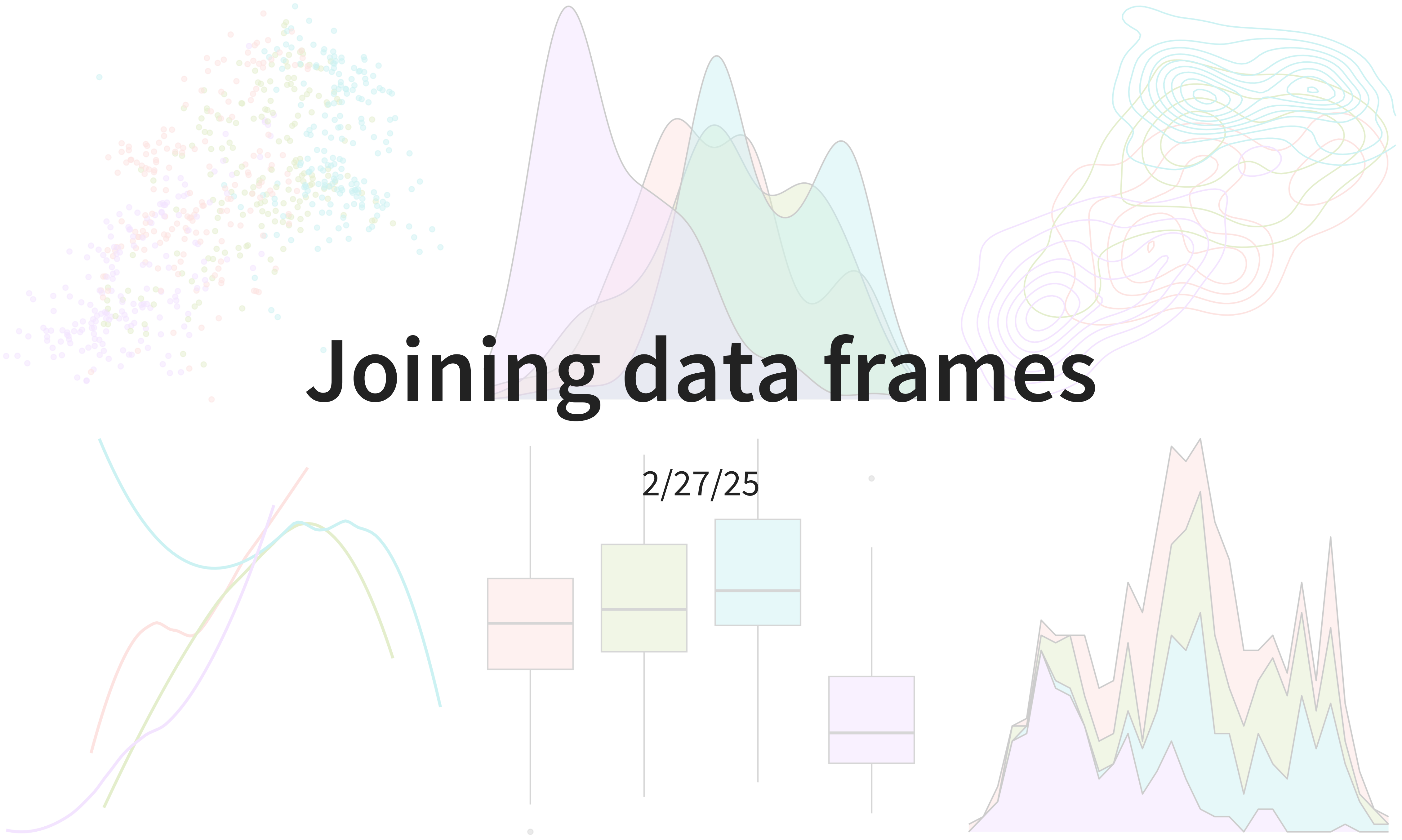


Joining data frames

2/27/25

Housekeeping

- Wrangling coding practice due tonight!
- Office hours tomorrow 10am-12pm
- Problem Set 3 has large R component

Joining data frames

Assume we have two data frame, `x` and `y`. There are some shared variables (i.e. columns) in the two. Suppose we want to combine them together into one single data frame.

```r
something_join(x, y)
```

