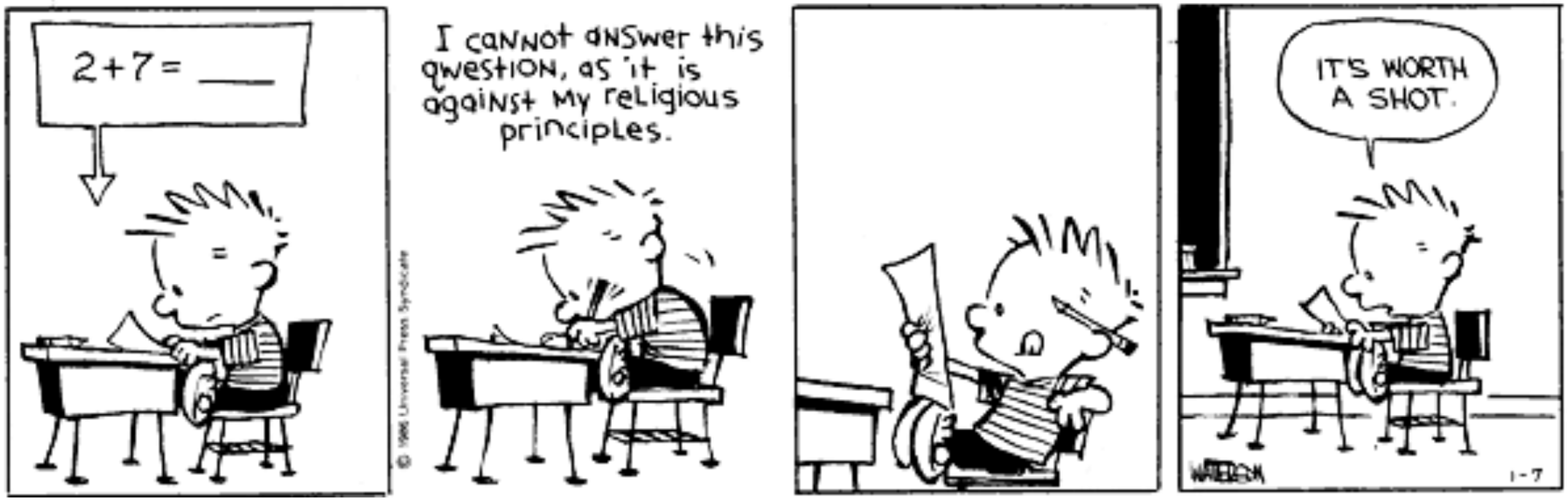
UNIVERSITÉ DE
NEUCHÂTEL

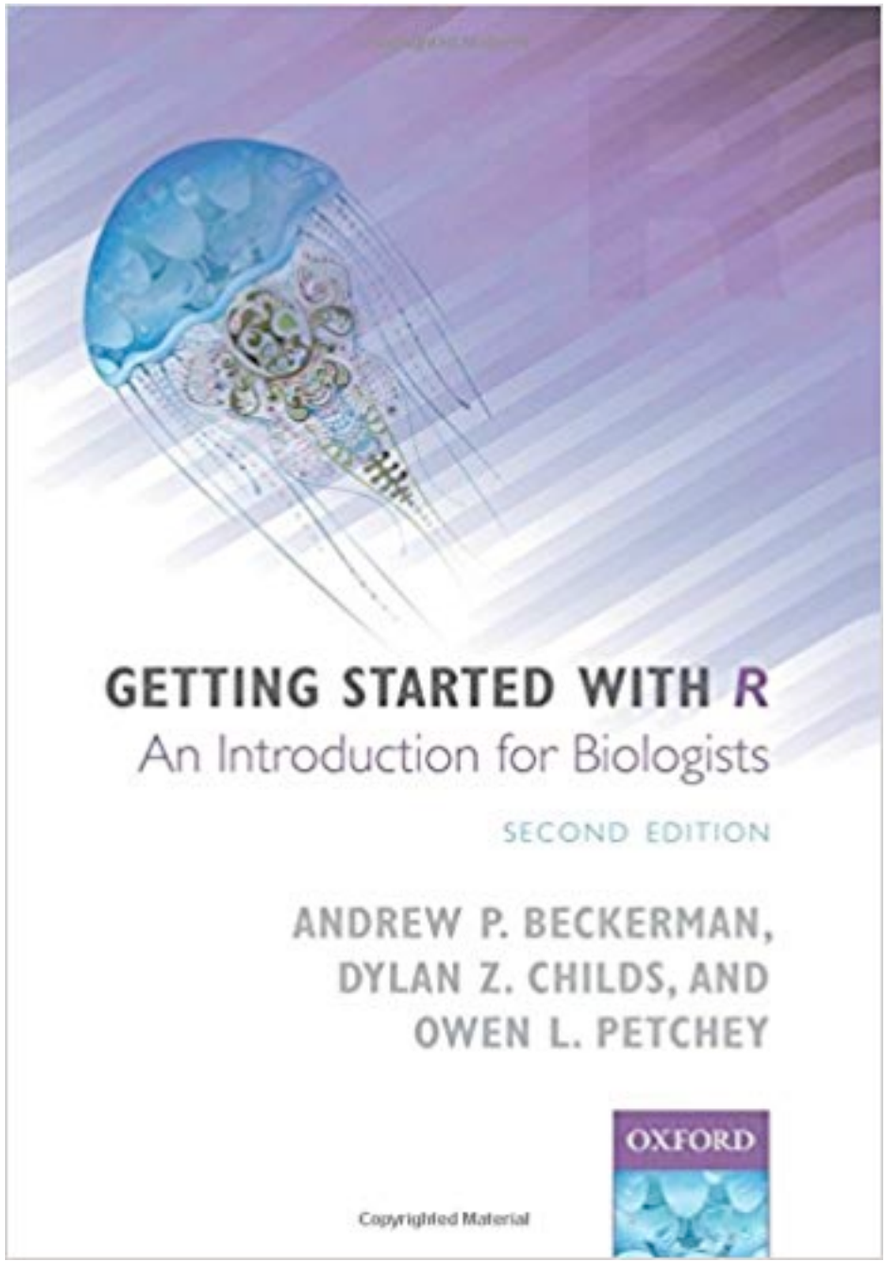
Introductory statistics with R

Solutions to exercises

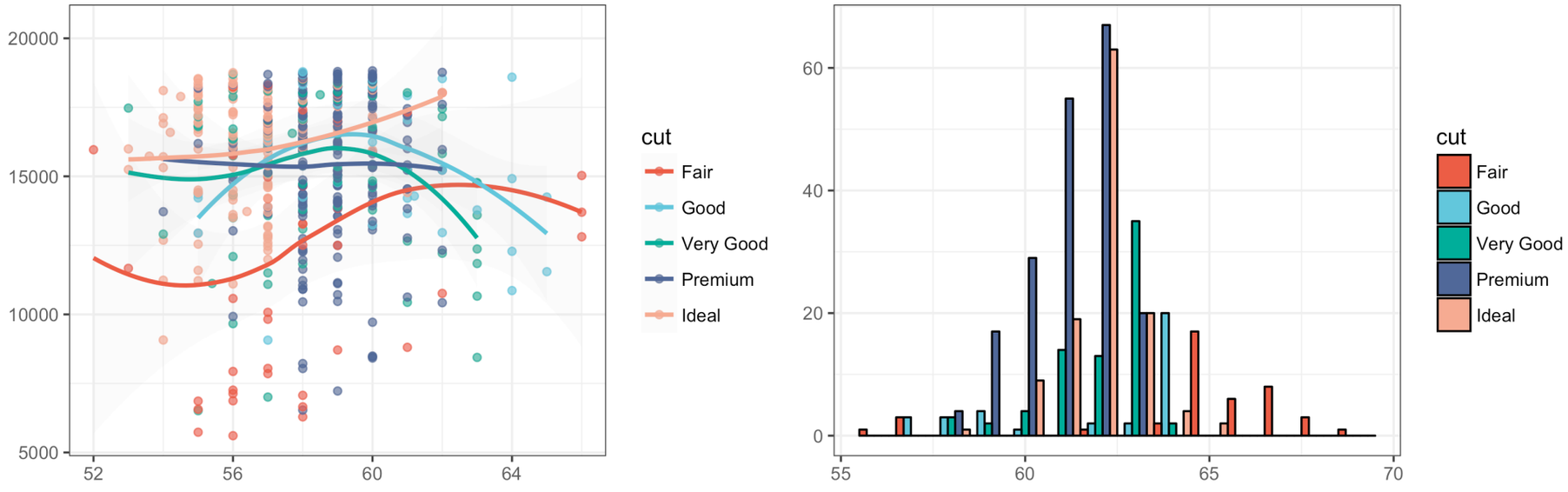


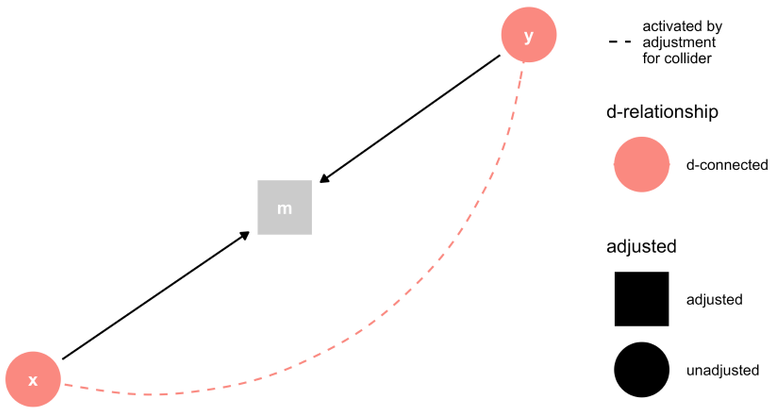
For next week

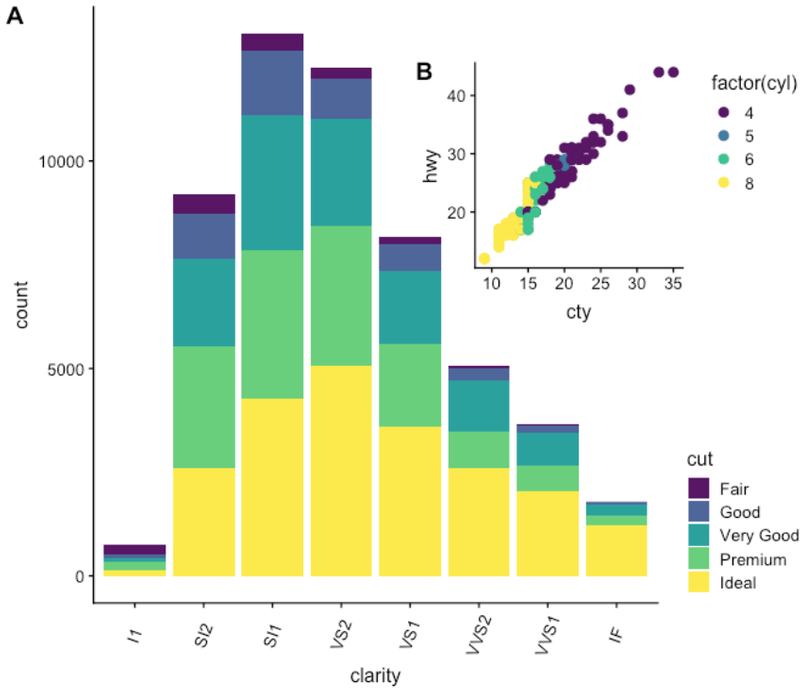
- We will read chapter 5 (only from 5.1 to 5.3).
