Time Series Project

Piyush Bhargava, Chuiyi Liu, Cong Qing, Meg Ellis

In the file “train.csv” you will find monthly data from January 1987 to December 2010 on the following variables: • Unemployment Rate • Population • Bankruptcy Rate • Housing Price Index

Choose Models

Loading the train dataset and looking at the time series plots.

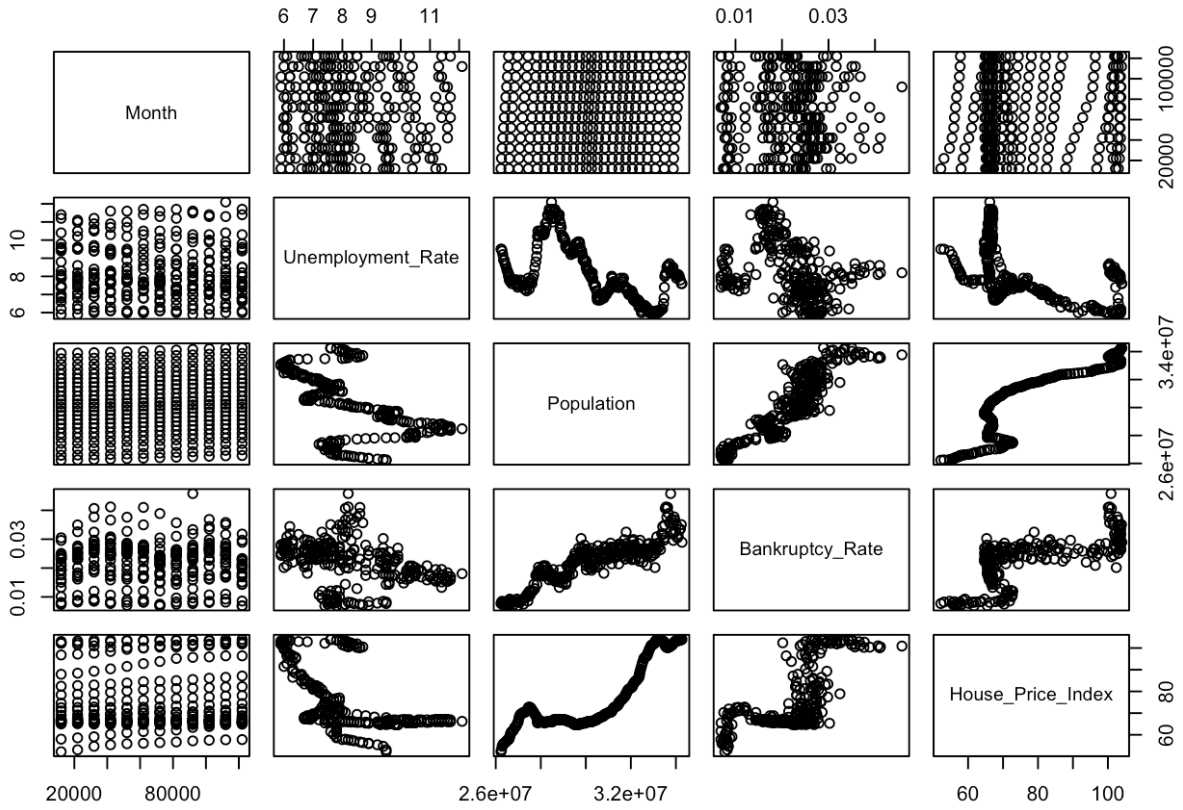
```
library(tseries, quietly = TRUE)
library(lawstat, quietly = TRUE)
```

```
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
##
##
## Attaching package: 'lawstat'
##
## The following object is masked from 'package:tseries':
##
##     runs.test
```

```
library(forecast)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Loading required package: timeDate
## This is forecast 6.2
```

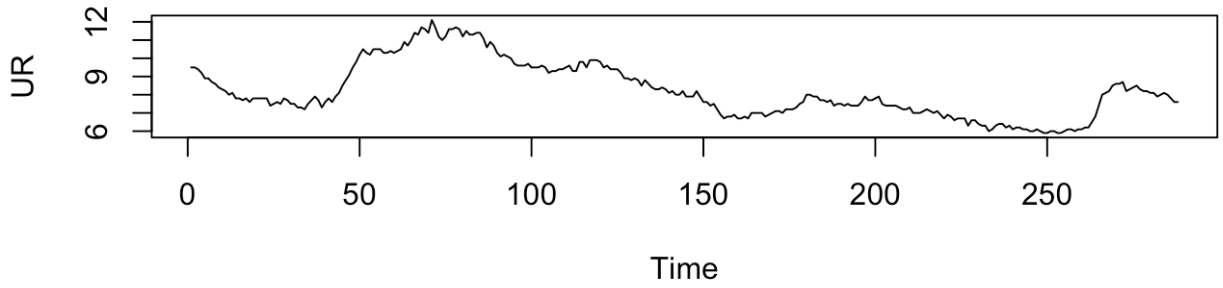
```
train <- read.csv('/Users/tracy/msan-ts/project/train.csv', header=TRUE)
par(mfrow=c(1,1))
plot(train)
```

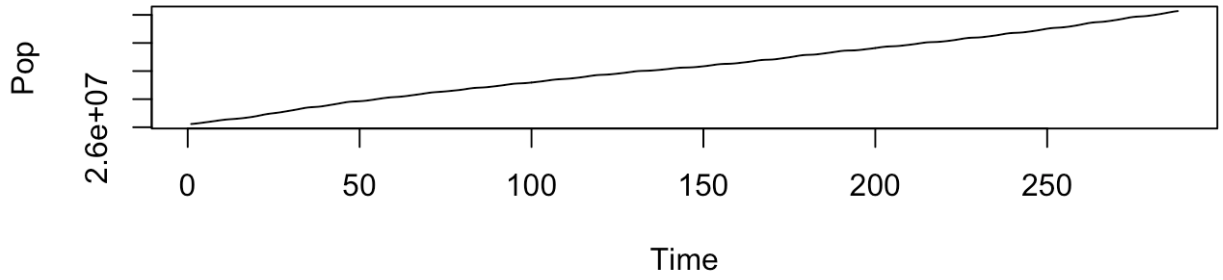



```
head(train)
```

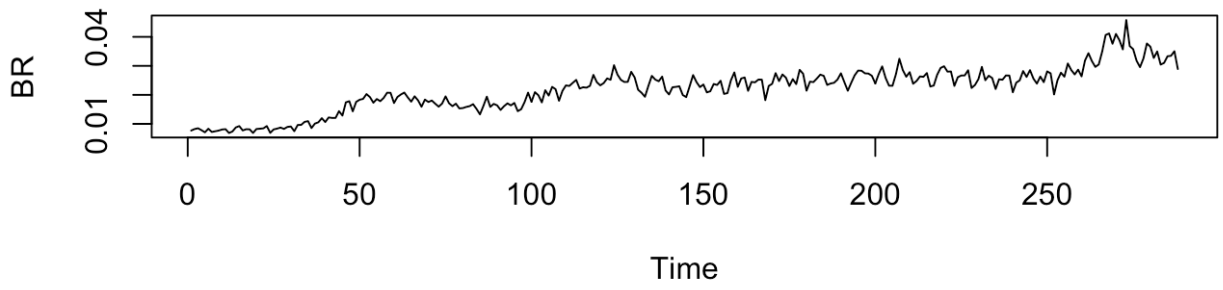
```
##   Month Unemployment_Rate Population Bankruptcy_Rate House_Price_Index
## 1 11987               9.5   26232423       0.0077004               52.2
## 2 21987               9.5   26254410       0.0082196               53.1
## 3 31987               9.4   26281420       0.0084851               54.7
## 4 41987               9.2   26313260       0.0078326               55.4
## 5 51987               8.9   26346526       0.0070901               55.9
## 6 61987               8.9   26379319       0.0083285               56.1
```

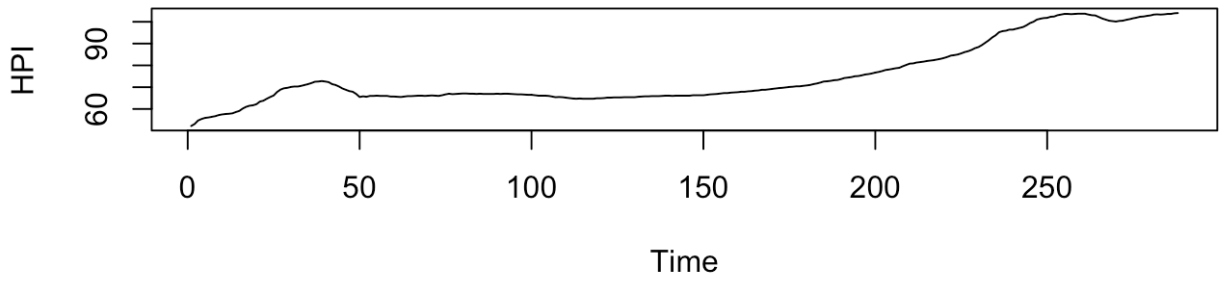
```
UR <- ts(train$Unemployment_Rate)
Pop <- ts(train$Population)
BR <- ts(train$Bankruptcy_Rate)
HPI <- ts(train$House_Price_Index)

#plot the variables:
par(mfrow=c(2,1))
plot(UR)
plot(Pop)
```





