Plotting in **R**

Liam D. Bailey

Leibniz Institute of Zoo and Wildlife Research

June 2018

Why plot in **R**?

- Powerful (Large range of plot types)
- Customizable (Make your own style)
- Practical (Integrate your plots and your code together)
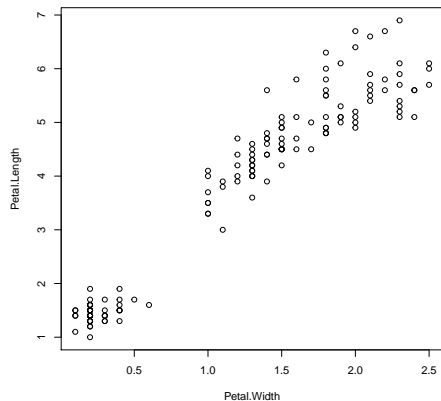
Plotting in **R**

Two ways to plot in **R**
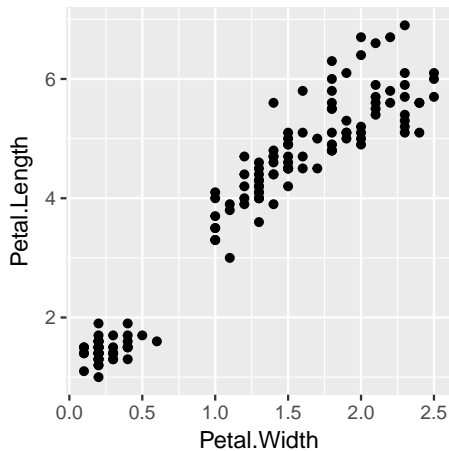
In **R** there are two (main) ways to make plots:

- **base plotting**: The standard way of plotting.
- **tidyverse plotting**: The most common alternative, using package *ggplot2*.

An example

```
plot(Petal.Length ~ Petal.Width, data = iris)
```

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

Plotting in **R**

Plotting in **R**

1. Two ways to plot in **R**

2. Plotting with baseplot
   - Scatter plot
   - Boxplot
   - Histograms
   - Bargraphs
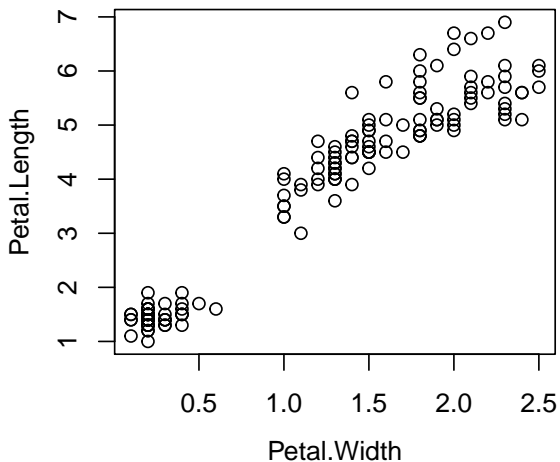   - Aesthetics
   - Saving your plot

3. Plotting with ggplot
   - Scatter plot
   - Boxplot
   - Histograms
   - Bargraphs
   - Aesthetics
   - Saving your plot

Baseplot: Scatter plots

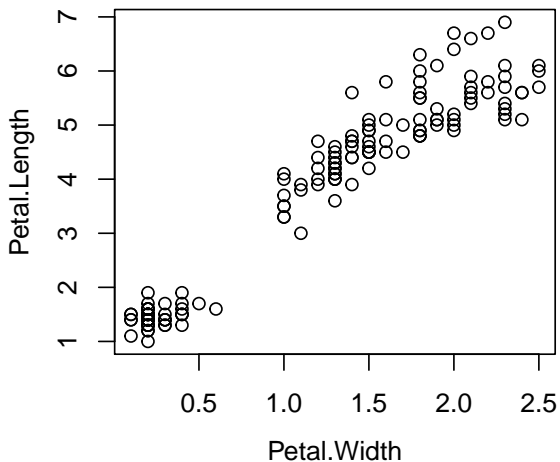In baseplot, you use slightly different functions for different plots.

```
plot(Petal.Length ~ Petal.Width, data = iris)
```

Baseplot: Scatter plots

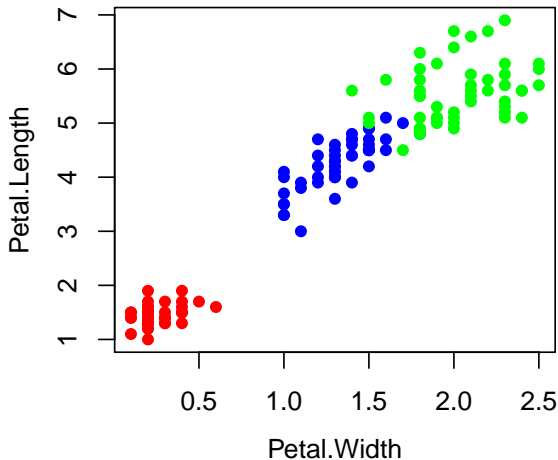You can choose what type of scatter plot to display with argument type.

```
plot(Petal.Length ~ Petal.Width, data = iris)
```

