Assignment Four

Jacob Berlin

CS432 – Spring 2016

1. 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.


```
<?xml version="1.0" encoding="UTF-8"?>
<data>
    <node id="Simeon_Warner_428351">
        <data key="Label">Simeon Warner</data>
        <data key="uid"><![CDATA[428351]]></data>
        <data key="name"><![CDATA[Simeon Warner]]></data>
        <data key="mutual_friend_count"><![CDATA[13]]></data>
        <data key="friend_count"><![CDATA[244]]></data>
    </node>
    <node id="Drew_Munro_1314586">
        <data key="Label">Drew Munro</data>
        <data key="uid"><![CDATA[1314586]]></data>
        <data key="name"><![CDATA[Drew Munro]]></data>
        <data key="mutual_friend_count"><![CDATA[17]]></data>
        <data key="friend_count"><![CDATA[575]]></data>
    </node>
    <node id="Mat_Kelly_2004483">
        <data key="Label">Mat Kelly</data>
        <data key="uid"><![CDATA[2004483]]></data>
        <data key="name"><![CDATA[Mat Kelly]]></data>
        <data key="mutual_friend_count"><![CDATA[12]]></data>
        <data key="friend_count"><![CDATA[421]]></data>
    </node>
    <node id="Benjamin_Lok_2037943">
        <data key="Label">Benjamin Lok</data>
        <data key="uid"><![CDATA[2037943]]></data>
        <data key="name"><![CDATA[Benjamin Lok]]></data>
        <data key="mutual_friend_count"><![CDATA[1]]></data>
        <data key="friend_count"><![CDATA[539]]></data>
    </node>
    <node id="Camden_Elliott_Matherne_2726573">
        <data key="Label">Camden Elliott Matherne</data>
        <data key="uid"><![CDATA[2726573]]></data>
        <data key="name"><![CDATA[Camden Elliott Matherne]]></data>
        <data key="mutual_friend_count"><![CDATA[8]]></data>
        <data key="friend_count"><![CDATA[784]]></data>
    </node>
```

Initially for this problem I started with the raw XML format that was retrieved from the ‘mln.graphml’ file and then created a python program to parse the data.

```
#-*- coding: utf-8 -*-

#URIFile = open('uriFile.txt', 'r')

import xml.etree.ElementTree as ET

unsortedFile = open('friendNums.txt', 'a')
outputFile = open('friendsFB.txt', 'a')

outputCounter = 1
counter = 0
tree = ET.parse("rawFile.xml")
root = tree.getroot()

print "MLN         154"
outputFile.write("MLN         154")
outputFile.write("\n")

#for initial in root.iter('node'):
#    unsortedFile.write(root[counter][4].text)
#    unsortedFile.write("\n")
#    counter += 1

with open('friendNums.txt') as f:
    lines = f.readlines()
    lines = [x.strip('\n') for x in lines]

for Nums in lines:
    outputFile.write("F" + str(outputCounter) + "            " + Nums)
    outputFile.write("\n")
    outputCounter += 1
```

