APPM 3170 Fall 2021
Tehya Laughlin and Claire Ely
Final Project Part One

Problem 1

Approach:

The density formula was made into a function called density with two parameters: the number of nodes (n), and the number of edges (m). Then using a script, a 2D array for each location's two density values was made and filled by calling the density function. The number of nodes and edges were taken from Table 1 of the Project Documentation. The preview of the Density Array table was copied and inserted with the appropriate titles, see Table 1.

For the degree of each vertex, and the degree distribution of each graph, a MATLAB function already exists, taking in a graph G, and the "nodeID." By already calculating the degree at each node, filling an array with the degree at each node being the vertex allowed for that array to then be input into the histogram() function that MATLAB also has. Unfortunately, the histogram creation process wasn't expedited, so each time the Command Window was used, and each histogram modified manually before saving them as PNGs.

In reflection, instead of hard-coding the script to call the fill functions for every data set, an array of arrays/matrices could have been created, and a for loop used.

There is no function in MATLAB for the clustering coefficient of a graph node, so we used MATLAB's vectorization and the properties of an adjacency array in order to calculate the number of triangles for each network. When an adjacency matrix is raised to a power, the resultant matrix shows if there exists a path of length equal to the exponent between the column and row indices. The diagonal shows the number of closed paths (such that the start and end nodes are the same). Using the cubed adjacency matrix, the diagonal shows closed paths of length three, or the number of triangles present.

The number of triangles present and the previously calculated density values for each node were plugged into the clustering coefficient formula given by equation (5) in the Project Documentation. This calculation uses element-wise operations, such that it returns an array where the clustering coefficients are stored at an index equal to their respective node. To ensure no errors, we remove any nodes where the degree equals one so that the calculation of the clustering coefficient doesn't result in division by zero. Learning from past mistakes, a new script contained automated histogram creation lines.

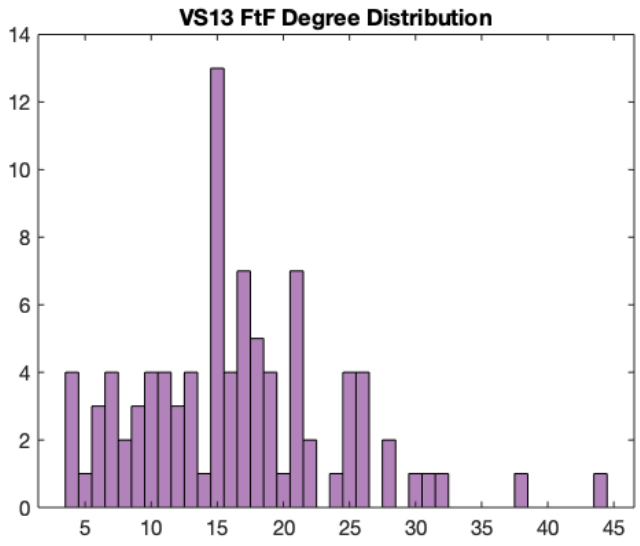
Results:

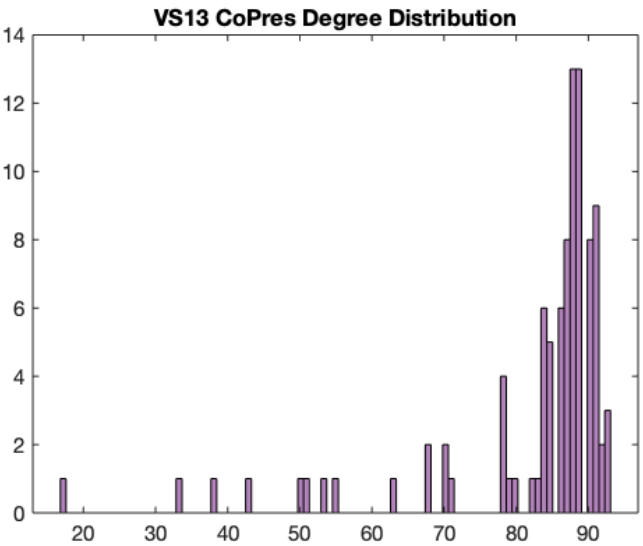
Table 1: Density for the face-to-face and co-presence data for each location

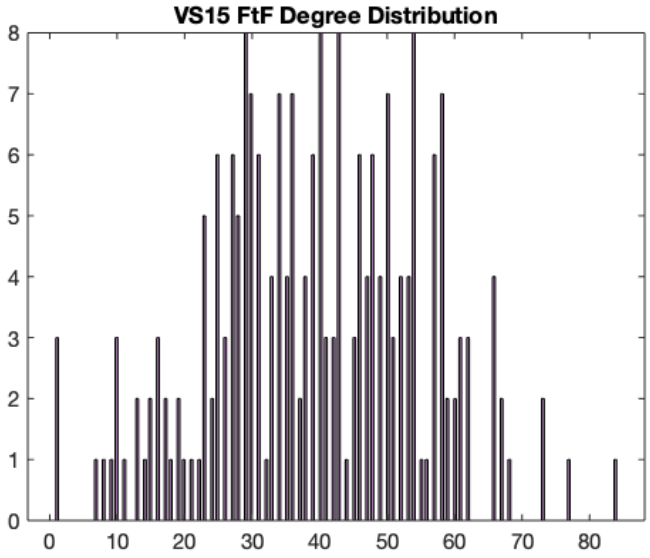
Location	Face-To-Face	Co-Presence
VS13	0.18036	0.65286
VS15	0.18237	0.70064
LH10	0.40561	0.52549
LyonSchool	0.28521	0.91197
SFHH	0.11808	0.90808
Thiers13	0.10915	0.81107

