Functions

Warmup activity

Work on the activity (handout) with a neighbor, then we will discuss as a class

Warmup

```
z_score <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

diamonds_new <- diamonds |>
  mutate(carat_z = z_score(carat))
```

What does this code do?

Warmup

```r
z_score <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

diamonds_new <- diamonds |>
  mutate(carat_z = z_score(carat))
```

The `diamonds` dataset has 53940 rows and 10 columns.

What will be the dimensions of the `diamonds_new` dataset?

Warmup

```r
z_score <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

diamonds_new <- diamonds |>
  mutate(carat_z = z_score(carat))

glimpse(diamonds_new)
```

```
Rows: 53,940
Columns: 11
$ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22,
0.23, 0.…
$ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very
Good, Ver…
$ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J,
J, J, I,…
$ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1,
SI1, VS1, …
$ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1,
59.4, 64…
```

Functions

name of function

arguments to function

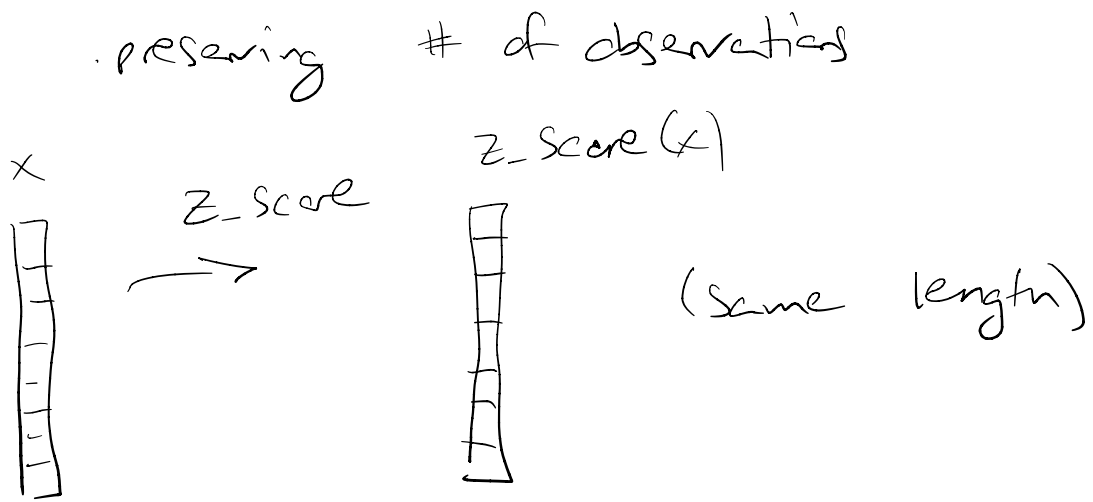
```
1 z_score <- function(x) {
2   (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
3 }
```

body of function (what function does)

