**Time Series Analysis on Cryptocurrencies and Trading Strategy**

# **Table of Contents**

[**Table of Contents** 1](#_Toc116047270)

[**Terminology** 2](#_Toc116047271)

[**Volatility of Cryptocurrencies** 3](#_Toc116047272)

[**Before diving into computations** 3](#_Toc116047273)

[**About the data** 3](#_Toc116047274)

[**Box-Jenkins Method** 4](#_Toc116047275)

[**Visualizing the Data** 4](#_Toc116047276)

[**Decomposition** 5](#_Toc116047277)

[**Augmented Dickey-Fuller Test** 6](#_Toc116047278)

[**Stationarize the data** 7](#_Toc116047279)

[**Build Model** 9](#_Toc116047280)

[**Residual Analysis** 10](#_Toc116047281)

[**Make Predictions** 11](#_Toc116047282)

[**Correlation of one cryptocurrencies with the variation of another** 12](#_Toc116047283)

[**Turtle Trading Strategy** 14](#_Toc116047284)

[**Statistical Testing** 17](#_Toc116047285)

[**Summarize results** 18](#_Toc116047286)

[**Further Exploration** 18](#_Toc116047287)

# 

# **Terminology**

Below are a few terminologies from the context of the project -

* **Cryptocurrency/Crypto** - A peer-to-peer system that can enable anyone anywhere to send and receive payments. It is a digital currency payment system that doesn’t rely on banks to verify payment transactions.
* **Bitcoin -** A type of decentralized digital cryptocurrency. It is valued the highest among all the other crypto and is possibly the most well-known as well.
* **Altcoins** - Cryptocurrencies other than bitcoin.
* **Open Price -** This refers to the price of the cryptocurrency at the start of an interval. For a time interval *[t₁, t₂],* theopen price is the price of the cryptocurrency at the time *t₁.*
* **Close Price -** This refers to the price of the cryptocurrency at the end of an interval. For a time interval *[t₁, t₂],* theopen price is the price of the cryptocurrency at the time *t₂.*
* **High Price -** This refers to the highest price of the cryptocurrency for an interval. For a time interval *[t₁, t₂],* the high price is the highest price crypto obtained in the interval *[t₁, t₂].*
* **Low Price -** This refers to the lowest price of the cryptocurrency for an interval. For a time interval *[t₁, t₂],* the high price is the lowest price crypto obtained in the interval *[t₁, t₂].*

# 

# **Volatility of Cryptocurrencies**

Since the 1st of November, bitcoin has been trading between 69k USD and 59k USD. 10k USD is an enormous fluctuation of prices for just 15 odd days. For comparison, the S & P 500 index has only risen by 2.5k USD in the last five years. Other cryptocurrencies also show similar kinds of volatilities. Many people have also lost massive amounts of money by investing in fraud altcoins by cases of rug pull. The prices of cryptos can change due to multiple reasons. Some of these include -

* The supply-demand variations of cryptocurrencies
* The cost of producing crypto by mining it
* Regulations and other rumors surrounding the development of a cryptocurrency.

For instance, since the RBI’s stern stand against the unregulated trading of cryptocurrencies and its opinion of banning it altogether, bitcoin and other altcoins have seen a steady decrease in price.

# **Before diving into computations**

## **About the data**

The data was extracted from the [bitfinex API](https://api.bitfinex.com/). It was 5-year data composed of 1740 rows, which had 4 columns. For each date we had open, high, low and close prices for the day.

## **Box-Jenkins Method**

This is a classical approach for fitting an ARIMA model. This method applies an autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models to find the best fit of a time-series model. Necessary steps are taken before arriving at the correct model that has to be used for the analysis.

