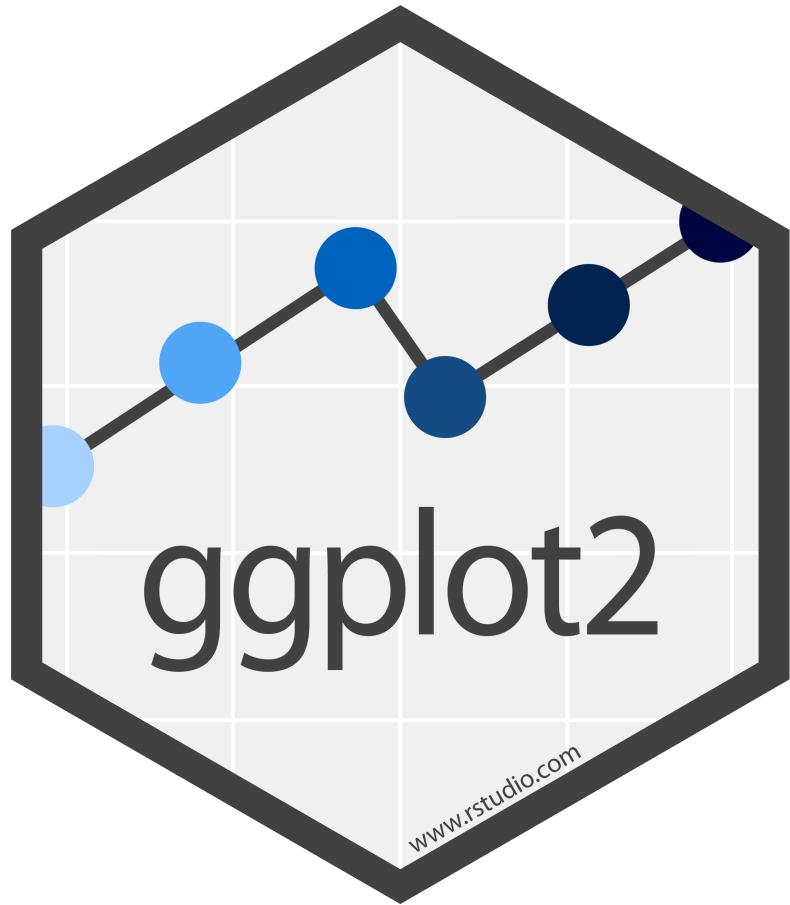


# Lec 09 - Visualization with ggplot2

Lecture 09

Dr. Colin Rundel



2

# The Grammar of Graphics

- Visualization concept created by Leland Wilkinson (*The Grammar of Graphics*, 1999)
- attempt to taxonimize 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...

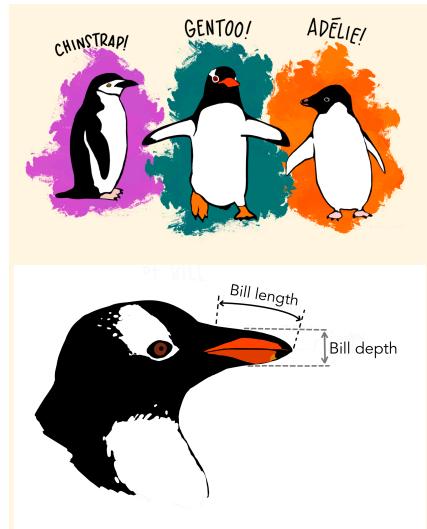
- mapping of **data**
- which may be **statistically transformed** (summarized, log-transformed, etc.)
- to **aesthetic attributes** (color, size, xy-position, etc.)
- using **geometric objects** (points, lines, bars, etc.)
- and mapped onto a specific **facet** and **coordinate system**

# 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.



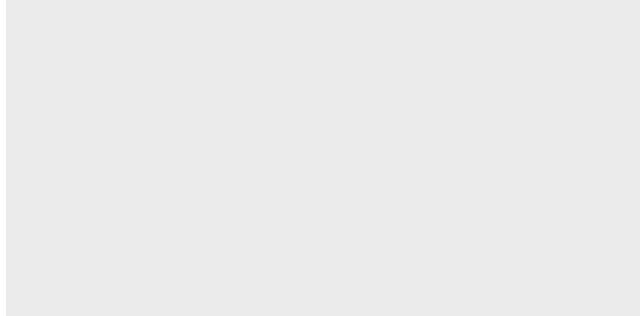
```
1 library(palmerpenguins)
2 penguins

# 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

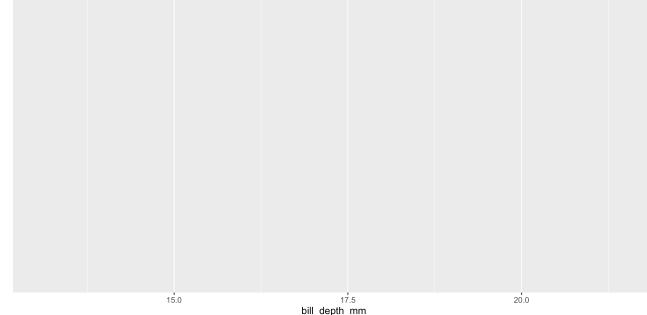
| Start with the `penguins` data frame

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



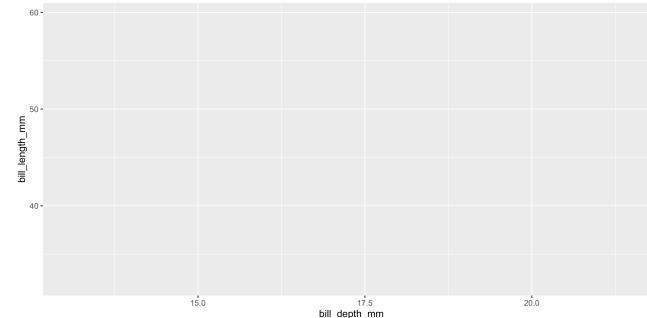
| Start with the `penguins` data frame, map bill depth to the x-axis

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



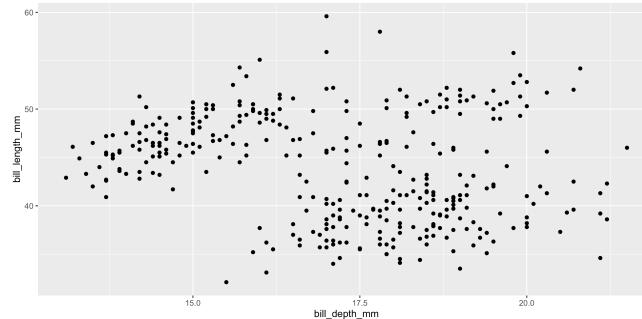
Start with the `penguins` data frame, map bill depth to the x-axis *and* map bill length to the y-axis.

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



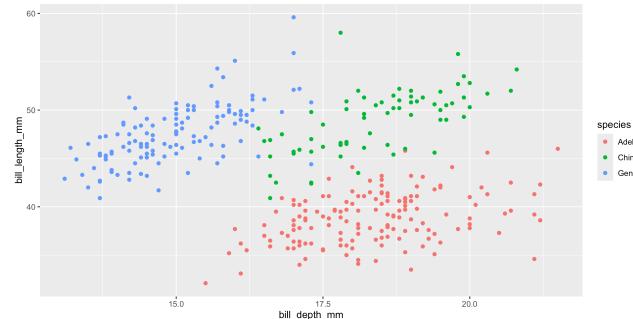
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis.  
*Represent each observation with a point*

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



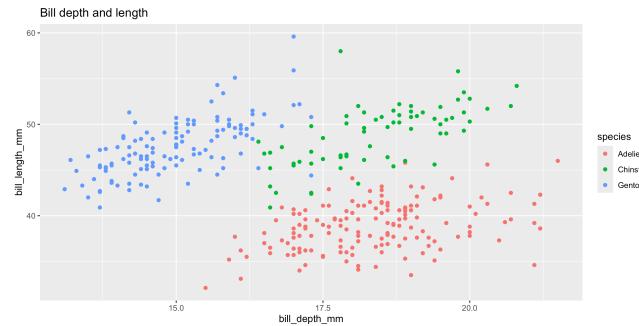
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis.  
Represent each observation with a point *and map species to the color of each point.*

```
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 )
```



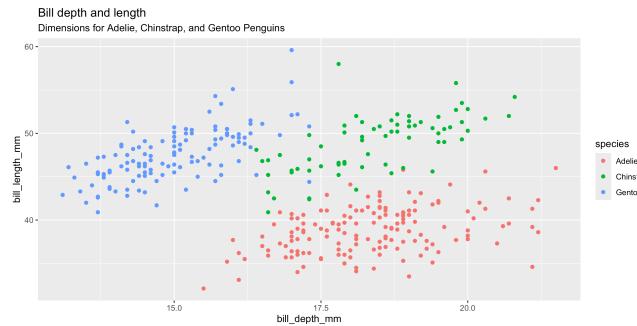
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. *Title the plot “Bill depth and length”*

```
1 ggpplot(  
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")
```



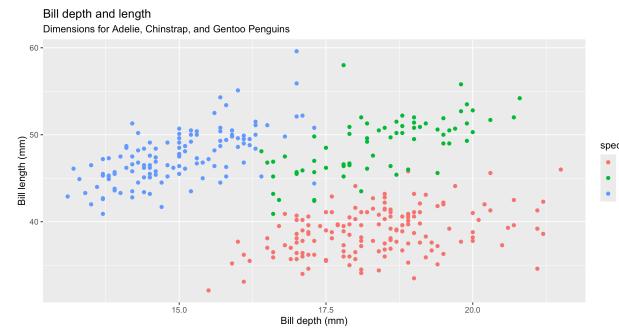
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”

```
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("Dimensions for Adelie,", 
14                      "Chinstrap, and Gentoo",
15                      "Penguins")
16   )
```



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("Dimensions for Adelie,", 
14                      "Chinstrap, and Gentoo",
15                      "Penguins"),
16     x = "Bill depth (mm)",
17     y = "Bill length (mm)"
18   )
```

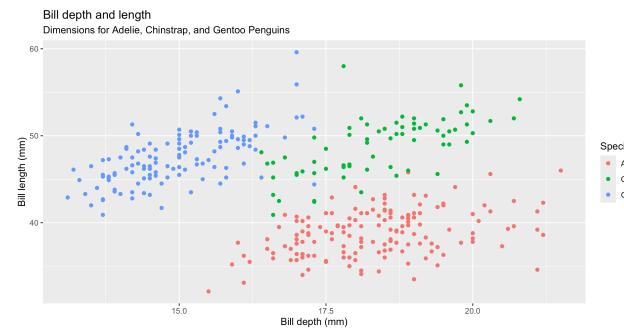


Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label 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("Dimensions for Adelie,",,
14                   "Chinstrap, and Gentoo",
15                   "Penguins"),
16     x = "Bill depth (mm)",
17     y = "Bill length (mm)",
18     color = "Species"
19 )

```

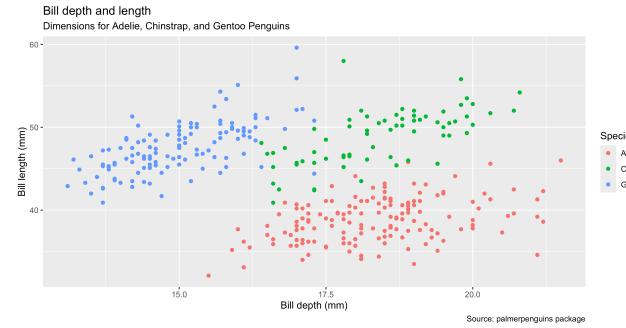


Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label the x and y axes as “Bill depth (mm)” and “Bill length (mm)”, respectively, label the legend “Species”, 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("Dimensions for Adelie,",
14                      "Chinstrap, and Gentoo",
15                      "Penguins"),
16     x = "Bill depth (mm)",
17     y = "Bill length (mm)",
18     color = "Species",
19     caption = "Source: palmerpenguins package"
20   )