Baseplot: Scatter plots

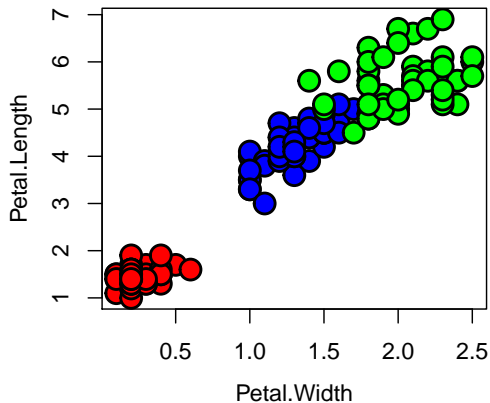You can change point shapes (pch) and colour (col).

```
palette(c("red", "blue", "green"))
plot(Petal.Length ~ Petal.Width, data = iris, pch = 16, col = iris$Species)
```
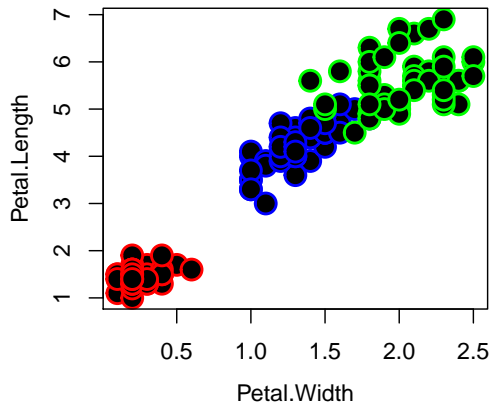
Beware

Be aware that for many elements they have both an outline colour (col) and background colour (bg).

```
palette(c("red", "blue", "green"))
plot(Petal.Length ~ Petal.Width, data = iris, cex = 2,
     pch = 21, bg = iris$Species, col = "black", lwd = 2)
```
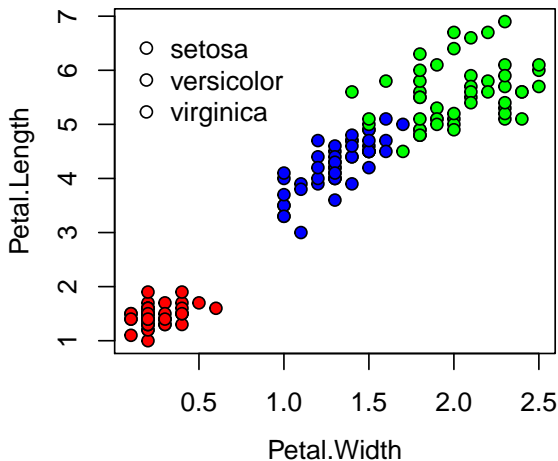
```
palette(c("red", "blue", "green"))
plot(Petal.Length ~ Petal.Width, data = iris, cex = 2,
     pch = 21, col = iris$Species, bg = "black", lwd = 2)
```

Baseplot: Scatter plots

Add a legend to make colours understandable.

```
palette(c("red", "blue", "green"))
plot(Petal.Length ~ Petal.Width, data = iris, pch = 21, bg = iris$Species)
legend(x = 0.05, y = 7, c("setosa", "versicolor", "virginica"), pch = 21, bg = c("red", "blue", "green"), bty = "n")
```
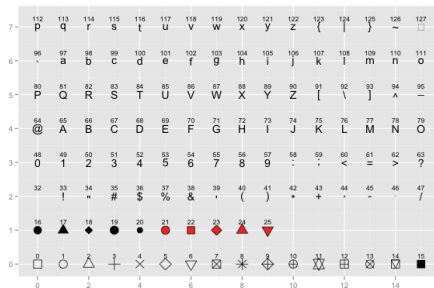
Baseplot: Scatter plots

You can add lines to the plot.

```
plot(Petal.Length ~ Petal.Width, data = iris, pch = 21, bg = iris$Species)
abline(v = 1.3, lty = 2, col = "red", lwd = 2)
abline(h = 4, lty = 3, col = "black", lwd = 2)
abline(a = 1.084, lty = 1, b = 2.23, lwd = 2)
```
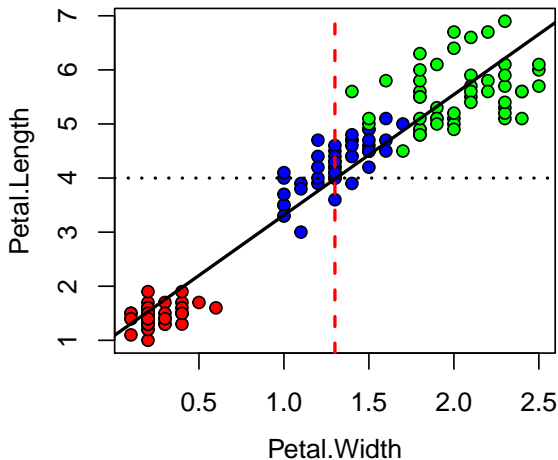
A side note on points and lines

There are standard values for each point and line type. These are **the same** for ggplot and baseplot.
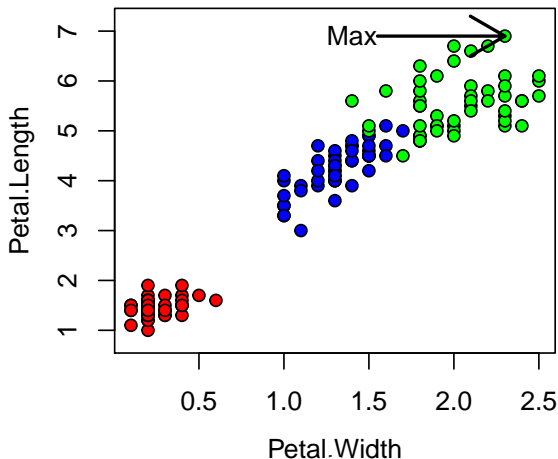
Baseplot: Scatter plots

