

# Assignment 4

Mohammed Shaaban

2016-02-25

## 1 Problem 1

In the first question I downloaded the XML file from the website using python that contains four defs(functions). First function contains the piece of code that gets the line from the XML file having the word “friends count” but as a whole string. Then I create another function to get the number that is specified with the “ friend count” and put it in a text file. Finally I perform the mean,standard deviation and median calculations to get the requirements.

The following is the piece of code that does all of that:

```
1 import re
2 import math
3 import statistics
4
5
6 def paresingxml():
7     count = 0
8     with open('xml.txt', 'r') as searchfile:
9         with open('freindscount.txt', 'w') as wf:
10             for line in searchfile:
11                 if '"friend_count"' in line:
12                     count = count +1
13                     counter = str(int(count))
14                     wf.write(counter)
15
16
17 def extractingnumber():
18     with open('freindscount.txt', 'r') as fc ,open('numbers
19         count.txt ','w') as nc:
20             for line in fc:
21                 second="".join(re.findall(r'\d+', line)) +"\n"
22                 nc.write(second)
23
24 def calculateMean():
25     lis=[]
26     with open ('numbers count.txt ', 'r') as nc:
27         total = sum(int(x))
28         for line in nc:
29             for x in line.split()
30                 #print ("Total = ",total)
```

```

31     mean = total /154
32     print ("Mean = ",mean)
33     return mean
34
35
36
37 def calculateSD (m):
38     ls=[]
39     i= 0
40     with open ('numbers count.txt ','r') as nc:
41         for line in nc:
42             subNo = int(line) - m
43             SquareSubno= subNo *subNo
44             ls.append(SquareSubno)
45     SqTotal = sum(ls)
46     Mean= SqTotal/ m
47     STD= math.sqrt(SqTotal)
48     print ("STD = ",STD)
49
50
51
52
53
54
55 def calculateMedian():
56     ls=[]
57     with open ('numbers count.txt ','r') as nc:
58         for line in nc:
59             ls.append(line.strip('\n'))
60     ls =list(map(int, ls))
61     #print (ls)
62     Median = statistics.median(ls)
63     print ("Median = ",Median)
64
65
66
67 #calculateMean()
68 paresingxml()
69 extractingnumber()
70 mean = calculateMean()
71 calculateSD (mean)
72 calculateMedian()

```

Listing 1: python code to get mean and standard deviation

```
Mohameds-MacBook-Pro:q1 mohamedshaaban$ python3 Q1-A4.py
Mean = 358.987012987013
STD = 4596.256299862528
Median = 266.5
Mohameds-MacBook-Pro:q1 mohamedshaaban$ █
```

