TIME SERIES ANALYSIS FINAL PROJECT

Predicting the value of bitcoin

MATH 1318: Time Series Analysis Project

- Ahmad Hasnain: s3712538
- Simarpreet Luthra: s3706588
- Pooja Dandge: s3698457
- Wei-Ting Lin (Ivy): s3698773

Table of Contents

Introduction

This report focuses on analysing the historical price of Bitcoin and forecasting the value. The data set was sourced from CoinMarketCap (CoinMarketCap 2019). It consists of the daily closing price of bitcoin from the 27th of April 2013 to the 24th of February 2019.The objective is to find the best model among a set of candidate models and forecast the value of bitcoin for the next 10 days. The report is composed of 6 parts: descriptive analysis, overcoming the nonstationary nature, model specification and parameter estimation, diagnostic checking and finally forecasting and the conclusion.

Part 1 – Descriptive Analysis

1.1 Time series plot

Based on the time series plot (Figure 1), there is obvious trend and latter part of the series has changing variance. Seasonality is not very obvious, while intervention point could exist. The succeeding observations imply the existence of autoregressive behavior.

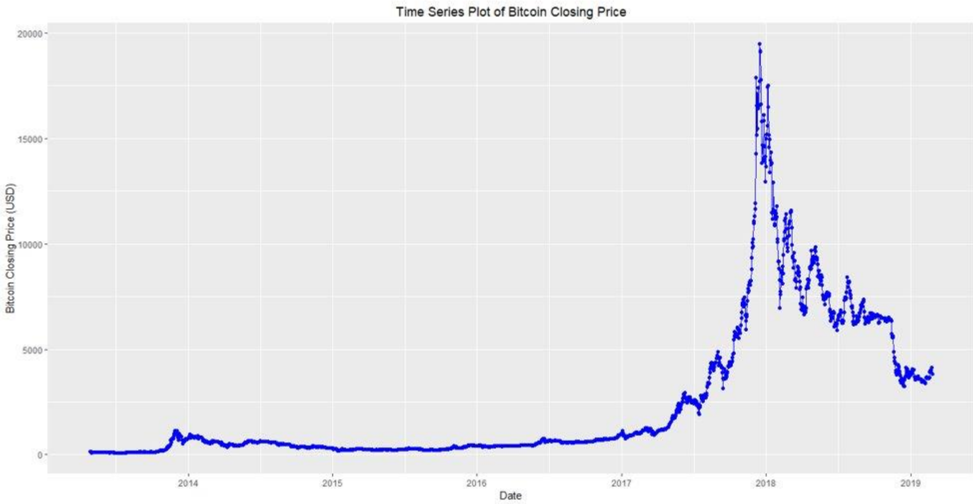


Figure 1 Time Series Plot of Bitcoin

1.2 Test stationarity and the existence of a trend in this series

From ACF and PACF plot (Figure 2), we observed slowly decaying significant lags in ACF and very high first correlation in PACF which imply the existence of trend and non-stationarity. Also, multiple significant PACF lags after a few insignificant lags prove the presence of changing variance.
After applying ADF test (Figure 3), we conclude that with a p-value of 0.23, the series is non-stationary.

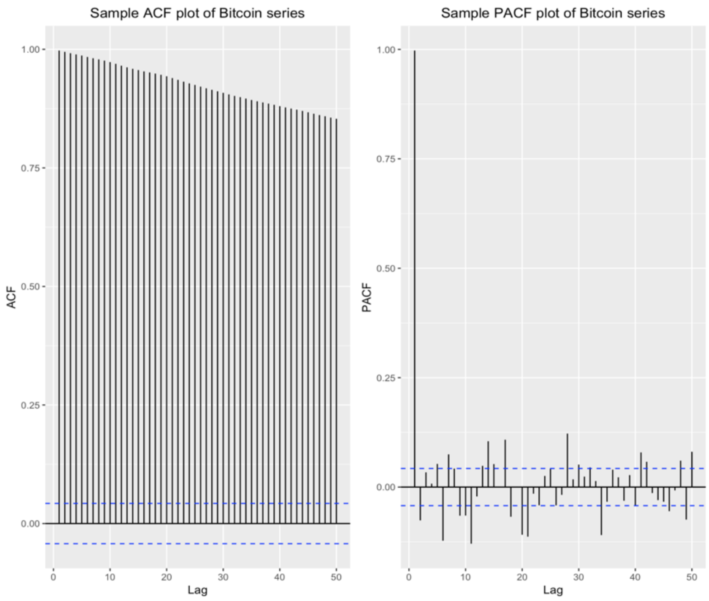
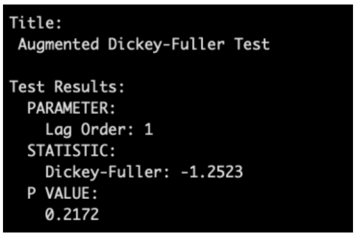


Figure 2 ACF & PACF plot



```
Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    Dickey-Fuller: -1.2523
  P VALUE:
    0.2172
```

Figure 3 ADF test

1.3 Checking for normality assumption

According to the histogram (Figure 4), the distribution is extremely right-skewed. Also, Q-Q plot shows strong deviations from normality line. After applying Shapiro-Wilk test for normality (Figure 5), We have significant evidence to reject the normality assumption with p-value < 2.2e-16.

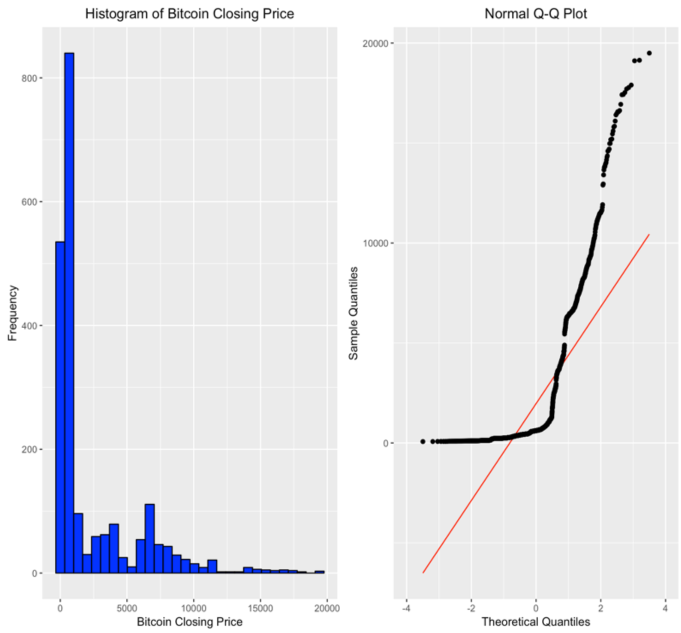


Figure 4. Normal Q-Q plot

```
        Shapiro-Wilk normality test

data:  bitcoin_ts
W = 0.68136, p-value < 2.2e-16
```

Figure 5. Shapiro-Wilk normality test

Part 2- Overcome the Non-stationarity of the Series

2.1 Apply Logarithmic Transformation

Firstly, we apply Box-Cox transformation. According to Figure 6, the 95% confidence interval for λ contains the value of λ=0, which is quite near its centre and strongly suggests a logarithmic transformation (λ=0) for the data.Taking the log of the series stabilizes the variance a bit, but there is still trend in the data as confirmed by the Dickey-Fuller unit-root test with p-value = 0.74 (Figure 7 & 8).

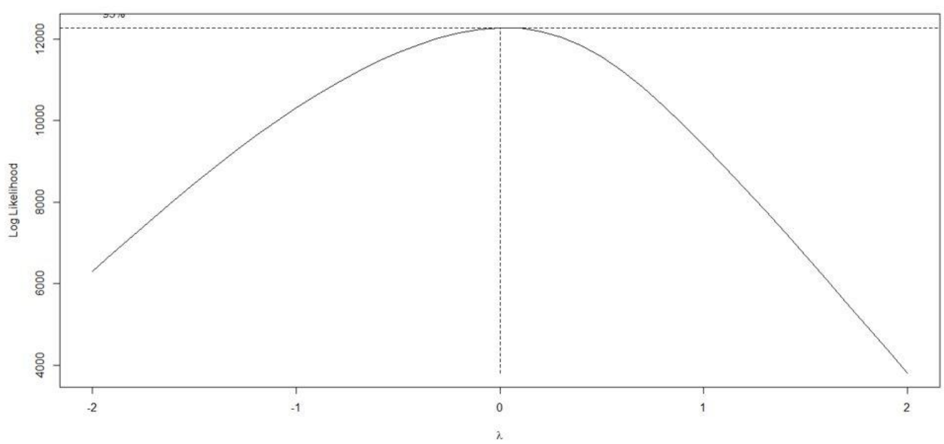


Figure 6

```
        Augmented Dickey-Fuller Test

data:  log(Bitcoin)
Dickey-Fuller = -1.6272, Lag order = 12, p-value = 0.7362
alternative hypothesis: stationary
```

