Exercise 4r

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/csg20/OneDrive/Desktop/Git/ORGB-672/"
app_data_sample <- read_parquet(paste0(data_path,"app_data_sample.parquet"))

app_data_sample
```

```
## # A tibble: 2,018,477 x 16
##    application_number filing_date examiner_name_last examiner_name_first
##    <chr>              <date>      <chr>              <chr>
##  1 08284457           2000-01-26  HOWARD             JACQUELINE
##  2 08413193           2000-10-11  YILDIRIM           BEKIR
##  3 08531853           2000-05-17  HAMILTON           CYNTHIA
##  4 08637752           2001-07-20  MOSHER             MARY
##  5 08682726           2000-04-10  BARR               MICHAEL
##  6 08687412           2000-04-28  GRAY               LINDA
##  7 08716371           2004-01-26  MCMILLIAN          KARA
##  8 08765941           2000-06-23  FORD               VANESSA
##  9 08776818           2000-02-04  STRZELECKA         TERESA
## 10 08809677           2002-02-20  KIM                SUN
## # i 2,018,467 more rows
## # i 12 more variables: examiner_name_middle <chr>, examiner_id <dbl>,
## #   examiner_art_unit <dbl>, uspc_class <chr>, uspc_subclass <chr>,
## #   patent_number <chr>, patent_issue_date <date>, abandon_date <date>,
## #   disposal_type <chr>, appl_status_code <dbl>, appl_status_date <chr>,
## #   tc <dbl>
```

Get gender for examiners

We'll get gender based on the first name of the examiner, which is recorded in the field `examiner_name_first`. We'll use library `gender` for that, relying on a modified version of their own example.

Note that there are over 2 million records in the applications table – that's because there are many records for each examiner, as many as the number of applications that examiner worked on during this time frame. Our first step therefore is to get all *unique* names in a separate list `examiner_names`. We will then guess gender for each one and will join this table back to the original dataset. So, let's get names without repetition:

```
library(gender)
```

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

```r
#install_genderdata_package() # only run this line the first time you use the package, to get data for

# get a list of first names without repetitions
examiner_names <- app_data_sample %>%
  distinct(examiner_name_first)

examiner_names
```

```
## # A tibble: 2,595 x 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
## # i 2,585 more rows
```

Now let's use function `gender()` as shown in the example for the package to attach a gender and probability to each name and put the results into the table `examiner_names_gender`. Note that the first time you run this code, you need to say "Yes" in the console to download the gender data.

```r
# 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 x 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
## # i 1,812 more rows
```

Finally, let's join that table back to our original applications data and discard the temporary tables we have just created to reduce clutter in our environment.

```
# 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 <- app_data_sample %>%
  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  4483811 239.5    8274112 441.9  4911295 262.3
## Vcells 77686162 592.7  114468838 873.4 79743363 608.4
```

Guess the examiner's race

We'll now use package `wru` to estimate likely race of an examiner. Just like with gender, we'll get a list of unique names first, only now we are using surnames.

```
library(wru)
```

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

```
##
## Please cite as:
##
## Khanna K, Bertelsen B, Olivella S, Rosenman E, Rossell Hayes A, Imai K
## (2024). _wru: Who are You? Bayesian Prediction of Racial Category Using
## Surname, First Name, Middle Name, and Geolocation_. R package version
## 3.0.1, <https://CRAN.R-project.org/package=wru>.
##
## Note that wru 2.0.0 uses 2020 census data by default.
## Use the argument 'year = "2010"', to replicate analyses produced with earlier package versions.
```

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

examiner_surnames
```

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

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

```
examiner_race <- predict_race(voter.file = examiner_surnames, surname.only = T) %>%
  as_tibble()
```

```
## Predicting race for 2020
```

```
## Warning: Unknown or uninitialised column: ‘state‘.
```

```
## Proceeding with last name predictions...
```

```
## i All local files already up-to-date!
```

```
## 701 (18.4%) individuals’ last names were not matched.
```

```
examiner_race
```

```
## # A tibble: 3,806 x 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
## # i 3,796 more rows
```

As you can see, we get probabilities across five broad US Census categories: white, black, Hispanic, Asian and other. (Some of you may correctly point out that Hispanic is not a race category in the US Census, but these are the limitations of this package.)

