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 Initial Diagnosis

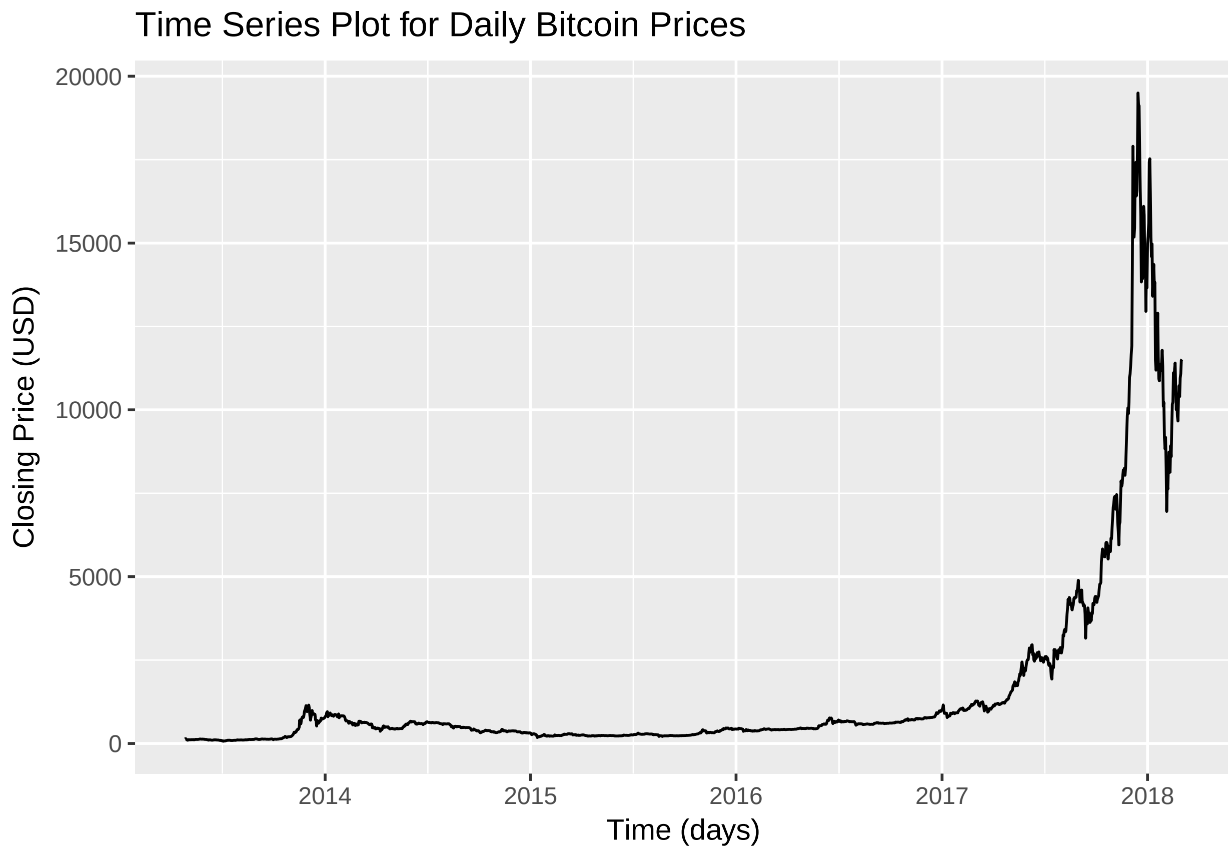
```r
# Import Libraries
library(TSA)
library(fUnitRoots)
library(forecast)
library(CombMSC)
library(lmtest)
library(fGarch)
library(rugarch)
library(zoo)
library(ggplot2)
require(readr)
require(FitAR)

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

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

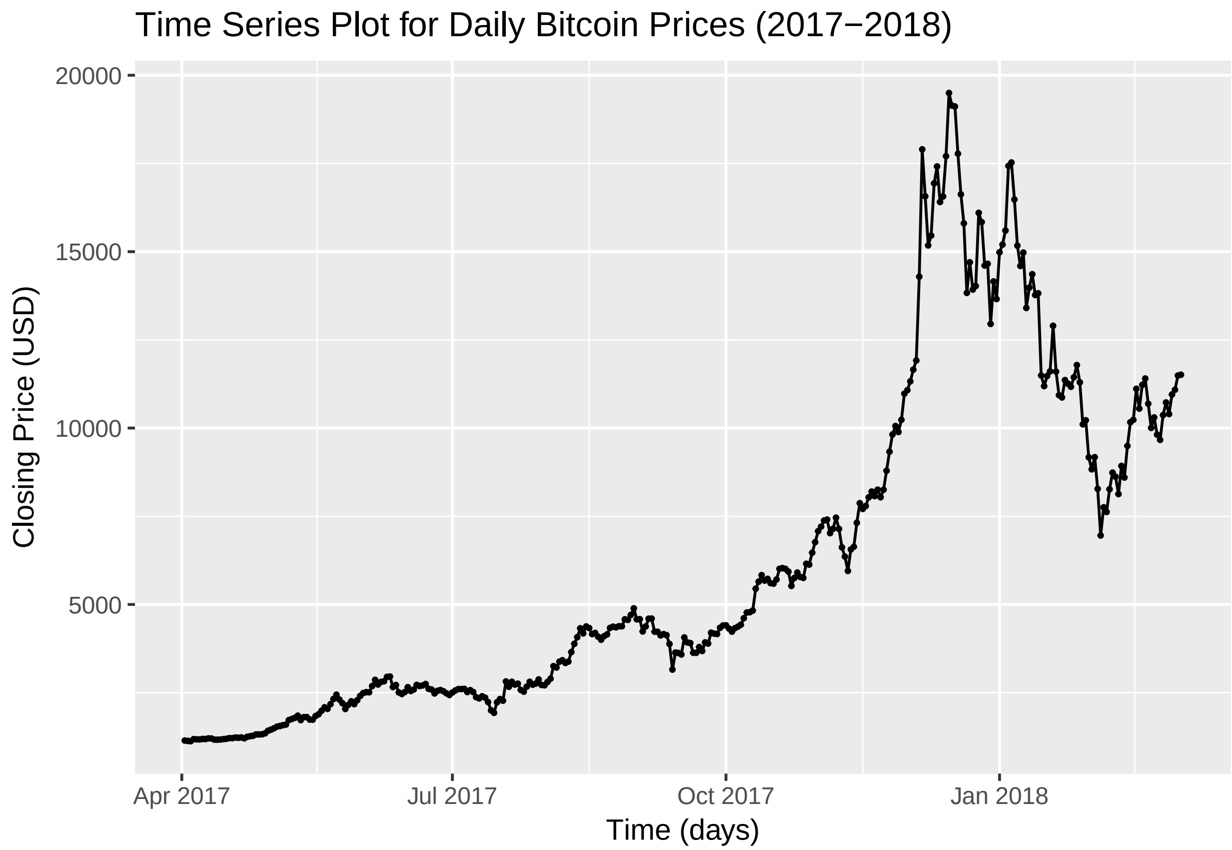
```
## [1] "zoo"
```

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

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

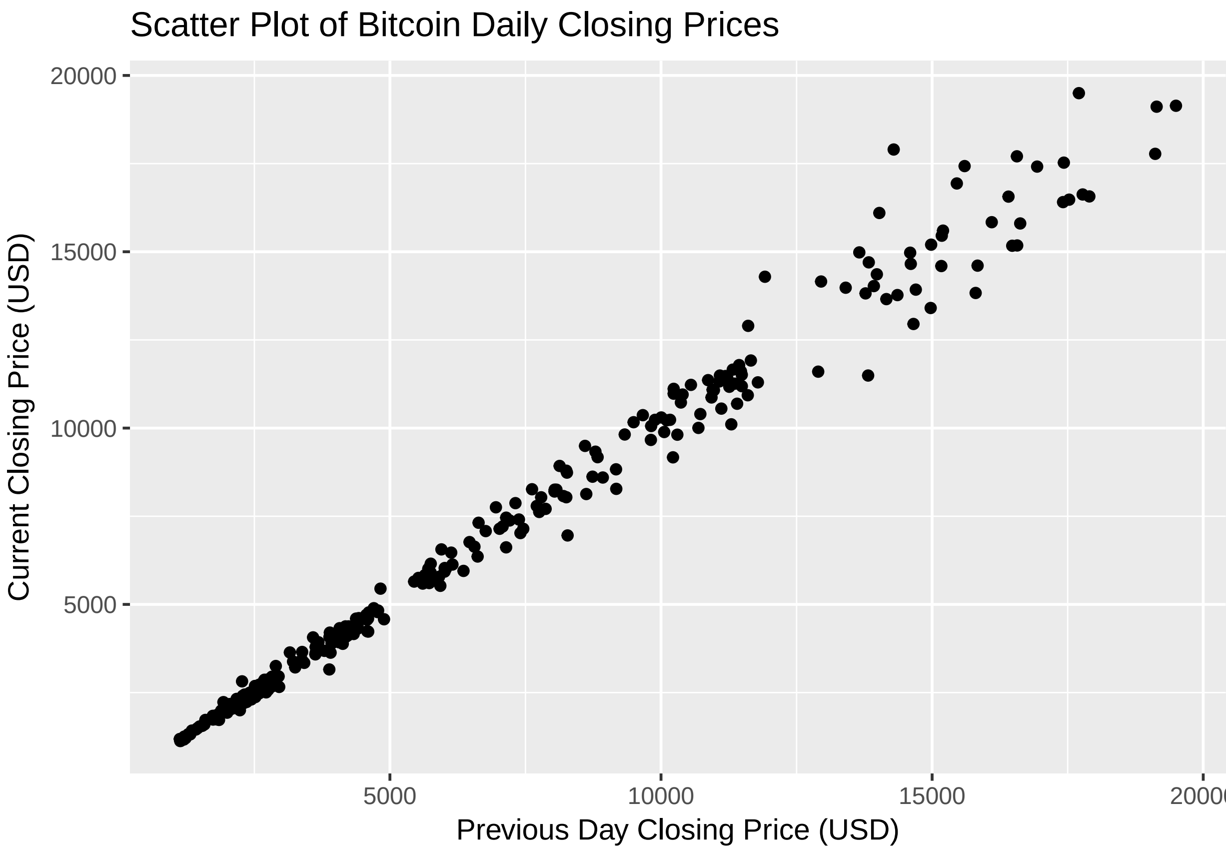



