Chloe Quinto
CS 581 Online Social Networks
Assignment 5
I pledge my honor that I have abided by the Stevens Honor System

Approach:

I wanted to start working on "epinions96.csv" because that was the smallest dataset. So first, I read in the file and looked at what we had to work with. I saw that each CSV had three identifiers. For each row, there was "reviewer, reviewee, trust". After trying to figure out how I would do this manually, I reread the assignment doc and realized that there was a useful library called networkx. Networkx is a python library that is useful for studying complex networks. So I decided to implement it.

```
G = nx.Graph()
with open(fileName) as csvFile:
    for row in csvFile:
        row = row.replace("\n", "") # get rid of new line breaks
        line = row.split(",") # split into an array
        G.add_edge(line[0], line[1], weight = line[2])
```

For each row, I added the reviewer, reviewee and the trust weights to a graph

I also wrote my own implementation of finding self loops since networkx didn't support a function.

```
def findSelfLoops(G):
    nodes_in_selfloops = []
    for i, v in G.edges():
        if i == v:
            nodes_in_selfloops.append(i)
    return len(nodes_in_selfloops)
```

Afterwards, calculating the necessary numbers were quite easy:

- Edges: Length of Graph's Edges
- SelfLoops: Call findSelfLoops(G)
- Total Edges: Number of edges - self loops
- Positive Edges: For each edge in graph, count edges that are positive
- Negative Edges: For each edge in graph, count edges that are negative
- Probability p: Number of positive edges / total edges
- Probability 1-p: 1 minus probability p
- Triads:
    - Generate a list of all cliques in Graphs call it cliq_list
    - For each item in cliq_list, find elements where the length is 3
- Now we want to get the weights and probabilities and put it in a table
    - Get weights from each edge.
        - Will look like this ('20', '50'), '1'

- We want to find all the triangles with 20 and 50
  - Will look like this (('5', '20'), '1'), (('5', '50'), '1'), (('20', '50'), '1')
- We want to now classify this triangle with the specific trust class
  - Will look like this (('5', '20'), '1'), (('5', '50'), '1'), (('20', '50'), '1'), **"TTT"**
- Table will now look like this

	trustCategory	edgeOne	trustOne	edgeTwo	trustTwo	edgeThree	TrustThree
0	TTT	(5, 20)	1	(5, 50)	1	(20, 50)	1
1	TTT	(5, 20)	1	(5, 52)	1	(20, 52)	1
2	TTT	(88, 87)	1	(88, 89)	1	(87, 89)	1
3	TTT	(88, 86)	1	(88, 89)	1	(86, 89)	1
4	TTT	(88, 86)	1	(88, 87)	1	(86, 87)	1
5	TTT	(25, 57)	1	(25, 21)	1	(57, 21)	1
6	TTT	(25, 35)	1	(25, 57)	1	(35, 57)	1
7	TTT	(79, 35)	1	(79, 57)	1	(35, 57)	1
8	TTT	(79, 25)	1	(79, 39)	1	(25, 39)	1
9	TTT	(79, 25)	1	(79, 57)	1	(25, 57)	1
10	TTT	(79, 25)	1	(79, 35)	1	(25, 35)	1
11	TTT	(20, 35)	1	(20, 57)	1	(35, 57)	1
12	TTT	(20, 25)	1	(20, 57)	1	(25, 57)	1
13	TTT	(20, 25)	1	(20, 35)	1	(25, 35)	1
14	TTT	(20, 79)	1	(20, 57)	1	(79, 57)	1
15	TTT	(20, 79)	1	(20, 35)	1	(79, 35)	1
16	TTT	(20, 79)	1	(20, 25)	1	(79, 25)	1
17	TTT	(20, 52)	1	(20, 57)	1	(52, 57)	1
18	TTT	(20, 50)	1	(20, 25)	1	(50, 25)	1
19	TTT	(86, 87)	1	(86, 89)	1	(87, 89)	1
20	TTD	(52, 25)	-1	(52, 57)	1	(25, 57)	1
21	TTD	(20, 52)	1	(20, 25)	1	(52, 25)	-1
22	TTD	(5, 20)	1	(5, 79)	-1	(20, 79)	1
23	TDD	(20, 23)	-1	(20, 25)	1	(23, 25)	-1
24	TDD	(20, 52)	1	(20, 23)	-1	(52, 23)	-1
25	TDD	(8, 6)	-1	(8, 7)	1	(6, 7)	-1
26	DDD	(52, 23)	-1	(52, 25)	-1	(23, 25)	-1

