Assignment #4

Wednesday, February 25TH, 2018

RANDY DO

Contents

Problem 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 extracted the csv to a text. (I just copy and pasted the cells on excel) The file is called friendcount.txt
I used this code to calculate the mean and median. *I was having issues calculating the standard deviation. So instead, I used a website called http://www.calculator.net/standard-deviation-calculator.html

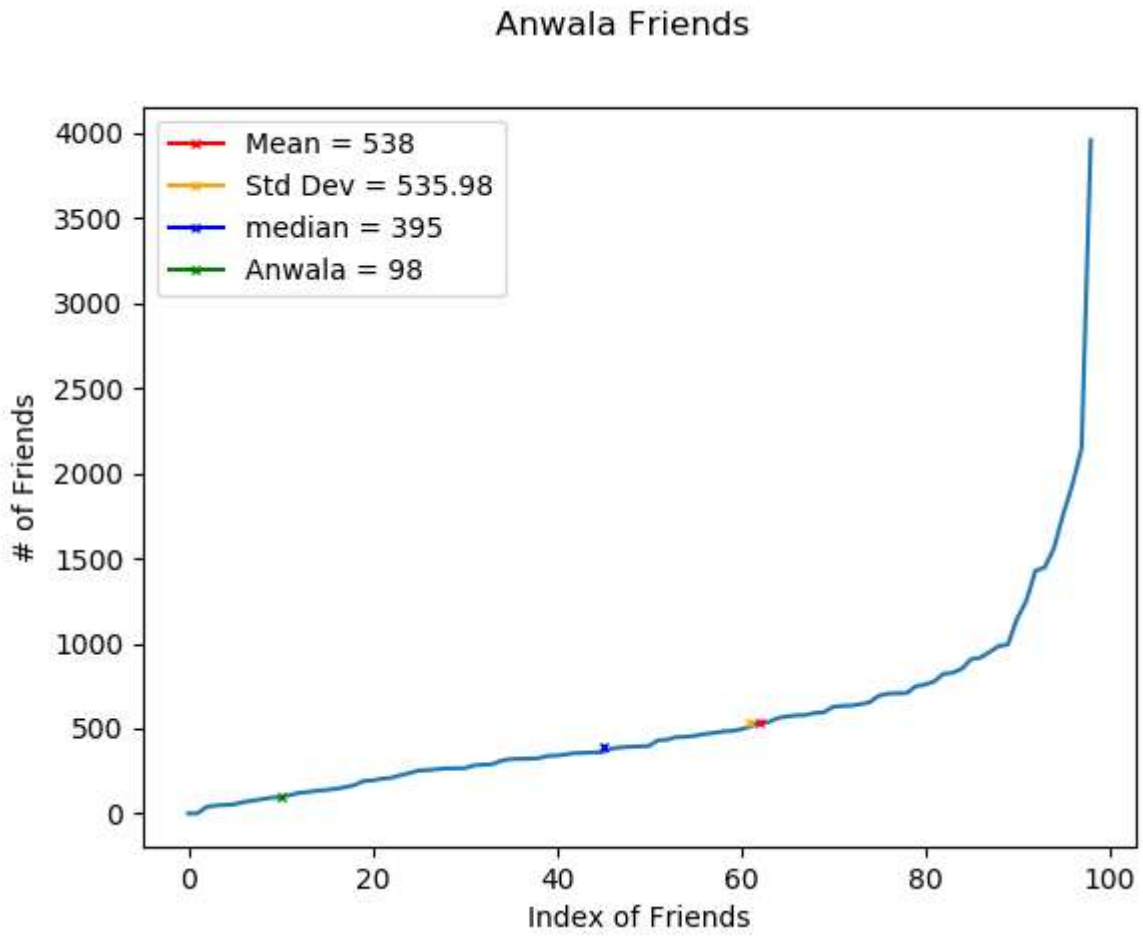
```
import re
import math
import statistics
import numpy



def calculateMean():
    lis=[]
    with open ('twitter_counts.txt ','r') as nc:
        total = sum(int(x)
        for line in nc
        for x in line.split())
    mean = total /99
    print ("Mean = ",mean)
    return mean



def calculateMedian():
    ls=[]
    with open ('twitter_counts.txt ','r') as n:
        for line in n:
            ls.append(line.strip('\n'+''))
        ls =list(map(int, ls))
        median = statistics.median(ls)
        print ("Median = ",median)

calculateMean()
calculateMedian()
```

```
C:\Python27>python values.py
('Mean = ', 538)
('Median = ', 395)
```

