Exercise 3

Heidi Al Wakeel

2023-03-24

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   0.3.5
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(arrow)
```

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

```
##
## Attaching package: 'arrow'
##
## The following object is masked from 'package:lubridate':
##
##     duration
##
## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(readr)
library(gender)
```

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

```
library(wru)
```

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

```
library(lubridate)
library(ggplot2)
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(ggraph)
library(tidygraph)
```

```
##
## Attaching package: 'tidygraph'
##
## The following object is masked from 'package:igraph':
```

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

```
data_path <- "C:/Users/Heidi Al Wakeel/Documents/2023-ona-assignments/"
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 x 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 1: application_number, 2: filing_date,
## #   3: examiner_name_last, 4: examiner_name_first, 5: examiner_name_middle,
## #   6: examiner_id, 7: examiner_art_unit, 8: uspc_class, 9: uspc_subclass
```

```
edges
```

```
## # A tibble: 32,906 x 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
```

Question 1

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:

```
#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 <- applications %>%
  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
## # ... with 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`

```
# 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
## # ... with 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 <- 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  4830206 258.0    8028798 428.8  5377062 287.2
## Vcells 50083834 382.2   96111708 733.3 80399611 613.5
```

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.

```
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
## # ... with 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()
```

```
## 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
## # ... with 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
## # ... with 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  4884169 260.9    8028798 428.8  7540451 402.8
## Vcells 54318728 414.5  115414049 880.6 96097706 733.2
```

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.

```
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
## # ... with 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
## # ... with 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   4890786 261.2   14306616  764.1  14306616  764.1
## Vcells 64583463 492.8  138576858 1057.3 138537816 1057.0
```

Question2

```
# we pick work group 179 and 176
w179 <- subset(applications, grepl("^179", applications$examiner_art_unit))
w179$gender <- factor(w179$gender)
w179$race <- factor(w179$race)
w176 <- subset(applications, grepl("^176", applications$examiner_art_unit))
w176$gender <- factor(w176$gender)
w176$race <- factor(w176$race)
```

```
# Summary statistics for work group 179
summary(w179$gender)
```

```
## female   male   NA's
##  43783  77344  12297
```

```
summary(w179$race)
```

```
##    Asian    black Hispanic    other    white
##    28335     3771     2449       24    98845
```

```
summary(w179$tenure_days)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     774    5080    6304    5712    6342    6391    1058
```

```
# Summary statistics for work group 176

summary(w176$gender)
```

```
## female   male   NA's
##  28075  53561   9740
```

```
summary(w176$race)
```

```
##    Asian    black Hispanic    white
##    23022     4010     2520    61824
```

```
summary(w176$tenure_days)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     339    4524    6294    5501    6342    6350    1017
```

Race and gender distribution for work group 179 and 176 respectively

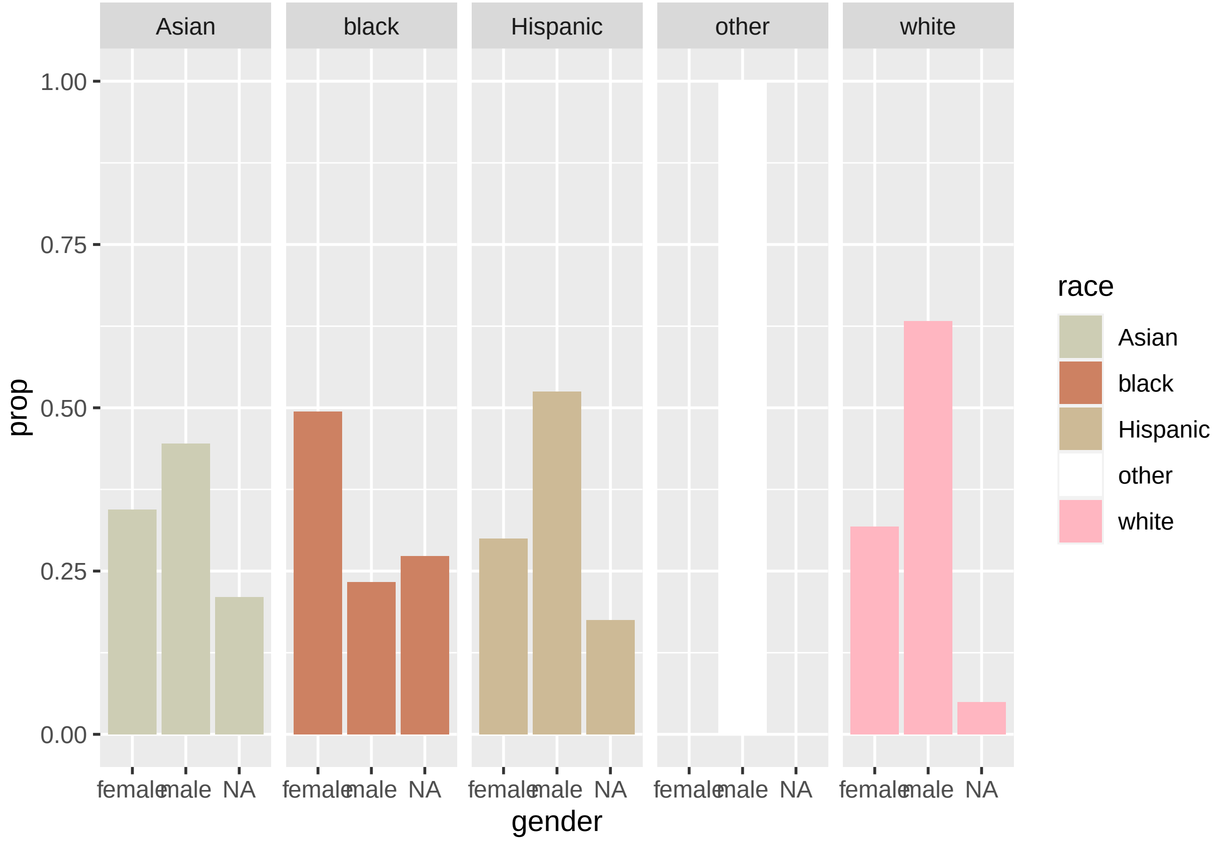
```
# merge
w179$workgroup <- c('179')
w176$workgroup <- c('176')
merged = union(x = w179,y = w176)
```

Gender distribution in races for work group 179

```
toPlot<-w179%>%
  group_by(gender, race)%>%
  summarise(n = n())%>%
  group_by(race)%>%
  mutate(prop = n/sum(n))
```

```
## ‘summarise()‘ has grouped output by ’gender’. You can override using the
## ‘.groups‘ argument.
```

```
ggplot(data = toPlot, aes(gender, prop, fill = race)) +
  geom_col() +
  facet_grid(~race)+
  scale_fill_manual(values = c("lightyellow3","lightsalmon3", "wheat3","white", "lightpink"))
```

