

Chapter 2: Causality

Data Transformation with Tidyverse Functions

Sho Miyazaki

Keio University

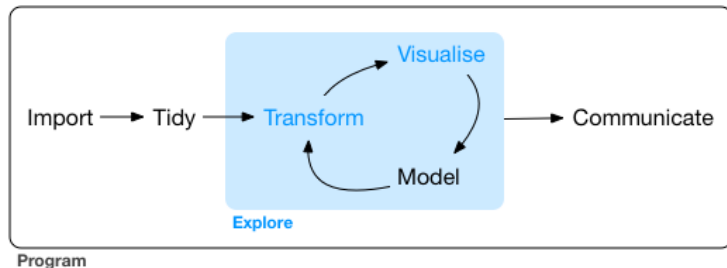
5/26/2022

- 1 Overview
- 2 Subset Data
- 3 Add New Variable
- 4 Summarize Data
- 5 Summary

Section 1

Overview

Data Transformation



Source: *R for Data Science*

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



arrange(.data, ...)

Order rows by values of a column (low to high), use with **desc()** to order from high to low.



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

Subset Data

Extract Columns (select)

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

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

```
resume_sex_race <- resume %>%  
  select(sex, race)  
resume_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
```

Extract Rows (filter)

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

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

```
resumeB <- resume %>%  
  filter(race == "black")  
resumeB
```

```
## # 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
```

Combining Functions

```
## subset data with white, male-sounding names  
## Then, let's remove the first name  
resumeWm_without_firstname <- resume %>%  
  filter(race == "white" & sex == "male") %>%  
  select(!firstname)  
resumeWm_without_firstname
```

```
## # A tibble: 575 x 3  
##   sex    race    call  
##   <chr> <chr> <dbl>  
## 1 male  white     0  
## 2 male  white     0  
## 3 male  white     0  
## 4 male  white     0  
## 5 male  white     0  
## 6 male  white     0  
## 7 male  white     0
```

Section 3

Add New Variable

Compute New Columns (mutate)

- mutate
- case_when

create a factor variable that takes one of the four values

```
resume <- 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"))  
resume
```

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

Section 4

Summarize Data

Compute Table Summaries (summarise)

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

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

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

```
## # A tibble: 1 x 1
##   callback_rate
##           <dbl>
## 1           0.0989
```

```
## difference between white and black women
Wf_callback - Bf_callback
```

```
##   callback_rate
## 1       0.03264689
```


Section 5

Summary

Overwhelmed?

Don't worry!

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

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

But more importantly...

Teaching Team

We are here for you!

- During / after the in-class exercises
- Office hours
- Perusal

Let's practice!



arrange(.data, ...)

Order rows by values of a column (low to high), use with **desc()** to order from high to low.



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