Exercise 2 starter

`{r setup, include=FALSE} install.packages("tidyverse") library(tidyverse) install.packages("lubridate") library(lubridate) install.packages("arrow") library(arrow)`

Load data

Load the following data: + applications from `app_data_sample.parquet` + edges from `edges_sample.csv`

```{r load-data} # change to your own path! applications <- read_feather("/Users/sheidamajidi/Desktop/Winter2024/COURSES/ORGB671/Project Data/app_data_starter.feather") #applications <- read_feather(paste0(data_path,"app_data_starter.feather"))

applications

````
Get gender for examiners

We'll get gender based on the first name of the examiner, which is recorded in the field `examiner_name_first

Note that there are over 2 million records in the applications table -- that's because there are many records

```{r gender-1}
library(gender)
#install_genderdata_package() # only run this line the first time you use the package, to get data for it

# get a list of first names without repetitions1
examiner_names <- applications %>%
  distinct(examiner_name_first)

examiner_names
````

Now let's use function `gender()` as shown in the example for the package to attach a gender and probability to each name and put the results into the table `examiner_names_gender`

```{r gender-2} # get a table of names and gender examiner_names_gender <- examiner_names %>% do(results = gender(.$examiner_name_first, method = "ssa")) %>% unnest(cols = c(results), keep_empty = TRUE) %>% select( examiner_name_first = name, gender, proportion_female )

examiner_names_gender

````
Finally, let's join that table back to our original applications data and discard the temporary tables we hav

```{r gender-3}
# 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()
````

Guess the examiner's race

We'll now use package `wru` to estimate likely race of an examiner. Just like with gender, we'll get a list of unique names first, only now we are using surnames.

```{r race-1} library(wru)

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

examiner_surnames

````
We'll follow the instructions for the package outlined here <https://github.com/kosukeimai/wru>.

```{r race-2}
examiner_race <- predict_race(voter.file = examiner_surnames, surname.only = T) %>%
  as_tibble()

examiner_race
````

As you can see, we get probabilities across five broad US Census categories: white, black, Hispanic, Asian and other. (Some of you may correctly point out that Hispanic is not a race category in the US Census, but these are the limitations of this package.)

Our final step here is to pick the race category that has the highest probability for each last name and then join the table back to the main applications table. See this example for comparing values across columns: https://www.tidyverse.org/blog/2020/04/dplyr-1-0-0-rowwise/. And this one for `case_when()` function: https://dplyr.tidyverse.org/reference/case_when.html.

```{r race-3} 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_ ))

examiner_race

````
Let's join the data back to the applications table.

```{r race-4}
# 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()
````

Examiner's tenure

To figure out the timespan for which we observe each examiner in the applications data, let's find the first and the last observed date for each examiner. We'll first get examiner IDs and application dates in a separate table, for ease of manipulation. We'll keep examiner ID (the field `examiner_id`), and earliest and latest dates for each application (`filing_date` and `appl_status_date` respectively). We'll use functions in package `lubridate` to work with date and time values.

```{r tenure-1} library(lubridate) # to work with dates

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

examiner_dates

````
The dates look inconsistent in terms of formatting. Let's make them consistent. We'll create new variables `s

```{r tenure-2}
examiner_dates <- examiner_dates %>%
  mutate(start_date = ymd(filing_date), end_date = as_date(dmy_hms(appl_status_date)))
````

Let's now identify the earliest and the latest date for each examiner and calculate the difference in days, which is their tenure in the organization.

````
```{r tenure-3} 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)
````

examiner_dates

````
Joining back to the applications data.

```{r tenure-4}
applications <- applications %>%
  left_join(examiner_dates, by = "examiner_id")

rm(examiner_dates)
gc()
````

Save file as processed variables, to skip these steps in the following exercises.

````
\`\`\`{r save-file} write_feather(applications,
"/Users/sheidamajidi/Desktop/Winter2024/COURSES/ORGB671/Project
Data/app_data_starter_coded.feather")

# Rest of the exercise
````

First, let's load the data previously saved back into R.

````
```{r echo=FALSE}
# Load data
data_path <- "/Users/sheidamajidi/Desktop/Winter2024/COURSES/ORGB671/Project Data/app_data_starter.feather"
applications <- read_feather(data_path)
````

To be able to analyze the quarter, we must convert the date column in the dataset into R's Date format.

`{r} # Convert 'date' column to a Date format applications <- applications %>%   mutate(filing_date = as.Date(filing_date))`

Now, we can create the panel dataset of quarterly aggregated production measures for each examiner based on the requirements for each field given in the instructions:

```
`{r} panel_data <- applications %>%   mutate(quarter =       paste(year(filing_date), quarter(filing_date), s
```

We can first try linear regression to estimate these variables as predictors.

