

***Servet Kocak***

***Arjun Khopkar***

***Syed Hassan Afsar***

***Radhika Zawar***

**Bitcoin Trend – Model Fitting and Forecasting**

*Time Series Analysis*

*MATH 1910*

*Final Project*

**Bitcoin Trend – Model Fitting and Forecasting**

Table of Contents

[Introduction 3](#_Toc10798315)

[Aim of the Report 3](#_Toc10798316)

[Analysis of Raw Data 3](#_Toc10798317)

[Trend Analysis 4](#_Toc10798318)

[Correlation 5](#_Toc10798319)

[Stationarity Check 6](#_Toc10798320)

[Transformation 6](#_Toc10798321)

[Differencing 7](#_Toc10798322)

[Changing Variance Check 8](#_Toc10798323)

[ARIMA Model Selection 8](#_Toc10798324)

[Candidate Model Selection from ACF and PACF 9](#_Toc10798325)

[Candidate Model Selection From EACF 9](#_Toc10798326)

[Candidate Model Selection from BIC table 9](#_Toc10798327)

[Parameter Estimation for Candidate ARIMA Models 10](#_Toc10798328)

[Model Fitting 21](#_Toc10798329)

[GARCH Model Selection 21](#_Toc10798330)

[Forecasting 27](#_Toc10798331)

[MASE 28](#_Toc10798332)

[Between the observed and the predicted value 28](#_Toc10798333)

[Between the observed and the fitted model 29](#_Toc10798334)

[Conclusion 29](#_Toc10798335)

[References 29](#_Toc10798336)

[Appendix 29](#_Toc10798337)

# Introduction

In recent years, virtual currencies also known as cryptocurrencies have gained immense popularity amongst traders because of its highly volatile nature. Bitcoin is the most popular form of cryptocurrency which has managed to capture maximum market share over the years.

**About Bitcoin: -**

* It was established and released as an open software by Satoshi Nakamoto in the year 2009 but extensive trading and mining only began in late 2016.
* The cryptocurrency is solely traded using blockchain technology.
* Over the years, Bitcoin has faced harsh criticism from various regulatory financial bodies because of its extremely high price volatility, thefts, high energy consumptions and the possibility of it being a massive economic bubble.

# Aim of the Report

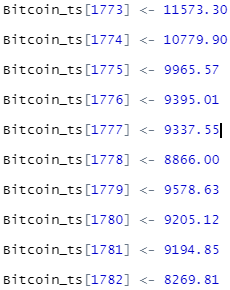
1. **Trend Analysis** - Analysing the bitcoin trend over the time period 2013-04-27 to 2019-02-24 and making observations regarding the various features of the time series.
2. **Model Fitting and Diagnostics** - Using RStudio and statistical analysis to fit a model on the time series which captures most of the data in the series and minimizes residuals.
3. **Forecasting** - Prediction of 10 daily values for the time period 2019-02-25 to 2019-03-06 using the model estimated in step 2.

# Analysis of Raw Data

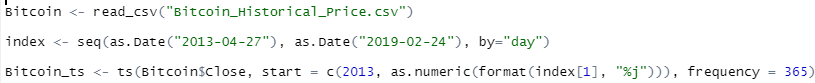
Daily closing bitcoin values for the period 2013-04-27 to 2019-02-24 were used for the analysis. The data was obtained in CSV format from *coinmarketcap.com*.

During the import of the file for analysis it was observed that the daily closing values for the period 4/3/18 to 13/3/18 were provided in a different format. The values were handled by *manual imputation*.

**Note**: *The values were imputed with the same values as provided and did not change the trend in any way.*



The CSV file was then read into R and subsequently converted to a time series object for further exploration and analysis.



# Trend Analysis

At the outset, the bitcoin time series plot (Refer *Figure 1)* for the given period was generated and examined for the following: -

1. **Trend**: The plot displayed an obvious upward trend
2. **Changing variance**: Highly volatile movement in the value of the bitcoin caused changing variance
3. **Intervention Point**: A sudden upward movement was observed towards the start of 2017. According to a research done by CNBC finance it was analysed that the spike in 2017 was majorly due to heavy price manipulations using another cryptocurrency named Tether.

<https://www.cnbc.com/2018/06/13/much-of-bitcoins-2017-boom-was-market-manipulation-researcher-says.html>

1. **Seasonality**: The trend displayed no sign of seasonality
2. **Behaviour**: Fluctuating successive points implied a moving average characteristic along with succeeding observations that implied the existence of autocorrelation.

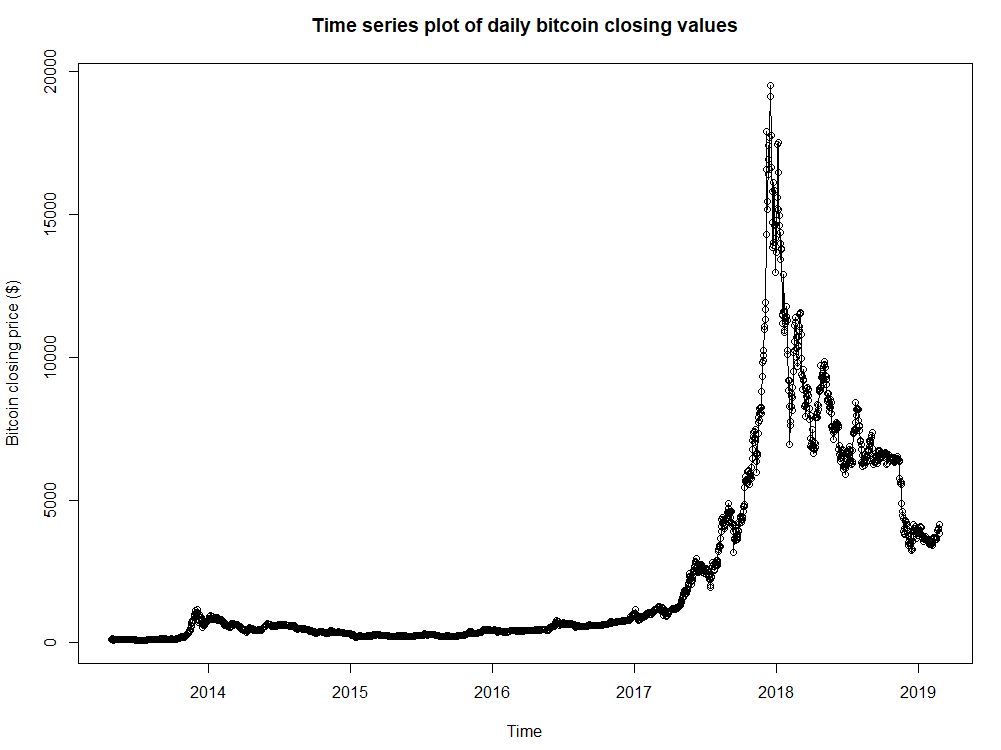


Figure 1 : Bitcoin time series trend plot

# Correlation

The correlation between the current daily closing value and the value lagged by one time period (in this case one day) was checked quantitatively as well as visually

**Visual –** A Scatterplot was generated to check for any obvious visual correlation. (Refer *Figure 2)*

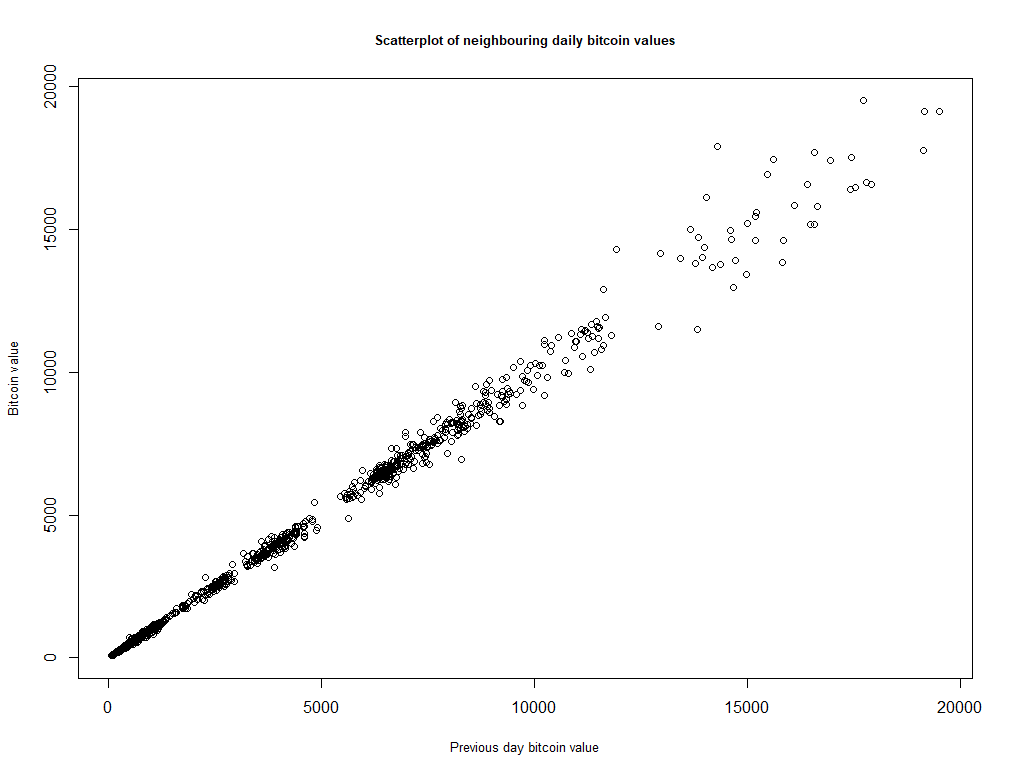
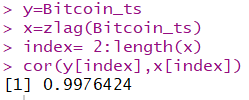


Figure 2: Scatterplot (Current daily value vs lagged by 1 day)

**Quantitative** – As an additional check the correlation between the two values was obtained using the zlag function and the cor function available in RStudio.



From both the scatterplot and the correlation function it was observed that there was a high positive linear correlation between the daily closing values and the data lagged by one day.

# Stationarity Check

The stationarity of the raw bitcoin time series was checked using visual and quantitative measures.

**Visual** – The ACF and PACF function available under RStudio was used to check the stationarity visually.

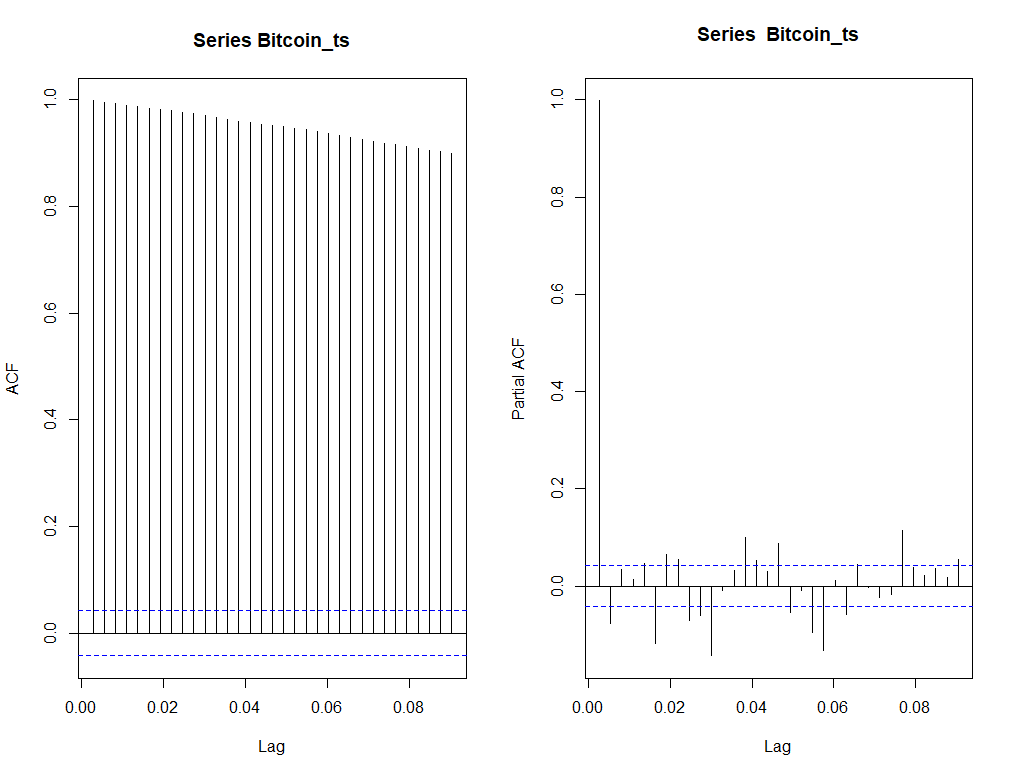
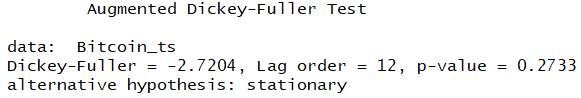


Figure 3: ACF and PACF of raw bitcoin time series

**Quantitative** – The Augmented Dickey Fuller Test (“ADF”) was run to test the bitcoin time series against the null hypothesis of the test (Series is non-stationary)

The test was performed using the adf.test function available in RStudio.



