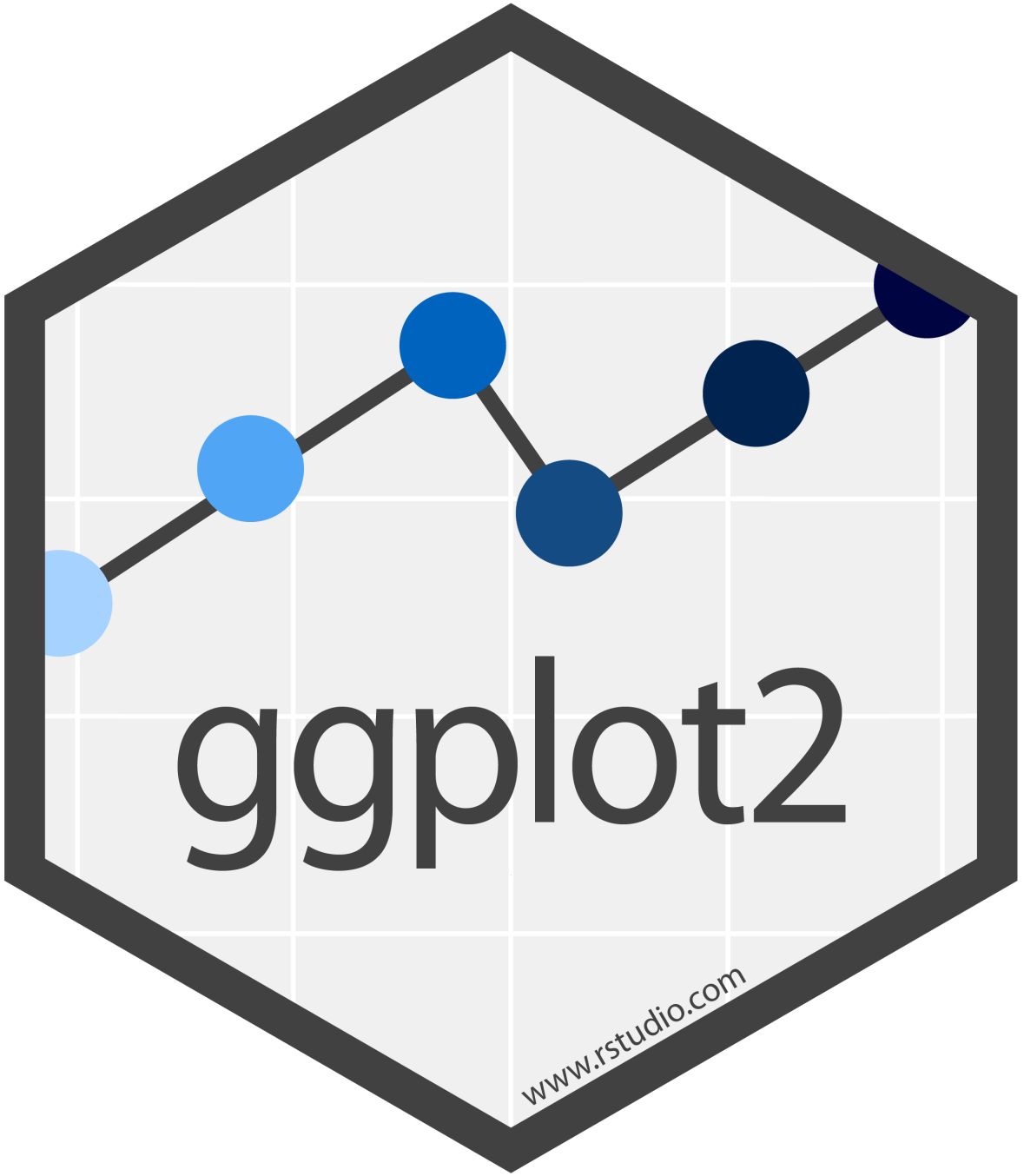
Visualization with ggplot2

Lecture 10

Dr. Colin Rundel


ggplot2
www.rstudio.com

The Grammar of Graphics

- Conceptualized by Leland Wilkinson in 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 - breaks down graphs into semantic components
  - not meant as a guide on which graph to use or how to best convey your data (more on that next time), but it does have some strong opinions.

Terminology

A statistical graphic is a…

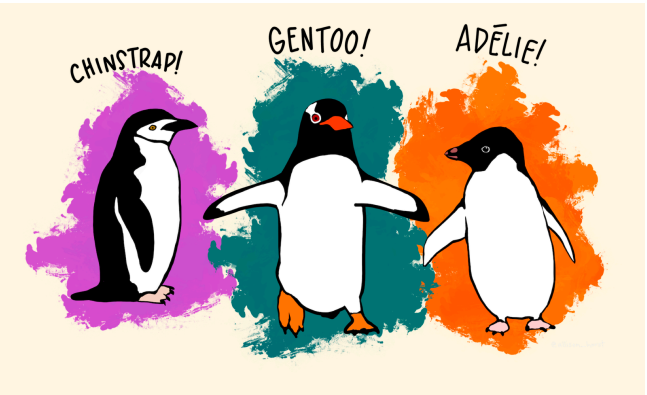
- mapping of **data**
- which may be **statistically transformed** (summarized, log-transformed, etc.)
- to **aesthetic attributes** (color, size, xy-position, etc.)
- using **geometric objects** (points, lines, bars, etc.)
- and mapped onto a specific **facet** and **coordinate system**

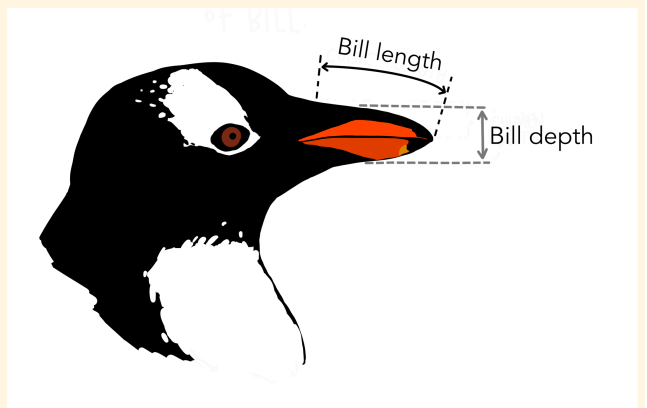
Anatomy of a ggplot call

```
ggplot(
  data = [dataframe],
  mapping = aes(
    x = [var x], y = [var y],
    color = [var color],
    shape = [var shape],
    ...
  )
) +
  geom_[some geom](
    mapping = aes(
      color = [var geom color],
      ...
    )
  ) +
  ... # other geometries
  scale_[some axis]_[some scale]() +
  facet_[some facet]([formula]) +
  ... # other options
```

Data - Palmer Penguins

Measurements of penguin species, island in the 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>
```

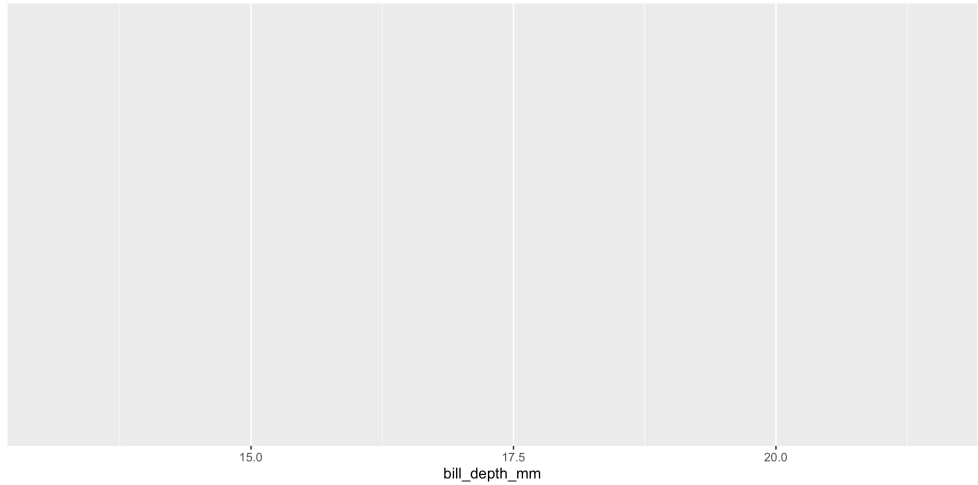
Text <-> Plot

> *Start with the `penguins` data frame*

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

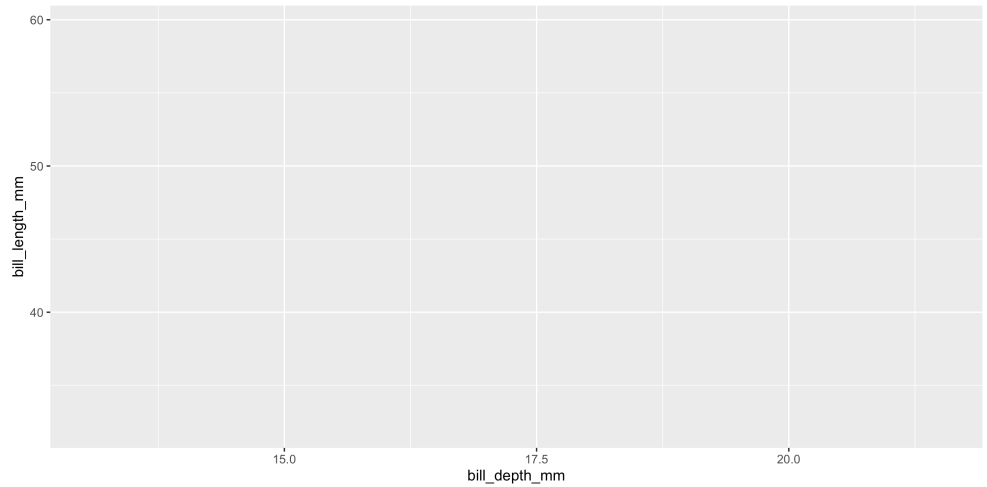
> Start with the `penguins` data frame, *map bill depth to the x-axis*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm
  )
)
```



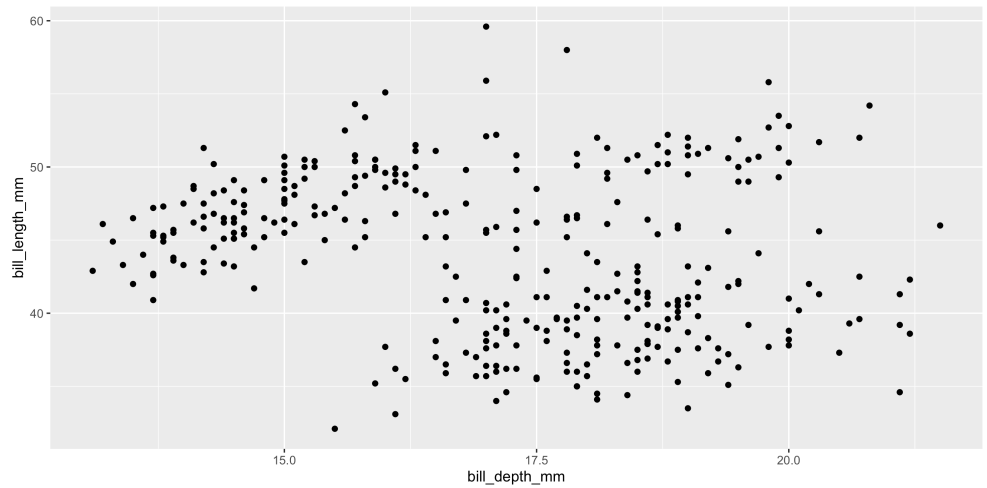
> Start with the `penguins` data frame, map bill depth to the x-axis *and map bill length to the y-axis.*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
)
```



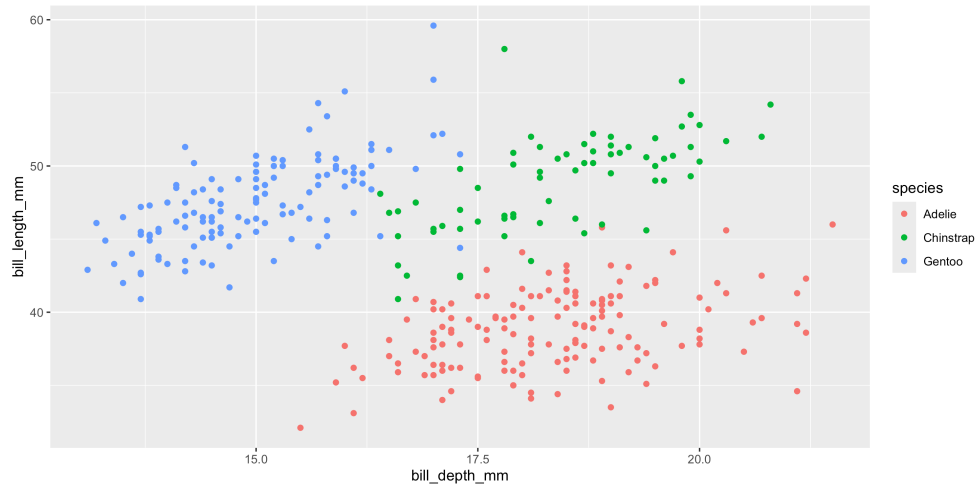
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. *Represent each observation with a point*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point()
```



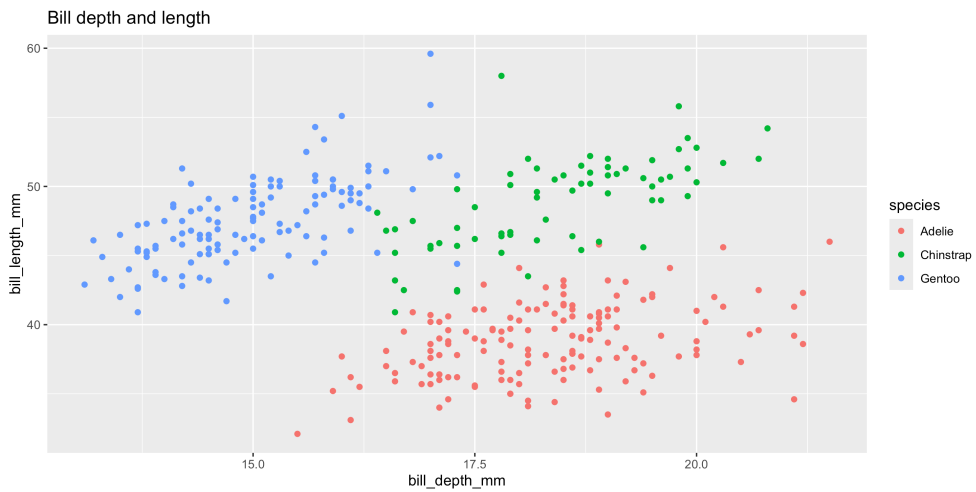
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point *and map species to the color of each point.*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  )
```



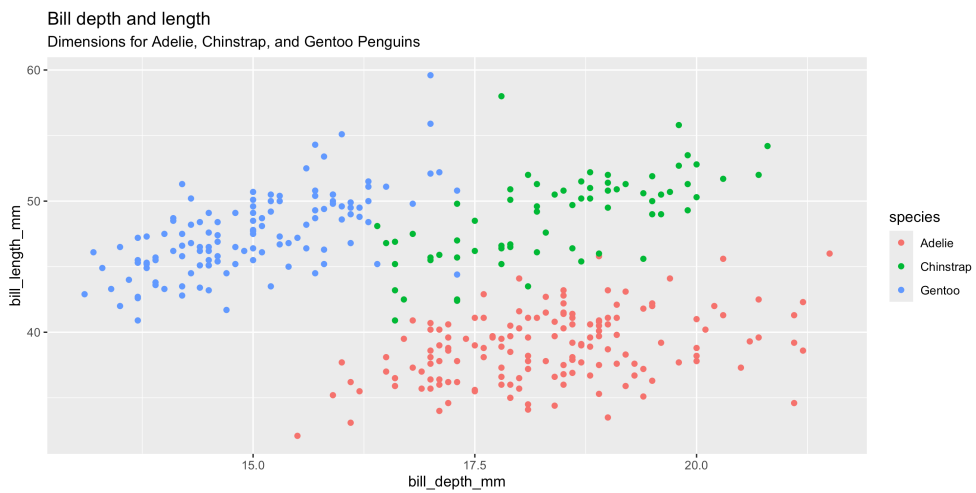
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. *Title the plot "Bill depth and length"*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(title = "Bill depth and length")
```



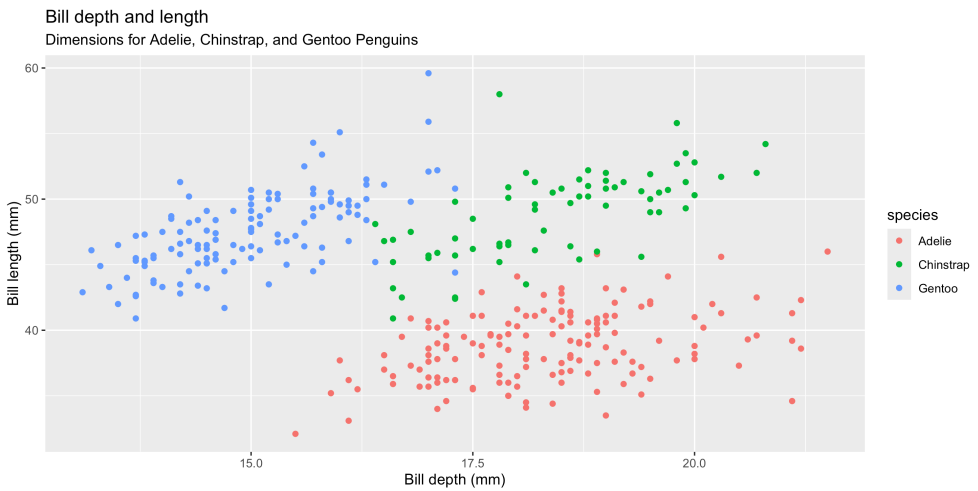
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", *add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins"*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins")
  )
```



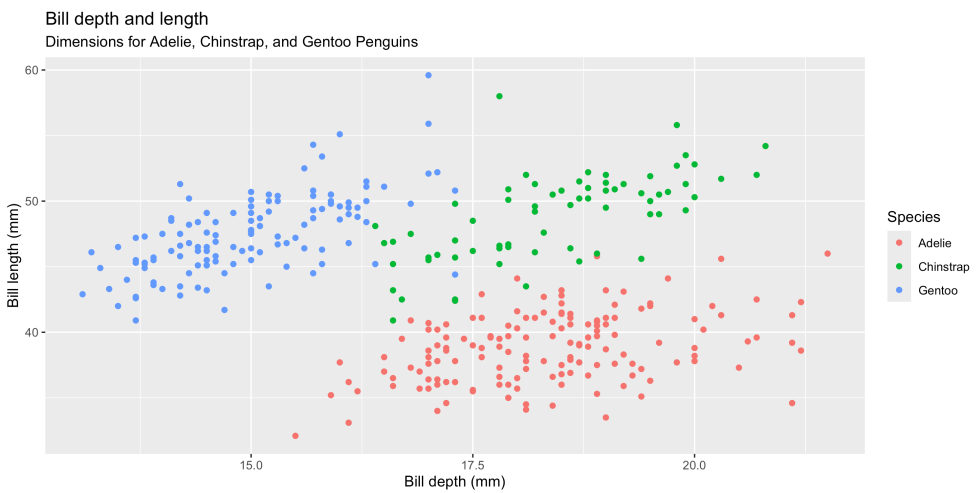
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",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

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    x = "Bill depth (mm)",
    y = "Bill length (mm)"
  )
```



