exercise_3_tashfeen_ahmed

2024-04-02

```
# Install and load the arrow package
#install.packages("arrow")
#install_genderdata_package()
#install.packages("gender")
#install.packages("devtools")
#devtools::install_github("ropensci/genderdata", type = "source")

#install.packages("path/to/wru_package_directory", repos = NULL, type = "source")
#install.packages("ethnicolr")
#install.packages("wru")

library(gender)
```

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

```
library(arrow)
```

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

##
## Attaching package: 'arrow'

## The following object is masked from 'package:utils':
##
##     timestamp
```

```
library(dplyr)
```

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

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(tidyr)
```

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

```
library(wru)
```

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

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

```
library(lubridate)
```

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

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:arrow':
##
##     duration
```

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

```
library(igraph)
```

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

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

```
## The following objects are masked from 'package:lubridate':
##
##     %--%, union
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing
```

```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```
# Read Parquet file
parquet_file <- "D:\\Google Drive\\McGill\\Winter Semester\\W2\\Talent-Analytics-Assignments\\Part 2\\E
applications  <- read_parquet(parquet_file)

# Read CSV file
edge_link <- "D:\\Google Drive\\McGill\\Winter Semester\\W2\\Talent-Analytics-Assignments\\Part 2\\Exer
edges <- read.csv(edge_link)
```

```
# Guess the 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 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  4452118 237.8    8120585 433.7  4473850 239.0
## Vcells 49399728 376.9   92868664 708.6 79715944 608.2
```

```r
# Guess the examiner's race

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

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

```
## Predicting race for 2020

## Warning: Unknown or uninitialised column: 'state'.

## Proceeding with last name predictions...

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

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

```r
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
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  4559424 243.5    8120585 433.7  6083187 324.9
## Vcells 51616280 393.9   92868664 708.6 90877008 693.4
```

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

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

examiner_dates <- examiner_dates %>%
```

```
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))

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)

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

rm(examiner_dates)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  4566206 243.9   14731266  786.8  14731266  786.8
## Vcells 63978727 488.2  133906875 1021.7 133594320 1019.3
```

```
# List all columns in the data frame
str(applications)
```

```
## tibble [2,018,477 x 21] (S3: tbl_df/tbl/data.frame)
##  $ application_number  : chr [1:2018477] "08284457" "08413193" "08531853" "08637752" ...
##  $ filing_date         : Date[1:2018477], format: "2000-01-26" "2000-10-11" ...
##  $ examiner_name_last  : chr [1:2018477] "HOWARD" "YILDIRIM" "HAMILTON" "MOSHER" ...
##  $ examiner_name_first : chr [1:2018477] "JACQUELINE" "BEKIR" "CYNTHIA" "MARY" ...
##  $ examiner_name_middle: chr [1:2018477] "V" "L" NA NA ...
##  $ examiner_id         : num [1:2018477] 96082 87678 63213 73788 77294 ...
##  $ examiner_art_unit   : num [1:2018477] 1764 1764 1752 1648 1762 ...
##  $ uspc_class          : chr [1:2018477] "508" "208" "430" "530" ...
##  $ uspc_subclass       : chr [1:2018477] "273000" "179000" "271100" "388300" ...
##  $ patent_number       : chr [1:2018477] "6521570" "6440298" "5607816" "6927281" ...
##  $ patent_issue_date   : Date[1:2018477], format: "2003-02-18" "2002-08-27" ...
##  $ abandon_date        : Date[1:2018477], format: NA NA ...
##  $ disposal_type       : chr [1:2018477] "ISS" "ISS" "ISS" "ISS" ...
##  $ appl_status_code    : num [1:2018477] 150 250 250 250 161 150 135 161 161 250 ...
##  $ appl_status_date    : chr [1:2018477] "30jan2003 00:00:00" "27sep2010 00:00:00" "30mar2009 00:00:0
##  $ tc                  : num [1:2018477] 1700 1700 1700 1600 1700 1700 1600 1600 1600 1700 ...
##  $ gender              : chr [1:2018477] "female" NA "female" "female" ...
##  $ race                : chr [1:2018477] "white" "white" "white" "white" ...
##  $ earliest_date       : Date[1:2018477], format: "2000-01-10" "2000-01-04" ...
##  $ latest_date         : Date[1:2018477], format: "2016-04-01" "2016-09-09" ...
##  $ tenure_days         : num [1:2018477] 5926 6093 6344 6331 6332 ...
```

```
str(edges)
```

```
## 'data.frame':    32906 obs. of  4 variables:
##  $ application_number: int  9402488 9402488 9402488 9445135 9445135 9445135 9479304 9479304 9479304 9
##  $ advice_date       : chr  "2008-11-17" "2008-11-17" "2008-11-17" "2008-08-21" ...
```

```
##  $ ego_examiner_id   : int  84356 84356 84356 92953 92953 92953 61767 61767 61767 61767 ...
##  $ alter_examiner_id : int  66266 63519 98531 71313 93865 91818 69277 92446 66805 70919 ...
```

