Assignment Six

Jacob Berlin

CS432 – Spring 2016

1. D3 graphing (5 points)

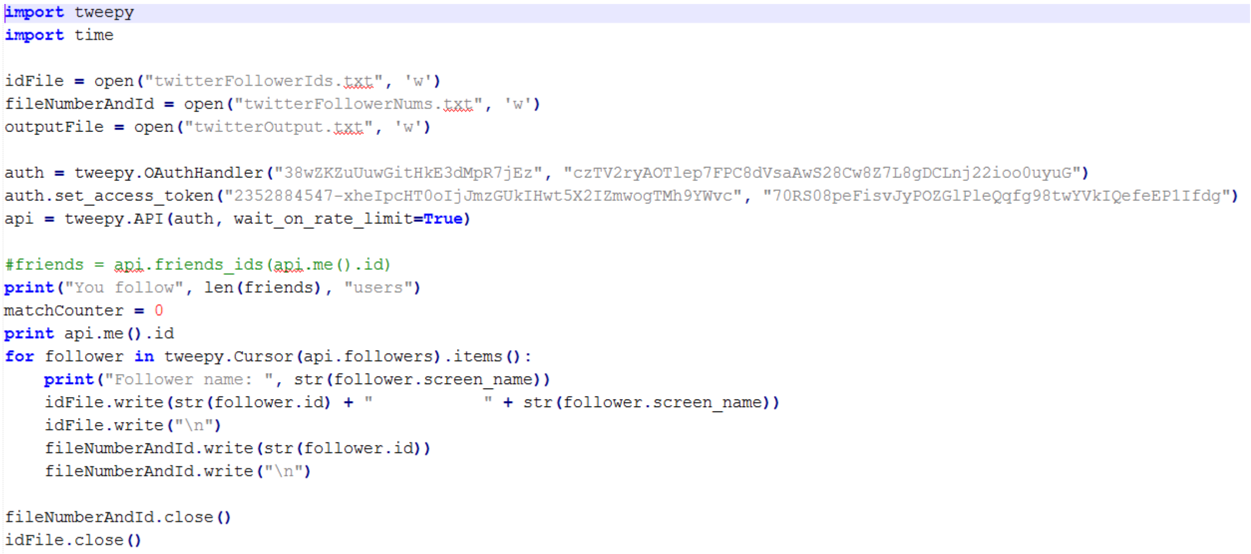
Use D3 to visualize your Twitter followers. Use my twitter account ("@phonedude_mln") if you do not have >= 50 followers. For example, @hvdsomp follows me, as does @mart1nkle1n. They also follow each other, so they would both have links to me and links to each other.

To see if two users follow each other, see: https://dev.twitter.com/rest/reference/get/friendships/show

Attractiveness of the graph counts! Nodes should be labeled (avatar images are even better), and edge types (follows, following) should be marked.

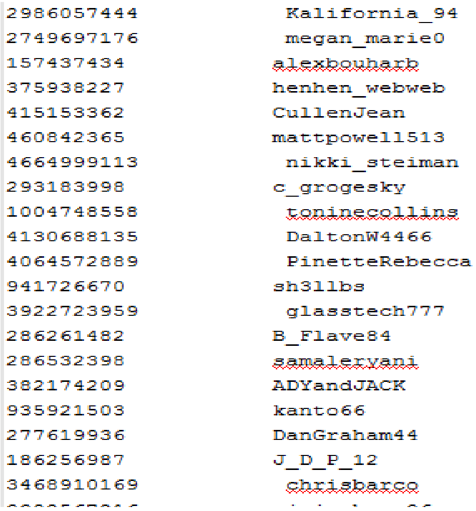
Note: for getting GitHub to serve HTML (and other media types), see: http://stackoverflow.com/questions/6551446/can-i-run-html-files-directly-from-github-instead-of-just-viewing-their-source

Be sure to include the URI(s) for your D3 graph in your report.