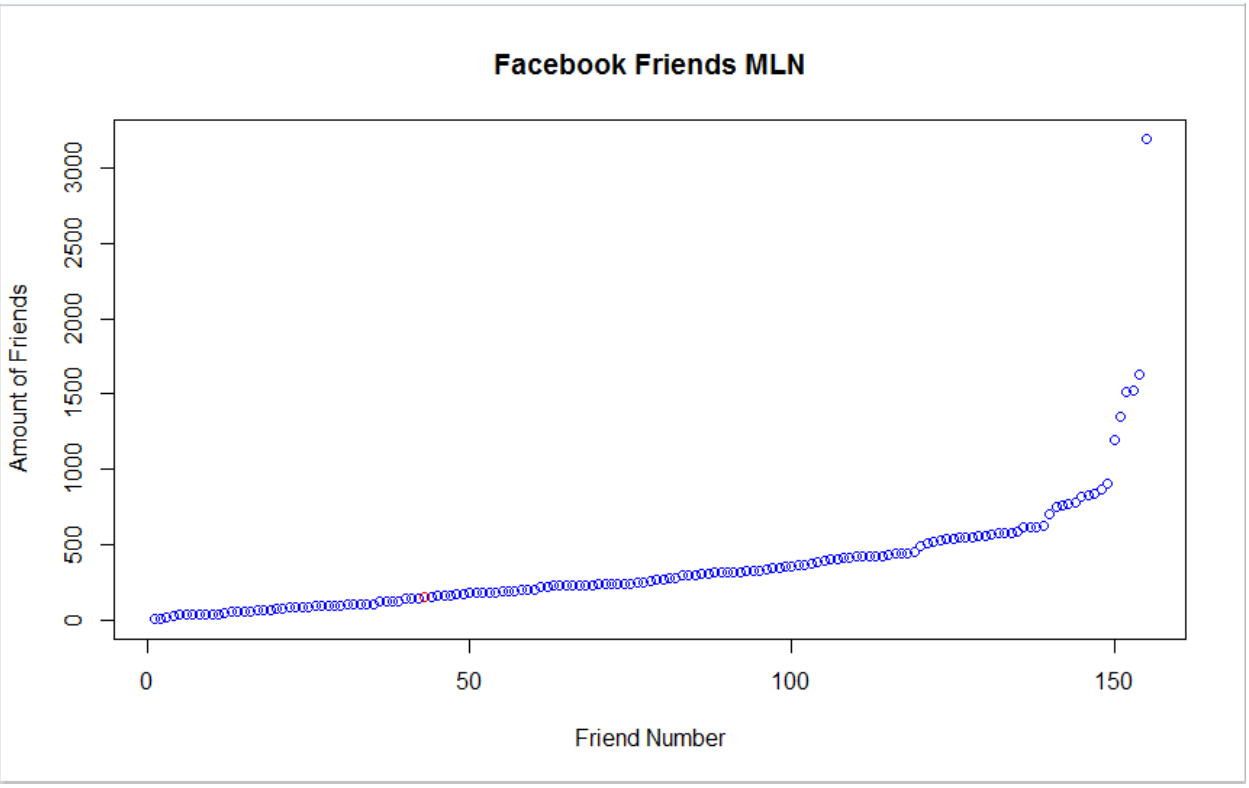
This file exclusively grabbed the amount of friends that each of the nodes in ‘rawFile.xml’ had and added it to a file called ‘friendNums.txt.’ Once I had retrieved all of the friends and the number of friends that each of them have, I sorted the list through the command-line using the sort –n command. After retrieving all of the data required, I parsed through the data and created a file with all of the friends along with adding the ‘MLN’ value to show you in the data.

```
$ cat friendNums.txt | sort -n
7
15
25
30
38
39
40
41
41
42
43
54
58
59
60
62
65
68
68
77
80
85
86
87
89
93
94
94
96
97
104
104
106
108
111
123
124
128
131
143
144
147
155
165
168
168
170
172
181
182
183
186
187
190
195
197
204
207
208
220
227
229
231
231
```

	Friend	FriendAmount
2	F1	7
3	F2	15
4	F3	25
5	F4	30
6	F5	38
7	F6	39
8	F7	40
9	F8	41
10	F9	41
11	F10	42
12	F11	43
13	F12	54
14	F13	58
15	F14	59
16	F15	60
17	F16	62
18	F17	65
19	F18	68
20	F19	68
21	F20	77
22	F21	80
23	F22	85
24	F23	86
25	F24	87
26	F25	89
27	F26	93
28	F27	94
29	F28	94
30	F29	96
31	F30	97
32	F31	104
33	F32	104
34	F33	106
35	F34	108
36	F35	111
37	F36	123
18	F17	65
19	F18	68
20	F19	68
21	F20	77
22	F21	80
23	F22	85
24	F23	86
25	F24	87
26	F25	89
27	F26	93
28	F27	94
29	F28	94
30	F29	96
31	F30	97
32	F31	104
33	F32	104
34	F33	106
35	F34	108
36	F35	111
37	F36	123
38	F37	124
39	F38	128
40	F39	131
41	F40	143
42	F41	144
43	F42	147
44	MLN	154
45	F43	155
46	F44	165
47	F45	168
48	F46	168
49	F47	170
50	F48	172
51	F49	181
52	F50	182
53	F51	183
54	F52	186
55	F53	187