Optional reading: Chapter 8 (As I will not go through R basics anymore, it is up to you if you want to pimp you figures).



How to visualise your data: ggplot







Why ggplot?

First plot with ggplot2: qplot

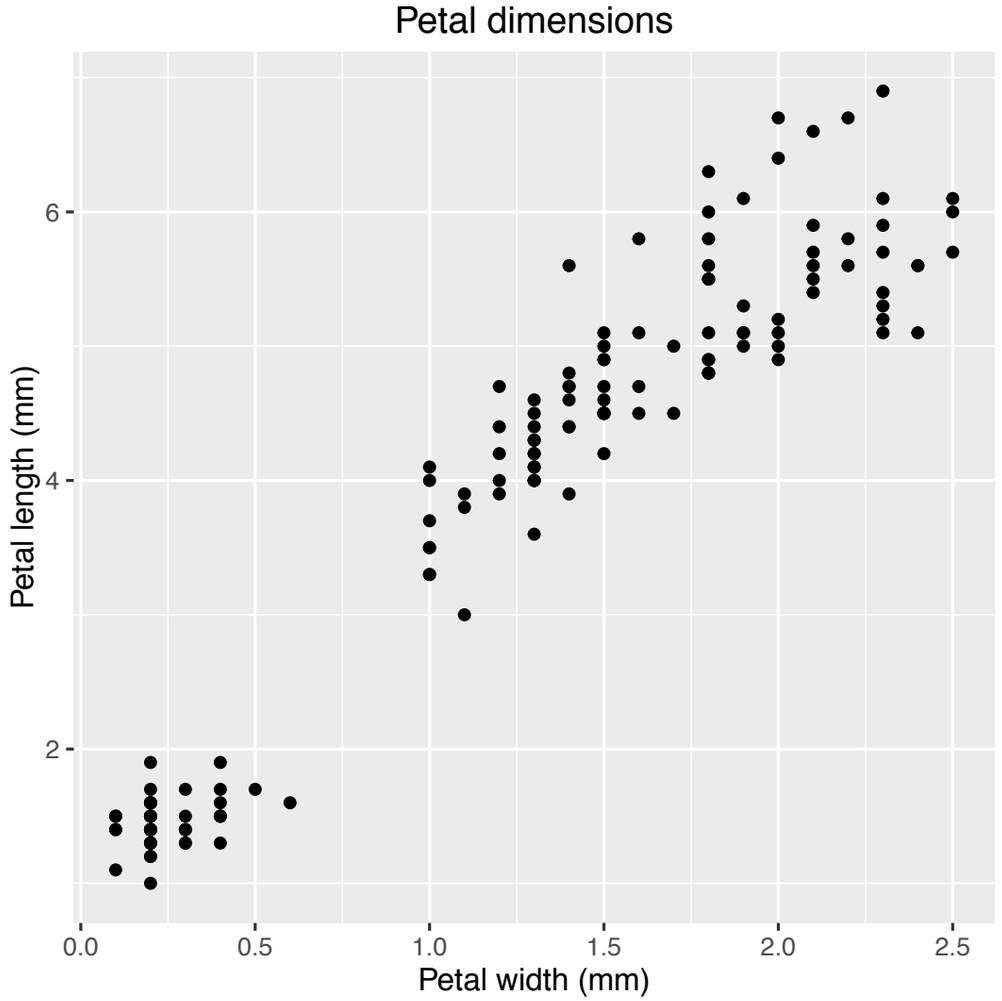
- It works similar to plot().

```
qplot(data = mydata,
y = var_y,
x = var_x, # could be optional
xlab = “x”,
ylab = “y”,
main = “a quick plot”)
```

First plot with ggplot2: qplot

- It works similar to plot().

