Project: Analysis of Salt tokens

Code

Nikita Vispute, Navya Hirebidanur Chandrashekara

December 01, 2018

INTRODUCTION:

Ethereum:

Ethereum is an open-source, public, blockchain-based distributed computing platform and operating system that enables developers to build and deploy decentralized applications. Vitalik Buterin is a Russian-Canadian programmer who is the co-founder of Ethereum.He released a white paper in 2013 describing an alternative platform designed for any type of decentralized application developers would want to build. The system was called ethereum. Ethereum makes it easy to create smart contracts, self-enforcing code that developers can tap for a range of applications.In the Ethereum blockchain, instead of mining for bitcoin, miners work to earn Ether, a type of crypto token that fuels the network. Beyond a tradeable cryptocurrency, Ether is also used by application developers to pay for transaction fees and services on the Ethereum network.

Token:

It is some form of money, but often it can represent something else entirely - membership in a program, for instance, or proof of ownership. In crypto, tokens have many meanings. Bitcoin is the primary token in cryptocurrency: the oldest, most valuable, and perhaps the most used.Most tokens have a fixed or limited supply decided by the issuer. Their value is often related to said supply, but it's not always important. In the case of the base Ether token (ETH), for instance, there is fundamentally an intent to continue minting new tokens forever. Ether is required to launch and use various other tokens on the Ethereum blockchain. Hence ETH is fairly often the most valuable token on its blockchain, in coin price and market cap terms.

ERC20 Tokens:

ERC-20 tokens are tokens designed and used solely on the Ethereum platform. ERC-20 is the universal language that all tokens on the Ethereum network use. It allows one token to be traded with another.They follow a list of standards so that they can be shared, exchanged for other tokens, or transferred to a crypto-wallet.
Optional: Token Name, Symbol,Decimal (up to 18)
Mandatory: totalSupply, balanceOf, transfer, transferFrom, approve, allowance

Selection of dataset:

UTD ID of Nikita: 2021421460
UTD ID of Navya: 2021430477
Sum of the IDs modulo 20: 18
18th biggest token dataset : networksaltTX

Salt Token :

Since the Ethereum blockchain network first launched in July 2015, some 500+ cryptocurrency projects have been built on top of it as ERC20, ERC223, or ERC777 tokens.Salt(Secured Automated Lending Platform) was launched in October 2017.It is led by Shawn Owen as CEO. The platform has almost 50,000 members and has funded over $7,000,000 in Bitcoin and Ethereum backed loans. SALT is the current leader in blockchain-based loans.The SALT lending platform is a great option if you want/need to make some real-world expenses and don’t want to lose the potential gains from your crypto holdings. Beyond that, the project works to solve a major problem of blockchain assets - illiquidity. By opening up an entirely new form of loans, the project brings more liquidity to the cryptocurrency market. Borrowers simply put their blockchain-based assets up as collateral in a smart contract and are quickly matched with capital from an extensive network of qualified lenders.

Description of what we are trying to model:

In this project, we will examine the ethereum dataset: salt. We will modify this data first to remove outliers, and then find the distribution for the number of transactions by the buyer and similarly for the seller. Then we will create layers for the number of transactions based on increasing token Amount and find the correlation between each layer and the token price dataset. Lastly we will create a multiple linear regression model using one single layer with highest correlation.

Loading the dataset:

Once the dataset is loaded into RStudio, we re-named columns of the dataset to:fromNodeID, toNodeID, unixTime, tokenAmount.