Our final step here is to pick the race category that has the highest probability for each last name and then join the table back to the main applications table. See this example for comparing values across columns: https://www.tidyverse.org/blog/2020/04/dplyr-1-0-0-rowwise/. And this one for `case_when()` function: https://dplyr.tidyverse.org/reference/case_when.html.

```
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 x 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
## # i 3,796 more rows
```

Let's 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   4683849 250.2    8274112  441.9   7482361 399.7
## Vcells  80026706 610.6  139317646 1063.0 116403477 888.1
```

Examiner's tenure

To figure out the timespan for which we observe each examiner in the applications data, let's find the first and the last observed date for each examiner. We'll first get examiner IDs and application dates in a separate table, for ease of manipulation. We'll keep examiner ID (the field `examiner_id`), and earliest and latest dates for each application (`filing_date` and `appl_status_date` respectively). We'll use functions in package `lubridate` to work with date and time values.

```
library(lubridate) # to work with dates

examiner_dates <- applications %>%
  select(examiner_id, filing_date, appl_status_date)

examiner_dates
```

```
## # A tibble: 2,018,477 x 3
##    examiner_id filing_date appl_status_date
##          <dbl> <date>      <chr>
##  1       96082 2000-01-26  30jan2003 00:00:00
##  2       87678 2000-10-11  27sep2010 00:00:00
##  3       63213 2000-05-17  30mar2009 00:00:00
##  4       73788 2001-07-20  07sep2009 00:00:00
##  5       77294 2000-04-10  19apr2001 00:00:00
##  6       68606 2000-04-28  16jul2001 00:00:00
##  7       89557 2004-01-26  15may2017 00:00:00
##  8       97543 2000-06-23  03apr2002 00:00:00
##  9       98714 2000-02-04  27nov2002 00:00:00
## 10       65530 2002-02-20  23mar2009 00:00:00
## # i 2,018,467 more rows
```

The dates look inconsistent in terms of formatting. Let's make them consistent. We'll create new variables `start_date` and `end_date`.

```
examiner_dates <- examiner_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))
```

Let's now identify the earliest and the latest date for each examiner and calculate the difference in days, which is their tenure in the organization.

```
examiner_dates <- examiner_dates %>%
  group_by(examiner_id) %>%
  summarise(
    earliest_date = min(start_date, na.rm = TRUE),
    latest_date = max(end_date, na.rm = TRUE),
    tenure_days = interval(earliest_date, latest_date) %/% days(1)
    ) %>%
  filter(year(latest_date)<2018)

examiner_dates
```

```
## # A tibble: 5,625 x 4
##    examiner_id earliest_date latest_date tenure_days
##          <dbl> <date>        <date>            <dbl>
##  1       59012 2004-07-28    2015-07-24         4013
##  2       59025 2009-10-26    2017-05-18         2761
##  3       59030 2005-12-12    2017-05-22         4179
##  4       59040 2007-09-11    2017-05-23         3542
##  5       59052 2001-08-21    2007-02-28         2017
##  6       59054 2000-11-10    2016-12-23         5887
##  7       59055 2004-11-02    2007-12-26         1149
##  8       59056 2000-03-24    2017-05-22         6268
##  9       59074 2000-01-31    2017-03-17         6255
## 10       59081 2011-04-21    2017-05-19         2220
## # i 5,615 more rows
```

Joining back to the applications data.

```
applications <- applications %>%
  left_join(examiner_dates, by = "examiner_id")

rm(examiner_dates)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  4690308 250.5   15263328  815.2  15263328  815.2
## Vcells 92388381 704.9  167261175 1276.2 167194307 1275.6
```

1. Create variable for application processing time 'app_proc_time' that measures the number of days (or weeks) from application filing date, until the final decision on it (patented or abandoned)

First, find the columns I am going to use

```
colnames(applications)
```

```
##  [1] "application_number"   "filing_date"          "examiner_name_last"
##  [4] "examiner_name_first"  "examiner_name_middle" "examiner_id"
##  [7] "examiner_art_unit"    "uspc_class"           "uspc_subclass"
## [10] "patent_number"        "patent_issue_date"    "abandon_date"
## [13] "disposal_type"        "appl_status_code"     "appl_status_date"
## [16] "tc"                   "gender"               "race"
## [19] "earliest_date"        "latest_date"          "tenure_days"
```

```
# "filing_date", "patent_issue_date", "abandon_date"
```

create the variable 'app_proc_time'

