Untitled

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.7     ✓ dplyr   1.0.9
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1

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

```
library(tidygraph)
```

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

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

```
library(ggraph)

# define nodes
node_names <- tibble(
  id   = c(1,2,3,4,5,6,7,8,9,10),
  name = c("1","2","A","B","C","6","D","3","4","5")
)
node_names
```

```
## # A tibble: 10 × 2
##       id name
##    <dbl> <chr>
##  1     1 1
##  2     2 2
##  3     3 A
##  4     4 B
##  5     5 C
##  6     6 6
##  7     7 D
##  8     8 3
##  9     9 4
## 10    10 5
```

```
# define connections (have to correspond to ties 1-2, 2-3, 2-4, 3-4)
# for each element in `from` there is a corresponding element in `to`
edge_list <- tibble(
  from = c(1,2,3,3,4,4,4,4,5,5,5,6,6,7,7,8,8),
  to   = c(2,3,4,5,5,6,7,8,7,8,9,7,10,8,10,9,10)
)
edge_list
```

```
## # A tibble: 17 × 2
##     from    to
##    <dbl> <dbl>
##  1     1     2
##  2     2     3
##  3     3     4
##  4     3     5
##  5     4     5
##  6     4     6
##  7     4     7
##  8     4     8
##  9     5     7
## 10     5     8
## 11     5     9
## 12     6     7
## 13     6    10
## 14     7     8
## 15     7    10
## 16     8     9
## 17     8    10
```

```
# combine this information into a network graph object
friendship_graph <- tbl_graph(nodes = node_names, edges = edge_list, directed = FALSE
friendship_graph
```

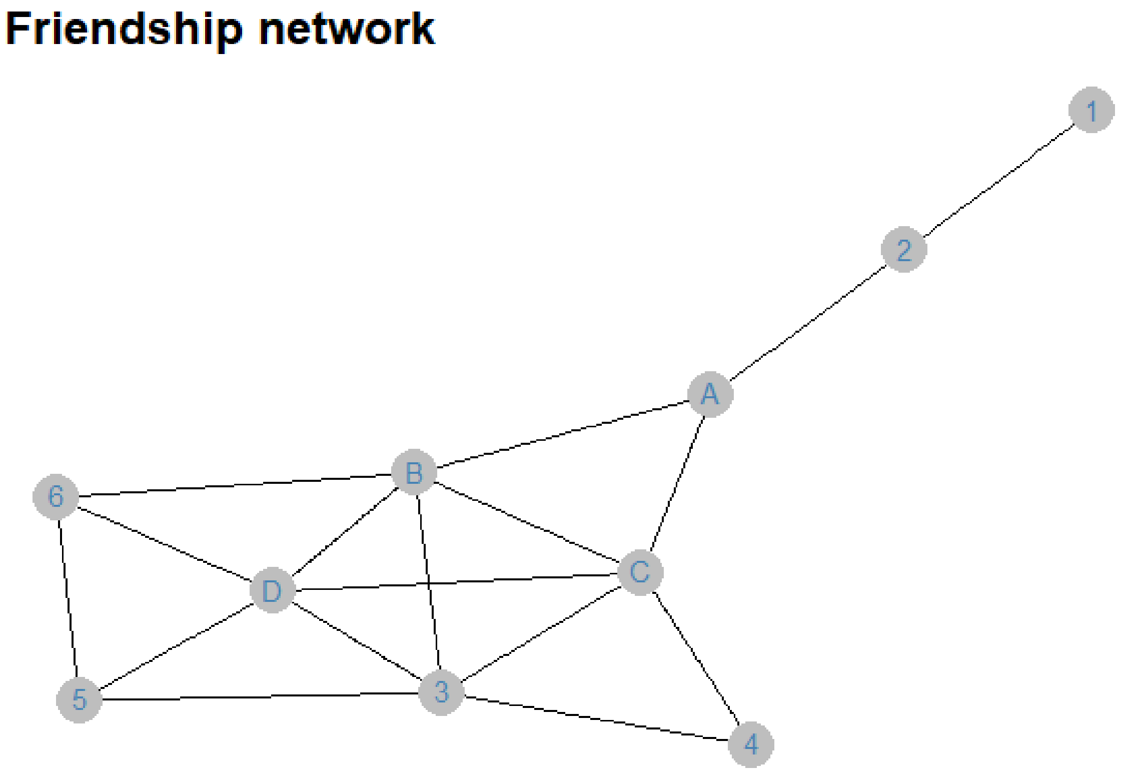
```
## # A tbl_graph: 10 nodes and 17 edges
## #
## # An undirected simple graph with 1 component
## #
## # Node Data: 10 × 2 (active)
##      id name
##   <dbl> <chr>
## 1     1 1
## 2     2 2
## 3     3 A
## 4     4 B
## 5     5 C
## 6     6 6
## # … with 4 more rows
## #
## # Edge Data: 17 × 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     2     3
## 3     3     4
## # … with 14 more rows
```

```
friendship_graph %>%
    ggraph(layout = 'kk') +
    geom_edge_link() +
    geom_node_point(size = 8, colour = 'gray') +
    geom_node_text(aes(label = name), colour = 'steelblue', vjust = 0.4) +
    ggtitle('Friendship network') +
    theme_graph()
```

```
## 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
```



```
friendship_graph <- friendship_graph %>%
  activate(nodes) %>% # we need to state we'll be adding to nodes, not edges
  mutate(d_centrality = centrality_degree()) %>%  # adding measure of degree centrali
  mutate(b_centrality = centrality_betweenness()) # adding betweenness centrality
```

```
## 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
```

```
## 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
```

```
friendship_graph
```

```
## # A tbl_graph: 10 nodes and 17 edges
## #
## # An undirected simple graph with 1 component
## #
## # Node Data: 10 × 4 (active)
##      id name  d_centrality b_centrality
##   <dbl> <chr>        <dbl>        <dbl>
## 1     1 1                1        0
## 2     2 2                2        8
## 3     3 A                3       14
## 4     4 B                5        9.03
## 5     5 C                5        8.6
## 6     6 6                3        0.933
## # … with 4 more rows
## #
## # Edge Data: 17 × 2
##    from    to
##   <int> <int>
## 1     1     2
## 2     2     3
## 3     3     4
## # … with 14 more rows
```

```
friendship_graph %>%
  ggraph(layout = 'kk') +
  geom_edge_link() +
  geom_node_point(aes(size = d_centrality, colour = b_centrality)) +
  scale_color_continuous(guide = 'legend') +
  geom_node_text(aes(label = name), colour = 'red', vjust = 1.6) +
  ggtitle('Friendship network') +
  theme_graph()
```

```
## 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
```

