Exe. 4

```
library(ggraph)
library(igraph)

library(arrow)
library(tidyverse)
library(gender)
library(wru)
library(lubridate)

library(ggplot2)
library(gridExtra)
library(grid)

library("gender")
library("mice")
```

```
app <- read_parquet('/Users/danystefan/Documents/01 McGill University/01 MMA/01 Summer 2022/ORGB 672/As

edges <- read_csv('/Users/danystefan/Documents/01 McGill University/01 MMA/01 Summer 2022/ORGB 672/Assi
```

Gender will be determined based on the examiner's first name, which is stored in the field examiner name first. We'll do that using library gender, based on a modified version of their own example.

The applications table has almost 2 million records, which is due to the fact that each examiner has as many records as the amount of applications the examiner worked on during this time period. As a result, our initial step is to collect all unique names into a distinct list called examiner names. We'll next make a guess about each person's gender and link this table back to the original dataset. So, without further ado, here are some names:

Add gender part

```
# first name
names <- app %>% distinct(examiner_name_first)
# names and gender
names_gender <- names %>%
  do(results = gender(.$examiner_name_first, method = "ssa")) %>%
  unnest(cols = c(results), keep_empty = TRUE) %>%
  select(
    examiner_name_first = name,
    gender,
    proportion_female
  )
names_gender <- names_gender %>% select(examiner_name_first, gender)

# join
app <- app %>% left_join(names_gender, by='examiner_name_first')
```

Add race part

```
# last names
sur <- app %>% select(surname = examiner_name_last) %>% distinct()

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

```
## [1] "Proceeding with surname-only predictions..."
```

```
race <- 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_
  ))

# cleanup
race <- race %>% select(surname, race)

#join
app <- app %>% left_join(race, by=c("examiner_name_last" = "surname"))
```

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.

Add tenure part

```
# get dates
dates <- app %>% select(examiner_id, filing_date, appl_status_date)
# calculate start and end date
dates <- dates %>% mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))

dates <- 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)

# join
app <- app %>% left_join(dates, by="examiner_id")
```

APT

```r
app$appl_end_date <- paste(app$patent_issue_date, app$abandon_date, sep=',')

# cleanup
app$appl_end_date <- gsub('NA', "", as.character(app$appl_end_date))
app$appl_end_date <- gsub(',', "", as.character(app$appl_end_date))

# date
app$appl_end_date <- as.Date(app$appl_end_date, format="%Y-%m-%d")
app$filing_date <- as.Date(app$filing_date, format="%Y-%m-%d")

app$appl_proc_days <- as.numeric(difftime(app$appl_end_date, app$filing_date, units=c("days")))

# cleanup
app <- app %>% filter(appl_proc_days >=0 | appl_proc_days != NA)
```

```r
# Find the count of missing values in each column
sapply(app, function(x) sum(is.na(x)))
```

```
##   application_number          filing_date    examiner_name_last
##                    0                    0                     0
##  examiner_name_first examiner_name_middle           examiner_id
##                    0               390396                  3746
##    examiner_art_unit           uspc_class         uspc_subclass
##                    0                    4                  1555
##        patent_number    patent_issue_date          abandon_date
##               601857               601383               1087295
##        disposal_type     appl_status_code      appl_status_date
##                    0                  355                   356
##                   tc               gender                  race
##                    0               253871                     0
##        earliest_date          latest_date           tenure_days
##                18240                18240                 18240
##        appl_end_date       appl_proc_days
##                    0                    0
```

```r
# Remove unnecessary columns for modelling
applications_mod <- subset(app, select = -c(filing_date, abandon_date, earliest_date, appl_end_date, ap

sapply(applications_mod, function(x) sum(is.na(x)))
```

```
##  application_number examiner_name_last examiner_name_first         examiner_id
##                   0                  0                   0                3746
##   examiner_art_unit         uspc_class       uspc_subclass       disposal_type
##                   0                  4                1555                   0
##    appl_status_code                 tc              gender                race
##                 355                  0              253871                   0
##         tenure_days     appl_proc_days
##               18240                  0
```