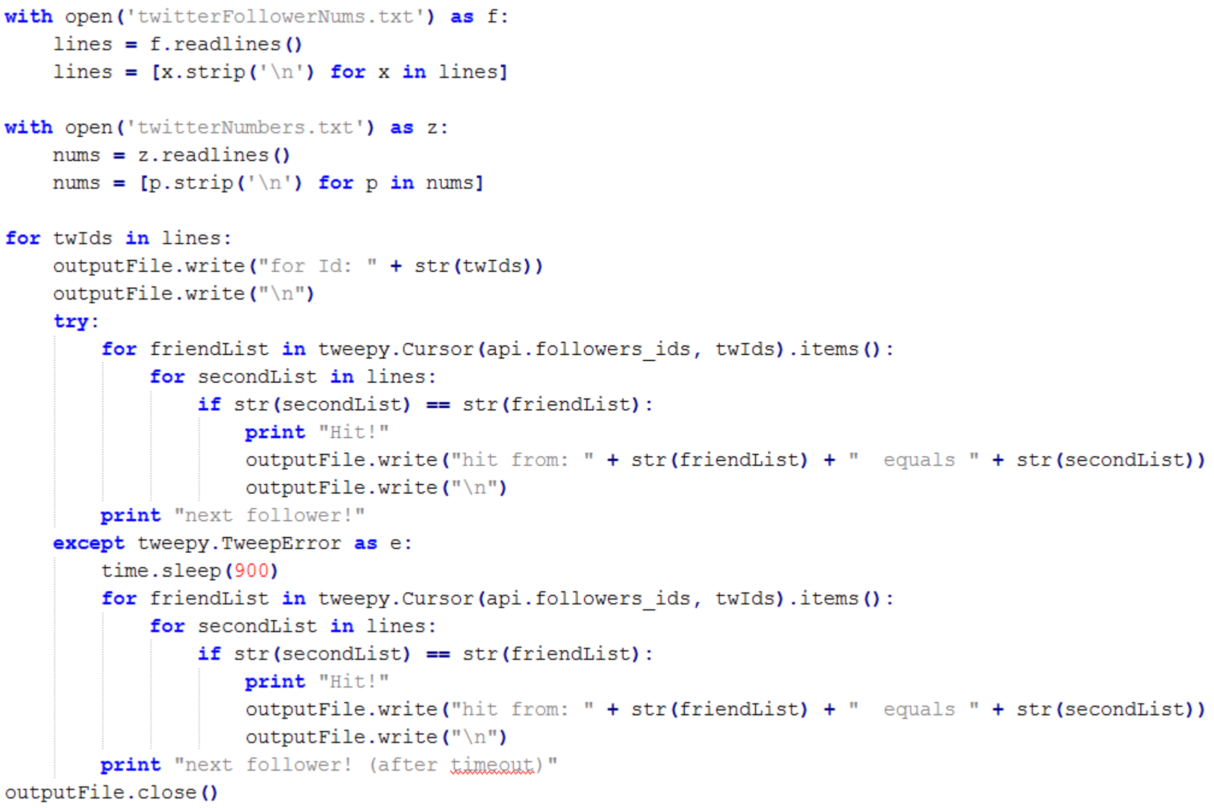


```
import tweepy
import time

idFile = open("twitterFollowerIds.txt", 'w')
fileNumberAndId = open("twitterFollowerNums.txt", 'w')
outputFile = open("twitterOutput.txt", 'w')

auth = tweepy.OAuthHandler("38wZKZuUuwGitHkE3dMpR7jEz", "czTV2ryAOTlep7FPC8dVsaAwS28Cw8Z7L8gDCLnj22ioo0uyuG")
auth.set_access_token("2352884547-xheIpcHT0oIjJmzGUkIHwt5X2IZmwogTMh9YWvc", "70RS08peFisvJyPOZGlPleQqfg98twYVkIQefeEP1Ifdg")
api = tweepy.API(auth, wait_on_rate_limit=True)

#friends = api.friends_ids(api.me().id)
print("You follow", len(friends), "users")
matchCounter = 0
print api.me().id
for follower in tweepy.Cursor(api.followers).items():
    print("Follower name: ", str(follower.screen_name))
    idFile.write(str(follower.id) + "            " + str(follower.screen_name))
    idFile.write("\n")
    fileNumberAndId.write(str(follower.id))
    fileNumberAndId.write("\n")

fileNumberAndId.close()
idFile.close()
```

To start this problem, I created a python program using the Tweepy library. What this did is it went through and grabbed all of the people that I (@JuicyJake1868) have following me. I received both their Id numbers and their Usernames shown in twitterFollowerIds.txt. At the end, I used 52 followers.

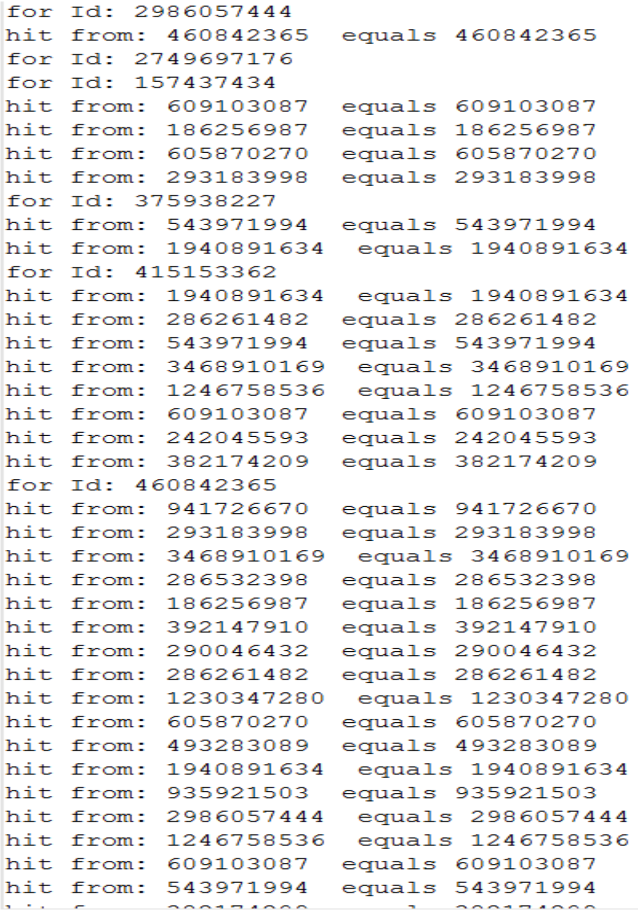


```
2986057444            Kalifornia_94
2749697176            megan_marie0
157437434            alexbouharb
375938227            henhen_webweb
415153362            CullenJean
460842365            mattpowell513
4664999113            nikki_steiman
293183998            c_grogesky
1004748558            toninecollins
4130688135            DaltonW4466
4064572889            PinetteRebecca
941726670            sh3llbs
3922723959            glasstech777
286261482            B_Flave84
286532398            samaleryani
382174209            ADYandJACK
935921503            kanto66
277619936            DanGraham44
186256987            J_D_P_12
3468910169            chrisbarco
```

