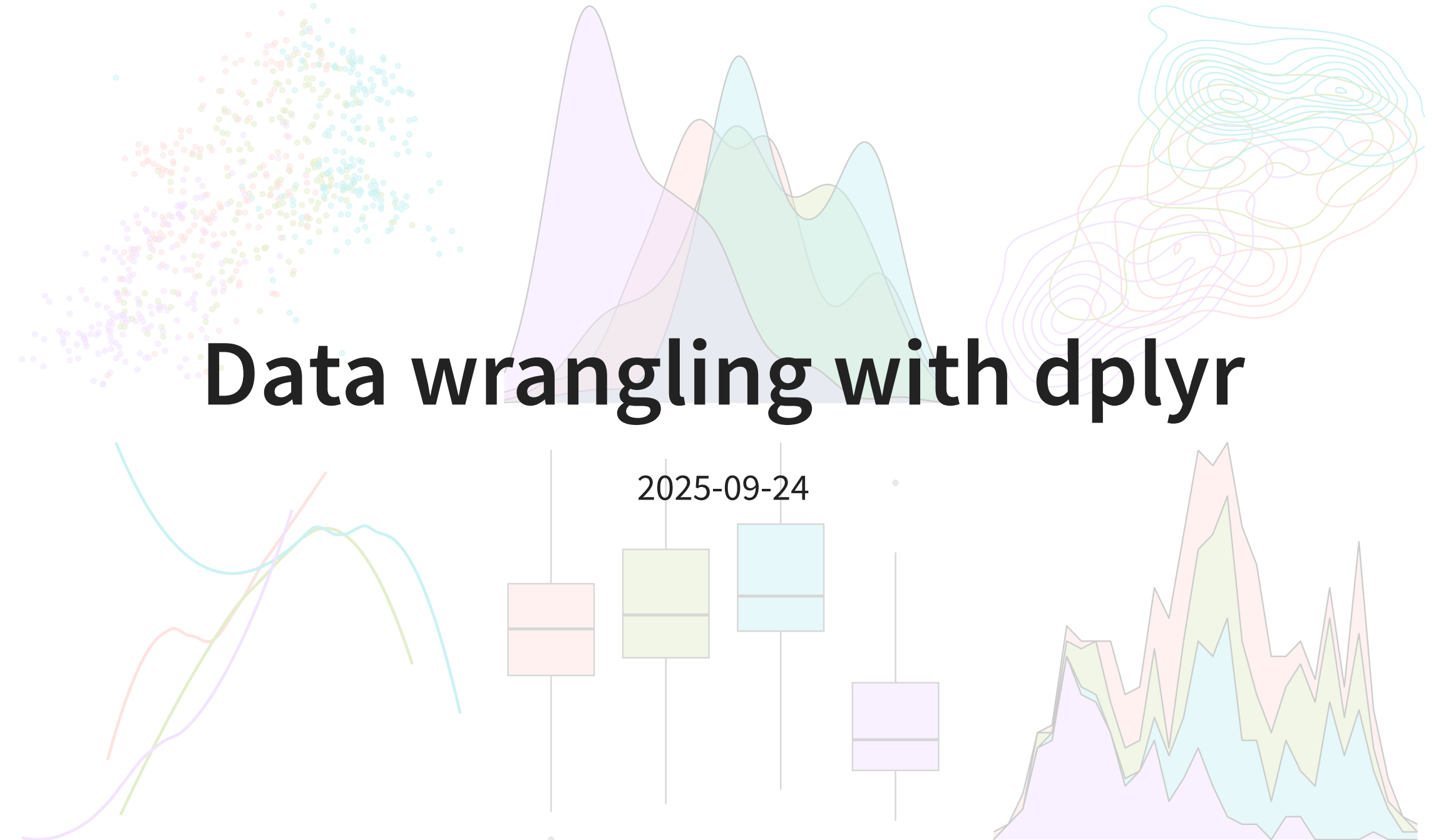


Data wrangling with dplyr

2025-09-24

Grammar of data wrangling

- Recall: data frames are objects in R that store tabular data in tidy form
- The `dplyr` package (included in `tidyverse` package) uses the concept of functions as verbs that manipulate data frames.
  - `filter()`: pick rows matching criteria
  - `mutate()`: add new variables as columns
  - `summarise()`: reduce variables to quantitative values
  - `group_by()`: for grouped operations based on a variable
  - `distinct()`: filter for unique rows
  - `select()`: pick columns by name
  - `slice()`: pick rows using indices
  - and many more!!!

Rules of dplyr functions

1. The first argument is *always* a data frame
2. Subsequent argument(s) say what to do with that data frame
    i. We connect lines to code using a *pipe* operator (see next slide)
3. *Always* return a data frame

Pipes

- In programming, a **pipe** is a technique for passing information from one process to another
- In `dplyr`, the pipes are coded as `|>` (i.e. vertical bar and greater than sign)
  - Not to be confused with `+` used to add layers in `ggplot`
- We can think about pipes as following a sequence of actions which provide a more natural and easier to read structure
- For example: suppose that in order to get to work, I need to find my car keys, start my car, drive to work, and then park my car

- Expressed using pipes, this may look like:

```
find("car_keys") |>
  start_car() |>
  drive(to = "work") |>
  park()
```

- Expressed as a set of nested R pseudocode, this may look like:

```
park(drive(start_car(find("car_keys")),
           to = "work"))
```

Logical operators in R

It is common to compare two quantities using logical operators. All of these operators will return a **logical** TRUE or FALSE. List of some common operators:

- <: less than
- <=: less than or equal to
- >: greater than
- >=: greater than or equal to
- ==: (exactly) equal to
- !=: not equal to

```
1 1 < 4
```

```
[1] TRUE
```

```
1 2 == 5
```

```
[1] FALSE
```

```
1 2 != 5
```

```
[1] TRUE
```

Logical operators (cont.)