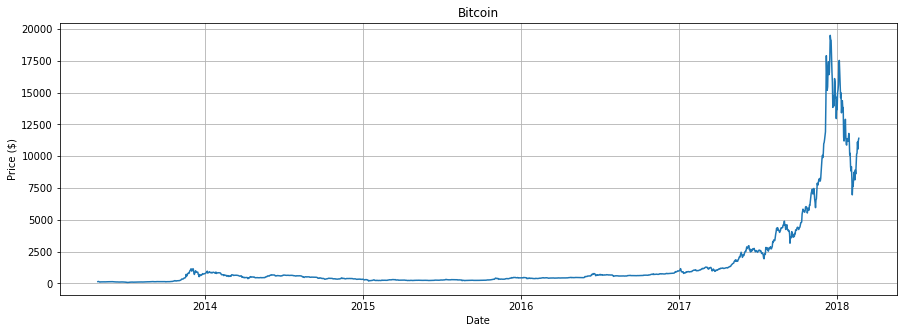
The entire process mainly consists of 3 steps -

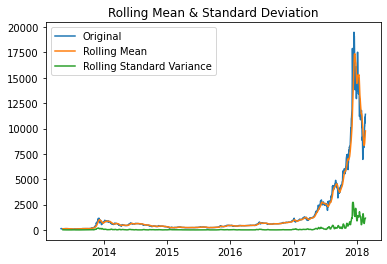
1. **Identification -**  Use the data and extract all the related information to help select a shortlist of the subclass of models that may best fit the data.
   1. Make sure the data is stationarized. Apply necessary transformations if it isn’t stationary already.
   2. Plot the ACF and PACF of the stationarized data to find proper parameters for time series forecasting.
2. **Estimation -** Train the model on the data to arrive at the best approximation of the coefficients.
3. **Diagnostic Checking -** Test this trained model on the available test data and search for areas where there is a scope of improvement.

These steps are iterated until a desirable fit is achieved.

# **Visualizing the Data**

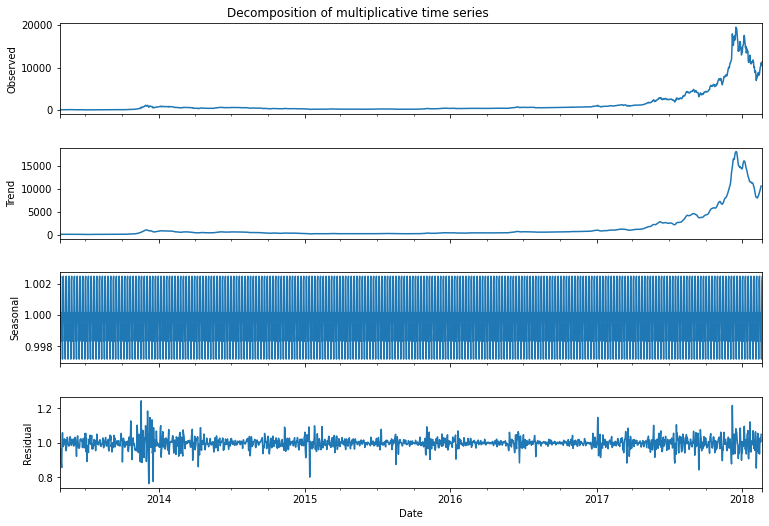
Let’s now visualize the data to get better insights and also understand what type of model we need to use. We have used just the normal plot of the data, the plots of 12 days rolling mean and standard deviation.



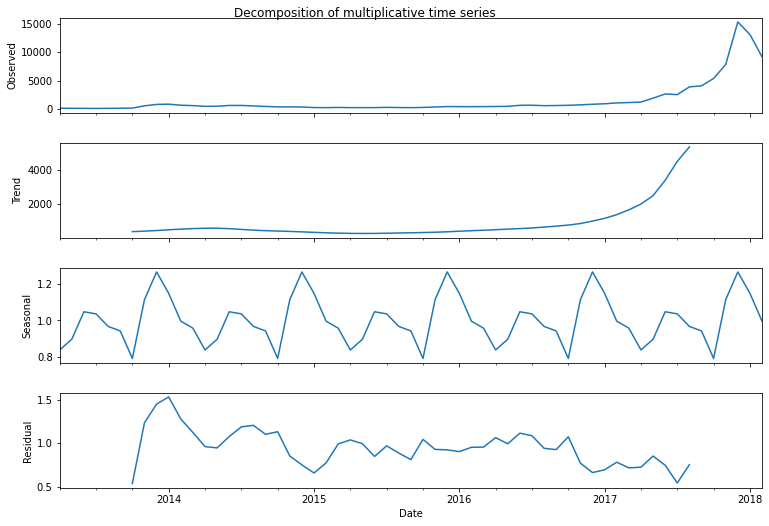


# **Decomposition**

We performed a multiplicative decomposition of time series to obtain the observed, trend, seasonal and residual plots of the data components of the data. We used [statsmodel API](https://www.statsmodels.org/stable/index.html) to perform the operations.



