Web Science cs532: Assignment #4

Due on Thursday, February 25, 2016

Dr.Michael.L.Nelson 4:20pm

Zetan Li

Contents

Problem 1 3

Problem 2 7

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.

= This used to be more interesting when you could more easily download your friend's friends data from Facebook. Facebook now requires each friend to approve this operation, effectively making it impossible.

I will email to the list the XML file that contains my Facebook friendship graph ca. Oct, 2013. The interesting part of the file looks like this (for 1 friend):

```
 <node id="Johan_Bollen_1448621116">
        <data key="Label">Johan Bollen</data>
        <data key="uid"><![CDATA[1448621116]]></data>
        <data key="name"><![CDATA[Johan Bollen]]></data>
        <data key="mutual_friend_count"><![CDATA[37]]></data>
        <data key="friend_count"><![CDATA[420]]></data>
</node>
```

It is in GraphML format: `http://graphml.graphdrawing.org/`

SOLUTION

First, we should able to parse the graphML. In python, we can use pygraphml to parse the file. Note that the pygraphml doesn't have any exception handlers, so any missing of keys or values will trigger the exception and interrupt the parsing.

And because of the poor documentation of pygraphml, I have to look into source file to get the idea of which function to use, in order to retrieve the nodes attribute. Then I find in node.py, there's an member function called nodes(), which can be use to get all nodes information without any searching or matching work, because the ID field will trigger the exception for some node's ID is 0.

Listing 1: Python script to parse and sort the number of friends

```
from pygraphml import Graph
from pygraphml import GraphMLParser
import numpy
import sys
reload(sys)
sys.setdefaultencoding('utf-8')
parser=GraphMLParser()
g=parser.parse("mln.graphml")

#g.show()
datafile=open("facebook.data","w")
dropfile=open("facebook_dropped.data","w")
nodes = g.nodes()
friendCounts=[]
for node in nodes:
     try:
          count=node['friend_count']
          friendCounts.append(int(count))
```

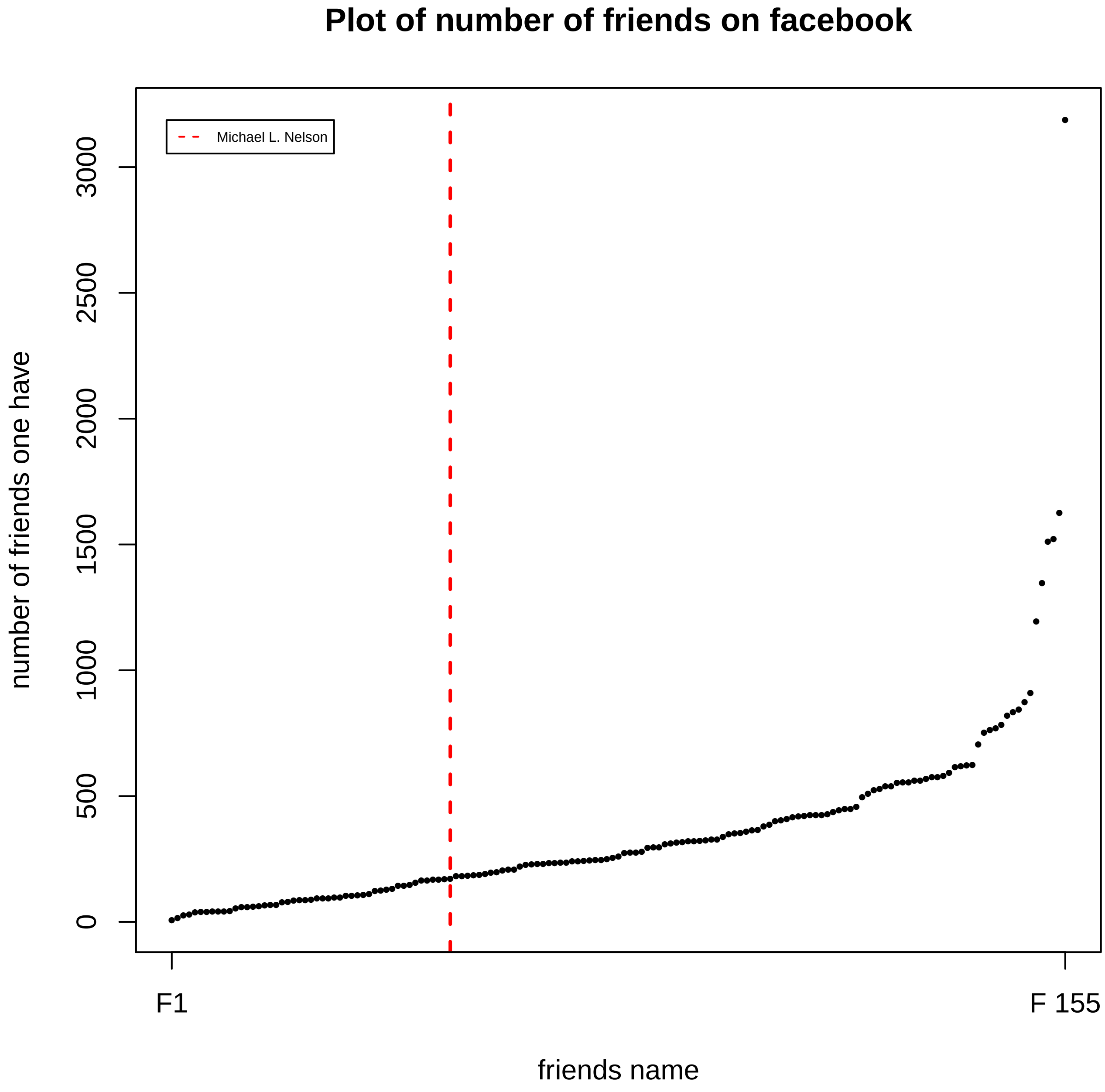
```
    except Exception, e:
        dropfile.write(str(node)+"\n")

friendCounts.append(len(nodes))
friendCounts=sorted(friendCounts)
datafile.write("Friend_count\t"+str(len(nodes))+"\n")
datafile.write("std\t"+str(numpy.std(friendCounts))+"\n")
datafile.write("mean\t"+str(numpy.mean(friendCounts))+"\n")
datafile.write("median\t"+str(numpy.median(friendCounts))+"\n")
for fcount in friendCounts:
    datafile.write(str(fcount)+"\n")
datafile.close()
dropfile.close()
```