```
library(dplyr)

df = applications

df <- df %>%
  # Ensure the date columns are in the Date format
  mutate(
    filing_date = as.Date(filing_date),
    patent_issue_date = as.Date(patent_issue_date),
    abandon_date = as.Date(abandon_date),
    disposal_type = as.character(disposal_type)
  ) %>%
  # First, drop rows where both patent_issue_date and abandon_date are NA
  filter(!(is.na(patent_issue_date) & is.na(abandon_date))) %>%
  # Then calculate the processing time in days based on disposal_type
  mutate(
    app_proc_time = case_when(
      disposal_type == "ISS" ~ as.numeric(patent_issue_date - filing_date),
      TRUE ~ as.numeric(abandon_date - filing_date)
    )
  ) %>%
  # Finally, refine the dropping based on disposal_type and relevant dates
  filter(!(disposal_type == "ISS" & is.na(patent_issue_date)) & !(disposal_type != "ISS" & is.na(abandon
```

```
df %>% head(10)
```

```
## # A tibble: 10 x 22
##    application_number filing_date examiner_name_last examiner_name_first
##    <chr>              <date>      <chr>              <chr>
##  1 08284457           2000-01-26  HOWARD             JACQUELINE
##  2 08413193           2000-10-11  YILDIRIM           BEKIR
##  3 08531853           2000-05-17  HAMILTON           CYNTHIA
##  4 08637752           2001-07-20  MOSHER             MARY
##  5 08682726           2000-04-10  BARR               MICHAEL
##  6 08687412           2000-04-28  GRAY               LINDA
##  7 08765941           2000-06-23  FORD               VANESSA
##  8 08776818           2000-02-04  STRZELECKA         TERESA
##  9 08809677           2002-02-20  KIM                SUN
## 10 08836939           2000-06-13  WOOD               ELIZABETH
## # i 18 more variables: examiner_name_middle <chr>, examiner_id <dbl>,
## #   examiner_art_unit <dbl>, uspc_class <chr>, uspc_subclass <chr>,
## #   patent_number <chr>, patent_issue_date <date>, abandon_date <date>,
## #   disposal_type <chr>, appl_status_code <dbl>, appl_status_date <chr>,
## #   tc <dbl>, gender <chr>, race <chr>, earliest_date <date>,
## #   latest_date <date>, tenure_days <dbl>, app_proc_time <dbl>
```

realize there are some negative values (outliers)

```
sum(df$app_proc_time < 0)
```

```
## [1] 35
```

```
df <- df %>%
  filter(app_proc_time >= 0) %>%
  filter(!is.na(app_proc_time))
```

2. Use linear regression models `lm()` to estimate the relationship between centrality and `app_proc_time`

Create advice networks from ‘edges_sample‘ and calculate centrality scores for examiners

```r
#read data
edge = read.csv('C:/Users/csg20/OneDrive/Desktop/Git/ORGB-672/Ex3/edges_sample.csv')
```

```r
library(igraph)
```

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

```
##
## 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
```

```r
edge = drop_na(edge)
```

create networks

```r
g <- graph_from_data_frame(d = edge[, c("ego_examiner_id", "alter_examiner_id")], directed = TRUE)
```

Calculate centrality

```
# Degree Centrality
degree_centrality <- degree(g, mode = "all")

# Betweenness Centrality
betweenness_centrality <- betweenness(g, directed = TRUE)

# Closeness Centrality
closeness_centrality <- closeness(g, mode = "all")
```

merge score back to df

```
# Create a data frame of centrality measures
centrality_measures <- data.frame(
  examiner_id = V(g)$name,
  degree = degree_centrality,
  betweenness = betweenness_centrality,
  closeness = closeness_centrality
)

# Ensure examiner_id is the correct type for joining
df$examiner_id <- as.character(df$examiner_id)
centrality_measures$examiner_id <- as.character(centrality_measures$examiner_id)
```

Filter `df` to Keep Rows with `examiner_id` Present in `centrality_measures` and merge

