Math 300 Lesson 7 Notes

join, select, rename, and top_n

Professor Bradley Warner

June, 2022

Contents

Objectives 1

Reading 1

Lesson 1

Documenting software 4

Objectives

1. Use the `inner_join()` function to combine data frames in order to explore, explain, and visualize.
2. Explain how to join data frame to include the use of keys and the advantages and the disadvantages of normal forms.
3. Use `rename()` and `select()` to reorganize data frames in order to explore, explain, and visualize. This includes all the ways to select columns including helper functions such as `everthing()`, `contains()`, etc.
4. Use `top_n()` to select a subset of a data frame.

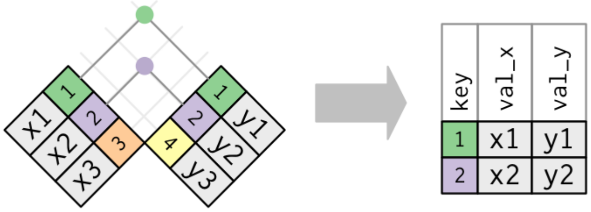
Reading

Chapter 3.7 - 3.9

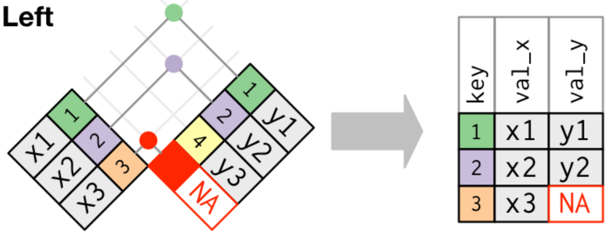
Lesson

Work through the learning checks LC3.13 - LC3.20. Complete the code as necessary.

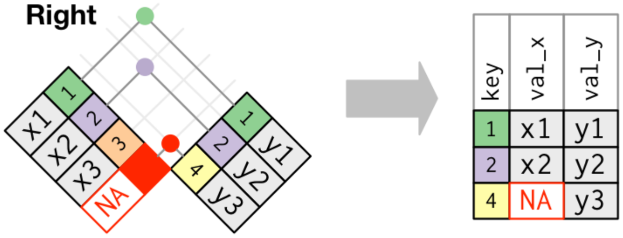
- We are using an inner join. The important ideas are the **key** variable to join the data frames and the type of join. Figure 3.8 shows an inner join. It only keeps observations that are in each data frame. See R for Data Science for more information to include a discussion of outer joins. The following figures can help.



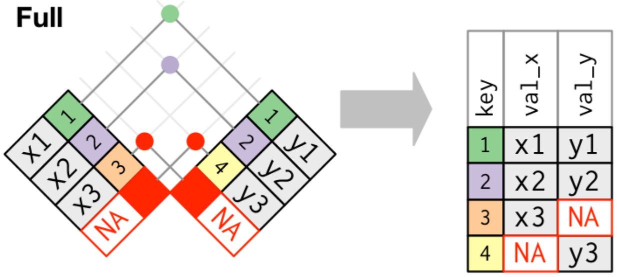
Left



Right



Full



- The *keys* don't have to be called the same name and as such will require the use of the `by` option.
- The use of `everthing()`, `contains()`, `starts_with()`, and `ends_with()` makes the use of `select()` easier.
- `rename()` creates the new variables and deletes the old. If we used `mutate()` it would keep both variables and we would then need to use `select()`.
- Likewise, `top_n()` could be achieved with `arrange()` and `head()`.

Setup

```
library(nycflights13)
library(ggplot2)
library(dplyr)
```

LC 3.13 (Objective 2)

(LC 3.13) Looking at Figure 3.7, when joining `flights` and `weather` (or, in other words, matching the hourly weather values with each flight), why do we need to join by all of `year`, `month`, `day`, `hour`, and `origin`, and not just `hour`?

Solution:

LC 3.14 (Objective 1, 3)

(LC 3.14) Recreate the data object `named_dests` from the reading. What surprises you about the top 10 destinations from NYC in 2013?

Solution:

We need to recreate the data object

```r
# Complete the code and then remove the comment symbols
# named_dests <- _______ %>%
#   group_by(dest) %>%
#   summarize(num_flights = ____) %>%
#   arrange(desc(_________)) %>%
#   inner_join(airports, by = c("dest" = "faa")) %>%
#   rename(airport_name = name)
```

LC 3.15 (Objective 2)

(LC 3.15) What are some advantages of data in normal forms? What are some disadvantages?

Solution:

LC 3.16 (Objective 3)

(LC 3.16) What are some ways to select all three of the `dest`, `air_time`, and `distance` variables from `flights`? Give the code showing how to do this in at least three different ways.

Solution:

LC 3.17 (Objective 3)

(LC 3.17) How could one use `starts_with`, `ends_with`, and `contains` to select columns from the `flights` data frame? Provide three different examples in total: one for `starts_with`, one for `ends_with`, and one for `contains`.

Solution:

```r
# Anything that starts with "d"
```

```r
# Anything ending in delay:
```

```r
# Anything containing dep:
```

LC 3.18 (Objective 3)

(LC 3.18) Why might we want to use the `select()` function on a data frame?

Solution:

LC 3.19 (Objective 4)

(LC 3.19) Create a new data frame that shows the top 5 airports with the largest average arrival delays from NYC in 2013.

Solution:

```
# Complete the code and then remove the comment symbols
# top_five <- flights %>%
#   group_by(_________) %>%
#   summarize(avg_delay = ________(arr_delay, na.rm = TRUE)) %>%
#   arrange(desc(________)) %>%
#   top_n(n = _______)
```

LC 3.20 (Many objectives of Chapter 3)

(LC 3.20) Using the datasets included in the `nycflights13` package, compute the available seat miles for each airline sorted in descending order. After completing all the necessary data wrangling steps, the resulting data frame should have 16 rows (one for each airline) and 2 columns (airline name and available seat miles). Here are some hints:

- **Crucial**: Unless you are very confident in what you are doing, it is worthwhile to not starting coding right away, but rather first sketch out on paper all the necessary data wrangling steps not using exact code, but rather high-level *pseudocode* that is informal yet detailed enough to articulate what you are doing. This way you won't confuse *what* you are trying to do (the algorithm) with *how* you are going to do it (writing `dplyr` code).
- Take a close look at all the datasets using the `View()` function: `flights`, `weather`, `planes`, `airports`, and `airlines` to identify which variables are necessary to compute available seat miles.
- Figure 3.7 above showing how the various datasets can be joined will also be useful.
- Consider the data wrangling verbs in Table 3.2 as your toolbox!

Solution:

Psuedo code

Documenting software

- File creation date: 2022-06-16
- R version 4.1.3 (2022-03-10)
- `ggplot2` package version: 3.3.6
- `dplyr` package version: 1.0.9
- `nycflights13` package version: 1.0.2