Group Project Network Analysis

group_2

2024-04-06

Load packages

We loaded the relevant datasets containing details of patent applications and examiner interactions. After ensuring the data completeness by removing entries with missing key examiner details, we explored the dimensions of our cleaned dataset. Guessed the gender and race and calculated tenure days for each examiner.

Introducing variables like gender, race, and tenure days into our analysis is essential to address potential disparities and biases in the patent examination process. Gender and race variables allow us to explore if and how the examination outcomes differ among different demographic groups, highlighting possible systemic biases. The tenure days variable helps us assess the impact of an examiner's experience on the efficiency and thoroughness of patent reviews. By examining these factors, we aim to provide insights into how the USPTO can improve fairness and efficiency in its operations.

```
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!

## 451 (15%) individuals' last names were not matched.
```

```
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  4046452 216.2    8278754 442.2  5492795 293.4
## Vcells 42147375 321.6   85891125 655.3 69863096 533.1
```

```
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  4053231 216.5   11977684 639.7  14972105 799.6
## Vcells 48872469 372.9  103149350 787.0 103101795 786.7
```

Data will be prepared for creating network graphs and adding centrality values

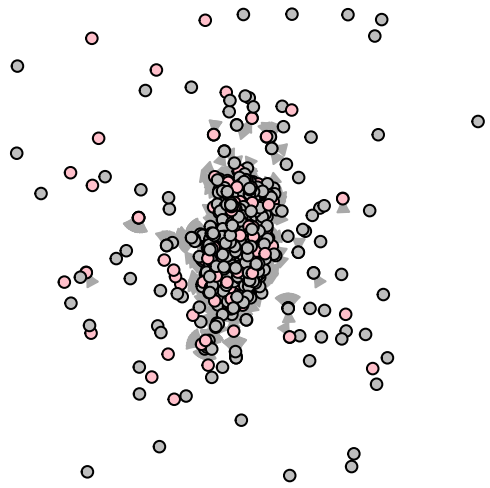
Obtain examiner attributes and join it to the edge dataset. In the edge dataset, there exist observations whose ego and alter examiner id are the same, filtered them out before proceeding with the analysis

```
# Ensure there are no duplicates and calculate mean tenure properly
ego <- advice_linkage_with_info %>%
  group_by(ego_examiner_id) %>%
  summarize(
```

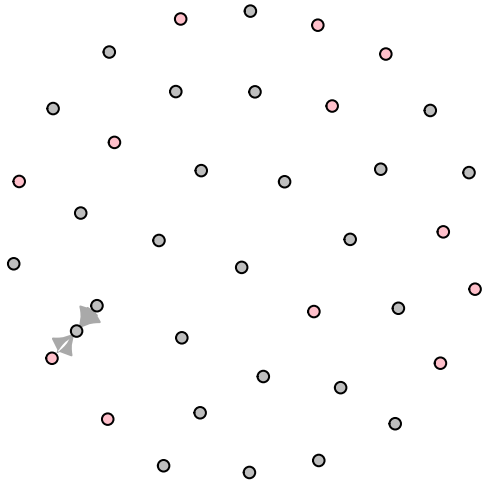
```
    race = names(which.max(table(race.x))),
    tenure = mean(tenure_days.x, na.rm = TRUE),
    gender = names(which.max(table(gender.x)))
  ) %>%
  ungroup() %>%
  distinct(ego_examiner_id, .keep_all = TRUE) %>%
  rename(examiner_id = ego_examiner_id)

alter <- advice_linkage_with_info %>%
  group_by(alter_examiner_id) %>%
  summarize(
    race = names(which.max(table(race.y))),
    tenure = mean(tenure_days.y, na.rm = TRUE),
    gender = names(which.max(table(gender.y)))
  ) %>%
  ungroup() %>%
  distinct(alter_examiner_id, .keep_all = TRUE) %>%
  rename(examiner_id = alter_examiner_id)

# Merge ego and alter data sets with a full join
examiner_attrs <- full_join(ego, alter, by = "examiner_id") %>%
  distinct(examiner_id, .keep_all = TRUE) %>%
  mutate(
    gender = coalesce(gender.x, gender.y),
    race = coalesce(race.x, race.y),
    tenure = coalesce(tenure.x, tenure.y)
  ) %>%
  select(-ends_with(".x"), -ends_with(".y"))
# Get IDs from the graph
graph_ids <- as.character(V(g)$name)

# Ensure examiner_attrs only contains IDs present in the graph
examiner_attrs <- examiner_attrs %>%
  filter(examiner_id %in% graph_ids)

# Check lengths again
if(nrow(examiner_attrs) != length(V(g))) {
  stop("Still a mismatch in examiner attributes and graph vertices")
}

# Assign attributes
V(g)$race <- examiner_attrs$race[match(graph_ids, examiner_attrs$examiner_id)]
V(g)$gender <- examiner_attrs$gender[match(graph_ids, examiner_attrs$examiner_id)]
V(g)$tenure <- examiner_attrs$tenure[match(graph_ids, examiner_attrs$examiner_id)]
```

```
# Set vertex colors based on gender
vertex_colors <- ifelse(V(g)$gender == "female", "pink", "gray")

# Loop through unique races in the graph to create plots for each race
for(r in unique(V(g)$race)) {
    # Subset the graph to include only vertices of a specific race
    g_sub <- induced_subgraph(g, V(g)$race %in% c(r))
```

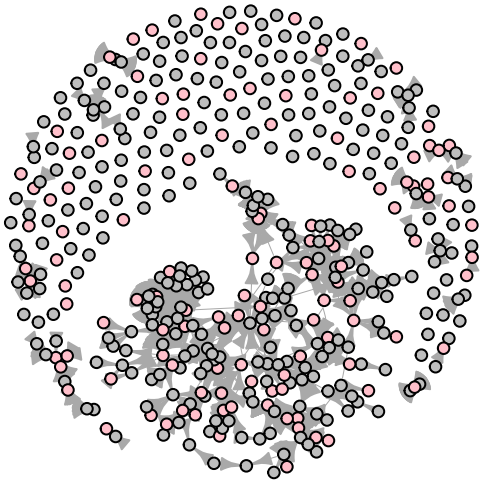
```
    # Plot the subgraph using igraph with specified layout and color by gender
    plot(g_sub, vertex.label = NA, vertex.color = vertex_colors[V(g_sub)], vertex.size = 5,
         edge.arrow.size = 0.5, edge.width = 0.5, layout = layout_with_fr(g_sub))
    title(r)
}
```



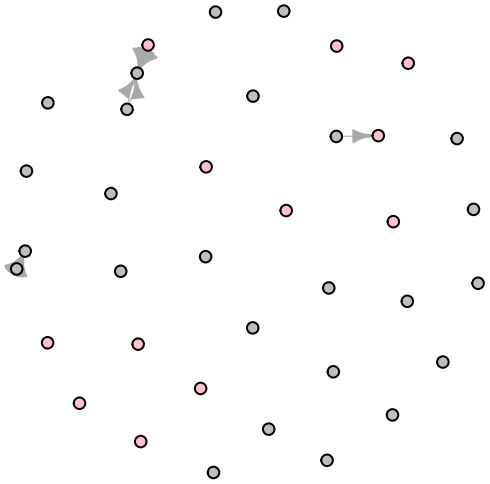
black

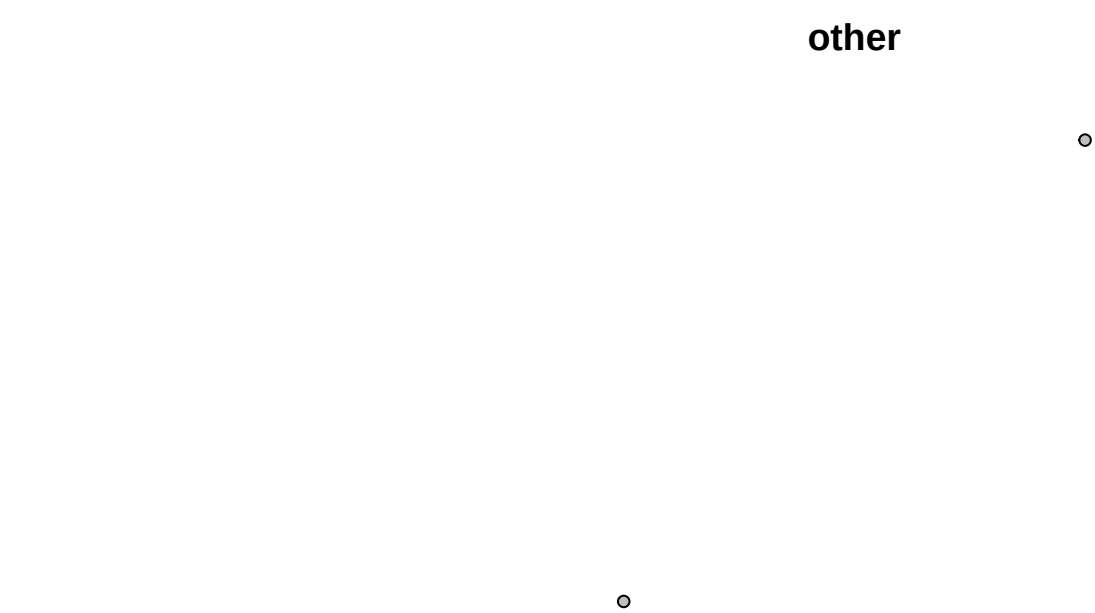


Asian



Hispanic





Analysis:

