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].

[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.

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].

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

```
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`.

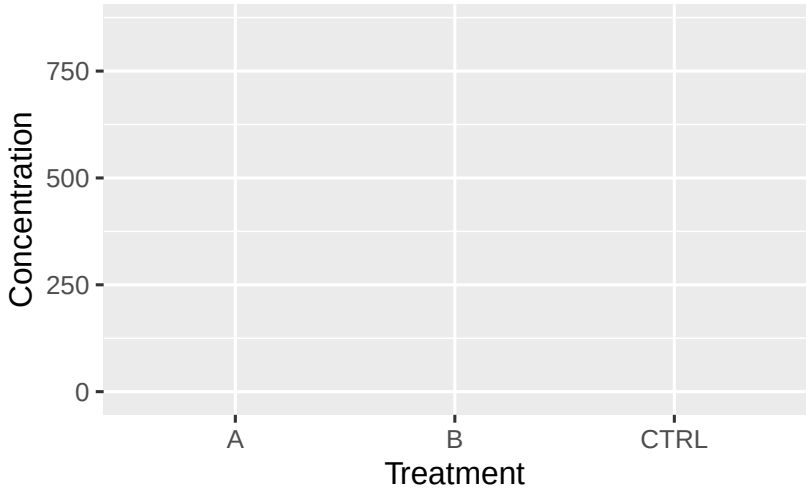
But, first of all, let's load up some data! I am going to reuse the dataset from Workshop 1, `metab-workshop1.csv`[4].

[4] Refer to Workshop 1 for the dataset description

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

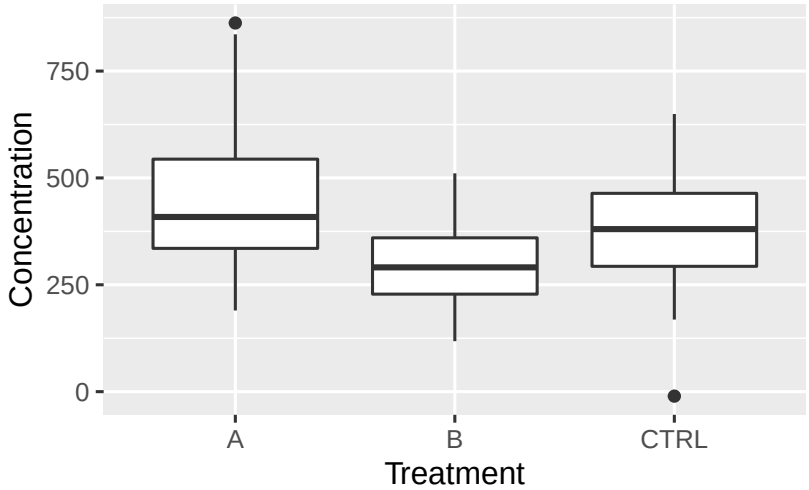
We can now pass the dataset to `ggplot`, and define the *aesthetics* that map the data to visual aspects of the plot.

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



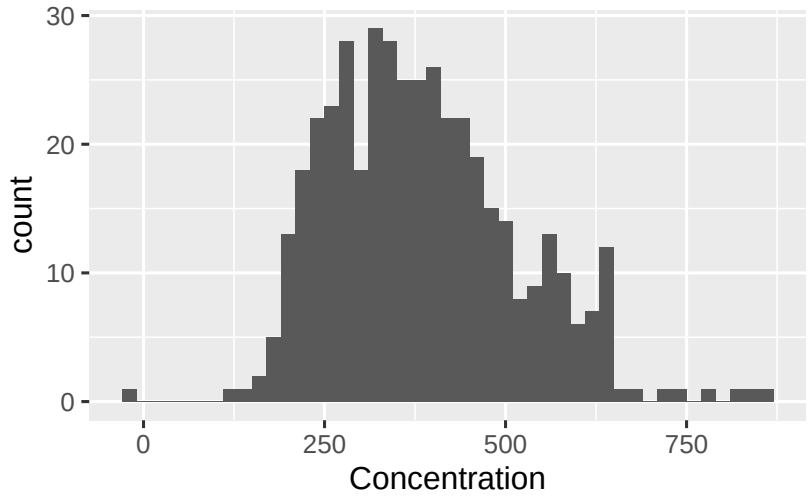
But... wait a moment, there is nothing on the plot! That is because we did not tell `ggplot` what type of plot we want. Let's try again... this time asking for a boxplot. This is done by using geometries, that are generated through the `geom_....` functions. In our case we are going to use `geom_boxplot`.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



If you wanted to plot an histogram, instead, you could do

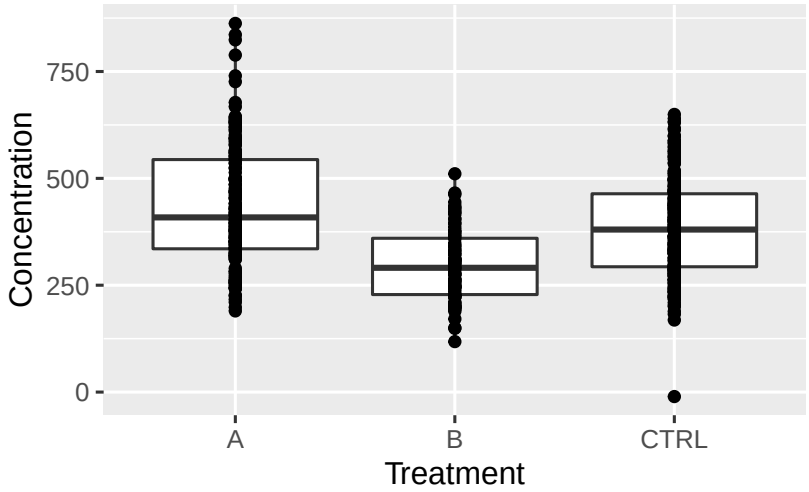
```
ggplot(data = metab, aes(x = Concentration)) +
  geom_histogram(binwidth = 20) # Or use bin to set the number of bins
```



But, let's say we want something more complicated, for instance adding some points over the boxplot, how do we go about it? We just add another layer using `geom_point`[5]

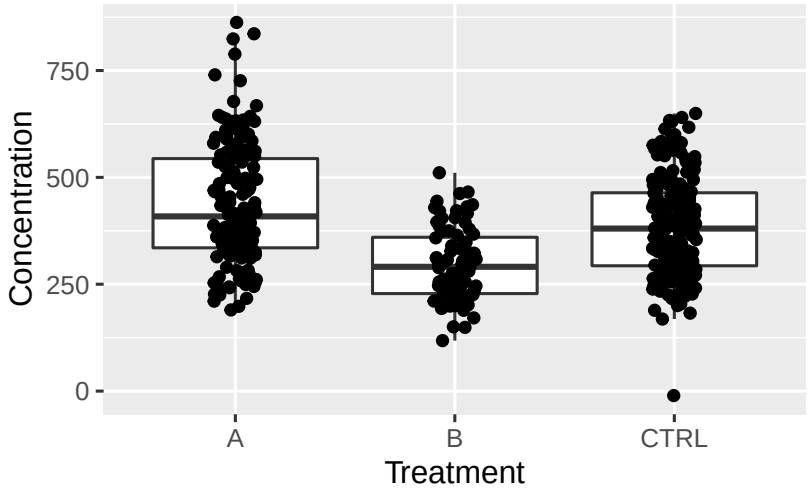
[5] Note: you can always save the result of the plot into a variable and then add to that. For example `g <- ggplot(....)` and then `g + geom_boxplot()`.

