Introduction to Web Science/595: Assignment #5

Dr. Nelson

Francis Pruter

Thursday, 16 October, 2014

Contents

Problem 1

1. Explore the friendship paradox for your Twitter account. Since Twitter has directional links (i.e., "followers" and "following"), we'll be investigating if the people you follow (Twitter calls these people "friends") follow more people than you. If you are following < 50 people, use my twitter account "phonedude_mln" instead of your own.

Create a graph of the number of friends (y-axis) and the friends sorted by number of friends (x-axis). (The friends don't need to be labeled on the x-axis as "Bob", "Mary", etc. -- just 1, 2, 3 ...) In other words, if you have 100 friends your x-axis will be 1..101 (100 + you), and the y-axis value will be number of friends that each of those friends has. The friend with the lowest number of friends will be first and the friend with the highest number of friends will be last.

Do include yourself in the graph and label yourself accordingly. Compute the mean, standard deviation, and median of the number of friends that your friends have.

The appropriate part of the Twitter API to use is:

https://dev.twitter.com/rest/reference/get/friends/list

SOLUTION

In order to solve this, I used Twython's get_friends_list() to retrieve the friend list and extract the number of friends for each.

Listing 1: twitterFriendCounter.py

```
from twython import Twython #http://twython.readthedocs.org/en/latest/
    # used http://www.craigaddyman.com/mining-all-tweets-with-python/
import time
import requests
from collections import OrderedDict


REQUEST_TOKEN_URL = "https://api.twitter.com/oauth/request_token"
AUTHORIZE_URL = "https://api.twitter.com/oauth/authorize?oauth_token="
ACCESS_TOKEN_URL = "https://api.twitter.com/oauth/access_token"

CONSUMER_KEY = "K7ddHMoZcHXu0WyWdnp6wdQxf"
CONSUMER_SECRET = "OcvSpsGJ5e1f9GvHbT1y5TXLzwETVkL9wkd1WDzL3FHgC7WDzk"

ACCESS_KEY = "2539409241-FFZPA5lh2vZklUx3q5jASFesRVVXPxqfS6nY8QB"
ACCESS_SECRET = "zzQVXih1Gy2cR1WdMtcMhmXbWpi1S45Jc7vlLlkHFWwm9"

twitter = Twython(CONSUMER_KEY,CONSUMER_SECRET,ACCESS_KEY,ACCESS_SECRET)
```

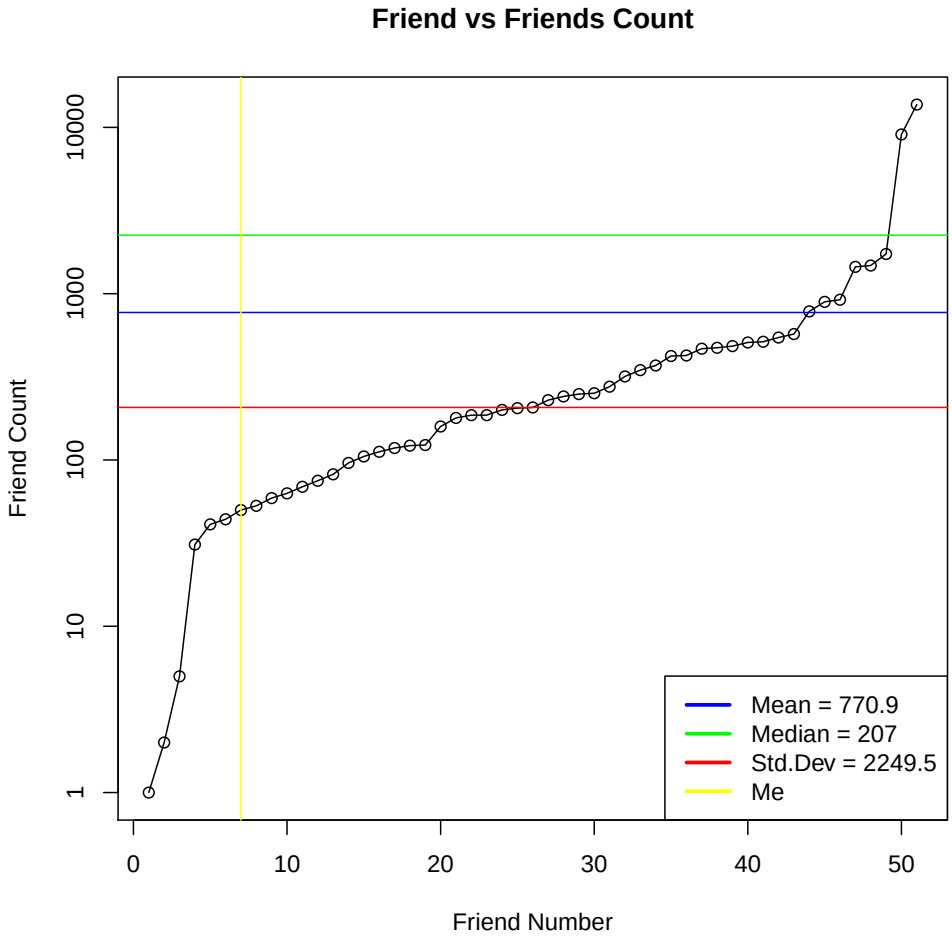
Problem 1 continued on next page...

```
20
user_Friends = twitter.get_friends_list(screen_name="Francis_Pruter", count=200,
    skip_status='true', include_user_entities='false')

25 ctr = 1

fcountlist={}

for friend in user_Friends['users']:
30   fcountlist[friend['name']] = friend['friends_count']
  ctr = ctr+1

fcountlist["ME"] = ctr-1

35 #sort list by friend count https://docs.python.org/2/library/collections.html#
    ordereddict-examples-and-recipes

fcountlist = OrderedDict(sorted(fcountlist.items(), key=lambda t: t[1]))

print fcountlist
40

ctr=1
with open('friendslist', 'w+') as f:
  for friend in fcountlist:
    f.write( str(ctr) + ":" + friend + ':' + str(fcountlist[friend]) + '\n' )
45    ctr=ctr+1
```

Problem 1 continued on next page...

Below is the plot for the extracted data. The graph below is a representation of the number of friends (y-axis) and the friends sorted by number of friends (x-axis).

mean = 770.8627
std dev = 2249.457
median = 207



As shown by the graph, the "friendship paradox" that says your friends have more friends th

References

[1] http://stackoverflow.com/questions/3453695/adding-text-to-a-plot

[2] http://www.r-bloggers.com/adding-a-legend-to-a-plot/

[3] http://twython.readthedocs.org/en/latest/

[4] http://www.craigaddyman.com/mining-all-tweets-with-python/

[5] https://docs.python.org/2/library/collections.html#ordereddict-examples-and-recipes