```
Bitcoin.2017 = Bitcoin[Bitcoin$Date > as.Date("2017-04-01"),]
Bitcoin.2017.zoo = zoo(Bitcoin.2017$Close, Bitcoin.2017$Date)
autoplot(Bitcoin.2017.zoo) +
  geom_point(size=.5) +
  ylab('Closing Price (USD)') +
  xlab('Time (days)') +
  ggtitle("Time Series Plot for Daily Bitcoin Prices (2017-2018)")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



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")
```




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

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

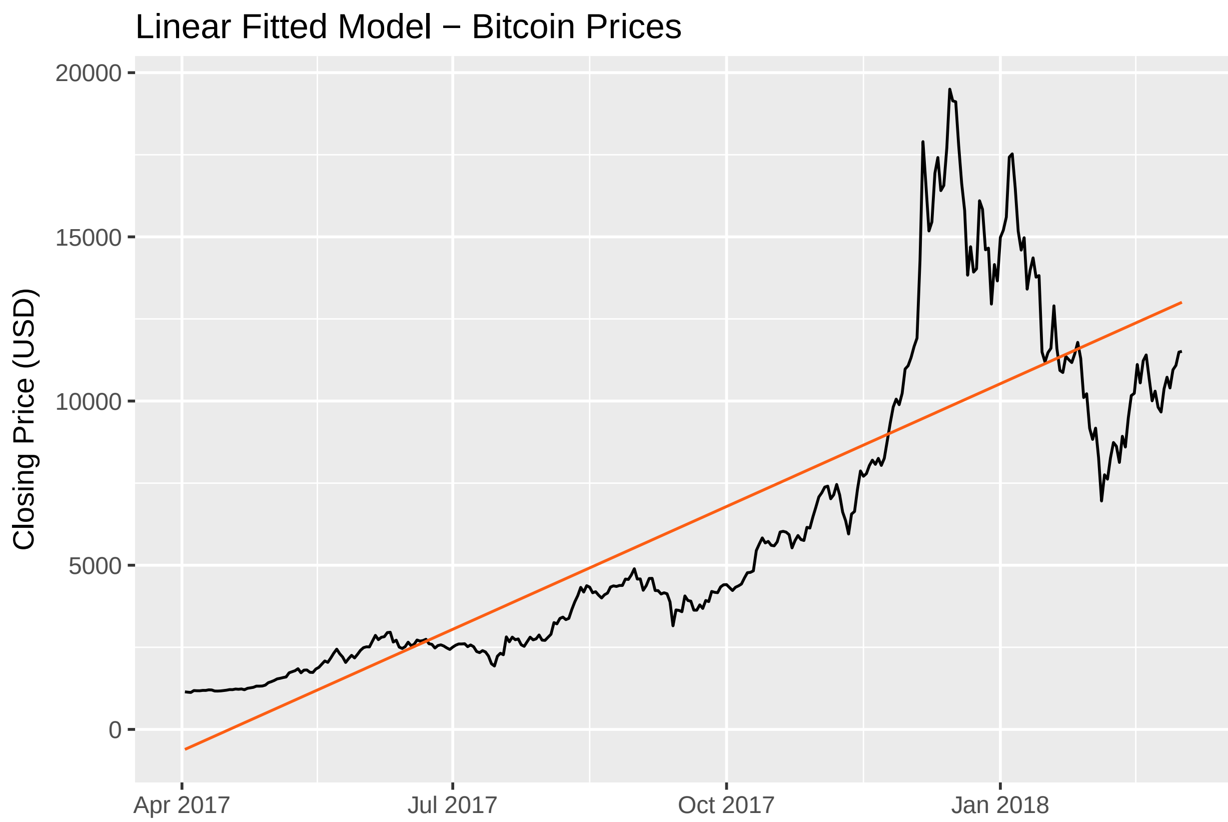
2.2 Linear Model

```r
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
```

```r
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

```r
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)
}

checkresiduals(model.ln)
```

