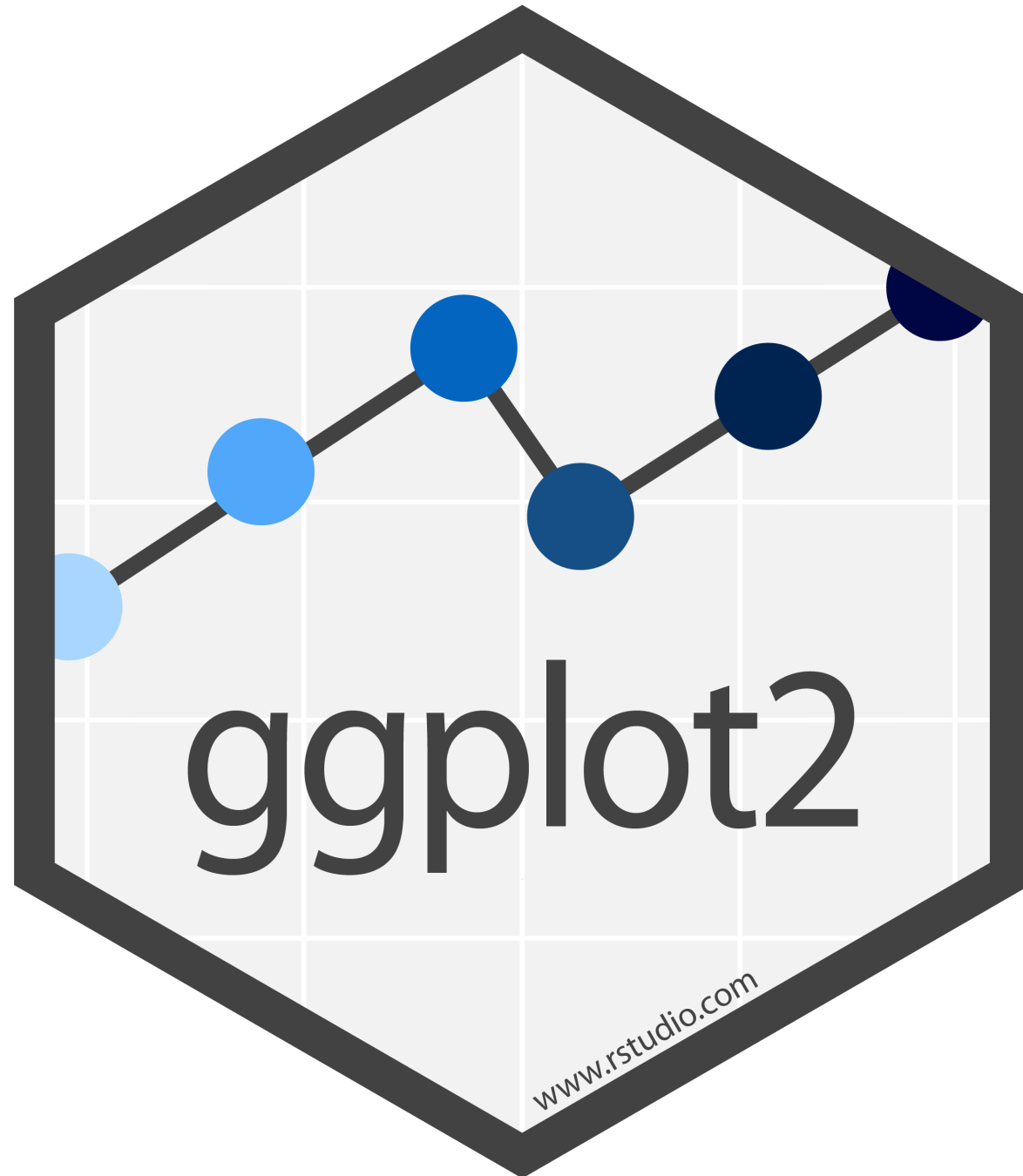


Visualization with ggplot2

Lecture 10

Dr. Colin Rundel



The Grammar of Graphics

- Visualization concept created by Leland Wilkinson (The Grammar of Graphics, 1999)
- attempt to taxonomize the basic elements of statistical graphics
- Adapted for R by Hadley Wickham (2009)
 - consistent and compact syntax to describe statistical graphics
 - highly modular as it breaks up graphs into semantic components
 - ggplot2 is not meant as a guide to which graph to use and how to best convey your data (more on that later), but it does have some strong opinions.

Terminology

A statistical graphic is a...

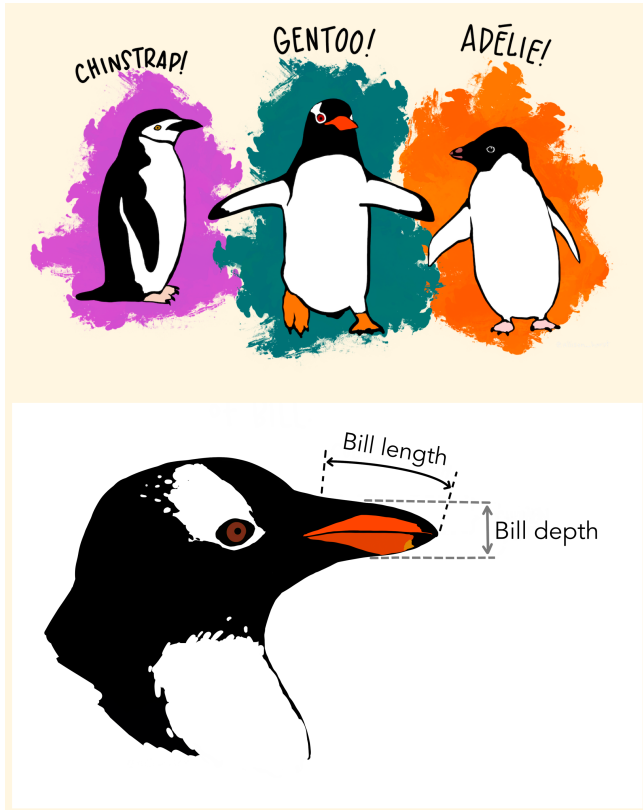
- mapping of **data**
- which may be **statistically transformed** (summarized, log-transformed, etc.)
- to **aesthetic attributes** (color, size, xy-position, etc.)
- using **geometric objects** (points, lines, bars, etc.)
- and mapped onto a specific **facet** and **coordinate system**

Anatomy of a ggplot call

```
1 ggplot(  
2   data = [dataframe],  
3   mapping = aes(  
4     x = [var x], y = [var y],  
5     color = [var color],  
6     shape = [var shape],  
7     ...  
8   )  
9 ) +  
10  geom_[some geom](  
11    mapping = aes(  
12      color = [var geom color],  
13      ...  
14    )  
15  ) +  
16  ... # other geometries  
17  scale_[some axis]_[some scale]() +  
18  facet_[some facet]([formula]) +  
19  ... # other options
```

Data - Palmer Penguins

Measurements for penguin species, island in Palmer Archipelago, size (flipper length, body mass, bill dimensions), and sex.



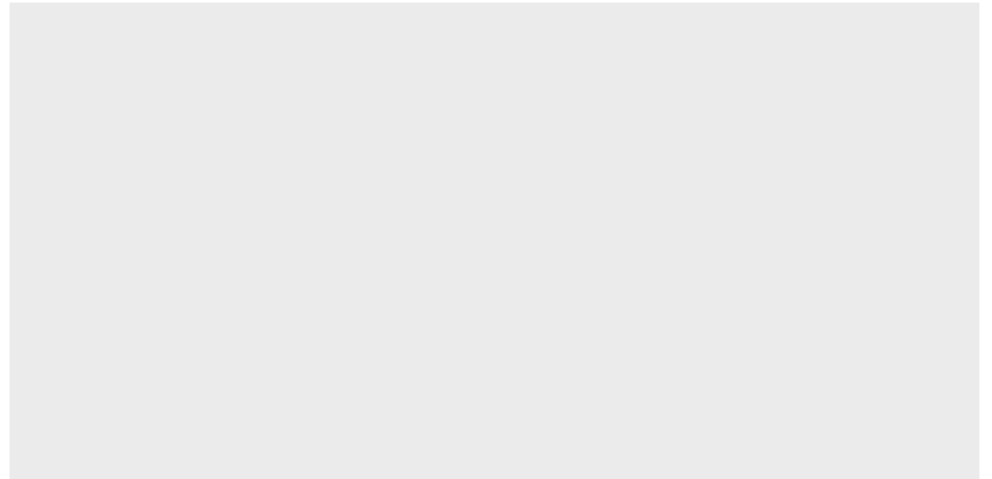
```
1 library(palmerpenguins)
2 penguins
```

```
# A tibble: 344 × 8
  species island bill_length_mm bill_depth_mm
  <fct>   <fct>         <dbl>         <dbl>
1 Adelie  Torgersen      39.1           18.7
2 Adelie  Torgersen      39.5           17.4
3 Adelie  Torgersen      40.3            18
4 Adelie  Torgersen      NA             NA
5 Adelie  Torgersen      36.7           19.3
6 Adelie  Torgersen      39.3           20.6
7 Adelie  Torgersen      38.9           17.8
8 Adelie  Torgersen      39.2           19.6
9 Adelie  Torgersen      34.1           18.1
10 Adelie Torgersen      42             20.2
# i 334 more rows
# i 4 more variables: flipper_length_mm <int>,
#   body_mass_g <int>, sex <fct>, year <int>
```

Text <-> Plot

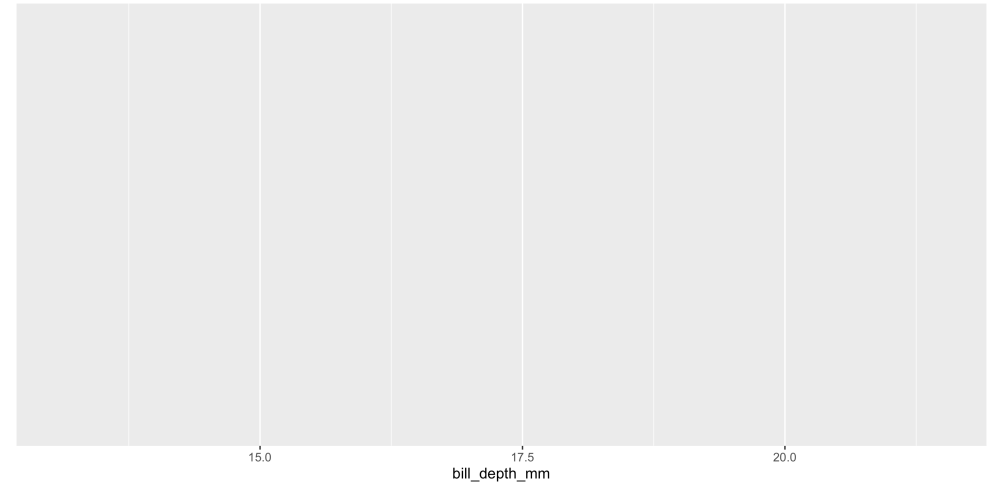
| Start with the *penguins* data frame

```
1 ggplot(data = penguins)
```



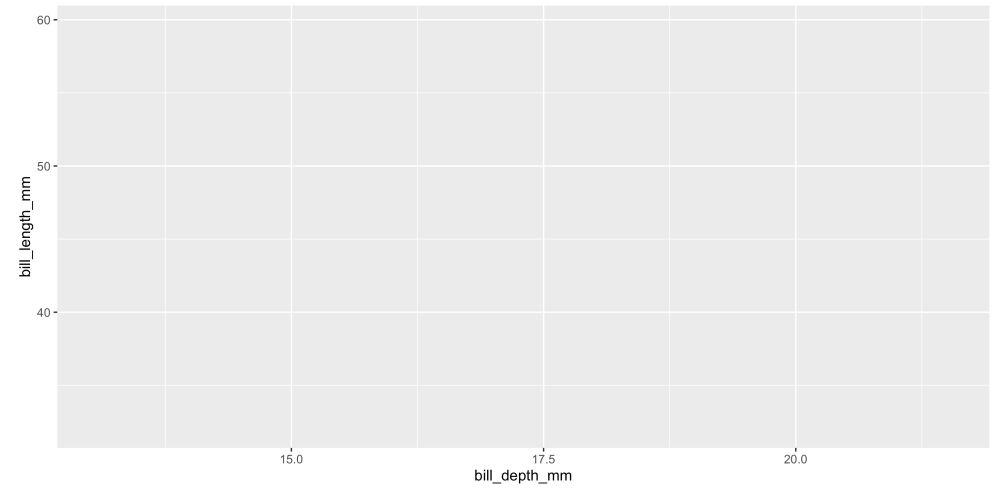
Start with the `penguins` data frame, *map bill depth to the x-axis*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm  
5   )  
6 )
```



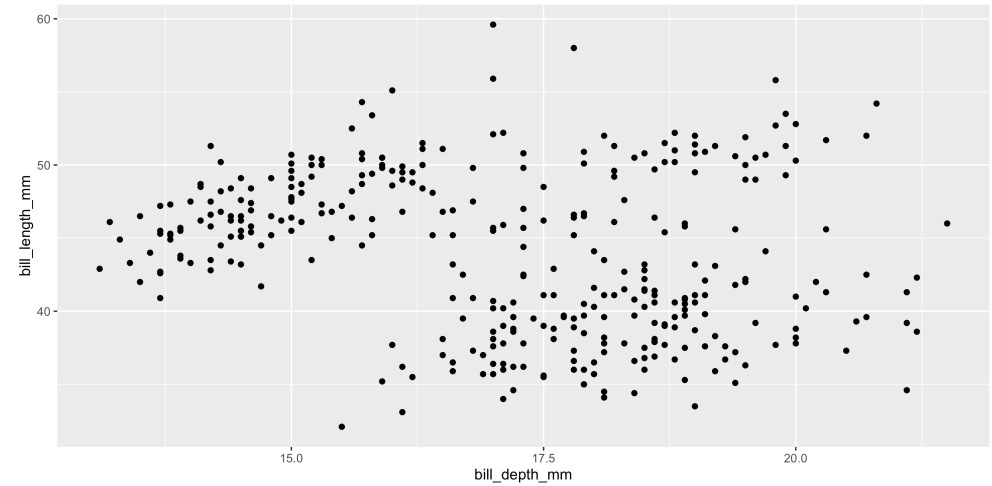
Start with the `penguins` data frame, map bill depth to the x-axis *and map bill length to the y-axis*.

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 )
```



