Introduction to visualisation in R with the package ggplot2

Introduction

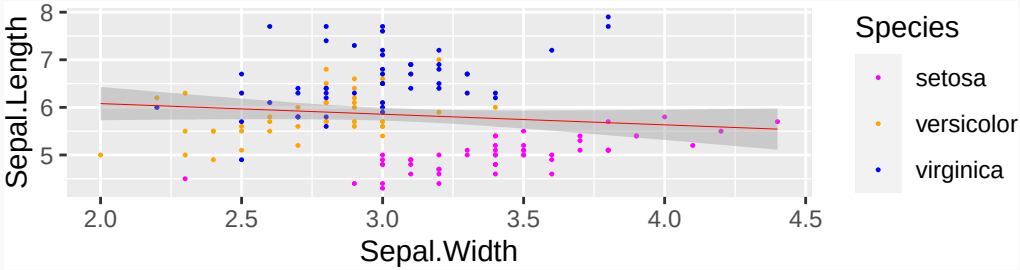
Jean-Michel Marin and Benjamin Charlier

January 2023

CNRS – IMAG (Montpellier, France)

Introduction

```
library(ggplot2)
ggplot(iris,aes(x=Sepal.Width, y=Sepal.Length, colour=Species))+
  geom_point(size=0.2)+
  scale_colour_manual(values=c("magenta", "orange", "blue"))+
  geom_smooth(method="lm", colour="red", size=0.15)
```



Introduction

- ggplot2 visualization package for R

```
install.packages("ggplot2")
install.packages("ggthemes")
```

- ggplot2 is a system for declaratively creating graphics. Allows representation and exploration of datasets
- Learning curve is steep but... investment is quickly extremely profitable
  - See the **documentation** at https://ggplot2.tidyverse.org/
  - See gallery of examples https://r-graph-gallery.com/ggplot2-package.html, https://exts.ggplot2.tidyverse.org/gallery/, http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html ...
- Other related project : plotly https://plotly.com/r/

Main principle of ggplot2 package

ggplot2 syntax is about adding successives layers

- First layer is the **graph canvas**
  - Importing the considered data set, and variables names to be plotted
- Second layer is **aesthetic mapping**
  - Choosing the type of graph you want to plot: scatterplot, boxplot, barplot...
- Then come the **refinement layers**
  - they will allow you to choose the colors, the axis scales, the legend options...

Instantiate a ggplot2 graph

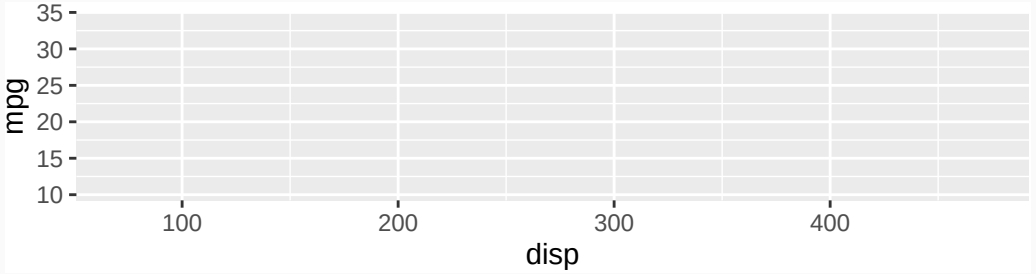
To instantiate the ggplot2 graph, we use `ggplot()`together with the argument `aes()`

- ggplot2 graphs always start with this code

```
ggplot(dataset, aes(x=, y = ))
```

Instantiate a ggplot2 graph

```
ggplot(mtcars, aes(x=disp,y=mpg))
```



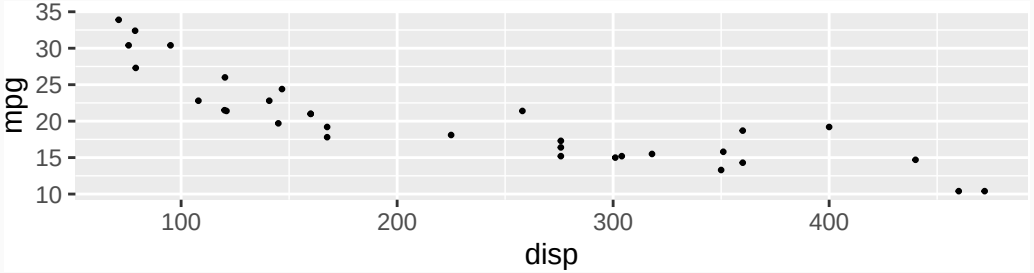
Choosing a plot type : geom_XXX

To define the type of plot : scatter plot, boxplot, barplot...

- Add a + operator at the end of the first line (ie ggplot2 class constructor), and add a new line with the desired function:
  - `geom_point()`: scatter plot
  - `geom_boxplot()`: boxplot
  - `geom_bar()`: barplot...