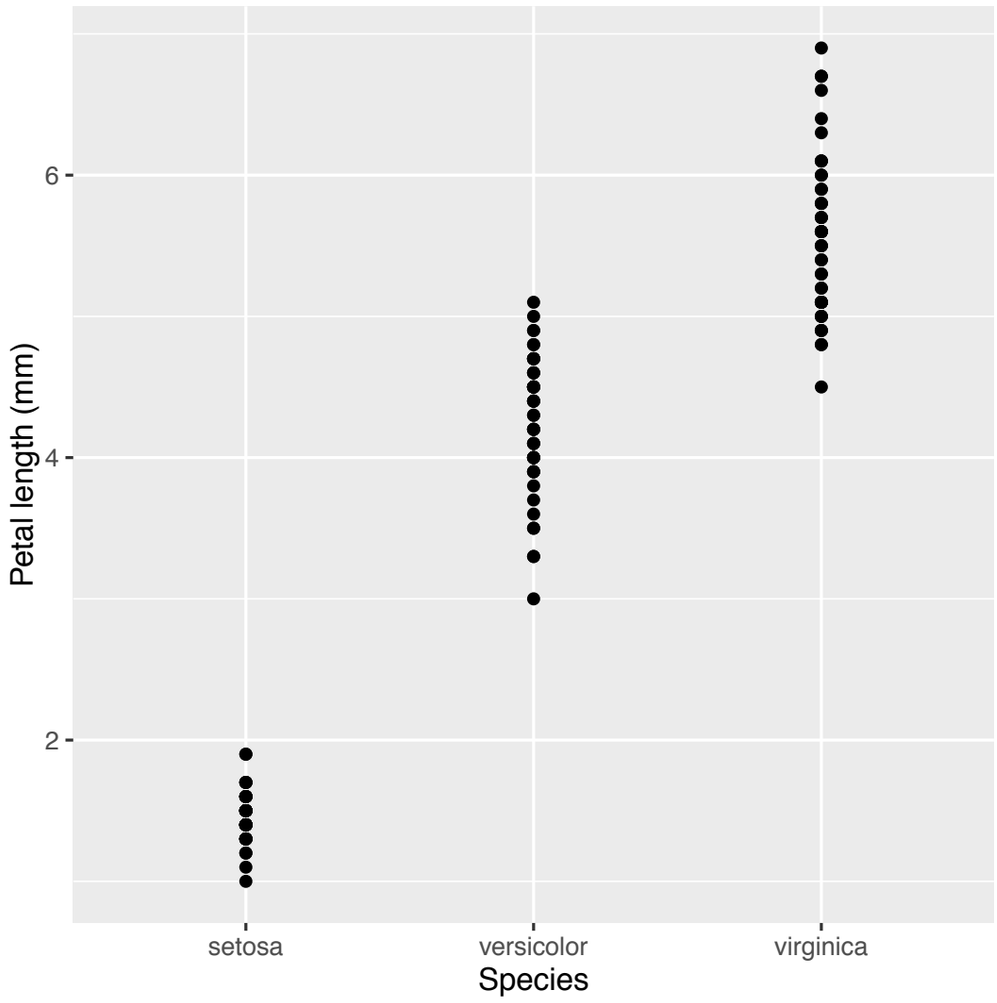
```
qplot(data = iris,
      y = Petal.Length,
      x = Petal.Width,
      xlab = “Petal width (mm)”,
      ylab = “Petal length (mm)”,
      main = “Petal dimensions”)
```



Lets try now with a categorical variable

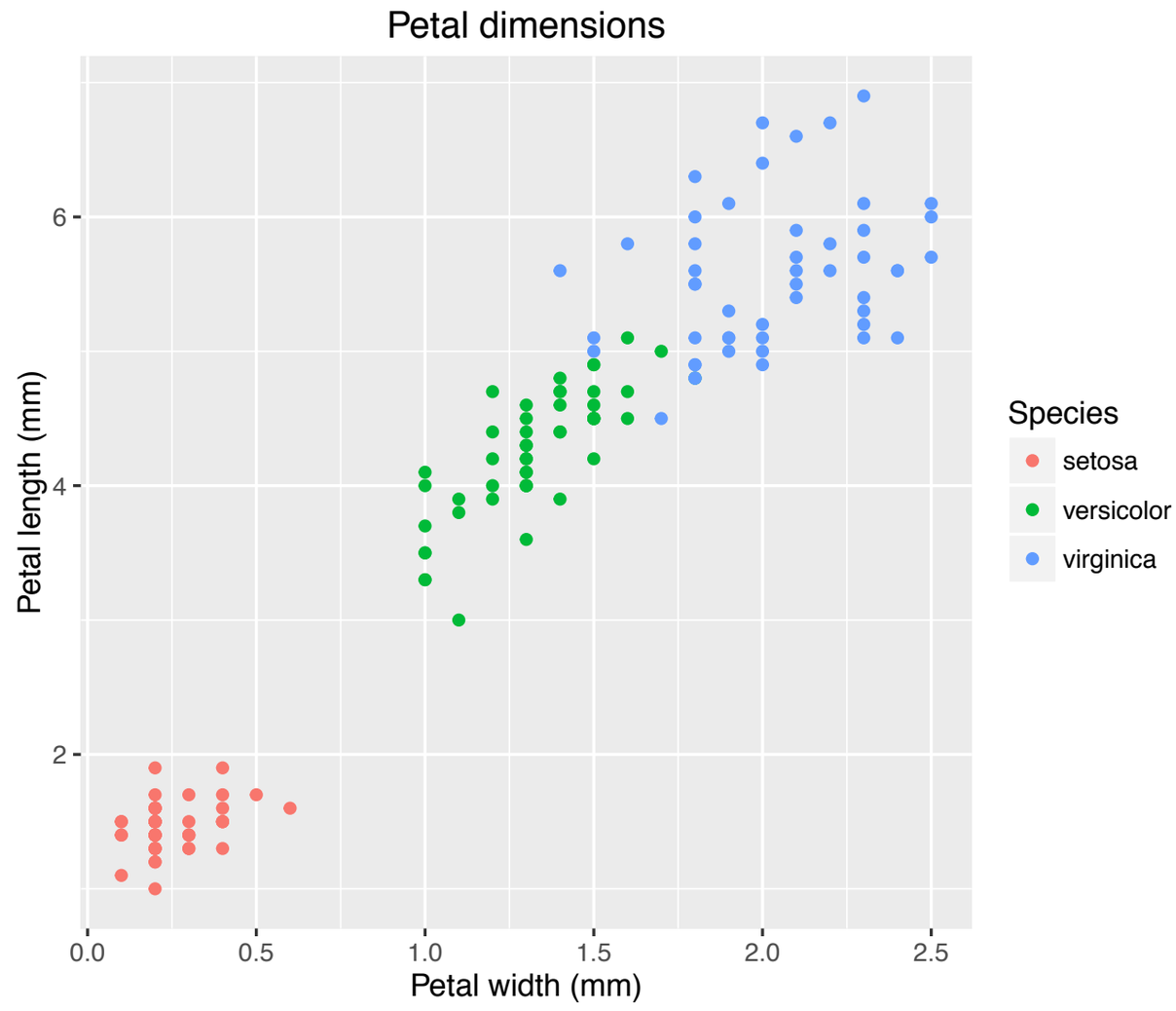
- It works similar to plot().

```
qplot(data = iris,

y = Petal.Length,

x = Species,

xlab = “Species”,

ylab = “Petal length (mm)”)
```



Lets pimp this a bit. (scale it!)

```
qplot(data = iris,

      y = Petal.Length,

      x = Petal.Width,

      xlab = “Petal width (mm)”,

      ylab = “Petal length (mm)”,

      main = “Petal dimensions”,

      colour = Species)
```

