Project

Libraries

```
library(ggraph)
library(igraph)

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

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

library(stargazer)
```

Data cleaning & Preprocessing section

Data

```
data_path <- "Data/"
applications <- read_parquet(paste0(data_path,"app_data_sample.parquet"))
edges <- read_csv(paste0(data_path,"edges_sample.csv"))
```

Add gender

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

# 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
  )
# remove extra columns 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  4714048 251.8    8247184 440.5  4733501 252.8
## Vcells 49754831 379.6   95716811 730.3 80070355 610.9
```

Add race

```
# get list of distinct last names
examiner_surnames <- applications %>%
  select(surname = examiner_name_last) %>%
  distinct()

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

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

```
# infer racial probabilities from surname tibble
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_
  ))

# removing extra columns and merge into application data
examiner_race <- examiner_race %>%
  select(surname,race)

applications <- applications %>%
  left_join(examiner_race, by = c("examiner_name_last" = "surname"))
```

```r
# cleanup
rm(examiner_race)
rm(examiner_surnames)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  5054059 270.0    8247184 440.5  5694182 304.2
## Vcells 53441388 407.8   95716811 730.3 94229196 719.0
```

Add tenure

```r
# get all application filing dates
examiner_dates <- applications %>%
  select(examiner_id, filing_date, appl_status_date)

# calculate start and end date from filing / status date respectively
examiner_dates <- examiner_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))

# for each examiner, get earliest and latest days, then interval between them as tenure in days
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)

# merge and clean
applications <- applications %>%
  left_join(examiner_dates, by = "examiner_id")

rm(examiner_dates)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  5068109 270.7   14721424  786.3  14721424  786.3
## Vcells 65820148 502.2  138008207 1053.0 137878248 1052.0
```

Add application duration

```r
# Since an application can only be issued or abandoned, one or the other will always be NA, therefore I

applications$appl_end_date <- paste(applications$patent_issue_date, applications$abandon_date, sep=',')

# Then I will clean up the column by removing instances of commas and NA's
applications$appl_end_date <- gsub('NA', "", as.character(applications$appl_end_date))
applications$appl_end_date <- gsub(',', "", as.character(applications$appl_end_date))
```

```r
# Ensure date format is consistent for both columns
applications$appl_end_date <- as.Date(applications$appl_end_date, format="%Y-%m-%d")
applications$filing_date <- as.Date(applications$filing_date, format="%Y-%m-%d")

# Finding the difference in days between the application end date and the filing date
applications$appl_proc_days <- as.numeric(difftime(applications$appl_end_date, applications$filing_date

# Remove instances where the filing date happens after the issue or abandon dates (these must be mistak
applications <- applications %>% filter(appl_proc_days >=0 & !is.na(appl_proc_days))

gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  4738684 253.1   14721424  786.3  14721424  786.3
## Vcells 61613026 470.1  138700712 1058.3 138700712 1058.3
```

Check completeness of the dataset to this point

```r
library(skimr)
applications %>% skim()
```

Table 1: Data summary

Name	Piped data
Number of rows	1688681
Number of columns	23
Column type frequency:	
character	11
Date	6
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1.00	8	8	0	1688681	0
examiner_name_last	0	1.00	2	17	0	3747	0
examiner_name_first	0	1.00	1	12	0	2549	0
examiner_name_middle	390396	0.77	1	12	0	512	0
uspc_class	4	1.00	3	3	0	413	0
uspc_subclass	1555	1.00	6	6	0	6093	0
patent_number	601857	0.64	4	7	0	1086823	0
disposal_type	0	1.00	3	3	0	2	0
appl_status_date	356	1.00	18	18	0	5680	0
gender	253871	0.85	4	6	0	2	0
race	0	1.00	5	8	0	5	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
filing_date	0	1.00	2000-01-02	2017-03-24	2008-03-14	6045
patent_issue_date	601383	0.64	2000-06-06	2017-06-20	2012-05-22	890
abandon_date	1087295	0.36	2000-03-07	2050-06-30	2011-04-19	5040
earliest_date	18240	0.99	2000-01-02	2015-02-26	2000-05-12	2244
latest_date	18240	0.99	2000-09-14	2017-12-06	2017-05-20	865
appl_end_date	0	1.00	2000-03-07	2050-06-30	2011-12-27	5053

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	3746	1.00	78650.65	13611.68	59012	66481	75149	93760	99990	
examiner_art_unit	0	1.00	1918.68	300.06	1600	1657	1771	2166	2498	
appl_status_code	355	1.00	164.38	30.73	16	150	150	161	854	
tc	0	1.00	1867.83	294.43	1600	1600	1700	2100	2400	
tenure_days	18240	0.99	5636.92	987.07	216	5128	6184	6337	6518	
appl_proc_days	0	1.00	1190.28	620.63	0	765	1079	1481	17898	

Given that our goal is to measure the relationship between centrality and application processing time, there are a few variables here that may be worth imputing to remove NaNs.

- Gender
- tenure days
- appl days

We will use R's mice package which performs multiple imputation under the assumption that any missing data is 'Missing At Random' ie the probability that a value is missing depends only on the observed value itself. Mice will impute data for each input variable by specifying a unique imputation model per-variable. Ie if our feature set consists of X1, X2, … Xn and X1 has missing values, it will be imputed based on the patterns observed in X2….Xn.

Before we do this, we have to remove some variables which may be missing not-at-random, or are deemed to be unhelpful for the later modelling stage.

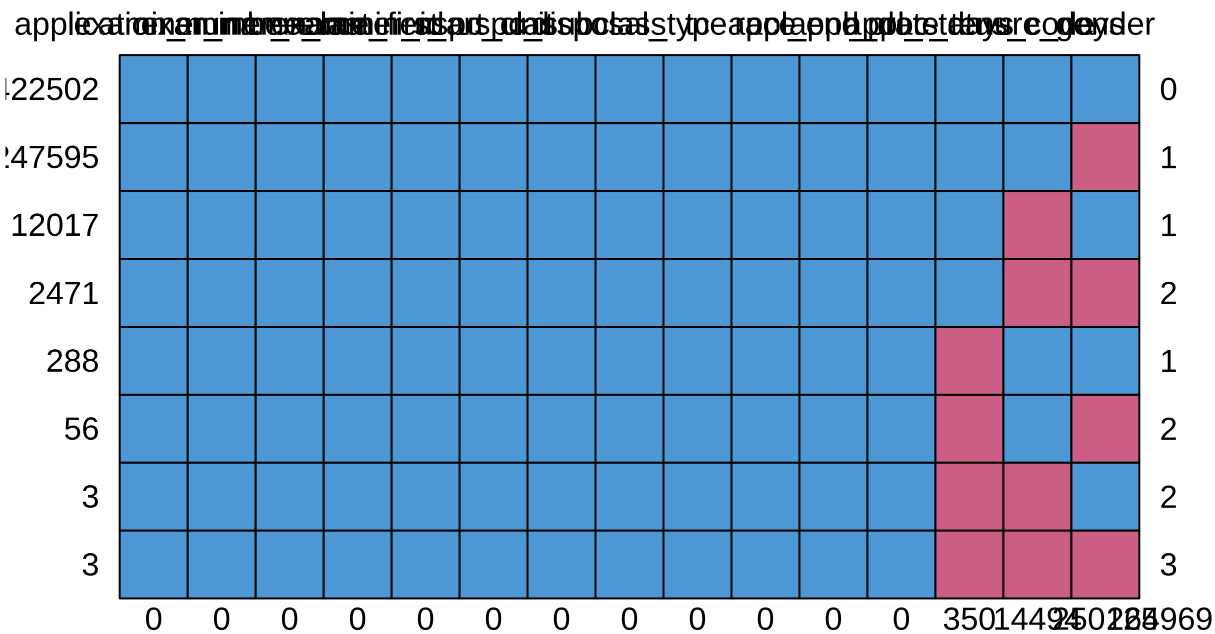
```
applications_subs = subset(applications, select=-c(examiner_name_middle,patent_number, appl_status_date
# Removal explanations:

# some people might not have a middle name by choice (ie it was not just randomly forgotten to be enter
# missing patent number means no patent issues, not missing at random
# appl_status_date for the same reason as patent number, and all of the related date-measurements arisi

# we remove the remaining date columns since we already have the metrics we need from them (tenure and 

# we want examiner_id to remain unique which will not be the case if we allow mice to impute it, so we 
applications_subs = applications_subs %>% drop_na(examiner_id)
```

```
library(mice)
md.pattern(applications_subs)
```




```
##         application_number examiner_name_last examiner_name_first examiner_id
## 1422502                  1                  1                   1           1
## 247595                   1                  1                   1           1
## 12017                    1                  1                   1           1
## 2471                     1                  1                   1           1
## 288                      1                  1                   1           1
## 56                       1                  1                   1           1
## 3                        1                  1                   1           1
## 3                        1                  1                   1           1
##                          0                  0                   0           0
##         examiner_art_unit uspc_class uspc_subclass disposal_type tc race
## 1422502                 1          1             1             1  1    1
## 247595                  1          1             1             1  1    1
## 12017                   1          1             1             1  1    1
## 2471                    1          1             1             1  1    1
## 288                     1          1             1             1  1    1
## 56                      1          1             1             1  1    1
## 3                       1          1             1             1  1    1
## 3                       1          1             1             1  1    1
##                         0          0             0             0  0    0
##         appl_end_date appl_proc_days appl_status_code tenure_days gender
## 1422502             1              1                1           1      1      0
## 247595              1              1                1           1      0      1
## 12017               1              1                1           0      1      1
## 2471                1              1                1           0      0      2
## 288                 1              1                0           1      1      1
```

```
## 56                    1              1                0           1       0       2
## 3                     1              1                0           0       1       2
## 3                     1              1                0           0       0       3
##                       0              0              350       14494  250125  264969
```

```
# there are 1696847 observations with no missing values (84% of the dataset)
# another 14% has just one missing value (gender)
# the remaining 2% of missing values is composed of the other features

applications_subs$gender = as.factor(applications_subs$gender) # mice will only impute on categorically

applications_full = complete(mice(applications_subs, m=3, maxit=3)) # impute using default mice imputat
```

```
##
##  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
```

```
rm(applications_subs)

applications_full %>% skim() # all done
```

