

# Media Hype and Cryptocurrencies Through the Investigation of Reddit Data

Sean Maloney\*      Sophia Margareta Carayannopoulos†

May 9, 2018

## Abstract

Investigating posts and submissions to the subreddit /r/CryptoCurrency/ we build a classification model for cryptocurrency price moments. Out of the four models we fit none seemed to have very promising results. It is likely there are no trends present within the subreddit /r/CryptoCurrency data or they have already been exploited.

---

\*Department of Statistics. Email: [maloney4@wisc.edu](mailto:maloney4@wisc.edu)

†Department of Statistics. Email: [carayannopou@wisc.edu](mailto:carayannopou@wisc.edu)

# 1 Introduction

Recently, cryptocurrencies have started to appear all over social and mainstream media. While the technology behind them is interesting what interests most people is the potential money. With bitcoin increasing over 10x in value during 2017 and the entire cryptocurrency ecosystem being measured in the hundreds of billions malicious actors might be trying to game the system through manipulated social media for their own financial gain.

One of the main platforms used for cryptocurrency discussions is Reddit. The subreddit /r/CryptoCurrency/ has over 600k worldwide members and is specifically committed to the discussion and distribution of information about cryptocurrencies.

Our goal to use the submissions and comments within /r/CryptoCurrency/ as metric of media hype to see if we could predict a cryptocurrency day-to-day performance.

## 2 Gathering Data

Our data set from r/CryptoCurrency/ was collected between January 1, 2017 and December 31, 2017 and included all posts and comments for the time period. The dataset in total has over 5 million comments, 132k posts and 22k active users, active users being all unique accounts that posted or commented. Our investigation will consider the top 10 coins (excluding bitcoin) by market cap on January 1, 2017 which from largest to smallest are as follows:

Ethereum, Ripple, Litecoin, Monero, Ethereum Classic, Dash, MaidSafe-Coin, Augar, Steem, NEM.

Bitcoin is purposely excluded in our investigation as it is used as a baseline price unit. Each of the top 10 coin's prices will be considered in bitcoin value. Pricing in terms of bitcoin rather than USD is important as all of the coins we consider are traded directly to bitcoin rather than USD pairs.

## 3 The Data Set

Our data set contained 26 predictor variables computed from the raw post and comment text. Each variable is computed over a single day. Each statistic for a coin has a corresponding control statistic for the day. This is important as the total number of posts and comments increased significantly throughout the year, see Figure 1.

The control prediction and coin prediction variables are listed and described (when unclear) below:

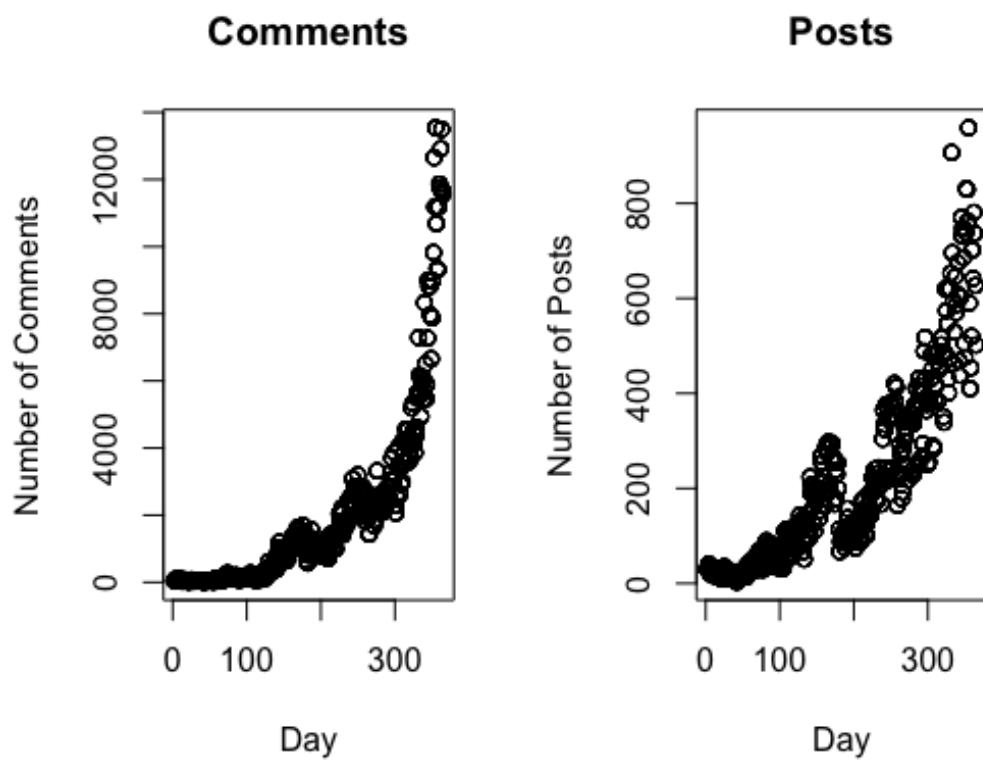


Figure 1: Comment and Post Trends

### 3.1 Control Variables

All statistics are computed for over a single day for all posts and comments.

