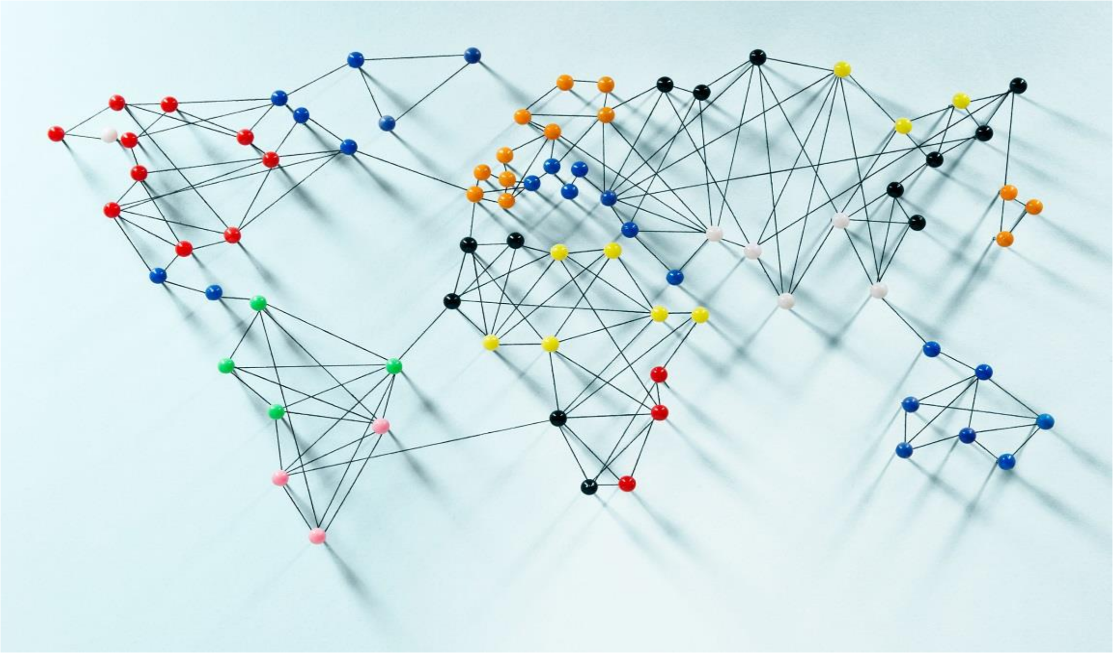
DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY

MASTER OF SCIENCE IN BUSINESS ANALYTICS



Course: Social Network Analysis

Instructor: Katia Papakonstantinopoulou

Homework 1: Network Analysis and Visualization with R and igraph

Student: Vogiatzis George

AM: P2821827

Question 1:

```
gotGraph <- graph_from_data_frame(edges, directed=FALSE, vertices=nodes)
```

Question 2:

Our goal is to obtain the graph properties.We start by calculating that our graph has 796 vertices and 2823 edges. In addition, the diameter of the graph path in the network, is 8. Another, property is the number of triangles that the graph has. In our graph was 5655.

Then we proceed with the top-10 characters of the network as far as their degree and their weighted degree are concerned. Code and results are provided below.

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

```
tail(sort(degree(gotGraph,loops = FALSE)),10)
```

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

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

```
tail(sort(strength(gotGraph, vids = V(gotGraph), mode = c("all"),
            loops = FALSE, weights = E(gotGraph)$weights)),10)
```

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

Question 3:

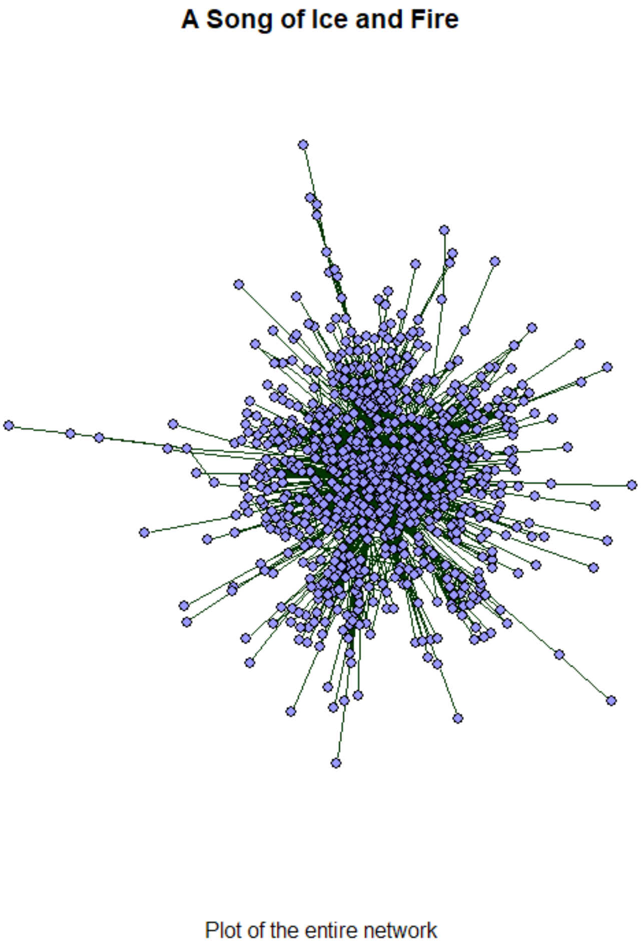


Figure 1.1: Plot of entire network

To continue we create the sub-graph which contains only the vertices with more than 10 connections. Code is provided below.

```
subGraph<-delete.vertices(gotGraph,

                        V(gotGraph)[ degree(gotGraph) <= 10  ] )


plot(subGraph,vertex.label = NA,vertex.color="#CC00CC",vertex.shape="circle",
edge.arrow.size = 0.5, edge.color="#A0A0A0", edge.width=0.4,vertex.size = 4, main = "A
Song of Ice and Fire",xlab="Plot for vertices with >=10 connections")
```

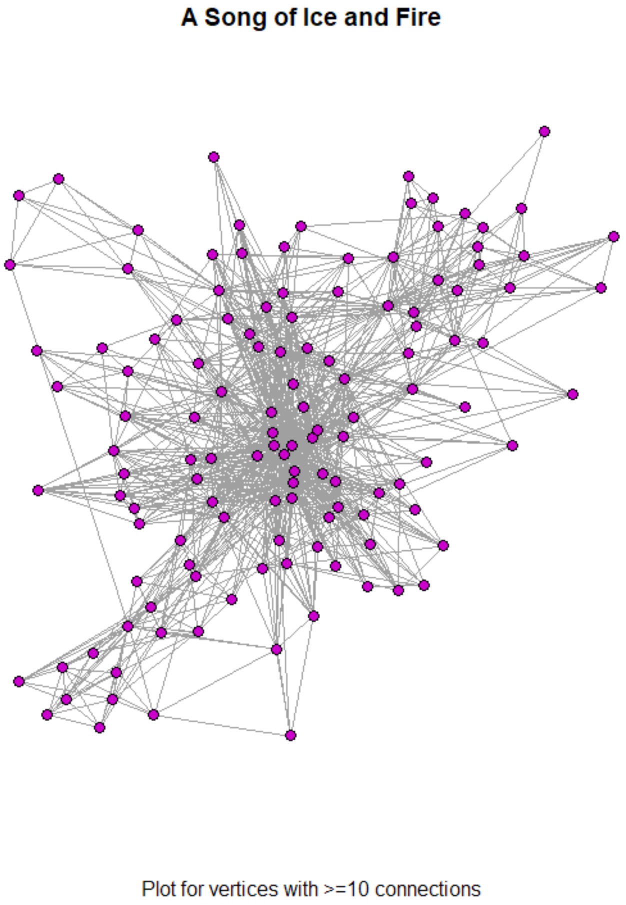


Figure 1.2: Subgraph plot.

We continue by calculating the edge density of the graph and the subgraph. Density is a measure of the proportion of possible ties, which are actualized among the members of a network. For our graph is 0.008921968 and for the subgraph is 0.1258612. It is normal for the subgraph to be more dense because it contains graphs with more than 10 connections.

We know that a complete linked network have density equal to one, therefore, we realize that the subgraph will have more connections than the original graph.