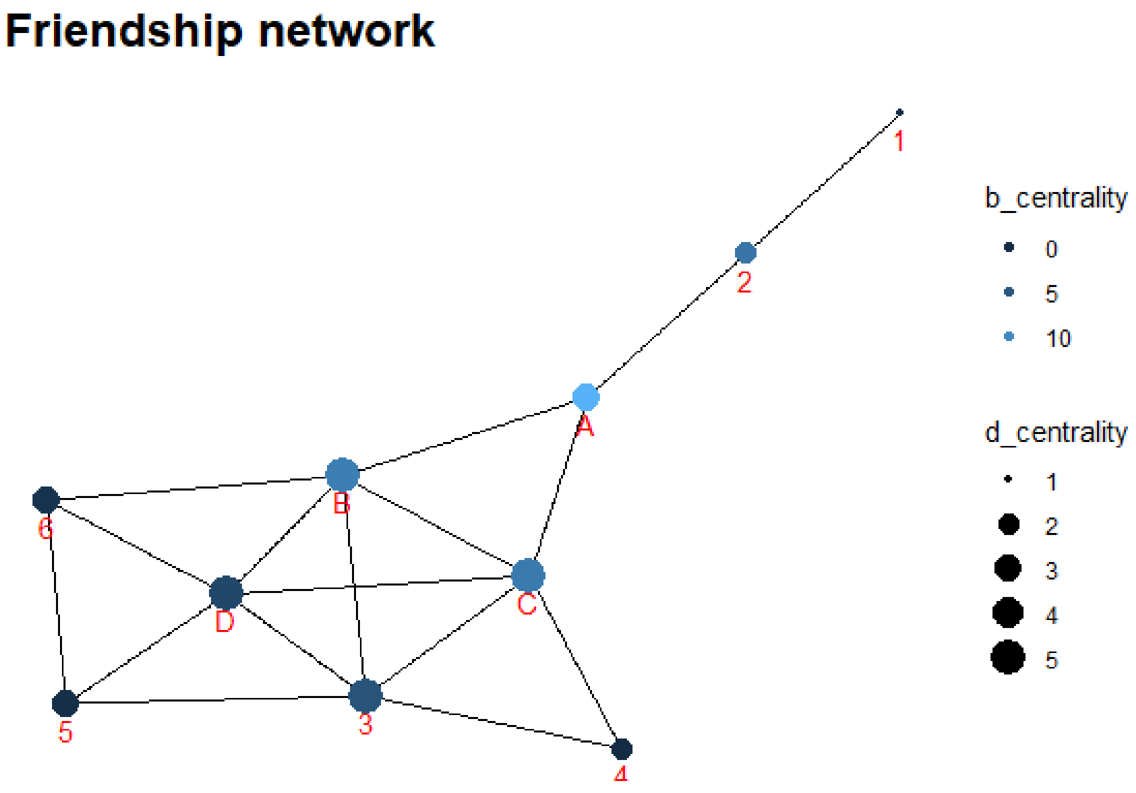
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## family not found in Windows font database
```

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

```
## 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

## 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
```



Based on the Colours shown on the graph, A seems to be the most central. It acts as a broker between the rest of the network and 1 and 2. A is also connected to B and C which are the 2 biggest nodes with the most connections at 5 each. A can reach anybody in the network within 3 jumps.

The next best options are B and C in terms of the nu,bers of connections they have as well as their appeared centrality within the network, having a large number of connections.