Web Science: Assignment #4

Alexander Nwala

Apurva Modi

Saturday, March 16, 2019

Contents

Problem 1

Determine if the friendship paradox holds for my Facebook account.* Compute the mean, median and standard deviation 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).

SOLUTION :

1. To Compute Mean

   ```
   Mean = Sum of all the Friend's friend Count / Total Friend Count
   ```

2. To compute Median

   ```
   Median =  Middle Item of the Sorted Friends count
   ```

3. To compute Standard Deviation

   ```
   Standard Deviation = SquareRoot {Sum of all([Square(friend count - Mean)])
   /Total Friend Count}
   ```

Listing 1: Assignmnent4_1.py

```
 #This is a Python2 Program
import csv
import math
from astropy.table import Table
import numpy as np

friend_Dict= {}
total_Count = 0
friend_List = []
friend_Count = 0
friend_Mean = 0
friend_Median = 0
friend_SD = 0
count=1
friend_count = 0
def mean():
     global friend_Mean
     friend_Mean = round((total_Count / friend_Count),2)
     return friend_Mean

def median():
     global friend_Median

     mid = 0
     friend_List.sort(key=int)

     mid = len(friend_List) / 2
     if(mid == 0):
          friend_Median = friend_List[mid]
```

Problem 1 continued on next page...

```
     else:
          mid = mid +1
          friend_Median = friend_List[mid]
     return friend_Median

def SD():
     global friend_SD
     global friend_Mean
     SD_Sum = 0
     for num in friend_List:
          SD_Sum = SD_Sum + ((int(num) - int(friend_Mean))
                             *  (int(num) - int(friend_Mean)))

     #print('SD_Sum',SD_Sum)

     friend_SD = round(math.sqrt(SD_Sum / friend_Count),2)
     return friend_SD

with open('acnwala-friendscount.csv') as csvfile:
     total_Count = 0
     readCSV = csv.reader(csvfile, delimiter=',')
     for row in readCSV:
          if  "FRIENDCOUNT" in row[1]:
               continue
          else:
               friend_List.append(row[1])
               total_Count = total_Count + int(row[1])
               friend_Count = friend_Count + 1
     print (total_Count)

friend_List.sort(key=int)
f = open('fb-Friends.txt','w')
for num in friend_List:
     print (num)
     f.write(str(count)+","+str(num))
     f.write('\n')
     count = count + 1
     friend_count = friend_count + int(num)
f.close()

mean = [str(mean())]
median = [str(median())]
SD = [str(SD())]
tableList = Table([mean, median, SD], names=('MEAN', 'MEDIAN', 'STANDARD DEVIATION'))
print (tableList)
```

Listing 2: graph4_1.py

```
#This is a Python2 Program

import matplotlib.pyplot as plt
import csv

x = []
y = []

with open('fb-Friends.txt','r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(int(row[0]))
        y.append(int(row[1]))

plt.plot(x,y)

plt.plot(11,98,marker=".", color='red',markersize=12)
plt.annotate('Dr. Nwala\'s Friends : 98' , xy=(17, 98))

plt.plot(52,431,marker=".", color='red',markersize=12)
plt.annotate('Median : 431' , xy=(40, 500))

plt.plot(63,536.67,marker=".", color='red',markersize=12)
plt.annotate('Standard Deviation : 536.67' , xy=(63, 350),)

plt.plot(65,542,marker=".", color='red',markersize=12)
plt.annotate('Mean : 542' , xy=(65, 700))

plt.xlim(0, 105)
plt.xlabel('Facebook Friends')

plt.ylabel('No. of Friends for Each Friend')
plt.title('Facebook Friends Vs Each Friend count')
plt.legend()
plt.show()
```

The below plot will show the friendship paradox for **Dr.Alexander Nwala's** facebook account

