

Project Report Tips

STA 101

By the end of today you will practice a few quarto/markdown tricks to polish your report and simplify your presentation. Specifically we will discuss:

- code chunk settings
- citations
- `kable()` tables

Getting started

Code chunk settings

Some options available for customizing output (see [quarto documentation](#) for more detail).

Option	Description
<code>eval</code>	Evaluate the code chunk (if <code>false</code> , just echos the code into the output)
<code>echo</code>	Include the source code in output
<code>warning</code>	Include warnings in the output
<code>message</code>	Whether to preserve messages emitted by <code>message()</code> (similar to the option <code>warning</code>)
<code>include</code>	Catch all for preventing any output (code or results) from being included (e.g. <code>include: false</code> suppresses all output from the code block)

These options can be applied globally (the whole document) or locally (a specific code chunk). Global settings are controlled in the YAML (see the top of the document) while local code chunk options can be applied with `#|` (see example below).

Exercise 1

In the code chunk below:

- set `warning` to `false`
- set `echo` to `false`

and re-render.

```
library(knitr)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.4
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.4.4      v tibble     3.2.1
v lubridate  1.9.2      v tidyr      1.3.0
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x purrr::%||%() masks base::%||%()
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(tidymodels)
```

```
-- Attaching packages ----- tidymodels 1.1.1 --
v broom      1.0.5      v rsample     1.2.0
v dials      1.2.0      v tune        1.1.2
v infer      1.0.5.9000 v workflows   1.1.3
v modeldata  1.2.0      v workflowsets 1.0.1
v parsnip    1.1.1      v yardstick   1.2.0
v recipes    1.0.8
-- Conflicts ----- tidymodels_conflicts() --
x purrr::%||%() masks base::%||%()
x scales::discard() masks purrr::discard()
x dplyr::filter() masks stats::filter()
x recipes::fixed() masks stringr::fixed()
x dplyr::lag() masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step() masks stats::step()
* Learn how to get started at https://www.tidymodels.org/start/
```

```
library(ggrepel)
library(ggthemes)
library(palmerpenguins)
```

Attaching package: 'palmerpenguins'

The following object is masked from 'package:modeldata':

penguins

```
library(gridExtra)
```

Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

combine

In addition to code chunks, figures have settings as well.

We can set captions and an `alt` attributes using `#| fig-cap:` and `#| fig-alt:` respectively. `alt` captions specify “alternate text” for an image. Alternative text appears if an image cannot be displayed and is also read by screen-readers.

Additional figure options include

Option	Description
<code>fig-width</code>	figure width in inches
<code>fig-height</code>	figure height in inches
<code>fig.align</code>	e.g. <code>fig.align: center</code> centers figure alignment
<code>fig.asp</code>	changes figure height based on aspect ratio with width
<code>out.width</code>	sets figure width relative to text (1000 = 100% text width), e.g. <code>out.width: 1000</code>

