

# R introduction

Sandra Cortijo [sandra.cortijo@slcu.cam.ac.uk](mailto:sandra.cortijo@slcu.cam.ac.uk)

*Last modified: 20 Feb 2018*

## What's *R*?

<http://www.r-project.org/>



[\[Home\]](#)

### Download

[CRAN](#)

### R Project

[About R](#)  
[Contributors](#)  
[What's New?](#)  
[Mailing Lists](#)  
[Bug Tracking](#)  
[Conferences](#)  
[Search](#)

### R Foundation

[Foundation](#)  
[Board](#)  
[Members](#)  
[Donors](#)  
[Donate](#)

## The R Project for Statistical Computing

### Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

### News

- **R 3.2.1 (World-Famous Astronaut) prerelease versions** will appear starting June 8. Final release is scheduled for 2015-06-18.
- **R version 3.2.0** (Full of Ingredients) has been released on 2015-04-16.
- **R version 3.1.3** (Smooth Sidewalk) has been released on 2015-03-09.
- **The R Journal Volume 6/2** is available.
- **useR! 2015**, will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- **useR! 2014**, took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.

- A statistical programming environment
- But offers much more than just statistics
- Open source and cross platform
- Extensive graphics capabilities
- Diverse range of add-on packages
- Active community of developers
- Thorough documentation

