Predicting Bitcoin and Etherium prices using ARIMA and VAR

Michał Szałanski

Abstract

For predicting Bitcoin and Etherium prices both ARIMA and VAR models seem to be good choices. ARIMA was slightly weaker with sMAPE of *4.3%*, compared to *3.4%* achieved by the VAR model. It was observed that, for the examined period of time, the Bitcoin and Etherium prices were cointegrated.

Aim of the study

This study tries to create optimal models for predicting Bitcoin and Etherium prices by testing currency cointegration and creating optimized ARIMA and VAR models.

Data

This study will be conducted on the daily closing prices of two cryptocurrencies - Bitcoin and Etherium. Bitcoin was chosen because it is a dominant force on the market, Etherium was chosen because it seems to follow Bitcoin price very closely.

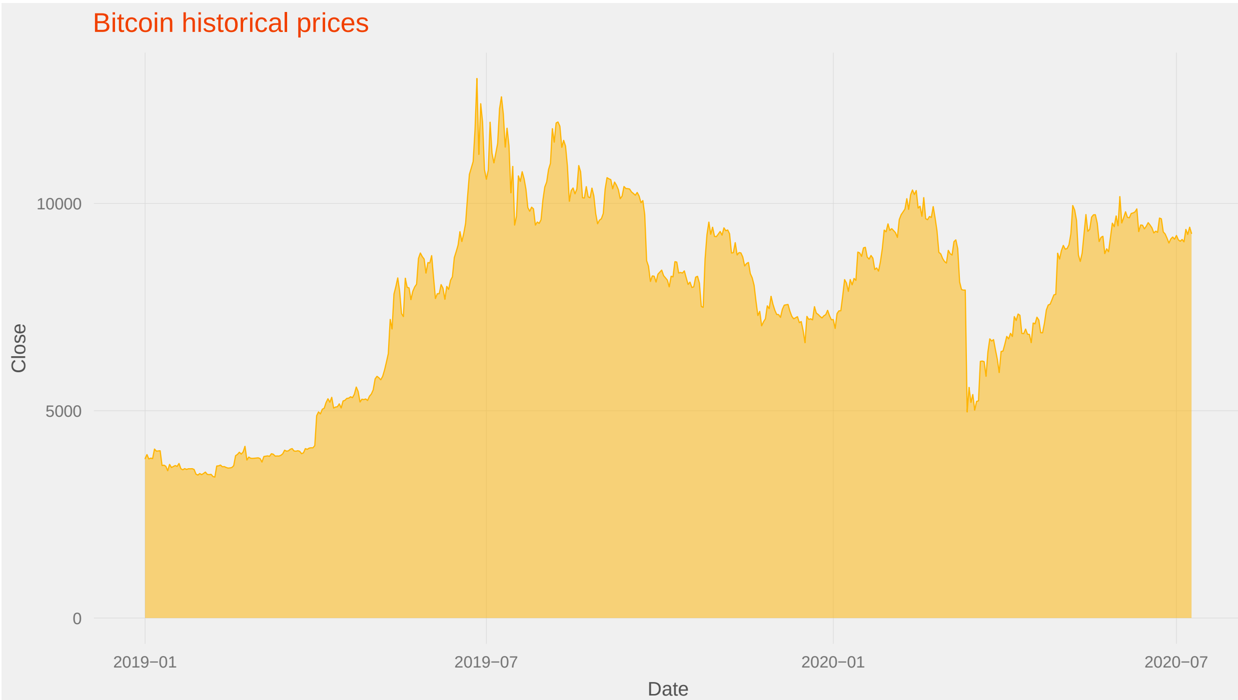
The dates are from 2019-01-01 to today. This gives enough data to estimate the models, while at the same time removes the one-of-a-kind huge price surge that happened in 2018.

Out of sample period for estimation was chosen to be 30 days. It should be enough to test the models, but not too long as to give increased errors.

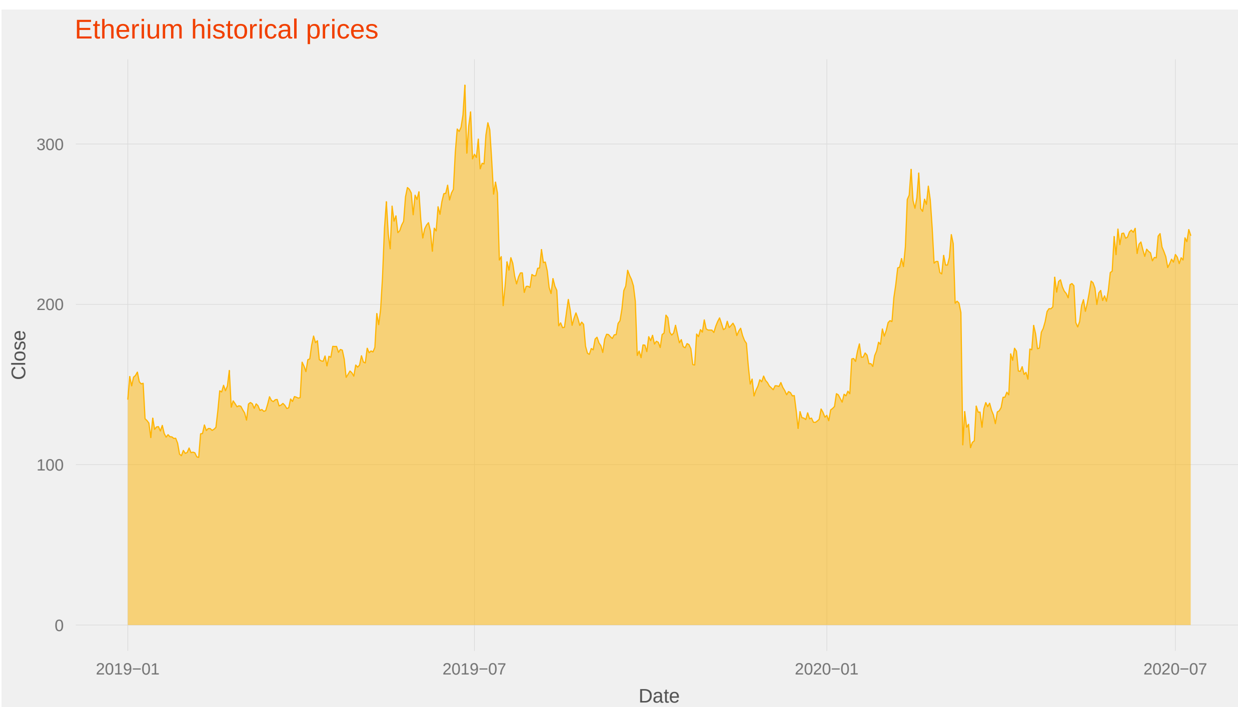
The data structure is as follows. It was downloaded from coinmarketcap.com and saved into csv files.

```
# A tibble: 6 x 7
  Date        Open  High   Low Close      Volume    MarketCap
  <date>     <dbl> <dbl> <dbl> <dbl>       <dbl>        <dbl>
1 2020-07-04 9084. 9183. 9054. 9132. 12290528515 168251076678
2 2020-07-05 9126. 9162. 8977. 9074. 12903406143 167181726154
3 2020-07-06 9073. 9375. 9059. 9375. 17889263252 172746103840
4 2020-07-07 9349. 9361. 9202. 9252. 13839652595 170485472276
5 2020-07-08 9253. 9450. 9250. 9428. 19702359883 173738543115
6 2020-07-09 9428. 9431. 9235  9278. 18000702524 170977231638
```

