

Assignment 4 Report

Professor Nelson

By Tyler Medina

3/2/17

Part 1

1.1 Extracting data from GRAPHML file

To get the desired data from the GRAPHML file that was provided, I had to create a python program that could extract the data. Using the ElementTree and minidom libraries I extracted the needed elements with the `getElementsByTagName()` function.

```
1  import xml.etree.ElementTree as ET
2  from xml.dom import minidom
3  import sys
4
5  def getFriend(node):
6      Friend = {}
7      elements = node.getElementsByTagName('data')
8
9      for element in elements:
10         Friend[element.attributes['key'].value] = element.firstChild.data
11     return Friend
12
13
14 def getAttributes(file):
15     friendList = {}
16     xml = minidom.parse(file)
17     friends = xml.getElementsByTagName('node')
18
19     for friend in friends:
20         data = getFriend(friend)
21         friendList[friend.getAttribute('id')] = data
22
```

1.2 Generating R program

Once the key elements that were needed to make the graph were extracted, I created an R program to plot the data. To get the mean, median and standard deviations, I used the built-in R functions.

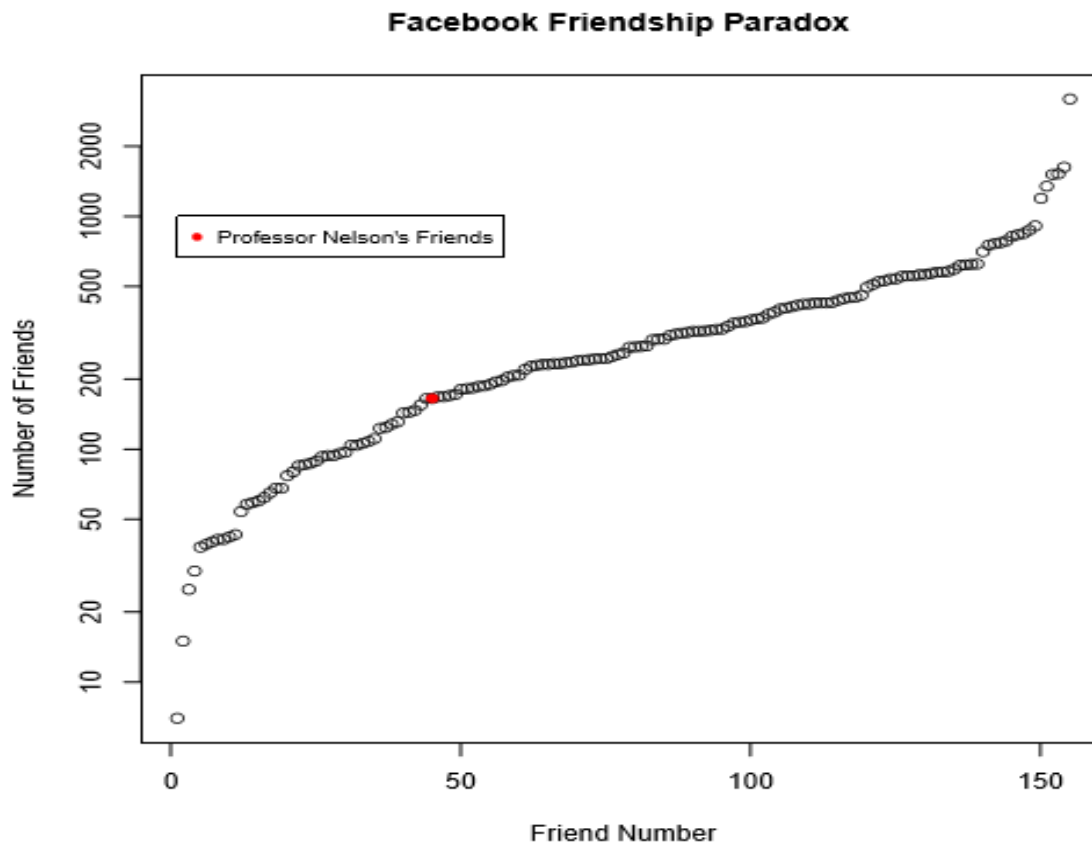
```

1 pdf("facebookGraph.pdf")
2 x_range <- c(1,155)
3 y_range <- c(7,3187)
4
5 x <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,
6 y <- c(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,
7 plot(x_range,y_range,type="n",xlab="Friend Number", ylab="Number of Friends",log="y")
8
9 points(x,y,col='black',pch=1,lwd=1)
10
11 nelsonY <- c(165)
12 nelsonX <- c(45)
13 points(nelsonX,nelsonY,type="b",lty=5,col='red',pch=16);
14 legend(1,1000,c("Professor Nelson's Friends"),pch=c(16),cex=.8,col=c("red"))
15 title("Facebook Friendship Paradox ")
16 dev.off()
17 paste("Mean:",mean(y))
18 paste("Standard Deviation: ",sd(y))
19 paste("Median: ",median(y))

```

Mean	357.735483870968
Standard Deviation	370.704490379149
Median	259

This space was intentionally left blank



1.3 Interpreting the data

Because Professor Nelson is positioned below the median, 45 and 259 respectively, it means that at least more than half of his friends contain more friends than he does. The friendship paradox holds true for Professor Nelson.