You can add text and arrows.

```
plot(Petal.Length ~ Petal.Width, data = iris, pch = 21, bg = iris$Species)
max.value <- iris[iris$Petal.Length == max(iris$Petal.Length), ]
arrows(x0 = max.value$Petal.Width - 0.75, y0 = max.value$Petal.Length,
       x1 = max.value$Petal.Width, y1 = max.value$Petal.Length, lwd = 2)
text(x = max.value$Petal.Width - 0.9, y = max.value$Petal.Length, labels = "Max")
```

Plotting in **R**

Baseplot: Boxplot

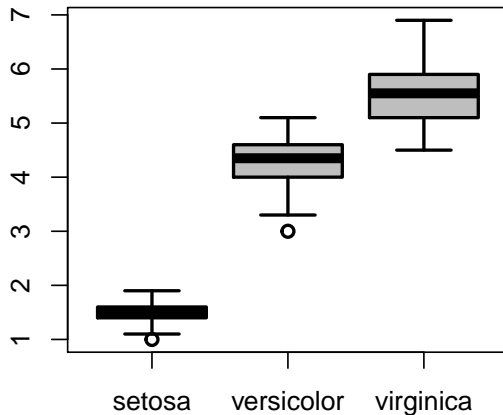Other plot types use different functions.

```
boxplot(Petal.Length ~ Species, data = iris)
```
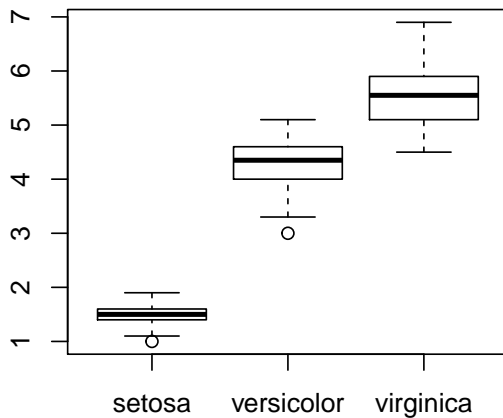
Baseplot: Boxplot

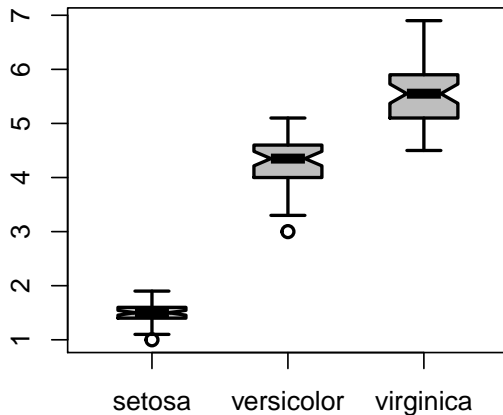Many of the same changes made to scatterplots can be made here.

```
boxplot(Petal.Length ~ Species, data = iris, col = "grey", lwd = 2, lty = 1)
```

Baseplot: Boxplot

There are also some boxplot specific arguments.

```
boxplot(Petal.Length ~ Species, data = iris, col = "grey", lwd = 2, lty = 1,
        width = c(1, 2, 1), notch = TRUE)
```
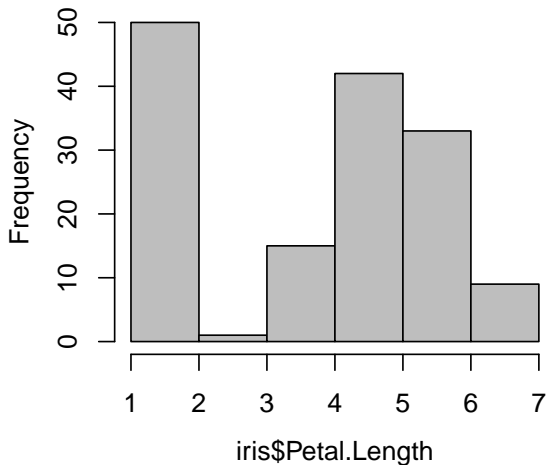
Plotting in **R**

Baseplot: Histogram

```r
hist(iris$Petal.Length, data = iris, main = "")
```
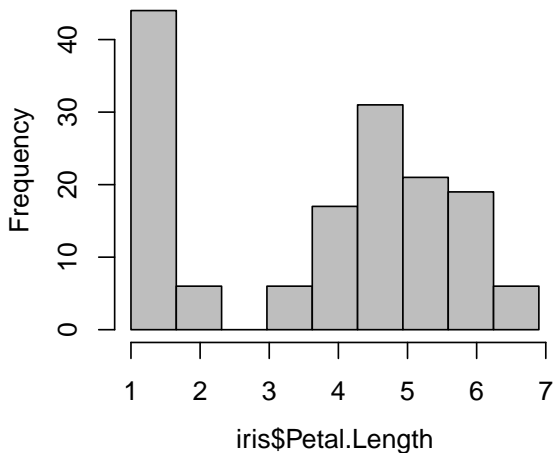
Baseplot: Histogram

```
hist(iris$Petal.Length, data = iris, main = "", breaks = 5, col = "grey")
```