160 and 175 are first 3 digits of workgroups choosen

Step 1: Filter the workgroups

```
applications$examiner_id <- as.character(applications$examiner_id)
edges$ego_examiner_id <- as.character(edges$ego_examiner_id)
edges$alter_examiner_id <- as.character(edges$alter_examiner_id)
edges$advice_date <- as.Date(edges$advice_date, format = "%Y-%m-%d")
applications$filing_date <- as.Date(applications$filing_date, format = "%Y-%m-%d")

# Select workgroups starting with 160 and 175
workgroups <- applications %>%
  filter(substr(examiner_art_unit, 1, 3) %in% c("160", "175"))

# Separate the dataframe into two, one for each workgroup
workgroup_160 <- workgroups %>% filter(substr(examiner_art_unit, 1, 3) == "160")
workgroup_175 <- workgroups %>% filter(substr(examiner_art_unit, 1, 3) == "175")
```

```
str(workgroup_160)
```

```
## tibble [155 x 21] (S3: tbl_df/tbl/data.frame)
##  $ application_number  : chr [1:155] "09491146" "09566266" "09570022" "09577601" ...
##  $ filing_date         : Date[1:155], format: "2000-01-25" "2000-05-05" ...
##  $ examiner_name_last  : chr [1:155] "LUCAS" "LUCAS" "LUCAS" "LUCAS" ...
##  $ examiner_name_first : chr [1:155] "ZACHARIAH" "ZACHARIAH" "ZACHARIAH" "ZACHARIAH" ...
##  $ examiner_name_middle: chr [1:155] NA NA NA NA ...
##  $ examiner_id         : chr [1:155] "75380" "75380" "75380" "75380" ...
##  $ examiner_art_unit   : num [1:155] 1600 1600 1600 1600 1600 1600 1600 1600 1600 1600 ...
##  $ uspc_class          : chr [1:155] "536" "424" "514" "435" ...
##  $ uspc_subclass       : chr [1:155] "023720" "189100" "015000" "320100" ...
##  $ patent_number       : chr [1:155] "6960659" "6855318" "6573244" "7015033" ...
##  $ patent_issue_date   : Date[1:155], format: "2005-11-01" "2005-02-15" ...
##  $ abandon_date        : Date[1:155], format: NA NA ...
##  $ disposal_type       : chr [1:155] "ISS" "ISS" "ISS" "ISS" ...
##  $ appl_status_code    : num [1:155] 250 250 250 250 250 250 250 250 250 250 ...
##  $ appl_status_date    : chr [1:155] "30nov2009 00:00:00" "20mar2013 00:00:00" "04jul2011 00:00:00"
##  $ tc                  : num [1:155] 1600 1600 1600 1600 1600 1600 1600 1600 1600 1600 ...
##  $ gender              : chr [1:155] "male" "male" "male" "male" ...
##  $ race                : chr [1:155] "white" "white" "white" "white" ...
##  $ earliest_date       : Date[1:155], format: "2000-01-14" "2000-01-14" ...
##  $ latest_date         : Date[1:155], format: "2017-05-12" "2017-05-12" ...
##  $ tenure_days         : num [1:155] 6328 6328 6328 6328 6328 ...
```

Step 2: Compare on examiners' demographics

```
# Summary statistics for workgroup 160
summary_160 <- workgroup_160 %>%
  group_by(gender, race) %>%
```

```
  summarize(count = n(),
            average_tenure_days = mean(tenure_days, na.rm = TRUE))
```