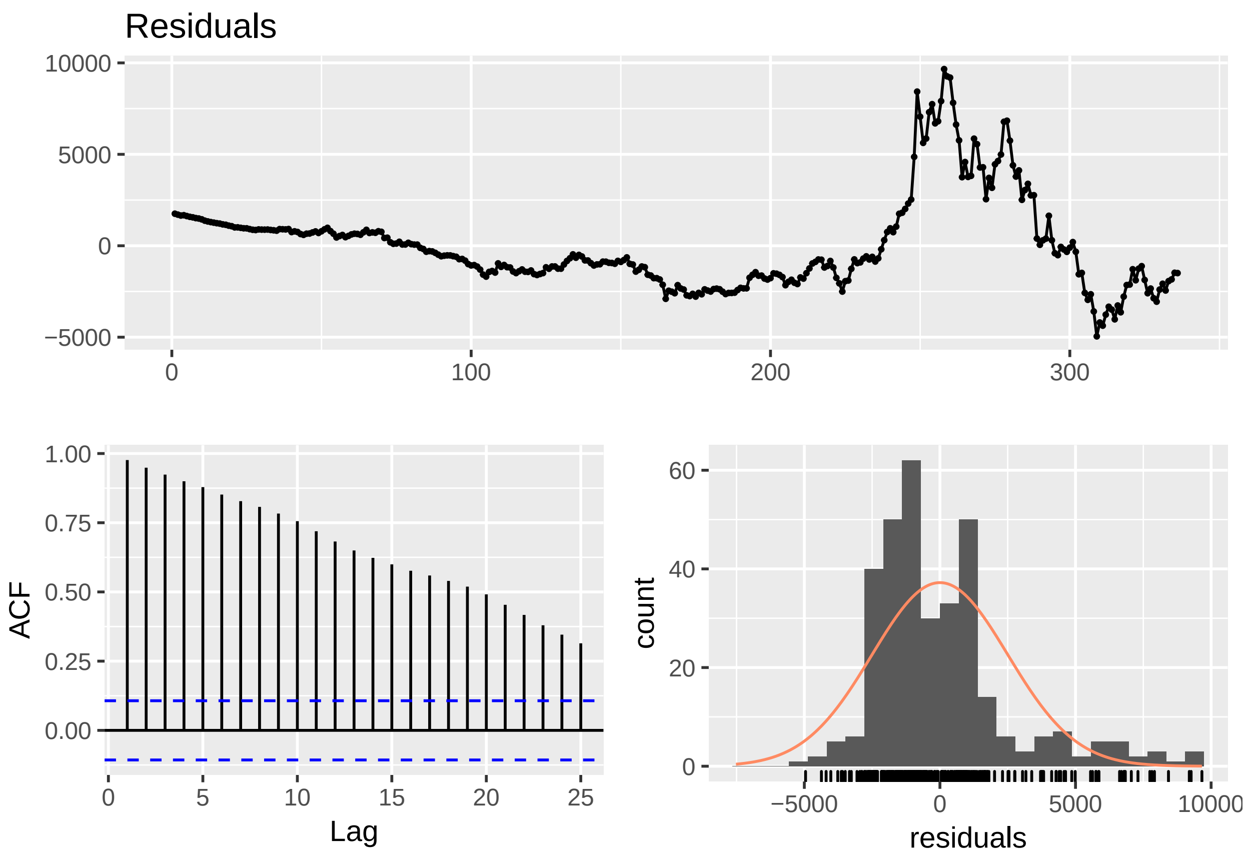


Figure 1: Residual Analysis Linear fitted Model

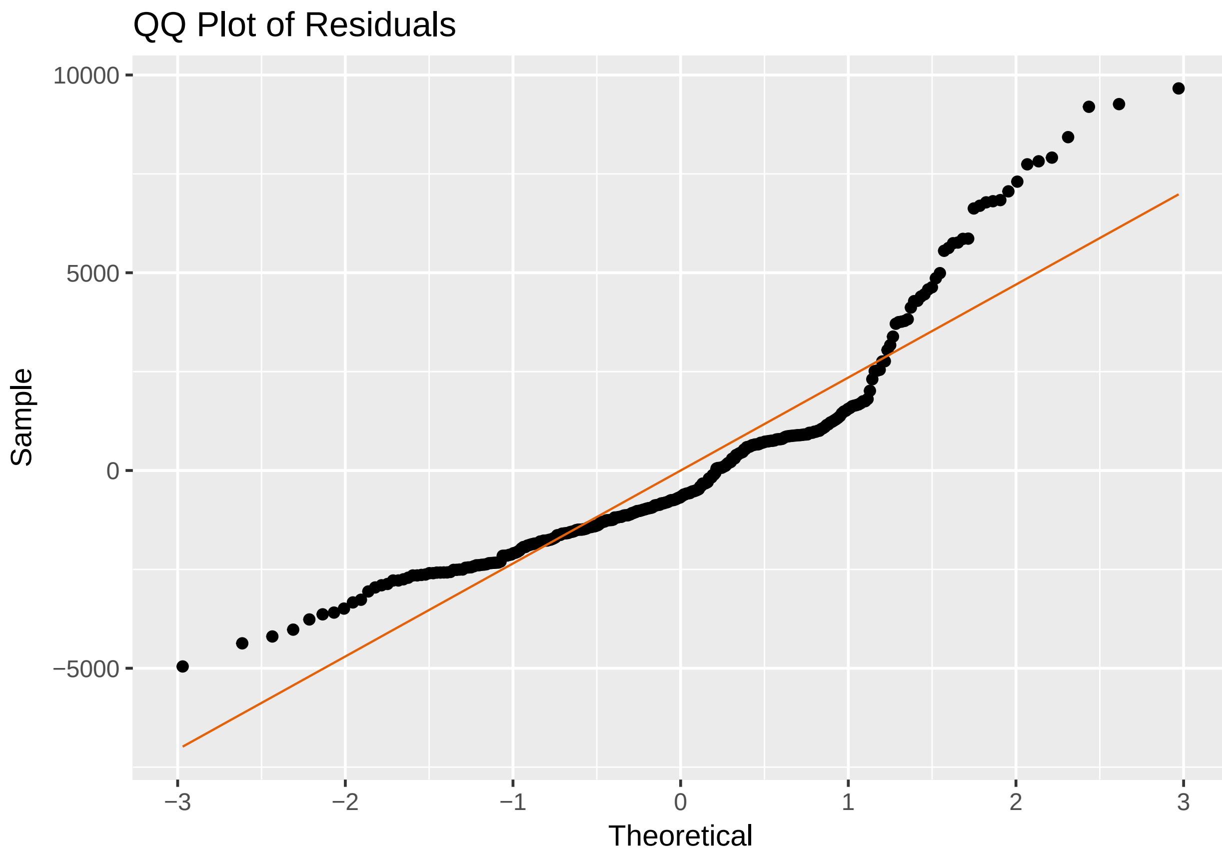


Figure 2: Residual Analysis Linear fitted Model

```
##
##  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)))
```

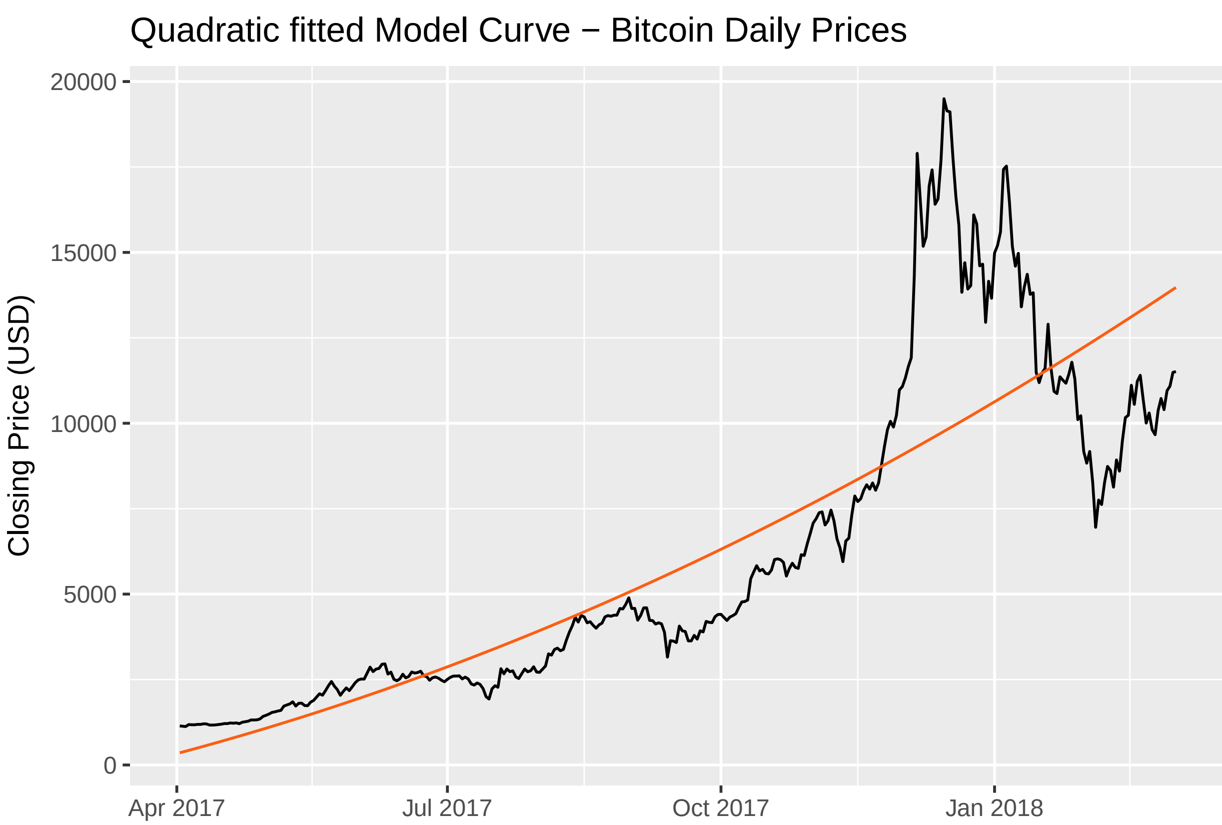
```
##
##  Shapiro-Wilk normality test
##
## data:  as.vector(residuals(model.ln))
## W = 0.87841, p-value = 1.204e-15
```