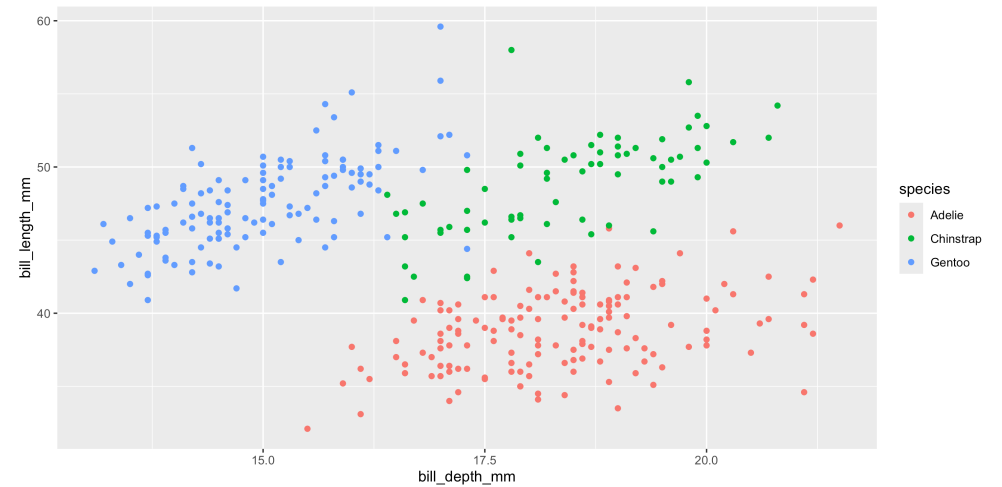
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. *Represent each observation with a point*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point()
```



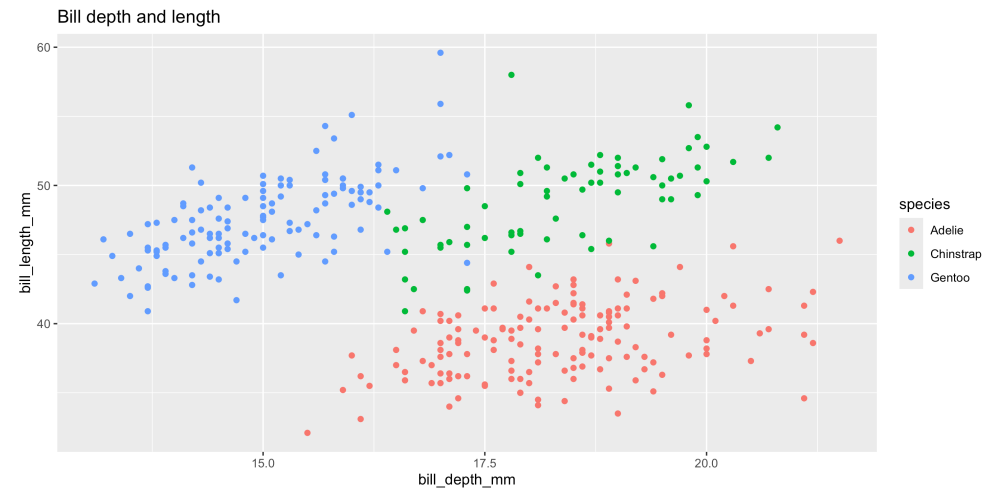
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point *and map species to the color of each point.*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8 geom_point(  
9   mapping = aes(color = species)  
10 )
```



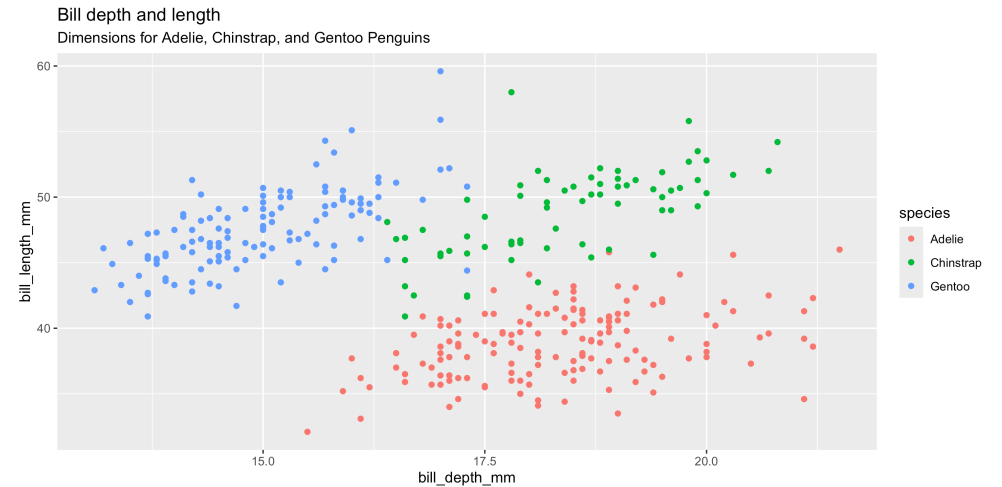
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. *Title the plot "Bill depth and length"*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(title = "Bill depth and length")
```



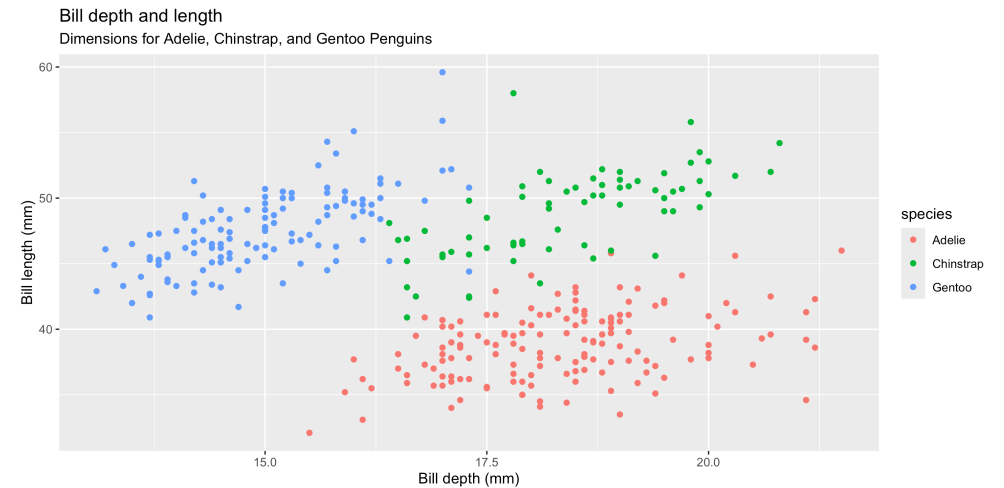
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, *add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(  
12    title = "Bill depth and length",  
13    subtitle = paste(  
14      "Dimensions for Adelie,",  
15      "Chinstrap, and Gentoo",  
16      "Penguins")  
17  )
```



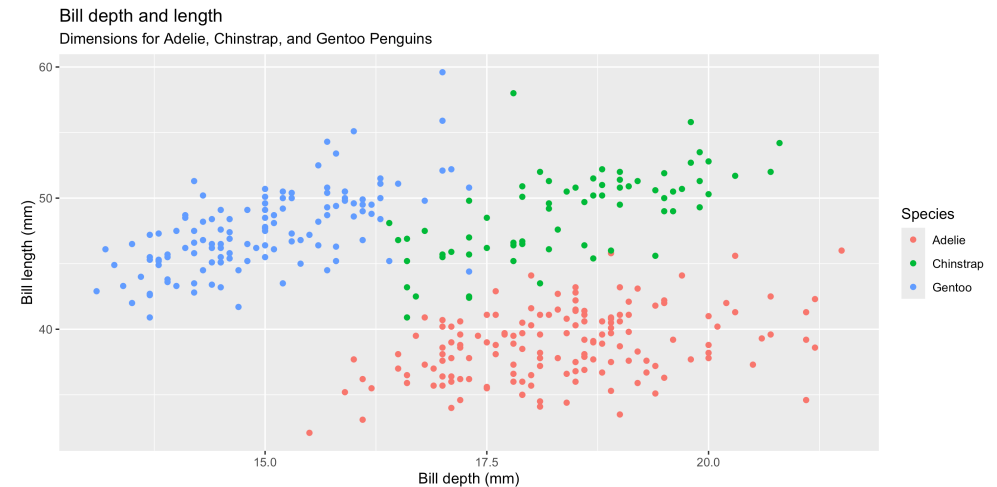
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, *label the x and y axes as “Bill depth (mm)” and “Bill length (mm)”, respectively*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(  
12    title = "Bill depth and length",  
13    subtitle = paste(  
14      "Dimensions for Adelie,",  
15      "Chinstrap, and Gentoo",  
16      "Penguins"),  
17    x = "Bill depth (mm)",  
18    y = "Bill length (mm)"  
19  )
```



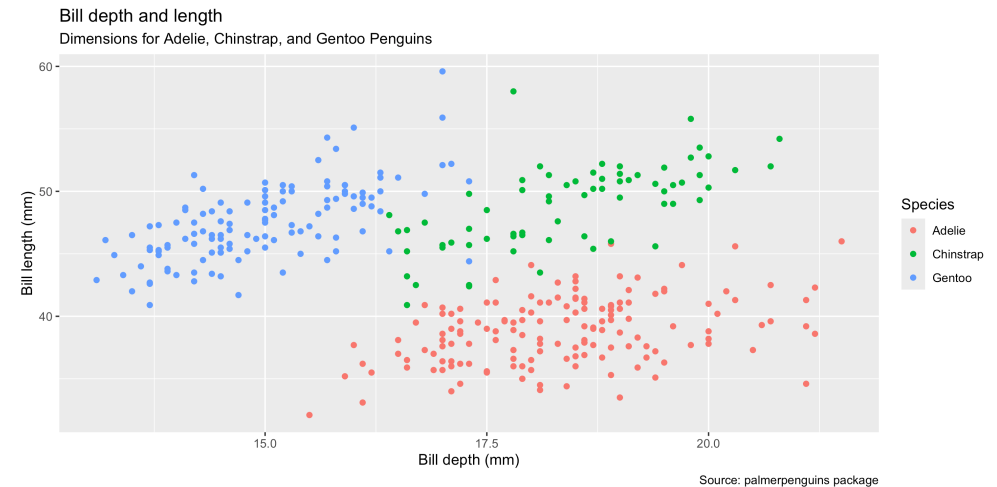
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label the x and y axes as “Bill depth (mm)” and “Bill length (mm)”, respectively, *label the legend “Species”*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(  
12    title = "Bill depth and length",  
13    subtitle = paste(  
14      "Dimensions for Adelie,",  
15      "Chinstrap, and Gentoo",  
16      "Penguins"),  
17    x = "Bill depth (mm)",  
18    y = "Bill length (mm)",  
19    color = "Species"  
20  )
```



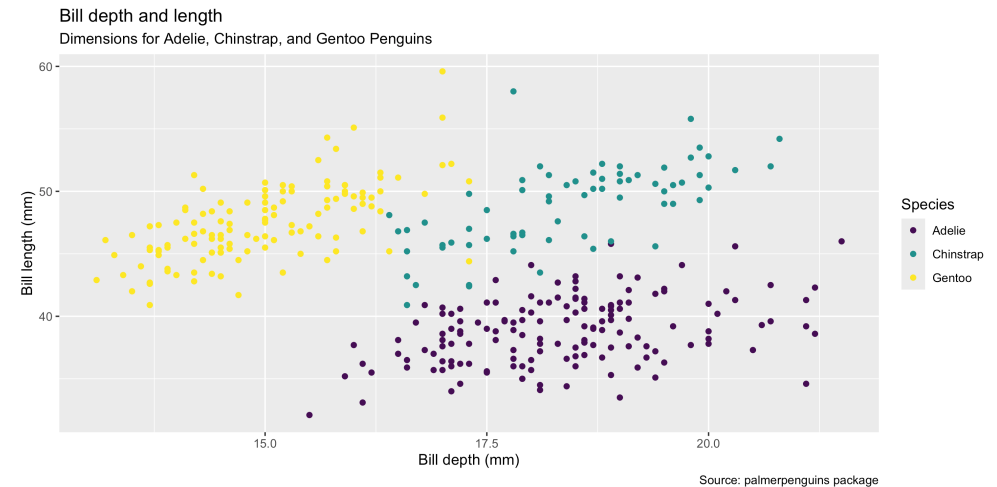
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label the x and y axes as “Bill depth (mm)” and “Bill length (mm)”, respectively, label the legend “Species”, and add a caption for the data source.

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(  
12    title = "Bill depth and length",  
13    subtitle = paste(  
14      "Dimensions for Adelie,",  
15      "Chinstrap, and Gentoo",  
16      "Penguins"),  
17    x = "Bill depth (mm)",  
18    y = "Bill length (mm)",  
19    color = "Species",  
20    caption = "Source: palmerpenguins package",  
21  )
```



Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label the x and y axes as “Bill depth (mm)” and “Bill length (mm)”, respectively, label the legend “Species”, and add a caption for the data source. *Finally, use the viridis color palette for all points.*

