Data visualisations Markdown

Julia Jagger

23rd June 2025

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com or alternatively click here.

Required packages

Note that I have turned of the output to the packages as I don't see the value in viewing that part.

```
install.packages("tidyverse")
library(tidyverse)
install.packages(("palmerpenguins"))
library(palmerpenguins)
```

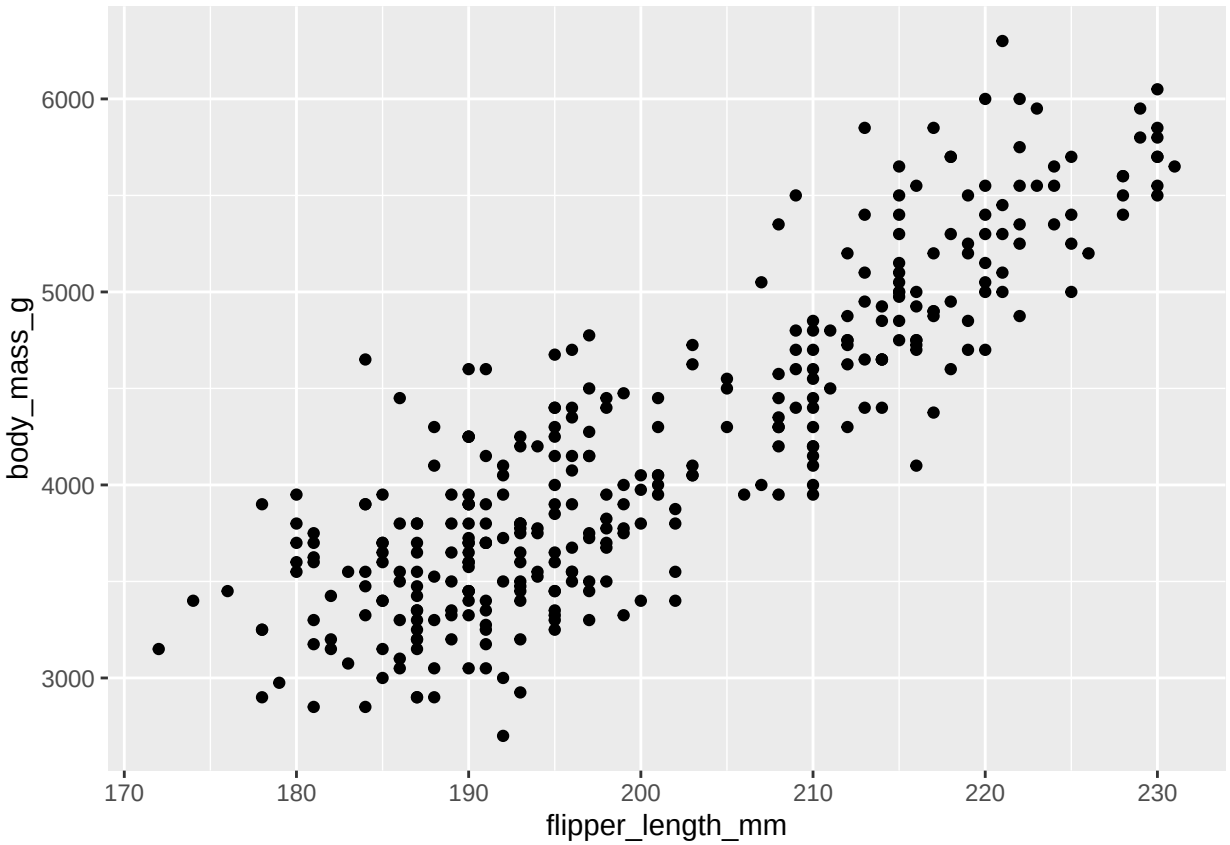
Plotting options

When plotting with ggplot2, the line will always begin with ggplot(data=<>). This tells ggplot2 which data frame to use. If you run it as is, it'll just bring up an empty plot:

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

Next plus sign adds layers to a plot. Firstly we add geom_point to tell R to use points to represent the data (scatter plot). Plus sign must be at the end of a line if you wish to break it up in this way. The + represents a new layer in the plot.

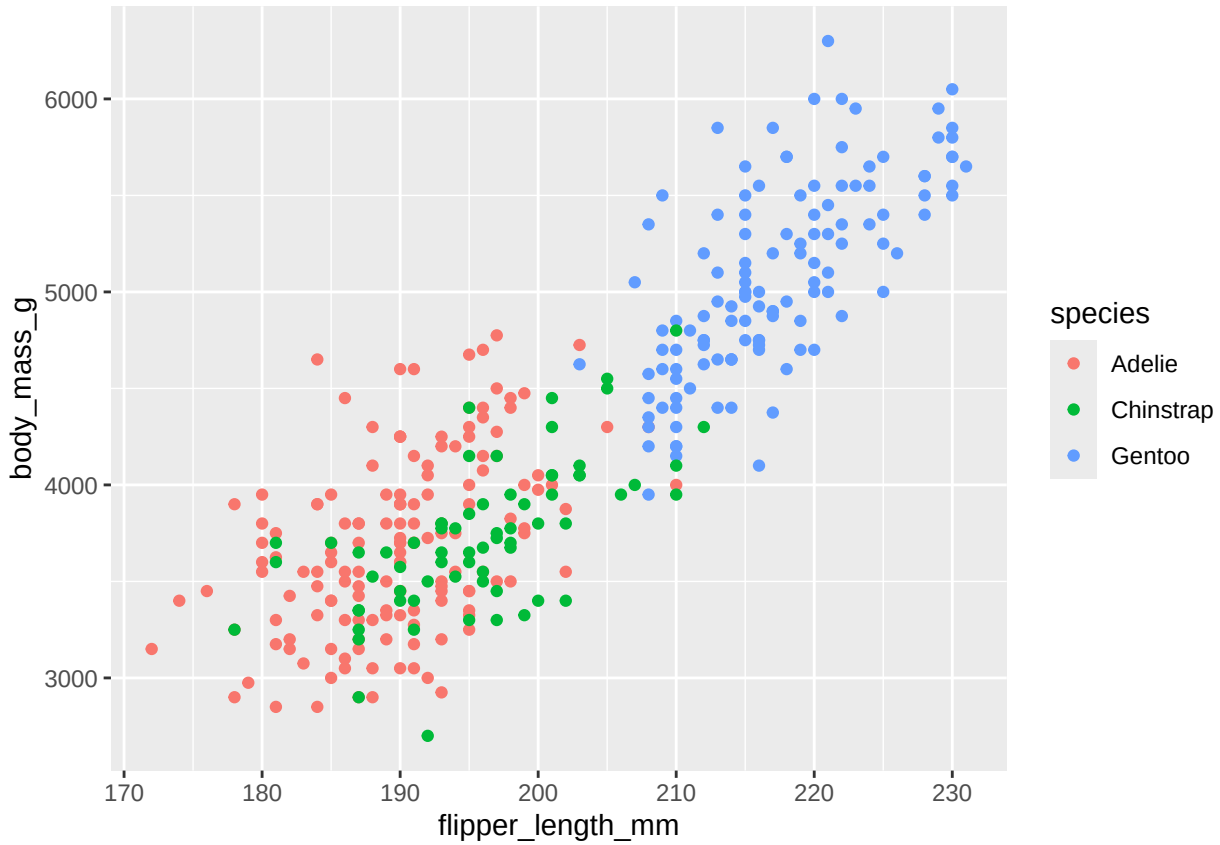
```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g))
```



Mapping means matching up a specific variable in your dataset with a specific aesthetic.

