How Fast Does News Spread?

An Investigation on the Rate of Spread of News on Digital Media

Statistic Competition 2019 Essay Category

River Valley High School (JC)

18J06 - Xavier Tan

18J06 - Yong Chen How

18J06 - Zhang Jikun

18J06 - Zhang Zeyu

Table of Contents

Introduction

Twitter is an American online news and social networking service on which users post and interact with messages known as "tweets". Misinformation and fake news are common in the site due to its fast sharing capabilities and no direct method for users to fact check. Automated bots can also influence opinions and decisions of the public by making a viewpoint seem more popular than it actually is. Hence, in the age of the new media, lies and fake news might spread faster than the truth [1], and become a weapon used in politics to incite conflict or to further political agenda. [2]

Research Question

What is a suitable model for the rate of spread of news on Twitter?

Hypothesis

1. The rate of spread of news follows a logarithmic trend.
2. The level of sensationalism or truthfulness of a news story affects its reach but does not deviate from the logarithmic trend.

Methodology

Data Collection

The number of ‘favourites’ and ‘retweets’ of tweets over time was monitored with a specifically designed twitter application built with python (See Annex) and ran on 9 different news sources. Each tweet is monitored for an hour at change is tracked using 10-seconds intervals. The ‘Tweepy’ Python library [3] and the Twitter application programming interface (API) [4] are employed. All data are collected on 10th March from 6am to 10am GMT.

News Source	Tweets Monitored
ABC News	10
BBC World	10
CNN	10
New York Times	10
Reuters	10
The Economist	10
Time	10
The Washington Post	10
Wall Street Journal	10

Data Normalization

The number of retweets and favorites from different news sources vary due to their relative popularity[1]. To compare them, the data is normalized by calculating

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

where $x = (x_1, ..., x_n)$ and x_i and z_i is the number of retweets and normalized data at the i^{th} time interval respectively. In our case, $min(x)$reduces to 0 and z_i is simply the ratio of x_i to the total number of retweets.

[1] This can be reflected by the amount of 'followers' each news source has on Twitter.

Results & Discussion

Retweets (n=90)

The average number of retweets R over time t (in intervals of 10 seconds) across all news sources studied follows a logarithmic trend

$$R(t) = \alpha ln(t) + \beta$$

where α and β are constants, with a R-squared value[2] of 97.82% (Figure 1).

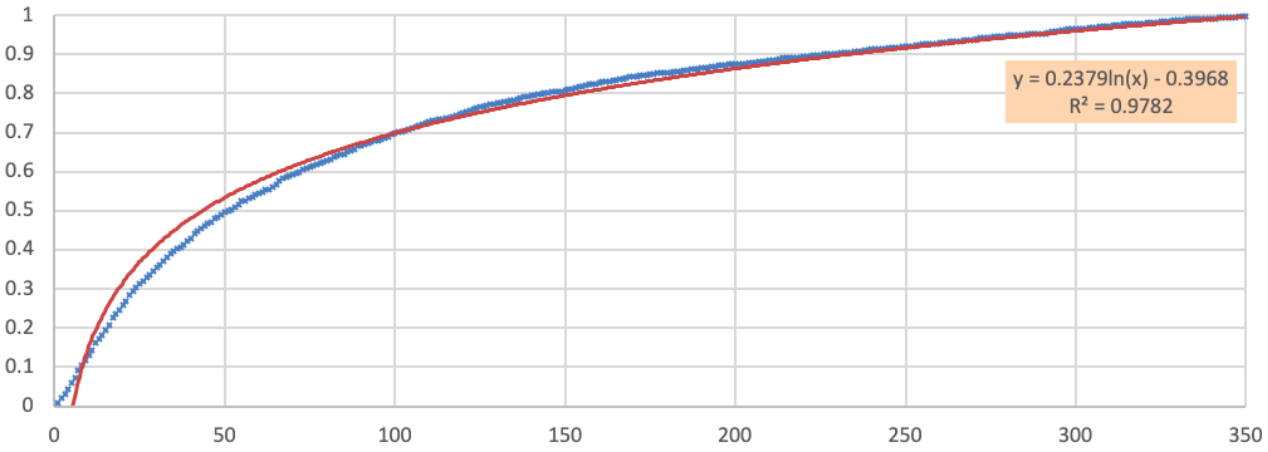


Figure 1: Average Number of Retweets Over Time

[2] Expressed as a percentage, R-squared is a statistical measure of how close the data are to the fitted regression line

It follows that the expected rate of change of R with respect to t is given by

$$\frac{dR}{dt} = \frac{\alpha}{t}$$

This is reflected in Figure 2.1 and 2.2.