Choosing a plot type : geom_XXX

```
ggplot(mtcars, aes(x=disp, y=mpg))+
  geom_point(size=0.5)
```



Graphical options

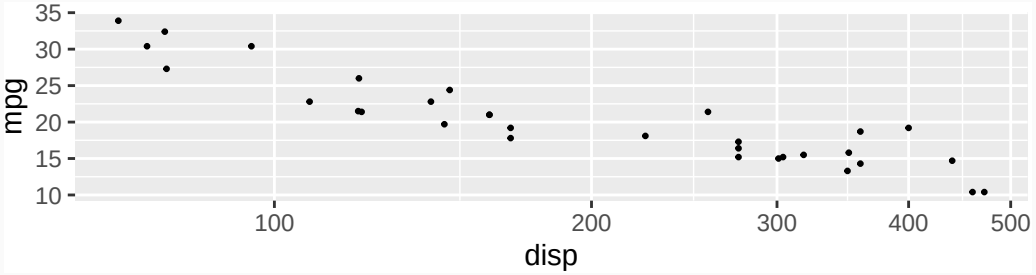
Finaly, fine tuning of aestethic options may be setup by adding extra layers

- axes sizes and scales : `scale_x_continuous(trans='log10')`, `xlim`, `expand_limits`, ...
- colours : `scale_colour_manual()`
- axes labels, with the functions `xlab()`, `ylab()`, ...
- legend or caption, e.g. `theme(legend.position,="bottom")`

Graphical options

To transform the x-axis into log scale add a layer with the fucntion `coord_trans()`

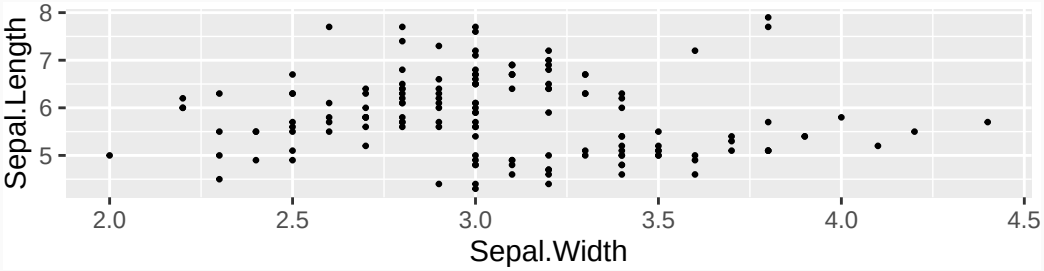
```
ggplot(mtcars, aes(x=disp, y=mpg))+
  geom_point(size=0.5)+
  coord_trans(x="log10")
```



Scatterplot with ggplot2

Assume we need to graph a scatter plot with iris dataset: with x-axis being `Sepal.Width` variable and y-axis being `Sepal.Length`:

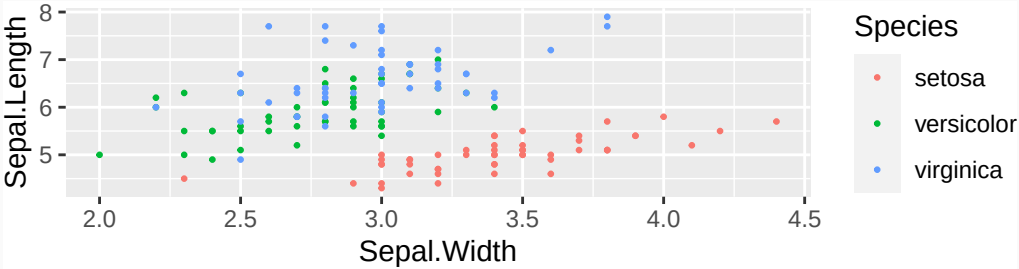
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length))+
  geom_point(size=0.5)
```



Scatterplot with ggplot2

The iris dataset is composed with 3 diferent iris species (variable `Species`)

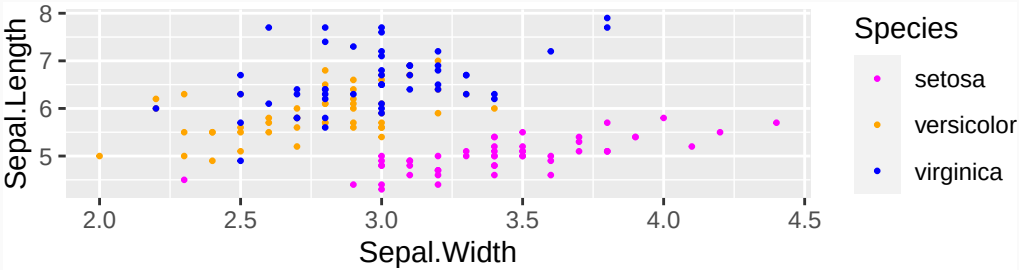
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, colour=Species))+
  geom_point(size=0.5)
```



Scatterplot with ggplot2

To change default colours with your custom choice, use the function `scale_colour_manual()`. Careful with red/green (color blind) and yellow (readibility)...