```

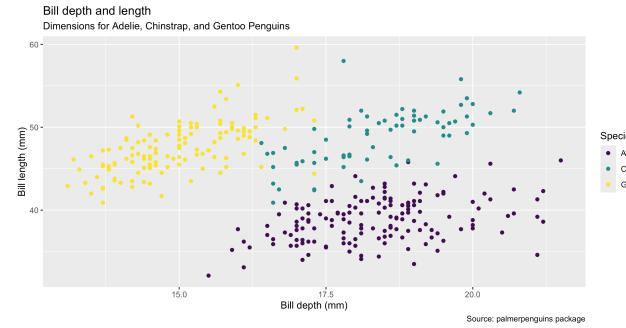


Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot “Bill depth and length”, add the subtitle “Dimensions for Adelie, Chinstrap, and Gentoo Penguins”, label 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. 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("Dimensions for Adelie",
14     "Chinstrap, and Gentoo",
15     "Penguins"),
16   x = "Bill depth (mm)",
17   y = "Bill length (mm)",
18   color = "Species",
19   caption = "Source: palmerpenguins package"
20 ) +
21   scale_color_viridis_d()

```



# Aesthetics

19

## 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)

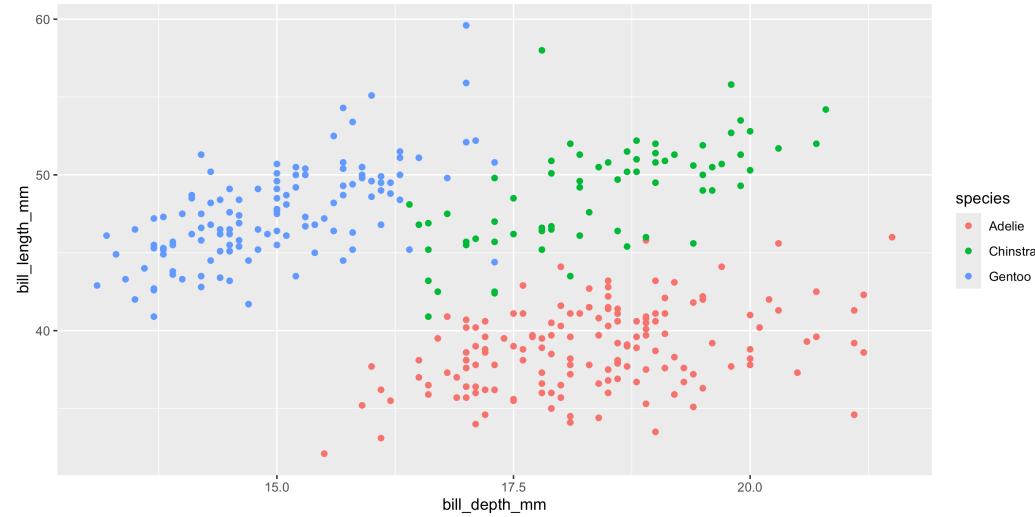
Different geometries have different aesthetics available - see the `ggplot2 geoms` help files for listings.

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

# color

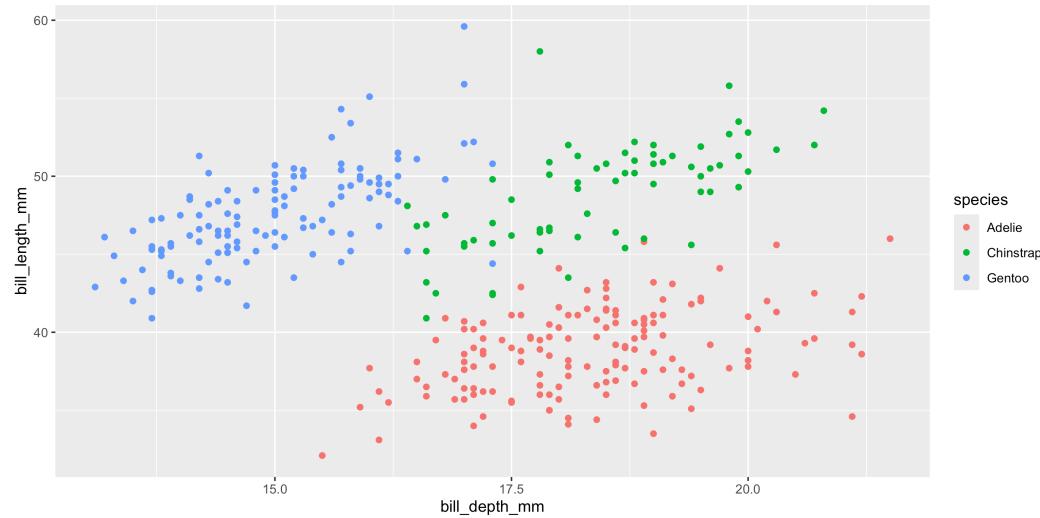
```
1 ggpplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species)  
6   )
```

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



# Stop the warning

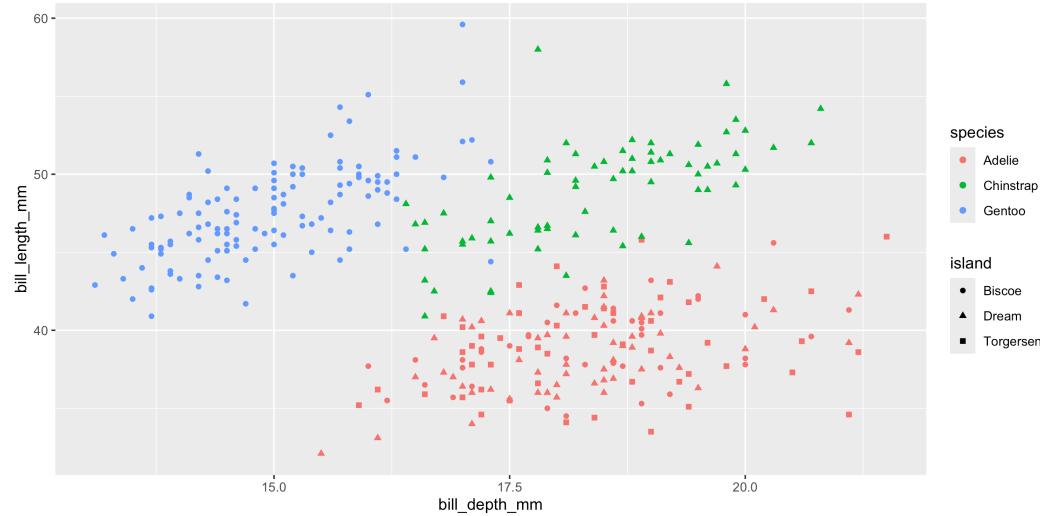
```
1 ggpplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species), na.rm=TRUE  
6   )
```



# Shape

Mapped to a different variable than color

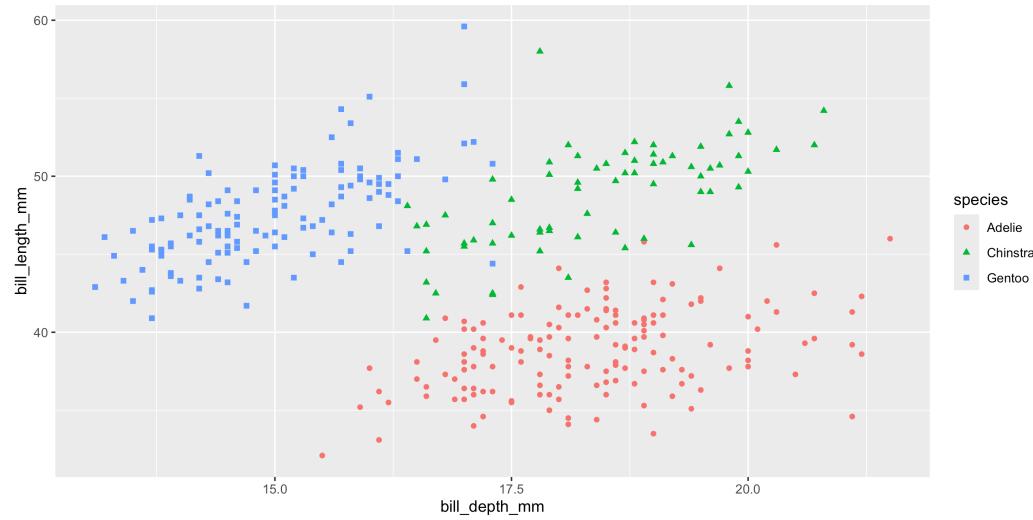
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = island), na.rm = TRUE  
6   )
```



# Shape

Mapped to same variable as color

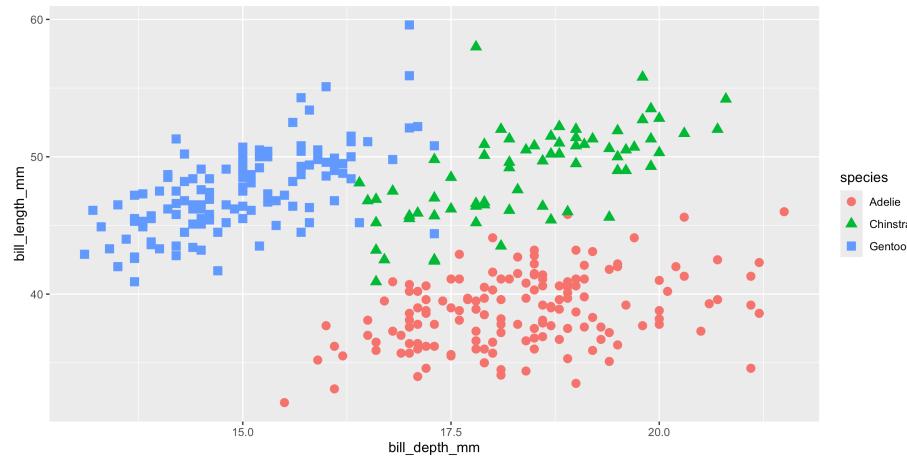
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = species), na.rm = TRUE  
6   )
```



# Size

Using a fixed value (note this value is outside of the `aes` call)

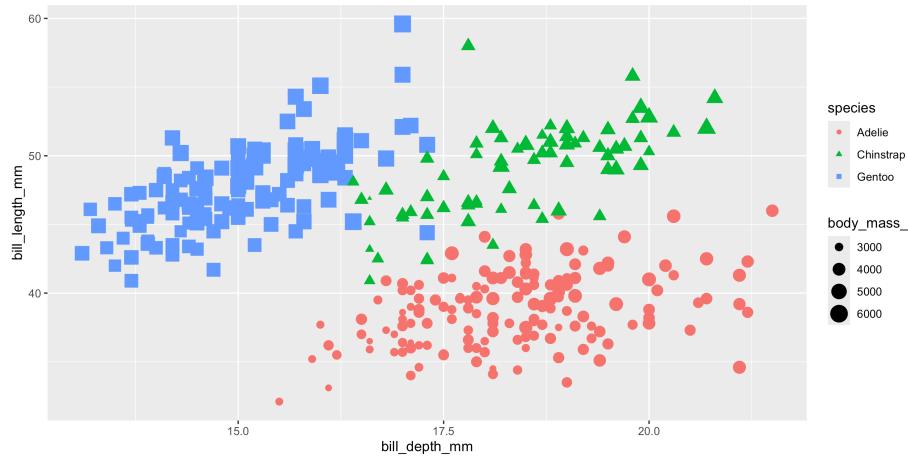
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm)  
3 ) +  
4   geom_point(  
5     aes(color = species, shape = species), na.rm = TRUE,  
6     size = 3  
7   )
```