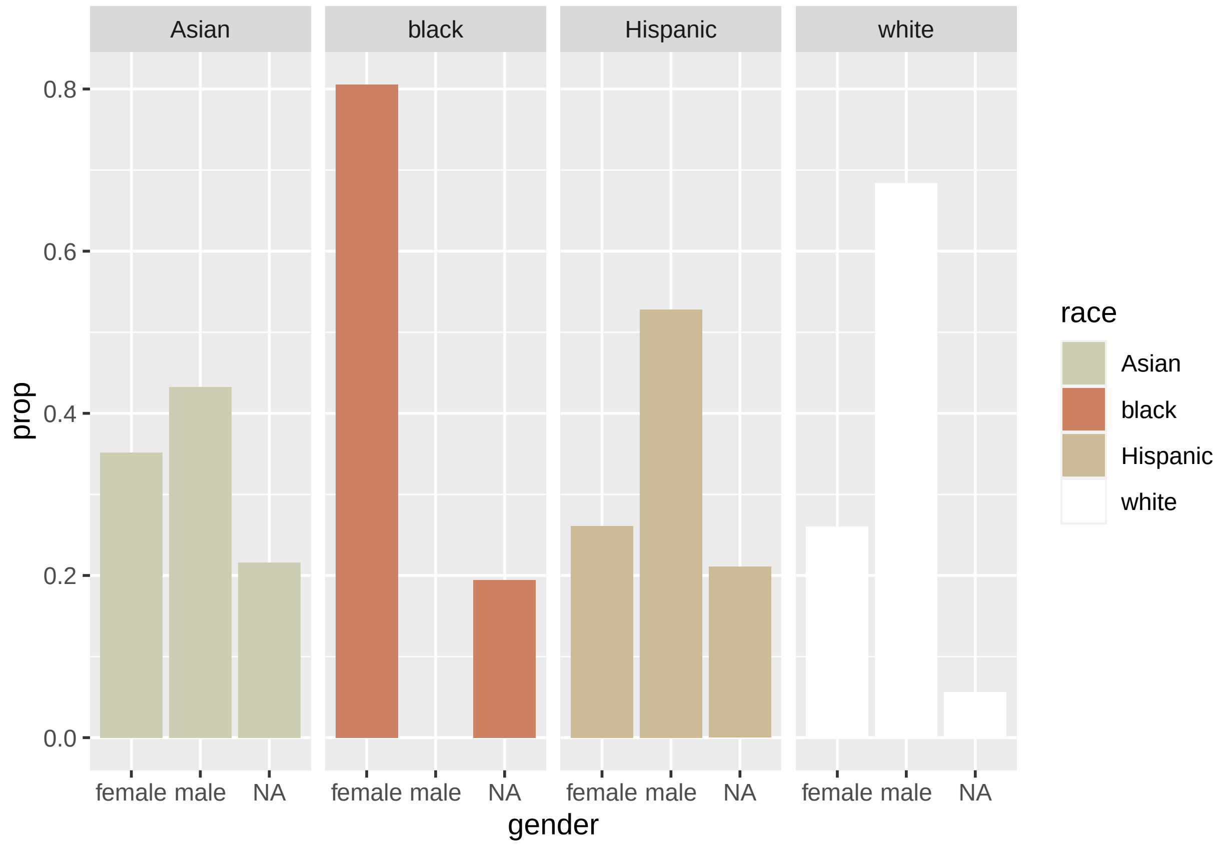


Gender distribution in races for work group 176

```
toPlot<-w176%>%
  group_by(gender, race)%>%
  summarise(n = n())%>%
  group_by(race)%>%
  mutate(prop = n/sum(n))
```

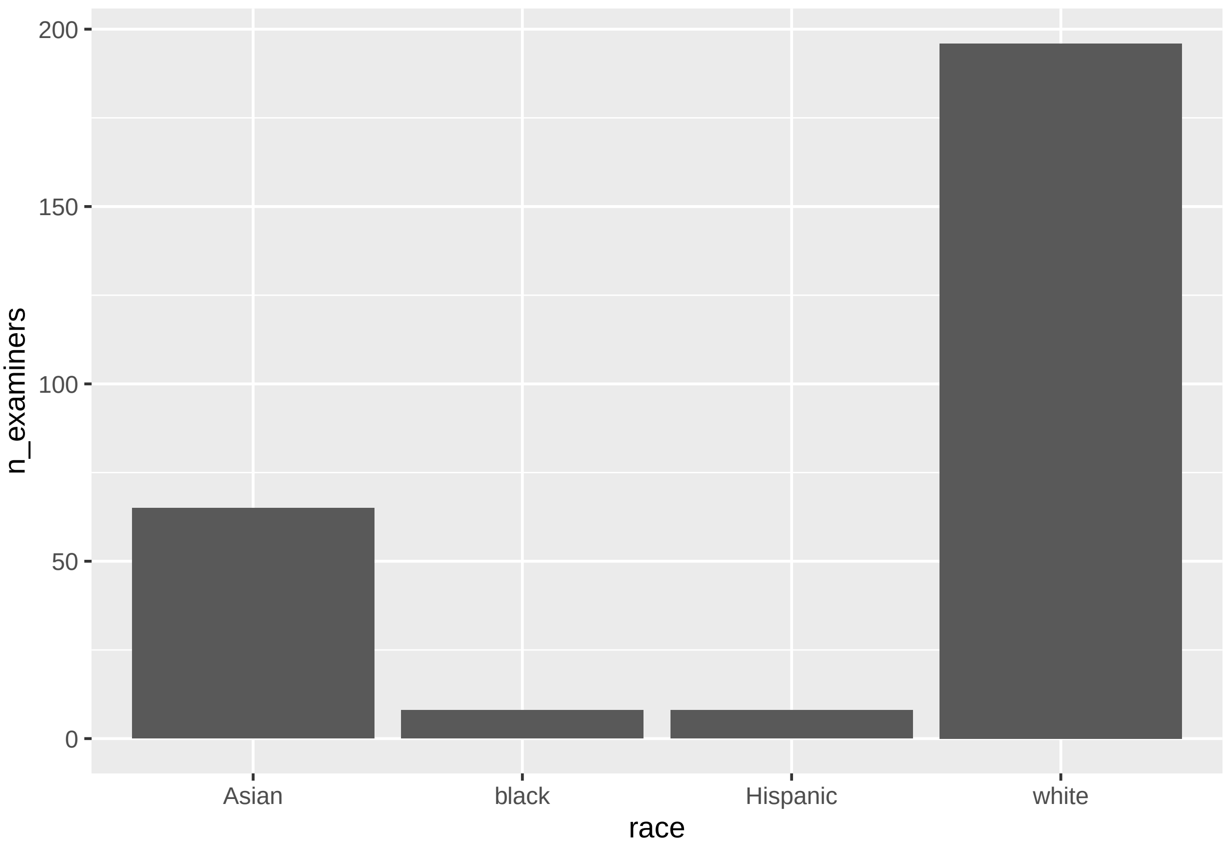
```
## ‘summarise()‘ has grouped output by ’gender’. You can override using the
## ‘.groups‘ argument.
```

```
ggplot(data = toPlot, aes(gender, prop, fill = race)) +
  geom_col() +
  facet_grid(~race)+
  scale_fill_manual(values = c("lightyellow3","lightsalmon3", "wheat3","white","lightpink"))
```

