Chapter 2: Causality

Data Transformation with Tidyverse Functions

Sho Miyazaki

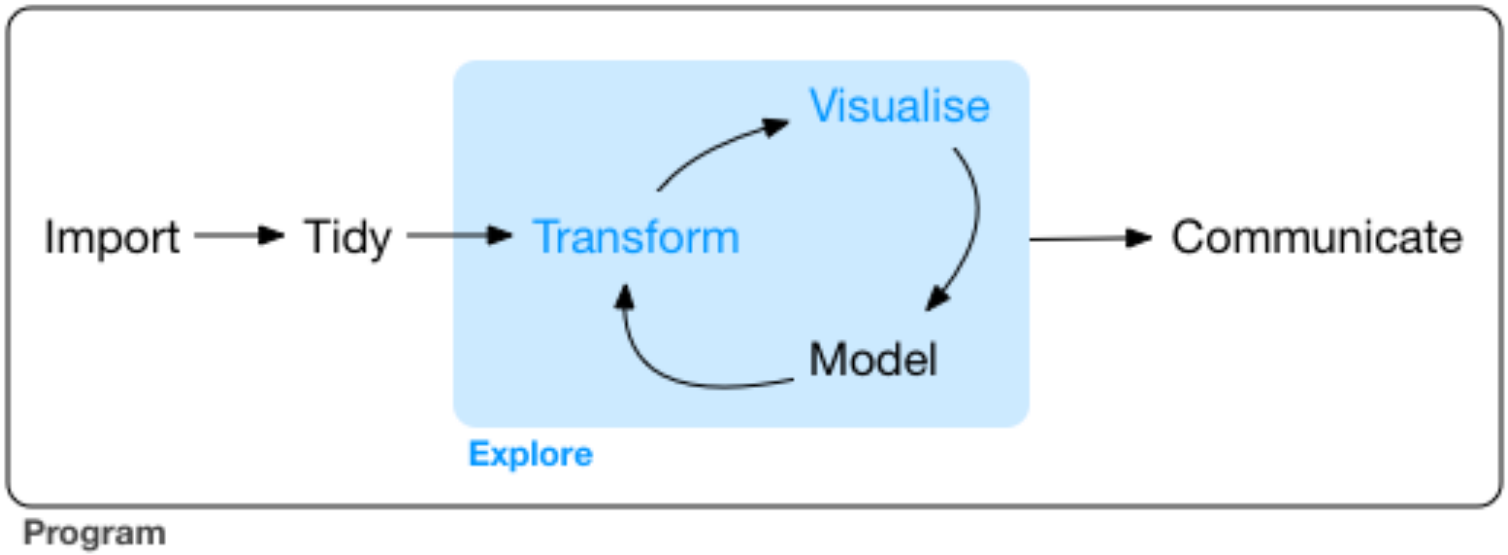
Keio University

5/26/2022

Section 1

Overview

Data Transformation



Source: R for Data Science

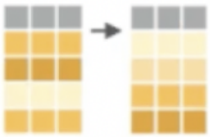

tidyr
tidyr.tidyverse.org
dplyr
dplyr.tidyverse.org

Load packages and 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
```

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`) and Rows (`filter`)

- `select`: Return columns by name/number/etc.
- `filter`: Return rows by name/number/etc.

```
## subset data with first name
resume_sex <- resume %>%
  select(sex)
head(resume_sex, 3)
```

```
## # A tibble: 3 x 1
##   sex
##   <chr>
## 1 female
## 2 female
## 3 female
```

```
## subset data with black names
resumeB <- resume %>%
  filter(race == "black")
head(resumeB, 3)
```

```
## # A tibble: 3 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
```

Combining Functions

```
## subset data with black, female-sounding names
## Then, let'ss remove the firstname
resumeBf_without_firstname <- resume %>%
  filter(race == "black" & sex == "female") %>%
  select(!firstname)
resumeBf_without_firstname
```

```
## # A tibble: 1,886 x 3
##    sex    race   call
##    <chr>  <chr> <dbl>
##  1 female black     0
##  2 female black     0
##  3 female black     0
##  4 female black     0
##  5 female black     0
##  6 female black     0
##  7 female black     0
```

Section 3

Summarize Data

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 4

Add New Variable

mutate()

calculate target values

The way we did previously with `filter()` and `summarise()`.

create factor variable with mutate

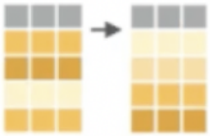
```
## create a factor variable that takes one of the four values
resume <- resume %>%
  mutate(type = case_when(race == "black" & sex == "female" ~ "BlackFemale",
                          race == "black" & sex == "male" ~ "BlackMale",
                          race == "white" & sex == "female" ~ "WhiteFemale",
                          race == "white" & sex == "male" ~ "WhiteMale",
                          TRUE ~ "Other"))

head(resume)
```

```
## # A tibble: 6 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
```

Section 5

Summary

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