Table 5: Data summary

Name	Piped data
Number of rows	1684935
Number of columns	15
Column type frequency:	
character	7
Date	1
factor	1
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1	8	8	0	1684935	0
examiner_name_last	0	1	2	17	0	3746	0
examiner_name_first	0	1	1	12	0	2548	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
uspc_class	0	1	3	3	0	412	0
uspc_subclass	0	1	6	6	0	6090	0
disposal_type	0	1	3	3	0	2	0
race	0	1	5	8	0	5	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
appl_end_date	0	1	2000-04-07	2050-06-30	2011-12-27	5003

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	mal: 1134112, fem: 550823

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	0	1	78650.65	13611.68	59012	66481	75149	93760	99990	
examiner_art_unit	0	1	1918.94	300.12	1600	1657	1771	2166	2498	
appl_status_code	0	1	164.39	30.75	18	150	150	161	854	
tc	0	1	1868.08	294.48	1600	1600	1700	2100	2400	
tenure_days	0	1	5638.21	986.17	216	5131	6185	6337	6518	
appl_proc_days	0	1	1192.41	619.59	0	768	1081	1482	17898	

With our remaining values imputed, we can proceed with demographics & constructing our advice network and calculating centralities

---

Demographics Insights

```
# filter for unique examiners only
final <- distinct(applications_full, examiner_id, .keep_all = TRUE)

# isolate for specific workgroups
final$wg = substr(final$examiner_art_unit, 1,3)

# create dataframes consisting of our specific workgroups
WG_219 <- final[final$wg == 219, ]
WG_162 <- final[final$wg == 162, ]
```

Race

Summarize Race Distribution by Working Group

```r
# obtain raw percentage of race by working group
WG219_Race <- WG_219 %>%
  group_by(race) %>%
  summarise(WorkGroup = "Work Group 219", count = n()) %>%
  mutate(percentage  = round(count / sum(count), 2)) %>%
  arrange(desc(percentage))
head(WG219_Race)
```

```
## # A tibble: 4 x 4
##   race     WorkGroup      count percentage
##   <chr>    <chr>          <int>      <dbl>
## 1 Asian    Work Group 219    86       0.49
## 2 white    Work Group 219    72       0.41
## 3 Hispanic Work Group 219    11       0.06
## 4 black    Work Group 219     5       0.03
```

```r
WG162_Race <- WG_162 %>%
  group_by(race) %>%
  summarise(WorkGroup = "Work Group 162", count = n()) %>%
  mutate(percentage = round(count / sum(count), 2)) %>%
  arrange(desc(percentage))
head(WG162_Race)
```

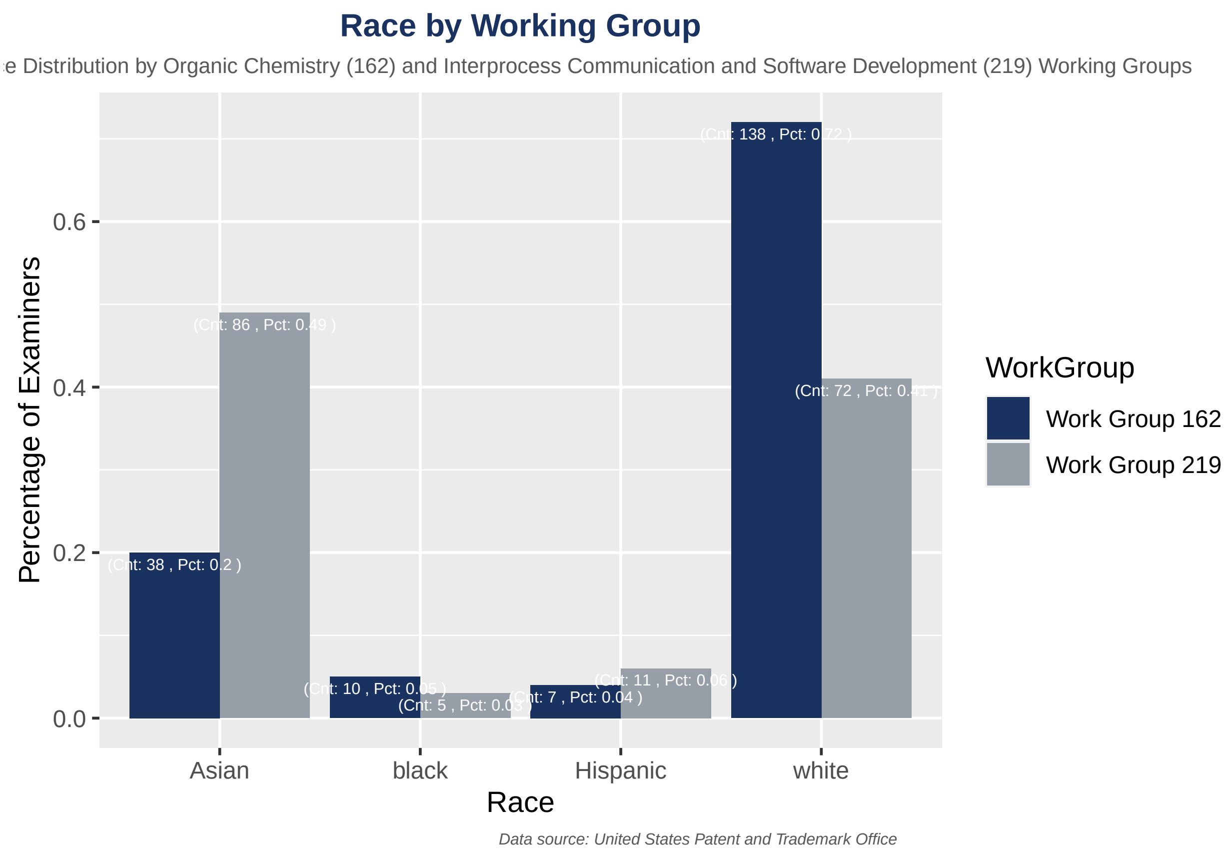
```
## # A tibble: 4 x 4
##   race     WorkGroup      count percentage
##   <chr>    <chr>          <int>      <dbl>
## 1 white    Work Group 162   138       0.72
## 2 Asian    Work Group 162    38       0.2
## 3 black    Work Group 162    10       0.05
## 4 Hispanic Work Group 162     7       0.04
```

```r
# join both together
comps_perc <- rbind(WG219_Race, WG162_Race)
```

Visualization of Race by Working Group

```r
# visualization of race by working group as a function of percentage
ggplot(comps_perc, aes(x=race, y=percentage, fill=WorkGroup)) +
  geom_bar(stat="identity", position="dodge") +
  # specify the color palette
  scale_fill_manual(values=c("#1a3260","#969fa7")) +
  labs(title = "Race by Working Group",
       subtitle= "Overview of Race Distribution by Organic Chemistry (162) and Interprocess Communicati
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Percentage of Examiners") +
```

```
  xlab("Race") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9)) +
  # add labels
  geom_text(aes(label=paste("(Cnt:",count,", Pct:",percentage,")")),
            colour = "white", size = 2,
            vjust = 1.5, position = position_dodge(.9))
```



Gender

Summarize Gender Distribution by Working Group

```
# obtain raw count and percentage of gender by working group
WG219_Gender <- WG_219 %>%
  group_by(gender) %>%
  summarise(WorkGroup = "Work Group 219", count = n()) %>%
  mutate(percentage = round(count / sum(count), 2)) %>%
  arrange(desc(percentage))
head(WG219_Gender)
```

```
## # A tibble: 2 x 4
```

```
##   gender WorkGroup      count percentage
##   <fct>  <chr>          <int>      <dbl>
## 1 male   Work Group 219   144       0.83
## 2 female Work Group 219    30       0.17
```