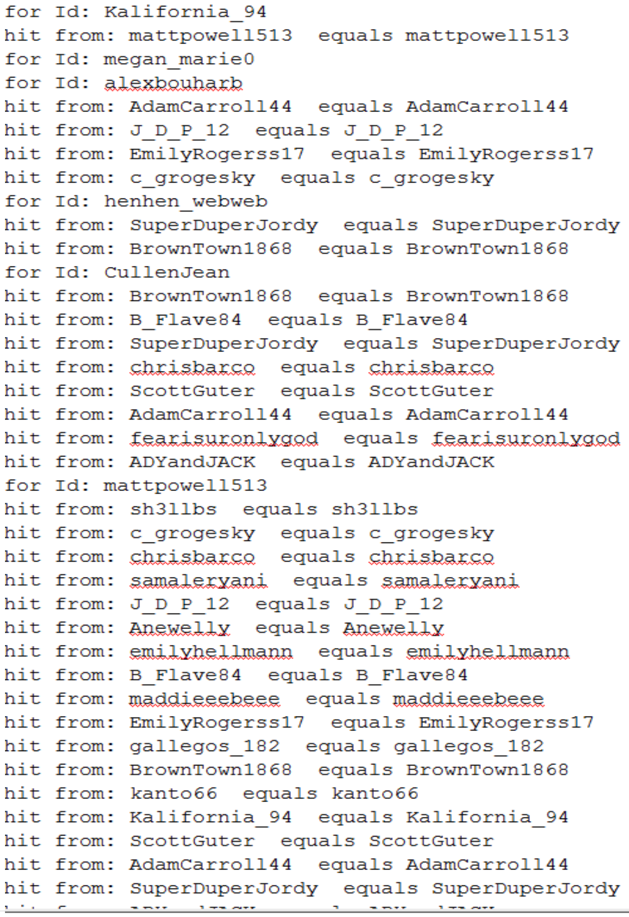


```
with open('twitterFollowerNums.txt') as f:
    lines = f.readlines()
    lines = [x.strip('\n') for x in lines]

with open('twitterNumbers.txt') as z:
    nums = z.readlines()
    nums = [p.strip('\n') for p in nums]

for twIds in lines:
    outputFile.write("for Id: " + str(twIds))
    outputFile.write("\n")
    try:
        for friendList in tweepy.Cursor(api.followers_ids, twIds).items():
            for secondList in lines:
                if str(secondList) == str(friendList):
                    print "Hit!"
                    outputFile.write("hit from: " + str(friendList) + "  equals " + str(secondList))
                    outputFile.write("\n")
        print "next follower!"
    except tweepy.TweepError as e:
        time.sleep(900)
        for friendList in tweepy.Cursor(api.followers_ids, twIds).items():
            for secondList in lines:
                if str(secondList) == str(friendList):
                    print "Hit!"
                    outputFile.write("hit from: " + str(friendList) + "  equals " + str(secondList))
                    outputFile.write("\n")
        print "next follower! (after timeout)"
outputFile.close()
```

After I had all of that data, I extended the python program to go into each of the follower's accounts and had grabbed all of the connections to each of my followers. When doing that, the Ids for each of the accounts were formatted and placed into the twitterOutput.txt file. The information was very hard to read in that state, so I went through and changed each person's Id into their screen-name.