Figure 7

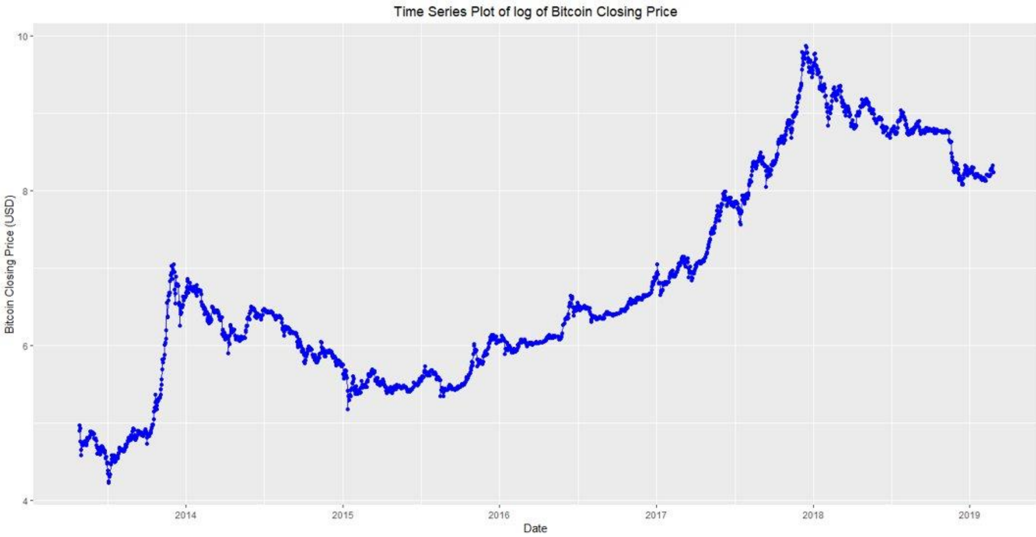


Figure 8 Time Series Plot: log of Bitcoin price

2.2 Taking First Difference

Taking the first difference of the log of the series, removes the trend and makes the series stationary as shows in Figure 9. The stationary pattern is also confirmed by the Dickey-Fuller unit-root test (Figure 10).

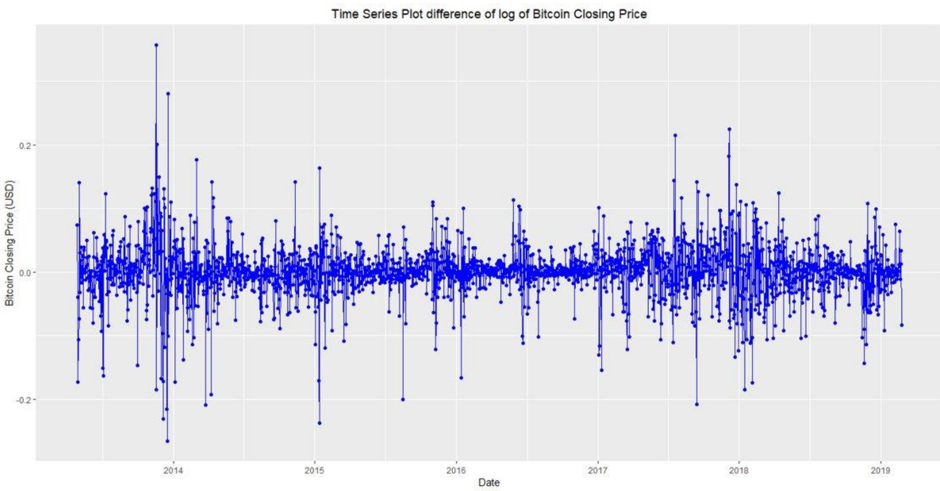


Figure 9 Time Series plot of Bitcoin after log and first differencing

```
	Augmented Dickey-Fuller Test

data:  diff.logBC
Dickey-Fuller = -11.171, Lag order = 12, p-value = 0.01
alternative hypothesis: stationary
```

Figure 10

Part 3 – Model Specification

According to the sample ACF and PACF plot (Figure 11), we can take q = 2 or 4 and p = 2 or 4. From EACF table in Figure 12, we see the vertex is at (0,0). It can be inferred from the table that p=0 or 1 and q= 1 or 2. We observe the possible set of models from EACF table are: ARIMA (0,1,1), ARIMA (1,1,1), ARIMA (0,1,2) and (1,1,2). Also, from BIC table (Figure 13), The smallest BIC contains lag 6 of the time series and lag 6 of the error process. Looking at set of possible models from the BIC table we infer ARIMA (2,1,2), ARIMA (4,1,4), ARIMA (5,1,5), ARIMA (6,1,6).

The final set of possible models is: {ARIMA (0,1,1), ARIMA (1,1,1), ARIMA (0,1,2), ARIMA (1,1,2), ARIMA (2,1,2), ARIMA (4,1,4), ARIMA (5,1,5), ARIMA (6,1,5), ARIMA (6,1,6) }

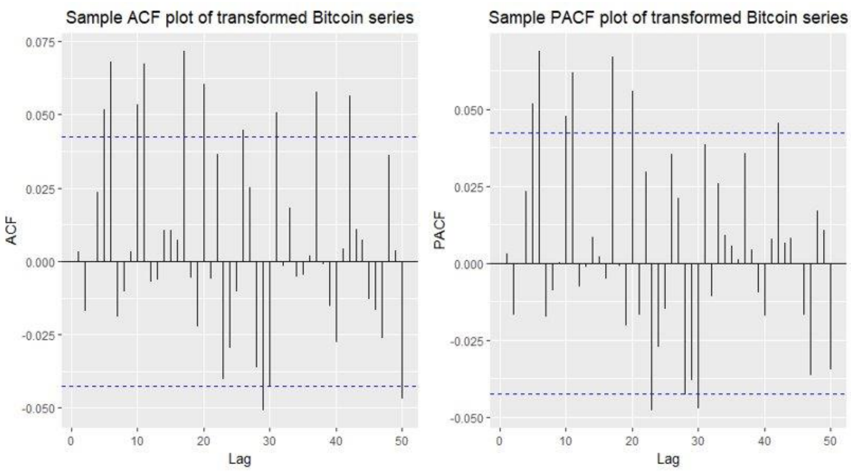


Figure 11 ACF & PACF plot

```
AR/MA
  0 1 2 3 4 5 6 7 8
0 o o o o x x o o o
1 x o o o o x o o o
2 o x o o o x o o o
3 o x o o o x o o o
4 x x o x o x o o o
5 x x x x x o o o o
6 x x x x x o o o o
7 x x o x x x x o o
8 x x x x x x x x o
```

Figure 12 EACF Table

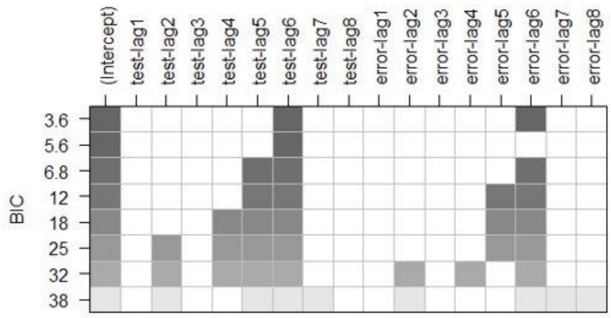


Figure 13 BIC Table

Part 4 – Model Fitting and Selection

Since the data is skewed with absence of normality (Figure 4), we use CSS method in parameter estimation as maximum likelihood (ML) has normality assumption.
According to table 1, ARIMA (2,1,2), ARIMA (4,1,4) and ARIMA (6,1,5) are significant models in coefficient testing. We shall conduct residual analysis and overfitting.

Table 1 Parameter Estimation with CSS

Model	Coefficients test with CSS	Observation
ARIMA (0,1,1)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ma1 0.0046305 0.0220369 0.2101 0.8336	Insignificant
ARIMA (0,1,2)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ma1 0.0046336 0.0216804 0.2137 0.8308 ma2 -0.0148010 0.0211223 -0.7007 0.4835	Insignificant
ARIMA (1,1,1)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ar1 -0.075717 0.260272 -0.2909 0.7711 ma1 0.081650 0.263542 0.3098 0.7567	Insignificant
ARIMA (1,1,2)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ar1 -0.077929 0.247528 -0.3148 0.7529 ma1 0.082717 0.247621 0.3340 0.7383 ma2 -0.011566 0.021253 -0.5442 0.5863	Insignificant
ARIMA (2,1,2)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ar1 0.964453 0.052087 18.516 < 2.2e-16 *** ar2 -0.820123 0.047772 -17.168 < 2.2e-16 *** ma1 -0.994856 0.053581 -18.567 < 2.2e-16 *** ma2 0.823164 0.049252 16.713 < 2.2e-16 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	All coefficients are significant

