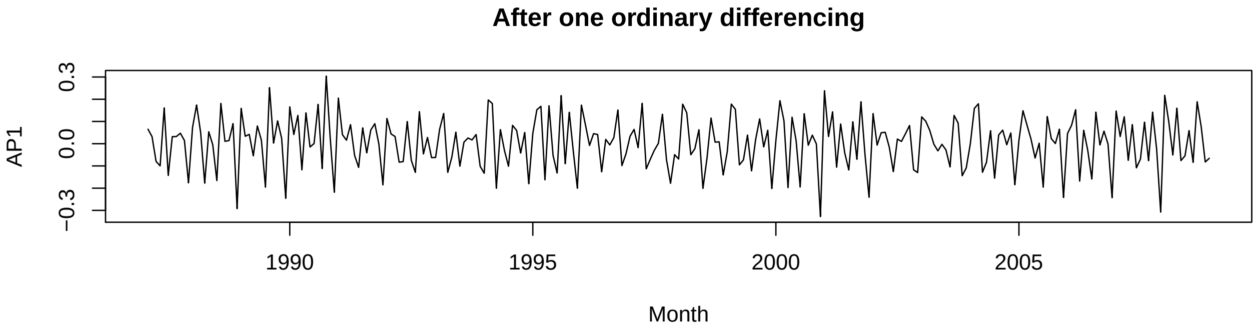

```
par(mfrow=c(2,1))
#take one ordinary differencing to remove trend
AP1 <- diff(newy)
```

```r
plot(AP1, ylab = "AP1",xlab = "Month", main="After one ordinary differencing")
adf.test(AP1)
```

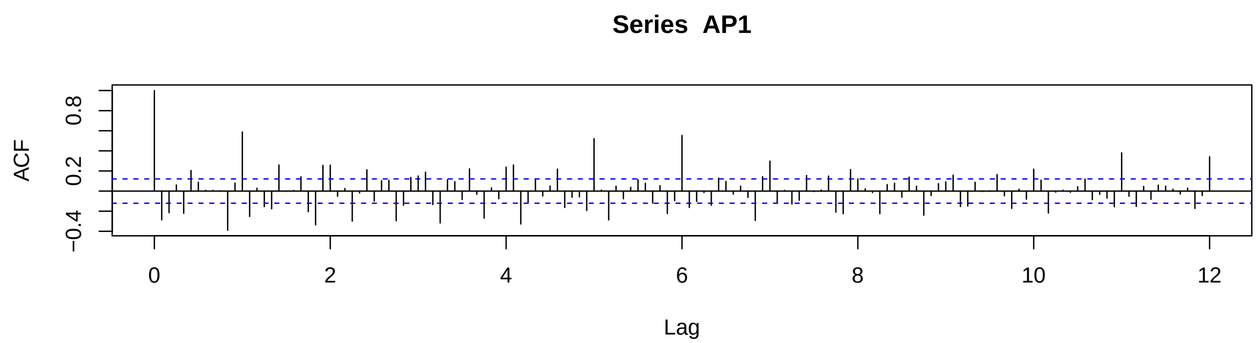
```
## Warning in adf.test(AP1): p-value smaller than printed p-value
```

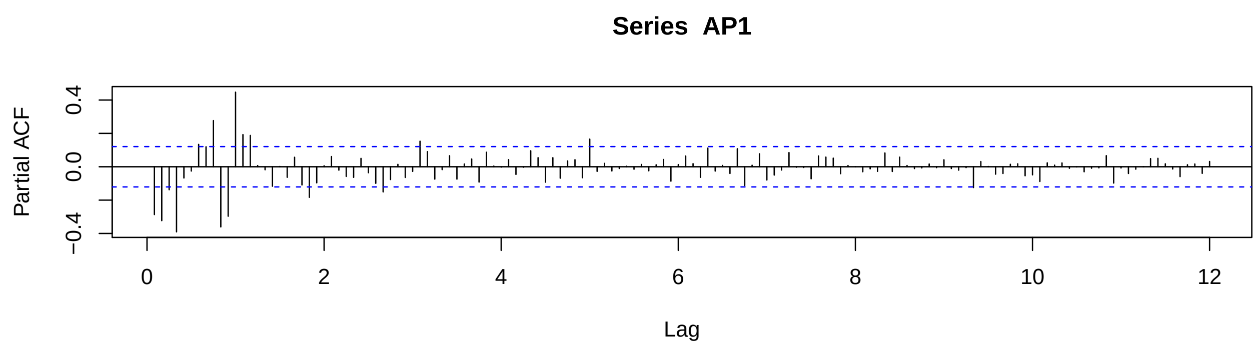
```
##
##  Augmented Dickey-Fuller Test
##
## data:  AP1
## Dickey-Fuller = -7.3423, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
par(mfrow=c(2,1))
```




```r
acf(AP1, lag.max = 144)
pacf(AP1, lag.max = 144)
```

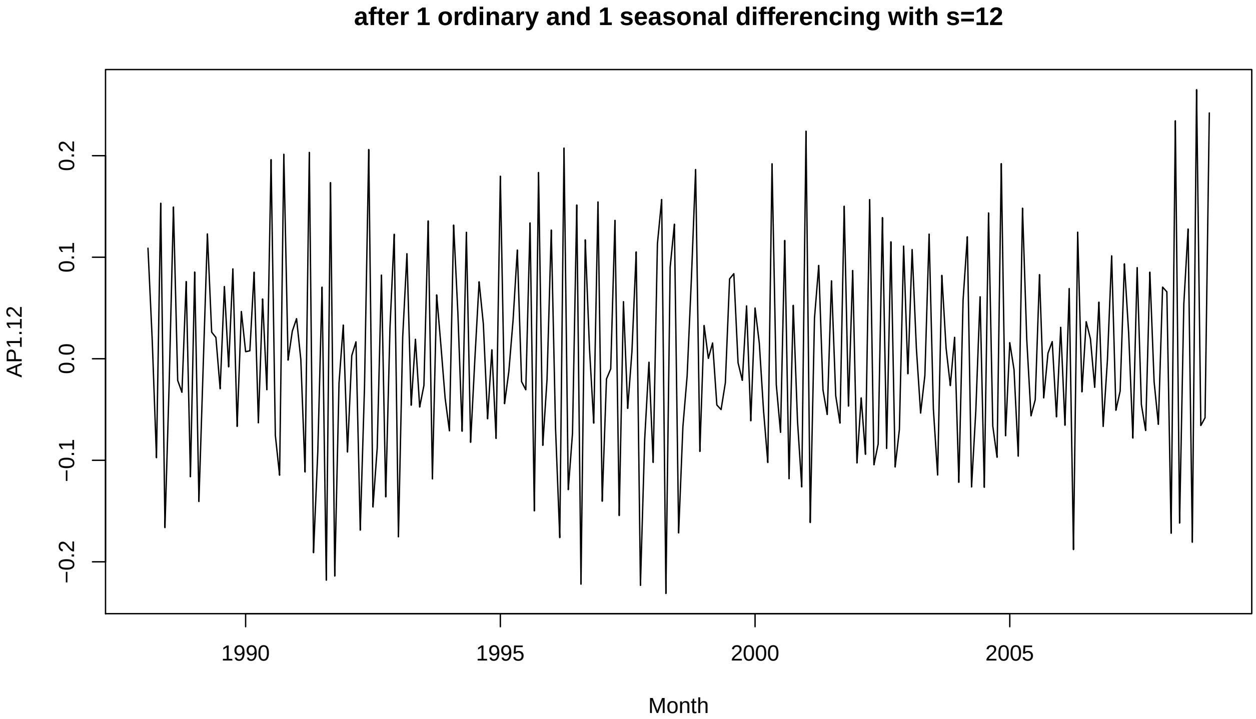





```r
#Take one seasonal differencing.
nsdiffs(AP1)
```

```
## [1] 0
```

```r
AP1.12 <- diff(AP1, lag=12)
plot(AP1.12, ylab = "AP1.12",xlab = "Month", main="after 1 ordinary and 1 seasonal differencing with s=
```

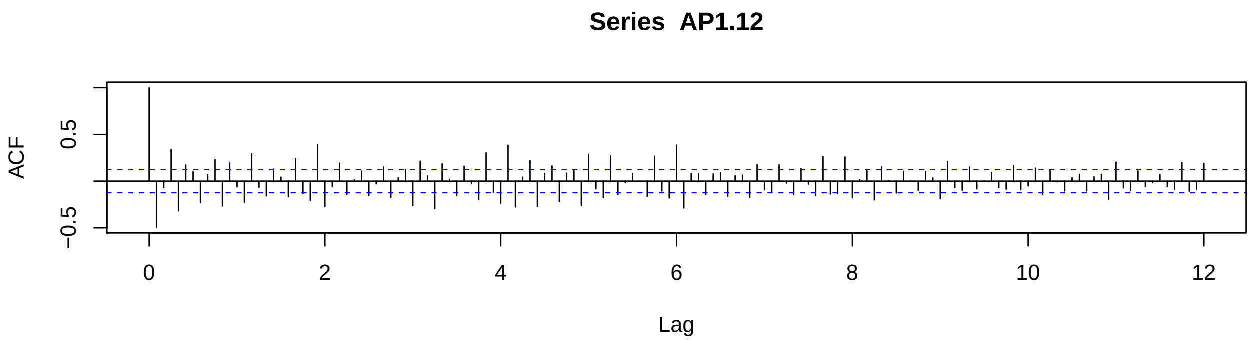


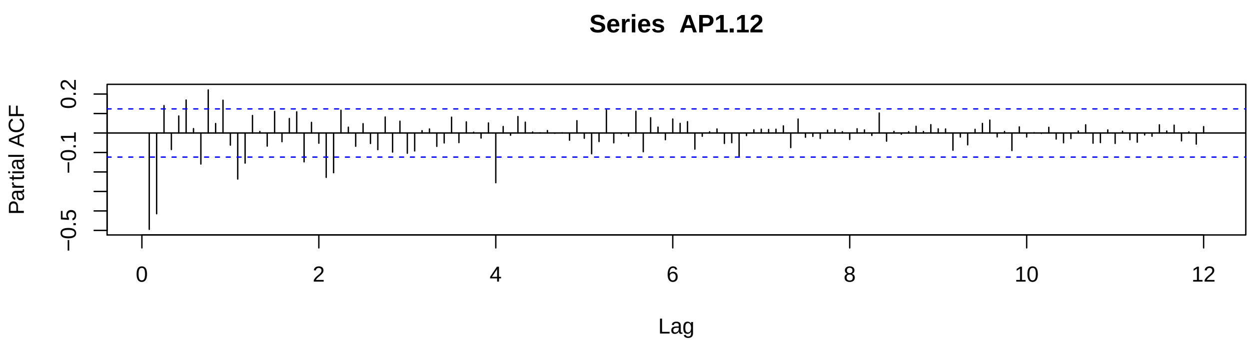

```r
adf.test(AP1.12)
```

```
## Warning in adf.test(AP1.12): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  AP1.12
## Dickey-Fuller = -5.0331, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
par(mfrow=c(2,1))
acf(AP1.12, lag.max = 144)
pacf(AP1.12, lag.max = 144)
```





Look at ACF and PACF plot to try p, q, P, Q which p<=3, q ~ exponential decay or q <=3, P <=3, Q ~ exponential decay or Q<=3.