```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8   geom_point(  
9     mapping = aes(color = species)  
10  ) +  
11  labs(  
12    title = "Bill depth and length",  
13    subtitle = paste(  
14      "Dimensions for Adelie,",  
15      "Chinstrap, and Gentoo",  
16      "Penguins"),  
17    x = "Bill depth (mm)",  
18    y = "Bill length (mm)",  
19    color = "Species",  
20    caption = "Source: palmerpenguins pack  
21  ) +  
22  scale_color_viridis_d()
```



Aesthetics

Aesthetics options

Commonly used characteristics of plotting geometries that can be **mapped to a specific variable** in the data, examples include:

- position (`x`, `y`)
- `color`
- `shape`
- `size`
- `alpha` (transparency)

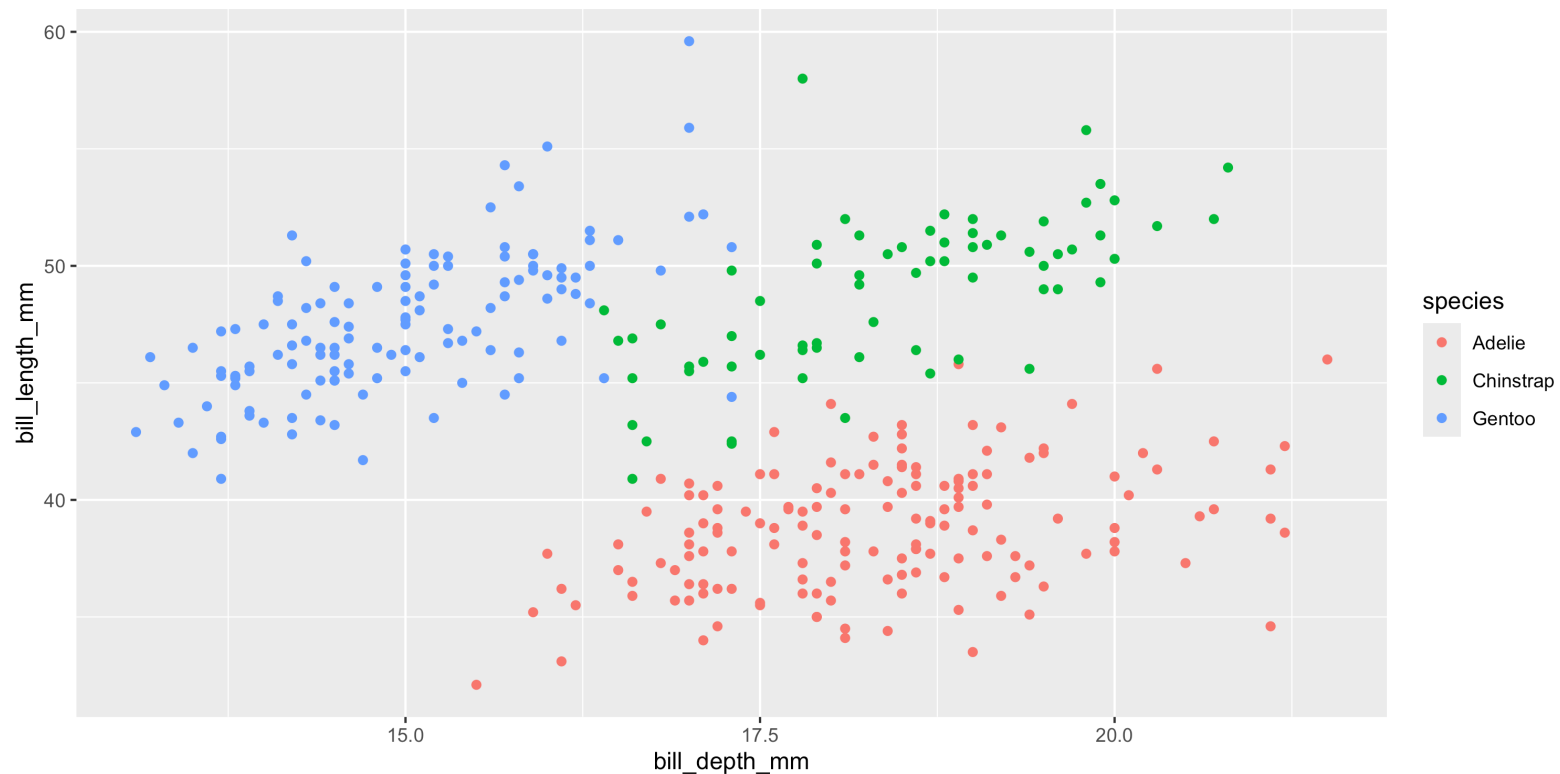
Different geometries have different aesthetics available - see the `ggplot2` `geoms` help files for listings.

- Aesthetics given in `ggplot()` apply to all `geoms`.
- Aesthetics for a specific `geom_*()` can be overridden via `mapping` or as an argument.

color

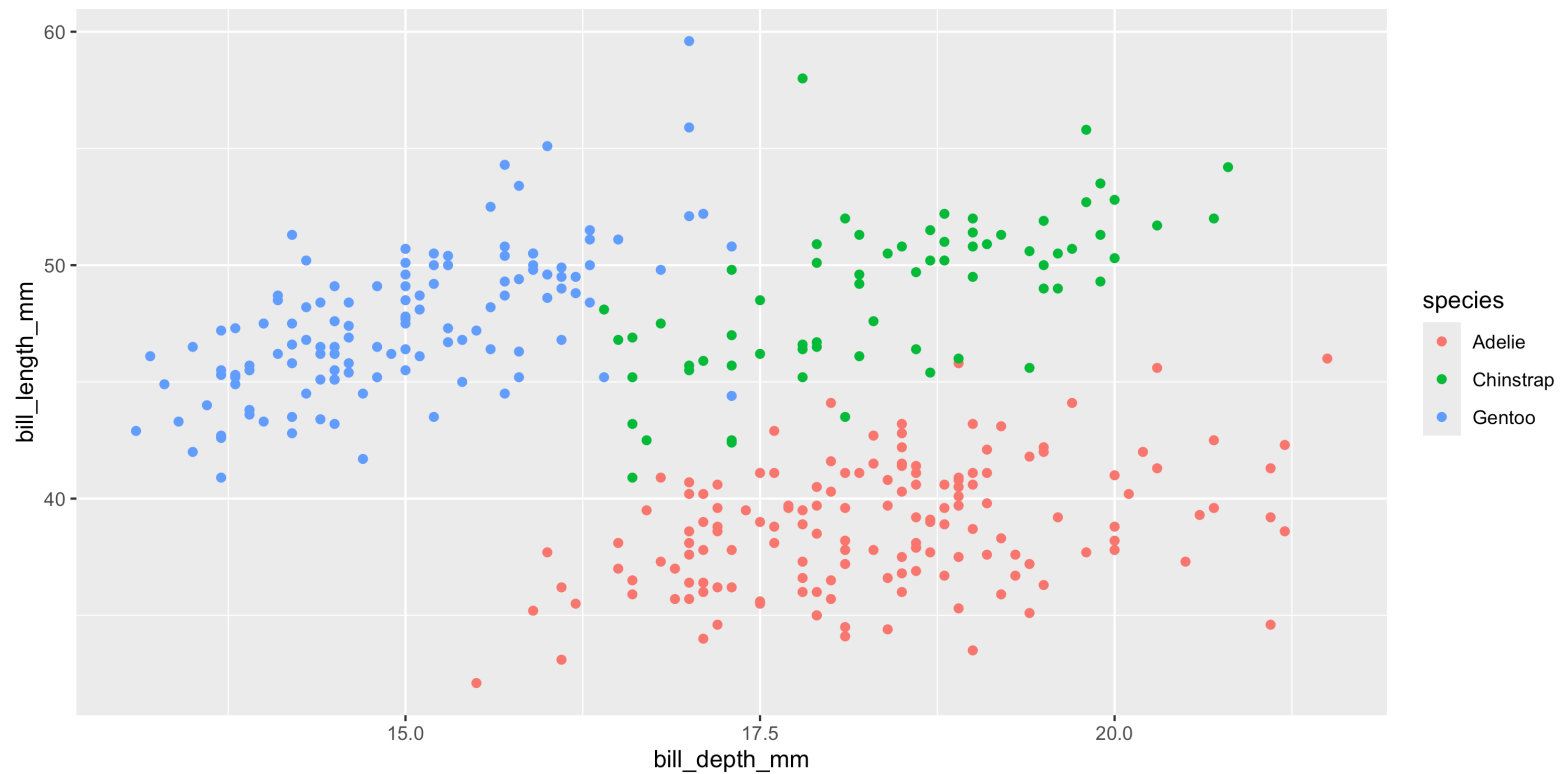
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species)  
6   )
```

Warning: Removed 2 rows containing missing values or values outside the scale range (`geom\_point()`).



Stop the warning

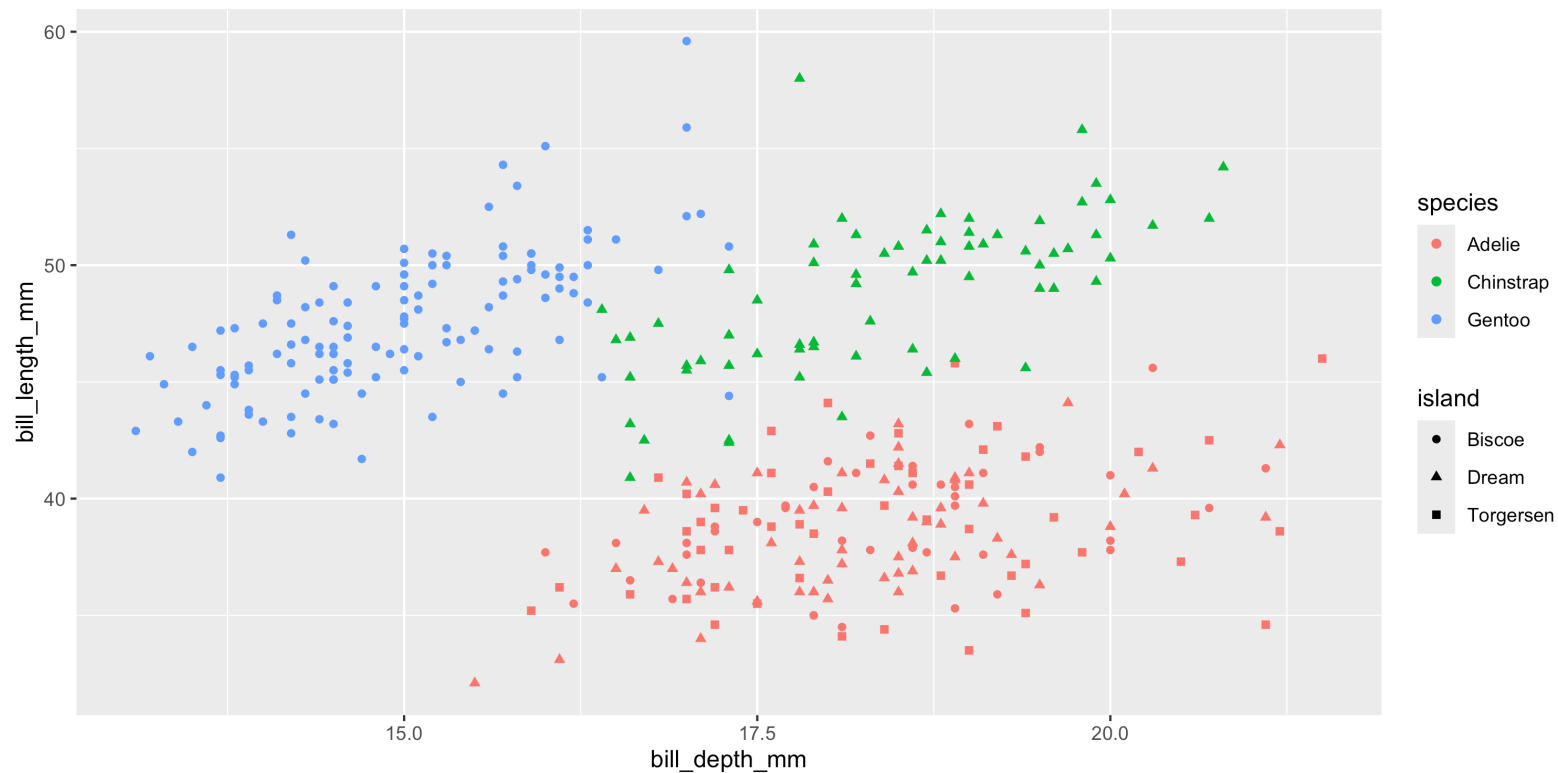
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species), na.rm=TRUE  
6   )
```



Shape

Mapped to a different variable than `color`

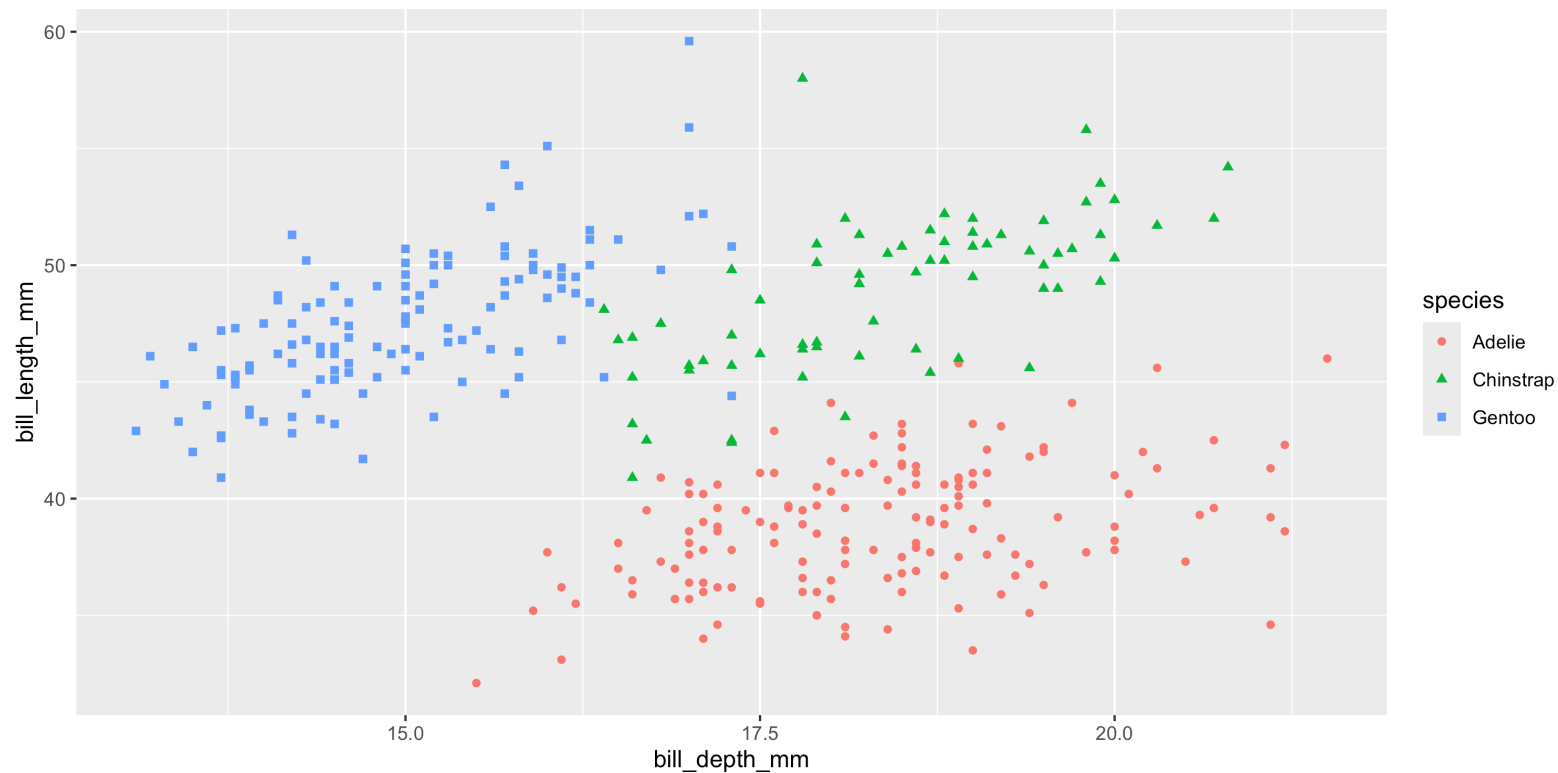
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = island), na.rm = TRUE  
6   )
```



