Assignment 5

CS 595: Introduction to Web Science
Fall 2013
Shawn M. Jones
Finished on October 20, 2013

1

Question

1. Determine if the friendship paradox holds for your Facebook account.
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.) 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.

You can download your network in an XML file by using the NameGenWeb
Facebook app:

https://apps.facebook.com/namegenweb/

You will need to give this app permission to access your Facebook
data. Make sure you select "Friend Count" as an Extended Attribute.
When you download the data, download it in the GraphML format.

If you do not have a Facebook account, email me and I will send you
my GraphML file.

Mean	302.555555555556
Median	225.5
Std Dev	236.389147508571

Table 1: Statistics on the count of my Facebook Friends' Friends, values straight from R

Answer

Downloading the graphl file from the NameGenWeb gave me nothing when I anonymized it, so I had to work with the non-anonymized data. The Python script `processFBGraph.py` used to process it is shown in Listing 1. It turns the data into a comma-separated stream that can be output to a file as shown below.

```
./processFBGraph.py not−anonymized−fb−data.graphml > fb−
    frienddata.csv
```

Processing the data was yet another adventure, the script shown in Listing 2, its statistics shown in Table 1, and the graph shown in Figure 1.

The R script runs like so:

```
bash $ −−> ./processFBGraphOutput.R fb−frienddata.csv q1−barplot
    .png 154 'ME!!!'
Mean:   302.555555555556
Median:   225.5
Std Dev:   236.389147508571
null device
          1
```

Seeing as I am the only person with 154 friends on my circle, I was able to color the single bar red using the code on lines 34 and 35 in Listing 2.

Figure 1 shows that I am more popular than about 25% of my friends, but my friends do have more friends than I do. Referencing Table 1, I have fewer friends than the median.

Friends of Friends on Facebook

ME!!!

Number of Friends

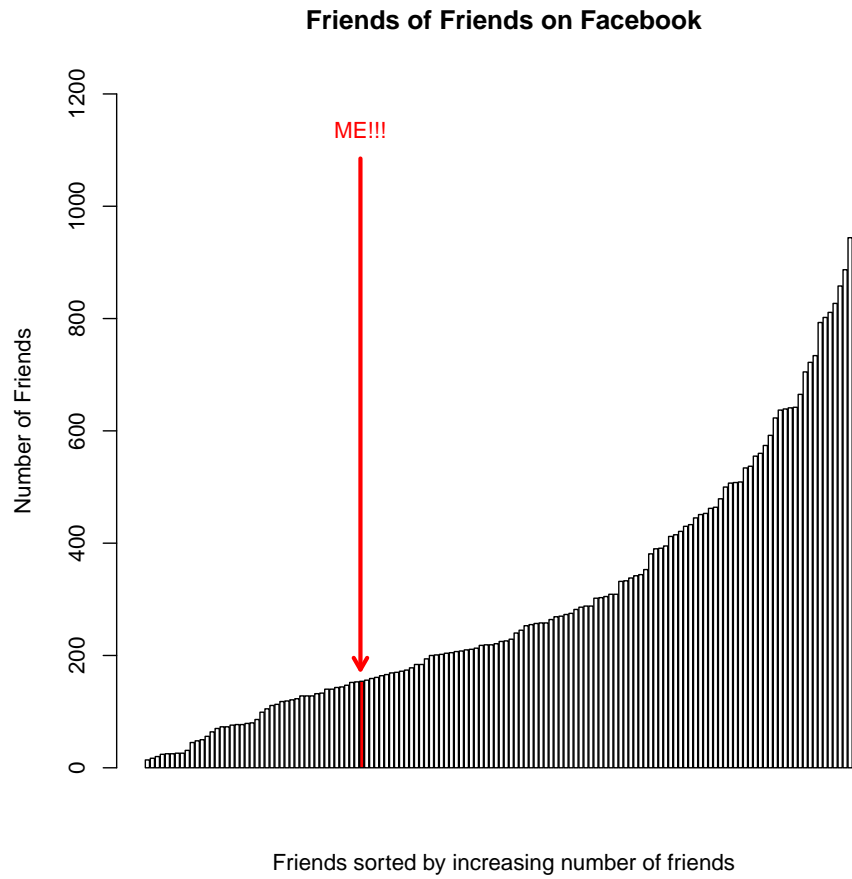Friends sorted by increasing number of friends