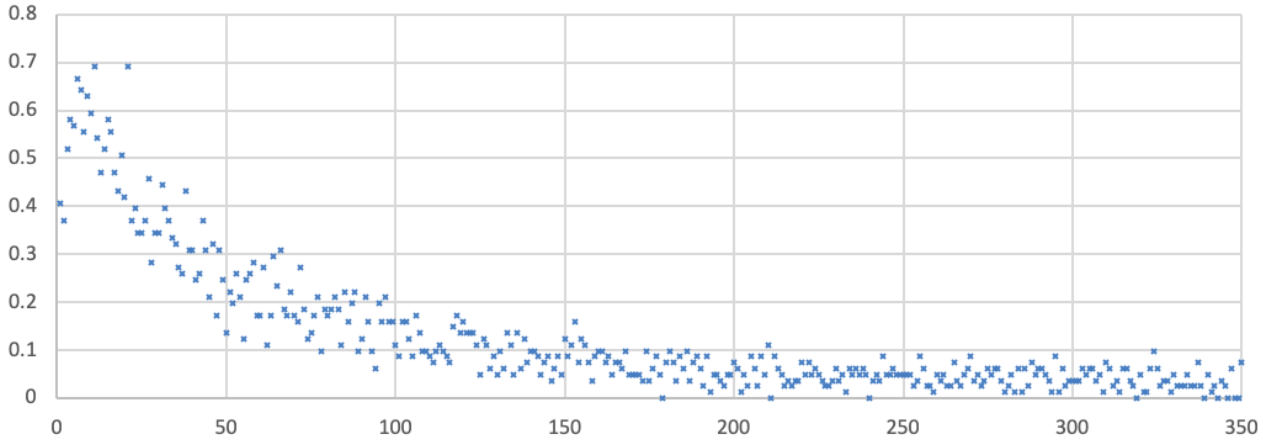


Figure 2.1: Average Rate of Change of Retweets Over Time (Linear Scale)

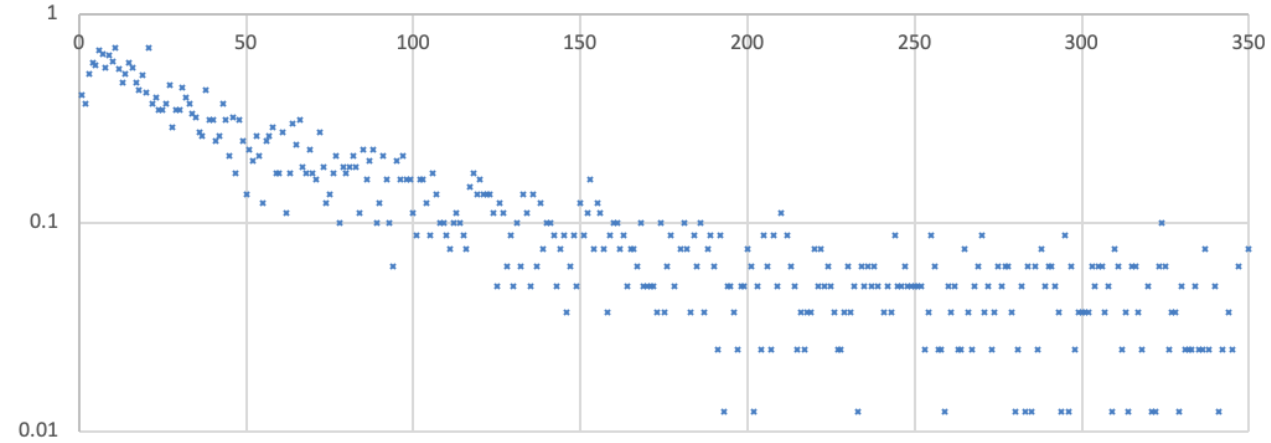


Figure 2.2: Average Rate of Change of Retweets Over Time (Logarithmic Scale)

Favorites (n=90)

The number of favorites from different news sources was similarly normalized and compared.

The average number of favorites Fover time t (in intervals of 10 seconds) across all news sources studied follows a logarithmic trend

$$F(t) = \alpha ln(t) + \beta$$

where α and β are constants, with a R-squared value of 97.61% (Figure 3).