Graph Density: edge_density(gotGraph)

Subgraph Density: edge_density(subGraph)

Question 4:

Now we find the top 15 nodes according to closeness centrality and betweeness centrality. The calculations become with the below code:

Closeness centrality:

tail(sort(closeness(gotGraph, vids= V(gotGraph), mode = c("all"), weights = E(gotGraph)$weights, normalized = FALSE)),15)

Betweeness centrality:

tail(sort(betweenness(gotGraph, v = V(gotGraph), weights = E(gotGraph)$weights, normalized = FALSE)),15)

We observe that Jon Snow is in the first and tenth place according to betweeness and closeness centrality respectively. So we conclude that Jon Snow connection other nodes, can influence the spread of information through the network but it is not in the center of the graph so he cannot quickly transmit information and influence through direct or short paths to others.

Question 5:

We calculate the Page rank value for each node and we plot the graph. Code provided below:

rank<-page_rank(gotGraph, algo = "prpack", vids = V(gotGraph), directed = FALSE, damping = 0.85,personalized = NULL, weights = E(gotGraph)$weights)

```
plot(gotGraph,vertex.label =
NA,vertex.color="#00CCCC",edge.color="#FFCCFF",edge.arrow.width=0.9,vertex.size=rank$v
ector*1000, xlab = "Node size based on PageRank value",main="A Song of Ice and Fire
Network")
```

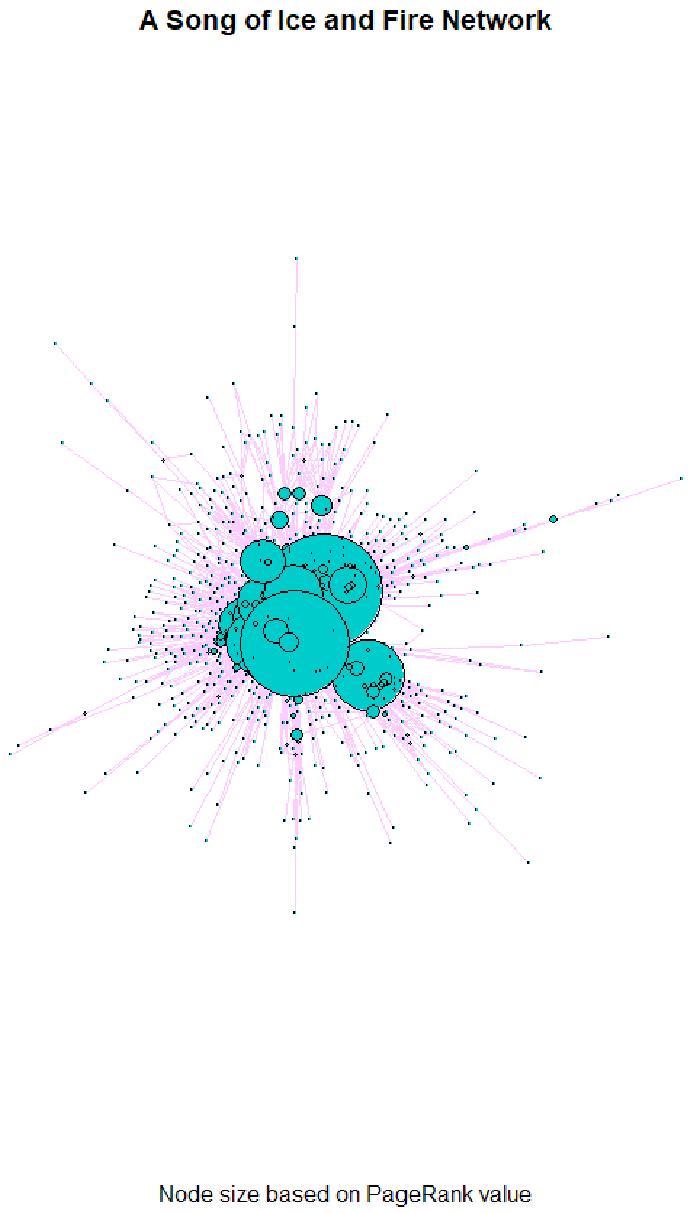


Figure 1-3: Graph plot showing the node size based on PageRank.