ETHEREUM ANALYSIS FOR LOOPRING, FUNFAIR AND VEROS TOKENS BASED ON TRANSACTIONS

MALEEHA KOUL AND SHAYAN SADAR

PROBLEM DESCRIPTION:

The project aims to analyze the Ethereum tokens data and find the pattern of buyer and seller transactions and deduce meaningful insights based on statistical methods of data science. We aim to find a distribution that best fits the transaction patterns between the buyer and seller pairs. It is also essential to find out the frequent buyers for each token and find the top 'k' buyers that gives the best correlation between the parameters that can help discover dependencies amongst data.

ETHEREUM AND ERC20 TOKENS:

Ethereum is a global, decentralized platform which is being used for money and other new kinds of applications. One can write code on the Ethereum network that controls money and build applications accessible anywhere in the world. Ethereum has the goal of using a blockchain to replace internet third parties — those that store data, transfer mortgages and keep track of complex financial instruments.

ERC: ERC20 is a guide of rules and regulations that will help create a blueprint for Ethereum-based smart contracts to create their tokens. The "ERC" stands for to Ethereum Request for Comment, while the number '20' is the number assigned to this request.

PRIMARY TOKENS:

LoopRing is an open-sourced token exchange protocol for decentralized exchanges. It is an "automated execution system" that facilitates trade across different cryptocurrency exchanges. Loopring trades are done via a set of smart contracts that are open and free and can be used by Decentralized Applications (dApps). The number of subunits are 10^18.

FunFair is a decentralized, cryptocurrency-based casino gaming platform built on the Ethereum smart-contract blockchain. The FUN token is the proprietary ERC-20 cryptocurrency token for the FunFair platform. It's the only token accepted for in-game credits, how game creators in the marketplace receive payment, what casinos must pay their licensing with and receive revenues in, and all fees on the platform must be paid in FUN. The number of subunits are 10^8.

VEROS Cryptocurrency for which you can buy both food and technology products from VeDH Platform's projects or its partners. Veros is a zero-fee fundraising platform where the majority of all advertisement revenue funnels directly into fundraisers launched on the platform. The number of subunits are 10^18.

APPROACH:

Data preprocessing is an essential part of this project. If the Circulation Amount was exceeded by the Token amount, then the data conveys that those particular transactions are spurious so we removed them.

The Buyer Seller pair frequencies were identified and the histogram for the frequencies was plotted and the distribution to fit has to be identified.

The population parameters for the data are studied to explore the direction in which meaningful insights could be found. Based on these parameters of interest, the top 'k' buyers are selected by tuning the 'k' value. Feature extraction is performed for these parameters that provide most information useful for creating the regression model.

PACKAGES USED:

- plyr
- MASS
- ggplot2
- fitdistrplus
- readxl

IMPLEMENTATION:

Read the data and count the number of rows.

```
raw_data <- read.csv(file = 'networkloopringTX.txt', header = F, sep = ' ')
colnames(raw_data) <- c("Buyer", "Seller", "Timestamp", "TokenAmount")
message ('The number of rows are: ', nrow(raw_data))
```

```
## The number of rows are: 204222
```

Find the number of outliers in the data by comparing the token amount with the total circulation amount.

```
Total_circulation_amount = 828954240 * 10^18
outliers <- subset(raw_data, TokenAmount > Total_circulation_amount)
message ('The number of outliers in the dataset are: ', nrow(outliers))
```

```
## The number of outliers in the dataset are: 2
```

Finding the Buyer-Seller pair frequencies.

```
library(plyr)
Buyer_Seller_Pair_Frequencies <- ddply(preprocessed_data, .(preprocessed_data$Buyer, preprocesse
d_data$Seller), nrow)
names(Buyer_Seller_Pair_Frequencies) <- c("Buyer", "Seller", "Frequency")
Buyer_Seller_Pair_Frequencies
```

Buyer <int>	Seller <int>	Frequency <int>
82	2964307	1
6	2964307	1
40002	3274516	1
82	1815762	42
44	4848203	1
3078280	3300522	1
222770	4848204	1
5	1991385	1
3300522	5	1
5	305723	1

