Analysis of Bitcoin Daily Closing Price

Akshay Sharma & Vishesh Jain

13 May 2018

INTRODUCTION

Any form of currency that only exists digitally relying on cryptography to prevent counterfeiting and fraudulent transactions is defined as cryptocurrency. Bitcoin was the very first Cryptocurrency. It was invented in 2009 by an anonymous person, or group of people, who referred to themselves as Satoshi Nakamoto.

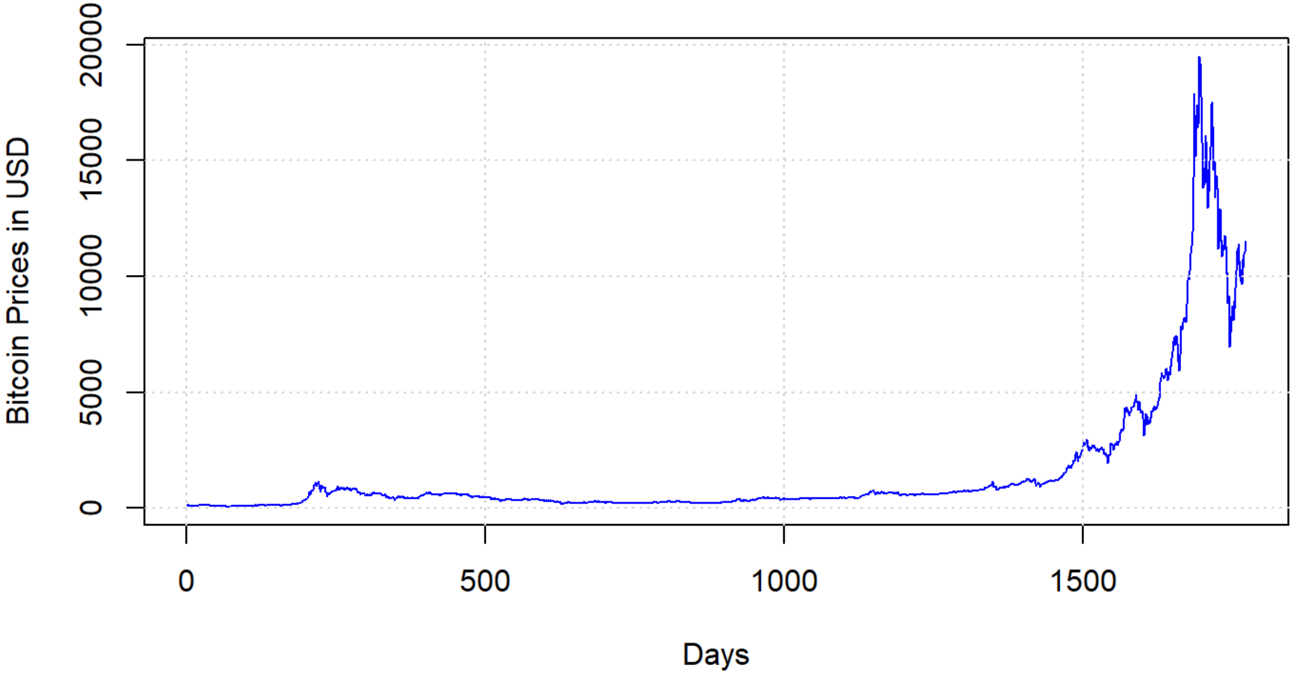
In April 2013, the value of 1 bitcoin (BTC) was around 100 USD. At the beginning of 2017 its value was 1,022 USD and by the 15th of December it was worth 19,497 USD. As of the 3rd of March 2018, 1 BTC sells for 11,513 USD. So, the time series analysis of bitcoin series is very challenging.

The challenge is to find the best fitting model to the given cryptocurrency series and Predict the value of bitcoin for next **10-days**. The data-set used is the daily closing price of bitcoin from **27-April-2013** to **03-March-2018**. The data has been gathered from https://coinmarketcap.com/

NOTE: The performance of the model will be measured using **Mean-Absolute Scaled Error (MASE)**. We will use the real values of bitcoin for 10 days of forecast period **(4 to 13-March-2018)**.

