Exercise 4

2024-04-05

Import Libraries

```
knitr::opts_chunk$set(tidy.opts=list(width.cutoff=60), format='latex', echo=TRUE)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(lubridate)
library(arrow)
```

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

```
library(igraph)
```

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

```
##
##     as_data_frame, groups, union
##
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
##
## The following object is masked from 'package:tidyr':
##
##     crossing
##
## The following object is masked from 'package:tibble':
##
##     as_data_frame
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

Import Dataset

```
data_path = "/Users/yuyichen/Desktop/Winter 2024/ORGB - 672/2024-ona-assignments/app_data_sample.parquet"
applications = arrow::read_parquet(data_path)
set.seed(123)
applications <- sample_n(applications, 200000)
attach(applications)
```

Adding gender to dataset based on surnames library

```
library(gender)

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

examiner_names_gender = examiner_names %>%
  do(results = gender(.$examiner_name_first, method = "ssa")) %>%
  unnest(cols = c(results), keep_empty = TRUE) %>%
  select(
    examiner_name_first = name,
    gender,
    proportion_female
  )

examiner_names_gender
```

```
## # A tibble: 1,698 x 3
##    examiner_name_first gender proportion_female
##    <chr>               <chr>              <dbl>
##  1 AARON               male              0.0082
##  2 ABDEL               male              0
##  3 ABDOU               male              0
##  4 ABDULHAKIM          male              0
##  5 ABDULLAH            male              0
##  6 ABDULLAHI           male              0
##  7 ABIGAIL             female            0.998
##  8 ABIMBOLA            female            0.944
##  9 ABRAHAM             male              0.0031
## 10 ABU                 male              0
## # i 1,688 more rows
```

```
gc()
```

```
##            used  (Mb) gc trigger  (Mb) limit (Mb) max used  (Mb)
## Ncells  1739624  93.0    4846349 258.9         NA  4532337 242.1
## Vcells 13828579 105.6   45581000 347.8      16384 45581000 347.8
```

