Time Series Analysis final Project - Competitive

MATH 1318 Time Series Analysis Final Project

Team Members: 1.Rahul k. gupta (s3635232) 2.Terrie christensen (s3664899) 3.Napapach Dechawatthanaphokin (s3613572)

14 May 2018

Contents

1 Introduction

Bitcoin or BTC is a digital currency, otherwise known as a cryptocurrency. it was introduced to markets in 2009 by 'Satoshi Nakamoto.' The most notable aspect of BTC is that no banks or financial institutions are needed to facilitate trades. Additionally, it runs as a virtually anonymous financial system whereby buyers and sellers do not need to input their names, addresses or any other personally identifiable information to transfer BTC.

Rahul Made these changes

2 Model Diagnosis

```
# Import Libraries
library(TSA)
library(fUnitRoots)
library(forecast)
library(CombMSC)
library(lmtest)
library(fGarch)
library(rugarch)
library(zoo)
library(ggplot2)
require(readr)
```

```
residual.analysis <- function(model, std = TRUE){
  library(TSA)
  library(FitAR)
  if (std == TRUE){
    res.model = rstandard(model)
  }else{
    res.model = residuals(model)
  }
  par(mfrow=c(3,2))
  plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of standardised residual
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  acf(res.model,main="ACF of standardised residuals")
  print(shapiro.test(res.model))
  k=0
  LBQPlot(res.model, lag.max = length(model$residuals)-1 , StartLag = k + 1, k = 0, SquaredQ = FALSE)
  par(mfrow=c(1,1))
}
```

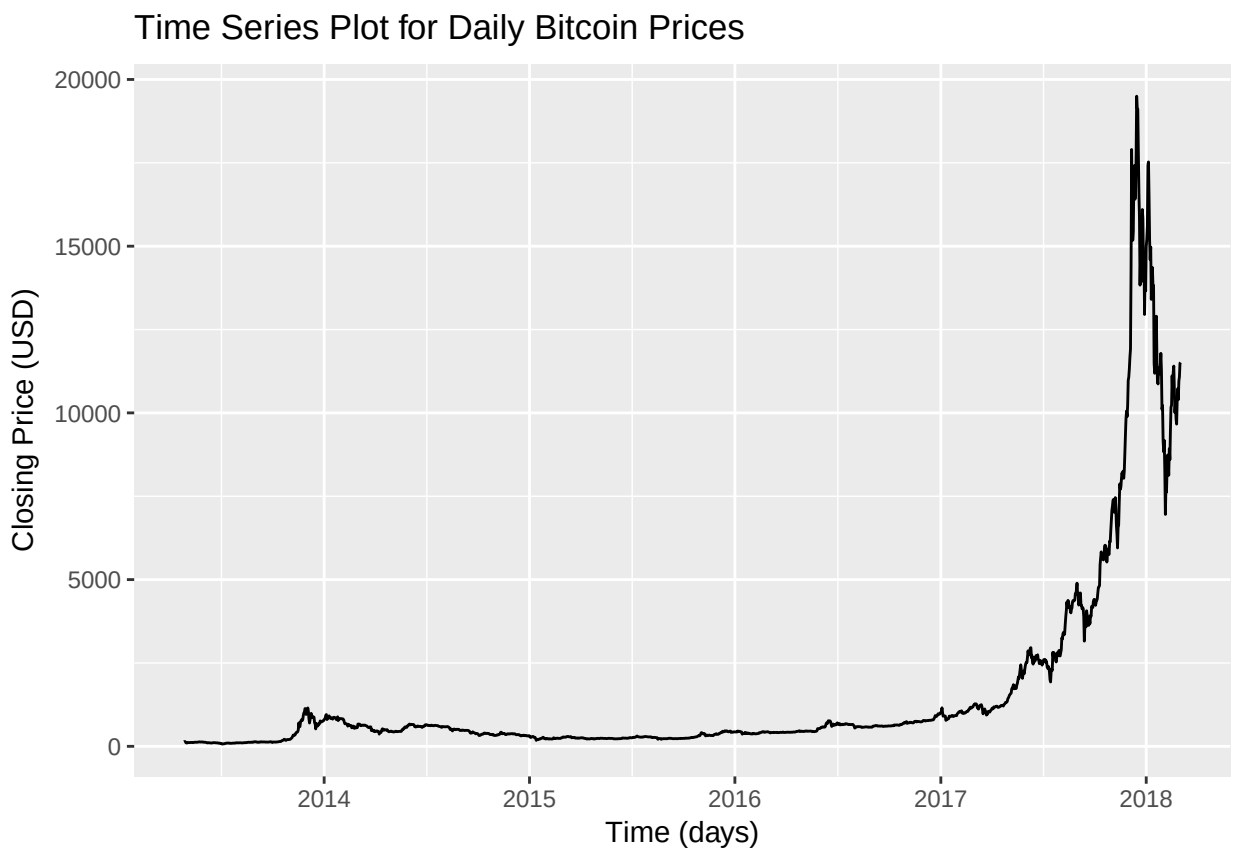
```
Bitcoin <- read.csv("../data/Bitcoin_Historical_Price.csv", header=TRUE)
Bitcoin$Date = as.Date(Bitcoin$Date,'%Y-%m-%d')
```

```
Bitcoin.zoo <- zoo(Bitcoin$Close, Bitcoin$Date)
class(Bitcoin.zoo)
```

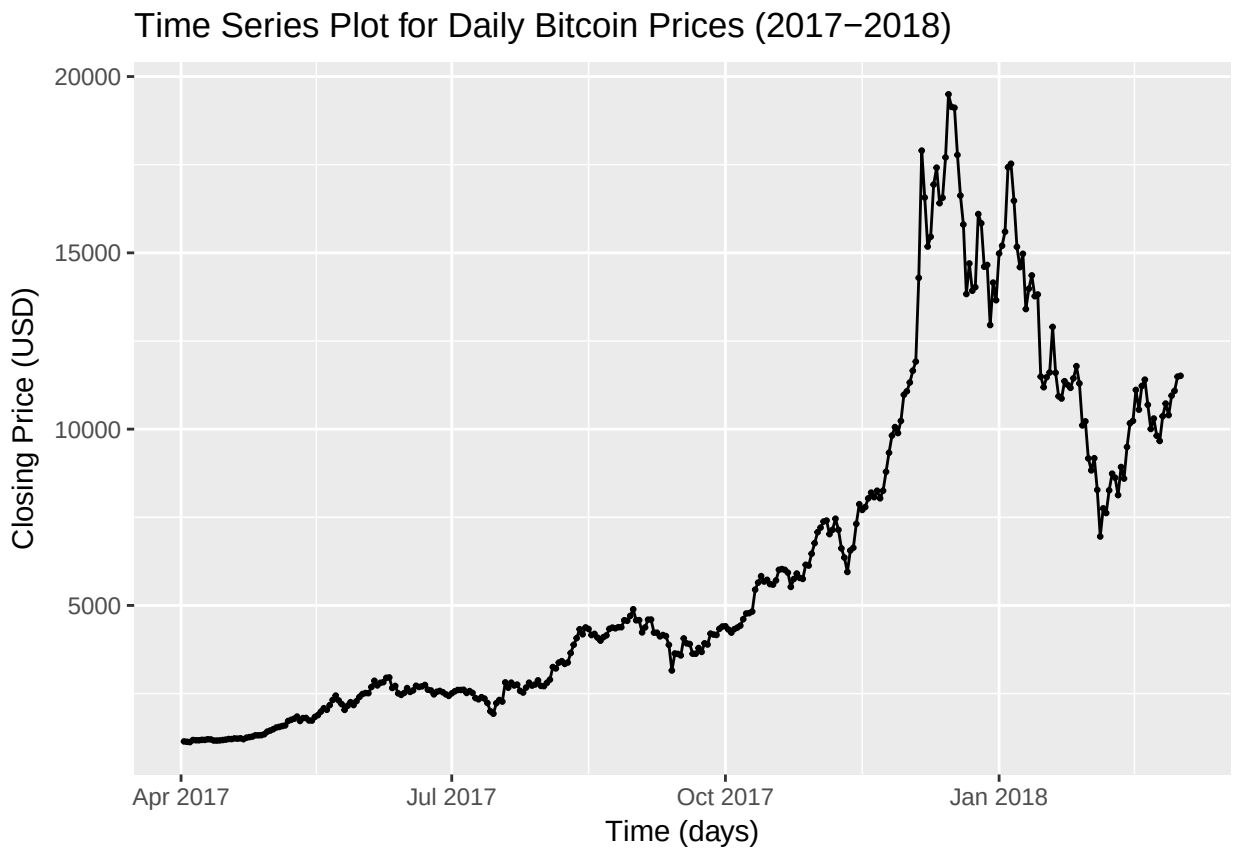
```
## [1] "zoo"
```

```
Bitcoin.raw = Bitcoin.zoo
```

```
autoplot.zoo(Bitcoin.zoo) +
  ylab('Closing Price (USD)') +
  xlab('Time (days)') +
  ggtitle("Time Series Plot for Daily Bitcoin Prices")
```