Shape

Mapped to same variable as `color`

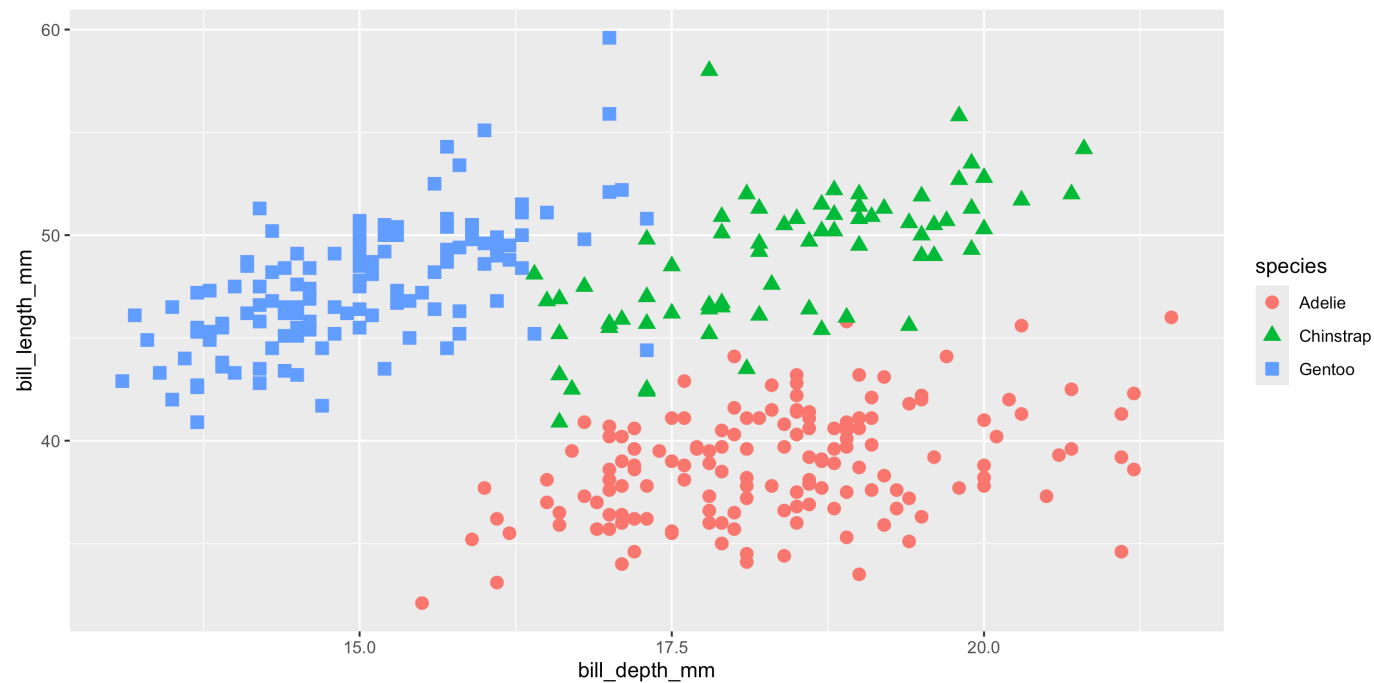
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = species), na.rm = TRUE  
6   )
```



Size

Using a fixed value - note that this value is outside of the `aes` call

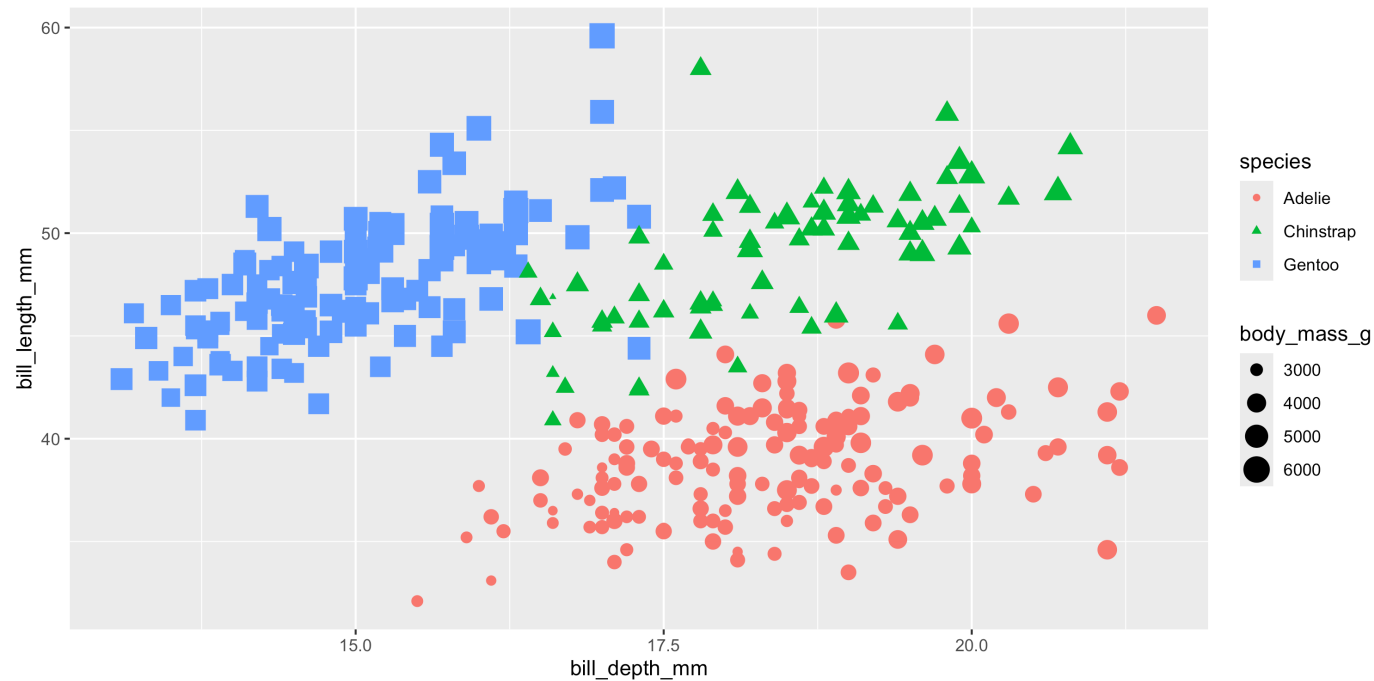
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = species), na.rm = TRUE,  
6     size = 3  
7   )
```



Size

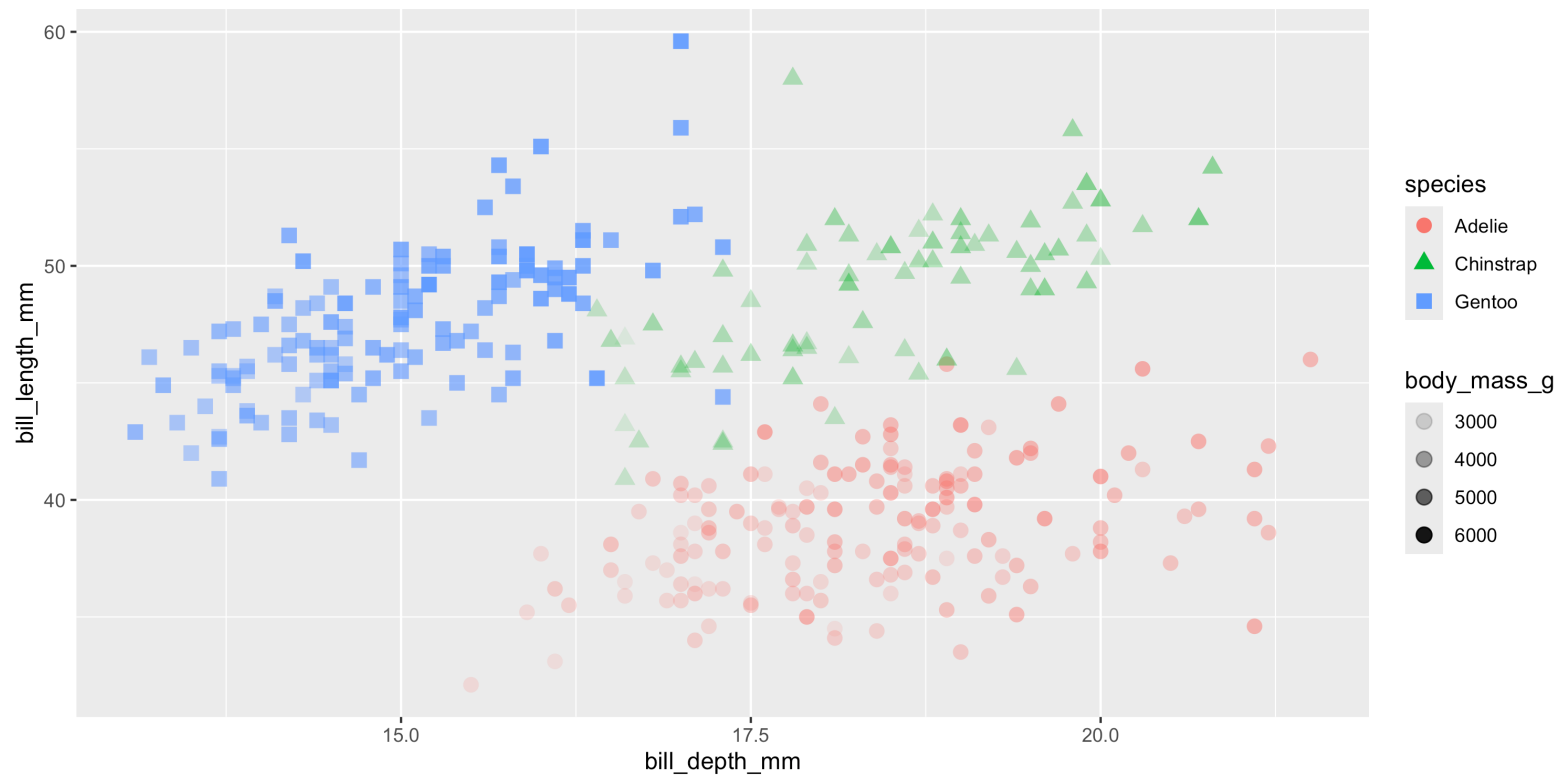
Mapped to a variable

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = species, size = body_mass_g), na.rm = TRUE  
6   )
```



Alpha

```
1 ggplot(  
2   penguins,  
3   aes(x = bill_depth_mm, y = bill_length_mm)  
4 ) +  
5   geom_point(  
6     aes(color = species, shape = species, alpha = body_mass_g), na.rm = TRUE,  
7     size = 3  
8   )
```



Mapping vs settings

- **Mapping** - Determine an aesthetic (the size, alpha, etc.) of a geom based on the values of a variable in the data
 - wrapped by `aes()` and pass as `mapping` argument to `ggplot()` or `geom_*()`.
- **Setting** - Determine an aesthetic (the size, alpha, etc.) of a geom using a constant value not directly from the data.
 - passed directly into `geom_*()` as an argument.

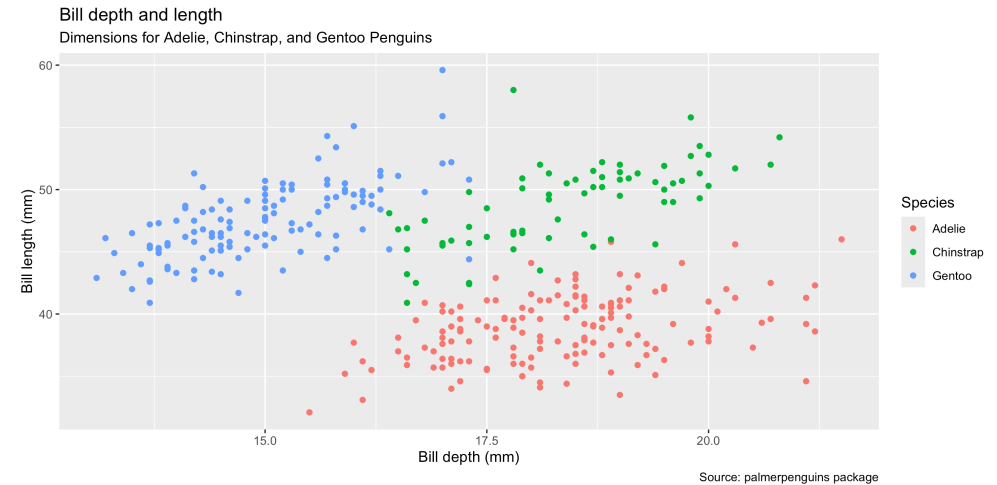
From the previous slide `color`, `shape`, and `alpha` are all **aesthetics** while `size` was a **setting**.

Labels

labs()

In our previous example we saw the use of `labs()` to provide human readable labels to various plot elements.

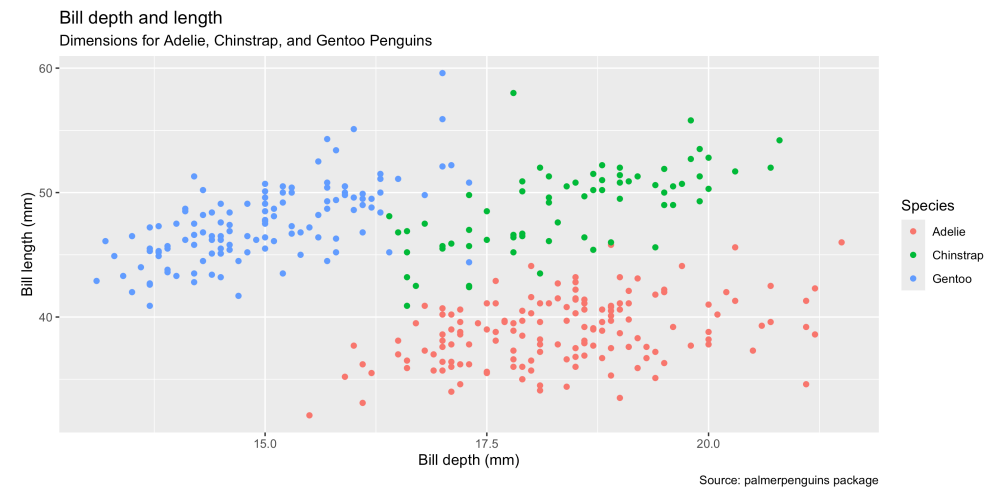
```
1 ggplot(  
2   data = penguins,  
3   mapping = aes(  
4     x = bill_depth_mm,  
5     y = bill_length_mm  
6   )  
7 ) +  
8 geom_point(  
9   mapping = aes(color = species), na.rm = TRUE  
10 ) +  
11 labs(  
12   title = "Bill depth and length",  
13   subtitle = paste(  
14     "Dimensions for Adelie,",  
15     "Chinstrap, and Gentoo",  
16     "Penguins"),  
17   x = "Bill depth (mm)",  
18   y = "Bill length (mm)",  
19   color = "Species",  
20   caption = "Source: palmerpenguins package"  
21 )
```



Labels

Instead of overriding with `labs()` we can instead annotate the data so that the label is generated automatically, by attaching a `label` argument to the appropriate column in our data frame.

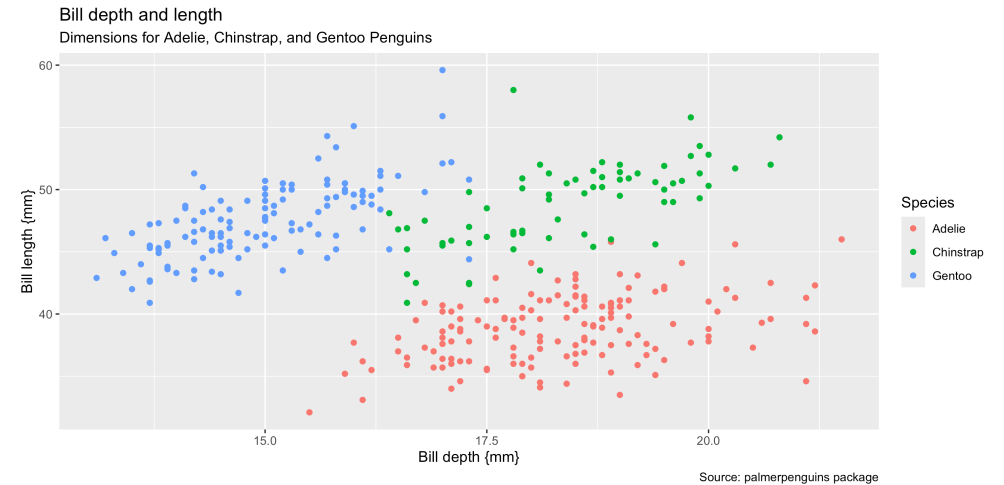
```
1 p_labeled = penguins
2 attr(p_labeled$species, "label") = "Species"
3 attr(p_labeled$bill_depth_mm, "label") = "Bill depth (mm)"
4 attr(p_labeled$bill_length_mm, "label") = "Bill length (mm)"
5
6 ggplot(
7   data = p_labeled,
8   mapping = aes(
9     x = bill_depth_mm,
10    y = bill_length_mm
11  )
12 ) +
13   geom_point(
14     mapping = aes(color = species), na.rm = TRUE
15   ) +
16   labs(
17     title = "Bill depth and length",
18     subtitle = paste(
19       "Dimensions for Adelie,",
20       "Chinstrap, and Gentoo",
21       "Penguins"),
22     caption = "Source: palmerpenguins package"
23   )
```



Dictionary

Alternatively, we can provide a dictionary / lookup table to `labs()` dictionary argument, which will then be used.

