Exercise 2

Himanshu Mayank

2023-03-20

It takes many rides together to develop an informal connection, so you should pick one seat and take it consistently. You notice that communication is really possible with adjacent seats only, but any type of adjacency works: side, front, back, diagonal, even across the aisle. You need to find the best seat to develop informal connections

1. Creating a graph

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
library(visNetwork)
```

```
## Warning: package 'visNetwork' was built under R version 4.2.3
```

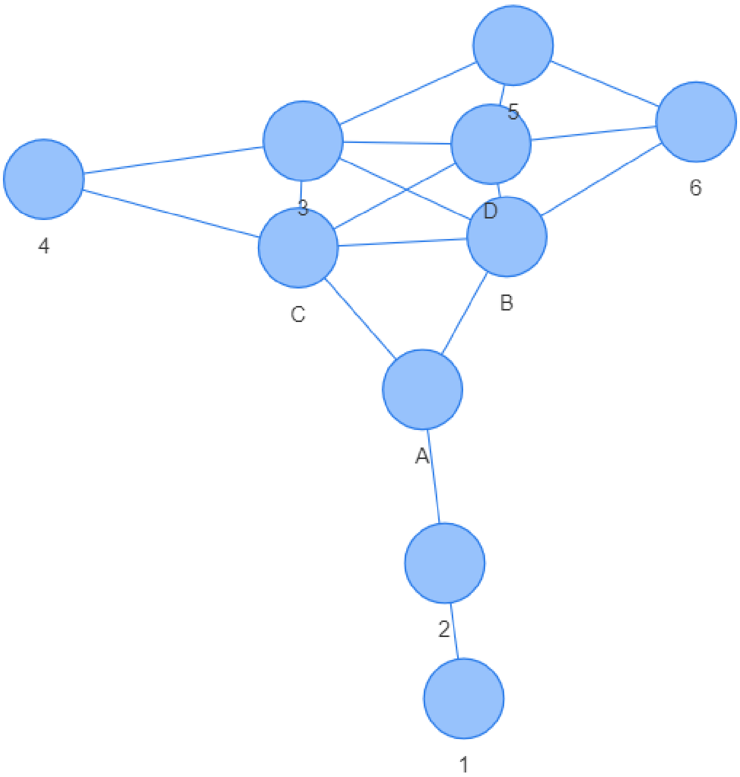
```
from_ <- c("1","2","3","3","3","3","3","4","5","5","6","6","D","D","B","B","A")
to_ <- c("2","A","D","C","B","4","5","C","D","6","D","B","B","C","C","A","C")
edge <- data.frame(from = from_, to=to_)
g <- graph_from_data_frame(edge, directed=FALSE)

# Create a data frame with node IDs and labels
nodes_df <- data.frame(id = V(g)$name, label = V(g)$name)

# Create a data frame with edge IDs and source/target nodes
edges_df <- get.data.frame(g, what = "edges")[, c("from", "to")]

# Create a visNetwork object with the nodes and edges data frames
visNetwork(nodes_df, edges_df) %>%
  visIgraphLayout(layout = "layout_with_fr") %>%
  visInteraction(navigationButtons = TRUE) %>%
  visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)
```

Select by id



Creating an adjacency matrix

```
# Get the adjacency matrix of the graph
adj_mat <- get.adjacency(g, sparse = FALSE)

# Show the adjacency matrix
adj_mat
```

```
##   1 2 3 4 5 6 D B A C
## 1 0 1 0 0 0 0 0 0 0 0
## 2 1 0 0 0 0 0 0 0 1 0
## 3 0 0 0 1 1 0 1 1 0 1
## 4 0 0 1 0 0 0 0 0 0 1
## 5 0 0 1 0 0 1 1 0 0 0
## 6 0 0 0 0 1 0 1 1 0 0
## D 0 0 1 0 1 1 0 1 0 1
## B 0 0 1 0 0 1 1 0 1 1
## A 0 1 0 0 0 0 0 1 0 1
## C 0 0 1 1 0 0 1 1 1 0
```

2. Calculating different types of centrality

Calculating the degree centrality

Degree centrality is a measure of node importance in a graph. It is based on the number of edges that are incident on a node, i.e., the number of connections that a node has with other nodes in the graph. Nodes with high degree centrality are those that are well-connected to other nodes in the graph and are therefore important in terms of information flow, communication, or influence.

```
degree_centrality <- degree(g, v = c("A", "B", "C", "D"), mode = "all")
print(degree_centrality)
```

```
## A B C D
## 3 5 5 5
```

Calculating the closeness centrality

Closeness centrality is based on the average length of the shortest path between a node and all other nodes in the graph. If a node has a high closeness centrality, it means that it can reach other nodes in the graph with fewer steps, and other nodes can reach it with fewer steps as well.

```
closeness_centrality <- closeness(g, v = c("A", "B", "C", "D"), mode = "all", normalized = TRUE)
print(closeness_centrality)
```

```
##         A         B         C         D
## 0.5625000 0.6428571 0.6428571 0.5625000
```

Calculating the betweenness centrality

Betweenness centrality is a measure of node importance in a graph that is based on the number of shortest paths between all pairs of nodes that pass through a particular node. Nodes with high betweenness centrality are those that lie on many of the shortest paths between other nodes in the graph. These nodes act as a bridge or bottleneck, facilitating the flow of information or resources between different parts of the network.

```
betweenness_centrality <- betweenness(g, v = c("A", "B", "C", "D"))
print(betweenness_centrality)
```

```
##         A         B         C         D
## 14.000000  9.033333  8.600000  3.266667
```

3. Discussing the consequences

Looking at the betweenness centrality B,C,D seem to be a good option because these seats will enable one to interact with maximum number of people. Among these seats B, C have higher closeness centrality, ie, because these nodes are connected to all other nodes in a more denser way.Lastly we look at the betweenness centrality, to ensure that the seat which we choose lies between the shortest path of other nodes. B has high betweeness centrality; hence B would be the most optimal node.

4. Plotting the network graph with labels and centrality values

```
# Calculate degree, closeness, and betweenness centrality
library(igraph)
library(visNetwork)

from_ <- c("1","2","3","3","3","3","3","4","5","5","6","6","D","D","B","B","A")
to_ <- c("2","A","D","C","B","4","5","C","D","6","D","B","B","C","C","A","C")
edge <- data.frame(from = from_, to=to_)
g <- graph_from_data_frame(edge, directed=FALSE)

# Calculate degree, closeness, and betweenness centrality
degree_centrality <- degree(g)
closeness_centrality <- closeness(g)
betweenness_centrality <- betweenness(g)

# Set the node labels to include degree, closeness, and betweenness centrality values
label_text <- paste("Degree: ", degree_centrality, " | ",
                    "Closeness: ", round(closeness_centrality, 2), " | ",
                    "Betweenness: ", round(betweenness_centrality, 2), sep = "")

# Create a data frame with node IDs and labels
nodes_df <- data.frame(id = V(g)$name, label = label_text)

# Create a data frame with edge IDs and source/target nodes
edges_df <- get.data.frame(g, what = "edges")[, c("from", "to")]

# Create a visNetwork object with the nodes and edges data frames
visNetwork(nodes_df, edges_df) %>%
  visIgraphLayout(layout = "layout_with_fr") %>%
  visInteraction(navigationButtons = TRUE) %>%
  visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%
  visNodes(size = 25, color = "lightblue")
```

Select by id