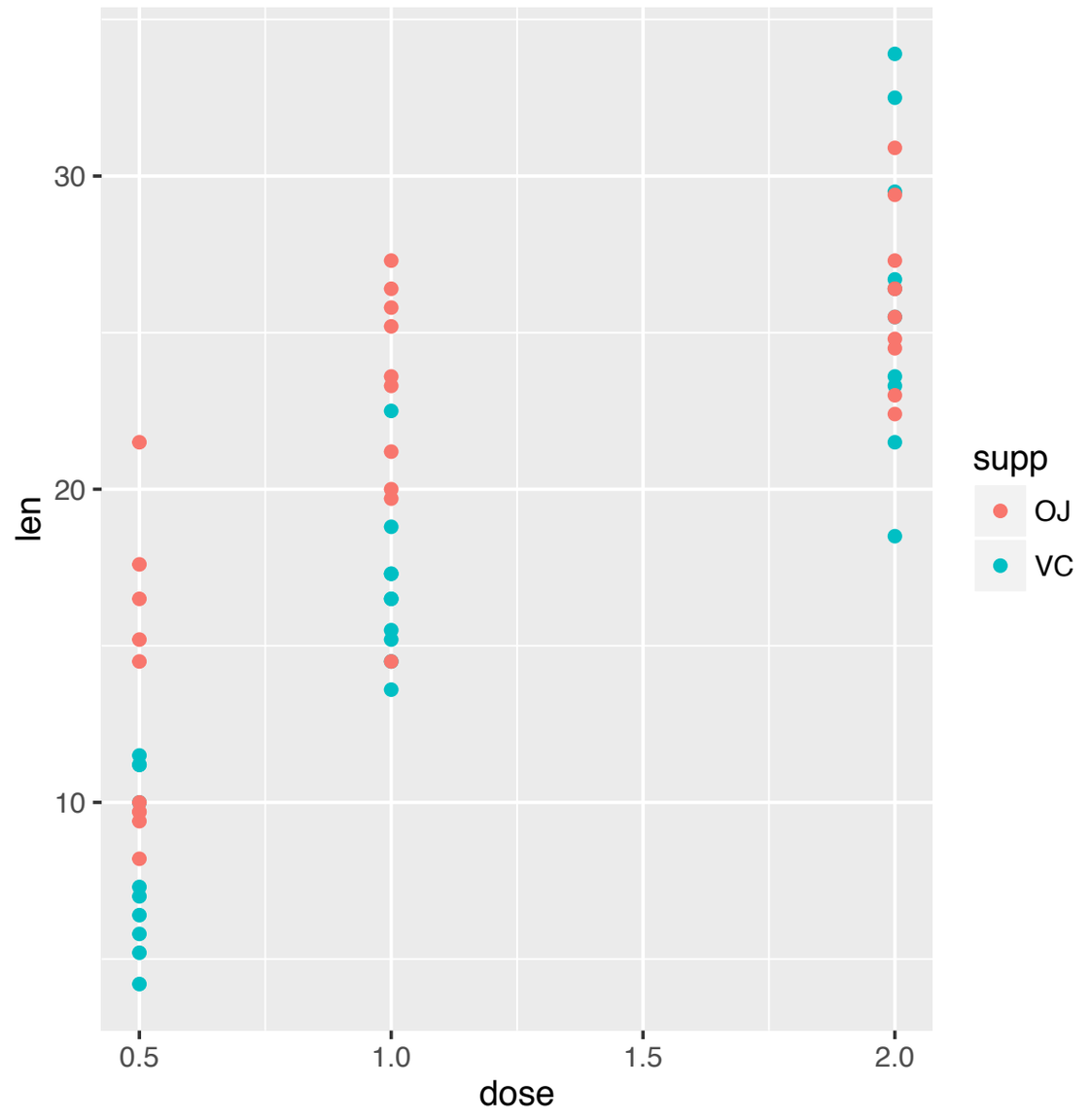


Try it yourself!

- Load the dataset ToothGrowth
  and produce a plot.

Try it yourself!

- Load the dataset ToothGrowth and produce a plot.



Layering in ggplot

Data

Layering in ggplot

Data

Mapping

Layering in ggplot

Data

Mapping

Geometry(ies)

Statistics

Layering in ggplot

Data

Mapping

Geometry(ies)

Statistics

Scales

Theme

How does it translate to R?

1. Create a plot object

```
myplot <- ggplot(data, aes(x,y))
```

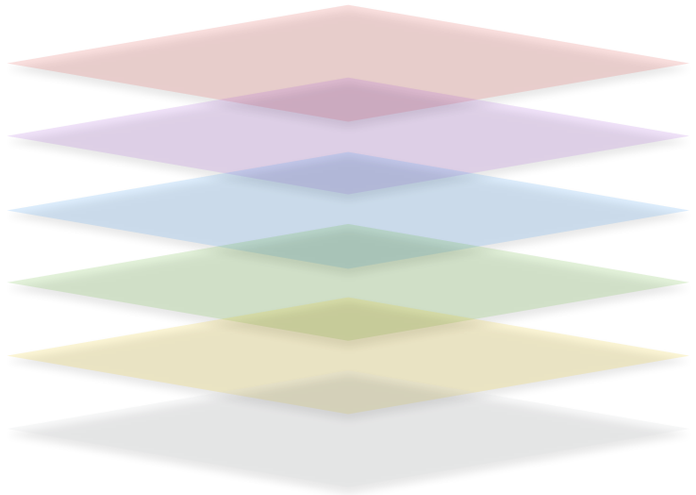
2. Add layers to it

```
myplot <- myplot + layer()
```

3. Repeat step 2 until you have all the layers that you need

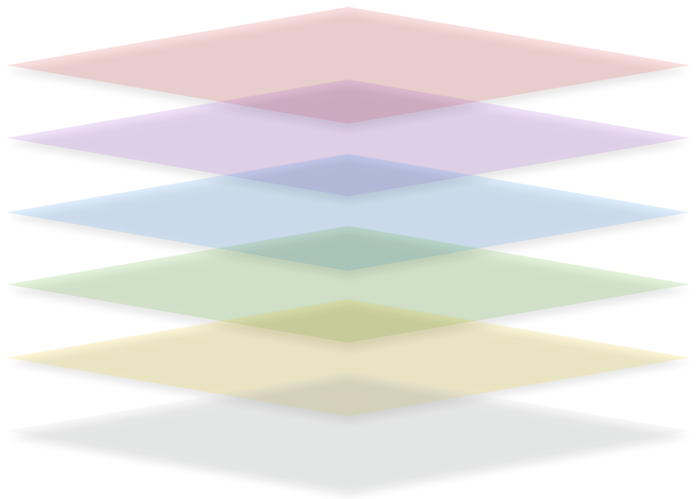
4. Print your plot

```
print(myplot)
```



