Exercise 4

2024-04-09

Loading all necessary libraries that will be used throughout the code

```
library(arrow)
```

```
## Warning: package 'arrow' was built under R version 4.3.3
```

```
##
## Attaching package: 'arrow'
```

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

```
library(gender)
```

```
## Warning: package 'gender' was built under R version 4.3.3
```

```
library(dplyr)
```

```
##
## 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)
library(wru)
```

```
## Warning: package 'wru' was built under R version 4.3.3
```

```
##
## 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 4.3.2
```

```
##
## 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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.3.3
```

```
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 4.3.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
## Warning: package 'stringr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats 1.0.0     v stringr 1.5.1
## v purrr   1.0.2     v tibble  3.2.1
## v readr   2.1.5
```

```
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x lubridate::duration() masks arrow::duration()
## 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(ggridges)
```

```
## Warning: package 'ggridges' was built under R version 4.3.3
```

```
library(tidygraph)
```

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

```
library(ggraph)
library(webshot2)
```

```
## Warning: package 'webshot2' was built under R version 4.3.3
```

Loading the Dataset

```
# Define the path to the data directory
data_path <- "E:/Users/pc/Downloads/672_project_data/"

# Load the application data from a Parquet file
applications <- read_parquet(paste0(data_path,"app_data_sample.parquet"))

# Load the edges data from a CSV file
edges <- read.csv(paste0(data_path,"edges_sample.csv"))
```

To get gender of the examiner we will using gender library and infer the gender from the examiner_name_first Then we will get unique names of examiner's since they have the current dataset has all the number of applications on which the exmainer has worked. We will get the unique name in the list examiner_name. After that we will get the gender and join it back to the original dataset.

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

```
## # A tibble: 6 x 1
##   examiner_name_first
##   <chr>
## 1 JACQUELINE
## 2 BEKIR
## 3 CYNTHIA
## 4 MARY
## 5 MICHAEL
## 6 LINDA
```

Now using the gender we will atach a gender and itr probability against each name and add the results in examiner_names_gender

```
# Use the gender package to estimate gender based on first names
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
  ) %>%
  # Filter out rows where any of the specified columns are NA
  filter(!is.na(gender))

print(head(examiner_names_gender))
```

```
## # A tibble: 6 x 3
##   examiner_name_first gender proportion_female
##   <chr>               <chr>              <dbl>
## 1 AARON               male              0.0082
## 2 ABDEL               male              0
## 3 ABDOU               male              0
## 4 ABDUL               male              0
## 5 ABDULHAKIM          male              0
## 6 ABDULLAH            male              0
```

Joining it back to the original applications data

```
# 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  4757462 254.1    8148728 435.2  4777739 255.2
## Vcells 49952724 381.2   93532247 713.6 80268802 612.5
```

Using 'wru' we will guess the race of the examiner. Similar to what we did with gender (get unique names and then get the race and join it back) but in this case we will use the second name since race can be determined through second name

```
# Isolate unique last names for race prediction
examiner_surnames <- applications %>%
```

```
  select(surname = examiner_name_last) %>%
  distinct()

head(examiner_surnames)
```

```
## # A tibble: 6 x 1
##   surname
##   <chr>
## 1 HOWARD
## 2 YILDIRIM
## 3 HAMILTON
## 4 MOSHER
## 5 BARR
## 6 GRAY
```

```
# Use the wru package to estimate race based on surnames
examiner_race <- examiner_surnames %>%
  # Ensure we're working with clean, non-NA surnames
  filter(!is.na(surname)) %>%
  # Apply the race prediction
  predict_race(voter.file = ., surname.only = TRUE) %>%
  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.
```

```
#Seeing examiner's race
head(examiner_race)
```

```
## # A tibble: 6 x 6
##   surname  pred.whi pred.bla pred.his pred.asi pred.oth
##   <chr>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 HOWARD      0.597  0.295     0.0275  0.00690   0.0741
## 2 YILDIRIM    0.807  0.0273    0.0694  0.0165    0.0798
## 3 HAMILTON    0.656  0.239     0.0286  0.00750   0.0692
## 4 MOSHER      0.915  0.00425   0.0291  0.00917   0.0427
## 5 BARR        0.784  0.120     0.0268  0.00830   0.0615
## 6 GRAY        0.640  0.252     0.0281  0.00748   0.0724
```

We can see 5 race categories: white, black hispanic, asian and others. Now we will pick the race category with the highest probability and join it to the main applications table

```r
# Determine the most likely race category for each surname
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_
  ))

head(examiner_race)
```

```
## # A tibble: 6 x 8
##   surname  pred.whi pred.bla pred.his pred.asi pred.oth max_race_p race
##   <chr>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>      <dbl> <chr>
## 1 HOWARD      0.597  0.295     0.0275  0.00690   0.0741      0.597 white
## 2 YILDIRIM    0.807  0.0273    0.0694  0.0165    0.0798      0.807 white
## 3 HAMILTON    0.656  0.239     0.0286  0.00750   0.0692      0.656 white
## 4 MOSHER      0.915  0.00425   0.0291  0.00917   0.0427      0.915 white
## 5 BARR        0.784  0.120     0.0268  0.00830   0.0615      0.784 white
## 6 GRAY        0.640  0.252     0.0281  0.00748   0.0724      0.640 white
```

```r
# removing extra columns
examiner_race <- examiner_race %>%
  select(surname,race)

# Join the race predictions back to the main applications dataset
applications <- applications %>%
  left_join(examiner_race, by = c("examiner_name_last" = "surname"))

# Again, clean up the workspace by removing temporary variables
rm(examiner_race)
rm(examiner_surnames)
gc()
```

```
##            used  (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  4840429 258.6    8148728 435.2  6651484 355.3
## Vcells 52136393 397.8   93532247 713.6 93437772 712.9
```