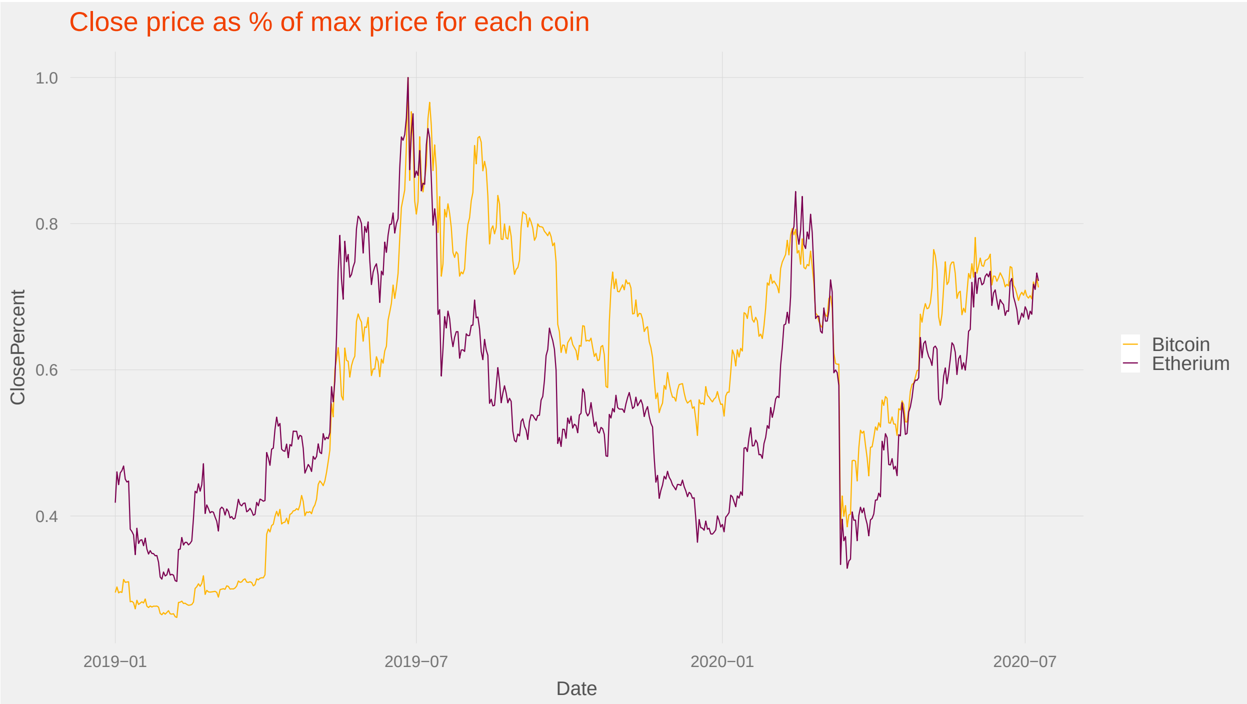
Bitcoin prices exhibit significant changes in price in both directions, with no visible trend. None of the changes come close to these observed in 2018.



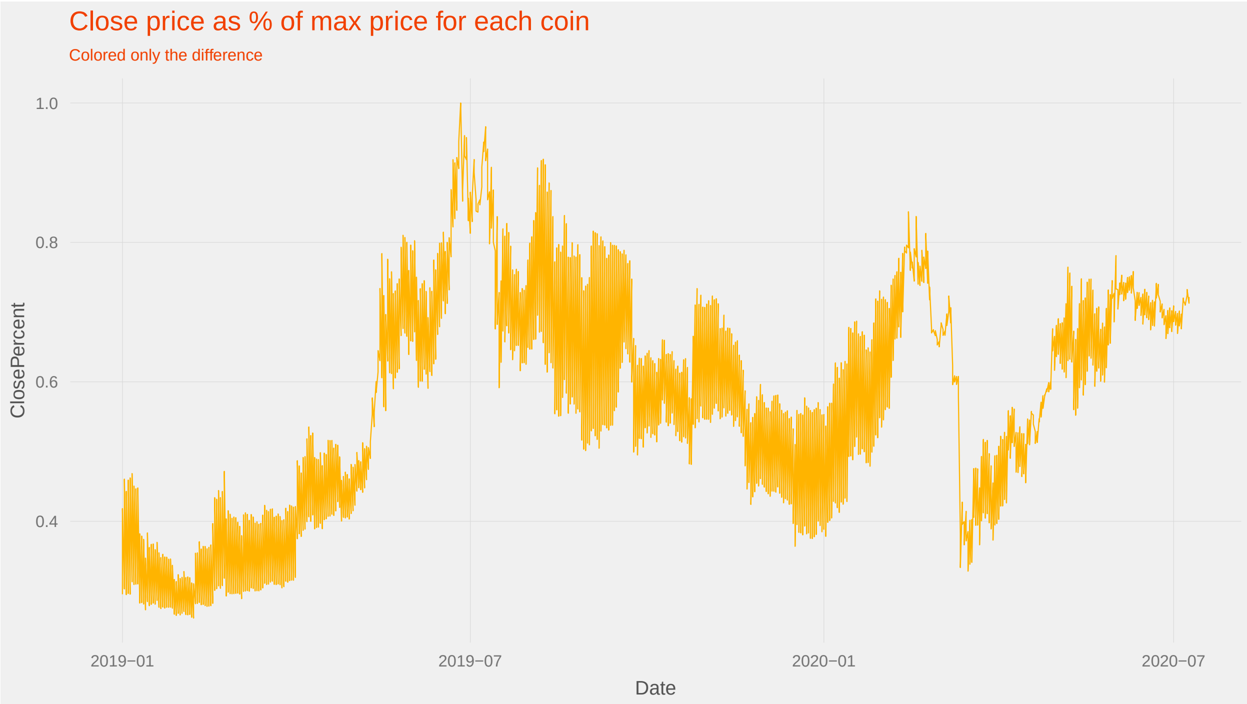
Etherium prices looks very similar to Bitcoin, although they seem to have slightly higher variance.



