Assignment 4

CS 532: Introduction to Web Science

Spring 2017

Grant Atkins

Finished on March 2, 2017

1

Question

1. Determine if the friendship paradox holds for my Facebook account.* Compute the mean, standard deviation, and median of the number of friends that my friends have. Create a graph of the number of friends (y-axis) and the friends themselves, sorted by number of friends (x-axis). (The friends don't need to be labeled on the x-axis: just f1, f2, f3, ... fn.) Do include me in the graph and label me accordingly.

* = This used to be more interesting when you could more easily download your friend's friends data from Facebook. Facebook now requires each friend to approve this operation, effectively making it impossible.

I will email to the list the XML file that contains my Facebook friendship graph ca. Oct, 2013. The interesting part of the file looks like this (for 1 friend):

```
<node id="Johan_Bollen_1448621116">
        <data key="Label">Johan Bollen</data>
        <data key="uid"><![CDATA[1448621116]]></data>
        <data key="name"><![CDATA[Johan Bollen]]></data>
        <data key="mutual_friend_count"><![CDATA[37]]></data>
        <data key="friend_count"><![CDATA[420]]></data>
</node>
```

It is in GraphML format: http://graphml.graphdrawing.org/

Answer

To solve this problem I wrote a script in python 3.6 called **facebookFriendship.py**, using the pygraphml dependency to help parse the xml in the **mln.graphml** file provided to us for this assignment [2]. I previously tried using the in-built xml library python offers, but found this library to be much quicker to use.

This python script goes through each node, which was a friend of Dr. Nelson on Facebook from 2013, and picks out the friend count for each user assigning that value to a dictionary with the key being the user's name. After iterating through all the nodes, Dr. Nelson was assigned a value of the total number of nodes in the file. It should be noted that there were 11 users that did not have a friend count, possibly due to privacy reason. The 11 users were: James Florance, Joy Gooden, Kim Beveridge, Alfredo Snchez, Sarah Shreeves, Sally Mauck, Dan Swaney, Robert Gordeaux, Joseph Kaplan, Michael Milner and Catherine Kemble Cronin. After this data was created I saved it to a csv called **facebookFriends.csv**.

It should also be noted that the number of nodes that the **mln.graphml** file provided was 165, but you'll notice my graph only goes up to 155. This is taken from the fact that some of the nodes in the xml file didn't have a friend count so they weren't included in this set.

I then wrote a script in R called **friendshipParadox.R** to plot these values in an ascending order based on friend count. In the plot shown in Figure 1, you can see that Dr. Nelson has many friends with higher friend counts than him, with his count being 154 (not including himself). Therefore, the friendship paradox does hold for Dr. Nelson's Facebook account. I also used this R script to compute the Mean, Standard Deviation and Median of Dr. Nelson's Facebook friend counts shown in Table 1.

Mean	**Standard Deviation**	**Median**
357.6645	370.7427	259

Table 1: Mean, Standard Deviation and Median generated from R Script for Facebook friend counts