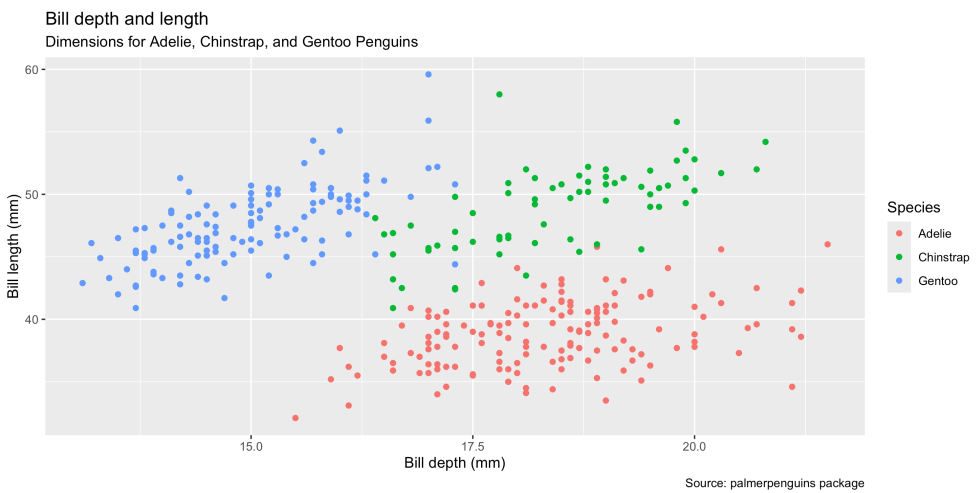
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",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

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    x = "Bill depth (mm)",
    y = "Bill length (mm)",
    color = "Species"
  )
```



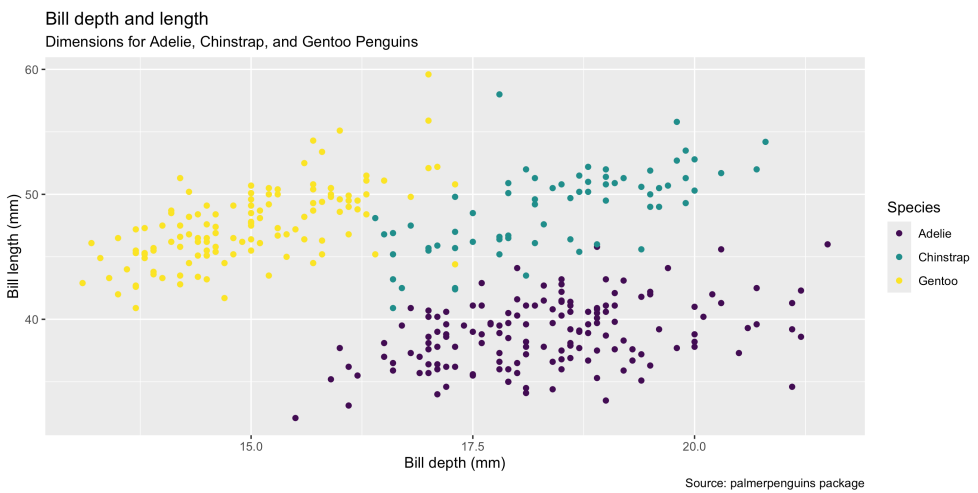
> Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",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

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    x = "Bill depth (mm)",
    y = "Bill length (mm)",
    color = "Species",
    caption = "Source: palmerpenguins packag
  )
```



> 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.*

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species)
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    x = "Bill depth (mm)",
    y = "Bill length (mm)",
    color = "Species",
    caption = "Source: palmerpenguins packa
  ) +
  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:

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

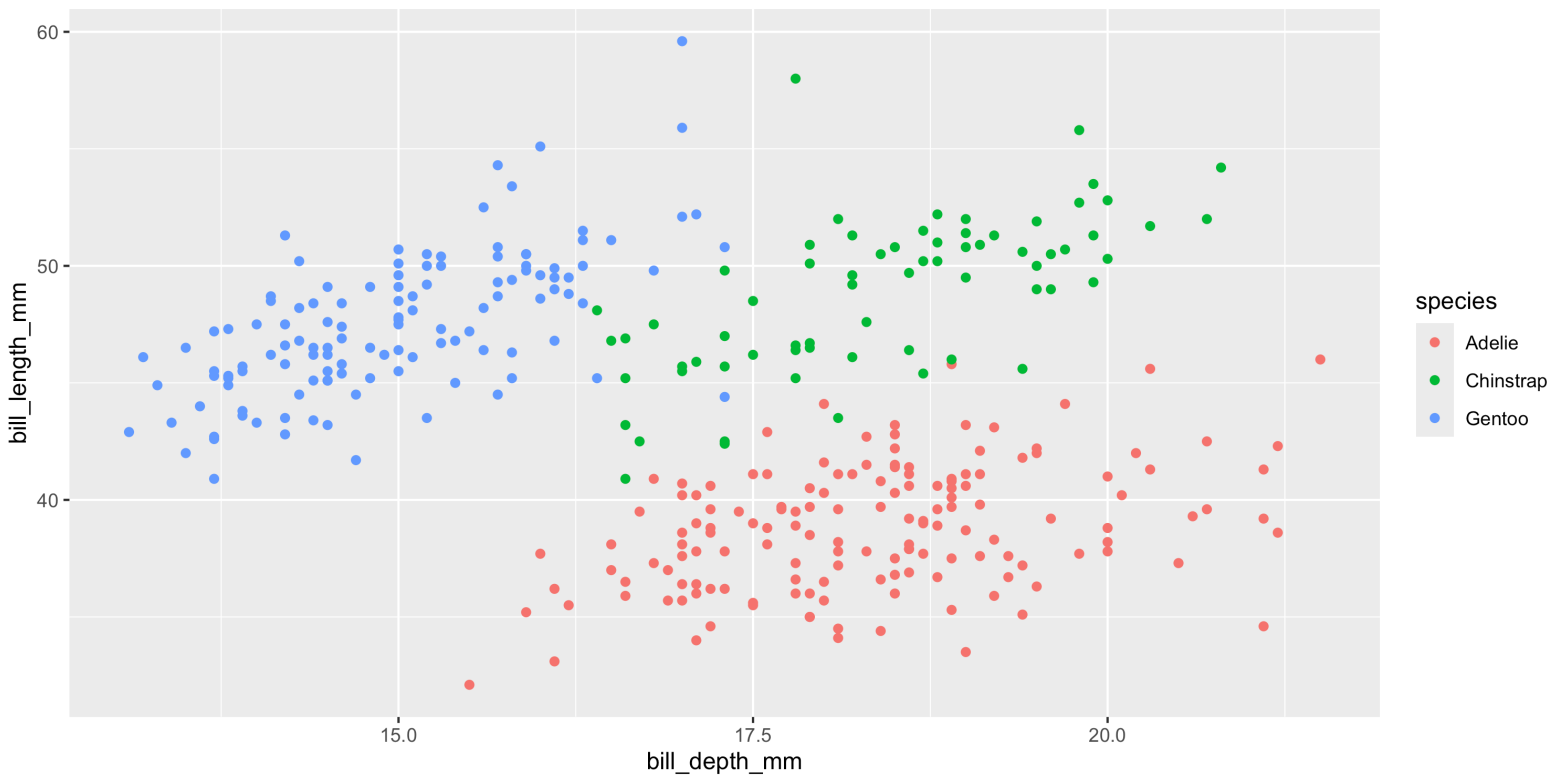
Different geometries have different aesthetics available - see the ggplot2 geoms help files for listings.

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

color

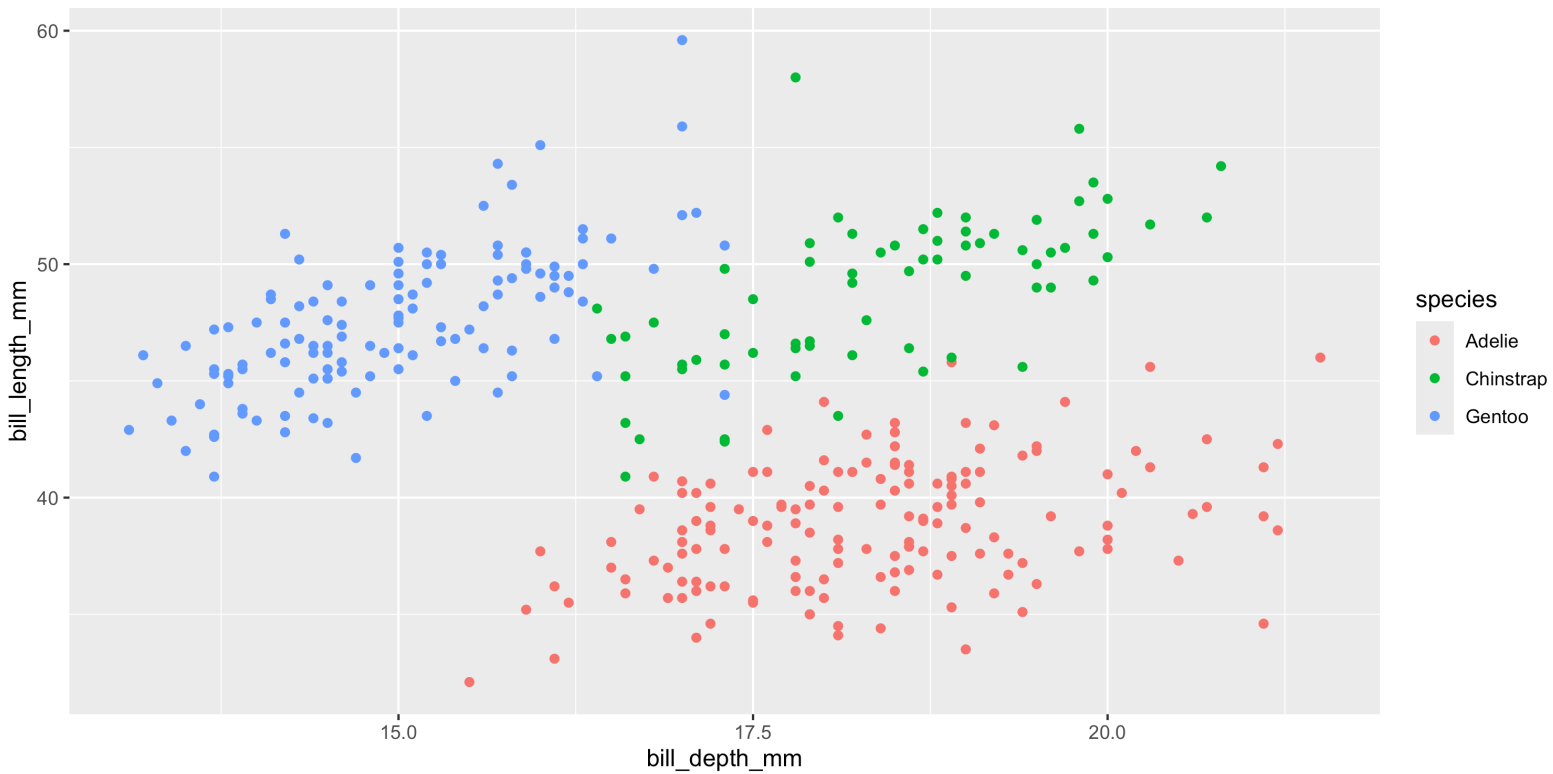
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species)
  )
```

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



Avoid the warning

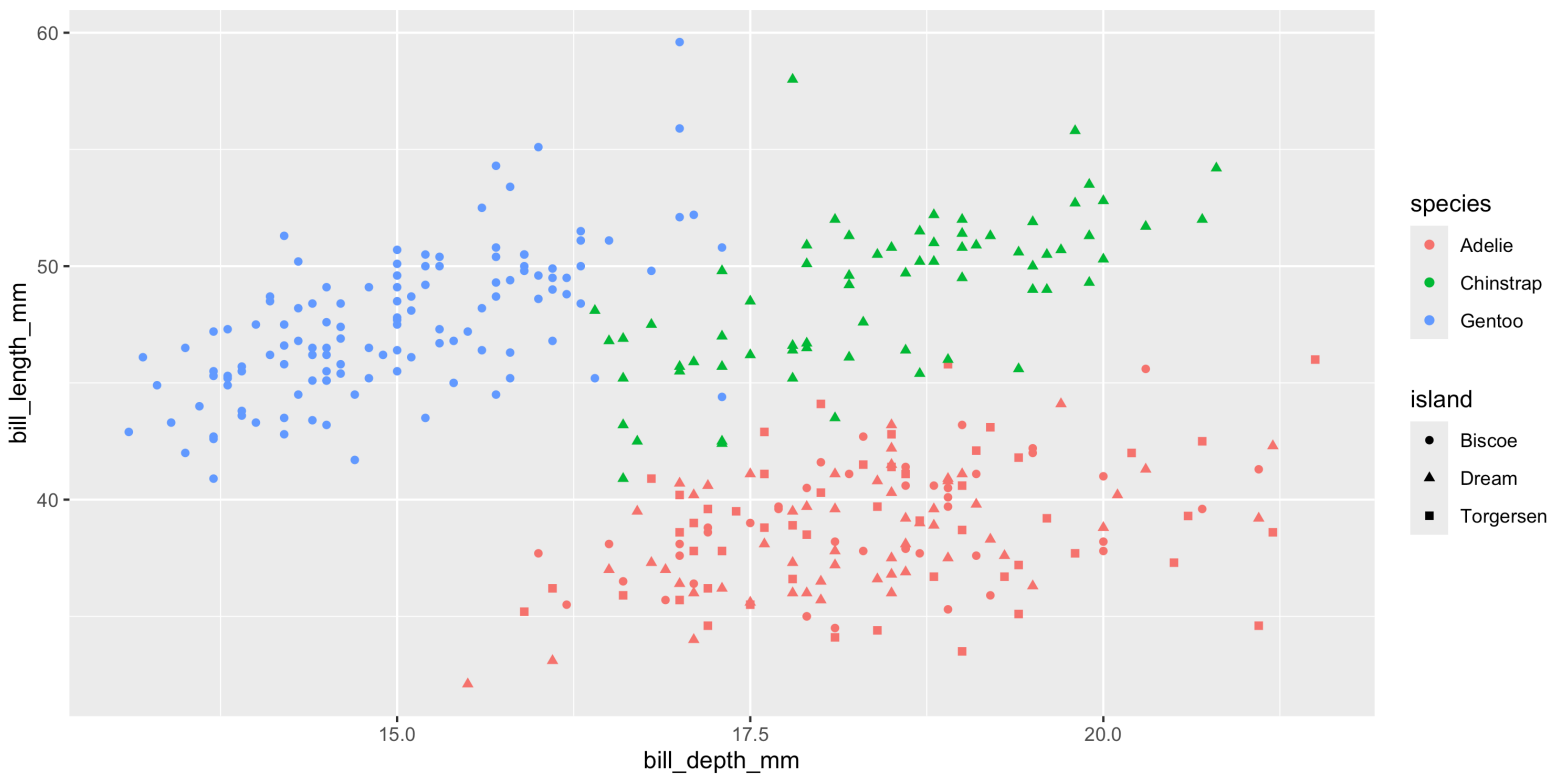
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species), na.rm=TRUE
  )
```



