HW4 - Social Media

Adeniran Adeniyi

March 17th 11:59 PM

Q1

Compute the mean, standard deviation, and median of the number of friends that the user's friends have.
Create a graph of the number of friends (y-axis) and the friends (x-axis) themselves, sorted by number of friends (y-axis). (The friends don't need to be labeled on the x-axis: just f1, f2, f3, ... fn.) Include the user in the graph (count the number of their friends) and label as U.

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar  9 00:08:55 2021

@author: adeni
"""
import pandas as pd
import matplotlib.pyplot as plt


filePath = "HW4-friend-count.csv"
df = pd.read_csv(filePath,index_col=False, encoding='utf-8')
#remove extra spaces from columns (bad columns)
df.columns =[col.strip() for col in df.columns]



df = df.sort_values(by="FRIENDCOUNT")
f = "f"
#create a list of new users
newUser= []
for x  in range(98):
    c = f + str(x)
    newUser.append(c)
# replace this new list to the column user in the dataframe
df.USER = newUser
print(df)
#print(df.keys)
print(df.FRIENDCOUNT.mean())
```

```
30 print(df.FRIENDCOUNT.std())
31 print(df.FRIENDCOUNT.median())
32 #using
33 plt.rcParams['figure.figsize'] =40,25
34 plt.rcParams['font.size'] = 17;
