# INTRODUCTION TO WEB SCIENCES:
# Assignment 5

Babitha Bokka

17 October 2014

# Contents

# 1 Question 1: Twitter Friendship Paradox

Explore the friendship paradox for twitter account.

## 1.1 Approach

Intial step taken is looking for Tweepy Twitter API, found a good documentation to extract the friends (people you follow )from twitter account.Since my twitter account does not have more than 50 friends I used "phonedude_mln" and extracted friends(whom phonedude_mln follow ).

## 1.2 Description of searchfriends.py

1. The program needs four Twitter API keys.
2. Using the keys we interact with twitter API.
3. Using the pre-defined methods we get the screen name and friend friends count.
4. Saved the results to friendsCount.txt.
5. Using the sort command sorted the friends count.
6. Saved them to friendsCount.txt.sort.

## 1.3 Description of sortFriends.py

1. The sorted file is given as input to the program.
2. The screen name (friends name) is given an ID and the ID and the count are saved to a file.
3. To maintain a record which user is assigned which ID a log file numberNameCount.txt is created.
4. The final friend ID and the count is saved to finalSorted.txt

## 1.4 Source Code

### 1.4.1 searchFriends.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import os, sys

import tweepy
import unicodedata
from tweepy import Stream
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener


consumer_key        = 'fpTauqKqCRj4Gp8m9jb9WCilk'
consumer_secret     = 'OrDd7NssqrvLgOXnzuDkGcS8UbTNoY1jFYJF0HS6daxELfyI2k'
access_token        = '2822384568-jleRlhWap2Y7SMDW9y9tXkji95GHYDJPHK2IZ0b'
access_token_secret = 'eVWGqNuLEk7xG1t47vLSkwBhJ6cQyNbeiZGShdRZXKF2A'


#Main Function
def main () :
  saveFile  = open('friendsCount.txt','w')
  # getting access to the twitter api
  auth      = tweepy.OAuthHandler(consumer_key, consumer_secret)
  auth.set_access_token(access_token, access_token_secret)
  api       = tweepy.API(auth)
  user      = api.get_user('phonedude_mln')
  userName  = user.screen_name
  userFriendsCount = user.friends_count
  print userFriendsCount , userName
  saveFile.write("{:<10} {} " .format(userFriendsCount ,userName ))
  for friend in user.friends(count = 1000):
    count = friend.friends_count
    name  = friend.screen_name
    print count , name
    saveFile.write('\n')
    saveFile.write("{:<10} {} " .format(count ,name ))
        saveFile.write('\n')
        saveFile.close()


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

### 1.4.2 sortingCommand.txt

```
sort -n friendsCount.txt -o friendsCount.txt.sort
```

### 1.4.3 sortFriends.py

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

import os
import sys

#Main Function
def main () :
    saveFile       = open('friendsCount.txt.sort','r')
    writeFile      = open('numberNameCount.txt','w')
    writeFile.write("{:<10} {:<10} {} " .format('friendId' , 'friendFriendsCount', '
    friendName'))
    writeFinalFile = open('finalSorted.txt','w')
    writeFinalFile.write("{:<10} {} " .format('friendId ', 'friendFriendsCount'))
    friendId          = 0
    for line in saveFile.readlines():
        nameCount          = line.split()
        friendFriendsCount = nameCount[0]
        friendName         = nameCount[1]
        print friendFriendsCount , friendName , friendId
        friendId += 1
        writeFile.write('\n')
        writeFinalFile.write('\n')
        writeFile.write("{:<10} {:<10} {} " .format(friendId , friendFriendsCount ,
    friendName))
        writeFinalFile.write("{:<10} {} " .format(friendId , friendFriendsCount))
    writeFile.close()
    writeFinalFile.close()


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

## 1.5 Input

To accomplish the task "phonedude_mln" twitter account is used.