Modeling Approach 1: Univariate SARIMA on Backruptcy (log) exhaustive search

```
y.train <- newy#define response variable in training
y.test <- valid$Bankruptcy_Rate#define response variable in validation
```

```
maxp <- 3
perm <- expand.grid(p = seq(0,maxp), q = seq(0,maxp), P = seq(0,maxp), Q = seq(0,maxp))
perm['df'] <- perm[1]+perm[2]+perm[3]+perm[4]
perm.sort <- perm[order(perm['df']),]

train.rmse <- rep(0,nrow(perm.sort))
test.rmse <- rep(0,nrow(perm.sort))
train.aic <- rep(0,nrow(perm.sort))
train.sigma<-rep(0,nrow(perm.sort))
train.loglik <- rep(0,nrow(perm.sort))

for (i in 1:nrow(perm.sort)){

  model <- tryCatch(arima(y.train, order = c(perm.sort[i,'p'], 1, perm.sort[i,'q']), seasonal = list(ord

  #if MLE is converged, append results, otherwise set results to NA.
  if ( (model[1] == "NC") == FALSE){
  #training rmse
  fitted <- y.train - model$residuals
  tr.rmse <- sqrt(mean((exp(fitted) - exp(y.train))^2))
```

```
  #test rmse
  yhat <- forecast(object = model, h=24, level = 0.95)
  te.rmse <- sqrt(mean((exp(yhat$mean) - y.test)^2))

  #Save training rmse, aic, sigma, loglik; test rmse
  train.rmse[i] <- tr.rmse
  test.rmse[i] <-te.rmse
  train.aic[i] <- model$aic
  train.sigma[i] <- model$sigma2
  train.loglik[i] <- model$loglik} else{

  print (paste0("MLE in model ", i, "is not converged", sep=" "))
  train.rmse[i] <- NA
  test.rmse[i] <- NA
  train.aic[i] <- NA
  train.sigma[i] <- NA
  train.loglik[i] <- NA}
 }

models.result <- data.frame(perm.sort, train.aic, train.sigma, train.loglik, train.rmse, test.rmse)

#plot test.rmse & train aic in the same plot
par(mfrow=c(2,2))
plot(models.result$train.aic, type="l", main="aic")
plot(models.result$train.sigma, type="l", main="sigma")
#plot(models.result$train.loglik, type="l", main="loglk")
plot(models.result$test.rmse, type="l", main="test rmse")
plot(models.result$train.rmse, type="l", main="train rmse")

setwd('/Users/xiaohui/Documents/0_2017_USF/MSAN_604_TS/Final project')
write.csv(models.result, file = "models.result_p3.csv")
```

Loglikelihood ratio test to compare models

```
#Function to perform log-likelihood ratio test
myLRT <- function(m1, m2){
  D <- -2*(m1$loglik - m2$loglik)
  pval <- 1-pchisq(D,length(m2$coef) - length(m1$coef))
  print(c("Test Statistic:",round(D, 4),"P-value:", round(pval, 4)))
}
```

```
#Conduct likelihood ratio test
#Compare models with TEST RMSE <0.004 AND training AIC<-650
#models.result <- read.csv("models.result_p3.csv")
#lrt.modesl <- models.result[which(models.result$train.aic < -650 & models.result$test.rmse < 0.004), ]

#list of comparable models
m.df6 <- arima(y.train, order = c(0,1,1), seasonal = list(order = c(3,1,2), period = 12), method = "CSS
m.df7 <- arima(y.train, order = c(0,1,1), seasonal = list(order = c(3,1,3), period = 12), method = "CSS
m.df8 <- arima(y.train, order = c(0,1,2), seasonal = list(order = c(3,1,3), period = 12), method = "CSS
m.df9 <- arima(y.train, order = c(1,1,2), seasonal = list(order = c(3,1,3), period = 12), method = "CSS
m.df10 <- arima(y.train, order = c(1,1,3), seasonal = list(order = c(3,1,3), period = 12), method = "CS
```

```
## Warning in log(s2): NaNs produced
```

```r
m.df11 <- arima(y.train, order = c(3,1,3), seasonal = list(order = c(2,1,3), period = 12), method = "CSS
m.df12 <- arima(y.train, order = c(3,1,3), seasonal = list(order = c(3,1,3), period = 12), method = "CSS
m.df82 <- arima(y.train, order = c(0,1,3), seasonal = list(order = c(2,1,3), period = 12), method = "CSS
m.df72 <- arima(y.train, order = c(0,1,2), seasonal = list(order = c(2,1,3), period = 12), method = "CSS
```

```r
myLRT(m.df12, m.df11)
```

```
## Warning in pchisq(D, length(m2$coef) - length(m1$coef)): NaNs produced

## [1] "Test Statistic:" "16.1796"         "P-value:"        "NaN"
```

```r
myLRT(m.df82, m.df11)
```

```
## [1] "Test Statistic:" "5.8176"          "P-value:"        "0.1208"
```

```r
myLRT(m.df72, m.df82)
```

```
## [1] "Test Statistic:" "4.0648"          "P-value:"        "0.0438"
```

```r
myLRT(m.df9, m.df10)
```

```
## [1] "Test Statistic:" "3.7136"          "P-value:"        "0.054"
```

```r
myLRT(m.df9, m.df11)
```

```
## [1] "Test Statistic:" "6.184"           "P-value:"        "0.0454"
```

```r
myLRT(m.df8, m.df9)
```

```
## [1] "Test Statistic:" "3.5512"          "P-value:"        "0.0595"
```

```r
myLRT(m.df7, m.df8)
```

```
## [1] "Test Statistic:" "2.909"           "P-value:"        "0.0881"
```

```r
myLRT(m.df6, m.df7)
```

```
## [1] "Test Statistic:" "20.462"          "P-value:"        "0"
```

```r
myLRT(m.df7, m.df9)
```

```
## [1] "Test Statistic:" "6.4603"          "P-value:"        "0.0396"
```

Optimal model for Univariate Sarima

```r
#optimal.sarima <- arima(y.train, order = c(1,1,2), seasonal = list(order = c(3,1,3), period = 12), met
#optimal.sarima <- arima(y.train, order = c(0,1,3), seasonal = list(order = c(2,1,3), period = 12), met
optimal.sarima <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), meth
#optimal.sarima <- arima(y.train, order = c(1,1,1), seasonal = list(order = c(2,1,2), period = 12), met

yhat <- forecast(object = optimal.sarima, h=24, level = 0.95) #predicted test
te.rmse <- sqrt(mean((exp(yhat$mean) - y.test)^2)) #test rmse
te.rmse
```

```
## [1] 0.003826419
```

```r
optimal.sarima$loglik
```

```
## [1] 349.2863
```

```
optimal.sarima$aic
```