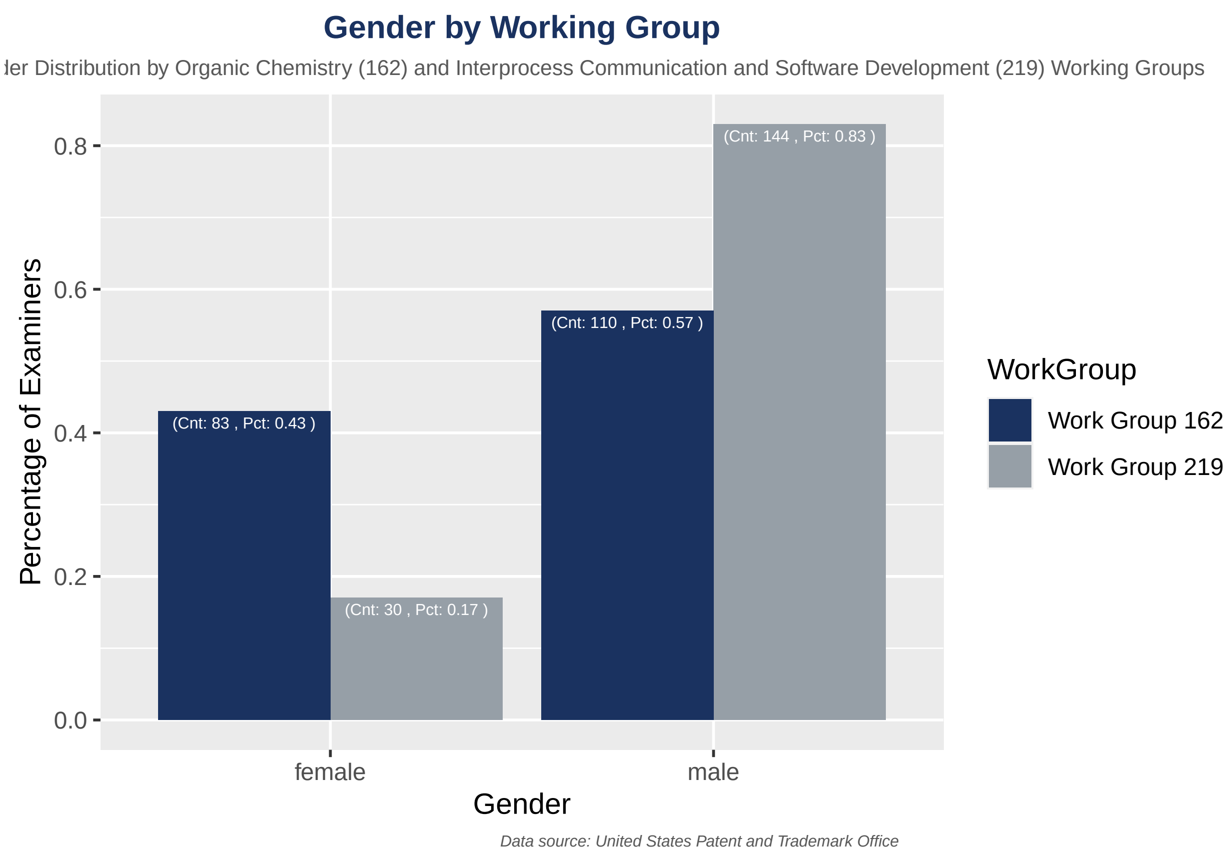
```
WG162_Gender <- WG_162 %>%
  group_by(gender) %>%
  summarise(WorkGroup = "Work Group 162", count = n()) %>%
  mutate(percentage = round(count / sum(count), 2)) %>%
  arrange(desc(percentage))
head(WG162_Gender)
```

```
## # A tibble: 2 x 4
##   gender WorkGroup      count percentage
##   <fct>  <chr>          <int>      <dbl>
## 1 male   Work Group 162   110       0.57
## 2 female Work Group 162    83       0.43
```

```
# join both together
comps_perc <- rbind(WG219_Gender, WG162_Gender)
```

Visualization of Gender by Working Group

```
# visualization of race by working group as a function of percentage
ggplot(comps_perc, aes(x=gender, y=percentage, fill=WorkGroup)) +
  geom_bar(stat="identity", position="dodge") +
  # specify the color palette
  scale_fill_manual(values=c("#1a3260","#969fa7")) +
  labs(title = "Gender by Working Group",
       subtitle= "Overview of Gender Distribution by Organic Chemistry (162) and Interprocess Communica
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Percentage of Examiners") +
  xlab("Gender") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9)) +
  # add labels
  geom_text(aes(label=paste("(Cnt:",count,", Pct:",percentage,")")),
            colour = "white", size = 2,
            vjust = 1.5, position = position_dodge(.9))
```



Tenure

Summarize Examiner Tenure by Race and Gender

```
# generate new variable which looks at average tenure days by race and gender
WG219_GenderRace <- WG_219 %>%
  group_by(gender, race) %>%
  summarise(tenure_days = mean(tenure_days), count = n())

WG162_GenderRace <- WG_162 %>%
  group_by(gender, race) %>%
  summarise(tenure_days = mean(tenure_days), count = n())

# generate a new variable to describe the workgroups
WG219_GenderRace$WorkGroup <- "Work Group 219"
WG162_GenderRace$WorkGroup <- "Work Group 162"

# rename gender for more clear visualization
WG219_GenderRace <- WG219_GenderRace %>%
    mutate(gender = recode(gender, female = "FM", male = "ML"))
WG162_GenderRace <- WG162_GenderRace %>%
    mutate(gender = recode(gender, female = "FM", male = "ML"))

# create new variable that is a combination of both
```

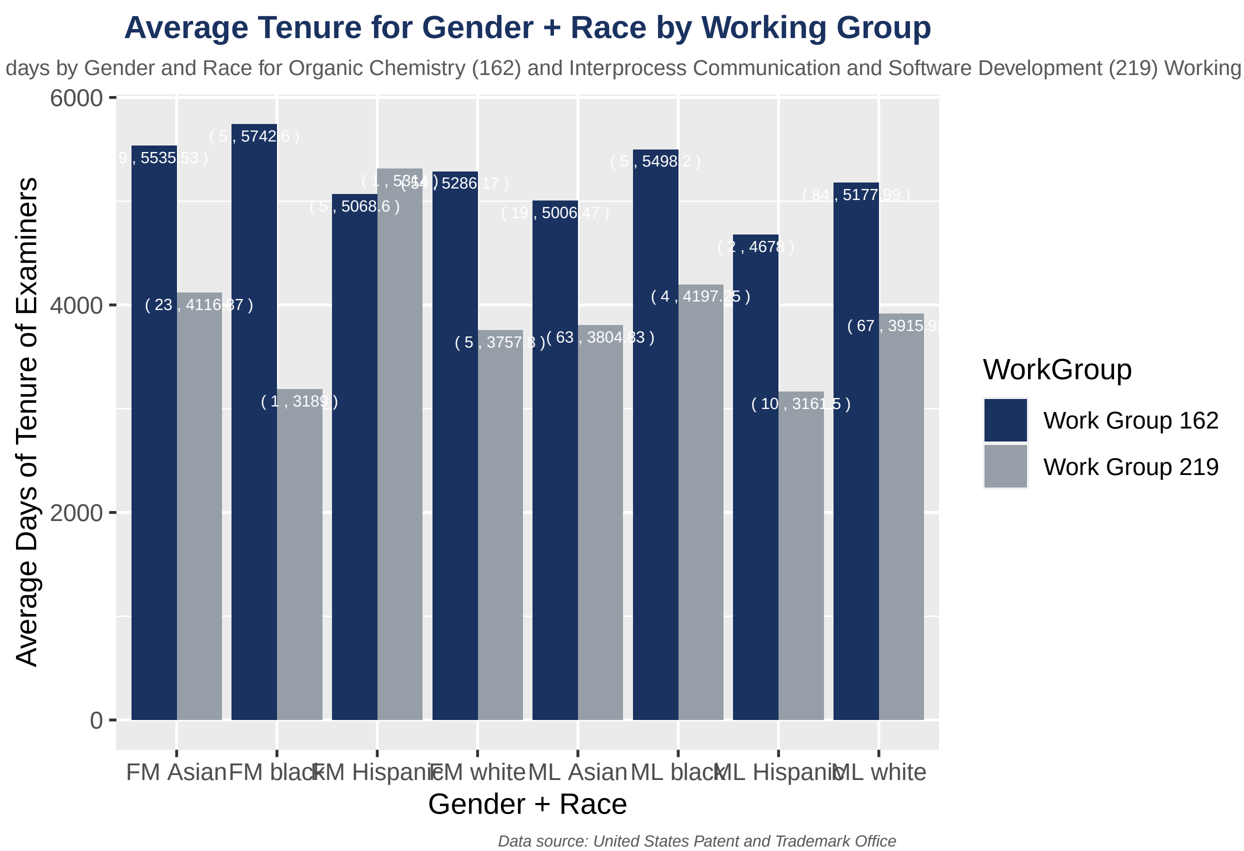
```r
WG219_GenderRace$gender_race <- paste(WG219_GenderRace$gender, WG219_GenderRace$race)
WG162_GenderRace$gender_race <- paste(WG162_GenderRace$gender, WG162_GenderRace$race)

# round the average tenure days
WG219_GenderRace$tenure_days <- round(WG219_GenderRace$tenure_days,digit=2)
WG162_GenderRace$tenure_days <- round(WG162_GenderRace$tenure_days,digit=2)

# add together
aggregate <- rbind(WG219_GenderRace, WG162_GenderRace)
```

