Assignment Four

Jacob Berlin

CS432 – Spring 2016

1. 1. Determine if the friendship paradox holds for my Facebook account.* Compute the mean, standard deviation, and median of the number of friends that my friends have. Create a graph of the number of friends (y-axis) and the friends themselves, sorted by number of friends (x-axis). (The friends don't need to be labeled on the x-axis: just f1, f2, f3, ... fn.) Do include me in the graph and label me accordingly.

```
<?xml version="1.0" encoding="UTF-8"?>
|<data>
    <node id="Simeon Warner 428351">
        <data key="Label">Simeon Warner</data>
        <data key="uid"><![CDATA[428351]]></data>
        <data key="name"><![CDATA[Simeon Warner]]></data>
        <data key="mutual_friend_count"><![CDATA[13]]></data>
        <data key="friend count"><![CDATA[244]]></data>
    </node>
    <node id="Drew Munro 1314586">
        <data key="Label">Drew Munro</data>
        <data key="uid"><![CDATA[1314586]]></data>
        <data key="name"><![CDATA[Drew Munro]]></data>
        <data key="mutual friend count"><![CDATA[17]]></data>
        <data key="friend_count"><![CDATA[575]]></data>
    </node>
    <node id="Mat_Kelly_2004483">
        <data key="Label">Mat Kelly</data>
        <data key="uid"><![CDATA[2004483]]></data>
        <data key="name"><![CDATA[Mat Kelly]]></data>
        <data key="mutual friend count"><![CDATA[12]]></data>
        <data key="friend count"><![CDATA[421]]></data>
    </node>
    <node id="Benjamin Lok 2037943">
        <data key="Label">Benjamin Lok</data>
        <data key="uid"><![CDATA[2037943]]></data>
        <data key="name"><![CDATA[Benjamin Lok]]></data>
        <data key="mutual friend count"><![CDATA[1]]></data>
        <data key="friend_count"><![CDATA[539]]></data>
    </node>
    <node id="Camden Elliott Matherne 2726573">
        <data key="Label">Camden Elliott Matherne</data>
        <data key="uid"><![CDATA[2726573]]></data>
        <data key="name"><![CDATA[Camden Elliott Matherne]]></data>
        <data key="mutual friend count"><![CDATA[8]]></data>
        <data key="friend count"><![CDATA[784]]></data>
    </node>
     ..... 24 marshara barara waran 020004000
```

Initially for this problem I started with the raw XML format that was retrieved from the 'mln.graphml' file and then created a python program to parse the data.

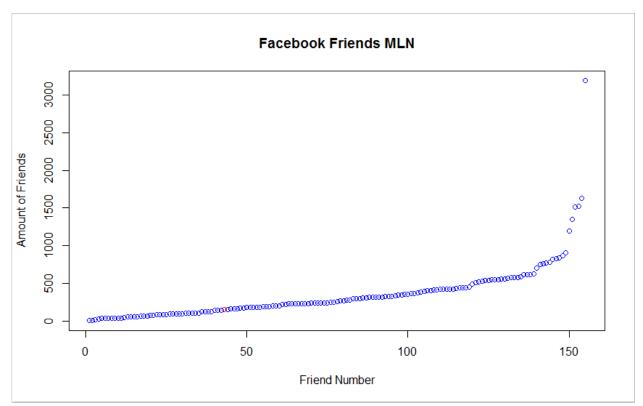
```
#-*- coding: utf-8 -*-
 1
 2
 3
     #URIFile = open('uriFile.txt', 'r')
 4
 5
     import xml.etree.ElementTree as ET
 6
7
     unsortedFile = open('friendNums.txt', 'a')
     outputFile = open('friendsFB.txt', 'a')
8
9
10
     outputCounter = 1
11
     counter = 0
12
     tree = ET.parse ("rawFile.xml")
13
     root = tree.getroot()
14
15
     print "MLN
                         154"
16
     outputFile.write("MLN
                                    154")
     outputFile.write("\n")
17
18
19
     #for initial in root.iter('node'):
          unsortedFile.write(root[counter][4].text)
20
          unsortedFile.write("\n")
21
22
     #
          counter += 1
23
24
   □with open('friendNums.txt') as f:
25
         lines = f.readlines()
26
         lines = [x.strip('\n') for x in lines]
27
28
   □for Nums in lines:
                                                                   " + Nums)
29
         outputFile.write("F" + str(outputCounter) + "
30
         outputFile.write("\n")
31
         outputCounter += 1
```

