Assignment 4

CS 532: Introduction to Web Science

Spring 2018

Chandrasekhar Reddy Muthyala

Finished on February 28, 2018

1

Question

```
The "friendship paradox" (http://en.wikipedia.org/wiki/Friendship_paradox)
says that your friends have more friends than you do.

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

To solve this problem I have gone through the above link and understood what friendship paradox is, and started writing a programming in python **friendsParadox.py** for displaying friendship paradox graph for Prof. anwalas Facebook account. Prof. anwala gave his facebook friends of friends count in **acnwala-friendscount.csv**.
List of python dependency libraries

- import matplotlib.pyplot as plt
- import pandas as pd
- import numpy as np
- import math

Initially in the program I am reading a csv using pandas and it will convert into data frame. From that data frame I took friends count column and converted in to list **y_axis** that acts as y-axis. Using that list length I have created a **x_axis** list to indicate number of friends. First Using numpy library I calculated mean, median and standard deviation (SD) and appending into y_axis list to plot the mark on the graph. I took **myFriends** variable which will store prof. Anwals friends count and appending into y_axis list just like mean, median and SD to plot where Prof. anwala stands in the graph whether above or below mean and median values.

Run on the command line

```
python friendsParadox.py
```

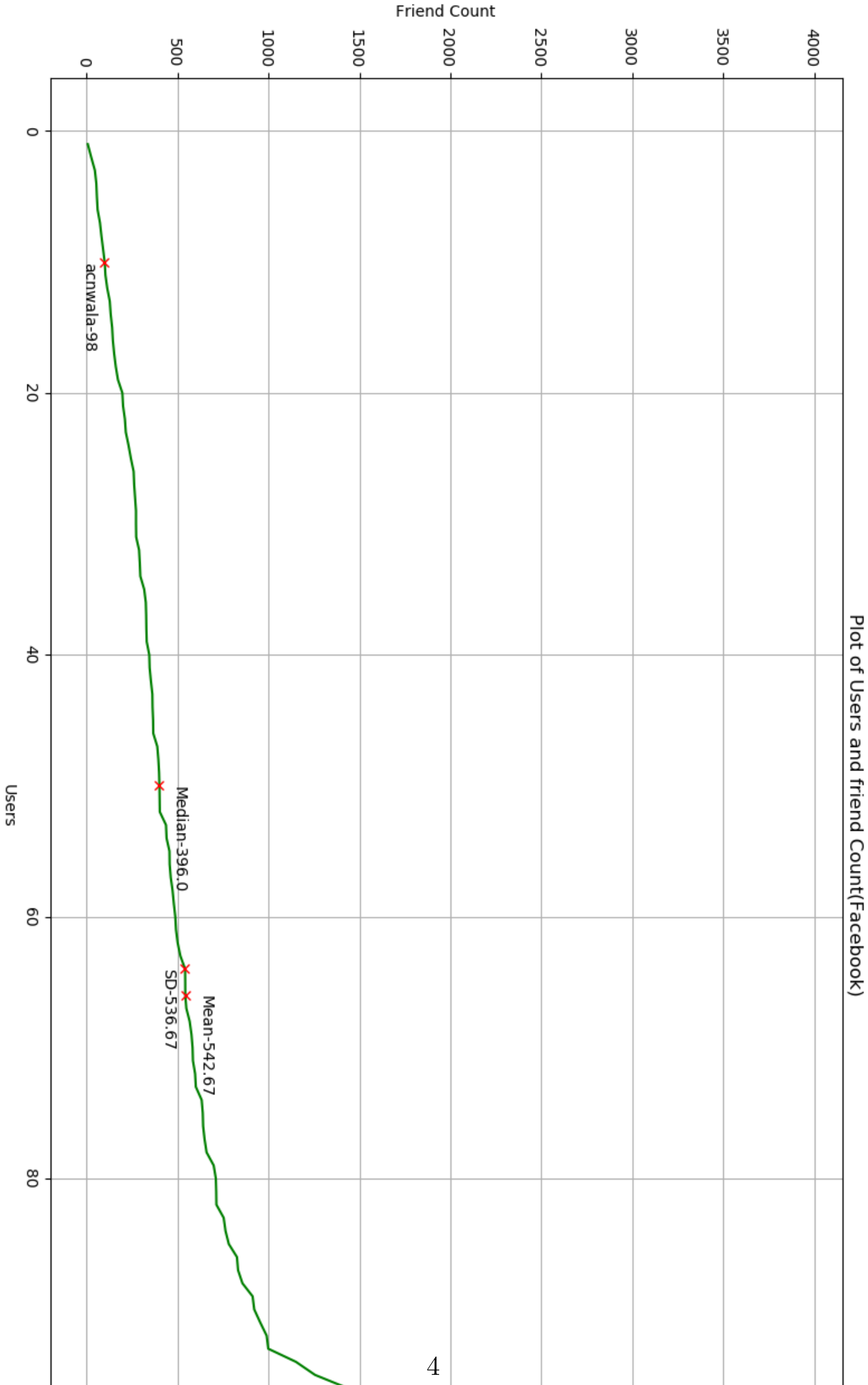
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
data = pd.read_csv('acnwala-friendscount.csv', skiprows=1, sep=',',header=None)
y_axis = data[1].values.tolist()
# my friends count =980
myFriends = 98
mean  = round(np.mean(y_axis),2)
median = round(np.median(y_axis),2)
sd = round(np.std(y_axis),2)
y_axis.append(mean)
y_axis.append(median)
y_axis.append(sd)
y_axis.append(myFriends)
y_axis.sort()


# print(myFriends)

x_axis = [i for i in range(1, len(y_axis)+1)]

plt.plot(x_axis, y_axis, color='g' )
plt.xlabel('Users')
plt.ylabel('Friend Count')
plt.plot(y_axis.index(mean),mean, marker='x', color='r')
plt.text(y_axis.index(mean),mean+100,"Mean-"+str(mean))

plt.plot(y_axis.index(median),median, marker='x', color='r')
plt.text(y_axis.index(median),median+100,"Median-"+str(median))

plt.plot(y_axis.index(sd),sd, marker='x', color='r')
plt.text(y_axis.index(sd),sd-100,"SD-"+str(sd))

plt.plot(y_axis.index(myFriends),myFriends, marker='x', color='r')
plt.text(y_axis.index(myFriends),myFriends-100,"acnwala-98")

plt.title('Plot of Users and friend Count(Facebook)')
# plt.xticks(x_axis)
plt.grid(True)
plt.show()
```