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
```

```r
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')
```



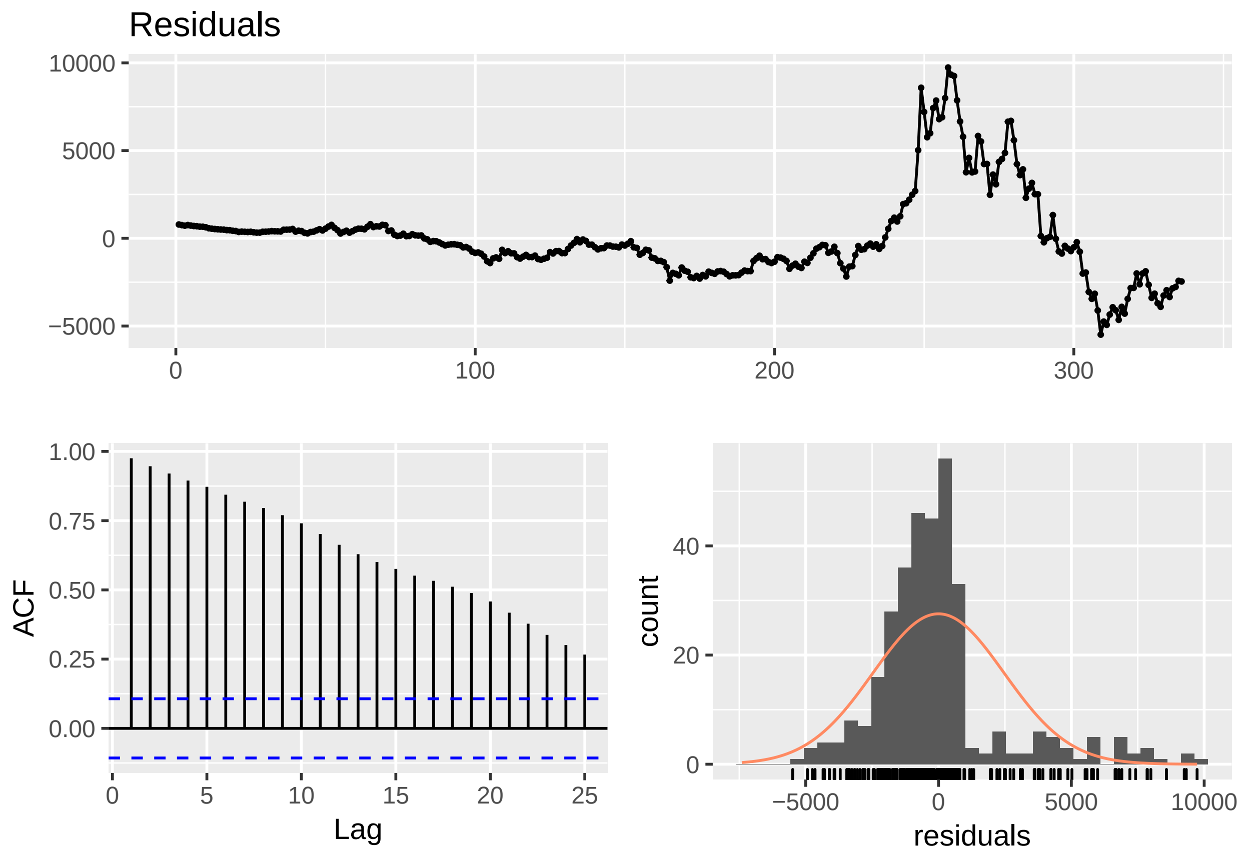


Figure 3: Residual Analysis Quadratic fitted Model

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
```

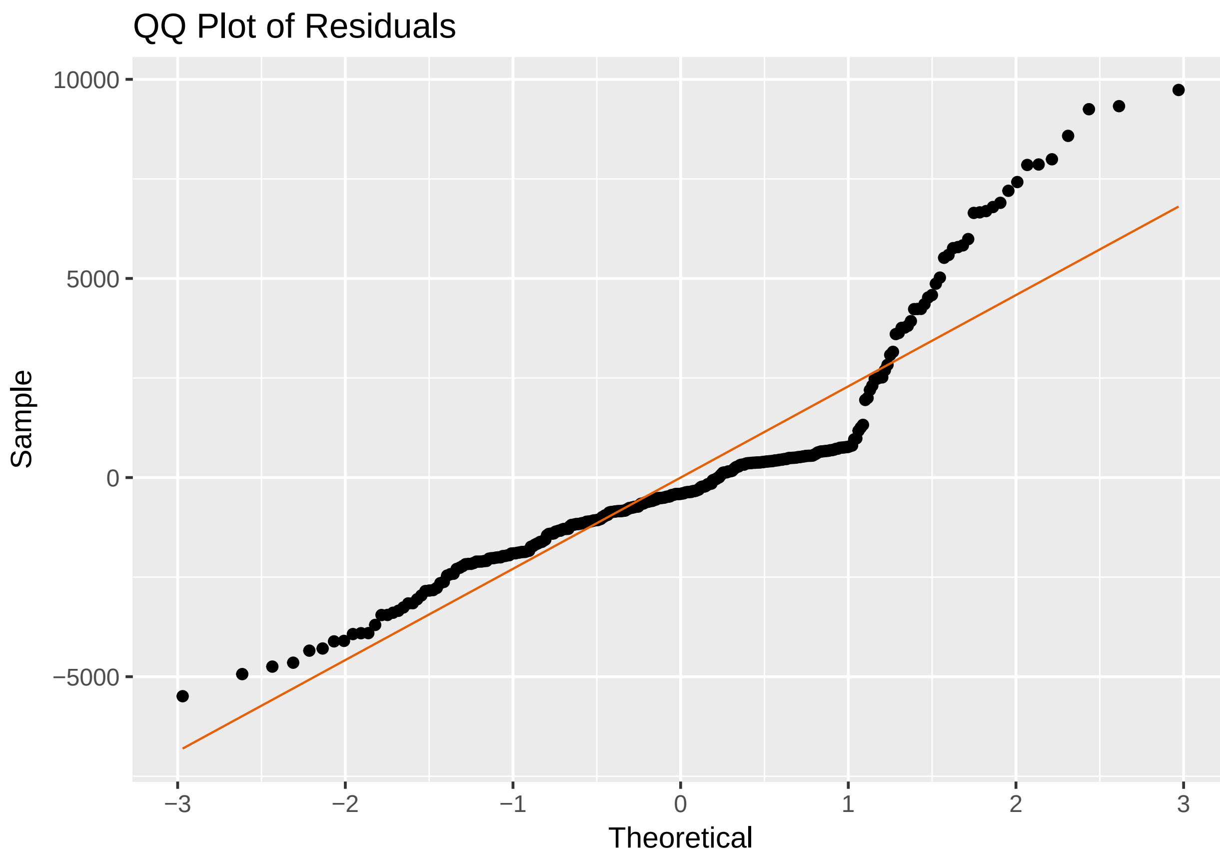
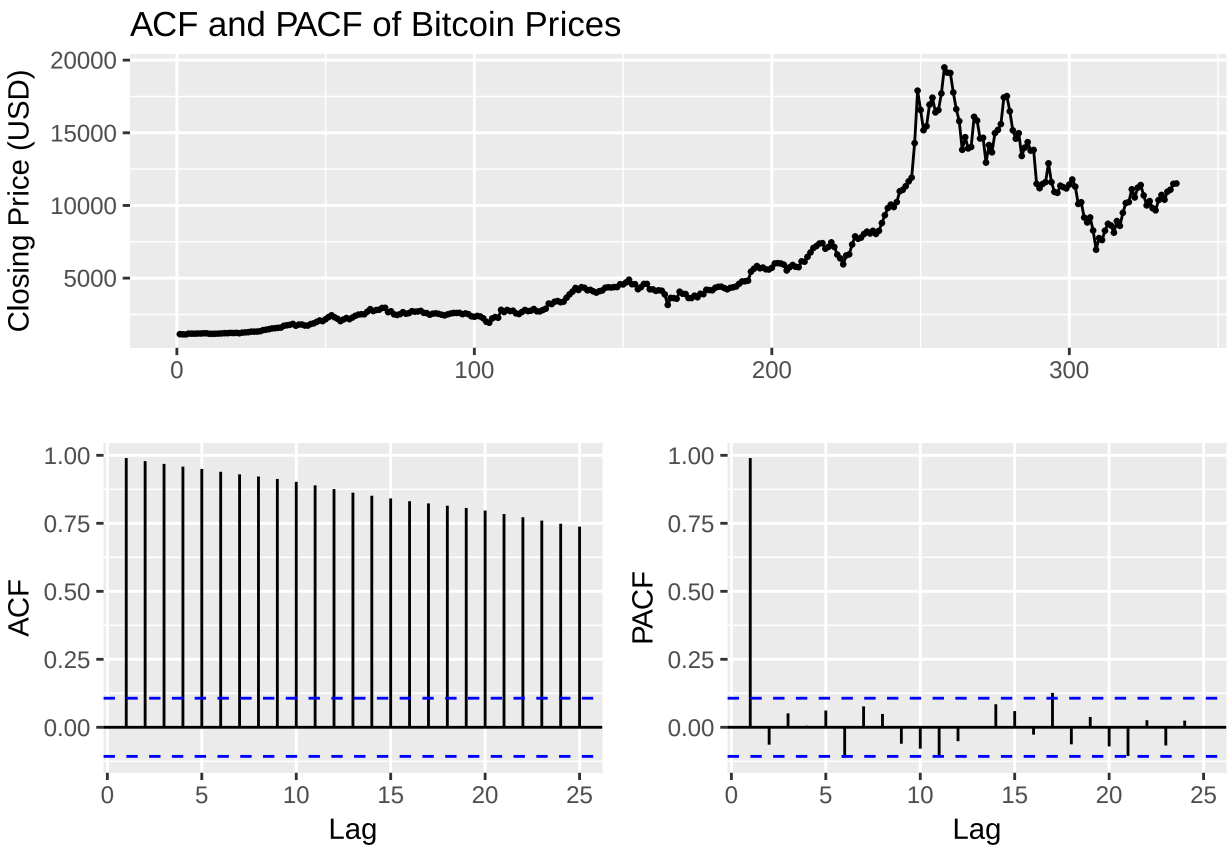