Code

Outliers in the Dataset :

Outliers in an Ethereum dataset are all those transactions whose tokenamount > TotalSupply*Decimal of those respective tokens. Based on the mentioned criteria, our dataset had more than 30 outliers and we decided to eliminate all those outlier transactions. Total Supply : 120,000,000 (Source:https://coinmarketcap.com/ (https://coinmarketcap.com/)) SALT & Decimal : 8 (Source:https://etherscan.io/token/0x4156D3342D5c385a87D264F90653733592000581 (https://etherscan.io/token/0x4156D3342D5c385a87D264F90653733592000581)) Calculating for the outliers whose tokenAmount > 120000000*10^8

Code

```
## Number of outliers in the SALT data set is : 114
```

Code

	fromNodeID <int>	toNodeID <int>	unixTime <int>	tokenAmount <dbl>
30326	361967	368287	1521603007	1.000000e+18
30357	361967	368287	1521610381	1.000000e+17
58844	5827459	1916571	1518064225	8.000000e+19
58854	5827459	1916571	1518065441	8.000000e+19
58863	5827459	1916571	1518066624	8.000000e+19
58869	5827459	1916571	1518067821	8.000000e+19
58876	5827459	1916571	1518069028	8.000000e+19
58881	5827459	1916571	1518070238	8.000000e+19
58890	5827459	1916571	1518071450	8.000000e+19
58895	5827459	1916571	1518072627	8.000000e+19

1-10 of 114 rows Previous 1 2 3 4 5 6 ... 12 Next

Code

```
## Number of users included in these transcations are : 17
```

Filtering Outliers:

We filter the dataset by removing the outliers found in the earlier step.

Code

Distribution for Number of times a user sells:

To find number of times a user sells, initially we considered the actual dataset without outliers and found the frequency of each user (UserIDs are present in the fromNodeID column) in the table and stored the result in “data1”" table.The next step was to find the number of times each value in frequency column of data1 table is repeated. The displayed resultant table was stored in “data2”.

Code

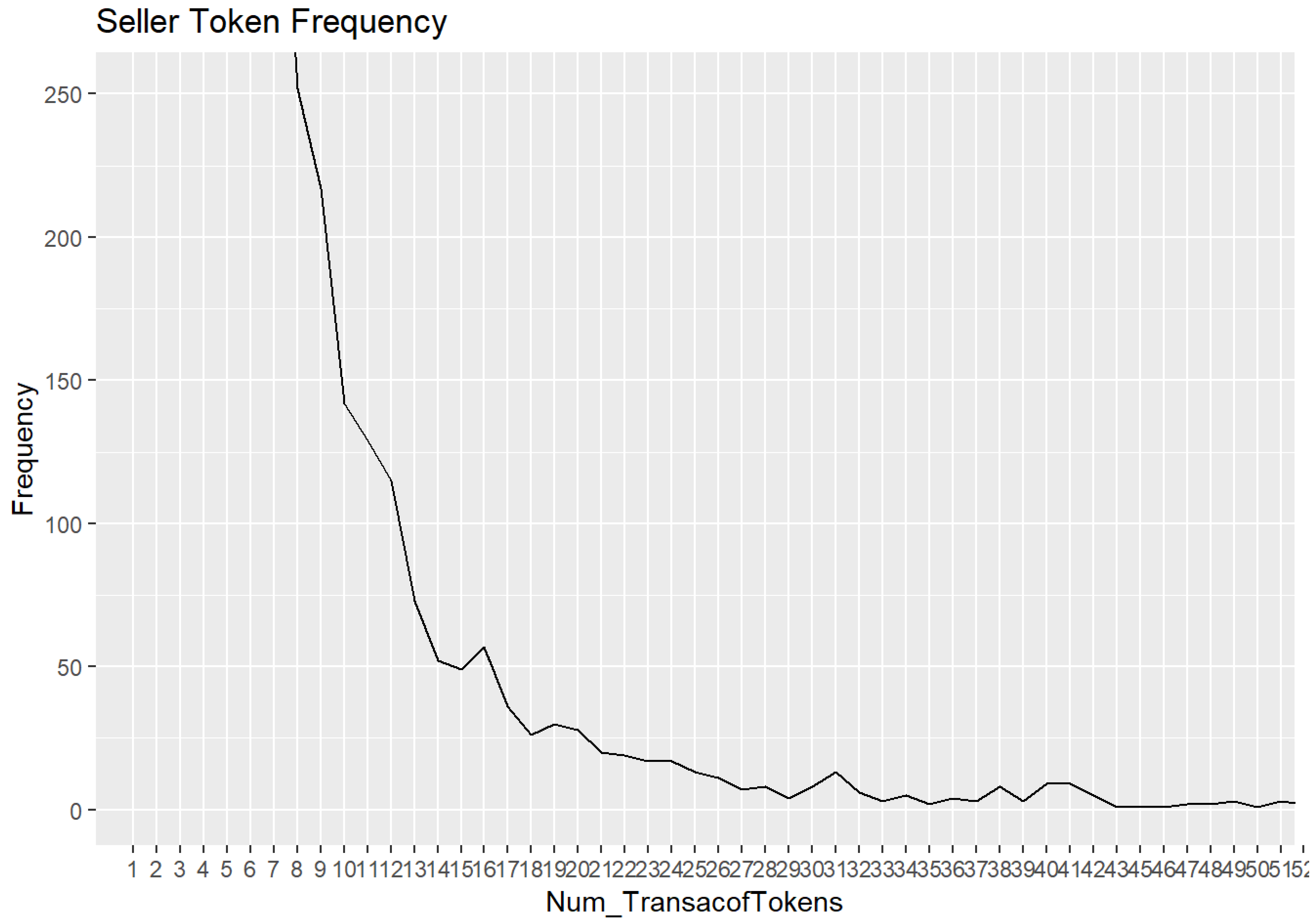
Count_freq_seller <fctr>	Freq <int>
1	73636
2	9207
3	3839

