# Q4_Network_Analysis

MeadhbhHealy

04/05/2021

The two primary components of representing a network or graph structure are entities and the relationships between them. Entities can be referred to as the nodes or vertices of a graph, and the connections or relationships can be referred to as links or edges.A graph or network structure is a pictorial representation of the interconnections between a set of entities. Network Analysis may be described as the process of examining the attributes and connections of a graph structure in order to gain information about them. Although R was not designed specifically for network analysis, it has developed into an effective apparatus overtime. The reasons for this include; R enables reproducible research and has many analytical tools for manipulating data. In order to be represented in R, raw data needs to be converted into to a particular format. The object classes for the network analysis packages igraph and tidygraph are based on adjacency matrices or sociomatrices. An adjacency matrix is a square matrix in which the row and column names are entities of the graph.They effect a distinct data structure that cannot be manipulated by the tidyverse tools. However, the specialized network objects can be created from edgelists or nodelists.

```
quaker <- tbl_graph(nodes,edges,directed=FALSE,node_key = 'Name')
summary(quaker)
```

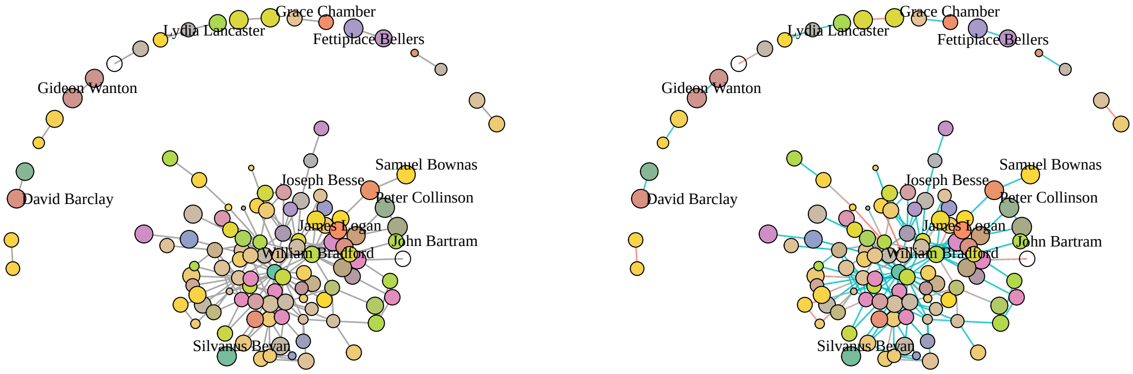
```
## IGRAPH 0e47074 U--- 119 174 --
## + attr: Name (v/c), Historical (v/c), Gender (v/c), Birthdate (v/n),
## | Deathdate (v/n), ID (v/n)
```

An edgelist is a dataframe containing a minimum of two columns. One column listing the source entity and the other being the target of the connected entity.If the the source and target are unordered and the connection between them does not matter, then the graph is called an undirected graph. If the connection is meaningful, the graph is referred to as directed. An edge may also have attributes such as the magnitude of the edge. If this is listed the edge is known as a weighted edge. A node list is a separate dataframe that contains the names and ids of each node. The advantage of creating a separate nodelist is that information and attributes about each entity can be added to this dataframe. The command listed above, ‘tbl_graph’ creates a network object from nodes and edges data. It is an object from the igraph library. A summary of the network object ‘quaker’ is listed underneath.

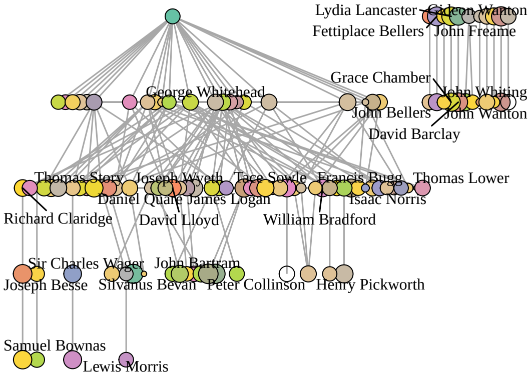
```
ggraph(quaker) + geom_node_point() + geom_edge_link()
```

