TIME SERIES FORECASTING

MODELING BITCOIN PRICE

George Washington University

Junzhe Yin

Junyi Qian

Meizi Yu

Yichen Li

Shengqi Zhou

Contents

Abstract

Since Bitcoin launched in 2009, it has become widely popular because its trading system doesn't need a third party and also due to high volatility of the price. In this project, 30 hold-out samples were selected for building a cyclical model of variable "High" with *AR(2)* serves as a fit error model and *ARIMA* model. Also, in this project, multivariate models were built to explore the correlations between *High*, *Volumn*, and *Close*. Model comparison was provided at the end of this paper.

1. Introduction and Overview

As the interest in cryptocurrencies grows, various individual currencies appear on the global market. From the dataset of historical cryptocurrency financial information, Bitcoin serves as the major research object out of the top 10 cryptocurrencies by market capability. This report mainly focuses on what drives the fluctuations of the bitcoin exchange price and to what extent they are predictable. The goal of this project is to analyze and predict how highest prices behave through a detailed time series analysis using the Bitcoin Historical data. For investors it can be an added advantage that they model the series for any future investment and make larger margin profits. Short term forecasting is good for trading.

There are a total of 1000 observations in the data and 150 hold out samples are selected out for further analysis. The dataset contains the opening and closing prices of bitcoins from Mar 10, 2017 to Dec 04, 2019 and other variables including High, Close,and Volume. Because of its better performance, the highest trading price of the day will be the target research object and volume will be used for further multivariate models.

- Date : from Mar 10, 2017 to Dec 04, 2019
- High : highest recorded trading price of the day
- Close: the price at the end of the day
- Volume : the monetary value of the currency traded in a 24 hour period, denoted in USD

For High:

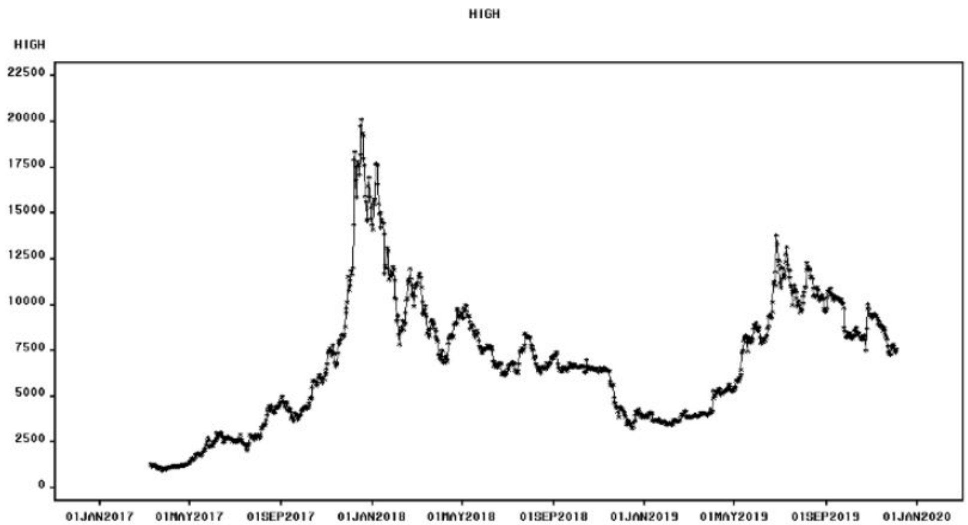


Figure1.1 The original time series graph of column high

Observed from *Figure 1.1*, there is a slightly upward trend among this 2 year and a half period for Bitcoin's highest price of the day. It’s hard to see a clear seasonality since the plot has alternate highs and lows. Obviously, there is an explosive price move in 2017 that was an outlier. The highest price Bitcoin ever reached until today was around $20,000 on Dec 18th,2017. Commentators and critics called this a price bubble. Indeed, just a few weeks later, the price bitcoin fell rapidly, crashing all the way down below $7,000 by April 2018.

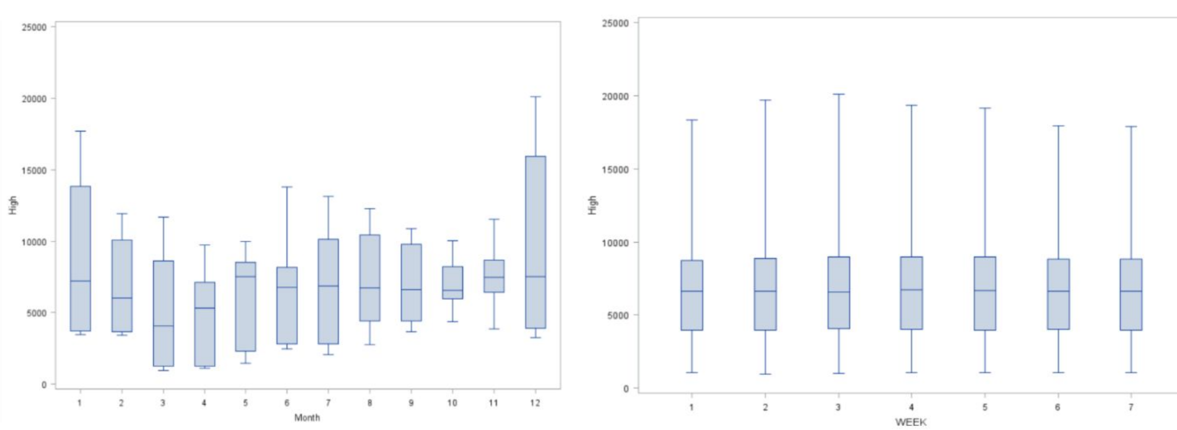


Figure 1.2 boxplot

Even though Bitcoin has a quite strong price volatility, there is no much high price difference between months or weeks. March is typically a weak month for Bitcoin. May and June are two of the strongest months of the year but there is no much difference in median from June to the end of the year. November and December also perform relatively well. From the boxplot of the week in *Figure 1.2*, the median of the seven day of the week are very similar.

2. Univariate Time-series models

2.1 Cyclical Trend and Error model.

The cyclical component of a time series refers to (regular or periodic) fluctuations around the trend, excluding the irregular component, revealing a succession of phases of expansion and contraction. From *Figure1.1*, the plot of HIGH shows a general cyclical trend which is suitable to build a cyclical model. Thus, the SAS is needed for the HIGH series periodogram.

The SAS System

Obs	FREQ	PERIOD	P_01
1	0.00000	.	0.00
2	0.00628	1000.00	2349.67
3	0.01257	500.00	4641.64
4	0.01885	333.33	1385.61
5	0.02513	250.00	608.94
6	0.03142	200.00	315.21
7	0.03770	166.67	229.10
8	0.04398	142.86	330.45
9	0.05027	125.00	61.56
10	0.05655	111.11	197.91
11	0.06283	100.00	112.34
12	0.06912	90.91	13.14
13	0.07540	83.33	23.87
14	0.08168	76.92	119.66
15	0.08796	71.43	56.57
16	0.09425	66.67	63.85
17	0.10053	62.50	124.54

Figure 2.1.1 SAS System of High

Based on *Figure 2.1.1*, the top 11 p-values are from observation 2 to 8, 10, 11 14 ,17, and all over the 100. In other words, lag 1 to 7, 9 10, 13, 16 might have hidden terms to explain the relatively large p-value, so these lags are appropriate to create sine and cosine terms as regressor, and fit cyclical models. The periodogram is also provided in *Figure 2.1.2*, which clearly records how the period fluctuates during times.

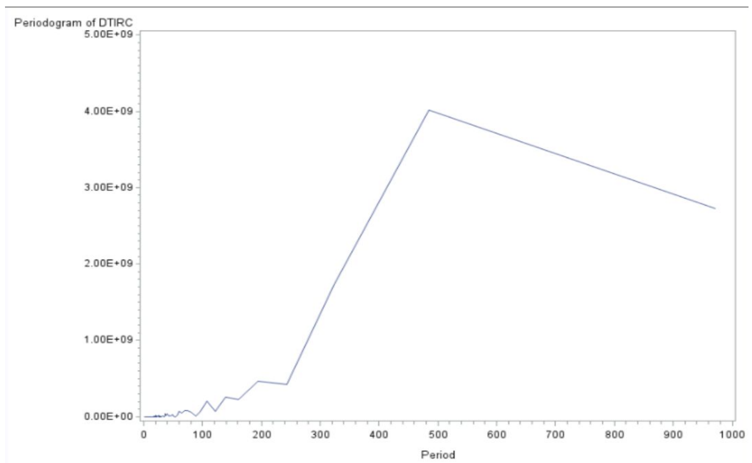
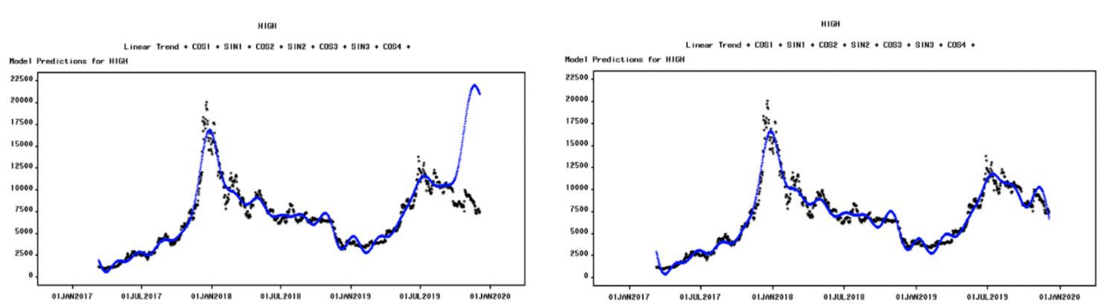