Shape

Mapped to a different variable than `color`

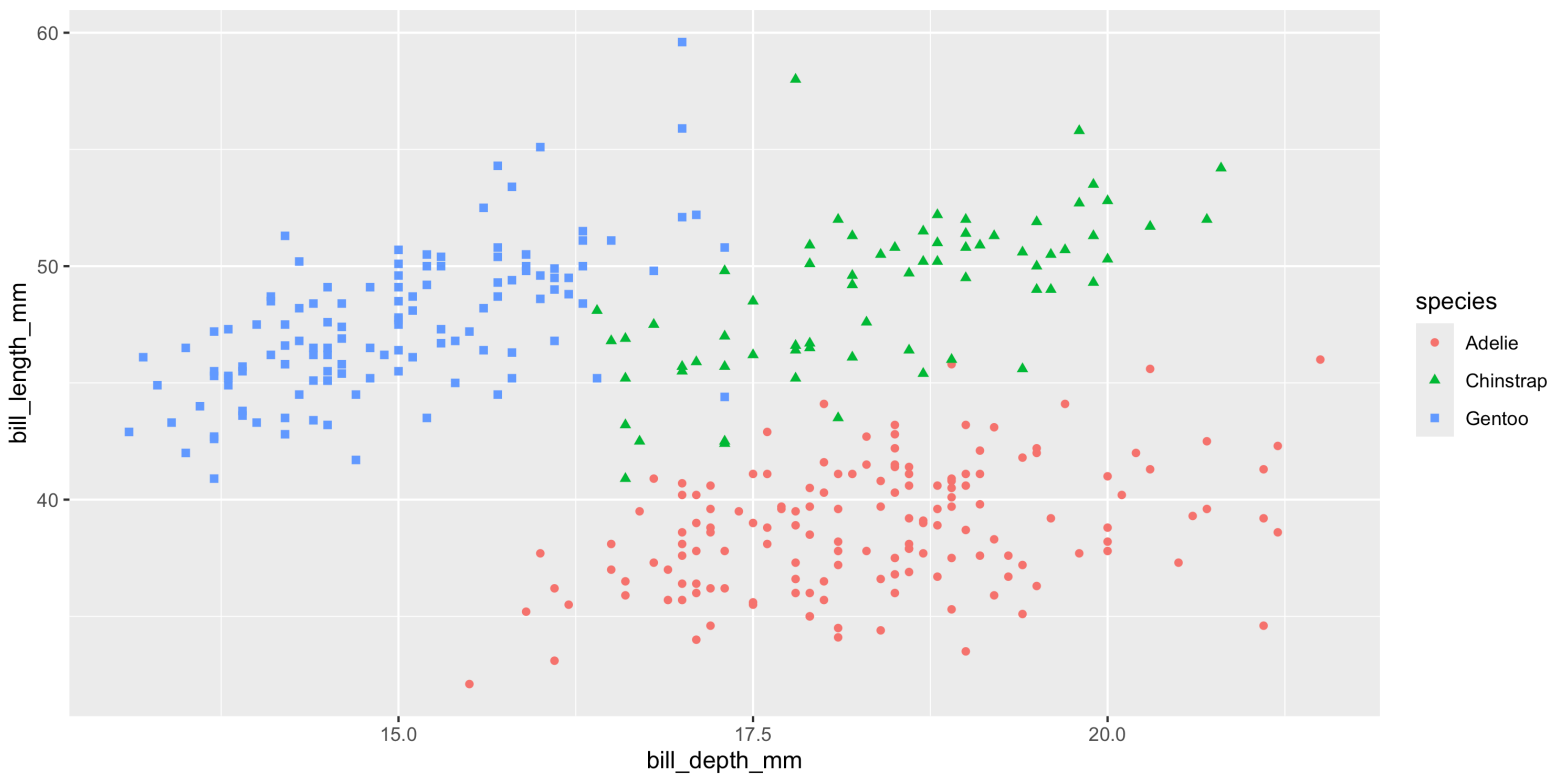
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species, shape = island), na.rm = TRUE
  )
```



Shape

Mapped to same variable as `color`

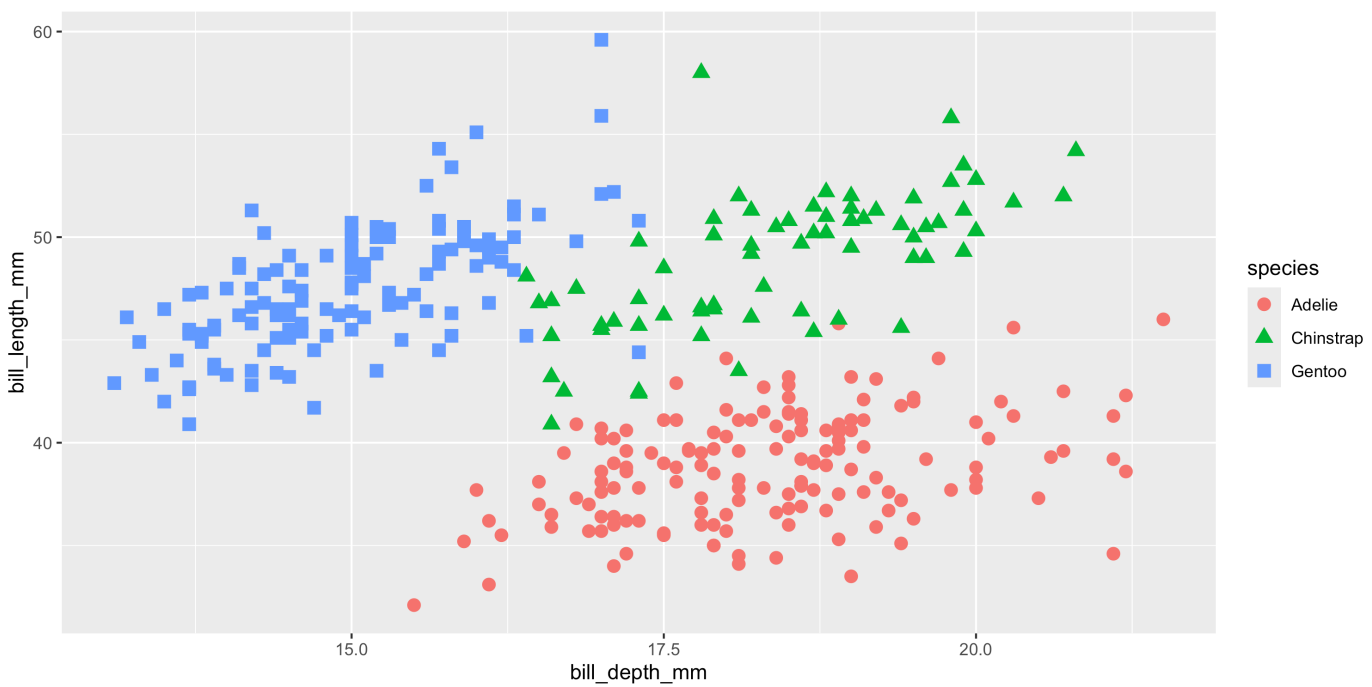
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species, shape = species), na.rm = TRUE
  )
```



Size

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

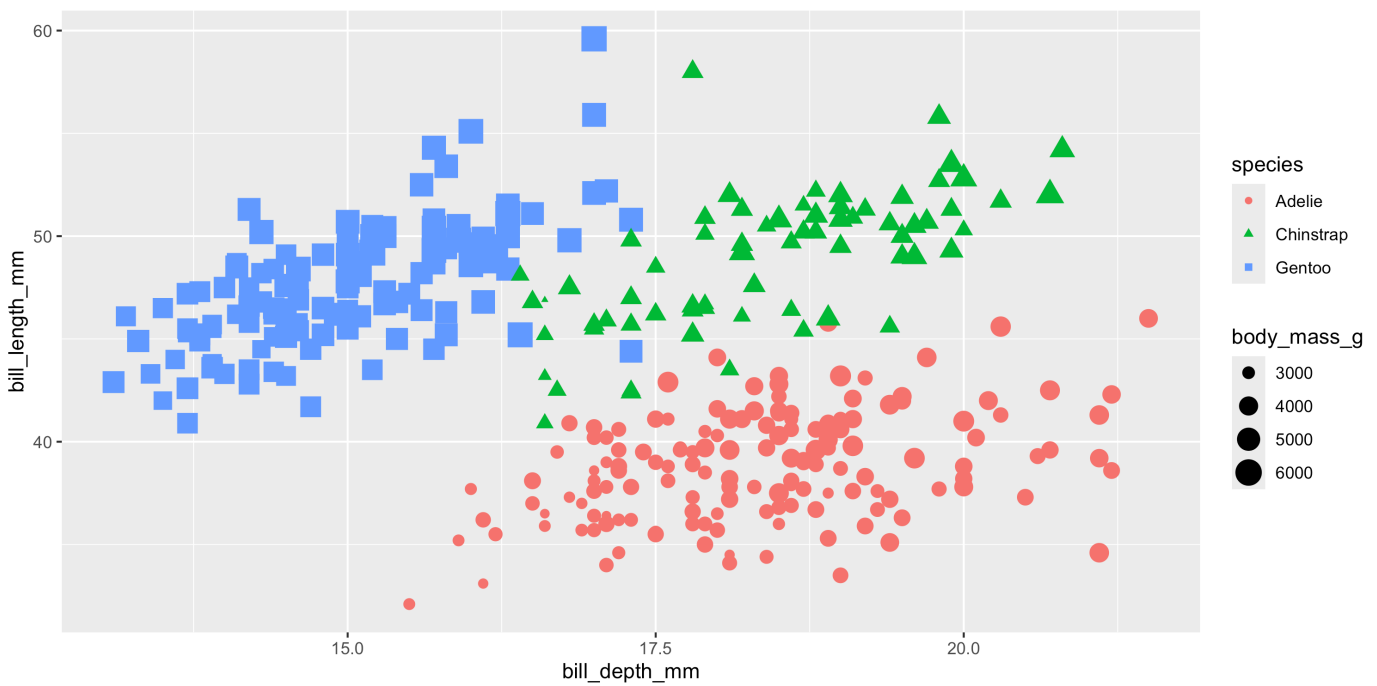
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species, shape = species), na.rm = TRUE,
    size = 3
  )
```



Size

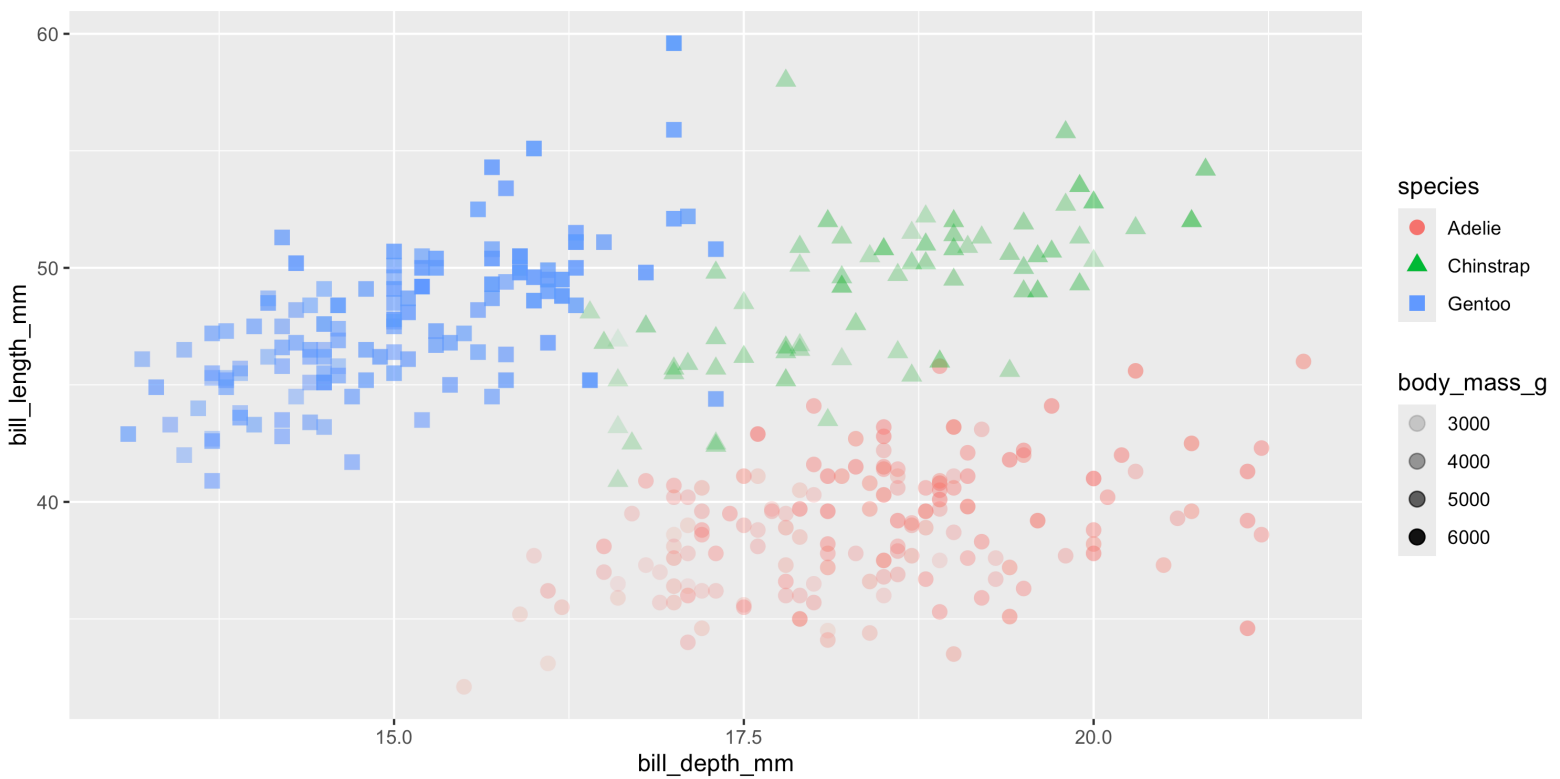
Mapped to a variable

```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(
    aes(color = species, shape = species, size = body_mass_g), na.rm = TRUE
  )
```



Alpha

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



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.

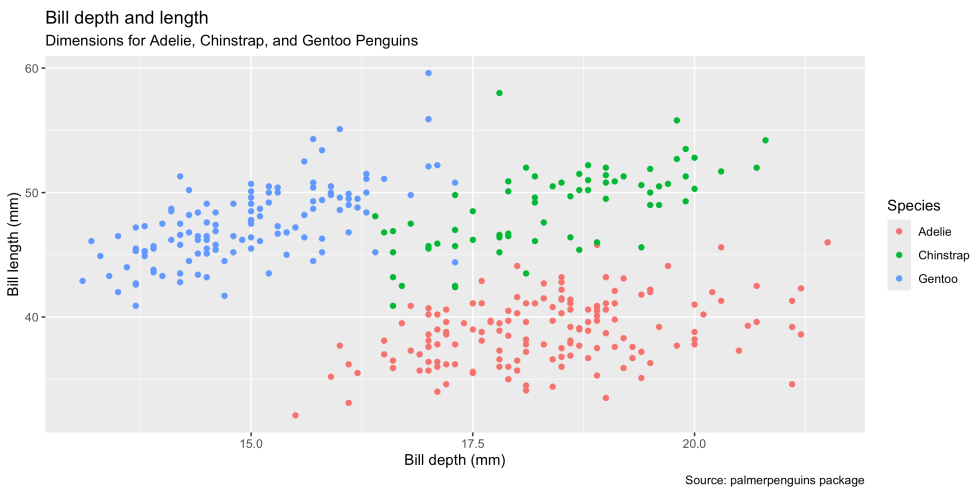
In this example `color`, `shape`, and `alpha` are all **mapping** while `size` is a **setting**.

Labels

`labs()`

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

```
ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species), na.rm = TRUE
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    x = "Bill depth (mm)",
    y = "Bill length (mm)",
    color = "Species",
    caption = "Source: palmerpenguins package"
  )
```

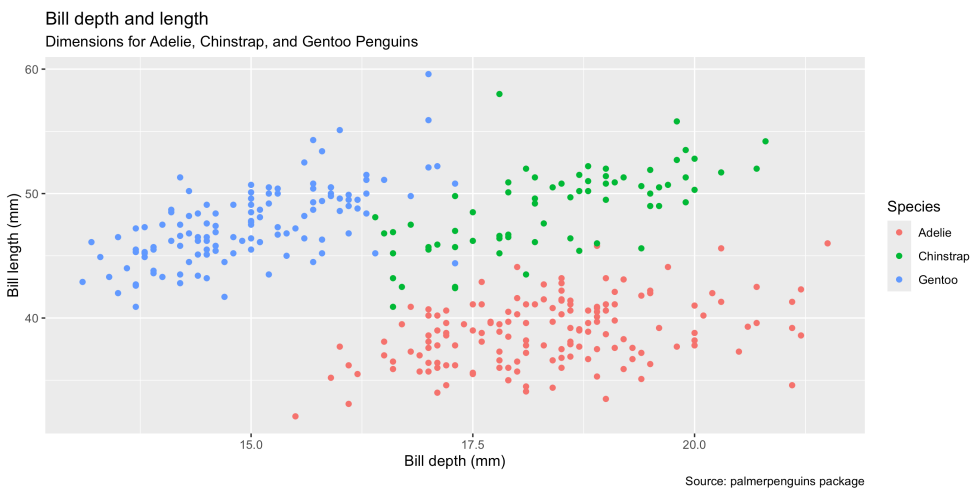


Labels

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

```
p_labeled = penguins
attr(p_labeled$species, "label") = "Species"
attr(p_labeled$bill_depth_mm, "label") = "Bill depth (mm)"
attr(p_labeled$bill_length_mm, "label") = "Bill length (mm

ggplot(
  data = p_labeled,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species), na.rm = TRUE
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    caption = "Source: palmerpenguins package"
  )
```

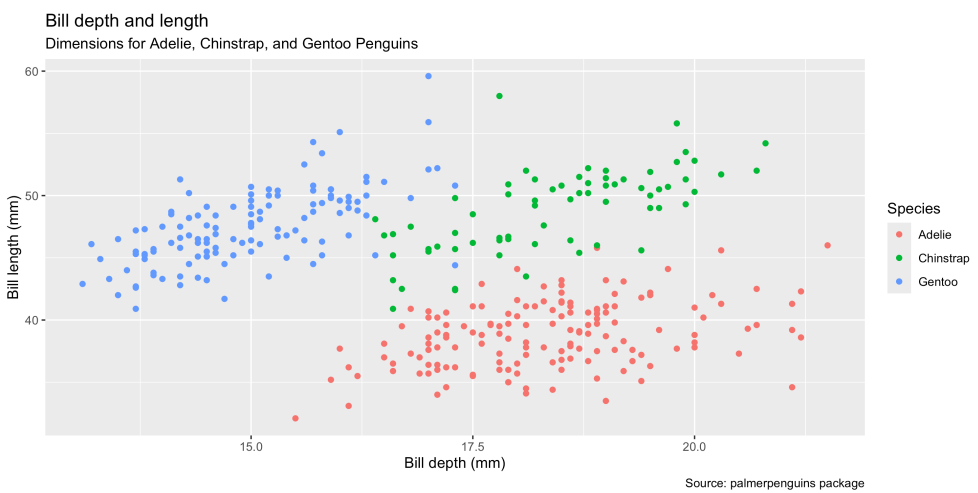


Dictionary

Alternatively, we can provide a dictionary / lookup table via the `dictionary` argument of `labs()`.

```
lookup = c(
  species = "Species",
  bill_depth_mm = "Bill depth (mm)",
  bill_length_mm = "Bill length (mm)"
)