A slowly decaying pattern in the ACF and one highly significant lag in the PACF suggested that the time series data was not stationary and suggested the existence of a trend. (Refer *Figure 3)*

Further, the p-value observed in the ADF test (0.2733) signified that the rejection of the null hypothesis of non- stationarity was not possible.

Thus, it could be concluded that the raw bitcoin time series was not stationary and the data would require transformation and differencing before it could be used for any further model exploration and analysis.

# Transformation

At the outset of converting the time series to a stationary dataset for model fitting, the boxcox test was run to generate a suitable lambda value to be used for a transformation which would be appropriate for the given time series.

The lambda observed was 0, which suggested the application of a logarithmic transformation on the original time series data.

Post applying the transformation, the stationarity of the time series was checked again using the ACF, PACF (Refer *Figure 4)* and ADF test.

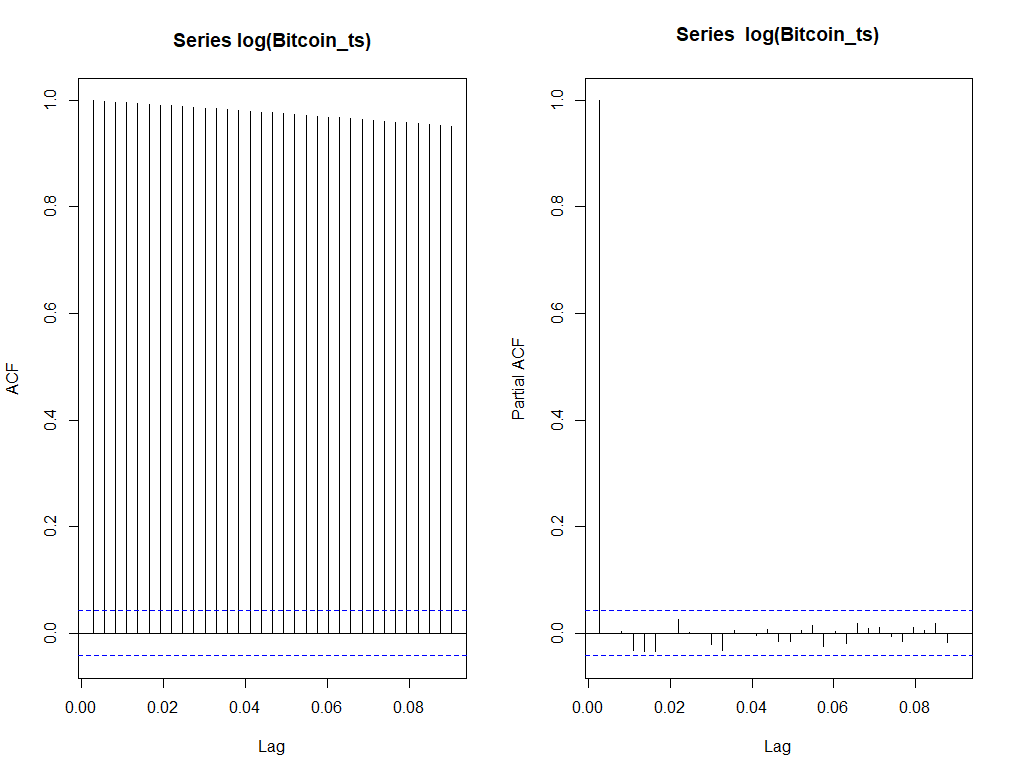
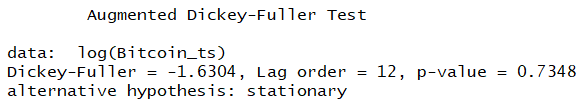


Figure 4: ACF and PACF of log transformed time series



A slowly decaying pattern in the ACF and one highly significant lag in the PACF suggested that the log transformed time series data was still not stationary and existence of the trend still persisted. (Refer *Figure 4*). The ADF test confirmed the same quantitatively.

# Differencing

First order differencing was applied on the log transformed time series to help achieve stationarity.

The trend plot of the log differenced bitcoin time series was inspected visually as well as a quantitative check was added to ensure that the time series had achieved required level of stationarity for the purpose of model fitting and parameter estimation.

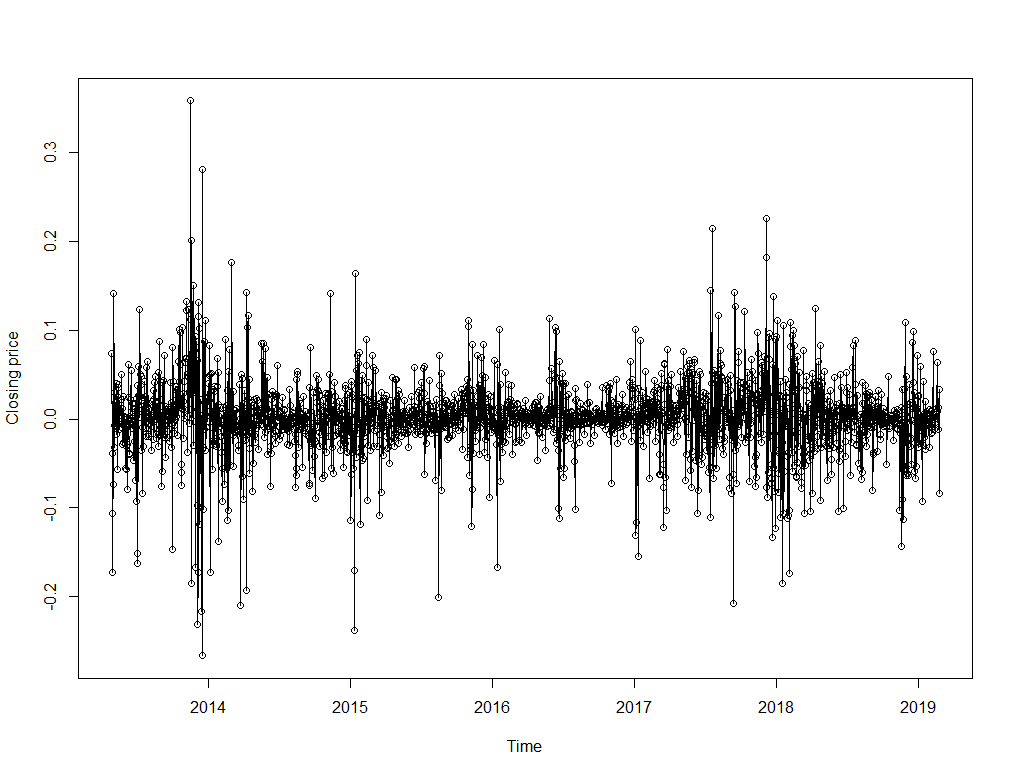
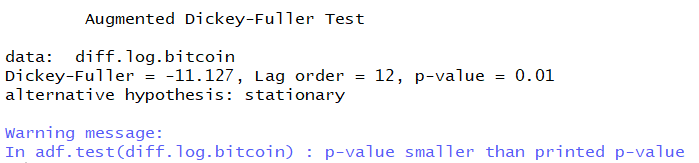


Figure 5: Trend plot of log differenced time series

The plot (Refer *Figure 5)* suggested that the trend was removed from the data series post the transformation and first order differencing.



The p-value of the ADF was observed to be lesser than 0.01 which suggested rejection of the null hypothesis of non-stationarity of the data series.

The log differenced time series could be used for further analysis and model exploration.

# Changing Variance Check

As a part of the initial assessment, the trend of the time series plot suggested changing variance (Refer *Figure 1).*

To crystallize the assessment, the Mcleod Li test was performed on the log differenced bitcoin series.

All the points were observed to be below the red line which indicated that there was an ARCH effect on the model which would need to be considered during model fitting. (Refer *Figure 6).*

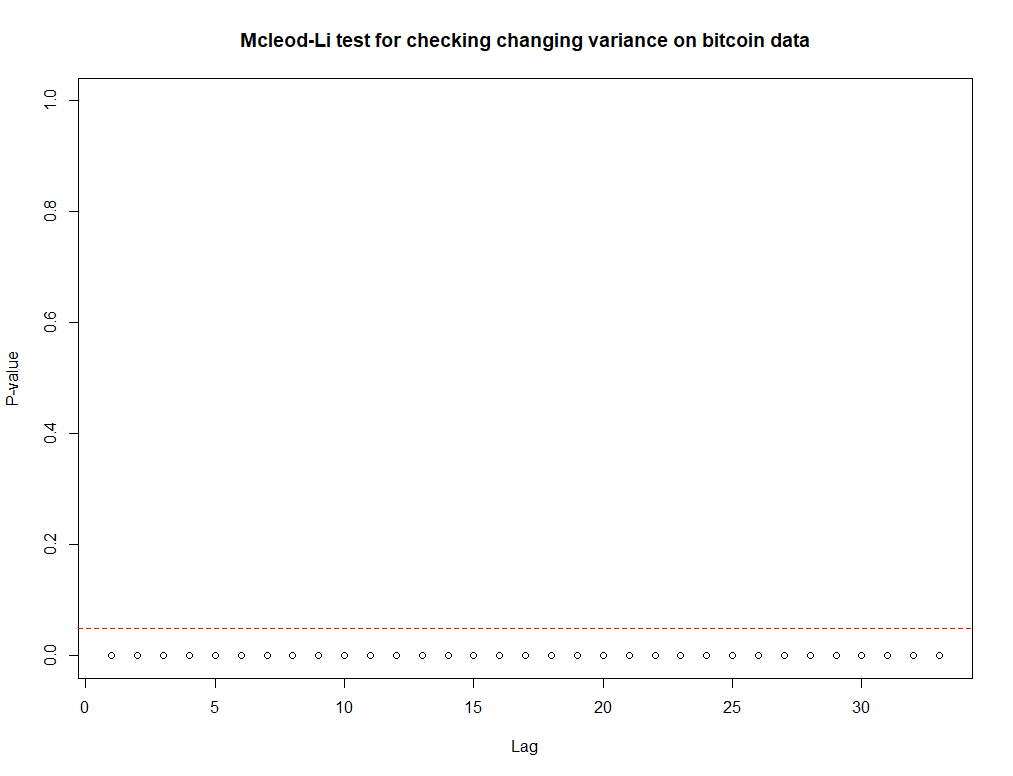


Figure 6: Mcleod Li test (Log Difference Series)

# ARIMA Model Selection

ARIMA models constitute a large class of models for the analysis of stationary and nonstationary time series.