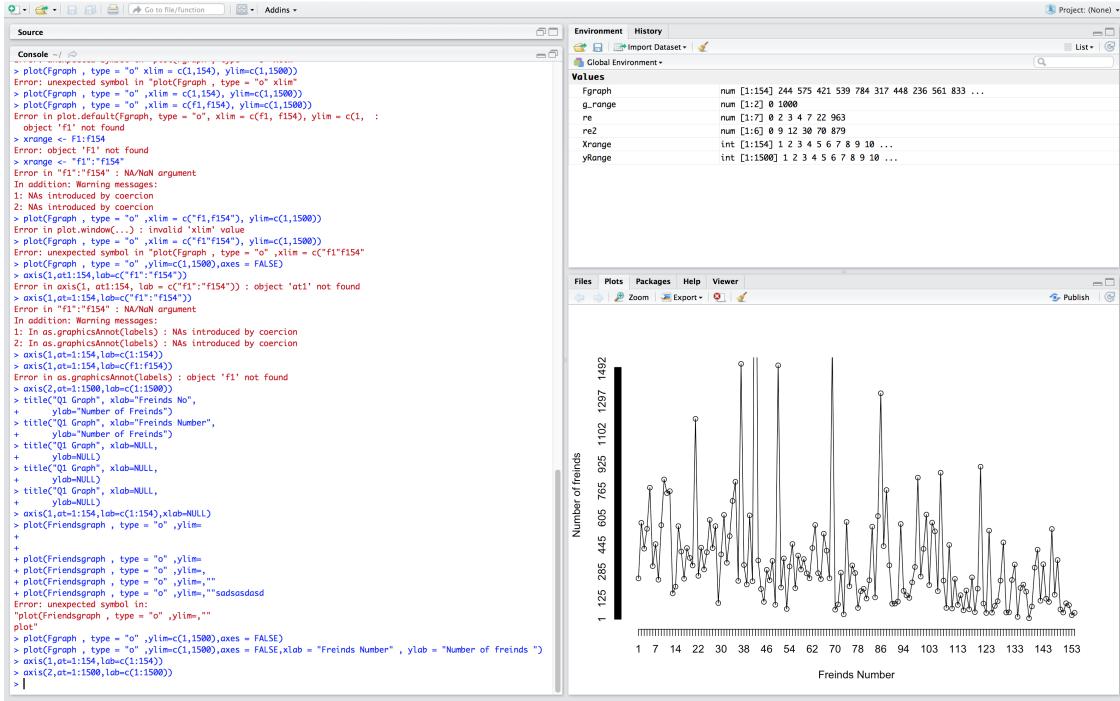


Figure 2: Graphgh in R

From the file “numbers count.txt” I drawn the graph above:

## 2 Problem 2

In problem 2, using Twitter API we extract the followers from twitter account after that I use the same program to develop the standard deviation and median to get the calculations based on every and each follower I have in the list.

The python code to get the followers is as follow:

```

1 import time
2 import tweepy
3
4 consumer_key = "SaSW8NrEZRUsGfZlcOEwwJOOx"
5 consumer_secret = ""
6         C4K6cXR2fTVWhTDas4oJaQU4e3eUdfdrjM8qmaZmAEh1Dym2AC"
7 access_key = "4879994613-6ZGf0doVWQIMCTXgXLQZ72lVpwloAhuSIwYc6UH"
8 access_secret = "BnmtgH0gVrMp5lni6c4sZKacW8Zqs6uCKIcBVwDVHVvQp"
9
10 auth = tweepy.OAuthHandler(consumer_key , consumer_secret )
11 auth.set_access_token(access_key , access_secret )
12
13 api = tweepy.API(auth)
```

```

14 user = api.get_user('Shaaban_Migo')
15
16 print "Number of Followers I have is = " + "{0}".format(user.
17   followers_count)
18
19 ids = []
20 i = 0
21 for user in tweepy.Cursor(api.followers, screen_name="farr007",
22   count = 50).items():
23   try:
24     name = api.get_user(user.screen_name)
25     print "Number of Followers "+ user.
26       screen_name + " have is = " + "{0}".
27         format(name.followers_count)
28     i+=1
29   except tweepy.TweepError, e:
30     if e == "[{u'message': u'Rate limit
31       exceeded', u'code': 88}]":
32       time.sleep(60*15) #Sleep for 5
33         minutes
34     else:
35       time.sleep(60*15)
36     print "Number of Followers "+ user.
37       screen_name + " have is = " + "{0}".
38         format(name.followers_count)
39
#rint e
40 print i
41 print "Number of Followers" + "{0}".format(i)

```

Listing 2: python code to get mean and standard deviation

The screenshot shows a Mac OS X desktop environment. At the top, there is a menu bar with options: Terminal, Shell, Edit, View, Window, Help. Below the menu bar is a dock containing various application icons. In the center, there is a Finder window showing a folder structure with subfolders like C++, Screens shots, Ahmed Flash, CS541, and Resume and Cover. Overlaid on the desktop is a terminal window titled "Q2-A4.py". The terminal contains the following Python code:

```

24
25
26 def calculateMean():
27     lis=[]
28     with open ('numberstwi.txt ','r') as nc:
29         total = sum(int(x)
30         for line in nc
31         for x in line.split())
32         #print ("Total = ",total)
33     mean = total /154
34     print ("Mean = ",mean)
35     return mean
36
37
38
39 def calculateSD(m):
40     ls=[]
41     i= 0
42     with open ('numberstwi.txt ','r') as nc:
43         for line in nc:
44             subNo = int(line) - m
45             SquareSubno= subNo *subNo
46             ls.append(SquareSubno)
47             SqTotal = sum(ls)
48             Mean= SqTotal/ l
49             STD= math.sqrt(SqTotal)

```

Below the code, the terminal shows the output of running the script:

```

Mohameds-MacBook-Pro:Q2 mohamedshaaban$ python3 Q2-A4.py
Mean = 215594.0909090909
STD = 21589264.08301531
Median = 313.0
Mohameds-MacBook-Pro:Q2 mohamedshaaban$ 

```

The terminal also has tabs for q1.py, Assign2-1.py, a4.py, Twi.txt, and Q2-A4.py. At the bottom of the terminal window, there are buttons for Find Prev, Find All, Tab Size: 4, and Python.