1. Total Number Comments
2. Total Mean Comment Length  
Comment raw text length in characters
3. Total Comment Different Authors  
Total unique authors for both comments and posts based on username
4. Total Mean Comment Score  
The mean comment score corresponds to the mean of upvotes and downvotes where each upvotes and downvotes have equal weight
5. Total Comment Number Fomo  
Number of times “Fomo” or “fear of missing out” appears as a literal in raw comment text
6. Total Comment Number Fud  
Number of times “FUD” or “fear, uncertainty and doubt” appears as a literal in raw comment text
7. Total Number Posts
8. Total Post Mean Score  
The mean post score corresponds to the mean of upvotes and downvotes where each upvotes and downvotes have equal weight
9. Total Posts Mean Length  
Mean text length of all posts in characters
10. Total Posts Number Different Authors  
Number of unique authors over all posts
11. Total Posts Mean Number Of Comments  
Mean number of comments posts receive
12. Total Posts Number Fomo  
Number of times “Fomo” or “fear of missing out” appears as a literal in raw posts text
13. Total Posts Number Fud  
Number of times “FUD” or “fear, uncertainty and doubt” appears as a literal in raw posts text

### 3.2 Coin Specific Variables

For the coin specific variables the post or comment was determined to be “for” the coin if they contained the coins name or the coins trading abbreviation in their raw text. All statistics are computed for each coin over a single day in the same fashion are the control variables. They are listed below and follow the same descriptions as the control variables above:

- |   |   |
|---|---|
| 1. Total Number Coin Comments                   | 8. Total Coin Mean Submissions Score                |
| 2. Total Mean Coin Comment Length               | 9. Total Coin Mean Submissions Length               |
| 3. Total Number Coin Comments Different Authors | 10. Total Coin Number Submissions Different Authors |
| 4. Total Mean Coin Comment Score                | 11. Total Coin Mean Submissions Number Of Comments  |
| 5. Total Coin Comment Number Fomo               | 12. Total Coin Submissions Number Fomo              |
| 6. Total Coin Comment Number Fud                | 13. Total Coin Submissions Number Fud               |
| 7. Total Coin Number Submissions                |   |

## 4 Preliminary Look

Our first step was to look at the bitcoin prices of each coin over the year. We saw that there were three possibly distinct price groups, see Figure 2. Upon further inspection, it seemed that that one distinct price group was: Augur, Steem, Ethereum, Ethereum Classic, which we refer to as Group 1, which can be seen in Figure 3a. They each have a very easy to follow distribution, unimodal, and bell shaped. It seemed that those in Group 1 had a peak between days 100 and 200. Upon further inspection it seemed that they also shared a peak around days 160 to 180.

The next distinct price group, Group 2, seemed to be: Dash, Litecoin, Monero, Maidsafe. These coins did not have an easy to follow curve like coins from Group 1. They seemed multimodal, but they also seemed to share a specific peak between 200 and 300 as seen in Figure 3b.

There is finally Group 3, though they were not as complicated to follow as the second group, their trend was not like Group 1 and deserved to be a separate and distinct group. It is possible that this group is bimodal but they share a peak between days 100-200. When we looked closer we saw that this

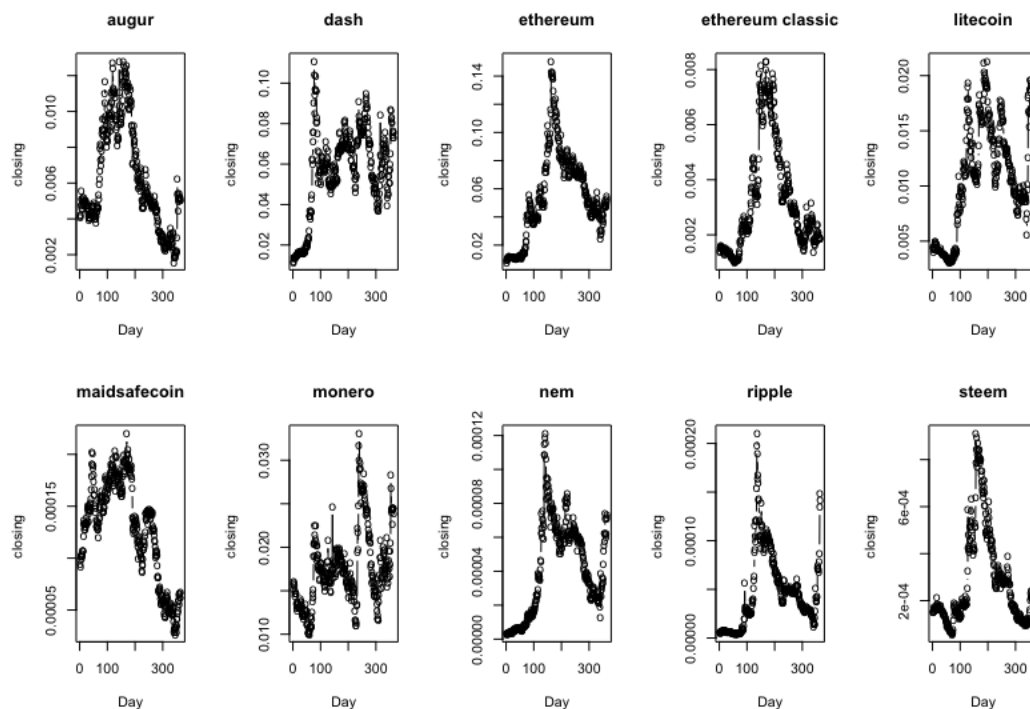


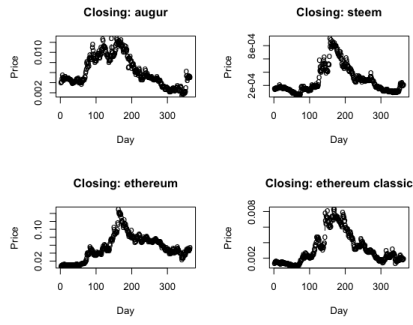
Figure 2: Trend of Each Coin

peak was between 120-150, leading us to further believe that this is a distinct group, Figure 3c.