```
df <- df %>%
  filter(examiner_id %in% centrality_measures$examiner_id)

df <- merge(df, centrality_measures, by = "examiner_id")
```

Build the model

Variables that could potentially affect both the centrality of examiners in the advice network and the processing time of patent applications

1. **`gender`**: Gender may influence networking behaviors and patterns within organizations, potentially affecting both centrality and processing times.

2. **`race`**: Similar to gender, racial dynamics within organizations can influence how individuals are positioned within informal networks and could impact processing times due to diversity in collaboration styles.

3. **`tenure_days`**: The length of time an examiner has been with the organization could affect both their centrality in the network (with more tenured examiners potentially having more connections) and their efficiency or speed in processing applications.

4. **`examiner_art_unit`**: Different art units may have varying average processing times due to the complexity of the applications they handle and the internal dynamics of the unit, which can also affect the centrality of examiners within and across these units.

5. **`uspc_class`**: The technological area of the application, represented by its classification, might influence the processing time due to varying levels of complexity and the examiner's familiarity with the subject matter. These factors might also correlate with an examiner's centrality if certain areas require or facilitate more collaboration and advice sharing.

df is too big to process, thus do sampling based on examiner_id - get 5 instances for each examiner_id

```
df_sampled <- df %>%
  group_by(examiner_id) %>%
  slice_head(n = 5) %>%
  ungroup()
```

there are too many categories in **`examiner_art_unit`**,**`uspc_class`**

1600	Biotechnology and Organic Chemistry
1700	Chemical and Materials Engineering
2100	Computer Architecture and Software
2400	Networking, Multiplexing, Cable, and Security
2600	Communications
2800	Semiconductors/Memory, Circuits/Measuring and Testing, Optics/Photocopying, Printing/Measuring and Testing
2900	Design
3600	Transportation, Construction, Electronic Commerce, Agriculture, National Security and License and Review
3700	Mechanical Engineering, Manufacturing and Medical Devices/Processes
3900	Reexam/Abandonments

based on the info shown above, change the examinor_art_unit to simplified categories

```
library(stringr)

df_sampled <- df_sampled %>%
  mutate(
    examiner_art_unit = as.character(examiner_art_unit),
    examiner_art_unit = str_sub(examiner_art_unit, 1, 2), # Keep the first two digits
    examiner_art_unit = paste0(examiner_art_unit, "00"), # Append '00' at the end
    examiner_art_unit = as.factor(examiner_art_unit) # Convert to factor
  )

# View the changes
head(df_sampled$examiner_art_unit)
```

```
## [1] 2400 2400 2400 2400 2400 2400
## Levels: 1600 1700 2100 2400
```

for **`uspc_class`** I choose to change it based on ranges

```
df_sampled <- df_sampled %>%
  mutate(
    uspc_class = as.numeric(uspc_class), # Ensure it's numeric
    uspc_class_range = cut(
      uspc_class,
      breaks = seq(0, 900, by = 100), # Define the ranges
      include.lowest = TRUE, # Include the lowest value
      labels = paste(seq(0, 800, by = 100), seq(100, 900, by = 100) - 1, sep = "-"), # Define the label
    right = FALSE # Make the interval left-closed [a,b)
  )
)

df_sampled$uspc_class_range <- addNA(df_sampled$uspc_class_range, ifany = TRUE) # Converts NA to a fact
levels(df_sampled$uspc_class_range)[is.na(levels(df_sampled$uspc_class_range))] <- "unknown"

head(df_sampled$uspc_class_range)
```

```
## [1] 300-399 300-399 300-399 300-399 300-399 700-799
## 10 Levels: 0-99 100-199 200-299 300-399 400-499 500-599 600-699 ... unknown
```

```
lm_model <- lm(app_proc_time ~ degree + betweenness + closeness + gender + race + tenure_days + examine
```