Figure 4: Residual Analysis Linear fitted Model

3 Models for Nonstationary Time Series

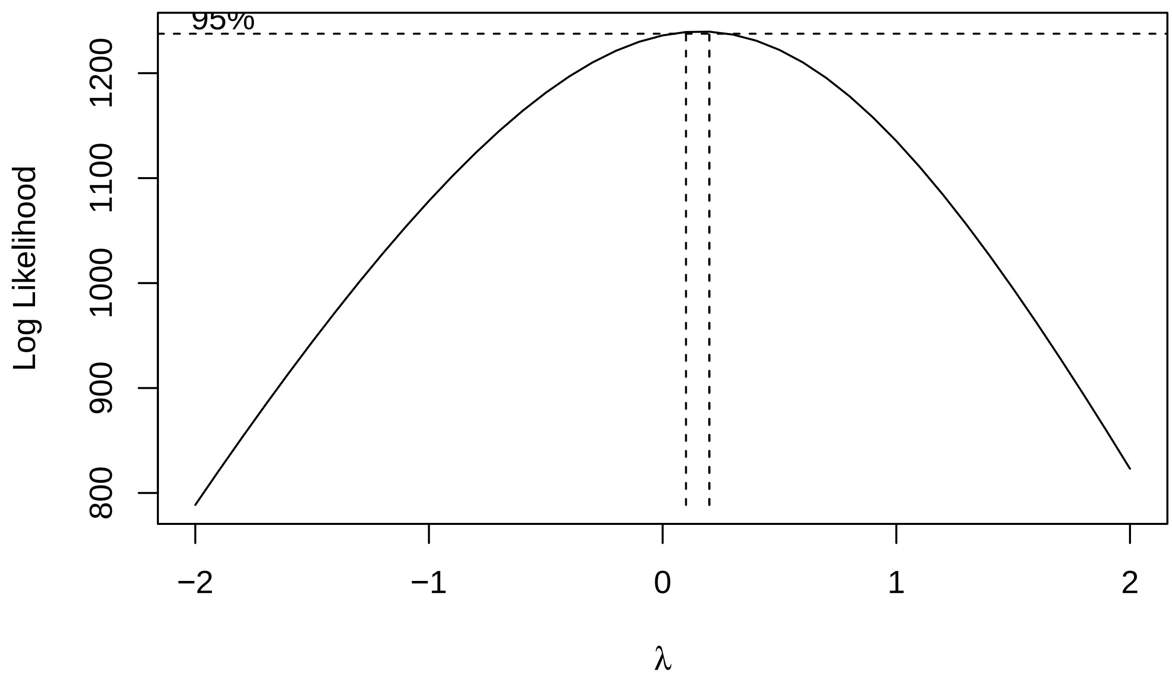
Auto regressive behaviour and non staionay Staionay is the first thing we need to check.

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

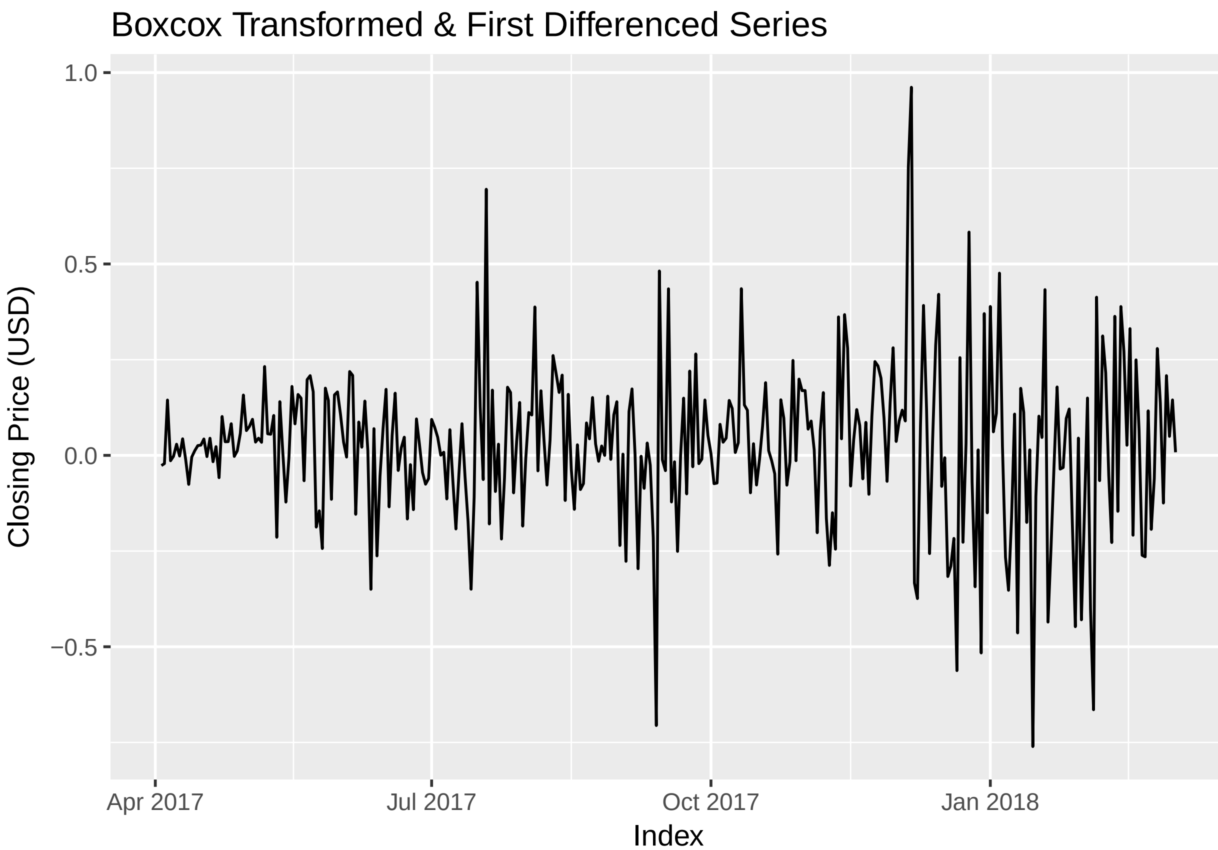


stategy to make stationay is transfromation.

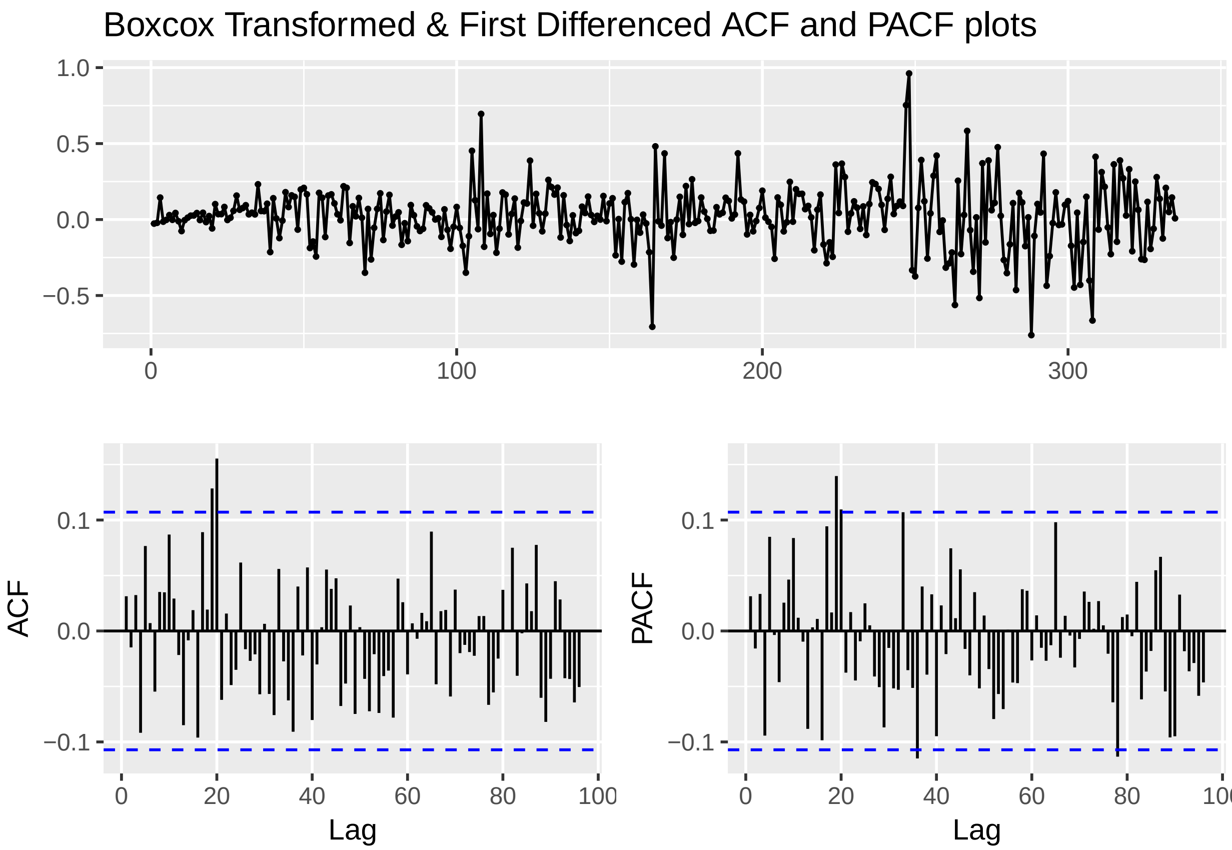
```
Bitcoin.transform = BoxCox.ar(Bitcoin.2017.zoo, method = 'yule-walker')
```




```r
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)
```

