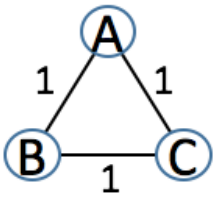
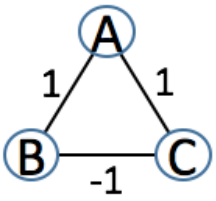
Assignment 05 – Graphs

For this assignment you will write a program to process graph data, and answer some questions about your results*.*

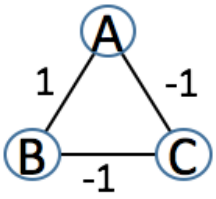
In chapter 5 of the text, we considered positive and negative relationships in networks, with triads of friends indicated by triangles of nodes. The diagrams to the right are similar, but in this case the edges represent trust. As with friendships, there are four possible relationships in a trust triad, represented by diagrams (a) through (d).



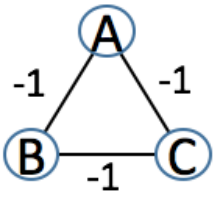
(a) A, B, C all trust each other



(b) A trusts B and C, but they don't trust each other



(c) A and B trust each other, but neither trust C



(d) None of them trust each other

For this assignment, you will study data from epinions.com that represents pairs of reviewers (the nodes) and whether they trust each other or not, represented by a 1 or -1 value for their connecting edge.

The dataset you will use comes from epinions.com, a consumer review site. Reviewers were asked whether they trusted or distrusted another reviewer. The dataset contains 841,372 entries*. Each has two numeric identifiers, representing the reviewer and reviewee, plus a 1 or -1; where 1 indicates the reviewer trusts the reviewee and -1 indicates that the reviewer distrusts the reviewee. The data can be used to create a graph that represents an undirected, signed network. The graph will have some self-loops (these are entries where the reviewer and reviewee are the same), but no duplicate or reciprocal pairs of nodes.

*Note: Depending on the software package and/or algorithm you use, you may not be able to process the **epinions.csv** file in a reasonable amount of time. [For example, when I used Python and networkx to retrieve all cliques and look at the ones of size 3 (the triangles), the program would not complete even if run overnight.] So, a smaller file, **epinions_small.csv** is also provided, which has fewer than 70,000 entries. You should try the larger **epinions.csv** file, but if that doesn't work, make note of that in your report and run your program on **epinions_small.csv** instead.

ASSIGNMENT:
Write a program to process the input file and identify triads. ***Your program must follow the programming guidelines and the specifications provided below.***

Also, write a one-page report documenting how you approached the problem, any problems you faced and any insights you gained. The report must include the following ***labeled*** sections: PURPOSE; INPUT; OUTPUT; WHAT THE PROGRAM DOES; ADDITIONAL INFORMATION. You will notice that the actual distribution of triad types differs from the expected distribution based on random assignment. Discuss briefly how they differ and why that might be so. The report should be submitted in .pdf or Word format, in a document ***labeled with your last name***, eg: Dugas_HW5_Report.pdf.

Zip your report, code, and a screen shot of ***the output of the runs of all files you successfully processed*** into a zip (compressed) file and submit in Canvas. *Do not submit the epinions files with your assignment. Screen shots may be included in your report, but also must be included separately in the zip file.*

PROGRAMMING GUIDELINES:
Programs will be screened for plagiarism. If you "borrow" code, be sure to document the details of the source; otherwise it will be considered plagiarism and result in a zero grade for the assignment. Borrowed code will not count toward your grade, only original code will be considered.

Programming can be done in a variety of languages. Programs should employ good programming practices. An example is the use of descriptive variable and function names.

Annotation and Comments: ***IMPORTANT***

- Program header must include **your** name and assignment information (use comments).
- Comments must also be used at the beginning of the program to give an overall description of the purpose of the program.
- Comments must also include detailed running instructions to run in a terminal window.
- Comments should also be used throughout the code to explain what it is doing. It should be possible to re-create your program based on the comments alone. Poorly commented programs will receive poor grades.

PROGRAM SPECIFICATIONS:

Your program should prompt the user to input a filename. This can be This can be epinions96.csv, epinions_small.csv, or epinions.csv. Do not use an argument list.