Count_freq_seller <fctr>	Freq <int>
4	1752
5	985
6	632
7	394
8	252
9	217
10	142

1-10 of 118 rows Previous **1** 2 3 4 5 6 ... 12 Next

A graph was plotted based on the data in the data2 table using the ggplot function.For this function we installed the ggplot2 library. Also a summary statistics for the data2 table is drawn wherein values for parameters like mean , median, maximum , minimum etc are found.

Code


Seller Token Frequency
Frequency
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
Num_TransacofTokens

Code

```
##  Count_freq_seller      Freq
##  1      :  1       Min.   :    1.0
##  2      :  1       1st Qu.:    1.0
##  3      :  1       Median :    2.0
##  4      :  1       Mean   :  779.3
##  5      :  1       3rd Qu.:    8.0
##  6      :  1       Max.   :73636.0
##  (Other):112
```

Based on the graph, we noticed the discrete distribution in which the data fits properly is exponential distribution.This means that lower frequency values are repeated more often and then it goes on decreasing exponentially.That is higher number of transaction of tokens are done by fewer sellers and transactions between 1 to 15 are done by more number of sellers in this dataset.

Distribution for Number of times user buys:

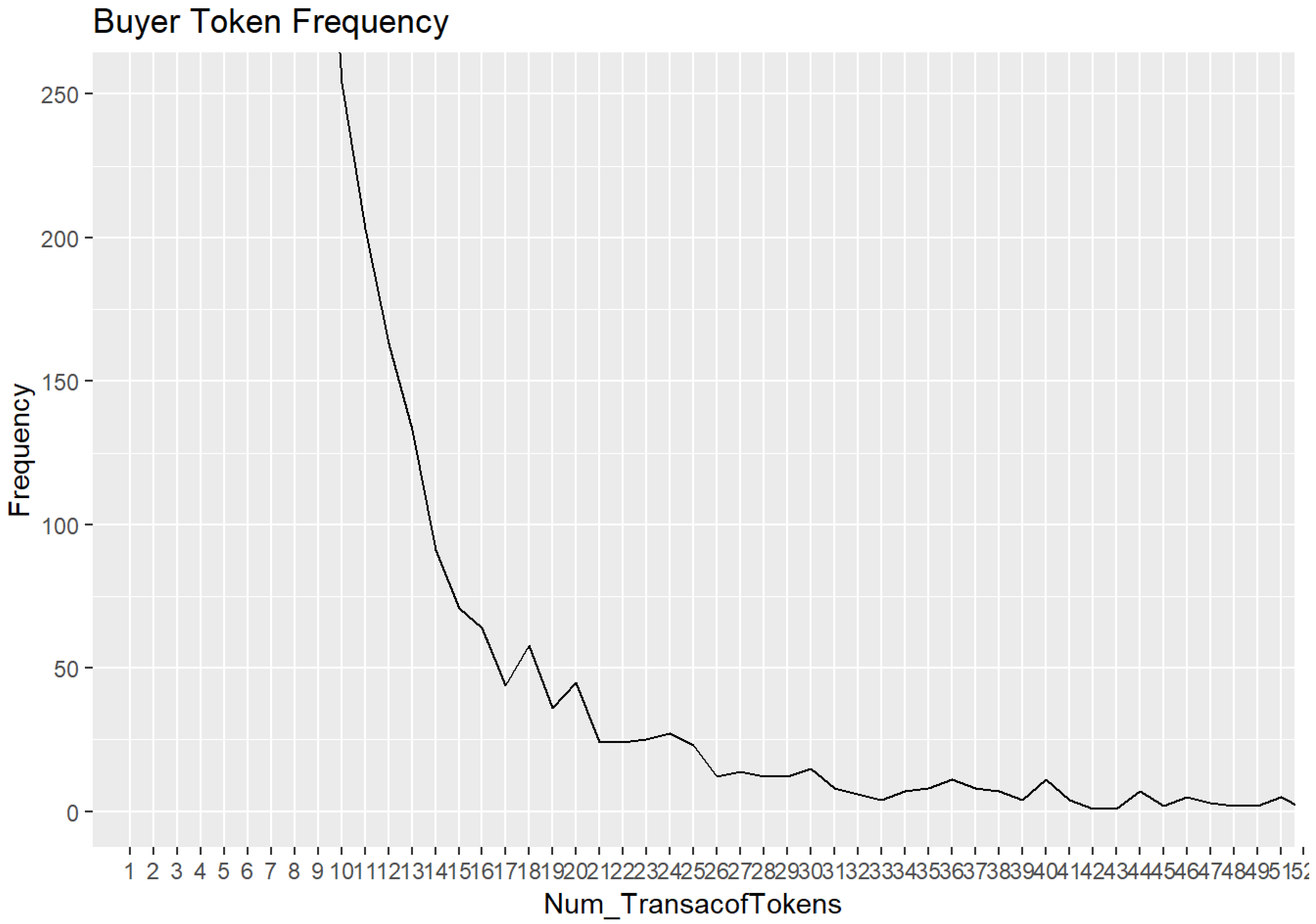