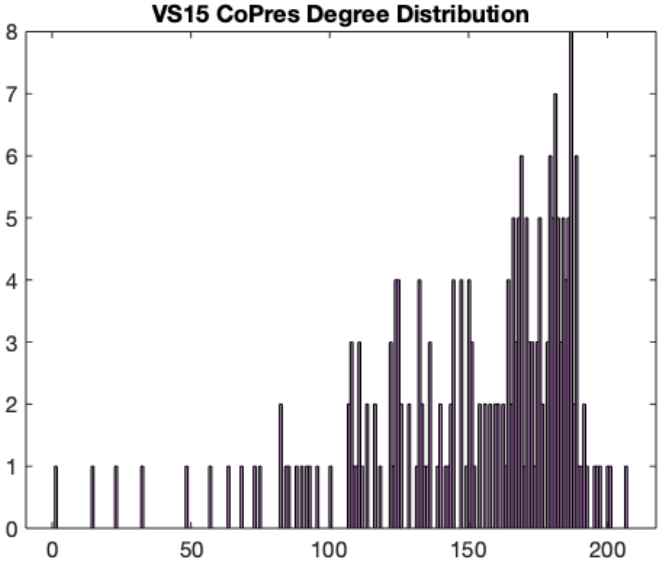
Graphs 1-12: Degree Distribution graphs titled “Degree Distribution”, and have purple coloring.

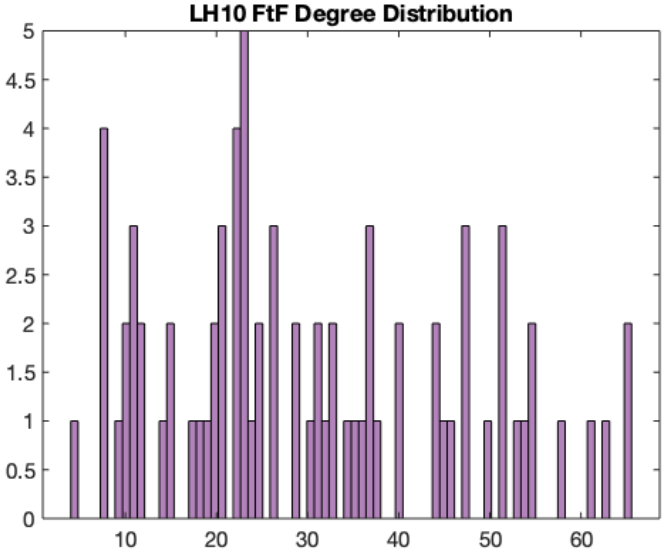
Graphs 13-24: Clustering Coefficient graphs are titled “CC”, and have green coloring.