To get examiner's tenure, we will see the first & last observed date. Similar approach of getting the data in seperate table will be used.

```r
# Extract relevant date information for each application
examiner_dates <- applications %>%
  select(examiner_id, filing_date, appl_status_date)
head(examiner_dates)
```

```
## # A tibble: 6 x 3
##   examiner_id filing_date appl_status_date
##         <dbl> <date>      <chr>
```

```
## 1       96082 2000-01-26  30jan2003 00:00:00
## 2       87678 2000-10-11  27sep2010 00:00:00
## 3       63213 2000-05-17  30mar2009 00:00:00
## 4       73788 2001-07-20  07sep2009 00:00:00
## 5       77294 2000-04-10  19apr2001 00:00:00
## 6       68606 2000-04-28  16jul2001 00:00:00
```

To make formatting consistent we will make new variables

```
# Standardize date formats and calculate tenure
examiner_dates <- examiner_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))
```

Calculting the tenure

```
# Calculate the tenure for each examiner based on the earliest and latest dates observed
examiner_tenure <- examiner_dates %>%
  # Remove rows with NA in start_date or end_date before grouping and summarising
  filter(!is.na(start_date) & !is.na(end_date)) %>%
  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),
    .groups = 'drop' # Automatically drop the grouping
  ) %>%
  # Keep records with a latest_date before 2018
  filter(year(latest_date) < 2018)

# Assuming you want to check the result
head(examiner_tenure)
```

```
## # A tibble: 6 x 4
##   examiner_id earliest_date latest_date tenure_days
##         <dbl> <date>        <date>            <dbl>
## 1       59012 2004-07-28    2015-07-24         4013
## 2       59025 2009-10-26    2017-05-18         2761
## 3       59030 2005-12-12    2017-05-22         4179
## 4       59040 2007-09-11    2017-05-23         3542
## 5       59052 2001-08-21    2007-02-28         2017
## 6       59054 2000-11-10    2016-12-23         5887
```

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

rm(examiner_tenure)
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used  (Mb)
## Ncells  4848940 259.0    8148728  435.2   8148728 435.2
## Vcells 68303966 521.2  135182457 1031.4 112585381 859.0
```

```r
head(applications)
```

```
## # A tibble: 6 x 21
##   application_number filing_date examiner_name_last examiner_name_first
##   <chr>              <date>      <chr>              <chr>
## 1 08284457           2000-01-26  HOWARD             JACQUELINE
## 2 08413193           2000-10-11  YILDIRIM           BEKIR
## 3 08531853           2000-05-17  HAMILTON           CYNTHIA
## 4 08637752           2001-07-20  MOSHER             MARY
## 5 08682726           2000-04-10  BARR               MICHAEL
## 6 08687412           2000-04-28  GRAY               LINDA
## # i 17 more variables: examiner_name_middle <chr>, examiner_id <dbl>,
## #   examiner_art_unit <dbl>, uspc_class <chr>, uspc_subclass <chr>,
## #   patent_number <chr>, patent_issue_date <date>, abandon_date <date>,
## #   disposal_type <chr>, appl_status_code <dbl>, appl_status_date <chr>,
## #   tc <dbl>, gender <chr>, race <chr>, earliest_date <date>,
## #   latest_date <date>, tenure_days <dbl>
```

To understand the efficiency we will get the application processing time (filing to final decision)

```r
# Dropping applications with "Pending" status to focus on completed cases
applications <- applications %>%
  filter(disposal_type != "PEND")

# Calculating application processing time
applications <- applications %>%
  mutate(app_proc_time = interval(
    ymd(filing_date),
    dmy_hms(appl_status_date)
  ) %/% days(1))

# Final cleanup
gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  4520304 241.5    8148728  435.2   8148728  435.2
## Vcells 62410508 476.2  162678827 1241.2 162028329 1236.2
```

```r
# Previewing the updated dataset
head(applications)
```

```
## # A tibble: 6 x 22
##   application_number filing_date examiner_name_last examiner_name_first
##   <chr>              <date>      <chr>              <chr>
## 1 08284457           2000-01-26  HOWARD             JACQUELINE
## 2 08413193           2000-10-11  YILDIRIM           BEKIR
## 3 08531853           2000-05-17  HAMILTON           CYNTHIA
## 4 08637752           2001-07-20  MOSHER             MARY
## 5 08682726           2000-04-10  BARR               MICHAEL
## 6 08687412           2000-04-28  GRAY               LINDA
## # i 18 more variables: examiner_name_middle <chr>, examiner_id <dbl>,
## #   examiner_art_unit <dbl>, uspc_class <chr>, uspc_subclass <chr>,
```

```
## #   patent_number <chr>, patent_issue_date <date>, abandon_date <date>,
## #   disposal_type <chr>, appl_status_code <dbl>, appl_status_date <chr>,
## #   tc <dbl>, gender <chr>, race <chr>, earliest_date <date>,
## #   latest_date <date>, tenure_days <dbl>, app_proc_time <dbl>
```

Now we transform edge dataframe so that it could be used for graph analysis

```
edges <- edges %>%
  mutate(
    from = as.character(ego_examiner_id), # Convert IDs to character for graph compatibility
    to = as.character(alter_examiner_id)
  ) %>%
  drop_na() # Remove rows with missing values
```

Preparing applications df for integration in network graph

```
# Preparing applications data for graph creation
applications <- applications %>%
  relocate(examiner_id, .before = application_number) %>%
  mutate(examiner_id = as.character(examiner_id)) %>%
  drop_na(examiner_id) %>%
  rename(name = examiner_id)