ARIMA (4,1,4)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ar1 0.839560 0.181633 4.6223 3.795e-06 *** ar2 0.028574 0.210996 0.1354 0.892276 ar3 -0.647537 0.197211 -3.2835 0.001025 ** ar4 0.587623 0.104431 5.6269 1.835e-08 *** ma1 -0.844983 0.189798 -4.4520 8.507e-06 *** ma2 -0.039235 0.217762 -0.1802 0.857016 ma3 0.665931 0.203437 3.2734 0.001063 ** ma4 -0.539018 0.108695 -4.9590 7.086e-07 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	Most of the coefficients are significant
ARIMA (5,1,5)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) ar1 1.01564 0.18945 5.3611 8.271e-08 *** ar2 -0.89449 0.34099 -2.6232 0.008711 ** ar3 -0.14402 0.57680 -0.2497 0.802829 ar4 0.42784 0.40466 1.0573 0.290384 ar5 -0.20496 0.27708 -0.7397 0.459472 ma1 -1.03441 0.18189 -5.6871 1.292e-08 *** ma2 0.89678 0.32413 2.7667 0.005662 ** ma3 0.17394 0.54518 0.3191 0.749682 ma4 -0.45127 0.38112 -1.1841 0.236386 ma5 0.25904 0.25680 1.0087 0.313116 --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	Few coefficients are significant
ARIMA (6,1,5)	Estimate Std. Error z value Pr(>|z|) ar1 0.254222 0.193271 1.3154 0.188388 ar2 -0.236218 0.113567 -2.0800 0.037527 * ar3 0.397170 0.125177 3.1729 0.001509 ** ar4 -0.304723 0.116098 -2.6247 0.008673 ** ar5 0.463020 0.098521 4.6997 2.605e-06 *** ar6 0.050363 0.036661 1.3737 0.169520 ma1 -0.261702 0.193535 -1.3522 0.176307 ma2 0.216780 0.119046 1.8210 0.068611 . ma3 -0.384661 0.120362 -3.1959 0.001394 ** ma4 0.324964 0.116536 2.7885 0.005295 ** ma5 -0.404033 0.099161 -4.0745 4.611e-05 *** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	Most of the coefficients are significant
ARIMA (6,1,6)	Estimate Std. Error z value Pr(>|z|) ar1 -0.335753 0.216210 -1.5529 0.1204466 ar2 -0.102417 0.168454 -0.6080 0.5431969 ar3 0.134972 0.149552 0.9025 0.3667866 ar4 -0.140161 0.108569 -1.2910 0.1967052 ar5 0.193837 0.134344 1.4428 0.1490659 ar6 0.441613 0.123689 3.5704 0.0003565 *** ma1 0.331001 0.224399 1.4751 0.1401979 ma2 0.088079 0.175442 0.5020 0.6156375 ma3 -0.132146 0.147344 -0.8969 0.3697984 ma4 0.180447 0.113207 1.5940 0.1109444 ma5 -0.119988 0.139784 -0.8584 0.3906800 ma6 -0.358845 0.121888 -2.9441 0.0032394 ** --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1	Very few coefficients are significant

Part 5 – Diagnostic Checking

We will perform residual analysis check on the chosen 3 significant models - ARIMA (2,1,2), ARIMA (4,1,4) and ARIMA (6,1,5) from Table 1.

Model 1 - ARIMA (2,1,2)

Residual check plots for ARIMA(2,1,2) are referred from Figure 14 gives following observations –

- Time Series plot of residuals looks random but displays volatility clustering.
- Histogram and QQ plot display wide tails proving that residuals are not normal

- ACF and PACF plots have many significant lags which proves there is an autocorrelation present in residuals.
- Ljung-Box test shows that all p-values are less than 5% significance level, suggesting that the squared residuals are correlated over time and hence, there's lot of autocorrelation present in the series

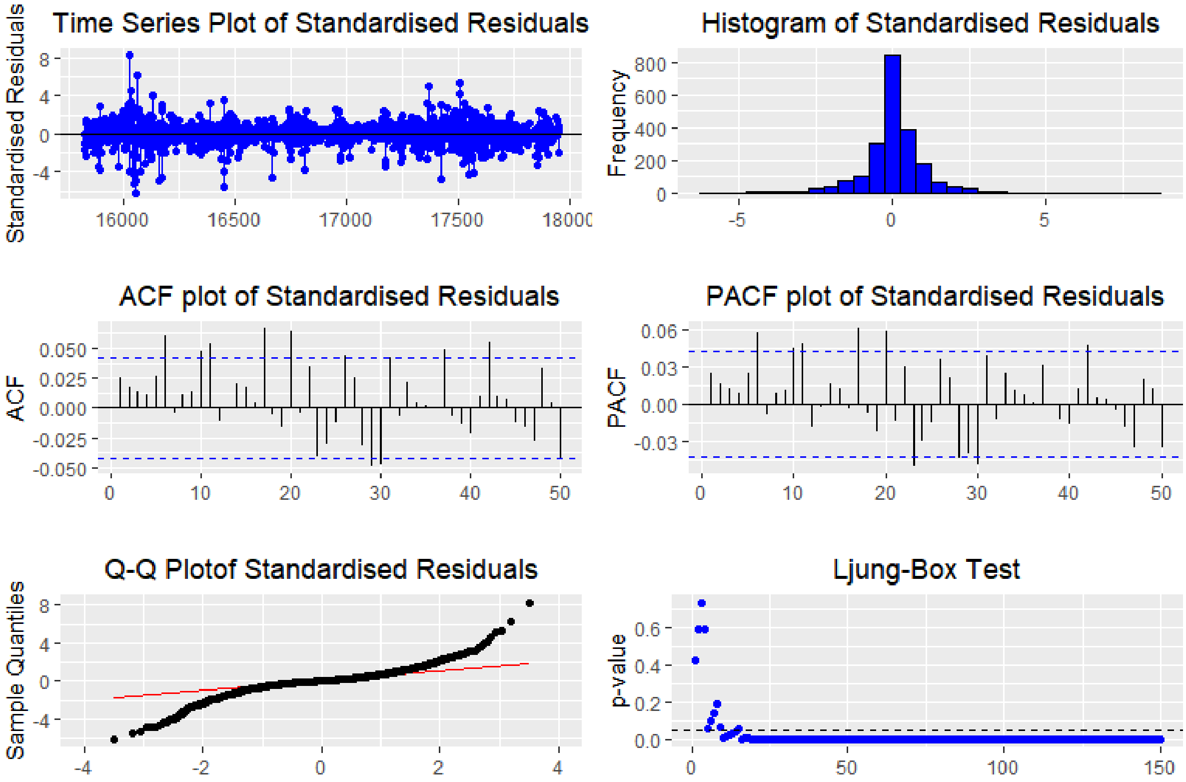


Figure 14 ARIMA(2,1,2) Residuals Check

Model 2 - ARIMA (4,1,4)

Residual check plots for ARIMA(4,1,4) are referred from Figure 15 gives following observations –

- Time Series plot of residuals looks random but displays volatility clustering.
- Histogram and QQ plot display wide tails proving that residuals are not normal
- ACF and PACF plots have fewer significant lags which proves there is an autocorrelation present in residuals.
- Ljung-Box test shows that all p-values are less than 5% significance level, suggesting that the squared residuals are correlated over time and hence, there's lot of autocorrelation present in the series.

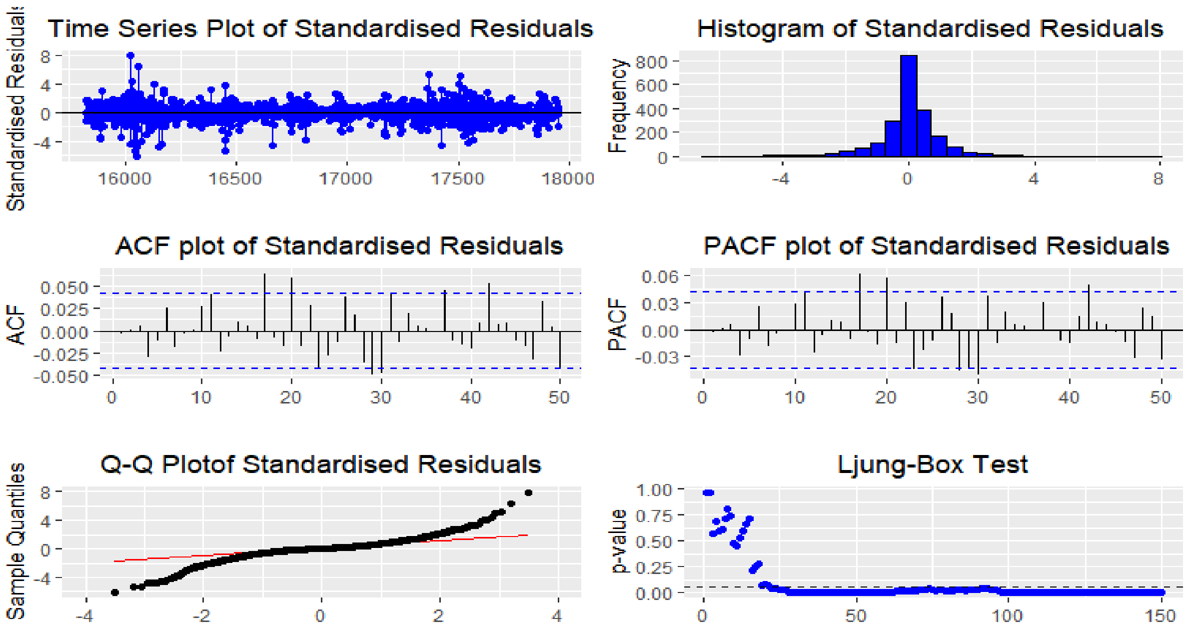


Figure 15 ARIMA(4,1,4) Residuals Check

Model 3 - ARIMA (6,1,5)

Residual check plots for ARIMA(4,1,4) are referred from Figure 16 gives following observations –

- Time Series plot of residuals looks random but displays volatility clustering.
- Histogram and QQ plot display wide tails proving that residuals are not normal
- ACF and PACF plots have few significant lags which proves there is an autocorrelation present in residuals.
- Ljung-Box test is improved but still shows that most of p-values are less than 5% significance level and there's some of autocorrelation present in the series.