The files provided are epinions96.csv (96 entries), epinions_small.csv (66,062 entries), and epinions.csv (841,372 entries)

As you process the input file, identify and count self-loops, but do not include them in the later triad analysis. Do not eliminate the node itself -- you still want to consider the triads that it is a part of. If you are using a software package (such as networkx for Python) it may ignore self-loops when identifying triads. If you are writing your own routine, you should skip over self-loop entries after including them in your counts.

As you process the input file, count the number of positive and negative reviews. These counts will be used to create an expected distribution of the four triad types to compare to the actual distribution. In creating the expected distribution of triad types, assume that the positive and negative trust values are randomly assigned. [See example below]

Identify all of the triads in the graph. Do not just count them: You will need to know the value of the edges in the triangle formed by the three nodes in the triad. For each triad, identify which of the four triad types it represents, and add to the appropriate count.

Your output should contain the following:

1. Number of edges in the network
2. Number of self-loops
3. Number of edges used to identify triads (referred to as TotEdges) [this should be edges - self-loops.]
4. Number of positive (trust) edges (ignore self-loops)

5. Number of negative (distrust) edges (ignore self-loops)
6. Probability p that an edge will be positive: (number of positive edges) / TotEdges
7. Probability that an edge will be negative: $1 - p$
8. Expected distribution of triad types (based on p and $1 - p$ applied to the number of triangles in the graph). Show number and percent.
    a. Trust-Trust-Trust
    b. Trust-Trust-Distrust
    c. Trust- Distrust -Distrust
    d. Distrust- Distrust- Distrust
    e. Total
9. Actual distribution of triad types. Show number and percent.
    a. Trust-Trust-Trust
    b. Trust-Trust-Distrust
    c. Trust- Distrust -Distrust
    d. Distrust- Distrust- Distrust
    e. Total

------- EXAMPLE ------

This example was run on a much smaller file, epinions96.csv, which is provided on the Canvas site. The file contains 96 entries. You might use it as a practice file, to see that you get the same results, before running your program on the larger files.

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 *			Actual Distribution		
Type	percent	number	Type	percent	number
TTT	35.5	9.6	TTT	74	20
TTD	43.9	11.9	TTD	11	3
TDD	18.1	4.9	TDD	11	3
DDD	2.5	0.7	DDD	4	1
Total	100	27.1	Total	100	27

*remember to consider all edge combinations. e.g. for a triad with edges a, b, c with value t or d each, a triad of type TTD can represent t, t, d or t, d, t or d, t, t. So percent is 3 x .71 x .71 x .29 = 43.9%

The actual distribution differs quite a bit from the expected distribution that assumes trust/distrust is randomly distributed. There may be a number of reasons for that. In the assignment you are asked to think about that and share your thoughts on the reasons for the differences.

Here are the triads that were found, sorted by type:

TTT		5	20	1		5	50	1		20	50	1
TTT		5	20	1		5	52	1		20	52	1
TTT		20	50	1		20	25	1		50	25	1
TTT		20	52	1		20	57	1		52	57	1
TTT		20	79	1		20	25	1		79	25	1
TTT		20	79	1		20	35	1		79	35	1
TTT		20	79	1		20	57	1		79	57	1
TTT		20	25	1		20	35	1		25	35	1
TTT		20	25	1		20	57	1		25	57	1
TTT		20	35	1		20	57	1		35	57	1
TTT		79	25	1		79	35	1		25	35	1
TTT		79	25	1		79	57	1		25	57	1
TTT		79	25	1		79	39	1		25	39	1
TTT		79	35	1		79	57	1		35	57	1
TTT		25	35	1		25	57	1		35	57	1
TTT		25	57	1		25	21	1		57	21	1
TTT		88	86	1		88	87	1		86	87	1
TTT		88	86	1		88	89	1		86	89	1
TTT		88	87	1		88	89	1		87	89	1
TTT		86	87	1		86	89	1		87	89	1
TTD		5	20	1		5	79	−1		20	79	1
TTD		20	52	1		20	25	1		52	25	−1
TTD		52	25	−1		52	57	1		25	57	1
TDD		8	6	−1		8	7	1		6	7	−1
TDD		20	52	1		20	23	−1		52	23	−1
TDD		20	23	−1		20	25	1		23	25	−1
DDD		52	23	−1		52	25	−1		23	25	−1