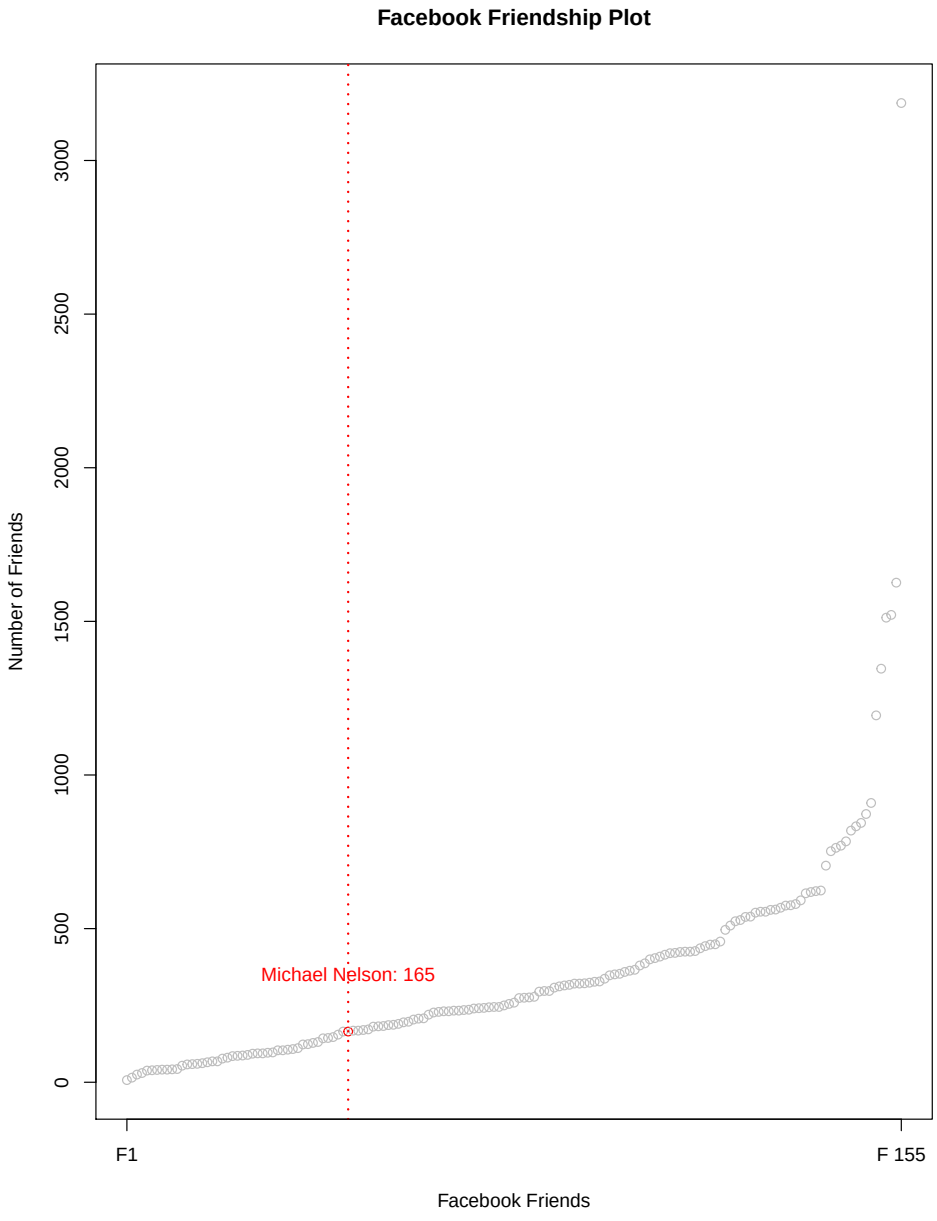

Figure 1: Plot of Dr. Nelson's Facebook friends vs. friend counts

2

Question

2. Determine if the friendship paradox holds for your Twitter account. Since Twitter is a directed graph, use "followers" as value you measure (i.e., "do your followers have more followers than you?").

Generate the same graph as in question #1, and calcuate the same mean, standard deviation, and median values.

For the Twitter 1.1 API to help gather this data, see:

https://dev.twitter.com/docs/api/1.1/get/followers/list

If you do not have followers on Twitter (or don't have more than 50), then use my twitter account "phonedude_mln".

Answer

Since my twitter account had no followers, I used Dr. Nelson's twitter account "*phonedude_mln*" for this problem. I wrote a script in python 3.6 using the Tweepy dependency to communicate with Twitter's api as shown in Listing 2. I used the pages option instead of the items option to reduce number of API calls, where pages return more json information per request. I retrieved this data and save the names of the users and their follower counts to a CSV file named **twitterFollowers.csv**.

I then wrote a script in R called **twitterFriendship.R** to plot these values in an ascending order based on follower count. In the plot shown in Figure 1, you can see that Dr. Nelson still has many followers with higher follower counts than him, with his count being 624 (not including himself). Therefore, the friendship paradox does hold for Dr. Nelson's Twitter account for followers. I also used this R script to compute the Mean, Standard Deviation and Median of Dr. Nelson's Facebook friend counts shown in Table 2.