Visualization of Tenure by Race and Gender

```r
ggplot(aggregate, aes(x=gender_race, y=tenure_days, fill=WorkGroup)) +
  geom_bar(stat="identity", position="dodge") +
  # specify the color palette
  scale_fill_manual(values=c("#1a3260","#969fa7")) +
  labs(title = "Average Tenure for Gender + Race by Working Group",
       subtitle= "Overview of Average tenure days by Gender and Race for Organic Chemistry (162) and In
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Average Days of Tenure of Examiners") +
  xlab("Gender + Race") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9)) +
  # add labels
  geom_text(aes(label=paste("(",count,",",tenure_days,")")),
            colour = "white", size = 2,
            vjust = 1.5, position = position_dodge(.9))
```



Processing Days

Summarize Examiner Processing Days by Race and Gender

```r
# generate new variable which looks at average processing days by race and gender
WG219_GenderRace <- WG_219 %>%
   group_by(gender, race) %>%
   summarise_at(vars("appl_proc_days"), mean)
WG162_GenderRace <- WG_162 %>%
   group_by(gender, race) %>%
   summarise_at(vars("appl_proc_days"), mean)

# generate a new variable to describe the workgroups
WG219_GenderRace$WorkGroup <- "Work Group 212"
WG162_GenderRace$WorkGroup <- "Work Group 162"

# rename gender for more clear visualization
WG219_GenderRace <- WG219_GenderRace %>%
    mutate(gender = recode(gender, female = "FM", male = "ML"))
WG162_GenderRace <- WG162_GenderRace %>%
    mutate(gender = recode(gender, female = "FM", male = "ML"))

# create new variable that is a combination of both
WG219_GenderRace$gender_race <- paste(WG219_GenderRace$gender, WG219_GenderRace$race)
```

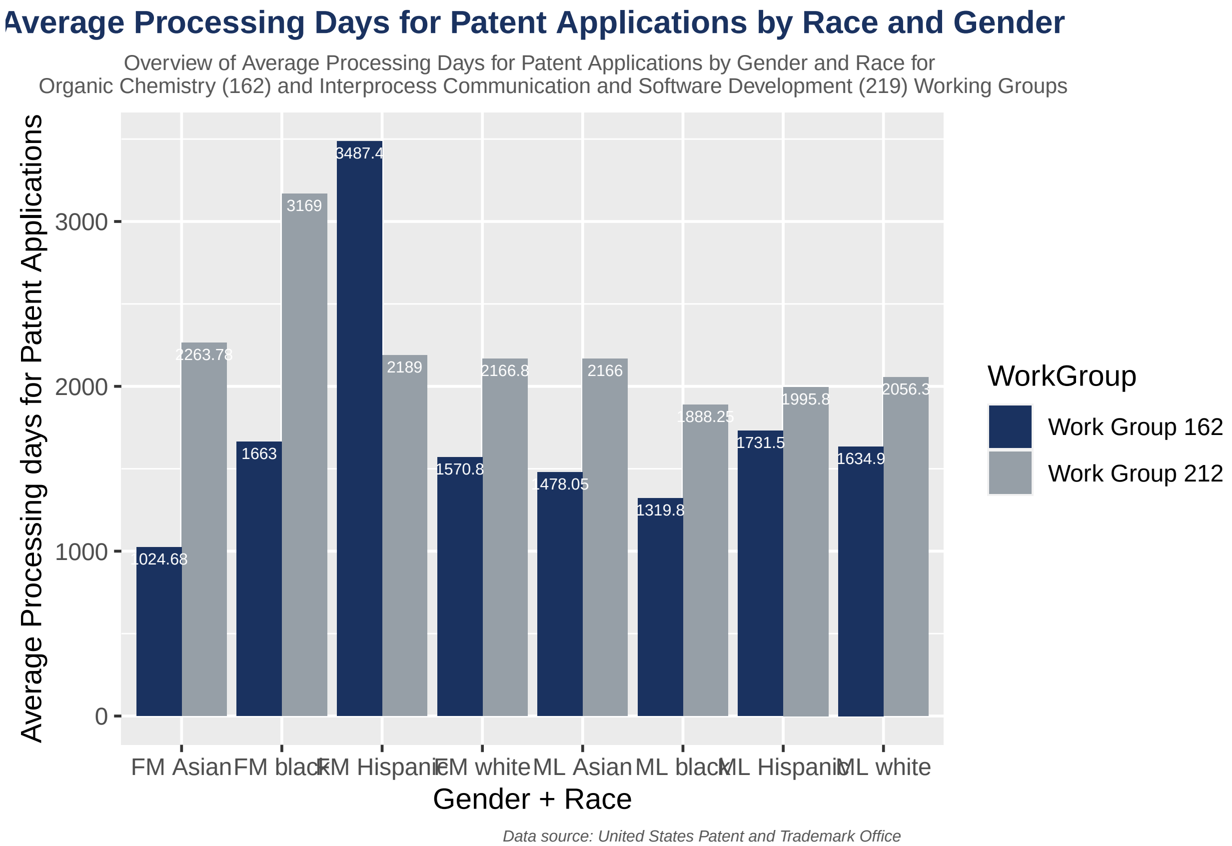
```r
WG162_GenderRace$gender_race <- paste(WG162_GenderRace$gender, WG162_GenderRace$race)

# round the average tenure days
WG219_GenderRace$appl_proc_days <- round(WG219_GenderRace$appl_proc_days,digit=2)
WG162_GenderRace$appl_proc_days <- round(WG162_GenderRace$appl_proc_days,digit=2)

# add together
aggregate <- rbind(WG219_GenderRace, WG162_GenderRace)
```

Visualization of Processing by Working Group, Race and Gender

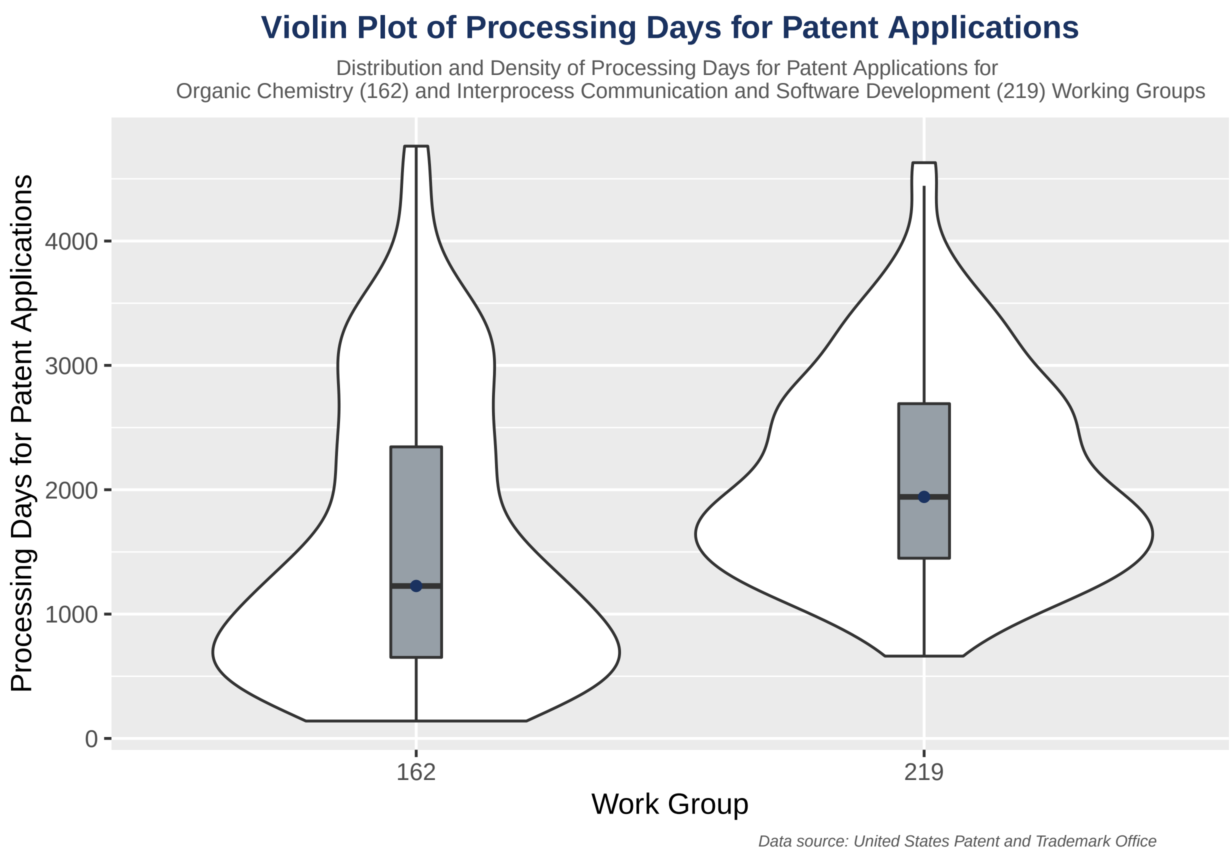
```r
aggregate <- rbind(WG219_GenderRace, WG162_GenderRace)
ggplot(aggregate, aes(x=gender_race, y=appl_proc_days, fill=WorkGroup)) + geom_bar(stat="identity", pos
  # specify color
  scale_fill_manual(values=c("#1a3260","#969fa7")) +
  labs(title = "Average Processing Days for Patent Applications by Race and Gender",
       subtitle= "Overview of Average Processing Days for Patent Applications by Gender and Race for
       Organic Chemistry (162) and Interprocess Communication and Software Development (219) Working Gr
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Average Processing days for Patent Applications") +
  xlab("Gender + Race") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9)) +
  # add labels
  geom_text(aes(label = appl_proc_days),
            colour = "white", size = 2,
            vjust = 1.5, position = position_dodge(.9))
```