Plot for white race: graph shows a discernible clustering in the center, indicating a core group of examiners (both male and female) who are likely more active in exchanging advice

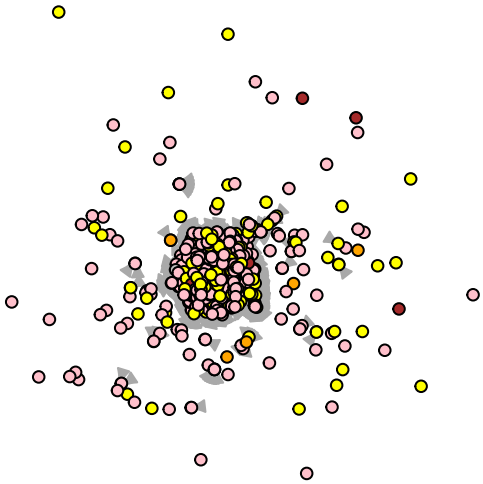
Plot for black race and Hispanic :both race examiners are not involved in giving each other advices among themesleves

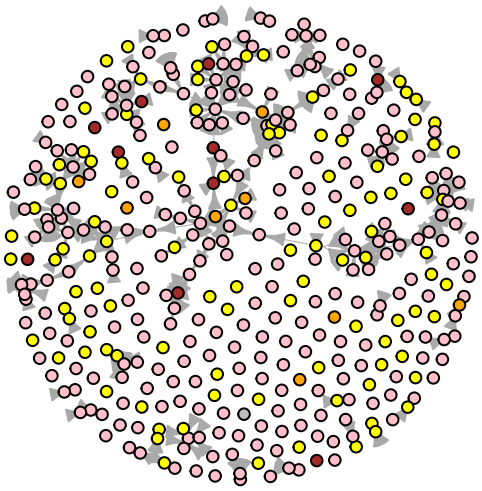
Plot for Asian race: a significant number of exmainers both male and female are active in exhanging advices

Conclusion: White and Asiana are more active in exhanging advices with white being more among themsleves

```
library(igraph)

# Create vertex colors based on race
vertex_colors <- ifelse(V(g)$race == "white", "pink",
                        ifelse(V(g)$race == "Asian", "yellow",
                               ifelse(V(g)$race == "black", "brown",
                                      ifelse(V(g)$race == "Hispanic", "orange", "gray"))))

# Loop through unique genders in the graph to create plots for each gender
for(r in unique(V(g)$gender)) {
    # Subset the graph to include only vertices of a specific gender
    g_sub <- induced_subgraph(g, V(g)$gender %in% r)

    # Plot the subgraph using igraph with specified layout and color by race
    plot(g_sub, vertex.label = NA, vertex.color = vertex_colors[V(g_sub)], vertex.size = 5,
         edge.arrow.size = 0.5, edge.width = 0.5, layout = layout_with_fr(g_sub))
```

```
    title(r)
}
```





Analysis:

Male Plot: a substantial cluster of examiners, suggesting active engagement in advice sharing. Asian and White male are more active in advice sharing especially between themsleves

Female Plot:Females are less involved in advice sharing and it can be seen that white females among themselves are more active in advice sharing similar is the case for Asian females

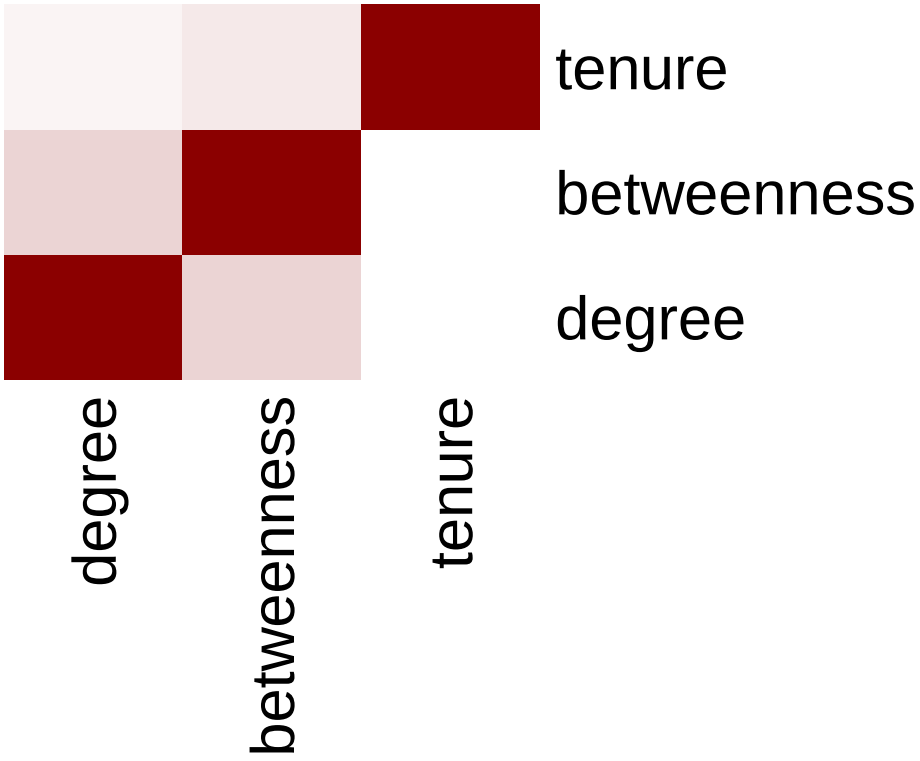
Further Processing the data

Calculate and degree and betweenness for the network and join it to examiner attributes.

```
cor_matrix <- cor(selected_centrality_scores[, c("degree", "betweenness","tenure")])

print(cor_matrix)
```

```
##                degree betweenness     tenure
## degree      1.00000000   0.2090235 0.01535878
## betweenness 0.20902352   1.0000000 0.04306280
## tenure      0.01535878   0.0430628 1.00000000
```

```
heatmap(
  cor_matrix,
  Rowv = NA,
  Colv = NA,
  col = colorRampPalette(c("white", "darkred"))(25),
  margins = c(15, 10)
)
```

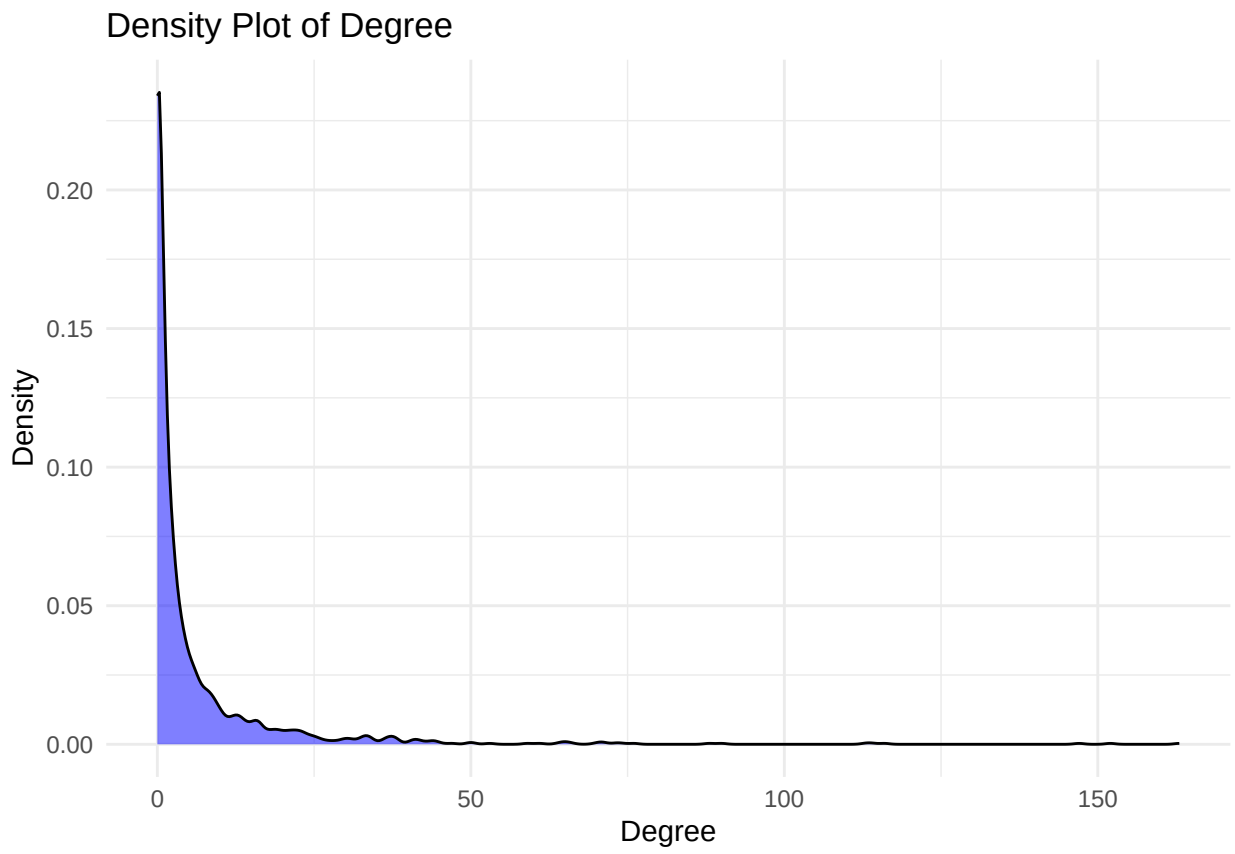


Correlation Analysis: it can be seen that tenure, degree and betweenes are not highly correlated with each other which is expected