Figure 1: Bar plot showing the count of my Facebook Friends' Friends

```
1   #!/usr/local/bin/python3
2
3   import sys
4   from xml.dom.minidom import parseString
5
6   def getFriendInfo(xml):
7
8       dom = parseString(xml)
9       countDict = {}
10
11      for element in dom.getElementsByTagName("data"):
12          if (element.attributes['key'].value == 'name'):
13              name = element.childNodes[0].data
14
15          if (element.attributes['key'].value == 'friend_count'):
16              count = element.childNodes[0].data
17
18              countDict[name] = count
19              name = ''
20              count = ''
21
22      return countDict
23
24  def getFriendCount(xml):
25
26      dom = parseString(xml)
27      return len(dom.getElementsByTagName("node"))
28
29
30  if __name__ == "__main__":
31
32      graphmlFile = sys.argv[1]
33
34      f = open(graphmlFile)
35      xml = f.read()
36      f.close()
37
38      myFriendCount = getFriendCount(xml)
39      friendInfo = getFriendInfo(xml)
40
41      print("Name, Friend Count")
42      print('ME,' + str(myFriendCount))
43
44      for friend in friendInfo:
45          print(friend + ',' + friendInfo[friend])
```

Listing 1: Python program for processing GraphML file from NameGenWeb Facebook App

```
1   #!/usr/bin/Rscript
2
3   args <- commandArgs(trailingOnly = TRUE)
4
5   inputfile <- args[1]
6   outputfile <- args[2]
7   mylocation <- args[3]
8   mytext <- args[4]
9
10  data <- read.csv(inputfile)
11
12  incdata <- sort(data$Friend.Count)
13
14  meanOut <- paste("Mean: ", mean(incdata), collapse = "")
15
16  medianOut <- paste("Median: ", median(incdata), collapse = "")
17
18  sdOut <- paste("Std Dev: ", sd(incdata), collapse = "")
19
20  write(meanOut, stdout())
21  write(medianOut, stdout())
22  write(sdOut, stdout())
23
24  pdf(outputfile)
25
26  # these are used to acquire names for labels later
27  #ndx = order(data$Friend.Count)
28  #xlabels <- data[ndx,]$Name
29
30  # for the coloring of specific bars in the barplot:
31  # http://stackoverflow.com/questions/13112974/change-colours-of-
        particular-bars-in-a-bar-chart
32  # create a vector containing the items equal to my number of
        friends
33  #mylocation = mylocation + 1
34  pos <- (incdata == mylocation)
35  cols <- c("white", "red") # colors to use (first is everyone but
        me)
36
37  # draw the barplot
38  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, max(incdata) +
        100))
39  #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, max(incdata) +
        100), names.arg=xlabels, las=3, cex.names=0.4)
```

```
40
41  # annotation  and  arrow
42  # http :// blog . earlh . com/ index . php /2009/07/ labeling −plots −
        annotations−legends−etc−part−6−in−a−series /
43  text (x=match(c(mylocation), incdata) + 8, y=max(incdata), labels
        =mytext,  col='red ')
44  arrows (x0=match(c(mylocation), incdata) + 8, y0=(max(incdata) −
        50), x1=match(c(mylocation), incdata) + 8, y1=175, length
        =0.1, lwd=3, col='red ')
45
46  dev . off ()
```

Listing 2: R program for bar plot shown in Figure 1

2

Question

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 calcuate 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

If you do not have followers on Twitter (or don't have more than 20), then use my twitter account "phonedude_mln".

Mean	520.846534653465
Median	199
Std Dev	1264.79341369106

Table 2: Statistics on the count of phonedude_mln's Twitter followers' followers, values straight from R

Answer

Because I use Twitter as more of a *content consumption* service, I have very few followers, so few that I lack sufficient sample size to actually answer "do your followers have more followers than you?". Fortunately, I have `phonedude_mln` that I can test with.