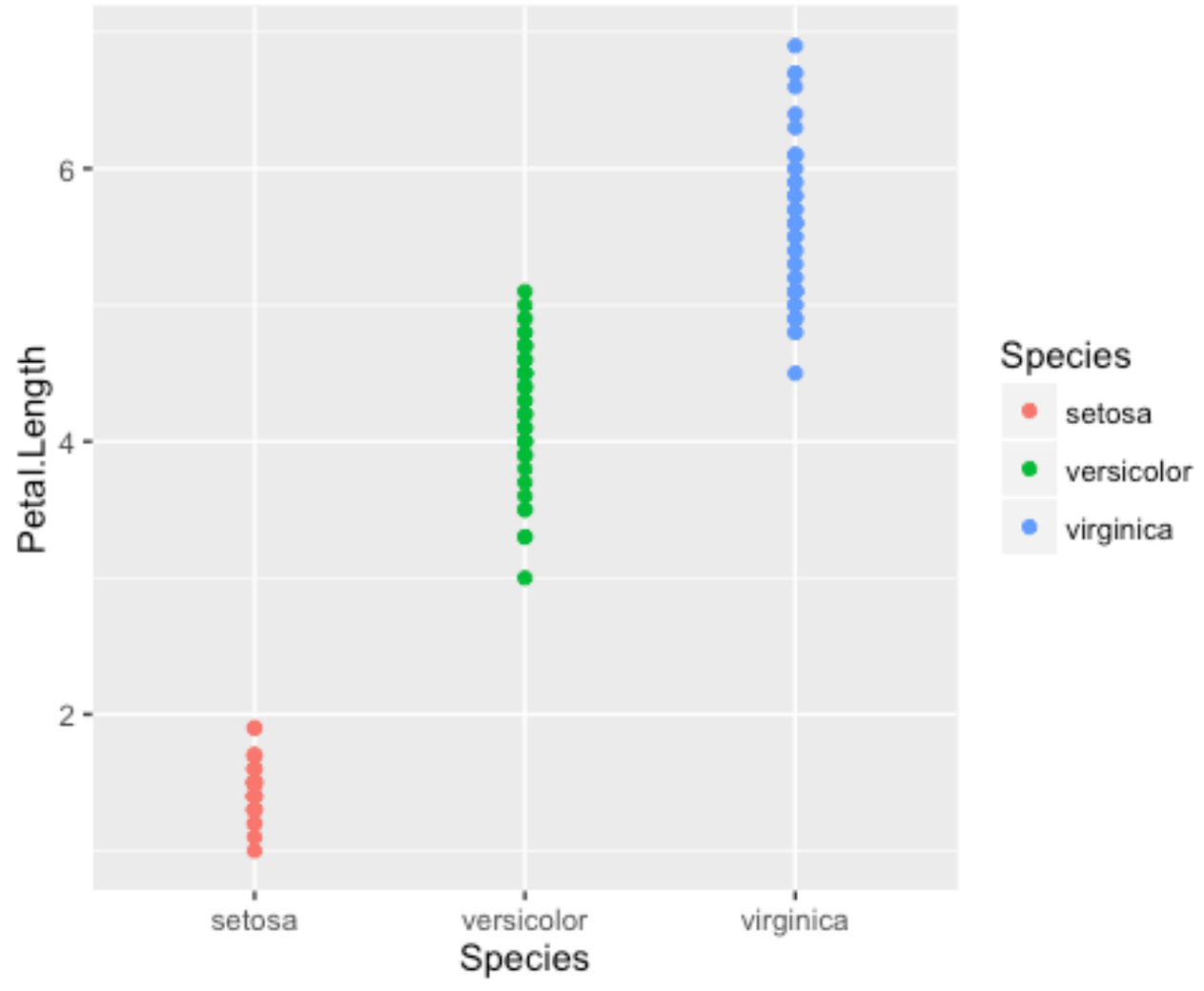
Plotting with ggplot()

```
> p <- ggplot(data = iris,
          aes(y = Petal.Length,
          x = Species,
          colour = Species)

> p + geom_point()
```



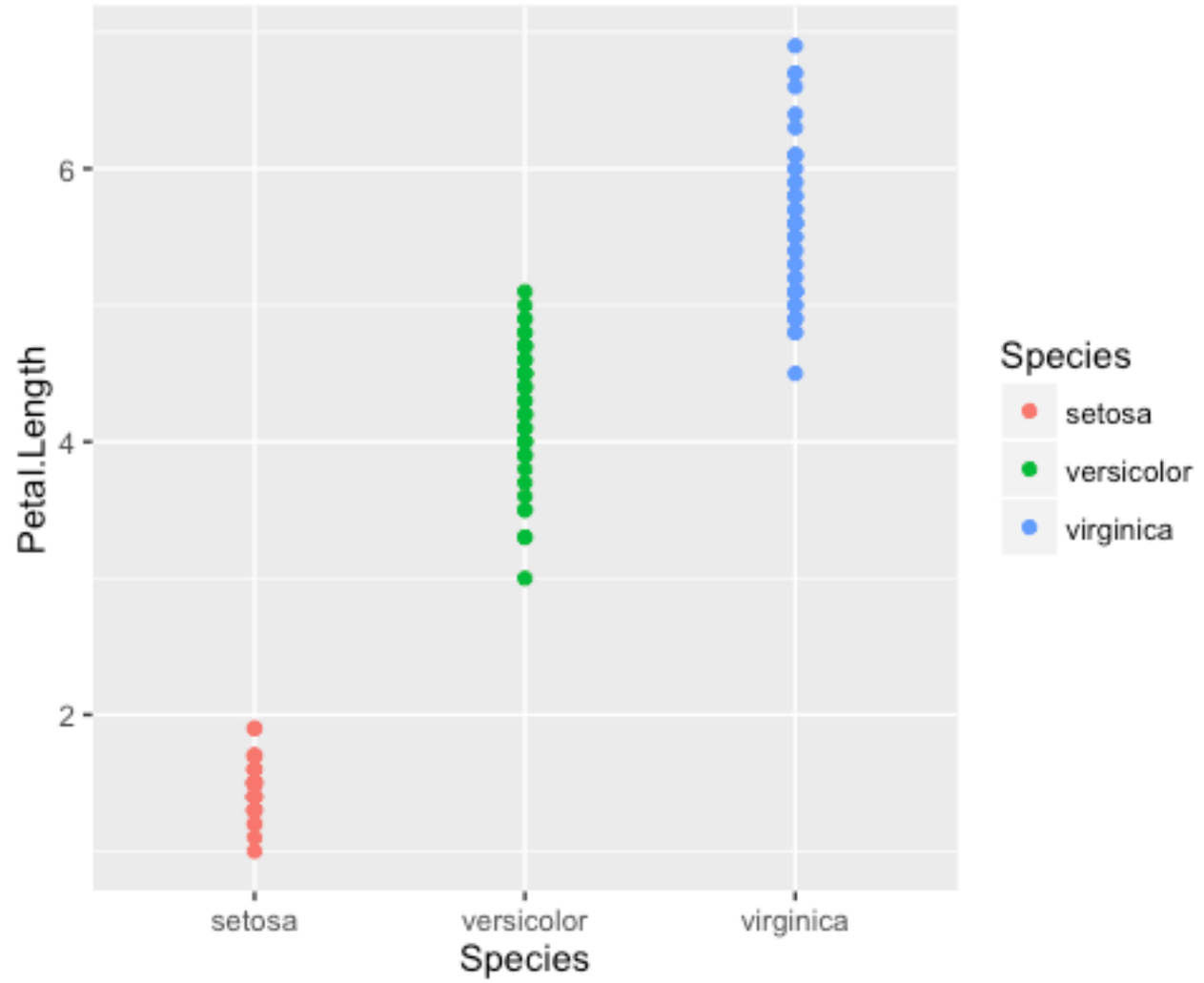
Plotting with ggplot()

```
> p <- ggplot(data = iris,
        aes(y = Petal.Length,
        x = Species,
        colour = Species)
> p + geom_point()
```



Plotting with ggplot()

```
> p <- ggplot(data = iris,
      aes(y = Petal.Length,
      x = Species,
      colour = Species)
> p + geom_point()
```