Plotting both prices as % of a max price confirms this. Etherium reacts to changes stronger than Bitcoin. They do however seem very cointegrated.



This is the same plot, but as a difference between prices. As it can be seen, when the change is very short and strong (e.g. around 2020-03), the prices move like one. This could be a result of automated trading, but more in-depth research would be required.



ARIMA - Bitcoin

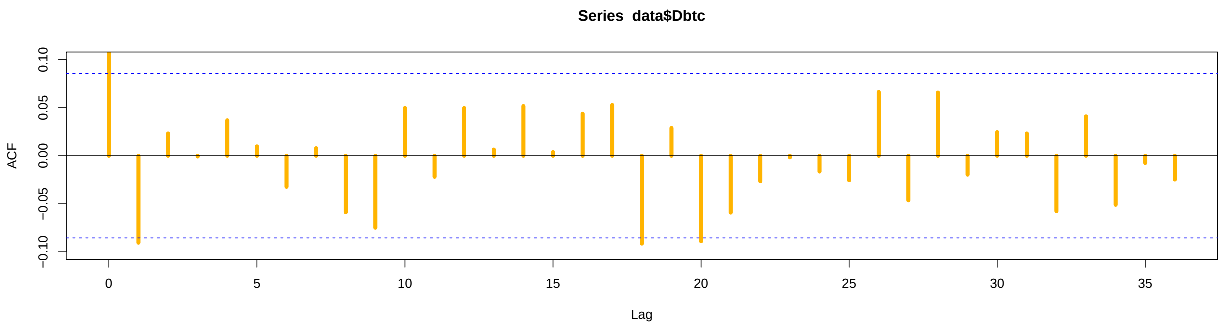
Breusch–Godfrey test signifies a autocorrelation of order 1, so at least one augmentation is needed.

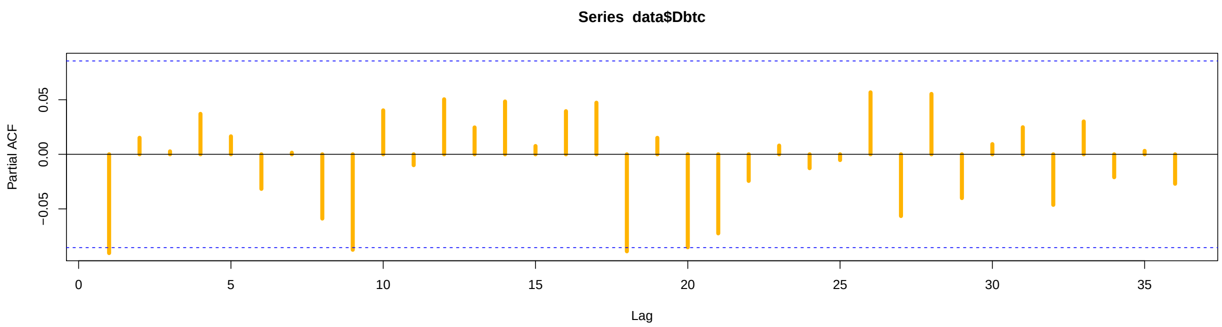
	augmentations	adf	p_adf	bgodfrey	p_bg
1	0	-1.893110	0.3530431	3.800753e+00	0.05122955
2	1	-1.806775	0.3853181	1.237688e-03	0.97193559
3	2	-1.828925	0.3770376	1.793005e-05	0.99662146
4	3	-1.840593	0.3726757	4.323380e-06	0.99834098
5	4	-1.844074	0.3713746	2.047855e-04	0.98858240
6	5	-1.878033	0.3586797	1.261465e-04	0.99103875

After differencing, all Breusch–Godfrey tests are insignificant. This means that the price of Bitcoin is a I(1) process.

	augmentations	adf	p_adf	bgodfrey	p_bg
1	0	-25.019339	0.01	1.240213e-03	0.9719070
2	1	-16.584823	0.01	1.062229e-06	0.9991777
3	2	-13.350334	0.01	5.285389e-06	0.9981657
4	3	-11.074898	0.01	1.453466e-04	0.9903810
5	4	-9.775438	0.01	2.321714e-04	0.9878430
6	5	-9.246796	0.01	7.814462e-07	0.9992947

Some lags seem to be on the verge of being significant. Thanks to the power of modern PC, running *auto.arima* on max parameters of (240, 1, 240) takes only a minute.

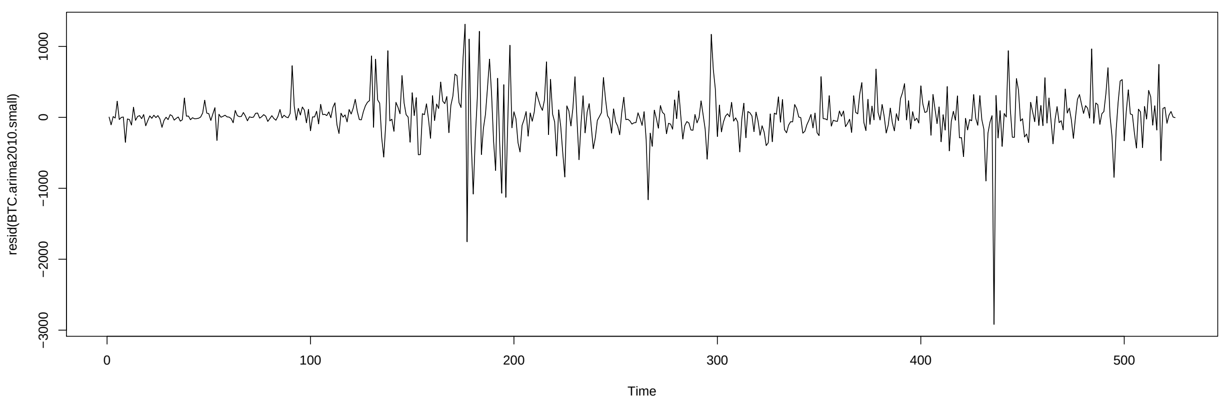




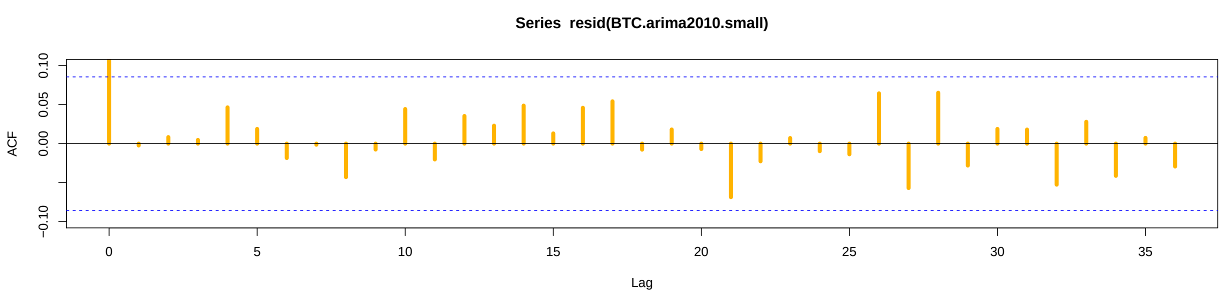
This is the final ARIMA(20, 1, 0) model, estimated after auto.arima and iterative removal of insignificant variables.