# Size

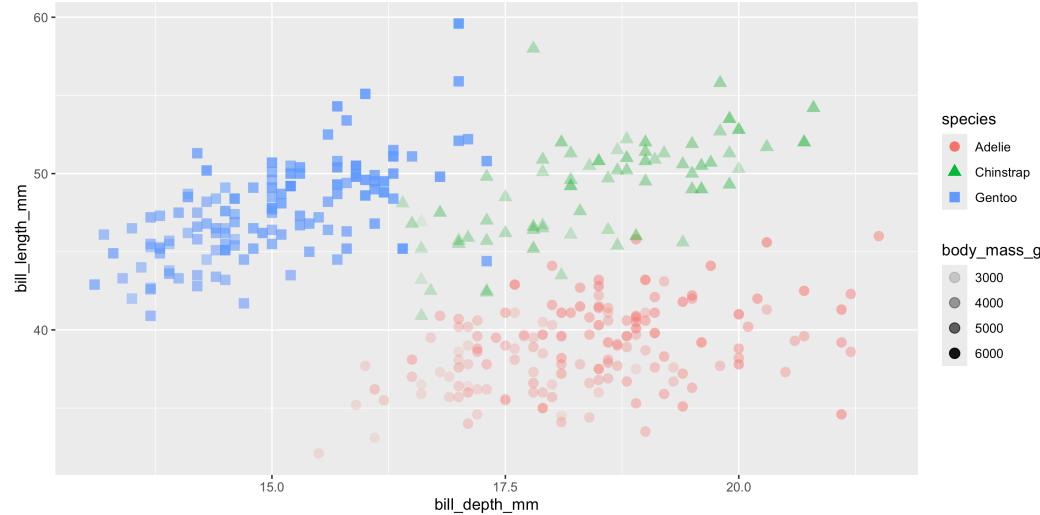
Mapped to a variable

```
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 ggpplot(  
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 **not** based on the values of a variable in the data, usually a constant value.
  - passed directly into `geom_*`() as an argument.

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

# Faceting

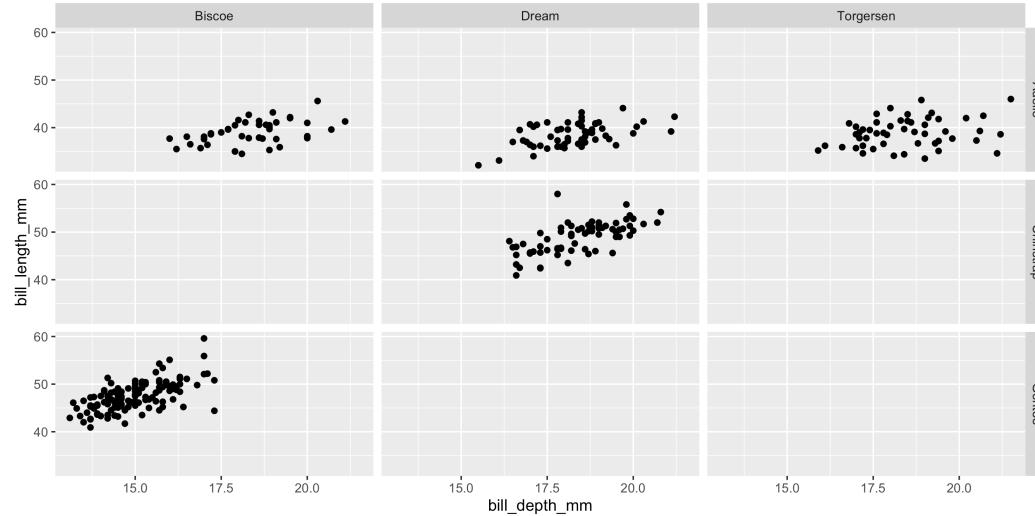
29

# Faceting

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

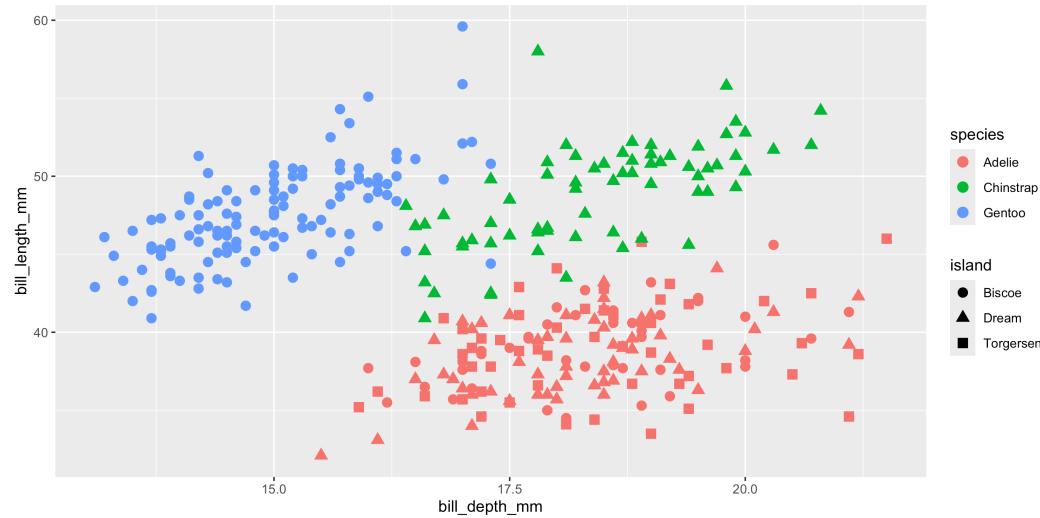
# facet\_grid

```
1 ggpplot(  
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   )
```



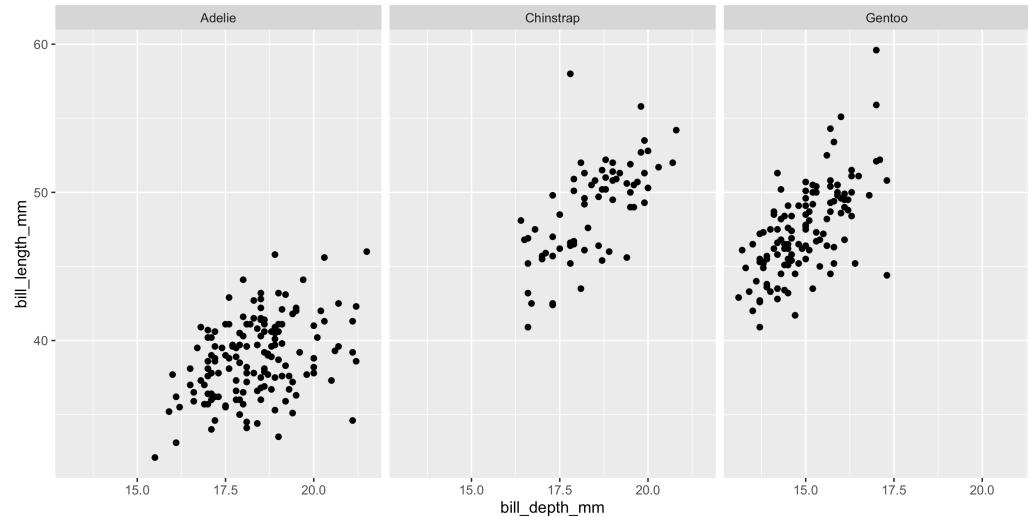
# Compare with ...

