

Teaching R with R

How to get started doing lectures with Quarto and reveal.js

Håvard R. Karlsen, NTNU – date: 2026-07-09



Introduction

Håvard R. Karlsen

- Associate professor of personality psychology and psychometrics at the Norwegian University of Science and Technology (NTNU).
- Teaches practical skills in R in [PSY3202 - Knowledge in Practice 2: Project Work in Data Management and Interdisciplinary Communication](#).

Today

*One of **many** ways to create slides.*

Tools

- **RStudio** to make slides.
- **Quarto** documents, with
- **reveal.js** to make the presentations. Present slides through
- **Google Chrome/Chromium** for best experience.

Assumed knowledge not covered here: The basics of reveal.js slidecrafting. See [The Quarto docs guide on reveal.js](#).

See code to generate these slides at [GitHub](#)

Make sure to use the right aspect ratio

To use all available screen real estate.

$$\text{Aspect ratio} = \frac{\text{width}}{\text{height}}$$

- Default: $\frac{1050}{700} = 1.5$
- My laptop: $\frac{1050}{682.26} = 1.539$

Define a custom aspect ratio in the YAML:

yaml

```
1 format:  
2   revealjs:  
3     height: 682.26
```

Absolute positioning

Absolute positioning: Full control of placement of content, but often breaks in other formats or aspect ratios. Vs. **relative positioning**.

```
{.absolute  
top="-10%" right="-40%"  
height="100%"}
```

Do note that the text will not respect the
might get overlaps.

To avoid overlap, use columns.

See [Emil Hvitfeldts post about slidecrafting](#) or his [book on the same subject](#) for more tips.



Columns helps you maximise your screen real estate

Example of use. This is column 1. Tweak the widths to accomodate images or code chunks.



```
1 # What day of the week is it?  
2 lubridate::wday("2026-07-09", label = TRUE)
```

```
[1] Thu  
Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

And this is column 2.

```
:::: {.columns}
```

```
::: {.column width="48%"}
```

Example of use. This is column 1. Tweak the widths to accomodate images or code chunks.

```
{width=70%}
```

```
# What day of the week is it?
```

```
lubridate::wday("2026-07-09", label = TRUE)
```

```
:::
```

```
:::: {.column width="52%"}
```

And this is column 2.

```
:::
```

```
::::
```

Display and run code: My two approaches

1. For simple, one line pieces of code or a small pipeline: Open up a script and do some live coding. Prepare a script beforehand to look at. Allows students to copy off the screen.
2. For longer pieces of code: Use **fragments** to show the code, and then the output, staggered.



Tip

To showcase faulty code, remember to set `eval: false` in chunk options or YAML to avoid it breaking the rendering.

Using fragments

Fragments reveal parts of the slide gradually. Can be **stylished extensively**.

Code

```
1 #| output-location: fragment
2
3 library(tidyverse)
4
5 penguins |>
6   filter(species == "Gentoo") |>
7   group_by(sex) |>
8   summarise(
9     body_mass_m = mean(body_mass,
10                        na.rm = TRUE),
11     body_mass_sd = sd(body_mass,
12                       na.rm = TRUE))
```

Presentation

```
1 library(tidyverse)
2
3 penguins |>
4   filter(species == "Gentoo") |>
5   group_by(sex) |>
6   summarise(
7     body_mass_m = mean(body_mass,
8                        na.rm = TRUE),
9     body_mass_sd = sd(body_mass,
10                      na.rm = TRUE))
```

```
# A tibble: 3 × 3
  sex      body_mass_m body_mass_sd
<fct>      <dbl>      <dbl>
1 female    4680.        282.
2 male      5485.        313.
3 <NA>      4588.        338.
```

Exercise sheet with answers

To avoid tedious copy-pasting of answer sheets for exercises, use parameters.

Define a YAML parameter `hide_answers` that is `true/false`. Wrap the content in a hide content div (`:::`) with R code (`if`) conditional on the parameter.

How this looks in practice:

yaml

```
1 params:
2   hide_answers: true
```

In your presentation slide:

slide

```
1 `r if (params$hide_answers) "::: {.content-hidden}"`
2
3 [Some code]
4
5 `r if (params$hide_answers) ":::"`
```

With `hide_answers = true`

Exercise 1

The Palmer penguins dataset has measured penguins' properties in mm. Turn the measurements into cm.

With `hide_answers = false`

Exercise 1

The Palmer penguins dataset has measured penguins' properties in mm. Turn the measurements into cm.

Answer

```
1 library(tidyverse)
2
3 penguins |>
4   mutate(
5     across(c(bill_len, bill_dep, flipper_len),
6             \(x) x / 10 )
7   ) |>
8   head()
```

	species	island	bill_len	bill_dep	flipper_len	body_mass	sex	year
1	Adelie	Torgersen	3.91	1.87	18.1	3750	male	2007
2	Adelie	Torgersen	3.95	1.74	18.6	3800	female	2007
3	Adelie	Torgersen	4.03	1.80	19.5	3250	female	2007
4	Adelie	Torgersen	NA	NA	NA	NA	<NA>	2007
5	Adelie	Torgersen	3.67	1.93	19.3	3450	female	2007
6	Adelie	Torgersen	3.93	2.06	19.0	3650	male	2007

Render the documents once with each parameter value (`true/false`) for both versions.

- Based on [Nicola Rennie's post](#).
- More at [Quarto docs](#).

Example of a nice section divider slide

```
1 # <span style="color: white;">Example of a nice section  
2 divider slide</span> {background-image="img/raccoon.jpg"}
```

Photo by [Jonatan Pie](#) on [Unsplash](#).

Styling/theming with scss

Apply a consistent style across all slides with scss. Create e.g. `mytheme.scss` and link it in the YAML. Helpful for text size, colours, defaults.

YAML

slides.qmd

```
1 format:
2   revealjs:
3     theme: mystyle.scss
```

Content of mystyle.scss

mystyle.scss

```
1 $body-color: #000 !default;
2 $presentation-font-size-root: 28px; /* Reduce the base text size */
3 $link-color: #689f38; /* Change links to green */
```

Check out mystyle.scss for more settings I have tweaked here.

Creating hand-outs: Printing the slideshow

Common problem: the canvas shifts, jumbling and/or removing text and images.

1. Make sure the aspect ratio matches your screen.
2. Render the slideshow.
3. Open it in the Chrome browser.
4. Press **e** to enter print mode.
5. Print to pdf with the browser's built-in print function (**cmd + p**).

Concluding tips

- Experiment a little bit every time you make a new presentation.
- Make notes of your findings to re-use them later.
- Read blogs from the R Community – lots of tips there.
- Have fun!

This presentation and all relevant materials can be found at [GitHub](#). Feel free to use it as an inspiration for your next slides.

Thank you for your attention!

Photo by [Jonatan Pie](#) on [Unsplash](#).