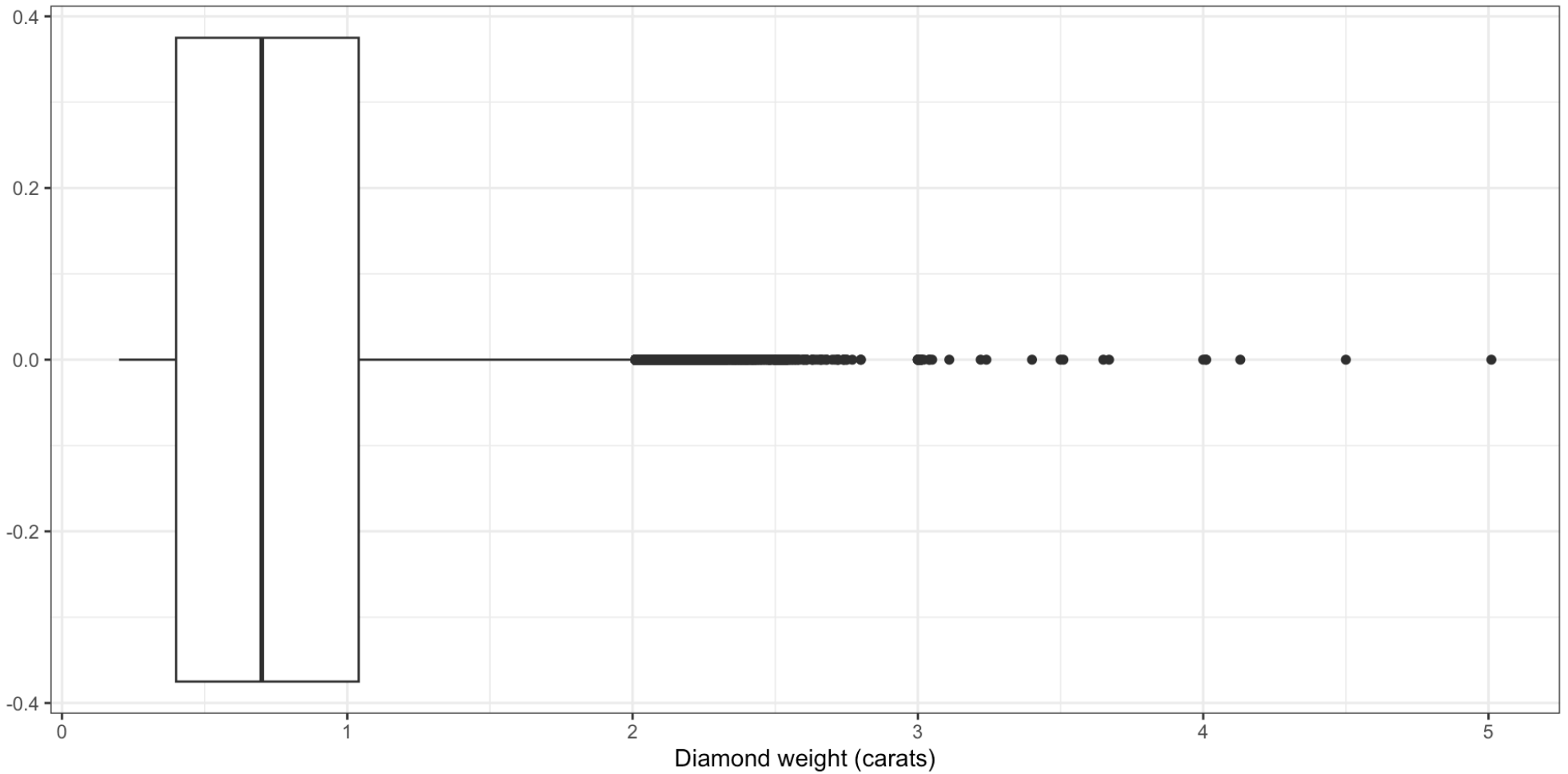
Functions

```r
z_score <- function(x) {
  (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)
}

diamonds_new <- diamonds |>
  mutate(carat_z = z_score(carat))
```

R for Data Science calls the `z_score` function a "mutate" function. Why?