Mean	Standard Deviation	Median
1510.395	10150.67	310

Table 2: Mean, Standard Deviation and Median generated from R Script for twitter follower count

```
1  setwd(getwd())
2
3  csv <- read.table('output/twitterFollowers.csv',header = FALSE,
       sep = ",")
4  numRows <- nrow(csv)
5  ordered <- csv[order(csv$V2),]
6  cols <- c("gray", "red")[(ordered$V1 == "phonedude_mln")+1]
7  # index position of 'Michael Nelson' in data table
8  nelsonIndex <- which(ordered$V1=="phonedude_mln")
9  count <- ordered[nelsonIndex,]
10
11 b <- plot(ordered$V2,col = cols,main="Twitter Follower Plot",
       xlab = "Twitter Followers",ylab = "Number of Followers",xaxt
       ='n')
12 abline(v=nelsonIndex,col="red",lwd=2,lty=3)
13
14 # Add f1-f_n on x-axis
15 axis(1, at=c(1,numRows),
16       lab=c("F1",paste("F",numRows)))
17 # y axis is just for padding
```

```
text(nelsonIndex,8,paste("Michael Nelson:",count$V2),col = "red
    ")
# Summary Stats
print("MEAN, SD, MEDIAN")
print(mean(ordered$V2))
print(sd(ordered$V2))
print(median(ordered$V2))
```

Listing 1: R Script for generating plot of twitter followers

```
import tweepy
import csv

# Variables that contains the user credentials to access Twitter
    API
access_token = "821042028800802816-
    E7SvwPXZKJRzazLctidudXhD0X0SgDZ"
access_token_secret = "
    hfEMDTkVBX6Kf7x8FddjBZi7joxKZIYYJztq1QFQcF8cp"
consumer_key = "RigRve4McsZdYXNpz2rwPRZfx"
consumer_secret = "
    EuFivjFeWCBmG205shXMjTPb0u56wTXJgRDRhqaWPRQU1CxYjW"


def getFollowers(api):
    data = {}
    pageJson = list()
    # limit by 200, used pages instead of items since its less
        likely to get timed out.
    for p in tweepy.Cursor(api.followers, screen_name="
        phonedude_mln",count=200).pages():
        # used extend since json would break for each page
        pageJson.extend(p)

    for user in pageJson:
        data[user.screen_name] = user.followers_count

    data["phonedude_mln"] = len(pageJson)
    writeCSV(data,"output/twitterFollowers.csv")


def getFollowing(api):
    data = {}
    pageJson = list()
    # limit by 200
    for p in tweepy.Cursor(api.friends, screen_name="
        phonedude_mln",count=200).pages():
        # used extend since json would break for each page
        pageJson.extend(p)
```

```
33
34      for user in pageJson:
35          data[user.screen_name] = user.friends_count
36
37      data["phonedude_mln"] = len(pageJson)
38      writeCSV(data,"output/twitterFollowing.csv")
39
40
41  def writeCSV(data, filename):
42      with open(filename, 'w', newline='') as file:
43          for f, count in data.items():
44              writer = csv.writer(file, delimiter=',')
45              row = [f, count]
46              writer.writerow(row)
47
48
49  if __name__ == "__main__":
50      auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
51      auth.set_access_token(access_token, access_token_secret)
52      api = tweepy.API(auth, wait_on_rate_limit=True,
53                       wait_on_rate_limit_notify=True)
54      try:
55          getFollowers(api)
56          getFollowing(api)
57      except KeyboardInterrupt:
58          print()
```

Listing 2: Python script for receiving twitter followers and friends from Dr. Nelson's twitter

