MS4218 $BTC Time Series Analysis

Daragh O'Brien (20247826)

April 2024

1 Introduction

This project will use time-series analysis methods to predict future price movement of Bitcoin (ticker $BTC). The dataset being used contains the monthly high prices from 1/10/2014 untill 26/02/2024. All model building will be completed on the first 90% of the data and then the final 10% of data will be used to make predictions. This project was chosen as it is common belief in the cryptocurrency sector that Bitcoins price movement stems from four-year cycles dictated by its halving periods, where the value of Bitcoin rewarded from mining halves. The next halving period is set to begin 20/4/2024. From the assignment, disregarding the introduction and conclusion, there is seven main steps. Steps one and two only require code and have nothing to show for. Therefore, this report will have a section for the five remaining steps where the requirements of the step will be given along with all work produced.

2 Step 3

3) Assess whether or not the series is stationary and apply appropriate transformations if necessary.

A graph of the unedited time series on the training data is given below by figure 1:

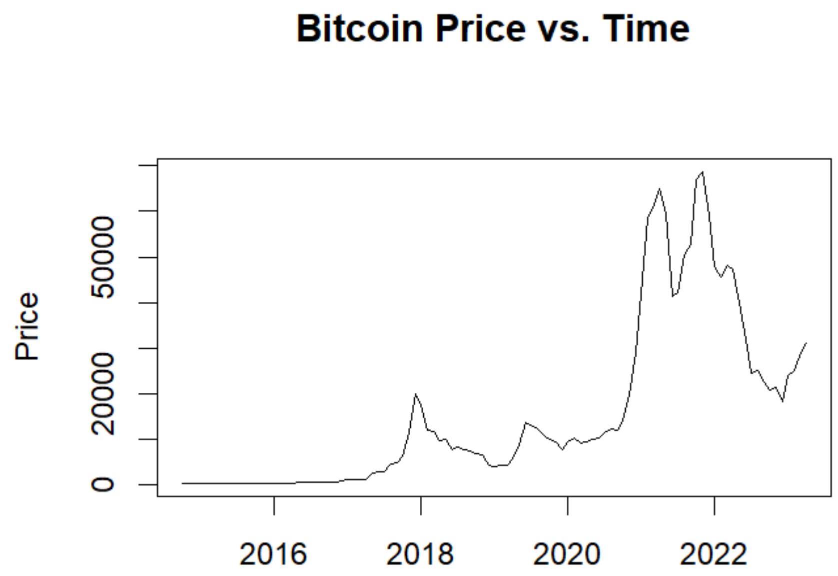


Figure 1: Bitcoin Time Series.

It seems that there is increasing trend and variance in the data, this violates the stationary assumption. The first thing that will be addressed is a potentially rising variance. A Box_Cox transformation can be used for this and is given by figure 2:

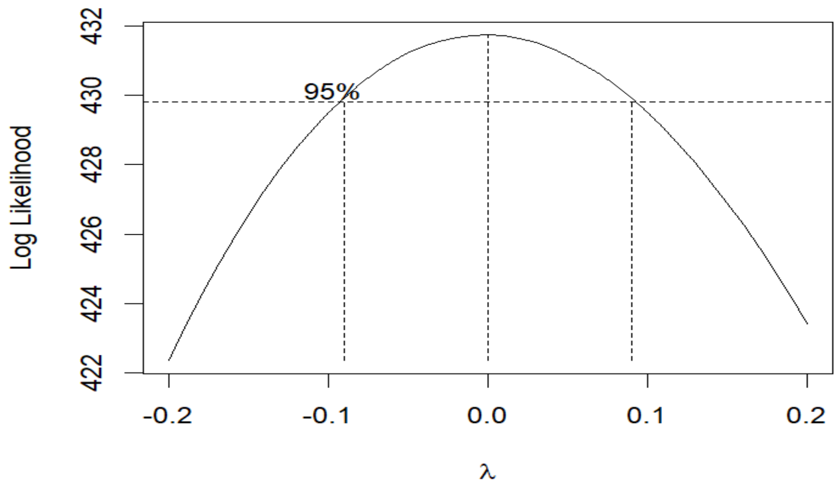


Figure 2: BoxCox.ar Output.

