Cryptocurrencies – Exploratory Data Analysis

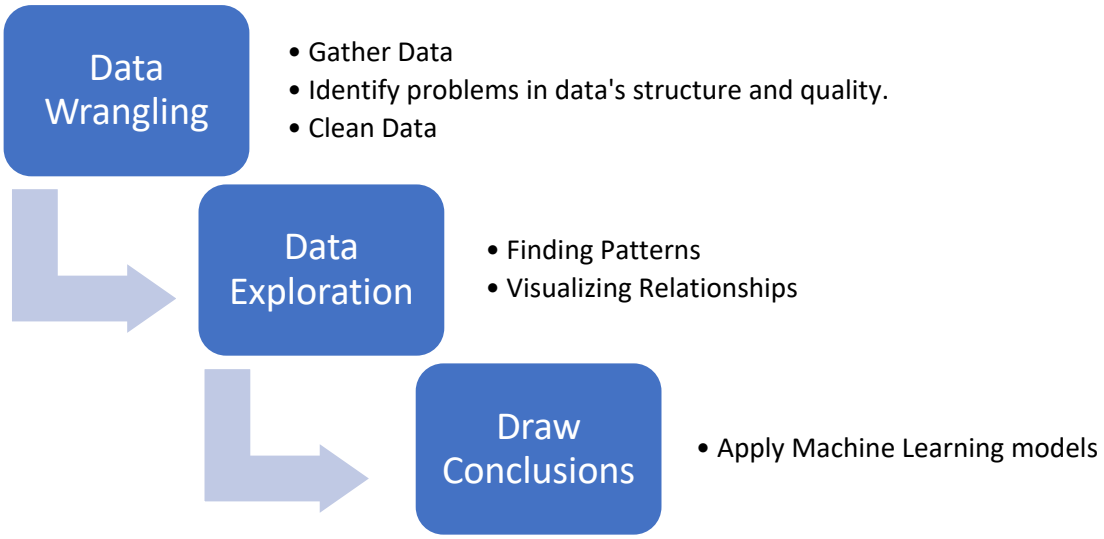
Smeet Chitalia - 1637694

Exploratory Data Analysis (EDA) is an approach for analyzing data using a variety of techniques. The purpose of Exploratory Data Analysis is to summarize visualizations and statistics to better understand data, its quality and structure. The goal here is to try and answer all questions with data.

EDA is considered to be an iterative process since assumptions are first made on the exploratory visualizations, then build some models. We then make visualizations of the model results and tune our models. The various techniques used in EDA for analyzing data are:

1. Maximize insight into a data set
2. Understand structure of data
3. Extract important variables
4. Detect outliers and anomalies
5. Develop machine learning models
6. Determine optimal factor settings

Exploratory Data Analysis (EDA) Process:



Data Wrangling

1. *Gather Data*

The Cryptocurrencies Historical Prices data set is available on Kaggle. This particular data consists of the prices for more than 1000 different cryptocurrencies.

There are a total of 651,366 entries and 10 columns in the dataset.

The data in the columns is:

Date – the day of recorded values
Open – the opening price (in USD)
High – the highest price (in USD)
Low – the lowest price (in USD)
Close – the closing price (in USD)
Volume – total exchanged volume (in USD)
Market.Cap – the total market capitalization for the coin (in USD)
Coin – the name of the coin
Delta – calculated as (Close – Open) / Open

```
Data columns (total 10 columns):
Unnamed: 0    651366 non-null int64
Date          651364 non-null object
Open          651014 non-null float64
High          650991 non-null float64
Low           651042 non-null float64
Close         651014 non-null float64
Volume        651366 non-null object
Market.Cap    651366 non-null object
coin          651366 non-null object
Delta         650998 non-null float64
dtypes: float64(5), int64(1), object(4)
```

After analyzing the data, the datatypes of the columns in the data set are:

Float64 datatype	Int64 datatype	Object datatype
Open	Unnamed: 0	Date
High		Volume
Low		Market.Cap
Close		coin
Delta		

2. *Identify problems in data's quality or structure*

The data set contains 651,366 entries and 10 columns. The column names are as follows:

```
Index(['Unnamed: 0', 'Date', 'Open', 'High', 'Low', 'Close', 'Volume',
       'Market.Cap', 'coin', 'Delta'],
      dtype='object')
```

The above column image shows a 'Unnamed: 0' table of int64 datatype which corresponds to the index of the rows. So, it hinders the structure of the data which is why it should be dropped in the data cleaning step.

	Unnamed: 0	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta
0	1	2018-01-04	15270.7	15739.7	14522.2	15599.2	21,783,200,000	256,250,000,000	BTC	0.021512
1	2	2018-01-03	14978.2	15572.8	14844.5	15201.0	16,871,900,000	251,312,000,000	BTC	0.014875
2	3	2018-01-02	13625.0	15444.6	13163.6	14982.1	16,846,600,000	228,579,000,000	BTC	0.099604
3	4	2018-01-01	14112.2	14112.2	13154.7	13657.2	10,291,200,000	236,725,000,000	BTC	-0.032242
4	5	2017-12-31	12897.7	14377.4	12755.6	14156.4	12,136,300,000	216,326,000,000	BTC	0.097591

Also, there are some null values in the data set since the number of entries for each column are not equal.

	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta
651361	2017-11-22	0.001781	0.001781	0.000339	0.001032	2,706	-	IPY	-0.420550
651362	2017-11-21	0.038891	0.039477	0.017002	0.017092	58,003	-	IPY	-0.560515
651363	2017-11-20	0.049463	0.058766	0.038952	0.038952	95,040	-	IPY	-0.212502
651364	NaN	NaN	NaN	NaN	NaN	No data was found for the selected time period.	No data was found for the selected time period.	QC	NaN
651365	NaN	NaN	NaN	NaN	NaN	No data was found for the selected time period.	No data was found for the selected time period.	FRCT	NaN

After further analysis, there are some column entries where data was not collected for that time period. All these entries will be deleted in the data cleaning step.