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

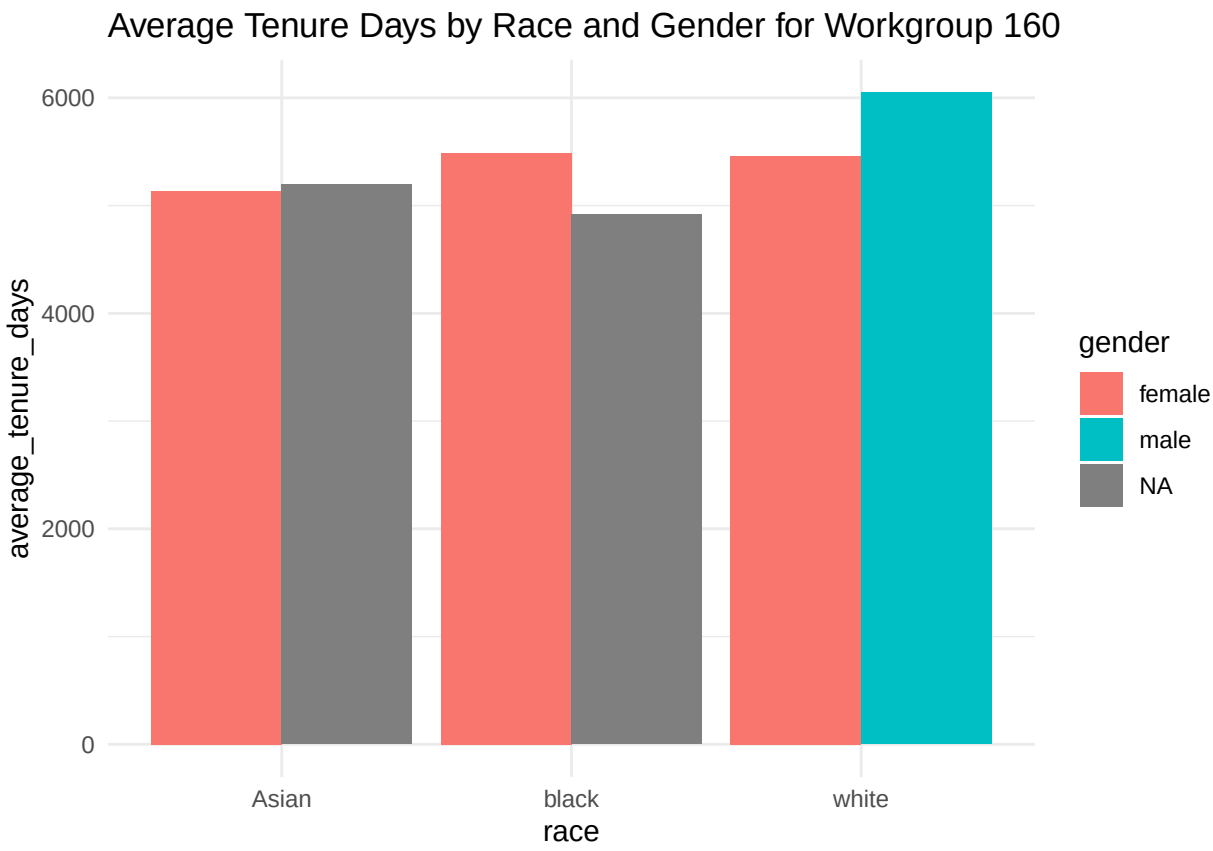
```
# Summary statistics for workgroup 175
summary_175 <- workgroup_175 %>%
  group_by(gender, race) %>%
  summarize(count = n(),
            average_tenure_days = mean(tenure_days, na.rm = TRUE))
```

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

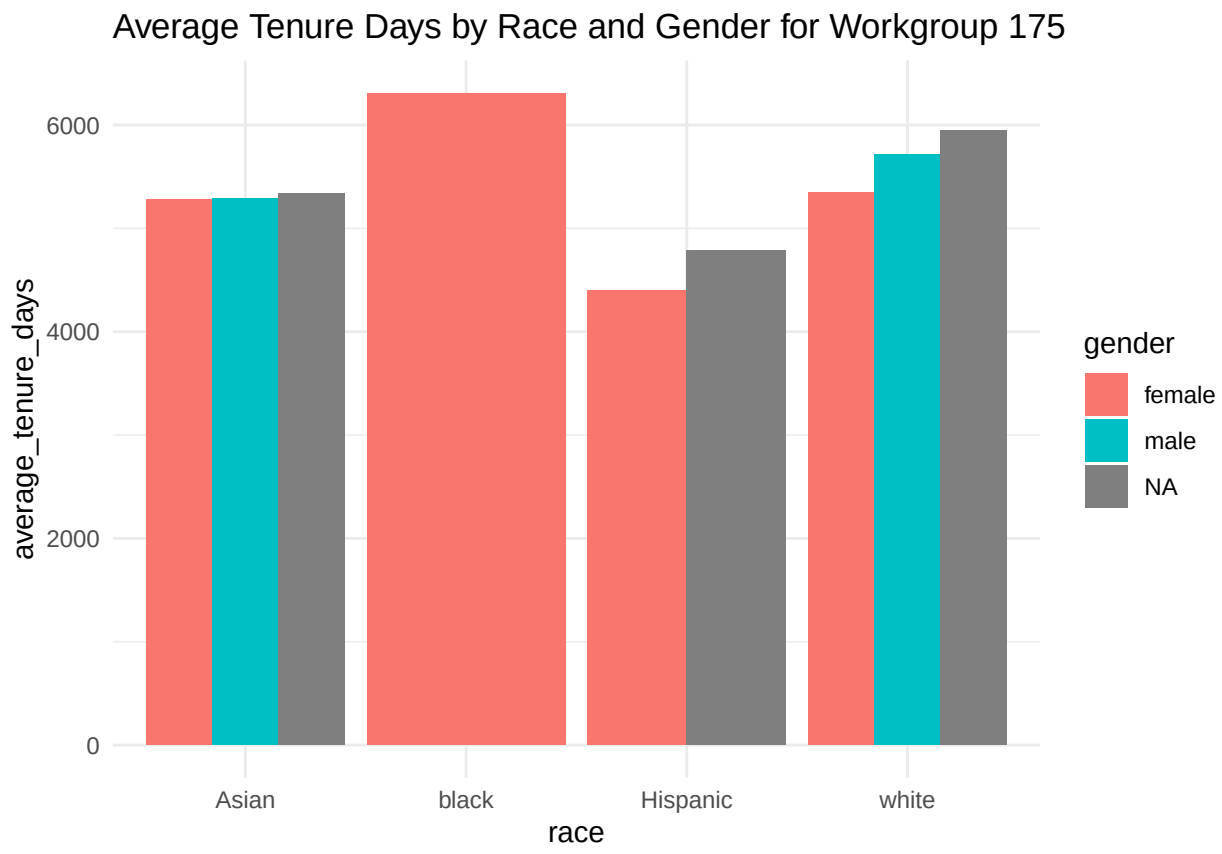
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