```
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   )
```



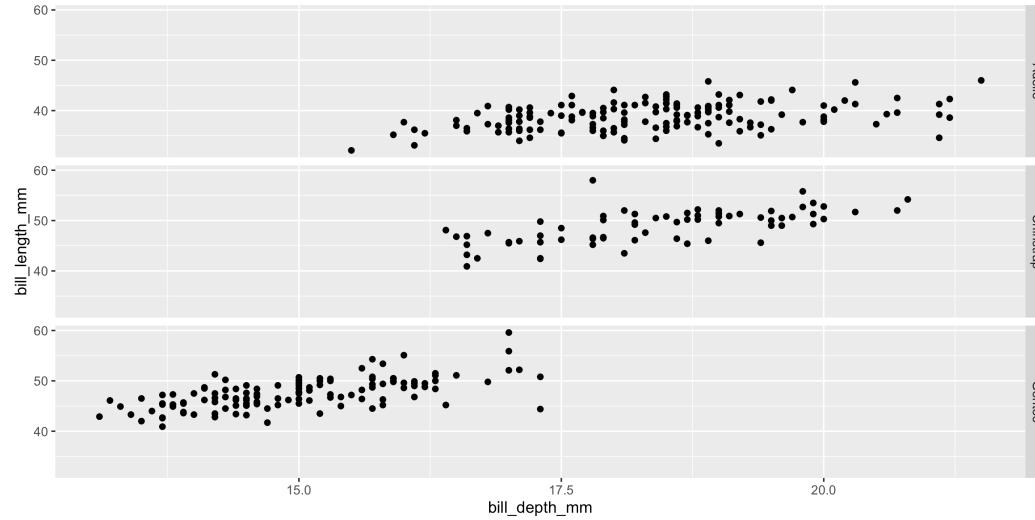
## facet\_grid (cols)

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



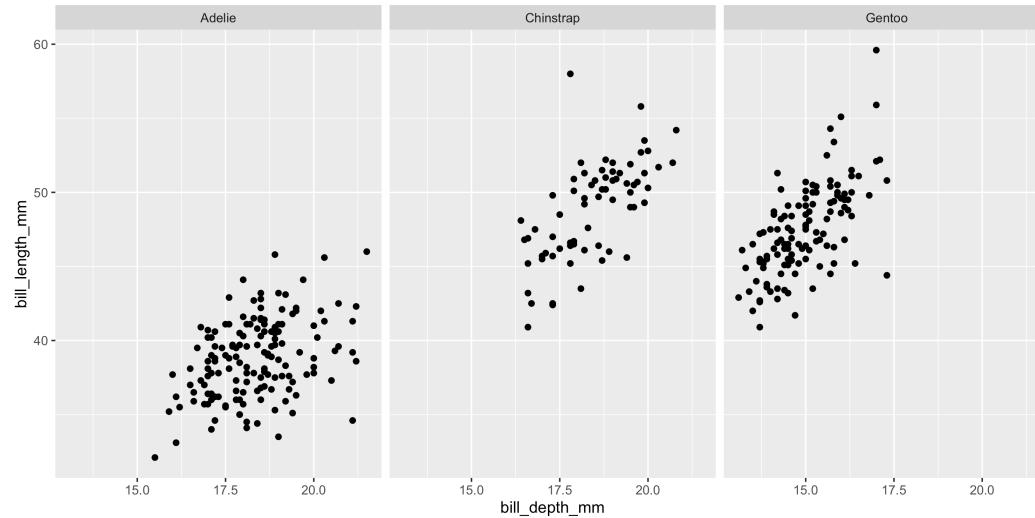
# facet\_grid (rows)

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



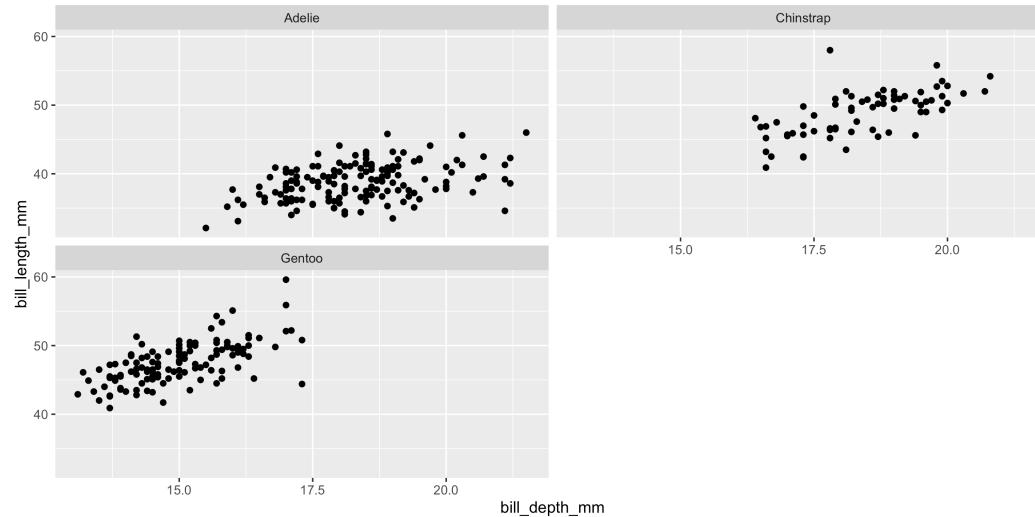
# facet\_wrap

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



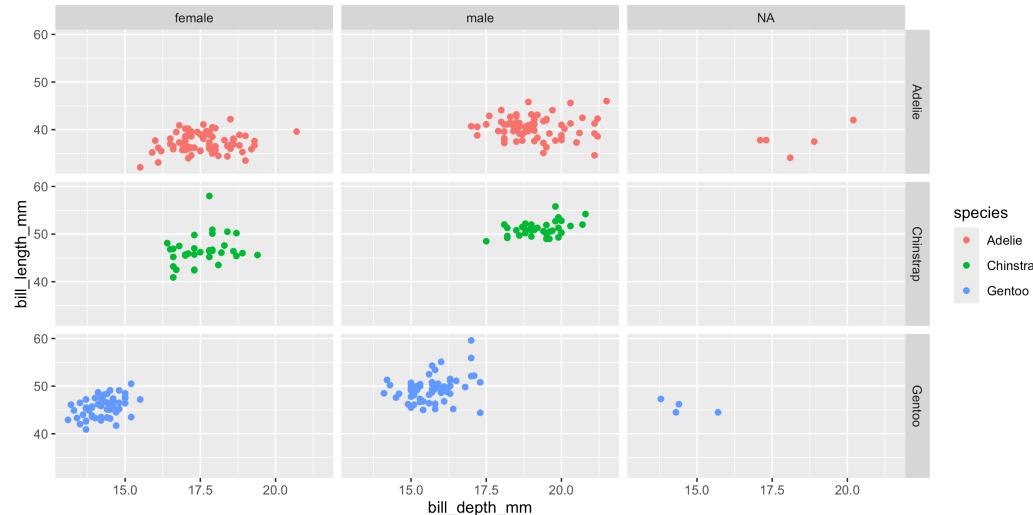
# facet\_wrap

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



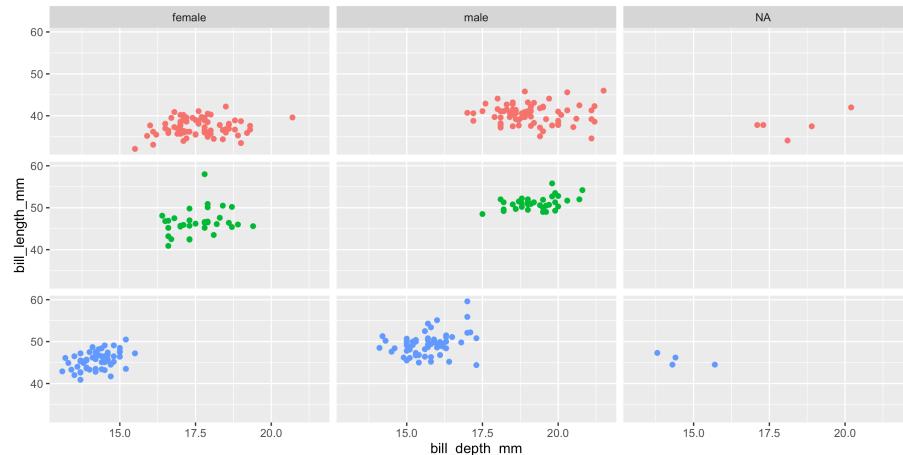
# Faceting and color

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



# Hiding legend elements

```
1 ggpplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point(na.rm = TRUE) +  
5   facet_grid(  
6     species ~ sex  
7   ) +  
8   guides(color = "none")
```



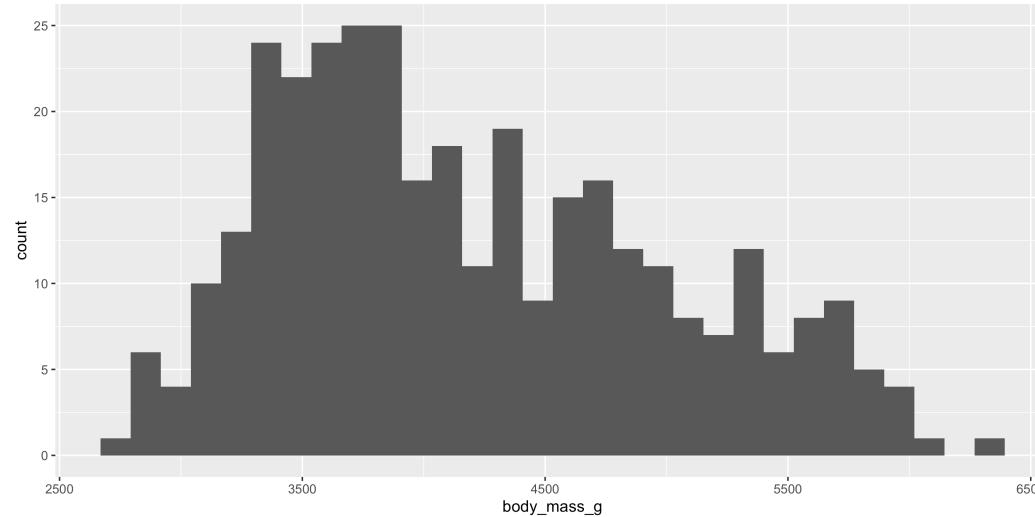
# A brief plot tour of ggplot2 plots

39

# Histograms

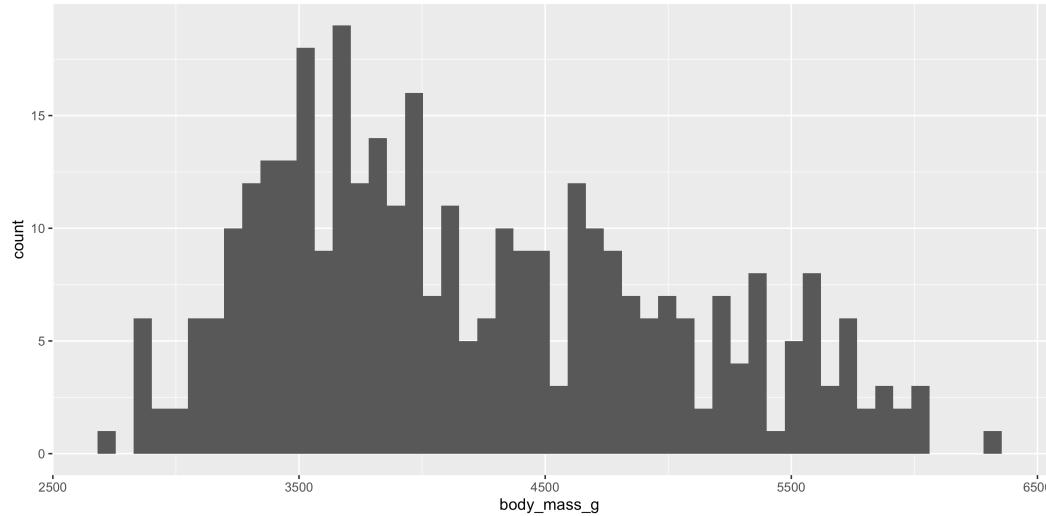
```
1 ggplot(  
2   penguins, aes(x = body_mass_g)  
3 ) +  
4   geom_histogram()
```

`stat\_bin()` using `bins = 30`. Pick better value with  
`binwidth`.



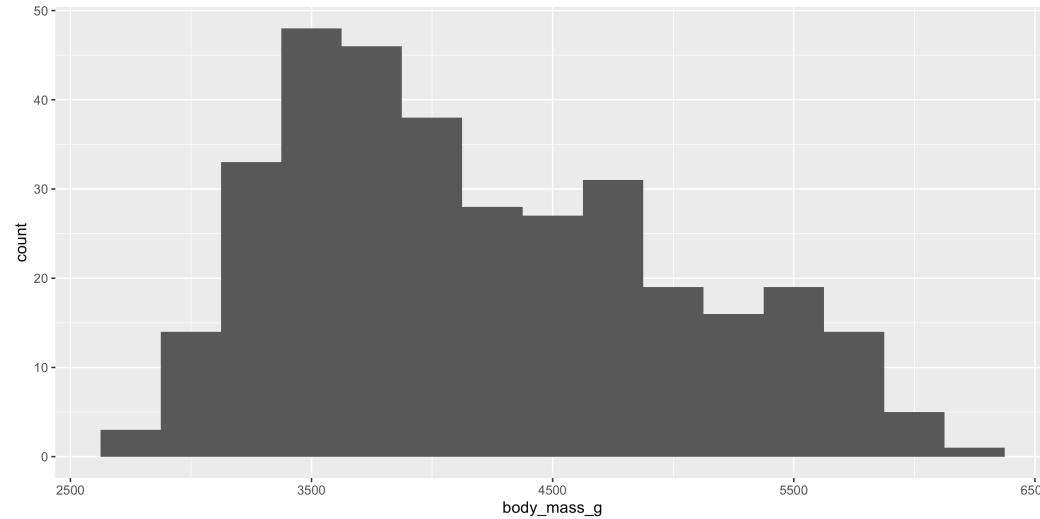
## Histograms - bins