```r
applications_mod <- applications_mod %>% drop_na(examiner_id)
```

```r
applications_mod$gender <- as.factor(applications_mod$gender)

applications_mod_imp <- complete(mice(applications_mod, m=3, maxit=3))
```

```
##
##  iter imp variable
##   1   1  appl_status_code  gender  tenure_days
##   1   2  appl_status_code  gender  tenure_days
##   1   3  appl_status_code  gender  tenure_days
##   2   1  appl_status_code  gender  tenure_days
##   2   2  appl_status_code  gender  tenure_days
##   2   3  appl_status_code  gender  tenure_days
##   3   1  appl_status_code  gender  tenure_days
##   3   2  appl_status_code  gender  tenure_days
##   3   3  appl_status_code  gender  tenure_days
```

Network

```
# workgroup
examiner_aus = distinct(subset(applications_mod_imp, select=c(examiner_art_unit, examiner_id)))

examiner_aus$wg = substr(examiner_aus$examiner_art_unit, 1,3)

# art unit
examiner_aus = examiner_aus[examiner_aus$wg==162 | examiner_aus$wg==219,]

# merge
adv_network = merge(x=edges, y=examiner_aus, by.x="ego_examiner_id", by.y="examiner_id", all.x=TRUE)
adv_network = adv_network %>% rename(ego_art_unit=examiner_art_unit, ego_wg=wg)

adv_network = drop_na(adv_network)


adv_network = merge(x=adv_network, y=examiner_aus, by.x="alter_examiner_id", by.y="examiner_id", all.x=
adv_network = adv_network %>% rename(alter_art_unit=examiner_art_unit, alter_wg=wg)
adv_network = drop_na(adv_network)

egoNodes = subset(adv_network, select=c(ego_examiner_id,ego_art_unit, ego_wg)) %>% rename(examiner_id=e
alterNodes = subset(adv_network, select=c(alter_examiner_id,alter_art_unit, alter_wg))%>% rename(examin
nodes = rbind(egoNodes, alterNodes)
nodes = distinct(nodes)

nodes = nodes %>% group_by(examiner_id) %>% summarise(examiner_id=first(examiner_id), art_unit=first(ar

network <- graph_from_data_frame(d=adv_network, vertices=nodes, directed=TRUE)
network
```

```
## IGRAPH 48d1ba2 DN-- 98 916 --
## + attr: name (v/c), art_unit (v/n), wg (v/c), application_number (e/c),
## | advice_date (e/n), ego_art_unit (e/n), ego_wg (e/c), alter_art_unit
## | (e/n), alter_wg (e/c)
## + edges from 48d1ba2 (vertex names):
##  [1] 59491->76935 59491->76935 59491->76935 59491->76935 59491->76935
##  [6] 59491->76935 59491->76935 59491->76935 59491->76935 61889->88905
## [11] 61889->88905 62114->72495 62114->66534 62114->72495 62114->66534
## [16] 62253->67690 62253->67690 62253->67690 62253->67690 63657->73150
## [21] 63657->73150 63657->73150 63657->73150 63657->73150 63657->73150
## [26] 63822->61417 64904->96780 65111->65111 65111->65111 65111->65111
```

```
## + ... omitted several edges
```

```
Degree <- degree(network)
Closeness <- closeness(network)
Betweenness <- betweenness(network)
Eig <- evcent(network)$vector