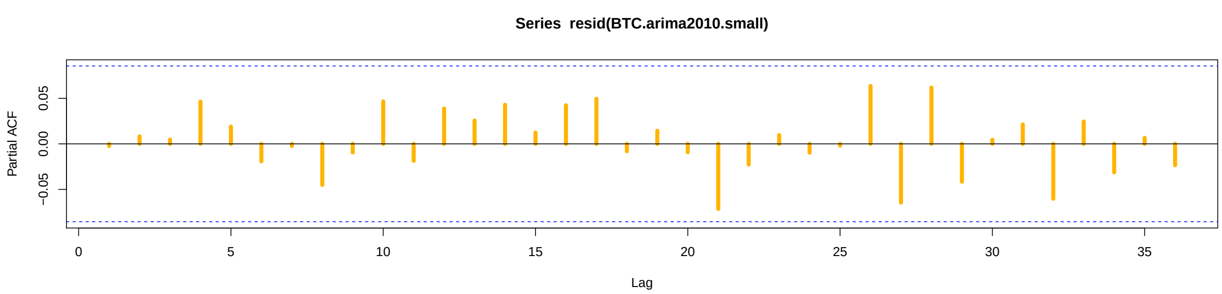
```
z test of coefficients:

      Estimate Std. Error z value Pr(>|z|)
ar1  -0.086464   0.043097 -2.0063  0.04483 *
ar9  -0.087163   0.043332 -2.0115  0.04427 *
ar18 -0.090388   0.043117 -2.0963  0.03605 *
ar20 -0.084343   0.042945 -1.9640  0.04953 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residuals have a mean of zero, but the variance is not constant. This is due to the changes in variance of the underlying variable.



Nothing is significant, which means that the residuals are not autocorrelated.





BJ test confirms that the residuals are not autocorrelated.

