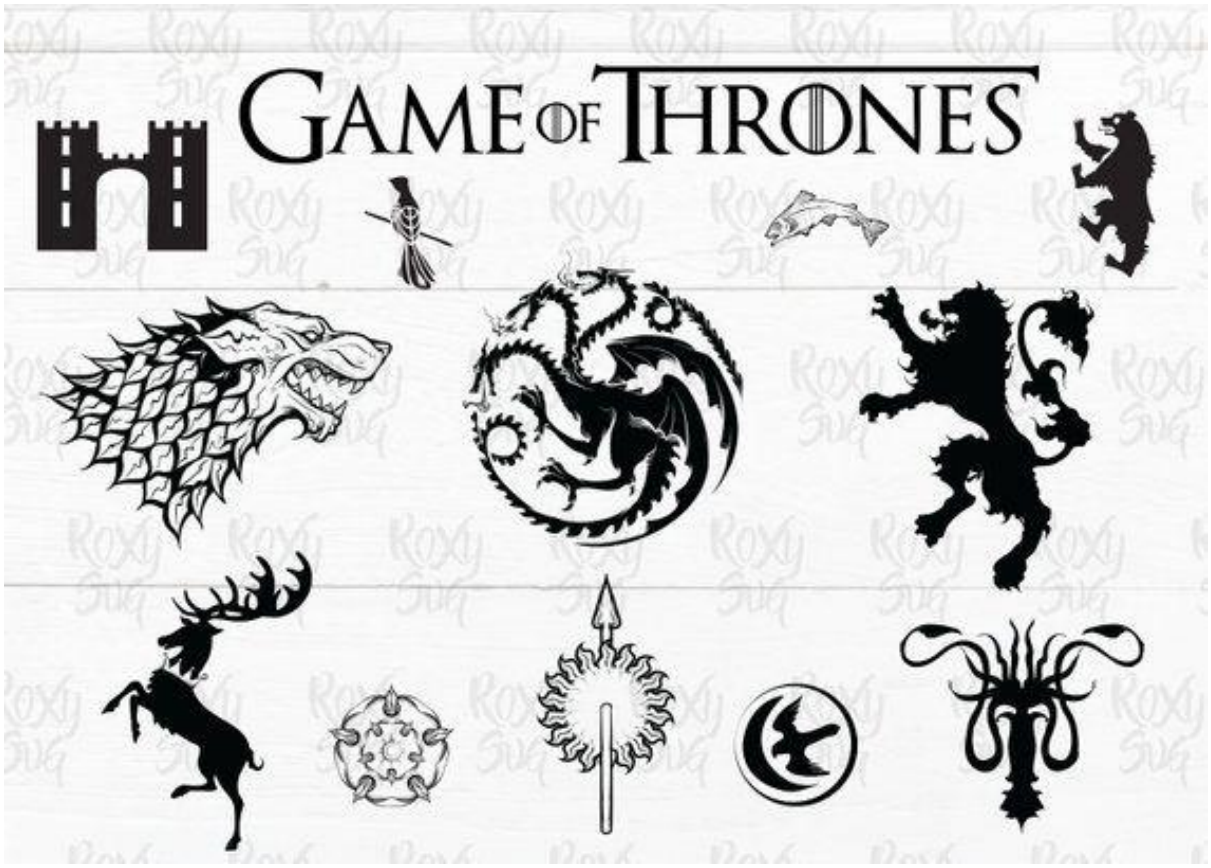
Network Analysis and Visualization with R and igraph



MSc in Business Analytics
Student: Vretteas Stylianos – p2822003
Course: Social Network Analysis
Homework 1: Network Analysis and Visualization with R and igraph

First of all, we are going to load the dataset. The dataset with Game of Thrones information consists of 28223 observations and five (5) columns named Source, Target, Type, id, weight. As per instructions we are going to keep only the Source, Target, and Weight columns and work with them in order to implement graph theory algorithms.

