Muhammad Faizan Shabbir
Sakib Shaikh

Introduction:

In an increasingly connected world, a person can broadcast their thoughts and opinions in real time as an event unfolds. We attempt to collect data and analyze public opinion during an event: in this case Superbowl 49. Superbowl 49 took place on February 1st between the 1st ranked AFC Patriots and 1st ranked NFC Seahawks. In the weeks leading up to the game, the Patriots were accused of deflating balls in the AFC championship game, leading to coining the term "DeflateGate". Ranked as one of the most evenly matched superbowls in history, the second half proved to be exciting featuring multiple lead changes. The game came down to the wire, with the outcome being decided with only 20 seconds left. Thus we chose to analyze how public opinion changed over the second half, February 1st 5:40 PM PST to 6:50 PM PST.

With over 500 million tweets per day, and 288 million active users, Twitter is a good indicator of public opinion. Twitter is an online social platform where users can post short, 140 character messages called tweets. Users can follow others and "retweet" messages from other twitter users to their own followers. As such, each user has a feed of tweets that is based upon the tweets of all the people they follow.

In order to study public opinion over this time interval, we used the Topsy API to query tweets made with hashtags of interest. We studied the following hashtags: #Superbowl, #NFL, #DeflateGate, #DeflatedBalls, #SNL, #Colts.

Analyzing Top Tweets:

We determined the ranking of tweets by a Twitter metric called impressions. An impression can be thought of as a tweet reaching a user. A simple intuition can be that the more impressions a tweet has, the more users it has reached. The Topsy API returns tweets in descending order of exposure by default. Exposure is the total number of impressions a tweet has. The Topsy API has a 'sort-by' method, and one can specify if they would like the tweets to be sorted by such things as relevance, exposure, etc. Starting with the hashtag: #Superbowl, we first analyzed the top 5 tweets made in the second half.

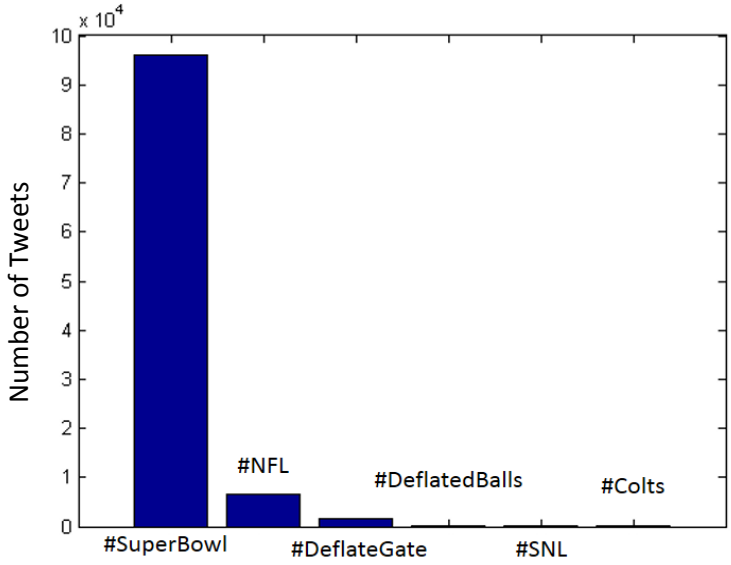
User	Tweet	Time Posted
britneyspears	Just ran into this guy \ud83d\ude0e @IamStevenT #SuperBowl #reunited #2001halftimeshow http://t.co/KWfgatPJlg	2/1/2015, 6:01:54 PM
piersmorgan	Brady's balls are clearly too inflated. #SuperBowl	2/1/2015, 5:49:25 PM
google	So ... about Tom Brady. #SuperBowl http://t.co/yIdBkp7fl2	2/1/2015, 5:43:00 PM

rickyrozay	#SuperBowl #blackbottle celebration http://t.co/S6TLzZ1zuk	2/1/2015, 5:45:38 PM
google	The Missy Elliott Halftime Show: Let ME work it. #SuperBowl http://t.co/kniMZnHvkl	2/1/2015, 6:21:33 PM