DATA EXPLORATION - TIME SERIES PLOT

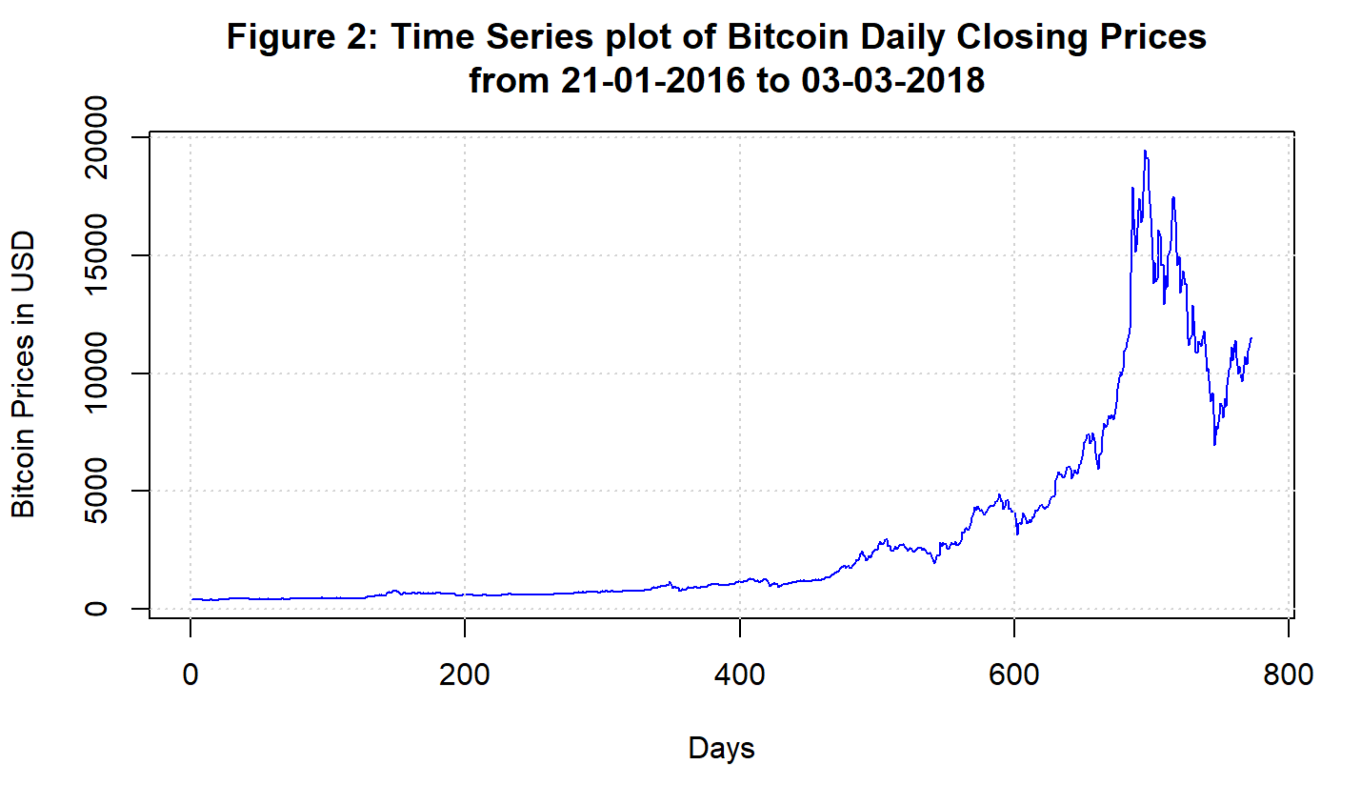
Figure 1: Time Series plot of Bitcoin Daily Closing Prices from 27-04-2013 to 03-03-2018



As we can see in this plot, there can be two different time series in the data

The purpose of this project is to predict future values, therefore, we will take the series after 1000 days i.e. from 21-January-2016.

```
bitcoin = bitcoin[1000:1772] # Taking closing prices from 1000 day (21-01-2016)
```



NORMALITY DISTRIBUTION OF SERIES

We can clearly see through figures 3 & 4, that the distribution of the series is not normal, this is a right-skewed distribution.

Figure 3: Normal Q-Q Plot

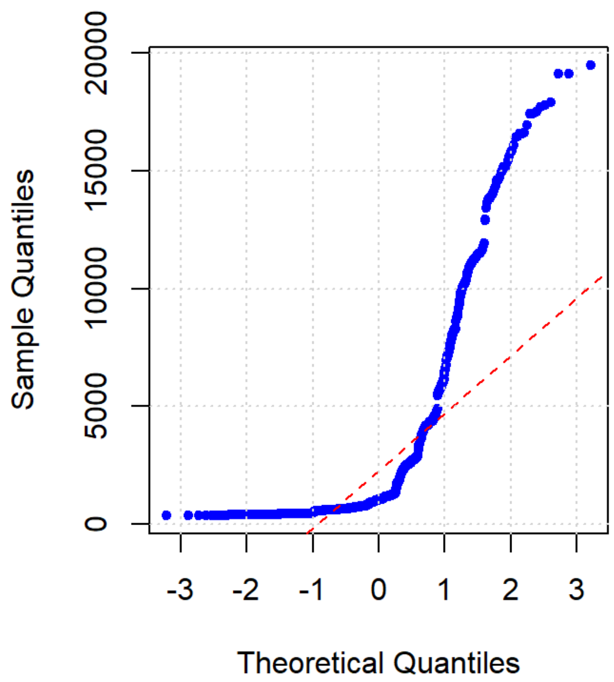
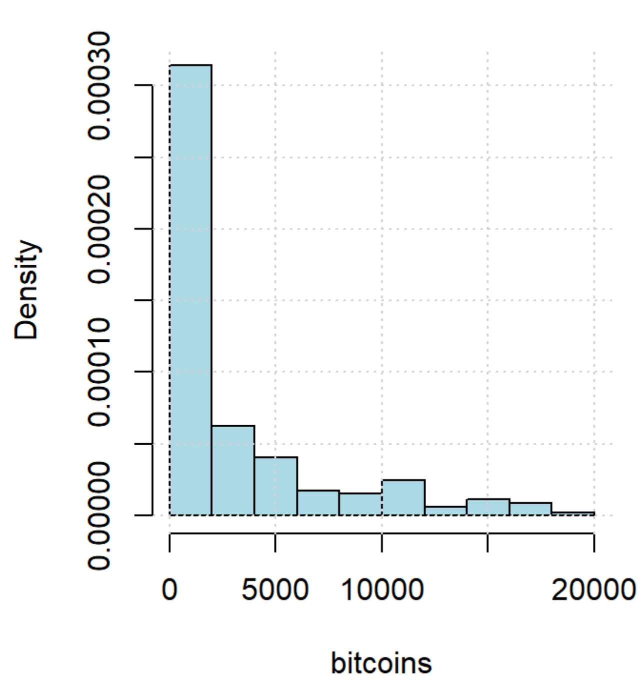