ggplot(
  data = penguins,
  mapping = aes(
    x = bill_depth_mm,
    y = bill_length_mm
  )
) +
  geom_point(
    mapping = aes(color = species), na.rm = TRUE
  ) +
  labs(
    title = "Bill depth and length",
    subtitle = paste(
      "Dimensions for Adelie,",
      "Chinstrap, and Gentoo",
      "Penguins"),
    caption = "Source: palmerpenguins package",
    dictionary = lookup
  )
```

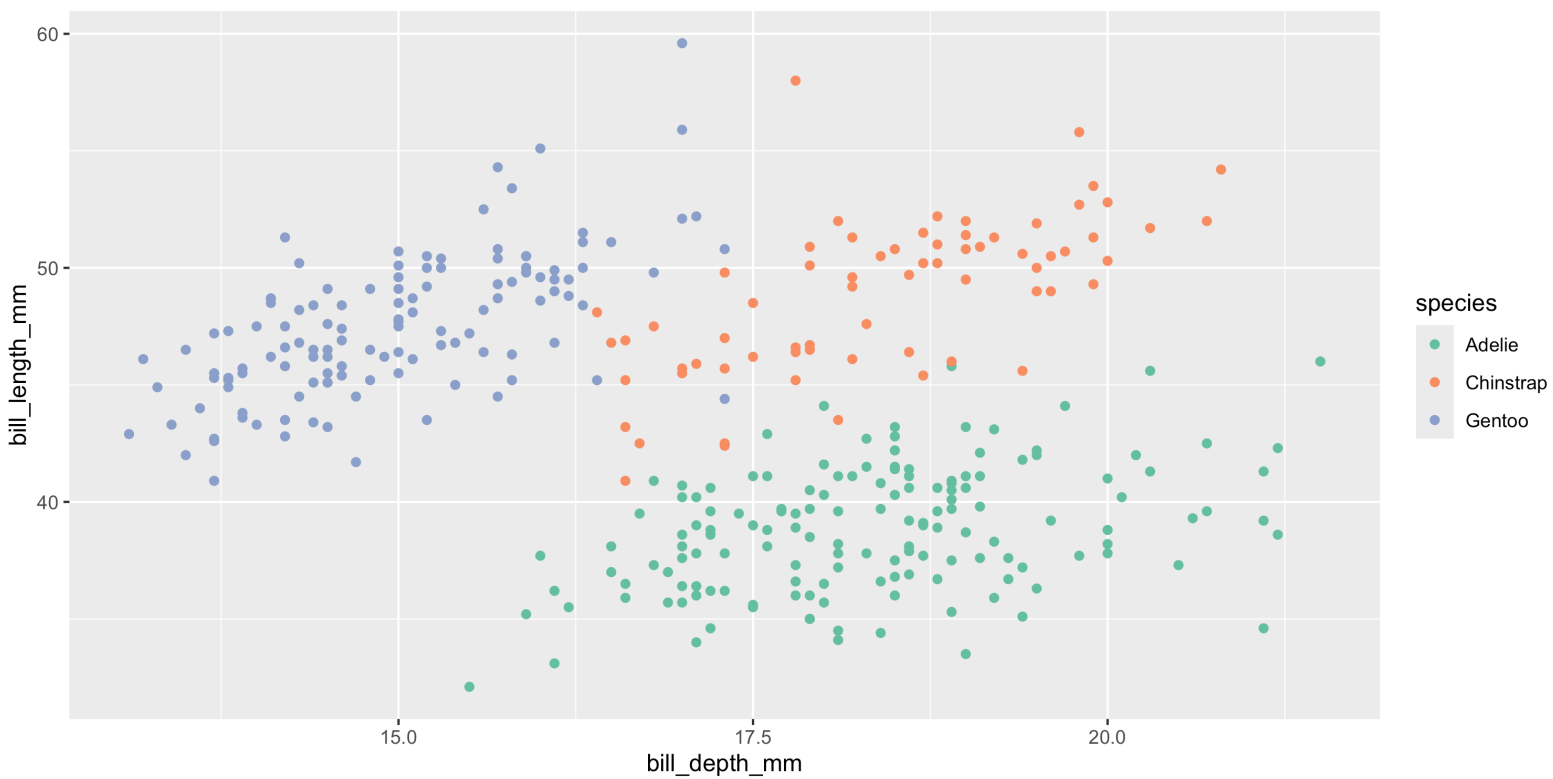


Scales

Scales

Scales control the mapping from data values to aesthetic values — they determine *how* a variable is translated into `color`, `size`, position, etc. Every aesthetic has a default scale, but we can override it:

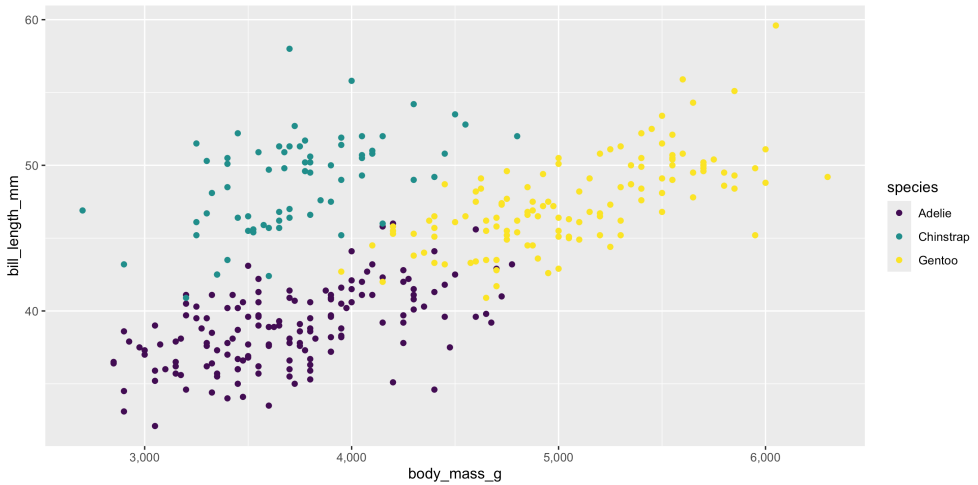
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(aes(color = species), na.rm = TRUE) +
  scale_color_brewer(type = "qual", palette = "Set2")
```



Axis scales

Scales also apply to positional aesthetics (x and y) — useful for transforming axes or customizing breaks and labels.

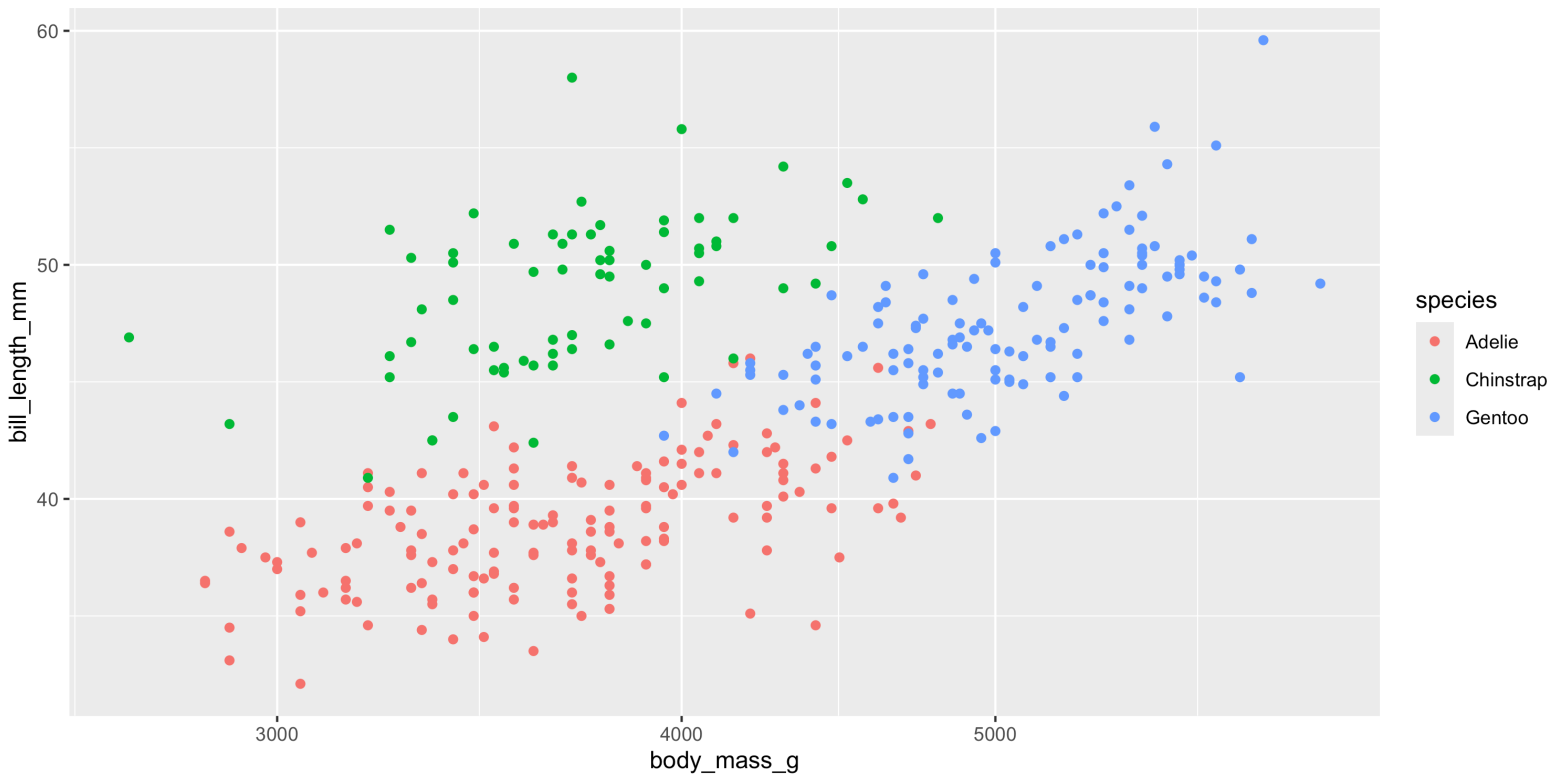
```
ggplot(
  penguins,
  aes(
    x = body_mass_g,
    y = bill_length_mm
  )
) +
  geom_point(
    aes(color = species), na.rm = TRUE
  ) +
  scale_x_continuous(
    labels = scales::label_comma(),
    breaks = seq(3000, 6000, by = 1000)
  ) +
  scale_color_viridis_d()
```



The scales package provides a variety of useful helper functions for formatting axis labels, e.g. `label_comma()`,

Log scales

```
ggplot(
  penguins, aes(x = body_mass_g, y = bill_length_mm)
) +
  geom_point(aes(color = species), na.rm = TRUE) +
  scale_x_log10()
```



This can also be achieved via `scale_x_continuous(transform = "log10")` which supports many additional

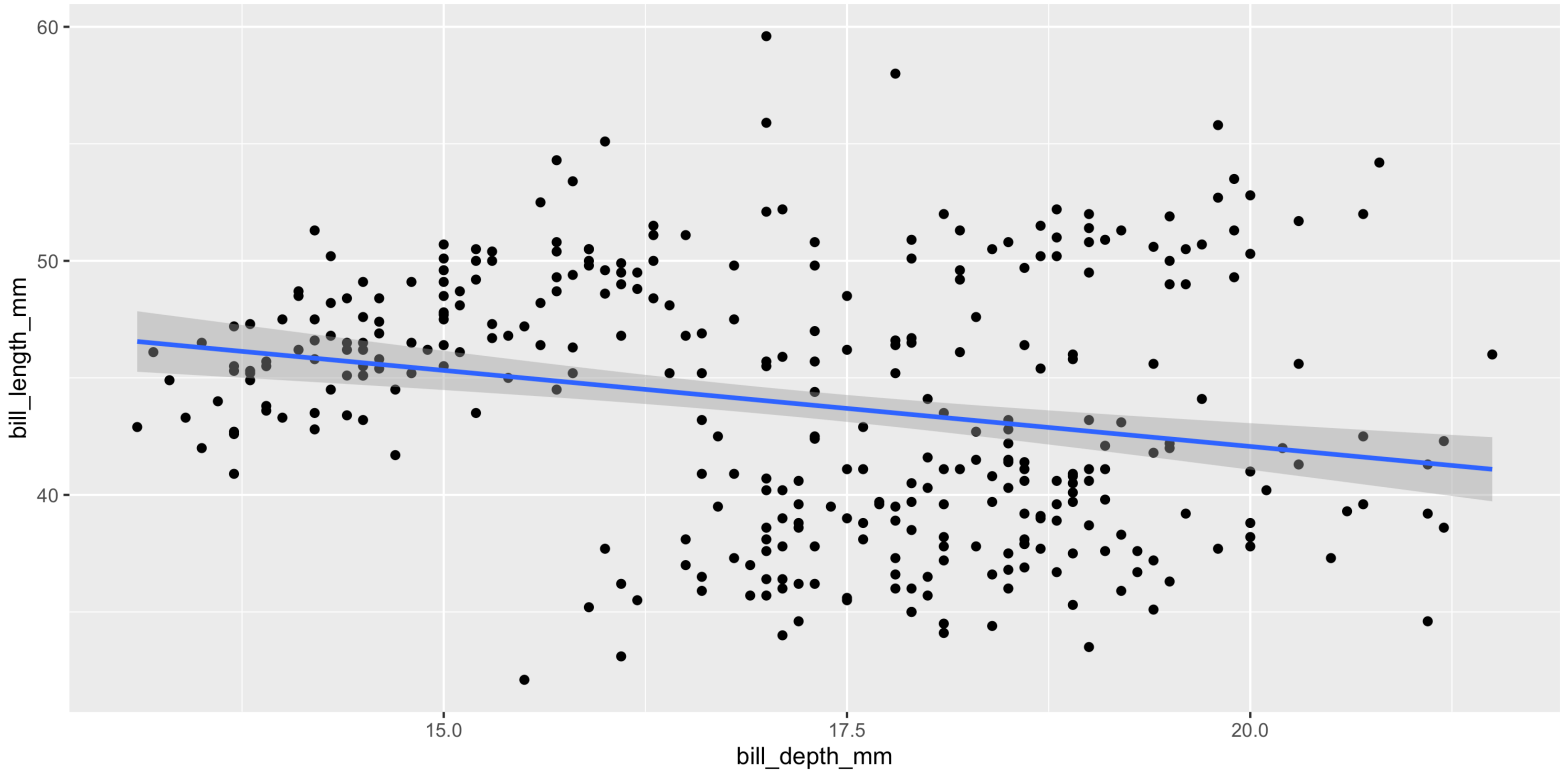
Statistical Transformations

Statistical Transformations

Many geoms apply a statistical transformation to the data before plotting — e.g. binning, counting, or fitting a model.

```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE) +
  geom_smooth(method = "lm", na.rm = TRUE)
```

```
`geom_smooth()` using formula = 'y ~ x'
```

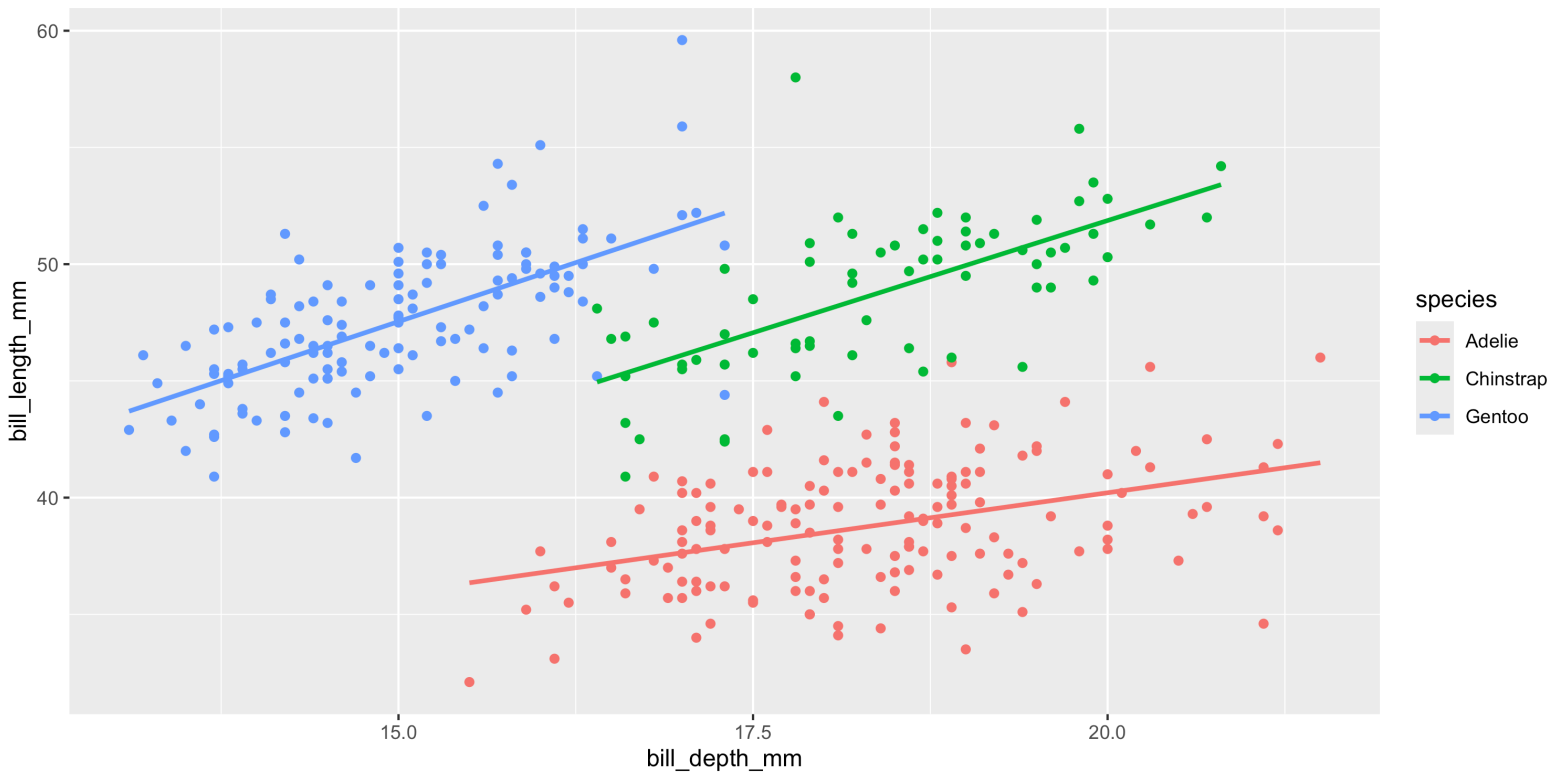

60
50
40
bill_length_mm
15.0
17.5
20.0
bill_depth_mm

Stat transformations + aesthetics

Statistical transformations respect aesthetic groupings.

```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)
) +
  geom_point(na.rm = TRUE) +
  geom_smooth(method = "lm", na.rm = TRUE, se = FALSE, )
```

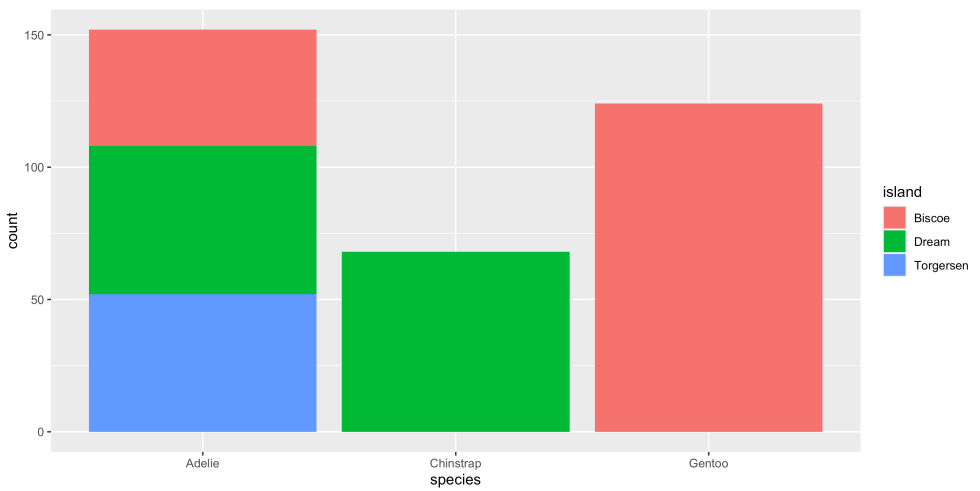
```
`geom_smooth()` using formula = 'y ~ x'
```



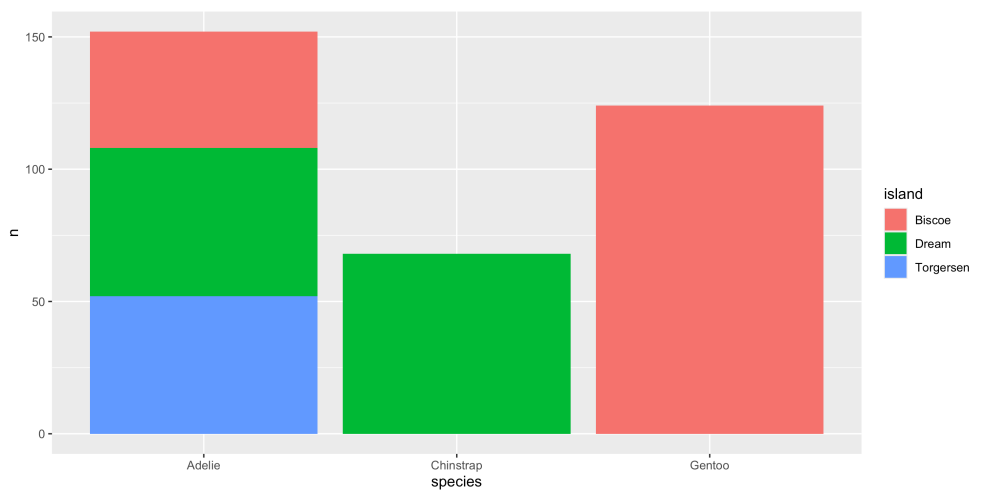
Implicit stat transformations

Some geoms imply a stat — `geom_bar()` uses `stat = "count"` by default, so it only needs an `x` aesthetic.

```
ggplot(
  penguins, aes(x = species, fill = island)
) +
  geom_bar()
```




```
penguins |>
  count(species, island) |>
  ggplot(
    aes(x = species, y = n, fill = island)
  ) +
    geom_bar(stat = "identity")
```