Setup

For the next few slides…

```
x
```

ID	x_val
1	x1
2	x2
3	x3

```
y
```

ID	y_val
1	y1
2	y2
4	y4

`left_join()`

Adds columns to x from y, matching all rows in x

```r
x
```

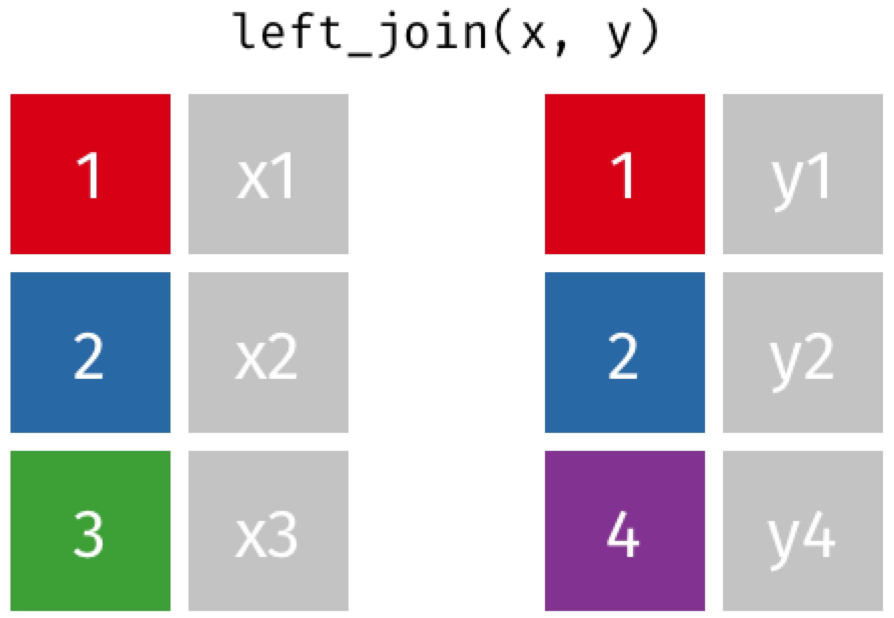
ID	x_val
1	x1
2	x2
3	x3

```r
y
```

ID	y_val
1	y1
2	y2
4	y4

```r
left_join(x, y, by = "ID")
```

ID	x_val	y_val
1	x1	y1
2	x2	y2
3	x3	NA



right_join()

Adds columns to x from y, matching all rows in y

```
x
```

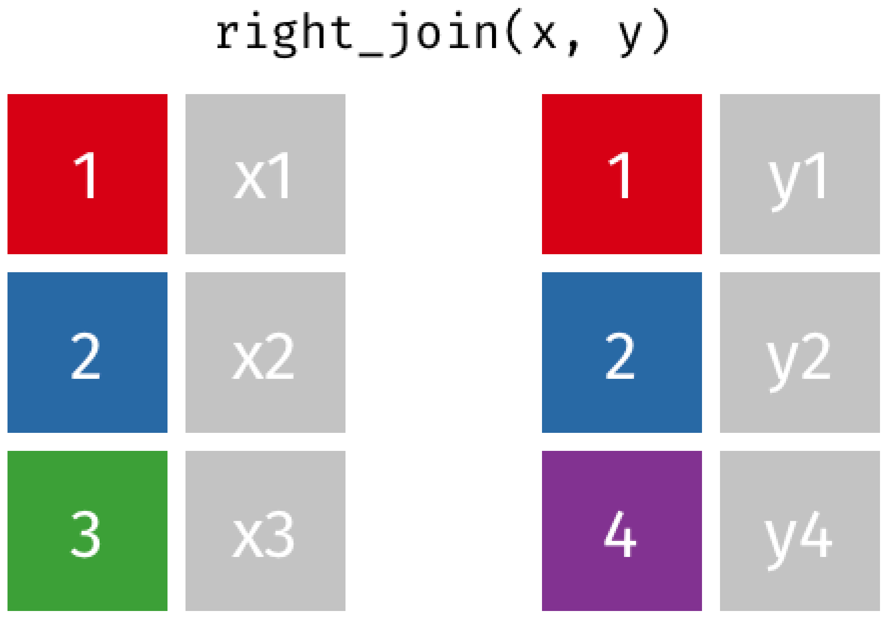
ID	x_val
1	x1
2	x2
3	x3

```
y
```

ID	y_val
1	y1
2	y2
4	y4

```r
right_join(x, y, by = "ID")
```

ID	x_val	y_val
1	x1	y1
2	x2	y2
4	NA	y4



full_join()

