R workshop #1 - revisited: plotting data using ggplot

Nicola Romanò

Introduction

Workshop 1 has introduced (or reminded) you to plotting data in R. This brief workshop will introduce you to a different way of plotting using R, using the `ggplot2` package. `ggplot2` is a system of creating visually pleasing graphics in a simple, easy-to-understand and visually pleasing manner[1]. Behind `ggplot2` lies a complex philosophy of visualisation, thus it is very hard to give you a quick, comprehensive view of it. This workshop will give you the basics and, if you are interested, you can pursue this further[2].

Learning objectives

After completing this workshop you will be able to:

- Use the basic features of ggplot

Installing and loading ggplot2

As always, when using a non standard R package, you need to install it first using[3].

```
install.package("ggplot2")
```

Once installed (which you only do once) you can load it using

```
library("ggplot2")
```

Aesthetics and geometries

The philosophy behind ggplot is that each plot is made out of *layers* that you can manipulate individually. The main command you are going to use for generating plots is `ggplot`.

But, first of all, let's load up some data! I am going to reuse the dataset from Workshop 1, `metab-workshop1.csv`[4].

```
metab <- read.csv("metab-workshop1.csv")
```

We can now pass the dataset to `ggplot`, and define the *aesthetics* that map the data to visual aspects of the plot.

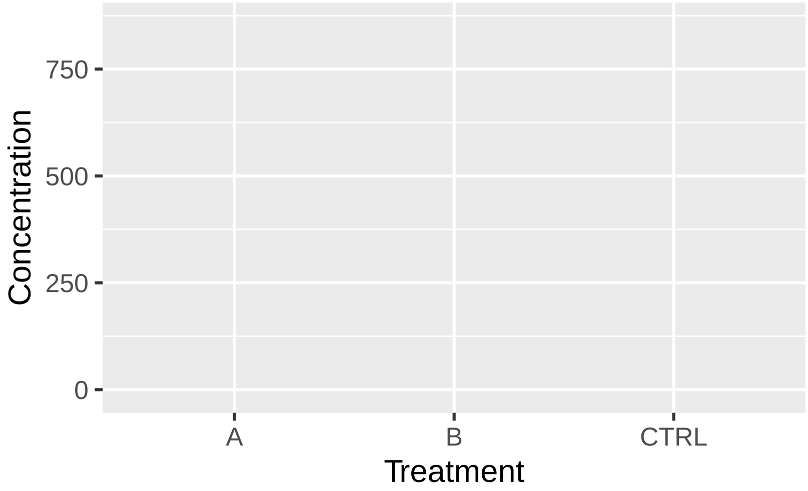
[1] `ggplot2` was reated by Hadley Wickham in 2005, based on theory developed by the statistician and computer scientist Leland Wilkinson in his 1999 book "The grammar of graphics".

[2] There are too many guides online (of which a lot are free) for me to list here. If you are really super-interested in this, you can try and read Hadley Wickham's book "ggplot2: Elegant Graphics for Data Analysis" which is probably one of the best references you can get.

[3] This may take a while, it's normal.

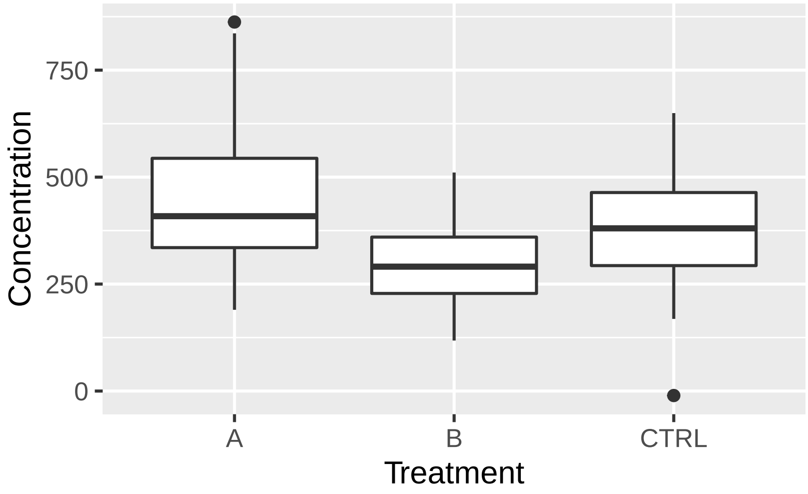
[4] Refer to Workshop 1 for the dataset description

```
ggplot(data = metab, aes(x = Treatment, y = Concentration))
```



