# Exercise 02

2024-03-19

# R Markdown

```
# Load necessary libraries
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(igraph)
```

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

```
## 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)
```

```
##
## Attaching package: 'tidygraph'
```

```
## The following object is masked from 'package:igraph':
##
##     groups
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

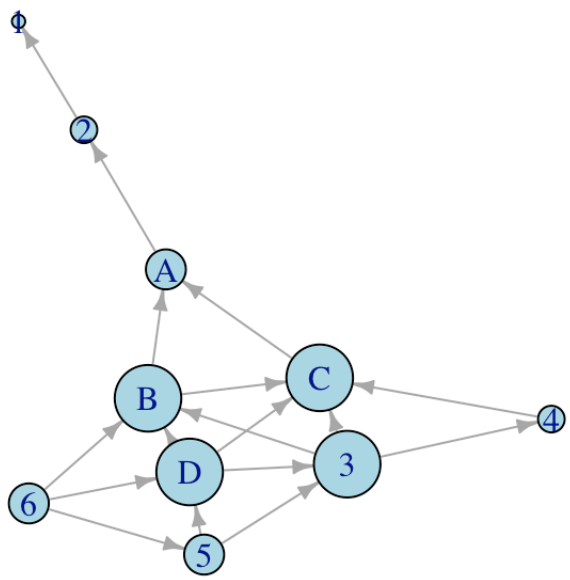
```
# Load necessary libraries
# Read the dataset

FACEBOOK <- read.csv("/Users/yuyichen/Desktop/Winter 2024/ORGB - 672/2024-ona-assignm
ents/FACEBOOK.csv")

# Convert to graph
g <- graph_from_data_frame(FACEBOOK, directed = TRUE)

# Step 3: Calculate centrality measures for A, B, C, and D
# Degree Centrality
degree_centrality <- degree(g, v = c("A", "B", "C", "D"))

# Closeness Centrality
closeness_centrality <- closeness(g, v = c("A", "B", "C", "D"))

# Betweenness Centrality
betweenness_centrality <- betweenness(g, v = c("A", "B", "C", "D"))

# Combine results into a data frame for easier viewing
centrality_measures <- data.frame(
  Node = c("A", "B", "C", "D"),
  Degree = degree_centrality,
  Closeness = closeness_centrality,
  Betweenness = betweenness_centrality
)

# Print the centrality measures
print(centrality_measures)
```

```
##   Node Degree  Closeness Betweenness
## A    A      3 0.33333333        14.0
## B    B      5 0.14285714         8.0
## C    C      5 0.16666667         7.5
## D    D      5 0.07142857         4.0
```

```
# Convert to graph
g <- graph_from_data_frame(FACEBOOK, directed = TRUE)

# Assuming the graph 'g' has already been created as before


# Ensure UTF-8 Encoding for the session
Sys.setlocale("LC_ALL", "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
# Function to clean text data
clean_text <- function(text) {
  iconv(text, to = "ASCII//TRANSLIT")
}

# Convert and clean vertex names to ensure compatibility
V(g)$name <- sapply(V(g)$name, clean_text)

# Calculate centrality measures for all nodes
degree_centrality_all <- degree(g)
closeness_centrality_all <- closeness(g)
betweenness_centrality_all <- betweenness(g)

# Pre-calculate the layout
layout <- layout_nicely(g)

# Plot the network graph
plot(g,
     main = "Network Graph with Centrality Measures",
     vertex.label = V(g)$name,
     vertex.color = "lightblue",
     edge.arrow.size = 0.5,
     vertex.size = degree_centrality_all * 5, # Adjust the size based on degree centr
ality
     layout = layout)

# Iterate through the specified nodes and add annotations for centrality measures
specified_nodes <- c("A", "B", "C", "D")
for (node in specified_nodes) {
  if(node %in% V(g)$name) {
    v_id <- which(V(g)$name == node)

    # Prepare and clean the text to display
    centrality_text <- sprintf("D: %0.2f\nC: %0.2f\nB: %0.2f",
                               degree_centrality_all[v_id],
                               closeness_centrality_all[v_id],
                               betweenness_centrality_all[v_id])
    centrality_text <- clean_text(centrality_text)

    x <- layout[v_id, 1]
    y <- layout[v_id, 2] + 0.15

    text(x, y, labels = centrality_text, cex = 0.7)
  }
}
```




```
# Load necessary libraries
library(igraph)
library(ggraph)
```

```
## Loading required package: ggplot2
```

```
library(tidygraph)
library(dplyr)

# Assuming 'g' is your igraph object

# Convert igraph object to a tbl_graph
graph_tbl <- as_tbl_graph(g)

# Calculate centrality measures
centrality <- data.frame(
  name = V(g)$name,
  degree = as.numeric(degree(g)),
  closeness = as.numeric(closeness(g)),
  betweenness = as.numeric(betweenness(g)),
  stringsAsFactors = FALSE
)

# Combine node name and centrality measures into a single label
centrality <- centrality %>%
  mutate(label = sprintf("%s\nD:%0.2f\nC:%0.2f\nB:%0.2f", name, degree, closeness, be
tweenness))

# Join centrality measures back to the tbl_graph
graph_tbl <- graph_tbl %>%
  activate(nodes) %>%
  left_join(centrality, by = "name")

# Adjusted plot with node names and centrality measures as labels
p <- ggraph(graph_tbl, layout = "kk") +
  geom_edge_diagonal(color = "gray", alpha = 0.4) +
  geom_node_point(size = 3) +  # Increase node size
  geom_node_text(aes(label = label), size = 3, vjust = 0.5) +  # Increase text size a
nd adjust vertical position
  theme_graph(base_size = 14) +  # Increase base font size

  ggtitle("Network Graph with Centrality Measures")

# Display the plot
print(p)
```

# Network Graph with Centrality Measures