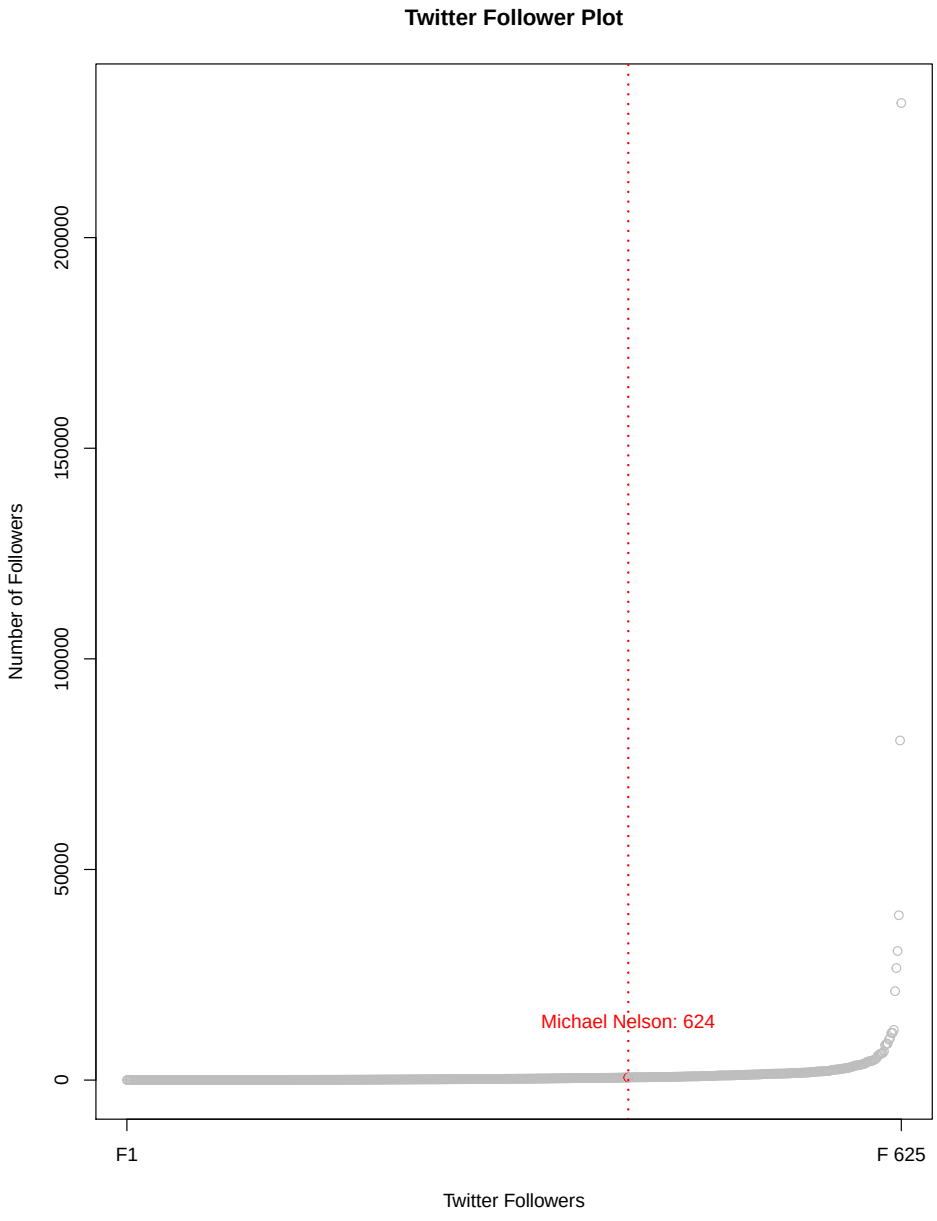

Figure 2: Plot of Dr. Nelson's Twitter followers vs. follower counts

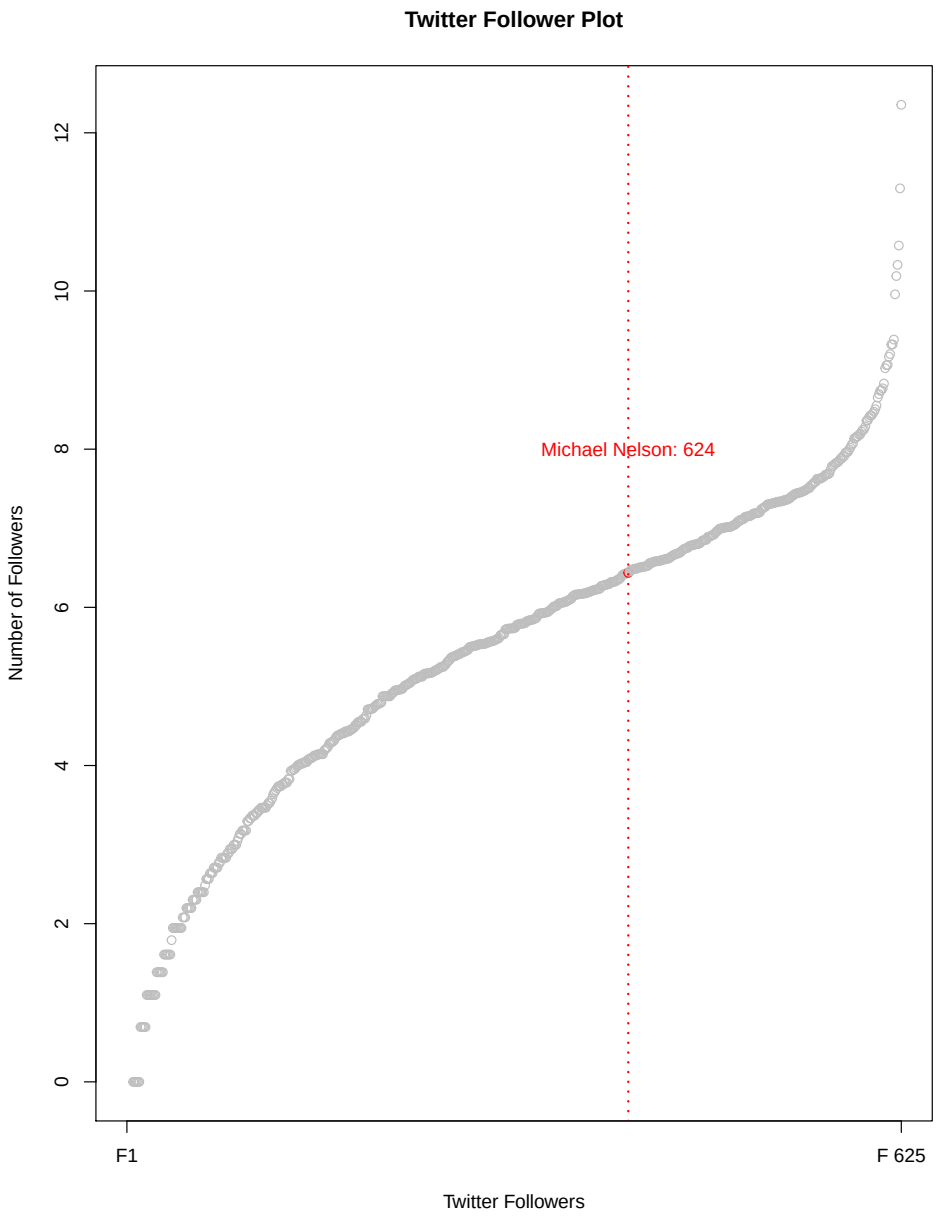

Figure 3: Natural log plot of Dr. Nelson's Twitter followers vs. follower counts

3

Question

```
Extra credit, 1 point:

5.  Repeat question #2, but change "followers" to "following"?  In
other words, are the people I am following following more people?
```

Answer

The Twitter API labels the connection between a user following another user a "friend" which just means I had to switch a value in my **twitterFriendship.py** script to check for friends rather than followers, this is shown above in Listing 2, which is the same script as earlier just with an extra function [3]. I retrieved this data and save the names of the users and their follower counts to a CSV file named **twitterFollowing.csv**. For this problem I used practically the same R script, **twitterFollowing.R**, from problem 2 as shown in Listing 3. This time I just interchanged the file names being used for data, as well as the labels. The CSV file is again read into the script taking both twitter usernames but this time with count of number of people they are following.