In order to select candidate ARIMA models, properties of following functions are used:

* Sample autocorrelation function
* Partial autocorrelation function
* Extended autocorrelation function
* The Akaike information criterion (AIC)
* Bayesian Information Criterion (BIC) or Schwartz’s Bayesian Criterion table

## Candidate Model Selection from ACF and PACF



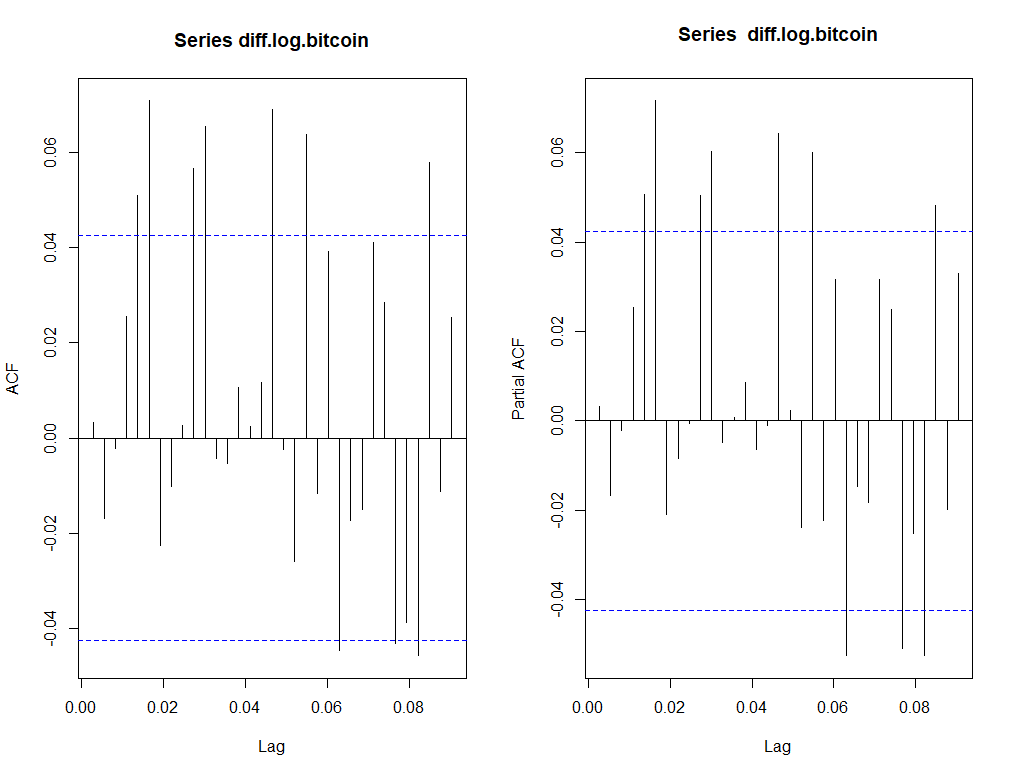


Figure 7: ACF and PACF of log transformed data series

From the ACF and PACF plot(Refer *Figure 7)*, it is clear that there was obvious ARCH effect; therefore, no model was selected from these plots.

## Candidate Model Selection From EACF



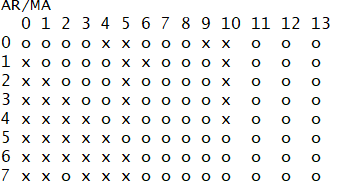


Figure 8:EACF plot for log transformed data series

From the EACF plot (Refer *Figure 8),* a clear vertex was observed at (1, 1), (1, 2) and (2, 2). Thus ARIMA (1,1,1), ARIMA (1,1,2), ARIMA (2,1,2) were shortlisted in the set of candidate model.

## Candidate Model Selection from BIC table

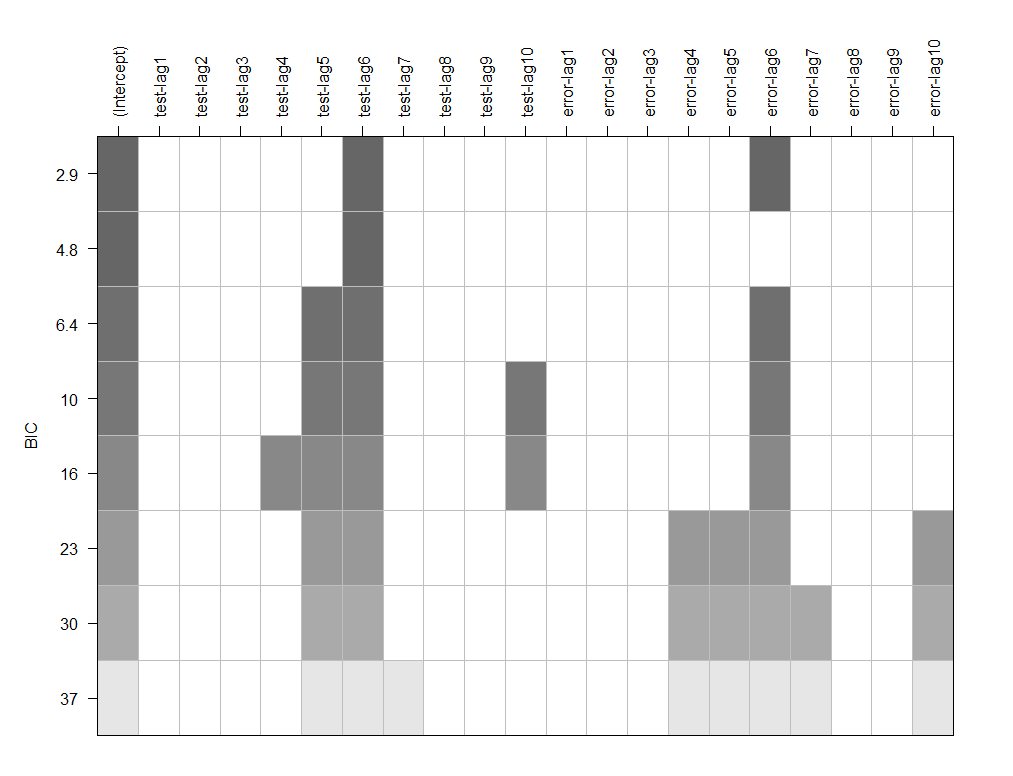


Figure 9 : BIC plot for the log transformed data series

From the BIC table (Refer *Figure 9),* AR (5), AR (6), MA (5) and MA (6) were observed to be significant. Hence ARIMA (5, 1, 5) and ARIMA (6, 1, 6) were added to the set of candidate models.

From ACF, PACF, EACF and BIC table, selected candidate models were:

*ARIMA (1,1,1), ARIMA (1,1,2), ARIMA (2,1,2), ARIMA (5,1,5) and ARIMA (6,1,6).*

## Parameter Estimation for Candidate ARIMA Models

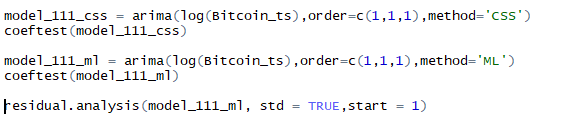
Two methods were used to estimates parameters:

* Maximum Likelihood estimation (“ML”)
* Conditional sum of square estimation (“CSS”)

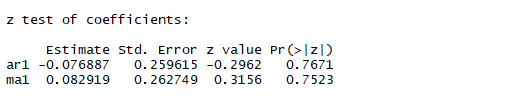
For residual analysis,

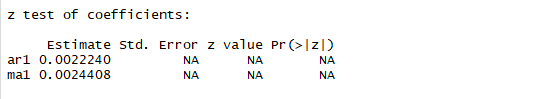
* **Normality** was checked using time series plot, histogram and QQ plot
* **Autocorrelation** was checked using ACF, PACF and Ljung-box test.

**ARIMA (1, 1, 1)**



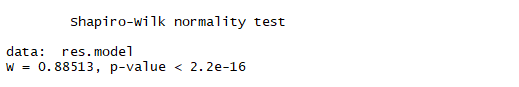
**Output:**





P value was observed to be greater than 0.05 for AR (1) and MA (1) coefficient, which signified that AR (1) and MA (1) were both are insignificant using CSS method. The ML method produced NA ‘s for both parameters.

**Residual Analysis ARIMA (1, 1, 1):**



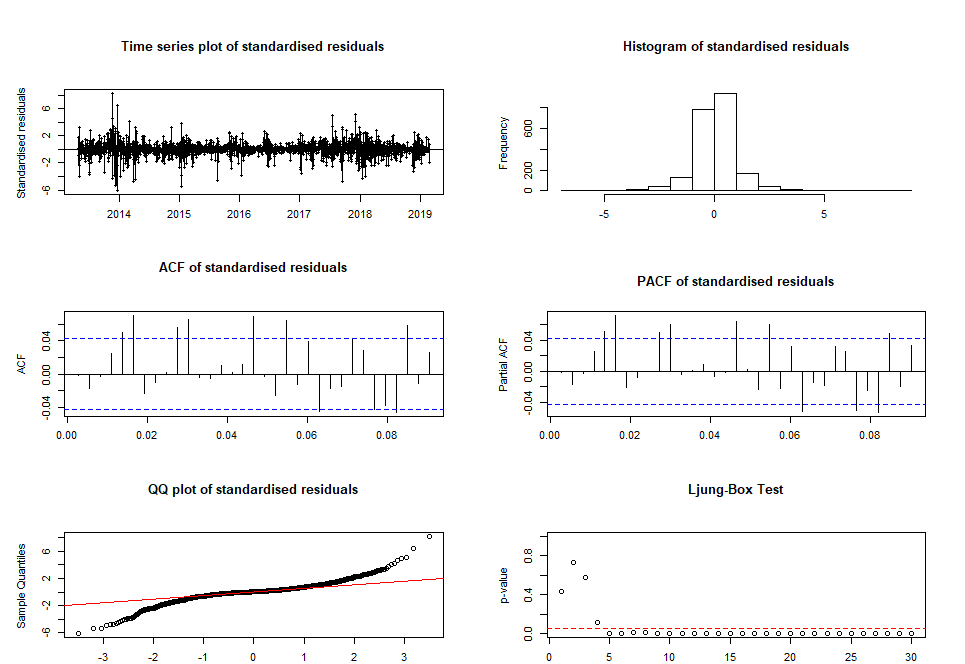


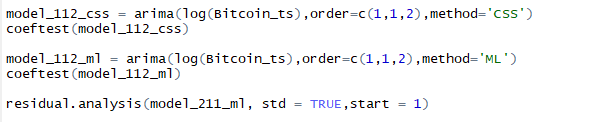
Figure 10 : Residual Analysis for ARIMA (1,1,1)

From *Figure 10* we observe:

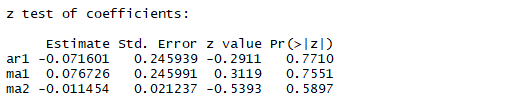
* **Residuals plot**: Although the plot showed the specified model de-trend the trend in the series, the changing variance was still very obvious in the residuals
* **Histogram** could not be viewed as symmetric
* **Q-Q plot**: Significant number of departing points from the red dash concluded that the normality assumption did not hold for this series. The thick tail implied existence of an ARCH effect in the series.
* **Shapiro-wilk test**: P – value was observed to be less than 0.01, thus the null hypothesis that the stochastic component of this model is normally distributed was rejected.
* **ACF and PACF plot** displayed multiple significant lags which confirmed that there was some autocorrelation left in the residuals.
* **The Ljung-box test**: Most of the points were observed to be below red dashed line, thus, the null hypothesis that the error terms were uncorrelated could be rejected.

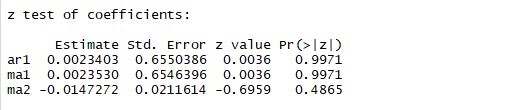
Based on the coefficient parameter estimates and the residual it was concluded that ARIMA (1,1,1) was not successfully capturing the dependence structure of Bitcoin time series.

**ARIMA (1, 1, 2)**



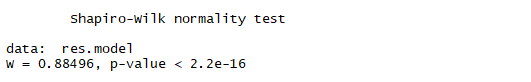
**Output:**





P value was observed to be greater than 0.05 for AR (1). MA (1) and MA (2) coefficient, which concluded that AR (1), MA (1) and MA (2) were insignificant on both ML and CSS methods.