Task1 - `A Song of Ice and Fire' network

A graph is a way of specifying relationships among a collection of items. A graph consists of *nodes* and *edges*, these two objects are connected by *links.* We say that two nodes are *neighbors* if they are connected by a link. Regarding the direction of these links, we separate the graph into two major categories Directed (usually when we want to express symmetric relationships) and Undirected (when we want to emphasize that a graph is not directed). Generally, most of the graphs are undirected.

In our case, with the use of igraph library we build an undirected graph network called network1 and we are going to plot it. There is a lot of confusion because there are too many nodes and edges.

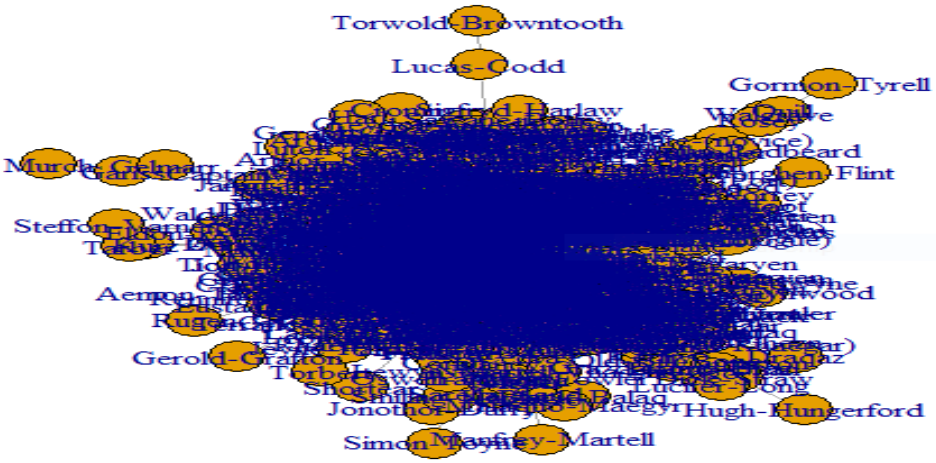


Figure 1 - Network Graph

Task2 - Network Properties

For task2 we are going to explore and print the Network and nodes descriptive measures. A concise explanation – definition will be given also.

Number of vertices: A vertex (plural vertices) or node is the fundamental unit of which graphs are formed: an undirected graph consists of a set of vertices and a set of edges (unordered pairs of vertices).

In our case we have 796 vertices or nodes.

Number of edges: An edge is the connection between two nodes. Each edge has two or more vertices to which it is attached, called its endpoints. In undirected graphs are called *links.*

In our case we have 2823 edges.

Diameter of the graph: Diameter of the graph is the maximum distance between pairs of vertices or length of the *shortest path* between two nodes.

In our case the diameter of the graph is 53.

Number of triangles: A cycle of length three in a graph, otherwise a triple connection of nodes is called a triangle.

In our case we have 1965 triangles.

The top-10 characters of the network as far as their degree is concerned:

Table 1 – Top 10 characters (degree is concerned)

Tyrion-Lannister	Jon-Snow	Jaime-Lannister	Cersei-Lannister	Stannis-Baratheon
122	114	101	97	89
Arya-Stark	Catelyn-Stark	Sansa-Stark	Eddard-Stark	Robb-Stark
84	75	75	74	74

A vertex can form an edge with all other vertices except by itself. So, the degree of a vertex will be up to the number of vertices in the graph minus 1.

We see as far their *degree* is concerned Tyrion Lannister is the top character.

The top-10 characters of the network as far as their weighted degree is concerned:

Table 2 - Top 10 characters (degree is not concerned)

```
  Tyrion-Lannister           Jon-Snow   Cersei-Lannister  Joffrey-Baratheon
              2873               2757               2232               1762
     Eddard-Stark Daenerys-Targaryen    Jaime-Lannister        Sansa-Stark
              1649               1608               1569               1547
       Bran-Stark   Robert-Baratheon
              1508               1488
> |
```

The *strength* of a graph is the minimum ratio of the number of edges removed from the graph to components created, over all possible removals.