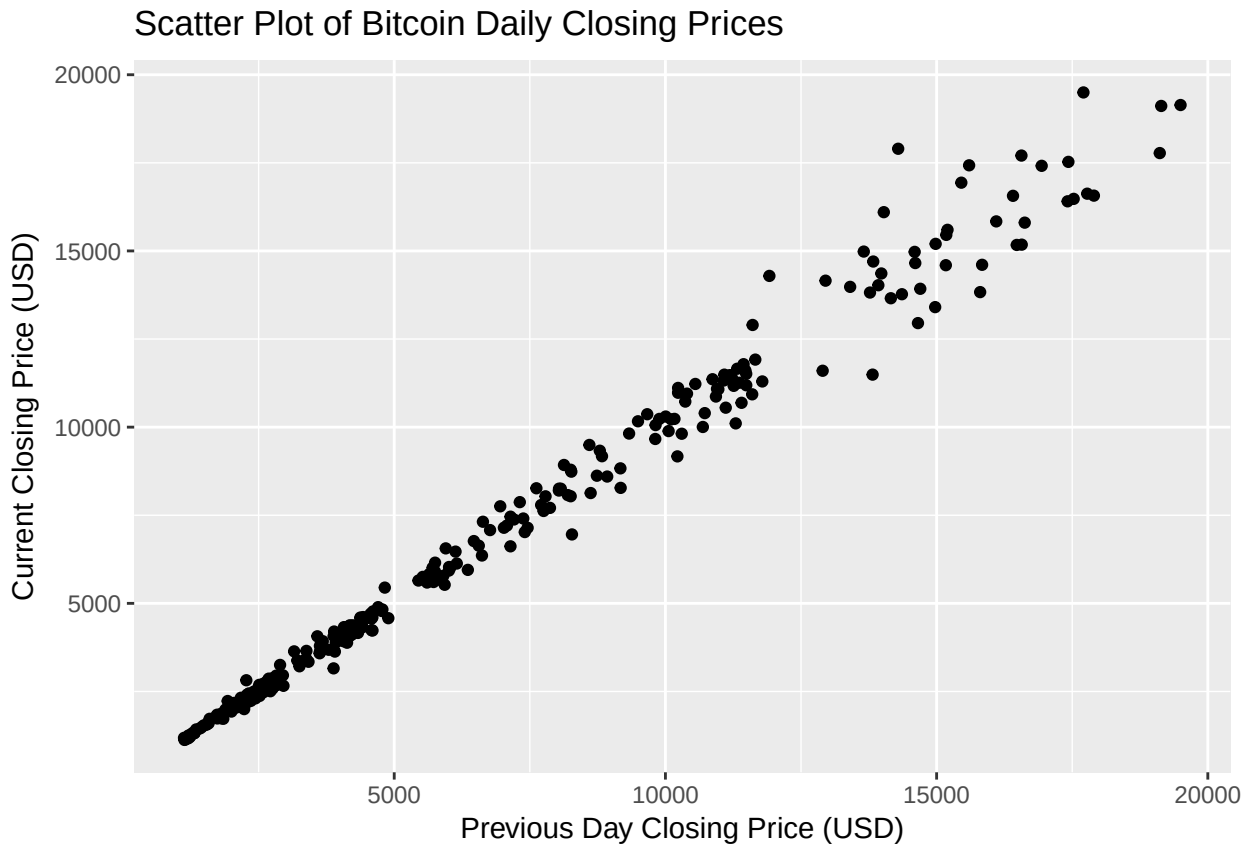

```
Bitcoin.2017 = Bitcoin[Bitcoin$Date > as.Date("2017-04-01"),]
Bitcoin.2017.zoo = zoo(Bitcoin.2017$Close, Bitcoin.2017$Date)
autoplot(Bitcoin.2017.zoo) +
  geom_point(size=.5) +
  ylab('Closing Price (USD)') +
  xlab('Time (days)') +
  ggtitle("Time Series Plot for Daily Bitcoin Prices (2017-2018)")
```



2.1 Scatter Plot and correlation

```
ggplot(Bitcoin.2017,aes(zlag(Close), Close)) + geom_point() +
  ylab('Current Closing Price (USD)') +
  xlab('Previous Day Closing Price (USD)') +
  ggtitle("Scatter Plot of Bitcoin Daily Closing Prices")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
y = as.vector(Bitcoin.2017.zoo)
x = zlag(Bitcoin.2017.zoo)
index = 2:length(x)
cor(y[index],x[index])
```

```
## [1] 0.9935557
```

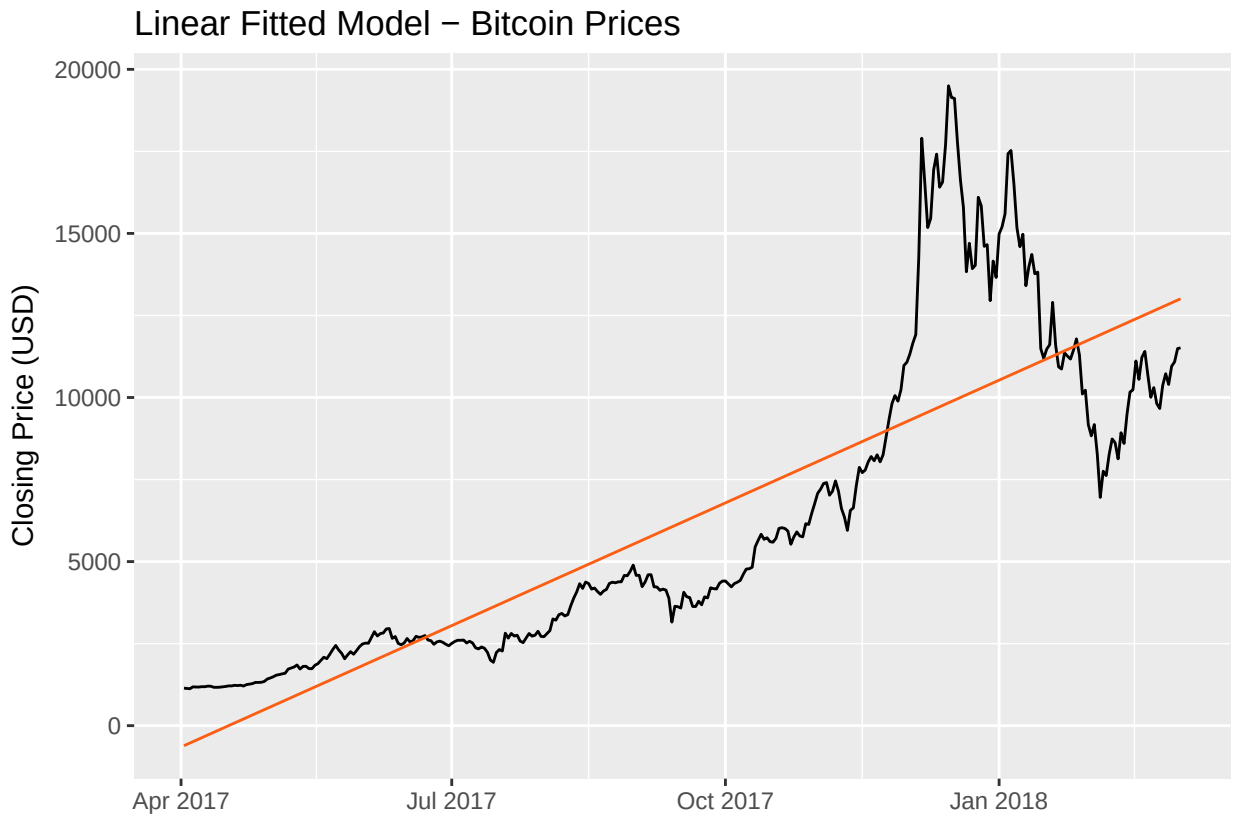
2.2 Linear Model

```
model.ln = lm(Bitcoin.2017.zoo~time(Bitcoin.2017.zoo)) # label the linear trend model as model.ln
summary(model.ln)
```

```
##
## Call:
## lm(formula = Bitcoin.2017.zoo ~ time(Bitcoin.2017.zoo))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4954.5 -1579.6  -668.9   881.2  9660.6
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -7.021e+05  2.461e+04  -28.53   <2e-16 ***
## time(Bitcoin.2017.zoo)  4.065e+01  1.412e+00   28.79   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2511 on 334 degrees of freedom
## Multiple R-squared:  0.7127, Adjusted R-squared:  0.7119
## F-statistic: 828.6 on 1 and 334 DF,  p-value: < 2.2e-16
```

```
ggplot(Bitcoin.2017,aes(Date,Close))+
  geom_line()  +
  ylab('Closing Price (USD)') +
  xlab('') +
  ggtitle('Linear Fitted Model - Bitcoin Prices') +
  geom_line(aes(y=fitted(model.ln)),color='#fc5e13')
```



2.3 Residual Analysis - Linear Model

Below are the findings of residuals from linear model

```
residual_analysis_qq <- function(myresiduals, title = 'QQ Plot of Residuals') {
data=as.data.frame(qqnorm( myresiduals , plot=F))
ggplot(data,aes(x,y)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE, color='#e36209', size=.4)+
  xlab('Theoretical') +
  ylab('Sample') +
  ggtitle(title)
```