From the result, overall statistics are:
standard deviation : 369.50
mean : 357.73
median : 259.00
As for the graph, despite the regular dot plot, we have to label the Micheal Nelson in the graph. To add this additional mark, we have to use abline in R, and conceal the regular x-axis labels. Just show the first and last label on the x-axis.[1]

Listing 2: R script to plot the facebook friends count

```
pdf("facebook_plot.pdf")
mydata=read.table("facebook.data",skip=4,header = FALSE)
num<-nrow(mydata)
#nmean<-mean(mydata$V1)
#nsd<-sd(mydata$V1)
#nmedian<-median(mydata$V1)
plot(sequence(num),mydata$V1, main = "Plot of number of friends on facebook",
    xlab = "friends name", ylab = "number of friends one have",xaxt='n',pch=16,
        cex=0.5)
abline(v=49,col="red",lwd=2,lty=2)
meanpos<-approx( x = mydata$V1,y = sequence(num), xout = nmean)
medianpos<-approx( x = mydata$V1,y = sequence(num), xout = nmedian)
sdpos<-approx( x = mydata$V1,y = sequence(num), xout = nsd)
#abline(v=meanpos$y,lwd=2,col="blue",lty=5)
#abline(v=medianpos$y,lwd=2,col="orange",lty=4)
#abline(v=sdpos$y,lwd=2,col="purple",lty=2)
axis(1, at=c(1,num),
    lab=c("F1",paste("F",num)))
legend(x=.1,y=max(mydata$V1),c("Michael L. Nelson"),
      col = c("red"),lty=c(2),
      cex=0.5)
dev.off()
```

Below is the graph of friends count.


Plot of number of friends on facebook
Michael L. Nelson
number of friends one have
0
500
1000
1500
2000
2500
3000
F1
F 155
friends name

Then, as we all know during the discussion, some nodes don't have "friends_count", those nodes have to be dropped. The dropped data is displayed below:

```
ID: 14
mutual_friend_count : 2
Label : James Florance
uid : 501351702
name : James Florance
label : James_Florance_501351702

ID: 31
mutual_friend_count : 0
Label : Joy Gooden
uid : 580143423
name : Joy Gooden
label : Joy_Gooden_580143423

ID: 52
mutual_friend_count : 8
Label : Kim Beveridge
uid : 662936475
name : Kim Beveridge
label : Kim_Beveridge_662936475

ID: 53
mutual_friend_count : 11
Label : Alfredo S nchez
uid : 667415071
name : Alfredo S nchez
label : Alfredo_Snchez_667415071

ID: 60
mutual_friend_count : 19
Label : Sarah Shreeves
uid : 700331809
name : Sarah Shreeves
label : Sarah_Shreeves_700331809

ID: 88
mutual_friend_count : 1
Label : Sally Mauck
uid : 1243862786
name : Sally Mauck
label : Sally_Mauck_1243862786

ID: 96
mutual_friend_count : 0
Label : Dan Swaney
uid : 1321960327
name : Dan Swaney
label : Dan_Swaney_1321960327

ID: 118
mutual_friend_count : 3
Label : Robert Gordeaux
uid : 1580113991
name : Robert Gordeaux
label : Robert_Gordeaux_1580113991

ID: 122
mutual_friend_count : 2
Label : Joseph Kaplan
uid : 1623901873
name : Joseph Kaplan
label : Joseph_Kaplan_1623901873

ID: 133
mutual_friend_count : 17
Label : Michael Milner
uid : 100000008814265
name : Michael Milner
label : Michael_Milner_100000008814265

ID: 134
mutual_friend_count : 3
Label : Catherine Kemble Cronin
uid : 100000016520821
name : Catherine Kemble Cronin
label :
    Catherine_Kemble_Cronin_100000016520821
```