```
 Box-Ljung test

data:  resid(BTC.arima2010.small)
X-squared = 3.6453, df = 10, p-value = 0.9619
```

ARIMA - Etherium

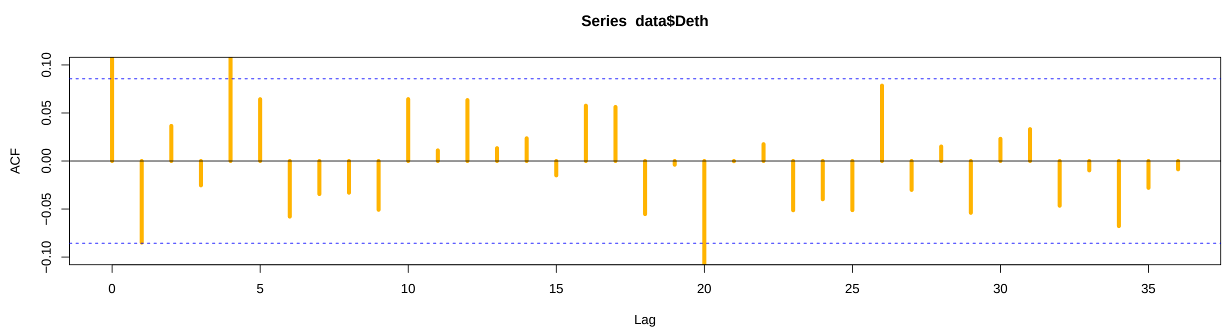
Judging by the Breusch–Godfrey test, Etherium prices are a I(1) process, so one augmentation should be enough.

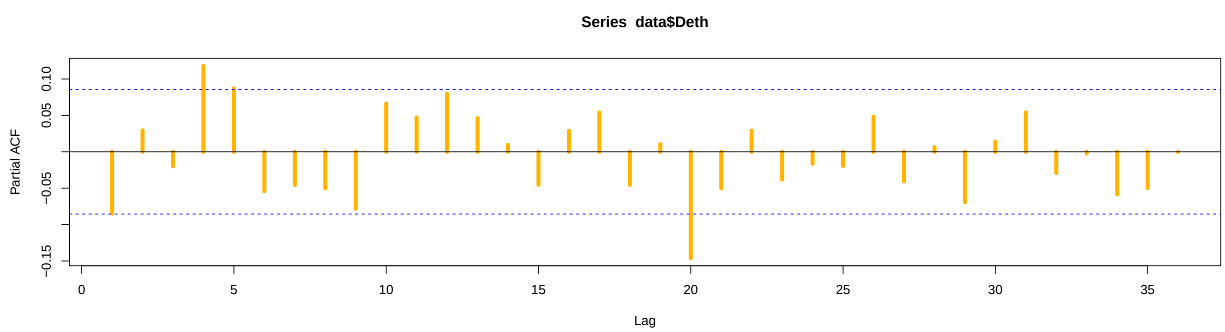
	augmentations	adf	p_adf	bgodfrey	p_bg
1	0	-2.065988	0.2884159	2.835975e+00	0.09217482
2	1	-1.900500	0.3502808	5.919384e-03	0.93867324
3	2	-1.955272	0.3298053	6.783775e-05	0.99342840
4	3	-1.911014	0.3463500	1.262208e-03	0.97165907
5	4	-2.189239	0.2423406	8.225769e-02	0.77426085
6	5	-2.446873	0.1460289	7.444745e-03	0.93124150

Testing the variable differenced once confirms this. Breusch–Godfrey is not significant here for 0 augmentations, which means that there is no autocorrelation in the new variable.

	augmentations	adf	p_adf	bgodfrey	p_bg
1	0	-24.932503	0.01	0.0086556146	0.9258753
2	1	-16.300985	0.01	0.0006124295	0.9802565
3	2	-13.518990	0.01	0.0025725328	0.9595485
4	3	-10.308620	0.01	0.0552412058	0.8141820
5	4	-8.589309	0.01	0.0117453610	0.9136974
6	5	-8.446794	0.01	0.0038156758	0.9507451

Some lags seem to be significant, so *auto.arima* was run on max parameters of (240, 1, 240).

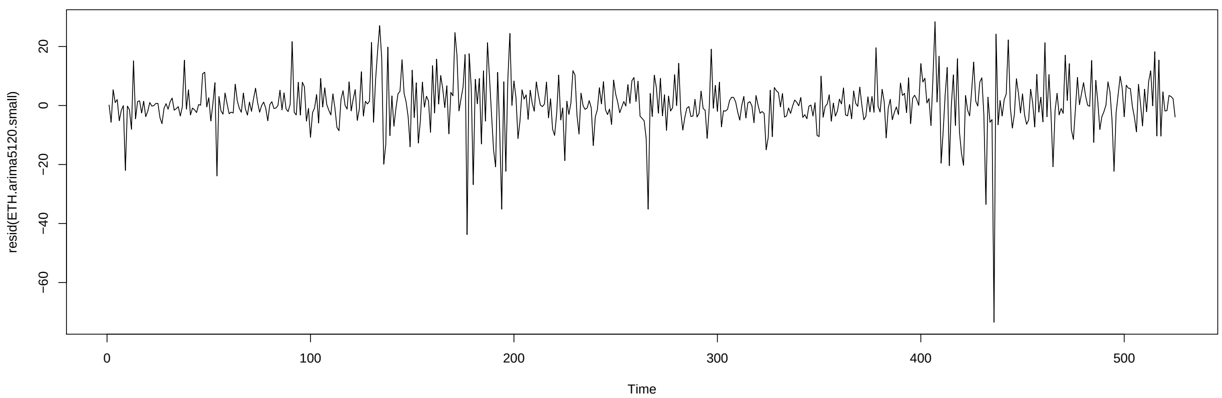




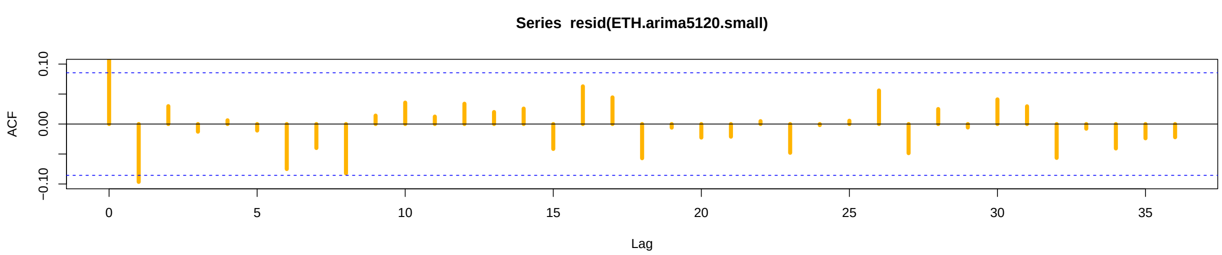
This is the final model, taken form auto.arima, with insignificant variables iteratively removed. It is ARIMA(5, 1, 20).

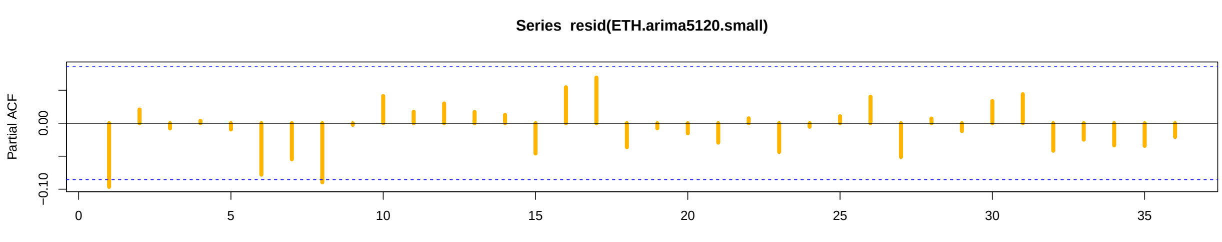
```
z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)
ar4   0.400138   0.087349  4.5809 4.630e-06 ***
ar5   0.460937   0.084462  5.4573 4.834e-08 ***
ma4  -0.270363   0.088343 -3.0604  0.002211 **
ma5  -0.388580   0.083880 -4.6325 3.612e-06 ***
ma9  -0.177681   0.047009 -3.7797  0.000157 ***
ma20 -0.152472   0.036982 -4.1228 3.742e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residuals have a mean of 0, but they are not stationary. This is because of the changes in variance. This is due to the original data that has significant spikes in price in some places, so nothing more can be done about it.



Everything is insignificant, or on the verge of significance, which is good news.