Adds columns to x from y, matching all rows in x OR y

```
x
```

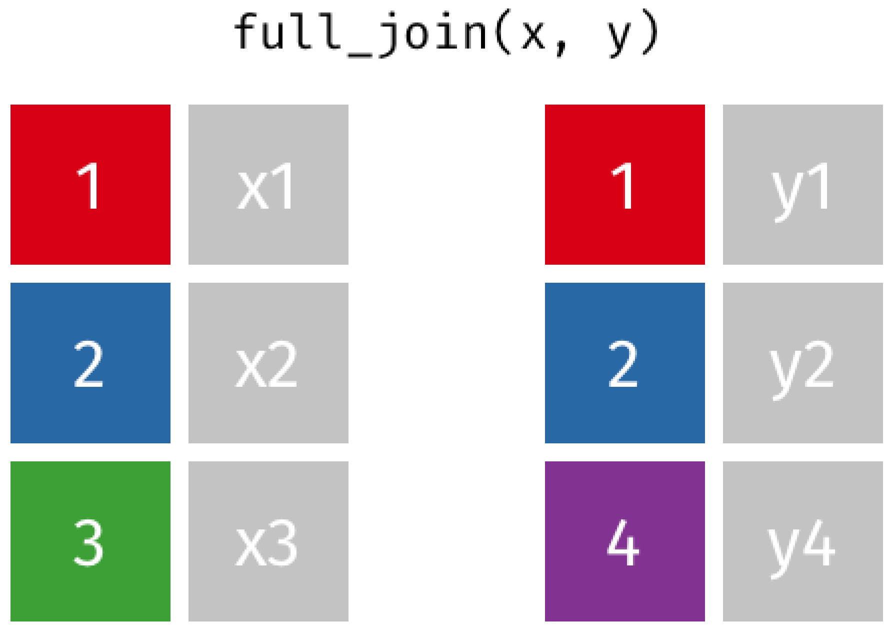
ID	x_val
1	x1
2	x2
3	x3

```
y
```

ID	y_val
1	y1
2	y2
4	y4

```
full_join(x, y, by = "ID")
```

ID	x_val	y_val
1	x1	y1
2	x2	y2
3	x3	NA
4	NA	y4



`inner_join()`

All rows from x where there are matching values in y, return all combination of multiple matches in the case of multiple matches

```r
x
```

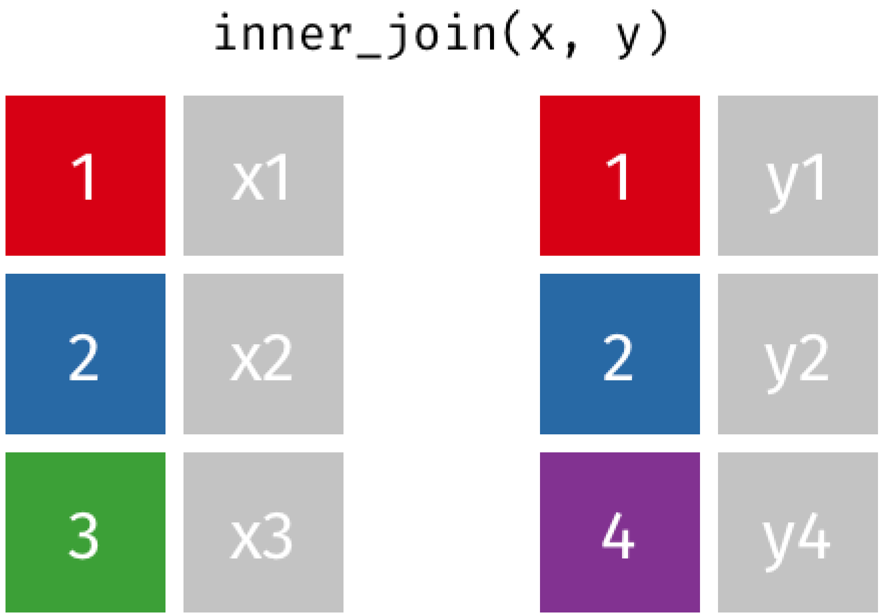
ID	x_val
1	x1
2	x2
3	x3

```r
y
```

ID	y_val
1	y1
2	y2
4	y4

```r
inner_join(x, y, by = "ID")
```

ID	x_val	y_val
1	x1	y1
2	x2	y2



`inner_join()` (cont.)