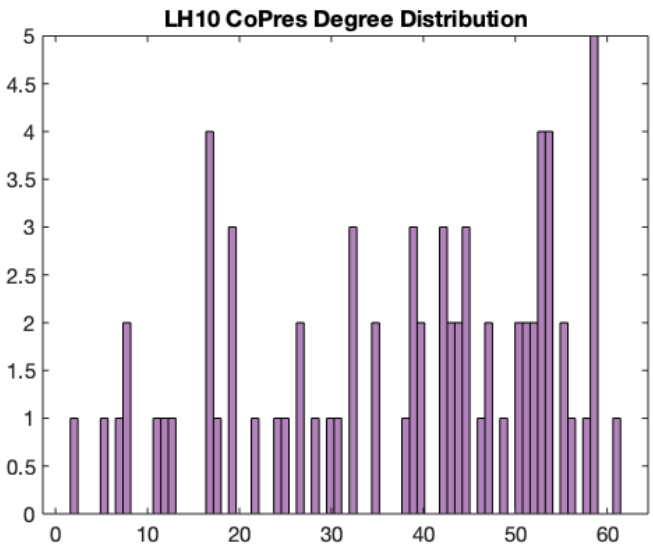

VS13 FtF Degree Distribution
0 2 4 6 8 10 12 14
5 10 15 20 25 30 35 40 45

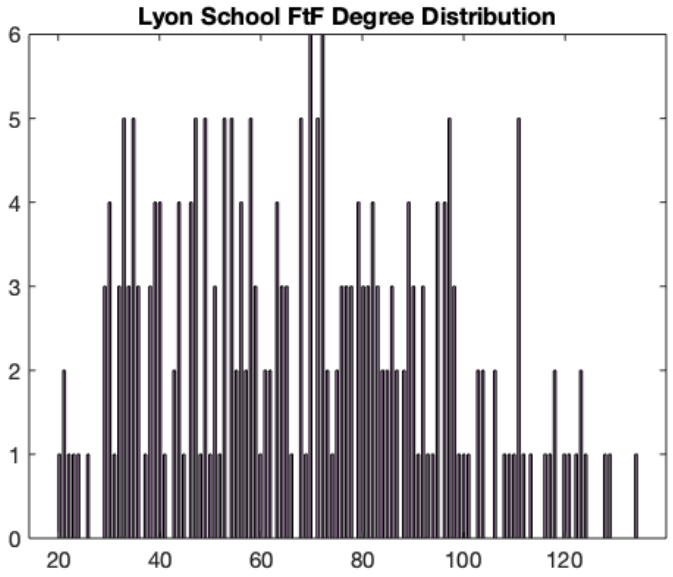

VS13 CoPres Degree Distribution
0 2 4 6 8 10 12 14
20 30 40 50 60 70 80 90

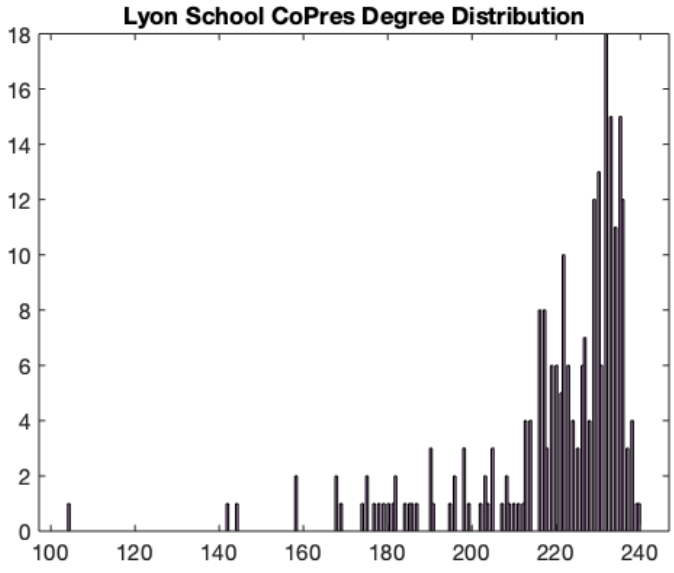

VS15 FtF Degree Distribution
0 1 2 3 4 5 6 7 8
0 10 20 30 40 50 60 70 80

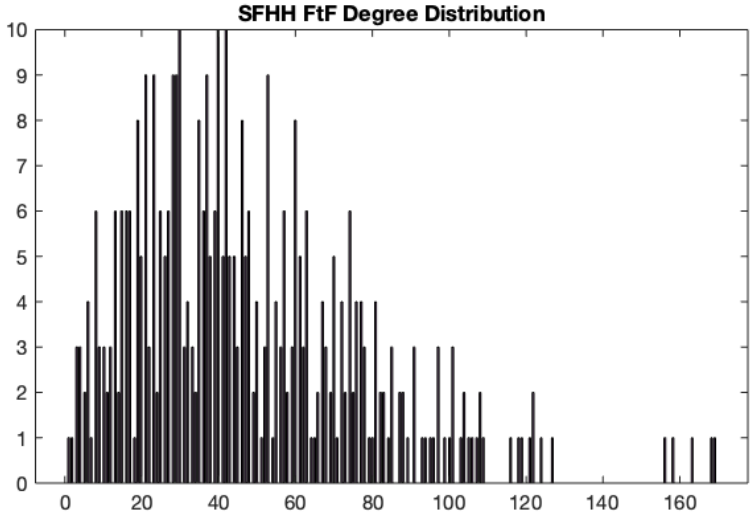

VS15 CoPres Degree Distribution
0 1 2 3 4 5 6 7 8
0 50 100 150 200

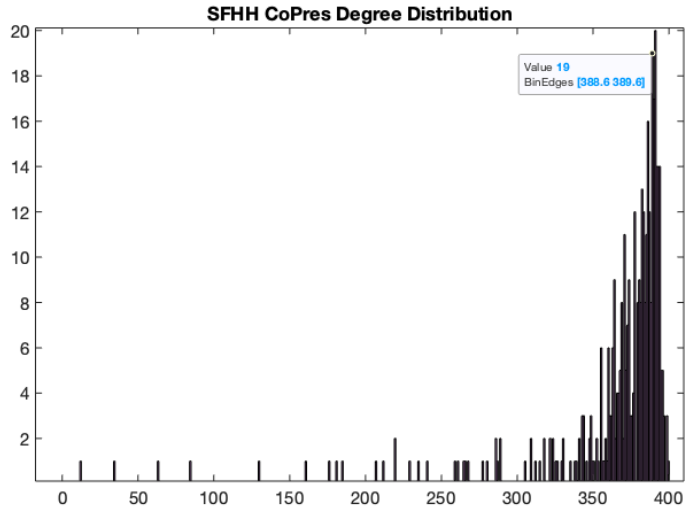

LH10 FtF Degree Distribution
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
10 20 30 40 50 60

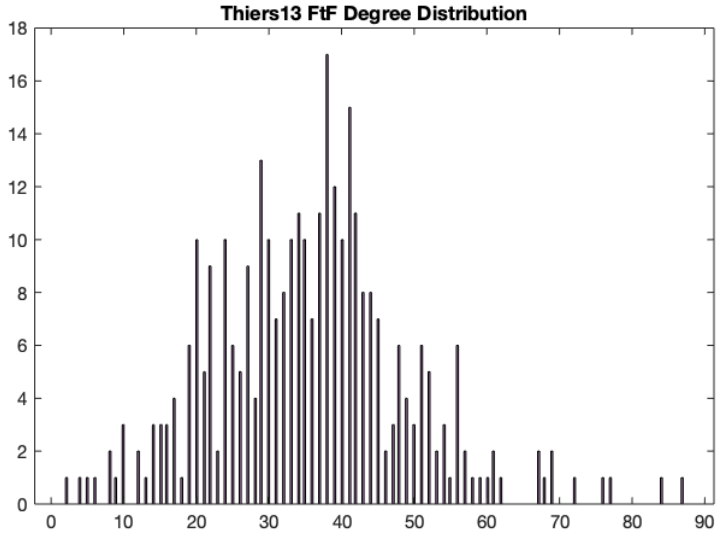

LH10 CoPres Degree Distribution
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
0 10 20 30 40 50 60