Baseplot: Histogram

```
hist_breaks <- seq(min(iris$Petal.Length), max(iris$Petal.Length), length.out = 10)
hist(iris$Petal.Length, data = iris, main = "", breaks = hist_breaks, col = "grey")
```

Plotting in **R**

Baseplot: Bargraph
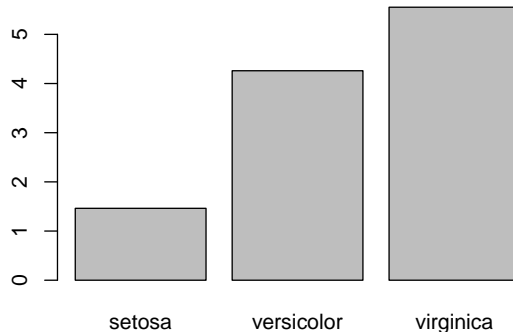
```
spp_means
barplot(height = spp_means$mean, names.arg = spp_means$Species, ylim = c(0, 7),
        xlab = "Species", ylab = "Mean petal length (cm)")
```
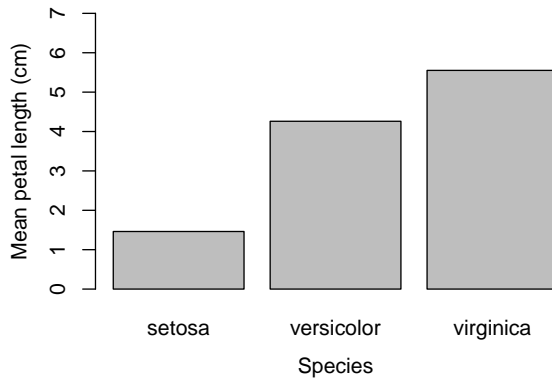
Baseplot: Bargraph

Adding errorbars can be done with the arrows function.

```
spp_means
bar_locations <- barplot(height = spp_means$mean, names.arg = spp_means$Species)
barplot(height = spp_means$mean, names.arg = spp_means$Species, ylim = c(0, 7),
        xlab = "Species", ylab = "Mean petal length (cm)")
arrows(x0 = bar_locations[, 1], x1 = bar_locations[, 1],
       y0 = spp_means$mean - spp_means$SE, y1 = spp_means$mean + spp_means$SE,
       angle = 90, lwd = 2, code = 3)
```

Baseplot: Bargraph
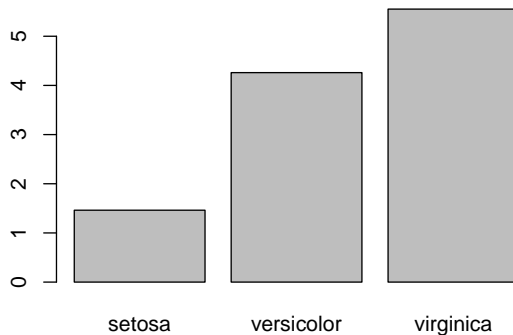
As before, there are similar arguments available.

```
spp_means
bar_locations <- barplot(height = spp_means$mean, names.arg = spp_means$Species)
barplot(height = spp_means$mean, names.arg = spp_means$Species, ylim = c(0, 7),
        xlab = "Species", ylab = "Mean petal length (cm)",
        col = "white", cex.axis = 0.75)
```

Plotting in **R**

Change text

You can size and colour of axis text easily.

```
plot(Petal.Length ~ Petal.Width, data = iris,
     xlab = "Petal Width (cm)", ylab = "Petal Length (cm)",
     cex.lab = 1.5, cex.axis = 1.5, col.axis = "red")
```
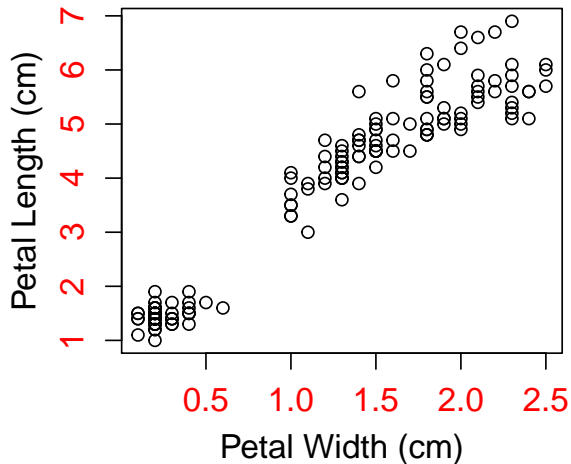
Combining plots

You can easily combine multiple baseplots together by changing the global parameters.
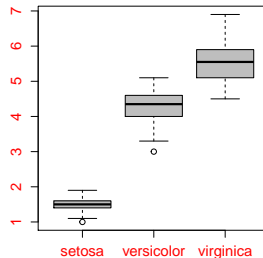
```r
par(mfrow = c(1, 2))
plot(Petal.Length ~ Petal.Width, data = iris,
     xlab = "Petal Width (cm)", ylab = "Petal Length (cm)",
     cex.lab = 1.5, cex.axis = 1.5, col.axis = "red")
boxplot(Petal.Length ~ Species, data = iris, col = "grey")
par(mfrow = c(1, 1))
```

