Data challenge: data exploration

true

March 2021, 22nd

Abstract

In this notebook the notions presented in the first class are used, such as data loading, data exploration, data management, and also computation of point estimates and their confidence intervals. First, this notebook proposes data exploration proceadure. Note that on the contrary to the Python notebook, it is less standardized (in term of string management or column processing), because we do not propose a pipeline to process data, but we rather propose to explore the data. This notebook contains simple, and less simple, plots using the library `ggplot2`. Don't hesitate to improve these plots to go further, but also to propose totally new representations. Their is not only one good way to visualize data, but rather plenty of ways. The end of the notebook proposes to study the salary difference between men and women and recalls how to compare the means of two groups with a proper confidence interval. The last code chunk proposes to match individuals with respect to their job class, and could be the first step to explain the salary difference previously measured.

Contents


```
library(ggplot2) # for histogram
library(lubridate) # for date management
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
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(stringr)
```

```
salaries <- read.csv(file = "./data/train_data.csv")
```

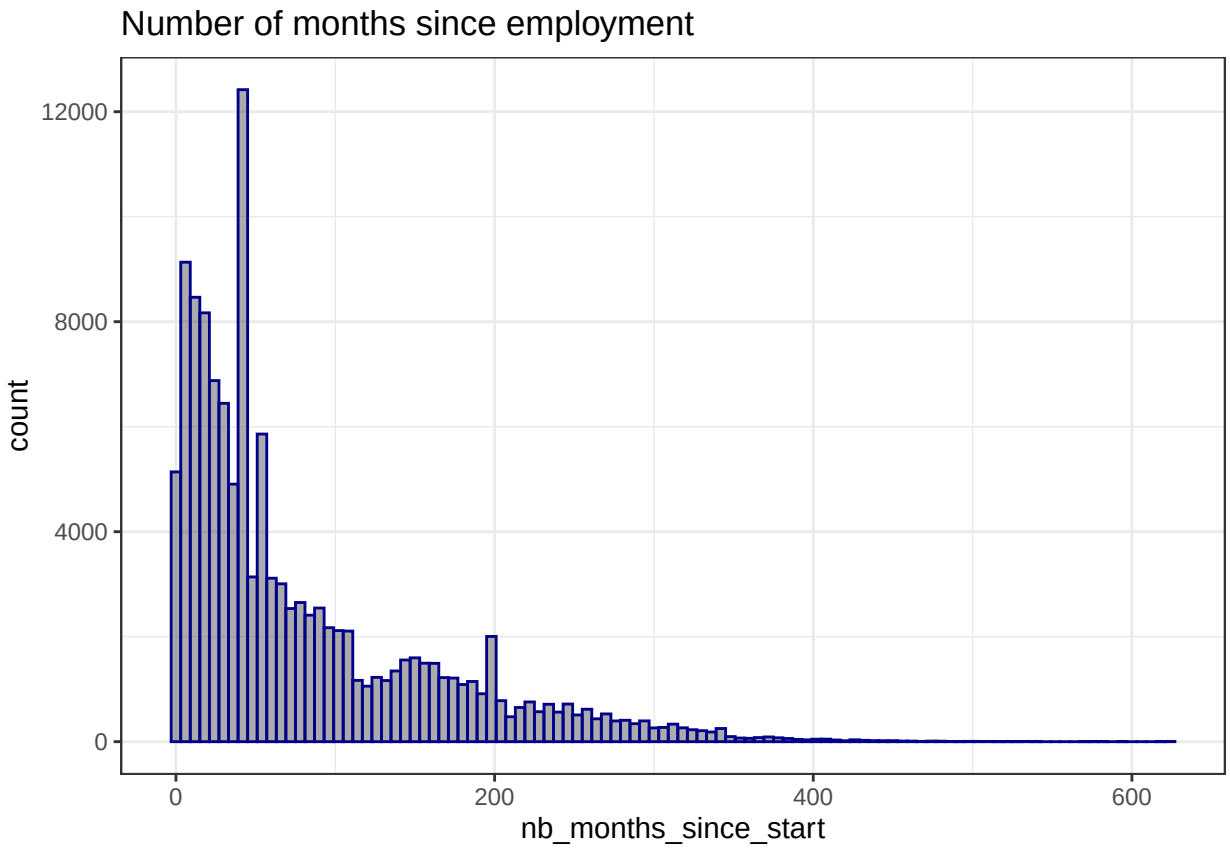
Clean data

```
# remove useless spaces in the sex names
salaries$SEX <- str_replace_all(salaries$SEX, '\\s+', "")
```

