Ryan Condotta

Final Report Assignment 4

Question 1:

In order to determine if the friendship paradox holds true for the Professor Nelson's friend, I first created the python file A4Q1.py. In the file, I opened the graphml file the Professor Nelson sent to us and observed the format of the graph before I began parsing through the file. Once I understood the file, I created a parser that searched through the file for lines containing three separate words of name, friend_count, and mutual_friend_count. The name represents the name of the user, friend_count represents the users friend count, and mutual_friend_count represents there mutual friend count. I put mutual friend count in there because friend count is in mutual friend count. So, if friend count was found in the file, I then determined if it was mutual friend count or friend count to get the right count for the user's friend. This process is shown below:

```
for line in text:
        if word in line:
                count += 1
                #print(line[27:len(line)-11])
                if count > 18:
                        names.append(line[27:len(line)-11])
                elif count < 18:
                        pass
        if word2 in line:
                if temp < 1:
                        pass
                        temp +=1
                else:
                        if word3 in line:
                                pass
                        else:
                                #print(line[35:len(line)-11])
                                friends.append(int(line[35:len(line)-11]))
names.append("Michael Nelson")
friends.append(int(len(friends)))
```

Once all of the names and there corresponding friend counts are stored in a couple of lists. I then outputted all of the information into an excel file named Table1.xlsx. It outputs the information into two categories of names and num of friends. The process of outputting the information in python is shown below:

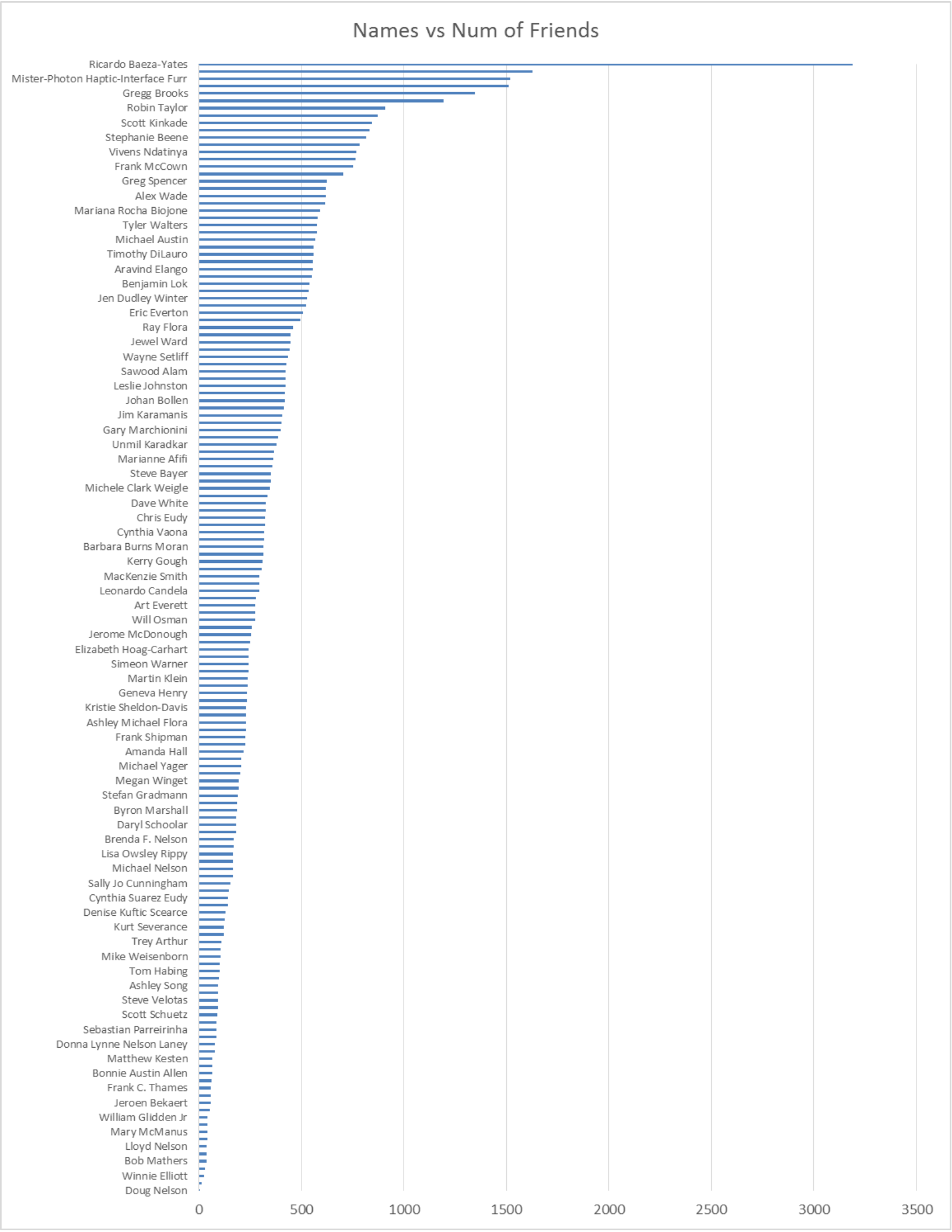
```
filename2 = r'C:\Users\Ryan\Documents\WebScience\Assignment4\Table1.xlsx'
workbook = xlsxwriter.Workbook(filename2)
worksheet = workbook.add_worksheet()

bold = workbook.add_format({'bold' : 1})
```

```
#data headers
worksheet.write('A1', 'Name', bold)
worksheet.write('B1', 'Num of Friends', bold)

row = 1
col = 0

#throw data into
for name in names:
        worksheet.write_string(row, col, name)
        row += 1

row1 = 1
col1 = 1
for Number in friends:
        worksheet.write_number(row1, col1, Number)
        row1 += 1

workbook.close()
```