3. *Clean Data*

The ‘Unnamed: 0’ column is dropped from the data set.

```
df.drop(['Unnamed: 0'], axis=1, inplace=True)
```

```
df.head()
```

	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta
0	2018-01-04	15270.7	15739.7	14522.2	15599.2	21,783,200,000	256,250,000,000	BTC	0.021512
1	2018-01-03	14978.2	15572.8	14844.5	15201.0	16,871,900,000	251,312,000,000	BTC	0.014875
2	2018-01-02	13625.0	15444.6	13163.6	14982.1	16,846,600,000	228,579,000,000	BTC	0.099604
3	2018-01-01	14112.2	14112.2	13154.7	13657.2	10,291,200,000	236,725,000,000	BTC	-0.032242
4	2017-12-31	12897.7	14377.4	12755.6	14156.4	12,136,300,000	216,326,000,000	BTC	0.097591

The column entries containing null values or no data are also dropped from the dataset to maintain consistency. Also, year has been extracted from date column to perform separate analysis on particular year’s data.

```
df_crypto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 562621 entries, 0 to 651039
Data columns (total 10 columns):
Date           562621 non-null object
Open           562621 non-null float64
High           562621 non-null float64
Low            562621 non-null float64
Close          562621 non-null float64
Volume         562621 non-null object
Market.Cap     562621 non-null object
coin           562621 non-null object
Delta          562621 non-null float64
Year           562621 non-null object
dtypes: float64(5), object(5)
memory usage: 47.2+ MB
```

```
df_crypto.head()
```

	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta	Year
0	2018-01-04	15270.7	15739.7	14522.2	15599.2	21,783,200,000	256,250,000,000	BTC	0.021512	2018
1	2018-01-03	14978.2	15572.8	14844.5	15201.0	16,871,900,000	251,312,000,000	BTC	0.014875	2018
2	2018-01-02	13625.0	15444.6	13163.6	14982.1	16,846,600,000	228,579,000,000	BTC	0.099604	2018
3	2018-01-01	14112.2	14112.2	13154.7	13657.2	10,291,200,000	236,725,000,000	BTC	-0.032242	2018
4	2017-12-31	12897.7	14377.4	12755.6	14156.4	12,136,300,000	216,326,000,000	BTC	0.097591	2017

The cleaned data set contains 562,621 entries with 10 columns.

After analyzing the data, the datatypes of the columns in the cleaned data set are:

Float64 datatype	Object datatype
Open	Date
High	Volume
Low	Market.Cap
Close	coin
Delta	Year

Data Exploration

1. *Finding Patterns*

The exploratory analysis is done on the data for the year 2017 which has 253,759 entries and 10 columns.

	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta	Year
4	2017-12-31	12897.7	14377.4	12755.6	14156.4	12,136,300,000	2.163260e+11	BTC	0.097591	2017
5	2017-12-30	14681.9	14681.9	12350.1	12952.2	14,452,600,000	2.462240e+11	BTC	-0.117812	2017
6	2017-12-29	14695.8	15279.0	14307.0	14656.2	13,025,500,000	2.464280e+11	BTC	-0.002695	2017
7	2017-12-28	15864.1	15888.4	13937.3	14606.5	12,336,500,000	2.659880e+11	BTC	-0.079273	2017
8	2017-12-27	16163.5	16930.9	15114.3	15838.5	12,487,600,000	2.709760e+11	BTC	-0.020107	2017

Here, I've extracted the month from the 'Year' column to make monthly analysis possible.

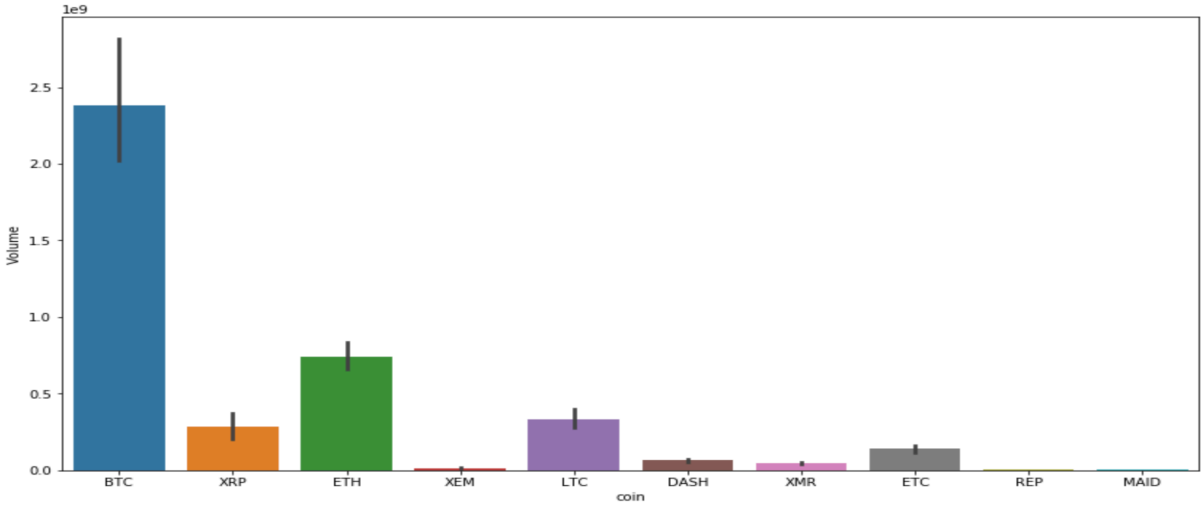
```
df_top['Month'] = df_top['Date'].apply(lambda x: x.split('-')[1])
```