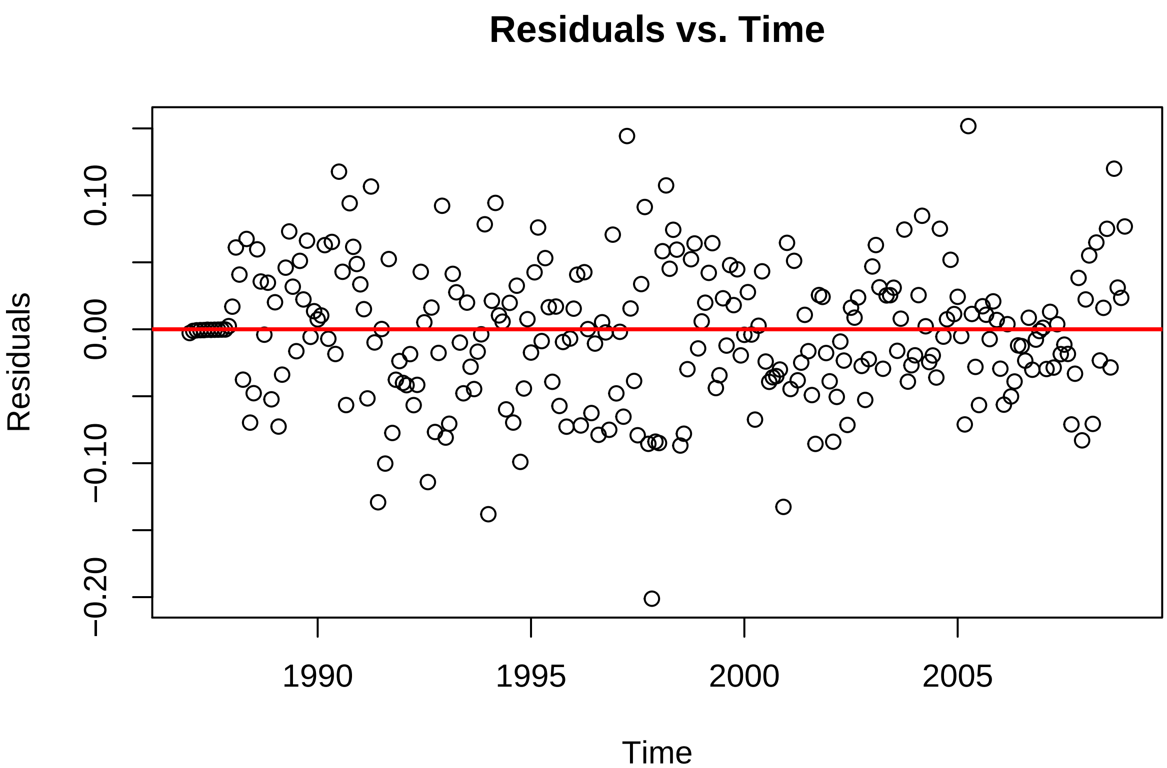
```
## [1] -680.5727
```

SARIMA Model residual diagnostic

```
e <- optimal.sarima$residuals
####(1) test whether residuals have zero mean
t.test(e)
```

```
##
##  One Sample t-test
##
## data:  e
## t = -0.79453, df = 263, p-value = 0.4276
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.008744887  0.003716524
## sample estimates:
##    mean of x
## -0.002514182
```

```
####(2) test heteroscedasticity
plot(e, main = "Residuals vs. Time", ylab = "Residuals", xlab = "Time", type='p') # plotting the residu
abline(h = 0, col = "red", lwd = 2) # plotting a horizontal line at 0
```




```
group <- cut(1:length(e), breaks=4, labels=(1:4))
leveneTest(e,group) #Levene
```

```
## Levene's Test for Homogeneity of Variance (center = median)
```

```
##        Df F value  Pr(>F)
## group   3  2.3711 0.07091 .
##       260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
bartlett.test(e,group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 7.3354, df = 3, p-value = 0.06194
```

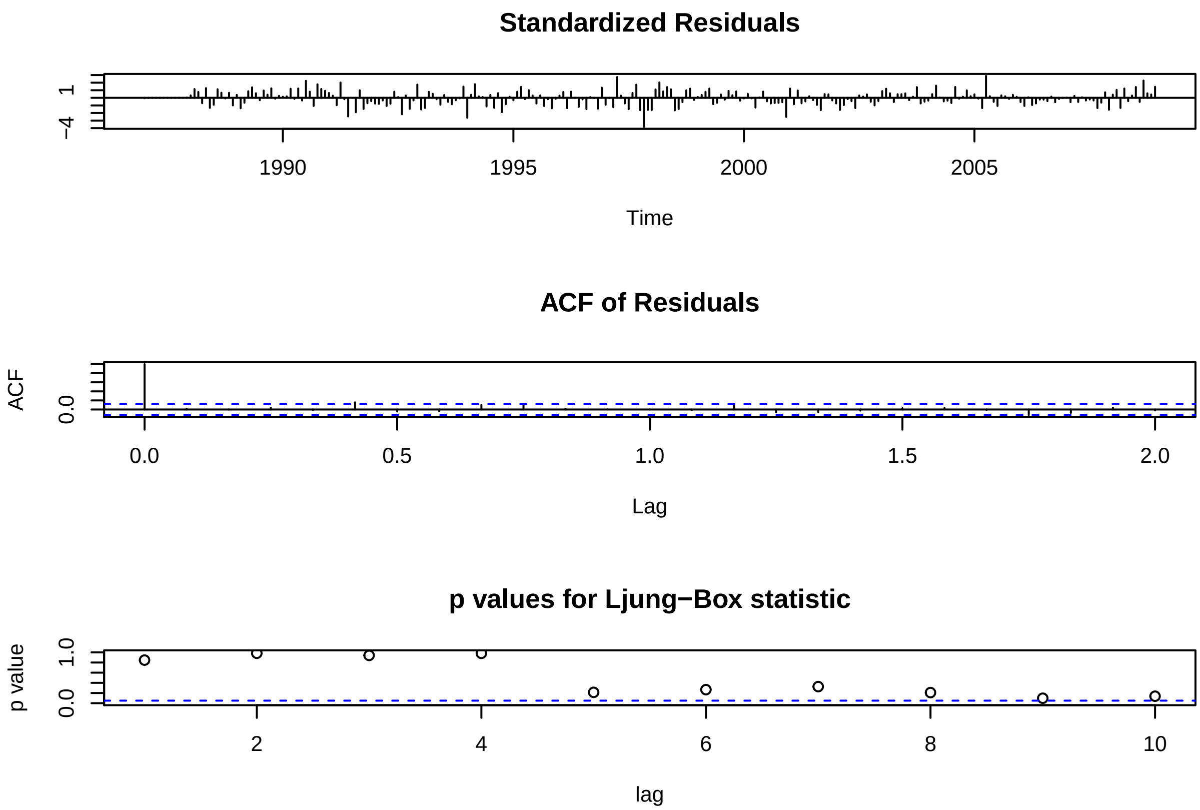
```r
####(3) test uncorrelatedness
Box.test(e, type='Ljung-Box', lag = 6)
```

```
##
##  Box-Ljung test
##
## data:  e
## X-squared = 7.6352, df = 6, p-value = 0.2661
```

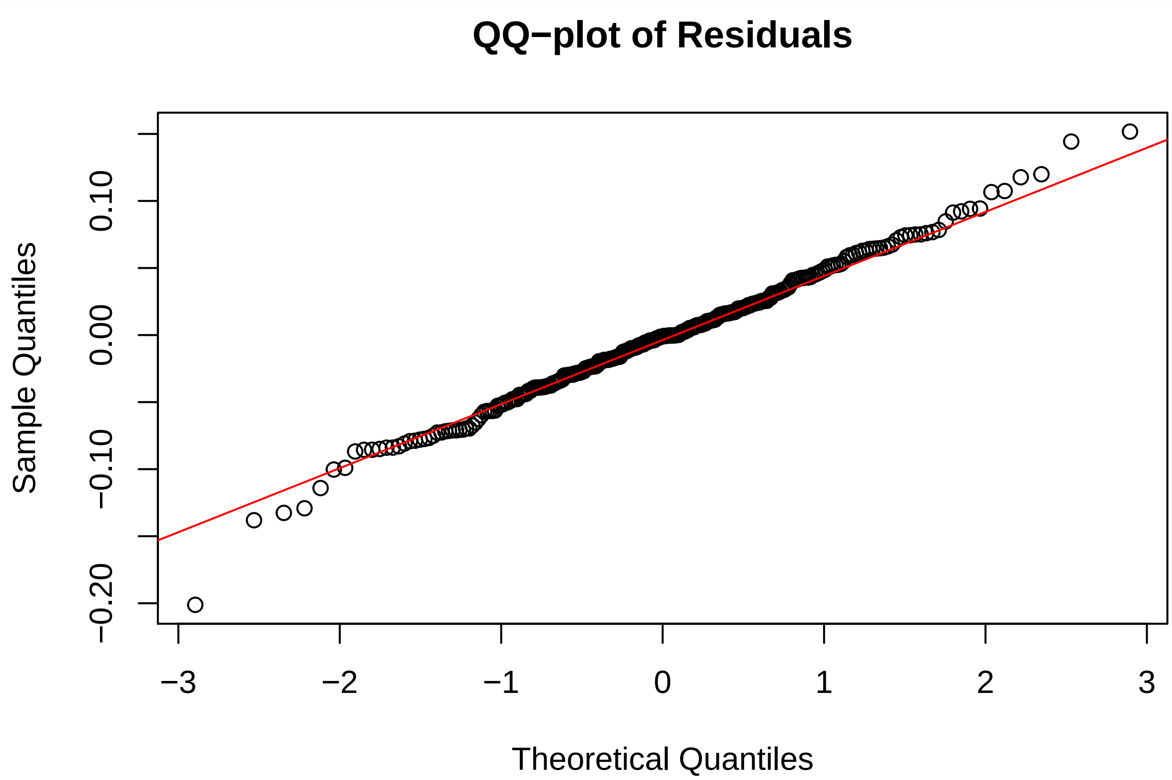
```r
Box.test(e, type='Ljung-Box', lag = 7)
```

```
##
##  Box-Ljung test
##
## data:  e
## X-squared = 8.0763, df = 7, p-value = 0.3259
```

```r
tsdiag(optimal.sarima)
```




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



```r
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.99391, p-value = 0.366
```

Modeling Approch 2: SARIMAX MODEL

```r
# 1. Add HPI
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg=   trai
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid$House_Price_Index)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.003246086
```

```r
m.sax1$aic
```

```
## [1] -699.0999
```

```r
m.sax1$loglik
```

```
## [1] 359.5499
```

```r
# 2. Add Unemployment rate
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg=   trai
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid$Unemployment_Rate)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.002923029
```

```r
m.sax1$aic
```

```
## [1] -682.1785
```

```r
m.sax1$loglik
```

```
## [1] 351.0892
```

```r
# 3. Add Population
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg=   trai
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid$Population)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.003276577
```

```r
m.sax1$aic
```

```
## [1] -679.727
```

```r
m.sax1$loglik
```

```
## [1] 349.8635
```

```r
# 4 HPI & Unemployment rate
train.ex <- subset(train, select = -c(Month, Bankruptcy_Rate, Population))
```

```r
valid.ex <- subset(valid, select = -c(Month, Bankruptcy_Rate, Population))
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg= train
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid.ex)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.003710359
```

```r
m.sax1$aic
```

```
## [1] -698.2308
```

```r
m.sax1$loglik
```

```
## [1] 360.1154
```

```r
# 5 HPI & Population
train.ex <- subset(train, select = -c(Month, Bankruptcy_Rate, Unemployment_Rate))
valid.ex <- subset(valid, select = -c(Month, Bankruptcy_Rate, Unemployment_Rate))
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg= train
```

```
## Warning in log(s2): NaNs produced

## Warning in log(s2): NaNs produced
```

```r
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid.ex)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.007657938
```

```r
m.sax1$aic
```

```
## [1] -704.5522
```

```r
m.sax1$loglik
```

```
## [1] 363.2761
```

```r
#6. Population & Unemployment rate
train.ex <- subset(train, select = -c(Month, Bankruptcy_Rate, House_Price_Index))
valid.ex <- subset(valid, select = -c(Month, Bankruptcy_Rate, House_Price_Index))
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg= train
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid.ex)#adding external
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.002963875
```

```r
m.sax1$aic
```

```
## [1] -680.9326
```

```r
m.sax1$loglik
```

```
## [1] 351.4663
```

```r
#7. Add all three
train.ex <- subset(train, select = -c(Month, Bankruptcy_Rate))
valid.ex <- subset(valid, select = -c(Month, Bankruptcy_Rate))
m.sax1  <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg= train
f.sax1 <- predict(m.sax1, n.ahead = 24, newxreg = valid.ex)#adding external
```

```r
rmse.sax1 <- sqrt(mean((exp(f.sax1$pred) - y.test)^2))
rmse.sax1
```

```
## [1] 0.007824753
```

```r
m.sax1$aic
```

```
## [1] -702.8928
```

```r
m.sax1$loglik
```

```
## [1] 363.4464
```