Time since employment

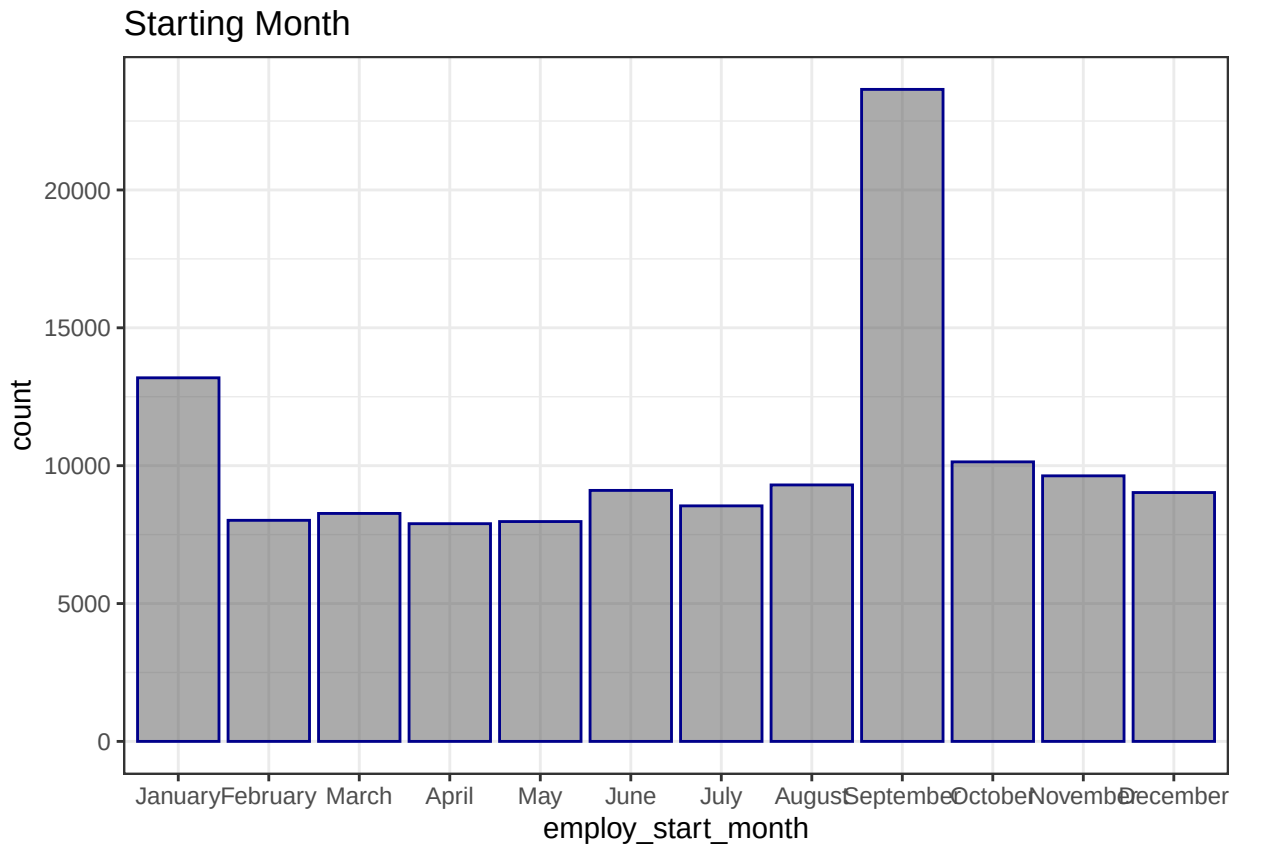
```
# Change date using package lubridate
salaries$date <- mdy(salaries$HIREDT)
# correct error for 2068
salaries[111174, "date"] <- "1968-12-01"


salaries$nb_months_since_start <- interval(salaries$date, "2021-03-01") %/% months(1)
salaries$nb_years_since_start <- floor(salaries$nb_months_since_start/12)
# create starting month and starting year
salaries$employ_start_month <- month(salaries$date, label = TRUE, abbr = FALSE)
salaries$employ_start_year <- year(salaries$date)
```

```
library(ggplot2)
ggplot(salaries, aes(x = nb_months_since_start)) +
  geom_histogram(binwidth = 6, alpha = 0.5, color = "darkblue") + # semester
  theme_bw() +
  ggtitle("Number of months since employment")
```

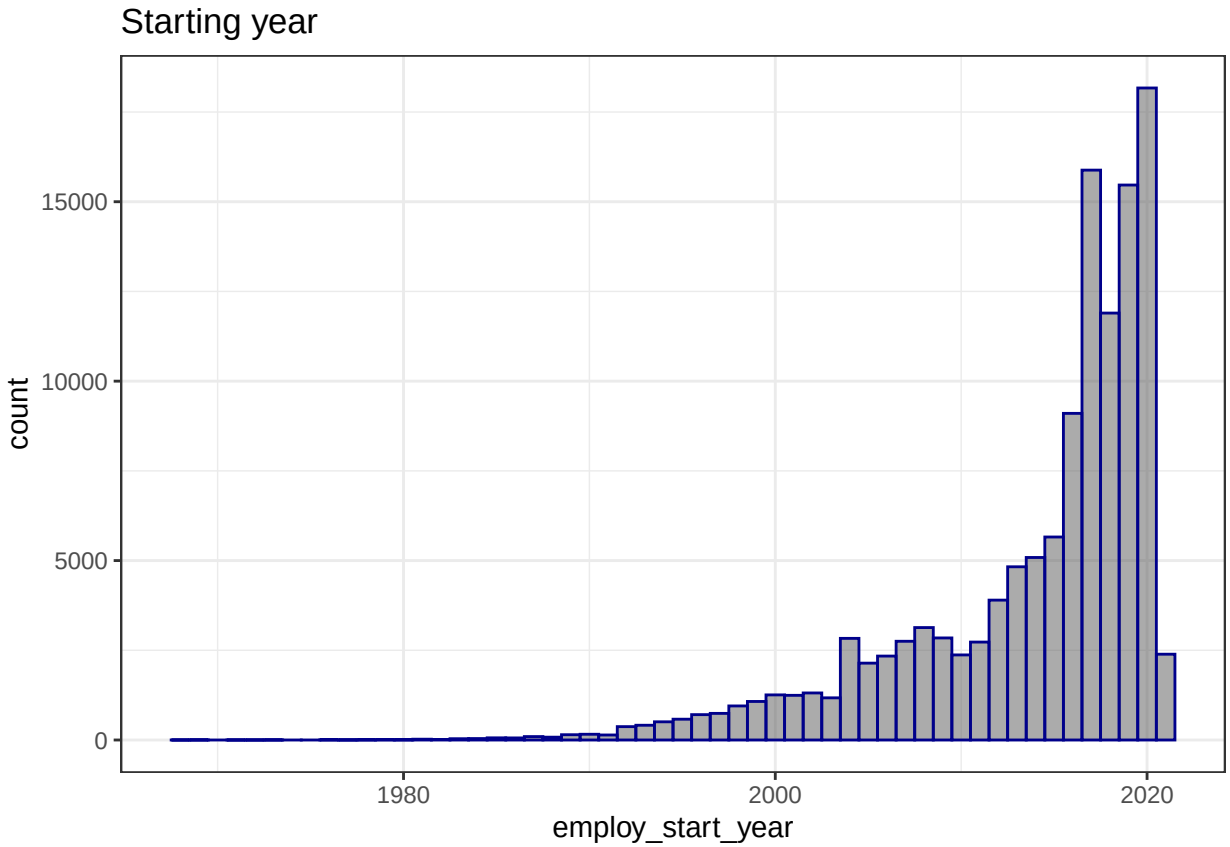



```
ggplot(salaries, aes(x = employ_start_month)) +
  geom_bar(alpha = 0.5, color = "darkblue") +
```

```
  theme_bw() +
  ggtitle("Starting Month")
```




```
ggplot(salaries, aes(x = employ_start_year)) +
  geom_histogram(binwidth = 1, alpha = 0.5, color = "darkblue") +
  theme_bw() +
  ggtitle("Starting year")
```