This file exclusively grabbed the amount of friends that each of the nodes in 'rawFile.xml' had and added it to a file called 'friendNums.txt.' Once I had retrieved all of the friends and the number of friends that each of them have, I sorted the list through the command-line using the sort —n command. After retrieving all of the data required, I parsed through the data and created a file with all of the friends along with adding the 'MLN' value to show you in the data.

25 2 F1 7 19 F18 68 30 3 F2 15 20 F19 68 38 4 F3 25 21 F20 77 40 5 F4 30 22 F21 80 41 6 F5 38 23 F22 85 43 7 F6 39 24 F23 86 54 8 F7 40 25 F24 87 58 8 F7 40 25 F24 87 60 9 F8 41 26 F25 89 60 9 F8 41 26 F25 89 60 9 F8 41 26 F25 89 62 10 F9 41 27 F26 93 65 30 F3 F27 94 48	\$ cat friendNums.txt sort -n	-1	- 1 1	- 1 1- 1-	18	F17	65
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 24 F23 86 41 F40 143 144 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 183 30 F29 96 48 F46 168 186 31 F30 97 49 F47 170 170 195 3	7	1	Friend	FriendAmount			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	25						
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	30 38						
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	39 40						
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	41	5	F4				
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	41 42	6	F5	38			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	43 54	7	F6	39			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	58	8	F7	40			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	60	9	F8	41			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	62 65	10	F9	41			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	68	11	F10	42			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	68 77	12	F11	43			
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	80 85	13	F12	54	30	F29	
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	86	14	F13	58	31	F30	
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	89	15	F14	59	32	F31	104
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	93 94	16	F15	60	33	F32	104
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	94	17		62	34	F33	106
104 19 F18 68 36 F35 111 106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 123 F22 85 40 F39 131 143 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F47 170 195 32 F3	97				35	F34	108
106 20 F19 68 37 F36 123 111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 128 22 F21 80 40 F39 131 143 24 F22 85 40 F39 131 144 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 29 F28 94 47 F45 168 186 30 F29 96 48 F46 168 187 195 32	104				36	F35	111
111 21 F20 77 38 F37 124 123 124 22 F21 80 39 F38 128 128 128 22 F21 80 40 F39 131 143 23 F22 85 41 F40 143 144 24 F23 86 41 F40 143 147 24 F23 86 42 F41 144 147 25 F24 87 43 F42 147 165 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 46 F43 155 181 29 F28 94 47 F45 168 182 30 F29 96 48 F46 168 186 31 F30 97 49 F47 170 195 32 F31	106 108				37	F36	123
124 22 F21 80 39 F38 128 128 128 40 F39 131 131 23 F22 85 40 F39 131 143 24 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 183 30 F29 96 47 F45 168 186 187 31 F30 97 48 F46 168 187 32 F31 104 50 F48 172 204 34 F33 106	111				38	F37	124