	Date	Open	High	Low	Close	Volume	Market.Cap	coin	Delta	Year	Month
4	2017-12-31	12897.7	14377.4	12755.6	14156.4	12,136,300,000	2.163260e+11	BTC	0.097591	2017	12
5	2017-12-30	14681.9	14681.9	12350.1	12952.2	14,452,600,000	2.462240e+11	BTC	-0.117812	2017	12
6	2017-12-29	14695.8	15279.0	14307.0	14656.2	13,025,500,000	2.464280e+11	BTC	-0.002695	2017	12
7	2017-12-28	15864.1	15888.4	13937.3	14606.5	12,336,500,000	2.659880e+11	BTC	-0.079273	2017	12
8	2017-12-27	16163.5	16930.9	15114.3	15838.5	12,487,600,000	2.709760e+11	BTC	-0.020107	2017	12

The information of the new data for the year 2017 is as follows:

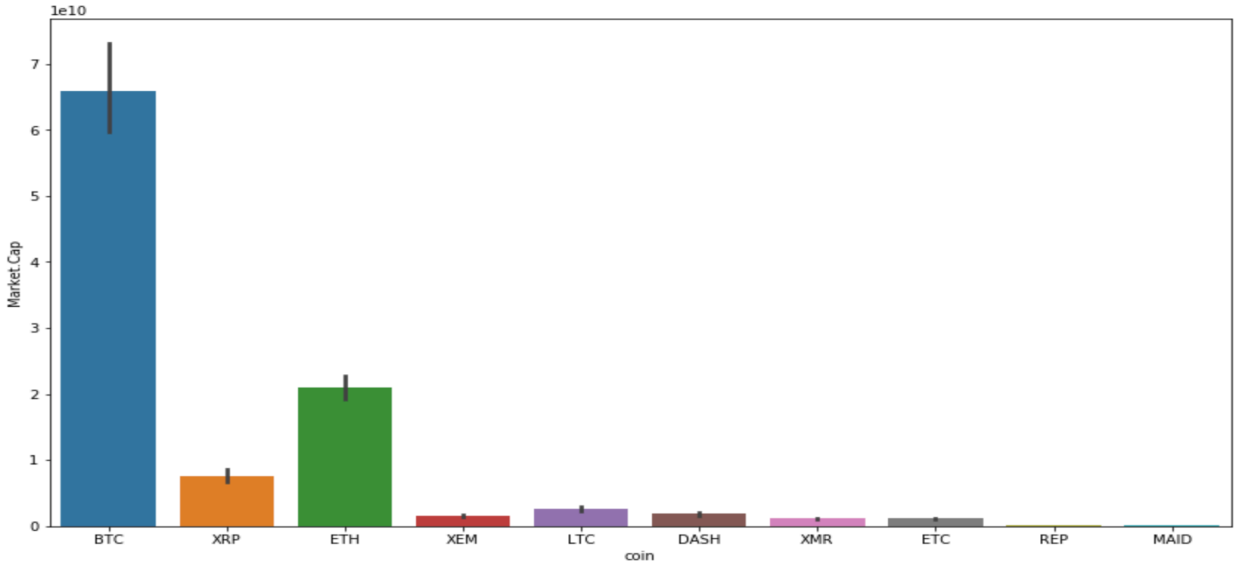
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3650 entries, 4 to 40885
Data columns (total 11 columns):
Date           3650 non-null object
Open           3650 non-null float64
High           3650 non-null float64
Low            3650 non-null float64
Close          3650 non-null float64
Volume         3650 non-null object
Market.Cap     3650 non-null float64
coin           3650 non-null object
Delta          3650 non-null float64
Year           3650 non-null object
Month          3650 non-null object
dtypes: float64(6), object(5)
memory usage: 342.2+ KB
```

The graph of volume of each cryptocurrency for the year 2017 is as follows:



The above graph shows that Bitcoin (BTC) tops the volume followed by Ethereum (ETH) & Litecoin (LTC). Those are followed by Ripple (XRP) and then Ethereum Classic (ETC).

The graph of Market Capitalization of each cryptocurrency for the year 2017, the value changes and many other cryptocurrencies come into the picture.



The top 3 cryptocurrencies remain the same Bitcoin (BTC) , Ethereum (ETH) followed by Ripple (XRP). But all the other cryptocurrencies are almost on the same level.