## 1.6 Output Files

### 1.6.1 friendsCount.txt

The file contains the "phonedude_mln" friends screen name and the count of their friends.

```
count       friends_name
0           511hamptonroads
0           AcademicsSay
0           TPDL2012
1           CommonwlthCup
1           W3CDataontheWeb
2           edwardafox
```

### 1.6.2 friendsCount.txt.sort

Sorting the friends based on the number of friends.

```
count       friends_name
0           511hamptonroads
0           AcademicsSay
0           TPDL2012
1           CommonwlthCup
1           W3CDataontheWeb
2           edwardafox
```

### 1.6.3 finalSorted.txt

The file contains the friend ID and number of friends for each friend.

```
friendId    friendFriendsCount
1           0
2           0
3           0
4           1
5           1
6           2
```

### 1.6.4 numberNameCount.txt

This is a log file which contains ID given to each friend, the count of friend friends, and screen name.

```
friendId    friendFriendsCount  friendName
1           0           511hamptonroads
2           0           AcademicsSay
3           0           TPDL2012
4           1           CommonwlthCup
5           1           W3CDataontheWeb
6           2           edwardafox
```

## 1.7 Scatterplot

### 1.7.1 Code to generate the Scatterplot scatterCommands.txt

```
Twitter ScatterPlot with points :
> plot(type="p" ,finalSorted$friendId , finalSorted$friendFriendsCount, xlab="Friends
    ",ylab="Number of friends",main ="Friendship Paradox",las=1,xlim=c(1,149),ylim=c
    (1, 4000),col=2,pch=16,cex=0.5)
> abline(v=66 , col=4 , lw=0.5)

Twitter ScatterPlot with line :
> plot(type="l" ,finalSorted$friendId , finalSorted$friendFriendsCount, xlab="Friends
    ",ylab="Number of friends",main          ="Friendship Paradox",las=1,xlim=c(10,149)
    ,ylim=c(1, 4000),col=2,pch=16,cex=0.5)
> abline(v=66 , col=4 , lw=0.5)
```

### 1.7.2 Description of graph

Figure 1 brings up the relation between the friends and number of friends. Figure 1 and Figure 2 are plotted on the same data but Figure 2 gives clear visualization.

