Video 22: dplyr - joins

Stats 102A

Miles Chen, PhD

Two-table verbs

Two-table verbs

dplyr also comes with some two-table verbs that allow you to combine tables.

Mutating joins add new variables to one table from matching rows in another

Not covered here, but you can read more at https://dplyr.tidyverse.org/articles/two-table.html

- **Filtering joins** filter observations from one table based on whether or not they match an observation in the other.
- **Set operations**, which combine the observations in the data sets as if they were set elements.

Toy tables

```r
people <- tibble(
  name = c("Adam", "Betty", "Carl", "Doug"),
  state = c("CA", "CA", "NY", "TX")
  )
states <- tibble(
  abbreviation = c("CA", "NY", "WA"),
  state_name = c("California", "New York", "Washington")
  )
```

left_join()

`left_join()` takes all the values in the left table and adds variables from the right table by matching values using a column that exists in both tables. Values that do not exist in the other table have NA returned.

```
people %>% left_join(states, by = c("state" = "abbreviation"))
```

```
# A tibble: 4 × 3
  name  state state_name
  <chr> <chr> <chr>
1 Adam  CA    California
2 Betty CA    California
3 Carl  NY    New York
4 Doug  TX    <NA>
```

right_join()

`right_join()` is similar to `left_join` except it keeps all the rows in the right table.

```
people %>% right_join(states, by = c("state" = "abbreviation"))
```

```
# A tibble: 4 × 3
  name  state state_name
  <chr> <chr> <chr>
1 Adam  CA    California
2 Betty CA    California
3 Carl  NY    New York
4 <NA>  WA    Washington
```

inner_join()

`inner_join()` keeps only rows that have values that exist in both tables. You can think of this as the intersection.

```
1 people %>% inner_join(states, by = c("state" = "abbreviation"))
```

```
# A tibble: 3 × 3
  name  state state_name
  <chr> <chr> <chr>
1 Adam  CA    California
2 Betty CA    California
3 Carl  NY    New York
```

full_join()

`full_join()` keeps all rows from both tables. You can think of this as the union. (in SQL this is called a full outer join)

```
1 people %>% full_join(states, by = c("state" = "abbreviation"))
```

```
# A tibble: 5 × 3
  name  state state_name
  <chr> <chr> <chr>
1 Adam  CA    California
2 Betty CA    California
3 Carl  NY    New York
4 Doug  TX    <NA>
5 <NA>  WA    Washington
```

Controlling how the tables are matched

Depending on the tables, the join operation can match tables on different variables.

In the previous examples, we used a named character vector `by = c("state" = "abbreviation")` specifying the name in the left table that matches the name in the right table.

Options for joining tables

- `by = NULL` (or don't specify anything): `dplyr` will use all variables that have the same name in both tables.
- `by = "x"`: dplyr will use only some of the variables that have the same name in both tables
- `by = c("x" = "y")`: this is the form that must be used if the matching columns do not have the same name in both tables.