

Chapter 2: Causality

Data Transformation with Tidyverse

Sho Miyazaki

Keio University

5/26/2022

- 1 Introduction
- 2 dplyr and %>%
- 3 Summary

Section 1

Introduction

Let's get started with Data

```
## load packages
library(tidyverse)
library(qss)
## load data
resume <- read_csv("data/resume.csv")
# check data
resume
```

```
## # A tibble: 4,870 x 4
##   firstname sex    race    call
##   <chr>      <chr>  <chr> <dbl>
## 1 Allison   female white    0
## 2 Kristen   female white    0
## 3 Lakisha   female black    0
## 4 Latonya   female black    0
## 5 Carrie    female white    0
## 6 Jay       male    white    0
## 7 Jill      female white    0
## 8 Kenya   female black    0
## 9 Latonya   female black    0
```

Today's Goal

Combine functions to get informative output

```
racial_gaps_by_sex <- resume %>%  
  group_by(race, sex) %>%  
    # using two variables to group the data  
  summarize(callback = mean(call)) %>%  
    # the callback rate for each group  
  pivot_wider(names_from = race,  
               # reshaping the data  
               values_from = callback) %>%  
  mutate(race_gap = white - black)
```

sex	black	white	race_gap
female	0.0662778	0.0989247	0.0326469
male	0.0582878	0.0886957	0.0304079

Tools



dplyr from Tidyverse



filter(.data, ...)

Extract rows that meet logical criteria.



select(.data, ...)

Extract columns by name.



mutate(.data, ...)

Compute new column(s).



summarise(.data, ...)

Compute table of summaries. Use **group_by()** to compute groupwise summaries.

Source: RStudio

Section 2

`dplyr` and `%>%`

What is “pipe %>%” ?



- “a good way to pronounce %>% when reading code is “**then**”.”

```
resume %>%  
  group_by(race, sex) %>%  
  summarize(callback = mean(call)) %>%  
  pivot_wider(names_from = race,  
              values_from = callback) %>%  
  mutate(race_gap = white - black)
```

```
by_race_sex <- group_by(resume, race, sex)  
resume <- summarize(by_race_sex,  
  count = n(),  
  call_back = mean(call, na.rm = TRUE))  
resume <- pivot_wider(resume,  
  names_from = race,  
  values_from = call_back)  
resume <- mutate(resume, race_gap = white - black)
```

Source: R for Data Science

Extract Rows (filter)

- filter: Return rows by name/number/etc.

```
## subset data with black names
```

```
resume %>%
```

```
  filter(race == "black")
```

```
## # A tibble: 2,435 x 4
```

```
##   firstname sex    race    call
```

```
##   <chr>      <chr>  <chr> <dbl>
```

```
## 1 Lakisha   female black     0
```

```
## 2 Latonya   female black     0
```

```
## 3 Kenya   female black     0
```

```
## 4 Latonya   female black     0
```

```
## 5 Tyrone    male    black     0
```

```
## 6 Aisha     female black     0
```

```
## 7 Aisha     female black     0
```

```
## 8 Aisha     female black     0
```

```
## 9 Tamika    female black     0
```

Extract Columns (select)

- select: Return columns by name/number/etc.

```
## Subset with sex and race columns
```

```
resume %>%
```

```
  select(sex, race)
```

```
## # A tibble: 4,870 x 2
```

```
##   sex    race
```

```
##   <chr> <chr>
```

```
## 1 female white
```

```
## 2 female white
```

```
## 3 female black
```

```
## 4 female black
```

```
## 5 female white
```

```
## 6 male    white
```

```
## 7 female white
```

```
## 8 female black
```

```
## 9 female black
```

Compute New Columns (mutate)

- mutate

```
## create a factor variable that takes one of the four values
```

```
resume %>%
```

```
  mutate(type = case_when(race == "black" & sex == "female" ~ "BlackFemale",  
                           race == "black" & sex == "male" ~ "BlackMale",  
                           race == "white" & sex == "female" ~ "WhiteFemale",  
                           race == "white" & sex == "male" ~ "WhiteMale",  
                           TRUE ~ "Other"))
```

```
## # A tibble: 4,870 x 5
```

```
##   firstname sex    race    call type  
##   <chr>      <chr> <chr> <dbl> <chr>  
## 1 Allison  female white      0 WhiteFemale  
## 2 Kristen  female white      0 WhiteFemale  
## 3 Lakisha  female black      0 BlackFemale  
## 4 Latonya  female black      0 BlackFemale  
## 5 Carrie   female white      0 WhiteFemale  
## 6 Jay      male   white      0 WhiteMale  
## 7 Jill     female white      0 WhiteFemale  
## 8 Kenya   female black      0 BlackFemale  
## 9 Latonya   female black      0 BlackFemale  
## 10 Tyrone   male   black      0 BlackMale  
## # ... with 4,860 more rows
```

Compute Table Summaries (summarise)

```
## callback rate for black female names
resume %>%
  filter(race == "black" & sex == "female") %>%
  summarize(callback_rate = mean(call, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   callback_rate
##           <dbl>
## 1           0.0663
```

Section 3

Summary

Overwhelmed?

Don't worry!

There are many resources you can use, and you don't have to memorize all the functions.

- QSS Textbook
 - Tidyverse Version is on Perusall
- Cheatsheets
 - Search “tidyverse cheatsheets”
 - <https://www.rstudio.com/resources/cheatsheets/>
- Online Resources
 - Google “tidyverse add column error”
 - official reference page, stackoverflow, RPubS, etc.

Teaching Team

We are here for you!

Let's practice!



filter(.data, ...)

Extract rows that meet logical criteria.



select(.data, ...)

Extract columns by name.



mutate(.data, ...)

Compute new column(s).



summarise(.data, ...)

Compute table of summaries. Use **group_by()** to compute groupwise summaries.

Source: RStudio