**Residual Analysis ARIMA (1, 1, 2):**



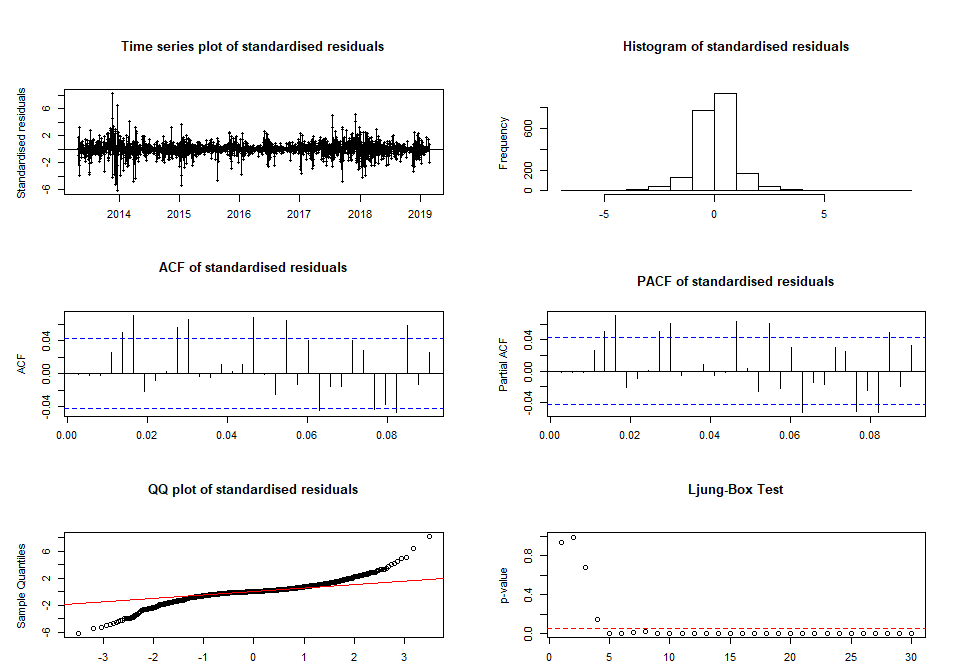


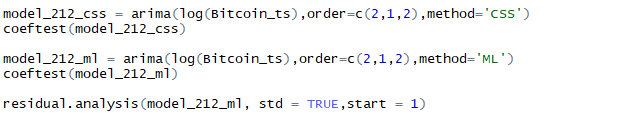
Figure 11: Residual Analysis for ARIMA (1,1,2)

From *Figure 11* we observe:

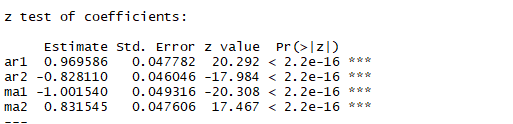
* **Residuals plot**: Although the plot showed the specified model de-trend the trend in the series, the changing variance was still very obvious in the residuals
* **Histogram** could not be viewed as symmetric
* **Q-Q plot**: Significant number of departing points from the red dash concluded that the normality assumption did not hold for this series. The thick tail implied existence of an ARCH effect in the series.
* **Shapiro-wilk test**: P – value was observed to be less than 0.01, thus the null hypothesis that the stochastic component of this model is normally distributed was rejected.
* **ACF and PACF plot** displayed multiple significant lags which confirmed that there was some autocorrelation left in the residuals.
* **The Ljung-box test**: Most of the points were observed to be below red dashed line, thus, the null hypothesis that the error terms were uncorrelated could be rejected.

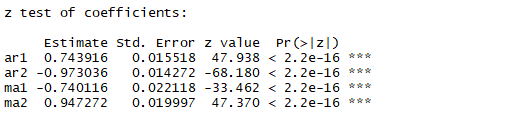
Based on the coefficient parameter estimates and the residual it was concluded that ARIMA (1,1,2) was not successfully capturing the dependence structure of Bitcoin time series.

**ARIMA (2, 1, 2)**



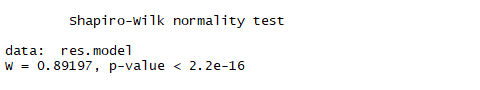
**Output**





P value was observed to be lesser than 0.05 for AR (1), AR (2). MA (1) and MA (2) coefficients, which indicated that AR (1), AR (2), MA (1) and MA (2) were significant on both ML and CSS methods.

**Residual Analysis ARIMA (2, 1, 2):**



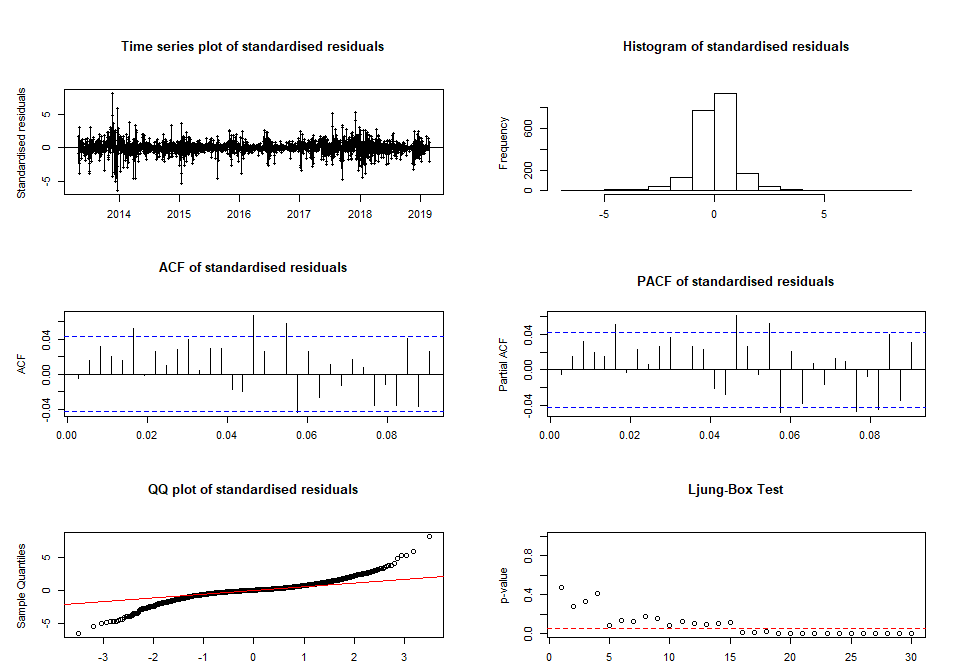


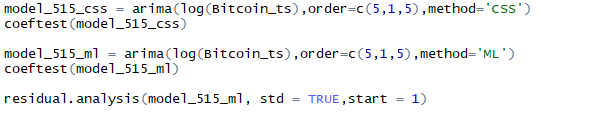
Figure 12: Residual Analysis for ARIMA (2,1,2)

From *Figure 12* we observe:

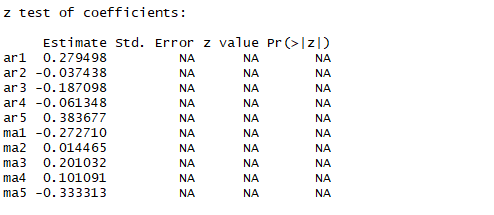
* **Residuals plot**: Although the plot showed the specified model de-trend the trend in the series, the changing variance was still very obvious in the residuals
* **Histogram** could be viewed as symmetric
* **Q-Q plot**: Significant number of departing points from the red dash concluded that the normality assumption did not hold for this series. The thick tail implied existence of an ARCH effect in the series.
* **Shapiro-wilk test**: P – value was observed to be less than 0.01, thus the null hypothesis that the stochastic component of this model is normally distributed was rejected.
* **ACF and PACF plot** displayed multiple early significant lags which confirmed that there was some autocorrelation left in the residuals.
* **The Ljung-box test**: Multiple points were observed to be below the red dashed line, thus, the null hypothesis that the error terms were uncorrelated could be rejected.

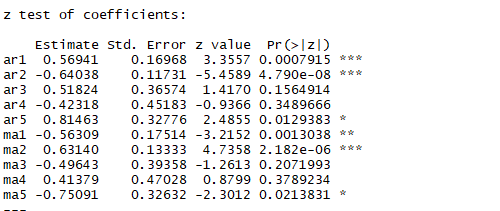
Based on the coefficient parameter estimates and the residual it was concluded that ARIMA (2,1,2) was not successfully capturing the dependence structure of Bitcoin time series.

**ARIMA (5, 1, 5)**



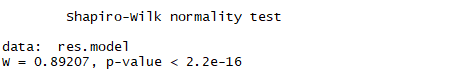
**Output:**





P value was observed to be lesser than 0.05 for AR (1), AR (2). AR (5), MA (1), MA (2) and MA (5) coefficients, which concluded that AR (1), AR (2). AR (5), MA (1), MA (2) and MA (5) were significant on ML while the remaining coefficients AR (3), AR (4), MA (3) and MA (4) were insignificant. All the parameter coefficients were observed to be NA on the CSS method.

**Residual Analysis ARIMA (5, 1, 5):**



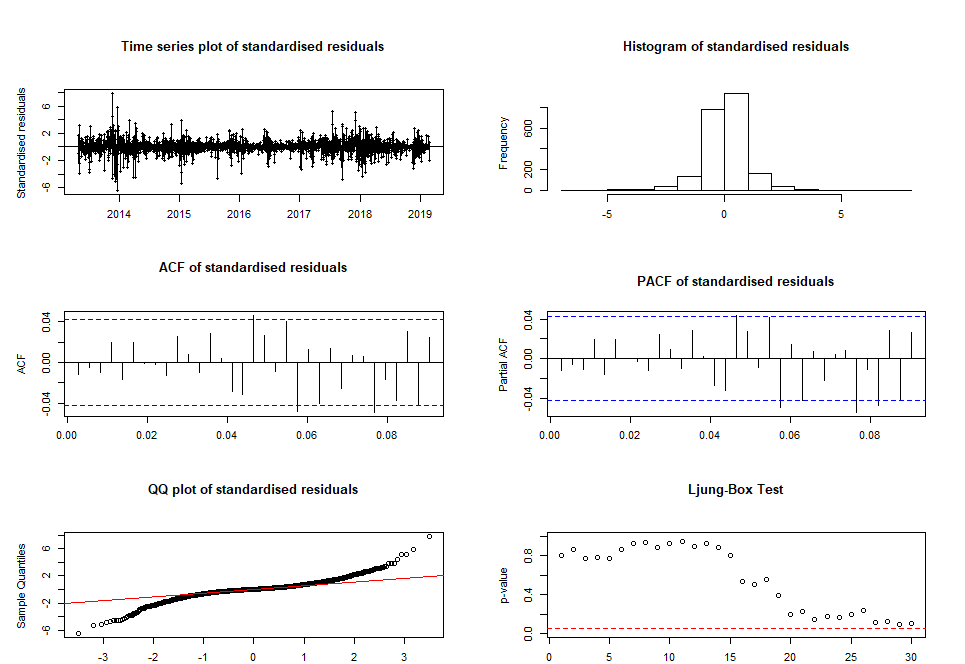


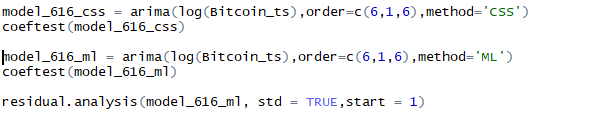
Figure 13: Residual Analysis for ARIMA (5,1,5)

From *Figure 13* we observe:

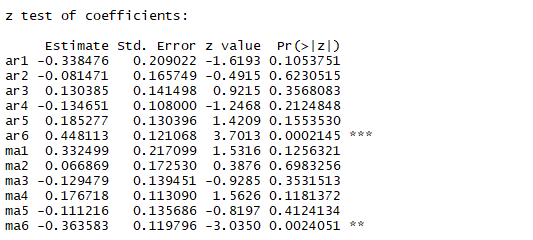
* **Residuals plot**: Although the plot showed the specified model de-trend the trend in the series, the changing variance was still very obvious in the residuals
* **Histogram** could be viewed as symmetric
* **Q-Q plot**: Significant number of departing points from the red dash concluded that the normality assumption did not hold for this series. The thick tail implied existence of an ARCH effect in the series.
* **Shapiro-wilk test**: P – value was observed to be less than 0.01, thus the null hypothesis that the stochastic component of this model is normally distributed was rejected.
* **ACF and PACF plot** displayed few significant late lags which suggested that there might be some autocorrelation left in the residuals.
* **The Ljung-box test**: All points were observed to be above the red dashed line, which suggested failure to reject the null hypothesis that the error terms were uncorrelated.

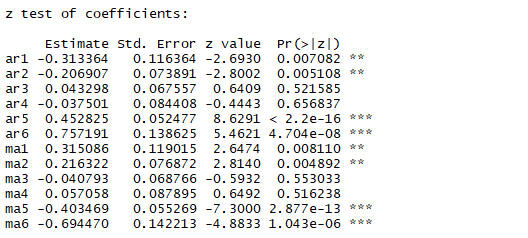
Overall based on the coefficients observed, ARIMA (5,1,5) was successful in capturing the trend in the series however from residual analysis normality didn’t hold true for ARIMA (5,1, 5). The parameter estimation and residual analysis suggested that this model could be analysed further in case a better candidate model specification was not met.