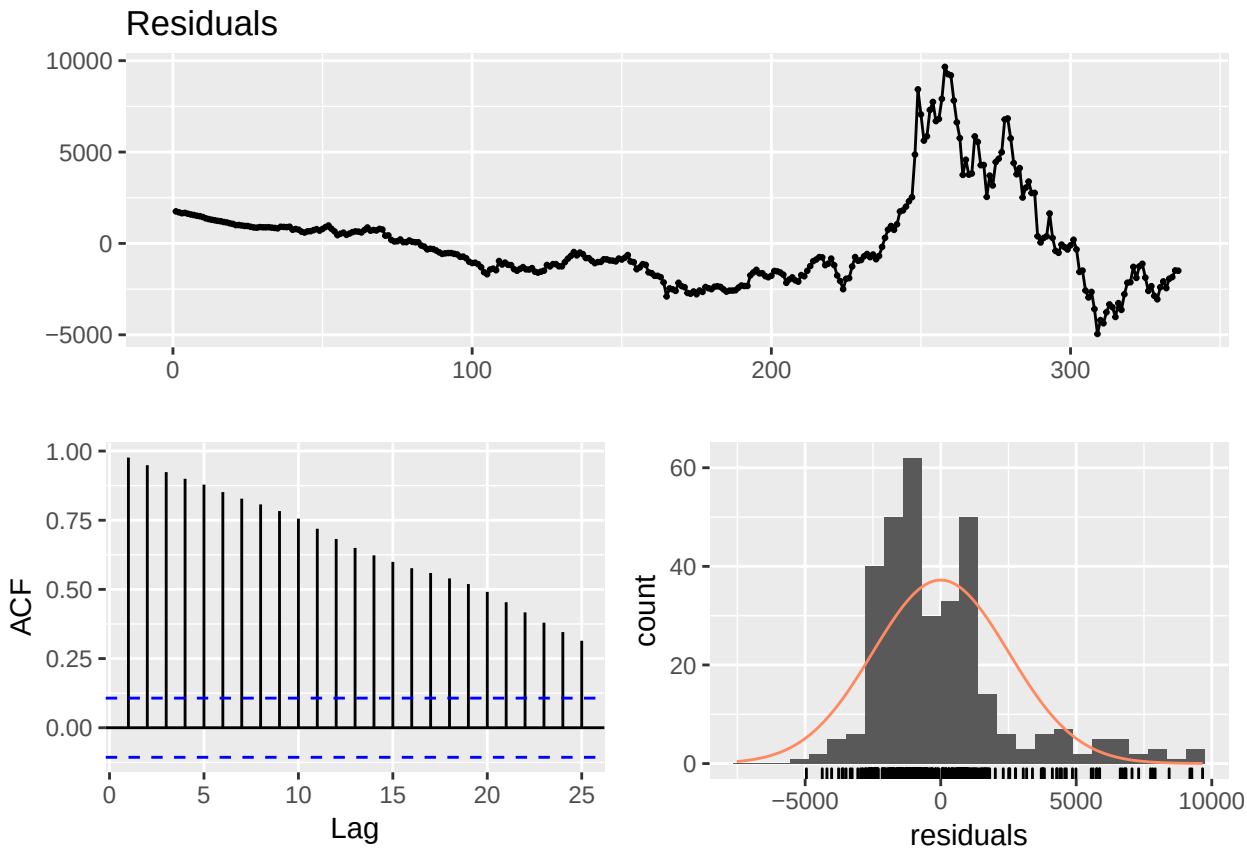


Figure 1: Residual Analysis Linear fitted Model

```
}

checkresiduals(model.ln)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 321.71, df = 10, p-value < 2.2e-16
```

```
residual_analysis_qq(residuals(model.ln))
```

```
shapiro.test(as.vector(residuals(model.ln)))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.vector(residuals(model.ln))
## W = 0.87841, p-value = 1.204e-15
```

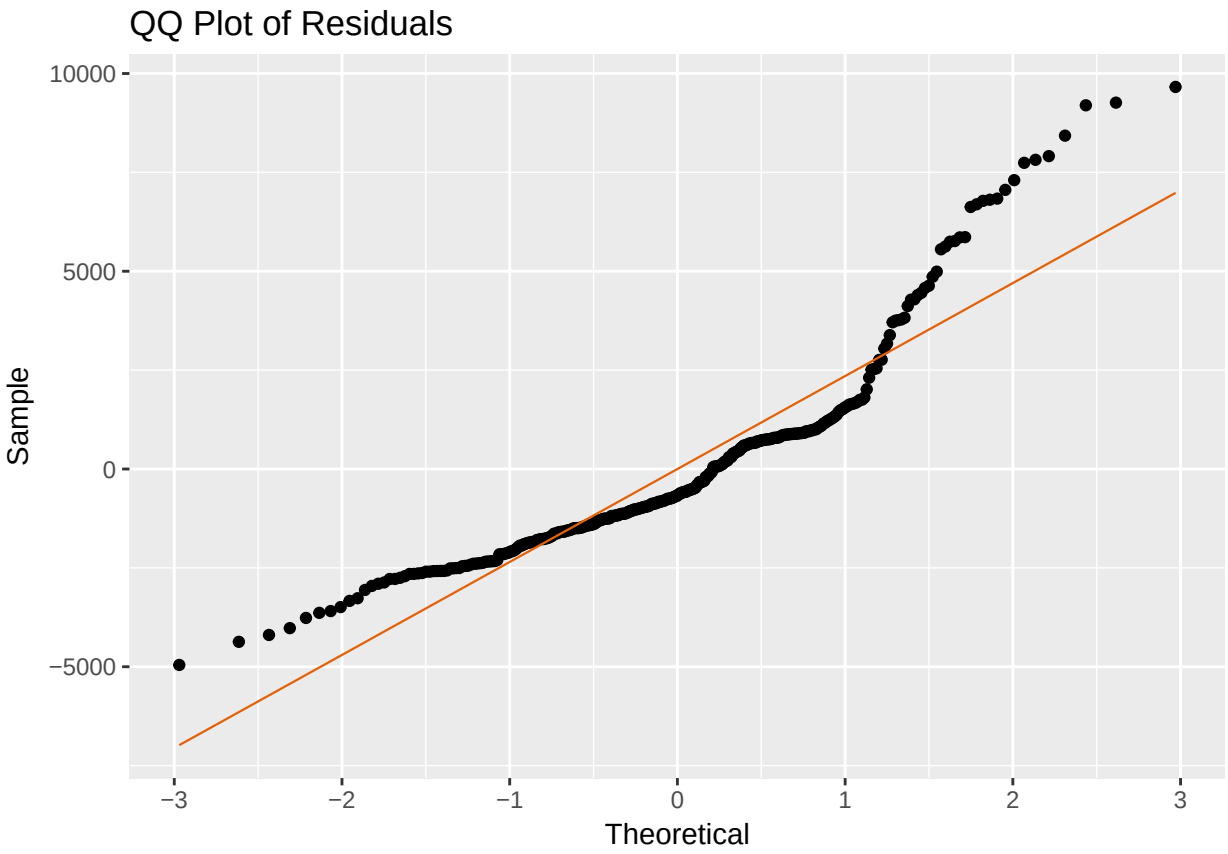


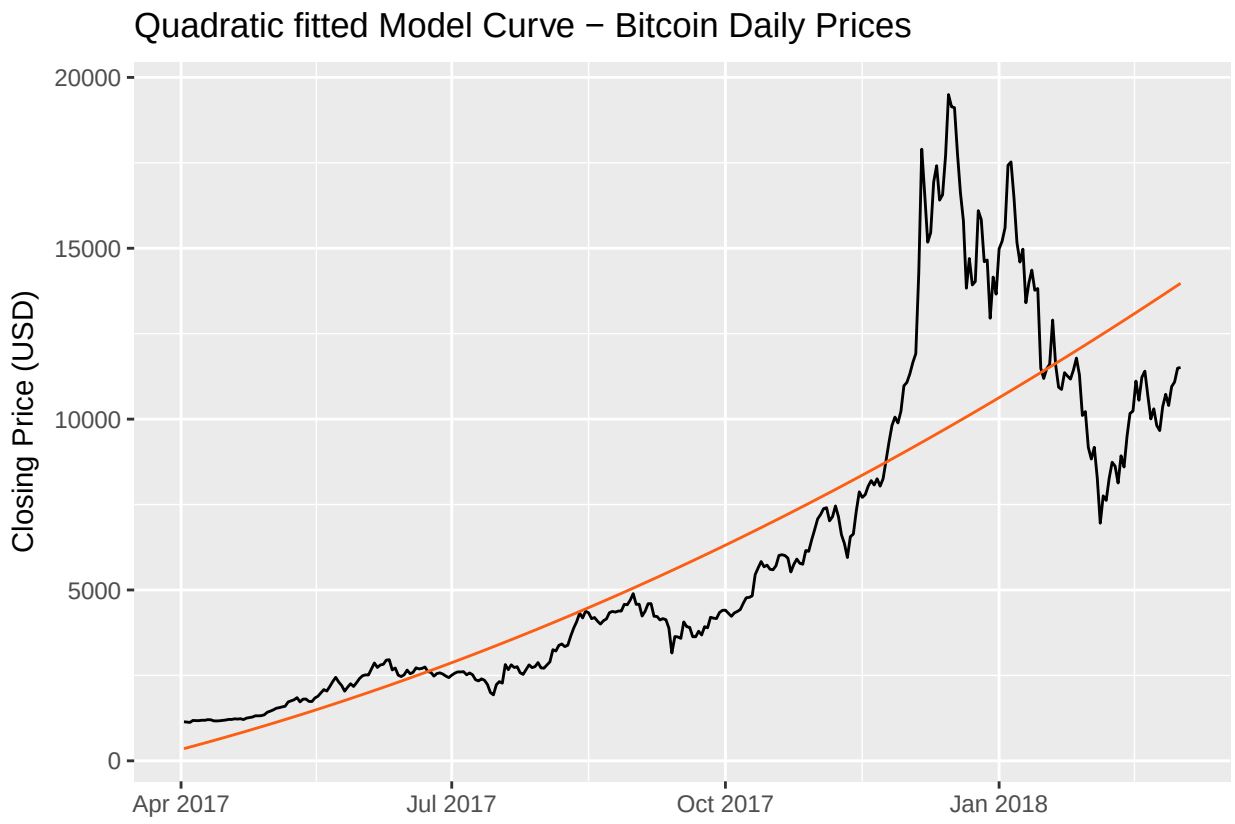
Figure 2: Residual Analysis Linear fitted Model