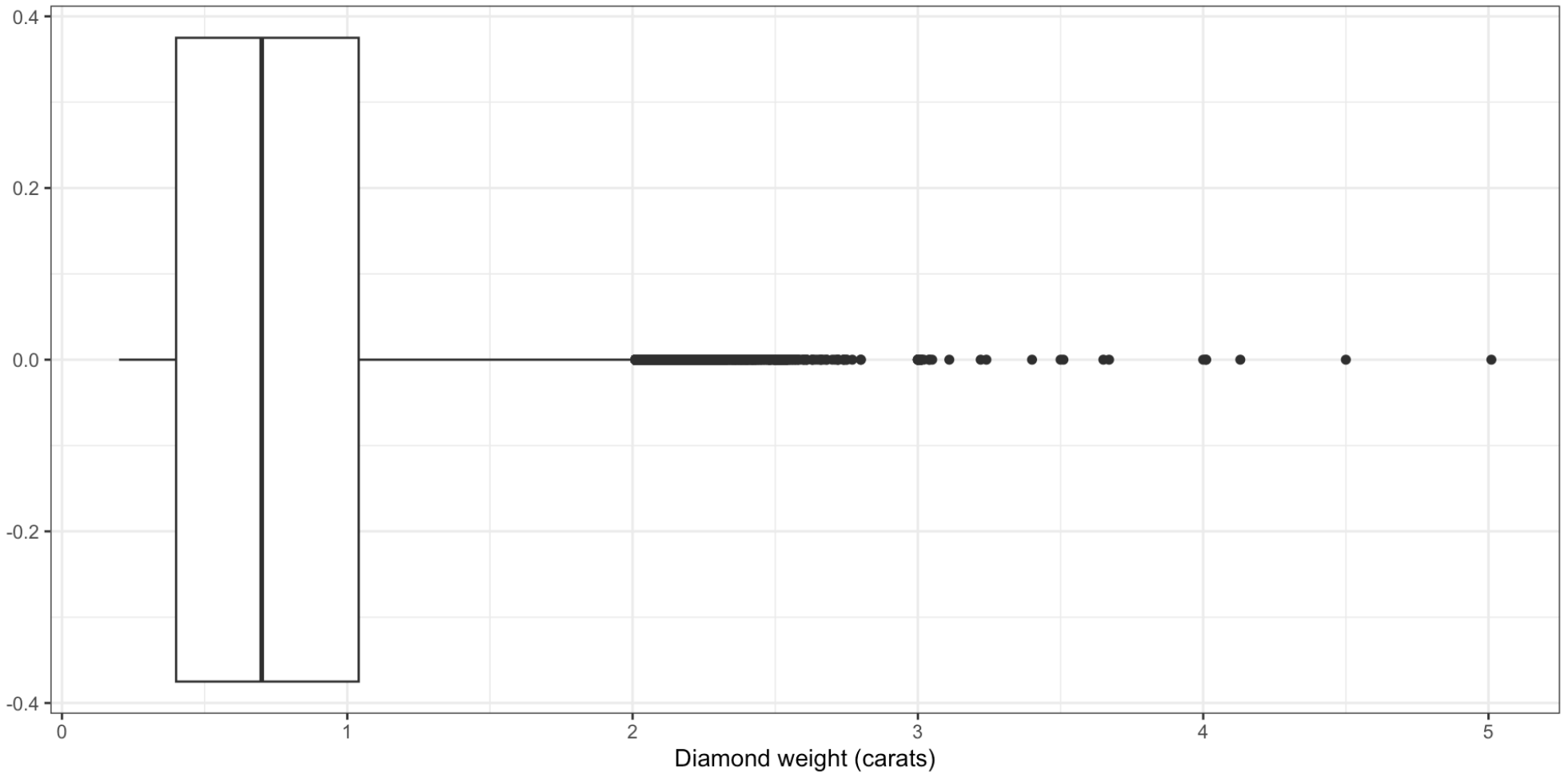


Another challenge



What are the individual points on the right of the boxplot?

Another challenge: identifying outliers



How do we identify outliers when constructing a boxplot?

points outside $(Q1 - 1.5IQR,\ Q3 + 1.5IQR)$

Identifying outliers

We wish to write a function that we can use to identify outliers in numeric variables. (using 1.5IQR rule)

What should we name the function?

something informative . e.g.

find_outliers

id_outliers

extract_outliers

Identifying outliers

We wish to write a function that we can use to identify outliers in numeric variables.

What should the input to the function be?

```
find_outliers <- function( x ) {

}
```

numeric vector (e.g., column in a data frame like in diamonds data)

Identifying outliers

```r
find_outliers <- function(x) {

}
```

What needs to happen inside the function?