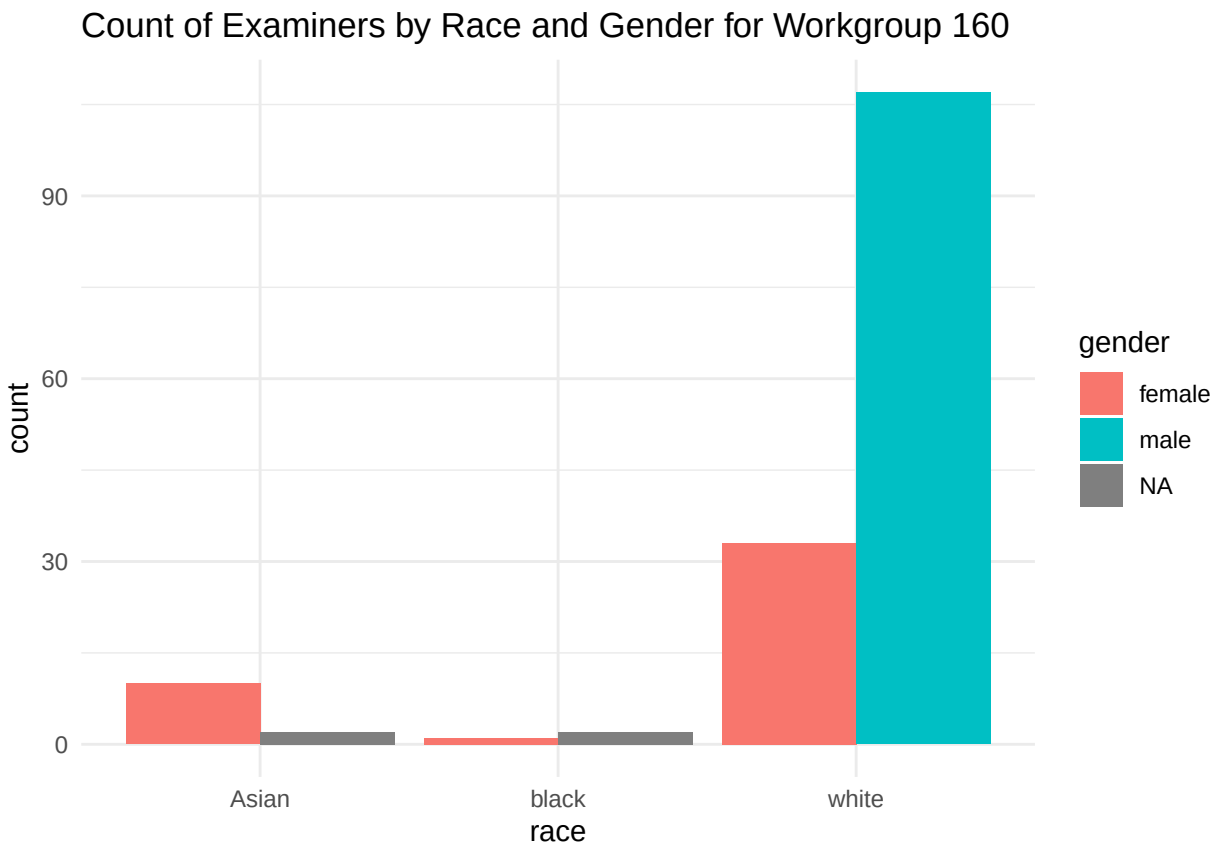
```
# Plot for workgroup 160
ggplot(summary_160, aes(x = race, y = average_tenure_days, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Average Tenure Days by Race and Gender for Workgroup 160")
```