Example with multiple matches:

```r
x2
```

ID	x_val
1	x1
2	x2
3	x3
1	new_x

```r
y2
```

ID	y_val
1	y1
2	y2
4	y4
1	new_y

```r
inner_join(x2, y2, by = "ID")
```

```
Warning in inner_join(x2, y2, by = "ID"): Detected an unexpected many-to-many relationship between `x`
and `y`.
ℹ Row 1 of `x` matches multiple rows in `y`.
ℹ Row 1 of `y` matches multiple rows in `x`.
ℹ If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

ID	x_val	y_val
1	x1	y1
1	x1	new_y
2	x2	y2
1	new_x	y1
1	new_x	new_y

semi_join()

Returns all rows from x **with** a match in y, but does not add columns from y

```
x
```

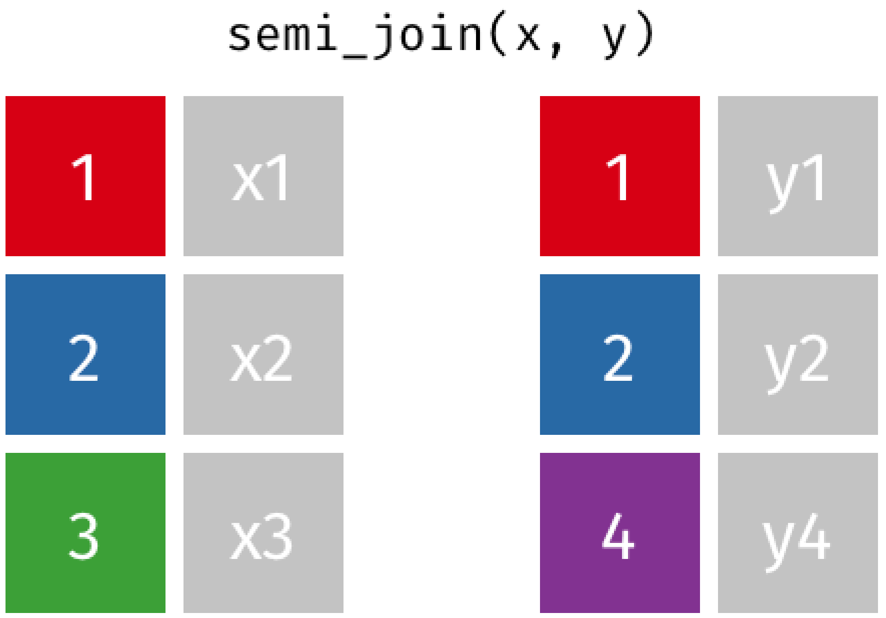
ID	x_val
1	x1
2	x2
3	x3

```
y
```

ID	y_val
1	y1
2	y2
4	y4

```
semi_join(x, y, by = "ID")
```

ID	x_val
1	x1
2	x2



anti_join()

Returns all rows from x **without** any match in y, and not add columns from y