```
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]

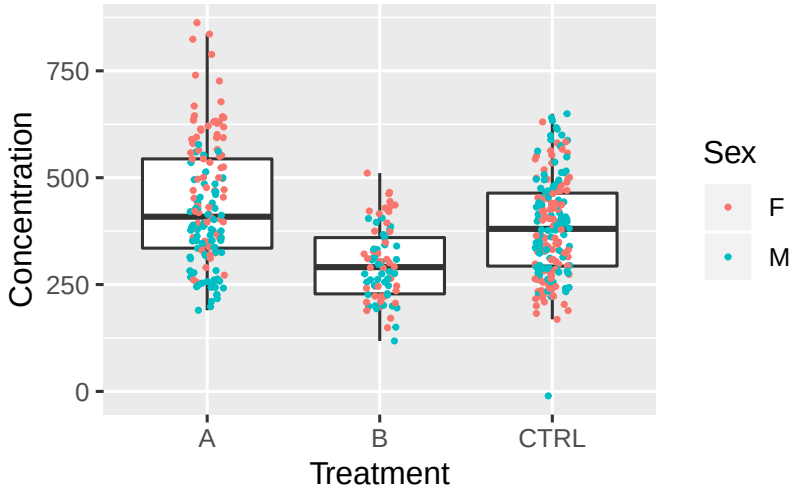
[6] Use the `width` parameter to change the amount of jitter.



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].