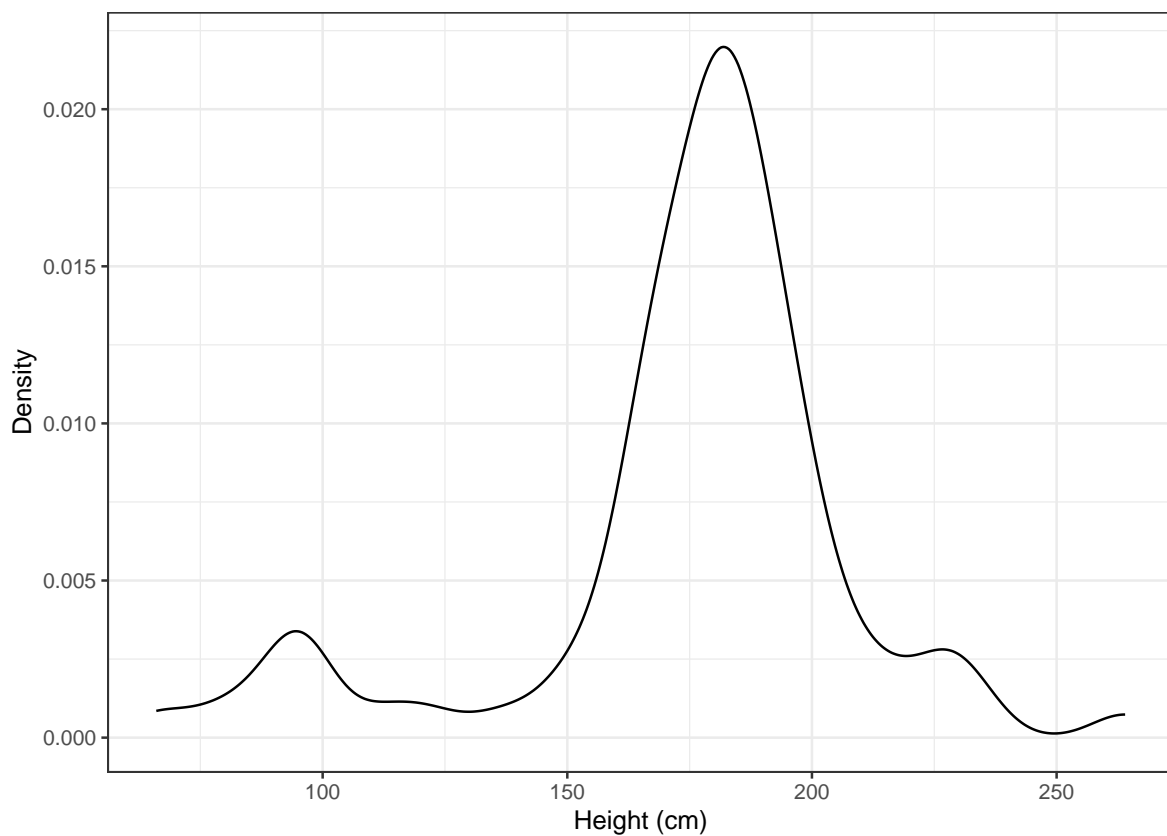
In all cases above, we can again set options locally or globally. Note: local options override global options.

Exercise 2

Add a figure caption to the figure below. Next, change the output width to be 50% of the text. Finally, align the figure with the center of the page.

```
starwars |>
  ggplot(aes(x = height)) +
  geom_density() +
  labs(x = "Height (cm)", y = "Density") +
  theme_bw()
```

Warning: Removed 6 rows containing non-finite values (`stat_density()`).



Project specific notes

For the project, you will set the option `echo: FALSE` and `warning: FALSE` to hide all code and warnings in your final report.

Suggestion: make your figures consistently themed, e.g. use similar figure size/aspect ratio and color scheme throughout your report. Change the default gray background, see [themes](#).

Exercise 3

Change the global code chunk settings so the document is formatted as your final project will be. Render and take a look at the updated PDF.

Citations

Your report will include citations, e.g. the data source, previous research, and other sources as needed. At a minimum, you should have a citation for the data source.

All of your bibliography entries will be stored in a `.bib` file. The entries of the bibliography are stored using BibTeX, i.e., a format to store citations in LaTeX. Let's take a look at `references.bib`.

In addition to the `.bib` file:

- Include `bibliography: references.bib` in the YAML.
- At the end of the report, include `## References`. This will list all of the references at the end of the document.

Learn more [here](#).

Convert website to citation: [here](#).

Or find on google scholar!

Citation examples

1. In Wickham, Chang, and Wickham (2016), the authors focus present the grammar of graphics package `ggplot2` for R.
2. Within the grammar of graphics, `ggplot()` is the first layer of any plot (Wickham, Chang, and Wickham 2016).