2.4 Quadratic Model

```
t = as.vector(time(Bitcoin.2017.zoo))
t2 = t^2
model.qa = lm(Bitcoin.2017.zoo~ t + t2) # label the quadratic trend model as model.qa
summary(model.qa)
```

```
##
## Call:
## lm(formula = Bitcoin.2017.zoo ~ t + t2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5490.1 -1286.7  -408.4   497.0  9733.1
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.504e+07  4.874e+06   3.085  0.00221 **
## t           -1.766e+03  5.594e+02  -3.156  0.00174 **
## t2           5.183e-02  1.605e-02   3.229  0.00137 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2476 on 333 degrees of freedom
## Multiple R-squared:  0.7214, Adjusted R-squared:  0.7198
## F-statistic: 431.2 on 2 and 333 DF,  p-value: < 2.2e-16
```

```
ggplot(Bitcoin.2017,aes(Date,Close))+
  geom_line()  +
  ylab('Closing Price (USD)') +
  xlab('') +
  ggtitle('Quadratic fitted Model Curve - Bitcoin Daily Prices') +
  geom_line(aes(y=fitted(model.qa)),color='#fc5e13')
```



2.5 Residual Analysis - Linear Model

Below are the findings of residuals from linear model

```
checkresiduals(model.qa)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 321.7, df = 10, p-value < 2.2e-16
```

```
residual_analysis_qq(residuals(model.qa))
```

```
shapiro.test(as.vector(residuals(model.qa)))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  as.vector(residuals(model.qa))
## W = 0.86085, p-value < 2.2e-16
```