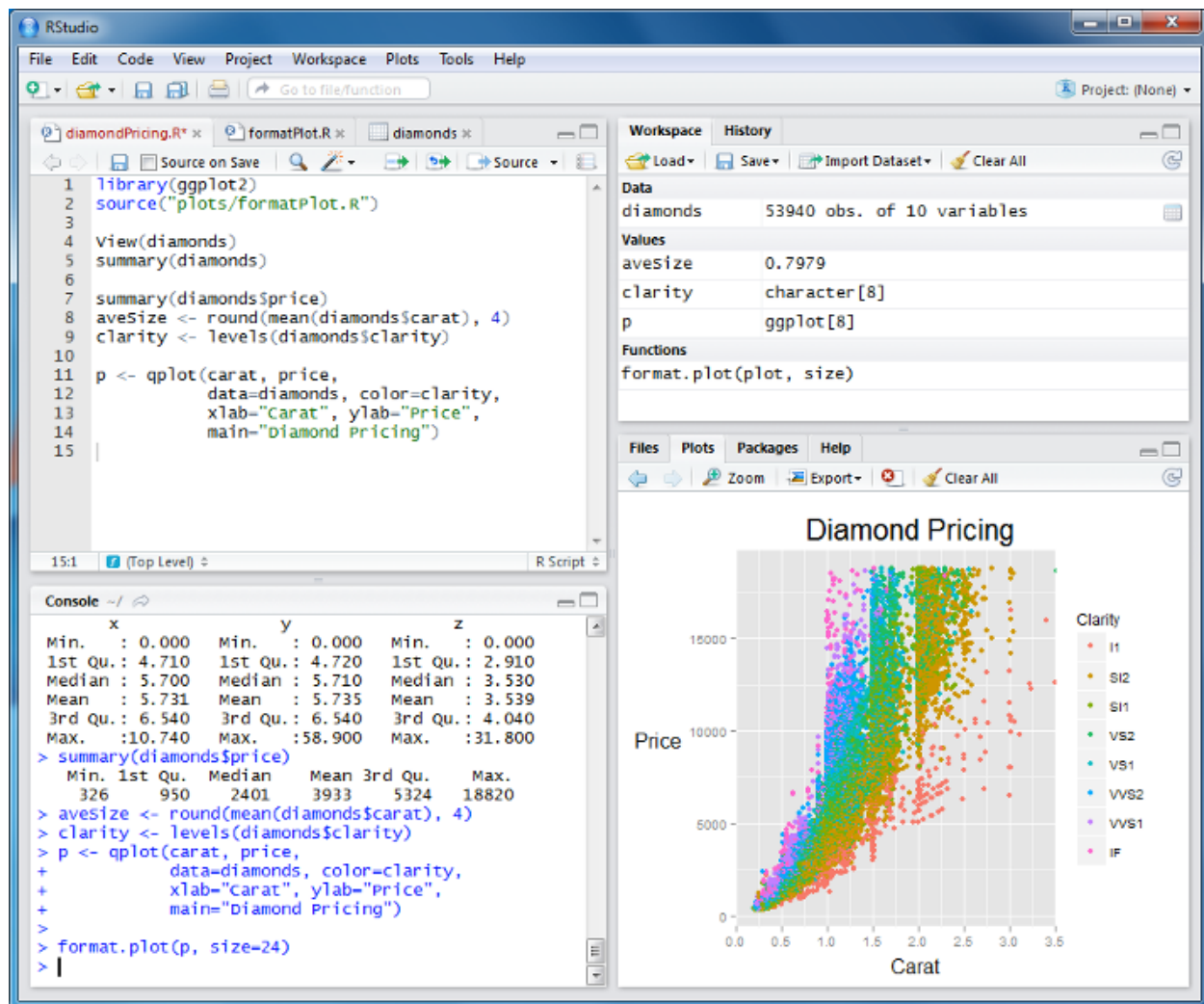
## Getting started

- On this course, we use the RStudio GUI ([www.rstudio.com](http://www.rstudio.com))



To launch *RStudio*, find the *RStudio* icon and click

This is how *RStudio* typically looks like:



## Running commands in R

The traditional way to enter R commands is via the **Console** in RStudio (bottom-left). This is where R commands are executed and give an output.

For example, we can use *R* as a calculator:

```

2 + 2
2 - 2
4 * 3
10 / 2

```

## Creating objects

We can store the output of a command in *objects* that we can name.

To store the result of some operation in a variable we use `<=`, which we can think of as an arrow pointing left (note: no spaces between `<` and `=`!).

```
# Store the result of this operation in a variable called "my.operation"
my.operation <- 3 + 2
```

We can then re-use this variable again and again, for example:

```
my.operation + 1
my.operation + 2
my.operation * 2 + 3
```

Note that the value of `my.operation` itself did not change:

```
my.operation
```

But we can change its value by “over-writing” it with a new value:

```
my.operation <- 6
```

Note that the previous value of `my.operation` is lost. If we want it back we’d have to run our previous commands again.

## The working directory

From now we will work on a new *R script*. For this, click to *File, New File, R script*.

You can type commands in the *R script* and then execute them by typing Ctrl+Enter. For example to know what is our working directory, type and then execute the following command:

```
getwd()
```

We now need to go in the right working directory for the rest of the tutorial:

```
setwd("../R_intro")
```

## Installing and loading packages

Today we will need a few R packages. Let’s install and load some of them.

You can install packages by using the `install.packages()` function. For example, if you wanted to install the `readr` package you would do:

```
install.packages("readr")
```

Once the packages are installed, we can load them in the current R session with the `library()` function:

```
library(readr)
```

Now, let’s try to load the `GenomicRanges` package, which we will use later on to manipulate genomic data.

```
library(GenomicRanges)
```

## Reading the data

Let’s read the file `H3K27ac_diffbind.csv` we will use later on, and save it in an object. This file is in another folder than our working directory: `~/Course_Materials/Differential_binding`

```
# Read data and store it in expt1 object
expt1 <- read_csv("../Differential_binding/H3K27ac_diffbind.csv")
```

You will notice the `read_csv()` function gives you a message referring to a “column specification”. This is referring to what type of data it thinks each of your columns contains.

## Getting help

Of course, it is difficult to memorize all the function names, what they are doing and how you should use them. Luckily, R has very convenient built-in help. To use it, type name of a function or any other object you are interested in preceded by `?`

```
?summary
```

R help might seem cryptic at first, but you will get used to it, you can always scroll down to the ‘examples’ section and try running some of them yourself to get an idea of what the function in question is capable of.

Also, a web-search is also just as effective (if not more!) as looking at the help pages.