Facilitation of Cryptocurrency Price Prediction by Sentiment Analysis

Team

Name	Hawk-id	Email-id
Sai Yeshwanth (lead)	A20514173	Sbolikonda@hawk.iit.edu
Ashlesh Khajbage	A20517913	akhajbage@hawk.iit.edu
Chandana poloju	A20527866	cpoloju@hawk.iit.edu
Swastika Pandit	A20503284	spandit3@hawk.iit.edu
Sai Deepthi Pinninti	A20513400	Spinninti@hawk.iit.edu

Illinois Institute of Technology
CSP571-Data Preparation and Analysis
Professor: Jawahar Panchal

Table of Contents

Abstract

Cryptocurrency is a dynamic and rapidly changing market, making it difficult for investors to make informed decisions. Sentiment analysis has emerged as a promising tool for analyzing the emotional tone behind news articles, social media posts, and other online content related to cryptocurrency. In this project, we aim to explore the use of sentiment analysis to facilitate cryptocurrency trading decisions. To achieve this goal, we scraped data from various news articles and collected price data for several cryptocurrencies during a specific time period. Using exploratory data analysis, we identified which cryptocurrencies to focus on and selected a few for further analysis. We implemented a machine learning model to predict price changes based on sentiment extracted from the text of news articles. Additionally, we conducted a time series analysis to further understand the relationship between news sentiment and cryptocurrency price movements.

1. Overview

1.1 Objective

The objective of the project is to investigate how news articles impact the prices of cryptocurrencies. The project aims to identify any relationship between news sentiment and price movements. Additionally, the project will identify the most actively traded coins in the cryptocurrency market to understand their significance and correlation. Ultimately, the project aims to build reliable models that can provide insights into the role of news sentiment in shaping cryptocurrency prices.

1.2 What we seek to address:

- Does historical data of BTC, ETH, status, or ratings influence this result?
- How were cryptocurrencies trending during that period?
- What were the top 10 cryptocurrencies and their market capitalization at that time?
- Do the total traded Quantity and Close Price of the coin hold any sort of relation between them?
- Which analysis model will be best for the analysis of the dataset?
- Can we analyze price movements and trading volumes and give 7-day predictions?

1.3 Literature Survey

We came across two research papers that have proposed different models for this task. Wong et al. (2018) proposed a model that uses LSTM and GRU networks to predict cryptocurrency prices based on sentiment analysis of news articles and social media posts [1]. Egging and Gregoriou (2015) explored the relationship between sentiment analysis of Twitter data and Bitcoin prices, using a simple regression model to predict Bitcoin prices based on Twitter sentiment [2]. These papers demonstrate the potential of sentiment analysis in predicting cryptocurrency prices and offer insights into the various approaches that can be used to achieve this goal.

1.4 Project Methodology

Step	Description
Web scraping	Collect news articles related to cryptocurrency from various sources, including Forbes, CNBC, Coin Telegraph, CoinDesk, and Bitcoin.
Data cleaning	Clean and preprocess the text data of the collected news articles to ensure that they are suitable for sentiment analysis.
Sentiment analysis	Analyze the sentiment of the news articles using a Custom sentiment analysis library to extract valuable information like sentiment scores.
Adjustments	Conduct exploratory data analysis on another related dataset (Coins) to identify the top coins and focus on them in our main analysis.
Exploratory data analysis	Visualize and analyze the collected data to identify patterns and relationships between sentiment and price movements.
Model selection	Choose appropriate machine learning models such as Lasso, Random Forest, and Multiple Linear Regression, and time series models such as ARIMA, Naive Method, and Exponential Smoothing.
Model building	Train and test the selected models on the collected data to determine their performance in predicting future price movements.
Model performance evaluation	Evaluate the performance of the built models using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).
Time series analysis	Conduct time series analysis to further understand the relationship between sentiment and price movements.
Conclusion	Draw conclusions based on the results obtained from the analysis and move to the recommendations and enhancements to our project.

2. Data Acquisition

For this project, we utilized two primary categories of data. The first type consisted of Investing.com's historical price data for 10 cryptocurrencies, which included Binance Coin, Bitcoin, Cardano, Dogecoin, Ethereum, Polkadot, Solana, Tether, USD Coin, and XRP. Extracting this data was a straightforward process, as the website provides a simple function to download the desired data in CSV format. The CSV data columns included Name, symbol, date, price, open, high, low, volume, market cap, and percentage of daily change.

The second type of data utilized in this project was 360 news articles from five different crypto news sites, including Bitcoin, CNBC, CoinDesk, Cointelegraph, and Forbes. These articles were scraped between May 16, 2017, and June 2018, and were used to analyze the sentiment of the news coverage on the 10 selected cryptocurrencies.

By analyzing the collected data, we aimed to gain insights into the relationships between the sentiment of news articles and the price changes of cryptocurrencies over time. This research has the potential to provide valuable insights into the factors that influence cryptocurrency prices and may be useful for investors and traders in making informed decisions.

3. Text Data Processing

3.1 Data cleaning

Firstly, we are segregating the data from CSV into data frames and encoding this to “UTF-8” and removing noises, such as HTML tags, special characters, and texts which are unnecessary for the sentiment analysis.

````
```{r}

csvForbes <- "Ripple_Forbes.csv"
csvCNBC <- "Ripple_CNBC.csv"
csvCointelegraph <- "Ripple_Cointelegraph.csv"
csvCoindesk <- "Ripple_Coindesk.csv"
csvBitcoin <- "Ripple_Bitcoin.csv"

#################### Make Data Frames ##########################
# FORBES
dfForbes <- read.csv(csvForbes, encoding = "UTF-8", header = TRUE, stringsAsFactors = FALSE)
# Edit Forbes Text and remove unwanted sentence artifacts
dfForbes$Text <- gsub("\\[Ed note: Investing in cryptocoins or tokens is highly speculative and the market is largely unregulated. Anyone considering it should be prepared to lose their entire investment.\\]", " ", dfForbes$Text)
dfForbes$Text <- gsub("\\[Ed. note: Investing in cryptocoins or tokens is highly speculative and the market is largely unregulated. Anyone considering it should be prepared to lose their entire investment.\\]", " ", dfForbes$Text)
dfForbes$Text <- gsub("\\[Ed note: Investing in cryptocoins or tokens is highly speculative and the market is largely unregulated. Anyone considering it should be prepared to lose their entire investment.\\]", " ", dfForbes$Text)
dfForbes$Text <- strsplit(dfForbes$Text, "Editor's Note & Disclosure:")[[1]]
dfForbes$Text <- strsplit(dfForbes$Text, "Editor's Note and Disclosure:")[[1]]
dfForbes

```
dfForbes

```
````

Description: df [49 × 6]

Date <chr>	Coin <chr>	website <chr>	Headline <chr>
2018-02-09 12:03:00	Ripple	forbes	Crypto Watch: Ripple (XRP) Price Surges 21% in 10 Hours
2018-02-07 04:45:00	Ripple	forbes	Mellon Banking Heir's New Crypto Fortune: Almost $1B In Ripple's XRP
2018-02-07 04:45:00	Ripple	forbes	Ripple CEO Brad Garlinghouse Has A Nine-Figure Fortune From XRP
2018-01-02 08:00:00	Ripple	forbes	Meet The Crypto Billionaires Getting Rich From Ripple's XRP
2018-01-14 11:46:00	Ripple	forbes	3 Reasons To Not Get Excited About Seagate's Investment In Ripple/XRP
2018-01-26 13:01:00	Ripple	forbes	Ripple Drops More Than 30% In A Week As Hype Fades
2017-12-20 10:27:00	Ripple	forbes	The Bear Case For XRP: Bitcoin Futures Edition
2018-01-16 12:13:00	Ripple	forbes	Ripple Has Dropped More Than 65% From Record High
2017-12-21 13:28:00	Ripple	forbes	Crypto Watch: Another Big Day For Ripple -- XRP Price Jumps 64% In 17 Hours
2018-01-10 11:05:00	Ripple	forbes	Ripple Falls Below $1.75, Down 55% From High

1–10 of 49 rows | 1–4 of 6 columns   Previous 1 2 3 4 5 Next

We performed similar activity with five dataframes (Forbes, CNBC, Cointelegraph, Coindesk and bitcoin) which we collected from the news articles.

````
```{r}
# COINDESK
dfCoindesk <- read.csv(csvCoindesk, encoding = 'UTF-8', header = TRUE, stringsAsFactors = FALSE)
# Edit Coindesk Text and remove unwanted sentence artifacts
dfCoindesk$Text <- gsub('Disclosure:', ' ', dfCoindesk$Text)
dfCoindesk$Text <- gsub('Disclaimer:', ' ', dfCoindesk$Text)

# BITCOIN
dfBitcoin <- read.csv(csvBitcoin, encoding = 'UTF-8', header = TRUE, stringsAsFactors = FALSE)
```

```{r}
site <- list(dfForbes, dfCNBC, dfCointelegraph, dfCoindesk, dfBitcoin)
site
```
````

R Console | data.frame 49 x 6 | data.frame 34 x 6 | data.frame 121 x 6 | data.frame 64 x 6 | data.frame 21 x 6

Description: df [21 × 6]

We then bound all the data frames into one data frame for sentiment analysis.

````
```{r}
df <- do.call(rbind, site)
df
```
````

Description: df [289 × 6]

Date <chr>	Coin <chr>	website <chr>
2018-02-09 12:03:00	Ripple	forbes
2018-02-07 04:45:00	Ripple	forbes
2018-02-07 04:45:00	Ripple	forbes
2018-01-02 08:00:00	Ripple	forbes
2018-01-14 11:46:00	Ripple	forbes
2018-01-26 13:01:00	Ripple	forbes
2017-12-20 10:27:00	Ripple	forbes
2018-01-16 12:13:00	Ripple	forbes
2017-12-21 13:28:00	Ripple	forbes
2018-01-10 11:05:00	Ripple	forbes

1–10 of 289 rows | 1–3 of 6 columns   Previous 1 2 3 4 5 6 ... 29 Next
````

3.2 Sentiment Analysis

To perform sentiment analysis on dataframe, we used Python code to compute sentiment scores based on positive and negative word libraries mentioned in Appendix A & B. As most of the appropriate libraries were supported by Python very accurately but not by R. The result was as shown below:

```python
df['hsentiment'] = df.Headline.apply(lambda x: coin_sentiment(x, True))
df[['Headline Positive','Headline Negative']] = df.hsentiment.apply(pd.Series)
df = df.drop('hsentiment', axis =1 )
df
```

Python

	Date	Coin	website	Headline	Text	Link	Positive Sentiment	Negative Sentiment	Headline Positive	Headline Negative
0	2018-02-09 12:03:00	Ripple	forbes	Crypto Watch: Ripple (XRP) Price Surges 21% in...	GERMANY, BONN - JANUARY 31: Symbol photo on th...	https://www.forbes.com/sites/jessedamiani/2018...	0.285714	-0.023810	1	0
1	2018-02-07 04:45:00	Ripple	forbes	Mellon Banking Heir's New Crypto Fortune: Almo...	\n\tThis story appears in the February 28, 201...	https://www.forbes.com/sites/nathanvardi/2018/...	0.300000	-0.005556	2	0
2	2018-02-07 04:45:00	Ripple	forbes	Ripple CEO Brad Garlinghouse Has A Nine-Figure...	\n\tThis story appears in the February 28, 201...	https://www.forbes.com/sites/laurashin/2018/02...	0.222222	0.000000	1	0
3	2018-01-02 08:00:00	Ripple	forbes	Meet The Crypto Billionaires Getting Rich From...	The value of cryptocurrencies skyrocketed in 2...	https://www.forbes.com/sites/laurashin/2018/01...	0.540797	-0.011385	1	0
4	2018-01-14 11:46:00	Ripple	forbes	3 Reasons To Not Get Excited About Seagate's I...	In May 2015, Seagate made an investment in Rip...	https://www.forbes.com/sites/chuckjones/2018/0...	0.256410	-0.019231	0	0
5	2018-01-26 13:01:00	Ripple	forbes	Ripple Drops More Than 30% In A Week As Hype F...	Sentiment caused XRP to decline 30% in one wee...	https://www.forbes.com/sites/cbovaird/2018/01/...	0.256579	-0.052632	0	-2
6	2017-12-20 10:27:00	Ripple	forbes	The Bear Case For XRP: Bitcoin Futures Edition	Miguel Vias, head of XRP markets at Ripple Lab...	https://www.forbes.com/sites/ksamani/2017/12/2...	0.241525	-0.029661	0	-1
7	2018-01-			Ripple Has Dropped						

Figure: - Sentiment Score results

3.3 Sentiment Aggregation

After calculating sentiment scores, all the text that had been scraped was analyzed to determine how positive or negative it was. The headlines and texts of the stories were both subjected to analysis. But initially, simply the text sentiment was looked at. The headline sentiment was held back in case further information was required to make judgments.