123 F22 85 40 F39 131 143 144 F23 86 41 F40 143 147 25 F24 87 42 F41 144 155 26 F25 89 43 F42 147 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 29 F28 94 46 F44 165 183 30 F29 96 48 F46 168 186 31 F30 97 49 F47 170 195 32 F31 104 50 F48 172 207 33 F32 104 51 F49 181 208 35 F34 108 53 F51 183 229 36 F35 111 54 <td< th=""><th>124</th><th></th><th></th><th></th><th>39</th><th>F38</th><th>128</th></td<>	124				39	F38	128
143 144 147 155 168 168 168 168 170 172 181 182 183 184 185 186 170 172 28 F27 94 45 44 MLN 155 181 29 182 94 46 F44 168 180 F29 96 47 48 F46 181 186 31 187 97 48 F46 49 F47 170 195 32 197 33 204 50 207 34 208 52 2207 34 220 35 33 106 52<	131				40	F39	131
147 155 168 168 168 170 172 181 182 183 184 185 186 187 181 182 183 184 185 186 187 188 180 181 182 183 184 185 186 187 187 188 180 180 181 182 183 184 185 186 187 187 188 180 181 182 184 185 184 184 184 184 184 184 184 184 184 184 184 184 184 184 184 184 </th <th>143</th> <th></th> <th></th> <th></th> <th>41</th> <th>F40</th> <th>143</th>	143				41	F40	143
165 26 F25 89 43 F42 147 168 168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 48 F45 168 186 31 F30 97 49 F47 170 195 32 F31 104 50 F48 172 204 37 F33 106 51 F49 181 208 35 F34 108 53 F51 183 229 36 F35 111 54 F52 186	147				42	F41	144
168 27 F26 93 44 MLN 154 170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 186 31 F30 97 49 F46 168 190 32 F31 104 50 F48 172 197 32 F31 104 50 F48 172 204 34 F33 106 52 F50 182 207 36 F34 108 53 F51 183 220 35 F35 111 54 F52 186	165				43	F42	147
170 28 F27 94 45 F43 155 181 29 F28 94 46 F44 165 182 30 F29 96 47 F45 168 183 31 F30 97 48 F46 168 190 32 F31 104 50 F48 172 197 33 F32 104 51 F49 181 204 34 F33 106 52 F50 182 220 35 F34 108 53 F51 183 227 36 F35 111 54 F52 186	168 168				44	MLN	154
181 29 F28 94 46 F44 165 183 30 F29 96 47 F45 168 186 31 F30 97 48 F46 168 190 32 F31 104 49 F47 170 197 32 F31 104 50 F48 172 204 33 F32 104 51 F49 181 207 34 F33 106 52 F50 182 220 35 F34 108 53 F51 183 229 36 F35 111 54 F52 186	170				45	F43	155
183 186 187 190 191 192 193 194 197 197 198 199 199 190 191 192 193 194 195 197 204 207 208 220 231 232 233 106 52 53 53 51 11 54 52 183 231 231	181				46	F44	165
186 187 190 195 197 204 207 204 207 208 220 227 229 231 232 7 7 7 236 7 7 7 106 52 53 53 54 52 111 54 55 123	183				47	F45	168
190 195 197 204 207 208 220 231 34 F33 F32 104 50 F48 172 170 170 170 195 197 198 199 199 190 190 190 190 190 190 190 190	186				48	F46	168
204 207 208 220 220 227 229 231 231 24 F33 26 F35 27 F36 27 F36 27 F36 28 F32 29 F36 20 F37 20 F36 21 F36 21 F36 22 F36 23 F37 23 F36 23 F36 24 F37 25 F36 27 F36 28 F37 29 F36 20 F36 20 F36 21 F36 22 F36 23 F37 24 F37 25 F36 26 F37 27 F38 28 F38 29 F38 20 F38 2	190				49	F47	170
204 207 208 220 220 227 229 231 231 24 F33 26 F35 27 F36 27 F36 27 F36 28 F32 29 F36 20 F37 20 F36 21 F36 21 F36 22 F36 23 F37 23 F36 23 F36 24 F37 25 F36 27 F36 28 F37 29 F36 20 F36 20 F36 21 F36 22 F36 23 F37 24 F37 25 F36 26 F37 27 F38 28 F38 29 F38 20 F38 2	195 197				50	F48	
208 34 F33 106 52 F50 182 220 35 F34 108 53 F51 183 229 36 F35 111 54 F52 186 231 37 F36 123 54 F52 186	204						
35 F34 108 53 F51 183 229 36 F35 111 54 F52 186	208						
229 231 37 F36 123 54 F52 186	227						
37 F26 122	229						
	231	37	F36	123		F52	187

Finally, after retrieving all of this data and formatting it as needed I went to RStudio and put the file into a plot generator. The graph I made shows each of the friends' amount of friends alongside your own value from least to greatest. The 'MLN' value is shown in red while all of the other values are shown in blue.