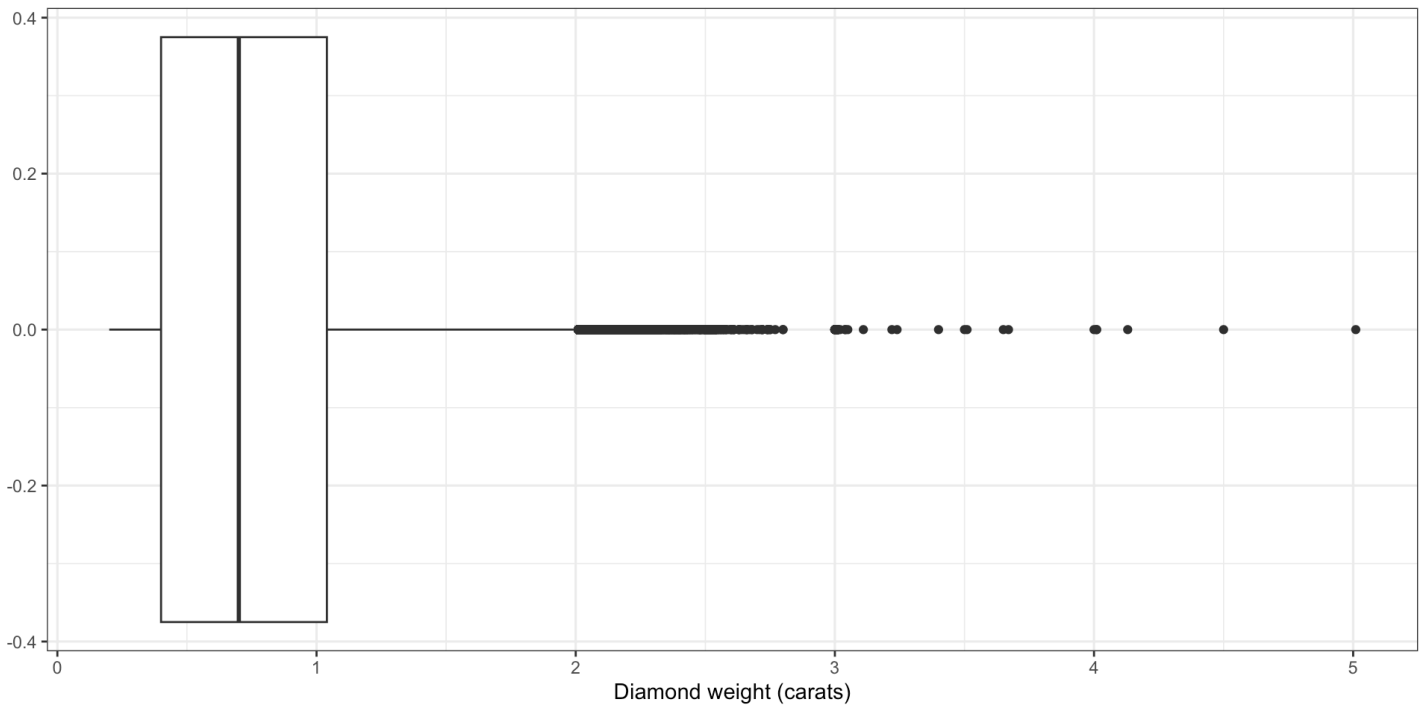
- calculate quartiles $Q1$ & $Q3$
- $IQR = Q_3 - Q_1$
- compare: x to $Q_3 + 1.5IQR$
  $Q1 - 1.5IQR$

Identifying outliers

```
find_outliers <- function(x) {

}
```

(Switch to R Studio)

Identifying outliers




```
diamonds |>
  mutate(carat_outliers = find_outliers(carat)) |>
  filter(carat_outliers) |>
  pull(carat) |>
  head()
```

```
[1] 2.06 2.14 2.15 2.22 2.01 2.01
```

Counting outliers

```
1 find_outliers <- function(x) {
2   q1 <- quantile(x, 0.25)
3   q3 <- quantile(x, 0.75)
4   iqr <- q3 - q1
5   (x > q3 + 1.5*iqr) | (x < q1 - 1.5*iqr)
6 }
```

What if we want to *count* the number of outliers?