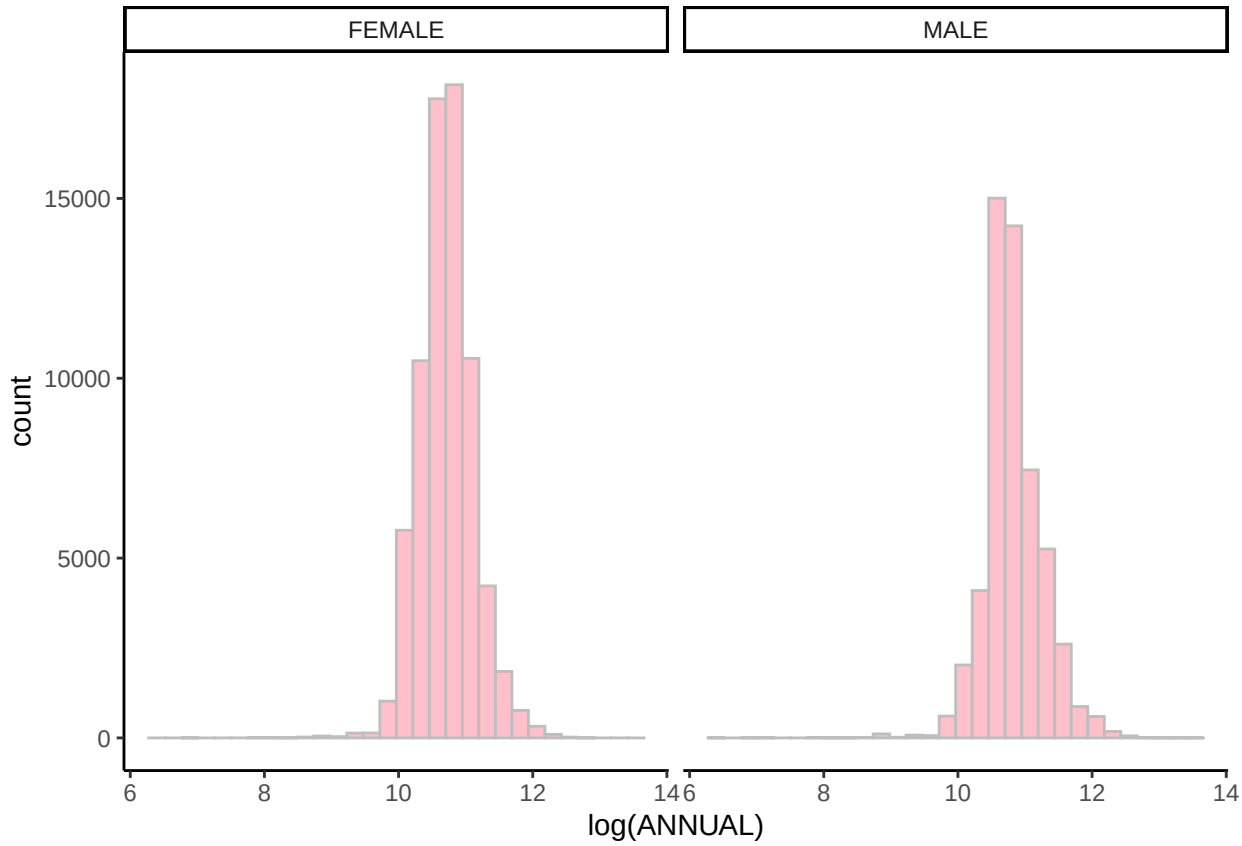



```
## solution de Tristan
# pacman::p_load(tidyverse, lubridate)
# raw_df <- read_csv("data/train_data.csv")
# data_df <-
#   raw_df %>%
#   mutate(HIREDT = parse_date_time2(HIREDT, "mdy", cutoff_2000 = 50L)) %>%
#   arrange(desc(HIREDT))
```

