Exercise2

2023-03-21

For this assignment, I first created 2 datasets with excel, one with the Nodes called nodes, the other for the Edges, and the From-To called egdes.

Here is how I resolve the assigment after that :

First, loading the libraries and the data.

```
## Warning: le package 'igraph' a été compilé avec la version R 4.2.2

##
## Attachement du package : 'igraph'

## Les objets suivants sont masqués depuis 'package:stats':
##
##     decompose, spectrum

## L'objet suivant est masqué depuis 'package:base':
##
##     union

## Warning: le package 'dbplyr' a été compilé avec la version R 4.2.2

##
## Attachement du package : 'tidyr'

## L'objet suivant est masqué depuis 'package:igraph':
##
##     crossing

## Warning: le package 'tidyverse' a été compilé avec la version R 4.2.2

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.4.0      v dplyr   1.0.10
## v tibble  3.1.8      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.4

## Warning: le package 'ggplot2' a été compilé avec la version R 4.2.2

## Warning: le package 'readr' a été compilé avec la version R 4.2.2
```

```
## Warning: le package 'forcats' a été compilé avec la version R 4.2.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()       masks igraph::compose()
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()        masks stats::filter()
## x dplyr::groups()        masks igraph::groups()
## x dplyr::ident()         masks dbplyr::ident()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()
## x dplyr::sql()           masks dbplyr::sql()
```

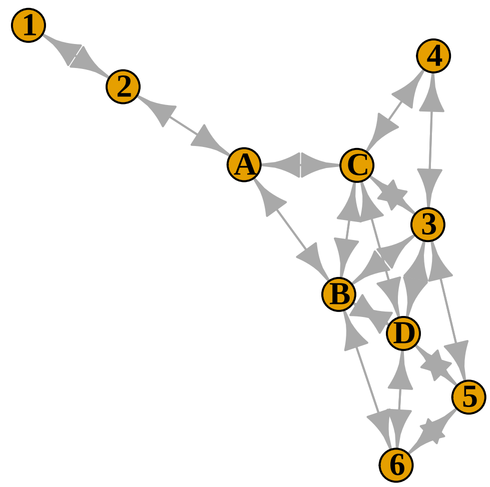
```
## Warning: le package 'tidygraph' a été compilé avec la version R 4.2.2
```

```
##
## Attachement du package : 'tidygraph'
##
## L'objet suivant est masqué depuis 'package:igraph':
##
##     groups
##
## L'objet suivant est masqué depuis 'package:stats':
##
##     filter
```

```
## Warning: le package 'visNetwork' a été compilé avec la version R 4.2.2
```

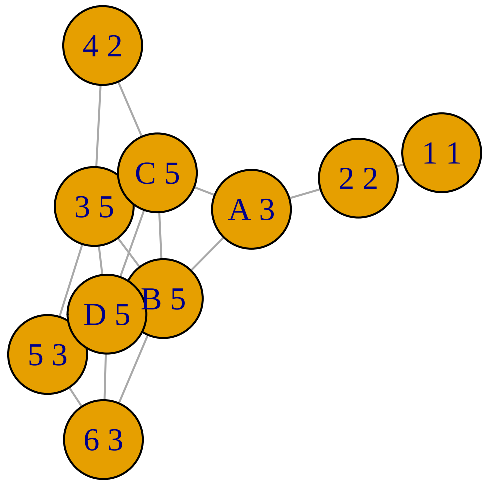
```
## Warning: le package 'readxl' a été compilé avec la version R 4.2.2
```

When creating the edges, I did it bi-directionally. Meaning that a relationship goes both ways, for example, node 1 is linked to node 2, and node 2 is linked to node 1.
I then plotted this relationship graph:



The next step was to calculate the degree of centrality of each node. I did it using the 'degree()' function from R. The degree of centrality measures how many connections an individual has in a network. A higher degree means the individual is more central, important, or influential because of his numerous connections. He can access more information. When I first calculated the degree centrality, I had a factor of two. I understood it came from my bidirectional relationship. To resolve that, I created an undirected graph and then use it for the rest of the exercise. Here is my result: the second number in the circle represents the degree of centrality.

