ONA_Exercise 2_Amr Maraqa

2023-03-21

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.5
## ✔ tibble  3.1.8     ✔ stringr 1.4.1
## ✔ readr   2.1.3     ✔ forcats 0.5.2
```

```
## Warning: package 'readr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.2
```

```
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.2.2
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```
library(tidygraph)
```

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

```
##
## Attaching package: 'tidygraph'
##
## The following object is masked from 'package:igraph':
##
##     groups
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggraph)
```

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

```
# Create Data
```

```
nodes <- c('A', 'B', 'C', 'D', '1', '2', '3', '4', '5', '6')
front <- c('2', 'A', NA, 'B',NA, '1', 'C', NA, 'D', NA)
back <- c('B', 'D', '3', '5', '2', 'A', NA, NA, NA, NA)
right <- c(NA, 'C', NA, '3', NA, NA, '4', NA, NA, 'D')
left <- c(NA, NA, 'B', '6', NA, NA, 'D', '3', NA, NA)
NE <- c('C', '3', '4', NA, NA, NA, NA, NA, NA, '5')
SE <- c(NA, NA, NA, 'C', NA, NA, NA, NA, '3', 'B')
NW <- c(NA, '6', 'D', NA, NA, NA, '5', NA, NA, NA)
SW <- c(NA, NA, 'A', NA, NA, NA, 'B', 'C', '6', NA)
seats_df <- cbind(nodes, front, back, right, left,NE,SE,NW,SW)

# Define nodes and edges
edges <- rbind(as.data.frame(cbind(nodes, front)) %>% rename(c('to' = front, 'from' = nodes)),
               as.data.frame(cbind(nodes, back)) %>% rename(c('to' = back, 'from' = nodes)),
               as.data.frame(cbind(nodes, right)) %>% rename(c('to' = right, 'from' = nodes)),
               as.data.frame(cbind(nodes, left)) %>% rename(c('to' = left, 'from' = nodes)),
               as.data.frame(cbind(nodes, NE)) %>% rename(c('to' = NE, 'from' = nodes)),
               as.data.frame(cbind(nodes, SE)) %>% rename(c('to' = SE, 'from' = nodes)),
               as.data.frame(cbind(nodes, NW)) %>% rename(c('to' = NW, 'from' = nodes)),
               as.data.frame(cbind(nodes, SW)) %>% rename(c('to' = SW, 'from' = nodes)))

edges <- edges %>% drop_na()
nodes <- as.data.frame(nodes)
colnames(nodes) <- c('node')
```

```
# Create Network
network <- tbl_graph(nodes = nodes, edges = edges, directed=FALSE)
```

```
# Calculating node metrics
network <- network %>%
  mutate(degree = centrality_degree()/2,
         closeness = centrality_closeness_harmonic(),
         betweenness = centrality_betweenness()) %>%
  mutate(avg = (degree + closeness + betweenness)/3) %>%
  mutate(label = paste0(node, '\n',
                        'Degree: ',round(degree,2), '\n',
                        'Closeness: ',round(closeness,2), '\n',
                        'Betweenness: ',round(betweenness,2), '\n',
                        'Avg: ',round(avg,2)))
net_data <- network %>% data.frame() %>% tibble()
```

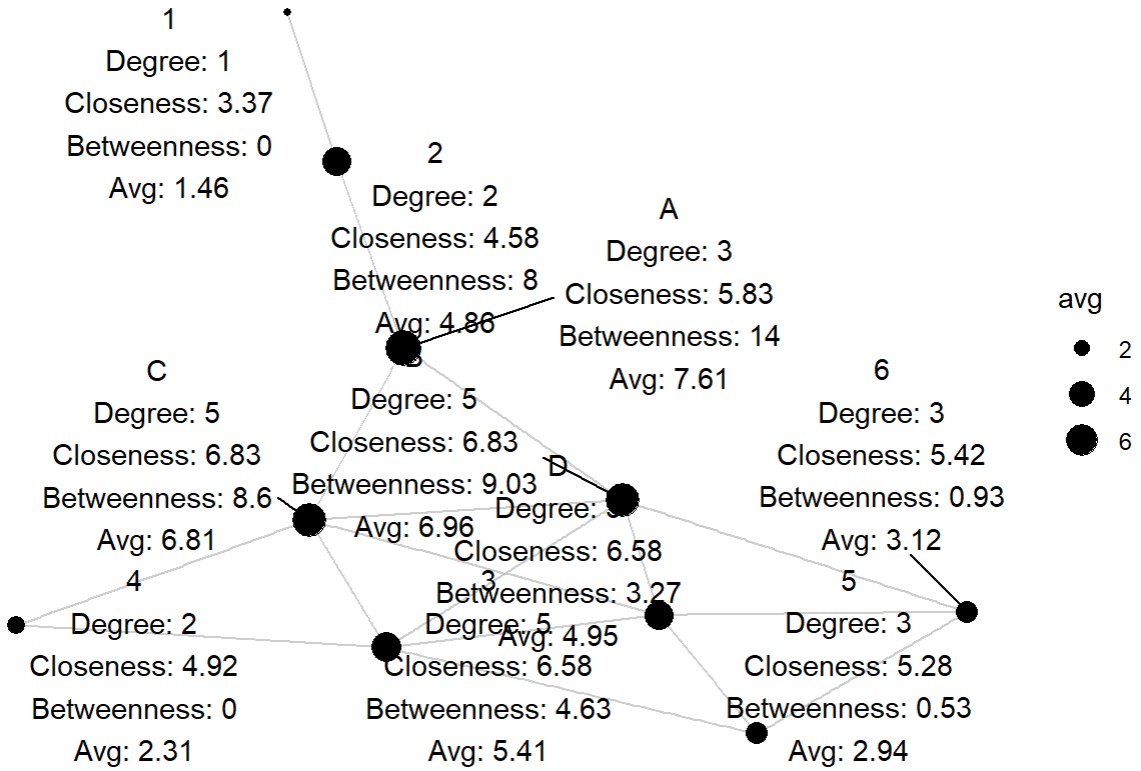
```
# Plot Network
network %>%
  ggraph(layout="graphopt") +
  geom_edge_link(edge_colour = "grey", alpha=0.5) +
  geom_node_point(aes(size=avg)) +
  geom_node_text(aes(label = label), repel=TRUE) +
  theme_graph(foreground=NA)
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family not
## found in Windows font database
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font family not
## found in Windows font database
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```



Seat Choice and Explanation

I would choose to sit in seat B on the fakebook bus. Even though my average score on that seat wouldn't be the highest (seat A avg = 7.61), I would have the highest degree and measure of closeness in the network.This means that seat B would allow me to be the most popular among the Fakebook interns and to form strong relationships with them as well. Although the person sitting in seat A would have a higher measure of betweenness than I would, seat B would still give me access to many friend circles on the bus. One of the drawbacks of my choice is it reliance on the person in seat A for conncection with those in seats 1 and 2, as he/she would be the only link between me and them. If those sitting in seats 1 and 2 happen to be crucial to my role or future in a way (group leader, well-connected), I would be greatly disadvantaged to be in seat B, because my access to the people in seat 1 and 2 would be subject to the unpredicatability of the behavior of the person in seat A.