Processing Days Overview

```
# isolate for working groups
combined <- subset(final, wg==219 | wg==162)

# violin plot
ggplot(combined, aes(wg, appl_proc_days)) +
  geom_violin() +
  geom_boxplot(width = .1, fill = "#969fa7", outlier.shape = NA) +
  stat_summary(fun.y = "median", geom = "point", col = "#1a3260") +
  labs(title = "Violin Plot of Processing Days for Patent Applications",
       subtitle= "Distribution and Density of Processing Days for Patent Applications for
       Organic Chemistry (162) and Interprocess Communication and Software Development (219) Working Gro
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Processing Days for Patent Applications") +
  xlab("Work Group") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9))
```

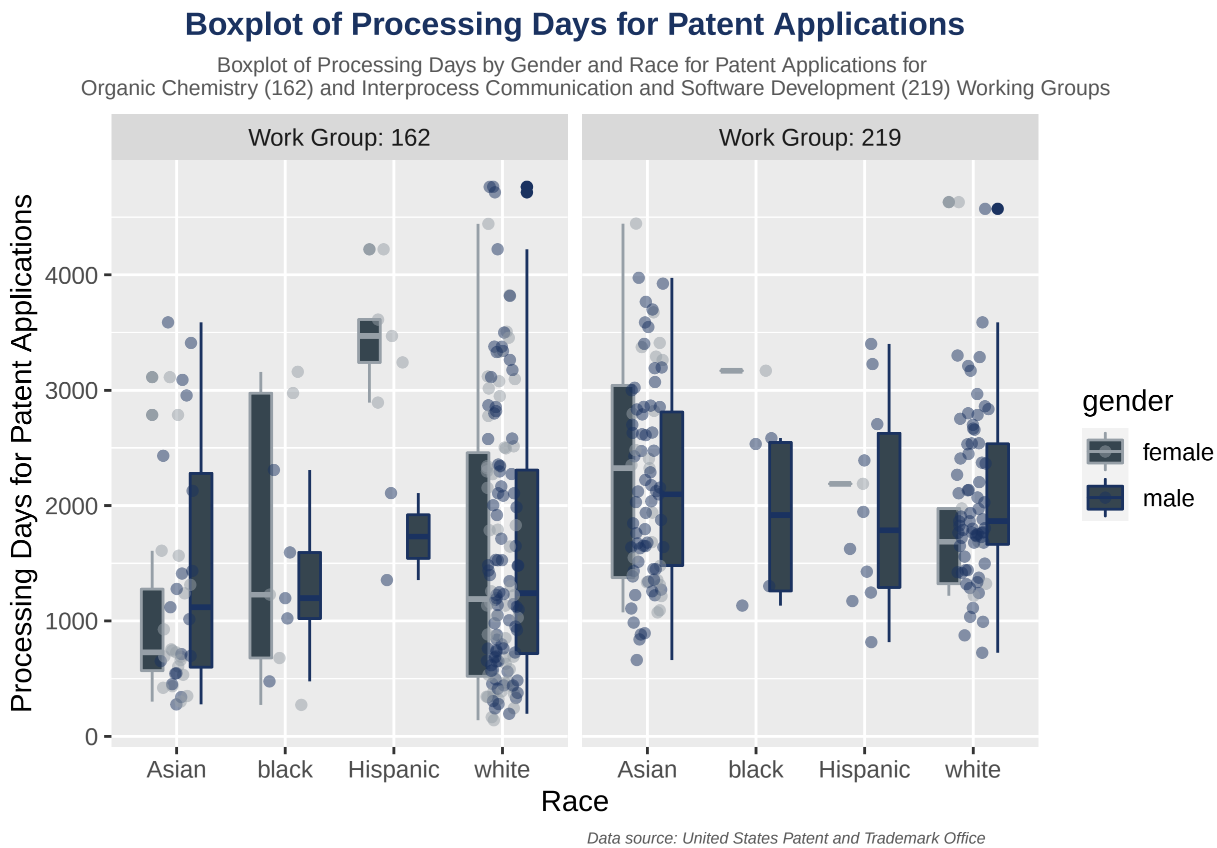


Processing Days Overview - Boxplot

```
# define function to return label for facet_wrap WG titles
label_facet <- function(original_var, custom_name){
  lev <- levels(as.factor(original_var))
  lab <- paste0(custom_name, ": ", lev)
  names(lab) <- lev
  return(lab)}

# box plot
combined %>%
  ggplot(aes(race, appl_proc_days, color = gender)) +
  geom_boxplot(width = .4, fill = "#36454F", position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("#36454F","#969fa7")) +
  scale_color_manual(values = c("#969fa7","#1a3260")) +
  geom_jitter(width=0.15, alpha=0.5) +
  labs(title = "Boxplot of Processing Days for Patent Applications",
       subtitle= "Boxplot of Processing Days by Gender and Race for Patent Applications for
       Organic Chemistry (162) and Interprocess Communication and Software Development (219) Working Gro
       caption = "Data source: United States Patent and Trademark Office") +
  ylab("Processing Days for Patent Applications") +
  xlab("Race") +
  # adjust title + subtitle formatting
  theme(plot.title = element_text(color = "#1a3260", size = 12, face = "bold", hjust = 0.5),
```

```
        plot.subtitle = element_text(color = "#585858", size = 8, hjust =0.5),
        plot.caption = element_text(color = "#585858", size = 6, face = "italic", hjust =0.9)) +
  facet_wrap( ~ wg, labeller = labeller(wg = label_facet(combined$wg, "Work Group")))
```



Advice Networks

```
# first get work group for each examiner and limit to our two wgs of interest
examiner_aus = distinct(subset(applications_full, select=c(examiner_art_unit, examiner_id)))
# we eventually want to make a network with nodes colored by work group, so lets add that indicator
examiner_aus$wg = substr(examiner_aus$examiner_art_unit, 1,3)
# restrict down to our selected art units to reduce merging complexity later on
# examiner_aus = examiner_aus[examiner_aus$wg==163 | examiner_aus$wg==176,]

# now we will merge in the aus df on applications
adviceNet = merge(x=edges, y=examiner_aus, by.x="ego_examiner_id", by.y="examiner_id", all.x=TRUE)
adviceNet = adviceNet %>% rename(ego_art_unit=examiner_art_unit, ego_wg=wg)
```

```
# drop edges which are missing ego or alter id
adviceNet = drop_na(adviceNet)

# now repeat for the alter examiners
adviceNet = merge(x=adviceNet, y=examiner_aus, by.x="alter_examiner_id", by.y="examiner_id", all.x=TRUE)
adviceNet = adviceNet %>% rename(alter_art_unit=examiner_art_unit, alter_wg=wg)
adviceNet = drop_na(adviceNet)

egoNodes = subset(adviceNet, select=c(ego_examiner_id,ego_art_unit, ego_wg)) %>%   rename(examiner_id=eg
alterNodes = subset(adviceNet, select=c(alter_examiner_id,alter_art_unit, alter_wg))%>% rename(examiner_
nodes = rbind(egoNodes, alterNodes)
nodes = distinct(nodes) #5412 examiners(but some are repeated because they move amongst art units)

# when we reduce to the list of distinct vertices, we actually have more than we should, since some exa
nodes = nodes %>% group_by(examiner_id) %>% summarise(examiner_id=first(examiner_id), art_unit=first(ar
# we are left with just 2400 unique examiners
```

Construct network and calculate centralities

```
adviceNet = graph_from_data_frame(d=adviceNet, vertices=nodes, directed=TRUE)
# centralities
Degree <- degree(adviceNet, v=V(adviceNet))
Betweenness <- betweenness(adviceNet)
Eigenvector <- evcent(adviceNet)$vector

V(adviceNet)$size = Degree
V(adviceNet)$eig = round(Eigenvector,2)
V(adviceNet)$bet = round(Betweenness,2)
```

Model the relationship between centralities and app_proc_time

```
# first we'll need to merge the centrality measurements back into the imputed applications set
centralities <- cbind(Degree, Eigenvector, Betweenness)
centralities = round(centralities,2)
centralities = data.frame(centralities)
centralities <- cbind(examiner_id = rownames(centralities), centralities)
rownames(centralities) <- 1:nrow(centralities)

centralities %>% skim() # no missing values but very heavily skewed towards 0 for all centrality measur
```

