Excercise 3 - Advice Networks at USPTO

Note: I received plenty help and guidance from my peer, Emery Dittmer, and my approaches to this exercise were the same as his.

Load data

Load the following data: + applications from `app_data_sample.parquet` + edges from `edges_sample.csv`

```
# change to your own path!
data_path <- "C:/Users/hugog/Desktop/Exercise 3/"
applications <- read_parquet(paste0(data_path,"app_data_sample.parquet"))
edges <- read_csv(paste0(data_path,"edges_sample.csv"))
```

```
## Rows: 32906 Columns: 4
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): application_number
## dbl  (2): ego_examiner_id, alter_examiner_id
## date (1): advice_date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
applications
```

```
## # A tibble: 2,018,477 × 16
##    applicat…¹ filing_d…² exami…³ exami…⁴ exami…⁵ exami…⁶ exami…⁷ uspc_…⁸ uspc_…⁹
##    <chr>      <date>     <chr>   <chr>   <chr>     <dbl>   <dbl> <chr>   <chr>
##  1 08284457   2000-01-26 HOWARD  JACQUE… V         96082    1764 508     273000
##  2 08413193   2000-10-11 YILDIR… BEKIR   L         87678    1764 208     179000
##  3 08531853   2000-05-17 HAMILT… CYNTHIA <NA>      63213    1752 430     271100
##  4 08637752   2001-07-20 MOSHER  MARY    <NA>      73788    1648 530     388300
##  5 08682726   2000-04-10 BARR    MICHAEL E         77294    1762 427     430100
##  6 08687412   2000-04-28 GRAY    LINDA   LAMEY     68606    1734 156     204000
##  7 08716371   2004-01-26 MCMILL… KARA    RENITA    89557    1627 424     401000
##  8 08765941   2000-06-23 FORD    VANESSA L         97543    1645 424     001210
##  9 08776818   2000-02-04 STRZEL… TERESA  E         98714    1637 435     006000
## 10 08809677   2002-02-20 KIM     SUN     U         65530    1723 210     645000
## # … with 2,018,467 more rows, 7 more variables: patent_number <chr>,
## #   patent_issue_date <date>, abandon_date <date>, disposal_type <chr>,
```

```
## #   appl_status_code <dbl>, appl_status_date <chr>, tc <dbl>, and abbreviated
## #   variable names ¹application_number, ²filing_date, ³examiner_name_last,
## #   ⁴examiner_name_first, ⁵examiner_name_middle, ⁶examiner_id,
## #   ⁷examiner_art_unit, ⁸uspc_class, ⁹uspc_subclass
```

```
edges
```

```
## # A tibble: 32,906 × 4
##    application_number advice_date ego_examiner_id alter_examiner_id
##    <chr>              <date>                <dbl>             <dbl>
##  1 09402488           2008-11-17            84356             66266
##  2 09402488           2008-11-17            84356             63519
##  3 09402488           2008-11-17            84356             98531
##  4 09445135           2008-08-21            92953             71313
##  5 09445135           2008-08-21            92953             93865
##  6 09445135           2008-08-21            92953             91818
##  7 09479304           2008-12-15            61767             69277
##  8 09479304           2008-12-15            61767             92446
##  9 09479304           2008-12-15            61767             66805
## 10 09479304           2008-12-15            61767             70919
## # … with 32,896 more rows
```

Get gender for examiners

```
library(gender)
```

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

```
examiner_names <- applications %>%
  distinct(examiner_name_first)
examiner_names
```

```
## # A tibble: 2,595 × 1
##    examiner_name_first
##    <chr>
##  1 JACQUELINE
##  2 BEKIR
##  3 CYNTHIA
##  4 MARY
##  5 MICHAEL
##  6 LINDA
##  7 KARA
```

```
##  8 VANESSA
##  9 TERESA
## 10 SUN
## # … with 2,585 more rows
```