We might also want to know if a certain quantity "behaves" a certain way. The following also return logical outputs:

- `is.na(x)`: test if `x` is `NA`
- `x %in% y`: test if `x` is in `y`
- `!x`: not `x`

```
1 is.na(NA)
```

```
[1] TRUE
```

```
1 is.na("apple")
```

```
[1] FALSE
```

```
1 3 %in% 1:10
```

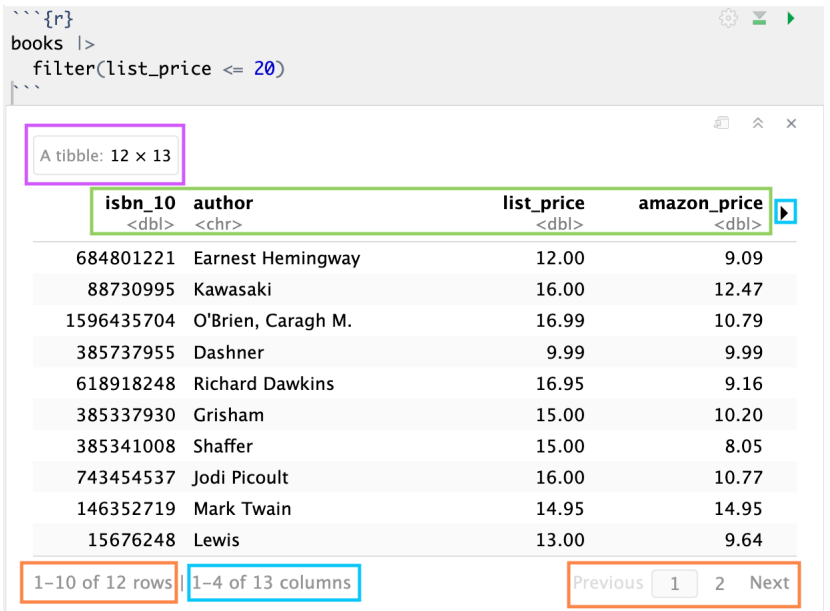
```
[1] TRUE
```

```
1 !TRUE
```

```
[1] FALSE
```

Working with data frames in RStudio

If executed code output in **Source**



```{r}
books |>
  filter(list_price <= 20)
```

A tibble: 12 × 13

isbn_10 <dbl>	author <chr>	list_price <dbl>	amazon_price <dbl>
684801221	Earnest Hemingway	12.00	9.09
88730995	Kawasaki	16.00	12.47
1596435704	O'Brien, Caragh M.	16.99	10.79
385737955	Dashner	9.99	9.99
618918248	Richard Dawkins	16.95	9.16
385337930	Grisham	15.00	10.20
385341008	Shaffer	15.00	8.05
743454537	Jodi Picoult	16.00	10.77
146352719	Mark Twain	14.95	14.95
15676248	Lewis	13.00	9.64

1–10 of 12 rows | 1–4 of 13 columns Previous 1 2 Next

If executed code output in **Console**

```
> books |>
+   filter(list_price <= 20)
# A tibble: 12 × 13
      isbn_10 author  list_price amazon_price num_pages cover pub_year height
        <dbl> <chr>        <dbl>        <dbl>     <dbl> <chr>    <dbl>  <dbl>
 1  684801221 Earnes…       12           9.09       128 P         1995    5.2
 2   88730995 Kawasa…       16          12.5        224 P         2000    8.2
 3 1596435704 O'Brie…       17.0        10.8        368 H         2011   NA
 4  385737955 Dashner        9.99        9.99       400 P         2010    8.2
 5  618918248 Richar…       17.0         9.16       464 P         2008    8.1
 6  385337930 Grisham       15          10.2        384 P         2004    8
 7  385341008 Shaffer       15           8.05       290 H         2009    7.8
 8  743454537 Jodi P…       16          10.8        448 P         2005    8.2
 9  146352719 Mark T…       15.0        15.0        276 P         2011    7.8
10   15676248 Lewis         13           9.64       168 P         1964    8
11 1565125606 Gruen        15.0         8.14       350 P         2007    8.2
12   60731338 Steven…      16.0         9.95       315 P         2009    7.9
# ℹ 5 more variables: width <dbl>, thick <dbl>, weight_oz <dbl>,
#   title <chr>, publisher <chr>
```

- Tibble (i.e. data frame) with 12 observations and 13 variables
- For variables shown, their names and types
  - Variables not displayed. In Source, you can click to see other variables.
- Source will display at most 10 observations, but you can click to see more.

Live code

Data from Amazon: we have data about several books available for purchase from Amazon. I took a random sample from the original sample of 325 cases from the original dataset.

Copy and paste the following line of code into a new code chunk in your live code! We will load in the data together and take a quick look at it before diving into data wrangling

```
1 url_file <- "https://raw.githubusercontent.com/midd-stat201a-fall25/midd-stat201a-fall25.github.io/re
```