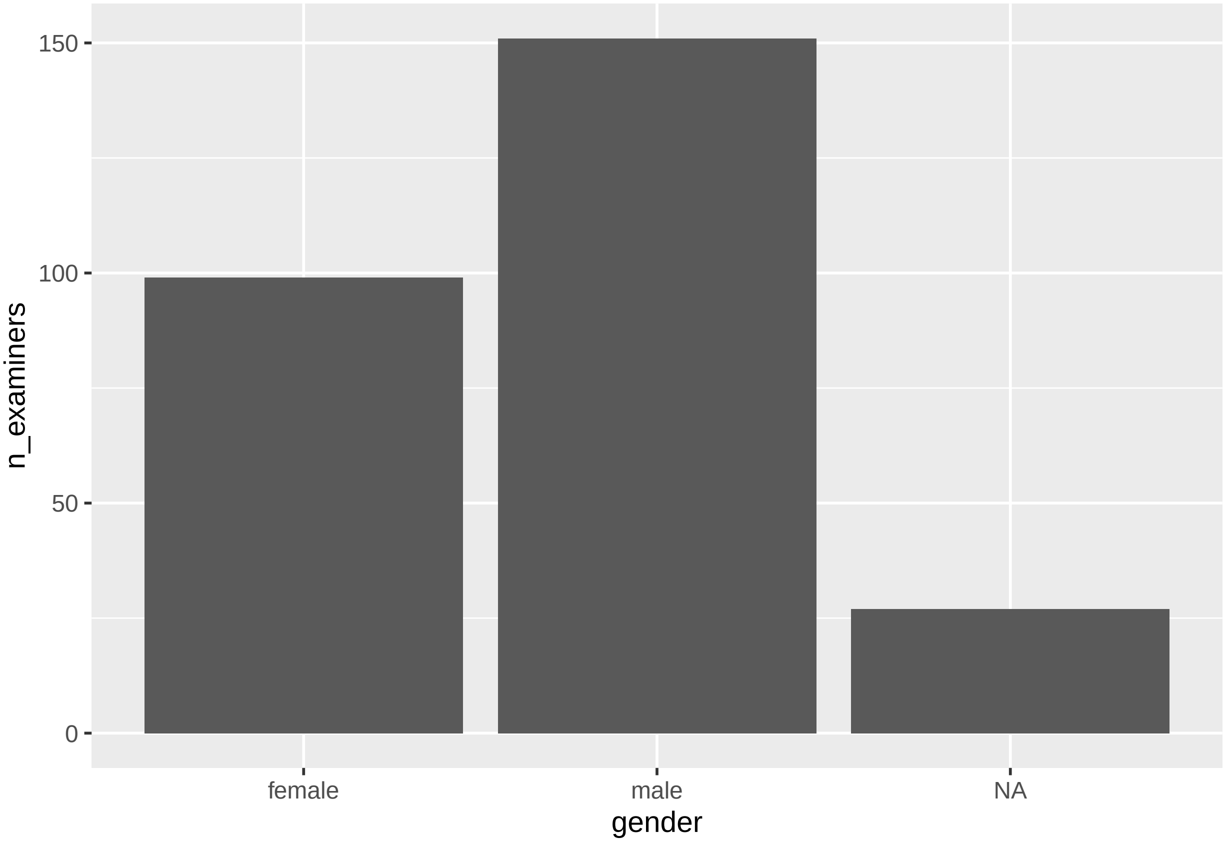


#Let's take a deeper dive:

```
# Choosing workgroups 176 and 179
wg1 = applications %>% filter(substr(examiner_art_unit, 1, 3) == '176' ) %>%
  arrange(application_number)
wg2 = applications %>% filter(substr(examiner_art_unit, 1, 3) == '179' ) %>%
  arrange(application_number)
#summary(wg1)
# distributions for wg 176
p1 = wg1 %>% group_by(race) %>% summarise(n_examiners = n_distinct(examiner_id)) %>%
  ggplot(aes(x = race, y = n_examiners)) + geom_bar(stat ='identity')
p2 = wg1 %>% group_by(gender) %>% summarise(n_examiners = n_distinct(examiner_id)) %>%
  ggplot(aes(x = gender, y = n_examiners)) + geom_bar(stat ='identity')
p3 = wg1 %>% ggplot(aes(x = tenure_days)) + geom_histogram()

print(p1)
```