Table 10: Data summary

Name	Piped data
Number of rows	2387
Number of columns	4

Column type frequency:

character	1
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
examiner_id	0	1	5	5	0	2387	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Degree	0	1	91.68	190.96	1	9	28	89	2844.00	
Eigenvector	0	1	0.00	0.03	0	0	0	0	1.00	
Betweenness	0	1	291.07	2549.03	0	0	0	0	62399.84	

```
# now merge on examiner_id
applications_final = merge(x=applications_full, y=centralities, by="examiner_id", all.x=TRUE)
applications_final %>% skim() # we will have quite a few NaNs popping back up for those examiners who d
```

Table 13: Data summary

Name	Piped data
Number of rows	1684935
Number of columns	18
Column type frequency:	
character	7
Date	1
factor	1
numeric	9
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1	8	8	0	1684935	0
examiner_name_last	0	1	2	17	0	3746	0
examiner_name_first	0	1	1	12	0	2548	0
uspc_class	0	1	3	3	0	412	0
uspc_subclass	0	1	6	6	0	6090	0
disposal_type	0	1	3	3	0	2	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
race	0	1	5	8	0	5	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
appl_end_date	0	1	2000-04-07	2050-06-30	2011-12-27	5003

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	mal: 1134112, fem: 550823

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	0	1.00	78650.65	13611.68	59012	66481	75149	93760	99990.00	
examiner_art_unit	0	1.00	1918.94	300.12	1600	1657	1771	2166	2498.00	
appl_status_code	0	1.00	164.39	30.75	18	150	150	161	854.00	
tc	0	1.00	1868.08	294.48	1600	1600	1700	2100	2400.00	
tenure_days	0	1.00	5638.21	986.17	216	5131	6185	6337	6518.00	
appl_proc_days	0	1.00	1192.41	619.59	0	768	1081	1482	17898.00	
Degree	656339	0.61	97.77	177.10	1	12	35	102	2844.00	
Eigenvector	656339	0.61	0.00	0.03	0	0	0	0	1.00	
Betweenness	656339	0.61	317.50	2560.50	0	0	0	1	62399.84	

```
# nothing to do there but remove the missing values
applications_final = drop_na(applications_final)

# clean
rm(examiner_aus)
rm(egoNodes)
rm(alterNodes)
rm(nodes)
rm(adviceNet)
gc()
```

```
##             used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   5007057 267.5   14721424  786.3  14721424  786.3
## Vcells 105816612 807.4  331961996 2532.7 414925006 3165.7
```

Modelling

```
# we wish to model the relationship between various centralities and appl_days
# we will make our first model as a simplistic model assuming no interactions among predictors
lm1 = lm(appl_proc_days~Degree+Eigenvector+Betweenness+tenure_days+gender+race, data=applications_final)
summary(lm1)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Degree + Eigenvector + Betweenness +
##     tenure_days + gender + race, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1383.2  -429.3  -113.5   294.2  4962.4
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.530e+03  5.521e+00 277.121  < 2e-16 ***
## Degree        7.024e-02  3.725e-03  18.857  < 2e-16 ***
## Eigenvector  -1.607e+02  2.397e+01  -6.701 2.07e-11 ***
## Betweenness   6.804e-03  2.493e-04  27.294  < 2e-16 ***
## tenure_days  -4.879e-02  9.011e-04 -54.145  < 2e-16 ***
## gendermale    1.520e+01  1.356e+00  11.210  < 2e-16 ***
## raceblack    -3.049e+01  3.154e+00  -9.667  < 2e-16 ***
## raceHispanic  1.615e+01  4.446e+00   3.633  0.00028 ***
## raceother     4.187e+01  3.508e+01   1.193  0.23273
## racewhite    -6.500e+01  1.352e+00 -48.072  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 628 on 1028586 degrees of freedom
## Multiple R-squared:  0.007128,   Adjusted R-squared:  0.007119
## F-statistic: 820.4 on 9 and 1028586 DF,  p-value: < 2.2e-16
```

Interpretations: - The “baseline” expectation for application processing time is 1500 - That would be for a female asian examiner who just started, 0 tenure days, and has never asked any advice

- Everytime an examiner asks advice to a new colleague examiner (increase degree by 1), we expect processing time to increase slightly (.07 days)
- Increasing an examiner’s importance as measured by eigenvector centrality is expected to decrease processing time by 160 days
- Increasing an examiner’s betweenness increases the processing time slightly (less than a day)
- It is important to note that the centrality measurements are all coupled, so in a vacuum these interpretations are valid, but in practice we could not increase an examiner’s degree without also altering in some way their eigenvector and betweenness centralities
- Longer tenured examiners process applications a bit faster with each additional day of tenure
- Male examiners are expected to take roughly 2 weeks longer than their female counterparts

- Black, Hispanic, and Other-raced examiners all take longer to process than asian
- White examiners process applications much faster than Asian, by about 60 days
- Important to note the out goodness of fit is very low, so these insights should be taken with a grain of salt

We can try to capture some of the more complex relationships among predictors by adding interactions

```
lm2 = lm(appl_proc_days~Degree+Eigenvector+Betweenness+tenure_days+gender+race+Degree*gender+Eigenvecto
summary(lm2)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Degree + Eigenvector + Betweenness +
##     tenure_days + gender + race + Degree * gender + Eigenvector *
##     gender + Betweenness * gender + Degree * race + Eigenvector *
##     race + Betweenness * race, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1415.8  -429.1  -113.4   294.3  4959.0
##
## Coefficients: (4 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.513e+03  5.597e+00 270.392  < 2e-16 ***
## Degree                    2.059e-01  8.498e-03  24.228  < 2e-16 ***
## Eigenvector              -4.546e+03  8.970e+02  -5.067 4.03e-07 ***
## Betweenness               1.869e-03  7.853e-04   2.380  0.01730 *
## tenure_days              -4.795e-02  9.026e-04 -53.128  < 2e-16 ***
## gendermale                1.855e+01  1.537e+00  12.073  < 2e-16 ***
## raceblack                -3.175e+01  3.660e+00  -8.673  < 2e-16 ***
## raceHispanic              5.594e+01  5.425e+00  10.310  < 2e-16 ***
## raceother                 4.206e+01  3.507e+01   1.199  0.23041
## racewhite                -5.045e+01  1.545e+00 -32.649  < 2e-16 ***
## Degree:gendermale        -4.924e-02  8.202e-03  -6.004 1.93e-09 ***
## Eigenvector:gendermale    1.917e+03  2.256e+02   8.494  < 2e-16 ***
## Betweenness:gendermale    4.859e-03  7.012e-04   6.929 4.24e-12 ***
## Degree:raceblack         -1.677e-02  2.622e-02  -0.640  0.52248
## Degree:raceHispanic      -4.504e-01  3.870e-02 -11.640  < 2e-16 ***
## Degree:raceother                 NA         NA      NA       NA
## Degree:racewhite         -1.527e-01  8.148e-03 -18.742  < 2e-16 ***
## Eigenvector:raceblack     3.056e+04  3.842e+03   7.954 1.80e-15 ***
## Eigenvector:raceHispanic         NA         NA      NA       NA
## Eigenvector:raceother            NA         NA      NA       NA
## Eigenvector:racewhite     2.593e+03  8.727e+02   2.971  0.00297 **
## Betweenness:raceblack     1.362e-02  2.594e-03   5.249 1.53e-07 ***
## Betweenness:raceHispanic -4.150e-02  9.257e-03  -4.483 7.37e-06 ***
## Betweenness:raceother            NA         NA      NA       NA
## Betweenness:racewhite     1.380e-03  5.362e-04   2.573  0.01008 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 627.8 on 1028575 degrees of freedom
```

```
## Multiple R-squared:  0.007845,   Adjusted R-squared:  0.007826
## F-statistic: 406.7 on 20 and 1028575 DF,  p-value: < 2.2e-16
```

- The baseline expectation is roughly the same as it was before, around 1500 days
- Increasing degree or betweenness centrality (in a vaccuum) is expected to increase processing time, while increasing eigenvector centrality decreases processing time quite significantly (4500 days)
- This model expects Black examiners to process faster than asian examiners, and every other race to be slower
- From interaction terms, we also know that increasing degree for male examiners decreases processing time significantly
- Male examiners with higher betweenness centrality have roughly the same expected processing time (0.005 days added)

Disclaimer: While these models are providing theoretically meaningful insights, we should note that the proportion of variance in the data explained by both of these models is around 1%, ie they are not particularly good models as far as goodness-of-fit is concerned.