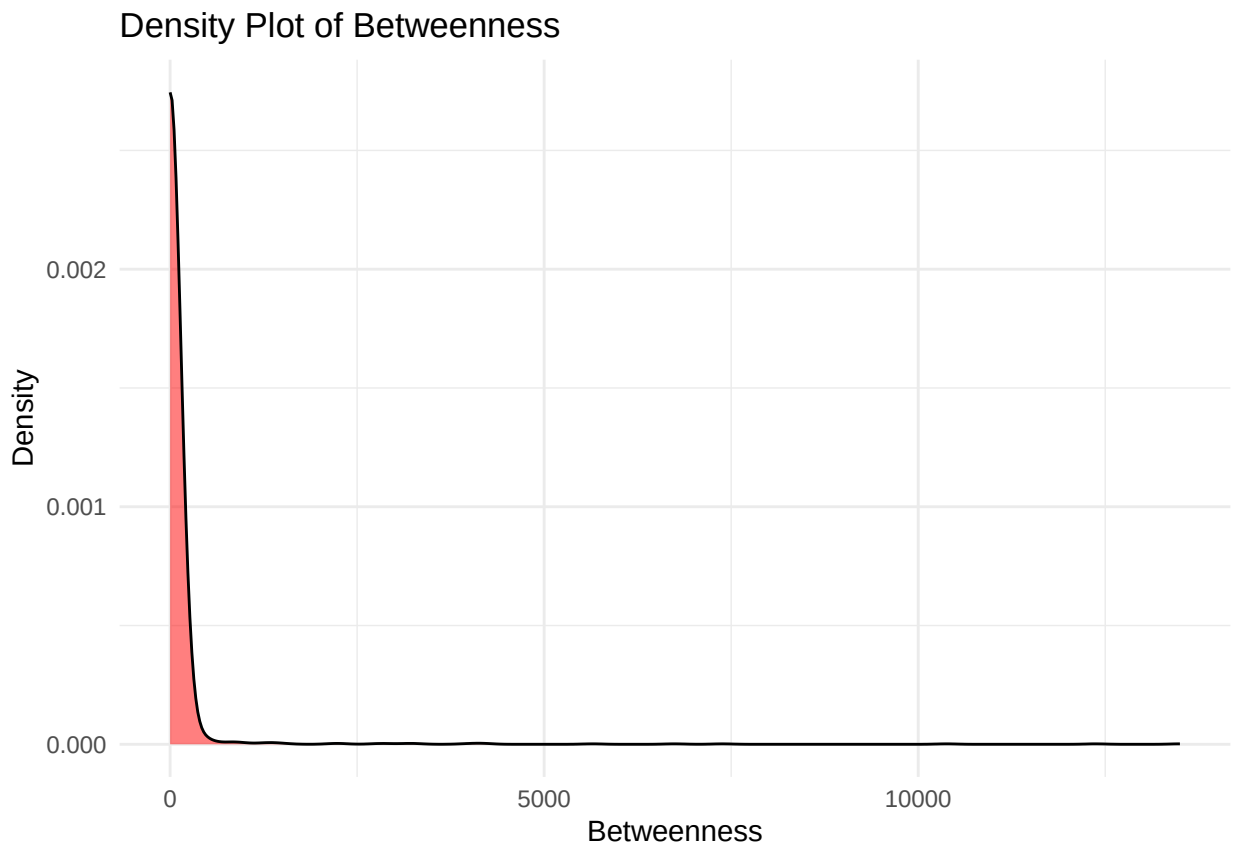
Further Processing the data

Join the centrality score to applications and process the merged dataset to obtain processing time. As degree, betweeness are right skewed, a log transformation and standardization is applied and Since tenure days and centrality are in different measures, they are standardized before the regression analysis.

```
# Density plot for 'degree'
ggplot(centrality, aes(x = degree)) +
  geom_density(fill = "blue", alpha = 0.5) +
  labs(title = "Density Plot of Degree", x = "Degree", y = "Density") +
  theme_minimal()
```




```r
# Density plot for 'betweenness'
ggplot(centrality, aes(x = betweenness)) +
  geom_density(fill = "red", alpha = 0.5) +
  labs(title = "Density Plot of Betweenness", x = "Betweenness", y = "Density") +
  theme_minimal()
```




```
centrality <- centrality %>% mutate(log_degree = log(degree), log_betweenness = log(betweenness))%>%
  mutate(
    log_degree = replace(log_degree, log_degree == -Inf, 0),
    log_betweenness = replace(log_betweenness, log_betweenness == -Inf, 0)
  )
```

```
applications_mutated <- applications %>%
  mutate(app_proc_time_issue = patent_issue_date - filing_date,
         app_proc_time_abandon = abandon_date - filing_date) %>%
  mutate(app_pro_time = ifelse(is.na(app_proc_time_issue),
                               app_proc_time_abandon, app_proc_time_issue)) %>%
  filter(app_pro_time > 0) %>%
  select(app_pro_time, gender, race, tenure_days, log_degree, log_betweenness, tc) %>%
  drop_na() %>%
  mutate(
    log_degree = (log_degree - mean(log_degree)) / sd(log_degree),
    log_betweenness = (log_betweenness - mean(log_betweenness)) / sd(log_betweenness),
    tenure_days = (tenure_days - mean(tenure_days)) / sd(tenure_days)
    # No change to tc as it is categorical
  )
applications_mutated <- applications_mutated %>%
  filter(app_pro_time <= 2000)
```

Note: After visualzaising applcaition processing time comes up later in EDA part it was seen that there are outliers in this so greater than 2000 days has been considered outlier also only processing times greater than 0 are included this is decided after looking at the distrubtion of values and considering mean

```r
# Factorizing the categorical variables
applications_mutated$gender <- as.factor(applications_mutated$gender)
applications_mutated$race <- as.factor(applications_mutated$race)
applications_mutated$tc <- as.factor(applications_mutated$tc)
```

Regression model

```r
model_2 <- lm(app_pro_time ~ log_degree + log_betweenness  +
                          gender + race + tenure_days +
                          log_degree:gender +
                          log_betweenness:gender + tenure_days:gender +
                 log_degree:race + log_betweenness:race + tenure_days:race + tc,  data = applications_mu
```

```r
summary(model_2)
```

```
##
## Call:
## lm(formula = app_pro_time ~ log_degree + log_betweenness + gender +
##     race + tenure_days + log_degree:gender + log_betweenness:gender +
##     tenure_days:gender + log_degree:race + log_betweenness:race +
##     tenure_days:race + tc, data = applications_mutated)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1206.6  -300.2   -23.8   286.9  1032.7
##
## Coefficients: (2 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1009.55621    1.69590 595.291  < 2e-16 ***
## log_degree                     25.40401    1.59915  15.886  < 2e-16 ***
## log_betweenness                12.72942    1.46346   8.698  < 2e-16 ***
## gendermale                     -6.35930    1.21307  -5.242 1.59e-07 ***
## raceblack                      11.23396    3.85076   2.917 0.003530 **
## raceHispanic                    1.13986    4.26892   0.267 0.789459
## raceother                     134.31215   13.94044   9.635  < 2e-16 ***
## racewhite                     -12.72897    1.32165  -9.631  < 2e-16 ***
## tenure_days                    -8.35162    1.34475  -6.211 5.28e-10 ***
## tc1700                         22.90810    1.42356  16.092  < 2e-16 ***
## tc2100                        169.04821    1.71483  98.580  < 2e-16 ***
## tc2400                         95.73142    1.98808  48.153  < 2e-16 ***
## log_degree:gendermale          -4.96000    1.33244  -3.722 0.000197 ***
## log_betweenness:gendermale      2.73935    1.34479   2.037 0.041649 *
## gendermale:tenure_days         -0.06023    1.27766  -0.047 0.962404
## log_degree:raceblack          -11.77472    4.62745  -2.545 0.010942 *
## log_degree:raceHispanic       -25.95658    4.21688  -6.155 7.49e-10 ***
## log_degree:raceother          -90.99896   14.55167  -6.254 4.02e-10 ***
## log_degree:racewhite          -10.70987    1.42758  -7.502 6.29e-14 ***
## log_betweenness:raceblack       2.14462    5.66081   0.379 0.704796
## log_betweenness:raceHispanic   21.44965    4.89239   4.384 1.16e-05 ***
## log_betweenness:raceother            NA         NA      NA       NA
## log_betweenness:racewhite     -12.05342    1.25049  -9.639  < 2e-16 ***
## raceblack:tenure_days         -42.02431    5.17635  -8.119 4.73e-16 ***
```

```
## raceHispanic:tenure_days        1.16061    5.02030   0.231 0.817173
## raceother:tenure_days                NA         NA      NA       NA
## racewhite:tenure_days         -14.46995    1.21934 -11.867  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 410.8 on 603840 degrees of freedom
## Multiple R-squared:  0.03136,    Adjusted R-squared:  0.03132
## F-statistic: 814.5 on 24 and 603840 DF,  p-value: < 2.2e-16
```

Explanation:

(Intercept): Represents the baseline patent prosecution time when all other variables are zero. It is significantly high, indicating a substantial processing time when no other factors are taken into account.