```
# 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) limit (Mb) max used  (Mb)
## Ncells  1739906  93.0    4846349 258.9         NA  4532337 242.1
## Vcells 17221558 131.4   45581000 347.8      16384 45581000 347.8
```

Add the column for Application processing time

When determining the final decision date for patents, the patent issue date is combined with the abandon date. The final decision date is considered to be whichever of these two dates occurs first, while disregarding any missing data (NAs) in the records. This approach ensures we capture the earliest decisive action in the patent's lifecycle.

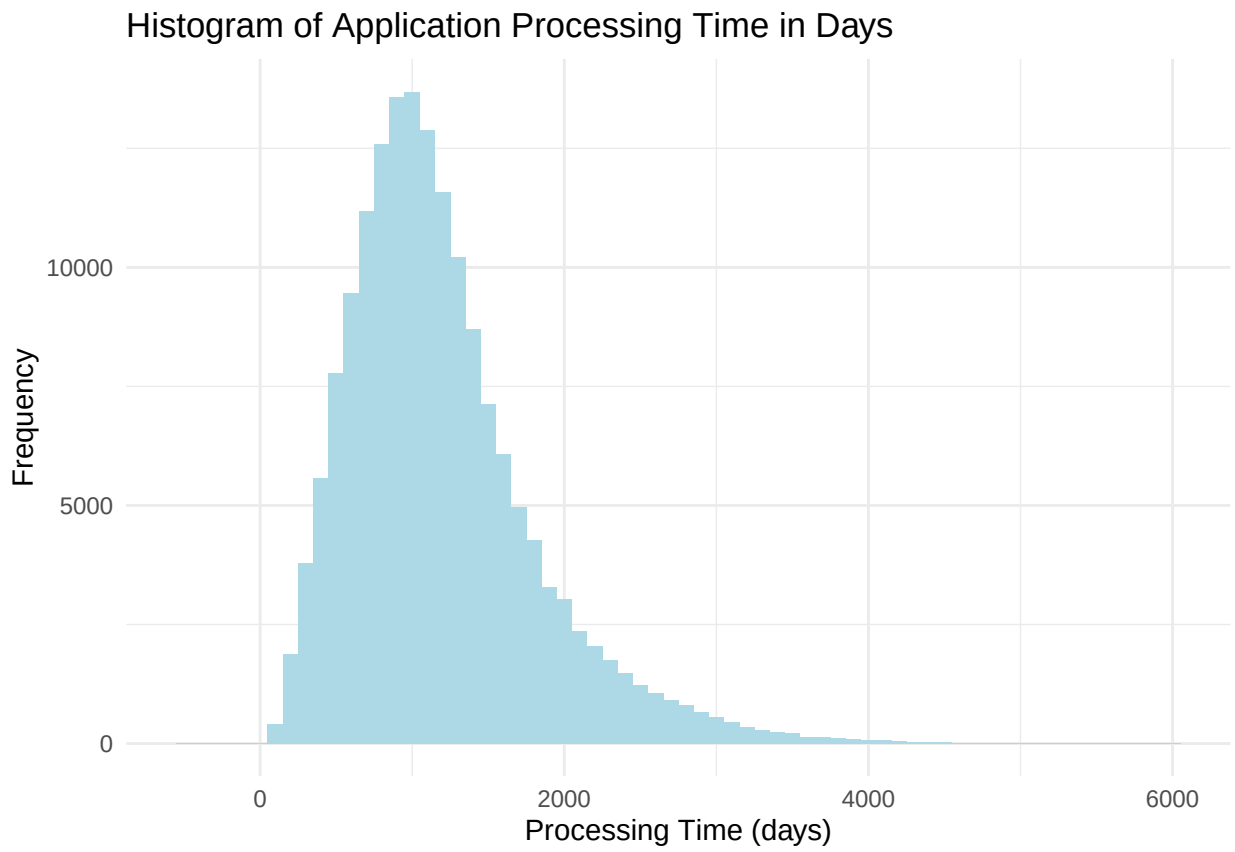
```
# combine patent issue date & abandon_date as the final decision date
# decision_date = the earlier of patent_issue_date and abandon_date, ignoring NAs
applications$decision_date = pmin(applications$patent_issue_date, applications$abandon_date, na.rm = TRU

applications = drop_na(applications,decision_date)
attach(applications)
```

```
## The following objects are masked from applications (pos = 4):
##
##     abandon_date, appl_status_code, appl_status_date,
##     application_number, disposal_type, examiner_art_unit, examiner_id,
##     examiner_name_first, examiner_name_last, examiner_name_middle,
##     filing_date, patent_issue_date, patent_number, tc, uspc_class,
##     uspc_subclass
```

```r
applications = applications %>%
  mutate(
    app_proc_time= as.numeric(ymd(decision_date) - ymd(filing_date)) # days
  )

# Histogram for the application processing days
ggplot(applications, aes(x = app_proc_time)) +
  geom_histogram(binwidth = 100, fill = "lightblue") +
  labs(title = "Histogram of Application Processing Time in Days",
       x = "Processing Time (days)",
       y = "Frequency") +
  theme_minimal()
```




```r
## Negative values in application processing time
# Calculate the total number of applications with negative processing times
neg_decision_date_counts = applications %>%
  filter(app_proc_time < 0) %>%
```

```
  summarise(total_negative_count = n())

# Calculate the total number of applications
total_applications_count = nrow(applications)

# Calculate the ratio of negative processing time counts to total applications
negative_processing_ratio = neg_decision_date_counts$total_negative_count / total_applications_count

# Output the count and the ratio
neg_decision_date_counts
```

```
## # A tibble: 1 x 1
##   total_negative_count
##                  <int>
## 1                    5
```