**ARIMA (6, 1, 6)**



**Output:**





P value was observed to be lesser than 0.05 for AR (1), AR (2). AR (5), AR (6), MA (1), MA (2), MA (5) and MA (6) coefficients, which meant that AR (1), AR (2). AR (5), AR (6), MA (1), MA (2), MA (5) and MA (6) were significant on ML while other coefficients AR (3), AR (4), MA (3) and MA (4) were insignificant. AR (6) and MA (6) were observed to be significant on CSS method.

**Residual Analysis ARIMA (6, 1, 6):**

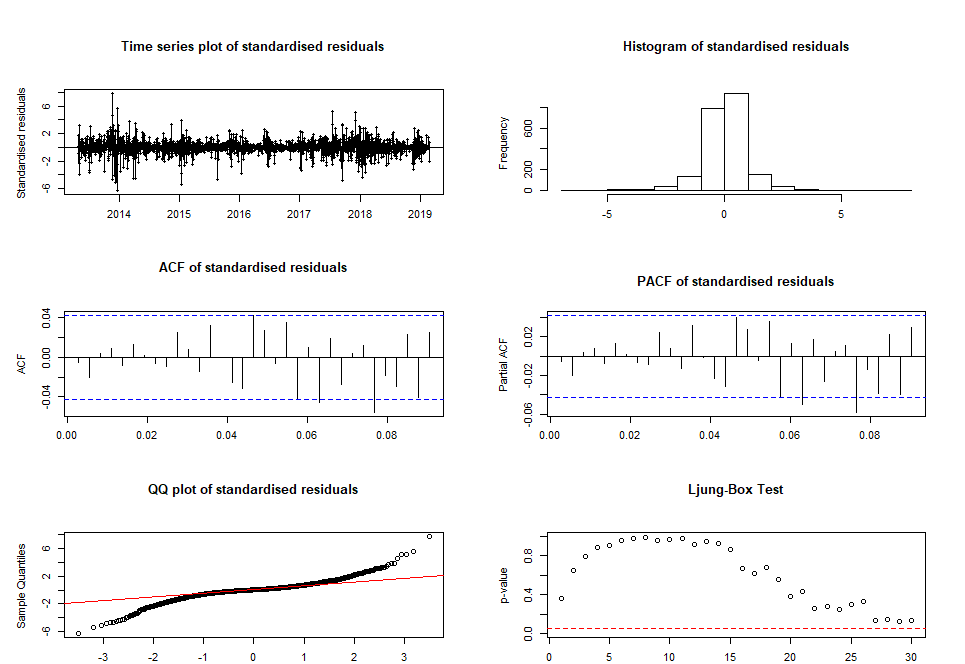
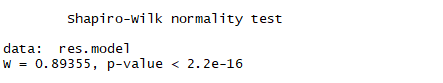


Figure 14: Residual Analysis for ARIMA (6,1,6)

From *Figure 14* we observe

* **Residuals plot**: Although the plot showed the specified model de-trend the trend in the series, the changing variance was still very obvious in the residuals
* **Histogram** could be viewed as symmetric
* **Q-Q plot**: Significant number of departing points from the red dash concluded that the normality assumption did not hold for this series. The thick tail implied existence of an ARCH effect in the series.
* **Shapiro-wilk test**: P – value was observed to be less than 0.01, thus the null hypothesis that the stochastic component of this model is normally distributed was rejected.
* **ACF and PACF plot** displayed a few significant late lags which could be attributed to the changing variance in the series.
* **The Ljung-box test**: All points were observed to be above the red dashed line, which suggested failure to reject the null hypothesis that the error terms were uncorrelated.

Overall, based on coefficient estimation and the residual analysis we observed that normality did hold true for ARIMA (6, 1, 6). Thus, the model was successful in capturing the dependence structure of Bitcoin time series.

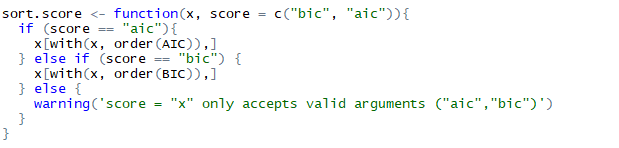
As a part of further analysis, the Akaike information criterion(“AIC”) and Bayesian information criterion (“BIC”) were checked to assess the quality of all the chosen candidate models.

**AIC and BIC values**

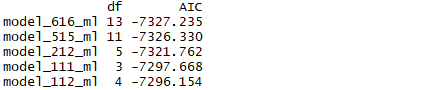
***The Akaike information criterion****is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models.*

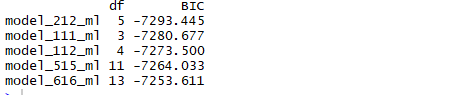
***The Bayesian information criterion****or Schwarz information criterion (also SIC, SBC, SBIC) is a criterion for model selection among a finite set of models; the model with the lowest BIC is preferred. It is based, in part, on the likelihood function and it is closely related to the Akaike information criterion (AIC).*

The sort score function was used to arrange the models based on their AIC and BIC values in order of model quality: -



**Output:**





As per AIC, the preferred model was observed to be ARIMA (6,1,6).

The BIC did not agree with the AIC on the preference of model selection which was in line with the initial expectation taking into consideration the large number of parameters in ARIMA (6,1,6) and the nature of the BIC test to penalise models with greater number of parameters.

Thus, considering the significance of the estimates obtained by the coefficient test, the behaviour of residuals observed via the residual analysis and the preference suggested by the AIC test, ARIMA (6,1,6) was selected as the final model.

## Model Fitting

In usual practice, overfitting on the final model is performed to check in case a better-quality model could be obtained which is not already a part of the candidate models.

However, in this case considering ARIMA (6,1,6) was already a large model, an attempt was made to observe the parameter estimations using under-fitting (i.e. reducing the MA and AR term by 1)

The models selected for the purpose of under-fitting were ARIMA (5,1,6) and ARIMA (6,1,5) however the coefficients in the parameter estimation and the residual suggested that they were not adequate models to represent the bitcoin time series.

# GARCH Model Selection

As observed in *Figure 1* and *Figure 6* the bitcoin time series displayed changing variance.

The close price dataset consisted of a high variation between the current and past values of the process. For example, daily returns of a stock were often observed to have larger conditional variance following a period of violent price movement than a relatively stable period. *This violates the constancy of variance of further steps ahead for ARIMA processes*.

The same was also visible when the residuals of ARIMA (6,1,6) models were plotted.



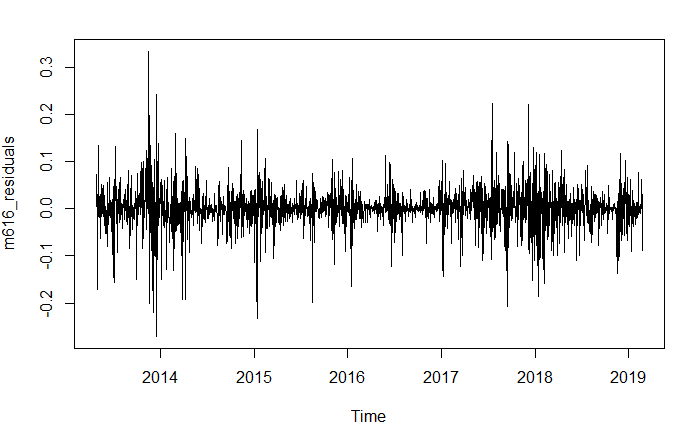


Figure 15. Time Series plot of ARIMA (6,1,6) Model

From *Figure 15*, it was observed that the returns were more volatile over some time periods and became very volatile toward the end of the study period. This indicated a sign of the changing variance effect.

*Let {pt} be the time series of, say, the daily price of some financial asset. The (continuously compounded) return on the t-th day is defined as*

rt =log(pt)-log(pt-1)

*Sometimes the returns are then multiplied by 100 so that they can be interpreted as percentage changes in the price. The multiplication may also reduce numerical errors as the raw returns could be very small numbers and render large rounding errors in some calculations.* (Demirhan, 2019)

However, the multiplication would not be required to be applied to this dataset since before applying the ARIMA models, the logarithmic transformation and differencing were applied in order to make the dataset stationary, making it a returns dataset.

*In addition to the visual tools, the Ljung-Box test can be applied to the transformed data. The test statistic of this test will be approximately chi-square distributed if there is no Autoregressive Conditional Heteroskedasticity (ARCH). This approach can be extended to the case when the conditional mean of the process is non-zero and if an ARMA model is adequate in describing the autocorrelation structure of the data. In which case, the first m autocorrelations of the squared residuals from this model can be used to test for the presence of ARCH.* (Demirhan, 2019)

We will refer to the test for ARCH effects using the Mcleod.Li test.



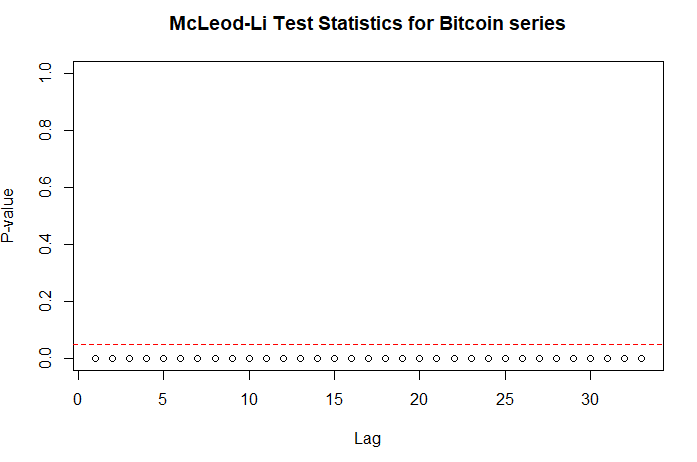


Figure 16. McLeod.Li Test Statistics Graph

As it seen in the plot(Refer *Figure 16*), the McLeod-Li tests were observed to be significant at a 5% level of significance for the all lags included in the test. This provided strong evidence for ARCH effect in this data.

In addition to the Mcleod-Li test, the Q-Q plots displayed (Refer *Figure 14*) thicker tails than that of a normal distribution.

Overall, the Bitcoin close price data was found to be serially uncorrelated but admitted a higher-order dependence structure, namely volatility clustering, and a heavy-tailed distribution. That concluded that the ARIMA model by itself was not successful in capturing every correlation in the dataset.

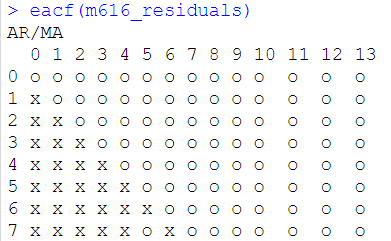


Figure 17: EACF of ARIMA (6,1,6) Residuals

The sample EACF (Refer *Figure 17*) also confirmed the existence of little serial correlation by suggesting a white noise series.