Get a table of names and gender

```
examiner_names_gender <- examiner_names %>%
  do(results = gender(.$examiner_name_first, method = "ssa")) %>%
  unnest(cols = c(results), keep_empty = TRUE) %>%
  select(
    examiner_name_first = name,
    gender,
    proportion_female
  )
examiner_names_gender
```

```
## # A tibble: 1,822 × 3
##    examiner_name_first gender proportion_female
##    <chr>               <chr>              <dbl>
##  1 AARON               male              0.0082
##  2 ABDEL               male              0
##  3 ABDOU               male              0
##  4 ABDUL               male              0
##  5 ABDULHAKIM          male              0
##  6 ABDULLAH            male              0
##  7 ABDULLAHI           male              0
##  8 ABIGAIL             female            0.998
##  9 ABIMBOLA            female            0.944
## 10 ABRAHAM             male              0.0031
## # … with 1,812 more rows
```

```
# remove extra colums from the gender table
examiner_names_gender <- examiner_names_gender %>%
  select(examiner_name_first, gender)
# joining gender back to the dataset
applications <- applications %>%
  left_join(examiner_names_gender, by = "examiner_name_first")
# cleaning up
rm(examiner_names)
rm(examiner_names_gender)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  4493926 240.1    7506837 401.0  4911510 262.4
```

```
## Vcells 49586412 378.4   95514760 728.8 79902128 609.7
```

```
library(wru)
```

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

```
examiner_surnames <- applications %>%
  select(surname = examiner_name_last) %>%
  distinct()
examiner_surnames
```

```
## # A tibble: 3,806 × 1
##    surname
##    <chr>
##  1 HOWARD
##  2 YILDIRIM
##  3 HAMILTON
##  4 MOSHER
##  5 BARR
##  6 GRAY
##  7 MCMILLIAN
##  8 FORD
##  9 STRZELECKA
## 10 KIM
## # … with 3,796 more rows
```

We'll follow the instructions for the package outlined here https://github.com/kosukeimai/wru.

NOTE: I was getting errors running the original code block for examiner_race. I tried updating packages, and debugging for a long time but I believe it is my computer's software/environment setup preventing me. I asked for the csv output from a peer and am importing it instead. Original code bloack: examiner_race <- predict_race(voter.file = examiner_surnames, surname.only = T) %>% as_tibble() examiner_race

```
examiner_race <- read_csv(paste0(data_path,"examiner_race.csv"))
```

```
## Rows: 3806 Columns: 6
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): surname
## dbl (5): pred.whi, pred.bla, pred.his, pred.asi, pred.oth
##
```

```
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
examiner_race
```

```
## # A tibble: 3,806 × 6
##    surname    pred.whi pred.bla pred.his pred.asi pred.oth
##    <chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1 HOWARD       0.597   0.295    0.0275   0.00690   0.0741
##  2 YILDIRIM     0.807   0.0273   0.0694   0.0165    0.0798
##  3 HAMILTON     0.656   0.239    0.0286   0.00750   0.0692
##  4 MOSHER       0.915   0.00425  0.0291   0.00917   0.0427
##  5 BARR         0.784   0.120    0.0268   0.00830   0.0615
##  6 GRAY         0.640   0.252    0.0281   0.00748   0.0724
##  7 MCMILLIAN    0.322   0.554    0.0212   0.00340   0.0995
##  8 FORD         0.576   0.320    0.0275   0.00621   0.0697
##  9 STRZELECKA   0.472   0.171    0.220    0.0825    0.0543
## 10 KIM          0.0169  0.00282  0.00546  0.943     0.0319
## # … with 3,796 more rows
```

```
examiner_race <- examiner_race %>%
  mutate(max_race_p = pmax(pred.asi, pred.bla, pred.his, pred.oth, pred.whi)) %>%
  mutate(race = case_when(
    max_race_p == pred.asi ~ "Asian",
    max_race_p == pred.bla ~ "black",
    max_race_p == pred.his ~ "Hispanic",
    max_race_p == pred.oth ~ "other",
    max_race_p == pred.whi ~ "white",
    TRUE ~ NA_character_
  ))
examiner_race
```

```
## # A tibble: 3,806 × 8
##    surname    pred.whi pred.bla pred.his pred.asi pred.oth max_race_p race
##    <chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>      <dbl> <chr>
##  1 HOWARD       0.597   0.295    0.0275   0.00690   0.0741      0.597 white
##  2 YILDIRIM     0.807   0.0273   0.0694   0.0165    0.0798      0.807 white
##  3 HAMILTON     0.656   0.239    0.0286   0.00750   0.0692      0.656 white
##  4 MOSHER       0.915   0.00425  0.0291   0.00917   0.0427      0.915 white
##  5 BARR         0.784   0.120    0.0268   0.00830   0.0615      0.784 white
##  6 GRAY         0.640   0.252    0.0281   0.00748   0.0724      0.640 white
##  7 MCMILLIAN    0.322   0.554    0.0212   0.00340   0.0995      0.554 black
##  8 FORD         0.576   0.320    0.0275   0.00621   0.0697      0.576 white
##  9 STRZELECKA   0.472   0.171    0.220    0.0825    0.0543      0.472 white
```

```
## 10 KIM          0.0169  0.00282  0.00546  0.943     0.0319      0.943 Asian
## # … with 3,796 more rows
```