Choose optimal sarimaX

```r
optimal.sax <- arima(y.train, order = c(2,1,0), seasonal = list(order = c(3,1,3), period = 12), xreg=
```

Likelihood ratio test: SARIMA VS.SARIMAX

```r
myLRT(optimal.sarima, optimal.sax)
```

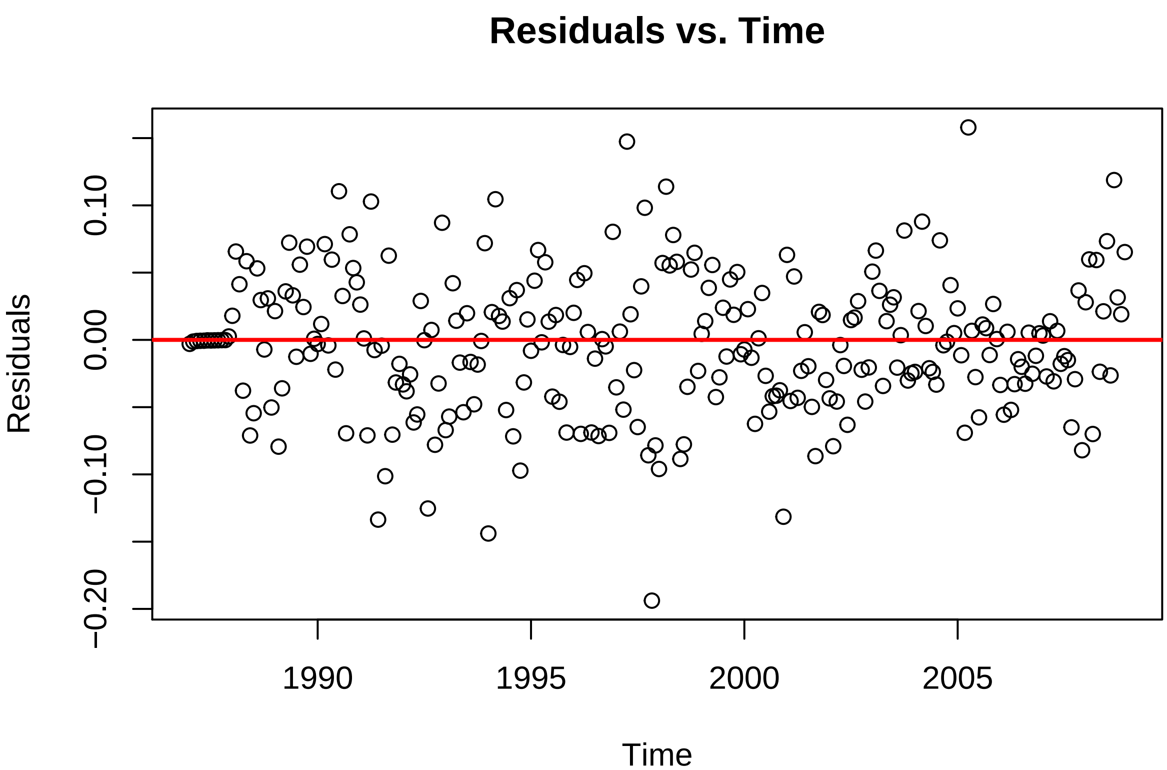
```
## [1] "Test Statistic:" "3.6058"          "P-value:"        "0.0576"
```

SARIMAX Model diagnostic

```r
e <- optimal.sax$residuals
####(1) test whether residuals have zero mean
t.test(e)
```

```
##
##  One Sample t-test
##
## data:  e
## t = -0.89669, df = 263, p-value = 0.3707
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.009017743  0.003374388
## sample estimates:
##    mean of x
## -0.002821678
```

```r
####(2) test heteroscedasticity
plot(e, main = "Residuals vs. Time", ylab = "Residuals", xlab = "Time", type='p') # plotting the residu
abline(h = 0, col = "red", lwd = 2) # plotting a horizontal line at 0
```




```
group <- cut(1:length(e), breaks=4, labels=(1:4))
leveneTest(e,group) #Levene
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   3   2.262 0.08168 .
##       260
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
bartlett.test(e,group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 7.1522, df = 3, p-value = 0.0672
```

```
####(3) test uncorrelatedness
Box.test(e, type='Ljung-Box', lag = 6)
```

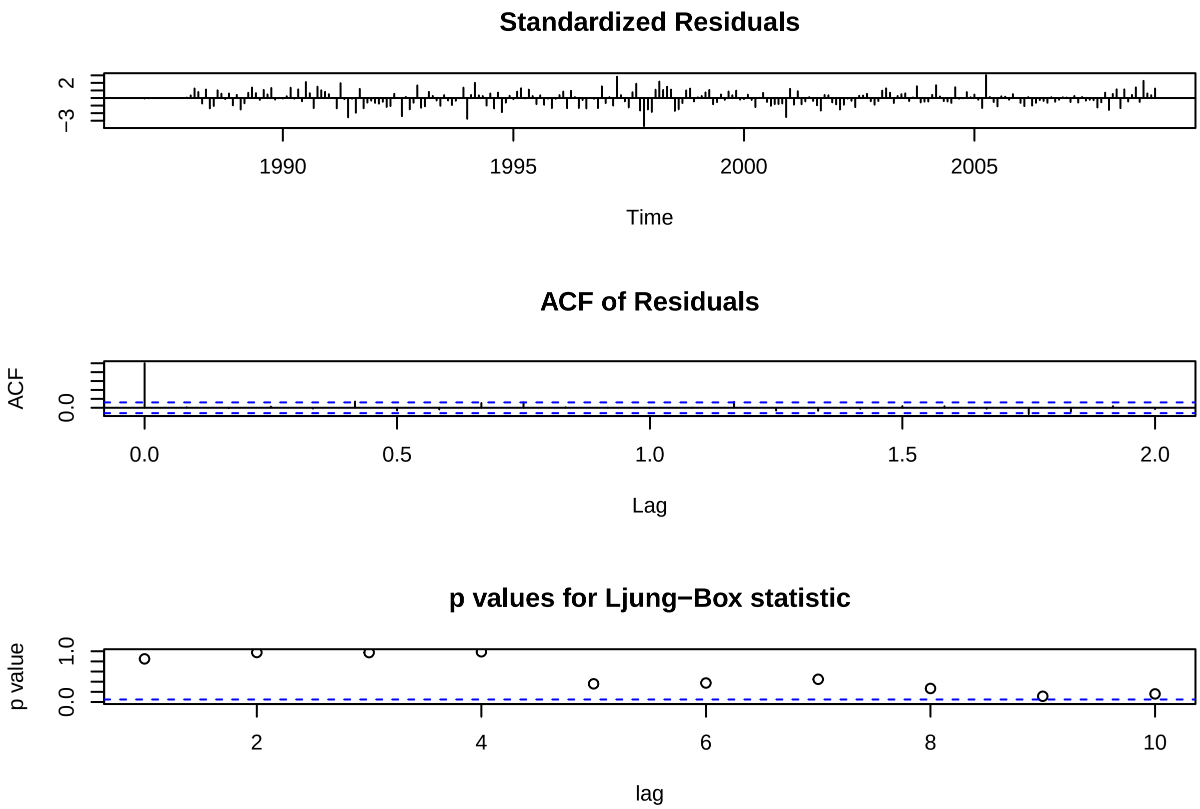
```
##
##  Box-Ljung test
##
## data:  e
## X-squared = 6.4526, df = 6, p-value = 0.3744
```

```
Box.test(e, type='Ljung-Box', lag = 7)
```

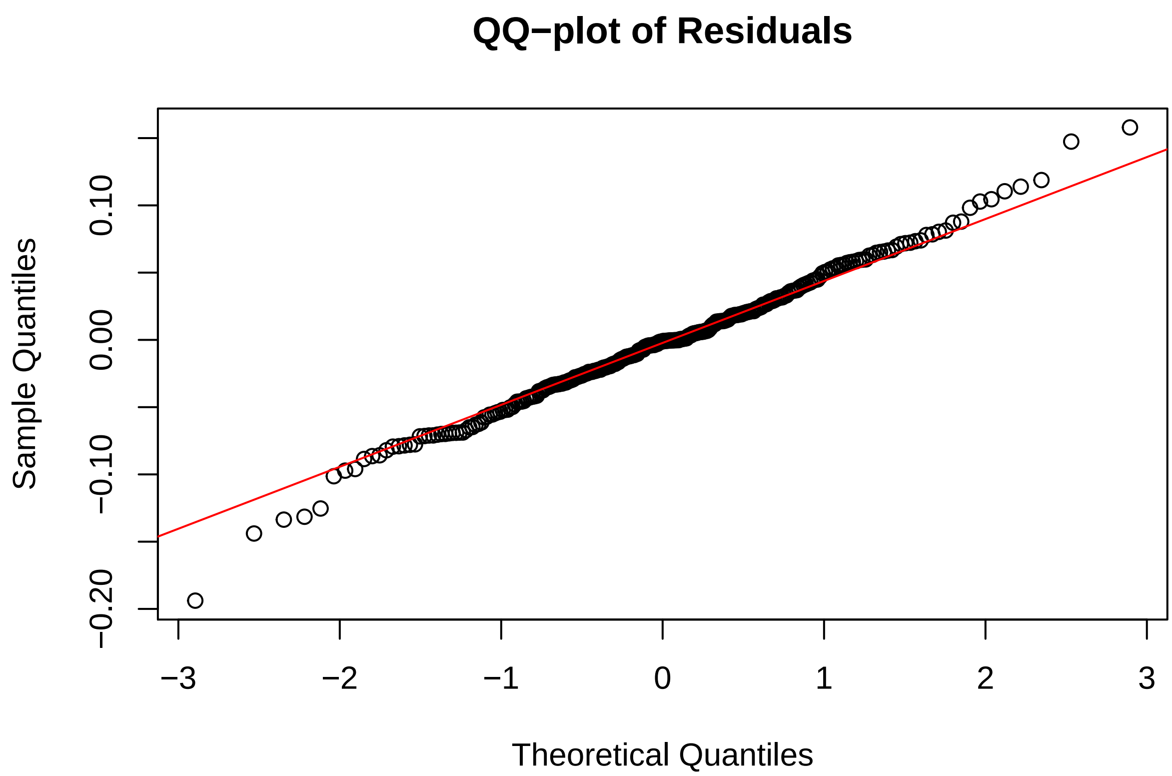
```
##
##  Box-Ljung test
##
```

```
## data:  e
## X-squared = 6.8227, df = 7, p-value = 0.4476
```

```r
tsdiag(optimal.sax)
```