Figure 2.1.2 Periodogram of HIGH

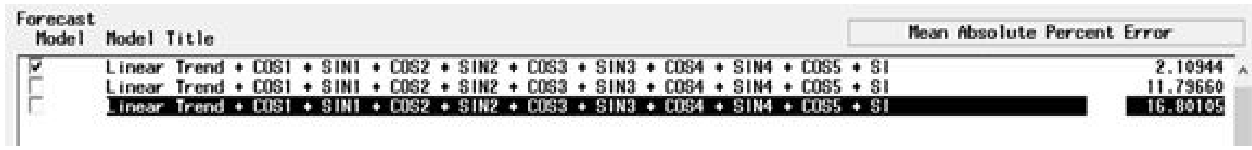
To build a cyclical model perfectly, the first step is choosing a hold-out sample. From *Figure2.1.1*, the original series trend is continuously increasing from January to July in 2019, and has turning points after July which reflects the actual trend of HIGH price is decreasing after July 2019. Therefore, given that the dataset has 1000 daily observations, to predict as fact, the right size of hold-out sample should be relatively small, 30-40(3%-4% instead of required 15%), to let the training set include data after turning points and fit a decreasing trend.



85 hold-out sample 30 hold-out sample

Figure 2.1.3 prediction of 85 and 30 hold-out sample

From *Figure2.1.3*, 30 hold-out samples clearly fit better than 85 hold-out samples. Then, the linear and cyclical trend model is built with the regressors, and sample size is 30 to 40 .



Forecast Model	Model Title	Mean Absolute Percent Error
✓	Linear Trend + COS1 + SIN1 + COS2 + SIN2 + COS3 + SIN3 + COS4 + SIN4 + COS5 + SI	2.10944
	Linear Trend + COS1 + SIN1 + COS2 + SIN2 + COS3 + SIN3 + COS4 + SIN4 + COS5 + SI	11.79660
	Linear Trend + COS1 + SIN1 + COS2 + SIN2 + COS3 + SIN3 + COS4 + SIN4 + COS5 + SI	16.80105

Figure 2.1.4 MAPE of 30+AR(2), 30 ,40 hold-out samples

From *Figure 2.1.4*, The mean absolute percent error(*MAPE*) of 30 hold-out samples is 11.79660, which is lower than 16.80105, *MAPE* of 40 hold-out samples. This means 30 fits the series better (the lower the *MAPE*, the better the plot of fit), so 30 hold-out sample could be better sample side in this project to build models, and the corresponding the autocorrelations function(*ACF*) and partial autocorrelation function(*PACF*) is shown as below.

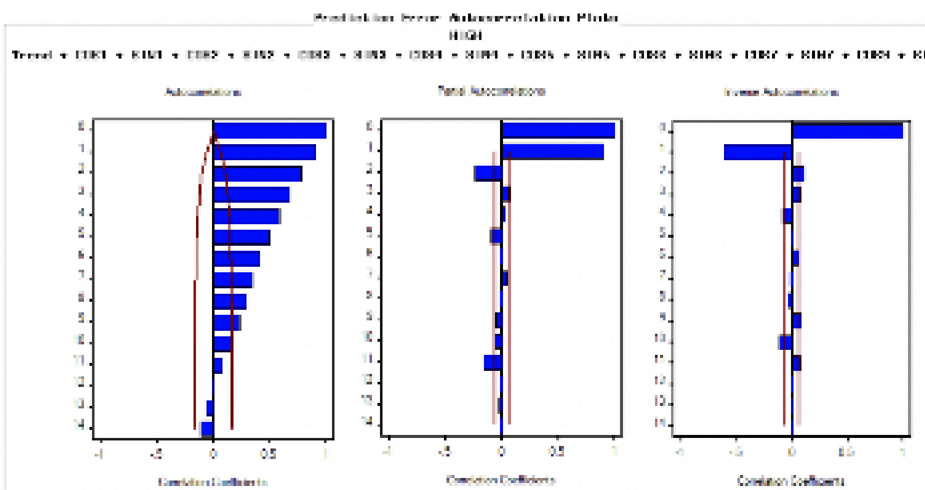


Figure 2.1.5 ACF, PACF and IACF of 30 hold-out sample

In *Figure 2.1.5*, the *ACF* decaying slowly which means it is non-stationary, so the series need error model to refit a better outcome, and when we see the *PACF* plot, it chopped off at lag 2, which brings us that *AR(2)* model will be a good error model for the series. Also, from Figure *2.1.4*, the *MAPE* of *AR(2)* model is 2.10994, which is significantly lower than the 30-hold-out sample model's *MAPE*, so *AR(2)* model needs to be built in the model.

From *Figure 2.1.6*, the prediction of the *AR(2)* series fit better than 30 in *Figure 2.1.3*. What's more, in *Figure2.1.7*, *ACF* decayed quickly, so the new model is stationary. Therefore, the *AR(2)* model of the cyclical model is relatively the best for HIGH variables.In addition, the parameter estimated table of the new *AR(2)* model is shown below(*Figure2.1.8*).

Figure 2.1.6 prediction of AR(2) model based

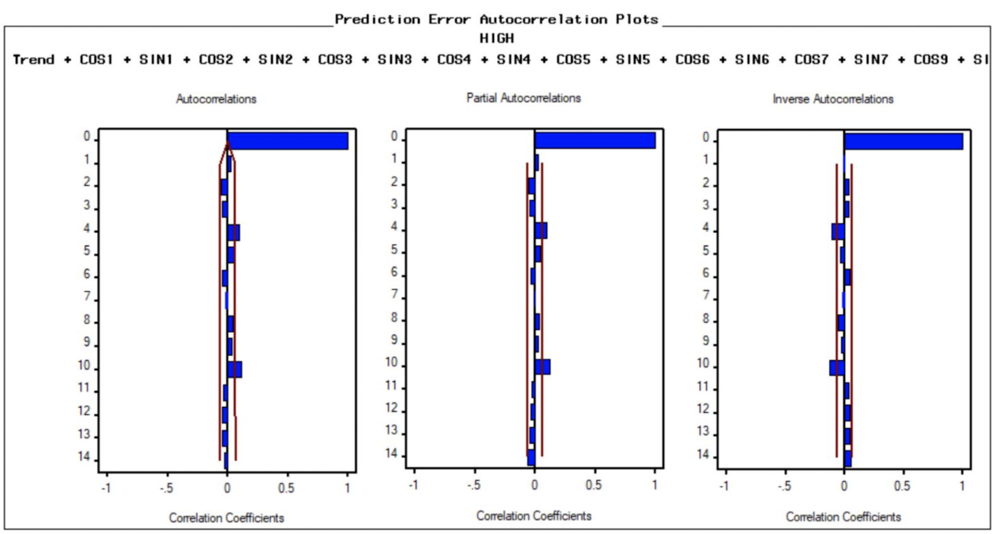


Figure 2.1.7 ACF, PACF and IACF of AR(2) model

According to *Figure 2.1.8*, the forecasting function could be written based on these coefficients. Furthermore, since the p-value of COS3, COS4, SIN5, COS6, SIN7, COS13 is too large(beyond 0.05), it is proper to drop off their term and consider them as unnecessary when it comes to the function part.

Parameter Estimates

HIGH

2 + SIN2 + COS3 + SIN3 + COS4 + SIN4 + COS5 + SIN5 + COS6 + SIN6 + COS7 + SIN7 + COS9 + SIN9 + COS10 + SIN10 + COS13

Model Parameter	Estimate	Std. Error	T	Prob>|T|
Autoregressive, Lag 1	1.13169	0.0315	35.9298	<.0001
Autoregressive, Lag 2	-0.24940	0.0315	-7.9174	0.0014
Linear Trend	4.47961	1.9600	2.2809	0.0847
COS1	-900.74950	140.1097	-6.4023	0.0029
SIN1	2061	652.4732	3.1631	0.0341
COS2	-636.23956	139.6652	-4.5555	0.0104
SIN2	-2955	328.8228	-8.9855	0.0008
COS3	109.17799	138.8051	0.7866	0.4755
SIN3	-1659	232.6539	-7.1298	0.0020
COS4	-271.18061	137.6297	-1.9704	0.1201
SIN4	1057	187.8604	5.6298	0.0049
COS5	-742.58060	136.1063	-5.4535	0.0055
SIN5	179.43327	163.1929	1.0995	0.3333
COS6	19.10612	134.4485	0.1421	0.8939
SIN6	-693.00440	148.2505	-4.6745	0.0095
COS7	835.01469	132.5145	6.3013	0.0032
SIN7	-43.51862	138.6015	-0.3140	0.7692
COS9	-422.55960	128.1653	-3.2970	0.0300
SIN9	-479.29221	127.4029	-3.7620	0.0197
COS10	328.87962	125.8363	2.6135	0.0592
SIN10	-581.55353	123.9437	-3.6794	0.0378
COS13	-68.06717	118.7087	-0.5729	0.5974
SIN13	-518.05135	117.0737	-4.4277	0.0114
COS16	-287.57873	111.9815	-2.5681	0.0621
SIN16	-449.16938	111.7401	-4.0198	0.0159
Model Variance (sigma squared)	108365	.	.	.