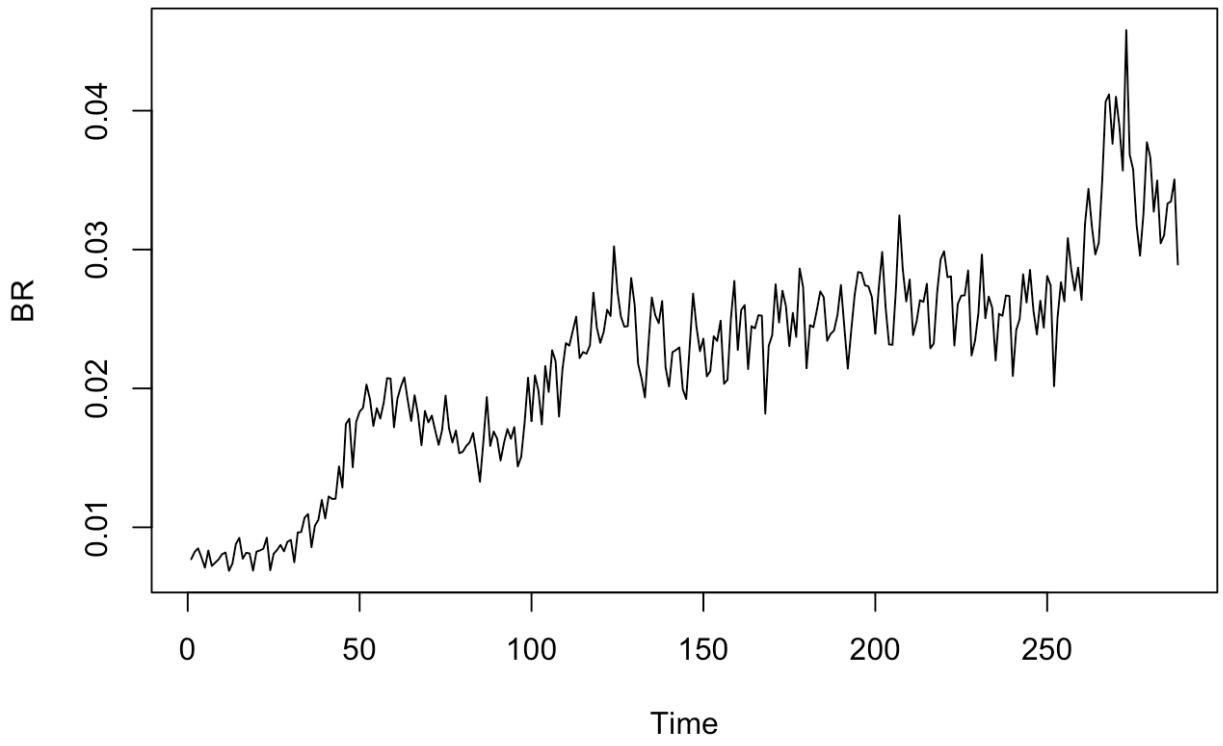

```
par(mfrow=c(2,1))
plot(BR)
plot(HPI)
```



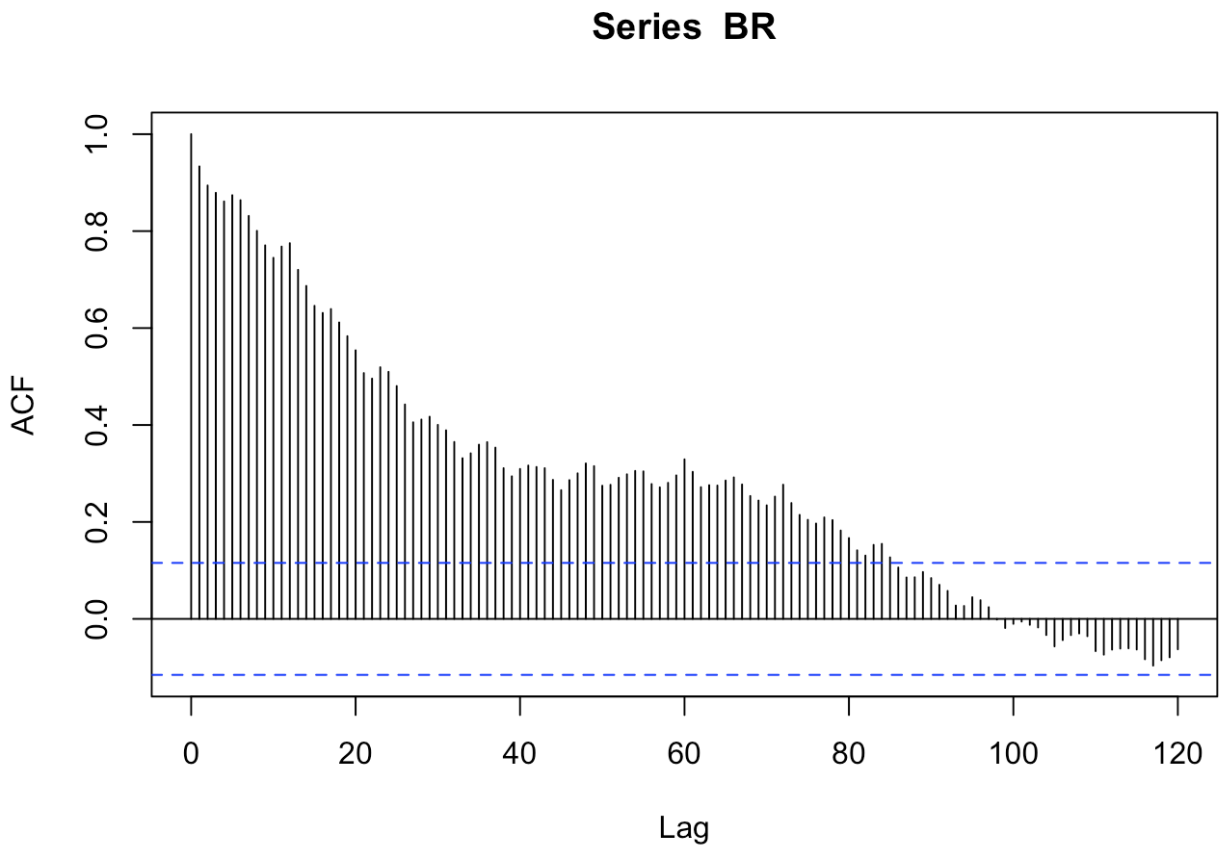


Looking at the plots and performing Augmented Dickey-Fuller Test for Bankruptcy rates and other time series above

```
par(mfrow=c(1,1))
plot(BR)
```




```
acf(BR, lag.max = 120)
```



```
ndiffs(BR)
```

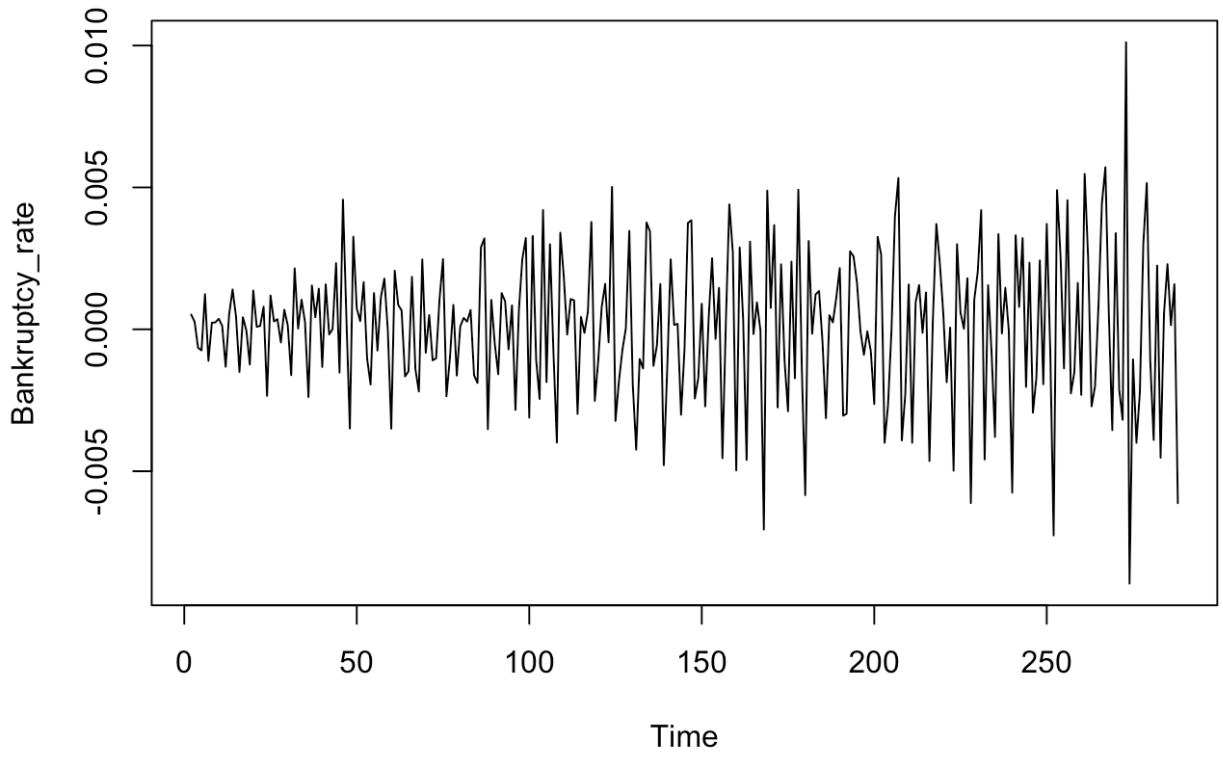
```
## [1] 1
```

```
adf.test(BR)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  BR
## Dickey-Fuller = -2.4516, Lag order = 6, p-value = 0.3859
## alternative hypothesis: stationary
```

The p-value from the ADF test is greater than 0.05 but the ACF plot indicates that the time series is not stationary. Hence, Differencing once and performing the Augmented Dickey-Fuller Test again.

```
BR.1 <- diff(BR)
plot(BR.1, ylab = "Bankruptcy_rate")
```

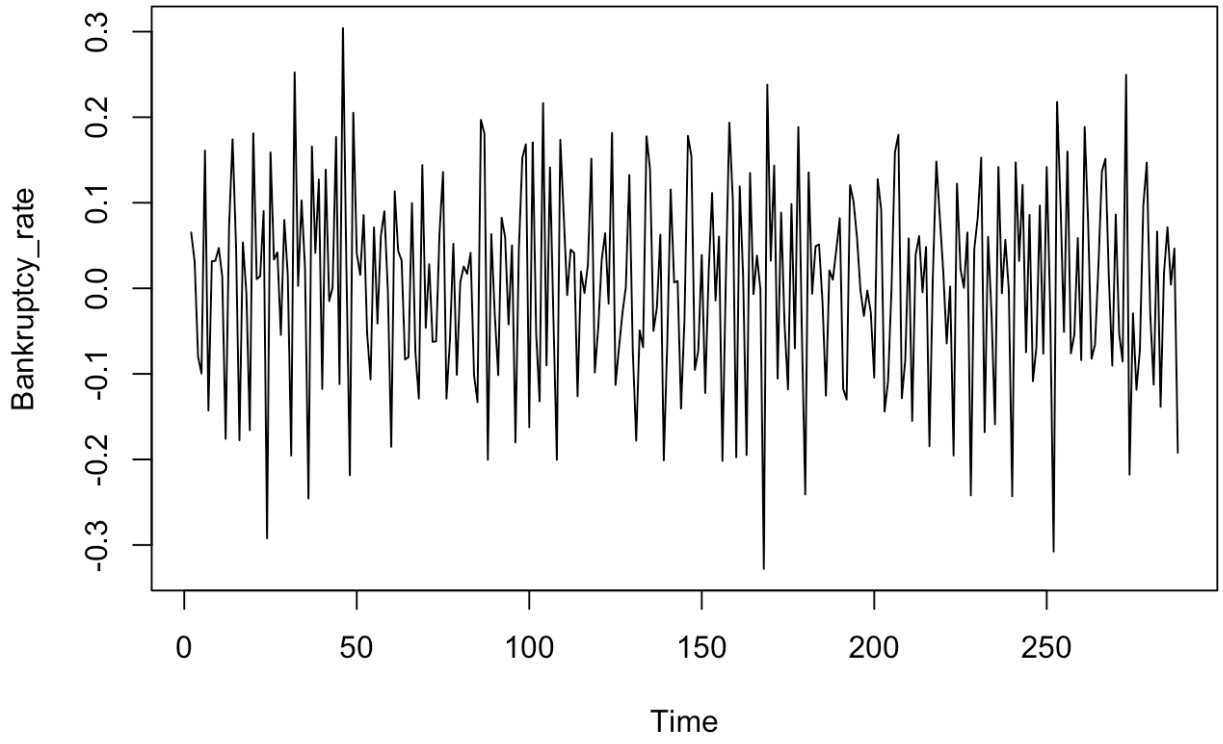



```
l.BR.1 <- diff(log(BR))
adf.test(l.BR.1)
```