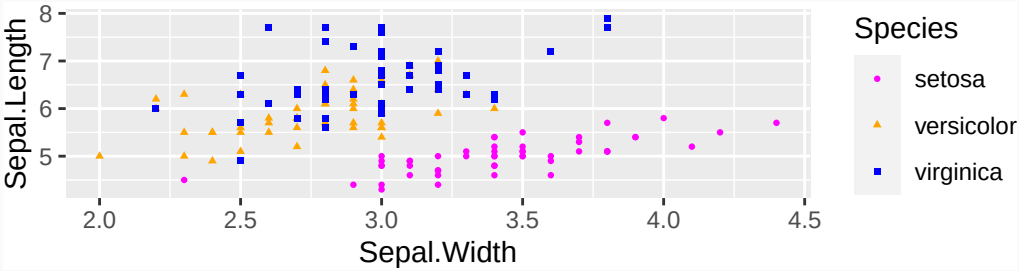
```
ggplot(iris,aes(x=Sepal.Width, y=Sepal.Length, colour=Species))+
  geom_point(size=0.5)+
  scale_colour_manual(values=c("magenta", "orange", "blue"))
```



Scatterplot with ggplot2

To display different point types, we use the shape argument in `aes()`.

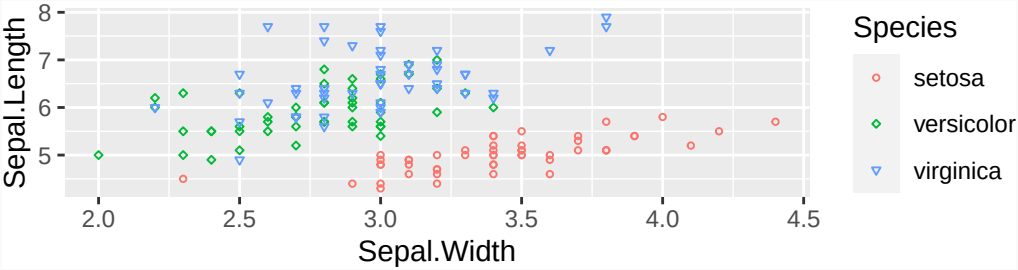
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, colour=Species, shape=Species))+
  geom_point(size=0.8)+
  scale_colour_manual(values=c("magenta", "orange", "blue"))
```



Scatterplot with ggplot2

To manually set point types, we use `scale_shape_manual()`

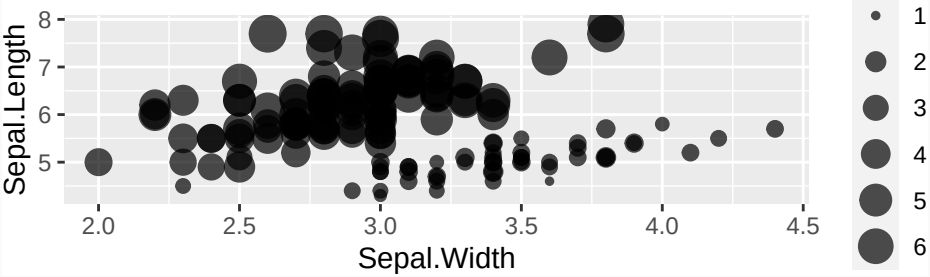
```
ggplot(iris,aes(x=Sepal.Width, y=Sepal.Length, colour=Species, shape=Species))+
  geom_point(size=0.8)+
  scale_shape_manual(values=c(21, 23, 25))
```



Scatterplot with ggplot2

We are able to change the size of points. For instance, let the size depends on the value of some variable

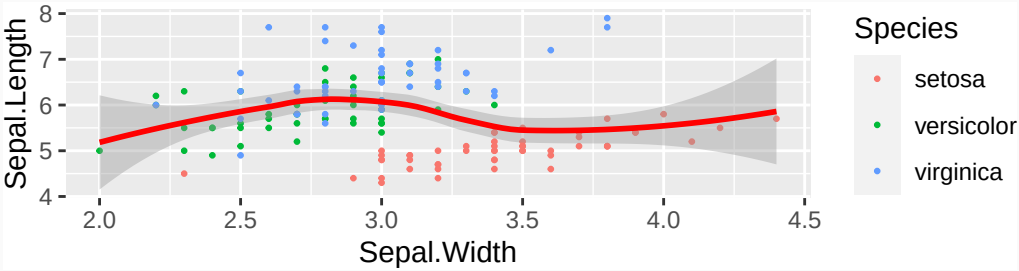
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, size=Petal.Length))+
  geom_point(alpha=0.7)
```