We can see that the seasonal component of the data is not visible in the above decomposition. We resampled data by the month and analyzed it again and the seasonal component is clearly visible.



# **Augmented Dickey-Fuller Test**

The Augmented Dickey-Fuller test is used to test the null hypothesis that a unit root is present in an autoregressive model, i.e, the time series is non-stationary . The alternative hypothesis is stationarity.

We used the *adfuller()* function of python to perform the test and the results were as follows:

Results of Dickey-Fuller Test:

==============================================

Test Statistic -1.360453

p-value 0.601082

#lags Used 25.000000

Number of Observations Used 1734.000000

Critical Value (1%) -3.434127

Critical Value (5%) -2.863208

Critical Value (10%) -2.567658

It was quite evident that the null hypothesis was accepted as . This means that the time series data we are analyzing has a unit root and the data isn’t stationary yet.

# **Stationarize the data**

We tried to stationarize the data by converting the data to month-wise data. The null hypothesis in the ADF test was accepted and the data wasn’t stationary yet.

We then took the difference in logarithmic data, which gave a when we ran the ADF test. This meant that the null hypothesis was rejected and the data was stationarized.

Results of Dickey-Fuller Test:

==============================================

Null hypothesis H\_0 rejected - The data is stationary

Test Statistic -5.098131

p-value 0.000014

#lags Used 0.000000

Number of Observations Used 57.000000

Critical Value (1%) -3.550670

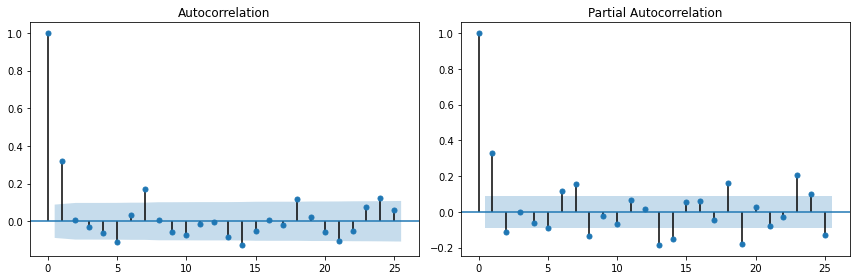
Critical Value (5%) -2.913766

Critical Value (10%) -2.594624

**Plot the ACF and PACF charts and find the optimal parameters**

* **Autocorrelation Function (ACF)** - This plot describes the correlation between observation and another observation at a prior time step, called lags. This includes direct and indirect dependence formation.
* **Partial Autocorrelation Function (PACF)** - This plot describes the correlation between an observation and lags that results by removing correlation effects due to terms in shorter lags.

Since we calculate the correlation for time series observation with observations of the previous steps, these correlations are also called serial correlations, or autocorrelation. Below are the plots of ACF and PACF of our time series data.

We can arrive at some key coefficients by carefully examining these plots -

● Autoregressive (AR with lag k) if the ACF trails off after a lag of k and correlation suddenly drops after lag k in the PACF. This lag value is taken as p.

● Moving Average (MA with lag k) if the PACF trails off after a lag of k and correlation suddenly drops after a log of k in the ACF. ​This lag value is taken as q.

● The model is a mix of AR and MA if both the ACF and PACF trail off.

On careful observation, we conclude that the appropriate values of p and q are both 1.

Now we proceed to fit an ARIMA(1,1,1) model into our data.

