ex3

Load the data

```
library(arrow)
```

```
## Warning: package 'arrow' was built under R version 4.1.2

##
## Attaching package: 'arrow'

## The following object is masked from 'package:utils':
##
##     timestamp
```

```
df = read_parquet("~/Desktop/McGill/ORGB/2022-ona-assignments/ex3/app_data_sample.parquet")
```

Predicting examiners' gender based on first name:

The gender package attempts to infer gender (or more precisely, sex assigned at birth) based on first names using historical data, typically data that was gathered by the state.

```
library(gender)
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
```

```
first_name = df %>%distinct(examiner_name_first)
gender_probability = gender(first_name$examiner_name_first)
gender_dictionary = gender_probability %>% select(name,gender)
df <- df %>% left_join(gender_dictionary, by = c("examiner_name_first" = "name"))
head(df$gender)
```

```
## [1] "female" NA       "female" "female" "male"   "female"
```

The gender package assign gender based on historical data. Some of the name is not in the data set, thus there are some missing gender information. I filled those values by distribution.

```
table(is.na(df$gender))
```

```
##
##   FALSE    TRUE
## 1714618  303859
```

```
gender_na = is.na(df$gender)
gender_fill = sample(df$gender[!gender_na], size = sum(gender_na), replace = TRUE)
df$gender[is.na(df$gender)] <- gender_fill
table(is.na(df$gender))
```

```
##
##   FALSE
## 2018477
```

All the missing value has been filled.

Predicting examiners' race based on last name:

The "predictrace" package predict the race of a surname using U.S. Census data which says how many people of each race has a certain surname.

```
library(predictrace)
race = predict_race(df$examiner_name_last, probability = FALSE)
df$race = race$likely_race
head(df$race,10)
```

```
##  [1] "white" "white" "white" "white" "white" "white" "black" "white" NA
## [10] "asian"
```

Again, fill the missing values based on distribution.

```
table(is.na(df$race))
```

```
##
##   FALSE    TRUE
## 1704131  314346
```

```
race_na = is.na(df$race)
race_fill = sample(df$race[!race_na], size = sum(race_na), replace = TRUE)
df$race[is.na(df$race)] <- race_fill
table(is.na(df$race))
```

```
##
##   FALSE
## 2018477
```

Calculate Tenure

To calculate tenure, I need to calculate the time the application stay in the system.

For most applications, the filing date is the date on which PTO received the application.

The appl_status_date variable indicates the date that the application entered its most recent status (or status as of the end of 2014).

```
tenure_info <- df %>% select(examiner_id, filing_date, appl_status_date)

library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:arrow':
##
##     duration

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

```
tenure_info = tenure_info %>% mutate(appl_status_date = as_date(dmy_hms(appl_status_date)))
tenure_info$tenure_days = as.numeric(difftime(tenure_info$appl_status_date,tenure_info$filing_date,unit

## detect missing values
table(is.na(tenure_info$tenure_days))
```

```
##
##   FALSE    TRUE
## 2013867    4610
```

```
## fill missing values
tenure_na = is.na(tenure_info$tenure_days)
tenure_fill = sample(tenure_info$tenure_days[!tenure_na], size = sum(tenure_na), replace = TRUE)
tenure_info$tenure_days[is.na(tenure_info$tenure_days)] <- tenure_fill
table(is.na(tenure_info$tenure_days))
```

```
##
##   FALSE
## 2018477
```

```
## join with df
df$tenure = tenure_info$tenure_days
```

Pick two workgroup

The two group I pick is 1648 and 1722. 1600 – Biotechnology 1700 – Chemical and Materials Engineering

```r
wg = as.numeric(substr(df$examiner_art_unit, 1, 3))
df$wg = wg
group_164 = df %>% filter(df$wg == 164)
group_172 = df %>% filter(df$wg == 172)
```

Examing Group 1648