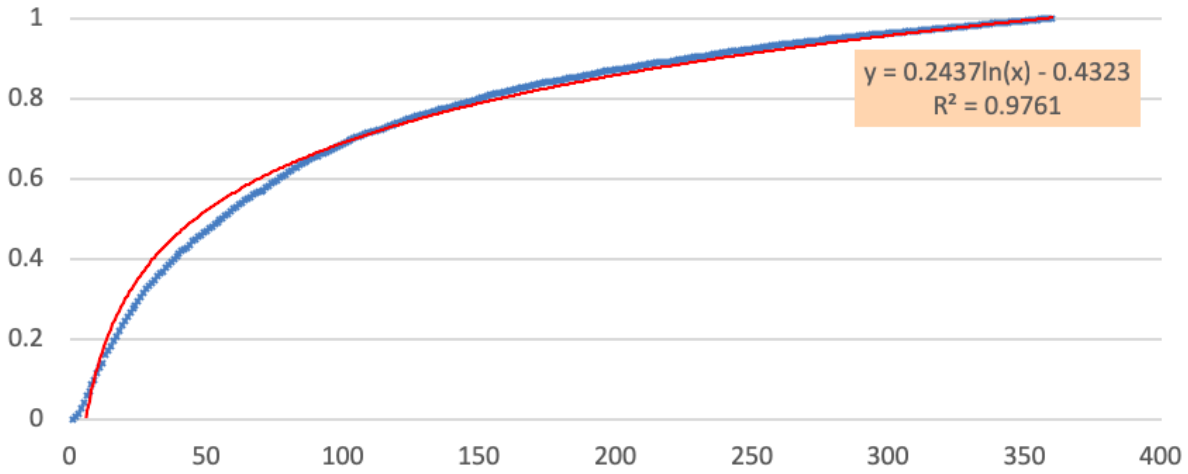


Figure 3: Average Number of Favorites Over Time

It follows that the expected rate of change of R with respect to t is given by

$$\frac{dR}{dt} = \frac{\alpha}{t}$$

This is reflected in Figure 4.1 and 4.2.

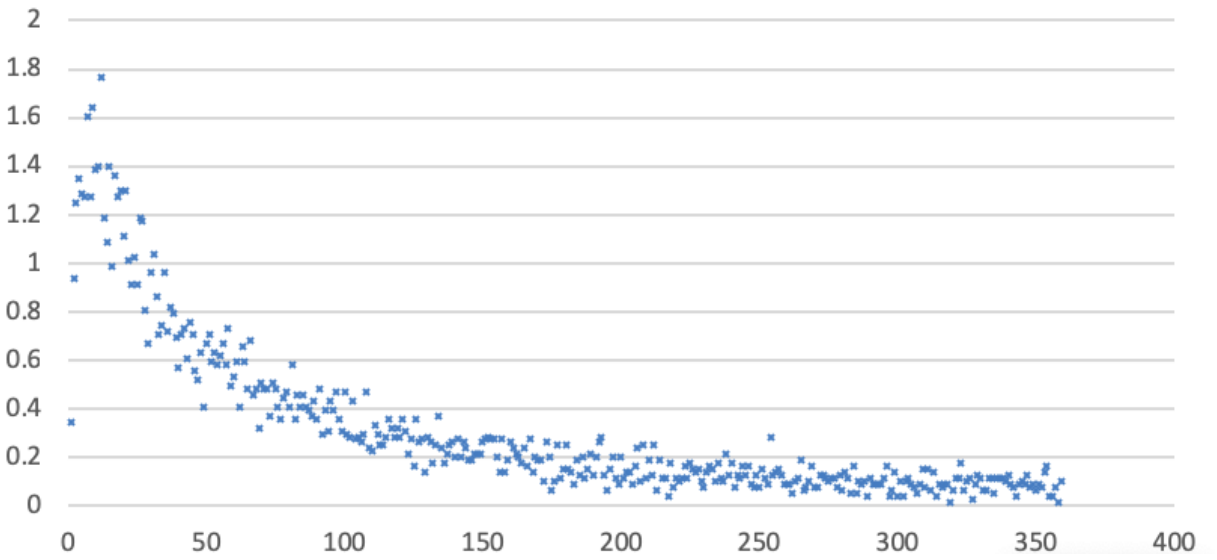


Figure 4.1: Average Rate of Change of Retweets Over Time (Linear Scale)