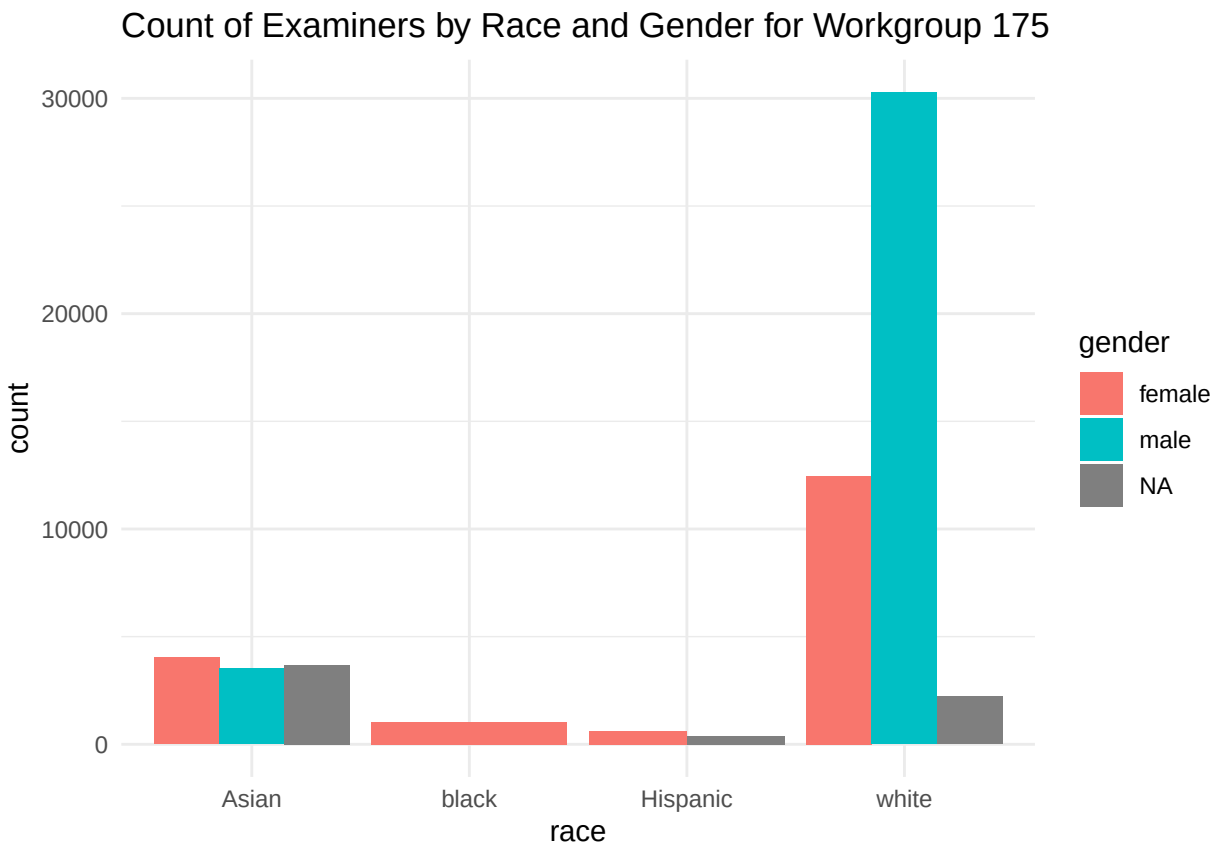
```
# Plot for workgroup 175
ggplot(summary_175, aes(x = race, y = average_tenure_days, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Average Tenure Days by Race and Gender for Workgroup 175")
```