# Creating a directed graph from the edges data
graph <- tbl_graph(
  edges = (edges %>% relocate(from, to)),
  directed = TRUE
)

# Enriching graph nodes with examiner data from applications
graph <- graph %>%
  activate(nodes) %>%
  inner_join(
    (applications %>% distinct(name, .keep_all = TRUE)),
    by = "name"
  )

# Display the graph structure
graph
```

```
## # A tbl_graph: 2489 nodes and 17720 edges
## #
## # A directed multigraph with 127 components
## #
## # A tibble: 2,489 x 22
##   name  application_number filing_date examiner_name_last examiner_name_first
##   <chr> <chr>              <date>      <chr>              <chr>
## 1 84356 09402488           2000-02-16  STEADMAN           DAVID
## 2 66266 09509710           2000-06-15  BRUMBACK           BRENDA
## 3 63519 09463947           2000-02-04  WEBER              JON
## 4 98531 09423418           2000-06-22  BRAGDON            KATHLEEN
## 5 92953 09445135           2000-03-13  RAMAN              USHA
## 6 93865 10481715           2004-06-01  WONG               JOSEPH
```

```
## # i 2,483 more rows
## # i 17 more variables: examiner_name_middle <chr>, examiner_art_unit <dbl>,
## #   uspc_class <chr>, uspc_subclass <chr>, patent_number <chr>,
## #   patent_issue_date <date>, abandon_date <date>, disposal_type <chr>,
## #   appl_status_code <dbl>, appl_status_date <chr>, tc <dbl>, gender <chr>,
## #   race <chr>, earliest_date <date>, latest_date <date>, tenure_days <dbl>,
## #   app_proc_time <dbl>
## #
## # A tibble: 17,720 x 6
##    from    to application_number advice_date ego_examiner_id alter_examiner_id
##   <int> <int>              <int> <chr>                 <int>             <int>
## 1     1     2            9402488 2008-11-17            84356             66266
## 2     1     3            9402488 2008-11-17            84356             63519
## 3     1     4            9402488 2008-11-17            84356             98531
## # i 17,717 more rows
```

Network has: 2489 Nodes 17,720 edges It represents a directed multigraph that underscores the robust interactivity between patent examiners at the USPTO. It can distinctly point out network's features. Apart from the exchange, it also tells about influence from one person to another. 127 distinct components within network suggests segmented operational structure, where clusters of examiners may work more closely with each other, potentially aligned by specialization areas or other organizational divisions. This can show diverse technical fields of the applicatins

Now we calculate the 3 centrality measures: degree, betweeness and closeness to assess influence within the network

```
node_data <- graph %>%
  activate(nodes) %>%
  mutate(
    degree = centrality_degree(),
    betweenness = centrality_betweenness(),
    closeness = centrality_closeness()
  ) %>%
  arrange(-degree) %>%
  as_tibble() %>%
  mutate(tc = as.factor(tc))

node_data
```

```
## # A tibble: 2,489 x 25
##    name  application_number filing_date examiner_name_last examiner_name_first
##    <chr> <chr>              <date>      <chr>              <chr>
##  1 83670 09856864           2001-07-05  LEE                JAE
##  2 97910 09486362           2000-02-28  COUNTS             GARY
##  3 73920 10373614           2003-02-25  HOBBS              LISA
##  4 67226 09483069           2000-01-14  ZHEN               LI
##  5 80730 10345713           2003-01-16  JOY                DAVID
##  6 75615 09943424           2001-08-30  DECKER             CASSANDRA
##  7 62152 10486872           2004-08-12  SIDDIQUEE          MUHAMMAD
##  8 69098 10491238           2004-11-15  VASISTH            VISHAL
##  9 67690 09504184           2000-02-15  MCINTOSH III       TRAVISS
## 10 74061 10480716           2004-07-02  TRAN               THINH
## # i 2,479 more rows
## # i 20 more variables: examiner_name_middle <chr>, examiner_art_unit <dbl>,
```