Date <date>	Positive Sentiment <dbl>	Negative Sentiment <dbl>
2013-12-19	0.140625000	-0.020833333
2017-05-16	0.187855787	-0.034155598
2017-05-26	0.461538462	-0.004524887
2017-06-07	0.181159420	-0.007246377
2017-06-21	0.175000000	-0.056250000
2017-07-20	0.250000000	-0.083333333
2017-07-21	0.615384615	0.000000000
2017-08-02	0.176870748	-0.072562358
2017-08-04	0.163615561	-0.102974828
2017-08-07	0.068870523	-0.099173554

Figure: - Quantifying Articles' Sentiment

3.3 Issues Encountered

How to rank dates that had multiple sentiment scores was the first problem that needed to be solved with this data. Multiple scores associated with a single date indicated that various articles were produced and processed on that day. For instance, on Jan 3rd, four different articles were written.

```
```{r}
# Sort the dataframe by row names (assuming it's an index)
df_sorted <- df[order(rownames(df)), ]

# Subset the dataframe to rows 26 to 30 (inclusive)
df_subset <- df_sorted[26:30, ]
df_subset
```
```

	Date <date>	Positive Sentiment <dbl>	Negative Sentiment <dbl>
121	2018-01-02	0.2302772	-0.115138593
122	2018-01-03	0.6818182	0.000000000
123	2018-01-03	0.1206897	-0.057471264
124	2018-01-03	0.4473684	-0.011695906
125	2018-01-03	0.5208333	-0.003472222

Figure: - Multiple Sentiment Scores for a Single Day

We were able to handle this in a few different ways:

- Firstly, aggregates of sentiment averaged, sentiment root mean square (RMS), and sentiment summed were calculated to later analyze the best metric.
- Secondly, “overall" sentiment had to be established together with the intrinsic values of positive and negative sentiment for each article. This "overall" had to accurately reflect the size of both intrinsic ratings. The "overall sentiment" was calculated using the same procedures as were used to aggregate the various articles. Each article's Positive and Negative Scores were averaged, RMS-summarized, and added together to provide an "overall sentiment" for that item.

The resulting time series data frame was refined to 155 rows and 10 columns.

	Date	Positive Sentiment	Negative Sentiment	Positive Sentiment Averaged	Positive Sentiment Summed	Positive Sentiment RMS	Negative Sentiment Averaged	Negative Sentiment Summed	Negative Sentiment RMS	Overall Sentiment	Overall Sentiment Averaged	Overall Sentiment Summed	Overall Sentiment RMS
1	2013-12-19	0.140625000	-0.020833333	0.14062500	0.14062500	0.14062500	-0.020833333	-0.020833333	0.020833333	0.119791667	0.119791667	0.119791667	0.119791667
2	2017-05-16	0.187855787	-0.034155598	0.18785579	0.18785579	0.18785579	-0.034155598	-0.034155598	0.034155598	0.153700190	0.153700190	0.153700190	0.153700190
3	2017-05-26	0.461538462	-0.004524887	0.46153846	0.46153846	0.46153846	-0.004524887	-0.004524887	0.004524887	0.457013575	0.457013575	0.457013575	0.457013575
4	2017-06-07	0.181159420	-0.007246377	0.18115942	0.18115942	0.18115942	-0.007246377	-0.007246377	0.007246377	0.173913043	0.173913043	0.173913043	0.173913043
5	2017-06-21	0.175000000	-0.056250000	0.17500000	0.17500000	0.17500000	-0.056250000	-0.056250000	0.056250000	0.118750000	0.118750000	0.118750000	0.118750000
6	2017-07-20	0.250000000	-0.083333333	0.25000000	0.25000000	0.25000000	-0.083333333	-0.083333333	0.083333333	0.166666667	0.166666667	0.166666667	0.166666667
7	2017-07-21	0.615384615	0.000000000	0.61538462	0.61538462	0.61538462	0.000000000	0.000000000	0.000000000	0.615384615	0.615384615	0.615384615	0.615384615
8	2017-08-02	0.176870748	-0.072562358	0.17687075	0.17687075	0.17687075	-0.072562358	-0.072562358	0.072562358	0.104308390	0.104308390	0.104308390	0.104308390
9	2017-08-04	0.163615561	-0.102974828	0.16361556	0.16361556	0.16361556	-0.102974828	-0.102974828	0.102974828	0.060640732	0.060640732	0.060640732	0.060640732
10	2017-08-07	0.068870523	-0.099173554	0.06887052	0.06887052	0.06887052	-0.099173554	-0.099173554	0.099173554	-0.030303030	-0.030303030	-0.030303030	0.030303030
11	2017-08-09	0.081081081	-0.081081081	0.08108108	0.08108108	0.08108108	-0.081081081	-0.081081081	0.081081081	0.000000000	0.000000000	0.000000000	0.000000000
12	2017-08-11	0.094086022	-0.067204301	0.09408602	0.09408602	0.09408602	-0.067204301	-0.067204301	0.067204301	0.026881720	0.026881720	0.026881720	0.026881720
13	2017-08-13	0.066315789	-0.185263158	0.06631579	0.06631579	0.06631579	-0.185263158	-0.185263158	0.185263158	-0.118947368	-0.118947368	-0.118947368	0.118947368
14	2017-08-15	0.700000000	0.000000000	0.70000000	0.70000000	0.70000000	0.000000000	0.000000000	0.000000000	0.700000000	0.700000000	0.700000000	0.700000000
15	2017-08-16	0.111111111	-0.044444444	0.11111111	0.11111111	0.11111111	-0.044444444	-0.044444444	0.044444444	0.066666667	0.066666667	0.066666667	0.066666667
16	2017-08-17	0.357142857	0.000000000	0.35714286	0.35714286	0.35714286	0.000000000	0.000000000	0.000000000	0.357142857	0.357142857	0.357142857	0.357142857
17	2017-08-18	0.111111111	-0.188888889	0.11111111	0.11111111	0.11111111	-0.188888889	-0.188888889	0.188888889	-0.077777778	-0.077777778	-0.077777778	0.077777778
18	2017-08-21	0.332330827	-0.006015038	0.33233083	0.33233083	0.33233083	-0.006015038	-0.006015038	0.006015038	0.326315789	0.326315789	0.326315789	0.326315789

Figure: - Aggregated Sentiment

4. Price Data Processing

```{r}
library(readxl)

# Read the Excel file into a data frame
crypto_df <- read_excel("/Users/ashleshkhajbage/Documents/CRYPTO Project/crypto_data.xlsx")


# Convert the date column to a proper date format
crypto_df$Date <- as.Date(crypto_df$Date, format = "%d/%m/%Y")

head(crypto_df)
```

A tibble: 6 × 9

Name <chr>	Symbol <chr>	Date <date>	High <dbl>	Low <dbl>	Open <dbl>	Close <dbl>	Volume <dbl>	Marketcap <dbl>
Binance Coin	BNB	2017-01-01	38.92818	37.04631	37.37457	37.90501	459165743	5473732252
Binance Coin	BNB	2017-01-02	38.83625	36.92560	37.91711	38.24159	521965394	5522336922
Binance Coin	BNB	2017-01-03	41.60632	37.81810	38.25373	41.14898	758008613	5942182741
Binance Coin	BNB	2017-01-04	43.13212	38.14398	41.19828	40.92635	807877171	5910034134
Binance Coin	BNB	2017-01-05	41.73460	38.97895	40.93728	41.73460	644270927	6026750037
Binance Coin	BNB	2017-01-06	42.19270	40.59253	41.77554	42.16595	641021601	6089040529

6 rows

4.1 Assumptions / Adjustments:

As we have 10 coins and we need to know the top 3 coins to perform the analysis. We collected the dataset from the website Investing.com and analyzed the correlation of the coins by performing univariate and bivariate analyses. This will determine the 3 coins on which we have to analyze further and build models. After getting the results from this section, we are moving forward with Exploratory data analysis on our main dataset.

4.2 Data Cleaning & Preparation:

- **Check for missing values** - No missing values present in a dataframe

```{r}
colSums(is.na(crypto_df))

#no missing values present
```

```
     Name    Symbol      Date      High       Low      Open     Close    Volume Marketcap 
        0         0         0         0         0         0         0         0         0 
```

- **Check for duplicates** - no duplicates.

```{r}
# Count duplicate rows in the crypto_df data frame
sum(duplicated(crypto_df))

#no duplicate values is present in dataset
```

```
[1] 0
```

- **Data Wrangling** - simple data manipulation tasks to convert the market capitalization and volume values to more interpretable units.

```{r}
# load the dplyr package
library(dplyr)

# create the new columns
crypto_df$market_billion_usd <- crypto_df$Marketcap / 1000000000
crypto_df$volume_billion_usd <- crypto_df$Volume / 1000000000

# remove the original columns
crypto_df <- select(crypto_df, -Volume, -Marketcap)

# print the updated dataframe
head(crypto_df)