The first script, shown in Listing 3, queries the Twitter API for information on the followers of `phonedude_mln` using the function on lines 84-98 and then prints it using the function on lines 100 - 124.

This script is run like below, to produce a CSV file.

```
./countTwitterFollowers.py 2 phonedude_mln > phonedude_mln.csv
```

From the output, I can see that `phonedude_mln` has 201 followers. The R script shown in Listing 4 creates a similar bar plot to that shown in answer one and produces the statistics shown in 2.

```
bash $ ——> ./processTwitterFollowerOutput.R phonedude_mln.csv q2
    −barplot.pdf 201 "phonedude_mln"
Mean:   520.846534653465
Median:   199
Std Dev:   1264.79341369106
null device
          1
```

It turns out that `phonedude_mln` is doing better on Twitter than I am doing on Facebook. He has more followers than the median of 199, but his followers still have more followers than he does.
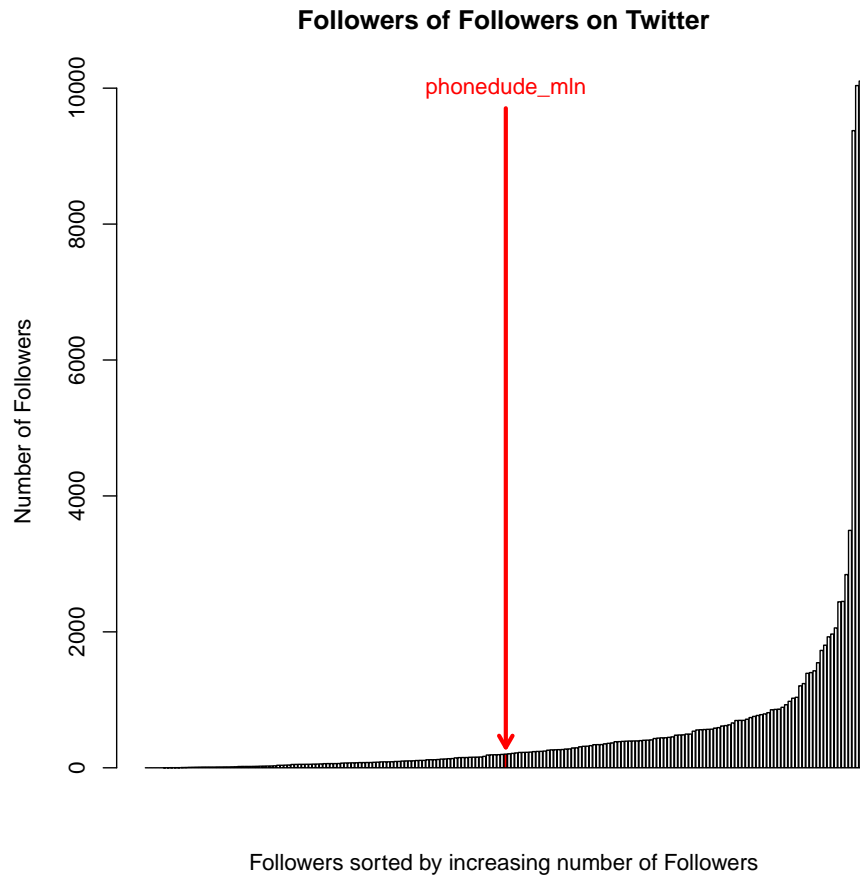
Followers of Followers on Twitter