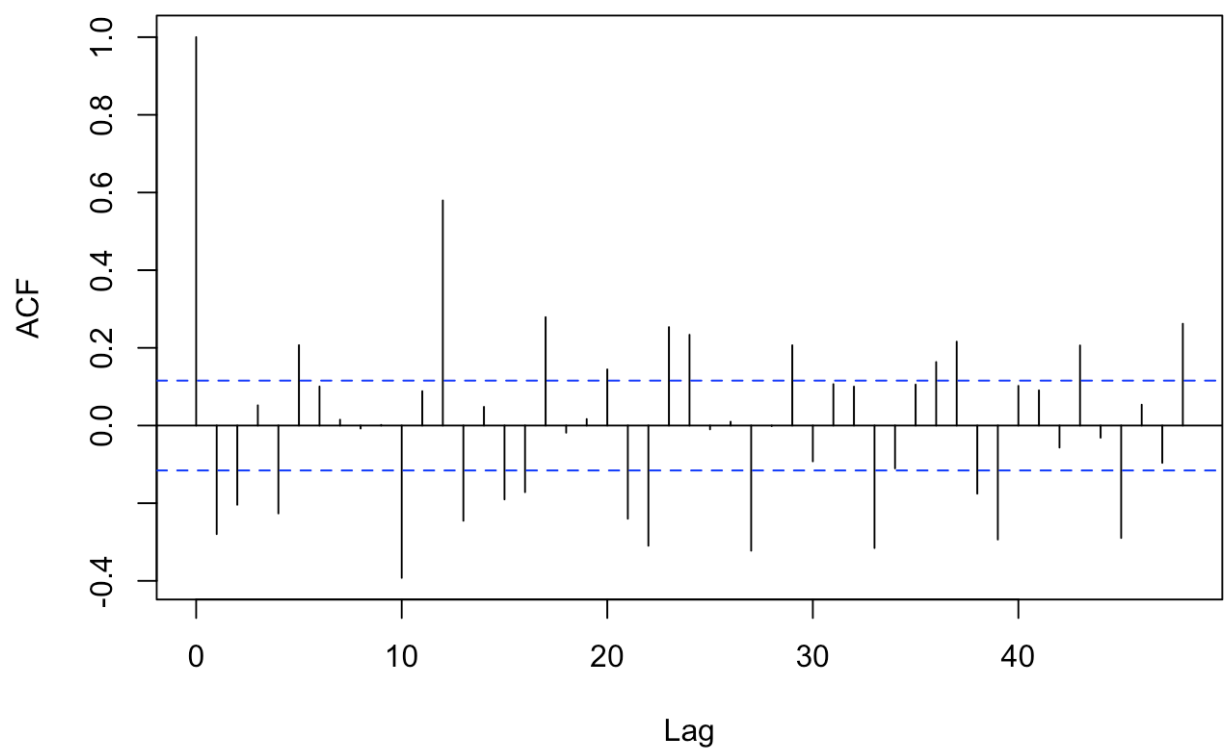
```
## Warning in adf.test(l.BR.1): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  l.BR.1
## Dickey-Fuller = -7.3221, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
par(mfrow=c(1,1))
plot(l.BR.1, ylab = "Bankruptcy_rate")
```




```
acf(l.BR.1, lag.max = 48)
```



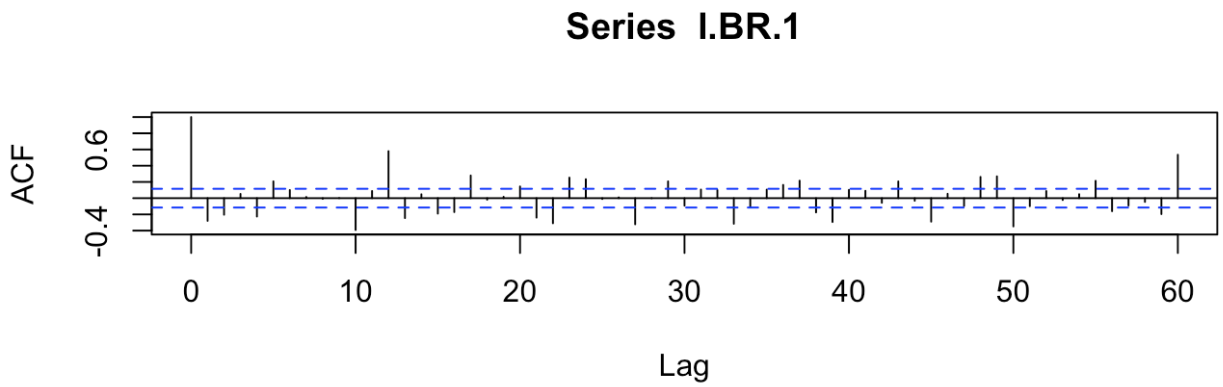

```
nsdiffs(log(BR),12)
```

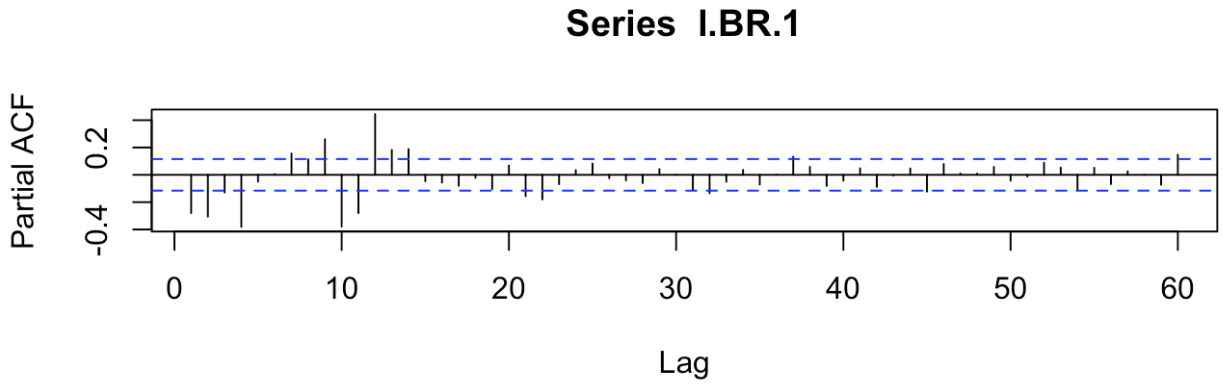
```
## [1] 0
```

Since the variation doesn't look constant, performing log transformation. This iteration passes the test easily, so we do not need to difference any further. Looking at the ACF plot, there seems to be monthly seasonality (period = 12) but the seasonal differencing may not be required. nsdiffs() also indicates that seasonal differencing is not required. Hence, choosing d=1, D=1, s=12.

Checking ACF and PACF to choose p, q, P, Q.

```
#order selection:
par(mfrow=c(2,1))
acf(l.BR.1, lag.max = 60)
pacf(l.BR.1, lag.max = 60)
```





Looking at ACF plot for q and Q, and PACF plot for p and P, maybe p = 2, q = 2, and P = 1, Q = 2 can be considered. Fitting the model using Least Squares (LS) as well as Maximum Likelihood (ML) estimation.

```
m.ml.1 <- arima(log(BR), order = c(2,1,2), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
m.ls.1 <- arima(log(BR), order = c(2,1,2), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "CSS")
m.ml.1
```

```
##
## Call:
## arima(x = log(BR), order = c(2, 1, 2), seasonal = list(order = c(1, 0, 2), p
eriod = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2     ma1     ma2    sar1     sma1     sma2
##       -1.0381  -0.6189  0.4434  0.0829  0.9996  -0.6481  -0.3094
## s.e.   0.0936   0.0897  0.1108  0.1572  0.0019   0.0853   0.0760
##
## sigma^2 estimated as 0.004045:  log likelihood = 365.93,  aic = -715.85
```

```
m.ls.1
```

```
##
## Call:
## arima(x = log(BR), order = c(2, 1, 2), seasonal = list(order = c(1, 0, 2), p
eriod = 12),
##     method = "CSS")
##
## Coefficients:
##           ar1      ar2     ma1      ma2    sar1     sma1     sma2
##       -1.1318  -0.5479  0.5334  -0.0934  0.9910  -0.5195  -0.3071
## s.e.   0.1171   0.0918  0.1378   0.1484  0.0111   0.0644   0.0670
##
## sigma^2 estimated as 0.004545:  part log likelihood = 366.75
```

Since the estimates from ML and LS are similar, ‘Normality’ assumption seems reasonable.