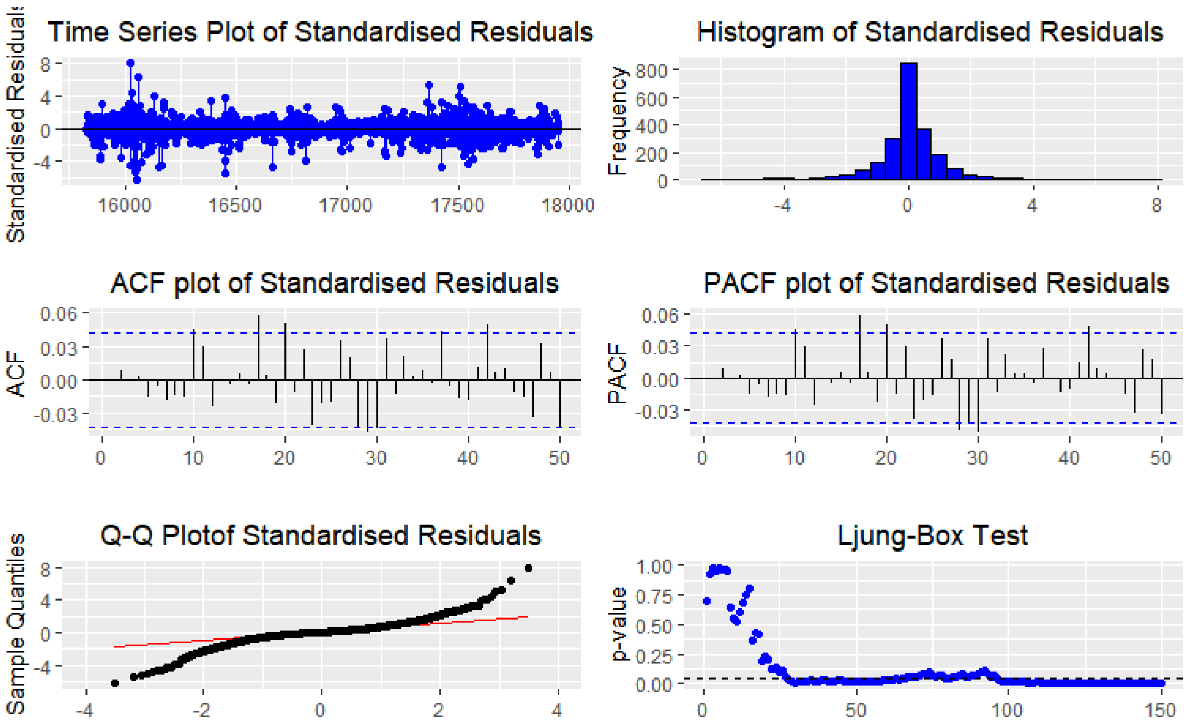


Figure 16 ARIMA(6,1,5) Residuals Check

Part 6 – Overfitting

Thus, from Diagnostic checking we choose the **best model as ARIMA(6,1,5)** and proceed with overfitting. The overfitted models for ARIMA(6,1,5) are ARIMA(6,1,6) and ARIMA(715).
From Table 1, ARIMA(6,1,6) model shows very few coefficients are significant so we reject this model.
Also from Table 2, ARIMA(7,1,5) model shows most of the coefficients are insignificant so we tend to reject this model.

Table 2 Overfitting model of ARIMA(6,1,5)

<table>
<tr><th>Model</th><th>Coefficients test with CSS</th><th>Observation</th></tr>
<tr><td>ARIMA (7,1,5)</td><td>z test of coefficients:

Estimate Std. Error z value Pr(>|z|)
ar1 0.432895 0.304667 1.4209 0.15535
ar2 0.157690 0.257632 0.6121 0.54049
ar3 -0.069852 0.220142 -0.3173 0.75101
ar4 0.149360 0.171837 0.8692 0.38474
ar5 0.219062 0.173844 1.2601 0.20763
ar6 0.047085 0.027743 1.6972 0.08966 .
ar7 -0.059487 0.026506 -2.2443 0.02482 *
ma1 -0.441403 0.304591 -1.4492 0.14729
ma2 -0.167611 0.260443 -0.6436 0.51986
ma3 0.086900 0.221699 0.3920 0.69508
ma4 -0.122831 0.177368 -0.6925 0.48861
ma5 -0.179772 0.168529 -1.0667 0.28610

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1</td><td>Most of the coefficients are Insignificant</td></tr>
</table>

Part 7 – Handle Changing Variance

The GARCH models are used to handle the changing variance and reduce the residuals to White-Noise. Since GARCH models only affects the conditional variance but not the mean prediction. Thus, we observe changing variance and it's effect in the autocorrelation of the residuals therefore, we apply ARCH-GARCH models to bring the residuals back down to White-Noise.

7.1 ARCH-GARCH Modelling Over Residuals

According to ACF and PACF plot (Figure 17), we can take p=1 and q=1. From EACF of residuals from Figure 18, supports ARMA(0,0) suggesting series is white noise in terms of ARMA components. It can be referred from EACF table that vertex is at (0,0), p = 0 or 1 and q = 0 or 1. We observe the possible set of models from EACF table are: ARMA (0,1), ARMA (1,0).
So, the total set of models – { ARMA(0,1), ARMA(1,0), ARMA(1,1) }

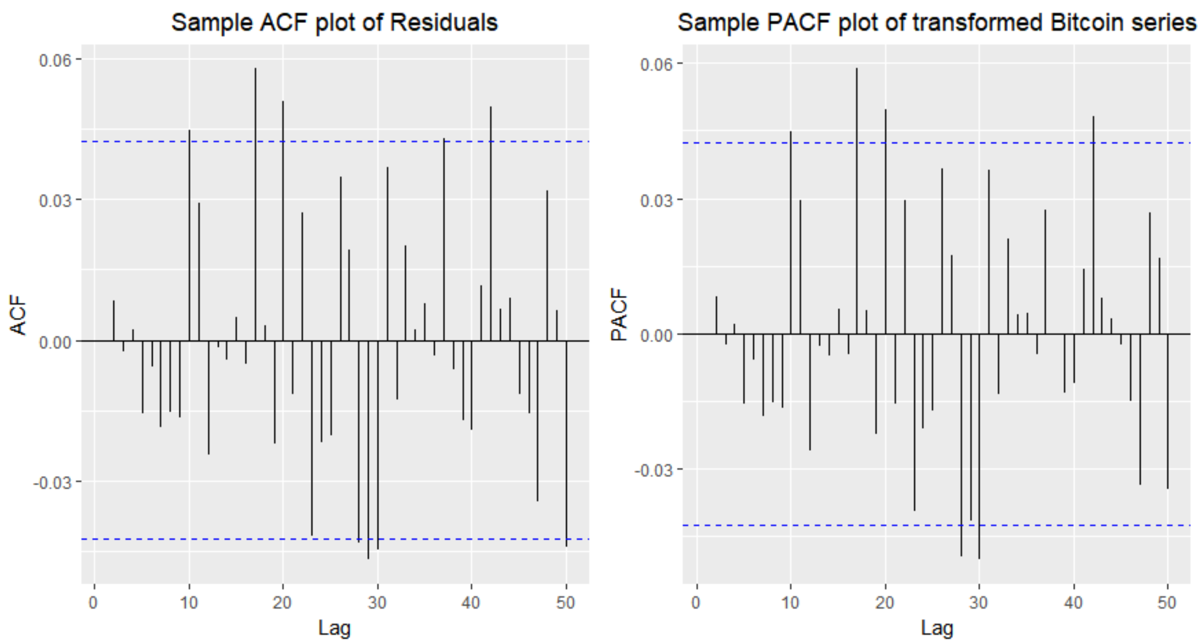


Figure 17 ACF & PACF plot

AR/MA

	0	1	2	3	4	5	6	7	8
0	o	o	o	o	o	o	o	o	o
1	o	o	o	o	o	o	o	o	o
2	x	x	o	o	o	o	o	o	o
3	x	o	x	o	o	o	o	o	o
4	x	x	x	x	o	o	o	o	o
5	x	x	o	x	x	o	o	o	o
6	x	x	o	o	x	o	o	o	o
7	x	x	x	o	x	x	o	o	o
8	x	o	x	x	x	o	o	o	o

Figure 18 EACF of Residuals

7.2 Absolute of Residuals

ACF, PACF and EACF suggest that series is not independently and identically distributed (Figure 19 & 20)

From EACF of Absolute Residuals (Figure 20), it can be inferred that vertex is at (1,1). We observe the possible set of models from EACF table are: ARMA (1,1), ARMA (1,2), ARMA(2,2).
So, the total set of models – { ARMA(1,1), ARMA(1,2), ARMA(2,2) }

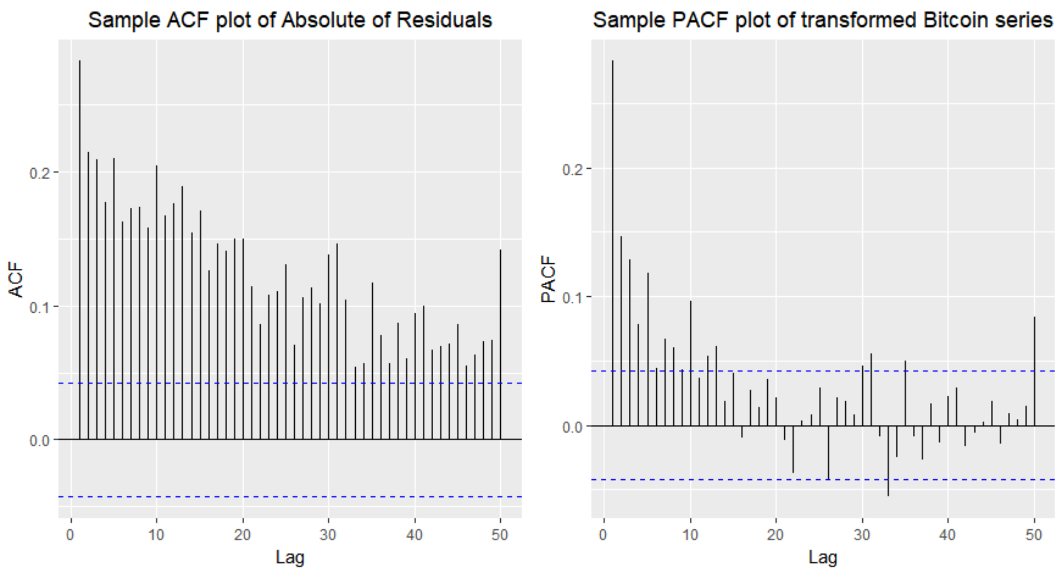


Figure 19 ACF & PACF plot

```
AR/MA
  0 1 2 3 4 5 6 7 8
0 x x x x x x x x x
1 x o o o x o o o o
2 x x o o o o o o o
3 x x x o o o o o o
4 x x x x o o o o o
5 x x x x o o o o o
6 x x x x x o o o o
7 x x x x x x x o o
8 x x x o x x x x o
```