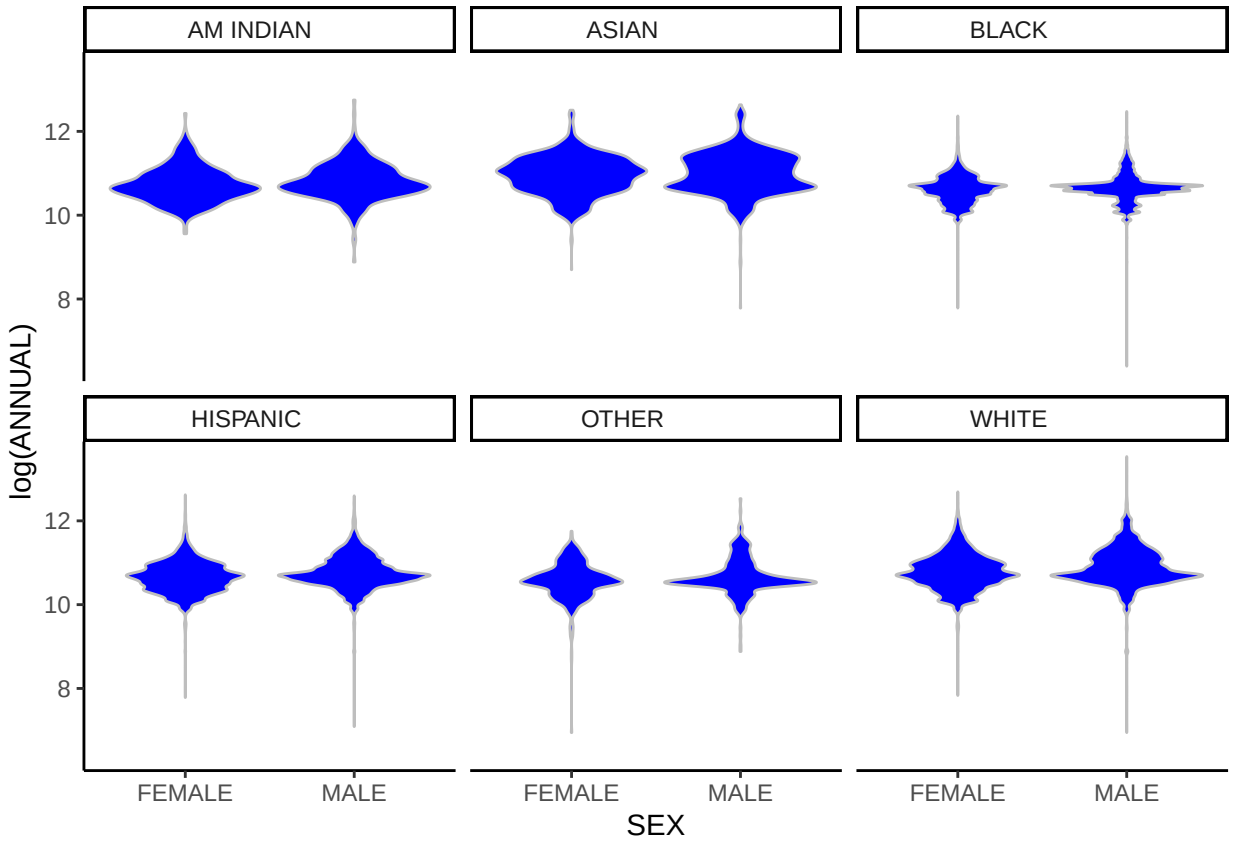
ANNUAL quick description

```
ggplot(salaries, aes(x = log(ANNUAL))) +
  geom_histogram(fill  = "pink", color = "grey") +
  facet_grid(~SEX) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



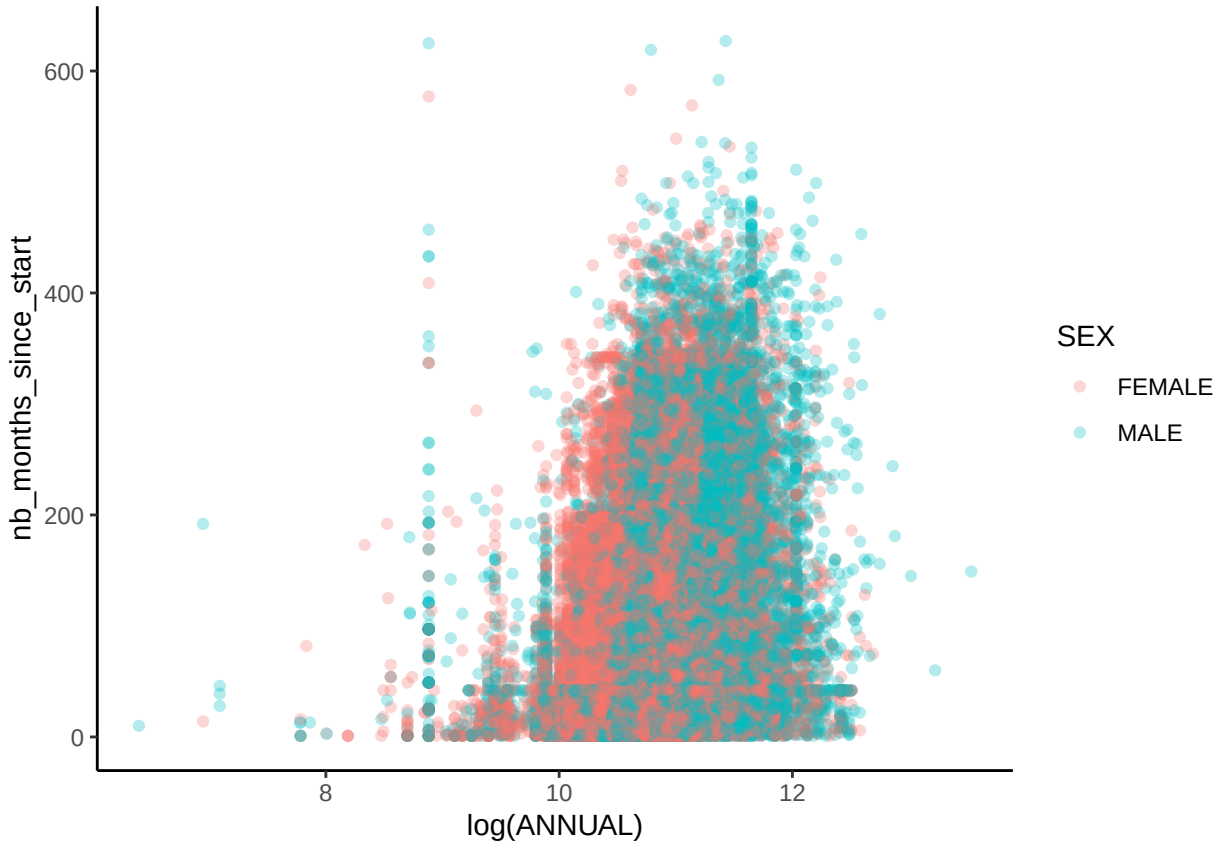

```
ggplot(salaries, aes(x = SEX, y = log(ANNUAL))) +
  geom_violin(fill  = "blue", color = "grey") +
  #geom_jitter(alpha = 0.3, width = 0.2) +
  facet_wrap(~RACE) +
  theme_classic()
```



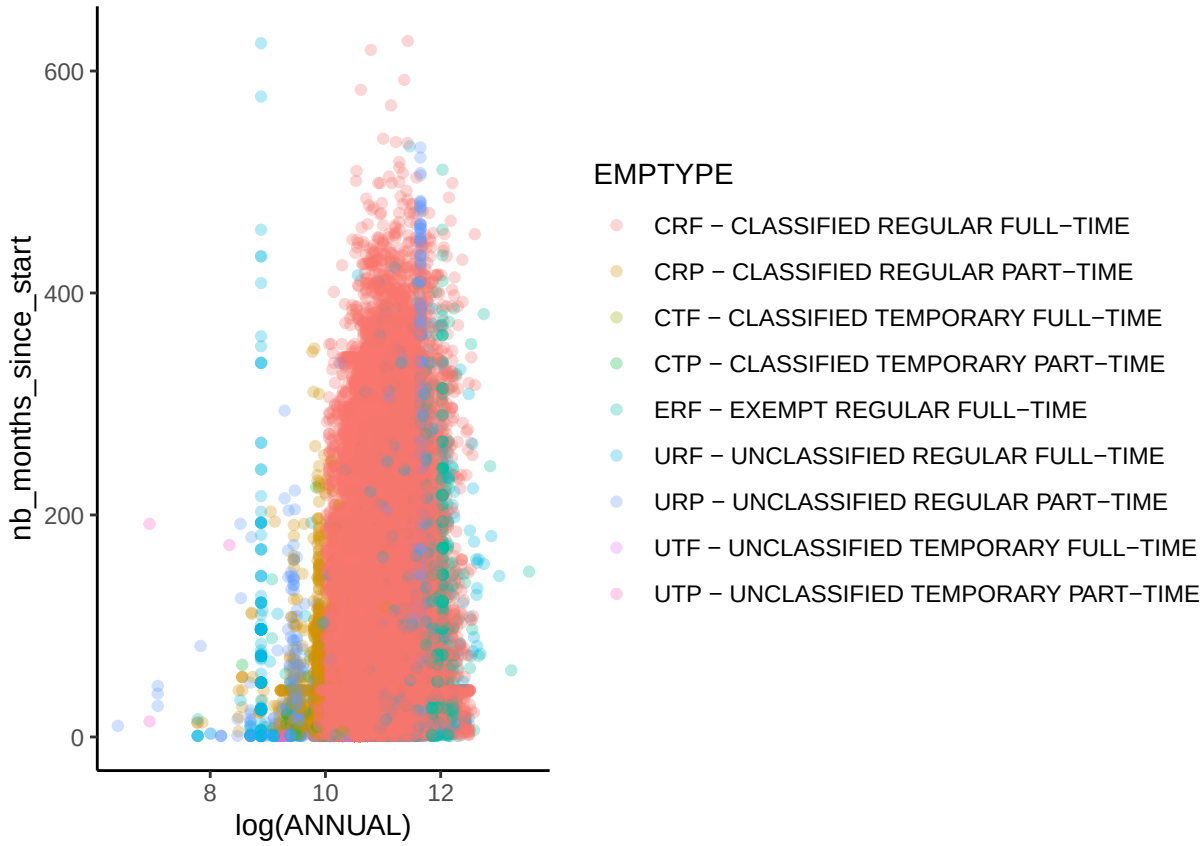