Join the data back to the applications table.

```
# removing extra columns
examiner_race <- examiner_race %>%
  select(surname,race)
applications <- applications %>%
  left_join(examiner_race, by = c("examiner_name_last" = "surname"))
rm(examiner_race)
rm(examiner_surnames)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  4578427 244.6    7506837 401.0  6326878 337.9
## Vcells 51712109 394.6   95514760 728.8 94916812 724.2
```

Summary Statistics and Plots

Pre-Processing

```
person_level_data <- applications %>%
  group_by(examiner_id) %>%
  summarise(
    art_unit = min(examiner_art_unit, na.rm = TRUE),
    gender = min(gender, na.rm = TRUE),
    race = min(race,na.rm=TRUE)) %>%
  mutate(
    tc = floor(art_unit/100)*100,
    work_group = as.factor(floor(art_unit/10)*10)
  ) %>%
  filter(!is.na(gender) & !is.na(race)) # dropping all records where we don't know the gender
```

```
## Warning: There were 800 warnings in `summarise()`.
## The first warning was:
## ℹ In argument: `gender = min(gender, na.rm = TRUE)`.
## ℹ In group 3: `examiner_id = 59030`.
## Caused by warning in `min()`:
```

```
## ! no non-missing arguments, returning NA
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 799 remaining warnings.
```

```
#Grouping by work unit
work_unit_level_data <-person_level_data %>%
  group_by(work_group,race,gender) %>%
  summarize(
    n=n()
  )
```

```
## `summarise()` has grouped output by 'work_group', 'race'. You can override
## using the `.groups` argument.
```

```
#aggregated by total number of people in work group
work_unit_aggregated <- work_unit_level_data %>%
  group_by(work_group) %>%
  summarize(
    n=sum(n)
  ) %>%
  arrange (desc(n))
```

Gender accross the two work groups with most examiners: 2130 & 1610

```
subset_app_data <- person_level_data %>%
  #here we make sure on ly the top 2 work groups are picked
  filter(work_group %in% head(work_unit_aggregated$work_group,2)) %>%
  mutate(race = race, gender =gender) %>%
  select(gender, race, work_group)

subset_app_data %>%
  count(gender) %>%
  mutate(pct = n/sum(n))
```

```
## # A tibble: 2 × 3
##   gender     n   pct
##   <chr>  <int> <dbl>
## 1 female   160 0.346
## 2 male     303 0.654
```

Summary of Gender & Race in Both Groups 2130 & 1610

```
subset_app_data %>%
  group_by(work_group) %>%
  count(gender) %>%
  mutate(pct = n/sum(n))
```

```
## # A tibble: 4 × 4
## # Groups:   work_group [2]
##   work_group gender     n   pct
##   <fct>      <chr>  <int> <dbl>
## 1 1610       female   108 0.478
## 2 1610       male     118 0.522
## 3 2130       female    52 0.219
## 4 2130       male     185 0.781
```

```
subset_app_data %>%
  group_by(work_group) %>%
  count(race) %>%
  mutate(pct = n/sum(n))
```