The BoxCox.ar function returned a maximum likelyhood estimate (mle) of 0, within a 95% confidence interval of -0.09, 0.09 which supports a log transformation and is evidence against not using a transformation as 1 is not in the interval. The logged data will now be used.

The next thing to address is a potentially rising trend. There are two ways of checking this, Firstly, different values of the autocorrelation function can be plotted, if the values fail to decay over time this may be an indicator of trend. This may be because if the points consistently go in one general direction they are more correlated to older points. Secondly, the augmented Dickey-Fuller test, which has null hypothesis and alternative hypothesis:

- H_0: Differencing needed to produce stationary series.
- H_1: Series is stationary.

The results of plotting the acf are given by figure 3:

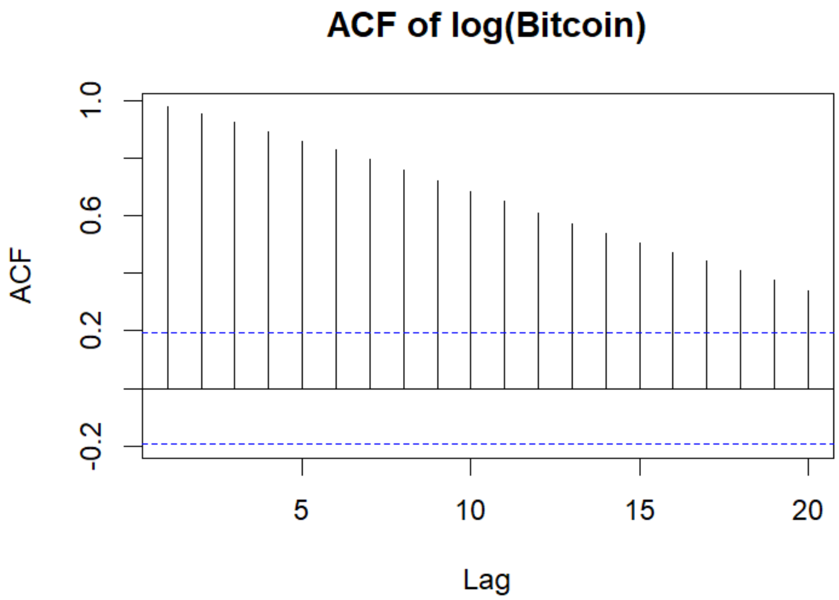


Figure 3: ACF of log(Bitcoin).

It is clear from figure 3 that the acf decays slowly with time, so an augmented Dickey-Fuller test will be performed and is given by figure 4:

```
	Augmented Dickey-Fuller Test

data:  train2
Dickey-Fuller = -1.9718, Lag order = 4, p-value = 0.5883
alternative hypothesis: stationary
```

Figure 4: adf.test of log(Bitcoin).

The null hypothesis H_0 has failed to be rejected and it is now clear that differencing is needed. The logged series is differenced once and the new acf plot is given by figure 5 and adf.test given by figure 6:

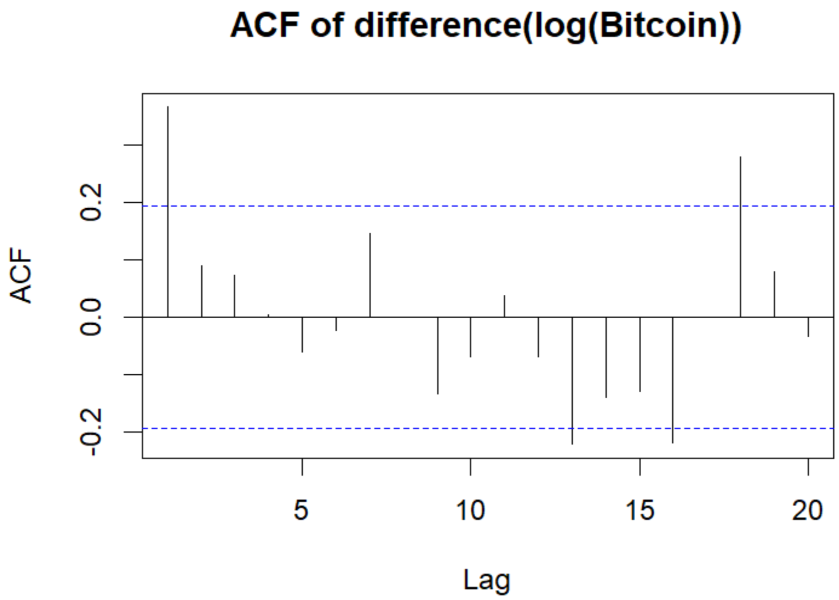


Figure 5: ACF Plot of difference(log(Bitcoin)).

```
        Augmented Dickey-Fuller Test

data:  difflogbtc
Dickey-Fuller = -4.2477, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Figure 6: adf.test of difference(log(Bitcoin)).

The behaviour of the acf is now more in line with that of a stationary series and the augmented Dickey-Fuller test rejects the null hypothesis H_0 that the series needs differencing.

3 Step 4

4) Decide on some reasonable models for the data using the methods covered on this course.

The ACF plot given by figure 5 suggests that the 1st, 13th, 16th and 18th lag autocorrelations terms are significant although one of these is likely to be at random. To continue information from the PACF will also be needed. The plot of the PACF is given below by figure 7:

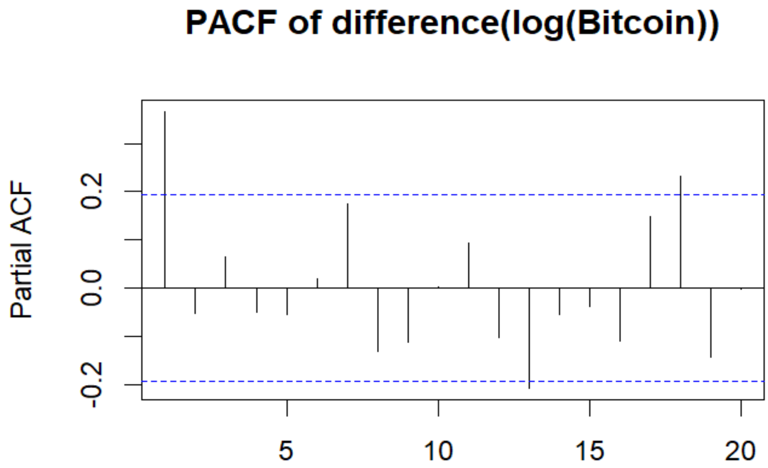


Figure 7: PACF Plot of difference(log(Bitcoin)).

From figure 7 it appears that the 1st, 13th and 18th lag partial autocorrelations terms are significant, although one of these is likely to be at random.

Seperately, figure 5 and figure 7 provide insight into what model could be used but when combining their results to gain input, given that for an AR process the ACF should decay, for an MA process the PACF should decay and for an ARMA process both should decay and neither figure 5 or figure 7 decays it is difficult to make a conclusion. The EACF will now be used and is given by figure 8:

```
> eacf(difflogbtc)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o o o o o o o o o  o  x  o
1 o o o o o o o o o o o  o  o  o
2 x o o o o o o o o o o  o  o  o
3 x o o o o o o o o o o  o  o  o
4 x x x o o o o o o o o  o  o  o
5 x o o o o o o o o o o  o  o  o
6 o o x x o o o o o o o  o  o  o
7 x x x o o o o o o o o  o  o  o
```

Figure 8: EACF Plot of difference(log(Bitcoin)).

The upper left corner of a triangle of zeros identifies the order of the ARMA model to be used. As denoted by the upper left of the triangle of zeros that originates from the second row, figure 8 suggests an AR(1). However, given that some corrolations are due to chance, it may be also worth suggesting an MA(1).

Another way of checking which model is most suitable is to use the armasubsets() function which compares multiple ARMA models with a BIC (performance measure). The output is given below by figure 9:

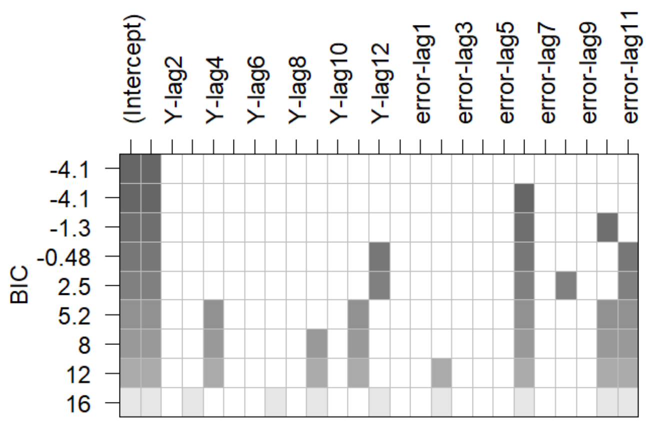


Figure 9: Output from armasubsets() for difference(log(Bitcoin)).

From figure 9, the lowest BIC score is awarded to an AR(1) model. Given that this is also the simplest model, it is what is ultimately suggested to be used by the output.

To conclude this step, an AR(1), MA(1), AR(18) and MA(18) should be applied to the data.

4 Step 5

5) Fit models to data.

As concluded in step 4, an AR(1), MA(1), AR(18) and MA(18) should be applied. Their output and models are given as follows:

```
Call:
arima(x = train2, order = c(1, 1, 0))

Coefficients:
         ar1
      0.3895
s.e.  0.0906

sigma^2 estimated as 0.03682:  log likelihood = 23.58,  aic = -45.16
```

Figure 10: Output from an AR(1) Fitted to the Differenced log(Bitcoin) Series.

```
Call:
arima(x = train2, order = c(18, 1, 0))

Coefficients:
         ar1     ar2     ar3      ar4     ar5      ar6     ar7      ar8      ar9
      0.3630  0.0422  0.0607  -0.0575  0.0523  -0.0259  0.2011  -0.0910  -0.0397
s.e.  0.0944  0.1023  0.1010   0.1030  0.1039   0.1042  0.1034   0.1054   0.1043
          ar10    ar11     ar12     ar13     ar14    ar15     ar16    ar17    ar18
      -0.0302  0.0814  -0.0143  -0.1505  -0.0106  0.0258  -0.1581  0.0657  0.2696
s.e.   0.1064  0.1054   0.1026   0.1049   0.1064  0.1047   0.1047  0.1058  0.0974

sigma^2 estimated as 0.02786:  log likelihood = 36.2,  aic = -36.4
```

Figure 11: Output from an AR(18) Fitted to the Differenced log(Bitcoin) Series.

From figure 10 and 11 it is seen that the coefficient of the Y_{t-1} component is significant (double its standard error) in both, however in figure 11 where an AR(18) is fitted the only additional term to be significant is the coefficient of the Y_{t-18} component. The AIC value (performance measure) of the AR(18) is also higher than the AIC value of the AR(1). These points together suggest that Y_{t-2} - Y_{t-17} components should not be in the model.

```
Call:
arima(x = train2, order = c(0, 1, 1))

Coefficients:
         ma1
      0.3926
s.e.  0.0892

sigma^2 estimated as 0.037:  log likelihood = 23.32,  aic = -44.64
```

Figure 12: Output from an MA(1) Fitted to the Differenced log(Bitcoin) Series.

```
Call:
arima(x = train2, order = c(0, 1, 18))

Coefficients:
         ma1     ma2     ma3      ma4      ma5     ma6     ma7     ma8     ma9    ma10
      0.3460  0.2795  0.2267  -0.0526  -0.0246  0.0968  0.3051  0.0916  0.0374  0.0087
s.e.  0.0981  0.1202  0.1170   0.1137   0.1139  0.1165  0.1171  0.1276  0.1272  0.1386
         ma11     ma12     ma13     ma14     ma15     ma16    ma17    ma18
      -0.0378  -0.0560  -0.0833  -0.0409  -0.1720  -0.3222  0.0571  0.3161
s.e.   0.1283   0.1282   0.1192   0.1193   0.1451   0.1589  0.1689  0.1253

