Salary report by instructor level

Nick Cauldron

2024-02-04

Data import

```
# Table parsing
# remotes::install_cran("tidyverse")
library(tidyverse)
library(scales)

# Adding gender
# remotes::install_cran("gender")
# remotes::install_github("lmullen/genderdata")
library(gender)
library(genderdata)
```

```
salary_csv_osu_2023_f <- "salaries.csv"

salaries_f <- salary_csv_osu_2023_f
salaries_raw <- read_csv(salaries_f, show_col_types = FALSE) %>%
  janitor::clean_names()

salaries_raw
```

```
## # A tibble: 3,921 x 14
##    name       first_hired home_orgn adj_service_date job_orgn job_type job_title
##    <chr>      <chr>       <chr>     <chr>            <chr>    <chr>    <chr>
##  1 Abakar, R~ 01-AUG-2022 MSA - De~ 01-AUG-2022      MSA - D~ P        OSU Assi~
##  2 Abbas, Ho~ 31-DEC-2018 ESE - Sc~ 31-DEC-2018      ESE - S~ P        Assistan~
##  3 Abbasi, B~ 01-AUG-2017 LCB - Ac~ 01-AUG-2017      LCB - A~ P        Associat~
##  4 Abbott, J~ 30-MAR-2015 XEM - Ad~ 30-MAR-2015      XEM - A~ P        Asst Dir~
##  5 Abel, Hen~ 01-JAN-1998 HHS - Hl~ 01-JAN-1998      HHS - H~ P        Motorcyc~
##  6 Ables, Sc~ 16-DEC-2018 CLA - Sc~ 16-DEC-2018      CLA - S~ P        Instruct~
##  7 Abney, La~ 17-JUN-2015 ENG - Co~ 17-JUN-2015      ENG - C~ P        Technica~
##  8 Abrams, E~ 01-JUN-2010 MRS - Re~ 22-AUG-2016      MRS - R~ P        ALI Oper~
##  9 Ackers, S~ 14-FEB-2000 AFW - Fi~ 14-FEB-2000      AFW - F~ P        Senior F~
## 10 Adam, Ben~ 01-AUG-2019 EMM - Sc~ 01-AUG-2019      EMM - S~ P        Assistan~
## # i 3,911 more rows
## # i 7 more variables: posn_suff <chr>, rank <chr>, rank_effective_date <chr>,
## #   appt_begin_date <chr>, appt_percent <dbl>, appt_end_date <chr>,
## #   annual_salary_rate <dbl>
```

Convert the dates to real date columns to work with them. For reference date we use the date the report was printed. I found that by hand in the header on each page of the PDF. This chunk also calculate some more info about how long the person has worked here.