comp <- data.frame(nodes, Degree, Eig, Closeness, Betweenness)
comp
```

```
##       examiner_id art_unit  wg Degree          Eig  Closeness Betweenness
## 59491       59491     2196 219     48 2.644753e-02 1.00000000        0.90
## 59664       59664     2193 219      3 1.875528e-03        NaN        0.00
## 60302       60302     1626 162      6 0.000000e+00        NaN        0.00
## 60465       60465     2193 219      5 1.897166e-03        NaN        0.00
## 60768       60768     2191 219      2 8.131513e-05        NaN        0.00
## 61064       61064     2192 219      5 1.221863e-04        NaN        0.00
## 61417       61417     1626 162      2 0.000000e+00        NaN        0.00
## 61529       61529     1628 162      3 0.000000e+00        NaN        0.00
## 61889       61889     2193 219     24 9.429415e-03 1.00000000        4.00
## 61980       61980     2195 219      1 1.856425e-05        NaN        0.00
## 62114       62114     2195 219      4 3.187399e-05 0.50000000        0.00
## 62253       62253     1623 162      5 0.000000e+00 1.00000000        0.00
## 62661       62661     1627 162     13 8.215403e-20        NaN        0.00
## 63657       63657     2196 219     87 2.804044e-03 1.00000000        0.00
## 63822       63822     1624 162      1 0.000000e+00 1.00000000        0.00
## 63971       63971     2192 219      2 5.780753e-05        NaN        0.00
## 64904       64904     2192 219      1 1.592129e-05 1.00000000        0.00
## 65111       65111     1623 162     20 0.000000e+00 1.00000000        0.00
## 65353       65353     2193 219      9 2.027090e-04        NaN        0.00
## 65536       65536     1625 162      1 0.000000e+00 1.00000000        0.00
## 65537       65537     1624 162      3 2.964756e-20 1.00000000        0.00
## 65554       65554     2194 219      4 3.320903e-02        NaN        0.00
## 65713       65713     1623 162     12 0.000000e+00 0.50000000        0.00
## 65737       65737     1621 162      1 0.000000e+00 1.00000000        0.00
## 66030       66030     2193 219     19 3.703110e-04 0.10000000        0.00
## 66359       66359     2194 219      6 1.844651e-05        NaN        0.00
## 66534       66534     2194 219     80 1.919068e-03        NaN        0.00
## 67078       67078     2196 219     12 5.682276e-03        NaN        0.00
## 67208       67208     1624 162    120 9.962709e-01 0.20000000        0.00
## 67226       67226     2197 219    136 1.000000e+00 0.50000000        3.10
## 67256       67256     1627 162     22 0.000000e+00 0.14285714        0.00
## 67581       67581     1621 162      7 0.000000e+00 0.25000000        0.00
## 67690       67690     1623 162     31 0.000000e+00        NaN        0.00
## 67731       67731     1621 162      1 0.000000e+00 1.00000000        0.00
## 67753       67753     1623 162      1 0.000000e+00        NaN        0.00
## 68166       68166     1625 162     11 1.406909e-18 1.00000000        0.00
## 68339       68339     1626 162      1 0.000000e+00 1.00000000        0.00
## 68695       68695     1627 162      7 0.000000e+00 1.00000000        1.00
## 68752       68752     2192 219     12 3.924478e-03        NaN        0.00
## 69896       69896     1628 162      1 0.000000e+00        NaN        0.00
## 70026       70026     2192 219     90 3.481434e-03 0.04000000        0.00
## 70206       70206     1627 162      4 0.000000e+00        NaN        0.00
## 70767       70767     1624 162     10 0.000000e+00        NaN        0.00
## 71175       71175     2193 219      6 1.896603e-03 0.08333333        2.25
```