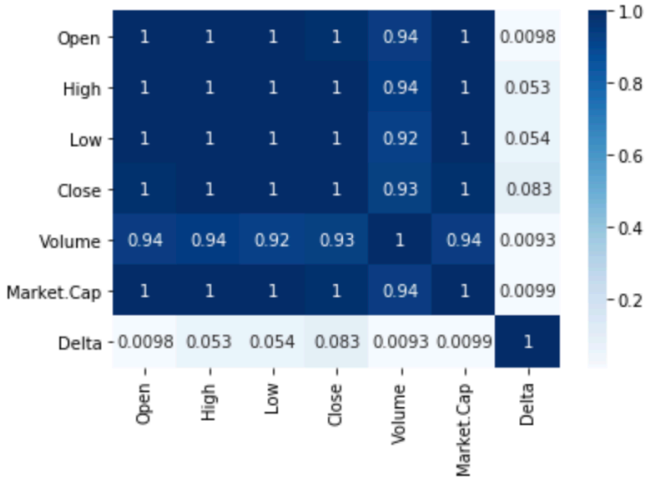
Here, I'm considering top 3 cryptocurrencies (Market Capitalization) for further analysis.

1. Bitcoin (BTC)
2. Ethereum (ETH)
3. Ripple (XRP)

2. *Visualizing Relationships*

Bitcoin (BTC)

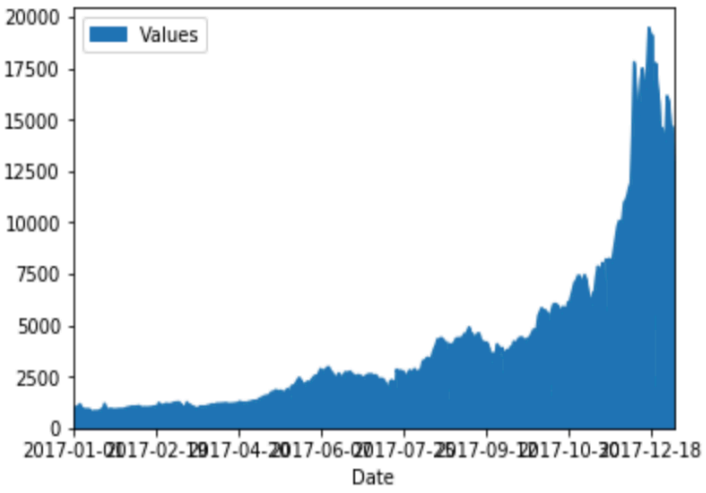
The Bitcoin correlation table is as follows.



Here, you can clearly see that 'Open' with 'Delta' has the lowest correlation value of 0.0098.

So, lets consider an area graph of 'Open' Attribute and make suitable analysis based On the graph.

The Open area graph is as follows:

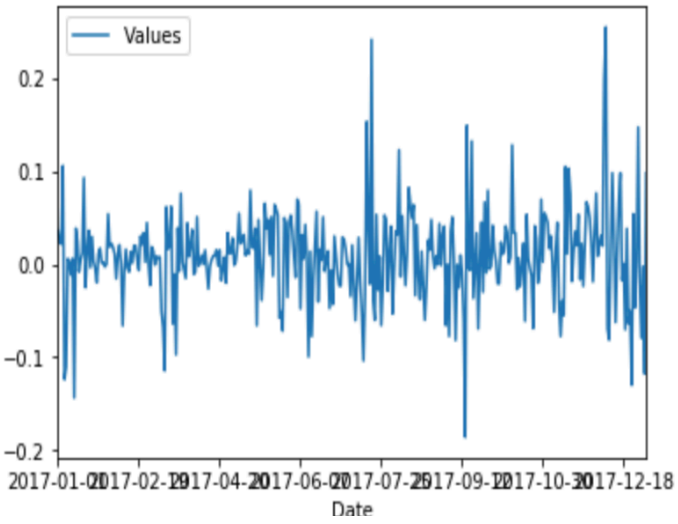


Here, the graph indicates that in the initial months of the year, you can see that the value of Bitcoin is pretty much stable but after that the value starts to fluctuate and then it shows an exponential increase and a sudden decrease.

To deduce further conclusions, let's have a look at the Delta graph.

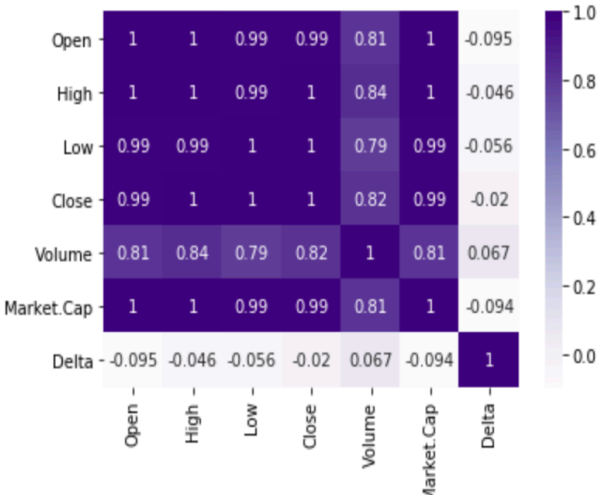
The 'Delta' graph indicates that the value of Bitcoin fluctuates a lot on daily basis.

Towards the end of the year, the graph shows that value is fluctuating very much in both directions.



Ethereum (ETH)

The Ethereum correlation table is as follows.

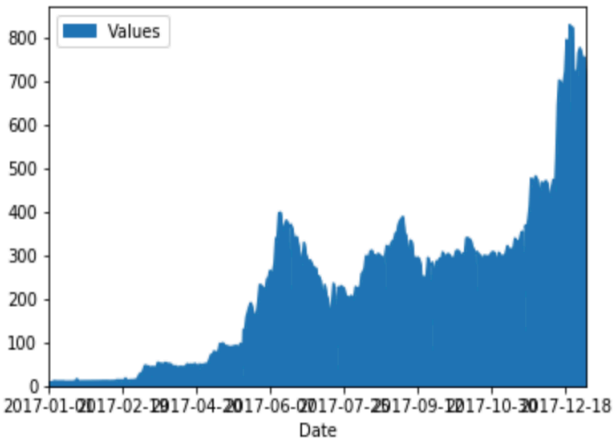


Here, you can clearly see that 'Open' with 'Delta' has the lowest correlation value -0.095.

Also, if you observe the correlation table, all the other values are very high.

Let's consider an area graph of 'Open' attribute and make suitable analysis based on the graph.

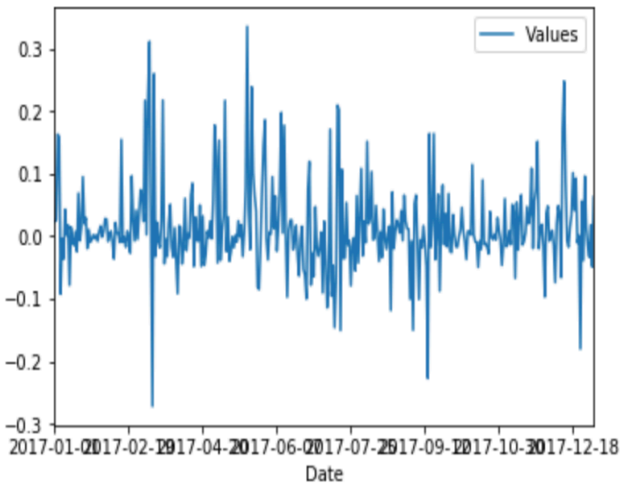
The Open area graph is as follows:



This graph shows that the value of Ethereum prices have sudden increases and decreases throughout the year.