sigma^2 estimated as 0.02581:  log likelihood = 36.57,  aic = -37.14
```

Figure 13: Output from an MA(18) Fitted to the Differenced log(Bitcoin) Series.

From figure 12 and 13 it is seen that the coefficient of the e_{t-1} component is significant (double its standard error) in both, however in figure 13 where an MA(18) is fitted all additional coefficients are not significant, only the ones of the e_{t-1}, e_{t-2}, e_{t-7}, e_{t-16} and e_{t-18} components. The AIC value of the MA(18) is also higher than the AIC value of the MA(1). These points together suggest that the e_{t-3} - e_{t-6}, e_{t-8} - e_{t-15} and e_{t-17} components should not be in the model.

Continuing from here, two new models were fitted to log(Bitcoin). The first differenced the series and has Y_{t-1} and Y_{t-18} components, this can be written as ARIMA$(1,1,0)\times(1,0,0)_{18}$ (model one). The second differenced the series and has e_{t-1} - e_{t-16} and e_{t-18} components, this can be written as ARIMA$(0,1,16)\times(0,0,1)_{18}$ (model two). The reason the first 16 components were used was because after only e_{t-1} and e_{t-18} components were used the 16th month lag was continuing to show correlation in the residual plot. It is not ideal to have a model with insignificant terms but the aim of building a time series model is to reduce the residuals to white noise and to do this the first 16 terms must be included. There is the argument that 1 in 20 terms in the acf plot are randomly outside the confidence bands but given that the coefficient of the e_{t-16} component is also significant in the MA(18) model given by figure 13 it probably isn't due to chance. The output of the two improved models are given by figures 14 and 15:

```
Call:
arima(x = train2, order = c(1, 1, 0), seasonal = list(order = c(1, 0, 0), period = 18))

Coefficients:
         ar1    sar1
      0.4222  0.3684
s.e.  0.0894  0.0909

sigma^2 estimated as 0.03114:  log likelihood = 30.78,  aic = -57.56
```

Figure 14: Output from ARIMA$(1,1,0)\times(1,0,0)_{18}$ Fitted to log(Bitcoin).

```
Call:
arima(x = train2, order = c(0, 1, 16), seasonal = list(order = c(0, 0, 1), period = 18))

Coefficients:
         ma1     ma2     ma3      ma4     ma5     ma6     ma7     ma8     ma9    ma10
      0.3751  0.2442  0.1612  -0.0560  0.0421  0.1372  0.3298  0.0870  0.0154  0.0421
s.e.  0.1007  0.1159  0.1125   0.1065  0.1116  0.1267  0.1566  0.1535  0.1388  0.1454
         ma11     ma12     ma13     ma14     ma15     ma16    sma1
      -0.0290  -0.0666  -0.1458  -0.0269  -0.2228  -0.3593  0.2810
s.e.   0.1114   0.1173   0.1425   0.1375   0.1564   0.1348  0.1079