MA(2) doesn’t look significant since SE is high

Now, fitting more models of lower orders using ML and using the output information such as Log likelihood, AIC and sigma^2 to select the model with the best fit.

```
m.ml.2 <- arima(log(BR), order = c(2,1,1), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
m.ml.3 <- arima(log(BR), order = c(2,1,0), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
m.ml.4 <- arima(log(BR), order = c(1,1,1), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
m.ml.5 <- arima(log(BR), order = c(1,1,0), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
m.ml.6 <- arima(log(BR), order = c(0,1,1), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
```

```
## Warning in arima(log(BR), order = c(0, 1, 1), seasonal = list(order =
## c(1, : possible convergence problem: optim gave code = 1
```

```
m.ml.7 <- arima(log(BR), order = c(0,1,0), seasonal = list(order = c(1,0,2), pe
riod = 12), method = "ML")
```

```
## Warning in arima(log(BR), order = c(0, 1, 0), seasonal = list(order =
## c(1, : possible convergence problem: optim gave code = 1
```

```
m.ml.8 <- arima(log(BR), order = c(2,1,1), seasonal = list(order = c(1,0,1), pe
riod = 12), method = "ML")
m.ml.9 <- arima(log(BR), order = c(2,1,0), seasonal = list(order = c(1,0,1), pe
riod = 12), method = "ML")
m.ml.10 <- arima(log(BR), order = c(1,1,1), seasonal = list(order = c(1,0,1), p
eriod = 12), method = "ML")
m.ml.11 <- arima(log(BR), order = c(1,1,0), seasonal = list(order = c(1,0,1), p
eriod = 12), method = "ML")
m.ml.12 <- arima(log(BR), order = c(0,1,1), seasonal = list(order = c(1,0,1), p
eriod = 12), method = "ML")
m.ml.13 <- arima(log(BR), order = c(0,1,0), seasonal = list(order = c(1,0,1), p
eriod = 12), method = "ML")
sigma2<-c(m.ml.1$sigma2,m.ml.2$sigma2,m.ml.3$sigma2,m.ml.4$sigma2,m.ml.5$sigma
2,m.ml.6$sigma2,m.ml.7$sigma2,m.ml.8$sigma2,m.ml.9$sigma2,m.ml.10$sigma2,m.ml.1
1$sigma2,m.ml.12$sigma2,m.ml.13$sigma2)
loglik<-c(m.ml.1$loglik,m.ml.2$loglik,m.ml.3$loglik,m.ml.4$loglik,m.ml.5$logli
k,m.ml.6$loglik,m.ml.7$loglik,m.ml.8$loglik,m.ml.9$loglik,m.ml.10$loglik,m.ml.1
1$loglik,m.ml.12$loglik,m.ml.13$loglik)
AIC<-c(m.ml.1$aic,m.ml.2$aic,m.ml.3$aic,m.ml.4$aic,m.ml.5$aic,m.ml.6$aic,m.m
l.7$aic,m.ml.8$aic,m.ml.9$aic,m.ml.10$aic,m.ml.11$aic,m.ml.12$aic,m.ml.13$aic)
d <- data.frame(sigma2,loglik,AIC)
d
```

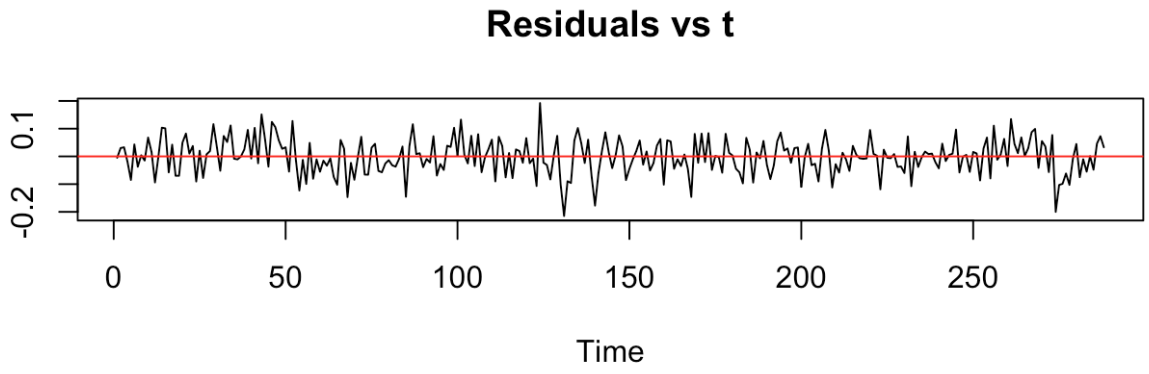
```
##          sigma2   loglik       AIC
## 1  0.004045279 365.9273 -715.8547
## 2  0.004058092 365.8086 -717.6172
## 3  0.004253167 359.7492 -707.4983
## 4  0.004592975 348.5836 -685.1671
## 5  0.005006833 336.5007 -663.0014
## 6  0.004731070 345.4669 -680.9337
## 7  0.006520623 301.1591 -594.3181
## 8  0.004442845 356.5567 -701.1134
## 9  0.004665449 350.2218 -690.4436
## 10 0.006203247 318.3259 -626.6518
## 11 0.006815926 304.8882 -601.7764
## 12 0.006266288 317.2271 -626.4543
## 13 0.008247147 278.1576 -550.3153
```

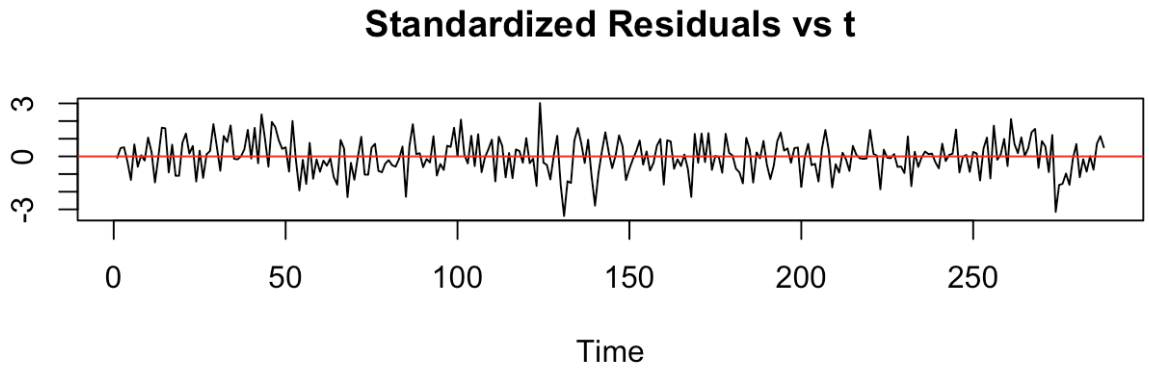
Comparing the values of Log Likelihood, AIC and sigma^2, it can be seen that the model m2 - (2,1,1) X (1,0,2) performs significantly better than other models. Compared to other models, the model has higher values of Log Likelihood and lower values for AIC and sigma^2.

Using appropriate formal and informal residual diagnostics, investigating whether m2 - (2,1,1) X (1,0,2) satisfies the following assumptions:

i. Zero-Mean

```
# Calculating Residuals
e <- m.ml.2$residuals # residuals
r <- e/sqrt(m.ml.2$sigma2) # standardized residuals

par(mfrow=c(2,1))
plot(e, main="Residuals vs t", ylab="")
abline(h=0, col="red")
plot(r, main="Standardized Residuals vs t", ylab="")
abline(h=0, col="red")
```





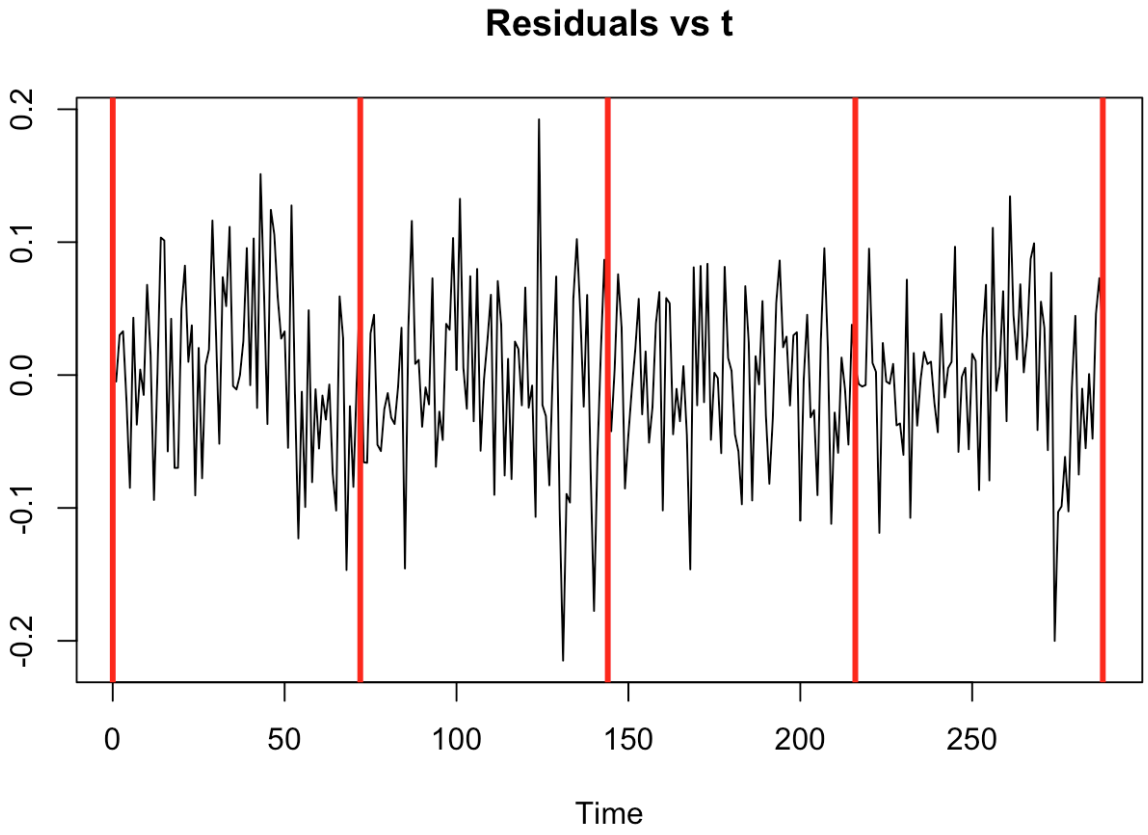

```
# test whether residuals have zero mean
t.test(e)
```

```
##
##  One Sample t-test
##
## data:  e
## t = -0.41902, df = 287, p-value = 0.6755
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.008958589  0.005813764
## sample estimates:
##    mean of x
## -0.001572413
```

The Zero-mean assumption seems satisfied using Informal residual diagnostics (Residuals / Standardized Residuals vs time plot). The p-value for the formal test (T-test) comes out to be greater than 0.05. Hence, we accept the null hypothesis that the expected value / true mean of residuals is equal to zero.

ii. Homoscedasticity

```
par(mfrow=c(1,1))
# 4 groups
plot(e, main="Residuals vs t", ylab="")
abline(v=c(0,72,144,216,288), lwd=3, col="red")
```



