

Lec 10 - Visualization with ggplot2

Statistical Programming

Sta 323 | Spring 2022

Dr. Colin Rundel



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

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



```
library(palmerpenguins)  
penguins
```

```
## # A tibble: 344 × 8  
##   species   island   bill_length_mm   bill_depth_mm flipper_length_mm  
##   <fct>     <fct>           <dbl>            <dbl>                <int>  
## 1 Adelie    Torgersen      39.1             18.7                 181  
## 2 Adelie    Torgersen      39.5             17.4                 186  
## 3 Adelie    Torgersen      40.3              18                  195  
## 4 Adelie    Torgersen       NA                NA                  NA  
## 5 Adelie    Torgersen      36.7             19.3                 193  
## 6 Adelie    Torgersen      39.3             20.6                 190  
## 7 Adelie    Torgersen      38.9             17.8                 181  
## 8 Adelie    Torgersen      39.2             19.6                 195  
## 9 Adelie    Torgersen      34.1             18.1                 193  
## 10 Adelie   Torgersen      42                20.2                 190  
## # ... with 334 more rows, and 3 more variables: body_mass_g <int>,  
## #   sex <fct>, year <int>
```

A basic ggplot

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

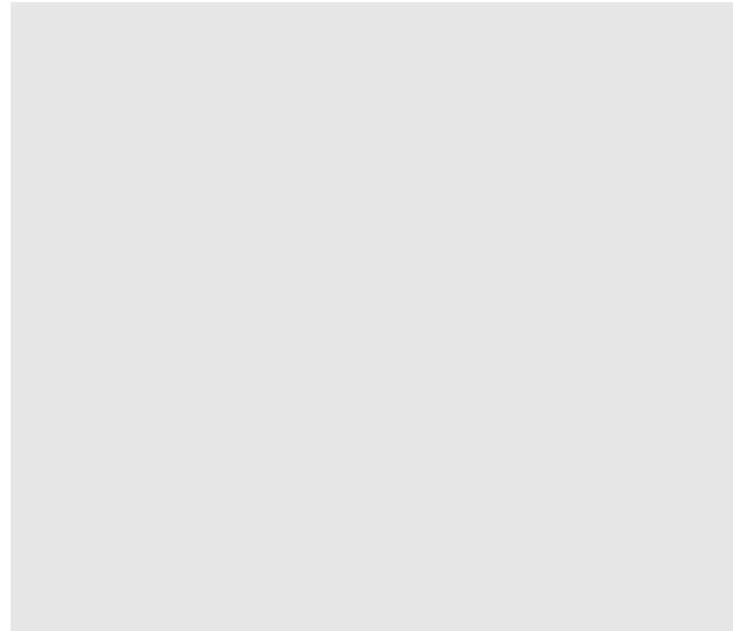
Warning: Removed 2 rows containing missing values



Text <-> Plot

Start with the penguins **data frame**

```
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")
```



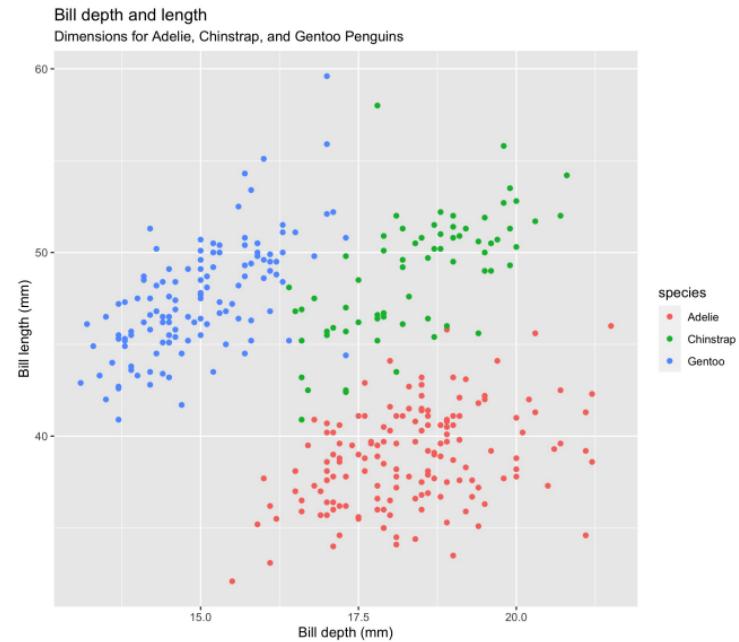
Start with the `penguins` data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", **add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins"**

```
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")  
)
```