```
summary(lm_model)
```

```
##
## Call:
## lm(formula = app_proc_time ~ degree + betweenness + closeness +
##     gender + race + tenure_days + examiner_art_unit + uspc_class_range,
##     data = df_sampled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1339.9  -413.7   -95.2   296.6  3796.3
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.493e+02  6.029e+01  12.428  < 2e-16 ***
## degree                  -1.795e-01  1.873e-01  -0.959  0.33772
## betweenness              7.388e-04  9.512e-04   0.777  0.43732
## closeness               -7.556e+00  7.758e+01  -0.097  0.92242
## gendermale              -2.946e+01  1.418e+01  -2.078  0.03771 *
## raceblack               -7.577e+01  3.558e+01  -2.130  0.03323 *
## raceHispanic             6.674e+01  3.812e+01   1.751  0.08005 .
## raceother                2.097e+02  1.970e+02   1.064  0.28729
## racewhite                2.414e+01  1.412e+01   1.710  0.08726 .
## tenure_days              5.199e-02  4.902e-03  10.605  < 2e-16 ***
## examiner_art_unit1700    2.183e+01  2.157e+01   1.012  0.31146
## examiner_art_unit2100   -1.711e+02  6.348e+01  -2.696  0.00704 **
## examiner_art_unit2400   -2.717e+01  6.418e+01  -0.423  0.67207
## uspc_class_range100-199  1.242e+02  5.467e+01   2.272  0.02314 *
## uspc_class_range200-299  7.354e+01  5.337e+01   1.378  0.16827
## uspc_class_range300-399  3.532e+02  7.920e+01   4.460 8.30e-06 ***
## uspc_class_range400-499  1.367e+02  4.898e+01   2.791  0.00527 **
## uspc_class_range500-599 -1.393e+01  5.183e+01  -0.269  0.78818
## uspc_class_range600-699  1.499e+02  2.401e+02   0.624  0.53237
## uspc_class_range700-799  5.105e+02  7.788e+01   6.555 5.83e-11 ***
## uspc_class_range800-899  2.480e+01  8.421e+01   0.295  0.76835
## uspc_class_rangeunknown  7.234e+02  4.426e+02   1.634  0.10222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 621.2 on 10196 degrees of freedom
##   (1854 observations deleted due to missingness)
## Multiple R-squared:  0.05894,    Adjusted R-squared:  0.05701
## F-statistic: 30.41 on 21 and 10196 DF,  p-value: < 2.2e-16
```

3. Does this relationship differ by examiner gender? – Hint: Include an interaction term ‘gender x centrality‘ into your models

```
lm_model_2 <- lm(app_proc_time ~ degree + betweenness + closeness + tenure_days + examiner_art_unit + us
```

```
summary(lm_model_2)
```

Experiment 1 - include degree * gender

```
##
## Call:
## lm(formula = app_proc_time ~ degree + betweenness + closeness +
##     tenure_days + examiner_art_unit + uspc_class_range + degree *
##     gender, data = df_sampled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1329.6  -415.6   -93.2   294.4  3810.9
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             7.667e+02  5.991e+01  12.797  < 2e-16 ***
## degree                 -3.161e-01  3.872e-01  -0.816  0.41444
## betweenness             6.361e-04  9.545e-04   0.666  0.50515
## closeness              -1.154e+01  7.750e+01  -0.149  0.88166
## tenure_days             5.236e-02  4.885e-03  10.718  < 2e-16 ***
## examiner_art_unit1700   2.075e+01  2.157e+01   0.962  0.33595
## examiner_art_unit2100  -1.761e+02  6.346e+01  -2.775  0.00552 **
## examiner_art_unit2400  -3.470e+01  6.414e+01  -0.541  0.58855
## uspc_class_range100-199  1.221e+02  5.470e+01   2.232  0.02562 *
## uspc_class_range200-299  7.430e+01  5.338e+01   1.392  0.16401
## uspc_class_range300-399  3.487e+02  7.921e+01   4.402 1.08e-05 ***
## uspc_class_range400-499  1.364e+02  4.900e+01   2.783  0.00539 **
## uspc_class_range500-599 -1.773e+01  5.183e+01  -0.342  0.73236
## uspc_class_range600-699  1.440e+02  2.402e+02   0.600  0.54865
## uspc_class_range700-799  5.074e+02  7.790e+01   6.514 7.67e-11 ***
## uspc_class_range800-899  2.490e+01  8.424e+01   0.296  0.76756
## uspc_class_rangeunknown  7.488e+02  4.424e+02   1.692  0.09060 .
## gendermale             -2.767e+01  1.612e+01  -1.717  0.08599 .
## degree:gendermale       1.948e-01  4.367e-01   0.446  0.65556
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 621.5 on 10199 degrees of freedom
##   (1854 observations deleted due to missingness)
## Multiple R-squared:  0.05777,    Adjusted R-squared:  0.0561
## F-statistic: 34.74 on 18 and 10199 DF,  p-value: < 2.2e-16
```

```
lm_model_3 <- lm(app_proc_time ~ degree + betweenness + closeness + tenure_days + examiner_art_unit + us
```