```
negative_processing_ratio
```

```
## [1] 2.986983e-05
```

This histogram represents the distribution of application processing times, measured in days. The shape of the distribution appears to be right-skewed, meaning that a majority of the applications have a shorter processing time, with a decline in frequency as the processing time increases. The peak of the histogram, where the frequency is highest, is relatively close to the origin, indicating that most applications are processed within a shorter period. There are also fewer applications that take an extremely long time to process, as shown by the long tail extending to the right.

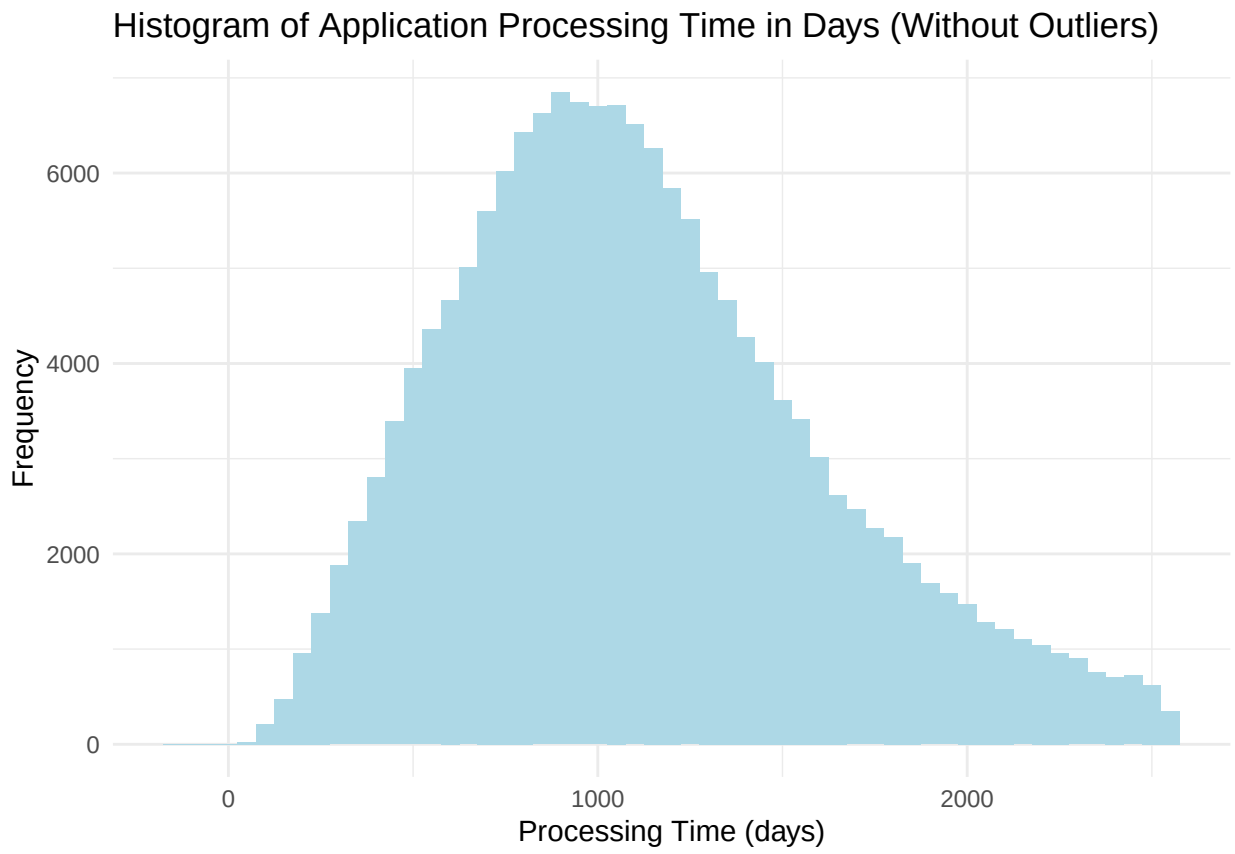
Ploting the histogram of of application processing time in days (without outliers by using IQR methods to filerting out the outlisers

```
#Filter out the outliers to have a clearer distribution graph
# Calculate the IQR
Q1 = quantile(applications$app_proc_time, 0.25)
Q3 = quantile(applications$app_proc_time, 0.75)
IQR = Q3 - Q1

# Define the upper and lower bounds for what is considered an outlier
upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR

# Filter out the outliers
applications_filtered = applications %>%
  filter(app_proc_time >= lower_bound & app_proc_time <= upper_bound)