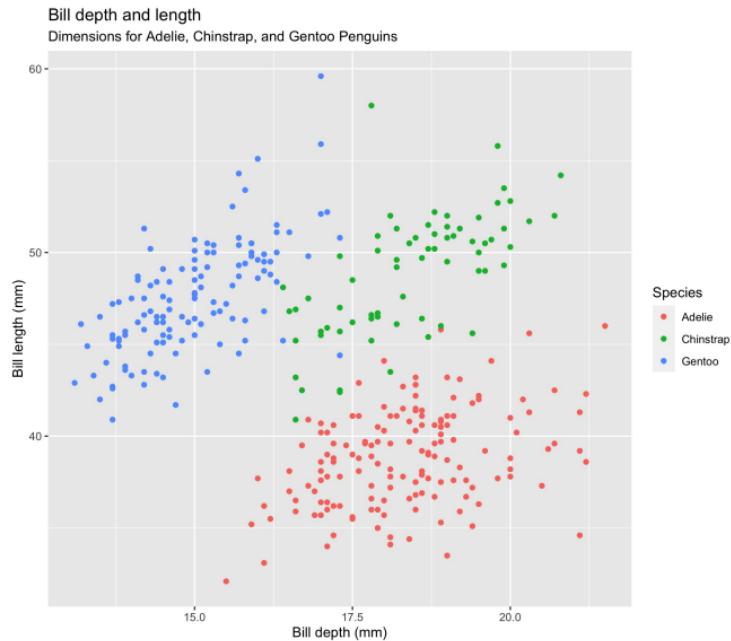
Start with the penguins data frame, map bill depth to the x-axis and map bill length to the y-axis. Represent each observation with a point and map species to the color of each point. Title the plot "Bill depth and length", add the subtitle "Dimensions for Adelie, Chinstrap, and Gentoo Penguins", **label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



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 package")
```



Argument names

Often we omit the names of first two arguments when building plots with `ggplot()`.

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

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

Note that `ggplot` and `geom_*` swap the order of the `data` and `mapping` arguments.

Aesthetics

Aesthetics options

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

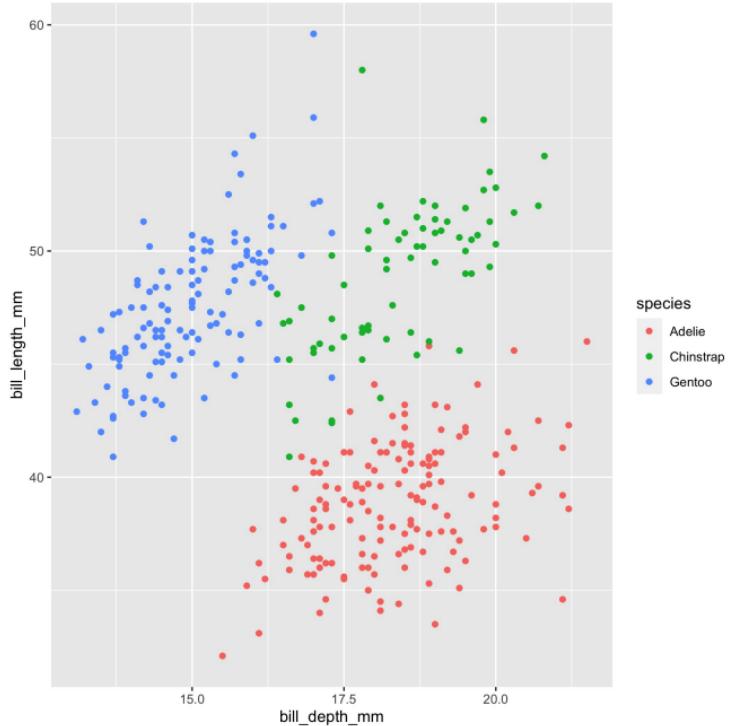
- position (x, y)
- color
- shape
- size
- alpha (transparency)

Different geometries have different aesthetics that can be used - 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 the `mapping` argument.

color

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



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  
    )  
)
```



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  
    )  
)
```



Size

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

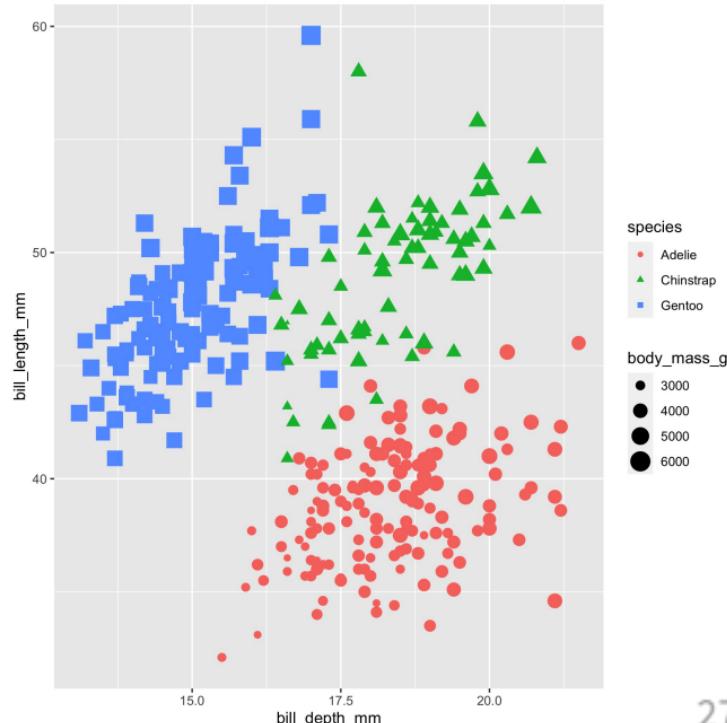
```
ggplot(  
  penguins,  
  aes(  
    x = bill_depth_mm,  
    y = bill_length_mm  
  )  
) +  
  geom_point(  
    aes(  
      color = species,  
      shape = species  
    ),  
    size = 3  
  )
```



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  
    ),  
  )
```