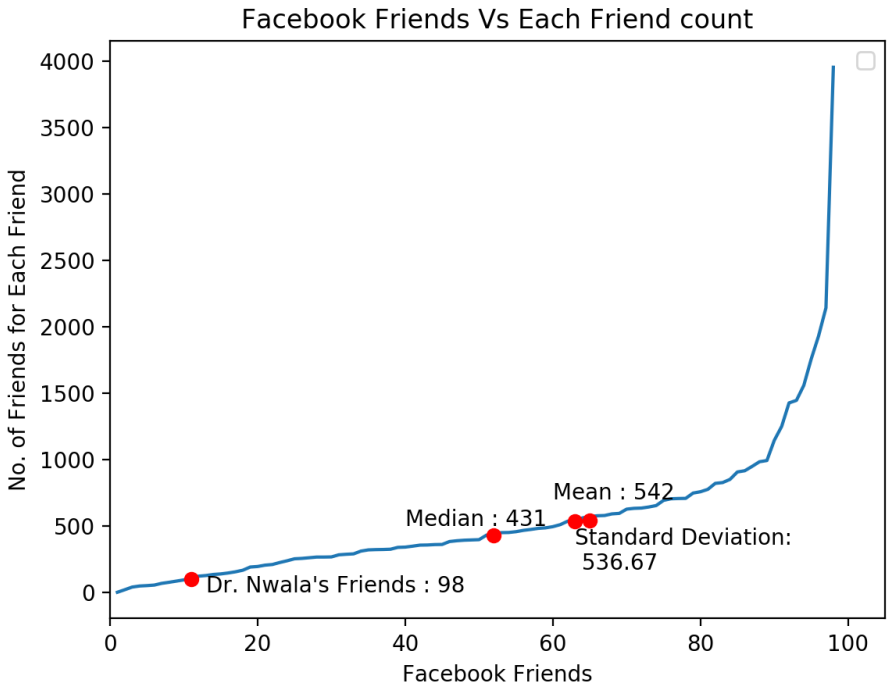

Figure 1: Friends vs Friends Count (Facebook)

Problem 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 number 1, and calculate the same mean, standard deviation, and median values..

SOLUTION

The program requires to use the access tokens generated while creating the twitter developer account .

1. Download and install the twitter API i.e. **tweepy** :

   ```
   pip install tweepy
   ```

2. Currently running the tweepy on the **acnwala** twitter handle:

   ```
   user = tweepy.Cursor(api.followers, screen_name=twitter_handle,
   count=200).items()
   ```

   The above statement will fetch all the followers details in JSON for the twitter handle.

Listing 3: Assignment4_2.py

```
 #This is a Python2 Program

import tweepy
import time
import csv
import math
from astropy.table import Table

#My twitter Account keys
ckey = 'mtDSeNYtJUZkKfspxTFmk7Nn8'
csecret = 'iYg5kksoIQKGsXVwGZ7bYpH0cFlxonNPg9hyhDKdGoP89bic6G'
atoken = '1094978973245812738-msYR1atvDnyfTTO46shWdnp5SIJcAA'
asecret = 'EMdqw7fA4IfkDYzKBqNQfoe5sAwz7dgCcRTWkpZOteUKd'
#Login Verifiavtion
auth = tweepy.auth.OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
api = tweepy.API(auth,wait_on_rate_limit=True)
if(api.verify_credentials):
    print ('Logged in successfully')

total_count = 0
listDict = {}
twitter_handle='acnwala'
friend_Mean = 0
friend_Median = 0
friend_SD = 0
friend_List = []
```

Problem 2 continued on next page. . .

```
count = 1
friend_count = 0


def mean():
    global friend_Mean
    friend_Mean = round((friend_count / total_count),2)
    return friend_Mean


def median():
    global friend_Median
    mid = 0
    friend_List.sort(key=int)
    avg = len(friend_List) % 2
    if(avg == 0):
        mid = len(friend_List) / 2
        friend_Median = friend_List[mid]
    else:
        mid = len(friend_List) / 2
        mid = mid +1
        friend_Median = friend_List[mid]
    return friend_Median


def SD():
    global friend_SD
    global friend_Mean
    stdDeviationSum = 0
    for num in friend_List:
        stdDeviationSum = stdDeviationSum + ((int(num) - int(friend_Mean)) *  (int(num) - int(friend_
        # print('stdDeviationSum',stdDeviationSum)
        friend_SD = round(math.sqrt(stdDeviationSum / total_count),2)
    return friend_SD

for follower in tweepy.Cursor(api.followers, screen_name=twitter_handle).items():
    total_count = total_count + 1
    listDict[follower.screen_name] = follower.friends_count
    friend_List.append(follower.friends_count)

friend_List.sort(key=int)
f = open('twitterFollowers-Friends.txt','w')
for friendCount in friend_List:
    f.write(str(count)+","+str(friendCount))
    f.write('\n')
    count = count + 1
    friend_count = friend_count + friendCount
f.close()

mean = [str(mean())]
median = [str(median())]
SD = [str(SD())]
tableList = Table([mean, median, SD], names=('MEAN', 'MEDIAN', 'STANDARD DEVIATION'))
```

```
print (tableList)
```