Fit Range: 10MAR2017 to 04NOV2019

Figure 2.1.8 parameter estimated table

2.2 ARIMA models

ARIMA Model can be applied when the time series is non-stationary and the differenced series is stationary. For stationary time series, the autocorrelations function(*ACF*) decays quickly as the lag increases. Nonstationary time series have that *ACF* decays very slowly. Typically, some parameters, *ARIMA(p,d,q)*, needs to be defined in SAS for building *ARIMA* models : *d* represents that after d^{th} differencing time series become stationary; *p* represents that partial autocorrelation function(*PACF*) chopped off after Lag P; and *q* represents that *ACF* chopped off after Lag *q*. The *ACF*, *Figure 2.2.1*, decays slowly which means the original time series is not stationary. After first differencing, the *ACF* decays quickly as shown in *Figure 2.2.2*. Thus *ARIMA* model could be applied to this time series data and *d* is equal to 1. Moreover, if the *ACF* after first differencing is interpreted as decaying quickly and *PACF* chopped off after Lag2 or

Lag 1, *d =1* and *p =2 or 1*, *ARIMA(2,1,0)* or *ARIMA(1,1,0)* can be applied. If the ACF after first differencing is interpreted as chopped off after Lag 1 and the *PACF* decaying quickly, *d = 1*and *q = 1*, *ARIMA(0,1,1)* can be applied. The hold out sample used is 30 for *ARIMA* models in this project.

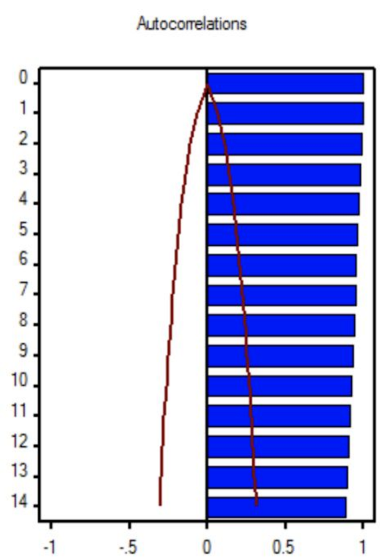


Figure 2.2.1 ACF of High

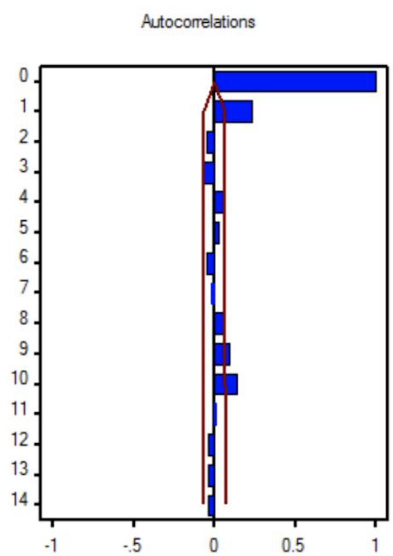


Figure 2.2.2 ACF after first differencing

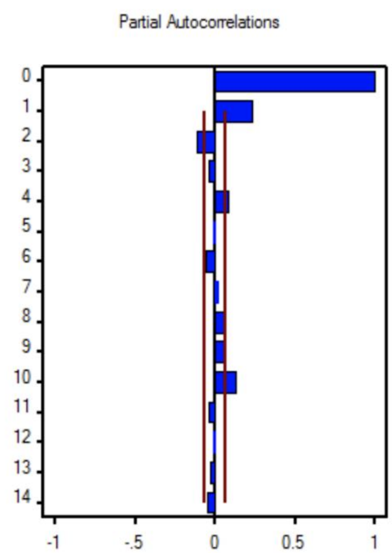


Figure 2.2.3 PACF after first differencing

There are three *ARIMA* models: *ARIMA(2,1,0)* , *ARIMA(1,1,0)* and *ARIMA(0,1,1)*, which could be applied to the time series data. Best fitted model can be chosen based on several factors: significance of estimated parameters, estimated model variance, and mean absolute percent error. When building the model, since *p* could be 2 or 1, *ARIMA(2,1,0)* was built first. However, the *p-value* of lag 2 is not significant shown as *Figure 2.2.4*. Thus *ARIMA(1,1,0)* would be a better model than *ARIMA(2,1,0)*.

Model Parameter	Estimate	Std. Error	T	Prob>|T|
Autoregressive, Lag 1	0.25333	0.0315	8.0470	<.0001
Autoregressive, Lag 2	-0.10946	0.0315	-3.4771	0.0005
Model Variance (sigma squared)	113498	.	.	.

Figure2.2.4 Parameter estimates table

After building *ARIMA(1,1,0)*, SAS returned a model prediction graph(*Figure2.2.5*) which shows the model fitted the time series data very well. Furthermore, the p-value of the estimated coefficient, 0.228, for Lag 1 is small enough to be significant and the square root of model variance estimate is around 342.78 shown as *Figure 2.2.6*.

Figure2.2.5 Model prediction for High of ARIMA(1,1,0)

Model Parameter	Estimate	Std. Error	T	Prob>¦T¦
Autoregressive, Lag 1	0.22783	0.0313	7.2801	<.0001
Model Variance (sigma squared)	117499	.	.	.

Figure2.2.6 Parameter estimates table of ARIMA(1,1,0)

Errors of prediction models are also important factors to compare the model performance. From *Figure2.2.7*, The mean square error(*MSE*) of *ARIMA(1,1,0)* is 26530.9. The mean absolute percent error(*MAPE*) is around 1.4968, which means there was about 1.5% data that could not fit in the model. Since the *MAPE* of this model is very small, it can be concluded that the model performance is good enough. Based on the parameter estimates table(*Figure2.2.6*), the final equation for predicting "High" price of Bitcoin from ARIMA(1,1,0) would be:

$$\widehat{P}_t = P_{t-1} + 0.228(P_{t-1} - P_{t-2}).$$

Statistic of Fit	Value
Mean Square Error	26530.9
Root Mean Square Error	162.88315
Mean Absolute Percent Error	1.49682
Mean Absolute Error	123.32323

Figure2.2.7 Statistic of Fit table of ARIAM(1,1,0)

Also, *ARIMA(0,1,1)* can be applied to the times series data in this project. After building the *ARIMA(0,1,1)* model, SAS also returned a model prediction graph(*Figure2.2.8*) which shows the model fitted performance. Based on the graph, the *ARIMA(0,1,1)* model predicts the time series data properly. Furthermore, the p-value of the estimated coefficient, -0.252, for Lag 1 is small enough to be significant and the square root of model variance estimate is around 341.50 shown as *Figure 2.2.9*.

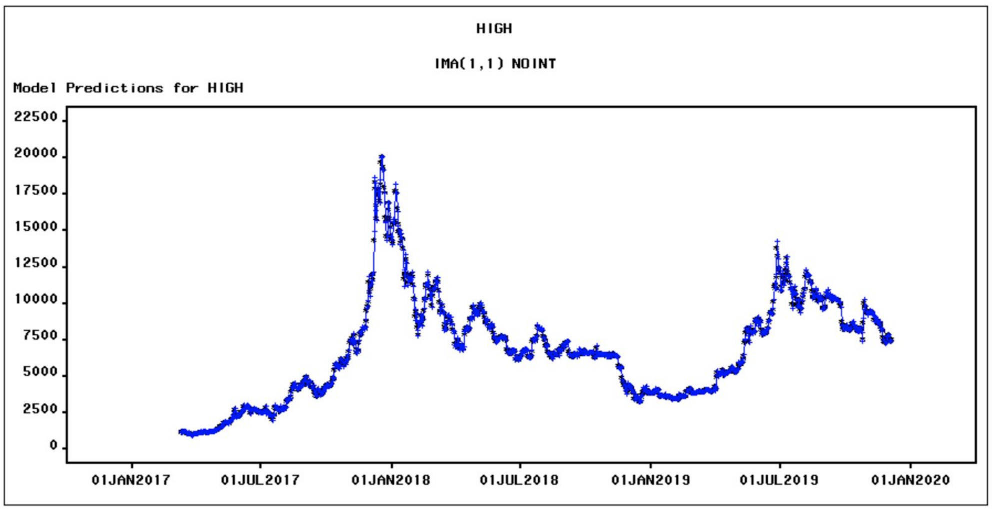


Figure2.2.8 Model predictions for HIGH of ARIAM(0,1,1)

Model Parameter	Estimate	Std. Error	T	Prob>¦T¦
Moving Average, Lag 1	-0.25218	0.0311	-8.1068	<.0001
Model Variance (sigma squared)	116619	.	.	.