Figure 4: Histogram Plot



Therefore, we will employ a Box-Cox Search method to find the best value of lambda and transform the series using:

$$\mathbf{X_t} = \left(\frac{X^\lambda - 1}{\lambda}\right)$$

```
BoxCoxSearch(bitcoin, plotit = F) # Using normality tests to find lambda
```

Through above search, we found

$$\lambda = -0.5$$

We will transform the series using the given formula:

```
bit = (bitcoin^(-0.5)-1)/(-0.5)
```

We can see that the distribution of the series has considerably improved after the transformation. As per figure 5, data is aligned with the normality line except for fat tails which is expected in a financial time-series.

Figure 5: Normal Q-Q Plot

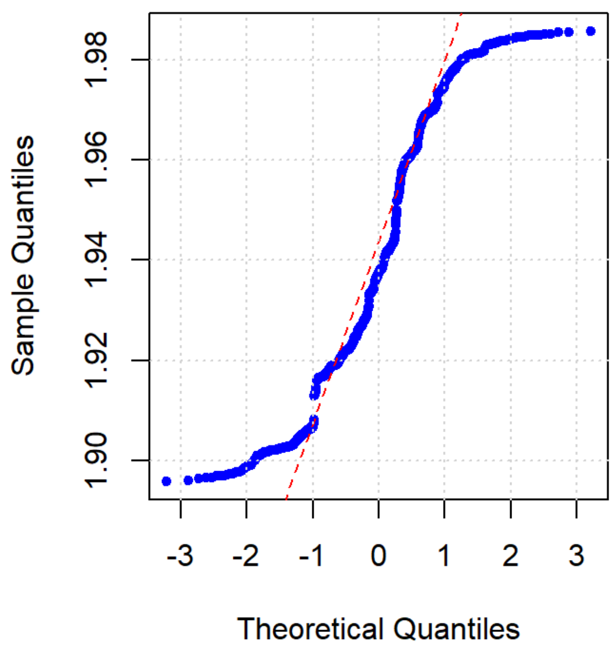
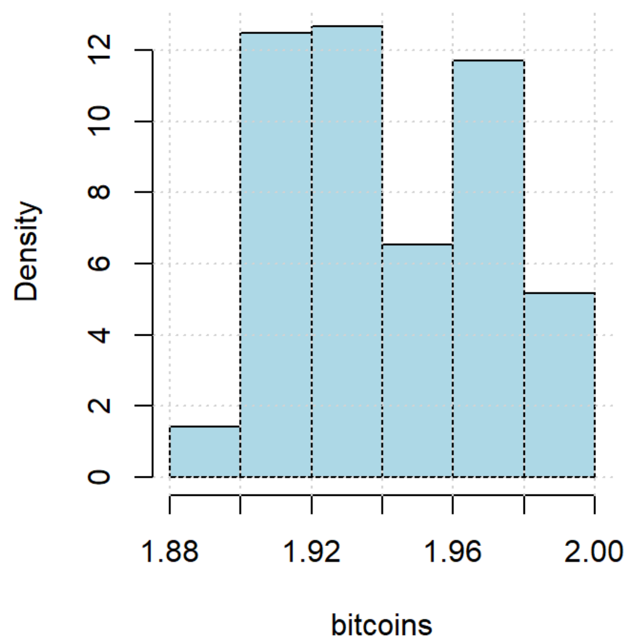
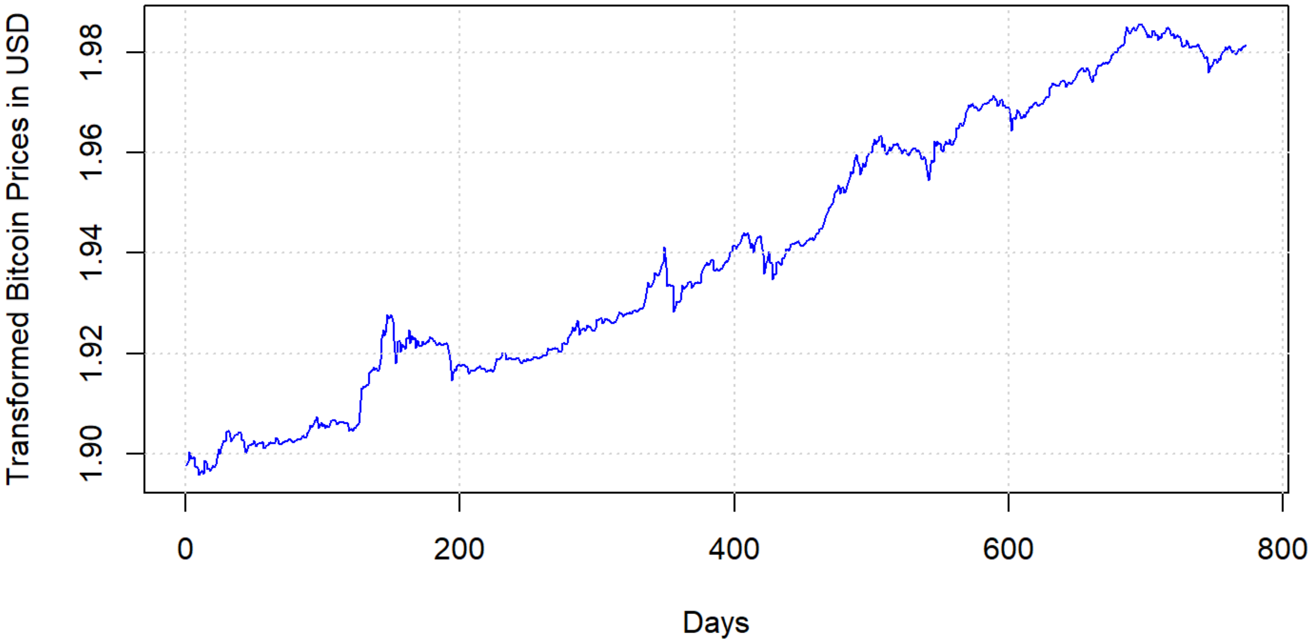