Text family

Global parameters can also be used to change the font family.

```
par(mfrow = c(1, 2), family = "serif")

plot(Petal.Length ~ Petal.Width, data = iris,
     xlab = "Petal Width (cm)", ylab = "Petal Length (cm)",
     cex.lab = 1.5, cex.axis = 1.5, col.axis = "red")
boxplot(Petal.Length ~ Species, data = iris, col = "grey")

par(mfrow = c(1, 1), family = "sans")
```
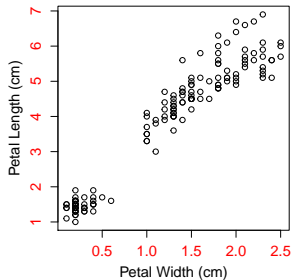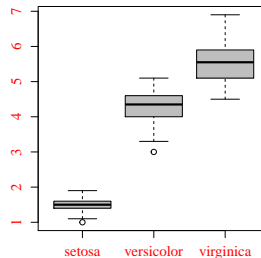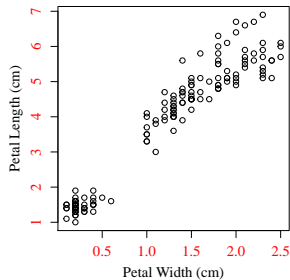
Plot margins

You can change individual plot margins (blue) and outer margins (green).

```
par(mfrow = c(1, 2), mar = c(4, 4, 1, 1),
    oma = c(1.5, 2, 1, 1))
plot(Petal.Length ~ Petal.Width, data = iris, xlab = "Petal Width (cm)", ylab = "Petal Length (cm)",
    cex.lab = 1.5, cex.axis = 1.5, col.axis = "red")
boxplot(Petal.Length ~ Species, data = iris, col = "grey", cex.lab = 0.75, cex.axis = 1.5, col.axis = "red")
```
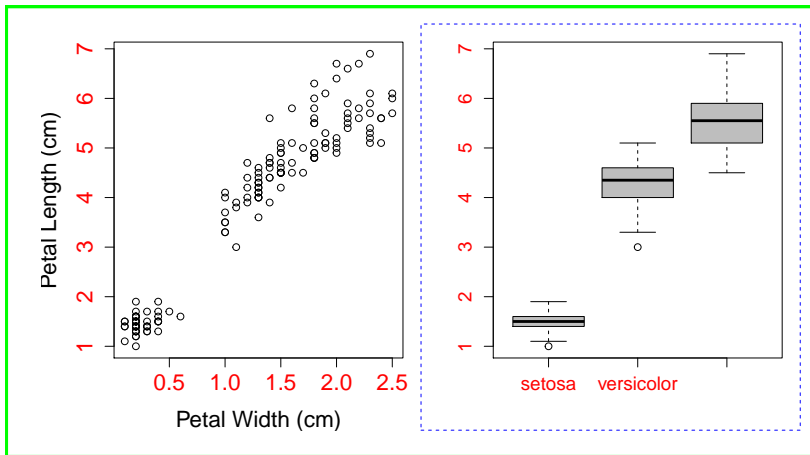
Plotting in **R**

Baseplot: Exporting

```
?pdf ?jpeg ?tiff ?bmp ?postscript
```

```r
pdf("ggplot.pdf", width = 15, height = 5)
plot(Petal.Length ~ Petal.Width, data = iris)
dev.off()
```

Plotting in **R**
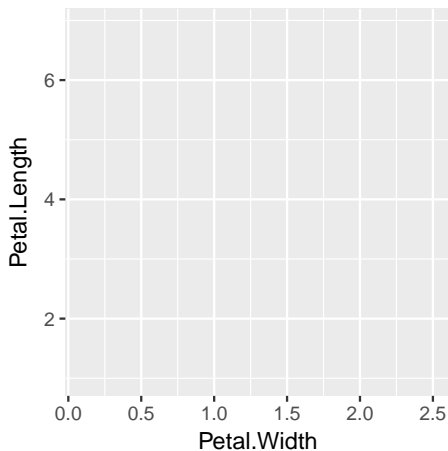
ggplot: An introduction

Unlike baseplot, ggplot works around a single function. We use different functions to add layers onto a ggplot.

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length))
```

Plotting in **R**

ggplot: Scatter plots

We build onto our initial ggplot argument.

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length)) +
  geom_point()
```
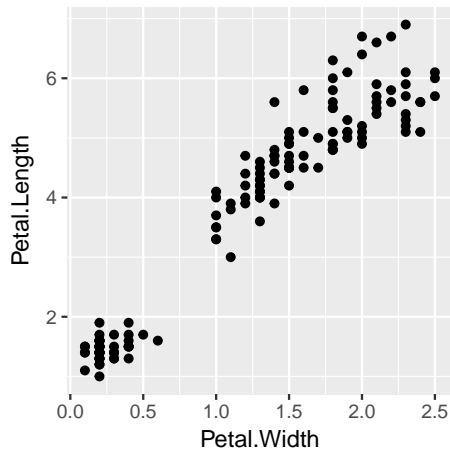
ggplot: Scatter plots

You can change point shapes (shape) and colour (colour). The numbers used here are **the same** as the ones we used above.