Next, I sorted the data for the names and friends from lowest to highest according to the number of friends as shown in the table. Then, I calculated the Mean, Standard Deviation, and Median of the data. This helped me to determine that the friendship paradox holds true for Professor Nelson because the average number of friends and the median value is greater than Professor Nelson friend count of 165. I also did a count to see how many of his friends were greater than him and less and the results were 110 were greater and 54 were less than his friend count. I did that because sometimes, the average and median values could lead to false assumptions.

Name	Num of Friends	Mean	Standard Deviation	Median
James Florance	0	334.0301	369.089	243
Joy Gooden	0			
Kim Beveridge	0	**Greater than Mr Nelson**	**Less than Mr Nelson**	
Alfredo Sanchez	0	110	54	

Lastly, I created the graph with friend names on the y-axis and the number of friends on the x-axis. This graph is shown on the next page with Professor Nelson appearing near the bottom of the graph for the y-axis.


Names vs Num of Friends
Ricardo Baeza-Yates
Mister-Photon Haptic-Interface Furr
Gregg Brooks
Robin Taylor
Scott Kinkade
Stephanie Beene
Vivens Ndatinya
Frank McCown
Greg Spencer
Alex Wade
Mariana Rocha Biojone
Tyler Walters
Michael Austin
Timothy DiLauro
Aravind Elango
Benjamin Lok
Jen Dudley Winter
Eric Everton
Ray Flora
Jewel Ward
Wayne Setliff
Sawood Alam
Leslie Johnston
Johan Bollen
Jim Karamanis
Gary Marchionini
Unmil Karadkar
Marianne Afifi
Steve Bayer
Michele Clark Weigle
Dave White
Chris Eudy
Cynthia Vaona
Barbara Burns Moran
Kerry Gough
MacKenzie Smith
Leonardo Candela
Art Everett
Will Osman
Jerome McDonough
Elizabeth Hoag-Carhart
Simeon Warner
Martin Klein
Geneva Henry
Kristie Sheldon-Davis
Ashley Michael Flora
Frank Shipman
Amanda Hall
Michael Yager
Megan Winget
Stefan Gradmann
Byron Marshall
Daryl Schoolar
Brenda F. Nelson
Lisa Owsley Rippy
Michael Nelson
Sally Jo Cunningham
Cynthia Suarez Eudy
Denise Kuftic Scearce
Kurt Severance
Trey Arthur
Mike Weisenborn
Tom Habing
Ashley Song
Steve Velotas
Scott Schuetz
Sebastian Parreirinha
Donna Lynne Nelson Laney
Matthew Kesten
Bonnie Austin Allen
Frank C. Thames
Jeroen Bekaert
William Glidden Jr
Mary McManus
Lloyd Nelson
Bob Mathers
Winnie Elliott
Doug Nelson
0
500
1000
1500
2000
2500
3000
3500