Figure 6: Histogram Plot



BOX-COX TRANSFORMED SERIES

Figure 7: Time Series plot of Transformed
Bitcoin Daily Closing Prices from 21-01-2016 to 03-03-2018



Observations:

- *Trend*: There is a clear upward trend in the series
- *Seasonality*: Seasonality is not clear in the series
- *Behavior*: This could be auto-regressive or moving-average or both
- *Variance*: Change in variance might not be present

Following the given observations, we will move ahead with ARIMA/ SARIMA modeling of the series.

Using ts() function from TSA package, the given series is converted to a time series object.

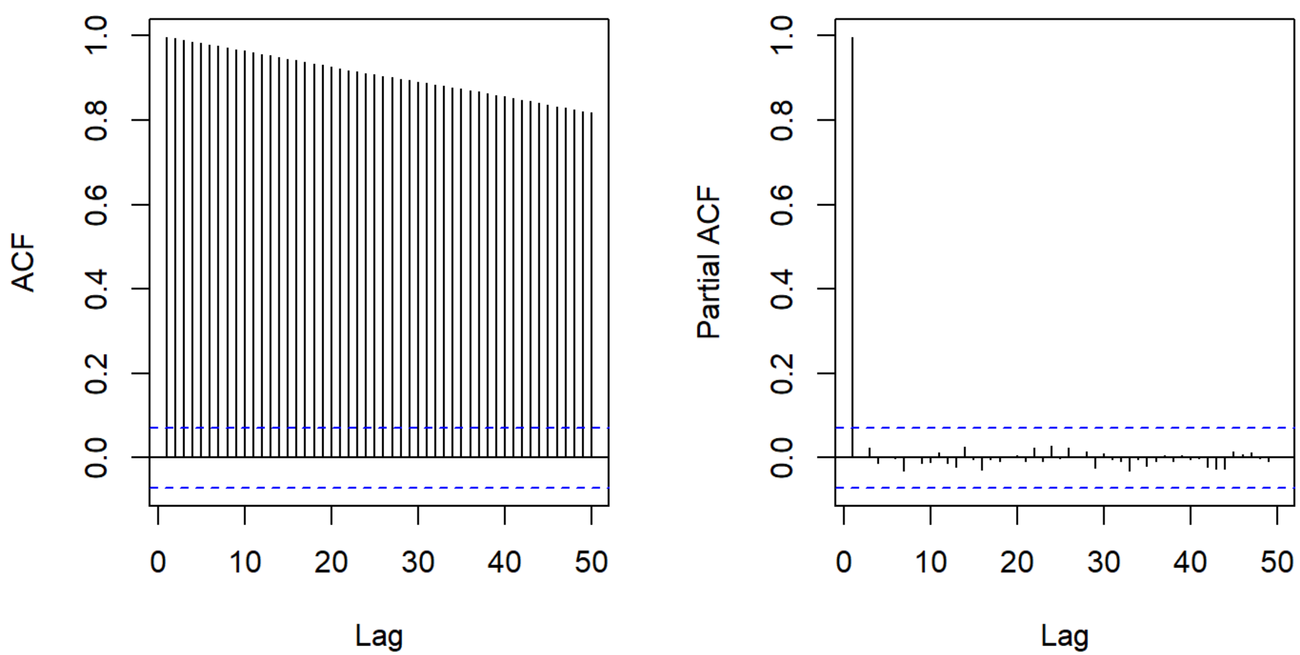
```
bit1 = ts(log(bit)) #Log is used to control the change in variance (if any)
```

STATIONARITY & CORRELATION IN THE SERIES

As seen in figure 8 & 9, there is ordinary trend in the series. Therefore, we will proceed with first difference of the series.