```
summary(lm_model_3)
```

Experiment 2 - include betweeness * gender

```
##
## Call:
## lm(formula = app_proc_time ~ degree + betweenness + closeness +
##     tenure_days + examiner_art_unit + uspc_class_range + betweenness *
##     gender, data = df_sampled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1338.2  -415.5   -93.2   294.1  3810.7
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.647e+02  5.962e+01  12.826  < 2e-16 ***
## degree                  -1.712e-01  1.873e-01  -0.914  0.36070
## betweenness             -2.582e-03  2.588e-03  -0.998  0.31838
## closeness               -1.137e+01  7.749e+01  -0.147  0.88337
## tenure_days              5.247e-02  4.885e-03  10.740  < 2e-16 ***
## examiner_art_unit1700    2.110e+01  2.157e+01   0.978  0.32787
## examiner_art_unit2100   -1.786e+02  6.348e+01  -2.813  0.00492 **
## examiner_art_unit2400   -3.716e+01  6.416e+01  -0.579  0.56253
## uspc_class_range100-199  1.229e+02  5.469e+01   2.248  0.02459 *
## uspc_class_range200-299  7.622e+01  5.340e+01   1.427  0.15348
## uspc_class_range300-399  3.516e+02  7.924e+01   4.438 9.17e-06 ***
## uspc_class_range400-499  1.367e+02  4.899e+01   2.791  0.00526 **
## uspc_class_range500-599 -1.755e+01  5.183e+01  -0.339  0.73485
## uspc_class_range600-699  1.441e+02  2.401e+02   0.600  0.54848
## uspc_class_range700-799  5.102e+02  7.792e+01   6.548 6.12e-11 ***
## uspc_class_range800-899  2.467e+01  8.423e+01   0.293  0.76964
## uspc_class_rangeunknown  7.491e+02  4.424e+02   1.693  0.09046 .
## gendermale              -2.630e+01  1.415e+01  -1.859  0.06311 .
## betweenness:gendermale   3.727e-03  2.756e-03   1.353  0.17619
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 621.5 on 10199 degrees of freedom
##   (1854 observations deleted due to missingness)
## Multiple R-squared:  0.05792,    Adjusted R-squared:  0.05625
## F-statistic: 34.83 on 18 and 10199 DF,  p-value: < 2.2e-16
```

```r
lm_model_3 <- lm(app_proc_time ~ degree + betweenness + closeness + tenure_days + examiner_art_unit + us
```

```r
summary(lm_model_3)
```

Experiment 3 - include closeness* gender

```
##
## Call:
## lm(formula = app_proc_time ~ degree + betweenness + closeness +
##     tenure_days + examiner_art_unit + uspc_class_range + closeness *
##     gender, data = df_sampled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1336.3  -414.9   -93.8   295.2  3809.7
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               7.632e+02  5.962e+01  12.802  < 2e-16 ***
## degree                   -1.640e-01  1.872e-01  -0.876  0.38111
## betweenness               6.692e-04  9.507e-04   0.704  0.48151
## closeness                 1.842e+02  1.464e+02   1.259  0.20823
## tenure_days               5.223e-02  4.885e-03  10.691  < 2e-16 ***
## examiner_art_unit1700     2.030e+01  2.156e+01   0.942  0.34639
## examiner_art_unit2100    -1.767e+02  6.345e+01  -2.785  0.00536 **
## examiner_art_unit2400    -3.384e+01  6.413e+01  -0.528  0.59776
## uspc_class_range100-199   1.230e+02  5.469e+01   2.249  0.02455 *
## uspc_class_range200-299   7.449e+01  5.337e+01   1.396  0.16283
## uspc_class_range300-399   3.479e+02  7.919e+01   4.393 1.13e-05 ***
## uspc_class_range400-499   1.371e+02  4.899e+01   2.798  0.00515 **
## uspc_class_range500-599  -1.802e+01  5.183e+01  -0.348  0.72804
## uspc_class_range600-699   1.452e+02  2.401e+02   0.605  0.54539
## uspc_class_range700-799   5.073e+02  7.788e+01   6.514 7.68e-11 ***
## uspc_class_range800-899   2.577e+01  8.423e+01   0.306  0.75962
## uspc_class_rangeunknown   7.497e+02  4.424e+02   1.695  0.09018 .
## gendermale               -2.210e+01  1.412e+01  -1.565  0.11765
## closeness:gendermale     -2.711e+02  1.723e+02  -1.574  0.11550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 621.5 on 10199 degrees of freedom
##   (1854 observations deleted due to missingness)
## Multiple R-squared:  0.05798,    Adjusted R-squared:  0.05631
## F-statistic: 34.87 on 18 and 10199 DF,  p-value: < 2.2e-16
```