Figure 20 EACF of Absolute Residuals

7.3 Square of Residuals

ACF, PACF and EACF suggest that series is not independently and identically distributed (Figure 21 & 22)

From EACF of Squared Residuals (Figure 22), it can be inferred that vertex is at (4,3). We observe the possible set of models from EACF table are: ARMA (4,3), ARMA (4,4)
So, the total set of models – { ARMA (4,3), ARMA (4,4) }

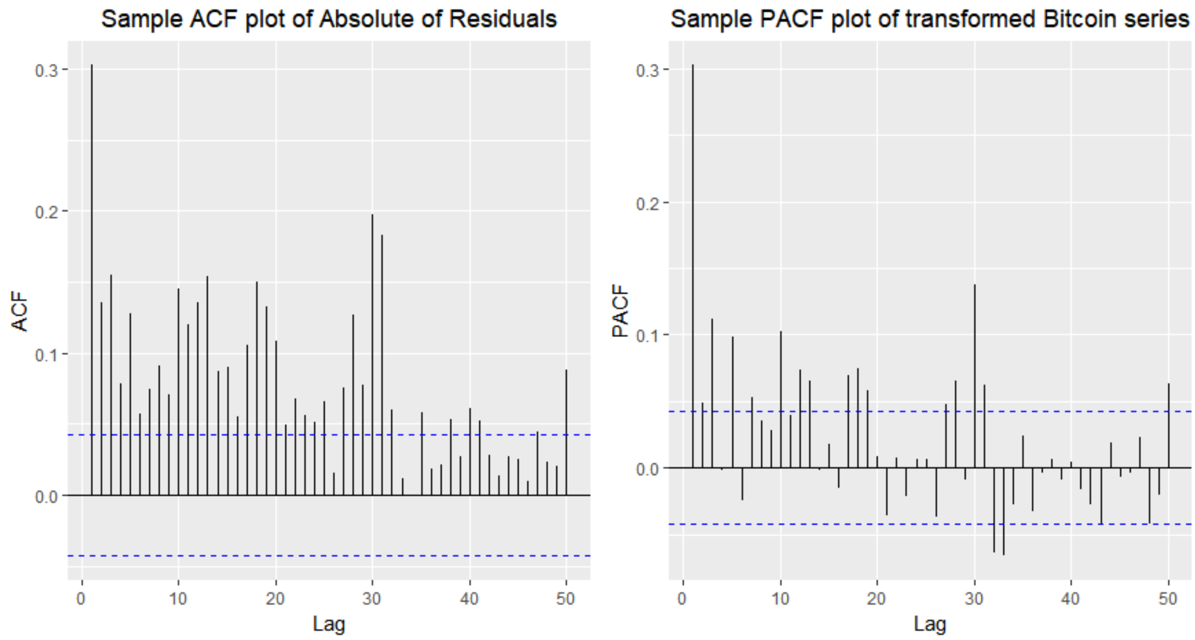


Figure 21 ACF & PACF plot

AR/MA	0	1	2	3	4	5	6	7	8
0	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	o	o	o
2	x	x	x	o	o	x	o	o	o
3	o	x	x	o	o	x	o	o	o
4	o	x	x	o	o	o	o	o	o
5	x	x	x	x	x	x	o	o	o
6	x	x	x	x	x	x	o	o	o
7	x	o	x	x	x	x	o	x	o
8	x	o	x	x	x	x	o	x	o

Figure 22 EACF of Squared Residuals

Final set of possible models from ARCH-GARCH Modelling Over Residuals, Absolute of Residuals and Square of Residuals - { ARMA(0,1), ARMA(1,0), ARMA(1,1), ARMA(1,2), ARMA(2,2), ARMA (4,3), ARMA (4,4) }

- Possible p-q orders ACF, PACF and EACF = {01, 11, 10, 12, 22, 43, 44}
- Thus, possible ARCH-GARCH orders (max(p,q),p) = {10, 11, 21, 22, 43, 44}

So, the possible set of GARCH models – GARCH(1,0), GARCH(1,1), GARCH(2,1), GARCH(2,2), GARCH(4,3), GARCH(4,4)

7.4 Model Specification of GARCH Models

According to table 3, GARCH(1,1), GARCH(2,1) and GARCH(2,2) are significant models in coefficient testing. We will conduct residual analysis on these 3 models.

Table 3 Parameter Estimation of GARCH models

Model	Coefficients test with CSS	Observation
GARCH (1,0)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) a0 0.0017471 NA NA NA b1 0.0500000 NA NA NA	Insignificant
GARCH (1,1)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) a0 3.7816e-05 3.1872e-06 11.865 < 2.2e-16 *** a1 1.1894e-01 7.4080e-03 16.055 < 2.2e-16 *** b1 8.7051e-01 6.2283e-03 139.766 < 2.2e-16 *** --- Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1	All coefficients are significant
GARCH (2,1)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) a0 5.1868e-05 4.7923e-06 10.8232 < 2.2e-16 *** a1 1.7403e-01 1.1454e-02 15.1939 < 2.2e-16 *** b1 2.3011e-01 4.8532e-02 4.7413 2.123e-06 *** b2 5.8161e-01 4.3800e-02 13.2787 < 2.2e-16 *** --- Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1	All coefficients are significant
GARCH (2,2)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) a0 5.1890e-05 5.7615e-06 9.0063 < 2.2e-16 *** a1 1.7398e-01 1.5926e-02 10.9248 < 2.2e-16 *** a2 9.4326e-05 2.0748e-02 0.0045 0.996373 b1 2.2989e-01 8.3465e-02 2.7543 0.005881 ** b2 5.8177e-01 7.2067e-02 8.0726 6.88e-16 *** --- Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1	Most of the coefficients are significant
GARCH (4,3)	z test of coefficients: Estimate Std. Error z value Pr(>|z|) a0 6.7044e-05 1.0344e-05 6.4817 9.067e-11 *** a1 1.8683e-01 1.8839e-02 9.9172 < 2.2e-16 *** a2 2.5422e-02 4.0314e-02 0.6306 0.52830 a3 3.5463e-02 2.7140e-02 1.3067 0.19133 b1 2.9421e-01 1.6363e-01 1.7980 0.07218 . b2 5.0811e-02 1.2246e-01 0.4149 0.67821 b3 4.1620e-07 9.2398e-02 0.0000 1.00000 b4 3.9344e-01 5.9951e-02 6.5627 5.285e-11 *** --- Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1	Very few coefficients are significant

GARCH (4,4)	`z test of coefficients:`
	
	`   Estimate Std. Error z value Pr(>|z|)`
	`a0 8.6942e-05 NA NA NA`
	`a1 1.3473e-01 NA NA NA`
	`a2 7.2201e-02 NA NA NA`
	`a3 9.2188e-02 NA NA NA`
	`a4 1.6596e-03 NA NA NA`
	`b1 1.0479e-01 NA NA NA`
	`b2 1.2204e-01 NA NA NA`
	`b3 1.8467e-01 NA NA NA`
	`b4 2.6812e-01 NA NA NA`
		Insignificant

7.5 Residual Analysis of GARCH Models

After residual analysis of GARCH models and AIC table (Figure 26), It's clear that the GARCH(2,1) has the best AIC and all significant components in terms of residuals.

Model 1 - GARCH (1,1)

Residual check plots for GARCH(1,1) are referred from Figure 23, gives following observations –

- Time Series plot of residuals looks random and displays constant variance.
- Histogram and QQ plot display wide tails proving that residuals are not normal.
- ACF and PACF plots has one-one significant lag which proves there is no significant autocorrelation present in residuals.
- Ljung-Box test shows that all p-values are greater than 5% significance level hence, there is no autocorrelation present in the series.

