

Assignment 4

CS 532: Introduction to Web Science

Spring 2018

Hrishi Gadkari

1

Question

1. 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 (y-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 upload a csv file of my 2014 friends list on the #assignment-4 slack channel

Answer

For the above problem, I wrote a script in R [3]. The script first extracts the **FRIENDSCOUNT** column from the **friendscount.csv** file and sorts the values in ascending order. It then calculates the mean, median and standard deviation of Alexander Nwalas Facebook friend counts as shown in Table 1. The references used for this calculation were [4] and [5].

Mean	Standard Deviation	Median
558.3176	571.784	397

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

The R script as shown in Listing 1 is stored in **facebook.r** file. The graph plotted as shown in Figure 1 tells us that Alexander Nwala has many friends which have higher friends count than him. This leads us to the conclusion that the friendship paradox holds for Alexander Nwalas Facebook friends.

```
1 setwd(getwd())
2 #getting values from csvfile
3 csv <- read.table('friendscount.csv', header = TRUE, sep = ",")
4
5 #sorting the values in ascending orders
6 y <- sort(csv$FRIENDSCOUNT)
7
8 #calculating mean
9 result.mean <- mean(y)
10 print(result.mean)
11
12 #calculating median
13 result.median <- median(y)
14 print(result.median)
15
16 #calculating standard deviation
17 result.sd <- sd(y)
18 print(result.sd)
19
20 #plotting the graph
21 plot(y, xlab="Friends", ylab="FriendsCount", col = "blue", type
      = "l")
22 title(main = "Facebook Friendship Paradox")
23
24 #setting up the text
25 text(10, 98, "x", col = 'red', cex=0.8)
26 text(55.5, result.mean, "x", col = 'red', cex = 0.8) # mean
27 text(41, result.median, "x", col = 'red', cex = 0.8) # median
```

```

28 | text(57, result.sd, "x", col = 'red', cex = 0.8) # standard
    | deviation
29 | text(41, 300, "Median: 397", cex = 0.8)
30 | text(62, 450, "Mean: 558.3176", cex = 0.8)
31 | text(66, 620, "SD: 571.784", cex = 0.8)
32 | text(24, 40, "Alexander's friends: 98", cex = 0.8)

```

Listing 1: R Script for generating plot of facebook friends

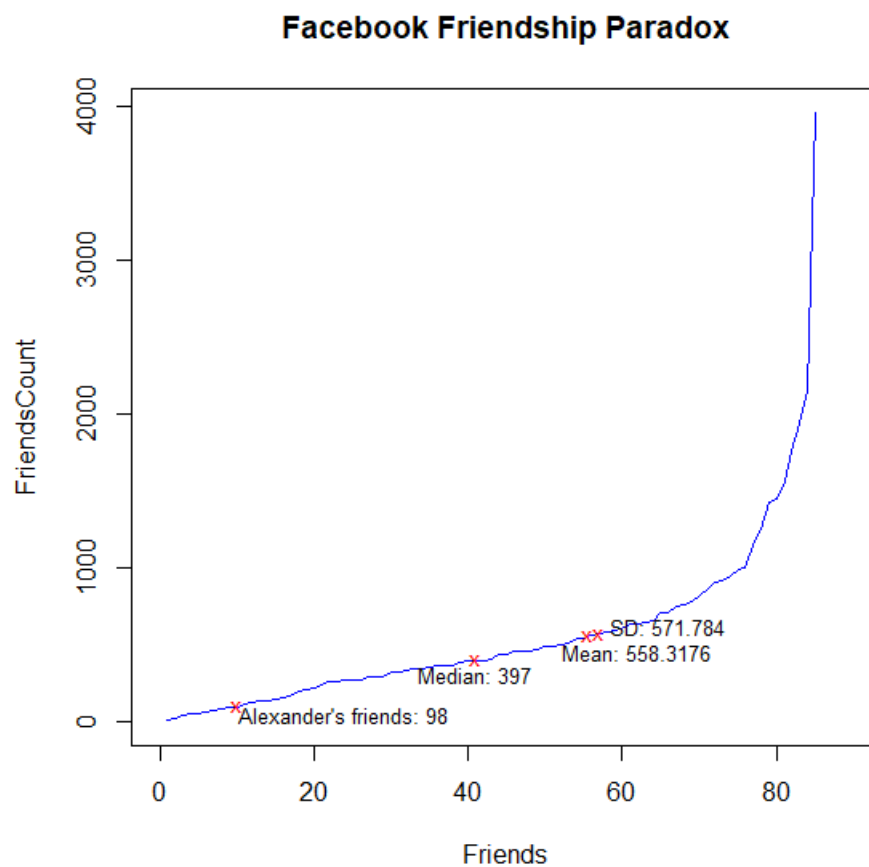


Figure 1: Plot of friends vs. friend counts

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 calculate the same mean, standard deviation, and median values.