To find number of times a user buys, initially we considered the actual dataset without outliers and found the frequency of each user (UserIDs are present in the toNodeID column) in the table and stored the result in “data3”" table.The next step was to find the number of times each value in frequency column of data1 table is repeated. The displayed resultant table was stored in “data4”. The distribution is drawn on table data4 and summary statistics for the data4 is also displayed.

Code

Count_freq_buyer <fctr>	Freq <int>
1	107779
2	18798
3	6813
4	3302
5	1790
6	1175
7	723
8	543
9	390
10	254

1-10 of 123 rows Previous **1** 2 3 4 5 6 ... 13 Next

Code


Buyer Token Frequency
Frequency
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 101112131415161718192021222324252627282930313233343536373839404142434445464748495152
Num_TransacofTokens

Code

```
##  Count_freq_buyer      Freq
##  1       :  1      Min.   :     1
##  2       :  1      1st Qu.:     1
##  3       :  1      Median :     2
##  4       :  1      Mean   :  1162
##  5       :  1      3rd Qu.:    11
##  6       :  1      Max.   :107779
##  (Other):117
```

Based on plot, the discrete distribution in which the data fits properly is exponential distribution.This means that lower frequency values are repeated more often and then it goes on decreasing exponentially.That is higher number of transaction of tokens are done by fewer buyers and transactions between 1 to 15 are done by more number of buyers in this dataset.