```r
## summary
table(group_164$gender)
```

```
##
## female   male
##  45817  47525
```

```r
table(group_164$race)
```

```
##
## american_indian           asian           black        hispanic           white
##               4           24553            3965            1405           63415
```

Examing Group 1722

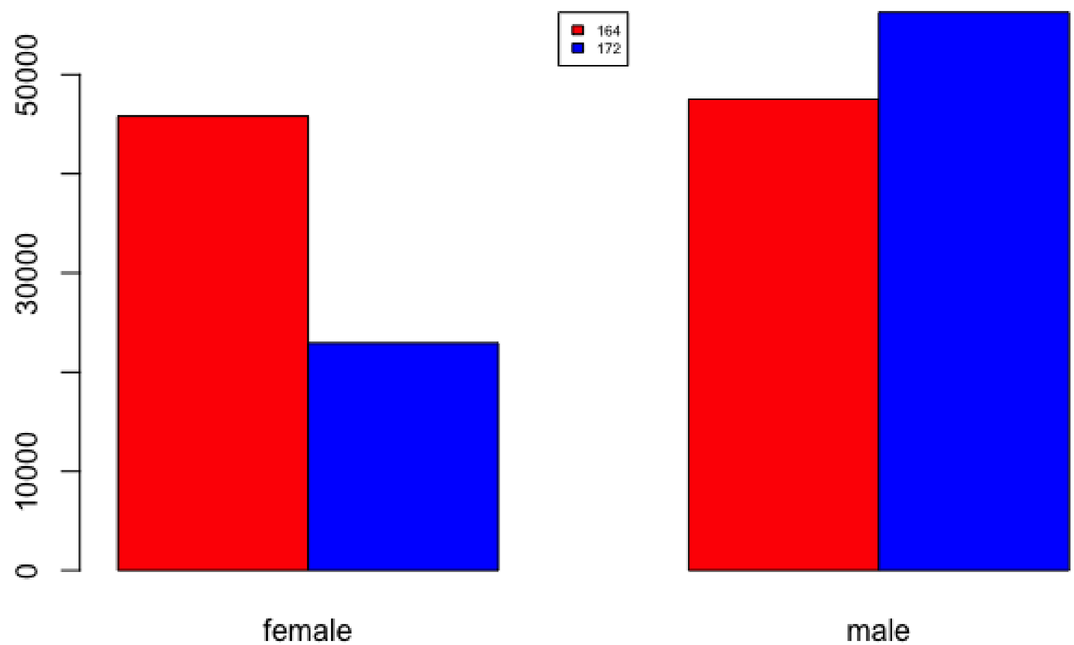
```r
## summary
table(group_172$gender)
```

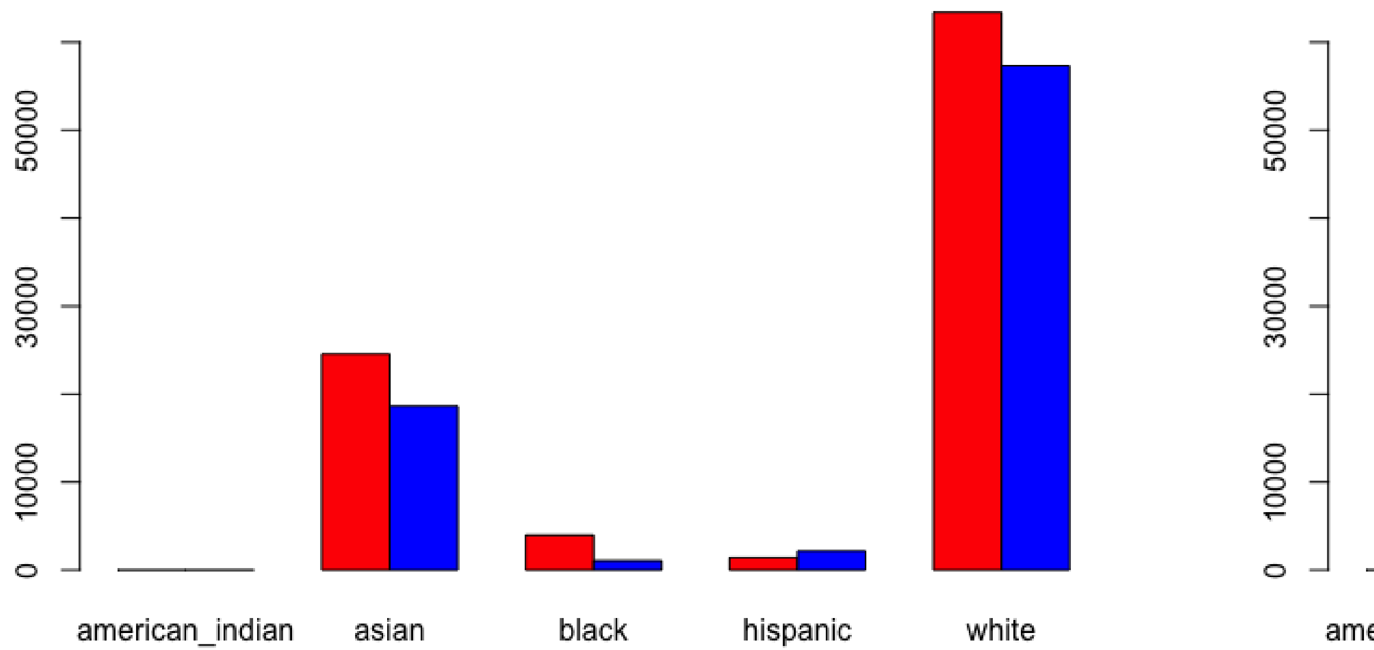
```
##
## female   male
##  22906  56289
```