sigma^2 estimated as 0.02696:  log likelihood = 36.21,  aic = -38.43
```

Figure 15: Output from ARIMA$(0,1,16)\times(0,0,1)_{18}$ Fitted to log(Bitcoin).

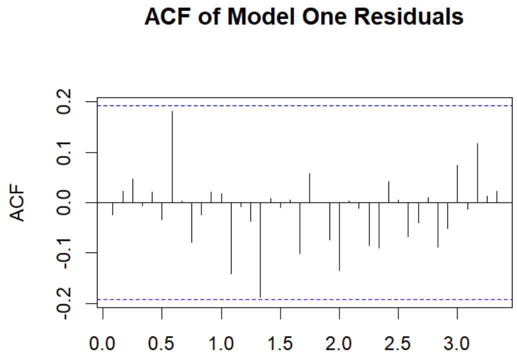
5 Step 6

6) Investigate residuals, compare models and decide on a final model.

There are five elements to checking residuals from time series models, these are:

1. Correlations in residuals
2. Normaility
3. Constant variance
4. Trends in Residuals
5. Ljung-Box Test

It is important to note that the standardised residuals are used for each step. The following two plots show the ACF of the residuals for the models and are given by figures 16 and 17:



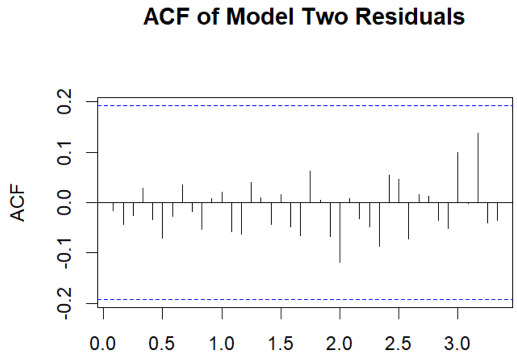
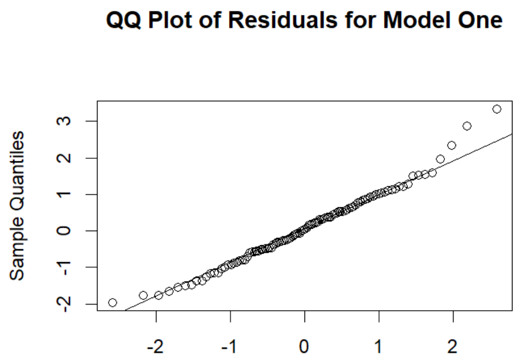


Figure 16: ACF Plot of Model One Residuals. Figure 17: ACF Plot of Model Two Residuals.

As seen from figures 16 and 17, there is no significant autocorrelation within the residuals, meaning there is now confidence the models have captured the correlation of the series and reduced the residuals to white noise.

The residual QQ plots for both models are given below by figures 18 and 19:



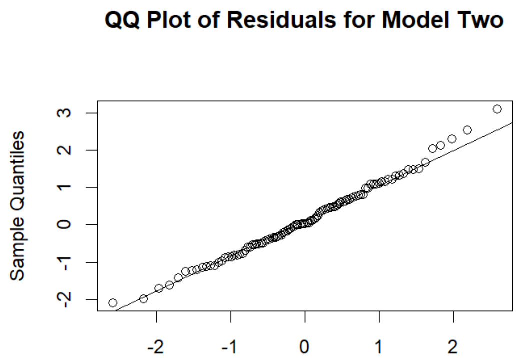


Figure 18: QQ Plot of Model One Residuals. Figure 19: QQ Plot of Model Two Residuals.

Based on both QQ plots given by figures 18 and 19, the residuals seem reasonabily normally distributed for both models. This is also supported by p values from the Shapiro-Wilks test of 0.2331 and 0.743 for model one and two respectively. For clarity, the null hypothesis of the Shapiro-Wilks test is that the sample comes from a normal distribution.

The residuals against fitted values for model one and two are given by figures 20 and 21 respectively:

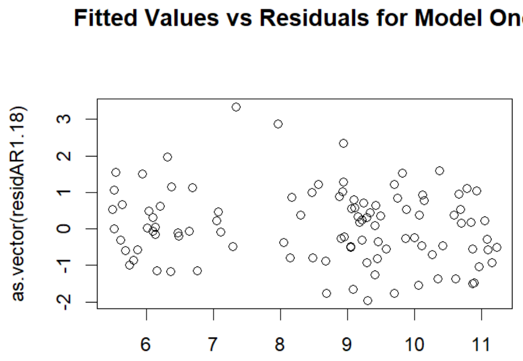


Figure 20: Fitted Values vs Residuals for Model One.

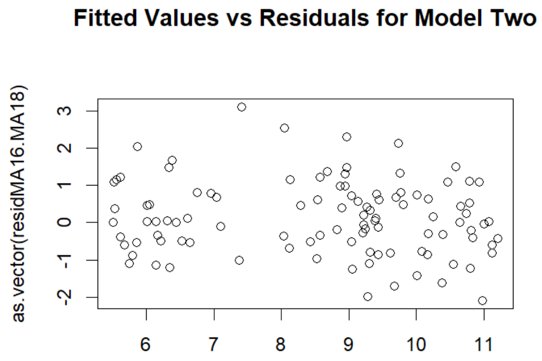


Figure 21: Fitted Values vs Residuals for Model Two.

As can be seen from figures 20 and 21, a reasonably random scatter of points can be observed, this is an indicator that the variance is constant.

