Time Series Final Project - Forecasting Bitcoin Prices

Jordi Solé Casaramona

June 10^{th}, 2020

1 Introduction

The Bitcoin cryptocurrency is a decentralized digital currency, that unlike currencies like the Dollar or the Euro, is not backed up by any central bank. It works using block-chain technology to ensure the anonymity of the transaction as well as to ensure the protection of peer-to-peer transactions.

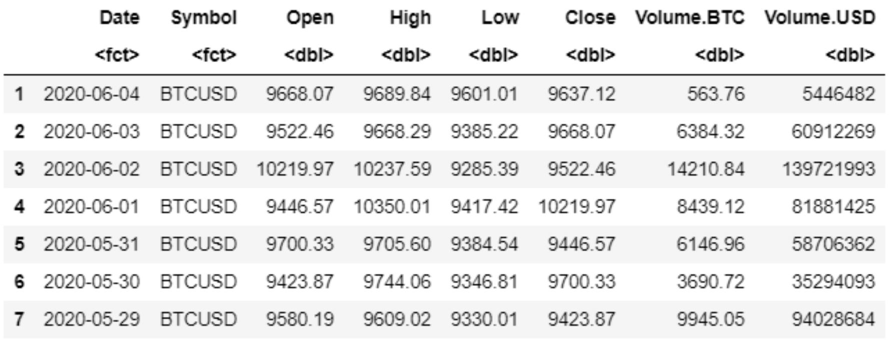
For the last recent years, Bitcoin has been given the name as *The best asset of the decade* by Bloomberg and others. This have raised the interest of people in Bitcoin and many have tried to predict its value in order to win money in short and long term trading.

In this analysis we are going to try to fit the best model possible and give some future price prediction of the asset by using the tools learned in Time Series class.

2 Load the data set

The dataset used in this project is based on the Bitcoin (BTC) value vs USD extracted from the well known cryptocurrencies trading platform Coinbase. The data starts from December the 1^{st} 2014 until when the data was extracted in June the 4^{th} 2020. Thus, the data contains information about 2001 days and has 8 different columns.

These columns of the dataset are the movements of Bitcoin stock (Open, High, Low and Close) in a daily manner can be obtained, as well as the BTC and USD volume for that day. See a short snippet on how the data looks:



	Date	Symbol	Open	High	Low	Close	Volume.BTC	Volume.USD
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2020-06-04	BTCUSD	9668.07	9689.84	9601.01	9637.12	563.76	5446482
2	2020-06-03	BTCUSD	9522.46	9668.29	9385.22	9668.07	6384.32	60912269
3	2020-06-02	BTCUSD	10219.97	10237.59	9285.39	9522.46	14210.84	139721993
4	2020-06-01	BTCUSD	9446.57	10350.01	9417.42	10219.97	8439.12	81881425
5	2020-05-31	BTCUSD	9700.33	9705.60	9384.54	9446.57	6146.96	58706362
6	2020-05-30	BTCUSD	9423.87	9744.06	9346.81	9700.33	3690.72	35294093
7	2020-05-29	BTCUSD	9580.19	9609.02	9330.01	9423.87	9945.05	94028684

Figure 1: First rows of the Bitcoin dataset.

3 First analysis

3.1 Plot the data

The next step in understanding the time series is simply plotting the data. For the price of the asset each day we pick the closing price:

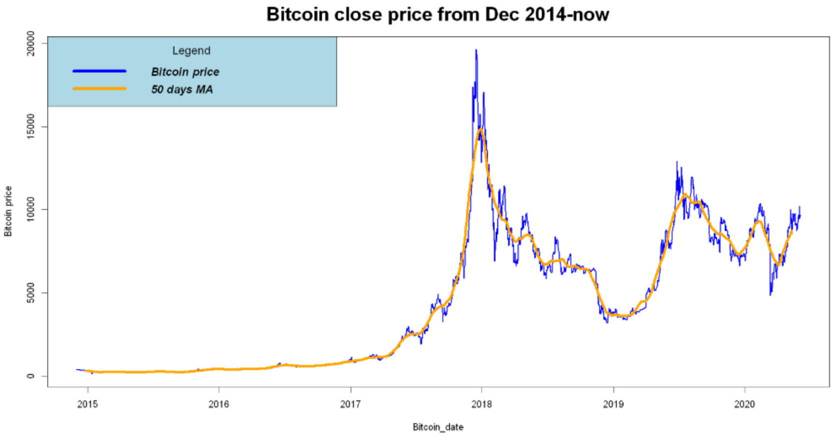


Figure 2: Bitcoin Closing prices of Bitcoin from December 2014 to the present and a 50 days Moving Average line.