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

<https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api->

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

Answer

Since my twitter account had followers less than 50, I user Alexander Nwala's twitter account. For the above problem I first went through the reference [1] mentioned in the question and came to know that in order to access the twitter followers API and get the followers list, I first need to register an application and get the keys and access tokens so as to access the API. To solve the problem I wrote a script in Python 3.5 as shown in Listing 2. The following dependencies were used:

- import requests
- import tweepy
- from tweepy import OAuthHandler
- import csv

To get the followers list for the screen_name: **acnwala** I went through the reference [7] and used the `tweepy.API()` and `tweepy.Cursor()` function. The program then gets the followers name and his followers counts one by one for Alexander Nwalas Twitter followers and stores in **followers.csv** file.

```
1 import requests
2 import tweepy
3 from tweepy import OAuthHandler
4 import csv
5
6 #get keys from: https://apps.twitter.com/
7 #consumer key, consumer secret, access token, access secret.
8 ckey = 'bSeaweiw68Hma0VLyeEd9se9u'
9 csecret = 'jch6kXwJociEynIHD0C8OunYLSYeRDCCjkaz0EUf3CSHzrNpSd'
10 atoken = '958819771000205312-w7L1GrIudQbONzjpfMRbwD33ITfWxnB'
11 asecret = 'MKE2Au1XVZDg1xV1F4USZsuIETm7WxgEuLACbDiQooxHG'
12
13 # csv file open to write headers
14 with open('followers.csv', 'a+', newline='') as csvfile:
15     fieldnames = ['Users', 'FollowersCount']
16     writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
17     writer.writeheader()
18 csvfile.close()
19
20 #OAuth authorization
21 auth = OAuthHandler(ckey, csecret)
22
23 auth.set_access_token(atoken, asecret)
24 #getting api access
```

```

25 api = tweepy.API(auth, wait_on_rate_limit=True,
    wait_on_rate_limit_notify=True, retry_count=3, retry_delay
    =60)
26
27 users=[]
28
29 #accessing followers api
30 for user in tweepy.Cursor(api.followers, screen_name='acnwala',
    count=200).pages():
31     users.extend(user)
32     for i in users:
33         count = i.followers_count
34         foll=i.screen_name
35 #writing output to csvfile
36         with open('followers.csv', 'a+', newline='') as
            csvfile:
37             writer = csv.DictWriter(csvfile,
                fieldnames=fieldnames)
38             writer.writerow({'Users': foll, '
                FollowersCount': count})
39             csvfile.close()

```

Listing 2: Python script for receiving twitter followers from Alexander Nwala's twitter

To plot the graph I wrote a script in R [3]. The script first extracts the **FollowersCount** column from the **followers.csv** file and sorts the values in ascending order. It then calculates the mean, median and standard deviation of Alexander Nwalas Twitter followers as shown in Table 2. The references used for this calculation were [4] and [5].

Mean	Standard Deviation	Median
2967.856	14353.86	274

Table 2: Mean, Standard Deviation and Median generated from R Script for Twitter follower counts

The R script as shown in Listing 3 is stored in **tweetfor.r** file

```

1 setwd(getwd())
2 #reading data from csv
3 csv <- read.table('followers.csv', header = TRUE, sep = ",")
4
5 #sort values in ascending order
6 y <- sort(csv$FollowersCount)
7
8 #calculating mean
9 result.mean <- mean(y)

```

```

10 print(result.mean)
11
12 #calculating median
13 result.median <- median(y)
14 print(result.median)
15
16 #calculating standard deviation
17 result.sd <- sd(y)
18 print(result.sd)
19
20 #plotting graph
21 plot(y, xlab="Followers", ylab="FollowersCount", col = "blue",
      type = "l")
22 title(main = "Twitter Followers Paradox")
23
24 #setttting up text
25 text(65, 194, "x", col = 'red', cex=0.8)
26 text(175, result.mean, "x", col = 'red', cex = 0.8) # mean
27 text(82, result.median, "x", col = 'red', cex = 0.8) # median
28 text(188, result.sd, "x", col = 'red', cex = 0.8) # standard
   deviation
29 text(92, 6000, "Median: 274", cex = 0.8)
30 text(170, -50, "Mean: 2967.856", cex = 0.8)
31 text(165, 14353, "SD: 14353.86", cex = 0.8)
32 text(45, 6000, "AN's followers: 194", cex = 0.8)

```

Listing 3: R Script for generating plot of twitter followers

The graph plotted as shown in Figure 2 tells us that Alexander Nwala has many followers which have higher followers count than him. This leads us to the conclusion that the followers paradox holds for Alexander Nwalas Twitter followers.