ggraph is an extension of ggplot2 and supports relational data such as trees, graphs and networks.There are three primary aspects to ggraph nodes, edges and layouts. If the layout of a ggraph object isn’t specified, it will be created automatically, (the ‘stress’ layout can be seen as default in the plot above.The structure of the command is similar to ggplot.
**Fig. 1 & 2**

```r
nb.cols <- 65
mycolors <- colorRampPalette(brewer.pal(8, "Set2"))(nb.cols)

p1<-ggraph(quaker, layout="kk") +
    geom_edge_link0(aes(edge_width = quaker$target),edge_colour = "grey66") +
    geom_node_point(aes(fill = Historical, size=Birthdate),shape=21)+
    geom_node_text(aes(filter = Deathdate >= 1750, label = Name),family="serif", repel=TRUE)+
    scale_fill_manual(values = mycolors)+
    theme_graph()+
  theme(legend.position = "none")

p2<-ggraph(quaker, layout="kk") +
    geom_edge_link2(aes(edge_colour = node.Gender),edge_width = 0.5) +
    geom_node_point(aes(fill = Historical, size=Birthdate),shape=21)+
    geom_node_text(aes(filter = Deathdate >= 1750, label = Name),family="serif",repel = TRUE)+
    scale_fill_manual(values = mycolors)+
    theme_graph()+
  theme(legend.position = "none")

grid.arrange(p1, p2, ncol = 2)
```



The basic network structure from ggraph does not provide a lot of information. As well as this, the automatic layout makes the position and relationships of the nodes quite hard to discern. The layout can be described as the horizontal and vertical positions of nodes when the network structure is plotted.Finding the optimal layout for the network structure improves the aesthetics of the graph and helps gain mroe infromation from them. Similar to ggplot, the grammar of ggraph works with layers. The node and edge specifications can be changed by adding an extra layers with the + sign. Fig. 1 & 2 shown above, show the same layout for each network. The nodes are coloured by their historical significance and sized according to how long ago their birthdate was. In addition, Fig. 2 has coloured the relationships by gender, showing male-male and female-female connections. All those who died after the year 1750 have been labeled.



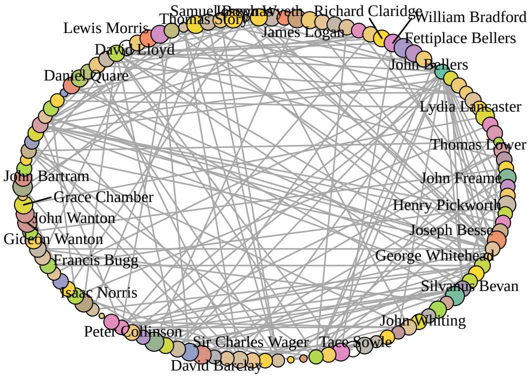


**Fig. 3 & 4**

**Fig. 5**

The drl layout seen in Fig. 5 is a force directed graph layout. In force directed networks, a force is assigned to a set of nodes or edges in a graph based on the tension of springs in Hooke's Law.

