Exercice2

```r
library(readxl)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'

## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag

## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidygraph)
```

```
## Warning: le package 'tidygraph' a été compilé avec la version R 4.1.3

##
## Attachement du package : 'tidygraph'

## L'objet suivant est masqué depuis 'package:stats':
##
##     filter
```

```r
library(tidyverse)
```

```
## Warning: le package 'tidyverse' a été compilé avec la version R 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidygraph::filter() masks dplyr::filter(), stats::filter()
## x dplyr::lag()        masks stats::lag()
```

```r
library(ggraph)
```

```
## Warning: le package 'ggraph' a été compilé avec la version R 4.1.3
```

```r
DATASET_EX2 <- read_excel("C:/Users/Mehdi/Desktop/2022-ona-assignements/EX2/DATASET_EX2.xlsx")
df = DATASET_EX2
df
```

```
## # A tibble: 34 x 2
##    X     Y
##    <chr> <chr>
##  1 1     2
##  2 2     A
##   3 2     1
##   4 A     2
##   5 A     B
##   6 A     C
##   7 B     A
##   8 B     C
##   9 B     D
##  10 B     3
## # ... with 24 more rows
```

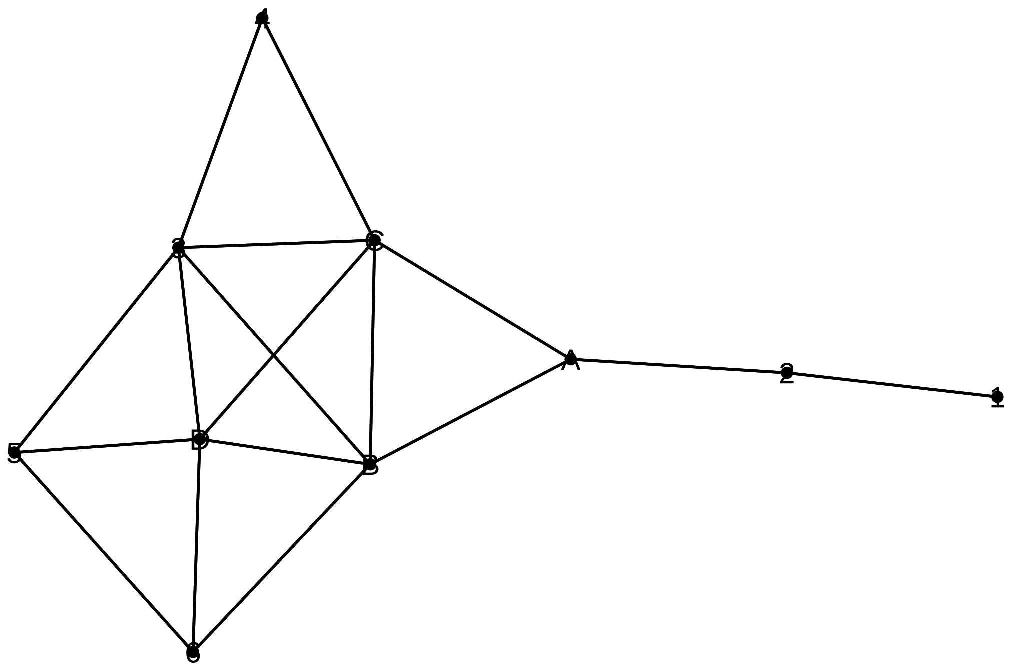
Including Plots

You can also embed plots, for example:

```
my_graph = as_tbl_graph(df)

ggraph(my_graph) +
  geom_edge_link() +
  geom_node_point()+
  geom_node_text(aes(label = name)) +
  theme_graph()
```

```
## Using `stress` as default layout
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Centrality degree

```
centrality_deg = my_graph %>% activate(nodes) %>% mutate(degree_of_centrality = centrality_degree())

centrality_deg
```

```
## # A tbl_graph: 10 nodes and 34 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 10 x 2 (active)
##   name  degree_of_centrality
##   <chr>                <dbl>
## 1 1                        1
## 2 2                        2
## 3 A                        3
## 4 B                        5
## 5 C                        5
## 6 6                        3
## # ... with 4 more rows
## #
## # Edge Data: 34 x 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     2     3
## 3     2     1
## # ... with 31 more rows
```

Betweeness degree

```
betweeness = centrality_deg %>% activate(nodes) %>% mutate(betweenness = centrality_betweenness())
```

```
## Warning in betweenness(graph = graph, v = V(graph), directed = directed, :
## 'nobigint' is deprecated since igraph 1.3 and will be removed in igraph 1.4
```