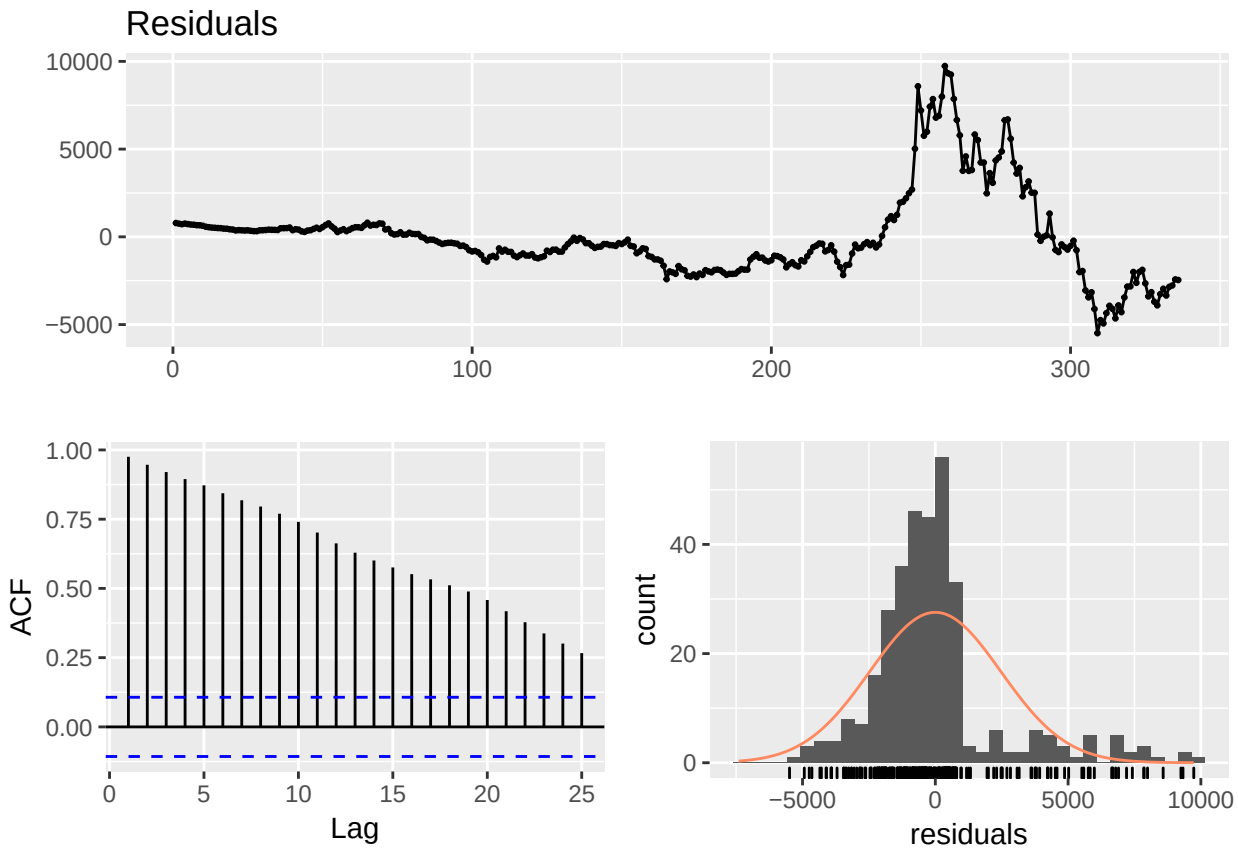


Figure 3: Residual Analysis Quadratic fitted Model

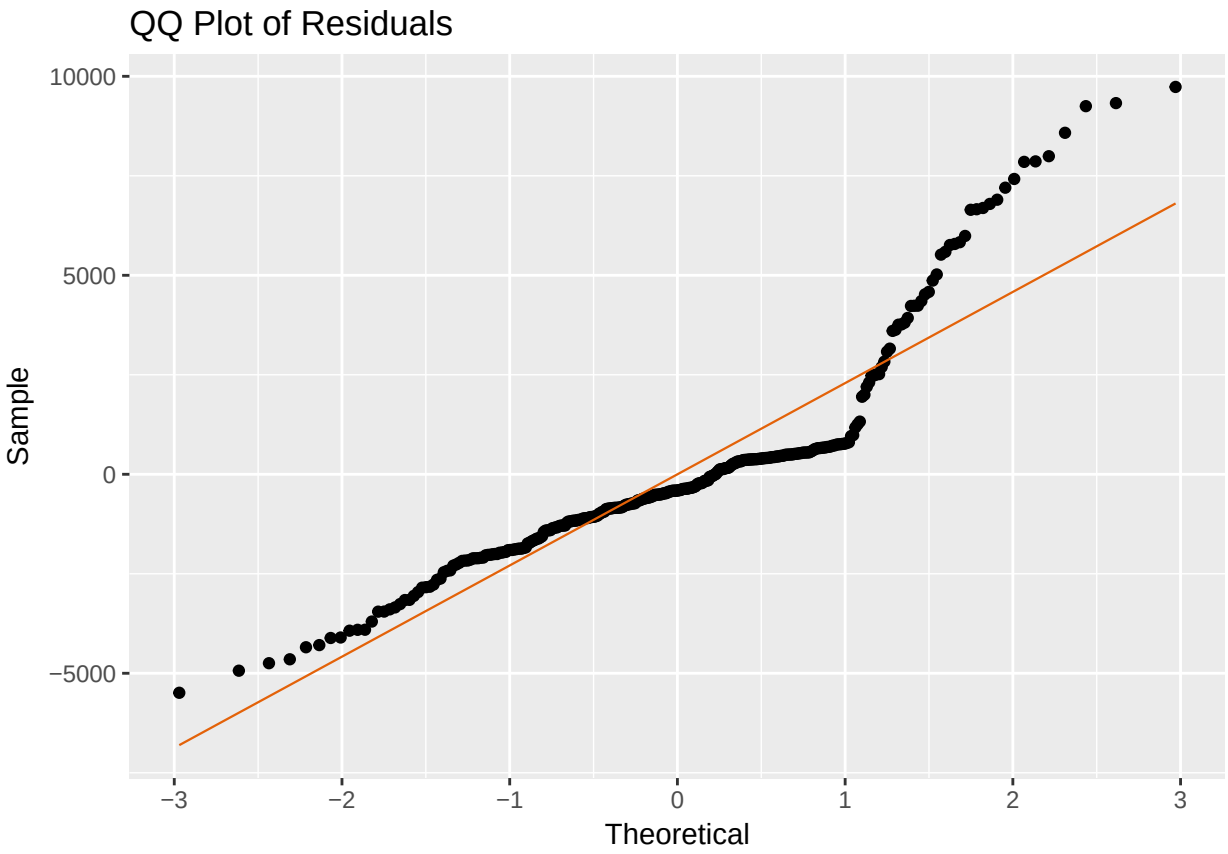
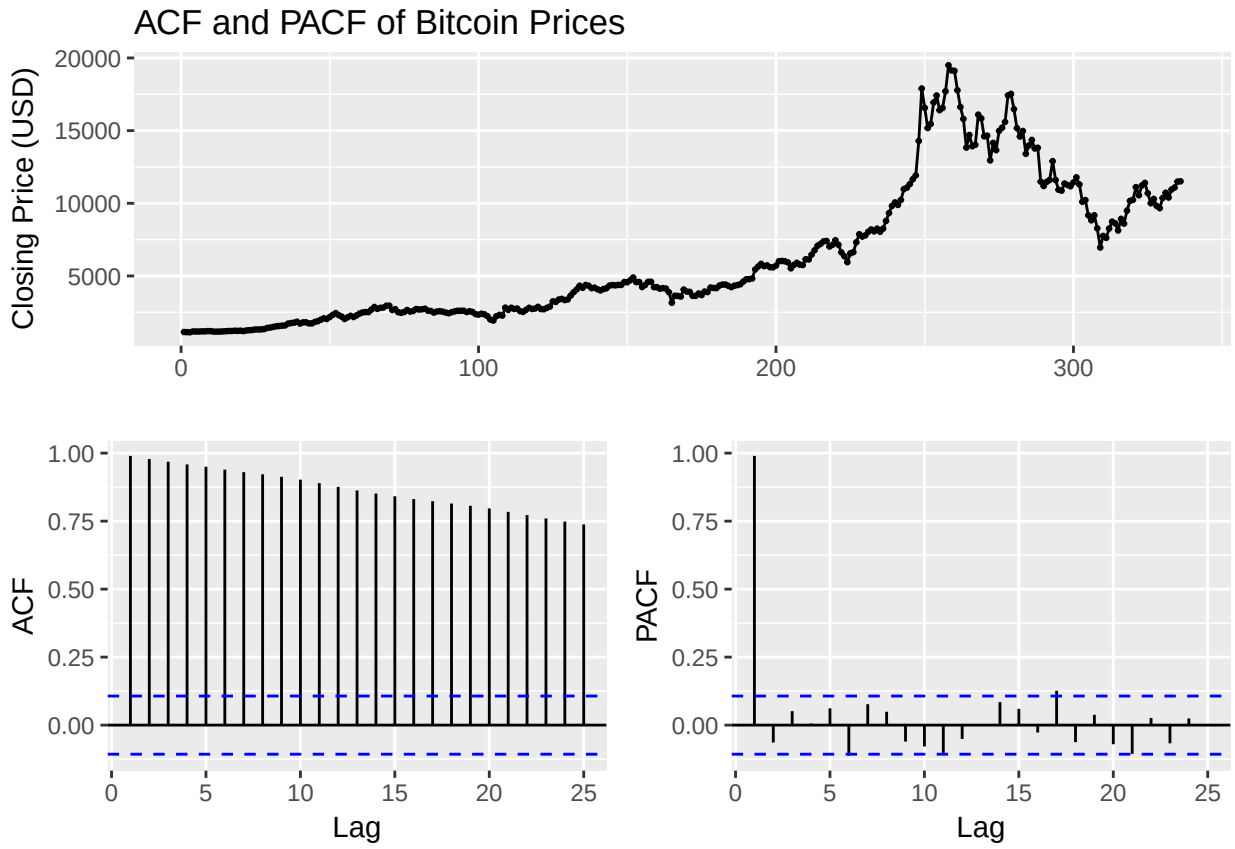


Figure 4: Residual Analysis Linear fitted Model

3 Models for Nonstationary Time Series

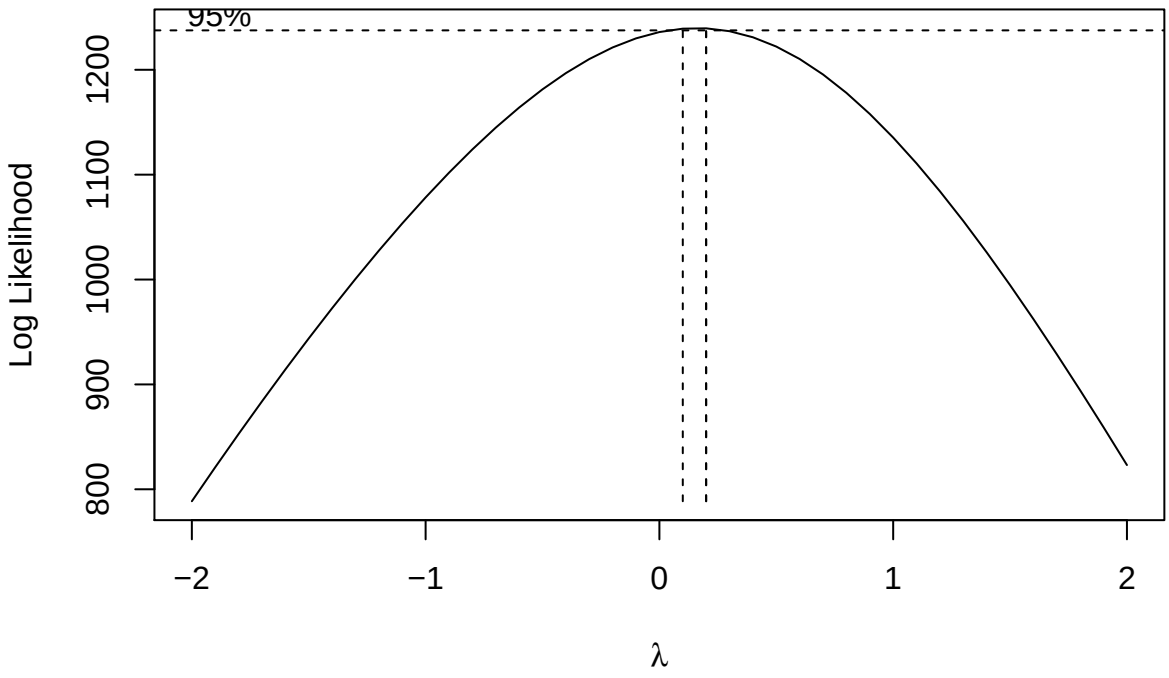
Auto regressive behaviour and non staionay Staionay is the first thing we need to check.

```
ggtsdisplay(Bitcoin.2017.zoo,
            main = 'ACF and PACF of Bitcoin Prices',
            ylab='Closing Price (USD)')
```

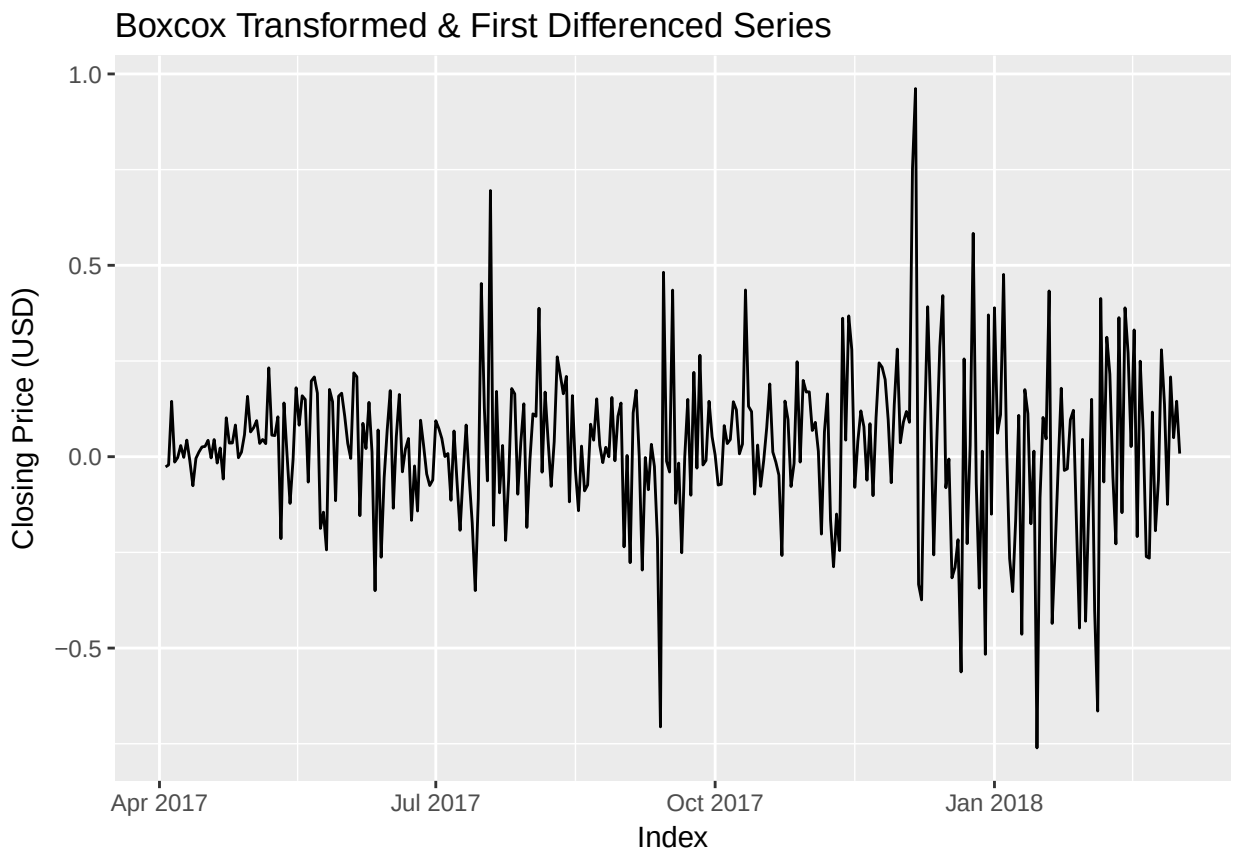


stategy to make stationay is transfromation.