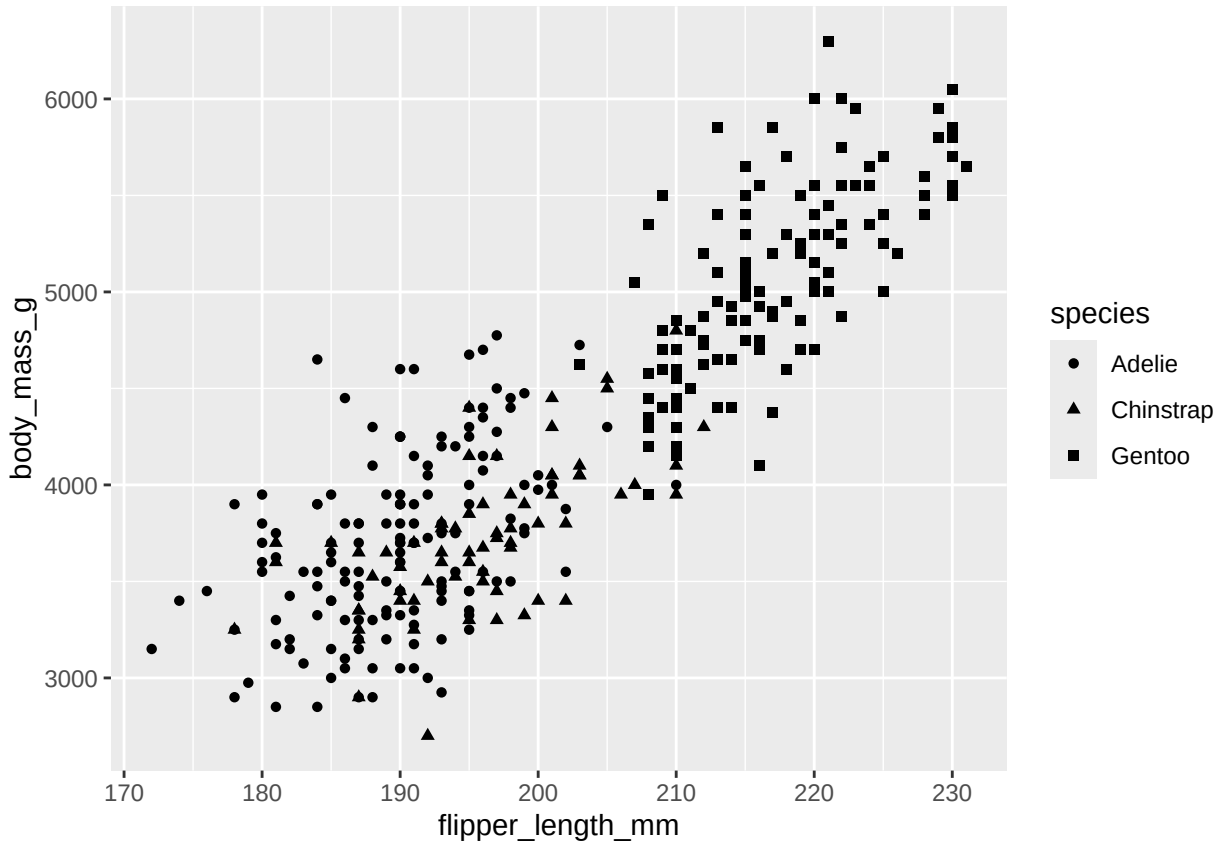
Refine mapping to add a third variable (species) in different colours:

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, colour=species))
```



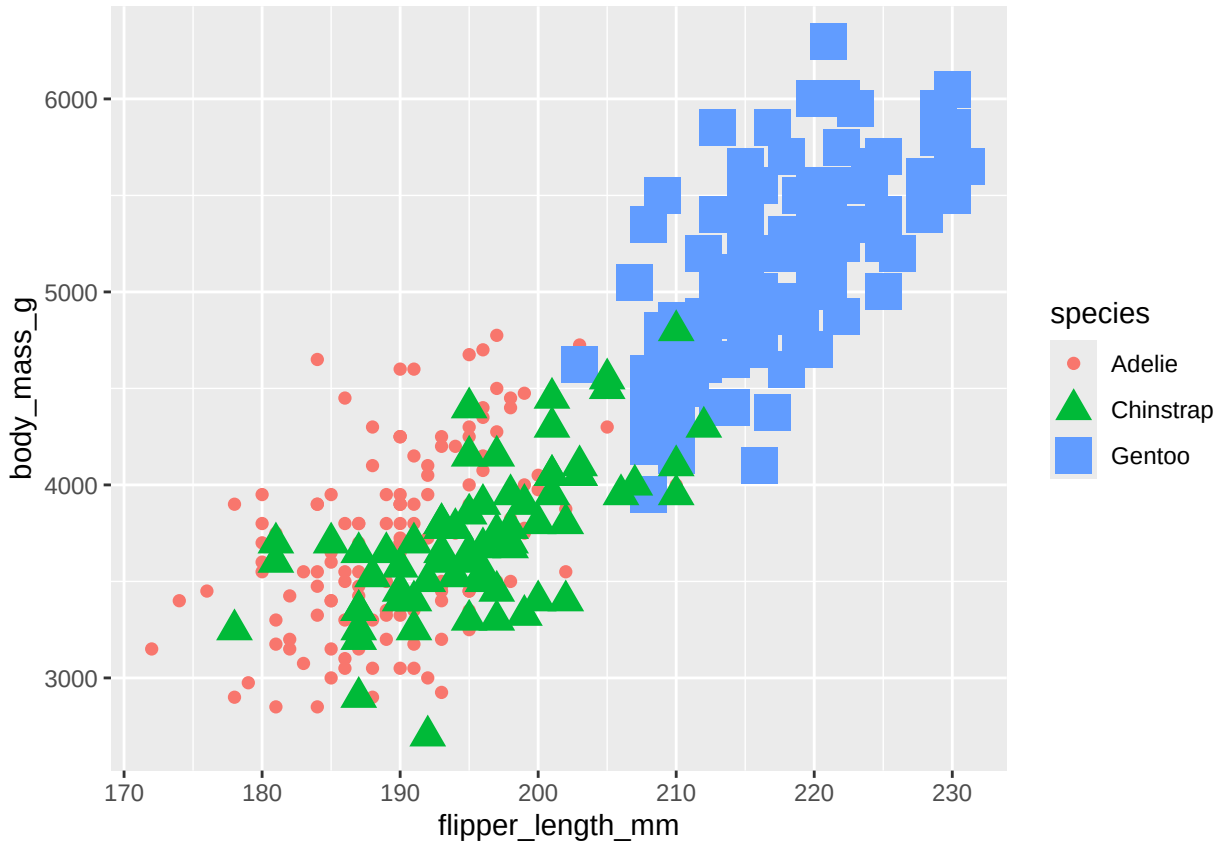
Refine mapping further to add a third variable (species) in different shapes:

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, shape=species))
```



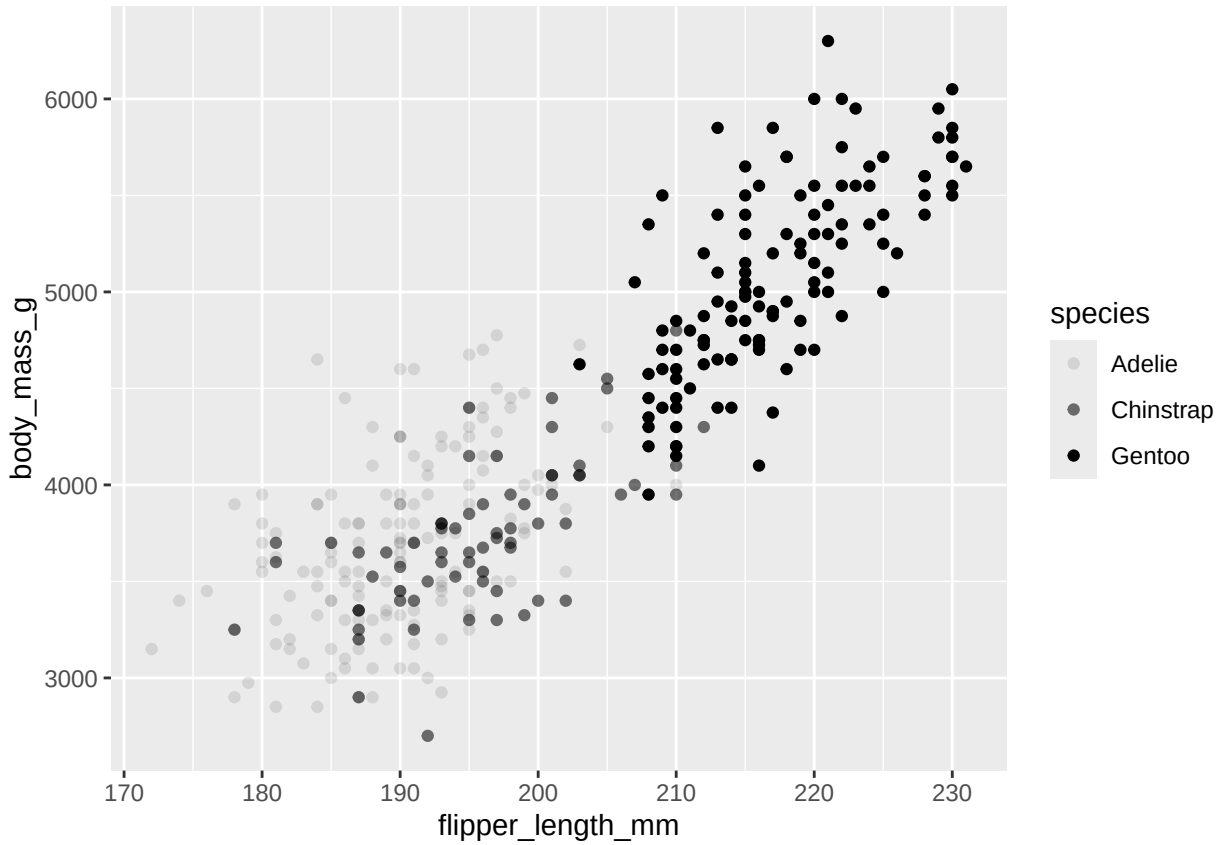
Refine mapping to add a third variable (species) in different shapes AND colours AND sizes:

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, shape=species, colour=species, size=specie
```



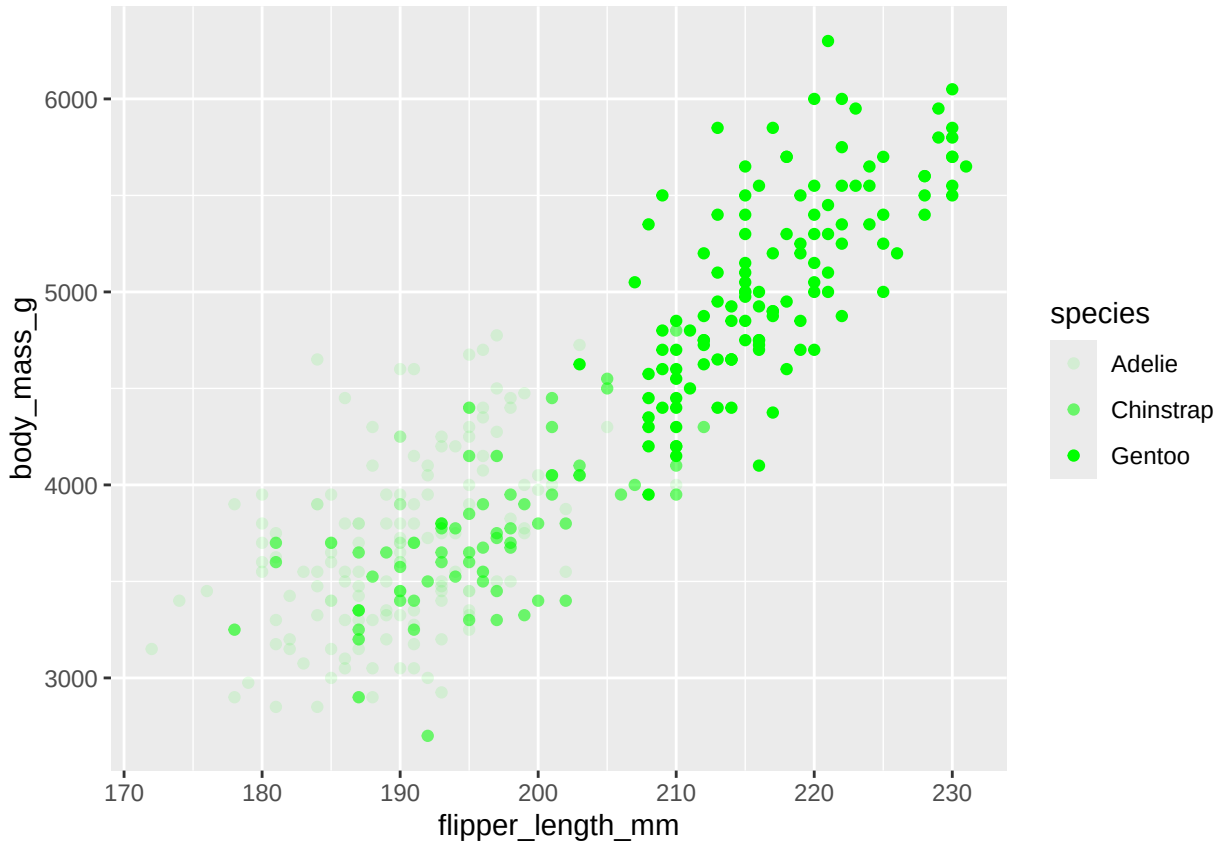
Transparency of the points can be edited by using alpha aesthetic. This works well when you've got a dense plot with lots of data points.

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, alpha=species))
```



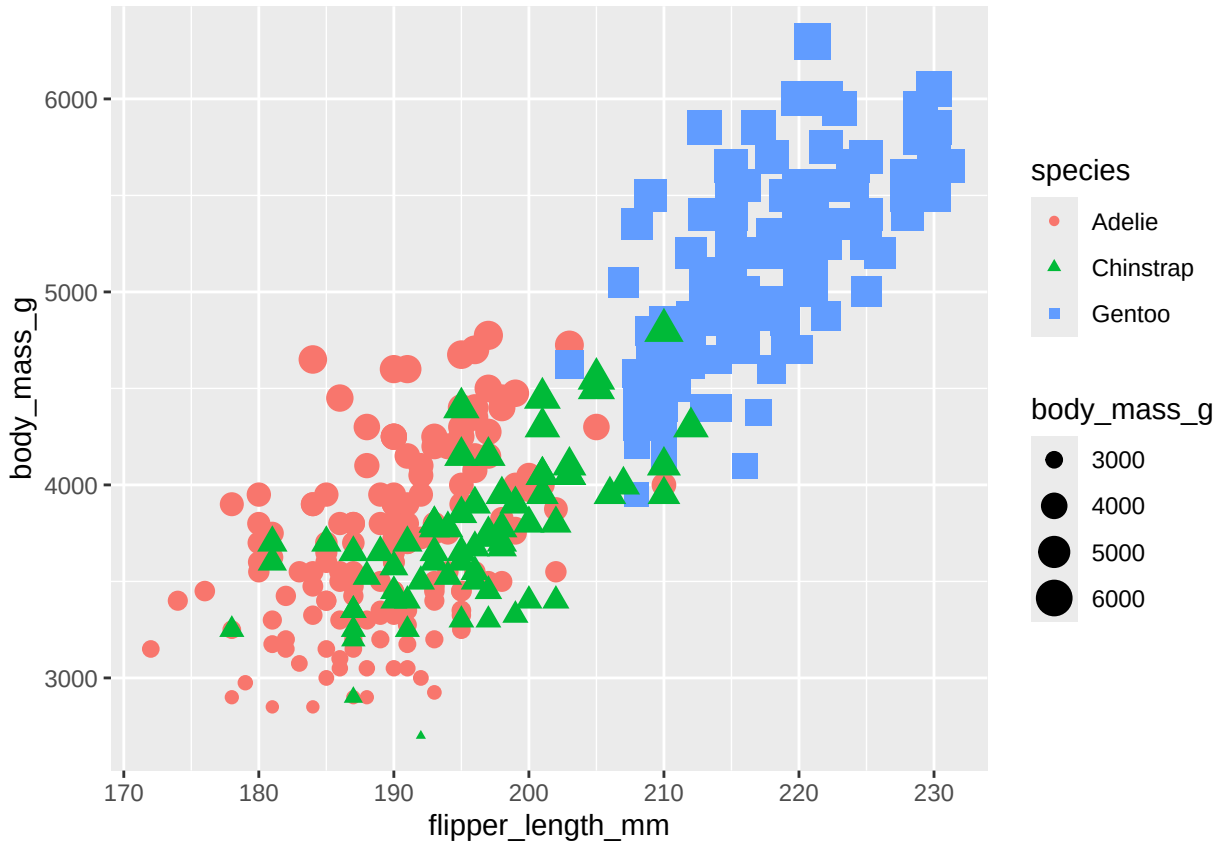
Writing code outside of the aes function changes the appearance of the plot overall (and ignoring specific variables).