```
## #   uspc_class <chr>, uspc_subclass <chr>, patent_number <chr>,
## #   patent_issue_date <date>, abandon_date <date>, disposal_type <chr>,
## #   appl_status_code <dbl>, appl_status_date <chr>, tc <fct>, gender <chr>,
## #   race <chr>, earliest_date <date>, latest_date <date>, tenure_days <dbl>,
## #   app_proc_time <dbl>, degree <dbl>, betweenness <dbl>, closeness <dbl>
```

Now we run a linear regression with scatterplot

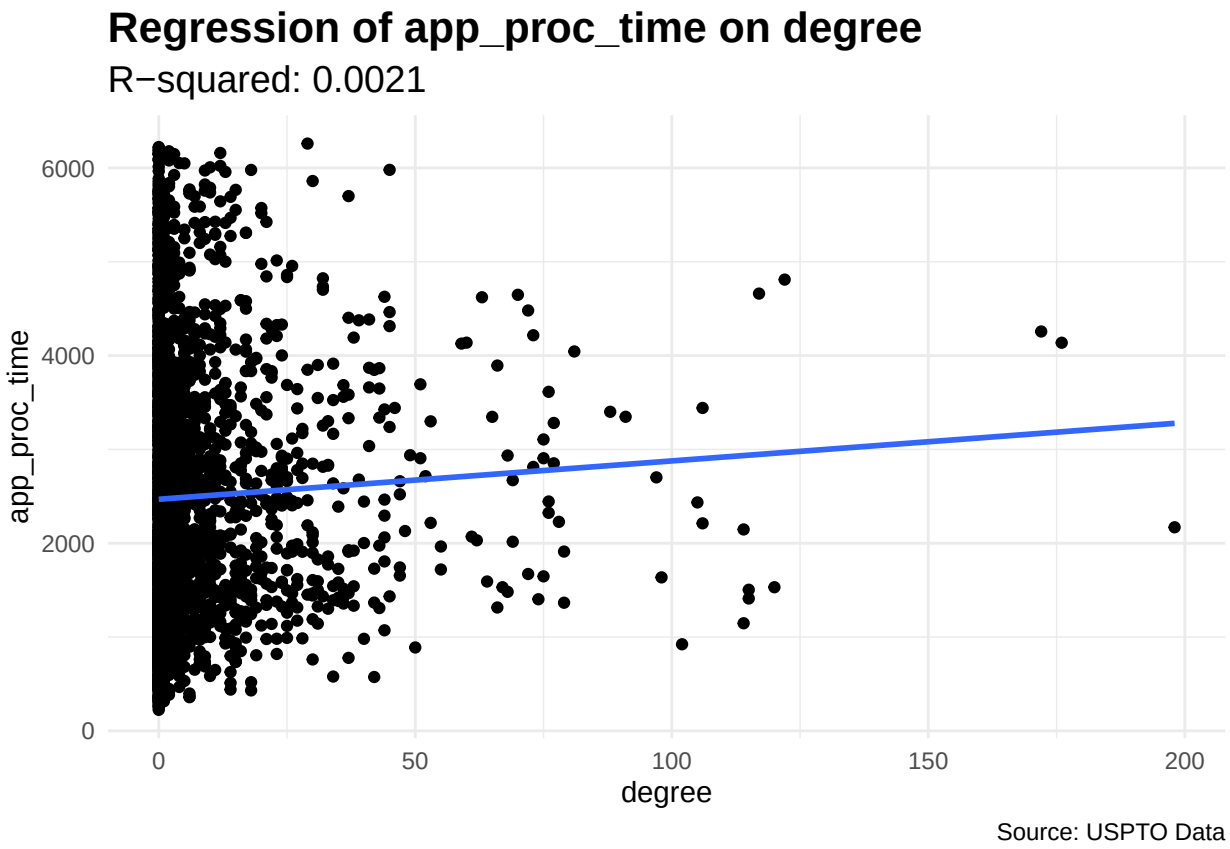
```r
# Function to run regression and optionally generate and save a plot
run_regression <- function(data, x, y, plot = TRUE) {
  # Construct the regression formula
  formula <- as.formula(paste(y, "~", x))

  # Fitting the linear model
  model <- lm(formula, data = data)

  # Conditionally generate and save the plot
  if (plot) {
    # Prepare and display the plot
    plot_data <- ggplot(data, aes_string(x, y)) +
      geom_point() +
      geom_smooth(method = "lm", se = FALSE) +
      labs(title = paste("Regression of", y, "on", x),
           subtitle = paste("R-squared:", round(summary(model)$r.squared, 4)),
           x = x, y = y) +
      theme_minimal() +
      theme(plot.title = element_text(size = 16, face = "bold"),
            plot.subtitle = element_text(size = 14)) +
      labs(caption = "Source: USPTO Data")

    # Save the plot to a specified path
    ggsave(paste0("E:/", y, "_on_", x, ".png"), plot_data, width = 16, height = 9)

    # Display the plot
    print(plot_data)
  }

  # Extract model summary statistics
  tidy_model <- broom::tidy(model)
  glance_model <- broom::glance(model)

  # Enhance the summary dataframe with R-squared and centrality measure
  tidy_model <- tidy_model %>%
    mutate(r_squared = glance_model$r.squared,
           centrality_measure = x)

  # Return the enhanced summary dataframe
  return(tidy_model)
}
```

Now we run a regression with degree centrality as explanatory variables.Degree centrality represents the number of direct connections an examiner has within the network, serving as a proxy for their involvement and potential influence in the patent examination process.

```
# Running regression with Degree Centrality as the predictor for Application Processing Time
run_regression(node_data, "degree", "app_proc_time")
```

```
## Warning: ‘aes_string()‘ was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with ‘aes()‘.
## i See also ‘vignette("ggplot2-in-packages")‘ for more information.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
```