Links

Add URLs to your document using the following syntax:

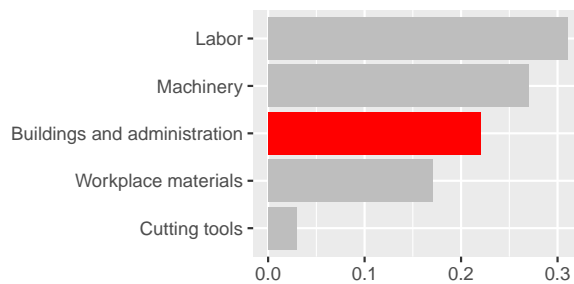
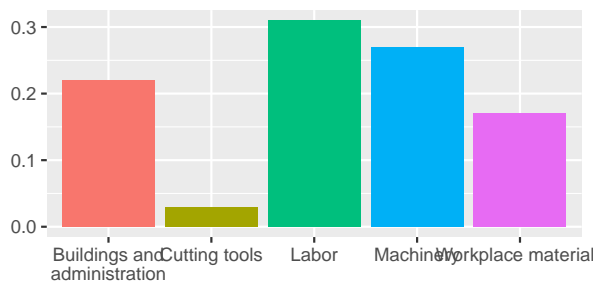
[Class website](#)

Use color to draw attention

```
d <- tribble(
  ~category,           ~value,
  "Cutting tools"      , 0.03,
  "Buildings and administration" , 0.22,
  "Labor"              , 0.31,
  "Machinery"          , 0.27,
  "Workplace materials" , 0.17
)
d
```

A tibble: 5 x 2

	category	value
	<chr>	<dbl>
1	Cutting tools	0.03
2	Buildings and administration	0.22
3	Labor	0.31
4	Machinery	0.27
5	Workplace materials	0.17

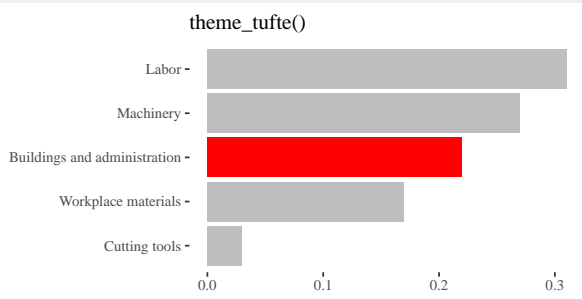
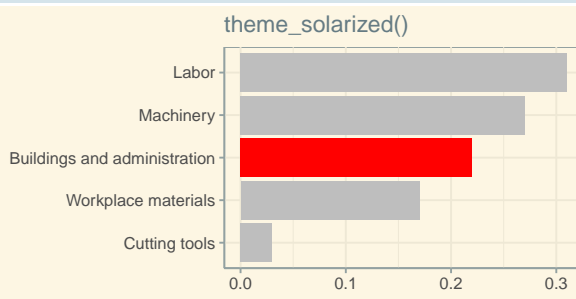
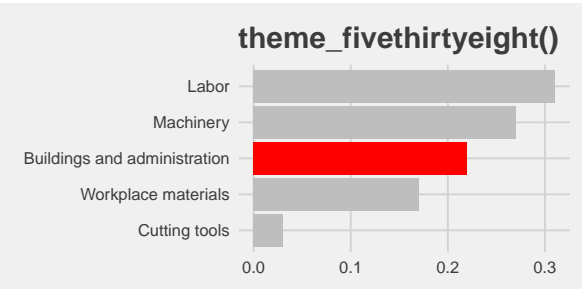
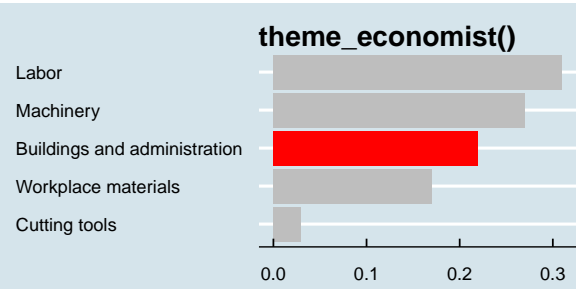
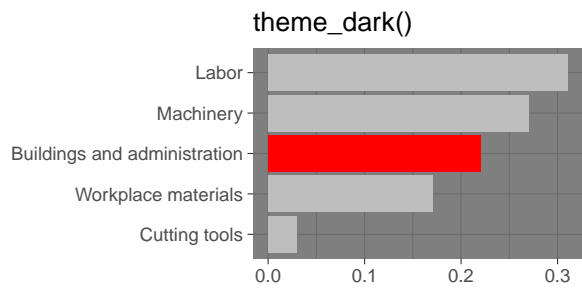
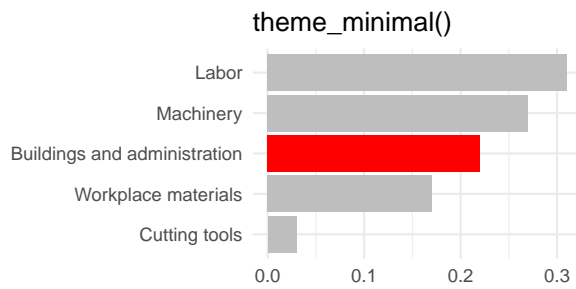
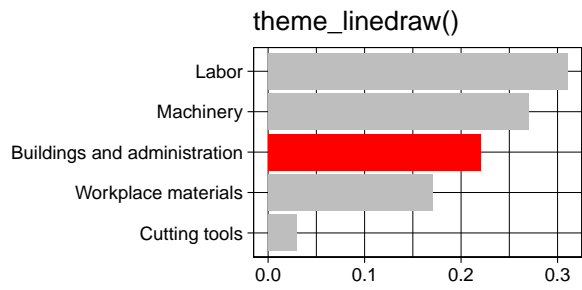
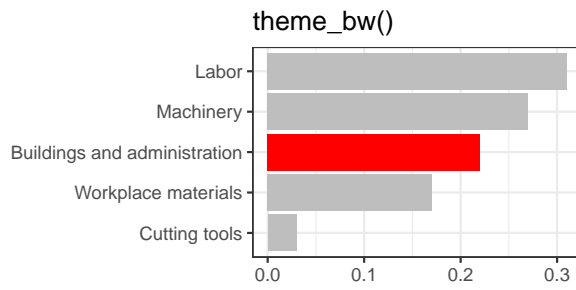