Scatterplot with ggplot2

To add a regression curve or line, use the function `geom_smooth()`

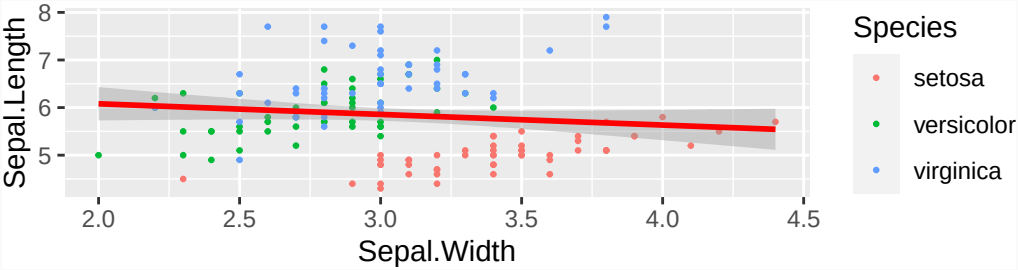
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, colour=Species))+
  geom_point(size=0.5)+
  geom_smooth(colour="red")
```



Scatterplot with ggplot2

Useful trick: adding a (least square) regression line, using the argument `method="lm"`

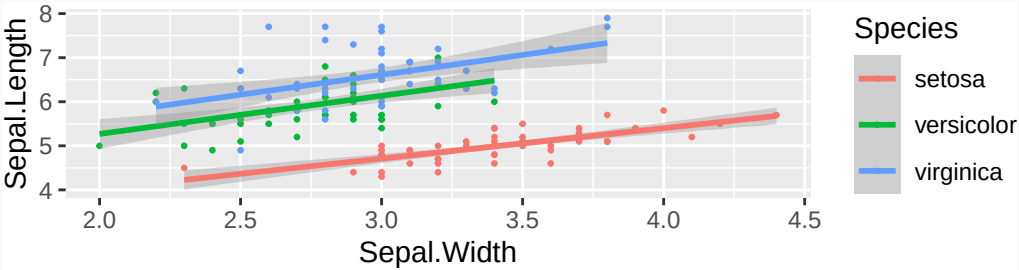
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, colour=Species))+
  geom_point(size=0.5)+
  geom_smooth(method="lm", colour="red")
```



Scatterplot with ggplot2

It is also possible to add a (least square) regression by species, using the argument `group` in `aes()` (first line)

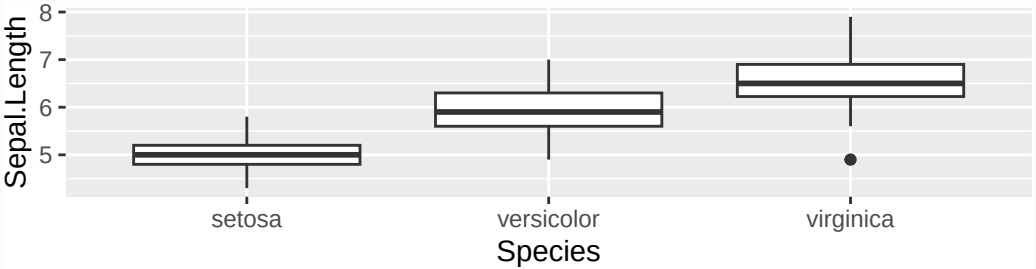
```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, colour=Species, group=Species))+
  geom_point(size=0.5)+
  geom_smooth(method="lm")
```



Boxplot with ggplot2

Assume we now need a boxplot of the `Sepal.Length` variable by species...

```
ggplot(iris,aes(y=Sepal.Length, x=Species))+
  geom_boxplot()
```



Boxplot with ggplot2

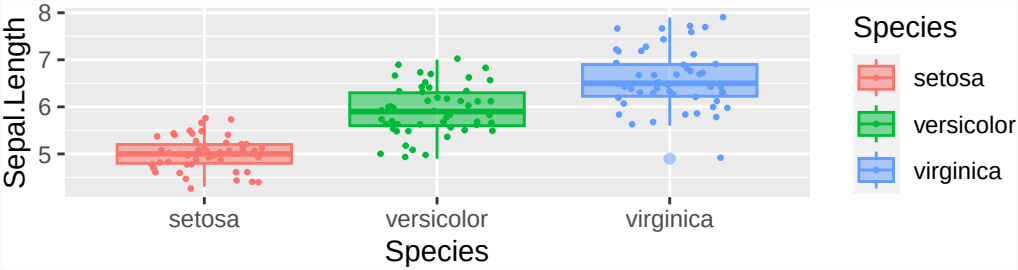
To change the colour of the boxes, we use the `fill` option (instead of `colour` as before), in function `aes()`:

```
ggplot(iris, aes(y=Sepal.Length, x=Species, fill=Species))+
  geom_boxplot()
```