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, alpha=species), colour="green")
```



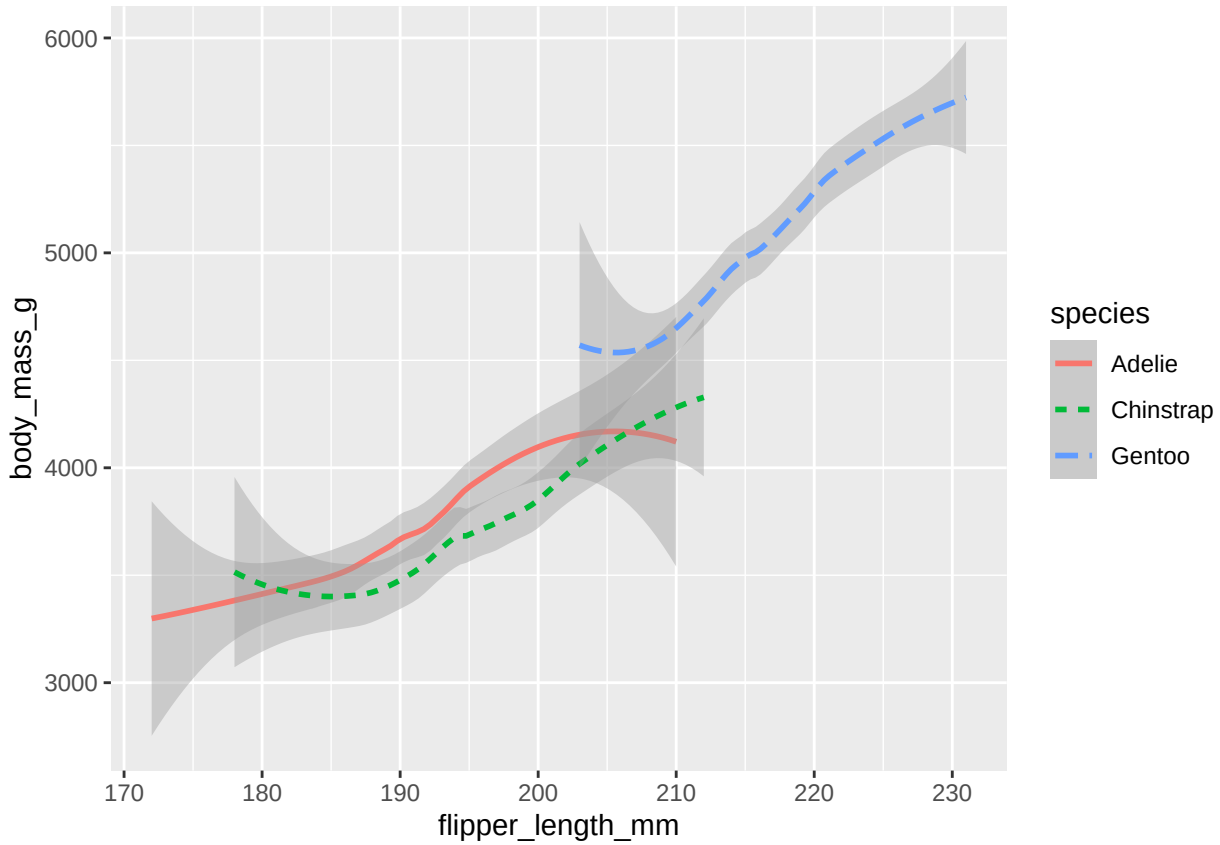
Three aesthetic attributes in ggplot2 are colour, size, and shape

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, colour=species, size=body_mass_g, shape=sp
```



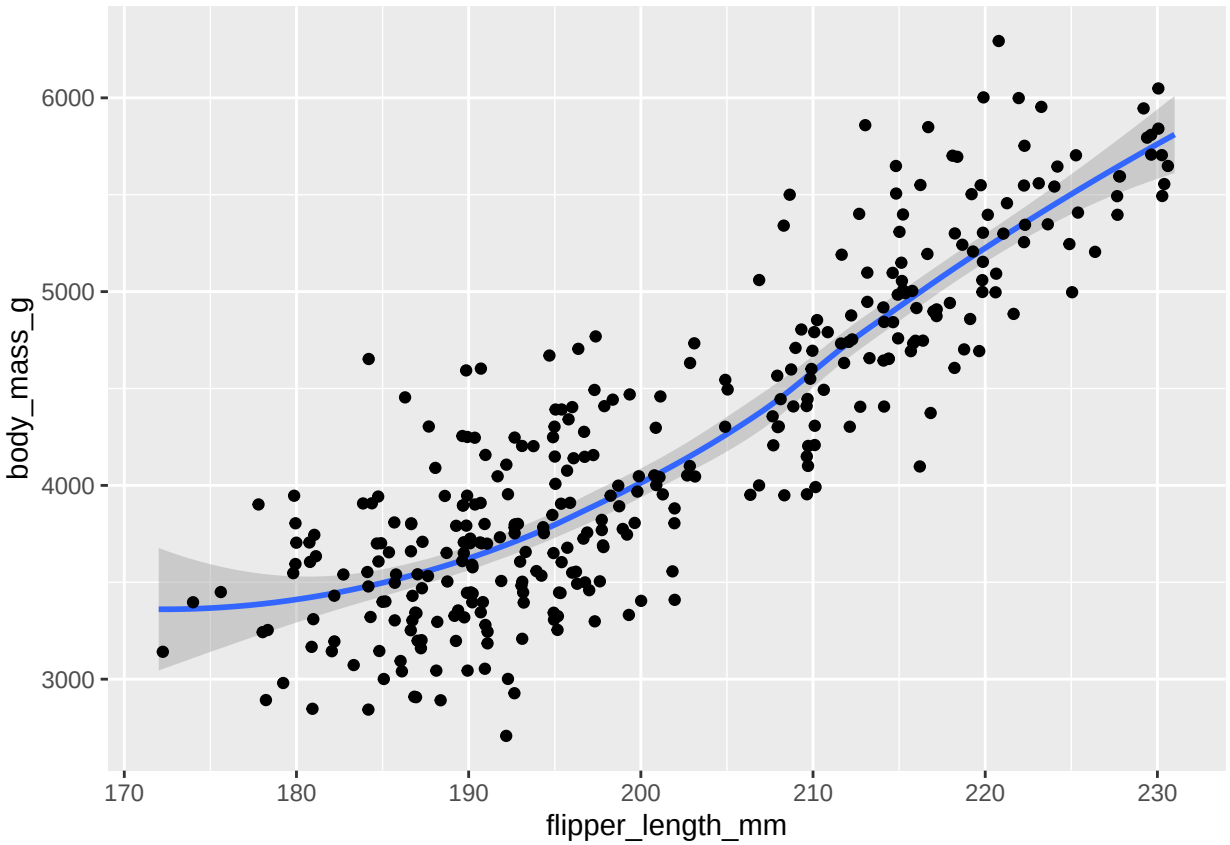
Looking at different geoms - the geometrical object used to reprensent your data. geom_point, geom_bar, geom_line etc

```
ggplot(data=penguins)+
  geom_smooth(mapping=aes(x=flipper_length_mm, y=body_mass_g, colour=species, linetype=species))
```



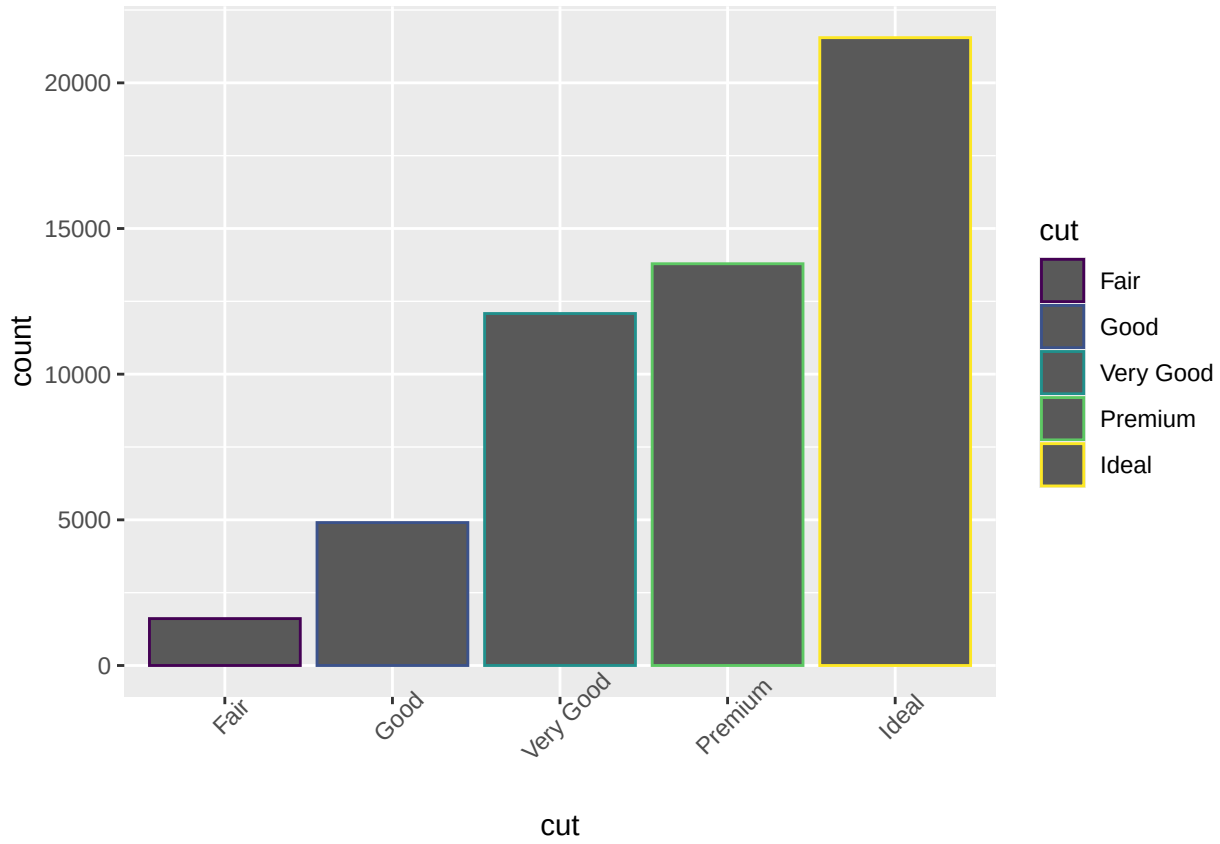
geom_jitter function creates a scatter plot and then adds a small amount of random noise to each point in the lot. It helps deal with over-plotting (when data points in a plot overlap with each other).

```
ggplot(data=penguins)+
  geom_smooth(mapping=aes(x=flipper_length_mm, y=body_mass_g))+
  geom_jitter(mapping=aes(x=flipper_length_mm, y=body_mass_g))
```