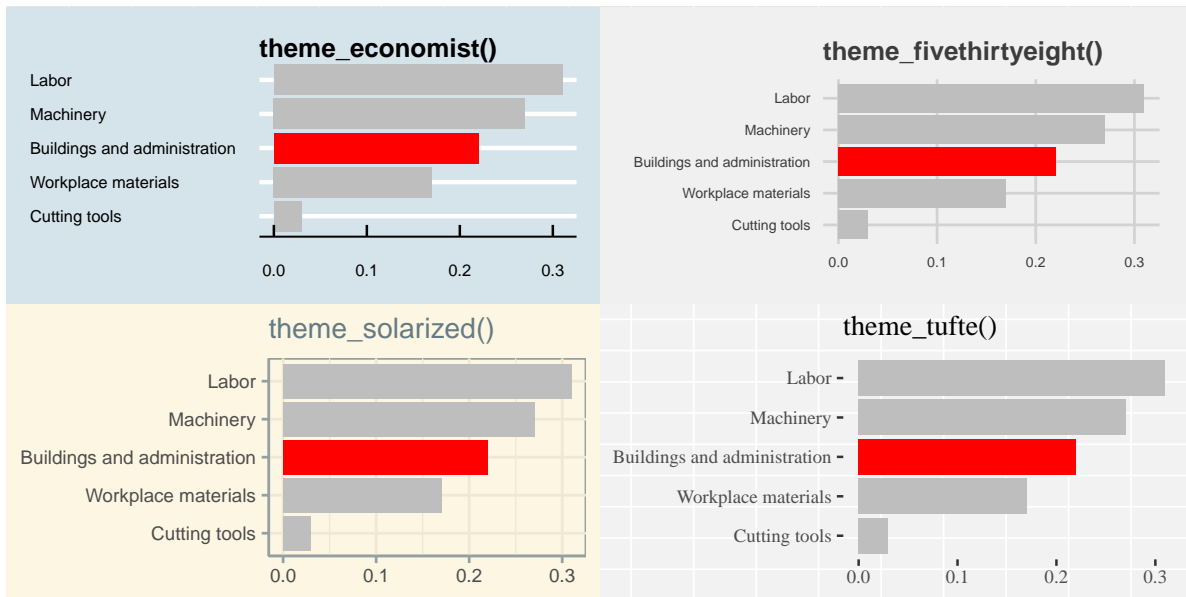
Note the orders in these plot!