- length(x[find_outliers(x)])

- T T F F

  T + T → 2

  F + F → 0

  T + F → 1

  etc.

find_outliers(x)

↳ T T F F T T F

sum(find_outliers(x))

↳ # of TRUEs

Counting outliers

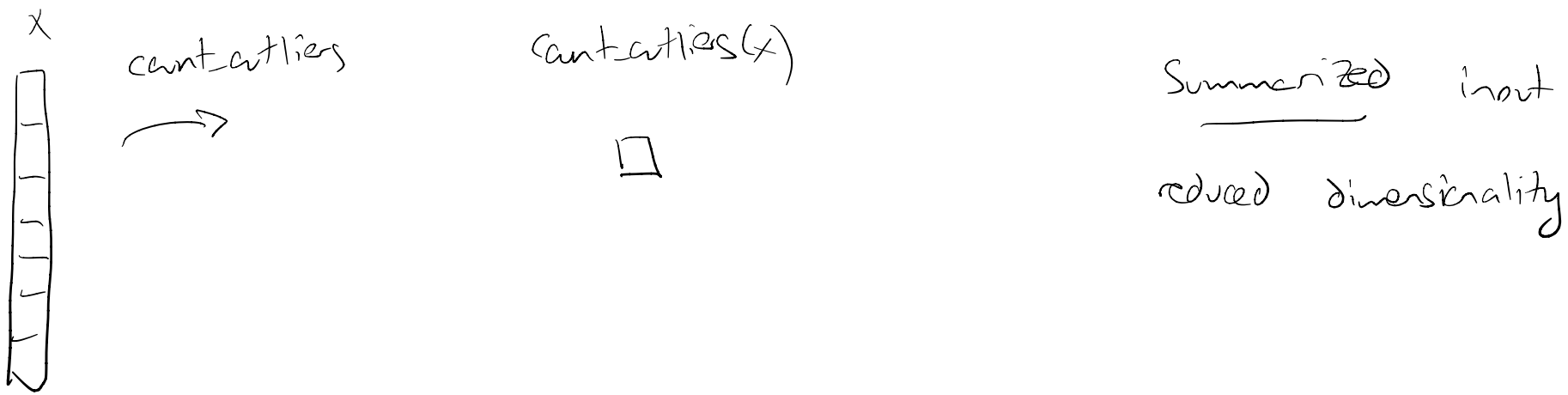
```r
find_outliers <- function(x) {
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.75)
  iqr <- q3 - q1
  (x > q3 + 1.5*iqr) | (x < q1 - 1.5*iqr)
}

count_outliers <- function(x) {
  sum(find_outliers(x))
}
```

Counting outliers

```
1 diamonds |>
2   summarize(carat_outliers = count_outliers(carat))
```

```
# A tibble: 1 × 1
  carat_outliers
           <int>
1           1889
```

R for Data Science would call the `count_outliers` function a "summarize" function. Why?



Counting outliers

```
1 diamonds |>
2   summarize(carat_outliers = count_outliers(carat))
```

```
# A tibble: 1 × 1
  carat_outliers
           <int>
1           1889
```

What if I want to count outliers for multiple variables?

Counting outliers

```
diamonds |>
  summarize(across(c(carat, depth, price),
                   list("outliers" = count_outliers)))
```

```
# A tibble: 1 × 3
  carat_outliers depth_outliers price_outliers
           <int>          <int>          <int>
1           1889           2545           3540
```

Class activity

https://sta279-f25.github.io/class_activities/ca_09.html

- Work with a neighbor on the class activity
- At the end of class, submit your work as an HTML file on Canvas (one per group, list all your names)

For next time, read:

- Chapter 25.3 in *R for Data Science*