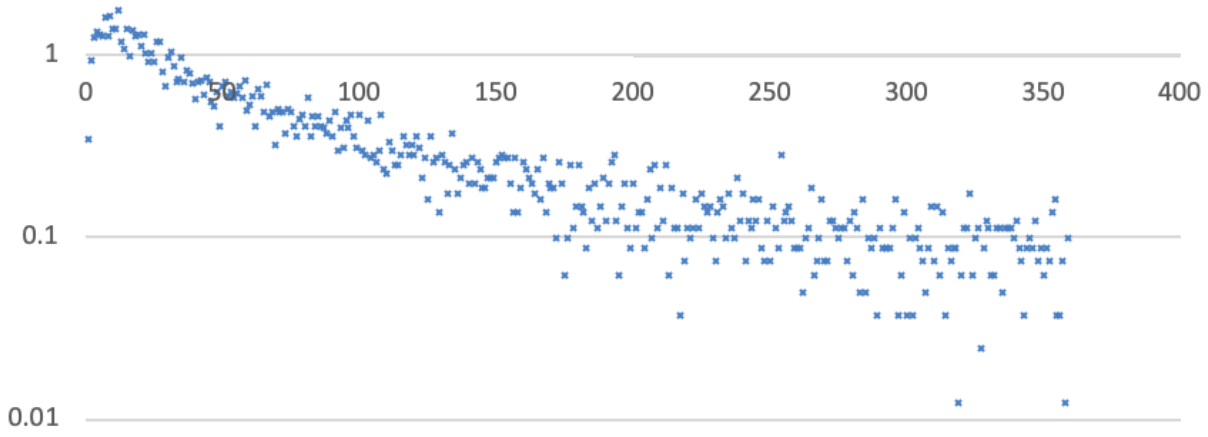


Figure 4.1: Average Rate of Change of Retweets Over Time (Logarithmic Scale)

Discussion

There are several implications of this model:

1. There is a large amount of retweets and favorites right after the tweet is posted, likely due to features such as follower alerts and 'trending' topics.

2. $\lim_{t \to \infty} \frac{dR}{dt} = \lim_{t \to \infty} \frac{dF}{dt} = 0$. As time increases, the rate of change of retweets and favorites decreases exponentially.

Additionally, the constant α differs significantly across different news stories, as seen in Figure 5. Some stories have a larger reach due to their level of sensationalism or shock value. However, it can be observed that all tweets follow the same general trend. We can hence conclude that the model is valid regardless of the type of news story monitored.

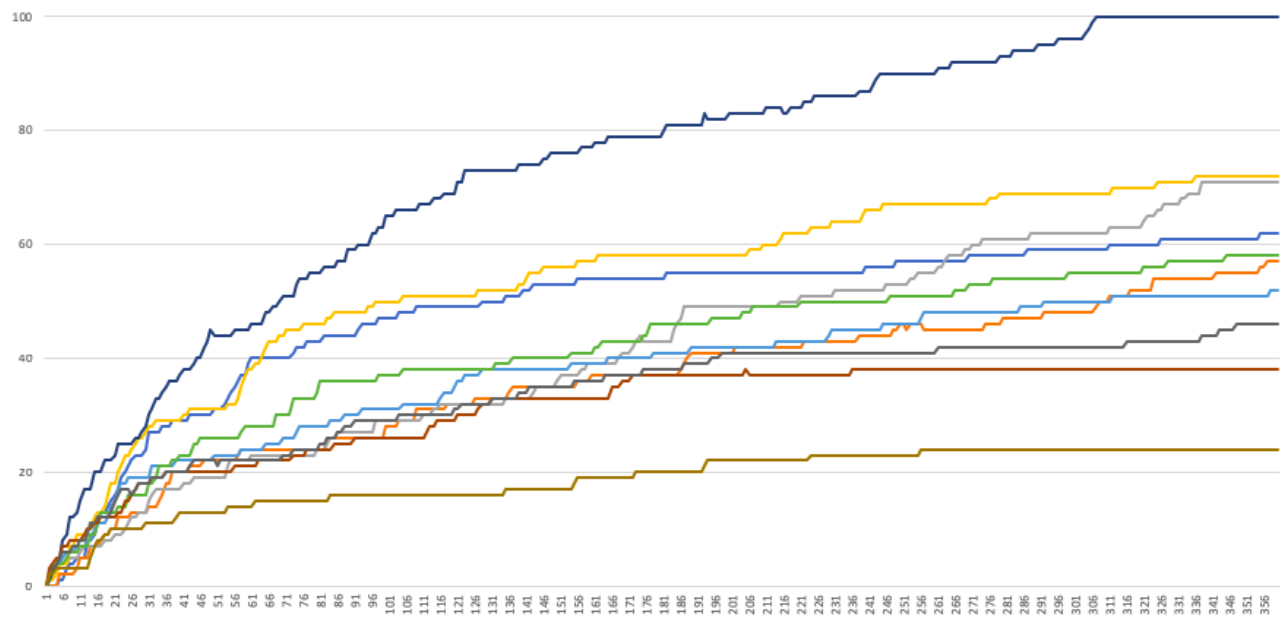


Figure 5: Number of Retweets Over Time of Different Tweets (CNN News)

We note that for time $t < 100$, the logarithmic model is not as good an approximation as the plotted points deviate from the best-fit curve. Perhaps for $0 \leq t \leq 50$, a logistic growth model would be more suitable, as the gradient of the curve is observed to increase, then decrease shortly after (Figure 6).