Creating layers for number of transactions based on Token Amount and finding their correlation with the token price:

In this section we are first converting the UnixTime column in the remaining filtered data values to Year/Month/Date format.Next we are reading the new dataset of token prices of salt to find correlation between the two datasets.We then convert the date in the new prices table to the same format as earlier data set and then create a subset of the prices table consisting of only date and closing values columns.We are considering closing price for the analysis. Then we joined the two datasets prices and data to get the common rows ordered by Date.

Code

Date <date>	Open <dbl>	High <dbl>	Low <dbl>	Close <dbl>	Volume <fctr>	Market cap <fctr>
2018-07-15	1.010000	1.09	1.010000	1.040000	677,831	63,686,600
2018-07-14	1.030000	1.03	1.000000	1.010000	429,912	64,447,100
2018-07-13	1.000000	1.08	0.994052	1.030000	625,296	61,889,600
2018-07-12	1.020000	1.05	0.964983	0.987042	2,405,950	63,180,800
2018-07-11	1.010000	1.06	0.973503	1.030000	1,898,350	62,525,200
2018-07-10	1.090000	1.09	0.969154	1.010000	1,716,250	66,812,200
2018-07-09	1.160000	1.17	1.080000	1.090000	768,971	69,144,000
2018-07-08	1.200000	1.22	1.160000	1.160000	772,671	71,711,000
2018-07-07	1.180000	1.21	1.130000	1.210000	1,044,180	70,254,600
2018-07-06	1.170000	1.23	1.130000	1.180000	1,822,960	69,410,800

1-10 of 290 rows Previous **1** 2 3 4 5 6 ... 29 Next

Code

Code

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

To run the query using SQL in R , we installed and imported the sqldf library package. ####Layer 1 In layer 1, we consider the token Amount values between 1000 to 1.0 x 10^8. Thus all transactions that are between these values will be grouped together in layer 1.Next correlation is found between the number of transactions and Closing value ordered against time which satisfy the tokenAmount condition.

Code

```
## Pearson correlation for layer 1 and price data is : 0.684576594298768
```

Layer 2

In layer 2, we consider the token Amount values between 1.0 x 10^8 to 1.0 x 10^9. All transactions between these values will be grouped together in layer 2. Next correlation is found between the number of transactions and Closing value ordered against time which satisfy the tokenAmount condition.

Code

```
## Pearson correlation for layer 2 and price data is : 0.698439649715961
```

Layer 3

In layer 3, we consider the token Amount values between 1.0 x 10^9 to 1.0 x 10^10.All transactions between these values will be grouped together in layer 3.Next correlation is found between the number of transactions and Closing value ordered against time which satisfy the tokenAmount condition.

Code

```
##  Pearson correlation for layer 3 and price data is: 0.769529627076074
```

Layer 4

In layer 4, we consider the token Amount values between 1.0 x 10^10 to 1.0 x 10^12. All transactions between these values will be grouped together in layer 4. Next correlation is found between the number of transactions and Closing value ordered against time which satisfy the tokenAmount condition.

Code

```
## Pearson correlation for layer 4 and price data is: 0.658418076756544
```

Layer 5

In layer 5, we consider the token Amount values greater than 1.0 x 10^12. All transactions greater than this value will be grouped together in layer 5. Next correlation is found between the number of transactions and Closing value ordered against time which satisfy the tokenAmount condition.

Code

```
## Pearson correlation for layer 5 and price data is: 0.16222409321021
```

Thus observe from the correlation coefficients at each layer that correlation is best at layer 3 with correlation at 0.76.Layer 1,2,4 also show averagely good correlation at 0.65 and more.Thus if we observe we can see that total number of transactions in layer 1,2,3,4 are more when closing value of the token price is more and the number of transactions are less when the closing value is less. Since there are very less transactions per date in layer 5 , the correlation with the price data is very low.

Extracting 3 features from the dataset and drawing up a multiple linear regression model.

We have chosen total no of transactions, high token price value and low token price value as the three regressors.On the Y-axis we have considered the close value of the token price.We first merged the data from the salt dataset for a single layer with highest correlation along with the token prices data based on the “Date” column. Then we drew the regression model and displayed the coefficients and the residuals.

