

# Statistical Modeling I

## Introduction to R

### Introduction to R

Most of you have used R before but a few may not.

R is a language for doing statistical work. It is open source so free to download. We are going to use Rstudio which gives a better interface to R. Rstudio is also open source so you can download it to try using it at home.

1. Log on to the computer and open Rstudio (not Rstudio Geography).
2. You will see a window with probably three panels. Drag down the one on the upper left labeled console so you can see one labeled source. This for entering commands.
3. After entering a command, click on run. This copies the command to the console window below on the left and executes it (You can also type a number of commands, highlight them and click on run).
4. Whilst it is possible to type the commands directly into the console you are recommended to type into the source panel as if you make a mistake you can easily correct it. You cannot correct the commands in the console.

The top-right panel can show the environment, i.e. what variables are defined, and also a history of what commands you have entered.

The bottom right panel can be used for looking at plots you will produce in future weeks and at the help which is available and has other possibilities, we will explore later.

I hope you will experiment with R and find out for yourselves some of the many features it has.

To start with I will remind you about some of the basics. To save a value in a variable we use the assignment operator `<-` (less than hyphen with no space between them).

```
> x <- 3
```

We can do arithmetic as follows

```
> x <- 3
> y <- 4
> z <- sqrt(x^2 + y^2)
```

You can also decide to print on the screen the value by using the command `print(x)`.

```
> print(z)
[1] 5
```

The variables `x`, `y`, `z` will keep the values you have assigned to them until you assign a new value or delete them by using the remove function. For example `rm(x)` will delete `x`.

You can see which variables are currently defined by using the `ls` function. Note that all functions have to be followed by a pair of brackets. Often this contains arguments which you need in the function but even if there are no arguments they are still needed. So you type `ls()` to use it.

To create a vector we use the `c(...)` operator so

```
> x <- c(1, 2, 3)
> y <- c(4, 5, 6)
> c(x, y)
[1] 1 2 3 4 5 6
```

To calculate basic summary statistics we can use the following functions (we assume `x`, `y` are vectors of the same length).

```
sum(x)
mean(x)
median(x)
sd(x)
var(x)
cor(x, y)
cov(x, y)
```

Define two six element vectors and find their means, variances and correlation.

Given two (or more) vectors we can add, subtract, multiply or divide them with the operations being used element-wise

```
> v <- c(11, 12, 13, 14, 15)
> w <- c(1, 2, 3, 4, 5)
> v + w
[1] 12 14 16 18 20
> v - w
[1] 10 10 10 10 10
> v * w
```

```
[1] 11 24 39 56 75
> v / w
[1] 11.000000 6.000000 4.333333 3.500000 3.000000
```

**If one operand is a scalar then the operation is performed between every vector element and the scalar**

```
> w + 2
[1] 3 4 5 6 7
>w / 2
[1] 0.5 1.0 1.5 2.0 2.5
> mean(w)
[1] 3
> w - mean(w)
[1] -2 -1 0 1 2
```