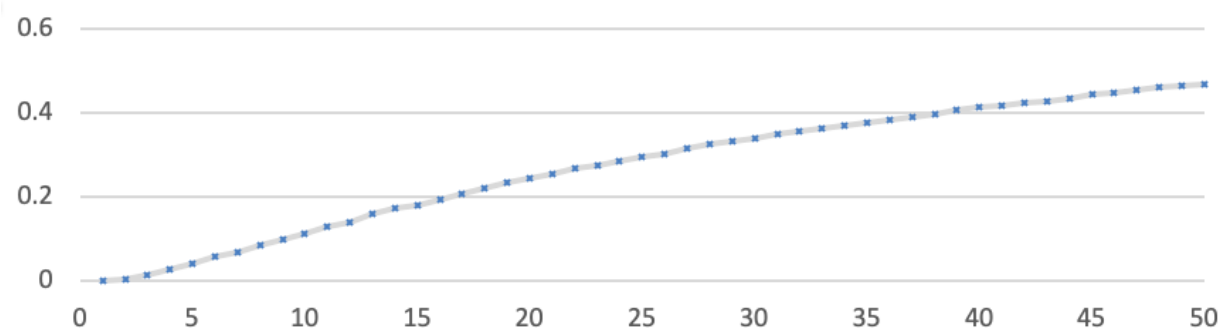


Figure 6: Average Number of Retweets Over Time for $0 \leq t \leq 50$

Hence we can derive the following,

$$\frac{dR}{dt} = rR\left(1 - \frac{R}{K}\right)$$

$$\frac{dR}{dt} = \frac{r}{K}\left(RK - R^2\right)$$

$$\frac{dR}{dt} = \frac{-r}{K}\left[\left(R - \frac{K}{2}\right)^2 - \frac{K^2}{4}\right]$$

$$\int\left[\left(R - \frac{K}{2}\right)^2 - \frac{K^2}{4}\right]^{-1} dR = \int\frac{-r}{K}dt$$

$$\frac{1}{K}ln\left|\frac{R-K}{R}\right| = \frac{-r}{K}t + C$$

$$ln\left|\frac{R-K}{R}\right| = CK - rt$$

$$\frac{R-K}{R} = Ae^{-rt},\ A = \pm e^{CK}$$

$$R\left(1 - Ae^{-rt}\right) = K$$

This yields R to be given by

$$R = \frac{K}{1-Ae^{-rt}}$$

where K is the maximum number of retweets, and A and r are constants.

The same analysis and result can be used for F.

$$F = \frac{K}{1-Ae^{-rt}}$$

where K is the maximum number of favorites, and A and r are constants.

Limitations

One limitation with our model is that the number of retweets or favorites may not be sufficient to indicate the spread of news. This is because people may view the news story but not retweet or favorite it.

Additionally, our logarithmic model does not account for the time period when the news article was just posted. The model is only valid for the range $t > 0$, since $ln(0)$is undefined. For a small amount of time dt after the news story is posted, it is difficult to model accurately the number of retweets, especially when the time axis is plotted in 10 second intervals. A more accurate model might then be $y = \alpha ln(t + 1)$, so that the graph passes through the origin. This would, however, affect the R-squared value near the origin.

Furthermore, the logarithmic model suggests that as $t \to \infty$, $R \to \infty$ and $F \to \infty$. However, this is unlikely in real life since the number of interactions usually tend towards a finite value as time goes by. This is because of the fundamental workings of social media platforms: as time increases, posts (tweets in our case) experience a heavy deay in their extent of reach.

The latter two limitations are not as significant as the first, since we are only interested in the rate of spread of news during the period of time when the news source is still relevant. Hence, we are not interested in $t = 0$, nor are we interested in $t \to \infty$. Since $\frac{dN}{dt}, \frac{dR}{dt} \to 0$ rather quickly, we can assume that there is a saturation point beyond which the rate of change of retweets and favorites becomes negligible.

Conclusion

Results obtained support the hypothesis that the rate of spread of news generally follows a logarithmic trend, and that the trend is valid regardless of the level of sensationalism in the news. For a short amount of time, a logistic growth model is a better approximation of the rate of spread of news. Keywords with shock value such as ‘NEW’ or ‘BREAKING’ or shocking stories with much ‘reshare value’ had a larger reach than normal stories.

