Getting / cleaning data 2

Joining datasets

Joining datasets

So far, you have only worked with a single data source at a time. When you work on your own projects, however, you typically will need to merge together two or more datasets to create the a data frame to answer your research question.

For example, for air pollution epidemiology, you will often have to join several datasets:

- Health outcome data (e.g., number of deaths per day)
- Air pollution concentrations
- Weather measurements (since weather can be a confounder)
- Demographic data

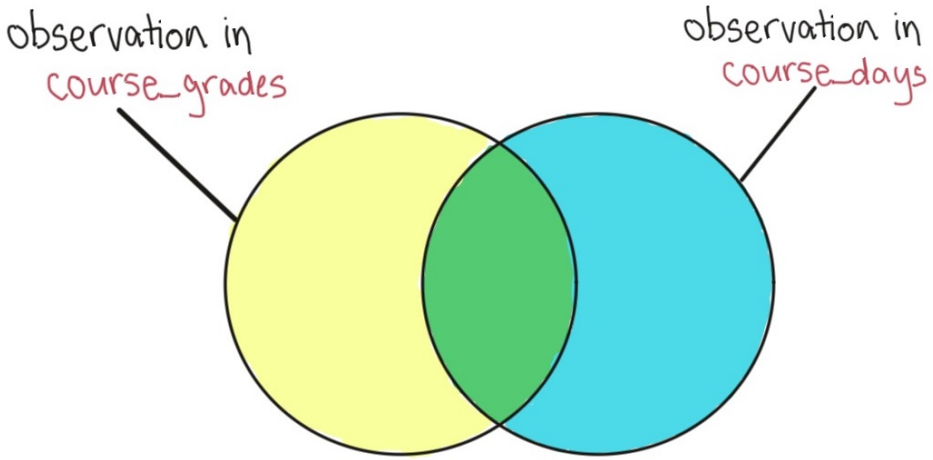
`*_join` functions

The `dplyr` package has a family of different functions to join two dataframes together, the `*_join` family of functions. These include:

- `inner_join`
- `full_join`
- `left_join`
- `right_join`

All combine two dataframes, which I'll call `course_grades` and `course_days` here.

`*_join` functions



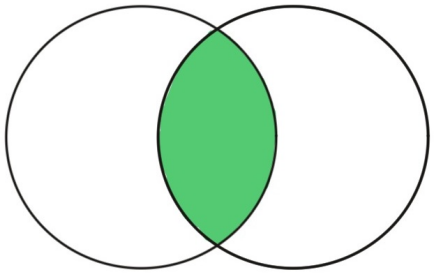
course_grades

course	grade
Math	90
Science	82
English	78

course_days

course	day
Math	Mon
English	Thur
Art	Tue/Wed

inner_join



inner_join

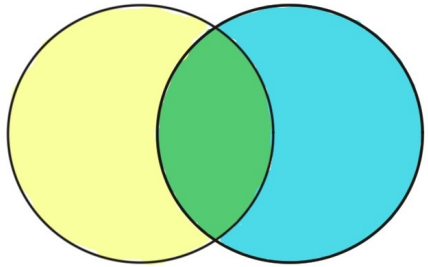
course	grade
Math	90
Science	82
English	78

course	day
Math	Mon
English	Thur
Art	Tue/Wed

inner_join(course_grades, course_days, by = "course")

course	grade	day
Math	90	Mon
English	78	Thur

full_join



full_join

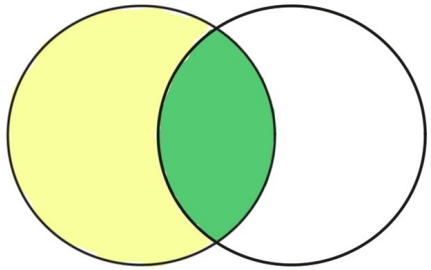
course	grade
Math	90
Science	82
English	78

course	day
Math	Mon
English	Thur
Art	Tue/Wed

full_join(course_grades, course_days, by = "course")

course	grade	day
Math	90	Mon
Science	82	NA
English	78	Thur
Art	NA	Tue/Wed

left_join



left_join

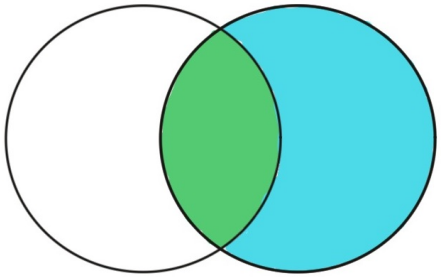
course	grade
Math	90
Science	82
English	78

course	day
Math	Mon
English	Thur
Art	Tue/Wed

left_join(course_grades, course_days, by = "course")

course	grade	day
Math	90	Mon
Science	82	NA
English	78	Thur

right_join



right_join

course	grade
Math	90
Science	82
English	78

course	day
Math	Mon
English	Thur
Art	Tue/Wed

right_join(course_grades, course_days, by = "course")

course	grade	day
Math	90	Mon
English	78	Thur
Art	NA	Tue/Wed

*_join functions

For some more complex examples of using join, I'll use these example datasets (x and y):

```
## # A tibble: 4 x 3
##   course grade student
##   <chr>  <dbl> <chr>
## 1 x         92 a
## 2 x         90 b
## 3 y         82 a
## 4 z         78 b
```

```
## # A tibble: 4 x 3
##   class day       student
##   <chr> <chr>     <chr>
## 1 w     Tues      a
## 2 x     Mon / Fri a
## 3 x     Mon / Fri b
## 4 y     Tue       a
```

*_join functions

If you have two datasets you want to join, but the column names for the joining column are different, you can use the by argument:

```
full_join(x, y, by = list(x = "course", y = "class"))
```

```
## # A tibble: 7 x 5
##   course grade student.x day       student.y
##   <chr>  <dbl> <chr>     <chr>     <chr>
## 1 x         92 a         Mon / Fri a
## 2 x         92 a         Mon / Fri b
## 3 x         90 b         Mon / Fri a
## 4 x         90 b         Mon / Fri b
## 5 y         82 a         Tue       a
## 6 z         78 b         <NA>      <NA>
## 7 w         NA <NA>      Tues      a
```

`*_join` functions

A few things to note about this example:

- The joining column name for the "left" dataframe (`x` in this case) is used as the column name for the joined data
- `student` was a column name in both `x` and `y`. If we're not using it to join the data, the column names are changed in the joined data to `student.x` and `student.y`.
- Values are recycled for rows where there were multiple matches across the dataframe (e.g., rows for course "x")

*_join functions

Sometimes, you will want to join by more than one column. In this example data, it would make sense to join the data by matching both course and student. You can do this by using a vector of all columns to join on:

```
full_join(x, y, by = list(x = c("course", "student"),
                          y = c("class", "student")))
```

```
## # A tibble: 5 x 4
##   course grade student day
##   <chr>  <dbl> <chr>   <chr>
## 1 x         92 a       Mon / Fri
## 2 x         90 b       Mon / Fri
## 3 y         82 a       Tue
## 4 z         78 b       <NA>
## 5 w         NA a       Tues
```