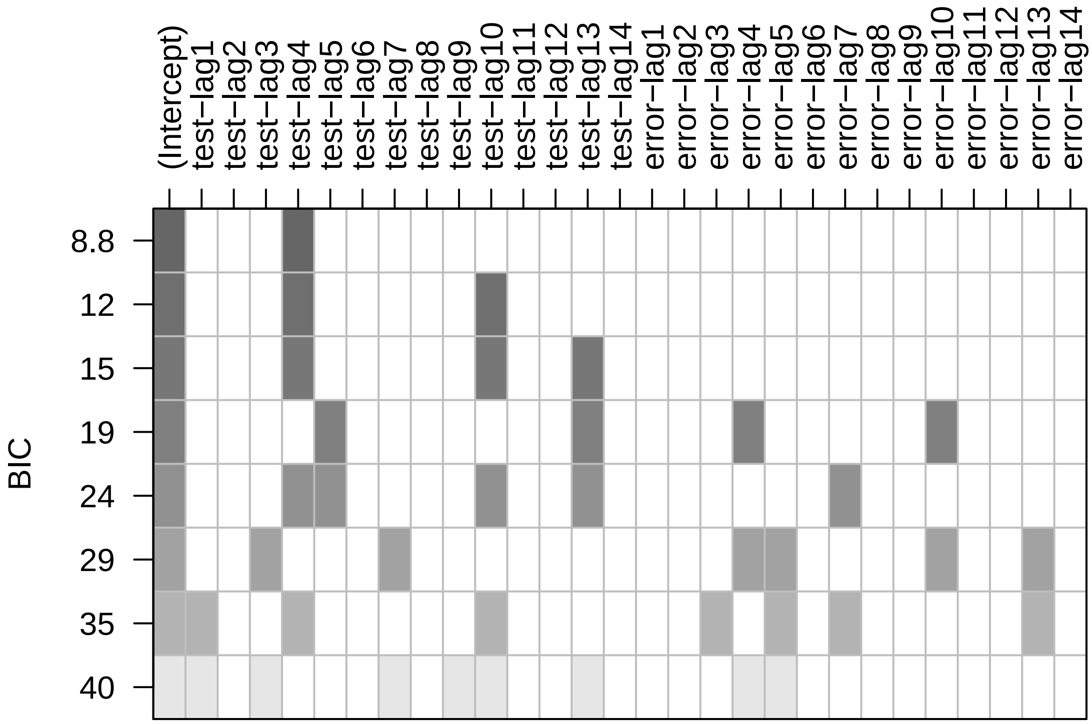
```
##
##  Augmented Dickey-Fuller Test
##
## data:  Bitcoin.diff
## Dickey-Fuller = -6.968, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
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')
plot(res1)
```




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

```r
#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)
```

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

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

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

```r
# 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 -0.028530   0.079410  -0.3593   0.7194
## ar2  0.906936   0.075863  11.9549   <2e-16 ***
## ma1  0.085848   0.084591   1.0149   0.3102
## ma2 -0.913597   0.083838 -10.8972   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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 -0.0021222  0.0653257  -0.0325   0.9741
## ar2  0.9248274  0.0619597  14.9263   <2e-16 ***
## ma1  0.0590080  0.0774408   0.7620   0.4461
## ma2 -0.9409861  0.0773613 -12.1635   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 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 -0.571168   0.021168 -26.983 < 2.2e-16 ***
## ar2  0.736466   0.038762  19.000 < 2.2e-16 ***
## ar3  0.842896   0.020398  41.324 < 2.2e-16 ***
## ma1  0.578964   0.046389  12.481 < 2.2e-16 ***
## ma2 -0.812537   0.031263 -25.991 < 2.2e-16 ***
## ma3 -0.837218   0.036048 -23.225 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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 -0.704438   0.087921 -8.0122 1.127e-15 ***
## ar2  0.657008   0.107606  6.1057 1.024e-09 ***
## ar3  0.890417   0.080571 11.0514 < 2.2e-16 ***
## ma1  0.750869   0.098881  7.5937 3.109e-14 ***
## ma2 -0.648222   0.131848 -4.9164 8.813e-07 ***
## ma3 -0.864195   0.095694 -9.0308 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 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 -0.606687   0.103204  -5.8786 4.139e-09 ***
## ar2  0.724842   0.045366  15.9775 < 2.2e-16 ***
## ar3  0.868001   0.088273   9.8332 < 2.2e-16 ***
## ar4 -0.034735   0.061179  -0.5678    0.5702
## ma1  0.633092   0.085276   7.4240 1.136e-13 ***
## ma2 -0.780848   0.042049 -18.5698 < 2.2e-16 ***
## ma3 -0.899628   0.092503  -9.7254 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
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 -0.681608   0.122969 -5.5429 2.975e-08 ***
## ar2  0.673771   0.133685  5.0400 4.656e-07 ***
## ar3  0.882439   0.082700 10.6704 < 2.2e-16 ***
## ar4 -0.020472   0.061829 -0.3311    0.7406
## ma1  0.743163   0.110054  6.7527 1.451e-11 ***
## ma2 -0.652845   0.146801 -4.4471 8.702e-06 ***
## ma3 -0.869292   0.099522 -8.7346 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.551983         NA       NA        NA
## ar2  0.172921   0.038259   4.5198 6.190e-06 ***
## ar3  0.915619   0.060298  15.1850 < 2.2e-16 ***
## ar4  0.435558         NA       NA        NA
## ma1  0.580501         NA       NA        NA
## ma2 -0.196556   0.043484  -4.5202 6.178e-06 ***
## ma3 -0.920758   0.052864 -17.4176 < 2.2e-16 ***
## ma4 -0.551651         NA       NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.858500   0.281071 -3.0544 0.0022552 **
## ar2  0.075681   0.128007  0.5912 0.5543661
## ar3  1.122140   0.098181 11.4293 < 2.2e-16 ***
## ar4  0.639474   0.253747  2.5201 0.0117314 *
## ma1  0.923109   0.264539  3.4895 0.0004839 ***
## ma2 -0.032671   0.177524 -0.1840 0.8539858
## ma3 -1.117752   0.155372 -7.1940  6.29e-13 ***
## ma4 -0.719323   0.226069 -3.1819 0.0014633 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## z test of coefficients:
##
##        Estimate  Std. Error  z value  Pr(>|z|)
## ar1  9.9074e-02  1.9514e-04  507.703 < 2.2e-16 ***
## ar2  1.3895e-01  3.0111e-04  461.475 < 2.2e-16 ***
## ar3  5.0499e-01  7.2128e-05 7001.257 < 2.2e-16 ***
## ar4  1.6450e-01  1.5559e-04 1057.251 < 2.2e-16 ***
## ar5  9.0947e-02  3.6146e-04  251.613 < 2.2e-16 ***
## ma1 -8.1368e-02  5.7083e-03  -14.254 < 2.2e-16 ***
## ma2 -1.9260e-01  1.6938e-02  -11.371 < 2.2e-16 ***
## ma3 -5.1953e-01  1.7609e-02  -29.504 < 2.2e-16 ***
## ma4 -3.1371e-01  6.3856e-03  -49.127 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.216613   0.581949  0.3722   0.7097
## ar2  0.125243   0.255393  0.4904   0.6239
## ar3  0.472679   0.519846  0.9093   0.3632
## ar4 -0.054074   0.426025 -0.1269   0.8990
## ar5  0.109269   0.078398  1.3938   0.1634
## ma1 -0.166104   0.585508 -0.2837   0.7766
## ma2 -0.144679   0.251819 -0.5745   0.5656
## ma3 -0.438182   0.523533 -0.8370   0.4026
## ma4 -0.056614   0.413995 -0.1368   0.8912
```

```r
source('sort.score.r')
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 -102.65580
## model_313_ml  7 -102.11673
## model_414_ml  9 -101.78404
## model_111_ml  3 -100.97940
## model_413_ml  8 -100.23246
## 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 -83.58515
## model_313_ml  7 -75.41781
## model_413_ml  8 -69.71941
## model_414_ml  9 -67.45687
## 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
##       -0.0041  0.9230  0.0612  -0.9387
## s.e.   0.0657  0.0623  0.0771   0.0770
##
## 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 16.4692 525.3894 292.1485 0.3296403 3.946196 0.996653
##                    ACF1
## Training set 0.05351391
```