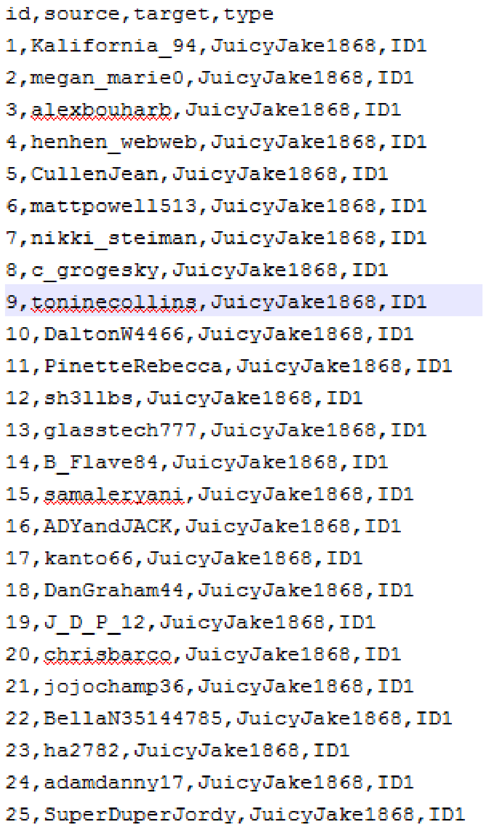


```
for Id: 2986057444
hit from: 460842365  equals 460842365
for Id: 2749697176
for Id: 157437434
hit from: 609103087  equals 609103087
hit from: 186256987  equals 186256987
hit from: 605870270  equals 605870270
hit from: 293183998  equals 293183998
for Id: 375938227
hit from: 543971994  equals 543971994
hit from: 1940891634  equals 1940891634
for Id: 415153362
hit from: 1940891634  equals 1940891634
hit from: 286261482  equals 286261482
hit from: 543971994  equals 543971994
hit from: 3468910169  equals 3468910169
hit from: 1246758536  equals 1246758536
hit from: 609103087  equals 609103087
hit from: 242045593  equals 242045593
hit from: 382174209  equals 382174209
for Id: 460842365
hit from: 941726670  equals 941726670
hit from: 293183998  equals 293183998
hit from: 3468910169  equals 3468910169
hit from: 286532398  equals 286532398
hit from: 186256987  equals 186256987
hit from: 392147910  equals 392147910
hit from: 290046432  equals 290046432
hit from: 286261482  equals 286261482
hit from: 1230347280  equals 1230347280
hit from: 605870270  equals 605870270
hit from: 493283089  equals 493283089
hit from: 1940891634  equals 1940891634
hit from: 935921503  equals 935921503
hit from: 2986057444  equals 2986057444
hit from: 1246758536  equals 1246758536
hit from: 609103087  equals 609103087
hit from: 543971994  equals 543971994
```



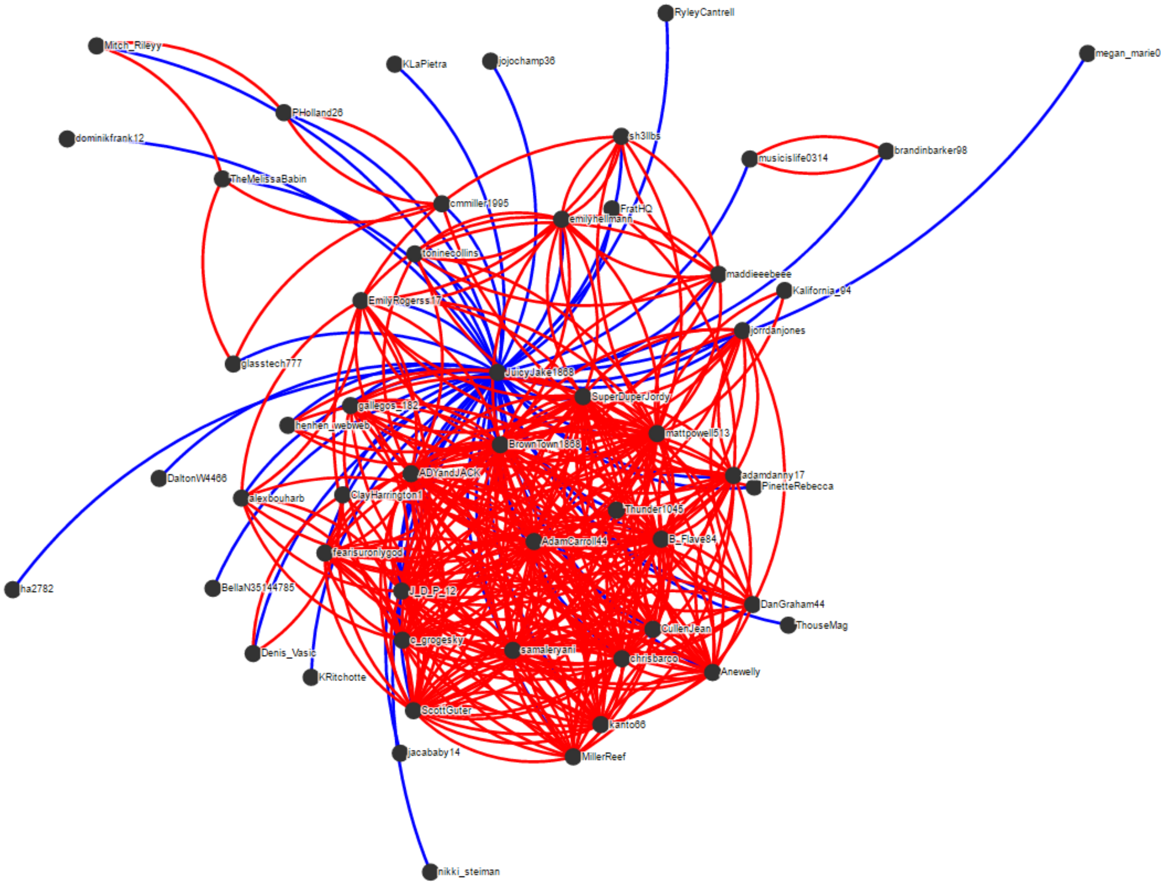
```
for Id: Kalifornia_94
hit from: mattpowell513  equals mattpowell513
for Id: megan_marie0
for Id: alexbouharb
hit from: AdamCarroll44  equals AdamCarroll44
hit from: J_D_P_12  equals J_D_P_12
hit from: EmilyRogerss17  equals EmilyRogerss17
hit from: c_grogesky  equals c_grogesky
for Id: henhen_webweb
hit from: SuperDuperJordy  equals SuperDuperJordy
hit from: BrownTown1868  equals BrownTown1868
for Id: CullenJean
hit from: BrownTown1868  equals BrownTown1868
hit from: B_Flave84  equals B_Flave84
hit from: SuperDuperJordy  equals SuperDuperJordy
hit from: chrisbarco  equals chrisbarco
hit from: ScottGuter  equals ScottGuter
hit from: AdamCarroll44  equals AdamCarroll44
hit from: fearisuronlygod  equals fearisuronlygod
hit from: ADYandJACK  equals ADYandJACK
for Id: mattpowell513
hit from: sh3llbs  equals sh3llbs
hit from: c_grogesky  equals c_grogesky
hit from: chrisbarco  equals chrisbarco
hit from: samaleryani  equals samaleryani
hit from: J_D_P_12  equals J_D_P_12
hit from: Anewelly  equals Anewelly
hit from: emilyhellmann  equals emilyhellmann
hit from: B_Flave84  equals B_Flave84
hit from: maddieeebeee  equals maddieeebeee
hit from: EmilyRogerss17  equals EmilyRogerss17
hit from: gallegos_182  equals gallegos_182
hit from: BrownTown1868  equals BrownTown1868
hit from: kanto66  equals kanto66
hit from: Kalifornia_94  equals Kalifornia_94
hit from: ScottGuter  equals ScottGuter
hit from: AdamCarroll44  equals AdamCarroll44
hit from: SuperDuperJordy  equals SuperDuperJordy
```

