**CSCI 3907/6444: Big Data and Analytics**

**Spring 2022**

**Simplification of the Graph**:

We are not able to do deep analysis of the dataset since we don’t have enough data. The objective of this project is to get you familiar with R and with some of the graph functions.

If we had a richly annotated data set, then some of the following methods would allow an initial sampling of the data set contents to get a sense of the type of data. This would be combined with some of the other methods that are contained in the rubric.

The objective is to “know” you data set before you begin a deeper analysis.

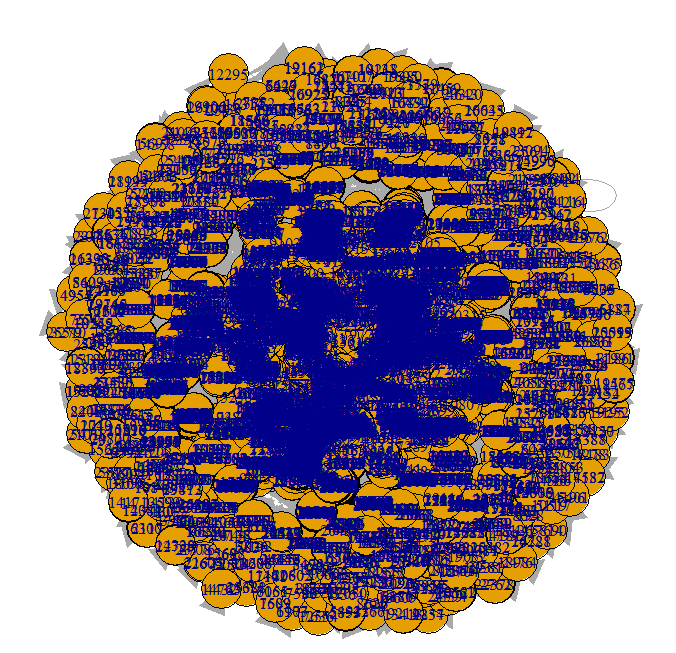
That said, these are suggestions for how to simplify the project data set graph and observe some of its structure.

There are several ways to simplify.

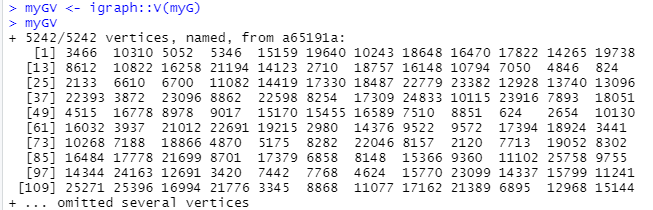
1) The most obvious is to removes nodes with a small number of edges.

So, once you have the Blue Blob or some complex graph, consider the graph structure.

Here is the Blue Blob:

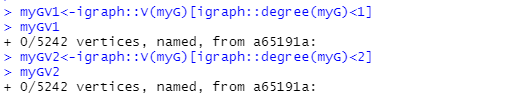


We find the vertex names using the following:



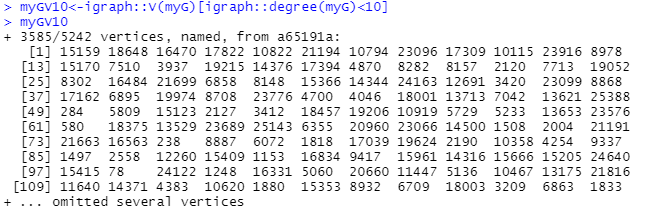
a. Remove all nodes with degree of 1, e.g., a node the connects to just one other node, Then, plot the graph again and also determine the number of nodes in the graph

If the visualization is still a blob/complex graph, then remove nodes with degree 2. Then, plot the graph again and also determine the number of nodes.



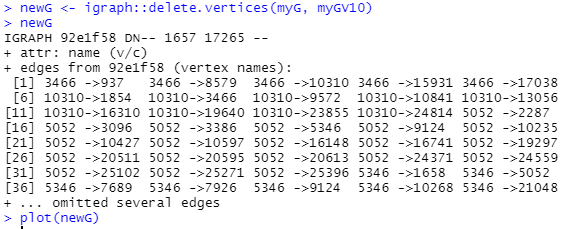
Well, it seems this data set does not have any vertices with degree 1 or 2.