```r
table(group_172$race)
```

```
##
## american_indian           asian           black        hispanic           white
##               1           18644            1058            2155           57337
```

```r
two_group_gender <- t(cbind(table(group_164$gender), table(group_172$gender)))

barplot(two_group_gender, beside=T, col=c("red","blue"))
par(xpd=T)
legend("top",legend = c("164","172"), fill=c("red","blue"), cex = 0.5)
```


164
172
50000
30000
10000
0
female
male



Create advice networks from edges-sample

```
library(tidyverse)
net = read_csv("~/Desktop/McGill/ORGB/2022-ona-assignments/ex3/edges_sample.csv")

edges_164 = inner_join(df %>% filter(wg == 164),net,by = c("application_number" = "application_number")

colnames(edges_164) = c("from","to","art_unit")
edges_164 = drop_na(edges_164)

edges_172 = inner_join(df %>% filter(wg == 172),net,by = c("application_number" = "application_number")

colnames(edges_172) = c("from","to","art_unit")
edges_172 = drop_na(edges_172)
```

Create Nodes

```
edges = rbind(edges_164,edges_172)
node_ego = edges %>% select(from,art_unit) %>%rename(id=from)
node_alter = edges %>% select(to,art_unit)%>%rename(id=to)
nodes_all <-rbind.data.frame(node_ego, node_alter)

nodes = nodes_all %>% distinct(id)
```

```r
nodes = nodes %>% mutate(id = as.character(id))
```

Create Graph

```r
library(igraph)
```

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

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following objects are masked from 'package:lubridate':
##
##     %--%, union

## 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
```

```r
net_164 = graph_from_data_frame(d=edges_164, vertices=nodes, directed=TRUE)
net_164
```

```
## IGRAPH ffcb60d DN-- 382 1320 --
## + attr: name (v/c), art_unit (e/n)
## + edges from ffcb60d (vertex names):
##  [1] 91688->71059 91688->67669 97910->59738 97910->99004 97910->67669
##  [6] 75775->69583 75775->83794 75775->70306 75775->91151 75775->71534
## [11] 70204->72882 70204->94911 71120->65790 59338->72882 61757->65024
## [16] 61757->72882 60067->91747 60067->71087 60067->73722 60067->81365
## [21] 96963->72882 97910->65790 97910->59738 97910->99004 93839->71946
## [26] 74224->65024 74224->94911 96963->67657 87897->69583 87897->72882
## [31] 75775->69583 75775->83794 75775->70306 93839->67669 93839->71946
```

```
## [36] 93839->67669 93839->95981 75775->69583 75775->69583 75775->69583
## + ... omitted several edges
```

```
net_172 = graph_from_data_frame(d=edges_172, vertices=nodes, directed=TRUE)
```