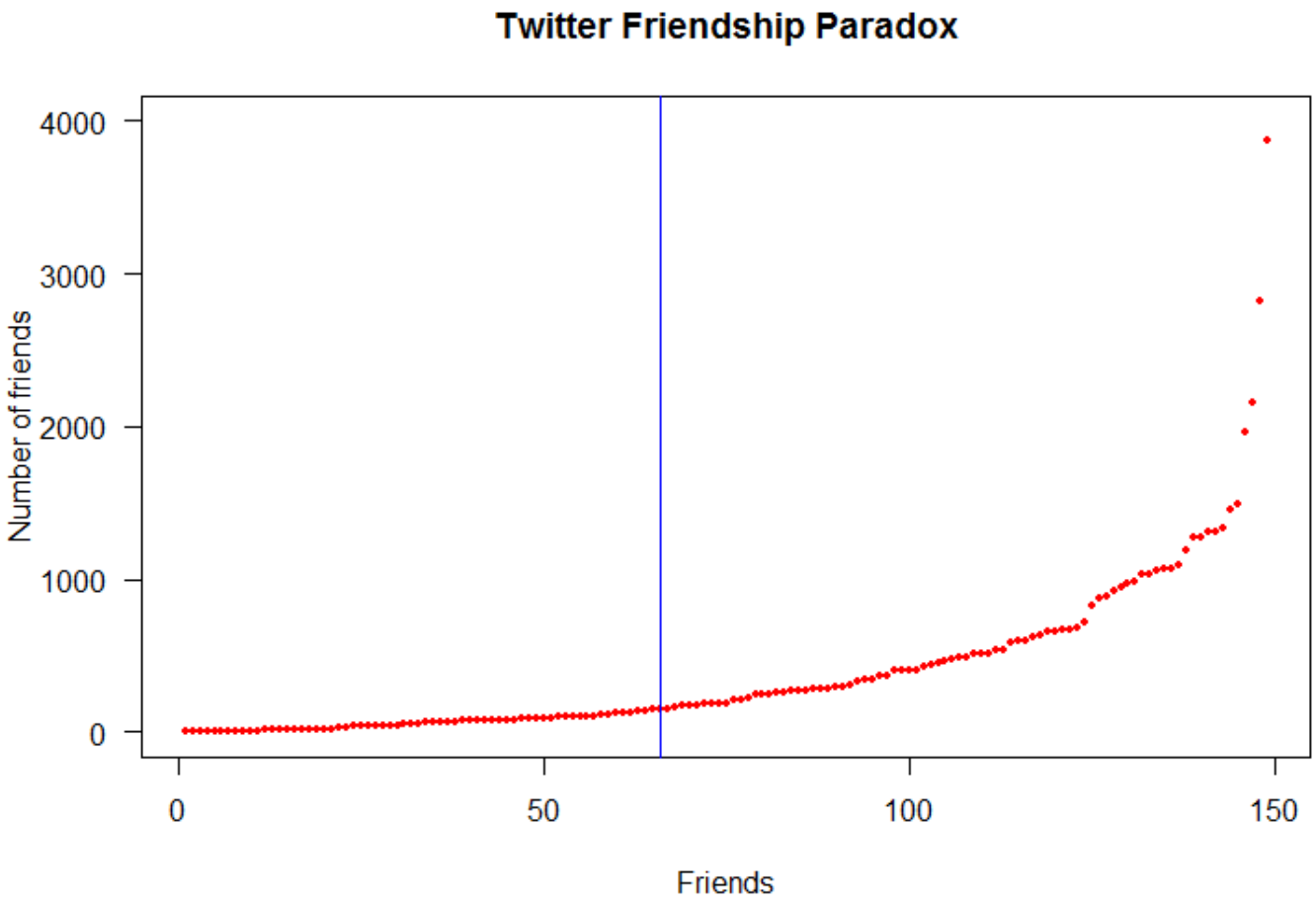


Figure 1: Twitter friendship paradox

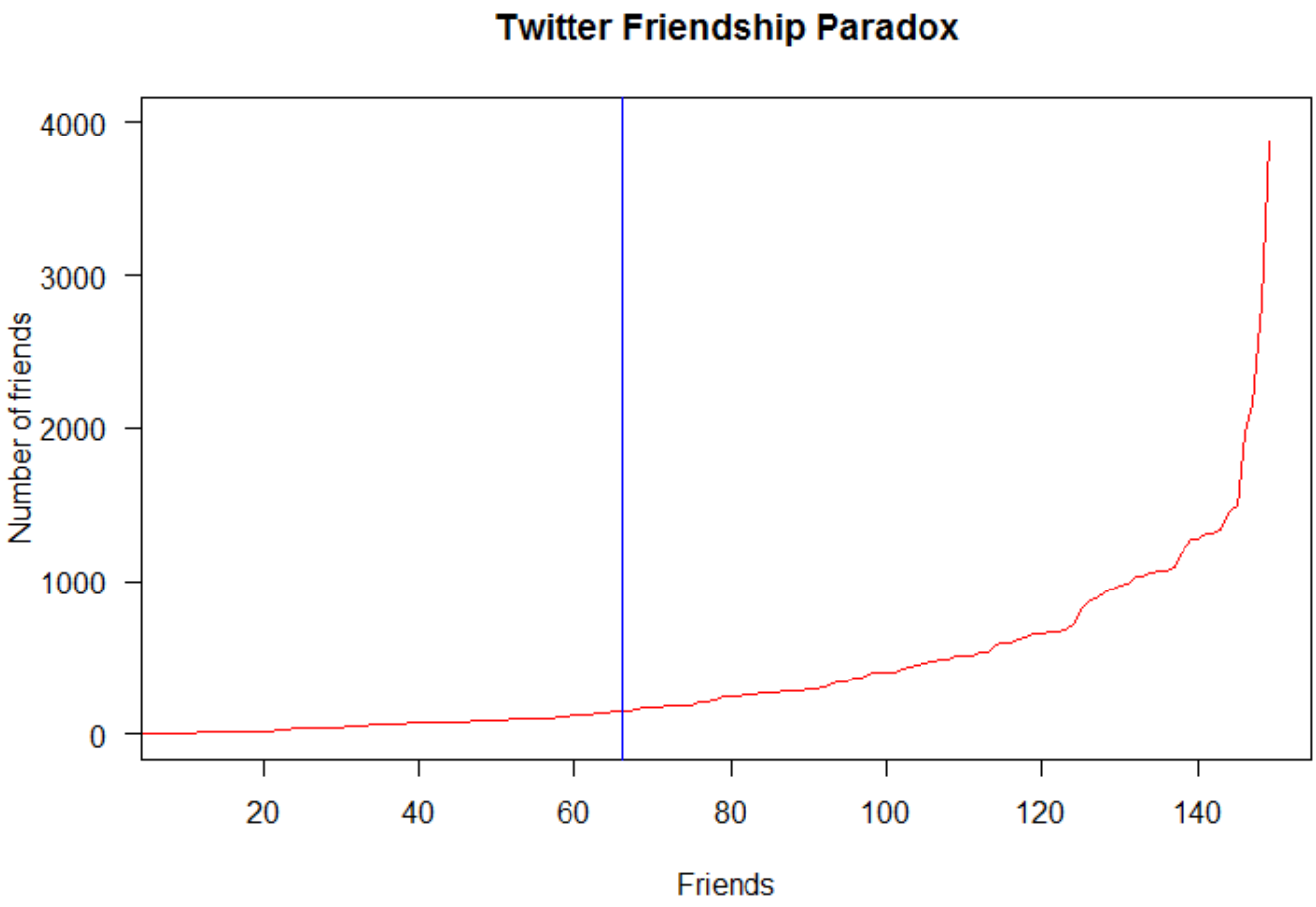


Figure 2: Twitter friendship paradox

## 1.8 Calculations

The mean, median, and standard deviation are calculated by using mosaic on "R".

### 1.8.1 Executed the following commands

install.packages("mosaic")
install.packages("lattice")
require(mosaic)
favstats(finalSorted$friendFriendsCount)

### 1.8.2 Results

Mean : 405.5772
Median : 191
Standard Deviation: 547.0392

# 2 Question 2: Facebook Friendship Paradox

Explore the friendship paradox for Facebook account.

## 2.1 Approach

I didn't follow any approach just used Alex's code. With the knowledge I got from my friends that by using Facebook API keys one cannot extract the number of friends due to several privacy issues.So, I used Alex's program to extract the friends. There was an issue when I try to use the program, the program never get's connected to my Facebook profile because of some security reason and was unable to find what is the reason.So,I used one of my friend's(Ramesh Govindarajulu) account and extracted the data.I did a very small change in the program like the way it was writing to the file.And used my sorting program to sort the friends count and saved results to a file and that it is loaded into "R" and the graphs are generated.

## 2.2 Source Code

### 2.2.1 searchFriends.py

```
#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 BeautifulSoup 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)

```

```
52
53    #http://stackoverflow.com/questions/7854077/using-a-session-cookie-from-selenium-in-
        urllib2
54    all_cookies = myFirefoxBrowser.get_cookies()
55    #cookies = {}
56    #for s_cookie in all_cookies:
57    #    cookies[s_cookie["name"]]=s_cookie["value"]
58
59
60    #open friends page