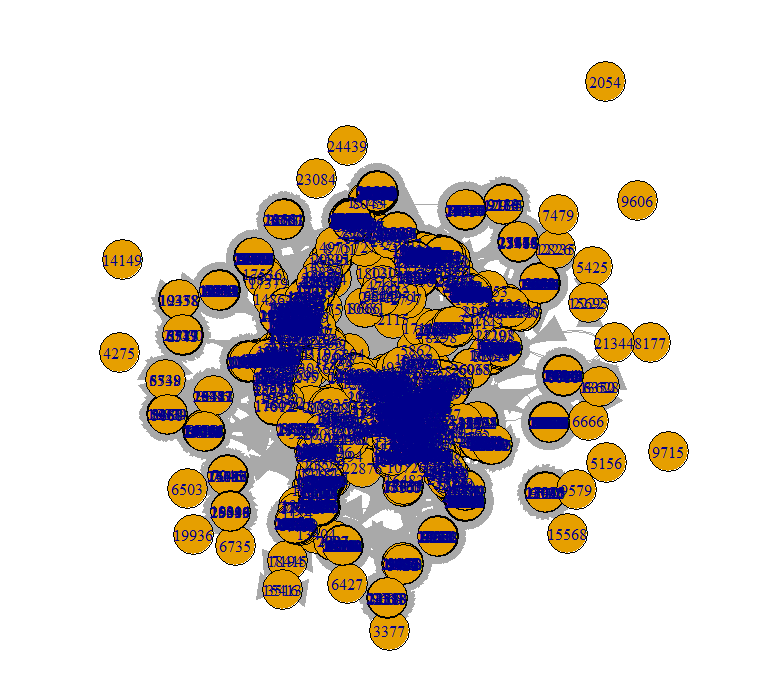
b. So, let’s try something larger, such as 10.



Well, 68.3% of the vertices have less than 10 edges.

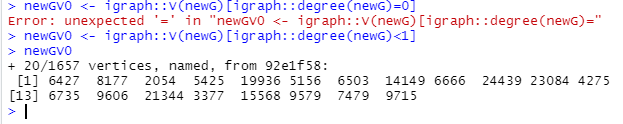
c. So, lets try to remove all vertices with less than 10 edges:





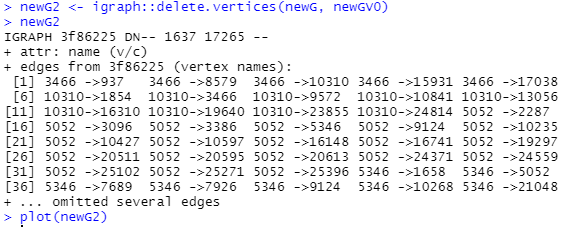
Note that we have a significant number of isolates in this graph.

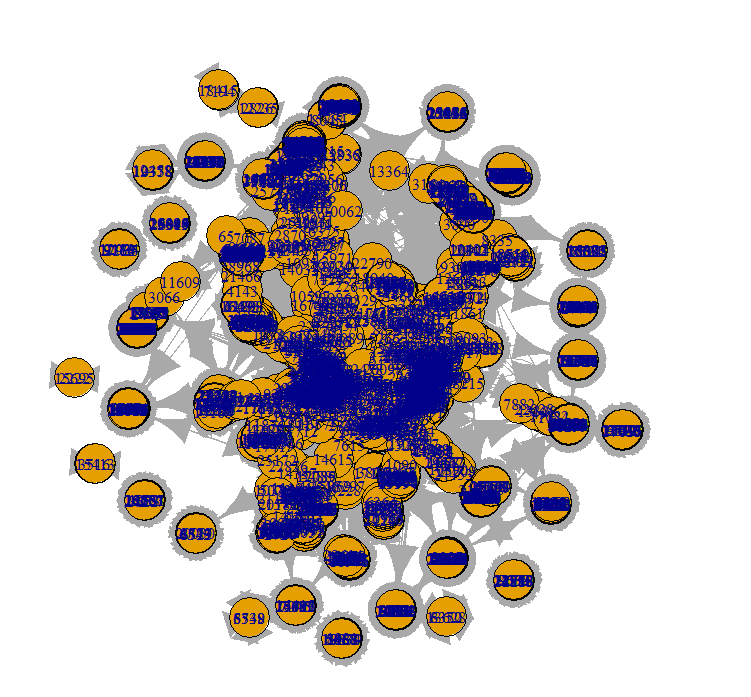
So, let’s also remove from this graph all vertices that have degree = 0.