Alpha

```
ggplot(  
  penguins,  
  aes(  
    x = bill_depth_mm,  
    y = bill_length_mm  
)  
) +  
  geom_point(  
    aes(  
      color = species,  
      shape = species,  
      alpha = body_mass_g  
,  
      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 **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` is a **setting**.

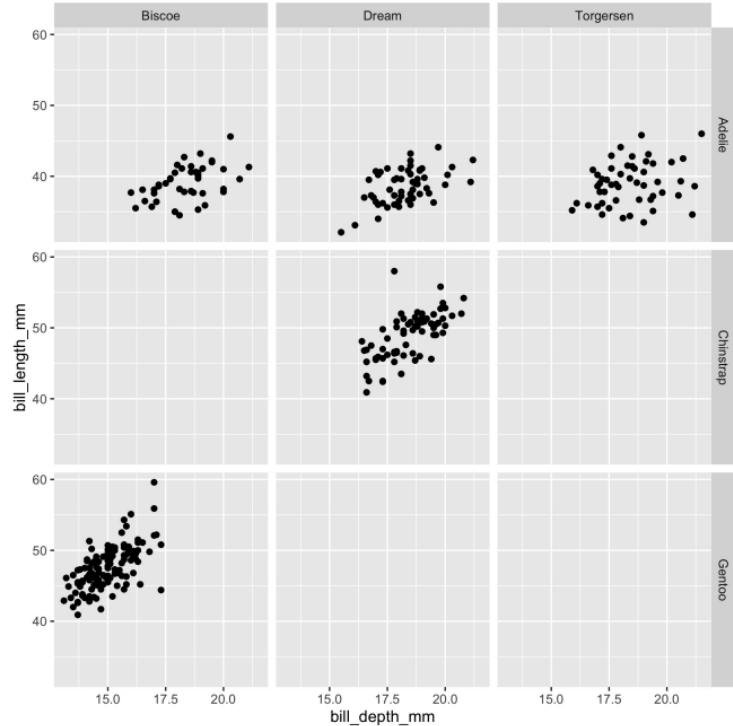
Faceting

Faceting

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

facet_grid

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



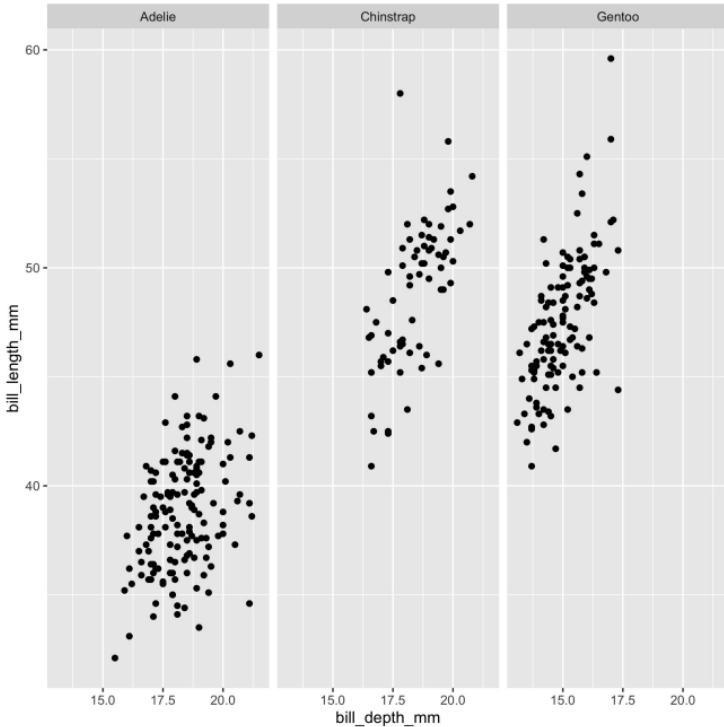
Compare with ...

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



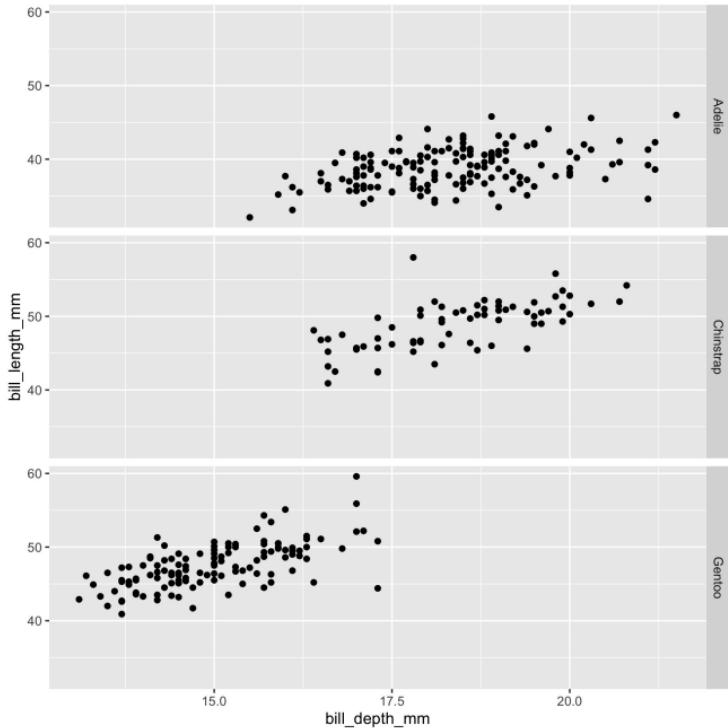
facet_grid (cols)

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



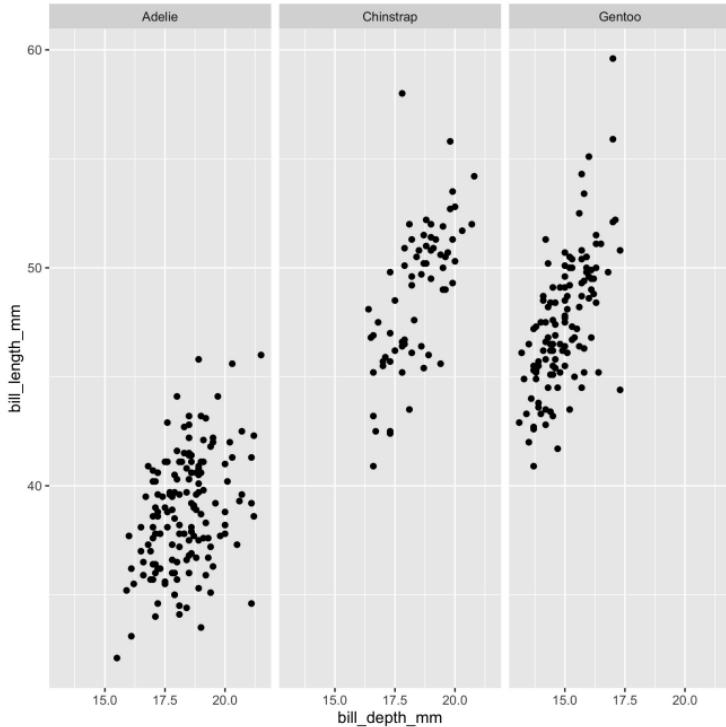
facet_grid (rows)

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



facet_wrap

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



facet_wrap

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



Faceting and color

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



Hiding redundancy

