

MAS115: R programming

Lecture 4: Plotting

Bryony Moody

March 2, 2021

Aims

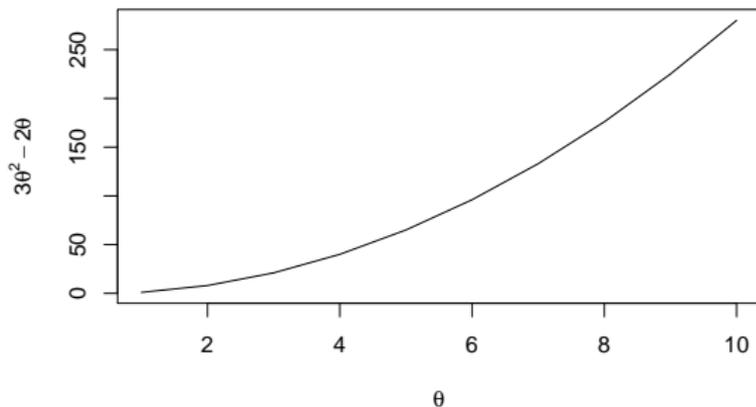
How to use the extensive set of graphics functions in R:

- ▶ The `plot()` function
- ▶ Adding information to the plot
- ▶ Using plots well

Plotting in R

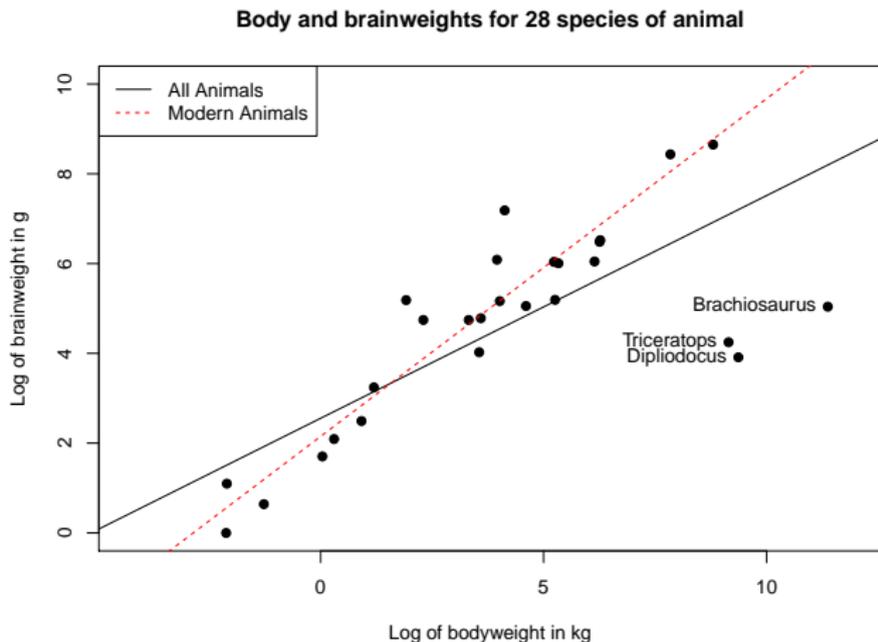
- ▶ Plotting is one of the great strengths of *R*
- ▶ The workhorse plotcommand in *R* is `plot()`. It is used as follows

```
x <- 1:10
y <- 3*x^2 - 2*x
plot(x,y, pch = 19)
```
- ▶ Can then build up the plot with new lines/points/legend



Plotting in R

From these basic commands you can quickly create something really quite complicated



When to use plots

- ▶ Plots can sometimes represent ideas or data more easily than words
- ▶ Visual illustrations can be used alongside text to help the reader understand new concepts
- ▶ Can represent conclusions from data analysis
- ▶ Appearance matters (pch, asp, ...)
- ▶ Be selective about the plot(s) to use

How to include them in a report

- ▶ Every plot in a report must add something to understanding
- ▶ It must be referenced in the text
- ▶ Use a caption in \LaTeX —perhaps self-contained
- ▶ Consider appendices for 'reference' information

Interacting with plots—identify

The `identify` function allows you to ‘interrogate’ a plot—typically a scatter plot or similar.

By clicking on (or ‘near’) a point, you can learn which element it represents in the vectors that defined its co-ordinates.

Note: finish by using `Esc`.

In basic usage, the indices of the points are returned once you have finished. You can use them to e.g. exclude outliers or special cases.

We’ll see later that more advanced use is possible, by using the returned indices to take actions automatically, or by requesting more detailed output from the function.

Interacting with plots—locator

The `locator` function similarly allows you to record specific locations on a plot, using the mouse.

This enables you to, for instance, ‘draw’ on a plot in a simple way with the mouse. Again, finish by using `Esc`.

The value returned by the function is a *list*, which we’ll discuss in detail next week. For now, you can save the output as for any function, and then use `$x` and `$y` to get vectors of co-ordinates.

```
my_points <- locator()  
(click on some points)  
my_points$x  
my_points$y
```