Note: asking for degree = 0 does not work for some reason, but I can choose all vertices that have degree less than 1.

So, now let’s remove them:

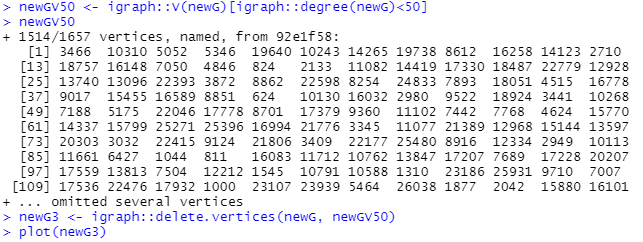


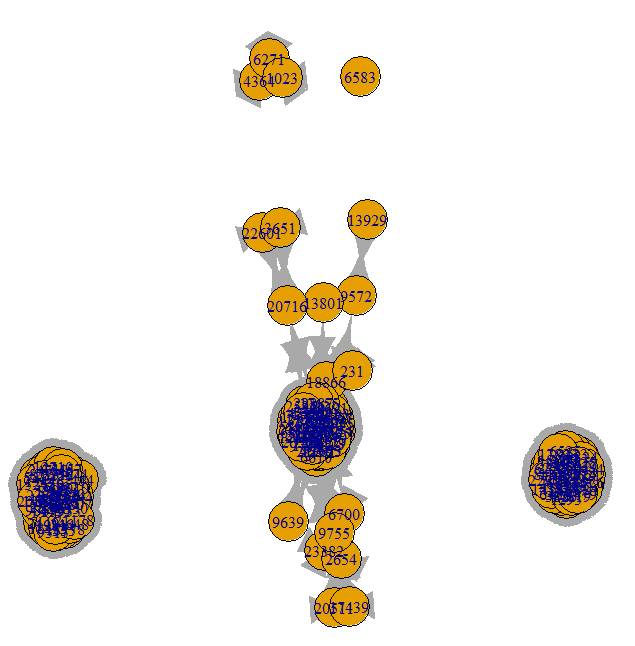


Well, didn’t help much, did it?

So, one last try just to see what we can get. Find all vertices with degree less than 50. We can work with newG because it has only vertices with degree .= 10.

I tried to eliminate nodes with less than and less than 30, but it didn’t make much difference. So, I took a Draconian approach and chose 50.

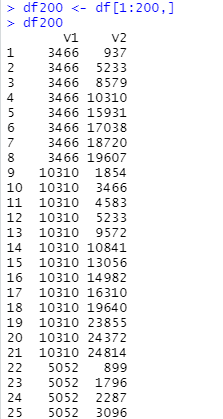




So, you should probably try 60 and 70 to see what happens.

This is a brute force approach, but it does allow you to see some of the internal structure. By internal structure we mean the nodes with the most degrees. Usually, these nodes will be interconnected at the “core” of the graph, but they may also represent nodes that are the cores of clusters.

2) Consider the first 200 records from the original data set and see what that presents.







What you see is three clusters

From here you can use the techniques identified above.

Alternatively, you can randomly select 300 notes, but that is likely to result in many more isolates.

What do you discern about the graph.

For very large graphs, e.g., those with tens of thousands of nodes or more, you might try to use 1% of the nodes – randomly selected and repeat this exercise.

Assuming that the nodes are not ordered, then random sampling may yield some impressions of the subgraphs of the graph.

3) Find the nodes bi with maximum betweenness centrality and nodes cj with maximum closeness centrality. There may be many such nodes in large graphs. There is NO guarantee that there is just one node with the maximum in either case or for other metrics as well.

Find their degrees, and plot each node xi (i =1, 10) with its edges connecting to nodes yij, and repeat for each of these latter nodes yij to their connecting nodes zijk. Is there any overlap?

To reiterate, if we had a richly labeled set of nodes and edges, we would be trying to get a “feel” for the attributes of nodes and edges, and for the occurrence of subgraphs.