Also, in the initial months of the year, the value of Ethereum is very low and then starts to increase in the middle months but is fluctuating very much in the middle.

To deduce further conclusions, let's have a look at the Delta graph.

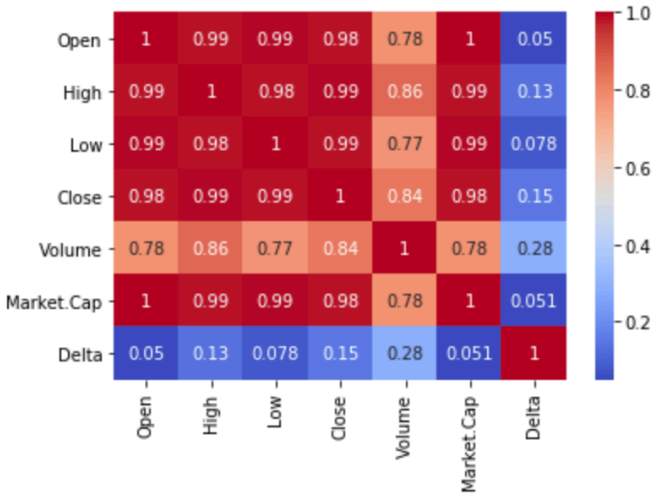


The adjacent graph shows that the price of Ethereum is flickering towards the 0.2 delta positive ones but there are a few negative ones as well.

Also, towards the end the value slightly drops ending the year on a pretty stable note.

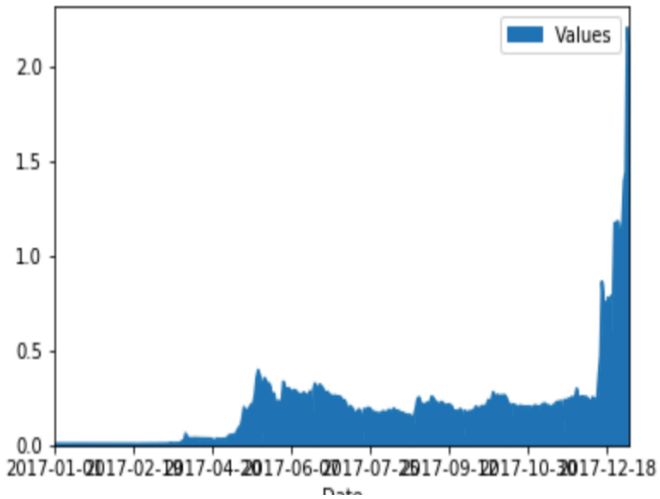
Ripple (XRP)

The Ripple correlation table is as follows:



Here, you can clearly see that 'Open' with 'Delta' has the lowest correlation value of 0.05

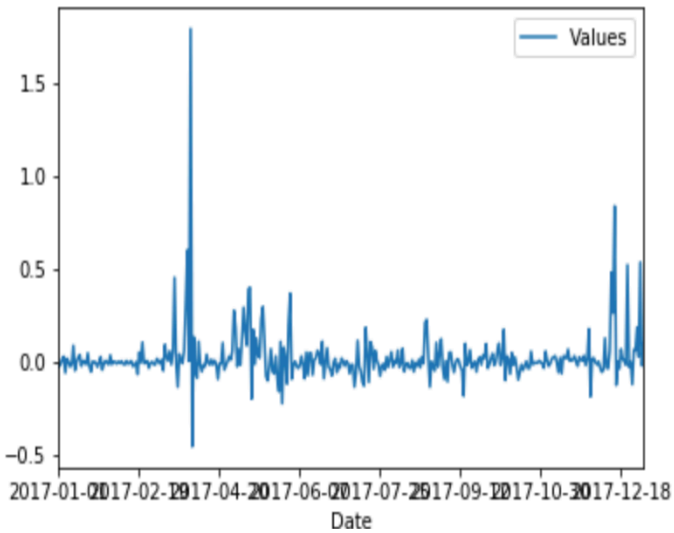
Let's consider the area graph of 'Open' attribute and make suitable analysis based on the graph.



Here, in the initial months of the year, the value of ripple is very low and then it starts to go up and become steady in the middle months.

Also, towards the end of the year, you can see that the value of ripple keeps on increasing steadily.

To deduce further conclusions, let's have a look at the Delta graph.



The Delta graph clearly indicates that the value of ripple is flickering on the positive side as the day ends which indicates that an investment could be a good idea in this case.

Also, in the initial months there is a sudden exponential increase in the value where the line reaches a very high value.

Draw Conclusions

I have applied a simple Linear Regression model on Bitcoin (BTC), Ethereum (ETH) and Ripple (XRP) to determine the daily closing price and also to plot scatter plots and distance plots of the results to make observations and draw conclusions.

Linear Model Bitcoin (BTC)

Here, the intercept values (test size = 0.3) for the columns are:

```
array([-1.18414439e-02,  1.10134285e+00,  3.31236371e-01, -4.28836094e-08,
       -2.49649181e-08])
```

The intercept values (test size = 0.1) for the columns are:

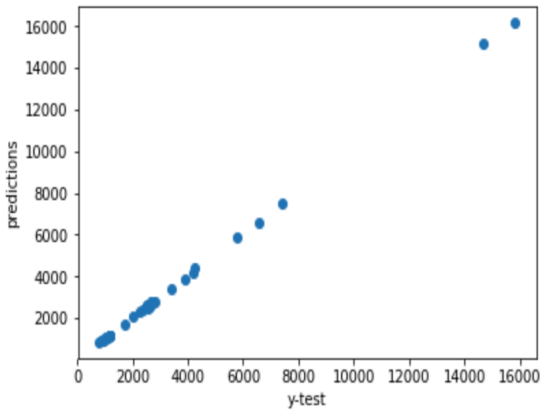
```
array([-1.10380263e-01,  1.08147240e+00,  3.61666052e-01, -3.93283690e-08,
       -1.96105382e-08])
```

The co-efficient data frame is as follows:

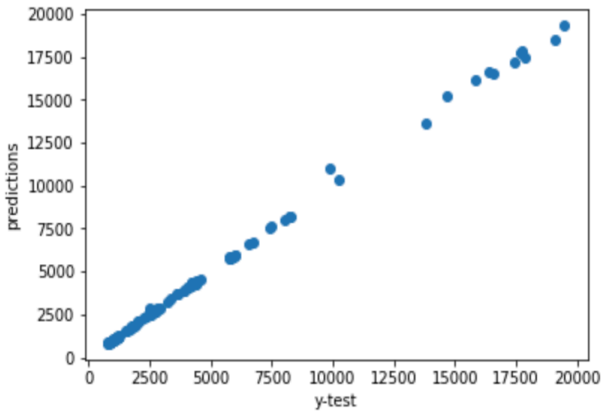
	Coeff
Open	-1.184144e-02
High	1.101343e+00
Low	3.312364e-01
Volume	-4.288361e-08
Market.Cap	-2.496492e-08

From the coefficient data frame, 'High' and 'Low' values are a major factor in determining daily closing price for Bitcoin.

To deduce further conclusions, let's have a look at the scatter plots for test sizes 0.3 and



Test size = 0.3

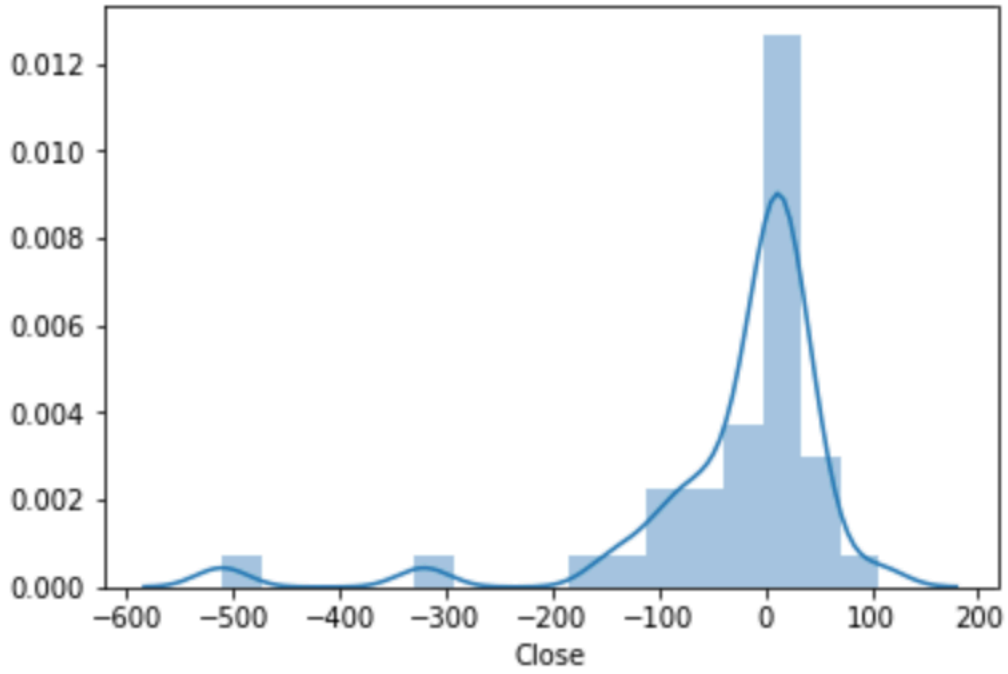


Test size = 0.1

From the above graphs, the scatter plot of test size 0.1 doesn't provide enough evidence as to how the Linear model fits the data.
However, when you observe the scatter plot of test size 0.3, linear model is able to tell us whether for a particular day, the price will go up or down except in a few cases.

Here, the linear model fits very well on test size 0.3. The distribution is pretty uniform except for a few cases which goes against the trend.

Now, let's look at the distance plot to make further conclusions.
The distance plot of 'y-test' – 'predictions' (Test size = 0.3) is as follows:



The distance plot looks pretty standard uniform distribution with a certain outliers on the negative size of the axes.

Here, since the price of Bitcoin (BTC) is high, this linear model is a good fit to determine the daily close price of BTC.

Linear Model for Ethereum (ETH)

Here, the intercept values (test size = 0.3) for the columns are:

```
array([-3.85224331e-01,  7.43186205e-01,  5.01283205e-01,  1.34095124e-09,
        1.40484853e-09])
```

The intercept values (test size = 0.1) for the columns are:

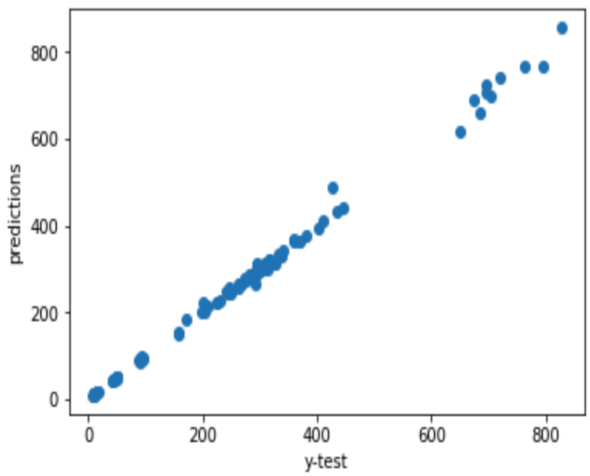
```
array([-4.37047007e-01,  7.76563645e-01,  5.34366037e-01,  1.28190429e-09,
        1.23680244e-09])
```

The co-efficient data frame is as follows:

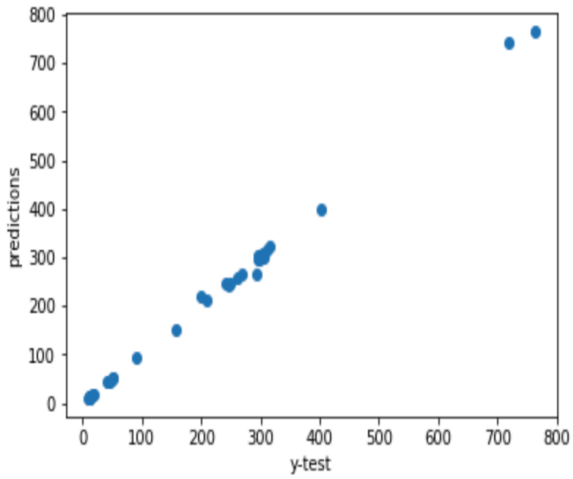
	Coeff
Open	-3.852243e-01
High	7.431862e-01
Low	5.012832e-01
Volume	1.340951e-09
Market.Cap	1.404849e-09

From the coefficient data frame, 'High' and 'Low', 'Volume' and 'Market.Cap' values are a major factor in determining daily closing price for Ethereum.

To deduce further conclusions, let's have a look at the scatter plots for test sizes 0.3 and 0.1



Test size = 0.3



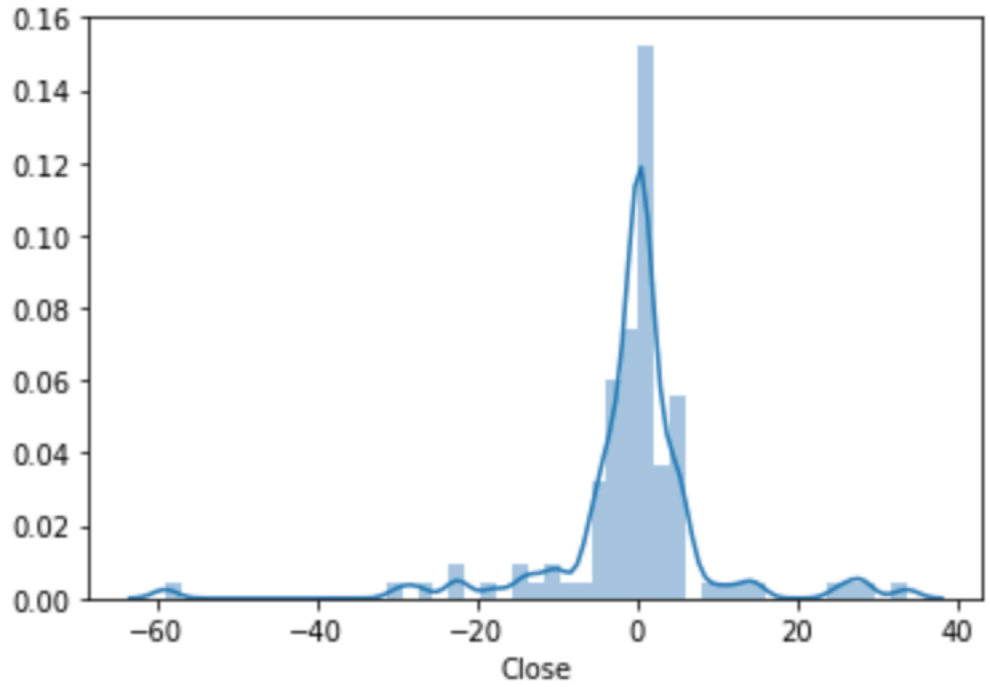
Test size = 0.1

From the above graph of scatter plot for test size 0.1, there is not enough data to draw accurate conclusion or make observations. Therefore, we will use scatter plot for test size 0.3 to make conclusions.

When you observe the scatter plot for test size 0.3, the model fits the data well with a uniform distribution with just few outcomes going against them.

If we had just used the linear model with test size 0.1, the observations would not have been accurate since there is not enough data.

Now, let's look at the distance plot to make further conclusions.
The distance plot of 'y-test' – 'predictions' (Test size = 0.3) is as follows:



Here, you can clearly observe that the distribution is uniform with just a few outliers on both sides of the axes.

Based on all the observations, we can successfully conclude that the Linear Regression model will have us in determining the daily closing price of Ethereum (ETH) with a few exceptions.

Linear Model for Ripple (XRP)

Here, the intercept values (test size = 0.3) for the columns are:

```
array([-3.71158709e+00,  1.25910924e+00,  5.13956256e-01, -6.87796853e-11,
        7.72056458e-11])
```

The intercept values (test size = 0.1) for the columns are:

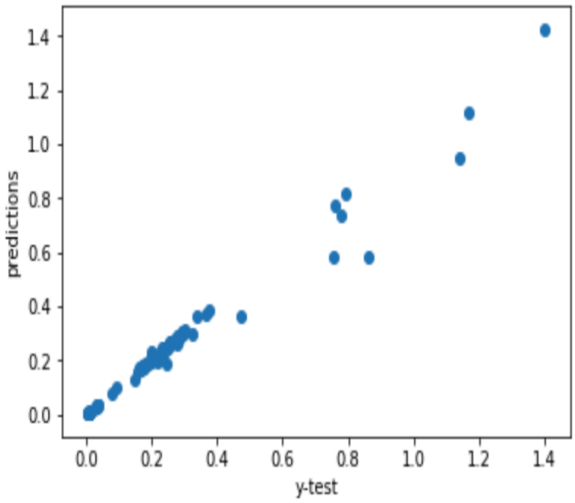
```
array([-3.03556084e-01,  8.72122158e-01,  4.37641617e-01, -1.11211925e-11,
       -6.74618908e-13])
```

The co-efficient data frame is as follows:

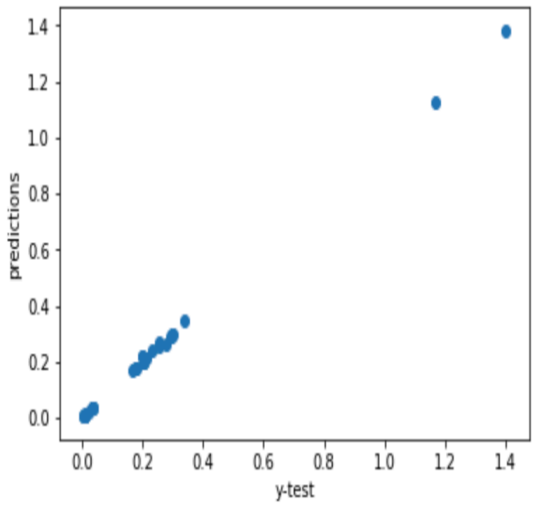
	Coeff
Open	-3.711587e+00
High	1.259109e+00
Low	5.139563e-01
Volume	-6.877969e-11
Market.Cap	7.720565e-11

From the coefficient data frame, 'High' and 'Low', and 'Market.Cap' values are a major factor in determining daily closing price for Ethereum.

To deduce further conclusions, let's have a look at the scatter plots for test sizes 0.3 and 0.1



Test size = 0.3



Test size = 0.1

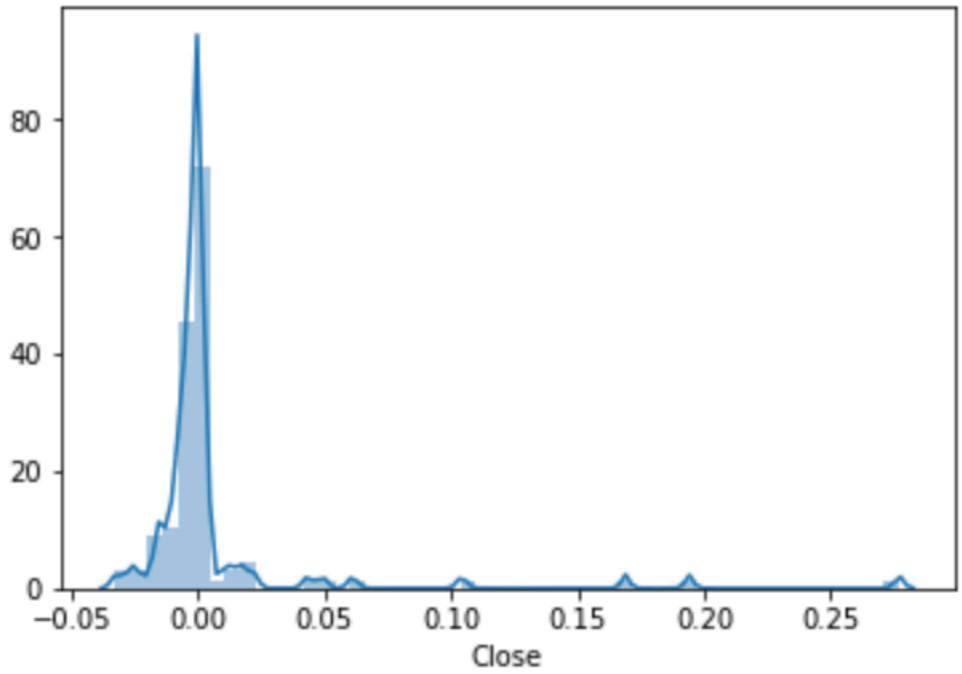
Here as well, from the above scatter plot of test size 0.1, you can clearly observe that there is not enough data present in the training data to make accurate observations and conclusions.

When you observe the scatter plot for test size 0.3, the model fits the data well and shows a linear trend and the distribution is uniform with few outliers.

Since the price of Ripple (XRP) is very low, we can't be sure whether the linear model can be used to predict its daily closing price.

Now, let's look at the distance plot to make further conclusions.

The distance plot of ‘y-test’ – ‘predictions’ (Test size = 0.3) is as follows:



The distance plot also shows a similar linear trend with the distribution being uniform and a few outliers on the positive side of the axes.

This successfully concludes that since the price is XRP is low, the linear model cannot be used to determine its daily closing price.

Conclusion

In this project, I have implemented the Exploratory Data Analysis (EDA) process to make accurate observations and conclusions on different trends of data.

Data Wrangling

How data was gathered is explained here. There were several issues with the structure of data which was determined in this step. After this, all those issues were resolved, and data was cleaned.

Data Exploration

Here, several area graphs and line plots were generated on different crypto currencies to determine patterns between different attributes and visualize relationships between them.

Draw Conclusions

Here, Linear Logistic Regression model was applied on top 3 cryptocurrencies to see their results and draw observations based on them.

References

1 http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/

2 https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm

3 https://www.sisense.com/blog/exploratory-data-analysis/

4 https://en.wikipedia.org/wiki/Exploratory_data_analysis