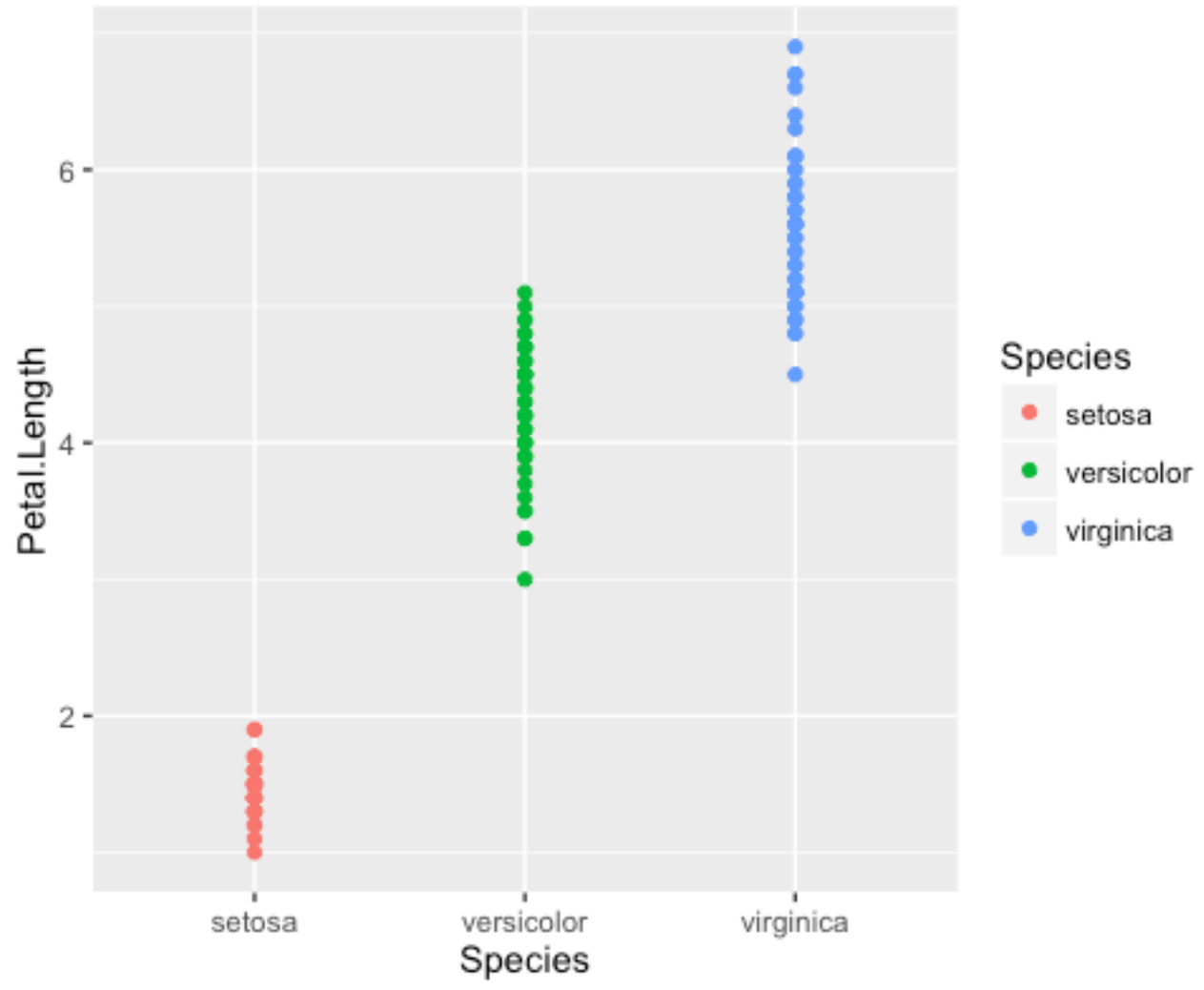


Aesthetics (mapping)

- x, y: positions of your data
- colour: colour of your points (or each point)
- group: assign groups to each point (e.g. treatment)
- shape: the form of the point
- linetype: the type of line to be used
- size: size of point or line
- alpha: transparency of the points

Plotting with ggplot()

```
> p <- ggplot(data = iris,
        aes(y = Petal.Length,
        x = Species,
        colour = Species)
> p + geom_point()
```

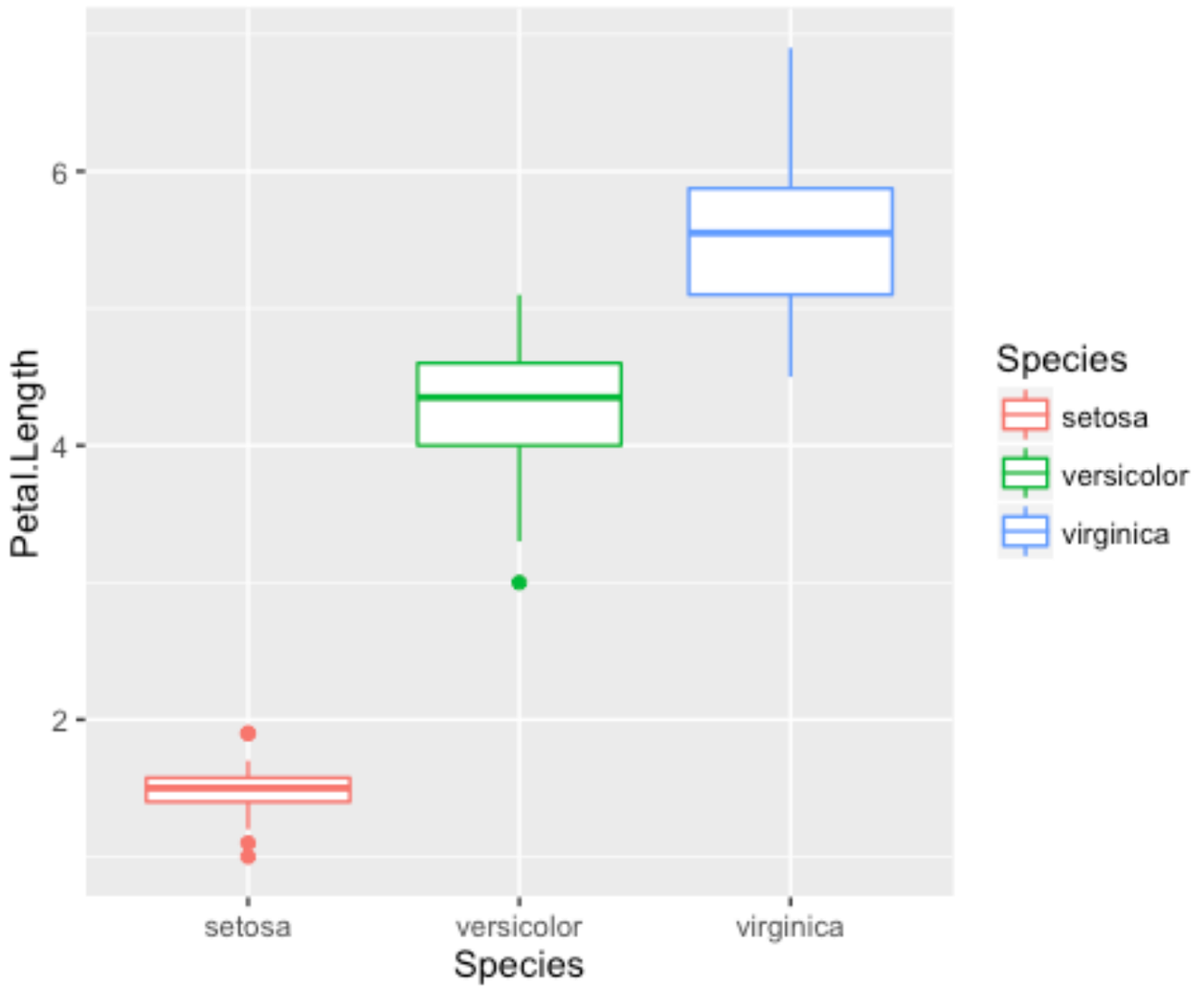


Geometry (geom_XX)

- points: scatterplot
- line: line plot (line increases with x)
- path: line plot, where line follows order of observations
- boxplot: box- and whisker-plots, very useful for categorical data on the y axis
- bar: barplots (but we rarely like those!)
- histogram: histograms (when you have only one variable)