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

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



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

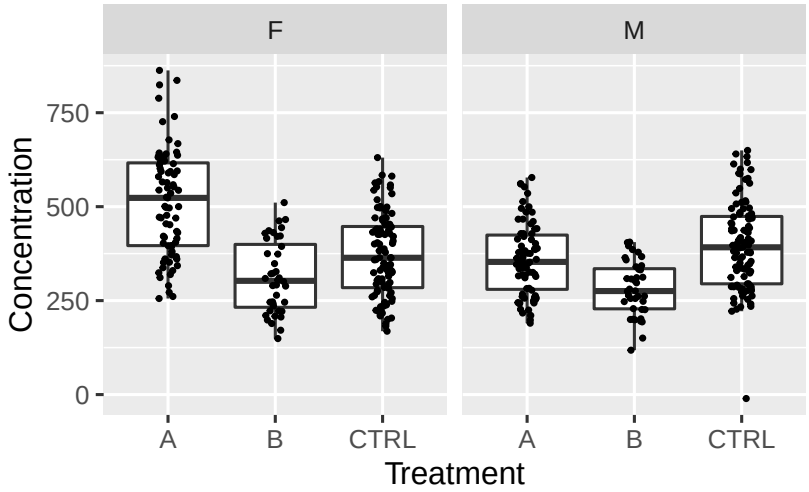
Challenge: can you plot two superimposed histograms of the metabolite concentration, one for men and one for women?

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)
```



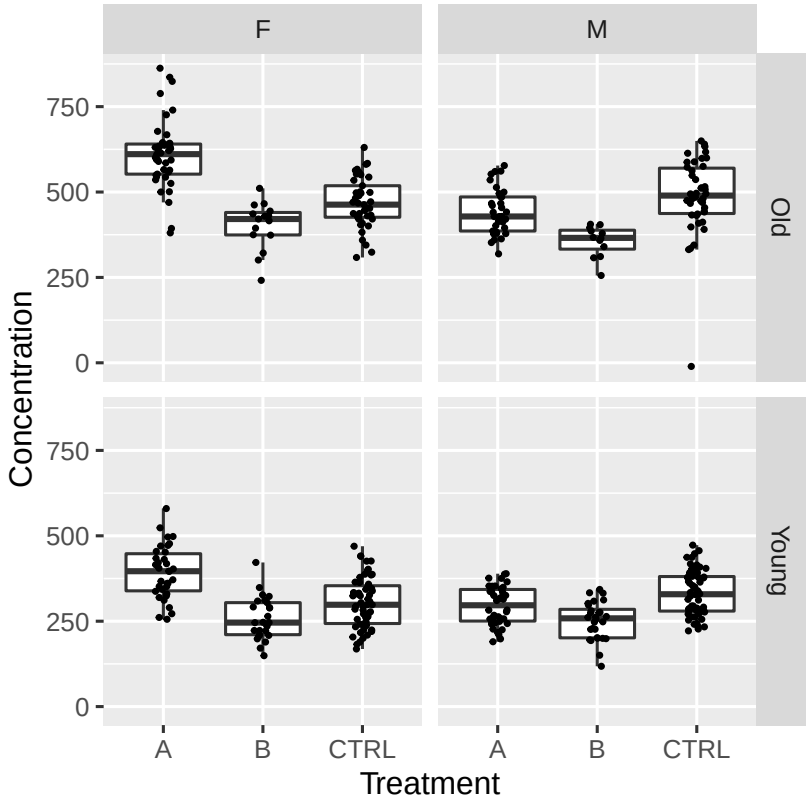
The ~Sex notation tells `ggplot2` to split the plot by Sex. This can also be used on multiple variables. Let's create a new one, by dividing people in Young (<=60 years old), and Old (>60 years old)[8]

[8] An absolutely arbitrary decision...

