Introduction to ggplot2

Author: Cole Brookson **Date:** 20 July 2022

One way to plot in R, is to use the `ggplot2` package in the Tidyverse. "gg" stands for the "Grammar of Graphics" and the package makes for a convenient and consistent way to make almost any plot you want. Typically it's easiest to plot data in `ggplot2` using a dataframe, which means we usually don't need to make too major of changes to our data if we decide we want to change how we're plotting something in particular.

Components of a `ggplot`

The function used to make a plot is `ggplot()`. There is a consistent template that we'll need to use to get our plots to work:

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
  <GEOM_FUNCTION>()
```

Here we have three main components:

- *the data call* - this will almost always refer to a dataframe
- *the aesthetic mappings* - these are the variables we're plotting, and the specifications of how we want them to be displayed
- *the geom function* - each type of plot has it's own geom function (i.e. `geom_point()` for a scatterplot, `geom_line()` to plot timeseries, etc.)

A small point, note here that unlike other Tidyverse packages, we do not use the pipe `%>%` to link different functions, we use an addition operator `+`.

By way of a quick example, we'll make a super quick and dirty scatterplot of some long-term ecological data on salamanders and trout.

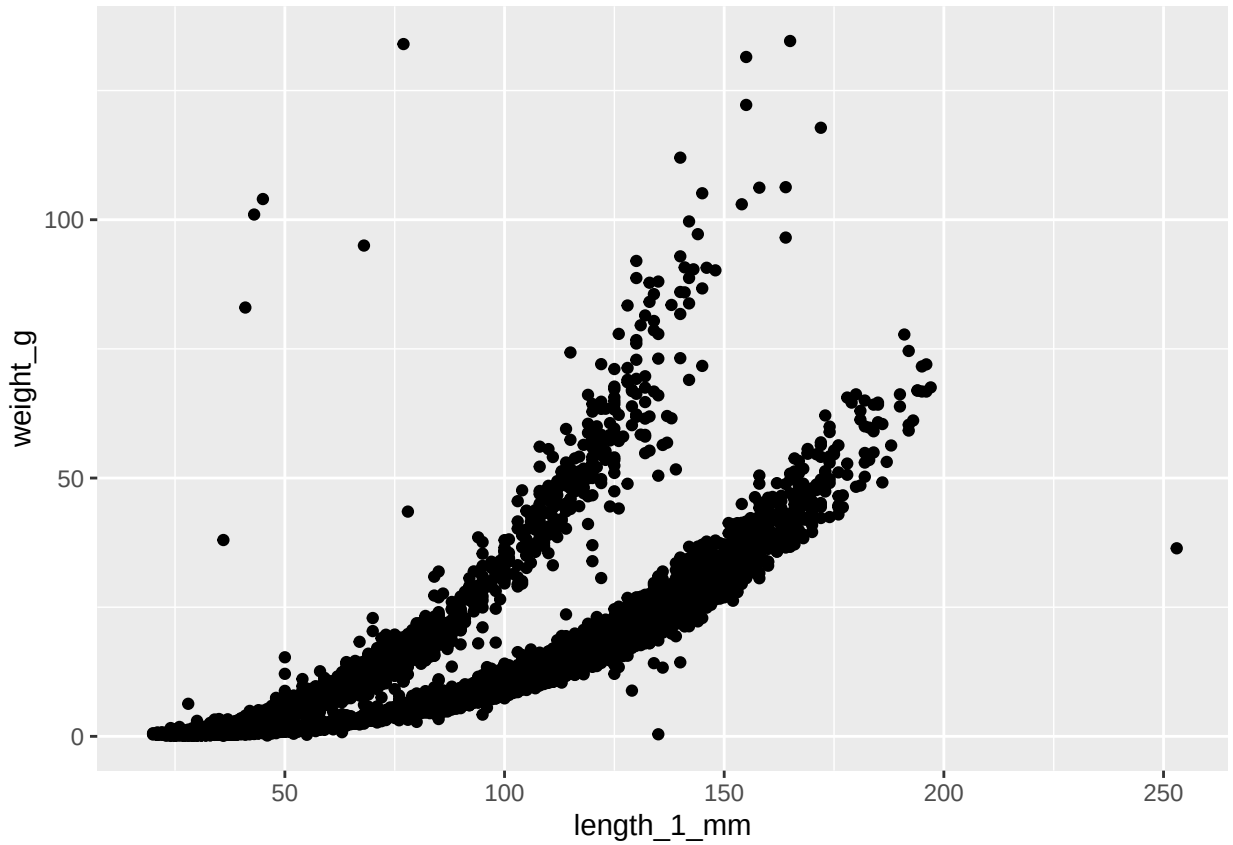
```
library(tidyverse)
library(lterdatasampler)

df <- lterdatasampler::and_vertebrates

names(df)
```

```
##  [1] "year"        "sitecode"    "section"     "reach"       "pass"
##  [6] "unitnum"     "unittype"    "vert_index"  "pitnumber"   "species"
## [11] "length_1_mm" "length_2_mm" "weight_g"    "clip"        "sampledate"
## [16] "notes"
```

The most basic version of the plot we may want to make is a scatter plot of two continuous variables, let's say length and weight:

```
ggplot(data = df, mapping = aes(x = length_1_mm, y = weight_g)) +
  geom_point()
```

```
## Warning: Removed 13279 rows containing missing values (geom_point).
```

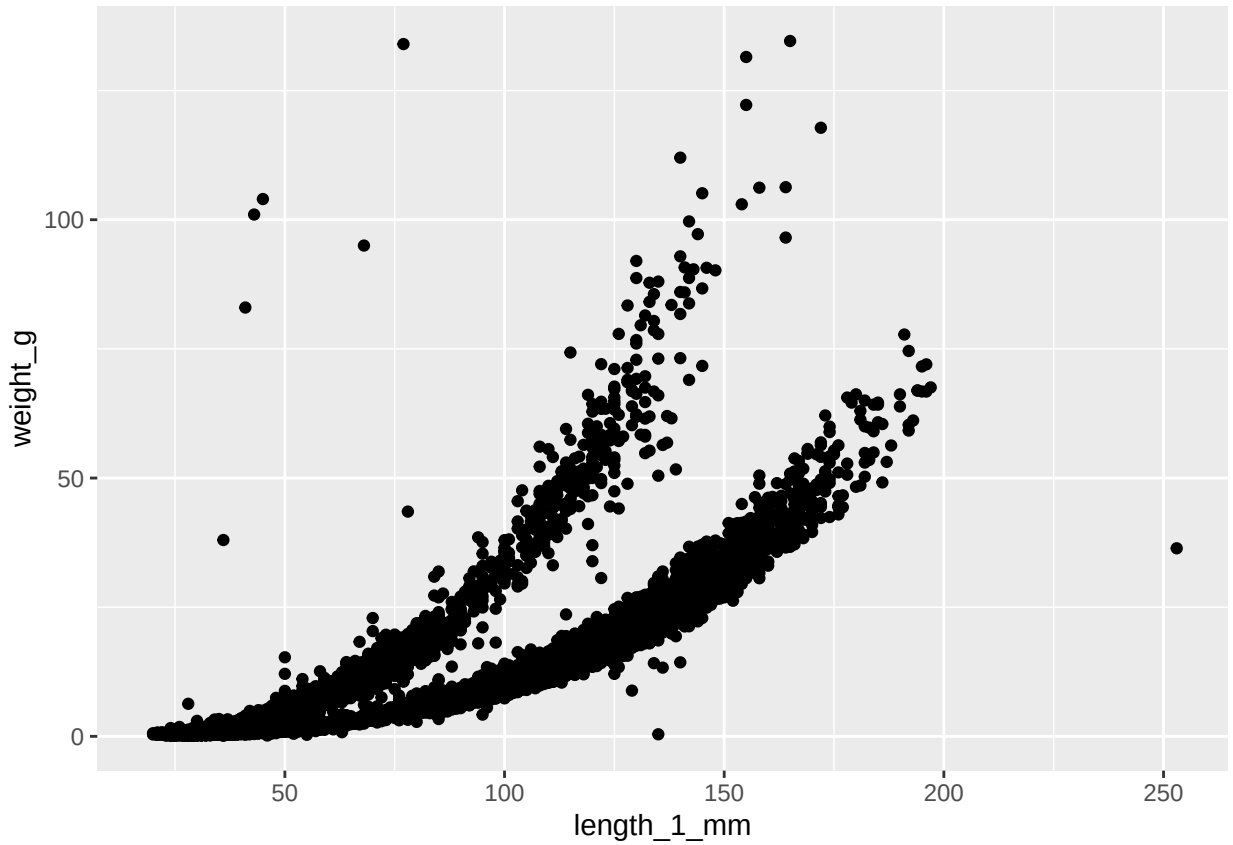


While this is completely correct, it is recommended to place the aesthetics arguments (`mapping = aes()`) in each `geom` argument individually. This will allow us to plot using multiple dataframes or variables.

So a better version may look like this:

```
ggplot() +
  geom_point(data = df, aes(x = length_1_mm, y = weight_g))
```

```
## Warning: Removed 13279 rows containing missing values (geom_point).
```



In the (LINK TO ITERATIVE PLOTTING) Iterative Plotting section we'll go over how to add components to go from an ugly plot like this to a publication-ready graphic.