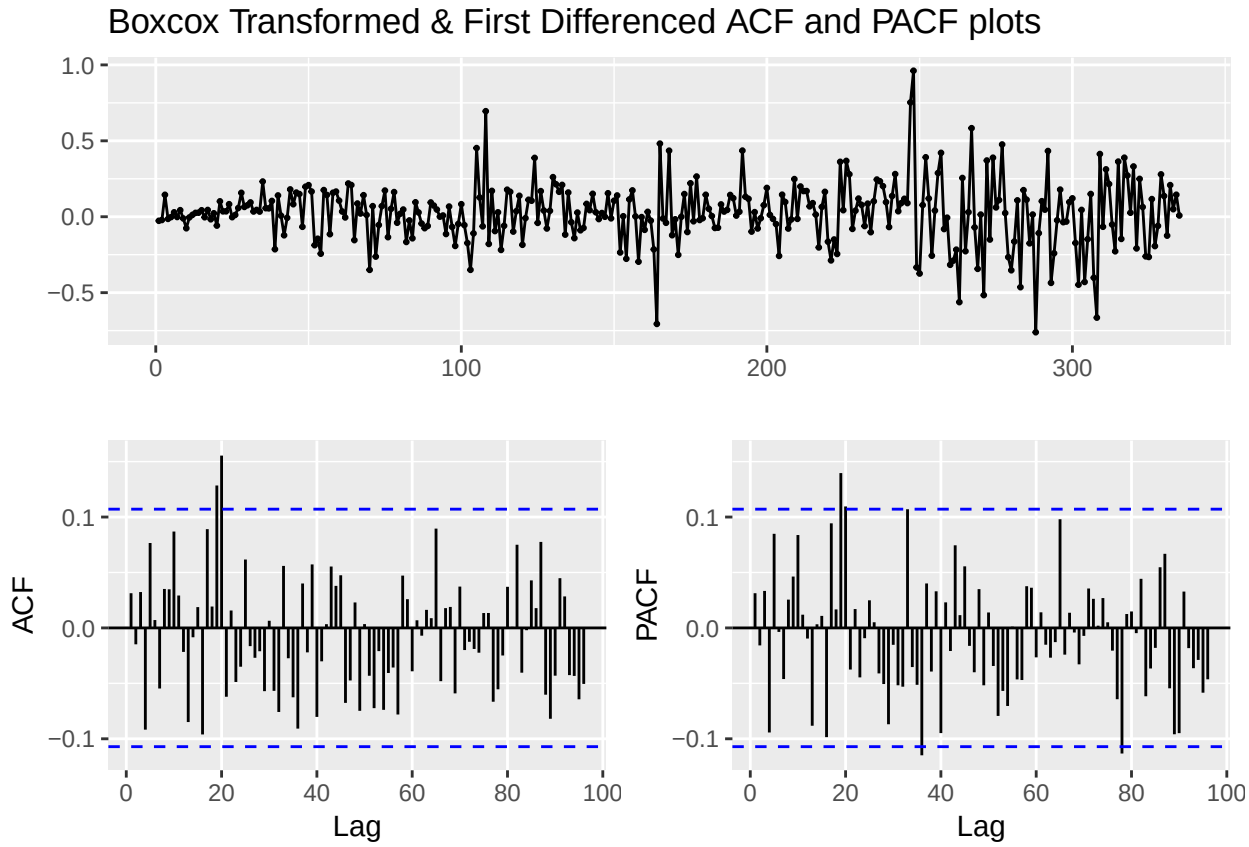
```
Bitcoin.transform = BoxCox.ar(Bitcoin.2017.zoo, method = 'yule-walker')
```




```
lambda = sum(Bitcoin.transform$ci)/length(Bitcoin.transform$ci)
Bitcoin.boxcox = (Bitcoin.2017.zoo^lambda - 1) / lambda
Bitcoin.diff = base::diff(Bitcoin.boxcox, differences = 1)
autoplot(Bitcoin.diff) +
  ylab('Closing Price (USD)') +
  ggtitle('Boxcox Transformed & First Differenced Series')
```




```
ggtsdisplay(Bitcoin.diff, lag.max = 96,ci.type='ma',
            main = 'Boxcox Transformed & First Differenced ACF and PACF plots',
            ylab='')
```




```
adf.test(Bitcoin.diff)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  Bitcoin.diff
## Dickey-Fuller = -6.968, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
eacf(Bitcoin.diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o o o o o o o o  o  o  o
## 1 x o o o o o o o o o o  o  o  o
## 2 x o o o o o o o o o o  o  o  o
## 3 x o x o o o o o o o o  o  o  o
## 4 x x x o o o o o o o o  o  o  o
## 5 o x o x o o o o o o o  o  o  o
## 6 o x o o o o o o o o o  o  o  o
## 7 x x x o x x o o o o o  o  o  o
```