```
betweeness
```

```
## # A tbl_graph: 10 nodes and 34 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 10 x 3 (active)
##   name  degree_of_centrality betweenness
##   <chr>                <dbl>       <dbl>
## 1 1                        1        0
## 2 2                        2       16
## 3 A                        3       28
## 4 B                        5       18.1
## 5 C                        5       17.2
## 6 6                        3        1.87
## # ... with 4 more rows
## #
## # Edge Data: 34 x 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     2     3
## 3     2     1
## # ... with 31 more rows
```

```
betweeness %>% filter(name %in% c('A','B','C','D'))
```

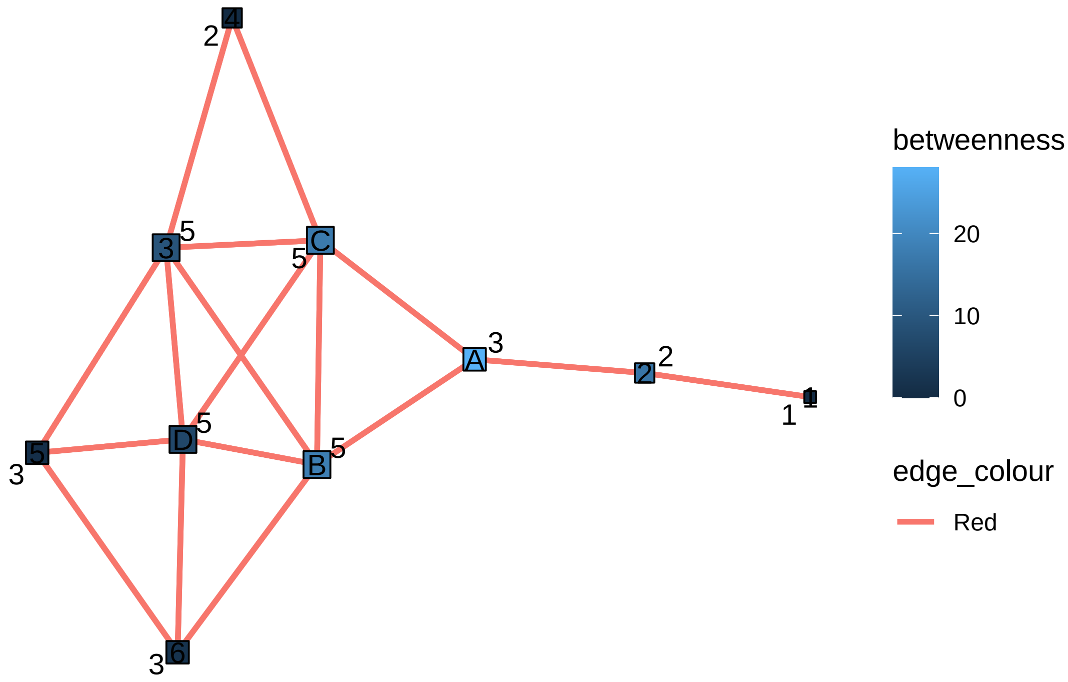
```
## # A tbl_graph: 4 nodes and 10 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 4 x 3 (active)
##   name  degree_of_centrality betweenness
##   <chr>                <dbl>       <dbl>
## 1 A                        3       28
## 2 B                        5       18.1
## 3 C                        5       17.2
## 4 D                        5        6.53
## #
## # Edge Data: 10 x 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     1     3
## 3     2     1
## # ... with 7 more rows
```

Conclusion

Regarding the centrality degree, it clearly comes that B,C and D are the most interesting seats. They all have the same centrality degree, equal to 5, while A is at 3. The betweeness score allows to select the seat B within the seats B,C,D because it's the one with the highest betweeness score. To note that the seat A is the one with the highest betweeness but considering that it's for a short period of time that the person will be taking the bus, it's more important to build a network rather than being the person that allows to connect people or groups. However, if it's for a long period, the seat A could be interesting as it will act like a bridge to connect two distant groups.

To sum up, if it's for short period, seat B is the more indicated. If it's for a long period, seat A is more important.

```
ggraph(betweeness,  "stress", bbox = 15) +
  geom_edge_link2(aes(edge_colour = "Red"), edge_width = 1) +
  geom_node_point(aes(fill = betweenness, size = degree_of_centrality), shape = 22) +
  geom_node_text(aes(label = name)) + geom_node_text(aes(label = degree_of_centrality), repel = TRUE) +
  scale_size(range = c(2, 5), guide = "none") +
  theme_graph(fg_text_colour = 'white', base_family = 'Helvetica')
```


4
2
3
5
C
5
A
3
2
2
1
1
5
3
D
5
B
5
6
3
betweenness
20
10
0
edge_colour
Red