Figure2.2.9 Parameter estimates table of ARIMA(0,1,1)

Also, it needs to check the model errors.From *Figure2.2.10*, The *MSE* is 26475.4. The *MAPE* is around 1.51, which means there was about 1.51% data that could not fit in the model. Since the *MAPE* is very small, it can be concluded that the model predictions perform properly. Based on the Figure2.2.9, the final equation for predicting “High” price of Bitcoin from *ARIMA(0,1,1)* would be:

$$\widehat{P}_t = P_{t-1} + 0.252\varepsilon_{t-1},\ where\ \varepsilon_{t-1} = P_{t-1} - \widehat{P}_{t-2}.$$

Statistic of Fit	Value
Mean Square Error	26475.4
Root Mean Square Error	162.71273
Mean Absolute Percent Error	1.50846
Mean Absolute Error	124.35233

Figure2.2.10 Statistic of Fit table of ARIAM(1,1,0)

2.3 Comparison of models

Based on what we have in *Section2.1* and *Section2.2*, *AR(2) cyclical trend model, ARIMA(0,1,1)* and *ARIMA(1,1,0)* all show good fit compared with reality and predict a decreasing trend. However, to become more specific about the advantages and disadvantages of different models, statistical indicators are considered to judge these models.

Hold Out	Models	MAPE	MAE	RMSE	Square root of model variance
30	Cyclical+AR(2)	2.10944	174.94778	201.75939	329.188
30	IAR(1,1)	1.49682	123.32323	162.88315	342.78
	IMA(1,1)	1.50846	124.35233	162.71273	341.50

Figure 2.3.1Model comparison table

Figure 2.3.1 shows the comparison of different models based on some various statistical indicators. In detail, based on *MAPE*, *MAE* and *RMSE*, *ARIMA* models are better than the *AR(2) cyclical model*, and all indicators in *ARIMA(1,1,0) is slightly* lower than that in *ARIMA(0,1,1),* which means *ARIMA(1,1,0)* is better in this situation. On the other hand, the square root of *AR(2)* cyclical trend model variance is lower than *ARIMA* models, which means *AR(2)* cyclical model is more steady.

To sum up, since variance is too big and the difference of square root of variance is relatively small, *MAPE* is considered as a more important factor so that *ARIMA(1,1,0)* model would be the best model of the three.

3. Multivariate Time Series Models

3.1 Transfer function model of High and Volume

3.1.1 Pre-whitening and Determine model

First, to discover the relationship between *High* and *Volume*, it is important to find if fluctuations of market trading affect the value of the highest price. To determine whether a TF noise model can be applied here, *CCF* between *High* and *Volume* and the plot of residuals needs to be checked. *High* as a dependent variable and *Volume* as input variable are setted.

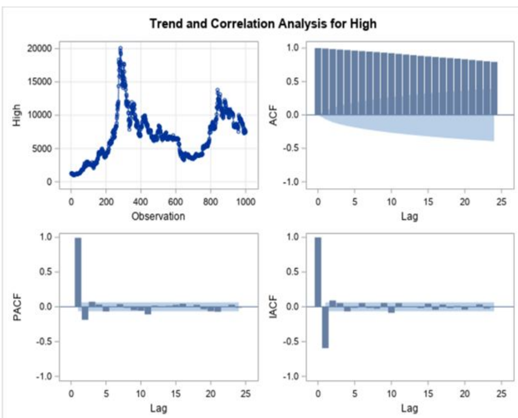


Figure 3.1.1 Original ACF of High

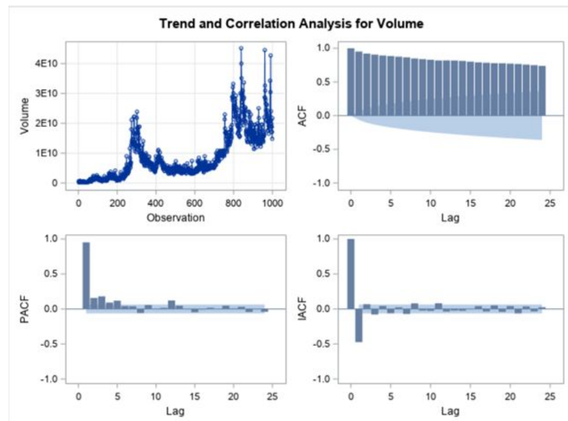


Figure 3.1.2 Original ACF of Volume

From *Figure3.1.1* and *Figure3.1.2*, the original series of both variables, *Volume* and *High*, are apparently non-stationary series due to the slowly decaying of the *ACF*. And after a first difference of both series, the *ACF* of both series are decaying quickly as shown in *Figure3.1.3* and *Figure3.1.4* below:

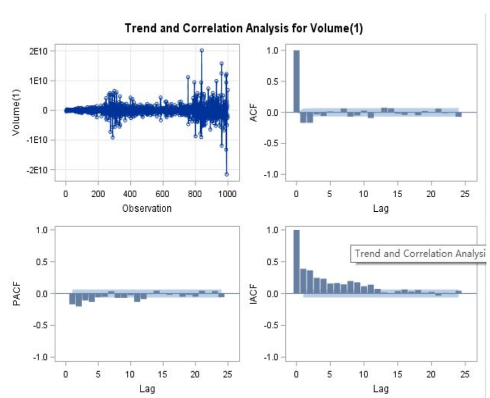


Figure 3.1.3 First difference of Volume

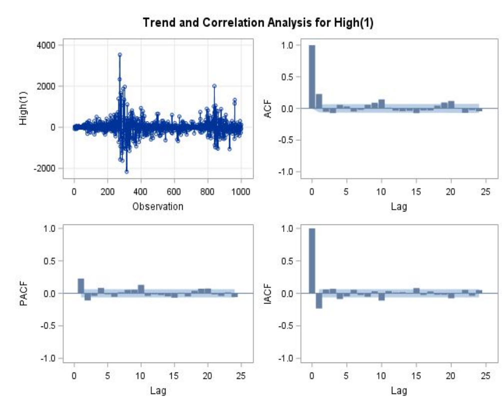
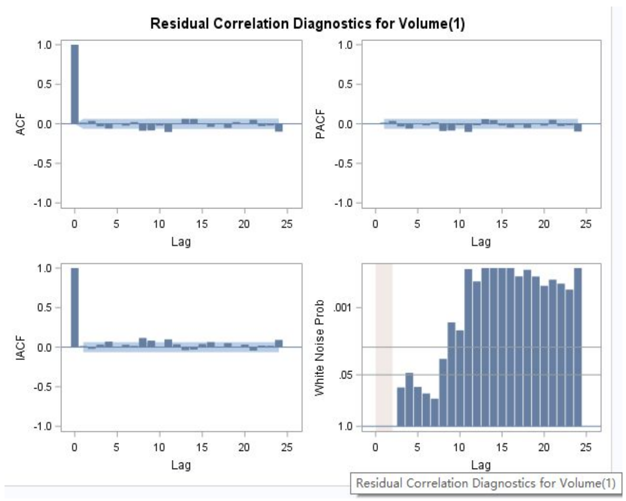


Figure 3.1.4 First difference of High

Both series become stationary, but *Volume* is not white noise. According to the *ACF* and *PACF* in F*igure 3.1.3*, *MA(2)* is suitable for pre-whitening progress to make input become white noise. *Figure 3.1.5* shows the result after pre-whitening, and the residuals of input become white noise.