Finally, after retrieving all of this data and formatting it as needed I went to RStudio and put the file into a plot generator. The graph I made shows each of the friends' amount of friends alongside your own value from least to greatest. The 'MLN' value is shown in red while all of the other values are shown in blue.

```
> plot(friendsFB$FriendAmount, col=ifelse(friendsFB$FriendAmount==154, 'red', 'blue'), xlab='Friend Number', ylab='Amount of F
riends', main='Facebook Friends MLN')
```



To calculate the standard deviation, mean, and median of the list was just as simple as putting the data into R's built in math functions. I made sure to take the 'MLN' value out of the dataset as to not skew the data.

```
> mean(friendsFB$FriendAmount)
[1] 358.987
> median(friendsFB$FriendAmount)
[1] 266.5
>
> sd(friendsFB$FriendAmount)
[1] 371.5853
```

With your total of 154 friends, the friendship paradox holds for your Facebook account due to the fact that the majority of your friends have more friends on Facebook than you do. This is shown by the fact that your friend amount is smaller than the mean, median, and mode of the dataset.

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.

```
import time
import tweepy

auth = tweepy.OAuthHandler("38wZKZuUuwGitHkE3dMpR7jEz", "czTV2ryAOTlep7FPC8dVsaAwS28Cw8Z7L8gDCLnj22ioo0uyuG")
auth.set_access_token("2352884547-xheIpcHT0oIjJmzGUkIHwt5X2IZmwogTMh9YWvc", "70RS08peFisvJyPOZGlPleQqfg98twYVkIQefeEP1Ifdg")

outputFile = open("twitterFollowerAmt.txt", 'w')
idFile = open("twitterFollowerIds.txt", 'w')

counter = 1
api = tweepy.API(auth)

ids = []

for page in tweepy.Cursor(api.followers_ids, screen_name="JuicyJake1868").pages():
    ids.extend(page)
    time.sleep(5)

print len(ids)

for name in ids:
    print name
    user = api.get_user(name)
    print user.followers_count
    outputFile.write(str(counter) + "           " + str(user.followers_count))
    outputFile.write("\n")
    print name
    idFile.write(str(name))
    idFile.write("\n")
    counter += 1

idFile.close()
outputFile.close()
```

For question two, I started in the same fashion of question whereas I created a python file to parse through each of my personal Twitter account. Using the ‘Tweepy’ library, I was able to grab a list of both the follower’s ids along with the amount of followers that each of my followers have. Only the ids were stored in the ‘twitterFollowersIds.txt’. Just the follower amount values were then sorted using the sort –n command and then the sorted follower numbers were stored in the ‘twitterFollowerAmt.txt’