```
1 ggplot(  
2   penguins, aes( x = body_mass_g )  
3 ) +  
4   geom_histogram(bins = 50)
```



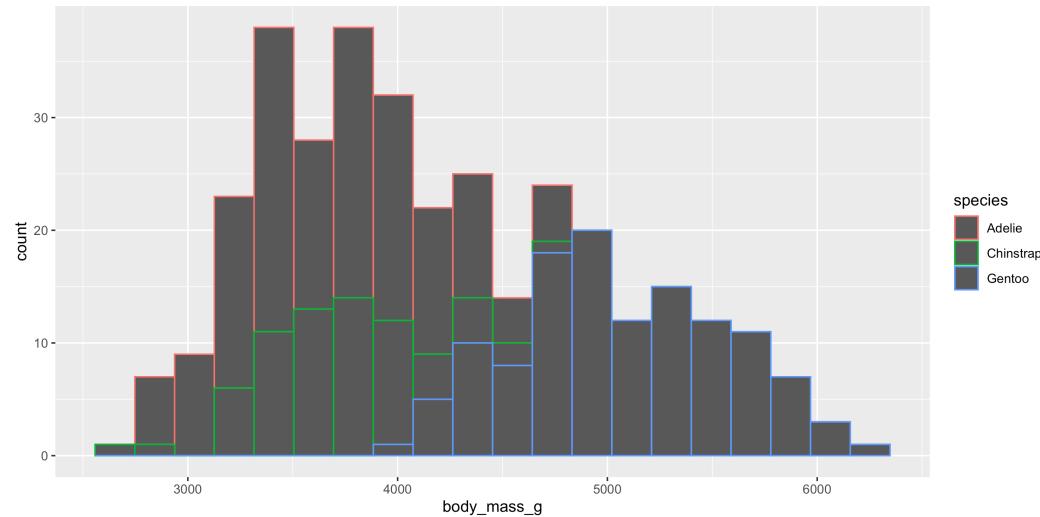
# Histograms - binwidth

```
1 ggplot(  
2   penguins, aes( x = body_mass_g )  
3 ) +  
4   geom_histogram(binwidth = 250)
```



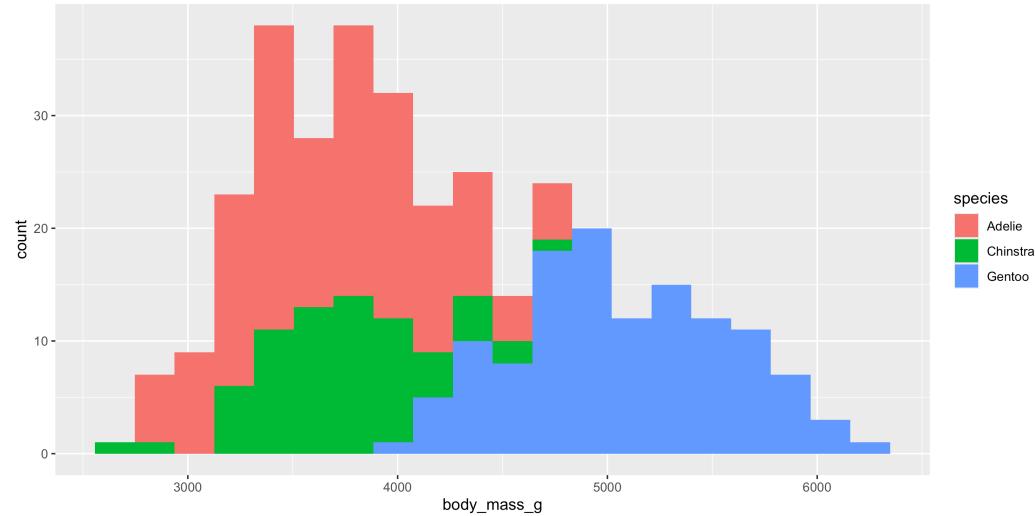
# Histograms - color

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, color = species)  
3 ) +  
4   geom_histogram(bins = 20)
```



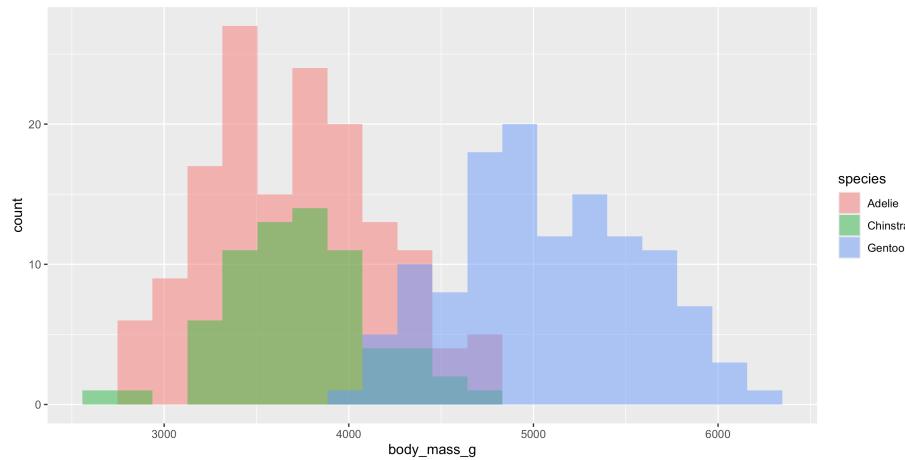
## Histograms - fill

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, fill = species)  
3 ) +  
4   geom_histogram(bins = 20)
```



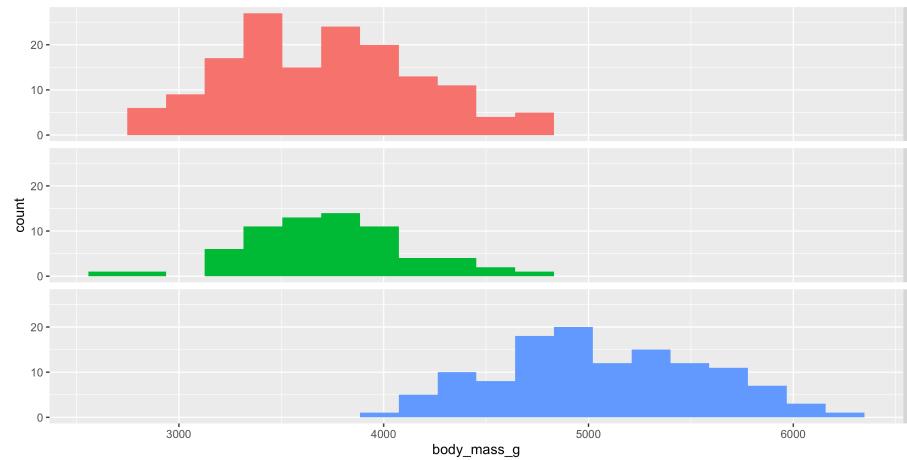
# Histograms - position

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, fill = species)  
3 ) +  
4   geom_histogram(  
5     bins = 20, alpha = 0.5, position = "identity"  
6   )
```



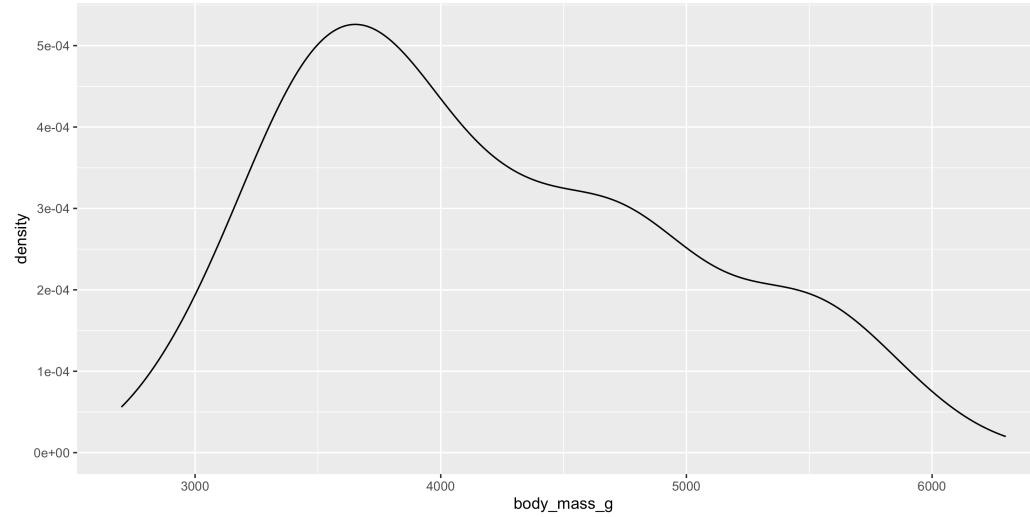
# Histograms - facets

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, fill = species))  
3 ) +  
4   geom_histogram(bins = 20) +  
5   facet_grid(species ~ .) +  
6   guides(fill = "none")
```



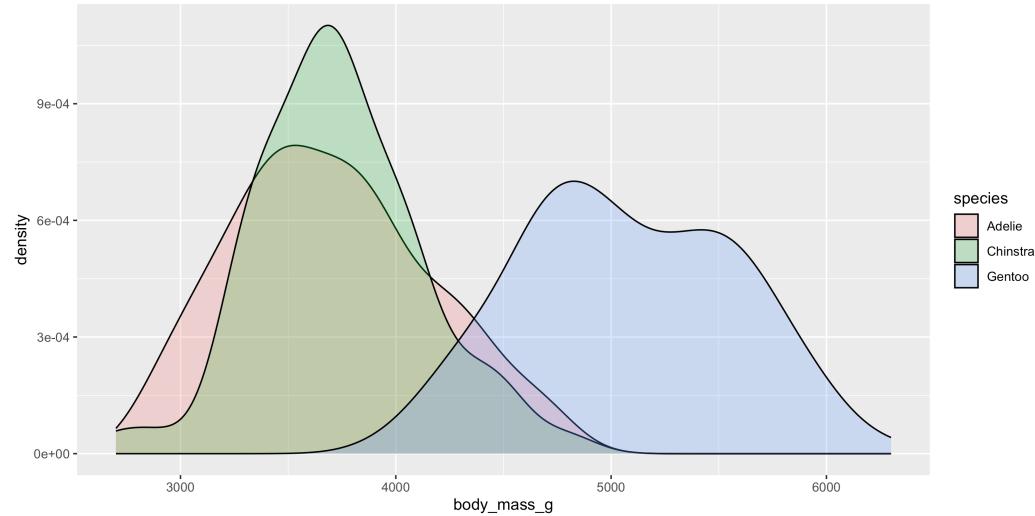
# Density plot

```
1 ggplot(  
2   penguins, aes(x = body_mass_g)  
3 ) +  
4   geom_density()
```



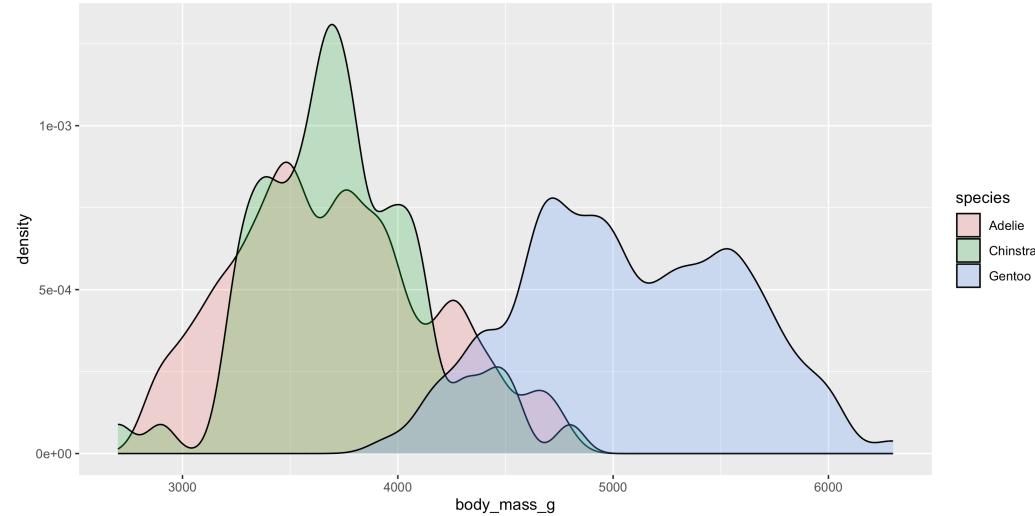
## Density plot - fill

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, fill = species))  
3 ) +  
4   geom_density(alpha = 0.25)
```



