Part 1

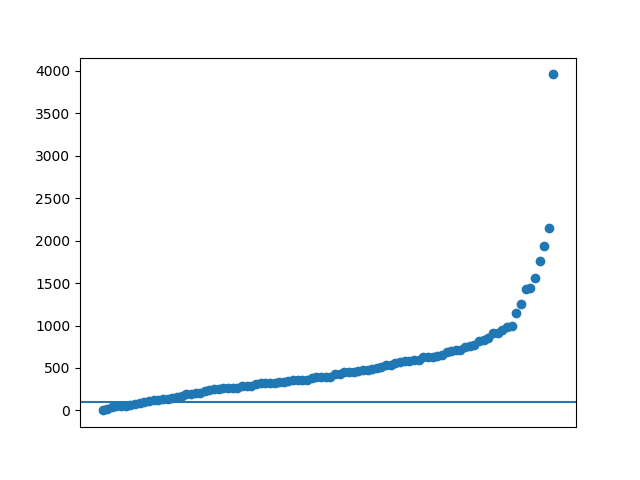
Wrote a python script using pandas addfriends.py that reads a csv formatted as the one provided and outputs the mean, median, and standard deviation for the friend counts in the CSV file. Results were:

mean = 542.6734693877551

median = 396.0

standard deviation = 539.4337385239659

Wrote a script using pandas and matplotlib makegraph.py that outputs a simple graph showing friend counts arranged lowest to highest, with a horizontal line representing the source user (in this case the instructor) who has 98 friends on facebook.



The graph clearly shows that the majority of the instructor’s friends on facebook have more friends than himself. This is backed up by the value of the mean and median, so the friendship paradox holds here.

Part 2

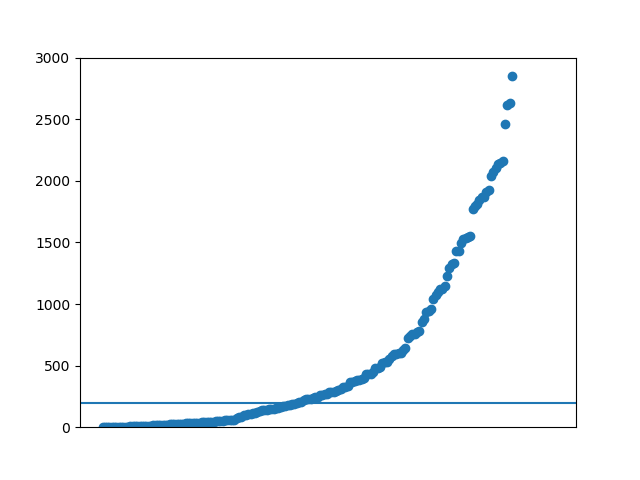
Using code in <https://psylica.com/blog/2017/May/03/listing-twitter-followers-tweepy.html>

as a base, wrote a script that pulls followers and number of followers’ followers into a csv file formatted the same as the one provided for facebook data, getfollowerlist.py. Using the scripts from part one, calculated the mean, median, and standard deviation of the dataset and created a graph.

mean = 6898.790816326531

meadian = 276.5

standard deviation = 56865.32356557514



For the graph, I left out the highest values because they were so great in magnitude they made the rest of the graph meaningless. By the calculated median, the friendship paradox appears to hold for the instructor’s twitter account. The graph for the number of twitter followers intersect the instructor’s follower count closer to the middle of the graph than it did on the graph for facebook, but the intersection is still on the left side of the graph, so it supports that the friendship paradox holds.

Code Used

#addfriends.py

#do calculations on freinds to see if friendship paradox holds

#will take .csv file with format "USER, FRIENDCOUNT"

#import csv

import sys

import pandas as pd

#import numpy as np

#from collections import Counter

infile = sys.argv[1] #input file

outfile = sys.argv[2] #output file

fields = ['USER', 'FRIENDCOUNT']

df = pd.read\_csv(infile, encoding="utf-8-sig", skipinitialspace=True, usecols=fields)

fmean = df['FRIENDCOUNT'].mean()

fmedian = df['FRIENDCOUNT'].median()

fstd = df['FRIENDCOUNT'].std()

print("mean = " + str(fmean), file=open(outfile, "a"))

print("meadian = " + str(fmedian), file=open(outfile, "a"))

print("standard deviation = " + str(fstd), file=open(outfile, "a"))

#getfollowerlist.py

import tweepy, sys, time

from secrets import \*

outfile = sys.argv[1]

global api

consumer\_key = "B5bqvMF1YcvWSXkjQ1IM2JJFj"

consumer\_secret = "nksLmNL0IVvKoK5OaBrWAr4UJhGD8o9nzvRvpHOQUaAE20RWqH"

access\_key = "367451421-uHFwpmINorzC9krB5NNXbByq2lnLjAfXweEULfe7"

access\_secret = "0uBi2FTWEVyKaHvAzyNq5NuU3Z3OcAvD5eTy3vlezTHPN"

auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)

auth.set\_access\_token(access\_key, access\_secret)

api = tweepy.API(auth)

users = tweepy.Cursor(api.followers, screen\_name="acnwala").items()

print('"USER", "FRIENDCOUNT"', file=open(outfile, "a"))

while True:

try:

user = next(users)

except tweepy.TweepError:

time.sleep(60\*15)

user = next(users)

except StopIteration:

break

#print (user.screen\_name, file=open(outfile, "a"))

name = str(user.screen\_name)

count = str(user.followers\_count)

print (name + ' , ' + count, file=open(outfile, "a"))

#makegraph.py

import matplotlib.pyplot as plt

#import numpy as np

import pandas as pd

import sys

infile = sys.argv[1]

fields = ['USER', 'FRIENDCOUNT']

df = pd.read\_csv(infile, encoding="utf-8-sig", skipinitialspace=True, usecols=fields)

#df.plot(kind='scatter', x='USER', y='FRIENDCOUNT')

#plt.show(block=True)

df2 = result = df.sort\_values(by=['FRIENDCOUNT'])

plt.scatter(df2['USER'], df2['FRIENDCOUNT'])

plt.axhline(y=196)

plt.ylim((0,3000))

plt.tick\_params(

axis='x', # changes apply to the x-axis

which='both', # both major and minor ticks are affected

bottom='off', # ticks along the bottom edge are off

top='off', # ticks along the top edge are off

labelbottom='off') # labels along the bottom edge are off

plt.show()

plt.savefig('followergraph.png')