Sample Standard Deviation, s	538.53208199653
Variance (Sample Standard), s^2	290016.80333952
Population Standard Deviation, σ	535.80531975665
Variance (Population Standard), σ^2	287087.34067952
Total Numbers, N	99
Sum:	53280
Mean (Average):	538.18181818182
Standard Error of the Mean ($SE_{\bar{x}}$):	54.124510713679

This is the code to create the graph. I got the x variables from slack

```
import operator
from collections import Counter
import numpy as np
import matplotlib.pyplot as plt
from statistics import median
import math


x = [1,2,40,48,51,55,68,76,85,94,98,108,122,127,135,139,146,155,167,191,195,205,210,225,238,253,256,261,266,
mean = 538
medium = 395
std = 535.98
anwala = 98

plt.plot(x)
plt.suptitle('Anwala Friends')
plt.plot([62],mean,marker='x',markersize=4, color='red', label = 'Mean = 538')
plt.plot([61],std,marker='x',markersize=4, color='orange', label = 'Std Dev = 535.98')
plt.plot([45],medium,marker='x',markersize=4, color='blue', label = 'median = 395')
plt.plot([10],anwala,marker='x',markersize=4, color='green', label = 'Anwala = 98')
plt.legend()
plt.ylabel('# of Friends')
plt.xlabel('Index of Friends')
plt.show()
```



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

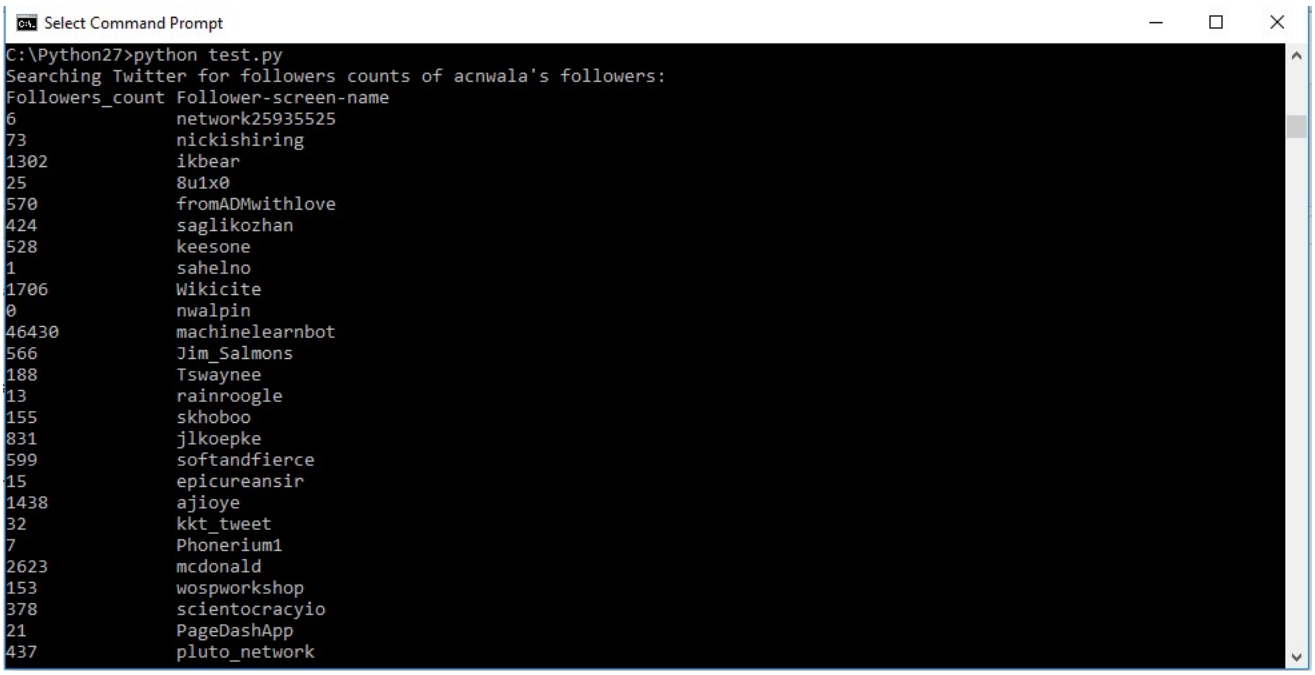
For the Twitter 1.1 API to help gather this data, see:

https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-list

Approach: Use twittergrab.py

```
if __name__ == "__main__":
    if not OAUTH_TOKEN:
        token, secret = setup_oauth()
        print "OAUTH_TOKEN: " + token
        print "OAUTH_TOKEN_SECRET: " + secret
        print
    else:
        twitterUser = "acnwala"

        print 'Searching Twitter for followers counts of '+twitterUser+"'s followers: "
        oauth = get_oauth()

        print '%-15s %-20s' %('Followers_count','Follower-screen-name')

        # initial reading from the twitter account where cursor = -1 (e.g. first page)
        r = requests.get(url="https://api.twitter.com/1.1/followers/list.json?cursor=-1&count=2000&screen_name="+twitt
        counter = 0
        res = r.json()
        while True:
            raw_res = res['users']
            for init_url in raw_res:
                counter = counter + 1
                print '%-15d %-20s' %(init_url['followers_count'],init_url['screen_name'].encode('ascii', 'replace'))
                with open('twitter_counts.txt', 'a') as outfile:
                    outfile.write('%-15d\n' %(init_url['followers_count']))
            if str(res['next_cursor']) == '0':
                break
            else:
                r = requests.get(url="https://api.twitter.com/1.1/followers/list.json?cursor="+str(res['next_cursor'])
                res = r.json()

print '\nNumber of '+twitterUser+"'s followers is: "+str(counter)
```

Example of the output in the cmd.
When I am writing this to a file name twitter_count, I am just getting the file count.



After getting the files, I used valuestwitter.py to get the median and mean `I used this code to calculate the mean and median. *I was having issues calculating the standard deviation. So instead, I used a website called http://www.calculator.net/standard-deviation-calculator.html`

```
import re
import math
import statistics
import numpy



def calculateMean():
    lis=[]
    with open ('twitter_counts.txt ','r') as nc:
        total = sum(int(x)
        for line in nc
        for x in line.split())
        #print ("Total = ",total)
    mean = total /191
    print ("Mean = ",mean)
    return mean



def calculateMedian():
    ls=[]
    with open ('twitter_counts.txt ','r') as n:
        for line in n:
            ls.append(line.strip('\n'+''))
        ls =list(map(int, ls))
        median = statistics.median(ls)
        print ("Median = ",median)

calculateMean()
calculateMedian()
```

```
C:\Python27>python valuestwitter.py
('Mean = ', 3011)
('Median = ', 274.0)
```

Sample Standard Deviation, s	14460.160369368
Variance (Sample Standard), s^2	209096237.90785
Population Standard Deviation, σ	14422.256869393
Variance (Population Standard), σ^2	208001493.20677
Total Numbers, N	191
Sum:	575181
Mean (Average):	3011.4188481675
Standard Error of the Mean ($SE_{\bar{x}}$):	1046.3004831731

Same as problem one, I created a graph via python

```
4308,
4385,
4405,
5165,
5241,
7014,
8938,
23890,
42627,
45255,
46430,
51007,
53954,
170147
]
mean = 3011
medium = 274
std = 14422

plt.plot(x)
plt.suptitle('Twitter Friends')
plt.plot([172],mean,marker='x',markersize=4, color='red', label = 'Mean = 538')
plt.plot([181],std,marker='x',markersize=4, color='orange', label = 'Std Dev = 535.98')
plt.plot([145],medium,marker='x',markersize=4, color='blue', label = 'median = 395')
plt.legend()
plt.ylabel('# of Followers')
plt.xlabel('Index of followers')
plt.show()
```