Pick the mesure of centrality

1. Degree centrality is defined as the number of links incident upon a node
2. Eigenvector Centrality is an algorithm that measures the transitive influence of nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores.
3. Closeness centrality is a measure of the average shortest distance from each vertex to each other vertex
4. Betweenness centrality is a way of detecting the amount of influence a node has over the flow of information in a graph.

```
## Degree Centrality
V(net_164)$dc <- degree(net_164)
V(net_172)$dc <- degree(net_172)

## Eigenvector Centrality
V(net_164)$ec <- evcent(net_164)$vector
V(net_172)$ec <- evcent(net_172)$vector

## Closeness Centrality
V(net_164)$cc <- closeness(net_164)
V(net_172)$cc <- closeness(net_172)

## Betweenness Centrality
V(net_164)$bc <- betweenness(net_164)
V(net_172)$bc <- betweenness(net_172)
```

Plot the network based on centrality

```
library(ggraph)
library(ggplot2)
library(ggpubr)
# Degree Centrality
dc_164 = ggraph(net_164, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=dc), show.legend=T) + ggtitle("Degree Centrality 164")

# Eigenvector Centrality
ec_164<-ggraph(net_164, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=ec), show.legend=T) + ggtitle("Eigenvector Centrality 164")

# Closeness Centrality
cc_164<-ggraph(net_164, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=cc), show.legend=T) + ggtitle("Closeness Centrality 164")
```

```r
# Betweenness Centrality
bc_164<-ggraph(net_164, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=bc), show.legend=T) + ggtitle("Betwenness Centrality 164")
```

Centrality Scores

```r
centrality_164 <- data.frame(
                             id = V(net_164)$name,
                         degree      = V(net_164)$dc,
                         closeness   = V(net_164)$cc,
                         betweenness = V(net_164)$bc,
                         eigenvector = V(net_164)$ec)
head(centrality_164)
```