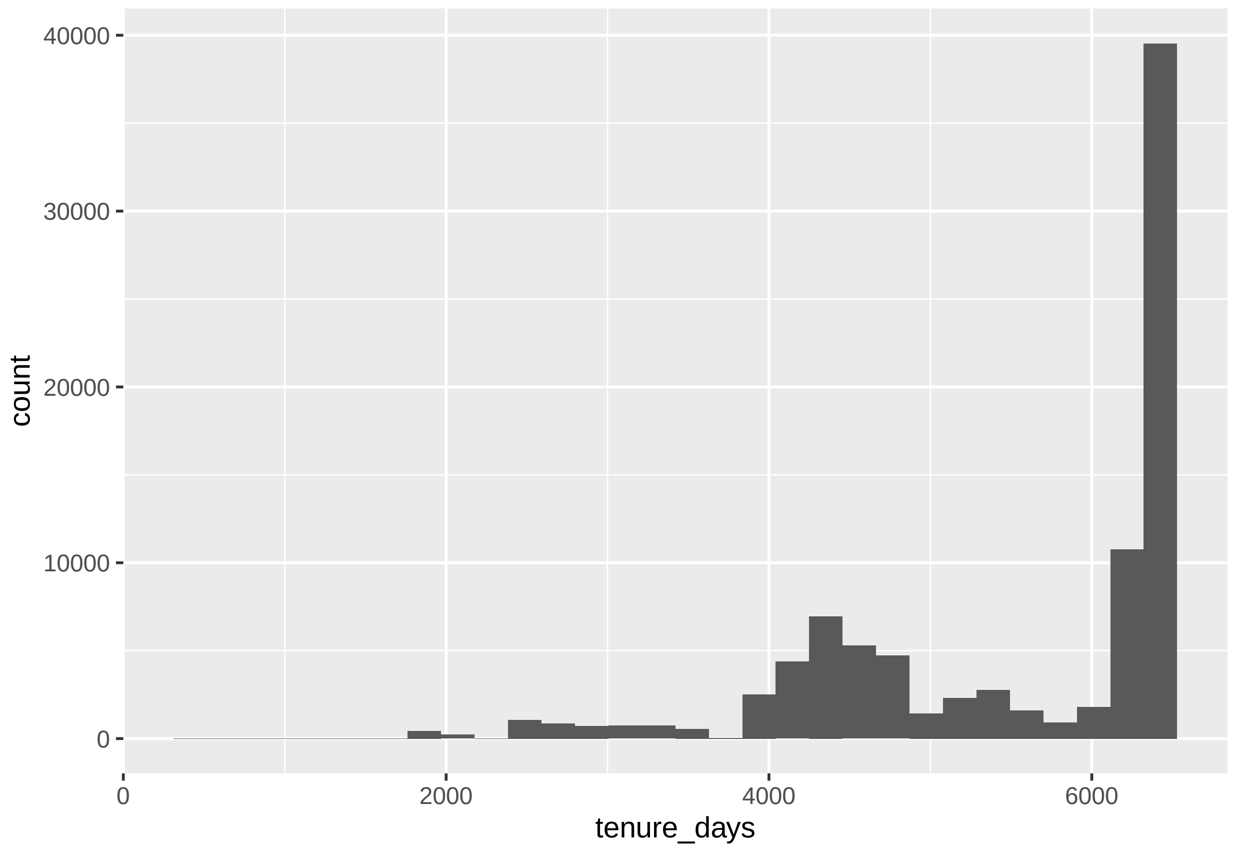
```
print(p2)
```



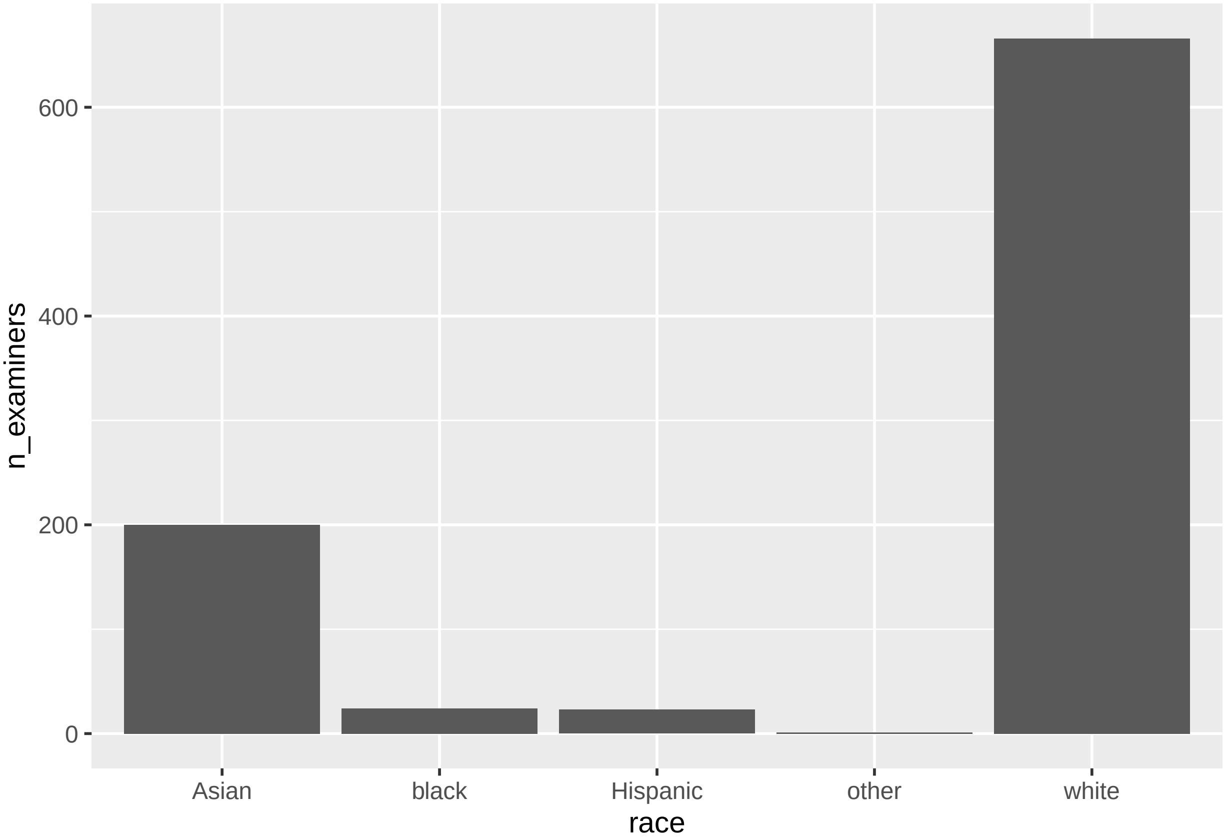
```
print(p3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

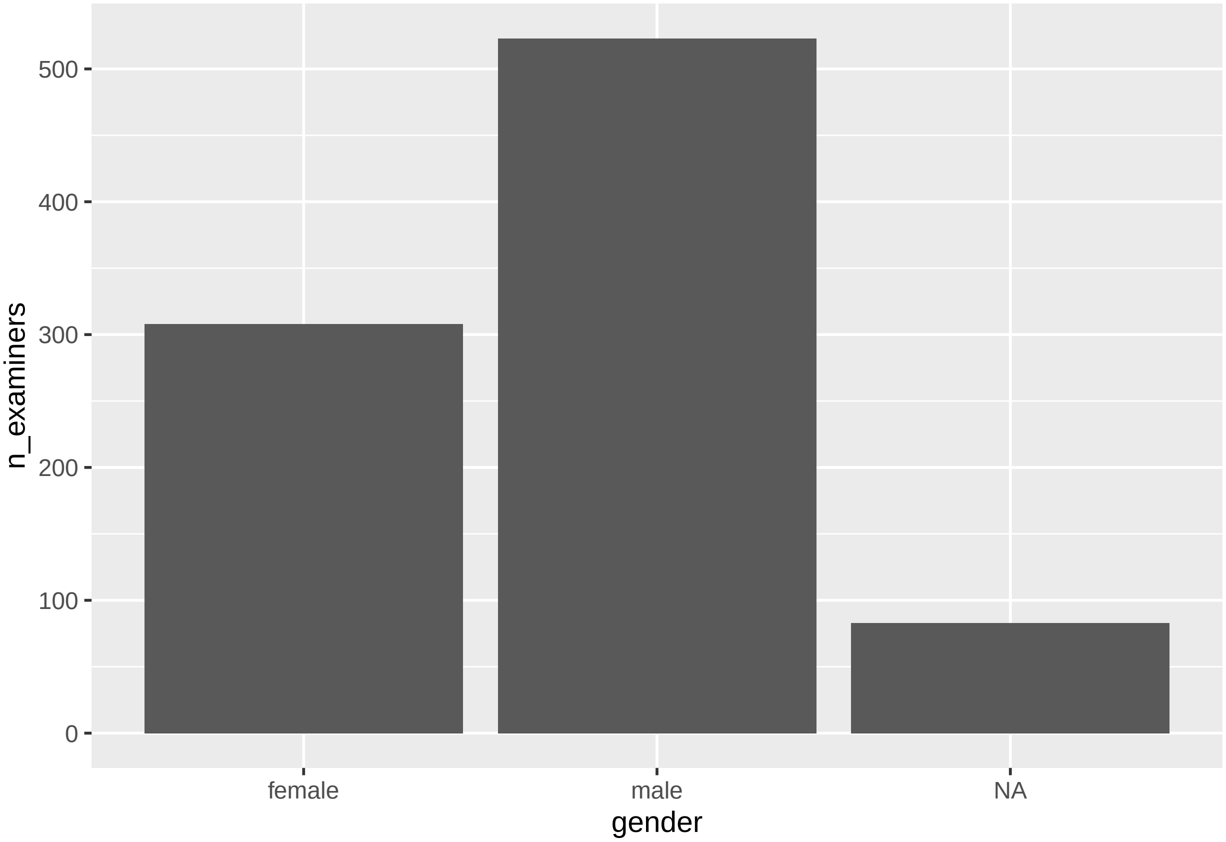
```
## Warning: Removed 1017 rows containing non-finite values (`stat_bin()`).
```