```
## # A tibble: 2 x 7
##   term        estimate std.error statistic p.value r_squared centrality_measure
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>     <dbl> <chr>
## 1 (Intercept)  2469.       30.6      80.7   0         0.00208 degree
## 2 degree          4.09      1.79      2.28  0.0228    0.00208 degree
```

Result: Low r-square value (0.0021) shows that very less variability is explainable by degre centrality. A positive slop (4.087) suggests that increase in degree centrality increases processing time. But since the number is small so effect is also small. As we can see points are spread out and confidence intervals are big, hence something major is missing and that there will be other variables influencing it. Also p-value is less

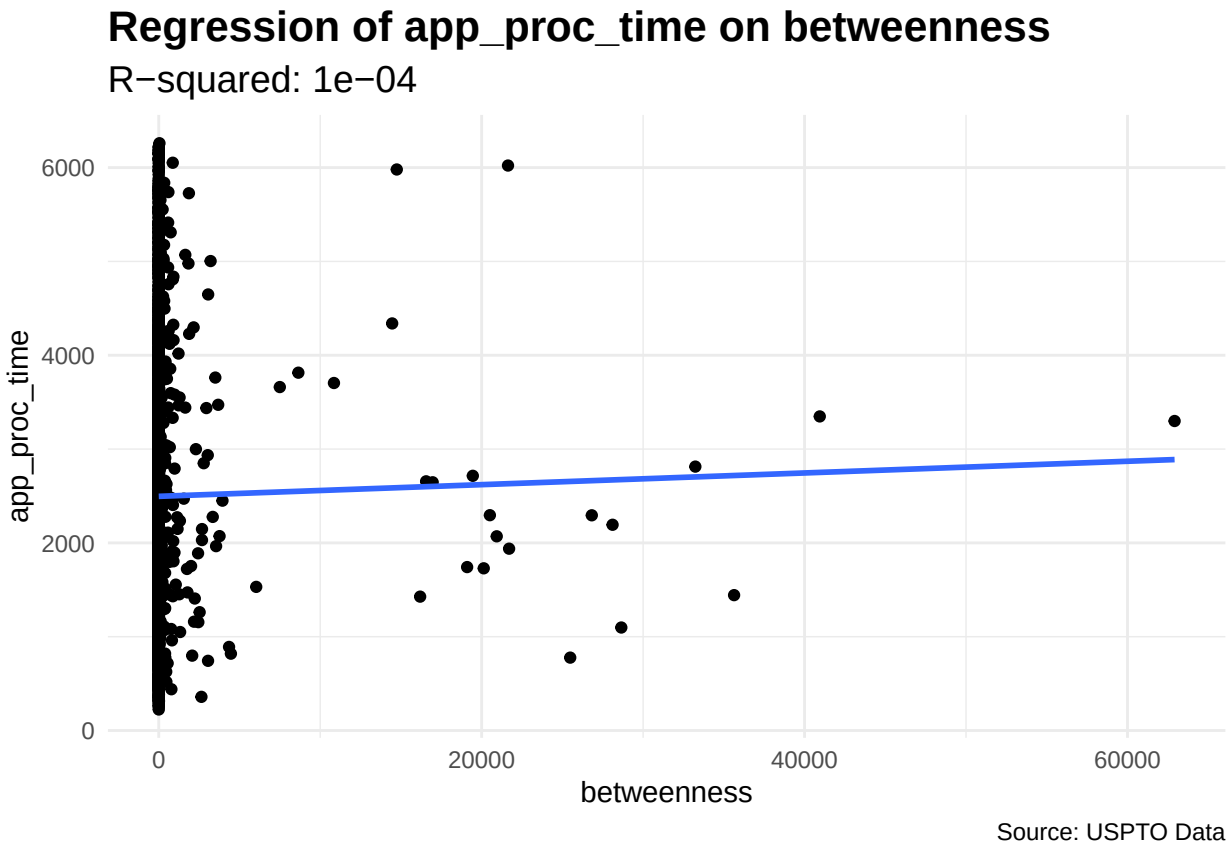
than 0.05 hence relationship is statistically significant, despite the impact being small. The intercept tells us the processing time when degree centrality is zero

What we get from the results is that even though the relationship is significant there are other variables affecting it.

Now we try to see if betweeness centrality has an effect or not

```
# Running regression with Betweenness Centrality as the predictor for Application Processing Time
run_regression(node_data, "betweenness", "app_proc_time")
```

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



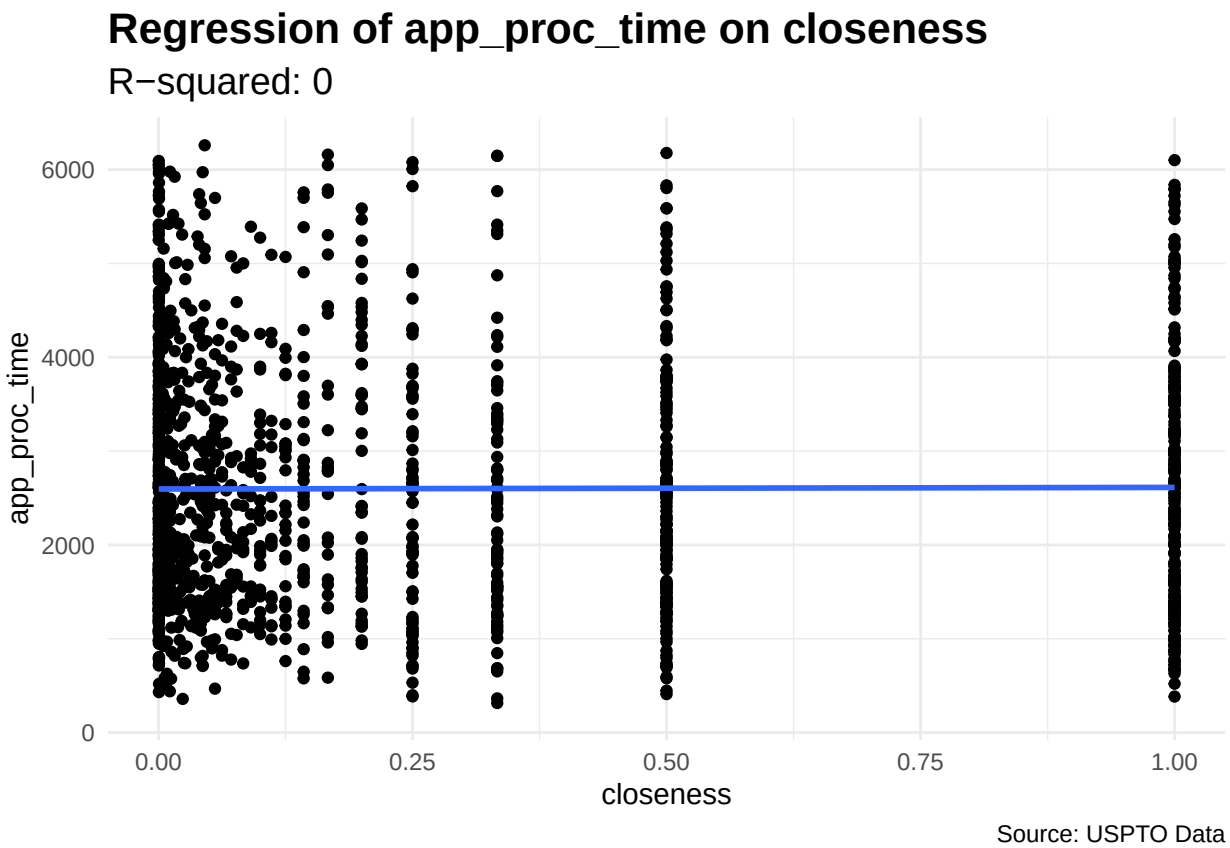