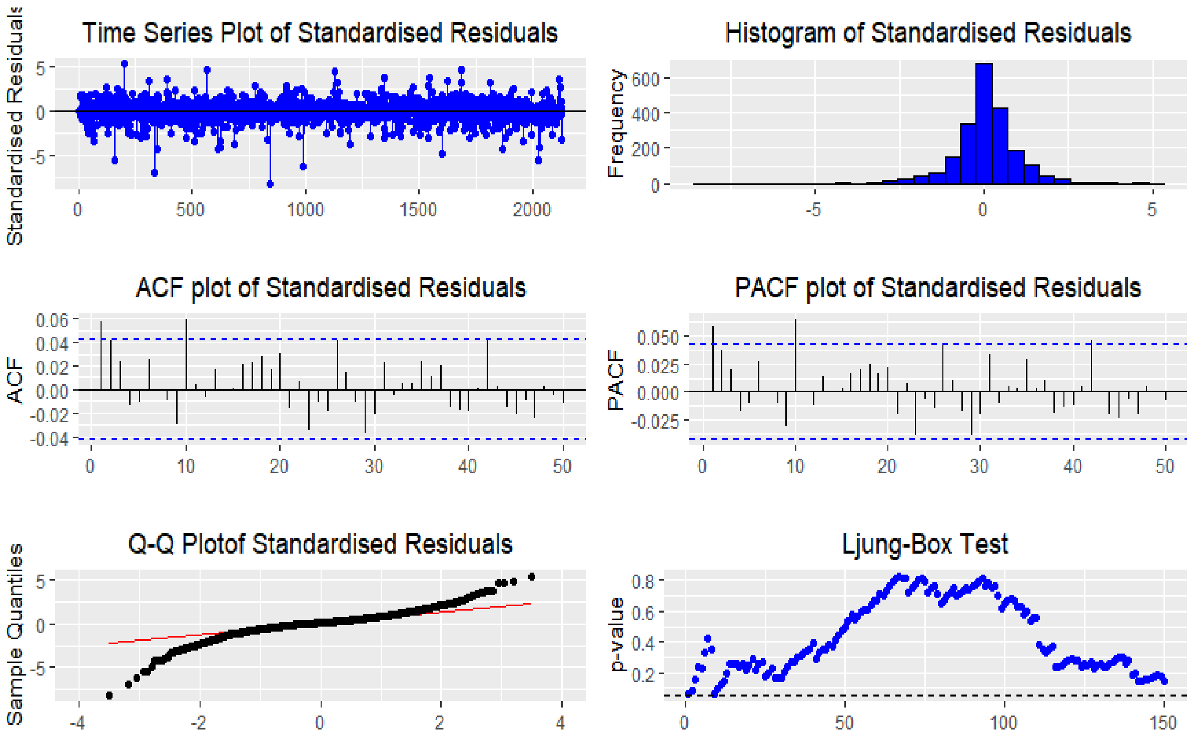


Figure 23 GARCH(1,1) Residuals Check

Model 2 - GARCH (2,1)

Residual check plots for GARCH(2,1) are referred from Figure 24, gives following observations –

- Time Series plot of residuals looks random and displays constant variance.
- Histogram and QQ plot display wide tails proving that residuals are not normal.
- ACF and PACF shows slight improvement in significant lags which further proves there is no significant autocorrelation present in residuals.
- Ljung-Box test gives almost significant values above 5% level. Thus, we can say that there is no autocorrelation present in the series.

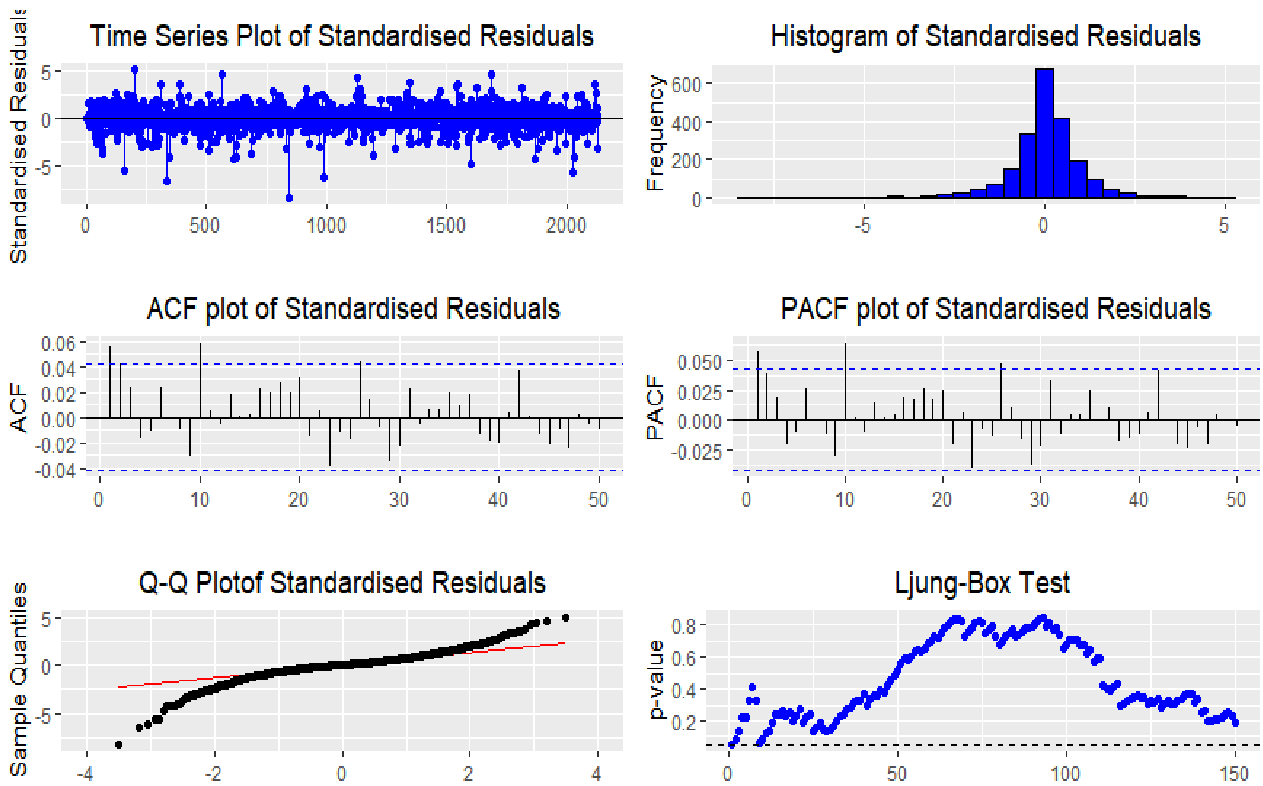


Figure 24 GARCH(2,1) Residuals Check

Model 3 - GARCH (2,2)

Residual check plots for GARCH(2,2) are referred from Figure 25, gives following observations –

- Time Series plot of residuals looks random enough and displays constant variance.
- Histogram and QQ plot display wide tails proving that residuals are not normal.
- ACF and PACF shows slight improvement in significant lags which further proves there is no significant autocorrelation present in residuals.
- Ljung-Box test gives almost significant values above 5% level. Thus, we can say that there is no autocorrelation present in the series.

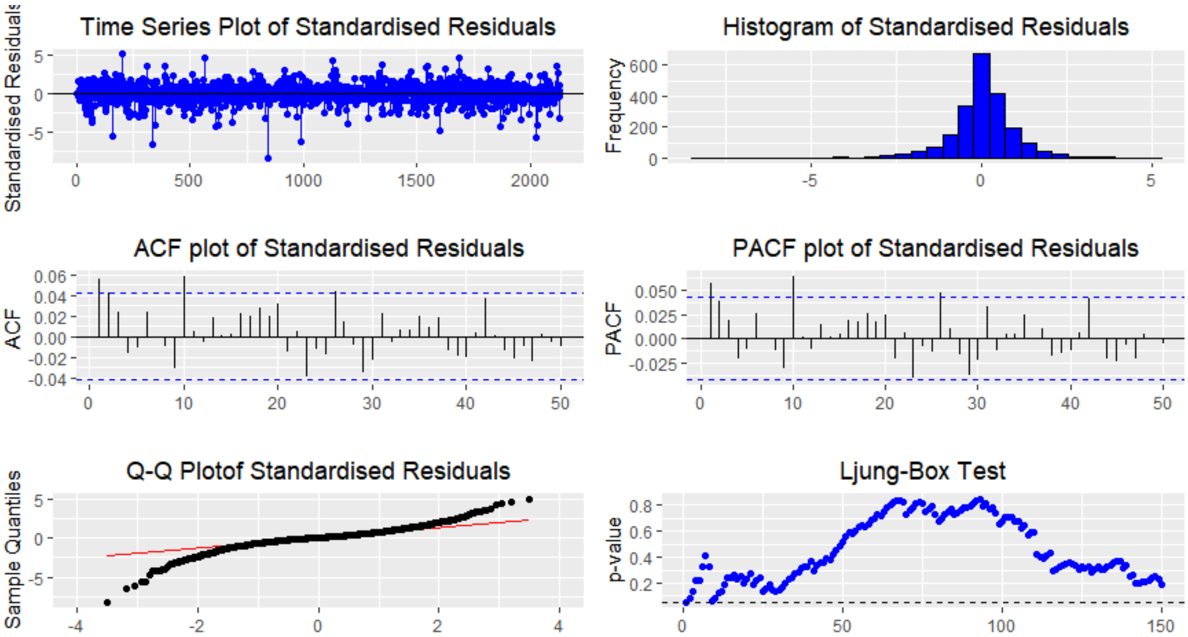


Figure 25 GARCH(2,2) Residuals Check

AIC Test

	df	AIC
garch.11	3	-8025.441
garch.21	4	-8031.708
garch.22	5	-8029.708

Figure 26 AIC Table

Part 8 – Forecasting

For predicting the mean values we used ARIMA(6,1,5) as the best fitted model and can be seen in the Forecasting Plot (Figure 27). Forecast the value of bitcoin for the next 10 days, starting Feb 24th 2019