61    friendsLink = 'https://www.facebook.com/friends/'
62    myFirefoxBrowser.get(friendsLink)
63    myFirefoxBrowser.maximize_window()
64
65
66    #scroll to bottom of page
67    previousCountOfFriends = -1
68    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(10,20)
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 ):
```

```
            potentialFriendsCount = potentialFriendsCount.text.split(' ')

            if( len(potentialFriendsCount) > 1 ):
              if( len(potentialFriendsCount[1]) > 0):
                if( potentialFriendsCount[1][0].lower() == 'f' ):

                  friendCount = potentialFriendsCount[0].replace(',','')

                  stringToWrite = friendCount + ', ' + friendName.text + '\n'
                  outputFile.write(stringToWrite)
                  goAheadFlag = True


      outputFile.close()


    return goAheadFlag

if __name__ == "__main__":


    print ''

    print 'Welcome to get fb friends of friends. If all goes well,'
    print 'The application will write your fb friends of friends into ./' +
     globalCSVOutputFile

    print ''
    userNameFacebook = raw_input("Email ID: ")
    passwordFacebook = getpass.getpass('Password: ')

    userNameFacebook = str(userNameFacebook)
    passwordFacebook = str(passwordFacebook)

    userNameFacebook = userNameFacebook.strip()
    passwordFacebook = passwordFacebook.strip()

    intGoAheadFlag = getHtmlOfAllFriends(userNameFacebook, passwordFacebook)

    if ( intGoAheadFlag > -1 ):
      facebookDumpInputFileName = globalHtmlOutputFile
      htmlText = getFBHtmlDump(facebookDumpInputFileName)
      boolGoAhead = getFriendOfFriendsFromHtml(htmlText)

      #open file
      if( boolGoAhead ):
        myFirefoxBrowser = webdriver.Firefox()
        filePath = 'file:///' + os.getcwd() + '/' + globalCSVOutputFile
        myFirefoxBrowser.get(filePath)
```