Reflection

We learnt the value of data and statistics in modelling real-world contexts. Here, we assume that the number of favourites is a representation of the number of people who have viewed the tweet and a good yardstick to measure the rate of spread of news. Through analysis of real world data, we are able to carefully evaluate the impacts of social media and new media in which the public assesses information. Misinformation and disinformation have plagued democracies around the world, induced hatred and polarized political landscapes. Learning more about the spread of news allows us to make informed decisions to curtail the spread of falsehoods.

Future studies should aim to model the extent of the reach of real and fake news by considering variables such as keywords in the news title, the news source, the time of publishing and its target audience.

References

[1] Vosoughi, S., Roy, D., & Aral, S. (2018, March 09). The spread of true and false news online. Retrieved from http://science.sciencemag.org/content/359/6380/1146.full

[2] Gu, L., Kropotov, V., & Yarochkin, F. (n.d.). The Fake News Machine - Trend Micro Internet Security. Retrieved from https://documents.trendmicro.com/assets/white_papers/wp-fake-news-machine-how-propagandists-abuse-the-internet.pdf

[3] Tweepy Documentation (n.d.). Retrieved from https://tweepy.readthedocs.io/en/v3.5.0/

[4] Twitter Developer Platform - Twitter Developers. (n.d.). Retrieved from https://developer.twitter.com/

Annex

Tracker App Code

Language: Python

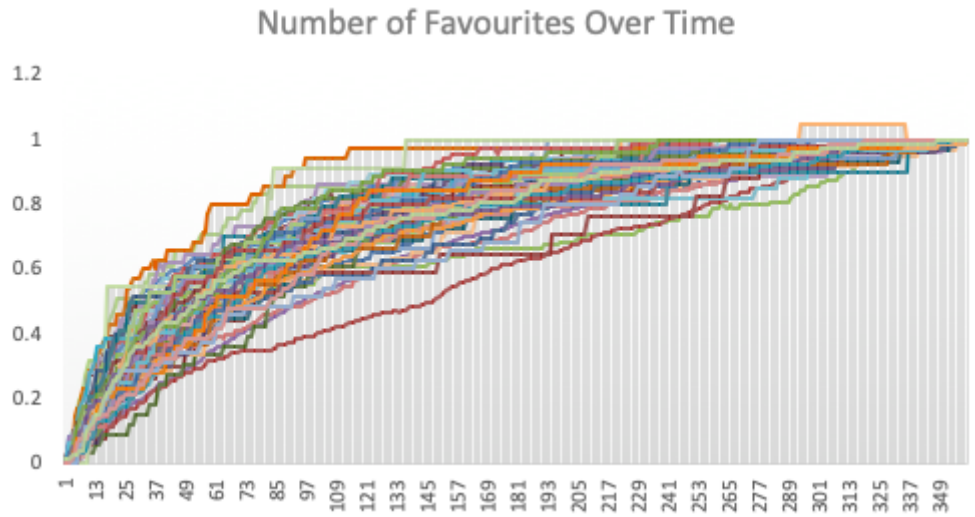
```
from credentials import twitter_consumer_key, twitter_consumer_secret,
twitter_assess_token, twitter_assess_token_secret
from datetime import datetime, timedelta
import time
import tweepy
import xlwt

auth = tweepy.OAuthHandler(twitter_consumer_key, twitter_consumer_secret)
auth.set_access_token(twitter_assess_token, twitter_assess_token_secret)

api = tweepy.API(auth)

class Tweet():
    def __init__(self, Id, text, starttime):
        self.id = Id
        self.text = text
        self.starttime = starttime
        self.favorite_count = []
        self.retweet_count = []
    def get_id(self):
        return self.id
    def add_favorite(self, favourite):
        self.favorite_count.append(favourite)
    def add_retweet(self, retweet):
        self.retweet_count.append(retweet)
    def get_starttime(self):
        return self.starttime
    def get_favourite(self):
        return self.favorite_count
    def get_retweet(self):
        return self.retweet_count

class tweet_tracker():
    def __init__(self, screen_name, tweet_limit, limit, time_interval):
        self.starttime = None
        self.screen_name = screen_name
        self.tweet_limit = tweet_limit
        self.limit = limit
        self.time_interval = timedelta(seconds = time_interval)
        self.tweets = []
```