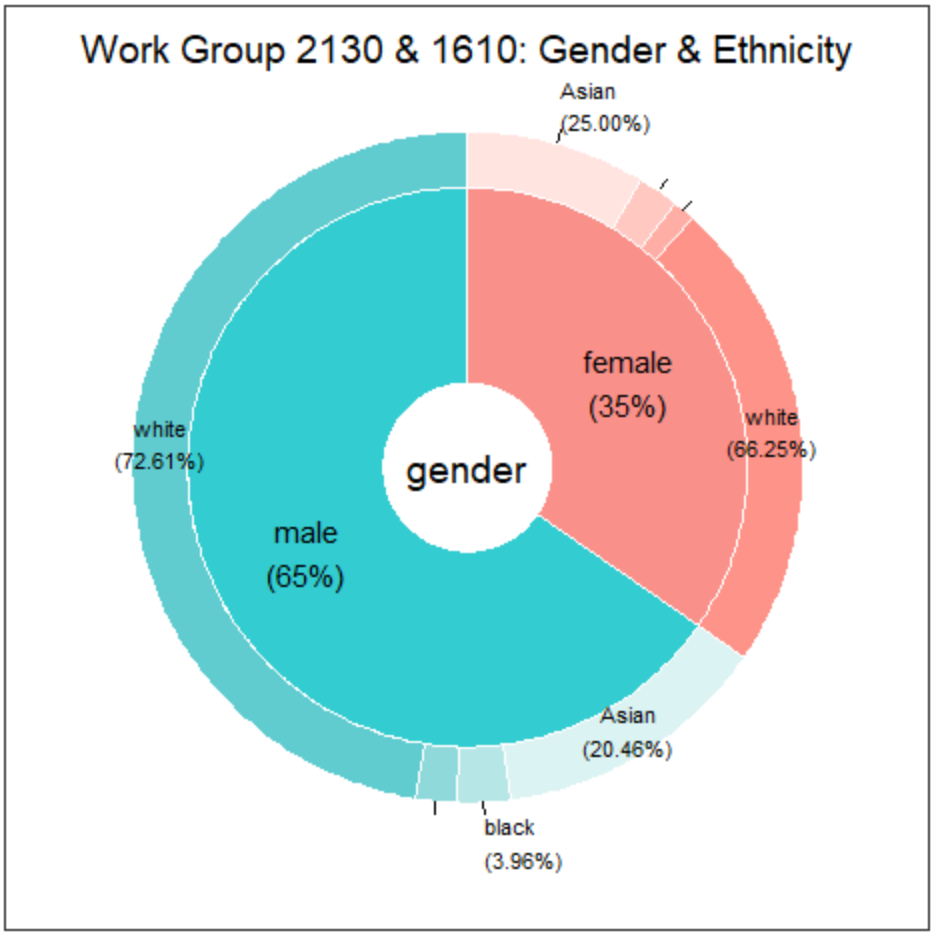
```
## # A tibble: 8 × 4
## # Groups:   work_group [2]
##   work_group race         n    pct
##   <fct>      <chr>    <int>  <dbl>
## 1 1610       Asian       33 0.146
## 2 1610       Hispanic     5 0.0221
## 3 1610       black        6 0.0265
## 4 1610       white      182 0.805
## 5 2130       Asian       69 0.291
## 6 2130       Hispanic     9 0.0380
## 7 2130       black       15 0.0633
## 8 2130       white      144 0.608
```

```
#install.packages('webr')
library(webr)
```

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

```
PieDonut(subset_app_data, aes(gender,race), title = "Work Group 2130 & 1610: Gender & Ethnicity
```

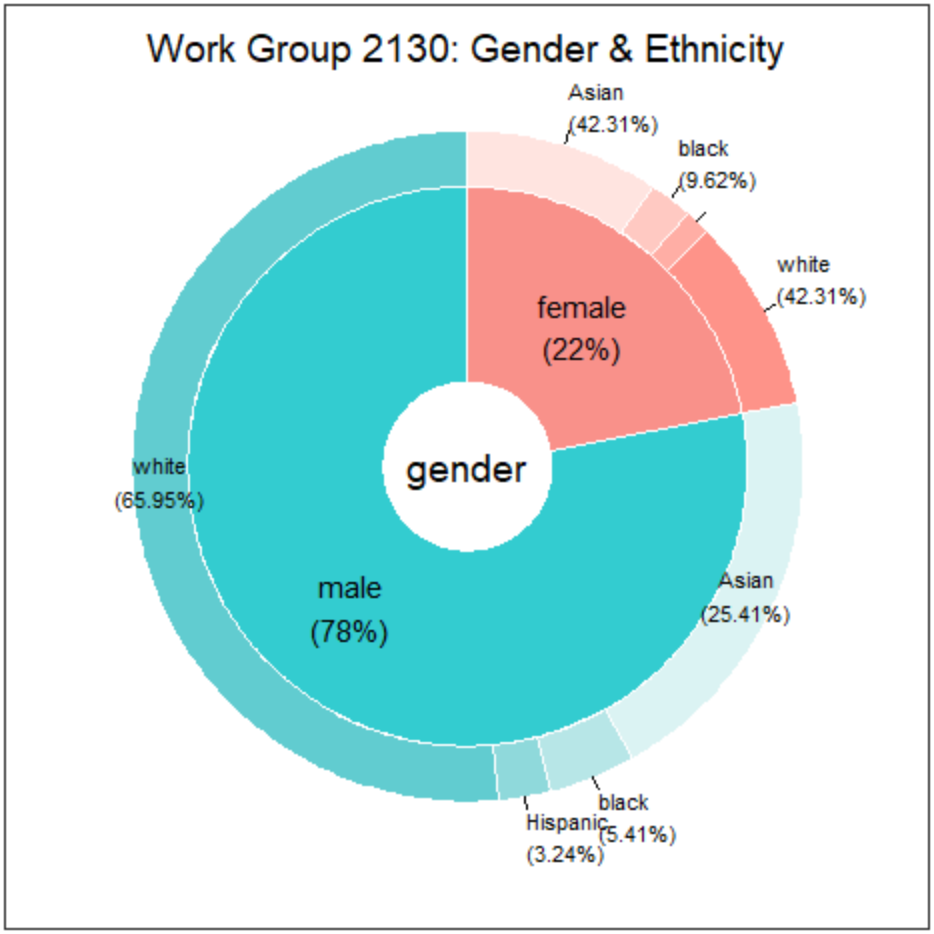
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## ℹ The deprecated feature was likely used in the webr package.
##   Please report the issue at <https://github.com/cardiomoon/webr/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Summary of Gender & Race in Work Group 2130

```
subset_app_data_2130 <- subset_app_data %>% filter(work_group==2130)
PieDonut(subset_app_data_2130, aes(gender,race), title = "Work Group 2130: Gender & Ethnicity",
```