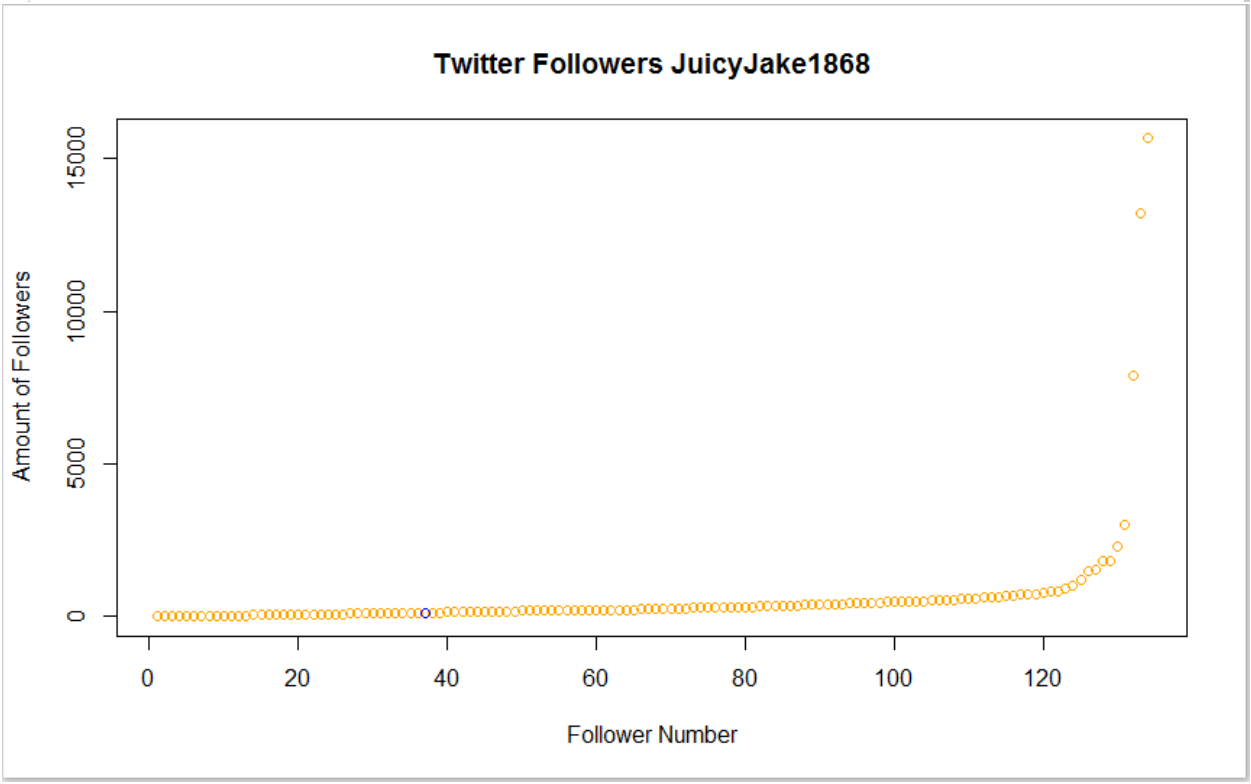
```
$ python twitterFollowerIds.py
133
460842365
376
460842365
4664999113
6
4664999113
293183998
731
293183998
3064238573
128
3064238573
1004748558
295
1004748558
4130688135
6
4130688135
4064572889
37
4064572889
941726670
210
941726670
3922723959
67
3922723959
286261482
1483
286261482
286532398
623
286532398
382174209
330
382174209
935921503
283
935921503
277619936
399
277619936
186256987
833
186256987
3468910169
100
3468910169
3382567216
134
3382567216
3103348295
3
3103348295
608698584
84
608698584
1601213750
141
1601213750
543971994
1836
```

98	472976560
99	1035149784
100	100002618
101	318487460
102	413598079
103	467954890
104	243467114
105	346964480
106	2304573203
107	2230299493
108	85005476
109	416048312
110	2345497260
111	255741404
112	849931044
113	429110769
114	320364459
115	261946574
116	232719227
117	1034930119
118	312427672
119	25005076
120	340047972
121	346705910
122	794711202
123	1122537024
124	60784841
125	1708408440
126	250966853
127	465066276
128	547267877
129	438474262
130	468602502
131	411211347
132	403723130
133	425379468

Follower	FollowerCount
1	0
2	2
3	3
4	6
5	6
6	8
7	14
8	16
9	19
10	23
11	32
12	36
13	37
14	52
15	54
16	57
17	59
18	63
19	64
20	66
21	67
22	73
23	75
24	76
25	77
26	84
27	93
28	94
29	100
30	103
31	105
32	107
33	123
34	127
35	127
36	128
37	134

Along with in the first problem, I now had the entire list of data needed to put into RStudio to generate the graph. The specific value for my amount of followers is detailed in blue while my follower's followers are detailed in orange.

```
> plot(twitterFollowerAmt$FollowerCount, col=ifelse(twitterFollowerAmt$FollowerCount==133, 'blue', 'orange'), xlab='Follower N
umber', ylab='Amount of Followers', main='Twitter Followers JuicyJake1868')
```



To compute the mean, median, and standard deviation I plugged the values into R after making sure to take the 'JWB' value out.

```
> mean(twitterFollowerAmt$FollowerCount)
[1] 637.1343
> median(twitterFollowerAmt$FollowerCount)
[1] 259.5
> sd(twitterFollowerAmt$FollowerCount)
[1] 1881.712
```

Therefore, by the rule of the friendship paradox and with my follower count being only at 133, the friendship paradox still holds for my account. This is due to my follower count being significantly less than all three of the calculated values.