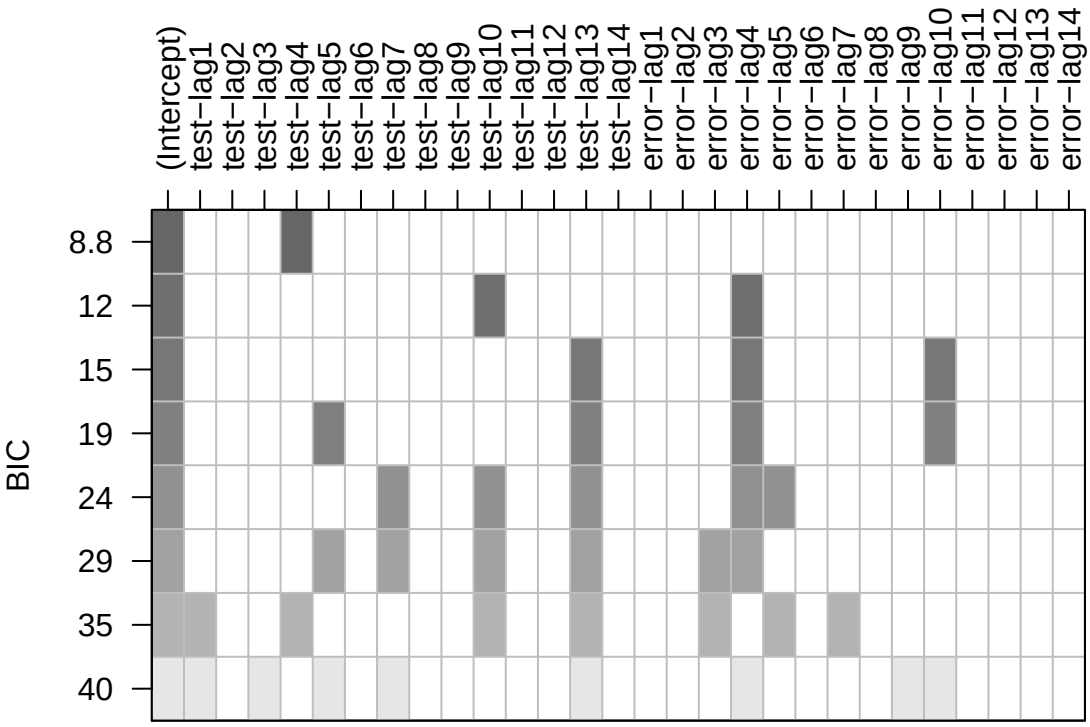
```
# ARIMA(0,1,0),ARIMA(1,1,1),ARIMA(2,1,2),ARIMA(4,1,3)
```

```
res1 = armasubsets(y=Bitcoin.diff,nar=14,nma=14,y.name='test',ar.method='mle')
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
```

```
## force.in = force.in, : 14 linear dependencies found
```

```
plot(res1)
```




```
#ARIMA(4,1,4),ARIMA(5,1,4)
```

```
#The final set of possible models is
# ARIMA(0,1,0),ARIMA(1,1,1),ARIMA(2,1,2),ARIMA(4,1,3)
# ARIMA(4,1,4),ARIMA(5,1,4)

# ARIMA(0,1,0)
model_111_css = arima(Bitcoin.boxcox, order=c(1,1,1),method='CSS')
coeftest(model_111_css)
```

```
## Warning in sqrt(diag(se)): NaNs produced
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.021830         NA      NA       NA
## ma1 0.022332         NA      NA       NA
```

```
model_111_ml = arima(Bitcoin.boxcox, order=c(1,1,1),method='ML')
coeftest(model_111_ml)
```

```
## Warning in sqrt(diag(se)): NaNs produced
```

```
##
```

```
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.020106         NA      NA       NA
## ma1 0.024617         NA      NA       NA
```

```
# ARIMA(2,1,2)
model_212_css = arima(Bitcoin.boxcox,order=c(2,1,2),method='CSS')
coeftest(model_212_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -0.028545   0.079418  -0.3594   0.7193
## ar2  0.906965   0.075874  11.9535   <2e-16 ***
## ma1  0.085822   0.084577   1.0147   0.3102
## ma2 -0.913644   0.083827 -10.8991   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_212_ml = arima(Bitcoin.boxcox,order=c(2,1,2),method='ML')
coeftest(model_212_ml)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value Pr(>|z|)
## ar1  0.0051321  0.0690326   0.0743   0.9407
## ar2  0.9315260  0.0654587  14.2307   <2e-16 ***
## ma1  0.0502674  0.0857493   0.5862   0.5577
## ma2 -0.9495800  0.0856749 -11.0835   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(2,1,2)
model_313_css = arima(Bitcoin.boxcox,order=c(3,1,3),method='CSS')
coeftest(model_313_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.577039   0.027215 -21.203 < 2.2e-16 ***
## ar2  0.733759   0.045579  16.099 < 2.2e-16 ***
## ar3  0.851447   0.024264  35.091 < 2.2e-16 ***
## ma1  0.586341   0.054769  10.706 < 2.2e-16 ***
## ma2 -0.805991   0.039380 -20.467 < 2.2e-16 ***
## ma3 -0.847992   0.036879 -22.994 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_313_ml = arima(Bitcoin.boxcox,order=c(3,1,3),method='ML')
coeftest(model_313_ml)
```

```
##
## z test of coefficients:
```

```
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.706045   0.084861 -8.3200 < 2.2e-16 ***
## ar2  0.658114   0.106881  6.1575 7.392e-10 ***
## ar3  0.893748   0.078074 11.4474 < 2.2e-16 ***
## ma1  0.751964   0.096353  7.8042 5.986e-15 ***
## ma2 -0.649078   0.131394 -4.9399 7.815e-07 ***
## ma3 -0.867606   0.093077 -9.3214 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(4,1,3)
model_413_css = arima(Bitcoin.boxcox,order=c(4,1,3),method='CSS')
coeftest(model_413_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.607213   0.098574  -6.1600 7.276e-10 ***
## ar2  0.728974   0.046521  15.6699 < 2.2e-16 ***
## ar3  0.867485   0.087776   9.8829 < 2.2e-16 ***
## ar4 -0.036714   0.061374  -0.5982    0.5497
## ma1  0.636864   0.080717   7.8901 3.019e-15 ***
## ma2 -0.780051   0.045228 -17.2473 < 2.2e-16 ***
## ma3 -0.902672   0.089098 -10.1312 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_413_ml = arima(Bitcoin.boxcox,order=c(4,1,3),method='ML')
coeftest(model_413_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.681403   0.118535 -5.7485 9.001e-09 ***
## ar2  0.673798   0.126698  5.3181 1.048e-07 ***
## ar3  0.882435   0.081890 10.7759 < 2.2e-16 ***
## ar4 -0.020609   0.061519 -0.3350    0.7376
## ma1  0.742604   0.105402  7.0454 1.849e-12 ***
## ma2 -0.653216   0.138429 -4.7188 2.372e-06 ***
## ma3 -0.869353   0.097313 -8.9336 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(4,1,4)
model_414_css = arima(Bitcoin.boxcox,order=c(4,1,4),method='CSS')
coeftest(model_414_css)
```

```
## Warning in sqrt(diag(se)): NaNs produced
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.523664         NA       NA        NA
```

```
## ar2  0.162762   0.041831   3.8910 9.985e-05 ***
## ar3  0.917653   0.074325  12.3465 < 2.2e-16 ***
## ar4  0.415562         NA       NA        NA
## ma1  0.552550         NA       NA        NA
## ma2 -0.181660   0.040086  -4.5317 5.850e-06 ***
## ma3 -0.928353   0.071005 -13.0744 < 2.2e-16 ***
## ma4 -0.532390         NA       NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_414_ml = arima(Bitcoin.boxcox,order=c(4,1,4),method='ML')
```

```
## Warning in stats:::arima(x = x, order = order, seasonal = seasonal, xreg =
## xreg, : possible convergence problem: optim gave code = 1
```

```
coeftest(model_414_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.857298   0.256941  -3.3366 0.0008482 ***
## ar2  0.075458   0.057469   1.3130 0.1891720
## ar3  1.120648   0.060162  18.6273 < 2.2e-16 ***
## ar4  0.637578   0.254442   2.5058 0.0122178 *
## ma1  0.922555   0.230314   4.0056 6.185e-05 ***
## ma2 -0.031454   0.060127  -0.5231 0.6008852
## ma3 -1.116176   0.056809 -19.6479 < 2.2e-16 ***
## ma4 -0.717939   0.228129  -3.1471 0.0016492 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ARIMA(5,1,4)
model_514_css = arima(Bitcoin.boxcox,order=c(5,1,4),method='CSS')
coeftest(model_514_css)
```

```
## Warning in sqrt(diag(se)): NaNs produced
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.0994981  0.0424432   2.3443 0.0190647 *
## ar2  0.1381344  0.0384323   3.5942 0.0003254 ***
## ar3  0.5069664  0.0030855 164.3061 < 2.2e-16 ***
## ar4  0.1650015  0.0261166   6.3179 2.652e-10 ***
## ar5  0.0902766  0.0577983   1.5619 0.1183062
## ma1 -0.0794650         NA       NA        NA
## ma2 -0.1856691  0.0430209  -4.3158 1.590e-05 ***
## ma3 -0.5132720  0.0420131 -12.2169 < 2.2e-16 ***
## ma4 -0.3071109         NA       NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_514_ml = arima(Bitcoin.boxcox,order=c(5,1,4),method='ML')
```

```
## Warning in stats:::arima(x = x, order = order, seasonal = seasonal, xreg =
```

```
## xreg, : possible convergence problem: optim gave code = 1
```

```
coeftest(model_514_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.216615   0.582131  0.3721   0.7098
## ar2  0.125234   0.255391  0.4904   0.6239
## ar3  0.472675   0.520008  0.9090   0.3634
## ar4 -0.054061   0.425997 -0.1269   0.8990
## ar5  0.109269   0.078409  1.3936   0.1634
## ma1 -0.166104   0.585688 -0.2836   0.7767
## ma2 -0.144673   0.251820 -0.5745   0.5656
## ma3 -0.438177   0.523688 -0.8367   0.4028
## ma4 -0.056627   0.413964 -0.1368   0.8912
```

```
sort.score(stats::AIC(model_111_ml,model_212_ml,model_313_ml,model_413_ml,model_414_ml,model_514_ml), s
```

```
##              df        AIC
## model_212_ml  5 -102.63500
## model_313_ml  7 -102.11536
## model_414_ml  9 -101.76633
## model_111_ml  3 -100.97940
## model_413_ml  8 -100.23271
## model_514_ml 10  -95.11157
```

```
sort.score(stats::BIC(model_111_ml,model_212_ml,model_313_ml,model_413_ml,model_414_ml,model_514_ml), s
```

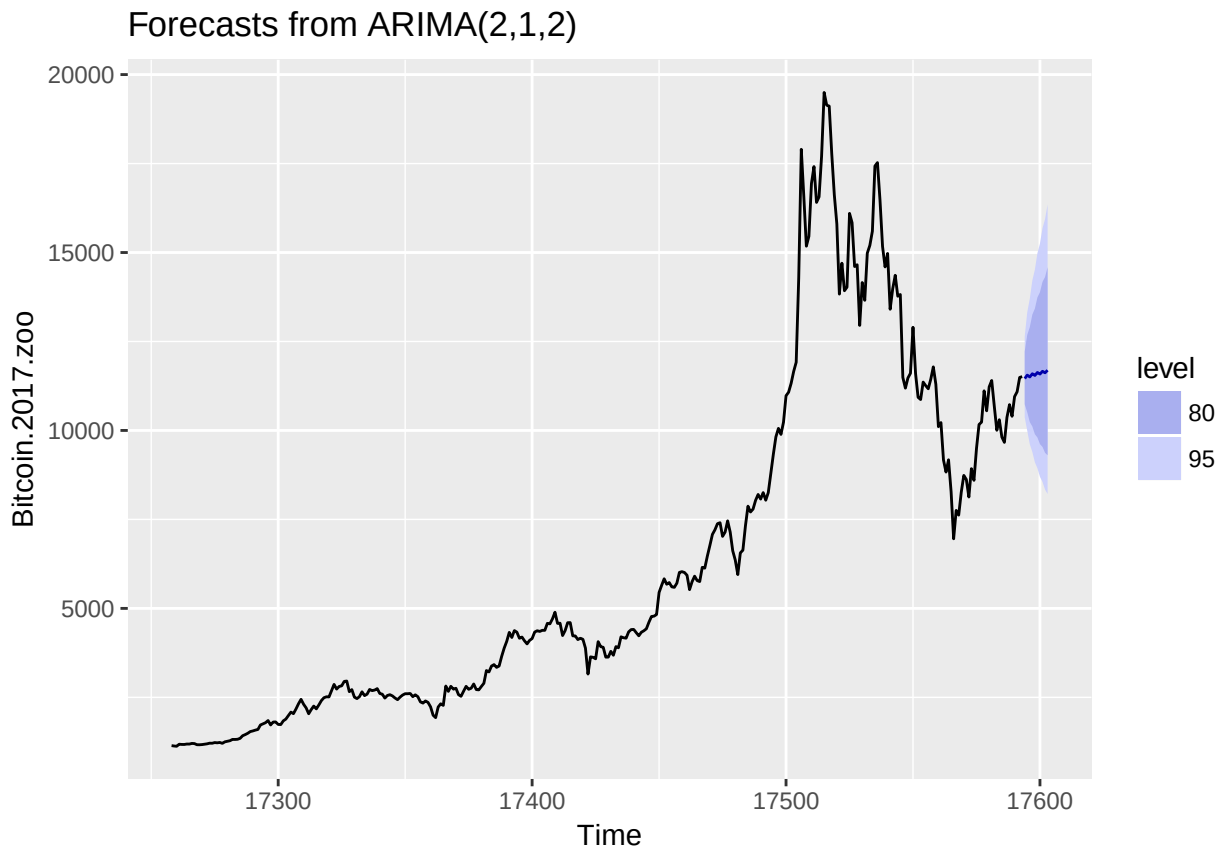
```
##              df       BIC
## model_111_ml  3 -89.53701
## model_212_ml  5 -83.56435
## model_313_ml  7 -75.41645
## model_413_ml  8 -69.71967
## model_414_ml  9 -67.43915
## model_514_ml 10 -56.97026
```

```
fit <- Arima(Bitcoin.2017.zoo, order=c(2,1,2), lambda = lambda)
summary(fit)
```

```
## Series: Bitcoin.2017.zoo
## ARIMA(2,1,2)
## Box Cox transformation: lambda= 0.15
##
## Coefficients:
##           ar1     ar2     ma1      ma2
##       -0.0045  0.9231  0.0615  -0.9385
## s.e.   0.0652  0.0620  0.0765   0.0764
##
## sigma^2 estimated as 0.04208:  log likelihood=56.33
## AIC=-102.66   AICc=-102.47   BIC=-83.59
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 16.49856 525.3934 292.1274 0.3300122 3.945909 0.9965809
```

```
##                      ACF1
## Training set 0.05364657
```

```
autoplot(forecast(fit,h=10))
```




```
Bitcoin.forecast <- read_csv("../data/Bitcoin_Prices_Forecasts.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_character(),
##   `Closing price` = col_double()
## )
```