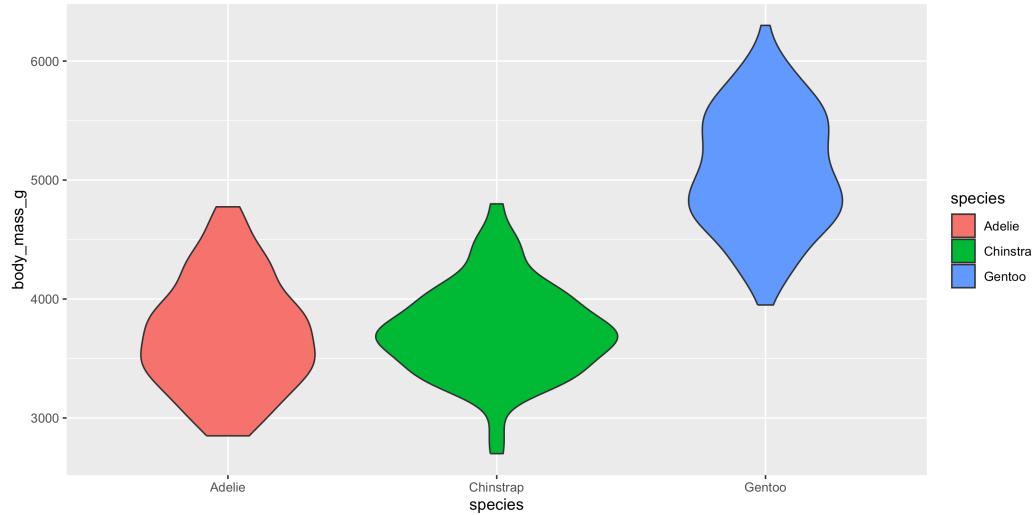
# Density plot - adjust

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, fill = species))  
3 ) +  
4   geom_density(  
5     adjust = 0.5, alpha = 0.25  
6   )
```



# Violin plot

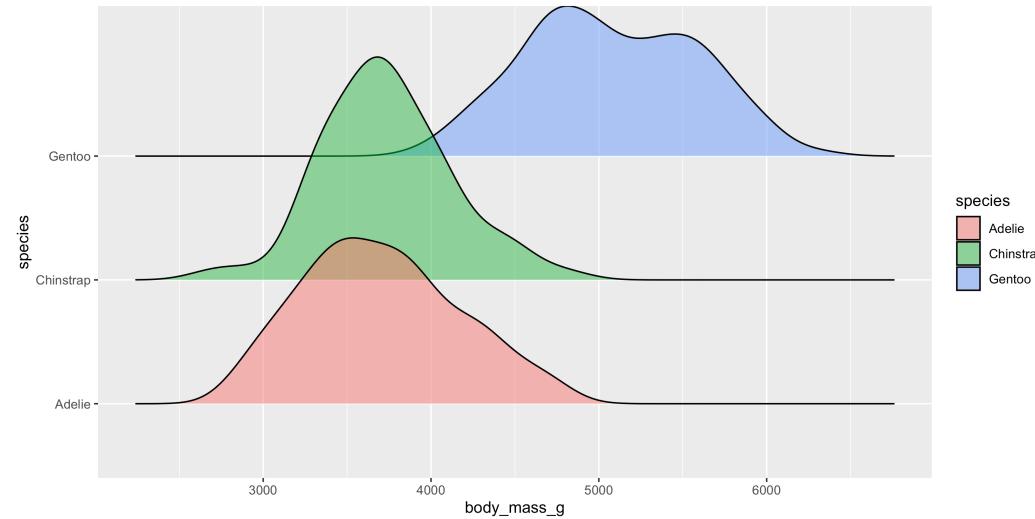
```
1 ggplot(  
2   penguins, aes(x = species, fill = species, y = body_mass_g )  
3 ) +  
4   geom_violin()
```



# Ridge plot

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, y = species, fill = species))  
3 ) +  
4   ggridges::geom_density_ridges(alpha = 0.5)
```

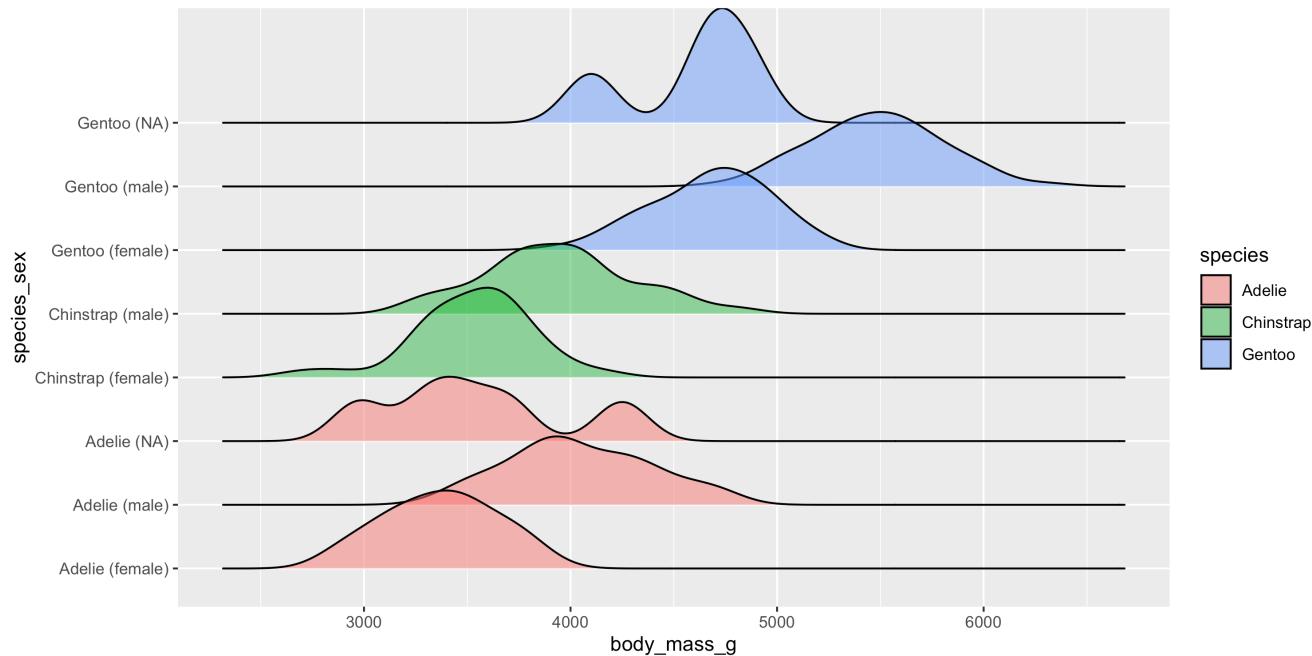
Picking joint bandwidth of 153



# Ridge plot - more categories + dplyr

```
1 penguins |>
2   mutate( species_sex = paste0( species, " (", sex, ")") ) |>
3   ggplot( aes(x = body_mass_g, y = species_sex, fill = species) ) +
4   ggridges::geom_density_ridges( alpha = 0.5 )
```

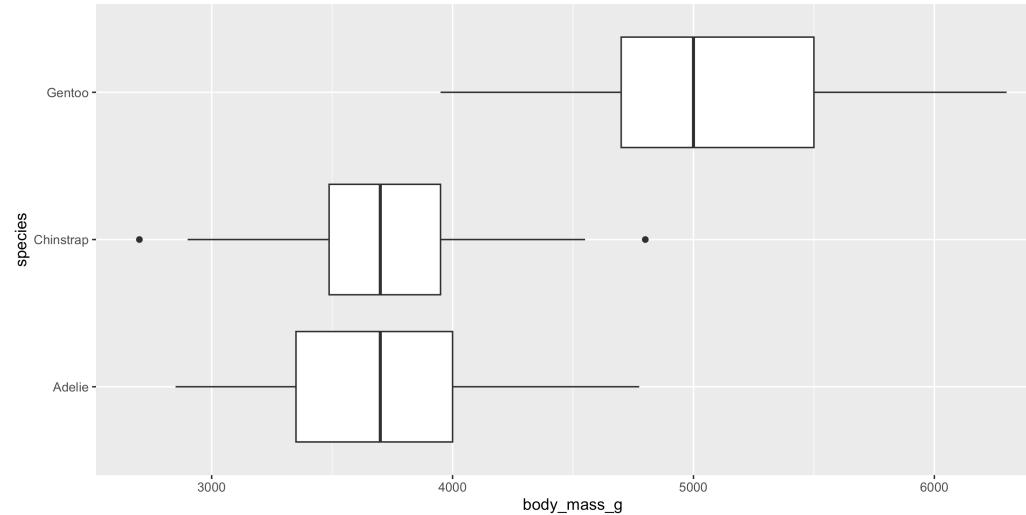
Picking joint bandwidth of 127





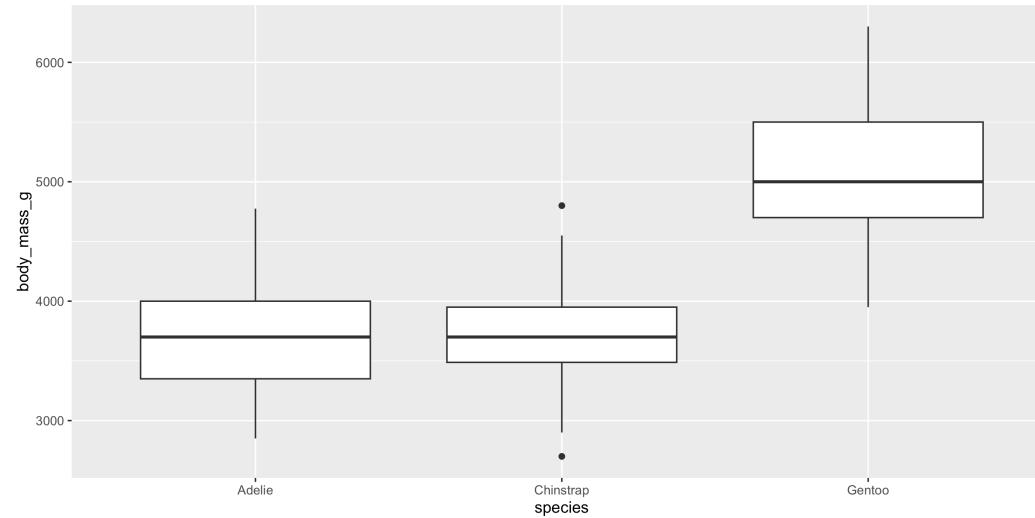
# Box plot

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, y = species)  
3 ) +  
4   geom_boxplot()
```



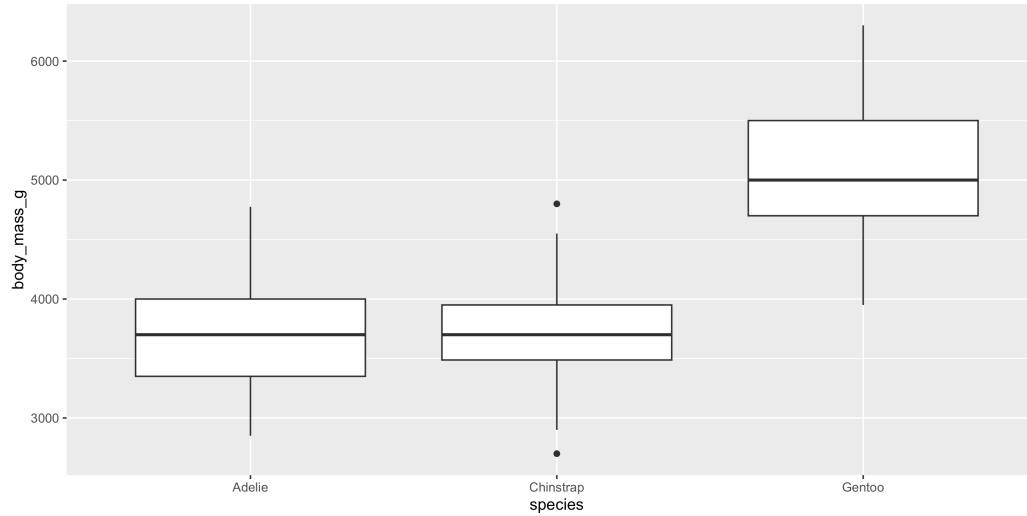
## Box plot - coord\_flip