But... wait a moment, there is nothing on the plot! That is because we did not tell `ggplot` what type of plot we want. Let's try again... this time asking for a boxplot. This is done by using geometries, that are generated through the `geom_....` functions. In our case we are going to use `geom_boxplot`. Because we want to add a new layer to our plot, we use + to add the boxplot. Easy, isn't it?

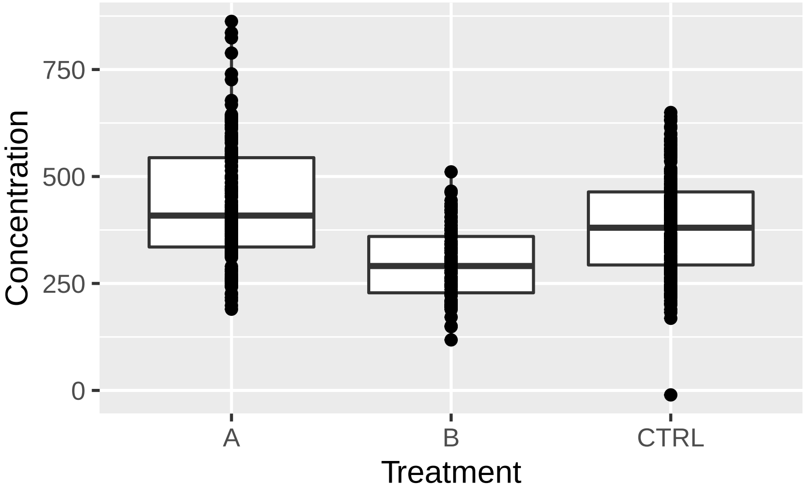
```
ggplot(data = metab, aes(x = Treatment, y = Concentration)) +
  geom_boxplot()
```



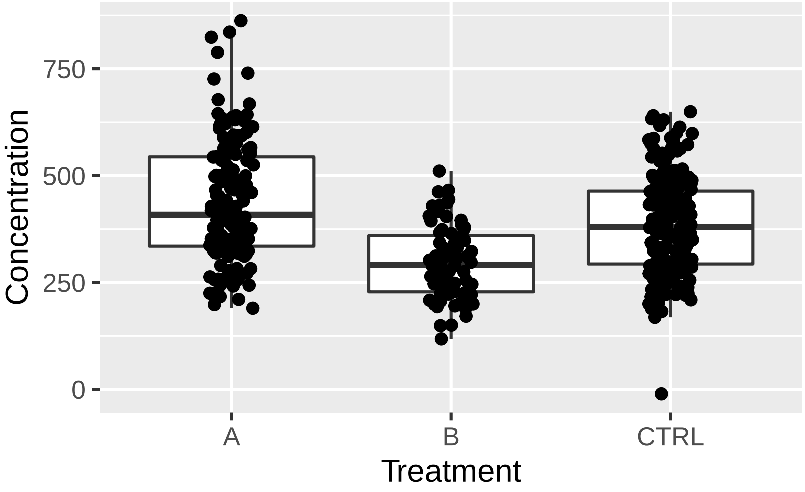
But, let's say we also want to add some points over the boxplot, how do we go about it? We just add another layer using `geom_point`[5]