In the above plot we see the full historical data from Bitcoin trading platform closing prices of Bitcoin. In orange, we can observe the 50 days Moving Average line that gives us a clearer view of the mid-term trends of the time series.
From the plots we can see some key points:

1. The **trend** varies depending on different time intervals. From the drop for Covid-19 outbreak in March 13^{th}, the Bitcoin price has experienced an upwards trend. The 50 days moving average help to better show the different trends.

2. The **volatility** of the market can be depicted as high. From September 2017 to January 2018 the price of Bitcoin increased from \$5000 to more than \$19.000. That is a 280% increase of the price in less than a quarter.
3. The **variance** of the time series is not always the same. From 2015 to 2017 the price is quite stable while the market capital was low. While from January 2018 to January 2019 the variance has been really high until the drop in November 2018.
4. There is no visible **seasonality** in the charts of Bitcoin.

Some further considerations:

- Because we see that Bitcoin prices have two different time periods with low market capital and low volatility from December 2014 to 2017 and a high volatility and market cap. from early 2017 on wards, the analysis will start from January the 1^{st} 2017.
- Because we are talking of monetary assets, it is a good practice to performing the log of the prices.

This changes are done in sake of a better model and analysis.
After applying the above points, the new data is the following:

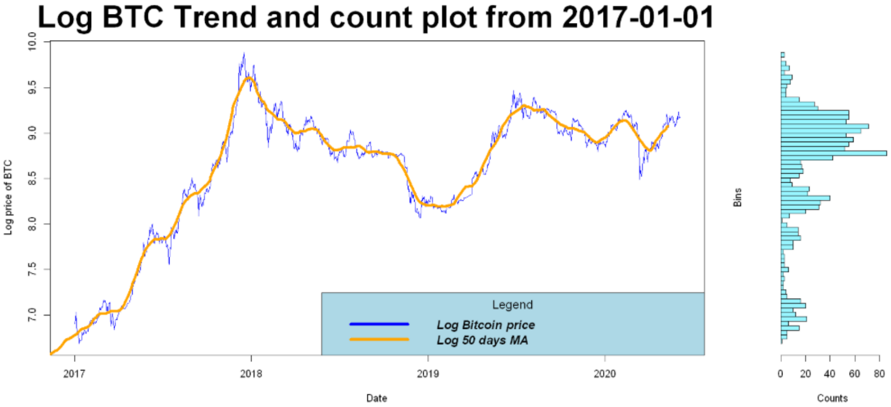


Figure 3: Log Bitcoin price (blue), 50 days MA (orange) and a bar plot of the count of Close Price values.

When we look at the log prices plot, we can see more clearly that the data from 2018 on wards has a support *(financial language)* around 8.9 or \$7330 with a bell shaped distribution around this value.

3.2 Seasonal, trend and residual components

The next step of the analysis is to focus on the seasonal, trend and residual components of the analyzed Bitcoin time series. We use the *stl* command in R to decompose a time series into seasonal, trend and irregular components, and this are the results:

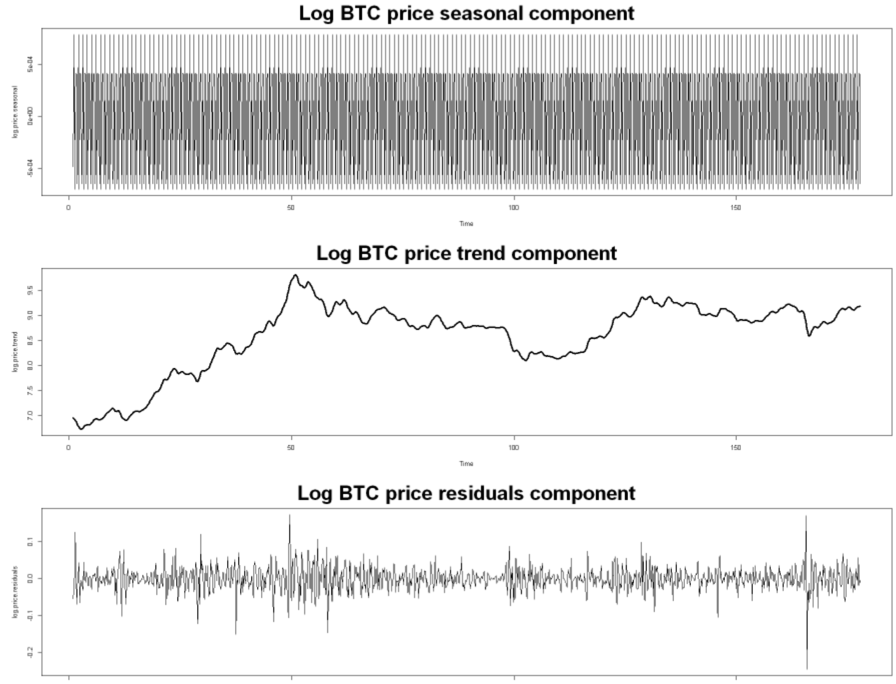


Figure 4: Log Bitcoin price seasonal, trend and residual components.

The **seasonality** plot looks like a white noise that seems constant with mean 0. Thus, showing no seasonality component is clearly observed in the plot.
For the **trend**, it seems there has been an upwards trend since 2017 that now is stabilizing.
When looking at the **residuals** it resembles a white noise, because it has a constant mean of 0, and possibly a constant variance.

3.3 ACF and PACF

Now lets see the Autocorrelation function and the Partial autocorrelation function to gain more insights of the time series:

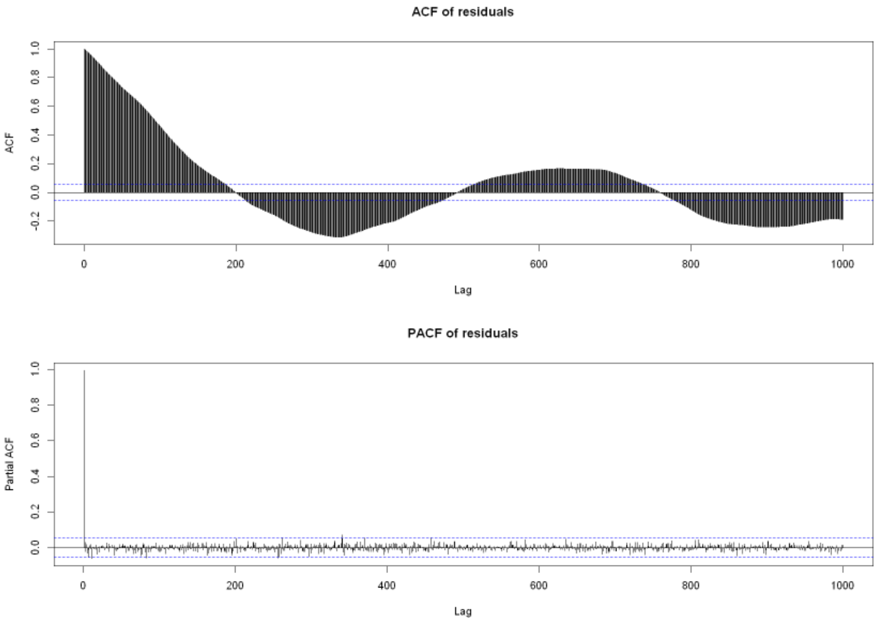


Figure 5: ACF and PACF of the Log Bitcoin price.

From the plot we can say that the gradual decay of the **ACF** that shows the high correlation among successive points and suggest that the duration of shocks is relatively persistent and influence the data several observations ahead. This is a strong evidence of an existence of a trend that was expected from the Bitcoin price plot. Looking at the **PACF** plot, we see only one significant correlation on the plot. Both the **ACF** and **PACF** suggest this data is taking a Random Walk, so a transformation should be applied to achieve the stationary of the series.

4 Box Cox Transformation

Box Cox Transformation

$$\begin{cases} y = \dfrac{x^{\lambda} - 1}{\lambda} \text{ where } \lambda \neq 0 \\ y = \ln x \text{ where } \lambda = 0 \end{cases}$$

Figure 6: Box Cox Transformation formula.

A Box Cox transformation is a way to transform non-normal dependent variables into a normal shape. Normality is an important assumption for many statistical techniques; if your data that is not normally distributed, applying a Box-Cox means that you are able to run a broader number of tests. With this we try to achieve a even more constant variance and helping to normalize the data.