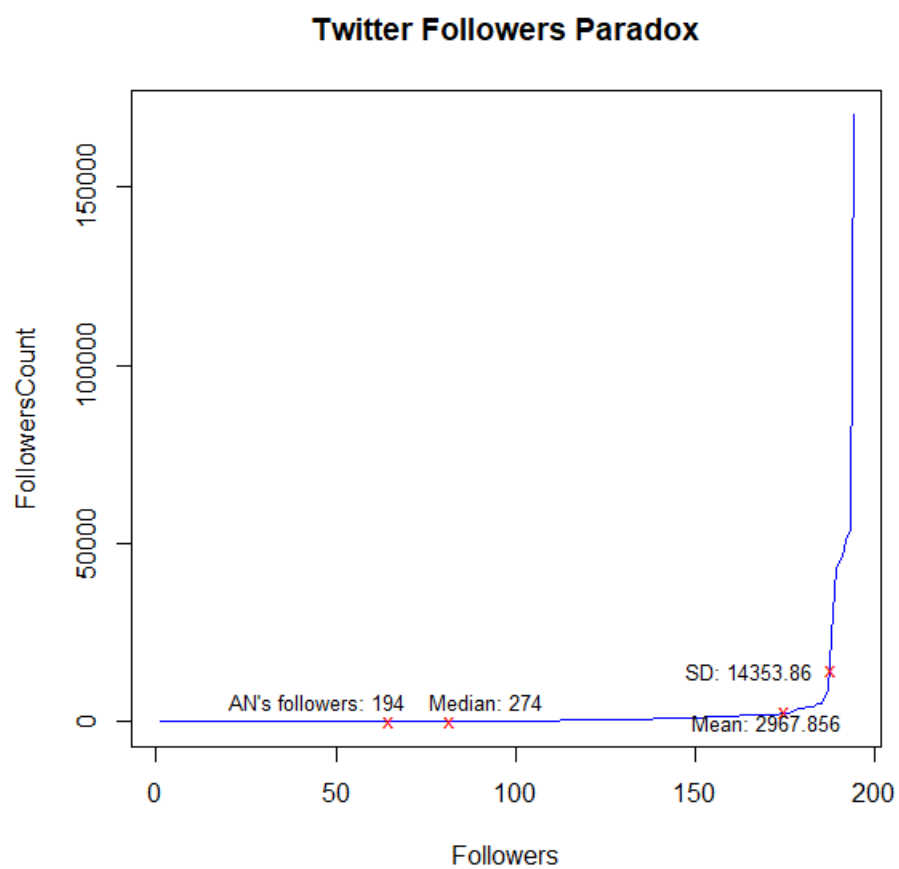


Figure 2: Plot of Alexander Nwala's Twitter followers vs. follower counts

3

Question

Extra credit, 1 point:

3. Repeat question #1, but with your (or a specified) LinkedIn profile.

Answer

For the above problem I went through the github documentation [6] of LinkedIn API. I first installed the linkedin python library using **pip install python3-linkedin**. I wrote a program in Python 3.5 as shown in Listing 4. The following dependencies were used:

- from linkedin import linkedin

The program gets the access tokens used for accessing the API and getting the profile for each user through OAuth by using id, secret key and return url. But after running the program it just returned the basic profile of my linked in profile giving the first name, last name and number of connections whereas access was denied for the `get_connections()` and `search_profile()`. I then googled and got to know that as of May 2017 [8], Linked in have restricted its full API access and can be accessed only through special permissions. As per the Figure in 3, we can see that its returning just the number of connections.

```
1 from linkedin import linkedin
2
3 APIKEY = '86kugk5q9awmxj'
4 APLSECRET = 'CUZoPYntqc3SEcRP'
5 RETURN_URL = 'http://www.cs.odu.edu/~anwala/'
6
7 authentication = linkedin.LinkedInAuthentication(APIKEY,
8           APLSECRET, RETURN_URL)
9 # Optionally one can send custom "state" value that will be
10 # returned from OAuth server
11 # It can be used to track your user state or something else (it's
12 # up to you)
13 # Be aware that this value is sent to OAuth server AS IS – make
14 # sure to encode or hash it
15 #authorization.state = 'your_encoded_message'
16 authentication.authorization_url # open this url on your
17 # browser
18 #Authentication
19 application = linkedin.LinkedInApplication(authentication)
20 authentication.authorization_code = '
21     AQQWMhrgDvMWS8zAKr4uApwt81fFqqvJA1ioRxbA1JMuKTaJYdrtnV2-
22     AUfSt8a_xI9AME14ewCrcGowNRwSLQaQXoP-kzzC3eaa-
23     wTHcMk0ylezMLZSjZFIVN7YNLn04547B6Jl4UFfQbk-uiLTNZAtpWTig'
24 #get access token
25 authentication.get_access_token()
26
27 #set access token
```

```

20 application = linkedin.LinkedInApplication(token=authentication.
    get_access_token())
21 #accessing basic profile api
22 print(application.get_profile(selectors=['num-connections']))
23 #accessing connections api
24 print(application.get_connections())
25 #accessing search profile api
26 print(application.search_profile(selectors=[{'people': ['first -
    name', 'last-name']}], params={'keywords': 'apple microsoft
    '}))

```

Listing 4: Python script for LinkedIn Profile API

```

Anaconda Prompt
(python3) C:\Users\GADKARI\Documents\Python Scripts\web_science\api\python link.py
('numConnections': 23)
Traceback (most recent call last):
  File "C:\Users\GADKARI\Anaconda3\lib\site-packages\linkedin\utils.py", line 49, in raise_for_error
    response.raise_for_status()
  File "C:\Users\GADKARI\Anaconda3\lib\site-packages\requests\models.py", line 935, in raise_for_status
    raise HTTPError(http_error_msg, response=self)
requests.exceptions.HTTPError: 403 Client Error: Forbidden for url: https://api.linkedin.com/v1/people/~connections?oauth2_access_token=AQUF73tCwJMaas8vU75ALR68muvR53qGq5Alu5n3lgoe_HB00va3Bk75CNVZlGw-3j1dJ1h1t-8dQeVvAhv
l4wctm1bqpmH4edQm8eav15SR5t52uam-BQ9thAhly8Q28u8P8F7C7hZu6d1d0d4v-H3z2p3m-ztBUqFy8V51AUW1x8_BuMqzyveemle87Neu1hMUR5k8GT1MeF3zanP8dUT0dFwLdWd8p8RagC6eHqQdAhDvZX8OT0w5xnnlCcl1vct3dmoFCDfNvUku
wQq4e164K2H8C9tmXUmkz7Lv5N7M-XkXbryV7W4sA2ig

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "link.py", line 22, in <module>
    print(application.get_connections())
  File "C:\Users\GADKARI\Anaconda3\lib\site-packages\linkedin\linkedin.py", line 250, in get_connections
    raise_for_error(response)
  File "C:\Users\GADKARI\Anaconda3\lib\site-packages\linkedin\utils.py", line 62, in raise_for_error
    raise ex(message)
linkedin.exceptions.LinkedinForbiddenError: 403 Client Error: Forbidden for url: https://api.linkedin.com/v1/people/~connections?oauth2_access_token=AQUF73tCwJMaas8vU75ALR68muvR53qGq5Alu5n3lgoe_HB00va3Bk75CNVZlGw-3j1dJ1
h1t-8dQeVvAhv14wctm1bqpmH4edQm8eav15SR5t52uam-BQ9thAhly8Q28u8P8F7C7hZu6d1d0d4v-H3z2p3m-ztBUqFy8V51AUW1x8_BuMqzyveemle87Neu1hMUR5k8GT1MeF3zanP8dUT0dFwLdWd8p8RagC6eHqQdAhDvZX8OT0w5xnnlCcl1vct3dmoFCDfNvUku
wQq4e164K2H8C9tmXUmkz7Lv5N7M-XkXbryV7W4sA2ig: Access to connections denied

(python3) C:\Users\GADKARI\Documents\Python Scripts\web_science\api>

```

Figure 3: Output for LinkedIn API

4

Question

Extra credit, 3 points:

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

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

<https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-friends-list>

Answer

For the above problem, I repeated the same steps as mentioned in question 2. I first went through the reference [2] mentioned in the question and came to know that in order to access the twitter's following/friends API and get the followings list, I first need to register an application and get the keys and access tokens so as to access the API. To solve the problem I wrote a script in Python 3.5 as shown in Listing 5. The following dependencies were used:

- import requests
- import tweepy
- from tweepy import OAuthHandler
- import csv

To get the followings list for the screen_name: **acnwala** I went through the reference [7] and used the tweepy.API() and tweepy.Cursor() function. The program then gets the friends name and his following counts one by one for Alexander Nwalas Twitter followings and stores in **following.csv** file.

```
1 import requests
2 import tweepy
3 from tweepy import OAuthHandler
4 import csv
5
6 #get keys from: https://apps.twitter.com/
7 #consumer key, consumer secret, access token, access secret.
8 ckey = 'bSeaweiw68Hma0VLyeEd9se9u'
9 csecret = 'jch6kXwJociEynIHD0C8OunYLSYeRDCCjkaz0EUf3CSHzrNpSd'
10 atoken = '958819771000205312-w7L1GrIudQbONzjpfMRbwD33ITfWxnB'
11 asecret = 'MKE2Au1XVZDg1xV1F4USZsuIETm7WxgEuLACbDiQooxHG'
12
13 # csv file open to write headers
14 with open('followers.csv', 'a+', newline='') as csvfile:
15     fieldnames = ['Users', 'FollowersCount']
16     writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
17     writer.writeheader()
18 csvfile.close()
19
20 #OAuth authorization
21 auth = OAuthHandler(ckey, csecret)
22
23 auth.set_access_token(atoken, asecret)
24 #getting api access
```

```

25 api = tweepy.API(auth, wait_on_rate_limit=True,
    wait_on_rate_limit_notify=True, retry_count=3, retry_delay
    =60)
26
27 users=[]
28
29 #accessing followers api
30 for user in tweepy.Cursor(api.followers, screen_name='acnwala',
    count=200).pages():
31     users.extend(user)
32     for i in users:
33         count = i.followers_count
34         foll=i.screen_name
35 #writing output to csvfile
36         with open('followers.csv', 'a+', newline='') as
            csvfile:
37             writer = csv.DictWriter(csvfile,
                fieldnames=fieldnames)
38             writer.writerow({'Users': foll, '
                FollowersCount': count})
39             csvfile.close()

```

Listing 5: Python script for receiving twitter followers from Alexander Nwala's twitter

To plot the graph I wrote a script in R [3]. The script first extracts the **FollowingCount** column from the **following.csv** file and sorts the values in ascending order. It then calculates the mean, median and standard deviation of Alexander Nwalas Twitter friends as shown in Table 3. The references used for this calculation were [4] and [5].

Mean	Standard Deviation	Median
1032.158	1549.282	480.5

Table 3: Mean, Standard Deviation and Median generated from R Script for Twitter following counts

The R script as shown in Listing 6 is stored in **tweetfol.r** file

```

1 setwd(getwd())
2 #reading data from csv
3 csv <- read.table('following.csv', header = TRUE, sep = ",")
4
5 #sort values in ascending order
6 y <- sort(csv$FollowingCount)
7
8 #calculating mean

```

```

9 | result.mean <- mean(y)
10 | print(result.mean)
11 |
12 | #calculating median
13 | result.median <- median(y)
14 | print(result.median)
15 |
16 | #calculating standard deviation
17 | result.sd <- sd(y)
18 | print(result.sd)
19 |
20 | #plotting the graph
21 | plot(y, xlab="Following", ylab="FollowingCount", col = "blue",
      |       xlim=c(0, 80), type = "l")
22 | title(main = "Twitter Following Paradox")
23 |
24 | #setting up the text
25 | text(10, 76, "x", col = 'red', cex=0.8)
26 | text(56, result.mean, "x", col = 'red', cex = 0.8) # mean
27 | text(38, result.median, "x", col = 'red', cex = 0.8) # median
28 | text(61.7, result.sd, "x", col = 'red', cex = 0.8) # standard
      | deviation
29 | text(38, 900, "Median: 480.5", cex = 0.8)
30 | text(56, 600, "Mean: 1032.158", cex = 0.8)
31 | text(70, 1449, "SD: 1549.282", cex = 0.8)
32 | text(10, 400, "AN Following: 76", cex = 0.8)

```

Listing 6: R Script for generating plot of twitter followers

The graph plotted as shown in Figure 4 tells us that Alexander Nwala has many twitter friends which have higher followings count than him. This leads us to the conclusion that the following paradox holds for Alexander Nwalas Twitter followers.