```
1 lookup = c(  
2   species = "Species",  
3   bill_depth_mm = "Bill depth {mm}",  
4   bill_length_mm = "Bill length {mm}"  
5 )  
6  
7 ggplot(  
8   data = penguins,  
9   mapping = aes(  
10     x = bill_depth_mm,  
11     y = bill_length_mm  
12   )  
13 ) +  
14   geom_point(  
15     mapping = aes(color = species), na.rm = TRUE  
16   ) +  
17   labs(  
18     title = "Bill depth and length",  
19     subtitle = paste(  
20       "Dimensions for Adelie,",  
21       "Chinstrap, and Gentoo",  
22       "Penguins"),  
23     caption = "Source: palmerpenguins package",  
24     dictionary = lookup  
25   )
```



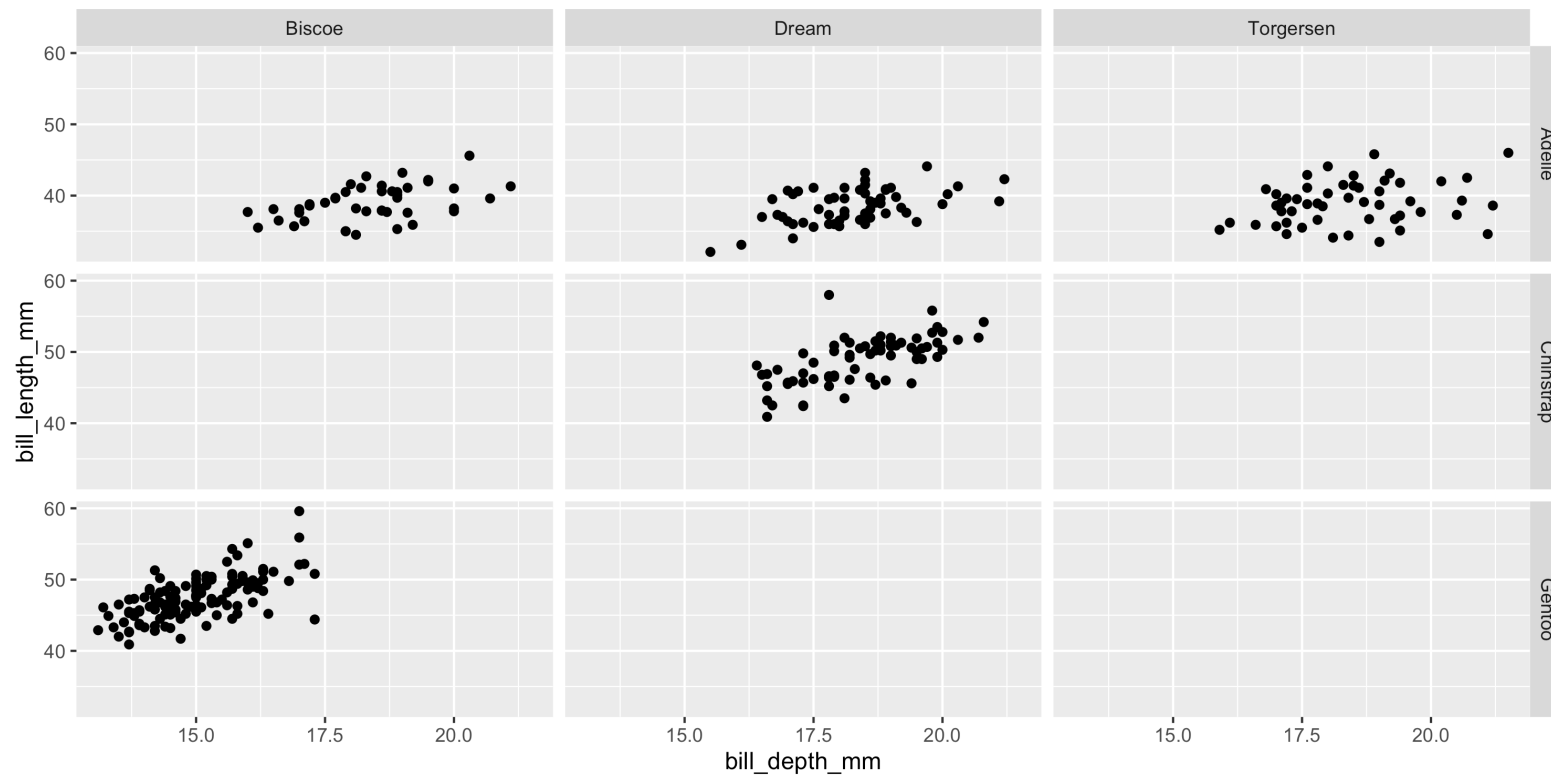
Faceting

Faceting

- Smaller plots that display different subsets of the data
- Useful for exploring conditional relationships and large data
- Sometimes referred to as “small multiples”

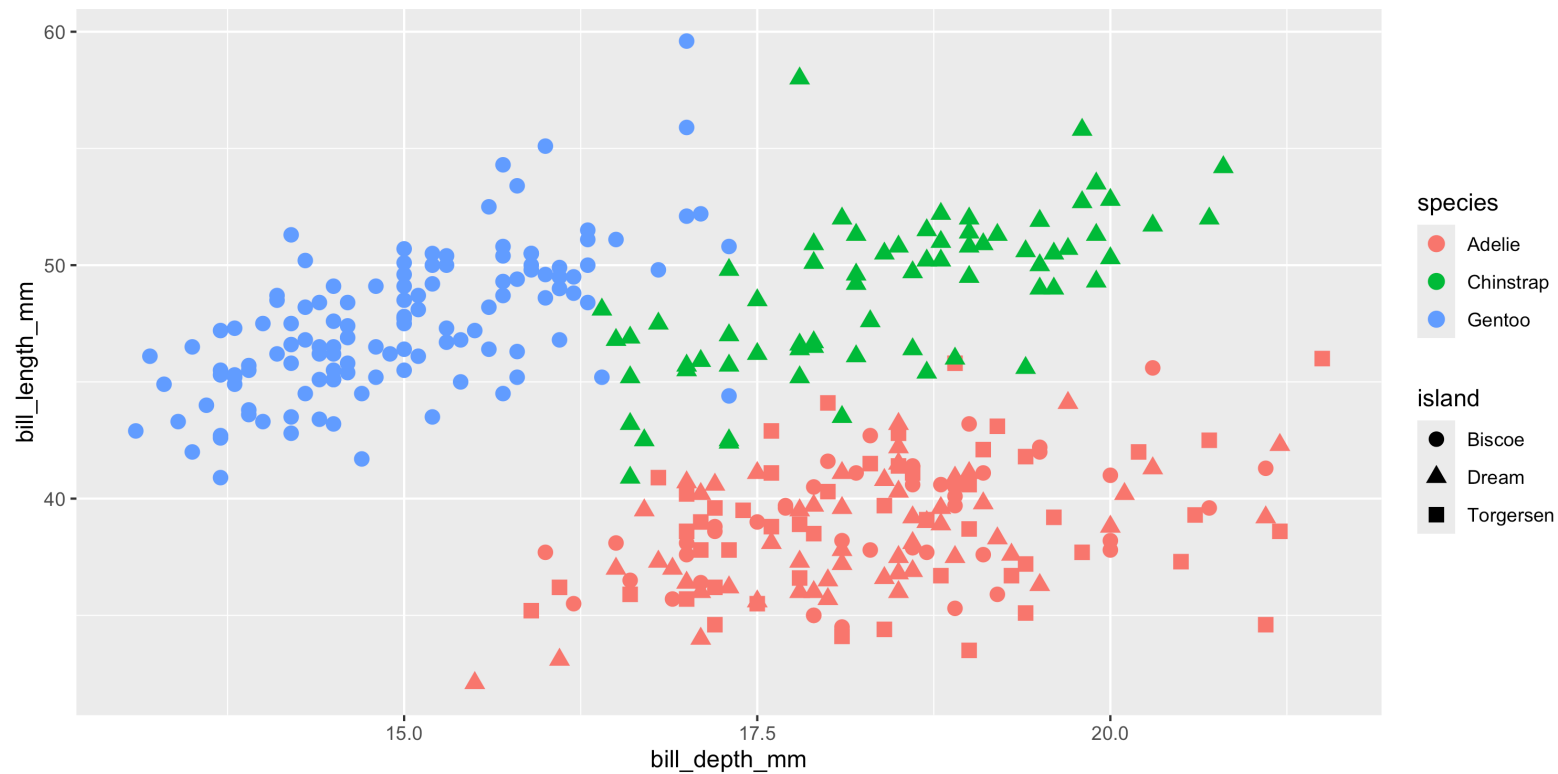
facet_grid

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(  
6     species ~ island  
7   )
```



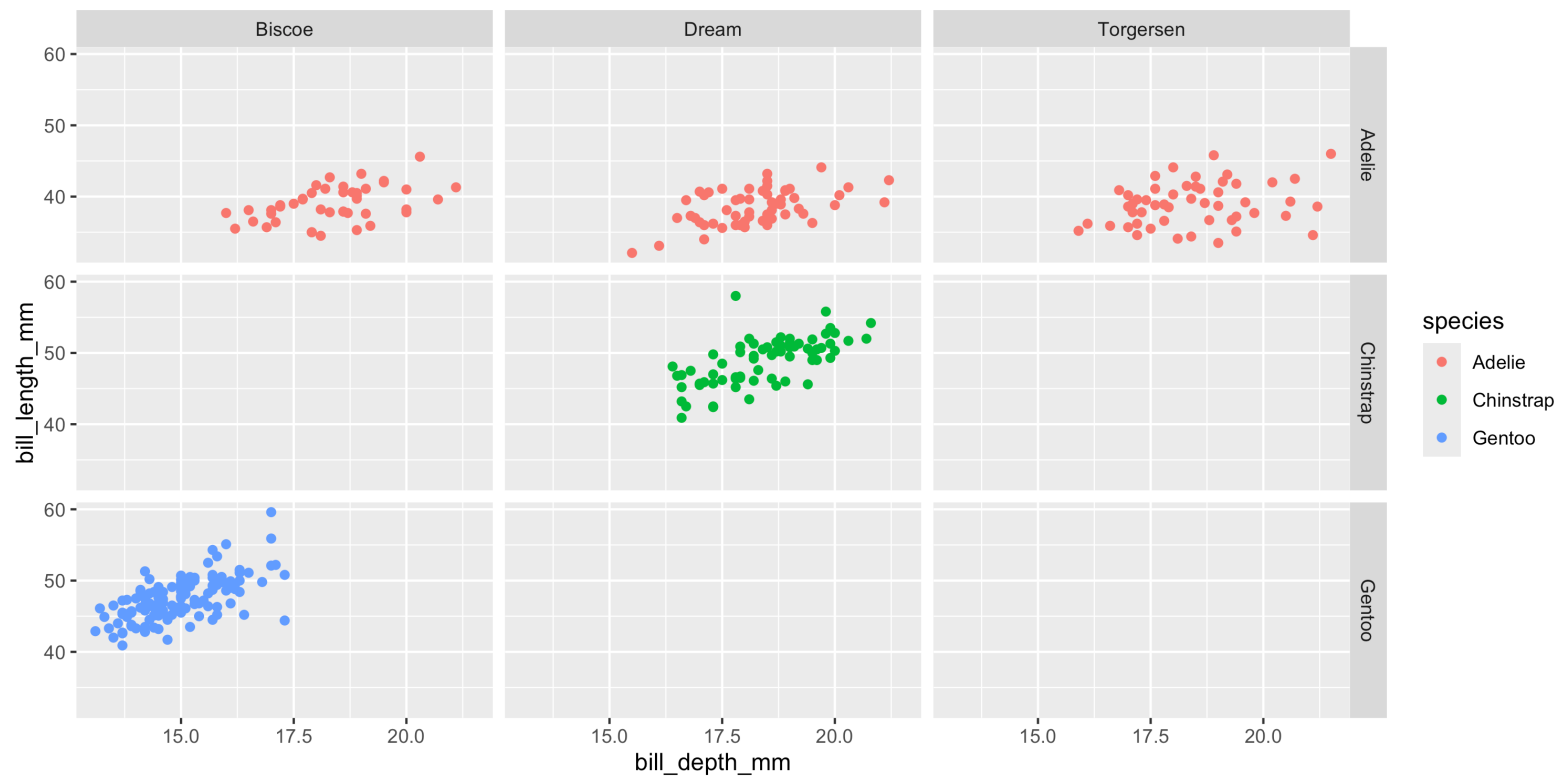
Compare with ...