```
metab$AgeCateg <- ifelse(metab$Age <= 60, "Young",
    "Old")
```

Now we can facet on Sex and AgeCateg

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



What does the following code do instead?

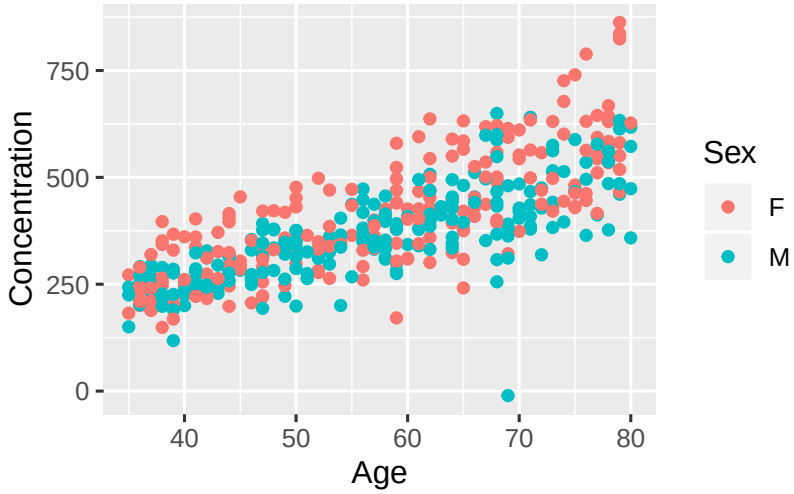
```
ggplot(data = metab, aes(x = Treatment, y = Concentration)) +
    geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.1,
    size = 0.5) + facet_grid(AgeCateg ~ .)
```

More complex features

We will finish this workshop with a more complex graph. This is really just the *tip of the iceberg*, but hopefully it has given you some inspiration to pursue this further in your own time!

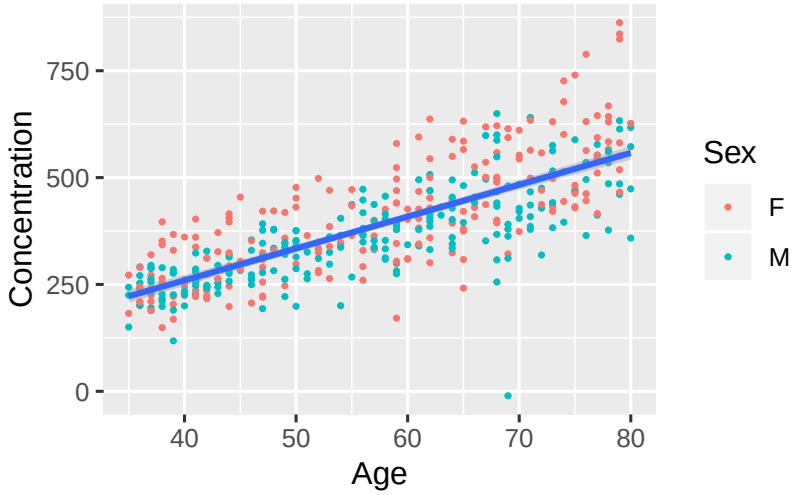
Consider the following plot

```
ggplot(data = metab, aes(x = Age, y = Concentration)) +
  geom_point(aes(col = Sex))
```



Let's say we want to fit a line through that. We can use a "smoother"

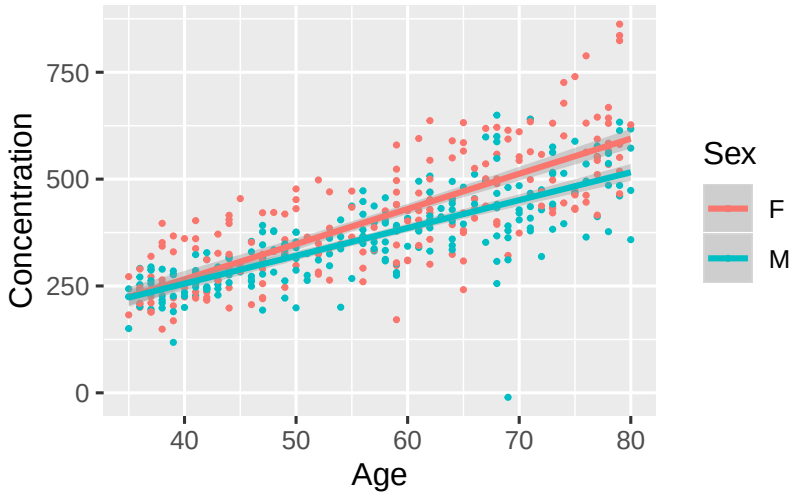
```
ggplot(data = metab, aes(x = Age, y = Concentration)) +
  geom_point(aes(col = Sex), size = 0.5) +
  geom_smooth(method = "lm")
```



This fits a line (calling `lm` in the background... have a look at your notes from last year if you don't remember what that does!) through all the points[9]. However, wouldn't it be nicer to have a line for males and one for females? That's easily done!

[9] Note the light gray shading around the line, those are confidence intervals for the fit.

```
ggplot(data = metab, aes(x = Age, y = Concentration)) +
  geom_point(aes(col = Sex), size = 0.5) +
  geom_smooth(method = "lm", aes(col = Sex))
```



Now it's your turn!

Try and use `ggplot2` to plot the other datasets from Workshop 1. You can also explore the different type of plots available at this address `https://www.r-graph-gallery.com/`.