lecture 2

Leah Brooks

January 25, 2021

Today

A. What is Merging?

B. How to Merge 1:1

C. How to Merge Many to 1

D. Cautions with merging

A. Merging

- if you have information in more than one dataframe
- you want to combine these pieces of information
- reliably and replicably
- this is an **enormous** advantage of statistical software

Examples of When You Need to Merge

Ex. 1:

- you have a dataset on crimes, with addresses
- you want to add the neighborhood median income
- $\rightarrow$ merge by neighborhood id!

Examples of When You Need to Merge

Ex. 1:

- you have a dataset on crimes, with addresses
- you want to add the neighborhood median income
- $\rightarrow$ merge by neighborhood id!

Ex. 2:

- you have a dataset of student performance
- you want to add information on teacher
- $\rightarrow$ merge by teacher id!

Merging Command Overview

```
merge(x = data.frame.1,
      y = data.frame.2,
      by = "varname",
      all = TRUE)
```

Merging Command Overview

```
merge(x = data.frame.1,
      y = data.frame.2,
      by = "varname",
      all = TRUE)
```

Now a very simple example

Sample dataframe 1: Class subjects

```
df1 <- data.frame(class = c(1,2,3),
                  subject = c("basics","basics","graphs"))
df1
```

```
##   class subject
## 1     1  basics
## 2     2  basics
## 3     3  graphs
```

Sample dataframe 2: Class attendance

```
df2 <- data.frame(class = c(1,2,3),
                  attendance = c(33,45,26))
df2
```

```
##   class attendance
## 1     1         33
## 2     2         45
## 3     3         26
```

B. Merge 1:1

```
df3 <- merge(x = df1,
             y = df2,
             by = "class",
             all = TRUE)
```

How many rows should d3 have?

B. Merge 1:1

```
df3 <- merge(x = df1,
             y = df2,
             by = "class",
             all = TRUE)
```

How many rows should d3 have?

```
df3
```

```
##   class subject attendance
## 1     1  basics         33
## 2     2  basics         45
## 3     3  graphs         26
```

C. Merge Many to 1

Many to 1 merge:

- this is a merge that has unique values in one dataset
- and repeat values in another

C. Merge Many to 1

Many to 1 merge:

- this is a merge that has unique values in one dataset
- and repeat values in another

Unique and repeat values:

- unique values: class in `df3`
- repeat values: `subject` in `df3`

```
df3
```

```
##   class subject attendance
## 1     1  basics         33
## 2     2  basics         45
## 3     3  graphs         26
```

Dataset to merge in

```
df4 <- data.frame(subject = c("basics","graphs"),
                  difficulty = c("easy","hard"))
df4
```

```
##   subject difficulty
## 1  basics       easy
## 2  graphs       hard
```

Merging in

```
df5 <- merge(x = df3,
             y = df4,
             by = "subject",
             all = TRUE)
```

How many rows should this have?

Merging in

```
df5 <- merge(x = df3,
             y = df4,
             by = "subject",
             all = TRUE)
```

How many rows should this have?

```
df5
```

```
##   subject class attendance difficulty
## 1  basics     1         33       easy
## 2  basics     2         45       easy
## 3  graphs     3         26       hard
```

D. Frequent Problems with Merging

- you want to merge 1:1 but one dataframe has repeat values

D. Frequent Problems with Merging

- you want to merge 1:1 but one dataframe has repeat values
- you want to merge 1:1 but the merge doesn't work as expected (see tutorial)

D. Frequent Problems with Merging

- you want to merge 1:1 but one dataframe has repeat values
- you want to merge 1:1 but the merge doesn't work as expected (see tutorial)

Why worry?

D. Frequent Problems with Merging

- you want to merge 1:1 but one dataframe has repeat values
- you want to merge 1:1 but the merge doesn't work as expected (see tutorial)

Why worry?

- bad merges yield garbage
- garbage in $\rightarrow$ garbage out