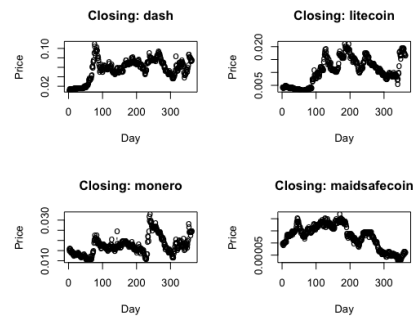
We could not discern why these coins fell into distinct groups based on the merits of the coins. For example Ethereum and Steem are in the same group, yet Ethereum is one of the top traded coins, while Steem is more of a niche coin and is not widely used. There may be a confounding variable that would be able to help more fully explain the similar trends. It is also possible that they are just random.

## 5 Statistical Analysis

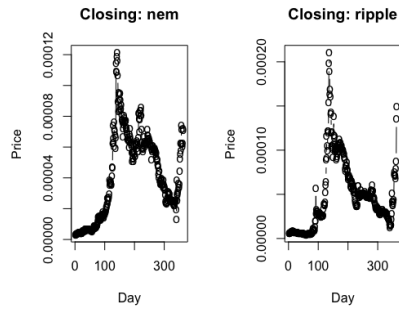
When we set out to do this project we knew we wanted to do some kind of prediction of the value of the coin. We knew that with all of this information/many variables we would be unable to do a straightforward linear regression of any sort. Our first step had to be making all the variables more manageable. The next step was looking at the odds of whether or not the coin would rise in value. Finally we wanted to see how robust our model was. We felt that it would appropriate to build four models. One model would be using all the coins and the other three would be for each distinct group of coins. We wanted to see if separating them into groups would greatly improve the model.



(a) Group 1 Trends



(b) Group 2 Trends



(c) Group 3 Trends

Figure 3: Trends By Groups

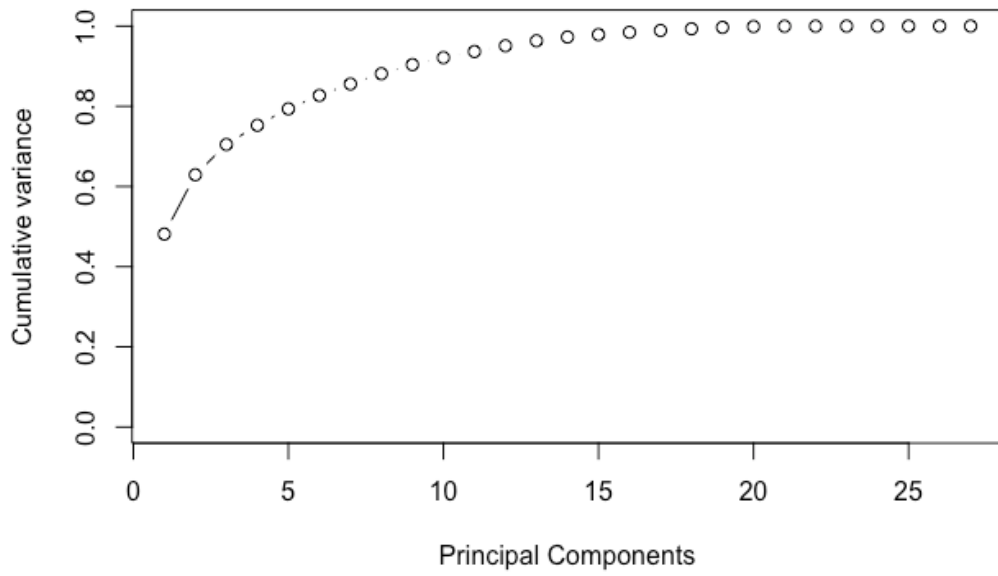


Figure 4: Measuring Variation Explained

Another thing we were looking for was to see if a certain group was making the model perform well while the other one or two were performing poorly.

As mentioned above, we have a lot of variables to be taken into account. The variables could also be correlated. It seemed that the most appropriate step would be to perform Principal Component Analysis, PCA. This technique is used for data sets like ours where we need to take the variables into a lower dimension. There are certain assumptions that need to be met before performing PCA. We are assuming that the variables have a linear relationship since PCA will be unable to detect nonlinear subspaces. An issue with that is we are not sure this holds for all of our variables, but we will hold the assumption that they are. Another assumption that is made before performing PCA is that we have a large and representative sample. There is no reason to believe that our dataset is not representative, let alone comprehensive. Once we performed PCA on all the variables except Coin, Close, High, Low, Open, we saw that to explain 80% of the variation we need 5 variables, Figure 4. On closer inspection of the components, we could not find any of the variables more significantly weighted than others. This is another drawback of PCA, when compacting the data into a lower dimension you lose some interpretation and understanding of the predictors.