Lastly it separates out names for gender prediction in the next chunk. The first/middle name separation isn’t perfect. There are some edge cases where I can't tell if there are two first names or two middle names. For the sake of simplicity, I use the first name following the comma as the "first name," and all after are part of their "middle name"

```
report_print_date <- dmy("17-OCT-2023")
salaries <- salaries_raw %>%
  # clean organization column
  separate_wider_delim(job_orgn, delim = " - ",
                       names = c("job_orgn_code", "job_orgn_desc"),
                       cols_remove = FALSE, too_many = "merge") %>%
  # clean dates
  mutate(
    across(c(contains("date"), contains("hired")),
    dmy)) %>%
  mutate(appt_worked_days = report_print_date-appt_begin_date,
         rank_worked_days = report_print_date-rank_effective_date,
         appt_duration_days = appt_end_date-appt_begin_date,
         appt_remaining = appt_end_date-report_print_date) %>%
  mutate(appt_completed = case_when(appt_remaining > 0 ~ FALSE,
                                    appt_remaining <= 0 ~ TRUE,
                                    is.na(appt_remaining) ~ NA)) %>%
  # clean names - important for gender
  separate_wider_delim(name, delim = ", ",
                       names = c("name_last", "name_first_middle"),
                       cols_remove = FALSE) %>%
  separate_wider_delim(name_first_middle, delim = " ",
                       names = c("name_first", "name_middle"),
                       cols_remove = FALSE, too_few = "align_start", too_many = "merge") %>%
  # If someones first name is only one letter, use their middle name instead
  mutate(name_first = stringr::str_replace(name_first, "^\\w$", NA_character_)) %>%
  mutate(name_first = coalesce(name_first, name_middle))

glimpse(salaries)
```

```
## Rows: 3,921
## Columns: 25
## $ name_last           <chr> "Abakar", "Abbas", "Abbasi", "Abbott", "Abel", "Ab~
## $ name_first          <chr> "Reiman", "Houssam", "Bahman", "Joanna", "Henry", ~
## $ name_middle         <chr> NA, NA, NA, NA, NA, NA, "Daniel", "F", "Harry", NA~
## $ name_first_middle   <chr> "Reiman", "Houssam", "Bahman", "Joanna", "Henry", ~
## $ name                <chr> "Abakar, Reiman", "Abbas, Houssam", "Abbasi, Bahma~
## $ first_hired         <date> 2022-08-01, 2018-12-31, 2017-08-01, 2015-03-30, 1~
## $ home_orgn           <chr> "MSA - Dean of Students", "ESE - Sch Elect Engr/Co~
## $ adj_service_date    <date> 2022-08-01, 2018-12-31, 2017-08-01, 2015-03-30, 1~
## $ job_orgn_code       <chr> "MSA", "ESE", "LCB", "XEM", "HHS", "CLA", "ENG", "~
## $ job_orgn_desc       <chr> "Dean of Students", "Sch Elect Engr/Comp Sci", "Ac~
## $ job_orgn            <chr> "MSA - Dean of Students", "ESE - Sch Elect Engr/Co~
## $ job_type            <chr> "P", "P", "P", "P", "P", "P", "P", "P", "P", "P", ~
## $ job_title           <chr> "OSU Assist Responder", "Assistant Professor", "As~
## $ posn_suff           <chr> "C11439-00", "C18336-00", "C11566-00", "C11138-00"~
## $ rank                <chr> "No Rank", "Assistant Professor", "Associate Profe~
## $ rank_effective_date <date> 2022-08-01, 2018-12-31, 2022-09-16, 2021-07-12, 2~
## $ appt_begin_date     <date> 2022-08-01, 2018-12-31, 2022-09-16, 2021-07-12, 2~
## $ appt_percent        <dbl> 100, 100, 100, 100, 2, 27, 100, 100, 100, 100, 100~
## $ appt_end_date       <date> NA, NA, NA, NA, NA, 2023-06-15, NA, NA, 2023-06-3~
## $ annual_salary_rate  <dbl> 65004, 107901, 122499, 57492, 55404, 49437, 71760,~
## $ appt_worked_days    <drtn> 442 days, 1751 days, 396 days, 827 days, 838 days~
## $ rank_worked_days    <drtn> 442 days, 1751 days, 396 days, 827 days, 3941 day~
## $ appt_duration_days  <drtn> NA days, NA days, NA days, NA days, NA days, 272 ~
## $ appt_remaining      <drtn> NA days, NA days, NA days, NA days, NA days, -124~
## $ appt_completed      <lgl> NA, NA, NA, NA, NA, TRUE, NA, NA, TRUE, TRUE, NA, ~
```

Now add their expected gender. A person's age is important to accurately assign gender.. **I assumed an age range of 18-90 years old** (born 1934-2005). We draw example names from Social Security Administration data in the United States from 1930-2012.

A more precise naming-by-age strategy could probably be designed for some jobs. For example, there's a very high probability that assistant professors are younger than full professors. It's probably not worth expanding on that for now.

```
yr_birth_min <- 1934
yr_birth_max <- 2005
salaries_names_genders <- gender(unique(salaries$name_first), method = "ssa",
                                 years = c(yr_birth_min, yr_birth_max)) %>%
  select(-starts_with("year_"))
salaries_names_genders
```

```
## # A tibble: 1,343 x 4
##    name    proportion_male proportion_female gender
##    <chr>              <dbl>             <dbl> <chr>
##  1 Aaron           0.991             0.0089 male
##  2 Abbey           0.002             0.998  female
##  3 Abby            0.0031            0.997  female
##  4 Abigail         0.0022            0.998  female
##  5 Abraham         0.996             0.0037 male
##  6 Abree           0                 1      female
##  7 Adam            0.996             0.0043 male
##  8 Addison         0.363             0.637  female
##  9 Adel            0.525             0.475  male
## 10 Adela           0                 1      female
## # i 1,333 more rows
```

```
# join to salary date
# a few names were not in data,
# leave them as missing so models are easy and binary instead of a 3rd category "unknown"
salaries <- left_join(salaries, salaries_names_genders, by = c("name_first" = "name"))
```