# **Build Model**

ARIMA Model Results

==============================================================================

Dep. Variable: D.Close No. Observations: 1759

Model: ARIMA(1, 1, 1) Log Likelihood -12036.324

Method: css-mle S.D. of innovations 226.728

Date: Wed, 17 Nov 2021 AIC 24080.648

Time: 17:04:12 BIC 24102.538

Sample: 02-19-2018 HQIC 24088.738

- 04-28-2013

=================================================================================

coef std err z P>|z| [0.025 0.975]

---------------------------------------------------------------------------------

const -6.3766 5.762 -1.107 0.269 -17.670 4.917

ar.L1.D.Close -0.7045 0.054 -12.946 0.000 -0.811 -0.598

ma.L1.D.Close 0.8168 0.043 18.779 0.000 0.732 0.902

Roots

=============================================================================

Real Imaginary Modulus Frequency

-----------------------------------------------------------------------------

AR.1 -1.4194 +0.0000j 1.4194 0.5000

MA.1 -1.2243 +0.0000j 1.2243 0.5000

-----------------------------------------------------------------------------

We used the [statsmodel](https://www.statsmodels.org/stable/index.html) library to build the model. We built the model for different sets of (p, d, q) in order to find the triplets that give us the best fit. We had anticipated that we should be using an ARIMA model based on the ACF and PACF plots. After obtaining the results we found that the ARIMA model was indeed the one we are required to go about with since it gave desirable results.

The table above describes the values of other parameters of an ARIMA model.

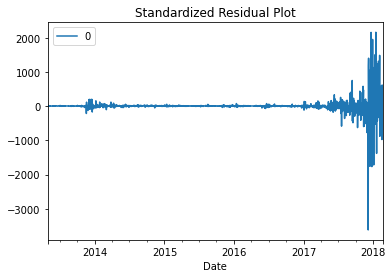
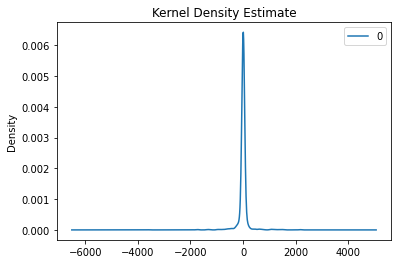
# **Residual Analysis**

We have tried to see if there is any residual trend left in the mode that we have not extracted by trying to plot the residuals in the model.

The **standardized residual plot** in the left shows that the residuals appear to be a horizontal line passing through zero on the y-axis and fluctuate after that without any noticeable trend or pattern. Hence, we can conclude that the residuals are completely random.

The **kernel density plot** on the right shows high density close to the origin and low density for the points away from the origin and it is also symmetric about the origin. Hence, even from this plot, we can be sure that there is no residual left in the model.

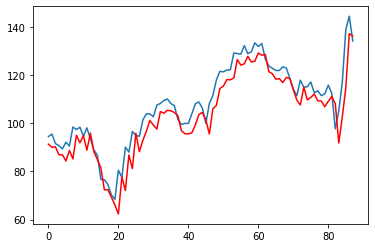
At this point, we can be sure that ARIMA is the right choice of model since it efficiently extracts data trends and patterns without leaving any residual.

****

# **Make Predictions**

Since we have now finalized the model and concluded that the model we have chosen is the right one we can now make predictions. By splitting the training and the test set in the ratio of 95:5 we were able to obtain the below results.