Boxplot with ggplot2

To add the observed data points, we use the `geom_jitter()` layer:

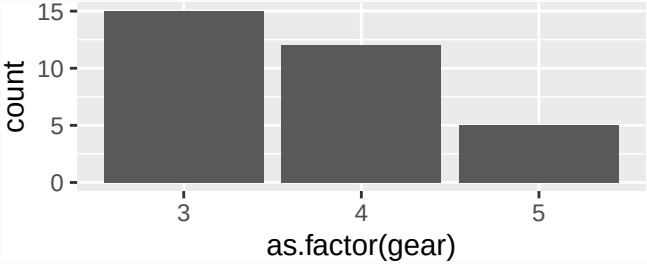
```
ggplot(iris,aes(y=Sepal.Length, x=Species, fill=Species, colour=Species))+
  geom_boxplot(alpha=0.5)+
  geom_jitter(width=0.25, size=0.5)
```



Barplots with ggplot2

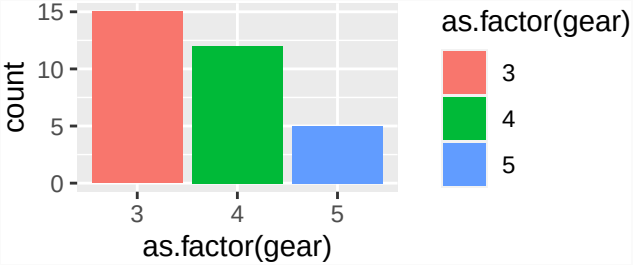
Counting barplot may be plotted easily:

```
ggplot(mtcars,aes(x=as.factor(gear)))+
  geom_bar()
```



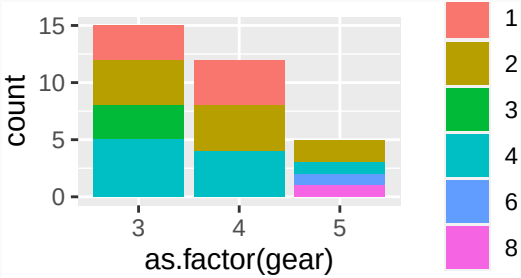
Barplots with ggplot2

```
ggplot(mtcars, aes(as.factor(gear), fill=as.factor(gear)))+
  geom_bar()
```



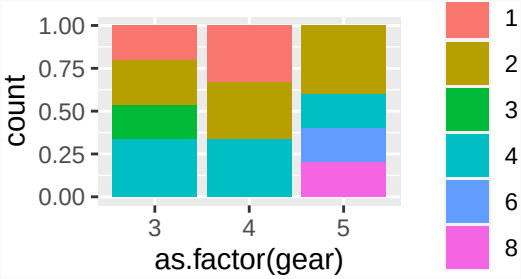
Barplots with ggplot2

```
ggplot(mtcars, aes(as.factor(gear), fill=as.factor(carb)))+
  geom_bar()
```



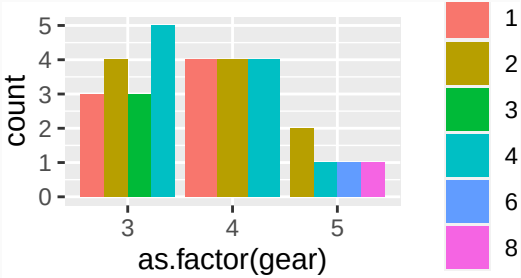
Barplots with ggplot2

```
ggplot(mtcars, aes(as.factor(gear)))+
  geom_bar(aes(fill=as.factor(carb)), position="fill")
```



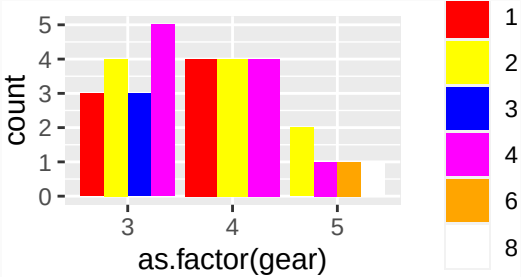
Barplots with ggplot2

```
ggplot(mtcars,aes(as.factor(gear),fill=as.factor(carb)))+
  geom_bar(position="dodge")
```



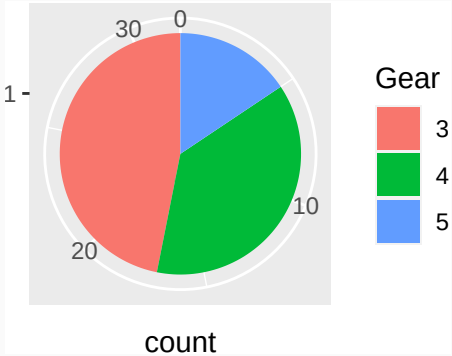
Barplots with ggplot2

```
ggplot(mtcars, aes(as.factor(gear)))+
  geom_bar(aes(fill=as.factor(carb)), position="dodge")+
  scale_fill_manual(values=c("red", "yellow", "blue", "magenta", "orange", "white"
```