Write to a file so others can join to salary data already parsed

```
write_csv(salaries_names_genders, file = "names_genders_USA.csv")
```

Names assigned to gender based on simple majority rules. Check confidence of that strategy by looking at the distribution of gender proportions that were assigned to each gender.

```
salaries_names_genders_analysis <- salaries_names_genders %>%
  pivot_longer(cols = c("proportion_male", "proportion_female"),
               values_to = "proportion",
               names_to = c("gender"),
               names_repair = "unique") %>%
  rename("gender_assigned" = gender...2, "gender_proportions" = gender...3) %>%
  mutate(gender_proportions = str_remove(gender_proportions, "proportion_"))
```

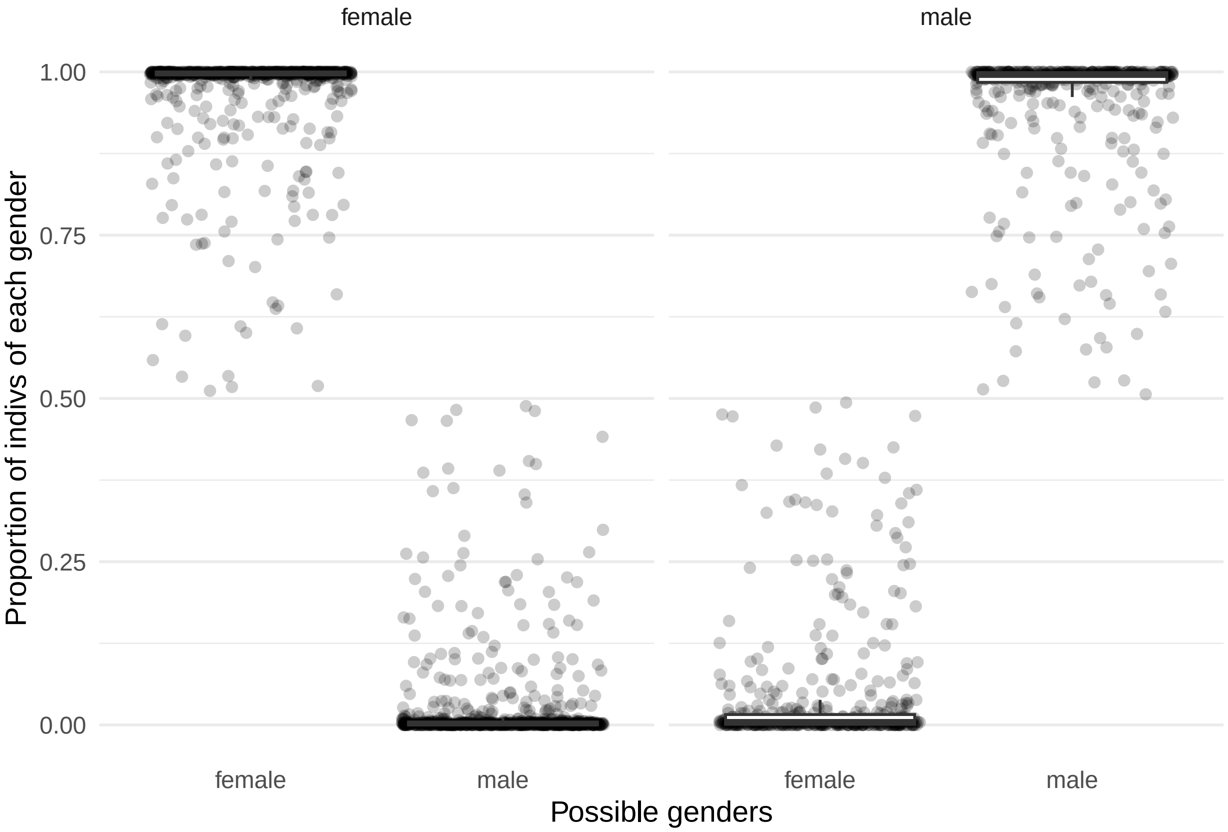
```
## New names:
```

```
## * `gender` -> `gender...2`
## * `gender` -> `gender...3`
```

```
salaries_names_genders_analysis
```