# Now, create the histogram with the filtered data
ggplot(applications_filtered, aes(x = app_proc_time)) +
  geom_histogram(binwidth = 50, fill = "lightblue") +
  labs(title = "Histogram of Application Processing Time in Days (Without Outliers)",
       x = "Processing Time (days)",
       y = "Frequency") +
  theme_minimal()
```



This histogram represents application processing times in days, with outliers removed for clarity. It appears to follow a normal distribution, showing that most application processes cluster around a central range of days, with fewer occurrences toward the extreme ends of the timeline. This histogram compared to the previous graph that included outliers, provides a more standardized understanding of processing times and suggests that extreme delays are less common than initially perceived. The distribution's peak indicates the most frequent processing time period, and the data spread reveals the variability around that peak.

Create and Calculate the Centrality Column

Please note that since the originial data set is too large, the below anaysis was built based on the sample data of 5000 records.

```
# create the centrality column
# choose closeness
library(igraph)
edges_sample = read_csv("/Users/yuyichen/Desktop/Winter 2024/ORGB - 672/2024-ona-assignments/edges_sampl
```

```
## Rows: 28614 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): ego_examiner_id, alter_examiner_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
edges_sample = drop_na(edges_sample)
edges_sample = select(edges_sample, ego_examiner_id, alter_examiner_id)

g = graph_from_data_frame(edges_sample, directed = FALSE)

# Calculate closeness centrality
closeness_centrality = closeness(g)
centrality_df = data.frame(examiner_id = V(g)$name, closeness_centrality = closeness_centrality)
applications = merge(applications, centrality_df, by.x = "examiner_id", by.y = "examiner_id", all.x = TR

# many examiners do not have centrality
sum(is.na(applications$closeness_centrality))
```

```
## [1] 61892
```

```r
# filter out the NAs on centrality measure
applications = drop_na(applications, closeness_centrality)
# Use the sample due to the computing complexity
set.seed(123)
applications = sample_n(applications, 50000)
attach(applications)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     closeness_centrality
##
## The following objects are masked from applications (pos = 3):
##
##     abandon_date, appl_status_code, appl_status_date,
##     application_number, decision_date, disposal_type,
##     examiner_art_unit, examiner_id, examiner_name_first,
##     examiner_name_last, examiner_name_middle, filing_date, gender,
##     patent_issue_date, patent_number, tc, uspc_class, uspc_subclass
##
## The following objects are masked from applications (pos = 5):
##
##     abandon_date, appl_status_code, appl_status_date,
##     application_number, disposal_type, examiner_art_unit, examiner_id,
##     examiner_name_first, examiner_name_last, examiner_name_middle,
##     filing_date, patent_issue_date, patent_number, tc, uspc_class,
##     uspc_subclass
```

Linear Model built between centrality and application process time

Features selected: Column “examiner_art_unit”,“uspc_class”,“disposal_type”,“appl_status_code”, “tc”, “gender”, and “closeness_centrality” were selected to perform linear relationship

- Examiner Art Unit & TC: These reflect the technological specialization and organization structure within the USPTO, which can significantly affect processing times due to differences in workload, expertise, and procedural nuances across technological areas.

- USPC Class: It indicates the technological category of the patent, which is essential because certain classes might be more complex or contested, leading to longer processing times.
- Disposal Type & Appl Status Code: These outcomes and statuses provide insights into the end-points of the patent examination process, helping to understand how centrality within the USPTO network might correlate with the efficiency or direction of processing.
- Gender: Including gender aims to investigate if network positions and interactions within the USPTO might vary by gender, potentially influencing processing times due to differences in networking or collaboration patterns.
- Closeness Centrality: Directly measures an entity's centrality in the network, crucial for examining how being more centrally positioned (indicating easier access to information or resources) might lead to more efficient processing times.