```

5. Exploratory Data Analysis

5.1 Summary Statistics

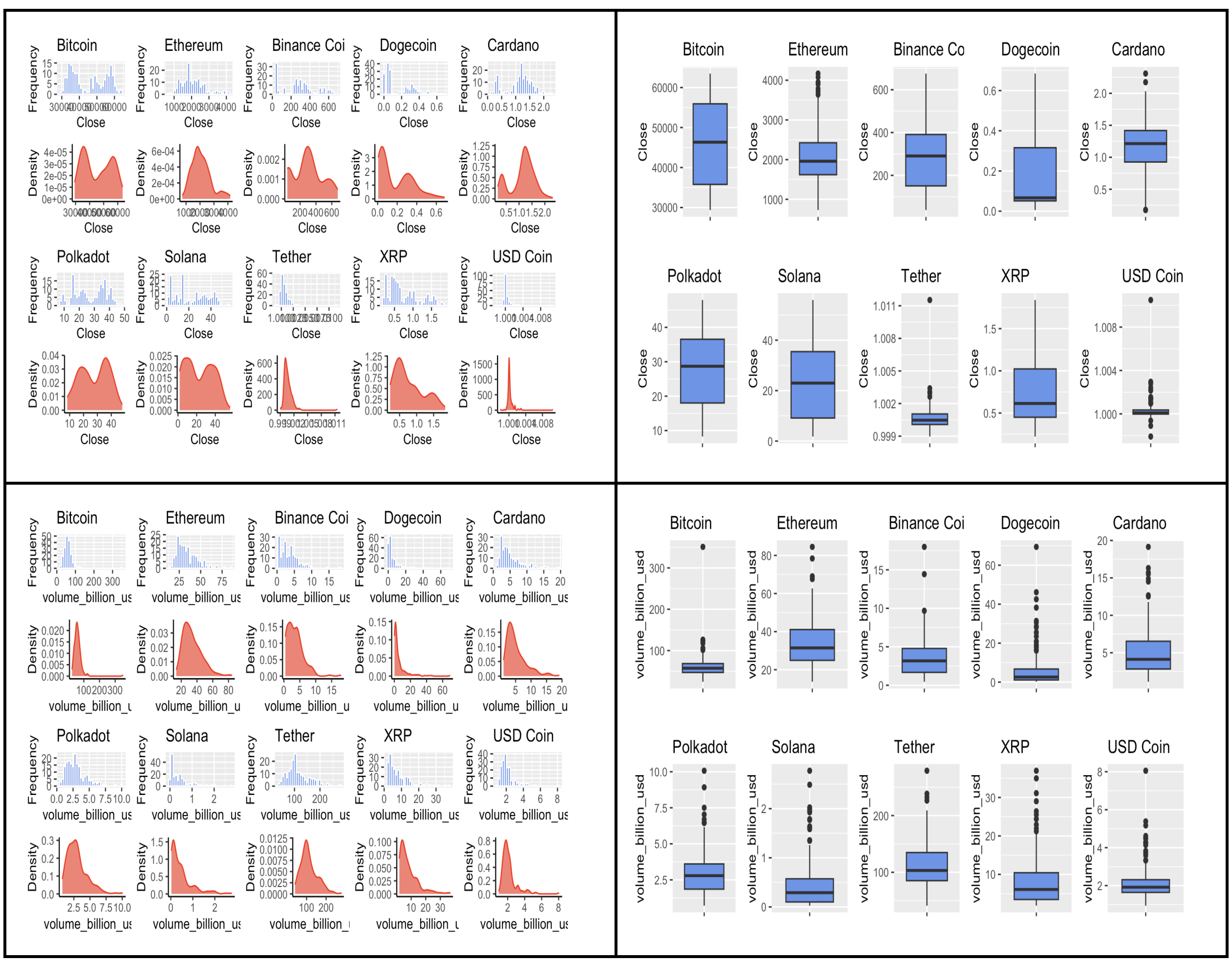
```
```{r}
summary(crypto_df)
```
```

```
     Name               Symbol              Date                 High                Low                 Open
 Length:1870        Length:1870        Min.   :2017-01-01   Min.   :    0.01   Min.   :    0.00   Min.   :    0.00
 Class :character   Class :character   1st Qu.:2018-01-21   1st Qu.:    1.00   1st Qu.:    1.00   1st Qu.:    1.00
 Mode  :character   Mode  :character   Median :2018-05-12   Median :    2.26   Median :    1.89   Median :    2.01
                                       Mean   :2018-04-17   Mean   : 4956.25   Mean   : 4598.01   Mean   : 4793.42
                                       3rd Qu.:2018-09-05   3rd Qu.:  307.66   3rd Qu.:  270.08   3rd Qu.:  289.98
                                       Max.   :2018-12-31   Max.   :64863.10   Max.   :62208.96   Max.   :63523.75
     Close           market_billion_usd  volume_billion_usd
 Min.   :    0.01   Min.   :   0.0838   Min.   :  0.0257
 1st Qu.:    1.00   1st Qu.:  11.4622   1st Qu.:  1.9228
 Median :    2.02   Median :  35.0107   Median :  4.3746
 Mean   : 4796.94   Mean   : 131.7729   Mean   : 23.5859
 3rd Qu.:  290.37   3rd Qu.:  62.5336   3rd Qu.: 31.4383
 Max.   :63503.46   Max.   :1186.3640   Max.   :350.9679
```

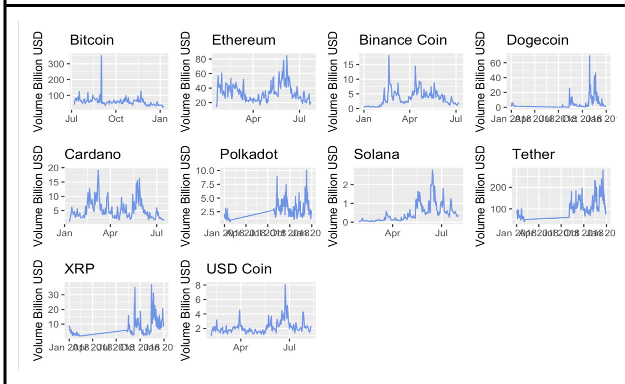
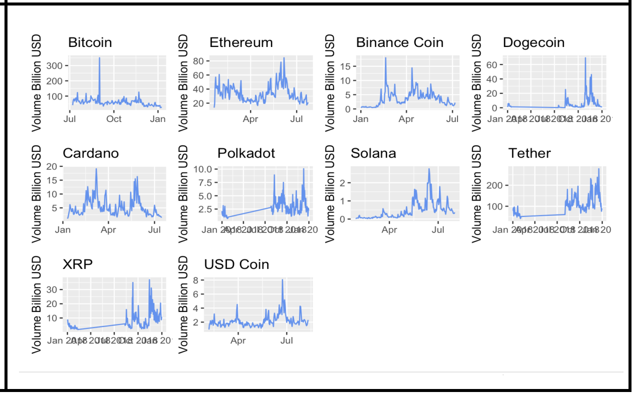
Now we will perform Univariate analysis and Bivariate analysis.

- In Univariate analysis, we examined the distribution and summary statistics of individual variables, while the Bivariate analysis we examined the relationship between two variables.
- These analyses helped us in providing insights into the nature of the data and helped to identify patterns or trends that may be of interest for further investigation.

5.2 Univariate Analysis



5.3 Bivariate Analysis

#Close Vs Date	#Volume Vs Date
	

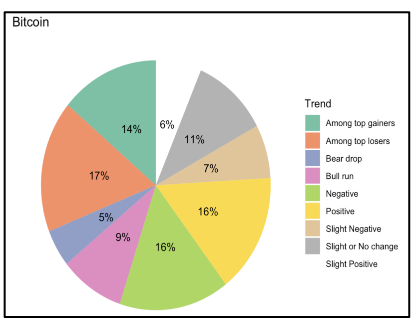
Inferences:

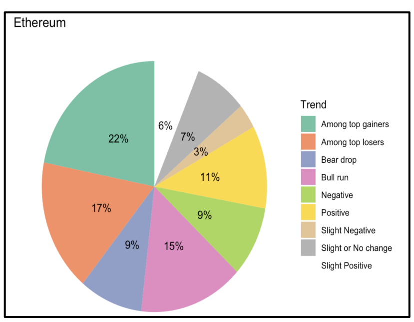
- While high, low, price, volume, and market values can also provide useful information for analyzing financial data, the closing price is typically the most used parameter for calculating daily returns and performing trend analysis.
- Above are the **distribution plot and summary box plot** for the **close price** of all currency

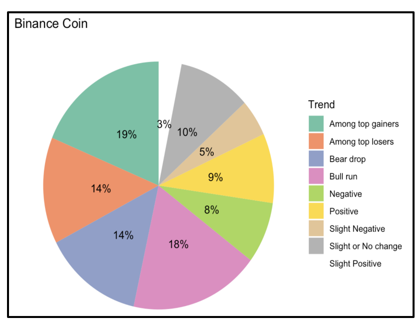
5.4 Trend Analysis

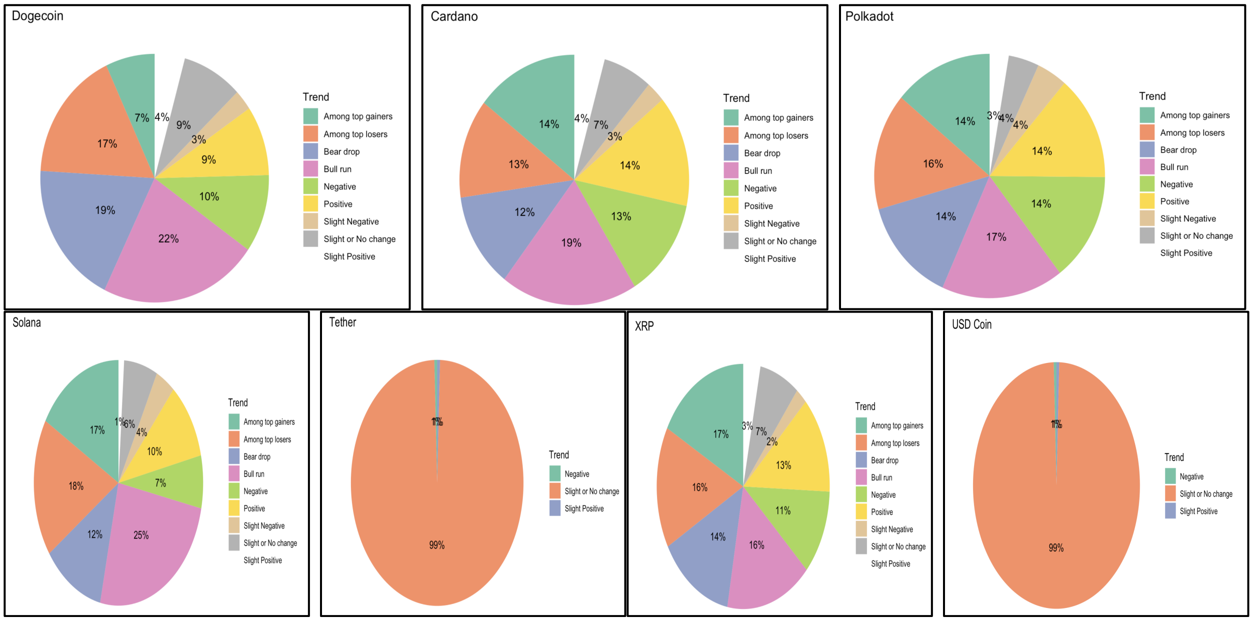
For trend Analysis, we added a new column 'Trend' whose values are based on the day-to-day percentage change which we calculated from the **closed price**.

- Trends like – "Slight or No change", "Slight Positive", "Slight Negative", "Positive", "Negative", "Among top gainers", "Among top losers", "Bull run", "Bear drop" were determined.
- We visualized it as a pie chart, with each sector representing the percentage of days each trend occurred.
- For this, we used the group by() function with the trend column to aggregate all days with the same trend into a single group before plotting the pie chart.





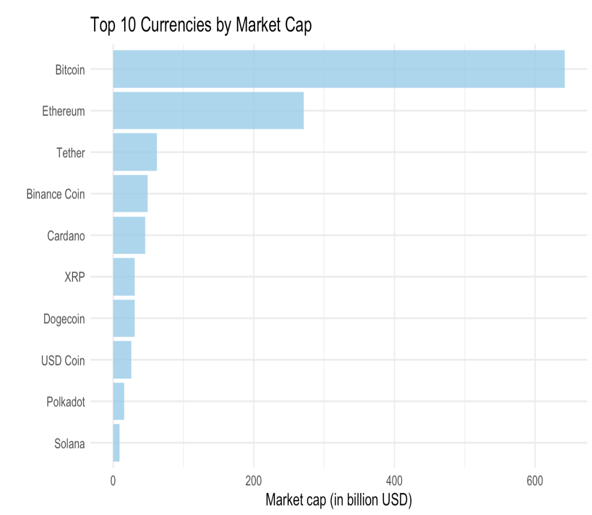




Observations:

From the trends, we can evidently assume that XRP would follow closely in the trend of either BTC or ETH. As their trend percentage are closely related. We can further analyze this from correlation and time series analysis.

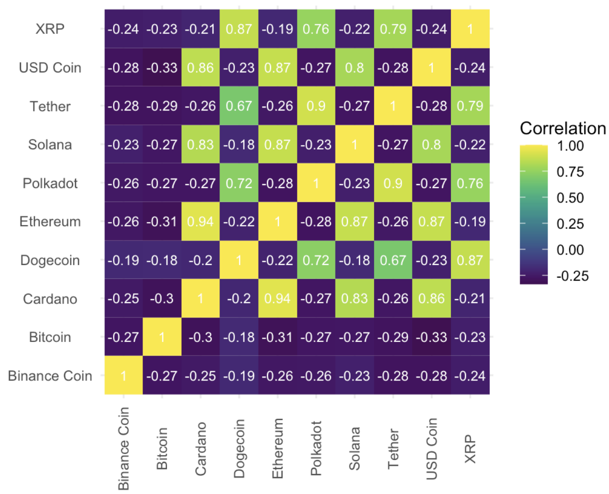
5.5 What were the top 10 cryptocurrencies and their market capitalization at that time?



- Market capitalization is a measure of the total value of a cryptocurrency, calculated by multiplying the current price of the cryptocurrency by the total number of coins in circulation.
- From the plot, we can see that Bitcoin has the largest market capitalization, followed by Ethereum and Tether. Binance Coin, Cardano, XRP.
- Dogecoin also have market capitalizations of over 30 billion USD, while USD Coin, Polkadot, and Solana have smaller market capitalizations.