Workgroup-specific analysis

After completing the general USPTO analysis, we have chosen to zoom in on two tech units: 1600 and 2100. We wanted to look at the STEM field and specifically the differences between life-science related patents (1600: Biotech and Organic Fields) and compute-science related patents (2100: Computer Architecture and Information Security)

We will use workgroups 162 and 219 as the representative work groups for these two tech units, and randomly sample from the larger workgroup to get two approximately evenly sized workgroup data sets.

```
# first get work group for each examiner and limit to our two wgs of interest
examiner_aus = distinct(subset(applications_full, select=c(examiner_art_unit, examiner_id,gender)), exam

# note we want distinct examiners, not just distinct art_unit+examiner combos, since examiners can move

# we eventually want to make a network with nodes colored by work group, so lets add that indicator
examiner_aus$wg = substr(examiner_aus$examiner_art_unit, 1,3)
# restrict down to our selected art units to reduce merging complexity later on
examiner_aus = examiner_aus[examiner_aus$wg==162 | examiner_aus$wg==219,]

# now we will merge in the aus df on applications
adviceNet = merge(x=edges, y=examiner_aus, by.x="ego_examiner_id", by.y="examiner_id", all.x=TRUE)
adviceNet = adviceNet %>% rename(ego_art_unit=examiner_art_unit, ego_wg=wg, ego_gender=gender)

# drop edges which are missing ego or alter id
adviceNet = drop_na(adviceNet)

# now repeat for the alter examiners
adviceNet = merge(x=adviceNet, y=examiner_aus, by.x="alter_examiner_id", by.y="examiner_id", all.x=TRUE
adviceNet = adviceNet %>% rename(alter_art_unit=examiner_art_unit, alter_wg=wg, alter_gender=gender)
adviceNet = drop_na(adviceNet)

egoNodes = subset(adviceNet, select=c(ego_examiner_id,ego_art_unit, ego_wg, ego_gender)) %>%   rename(e
```

```
alterNodes = subset(adviceNet, select=c(alter_examiner_id,alter_art_unit, alter_wg, alter_gender))%>% re
nodes = rbind(egoNodes, alterNodes)
nodes = distinct(nodes)

# note we have fewer examiners than we started with due to some examiners never asking each other for a


# when we reduce to the list of distinct vertices, we actually have more than we should, since some exam
#nodes = nodes %>% group_by(examiner_id) %>% summarise(examiner_id=first(examiner_id), art_unit=first(a
```

Repeat centralities analysis

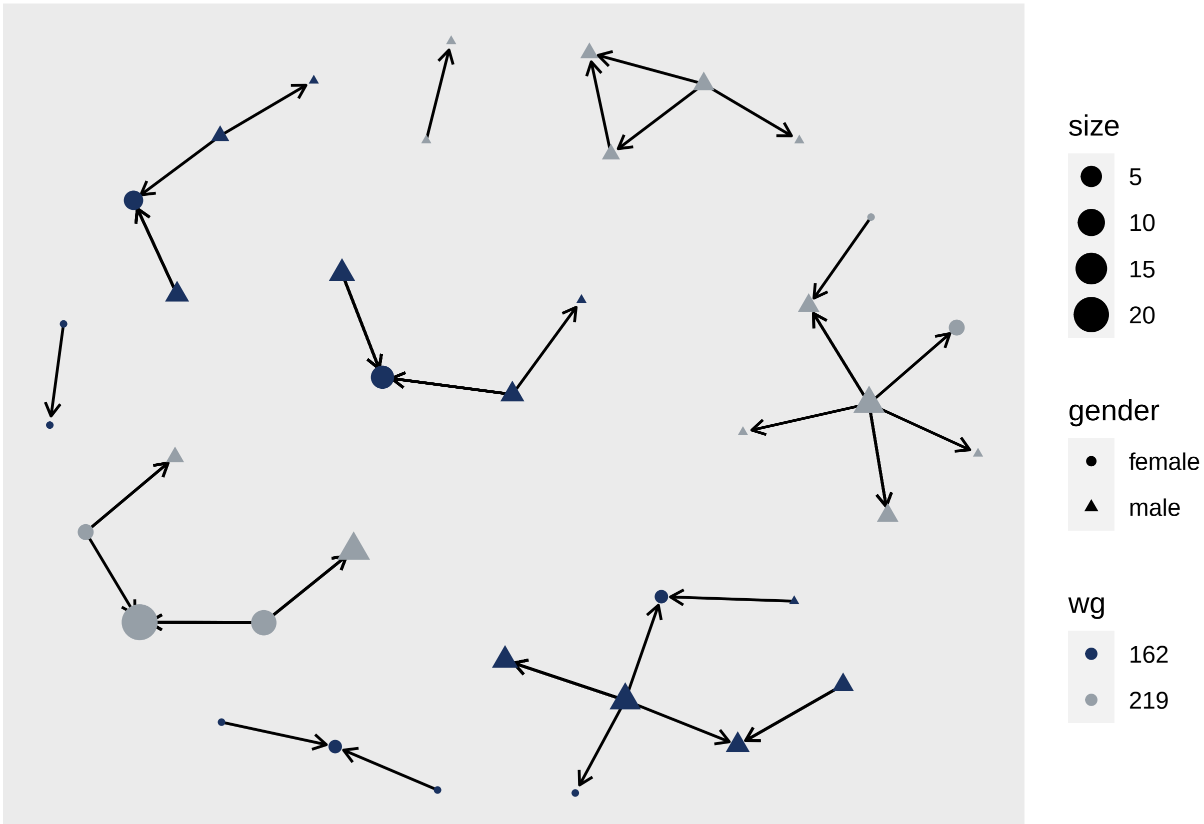
Construct network and calculate centralities

```
adviceNet = graph_from_data_frame(d=adviceNet, vertices=nodes, directed=TRUE)
# centralities
Degree <- degree(adviceNet, v=V(adviceNet))
Betweenness <- betweenness(adviceNet)
Eigenvector <- evcent(adviceNet)$vector

V(adviceNet)$size = Degree
V(adviceNet)$eig = round(Eigenvector,2)
V(adviceNet)$bet = round(Betweenness,2)
V(adviceNet)$wg = nodes$wg
V(adviceNet)$gender = as.character(nodes$gender)
```

Visualize

```
ggraph(adviceNet, layout="kk") +
  geom_edge_link(arrow=arrow(length=unit(2,'mm')), end_cap=circle(1.2,'mm'))+
  geom_node_point(aes(size=size, color=wg, shape=gender), show.legend=T) +
  scale_color_manual(values=c('#1a3260', '#969fa7'))
```



We have a much sparser network here with many components instead of one or two large components. This is likely due to the restrictive size of our analysis, however it is still interesting to note the existence of these cliques, especially given that for some examiners we have 15-20 instances of advice asking. This shows a clear preference amongst the examiners in both 162 and 219 to stick with their local friend group when resolving issues.

```
unique <- applications_final[!duplicated(applications_final[, c('examiner_id')]), ]
unique$wg = substr(unique$examiner_art_unit,1,3)
summary_df <- applications_final %>% group_by(examiner_id) %>% summarise(Applications = length(applicati

# ggplot(unique, aes(x=Degree, y=tenure_days)) +
#   geom_point(aes(color=as.factor(wg)), show.legend=T) +
#   scale_color_manual(values=c('#1a3260', '#969fa7'))
#
# ggplot(summary_df, aes(x=Degree, y=Avg_Proc_Time)) +
#   geom_point() +
#   scale_color_manual(values=c('#1a3260'))
```

Model the relationship between centralities and app_proc_time

```
# first we'll need to merge the centrality measurements back into the imputed applications set
centralities <- cbind(Degree, Eigenvector, Betweenness)
centralities = round(centralities,2)
centralities = data.frame(centralities)
centralities <- cbind(examiner_id = rownames(centralities), centralities)
```

```
rownames(centralities) <- 1:nrow(centralities)

centralities = merge(centralities, subset(examiner_aus, select=-c(wg,gender)), by="examiner_id") # need

# now merge on examiner_id
applications_final = merge(x=applications_full, y=centralities, by=c("examiner_id","examiner_art_unit")
applications_final %>% skim() # we will have quite a few NaNs popping back up for those examiners who d
```

Table 18: Data summary

Name	Piped data
Number of rows	19502
Number of columns	18
Column type frequency:	
character	7
Date	1
factor	1
numeric	9
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
application_number	0	1	8	8	0	19502	0
examiner_name_last	0	1	2	16	0	38	0
examiner_name_first	0	1	3	11	0	35	0
uspc_class	0	1	3	3	0	86	0
uspc_subclass	0	1	6	6	0	1257	0
disposal_type	0	1	3	3	0	2	0
race	0	1	5	8	0	3	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
appl_end_date	0	1	2000-08-22	2017-06-20	2010-08-03	2928

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	fem: 12205, mal: 7297

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
examiner_id	0	1	70237.43	8351.03	59491	63822	67753	75034	98717	
examiner_art_unit	0	1	1744.65	231.70	1621	1624	1626	1627	2199	
appl_status_code	0	1	172.65	38.43	30	150	161	161	454	
tc	0	1	1705.45	203.98	1600	1600	1600	1600	2100	
tenure_days	0	1	6027.99	520.54	1526	5872	6311	6345	6346	
appl_proc_days	0	1	1100.76	611.09	96	651	985	1428	4981	
Degree	0	1	2.68	3.02	1	1	1	3	24	
Eigenvector	0	1	0.02	0.11	0	0	0	0	1	
Betweenness	0	1	0.00	0.00	0	0	0	0	0	

```r
# nothing to do there but remove the missing values
applications_final = drop_na(applications_final)
```