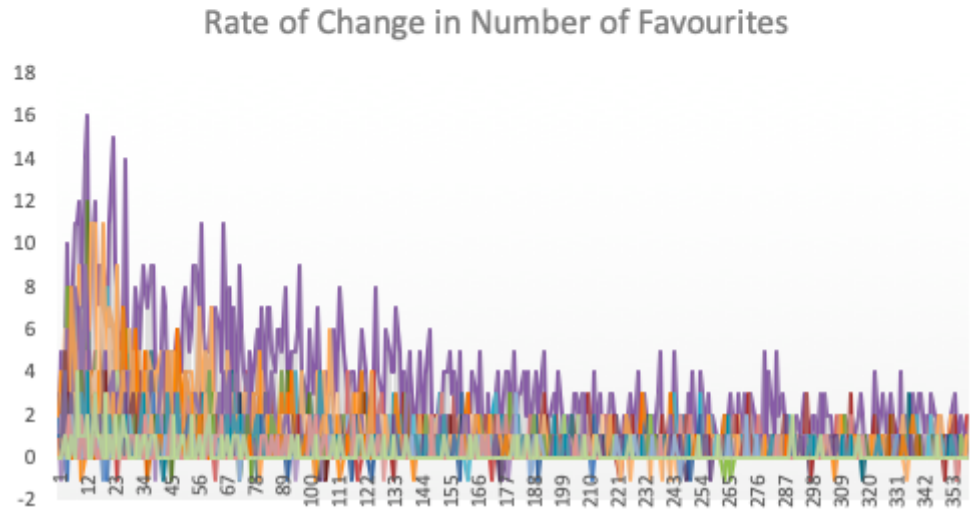
```
    def run(self):
        print("Warning: DO NOT INTERRUPT THE PROGRAM MID RUN, THE OUTPUT FILE WILL
ONLY BE SAVED AFTER PROCESS IS COMPLETED!")
        if self.time_interval < timedelta(seconds = 10):
              print("Warning: Having a time interval less than 10s might result in
inaccurate data due to slow server respond time. Especially if there is a large
number of tweets being tracked.")
        print("Start time:", datetime.now())
        self.starttime = datetime.now()
        last_tweet = api.user_timeline(screen_name = self.screen_name, count =
1)[0].id
        start_tweet = last_tweet
        wf = xlwt.Workbook()
        ws_favorite = wf.add_sheet('favourites')
        ws_retweet = wf.add_sheet('retweets')
        check_complete = False
        wait = self.starttime + self.time_interval
        while not check_complete:
              if len(self.tweets) < self.tweet_limit:
                    tweets = api.user_timeline(screen_name = self.screen_name,
since_id = start_tweet, count =200)[::-1]
                    for tweet in tweets:
                          if tweet.id > last_tweet and tweet.text[:4] != "RT @"
and len(self.tweets) < self.tweet_limit:
                                print("New tweet -", "ID:" + str(tweet.id),
"Text:" + tweet.text)
                                ws_favorite.write(0, len(self.tweets), tweet.id)
                                ws_favorite.write(1, len(self.tweets), tweet.text)
                                ws_retweet.write(0, len(self.tweets), tweet.id)
                                ws_retweet.write(1, len(self.tweets), tweet.text)
                                last_tweet = tweet.id
                                new_tweet = Tweet(tweet.id, tweet.text,
datetime.now())
                                self.tweets.append(new_tweet)
              else:
                    check_complete = True
                    tweets = api.user_timeline(screen_name = self.screen_name,
since_id = start_tweet, max_id = last_tweet)[::-1]
              finished = 0
              for i in range(len(self.tweets)):
                    if len(self.tweets[i].get_favourite()) < self.limit:
                          for tweet in tweets:
                                if tweet.id == self.tweets[i].id:
                                      check_complete = False

ws_favorite.write(len(self.tweets[i].get_favourite()) + 2, i,
tweets[i].favorite_count)
```

