Prelude: R as a calculator

R works by reading your commands from the console (or from your script) and executing them.

In RStudio, you can type commands directly into the console (bottom left), but it is often better to work in a script file (File->New) that you can save. In a script file, press control-enter (or click run) to send the current line or selection to R.

In a trivial way, R can be used as a calculator:

```
2+2
```

```
## [1] 4
```

```
(3*4) / 9
```

```
## [1] 1.333333
```

In the example above, the lines in a (very light) grey box such as `2+2` are the input for R. You can type that into the R console to follow along. The white box below it, with lines starting with `##`, is the R output from these lines.

Storing results as variables

Results from a calculation, or in fact of all operations, can be stored in a variable. In RStudio, you will see the variable appearing in the environment pane (top right), and the variable can be used in further commands:

```
x = 2+2
y = 3 * x
y / 10
```

```
## [1] 1.2
```

Note: You can use the Clear button to clear the current environment. This is often a good idea to make sure that there is no old data lying around that can cause confusion, but you should obviously not do that if there is any important data that you haven't saved yet.

Functions

Most R functionality is exposed through /functions/. Functions take one or more arguments and return a value based on these arguments. For example, you can use the `sum` function to sum a number of values, or use the `mean` function to compute a mean

```
s = sum(1, 2, 3)
m = mean(3, 4, 5)
```

This is a pattern that you will see quite a lot: /result = function(arguments)/. Use a function to calculate some sort of result, and store that as a new variable.

For any function, you can type `?sum` to get the help for that function, or type the function name into the search box in the Help pane in RStudio (bottom right).

Vectors and Data Frames

In R, all variables have a data type. The most common types are vectors and data frames. A vector is a collection of numbers (or texts items), and in fact every number in R is really a vector (of length 1):

```
x = 3
length(x)
```

```
## [1] 1
```

Using the `c()` function, you can create a vector of multiple numbers:

```
x = c(1, 2, 3)
length(x)
```

```
## [1] 3
```

The second important data type is the data frame. This is most similar to the data sets in programs like SPSS. A data frame consists of multiple columns, where each column is a vector again. As will be demonstrated below, you can create a data frame by e.g. reading a CSV file or SPSS file, but you can also create them directly from two (or more) vectors:

```
subject = c(1, 2, 3, 4)
iq = c(108, 97, 112, 101)
d = data.frame(subject, iq)
d
```

```
##   subject  iq
## 1       1 108
## 2       2  97
## 3       3 112
## 4       4 101
```

You can then access the individual members of the data frame using a dollar sign, and the result is a vector:

```
x = d$iq
length(x)
x
```