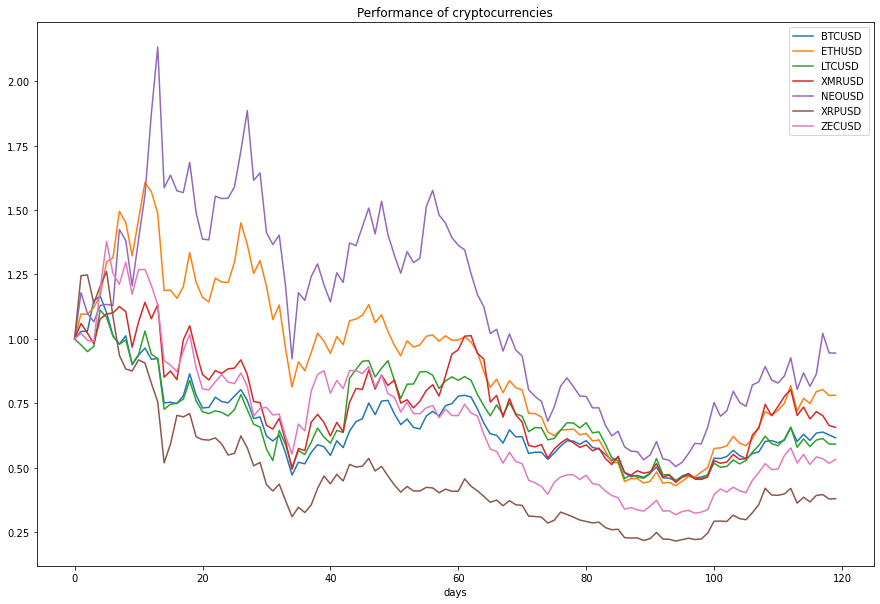
The model slightly underestimates the prices. If someone were to short the bitcoin this can be problematic. But on the other hand, if we were to hold a long position our model shouldn’t cause any problem.

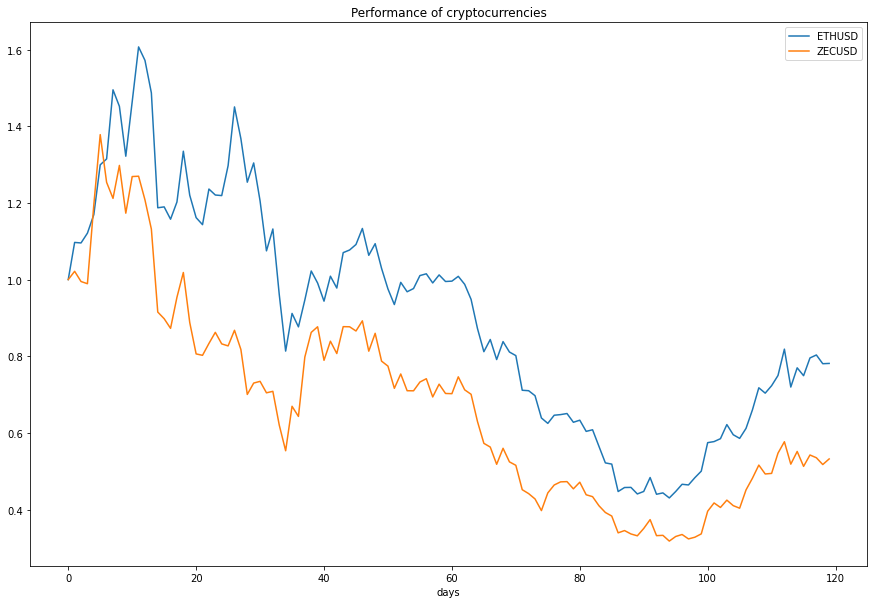


# **Correlation of one cryptocurrencies with the variation of another**

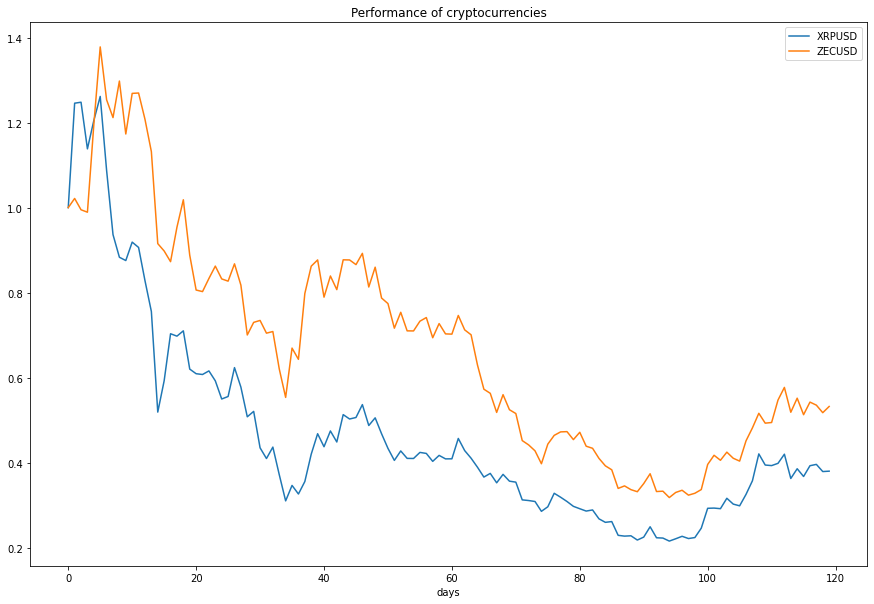
It is often the case that the variation of one cryptocurrency leads to a major variation in the other. We tried to analyse the same.