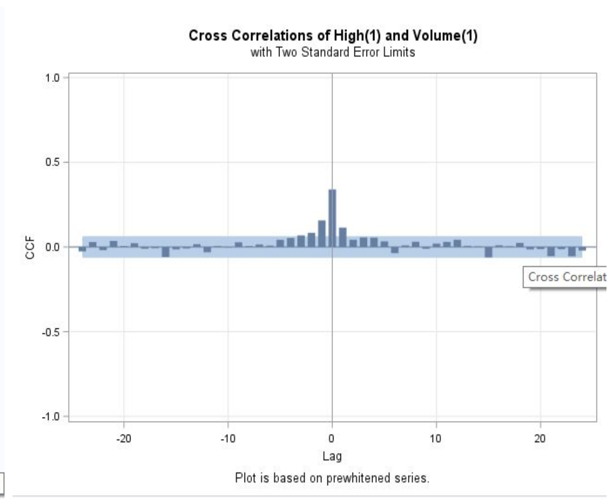


Figure 3.1.5 Residual check after pre-whitening

Figure 3.1.6 CCF plot

After data processing, based on the Cross Correlation plot in *Figure 3.1.6*, the parameters of the TF model can be determined, b=0, r=0 and s=0. The model is built by these parameters and Model adequacy can be checked in *Figure 3.1.7* and *Figure3.1.8*:

Crosscorrelation Check of Residuals with Input Volume									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	9.40	5	0.0941	0.013	0.066	-0.017	0.036	0.057	-0.001
11	23.68	11	0.0141	-0.070	0.005	0.077	-0.002	0.012	0.058
17	34.55	17	0.0071	0.027	-0.045	-0.027	-0.067	0.050	0.018
23	45.41	23	0.0035	0.039	-0.039	-0.020	-0.074	0.021	-0.038
29	54.10	29	0.0032	0.026	0.009	0.002	-0.044	-0.077	0.007
35	56.61	35	0.0118	0.012	-0.006	-0.030	-0.019	0.033	-0.001
41	71.26	41	0.0024	-0.008	0.036	0.056	0.080	-0.016	0.060
47	84.35	47	0.0007	-0.081	0.049	0.037	0.047	-0.012	0.019

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	57.69	6	<.0001	0.217	-0.024	-0.090	0.032	0.023	-0.018
12	98.56	12	<.0001	-0.018	0.042	0.097	0.165	-0.009	-0.039
18	109.21	18	<.0001	-0.053	-0.040	-0.053	-0.022	-0.043	0.031
24	147.89	24	<.0001	0.103	0.134	0.020	-0.074	-0.031	-0.051
30	154.10	30	<.0001	0.002	-0.021	-0.018	-0.012	0.032	0.064
36	162.98	36	<.0001	-0.025	-0.061	-0.016	-0.027	0.020	-0.053
42	172.59	42	<.0001	-0.030	0.030	0.053	0.015	-0.061	-0.026
48	202.08	48	<.0001	0.084	0.118	0.034	-0.018	-0.077	0.002

Figure 3.1.7 Crosscorrelation check table

Figure 3.1.8 Autocorrelation check table

Although the p-value is all large in Crosscorrelation Check of Residuals, the *ACF* check of residuals from *Figure3.1.8* is small which means a noise model should be applied. Based on *Figure3.1.9*, *PACF* decayed quickly and *ACF* is significant at lag1 and lag 10, *MA(1,10)* noise model would be a proper noise model.

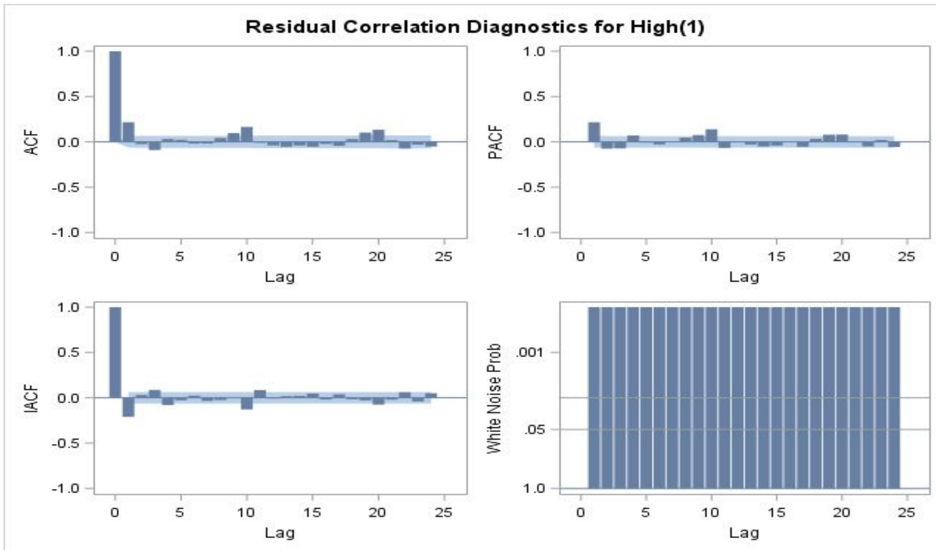


Figure 3.1.9 Residual plot used to determine the noise mode

After adding a noise model, the residuals finally become white noise as shown in *Figure 3.1.10*:

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	13.96	4	0.0074	0.002	-0.008	-0.095	0.060	0.020	-0.029
12	18.01	10	0.0548	-0.002	0.035	0.047	0.012	-0.011	-0.018
18	25.85	16	0.0562	-0.031	-0.026	-0.055	0.005	-0.049	0.025
24	49.59	22	0.0007	0.068	0.109	0.012	-0.067	0.001	-0.045
30	53.58	28	0.0025	0.020	-0.018	0.000	-0.023	0.005	0.051
36	62.39	34	0.0021	-0.020	-0.043	-0.005	-0.044	0.033	-0.056
42	69.49	40	0.0026	-0.013	0.025	0.053	0.014	-0.050	-0.022
48	93.96	46	<.0001	0.064	0.116	0.006	0.002	-0.073	0.025

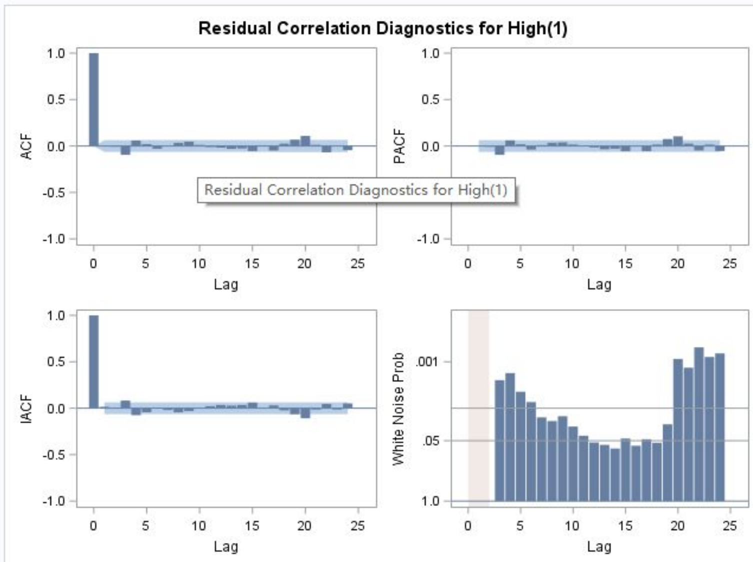


Figure 3.1.10 Residual correlation after added noise model

3.1.2 Final TF model by SAS forecast system

After confirming parameters and noise model, *SAS* forecast system can be used to build the final model. Hold-out samples 30 is used, which is the same quantity as in the univariate model part. By Factored *ARIMA* Model in SAS, the result of b=0, r=0, s=0 and *MA(1,10)* noise model is shown in *Figure3.1.11* and *3.1.12*:

HIGH
VOLUME + IMA d=(1) q=(1, 10) NOINT

Model Parameter	Estimate	Std. Error	T	Prob>|T|
MA factor 1 lag 1	-0.23082	0.0311	-7.4134	<.0001
MA factor 1 lag 10	-0.13987	0.0311	-4.5040	0.0001
VOLUME	5.77879E-8	4.5897E-9	12.5908	<.0001
Model Variance (sigma squared)	98729	.	.	.

Figure 3.1.11

Statistic of Fit	Value
Mean Square Error	172551.7
Root Mean Square Error	415.39348
Mean Absolute Percent Error	3.59741
Mean Absolute Error	280.09933

Figure 3.1.12

From *Figure3.1.11*, all parameters have a small P-value which means significant. Based on *Figure3.1.12*, *MAPE* is 3.597 and *RMSE* is 415.39 which means model fits properly and *Figure3.1.13* is the fitted plot:

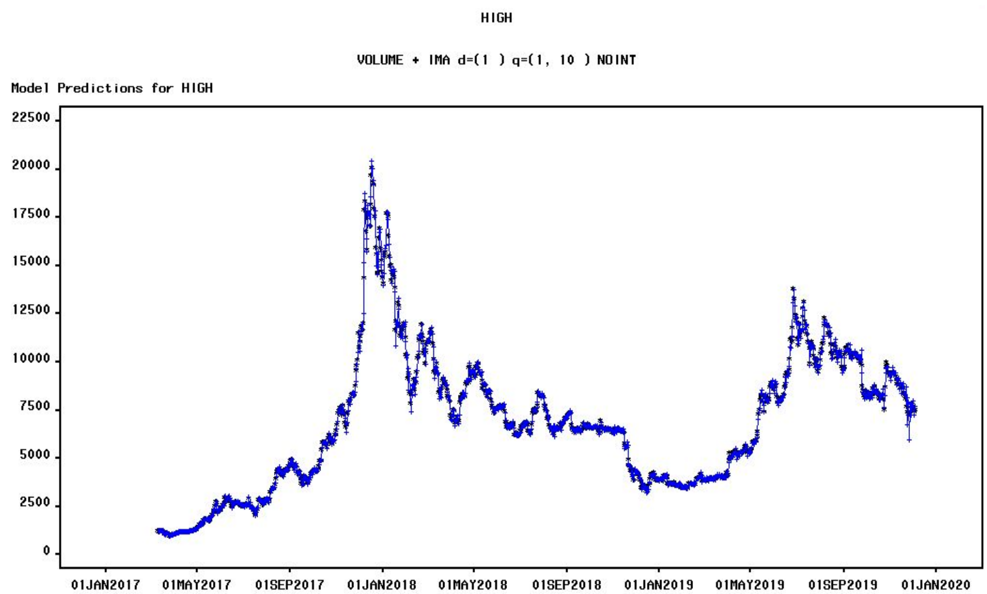


Figure 3.1.13

3.2 Transfer function model of High and Close

3.2.1 Pre-whitening and Determine model

Furthermore, there might be variables other than *Volume* having relation with *High*. To explore the relationship between *Close* and *High*, same procedures can be applied. In this part, *Close* is set to be input variabel and High to be dependent variable. Since both of the origin series are not stationary, first differenced is applied and results is shown as *Figure3.2.1* and *Figure3.2.2*:

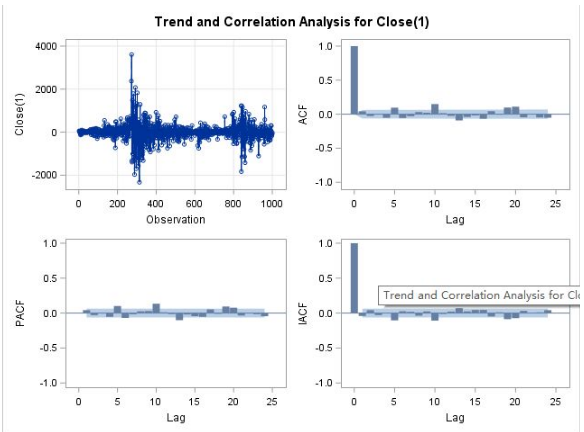


Figure 3.2.1 First difference of Close

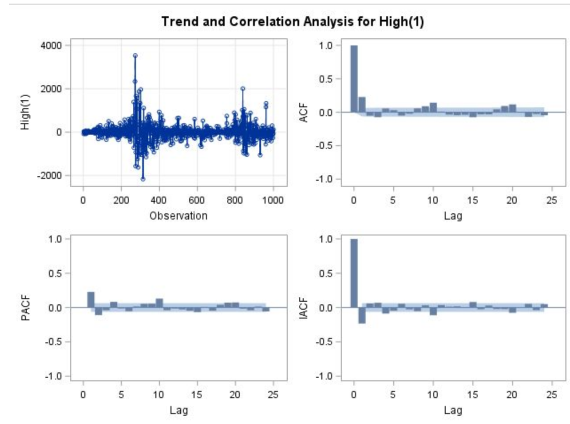


Figure 3.2.2 First difference of High

Based on *Figure3.2.1*, after the first difference, *Close* becomes stationary and white noise, so there is no need to make pre-whitening. According to *Figure3.2.2*, *High* becomes stationary after the first difference which meets the requirement of the dependent variable. Then from *Figure3.2.3*, the parameters of the TF model are b=0, r=0, s=1.

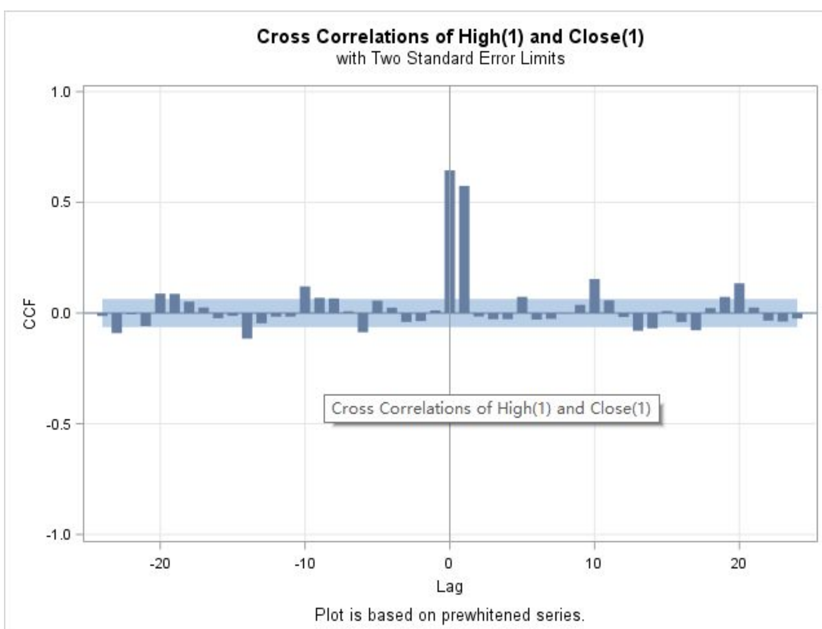


Figure 3.2.3 CCF plot of High and Close

After determining the parameters, *Figure 3.2.4* and *Figure3.2.5* contain the information of model adequacy.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	212.53	6	<.0001	-0.426	-0.051	-0.076	0.132	-0.063	-0.032
12	228.84	12	<.0001	0.031	0.039	-0.072	0.063	-0.066	-0.014
18	266.59	18	<.0001	-0.010	0.114	-0.128	0.072	0.023	-0.045
24	280.38	24	<.0001	-0.019	0.040	0.024	-0.071	0.075	0.015
30	334.31	30	<.0001	-0.062	-0.078	0.168	-0.043	-0.099	0.051
36	344.56	36	<.0001	0.058	-0.032	-0.053	0.048	-0.018	0.007
42	398.01	42	<.0001	0.055	-0.112	-0.011	0.086	0.077	-0.149
48	414.79	48	<.0001	0.027	0.051	-0.025	0.015	-0.058	0.092

Figure 3.2.4 Autocorrelation check of residuals

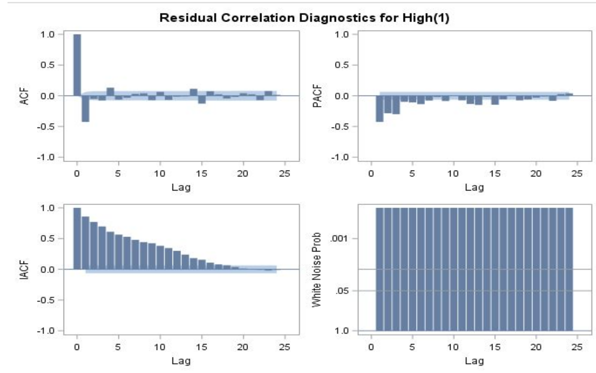


Figure 3.2.5 ACF of residuals

From *Figure3.2.4*, the p-value of residuals are small and significant, so it still needs to add a noise model to make residuals to white noise. Since *PACF* decayed quickly and *ACF* chopped off after lag 1, *MA(1)* is suitable for the noise model based on *Figure3.2.5*. After adding *MA(1)* noise model, the residuals finally become white noise as shown in *Figure3.2.6*. Also the p-value in the Crosscorrelation check of residuals is proper, shown as *Figure3.2.7*. Although p-value in several lags is not large enough, it is still accepted due to the large number of observations.

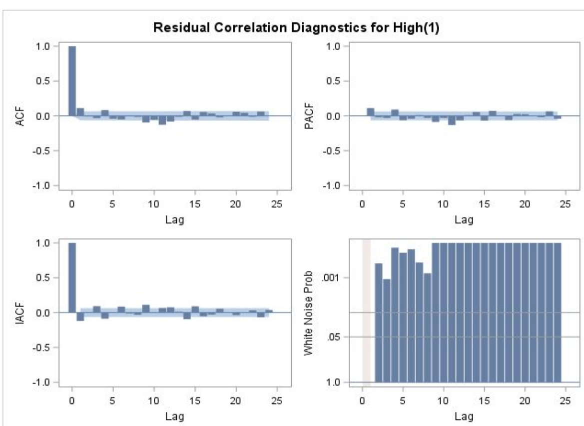


Figure 3.2.6 Residual plot after add noise model

Crosscorrelation Check of Residuals with Input Close									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	10.51	4	0.0327	-0.000	0.028	-0.026	-0.057	-0.042	0.063
11	31.01	10	0.0006	-0.055	0.003	-0.008	0.009	0.126	0.037
17	56.70	16	<.0001	0.006	-0.018	-0.004	0.096	0.108	-0.068
23	62.67	22	<.0001	-0.052	-0.006	0.029	0.012	-0.019	-0.044
29	72.95	28	<.0001	0.040	-0.020	0.058	-0.058	-0.040	-0.001
35	78.40	34	<.0001	-0.038	-0.012	0.027	0.032	-0.041	0.020
41	91.84	40	<.0001	0.050	-0.079	0.027	-0.021	0.055	-0.022
47	94.04	46	<.0001	0.008	-0.003	-0.015	-0.008	0.028	0.033

Figure 3.2.7 Crosscorrelation check table

3.2.2 Final TF model by SAS forecast system

After confirming parameters and noise model, *SAS* forecast system can be used to build the final model. Hold-out samples 30 is used, which is the same quantity as in the univariate model part. By Factored *ARIMA* Model, the result of b=0, r=0, s=1 and *MA(1)* noise model is shown in

Figure3.2.8 and *Figure3.2.9*. From *Figure3.2.8*, all parameters are significant and from *Figure3.2.9*, the MAPE is 0.854 and RMSE is 83.74 which means the model fits well. *Figure3.2.10* is the fitted plot, and it shows that most points are included in the fitting line.