Figure 8: ACF Plot of Transformed Series

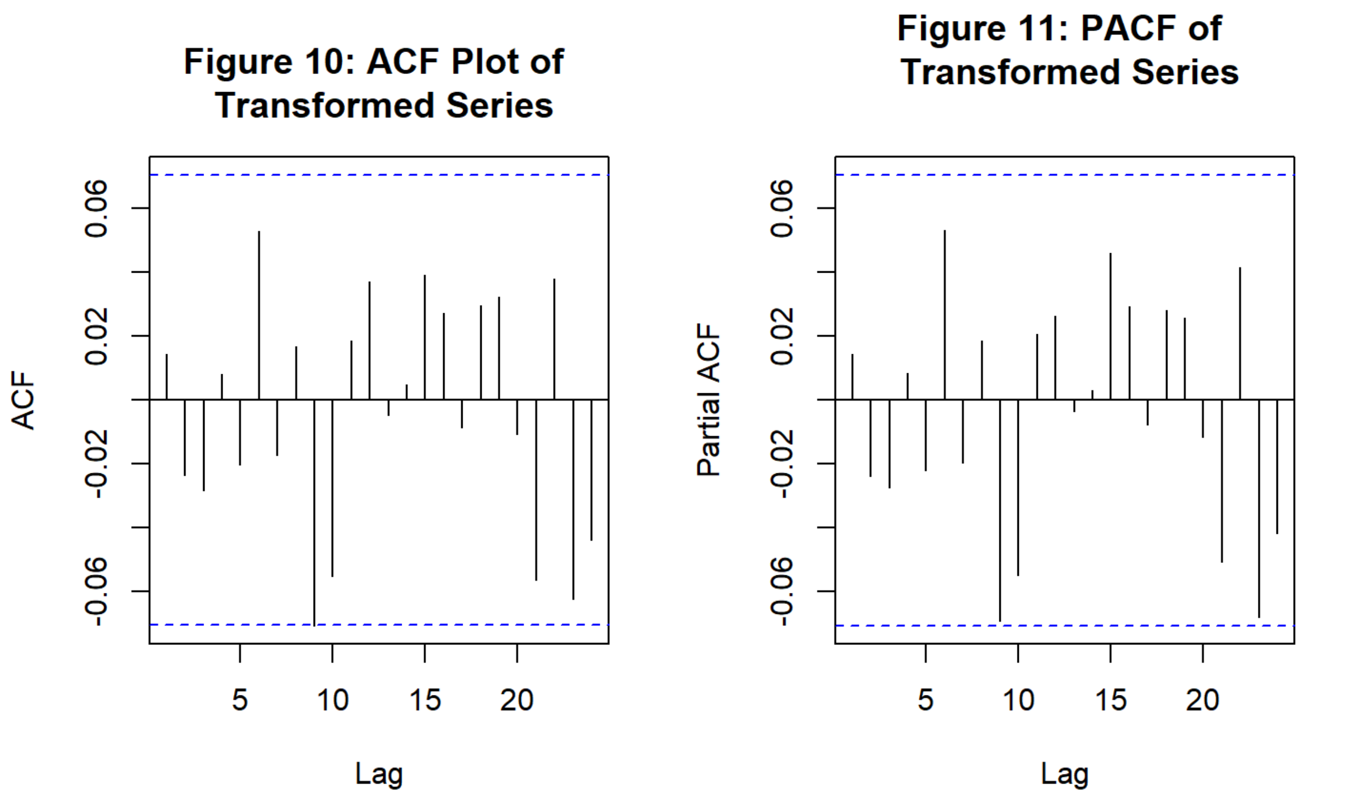
Figure 9: PACF of Transformed Series



```
bitd = diff(bit1)
```

From figure 10 & 11, we can confirm that there is no ordinary trend left in the series. Also, there are no significant AR/MA lags. However, we can see significant lags at period 6.

Note: This could be a sign of $weak seasonality$ in the series




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

ARIMA/ SARIMA MODELING

No-Differencing : ADF and KPSS test shows that the series is not stationary

First-Difference : ADF and KPSS test shows that the series is stationary now

Remember in ACF & PACF plots of the first-differenced series, there were no significant lags. Therefore, we will use **EACF** and **BIC** to find the suitable candidate models.