Our next step was to use components to fit a logistic regression model. We thought it would be interesting to see if we would be able to predict if a coin went up or not. This made our logistic linear regression model a binomial one. One of the assumptions that could possibly be problematic is that we assume that all the variables are independent. Though we used PCA to deal with collinearity, it does not take away that a comment might influence an up vote and so on and so forth. We are working on the assumption that everyone can think for himself or herself and is not influenced by others. There is the assumption, once again, that we need a large data set: we have met this. Due to using the components in our logistic regression, we did not feel it was appropriate to include a formal equation. For those who would like to see the model with the components fitted, the R code is included in the appendix

After fitting the full model and the group models, we wanted to create confidence intervals for how often we saw the price going up, the price going down, type I error and type II error. We decided that nonparametric bootstrap would be appropriate since we do not know that distribution of the population we are sampling from. For the assumption of bias, we might run into an issue because we are testing the model with the same data we used to fit the model. Another assumption that is made is that the data set is large enough: there is no reason to believe that the data set is not large enough. When we created our confidence intervals for the above we found that in general our models did not perform very well and had high type II errors, Tables 2. It seemed that Group 1 is the only group that performs decently but even then, it would not be enough for someone to exploiting Reddit for gain.



There are also negative values in our confidence intervals. As we know it is impossible to get a negative amount correct or incorrect, but this shows how much variance is in our model.

## 6 Conclusion

Overall, our models did not perform well. From the variables that we selected it does not seem that anyone can predict if a coin is going up or down in value. From our confidence intervals, we saw that our model is extremely variant and is therefore not good for prediction. We also saw that our type II error is quite high for most of the models. Groups 2 and Groups 3 were poorly fitted while Group 1 performed significantly better. Group 1 is also what contributed to the full model performing better. The model for Group 1 is still neither accurate nor precise enough to be considered a model that could be or could have been used for exploitation.

## 7 Next Steps

One of the issues in our model fits might be that we were not considering appropriate variables to measure media hype. Looking at the karma of users could add helpful insights to the legitimacy of a coin. Another thing we can inspect more closely in the future are the coins in Group 1. Finally, we might change the focus of the next analysis. Working off the idea that we should be looking for other factors, we could consider looking for specific accounts that might be purposefully trying to influence the value of a coin.

Labels	
Correct: Went Up	Type I Error
Type II Error	Correct: Did Not Go Up

Table 1: Labels for Tables

Full Model		Group 1	
0.047	.042	0.028	0.022
0.418	0.490	0.412	0.535
Group 2		Group 3	
0.102	0.095	0.187	0.148
0.367	0.434	0.151	0.184

Table 2: Estimates

Full Model		Group 1	
(-0.050, 0.145)	(-.046, 0.132)	(-0.026,0.083)	(-0.021,0.066)
(.334, 0.503)	(.386, 0.595)	(0.370,0.455)	(0.471, 0.599)
Group 2		Group 3	
(-0.060,0.266)	(-0.048,0.239)	(0.103,0.478)	(0.068, 0.365)
(0.223,0.51)	(0.268,0.600)	(0.065, 0.368)	(0.089, 0.459)

Table 3: Estimates

## Reference

Frisvad, Jens C. Principal Component Analysis.  
[www2.imm.dtu.dk/courses/27411/doc/Lecture03/PCA2013.pdf](http://www2.imm.dtu.dk/courses/27411/doc/Lecture03/PCA2013.pdf).  
“Assumptions of Logistic Regression: Statistics Solutions”,  
[www.statisticssolutions.com/assumptions-of-logistic-regression/](http://www.statisticssolutions.com/assumptions-of-logistic-regression/).

Brems, Matt. “A One-Stop Shop for Principal Component Analysis:  
Towards Data Science.”  
Towards Data Science, Towards Data Science, 17 Apr. 2017,  
[towardsdatascience.com/a-one-stop-shop-for  
-principal-component-analysis-5582fb7e0a9c](https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c)

## Appendix: R code

The R code used for analysis is included below. Python and other R code used for data gathering and cleaning can be found at: <https://github.com/Soph-PhiaC/Stats471>

```
---
title: "Project 471"
author: "Sophia Margareta Carayannopoulos"
date: "4/24/2018"
output:
  pdf_document: default
  html_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
rm(list=ls())
all_coin_data = read.csv("all_coin_data.csv")

colnames(all_coin_data )
TOP_COINS = c(c("eth", "ethereum"), c("xrp","ripple"),c("ltc", "litecoin"),
c("xmr", "monero"),c("etc", "ethereum classic"),c
("dash", "dash"),c("maid", "maidsafecoin"),
c("rep", "augur"), c("steem", "steem"), c("xem", "nem"))
```

#Playing With Data

###All Levels:
```{r, echo=FALSE}

coins = levels(all_coin_data$Coin)

```

###Closing worth each day for a year.
```{r, AllCoinsSeries}
par(mfrow=c(2,5))
for (i in coins) {
  plot(y= all_coin_data$Close[which(as.character(all_coin_data$Coin) == i)],
    x = all_coin_data$Day.1[which(as.character(all_coin_data$Coin) == i)],
    main = i, type = 'b', pch = 'o', xlab="Day", ylab="closing ")
}

```

Group 1: Steem, augur, ethereum, ethereum classic: have similar shapes.
They have this hump between 100 and 250

Group 2: Dash, litecoin, monero, maidsafecoin: have a lot more up and down
```

Group 3: nem and ripple: also have super similar shapes, having a peak between 100 and 200