```
x
```

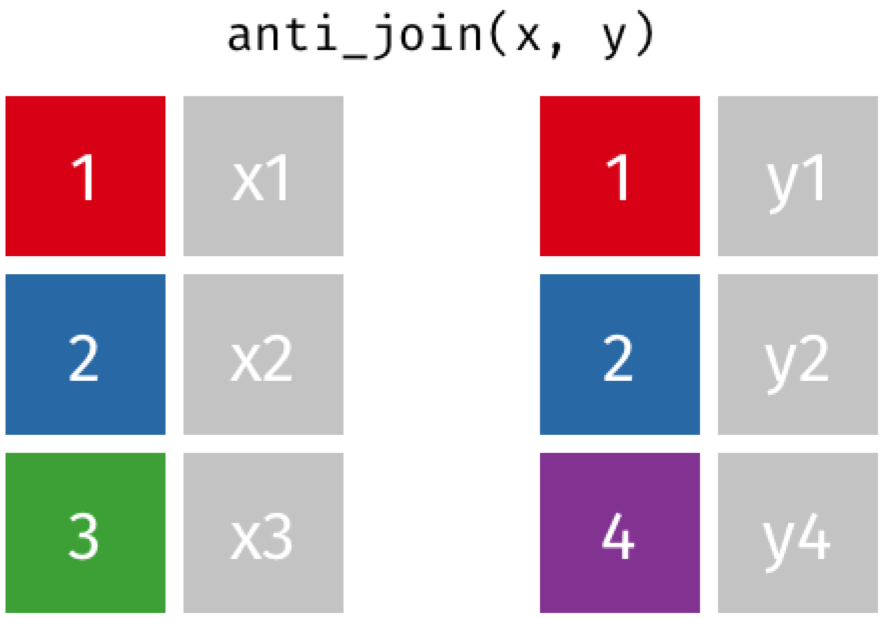
ID	x_val
1	x1
2	x2
3	x3

```
y
```

ID	y_val
1	y1
2	y2
4	y4

```
anti_join(x, y, by = "ID")
```

ID	x_val
3	x3



Joining with different variable names

If the variables in x and y have different names but we know they represent the same variable:

```r
x
```

ID	x_val
1	x1
2	x2
3	x3

```r
y3
```

ID_y	y_val
1	y1
2	y2
4	y4

```r
left_join(x, y3, by =  c("ID" = "ID_y"))
```

ID	x_val	y_val
1	x1	y1
2	x2	y2
3	x3	NA

Joining on multiple variables

Can specify more than one variable in the `by` argument. Will need to a vector of character objects.

```r
enrollment
```

student_id	course	start_year
1	STAT 310	F22
1	MATH 223	F22
2	STAT 310	F23
3	STAT 201	W24

```r
payment
```

student_id	Course	status
1	STAT 310	paid
1	MATH 223	paid
2	STAT 310	unpaid
3	STAT 201	paid

```r
# note, multiple join functions would work in this example!
inner_join(enrollment, payment, by = c("student_id", "course" = "Course"))
```

student_id	course	start_year	status
1	STAT 310	F22	paid
1	MATH 223	F22	paid
2	STAT 310	F23	unpaid
3	STAT 201	W24	paid

Live code

Example

We have data on fishery harvests (in tons) by countries from 2016:

```r
fish |>
    slice(1:9)
```

country	capture	aquaculture
Afghanistan	1000	1200
Albania	7886	950
Algeria	95000	1361
American Samoa	3047	20
Andorra	0	0
Angola	486490	655
Antigua and Barbuda	3000	10
Argentina	755226	3673
Armenia	3758	16381

Bringing in continent

Suppose I would like to explore the data on a continent level. We don't have continent in the current data frame, but we could join in the following data:

```r
continents |>
  slice(1:5)
```

country	continent
Afghanistan	Asia
Åland Islands	Europe
Albania	Europe
Algeria	Africa
American Samoa	Oceania

- We want to keep all rows and columns from `fish` and add a column for corresponding continents. Which join function should we use?
- We want to keep all rows from `fish` for which we have a corresponding continent and add a column for corresponding continents. Which join function should we use?

Example (cont.)

```r
left_join(fish, continents, by = "country") |>
  slice(1:9)
```

country	capture	aquaculture	continent
Afghanistan	1000	1200	Asia
Albania	7886	950	Europe
Algeria	95000	1361	Africa
American Samoa	3047	20	Oceania
Andorra	0	0	Europe
Angola	486490	655	Africa
Antigua and Barbuda	3000	10	NA
Argentina	755226	3673	Americas
Armenia	3758	16381	Asia

```r
inner_join(fish, continents, by = "country") |>
  slice(1:9)
```

country	capture	aquaculture	continent
Afghanistan	1000	1200	Asia
Albania	7886	950	Europe
Algeria	95000	1361	Africa
American Samoa	3047	20	Oceania
Andorra	0	0	Europe
Angola	486490	655	Africa
Argentina	755226	3673	Americas
Armenia	3758	16381	Asia
Aruba	142	0	Americas

- Notice the NA
- Could also use the following piping code:

```r
fish |>
  left_join(continents, by = "country")
```

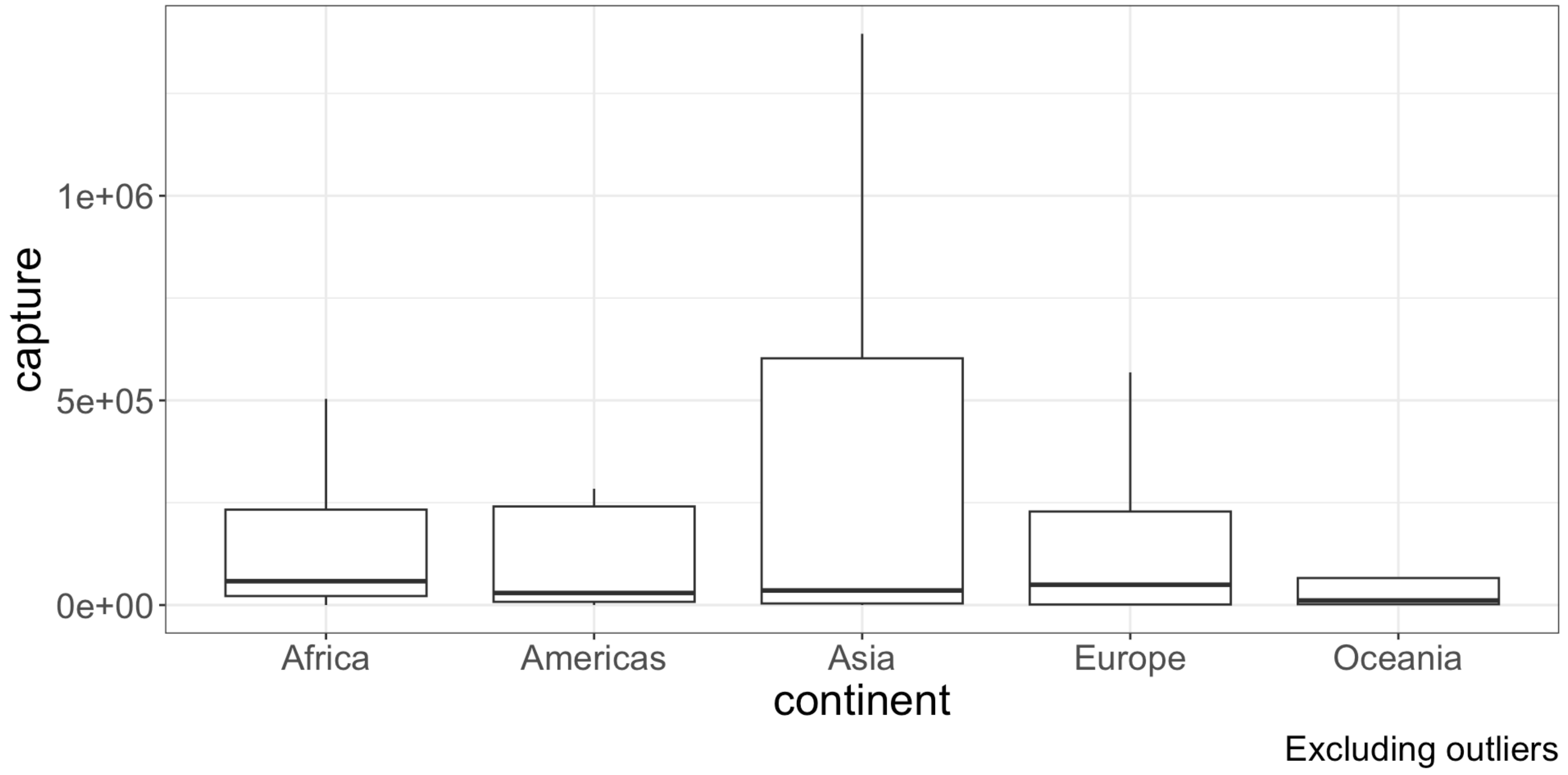
group_by()

```r
fish |>
  left_join(continents, by = "country") |>
  group_by(continent) |> # get continent-level summary stats
  summarise(mean_capture = mean(capture), sd_capture = sd(capture))
```