Play with themes for a non-standard look

Go beyond ggplot2 themes – ggthemes

How to save the image for the presentation:





Plot sizing and layout

Sample plots

```
p_hist <- ggplot(mtcars, aes(x = mpg)) +
  geom_histogram(binwidth = 2)

p_text <- mtcars |>
  rownames_to_column() |>
  ggplot(aes(x = disp, y = mpg)) +
  geom_text_repel(aes(label = rowname)) +
  coord_cartesian(clip = "off")
```

Small fig-width

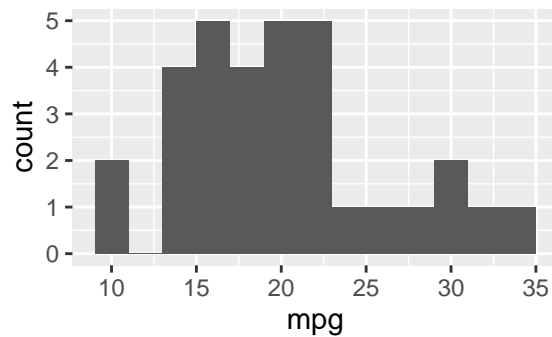
For a zoomed-in look

```
```{r}
#| fig-width: 3
#| fig-asp: 0.618
```



```
p_hist
ggsave("hist.png")
````
```

Saving 3 x 1.85 in image



Large fig-width

For a zoomed-out look

```
````{r}
#| fig-width: 6
#| fig-asp: 0.618

p_hist
````
```

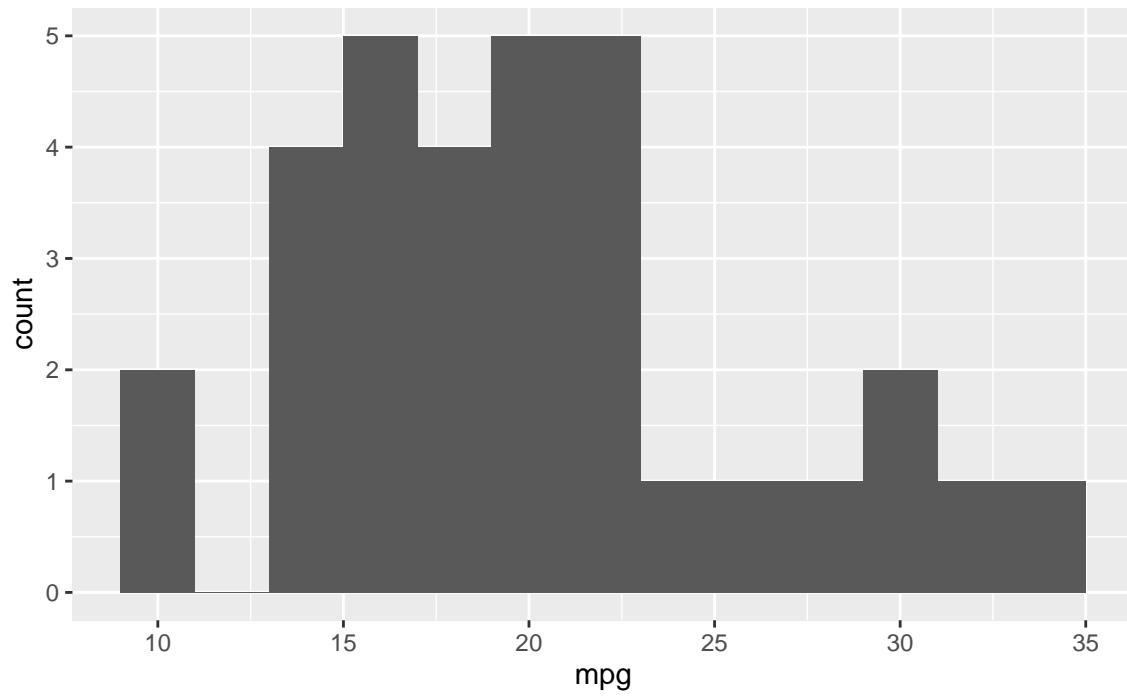
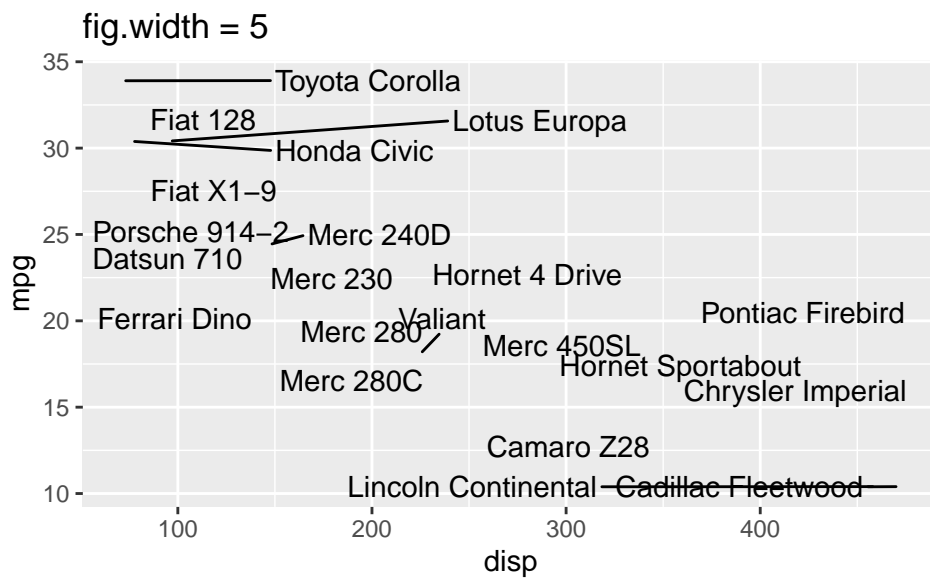
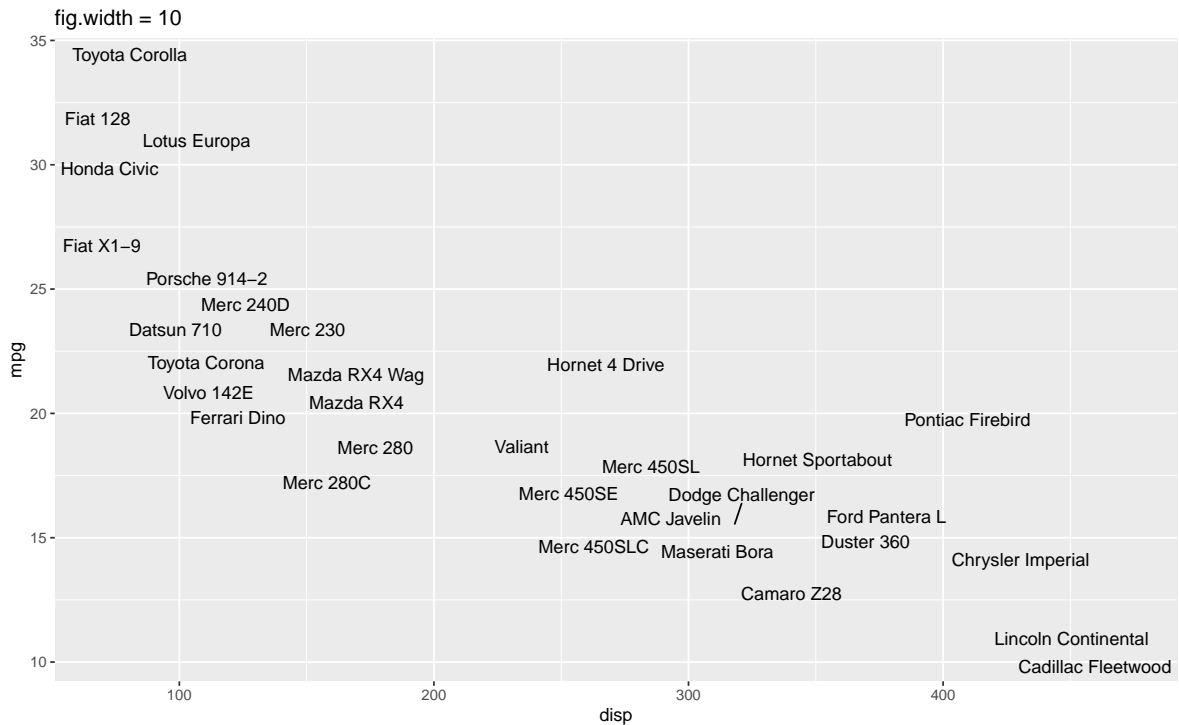


fig-width affects text size





Multiple plots on a slide

First, ask yourself, must you include multiple plots on a slide? For example, is your narrative about comparing results from two plots?