After grabbing all of this data, I entered the data into test.csv which contained the formatting required for my d3 graph.



```
id,source,target,type
1,Kalifornia_94,JuicyJake1868,ID1
2,megan_marie0,JuicyJake1868,ID1
3,alexbouharb,JuicyJake1868,ID1
4,henhen_webweb,JuicyJake1868,ID1
5,CullenJean,JuicyJake1868,ID1
6,mattpowell513,JuicyJake1868,ID1
7,nikki_steiman,JuicyJake1868,ID1
8,c_grogesky,JuicyJake1868,ID1
9,toninecollins,JuicyJake1868,ID1
10,DaltonW4466,JuicyJake1868,ID1
11,PinetteRebecca,JuicyJake1868,ID1
12,sh3llbs,JuicyJake1868,ID1
13,glasstech777,JuicyJake1868,ID1
14,B_Flave84,JuicyJake1868,ID1
15,samaleryani,JuicyJake1868,ID1
16,ADYandJACK,JuicyJake1868,ID1
17,kanto66,JuicyJake1868,ID1
18,DanGraham44,JuicyJake1868,ID1
19,J_D_P_12,JuicyJake1868,ID1
20,chrisbarco,JuicyJake1868,ID1
21,jojochamp36,JuicyJake1868,ID1
22,BellaN35144785,JuicyJake1868,ID1
23,ha2782,JuicyJake1868,ID1
24,adamdanny17,JuicyJake1868,ID1
25,SuperDuperJordy,JuicyJake1868,ID1
```

Finally, with all of the data received I was able to create the D3 graph to amply show all of the data.