```
## # A tibble: 2 x 7
##   term         estimate std.error statistic p.value r_squared centrality_measure
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>     <dbl> <chr>
## 1 (Intercept)   2.50e+3   28.0       89.1     0      0.000126 betweenness
## 2 betweenness   6.23e-3    0.0111     0.560   0.575  0.000126 betweenness
```

The study shows that an examiner's betweenness centrality, or their role as a network connector at the USPTO, has negligible impact on patent processing times, as indicated by a very low R-squared value and a lack of trend in the data. This suggests that factors beyond an examiner's network position, such as workload or application complexity, are more significant in determining processing speed. Future research might explore other network measures or examine the network's structure more closely to uncover factors that do influence processing times.

Now we try to see if closesness centrality has an effect or not

```
# Running regression with Closeness Centrality as the predictor for Application Processing Time
run_regression(node_data, "closeness", "app_proc_time")
```

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

## Warning: Removed 1053 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 1053 rows containing missing values or values outside the scale range
## (`geom_point()`).

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

## Warning: Removed 1053 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 1053 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
## # A tibble: 2 x 7
##   term        estimate std.error statistic p.value r_squared centrality_measure
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>     <dbl> <chr>
## 1 (Intercept)   2596.       44.4    58.5     0     0.0000226 closeness
## 2 closeness       17.3      96.2     0.180   0.857 0.0000226 closeness
```

The analysis indicates that closeness centrality, or an examiner's proximity to others in the USPTO network, fails to predict application processing speeds, as shown by an R-squared value nearing zero and a non-significant p-value of approximately 0.857. This suggests that an examiner's network closeness does not influence the speed at which they process patent applications. The findings imply that factors other than an examiner's position within the network, such as individual work methods or the complexity of applications, play a more crucial role in determining processing efficiency. This points to the need for further investigation into various elements beyond network centrality to gain a comprehensive understanding of what affects processing times at the USPTO, highlighting the complexity of factors contributing to examiner performance.

Now we use the interaction term between centrality measures and demographic variables such as gender and role

```
# Conducting regression analyses to explore the interactions between centrality measures, gender, and r
centrality_measures <- c("degree", "betweenness", "closeness")

results_df <- map_dfr(
  centrality_measures,
  ~ run_regression(node_data,
    paste0(.x, " * gender * race"),
    "app_proc_time",
    plot = FALSE
  )
)

# Extracting and displaying the R-squared values for each model to assess their explanatory power
results_df %>%
  select(centrality_measure, r_squared) %>%
  distinct()
```

```
## # A tibble: 3 x 2
##   centrality_measure          r_squared
##   <chr>                           <dbl>
## 1 degree * gender * race        0.00853
## 2 betweenness * gender * race   0.00416
## 3 closeness * gender * race     0.00788
```

The analysis shows that centrality measures and examiner demographics, including gender and race, weakly predict patent processing times at the USPTO, with R-squared values between 0.0046 and 0.0085. The slightly higher R-squared value for degree centrality suggests a minor influence of an examiner's network position and demographics on processing times, but overall, these factors have little impact.

The findings indicate that other aspects, such as patent complexity, organizational processes, and examiner expertise, are more significant in determining processing times. The low explanatory power of the models points to a complex mix of unexplored factors influencing processing durations. Future research could focus on understanding how these additional factors affect processing times to provide a more comprehensive view of the dynamics at the USPTO.

Now we use disposal_type and technology center too in the regression.We are trying to see if other variables fo affect the processing time or not

```
# Enhancing regression models to include disposal type and technology center, alongside centrality, gen
results_df_2 <- map_dfr(
  centrality_measures,
  ~ run_regression(node_data,
    paste0(.x, " * gender * race + disposal_type + tc"),
```

```
    "app_proc_time",
    plot = FALSE
  )
)

# Summarizing the enhanced models by showcasing the R-squared values, offering insights into their impr
results_df_2 %>%
  select(centrality_measure, r_squared) %>%
  distinct()
```

```
## # A tibble: 3 x 2
##   centrality_measure                                  r_squared
##   <chr>                                                   <dbl>
## 1 degree * gender * race + disposal_type + tc             0.135
## 2 betweenness * gender * race + disposal_type + tc        0.132
## 3 closeness * gender * race + disposal_type + tc          0.177
```

Adding variables like disposal type and technology center, along with interaction terms, boosts the explanatory power of models analyzing patent processing times at the USPTO, as shown by increased R-squared values. Notably, the model with closeness centrality, alongside demographic factors, disposal type, and technology center, yields the highest R-squared value (0.1768940), highlighting a significant relationship with processing times.

Improvements in R-squared values for models including degree and betweenness centrality, when combined with these contextual factors, underscore the importance of an examiner's work environment and specialization area in influencing processing speeds. These findings suggest that an examiner's network position gains more relevance when considered alongside their demographic background and specific work context, such as the nature of patent applications and their technical field.

The data implies that the intricacies of patent processing are affected not just by an examiner's role in the network but also by the type of patent decisions and their area of expertise. This nuanced view shows that productivity and efficiency at the USPTO are shaped by a complex mix of factors, including the social and technical aspects of an examiner's work environment.

