**Network Analysis**

Network theory is the application of [graphs](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) as a representation of either [symmetric relations](https://en.wikipedia.org/wiki/Symmetric_relation) or [asymmetric relations](https://en.wikipedia.org/wiki/Directed_graph) between discrete objects. Network analysis is a set of techniques derived from network theory, which has expand from computer science to demonstrate the power of social network influences. It is an unsupervised learning technique. In network analysis we analyze strength of the node and strength of the connection. The visualization technique associates mapping relationships among entities based on the symmetry or asymmetry of their relative proximity.

There are two directions in network analysis:

* Uni-direction
* Bi-direction

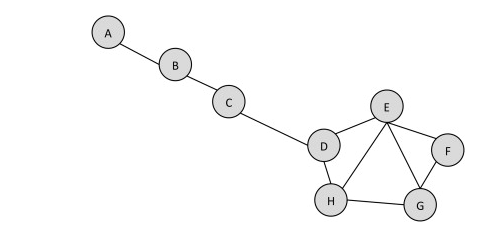
**Degree centrality:**

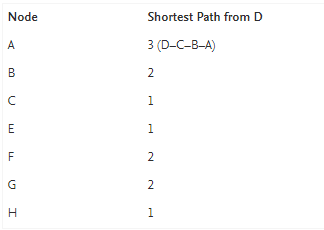
It is also called as local measure. Degree centrality is the simplest [centrality measure](https://www.sciencedirect.com/topics/computer-science/centrality-measure) to figure out. Recall that a node's degree is simply a count of how many social connections (i.e., edges) it has. The degree centrality for a node is commonly its degree. A node with 10 social connections would have degree centrality of 10. A node with 1 edge would have a degree centrality of 1.

**Closeness centrality:**

Closeness centrality is a way of detecting nodes that are able to spread information very efficiently through a graph. Closeness centrality means how close the node is to other nodes in the network. Nodes with a high closeness score have the shortest distances to all other nodes.

* For better value we can use “closeness centrality”
* Closeness = 1/(sum of distance to all other nodes)
* Normalized closeness = (total number of nodes – 1)\*closeness

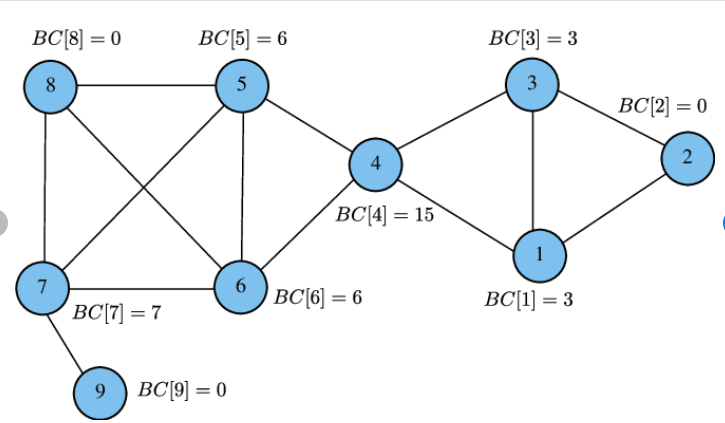
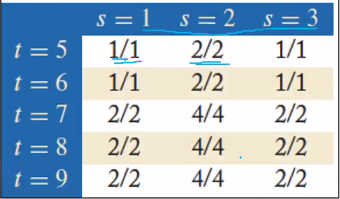


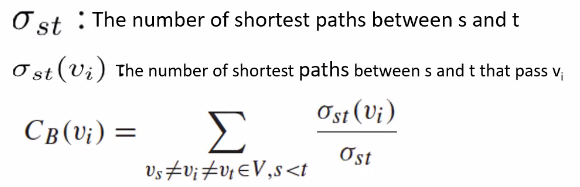


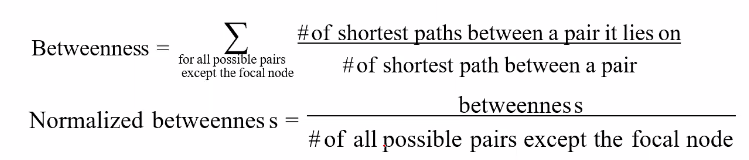
In the above figure, we measure the distance of node D to all its neighbors that can be represented in adjacency matrix.