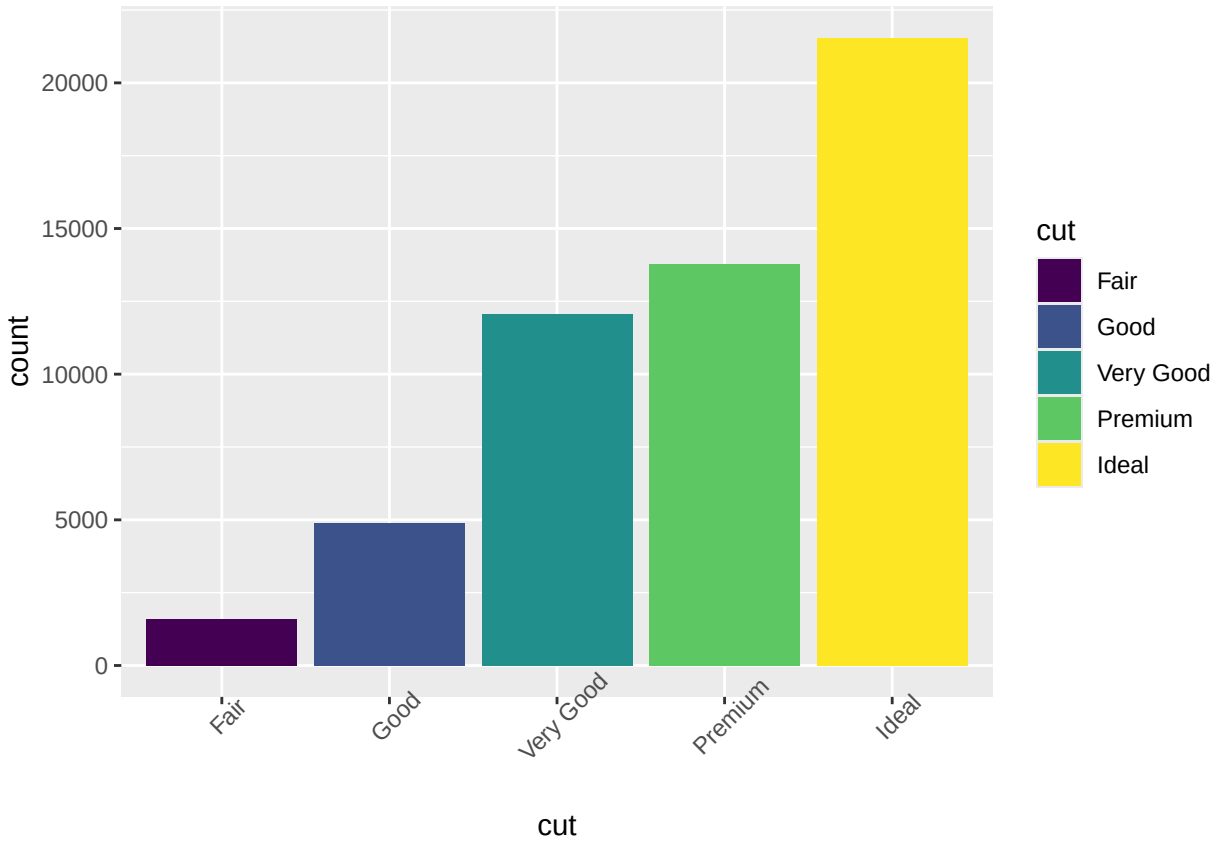
Working with the diamond dataset. Here the outline of each bar has been coloured. In addition, the labels at the bottom have been angled. If the text was longer, this angling makes them easier to read:

```
ggplot(data=diamonds)+
  geom_bar(mapping=aes(x=cut, colour=cut))+
  theme(axis.text.x = element_text(angle = 45))
```



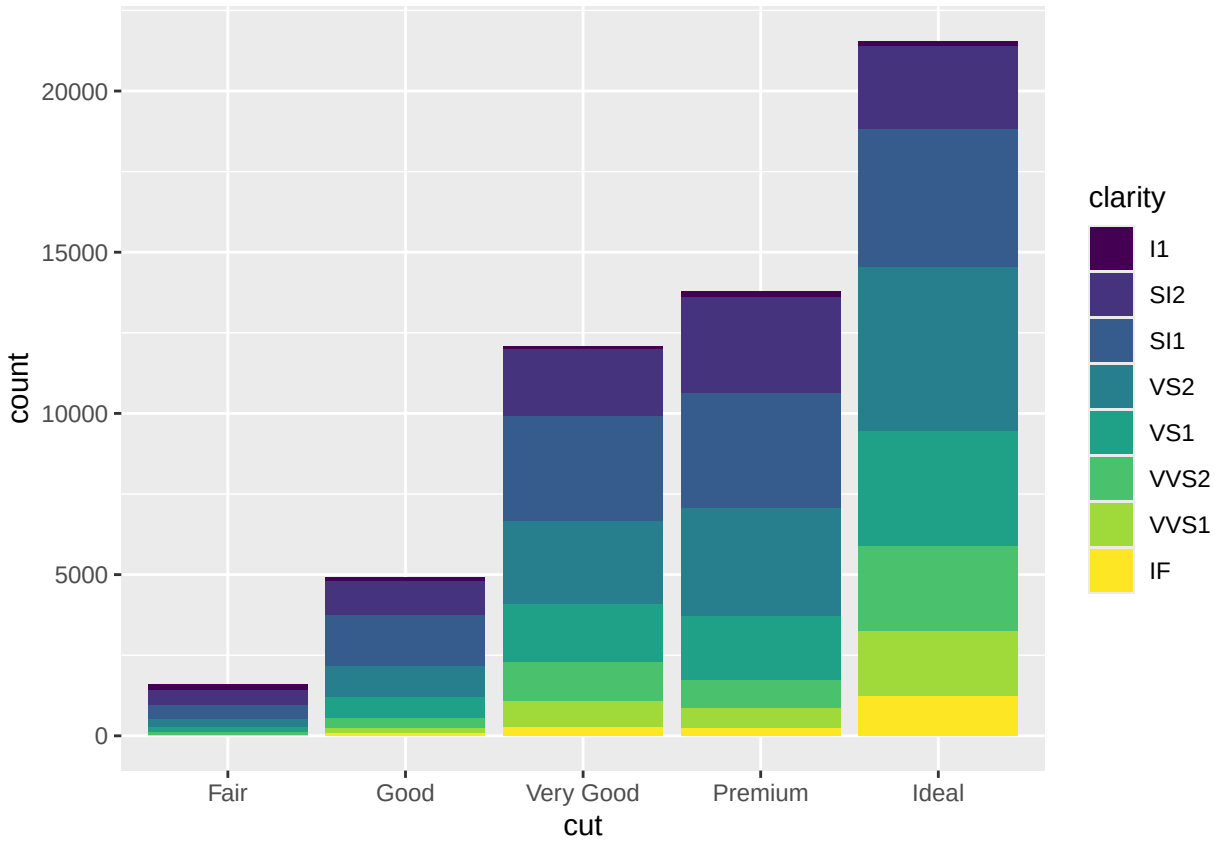
Using the fill function colours the whole bar, not just the outline:

```
ggplot(data=diamonds)+
  geom_bar(mapping=aes(x=cut, fill=cut))+
  theme(axis.text.x = element_text(angle = 45))
```



If fill is mapped to a new variable, geom_bar displays a stacked bar chart:

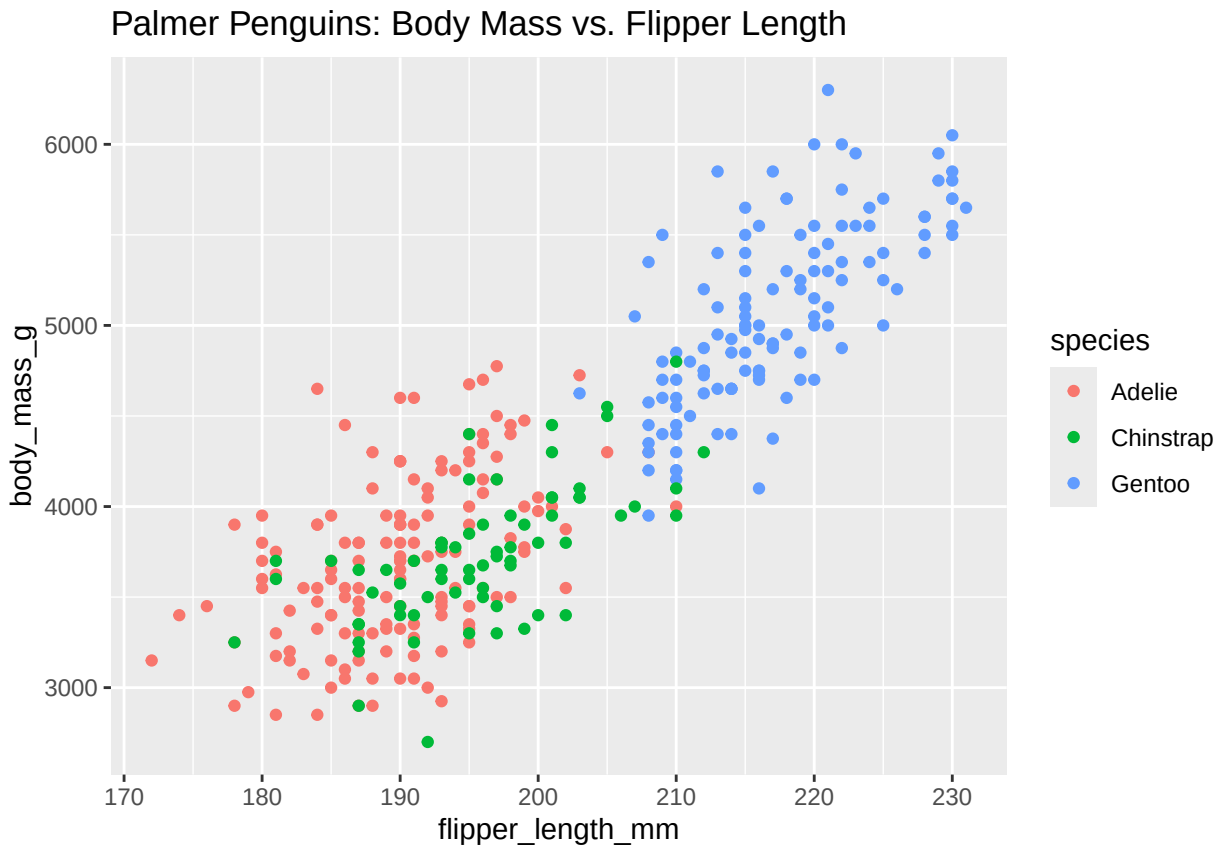
```
ggplot(data=diamonds)+
  geom_bar(mapping=aes(x=cut, fill=clarity))
```



Labels and annotations

Labels are information that is shared outside of the plot range, annotations are shared inside it.

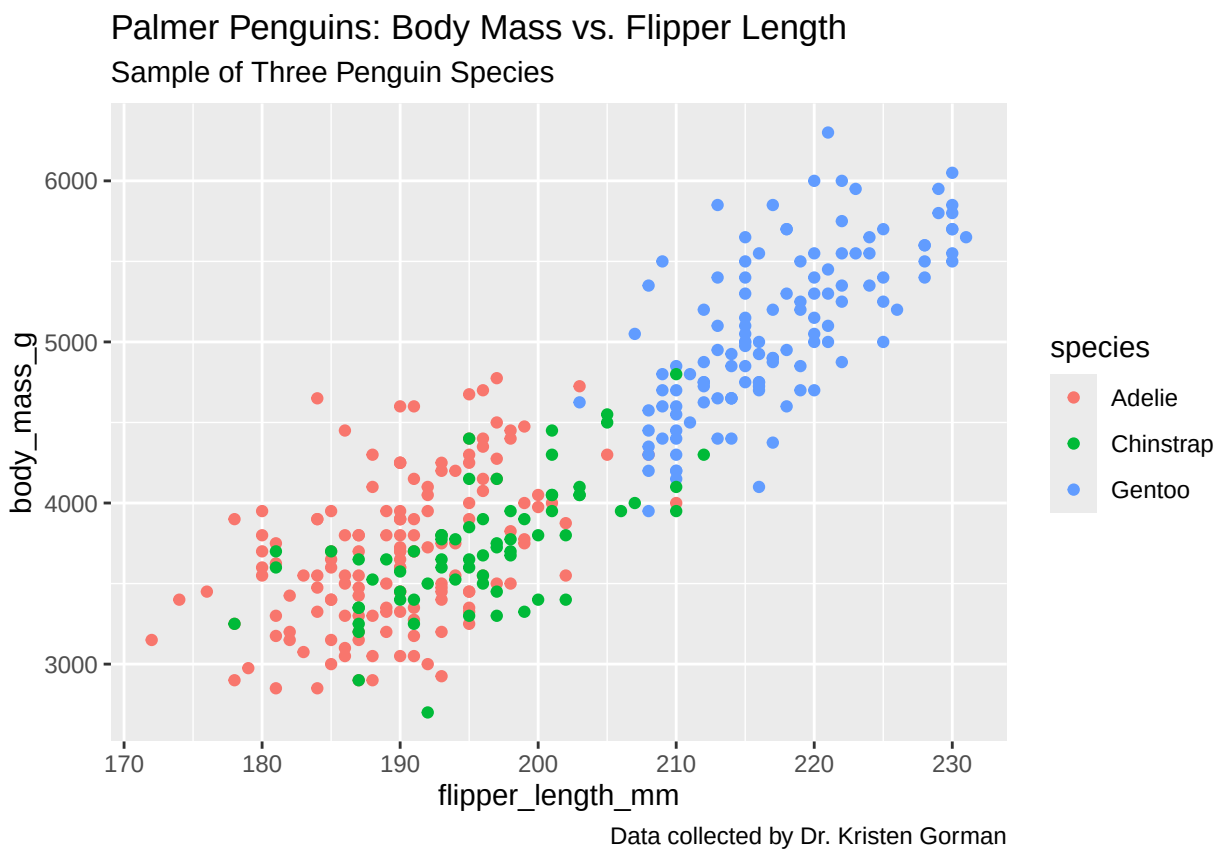
```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,colour=species))+
  labs(title="Palmer Penguins: Body Mass vs. Flipper Length")
```



Note that title is automatically at the top of the plot.

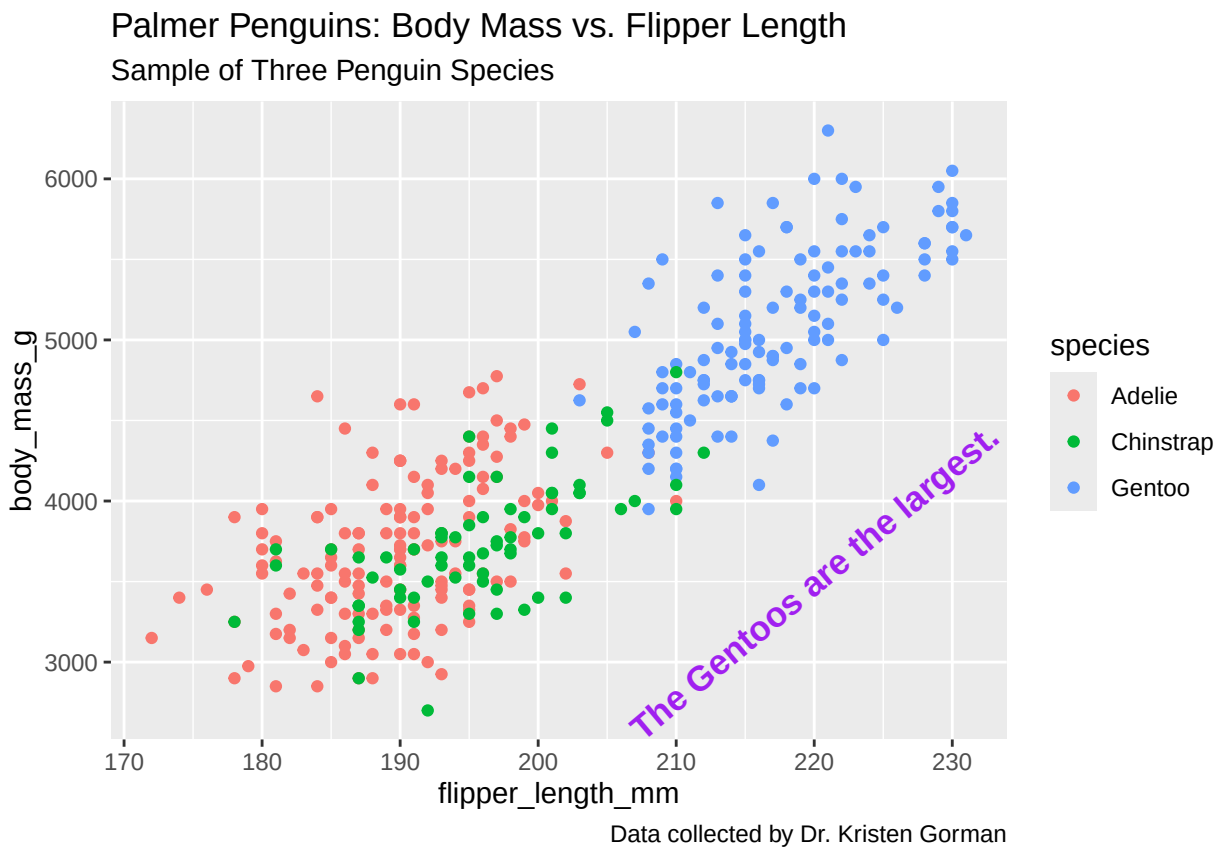
As well as title, you can have subtitles and captions whose default positions are shown below