```
# Column "examiner_art_unit","uspc_class","disposal_type","appl_status_code", "tc", "gender", and "clos

# Subset the dataframe to include only the specified columns and y
selected_columns <- c("examiner_art_unit", "uspc_class", "disposal_type","appl_status_code", "tc", "app

applications_subset <- applications %>%
  select(all_of(selected_columns)) %>%
  drop_na()

# Convert categorical variables to dummy variables
categorical_columns <- c("gender", "examiner_art_unit", "uspc_class", "disposal_type","appl_status_code
applications_subset_matrix <- applications_subset %>%
  mutate(across(c(examiner_art_unit, uspc_class, disposal_type,appl_status_code, tc), factor)) %>%
  model.matrix(~ . - app_proc_time - 1, data = .)
applications_feature = as.data.frame(applications_subset_matrix)

y <- applications_subset$app_proc_time

set.seed(123)
lm_model <- lm(y~., data = applications_feature)
options(max.print = 10000)
summary(lm_model)
```

Findings from the above model:

- Gender Impact: The significant negative coefficient for gendermale suggests that applications associated with male inventors are processed approximately 17.5 days faster than those associated with female inventors, after controlling for other factors. This finding indicates a gender disparity in processing times, highlighting a potential area for further investigation into systemic biases within the patent processing framework.
- Influence of Closeness Centrality: The lack of statistical significance for the closeness_centrality coefficient suggests that an applicant's or inventor's position within the USPTO's network—as measured by closeness centrality—does not significantly affect the speed of patent processing.
- Overall Model Effectiveness: The model demonstrates a moderate explanatory power with an R-squared value of 0.179, indicating that the selected variables capture some, but not all, of the variance in patent processing times.

Implications for the USPTO:

The finding of a gender difference in processing times raises important questions about the equity of the patent examination process, suggesting the need for further scrutiny into how and why these disparities exist.

The non-significant impact of closeness centrality on processing times suggests that the efficiency of patent processing at the USPTO may not be directly influenced by an individual's network position. This indicates that other factors, perhaps related to the quality of the applications, the complexity of the technology, or institutional processes, are more critical in determining how quickly patents are processed.

The moderate explanatory power of the model underscores the complexity of patent processing times, encouraging further research to explore additional variables and their interactions that could shed light on the intricacies of the patent examination process.

Building Linear Model to Estimate the Relationship between Centrality and Application Process Time

```
# set the formula
formula_interaction <- as.formula("y ~ . + gendermale*closeness_centrality")
set.seed(123)
lm_model_interaction <- lm(formula_interaction, data = applications_feature)
options(max.print = 10000)
summary(lm_model_interaction)
```

Findings of the Model with the interation:

- Gender: The coefficient for gendermale is significantly negative (-16.3629) with a p-value of 0.015643, indicating that patents associated with male inventors are processed faster by approximately 16.4 days compared to those associated with female inventors. This difference underscores a gender-based disparity in processing times, signaling potential biases or structural differences in the handling of patent applications based on the gender of the inventor.
- Closeness Centrality: There is no statistically significant effect of an inventor's centrality within the USPTO network on processing times, indicating that an inventor's network position does not markedly influence how quickly their patent is processed.

Interaction between Gender and Closeness Centrality: The interaction between gender and centrality suggests a nuanced effect where being more centrally connected might reduce the processing time advantage for male inventors, though this effect is not statistically significant.

Model Performance: The model explains a modest portion of the variance in processing times and confirms significant predictors, suggesting that gender plays a role in processing times but network centrality does not.

Implications for the USPTO:

- Gender Bias Mitigation: The significant negative effect of male gender on processing times highlights the need for the USPTO to address potential gender biases. Implementing bias training and gender-neutral policies could help mitigate these disparities.

- Network Centrality's Limited Role: The absence of a significant effect from closeness centrality on processing times challenges assumptions about the benefits of network positions, suggesting that the USPTO's processes may prioritize merit over social connections. This finding indicates the organization's effectiveness in neutralizing potential network-based biases.
- Gender and Network Interplay: The interaction between gender and network centrality, although not significant, hints at the complexity of how social networks and gender may influence processing times. This calls for further research to understand these dynamics better and develop interventions to ensure equitable processing times.
- Policy Implications: These findings emphasize the importance of incorporating social network analysis in the USPTO's policy reviews to ensure fairness and efficiency in patent examinations. Future strategies could focus on enhancing transparency and fostering a culture that values diversity and equity.