It is interesting to note that the top tweets for #SuperBowl hashtag had little to do with the actual game being played between the two teams but with the event as a whole. The tweet that generated the most number of impressions was of Britney Spears with another celebrity. It makes sense that it generated the most number of impressions as Britney Spears has nearly 41 million followers, and other popular celebrities might retweet this generating a large number of impressions as it would show up many users feeds. Compare this to the second highest rated tweet is a tweet by Piers Morgan who only has ~4.3 million followers. However, his tweet had to do with the actual game, complimenting Brady's gameplay and referencing the Deflate Gate scandal by saying "balls. . . too inflated". Our guess is that, even though he doesn't have as many followers, given the relevancy of his tweet and his celebrity status, he might have generated key retweets that increased the tweet's number of total impressions or exposure. It is also interesting that two of the top five tweets are actually related to data science and statistics as Google referenced search statistics showing the frequency of search of Brady versus Russell throughout the United States and search statistics on the most popular search terms with the name "Missy Elliot" during the half time show.

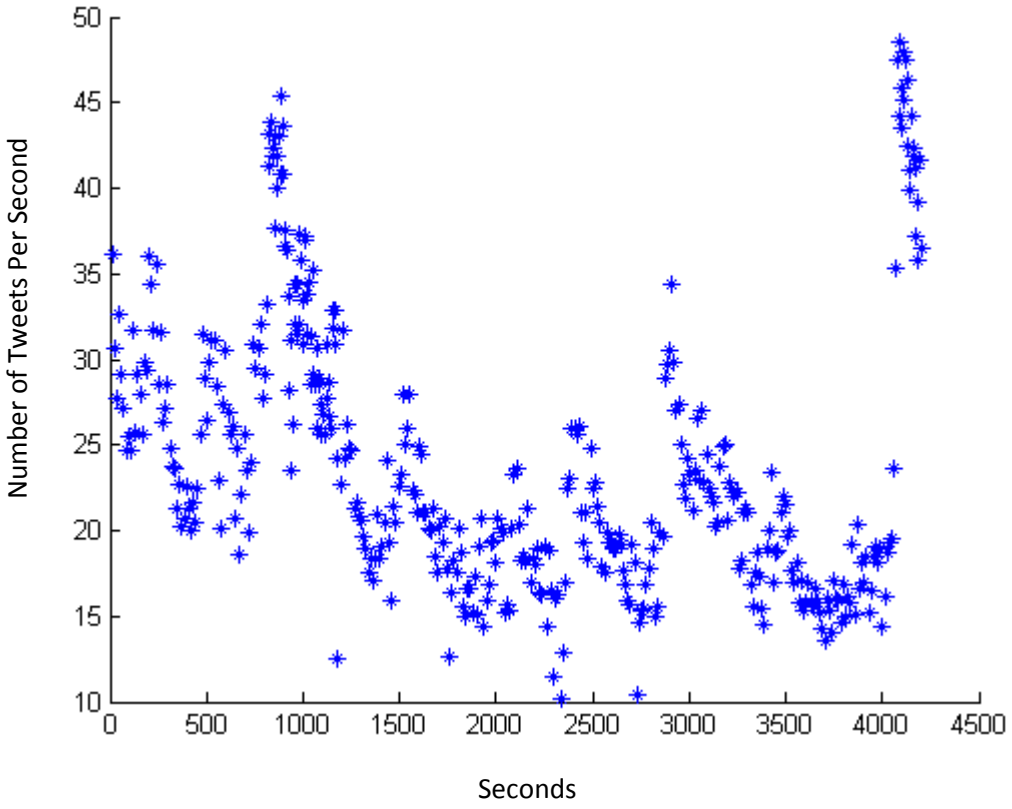
Number of Tweets Versus Hashtag:

To get the total number of tweets per hashtag, we had to query the API to build a tweet list of all the tweets over the second half. The API is limited to a maximum of 500 tweets per API call, so it was first important to design an algorithm to query in appropriate time intervals. Starting with a time interval of a 100 seconds, we would halve our time interval and query again if we detected the maximum of 500 tweets in the API response. The smallest time interval the algorithm queried for was six seconds for the #SuperBowl hashtag. The following is a bar graph showing the frequency of tweets for each hashtag.