Pie chart with ggplot2

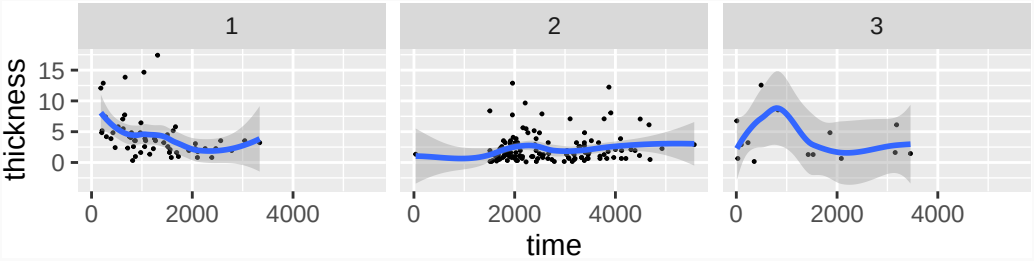
```
ggplot(mtcars, aes(x=factor(1), fill=as.factor(gear)))+
  geom_bar(width=1)+
  coord_polar("y")+
  labs(x="", fill="Gear")
```



Facetting (a.k.a. subplots)

Dividing a graph into subplots, according to the modalities of one or more variables

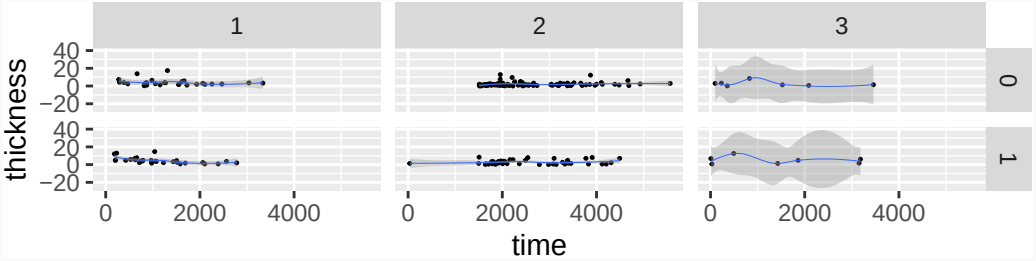
```
library(MASS)
ggplot(Melanoma, aes(y=thickness,x=time))+
  geom_point(size=0.2)+
  geom_smooth()+
  facet_wrap(~status)
```



Facetting (a.k.a. subplots)

Dividing a plot according to the values of 2 variables

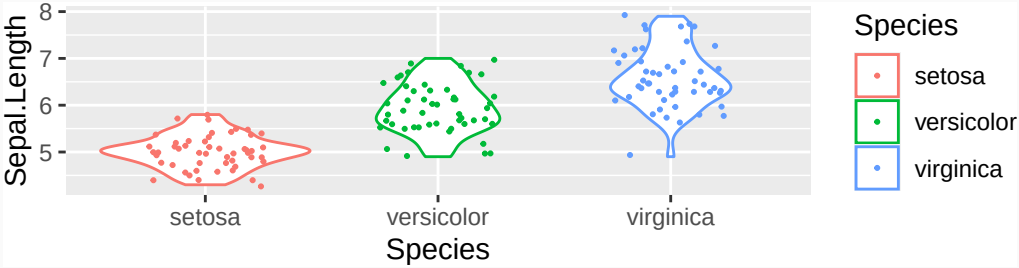
```
ggplot(Melanoma, aes(y=thickness, x=time))+
  geom_point(size=0.2)+
  geom_smooth(size=0.15)+
  facet_grid(sex~status)
```



Mapping

The option `colour=Species` in the `aes()` function is **inherited** in every subsequent layers...

```
ggplot(iris, aes(y=Sepal.Length, x=Species, colour=Species))+
  geom_violin()+
  geom_jitter(width=0.25, size=0.4)
```



Mapping

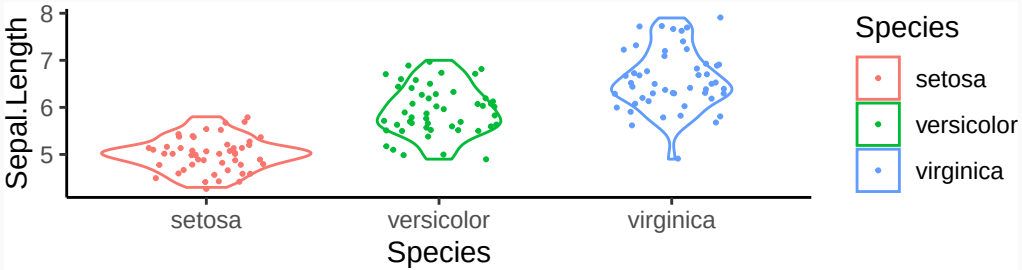
To fix the colour in a particular layer, we have to explicitely call the `colour` option in this layer

```
ggplot(iris, aes(y=Sepal.Length,x=Species,colour=Species))+
  geom_violin()+
  geom_jitter(width=0.25, colour="magenta", alpha=0.5, size=0.4)
```

