exercise3

jingwen he

2024-04-02

# change to your own path!  
 install.packages("arrow")  
library(arrow)  
data\_path <- "C:/Users/17286/Downloads/"  
applications <- read\_parquet(paste0(data\_path,"app\_data\_sample.parquet"))  
install.packages("readr")  
library(readr)  
edges <- read\_csv(paste0(data\_path,"edges\_sample.csv"))  
applications  
edges  
  
install.packages("gender")  
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 repetitions  
library(magrittr)  
library(dplyr)  
  
examiner\_names <- applications %>%   
 distinct(examiner\_name\_first)  
examiner\_names  
  
# get a table of names and gender  
install.packages("tidyr")  
library(tidyr)  
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  
  
# 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()  
install.packages("wru")  
library(wru)  
  
examiner\_surnames <- applications %>%   
 select(surname = examiner\_name\_last) %>%   
 distinct()  
  
examiner\_surnames  
  
examiner\_race <- predict\_race(voter.file = examiner\_surnames, surname.only = T) %>%   
 as\_tibble()  
examiner\_race  
  
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  
# 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()  
  
library(lubridate) # to work with dates  
  
examiner\_dates <- applications %>%   
 select(examiner\_id, filing\_date, appl\_status\_date)   
  
examiner\_dates  
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)  
  
examiner\_dates  
  
applications <- applications %>%   
 left\_join(examiner\_dates, by = "examiner\_id")  
  
rm(examiner\_dates)  
gc()

# split two workgroup  
group1 <- filter(applications, substr(examiner\_art\_unit, 1, 3) == "176")  
group2 <- filter(applications, substr(examiner\_art\_unit, 1, 3) == "162")  
  
library(dplyr)  
library(tidyr)  
  
# gender stats  
gender\_stats\_group1 <- group1 %>% count(gender) %>% mutate(group = "Group 1")  
gender\_stats\_group2 <- group2 %>% count(gender) %>% mutate(group = "Group 2")  
  
# race stats  
race\_stats\_group1 <- group1 %>% count(race) %>% mutate(group = "Group 1")  
race\_stats\_group2 <- group2 %>% count(race) %>% mutate(group = "Group 2")  
  
# tenue stats  
tenure\_stats\_group1 <- group1 %>% summarise(mean\_tenure = mean(tenure\_days, na.rm = TRUE))  
tenure\_stats\_group2 <- group2 %>% summarise(mean\_tenure = mean(tenure\_days, na.rm = TRUE))  
  
# overall  
gender\_stats <- bind\_rows(gender\_stats\_group1, gender\_stats\_group2)  
race\_stats <- bind\_rows(race\_stats\_group1, race\_stats\_group2)  
tenure\_stats <- bind\_rows(tenure\_stats\_group1, tenure\_stats\_group2, .id = "group")  
print(gender\_stats)  
print(race\_stats)  
print(tenure\_stats)  
  
library(ggplot2)  
  
# plot the gender comparison  
ggplot(gender\_stats, aes(x=gender, y=n, fill=group)) +  
 geom\_bar(stat="identity", position=position\_dodge()) +  
 labs(title="Gender Distribution", x="Gender", y="Count")  
  
# plot the tace comparison  
ggplot(race\_stats, aes(x=race, y=n, fill=group)) +  
 geom\_bar(stat="identity", position=position\_dodge()) +  
 labs(title="Race Distribution", x="Race", y="Count")  
  
# plot the tenue comparison  
ggplot(tenure\_stats, aes(x=group, y=mean\_tenure, fill=group)) +  
 geom\_bar(stat="identity") +  
 labs(title="Average Tenure", x="Group", y="Mean Tenure (days)")