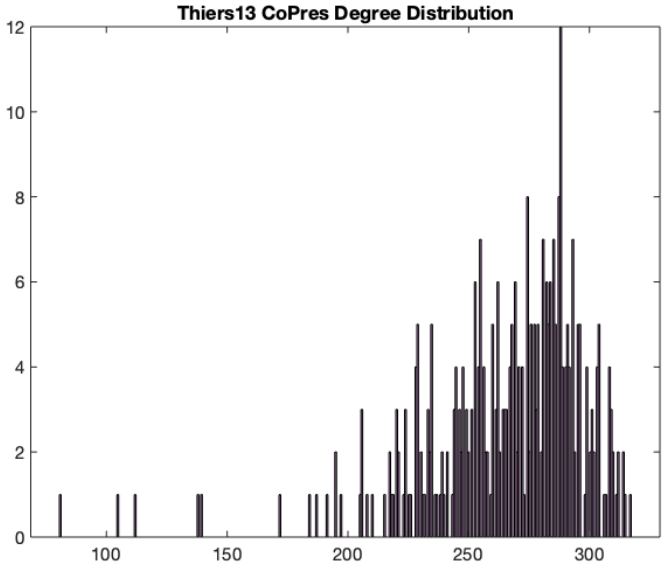

Lyon School FtF Degree Distribution
0
1
2
3
4
5
6
20
40
60
80
100
120

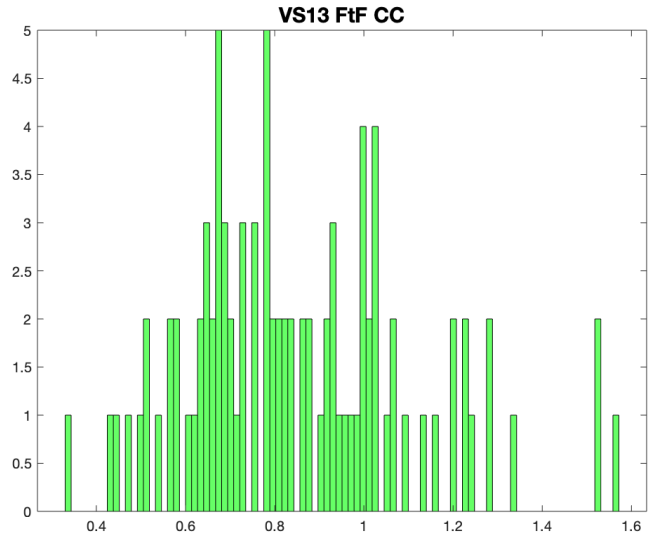

Lyon School CoPres Degree Distribution
0
2
4
6
8
10
12
14
16
18
100
120
140
160
180
200
220
240

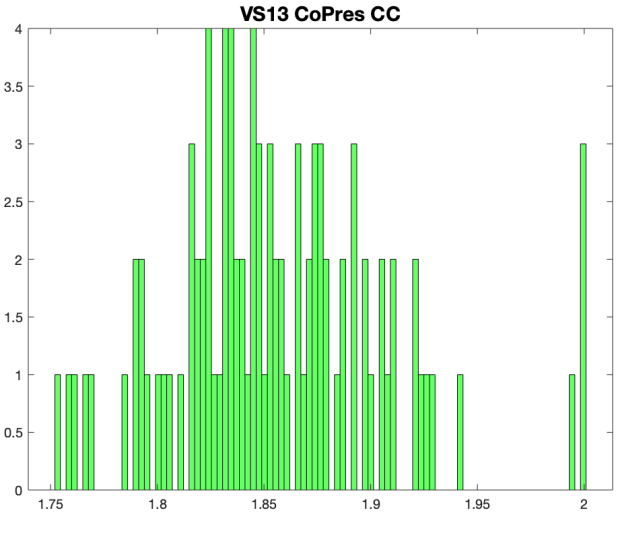

SFHH FtF Degree Distribution
0
1
2
3
4
5
6
7
8
9
10
0
20
40
60
80
100
120
140
160

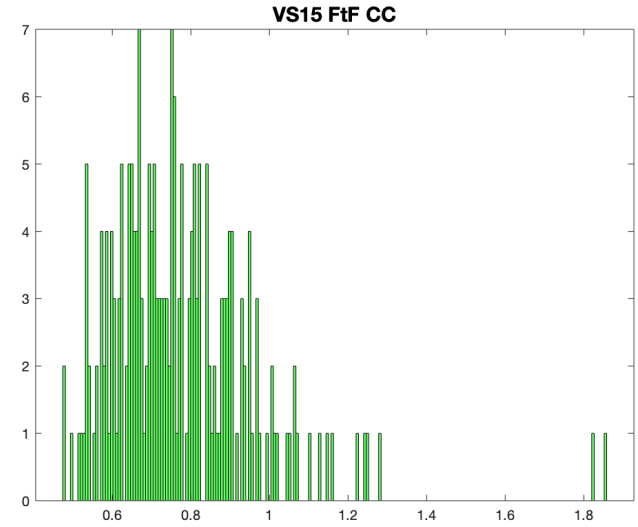

SFHH CoPres Degree Distribution
Value 19
BinEdges [388.6 389.6]
2
4
6
8
10
12
14
16
18
20
0
50
100
150
200
250
300
350
400

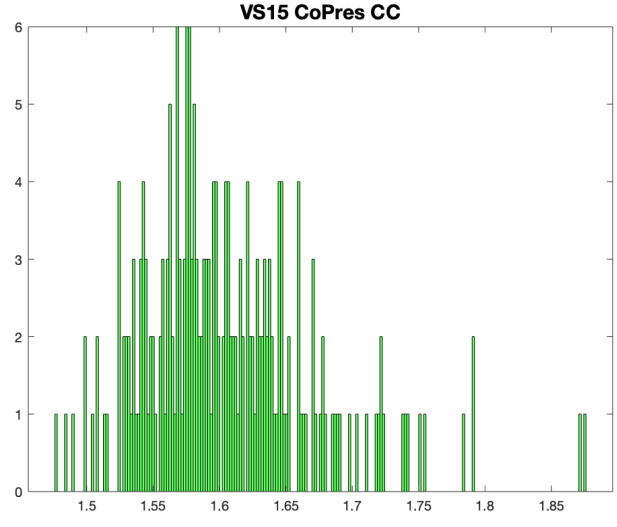

Thiers13 FtF Degree Distribution
0
2
4
6
8
10
12
14
16
18
0
10
20
30
40
50
60
70
80
90

