

INTRO. TO WEB SCIENCE: CS 532: A4

Due on Thursday, March 02, 2017

Dr. Nelson

Udochukwu Nweke

Contents

Problem 1	3
Problem 2	4

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 (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.

Listing 1: Extract Facebook Friend Count Code

```

from util import *

from os.path import dirname, abspath
from bs4 import BeautifulSoup

5 def workingFolder():
    return dirname(abspath(__file__)) + '/'

def getFriendCountDict(friendsTxt):
10
    friendsDict = {}
    problematicFriendsCount = 0
    try:
        soup = BeautifulSoup(friendsTxt, 'xml')
15 graph = soup.find('graph')

        if( graph is not None ):
            nodes = graph.findAll('node')

20
            for node in nodes:
                try:
                    friendName = node.find('data', {'key': 'Label'}).text
                    friendCount = node.find('data', {'key': 'friend_count'}).text
                    friendsDict[ node['id'] ] = {'friend_name': friendName,
25 'friend_count'
                        : int(friendCount)}
                except:
                    errorMsg()
                    problematicFriendsCount += 1

30 except:
    errorMsg()

    print('\nfriendsDict.len:', len(friendsDict))
    print('problematicFriendsCount:', problematicFriendsCount, '\n')
35 sortedIDs = sorted(friendsDict, key=lambda x: friendsDict[x]['friend_count'])

    try:
        outfile = open('./mlnFriends.csv', 'w')

40
        outfile.write('Friend, FriendsCount\n')
        for friendID in sortedIDs:
            print(friendsDict[friendID]['friend_name'], friendsDict[friendID]
                ['friend_count'])
            outfile.write( friendsDict[friendID]['friend_name'] + ', ' +
45 str(friendsDict[friendID]['friend_count']) + '\n' )

```

```

        outfile.close()
    except:
        errorMsg()
50 friendsTxt = readTextFromFile(workingFolder() + 'mlnFriends.xml')
    getFriendCountDict(friendsTxt)

```

Listing 2: R Script to plot Friend Count. Calculate SD and Mean and Median of Friend Count

```

friendsCount <- read.csv('./fb.mlnFriends.csv', head=TRUE, sep=',')
plot(friendsCount$FriendsCount, xlab='Friends', ylab='Number of friends',
main='')
text(100, 120, 'Dr. Nelson has 165 friends')
5 abline(h=165, v=0)

summary(friendsCount$FriendsCount)
sd(friendsCount$FriendsCount)

10 friendsCount <- read.csv('./twitter.mlnFriends.csv', head=TRUE, sep=',')
plot(friendsCount$FriendsCount, log='y', xlab='Followers', ylab='Number of Followers',
main='Dr. Nelson\'s Twitter Follower vs Number of Followers')
text(100, 10, 'Dr. Nelson has 304 Followers')
abline(h=30, v=0)
15 summary(friendsCount$FriendsCount)
sd(friendsCount$FriendsCount)

```

Solution 1:

1. In order to get the names of Dr. Nelson's friend and their friends count, I used Beautiful soup to parse the given xml (mlnFriend.xml) into a structure. This is shown in listing 1, line 14- 15.

I wrote a dictionary *getFriendCountDict* that generates the Friend and Friends counts and wrote the result into *fb.mlnFriends.csv* file. This is seen in listing 1, and Table 2

Dr. Nelson has a total number of 165 friends and 11 of his friends didn't have friend count. I called the 11 friends the problematic friends and skipped them in my *fb.mlnFriends.csv* file record. *nelsonFriendsStat.txt* contains the total number of freinds and number of problematic friends (Friends without friend count).

2. I wrote an R script to plot the friendship graph (Fig. 1) and compute the mean, median, and standard deviation. This is demonstrated in listing 2 line 1-5, and Table 1.

Table 1: Mean, median and SD numbers of Dr. Nelson's Facebook Friends

Mean	Median	Standard Deviation
359.0	266.5	371.5853

Based on the graph and the statistics, Dr. Nelson has a total number of 154 friends. The mean and meadian number of his friends are 359.0 and 266.5 respectively. Therefore, the frienship paradox holds.

Problem 2

Figure 1: Dr. Nelson's Friends vs Number of Friends

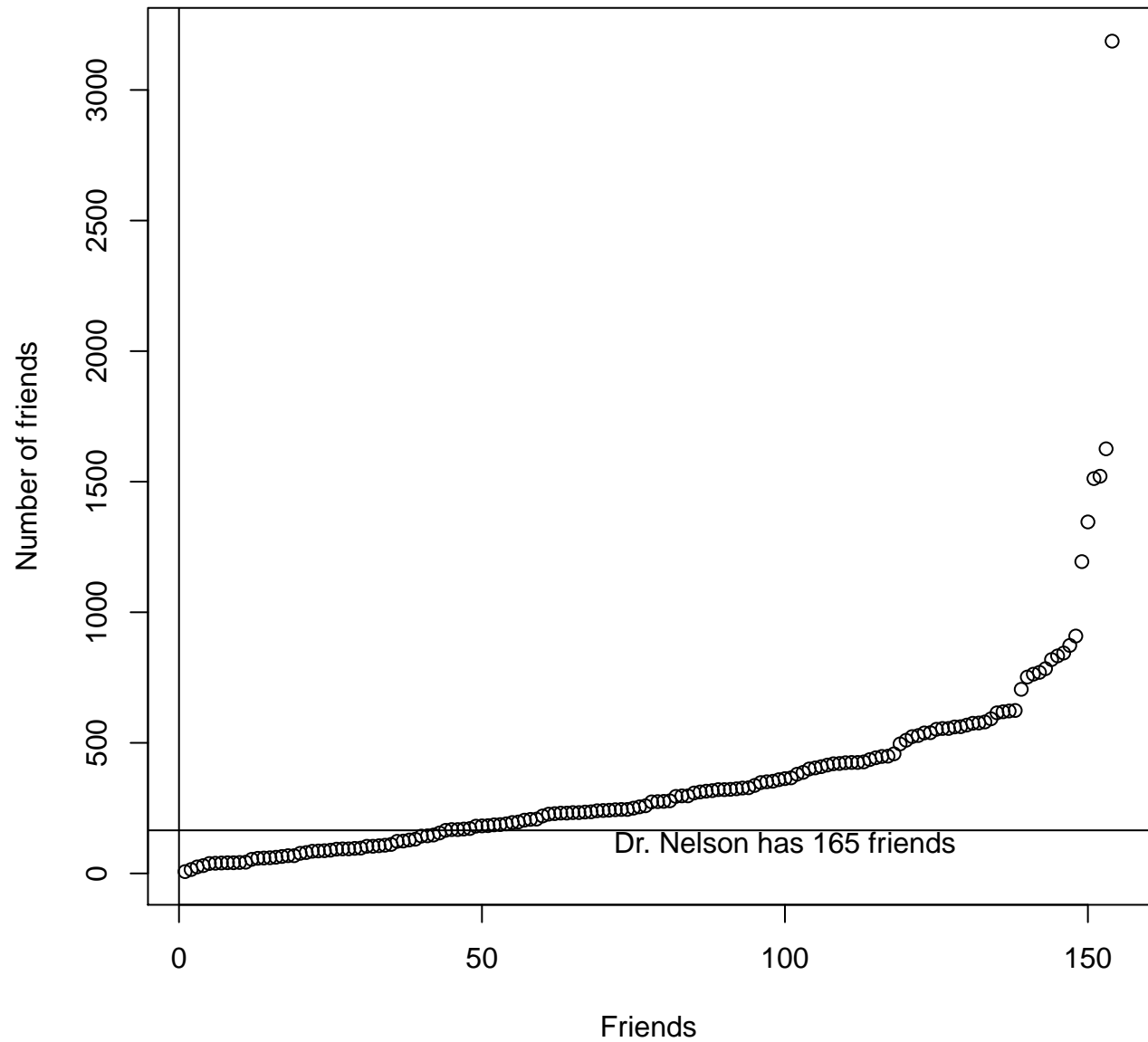


Table 2: Subset of Dr. Nelson's Facebook Friends and their respective Friends Count

Friend	FriendsCount
Doug Nelson	7
Brian K. Saunders	15
Winnie Elliott	25
Joan A. Smith	30
Bob Mathers	38
Thomas Allen	39
Lloyd Nelson	40
Scott W Laney	41
Mary McManus,	41
Greg Szalkowski	42
Calvin Edward Mackey	1626
Ricardo Baeza-Yates	3187

Listing 3: Retrieve Dr. Nelson's Twitter Followers Code

```

from util import *
import tweepy
import json

5 consumer_key = '01oEm0j9Y52TftjA5wgRHt9z6'
consumer_secret = 'MWPYnHvrezvhuuFQkOQgnBMYwXZD6SJpnoI8Q1E7ZajgQMUKKs'
access_token = '154076252-uK6XnhweIkuc0qIvsNmGiiRebLqvYHbtWDgA5PBi'
access_token_secret = 'LDA5Qel3UQtIwUhvAZLffCGZ9pmmc7wkFOL5k0xx5Yt90'

10 auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

15 def writeFollowers(response, friendsDict):

    for follower in response:
        friendsDict[follower.screen_name] = follower.followers_count

20 def getFriendsDict (twitterName):

    try:
        rateLimitQuota = api.rate_limit_status()['resources']['friends']
25         ['/friends/list']['remaining']
        print('\trateLimitQuota:', rateLimitQuota)

        twitterAccount = api.get_user(twitterName)
30         #followersCount = int(twitterAccount._json['friends_count'])

        friendsDict = {}
        response = twitterAccount.friends(count=200, cursor=-1)
        nextCursor = response[1][1]

```

```

35     writeFollowers(response[0], friendsDict)

    response = twitterAccount.friends(count=200, cursor=nextCursor)
    writeFollowers(response[0], friendsDict)

40     sortedIDs = sorted(friendsDict, key=lambda x: friendsDict[x])
    outfile = open('./twitter.mlnFriends.csv', 'w')

    outfile.write('Friend,FriendsCount\n')
    for screen_name in sortedIDs:
45         print(screen_name, friendsDict[screen_name])
        outfile.write( screen_name + ', ' + str(friendsDict[screen_name]) + '\n' )

    outfile.close()

50 except:
    errorMsg()

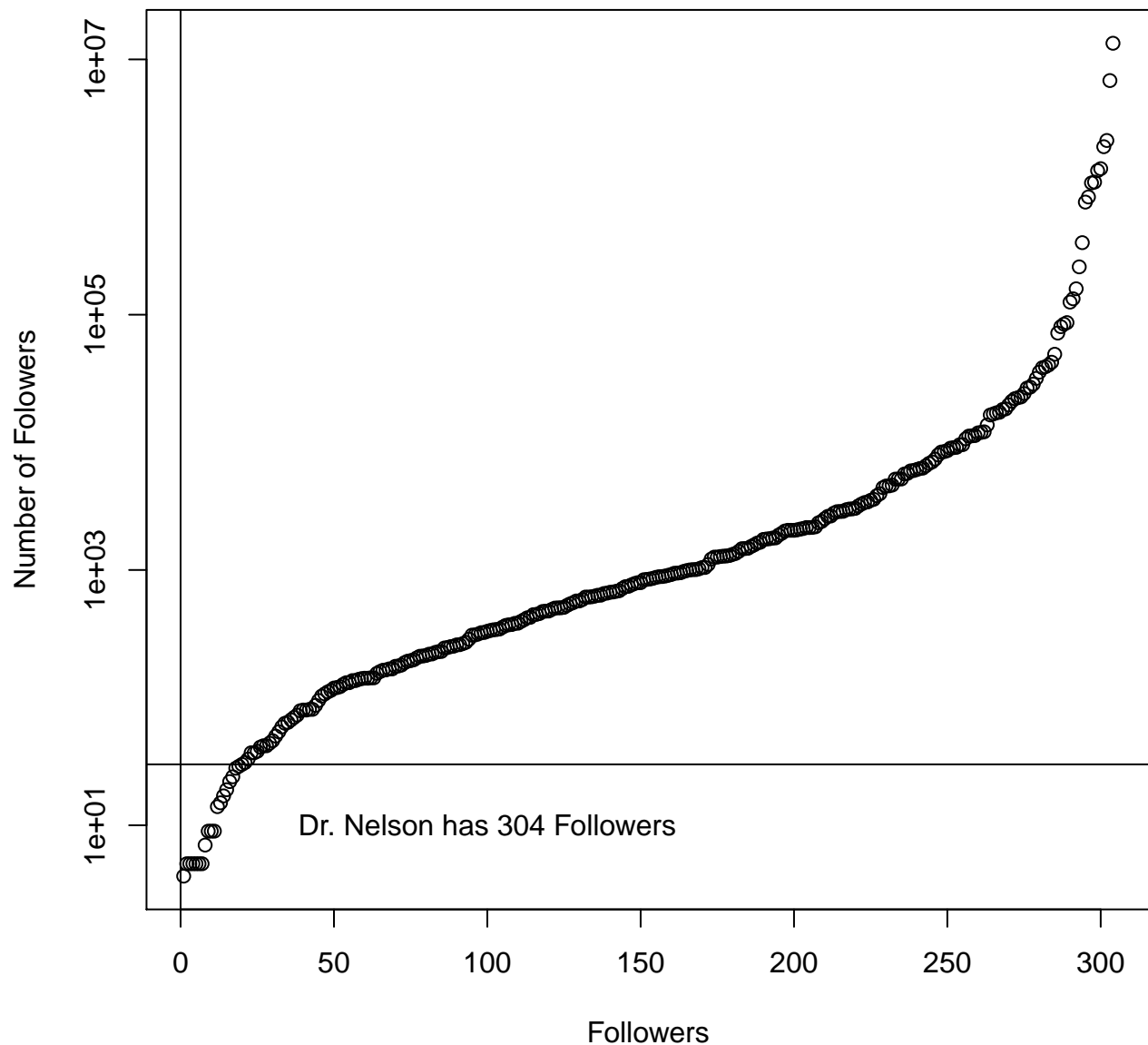
getFriendsDict('phonedude_mln')

```

Table 3: Subset of Dr Nelson's Twitter Friends and their respective Friends Count

Friend	FriendsCount
KariHeffner	4
Past_Pages	5
ThoughtsFromBEL	5
normeu	5
involutish	5
karensnet	5
awptix	5
DanMilanko	7
SimpleSimon2013	9
AmberBoehnlein	9
neiltyson	6829297
UberFacts	13388960

Dr. Nelson's Twitter Follower vs Number of Followers



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://dev.twitter.com/docs/api/1.1/get/followers/list>

Solution 2:

1. In other to retrieve Dr. Nelson’s followers, I used Dr. Nelson’s Twitter (phonedude_mln) account and

used a Twitter api wrapper-Tweepy to extract his followers. The function *getFriendsDict* retrieves all Dr. Nelson's followers. This is demonstrated in listing 3.

2. After getting the number of friends with *getFriendsDict* function, I used *writeFollowers* function to get Dr. Nelson's followers' count (friend count). This is shown in listing 3.

Table 2 contains a subset of Dr. Nelson's friend and their respective friends count. The complete list of Dr. Nelson's friends and their respective friends count is written in *twitter.mlnFriends.csv*

I wrote an R script to generate the graph as outline in Listing 2, line 10-14.

Table 4: Mean, median and SD numbers of Dr. Nelson's Twitter Friends

Mean	Median	Standard Deviation
110300	847	889156.8

Based on the graph and the statistics Table 4, Dr. Nelson has 304 friends on twitter, but the mean and median of his friends are 110300 and 847 respectively. Hence, the friendship paradox holds.

References

- [1] Python Library. <http://stackoverflow.com/questions/7785831/xml-parsing-in-python-using-beautifulsoup>. Accessed: 2017-25-02.
- [2] Tweepy. <http://www.tweepy.org/>. Accessed: 2017-25-02.
- [3] Twitter API. <https://dev.twitter.com/>. Accessed: 2017-23-02.