log_degree: Exhibits a positive coefficient, indicating that as an examiner's network connectivity increases, so does the processing time for patents.

log_betweenness: Also has a positive coefficient, suggesting that examiners who frequently bridge communication between other examiners are associated with longer patent processing times.

gendermale: Shows a negative coefficient, which means male examiners are associated with shorter patent processing times compared to their female counterparts.

raceblack: This positive coefficient signifies that examiners identified as black are associated with longer processing times compared to the baseline race category.

raceHispanic: Although this has a positive coefficient, it's not statistically significant, suggesting no clear association with processing time for Hispanic examiners.

raceother: Has a very large positive coefficient, indicating a strong association with increased processing times for examiners categorized in the 'other' race group.

racewhite: With a negative coefficient, white examiners are associated with shorter processing times compared to the baseline race category.

tenure_days: A negative coefficient here indicates that with each additional tenure day, an examiner's patent processing time decreases, suggesting more experienced examiners work faster.

tc1700, tc2100, tc2400: Each of these technology center variables has a positive coefficient, with 'tc2100' having the largest, meaning patent applications in these centers take longer to process, particularly in 'tc2100'.

log_degree:gendermale: The negative interaction term suggests the increase in processing time due to network degree is less for male examiners.

log_betweenness:gendermale: A positive coefficient here, though smaller, indicates male examiners with high betweenness might experience a slight increase in processing times.

gendermale:tenure_days: This interaction term is not statistically significant, indicating tenure does not affect processing times for male examiners differently than for females.

log_degree:raceblack, log_degree:raceHispanic, log_degree:raceother, log_degree:racewhite: These terms all have negative coefficients (except for Hispanic which is positive), suggesting that increased network degree leads to longer processing times, but the effect is more pronounced for 'raceother' and less for 'racewhite'.

log_betweenness:raceblack, log_betweenness:raceHispanic, log_betweenness:racewhite: Here we see varying effects. The positive coefficient for 'raceHispanic' is significant, suggesting Hispanic examiners with higher betweenness see a more pronounced increase in processing times. The negative coefficient for 'racewhite' indicates the opposite.

raceblack:tenure_days, raceHispanic:tenure_days, racewhite:tenure_days: The tenure of black examiners significantly reduces processing time, as indicated by the negative coefficient. This reduction is not observed for Hispanic examiners, while it is even more pronounced for white examiners

```r
# Install necessary packages if not already installed
if (!require("randomForest")) install.packages("randomForest", dependencies = TRUE)
```

```
## Loading required package: randomForest

## Warning: package 'randomForest' was built under R version 4.2.3

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
if (!require("shapr")) install.packages("shapr", dependencies = TRUE)
```

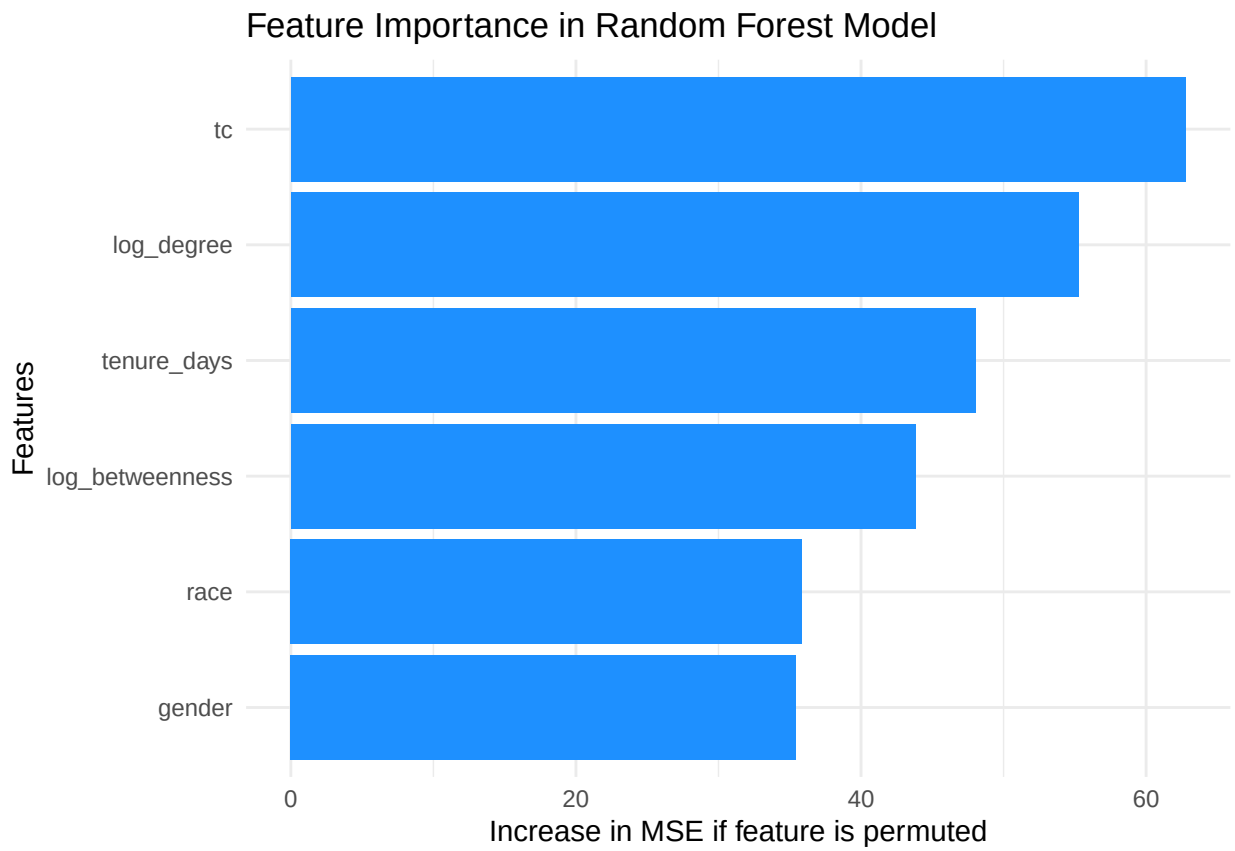
```
## Loading required package: shapr

## Warning: package 'shapr' was built under R version 4.2.3

##
## Attaching package: 'shapr'

## The following object is masked from 'package:dplyr':
##
##     explain
```

```r
if (!require("ggplot2")) install.packages("ggplot2", dependencies = TRUE)

# Load the packages
library(randomForest)
library(shapr)
library(ggplot2)
```

```r
library(randomForest)  # Load randomForest library
library(ggplot2)       # Load ggplot2 for plotting

applications_mutated <- applications_mutated %>% slice_sample(n = 50000)

rf_model <- randomForest(app_pro_time ~ log_degree + log_betweenness +
                         gender + race + tenure_days +
```

```
                          log_degree:gender +
                          log_betweenness:gender + tenure_days:gender +
                          log_degree:race + log_betweenness:race + tenure_days:race + tc,
                        data = applications_mutated,
                        ntree = 100,
                        importance = TRUE)
```

```
# Extracting importance
importance_data <- importance(rf_model)

# Creating a data frame for plotting
feature_importance <- data.frame(
  Feature = rownames(importance_data),
  Importance = importance_data[, '%IncMSE']
)

# Plotting feature importance using ggplot2
plot <- ggplot(feature_importance, aes(x = reorder(Feature, Importance), y = Importance)) +
  geom_col(fill = "dodgerblue") +
  coord_flip() +  # Flips the axes to make the plot horizontal
  theme_minimal() +
  labs(title = "Feature Importance in Random Forest Model",
       x = "Features",
       y = "Increase in MSE if feature is permuted")

print(plot)
```



Explanation:

Technology Center (tc): This feature has the highest impact on MSE when permuted, indicating that the specific technology center handling the patent application is the most significant predictor of processing time. Variations between centers could reflect differences in process efficiency, case complexity, or resource allocation.

Network Metrics (log_betweenness, log_degree): These features also show a substantial effect on the MSE, highlighting the significant role that an examiner's position within the advice-sharing network has on patent prosecution times. Examiners who are central in the network or have more connections might be handling more complex applications or be more integral to the examination process, affecting timeframes.

Tenure (tenure_days): Examiner tenure is another influential factor, suggesting that more experienced examiners have a notable impact on prosecution times, potentially due to better familiarity with the patent process or greater efficiency in handling applications.