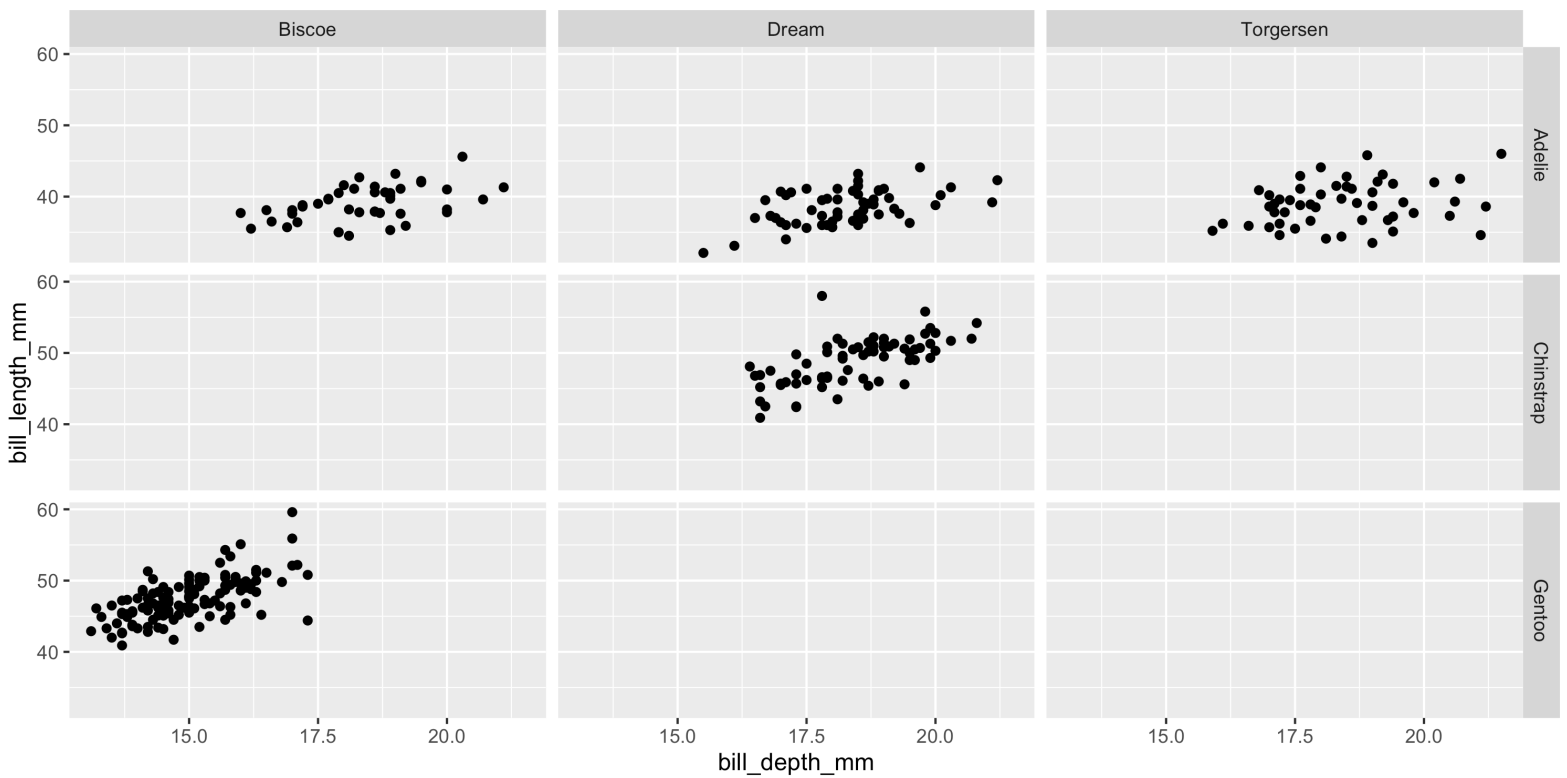
Faceting

Faceting

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

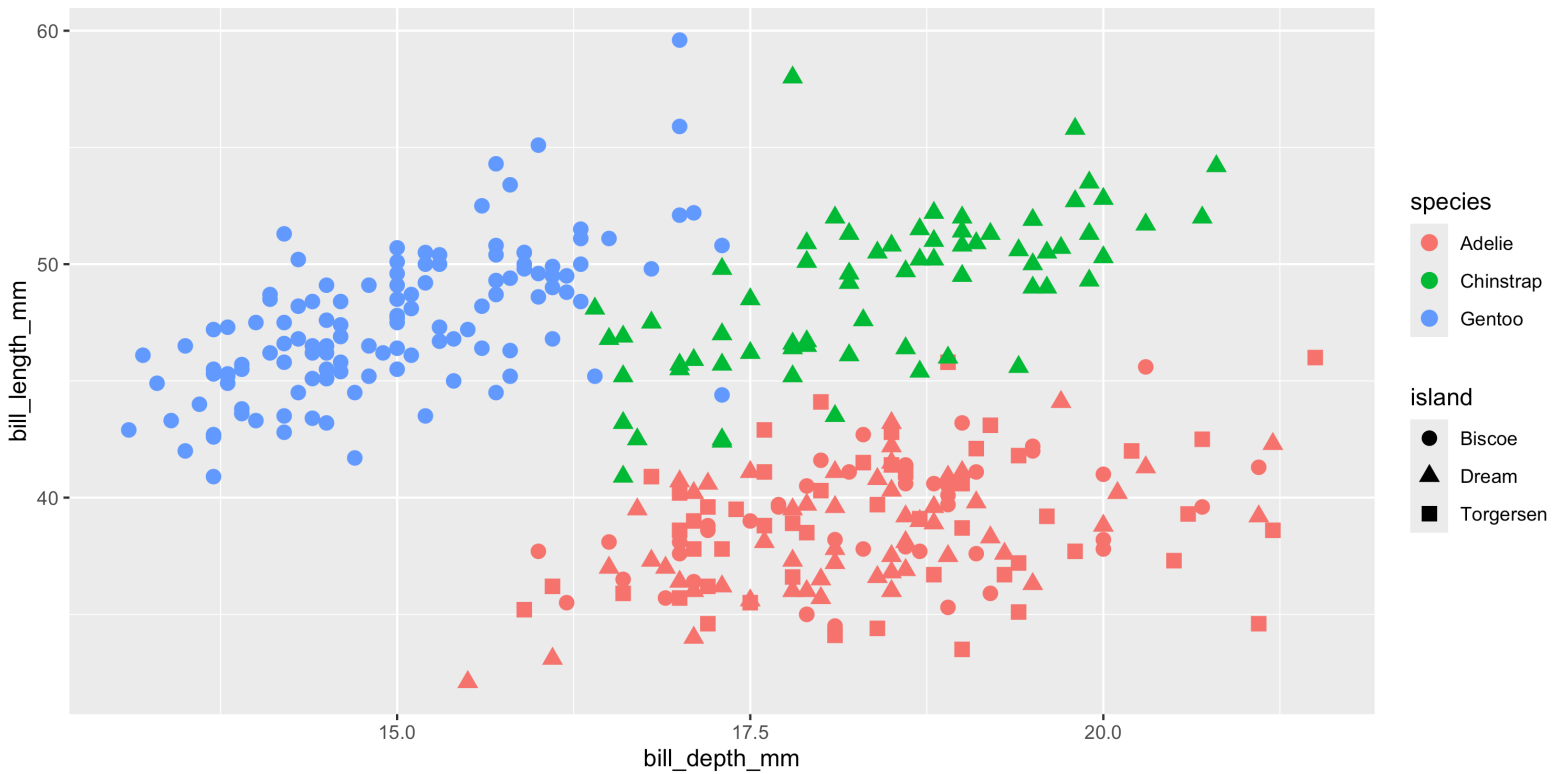
facet_grid

```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE) +
  facet_grid(
    species ~ island
  )
```



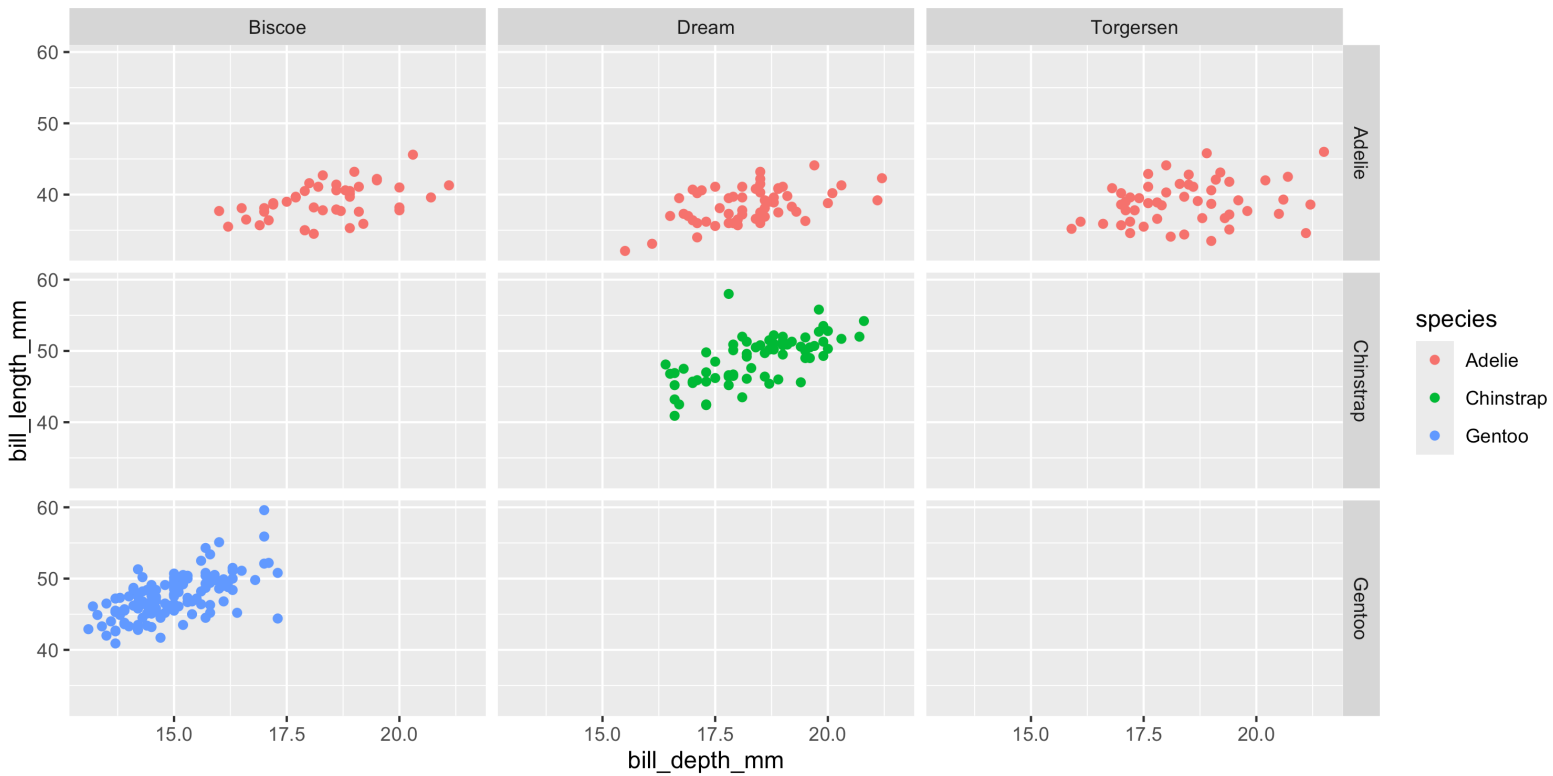
Compare with ...