HIGH

Close[N(1)] + IMA(1,1) NOINT

Model Parameter	Estimate	Std. Error	T	Prob>|T|
Moving Average, Lag 1	0.87336	0.0159	54.8909	<.0001
CLOSE[N(1)]	0.55454	0.0124	44.7400	<.0001
CLOSE[N(1)] Num1	-0.48321	0.0124	-39.0757	<.0001
Model Variance (sigma squared)	22616	.	.	.

Figure 3.2.8

Statistic of Fit	Value
Mean Square Error	7012.5
Root Mean Square Error	83.74079
Mean Absolute Percent Error	0.85366
Mean Absolute Error	70.93325

Figure 3.2.9

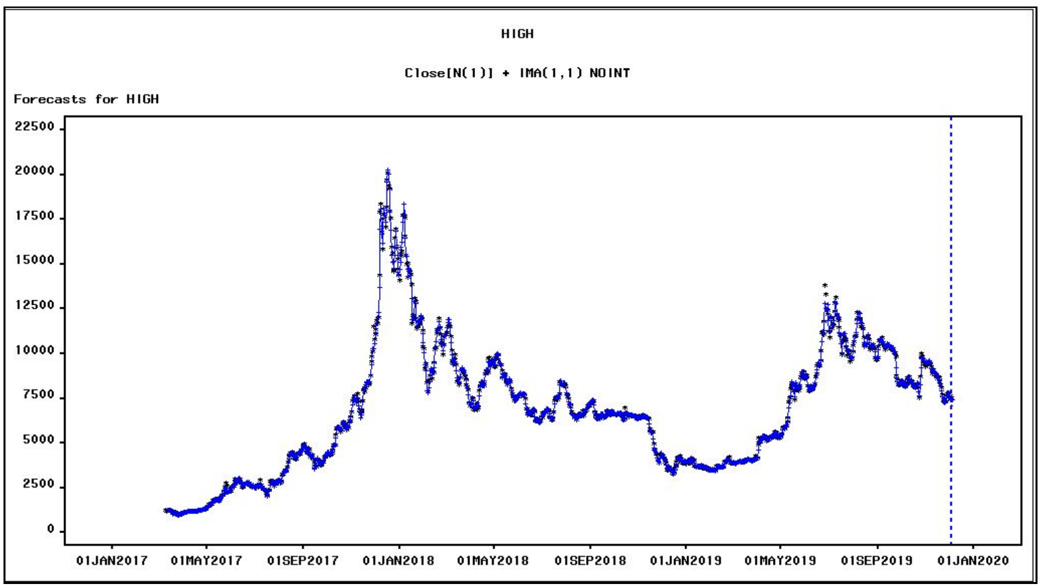


Figure 3.2.10

3.3 Multiple input TF model.

Besides exploring the relationship of *High* with *Volume* and *Close* individually, a multiple input model can be applied to see the relations between *High* as the dependent variable and *Volume* and *Close* together as the input variables. Based on b=r=s=0 when *Volume* is the single input variable and b=r=0, s=1 when *Close* is the single input variable, the multiple input model is built with these parameters unchanged, which is shown in *Figure3.3.1*.

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	300.29	6	<.0001	-0.510	0.051	-0.096	0.141	-0.092	-0.013
12	337.75	12	<.0001	0.057	0.012	-0.098	0.125	-0.091	0.015
18	372.75	18	<.0001	-0.046	0.090	-0.109	0.090	0.023	-0.061
24	392.31	24	<.0001	-0.023	0.048	0.032	-0.094	0.077	0.023
30	435.77	30	<.0001	-0.049	-0.081	0.160	-0.061	-0.054	0.035
36	441.12	36	<.0001	0.048	-0.049	-0.019	0.009	0.000	0.002
42	475.75	42	<.0001	0.052	-0.080	-0.008	0.079	0.018	-0.132
48	515.11	48	<.0001	0.064	0.064	-0.070	0.051	-0.082	0.122

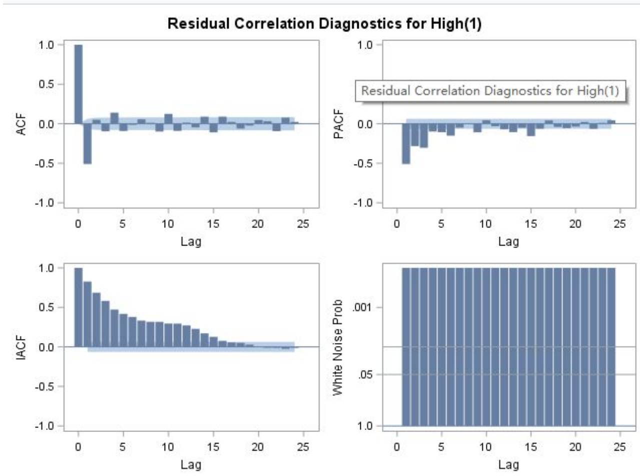


Figure 3.3.1 multiple inputs model

It is shown that the residuals are not white noise. Since *PACF* decayed quickly, *ACF* chopped off after lag1, *MA(1)* model is suitable for the noise model. After adding the noise model, the results are better than before. From *Figure3.3.2*, the residuals become wihte noise and the crosscorrelation check table also looks like properly from *Figure 3.3.3*.

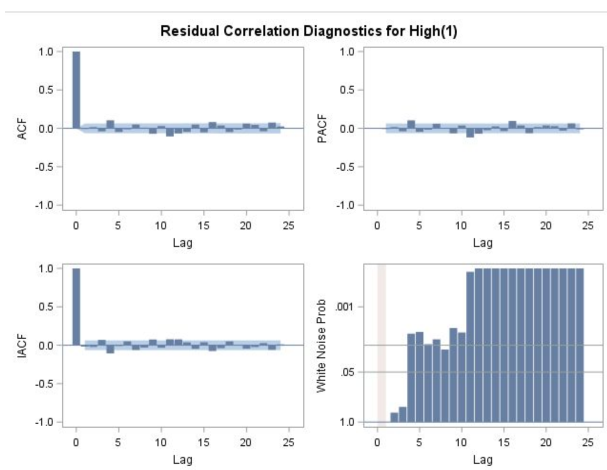


Figure 3.3.2 Residual plot after adding noise model

Crosscorrelation Check of Residuals with Input Volume									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	13.30	5	0.0207	-0.033	-0.046	-0.084	0.034	0.041	0.010
11	23.67	11	0.0142	-0.040	-0.020	0.077	-0.033	-0.034	0.015
17	39.34	17	0.0016	0.023	-0.033	0.066	-0.083	0.048	-0.021
23	55.62	23	0.0002	0.072	-0.054	0.039	-0.060	0.048	-0.027
29	58.58	29	0.0009	0.036	-0.021	0.008	-0.012	-0.029	-0.015
35	67.70	35	0.0007	-0.022	0.065	0.025	-0.034	0.049	-0.013
41	86.97	41	<.0001	-0.045	0.036	-0.052	0.082	-0.043	0.069
47	114.89	47	<.0001	-0.136	0.030	-0.021	0.078	-0.038	0.026

Crosscorrelation Check of Residuals with Input Close									
To Lag	Chi-Square	DF	Pr > ChiSq	Crosscorrelations					
5	12.15	4	0.0163	-0.000	0.000	-0.059	-0.033	-0.003	0.087
11	38.98	10	<.0001	-0.100	0.052	-0.017	0.027	0.095	-0.064
17	61.87	16	<.0001	-0.028	-0.016	0.018	0.074	0.038	-0.121
23	66.53	22	<.0001	-0.012	0.035	0.039	-0.032	-0.014	-0.024
29	83.71	28	<.0001	0.047	-0.029	0.073	-0.088	-0.014	0.030
35	93.55	34	<.0001	-0.024	0.006	0.037	0.012	-0.065	0.059
41	126.08	40	<.0001	0.047	-0.120	0.078	-0.035	0.067	-0.065
47	128.20	46	<.0001	0.023	-0.009	-0.027	0.017	0.021	0.009

Figure 3.3.3 Crosscorrelation check table

SAS forecasting system can be used to fit the model. The model still uses 30 hold-out samples and the final results shown in below. In *Figure3.3.4*, all parameters are significant and from *Figure3.3.5*, *MAPE* is 1.595 and *RMSE* is 179.396 which means the model fits properly. And *Figure3.3.6* shows that the fitting line goes through most of the points.