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




```
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.99352, p-value = 0.3121
```

Modeling Approch 3: Vector autoregression

```
#Month indicator
train.sea <- train
train.sea['month'] = seq(1,12)
test.seas<- valid
test.seas['month'] = seq(1,12)

train.seas.month <- data.frame(feb = (train.sea$month==2)*1, mar = (train.sea$month==3)*1, apr = (train

test.seas.month <- data.frame(feb = (test.seas$month==2)*1, mar = (test.seas$month==3)*1, apr = (test.s
```

```
# (1) Include all 3 variables as endogenous vars
vardf <-  data.frame(y.train, subset(train, select = -c(Month, Bankruptcy_Rate)))
my.var<- VAR(y = vardf, ic = 'AIC', lag.max=3)
#summary(my.var)
test.pred <-predict(my.var, n.ahead=24, ci=0.95)
predict.y <- test.pred$fcst$y.train
rmse.var <- sqrt(mean( (exp(predict.y[,1]) - y.test)^2))
rmse.var
```

```
## [1] 0.005657714
```

```r
# (2) Include housing index only
vardf <-  data.frame(y.train, train$House_Price_Index)
my.var<- VAR(y = vardf, ic = 'AIC', lag.max=3)
#summary(my.var)
test.pred <-predict(my.var, n.ahead=24, ci=0.95)
predict.y <- test.pred$fcst$y.train
rmse.var <- sqrt(mean( (exp(predict.y[,1]) - y.test)^2))
rmse.var
```

```
## [1] 0.006244834
```

With season

```r
# (3) Include seasonal indicators
vardf <-  data.frame(y.train, subset(train, select = -c(Month, Bankruptcy_Rate)))
my.var<- VAR(y = vardf, ic = 'AIC', lag.max=2,  exogen=train.seas.month)
test.pred <-predict(my.var, n.ahead=24, ci=0.95, dumvar= test.seas.month)
predict.y <- test.pred$fcst$y.train
rmse.var <- sqrt(mean( (exp(predict.y[,1]) - y.test)^2))
rmse.var
```

```
## [1] 0.01695947
```

Choose optiomal VAR(p) model

```r
vardf <-  data.frame(y.train, subset(train, select = -c(Month, Bankruptcy_Rate)))
optimal.var<- VAR(y = vardf, ic = 'AIC', lag.max=3)
optimal.var$varresult
```

```
## $y.train
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##           y.train.l1  Unemployment_Rate.l1         Population.l1
##            3.085e-01             5.127e-02             4.988e-06
## House_Price_Index.l1            y.train.l2  Unemployment_Rate.l2
##           -8.098e-02             4.463e-02            -7.052e-03
##        Population.l2  House_Price_Index.l2            y.train.l3
##           -1.286e-05             2.874e-02             3.258e-01
## Unemployment_Rate.l3         Population.l3  House_Price_Index.l3
##           -3.862e-02             7.931e-06             5.091e-02
##                const
##           -3.030e+00
##
##
## $Unemployment_Rate
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
```

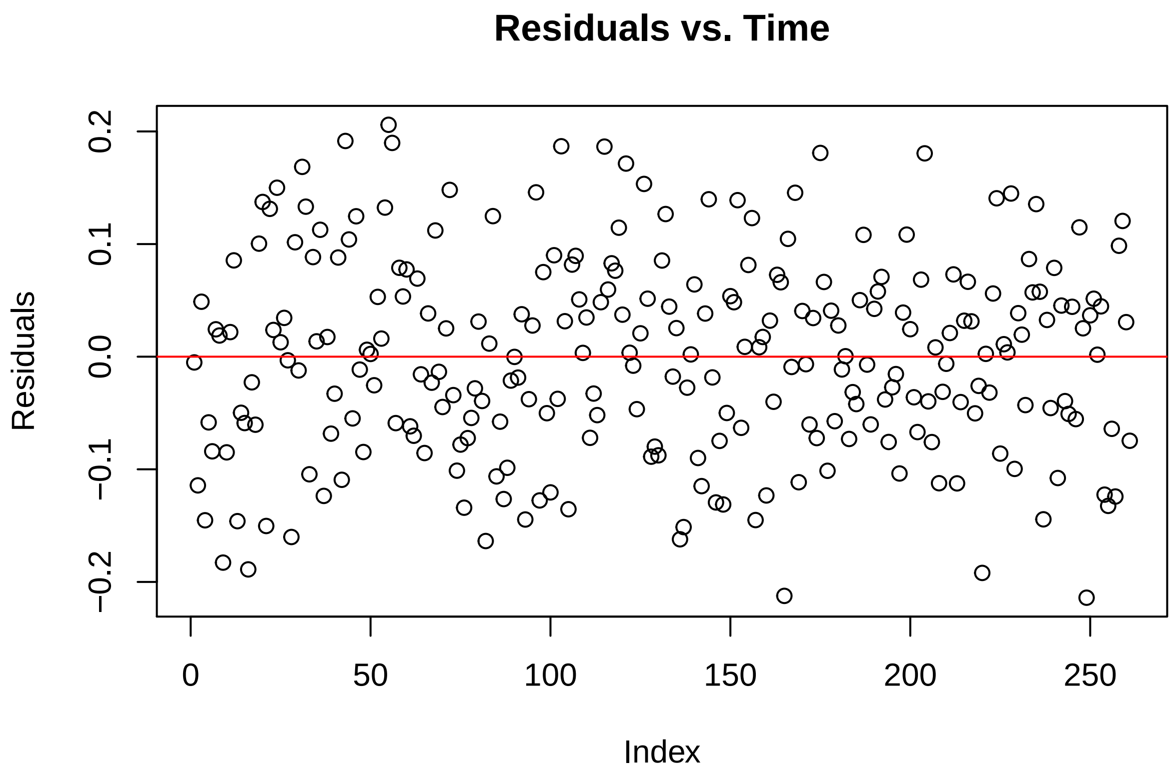
```
##
## Coefficients:
##           y.train.l1  Unemployment_Rate.l1         Population.l1
##            1.969e-01             8.703e-01            -2.069e-06
## House_Price_Index.l1            y.train.l2  Unemployment_Rate.l2
##           -5.542e-02             2.626e-01            -3.376e-02
##        Population.l2  House_Price_Index.l2            y.train.l3
##            4.501e-06             2.596e-02            -2.987e-02
## Unemployment_Rate.l3         Population.l3  House_Price_Index.l3
##            1.149e-01            -2.570e-06             3.851e-02
##                const
##            5.587e+00
##
##
## $Population
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##           y.train.l1  Unemployment_Rate.l1         Population.l1
##           -9.450e+03             6.024e+02             2.627e+00
## House_Price_Index.l1            y.train.l2  Unemployment_Rate.l2
##           -4.766e+02            -4.648e+03             8.833e+02
##        Population.l2  House_Price_Index.l2            y.train.l3
##           -2.608e+00            -8.178e+00             8.743e+03
## Unemployment_Rate.l3         Population.l3  House_Price_Index.l3
##           -1.542e+03             9.814e-01             5.742e+02
##                const
##           -2.479e+04
##
##
## $House_Price_Index
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##           y.train.l1  Unemployment_Rate.l1         Population.l1
##            2.618e-04            -8.407e-02            -1.504e-06
## House_Price_Index.l1            y.train.l2  Unemployment_Rate.l2
##            1.451e+00            -4.397e-02            -1.566e-02
##        Population.l2  House_Price_Index.l2            y.train.l3
##            9.159e-07            -2.052e-01            -1.661e-01
## Unemployment_Rate.l3         Population.l3  House_Price_Index.l3
##            9.156e-02             6.461e-07            -2.490e-01
##                const
##           -2.190e+00
```

```r
#Model diagnostic
une <- optimal.var$varresult$Unemployment_Rate
pop <- optimal.var$varresult$Population
br <- optimal.var$varresult$y.train
hi <- optimal.var$varresult$House_Price_Index
```