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



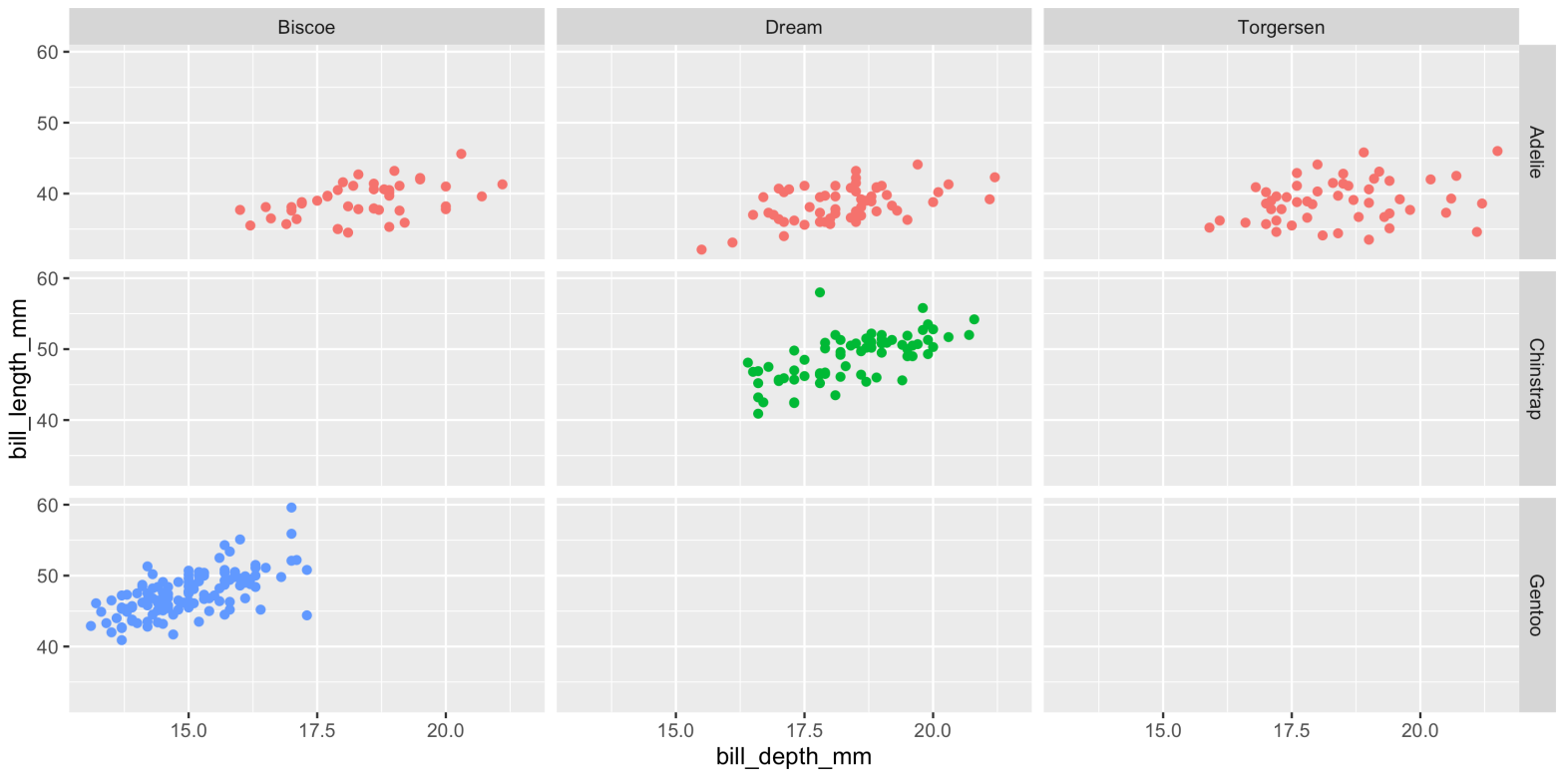
Faceting and color

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



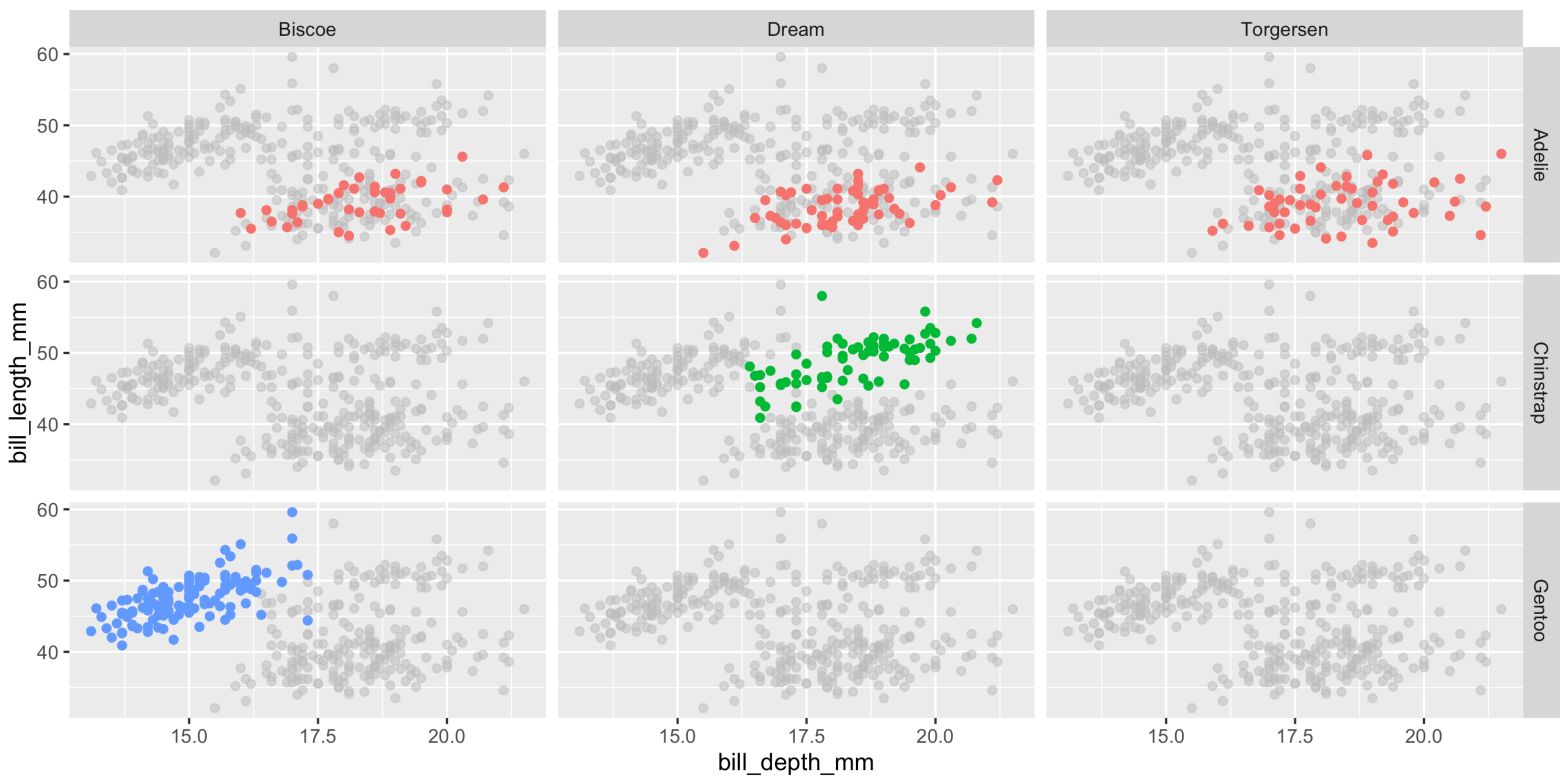
Hiding legend elements

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



Facet layout - context

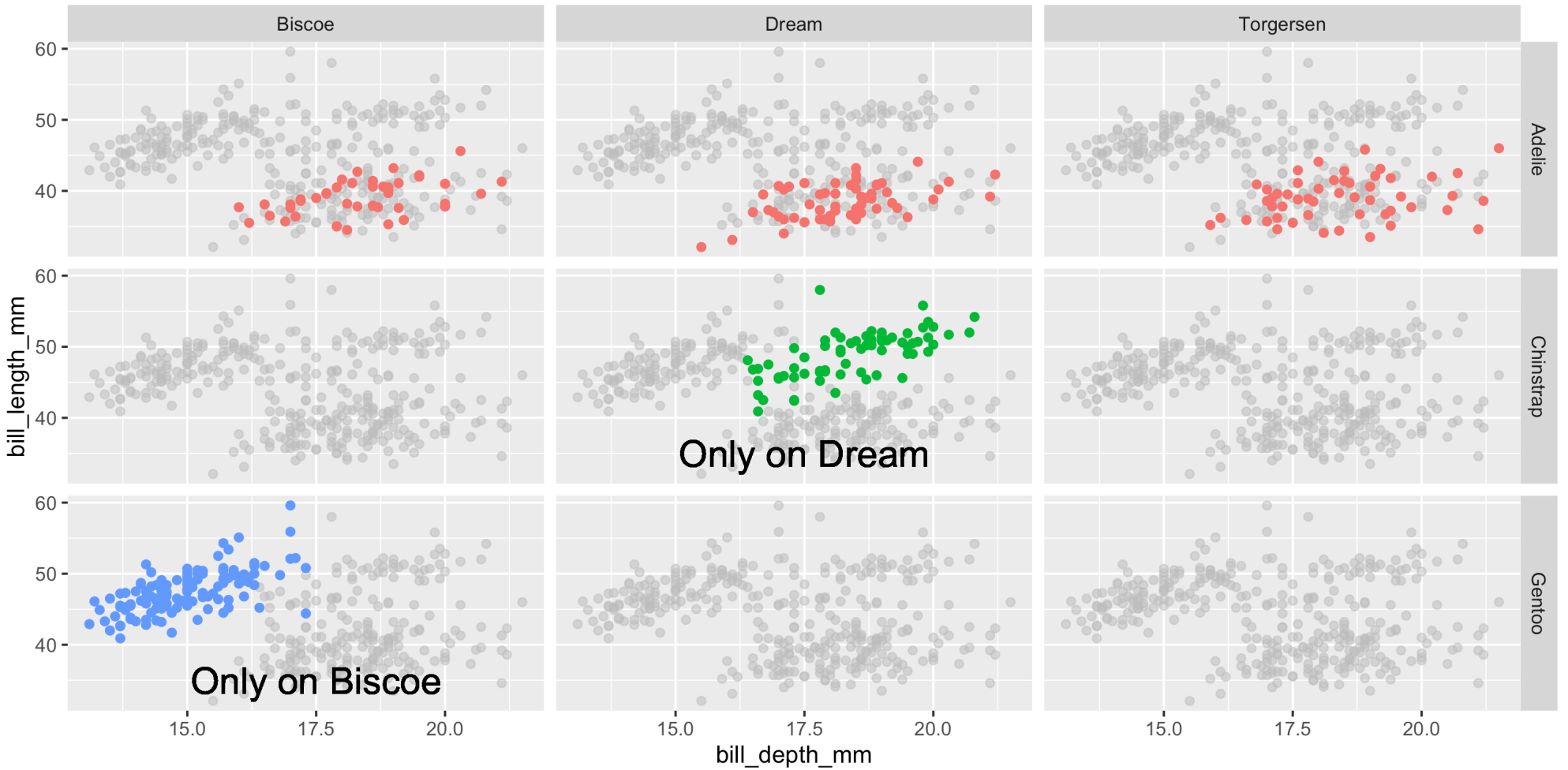
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)
) +
  geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +
  geom_point(na.rm = TRUE) +
  facet_grid(species ~ island) +
  guides(color = "none")
```



Facet layout - annotation

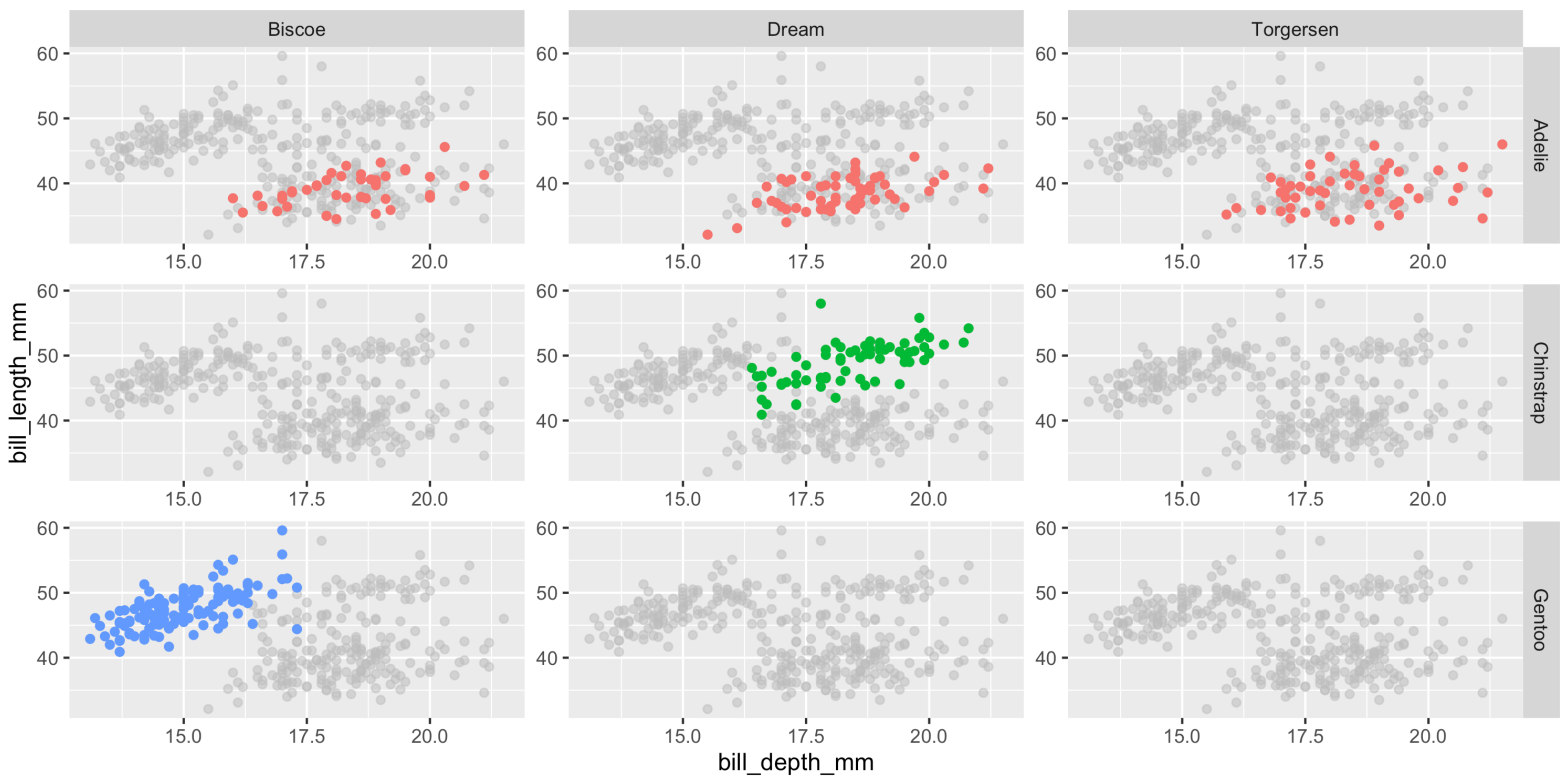
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)
) +
  geom_point(color = "grey", alpha = 0.5, na.rm = TRUE, layout = "fixed") +
  geom_point(na.rm = TRUE) +
  facet_grid(species ~ island) +
  guides(color = "none") +
  geom_text(
    x = 17.5, y = 35, label = "Only on Dream", size = 6, color = "black",
    layout = 5
  ) +
  geom_text(
    x = 17.5, y = 35, label = "Only on Biscoe", size = 6, color = "black",
    layout = 7
  )
```

Facet layout - annotation



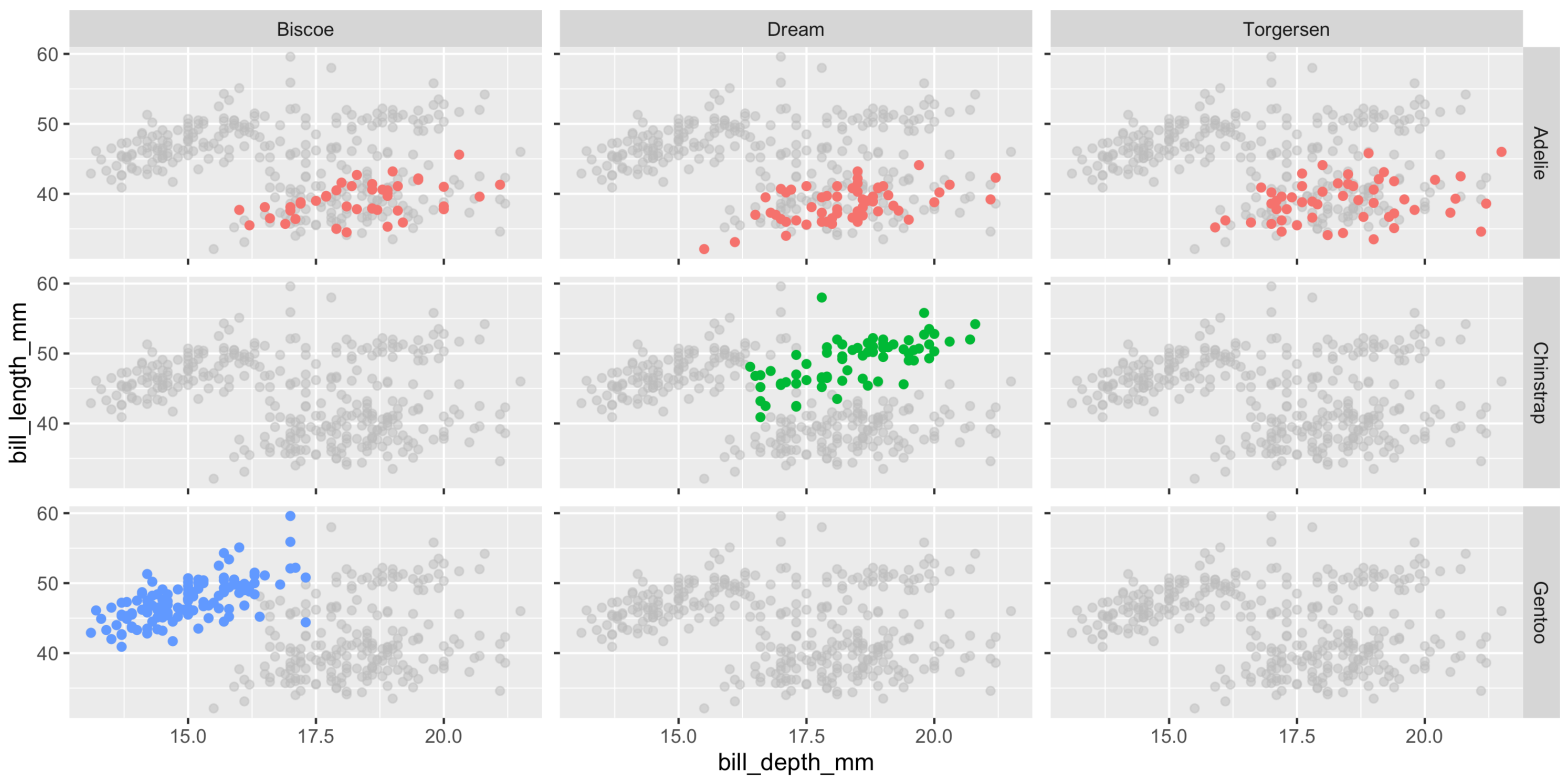
Facet axes

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



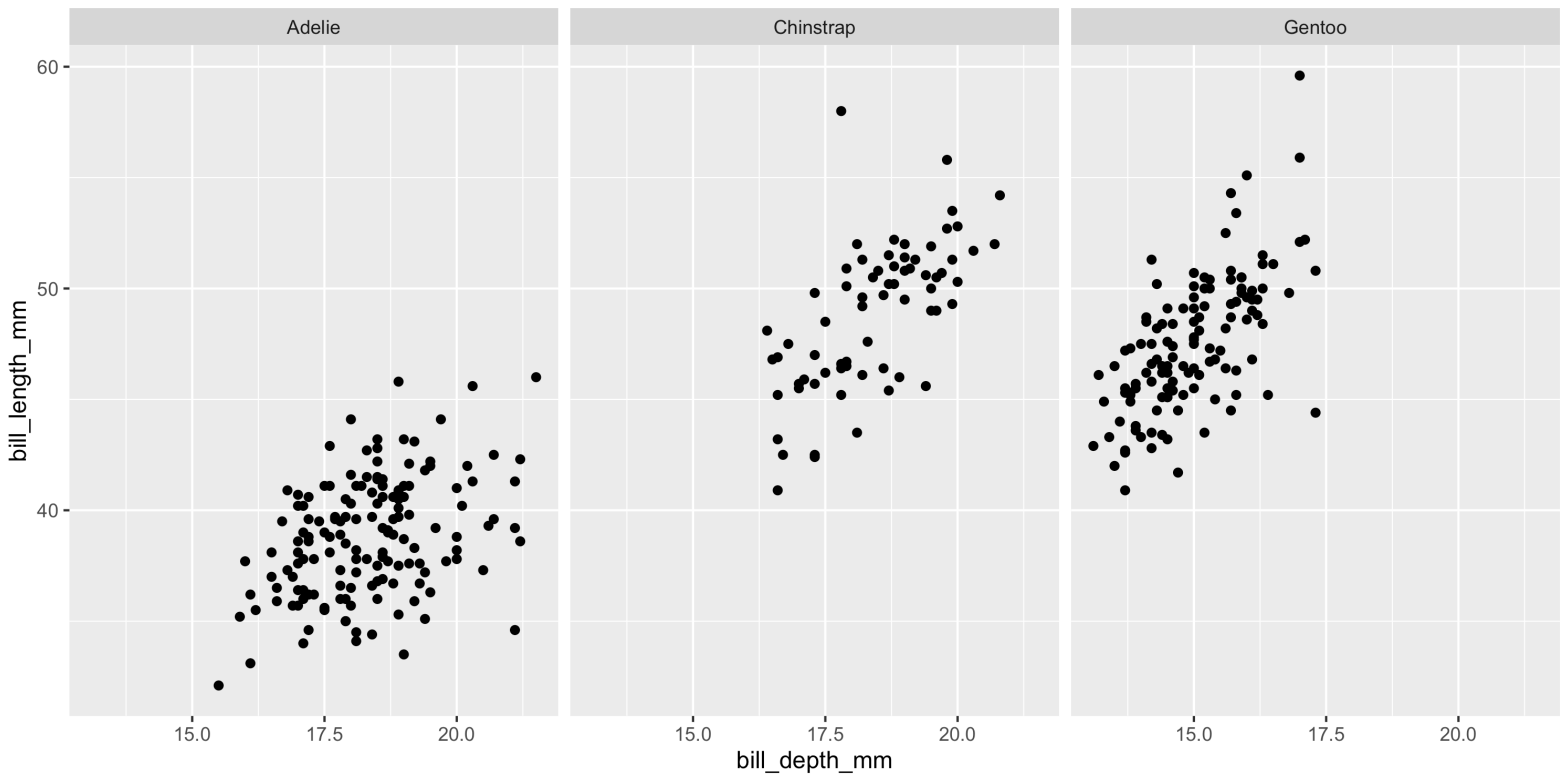
Facet axes - labels

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



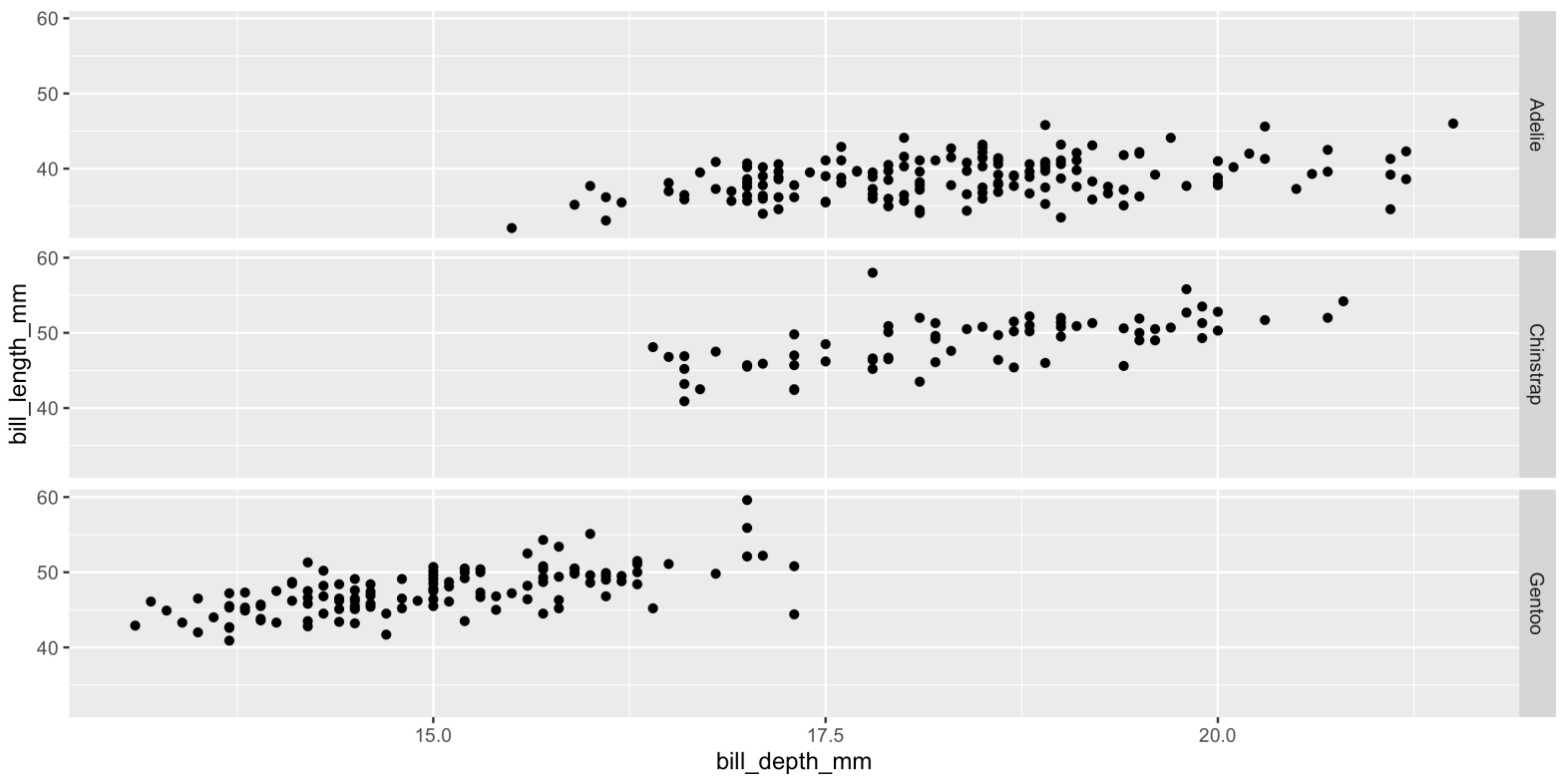
facet_grid (columns)