```
##Function for graphing groups
'''{r, basicFunctions, echo=FALSE}
Setting_Window<- function(x){
  if( x == 1){
    par(mfrow=c(1,1))
  } else if(x == 2){
    par(mfrow=c(1,2))
  } else if( x == 3){
    par(mfrow=c(1,3))
  }else{
    par(mfrow=c(ceiling(x/2),ceiling(x/2)))
  }
}

plot_dot_closing <- function(groups, range_values){
  Setting_Window(length(groups))

  for (i in groups) {
    plot(y = all_coin_data$Close[which(all_coin_data$Coin == i & all_coin_data$Day.1
    >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
        x = all_coin_data$Day.1[which(all_coin_data$Coin == i & all_coin_data$Day.1
        >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
        main = paste("Closing:", i), type = 'b', pch = 'o', xlab= "day", ylab = "Quantity")
  }
}

plot_dot_Total_Coin_Number_Comments <- function(groups, range_values){

  Setting_Window(length(groups))

  for (i in groups) {
    plot(y= all_coin_data$Total_Number_Coin_Comments[which(all_coin_data$Coin == i
    & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1 <
    range_values[2])],
        x = all_coin_data$Day.1[which(all_coin_data$Coin == i & all_coin_data$Day.1
        >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
        main = paste("Total Number Coin Comments", i),
        type = 'b', pch = 'o', xlab= "day", ylab = "Quantity")
  }

}

plot_dot_Total_Coin_Number_Submissions_FOMO <- function(groups, range_values){
  Setting_Window(length(groups))
  for (i in groups) {
    plot(y= all_coin_data$Total_Coin_Submissions_Num_Fomo[which(all_coin_data$Coin == i
    & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1
```

```

        < range_values[2]]],
        x = all_coin_data$Day.1[which(all_coin_data$Coin == i & all_coin_data$Day.1
        >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
        main = paste("Total Number Coin Submission FOMO", i),
        type = 'b', pch = 'o', xlab= "day", ylab = "Quantity")
    }
}

plot_dot_Total_Coin_Number_Submissions_FUD <- function(groups, range_values){
    Setting_Window(length(groups))

    for (i in groups) {
        plot(y= all_coin_data$Total_Coin_Submissions_Num_Fud[which(all_coin_data$Coin == i
        & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1
        < range_values[2])],
        x = all_coin_data$Day.1[which(all_coin_data$Coin == i & all_coin_data$Day.1
        >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
        main = paste("Total Number Coin Submissions FUD", i), type = 'b',
        pch = 'o', xlab= "day", ylab = "Quantity")
    }
}

plot_dot_Total_Coin_Number_Comments_FOMO <- function(groups, range_values){
    Setting_Window(length(groups))
    for(i in groups){
        plot( y = all_coin_data$Total_Coin_Comment_Num_Fomo[which(all_coin_data$Coin == i
        & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1
        < range_values[2])],
            x = all_coin_data$Day.1[which(all_coin_data$Coin == i
            & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1
            < range_values[2])],
            main = paste("Quantity Comments FOMO", i), type = 'b',
            pch = 'o', xlab= "day", ylab = "Quantity")
    }
}

plot_dot_Total_Coin_Number_Comments_FUD <- function(groups, range_values){
    Setting_Window(length(groups))

    for(i in groups){
        plot( y = all_coin_data$Total_Coin_Comment_Num_Fomo[which(all_coin_data$Coin == i
        & all_coin_data$Day.1 >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
            x = all_coin_data$Day.1[which(all_coin_data$Coin == i & all_coin_data$Day.1
            >= range_values[1] & all_coin_data$Day.1 < range_values[2])],
            main = paste("Quantity Comments FUD", i), type = 'b',
            pch = 'o', xlab= "day", ylab = "Quantity")
    }
}

'''

```

```

###Inspecting humps in first group
```{r, LookingAtG1}
Group_1 <- c(coins[1], coins[10], coins[3], coins[4])
plot_dot_closing(Group_1, c(100, 250))
```

```

Seems that there is peaks in between 150 days and 200

```

```{r, continue_G1}
plot_dot_closing(Group_1, c(150, 200))
```

```

Looking at these graphs something is definitely going on between days 160 and 180

```

```{r, continue2_G1}
plot_dot_closing(Group_1, c(160, 175))
```

```

```

###Inspceting humps in the second group
```{r, LookingG2}
Group_2 <- c(coins[2], coins[5], coins[7], coins[6])
plot_dot_closing(Group_2, c(200, 300))
```

```

Definitely something between 220 and 280 for these coins.  
interesting that this is at completely different time then group 1.

```

```{r, continue_G2}
plot_dot_closing(Group_2, c(220,280))
```

```

okay looking at these closer it seems like we should inspect between 230 and 250

```

```{r, continue2_G2}
plot_dot_closing(Group_2, c(230, 250))
```

```

From here it seems that there is something going on with day 240ish, there seems to be a change or something happening in particular

```

###Looking through group 3
```{r, LookingG3}
Group_3 = c()
for( i in coins){
  if(!(i %in% Group_1) & !(i %in% Group_2)){
    Group_3 = c(Group_3, i)
  }
}

```

```

plot_dot_closing(Group_3, c(1, 365))
par(mfrow=c(2,1))
plot_dot_closing(Group_3, c(100, 200))
plot_dot_closing(Group_3, c(120, 180))
plot_dot_closing(Group_3, c(120, 150))
```

```

So once again we have found a special hump.

## ##SUMMARY AFTER LOOKING AT 3 GROUPS

They each have their own particular days where closing coin value spikes up. None of them overlap or by very little if at all. Things to look into from here. Is there any expertise as to why these coins are grouped like this (meaning why do they have similar distributions) Also did anything else peak during that time? Anything else change out of the ordinary?

###Looking At Comments in General.