```
1 ggplot(  
2   penguins, aes(x = body_mass_g, y = species)  
3 ) +  
4   geom_boxplot() +  
5   coord_flip()
```



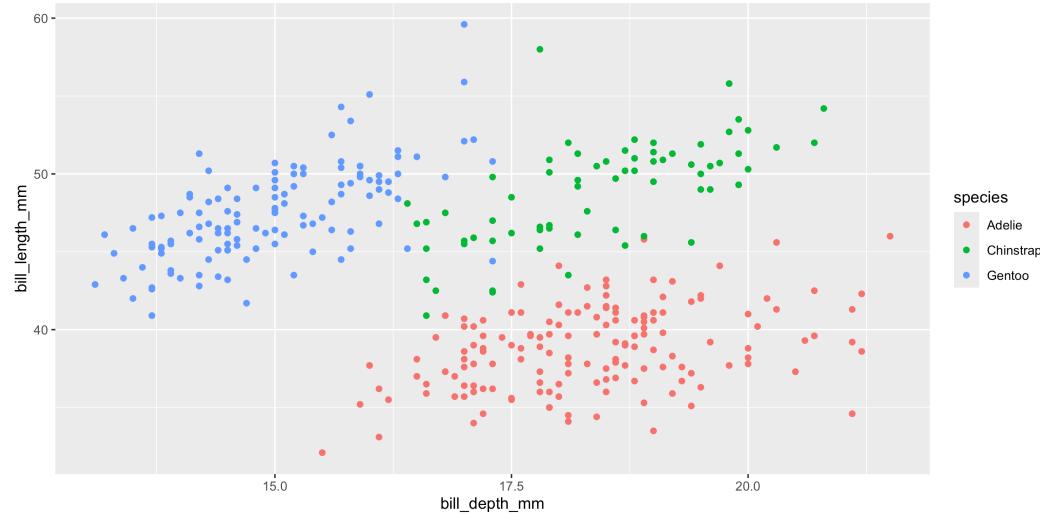
## Box plot - swap coords

```
1 ggplot(  
2   penguins, aes(x = species, y = body_mass_g)  
3 ) +  
4   geom_boxplot()
```



# Scatter plot

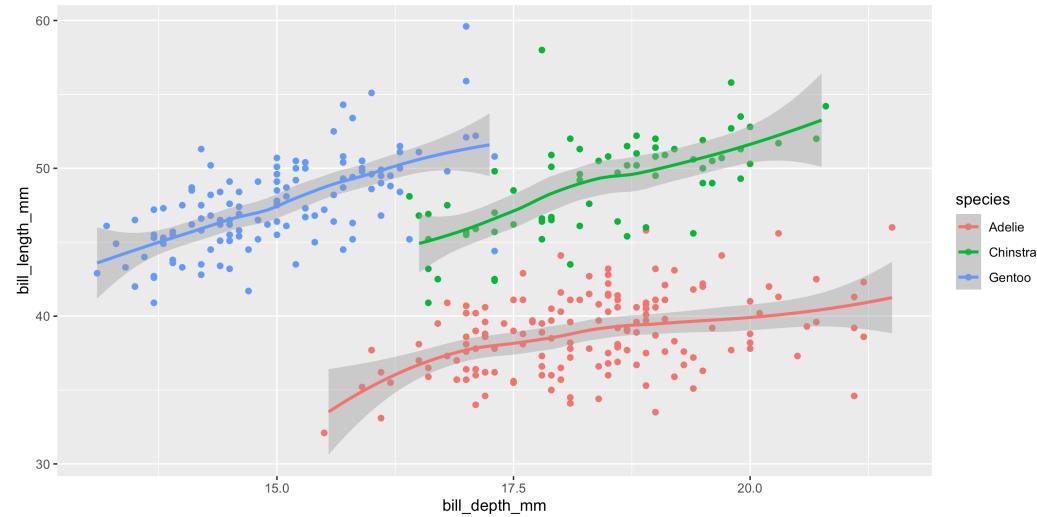
```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species)  
3 ) +  
4   geom_point()
```



# Scatter plot - geom\_smooth

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

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

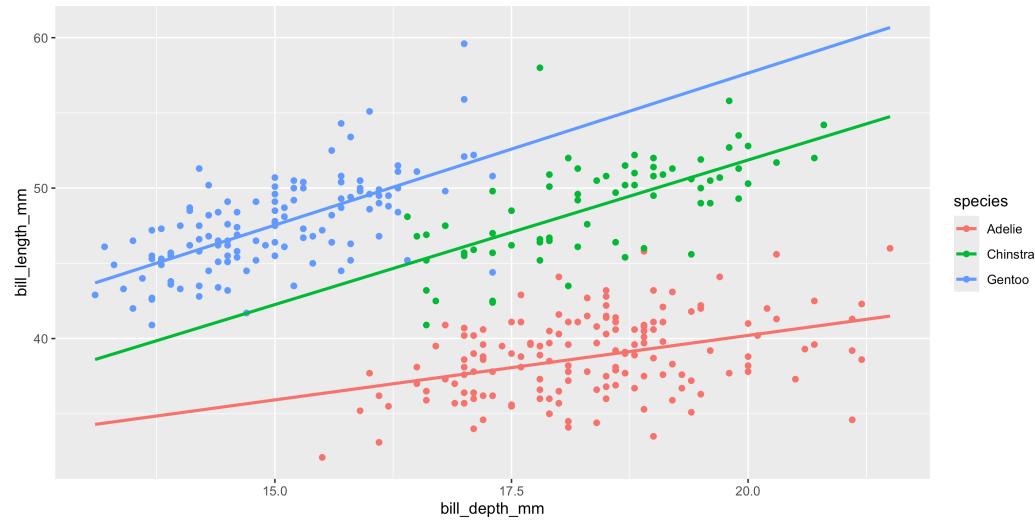




# Scatter plot - geom\_smooth w/ lm

```
1 ggplot(  
2   penguins, aes(x = bill_depth_mm, y = bill_length_mm, color = species))  
3 ) +  
4   geom_point() +  
5   geom_smooth(  
6     method = "lm", se = FALSE, fullrange = TRUE  
7   )
```

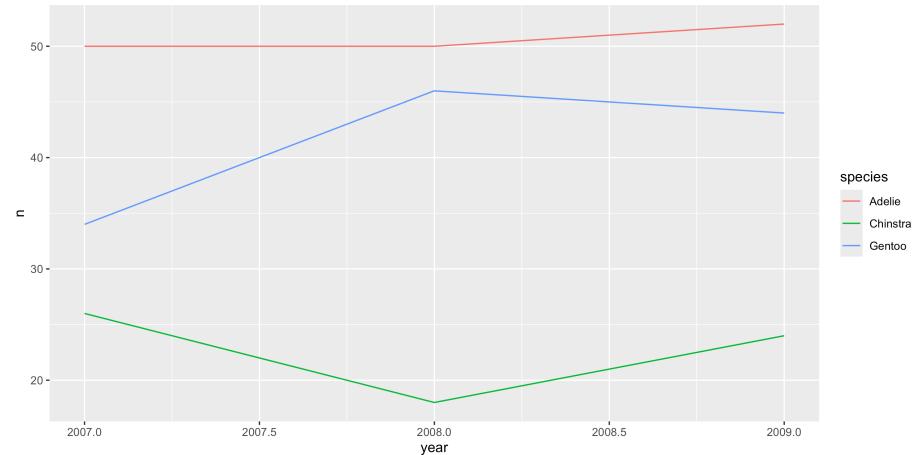
`geom\_smooth()` using formula = 'y ~ x'





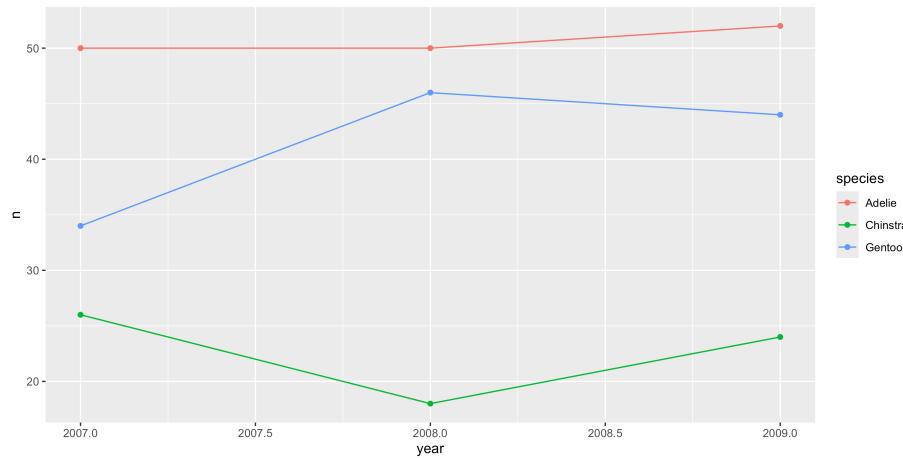
# Line plot

```
1 penguins |>
2   count(species, year) |>
3   ggplot(
4     aes(x = year, y = n, color = species, group = species)
5   ) +
6   geom_line()
```



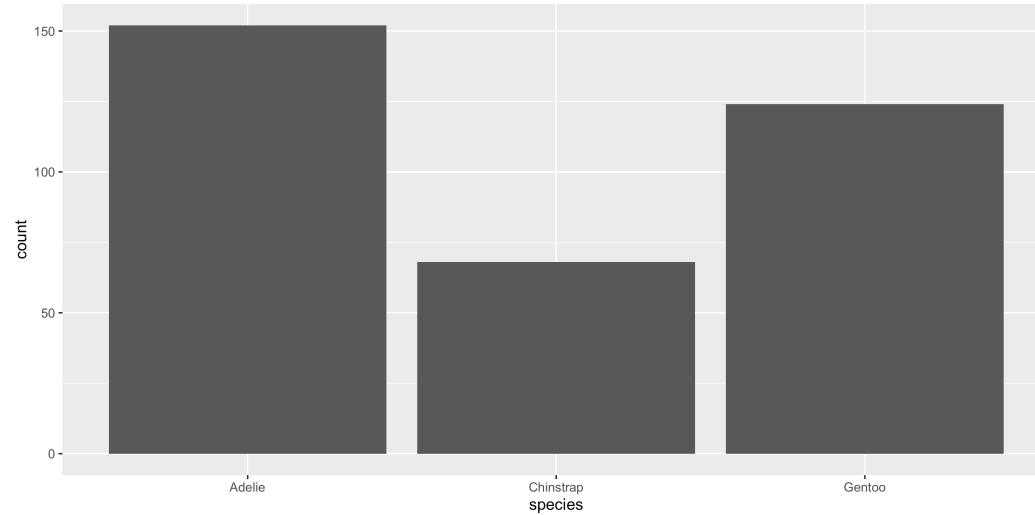
# Line plot - with points