3.1 Residual Analysis - ARIMA Model

Below are the findings of residuals from linear model

```
checkresiduals(fit)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2)
## Q* = 9.2862, df = 6, p-value = 0.1581
##
## Model df: 4.   Total lags used: 10
```

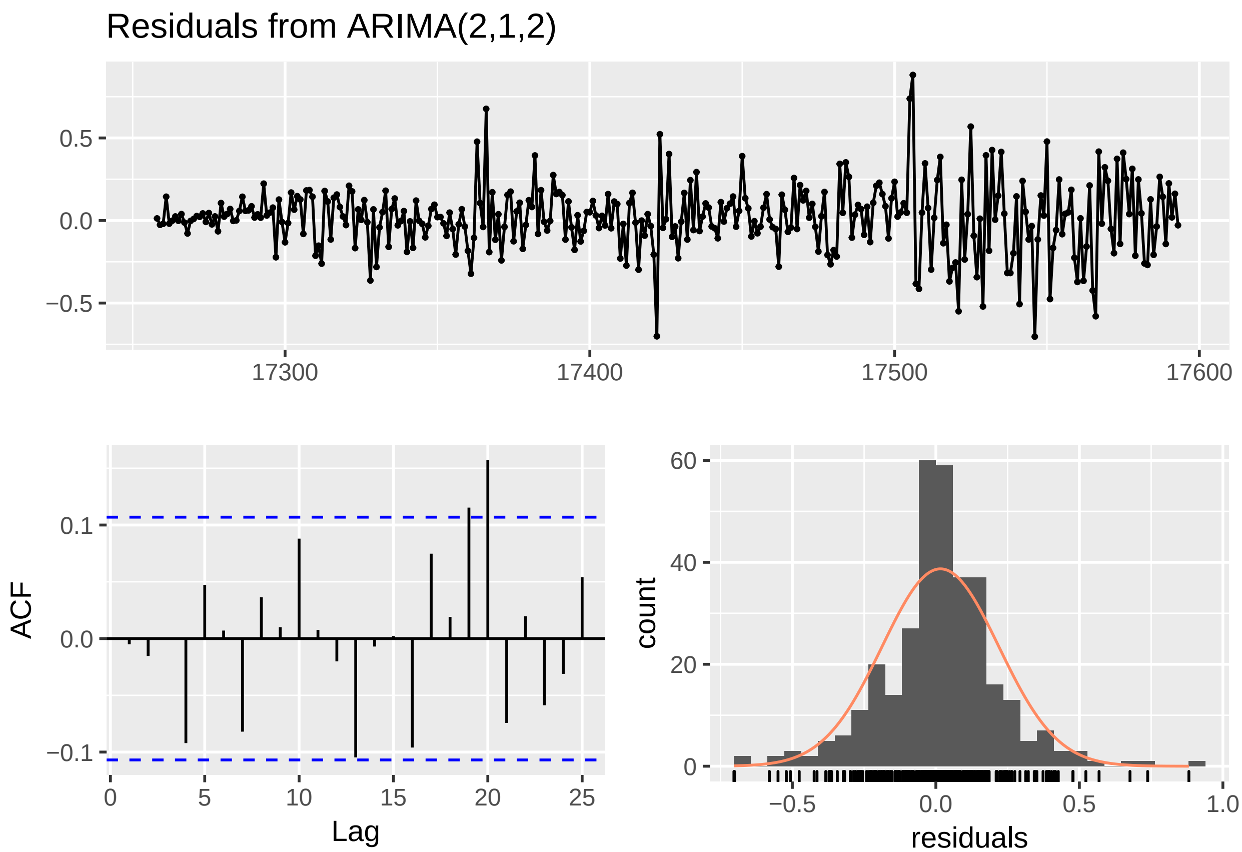


Figure 5: Residual Analysis Quadratic fitted Model

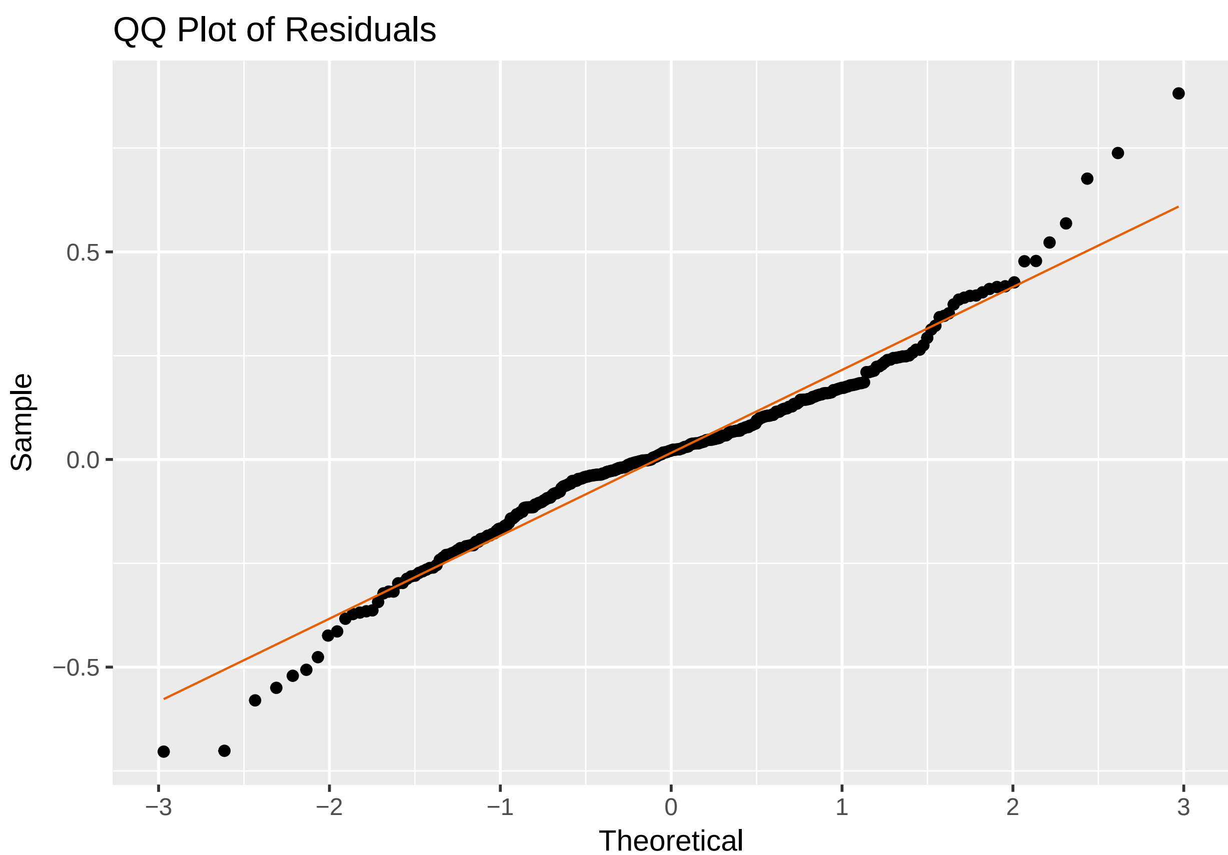


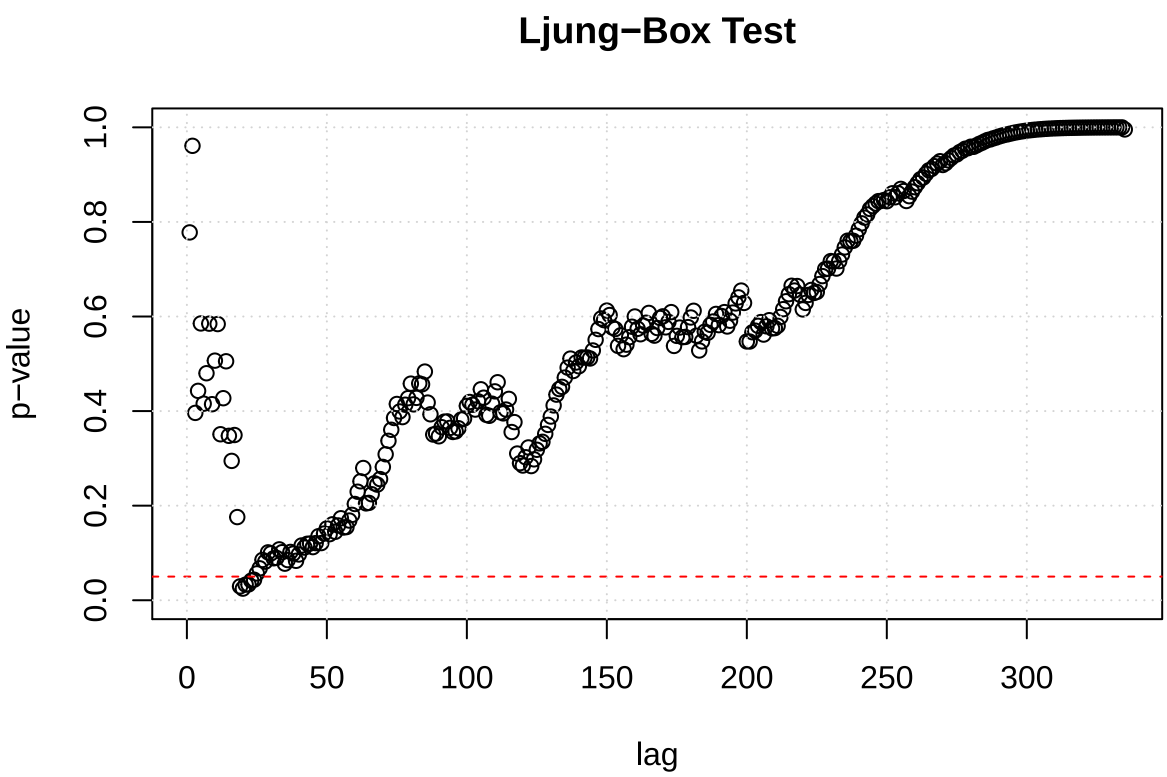
Figure 6: Residual Analysis Linear fitted Model

```
residual_analysis_qq(residuals(fit))
```

```
shapiro.test(as.vector(residuals(fit)))
```

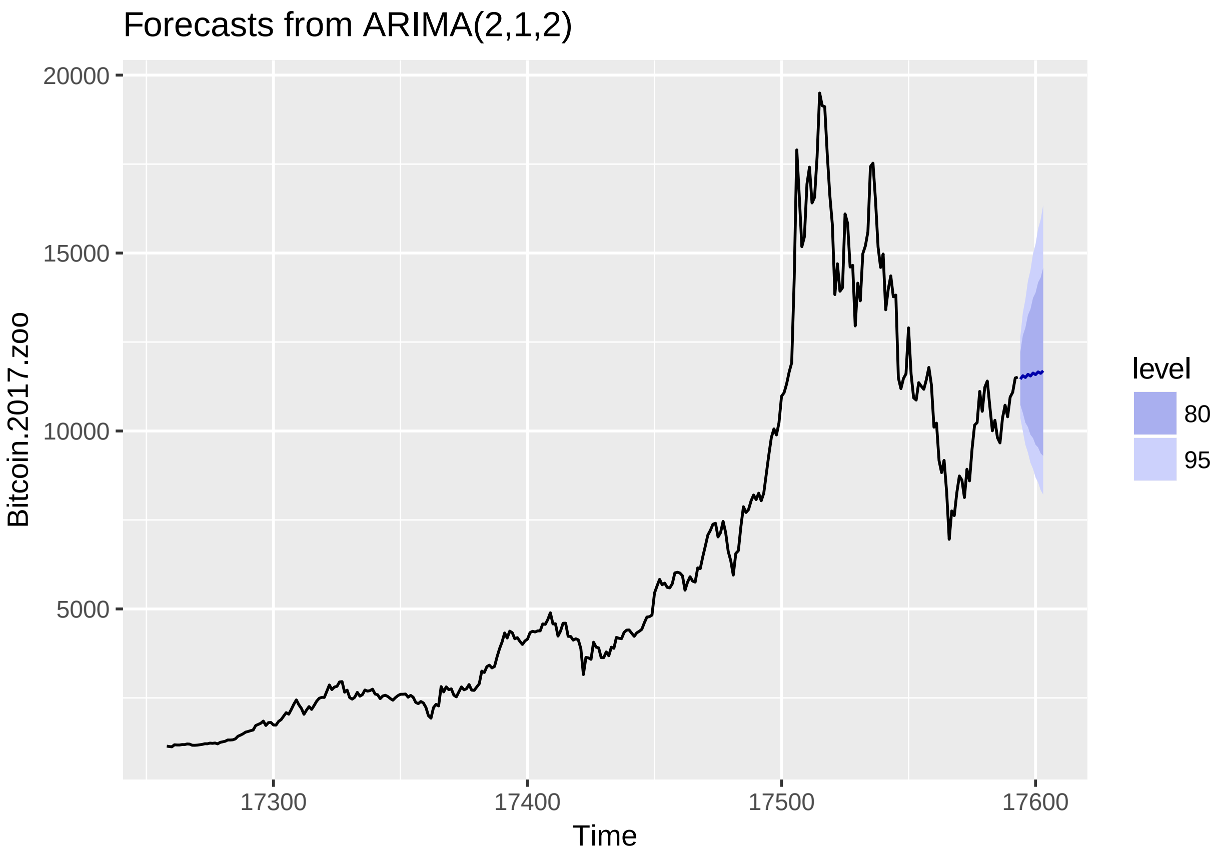
```