To run program:

- Use command “python3 main.py”

Actual Distribution

Type	Percent	Number
TTT	length of column ‘trustCategory’ where value is equal to “TTT” / the total number in actual distribution	length of column ‘trustCategory’ where value is equal to “TTT”
TTD	length of column ‘trustCategory’ where value is equal to “TTD” / the total number in actual distribution	length of column ‘trustCategory’ where value is equal to “TTD”
TDD	length of column ‘trustCategory’ where value is equal to “TDD” / the total number in actual distribution	length of column ‘trustCategory’ where value is equal to “TDD”
DDD	length of column ‘trustCategory’ where value is equal to “DDD” / the total number in actual distribution	length of column ‘trustCategory’ where value is equal to “DDD”

Expected Distribution

Type	Percent	Number
TTT	probability of p * probability of p * probability of p	eTTT_percent * aTotal_num
TTD	3 * probability of p* probability of p * probability of 1-p	eTTD_percent * aTotal_num
TDD	3 * probability of p* probability of 1-p * probability of 1-p	eTDD_percent * aTotal_num
DDD	probability of 1-p * probability of 1-p * probability 1-p	eDDD_percent * aTotal_num

Output of “epinions96.csv”

Run Time: 0.02 seconds

```
MacBook-Pro-7:a05 chloequinto$ python3 main.py
====================================
Enter the name of the file:
epinions96.csv
Attempting to read: epinions96.csv

Edges in network: 96
Self-Loops: 0
Edges used - TotEdges: 96
Trust Edges: 68 |  probability p: 0.71
Distrust Edges: 28 |  probability 1-p: 0.29
Triangles: 27

Expected Distribution
Type      percent    number
------  ---------  --------
TTT          35.5       9.6
TTD          43.9      11.9
TDD          18.1       4.9
DDD           2.5       0.7
Total         100      27.1


Actual Distribution
Type      percent    number
------  ---------  --------
TTT            74        20
TTD            11         3
TDD            11         3
DDD             4         1
Total         100        27



   trustCategory   edgeOne trustOne   edgeTwo trustTwo edgeThree TrustThree
0            TTT   (5, 20)        1   (5, 50)        1  (20, 50)          1
1            TTT   (5, 20)        1   (5, 52)        1  (20, 52)          1
2            TTT  (88, 87)        1  (88, 89)        1  (87, 89)          1
3            TTT  (88, 86)        1  (88, 89)        1  (86, 89)          1
4            TTT  (88, 86)        1  (88, 87)        1  (86, 87)          1
5            TTT  (25, 57)        1  (25, 21)        1  (57, 21)          1
6            TTT  (25, 35)        1  (25, 57)        1  (35, 57)          1
7            TTT  (79, 35)        1  (79, 57)        1  (35, 57)          1
8            TTT  (79, 25)        1  (79, 39)        1  (25, 39)          1
9            TTT  (79, 25)        1  (79, 57)        1  (25, 57)          1
10           TTT  (79, 25)        1  (79, 35)        1  (25, 35)          1
11           TTT  (20, 35)        1  (20, 57)        1  (35, 57)          1
12           TTT  (20, 25)        1  (20, 57)        1  (25, 57)          1
13           TTT  (20, 25)        1  (20, 35)        1  (25, 35)          1
14           TTT  (20, 79)        1  (20, 57)        1  (79, 57)          1
15           TTT  (20, 79)        1  (20, 35)        1  (79, 35)          1
16           TTT  (20, 79)        1  (20, 25)        1  (79, 25)          1
17           TTT  (20, 52)        1  (20, 57)        1  (52, 57)          1
18           TTT  (20, 50)        1  (20, 25)        1  (50, 25)          1
19           TTT  (86, 87)        1  (86, 89)        1  (87, 89)          1
20           TTD  (52, 25)       -1  (52, 57)        1  (25, 57)          1
21           TTD  (20, 52)        1  (20, 25)        1  (52, 25)         -1
22           TTD   (5, 20)        1   (5, 79)       -1  (20, 79)          1
23           TDD  (20, 23)       -1  (20, 25)        1  (23, 25)         -1
24           TDD  (20, 52)        1  (20, 23)       -1  (52, 23)         -1
25           TDD    (8, 6)       -1    (8, 7)        1    (6, 7)         -1
26           DDD  (52, 23)       -1  (52, 25)       -1  (23, 25)         -1


Total Time for Program to Run:  0.02
====================================
```