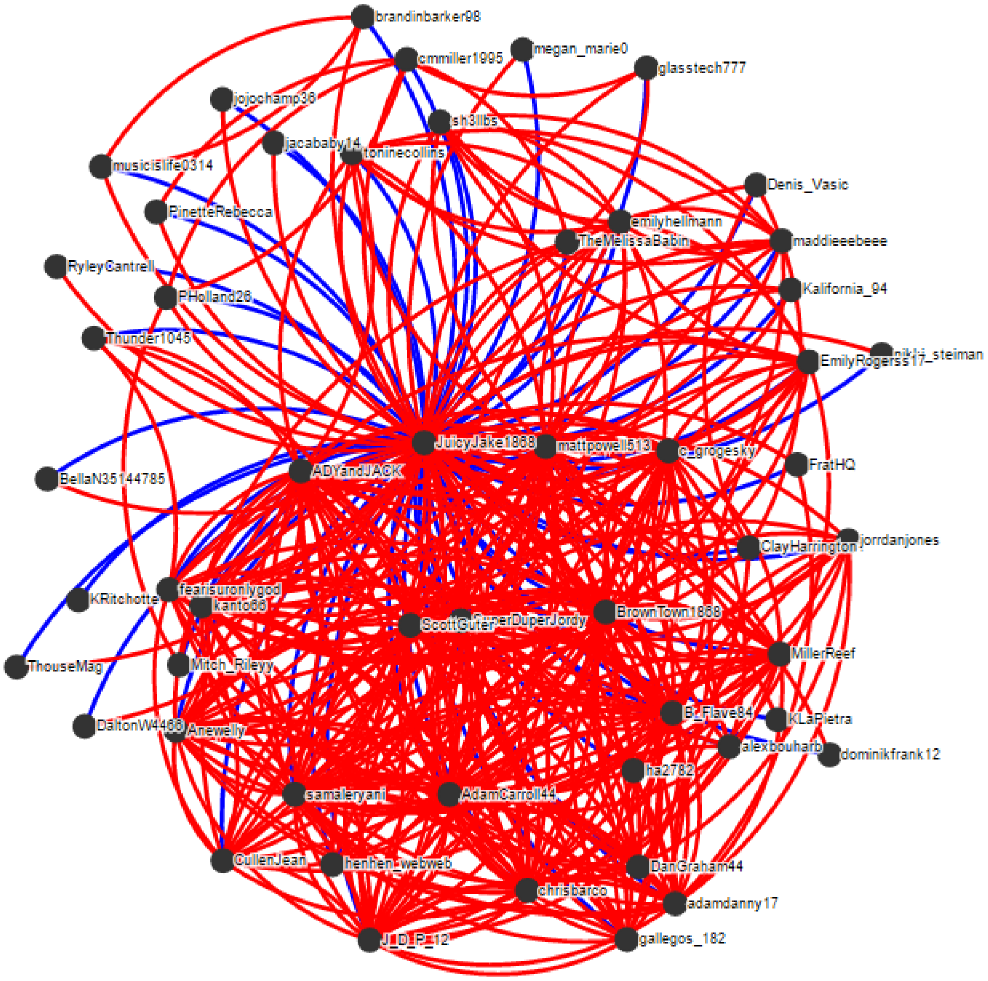


This graph shows all of the collections of data with myself and my connections shown in blue, while everyone else is shown in red. Some observations to notice are that the great jumble of red at the bottom are all of my fraternity brothers and sorority friends that go to Old Dominion University. The small links together near the top are some friends from Stafford High School that connect with each other. The random blue links around the place are some friends that I have met that do not go to Old Dominion University or go to Stafford High School. I have taken out the connections from myself to others in this graph to show everything more clearly.

Link to access graph:

http://htmlpreview.github.io/?https://github.com/Jberl002/D3Work/blob/master/relationGraph.html

This is the graph with my personal follow links added.



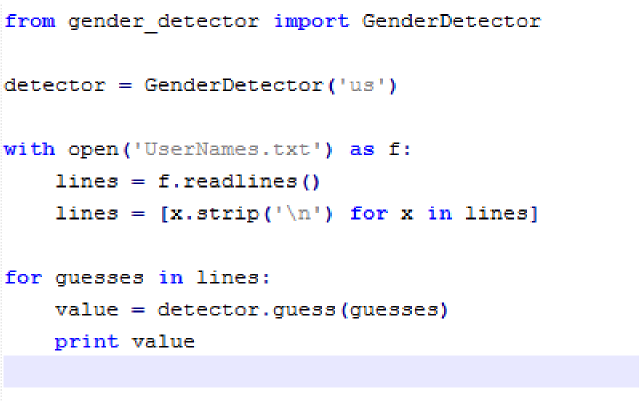
2. Gender homophily in your Twitter graph (5 points)

Take the Twitter graph you generated in question #1 and test for male-female homophily. For the purposes of this question you can consider the graph as undirected (i.e., no distinction between "follows" and "following"). Use the twitter name (not "screen name"; for example "Michael L. Nelson" and not "@phonedude_mln") and programatically determine if the user is male or female.

Create a table of Twitter users and their likely gender. List any accounts that can't be determined and remove them from the graph.

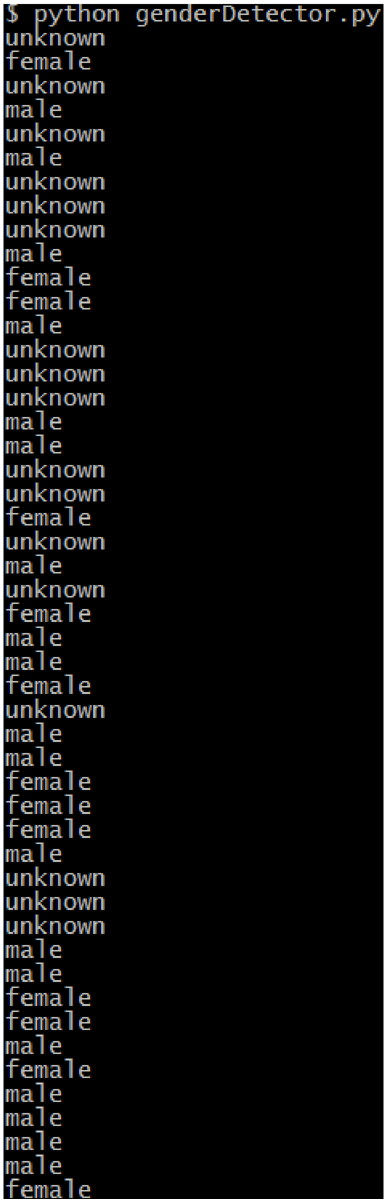
Perform the homophily test as described in slides 11-15, Week 7.