We see as far their *weighted degree* is concerned Tyrion Lannister is the top character again.

The differences between the two outputs. Characters that are not included in each other measure.

Table 3 - Differences

```
> # differences
> degrees_top10[!(names(degrees_top10) %in% names(w_degrees_top10))]
Stannis-Baratheon        Arya-Stark     Catelyn-Stark        Robb-Stark
               89                84                75                74
> w_degrees_top10[!(names(w_degrees_top10) %in% names(degrees_top10))]
 Joffrey-Baratheon Daenerys-Targaryen         Bran-Stark   Robert-Baratheon
              1762               1608               1508               1488
> |
```

Task3 – Subgraph

For task3 we are going to create two visualization, one of the entire network and one of a subset of the whole network. In order to create the subgraph, exclude all vertices that have less than 10 connections in the network. Finally, we calculate the edge density of these two and give a short definition.

Plot the entire network:

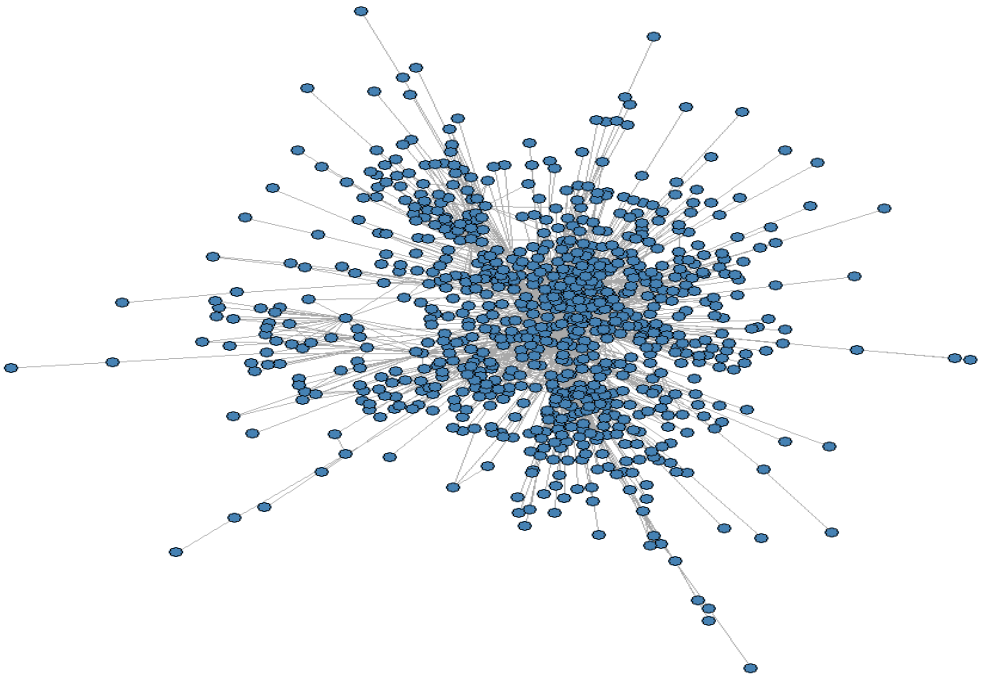


Figure 2 – Entire network

create a subgraph of the network:

In graph theory, an induced subgraph of a graph is another graph, formed from a subset of the vertices of the graph and all of the edges connecting pairs of vertices in that subset.