Timeplots of the residuals are given by figures 22 and 23 for models one and two respectively in order to identify trend:

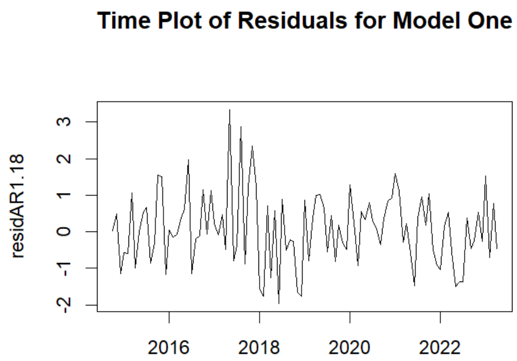


Figure 22: Time Plot of Residuals for Model One.

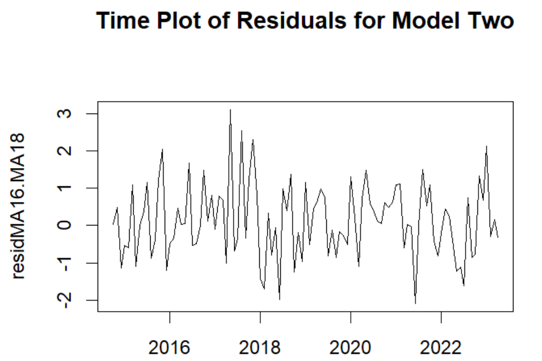


Figure 23: Time Plot of Residuals for Model Two.

As can be seen from figures 22 and 23, there is no trend in the residuals over time. There is a spike at the 35-40 month mark, however this may just be an outlier and for the purposes of the plot the residuals behave well.

The output of the Ljung-Box test for the residuals of model one and two are given by figures 24 and 25 respectively:

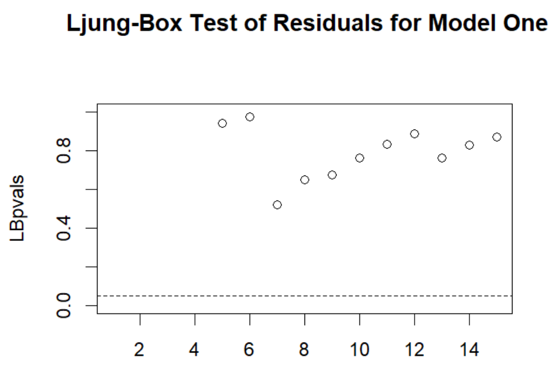


Figure 24: Ljung-Box Test of Residuals for Model One.

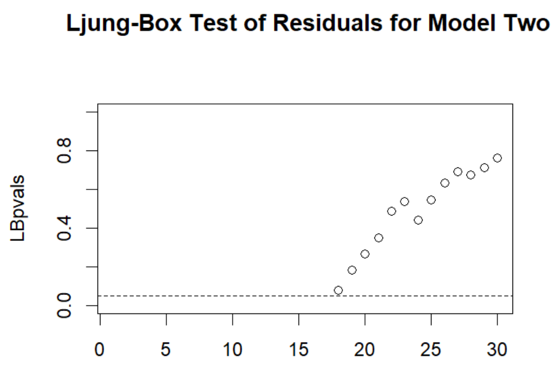


Figure 25: Ljung-Box Test of Residuals for Model Two.

The Ljung-Box test considers the magnitude of the autocorrelations as a group with a null hypothesis that there is no significant autocorrelation. As can be seen from figures 24 and 25, we fail to reject the null hypothesis for both models residuals.

The next step is overfitting the model, the aim here is to check if the current model is the best it can be. If additional terms added to either model are significant and there is a lower AIC value it is an indication that the model could be improved.

Each model was overfitted in two seperate instances. Model one was first overfitted by adding a e_{t-1} component which returned a higher AIC score of -55.89 compared to the original of -57.56. The coefficient of the e_{t-1} component in the model was also not significant. The second case of overfitting for model one was adding a Y_{t-2} component, which similarly to the first case did not improve the performance of the model. The AIC rose to -55.83 and the coefficient of the Y_{t-2} component was insignificant. Model two was first overfitted with a Y_{t-1} component which rose the AIC to -35.35 from -38.43. The coefficient of the Y_{t-2} component was also insignificant. The second case of overfitting for model two was adding a e_{t-17} component, which similarly to the first case did not improve the performance of the model. The AIC rose to -36.49 and the coefficient of the e_{t-17} component was insignificant.