35 plt.plot(df.USER,df.FRIENDCOUNT, label="FriendCount",marker='o')
36 plt.grid(True)
37 plt.xticks(rotation=90)
38 plt.xlabel(xlabel = "friends", fontsize=20)
39 plt.ylabel(ylabel= "number of friends", fontsize=20)
40 plt.legend(loc=6,fontsize=25);
```

Listing 1: friendPradox.py

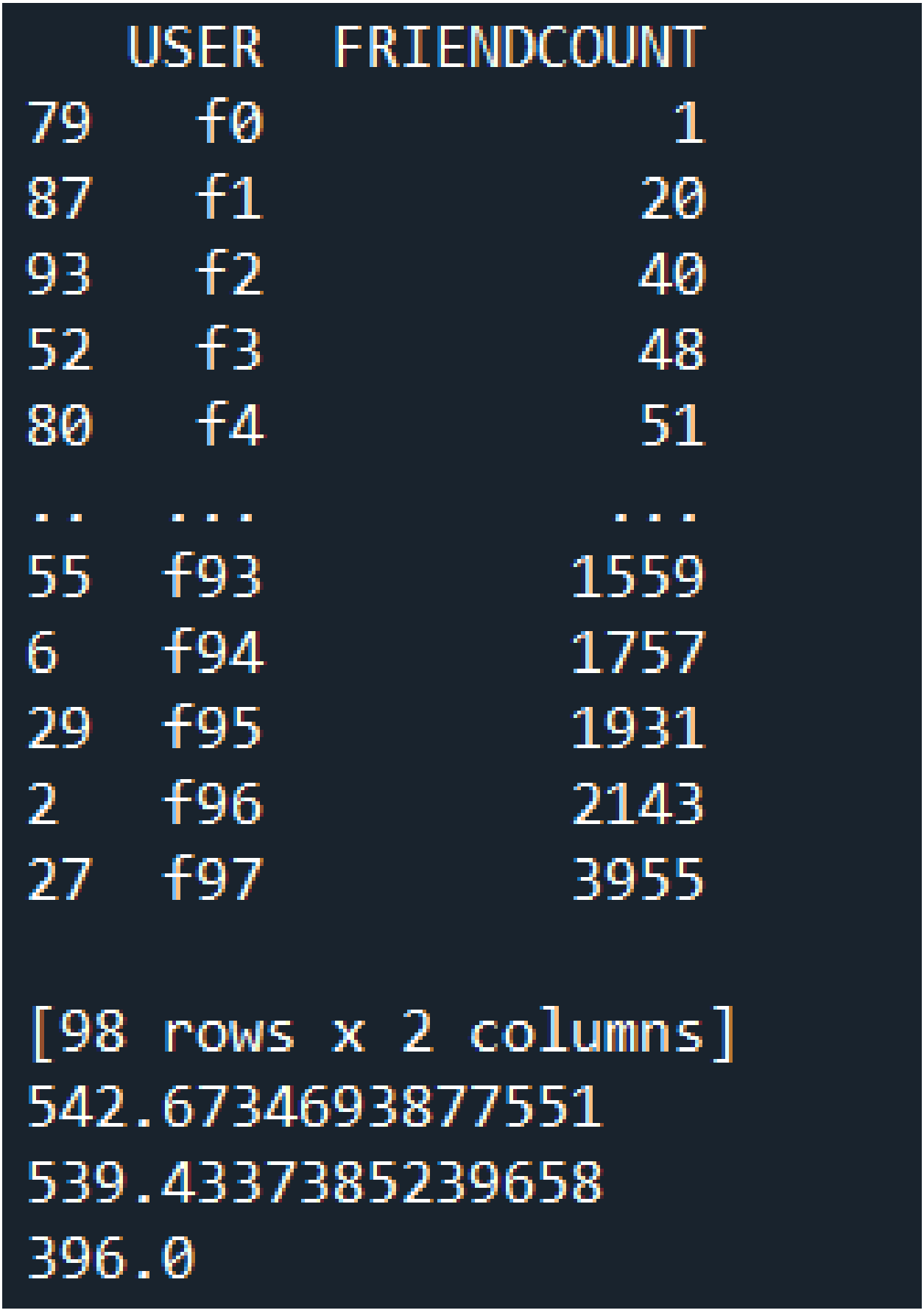

Figure 1: Shows the pandas data and mean,std and meadian respectsively

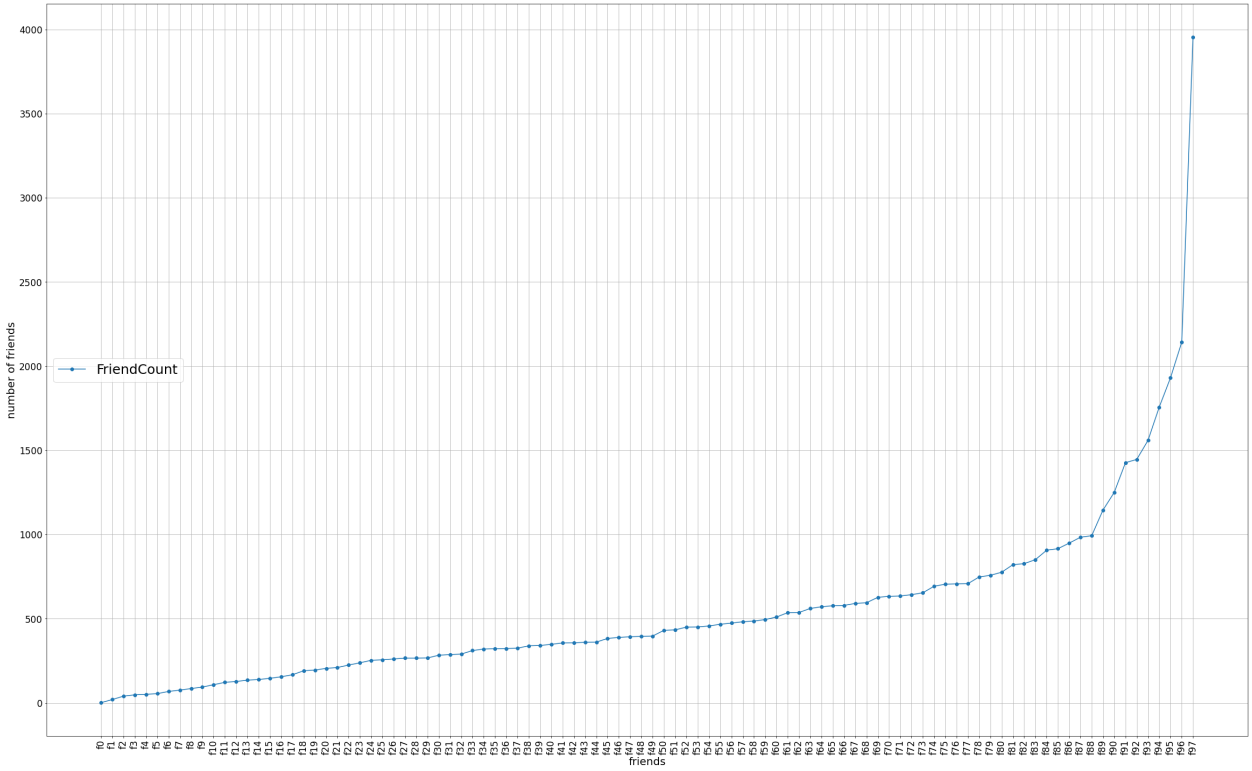

Figure 2: The plot for Q1

Discussion

I followed these instruction below:

- I read the csv file using the pandas library
  - Removed the extra space in the column head on line 14 of friendPradox.py
- sorted the values of the FriendCount column
- constructed a list of f0, f1,f2 ,..., fn - iitems equal to the number of size of the data in pandas dataframe on line 19 - 24
- Replaced the User column in dataframe with the list values. on line 26
- Using the data frame built in function I was able to calculate:
  Mean to be: 542.67
  Standard deviation:539.43
  Median to be:396.0
- finally plotted the graph in Figure 2

Q2

Determine if the friendship paradox holds for your Twitter account. Since Twitter is a directed graph, use followers as the value you measure (i.e., "do your followers have more followers than you?"). Due to Twitter rate limits, this part will take some time to complete. Generate the same graph as in Q1, and calculate sthe same mean, standard deviation, and median values.