As seen, the #SuperBowl hashtag has a much higher number of tweets, around 96,000 tweets. It took ~400 megabytes of data to store all the tweets for the #SuperBowl hashtag during the second half. The #Colts and #SNL hashtags had the lowest number of tweets with 96 and 139, as expected. Most people that tweeted about the game were talking about current events. Perusing some of the #Colts tweets, it was mostly Colts fans talking about how they wished their team had made it to the SuperBowl or using certain Seahawks plays and players as examples of what the Colts should have done or added to beat the Patriots. We expected there to be more tweets about DeflateGate and DeflatedBalls (1600 and 186 respectively) as we thought Patriots fans would be itching to brag that their team could keep up with the best defense in the league post controversy.

It would be interesting to study not only the total number of tweets but the rate of those tweets. Here is a graph showing the rate of tweets during the second half for #SuperBowl hashtag.

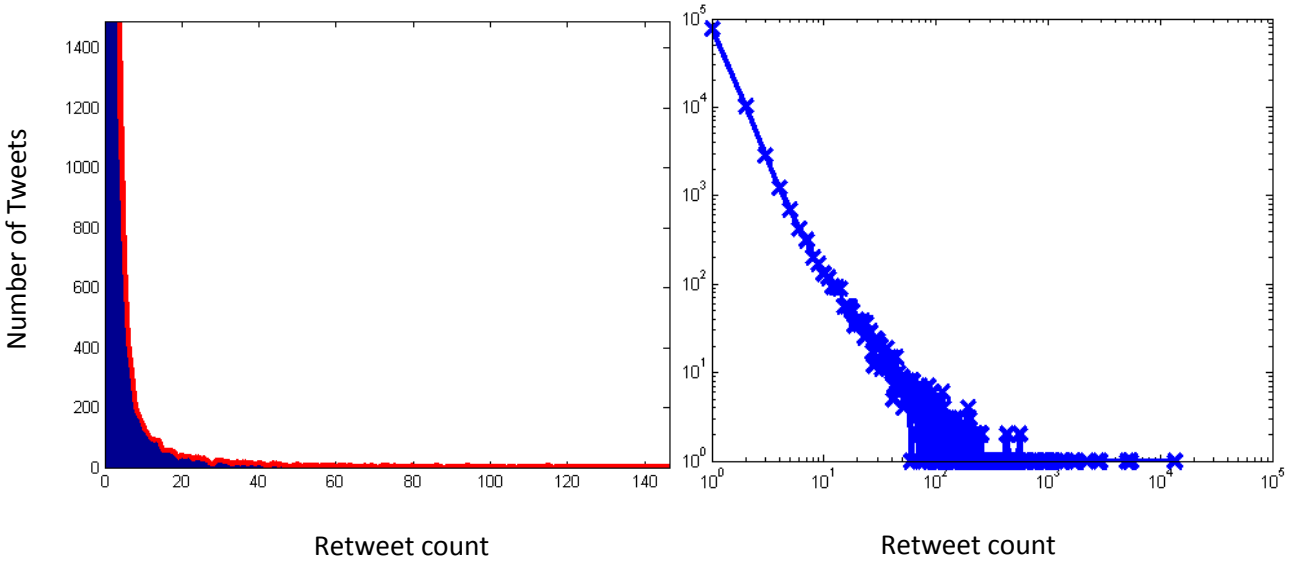


We can see some spikes in the rate of tweets per second. We will attempt to study this phenomenon later, by comparing the rate of tweets per second over time between the two most popular hashtags. This is because studying this for only one hashtag may be subject to noise, so it would be wise to correlate this by using another hashtag. We study this in detail in the section titled “Analyzing Tweet Rate”.