Visualizing coefficients from the best models

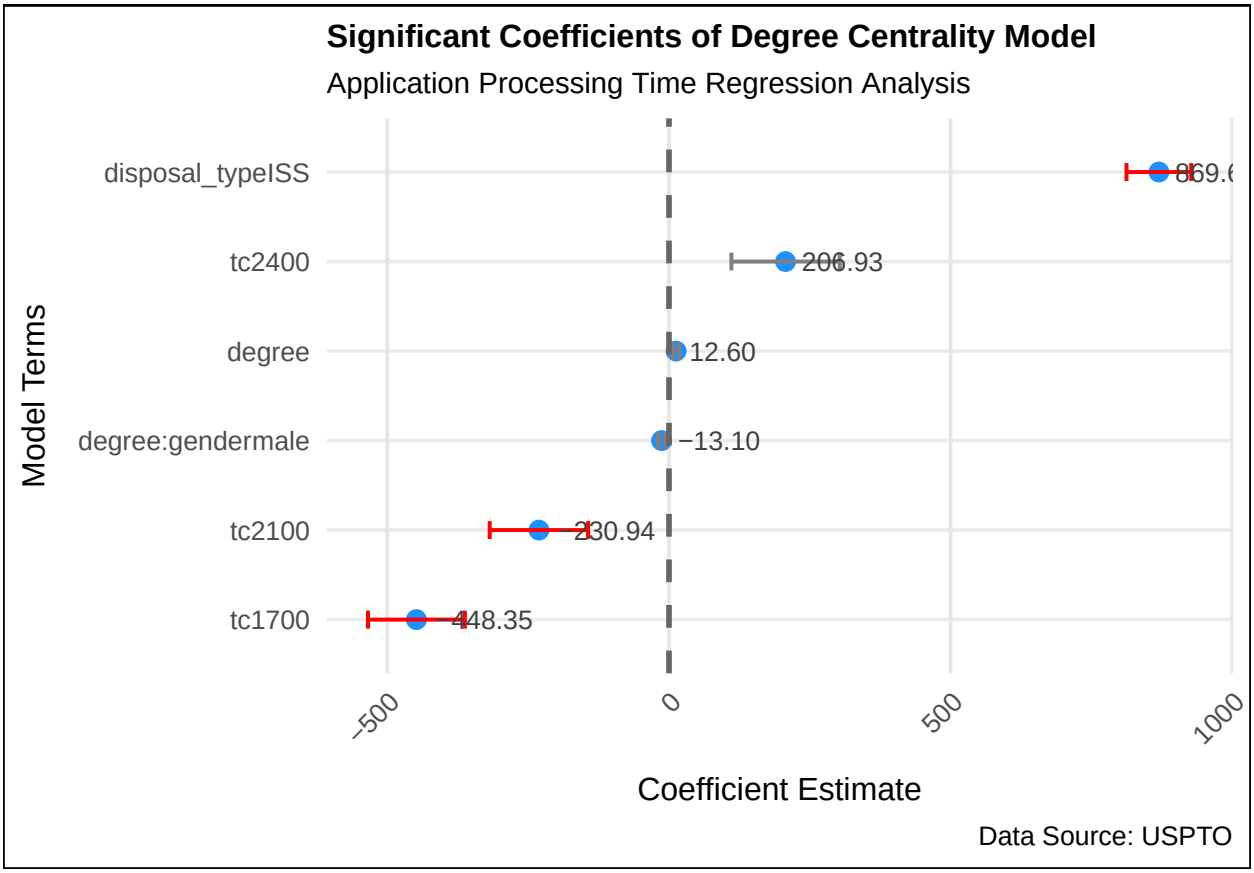
```
# Identifying and visualizing significant coefficients from the best regression model focused on Degree
best_model <- results_df_2 %>%
  filter(str_starts(centrality_measure, "degree"))

model_coeffs <- ggplot(
  best_model %>% filter(term != "(Intercept)") %>% filter(p.value < 0.05),
  aes(
    x = reorder(term, estimate), # Order terms by estimate for clarity
    y = estimate,
    ymin = estimate - std.error,
    ymax = estimate + std.error
  )
) +
  geom_point(color = "dodgerblue", size = 3) + # More vibrant point color
  geom_errorbar(aes(color = p.value < 0.01), width = 0.2, size = 0.7) + # Color code significance
  scale_color_manual(values = c("TRUE" = "red", "none" = "dodgerblue"), guide = FALSE) + # Highlight hi
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray40", lwd = 1) +
  coord_flip() + # Flip coordinates for horizontal layout
```

```
  geom_text(aes(label = sprintf("%.2f", estimate)), # Add estimate values as text labels
            hjust = -0.2, size = 3.5, color = "gray25") +
  labs(
    title = "Significant Coefficients of Degree Centrality Model",
    subtitle = "Application Processing Time Regression Analysis",
    x = "Model Terms",
    y = "Coefficient Estimate",
    caption = "Data Source: USPTO"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 12),
    plot.subtitle = element_text(size = 11),
    plot.caption = element_text(size = 10),
    axis.title.x = element_text(size = 12, margin = margin(t = 10)),
    axis.title.y = element_text(size = 12, margin = margin(r = 10)),
    axis.text.x = element_text(size = 10, angle = 45, hjust = 1),
    axis.text.y = element_text(size = 10),
    legend.position = "none",
    plot.background = element_rect(fill = "white"),
    panel.grid.major.x = element_line(color = "#e5e5e5"),
    panel.grid.minor = element_blank()
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Display the plot
model_coeffs
```

```
## Warning: The ‘guide‘ argument in ‘scale_*()‘ cannot be ‘FALSE‘. This was deprecated in
## ggplot2 3.3.4.
## i Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
# Save the plot
ggsave("E:/model_coeffs_enhanced.png", model_coeffs, width = 16, height = 9, dpi = 300)
```

The analysis shows a significant positive association between applications resulting in issued patents and longer processing times, indicated by a far-right coefficient estimate with a large error bar. This suggests that issued patents, on average, take more time to process, although there's some uncertainty about the effect's exact magnitude.