```
# distributions for wg 179
p1 = wg2 %>% group_by(race) %>% summarise(n_examiners = n_distinct(examiner_id)) %>%
  ggplot(aes(x = race, y = n_examiners)) + geom_bar(stat ='identity')
p2 = wg2 %>% group_by(gender) %>% summarise(n_examiners = n_distinct(examiner_id)) %>%
  ggplot(aes(x = gender, y = n_examiners)) + geom_bar(stat ='identity')
p3 = wg2 %>% ggplot(aes(x = tenure_days)) + geom_histogram()
#par(mfrow=c(1,3))
#grid.arrange(p1, p2, p3, ncol=3)
p1
```


600
400
200
0
n_examiners
Asian
black
Hispanic
other
white
race

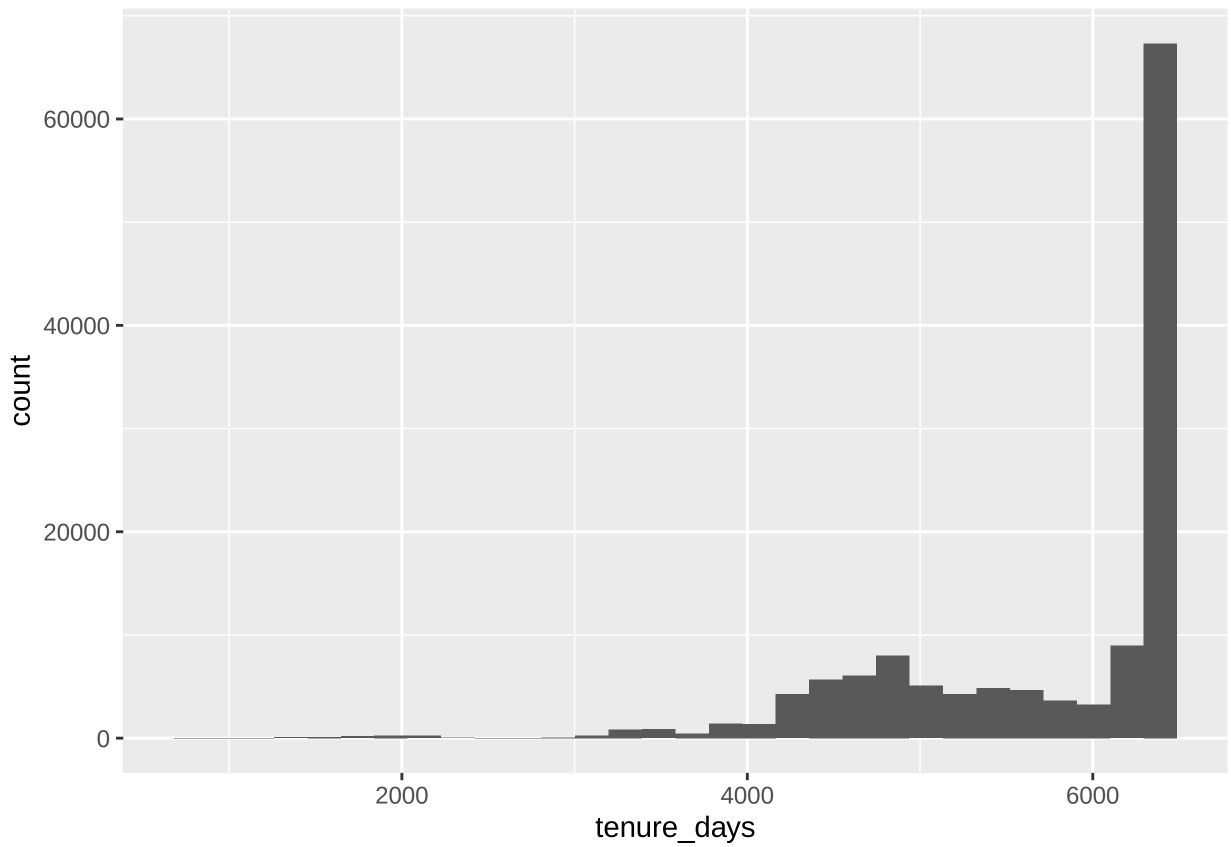
```
p2
```



```
p3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1058 rows containing non-finite values (`stat_bin()`).
```



Question 3

Create node lists for eacch work group

```
# join selected work groups with edges list
edges <- drop_na(edges, ego_examiner_id)
edges <-drop_na(edges, alter_examiner_id)
w179_2 <- inner_join(w179, edges, by = "application_number", copy = FALSE)
```

```
## Warning in inner_join(w179, edges, by = "application_number", copy = FALSE): Each row in ‘x‘ is expe
## i Row 109 of ‘x‘ matches multiple rows.
## i If multiple matches are expected, set ‘multiple = "all"‘ to silence this
##   warning.
```

```
w176_2 <- inner_join(w176, edges, by = "application_number", copy = FALSE)
```

```
## Warning in inner_join(w176, edges, by = "application_number", copy = FALSE): Each row in ‘x‘ is expe
## i Row 17239 of ‘x‘ matches multiple rows.
## i If multiple matches are expected, set ‘multiple = "all"‘ to silence this
##   warning.
```

