**Assignment 4:** Your task in this week’s assignment is to identify an interesting set of network data that is available on the web (either through web scraping or web APIs) that could be used for analyzing and comparing centrality measures across nodes. As an additional constraint, there should be at least one categorical variable available for each node (such as “Male” or “Female”; “Republican”, “Democrat,” or “Undecided”, etc.). In addition to identifying your data source, you should create a high level plan that describes how you would load the data for analysis, and describe a hypothetical outcome that could be predicted from comparing degree centrality across categorical groups.

For Assignment 4, **my first choice** **is a** **movie database from IMDB’s** web site to analyze centrality measures across nodes. The data will be loaded using python and movie data will be extracted to create both nodes and edges.

One node type called **Movie** will contain title, year, length and average rating.

The second node type **Genre** will contain 7 genres (Action, Animation, Comedy, Drama, Documentary, Romance and Short). **This node is categorical**.

A binary field [0, 1] for each genre indicates an edge exist between Movie and Genre. I may add the **average rating as an edge property** and the **year the movie was released as an** **edge property**.

The program will calculate degree centrality, eigenvector centrality, between-ness centrality and possible closeness centrality.

Degree centrality counts the number of edges incident on a node (both in and out bound). It is a measure of the most popular genres.

Eigenvector centrality is a measure of how a node (movie or genre) importance increases based on how it is connected to other important nodes (movie or genre).

Between-ness centrality is a count of the number of times a node (movie or genre) acts as a bridge the shortest path between two nodes. It is a way to find a nodes importance due based on its connectivity.

Degree centrality will be compared across genres. If we compare centralities using the T-Test or Z-Statistic and find a genre is significant then it might indicate that new movies made in this genre would be highly watched and therefore profitable. Degree Centrality compared across genre grouped by release year can also be tied in. Significant genre centrality may vary by year as well. **I would like to work with the first choice (movies/genre from the IMBD web site)**

The second choice may be **Amazon’s movie review** data. The data is from snap.stanform.edu/data/web-movies. The data will be loaded using python and movie review data will be extracted to create both nodes and edges.

The file consists of the following fields:

* product/productId: [asin](http://en.wikipedia.org/wiki/Amazon_Standard_Identification_Number), e.g. [amazon.com/dp/B00006HAXW](http://amazon.com/dp/B00006HAXW/)
* review/userId: id of the user, e.g. [A1RSDE90N6RSZF](http://www.amazon.com/gp/cdp/member-reviews/A1RSDE90N6RSZF)
* review/profileName: name of the user
* review/helpfulness: fraction of users who found the review helpful
* review/score: rating of the product
* review/time: time of the review (unix time)
* review/summary: review summary

User nodes will consist of (User id, User name, user helpfulness and review score) and Movie nodes will consist of (Movie ID, movie name). Edges will be User **reviews** Movie and Movie—**reviewed by** Users. Edges will be both inbound (Users->Movie) and outbound (Movie->Users).

In bound Degree Centrality measures popularity.

Eigenvector centrality is a measure of how a node’s importance increases based on how it is connected to other important nodes. Highly reviewed movies are connected to other highly reviewed movies become more important in the network.

Between-ness centrality is a count of the number of times a node (movie) acts as bridge to the Shortest Path between two nodes. It is a way to find a nodes importance due based on its connectivity.

I have not yet worked out the details on this graph yet.

Another choice may be **Amazon’s book review** data. The data will be loaded using python and Book review data will be extracted to create both nodes and edges.

The file consists of users (User id, name, book reviewed and rating) and the book reviewed (Book ID, Title, Price, and Category). The user and book will be nodes. The edges will be User [:Revied]🡪 Book and Book—{:Reviewed by]-> Users.

Degree centrality measures a book’s popularity

Eigenvector centrality is a measure of how a node (book) importance increases based on how it is connected to other important nodes (books).

Between-ness centrality is a count of the number of times a node (book) acts as a bridge the shortest path between two nodes. It is a way to find a nodes importance due based on its connectivity.

I have not yet worked out the details on this graph yet.