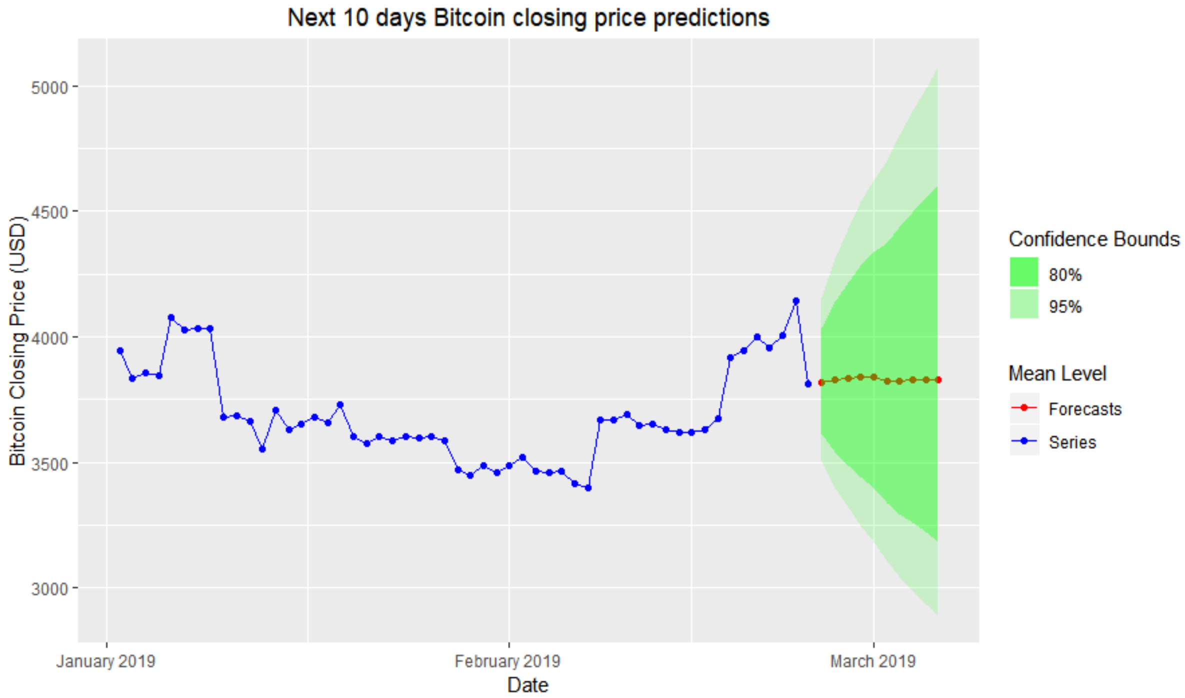


Figure 27 Forecast the value of bitcoin for the next 10 days

Part 9 – Conclusion

The series was log transformed to make it more stationary. ARIMA modelling with p,d,q order 6,1,5 respectively is used to make the mean value forecasts. The changing variance is well handled using the GARCH(2,1) model which has improved residuals and best AIC value.

9.1 Goodness of Fit

The model ARIMA(6,1,5) is the best model for prediction
Mean Absolute Scaled Error (MASE) for the fit = 0.996

9.2 Prediction Error

Mean Absolute Scaled Error (MASE) for the forecast = 1.311

Reference

- CoinMarketCap 2019, *Bitcoin*, data file, CoinMarketCap, viewed 20 May 2019, < https://coinmarketcap.com/ >