Output of “epinions_small.csv”

Run Time: 155 seconds

```
MacBook-Pro-7:a05 chloequinto$ python3 main.py
====================================
Enter the name of the file:
epinions_small.csv
Attempting to read: epinions_small.csv

Edges in network: 54928
Self-Loops: 73
Edges used - TotEdges: 54855
Trust Edges: 46282 |  probability p: 0.84
Distrust Edges: 8646 |  probability 1-p: 0.16
Triangles: 217996

Expected Distribution
Type      percent    number
------  ---------  --------
TTT          60.1  130929
TTD          33.4   72757.6
TDD           6.2   13477.2
DDD           0.4     832.1
Total       100.1  217996


Actual Distribution
Type      percent    number
------  ---------  --------
TTT            73    159876
TTD            15     33443
TDD            10     21292
DDD             2      3385
Total         100    217996



       trustCategory           edgeOne trustOne           edgeTwo trustTwo  \
0                TTT  (264472, 291725)        1  (264472, 205645)        1
1                TTT  (280252, 260251)        1  (280252, 261351)        1
2                TTT  (280252, 240308)        1  (280252, 231108)        1
3                TTT  (280252, 240308)        1  (280252, 223677)        1
4                TTT  (280252, 240308)        1  (280252, 236323)        1
...              ...                 ...     ...                ...     ...
217991           DDD  (255970, 255674)       -1  (255970, 288371)       -1
217992           DDD  (201495, 295491)       -1  (201495, 240913)       -1
217993           DDD  (216022, 201535)       -1  (216022, 200573)       -1
217994           DDD  (216022, 201535)       -1  (216022, 241572)       -1
217995           DDD  (216022, 256303)       -1  (216022, 249399)       -1

              edgeThree TrustThree
0       (291725, 205645)          1
1       (260251, 261351)          1
2       (240308, 231108)          1
3       (240308, 223677)          1
4       (240308, 236323)          1
...                  ...        ...
217991  (255674, 288371)         -1
217992  (295491, 240913)         -1
217993  (201535, 200573)         -1
217994  (201535, 241572)         -1
217995  (256303, 249399)         -1

[217996 rows x 7 columns]


Total Time for Program to Run:  155.89
====================================
```

Discussion:

Problems

- The most difficult part for this assignment was trying to figure out expected distributions. I had to reread the chapter to get a better understanding of triads and I eventually got it.
- It took too long to run epinions.csv. So I did not capture the output.

Why was the actual distribution and expected distribution different?

- I think the reason why we see such different distributions is because in expected distribution, we're calculating based on probability and random distributions whereas in actual distribution, we're just counting actual weight edges and actual triad types.