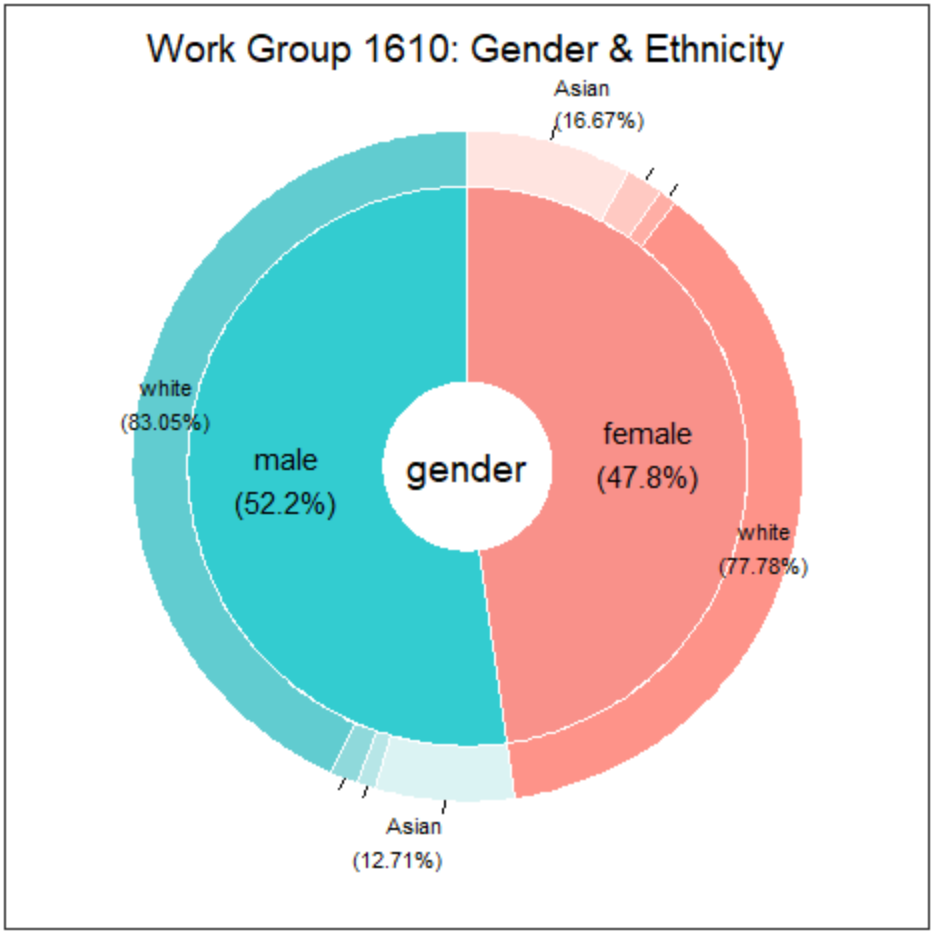
```
## Warning in geom_arc_bar(aes_string(x0 = "x", y0 = "y", r0 = as.character(r1), :
## Ignoring unknown aesthetics: explode
```



Summary of Gender & Race in Work Group 1610

```
subset_app_data_1610 <- subset_app_data %>% filter(work_group==1610)
PieDonut(subset_app_data_1610, aes(gender,race), title = "Work Group 1610: Gender & Ethnicity",
```

```
## Warning in geom_arc_bar(aes_string(x0 = "x", y0 = "y", r0 = as.character(r1), :
## Ignoring unknown aesthetics: explode
```