5.6 Correlation Analysis among Coins

To verify the observation made in Trend analysis, we plotted a heatmap of the correlation matrix between the different cryptocurrency prices over time. Each cell in the heatmap represents the correlation coefficient between two cryptocurrency prices. By looking at the plot, we made following observations:



Observations:

- Strong correlation between Bitcoin and other cryptocurrencies (Ethereum, USD Coin, and Tether)
- This suggests that changes in the price of Bitcoin are likely to have a similar effect on the prices of these cryptocurrencies.
- Weak/negative correlations between some cryptocurrencies. (Eg. Binance coin and XRP)

6. More Exploratory Data Analysis

We have done EDA on this many cryptocurrencies, however, to understand the correlation between coins, we selected XRP, Bitcoin, and Ethereum. In addition to the above. These coins were chosen due to their widespread usage in purchasing other cryptocurrencies and because the market generally follows the trend of Bitcoin.

- We subset out these coin's data
- A multi-index data frame was used to display the price information for all 3 coins.
- Renamed column for understanding to join it with their sentiment.

~~~~
```{r}
str(df3)
```

 'data.frame':   255 obs. of  28 variables:
  $ Date                      : Date, format: "2017-08-01" "2017-08-02" "2017-08-03" "2017-08-04" ...
  $ Positive Sentiment Averaged: num  0 0.177 0 0.164 0 ...
  $ Positive Sentiment Summed  : num  0 0.177 0 0.164 0 ...
  $ Positive Sentiment RMS     : num  0 0.177 0 0.164 0 ...
  $ Negative Sentiment Averaged: num  0 -0.0726 0 -0.103 0 ...
  $ Negative Sentiment Summed  : num  0 -0.0726 0 -0.103 0 ...
  $ Negative Sentiment RMS     : num  0 0.0726 0 0.103 0 ...
  $ Overall Sentiment Averaged : num  0 0.1043 0 0.0606 0 ...
  $ Overall Sentiment Summed   : num  0 0.1043 0 0.0606 0 ...
  $ Overall Sentiment RMS      : num  0 0.1043 0 0.0606 0 ...
  $ XRP_Change                 : num  -4.59 2.46 -0.57 5.99 -2.01 -0.72 9.44 -6.38 -1.42 -1.83 ...
  $ XRP_High                   : num  0.185 0.18 0.177 0.178 0.192 ...
  $ XRP_Low                    : num  0.159 0.168 0.169 0.171 0.171 ...
  $ XRP_Open                   : num  0.163 0.179 0.17 0.175 0.174 ...
  $ XRP_Price                  : num  0.17 0.175 0.174 0.184 0.18 ...
  $ XRP_Vol                    : num  26.82 11.47 4.59 7.21 20.38 ...
  $ BTC_Change                 : num  -4.37 -1.07 3.27 2.5 13.86 ...
  $ BTC_High                   : num  2902 2758 2813 2875 3332 ...
  $ BTC_Low                    : num  2616 2640 2698 2763 2855 ...
  $ BTC_Open                   : num  2854 2734 2702 2790 2860 ...
  $ BTC_Price                  : num  2731 2702 2790 2860 3256 ...
  $ BTC_Vol                    : num  42.3 23.8 16.9 19 50.6 ...
  $ ETH_Change                 : num  12.53 -3.53 3.11 -1.81 15.01 ...
  $ ETH_High                   : num  233 230 228 228 260 ...
  $ ETH_Low                    : num  201 215 217 219 219 ...
  $ ETH_Open                   : num  201 226 218 225 221 ...
  $ ETH_Price                  : num  226 218 225 221 254 ...
  $ ETH_Vol                    : num  352.3 159.5 96.2 83.5 234.9 ...
~~~~

6.1 XRP Correlation Analysis

The price of XRP was compared to all other variables (the data frame's columns) because XRP was the dependent variable of interest. Since the entire market appears to follow Bitcoin, it was once thought that Bitcoin (BTC) would have the highest correlation with the price of XRP. The outcomes of such analysis are highlighted in the correlation matrix below:

	XRP_Change	XRP_Price
XRP_Change	1.00000000	0.9415002
XRP_High	0.94210546	0.9999982
XRP_Low	0.94125770	0.9999997
XRP_Open	0.94150142	1.0000000
XRP_Price	0.94150016	1.0000000
XRP_Vol	0.97928662	0.8677764
BTC_Change	0.97204900	0.9930998
BTC_High	-0.11954480	-0.4200027
BTC_Low	-0.10890065	-0.4117214
BTC_Open	-0.11746691	-0.4184246
BTC_Price	-0.11820222	-0.4188621
BTC_Vol	0.93325842	0.7718550
ETH_Change	0.97297449	0.9923168
ETH_High	0.07716504	-0.2334398
ETH_Low	0.10329802	-0.2099417
ETH_Open	0.09232334	-0.2212869
ETH_Price	0.09132976	-0.2222087

Figure: - XRP Price Correlation Matrix.

Visualizing the Bitcoin-XRP & Ethereum-XRP pair plots further explained the correlation results.

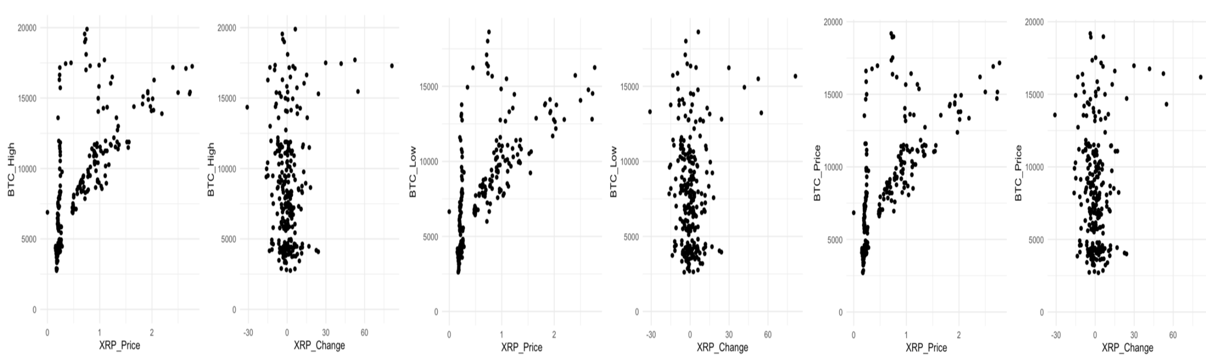


Figure: - Bitcoin vs XRP Price correlation pair plots

The positive link is evident when plotting Bitcoin prices (High, Low, Close) vs. XRP prices. But it is now transparent why BTC failed to merit a correlation. There seems to have been a period where the value of BTC rose significantly while XRP stayed largely unchanged. The idea that the market closely tracks BTC goes against this.

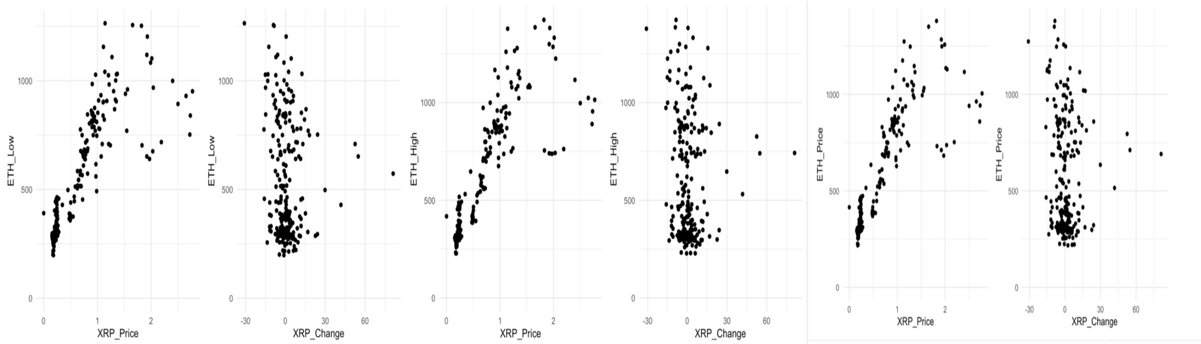


Figure: - Ethereum vs XRP Price correlation pair plots

Upon analyzing the pair plot of ETH and XRP, it was discovered that their movements were nearly identical in a 1:1 ratio. However, XRP experienced a sudden exponential surge in price only after it had reached a certain point, while ETH continued to follow a linear trend. This observation suggests that ETH is a more reliable indicator of XRP's price, thereby dismissing the idea that BTC is the dominant trend to monitor.

6.2 Time Series Analysis

Until this point, it is reasonable to expect that XRP's trends would closely follow those of BTC or ETH. This assumption was confirmed through the previous correlation analysis. The correlation characteristics of all three-time series were fully revealed when plotted together. Prior to plotting the prices, they were normalized and scaled to 1 since Bitcoin reached nearly $20,000 at its peak, while XRP only reached a low of $0.14 cents.

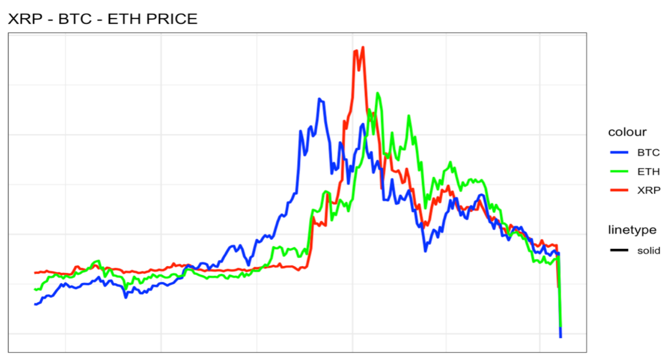


Figure: - XRP, BTC, ETH time series

Analysis & Observations:

- It appears that XRP tends to follow ETH's price fluctuations with **a delay of one or two days.**
- However, both XRP and ETH can trail behind BTC's price movements for up to a week or longer. This explains why there is a **flat line correlation trend between XRP and BTC in the pair plot.**
- Even though the trading volume of XRP was much higher than BTC and ETH combined, with XRP trading in the millions and BTC and ETH trading in the hundreds of thousands,

the price of XRP remained relatively stable while BTC experienced a significant price surge.

- This is intriguing since not only does the market typically follow BTC, but XRP's trading volume was significantly higher. It is possible that this is due to differences in price points, as the average price of BTC was $8300, ETH was approximately $550, and XRP was around $0.62.
- This suggests that a dollar has more buying power with respect to XRP.
- However, it is also possible to speculate that with such high trading volumes in the hundreds of millions, the demand for XRP should be high, and thus the price should be as well.

In hopes of revealing further trend characteristics, XRP price was plotted against the full High, Low, Close (HLC) characteristics for BTC and ETH.

Figure: - Bitcoin High/Low/Close vs XRP

Figure: - Ethereum High/Low/Close vs XRP

Analysis & Observations:

- Despite attempting to break down BTC and ETH into their respective High, Low, and Close (HLC) constituents, no further trend was discovered.
- However, it was observed that both ETH and XRP experienced significant price surges after BTC's price peaked and then declined.
- This suggests that investors may have taken advantage of the BTC surge until it started to fall, and then shifted their attention to other coins such as XRP and ETH.
- This could be an additional trend to monitor, where a sudden drop in a particular coin's price may lead to an increase in the price of other coins.

6.3 XRP Response to Sentiment

Following the time series analysis, the study explored how sentiment affected prices. Three sentiment aggregates were created: summed, averaged, and RMS. Since RMS is a form of averaging, their values were quite similar, making it somewhat redundant to analyze both at the same time. Therefore, the focus was primarily on summed and RMS. If RMS had proven to provide more insights, then the averaged statistic would have been investigated further. However, while extracting the linear regression line for sentiment and price, and plotting the bivariate distributions, conflicting narratives emerged.