To calculate the standard deviation, mean, and median of the list was just as simple as putting the data into R's built in math functions. I made sure to take the 'MLN' value out of the dataset as to not skew the data.

```
- mean(friendsFB$FriendAmount)
[1] 358.987
> median(friendsFB$FriendAmount)
[1] 266.5
> sd(friendsFB$FriendAmount)
[1] 371.5853
```

With your total of 154 friends, the friendship paradox holds for your Facebook account due to the fact that the majority of your friends have more friends on Facebook than you do. This is shown by the fact that your friend amount is smaller than the mean, median, and mode of the dataset.

2. Determine if the friendship paradox holds for your Twitter account. Since Twitter is a directed graph, use "followers" as value you measure (i.e., "do your followers have more followers than you?").

Generate the same graph as in question #1, and calcuate the same mean, standard deviation, and median values.

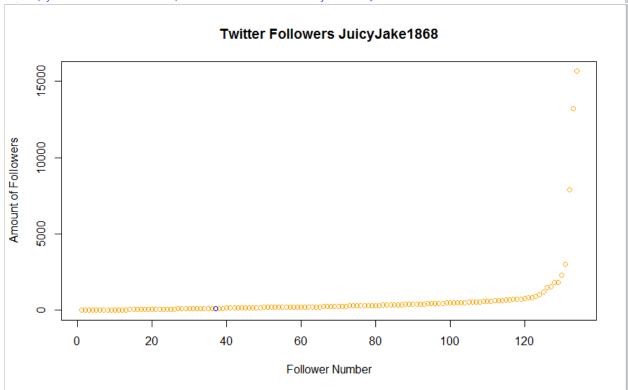
```
1 import time
    import tweepy
    auth = tweepy.OAuthHandler("38wZKZuUuwGitHkE3dMpR7jEz", "czTV2ryAOTlep7FPC8dVsaAwS28Cw8Z7L8gDCLnj22ioo0uyuG")
    auth.set_access_token("2352884547-xheipcHT00ijJmzGUkiHwt5X2IZmwogTMh9YWvc", "70RS08peFisvJyP0ZGlPleQqfg98twYVkiQefeEP1ifdg")
    outputFile = open("twitterFollowerAmt.txt", 'w')
    idFile = open("twitterFollowerIds.txt", 'w')
   counter = 1
    api = tweepy.API(auth)
    ids = []
14
15 For page in tweepy.Cursor(api.followers_ids, screen_name="JuicyJake1868").pages():
16
         ids.extend(page)
        time.sleep(5)
18
19
   print len(ids)
21 For name in ids:
        print name
        user = api.get user(name)
24
25
         print user.followers count
        outputFile.write(str(counter) + " " + str(user.followers_count))
26
         outputFile.write("\n")
         print name
28
         idFile.write(str(name))
29
         idFile.write("\n")
30
        counter += 1
    idFile.close()
    outputFile.close()
```

For question two, I started in the same fashion of question whereas I created a python file to parse through each of my personal Twitter account. Using the 'Tweepy' library, I was able to grab a list of both the follower's ids along with the amount of followers that each of my followers have. Only the ids were stored in the 'twitterFollowersIds.txt'. Just the follower amount values were then sorted using the sort –n command and then the sorted follower numbers were stored in the 'twitterFollowerAmt.txt'