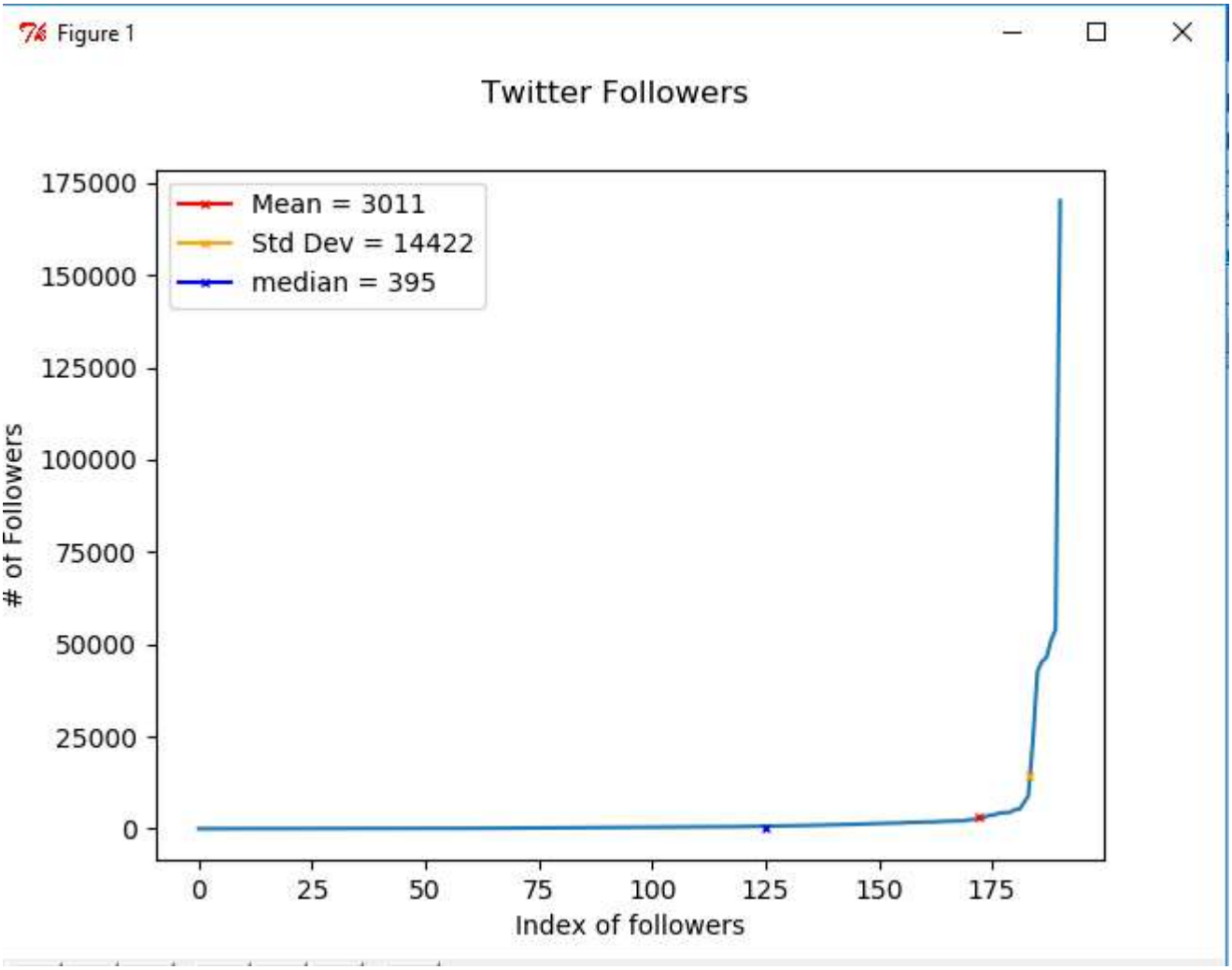


Figure 1
Twitter Followers
Mean = 3011
Std Dev = 14422
median = 395
# of Followers
Index of followers
175000
150000
125000
100000
75000
50000
25000
0
0
25
50
75
100
125
150
175

Date