```
## 71558       71558   2191 219     2 8.131513e-05        NaN       0.00
## 71996       71996   2192 219    52 1.618863e-03        NaN       0.00
## 72089       72089   2195 219    12 2.236048e-03 0.16666667       0.00
## 72495       72495   2193 219     2 5.292522e-07        NaN       0.00
## 72941       72941   1626 162     1 0.000000e+00        NaN       0.00
## 73150       73150   2192 219    44 1.151042e-03        NaN       0.00
## 73364       73364   1629 162     1 0.000000e+00        NaN       0.00
## 73777       73777   1623 162     7 0.000000e+00 1.00000000       0.00
## 75034       75034   1626 162     3 0.000000e+00 1.00000000       1.00
## 75431       75431   2195 219     6 3.751055e-03        NaN       0.00
## 75940       75940   2191 219    52 8.568931e-03        NaN       0.00
## 76141       76141   2191 219    30 7.733290e-03 0.50000000       4.00
## 76935       76935   2199 219    24 5.200072e-02        NaN       0.00
## 77348       77348   1626 162     1 0.000000e+00 1.00000000       0.00
## 81211       81211   1628 162     3 0.000000e+00        NaN       0.00
## 81865       81865   1621 162     6 0.000000e+00 1.00000000       0.00
## 81959       81959   1629 162     3 0.000000e+00        NaN       0.00
## 82386       82386   2191 219     8 0.000000e+00        NaN       0.00
## 83552       83552   2194 219     8 1.338727e-03 0.25000000       0.00
## 84460       84460   2193 219    15 8.461901e-04 0.16666667      12.00
## 85216       85216   1627 162     5 0.000000e+00        NaN       0.00
## 87028       87028   2191 219   186 4.897170e-03 0.09090909      16.00
## 87486       87486   1621 162     5 0.000000e+00 0.14285714       0.00
## 87994       87994   2191 219     5 1.020354e-05 1.00000000       0.00
## 88077       88077   2191 219    80 2.649855e-03        NaN       0.00
## 88508       88508   1621 162     3 0.000000e+00        NaN       0.00
## 88905       88905   2199 219     2 1.565709e-04        NaN       0.00
## 89882       89882   1623 162     4 0.000000e+00 0.50000000       0.00
## 91747       91747   1627 162     1 0.000000e+00 1.00000000       0.00
## 91956       91956   1627 162     3 0.000000e+00        NaN       0.00
## 92902       92902   1621 162     1 0.000000e+00        NaN       0.00
## 93403       93403   1626 162     9 0.000000e+00 0.25000000       0.00
## 93677       93677   1623 162     1 0.000000e+00        NaN       0.00
## 94070       94070   1623 162    10 0.000000e+00 0.50000000       0.00
## 94513       94513   2193 219     5 1.927300e-03        NaN       0.00
## 94925       94925   1626 162     4 0.000000e+00        NaN       0.00
## 95339       95339   2193 219    47 2.814301e-02        NaN       0.00
## 95446       95446   1625 162    17 0.000000e+00        NaN       0.00
## 95769       95769   2193 219     4 1.882553e-03 0.08333333       0.75
## 95997       95997   2192 219     4 0.000000e+00        NaN       0.00
## 96206       96206   2194 219     8 4.067542e-05        NaN       0.00
## 96780       96780   2192 219     7 1.917706e-03        NaN       0.00
## 96898       96898   1628 162     2 0.000000e+00        NaN       0.00
## 97328       97328   2199 219   195 7.530191e-02 0.02857143       0.00
## 97520       97520   1624 162     1 0.000000e+00 1.00000000       0.00
## 97590       97590   2193 219     2 6.956133e-05        NaN       0.00
## 97673       97673   2193 219     6 3.751055e-03        NaN       0.00
## 98228       98228   2195 219    25 1.502279e-02        NaN       0.00
## 98700       98700   1625 162    13 0.000000e+00 0.25000000       0.00
## 98717       98717   2192 219    31 7.914968e-04 0.04347826       0.00
## 99047       99047   1627 162     4 4.215424e-18        NaN       0.00
## 99346       99346   2192 219    14 3.453124e-04 1.00000000       0.00
## 99424       99424   1625 162     1 8.250416e-20        NaN       0.00
## 99514       99514   2192 219     8 9.570456e-05        NaN       0.00
```

Final Merge

```
applications_final <- merge(x=applications_mod_imp, y=comp, by='examiner_id', all.x=TRUE)
```

```
applications_final = applications_final %>% filter(wg==162 | wg==219)

applications_final <- drop_na(applications_final)
```

Model

Simple linear model