- If no, then don't! Move the second plot to to the next slide!
- If yes, use columns and sequential reveal.

Cross referencing figures

As seen in Figure 1, there is a positive and relatively strong relationship between body mass and flipper length of penguins.

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point()
```

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

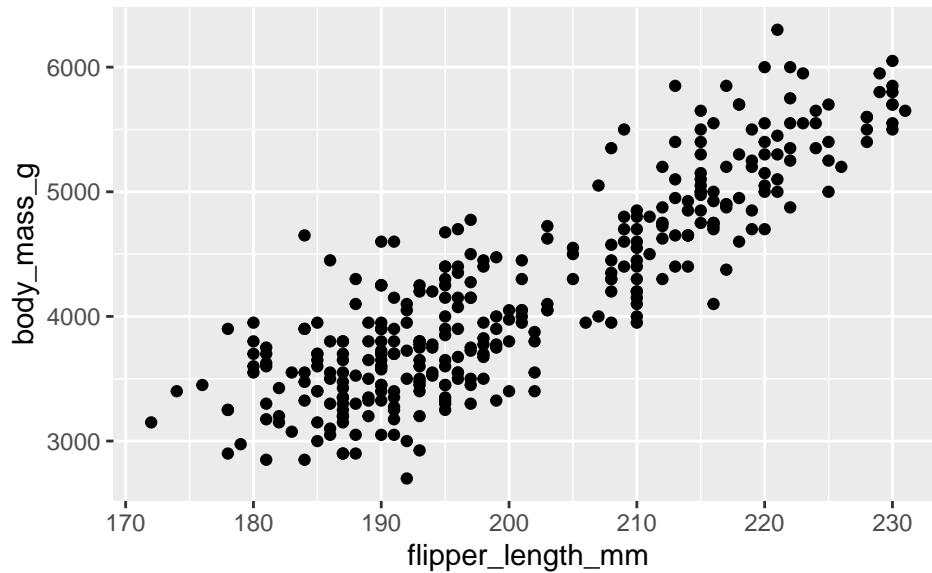


Figure 1: The relationship between body mass and flipper length of penguins.

Cross referencing tables

The regression output is shown in Table 3.

```
penguins_fit <- linear_reg() |>
  fit(body_mass_g ~ flipper_length_mm, data = penguins)

tidy(penguins_fit) |>
  knitr::kable(digits = 3)
```

Table 3: The regression output for predicting body mass from flipper length of penguins.

| term | estimate | std.error | statistic | p.value |
|-------------------|-----------|-----------|-----------|---------|
| (Intercept) | -5780.831 | 305.815 | -18.903 | 0 |
| flipper_length_mm | 49.686 | 1.518 | 32.722 | 0 |

Similarly, looking at the dataset we created earlier:

```
d |> knitr::kable()
```

| category | value |
|------------------------------|-------|
| Cutting tools | 0.03 |
| Buildings and administration | 0.22 |
| Labor | 0.31 |
| Machinery | 0.27 |
| Workplace materials | 0.17 |

Writing your project report with Quarto

- Figure sizing: `fig-width`, `fig-height`, etc. in code chunks.
- Figure layout: `layout-ncol` for placing multiple figures in a chunk.
- Further control over figure layout with the **patchwork** package.
- Chunk options around what makes it in your final report: `message`, `echo`, etc.
- Cross referencing figures and tables.
- Adding footnotes and citations.

References

Wickham, Hadley, Winston Chang, and Maintainer Hadley Wickham. 2016. “Package ‘Ggplot2’.” *Create Elegant Data Visualisations Using the Grammar of Graphics. Version 2* (1): 1–189.