```
ggplot(  
  penguins,  
  aes(  
    x = bill_depth_mm,  
    y = bill_length_mm,  
    color = species  
  )  
) +  
  geom_point() +  
  facet_grid(species ~ sex) +  
  guides(color = "none")
```

Warning: Removed 2 rows containing missing value

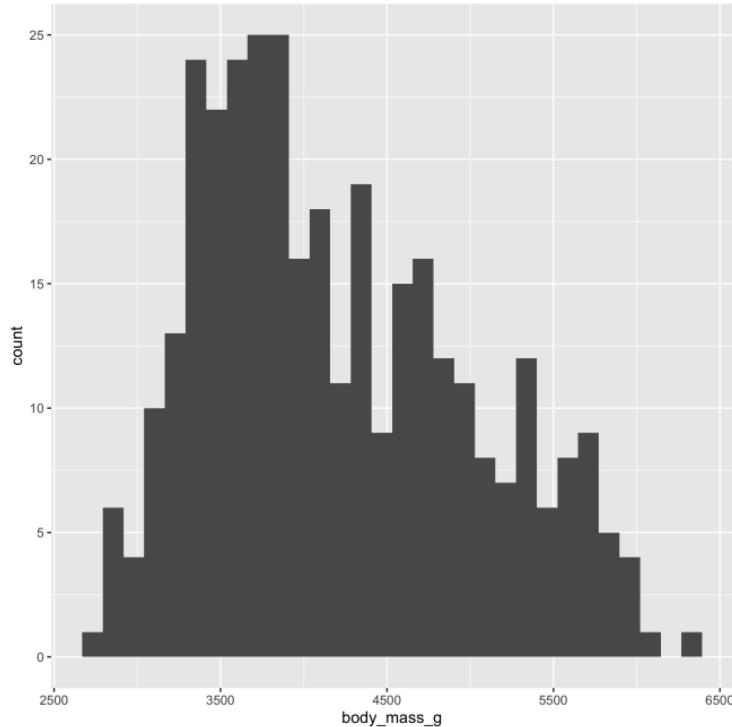
Warning: Removed 2 rows containing missing values (g



A brief plot Tour of ggplot2 plots

Histograms

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
  )) +  
  geom_histogram()  
  
## `stat_bin()` using `bins = 30`. Pick better val  
## `binwidth`.
```