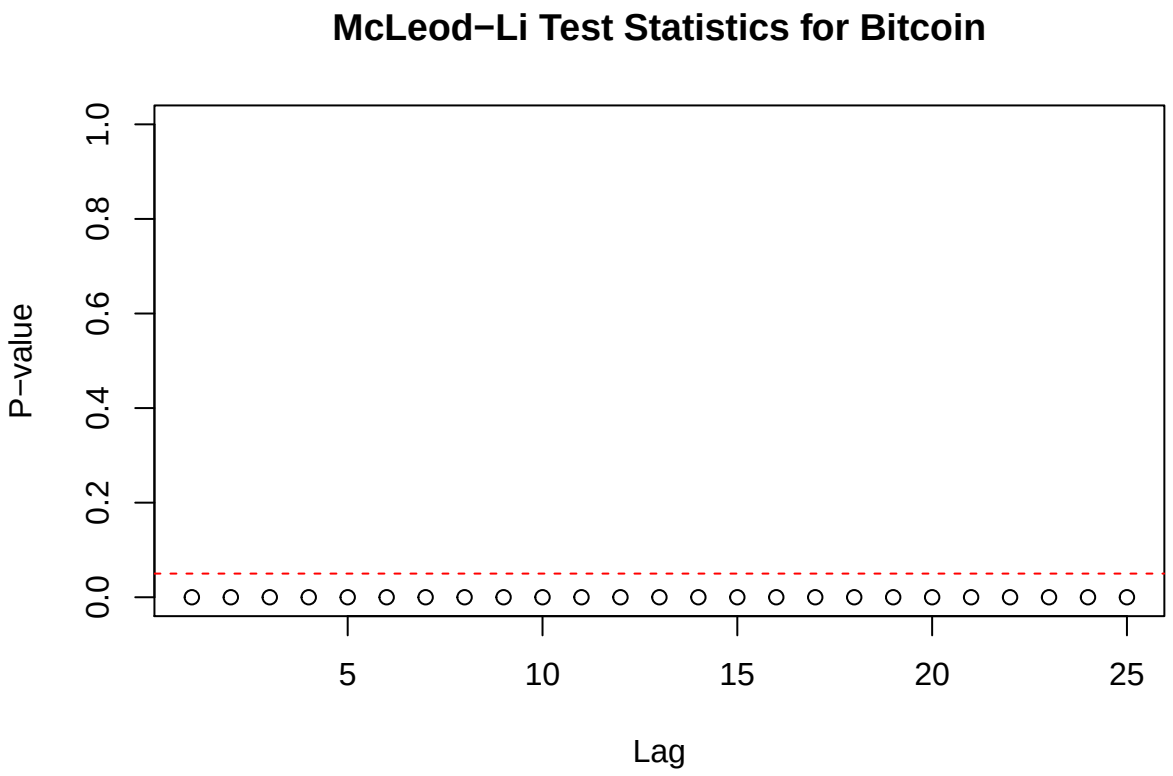
```
Bitcoin.forecast$Date = as.Date(Bitcoin.forecast$Date,'%d/%m/%y')
```

```
source('MASE.r')
MASE(Bitcoin.forecast$Closing.price, as.vector(tail(fitted(forecast(fit,h=10)),10)))
```

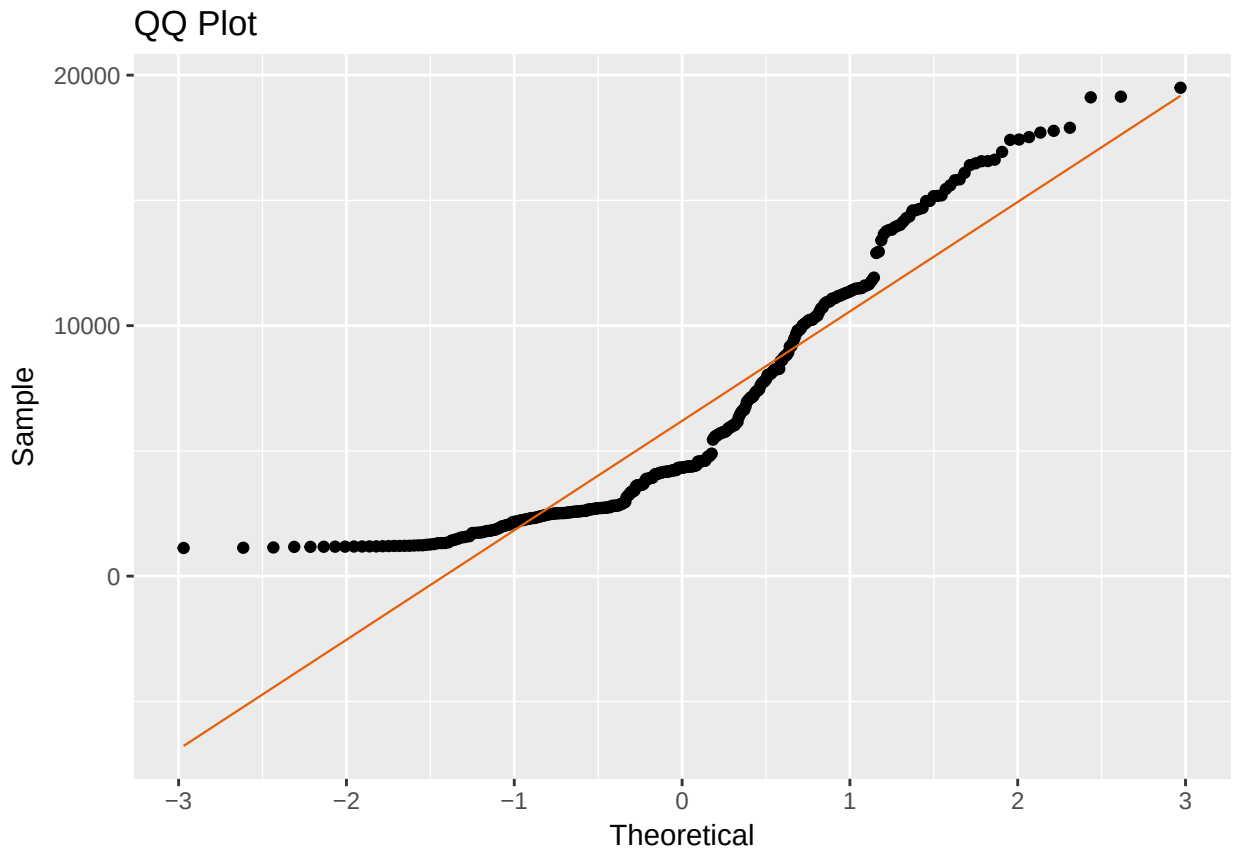
```
## Warning: Unknown or uninitialised column: 'Closing.price'.
```

```
## $MASE
##   MASE
## 1  NaN
```

```
McLeod.Li.test(y=Bitcoin.2017.zoo,main="McLeod-Li Test Statistics for Bitcoin")
```




```
residual_analysis_qq(Bitcoin.2017.zoo, 'QQ Plot')
```


QQ Plot
20000
10000
0
Sample
−3
−2
−1
0
1
2
3
Theoretical