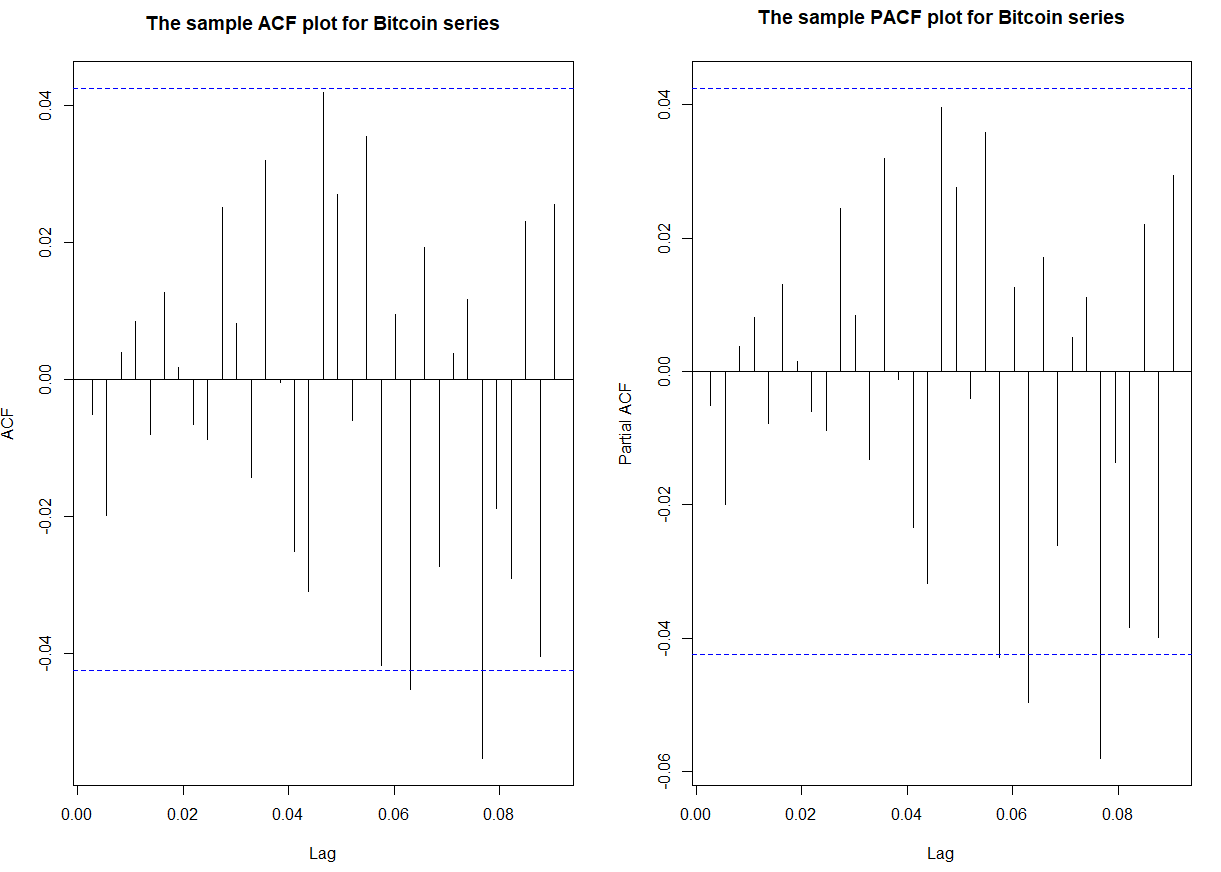


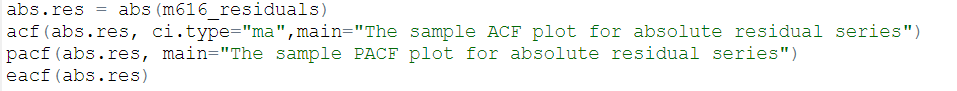
Figure 18: ACF and PACF (Residuals of ARIMA (6,1,6))

Significant late lags were observed in the ACF and PACF (Refer *Figure 18*) of the residuals of ARIMA (6,1,6) which also supported the changing variance effect.

The problem here was that the data was not independently and identically distributed due to the significant volatility.

*If the series values were truly independent, then nonlinear transformations such as log, absolute value, or square root preserve independence. However, the same is not true of correlation, since correlation is only a measure of linear dependence. Thus, we can detect violations of iid assumption over the ACF or PACF of transformed series.* (Demirhan, 2019)

The following plots(Refer *Figure 19* and *Figure 21*) show the sample ACF and the sample PACF of absolute and squared returns.



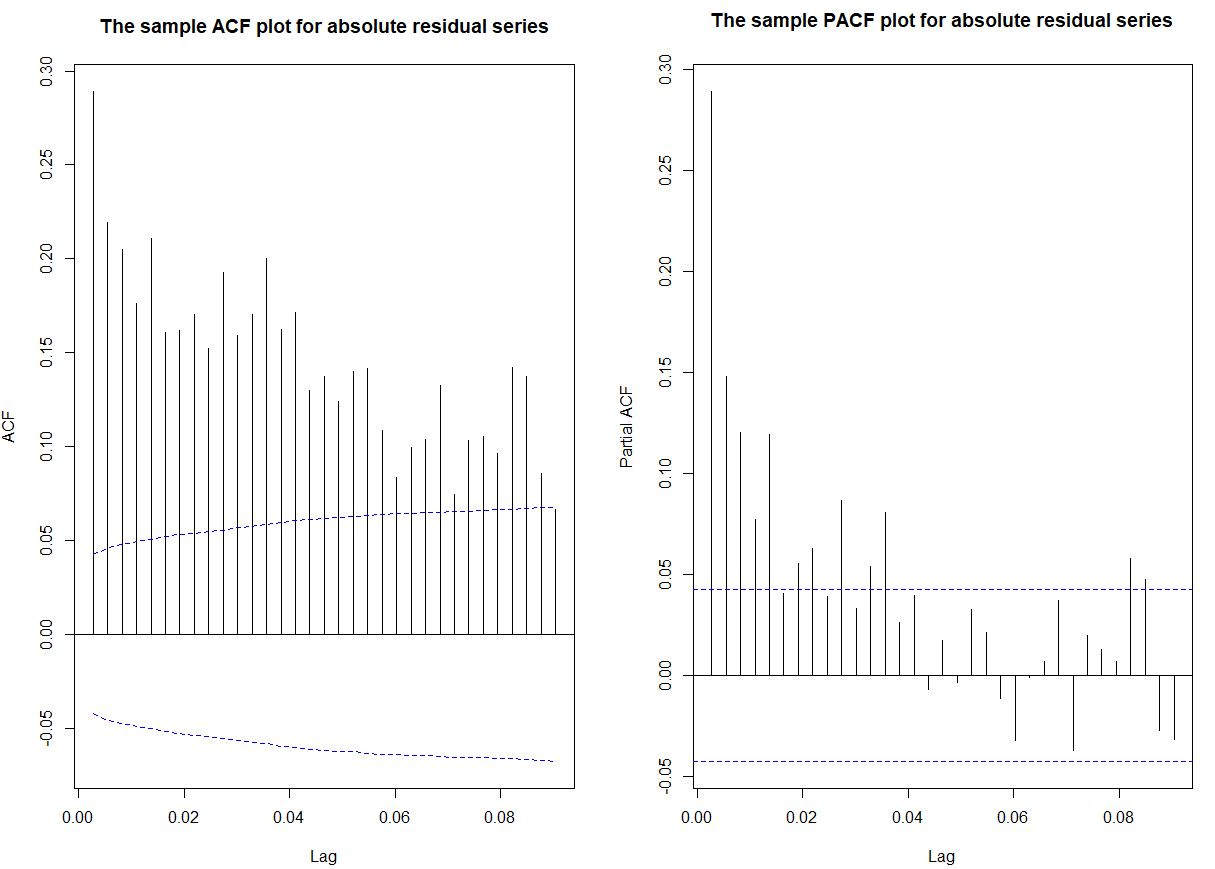


Figure 19: ACF and PACF of absolute residuals of ARIMA (6,1,6)

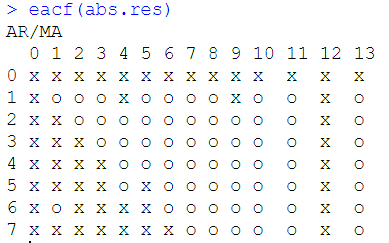
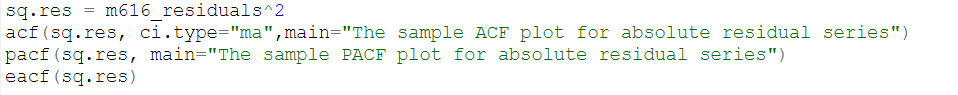


Figure 20: EACF of absolute residuals of ARIMA (6,1,6)

From the EACF of absolute residuals (Refer *Figure 20*), ARMA (1,1), ARMA (1,2), ARMA (2,2) models were identified. These models corresponded to parameter settings of [max (1,1),1], [max (1,2),1], and [max (2,2),2]. The tentative garch models selected were:

*GARCH (1,1), GARCH (2,1), GARCH (2,2).*



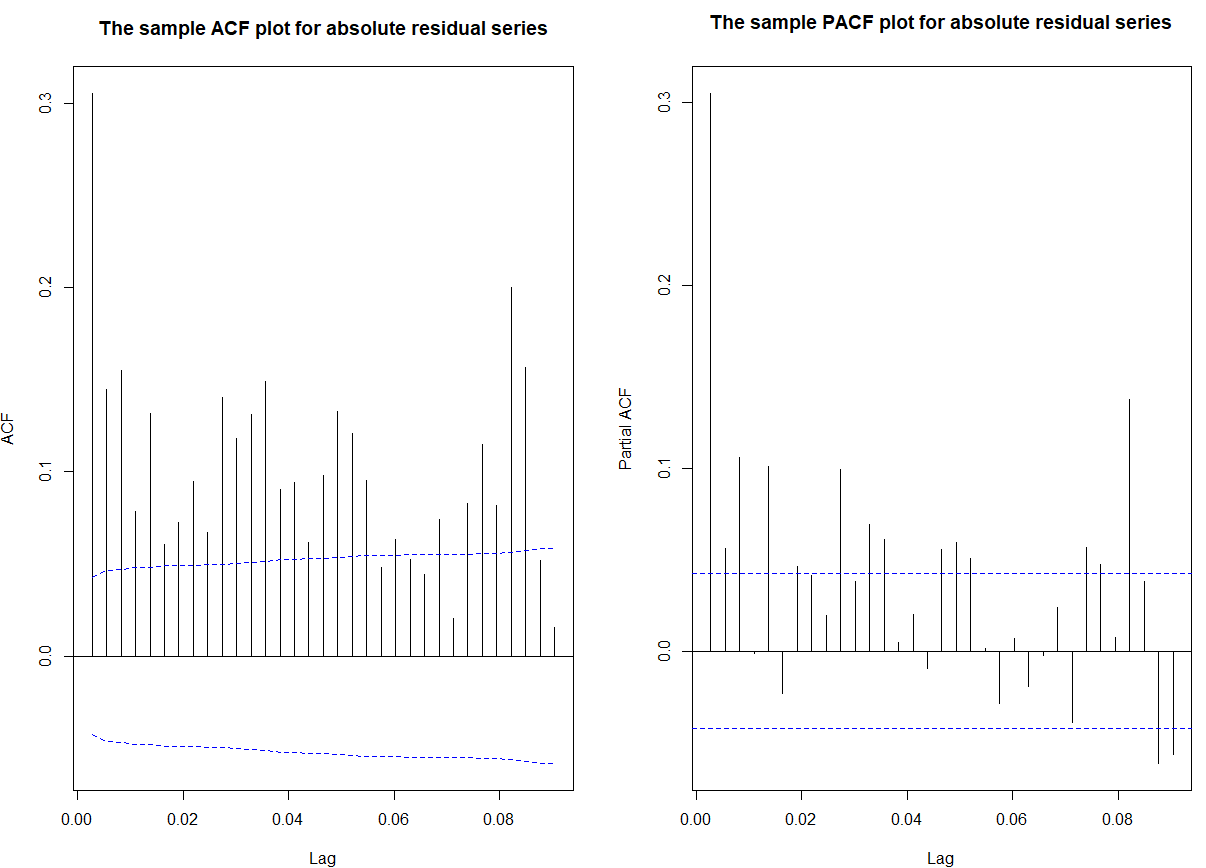


Figure 21: ACF and PACF of squared residuals of ARIMA (6,1,6)

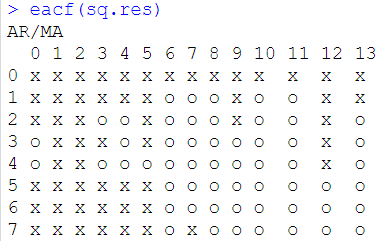


Figure 22: EACF of squared residuals of ARIMA (6,1,6)

From the EACF of squared residuals (Refer *Figure 22*), ARMA (2,3), ARMA (2,4), ARMA (3,4) models were identified. These models corresponded to parameter settings of [max (2,3),2], [max (2,4),2], and [max (3,4),3]. The tentative garch models selected were

*GARCH (3,2), GARCH (4,2), GARCH (4,3)*

After narrowing down on the candidate garch models, each model was plotted and checked for parameter significance.

It was observed that GARCH (1,1) and GARCH (2,1) were the models with most of the parameters displaying significant estimates.