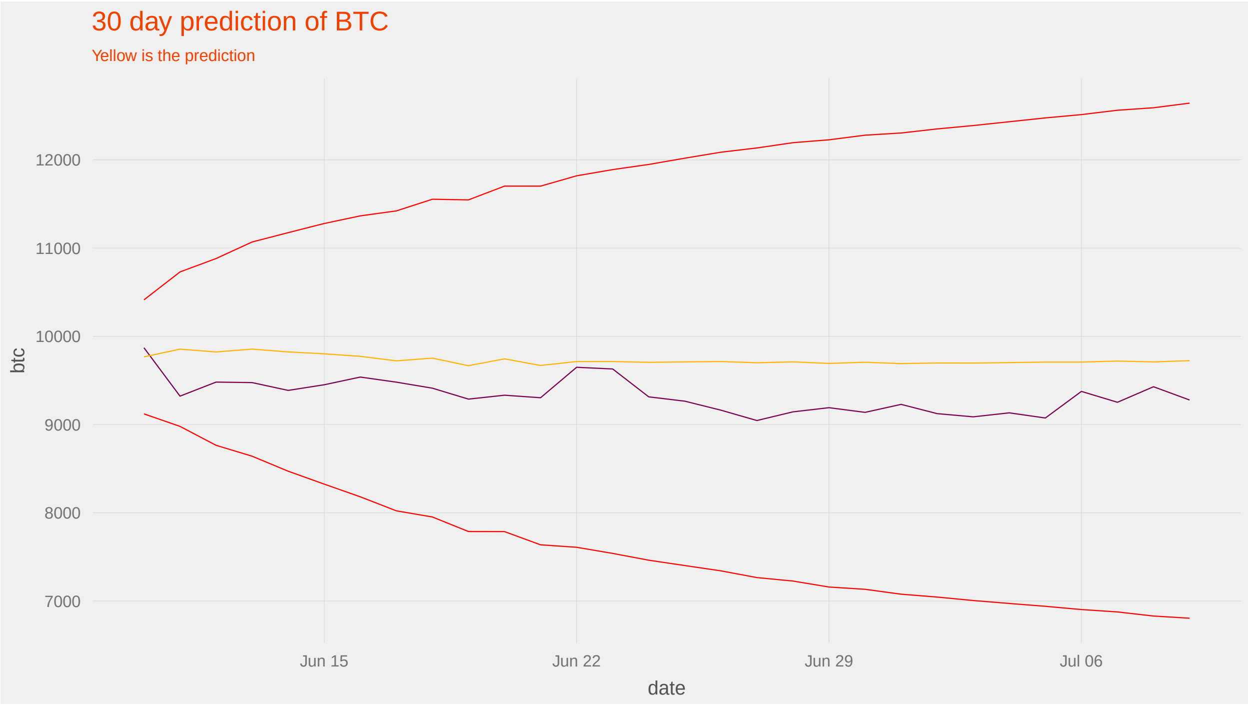


Box-Ljung test confirms that the residuals are not in fact autocorrelated.

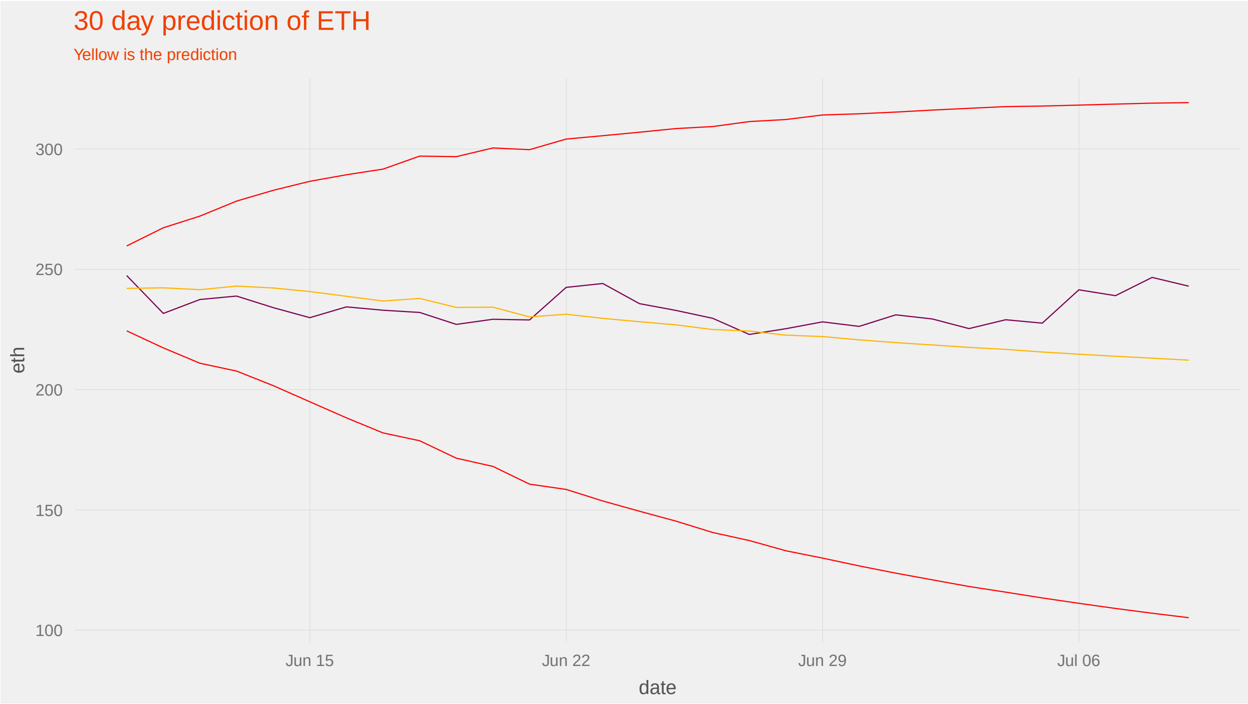
```
	Box-Ljung test

data:  resid(ETH.arima5120.small)
X-squared = 13.822, df = 10, p-value = 0.1812
```

ARIMA forecasting

30 day ARIMA forecast for the Bitcoin is close to the real values, although it seems to overestimate the price and ignore the downward trend present in the data. Because the price is so stable, the predicted values looks kind of like a naive prediction.



Similar results for Etherium - although here it seems that the model was able to predict the general downward trend. It can be said that it predicted it too much.



Cointegration testing

Since Bitcoin and Etherium are both integrated of order I(1), they could be cointegrated with each other.

To test this cointegration, a simple linear model can be estimated. The coefficient for eth is very significant, which means that they are in fact cointegrated.

It seems that a increase in price of ETH by 1 USD means an increase of BTC price by around 37 USD.

```
Call:
lm(formula = btc ~ eth, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2793.8 -1629.8   224.4  1170.0  3253.0 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  913.089    267.719   3.411 0.000698 ***
eth           36.789      1.405  26.180  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1551 on 523 degrees of freedom
Multiple R-squared:  0.5672, Adjusted R-squared:  0.5664 
F-statistic: 685.4 on 1 and 523 DF,  p-value: < 2.2e-16
```

But, are the residuals autocorrelated?

Yes, they are autocorrelated of order one. This means that lagged residuals need to be included in the model.

```
  augmentations       adf     p_adf     bgodfrey       p_bg
1             0 -1.936571 0.3367960 5.375543e+00 0.02042098
2             1 -1.994426 0.3151682 5.177530e-03 0.94263767
3             2 -2.049765 0.2944805 6.741192e-04 0.97928621
4             3 -2.048806 0.2948390 6.209291e-06 0.99801180
```

The inclusion of lagged residuals seem to help. The coefficient is significant, which hints that there is a long term relationship between this variables in addition to the short term one (indicated by eth).

```
Call:
lm(formula = btc ~ eth + lresid, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-964.56  -77.85   -3.08   81.18  728.22 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 926.11670   32.77928   28.25   <2e-16 ***
eth          36.74511    0.17200  213.63   <2e-16 ***
lresid        0.98965    0.00535  184.97   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 189.8 on 521 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9935, Adjusted R-squared:  0.9935
F-statistic: 3.99e+04 on 2 and 521 DF,  p-value: < 2.2e-16
```

Granger test confirms that, for 5 lags, there exist a Granger Causality eth -> btc.

```
Granger causality test

Model 1: btc ~ Lags(btc, 1:5) + Lags(eth, 1:5)
Model 2: btc ~ Lags(btc, 1:5)
  Res.Df Df      F   Pr(>F)
1    509
2    514 -5 3.7728 0.002304 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The same results for the other direction. Significant Granger Causality for btc -> eth.