phonedude_mln

Number of Followers

Followers sorted by increasing number of Followers

Figure 2: Bar plot showing the count of phonedude_mln's Twitter followers' followers

```
 1  #!/usr/local/bin/python3
 2
 3  # −∗− encoding: utf−8 −∗−
 4  from __future__ import unicode_literals
 5  import requests
 6  from requests_oauthlib import OAuth1
 7  from urllib.parse import parse_qs
 8  import json
 9  import time
10  import sys
11
12  # ugly, but necessary, globals; saw no need to change this
13  # strategy from the example
14  REQUEST_TOKEN_URL = "https://api.twitter.com/oauth/request_token
        "
15  AUTHORIZE_URL = "https://api.twitter.com/oauth/authorize?
        oauth_token="
16  ACCESS_TOKEN_URL = "https://api.twitter.com/oauth/access_token"
17
18  CONSUMER_KEY = "n7jt1uMTwGCcIzDvey8g0A"
19  CONSUMER_SECRET = "0r6HUrVD36W4MULgWETKMxrQsCICNy1OFFNc2iW4o"
20
21  OAUTH_TOKEN = "528649269−
        SffJ0Rei5PzLYd2NSJPnnm28dP5nlAnt7E1gRGwo"
22  OAUTH_TOKEN_SECRET = "htrwXF09pS8tP8cMzFrxmMryavdPXd0zPiJHRnLs"
23
24  class APIError(Exception):
25      """
26          If something goes wrong with the API, throw one of these
              .
27          (avoids sys.exit in the middle of the program)
28      """
29
30      def __init__(self, value):
31          self.value = value
32
33      def __str__(self):
34          return repr(self.value)
35
36  def setup_oauth():
37      """
38          Authorize your app via identifier.
39          Code inspired by:
40          http://thomassileo.com/blog/2013/01/25/using−twitter−
              rest−api−v1−dot−1−with−python/
41      """
42
43      # Request token
```

```python
44        oauth = OAuth1(CONSUMER_KEY, client_secret=CONSUMER_SECRET)
45        r = requests.post(url=REQUEST_TOKEN_URL, auth=oauth)
46
47        credentials = parse_qs(r.content)
48
49        resource_owner_key = credentials[b'oauth_token'][0].decode(
              encoding='UTF-8')
50        resource_owner_secret = credentials[b'oauth_token_secret'
              ][0].decode(encoding='UTF-8')
51
52        # Authorize
53        authorize_url = AUTHORIZE_URL + resource_owner_key
54        print('Please go here and authorize: ' + authorize_url)
55
56        verifier = input('Please input the verifier: ')
57        oauth = OAuth1(CONSUMER_KEY,
58                          client_secret=CONSUMER_SECRET,
59                          resource_owner_key=resource_owner_key,
60                          resource_owner_secret=resource_owner_secret,
61                          verifier=verifier)
62
63        # Finally, Obtain the Access Token
64        r = requests.post(url=ACCESS_TOKEN_URL, auth=oauth)
65        credentials = parse_qs(r.content)
66        token = credentials[b'oauth_token'][0].decode(encoding='UTF
              -8')
67        secret = credentials[b'oauth_token_secret'][0].decode(
              encoding='UTF-8')
68
69        return token, secret
70
71
72  def get_oauth():
73        """
74              Code inspired by:
75              http://thomassileo.com/blog/2013/01/25/using-twitter-
                    rest-api-v1-dot-1-with-python/
76        """
77        oauth = OAuth1(CONSUMER_KEY,
78                          client_secret=CONSUMER_SECRET,
79                          resource_owner_key=OAUTH_TOKEN,
80                          resource_owner_secret=OAUTH_TOKEN_SECRET)
81        return oauth
82
83
84  def call_followers_list_api(oauth, count, screenName, cursor):
85        url = \
86              "https://api.twitter.com/1.1/followers/list.json?
                    screen_name=" + \
```

```
87              screenName + "&count=" + str(count) + "&cursor=" +
                    cursor
88
89      response = requests.get( url, auth=oauth )
90
91      if 'errors' in response:
92          raise APIError(
93              json.dumps(
94                  response.json(), sort_keys=True,
95                  indent=4, separators=(',', ': ' ))
96                  )
97
98      return response
99
100  def print_friend_counts(oauth, numberOfCalls, count, screenName)
        :
101
102      cursor="-1"
103
104      print("Name, Friend Count, Followees")
105
106      followers_count = 0
107
108      for i in range(0, numberOfCalls):
109
110          response = call_followers_list_api(oauth, count,
                 screenName, cursor)
111
112          # as per:
113          # https://dev.twitter.com/discussions/1053
114          # friends_count - number of users the user follows
115          # followers_count - number of users that follow the user
116          for entry in response.json()['users']:
117              ident = str(entry['screen_name'])
118              followers = str(entry['followers_count'])
119              print(ident + ',' + followers)
120              followers_count += 1
121
122          cursor = str(response.json()['next_cursor'])
123
124      print(screenName + ',' + str(followers_count))
125
126  def usage():
127
128      print("Usage: " + sys.argv[0] + " <apiCalls> <screenName>")
129
130
131  if __name__ == "__main__":
132
```