Netwrok Graph - Pre-Processing: Edges & Nodes

```
#copy data as best practice
edges_full <- edges
edges <- edges_full

subset_exam_id <- person_level_data %>%
  filter(work_group %in% head(work_unit_aggregated$work_group,2)) %>%
  select(examiner_id,work_group) %>%
  drop_na()

#create the edges
edges <- edges %>%
  filter(ego_examiner_id %in% subset_exam_id$examiner_id)%>%
  drop_na() %>%
  mutate(from=ego_examiner_id,to=alter_examiner_id) %>%
  select(from, to)

nodes_all <-as.data.frame(do.call(rbind,append(as.list(edges$from),as.list(edges$to))))

# create the nodes
nodes_all <- nodes_all %>%
  mutate(id=V1) %>%
  select(id) %>%
  distinct(id) %>%
```

```
  drop_na()
nodes <- nodes_all
```

Degree Centrality

```
library(tidyverse)
library(igraph)
```

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

## 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: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: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
```

```
g <- igraph::graph_from_data_frame(edges, vertices = nodes) %>% as_tbl_graph(directed=TRUE)
g <- g %>%
  activate(nodes) %>%
  mutate(degree = centrality_degree()) %>%
  activate(edges)

tg_nodes <-
  g %>%
  activate(nodes) %>%
  data.frame() %>%
  arrange(desc(degree)) %>%
  rename(Centrality_Degree=degree) %>%
  mutate(name=as.integer(name))

nodes_all <- nodes_all %>%
  left_join(tg_nodes,by=c("id"="name"))

remove(g,tg_nodes)
```

Closeness centrality

Indicates who is at the heart of a social network. Node with high closeness centrality also tends to be close to most people. That means the person will be in a good position to hear from most friends of friends. They will be a good source of second hand information since it can reach them quite easily.

```
g <- igraph::graph_from_data_frame(edges, vertices = nodes) %>% as_tbl_graph(directed=TRUE)

g <- g %>%
  activate(nodes) %>%
  mutate(degree = centrality_closeness()) %>%
  activate(edges)

tg_nodes <-
  g %>%
  activate(nodes) %>%
  data.frame() %>%
  arrange(desc(degree)) %>%
  rename(Centrality_Closeness=degree) %>%
  mutate(name=as.integer(name))

nodes_all <- nodes_all %>%
```

```
  left_join(tg_nodes,by=c("id"="name"))
remove(g,tg_nodes)
```

Betweenness centrality

How important the node is to the flow of information through a network - describes people who connect social circles. Node with high betweenness is likely to yield insights about what both groups are doing and what is going on between those two groups.

```
g <- igraph::graph_from_data_frame(edges, vertices = nodes) %>% as_tbl_graph(directed=TRUE)

g <- g %>%
  activate(nodes) %>%
  mutate(degree = centrality_betweenness()) %>%
  activate(edges)

tg_nodes <-
  g %>%
  activate(nodes) %>%
  data.frame() %>%
  arrange(desc(degree)) %>%
  rename(Centrality_Betweenness=degree) %>%
  mutate(name=as.integer(name))

nodes_all <- nodes_all %>%
  left_join(tg_nodes,by=c("id"="name"))
remove(g,tg_nodes)
```

Visualize Networkk Graph with Centrality Scores (Zoom Into the Graph to see scores at each node for each Centrality score)

Note: This Chunk would not knit so I am not “chunking” it

nodes <- nodes_all %>% left_join(subset_exam_id,by=c(“id”=“examiner_id”)) %>% mutate(label = paste(“Examiner:”,id,“”, “Centrality Degre:”,format(Centrality_Degree, digits = 2),“”, “Closenness:”,format(Centrality_Closeness, digits = 2),“”, “Betweenness:”,format(Centrality_Betweenness, digits = 2),“”, sep = " "), group=work_group) %>% mutate(font.size = 12) %>% drop_na()

visNetwork(nodes, edges)%>% visLegend() %>% visEdges(arrows =“to”)%>% visEdges(arrows =“from”)

Vizualize Netwrok with Igraph (Not Very Readable)

```
net <- igraph::graph_from_data_frame(edges, vertices = nodes_all) %>% as_tbl_graph(directed=TRU
plot(net, edge.arrow.size=.4,vertex.label.cex=.4,vertex.label.dist=1,vertex.size=4)
```