Answer

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar 10 23:31:36 2021

@author: adeni
"""

import  tweepy
import pandas as pd
import matplotlib.pyplot as plt

def auth():#authorization fo consumer key and consumer secret
    consumer_key ="pnUItdX31QmYpHBFlVcYbocKQ"
    consumer_secret ="
   gFNX2iztwhfL1tROFCX3UomwRbU8GjUJhHzLQat8DGxvBcyVmw"
    access_token ="3311358952-Gb5GEkFGvsv2IUFPuCEzChdkJCssh9B2mW9VsFG"
    access_token_secret ="t1UGfFdJR8qUmcxnTEMKkg3899iU63qd2zCwaNYgxPvcV
   "
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    api = tweepy.API(auth, wait_on_rate_limit= True)
    return api
def plot_the_pandas_data(df,fwfing):
    #using
    if fwfing == 0:
        plt.title("Followers")
    else:
        plt.title("Followings")
    plt.rcParams['figure.figsize'] =40,25
    plt.rcParams['font.size'] = 17;
    plt.plot(df.USER,df.FRIENDCOUNT, label="FriendCount",marker='o')
    plt.grid(True)
    plt.xticks(rotation=90)
    plt.xlabel(xlabel = "friends", fontsize=20)
    plt.ylabel(ylabel= "number of friends", fontsize=20)
    plt.legend(loc=6,fontsize=25);