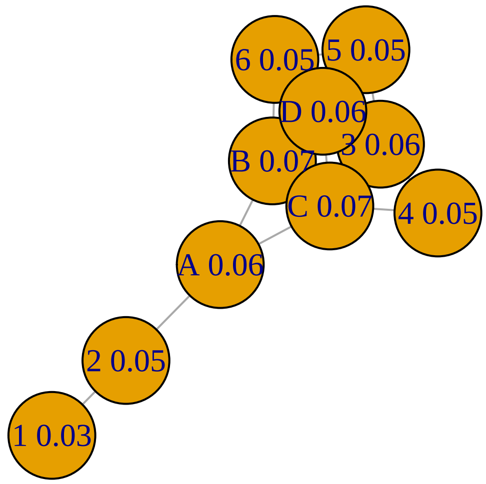
```
##  [1] 1 2 5 2 3 3 3 5 5 5
```



From what we can see, the most valuable seats are B, C, and D as they have a centrality degree of 5.

Now looking at the closeness centrality. It measures how close an individual is to all other individuals of the network. A high closeness centrality means the person can access information and resources more quickly because they are closer to other individuals in the network. I got the closeness centrality using R's 'closeness()' function. Here is my result: the second number in the circle represents the degree of closeness.

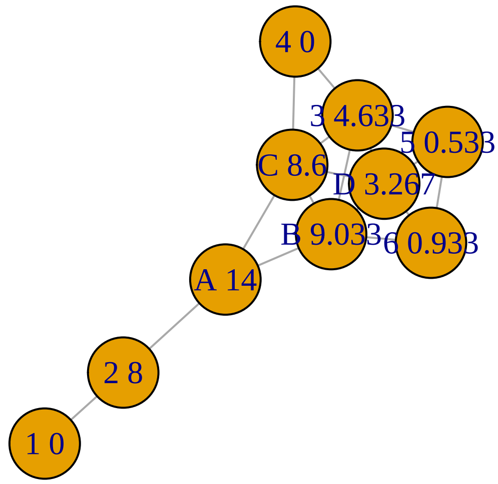
```
##  [1] 0.03333333 0.04545455 0.06250000 0.05000000 0.04761905 0.05263158
##  [7] 0.06250000 0.07142857 0.07142857 0.06250000
```



From the graph, we can see that in this case, B and C have the best score with 0.07. They are then followed by D and A with 0.06.

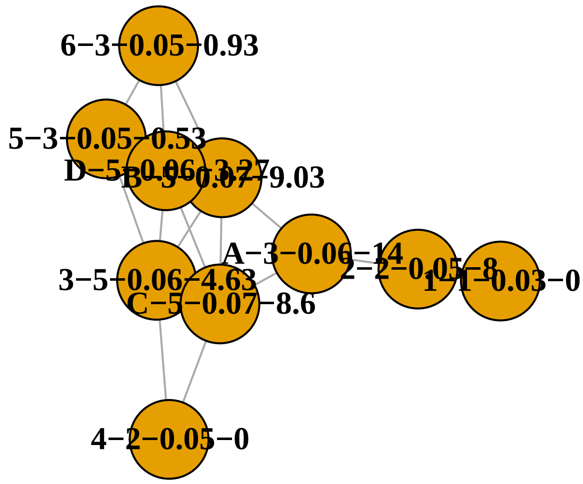
The last step was studying the betweenness centrality. This measures how often an individual is on the shortest path between others in the network. An individual with high betweenness centrality has more control over the flow of information and resources. They are critical to communication. I got the betweenness centrality using R's 'betweenness()' function. Here is my result: the second number in the circle represents the degree of betweenness.

```
##  [1]  0.0000000  8.0000000  4.6333333  0.0000000  0.5333333  0.9333333
##  [7] 14.0000000  9.0333333  8.6000000  3.2666667
```



In this case, the node with the highest score is node A , with 14. It is then followed by B (9.033), then C (8.6), and finally D (3.267).

Finally, I plotted the network with all the centrality values.



I would choose seat B on the bus, which seems to have the best node results. It has the best value for the centrality degree and closeness degree. This means sitting at that seat makes you an influential person in the network as you have more connections and can access more information and resources, all of that more quickly as you are closer to the others individuals. Nevertheless, the downside of this seat is that it's not the one with the most betweenness centrality. The score is good, but it's not from this seat that you can have the most control over the info and impact communication and coordination between others.