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

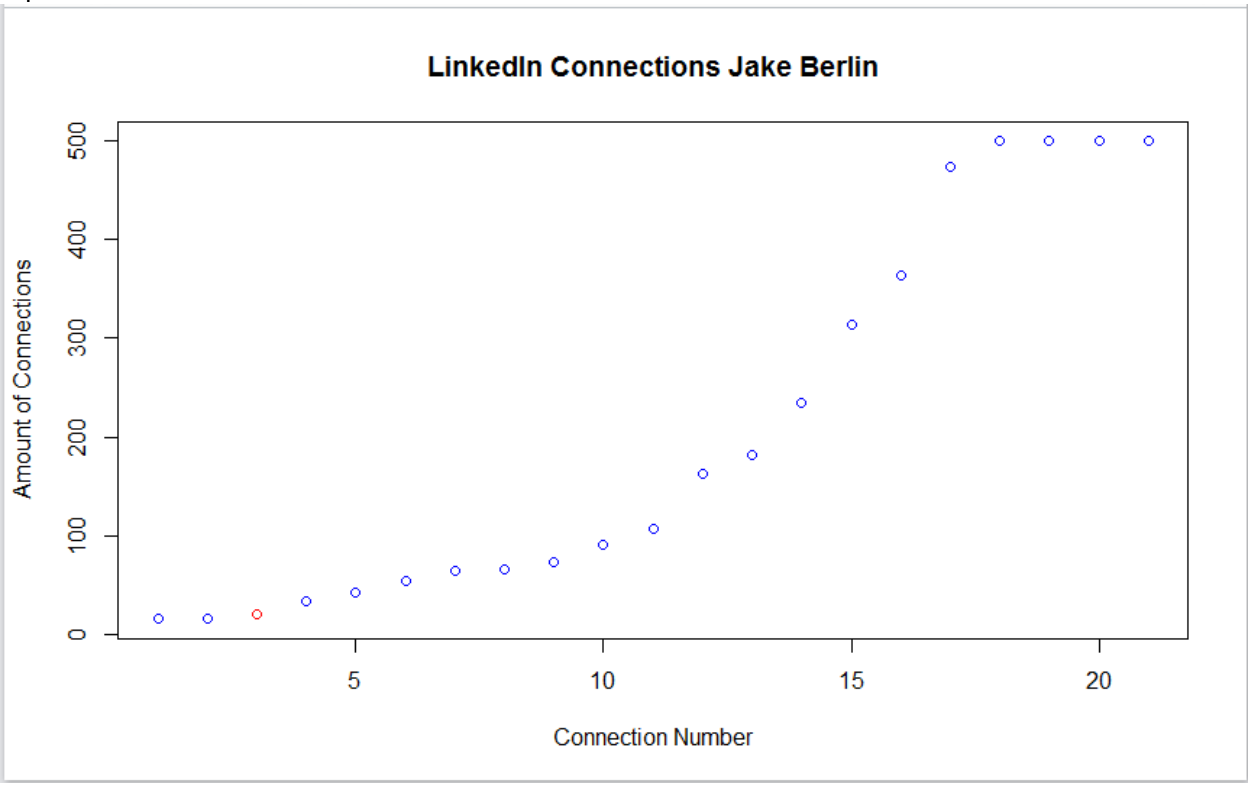
On my LinkedIn profile I only have 20 connections. (Account name is Jake Berlin)

Due to the LinkedIn API not working as of 2/20/2016, I have manually gone through each of my connections and have grabbed all of their data.

Data:

Connection	ConnectionAmt
C1	15
C2	15
JWB	20
C3	33
C4	42
C5	54
C6	64
C7	65
C8	73
C9	90
C10	106
C11	162
C12	181
C13	234
C14	314
C15	364
C16	474
C17	500
C18	500
C19	500
C20	500

```
> plot(LinkedIn$ConnectionAmt, col=ifelse(LinkedIn$ConnectionAmt==20, 'red', 'blue'), xlab='Connection Number', ylab='Amount o
f Connections', main='LinkedIn Connections Jake Berlin')
```



As in question 1, I created a plot graph based on the amount of connections I had and the connections that each of my connections had. The connection in red is the 'JWB' connection, showing myself alongside the rest of the data.

I then calculated the standard deviation, mean, and median in R after making sure to take out the 'JWB' value.

```
> sd(LinkedIn$ConnectionAmt)
[1] 190.5517
> median(LinkedIn$ConnectionAmt)
[1] 106
> mean(LinkedIn$ConnectionAmt)
[1] 205.0476
```

Finally after retrieving all of the data needed and with only 20 connections the friendship paradox holds for my LinkedIn account due to the fact that my connection amount is marginally lower than the three values I calculated.