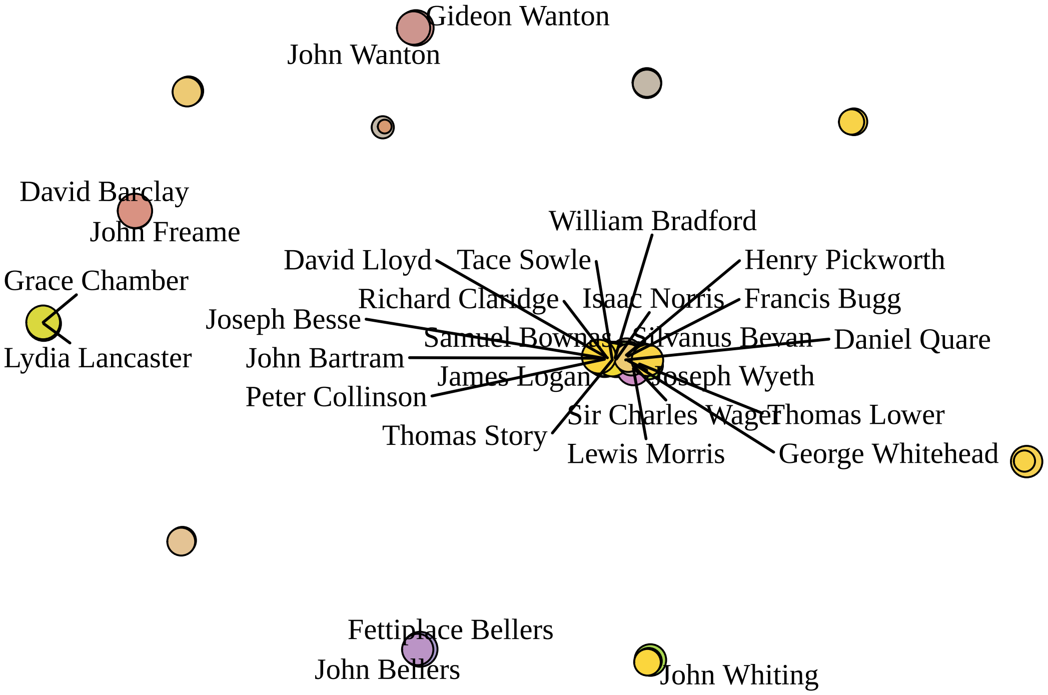


**Fig. 6 & 7**

In order to create a data subset it is possible to filter the data. %E>% is used for an edge operation and %N>% for a node. NOTE: For a subset the mode or direction of ggraph must be specified.

```
quaker1<- quaker %N>% filter(Historical==’Quaker preacher and writer’| Historical==’Quaker activist’ | I
quaker2<- quaker %N>% filter(Historical==’Quaker preacher and writer’| Historical==’Quaker activist’ |
  filter(centrality_degree() > 0)

r1<-ggraph(quaker1, ’star’) + geom_node_point() + geom_edge_link()
r2<-ggraph(quaker2, ’gem’) + geom_node_point() + geom_edge_link()

grid.arrange(r1, r2, ncol = 2)
```

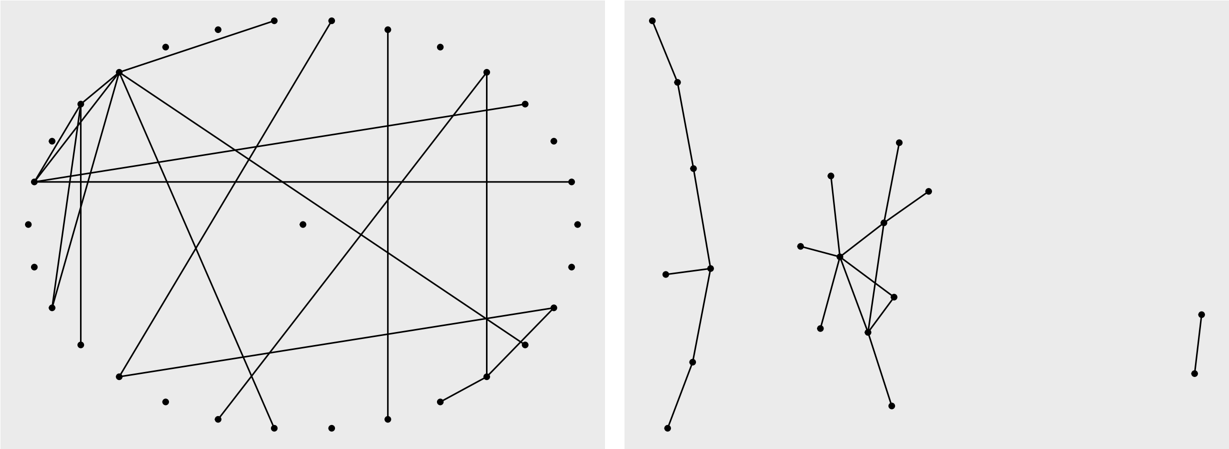


Fig. 6 & 7 above shows the same filtered subset of the network object. It can be seen that there are orphaned nodes- nodes with no connections- in Fig. 6. The command ‘filter(centrality_degree() > 0)’ removes the orphaned nodes in Fig. 7.

**Analysis**

One of the basic metrics of network analysis is measuring density. The igraph command is below. This is simply the ratio of actual edges in the network to all possible edges in the network

```
edge_density(quaker, loops = FALSE)
```

```
## [1] 0.02478279
```

On a scale of 0 to 1, we see that the density of the network is quite low. 0 represents no edges at all and 1 represents a perfectly connected network.
Transitivity measures the probability that the adjacent vertices of a vertex are connected.In this example, it means that if a person is connected to two others, how likely are the two connections to also being connected.

```
transitivity(quaker)
```

```
## [1] 0.169378
```

It can be seen that the level of transitivity in the graph is quite low.
The degree of a vertex is its most basic structural property, the number of its adjacent edges. The importance of a node can be measured by observing its degree.