Figure 3: running the same program created a text file as shown in line 1 “26”. we perform on this text file exactly what we did on the text file to get these numbers

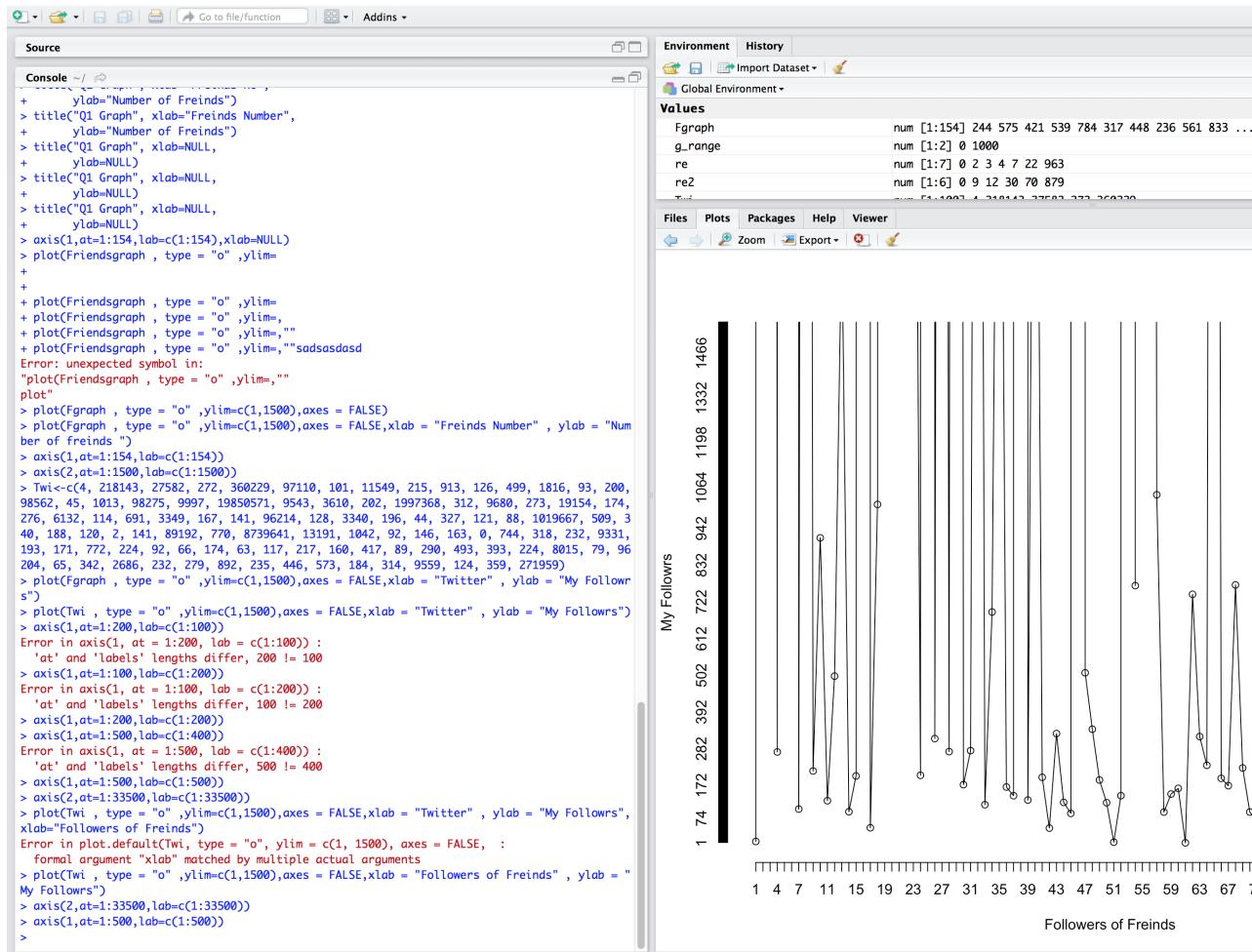


Figure 4: Finally the graph represented in the following figure based on the calculations