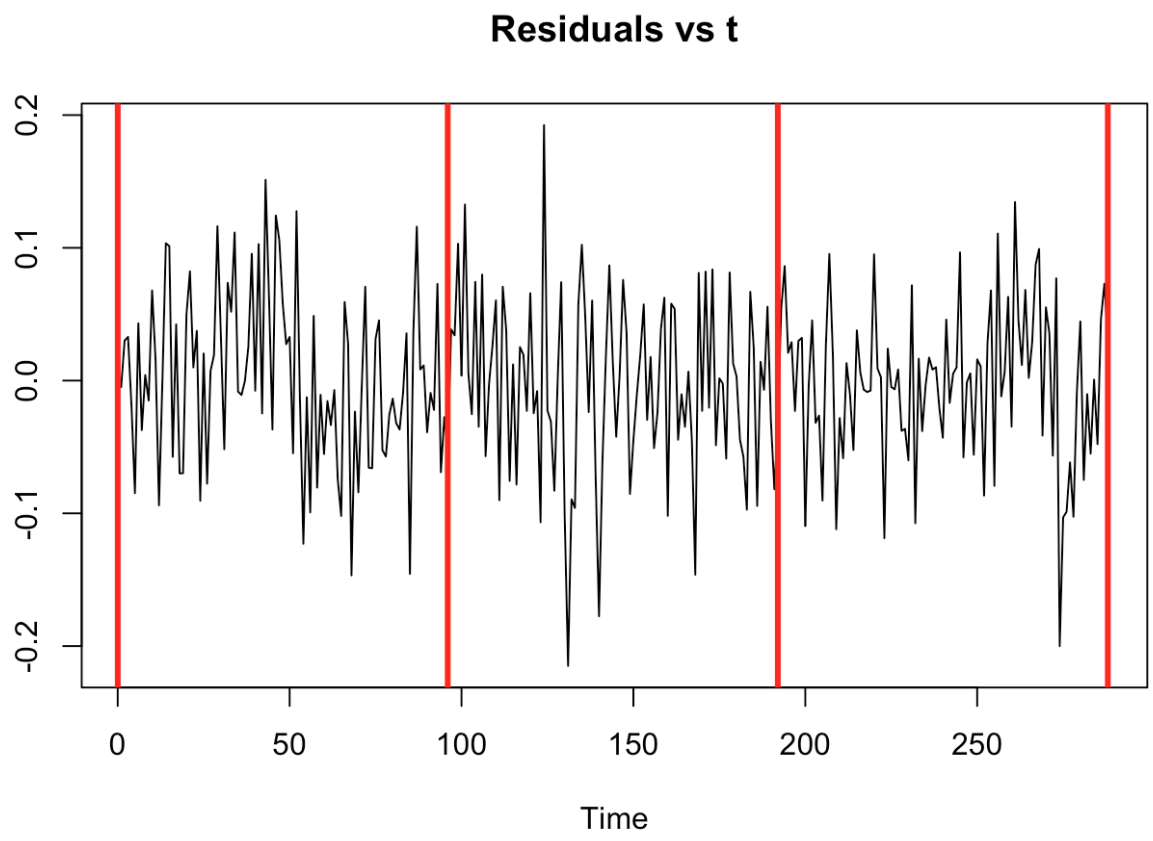

```
group <- c(rep(1,72),rep(2,72), rep(3,72), rep(4,72))
levene.test(e,group) #Levene
```

```
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  e
## Test Statistic = 1.4152, p-value = 0.2385
```

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

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 5.147, df = 3, p-value = 0.1613
```

```
# 3 groups
plot(e, main="Residuals vs t", ylab="")
abline(v=c(0,96,192,288), lwd=3, col="red")
```




```
group <- c(rep(1,96),rep(2,96), rep(3,96))
levene.test(e,group) #Levene
```

```
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  e
## Test Statistic = 1.2618, p-value = 0.2847
```

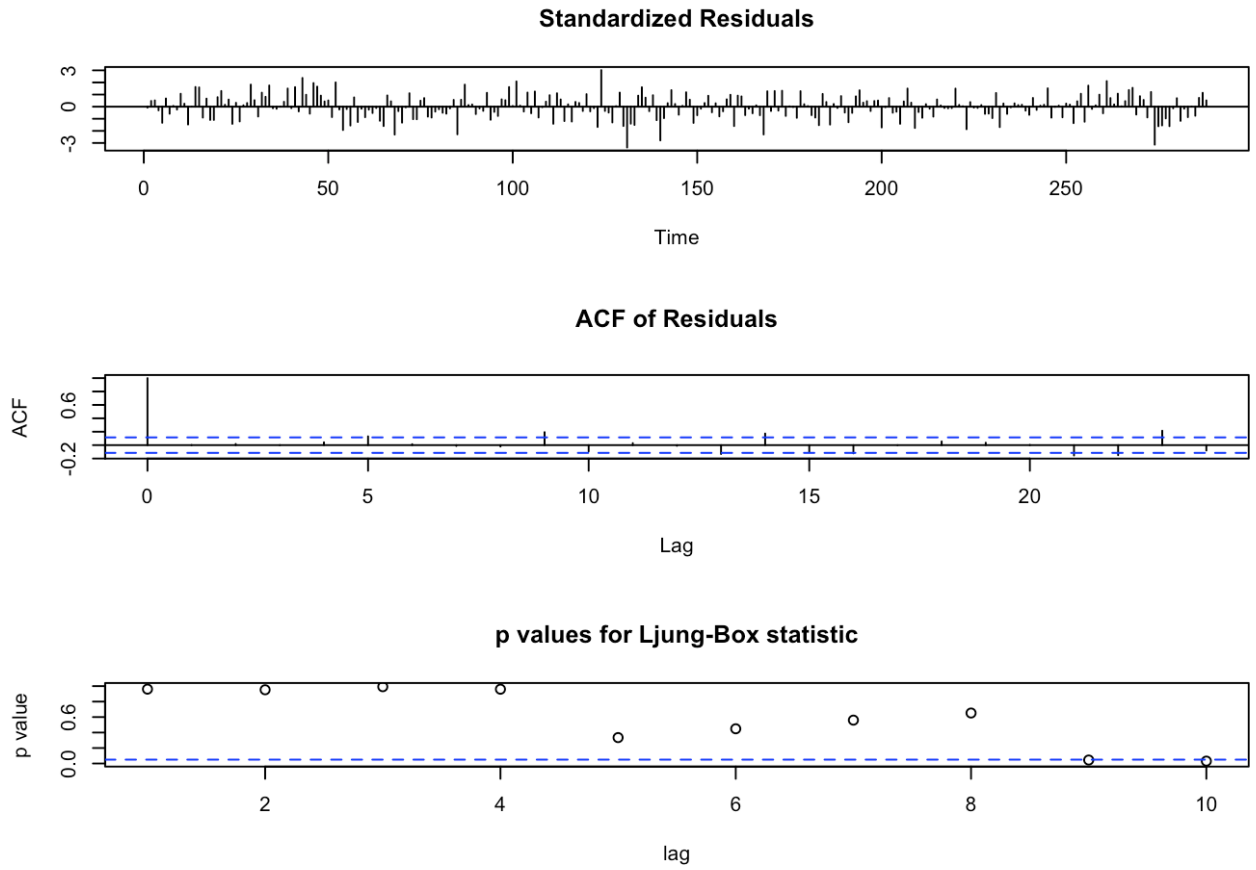
```
bartlett.test(e,group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 1.8624, df = 2, p-value = 0.3941
```

The Homoscedasticity assumption seems fine after taking the log. The assumption is also confirmed by Levene's and Bartlett's test.

iii. Zero-Correlation

```
tsdiag(m.ml.2) #ACF and Ljung-Box test all in one!
```



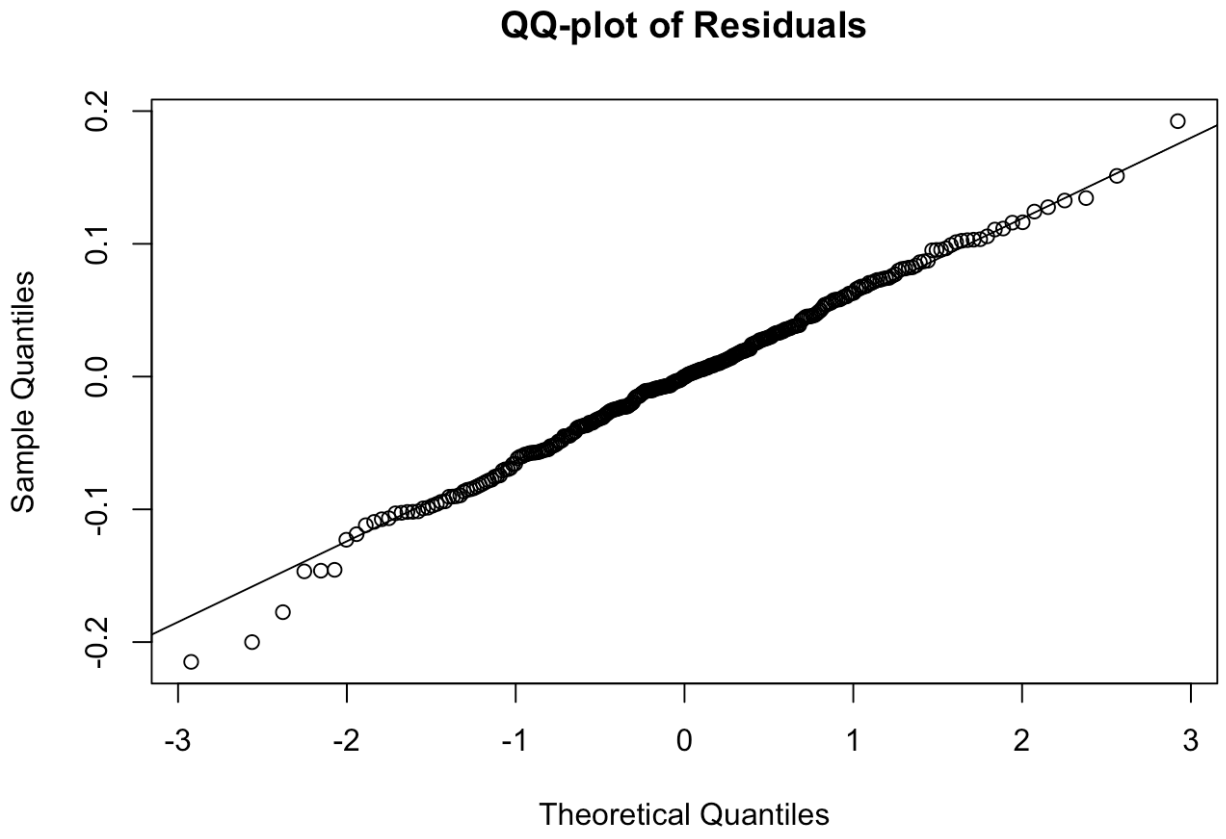

```
runs.test(e) #Runs test for randomness
```

```
##
##  Runs Test - Two sided
##
## data:  e
## Standardized Runs Statistic = 0.35417, p-value = 0.7232
```

The p-value for the formal 'Ljung-Box' test comes out to be less than 0.05 only for lags 9 and 10. For other lags, p-values are fine. The p-value for the formal 'Runs' test comes out to be greater than 0.05. Hence, we accept the null hypothesis that the residuals are uncorrelated.

iv. Normality

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




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

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.99603, p-value = 0.6828
```

Informal residual diagnostics (QQ Plot) indicates that the Normality assumption is valid as the sample quantiles mirror theroretical quantiles. The p-value for the formal test (Shapiro Wilk test) comes out to be greater than 0.05. Hence, we accept the null hypothesis that the residuals are normally distributed.