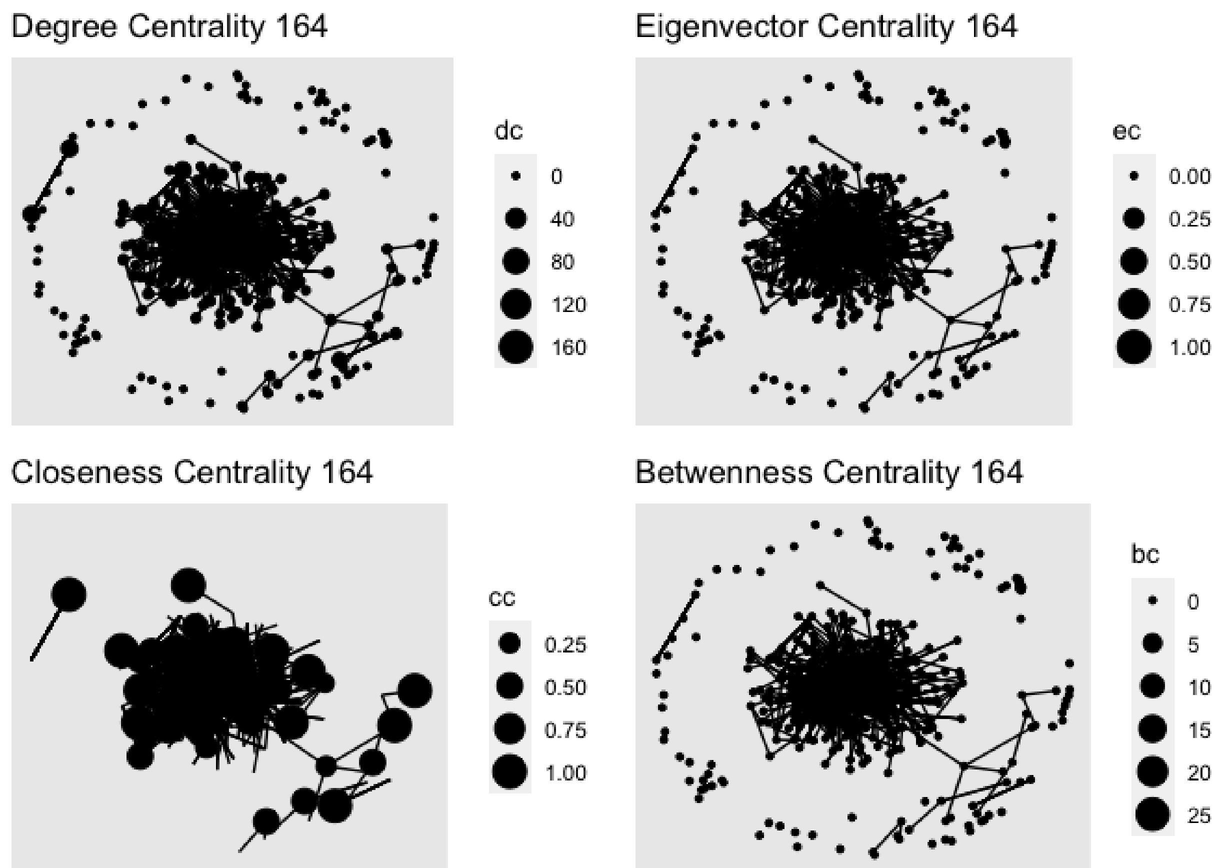
```
##      id degree  closeness betweenness  eigenvector
## 1 91688      2 0.50000000           0 0.0007440125
## 2 97910    170 0.01098901           0 1.0000000000
## 3 75775     74 0.05882353           0 0.0528936163
## 4 70204     50 0.14285714           0 0.2087178442
## 5 71120      1 1.00000000           0 0.0007229780
## 6 59338     17 0.07142857           0 0.0236891407
```

```r
centrality_172 <- data.frame(id = V(net_164)$name,
                         degree      = V(net_172)$dc,
                         closeness   = V(net_172)$cc,
                         betweenness = V(net_172)$bc,
                         eigenvector = V(net_172)$ec)
head(centrality_172)
```

```
##      id degree closeness betweenness  eigenvector
## 1 91688      0       NaN           0 1.650984e-17
## 2 97910      0       NaN           0 1.650984e-17
## 3 75775      0       NaN           0 1.650984e-17
## 4 70204      0       NaN           0 1.650984e-17
## 5 71120      0       NaN           0 1.650984e-17
## 6 59338      0       NaN           0 1.650984e-17
```

```r
ggarrange(dc_164,ec_164,cc_164,bc_164,ncol = 2, nrow = 2)
```