Histograms - bins

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
  )) +  
  geom_histogram(bins = 50)
```



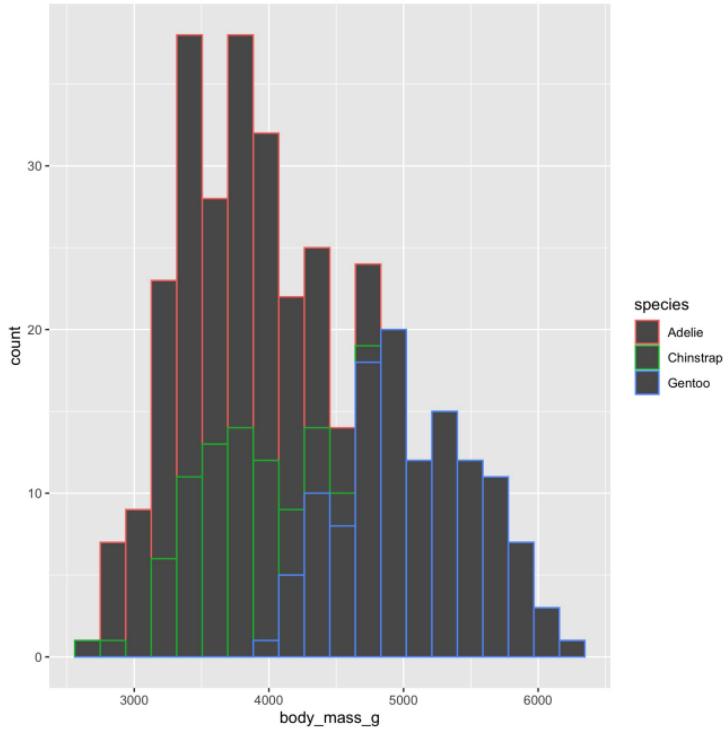
Histograms - binwidth

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
  )) +  
  geom_histogram(binwidth = 250)
```



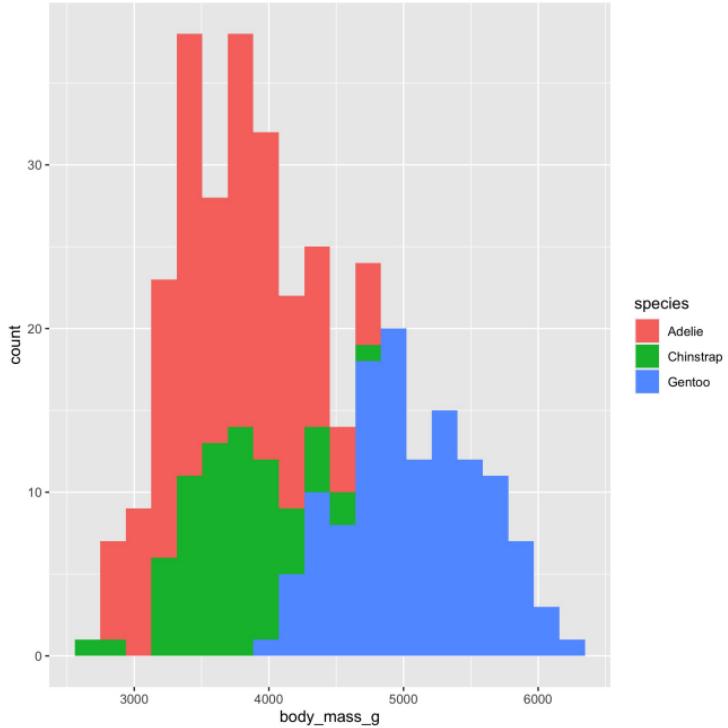
Histograms - color

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



Histograms - fill

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



Histograms - position

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    fill = species  
  )  
) +  
  geom_histogram(  
    bins = 20,  
    position = "identity",  
    alpha = 0.5  
)
```



Histograms - facets

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    fill = species  
  )  
) +  
  geom_histogram(bins = 20) +  
  facet_grid(species ~ .) +  
  guides(fill = "none")
```