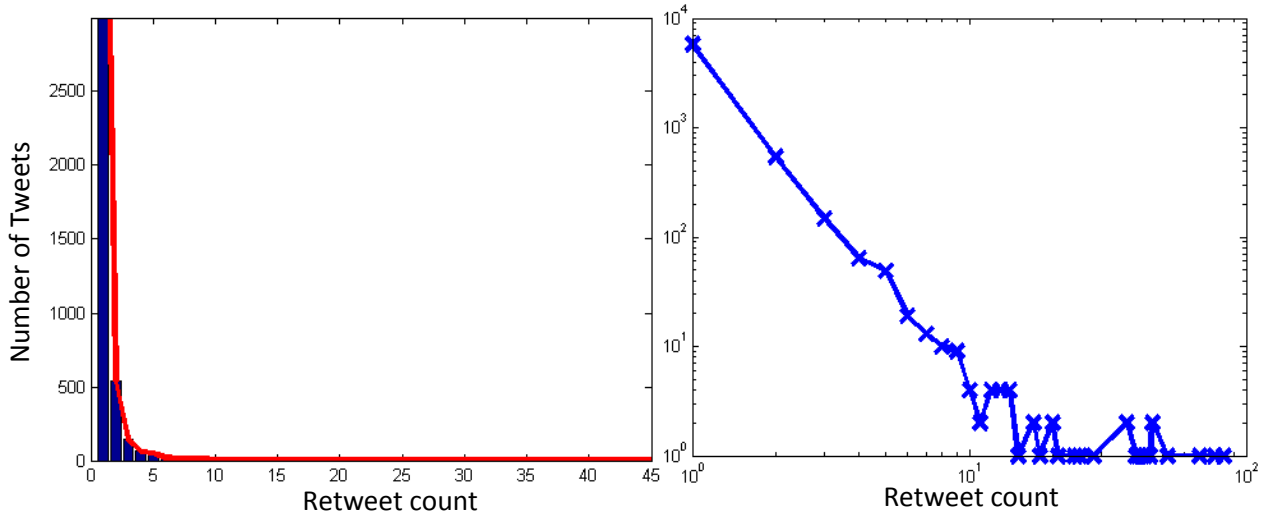
Number of Tweets versus Retweets:

Plotting number of tweets versus retweets for each hashtag showed a clear relationship in both linear and log scales. As the number of retweets increases, the corresponding number of tweets falls greatly. This corresponds to intuition as we expect that lot of tweets can get retweeted a few times, but only a few tweets get retweeted hundreds of times. In the linear scale, we see a clear exponential decay. In the corresponding log plots, we see a linear decay which is an equivalent exponential decay. Please note that for certain graphs such as the one for #Superbowl hashtag, the x and y axis have been clipped to show the relationship between number of tweets and retweets. The #Superbowl hashtag had tweets that were retweeted thousands of times, and thousands of tweets were retweeted one to two times, following the expected exponential decay. The following are graphs of numbers of tweets versus number of retweets for each hashtag. Please note that we did not consider the case of number of tweets that were retweeted zero times, as it would not add significant information.

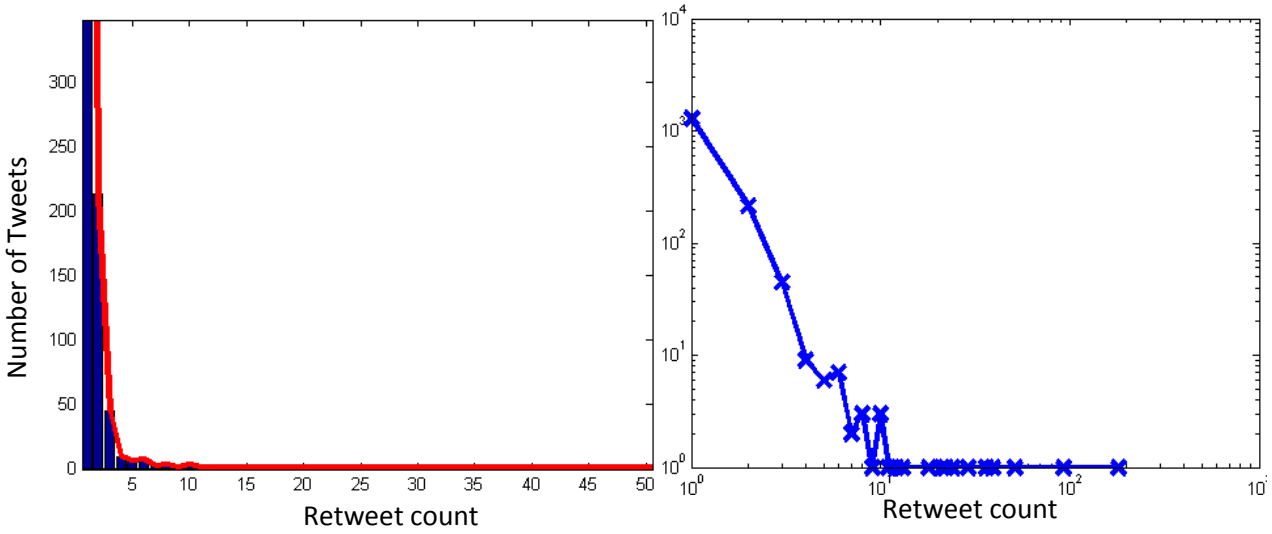
Number of Tweets versus Retweets for #Superbowl