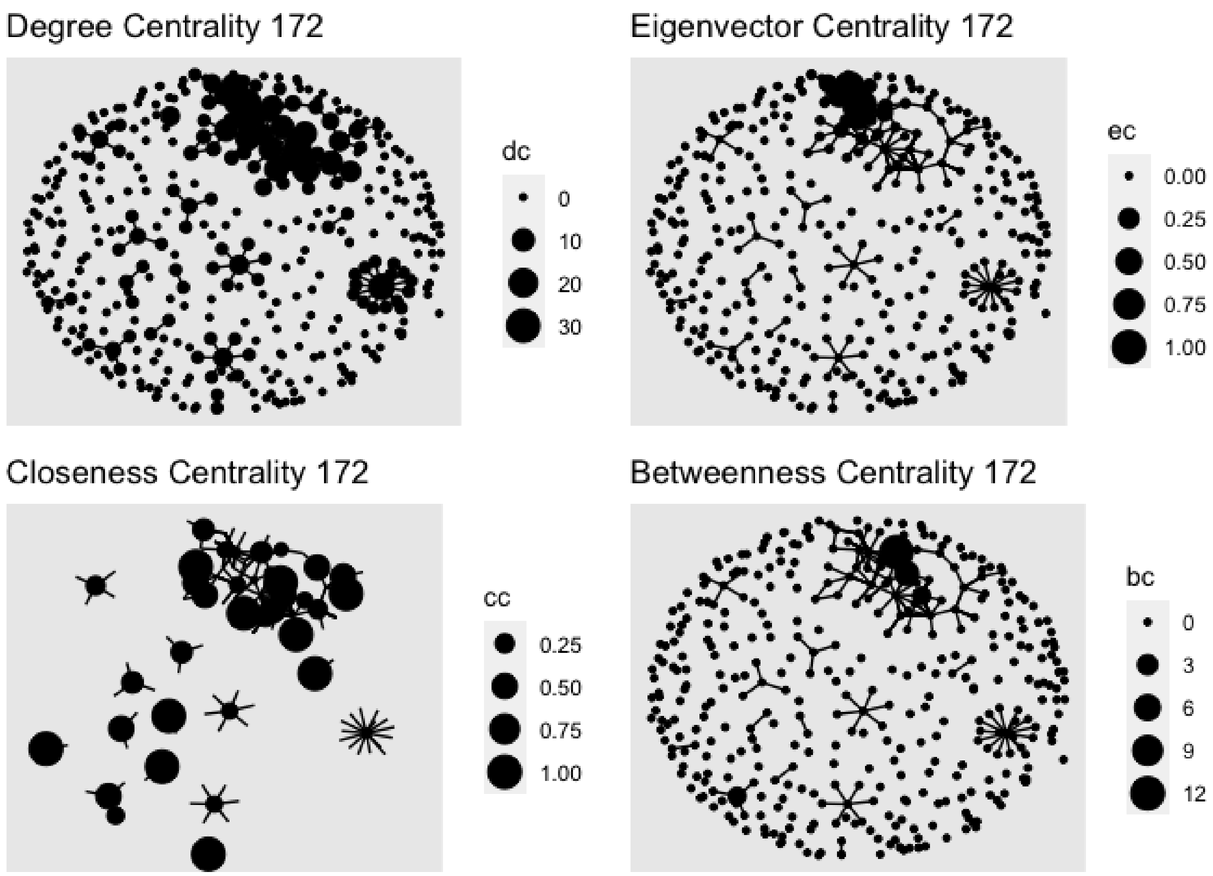
```
## Warning: Removed 293 rows containing missing values (geom_point).
```




```
dc_172 = ggraph(net_172, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=dc), show.legend=T) + ggtitle("Degree Centrality 172")

# Eigenvector Centrality
ec_172<-ggraph(net_172, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=ec), show.legend=T) + ggtitle("Eigenvector Centrality 172")

# Closness Centrality
cc_172<-ggraph(net_172, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=cc), show.legend=T) + ggtitle("Closeness Centrality 172")

# Betwenness Centrality
bc_172<-ggraph(net_172, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=bc), show.legend=T) + ggtitle("Betweenness Centrality 172")
```

```
ggarrange(dc_172,ec_172,cc_172,bc_172,ncol = 2, nrow = 2)
```

```
## Warning: Removed 345 rows containing missing values (geom_point).
```