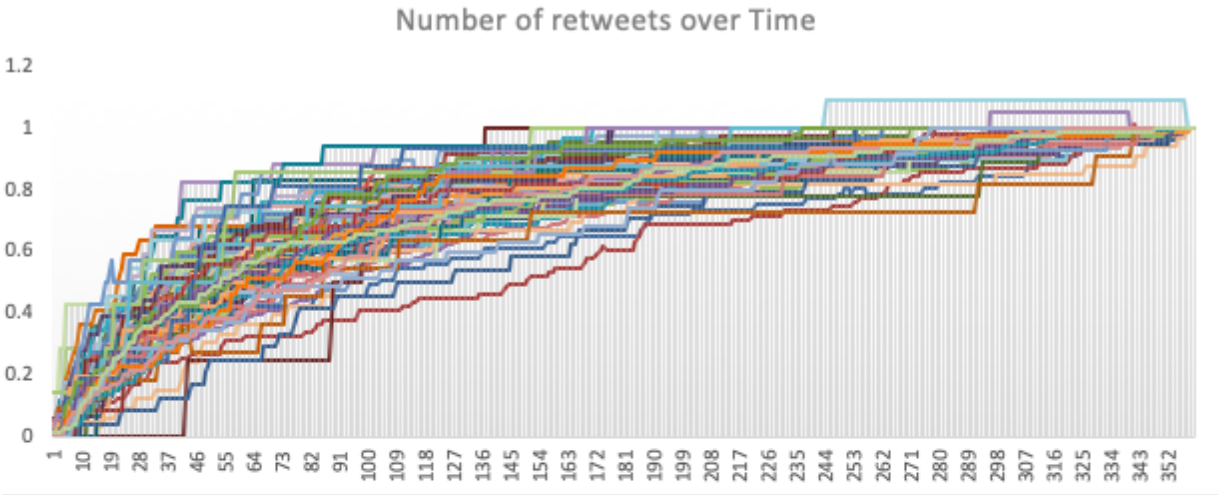
```
ws_retweet.write(len(self.tweets[i].get_retweet()) + 2, i,
tweets[i].retweet_count)

self.tweets[i].add_favorite(tweets[i].favorite_count)

self.tweets[i].add_retweet(tweets[i].retweet_count)
                    else:
                          finished += 1
              print("Log time - Ideal:", wait - self.time_interval, "Actual:",
datetime.now(), "\nTweets- Completed:", finished, "In progress:", len(self.tweets)
- finished, "Not started:", self.tweet_limit - len(self.tweets))
              if wait > datetime.now():
                    time.sleep((wait - datetime.now()).seconds)
              wait = wait + self.time_interval
              wf.save(self.screen_name+'.xls')
       print("Process completed, output file is", self.screen_name+'.xls', "Total
runtime:", str(datetime.now()-self.starttime))
```

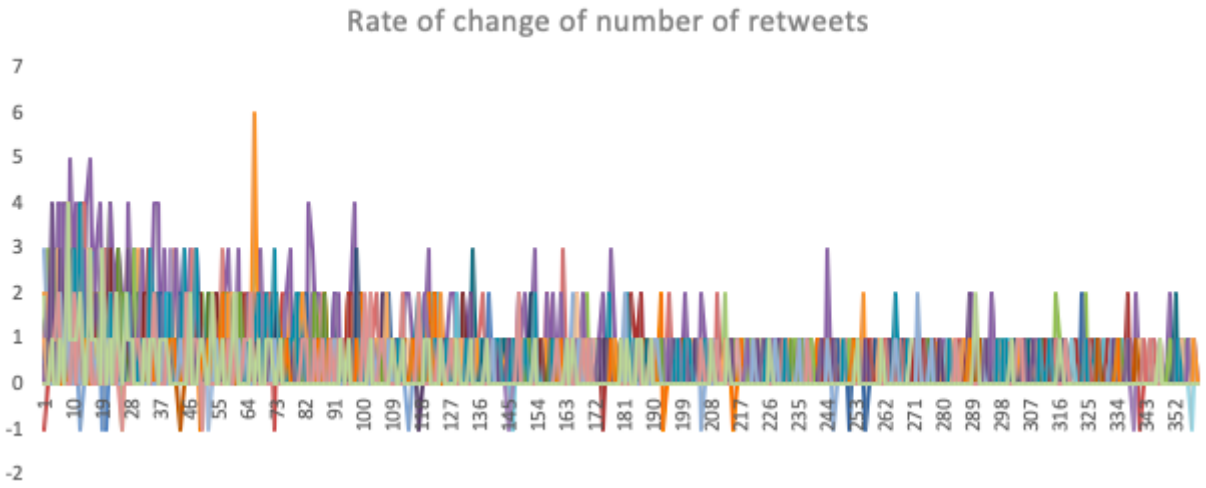
Additional Data

All tweets monitored are plotted in the graphs below.






Number of retweets over Time
1.2
1
0.8
0.6
0.4
0.2
0
1
10
19
28
37
46
55
64
73
82
91
100
109
118
127
136
145
154
163
172
181
190
199
208
217
226
235
244
253
262
271
280
289
298
307
316
325
334
343
352


Rate of change of number of retweets
7
6
5
4
3
2
1
0
-1
-2
1
10
19
28
37
46
55
64
73
82
91
100
109
118
127
136
145
154
163
172
181
190
199
208
217
226
235
244
253
262
271
280
289
298
307
316
325
334
343
352