```
lm1 <- lm(appl_proc_days~Eig + Degree + Closeness + Betweenness + gender + tenure_days, data=applicatio
summary(lm1)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Eig + Degree + Closeness + Betweenness +
##     gender + tenure_days, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1964.9  -410.0   -82.4   303.9  4233.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.603e+03  6.311e+01  25.404  < 2e-16 ***
## Eig         -2.308e+02  3.027e+01  -7.623 2.56e-14 ***
## Degree       2.942e+00  1.335e-01  22.043  < 2e-16 ***
## Closeness   -1.361e+02  1.143e+01 -11.907  < 2e-16 ***
## Betweenness  2.652e+01  1.767e+00  15.014  < 2e-16 ***
## gendermale   3.533e+00  7.474e+00   0.473    0.636
## tenure_days -6.773e-02  9.975e-03  -6.789 1.15e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 572.5 on 28881 degrees of freedom
## Multiple R-squared:  0.05473,    Adjusted R-squared:  0.05453
## F-statistic: 278.7 on 6 and 28881 DF,  p-value: < 2.2e-16
```

- The statistical model is significant, and most factors are significant, with the exception of any gender impact. - The baseline processing time is 1604 days (Female, 0 days of tenure, never sought advice) - Increasing the relevance of an examiner's eigenvector from 0 to 1 should reduce processing time by 236 days (i.e, the more important). This makes sense since if a given examiner has a lot of clout in an advice network, with a lot of other examiners seeking their assistance, they are likely to be a subject matter expert and would need to spend less time looking for answers online or from other people to process the application.
- An examiner requesting guidance from another examiner one more time (an increase of one degree) results in a processing time increase of around three days. This might make sense because getting extra guidance or having others come to you for advise is time intensive and could divert time away from processing applications. - A 138-day reduction in processing time would be projected if closeness centrality increased from 0 to 1. This makes sense because a high closeness centrality corresponds to a well-connected examiner inside the network. Even if they don't know someone who is an expert in a certain field, they are very certain to know someone who knows someone. This might make locating the information they need quicker and reduce the amount of time it takes to complete the application - A

27-day increase in processing time equates to a one-unit rise in betweenness centrality. If an examiner is a primary gate for information to travel in the network, similar to degree centrality, this might be time intensive and take time away from them processing applications. - Finally, a one-day increase in tenure results in a small reduction in processing time. It would seem logical that having more experience will result in faster processing times.

Some more varibales:

```
lm2 <- lm(appl_proc_days~Eig + Degree + Closeness + Betweenness + gender + tenure_days + Degree*gender ·
summary(lm2)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Eig + Degree + Closeness + Betweenness +
##     gender + tenure_days + Degree * gender + Eig * gender + Closeness *
##     gender + Betweenness * gender, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1632.7  -404.7   -83.4   298.8  4233.2
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1391.11240   66.40451  20.949  < 2e-16 ***
## Eig                    -535.38072   68.58969  -7.806 6.12e-15 ***
## Degree                    5.22062    0.56231   9.284  < 2e-16 ***
## Closeness                23.05344   29.22464   0.789     0.43
## Betweenness             101.77642    6.92029  14.707  < 2e-16 ***
## gendermale              191.73229   31.01534   6.182 6.42e-10 ***
## tenure_days              -0.06194    0.01003  -6.178 6.57e-10 ***
## Degree:gendermale        -4.34747    0.60226  -7.219 5.38e-13 ***
## Eig:gendermale         7069.60726  636.06235  11.115  < 2e-16 ***
## Closeness:gendermale   -159.21115   31.85330  -4.998 5.82e-07 ***
## Betweenness:gendermale  -74.83568    7.17803 -10.426  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 569.2 on 28877 degrees of freedom
## Multiple R-squared:  0.06574,    Adjusted R-squared:  0.06542
## F-statistic: 203.2 on 10 and 28877 DF,  p-value: < 2.2e-16
```

- All factors are significant with at least 90% confidence in the stat significant model. - Relationships for variables from the previous model are comparable, with the exception that proximity now has a higher positive link with processing time (a unit increase in closeness centrality correlates with a 48 day increase in processing time) - Tenure still reduces processing time for male examiners - A unit increase in degree for male examiners reduces processing time by 4.5 days - A unit increase in eig importance for male examiners increases processing time by 7270 days - A unit increase in closeness for males decreases processing time by 191 days - A unit increase in betweenness for males decreases processing time by 78 days -