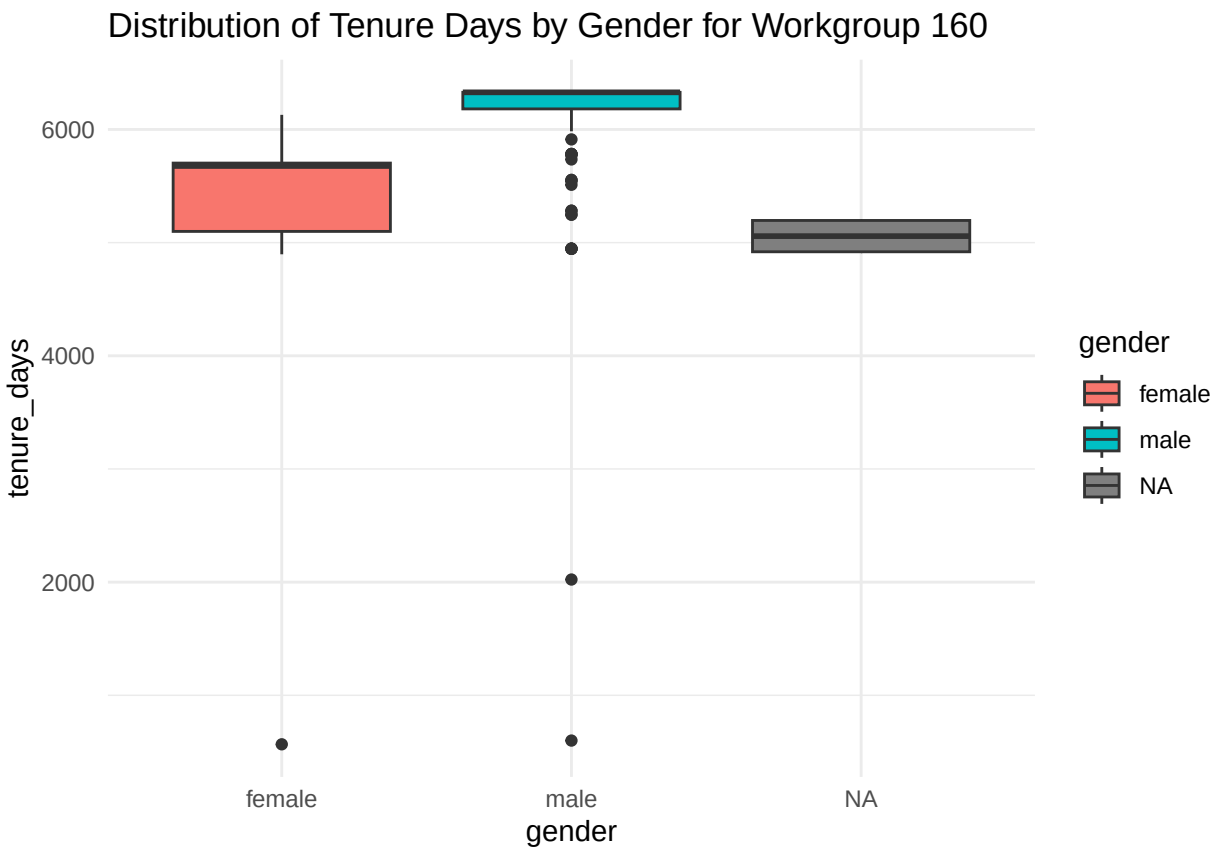

```
#Count of examiners by race and gender
# Plot the count of examiners by race and gender for workgroup 160
ggplot(summary_160, aes(x = race, y = count, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Count of Examiners by Race and Gender for Workgroup 160")
```




```
# Plot the count of examiners by race and gender for workgroup 175
ggplot(summary_175, aes(x = race, y = count, fill = gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  theme_minimal() +
  labs(title = "Count of Examiners by Race and Gender for Workgroup 175")
```