This method finds a value λ which maximizes the log-likelihood and then the data is raised to the power of λ. The higher the likelihood, the better (more likely) the model parameter. To do this transformation, we use the *BoxCox()* command.

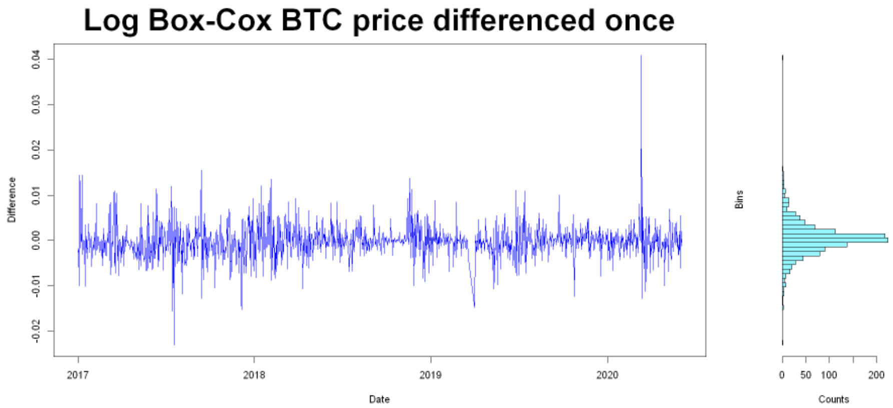
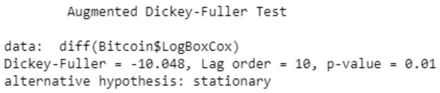


Figure 7: Log Bitcoin price with Box Cox Transformation and a first difference.

After the transformation, the data now has a smaller variance (about 0.01) and a mean of 0. Thus, the trend has been nullified by the difference. From the bar plot on the right we see that the distribution of the Log Box-Cox BTC price follows a normal distribution with mean 0. Let's now check the ACF and PACF together with the Augmented Dicky-Fuller test to test stationary.



```
	Augmented Dickey-Fuller Test

data:  diff(Bitcoin$LogBoxCox)
Dickey-Fuller = -10.048, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

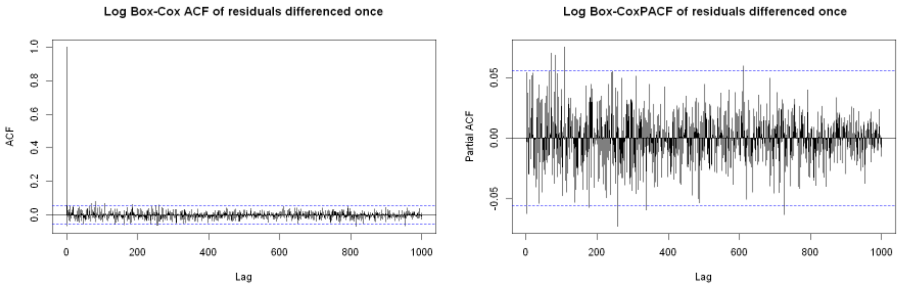


Figure 8: ACF and PACF of Log Bitcoin price with Box Cox Transformation and a first difference. Above, the Augmented Dicky-Fuller test.

The p-value obtained in the Augmented Dicky-Fuller test is 0.01, being lower than confidence level at 0.05 gives. Hence, we can reject the null hypothesis and affirm that the data is stationary after the difference.

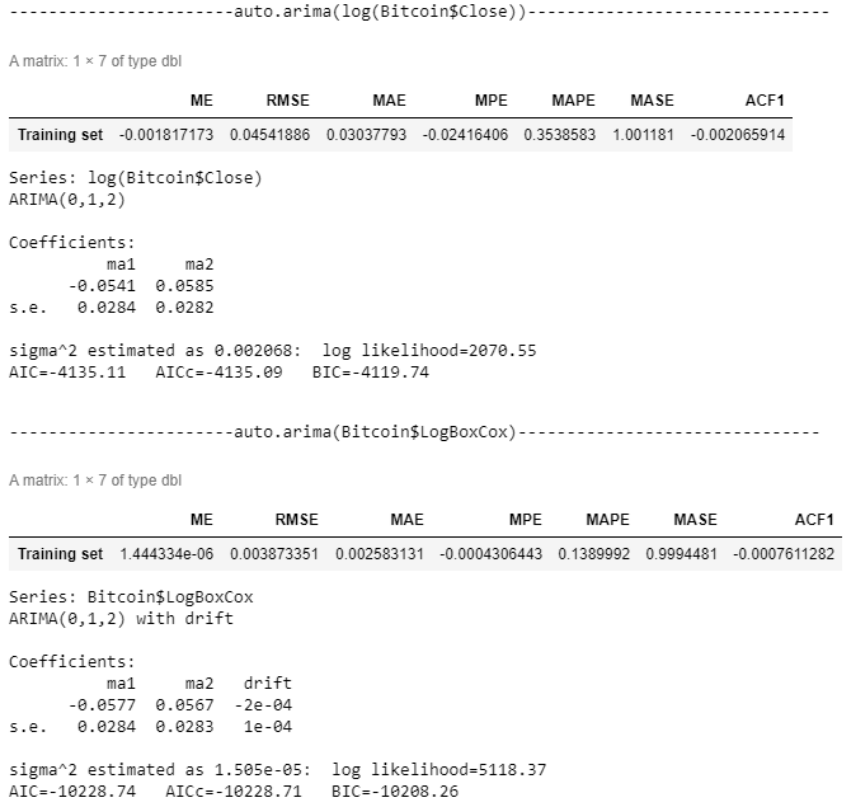
After checking that the data is indeed stationary, we focus on the **ACF** and

PACF plots. It can be seen from the **ACF** plot that it has only one significant lag, and can suggest that a moving average term is in the data. For the **PACF** very few lags are significant from an observation of 1000.

5 Fit model and goodness of fit

5.1 ARIMA model

In this section we aim to fit the best model to our Bitcoin time series. To do that, we first used the auto.arima() function to get the best ARIMA model possible from the data with the lowest Akaike Information Criterion (AIC).
In order to check the effectiveness of the Box Cox transformation, two auto.arima models are going to be fitted. On with the log close prices of Bitcoin and the other one, with this same data but after the Box Cox model. This are the results:



```
----------------------auto.arima(log(Bitcoin$Close))------------------------------
```

A matrix: 1 × 7 of type dbl

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	-0.001817173	0.04541886	0.03037793	-0.02416406	0.3538583	1.001181	-0.002065914

```
Series: log(Bitcoin$Close)
ARIMA(0,1,2)

Coefficients:
          ma1     ma2
      -0.0541  0.0585
s.e.   0.0284  0.0282

sigma^2 estimated as 0.002068:  log likelihood=2070.55
AIC=-4135.11   AICc=-4135.09   BIC=-4119.74
```

```
----------------------auto.arima(Bitcoin$LogBoxCox)------------------------------
```

A matrix: 1 × 7 of type dbl

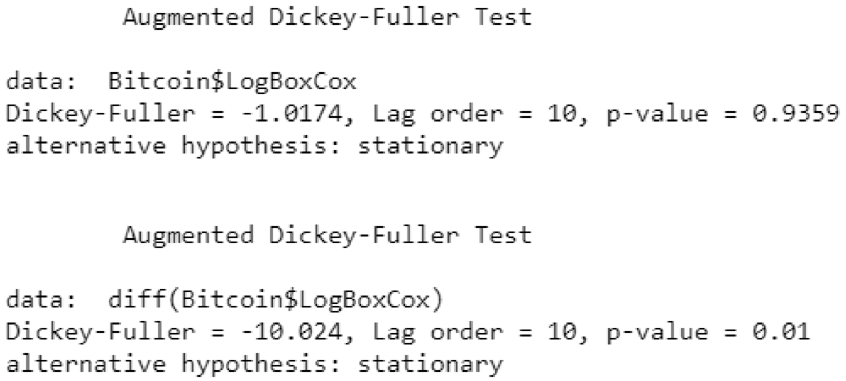
	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	1.444334e-06	0.003873351	0.002583131	-0.0004306443	0.1389992	0.9994481	-0.0007611282