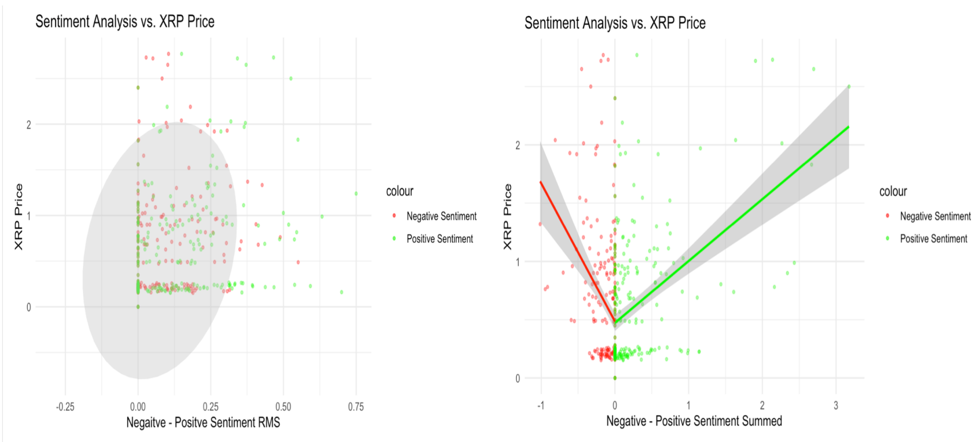


Figure: - Sentiment vs XRP Price. The figure on the left represents sentiment RMS. The figure on the right represent sentiment summed.

Observations:

1. The bivariate distribution shows negative sentiment is associated with a decrease in XRP price and positive sentiment with an increase in XRP price.
2. Clusters of negative and positive points around the lowest XRP price created an unexpected trend of increasing XRP price with increasing negative sentiment.
3. The RMS statistic may be too sensitive and not representative of how sentiment affects the price, while **summed sentiment appeared to be a better metric.**

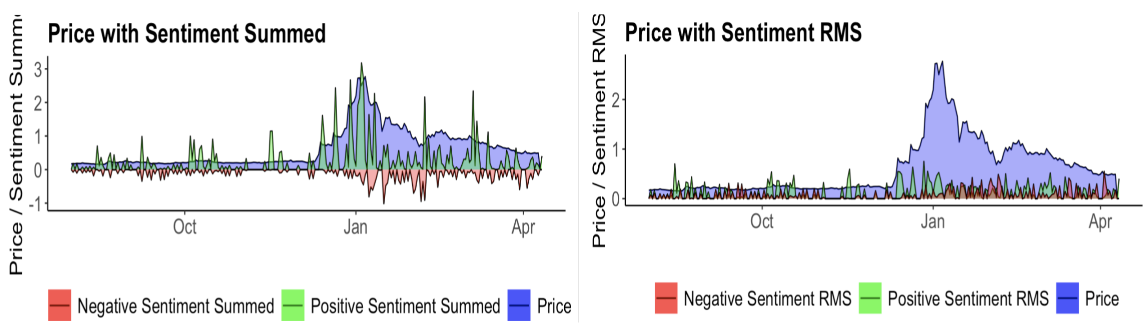


Figure: - XRP Price and Sentiment RMS time series.

XRP Price and Sentiment Summed time series.

6.4 Price Range or Exact Price

Correlating exact prices may have been too precise, so as an alternative method, XRP price was divided into 50 different ranges with an approximately $0.05 cent range for each. These ranges were labeled 0-49. While this method reduced price resolution, overlaying price ranges with actual prices still showed momentum consistency. A correlation matrix was then created using all variables to compare the correlation coefficients between the exact prices and the $ 0.05-cent price ranges. We visualized it as follows:

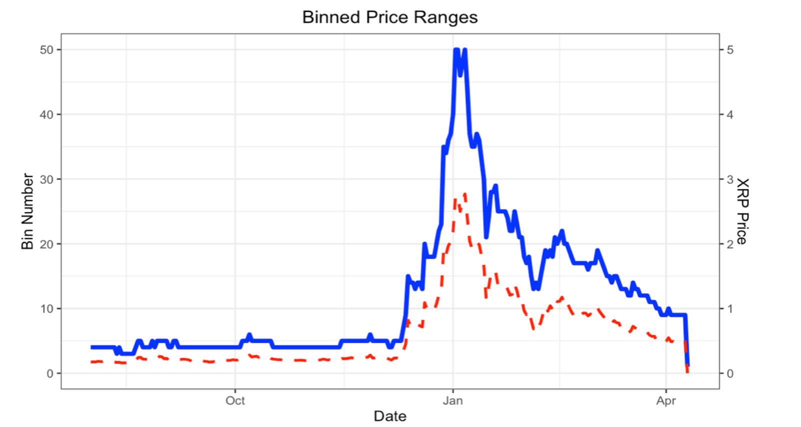


Figure: - Price ranges represented by bin numbers are illustrated on the left y-axis.

While binning the prices did not significantly improve any correlation coefficients, it also did not degrade them. Thus, it was considered more practical to use price ranges when predicting XRP price for the sake of accuracy.

7. Model Training & Evaluation

We evaluated the impact of sentiment (from Section 6.3) on the prices of ETH_Change, BTC_Change, and XRP_Change by testing three different models: **Random Forest, Lasso Regression, and Multiple Linear Regression**. We compared the performance of these models to select the best one for predicting the target variables based on the sentiment predictors.

Steps we Performed:

- **Feature engineering:** In each code, we prepared the data for training by selecting the predictor variables and target variable and splitting the data into training and testing sets using rsample. The training data is then used to build a random forest, Lasso, and Linear regression model using the appropriate package.
- **Evaluation metrics:** We evaluated model performance on the testing data using three evaluation metrics: mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R2). These metrics are commonly used to assess the accuracy and precision of regression models.
- **Model selection:** Based on the metrics computed, we made the comparisons between the models and determine which model was suitable.

7.1 Model Comparison & Validation for ETH

<table>
<tr><th>Random Forest</th><th>Multiple Regression</th><th>Lasso regression</th></tr>
<tr>
<td>Feature importance:
%IncMSE IncNodePurity
Positive Sentiment Averaged 4.856823 105.35951
Positive Sentiment Summed 3.533669 125.68368
Positive Sentiment RMS 2.731513 79.35708
Negative Sentiment Averaged 1.885812 101.29191
Negative Sentiment Summed 3.424963 286.75003
Negative Sentiment RMS 3.536219 143.08557
Overall Sentiment Averaged 4.898583 121.62625
Overall Sentiment Summed 3.219402 147.70866
Overall Sentiment RMS 3.569607 114.95023
ETH_Change 64.887106 4309.67462
ETH_High 7.093668 256.75443
ETH_Low 4.553614 364.96991
ETH_Open 7.332668 444.63137
ETH_Price 6.682067 355.75905</td>
<td>Call:
lm(formula = y_train ~ ., data = cbind(x_train, y_train))

Residuals:
Min 1Q Median 3Q Max
-7.923e-16 -1.134e-16 -7.550e-18 1.177e-16 9.982e-16

Coefficients: (3 not defined because of singularities)
Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.003e-16 4.675e-17 2.146e+00 0.0333 *
`Positive Sentiment Averaged` -6.077e-16 1.660e-15 -3.660e-01 0.7147
`Positive Sentiment Summed` -3.457e-18 7.652e-17 -4.500e-02 0.9640
`Positive Sentiment RMS` 1.788e-16 1.634e-15 1.090e-01 0.9130
`Negative Sentiment Averaged` 1.606e-16 2.364e-15 6.800e-02 0.9459
`Negative Sentiment Summed` 0.000e+00 3.323e-16 0.000e+00 1.0000
`Negative Sentiment RMS` 0.000e+00 2.537e-15 0.000e+00 1.0000
`Overall Sentiment Averaged` NA NA NA NA
`Overall Sentiment Summed` NA NA NA NA
`Overall Sentiment RMS` NA NA NA NA
ETH_Change 1.000e+00 6.680e-18 1.497e+17 <2e-16 **
ETH_High 7.935e-19 1.457e-18 5.450e-01 0.5867
ETH_Low -1.774e-18 7.743e-19 -2.291e+00 0.0232 *
ETH_Open 2.116e-19 1.332e-18 1.590e-01 0.8740
ETH_Price 6.185e-19 1.446e-18 4.280e-01 0.6694

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.608e-16 on 164 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 9.628e+33 on 11 and 164 DF, p-value: < 2.2e-16</td>
<td>15 x 1 sparse Matrix of class "dgCMatrix"
s1
(Intercept) 1.575710408
`Positive Sentiment Averaged` .
`Positive Sentiment Summed` 0.289774279
`Positive Sentiment RMS` -0.944771734
`Negative Sentiment Averaged` .
`Negative Sentiment Summed` -3.347228150
`Negative Sentiment RMS` 2.118187724
`Overall Sentiment Averaged` .
`Overall Sentiment Summed` .
`Overall Sentiment RMS` .
ETH_High .
ETH_Low -0.017273395
ETH_Open -0.113452167
ETH_Price 0.128841548
ETH_Vol -0.005537038
[1] "CV-MSE = 9.63"
[1] "CV-RMSE = 3.1"
[1] "CV-MAE = NaN"
[1] "MSE = 9.43"
[1] "RMSE = 3.07"
[1] "MAE = 1.93"</td>
</tr>
<tr>
<td>Model Test Results:
MAE: 1.16789
RMSE: 2.319866
R2: 0.885162</td>
<td>Model Test Results:
MAE: 4.662937e-15
RMSE: 7.936827e-15
R2: 1</td>
<td>Model Test Results:
MAE: 1.93
RMSE: 3.07
MSE: 9.43</td>
</tr>
</table>

All Plot is in Appendix C

Observation and Performance analysis (ETH_change):

Based on the metrics computed, we can make the following comparisons between the models:
- Random Forest has the lowest MAE and RMSE values, indicating better performance in terms of predicting the outcome variable.
- Linear regression has the highest R2 value, indicating the highest proportion of variance in the outcome variable that is explained by the predictor variables.
- Lasso regression has a higher RMSE and MAE than random forest, indicating poorer performance in predicting the outcome variable.
- However, Lasso regression does feature a lower number of predictor variables compared to random forest and may be preferred if feature selection is a priority.
- Overall, it seems that the **random forest model performs the best for predicting the ETH_Change variable.**

7.2 Model Comparison & Validation for XRP:

<table>
<tr><th>Random Forest</th><th>Multiple Regression</th><th>Lasso regression</th></tr>
<tr>
<td>Feature importance:
%IncMSE IncNodePurity
Positive Sentiment Averaged 8.170636 765.3795
Positive Sentiment Summed 4.083855 895.0518
Positive Sentiment RMS 7.361595 622.1261
Negative Sentiment Averaged 5.782527 209.6924
Negative Sentiment Summed 5.092410 194.1213
Negative Sentiment RMS 6.036347 221.1003
Overall Sentiment Averaged 6.008190 703.6800
Overall Sentiment Summed 6.620908 1647.7491
Overall Sentiment RMS 7.229924 735.0136
XRP_Change 40.180254 6179.2664
XRP_High 3.923946 297.7168
XRP_Low 5.299040 258.7642
XRP_Open 7.402623 338.3930
XRP_Price 5.728750 325.6622</td>
<td>Call:
lm(formula = y_train ~ ., data = cbind(x_train, y_train))

Residuals:
Min 1Q Median 3Q Max
-1.152e-15 -1.262e-16 5.260e-18 1.267e-16 1.130e-15

Coefficients: (3 not defined because of singularities)
Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.382e-17 4.056e-17 -2.066e+00 0.0404 *
`Positive Sentiment Averaged` 1.171e-15 2.430e-15 4.820e-01 0.6306
`Positive Sentiment Summed` 2.821e-17 1.012e-16 2.790e-01 0.7807
`Positive Sentiment RMS` -1.179e-15 2.399e-15 -4.910e-01 0.6238
`Negative Sentiment Averaged` -1.426e-15 3.341e-15 -4.270e-01 0.6700
`Negative Sentiment Summed` 0.000e+00 4.513e-16 0.000e+00 1.0000
`Negative Sentiment RMS` 0.000e+00 3.628e-15 0.000e+00 1.0000
`Overall Sentiment Averaged` NA NA NA NA
`Overall Sentiment Summed` NA NA NA NA
`Overall Sentiment RMS` NA NA NA NA
XRP_Change 1.000e+00 4.768e-18 2.097e+17 <2e-16 ***
XRP_High -2.297e-17 5.993e-16 -3.800e-02 0.9695
XRP_Low 6.296e-17 4.756e-16 1.320e-01 0.8949
XRP_Open -8.293e-16 5.888e-16 -1.409e+00 0.1609
XRP_Price 7.596e-16 6.693e-16 1.135e+00 0.2581