From the graph, we can see most friends are have more friends than you (the mark of Michael, with 165 friends , is closer to the left). So the friendship paradox is true for Michael's facebook account.

Problem 1 continued on next page. . .

Problem 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 50), then use my twitter account “phonedude_mln”.

SOLUTION

As we already did the similar thing with facebook. This time we just have to find the twitter API, invoke them to get the follower count, and plot the graph by same mechanism.

Note that twitter api limits the number of requests to 15 times every 15 minutes. If we are going to fetch the follower data one by one, it most like we will exceed the rate limit and have to wait another 15 minutes for next query.

Luckily we can use follower/list api to fetch at most 200 follower information a time. So it only require 3 queries to get the information of around 500 followers of Michael Nelson. (My twitter account is brand new with only 2 I-don’t-know-who-she-is followers, so I choose to do the statistics with Michael’s account)

The statistics of the twitter follower is (in Feb 24,2016):
standard deviation : 4140
mean : 1042
median : 255

Listing 3: Python script to get the follower’s follower information

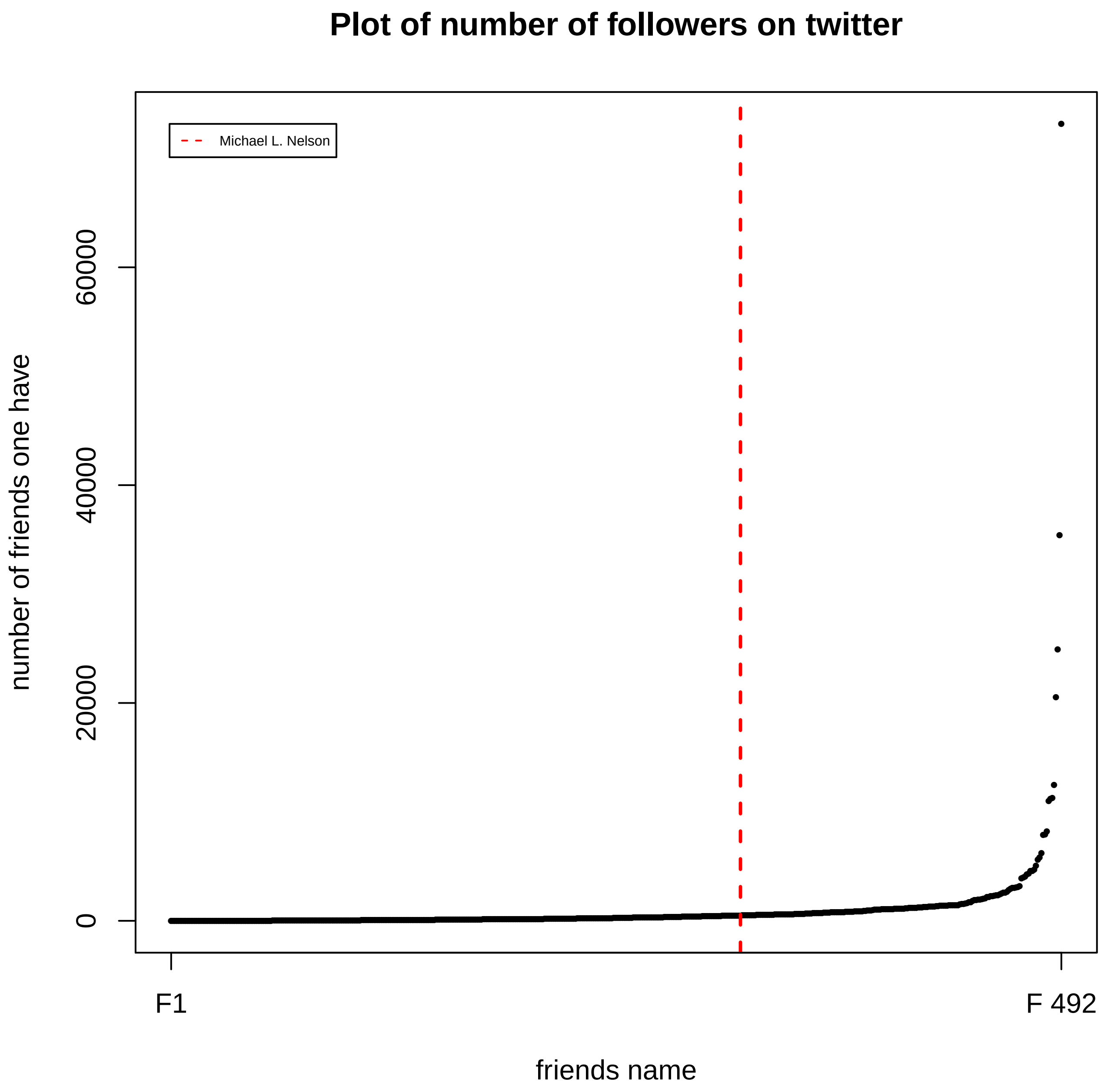
```
import tweepy
from tweepy import *
import time
import os
CONSUMER_KEY = "YwekEH9UrlXUFKUH2XmImE681"
CONSUMER_SECRET = "NWaB0jN5HaQ3f9VqCrMhP2nP8154KGXoPeDxZk6TIOVptAxErb"

OAUTH_TOKEN = "4860544225-qbjIQvrGlrj493eIHAjNu2OH0rdrHlM94XMHe1x"
OAUTH_TOKEN_SECRET = "vfAc4sJwbWExBjrMZiMqRMuknTzbl2le55DKX5gGYOOXR"

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(OAUTH_TOKEN,OAUTH_TOKEN_SECRET)
api=tweepy.API(auth)
users = []
fdata=open('twitter.data.raw','w')
user=api.get_user('phonedude_mln')
users.append({'count':user.followers_count,'name':'Michael L. Nelson'})
```

```
for user in tweepy.Cursor(api.followers, screen_name="phonedude_mln",count=200).
    items():
     #users.append(user)