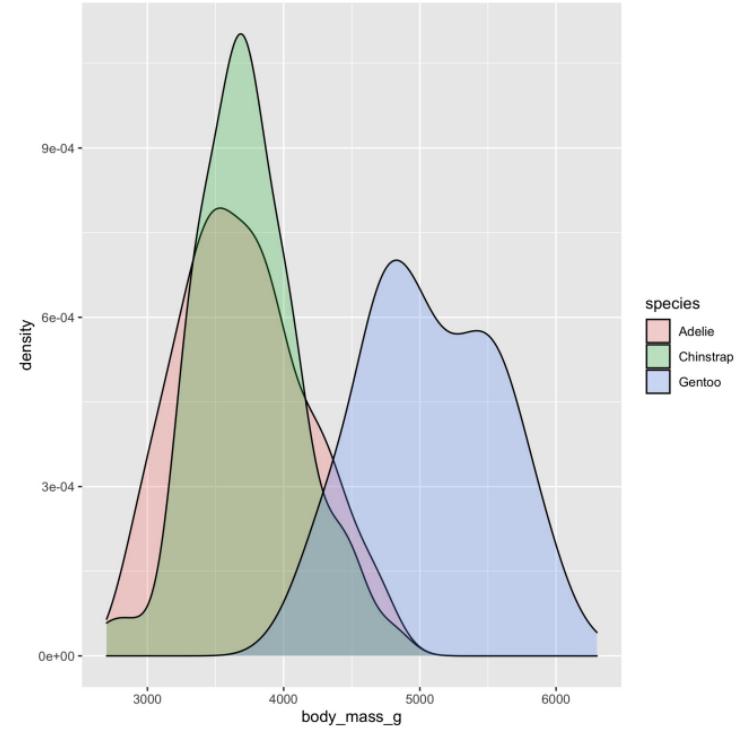
Density plot

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g  
  )  
) +  
  geom_density()
```



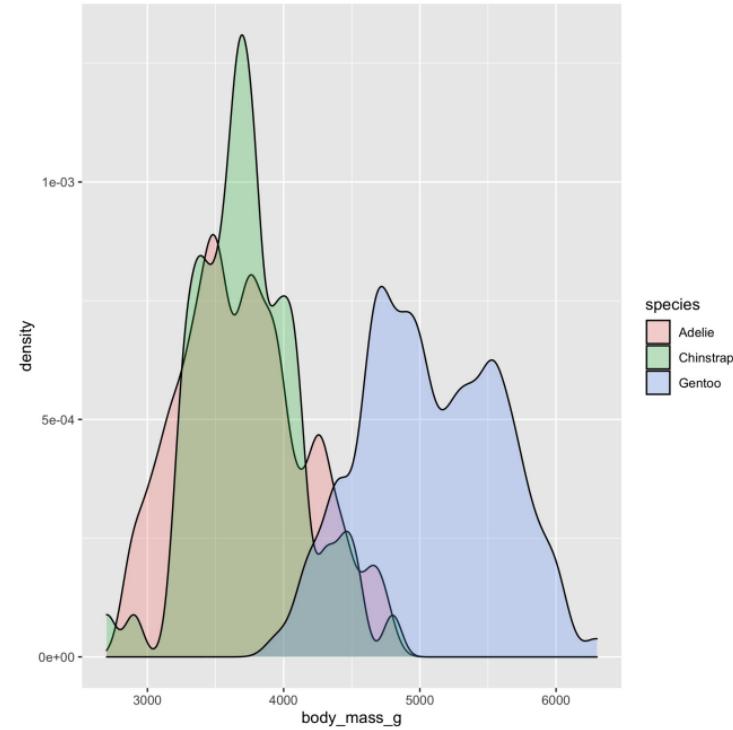
Density plot - fill

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



Density plot - adjust

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    fill = species  
  )  
) +  
  geom_density(  
    adjust = 0.5,  
    alpha = 0.25  
  )
```



Violin plot

```
ggplot(  
  penguins,  
  aes(  
    x = species,  
    y = body_mass_g,  
    fill = species  
  )  
) +  
  geom_violin()
```



Ridge plot

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    y = species,  
    fill = species  
  )  
) +  
  ggridges::geom_density_ridges(alpha = 0.5)  
  
## Picking joint bandwidth of 153
```



Ridge plot - more categories + dplyr

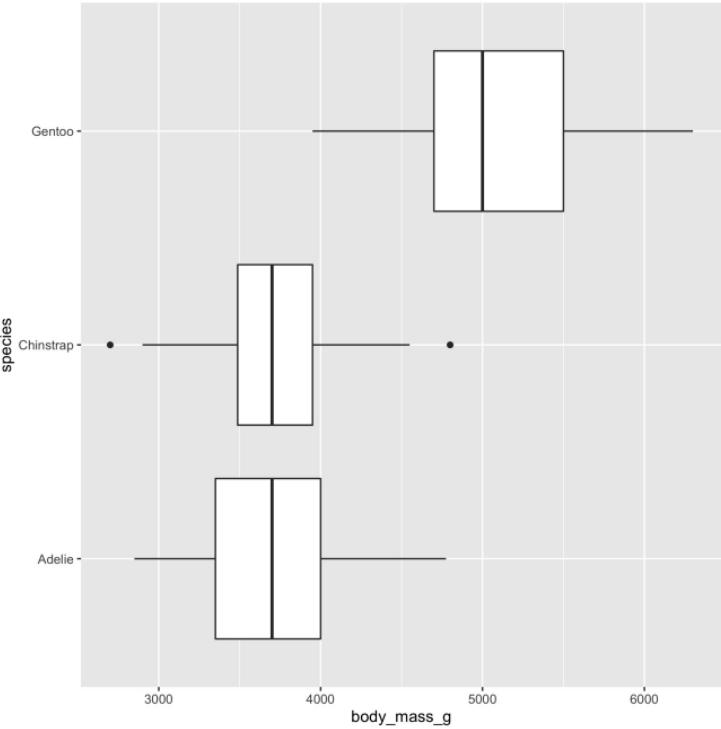
```
penguins %>%  
  mutate(  
    species_sex = paste0(species, " (", sex, ")")  
  ) %>%  
  ggplot(  
    aes(  
      x = body_mass_g,  
      y = species_sex,  
      fill = species  
    )  
  ) +  
  ggridges::geom_density_ridges(alpha = 0.5)
```

Picking joint bandwidth of 127



Box plot

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    y = species  
  )  
) +  
  geom_boxplot()
```