```

```
35      return plt.show()
36  def handlePandasDF_Replacement(df):
37      df = df.sort_values(by="FRIENDCOUNT")
38      #print(df)
39      f = "f"
40      #create a list of new users
41      newUser= []
42      try:
43          for x  in range(len(df)):
44              c = f + str(x)
45              newUser.append(c)
46          # replace this new list to the column user in the dataframe
47          df.USER = newUser
48      except Exception as p:
49          print(p)
50      print(df)
51      return df
52  def get_number_of_followers_followings(id,ingers):
53      #get user
54
55      user = auth().get_user(id)
56      try:
57          if ingers == 0:
58              #used to get the user ids followers count
59              fwers_flowing = user.followers_count
60          else:
61              #used to get the users following count
62              fwers_flowing = user.friends_count
63          #print("{}: {}\n".format(user.screen_name,fwers_flowing))
64      except Exception as p:
65          print(p)
66      return user.screen_name,fwers_flowing
67  def getUserFollowers(screen_name):
68     users = []
69     followersC = []
70      #followers
71     p= tweepy.Cursor(auth().followers_ids,id=screen_name,
       wait_on_rate_limit= True).items(5000)
72     c =1
73     for ps in p:
74         #parse in 0 to get the followers count
75         us, count = get_number_of_followers_followings(ps,0)
76         users.append(us)
77         followersC.append(count)
78         c=c +1
79         if(c > 98):
80             break;
```

```
    #inset into pandas
    prod = pd.DataFrame(list(zip(users,followersC)))
    #users.clear()
    #followersC.clear()
    #create a column names
    prod.columns = ["USER", "FRIENDCOUNT"]
    #sort and replace the names symbol fn
    prod = handlePandasDF_Replacement(prod)
    return prod
def getUserFollowings(screen_name):
    users = []
    followingsC = []
    #followings
    p= tweepy.Cursor(auth().friends,id=screen_name, wait_on_rate_limit=
    True).items(5000)
    c =1
    for ps in p:
        #parse in 1 to get the following count instead
        us, count = get_number_of_followers_followings(ps.id,1)
        users.append(us)
        followingsC.append(count)
        c=c +1
        if(c > 98):
            break;
    #inset into pandas
    prod = pd.DataFrame(list(zip(users,followingsC)))
    users.clear()
    followingsC.clear()
    #create a column names
    prod.columns = ["USER", "FRIENDCOUNT"]

    #Sort and replace the names symbol fn
    prod = handlePandasDF_Replacement(prod)
    return prod
screen_name = "adeniran827"
try:
    """
    ===============Q2================
    """
    print("\n\n Get into the follower Count \n\n")
    f = getUserFollowers(screen_name)
    #data has been already sorted
    #get the men, std and meandia
    print("Mean : {}".format(f.FRIENDCOUNT.mean()))
    print("Standard deviation: {}".format(f.FRIENDCOUNT.std()))
    print("Median: {}".format(f.FRIENDCOUNT.median()))
    #print as a csv file
```

```
    f.to_csv("Q2/finalFollowers.csv",index=False)
    #plot the graph
    plot_the_pandas_data(f,0)
    #replace the names with fn

    #returns the dataframe of two columns USER and FRIENDCOUNT
    #comes with user screen name and followers' count
    """
    ===============Q3================
    """
    print("\n\n Get into the followings Count \n\n")
    p = getUserFollowings(screen_name)
    #data has been already sorted
    #get the men, std and meandia
    print("Mean: {}".format(p.FRIENDCOUNT.mean()))
    print("Standard deviation: {}".format(p.FRIENDCOUNT.std()))
    print("Median: {}".format(p.FRIENDCOUNT.median()))
    #print as a csv file
    p.to_csv("Q3/finalFollowings.csv",index=False)
    #plot the graph
    plot_the_pandas_data(p,1)
    #REPEAT same code as friendPradoxpy -- I know I can make it a whole
    py file
except Exception as r:
     print(r)
```