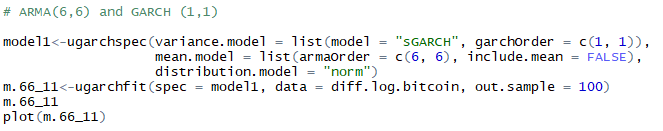
However, on plotting the residual analysis of the two GARCH models, it could be observed that the QQ-plot and ACF of squared standardized residuals of both the models were almost the same, however the ACF of standardized residuals plot of GARCH (2,1) was far better than GARCH (1,1) as there were no significant values in it, while in GARCH (1,1) there was one clear significant value and 3 values almost touching the line of significance. (Refer *Figure 23 and Figure 24)*

In GARCH (1,1) all the values were significant with the Akaike and Bayesian value of -3.7618 and -3.7203 respectively and all the other tests displayed good results.

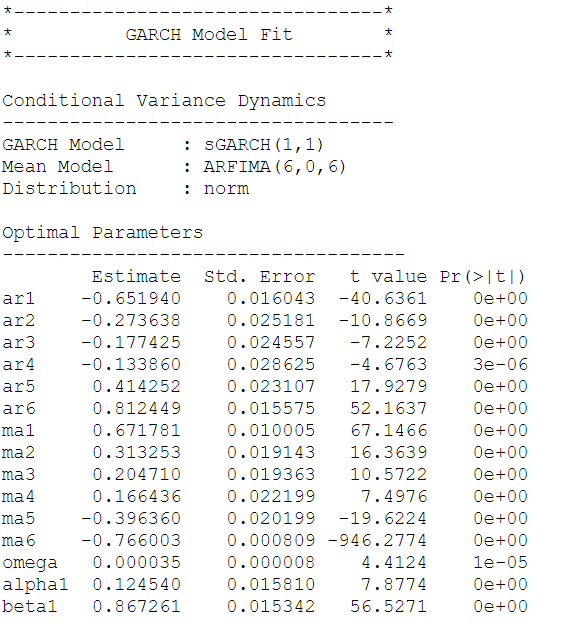
While in GARCH (2,1) it was observed that the values of ma2 and alpha2 were not significant and the Akaike and Bayes values were -3.7582 and -3.7139 respectively and all the other tests displayed good results as well.

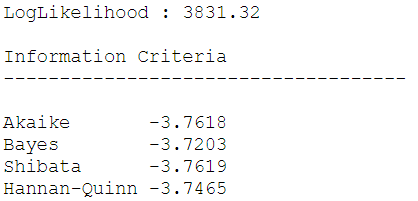
So, after carefully considering all the aspects it was finalized that the model GARCH (2,1) was the best model to capture the effect of the changing variance of the bitcoin time series. The values of the tests of both the models were almost similar however it was concluded based on the plots of residual analysis (ACF of standardized residuals).

**GARCH (1,1)**



**Output:**





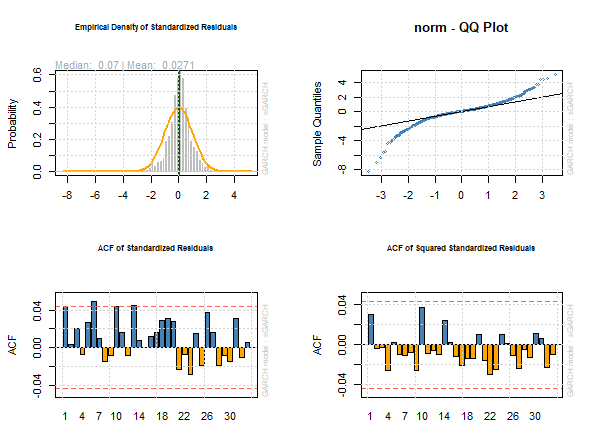
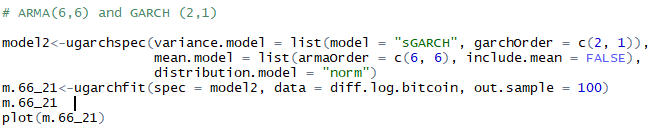
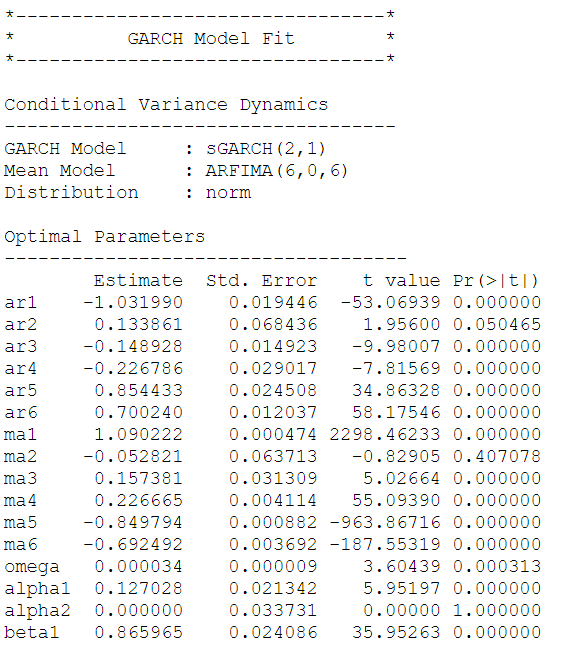


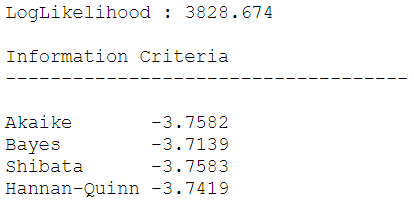
Figure 23: Standardized residuals of GARCH (1,1)

**GARCH (2,1)**



**OUTPUT:**





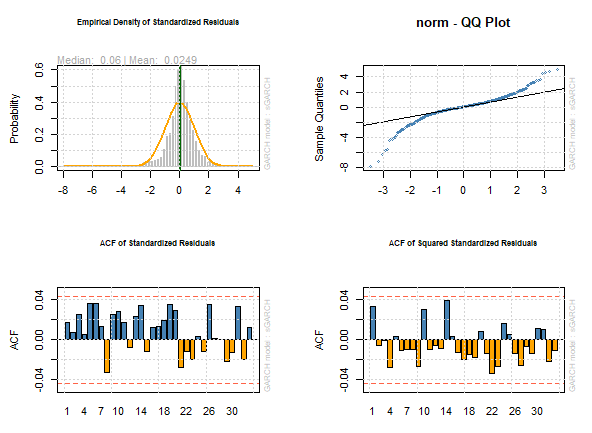


Figure 24: Standardized residuals of GARCH (2,1)

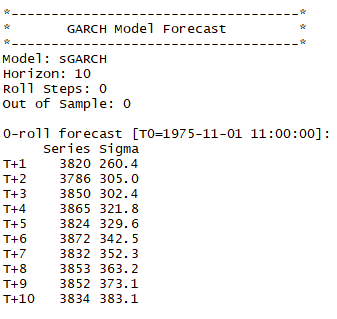
# Forecasting

Forecasts for the next 10 daily values were generated using a combination of ARIMA (6,1,6) and GARCH (2,1).

For the purpose of forecasting the whole series (logarithmic differenced) was reversed using difference and log inversing.

Forecasting was performed using the *ugarch* function taking ARIMA (6,6) and GARCH (2,1) on the first differenced dataset. It was observed that the values of the sigma were in the range of 260 to 383 but did not have any effect on the forecasts and the forecasted valued obtained were quite similar to the observed values.

The values obtained for the next 10 days have been displayed below:



The values were plotted to show a visual representation of the bitcoin forecasts. (Refer *Figure 25*)

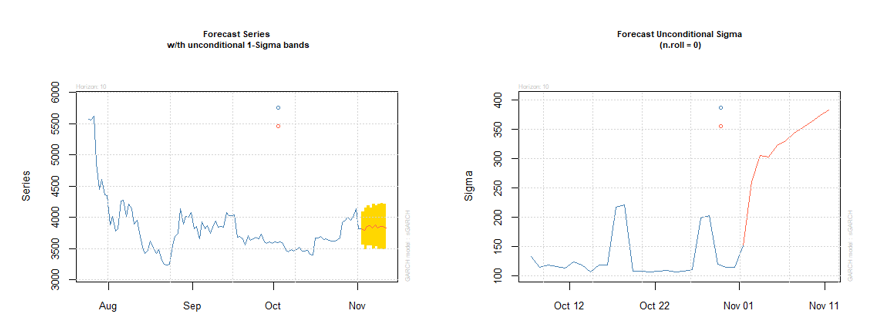


Figure 25: Forecast plots

# MASE

The Mean Absolute Scaled Error (“MASE”) statistic was used to check the deviation between the observed values and the forecasted value.

Two different MASE values were computed to check the following:

1. **The fit of the model on the bitcoin time series**
2. **The accuracy of the model in terms of prediction power**

## Between the observed and the predicted value

The MASE computed between the actual observed values and the predicted value was computed to be 1.26. A low MASE value signified high prediction power of the selected model.





## Between the observed and the fitted model

The MASE computed between the observed and fitted model was observed to 1.0002. A low MASE in this case signified that the model selection helped capture most of the data in the bitcoin time series.



# Conclusion

As a part of this report, after achieving stationarity of the time series using log differencing the candidate models were chosen and assessed based on the diagnostic checking and parameter estimations.

The changing variance of the time series was observed and handled using GARCH estimation techniques.

After considering all aspects of the model and residuals it was concluded that ARIMA (6,1,6) + GARCH (2, 1) was the best fit model for the daily closing price of bitcoin time series from 27th April 2013 to 24th Feb 2019.

The low MASE values observed between the predictions and true observed values as well as the low MASE between the observed and fitted model instilled confidence that the model was highly accurate in terms of data capture and forecasting power.

|  |  |
| --- | --- |
| **Component** | **MASE** |
| *Observed vs Predictions* | **1.261873** |
| *Observed vs Fitted model* | **1.0002** |

# References

Dr Haydar Demirhan (2019). Time Series Analysis MATH 1318. RMIT UNIVERSITY Melbourne.

# Appendix

*# Packages List*

*library(readr)*

*library(fUnitRoots)*

*library(tseries)*

*library(lmtest)*

*library(TSA)*

*library(forecast)*

*library(CombMSC)*

*library(fGarch)*

*library(rugarch)*

*# Functions List*

*sort.score <- function(x, score = c("bic", "aic")){*

*if (score == "aic"){*

*x[with(x, order(AIC)),]*

*} else if (score == "bic") {*

*x[with(x, order(BIC)),]*

*} else {*

*warning('score = "x" only accepts valid arguments ("aic","bic")')*

*}*

*}*

*residual.analysis <- function(model, std = TRUE,start = 2, class = c("ARIMA","GARCH","ARMA-GARCH")[1]){*

*# If you have an output from arima() function use class = "ARIMA"*

*# If you have an output from garch() function use class = "GARCH"*

*# If you have an output from ugarchfit() function use class = "ARMA-GARCH"*

*library(TSA)*

*library(FitAR)*

*if (class == "ARIMA"){*

*if (std == TRUE){*

*res.model = rstandard(model)*

*}else{*

*res.model = residuals(model)*

*}*

*}else if (class == "GARCH"){*

*res.model = model$residuals[start:model$n.used]*

*}else if (class == "ARMA-GARCH"){*

*res.model = model@fit$residuals*

*}else {*

*stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")*

*}*

*par(mfrow=c(3,2))*

*plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of standardised residuals")*

*abline(h=0)*

*hist(res.model,main="Histogram of standardised residuals")*

*acf(res.model,main="ACF of standardised residuals")*

*pacf(res.model,main="PACF of standardised residuals")*

*qqnorm(res.model,main="QQ plot of standardised residuals")*

*qqline(res.model, col = 2)*

*print(shapiro.test(res.model))*

*k=0*

*LBQPlot(res.model, lag.max = 30, StartLag = k + 1, k = 0, SquaredQ = FALSE)*

*}*

*MASE = function(observed , fitted ){*

*# observed: Observed series on the forecast period*

*# fitted: Forecast values by your model*

*Y.t = observed*

*n = length(fitted)*

*e.t = Y.t - fitted*

*sum = 0*

*for (i in 2:n){*

*sum = sum + abs(Y.t[i] - Y.t[i-1] )*

*}*

*q.t = e.t / (sum/(n-1))*

*MASE = data.frame( MASE = mean(abs(q.t)))*

*return(list(MASE = MASE))*

*}*

*# Reading the dataset*

*Bitcoin<- read\_csv("C:/Users/Syed Hassan Afsar/Downloads/RMIT/2nd Semester/Time Series Analysis/Final Project/Bitcoin\_Historical\_Price.csv")*