Box plot - coord_flip

```
ggplot(  
  penguins,  
  aes(  
    x = body_mass_g,  
    y = species  
  )  
) +  
  geom_boxplot() +  
  coord_flip()
```



Box plot - swap coords

```
ggplot(  
  penguins,  
  aes(  
    x = species,  
    y = body_mass_g  
  )  
) +  
  geom_boxplot()
```



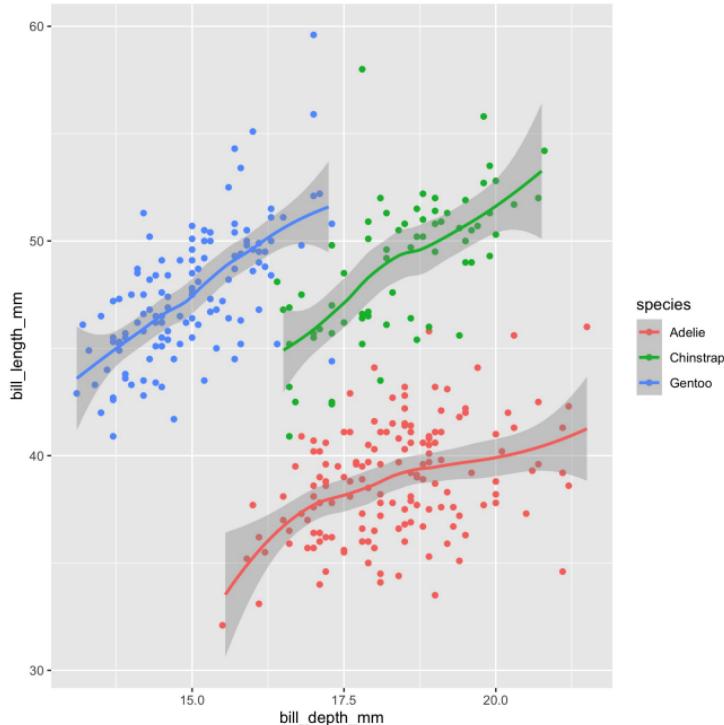
Scatter plot

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



Scatter plot - geom_smooth

```
ggplot(  
  penguins,  
  aes(  
    x = bill_depth_mm,  
    y = bill_length_mm,  
    color = species  
)  
) +  
  geom_point() +  
  geom_smooth(  
    fullrange = TRUE  
)  
  
## `geom_smooth()` using method = 'loess' and form
```



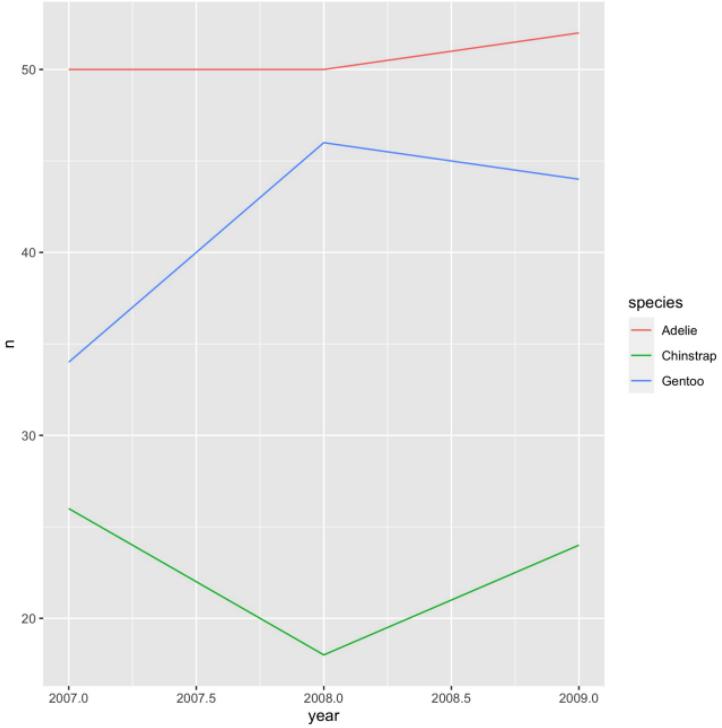
Scatter plot - geom_smooth w/ lm

```
ggplot(  
  penguins,  
  aes(  
    x = bill_depth_mm,  
    y = bill_length_mm,  
    color = species  
)  
) +  
  geom_point() +  
  geom_smooth(  
    method = "lm",  
    se = FALSE,  
    fullrange = TRUE  
)  
  
## `geom_smooth()` using formula 'y ~ x'
```



Line plot

```
penguins %>%
  count(species, year) %>%
  ggplot(
    aes(
      x = year,
      y = n,
      color = species,
      group = species
    )
  ) +
  geom_line()
```



Line plot - with points

```
penguins %>%
  count(species, year) %>%
  ggplot(
    aes(
      x = year,
      y = n,
      color = species,
      group = species
    )
  ) +
  geom_line() +
  geom_point()
```