\$ python twitterFollowerIds.py	98	472976560	Fol	lower	FollowerCount
133 460842365	99	1035149784	1	0	
376 460842365	100	100002618	2	2	
4664999113	101	318487460	3	3	
4664999113	102	413598079	4	6	
293183998 731	103	467954890	5	6	
293183998 3064238573	104	243467114	6	8	
128 3064238573	105	346964480	7	14	
1004748558 295	106	2304573203	8	16	
1004748558	107	2230299493	9	19 23	
4130688135 6	108	85005476	11	32	
4130688135 4064572889	109	416048312	12	36	
37 4064572889	110	2345497260	13	37	
941726670 210	111	255741404	14	52	
941726670	112	849931044	15	54	
3922723959 67	113	429110769	16	57	
3922723959 286261482	114	320364459	17	59	
1483 286261482	115	261946574	18	63	
286532398 623	116	232719227	19	64	
286532398	117	1034930119	20	66	
382174209 330	118	312427672	21	67	
382174209 935921503	119	25005076	22	73 75	
283 935921503	120	340047972	24	75 76	
277619936 399	121	346705910	25	77	
277619936 186256987	122	794711202	26	84	
833	123	1122537024	27	93	
186256987 3468910169	124	60784841	28	94	
100 3468910169	125	1708408440	29	100	
3382567216 134	126	250966853	30	103	
3382567216 3103348295	127	465066276	31	105	
3	128	547267877	32	107	
3103348295 608698584	129	438474262	33	123	
84 608698584	130	468602502	34	127	
1601213750 141	131	411211347	35	127 128	
1601213750 543971994	132	403723130	36 37	134	
1836	133	425379468	37	131	

Along with in the first problem, I now had the entire list of data needed to put into RStudio to generate the graph. The specific value for my amount of followers is detailed in blue while my follower's followers are detailed in orange.





To compute the mean, median, and standard deviation I plugged the values into R after making sure to take the 'JWB' value out.

```
> mean(twitterFollowerAmt$FollowerCount)
[1] 637.1343
> median(twitterFollowerAmt$FollowerCount)
[1] 259.5
> sd(twitterFollowerAmt$FollowerCount)
[1] 1881.712
```

Therefore, by the rule of the friendship paradox and with my follower count being only at 133, the friendship paradox still holds for my account. This is due to my follower count being significantly less than all three of the calculated values.

3. E.C. Repeat question #1, but with your LinkedIn profile.

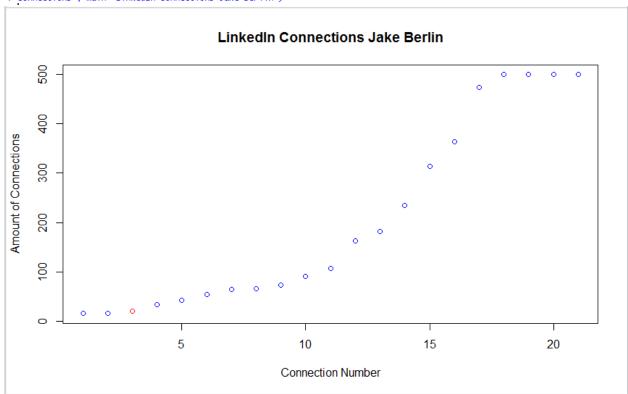
On my LinkedIn profile I only have 20 connections. (Account name is Jake Berlin)

Due to the LinkedIn API not working as of 2/20/2016, I have manually gone through each of my connections and have grabbed all of their data.

Data:

Connection	ConnectionAmt
C1	15
C2	15
JWB	20
C3	33
C4	42
C5	54
C6	64
C7	65
C8	73
C9	90
C10	106
C11	162
C12	181
C13	234
C14	314
C15	364
C16	474
C17	500
C18	500
C19	500
C20	500





As in question 1, I created a plot graph based on the amount of connections I had and the connections that each of my connections had. The connection in red is the 'JWB' connection, showing myself alongside the rest of the data.

I then calculated the standard deviation, mean, and median in R after making sure to take out the 'JWB' value.

```
> sd(LinkedIn$ConnectionAmt)
[1] 190.5517
> median(LinkedIn$ConnectionAmt)
[1] 106
> mean(LinkedIn$ConnectionAmt)
[1] 205.0476
```

Finally after retrieving all of the data needed and with only 20 connections the friendship paradox holds for my LinkedIn account due to the fact that my connection amount is marginally lower than the three values I calculated.