Listing 1: Python script for ploting friends paradox for Facebook account

In the plot shown in below Figure , you can see that Prof.Anwala has many friends with higher friend counts than him, with his count being 98 (not including himself). Therefore, the friendship paradox does hold for Prof.Anwala's Facebook account. I also used this Python script to compute the Mean, Standard Deviation and Median of Prof.Anwala's Facebook friend counts shown in Table.

Mean	**Standard Deviation**	**Median**
542.67	536.67	396.0

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


Plot of Users and friend Count(Facebook)
Friend Count
0
500
1000
1500
2000
2500
3000
3500
4000
Users
0
20
40
60
80
100
acnwala-98
Median-396.0
SD-536.67
Mean-542.67

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://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users

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

Answer

To solve this problem I have gone through the twitter API Tweepy and found one solution for getting followings and followers based on the screen name given to that API. There is a cursor function in tweepy library to get details of friends of friends in twitter. For this problem I am passing screen name and variable which I wanted to extracting from my friends account let say in this problem I want following count of my friends (that is whom I am following in twitter). I am saving screen name of my friends and their corresponding following count in **friendsCount.csv** file, like wise for followers of my followers I am passing followers and screen name to the cursor function which will give me count of my flowers of followers count ad storing in **followersCount.csv** file. I have collected data for problem#2 in **friendsCount.csv** and problem#4 in **followersCount.csv** to plot friendship paradox.
List of python dependency libraries to extract above data

- import tweepy
- import csv

Run on the command line

```
python friendsofFriendsFromTwitter.py
```

```
import tweepy
import csv
ckey=”v5YfpnUYDpYFrm9hn4wfV6hCw”
csecret=”2nOHkJyPqteSudHrlS8YWcsuK8wvX7AgIPVIK8ySHj3dU5TnF6”
atoken=”956029983469776897−GUROfj1hmnuImgewZAZrKAog5dJ1q3O”
asecret=”UipAVctbQJUQEczQ2s7Dd5J3teLa6ZwEAWWbaceTSSfca”
# OAuth process, using the keys and tokens
auth = tweepy.OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)

# Creation of the actual interface, using authentication
api = tweepy.API(auth, wait_on_rate_limit=True,
wait_on_rate_limit_notify=True, compression=True)
screenName = ”hemanthmalla”
myFriends = 0
myfollowers = 0
with open(’friendsCount.csv’, ’w’,newline=’’) as csvfriends:
  fieldnames = [’USER’, ’FRIENDCOUNT’]
  writerriends = csv.DictWriter(csvfriends, fieldnames=fieldnames)
  writerriends.writeheader()
  for friend in tweepy.Cursor(api.friends, screen_name=screenName).items():
    myFriends += 1
    writerriends.writerow({’USER’: friend.screen_name, ’FRIENDCOUNT’:friend.friends_count})
    print(friend.screen_name,”:”,friend.friends_count)
  writerriends.writerow({’USER’: screenName, ’FRIENDCOUNT’: myFriends})

with open(’followersCount.csv’, ’w’,newline=’’) as csvfollowers:
  fieldnames  = [’USER’, ’FOLLOWERSCOUNT’]
  writerfollowers = csv.DictWriter(csvfollowers, fieldnames=fieldnames)
  writerfollowers.writeheader()
  for follower in tweepy.Cursor(api.followers, screen_name=screenName).items():
    myfollowers +=1
    writerfollowers.writerow({’USER’: follower.screen_name, ’FOLLOWERSCOUNT’:follower.friends_count})
    print(follower.screen_name,”:”,follower.friends_count)
  writerfollowers.writerow({’USER’: screenName, ’FOLLOWERSCOUNT’: myfollowers})


print(”myFriends: ”, myFriends, ”myfollowers: ”,myfollowers)
```

Listing 2: Python script for extracting friends and followers of friends from twitter

To draw the friendship paradox of twitter friends I wrote a program just like problem#1 named as **twitterFriendsParadox.py**

Run on the command line

```
python twitterFriendsParadox.py
```

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
data = pd.read_csv(’friendsCount.csv’, skiprows=1, sep=’,’, header=None)
y_axis = data[1].values.tolist()
# my friends count =980
```

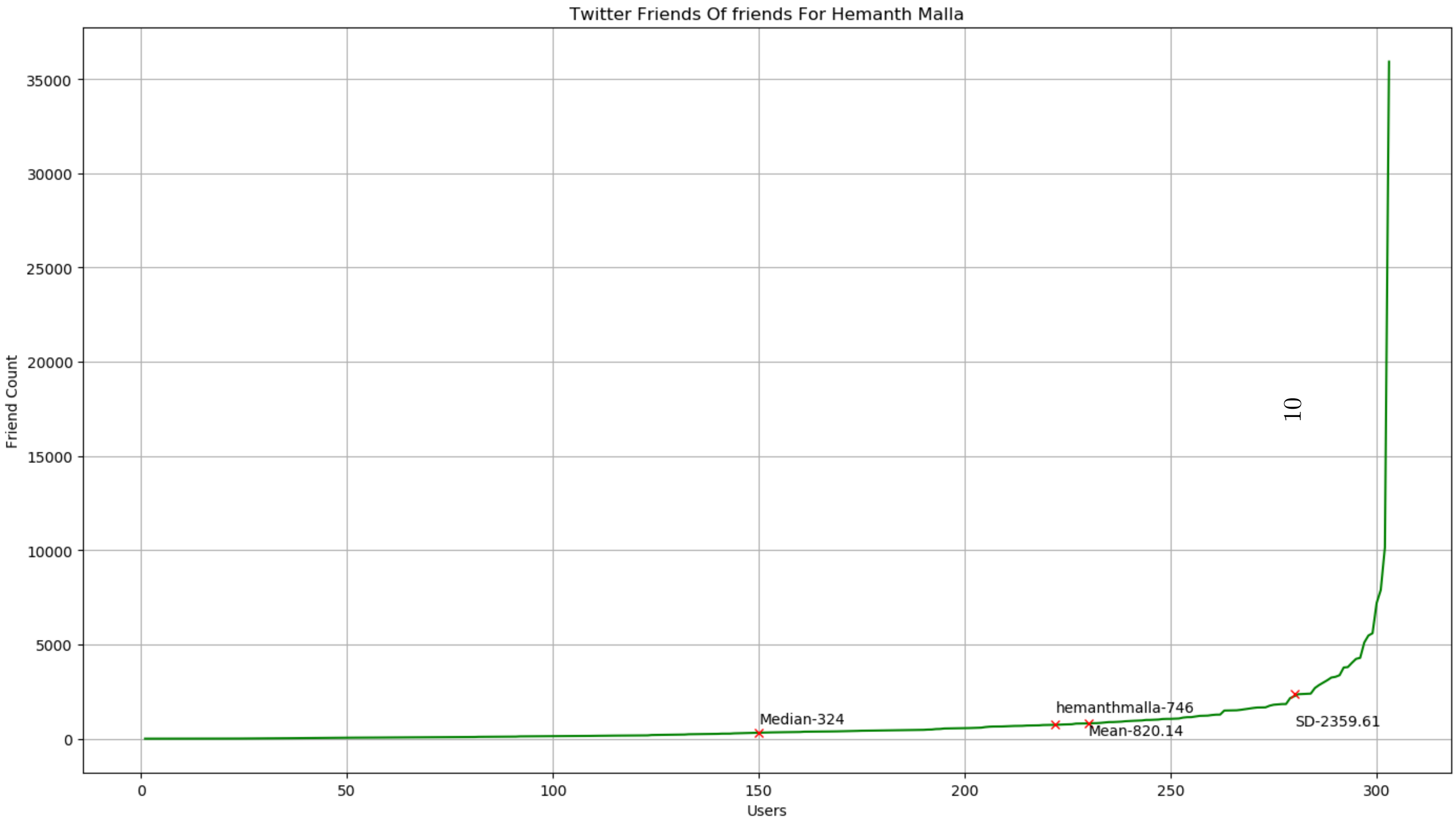
```
myFriends = 746
mean  = round(np.mean(y_axis),2)
median = round(np.median(y_axis),2)
sd = round(np.std(y_axis),2)
y_axis.append(mean)
y_axis.append(median)
y_axis.append(sd)
# y_axis.append(myFriends)
y_axis.sort()


# print(myFriends)

x_axis = [i for i in range(1, len(y_axis)+1)]

plt.plot(x_axis, y_axis, color=’g’ )
plt.xlabel(’Users’)
plt.ylabel(’Friend Count’)
plt.plot(y_axis.index(mean),mean, marker=’x’, color=’r’)
plt.text(y_axis.index(mean),mean−10000,”Mean–”+str(mean))

plt.plot(y_axis.index(median),median, marker=’x’, color=’r’)
plt.text(y_axis.index(median),median+5000,”Median–”+str(int(
median)))

plt.plot(y_axis.index(sd),sd, marker=’x’, color=’r’)
plt.text(y_axis.index(sd),sd−100,”SD–”+str(sd))

plt.plot(y_axis.index(myFriends),myFriends, marker=’x’, color=’r
’)
plt.text(y_axis.index(myFriends),myFriends+5000,”hemanthmalla
−746”)

plt.title(’Twitter Friends Of friends For Hemanth Malla’)
# plt.xticks(x_axis)
plt.grid(True)
plt.show()
```

Listing 3: Python script for generating a plot Twitter Friends Of friends Of Hemanth Malla

Mean	**Standard Deviation**	**Median**
820.14	2359.61	324

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

In the plot shown in below Figure , you can see that Hemanth malla has many friends with higher friend counts than him, with his count being 746 (not including himself). Therefore, the friendship paradox does hold for Hemanth malla's Twitter account.


Twitter Friends Of friends For Hemanth Malla
Friend Count
35000
30000
25000
20000
15000
10000
5000
0
Users
0
50
100
150
200
250
300
Median-324
hemanthmalla-746
Mean-820.14
SD-2359.61

3

Question

```
Extra credit, 1 point:

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

Answer

To solve this problem I have gone through LinkedIn api for developers documentation, after gone through the documentation I understood that developers account is to get the basic information of my own profile and it is not providing any information of my connections profile. There is REST Console to check the api, when I tested it on the console by hitting the below URL I got below json as a result.

```
https://api.linkedin.com/v1/people/~?format=json
```

```
{
  "firstName": "Chandra Sekhar Reddy",
  "headline": "Actively seeking Summer Internship.",
  "id": "ieRtV_6P-3",
  "lastName": "muthyala",
  "siteStandardProfileRequest": {"url": "https://www.linkedin.com/profile/view?id=AAoAABAxil4Bvywspn9jEWbxrqYnUoABFEL_rYE&authType=name&authToken=yJSt&trk=api*a3227641*s3301901*"}
}
```

4

Question

```
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/
```

Answer

To solve this problem, we got data for this problem from **friendsofFriendsFromTwitter.py** and stored in **followersCount.csv** file.

To draw the friendship paradox of twitter followers of my screen name followers I wrote a program just like problem#1 named as **twitterFriendsParadox.py**

Run on the command line

```
pythontwitterFollowersParadox.py
```

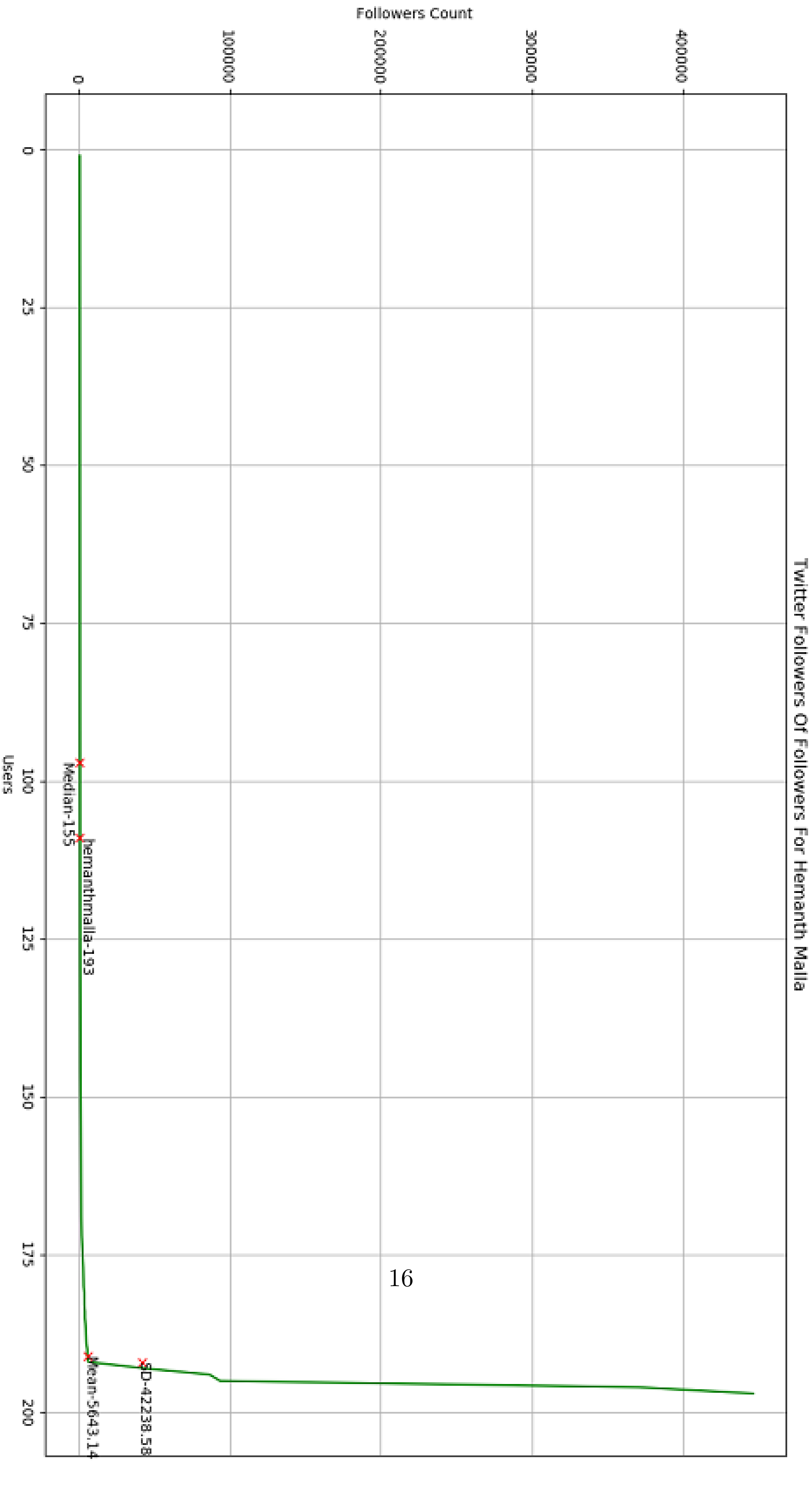
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import math
data = pd.read_csv('followersCount.csv', skiprows=1, sep=',', header=None)
y_axis = data[1].values.tolist()
# my friends count =980
myFriends = 193
mean  = round(np.mean(y_axis),2)
median = round(np.median(y_axis),2)
sd = round(np.std(y_axis),2)
y_axis.append(mean)
y_axis.append(median)
y_axis.append(sd)
# y_axis.append(myFriends)
y_axis.sort()


# print(myFriends)

x_axis = [i for i in range(1, len(y_axis)+1)]

plt.plot(x_axis, y_axis, color='g' )
plt.xlabel('Users')
plt.ylabel('Followers Count')
plt.plot(y_axis.index(mean),mean, marker='x', color='r')
plt.text(y_axis.index(mean),mean+500,"Mean-"+str(mean))

plt.plot(y_axis.index(median),median, marker='x', color='r')
plt.text(y_axis.index(median),median-10000,"Median-"+str(int(median)))

plt.plot(y_axis.index(sd),sd, marker='x', color='r')
plt.text(y_axis.index(sd),sd-1000,"SD-"+str(sd))

plt.plot(y_axis.index(myFriends),myFriends, marker='x', color='r')
plt.text(y_axis.index(myFriends),myFriends+3000,”hemanthmalla−193”)

plt.title('Twitter Followers Of Followers For Hemanth Malla')
# plt.xticks(x_axis)
plt.grid(True)
plt.show()
```

Listing 4: Python Script for generating plot Twitter Followers Of Followers Of Hemanth Malla

Mean	Standard Deviation	Median
5643.14	42238.58	155

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

In the plot shown in below Figure , you can see that Hemanth malla has many Followers with higher Follwers counts than him, with his count being 193 (not including himself). Therefore, the friendship paradox does hold for Hemanth malla's Twitter account.


Twitter Followers Of Followers For Hemanth Malla
Followers Count
0
100000
200000
300000
400000
Users
0
25
50
75
100
125
150
175
200
Median-155
hemanthmalla-193
Mean-5643.14
SD-42238.58

References

[1] https://developer.linkedin.com/docs/rest-api.

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