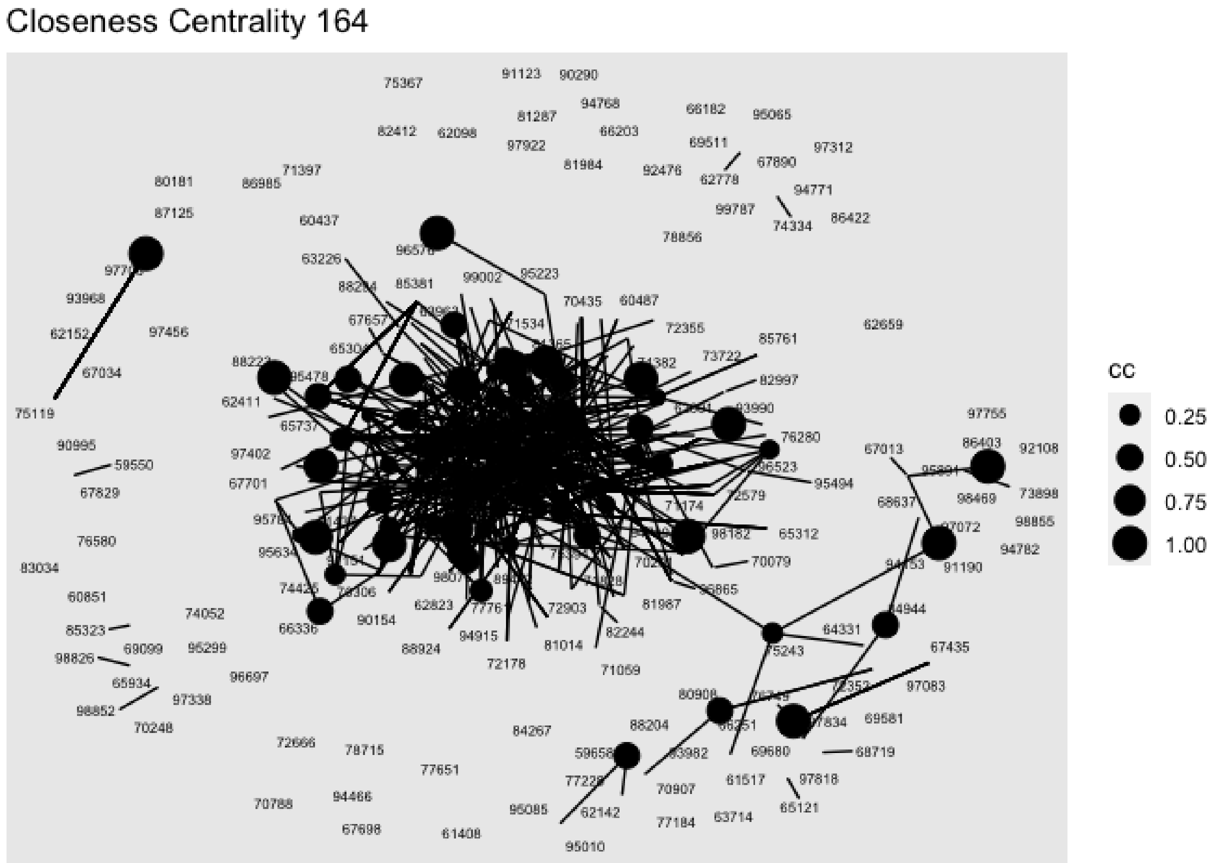


Based on the graph, seems like closeness centrality has clearer cluster center.

Characterize and discuss the relationship between centrality and other examiners'characteristics