```
```{r, Comments}
mean(all_coin_data$Total_Number_Comments)
median(all_coin_data$Total_Number_Comments)
hist(all_coin_data$Total_Number_Comments )

par(mfrow=c(1,2))
plot(x = all_coin_data$Day.1, y = all_coin_data$Total_Number_Comments,
xlab = "Day", ylab = "Number of Comments")
plot(x = all_coin_data$Day.1, y = all_coin_data$Total_Number_Submissions,
xlab = "Day", ylab = "Number of Posts")
```
```

We see that in general that the comments have an upward trend in general. However we are able to see two bumps, one between 100-200 and the other is between 200 -300. These could possibly line up with the bumps we saw in the above 3 groups. We will explore each of these groups individually.

#100-200 Hump

```
```{r, LookingHump1}
par(mfrow=c(2,2))
plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 100 & all_coin_data$Day.1 < 200)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1
>= 100 & all_coin_data$Day.1 < 200)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 140 & all_coin_data$Day.1 < 200)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1
>= 140 & all_coin_data$Day.1 < 200)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 160 & all_coin_data$Day.1 < 200)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1
>= 160 & all_coin_data$Day.1 < 200)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 160 & all_coin_data$Day.1
< 180)], y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1
>= 160 & all_coin_data$Day.1 < 180)], ylab= "comments", xlab = "day")
```
```

Once we get to the 160-180 it is harder to see that hump, but before that you can definitely see the hump.

Makes me believe something happened in those days.

So we will inspect 160-180.

This lines up with Group\_1. So let's look as to what is going on.



```

#What type of comments
```{r, typesCommentsHump1, echo=FALSE}
par(mfrow=c(2,1))
plot(all_coin_data$Day.1,
all_coin_data$Total_Number_Comments, ylab= "comments",
xlab = "day", main = "Total Number Comments")
plot(all_coin_data$Day.1[which(all_coin_data$Day.1
>= 160 & all_coin_data$Day.1 < 180)], all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1
>= 160 & all_coin_data$Day.1 < 180)],
ylab= "comments", xlab = "day",
main = "Total Number Comments 160-180")

plot_dot_Total_Coin_Number_Comments(Group_1, c(100,200))
```

Oh you can totally see that the comments for that time in all those coins.
So maybe # comments means something

Let's start looking at comments more specifically.
```{r, LookingCommentsSpecifics, echo=FALSE}

plot(x = all_coin_data$Day.1,
y=all_coin_data$Total_Coin_Comment_Num_Fomo,
ylab= "Day", xlab="Quantity", main= "FOMO comments all coins")
cat("So Fomo looks like is increasing exponetially. ")

plot_dot_Total_Coin_Number_Comments(Group_1, c(1,365))
plot_dot_Total_Coin_Number_Comments_FOMO(Group_1, c(1,365))
plot_dot_Total_Coin_Number_Comments_FUD(Group_1, c(1,365))

plot_dot_Total_Coin_Number_Comments_FOMO(Group_1, c(160,180))
plot_dot_Total_Coin_Number_Comments_FUD(Group_1, c(160, 180))
cat("now let's expan the range a bit to see if
anything suspicious happen days before after")

plot_dot_Total_Coin_Number_Comments_FOMO(Group_1, c(140, 200))
plot_dot_Total_Coin_Number_Comments_FUD(Group_1, c(140, 200))

all_coin_data[which(all_coin_data$Day.1 < 180
& all_coin_data$Day.1
> 150 & all_coin_data$Coin == "steem"
& all_coin_data$Total_Coin_Comment_Num_Fomo == 1),"Day.1"]

all_coin_data[which(all_coin_data$Day.1 < 180
& all_coin_data$Day.1
> 150 & all_coin_data$Coin == "steem"
& all_coin_data$Total_Coin_Comment_Num_Fud == 1),"Day.1"]

```

We see the humps of # comments also peaks at the same time their price peaks.
Possible peaks at the 160-180 mark

```

Looking at Submissions in general and specifics

```
```{r, look}
plot_dot_Total_Coin_Number_Submissions_FOMO(Group_1, c(1,365))
plot_dot_Total_Coin_Number_Submissions_FUD(Group_1, c(1, 365))

plot_dot_Total_Coin_Number_Submissions_FOMO(Group_1, c(140, 200))
plot_dot_Total_Coin_Number_Submissions_FUD(Group_1, c(140, 200))

plot_dot_Total_Coin_Number_Submissions_FOMO(Group_1, c(160, 180))
plot_dot_Total_Coin_Number_Submissions_FUD(Group_1, c(160, 180))

mean(all_coin_data$Total_Coin_Number_Submissions[which(all_coin_data$Day.1
<= 200 & all_coin_data$Day.1 >= 180 & all_coin_data$Coin == "steem" )])
```
```

Definitely looking like those humps mean something.

Fud Submissions seem to be about the coin seem to be as it starts to fall.

#200-300 Hump