```
# nodes dataframe of work groups and merge them
w179_nodes1 <- w179_2 %>%
  distinct(ego_examiner_id) %>%
  rename(ID = ego_examiner_id)
w179_nodes2 <- w179_2 %>%
  distinct(alter_examiner_id) %>%
  rename(ID = alter_examiner_id)
w176_nodes1 <- w176_2 %>%
  distinct(ego_examiner_id) %>%
  rename(ID = ego_examiner_id)
w176_nodes2 <- w176_2 %>%
  distinct(alter_examiner_id) %>%
  rename(ID = alter_examiner_id)
# merge the two dataframes for each work goup
w179_nodes <- union_all(w179_nodes1, w179_nodes2)
w176_nodes <- union_all(w176_nodes1, w176_nodes2)
w179_nodes <- unique(w179_nodes)
w176_nodes <- unique(w176_nodes)
head(w179_nodes, 5)
```

```
## # A tibble: 5 x 1
##      ID
##   <dbl>
## 1 61043
## 2 65547
## 3 60837
## 4 62778
## 5 72332
```

Create final edge list

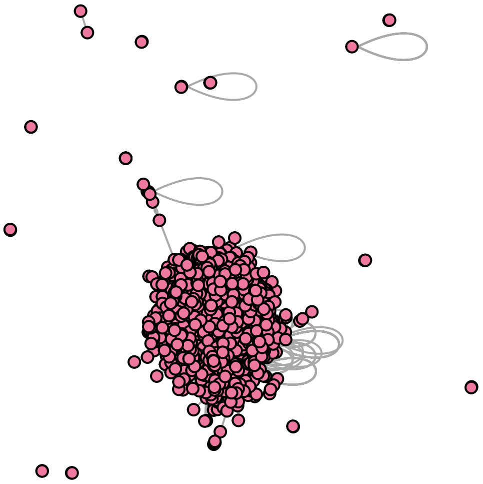
```
w179_edges <- w179_2 %>%
  select(ego_examiner_id, alter_examiner_id)
w176_edges <- w176_2 %>%
  select(ego_examiner_id, alter_examiner_id)
head(w179_edges, 5)
```

```
## # A tibble: 5 x 2
##   ego_examiner_id alter_examiner_id
##             <dbl>             <dbl>
## 1           61043             92569
## 2           65547             95660
## 3           65547             66762
## 4           60837             63938
## 5           60837             98804
```

```
g_w179 <- graph_from_data_frame(w179_edges, directed=FALSE)
g_w176 <- graph_from_data_frame(w176_edges, directed=FALSE)
```

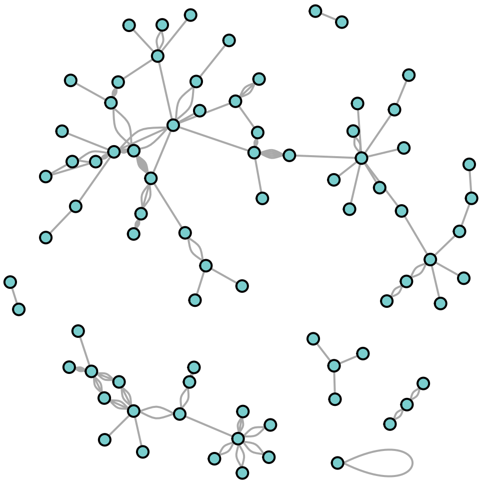
Plot vertex graph for work group 179

```
plot(g_w179, layout=layout.fruchterman.reingold,
    vertex.size = 5,
    vertex.label = NA,
    vertex.color = "palevioletred2")
```



Plot vertex graph for work group 176

```
plot(g_w176, layout=layout.fruchterman.reingold,
    vertex.size = 5,
    vertex.label = NA,
    vertex.color = "darkslategray3")
```



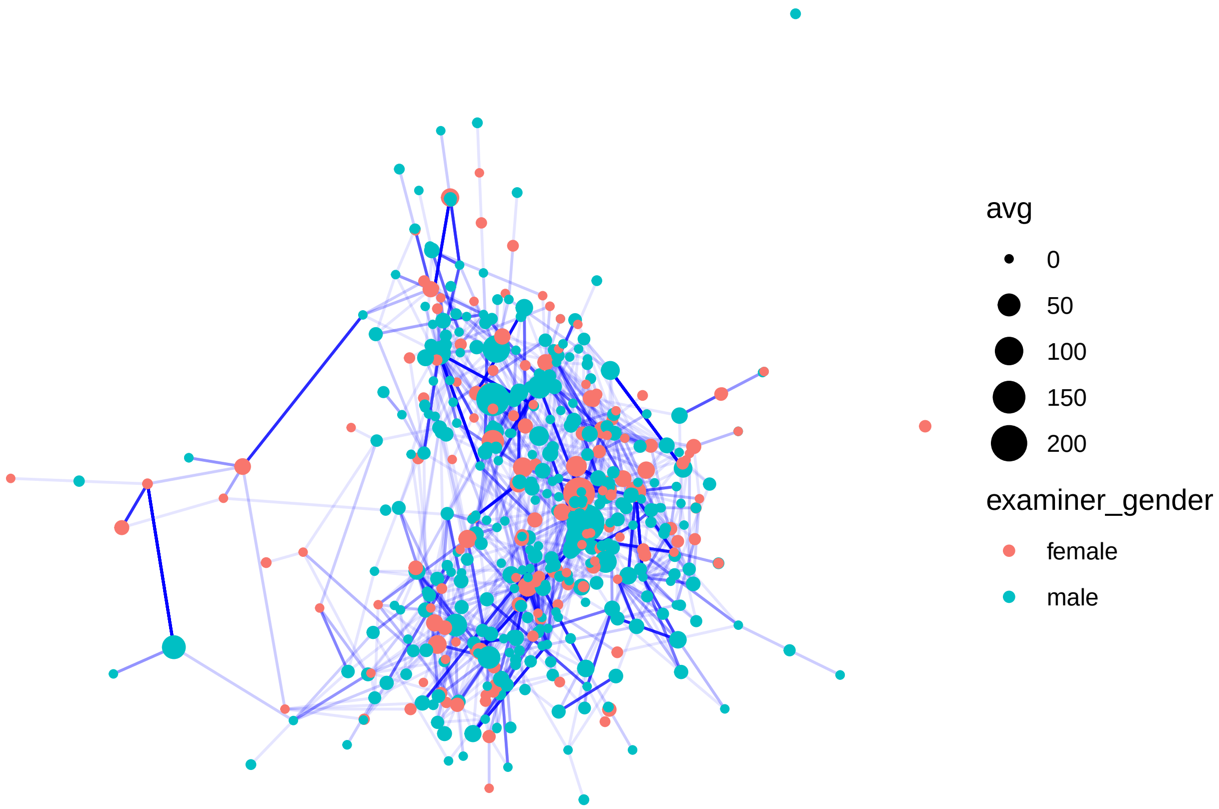
```
applications <- applications %>%
  mutate(examiner_workgroup = str_sub(examiner_art_unit, 1, -2))