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = island), na.rm = TRUE, size = 3  
6   )
```



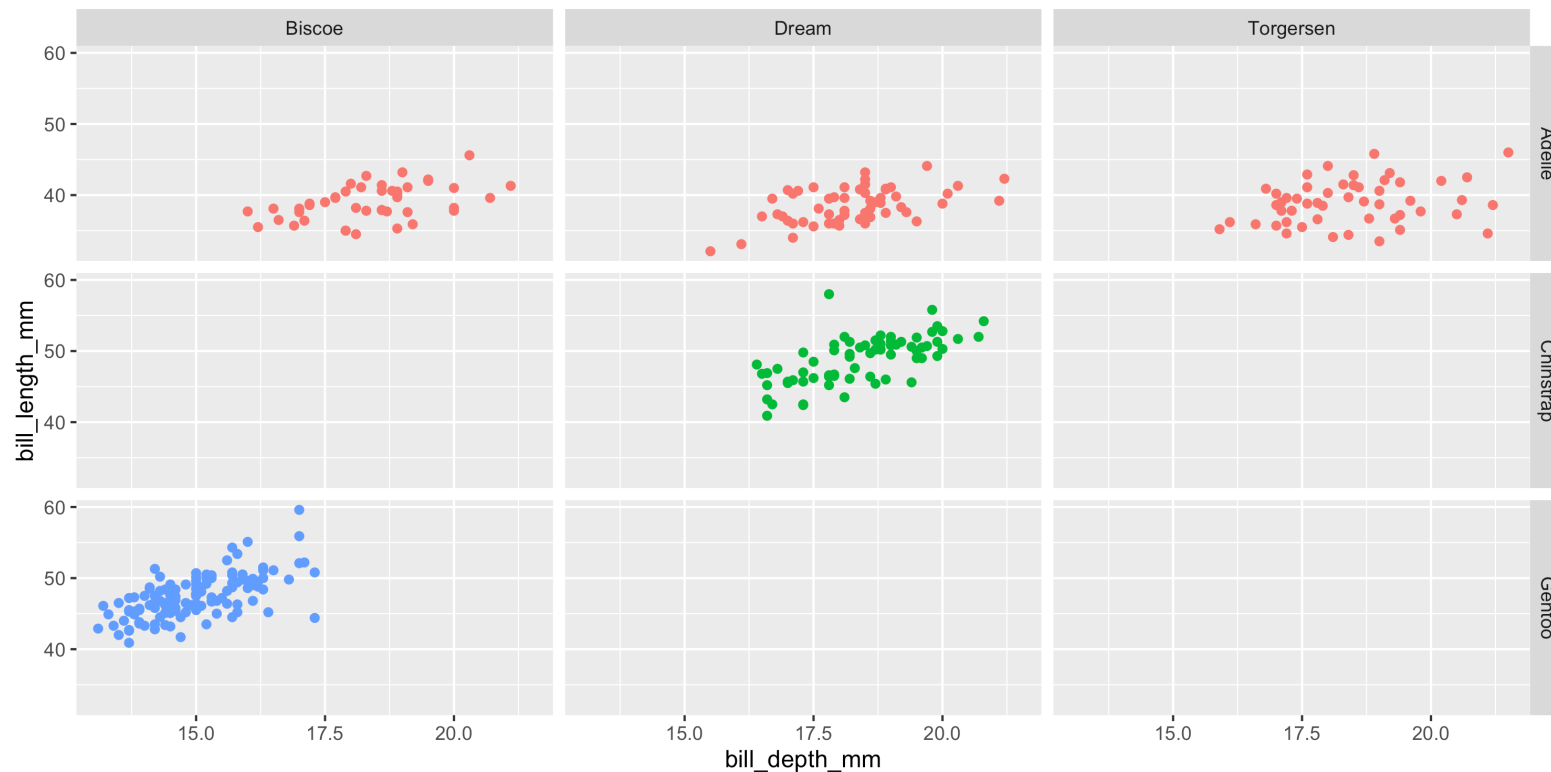
Faceting and color

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(species ~ island)
```



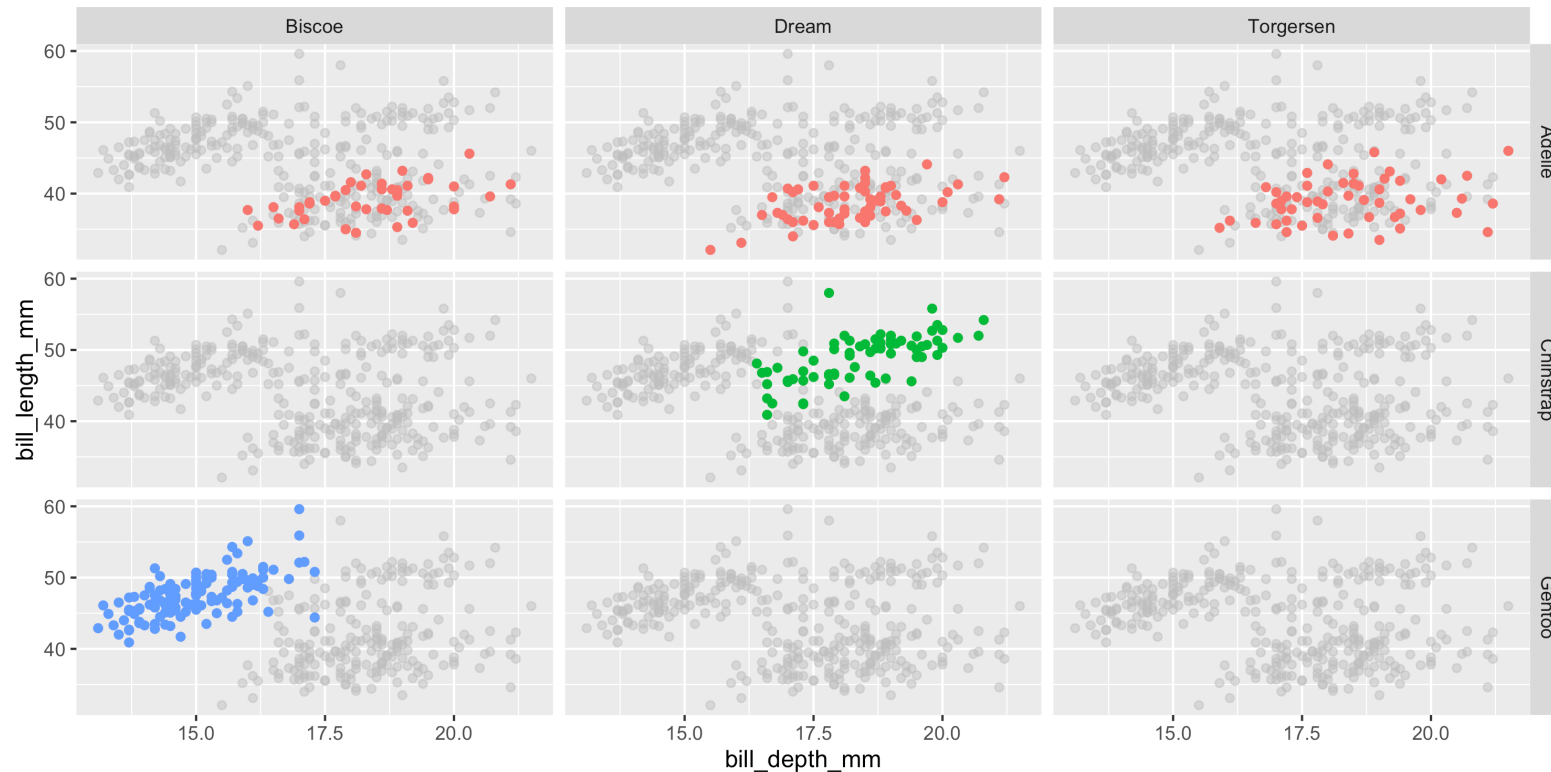
Hiding legend elements

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(species ~ island) +  
6   guides(color = "none")
```



Facet layout - context

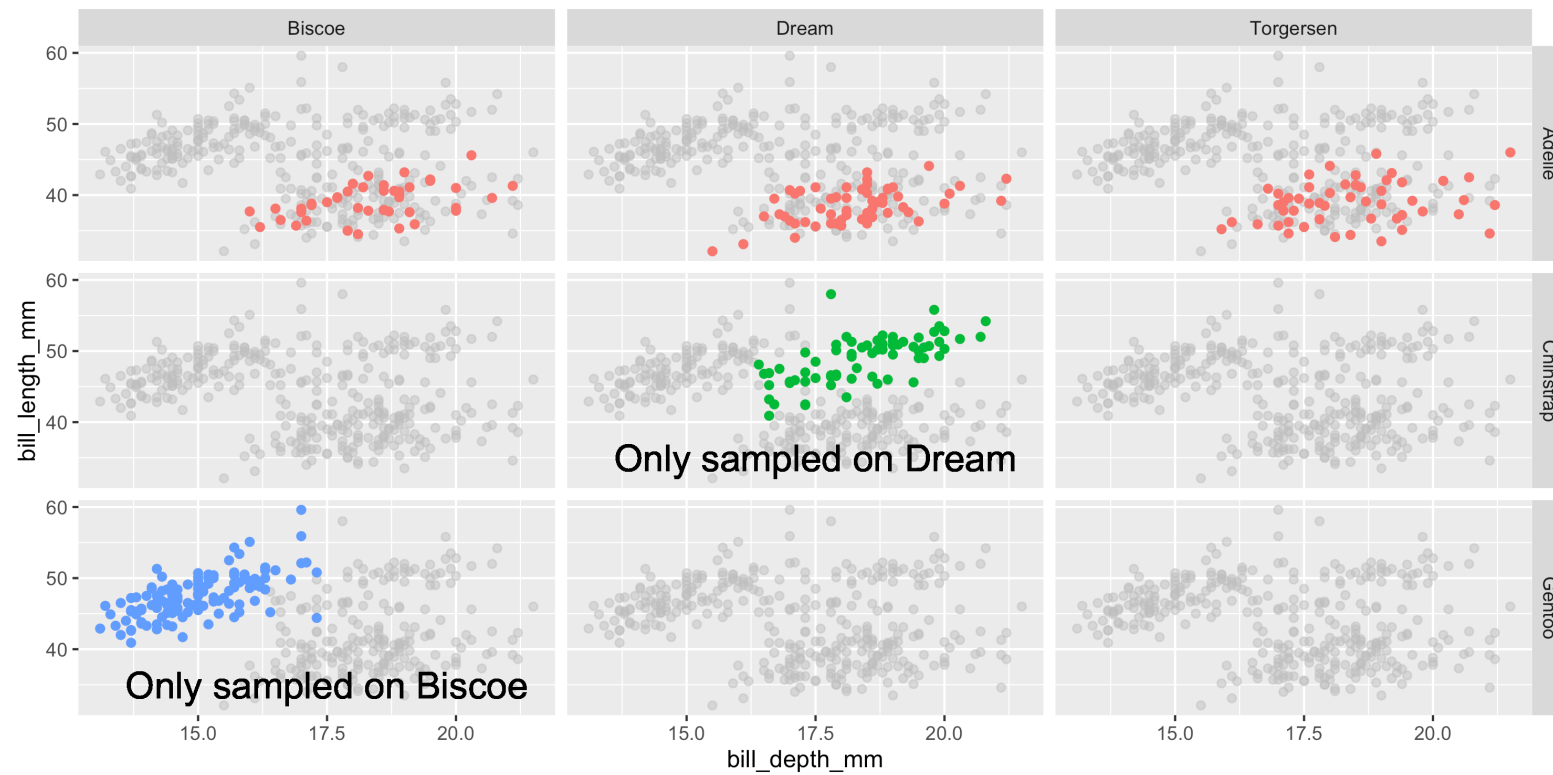
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +  
5   geom_point(na.rm = TRUE) +  
6   facet_grid(species ~ island) +  
7   guides(color = "none")
```



Facet layout - annotation

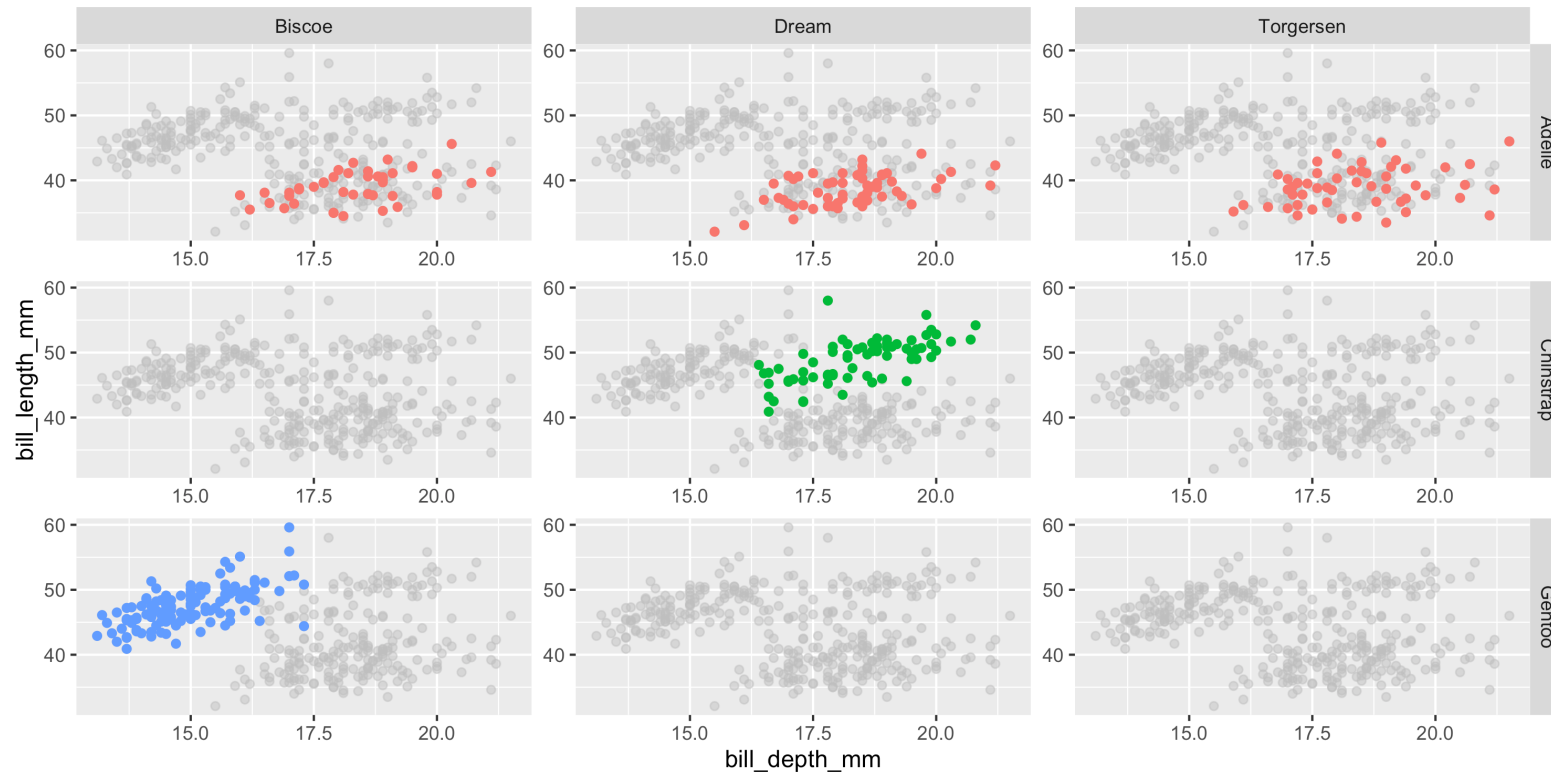
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +  
5   geom_point(na.rm = TRUE) +  
6   facet_grid(species ~ island) +  
7   guides(color = "none") +  
8   geom_text(  
9     x = 17.5, y = 35, label = "Only sampled on Dream", size = 6, color = "black",  
10    layout = 5  
11 ) +  
12   geom_text(  
13     x = 17.5, y = 35, label = "Only sampled on Biscoe", size = 6, color = "black",  
14     layout = 7  
15 )
```

Facet layout - annotation



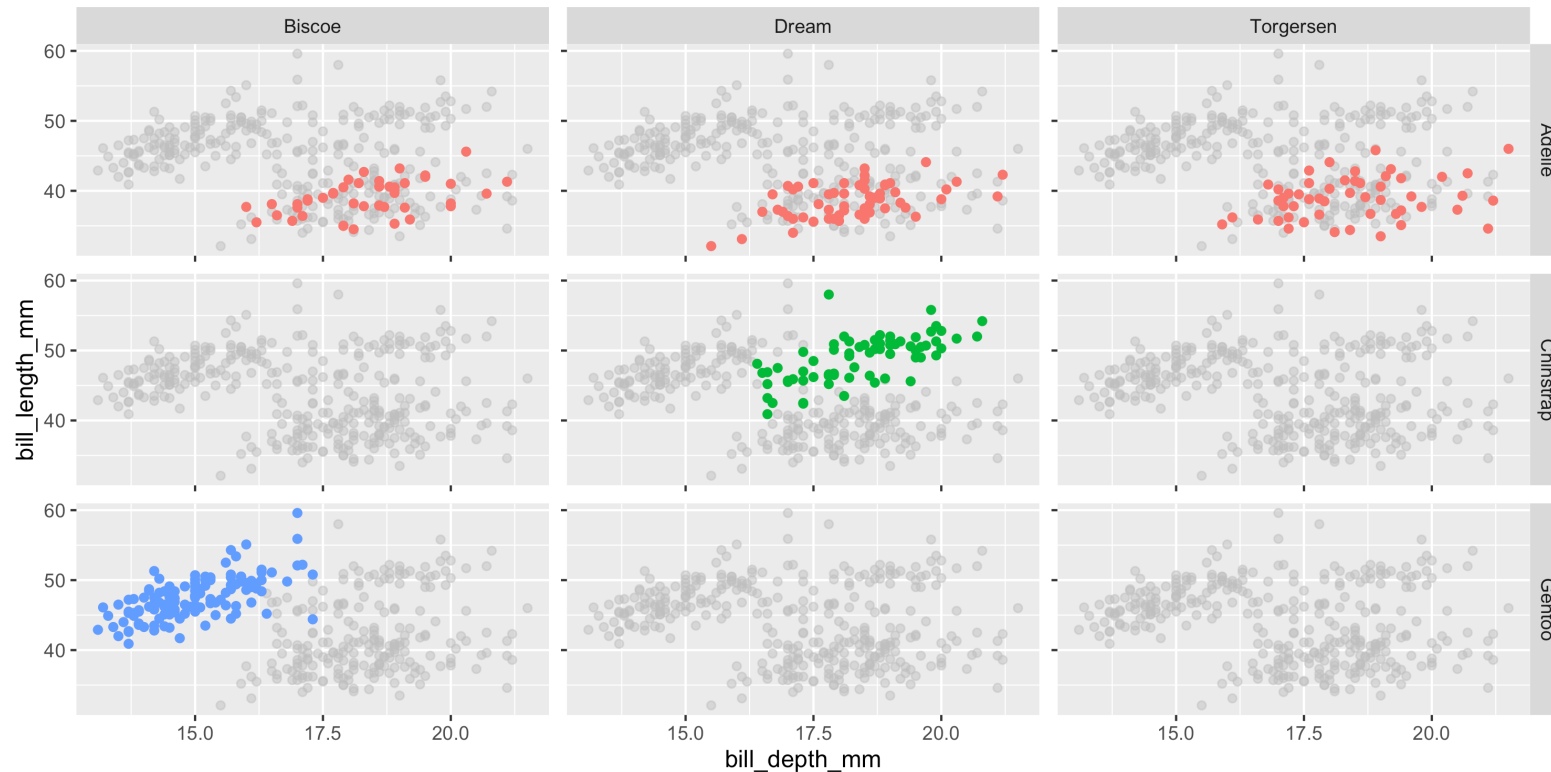
Facet axes

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +  
5   geom_point(na.rm = TRUE) +  
6   facet_grid(species ~ island, axes = "all") +  
7   guides(color = "none")
```