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,colour=species))+
  labs(title="Palmer Penguins: Body Mass vs. Flipper Length",
       subtitle="Sample of Three Penguin Species",
       caption="Data collected by Dr. Kristen Gorman")
```



Finally, you can angle and format text as required:

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,colour=species))+
  labs(title="Palmer Penguins: Body Mass vs. Flipper Length",
       subtitle="Sample of Three Penguin Species",
       caption="Data collected by Dr. Kristen Gorman")+
  annotate("text",x=220,y=3500,label="The Gentoos are the largest.",
           colour="purple",
           fontface="bold", size=4.5, angle=40)
```

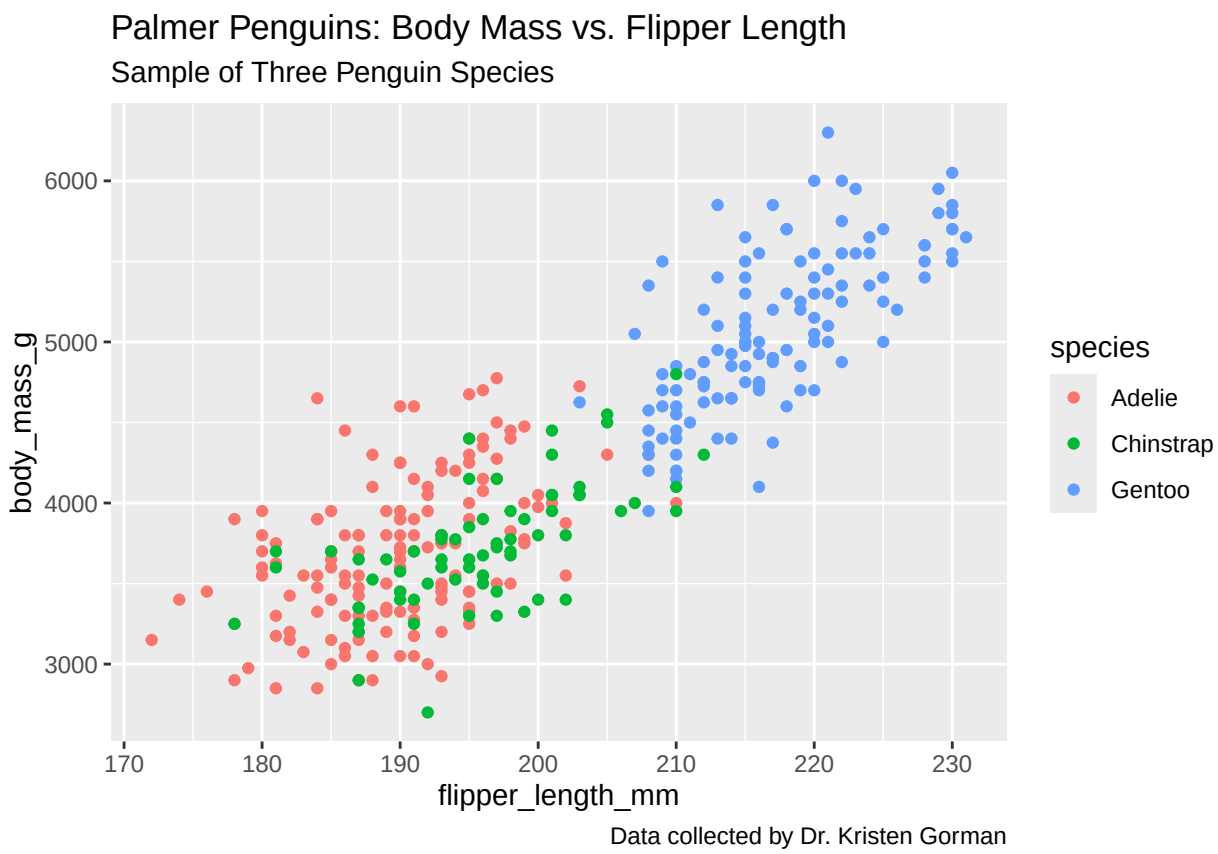


Storing plot as a variable.

Assign a plot to a variable in the usual way (with <-). To view the plot after this, simply type the name of the variable (in this case p).

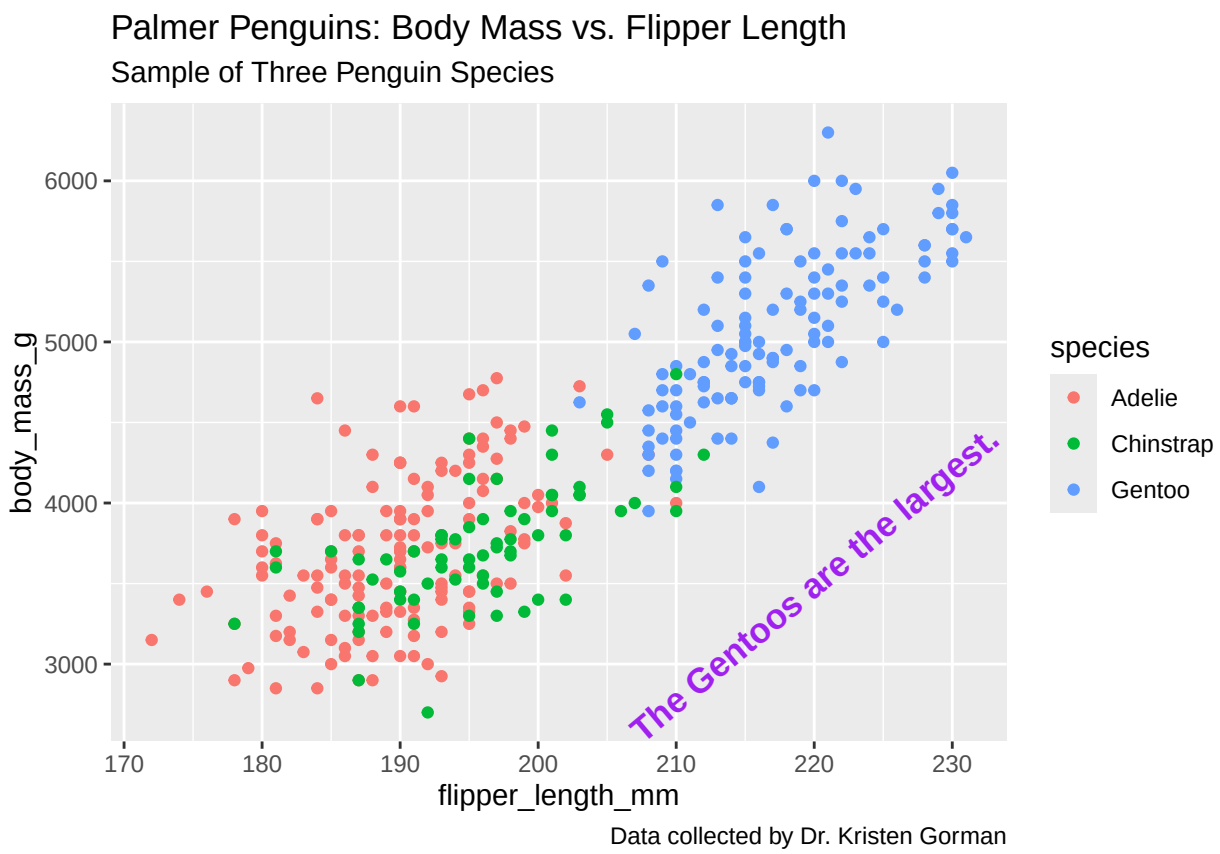
```
p <- ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm,y=body_mass_g,colour=species))+
  labs(title="Palmer Penguins: Body Mass vs. Flipper Length",
       subtitle="Sample of Three Penguin Species",
       caption="Data collected by Dr. Kristen Gorman")
```

```
p
```