applications <- applications %>% drop_na(gender, tenure_days, race)

examiner_data <- applications %>%
  distinct(examiner_id, examiner_gender = gender,
           examiner_race = race, examiner_tenure = tenure_days)

examiner_subset <- applications %>%
  filter(examiner_workgroup %in% c(179, 176)) %>%
  distinct(examiner_id, examiner_workgroup) %>%
  left_join(examiner_data, by='examiner_id')
```

Create a network

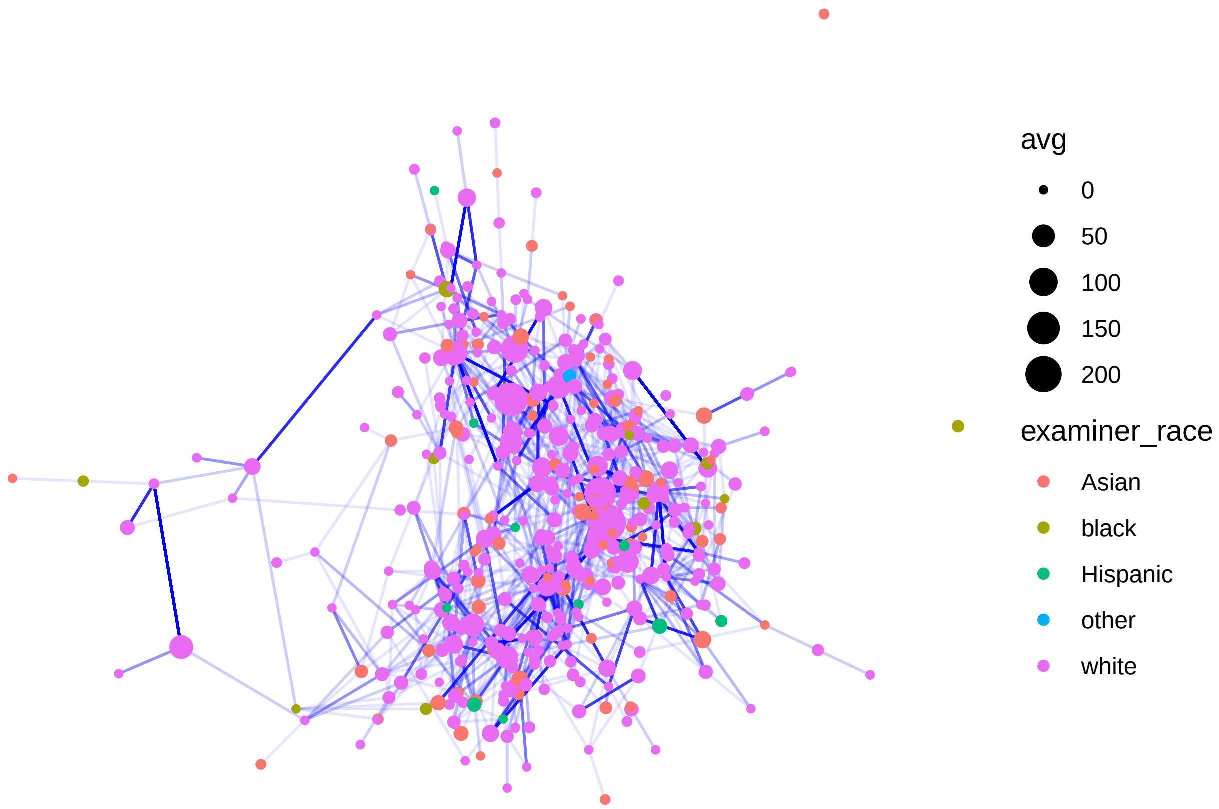
```
edge_subset <- edges %>%
  filter(ego_examiner_id %in% examiner_subset$examiner_id &
           alter_examiner_id %in% examiner_subset$examiner_id) %>%
  drop_na() %>%
  select(to = ego_examiner_id, from = alter_examiner_id)
node_subset <- edge_subset %>%
  pivot_longer(cols=c('from','to')) %>%
  distinct(examiner_id = value) %>%
  left_join(examiner_data, by='examiner_id') %>%
  distinct(examiner_id, examiner_gender, examiner_race, examiner_tenure) %>%
  rename(name = examiner_id) %>%
  mutate(name = as.character(name))
network <- graph_from_data_frame(edge_subset, directed = TRUE) %>%
  as_tbl_graph() %>%
  left_join(node_subset, by='name')


network <- network %>%
  mutate(degree = centrality_degree(),
         betweenness = centrality_betweenness()) %>%
  mutate(avg = (degree + betweenness)/2) %>%
  mutate(label = paste0(name, '\n',
                        'Degree: ',round(degree,2), '\n',
                        'Betweenness: ',round(betweenness,2), '\n',
                        'Avg: ',round(avg,2)))

set.seed(1)
net_gender <- network %>%
  ggraph(layout="mds") +
  geom_edge_link(edge_colour = "#0000FF", alpha=0.1) +
  geom_node_point(aes(color=examiner_gender, size=avg)) +
  theme_void()
set.seed(1)
net_race <- network %>%
  ggraph(layout="mds") +
  geom_edge_link(edge_colour = "#0000FF", alpha=0.1) +
  geom_node_point(aes(color=examiner_race, size=avg)) +
  theme_void()
```

```
net_gender
```

```
## Warning: Using the `size` aesthetic in this geom was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` in the `default_aes` field and elsewhere instead.
```



```
net_race
```



Calculate centralities

```
# betweenness
bc_w179 <- sort(betweenness(g_w179), decreasing = TRUE)
bc_w176 <- sort(betweenness(g_w176), decreasing = TRUE)
# degree
dg_w179 <- sort(degree(g_w179), decreasing = TRUE)
dg_w176 <- sort(degree(g_w176), decreasing = TRUE)
# closeness
cc_w179 <- sort(closeness(g_w179), decreasing = TRUE)
cc_w176 <- sort(closeness(g_w176), decreasing = TRUE)
print("top 5 of betwenness centrality for work group 179")
```

```
## [1] "top 5 of betwenness centrality for work group 179"
```

```
print(head(bc_w179,5))
```

```
##     92569     77068     71119     66910     63176
## 163432.47 161656.68  58206.76  57543.18  55264.72
```

```
print("top 5 of betwenness centrality for work group 176")
```

```
## [1] "top 5 of betwenness centrality for work group 176"
```

```
print(head(bc_w176,5))
```

```
##    72809    96532    91824    75387    99845
## 736.4535 643.0000 621.2857 558.0000 328.7524
```