**Betweenness centrality:**

Betweenness centrality is a measure of [centrality](https://en.wikipedia.org/wiki/Centrality) in a [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)) based on [shortest paths](https://en.wikipedia.org/wiki/Shortest_path_problem) (distance).  It perform the degree of which nodes stand between each other. It is how often it lies on the shortest path between pairs (or) how strong to the network.





* Number of all pairs except the focal node = (n-1)(n-2)/2

**R code:**

**Packages:**

install.packages("igraph")

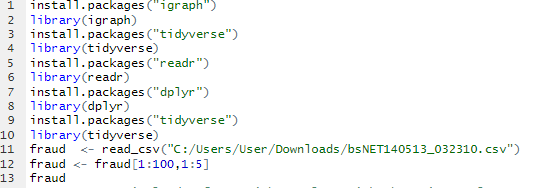
install.packages("tidyverse")

install.packages("dplyr")

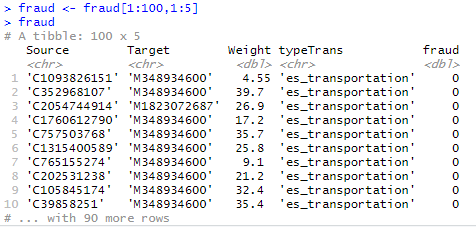
install.packages("tidyverse")

install.packages("network")

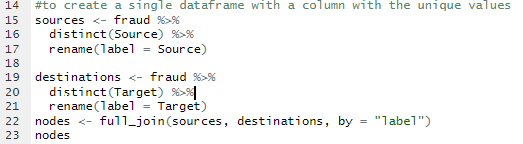
* to read the csv file using read.csv() function



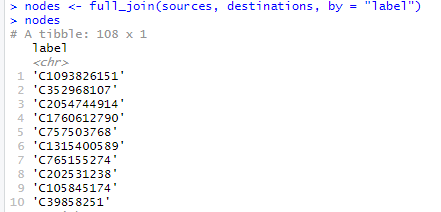
Output:



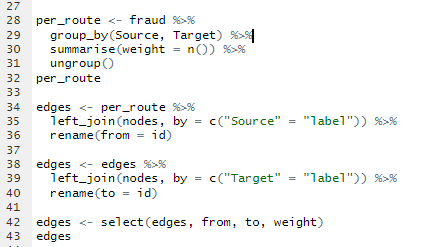
* Creating edge and node lists using dplyr package
* full\_join() to create a single dataframe with a column with the unique locations



Output:



* Creating an edge list is similar ,it is complicated by the need to deal with two ID columns instead of one.



## Creating network object using network() function

## To calculate degree of centrality using degree()

## To calculate closeness centrality using closeness()

## To calculate betweenness centrality using betweenness()

## 

**Python code:**

**Packages:**

import pandas as pd is used for data manipulation tasks.

import networkx as nxis used to create an empty graph with no nodes and no edges.

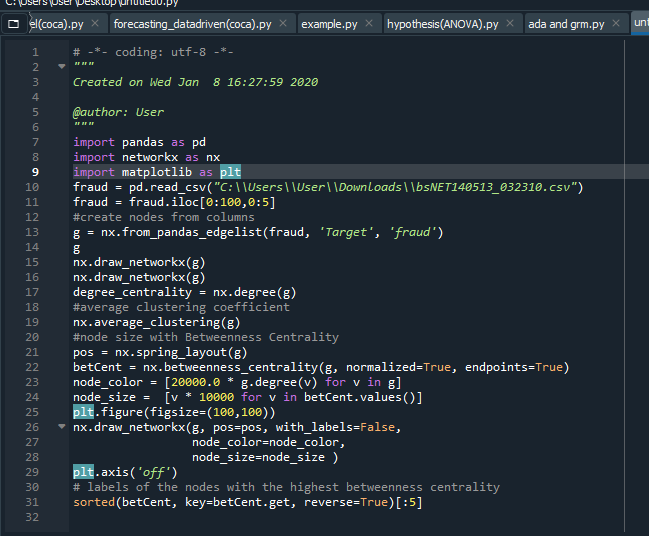
import matplotlib as plt is used for visualization.

Functions:

draw\_networkx() – is used to visualize the network

degree() – to find the degree of centrality

average\_clustering() – to calculate the average clustering coefficient



Output:

* labels of the nodes with the highest betweenness centrality