EACF - Possible candidate models are: $ARIMA(1,1,1), ARIMA(0,1,1) ARIMA(2,1,2)$

BIC - Possible models are: $ARIMA(3,1,6), ARIMA(6,1,6), ARIMA(2,1,3), ARIMA(2,1,6)$

MODEL FITTING

Through model fitting and coefficient testing, we found that ARIMA(2,1,2) is the best fit model.

After conducting the coefficient test fo ARIMA(2,1,2), we found 'NA' in the standard error and z-value of the model, which means indicates towards seasonality in the data.

We will next explore the seaonal model.

```
m.arima = arima(bit1, order = c(2,1,2))
res.arima = m.arima$residuals
coeftest(m.arima)
```

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

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -1.65207         NA      NA       NA
## ar2 -0.77162         NA      NA       NA
## ma1  1.68510         NA      NA       NA
## ma2  0.81629         NA      NA       NA
```

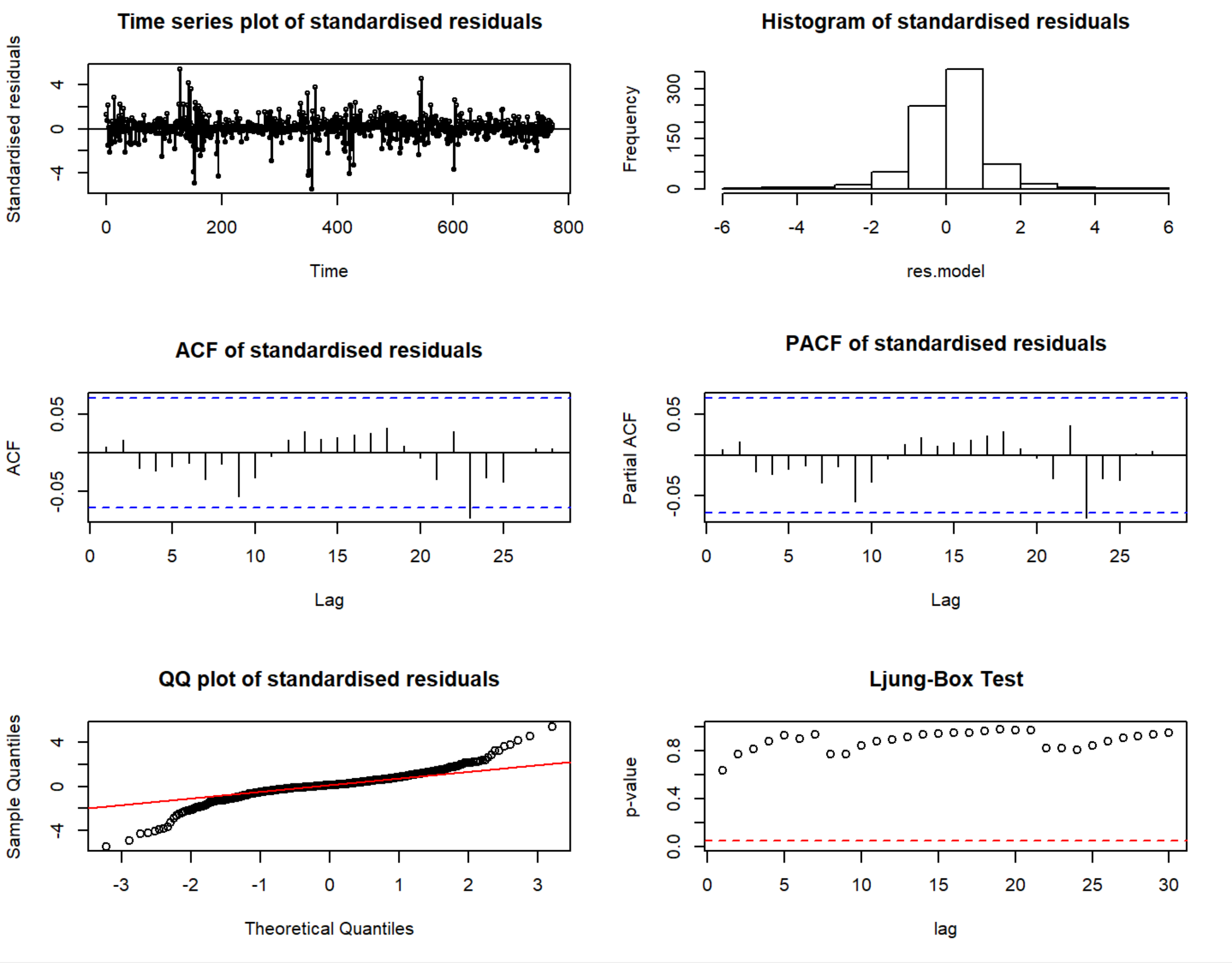
This was modeled using a SARIMA(2,1,2)X$(0,0,1)_6$

```
m1 = Arima(bit1, order = c(2,1,2), seasonal = list(order = c(0,0,1), period = 6))
coeftest(m1)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  -0.214893   0.026120  -8.2271 < 2.2e-16 ***
## ar2  -0.967585   0.029948 -32.3089 < 2.2e-16 ***
## ma1   0.229169   0.037480   6.1144 9.692e-10 ***
## ma2   0.943695   0.038219  24.6916 < 2.2e-16 ***
## sma1  0.095924   0.037659   2.5472   0.01086 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the parameteres are found to be significant at 5% significance level.

