Class 4: Grammar of data analysis - Part II

API-201
Quantitative Analysis and Empirical Methods I

Profs. Goel and Taylor
Harvard Kennedy School

Agenda

1. Review the five verbs of data analysis `[tidyverse]`
2. Introduce `joins`
3. Continue applying this language to study police stops

Five verbs of data manipulation

- **`filter`**
- **`select`**
- **`mutate`**
- **`summarize`**
- **`arrange`**

The `group_by` operation

The five verbs of data manipulation are particularly powerful when we combine them with the `group_by` operation.

Select the two appropriate commands

You have a data frame of grant recipients. Each row contains the grantee's name, the funding area (e.g., education, health, climate, etc.), and funding amount.

You would like to compute the total amount of funding distributed to each area.

Which two commands would you use?

- filter and summarize
- group_by and mutate
- group_by and summarize
- select and summarize

Joining datasets

`https://r4ds.hadley.nz/joins.html`

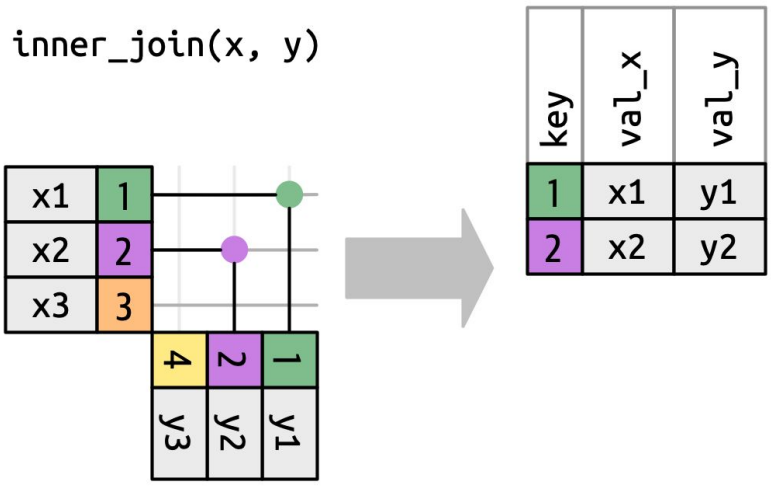
In many cases, the data we want to analyze is split across multiple data frames.

For example, police stops may be stored in one table but information about the involved officers may be stored in another.

`Joins` let us combining columns from multiple data frames

Joining datasets [*inner* join]

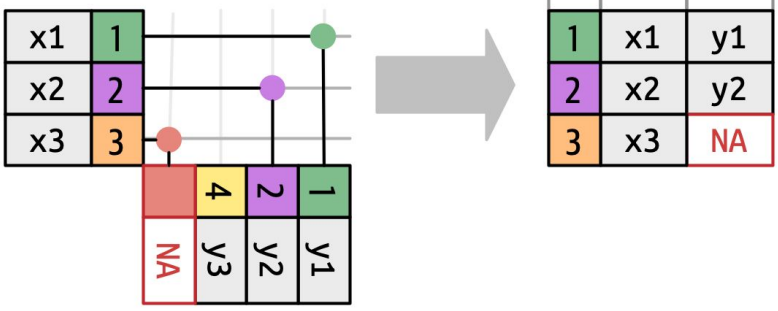
`https://r4ds.hadley.nz/joins.html`



Joining datasets [*left* join]

https://r4ds.hadley.nz/joins.html

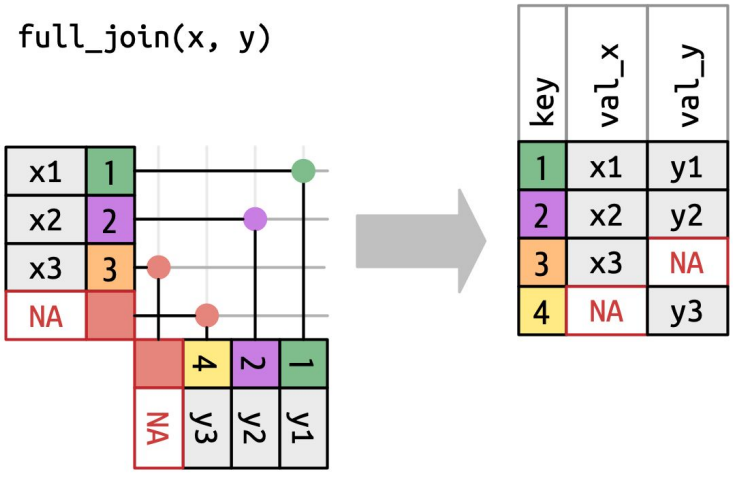
left_join(x, y)



Joining datasets [*full* join]

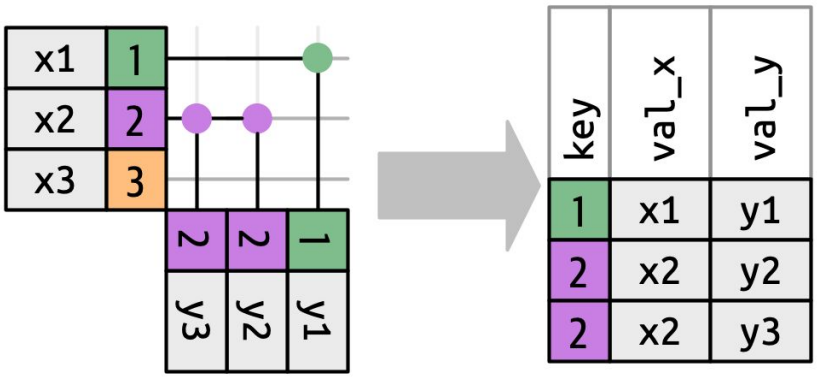
https://r4ds.hadley.nz/joins.html

full_join(x, y)



Joining datasets [duplicates]

https://r4ds.hadley.nz/joins.html



Select the appropriate type of join

0

One data frame contains household income for every resident of Boston, with a unique person identifier.

Another data frame lists the set of individuals on government-sponsored health insurance across Massachusetts.

You would like to compute the proportion of Boston residents in each income bracket receiving the insurance.

Which type of join would you do?

- Inner join **(A)**
- Left join **(B)**
- Full join **(C)**

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

`if_else()`

The command **`if_else`** lets us select between different values depending on some *test*.

`if_else` takes three parameters:

- A ***test*** vector of TRUE/FALSE values
- A ***yes*** value to return when the corresponding entries in the TRUE/FALSE vector are `TRUE`
- A ***no*** value to return when the corresponding entries in the TRUE/FALSE vector are `FALSE`

`if_else()`

```
my_vec <- c(-1, 7, -3, 1, 5)

if_else(my_vec > 0, my_vec*2, 0)
```

Exercise

`residents` has two columns: **`person_id`** and **`income_bracket`**

`insurance` has two columns: **`person_id`** and **`subsidy`**

Your goal is to compute the average subsidy for individuals in each income bracket. This average should be computed over all individuals, not just those who received a subsidy. [Discuss with your neighbor.]

```
residents %>%
   ________(insurance, by = "________") %>%
   mutate(subsidy = if_else(is.na(subsidy), 0, ________)) %>%
   ________(income_bracket) %>%
   ________(avg_subsidy = ________(subsidy))
```

Key ideas

The **five verbs** of data manipulation — plus the **`group_by`** operation — allow us to carry out sophisticated descriptive statistical analyses.

Joins allows us to analyze complex patterns spread across multiple datasets.

Putting it all together

Now we'll use what we've learned to continue investigating racial discrimination in police stops.

`https://bit.ly/API201-dplyr`