Again this graph shows that Dr. Nelson is following people who also seem to follow more people than him, showing that the Friendship paradox is also true in this case as well.

Mean	Standard Deviation	Median
853.816	4868.175	256

Table 3: Mean, Standard Deviation and Median generated from R Script for twitter following count

```
1  setwd(getwd())
2
3  csv <- read.table('output/twitterFollowing.csv',header = FALSE,
       sep = ",")
4  numRows <- nrow(csv)
5  ordered <- csv[order(csv$V2),]
6  cols <- c("gray", "red")[(ordered$V1 == "phonedude_mln")+1]
7  # index position of 'Michael Nelson' in data table
8  nelsonIndex <- which(ordered$V1=="phonedude_mln")
9  count <- ordered[nelsonIndex,]
10
11 b <- plot(ordered$V2,col = cols,main="Twitter Following/Friends
       Plot",xlab = "Twitter Users Followed",ylab = "Following Count
       ",xaxt='n')
12 abline(v=nelsonIndex,col="red",lwd=2,lty=3)
13
14 # Add f1-f_n on x-axis
15 axis(1, at=c(1,numRows),
16       lab=c("F1",paste("F",numRows)))
17 # y axis is just for padding
18 text(nelsonIndex,5000,paste("Michael Nelson:",count$V2),col = "
       red")
```

```
19 # Summary Stats
20 print("MEAN, SD, MEDIAN")
21 print(mean(ordered$V2))
22 print(sd(ordered$V2))
23 print(median(ordered$V2))
```