All the residuals diagnostics are satisfied for our chosen SARIMA model of Bankruptcy rates. We proceed to include co-variates in this model.

Looking at the correlations between co-variates and bankruptcy rates:

```
cor(data.frame(BR, UR, Pop, HPI))
```

```
##            BR         UR        Pop        HPI
## BR   1.000000 -0.3169070  0.8984050  0.6897080
## UR  -0.316907  1.0000000 -0.5431182 -0.5430593
## Pop  0.898405 -0.5431182  1.0000000  0.8601513
## HPI  0.689708 -0.5430593  0.8601513  1.0000000
```

Bankruptcy rate is highly correlated with population and HPI. Population is always on a constantly increasing trend and hence would not add any value in predicting Bankruptcy rate. So, population is excluded. Including both, HPI and Unemployment rate as co-variates in the SARIMA model and estimating the model equation again:

```
#Fit an SARIMA(2,1,1) X (1,0,2) s=12 model with covariate information
m.co.1 <- arima(log(BR), order = c(2,1,1), seasonal = list(order = c(1,0,2), pe
riod = 12), xreg = data.frame(UR, HPI))
m.co.1
```

```
##
## Call:
## arima(x = log(BR), order = c(2, 1, 1), seasonal = list(order = c(1, 0, 2), p
eriod = 12),
##     xreg = data.frame(UR, HPI))
##
## Coefficients:
##           ar1      ar2     ma1    sar1     sma1     sma2      UR      HPI
##       -1.0098  -0.5865  0.3362  0.9999  -0.6480  -0.3289  0.0114  -0.0284
## s.e.   0.0983   0.0584  0.1212  0.0004   0.0698   0.0648  0.0140   0.0062
##
## sigma^2 estimated as 0.003652:  log likelihood = 378.01,  aic = -738.02
```

The coefficient for Unemployment rate is not different than zero statistically since 95% confidence limit for coefficient includes zero. So, excluding Unemployment rate and estimating the model equation again:

```
m.co.2 <- arima(log(BR), order = c(2,1,1), seasonal = list(order = c(1,0,2), pe
riod = 12), xreg = data.frame(HPI))
m.co.2
```

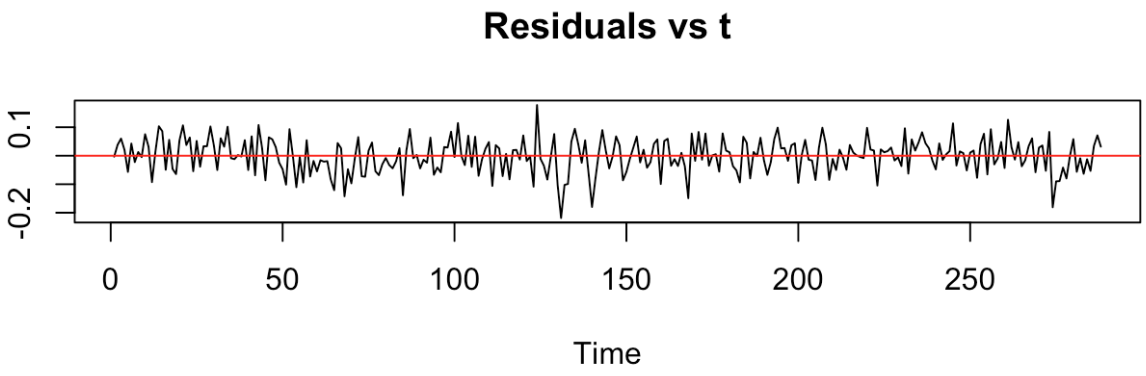
```
##
## Call:
## arima(x = log(BR), order = c(2, 1, 1), seasonal = list(order = c(1, 0, 2), p
eriod = 12),
##     xreg = data.frame(HPI))
##
## Coefficients:
##           ar1      ar2     ma1    sar1     sma1     sma2      HPI
##       -1.0142  -0.5852  0.3461  0.9999  -0.6435  -0.3319  -0.0302
## s.e.   0.0982   0.0582  0.1204  0.0005   0.0710   0.0651   0.0059
##
## sigma^2 estimated as 0.003665:  log likelihood = 377.68,  aic = -739.36
```

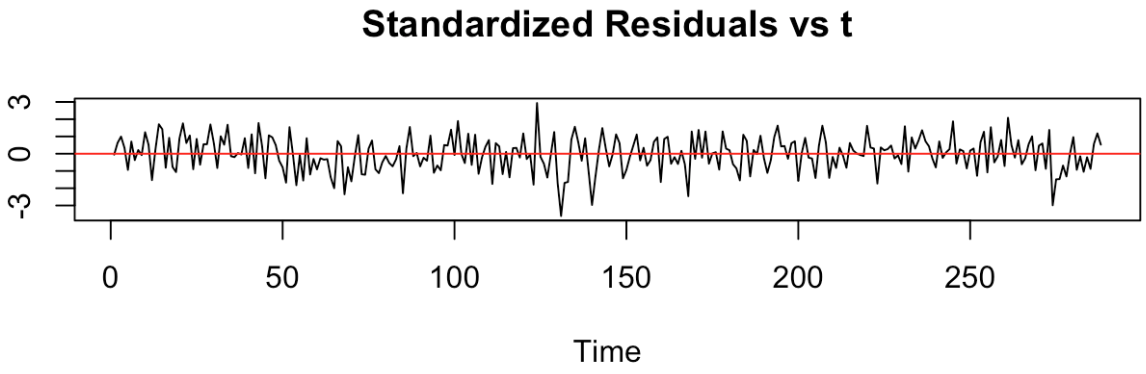
The model looks better than only SARIMA model. The statistics such as Log Likelihood, AIC and sigma^2 are better than only SARIMA model. All the coefficients including HPI's are significant. Hence, we finalise this model and perform residual diagnostics.

Using appropriate formal and informal residual diagnostics, investigating whether SARIMA(2,1,1) X (1,0,2) s=12 model with HPI as covariate satisfies the following assumptions:

i. Zero-Mean

```
# Calculating Residuals
e <- m.co.2$residuals # residuals
r <- e/sqrt(m.co.2$sigma2) # standardized residuals

par(mfrow=c(2,1))
plot(e, main="Residuals vs t", ylab="")
abline(h=0, col="red")
plot(r, main="Standardized Residuals vs t", ylab="")
abline(h=0, col="red")
```





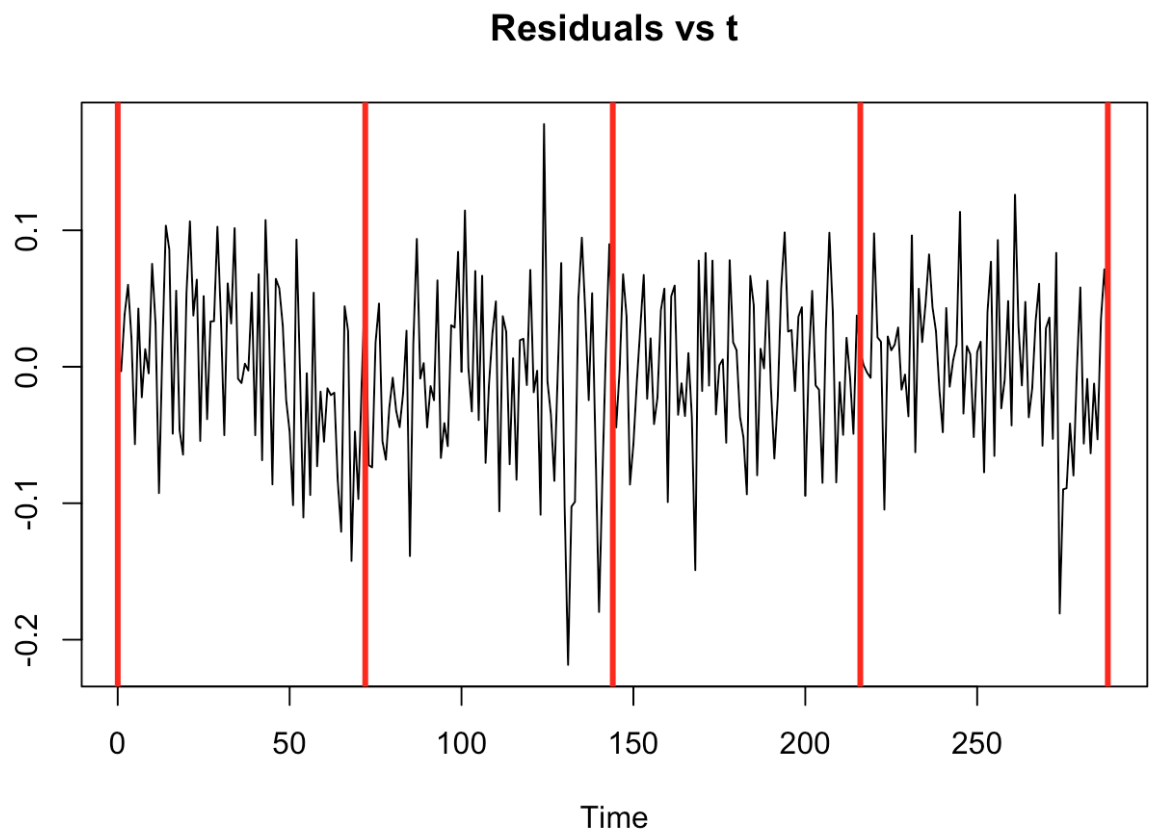

```
# test whether residuals have zero mean
t.test(e)
```

```
##
##  One Sample t-test
##
## data:  e
## t = -0.44576, df = 287, p-value = 0.6561
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.008608638  0.005429405
## sample estimates:
##    mean of x
## -0.001589616
```

The Zero-mean assumption seems satisfied using Informal residual diagnostics (Residuals / Standardized Residuals vs time plot). The p-value for the formal test (T-test) comes out to be greater than 0.05. Hence, we accept the null hypothesis that the expected value / true mean of residuals is equal to zero.