```
print("top 5 of degree centrality for work group 179")
```

```
## [1] "top 5 of degree centrality for work group 179"
```

```
print(head(dg_w179,5))
```

```
## 91824 92569 93896 77648 71353
##   239   197   162   160   150
```

```
print("top 5 of degree centrality for work group 176")
```

```
## [1] "top 5 of degree centrality for work group 176"
```

```
print(head(dg_w176,5))
```

```
## 99845 77648 71353 98763 91824
##    23    17    15    14    13
```

```
print("top 5 of closeness centrality for work group 179")
```

```
## [1] "top 5 of closeness centrality for work group 179"
```

```
print(head(cc_w179,15))
```

```
## 76532 95721 98518 65307 60106 62506 71760 93933 59454 65329 78905 70843 78231
##     1     1     1     1     1     1     1     1     1     1     1     1     1
## 59616 67762
##     1     1
```

```
print("top 5 of closeness centrality for work group 176")
```

```
## [1] "top 5 of closeness centrality for work group 176"
```

```
print(head(cc_w176,15))
```

```
##      67331      94543      66450      92569      73692      93896      77068
## 1.00000000 1.00000000 1.00000000 1.00000000 0.50000000 0.33333333 0.33333333
##      94899      72112      98297      88291      63363      85449      71143
## 0.33333333 0.20000000 0.20000000 0.20000000 0.02941176 0.02857143 0.02564103
##      67904
## 0.02272727
```

My Choice of Measures:

Patent examination is a complex task that involves coordinating and communicating with other examiners, applicants, and stakeholders. Examining patents is also a highly specialized field, and patent examiners often work in specific technology areas. Therefore, it is important to understand the centrality of patent examiners in their workgroups to identify potential bottlenecks or inefficiencies in the examination process.

I picked degree centrality because it can be used to identify examiners who are highly connected to other examiners in their workgroup. These examiners are likely to be important in terms of sharing information and knowledge within the group, and they may also be influential in terms of decision-making or providing feedback to other examiners.

Betweenness centrality can be used to identify examiners who act as intermediaries or connectors between different technology areas or subgroups within the workgroup. These examiners may play a critical role in facilitating communication and information flow between different parts of the group and ensuring that the examination process is efficient.

Closeness centrality can be used to identify examiners who are well-positioned to receive and disseminate information within their workgroup. These examiners are likely to have a good understanding of what is happening within the group and may be able to provide valuable feedback to other examiners.

By analyzing the centrality measures of examiners within workgroups, we can gain insights into how the examination process is working and identify potential areas for improvement. For example, if certain examiners have low centrality measures, it may indicate that they are not as well-connected or influential within the group, and they may benefit from more communication or training. On the other hand, if certain examiners have very high centrality measures, it may indicate that they are overloaded with work or that the examination process is too dependent on them, and efforts may be needed to redistribute workload or improve communication within the group.

Taken together, these centrality measures can provide a more nuanced understanding of the dynamics and functioning of workgroups within the USPTO. By identifying examiners who are particularly influential or central within the group, we can better understand the social structure and flow of work within the group, and potentially even identify areas for improvement or intervention.

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

For Work Group 179, we can see that the top 5 nodes with the highest betweenness centrality are all males, with tenure days ranging from 6342 to 6391. This indicates that these individuals play a critical role in connecting different nodes in the network and facilitating communication and information flow between different subgroups within the work group. In terms of race, all five individuals are White, which suggests that individuals from this racial group may have a higher level of influence and power within the organization.

For degree centrality, we see that the top 5 nodes are also dominated by males, with one female (node 93896) included. These individuals have the highest number of connections to other nodes within the network, which suggests that they may be important sources of information and knowledge for other members of the work group. Interestingly, the top 5 nodes with the highest degree centrality do not overlap with those with the highest betweenness centrality, which suggests that there may be different types of influential individuals within the network.

For closeness centrality, we see that the top nodes are predominantly females, with one male (node 76532) included. These individuals have the shortest paths to other nodes within the network, which suggests that they may be well-positioned to receive and disseminate information quickly and efficiently. Interestingly, the top nodes with the highest closeness centrality do not overlap with those with the highest betweenness or degree centrality, which again suggests that there may be different types of influential individuals within the network.

For Work Group 176, we see a slightly different pattern. The top nodes with the highest betweenness centrality are all males, with the highest value (72809) being much smaller than those for Work Group 179. This suggests that there may be less variation in the extent to which different individuals are able to facilitate communication and information flow within this work group. In terms of race, all top nodes are either Asian or White, with no Black or Hispanic individuals included.

For degree centrality, we see that the top nodes are dominated by nodes from Work Group 179, which may reflect the fact that individuals in this work group have more connections to other nodes within the overall network. Interestingly, the top nodes with the highest degree centrality do not overlap with those with the highest betweenness centrality, which again suggests that there may be different types of influential individuals within the network.

For closeness centrality, we see that the top nodes are all either males or females with NA values for gender. These individuals have the shortest paths to other nodes within the network, which suggests that they may be well-positioned to receive and disseminate information quickly and efficiently. However, the top nodes with the highest closeness centrality do not overlap with those with the highest betweenness or degree centrality, which suggests that there may be different types of influential individuals within the network.

Overall, these findings suggest that there are complex relationships between different types of centrality measures and other examiners' characteristics such as gender, race, and tenure days. While some patterns emerge, there is also a degree of heterogeneity in terms of which individuals are most influential within each work group, suggesting that different types of centrality measures may capture different aspects of influence and power within the network.

A Deeper Dive on Demographics

Gender:

Looking at the gender breakdown of the two work groups, we can see that work group 179 has more female examiners (43,783) than male examiners (77,344), while work group 176 has more male examiners (53,561) than female examiners (28,075). Interestingly, the top 5 betweenness centrality scores for work group 179 are all male examiners, while the top 5 for work group 176 are a mix of male and female examiners. This could suggest that male examiners in work group 179 may have more influence or play a more important role in the overall communication and collaboration patterns within the group.

Race:

The racial breakdown of the two work groups shows that work group 179 has a higher proportion of white examiners (98,845) than any other race, while work group 176 has a relatively even distribution of white (61,824) and Asian (23,022) examiners. Looking at the betweenness centrality scores, we see that the top 5 for work group 179 are all white examiners, while the top 5 for work group 176 are a mix of Asian and white examiners. This could suggest that white examiners in both work groups may have more influence or play a more important role in communication and collaboration patterns within their respective groups.

Tenure:

Examining the tenure distribution for the two work groups, we see that work group 179 has a lower mean tenure (5,712 days) than work group 176 (5,501 days). Interestingly, the top 5 degree centrality scores for work group 179 include examiners with both relatively high tenure (e.g. examiner 91824 with a degree centrality score of 239) and lower tenure (e.g. examiner 92569 with a degree centrality score of 197). In contrast, the top 5 degree centrality scores for work group 176 all belong to examiners with relatively low tenure (ranging from 13 to 23 days). This could suggest that in work group 179, examiners with both high and low tenure levels are equally important for communication and collaboration, while in work group 176, newer examiners may be more important for these patterns.