Question 2:

In order to solve this problem of determining if the friendship paradox holds true for me from my followers on twitter, I first created the file of A4Q2.py. Then, in the file I had to figure out a way of accessing my followers friends count which I did through the tweepy package in python. This package allowed me to use a cursor to access the user name and the users friend count and store them into lists of names and followers but before I could do that I had to get all of the access tokens and secret tokens in order to access my twitter account. The process of accessing twitter and storing variables is shown below:

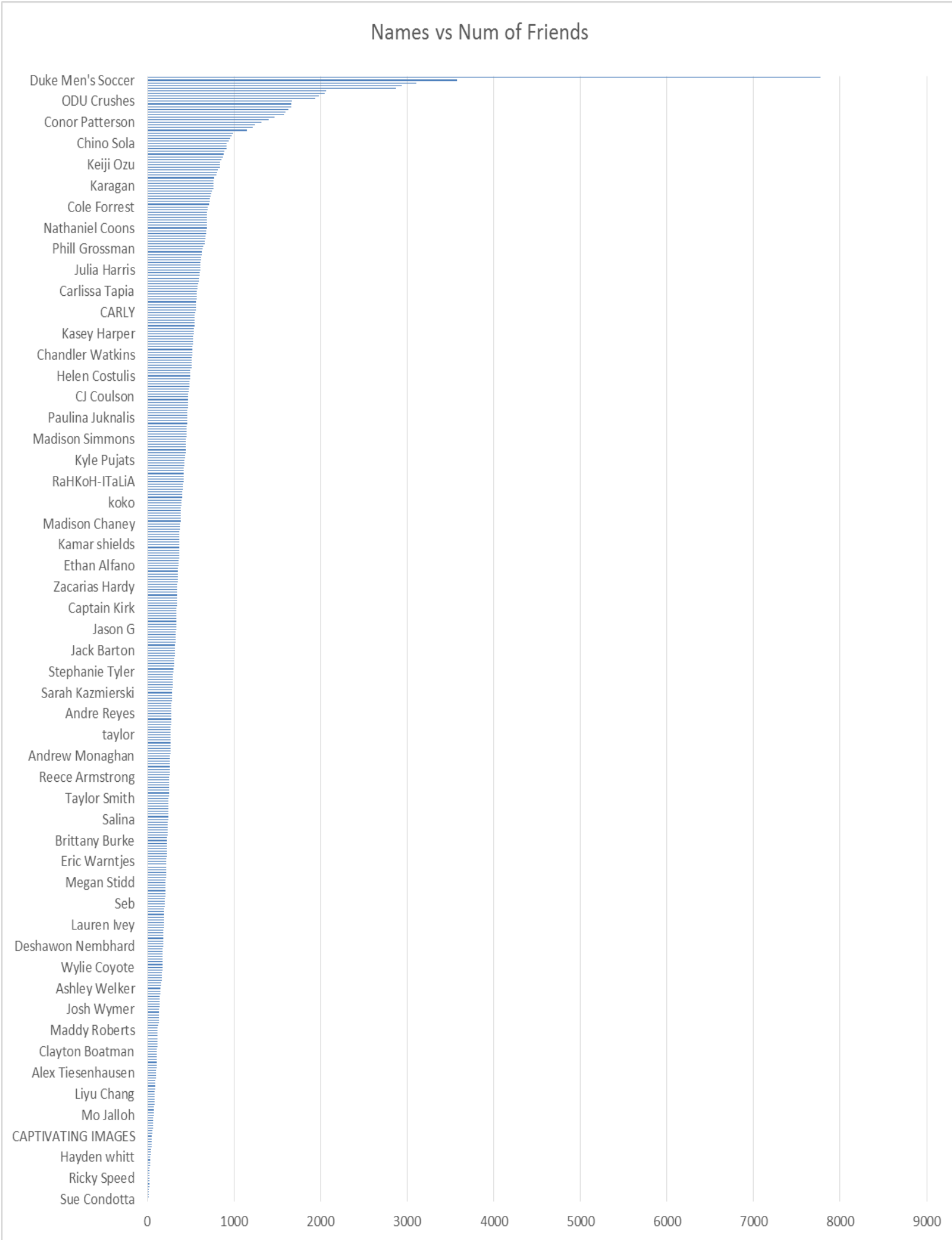
```
auth = tweepy.OAuthHandler(ckey, csecret)
auth.set_access_token(atoken, asecret)

api = tweepy.API(auth)

names = []
followers = []
#for user in tweepy.Cursor(api.followers, screen_name="twitter").items():
count = 0
for user in tweepy.Cursor(api.friends, count = 200).items():
        try:
                #print(count)
                print(user.name, user.followers_count)
                names.append(str(user.name))
                followers.append(int(user.followers_count))
                process_status(user)
                count+=1
        except:
                pass
                count+=1
print(len(names))
print(len(followers))
```