We plotted data over the last 120 days of various cryptocurrencies ( BTCUSD, ETHUSD, LTCUSD, XMRUSD, NEOUSD, XRPUSD, ZECUSD ) and the plot can be found below:

Next we found out the correlation between various cryptocurrencies. While some showed considerable correlation with another, the others did not. The P value varied between . This can further be used to optimize strategies and devise low-risk trading techniques.



ETHUSD and ZECUSD: p-value = 0.008516 (High Correlation)



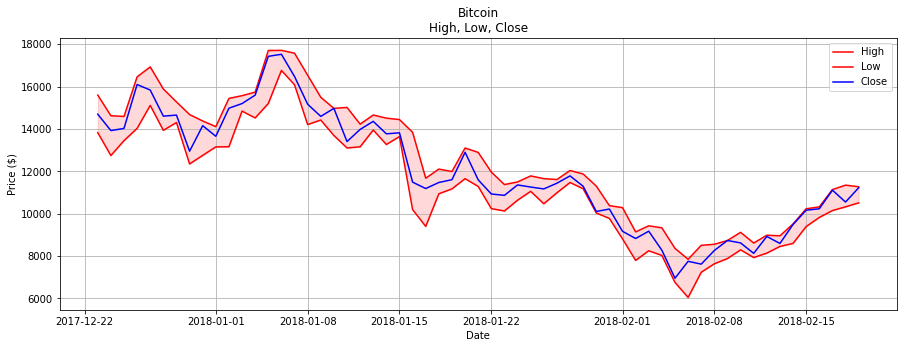
ZECUSD and XRPUSD: p-value = 0.893919 (Low Correlation)

# **Turtle Trading Strategy**

We used the simplest trading strategy called the turtle trading strategy. It achieved tremendous success when it was found and is in use till date.

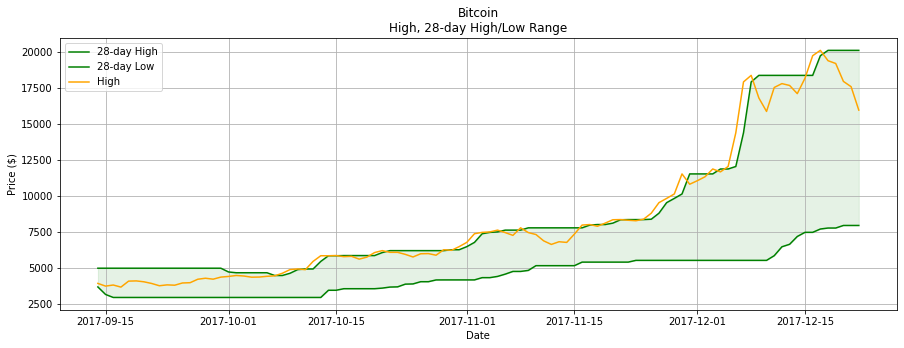
It is one of the most famous trend-following strategies and is based on purchasing stocks during breakout and selling on retracement.

Continuing the analysis of Cryptocurrency in order to get a closer look at a smaller slice of time (60 days) where we can see the High and Low plotted as well



In order to examine the simplified Turtle trading strategy we need to create two new rows of data, one for the 28-day High and one for the 28-day Low.