```
`{r} library(gtsummary) library(lme4)`
```

Example: Linear Regression for Turnover Prediction

````
``` {r} model_turnover \<- lm(current_art_unit ~ num_new_applications +
num_abandoned_applications + num_allowed_applications +
num_inprocess_applications + num_people_art_unit + num_women_art_unit +
num_examiners_by_race + separation_indicator + au_move_indicator, data =
panel_data) summary(model_turnover)
````

We can then try to calculate the turnover rate per quarter.

````
``` {r}
library(dplyr)

# Calculate turnover rate per quarter
turnover_rate_per_quarter <- panel_data %>%
  group_by(quarter) %>%
  summarise(
    total_examiners = n(), # Total number of examiners in the quarter
    total_separations = sum(separation_indicator, na.rm = TRUE), # Total number of separations
    turnover_rate = total_separations / total_examiners # Turnover rate calculation
  )

# Viewing the calculated turnover rates
print(turnover_rate_per_quarter)
````

We had some trouble figuring out the turnover rate. We used the dates when patents were issued and abandoned but often when we have one date, the other is missing. As such, we didn't have a comprehensive result for turnover rate.

Lastly, we export our results for submission.

```
`{r} rmarkdown::render("Group3_Exercise2.Rmd", output_format =  "md_document")`
```

Given the code and steps above, we can summarize our work in the following way.

Objective: To analyze patent examiner data for insights into demographics, work patterns, and decision-making processes.

Data Preparation: Data from app_data_sample.parquet and edges_sample.csv are loaded, Libraries like tidyverse, lubridate, and arrow are used for data manipulation and date handling.

Gender Identification: Using gender library on first names from examiner_name_first field.

Race Estimation: Utilizing wru package to estimate racial demographics based on surnames.

Professional Tenure Analysis: Calculating tenure of each examiner in the organization by determining the time interval between their first and last observed application dates.

Quarterly Performance Analysis: Aggregating data quarterly and examining various performance metrics like number of new, abandoned, allowed, and in-process applications.

Statistical Modeling: Implementing linear and logistic regression models to predict factors influencing examiner turnover and changes in Art Units (AU).

Utilizing gtsummary and lme4 for model summaries and visualization.

Our Rationale

Data-Driven Decisions: Better understanding of the workforce dynamics and decision-making patterns in the examination process.

Inclusivity & Fairness: Demographic analysis (gender and race) ensures a diverse and equitable work environment.

Efficiency Improvement: Insights from tenure and quarterly performance analysis can optimize resource allocation and process efficiency.

Predictive Modeling: Regression models provide predictive insights for better planning and policy-making.

Results interpretation

Gender prediction: The gender of examiners was predicted from first names. This method is not foolproof, especially for unisex or culturally diverse names. The predictions should be treated as estimates, not absolute identifications.

Race estimation: Racial categories were estimated from surnames, a method that has inherent limitations and may not accurately reflect the complex nature of racial identities. These estimations are broad and probabilistic.

Merging demographic data: Gender and race estimations were merged into the main dataset, adding demographic dimensions. It's important to remember these are based on estimations and carry uncertainties.

Tenure calculation: Examiner tenure was calculated from the range of observed application dates. This provides a proxy for the length of time examiners have been associated with the organization but may not precisely represent their actual employment period.

Quarterly data aggregation: The data was transformed for quarterly trend analysis. This approach helps in understanding patterns over time but may miss finer details visible in a shorter time frame.

"panel_data" for trend analysis: Panel data was created to analyze examiners' performance and behavior over time. This aggregation allows for a broad view of trends but can generalize individual variations.

Underlying Assumptions (Code)

The method of predicting gender using the `gender` package and race using the `wru` package based on names has inherent limitations, as discussed in Holland's "Causation and Race" Report. The accuracy of these predictions may vary due to cultural diversity and the evolving nature of names and racial identities.

The calculation of tenure using `lubridate` is based on the range of observed application dates. This method provides a useful proxy for understanding an examiner's duration with the organization, but it may not accurately reflect their actual employment period. Implications of this approach on understanding workforce dynamics are explored in Rosenow's article.

The use of `gtsummary` for creating descriptive tables is based on the assumption that summarizing complex data in a comprehensible format enhances the interpretability of the results.

Links to Readings & Course Materials

The demographic analysis and performance metrics are assumed to contribute to a more inclusive, fair, and efficient work environment. These assumptions are grounded in contemporary organizational theories and practices, as suggested by the foundational principles in the regression readings.

Quarterly aggregation of production data adds a time element to the analysis, underscoring the value of temporal insights in discerning work trends and decision-making.

Utilizing regression models to predict turnover and mobility emphasizes the complexity of modeling human behavior and organizational dynamics as described in Biderman's paper on predicting turnover using alternative analytics.
```