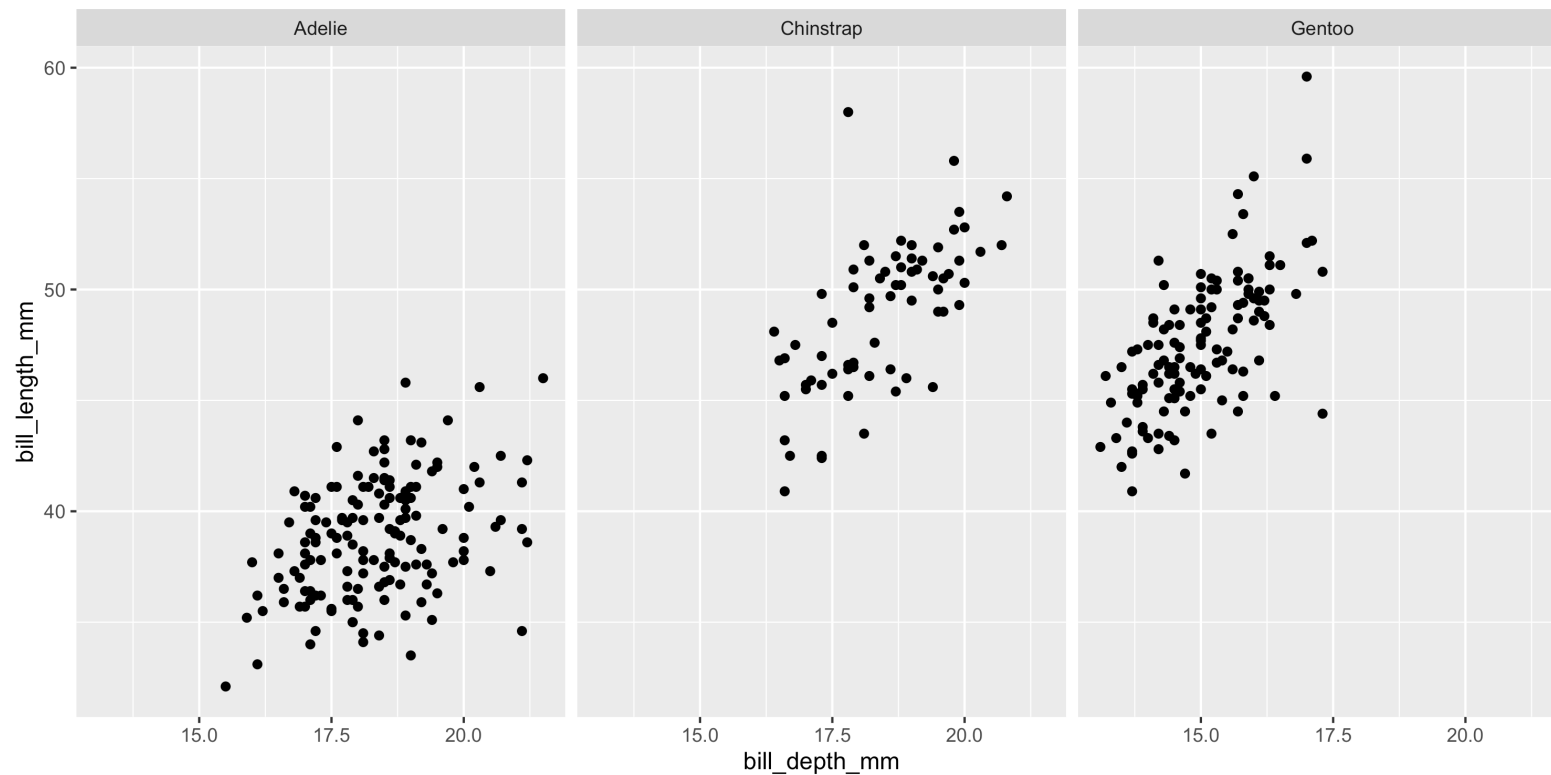
Facet axes - labels

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +  
5   geom_point(na.rm = TRUE) +  
6   facet_grid(species ~ island, axes = "all", axis.labels = "margins") +  
7   guides(color = "none")
```



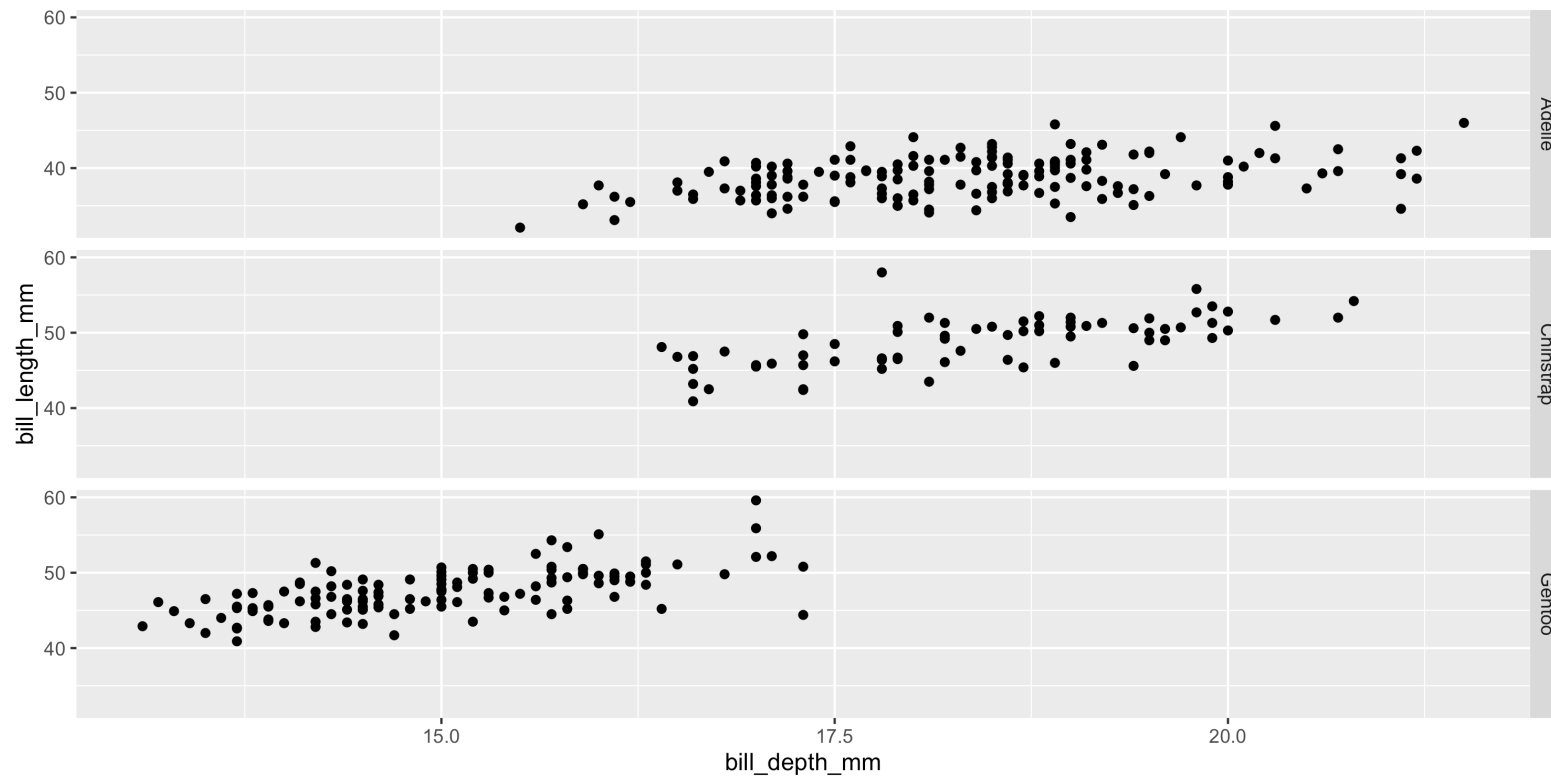
facet_grid (columns)

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(~ species)
```



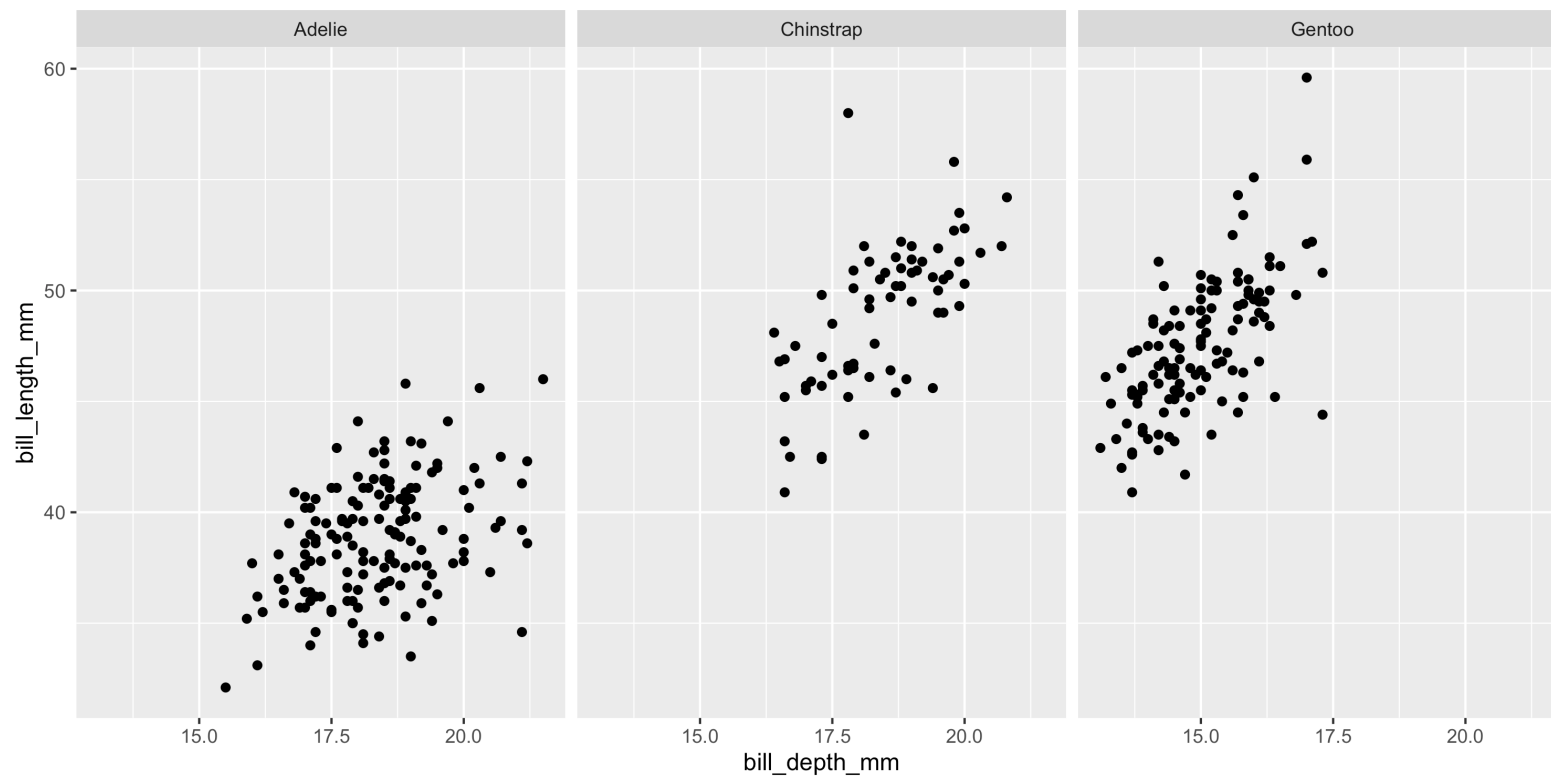
facet_grid (rows)