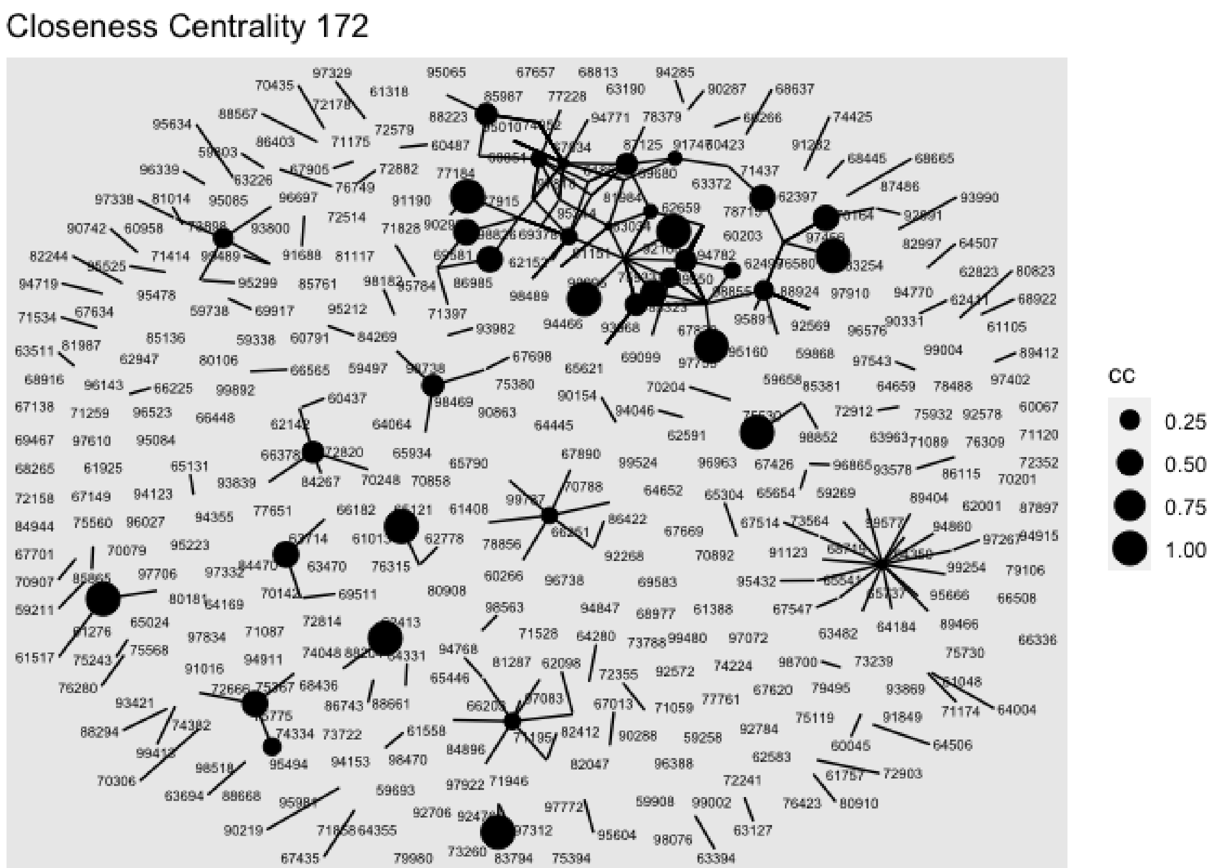
```
ggraph(net_164, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=cc), show.legend=T) +geom_node_text(aes(label = centrality_164$id), repel=TRU
```

```
## Warning: Removed 293 rows containing missing values (geom_point).

## Warning: ggrepel: 226 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```r
ggraph(net_172, layout="kk") +
  geom_edge_link()+
  geom_node_point(aes(size=cc), show.legend=T) +geom_node_text(aes(label = centrality_172$id), repel=TRU
```

```
## Warning: Removed 345 rows containing missing values (geom_point).

## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Gather all examiner characteristics

```
examiner = df %>% select(examiner_id,examiner_art_unit,gender,race,tenure)
examiner = distinct(examiner)
```

Examiner that are in group 164 and has the highest closeness centrality

```
max_cc_164 = max(centrality_164$closeness[!is.na(centrality_164$closeness)])
max_cc_164_id = centrality_164 %>% filter(centrality_164$closeness ==max_cc_164) %>%select(id)
max_cc_164_id = max_cc_164_id %>% mutate(id = as.numeric(id))
max_cc_164_info = examiner %>%filter(examiner_id == max_cc_164_id$id)

table(max_cc_164_info$gender)
```

```
##
## female   male
##    205    323
```

```
table(max_cc_164_info$race)
```

```
##
##    asian    black hispanic    white
##      138       32        2      356
```

Examiners that has higher closeness centrality in group 164, are more likely to be while male.

Examiner that are in group 172 and has the highest closeness centrality

```
max_cc_172 = max(centrality_172$closeness[!is.na(centrality_172$closeness)])
max_cc_172_id = centrality_172 %>% filter(centrality_172$closeness ==max_cc_172) %>%select(id)
max_cc_172_id = max_cc_172_id %>% mutate(id = as.numeric(id))
max_cc_172_info = examiner %>%filter(examiner_id == max_cc_172_id$id)

table(max_cc_172_info$gender)
```

```
##
## female   male
##     68    347
```

```
table(max_cc_172_info$race)
```

```
##
##    asian hispanic    white
##       49       46      320
```

The examiners that has higher closeness centrality in group 172 are mostly male comparing to group 164. Also, there are more Hispanic examiners that are influencial in this group.