Listing 4: graph4_2.py

```
#This is a Python2 Program

import matplotlib.pyplot as plt
import csv

x = []
y = []

with open('twitterFollowers-Friends.txt','r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(int(row[0]))
        y.append(int(row[1]))

plt.plot(x,y)
plt.plot(65,250,marker='.', color='red',markersize=12)
plt.annotate('Dr.Nwala\'s Followers:250' , xy=(2, 620))

plt.plot(117,560,marker='.', color='red',markersize=12)
plt.annotate('Median:560' , xy=(120, 150))

plt.plot(185,1461.0,marker='.', color='red',markersize=12)
plt.annotate('Mean:1461.0' , xy=(190, 1260))


plt.plot(225, 3634.72,marker='.', color='red',markersize=12)
plt.annotate('Standard\nDeviation:3634.72' , xy=(140, 4000))

plt.xlim(0, 300)
plt.ylim(0, 10000)

plt.xlabel('All Followers')
plt.ylabel('No. of Friends for Each Follower')
plt.title('Twitter Followers Vs Each Follower\'s Friend count')
plt.legend()
plt.show()
```

The below plot will show the friendship paradox for **Dr.Alexander Nwala's** twitter handle.

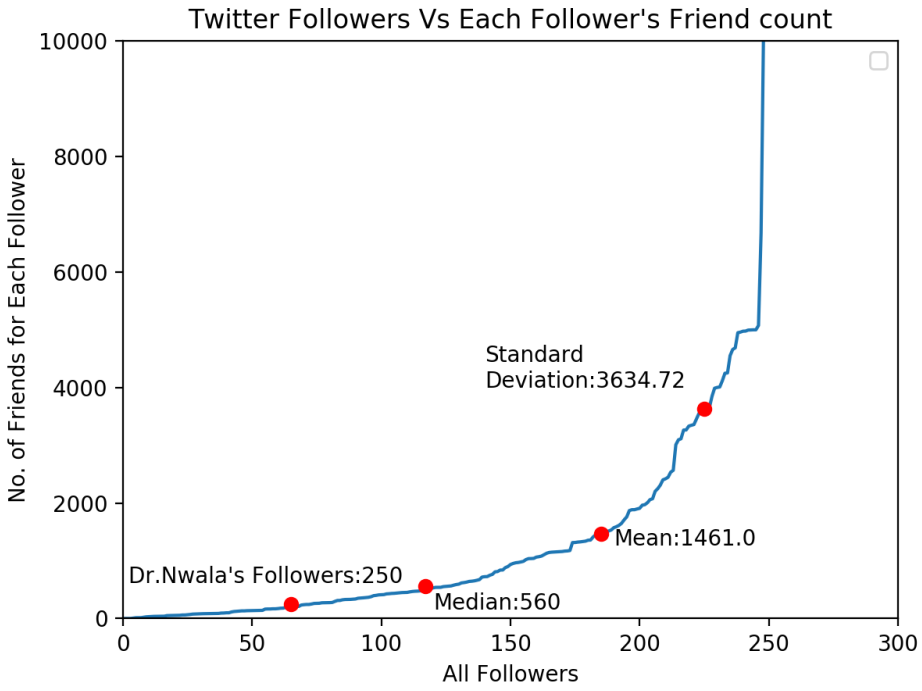

Figure 2: Followers vs Followers Count (Twitter)

The Sorted friend's list can be found in the **twitterFollowers-Friends.txt** text file.

Problem 3

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

SOLUTION

The program requires to use the access tokens generated while creating the twitter developer account .

1. Running the tweepy on the **acnwala** twitter handle:

   ```
   user = tweepy.Cursor(api.friends, screen_name=twitter_handle,
   count=200).items()
   ```

   The above statement will fetch all the friends list in JSON for the twitter handle.

Listing 5: AssignmentExC4_4.py

```
#This is a Python2 Program
import tweepy
import time
import csv
import math
from astropy.table import Table

#My twitter Account keys
ckey = 'mtDSeNYtJUZkKfspxTFmk7Nn8'
csecret = 'iYg5kksoIQKGsXVwGZ7bYpH0cF1xonNPg9hyhDKdGoP89bic6G'
atoken = '1094978973245812738-msYR1atvDnyfTTO46shWdnp5SIJcAA'
asecret = 'EMdqw7fA4IfkDYzKBqNQfoe5sAwz7dgCcRTWkpZOteUKd'
#Login Verifiavtion
auth = tweepy.auth.OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)
api = tweepy.API(auth,wait_on_rate_limit=True)
if(api.verify_credentials):
    print ('Logged in successfully')

total_count = 0
listDict = {}
twitter_handle='acnwala'
friend_Mean = 0
friend_Median = 0
friend_SD = 0
friend_List = []
count = 1
friend_count = 0


def mean():
    global friend_Mean
    friend_Mean = round((friend_count / total_count),2)
    return friend_Mean