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

Residual standard error: 3.051e-16 on 163 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 1.396e+34 on 11 and 163 DF, p-value: < 2.2e-16</td>
<td>15 x 1 sparse Matrix of class "dgCMatrix"
s1
(Intercept) -1.483376e+00
`Positive Sentiment Averaged` .
`Positive Sentiment Summed` 1.897174e+00
`Positive Sentiment RMS` -1.186528e+00
`Negative Sentiment Averaged` .
`Negative Sentiment Summed` -1.168960e+00
`Negative Sentiment RMS` .
`Overall Sentiment Averaged` .
`Overall Sentiment Summed` .
`Overall Sentiment RMS` -1.729035e+00
XRP_High .
XRP_Low -2.814766e+00
XRP_Open -7.746569e+01
XRP_Price 7.908542e+01
XRP_Vol 9.648412e-05
[1] "CV-MSE = 50.57"
[1] "CV-RMSE = 7.11"
[1] "CV-MAE = NaN"
[1] "MSE = 19.59"
[1] "RMSE = 4.43"
[1] "MAE = 2.84"</td>
</tr>
<tr>
<td>Model Test Results:
MAE: 3.459598
RMSE: 8.214477
R2: 0.7859198</td>
<td>Model Test Results:
MAE: 1.76604e-15
RMSE: 2.607095e-15
R2: 1</td>
<td>Model Test Results:
MAE: 2.84
RMSE: 4.43
MSE: 19.59</td>
</tr>
</table>

All Plot are in Appendix C

Observation and Performance Analysis (XRP_change):

Based on the metrics provided, we can make the following comparisons between the models for predicting XRP_Change:
- Linear regression has the highest R2 value, indicating the highest proportion of variance in the outcome variable that is explained by the predictor variables.
- Lasso regression has a lower RMSE and MAE than random forest, indicating better performance in terms of predicting the outcome variable.
- Similar to the previous Model set, Since Lasso regression does feature a lower number of predictor variables compared to the random forest and may be preferred if feature selection is a priority.
- Random forest has the highest MAE and RMSE values, indicating poorer performance in terms of predicting the outcome variable compared to the other models.
- Overall, it seems that **the Lasso regression model performs the best for predicting the XRP_Change** variable.

7.3 Model Comparison & Validation for BTC:

Random Forest	Multiple Regression	Lasso regression
Feature importance: %IncMSE IncNodePurity Positive Sentiment Averaged 4.005675 98.05716 Positive Sentiment Summed 4.529305 114.18263 Positive Sentiment RMS 4.169218 107.41353 Negative Sentiment Averaged 5.748218 163.27819 Negative Sentiment Summed 5.570726 348.72101 Negative Sentiment RMS 5.278944 194.38201 Overall Sentiment Averaged 4.711107 133.54507 Overall Sentiment Summed 3.400478 186.07332 Overall Sentiment RMS 5.332314 112.00243 BTC_Change 62.097418 3664.11932 BTC_High 6.182752 177.28534 BTC_Low 7.423090 231.61088 BTC_Open 7.917085 276.89939 BTC_Price 5.980636 335.52130	Residuals: Min 1Q Median 3Q Max -7.174e-14 -1.410e-16 4.220e-16 8.950e-16 6.386e-15 Coefficients: (3 not defined because of singularities) Estimate Std. Error t value Pr(>| (Intercept) -1.407e-15 1.129e-15 -1.246e+00 0.2 `Positive Sentiment Averaged` 1.754e-14 3.792e-14 4.620e-01 0.6 `Positive Sentiment Summed` -1.934e-15 1.523e-15 -1.270e+00 0.2 `Positive Sentiment RMS` -1.473e-14 3.702e-14 -3.980e-01 0.6 `Negative Sentiment Averaged` 5.897e-15 5.703e-14 1.030e-01 0.9 `Negative Sentiment Summed` -1.285e-14 7.255e-15 -1.771e+00 0.0 `Negative Sentiment RMS` 7.059e-15 6.147e-14 1.150e-01 0.9 `Overall Sentiment Averaged` NA NA NA `Overall Sentiment Summed` NA NA NA `Overall Sentiment RMS` NA NA NA BTC_Change 1.000e+00 1.791e-16 5.584e+15 <2e BTC_High 8.491e-20 2.450e-18 3.500e-02 0.9 BTC_Low 3.431e-19 1.460e-18 2.350e-01 0.8 BTC_Open 1.061e-18 2.113e-18 5.020e-01 0.6 BTC_Price -1.356e-18 2.863e-18 -4.740e-01 0.6 --- Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Residual standard error: 5.725e-15 on 164 degrees of freedom Multiple R-squared: 1, Adjusted R-squared: 1 F-statistic: 1.759e+31 on 11 and 164 DF, p-value: < 2.2e-16	15 x 1 sparse Matrix of class "dgCMatrix" s1 (Intercept) 1.373243038 `Positive Sentiment Averaged` -1.381825899 `Positive Sentiment Summed` 0.018774472 `Positive Sentiment RMS` . `Negative Sentiment Averaged` . `Negative Sentiment Summed` . `Negative Sentiment RMS` . `Overall Sentiment Averaged` -0.476124831 `Overall Sentiment Summed` 0.229460090 `Overall Sentiment RMS` . BTC_High . BTC_Low -0.001562280 BTC_Open -0.007967875 BTC_Price 0.009433346 BTC_Vol -0.016039436 [1] "CV-MSE = 6.24" [1] "CV-RMSE = 2.5" [1] "CV-MAE = NaN" [1] "MSE = 4.64" [1] "RMSE = 2.15" [1] "MAE = 1.29"
Model Test Results: MAE: 1.170659 RMSE: 1.968712 R2: 0.9009511	**Model Test Results:** MAE: 2.278619e-15 RMSE: 3.055064e-15 R2: 1	**Model Test Results:** MAE: 1.29 RMSE: 2.15 MSE: 4.64

All Plot is in Appendix C

Observation and Performance Analysis (BTC_change):

Based on the metrics provided, we can make the following comparisons between the models for BTC_Change prediction:

- Random forest has the lowest MAE and RMSE values, indicating better performance in terms of predicting the outcome variable.
- Linear regression has the highest R2 value, indicating the highest proportion of variance in the outcome variable that is explained by the predictor variables.
- Lasso regression has a lower RMSE and MAE than random forest, indicating better performance in predicting the outcome variable.
- Overall, it seems that the **random forest model performs the best for predicting the BTC_Change** variable, similar to the ETH_Change prediction.

8. Price Forecasting

The following steps were taken to perform Price Forecasting-

1. **Data preparation**:	Clean, transform, normalize, and account for seasonality, trend, and cyclical patterns.
2.Feature engineering:	Created additional features to improve the model's performance, such as lagged variables, moving averages, or Fourier transformations
3. **Train-test split:**	Split the data into training and testing sets.
4. **Model selection**:	Selected an appropriate time series model, such as ARIMA, exponential smoothing, and naïve method models
5. **Model fitting:**	Fit the chosen time series model to the training data using log-likelihood estimation.
6. **Model evaluation:**	Evaluate the performance of the model on the testing data using metrics such as mean absolute error (MAE), root mean squared error (RMSE), or mean absolute percentage error (MAPE).
7. **Forecasting:**	Used the fitted model to forecast future values of the time series, we forecasted future values for the next 7 days.

Following are the Model Statistics and Plots

8.1 ARIMA

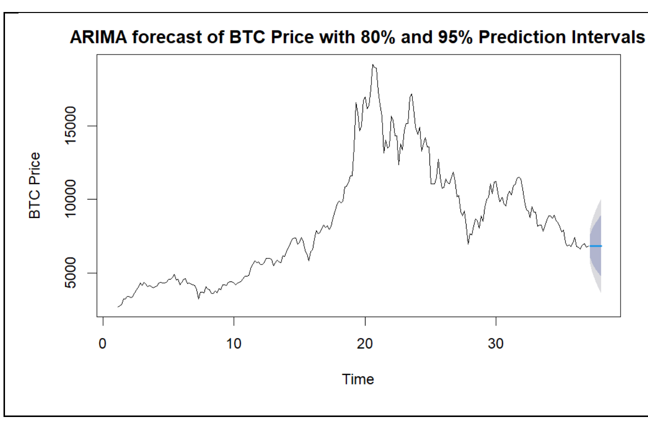
1. Model Statistics:

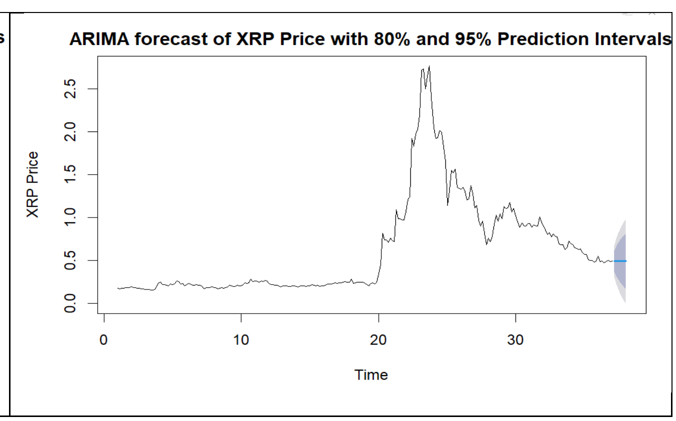
```
Series: ts_xrp
ARIMA(0,1,0)

sigma^2 = 0.009205:  log likelihood = 233.12
AIC=-464.25   AICc=-464.23   BIC=-460.72

Training set error measures:
                      ME       RMSE        MAE         MPE     MAPE     MASE
Training set 0.001233433 0.09575035 0.04434055 -0.01139132 5.775973 0.28168
                  ACF1
Training set 0.0811575
```

2. Model fitting and Price Forecasting (7-Day):





8.2 Exponential smoothing model

1. Model Statistics:

```
Call:
 ets(y = ts_xrp)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.0847

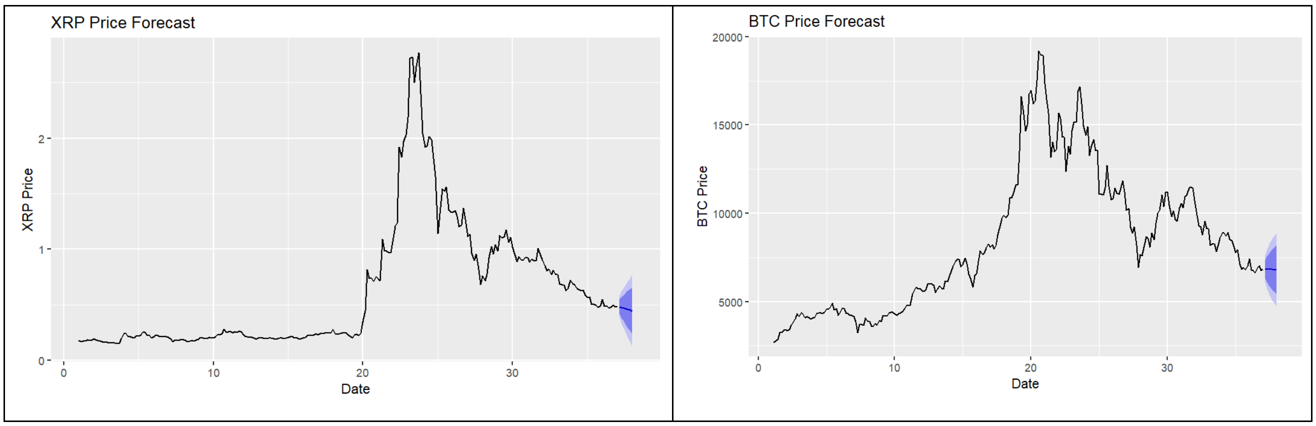
  Initial states:
    l = 0.1772
    b = -8e-04

  sigma:  0.1034

     AIC      AICc       BIC
-170.7379 -170.4950 -153.0710

Training set error measures:
                       ME       RMSE       MAE       MPE     MAPE      MASE
Training set -0.00025826 0.09678306 0.0457585 0.1398367 5.978353 0.2906877
                   ACF1
Training set 0.02155518
```