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(colour = Species), shape = 16, size = 3)
```
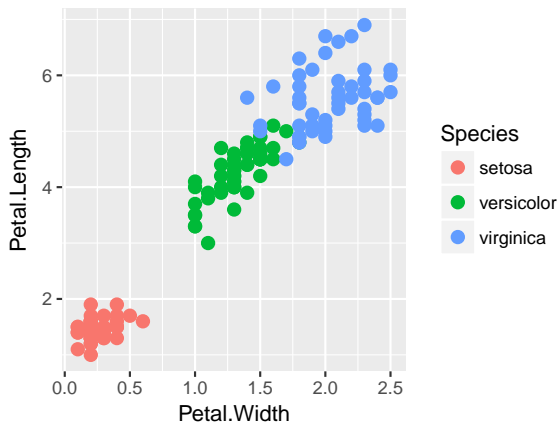
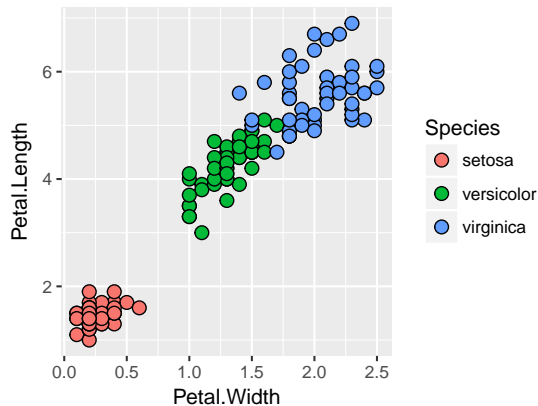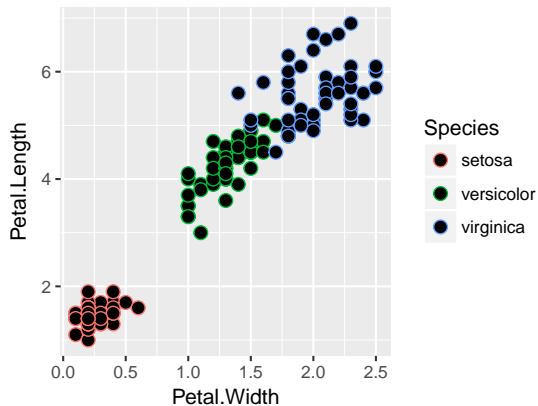## Beware

Again, there is a difference between outline colour (colour) and backgroun colour (fill).

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length))
  geom_point(aes(colour = Species), fill = "black", shape = 21, size
```

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length))
  geom_point(aes(fill = Species), colour = "black", shape = 21, size =
```

ggplot: Scatter plots

In ggplot the legend is included by default. You will need to manuall remove it.

```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(fill = Species), shape = 21, size = 3)+
  theme(legend.position = "none")
```

ggplot: Scatter plots

You can add lines to the plot.
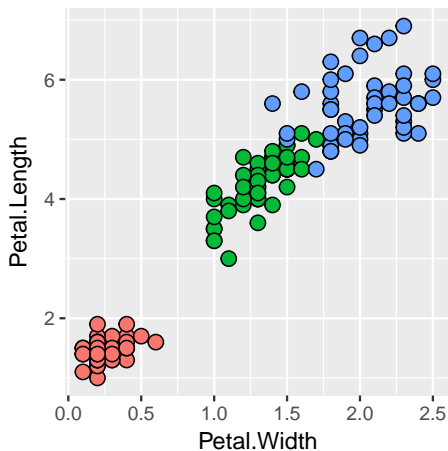
```
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(fill = Species), shape = 21, size = 3)+
  theme(legend.position = "none")+
  geom_hline(yintercept = 4, lty = 3)+
  geom_vline(xintecept = 1.3, lty = 2)+
  geom_abline(intercept = 1.084, slope = 2.23, lty = 1)
```
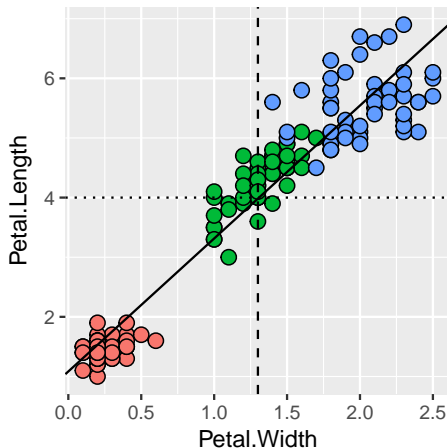
Baseplot: Scatter plots

You can add text and arrows.

```
max.value <- iris[iris$Petal.Length == max(iris$Petal.Length), ]
ggplot(data = iris, mapping = aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(fill = Species), shape = 21, size = 3)+
  theme(legend.position = "none")+
  geom_segment(aes(x = max.value$Petal.Width - 0.75, xend = max.value$Petal.Width,
                 y = max.value$Petal.Length, yend = max.value$Petal.Length), size = 1, arrow = arrow(length = unit(0.5, "cm")))+
  geom_text(aes(x = max.value$Petal.Width - 0.9, y = max.value$Petal.Length, label = "Max"))
```
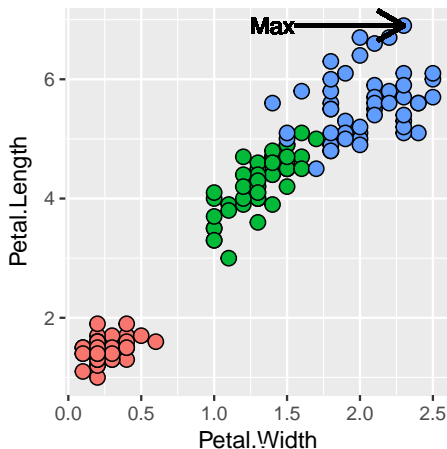
Plotting in **R**

ggplot: Boxplot

We build different plot types in a similar way.