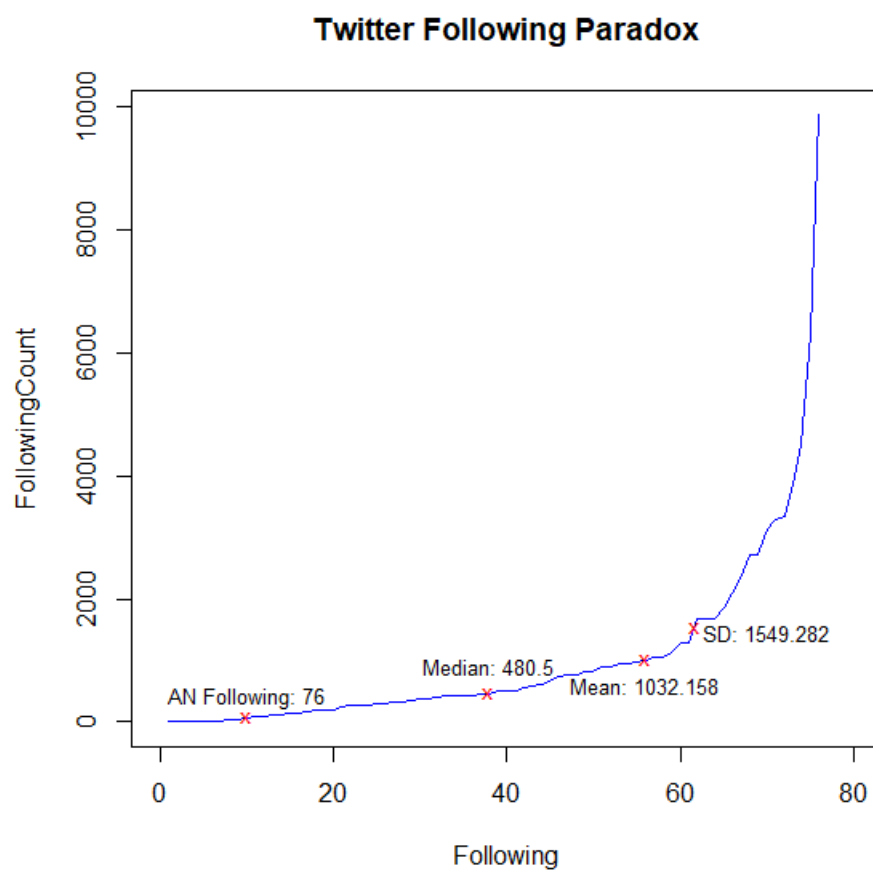


Figure 4: Plot of Alexander Nwala's Twitter followings vs. following counts

References

- [1] GET followers/list - Twitter Developers. “Twitter.” N.p., n.d. 27 February 2018.<https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-list>.
- [2] GET friends/list - Twitter Developers. “Twitter.” N.p., n.d. 27 February 2018.<https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-list>.
- [3] Search all 14,436 CRAN, BioConductor and Github packages. “R Documentation and manuals — R Documentation.” N.p., n.d. Web. 27 February 2018 <https://www.rdocumentation.org/>.
- [4] base. “function — R Documentation”., n.d. Web. 27 February 2018. <https://www.rdocumentation.org/packages/base/versions/3.4.3/topics/sort>.
- [5] How to calculate mean, median, mode, std dev from distribution. “r - How to calculate mean, median, mode, std dev from distribution - Cross Validated.” N.p., n.d. Web. 27 February 2018 <https://stats.stackexchange.com/questions/157661/how-to-calculate-mean-median-mode-std-dev-from-distribution>.
- [6] Ozgur. “Ozgur/python-linkedin.”, Github, 26 June 2015, n.d. Web. 27 February 2018 <https://github.com/ozgur/python-linkedin>.
- [7] Get all followers and friends of a Twitter user. “python - Get all followers and friends of a Twitter user - Code Review Stack Exchange.” N.p., n.d. Web. 27 February 2018 <https://codereview.stackexchange.com/questions/101905/get-all-followers-and-friends-of-a-twitter-user>.
- [8] Developer Program Transition — LinkedIn . “LinkedIn Developers.” N.p., n.d. Web. 27 February 2018 <https://developer.linkedin.com/support/developer-program-transition>.