Listing 3: R Script for generating plot of twitter followers

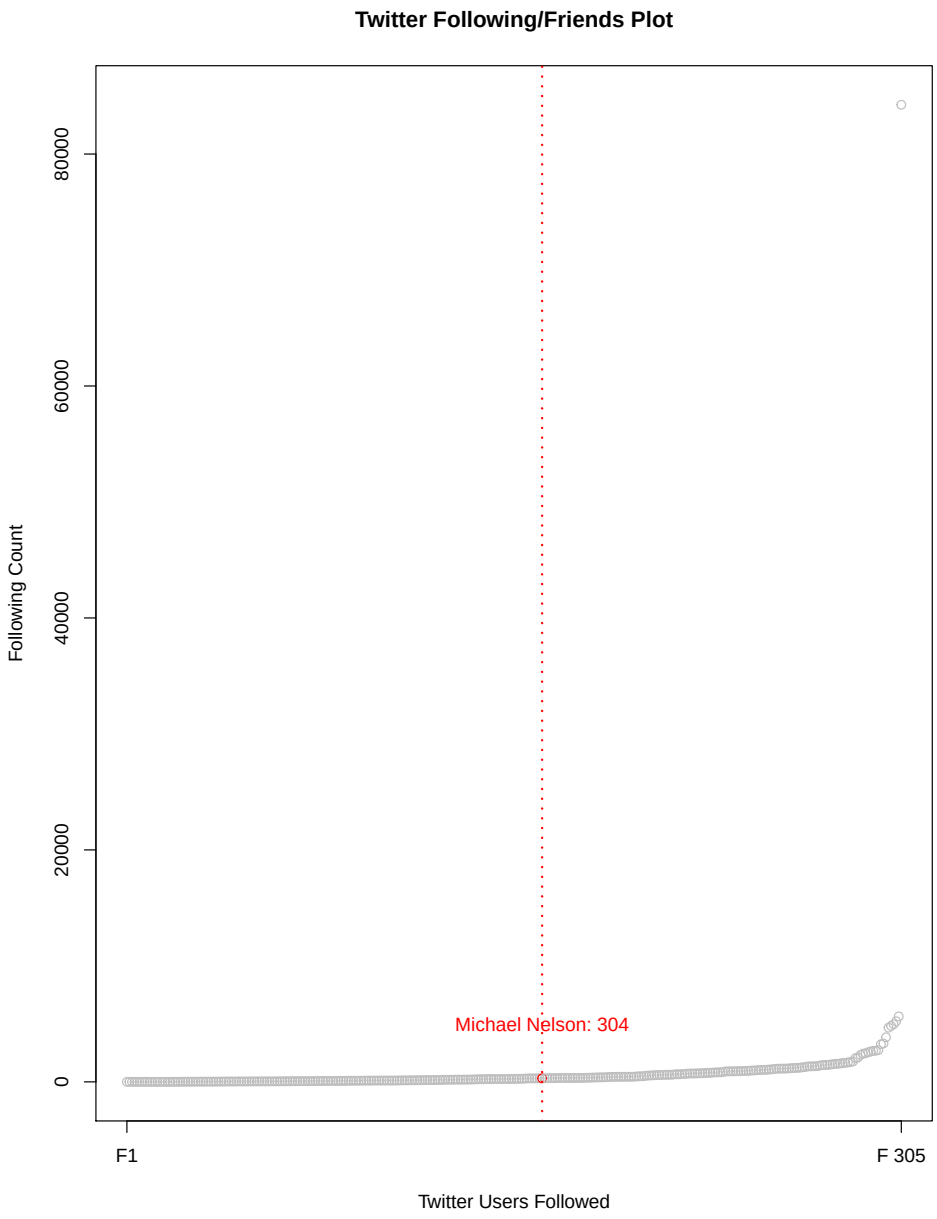

Figure 4: Plot of Dr. Nelson's Facebook Twitter following vs. following counts

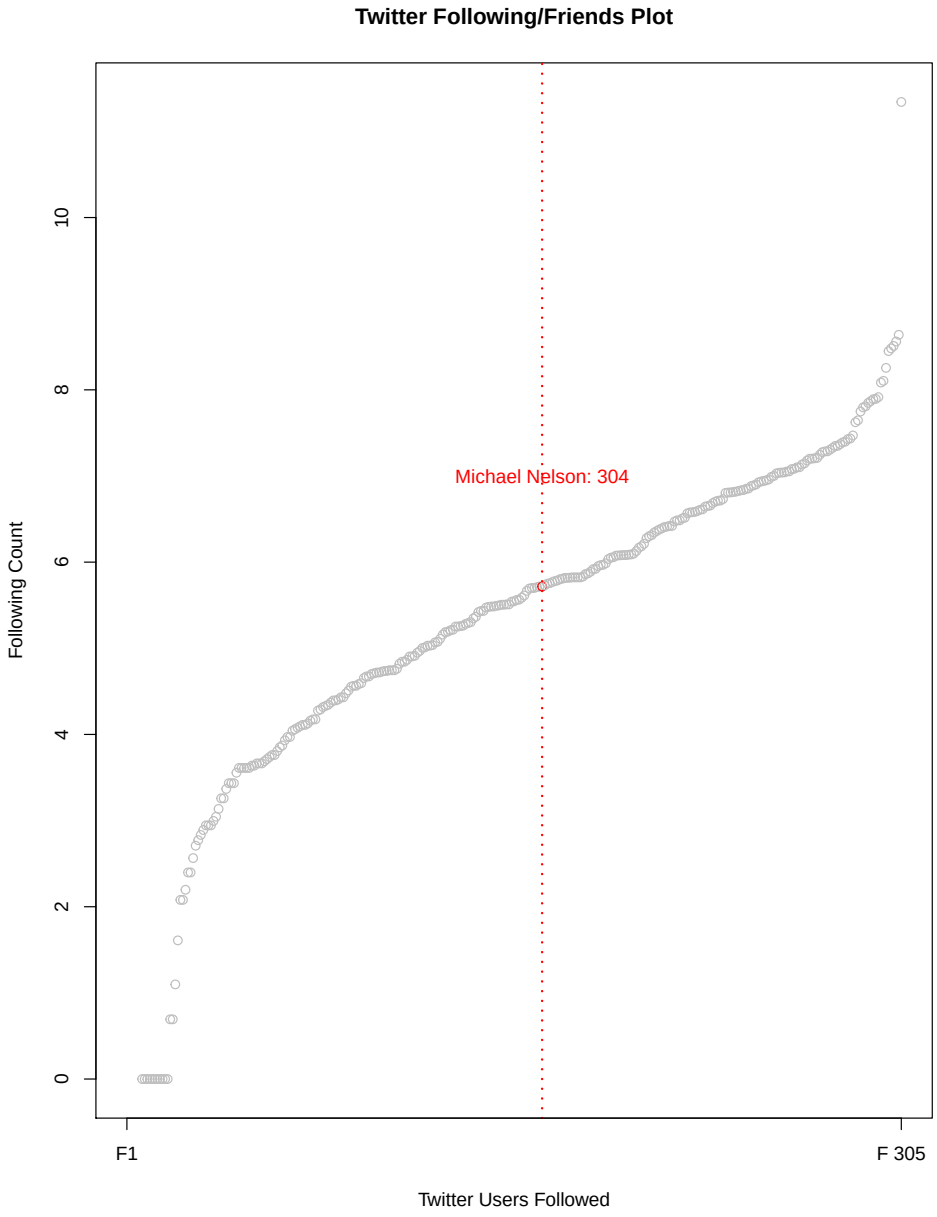

Figure 5: Natural log plot of Dr. Nelson's Facebook Twitter following vs. following counts

References

[1] Nelson, Michael. "Facebook Friends GraphML." cs532-s17 Github Repository. N.p., 1 March. 2017. Web. 1 March 2017.https://github.com/grantat/cs532-s17/blob/master/assignments/A3/src/output/mln.graphml.

[2] Mary, Hadrien. "pygraphml API documentation."N.p., n.d. Web. 1 March 2017 http://hadim.fr/pygraphml/reference.html.

[3] "Twitter Developer Documentation". Twitter. Twitter, n.d. Web. 1 March 2017. https://dev.twitter.com/rest/reference/get/followers/list.