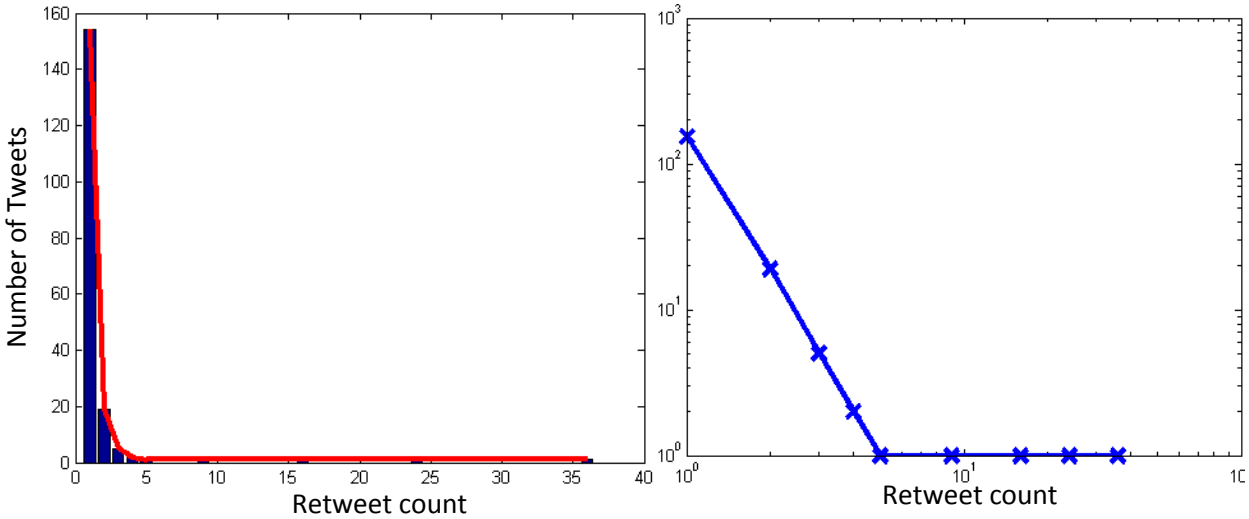
Number of Tweets versus Retweets for #NFL



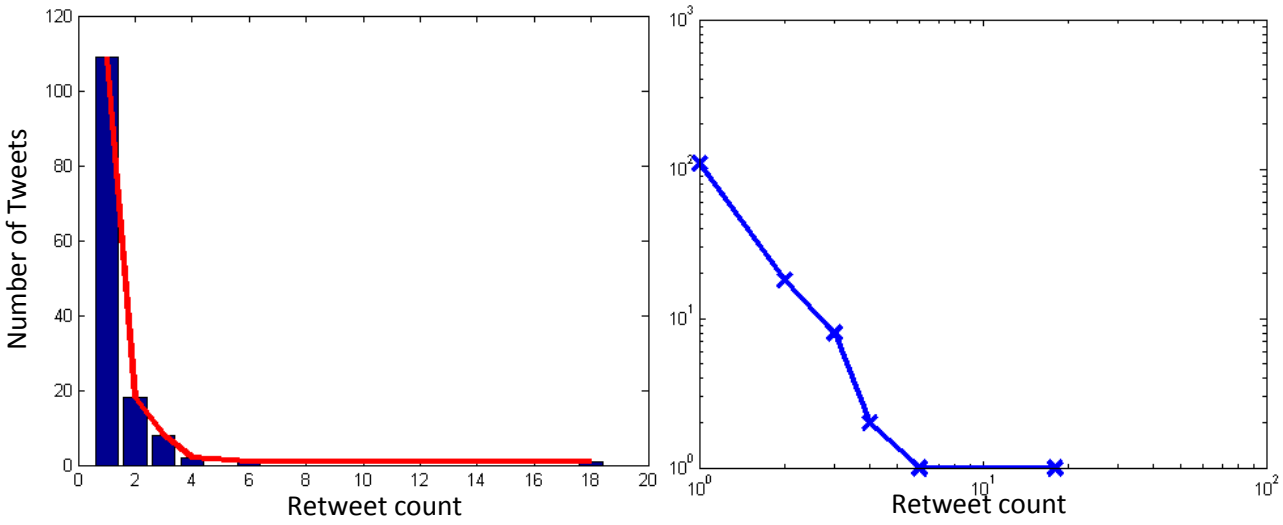
Number of Tweets versus Retweets for #DeflateGate