##
##  Shapiro-Wilk normality test
##
## data:  as.vector(residuals(fit))
## W = 0.96554, p-value = 3.852e-07
```

```
x = residuals(fit)
k=0
LBQPlot(x, lag.max = length(x)-1 , StartLag = k + 1, k = 0, SquaredQ = FALSE)
grid()
```



3.2 Forecast

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



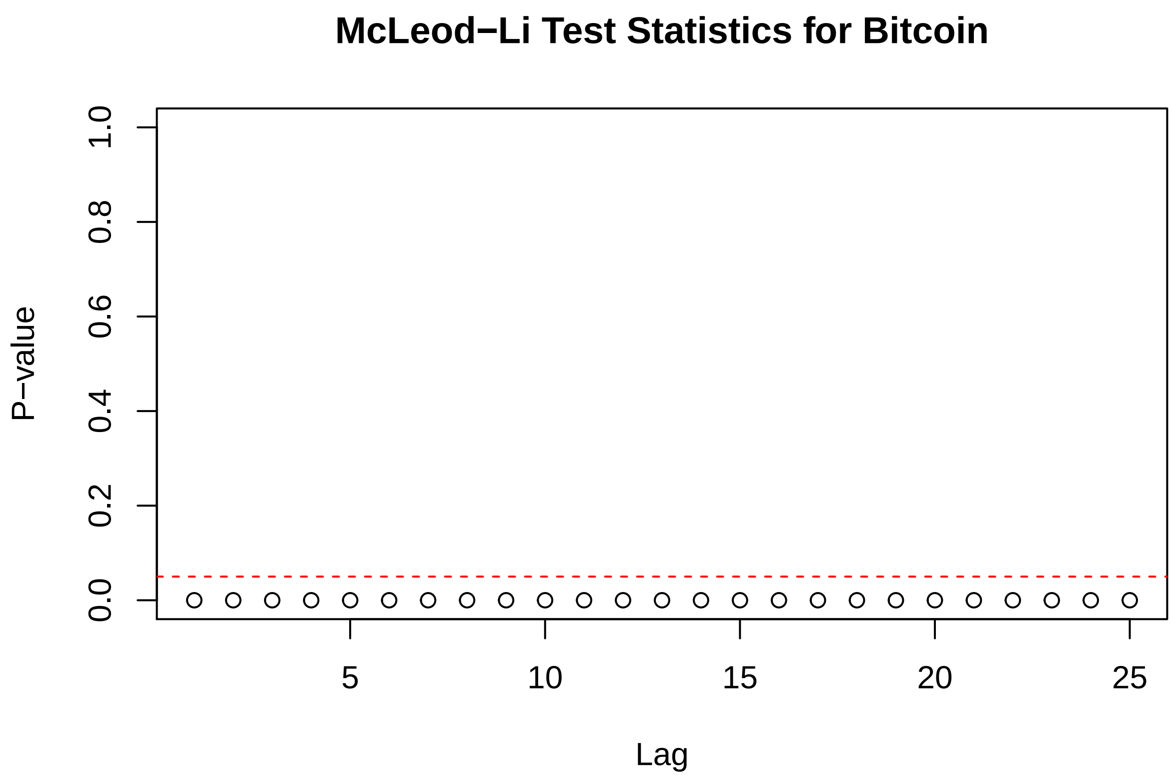

```r
Bitcoin.forecast <- read_csv("../data/Bitcoin_Prices_Forecasts.csv")
Bitcoin.forecast$Date = as.Date(Bitcoin.forecast$Date,'%d/%m/%y')
```

3.3 MASE Error

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

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

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




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