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

The summary of the pair frequencies tells us that the data is highly skewed.

```
##      Buyer              Seller           Frequency
##  Min.   :      4   Min.   :      4   Min.   :   1.000
##  1st Qu.:  89826   1st Qu.: 289397   1st Qu.:   1.000
##  Median :1871198   Median :1935441   Median :   1.000
##  Mean   :2262788   Mean   :2345840   Mean   :   1.854
##  3rd Qu.:4849192   3rd Qu.:4852638   3rd Qu.:   1.000
##  Max.   :4876603   Max.   :4876611   Max.   :1469.000
```

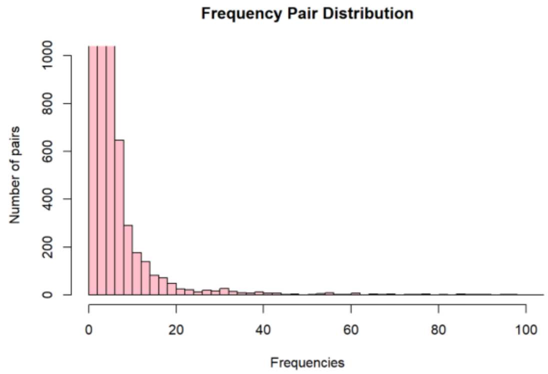
```
message ('Variance: ', var(Buyer_Seller_Pair_Frequencies$Frequency))
```

```
## Variance: 215.627814313744
```

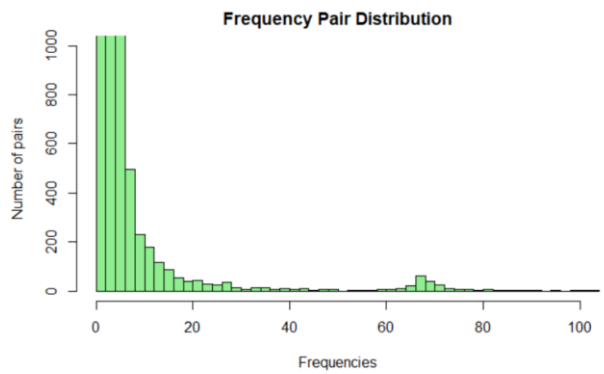
```
message ('Standard Deviation: ', sd(Buyer_Seller_Pair_Frequencies$Frequency))
```

```
## Standard Deviation: 14.6842709833939
```

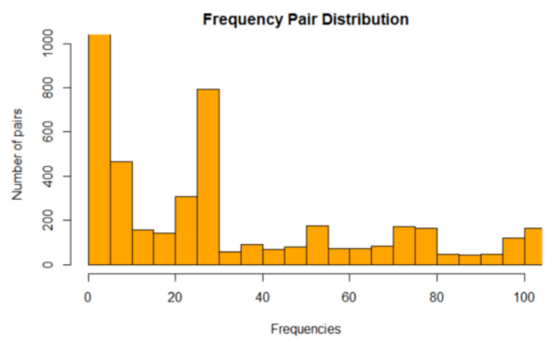
The frequency pair distributions for our tokens is as follows:



Funfair

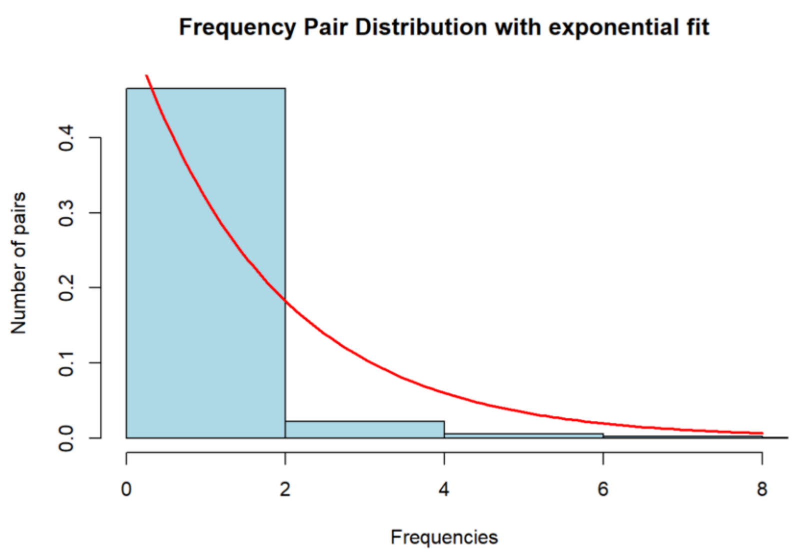


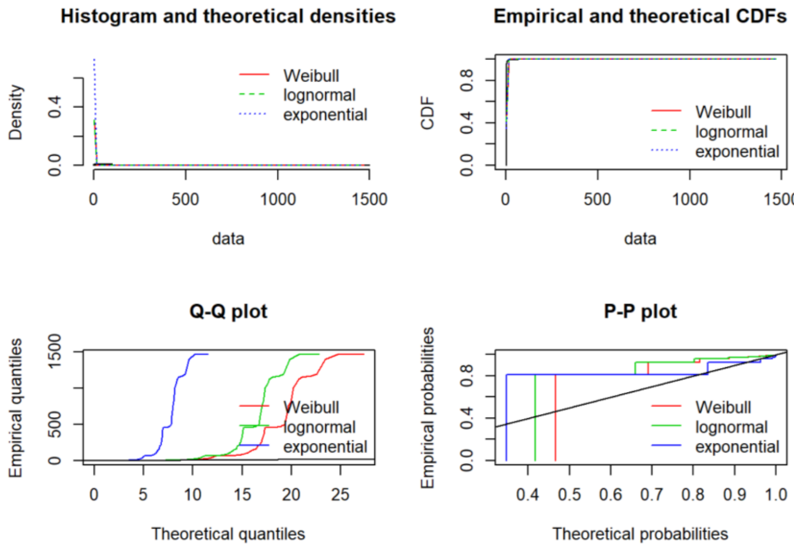
Loopring



Veros

As the data is highly skewed we standardized the bars to see if the distribution that fits on the entire also fits on the standardized data. Both Loopring and Funfair did not change much, but Veros did a tad bit. But on the whole Veros also fits the exponential curve like the other tokens.





After plotting the distributions and analyzing them, the next step is to find the features from the top 'k' buyers to infer correlations for studying the patterns in Token price for each of the selected tokens.

The timestamp from the existing dataframe had to be converted from UNIX to YYYY-MM-DD for joining with the historical data from Coin Market Cap.

	Buyer <int>	Seller <int>	Timestamp <date>	TokenAmount <dbl>
1	82	2964307	2018-04-24	9.510000e+01
2	6	2964307	2018-04-24	1.870149e+04
3	40002	3274516	2018-04-24	1.839461e+03
4	82	1815762	2018-04-24	8.389100e+03

The Coin Market Cap historical data looks as follows:

Timestamp <date>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <dbl>	MarketCap <dbl>
2019-04-23	0.067044	0.068005	0.062899	0.063292	13938070	52466100
2019-04-22	0.068028	0.069217	0.065807	0.067036	15094051	55569829
2019-04-21	0.074267	0.074674	0.065887	0.068242	17061511	56569575
2019-04-20	0.076099	0.076315	0.073206	0.074280	17978822	61575008
2019-04-19	0.073970	0.078721	0.071970	0.076099	21377192	63082414
2019-04-18	0.071932	0.076205	0.071384	0.073996	21031814	61339614

By joining the two dataframes we were able to identify outliers as some of the rows were NA as the all the dates did not match on joining. The second method of identifying additional outliers is by checking the token amount to market cap ratio and remove rows with a ratio higher than 1.

```
Top_Buyers <- subset(joined_df, percentage < 100)
track_k_buyers <- head(Top_Buyers, 70)
nk <- (unique(track_k_buyers))
Top_Buyers
```

	Buyer	Timestamp	TokenAmount	Open	High	Low	Close	Volume	Market
	<int>	<date>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<(
4	4853589	2018-03-12	161067928.53	0.376652	0.377374	0.338550	0.346355	3356110	198140
5	299810	2018-03-12	140000000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
12	4853608	2018-03-12	89980000.00	0.376652	0.377374	0.338550	0.346355	3356110	198140
17	4853600	2018-03-12	54001283.71	0.376652	0.377374	0.338550	0.346355	3356110	198140

ANALYSIS AND INFERENCE:

Inference 1: Buyer Seller pairs have the highest frequency for the interval of 0-20 transactions as there are many low count transactions taking place which means buyer seller pairs do not transact with same ID exceptionally.