Themes

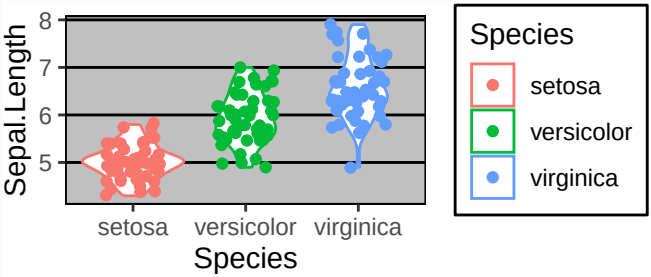
There is a variety of different graphical theme

```
ggplot(iris, aes(y=Sepal.Length, x=Species, colour=Species))+
  geom_violin()+
  geom_jitter(width=0.25, size=0.4)+
  theme_classic()
```



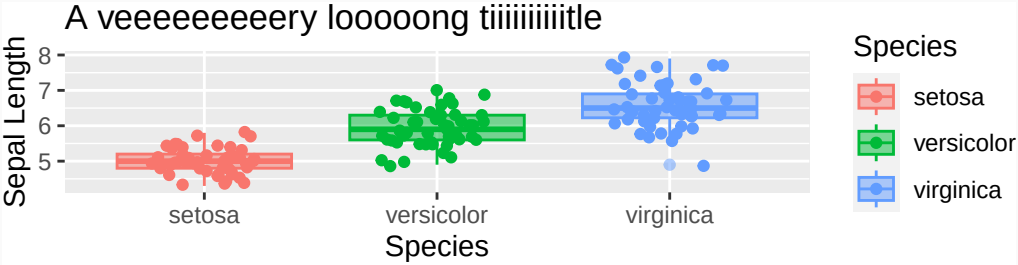
Themes

```
library(ggthemes)
ggplot(iris, aes(y=Sepal.Length, x=Species, colour=Species))+
  geom_violin()+
  geom_jitter(width=0.25)+
  theme_excel()
```



Axes, title and legend

```
ggplot(iris,aes(y=Sepal.Length, x=Species, fill=Species,colour=Species))+
  geom_boxplot(alpha=0.5)+
  geom_jitter(width=0.25)+ggtitle("A veeeeeeeeery looooong tiiiiiiiiiitle")+
  ylab("Sepal Length")+
  xlab("Species")
```



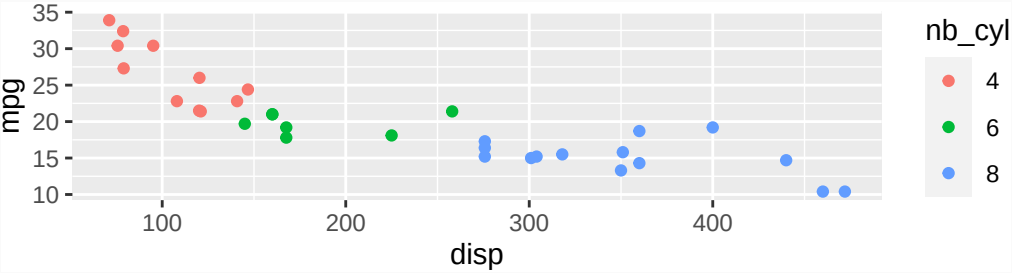
Outro

Consider ggplot2 extensions like `animate` :
`https://gganimate.com/#yet-another-example`

Exercise 1

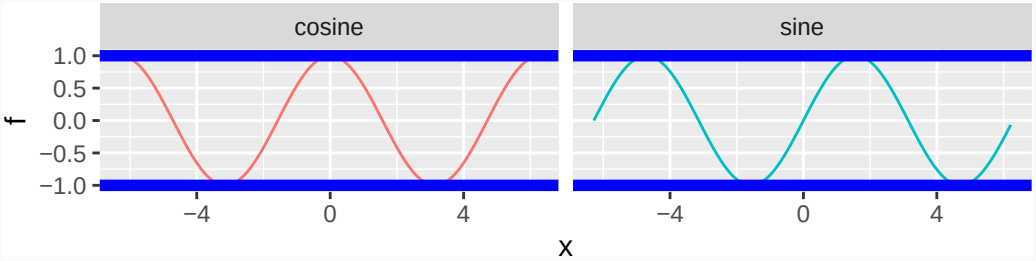
With the `mtcars` dataset (from the R package `datasets`)

1. Plot an histogram of `mpg` variable
2. Plot a barplot of variable `cyl`
3. Plot the point cloud `disp` vs `mpg` with a different colour depending on the values of `cyl` variable



Exercise 2

1. Use ggplot2 to plot the sine function on the $[-2\pi, 2\pi]$.
2. Add the lines $y = -1$ and $y = 1$ in thick blue.
3. Add the cosine function.
4. Add legend to identify the sine and cosine function.
5. Plot cosine and sine on two subgraphs (use `facet_wrap`).