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(species ~ .)
```



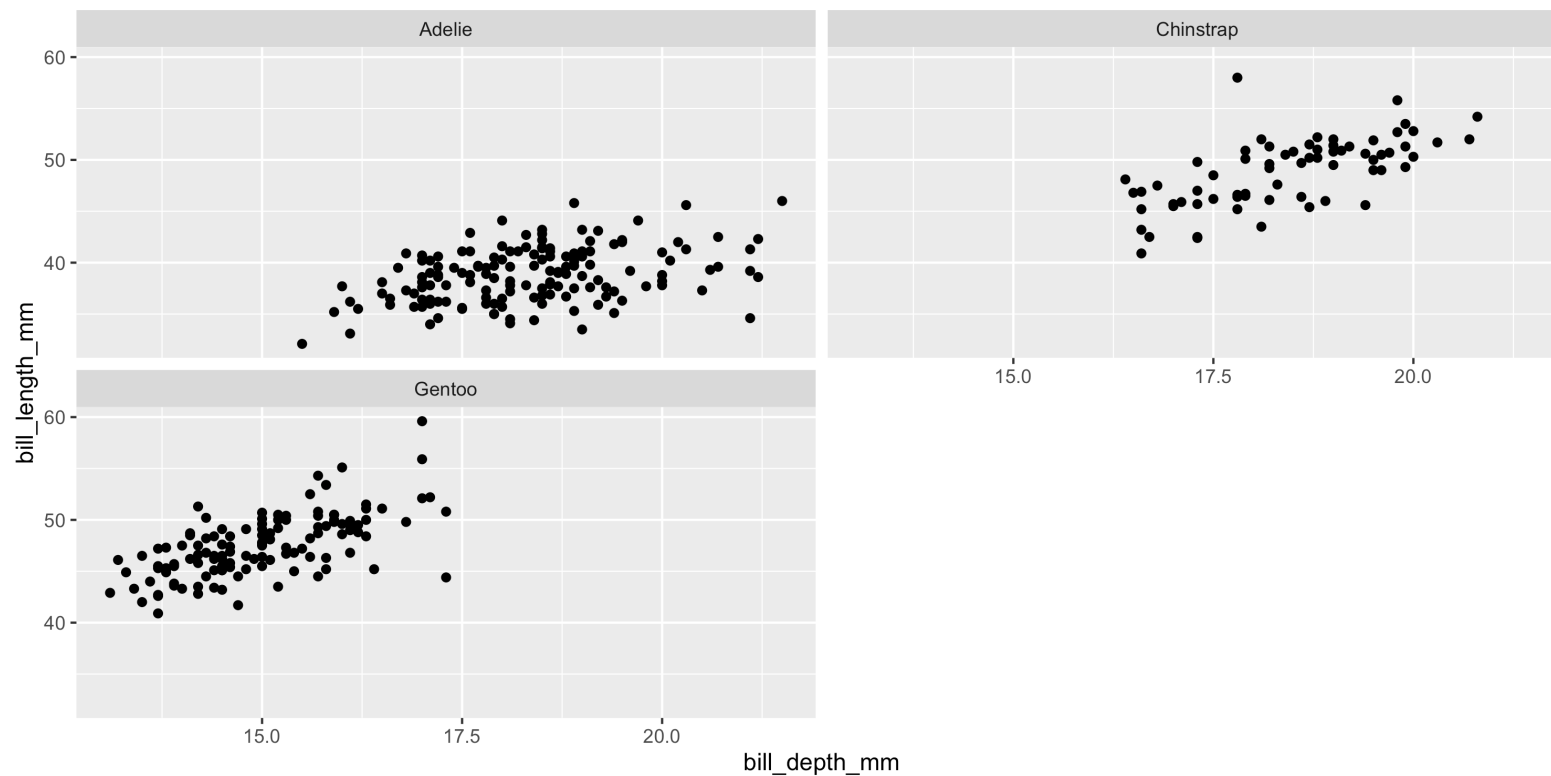
facet_wrap

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_wrap(~ species)
```



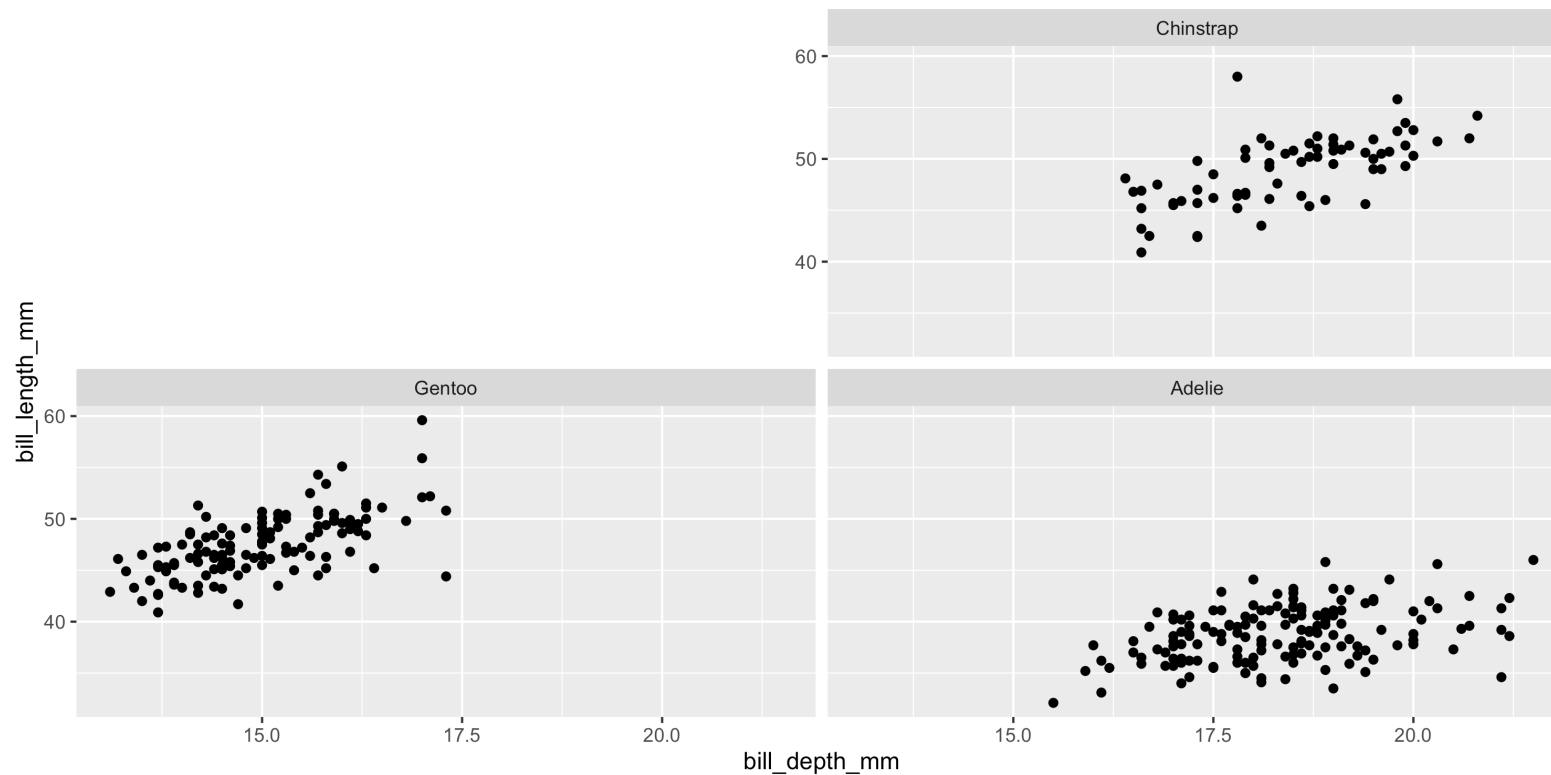
facet_wrap

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_wrap(~ species, ncol = 2)
```



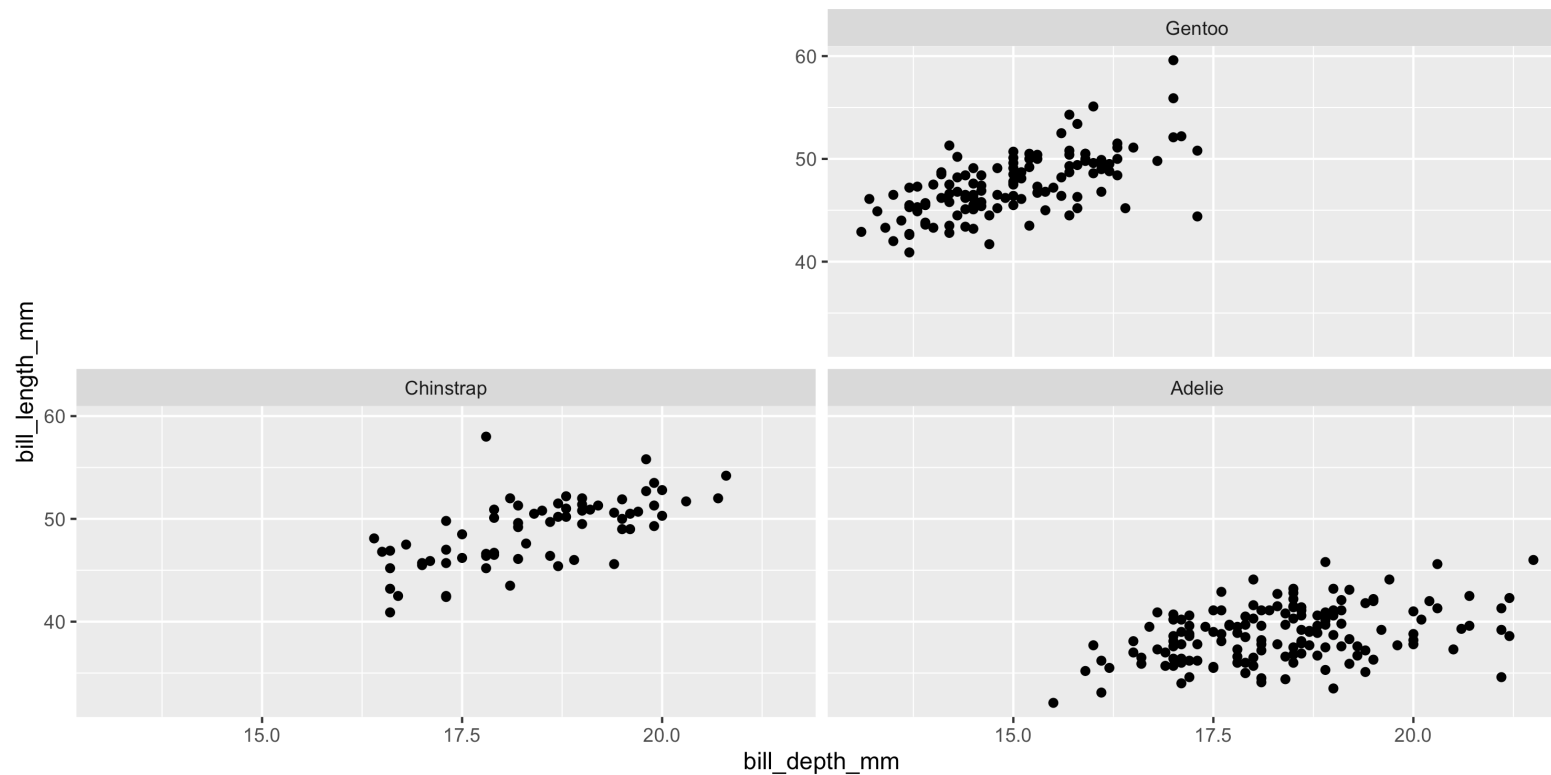
facet_wrap - direction

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_wrap(~ species, ncol = 2, dir = "br")
```



facet_wrap - direction

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_wrap(~ species, ncol = 2, dir = "rb")
```



Learning more

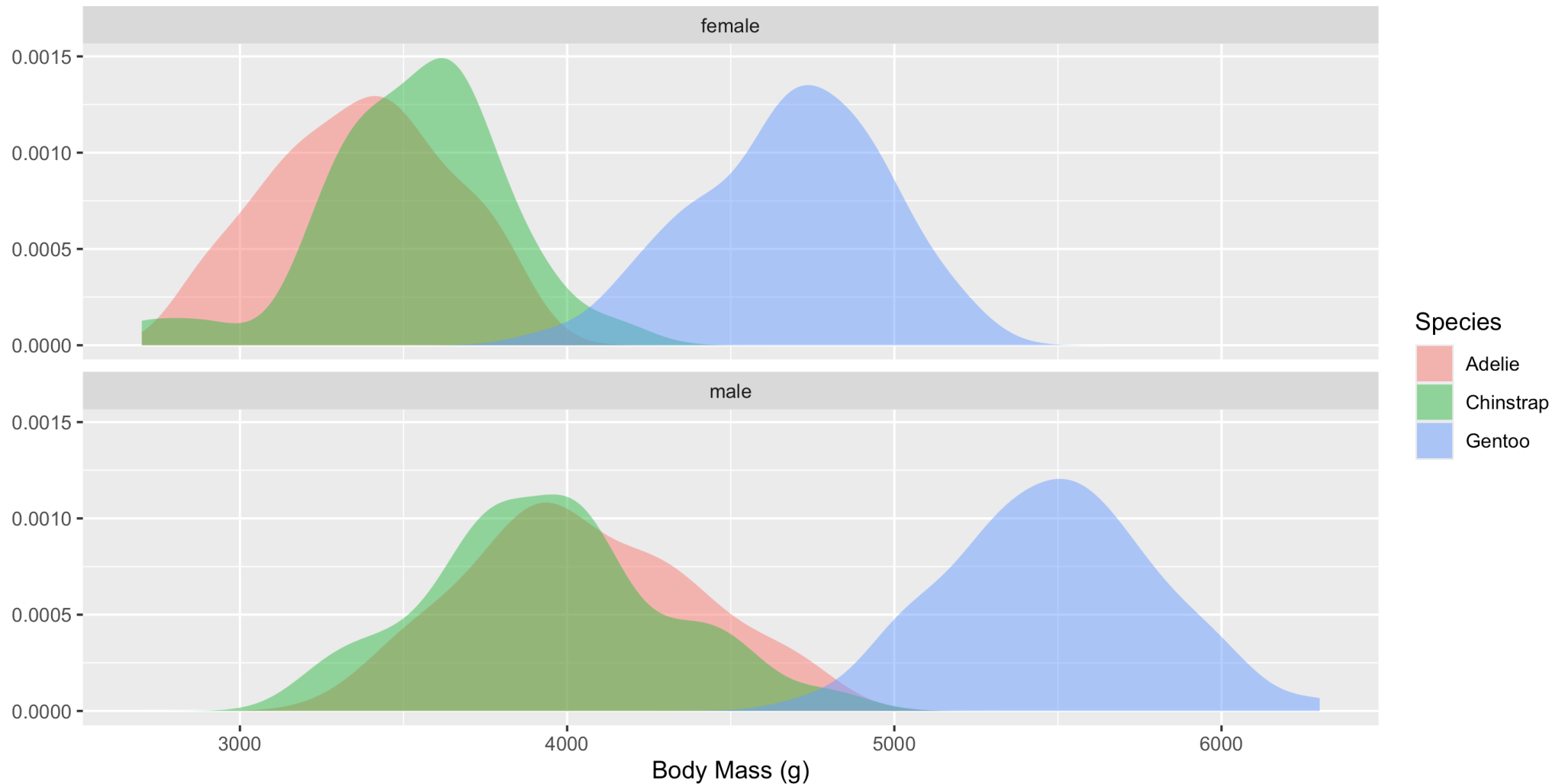
geom tour

ggplot2.tidyverse.org/reference/index.html

Exercises

Exercise 1

Recreate, as faithfully as possible, the following plot using ggplot2 and the [penguins](#) data.



Exercise 2

Recreate, as faithfully as possible, the following plot from the [palmerpenguin](#) package README in ggplot2.