Appendix – R code

```
# Load packages
library(pacman)
p_load(TSA)
p_load(FSAdata)
p_load(fUnitRoots)
p_load(lmtest)
p_load(data.table)
p_load(lubridate)
p_load(tidyverse)
p_load(tseries)
p_load(forecast)
p_load(qqplotr)
p_load(FitAR)
p_load(cowplot)
theme_set(theme_gray())
library(ggplot2)

# Function to calculate Mean Absolute Scaled Error

MASE = function(observed , fitted ){
  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))
}

# Function to plot residual analysis graphs

residual_analysis <- function(model, std = TRUE,start = 2, class = c("ARIMA","GARCH","ARMA-GARCH")[1]){
  p_load(TSA)
  p_load(FitAR)
  p_load(tidyverse)
  p_load(cowplot)
  p_load(qqplotr)
  theme_set(theme_gray())
  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$residuals
 }else {
   stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
 }
 timeseries <- ggplot(mapping=aes(y=res.model,x=time(res.model))) + geom_point(color = "blue") + geom_line(color = "blue") + geom_hline(yintercept = 0) + ylab("Standardised Residuals") + xlab("") + ggtitle("Time Series Plot of Standardised Residuals") + theme(plot.title = element_text(hjust = 0.5))
 histogram <- ggplot(mapping=aes(x=res.model)) + geom_histogram(color = "black",fill = "blue") + xlab("") + ylab("Frequency") + ggtitle("Histogram of Standardised Residuals") + theme(plot.title = element_text(hjust = 0.5))
 qqplot <- ggplot(data=NULL,aes(sample=res.model)) + geom_qq_line(color = "red") + geom_qq() + ylab("Sample Quantiles") + xlab("") + ggtitle("Q-Q Plotof Standardised Residuals") + theme(plot.title = element_text(hjust = 0.5)) + xlim(-4,4)
 shapiro.test(res.model)
 acf <- autoplot(acf(res.model, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ylab("ACF") + xlab("") + ggtitle("ACF plot of Standardised Residuals") + theme(plot.title = element_text(hjust = 0.5))
 pacf <- autoplot(acf(res.model,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ylab("PACF") + xlab("") + ggtitle("PACF plot of Standardised Residuals")  + theme(plot.title = element_text(hjust = 0.5))
 lbq <- LjungBoxTest(res.model,lag.max = 150) %>% data.frame()
 lbqplot <- ggplot(mapping=aes(y=lbq$pvalue,x=lbq$m)) + geom_point(color = "blue") + geom_hline(yintercept = 0.05,linetype="dashed") + ylab("p-value") + xlab("") + ggtitle("Ljung-Box Test") + theme(plot.title = element_text(hjust = 0.5))
 plot_grid(timeseries,histogram,acf,pacf,qqplot,lbqplot,ncol = 2,nrow = 3,align = "h")
}

# Reading and cleaning the data
bitcoin <- fread("Bitcoin_Historical_Price.csv") # Actual data
bitcoin_forecast <- fread("Bitcoin_Prices_Forecasts.csv") # Actual forecast values
bitcoin$Close <- str_remove_all(bitcoin$Close,",") # Removing extra ',' in the closing price
bitcoin$Close <- as.numeric(bitcoin$Close) # Numeric conversion
head(bitcoin) # Top 6 rows of the data
summary(bitcoin) # Summary statistics
bitcoin$Date <- dmy(bitcoin$Date) # Converting date fromnumeric to date object
bitcoin_ts <- ts(bitcoin$Close,start = min(bitcoin$Date),end = max(bitcoin$Date)) # Converting data to timeseries object
summary(bitcoin_ts) # Summary statistics of timeseries data
time <- time(bitcoin_ts) %>% as.numeric()
summary(as.Date.numeric(time,origin = "1970-01-01")) # Summary statistics of Date

# Timeseries plot of Closing Price

ggplot(mapping=aes(y=bitcoin_ts,x=bitcoin$Date)) + geom_point(color = "blue") + geom_line(color = "blue") + ylab("Bitcoin Closing Price (USD)") + xlab("Date") + ggtitle("Time Series Plot of Bitcoin Closing Price") + theme(plot.title = element_text(hjust = 0.5)) + scale_x_date(breaks = "1 year",date_labels = "%Y")
# Visible trend
# Changing variance in the latter part
# Seasonality not very obvious
```

```
# Intervention point could exist
# AR behaviour mostly

# Autocorrelation function plots

acf <- autoplot(acf(bitcoin_ts, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ggtitle("Sample ACF plot of Bitcoin series") + theme(plot.title = element_text(hjust = 0.5))
pacf <- autoplot(acf(bitcoin_ts,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ylab("PACF") + ggtitle("Sample PACF plot of Bitcoin series") + theme(plot.title = element_text(hjust = 0.5))
plot_grid(acf, pacf, ncol=2, align="h")

# Slowly decaying pattern in ACF confirms presence of trend
# Multiple significant lags after few insignificant lags suggest changing variance

# Normality check

shapiro.test(bitcoin_ts)
histogram <- ggplot(mapping=aes(x=bitcoin_ts)) + geom_histogram(color = "black",fill = "blue") + xlab("Bitcoin Closing Price") + ylab("Frequency") + ggtitle("Histogram of Bitcoin Closing Price") + theme(plot.title = element_text(hjust = 0.5))
qqplot <- ggplot(data=NULL,aes(sample=bitcoin_ts)) + geom_qq_line(color = "red") + geom_qq() + ylab("Sample Quantiles") + xlab("Theoretical Quantiles") + ggtitle("Normal Q-Q Plot") + theme(plot.title = element_text(hjust = 0.5)) + xlim(-4,4)
plot_grid(histogram, qqplot, ncol=2, align="h")
# Data is non-normal - extremely right skewed

# BoxCox transformation

b <- BoxCox.ar(bitcoin_ts,method = "yw")
ggplot(mapping = aes(y=b$loglike,x=b$lambda)) + geom_line(color="blue") + geom_vline(xintercept = b$ci,color="red",linetype="dashed") + geom_vline(xintercept = b$mle,linetype="dashed") + geom_hline(yintercept = quantile(b$loglike,0.95),linetype="dashed") + geom_text(aes(x=-1.5,y=max(b$loglike)*1.02),label = "95% Confidence Interval",size = 2.5)
# Suggested lambda value = 0 implying log transformation

bitcoin_log <- log(bitcoin_ts)
ggplot(mapping=aes(y=bitcoin_log,x=bitcoin$Date)) + geom_point(color = "blue") + geom_line(color = "blue") + ylab("Bitcoin Closing Price (USD)") + xlab("Date") + ggtitle("Time Series Plot of log of Bitcoin Closing Price") + theme(plot.title = element_text(hjust = 0.5)) + scale_x_date(breaks = "1 year",date_labels = "%Y")
# Trend is still present
# Effect of changing variance looks reduced
# Seasonality not very obvious
# Intervention point does not exist
# AR behaviour mostly

# Augmented Dickey-Fuller Test

ar(diff(bitcoin_log))
adfTest(bitcoin_log,31)
```

```
# ADF Test confirms the presence of trend

# Differencing

bitcoin_log_diff <- diff(bitcoin_log)
ggplot(mapping=aes(y=bitcoin_log_diff,x=bitcoin$Date[-1])) + geom_point(color = "blue") + geom_line(color = "blue") + ylab("Bitcoin Closing Price (USD)") + xlab("Date") + ggtitle("Time Series Plot of difference of log of Bitcoin Closing Price") + theme(plot.title = element_text(hjust = 0.5)) + scale_x_date(breaks = "1 year",date_labels = "%Y")
# Trend has been removed
# Visible volatility clustering
# Seasonality not very obvious
# Intervention point does not exist
# Both AR and MA behaviour are exhibited

# Augmented Dickey-Fuller Test

ar(diff(bitcoin_log_diff))
adfTest(bitcoin_log_diff,32)
# ADF Test confirms that the series has been detrended

# Parameter Estimation

acf <- autoplot(acf(bitcoin_log_diff, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ggtitle("Sample ACF plot of transformed Bitcoin series") + theme(plot.title = element_text(hjust = 0.5))
pacf <- autoplot(acf(bitcoin_log_diff,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ylab("PACF") + ggtitle("Sample PACF plot of transformed Bitcoin series")  + theme(plot.title = element_text(hjust = 0.5))
plot_grid(acf, pacf, ncol=2, align="h")
# ACF and PACF have similar pattern, Values of p and q can be 2 or 4
# Thus the set of possible models from ACF and PACF = {ARIMA(2,1,2),ARIMA(4,1,4)}

eacf(bitcoin_log_diff,ar.max = 8,ma.max = 8)
# EACF has vertex at (0,0)
# Thus the set of possible models from EACF = {ARIMA(0,1,1),ARIMA(1,1,1),ARIMA(0,1,2),ARIMA(1,1,2)}

res = armasubsets(y=bitcoin_log_diff,nar=8,nma=8,y.name='test',ar.method='ols')
plot(res)
# The set of possible models from BIC Table = {ARIMA(5,1,5),ARIMA(6,1,5),ARIMA(6,1,6)}

# Thus the final set of possible models is {ARIMA(0,1,1),ARIMA(0,1,2),ARIMA(1,1,1),ARIMA(1,1,2),ARIMA(2,1,2),ARIMA(4,1,4),ARIMA(5,1,5),ARIMA(5,1,6),ARIMA(6,1,5),ARIMA(6,1,6)}

# Model Specification

model.011.css = arima(bitcoin_log,order=c(0,1,1),method='CSS')
coeftest(model.011.css)
# Insignificant
```

```
model.012.css = arima(bitcoin_log,order=c(0,1,2),method='CSS')
coeftest(model.012.css)
# Insignificant

model.111.css = arima(bitcoin_log,order=c(1,1,1),method='CSS')
coeftest(model.111.css)
# Insignificant

model.112.css = arima(bitcoin_log,order=c(1,1,2),method='CSS')
coeftest(model.112.css)
# Insignificant

model.212.css = arima(bitcoin_log,order=c(2,1,2),method='CSS')
coeftest(model.212.css)
# All coefficients significant

model.414.css = arima(bitcoin_log,order=c(4,1,4),method='CSS')
coeftest(model.414.css)
# Mostly coefficients significant

model.515.css = arima(bitcoin_log,order=c(5,1,5),method='CSS')
coeftest(model.515.css)
# Very few coefficients significant - Rejected

model.615.css = arima(bitcoin_log,order=c(6,1,5),method='CSS')
coeftest(model.615.css)
# Mostly coefficients significant

model.616.css = arima(bitcoin_log,order=c(6,1,6),method='CSS')
coeftest(model.616.css)
# Very few coefficients significant - Rejected

# Residual Analysis for significant models

residual_analysis(model.212.css)
residual_analysis(model.414.css)
residual_analysis(model.615.css)
# ARIMA(6,1,5) selected due to best residuals


# Overfitting

model.715.css = arima(bitcoin_log,order=c(7,1,5),method='CSS')
coeftest(model.715.css)
# Coefficients mostly insignificant - Rejected

# Arch-Garch modelling over residuals to handle changing variance

res <- residuals(model.615.css)
acf <- autoplot(acf(res, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ggtitle("Sample
ACF plot of Residuals") + theme(plot.title = element_text(hjust = 0.5))
```

```
pacf <- autoplot(acf(res,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0))
+ ylab("PACF") + ggtitle("Sample PACF plot of transformed Bitcoin series")  + theme(plot.title =
element_text(hjust = 0.5))
plot_grid(acf, pacf, ncol=2, align="h")
# Multiple significant lags in ACF and PACF

eacf(res,ar.max = 8,ma.max = 8)
# EACF supports ARMA(0,0) suggesing series is white noise in terms of ARMA components

# Absolute of Residuals

abs <- abs(res)
acf <- autoplot(acf(abs, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ggtitle("Sample
ACF plot of Absolute of Residuals") + theme(plot.title = element_text(hjust = 0.5))
pacf <- autoplot(acf(abs,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0))
+ ylab("PACF") + ggtitle("Sample PACF plot of transformed Bitcoin series")  + theme(plot.title =
element_text(hjust = 0.5))
plot_grid(acf, pacf, ncol=2, align="h")
# ACF and PACF suggest that series is not independently and identically distributed
eacf(abs,ar.max = 8,ma.max = 8)

sq <- (res)^2
acf <- autoplot(acf(sq, plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0)) + ggtitle("Sample
ACF plot of Absolute of Residuals") + theme(plot.title = element_text(hjust = 0.5))
pacf <- autoplot(acf(sq,type = "partial", plot = FALSE,lag.max=50)) + geom_hline(aes(yintercept = 0))
+ ylab("PACF") + ggtitle("Sample PACF plot of transformed Bitcoin series")  + theme(plot.title =
element_text(hjust = 0.5))
plot_grid(acf, pacf, ncol=2, align="h")
eacf(sq,ar.max = 8,ma.max = 8)
# ACF, PACF and EACF suggest that series is not independently and identically distributed

# Possible p-q orders ACF, PACF and EACF = {01, 11, 10, 12, 22, 34, 44}
# Thus, possible Arch-Garch orders (max(p,q),p) = {10, 11, 21, 22, 43, 44}

# Model Specification

garch.10 <- garch(res,c(1,0))
coeftest(garch.10)
# Insignificant

garch.11 <- garch(res,c(1,1))
coeftest(garch.11)
# All significant

garch.21 <- garch(res,c(2,1))
coeftest(garch.21)
# All significant

garch.22 <- garch(res,c(2,2))
coeftest(garch.22)
# Mostly significant
```

```
garch.43 <- garch(res,c(4,3))
coeftest(garch.43)
# Very few significant - rejected

garch.44 <- garch(res,c(4,4))
coeftest(garch.44)
# Insignificant

# Residual Analysis
residual_analysis(garch.11, std = TRUE,start = 3, class = "GARCH")
residual_analysis(garch.21, std = TRUE,start = 3, class = "GARCH")
residual_analysis(garch.22, std = TRUE,start = 3, class = "GARCH")
# Similar residuals
# Series looks almost white noise

AIC(garch.11,garch.21,garch.22)
# garch(2,1) has the best AIC and all significant components

# Goodness of Fit
mod = Arima(bitcoin_ts,order=c(6,1,5),method='CSS',lambda = 0)
MASE_fit <- MASE(bitcoin_ts,mod$fitted)$MASE
paste("Mean Absolute Scaled Error for the fit = ",round(MASE_fit,3))

# Forecasting

pred <- forecast(mod,h=10)

# Forecast plot

p <- ggplot() + geom_point(data=bitcoin[bitcoin$Date>dmy("01-01-2019")],mapping=aes(y=Close,x=Date,color = "Series")) + geom_line(data=bitcoin[bitcoin$Date>dmy("01-01-2019")],mapping=aes(y=Close,x=Date,color = "Series")) + ylab("Bitcoin Closing Price (USD)") + xlab("Date") + ggtitle("Next 10 days Bitcoin closing price predictions") + theme(plot.title = element_text(hjust = 0.5)) + scale_x_date(breaks = "1 month",date_labels = "%B %Y")
forecast_date <- seq(max(bitcoin$Date)+1,max(bitcoin$Date)+10,1)
p + geom_point(mapping=aes(y=pred$mean,x=forecast_date,color = "Forecasts")) + geom_line(mapping=aes(y=pred$mean,x=forecast_date,color = "Forecasts")) + geom_ribbon(mapping = aes(x=forecast_date,ymin=pred$lower[,1],ymax=pred$upper[,1],alpha = "80%"),fill="green") + geom_ribbon(mapping = aes(x=forecast_date,ymin=pred$lower[,2],ymax=pred$upper[,2],alpha = "95%"),fill="green") + scale_colour_manual(name="Mean Level",values=c("Series" = "blue","Forecasts" = "red"), guide = guide_legend(fill = NULL,colour = NULL)) + scale_alpha_manual(name="Confidence Bounds",values=c("80%" = 0.35,"95%" = 0.15), guide = guide_legend(fill = NULL,colour = NULL))

# Prediction error

MASE_forecast <- MASE(bitcoin_forecast$`Closing price`,pred$mean)$MASE
paste("Mean Absolute Scaled Error for the forecast = ",round(MASE_forecast,3))
```