Part2

2.1 Getting number of followers

To get the number of Professor Nelson's followers, I used a program very similar to the one in assignment two. This streams the how many followers the followers of Professor Nelson have, but the data is out of order. To put the data in order, I used a small sort function.

2.2 Generating R program

Using the same code from part 1, I generated a program that graphs the follower count of Professor Nelson's followers. Just like part 1, I use the built-in R mean, standard deviation, and median functions to calculate those values.

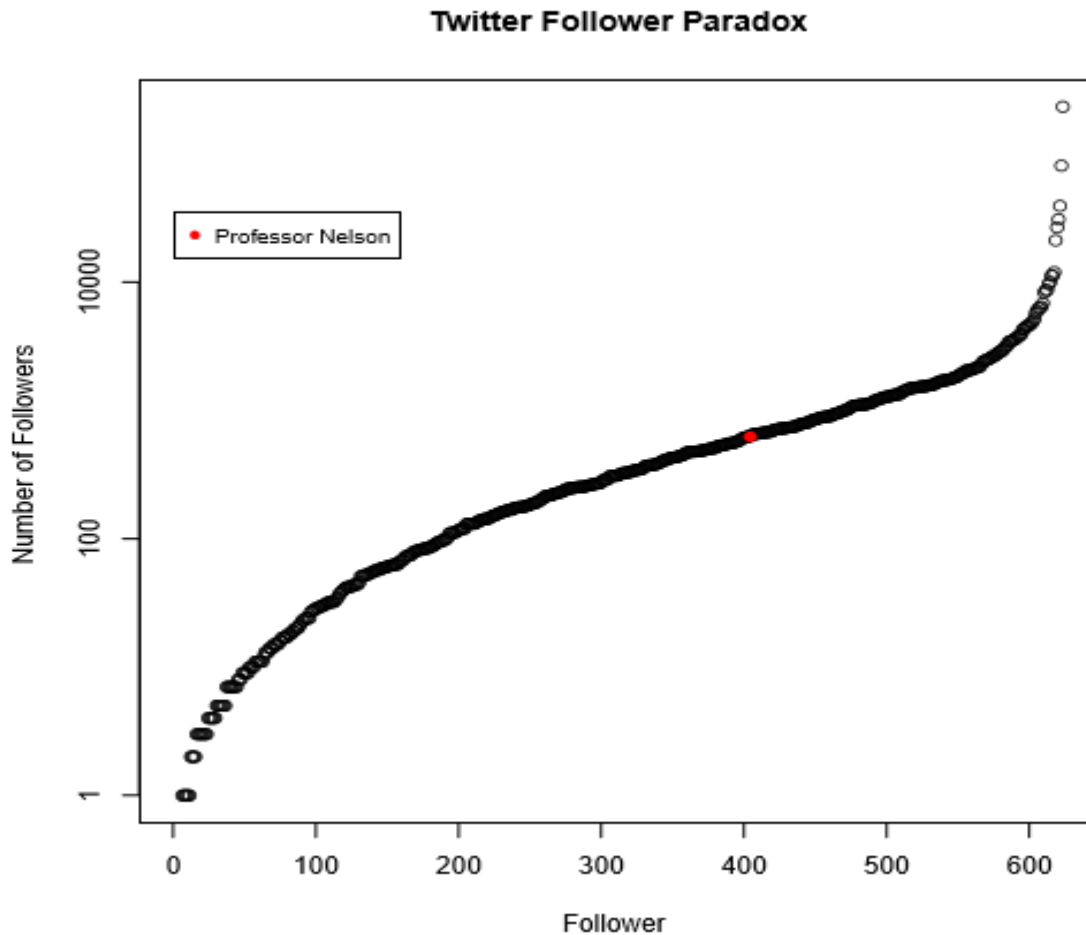
```

1 pdf("graphFollowers.pdf")
2 x_range <- c(1,622)
3 y_range <- c(1,231964)
4
5 x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, ...)
6 y <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, ...)
7 plot(x_range,y_range,type="n",xlab="Follower ", ylab="Number of Followers ",log="y")
8
9 points(x,y,col='black',pch=1,lwd=1)
10 mlnX <- c(404)
11 mlnY <- c(622)
12
13 points(mlnX,mlnY,type="b",lty=5,col='red',pch=16);
14 legend(1,35000,c("Professor Nelson"),pch=c(16),cex=.8,col=c("Red"))
15 title("Twitter Follower Paradox")
16 dev.off()
17
18 paste("Mean:",mean(y))
19 paste("Standard Deviation: ",sd(y))
20 paste("Median: ",median(y))

```

Mean	1514.8202247191
Standard Deviation	10167.031694119
Median	312

This space was intentionally left blank



2.3 Interpreting the Data

Professor Nelson is above the median, so he has more followers than most of his followers. This indicates that the friendship paradox does not hold true for his Twitter profile. There could be a couple of reasons for this. One, because you don't have to follow someone back that follows you, like in Facebook, there can be a growing disparity between popular and unpopular tweeters. Another reason, is that people generally follow a person whose content or views they like and agree with. This is a different relation from Facebook, where people have personal connections.

References

The Document Object Model API. Python Software Foundation. Version 2.0. Available at:
<https://docs.python.org/2/library/xml.dom.html>