```
```{r, LookingHump2, echo=FALSE}
par(mfrow=c(2,2))
plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 200
& all_coin_data$Day.1 < 300)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1 >= 200
& all_coin_data$Day.1 < 300)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 220
& all_coin_data$Day.1 < 280)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1 >= 220
& all_coin_data$Day.1 < 280)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 230
& all_coin_data$Day.1 < 270)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1 >= 230
& all_coin_data$Day.1 < 270)], ylab= "comments", xlab = "day")

plot(x = all_coin_data$Day.1[which(all_coin_data$Day.1 >= 240 & all_coin_data$Day.1 < 260)],
y = all_coin_data$Total_Number_Comments[which(all_coin_data$Day.1 >= 240
& all_coin_data$Day.1 < 260)], ylab= "comments", xlab = "day")
```
```

Once again, you see that the 20 day one you don't see that hump,

but as soon as you widen the scope you can see that this is a hump.

After looking

```
```{r, util}
##Helps to build the binary responses
build_response = function(all_coin_data, coins){
  row_data_response = c()
  for(i in 3:365){
```

```

for(j in seq(2, length(coins), 2)){
  day_curr = which(all_coin_data$Day.1 == i & all_coin_data$Coin == coins[j])
  day_before = which(all_coin_data$Day.1 == (i-1) & all_coin_data$Coin == TOP_COINS[j])
  if((all_coin_data$Close[day_curr]-all_coin_data$Close[day_before]) > 0){
    #print("Went up")
    row_data_response = c(row_data_response, 1)
  }else{
    #print("Went down")
    row_data_response = c(row_data_response, 0)
  }
}
}
row_data_response = c(rep(0, length(coins)/2),row_data_response)
return(row_data_response)
}

all_coin_data = cbind(build_response(all_coin_data, TOP_COINS), all_coin_data)
colnames(all_coin_data)[1] = "Response"

compute_errors = function(truth, guess){
  output = c()
  for(i in 1:length(guess)){
    if(guess[i] == 1 && truth[i] == 1){
      output = c(output, 1)
    }

    if(truth[i] == 1 && guess[i] == 0){
      output = c(output, 2)
    }

    if(truth[i] == 0 && guess[i] == 1){
      output = c(output, 3)
    }

    if(truth[i] == 0 && guess[i] == 0){
      output = c(output, 4)
    }
  }
  return(output)
}

'''

##Fitting A Model
'''{r, FittingPCR}

build_all_model = function(all_coin_data, verbose=FALSE){
  ##Fill NA with zero
  all_coin_data[is.na(all_coin_data)] <- 0

```

```

##Exclude the price data
data_refined = all_coin_data[,-c(1,2,c(5:8))]
response = all_coin_data[,1]

##Exclude the coin factor
Sigma = cor(data_refined[,-1])

evec = eigen(Sigma)$vectors
eval = eigen(Sigma)$values

##Exclude the coin factor
pr.out = prcomp(data_refined[,-1], scale=TRUE)
pve = eval/sum(eval)
yvec = cumsum(pve)

if(verbose){
  ##Find 80% cut off
  plot(yvec, type='b', xlab = "Principal Components",
        ylab = "Cumulative variance", main = "",ylim=c(0,1))
}

##Lets take first 5 about 80%

##Compute the data value for principle component #1 at each data point, exclude coin
pca_var = data.matrix(data_refined[,-1])%*%matrix(pr.out$rotation[,1:5], nrow=27, ncol=5)

model = glm(response~pca_var)

if(verbose){
  summary(model)
}

##
outputs = model$coefficients[1]
+ model$coefficients[2]*pca_var[,1]
+ model$coefficients[3]*pca_var[,2]
+ model$coefficients[4]*pca_var[,3]
+ model$coefficients[5]*pca_var[,4]
+ model$coefficients[6]*pca_var[,5]

as_binary = as.integer(outputs > .5)

if(verbose){
  cat("Accuracy:", mean(as_binary == response))
  plot(as_binary)
}
return(c(list(response), list(as_binary), list(model)))
}

full_model = build_all_model(all_coin_data)

plot(all_coin_data$Total_Number_Comments[1:364],

```

```

main="Total Number Comments", xlab="Day", ylab="Total Comments", type='l')
'''

##Fitting A Model - Groups
'''{r, fitting_model_groups}

colnames(all_coin_data)

build_group_model = function(Group, all_coin_data, verbose=FALSE){
  ##Fill NA with zero
  all_coin_data[is.na(all_coin_data)] = 0

  ##Exclude the price data
  data_refined = all_coin_data[,-c(1,2,c(5:8))]

  response = all_coin_data[which(data_refined$Coin%in%Group),1]

  ##Find group that is all coins. This is a bit sloppy
  if(length(Group) < 9){
    data_refined = data_refined[which(data_refined$Coin%in%Group),]
  }

  ##Exclude the coin factor
  Sigma = cor(data_refined[,-1])

  evec = eigen(Sigma)$vectors
  eval = eigen(Sigma)$values

  ##Exclude the coin factor
  pr.out = prcomp(data_refined[,-1], scale=TRUE)
  pve = eval/sum(eval)
  yvec = cumsum(pve)

  pr.out
  ##Find 80% cut off
  if(verbose){
    plot(yvec, type='b', xlab = "Principal Components",
         ylab = "Cumulative variance", main = "",ylim=c(0,1))
  }

  ##Lets take first 5 about 80%

  ##Compute the data value for principle component #1 at each data point, exclude coin
  pca_var = data.matrix(data_refined[,-1])%*%matrix(pr.out$rotation[,1:5], nrow=27, ncol=5)

  model = glm(response~pca_var)

  if(verbose){
    summary(model)
  }
}