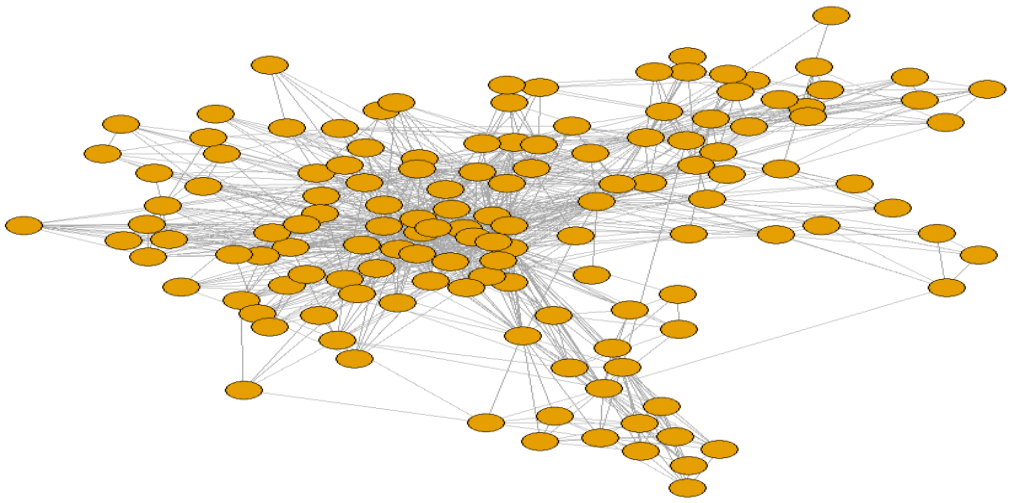


Figure 3 – Subgraph plot (vertices with less than ten (10) connections are excluded)

calculate the edge density of the entire graph:

The *density* of a graph is the ratio of the number of edges and the number of possible edges In our case the edge density value of the full graph is **0.0089** while the edge density of the subgraph is **0.11.** The subgraph has greater density, this difference is due to the fact that the subgraph is closer to its potential maximum edges than the entire graph which is fair reasonable since the subgraph is a subset of the entire graph.

Task4 – Centrality

Calculate the top-15 nodes according to closeness centrality and betweenness centrality:

In graph theory the concept of *Centrality* identifies the most important vertices within the network. The closeness centrality (or closeness) of a node is the average length of the shortest path between the node and all other nodes in the graph. The more central a node is in a network the closer will be with all the other nodes.
Below the top 15 characters based on **closeness** centrality.

Table 4 – Top 15 characters based on Closeness Centrality

```
> closeness_top15
  Jaime-Lannister  Robert-Baratheon     Theon-Greyjoy       Jory-Cassel Stannis-Baratheon
     0.0001193460      0.0001137527      0.0001135203      0.0001131734      0.0001131606
  Tywin-Lannister  Cersei-Lannister  Tyrion-Lannister  Brienne-of-Tarth          Jon-Snow
     0.0001128286      0.0001116695      0.0001114454      0.0001112718      0.0001106072
Joffrey-Baratheon     Rodrik-Cassel     Doran-Martell      Eddard-Stark       Harys-Swyft
     0.0001093733      0.0001083658      0.0001079098      0.0001073192      0.0001072961
```

On the other hand, the *betweenness* centrality determines the relative importance of a node by measuring the amount of traffic flowing through that node to other nodes in the network. This is based on the number of times a node acts like a bridge along the shorter path between two other nodes.
Below the top 15 characters based on betweenness centrality.

Table 5 - Top 15 characters based on Betweeness Centrality

```
> betweeness_top15
          Jon-Snow      Theon-Greyjoy    Jaime-Lannister Daenerys-Targaryen
          41698.94          38904.51          36856.35          29728.50
 Stannis-Baratheon   Robert-Baratheon   Tyrion-Lannister   Cersei-Lannister
          29325.18          29201.60          28917.83          24409.67
   Tywin-Lannister         Robb-Stark         Arya-Stark    Barristan-Selmy
          20067.94          19870.45          19354.54          17769.29
       Eddard-Stark       Sansa-Stark   Brienne-of-Tarth
          17555.36          15913.44          15614.41
>
```

We can see that regarding closeness Centrality Jon Snow is ranked 10th while based on betweenness Betweenness is 1st.