```
ggplot(iris, aes(x = Species, y = Petal.Length))+
  geom_boxplot()
```
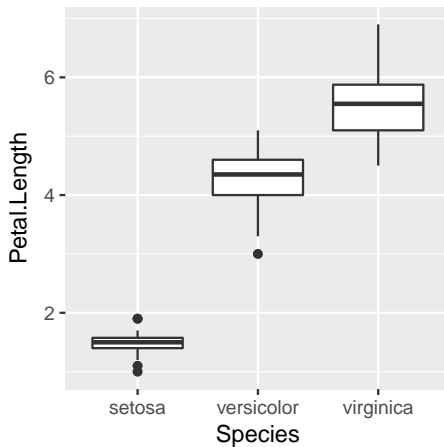
ggplot: Boxplot

```
ggplot(iris, aes(x = Species, y = Petal.Length))+
  geom_boxplot(fill = "grey", size = 2, lty = 1)
```

Plotting in **R**

ggplot: Histogram
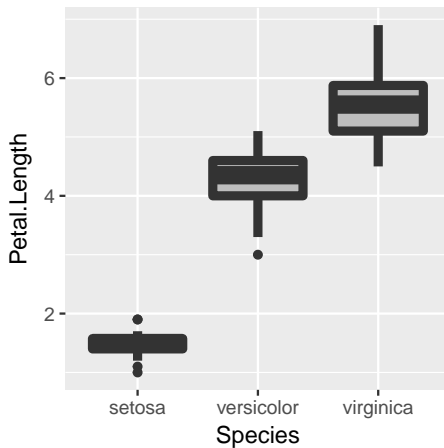
```r
ggplot(iris, aes(x = Petal.Length))+
  geom_histogram()
```

ggplot: Histogram

```
ggplot(iris, aes(x = Petal.Length))+
  geom_histogram(binwidth = 0.5, colour = "black", fill = "grey")
```
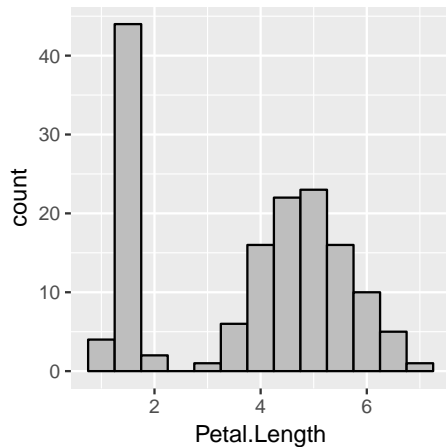
Plotting in **R**

ggplot: Bargraph

```
ggplot(spp_means, aes(x = Species, y = mean))+
  geom_col()
```
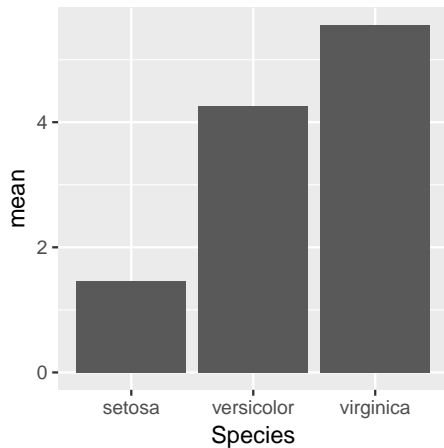
ggplot: Bargraph

Adding errorbars in ggplot is much easier than baseplot.

```
ggplot(spp_means, aes(x = Species, y = mean))+
  geom_col()+
  geom_errorbar(aes(ymin = mean - SE, ymax = mean + SE), size = 1, width = 0.5)
```
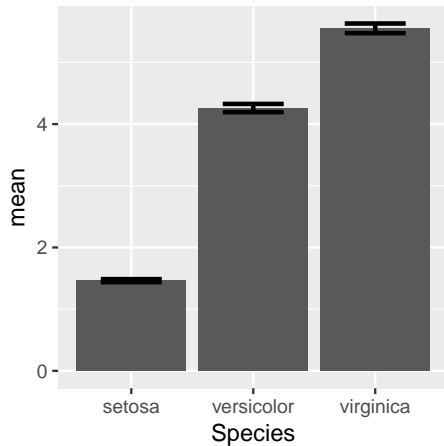
ggplot: Bargraph

```
ggplot(spp_means, aes(x = Species, y = mean))+
  geom_col(fill = "white", colour = "black")
```
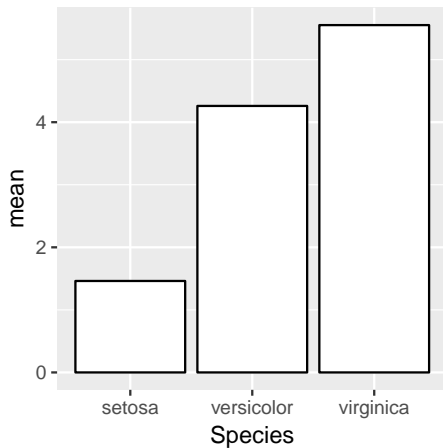
Plotting in **R**

Introduction to themes

In ggplot, you can change aesthetics in individual segments of the code *or* you can change information for the whole plot using theme.

Change text

You can size and colour of axis text easily. Note that text size uses different measurement units.