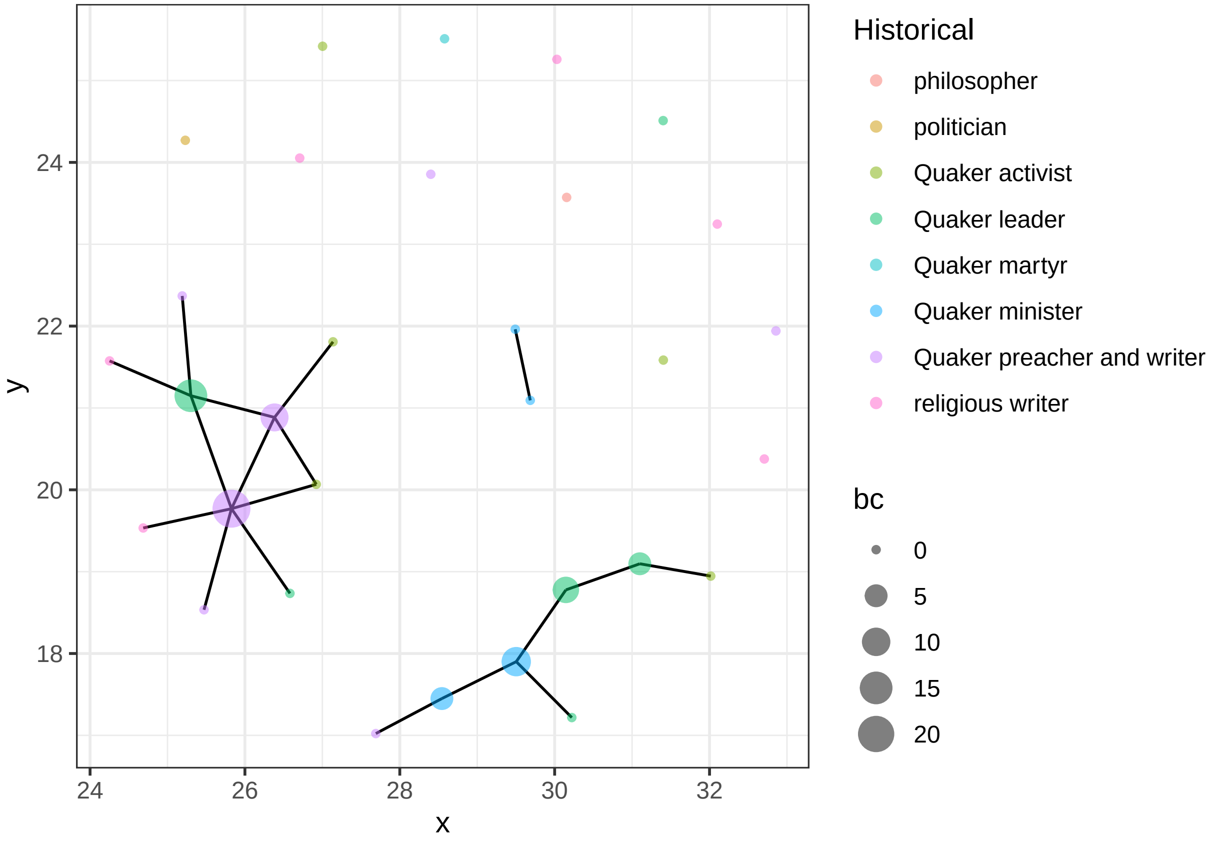
```
V(quaker)$Name[degree(quaker,mode="in")==max(degree(quaker,mode="in"))]
```

```
## [1] "George Fox"
```

It can be seen that George Fox, the founder of the quakers, has the highest degree.
'centrality_degree' is one of the techniques available in ggraph to measure node importance. It removes nodes that do not have relationships with other nodes. Of course, it is not possible to measure the strength of the edge or connection between nodes with this command.
Another technique to measure node importance is 'centrality_betweenness'. This measures the number of shortest routes from every node to another node, which pass through the given node. The output of Betweenness Centrality can be seen in Fig. 8.

**Fig. 8**

```
quaker1 <- quaker1 %N>% mutate(bc=centrality_betweenness())
ggraph(quaker1, layout="fr") +
geom_edge_link() +
geom_node_point(aes(col=Historical, size=bc), alpha=0.5)+
  theme_bw()
```


y
x
Historical
philosopher
politician
Quaker activist
Quaker leader
Quaker martyr
Quaker minister
Quaker preacher and writer
religious writer
bc