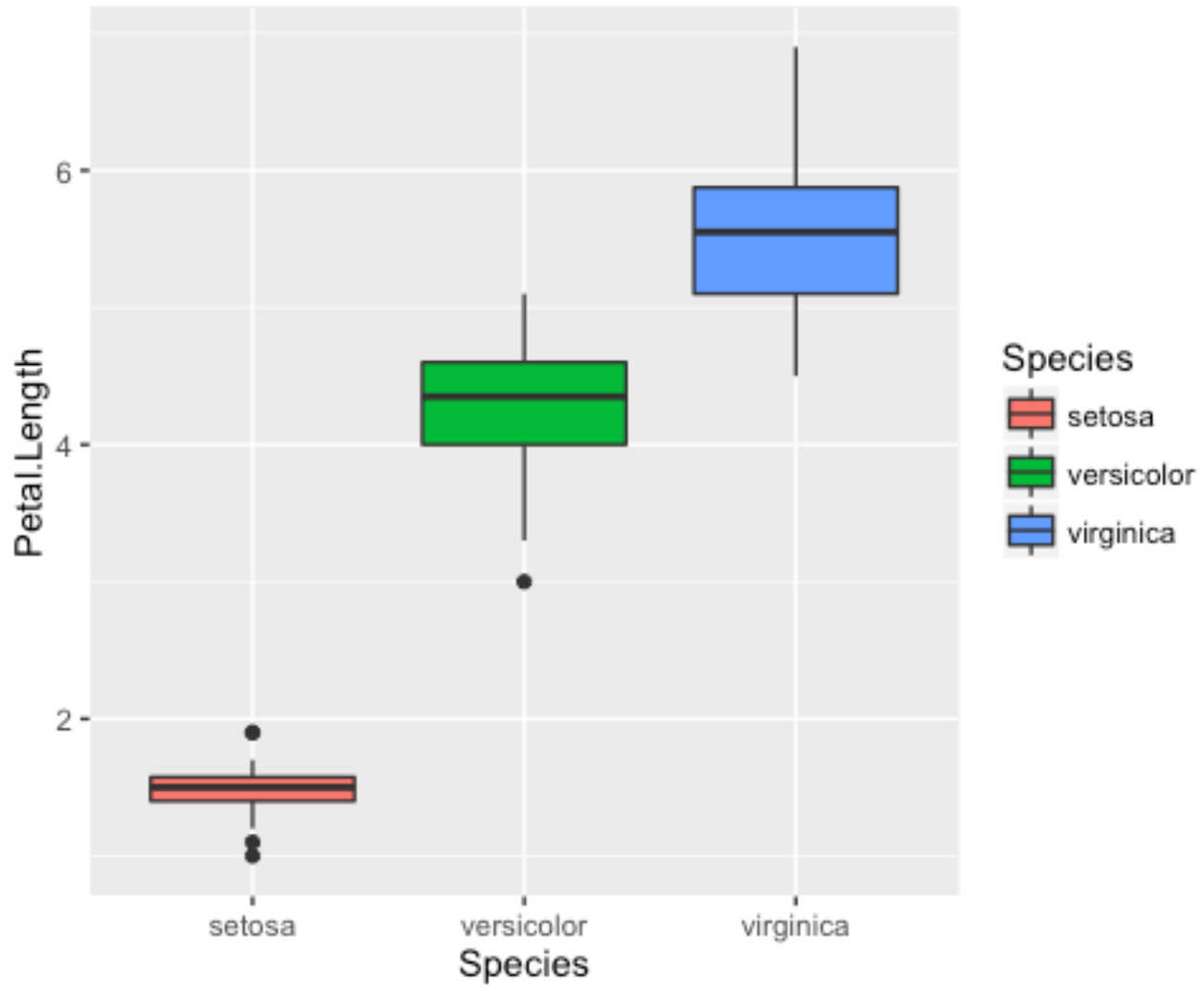
Plotting with ggplot()

```
> p <- ggplot(data = iris,
          aes(y = Petal.Length,
          x = Species,
          colour = Species)
> p + geom_boxplot()
```



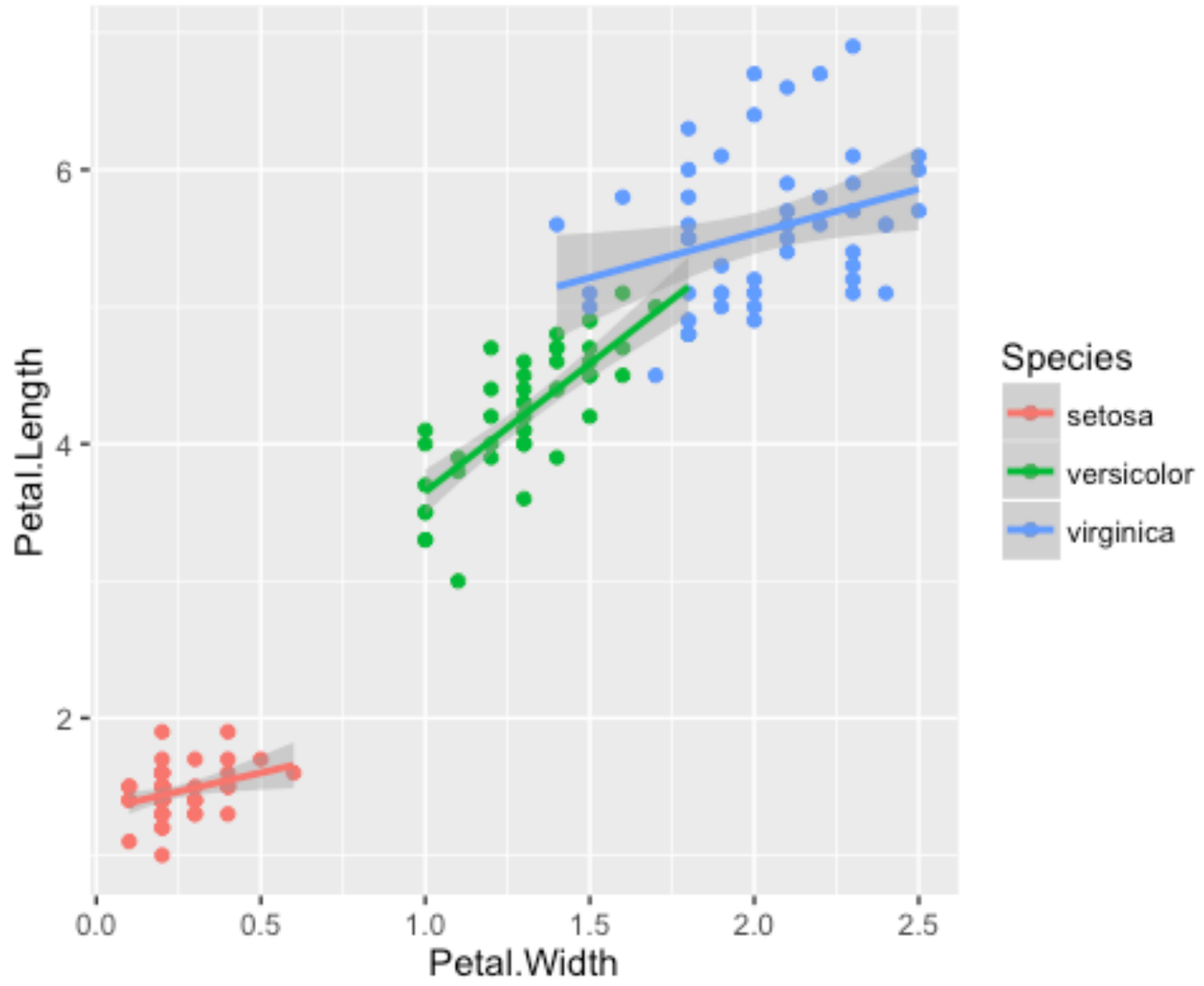
Plotting with ggplot()

```
> p <- ggplot(data = iris,
          aes(y = Petal.Length,
          x = Species,
          fill = Species)

> p + geom_boxplot()
```