ii. Homoscedasticity

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

# 4 groups
plot(e, main="Residuals vs t", ylab="")
abline(v=c(0,72,144,216,288), lwd=3, col="red")
```



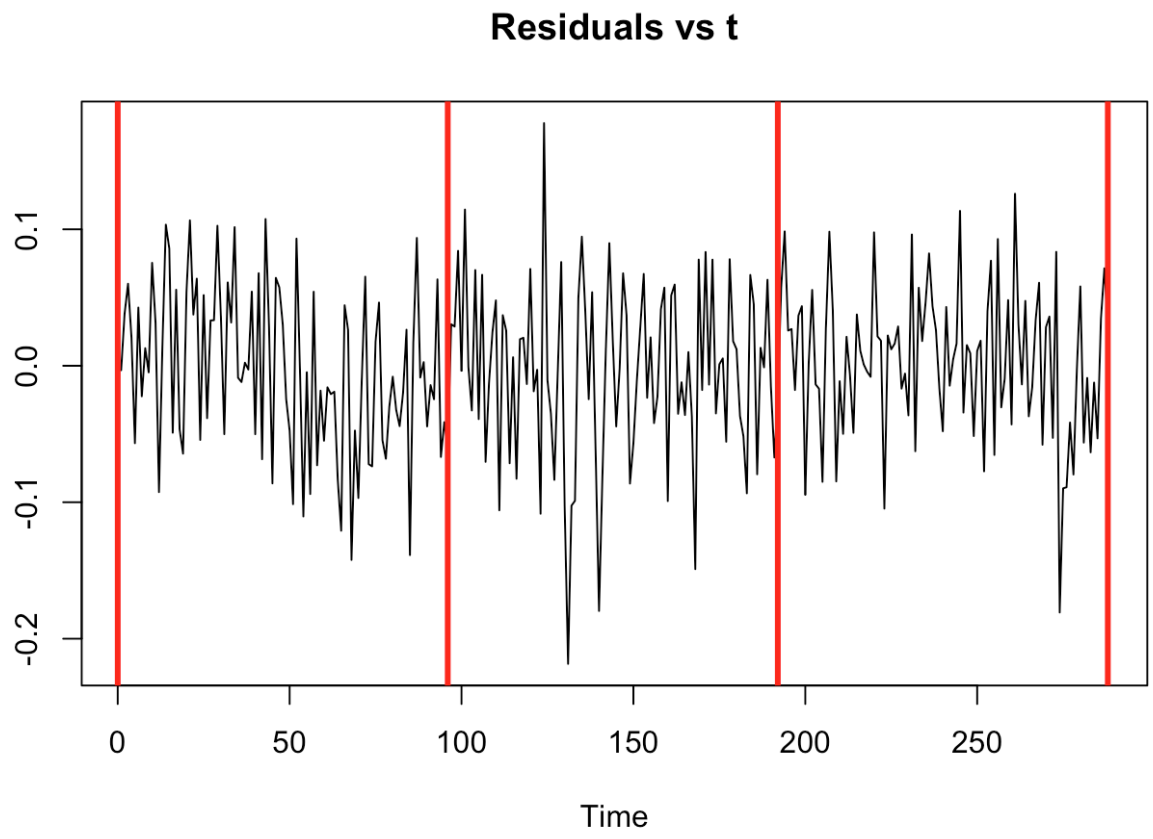
```
group <- c(rep(1,72),rep(2,72), rep(3,72), rep(4,72))
levene.test(e,group) #Levene
```

```
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  e
## Test Statistic = 1.4521, p-value = 0.2279
```

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

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 5.3685, df = 3, p-value = 0.1467
```

```
# 3 groups
plot(e, main="Residuals vs t", ylab="")
abline(v=c(0,96,192,288), lwd=3, col="red")
```



```
group <- c(rep(1,96),rep(2,96), rep(3,96))
levene.test(e,group) #Levene
```

```
##
##  modified robust Brown-Forsythe Levene-type test based on the
##  absolute deviations from the median
##
## data:  e
## Test Statistic = 1.3462, p-value = 0.2619
```

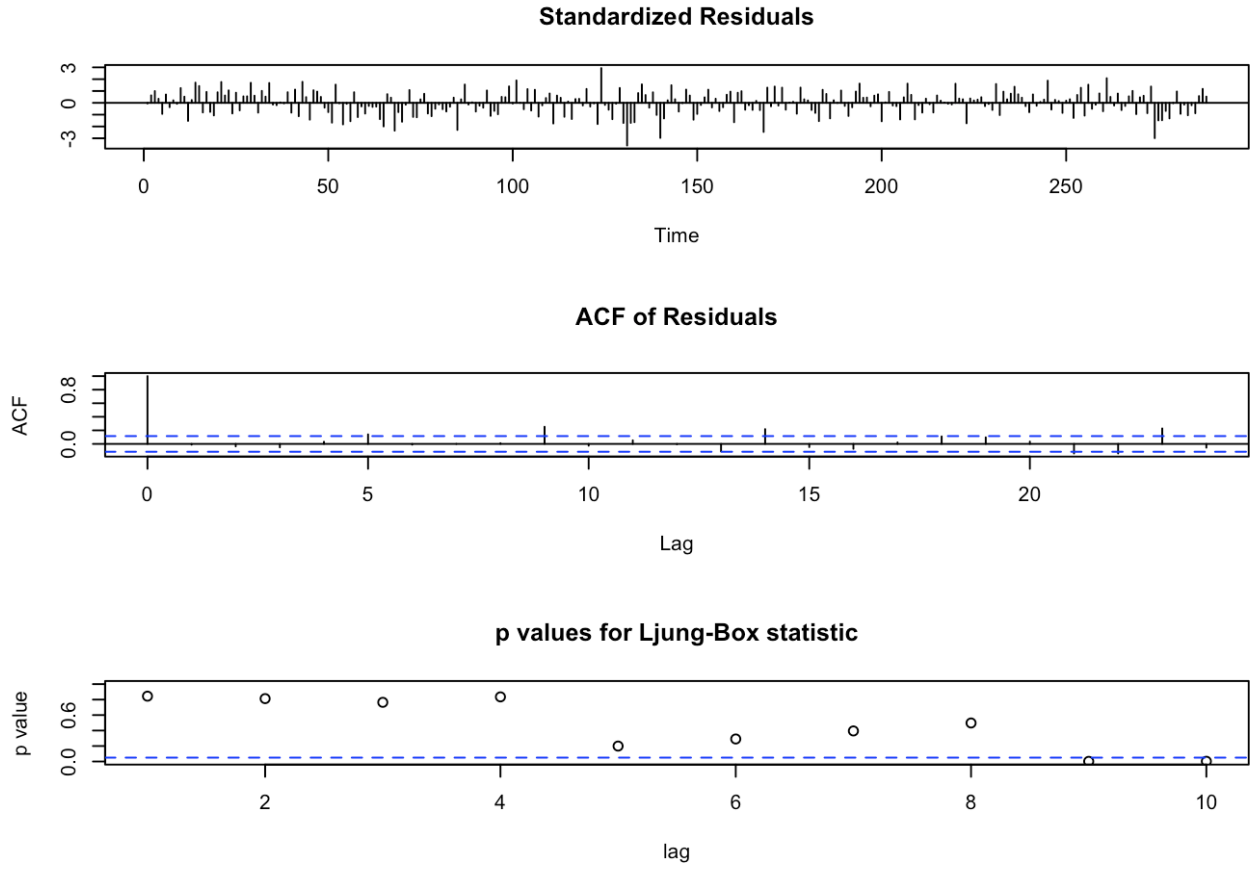
```
bartlett.test(e,group) #Bartlett
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  e and group
## Bartlett's K-squared = 3.152, df = 2, p-value = 0.2068
```

The Homoscedasticity assumption seems fine after including HPI. The assumption is also confirmed by Levene's and Bartlett's test using 2 different group sizes.

iii. Zero-Correlation

```
tsdiag(m.co.2) #ACF and Ljung-Box test all in one!
```



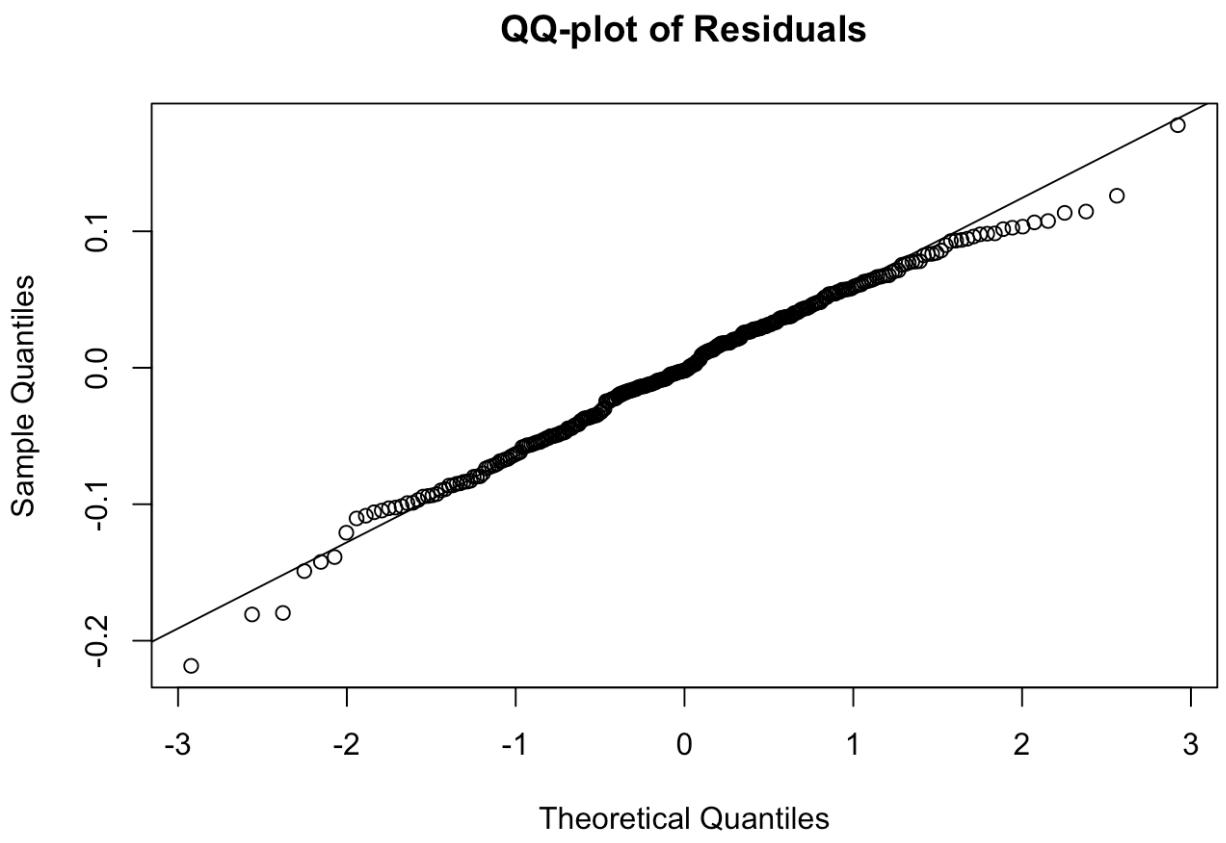
```
runs.test(e) #Runs test for randomness
```

```
##
##  Runs Test - Two sided
##
## data:  e
## Standardized Runs Statistic = 0.59028, p-value = 0.555
```

The p-value for the formal ‘Ljung-Box’ test comes out to be less than 0.05 only for lags 9 and 10. For other lags, p-values are fine. The p-value for the formal ‘Runs’ test comes out to be greater than 0.05. Hence, we accept the null hypothesis that the residuals are uncorrelated.

iv. Normality

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




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

```
##
##  Shapiro-Wilk normality test
##
## data:  e
## W = 0.99142, p-value = 0.09222
```

Informal residual diagnostics (QQ Plot) indicates that the Normality assumption is valid as the sample quantiles mirror theroretical quantiles except for few deviations at the tail. The p-value for the formal test (Shapiro Wilk test) comes out to be greater than 0.05. Hence, we accept the null hypothesis that the residuals are normally distributed.