```
ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  xlab("Petal Width (cm)")+
  ylab("Petal Length (cm)")+
  theme(axis.text = element_text(size = 12, colour = "red"),
        axis.title = element_text(size = 12))
```
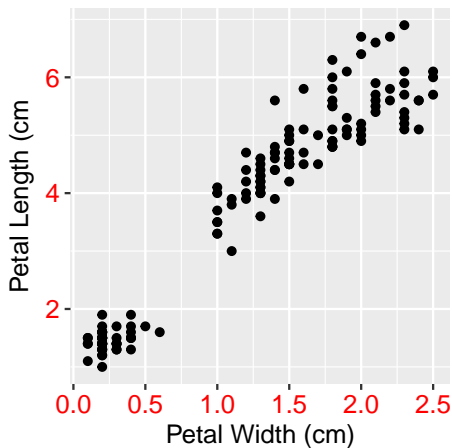
Combining plots

Combining plots it much less straightforward in ggplot. You need to use an additional package. Here we will use gridExtra.

```r
scatter <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  xlab("Petal Width (cm)")+
  ylab("Petal Length (cm)")+
  theme(axis.text = element_text(size = 12, colour = "red"), axis.title = element_text(size = 12))
box      <- ggplot(iris, aes(x = Species, y = Petal.Length))+
  geom_boxplot()+
  theme(axis.text = element_text(size = 12, colour = "red"), axis.title = element_text(size = 12))
gridExtra::grid.arrange(scatter, box, nrow = 1)
```

Text family

Unlike baseplot, in ggplot you can change font family of individual elements in theme.
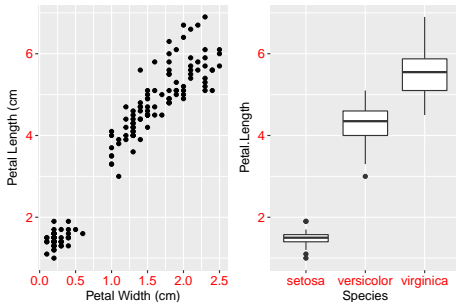
```
scatter <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  xlab("Petal Width (cm)")+
  ylab("Petal Length (cm)")+
  theme(axis.text = element_text(size = 12, colour = "red", family = "serif"), axis.title = element_text(size = 12, family = "mono"))
box     <- ggplot(iris, aes(x = Species, y = Petal.Length))+
  geom_boxplot()+
  theme(axis.text = element_text(size = 12, colour = "red", family = "sans"), axis.title = element_text(size = 12, family = "sans"))
gridExtra::grid.arrange(scatter, box, nrow = 1)
```

Plot margins

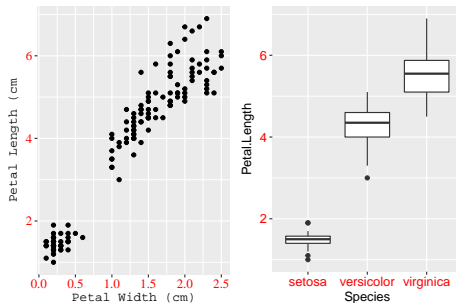Plot margins are also controlled in theme of each plot individually.

```
scatter <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  xlab("Petal Width (cm)")+
  ylab("Petal Length (cm)")+
  theme(axis.text = element_text(size = 12, colour = "red"), axis.title = element_text(size = 12,
        plot.margin = unit(c(4, 4, 1, 1), "mm"))
box     <- ggplot(iris, aes(x = Species, y = Petal.Length))+
  geom_boxplot()+
  theme(axis.text = element_text(size = 12, colour = "red"), axis.title = element_text(size = 12))
gridExtra::grid.arrange(scatter, box, nrow = 1)
```
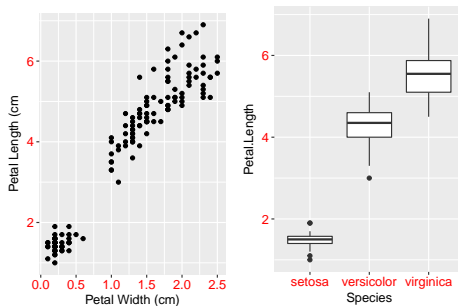
Preset themes

ggplot also has a number of preset themes that you can use.

```
grey <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  theme_grey()
bw <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  theme_bw()
classic <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  theme_classic()
gridExtra::grid.arrange(grey, bw, classic, nrow = 1)
```
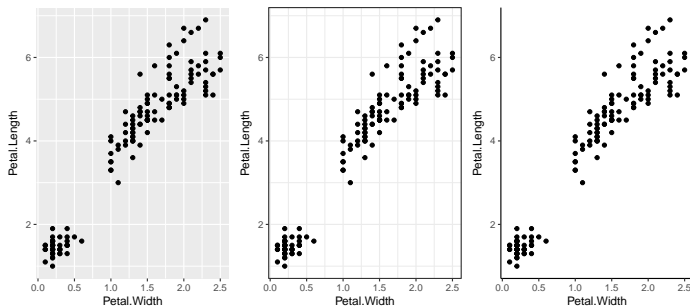
Plotting in **R**

1. Two ways to plot in **R**

2. Plotting with baseplot
   - Scatter plot
   - Boxplot
   - Histograms
   - Bargraphs
   - Aesthetics
   - Saving your plot

3. Plotting with ggplot
   - Scatter plot
   - Boxplot
   - Histograms
   - Bargraphs
   - Aesthetics
   - Saving your plot

ggplot: Exporting

```
?ggsave
```

```
classic <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length))+
  geom_point()+
  theme_classic()
ggsave("ggplot.pdf", plot = classic, width = 15, height = 5)
```