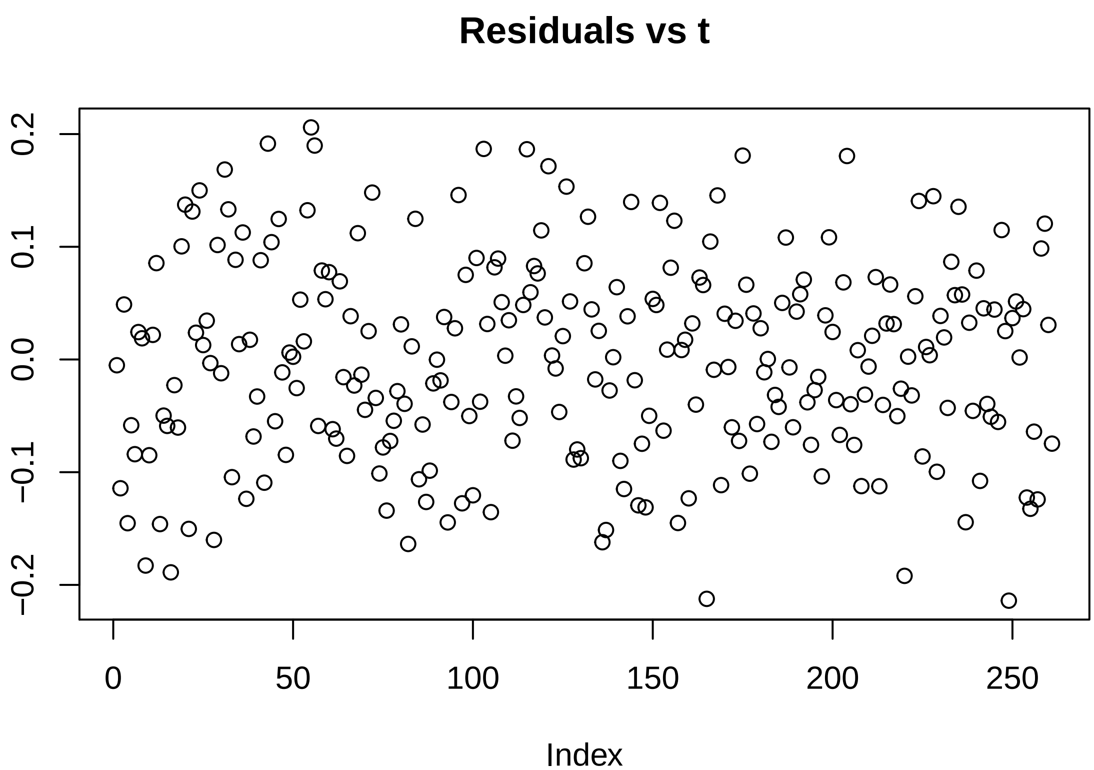
```r
e.une <- une$residuals
e.pop <- pop$residuals
e.br <- br$residuals
e.hi <- hi$residuals

# Residual Diagnostics:
# test whether residuals have zero mean pass
t.test(e.br)
```

```
##
##  One Sample t-test
##
## data:  e.br
## t = 1.7044e-16, df = 260, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.01078894  0.01078894
## sample estimates:
##    mean of x
## 9.338262e-19
```

```r
plot(e.br, main = "Residuals vs. Time", ylab = "Residuals")
abline(h = 0, col = "red")
```




```r
# test for heteroscedasticity
par(mfrow=c(1,1))
plot(e.br, main="Residuals vs t", ylab="")
abline(v=c(1992,1997,2003), lwd=3, col="red")
```



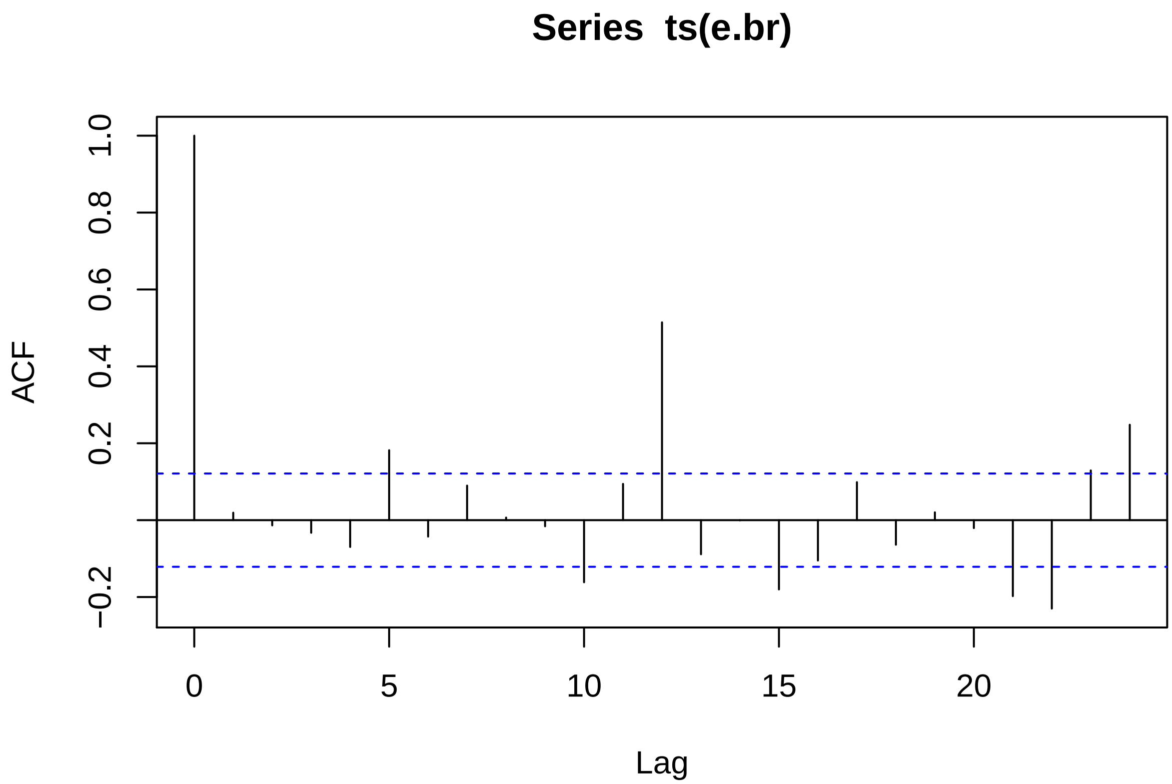

```r
#group <- c(rep(1,52),rep(2,52),rep(3,52),rep(4,53),rep(5,54))
group <- cut(1:length(e.br), breaks=4, labels=(1:4))
leveneTest(e.br,group) #Levene
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  0.7824 0.5047
##       257
```

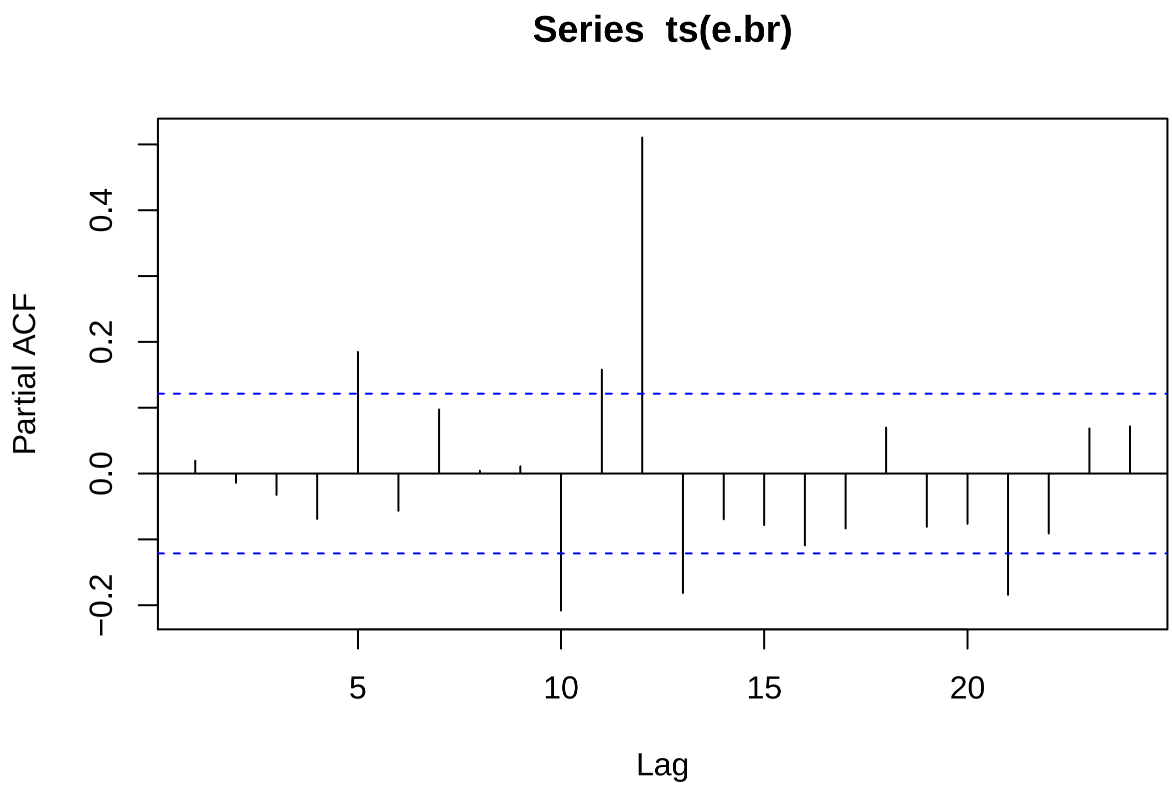
```r
bartlett.test(e.br, group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e.br and group
## Bartlett's K-squared = 1.9864, df = 3, p-value = 0.5752
```

```r
# test for uncorrelatedness
acf(ts(e.br))
```