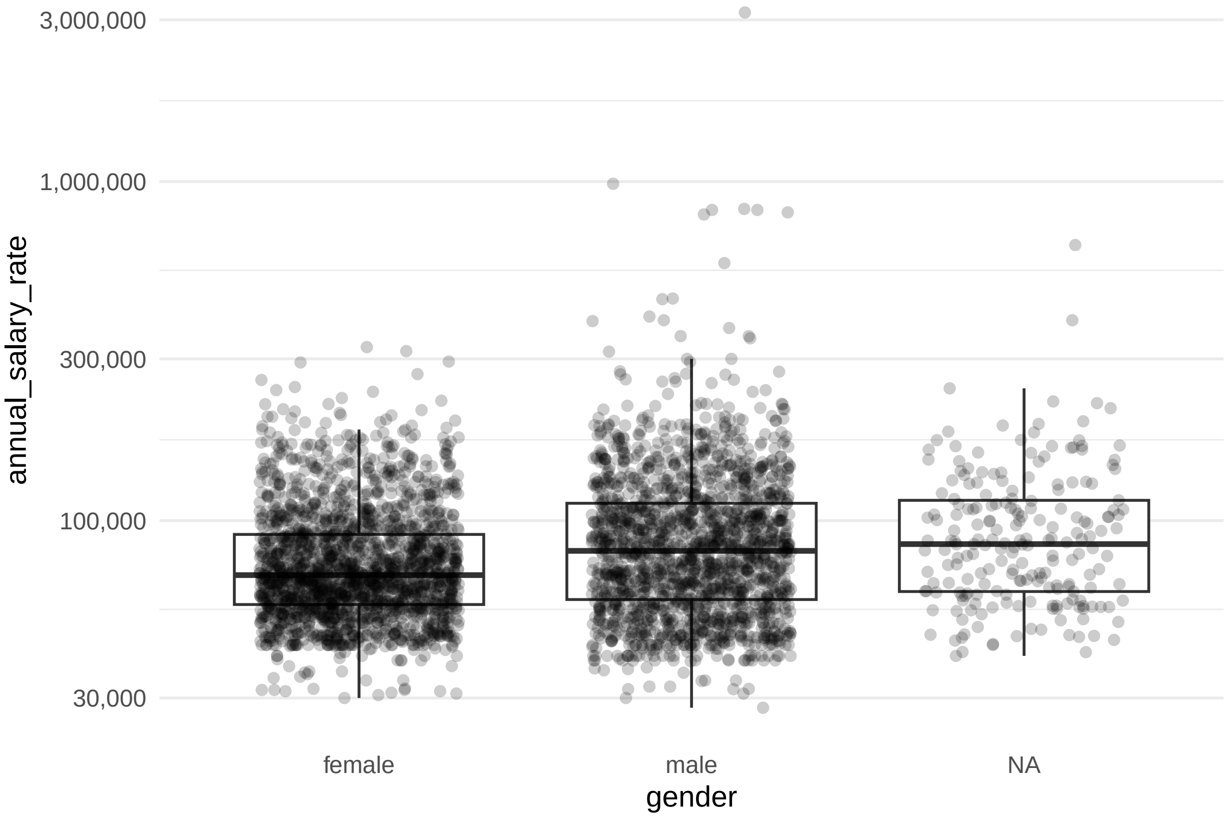
```
## # A tibble: 2,686 x 4
##    name    gender_assigned gender_proportions proportion
##    <chr>   <chr>           <chr>                   <dbl>
##  1 Aaron   male            male                   0.991
##  2 Aaron   male            female                 0.0089
##  3 Abbey   female          male                   0.002
##  4 Abbey   female          female                 0.998
##  5 Abby    female          male                   0.0031
##  6 Abby    female          female                 0.997
##  7 Abigail female          male                   0.0022
##  8 Abigail female          female                 0.998
##  9 Abraham male            male                   0.996
## 10 Abraham male            female                 0.0037
## # i 2,676 more rows
```

```
ggplot(salaries_names_genders_analysis, aes(x = gender_proportions, y = proportion)) +
  geom_jitter(width = 0.4, alpha = 0.2) +
  geom_boxplot(outlier.shape = NA) +
  facet_wrap(vars(gender_assigned)) +
  labs(y = "Proportion of indivs of each gender",
       x = "Possible genders") +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank())
```



Widely, distribution of pay per gender?

```
ggplot(salaries, aes(x = gender, y = annual_salary_rate)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.3, alpha = 0.2) +
  scale_y_log10(labels = scales::comma) +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank())
```


annual_salary_rate
female
male
NA
gender