Code

	Prev_price_diff <dbl>	previous_close <dbl>	previous_marketcap <int>	price_return <dbl>
1	4.96	4.34	223	4.340000000
2	4.96	4.34	223	0.308755760
3	2.11	5.68	195	0.281690141
4	2.77	7.28	120	-0.103021978

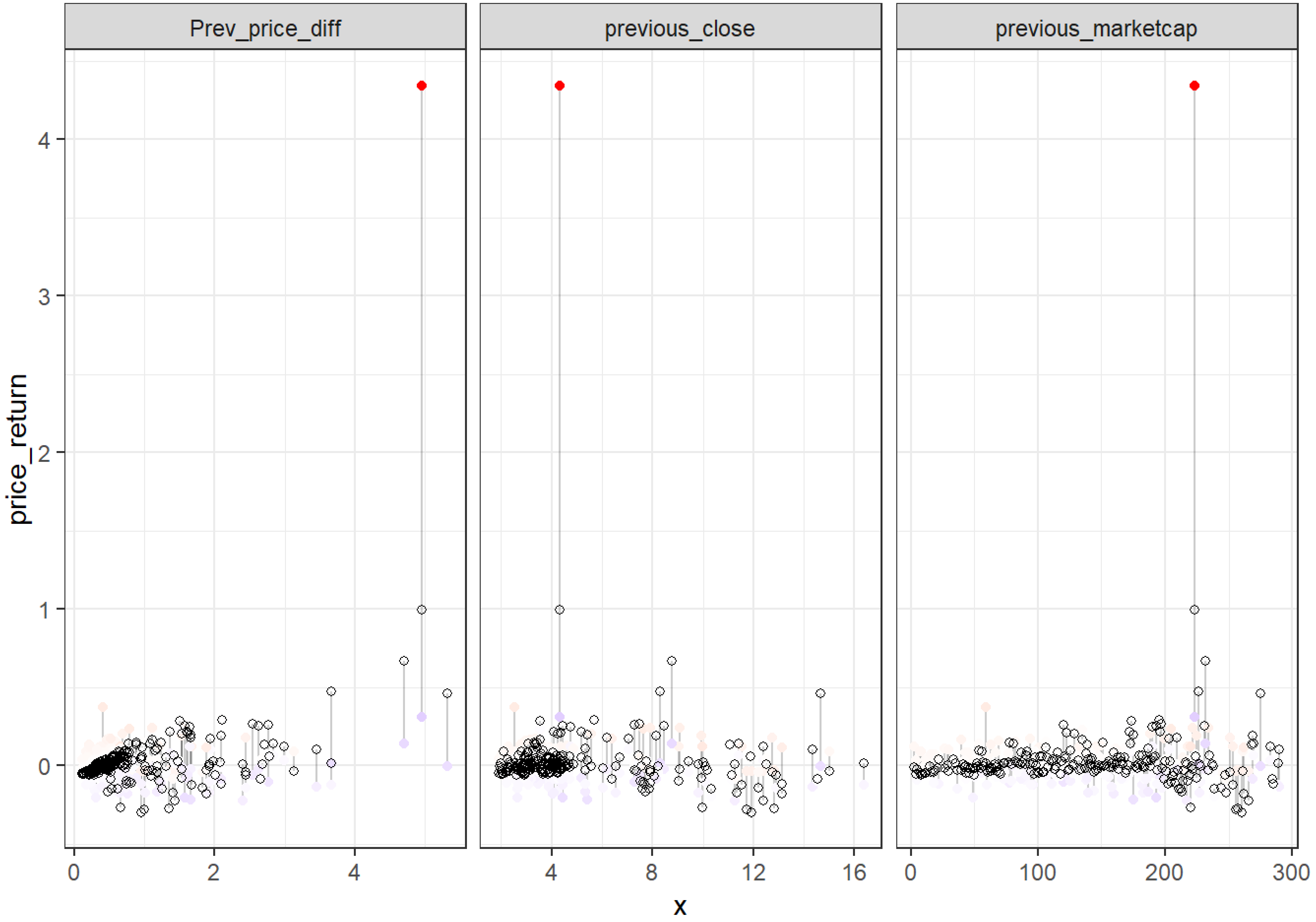
	Prev_price_diff	previous_close	previous_marketcap	price_return
	<dbl>	<dbl>	<int>	<dbl>
5	1.01	6.53	183	-0.171516080
6	1.66	5.41	175	-0.216266174
7	1.53	4.24	129	-0.011792453
8	0.51	4.19	162	-0.033412888
9	0.77	4.05	160	-0.177777778
10	0.89	3.33	151	0.042042042