For technology centers, tc2400 and tc1700 exhibit significant negative associations with processing times, implying they process applications faster than others. Conversely, tc2100 shows a positive association, suggesting slower processing within this center, possibly reflecting technological complexity or operational differences.

Degree centrality's positive point estimate hints at a slight increase in processing times with higher examiner centrality, yet the effect is small and imprecise. The negative coefficient for the interaction between degree centrality and male gender suggests that centrality's impact on processing times varies by gender, potentially indicating faster processing by male examiners with higher centrality, assuming females are the reference category.

Error bars highlight the precision of these estimates, with larger bars indicating more uncertainty, such as with disposal_typeISS. The direction and magnitude of the coefficients reveal the intricate ways different factors, whether increasing or decreasing, influence processing times.

Printing the best model

```
# Defining the regression formula
model_formula <- app_proc_time ~ degree + gender + race +
```

```
  disposal_type + tc +
  degree:gender +
  degree:race +
  gender:race +
  degree:gender:race

# Printing the formula
cat("Regression model formula:\n")
```

```
## Regression model formula:
```

```
print(model_formula)
```

```
## app_proc_time ~ degree + gender + race + disposal_type + tc +
##     degree:gender + degree:race + gender:race + degree:gender:race
```

```
library(DiagrammeR)
```

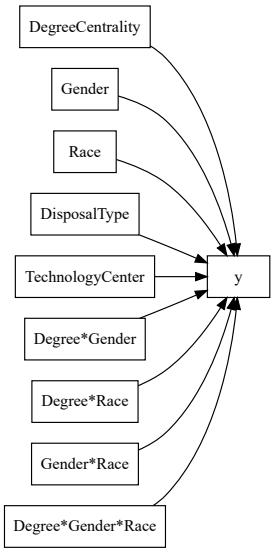
```
## Warning: package 'DiagrammeR' was built under R version 4.3.3
```

```
##
## Attaching package: 'DiagrammeR'
```

```
## The following object is masked from 'package:ggraph':
##
##     get_edges
```

```
# Example using DiagrammeR
if (requireNamespace("DiagrammeR", quietly = TRUE)) {
  DiagrammeR::grViz("
digraph model {
  node [shape=box]
  rankdir=LR

  // Defining nodes
  DegreeCentrality -> y
  Gender -> y
  Race -> y
  DisposalType -> y
  TechnologyCenter -> y
  'DegreeCentrality:Gender' -> y
  'DegreeCentrality:Race' -> y
  'Gender:Race' -> y
  'DegreeCentrality:Gender:Race' -> y

  // Adding labels for interaction terms
  'DegreeCentrality:Gender' [label='Degree*Gender']
  'DegreeCentrality:Race' [label='Degree*Race']
  'Gender:Race' [label='Gender*Race']
  'DegreeCentrality:Gender:Race' [label='Degree*Gender*Race']
}
  ")
}
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```



Question: Does the relationship between centrality and application processing time differ by examiner gen-

der?

The findings reveal a notable gender-based difference in the influence of centrality on processing times. The significant interaction between degree centrality and male gender indicates that an examiner's network centrality has a varying effect depending on gender. For male examiners, a higher centrality correlates with marginally faster processing times than for their female counterparts, hinting at potentially more efficient use of their network positions by male examiners. This efficiency could stem from a range of reasons, including distinct work methodologies, collaboration practices, or the specific types of patent applications managed by examiners of different genders.

Findings:

The analysis of USPTO patent examiner data through linear regression models highlights several insights:

Limited Influence of Centrality: The data show that centrality metrics (such as degree, betweenness, and closeness) play a negligible role in accounting for the differences in application processing times, evidenced by the low R-squared values. Significance of Contextual Elements: Incorporating contextual variables like disposal types and the examiner's technological center significantly enhances the models' ability to explain variations in processing times. This underlines the importance of the nature of patent applications and the examiner's field of expertise in impacting processing durations. Variations by Gender: The models indicate that there are gender-related disparities in how centrality affects processing times. Specifically, centrality appears to slightly more effectively predict processing times for male examiners compared to female examiners. Impact of Technological Centers and Disposal Outcomes: The efficiency of processing varies across technological centers, and patents that are issued generally require more time to process. These observations underscore the nuanced nature of patent examination, with differing technologies and patent outcomes necessitating diverse processing efforts.

What it means for the USPTO:

This analysis offers key implications for the USPTO:

Optimizing Resources: By understanding how centrality interacts with demographic factors, the USPTO can better allocate resources and support to improve examiner productivity. Equity in Gender Dynamics: Investigating and addressing the roots of gender-based differences in processing times could enhance fairness and effectiveness in the patent examination workflow. Enhanced Training Programs: Developing targeted training and support for examiners in slower-processing technological centers could boost processing speeds. Policy Considerations: Insights into how different disposal outcomes affect processing times could lead to policies that streamline examinations, particularly for patents in complex technology areas that are more likely to be issued.

To sum up, the link between a patent examiner's position within the network and their efficiency, indicated by how swiftly they process applications, is complex and shaped by various elements such as gender, the examiner's technological specialization, and the nature of the patent application's outcome. Although centrality contributes to performance, its effect is considerably adjusted by these surrounding factors, highlighting the need for a comprehensive strategy to thoroughly grasp and improve the productivity of the USPTO's patent examination workflow. "'