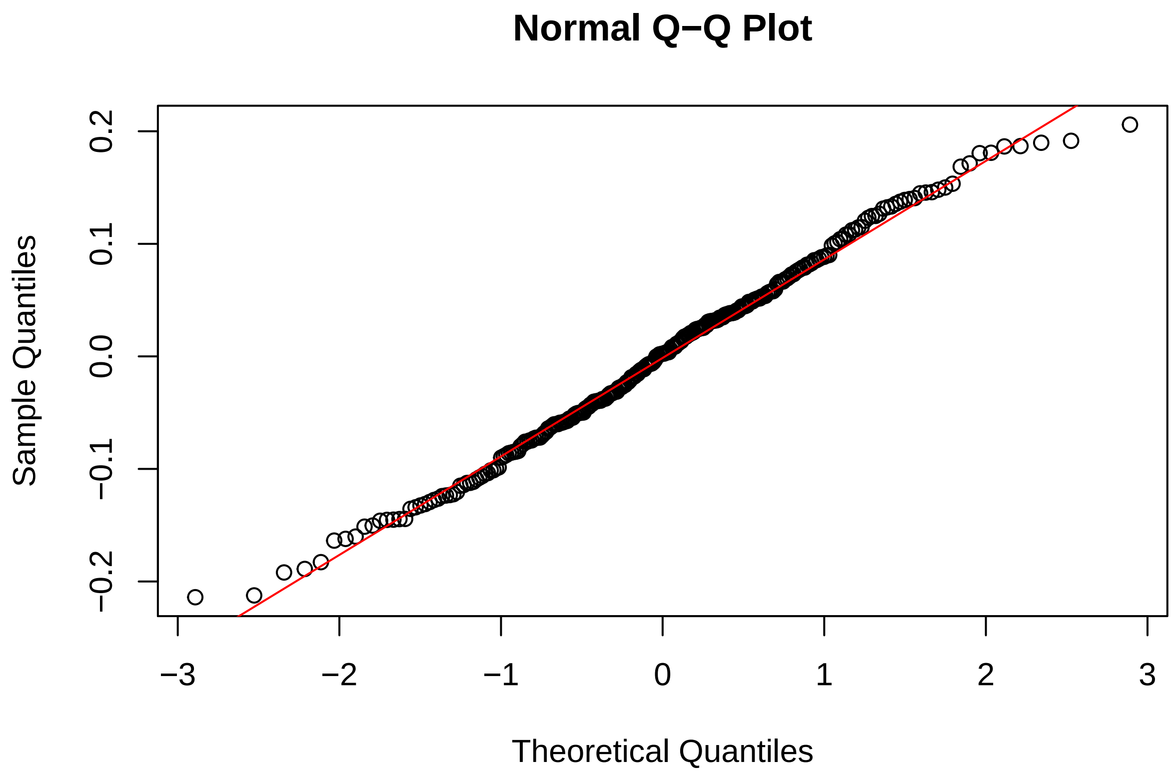



```
pacf(ts(e.br))
```




```
# test for normality pass
par(mfrow=c(1,1))
qqnorm(e.br)
```

```r
qqline(e.br, col = "red")
```




```r
shapiro.test(e.br)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e.br
## W = 0.99311, p-value = 0.2711
```

test for uncorrelatedness doesn't pass.

Modeling Approch 4: Holt-Winters

```r
##Additive Holt-Winters approach
m_holt_winters <- HoltWinters(x = y.train, seasonal = 'add')
pred_holt_winters <- forecast(m_holt_winters, h = 24, prediction.interval = T, level = 0.95)

#Taking exponential of predictions
pred_holt_winters$x     <- exp(pred_holt_winters$x)
pred_holt_winters$mean  <- exp(pred_holt_winters$mean)
pred_holt_winters$upper <- exp(pred_holt_winters$upper)
pred_holt_winters$lower <- exp(pred_holt_winters$lower)

rmse <- sqrt(mean((y.test - pred_holt_winters$mean)^2))
rmse
```

```
## [1] 0.01655722
```

```r
##Multiplicative Holt-Winters approach
m_holt_winters <- HoltWinters(x = y.train, seasonal = 'mult')
pred_holt_winters <- forecast(m_holt_winters, h = 24, prediction.interval = T, level = 0.95)

#Taking exponential of predictions
pred_holt_winters$x     <- exp(pred_holt_winters$x)
pred_holt_winters$mean  <- exp(pred_holt_winters$mean)
pred_holt_winters$upper <- exp(pred_holt_winters$upper)
pred_holt_winters$lower <- exp(pred_holt_winters$lower)

rmse <- sqrt(mean((y.test - pred_holt_winters$mean)^2))
rmse
```

```
## [1] 0.01540406
```

Choose the final optimal model to forecast bankruptcy in test data

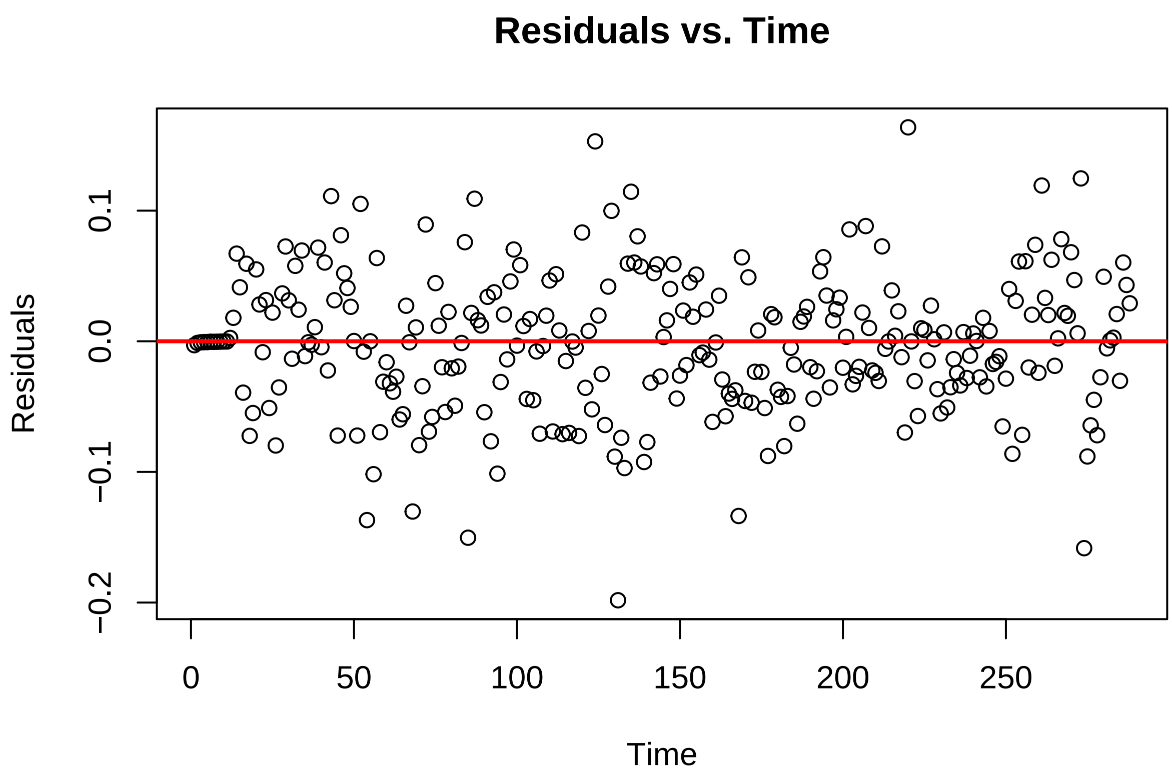
```r
# Use all data to train selected SARIMAX model
optimal.final <- arima(log(data$Bankruptcy_Rate), order = c(2,1,0), seasonal = list(order = c(3,1,3), p
```

Final Model diagnostic

```r
e <- optimal.final$residuals
####(1) test whether residuals have zero mean
t.test(e)
```

```
##
##  One Sample t-test
##
## data:  e
## t = -0.81143, df = 287, p-value = 0.4178
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.008652308  0.003600854
## sample estimates:
##    mean of x
## -0.002525727
```

```r
####(2) test heteroscedasticity
plot(e, main = "Residuals vs. Time", ylab = "Residuals", xlab = "Time", type='p') # plotting the residu
abline(h = 0, col = "red", lwd = 2) # plotting a horizontal line at 0
```




```r
group <- cut(1:length(e), breaks=3, labels=(1:3))
leveneTest(e,group) #Levene
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   2  1.1011 0.3339
##       285
```

```r
bartlett.test(e,group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 1.5755, df = 2, p-value = 0.4549
```

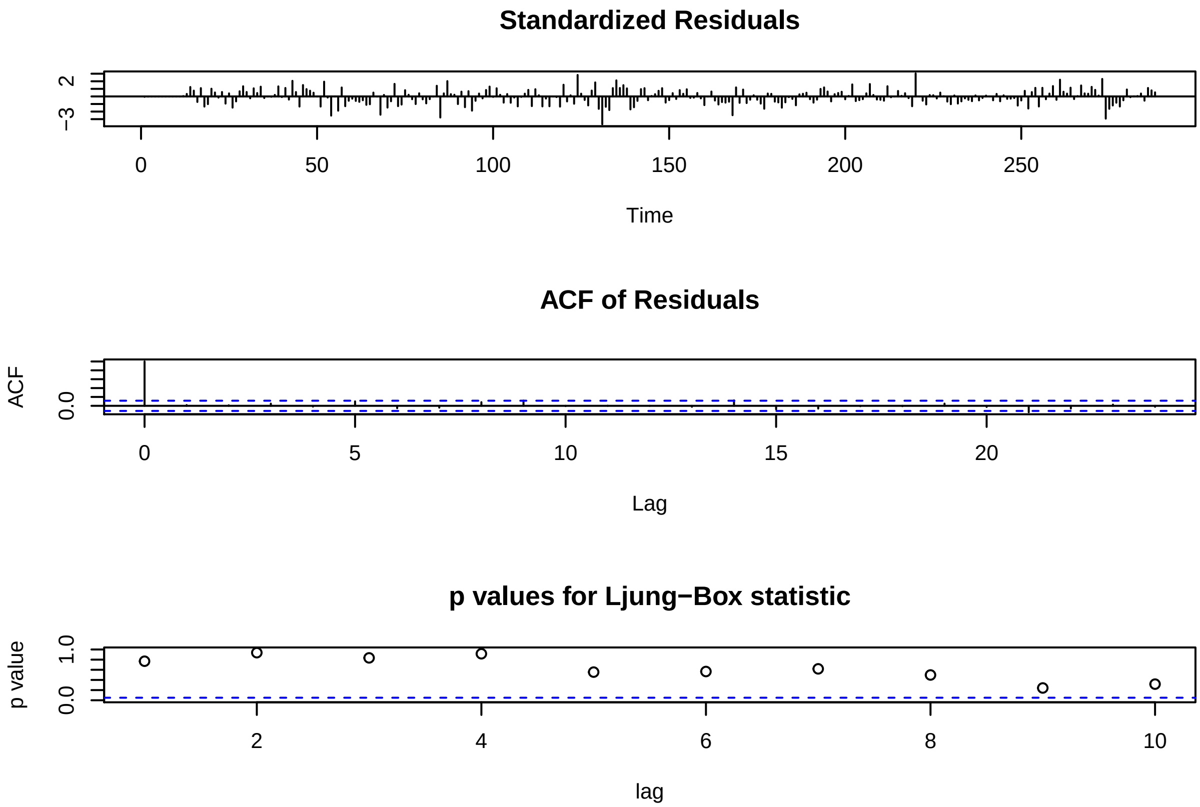
```r
####(3) test uncorrelatedness
Box.test(e, type='Ljung-Box', lag = 6)
```

```
##
##  Box-Ljung test
##
## data:  e
## X-squared = 4.8291, df = 6, p-value = 0.5659
```