Inference 2:

For token Loopring we got the value of K as:

```
message('The value of K is: ',nrow(count(nk)))
```

```
## The value of K is: 35
```

K Value for –
Veros: 7
FunFair: 58

The Pearson's correlation between Token Amount and MarketCap shows that there is somewhat dependency between the two features. The Token Amount somewhat drives the value of MarketCap.

```
cor.test(track_k_buyers$TokenAmount, track_k_buyers$MarketCap, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  track_k_buyers$TokenAmount and track_k_buyers$MarketCap
## t = 4.466, df = 68, p-value = 3.085e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2716537 0.6396279
## sample estimates:
##       cor
## 0.4762292
```

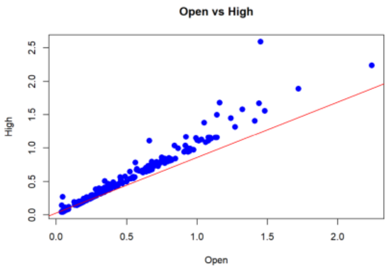
Correlation between the Open-High and Close-High was also found. A higher correlation is found between Close-High which shows that the highest token price during the day is better correlated to the closing price of the token when compared against the opening price.
Approximately were found in other correlations as well for Token Amount and Market Cap:
Loopring: 0.48
FunFair: 0.51
Veros: 0.86

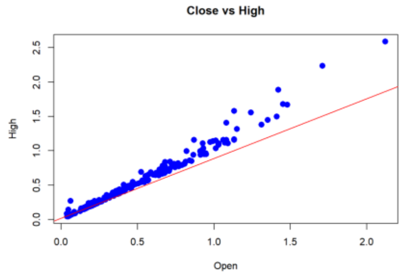
The Linear Model was fitted between these features:

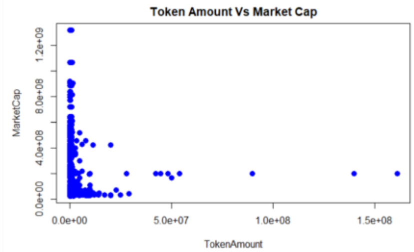
```
linearModOH <- lm(Open ~ High, data=Top_Buyers)  # build linear regression model on full data
linearModCH <- lm(Close ~ High, data=Top_Buyers)
linearModTM <- lm(Close ~ High, data=Top_Buyers)
summary(linearModOH)
```

```
##
## Call:
## lm(formula = Open ~ High, data = Top_Buyers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73137 -0.02545 -0.00422  0.03887  0.34917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0314175  0.0003413   92.06   <2e-16 ***
## High        0.8300967  0.0004613 1799.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09558 on 183757 degrees of freedom
## Multiple R-squared:  0.9463, Adjusted R-squared:  0.9463
## F-statistic: 3.238e+06 on 1 and 183757 DF,  p-value: < 2.2e-16
```

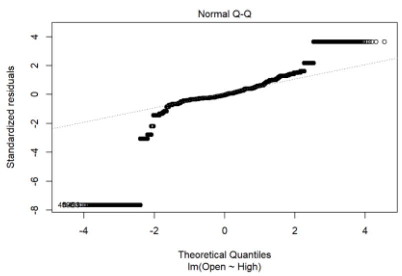
The plots are as follows:

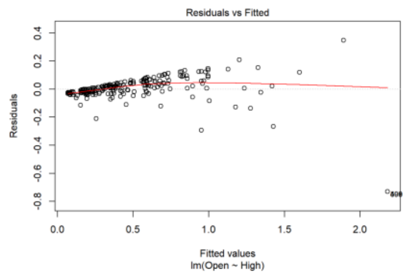


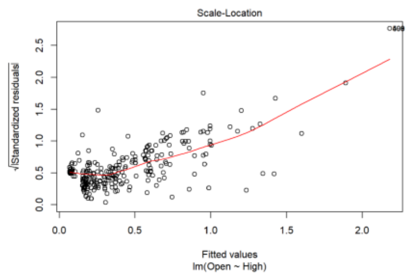




The plots for residuals from the above correlations are:







CONCLUSION:

After analyzing the token data, we found out that it's price did not depend on the transaction information greatly. We got a rather high correlation with the coin price for the opening and the highest price for the day. Also barring few spikes, there was not much jump in the coin prices as well. So we do not have enough data to hypothesize any conclusion regarding how the price should change for this token. More data is required to have a deeper understanding of the transactions. For example, looking at the transaction data, we cannot tell how this affects the net available coins in the market and so on. Gathering that information will help us greatly to find a strong model for the Ethereum tokens.