Demographics (gender, race): These features have a lesser but still meaningful impact on MSE when permuted, implying that demographic characteristics of the examiners do contribute to variations in processing times, though their influence is not as strong as organizational factors like technology center designation or network metrics

```
applications <- applications %>%
  mutate(
    filing_date = as.Date(filing_date),
    patent_issue_date = as.Date(patent_issue_date),
    abandon_date = as.Date(abandon_date),
    final_decision_date = coalesce(patent_issue_date, abandon_date),
    app_proc_time = as.numeric(final_decision_date - filing_date),
```

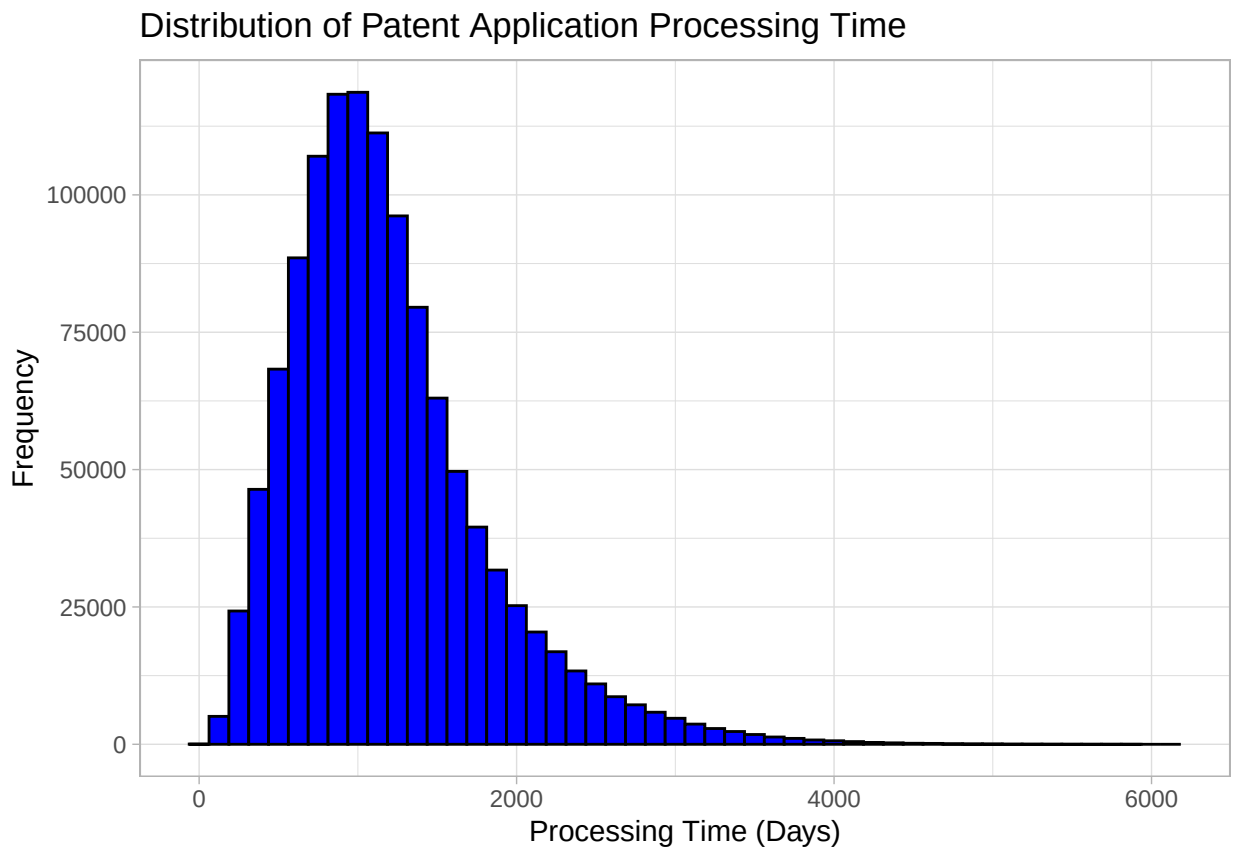
```
    # Replace negative app_proc_time with NA
    app_proc_time = ifelse(app_proc_time < 0, NA, app_proc_time)
  )
```

Plots and Graphs: EDA

```
library(ggplot2)

# Set the theme globally for all ggplot2 plots
theme_set(theme_light())
```

```
theme_set(theme_light())
# Plot : Histogram of Application Processing Times
library(ggplot2)

# Assuming 'app_proc_time' is in days.
ggplot(applications, aes(x = app_proc_time)) +
  geom_histogram(bins = 50, fill = "blue", color = "black") +
  labs(title = "Distribution of Patent Application Processing Time",
       x = "Processing Time (Days)",
       y = "Frequency") +
  theme(
    plot.background = element_rect(
      fill = "white",
      colour = "white"
    )
  )
```

```
## Warning: Removed 223738 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
# Save the plot to a new directory
dir.create("USPTO_Analysis_Plots")
```

```
## Warning in dir.create("USPTO_Analysis_Plots"): 'USPTO_Analysis_Plots' already
## exists
```

```
ggsave("USPTO_Analysis_Plots/histogram_processing_time.png", type = "cairo")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Using ragg device as default. Ignoring `type` and `antialias` arguments
## Removed 223738 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

Explanation: The distribution of application shows right skewed graph with clearly outliers seen. Outliers are consider as greater than 2000 days and are removed before runnning the model. Most application take around 1000 days to process

