Manipulating and Visualizing Data

Why do we need an extra week on this?

Base `R` is limited for data manipulation and visualization

It can be:

- Hard to read, especially across languages
- Inefficient (many lines of code to perform a common operation)

What is Base R

"Base R" = What is available in R with no added packages.

Note: There are benefits to Base R.

- "Lightweight" (no additional time/space to install)
- Especially stable across updates

What is the alternative?

"Tidyverse" describes a suite of `R` packages supported by Posit

Information at: https://tidyverse.org

Associated with "R for Data Science": https://r4ds.hadley.nz

Install it!

What will we learn about the "Tidyverse"?

We will learn about:

- `dplyr`: Helps filter and aggregate data
- `ggplot2`: Helps create nice plots

Getting Ready to Explore “dplyr“‘

We’re going to use `dplyr` to summarize data on flights that departed New York City from JFK, LGA, or EWR in 2013.

```
library(nycflights13)
library(dplyr)
flights <- nycflights13::flights
```

The Flights Data is a tibble

```
str(flights)
```

```
tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
 $ year          : int [1:336776] 2013 2013 2013 2013 2013
 $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
 $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time      : int [1:336776] 517 533 542 544 554 554 5
 $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 6
 $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -
3 -3 -2 ...
 $ arr_time      : int [1:336776] 830 850 923 1004 812 740
 $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728
 $ arr_delay     : num [1:336776] 11 20 33 -18 -
25 12 19 -14 -8 8 ...
 $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
 $ flight        : int [1:336776] 1545 1714 1141 725 461 16
 $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA
 $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK"
```

Converting the Flights Data to a Data Frame

To avoid learning about a new data structure, we'll convert the `tibble` to a data frame.

```
flights <- as.data.frame(flights)
```

The `filter` Function

Allows us to easily subset data according to the values that a variable takes on

```
filter(flights, month == 1, day == 1)
```

Previously we did this by indexing using logical vectors

```
flights[flights$month == 1 &
          flights$day == 1, ]
```

Filtering on Multiple Values

Just like we saw before when learning about logical vectors, there's multiple ways to do this.

```
filter(flights, month == 11 | month == 12)
filter(flights, month %in% c(11, 12))
```

Missing Value Handling

Missing values (`NA`) are handled in natural ways by `filter`

```
filter(flights,
       !(arr_delay > 120 |
           dep_delay > 120))
```

Before, we would have needed to type

```
flights[!is.na(flights$arr_delay) &
          !is.na(flights$dep_delay) &
          flights$arr_delay <= 120 &
          flights$dep_delay <= 120, ]
```

Filtering Using Intervals

The `between` function with `left` and `right` arguments allow us to filter on variables being in an interval.

```
filter(flights, arr_delay <= 120,
       between(x = dep_delay, left = 80, right = 120))
```

This is much uglier with base `R`!

```
flights[!is.na(flights$arr_delay) &
          !is.na(flights$dep_delay) &
          flights$arr_delay <= 120 &
          flights$dep_delay >= 80 &
          flights$dep_delay <= 120, ]
```

Letting Missing Values Through

Sometimes, we want to retain some observations with missing values.

```
filter(flights, is.na(arr_delay) |
         arr_delay <= 120,
       between(x = dep_delay, left = 80, right = 120))
```

In base `R` this would have been

```
flights[!is.na(flights$dep_delay) &
          flights$arr_delay <= 120 &
          flights$dep_delay >= 80 &
          flights$dep_delay <= 120, ]
```

Activities to Try Yourself

Find all flights that were:

- Delayed two or more hours
- Flew to Houston
- Were operated by United, American, or Delta
- Departed in summer (July, August, and September)
- Arrived more than two hours late, but didn't leave late
- Delayed at least an hour, but made up over 30 minutes in flight
- Departed between midnight and 6:00AM (inclusive)