CS 595: Assignment 2

Mallika Kogatam

Fall 2014

Contents

1 Problem 1

Question

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

1. Explore the friendship paradox for your Twitter account. Since Twitter has directional links (i.e., "followers" and "following"), we'll be investigating if the people you follow (Twitter calls these people "friends") follow more people than you. If you are following < 50 people, use my twitter account "phonedude_mln" instead of your own.

Create a graph of the number of friends (y-axis) and the friends sorted by number of friends (x-axis). (The friends don't need to be labeled on the x-axis as "Bob", "Mary", etc. -- just 1, 2, 3 ...) In other words, if you have 100 friends your x-axis will be 1..101 (100 + you), and the y-axis value will be number of friends that each of those friends has. The friend with the lowest number of friends will be first and the friend with the highest number of friends will be last.

Do include yourself in the graph and label yourself accordingly. Compute the mean, standard deviation, and median of the number of friends that your friends have.

The appropriate part of the Twitter API to use is:

https://dev.twitter.com/rest/reference/get/friends/list

1. This assignment was comparatively easy when compared to the second assignment.
2. I used twittersearch library to get 1000 URIs in second assignment, but to get the list of friends in the twitter this library is of no help.
3. So I figured tweepy does that job with out any trouble.
4. As I already have the consumer key and tokens I used them here in order get the data.
5. In tweepy there are these predefined functions "friends_count" which will give the count of the friends whom we follow and function "friends" gives the list of friends for a particular user.
6. Function "friends" by default gives only list of 20 friends. After a little research I figured we can set the list to a desired count.
7. My twitter account is very new so I don't have many friends so I used "phonedude_mln"
8. I wrote a Python program to get the list of friends and their friends. The output(followed.txt) from this program is not sorted.
9. So I sorted the list by using a simple command in Linux. The Linux command is as below and the result is stored in followed-sorted.txt file.

```
sort -n -k1 followed.txt > followed-sorted.txt
```

10. "Phonedude_mln" had 148 friends in total. But there is an other person who have the same number of friends so when I am highlighting "phonedude_mln" two bars in the bar plot got highlighted.
11. I calculated Mean , Standard Deviation and median in R using simple commands as shown in Listing 2.
    - Mean: 404.926174496644
    - Median: 191
    - Standard Deviation: 545.993999248776
12. I plotted the graph in scatter plot which can be seen Figure 2, but it is not as clear as the bar plot.

```
#! /usr/bin/env python

import tweepy
import time
import sys


CONSUMER_KEY = "1QjOMB7IZ00zaTvPmZ6tUrR2R"
CONSUMER_SECRET = "4aolvqxZYw3XwyrzWjiHb6aFAlg7iNodyxmyiIyX8NowefLL53"
OAUTH_TOKEN = "2820592280-iVQDNegTG1CtQLuQ14N9kwIefiVPSTA75zAKeye"
OAUTH_TOKEN_SECRET = "oaUFcXLImxaL0XGUvy3nT1XfBXkVkLNWDmRHkWvvDCozi"

def main():
    auth = tweepy.auth.OAuthHandler(CONSUMER_KEY,CONSUMER_SECRET)
    auth.set_access_token(OAUTH_TOKEN,OAUTH_TOKEN_SECRET)

    api = tweepy.API(auth)

    followed = []

    out_file = open( "followed.txt", "w" )

    user = api.get_user("phonedude_mln")


    out_file.write( "{} {}\n".format( user.friends_count, user.screen_name ) )

    for friend in user.friends(count = 5000):
        out_file.write( "{} {}\n".format( friend.friends_count, friend.screen_name ) )

    out_file.close()



if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

Listing 1: Python Program for printing the list of followers

```
1
2  data <- read.csv("followed-sorted.txt", stringsAsFactors = F, header = FALSE, sep = " ")
3
4  incdata = data[,1]
5
6  meanOut <- paste("Mean: ", mean(incdata), collapse = "")
7
8  medianOut <- paste("Median: ", median(incdata), collapse = "")
9
10 sdOut <- paste("Std Dev: ", sd(incdata), collapse = "")
11
12 write(meanOut, stdout())
13 write(medianOut, stdout())
14 write(sdOut, stdout())
15
16 pdf("barplot.pdf")
17 pos <- (incdata == 148)
18
19 cols <- c("white", "red")
20
21
22 barplot(incdata, main="Friends of Friends on Twitter", xlab="Friends sorted by increasing
       number of friends", ylab="Number of Friends", col=cols[pos + 1],xlim = c(0,175), ylim=c
       (0, 4000))
23 text(x=match(c(148), incdata)+13, y=max(incdata), labels="phonedude_mln", col='red')
24 arrows(x0=match(c(148), incdata)+13, y0=(max(incdata) - 80), x1=match(c(148), incdata)+13,
       y1=175, length=0.1, lwd=3, col='red')
```

Listing 2: R script for statistics and bar plot shown in Figure 1

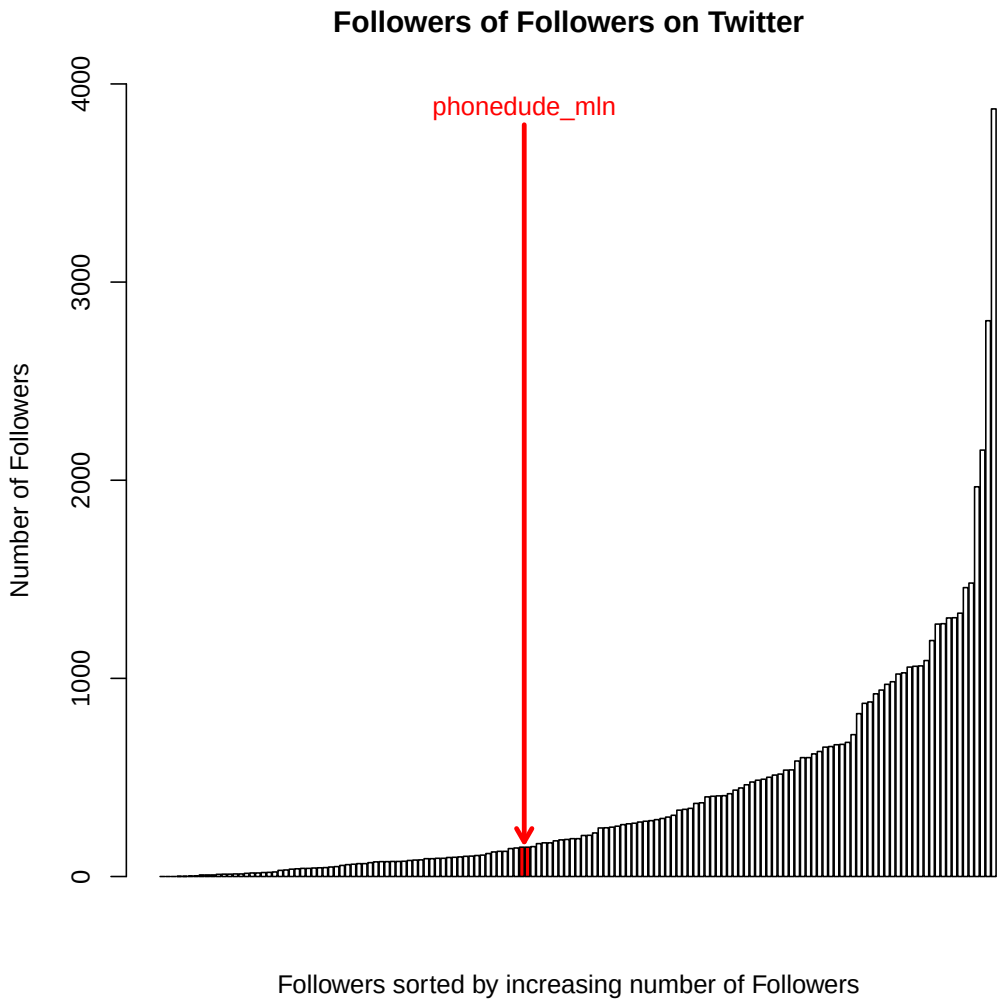

Figure 1: BarPlot showing the count of phonedude_mln's Twitter follower's followers

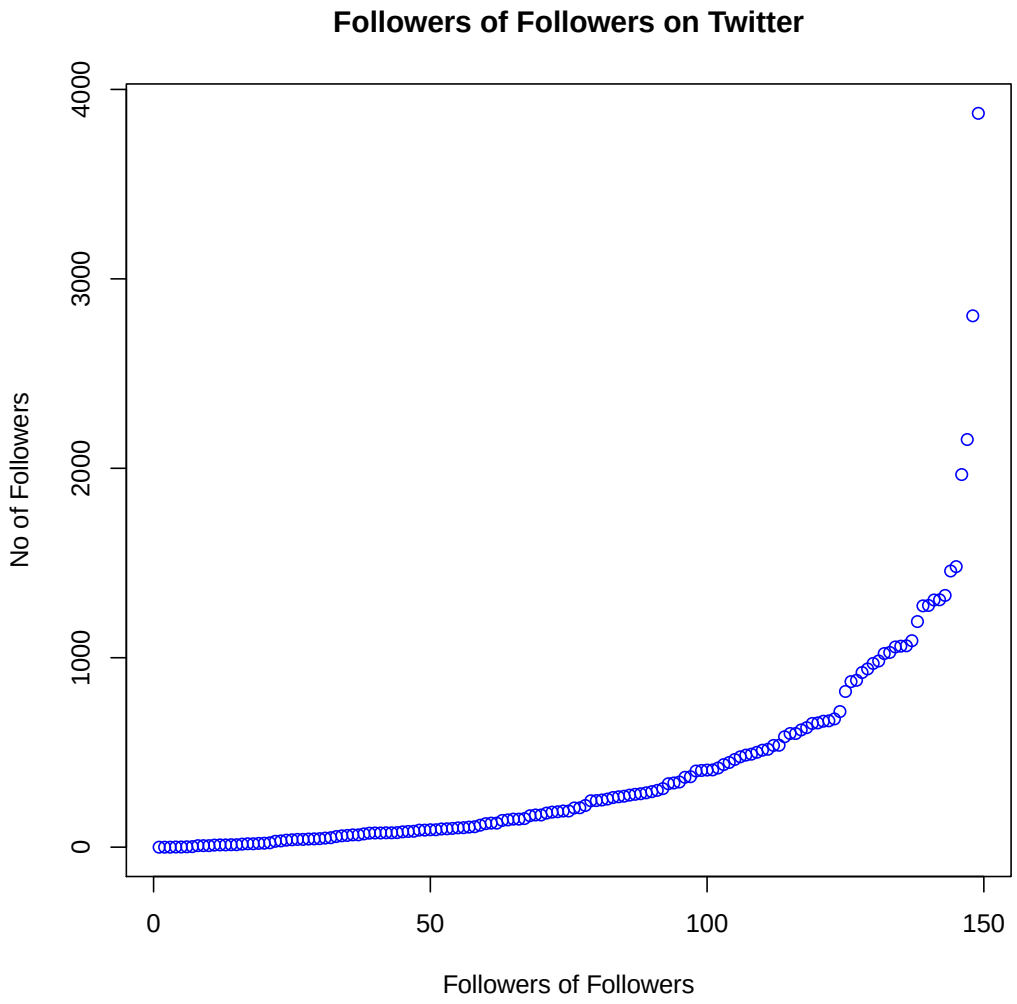

Figure 2: ScatterPlot showing the count of phonedude_mln's Twitter follower's followers

2 Problem 2

Question

```
2.  Using your facebook account, repeat question #1 (if you have >
50 friends).

Start at: https://developers.facebook.com/docs/graph-api/reference/v2.1/user/friends

or perhaps:

http://socialnetimporter.codeplex.com/
```

1. After the second assignment I assumed that we should use API keys in order to get the data from any of the social networking sites. So I spend good amount of time in trying to figure out how to get the information by using these Keys and tokens

2. I registered for Facebook APP in order to get the API keys and wrote a piece of code to get the token.

```
from facepy import utils
app_id = 14867915882649321
app_secret = "93595d85e282ce2bb395a21c921396312"
oath_access_token = utils.get_application_access_token(app_id, app_secret)
print oath_access_token
```

3. I tried using the library "facebook" but it did not work for me, so I shifted to facepy which worked fine for me.

4. But I realized the token I am computing from the above python code is not what I need. I figured that the token I want is generated here `https://developers.facebook.com/tools/explorer/?method=GET&path=1310493851%3Ffields%3Dfriends&version=v2.1` and this expires every couple of hours.

5. The python program 3 gives the list of my friends, but when I observed the output I see only 198 but in real I have 311 friends. By this observation it is clear that I am getting the list of friends who are sharing their friends list.

6. When I tried to find the number of friends each of my friends have through the python program 3 I was shocked to see that the result came up for only 3 of my friends. And for others I get an error"Unsupported Operation".

7. After some research I understood that through tokens we can not get full data from the Facebook profile, the friend count can only be retrieved for those friends who have registered for developer account and allowed access to their friend list.

8. So I ended up using the code 4 ALEX had shared in the group to get the friend count. I retrieved friend count of each of my friends.

9. The output for the code 4 is not formatted. So I wrote a piece of code to format it as I want it. The code is listed in5

10. I sorted the output retrieved from program 5 by a simple command in Linux

```
sort -n -k1 facebook.txt > fbSorted.txt
```

11. seeing as I am only person with 311 friends on my circle, it was easy to color the single bar with red color using the code on lines 11 and 12 in listing 6.

12. some friends do not share their friends counts, so they are left out of the data collected.

13. The median, Standard Deviation and mean is calculated by using the code on lines 4,5 and 6.

    - Mean: 487.505050505051
    - Median: 409.5
    - Standard Deviation: 389.037217791762

14. I tried plotting the graph in scatter plot but I liked the bar plot better.

```
#import facebook
from facepy import GraphAPI
from facepy import utils

def main():
    app_id = 1486791588264932
    app_secret = "93595d85e282ce2bb395a21c92139631"
    oath_access_token = utils.get_application_access_token(app_id, app_secret)

    token = "CAACEdEose0cBAHdKj7pO0wWcQ1l7Nk5SXOUZAWdIEOZCvFg6JuL8IRJQcTJZBg
    GpNgoO5M2d3nZBYOhOdnDgn9gJc9mZCk8bC9N5yIMAEVpZAWX8pgIv
    CSZCGWu6vkfX5I1VDCfkyLHaKkxpgLOyTsnOCHuWkiBrjlW5dN1IHNLmZ
    C168XZBMJIc1r5UA7ssNZAUhJPsbNzaR1rer93oSuQA4kZAfNIjGCp95IZD"

    print oath_access_token

    graph = GraphAPI(token)

    for friendlist in graph.get('me/friendlists/')["data"]:
        print

        for friend in graph.get( friendlist["id"] + "/members" )["data"]:
            print friend["name"]

            try:
            print len( graph.get(friend["id"] + "/friends/")["data"] )
                print len(frnd_friendlist)
            except :
                print 0


if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

Listing 3: Python program for getting the list of Friends by using facepy

```
#credit to:
#https://gist.github.com/leostera/3535568
#https://pypi.python.org/pypi/selenium
#cookies problem: http://stackoverflow.com/questions/7854077/using-a-session-cookie-from-selenium-in-urllib2
#http://stackoverflow.com/questions/14583560/selenium-retrieve-data-that-loads-while-scrolling-down

from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
from selenium.webdriver.common.by import By
import os, sys
from bs4 import BeautifulSoup
import codecs
from random import randint
import getpass
import os


globalHtmlOutputFile = 'allFacebookFriends.html'
globalCSVOutputFile = 'facebookFriendFriendsCountTuples.txt'

#output file: globalHtmlOutputFile
def getHtmlOfAllFriends(userFaceBookEmail, userFaceBookPassword):

        if( len(userFaceBookEmail) > 0 and len(userFaceBookPassword) > 0 ):
                pass
        else:
                print "userFaceBookEmail and/or userFaceBookPassword: bad length"
                return

        try:
                htmlOutputFile = open(globalHtmlOutputFile, 'w')
        except:
                exc_type, exc_obj, exc_tb = sys.exc_info()
                fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
                print fname, exc_tb.tb_lineno, sys.exc_info()
                return

        myFirefoxBrowser = webdriver.Firefox()
        myFirefoxBrowser.implicitly_wait(3)
        # or you can use Chrome(executable_path="/usr/bin/chromedriver")
        myFirefoxBrowser.get("http://www.facebook.org")
        assert "Facebook" in myFirefoxBrowser.title


        elem = myFirefoxBrowser.find_element_by_id("email")
        elem.send_keys(userFaceBookEmail)
        elem = myFirefoxBrowser.find_element_by_id("pass")
        elem.send_keys(userFaceBookPassword)
        elem.send_keys(Keys.RETURN)


        #http://stackoverflow.com/questions/7854077/using-a-session-cookie-from-selenium-in-urllib2
        all_cookies = myFirefoxBrowser.get_cookies()
        #cookies = {}
        #for s_cookie in all_cookies:
        #    cookies[s_cookie["name"]]=s_cookie["value"]


        #open friends page
        friendsLink = 'https://www.facebook.com/friends/'
        myFirefoxBrowser.get(friendsLink)
        myFirefoxBrowser.maximize_window()


        #scroll to bottom of page
        previousCountOfFriends = -1
        while True:
```

```
69
70                  myFirefoxBrowser.execute_script("return window.scrollTo(0, document.body.
                        scrollHeight);")
71                  html = myFirefoxBrowser.page_source.encode('utf-8')
72
73                  soup = BeautifulSoup(html)
74                  parentOfUIProfileBlockContent = soup.findAll('div', { 'class' : '
                        uiProfileBlockContent' })
75
76                  #lastIndexOfFriends = html.rfind('<div class="uiProfileBlockContent">')
77                  lastIndexOfFriends = len(parentOfUIProfileBlockContent)
78
79                  #'Friends' not found
80                  if( lastIndexOfFriends == -1 ):
81                          break
82
83                  #No new entry
84                  if( previousCountOfFriends == lastIndexOfFriends ):
85                          htmlOutputFile.write(html)
86                          break
87                  else:
88                          previousCountOfFriends = lastIndexOfFriends
89
90                  sleepTime = randint(3,7)
91                  print "...sleeping for", sleepTime, "seconds"
92                  time.sleep(sleepTime)
93
94
95          myFirefoxBrowser.close()
96          return previousCountOfFriends
97
98  def getCredentials():
99
100         userName = ''
101         password = ''
102
103         try:
104                 credentialsFile = open('credentials.txt')
105                 credInfo = credentialsFile.readlines()
106
107                 if( len(credInfo) > 1 ):
108                         userName = credInfo[0]
109                         password = credInfo[1]
110         except:
111                 exc_type, exc_obj, exc_tb = sys.exc_info()
112                 fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
113                 print fname, exc_tb.tb_lineno, sys.exc_info()
114                 return
115
116
117         return userName, password
118
119 def getFBHtmlDump(inputFileName):
120
121         htmlText = ''
122
123         if( len(inputFileName) > 0 ):
124                 try:
125                         inputFile = open(inputFileName, 'r')
126                         htmlText = inputFile.read()
127                 except:
128                         exc_type, exc_obj, exc_tb = sys.exc_info()
129                         fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
130                         print fname, exc_tb.tb_lineno, sys.exc_info()
131
132
133         return htmlText
134
135 #writes tuples <friend, friendCount> into globalCSVOutputFile
136 def getFriendOfFriendsFromHtml(htmlText):
137
138         goAheadFlag = False
```

```
139
140         if( len(htmlText) > 0 ):
141
142                 try:
143                         outputFile = codecs.open(globalCSVOutputFile, 'w', 'utf-8')
144                         outputFile.write('"USER", "FRIENDCOUNT"\n')
145                 except:
146                         exc_type, exc_obj, exc_tb = sys.exc_info()
147                         fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
148                         print fname, exc_tb.tb_lineno, sys.exc_info()
149                         return
150
151
152                 soup = BeautifulSoup(htmlText)
153                 parentOfUIProfileBlockContent = soup.findAll('div', { 'class' : '
                        uiProfileBlockContent' })
154
155                 for profile in parentOfUIProfileBlockContent:
156
157                                 friendName = profile.find('div', { 'class' : 'fsl fwb fcb'
                                        })
158                                 potentialFriendsCount = profile.find('a', { 'class' : '
                                        uiLinkSubtle' })
159
160                                 #potentialFriendsCount: x (f)riends | x mutual friends, etc,
                                        so split
161                                 if( potentialFriendsCount is not None ):
162
163                                         potentialFriendsCount = potentialFriendsCount.text.
                                                split(' ')
164
165                                         if( len(potentialFriendsCount) > 1 ):
166                                                 if( len(potentialFriendsCount[1]) > 0):
167                                                         if( potentialFriendsCount[1][0].
                                                                lower() == 'f' ):
168
169                                                                 friendCount =
                                                                        potentialFriendsCount
                                                                        [0].replace(',','')
170
171                                                                 stringToWrite = friendName.
                                                                        text + ', ' +
                                                                        friendCount + '\n'
172                                                                 outputFile.write(
                                                                        stringToWrite)
173                                                                 goAheadFlag = True
174
175
176                 outputFile.close()
177
178
179         return goAheadFlag
180
181 if __name__ == "__main__":
182
183
184         print ''
185
186         print 'Welcome to get fb friends of friends. If all goes well,'
187         print 'The application will write your fb friends of friends into ./' +
                globalCSVOutputFile
188
189         print ''
190         userNameFacebook = raw_input("Email ID: ")
191         passwordFacebook = getpass.getpass('Password: ')
192
193         userNameFacebook = str(userNameFacebook)
194         passwordFacebook = str(passwordFacebook)
195
196         userNameFacebook = userNameFacebook.strip()
197         passwordFacebook = passwordFacebook.strip()
198
```

```
199         intGoAheadFlag = getHtmlOfAllFriends(userNameFacebook, passwordFacebook)
200
201         if ( intGoAheadFlag > -1 ):
202                 facebookDumpInputFileName = globalHtmlOutputFile
203                 htmlText = getFBHtmlDump(facebookDumpInputFileName)
204                 boolGoAhead = getFriendOfFriendsFromHtml(htmlText)
205
206                 #open file
207                 if( boolGoAhead ):
208                         myFirefoxBrowser = webdriver.Firefox()
209                         filePath = 'file:///' + os.getcwd() + '/' + globalCSVOutputFile
210                         myFirefoxBrowser.get(filePath)
```

Listing 4: Python program for getting the list of Friends by using selenium

```
#! /usr/bin/env python

import sys


def main():
    file_in = open( "fb_frnds.txt", "r" )
    file_out = open( "facebook.txt", "w" )

    counter = 1

    for line in file_in:
        line = line.strip().split(",")

        line = [ x.strip() for x in line ]

        file_out.write( "{} {}\n".format( line[1], line[0].replace(" ","") ) )


    file_out.close()


if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

Listing 5: Python program for formatting the list

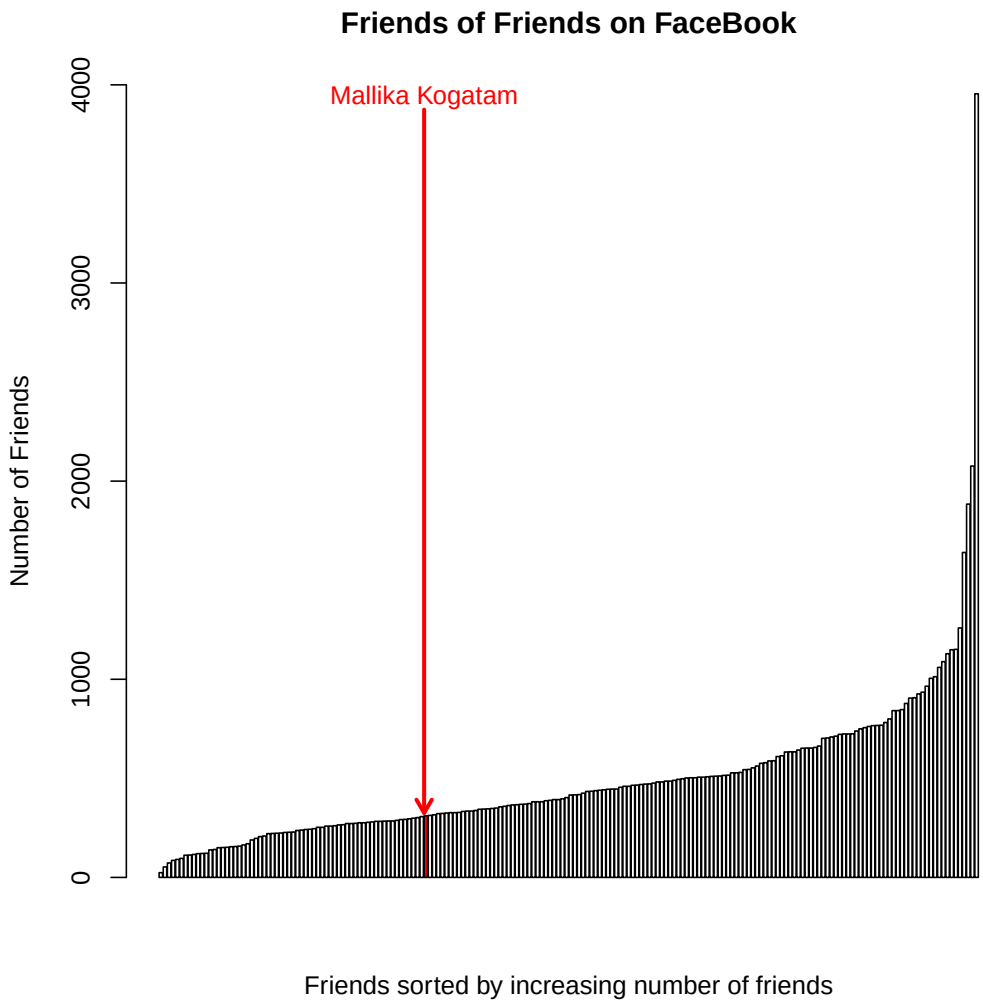

Figure 3: BarPlot for no of friends and the friends sorted by number of friends

```
1
2  data <- read.csv("fbSorted.txt", stringsAsFactors = F, header = FALSE, sep = " ")
3  incdata = data[,1]
4  meanOut <- paste("Mean: ", mean(incdata), collapse = "")
5  medianOut <- paste("Median: ", median(incdata), collapse = "")
6  sdOut <- paste("Std Dev: ", sd(incdata), collapse = "")
7  write(meanOut, stdout())
8  write(medianOut, stdout())
9  write(sdOut, stdout())
10 pdf("fb_barplot.pdf")
11 pos <- (incdata == 311)
12 cols <- c("white", "red")
13 barplot(incdata, main="Friends of Friends on FaceBook", xlab="Friends sorted by increasing
       number of friends", ylab="Number of Friends", col=cols[pos + 1], ylim=c(0, 4000))
14 text(x=match(c(311), incdata)+12, y=(max(incdata)-20), labels="Mallika Kogatam", col='red')
15 arrows(x0=match(c(311), incdata)+12, y0=(max(incdata) - 80), x1=match(c(311), incdata)+12,
       y1=325, length=0.1, lwd=2.5, col='red')
```

Listing 6: R script for bar plot shown in Figure 3

Bibliography

[1] Facebook api. http://stackoverflow.com/questions/12676634/python-facebook-api-showing-invalid-token.

[2] Facebook api example. http://stackoverflow.com/questions/11510850/python-facebook-api-need-a-working-example.

[3] Facebook graph api. https://developers.facebook.com/bugs/356511554434996.

[4] How to get facebook access token. http://stackoverflow.com/questions/16739894/how-to-get-facebook-access-token-using-python-library.

[5] Quatation marks. http://www.maths.tcd.ie/ dwilkins/LaTeXPrimer/QuotDash.html.

[6] Using graph api. https://developers.facebook.com/docs/graph-api/using-graph-api/v2.1.

[7] Johannes Gorset. Facepy graph api, november 2011.

[8] Joshua Roesslein. *tweepyDocumentation*, chapter 1, pages 3–4. 2014.