```
{r}
```

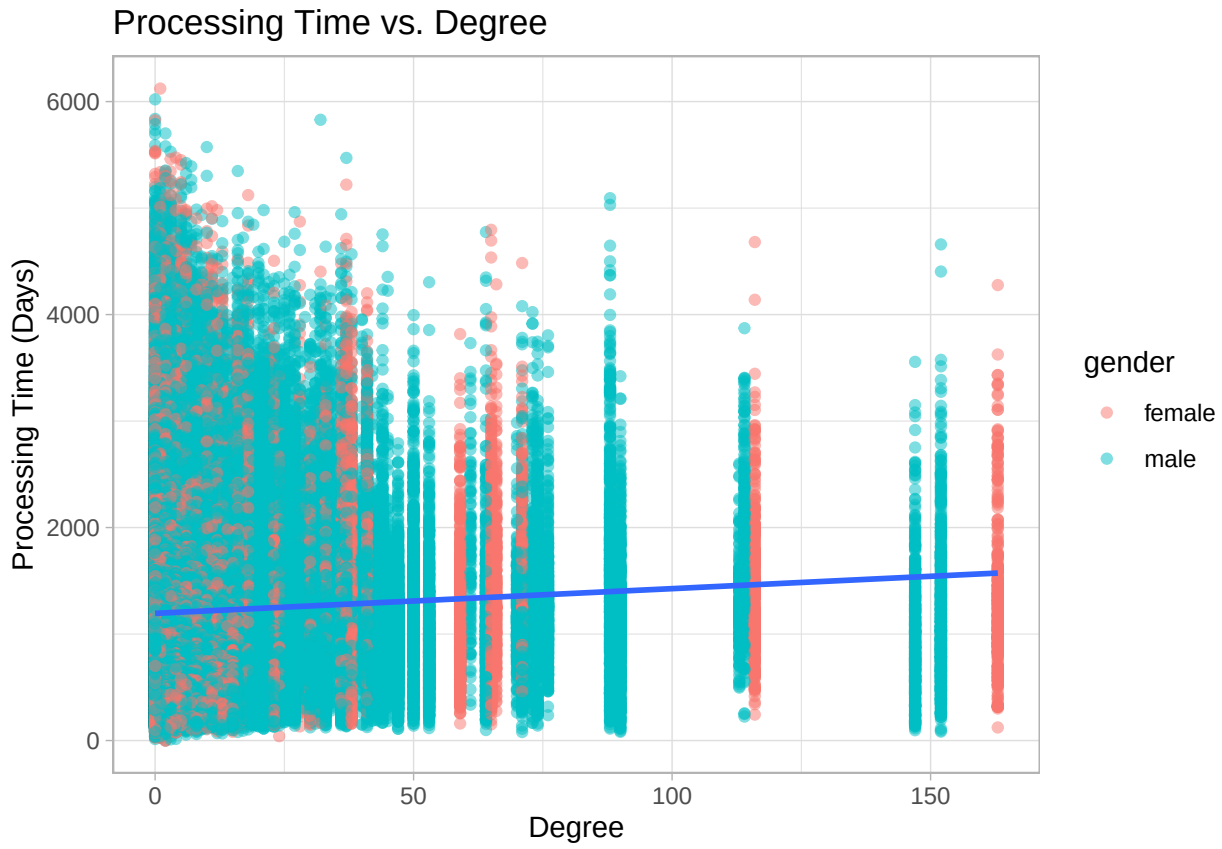
```
## [1] "female"
```

```r
# Plot : Scatter Plot of Processing Time vs. Examiner Betweenness
ggplot(applications, aes(x = degree, y = app_proc_time)) +
  geom_point(aes(color = gender), alpha = 0.5) +
  geom_smooth(method = "lm") +
  labs(title = "Processing Time vs. Degree",
       x = "Degree",
       y = "Processing Time (Days)") +
    theme(
    plot.background = element_rect(
      fill = "white",
      colour = "white"
    )
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 726524 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 726524 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
ggsave("USPTO_Analysis_Plots/scatter_processing_time_betweenness.png")
```

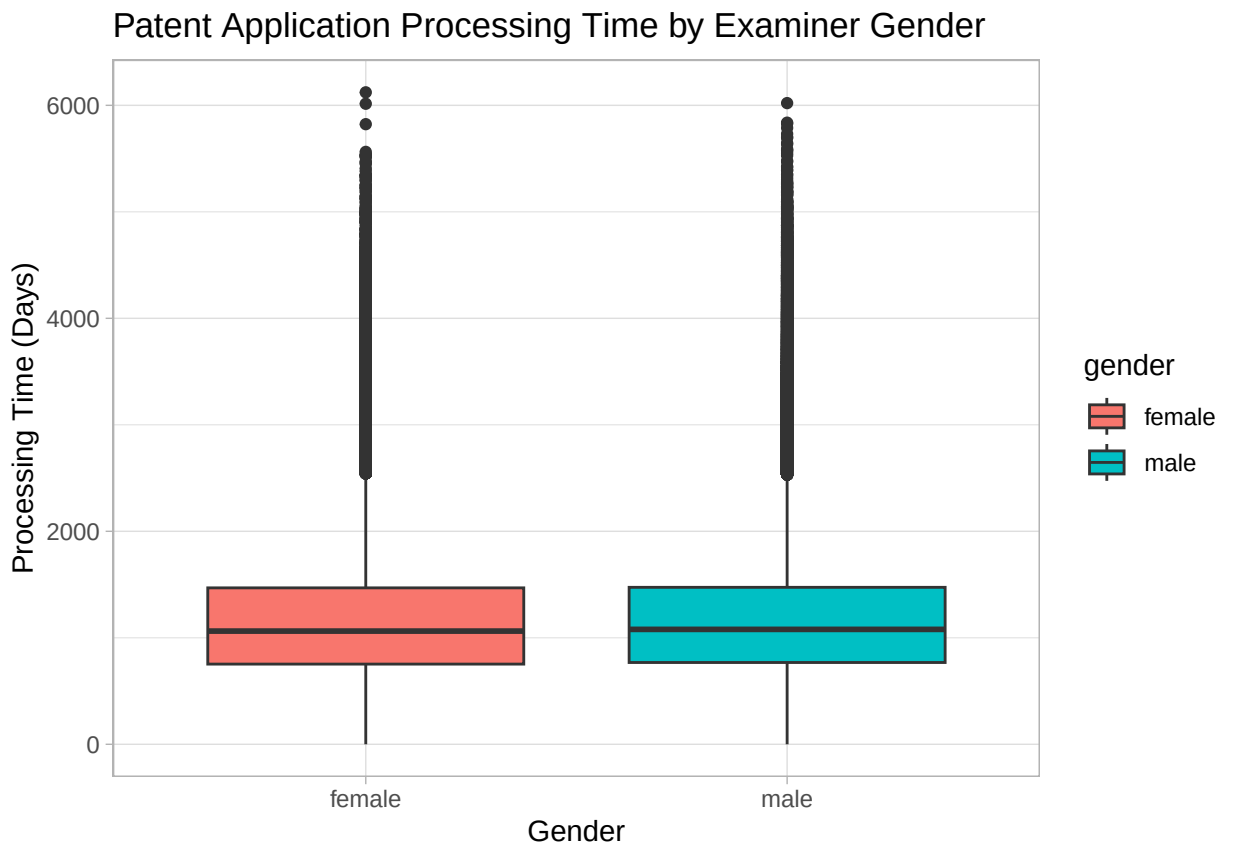
```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 726524 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 726524 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Explanation: Even though not obviuos but it seen that processing time increase with increase in degreee for both genders

```
# Plot : Boxplot of Processing Time by Examiner Gender
ggplot(applications, aes(x = gender, y = app_proc_time, fill = gender)) +
  geom_boxplot() +
  labs(title = "Patent Application Processing Time by Examiner Gender",
       x = "Gender",
       y = "Processing Time (Days)") +
  theme(
    plot.background = element_rect(
      fill = "white",
      colour = "white"
    )
  )
```

```
## Warning: Removed 223738 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



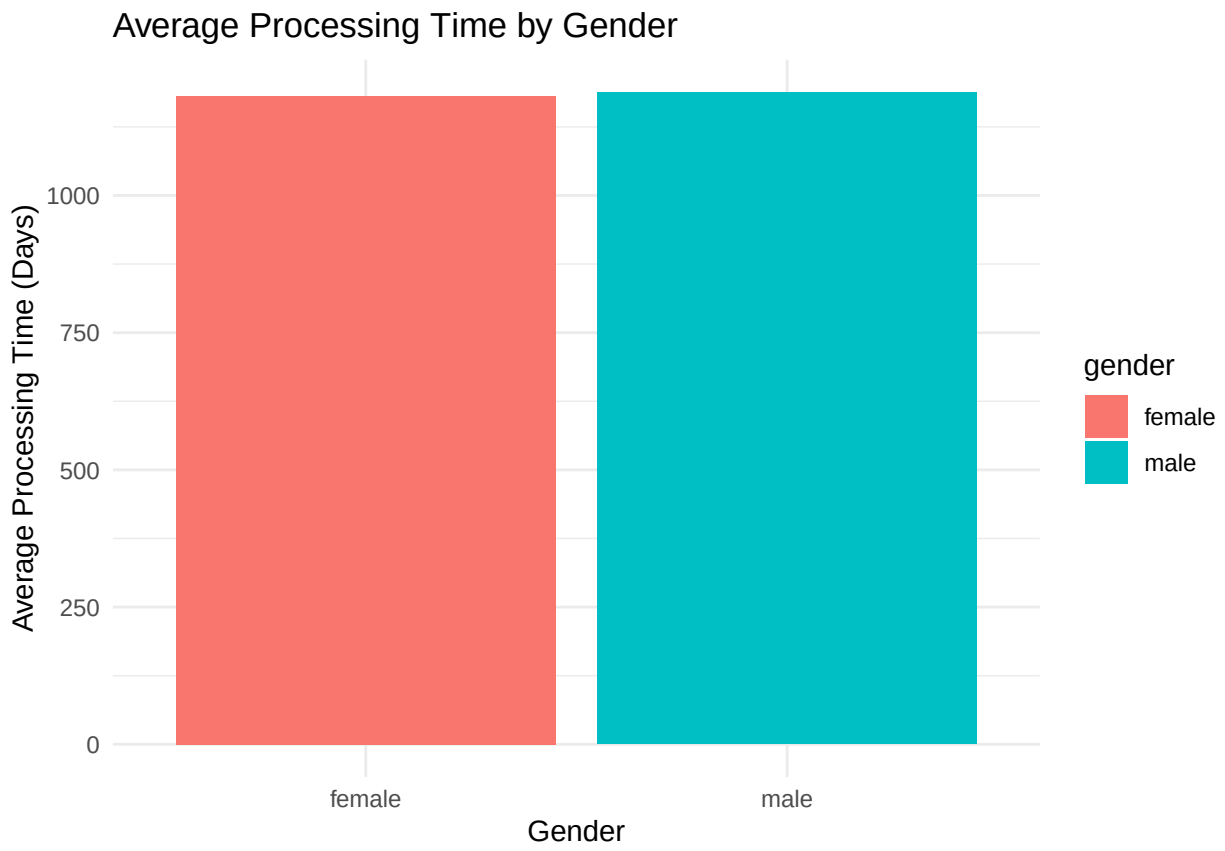

```
ggsave("USPTO_Analysis_Plots/boxplot_processing_time_gender.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 223738 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

Explanation: both gender have mean processing times of 1000 days and utliers seen for both which are removed

```
# Plot : Bar Graph of Average Processing Time by Gender

# Calculate average processing time by gender
avg_time_by_gender <- aggregate(app_proc_time ~ gender, data = applications, FUN = mean)

# Bar Graph of Average Processing Time by Gender
ggplot(avg_time_by_gender, aes(x = gender, y = app_proc_time, fill = gender)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Processing Time by Gender",
       x = "Gender",
       y = "Average Processing Time (Days)") +
  theme_minimal()
```



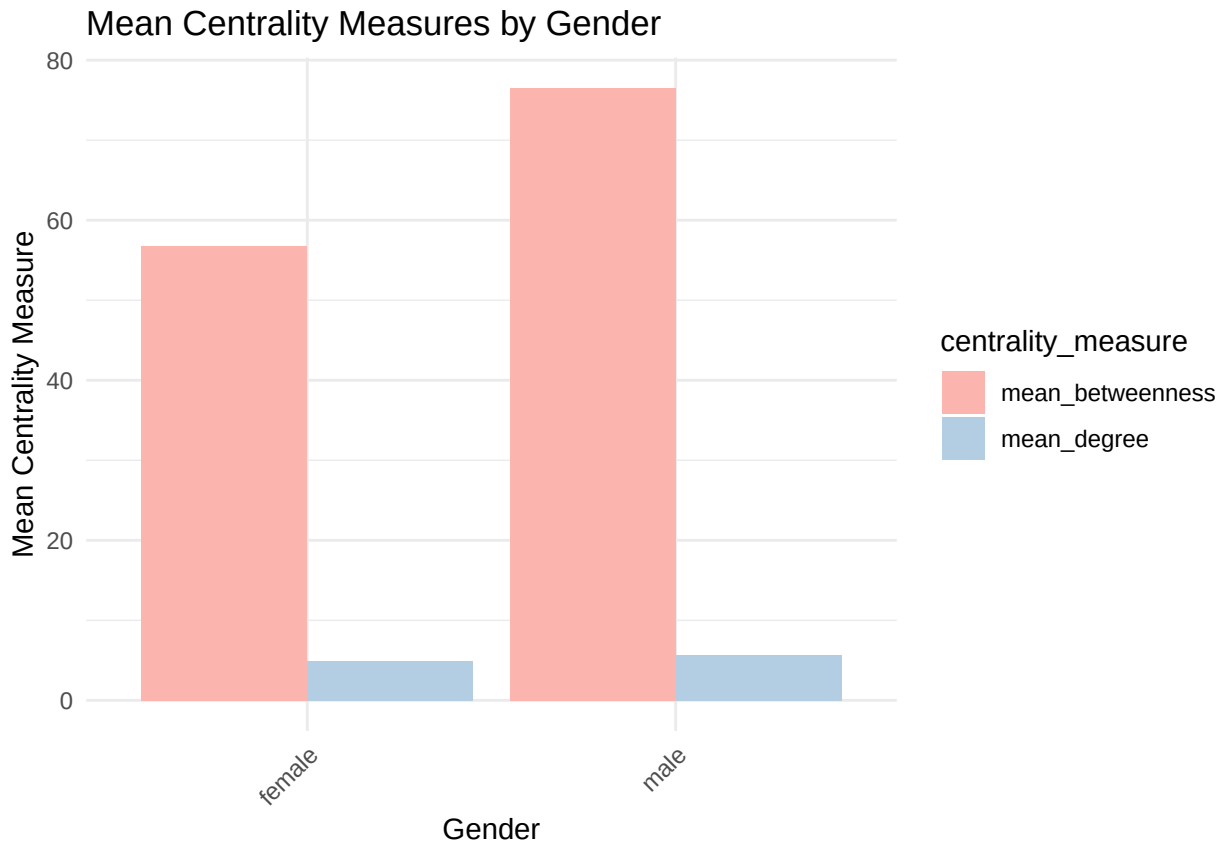

```
# Save the plot
ggsave("USPTO_Analysis_Plots/avg_processing_time_by_gender.png", width = 8, height = 6)
```

```
{r}
```

```
## [1] "female"
```

```
library(dplyr)
library(ggplot2)
library(tidyr)

# Calculate mean centrality measures by gender
centrality_by_gender <- applications %>%
  group_by(gender) %>%
  summarize(mean_degree = mean(degree, na.rm = TRUE),
            mean_betweenness = mean(betweenness, na.rm = TRUE)) %>%
  ungroup()
# Reshape data for plotting
centrality_long <- centrality_by_gender %>%
  pivot_longer(cols = -gender, names_to = "centrality_measure", values_to = "mean_value")
# Plot mean centrality measures by gender
ggplot(centrality_long, aes(x = gender, y = mean_value, fill = centrality_measure)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_brewer(palette = "Pastel1") +
  labs(title = "Mean Centrality Measures by Gender",
```