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



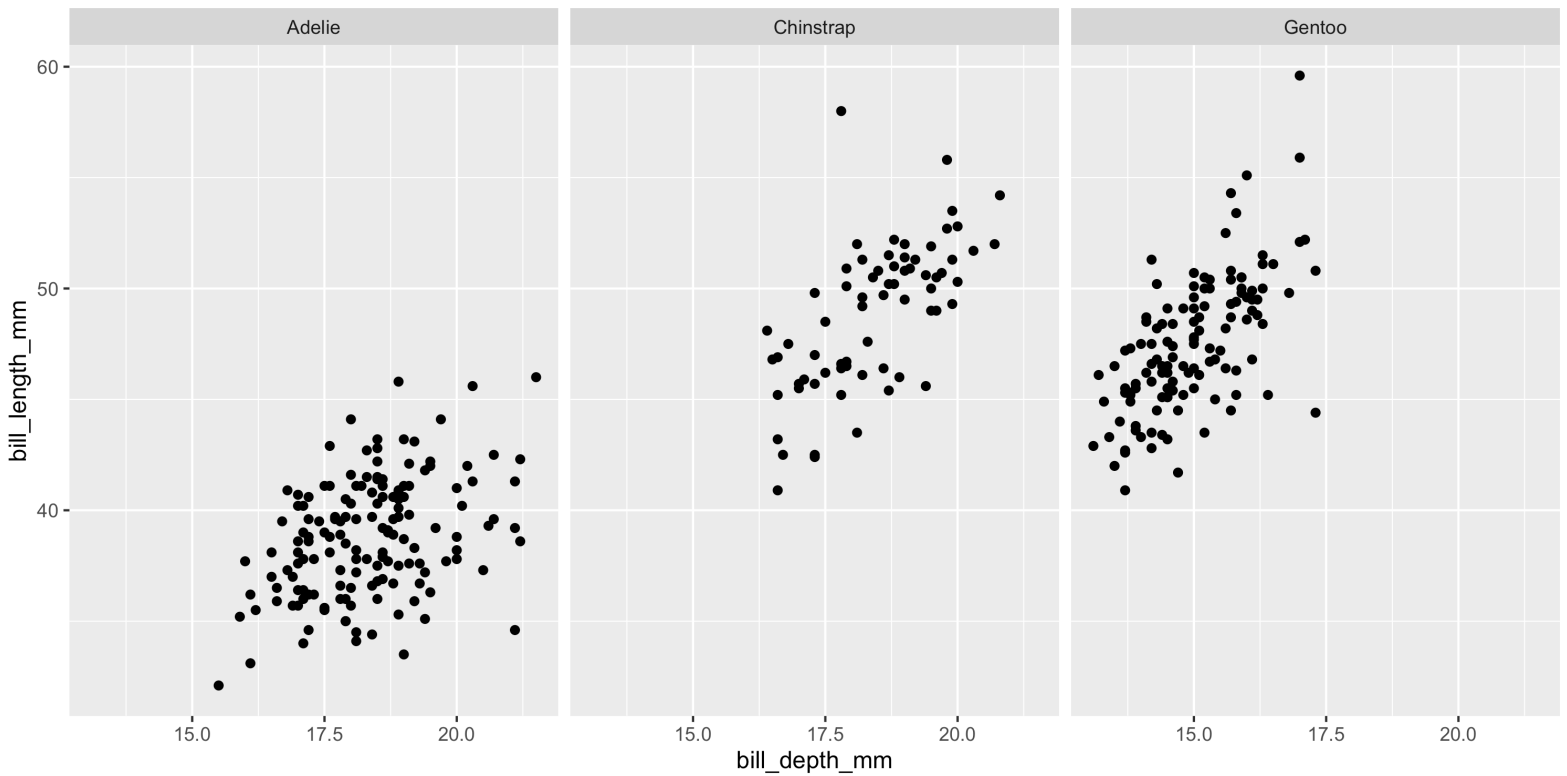
facet_grid (rows)

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



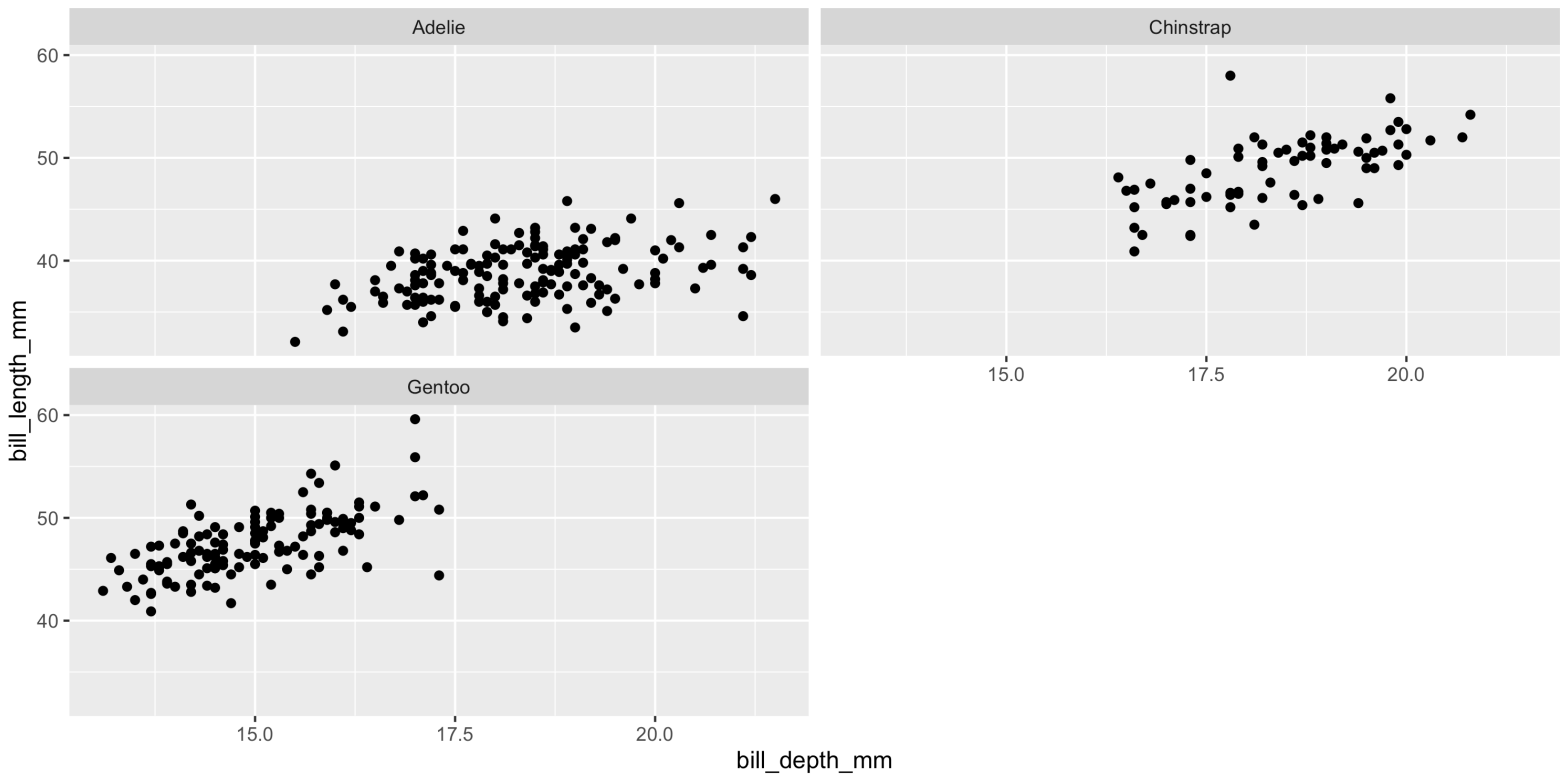
facet_wrap

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



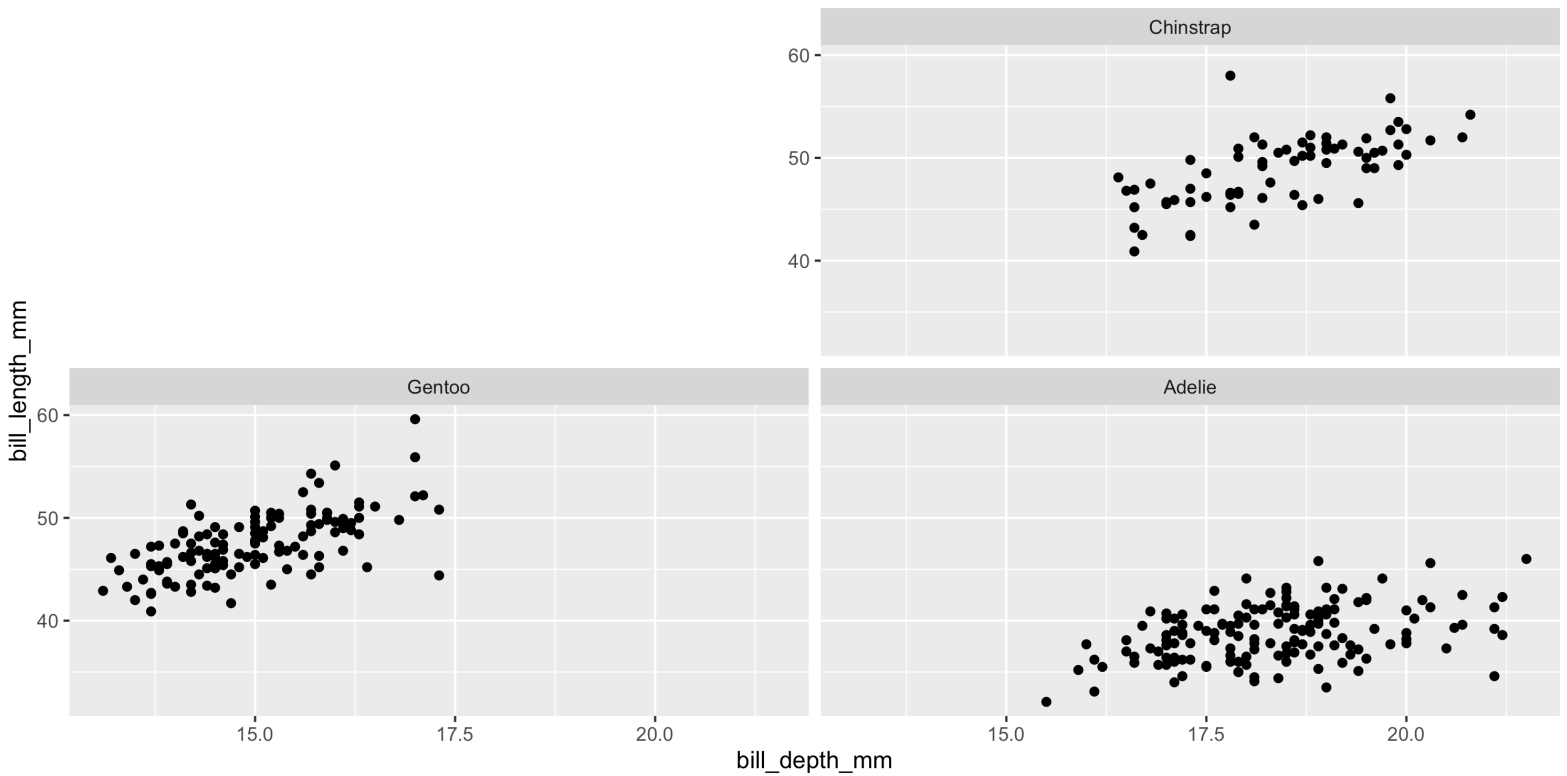
facet_wrap

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



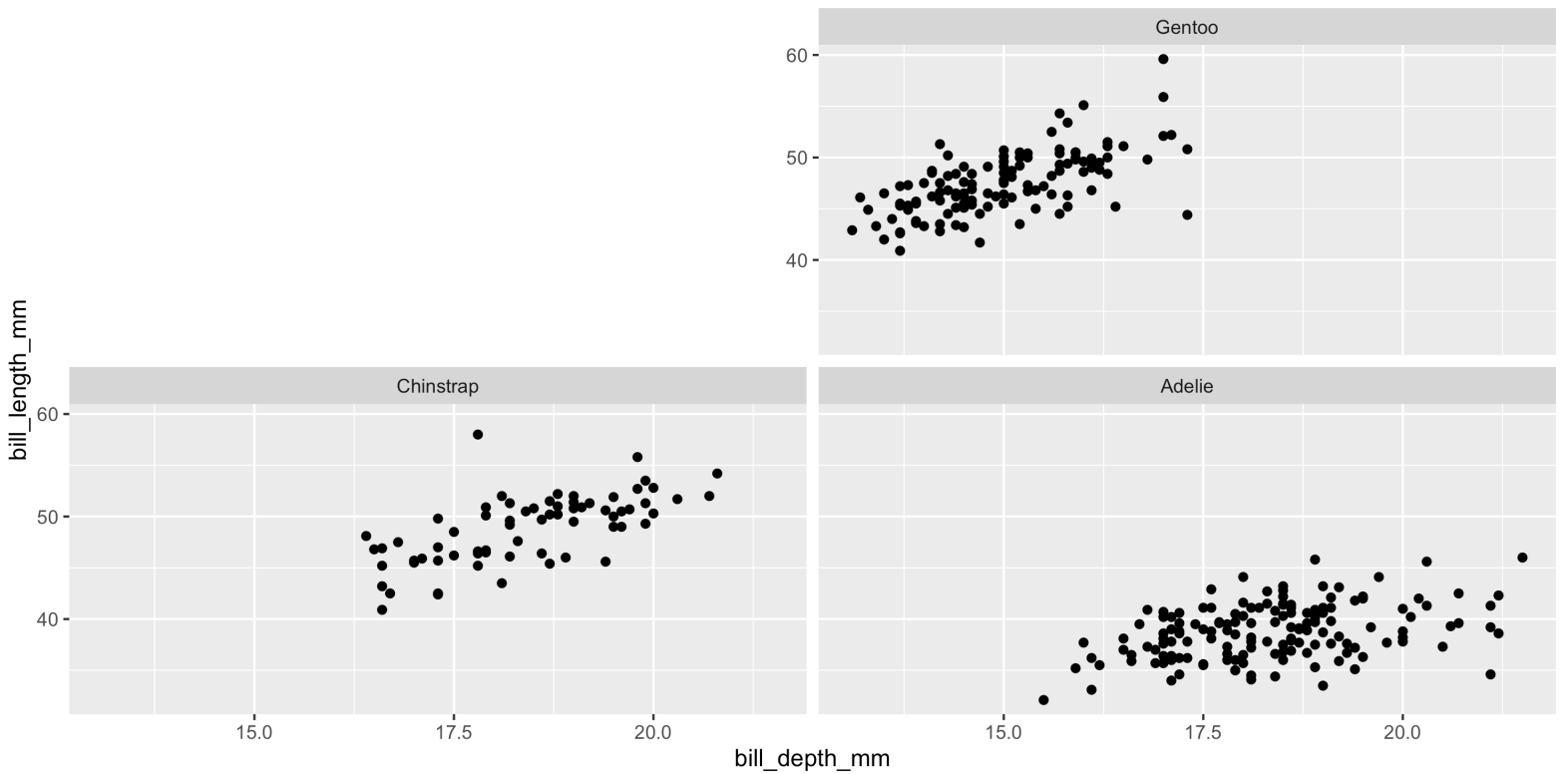
facet_wrap - direction

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



facet_wrap - direction

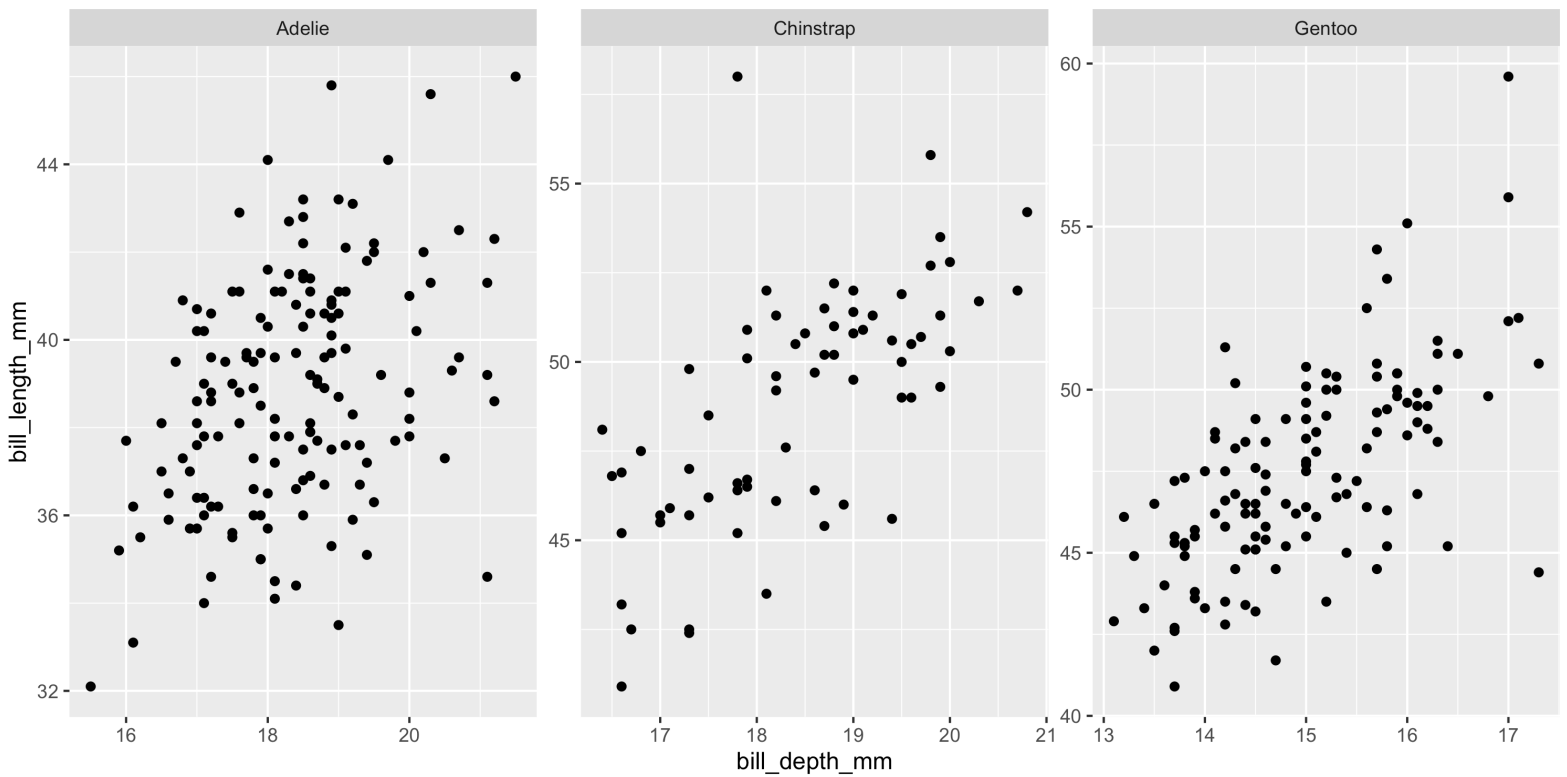
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE) +
  facet_wrap(~ species, ncol = 2, dir = "rb")
```



facet_wrap - free scales

By default, all facets share the same axis limits. Use `scales` to let axes vary across panels.