Listing 2: followerExtactor.py

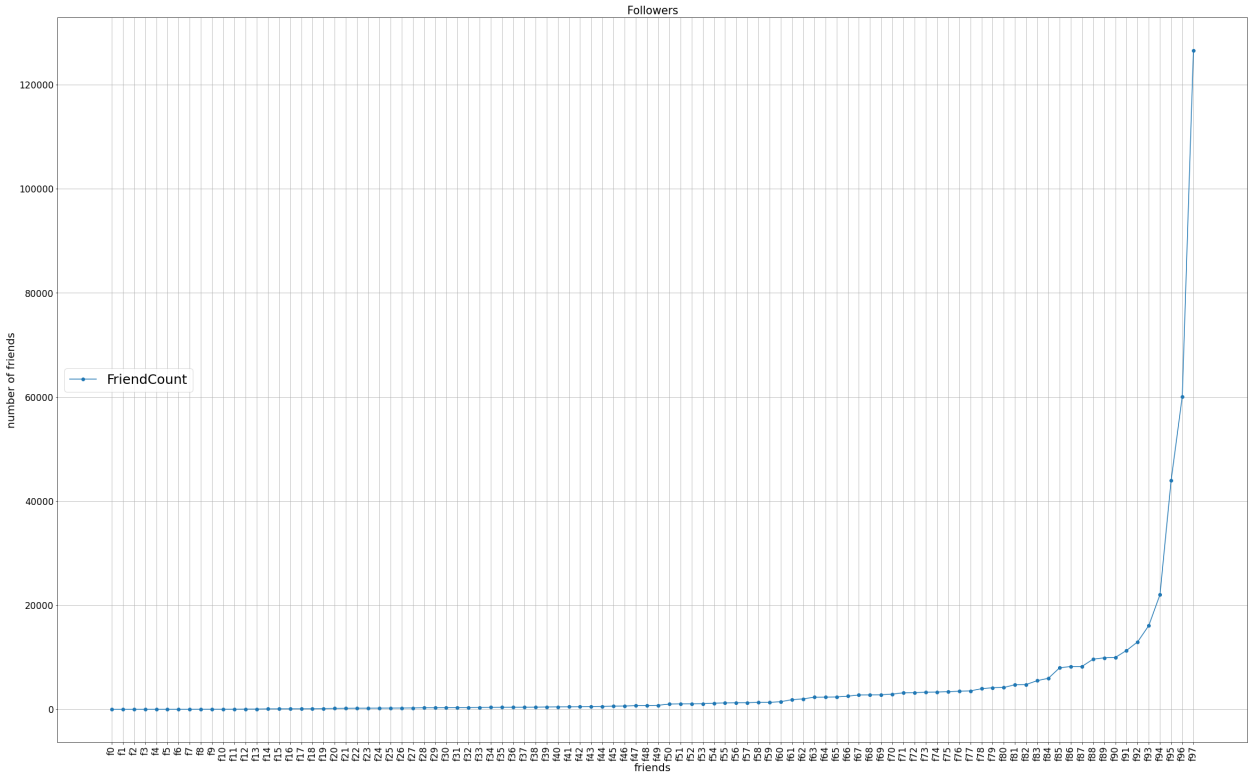

Figure 3: The plot for Q2 followers

```
 Get into the follower Count


    USER  FRIENDCOUNT
19    f0           4
15    f1           8
33    f2          12
20    f3          15
71    f4          17
..   ...         ...
96   f93       16121
88   f94       22043
50   f95       43974
76   f96       60110
30   f97      126568

[98 rows x 2 columns]
Mean : 4657.571428571428
Standard deviation: 14819.39372272963
Median: 772.5
```

Figure 4: shows the console output for Q2

Discussion

I followed these instruction below:

- I created a function called getUserFollowers in line 67 of followerExtractor.py

```
67 def getUserFollowers(screen_name):
```

Listing 3: The function in followerExtractor.py

  - Using my twiter account screen name(adeniran827),

```
114 screen_name = "adeniran827"
```

Listing 4: Twitter name used in followerExtractor.py

I retrived all of my followers ID (Limited this list to 98 on line 70 -71)

```
70     #followers
71    p= tweepy.Cursor(auth().followers_ids,id=screen_name,
      wait_on_rate_limit= True).items(5000)
```

Listing 5: Using tweepy to store the list of my followers saved ids in p followerExtractor.py

```
79         if(c > 98):
80             break;
```

Listing 6: Limiting the list gotten from p in followerExtractor.py

  - I Passed each user Id i got to the get_number_of_followers_followings function. This function returned followers count of the associated user Id
    - In line 57 to 59, ensured that followers count was possible, since 0 was parsed into the function along with the id of the user

```
52 def get_number_of_followers_followings(id,ingers):
53     #get user
54
55     user = auth().get_user(id)
56     try:
57         if ingers == 0:
58             #used to get the user ids followers count
59             fwers_flowing = user.followers_count
```

Listing 7: Using follower_count to get the total followers a user have in followerExtractor.py

  - Parsed the generated list of users and followers count as a pandas dataframe to handledPandasDF_Replacement function
  - Sort the data frame in handledPandasDF_Replacement
  - Performed silimar operation of Q1, replacing the names with f0, f1,f2 ,..., fn - names
  - Got the dataframe at line 120 of followerExtractor.py

- Got values of:
  mean: 4657 followers for my followers followers count
  std:14819.39
  median:772.5
- finally plotted the graph in Figure 5
- Comparing the mean of my followers followers' count 4657 to my followers count of 234. Those that I follow have a higher followers count than I do.
- Comparing the mean to my current twitter followers of 234, the first 98 of those that follow me have a higher following than I do.

Q3

Answer

```
134    """
135    ===============Q3================
136    """
137    print("\n\n Get into the followings Count \n\n")
138    p = getUserFollowings(screen_name)
139    #data has been already sorted
140    #get the men, std and meandia
141    print("Mean: {}".format(p.FRIENDCOUNT.mean()))
142    print("Standard deviation: {}".format(p.FRIENDCOUNT.std()))
143    print("Median: {}".format(p.FRIENDCOUNT.median()))
144    #print as a csv file
145    p.to_csv("Q3/finalFollowings.csv",index=False)
146    #plot the graph
147    plot_the_pandas_data(p,1)
```