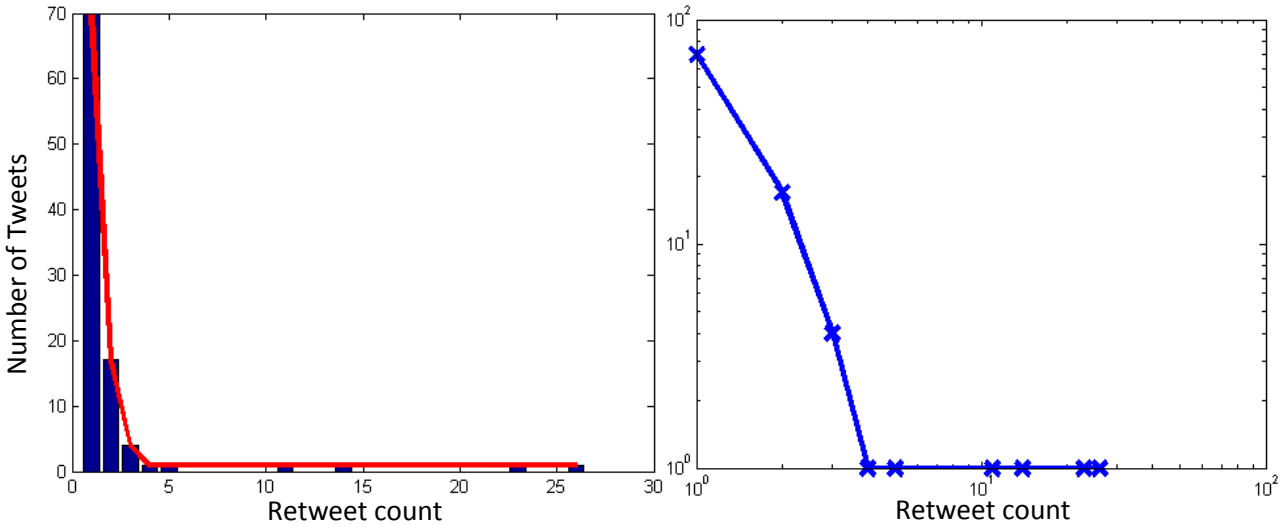
Number of Tweets versus Retweets for #DeflatedBalls



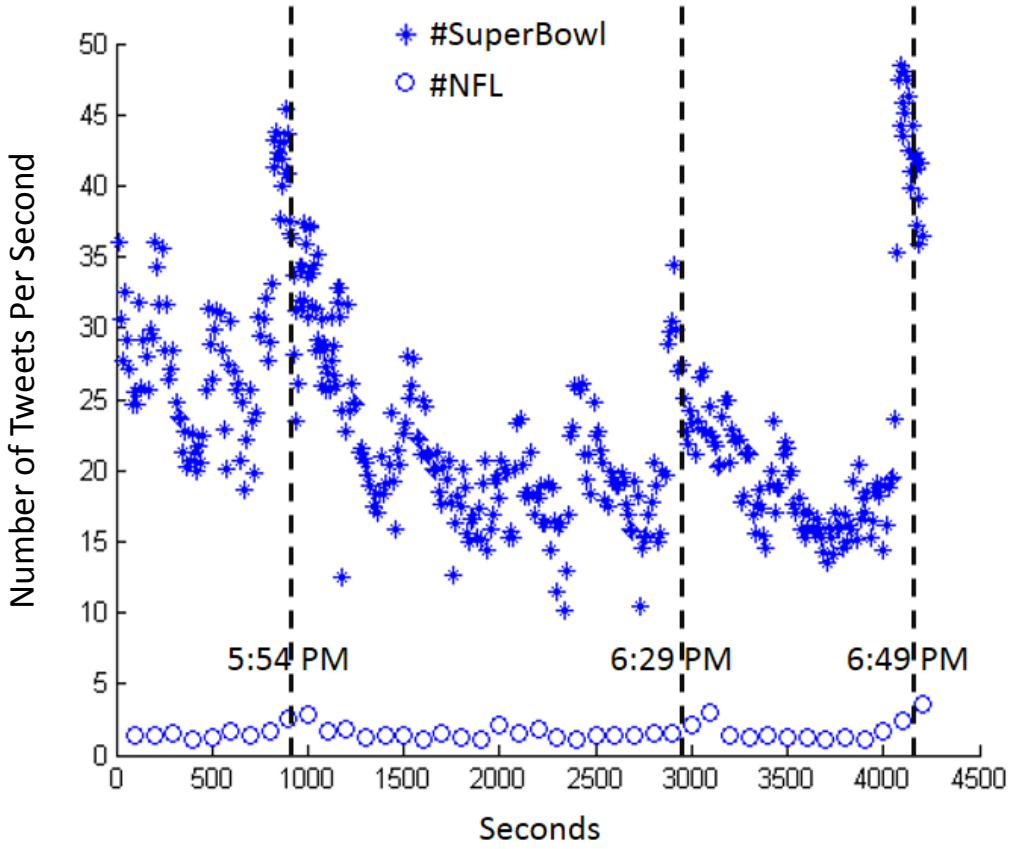
Number of Tweets versus Retweets for #SNL