```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm)
) +
  geom_point(na.rm = TRUE) +
  facet_wrap(~ species, scales = "free")
```

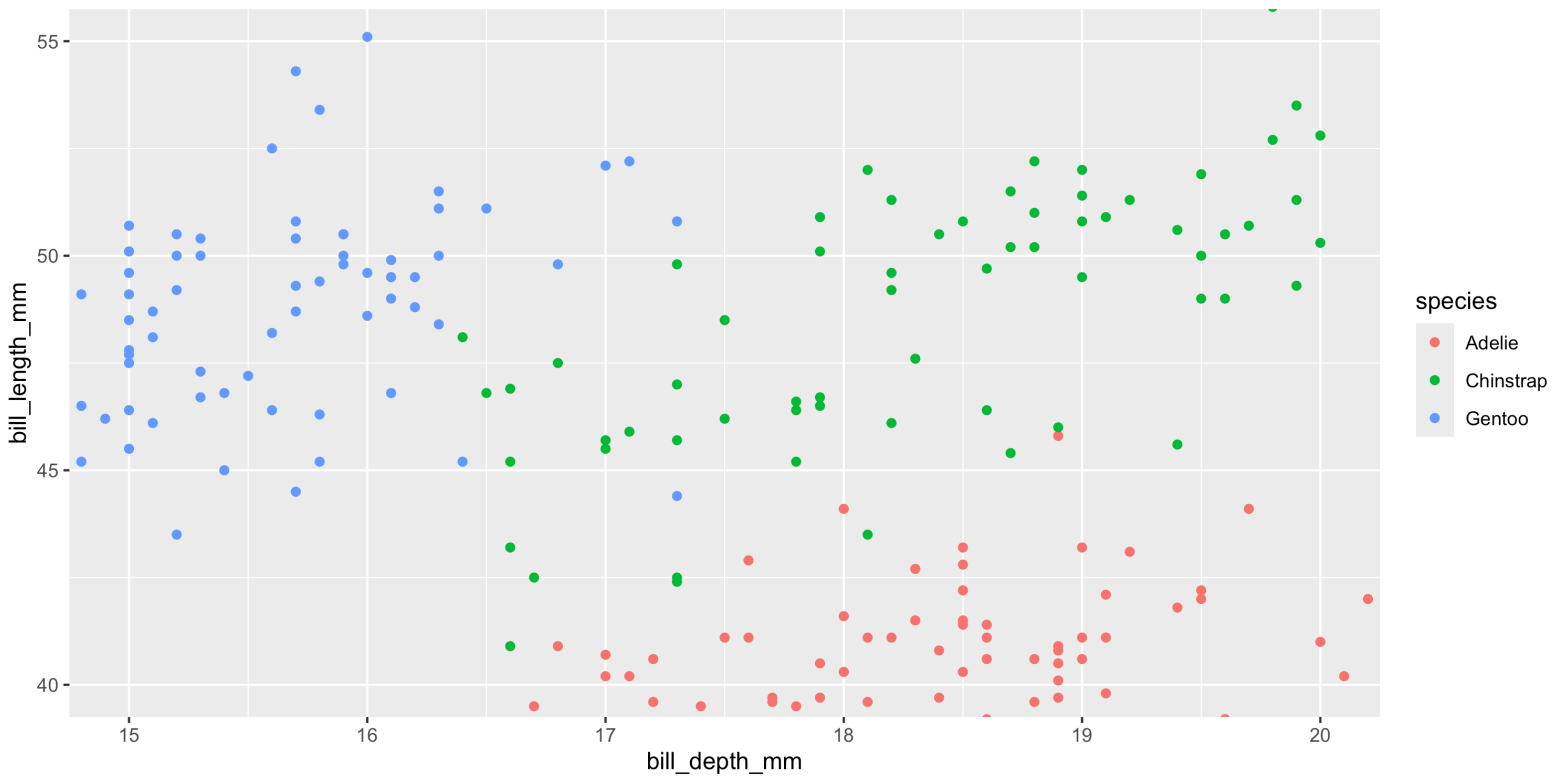


Coordinate Systems

coord_cartesian - zooming

`coord_cartesian()` zooms into the plot without dropping data — unlike setting `scale` limits which removes points before stat computations.

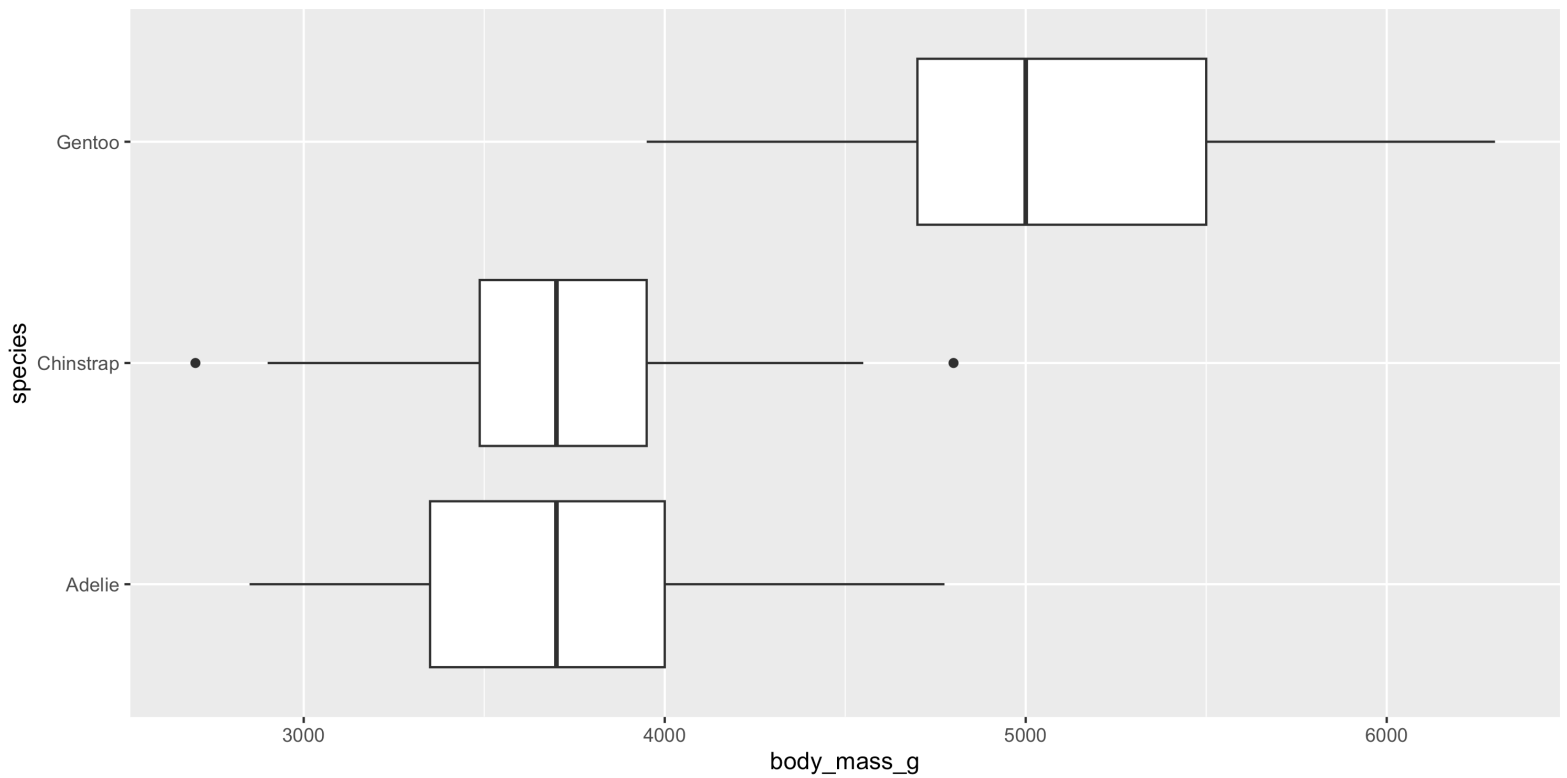
```
ggplot(
  penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)
) +
  geom_point(na.rm = TRUE) +
  coord_cartesian(xlim = c(15, 20), ylim = c(40, 55))
```



coord_flip

`coord_flip()` swaps the x and y axes — useful for making horizontal bar charts or boxplots more readable.

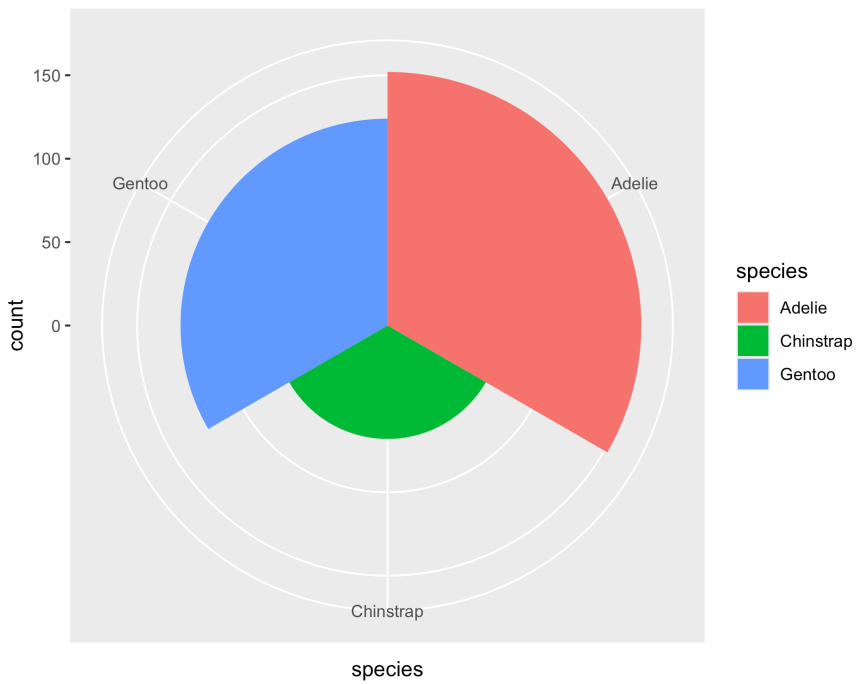
```
ggplot(
  penguins, aes(x = species, y = body_mass_g)
) +
  geom_boxplot(na.rm = TRUE) +
  coord_flip()
```



coord_polar

`coord_polar()` maps position onto a circular coordinate system.

```
ggplot(
  penguins |> dplyr::filter(!is.na(species)),
  aes(x = species, fill = species)
) +
  geom_bar(width = 1) +
  coord_polar()
```



Learning more

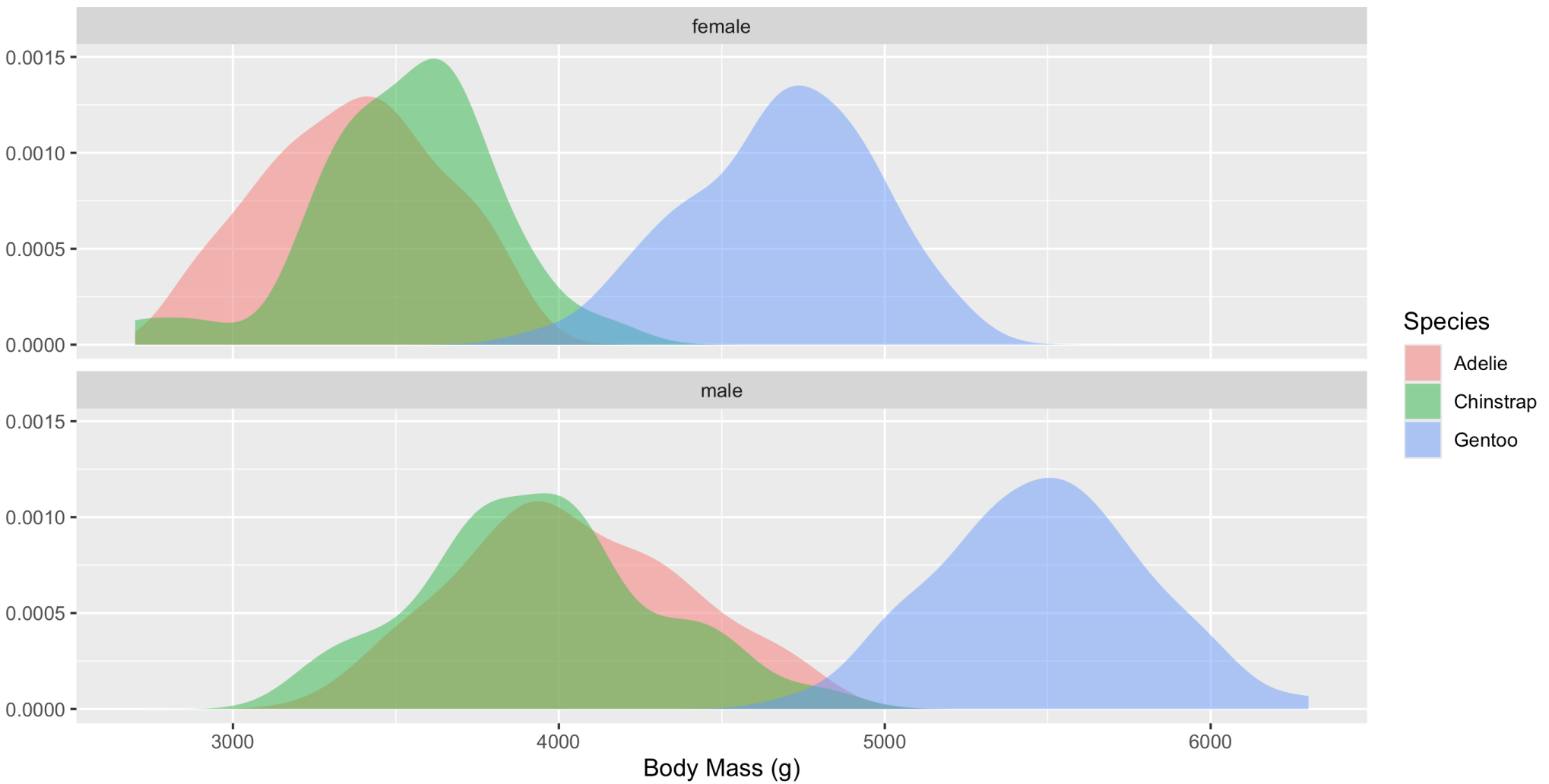
geom tour

ggplot2.tidyverse.org/reference/index.html

Exercises

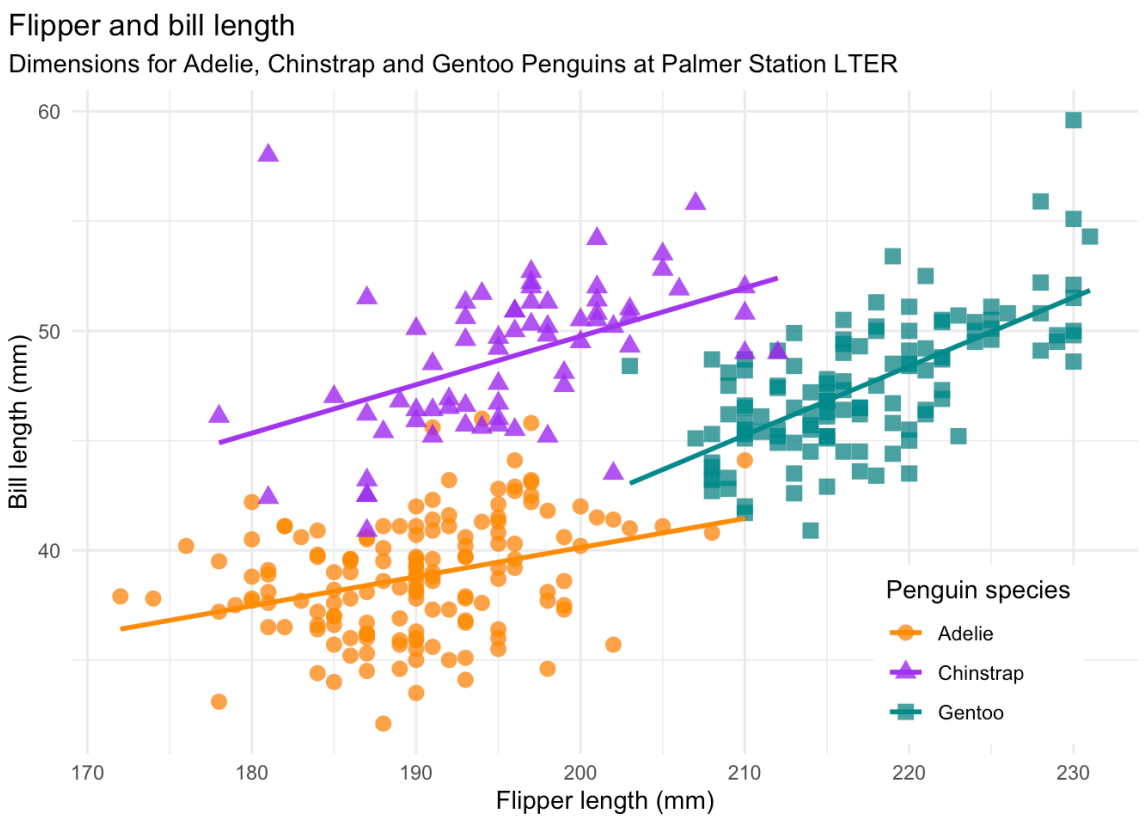
Exercise 1

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 `palmerpenguins` package README in ggplot2.



See the `palmerpenguins` pkgdown site