All the residuals diagnostics are satisfied for our chosen SARIMA model with HPI as a covariate.

Now, we look at the RMSE (root-mean-square error) for the chosen SARIMA model with and without HPI as a covariate. For this, we train our model on 90% data points and test our model on the remaining 10% data points and measure the error in prediction.

```
train.index <- c(1:round(0.9*length(BR)))
BR.test <- BR[-train.index]
BR.train <- BR[train.index]
HPI.test <- HPI[-train.index]
HPI.train <- HPI[train.index]

l.BR.test <- log(BR.test)
l.BR.train <- log(BR.train)
l.HPI.test <- log(HPI.test)
l.HPI.train <- log(HPI.train)

pred.result <- c()
pred.result.df <- data.frame()
for(i in 1:length(BR.test)){
    m.3 <- arima(l.BR.train, order = c(2,1,1),
                    seasonal = list(order = c(1,0,2), period = 12))
    pred.l.BR <- predict(m.3, n.ahead = 1)$pred
    se <- predict(m.3, n.ahead = 1)$se
    upper <- pred.l.BR + 1.96*se
    lower <- pred.l.BR - 1.96*se
    l.BR.train <- c(l.BR.train, pred.l.BR)
    pred.result <- c(pred.l.BR, lower, upper)
    pred.result.df <- rbind(pred.result.df, pred.result)

}
pred.result.3 <- exp(pred.result.df)
names(pred.result.3) <- c("lower", "mean", "upper")
rmse.3 <- sqrt(mean(pred.result.3$mean - BR.test)^2)
rmse.3
```

```
## [1] 0.007074261
```

Then let's take a look at the model with covariate HPI. We choose the log of HPI because log(HPI) has higher correlation with BR than HPI.

```
l.BR.test <- log(BR.test)
l.BR.train <- log(BR.train)
l.HPI.test <- log(HPI.test)
l.HPI.train <- log(HPI.train)

pred.result <- c()
pred.result.df <- data.frame()
for(i in 1:length(BR.test)){
    m.co.3 <- arima(l.BR.train, order = c(2,1,1),
                    seasonal = list(order = c(1,0,2), period = 12),
                    xreg = data.frame(l.HPI.train))
    pred.l.BR <- predict(m.co.3, n.ahead = 1, newxreg = l.HPI.test[i])$pred
    se <- predict(m.co.3, n.ahead = 1, newxreg = l.HPI.test[i])$se
    upper <- pred.l.BR + 1.96*se
    lower <- pred.l.BR - 1.96*se
    l.BR.train<- c(l.BR.train, pred.l.BR)
    l.HPI.train <- c(l.HPI.train, l.HPI.test[i])
    pred.result <- c(lower, pred.l.BR, upper)
    pred.result.df <- rbind(pred.result.df, pred.result)

}
pred.result.c <- exp(pred.result.df)
names(pred.result.c) <- c("lower", "mean", "upper")
rmse.c <- sqrt(mean(pred.result.c$mean - BR.test)^2)
rmse.c
```

```
## [1] 0.0003701666
```

SARIMA model with HPI as a covariate has a lower RMSE than an only SARIMA model. So, we choose SARIMA model with HPI as a covariate for forecasting

Prediction

Loading the test dataset

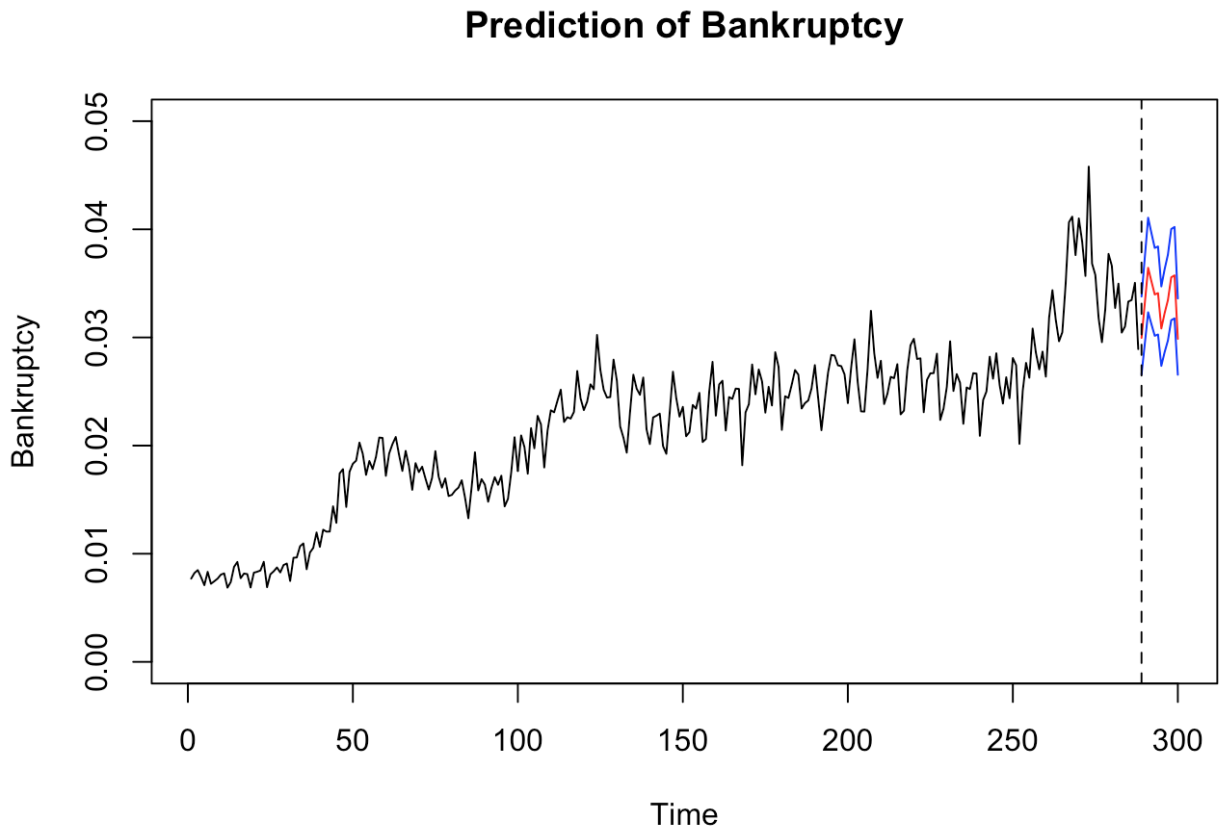
```
test <- read.csv('/Users/tracy/msan-ts/project/test.csv', header = TRUE)
train <- read.csv('/Users/tracy/msan-ts/project/train.csv', header=TRUE)
```

Now, we forecast Bankruptcy rates using the chosen model. We use the rolling window to make the predictions.

```
BR.train <- train$Bankruptcy_Rate
l.BR.train <- log(BR.train)

HPI.train <- train$House_Price_Index
l.HPI.train <- log(HPI.train)

HPI.test <- test$House_Price_Index
l.HPI.test <- log(HPI.test)

HPI <- ts(train$House_Price_Index)

pred.result.df <- data.frame()
for(i in 1:length(HPI.test)){
    m.co.3 <- arima(l.BR.train, order = c(2,1,1),
                    seasonal = list(order = c(1,0,2), period = 12),
                    xreg = data.frame(l.HPI.train))
    pred.l.BR <- predict(m.co.3, n.ahead = 1, newxreg = l.HPI.test[i])$pred
    se <- predict(m.co.3, n.ahead = 1, newxreg = l.HPI.test[i])$se
    upper <- pred.l.BR + 1.96*se
    lower <- pred.l.BR - 1.96*se
    l.BR.train<- c(l.BR.train, pred.l.BR)
    l.HPI.train <- c(l.HPI.train, l.HPI.test[i])
    pred.result <- c(pred.l.BR, lower, upper)
    pred.result.df <- rbind(pred.result.df, pred.result)

}
pred.result.c <- exp(pred.result.df)
names(pred.result.c) <- c("mean", "lower", "upper")

BR.train <- as.ts(BR.train)
t.test <- c(289:300)
```

Plot the prediction of bankruptcy.

```
par(mfrow=c(1,1))
plot(BR.train, xlim=c(1,300), ylim=c(0,0.05), ylab = "Bankruptcy", main = "Pred
iction of Bankruptcy")

lines(pred.result.c$mean~t.test, col='red')
lines(pred.result.c$lower~t.test, col='blue')
lines(pred.result.c$upper~t.test, col='blue')
abline(v=289, lty=2)
```



The prediction results.

```
pred.result.c
```

```
##          mean      lower      upper
## 1  0.02995982 0.02656665 0.03378637
## 2  0.03319114 0.02943894 0.03742158
## 3  0.03643629 0.03232468 0.04107087
## 4  0.03518602 0.03122348 0.03965143
## 5  0.03397475 0.03015488 0.03827851
## 6  0.03409939 0.03027207 0.03841061
## 7  0.03082918 0.02737474 0.03471954
## 8  0.03224749 0.02864027 0.03630904
## 9  0.03346011 0.02972342 0.03766656
## 10 0.03556058 0.03159570 0.04002301
## 11 0.03573947 0.03176185 0.04021522
## 12 0.02987100 0.02655187 0.03360504
```