This is happening because Jon Snow's node in the network may not be in a central position compared to others but measuring the number of times that acts as a bridge between the other nodes is first. Specifically, this is true because as the story of Game of Thrones goes on Jon Snow interacts more and more with other characters.

Task5 - 5 Ranking and Visualization

PageRank is a webpage measure of importance, specifically is a numeric value that represents how important a page is on the web based at the number of times each paged is mentioned in the web and by who. Web pages are organized in a network, each webpage is represented as a node while each hyperlink is a directed edge, the entire web can be viewed as a directed graph.

We are going to implement the PageRank algorithm into our Game of Thrones network in order to calculate the importance of each character. Below are the top 10 characters based on the PageRank algorithm. We can see that Jon-Snow is first.

Table 6 - PageRank character ranking

```
> # get the first 10
> p_rank_df[1:10,]
             Character  Page_Rank
1            Jon-Snow 0.03570539
2    Tyrion-Lannister 0.03291094
3    Cersei-Lannister 0.02366461
4  Daenerys-Targaryen 0.02228040
5     Jaime-Lannister 0.01979001
6        Eddard-Stark 0.01896426
7          Arya-Stark 0.01857171
8   Stannis-Baratheon 0.01805099
9   Joffrey-Baratheon 0.01746037
10         Robb-Stark 0.01736071
>
```

Visualization

Below is a visualization of the network based on the Page_Rank metric, the bigger nodes are the nodes with the greatest Page_Rank value.

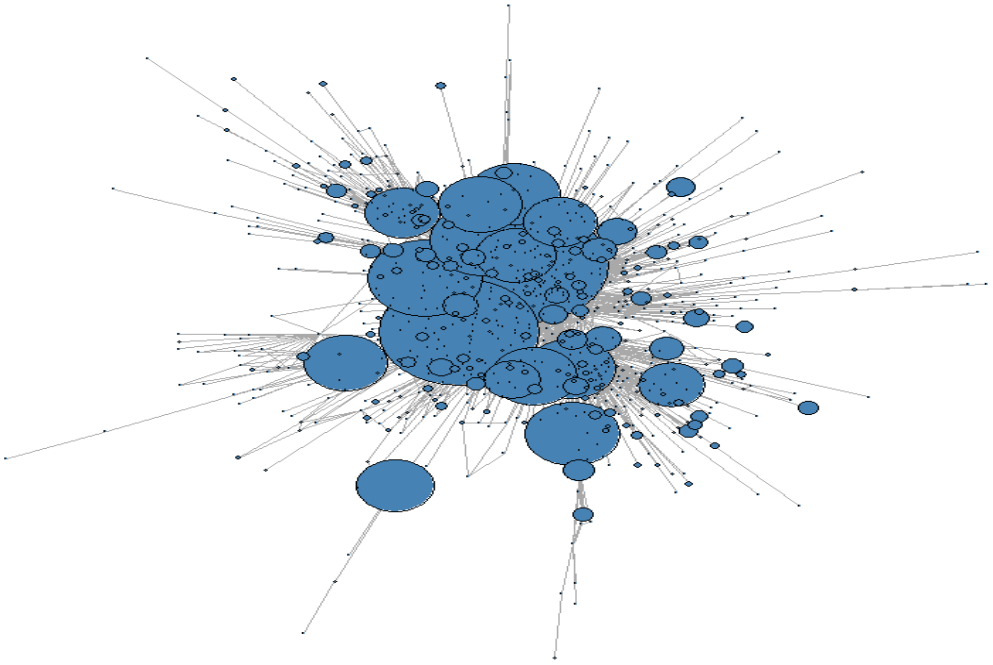


Figure 4 - Pagerank Visualization