```
Granger causality test

Model 1: eth ~ Lags(eth, 1:5) + Lags(btc, 1:5)
Model 2: eth ~ Lags(eth, 1:5)
  Res.Df Df      F  Pr(>F)
1    509
2    514 -5 2.6875 0.02071 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

VAR model

VARselect function is very useful when searching for the right number of lags in the VAR model. Here, the optimal number seem to be 7 lags, based on AIC and FPE. Could also be 1, but that is not very interesting.

```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     7      1      1      7

$criteria
                  1            2            3            4            5
AIC(n) 1.485490e+01 1.484990e+01 1.486224e+01 1.486040e+01 1.484772e+01
HQ(n)  1.487427e+01 1.488220e+01 1.490745e+01 1.491854e+01 1.491878e+01
SC(n)  1.490434e+01 1.493231e+01 1.497761e+01 1.500874e+01 1.502903e+01
FPE(n) 2.827481e+06 2.813399e+06 2.848316e+06 2.843113e+06 2.807309e+06
                  6            7            8            9           10
AIC(n) 1.484984e+01 1.484766e+01 1.485891e+01 1.486343e+01 1.486663e+01
HQ(n)  1.493381e+01 1.494455e+01 1.496872e+01 1.498616e+01 1.500228e+01
SC(n)  1.506411e+01 1.509490e+01 1.513911e+01 1.517659e+01 1.521276e+01
FPE(n) 2.813279e+06 2.807197e+06 2.838997e+06 2.851910e+06 2.861108e+06
```

A large number of variables are significant for $p < 0.05$. Especially lags 3 and 4 seem to be important for this model.

```
VAR Estimation Results:
=========================
Endogenous variables: btc, eth
Deterministic variables: const
Sample size: 518
Log Likelihood: -5287.369
Roots of the characteristic polynomial:
0.9879 0.9746 0.6873 0.6873 0.6822 0.6822 0.6664 0.6664 0.6414 0.6414 0.6372 0.6372 0.6225 0.043
Call:
VAR(y = data[, c("btc", "eth")], p = 7)


Estimation results for equation btc:
====================================
btc = btc.l1 + eth.l1 + btc.l2 + eth.l2 + btc.l3 + eth.l3 + btc.l4 + eth.l4 + btc.l5 + eth.l5 + b

        Estimate Std. Error t value Pr(>|t|)
btc.l1   0.97243    0.08116  11.982  < 2e-16 ***
eth.l1  -2.03335    2.94166  -0.691  0.48974
btc.l2  -0.06257    0.11916  -0.525  0.59974
eth.l2   6.63201    4.28756   1.547  0.12254
btc.l3   0.25980    0.11927   2.178  0.02985 *
eth.l3 -12.08314    4.30360  -2.808  0.00518 **
btc.l4  -0.35991    0.11969  -3.007  0.00277 **
eth.l4  17.08747    4.33169   3.945 9.12e-05 ***
btc.l5   0.13475    0.12064   1.117  0.26454
eth.l5  -6.83456    4.39307  -1.556  0.12039
btc.l6   0.20331    0.12026   1.691  0.09153 .
```

```
eth.l6 -10.32454    4.38515  -2.354  0.01893 *
btc.l7  -0.15958    0.08127  -1.963  0.05014 .
eth.l7   7.57216    2.95234   2.565  0.01061 *
const   98.96457   59.20818   1.671  0.09525 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 328.7 on 503 degrees of freedom
Multiple R-Squared: 0.9806,  Adjusted R-squared: 0.9801
F-statistic:  1820 on 14 and 503 DF,  p-value: < 2.2e-16


Estimation results for equation eth:
====================================
eth = btc.l1 + eth.l1 + btc.l2 + eth.l2 + btc.l3 + eth.l3 + btc.l4 + eth.l4 + btc.l5 + eth.l5 + 

        Estimate Std. Error t value Pr(>|t|)
btc.l1 -0.003230   0.002250  -1.436 0.151709
eth.l1  1.013036   0.081544  12.423  < 2e-16 ***
btc.l2  0.001461   0.003303   0.442 0.658407
eth.l2  0.070622   0.118852   0.594 0.552645
btc.l3  0.007635   0.003306   2.309 0.021327 *
eth.l3 -0.293363   0.119297  -2.459 0.014264 *
btc.l4 -0.009629   0.003318  -2.902 0.003866 **
eth.l4  0.441083   0.120076   3.673 0.000265 ***
btc.l5  0.002793   0.003344   0.835 0.404073
eth.l5 -0.130994   0.121777  -1.076 0.282584
btc.l6  0.003385   0.003333   1.015 0.310363
eth.l6 -0.231754   0.121557  -1.907 0.057150 .
btc.l7 -0.002410   0.002253  -1.070 0.285326
eth.l7  0.111655   0.081840   1.364 0.173078
const   3.756791   1.641266   2.289 0.022495 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 9.111 on 503 degrees of freedom
Multiple R-Squared: 0.9655,  Adjusted R-squared: 0.9646
F-statistic:  1006 on 14 and 503 DF,  p-value: < 2.2e-16



Covariance matrix of residuals:
       btc     eth
btc 108025 2509.38
eth   2509   83.01

Correlation matrix of residuals:
      btc   eth
btc 1.000 0.838
eth 0.838 1.000
```

Portmanteau test is on the verge of significance, so another test is needed.