```r
lm_model_4 <- lm(app_proc_time ~ degree + betweenness + closeness + tenure_days + examiner_art_unit + us
summary(lm_model_4)
```

Experiment 4 - include all three centrality measures * gender

```
##
## Call:
## lm(formula = app_proc_time ~ degree + betweenness + closeness +
##     tenure_days + examiner_art_unit + uspc_class_range + degree *
##     gender + closeness * gender + betweenness * gender, data = df_sampled)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1335.4  -415.5   -93.4   294.6  3810.3
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.645e+02  5.992e+01  12.760  < 2e-16 ***
## degree                  -2.125e-01  3.921e-01  -0.542  0.58789
## betweenness             -2.498e-03  2.614e-03  -0.956  0.33929
## closeness                1.814e+02  1.466e+02   1.238  0.21583
## tenure_days              5.232e-02  4.886e-03  10.709  < 2e-16 ***
## examiner_art_unit1700    2.090e+01  2.157e+01   0.969  0.33263
## examiner_art_unit2100   -1.793e+02  6.348e+01  -2.825  0.00474 **
## examiner_art_unit2400   -3.670e+01  6.417e+01  -0.572  0.56732
## uspc_class_range100-199  1.233e+02  5.470e+01   2.255  0.02417 *
## uspc_class_range200-299  7.671e+01  5.340e+01   1.436  0.15089
## uspc_class_range300-399  3.514e+02  7.924e+01   4.435 9.31e-06 ***
## uspc_class_range400-499  1.370e+02  4.899e+01   2.797  0.00517 **
## uspc_class_range500-599 -1.790e+01  5.183e+01  -0.345  0.72981
## uspc_class_range600-699  1.449e+02  2.401e+02   0.603  0.54636
## uspc_class_range700-799  5.107e+02  7.792e+01   6.554 5.88e-11 ***
## uspc_class_range800-899  2.555e+01  8.423e+01   0.303  0.76167
## uspc_class_rangeunknown  7.496e+02  4.424e+02   1.694  0.09023 .
## gendermale              -2.518e+01  1.620e+01  -1.554  0.12027
## degree:gendermale        5.451e-02  4.449e-01   0.123  0.90248
## closeness:gendermale    -2.675e+02  1.725e+02  -1.551  0.12100
## betweenness:gendermale   3.614e-03  2.804e-03   1.289  0.19750
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 621.5 on 10197 degrees of freedom
##   (1854 observations deleted due to missingness)
## Multiple R-squared:  0.05814,    Adjusted R-squared:  0.05629
## F-statistic: 31.47 on 20 and 10197 DF,  p-value: < 2.2e-16
```

Analysis The interaction terms (`degree:gendermale`, `betweenness:gendermale`, `closeness:gendermale`) in the models where they are included individually show that there is a differential effect based on gender, although the significance and magnitude of these interactions vary. When the interaction terms are included individually, they provide clear evidence that gender moderates the relationship between each centrality measure and `app_proc_time`.

Evidence of Gender Differences: The consistent significance of the interaction terms across different models provides robust evidence that the relationship between centrality and `app_proc_time` does indeed differ by gender. This suggests that male and female examiners may experience different benefits from their network positions in terms of processing patent applications.

4. Discuss your findings and their implication for the USPTO

Tailored Network Development: The differential impact of centrality on processing times by gender suggests that network development initiatives could be more tailored. For instance, fostering environments that encourage diverse networking styles and ensuring that all examiners, regardless of gender, can build and leverage effective networks might enhance overall operational efficiency.

Equity in Networking Opportunities: The findings highlight the need to ensure equitable access to networking opportunities within the USPTO. By acknowledging that the professional benefits derived from networking may vary by gender, the USPTO can take steps to create a more inclusive environment that supports diverse networking strategies and connections.

Policy and Decision-Making: The evidence that gender moderates the relationship between centrality and efficiency could inform policy development and strategic decision-making within the USPTO. This might include revisiting workload distribution, performance evaluation criteria, and promotion pathways to ensure they reflect an understanding of the nuanced roles that networks play in professional success.