12

```
133        #startingid = "400000000000000000"
134        try:
135              apiCalls = int(sys.argv[1])
136              screenName = sys.argv[2]
137        except IndexError as e:
138              usage()
139              sys.exit(1)
140
141        if not OAUTH_TOKEN:
142              token, secret = setup_oauth()
143              print( "OAUTH_TOKEN: " + token )
144              print( "OAUTH_TOKEN_SECRET: " + secret )
145              print( )
146        else:
147              oauth = get_oauth()
148              count = 200
149
150              try:
151                    print_friend_counts(oauth, apiCalls, count,
                            screenName)
152              except APIError as e:
153                    sys.stderr.write(e.value)
154                    sys.exit(254)
```

Listing 3: Python program for acquiring Twitter followers for phonedude_mln

```
1   #!/usr/bin/Rscript
2
3   args <- commandArgs(trailingOnly = TRUE)
4
5   inputfile <- args[1]
6   outputfile <- args[2]
7   mylocation <- as.integer(args[3])
8   mytext <- args[4]
9
10  data <- read.csv(inputfile)
11
12  incdata <- sort(data$Friend.Count)
13
14  meanOut <- paste("Mean: ", mean(incdata), collapse = "")
15
16  medianOut <- paste("Median: ", median(incdata), collapse = "")
17
18  sdOut <- paste("Std Dev: ", sd(incdata), collapse = "")
19
20  write(meanOut, stdout())
21  write(medianOut, stdout())
22  write(sdOut, stdout())
23
24  pdf(outputfile)
25
26  #ndx = order(data$Friend.Count)
27  #xlabels <- data[ndx,]$Name
28
29  # for the coloring of specific bars in the barplot:
30  # http://stackoverflow.com/questions/13112974/change-colours-of-
        particular-bars-in-a-bar-chart
31  # create a vector containing the items equal to my number of
        friends
32  #mylocation = mylocation + 1
33  pos <- (incdata == mylocation)
34  cols <- c("white", "red") # colors to use (first is everyone but
        me)
35
36  # draw the barplot
37  barplot(incdata, main="Followers of Followers on Twitter", xlab=
        "Followers sorted by increasing number of Followers", ylab="
        Number of Followers", col=cols[pos + 1], ylim=c(0, max(
        incdata) + 100))
38  #barplot(incdata, main="Followers of Followers on Twitter", xlab
        ="Friends sorted by increasing number of friends", ylab="
        Number of Friends", col=cols[pos + 1], ylim=c(0, max(incdata)
        + 100), names.arg=xlabels, las=3, cex.names=0.4)
39
```

```
40  # annotation and arrow
41  # http://blog.earlh.com/index.php/2009/07/labeling-plots-
        annotations-legends-etc-part-6-in-a-series/
42  text(x=match(c(mylocation), incdata) + 20, y=max(incdata) - 100,
        labels=mytext, col='red')
43  arrows(x0=match(c(mylocation), incdata) + 20, y0=(max(incdata))
        - 400, x1=match(c(mylocation), incdata) + 20, y1=300, length
        =0.1, lwd=3, col='red')
44
45  dev.off()
```

Listing 4: R program for bar plot shown in Figure 2

3

Question

Extra credit, 2 points:

3. Repeat question #1, but with your LinkedIn profile.

Answer

Not attempted.

4

Question

Extra credit, 1 point:

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

Answer

Not attempted.