```
table(salaries$RACE)
```

```
##
## AM INDIAN       ASIAN           BLACK           HISPANIC        OTHER
##             592            3725            29713           33831            1069
## WHITE
##           55797
```

```
ggplot(salaries, aes(y = nb_months_since_start, x = log(ANNUAL), color = SEX)) +
  geom_point(alpha = 0.3) +
  theme_classic()
```




```
ggplot(salaries, aes(y = nb_months_since_start, x = log(ANNUAL), color = EMPTYPE)) +
  geom_point(alpha = 0.3) +
  theme_classic()
```



Emptype is probably very useful when doing the prediction task (at least for extreme wages prediction)! Also, interestingly, some emptype seem to have the same salaries no matter the experience in the administration.

```
salaries[, c("EMPTYPE", "nb_years_since_start", "ANNUAL")] %>%
  group_by(EMPTYPE, nb_years_since_start) %>%
  summarise(mean_annual = mean(ANNUAL), n = n())
```

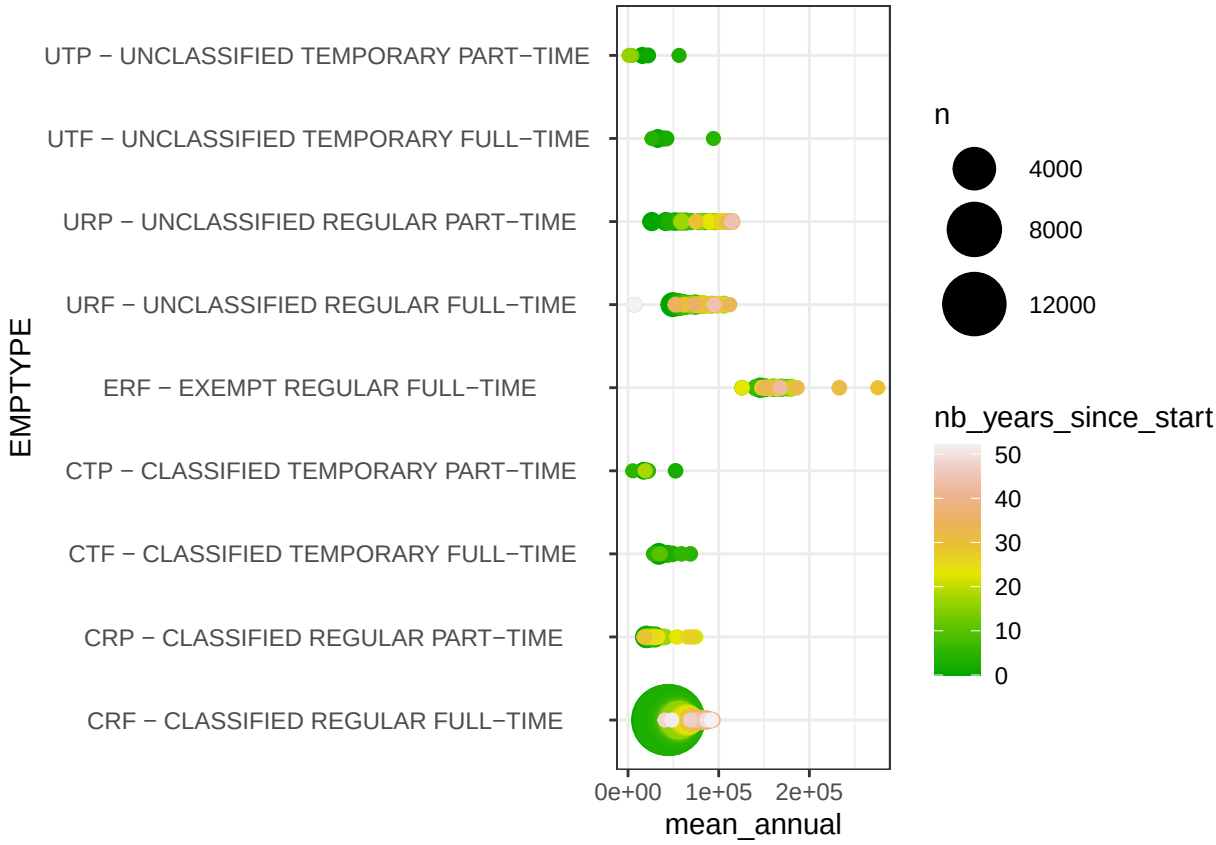
```
## `summarise()` regrouping output by 'EMPTYPE' (override with `.groups` argument)
```

```
## # A tibble: 228 x 4
## # Groups:   EMPTYPE [9]
##    EMPTYPE                                  nb_years_since_sta~ mean_annual     n
##    <fct>                                                  <dbl>       <dbl> <int>
##  1 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    0      42856. 15714
##  2 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    1      44046. 15611
##  3 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    2      46363. 12038
##  4 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    3      45496. 15976
##  5 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    4      53048.  9024
##  6 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    5      52271.  5611
##  7 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    6      54322.  4866
##  8 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    7      54960.  4741
##  9 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    8      55710.  3868
## 10 "CRF - CLASSIFIED REGULAR FULL-TIME   ~                    9      54431.  2930
## # ... with 218 more rows
```

```
salaries %>%
  group_by(EMPTYPE, nb_years_since_start) %>%
  summarise(mean_annual = mean(ANNUAL), n = n()) %>%
  ggplot(aes(y = mean_annual, x = EMPTYPE, size = n, colour = nb_years_since_start)) +
```