2. Model Fitting and Price Forecasting(7-Day)



8.3 Naive model

1. Model Statistics:

```
Forecast method: Seasonal naive method

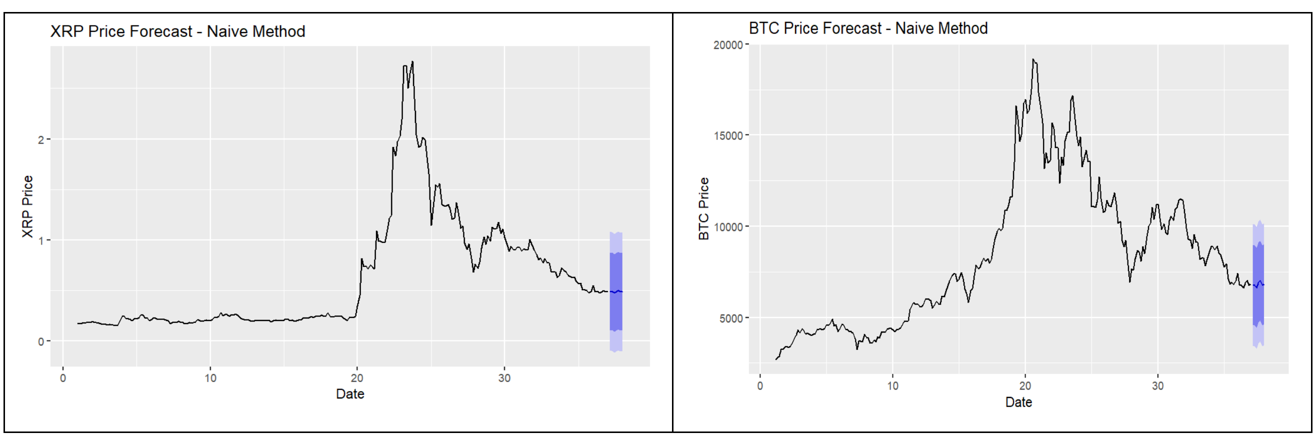
Model Information:
Call: snaive(y = ts_xrp)

Residual sd: 0.2995

Error measures:
                      ME      RMSE       MAE        MPE     MAPE MASE      ACF1
Training set 0.008819593 0.2995429 0.1574146 -0.7186905 17.83764    1 0.8965148

Forecasts:
```

2. Model fitting and Price Forecasting (7-Day)



8.4 Model Validation

To determine which forecasting method is better among ARIMA, seasonal naive method, and exponential smoothing model, we can compare their error measures and AIC/BIC values.

<u>For ARIMA:</u>

AIC=-464.25, AICc=-464.23, BIC=-460.72

Training set RMSE = 0.09575035

<u>For Exponential Smoothing:</u>

AIC=-170.7379, AICc=-170.4950, BIC=-153.0710

Training set RMSE = 0.09678306

<u>For Seasonal Naive Method:</u>

Training set RMSE = 0.2995429

Based on the AIC/BIC values, the **Exponential Smoothing** model has the lowest AIC/BIC value, indicating that it best balances goodness-of-fit and model complexity.
Based on the RMSE values, the ARIMA model has the lowest training set RMSE, indicating that it best fits the training data.

9. Conclusion

Based on the analysis performed, we can conclude that the models implemented to predict price change using sentiment analysis for Ethereum, XRP, and BTC were successful in predicting the trends in the market. The Random Forest model was found to be the best fit for Ethereum and BTC, while the Lasso regression model performed the best for XRP. In the Time Series Forecasting, the Exponential Smoothing model was found to provide the best balance between goodness-of-fit and model complexity, although the ARIMA model had the lowest RMSE values. These results can be useful for making informed decisions regarding cryptocurrency investments and trading. However, it is important to note that cryptocurrency markets are highly volatile, and the performance of these models may change over time. Therefore, further research and testing are recommended to ensure the effectiveness of these models in the long term.

10. Future Scope

For future work, we can explore more recent libraries such as XGBoost regression and NN (Neural Network) models in order to improve the predictions. Other than the recent models, we can also explore various machine learning algorithms such as SVM and Decision Trees in hopes of improving the results.

Bibliography

1. J. I. Wong, J. Goh, and J. W. Lim, "Cryptocurrency price prediction using news and social media sentiment with LSTM and GRU networks," IEEE Access, vol. 6, pp. 69195-69206, 2018.
2. F. Egging and A. Gregoriou, "Predicting the price of Bitcoin using sentiment analysis," in 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), Las Vegas, NV, 2015, pp. 467-472.
3. H. Wang and J. Zhang, "Bitcoin price prediction using sentiment analysis," in 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China, 2018, pp. 175-179.
4. A. Gautam and G. Ganapathy, "Predicting cryptocurrency prices with machine learning using technical indicators and sentiment analysis," in 2020 IEEE 16th International Conference on Intelligent Computing and Smart Systems (ICICSS), Dubai, United Arab Emirates, 2020, pp. 166-171.

APPENDIX A POSITIVE WORD LIBRARY

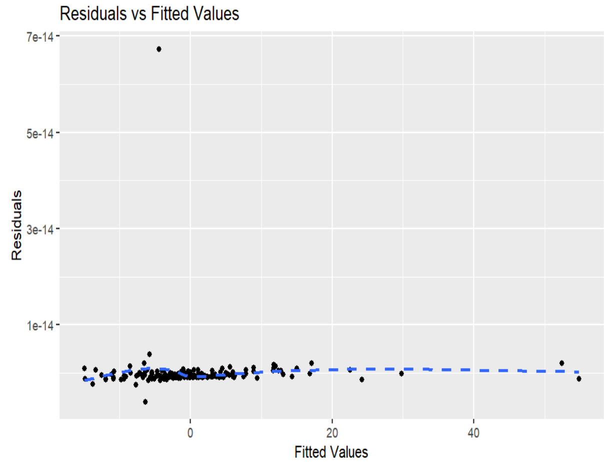
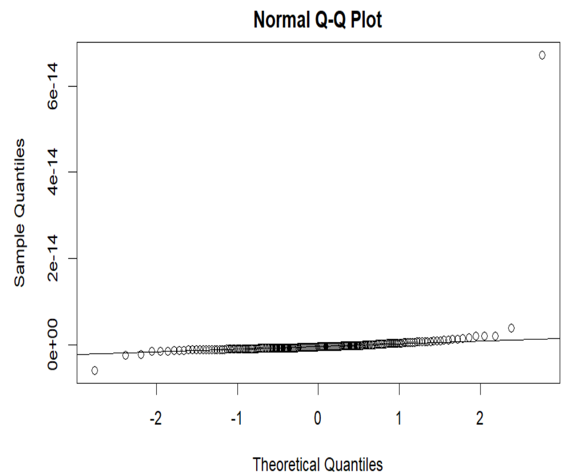
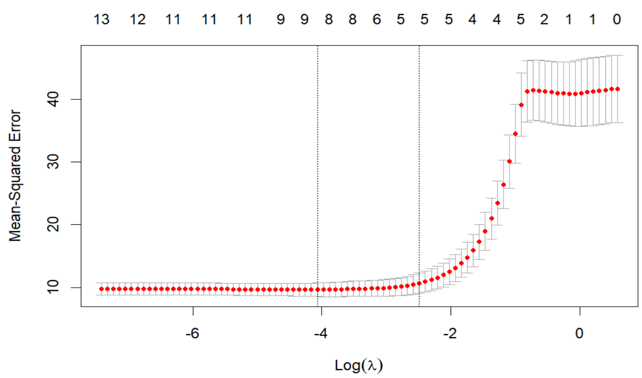
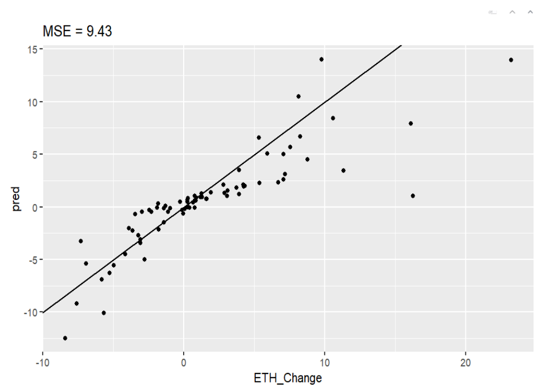
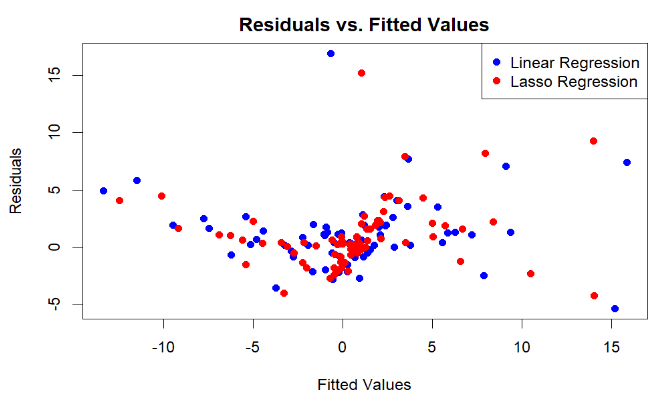
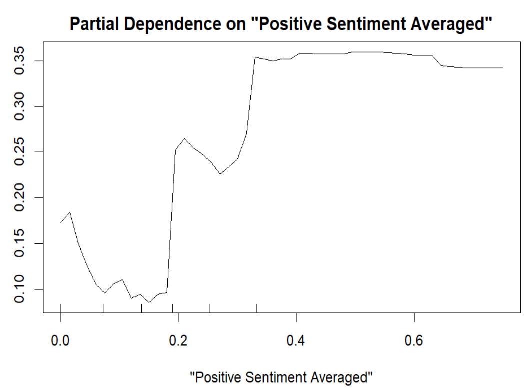
climbs	useful	benefited	opportun	cheap
climbing	spiked	benefit	winner	cheaply
climbed	spike	high	win	testing
climb	strong	additional	boom	test
surge	increase	stable	uptick	driving
surges	rise	safe	surpass	drive
surged	recovered	appealing	frenzi	velocity
surging	recover	appeal	largest	bull
surgin	recov	soar	large	bullish
surg	new	driving	rose	best
up	partnership	drive	rise	milestone
big	bull	record	climbed	rich
leap	optimistic	moon	climb	richer
leaps	stong	gain	blockbuster	richest
jumps	increase	increase	appreciating	soared
jump	value	increas	appreciate	soar
jum	serious	up	rally	outpacing
higher	posit	stabl	rallies	outpace
high	surgeasy	appeal	rallying	outpac
growth	easier	buzz	increase	best-performing
unprecedented	easi	thriving	momentum	overtook
unpreced	run	thrive	robust	overtake
gains	advantage	banner	benefit	increase
gain	superior	leapt	benefitting	increas
fortune	partnership	leap	interested	growing
earned	partnering	grew	interest	grow
earn	risen	shot	blockbust	cement
skyrocketed	strong	progressing	appreci	cemented
millionaires	climbed	progress	ralli	attract
lucky	climb	astounding	increas	attracted
valuable	rising	astound	benefit	fuel
wealthiest	rise	above	swell	recovery
surpassed	fantastic	explosion	strong	recover
richest	fantast	unprecedented	upward	recovered
richahead	inch	newcomers	milestones	impress
headlines	closer	popular	milestone	impressive
rich	close	record	appreciation	optimistic
largest	explosive	reliable	partnership	upside
huge	explos	mad	partnering	implement
skyrocket	gains	comfortable	boost	implementing
lucky	gain	opportunity	benefits	overtake
lucki	growth	uptick	benefit	overtaken
billionair	jumping	frenzy	positive	overtaking
billionaire	jump	useful	fomo	rebound
surpass	historic	appealing	appreci	rebounds
richest	histor	big	higher	rebounding
ahead	milestone	jump	raised	strong
return	mileston	fascinating	raise	stronger
rich	premier	fascinate	rally	bargain
spiked	record	explos	admire	bargains
spike	quick	unpreced	admirable	testing
spikes	cost-effective	hord	attract	heights
spiking	astronomical	newcom	attracted	