```

```

##
outputs = model$coefficients[1]
+ model$coefficients[2]*pca_var[,1]
+ model$coefficients[3]*pca_var[,2]
+ model$coefficients[4]*pca_var[,3]
+ model$coefficients[5]*pca_var[,4]
+ model$coefficients[6]*pca_var[,5]

as_binary = as.integer(outputs > .5)

if(verbose){
  cat("Accuracy:", mean(as_binary == response))
  plot(as_binary)
}

## real data, model's guess, model
return(c(list(response), list(as_binary), list(model)))
}

Group_1 = c(c("eth", "ethereum"),c("etc", "ethereum classic"), c("rep", "augur"),
  c("steem", "steem"))
model_1 = build_group_model(Group_1, all_coin_data)

Group_2 = c(c("dash", "dash"),c("ltc", "litecoin"), c("xmr", "monero"),
  c("maid", "maidsafecoin"))
model_2 = build_group_model(Group_2, all_coin_data)

Group_3 = c(c("xem", "nem"),c("xrp", "ripple"))
model_3 = build_group_model(Group_3, all_coin_data, verbose=TRUE)

'''

'''{r, bootstrap}
##Bootstrap to try to see the standard error of our estimator
## It looks great to see what we are ~53% correct
## But looking at # of times it says up it is very few.
## Looking at percent of times it says yes and it is actually yes is more showing
## This is near 50% or random

error_full_1 = c()
error_full_2 = c()
error_full_3 = c()
error_full_4 = c()

error_1_1 = c()
error_1_2 = c()
error_1_3 = c()
error_1_4 = c()

```

```

error_2_1 = c()
error_2_2 = c()
error_2_3 = c()
error_2_4 = c()

error_3_1 = c()
error_3_2 = c()
error_3_3 = c()
error_3_4 = c()

for(i in 1:50){
  ##Fill NA with zero
  data = all_coin_data[sample(nrow(all_coin_data), replace = TRUE, nrow(all_coin_data)),]

  full_model = build_all_model(data)
  error_full = compute_errors(full_model[[1]], full_model[[2]])
  error_full_1 = c(error_full_1, mean(error_full == 1))
  error_full_2 = c(error_full_2, mean(error_full == 2))
  error_full_3 = c(error_full_3, mean(error_full == 3))
  error_full_4 = c(error_full_4, mean(error_full == 4))

  model_1 = build_group_model(Group_1, data)
  error_1 = compute_errors(model_1[[1]], model_1[[2]])
  error_1_1 = c(error_1_1, mean(error_1 == 1))
  error_1_2 = c(error_1_2, mean(error_1 == 2))
  error_1_3 = c(error_1_3, mean(error_1 == 3))
  error_1_4 = c(error_1_4, mean(error_1 == 4))

  model_2 = build_group_model(Group_2, data)
  error_2 = compute_errors(model_2[[1]], model_2[[2]])
  error_2_1 = c(error_2_1, mean(error_2 == 1))
  error_2_2 = c(error_2_2, mean(error_2 == 2))
  error_2_3 = c(error_2_3, mean(error_2 == 3))
  error_2_4 = c(error_2_4, mean(error_2 == 4))

  model_3 = build_group_model(Group_3, data)
  error_3 = compute_errors(model_3[[1]], model_3[[2]])
  error_3_1 = c(error_3_1, mean(error_3 == 1))
  error_3_2 = c(error_3_2, mean(error_3 == 2))
  error_3_3 = c(error_3_3, mean(error_3 == 3))
  error_3_4 = c(error_3_4, mean(error_3 == 4))
}

cat("\n Error Grid: Full Model")
e_f = c(mean(error_full_1), mean(error_full_2), mean(error_full_3), mean(error_full_4))
e_f_sd = 2*c(sd(error_full_1), sd(error_full_2), sd(error_full_3), sd(error_full_4))
matrix(e_f, ncol=2, nrow=2, byrow = TRUE)
matrix(e_f_sd, ncol=2, nrow=2, byrow = TRUE)

```

```

cat("\n Error Grid: Model 1")
e_1 = c(mean(error_1_1), mean(error_1_2), mean(error_1_3), mean(error_1_4))
e_1_sd = 2*c(sd(error_1_1), sd(error_1_2), sd(error_1_3), sd(error_1_4))
matrix(e_1, ncol=2, nrow=2, byrow = TRUE)
matrix(e_1_sd, ncol=2, nrow=2, byrow = TRUE)

cat("\n Error Grid: Model 2")
e_2 = c(mean(error_2_1), mean(error_2_2), mean(error_2_3), mean(error_2_4))
e_2_sd = 2*c(sd(error_2_1), sd(error_2_2), sd(error_2_3), sd(error_2_4))
matrix(e_2, ncol=2, nrow=2, byrow = TRUE)
matrix(e_2_sd, ncol=2, nrow=2, byrow = TRUE)

cat("\n Error Grid: Model 3")
e_3 = c(mean(error_3_1), mean(error_3_2), mean(error_3_3), mean(error_3_4))
e_3_sd = 2*c(sd(error_3_1), sd(error_3_2), sd(error_3_3), sd(error_3_4))
matrix(e_3, ncol=2, nrow=2, byrow = TRUE)
matrix(e_3_sd, ncol=2, nrow=2, byrow = TRUE)

'''

```