```
  geom_point() +
  scale_size(range = c(2, 12)) +
  coord_flip() +
  scale_colour_gradientn(colours = terrain.colors(10)) +
  theme_bw()
```

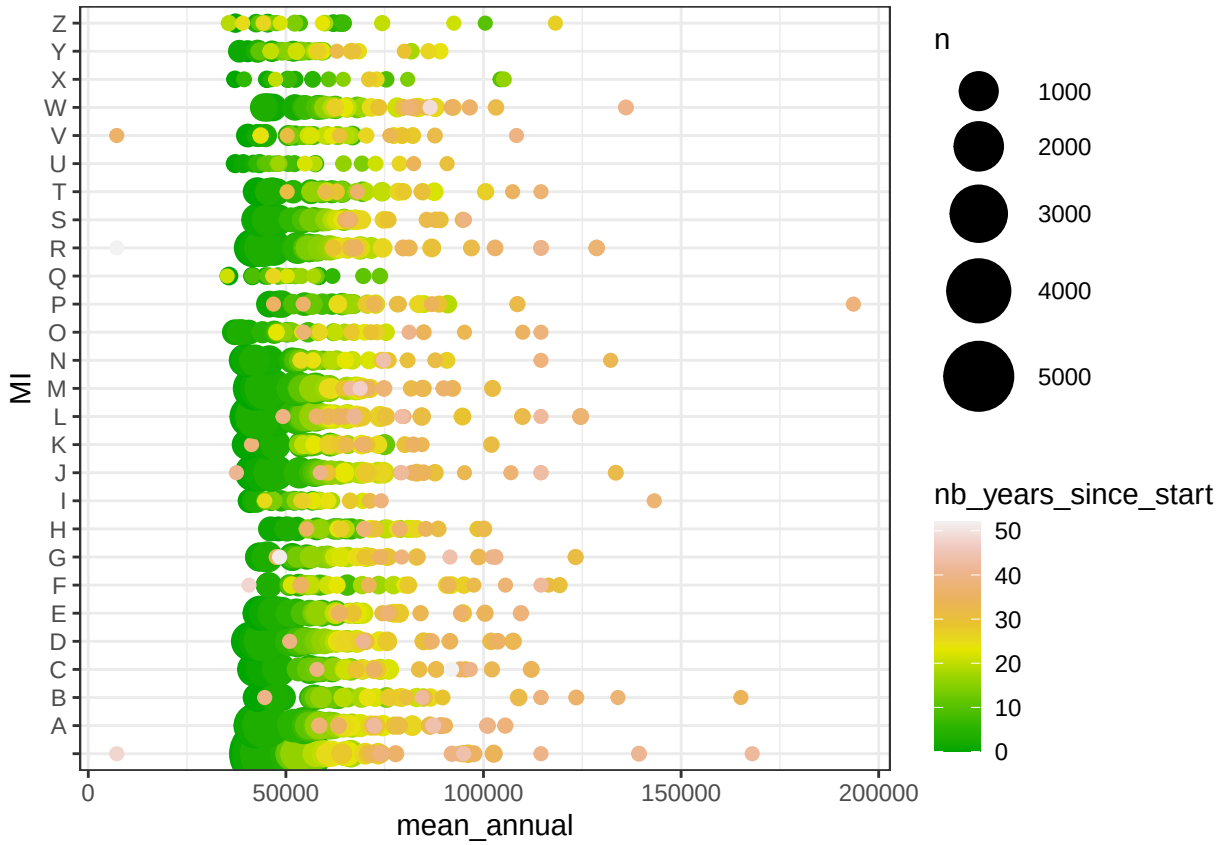
```
## `summarise()` regrouping output by 'EMPTYPE' (override with `.groups` argument)
```



Is MI important?

```
salaries %>%
  group_by(MI, nb_years_since_start) %>%
  summarise(mean_annual = mean(ANNUAL), n = n()) %>%
  ggplot(aes(y = mean_annual, x = MI, size = n), colour = nb_years_since_start) +
  geom_point(aes(colour = nb_years_since_start)) +
  scale_size(range = c(2, 12)) +
  coord_flip() +
  scale_colour_gradientn(colours = terrain.colors(10)) +
  theme_bw()
```

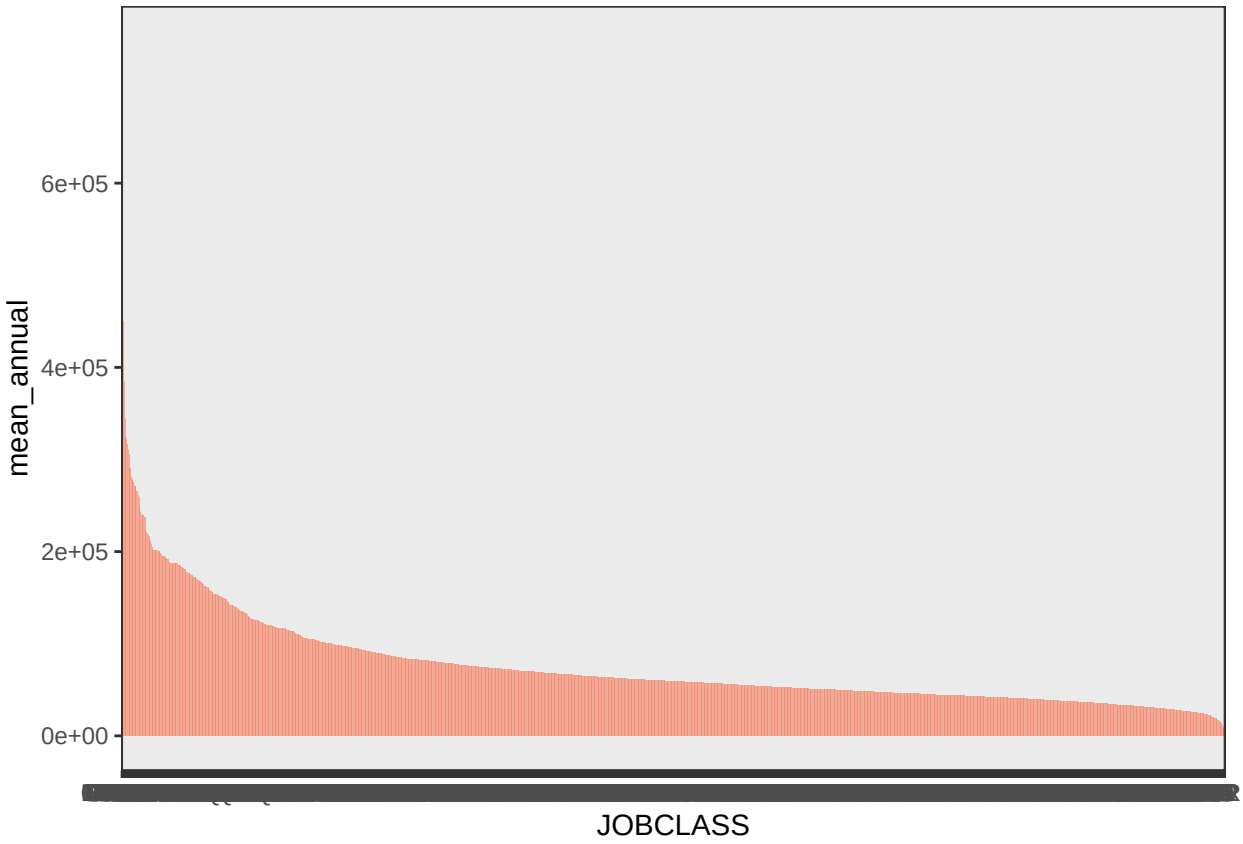
```
## `summarise()` regrouping output by 'MI' (override with `.groups` argument)
```



MI does not seem correlated to the progression in time, and not so related to a specific mean salary.