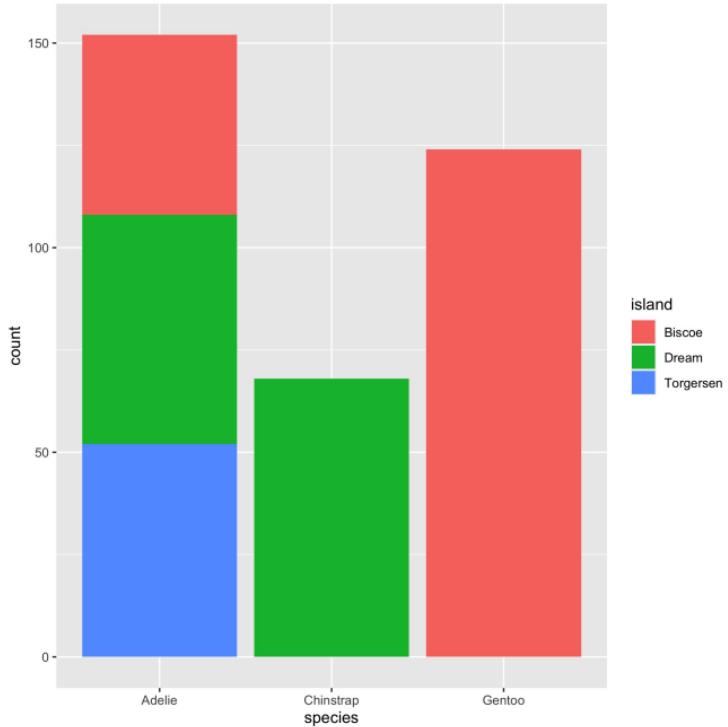
Bar plot

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



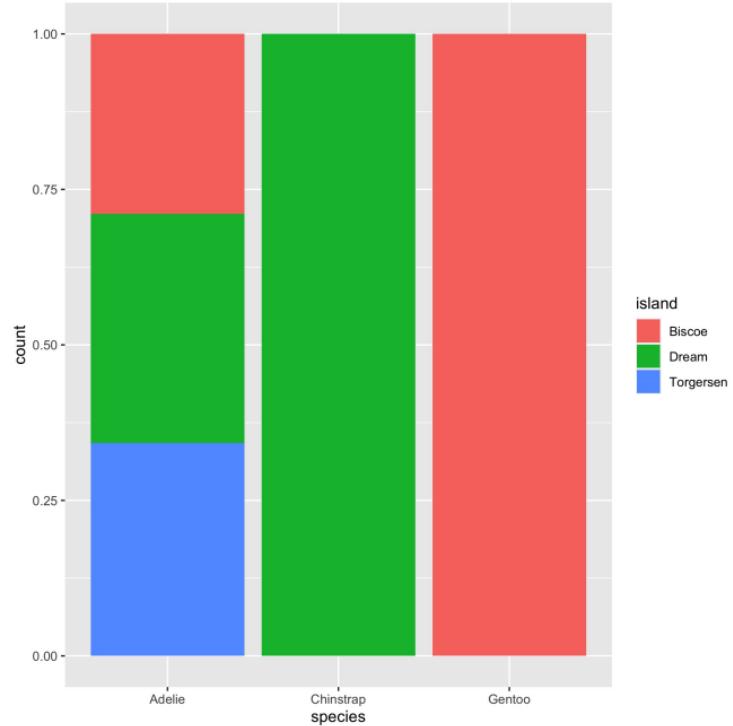
Stacked bar plot

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



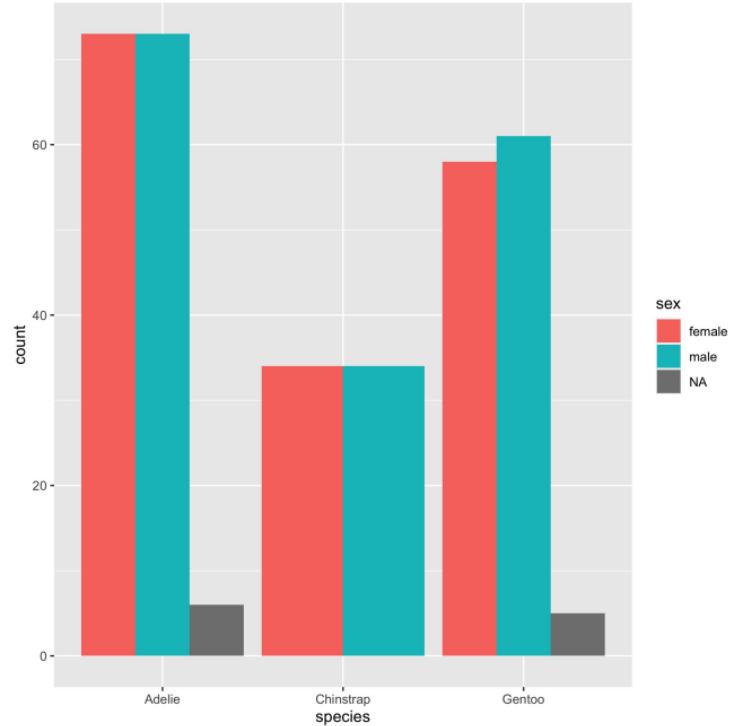
Stacked relative frequency bar plot

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



Dodged bar plot

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



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`.