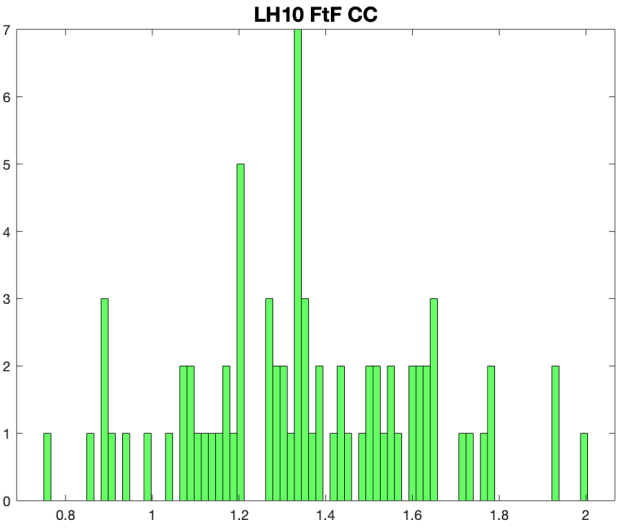

Thiers13 CoPres Degree Distribution
0
2
4
6
8
10
12
100
150
200
250
300

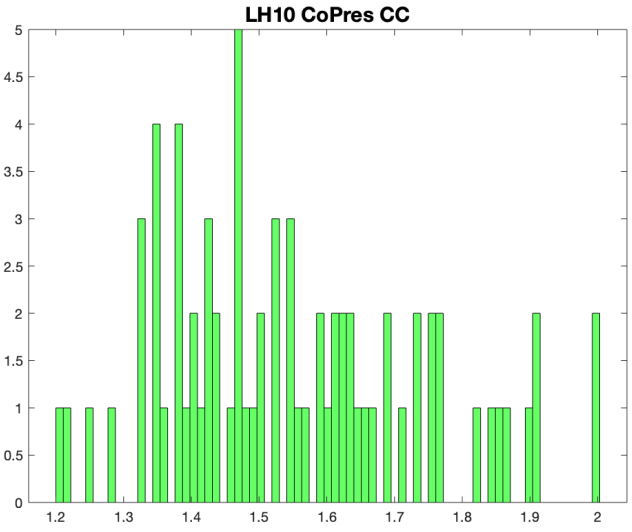

VS13 FtF CC

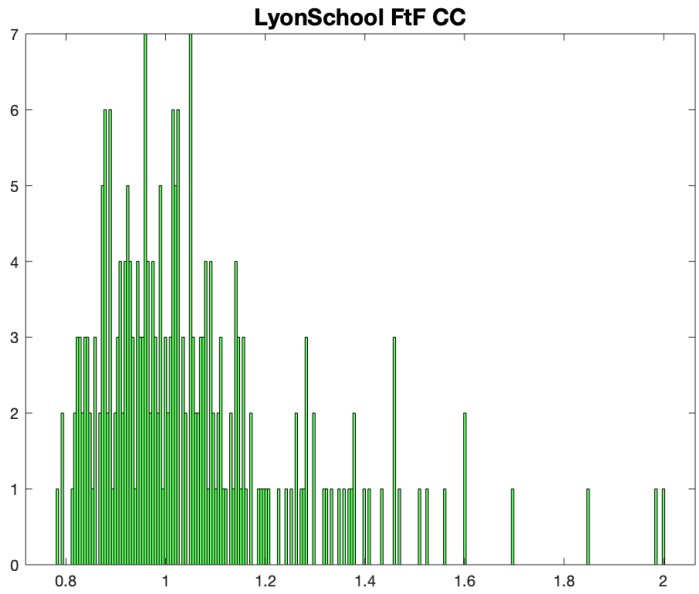

VS13 CoPres CC

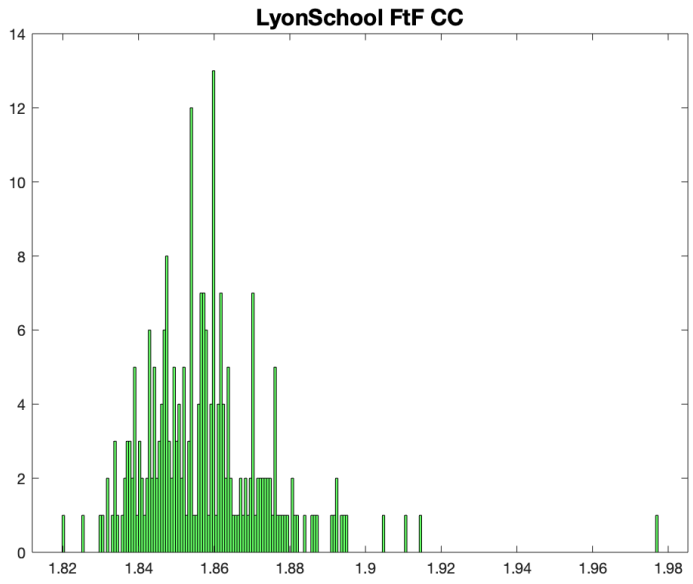

VS15 FtF CC

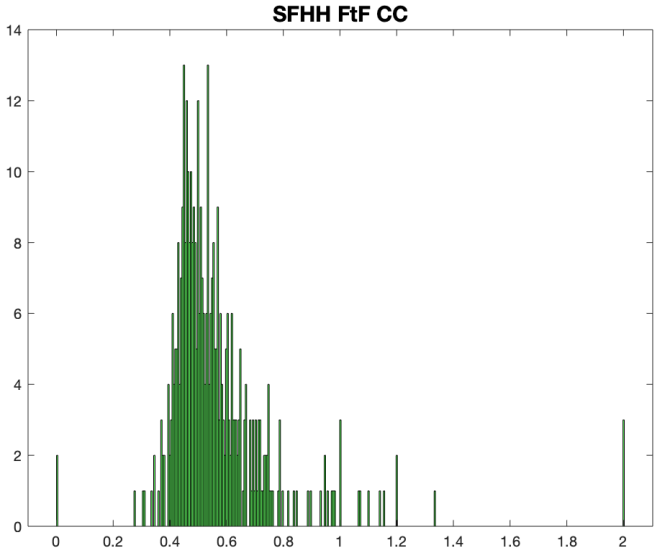

VS15 CoPres CC

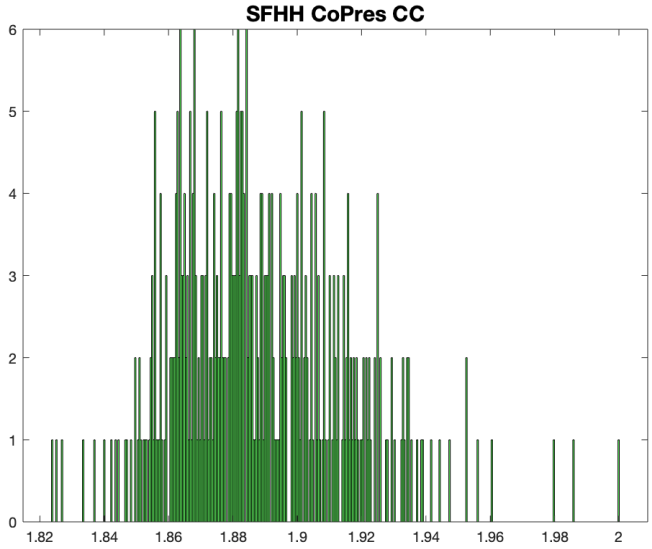

LH10 FtF CC

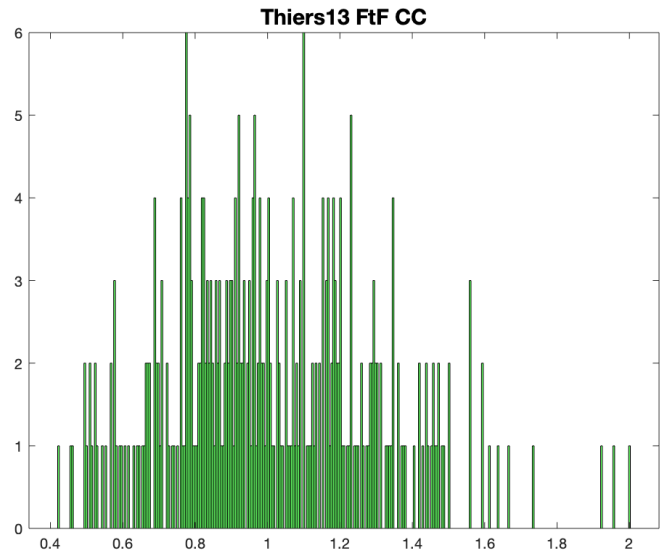

LH10 CoPres CC

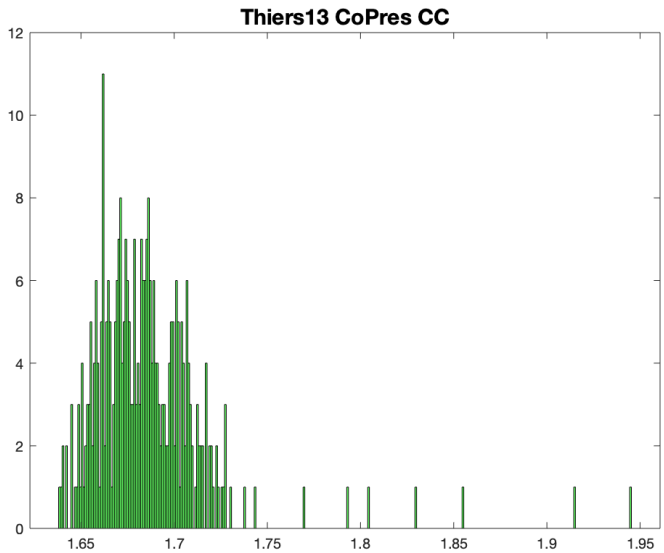

LyonSchool FtF CC
0 1 2 3 4 5 6 7
0.8 1 1.2 1.4 1.6 1.8 2


LyonSchool FtF CC
0 2 4 6 8 10 12 14
1.82 1.84 1.86 1.88 1.9 1.92 1.94 1.96 1.98


SFHH FtF CC
0 2 4 6 8 10 12 14
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2


SFHH CoPres CC
0 1 2 3 4 5 6
1.82 1.84 1.86 1.88 1.9 1.92 1.94 1.96 1.98 2


Thiers13 FtF CC
0 1 2 3 4 5 6
0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2


Thiers13 CoPres CC
0 2 4 6 8 10 12
1.65 1.7 1.75 1.8 1.85 1.9 1.95

Explanation:

Code:

A) Density

a) Density Function: Direct implementation of the given formula (3) from the Project Documentation.

```
function dens = density(a,b)