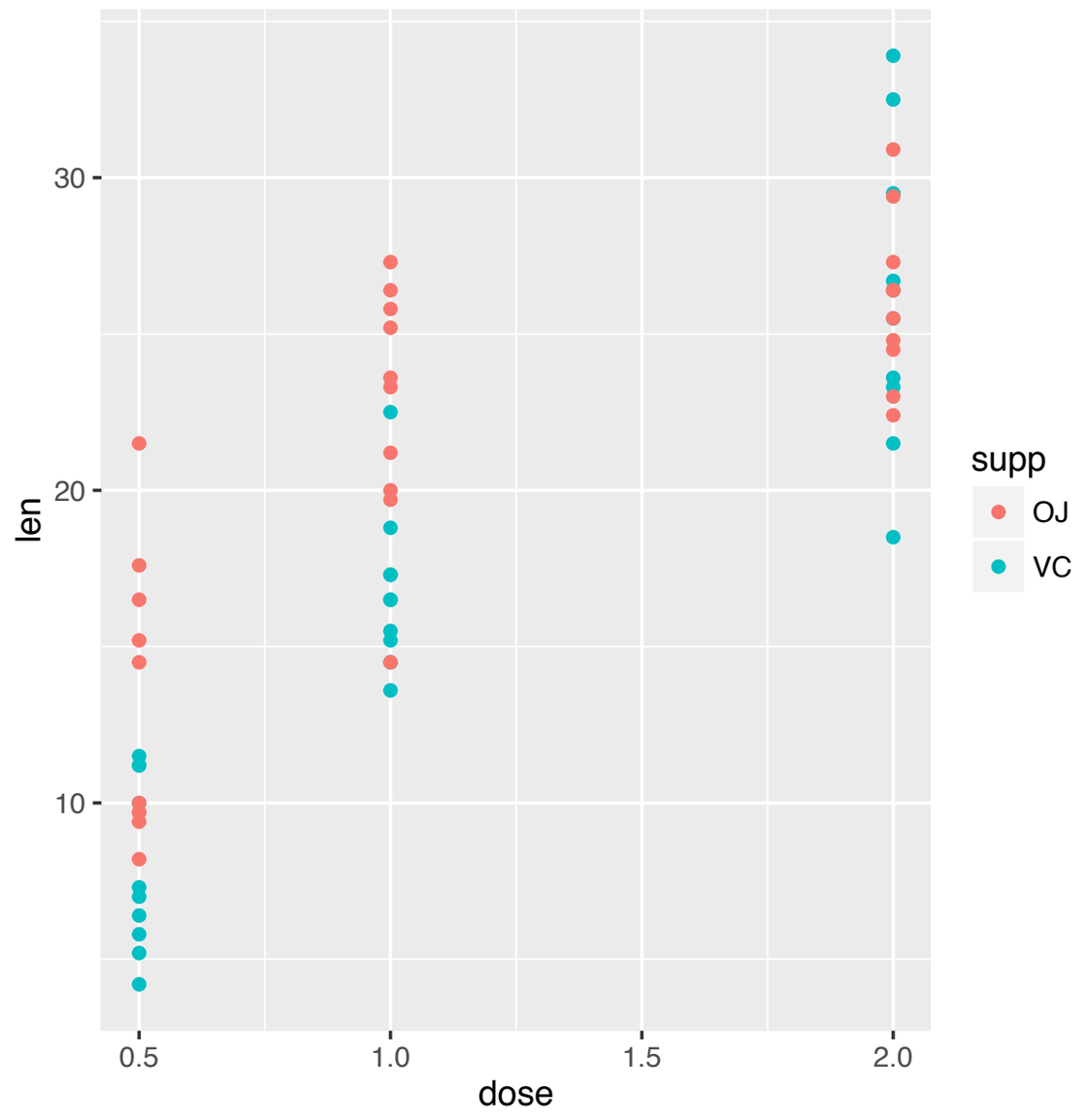
Using statistics

```
> p <- ggplot(data = iris,
        aes(y = Petal.Length,
            x = Petal.Width,
            colour = Species)
> p + geom_point() +
stat_smooth(method = “lm”)
```



Try it yourself with ggplot()!

- Can you make this plot look even better using the command ggplot().
- What layers would you choose and why?

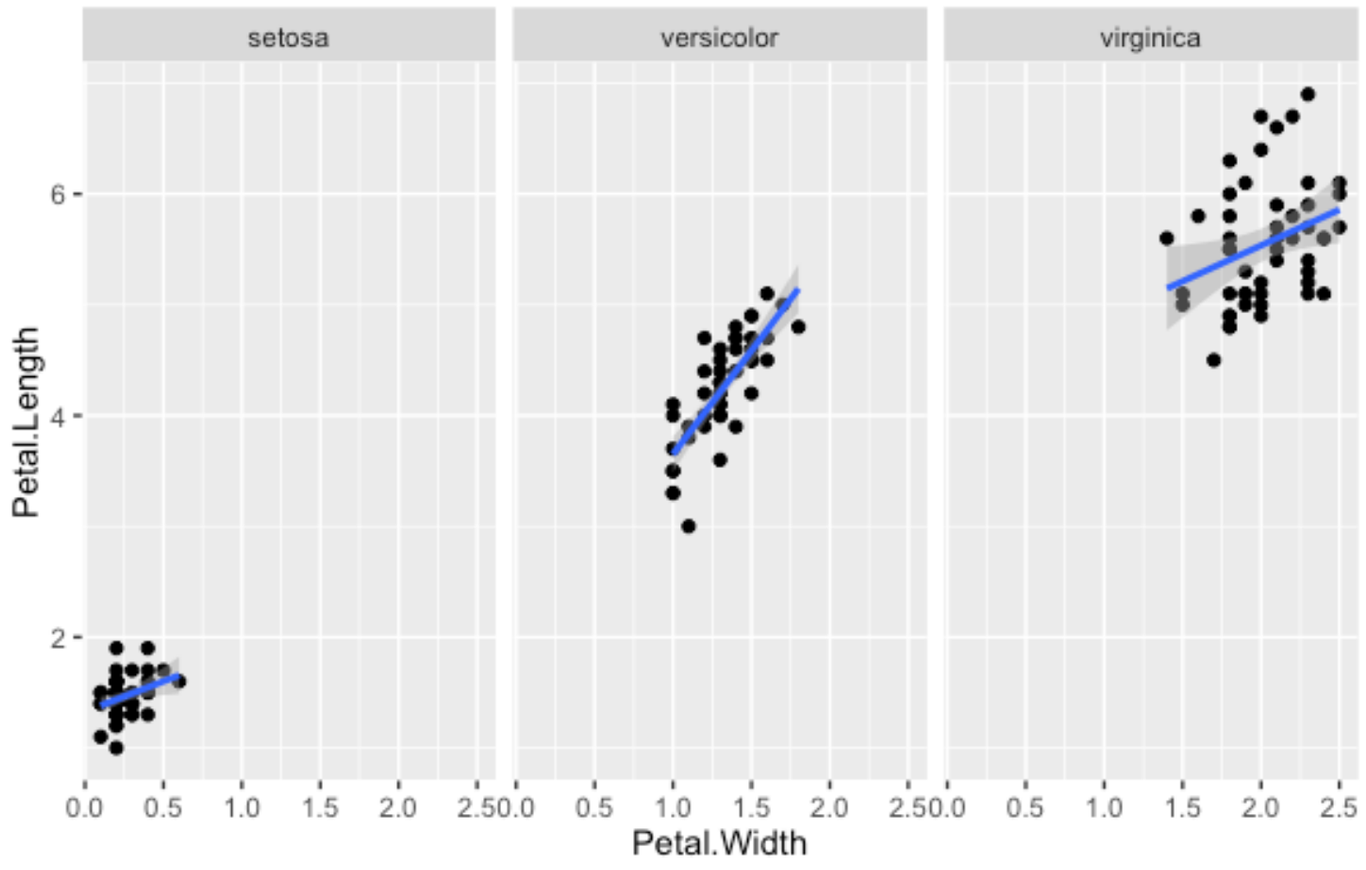


Facets

- Used to create a multipanel plot based on some grouping variable.
- Notation: face_grid(rows~columns)
- If you just want columns, then .~columns, and vice versa.

Let’s use facets to see things better

```
> p <- ggplot(data = iris,

        aes(y = Petal.Length,

        x = Petal.Width)

p + geom_point() +

stat_smooth(method = “lm”) +

face_grid(.~Species)
```



Final retouches: scale_XX

Scales are the equivalent of make-up on your plot:

- Change your axis (e.g. scale_x_continous)
- Labels (e.g. labs)
- Colours (e.g. scale_colour_manual)
- and lot more!

Ok, Enough! I know you are about to get a stroke! (me too)