APPENDIX B NEGATIVE WORD LIBRARY

loss	losing	spooked	collapse	removing
losses	declines	spook	regulating	restrict
lose	correction	suffer	regulate	restricting
loser	fades	suffering	regulated	regulating
drop	fading	suffered	oversold	regulate
dropped	fade	underperformance	selloff	regulations
drops	fad	underperforming	bubble	sink
ban	correcting	underperform	crushed	sinking
banned	pullback	overvalued	crush	sinks
banning	retract	overvalue	burst	struggle
crackdown	shed	plunge	bursting	struggles
worst	lower	plunging	pop	struggling
worst-performing	low	dips	popping	sell
worthless	lowest	dip	popped	invalidate
fell	retractment	dipping	worse	invalidated
fall	uncertain	depreciating	worst	tricky
decrease	uncertainty	depreciate	worsen	reluctant
weak	adverse	depreciated	crashing	reluctance
weakening	ban	nervous	crashed	barrier
weakened	banning	losing	crash	barriers
less	banned	suffered	exclude	excludes
volatile	fud	suffer	excluding	exclude
hard	pull	suffering	removed	fomo
fade	back	shaken	remove	fear
fades	low	shaking	removes	risk
decline	lows	shake	complain	risks
declined	decline	dips	complains	risking
declining	pullback	dipping	complained	vanish
down	fade	dipped	failure	vanishing
bear	fading	shaken	failures	bubble
bearish	faded	shy	failed	tumble
bear	dumping	shaky	fail	tumbles
bears	dumped	skittish	resist	tumbled
panic	dump	volatile	resistance	flee
bad	corrections	tumultuous	resisting	nervous
skeptical	below	headwinds	resisted	freeze
worse	restrict	headwind	weak	freez
artificial	restricting	negative	turmoil	red
problem	restrictions	battered	difficult	decline
problems	regulation	batter	slump	disappoint
risk	regulating	banned	slumped	disappointment
risky	regulate	ban	slumping	disappointing
fallen	red	banning	failed	scrutiny
falling	downward	collapsed	failing	
fallin	crack	collapsing	fail	

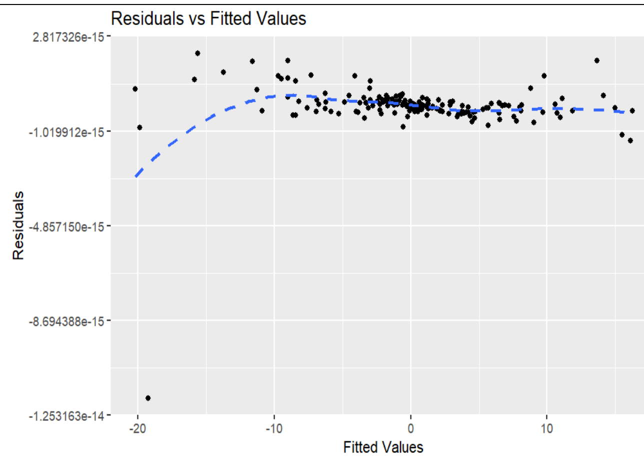
Appendix C

Implementation of the model on ETH data:

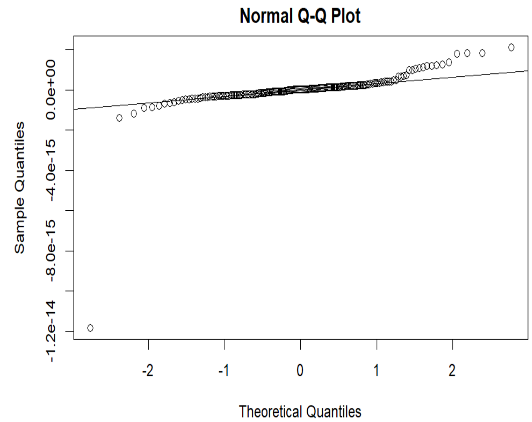
Plot for Multiple linear regression	Plot for Multiple linear regression
	
Plot for Lasso Regression:	**Plot for Lasso Regression:**
	
Plot for Lasso Regression:	**Plot for Random Forest:**
	

Implementation of model on XRP data:

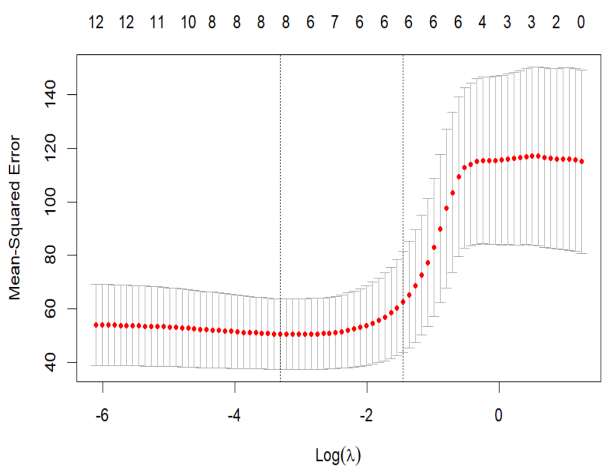
Plot for Multiple linear regression



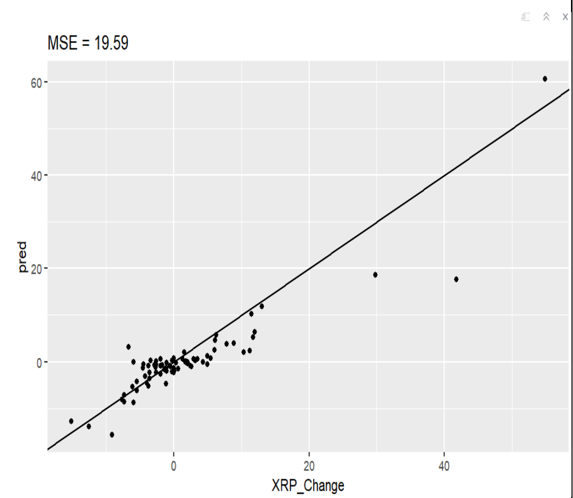
Plot for Multiple linear regression



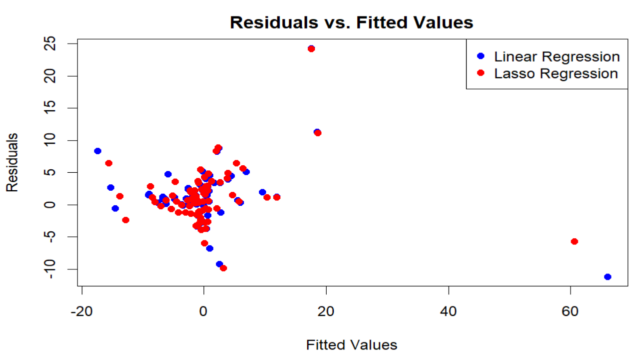
Plot for Lasso Regression:



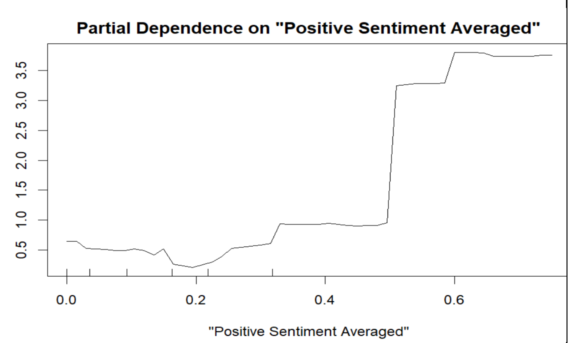
Plot for Lasso Regression:



Plot for Lasso Regression:

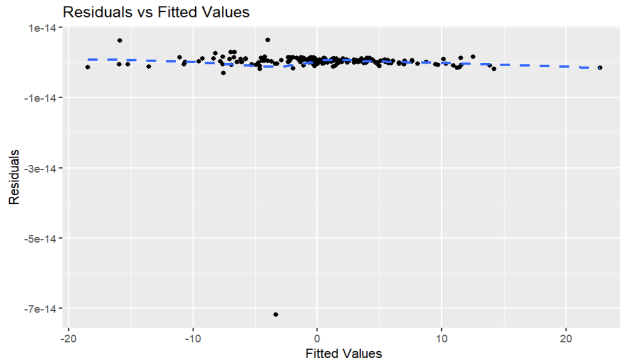


Plot for Random Forest:

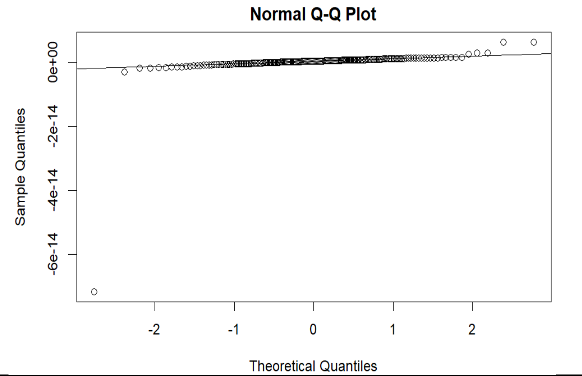


Implementation of model on BTC data:

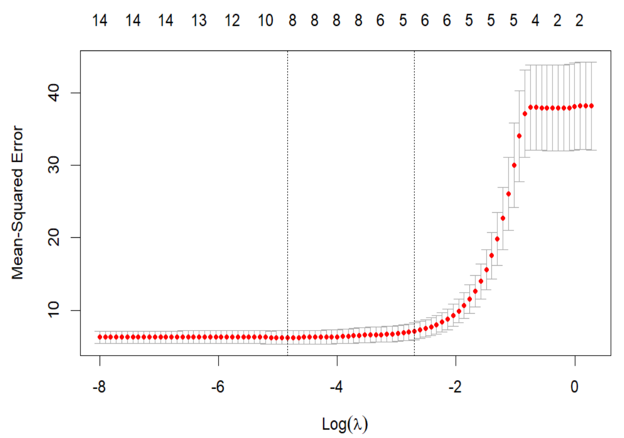
Plot for Multiple linear regression



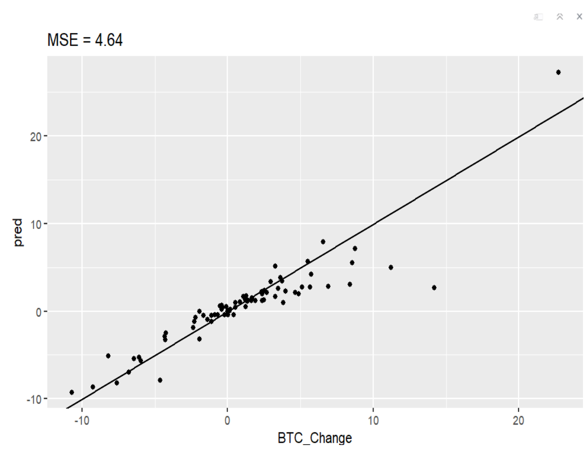
Plot for Multiple linear regression



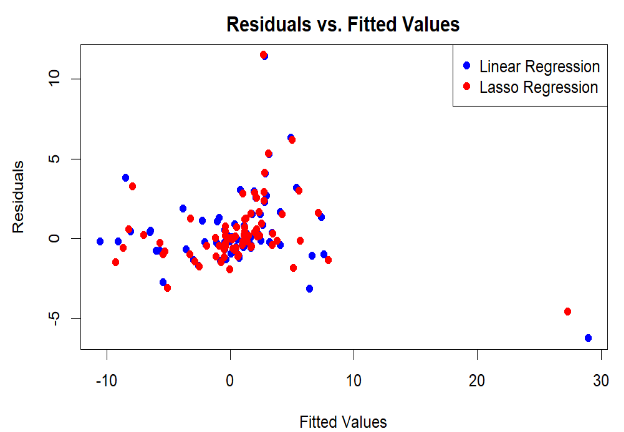
Plot for Lasso Regression:



Plot for Lasso Regression:



Plot for Lasso Regression:



Plot for Random Forest: