Srividya Majeti

# Assignment 4

CS 532: Introduction to Web Science
Dr. Michael Nelson
Spring 2016

February 26, 2016

# Contents

# 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):**

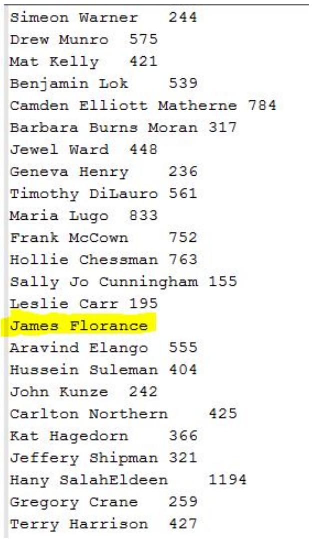
```
<node id="Johan_Bollen_1448621116">
        <data key="Label">Johan Bollen</data>
        <data key="uid"><![CDATA[1448621116]]></data>
        <data key="name"><![CDATA[Johan Bollen]]></data>
        <data key="mutual_friend_count"><![CDATA[37]]></data>
        <data key="friend_count"><![CDATA[420]]>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

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

Following are the steps that I have taken to solve the problem:

- I downloaded the XML file with the facebook friendship graph and using the 'xml.etree' library I parsed through the graph to retrieve the user name and friend count.
- I stored the retrieved data in a file 'friendNameAndCount'. This code is listed in Listing 1.1

- The sample output of the above mentioned file with user name and friend count is in Figure 1.1



```
Simeon Warner   244
Drew Munro  575
Mat Kelly   421
Benjamin Lok    539
Camden Elliott Matherne 784
Barbara Burns Moran 317
Jewel Ward  448
Geneva Henry    236
Timothy DiLauro 561
Maria Lugo  833
Frank McCown    752
Hollie Chessman 763
Sally Jo Cunningham 155
Leslie Carr 195
James Florance
Aravind Elango  555
Hussein Suleman 404
John Kunze  242
Carlton Northern    425
Kat Hagedorn    366
Jeffery Shipman 321
Hany SalahEldeen    1194
Gregory Crane   259
Terry Harrison  427
```

**Fig. 1.1.** Output file with user name and friend count

- As highlighted in Figure 1.1, I noticed that 11 friends did not have a friend count. Their names are as follows 'James Florance', 'Joy Gooden', 'Kim Beveridge', 'Alfredo Snchez', 'Sarah Shreeves', 'Sally Mauck', 'Dan Swaney', 'Robert Gordeaux', 'Joseph Kaplan', 'Michael Milner' and 'Catherine Kemble Cronin'.
- I stored only the friend count in a file 'friendCount'. By taking this file as input I calculated the mean, median and standard deviation of the number of friends that your friends have. This code is listed in Listing 1.2. The output of mean, median and standard deviation are given in Table 1.1.

**Table 1.1.** Mean, Median and Standard Deviation of number of friends of friends

| Key | Value |
|---|---|
| Mean | 358.987012987 |
| Median | 266.5 |
| Standard Deviation | 370.376887498 |

- Figure 1.2 illustrates your ranking(in red) in terms of number of friends(y-axis) in comparison to your friends.

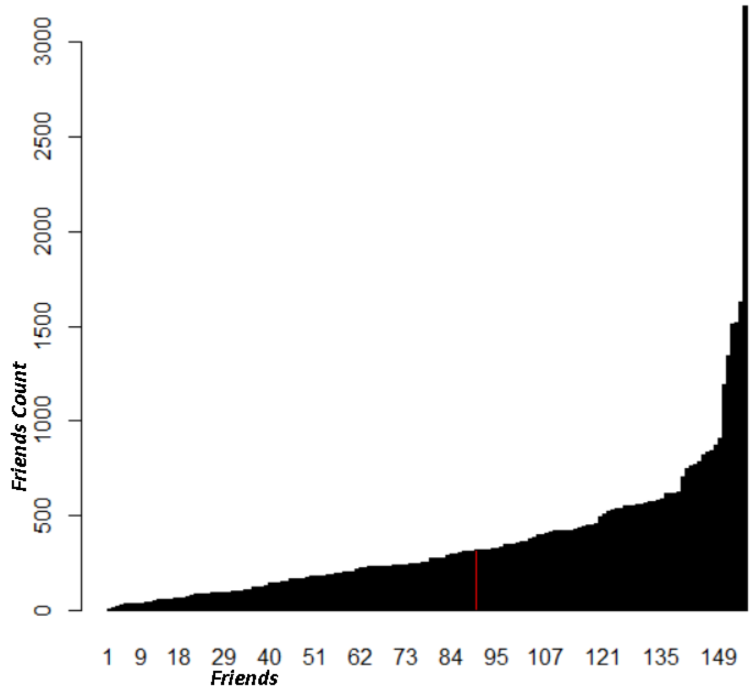


**Fig. 1.2.** Graph with number of friends on y-axis and the friend index on x-axis

- In conclusion, the above figure clearly indicates that you have less number of friends in comparison to your friends.

### Code Listing

```
from xml.etree import ElementTree

with open('mln.graphml', 'rt') as f:
  f1= open('friendCount','w')
  tree = ElementTree.parse(f)
  root = tree.getroot()
  print root
  list=[]
  for parent in root:
    for child in parent:
      if child.tag == ('{http://graphml.graphdrawing.org/xmlns}node'):
        for children in child:
          try:
            if children.attrib.get('key') == 'name':
              name=children.text
          except:
            name = name.encode('ascii',errors='ignore')
            print name
          if children.attrib.get('key') == 'friend_count':
            friendCount=children.text
            if friendCount >0:
              f1.write(str(friendCount)+”\n”)
            else:
              f1.write(” None \n”)
  f1.close()
```

**Listing 1.1.** Python code for extracting friends count from XML file

### Code Listing

```
import numpy as np

data = np.loadtxt('friendCount')

mean=np.average(data)
median=np.median(data)
standardDeviation=np.std(data)

f = open('MeanMedianStd','w')
f.write("mean:"+str(mean)+"\n")
f.write("median:"+str(median)+"\n")
f.write("standardDeviation:"+str(standardDeviation)+"\n")
```

**Listing 1.2.** Python code for calculating mean median and standard deviation

# 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".**

Following are the steps that I have taken to solve this problem:

- I did not have more than 50 followers, so I randomly picked a user with the screen name 'ohttic' from your followers list. He has 258 Followers and is following 1,923 people.
- Twitter provides an API to get the list of followers. This API returns an object of followers with the profile information. 'Tweepy' library is a wrapper around Twitter that makes it easier to retrieve Twitter data. I used this library to get the followers data.
- I iterated through the list of users and retrieved the 'screen_name', 'followers_count' and 'friends_count'. I stored this information in a JSON structure and saved it into a file 'userFollowerdata' . This code is listed in Listing 2.1.
- Furthermore, I extracted the followers count from the JSON and stored it in a file 'followersCount'. This code is listed in Listing 2.2
- I calculated the mean, median and standard deviation for the followers count. This code is listed in Listing 2.3. The mean, median and standard deviation are given in Table 2.1

**Table 2.1.** Mean, Median and Standard Deviation of number of followers of followers

| Key | Value |
|---|---|
| Mean | 54474.2 |
| Median | 5477.0 |
| Standard Deviation | 120893.613437 |

- Figure 2.1 illustrates the ranking of ‘ohttic’ in terms of followers count in comparison with his followers.

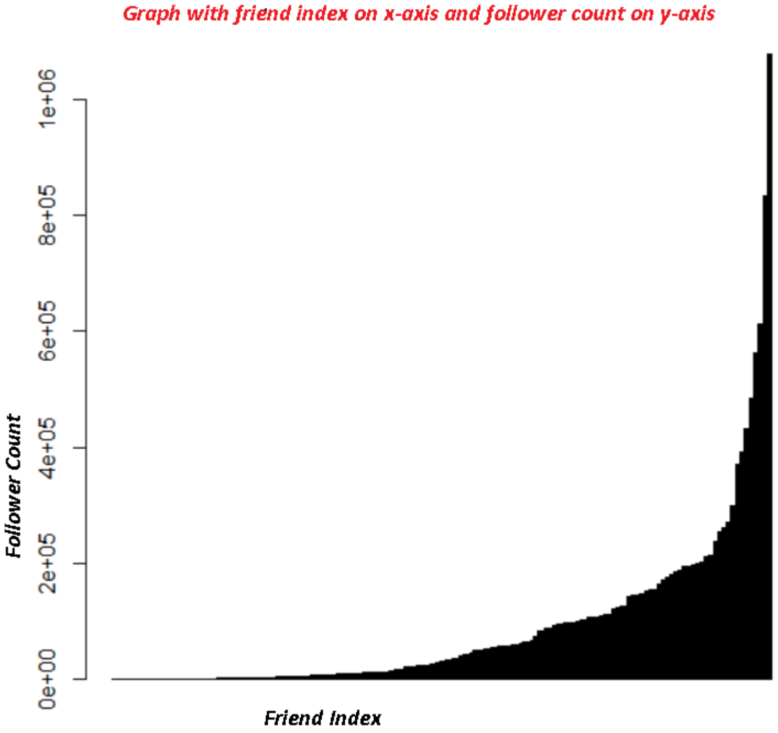


**Fig. 2.1.** Graph with number of followers on y-axis and followers on x-axis

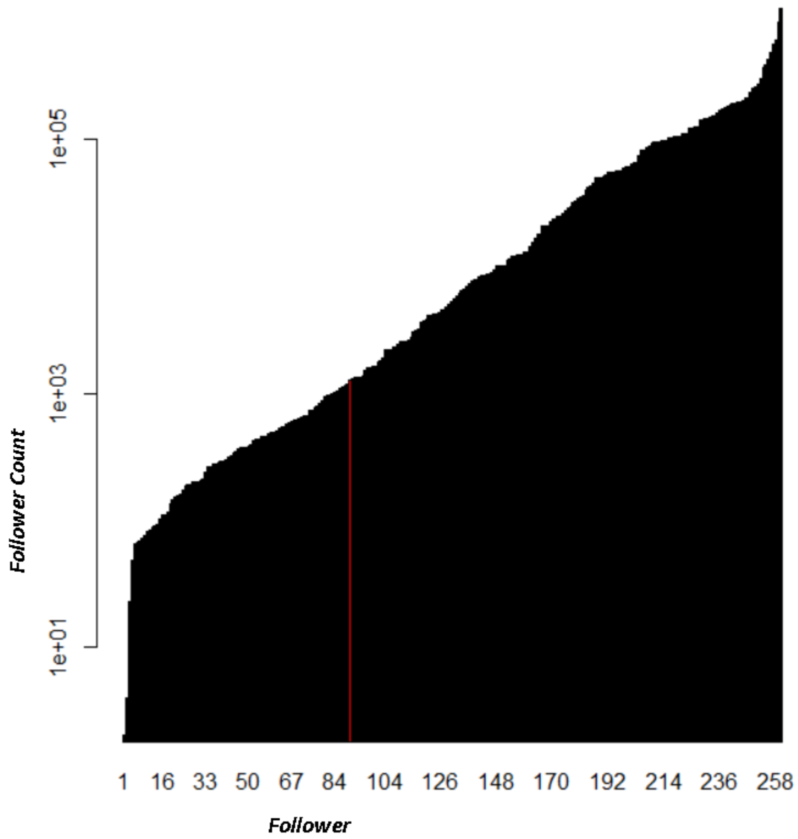


**Fig. 2.2.** Graph with number of followers on y-axis in log scale and followers on x-axis

- From the calculated median value ‘5477.0’ and number of followers ‘ohttic’ have ‘258’ we can say that he have less number of followers than his followers.

### Code Listing

```
import tweepy
import sys
import json
import time

CONSUMER_KEY = 'wTSsHE3PTA3ZZPiaKHEiQnLtf'
CONSUMER_SECRET = '
    UblYYCmNYIEffAY4T4QHGHXwAWMFqiueXdxf35xZFhoK3AECP1'
ACCESS_KEY = '157985123-
    WFvzlfDa8KStBZzevMfQBTM7fi8zKHYl2LQpTfGr'
ACCESS_SECRET = '
    lSax0XLwIimJ4VVbuU5OY9BpBic4vsSFi0riAq3DPvTxU'

auth = tweepy.auth.OAuthHandler(CONSUMER_KEY,
    CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
api = tweepy.API(auth)

f = open('userFollowerData','w')
def get_followers():
  users = []
  page_count = 0
  userData = []
  for user in tweepy.Cursor(api.followers, screen_name='
      ohttic').items():
    usr = {}
    usr['screen_name'] = user.screen_name
    usr['followers_count'] = user.followers_count
    usr['friends_count'] = user.friends_count
    page_count += 1
    userData.append(usr)
    print str(page_count)
  f.write(str(json.dumps(userData)))
  f.close()

get_followers()
```

**Listing 2.1.** Python code for retrieving friends data and storing screenName followersCount and friendsCount in a JSON structure

**Code Listing**

```
import json

read=open(’userFollowerData’,’r’)
f1= open(’followersCount’,’w’)
for line in read:
  data= json.loads(line)
  for user in data:
    f1.write(str(user[’followers_count’]) +”\n”)
```

**Listing 2.2.** Python code for extracting followers count from JSON structure

**Code Listing**

```
import numpy as np

data = np.loadtxt(’followersCount’)

mean=np.average(data)
median=np.median(data)
standardDeviation=np.std(data)

f = open(’MeanMedianStd’,’w’)
f.write(”mean:”+str(mean)+”\n”)
f.write(”median:”+str(median)+”\n”)
f.write(”standardDeviation:”+str(standardDeviation)+”\n”)
```

**Listing 2.3.** Python code for calculating mean median and standard deviation

# 3

# Extra-Credit Question 4

**Repeat question 2, but change “followers” to “following”? In other words, are the people I am following following more people?**

- Using the ‘userFollowerData’ file generated in question 2 as input, I extracted the following _count from the JSON structure and stored it in a file ‘followingCount’. This code is listed in Listing 3.1
- I calculated the mean, median and standard deviation for the following count. This code is listed in Listing 3.2. The output of mean, median and standard deviation are given in Table 3.1

**Table 3.1.** Mean, Median and Standard Deviation of number of people followed by the people followed by ‘ohttic’

| Key | Value |
|---|---|
| Mean | 45183.2 |
| Median | 4609.5 |
| Standard Deviation | 96280.5156542 |

- I created a graph with following count on y-axis and the friends themselves on x-axis including ‘ohttic’. The graph is summarized in Figure 3.1.

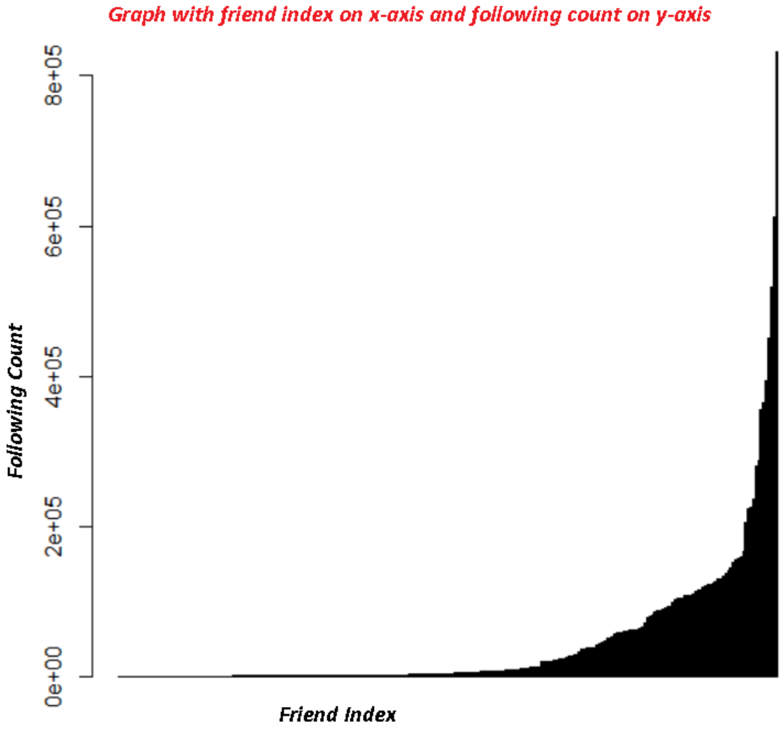


**Fig. 3.1.** Graph with following count on y-axis and the friend index on x-axis

- From the calculated median value ‘4609.5’ and following count of ‘ohttic’ ‘1,923’ we can say that ‘ohttic’ have less following count than his friends. 3.1.

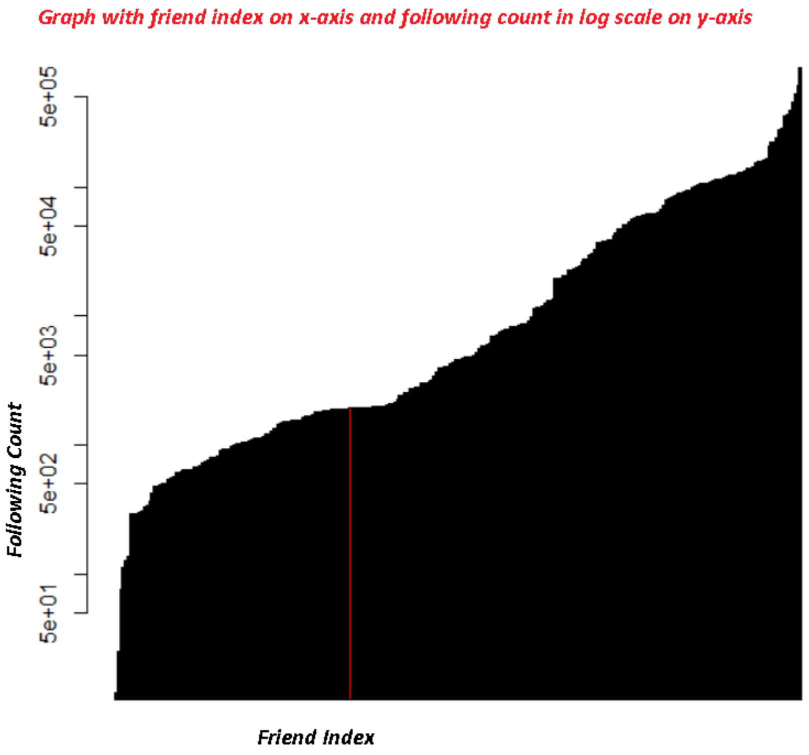


**Fig. 3.2.** Graph with following count in log scale on y-axis and the friend index on x-axis

### Code Listing

```
import json

read=open('userFollowerData','r')
f2= open('followingCount','w')
for line in read:
  data= json.loads(line)
  for user in data:
    f2.write(str(user['friends_count'])+ "\n")
```

**Listing 3.1.** Python code for extracting friends count from JSON structure

### Code Listing

```
import numpy as np

data = np.loadtxt('followingCount')

mean=np.average(data)
median=np.median(data)
standardDeviation=np.std(data)

f = open('MeanMedianStd','w')
f.write("mean:"+str(mean)+"\n")
f.write("median:"+str(median)+"\n")
f.write("standardDeviation:"+str(standardDeviation)+"\n")
```

**Listing 3.2.** Python code for calculating mean median and standard deviation

# References

1. How to parse xml: http://docs.python-guide.org/en/latest/scenarios/xml/, A Kenneth Reitz, 2016
2. How to parse xml tree: https://pymotw.com/2/xml/etree/ElementTree/parse.html, Doug Hellmann
3. How to get followers list: https://dev.twitter.com/rest/reference/get/followers/list, Twitter, Inc 2015
4. How to get list of followers: http://stackoverflow.com/questions/17455107/the-best-way-to-get-a-list-of-followers-in-python-with-tweepy , Joel Spolsky, 2008
5. How to get followers and friends of a twitter user: http://codereview.stackexchange.com/questions/101905/get-all-followers-and-friends-of-a-twitter-user, Joel Spolsky, 2008
6. How to calculate standard deviation: http://stackoverflow.com/questions/15389768/standard-deviation-of-a-list, Joel Spolsky, 2008
7. How to calculate standard median: http://stackoverflow.com/questions/24101524/finding-median-of-list-in-python, Joel Spolsky, 2008
8. How to calculate mean: http://stackoverflow.com/questions/19870293/how-to-find-the-average-of-values-in-a-txt-file, Joel Spolsky, 2008