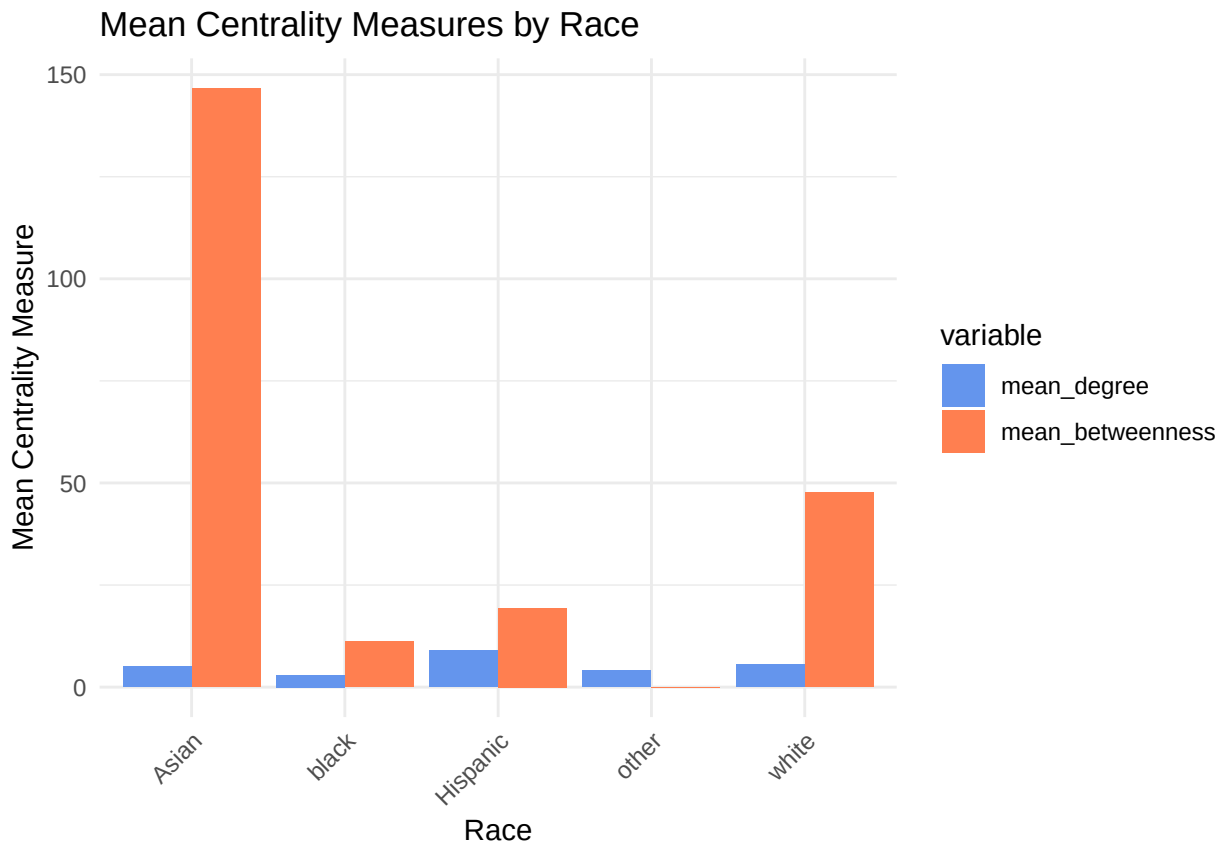
```
       x = "Gender",
       y = "Mean Centrality Measure") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



explanation: similar mean degree measure for both gender showing both genders are equally involed in advice sahring. males have higher betweenes which means they are more important in organziation when it comes to transfering information from one cluster to another

```
# Calculate mean centrality measures by race
centrality_by_race <- applications %>%
  group_by(race) %>%
  summarize(mean_degree = mean(degree, na.rm = TRUE),
            mean_betweenness = mean(betweenness, na.rm = TRUE))

# Melt the data for plotting
centrality_melted <- melt(centrality_by_race, id.vars = "race")

# Use ggplot2 to create a bar plot
ggplot(centrality_melted, aes(x = race, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("cornflowerblue", "coral")) +
  labs(title = "Mean Centrality Measures by Race",
       x = "Race",
       y = "Mean Centrality Measure") +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

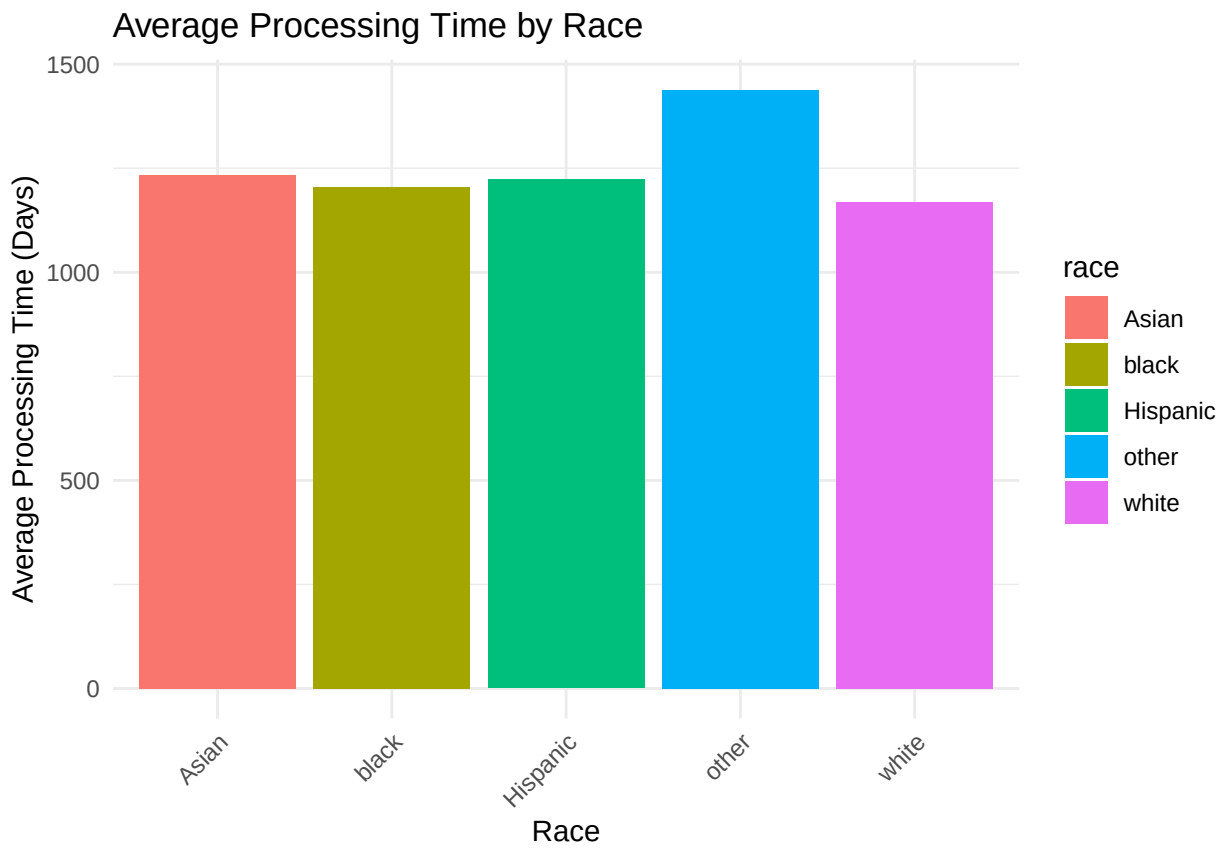


Explanation: Asian have consideraly high betweeenes which could either be an anomaly or they are very important in transfering information from one cluster to another compared to other. White also have high betweeneess. All race have approximately the same degree with hispanic being th e highest showing they are more involved in advice sharing

```
{r}
```

```
## [1] "female"
```