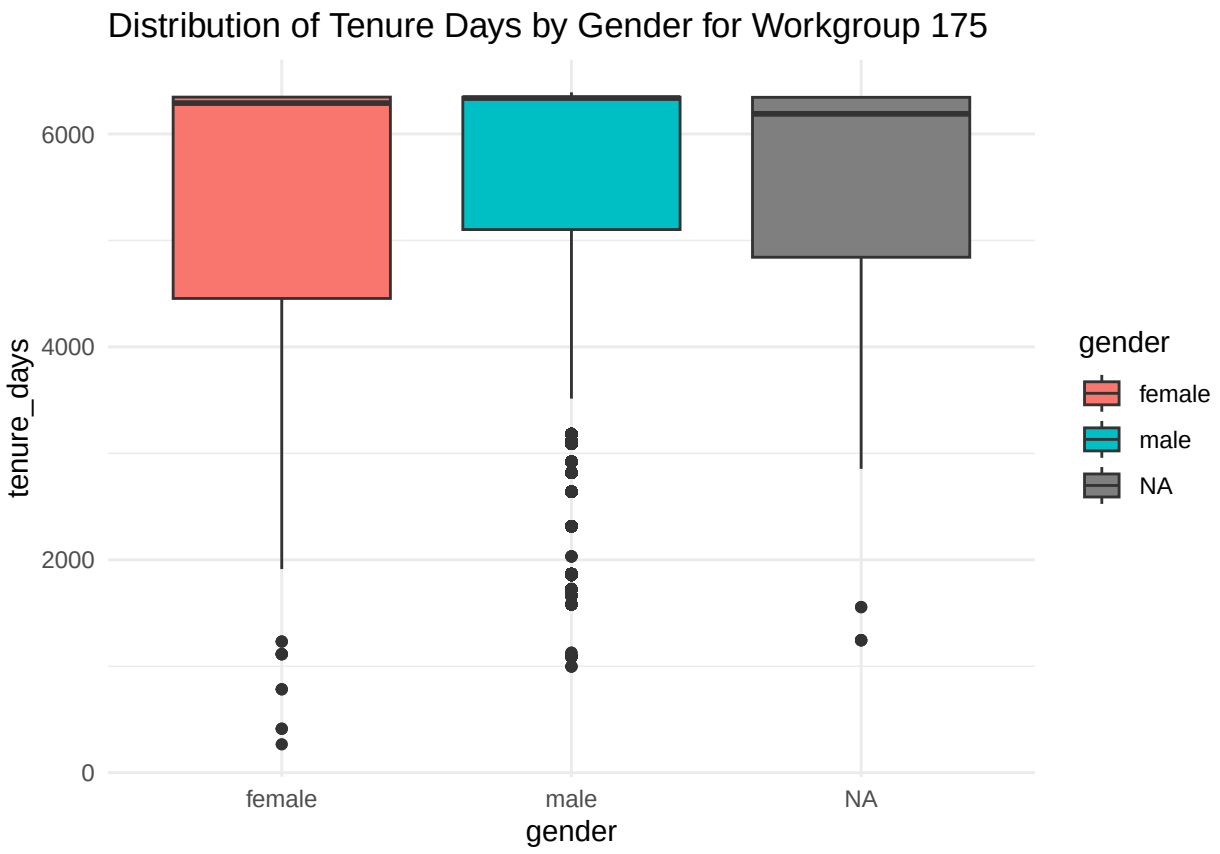



```
#Plot 2: Distribution of tenure days by gender
# Plot the distribution of tenure days by gender for workgroup 160
ggplot(workgroup_160, aes(x = gender, y = tenure_days, fill = gender)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Distribution of Tenure Days by Gender for Workgroup 160")
```




```
# Plot the distribution of tenure days by gender for workgroup 175
ggplot(workgroup_175, aes(x = gender, y = tenure_days, fill = gender)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Distribution of Tenure Days by Gender for Workgroup 175")
```

```
## Warning: Removed 891 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
#Summary Statistics Table:
# Add a workgroup identifier to each summary
summary_160 <- summary_160 %>% mutate(workgroup = "160")
summary_175 <- summary_175 %>% mutate(workgroup = "175")

# Combine the summaries into one table
combined_summary <- bind_rows(summary_160, summary_175)

# Print the combined summary table
head(combined_summary,5)
```

```
## # A tibble: 5 x 5
## # Groups:   gender [3]
##   gender race  count average_tenure_days workgroup
##   <chr>  <chr> <int>               <dbl> <chr>
## 1 female Asian    10               5134. 160
## 2 female black     1               5490  160
## 3 female white    33               5459. 160
## 4 male   white   107               6051. 160
## 5 <NA>   Asian     2               5196  160
```

```
str(edges)
```

```
## 'data.frame':    32906 obs. of  4 variables:
##  $ application_number: int  9402488 9402488 9402488 9445135 9445135 9445135 9479304 9479304 9479304 9
```

```
##  $ advice_date       : Date, format: "2008-11-17" "2008-11-17" ...
##  $ ego_examiner_id   : chr  "84356" "84356" "84356" "92953" ...
##  $ alter_examiner_id : chr  "66266" "63519" "98531" "71313" ...
```

Step 3: Create advice networks and calculate centrality

#“‘{r} library(igraph)

Ensure character data type consistency for IDs

edges$ego_examiner_id <- as.character(edges$ego_examiner_id) edges$alter_examiner_id <- as.character(edges$alter_examine
applications$examiner_id <- as.character(applications$examiner_id)

Convert ‘advice_date’ to Date format for potential future use

edges$advice_date <- as.Date(edges$advice_date, format = “%Y-%m-%d”)

Convert `examiner_art_unit` to character to extract the first 3 digits correctly

applications$examiner_art_unit <- as.character(applications$examiner_art_unit)