It is now reasonable to assume that both models are in thier final form. So from here one must be chosen. Model one has an AIC value of -57.56 whilst model two has a higher AIC value of -38.43. Model one is also simpler that model two, for these reasons combined model one will be chosen. Model one is given by equation:

$$Y_t = \nabla log(W_t), Y_t = 0.4222Y_{t-1} + 0.3684Y_{t-18} + e_t \tag{1}$$

Where ∇ denotes differencing, W_t denotes the original series, e_t is an error term and t is time.

6 Step 7

7) Predict into the future using your model and compare these forecasts to the 10% of observations removed at the start.

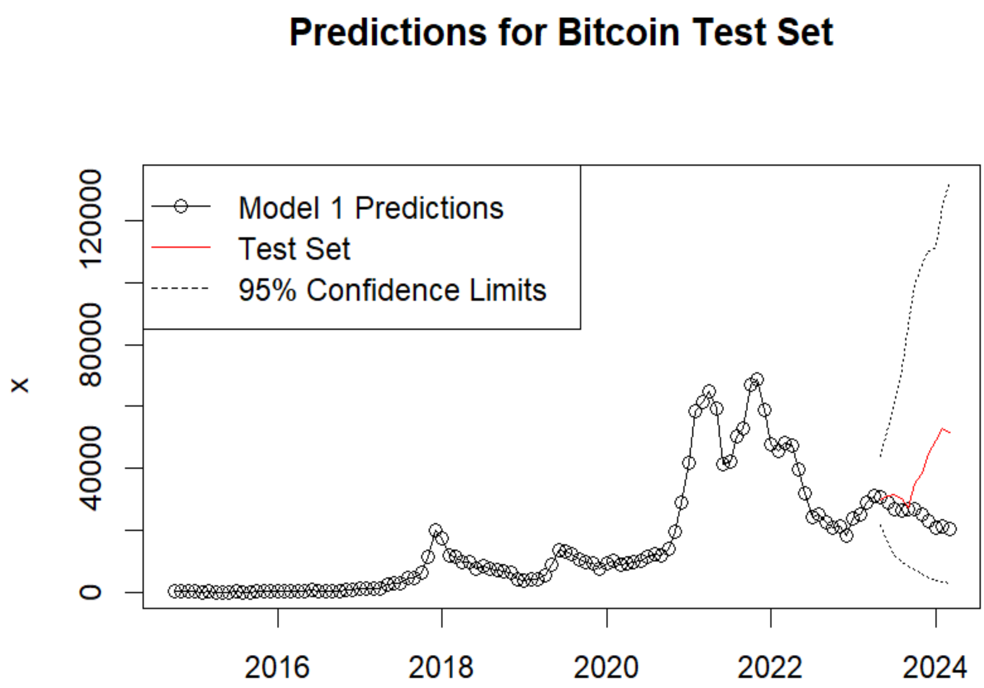


Figure 26: Predicions vs Test Set of Model One.

As can be seen from figure 26, the predicted data for the model clearly converge to the mean which is in line with the behaviour of time series models. The predictions themselves are not similar to the test data, however, the test data is inside the 95% confidence limits. This is useful as the model has correctly displayed with 95% confidence where the future data will be. The prediction limits also grow with time, this is because the series is not stationary.

7 Conclusion

After completing this report there have been some insights obtained from the price data on Bitcoins monthly highs from 10/2014. Firstly, there appears to be rising trend and increasing variance in the data. This is not surprising considering Bitcoin sits at higher prices every roughly four years and is known to be an incredibly volatile asset. Secondly, the best model that also reduced the residuals to white noise was an ARIMA$(1, 1, 0)\times(1, 0, 0)_{18}$ meaning the current price of Bitcoin, after treating the variance via a log transformation and the trend via differencing, is related to its previous one month and eighteen month prices.