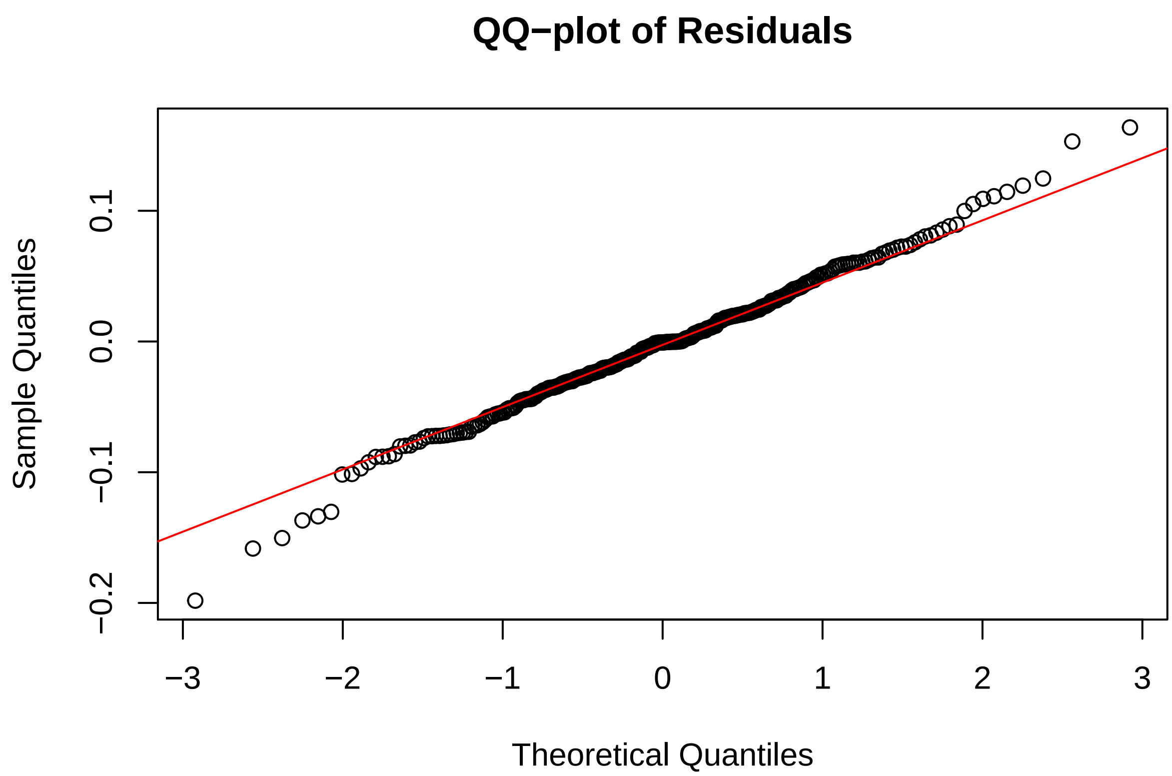
```r
Box.test(e, type='Ljung-Box', lag = 7)
```

```
##
##  Box-Ljung test
##
## data:  e
## X-squared = 5.3425, df = 7, p-value = 0.6182
```

```r
tsdiag(optimal.final)
```




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



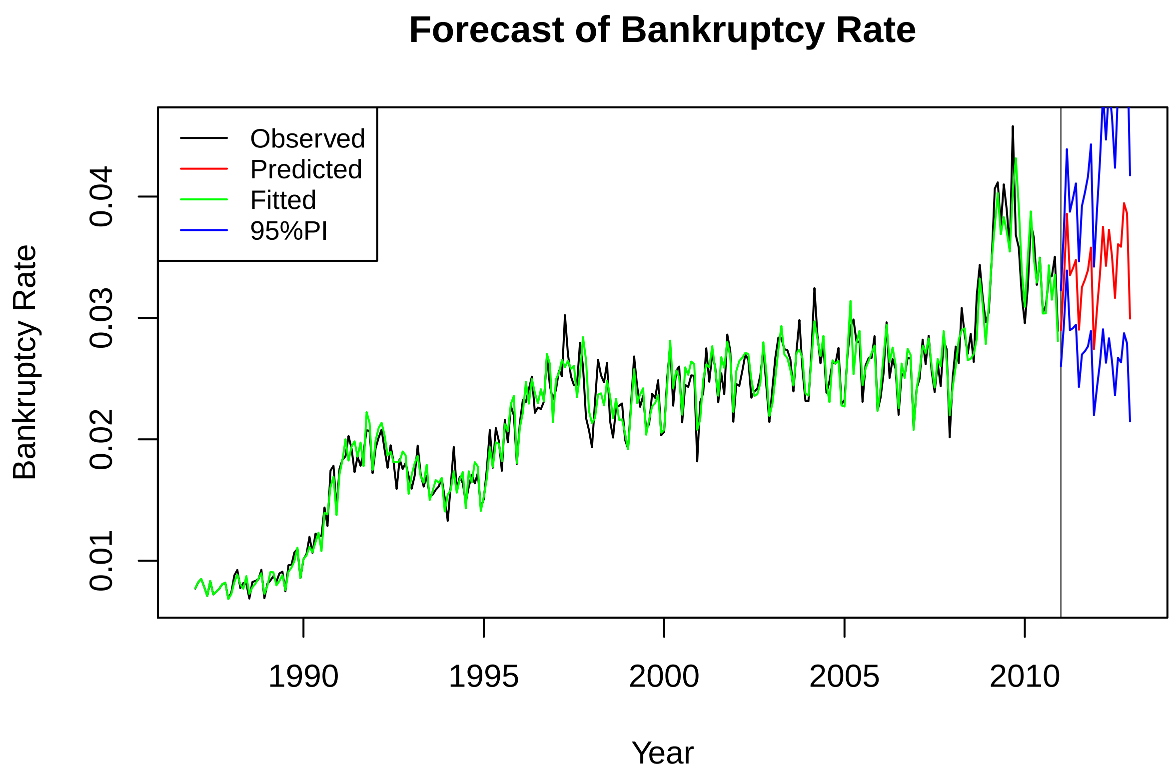

```r
shapiro.test(e)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.99263, p-value = 0.1657
```

Forecasting test data

```r
fitted.y <- exp(fitted(optimal.final)) #fitted value
fitted.y.ts <- ts(fitted.y, start = c(1987,1), frequency = 12)
fit.l.ts <- window(fitted.y.ts, end=c(2010,12))

f.sax1 <- forecast(object = optimal.final, h=24, level = 0.95, xreg = test$Unemployment_Rate) #forecast
pred.ts  <-  ts(exp(f.sax1$mean), start = c(2011,1), frequency = 12)
low95.ts <- ts(exp(f.sax1$lower), start = c(2011,1), frequency = 12)
upper95.ts <-ts(exp(f.sax1$upper),  start = c(2011,1), frequency = 12)

# Generate forecast results in graph
par(mfrow  =  c(1,  1))
data <- read.csv("/Users/xiaohui/Documents/0_2017_USF/MSAN_604_TS/Final project/train.csv", header = TRU
plot(ts(data$Bankruptcy_Rate, start = c(1987,1), frequency = 12), type='l', main = "Forecast of Bankrup
abline(v = 2011,  lwd  =  0.5,  col  =  "black")


lines(fit.l.ts ,col='green', type='l')
lines(pred.ts  ,col='red', type='l')
```

```r
lines(low95.ts ,col='blue', type='l')
lines(upper95.ts ,col='blue', type='l')

legend("topleft",lty=c(1,1),cex=0.8,
       col=c("black","red",'green','blue'),
       legend=c('Observed','Predicted','Fitted','95%PI'))
```




```r
# Generate forecasting results in table
month <- gsub('.{3}$', '', seq(as.Date("2011/1/1"), by="month", length=24))
prediction.final <- data.frame(month, c(pred.ts),c(low95.ts),c(upper95.ts))
colnames(prediction.final) <- c("Month","Prediction","Lower Bound(95%)", "Upper Bound(95%)")
knitr::kable(prediction.final, digits = 4, align = "r")
```

Month	Prediction	Lower Bound(95%)	Upper Bound(95%)
2011-01	0.0290	0.0260	0.0323
2011-02	0.0332	0.0295	0.0373
2011-03	0.0386	0.0339	0.0439
2011-04	0.0335	0.0290	0.0388
2011-05	0.0341	0.0292	0.0398
2011-06	0.0348	0.0294	0.0411
2011-07	0.0290	0.0243	0.0347
2011-08	0.0325	0.0270	0.0392
2011-09	0.0332	0.0273	0.0403
2011-10	0.0339	0.0277	0.0416
2011-11	0.0358	0.0289	0.0443
2011-12	0.0274	0.0220	0.0342
2012-01	0.0307	0.0243	0.0388
2012-02	0.0336	0.0263	0.0430
2012-03	0.0375	0.0291	0.0484

Month	Prediction	Lower Bound(95%)	Upper Bound(95%)
2012-04	0.0343	0.0263	0.0447
2012-05	0.0373	0.0283	0.0490
2012-06	0.0351	0.0264	0.0466
2012-07	0.0316	0.0236	0.0424
2012-08	0.0361	0.0267	0.0487
2012-09	0.0359	0.0263	0.0488
2012-10	0.0395	0.0287	0.0542
2012-11	0.0386	0.0279	0.0534
2012-12	0.0299	0.0215	0.0417