```
library(forcats) # reorder
salaries %>%
  group_by(JOBCLASS) %>%
  summarise(mean_annual = mean(ANNUAL), n = n()) %>%
  mutate(JOBCLASS = fct_reorder(JOBCLASS, desc(mean_annual))) %>%
  ggplot(aes(x = JOBCLASS, y = mean_annual)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
  scale_size(range = c(2, 12)) +
  theme_bw()
```

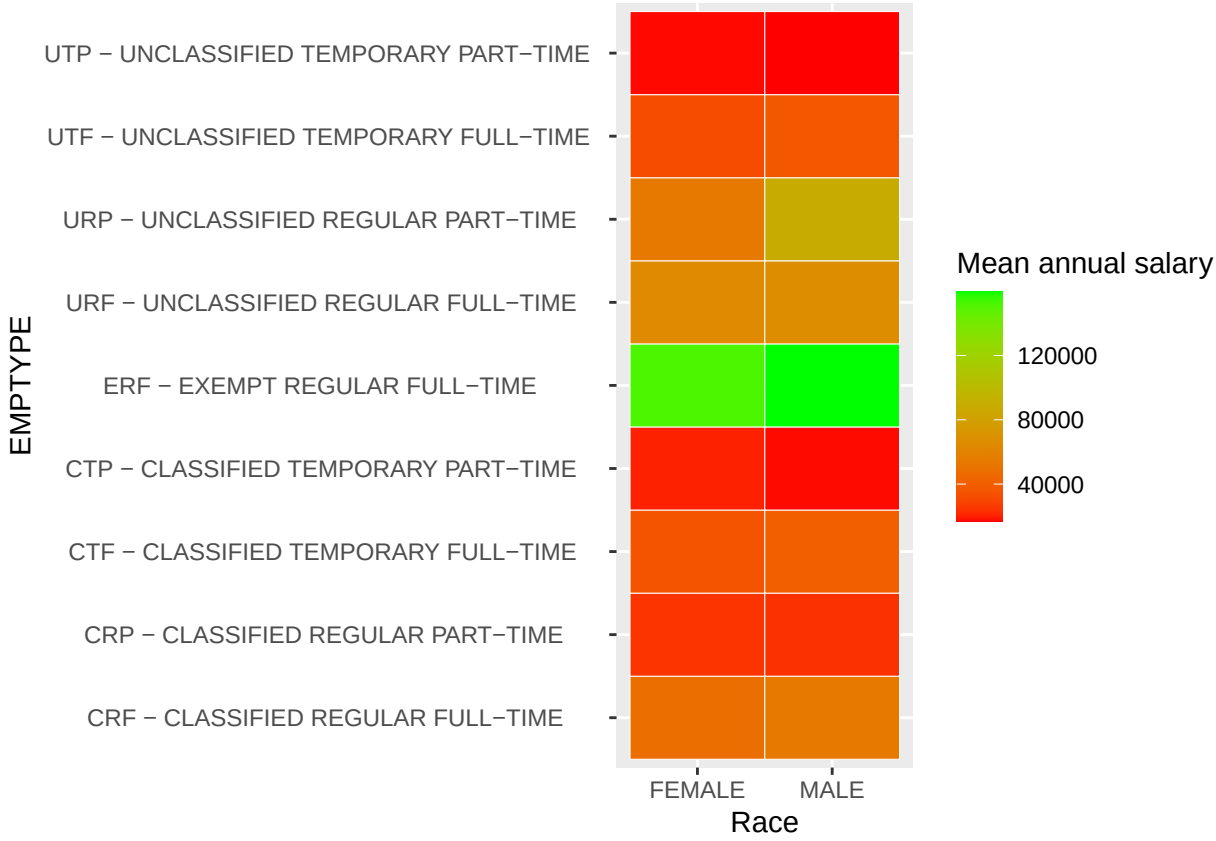
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



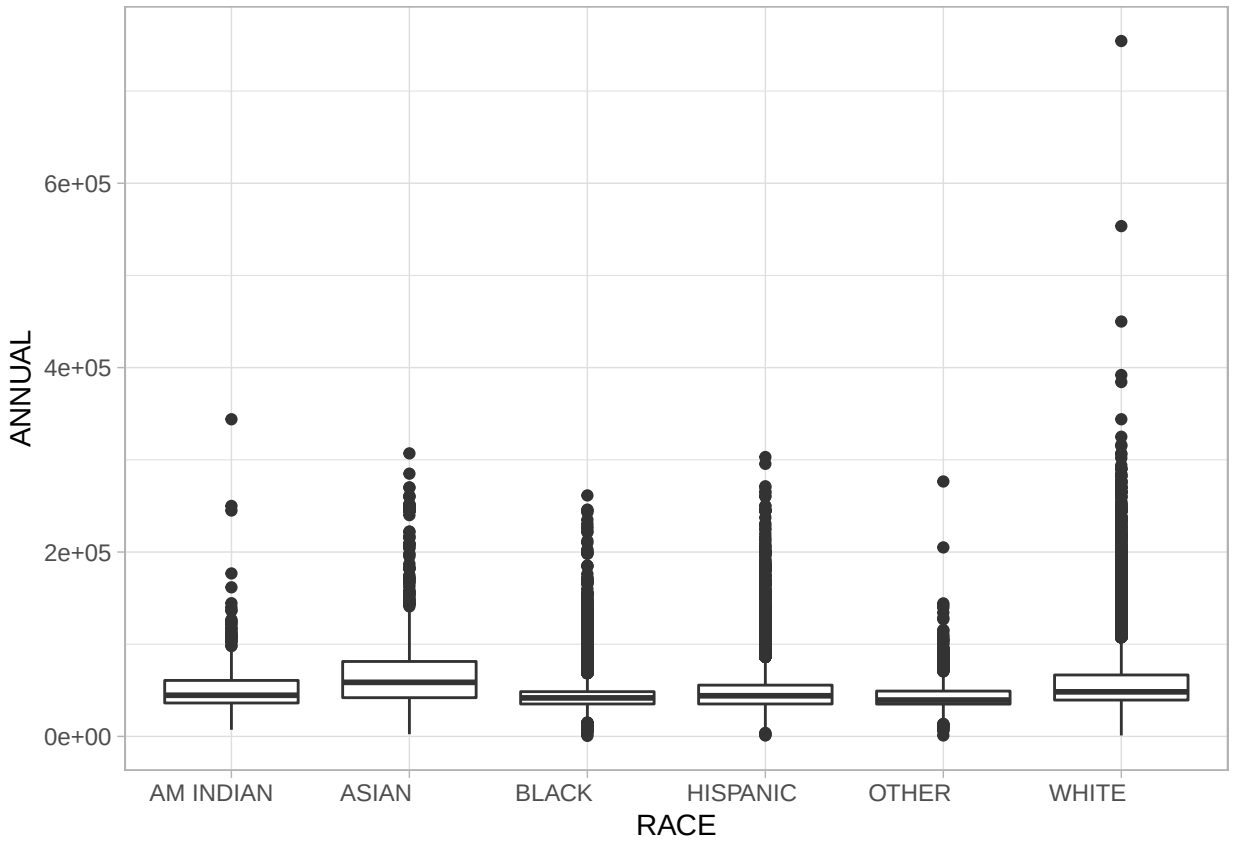
JOBCLASS seems important also.