%  Calculates the density of the matrix by using the number
of nodes (n), and edges (m) --> Stored in _Arr_ arrays
dens = (2*a)/(b*(b-1))
end
```

b) Density Fill Script: Created extra unused columns, but otherwise worked.

```
% this script runs the density function and stores the
densities for each data set as a row for the same location

DensitiesArr = zeros(2,6);

% VS13
DensitiesArr(1,1) = density(VS13ArrF(2), VS13ArrF(1));
DensitiesArr(1,2) = density(VS13ArrP(2), VS13ArrP(1));

% VS15
DensitiesArr(2,1) = density(VS15ArrF(2), VS15ArrF(1));
DensitiesArr(2,2) = density(VS15ArrP(2), VS15ArrP(1));

% LH10
DensitiesArr(3,1) = density(LH10ArrF(2), LH10ArrF(1));
DensitiesArr(3,2) = density(LH10ArrP(2), LH10ArrP(1));

% LyonS
DensitiesArr(4,1) = density(LyonSArrF(2), LyonSArrF(1));
DensitiesArr(4,2) = density(LyonSArrP(2), LyonSArrP(1));

% SFHH
DensitiesArr(5,1) = density(SFHHArrF(2), SFHHArrF(1));
DensitiesArr(5,2) = density(SFHHArrP(2), SFHHArrP(1));
```

```
% Thiers13
DensitiesArr(6,1) = density(Thiers13ArrF(2),
Thiers13ArrF(1));
DensitiesArr(6,2) = density(Thiers13ArrP(2),
Thiers13ArrP(1));
```