1-10 of 220 rows Previous **1** 2 3 4 5 6 ... 22 Next

Code

```
##
## Call:
## lm(formula = price_return ~ Prev_price_diff + previous_close +
##     previous_marketcap, data = table_regression)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6906 -0.0915 -0.0085  0.0692  3.3407
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0407801  0.0399963   1.020    0.309
## Prev_price_diff     0.2100164  0.0288233   7.286  5.9e-12 ***
## previous_close     -0.0642552  0.0118581  -5.419  1.6e-07 ***
## previous_marketcap  0.0008778  0.0004733   1.855    0.065 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2754 on 216 degrees of freedom
## Multiple R-squared:  0.2214, Adjusted R-squared:  0.2106
## F-statistic: 20.47 on 3 and 216 DF,  p-value: 1.027e-11
```

Code


Prev_price_diff
previous_close
previous_marketcap
price_return
4
3
2
1
0
0
2
4
4
8
12
16
0
100
200
300
x

In the above plot, black points are the actual values and White points are predicted values.With this in mind, we can see, as expected, that there is less variability in the predicted values than the actual values.

Printing the coefficients : B0, B1, B2, B3

Code

```
## B0 (y intercept price-return) is: 0.040780143375459
```

Code

```
## B1 (Prev_price_diff) is: 0.210016436998744
```

Code

```
## B2 (previous_close) is: -0.0642551911459204
```

Code

```
## B3 (previous_marketcap) is: 0.000877801682987224
```

Conclusion on the project performed:

Based on the dataset analysis, we found that number of transactions of sellers and buyers is exponentially distributed. The correlation coefficient is higher in layer3 (1.0 x 10^9 < TokenAmount <= 1.0 x 10^10) ie 0.76, which means the number of transactions in this layer is highly correlated with the price when compared to the other layers.Since there are

very less transactions per date in layer 5 (TokenAmount >1.0 x 10^12), the correlation with the price data is very low.
Regression Model: Our Regression model equation is Y=0.04+0.21(x1)-0.06(x2)+0.001(x3)
Residual standard error: 0.2754, is the deviation of observed value from the actual value.Ideal prediction model should have a RSE of zero.
We are using R squared value to explain the adequacy of our regression model.This value describes the proportion of variation in the response data that is explained by the regression model. When all the points in a data set lie on the regression model, the largest possible value of r2= 1 is obtained, while a minimum possible value of r2=0 is obtained when there is only one data point or if the straight line regression model is a constant line.
We have obtained a R squared value of 0.22

References:

1. https://blockgeeks.com/guides/what-is-ethereum/ (https://blockgeeks.com/guides/what-is-ethereum/)
2. https://www.investopedia.com/articles/investing/022516/what-ethereum.asp (https://www.investopedia.com/articles/investing/022516/what-ethereum.asp)
3. https://cryptobriefing.com/what-is-a-token-in-cryptocurrency/ (https://cryptobriefing.com/what-is-a-token-in-cryptocurrency/)
4. https://cointelegraph.com/explained/erc-20-tokens-explained (https://cointelegraph.com/explained/erc-20-tokens-explained)
5. https://www.investinblockchain.com/top-ethereum-tokens/ (https://www.investinblockchain.com/top-ethereum-tokens/)
6. https://saltlending.com/ (https://saltlending.com/)
7. https://coincentral.com/salt-lending-beginner-guide/ (https://coincentral.com/salt-lending-beginner-guide/)