     users.append(dict({'name':user.screen_name,'count':user.followers_count}))
print("followers list got.")
#print("request follower count for each follower...")
# for u in users:
#    try:
#         fuser=api.get_user(u)
#         users.append({u,fuser.followers_count})
#    except Exception, e:
#         print ('We got a timeout ... Sleeping for 15 minutes')
#         time.sleep(15*60)
#         fuser=api.get_user(u)
#         users.append({u,fuser.followers_count})

print("Finished, writing files...")
for tu in users:
     fdata.write(str(tu['count'])+"\t"+tu['name']+'\n')
os.system('sort -n -k1 twitter.data.raw > twitter.data')
fdata.close()
```

Then we are going to do the same thing as we did in problem 1 to plot the graph.

Listing 4: R script to plot the graph of twitter followers

```
pdf("twitter_plot.pdf")
mydata=read.table("twitter.data",sep="\t",header = FALSE)
num<-nrow(mydata)
#nmean<-mean(mydata$V1)
#nsd<-sd(mydata$V1)
#nmedian<-median(mydata$V1)
mlnIdx<-which(mydata$V2 == "Michael L. Nelson")
plot(sequence(num),mydata$V1, main = "Plot of number of followers on twitter",
     xlab = "friends name", ylab = "number of friends one have",xaxt='n',pch=16,
         cex=0.5)
abline(v=mlnIdx,col="red",lwd=2,lty=2)
meanpos<-approx( x = mydata$V1,y = sequence(num), xout = nmean)
medianpos<-approx( x = mydata$V1,y = sequence(num), xout = nmedian)
sdpos<-approx( x = mydata$V1,y = sequence(num), xout = nsd)
#abline(v=meanpos$y,lwd=2,col="blue",lty=5)
#abline(v=medianpos$y,lwd=2,col="orange",lty=4)
#abline(v=sdpos$y,lwd=2,col="purple",lty=2)
axis(1, at=c(1,num),
     lab=c("F1",paste("F",num)))
legend(x=.1,y=max(mydata$V1),c("Michael L. Nelson"),
       col = c("red"),lty=c(2),
       cex=0.5)
dev.off()
```



Though this time the mark is rightward a little bit, but there's still many followers who have more followers than Michael does. So the friendship paradox is true here as well.

References

[1] rputikar. *R plot x-axis label show first and last value of domain*, 2013 (accessed Febrary 24, 2016).