```
Series: Bitcoin$LogBoxCox
ARIMA(0,1,2) with drift

Coefficients:
          ma1     ma2   drift
      -0.0577  0.0567  -2e-04
s.e.   0.0284  0.0283   1e-04

sigma^2 estimated as 1.505e-05:  log likelihood=5118.37
AIC=-10228.74   AICc=-10228.71   BIC=-10208.26
```

Figure 9: Auto ARIMA of Bitcoin Log Close prices and the same data after the Box Cox transformation.

We see that before the selected ARIMA model, we see a grey bar displayed using the *accuracy()* function. This shows that the values for the errors of the

data with the Box Cox transformation is much lower in all of the indicators. A sign that the transformation works.
Moving on, we see that for both auto ARIMA, the selected models are ARIMA(0,1,2), with the only difference that for the data with the Box Cox transformation it has a small negative drift component. This drift is obtained becaues the difference is 1, ($d = 1$) and it shows that there is a trend with slope $\mu = -2e^{-04}$.
Last, we see that the AIC and BIC scores, as well as the log likelihood of the ARIMA model fitted to the Box Cox data is better than the fitted data without the transformation. Hence we select this model.
Lastly, we use the Augmented Dickey-Fuller test to check trend-stationary.



```
        Augmented Dickey-Fuller Test

data:  Bitcoin$LogBoxCox
Dickey-Fuller = -1.0174, Lag order = 10, p-value = 0.9359
alternative hypothesis: stationary


        Augmented Dickey-Fuller Test

data:  diff(Bitcoin$LogBoxCox)
Dickey-Fuller = -10.024, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Figure 10: Augmented Dicky-Fuller Tests with the data fitted to the model and with the differenced once data.

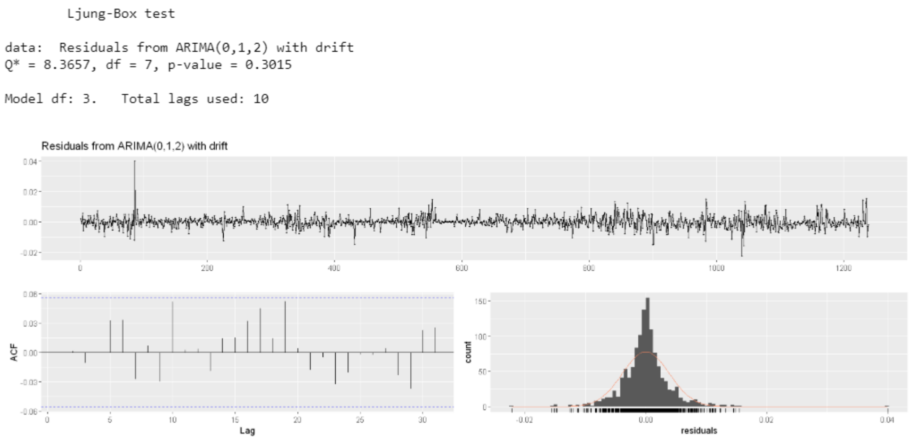
Because the p-value is 0.95 we can't reject the null hypothesis and we affirm that the time series is not stationary. But if we difference once the data, the p-value drops to 0.01. Now we can reject the null and say that the time series is stationary if differenced once ($d = 1$) to remove the stochastic trend.

5.2 GARCH model

Note that a fit of a GARCH model was also tried because, form the notes we know that GARCH models help to describe financial markets in which volatility and variance. But the result obtained after fitting a GARCH(1,1) model was an Akaike information criterion of 5060.818. Very far from the AIC=-10228.74 obtained with the ARIMA(0,1,2) with drift.

5.3 Checking the residuals

The following step is to check the residuals to see if the above ARIMA(0,1,2) is a good model. to check this, we will use the *checkresiduals()* function that produces a time plot of the residuals, the corresponding ACF, and a histogram.



```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,2) with drift
Q* = 8.3657, df = 7, p-value = 0.3015

Model df: 3.   Total lags used: 10
```

Figure 11: Checkresiduals() for the ARIMA(0,1,2) model with drift.

First, we see the Ljung-Box test of the residuals of the model. Because the p-value is 0.3015, the null hypothesis can't be rejected an thus, the data are independently distributed.
The first plot we see the is for the residuals from the fitted model. The residuals seem to be similar to a IID noise with a constant mean an almost constant and small variance.
On the bottom left we see the ACF from the residuals. We also see that it looks like a white noise because there is no significant lag.
Last, in the bottom right we see that the residuals follow a kind of close to normal distribution with a high concentration with residuals of value 0.

5.4 Q-Q Plot

We use of the Q-Q plot to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal.