```
1 penguins |>
2   count(species, year) |>
3   ggplot(
4     aes(x = year, y = n, color = species, group = species)
5   ) +
6   geom_line() +
7   geom_point()
```



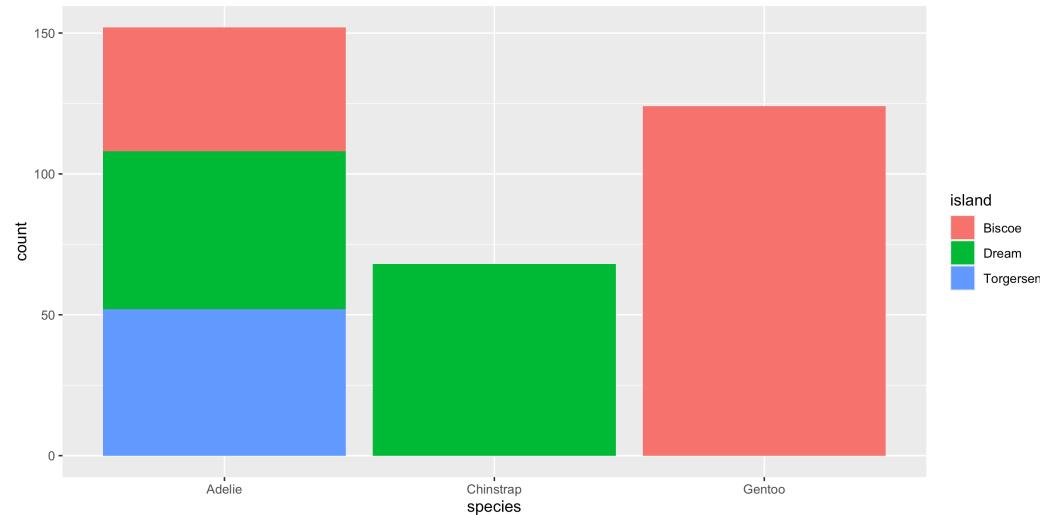
# Bar plot

```
1 ggplot(  
2   penguins, aes(x = species)  
3 ) +  
4   geom_bar()
```



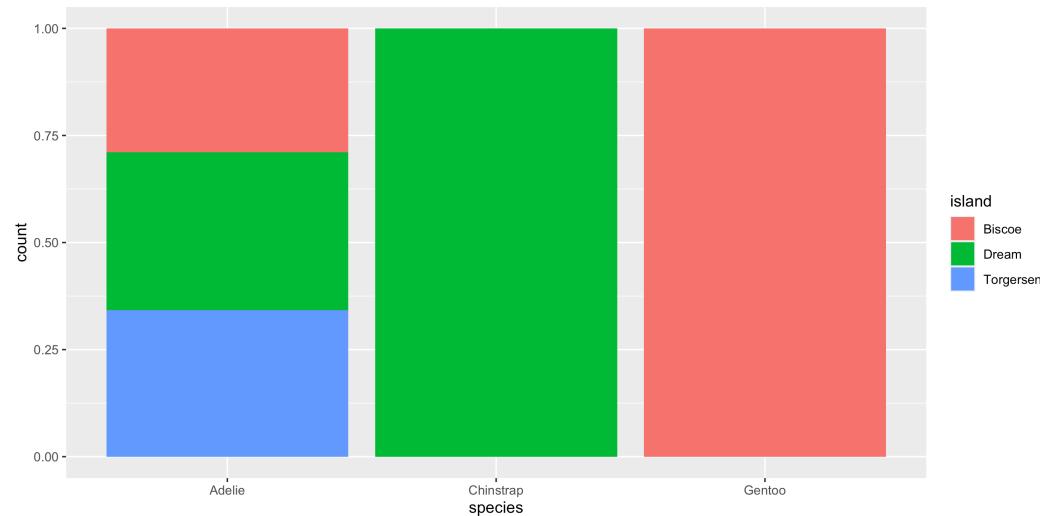
# Stacked bar plot

```
1 ggplot(  
2   penguins, aes(x = species, fill = island)  
3 ) +  
4   geom_bar()
```



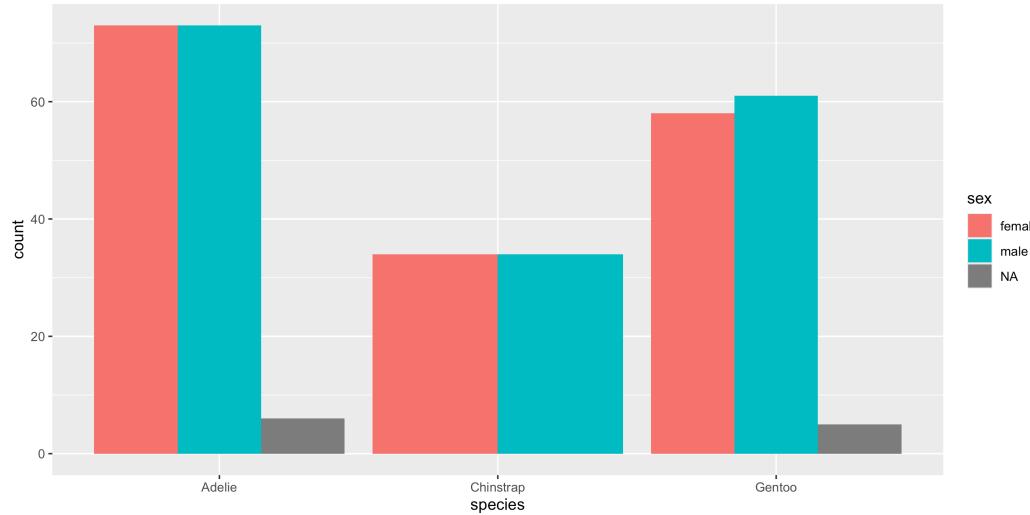
# Stacked relative frequency bar plot

```
1 ggplot(  
2   penguins, aes(x = species, fill = island)  
3 ) +  
4   geom_bar(position = "fill")
```



# Dodged bar plot

```
1 ggplot(  
2   penguins, aes(x = species, fill = sex)  
3 ) +  
4   geom_bar(position = "dodge")
```

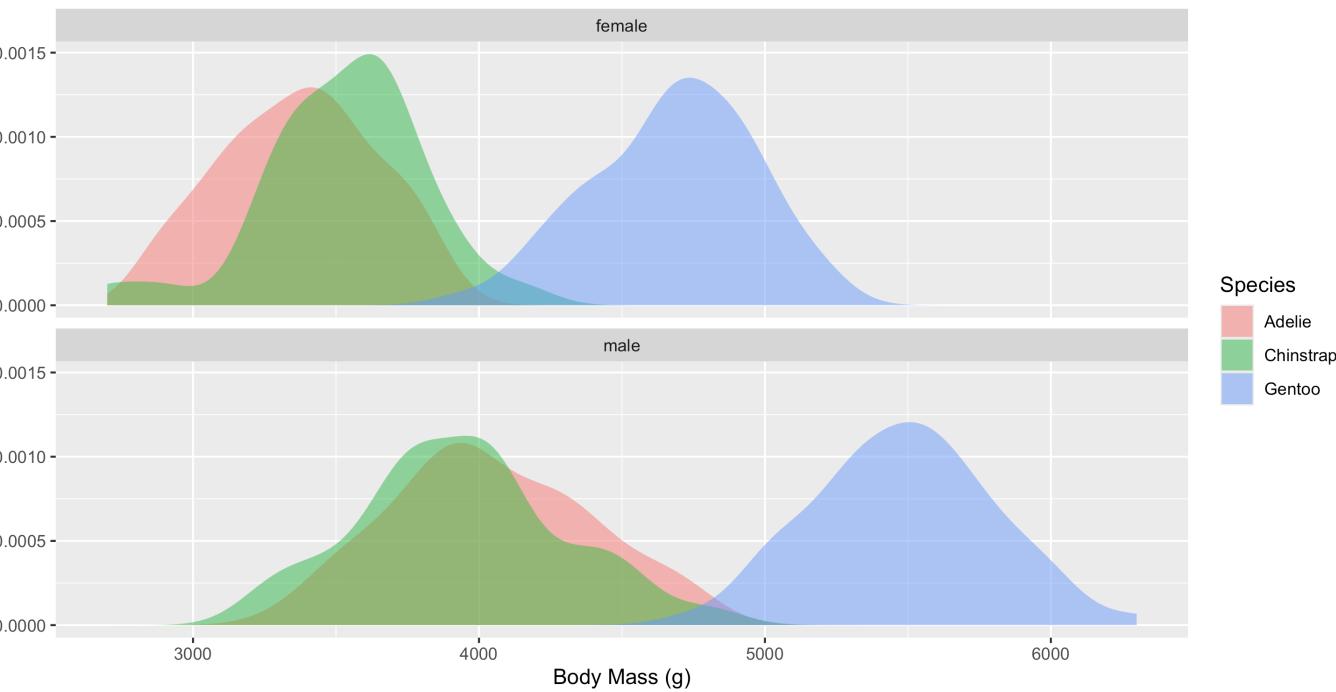


# Exercises

65

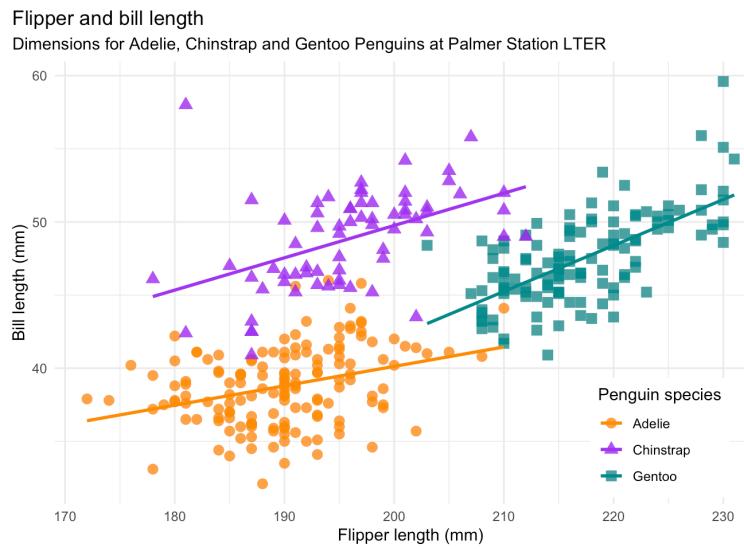
# Exercise 1

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.



See the [palmerpenguins](#) pkgdown site

67