### 2.2.2 sortingCommand.txt

```
sort -n facebookFriendFriendsCountTuples.txt -o sort.txt
```

### 2.2.3 sortFriends.py

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

import os
import sys

#Main Function
def main () :
    saveFile       = open('friendsCount.txt.sort','r')
    writeFile      = open('numberNameCount.txt','w')
    writeFile.write("{:<10} {:<10} {} " .format('friendId' , 'friendFriendsCount', 'friendName'))
    writeFinalFile = open('finalSorted.txt','w')
    writeFinalFile.write("{:<10} {} " .format('friendId ', 'friendFriendsCount'))
    friendId           = 0
    for line in saveFile.readlines():
        nameCount          = line.split()
        friendFriendsCount = nameCount[0]
        friendName         = nameCount[1]
        print friendFriendsCount , friendName , friendId
        friendId += 1
        writeFile.write('\n')
        writeFinalFile.write('\n')
        writeFile.write("{:<10} {:<10} {} " .format(friendId , friendFriendsCount , friendName))
        writeFinalFile.write("{:<10} {} " .format(friendId , friendFriendsCount))
    writeFile.close()
    writeFinalFile.close()


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

## 2.3 Input

To accomplish the task "Ramesh Govindarajulu" facebook account is used.

## 2.4 Output Files

### 2.4.1 facebookFriendFriendsCountTuples.txt

The file contains the "Ramesh Govindarajulu" friends name and the count of their friends.

```
count friend_Name
907, Deborah Edds
3956, Kurnia Foe
762, Sherni Minu
153, Aida Gha
266, Babitha Reddy
841, Barbara King
```

### 2.4.2 sortFacebook.txt

Sorting the friends based on the number of friends.

```
count friend_Name
73, Obaid Hafeez Mohammed
116, Alexander II
121, Sruthi Konduru
139, Alex Dohrn
152, Srikanth Bharadwaj
153, Aida Gha
```

### 2.4.3 finalSortedFacebook.txt

The file contains the friend ID and number of friends for each friend.

```
friendId    friendFriendsCount
1           73
2           116
3           121
4           139
5           152
6           153
```

### 2.4.4 numberNameCountFb.txt

This is a log file which contains ID given to each friend, count of friend friends, and user name.

```
friendId    friendFriendsCount friendName
1           73,         Obaid
2           116,        Alexander
3           121,        Sruthi
4           139,        Alex
5           152,        Srikanth
6           153,        Aida
```

## 2.5 Scatterplot

### 2.5.1 Code to generate the Scatterplot scatterCommands.txt

```
Facebook ScatterPlot with points :
plot(type="p" ,finalSortedFacebook$friendId , finalSortedFacebook$friendFriendsCount,
    xlab="Friends",ylab="Number of friends",main="Facebook Friendship Paradox",las=1,
    xlim=c(1,80),ylim=c(1, 4000),col=2,pch=16,cex=0.5)
> abline(v=66 , col=4 , lw=0.5)

Facebook ScatterPlot with line :
plot(type="l" ,finalSortedFacebook$friendId , finalSortedFacebook$friendFriendsCount,
    xlab="Friends",ylab="Number of friends",main="Facebook Friendship Paradox",las=1,
    xlim=c(1,80),ylim=c(1, 4000),col=2,pch=16,cex=0.5)
> abline(v=66 , col=4 , lw=0.5)
```

