

Functions

Warmup activity

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

Warmup

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by(group_var) |>  
4     summarize(max(max_var, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

What is this code *trying* to do?

Warmup

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by(group_var) |>  
4     summarize(max(max_var, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.

What is causing the error?

Warmup

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by(group_var) |>  
4     summarize(max(max_var, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.

What should we change so the code runs correctly?

Embracing

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by({{ group_var }}) |>  
4     summarize(max({{ max_var }}, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

A tibble: 3 × 2

	species	`max(bill_depth_mm, na.rm = T)`
	<fct>	<dbl>
1	Adelie	21.5
2	Chinstrap	20.8
3	Gentoo	17.3

Why do we need embracing?

```
1 penguins |>  
2   filter(species == "Adelie")
```

This code contains two different types of variables:

- `penguins` is an **env-variable** (environment variable)
- `species` is a **data-variable** (it makes sense only within the context of a data frame)

Env-variables

Env-variables are objects in the R environment that we can interact with directly. For example:

```
1 head(penguins)
```

```
# A tibble: 6 × 8
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm
body_mass_g	<fct>	<fct>	<dbl>	<dbl>	<int>
<int>					
1	Adelie	Torgersen	39.1	18.7	181
3750					
2	Adelie	Torgersen	39.5	17.4	186
3800					
3	Adelie	Torgersen	40.3	18	195
3250					
4	Adelie	Torgersen	NA	NA	NA
NA					

Data-variables

Data-variables only exist in the context of a data frame:

```
1 # R doesn't know what 'species' is:  
2 species
```

Error: object 'species' not found

```
1 # R DOES understand species in the context of penguins:  
2 penguins$species
```

```
[1] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie  
[8] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie  
[15] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie  
[22] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie  
[29] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie  
[36] Adelie Adelie Adelie Adelie Adelie Adelie  
Adelie
```

Tidy evaluation

Many tidyverse functions are nice and allow us to reference *data-variables*:

```
1 penguins |>  
2   filter(species == "Adelie")
```

Here `filter` **knows** to look for a column called `species` in the `penguins` data.

Tidy evaluation

Of course, you will get an error if you try to reference a data-variable that doesn't exist! E.g. if we mis-spell the name:

```
1 penguins |>  
2   filter(speices == "Adelie")
```

```
Error in `filter()`:  
i In argument: `speices == "Adelie"`.  
Caused by error:  
! object 'speices' not found
```

Tidy evaluation

Of course, you will get an error if you try to reference a data-variable that doesn't exist!

```
1 penguins |>
2   group_by(group_var) |>
3   summarize(max(max_var, na.rm=T))
```

```
Error in `group_by()` :
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.
```

The problem: `group_var` and `max_var` are not columns in the `penguins` data!

Tidy evaluation

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by(group_var) |>  
4     summarize(max(max_var, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.

What we want R to run:

```
1 penguins |>  
2   group_by(species) |>  
3   summarize(max(bill_depth_mm, na.rm=T))
```

```
# A tibble: 3 × 2  
  species    `max(bill_depth_mm, na.rm = T)`  
  <fct>                                <dbl>  
1 Adelie                                21.5
```

2 Chinstrap

20.8

3 Gentoo

17.3

Tidy evaluation

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by(group_var) |>  
4     summarize(max(max_var, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.

What R is **actually** running:

```
1 penguins |>  
2   group_by(group_var) |>  
3   summarize(max(max_var, na.rm=T))
```

Error in `group_by()`:
! Must group by variables found in `.data`.
✖ Column `group_var` is not found.

The solution: embracing

```
1 grouped_max <- function(df, group_var, max_var) {  
2   df |>  
3     group_by({{ group_var }}) |>  
4     summarize(max({{ max_var }}, na.rm=T))  
5 }  
6  
7 grouped_max(penguins, species, bill_depth_mm)
```

```
# A tibble: 3 × 2  
  species    `max(bill_depth_mm, na.rm = T)`  
  <fct>                <dbl>  
1 Adelie                21.5  
2 Chinstrap             20.8  
3 Gentoo                17.3
```

What R is running now:

```
1 penguins |>  
2   group_by(species) |>  
3   summarize(max(bill_depth_mm, na.rm=T))
```



```
# A tibble: 3 × 2
  species    `max(bill_depth_mm, na.rm = T)`
  <fct>                <dbl>
1 Adelie             21.5
2 Chinstrap          20.8
3 Gentoo             17.3
```

Another example

Suppose we want to fit a simple linear regression model:

```
1 penguins |>  
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>  
3   coef()
```

```
(Intercept) bill_depth_mm  
55.0673698   -0.6498356
```

Another example

```
1 penguins |>
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>
3   coef()
```

```
(Intercept) bill_depth_mm
55.0673698    -0.6498356
```

```
1 lm_coef <- function(df, x, y) {
2   df |>
3     lm({{ y }} ~ {{ x }}, data = _) |>
4     coef()
5 }
6
7 lm_coef(penguins, bill_depth_mm, bill_length_mm)
```

Do you think this code will work?

Another example

```
1 penguins |>
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>
3   coef()
```

```
(Intercept) bill_depth_mm
55.0673698    -0.6498356
```

```
1 lm_coef <- function(df, x, y) {
2   df |>
3     lm({{ y }} ~ {{ x }}, data = _) |>
4     coef()
5 }
6
7 lm_coef(penguins, bill_depth_mm, bill_length_mm)
```

Error: object 'bill_length_mm' not found

Why does this code fail?

Another example

```
1 penguins |>
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>
3   coef()
```

```
(Intercept) bill_depth_mm
55.0673698    -0.6498356
```

```
1 lm_coef <- function(df, x, y) {
2   df |>
3     lm({{ y }} ~ {{ x }}, data = _) |>
4     coef()
5 }
6
7 lm_coef(penguins, bill_depth_mm, bill_length_mm)
```

Error: object 'bill_length_mm' not found

Problem: The `lm` function does not support tidy evaluation!
(To see if a function does support tidy evaluation, look for

Fixing the issue

```
1 penguins |>
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>
3   coef()
```

```
(Intercept) bill_depth_mm
55.0673698    -0.6498356
```

```
1 lm_coef <- function(df, x, y) {
2   df |>
3     lm({{ y }} ~ {{ x }}, data = _) |>
4     coef()
5 }
6
7 lm_coef(penguins, bill_depth_mm, bill_length_mm)
```

Error: object 'bill_length_mm' not found

If `lm` doesn't support tidy evaluation, what could we do differently?

Fixing the issue

$$\text{SLR slope: } \hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

```
1 penguins |>
2   lm(bill_length_mm ~ bill_depth_mm, data = _) |>
3   coef()
```

```
(Intercept) bill_depth_mm
55.0673698   -0.6498356
```

```
1 penguins |>
2   summarize(slope = cov(bill_depth_mm, bill_length_mm,
3                           use="complete.obs")/
4                           var(bill_depth_mm, na.rm=T))
```

```
[1] -0.6498356
```

Fixing the issue

```
1 penguins |>
2   summarize(slope = cov(bill_depth_mm, bill_length_mm,
3                         use="complete.obs")/
4                 var(bill_depth_mm, na.rm=T))
```

How would I turn this into a function?

```
1 slr_slope <- function(df, x, y) {
2
3
4
5
6 }
```


Fixing the issue

```
1 slr_slope <- function(df, x, y) {  
2   df |>  
3     summarize(slope = cov({{ x }}, {{ y }}, use="complete.obs")/  
4               var({{ x }}, na.rm=T))  
5 }  
6  
7 slr_slope(penguins, bill_depth_mm, bill_length_mm)
```

```
# A tibble: 1 × 1  
  slope  
  <dbl>  
1 -0.650
```

```
1 slr_slope(penguins, flipper_length_mm, bill_length_mm)
```

```
# A tibble: 1 × 1  
  slope  
  <dbl>  
1 0.255
```

Class activity

https://sta279-f25.github.io/class_activities/ca_10.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 26.3 in *R for Data Science*