```
salaries %>%
  group_by(EMPTYPE, SEX) %>%
  summarise(mean_annual = mean(ANNUAL)) %>%
  ggplot(aes(SEX, EMPTYPE, fill = mean_annual)) +
  geom_tile(colour = "white") +
  scale_fill_gradient(low="red", high="green") +
  labs(x="Race",
       y="EMPTYPE",
       fill="Mean annual salary")
```

```
## `summarise()` regrouping output by 'EMPTYPE' (override with `.groups` argument)
```




```
ggplot(salaries, aes(x = RACE, y = ANNUAL)) +
  geom_boxplot() +
  theme_light()
```




```
tmp <-salaries %>%
  group_by(RACE, SEX) %>%
  summarise(mean_exp = mean(nb_months_since_start), mean_annual = mean(ANNUAL), count = n())
```

```
## `summarise()` regrouping output by 'RACE' (override with `.groups` argument)
```

```
tmp
```

```
## # A tibble: 12 x 5
## # Groups:   RACE [6]
##    RACE              SEX    mean_exp mean_annual count
##    <fct>             <chr>     <dbl>       <dbl> <int>
##  1 "AM INDIAN      " FEMALE     70.7      50020.   309
##  2 "AM INDIAN      " MALE       82.9      55001.   283
##  3 "ASIAN          " FEMALE     65.0      63169.  2084
##  4 "ASIAN          " MALE       73.5      67421.  1641
##  5 "BLACK          " FEMALE     70.1      43427. 19636
##  6 "BLACK          " MALE       69.2      44755. 10077
##  7 "HISPANIC       " FEMALE     77.0      45708. 19896
##  8 "HISPANIC       " MALE       85.8      51288. 13935
##  9 "OTHER          " FEMALE     22.9      42362.   657
## 10 "OTHER          " MALE       21.8      47319.   412
## 11 "WHITE          " FEMALE     79.0      53424. 28858
## 12 "WHITE          " MALE       96.8      60458. 26939
```

```
# here are others command lines that could be useful to pivot your data
#pivot_longer()
#pivot_wider()
```

Effect of gender

A first naive approach is to compare the mean of the two groups.

```
mean(salaries[salaries$SEX == "MALE", "ANNUAL"]) - mean(salaries[salaries$SEX == "FEMALE", "ANNUAL"])
```

```
## [1] 6479.095
```

We observe a difference, but is it significant?

Here is the detailed code to measure the difference in means between women and men, and the confidence interval.

To compute the confidence interval, it uses the central limit theorem, and the fact that the empirical mean converges toward a normal distribution. So that we can estimate the confidence interval with a typical gaussian density function (it corresponds to the line with the 1.96)

```
# Filter treatment / control observations, pulls outcome variable as a vector
target_women <- salaries[salaries$SEX == "FEMALE", "ANNUAL"] # Outcome for women
target_men <- salaries[salaries$SEX == "MALE", "ANNUAL"] # Outcome for men

n_women <- nrow(salaries[salaries$SEX == "FEMALE",]) # Number of women
n_men <- nrow(salaries[salaries$SEX == "MALE",]) # Number of men

# Difference in means
estimated_difference <- mean(target_men) - mean(target_women)

# 95% Confidence intervals
se_hat <- sqrt(var(target_women)/(n_women-1) + var(target_men)/(n_men-1) )
lower_ci <- estimated_difference - 1.96 * se_hat
upper_ci <- estimated_difference + 1.96 * se_hat

result = c(Difference = estimated_difference, lower_ci = lower_ci, upper_ci = upper_ci)
print(result)
```

```
## Difference   lower_ci   upper_ci
##   6479.095   6185.333   6772.857
```

We find that men have a higher wage than women + 6479, and the confidence interval CI[6185 - 6772] allows to say it is significant.

You can also use the regular R command we presented in class, and check we have the same result (the small difference is probably due to the fact that in the manual coded version we used 1.96)

```
t.test(target_women, target_men)
```

```
##
##  Welch Two Sample t-test
##
## data:  target_women and target_men
## t = -43.229, df = 99436, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6772.853 -6185.337
## sample estimates:
## mean of x mean of y
##  48695.36  55174.46
```

Don't forget to provide confidence interval rather than point estimate! If you don't understand the whole process to compute the confidence interval is not so important. The most important thing is to remember

how to compute the confidence interval when you have to compare the means of two groups, or at least to remember that this is a key information before sketching the conclusion.

Usually the result in newspaper is in percentage.

```
mean_men = mean(target_men)
mean_women = mean(target_women)
percentage = mean_men/mean_women
percentage = (percentage-1)*100
percentage
```

```
## [1] 13.30536
```

On average, men are paid 13% more than women.

But, is it due to the fact that women are women, so that they are less paid? Or is it due to the fact that men occupy different work position than women?

How can we get a little bit further than the naive difference in means?

For example, we can investigate if there is a difference in the salary for women and men with same job and ethnicity. If we observe a difference, we could suppose that sexism exists. If not, the salary difference is probably due to the fact that women occupy different work categories.

> N.B: we have no judgment on this political and sociological question, but understanding the reason for this difference is very important before drawing any public policy. This question highlights how complicated such a question can be.

```
# For matching

library(MatchIt)
```

```
## Warning: package 'MatchIt' was built under R version 3.6.2
```

```
covariates_to_keep <- c("SEX", "RACE", "EMPTYPE", "NAME", "JC.TITLE")
m.out <- matchit(SEX ~ ., method = "exact", data = salaries[, covariates_to_keep])
salaries.matched <- match.data(m.out)

# # you can check that each subgroup is the same respectively to the covariates chosen to match
engineer <- salaries.matched[salaries.matched$subclass == 10,]
```