### 2.5.2 Description of graph

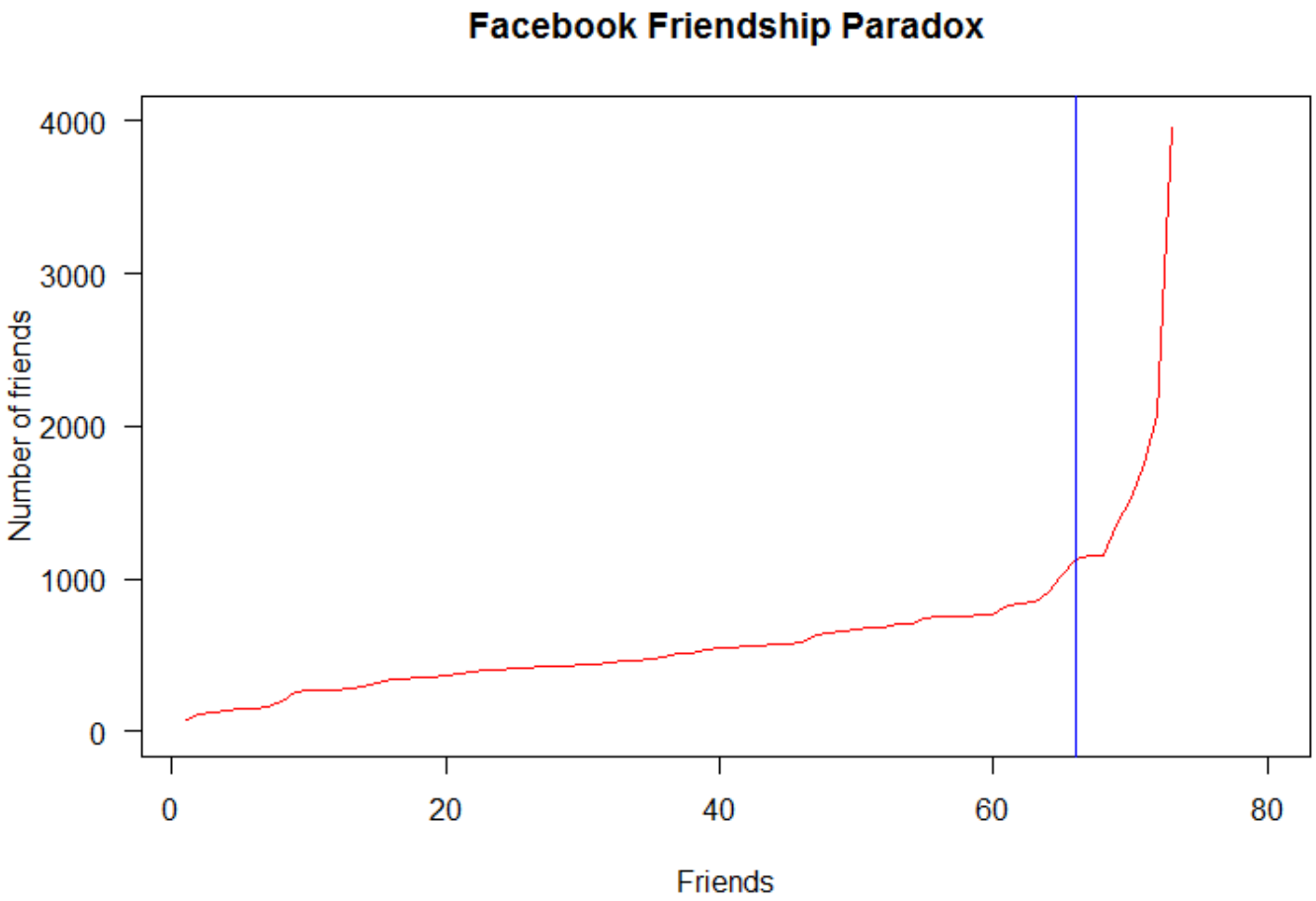


Figure 3: Facebook friendship paradox

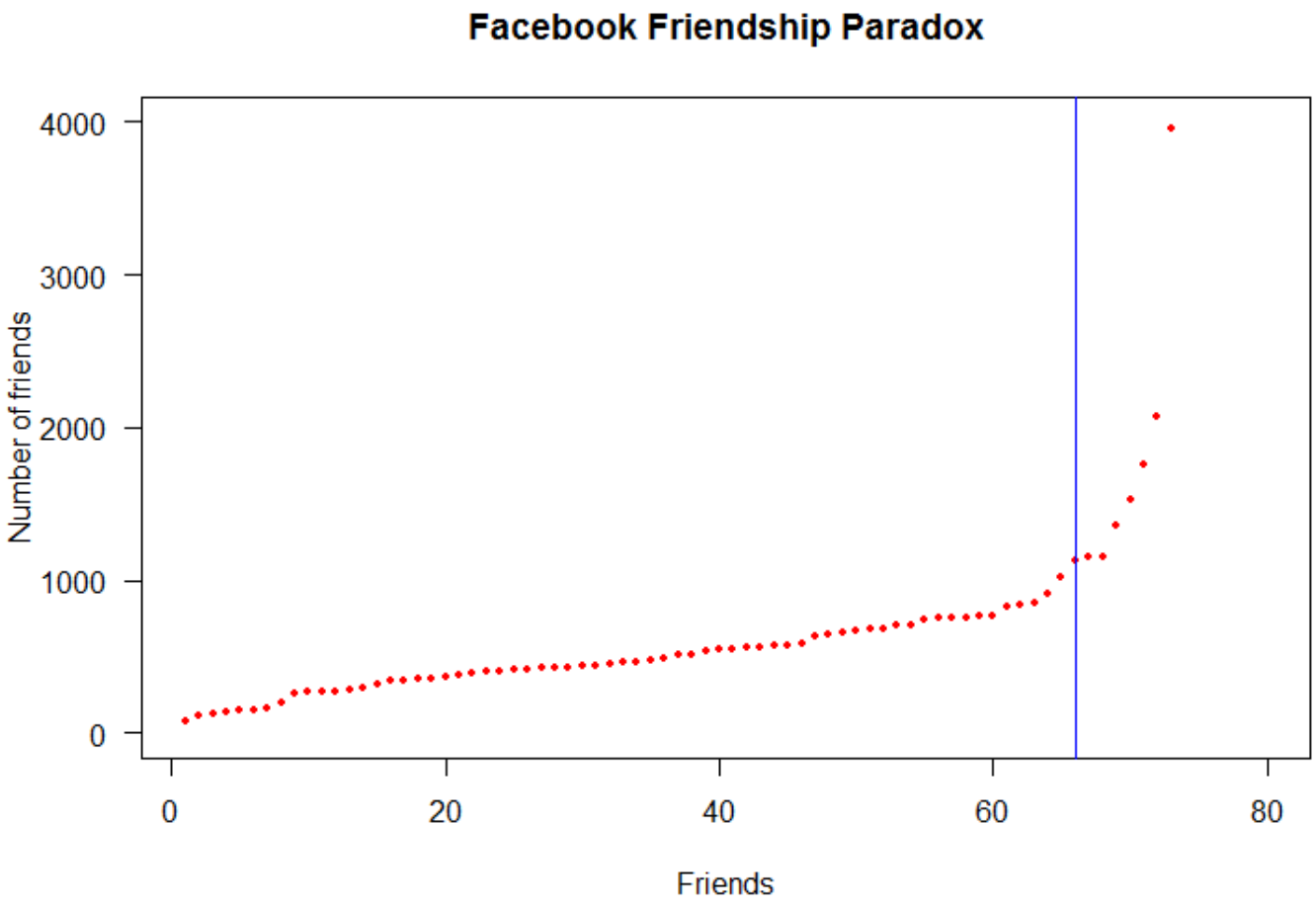


Figure 4: Facebook friendship paradox

## 2.6 Calculations

The mean, median, and standard deviation are calculated by using mosaic on R.

### 2.6.1 Executed the following commands

install.packages(”mosaic”)
install.packages(”lattice”)
require(mosaic)
favstats(finalSorted$friendFriendsCount)

### 2.6.2 Results

Mean : 626.8904
Median : 510
Standard Deviation: 538.8189

# References


[1] Mosaic. http://cran.r-project.org/web/packages/mosaic/mosaic.pdf.

[2] Scatteplotpaper. http://www.calvin.edu/r̃pruim/talks/SC11/Seattle/RatSC11/Master-StatsForScience.pdf.

[3] scatterplot. http://www.r-tutor.com/elementary-statistics/quantitative-data/scatter-plot.

[4] Youtubetutorial. https://www.youtube.com/watch?v=U64yNvlhv9I.

[5] Joshua Roesslein. Tweepydocumentation. https://media.readthedocs.org/pdf/tweepy/latest/tweepy.pdf.


[]