Now let's look at the same small window we did before only with the High price plotted with the 28-day High/Low range



So we see here that the green range is the 28-day High/Low and the orange line is the High for the day. So the signals we are looking for are anytime the orange line breaks above the green line. This is our entry point for the trading strategy.

Now that we have the data we need and have gotten to take a look at it in a few general ways, let's create the data that we need in order to test the actual trading strategy. The end goal is to have a list of price changes from each entry point as described above, to a period in time 28 days later. That way we can see if breakouts above the 28-day high are leading to higher prices.

So over the entire period we counted ***235 trades*** with the ***average price change being $592.82***

# **Statistical Testing**

We now have the data we want and the sample mean. It is time to pose the hypothesis and test it.

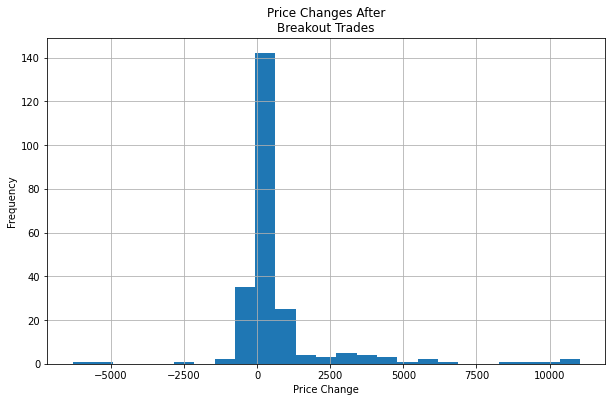
***Define Null and Alternative Hypotheses***

**Null Hypothesis:** Price change is random and therefore any mean of sample price differences is zero.

**Alternative Hypothesis:** A breakout above a 28-day high leads to higher prices and therefore the mean of the sample of price differences is significantly greater than zero.

**Check data for Normal Distribution**

In this example our underlying assumption is that price change is random and therefore is normally distributed. We want to test whether our sample data falls in line with our assumption. Here is a bar chart of the sample data so we can quickly inspect whether or not we believe it is normally distributed before applying the test



Because of the significant outliers it doesn't appear that the data is normally distributed. Confirm this below with a Kolmogorov–Smirnov Test for goodness-of-fit.

KstestResult(statistic=0.2800505431028666, p value=8.586435881649918e-17)

Given the p-value is basically zero, we reject the null hypothesis that the data follow a normal distribution. Thus we cannot use a t-test here.

So we have to find another test to use and in this case we believe the Wilcoxon rank-sum test is appropriate. It tests whether two sets of measurements are drawn from the same distribution. In this case the first set would be the hypothetical set of 235 trades that had $0 price change after 28 days. The second set is our data.

# **Summarize results**

Given the positive test statistic and p-value below 0.05 we reject the null hypothesis that the two samples came from the same distribution. Because this is a two-sided test the positive statistic shows that after entry of our sample trades the price of Bitcoin did tend to rise in the next 28 days.

By visually inspecting the data this seems to be confirmed. However, for us, it is quite surprising to see the results not skewed even more to the positive side. This is a market that has been in a clear uptrend and when it does go, it moves violently. The high variance is certainly expected and seeing some of the significant price changes over the course of these four-week periods gives clear confirmation that this is an extremely volatile market.

# **Further Exploration**

There are a number of places to go from here as far as continuing and improving the analysis in order to make some real trading decisions down the road. We will list a few here and hopefully explore them soon or as part of future projects.

What about the short side of things? This analysis only took into account long trades for simplicity but it would be interesting to see if break-outs below the lows of the range showed similar results.

The real Turtle traders used a breakout system similar to this one to determine their entry points into the market. However, entry point is only one of many aspects that make up a real trading plan. We want to further explore setting risk and profit targets based on price volatility and then seeing what the overall expected value of the trading strategy would be.

Why confine the results to Bitcoin? A robust trading strategy should be able to show positive expectancy for other cryptocurrencies, stocks, commodities, etc. Abstracting this study and then using it on other products is a real world project that we are sure many people are pursuing with all kinds of strategies right now.