Filter applications for workgroups 160 and 175

wg_160_175_ids <- applications %>% filter(substr(examiner_art_unit, 1, 3) %in% c(“160”, “175”)) %>%
select(examiner_id) %>% distinct() %>% pull()

Ensure `ego_examiner_id` in edges is a character for accurate filtering

edges$ego_examiner_id <- as.character(edges$ego_examiner_id)

Filter edges based on the examiner IDs

edges_filtered <- edges %>% filter(ego_examiner_id %in% wg_160_175_ids)

Create network from filtered edges

network <- graph_from_data_frame(edges_filtered, directed = TRUE)

Get the IDs of examiners present in the network

network_examiner_ids <- unique(c(edges_filtered$ego_examiner_id, edges_filtered$alter_examiner_id))

Filter the degree centrality dataframe for examiners present in the network

degree_df <- degree_df %>% filter(examiner_id %in% network_examiner_ids)

#“‘

#“‘{r} filtered_row <- applications %>% filter(examiner_id == “9544253”)

View the filtered row

print(filtered_row) #“‘ I ran into a problem here that none of the examiner ids in edges dataframe that i found are being matched with the applications data

Logic being used to answer the 3 question: Filtering Applications: We filter the applications dataframe to include only those examiners belonging to workgroups 160 and 175 by selecting unique examiner_ids based on the first 3 digits of examiner_art_unit.

Filtering Edges: We filter the edges dataframe to include only those edges where the ego examiner belongs to the selected workgroups.

Create Network: We create a network from the filtered edges using the graph_from_data_frame function, considering the directed nature of the relationships.

Calculating Centrality: We calculate the degree centrality for each examiner in the network, which measures the number of connections an examiner has. Degree centrality is chosen as it's a simple and widely used measure of centrality, indicating the prominence or importance of a node in a network based on its connections. It's particularly relevant in this context as it helps identify how well-connected or influential each examiner is within the advice network

we visualize the degree centrality distribution for workgroups using a histogram to understand the distribution of centrality scores among examiners in that workgroup.

This approach helps us understand the centrality of examiners within the advice networks, providing insights into their influence or importance in information flow and decision-making processes within the selected workgroups.

To delve deeper into the relationship between degree centrality and other examiner characteristics, additional analyses can be performed using the merged applications dataframe. Descriptive statistics, visualizations, and statistical tests can be employed to explore correlations between centrality scores and variables such as gender, race, tenure, and others. This comprehensive approach facilitates a nuanced understanding of how centrality relates to various examiner attributes, shedding light on organizational dynamics and collaboration patterns within the patent examination domain

#“‘{r} # Merge with applications data for further analysis applications <- left_join(applications, degree_df, by = “examiner_id”)

#“‘

#“‘{r} # Visualization of degree centrality for workgroup 160 as an example applications$examiner_art_unit <- trimws(applications$examiner_art_unit)

wg_160_data <- applications %>% filter(substr(examiner_art_unit, 1, 3) == “160” & !is.na(degree_centrality))

ggplot(wg_160_data, aes(x = degree_centrality)) + geom_histogram(binwidth = 1, fill = “blue”, alpha = 0.7) + theme_minimal() + labs(title = “Degree Centrality Distribution - Workgroup 160”, x = “Degree Centrality”, y = “Frequency”) ##{r} # Visualization of degree centrality for workgroup 175 as an example applications$examiner_art_unit < -trimws(applications$examiner_art_unit)

wg_160_data <- applications %>% filter(substr(examiner_art_unit, 1, 3) == “175” & !is.na(degree_centrality))

ggplot(wg_160_data, aes(x = degree_centrality)) + geom_histogram(binwidth = 1, fill = “blue”, alpha = 0.7) + theme_minimal() + labs(title = “Degree Centrality Distribution - Workgroup 175”, x = “Degree Centrality”, y = “Frequency”) ##{r} # Descriptive statistics summary(applications$degree_centrality)

Visualizations

Scatter plot of degree centrality against tenure

ggplot(applications, aes(x = tenure_days, y = degree_centrality)) + geom_point() + labs(title = “Scatter Plot of Degree Centrality vs Tenure”, x = “Tenure (Days)”, y = “Degree Centrality”)

Box plot of degree centrality by gender

ggplot(applications, aes(x = gender, y = degree_centrality)) + geom_boxplot() + labs(title = “Box Plot of Degree Centrality by Gender”, x = “Gender”, y = “Degree Centrality”)

Statistical tests

Correlation between degree centrality and tenure

cor.test(applications$tenure_days, applications$degree_centrality)

ANOVA test for degree centrality across different races

anova_model <- aov(degree_centrality ~ race, data = applications) summary(anova_model)

#“‘