Does your Twitter graph exhibit gender homophily?

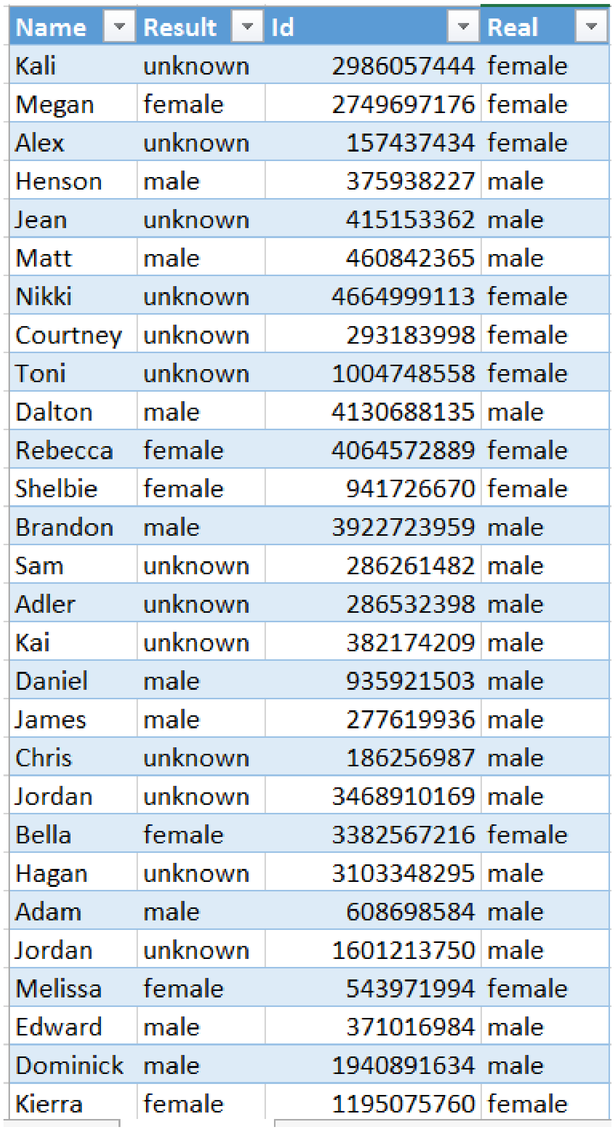


```
from gender_detector import GenderDetector

detector = GenderDetector('us')

with open('UserNames.txt') as f:
    lines = f.readlines()
    lines = [x.strip('\n') for x in lines]

for guesses in lines:
    value = detector.guess(guesses)
    print value
```

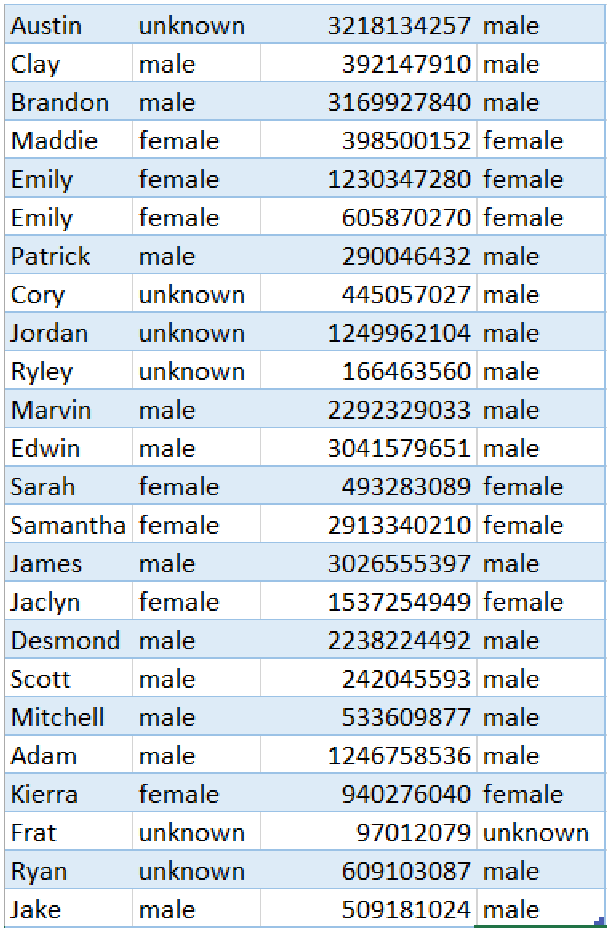
For this problem, I initially created a program using the GenderDetector library in python. This was an each creation, where all I had to do was pass in the names of the people who were following me on Twitter. The only issues that I ran into were cases where the name was of a group (e.g.: FratHQ) which neither is a female or a male. After getting all of this data, I excluded the names that came out as unknown and calculated the homophily. My results are all shown on the next few pages.



```
$ python genderDetector.py
unknown
female
unknown
male
unknown
male
unknown
unknown
unknown
male
female
female
male
unknown
unknown
unknown
male
male
unknown
unknown
female
unknown
male
unknown
female
male
male
female
unknown
male
male
female
female
female
male
unknown
unknown
unknown
male
male
female
female
male
female
male
male
male
male
female
```