```
# A tibble: 6 × 3
  continent mean_capture sd_capture
  <chr>            <dbl>      <dbl>
1 Africa         180705.    266107.
2 Americas       433235.   1038899.
3 Asia          1036018.   2869652.
4 Europe         317874.    797072.
5 Oceania         74660.    121027.
6 <NA>           134722.    448079.
```

Visualize

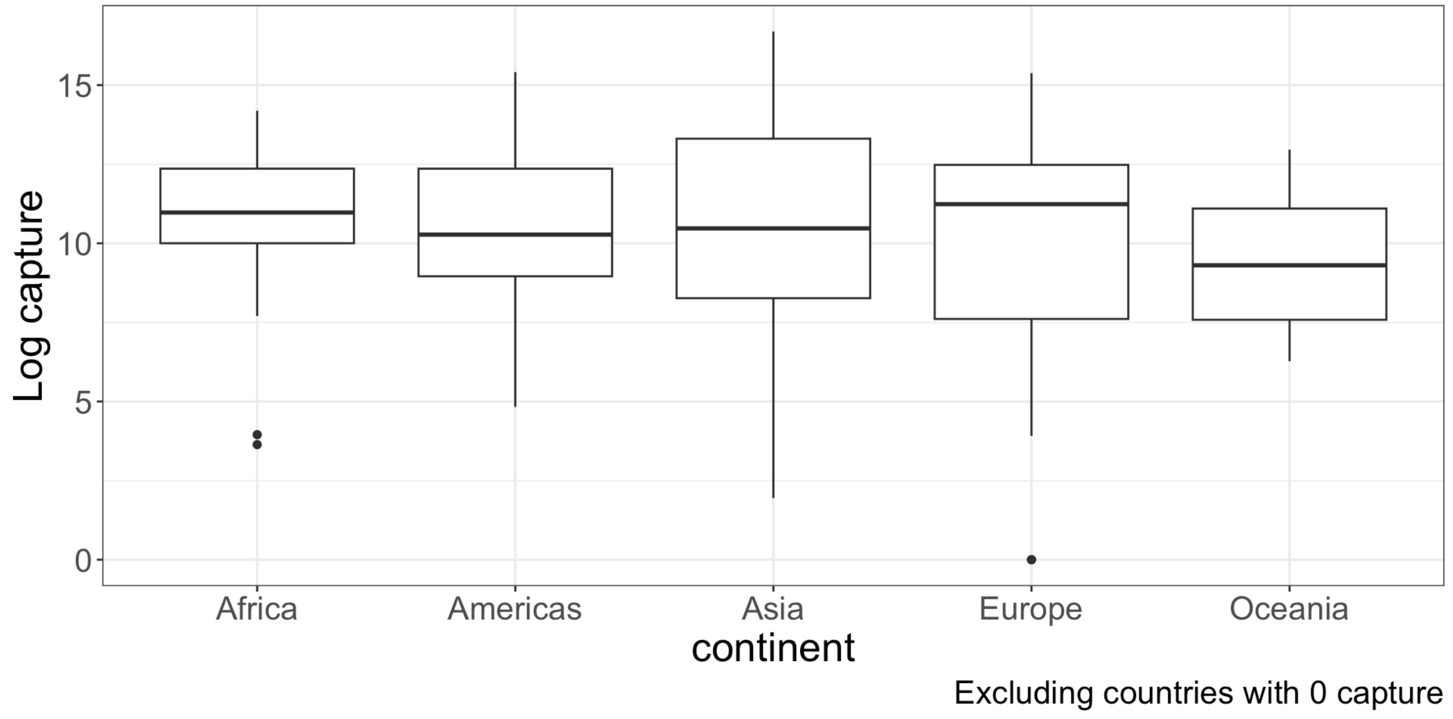
```r
fish |>
  na.omit() |> #remove observations with any NAs
  left_join(continents, by = "country") |>
  ggplot(aes(x = continent, y = capture)) +
  geom_boxplot(outliers = F) +
  theme_bw() +
  labs(caption = "Excluding outliers")
```



Visualize (cont.)

```r
fish |>
  mutate(log_capture = log(capture)) |>
  left_join(continents, by = "country") |>
  na.omit() |> #remove observations with any NAs
  ggplot(aes(x = continent, y = capture)) +
  geom_boxplot(outliers = F) +
  theme_bw() +
  labs(caption = "Excluding outliers", y = "Log capture")
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```


Log capture
Africa
Americas
Asia
Europe
Oceania
continent
Excluding countries with 0 capture