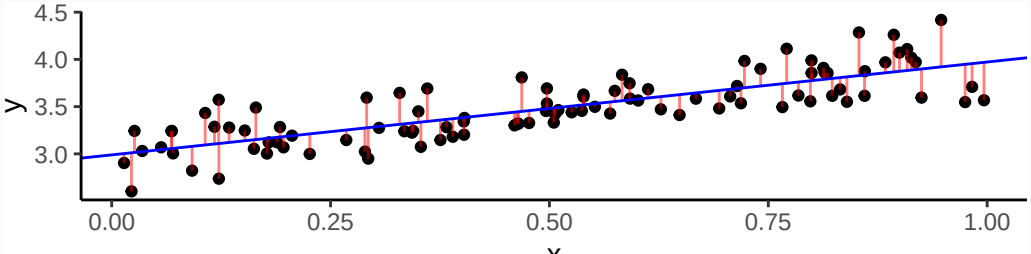


Exercise 3

1. Simulate a sample (x_i, y_i), $i = 1, \cdots, 100$, following the model

$$Y_i = 3 + X_i + \varepsilon_i$$

   where X_i are i.i.d. uniform on $[0, 1]$ random variables and ε_i are i.i.d. centered Gaussian of variance 0.04.
2. Plot the corresponding dataset as a point cloud and the least square regression line with the `geom_smooth` function.
3. Plot the residuals: add a vertical segment linking each point to the least square regression line.



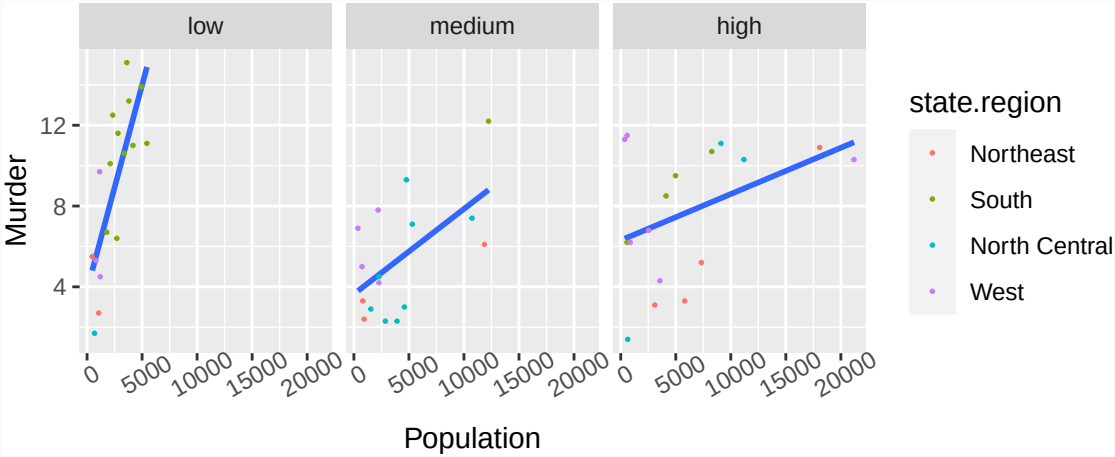
Exercise 4

Consider the dataset `states` from datasets package:

```
data(state)
states <- data.frame(state.x77,state.name=rownames(state.x77),
                     state.region=state.region)
```

1. Create a variable `Income1` taking the value `low` if the `Income` is in the first tercile, `medium` in the second and `high` in the third one (use functions `quantile` and `cut`).
2. Plot the point clouds `Population` vs `Murder` for each value of `Income1` (3 point clouds).
3. Use a different colour for each point according to the variable `state.region` and add the least square regression line on each graph.

Exercise 4


low
medium
high
Murder
12
8
4
0
5000
10000
15000
20000
Population
state.region
Northeast
South
North Central
West

Bibliography

- P.-A. Cornillon, A. Guyader, F. Husson, N. Jégou, J. Josse, N. Klutchnikoff, E. Le Pennec, E. Matzner-Løber, L. Rouvière, B. Thieurme - *R pour la statistique et la science des données*. Pratique de la statistique, Presses universitaires de Rennes. 2018
- ggplot2 documentation. `https://ggplot2.tidyverse.org/`