Number of Tweets versus Retweets for #Colts



Analyzing Tweet Rate:



Analyzing tweet rate (or the number of tweets per second) over the second half gives us insight into when users were posting the most. We analyzed the tweet rate for the two most popular hashtags during the second half: #SuperBowl and #NFL respectively. However, as shown previously, the #SuperBowl hashtag had many more tweets than the #NFL hashtag. There are three main spikes in the tweet rate over time. We approximately noted the times of these spikes and found that they all corresponded with very significant events in the game. At 5:55 PM, the Seahawks scored a touchdown, giving them a 10 point lead in the game. At 6:28 PM, the Patriots scored a touchdown with Danny Amendola, beginning their comeback. At 6:48 PM, the Patriots scored another touchdown giving them the lead with only two minutes left on the clock in the game. It is also interesting to note that immediately following the end of halftime, in the first few hundred seconds of the graph, there is an average higher tweet rate than the average baseline in the later three quarters of the graph. This may be because people are still excited about the halftime show or the game returning or a combination of both.

APPENDIX:

Code Usage:

We used python to scrape tweets and generate tweet and text files. The python file is called build-tweets.py and is located in the root directory. The time to query is specified at the beginning of the code as global variables titled mintime and maxtime. Main function is located at the bottom of the file. The appropriate comments are located in the file.

To get the top 5 tweets for a hashtag, call the "get_top_tweet" function with the desired hashtag as an argument. This generates a text file under the folder "top_tweets" appended with the query hashtag as part of the file name.

To scrape tweets for a hashtag (or multiple hastags), call the "twitter_crawler" function, specifying the query string, mintime, maxtime, and the starting time interval over which we initially query the API. If the maximum number of tweets (500) is reached while querying the time interval, it is automatically halved and called. The function loops using the time interval to grab all the tweets for the hashtag between mintime and maxtime. This generates three text files for each call. One is a text file called tweets located under the "tweets" folder which actually contains all the tweets and metrics for each tweet in JSON format. The other two text files are logs and statistics for the results of each API call that got the corresponding tweets. They are saved under the "logs" folder and contain information on the query string, the from timestamp, the to timestamp, and the number of results returned.

To find number of tweets versus number of retweets, call the "unique_tweets" function which parses previous results for only unique tweets for a hashtag and finds its corresponding number of retweets using the ['metrics']['citations']['total'] field. The function generates a text file in the "tweet_counts" folder.

To parse all the tweets for a particular hashtag, call the "parse_tweets" function. It will parse the tweet file and output the user, post date, number of retweets and tweet text for each tweet in the text file.

Data Analysis:

We used Matlab to analyze number of tweets, tweet rate, line fits, and generate figures. The scripts are located under the "matlab" folder. Calling the script "parse_logs" for each hashtag will read in the log text files generated by python to generate information about tweets in the workspace. Calling "analyze_tweet_rate" will perform number of tweet analysis and tweet rate analysis. Calling "parse_retweet_count" reads in the text file under "tweet_counts" for a particular hashtag. Call "analyze_retweet_count" to analyze information of number of tweets versus number of retweets.