Modelling

```r
applications_final$wg = substr(applications_final$examiner_art_unit,1,3)
applications_final$wg = as.factor(applications_final$wg)

# rename gender var to fix a knitting error


#unique(applications_final$race) # Just Asian, White, or Hispanic examiners present in this dataset



# for our first model we will once again cover no interactions and just look at base variables
# also, we have dropped betweenness because it is 0 for all examiners, probably due to the lack of conn
lm1 = lm(appl_proc_days~Degree+Eigenvector+tenure_days+race+gender+wg, data=applications_final)
summary(lm1)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Degree + Eigenvector + tenure_days +
##     race + gender + wg, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1512.5  -366.7   -65.6   279.8  3922.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.036e+03  5.537e+01  18.720  < 2e-16 ***
## Degree        1.915e+01  2.386e+00   8.027 1.06e-15 ***
## Eigenvector  -5.872e+02  6.640e+01  -8.843  < 2e-16 ***
## tenure_days  -1.740e-02  8.703e-03  -1.999 0.045604 *
## raceHispanic -1.067e+02  3.161e+01  -3.374 0.000743 ***
## racewhite    -2.213e+01  9.498e+00  -2.330 0.019808 *
## gendermale    9.912e+00  8.677e+00   1.142 0.253371
## wg219         6.662e+02  1.089e+01  61.191  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 547.3 on 19494 degrees of freedom
## Multiple R-squared:  0.1982, Adjusted R-squared:  0.1979
## F-statistic: 688.3 on 7 and 19494 DF,  p-value: < 2.2e-16
```

Our work-group specific analysis gives much different results from before

First, our baseline estimate (Female, Asian, 0 tenure days and no prior connections) for application time is 1038 days

In addition, we assume a further increase in processing time for each advice-sought by about 20 days

One notable insight is that examiners from work group 219 are expected to take significantly longer in processing applications than for those in workgroup 162. This could potentially be due to the larger size of workgroup 162, allowing for lower on-average workload. It is also possible the discrepency is due to a simple difference in the nature/complexity of Biotech vs CS -oriented patents.

We also expect Hispanic and White examiners to complete applications faster than Asian examiners.

Based on this simplistic model, we would naively conclude that the USPTO should focus on hiring more Hispanic and White female examiners, as we expect them to process applications much faster than all male examiners, and especially faster than male asian examiners.

Of course, we know this model is missing the whole picture and we ought to increase its complexity before making conclusions…

```r
# add interactions
lm2 = lm(appl_proc_days~Degree+Eigenvector+tenure_days+gender+race+wg
         +gender*Degree+race*Degree+gender*Eigenvector+race*Eigenvector, data=applications_final)
summary(lm2)
```

```
##
## Call:
## lm(formula = appl_proc_days ~ Degree + Eigenvector + tenure_days +
##     gender + race + wg + gender * Degree + race * Degree + gender *
##     Eigenvector + race * Eigenvector, data = applications_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1475.0  -366.0   -60.7   279.2  3762.5
##
## Coefficients: (2 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.385e+03  5.837e+01  23.734  < 2e-16 ***
## Degree                   -1.045e+02  8.895e+00 -11.746  < 2e-16 ***
## Eigenvector               2.083e+03  1.977e+02  10.538  < 2e-16 ***
## tenure_days              -5.327e-02  8.842e-03  -6.024 1.73e-09 ***
## gendermale                1.134e+02  1.330e+01   8.529  < 2e-16 ***
## raceHispanic             -2.552e+01  3.172e+01  -0.805 0.420970
## racewhite                -2.251e+02  1.581e+01 -14.237  < 2e-16 ***
## wg219                     7.495e+02  1.167e+01  64.199  < 2e-16 ***
## Degree:gendermale        -5.437e+01  4.604e+00 -11.810  < 2e-16 ***
## Degree:raceHispanic              NA         NA      NA       NA
## Degree:racewhite          1.536e+02  9.188e+00  16.718  < 2e-16 ***
```

```
## Eigenvector:gendermale    7.762e+03  2.212e+03   3.509 0.000451 ***
## Eigenvector:raceHispanic         NA         NA      NA       NA
## Eigenvector:racewhite     4.512e+03  5.426e+04   0.083 0.933735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 541.6 on 19490 degrees of freedom
## Multiple R-squared:  0.215,  Adjusted R-squared:  0.2145
## F-statistic: 485.3 on 11 and 19490 DF,  p-value: < 2.2e-16
```

```
# several interactions are omitted due to insufficient data/not statistically significant results:
# gender*race, all combinations are not statistically significant, probably due to having only 38 uniqu
# tenure*gender
# tenure*race
stargazer(lm1, lm2,
          type="latex",
          dep.var.labels = "Application Processing Time",
          covariate.labels= c("Degree Centrality", "Eigenvector Centrality", "Tenure (days)", "Male", "I
          digits = 2,
          font.size="LARGE")
```

```
##
## % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harva
## % Date and time: Sun, Jun 05, 2022 - 12:38:43 PM
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \LARGE
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{2}{c}{\textit{Dependent variable:}} \\
## \cline{2-3}
## \\[-1.8ex] & \multicolumn{2}{c}{Application Processing Time} \\
## \\[-1.8ex] & (1) & (2)\\
## \hline \\[-1.8ex]
##  Degree Centrality & 19.15$^{***}$ & $-$104.49$^{***}$ \\
##   & (2.39) & (8.90) \\
##   & & \\
##  Eigenvector Centrality & $-$587.16$^{***}$ & 2,082.95$^{***}$ \\
##   & (66.40) & (197.66) \\
##   & & \\
##  Tenure (days) & $-$0.02$^{**}$ & $-$0.05$^{***}$ \\
##   & (0.01) & (0.01) \\
##   & & \\
##  Male & $-$106.65$^{***}$ & $-$25.52 \\
##   & (31.61) & (31.72) \\
##   & & \\
##  Hispanic & $-$22.13$^{**}$ & $-$225.12$^{***}$ \\
##   & (9.50) & (15.81) \\
##   & & \\
##  White & 9.91 & 113.41$^{***}$ \\
##   & (8.68) & (13.30) \\
##   & & \\
```

```
##  Work Group 219 & 666.22$^{***}$ & 749.51$^{***}$ \\
##   & (10.89) & (11.67) \\
##   & & \\
##  Degree:Male &  & $-$54.37$^{***}$ \\
##   &  & (4.60) \\
##   & & \\
##  Degree:Hispanic &  &  \\
##   &  &  \\
##   & & \\
##  Degree:White &  & 153.60$^{***}$ \\
##   &  & (9.19) \\
##   & & \\
##  Eigenvector:Male &  & 7,761.68$^{***}$ \\
##   &  & (2,211.85) \\
##   & & \\
##  Eigenvector:Hispanic &  &  \\
##   &  &  \\
##   & & \\
##  Eigenvector:White &  & 4,511.51 \\
##   &  & (54,259.36) \\
##   & & \\
##  Constant & 1,036.48$^{***}$ & 1,385.41$^{***}$ \\
##   & (55.37) & (58.37) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 19,502 & 19,502 \\
## R$^{2}$ & 0.20 & 0.21 \\
## Adjusted R$^{2}$ & 0.20 & 0.21 \\
## Residual Std. Error & 547.30 (df = 19494) & 541.59 (df = 19490) \\
## F Statistic & 688.29$^{***}$ (df = 7; 19494) & 485.25$^{***}$ (df = 11; 19490) \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

Our baseline estimate is higher, around 1400 days.

Examiners are now expected to take less processing time with each additional advice-seeking, by 106 days. Eigenvector centrality has a negative (longer) impact on processing time by 2117 days.(?) With each additional day of tenure, female examiners shave about 0.05 days off their expected processing time. Male examiners are expected to take about 110 days longer than their female counterparts. Both hispanic and White examiners are expected to process faster than their asian colleagues. As before, examiners from workgroup 219 appear to require much longer to process applications.

Among interaction terms, we expect male examiners to remove about 53 days of processing time when seeking advice (inc degree by 1) compared to women. - This would seem to imply that "importance" as measured by degree centrality is more meaningful for male examiners than it is for female examiners

Lets investigate that insight with some predictions:

```
baseline = predict(lm2, data.frame(Degree=0,Eigenvector=0,tenure_days=0,gender='female',race='Asian',wg
lowDegMale = predict(lm2, data.frame(Degree=0,Eigenvector=0,tenure_days=0,gender='male',race='Asian',wg
lowDegFemale = baseline
highDegMale = predict(lm2, data.frame(Degree=5,Eigenvector=0,tenure_days=0,gender='male',race='Asian',w
```

```
highDegFemale = predict(lm2, data.frame(Degree=5,Eigenvector=0,tenure_days=0,gender='female', race='Asia

data.frame(baseline=baseline, unimportant_male=lowDegMale, important_male=highDegMale, unimportant_femal
```

```
##   baseline unimportant_male important_male unimportant_female important_female
## 1 1385.409         1498.822       704.5318           1385.409          862.975
```

This affirms what we saw when examining the model summary: men seem to gain more benefit (in terms of reducing processing time) from advice seeking than women. We can't deduce why that is from this model, but conjecture might say that since this is a male-dominated organization, men seem to benefit (at least in terms of reducing processing time) from advice-seeking more than women.

We can additionally see from the model summary that