The next part for the python code was to again output it to an excel file where I could easily sort the data and calculate mean, standard deviation, and median values. The process for outputting to excel is the same as in the question 1 but with different list names of followers and names as well as a new excel file of Table2.xlsx.

In the Table2.xlsx file, the name, num of friends, Mean, Standard Deviation, Median, More Followers than me, and less followers than me. From the data, based on the Mean value solely, I would come to the conclusion that the friendship paradox holds true. However, after doing the count for more followers than me and less followers than me, it shows that the friendship paradox does not hold true for my circumstance. The reason I got such weird values for Mean and Standard Deviation is because I follow celebrities and new reports who have a huge amount of followers in the 100 of thousands in value which throws the rest of my values out of whack.

Name	Num of Friends	Mean	Standard Deviation	Median
Sue Condotta	3	492757.1	2693292.122	367
patty Haberkorn	16			
skylar	18			
Colin Springmeier	18		More Followers than me	Less Followers than me
Joel Mendoza	19		281	212

The last part of the assignment was to create a graph with the number of friends on the x-axis and the name of friends on the y-axis which is shown on the next page. I had to construe the values and limit the range of the graph to only include the important information that was shown and not show the extremely high values that would misconstrue the graph and make it unreasonable. The result is shown on the next page.


Names vs Num of Friends
Duke Men's Soccer
ODU Crushes
Conor Patterson
Chino Sola
Keiji Ozu
Karagan
Cole Forrest
Nathaniel Coons
Phill Grossman
Julia Harris
Carlissa Tapia
CARLY
Kasey Harper
Chandler Watkins
Helen Costulis
CJ Coulson
Paulina Juknalis
Madison Simmons
Kyle Pujats
RaHKoH-ITaLiA
koko
Madison Chaney
Kamar shields
Ethan Alfano
Zacarias Hardy
Captain Kirk
Jason G
Jack Barton
Stephanie Tyler
Sarah Kazmierski
Andre Reyes
taylor
Andrew Monaghan
Reece Armstrong
Taylor Smith
Salina
Brittany Burke
Eric Warntjes
Megan Stidd
Seb
Lauren Ivey
Deshawon Nembhard
Wylie Coyote
Ashley Welker
Josh Wymer
Maddy Roberts
Clayton Boatman
Alex Tiesenhausen
Liyu Chang
Mo Jalloh
CAPTIVATING IMAGES
Hayden whitt
Ricky Speed
Sue Condotta
0
1000
2000
3000
4000
5000
6000
7000
8000
9000