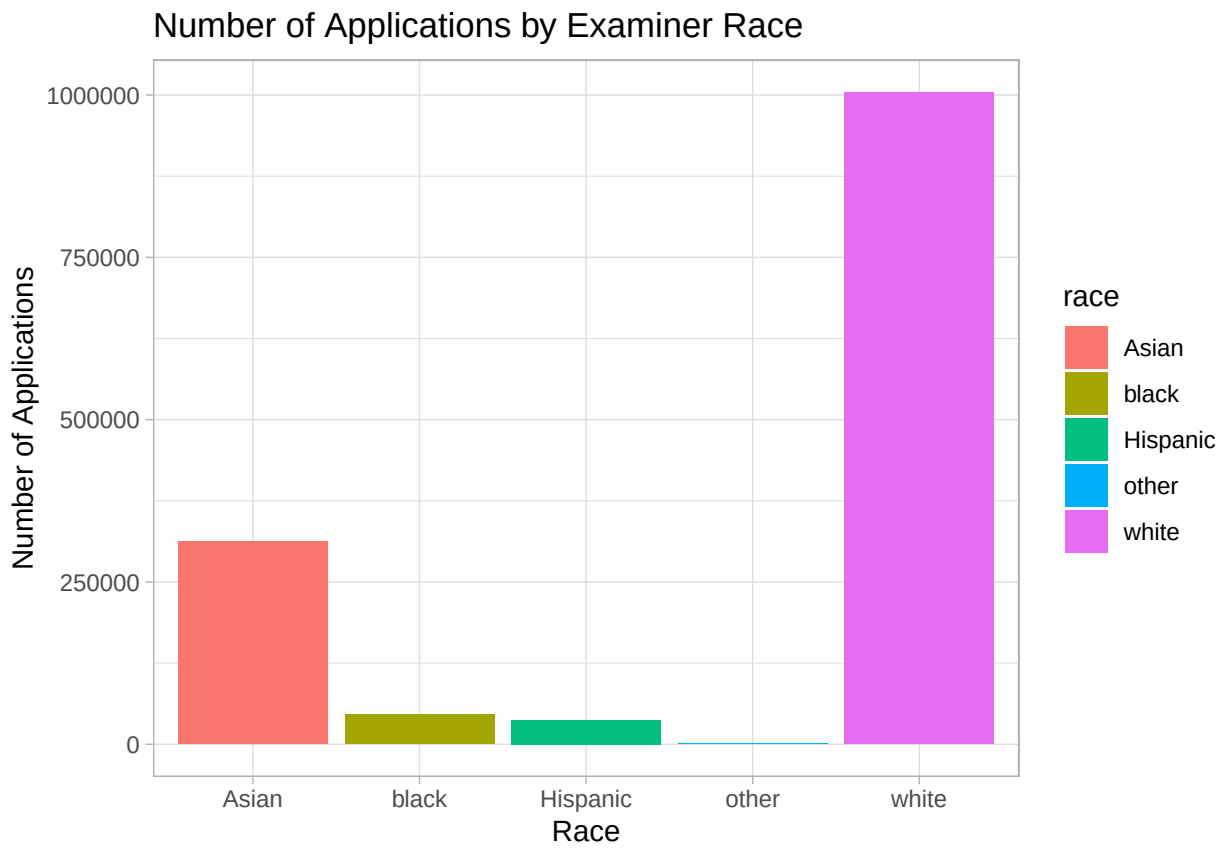
```
# Plot 5: Bar Graph of Average Processing Time by Different Races
# Calculate average processing time by race
avg_time_by_race <- aggregate(app_proc_time ~ race, data = applications, FUN = mean)

# Bar Graph of Average Processing Time by Race
ggplot(avg_time_by_race, aes(x = race, y = app_proc_time, fill = race)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Processing Time by Race",
       x = "Race",
       y = "Average Processing Time (Days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Adjust text angle for better readability i
```




```
# Save the plot
ggsave("USPTO_Analysis_Plots/avg_processing_time_by_race.png", width = 8, height = 6)
```

Explanation: Similar processing times can be seen for all the races at around 1250 days

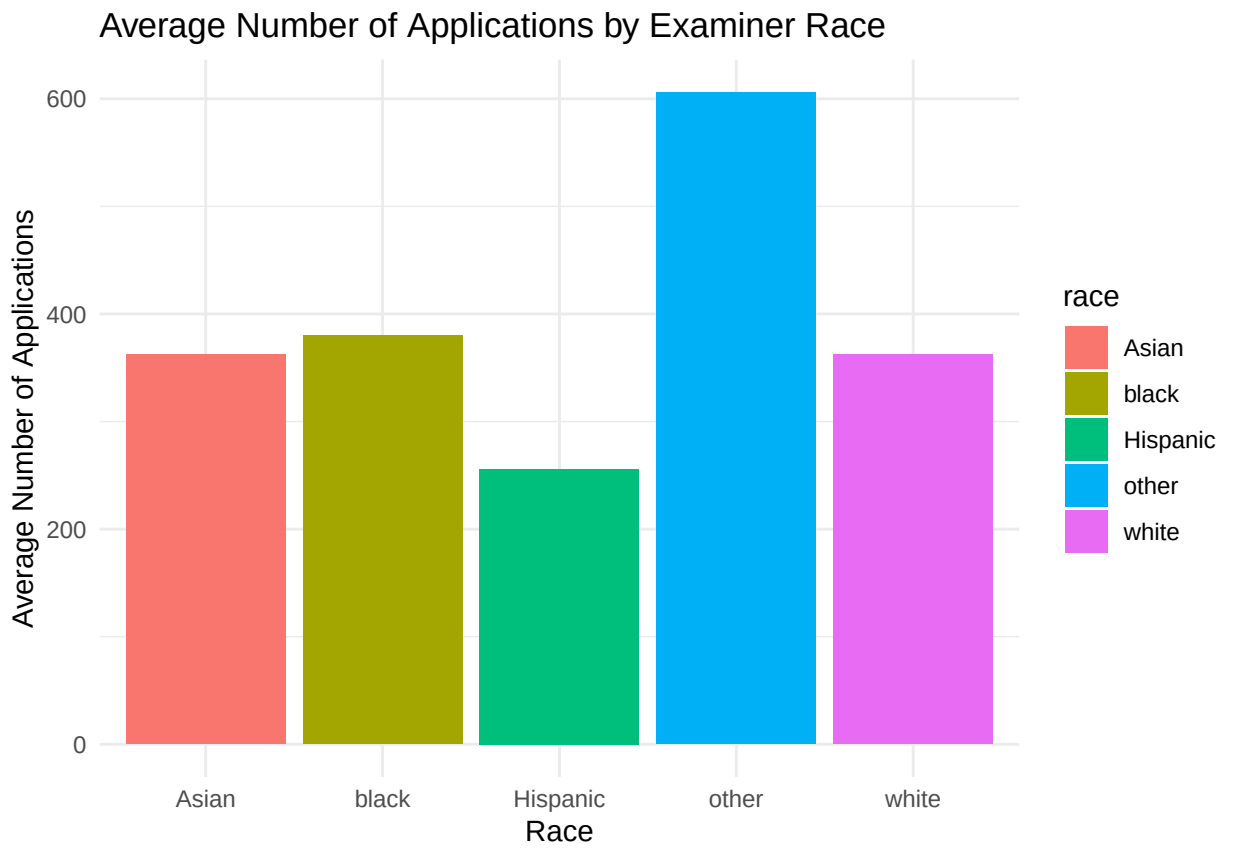
```
# Plot 4: Bar Chart of Number of Applications by Examiner Race
ggplot(applications, aes(x = race, fill = race)) +
  geom_bar() +
  labs(title = "Number of Applications by Examiner Race",
       x = "Race",
       y = "Number of Applications") +
  theme(
    plot.background = element_rect(
      fill = "white",
      colour = "white"
    )
  )
```




```
ggsave("USPTO_Analysis_Plots/bar_applications_race.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
library(ggplot2)
library(dplyr)

# Assuming you have a column 'examiner_id' to identify each examiner
# and the applications dataframe has a 'race' column

# Calculate the average number of applications per examiner by race
avg_applications_by_race <- applications %>%
  group_by(race, examiner_id) %>% # Group by race and examiner
  summarise(num_applications = n(), .groups = "drop") %>%
  group_by(race) %>%
  summarise(avg_applications = mean(num_applications), .groups = "drop")

# Plot the average number of applications by examiner race
ggplot(avg_applications_by_race, aes(x = race, y = avg_applications, fill = race)) +
  geom_bar(stat = "identity") + # stat="identity" to use the actual y values
  labs(title = "Average Number of Applications by Examiner Race",
       x = "Race",
       y = "Average Number of Applications") +
  theme_minimal() + # Using a minimal theme for cleaner look
  theme(plot.background = element_rect(fill = "white", colour = "white"))
```




```
# Save the plot
ggsave("USPTO_Analysis_Plots/avg_bar_applications_race.png")
```

```
## Saving 6.5 x 4.5 in image
```

Explanation: White race has processed highest applications however avg applications per race is same for all races which shows we just have more white examiners. so there seems to be no discrimination when giving appplicaiton to races however when selecting an examiner there could be. Other is ignored as that is an outlier here.