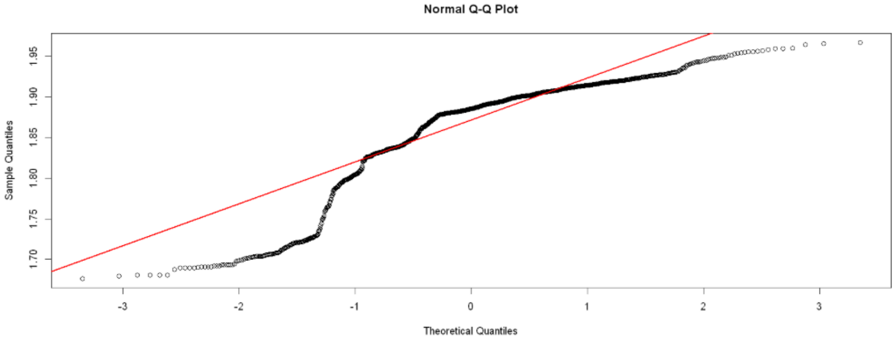


Figure 11: Q-Q Plot of the Bitcoin price data.

The values for the processes seem to deviate in all cases from the normal distribution (red line) in the tails, and thus, the data doesn't seem to come from a normal distribution.

6 Forecasting

Finally, after we obtained the ARIMA model and performed all the test to ensure that its a good fit we can proceed to predict the value of Bitcoin.

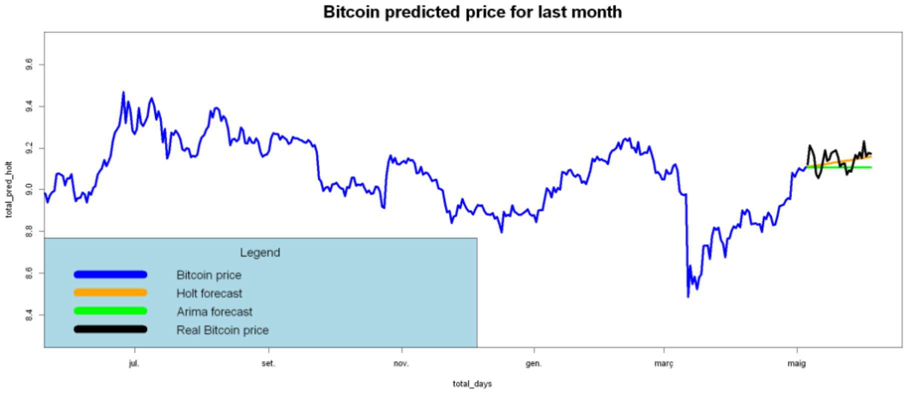


Figure 12: Bitcoin price prediction of the following 30 days.

+In the plot above we can see the Log Bitcoin price for the last year. In here, the blue line represents the real price obtained from the Bitcoin platform. From 06/05/2020 to 04/06/2020, a prediction was performed for the 30 next days was performed:

- The first method, in orange is the Holt-Winters forecasting method, also known as linear exponential smoothing was applied. The model predicts a current or future value by computing the combined effects of value, trend, and seasonality.
- The second method, in green, is calling the forecast function with the previously fitted ARIMA model.
- Finally, the black line represents the true price of Bitcoin this last 30 days to check the different predictions visually.

As we can see, the forecasting that better fits the real data is the Holt-Winters forecasting method (orange line).

7 Conclusions

From the analysis above we can say that a complex trend, that can be described as a Random Walk as seen from the ACF and PACF.

We saw that there is two differentiated periods on Bitcoin price history and that for the sake of a better model, we decided to focus on the data from January 2017 onwards where the data has more volatility and the market capital is higher.
We have also proved that the Box Cox transformation has been beneficial in order to obtain a better fitted model that resulted to be an ARIMA(0,1,2) with a small negative drift of $\mu = -2e^{-4}$. This could tell us that the Bitcoin price could be broadly modeled as:

$$\text{Future price} = \text{Current price} + \text{MA}(2) - 2e^{-4} + RandomNoise$$

Meaning that the price is influenced by the two previous periods and that there is a downwards trend in the Bitcoin price according to the ARIMA(0,1,2) with drift model.
Regarding other models, more advanced GARCH models could be applied in the future in order to try to obtain a better result than the fitted GARCH(1,1) model.
Lastly, we have forecast accurately the Bitcoin price of the next 30 days using the Holt-Winters forecasting method that gave a better result than the ARIMA forecasting method.