B) Degree

a) Function to fill an array with the degrees at each node, uses MATLAB's degree function.

```
function Arr = fillDegrees(A, size)
% This function fills an array with the degree, where the
index is there
% vertex/node, using degree(G,nodeIDs)
    Arr = zeros(1,size,'double');
    G = graph(A);
    for i = 1: size
        Arr(i) = degree(G,i);
    end
end
```

b) Degree Fill Script: calls fillDegrees for every data set.

```
% This script makes an array for each location's 2 data
sets, and fills it
% with the degree for each v node, 1 to the size of the
matrix using the
% degree(G, nodeIDs) function which is in the fillDegrees
function

%VS13
VS13DegreeF = fillDegrees(InVS13,92);
VS13DegreeP = fillDegrees(InVS13_pres,95);

%VS15
VS15DegreeF = fillDegrees(InVS15,217);
VS15DegreeP = fillDegrees(InVS15_pres,219);

%LH10
LH10DegreeF = fillDegrees(LH10,76);
LH10DegreeP = fillDegrees(LH10_pres,73);

%LyonS
```

```
LyonSDegreeF = fillDegrees(LyonS,242);
LyonSDegreeP = fillDegrees(LyonS_pres,242);

%SFHH
SFHHDegreeF = fillDegrees(SFHH,403);
SFHHDegreeP = fillDegrees(SFHH_pres,403);

%Thiers13
Thiers13DegreeF = fillDegrees(Thiers13,327);
Thiers13DegreeP = fillDegrees(Thiers13_pres,328);
```

C) Clustering Coefficient

a) Function that fills the CC distribution array:

```
function cc = fillCCArr(A)

degr = sum(A,2);
nonzr = find(degr >1);

path3 = A * A * A ;

trngl = diag(path3);

cc = 2 .* trngl(nonzr) ./ (degr(nonzr) .* (degr(nonzr) -1 )
);

end
```

b) Clustering Coefficient Fill Script:

```
% makes each matrix a graph

% VS13
VS13CCArrF = fillCCArr(InVS13,92);
VS13CCArrP = fillCCArr(InVS13_pres,95);

% VS15
VS15CCArrF = fillCCArr(InVS15,217);
VS15CCArrP = fillCCArr(InVS15_pres,219);

% LH10
```

```
LH10CCArrF = fillCCArr(LH10,76);
LH10CCArrP = fillCCArr(LH10_pres,73);

% LyonS
LyonSCCArrF = fillCCArr(LyonS,242);
LyonSCCArrP = fillCCArr(LyonS_pres,242);

% SFHH
SFHHCCArrF = fillCCArr(SFHH,403);
SFHHCCArrP = fillCCArr(SFHH_pres,403);

% Thiers13
Thiers13CCArrF = fillCCArr(Thiers13,327);
Thiers13CCArrP = fillCCArr(Thiers13_pres,328);
```

Problem 2

Density:

The calculated density value shows the ratio of the number of edges (how interconnected the nodes, or people, are) over the maximum possible edges. In general, the co-presence data has higher density values than the face-to-face data. All of the face-to-face density values were less than 0.5000, showing that less than half of the possible connections that could've occurred actually did. All of the co-presence density values were greater than 0.5000. This logically follows from the method of data collection (or what is considered an edge) as co-presence data has lower standards for what counts as an interaction. As such, all sets of co-presence data had relatively more interactions and a higher density value.

For specific data sets, almost all of them showed a similar difference in the density values for face-to-face interactions compared to co-presence interactions, with an average difference of 0.538. However, LH10 had the closest difference, with a FtF density value of 0.40561 and a Co-Pres density value of .52549. The greatest difference was for the SFHH location, with a density difference of 0.79.

For the application of density data, it is relevant to note the likelihood of transmission based on the type of interaction. As co-presence is less likely to lead to transmission relative to face-to-face interactions, using the high density value from the co-presence data as a basis for high rates of disease spread is an ineffective measure. As such, any disease modeling that relies on density data would have to specifically control for the type of interaction being measured.

Degree:

Degree represents the connectivity of the nodes in the graph, as the degree of a given node is the number of neighbors the distance of one edge away (the number of first-degree or direct connections). The degree distribution that is clustered towards one quarter to one half of the number of nodes shows that the nodes are less connected, as compared to the co-presence data where the degree distribution is on the upper half to upper quarter of the number of nodes. Some locations have a larger difference in average degree, whereas LH10 has only a difference of 8 between the face-to-face and co-presence average degree. With the 3 locations that had large amounts of nodes, SFHH, Lyon School, and Thiers13, the bell curve is more pronounced for both face-to-face and co-presence data. However, the difference between the average degree is more pronounced. For SFHH, a Hospital Hygiene conference, the average degree for the face-to-face data is 48, and for the co-presence, 365. Also, the co-presence data for every location had many more edges. This means that more people were connected, so the degree is also higher.

The average degree for face-to-face data of Thiers13 and VS15 are similar, despite their difference in number of nodes. The social network, or individual interactions of the people in both the High School (Thiers13) and the Health Institute are similar. However, the co-presence data diverges, and Thiers13 pulls ahead in terms of interactions, this makes sense because

students may be moving around throughout rooms multiple times a day, whereas people in the Health institute may be more stationary.

For the degree graphs, there's not only a visual shift in where the bell curve is, but also a shift in scale for the co-presence data. The bell curves for SFHH and Lyon School's co-presence data have a greater maximum. Their curves are compressed, and there's more nodes with higher degrees, rather than a slightly more spread out distribution that their face-to-face histograms show.

Clustering coefficient:

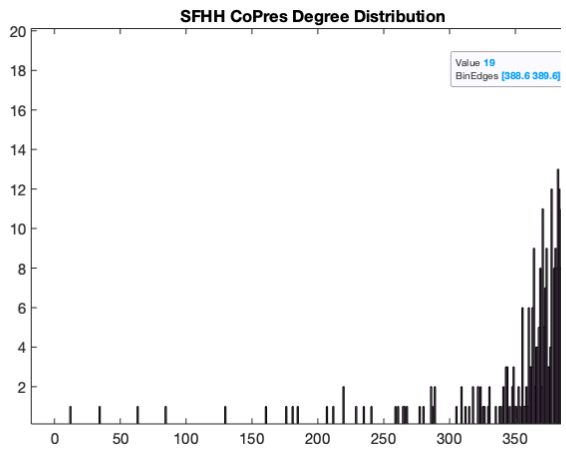
The clustering coefficient shows the proportion of the actual number of triangles a given node is a part of to the possible number of triangles. The concept of triangles represents the transitivity of a network, and can be used to show the levels of interconnections (how many connections or edges share the same connection). This is important in characterizing social networks or spread of disease in a social environment, as it shows how "clustered" a set of interactions is.