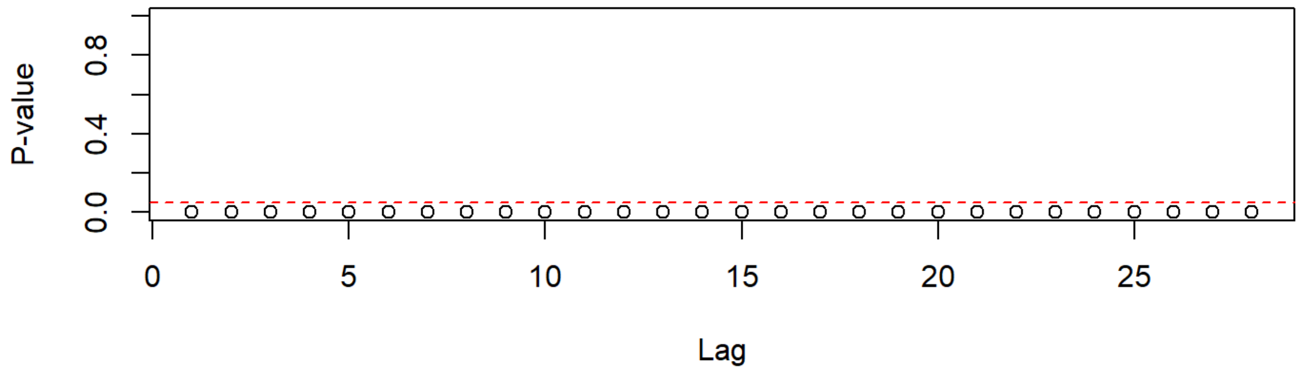
RESIDUAL ANALYSIS



GARCH MODELING

Although, the residuals of the fitted model shows no auto-correlation, there might be presence of **volatility-clustering** in the series. We will perform a test to check the presence of this clustering.

Figure 13: Mc Leod Li Test of Residuals



The McLeod Li Test confirms the presence of volatility clustering at all lags. To model this clustering, we will use absolute and squared residuals and analyse ACF, PACF & EACF for possible GARCH models.

ACF & PACF - ABSOLUTE & SQUARE RESIDUALS

Figure 14: Absolute Residuals ACF

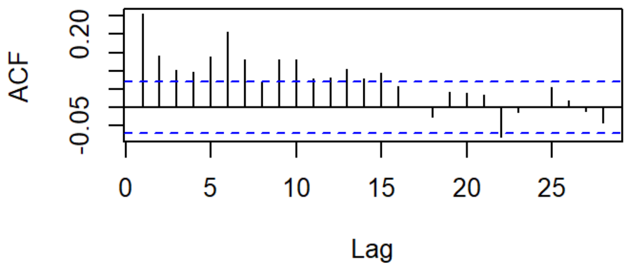


Figure 15: Absolute Residuals PACF

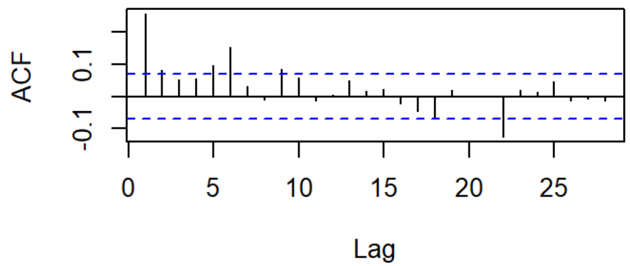


Figure 16: Squared Residuals ACF

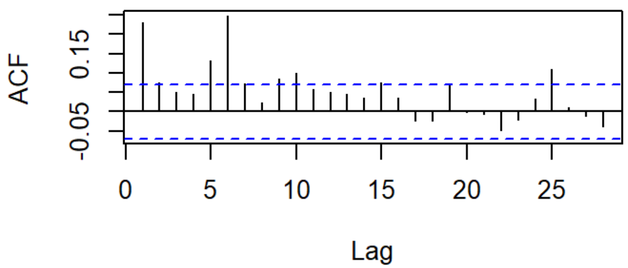
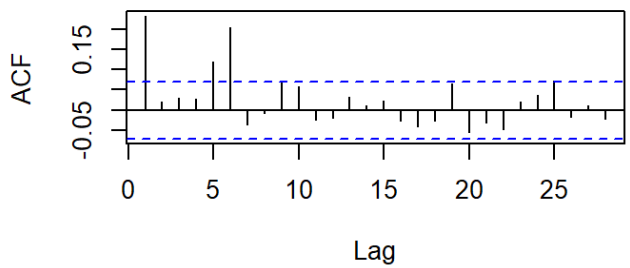


Figure 17: Squared Residuals PACF



We can see many significant lags from the ACF & PACF plots. The p value found from EACF of absolute residuals is 1,2 and from EACF of squared residuals is 2,3.

```
## [1] "----Absolute Residuals EACF-----"
```

GARCH MODELS

Through ACF, PACF & EACF of the absolute and squared residuals, we found possible models as; GARCH(1,2), GARCH(2,2), GARCH(1,3), GARCH(2,3)

After, comparing significance of the coefficients and AIC values of the possible candidate models, we found GARCH(1,2) with **AIC=-2663.847** as the best fitted model for the residuals.