Once variable is defined it can be annotated in the following way:

```
p+annotate("text",x=220,y=3500,label="The Gentoos are the largest.",
           colour="purple",
           fontface="bold", size=4.5, angle=40)
```

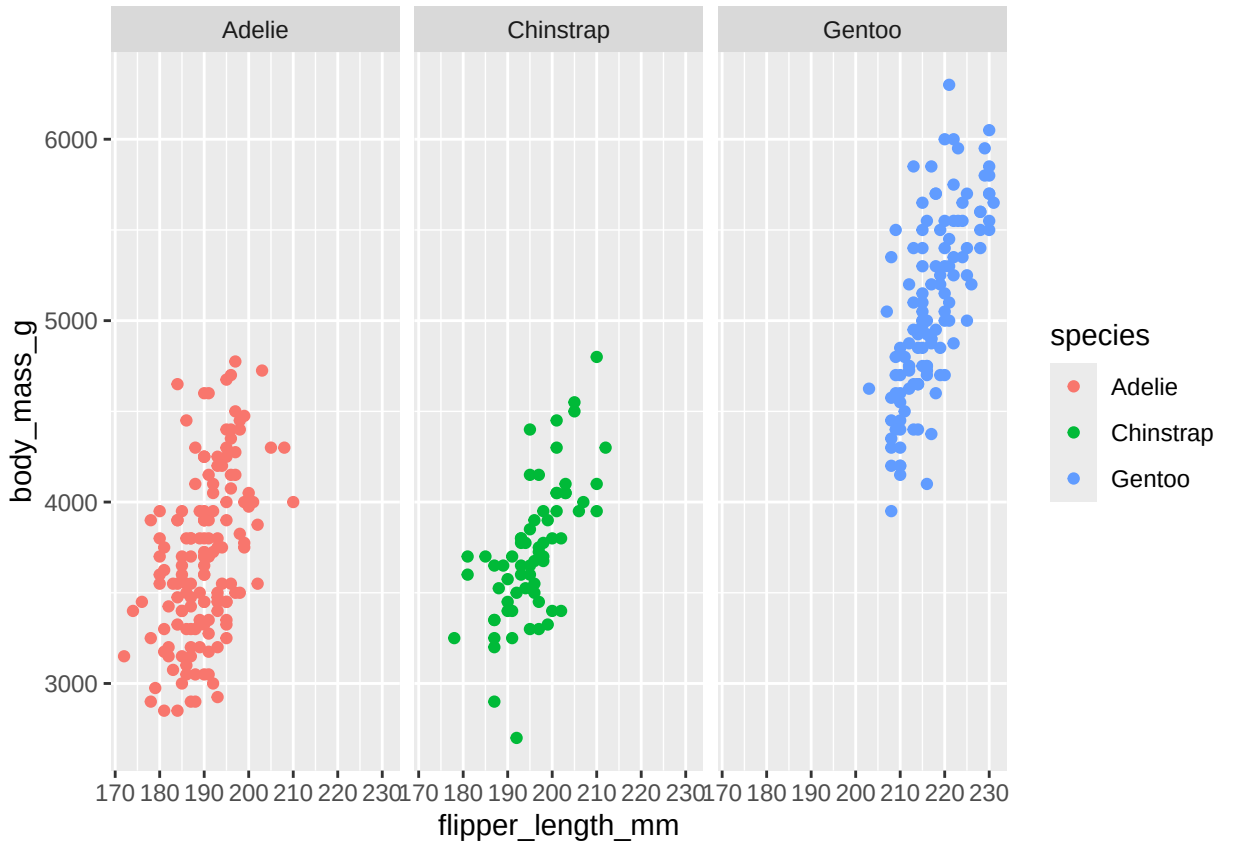


Facets

facet_wrap and facet_grid

Facets show different sides of your data by placing each subset on its own plot. Faceting can help you discover new patterns in your data and focus on relationships between different variables.

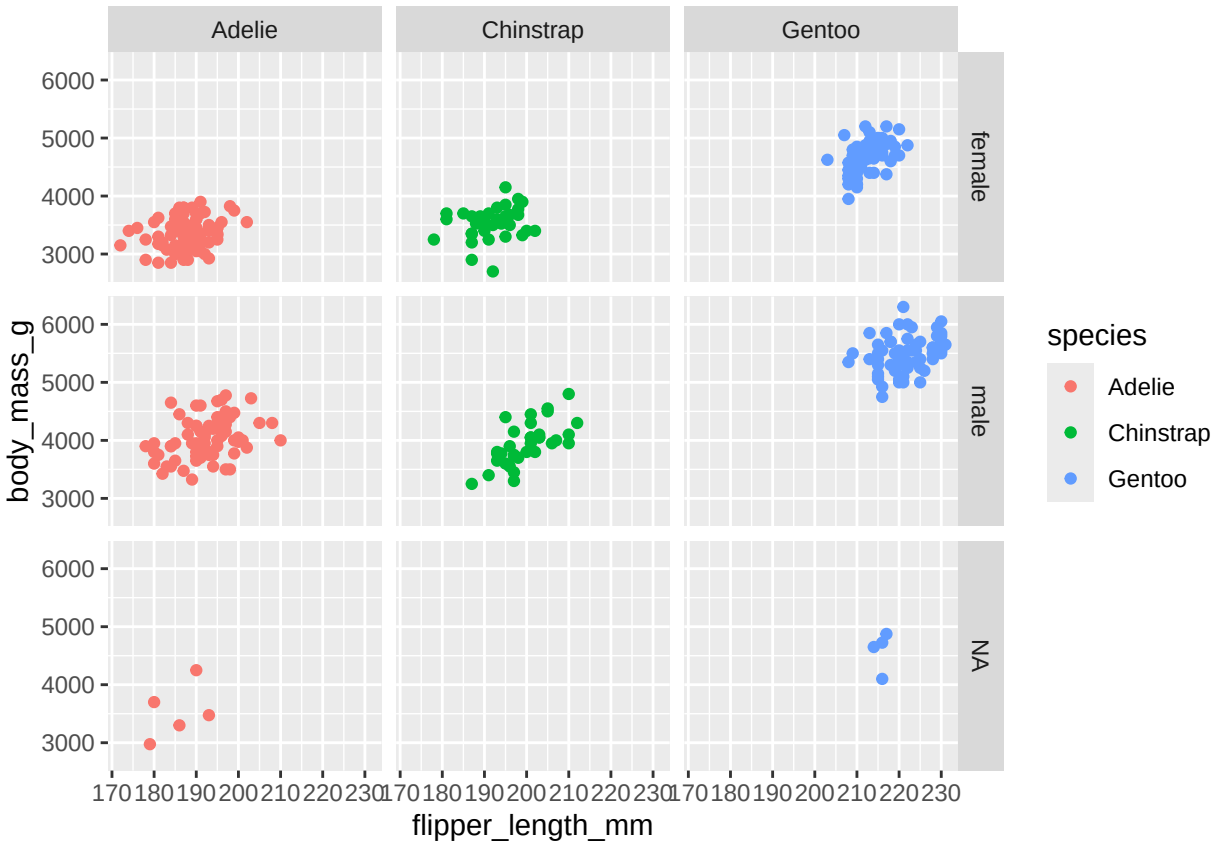
```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, colour=species))+
  facet_wrap(~species)
```



Tilde operator is used to define the relationship between dependent variable and independent variables in a statistical model formula. The variable on the left-hand side of tilde operator is the dependent variable and the variable(s) on the right-hand side of tilde operator is/are called the independent variable(s). So, tilde operator helps to define that dependent variable depends on the independent variable(s) that are on the right-hand side of tilde operator.

To facet a plot with two variables, use the facet_grid function. This will split the plot into facets vertically by the values of the first variable and horizontally by the values of the second variable.

```
ggplot(data=penguins)+
  geom_point(mapping=aes(x=flipper_length_mm, y=body_mass_g, colour=species))+
  facet_grid(sex~species)
```


Adelie
Chinstrap
Gentoo
female
male
NA
body_mass_g
6000
5000
4000
3000
170
180
190
200
210
220
230
flipper_length_mm
species
Adelie
Chinstrap
Gentoo