Example notebook

Madeline Gillman

2023-11-05

Analysis aims

The aim of this analysis is to learn how to create a neat, professional, and shareable pdf document describing basic analyses of the `penguins` dataset in R.

Creating tables

Creating tables in quarto is straightforward. The default R output works just fine, but there are many packages that can help you easily create publication quality tables. I recommend using the package `kableExtra` as it is the best at producing latex tables (i.e., for creating pdfs). If you don't intend to create a pdf, other great packages for creating nice tables include `gt` and DT. If you use those other packages and render your document to a pdf, you might get some unexpected behavior. This might change in the future–Quarto is relatively new, and packages like `gt` are being updated to better accommodate different output types.

```r
penguins %>%
  count(species)
```

```
# A tibble: 3 x 2
  species        n
  <fct>      <int>
1 Adelie       152
2 Chinstrap     68
3 Gentoo       124
```

Hmm, that table is not cute. By default, Quarto prints data frames and matrices as you'd see them in the R console. Let's try a few other options.

Note that we have to add `kableExtra::kable_styling(latex_options = "HOLD_position")`: this prevents the table from always appearing at the top of the page.

Table 1 Number of penguins per species formatted using `kableExtra::kable`

```
penguins %>%
  count(species) %>%
  kableExtra::kable() %>%
  kableExtra::kable_styling(latex_options = "HOLD_position")
```

Table 1: Number of penguins per species.

species	n
Adelie	152
Chinstrap	68
Gentoo	124

Let's try some other formatting options! It might be helpful to add stripes to make distinguishing the rows easier, or even highlighting certain rows. Try `??kableExtra::kable_styling` for more options.

Table 2 Number of penguins per species formatted using `kableExtra::kable` with other formatting options

```
penguins %>%
  count(species) %>%
  kableExtra::kable() %>%
  kableExtra::kable_styling(latex_options = c("HOLD_position", "striped"),
                            stripe_color = "#7BAFD4", stripe_index = 2)
```

Table 2: Number of penguins per species.

species	n
Adelie	152
Chinstrap	68
Gentoo	124

There are many chunk options for formatting tables. See https://quarto.org/docs/authoring/tables.html for more!

Sometimes, tables will be too wide and run off the page when rendered to a pdf. If this is the case, you can rotate that specific page using the lscape latex package. This will only affect the pdf output–it will still be portrait in html.

```
\newpage

\blandscape

# your code that makes the super wide table here

\elandscape
```

```
penguins %>%
  slice_head(n = 10) %>%
  kable() %>%
  kable_styling(latex_options = "HOLD_position")
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

You could also try the `scale_down` option in `kable_styling()`:

```r
penguins %>%
  slice_head(n = 10) %>%
  kable() %>%
  kable_styling(latex_options = c("HOLD_position", "scale_down"))
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

Plots and images

You can easily add plots or insert images (clip art, photos, .pngs of plots you've already made). Helpful code chunk options when making figures:

- `#| echo: false` which will exclude source code in output

- `#| warning: false` which will not print warnings to the output

- `#| fig-width: 5` controls the width of the figure.

- `#| fig-cap: "Polar axis plot"` adds a figure caption.

- `#| fig-alt: "A line plot on a polar axis"` adds alt text.

You may have noticed that you can add references (links to sections in the document) to tables and figures. To use these, add one of the below labels to the code chunk. Note that figures must start with `fig` and tables must start with `tbl`. You can later reference these throughout the document by typing @

- `#| label: fig-descriptive-yet-short-name-for-figure`

- `#| label: tbl-descriptive-yet-short-name-for-table`

For more things you can do such as more figure/image sizing options, alignment, and caption locations, see https://quarto.org/docs/authoring/figures.html

```
ggplot(
  penguins,
  aes(x = bill_length_mm, y = bill_depth_mm,
      color = species, shape = species)) +
  geom_point() +
  theme_minimal() +
  labs(x = "Bill length (mm)", y = "Bill depth (mm)")
```

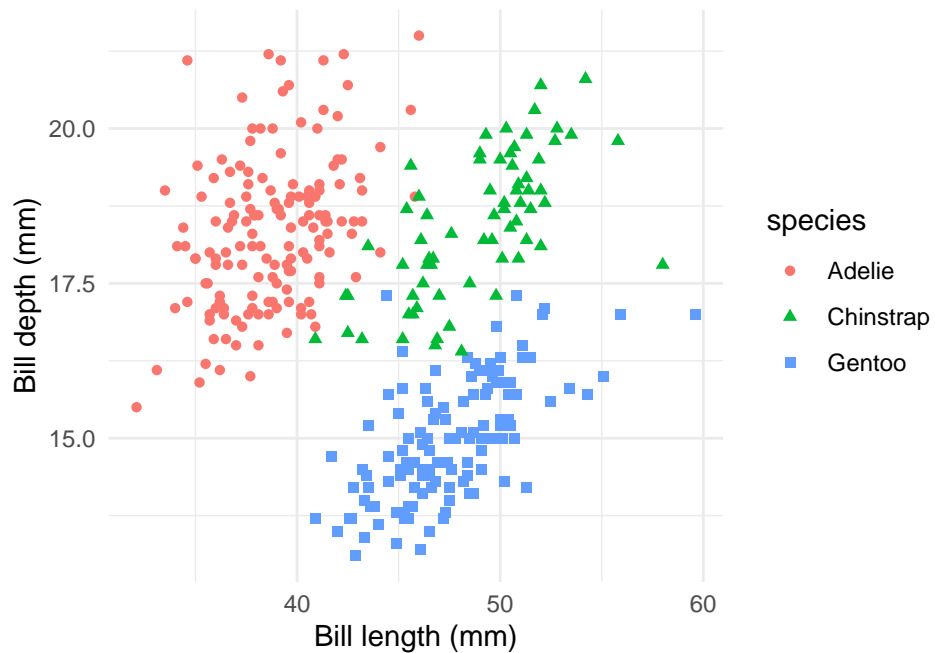Warning: Removed 2 rows containing missing values (`geom_point()`).

Figure 1: Bill length vs bill depth for three penguin species

Note that a lot of the options overlap slightly in function! See this table from R for Data Science: (also an example of adding an image)

Option	Run code	Show code	Output	Plots	Messages	Warnings
`eval: false`	X		X	X	X	X
`include: false`		X	X	X	X	X
`echo: false`		X				
`results: hide`			X			
`fig-show: hide`				X		
`message: false`					X	
`warning: false`						X