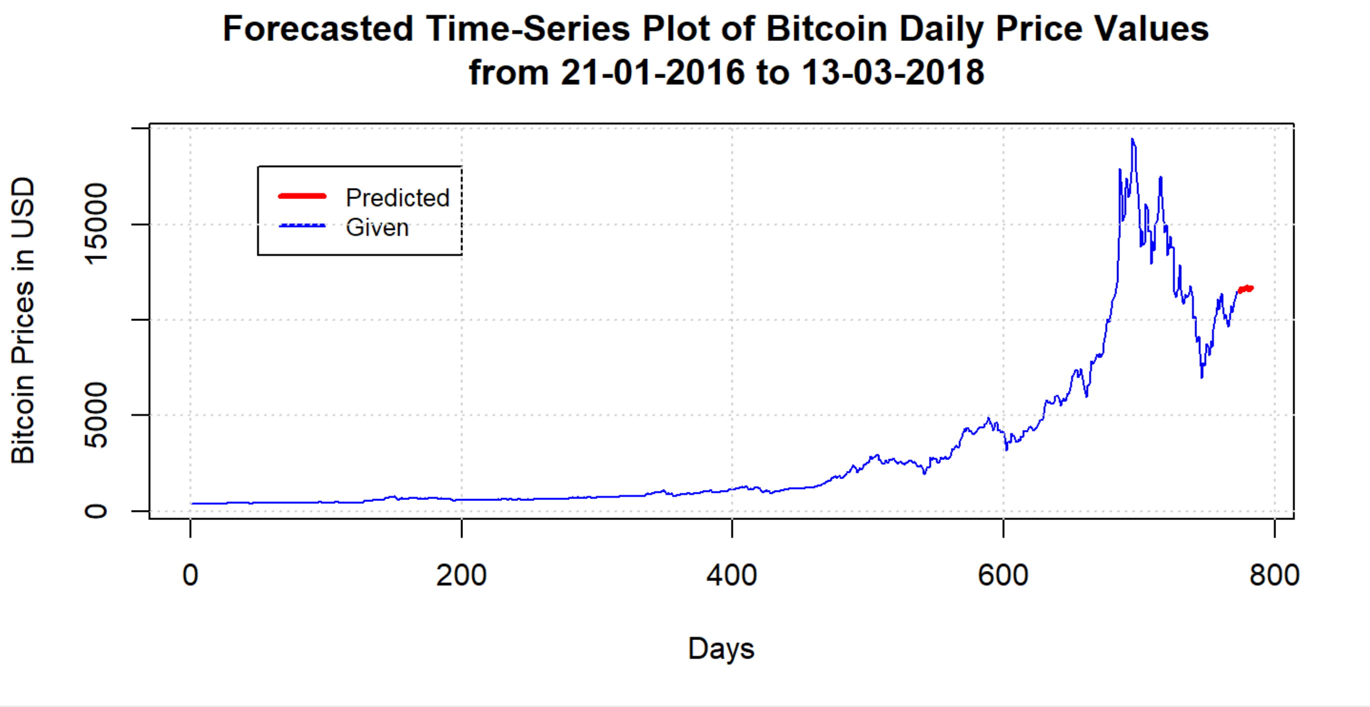
All the parameters are significant at 5% significance level.

```
m2<- garch(bitr*100, order = c(2,1), trace = F)
# arima_garch_model = ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
#                    mean.model = list(armaOrder = c(1, 0), include.mean = FALSE),
#                    distribution.model = "norm")
# arima_garch_fit <-ugarchfit(spec=arima_garch_model,data=diff(bit1))
# arima_garch_fit
```

SARIMA(2,1,2)X$(0,0,1)_6$ + GARCH(1,2) FORECASTING

To proceed with this, we will first predict the conditional variance for the next 10 days.

Next, we will forecast the 10-ahead values from SARIMA model and add the conditional variance found earlier. Then we will untransform the predicted values.



PERFORMANCE MEASURE

As mentioned earlier, **MASE** will be used to check the performance of the model. The series as found during the analysis is seasonal, therefore, **MASE** function for seasonal series will be used.

$$\mathbf{MASE} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{|e_t|}{\frac{1}{T-m}\sum_{t=m+1}^{T}|Y_t - Y_{t-m}|}\right)$$

$m = 6 : seasonal period$

```
observed = as.numeric(realVal$Closing.price)
fitted = as.numeric(bitrl.2[774:783])

MASE(observed, fitted)
```

```
## $MASE
##        MASE
## 1 0.7261805
```

MASE = 0.7261805