Listing 8: snapshot that shows the followings process in followerExtractor.py

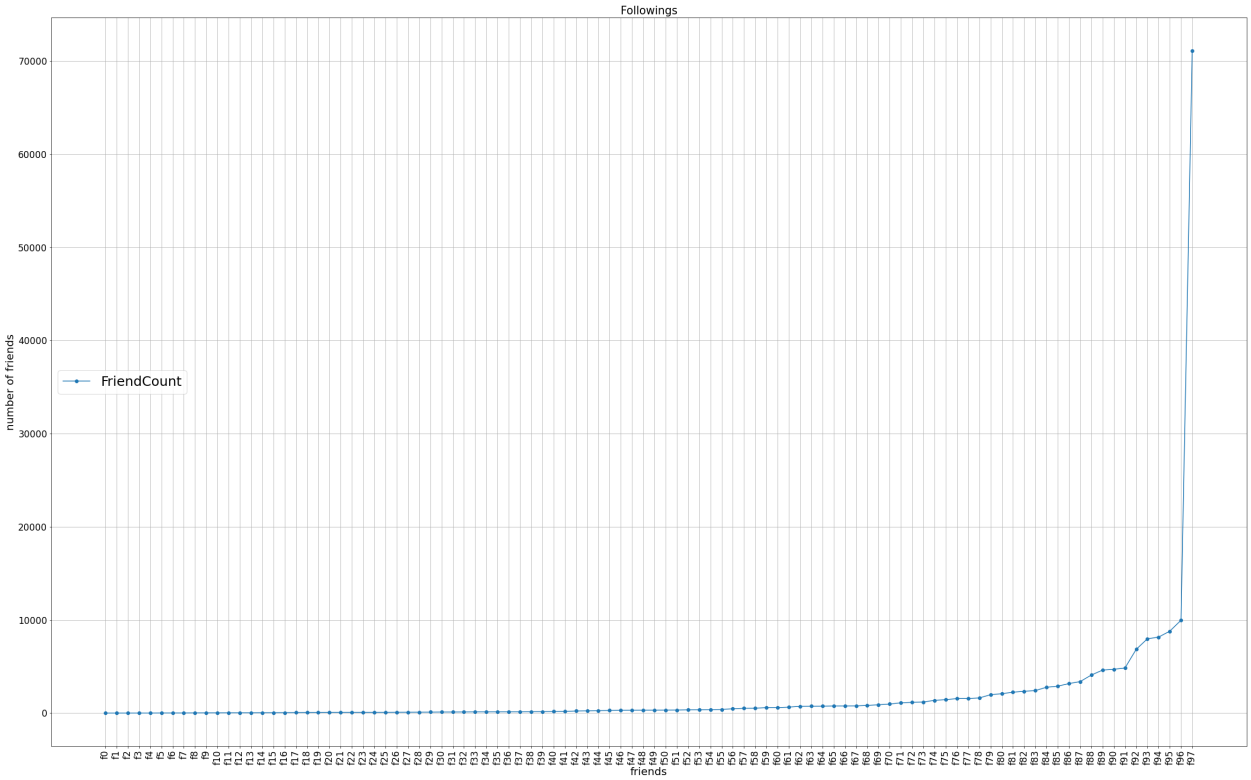

Figure 5: The plot for Q3 followings

```
 Get into the followings Count


    USER  FRIENDCOUNT
69    f0           0
31    f1           0
84    f2           0
16    f3           1
53    f4           1
..   ...         ...
25   f93        7972
52   f94        8154
61   f95        8792
78   f96        9965
49   f97       71102

[98 rows x 2 columns]
Mean: 1867.5510204081634
Standard deviation: 7347.861919363338
Median: 311.0
```

Figure 6: shows the console output for Q3

Discussion

For the major part Q3 followed samples for Q2, the major differences will only be listed.

I followed these instruction below:

- I used the getUserFollowings function in line 138.

```
138    p = getUserFollowings(screen_name)
```

Listing 9: snapshot the getUserFollowings function used in followerExtractor.py

- This function used auth().friends on 98 instead of auth().followers_ids, since it was retriving my following list

```
94    p= tweepy.Cursor(auth().friends,id=screen_name,
      wait_on_rate_limit= True).items(5000)
```

Listing 10: snapshot the getUserFollowings function used in followerExtractor.py

- The last difference is the friends following count instead of their followers_count, in line 60 to 62

```
60          else:
61              #used to get the users following count
62              fwers_flowing = user.friends_count
```

Listing 11: snapshot the getUserFollowings function used in followerExtractor.py

- Got values of:
  mean: 1867.55 followings
  std:7347.86
  median:311.0

- Comparing the mean of followings 1867.55 to my followings of 454 those that follow me have a higher followings than I do.

References

- https://stackabuse.com/rotate-axis-labels-in-matplotlib/
- https://www.youtube.com/watch?v=AYorFcI1MTU&t=212s
- http://jonathansoma.com/lede/foundations/classes/pandas%20columns%20and%20functions/fixing-column-names-in-pandas/

- https://www.geeksforgeeks.org/python-tweepy-getting-the-number-of-followers-of-a-user/
- https://stackoverflow.com/questions/15943769/how-do-i-get-the-row-count-of-a-pandas-dataframe
- https://www.geeksforgeeks.org/python-user-object-in-tweepy/