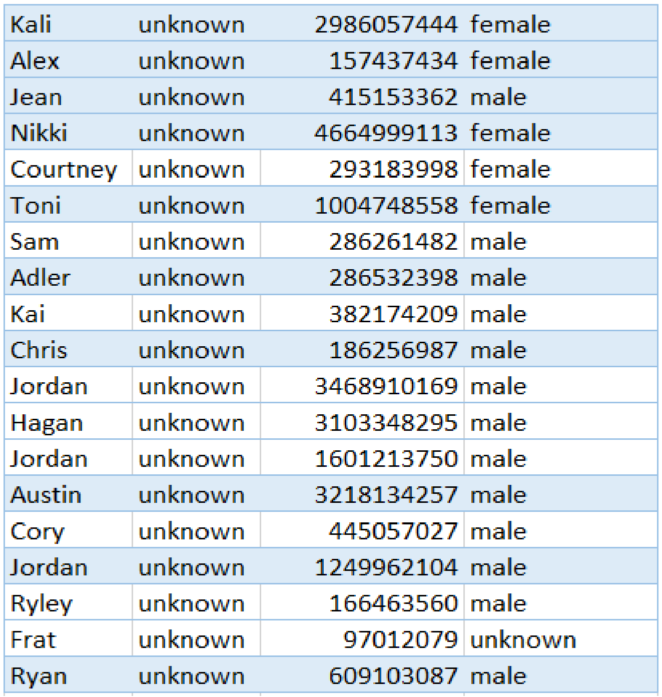


Name	Result	Id	Real
Kali	unknown	2986057444	female
Megan	female	2749697176	female
Alex	unknown	157437434	female
Henson	male	375938227	male
Jean	unknown	415153362	male
Matt	male	460842365	male
Nikki	unknown	4664999113	female
Courtney	unknown	293183998	female
Toni	unknown	1004748558	female
Dalton	male	4130688135	male
Rebecca	female	4064572889	female
Shelbie	female	941726670	female
Brandon	male	3922723959	male
Sam	unknown	286261482	male
Adler	unknown	286532398	male
Kai	unknown	382174209	male
Daniel	male	935921503	male
James	male	277619936	male
Chris	unknown	186256987	male
Jordan	unknown	3468910169	male
Bella	female	3382567216	female
Hagan	unknown	3103348295	male
Adam	male	608698584	male
Jordan	unknown	1601213750	male
Melissa	female	543971994	female
Edward	male	371016984	male
Dominick	male	1940891634	male
Kierra	female	1195075760	female



Austin	unknown	3218134257	male
Clay	male	392147910	male
Brandon	male	3169927840	male
Maddie	female	398500152	female
Emily	female	1230347280	female
Emily	female	605870270	female
Patrick	male	290046432	male
Cory	unknown	445057027	male
Jordan	unknown	1249962104	male
Ryley	unknown	166463560	male
Marvin	male	2292329033	male
Edwin	male	3041579651	male
Sarah	female	493283089	female
Samantha	female	2913340210	female
James	male	3026555397	male
Jaclyn	female	1537254949	female
Desmond	male	2238224492	male
Scott	male	242045593	male
Mitchell	male	533609877	male
Adam	male	1246758536	male
Kierra	female	940276040	female
Frat	unknown	97012079	unknown
Ryan	unknown	609103087	male
Jake	male	509181024	male

Names that were excluded from calculation:



Kali	unknown	2986057444	female
Alex	unknown	157437434	female
Jean	unknown	415153362	male
Nikki	unknown	4664999113	female
Courtney	unknown	293183998	female
Toni	unknown	1004748558	female
Sam	unknown	286261482	male
Adler	unknown	286532398	male
Kai	unknown	382174209	male
Chris	unknown	186256987	male
Jordan	unknown	3468910169	male
Hagan	unknown	3103348295	male
Jordan	unknown	1601213750	male
Austin	unknown	3218134257	male
Cory	unknown	445057027	male
Jordan	unknown	1249962104	male
Ryley	unknown	166463560	male
Frat	unknown	97012079	unknown
Ryan	unknown	609103087	male

For calculating the homophily:

33 total links

13 female

20 male

P = 20/33 = .606

Q = 13/33 = .394

2PQ = 0.477528

Based on the graph from part one, the number of cross-gender links are:

Total links in graph: 282

Amount of cross-gender links = 83

C-G links / total = cross-gender chance

83/282 = .2943

Therefore, since the cross-gender chance (.2943) is much lower than the 2pq calculation (0.477528), my graph does show evidence of homophily.