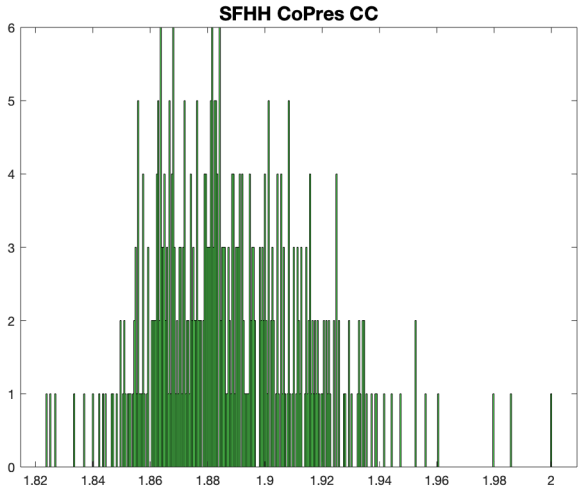
Notably, the most common clustering coefficient for the face-to-face data is zero, with the highest frequency out of any c value. This means that the most common clustering coefficient for a node in any face-to-face data set was zero, such that the node forms no triangles with two other nodes. For face-to-face interactions, we see that it is most likely that "the friend of your friend" is *not* your friend, or that you are unlikely to interact face-to-face with someone connected to your neighbor.

Looking at the equation to calculate the clustering coefficient:

$$cc(v) = \frac{2|\Delta(v)|}{deg(v)(deg(v)-1)}$$

We see that the value will be inversely proportional to the degree at the given node. Let us consider a data set where the degree is skewed to the right, for example the SFHH Co-Presence data. When compared to the SFHH Co-Presence clustering coefficient data, we see the inverse proportion as the c value is skewed to the left.





The same trend follows for other skewed data, such as the VS13 Co-Presence data set. Much of the co-presence degree is skewed to the right, as such the clustering coefficient is generally skewed left. The degree and clustering coefficient graphs take a similar shape, showing an inverse relationship. For clustering coefficient graphs that have more random distribution, the corresponding degree distribution is also fairly random. The face-to-face data is more evenly distributed (without taking into consideration the peak at a coefficient of zero). The zero value for the clustering coefficient is irrelevant of degree, as it moots any effect of the denominator.

While the relationship between the two distributions is fairly clear, it is also imperfect to reduce the trends in the clustering coefficient data to just connect to the degree distribution. The clustering coefficient also depends on the transitivity, or the number of "friends of a friend." Transitivity differs based on the type of data collected. The co-presence data for clustering coefficients trends higher, as it counts an interaction as being in the same room, rather than having a face-to-face interaction. It is more likely that someone will interact, or graphically be a "neighbor" with, a "friend of a friend" if they are in the same room compared to interacting face-to-face. The analysis shows a higher average and a higher maximum clustering coefficient for the co-presence data compared to the face-to-face data.

Location	**Density**		**Average Degree**		**Average CC**	
	FtF	**Co-Pres**	**FtF**	**Co-Pres**	**FtF**	**Co-Pres**
VS13	0.18036	0.65286	16	82	0.852	1.856
VS15	0.18237	0.70064	39	153	0.773	1.605
LH10	0.40561	0.52549	30	38	1.357	1.552
LyonSchool	0.28521	0.91197	69	220	1.051	1.858
SFHH	0.11808	0.90808	48	365	0.565	1.888
Thiers13	0.10915	0.81107	35	265	1.007	1.685

Analysis by Location:

InVS13 and InVS15:

InVS13 and 15 are the French Institute for Public Health Surveillance (two years later). As such, the InVS15 data should closely reflect the InVS13 data absent any major differences in types of interactions or any inconsistencies in the data. The table above does show that the data between the two locations follows similar trends, even with a significantly greater sample size for InVS15. InCS13 had 92 FtF nodes, and 95 co-presence nodes, while InVS15 had 217 FtF and

219 co-presence. The differences between density, average degree, and average clustering coefficient between the two sets are pretty much the same given slight potential variation.

LH10:
LH10 is a hospital ward in Lyon, France. As discussed above, LH10 has the closest values of average density. This differs from the other networks as the density difference is 0.538 while the difference for LH10 is .120. This is also reflected in the average degrees of the network between the FtF and co-presence data, which is significantly closer than any of the other data sets. As LH10 is a hospital ward, it is likely that there is a similar number of face-to-face connections where there are connections by just being in the same room.

LyonSchool:
The Lyon School location is a primary school in Lyon, France. It had higher average clustering coefficients than the other locations, which follows by the social nature of classrooms (particularly elementary schools, which prioritize group work and interactions). The LyonSchool had the highest co-presence density, showing a high density of edges in the network (such that most students had interacted with almost all (about 90%) of the other students. Again, this follows from the school setting.

SFHH:
The SFHH location was the 2009 French Society for Hospital Hygiene Conference. The conference had high co-presence clustering, but low FtF clustering. This logically follows, as the conference likely had large communal gatherings in the same room, but there was minimal face-to-face interaction of a "closed loop," given the general meeting of new people at a conference. Similarly, there was a large difference between average degree and density (which shows a higher degree of connectivity for the co-presence data).

Thiers13:
Thiers13 data was collected from a high school in Marseilles, France. It had generally higher clustering values than other locations, which makes sense due to the social circles that would be present in a high school setting. However, the CC values for Thiers were lower than elementary school (Lyon School), which shows slightly more connectivity between younger students in a smaller school. It makes more sense that there are less connected, or less social students, in a high school setting, where it is often easier.