```

```
def median():
    global friend_Median
    mid = 0
    friend_List.sort(key=int)
    avg = len(friend_List) % 2
    if(avg == 0):
        mid = len(friend_List) / 2
        friend_Median = friend_List[mid]
    else:
        mid = len(friend_List) / 2
        mid = mid +1
        friend_Median = friend_List[mid]
    return friend_Median


def SD():
    global friend_SD
    global friend_Mean
    stdDeviationSum = 0
    for num in friend_List:
        stdDeviationSum = stdDeviationSum + ((int(num) - int(friend_Mean)) *  (int(num) - int(friend_
        # print('stdDeviationSum',stdDeviationSum)
        friend_SD = round(math.sqrt(stdDeviationSum / total_count),2)
    return friend_SD

for follower in tweepy.Cursor(api.friends, screen_name=twitter_handle).items():
    total_count = total_count + 1
    listDict[follower.screen_name] = follower.friends_count
    friend_List.append(follower.friends_count)

friend_List.sort(key=int)
f = open('twitterFollowing.txt','w')
for friendCount in friend_List:
    f.write(str(count)+","+str(friendCount))
    f.write('\n')
    count = count + 1
    friend_count = friend_count + friendCount
f.close()

mean = [str(mean())]
median = [str(median())]
SD = [str(SD())]
tableList = Table([mean, median, SD], names=('MEAN', 'MEDIAN', 'STANDARD DEVIATION'))
print (tableList)
```

Listing 6: graphExC4_4.py

```
#This is a Python2 Program
import matplotlib.pyplot as plt
import csv

x = []
y = []

with open('twitterFollowing.txt','r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(int(row[0]))
        y.append(int(row[1]))

plt.plot(x,y)
plt.plot(10,92,marker='.', color='red',markersize=12)
plt.annotate('Dr.Nwala\'s Friends: 92', xy=(1, 392))

plt.plot(43,461,marker='.', color='red',markersize=12)
plt.annotate('Median: 461' , xy=(42, 121))

plt.plot(73.5,1255,marker='.', color='red',markersize=12)

plt.annotate('Mean: 1255.0' , xy=(76, 1250.0))

plt.plot(81.5,2625,marker='.', color='red',markersize=12)
plt.annotate('Standard\nDeviation: 2625.42' , xy=(51, 2500.42))

plt.ylim(0, 7000)
plt.xlim(0, 100)

plt.xlabel('All Friends')
plt.ylabel('No. of Friends for Each Friend')
plt.title('Twitter Friends Vs Each Friend count')
plt.legend()
plt.show()
```

The below plot will show the friendship paradox for **Dr.Alexander Nwala's** twitter handle.

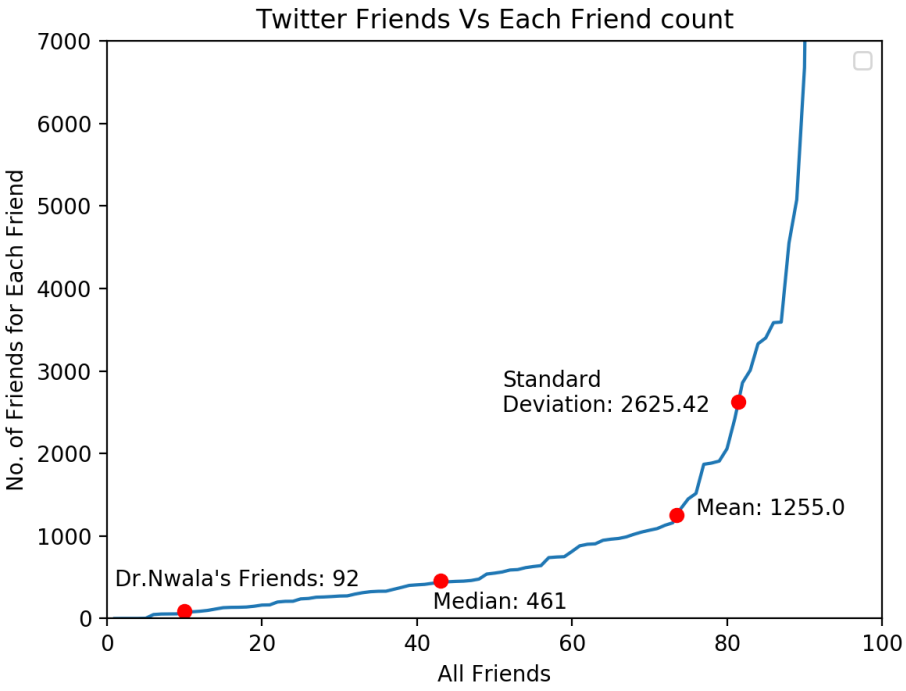

Figure 3: Following vs Following Count (Twitter)

The Sorted friend's list can be found in the **twitterFollowing.txt** text file.