[5] Note: you can always save the result of the plot into a variable and then add to that. For example `g <- ggplot(....)` and then `g + geom_boxplot()`.

```
ggplot(data = metab, aes(x = Treatment, y = Concentration)) +
  # Avoid plotting outliers on the boxplot, since we are adding points on top
  geom_boxplot(outlier.shape = NA) +
  geom_point()
```



Alternatively, try to use `geom_jitter` instead of `geom_point` to get some *jittered* points, as below[6]

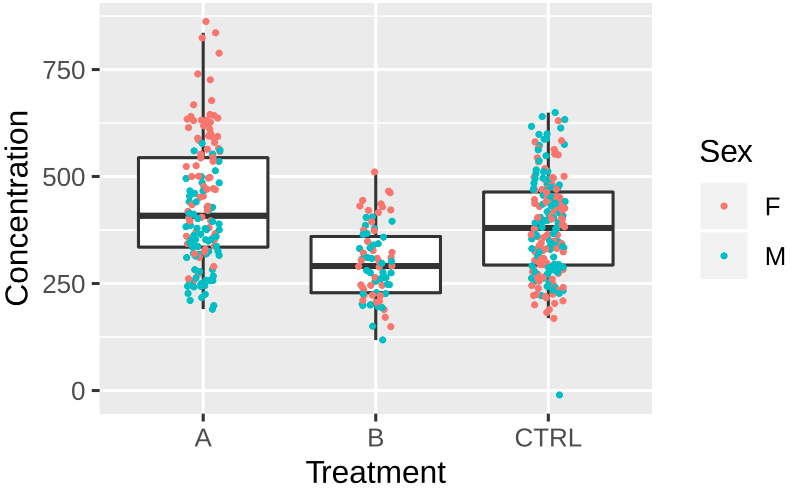


Let's now colour the points by sex, and make them smaller. Note that, since we are mapping a new variable to an aspect of the plot we need to be redefining the plot aesthetics[7].

```
ggplot(data = metab, aes(x = Treatment, y = Concentration)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.1, aes(col = Sex), size = 0.5)
```

[6] Use the `width` parameter to change the amount of jitter.

[7] Why don't we put the *col* parameter in the first `aes` block? Try it for yourself and see!

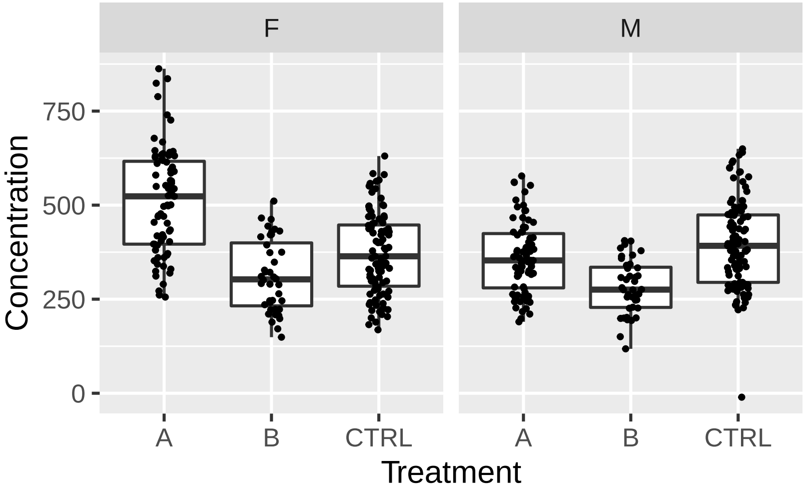


What happens if you map the color to Age instead? Try it and see why `ggplot2` makes it so easy to produce neat plots.

Faceting

The plot above is very pretty, but it is quite complicated to clearly see M vs F. One way we could go about this is faceting, that is, splitting the plot into subplots as follows

```
ggplot(data = metab, aes(x = Treatment, y = Concentration)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.1, size = 0.5) +
  facet_grid(~Sex)
```


F
M
Concentration
A
B
CTRL
A
B
CTRL
Treatment