*index <- seq(as.Date("2013-04-27"), as.Date("2019-02-24"), by="day")*

*# Convertiong the dataset to timeseries object*

*Bitcoin\_ts <- ts(Bitcoin$Close, start = c(2013, as.numeric(format(index[1], "%j"))), frequency = 365)*

*class(Bitcoin\_ts)*

*# Checking for missing values and imputing*

*which(is.na(Bitcoin\_ts))*

*Bitcoin\_ts[1773] <- 11573.30*

*Bitcoin\_ts[1774] <- 10779.90*

*Bitcoin\_ts[1775] <- 9965.57*

*Bitcoin\_ts[1776] <- 9395.01*

*Bitcoin\_ts[1777] <- 9337.55*

*Bitcoin\_ts[1778] <- 8866.00*

*Bitcoin\_ts[1779] <- 9578.63*

*Bitcoin\_ts[1780] <- 9205.12*

*Bitcoin\_ts[1781] <- 9194.85*

*Bitcoin\_ts[1782] <- 8269.81*

*which(is.na(Bitcoin\_ts))*

*# Visualisation using ACF, PACF plot*

*qqnorm(Bitcoin\_ts,main="Q-Q Normal Plot of Bitcoin series")*

*qqline(Bitcoin\_ts) # Fat tails is in accordance with volatiliy clustering*

*par(mfrow=c(1,2))*

*acf(Bitcoin\_ts, main="The sample ACF plot for Bitcoin series")*

*pacf(Bitcoin\_ts, main="The sample PACF plot for Bitcoin series")*

*eacf(Bitcoin\_ts)*

*# Checking for ARCH effect*

*McLeod.Li.test(y=Bitcoin\_ts,main="McLeod-Li Test Statistics for Bitcoin series")*

*# Transformations*

*BC.Bitcoin <- BoxCox.ar(Bitcoin\_ts, method="yule-walker")*

*# 0*

*# log transformation*

*par(mfrow=c(1,2))*

*acf(log(Bitcoin\_ts))*

*pacf(log(Bitcoin\_ts))*

*# stil there is a trend / decaying pattern*

*# high first lag in the pacf*

*qqnorm(log(Bitcoin\_ts), main = "QQplot for natural log of bitcoin series")*

*qqline(log(Bitcoin\_ts))*

*adf.test(log(Bitcoin\_ts))*

*# 0.7348*

*# Still not stationary , hence we go for differencing*

*diff.log.bitcoin = diff(log(Bitcoin\_ts),difference = 1)*

*plot(diff.log.bitcoin, type = 'o', ylab = 'Closing price')*

*order = ar(diff(diff.log.bitcoin))$order*

*adfTest(diff.log.bitcoin,lags = order, title = NULL, description = NULL) #Stationary with lag order 32*

*McLeod.Li.test(y=diff.log.bitcoin, main = "Mcleod-Li test for checking changing variance on bitcoin data")*

*# ARIMA Model Selection*

*par(mfrow=c(1,2))*

*acf(diff.log.bitcoin)*

*pacf(diff.log.bitcoin)*

*# arch effect visible in the both acf and pacf plots*

*eacf(diff.log.bitcoin)*

*# ARIMA(2,1,2), ARIMA(1,1,2), ARIMA(1,1,1)*

*res = armasubsets(y=diff.log.bitcoin,nar=10,nma=10,y.name='test',ar.method='ols')*

*plot(res)*

*# ARIMA(5,1,5),ARIMA(6,1,6)*

*# Overall; ARIMA(2,1,2), ARIMA(1,1,2), ARIMA(1,1,1), ARIMA(5,1,5),ARIMA(6,1,6)*

*# ARIMA(1,1,1) # not significant*

*model\_111\_css = arima(log(Bitcoin\_ts),order=c(1,1,1),method='CSS')*

*coeftest(model\_111\_css)*

*model\_111\_ml = arima(log(Bitcoin\_ts),order=c(1,1,1),method='ML')*

*coeftest(model\_111\_ml)*

*residual.analysis(model\_111\_ml, std = TRUE,start = 1)*

*# ARIMA(1,1,2) # Not significant*

*model\_112\_css = arima(log(Bitcoin\_ts),order=c(1,1,2),method='CSS')*

*coeftest(model\_112\_css)*

*model\_112\_ml = arima(log(Bitcoin\_ts),order=c(1,1,2),method='ML')*

*coeftest(model\_112\_ml)*

*residual.analysis(model\_211\_ml, std = TRUE,start = 1)*

*# ARIMA(2,1,2) #All significant*

*model\_212\_css = arima(log(Bitcoin\_ts),order=c(2,1,2),method='CSS')*

*coeftest(model\_212\_css)*

*model\_212\_ml = arima(log(Bitcoin\_ts),order=c(2,1,2),method='ML')*

*coeftest(model\_212\_ml)*

*residual.analysis(model\_212\_ml, std = TRUE,start = 1)*

*# ARIMA(5,1,5) # not significant*

*model\_515\_css = arima(log(Bitcoin\_ts),order=c(5,1,5),method='CSS')*

*coeftest(model\_515\_css)*

*model\_515\_ml = arima(log(Bitcoin\_ts),order=c(5,1,5),method='ML')*

*coeftest(model\_515\_ml)*

*residual.analysis(model\_015\_ml, std = TRUE,start = 1)*

*# ARIMA(6,1,6) #not significant*

*model\_616\_css = arima(log(Bitcoin\_ts),order=c(6,1,6),method='CSS')*

*coeftest(model\_616\_css)*

*model\_616\_ml = arima(log(Bitcoin\_ts),order=c(6,1,6),method='ML')*

*coeftest(model\_616\_ml)*

*residual.analysis(model\_616\_ml, std = TRUE,start = 1)*

*# Comparing the AIC and BIC results*

*sort.score(AIC(model\_111\_ml,model\_112\_ml,model\_212\_ml,model\_515\_ml, model\_616\_ml), score="aic")*

*sort.score(BIC(model\_111\_ml,model\_112\_ml,model\_212\_ml,model\_515\_ml, model\_616\_ml), score="bic")*

*#Underfitting check of ARIMA(6,1,6)*

*# ARIMA(5,1,6)*

*model\_516\_css = arima(log(Bitcoin\_ts),order=c(5,1,6),method='CSS')*

*coeftest(model\_516\_css)*

*model\_516\_ml = arima(log(Bitcoin\_ts),order=c(5,1,6),method='ML')*

*coeftest(model\_516\_ml)*

*residual.analysis(model\_516\_ml, std = TRUE,start = 1)*

*# ARIMA(6,1,5)*

*model\_615\_css = arima(log(Bitcoin\_ts),order=c(6,1,5),method='CSS')*

*coeftest(model\_615\_css)*

*model\_615\_ml = arima(log(Bitcoin\_ts),order=c(6,1,5),method='ML')*

*coeftest(model\_615\_ml)*

*residual.analysis(model\_615\_ml, std = TRUE,start = 1)*

*# Residual Analysis of ARIMA(6,1,6)*

*m616\_residuals = model\_616\_ml$residuals*

*# Absolute Value and Square Root Transformations*

*abs.res = abs(m616\_residuals)*

*sq.res = m616\_residuals^2*

*# Absolute Value Part*

*acf(abs.res, ci.type="ma",main="The sample ACF plot for absolute residual series")*

*pacf(abs.res, main="The sample PACF plot for absolute residual series")*

*eacf(abs.res)*

*#Square Root Part*

*acf(sq.res, ci.type="ma",main="The sample ACF plot for absolute residual series")*

*pacf(sq.res, main="The sample PACF plot for absolute residual series")*

*eacf(sq.res)*

*# From the EACF of absolute residuals, we can identify ARMA(1,1),ARMA(1,2),ARMA(2,2) models for absolute residual series.*

*# These models correspond to parameter settings of [max(1,1),1], [max(1,2),1], and [max(2,2),2]. So the corresponding*

*# tentative GARCH models are GARCH(1,1),GARCH(2,1),GARCH(2,2).*

*# From the EACF of squared residuals, we can identify ARMA(2,3),ARMA(2,4),ARMA(3,4) models for absolute residual series.*

*# These models correspond to parameter settings of [max(2,3),2], [max(2,4),2], and [max(3,4),3]. So the corresponding*

*# tentative GARCH models are GARCH(3,2),GARCH(4,2), GARCH(4,3).*

*# ARMA + GARCH*

*# ARMA(6,6) and GARCH (1,1)*

*model1<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),*

*mean.model = list(armaOrder = c(6, 6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_11<-ugarchfit(spec = model1, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_11*

*plot(m.66\_11)*

*# ARMA(6,6) and GARCH (2,1)*

*model2<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 1)),*

*mean.model = list(armaOrder = c(6, 6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_21<-ugarchfit(spec = model2, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_21*

*plot(m.66\_21)*

*# ARMA(6,6) and GARCH (2,2)*

*model3<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(2, 2)),*

*mean.model = list(armaOrder = c(6, 6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_22<-ugarchfit(spec = model3, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_22*

*plot(m.66\_22)*

*# ARMA(6,6) and GARCH (4,3)*

*model4<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(4, 3)),*

*mean.model = list(armaOrder = c(6, 6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_43<-ugarchfit(spec = model4, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_43*

*plot(m.66\_43)*

*# ARMA(6,6) and GARCH (4,2)*

*model5<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(4, 2)),*

*mean.model = list(armaOrder = c(6,6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_42<-ugarchfit(spec = model5, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_42*

*plot(m.66\_42)*

*# ARMA(6,6) and GARCH (4,3)*

*model5<-ugarchspec(variance.model = list(model = "sGARCH", garchOrder = c(4, 3)),*

*mean.model = list(armaOrder = c(6,6), include.mean = FALSE),*

*distribution.model = "norm")*

*m.66\_43<-ugarchfit(spec = model5, data = diff.log.bitcoin, out.sample = 100)*

*m.66\_43*

*plot(m.66\_43)*

*# Forecasting*

*forc.66\_21 = ugarchforecast(m.66\_21, data = diff.log.bitcoin, n.ahead = 10, n.roll = 10)*

*plot(forc.66\_21, which = "all")*

*forc.66\_21*

*# Reversing data to original*

*log.data.diff1.back = diffinv(diff.log.bitcoin, xi = log(Bitcoin\_ts)[1])*

*log.data.diff1.back = exp(log.data.diff1.back)*

*spec <- ugarchspec(variance.model = list(model ="sGARCH", garchOrder = c(2,1), submodel =NULL,external.regressors =NULL, variance.targeting =FALSE),mean.model= list(armaOrder = c(6,6),external.regressors =NULL,distribution.model ="std",start.pars = list(),fixed.pars = list()))*

*model.AR\_GARCH <- ugarchfit(spec = spec, data = Bitcoin\_ts,solver.control = list(trace=0))*

*model.AR\_GARCH*

*forc <- ugarchforecast(model.AR\_GARCH,n.ahead=10,data=Bitcoin\_ts)*

*plot(forc)*

*forc*

*a = forc@forecast$seriesFor[,ncol(forc@forecast$seriesFor)]*

*a*

*# MASE For Model Fitting*

*which(is.na(Bitcoin$Close))*

*Bitcoin$Close[1773] <- 11573.30*

*Bitcoin$Close[1774] <- 10779.90*

*Bitcoin$Close[1775] <- 9965.57*

*Bitcoin$Close[1776] <- 9395.01*

*Bitcoin$Close[1777] <- 9337.55*

*Bitcoin$Close[1778] <- 8866.00*

*Bitcoin$Close[1779] <- 9578.63*

*Bitcoin$Close[1780] <- 9205.12*

*Bitcoin$Close[1781] <- 9194.85*

*Bitcoin$Close[1782] <- 8269.81*

*which(is.na(Bitcoin$Close))*

*model\_616\_ml <- ugarchspec()*

*m.fit <- ugarchfit(spec = model\_616\_ml, data = Bitcoin$Close)*

*fitted.values = fitted(m.fit)*

*MASE(Bitcoin$Close,fitted.values)*

*# MASE For Forecast*

*Bitcoin\_Prices\_Forecasts <- read\_excel("C:/Users/Syed Hassan Afsar/Downloads/RMIT/2nd Semester/Time Series Analysis/Final Project/Bitcoin\_Prices\_Forecasts.xlsx")*

*MASE(Bitcoin\_Prices\_Forecasts$`Closing price`,as.numeric(a))*