HIGH
VOLUME + Close[N(1)] + IMA(1,1) NOINT

Model Parameter	Estimate	Std. Error	T	Prob>|T|
Moving Average, Lag 1	0.83983	0.0178	47.1395	<.0001
VOLUME	3.50421E-8	1.73E-9	20.2554	<.0001
CLOSE[N(1)]	0.52660	0.0105	50.2569	<.0001
CLOSE[N(1)] Num1	-0.46260	0.0104	-44.4032	<.0001
Model Variance (sigma squared)	15915	.	.	.

Figure 3.3.4 Model parameter estimate table

Statistic of Fit	Value
Mean Square Error	32183.0
Root Mean Square Error	179.39615
Mean Absolute Percent Error	1.59531
Mean Absolute Error	126.38292

Figure 3.3.5 Error table

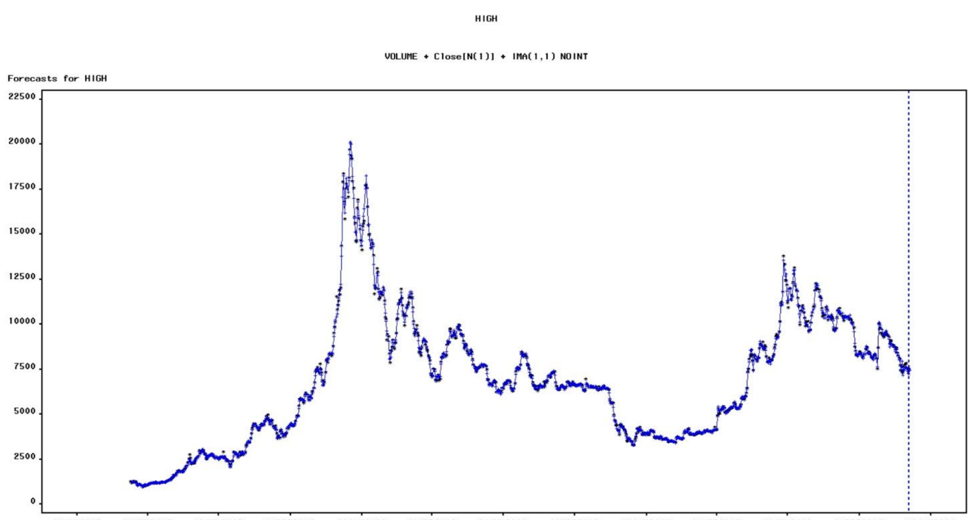


Figure 3.3.6 Forecasting plot for High

4. Conclusion

Since in *Section2.3*, *ARIMA(1,1,0)* performs best among the univariate models, the model comparison table, *Figure4.1*, only includes *ARIMA(1,1,0)* and TF models. From the perspective of fitting, TF models of *High-Close* and *High-Volume & Close* have significantly smaller Square Root Variance. From the perspective of prediction, the TF model of *High-Volume* has the smallest *MAPE* and *MAE* so that this model has the best performance to predict the High price of bitcoin. In conclusion, the *High-Close* TF model has the best performance on both prediction and fitting. But if there is only the data of *High*, then *ARIMA(1,1,0)* is the best choice.

Hold Out	**Models**	**MAPE**	**Type**	**MAE**	**RMSE**	**Square Root of model variance**
30	IAR(1,1)	1.49682	Univariate	123.32323	162.88315	342.78
30	TF model (High - Volume)	3.59741	Multivariate	280.09933	415.39348	314.21
30	TF model (High - Close)	0.85366	Multivariate	70.93325	83.74079	150.39
30	TF model (High - Volume & Close)	1.59531	Multivariate	126.38292	179.39615	126.15

Figure 4.1 Model comparison

5. Appendix

Cyclical model
1.High

```
DATA NEW;

SET WORK.HVS;

TIME=_N_;

HIGH=HIGH/1000;

*detrending the series;

PROC REG;

MODEL HIGH=TIME;

OUTPUT OUT=TRENDOUT R=DTUSE;

PROC SPECTRA DATA=TRENDOUT P;

VAR DTUSE;

PROC PRINT;

Run;

PROC GPLOT;

PLOT P_01*PERIOD;

SYMBOL I=JOIN;

RUN;

DATA NEWBITCOIN;

SET WORK.HVS;

TIME=_N_;

* CREATING SINE AND COSINE TERMS
AFTER LOOKING AT THE PERIODOGRAM;

COS1=COS(2*3.14159*TIME*1/1000);

SIN1=SIN(2*3.14159*TIME*1/1000);

COS2=COS(2*3.14159*TIME*2/1000);

SIN2=SIN(2*3.14159*TIME*2/1000);

COS3=COS(2*3.14159*TIME*3/1000);

SIN3=SIN(2*3.14159*TIME*3/1000);

COS4=COS(2*3.14159*TIME*4/1000);

SIN4=SIN(2*3.14159*TIME*4/1000);

COS5=COS(2*3.14159*TIME*5/1000);

SIN5=SIN(2*3.14159*TIME*5/1000);

COS6=COS(2*3.14159*TIME*6/1000);

SIN6=SIN(2*3.14159*TIME*6/1000);

COS7=COS(2*3.14159*TIME*7/1000);

SIN7=SIN(2*3.14159*TIME*7/1000);

COS9=COS(2*3.14159*TIME*9/1000);

SIN9=SIN(2*3.14159*TIME*9/1000);

COS10=COS(2*3.14159*TIME*10/1000);

SIN10=SIN(2*3.14159*TIME*10/1000);

COS13=COS(2*3.14159*TIME*13/1000);

SIN13=SIN(2*3.14159*TIME*13/1000);

COS16=COS(2*3.14159*TIME*16/1000);

SIN16=SIN(2*3.14159*TIME*16/1000);

* TRY THE REGRESSION HERE BEFORE YOU
MOVE TO THE SAS FORECASTING SYSTEM;

 PROC REG;
```

```
 MODEL HIGH=TIME COS1 SIN1 COS2 SIN2
COS3 SIN3 COS4 SIN4 COS5 SIN5 COS6
SIN6 COS7 SIN7COS9 SIN9 COS10 SIN10
COS13 SIN13 COS16 SIN16;

RUN;
```

2.periodogram

```
DATA NEWE;

SET WORK.NEWBITCOIN;

* SET SASUSER.HIGH;

TIME=_N_;

*REMOVE LAST 30 VALUES OF HIGH;

IF TIME>970 THEN HIGH=.;

*detrending the series;

PROC REG;

MODEL HIGH=TIME;

OUTPUT OUT=TRENDOUT R=DTIRC;

PROC SPECTRA DATA=TRENDOUT P;

VAR DTIRC;

PROC PRINT;

PROC GPLOT;

PLOT P_01*PERIOD;

SYMBOL I=JOIN;

RUN;
```

TF model:

1. High vs Volume:

```
DATA NEW;

SET WORK.Bitwhole;

PROC ARIMA;

IDENTIFY VAR=Volume;

IDENTIFY VAR=High;

RUN;

PROC ARIMA;

IDENTIFY VAR=volume(1);

IDENTIFY VAR=high(1);

RUN;

PROC ARIMA;

IDENTIFY VAR=volume(1) NOPRINT;

ESTIMATE Q=2 NOCONSTANT METHOD=ML;

IDENTIFY VAR=high(1)
CROSSCOR=volume(1);

RUN;

PROC ARIMA;

IDENTIFY VAR=volume(1) NOPRINT;

ESTIMATE Q=2 NOCONSTANT METHOD=ML;

IDENTIFY VAR=high(1)
CROSSCOR=volume(1);

ESTIMATE INPUT=(0$(0)/volume)
Q=(1,10) NOCONSTANT METHOD=ML;

RUN;
```

2. High vs Close:

```
DATA NEW;

SET WORK.Bitwhole;

PROC ARIMA;

IDENTIFY VAR=close;

IDENTIFY VAR=High;

RUN;
```

```
PROC ARIMA;                                    RUN;

IDENTIFY VAR=close(1);

IDENTIFY VAR=high(1);

RUN;

PROC ARIMA;

IDENTIFY VAR=close(1) NOPRINT;

ESTIMATE NOCONSTANT METHOD=ML;

IDENTIFY VAR=high(1)
CROSSCOR=close(1);

RUN;

PROC ARIMA;

IDENTIFY VAR=close(1) NOPRINT;

ESTIMATE NOCONSTANT METHOD=ML;

IDENTIFY VAR=high(1)
CROSSCOR=close(1);

ESTIMATE INPUT=(0$(0)/(1)close) Q=1
NOCONSTANT METHOD=ML;

RUN;
```

3. High vs Close & Volume:

```
PROC ARIMA;

IDENTIFY VAR=close(1) NOPRINT;

ESTIMATE NOCONSTANT METHOD=ML;

IDENTIFY VAR=volume(1) NOPRINT;

ESTIMATE Q=2 NOCONSTANT METHOD=ML;

IDENTIFY VAR=high(1)
CROSSCOR=(volume(1)close(1))
NOPRINT;

ESTIMATE INPUT=(0$(0)/volume
0$(1)/(0)close) Q=1 NOCONSTANT
METHOD=ML;
```