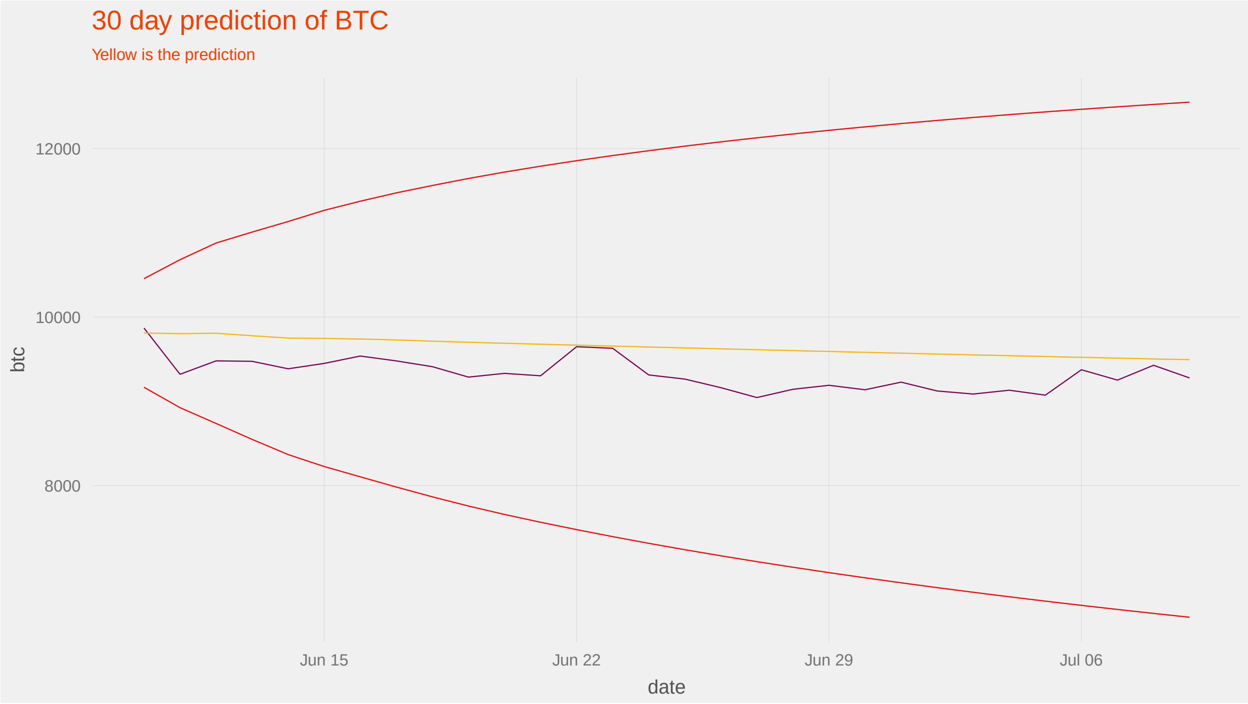
```
 Portmanteau Test (asymptotic)

data:  Residuals of VAR object var.model
Chi-squared = 52.111, df = 36, p-value = 0.04021
```

Breusch-Godfrey test finds no reason to reject the null hypothesis that the residuals are not autocorrelated.

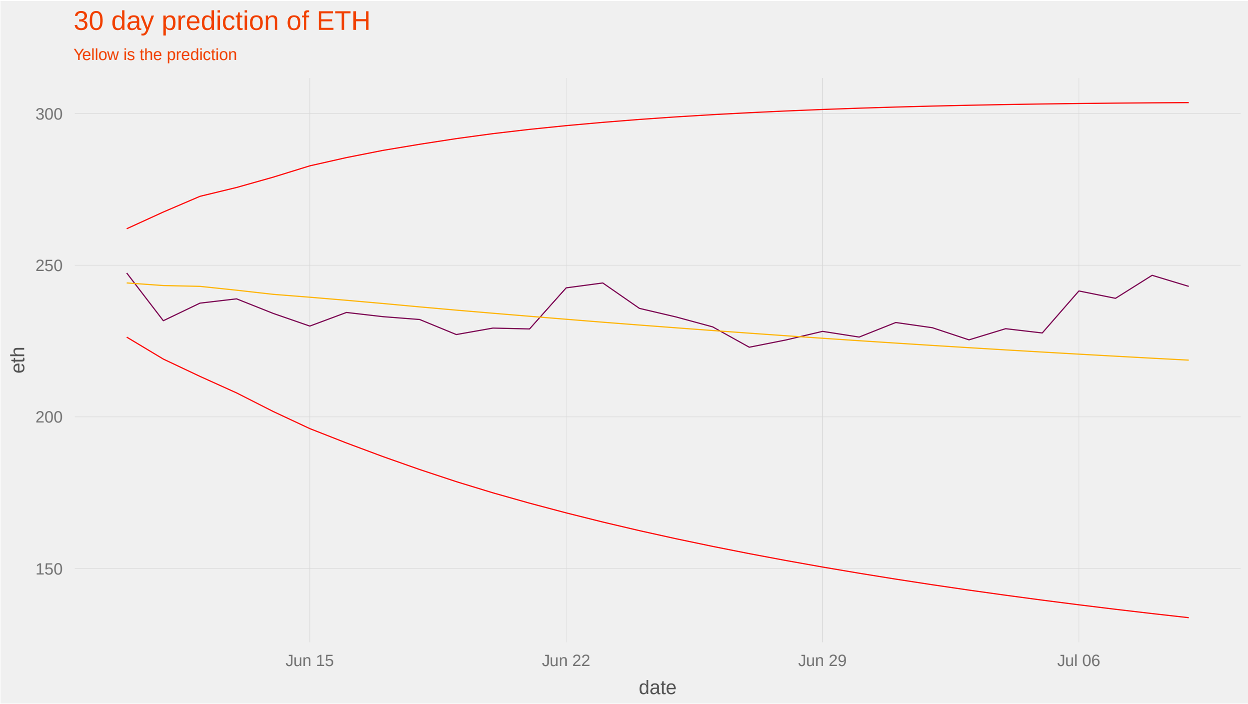
```
 Breusch-Godfrey LM test

data:  Residuals of VAR object var.model
Chi-squared = 22.463, df = 20, p-value = 0.3159
```

Forecasting VAR

The 30 day forecast for Bitcoin using the VAR model is very close to the real values. Because the currency did not move significantly the predictions are also very stable. In some sense they even look like a simple linear regression, but this is only due to the previous prices.



Same story with Etherium. The predictions are very stable, but they do follow the general trend in the data. They also look kind of like a linear regression.



Forecast comparison

Metric used here is sMAPE, Symmetric Mean Absolute Percentage Error. It was chosen because it is unit indifferent and it calculates negative errors with the same weight as the positive ones.

The results are good, sMAPE of *4.3%* for ARIMA and *3.4%* for VAR are very solid. It can be seen that for this data, the VAR model performed significantly better than ARIMA.

The differences between BTC and ETH within both models were insignificant, which hints that these type of models could be used with other cryptocurrencies.

```
   ARIMA_BTC  ARIMA_ETH    VAR_BTC    VAR_ETH
1   0.043256 0.04368248 0.03397326 0.03325533
```

Conclusion

Based on the observed period, it seems that both ARIMA and VAR models are good choices for predicting the price of Bitcoin and Etherium. Final models for ARIMA were:

- ARIMA(20, 1, 0) - Bitcoin
- ARIMA(5, 1, 20) - Etherium

For VAR, the best number of lags was 7. The VAR model achieved better results, thanks to very strong cointegration of the two prices.

Next steps would be to test more currency pairs (ideally using some kind of automation automation). If good results were found for some smaller currencies, they could be used for trading. When testing multiple times, one needs to be wary of the type one error. Assuming a lower p value of e.g. 0.01 would be advisable.

Other possibilities could be to use cross validation to test the performance of the models on different parts of data (the one chosen was quite calm).