Minilab 1a Worksheet

Installing R and RStudio

In this minilab, we introduce R and RStudio. R is a (free) programming language and environment for statistical computing (to analyse data) and graphics (to visualise data). RStudio is an Integrated Development Environment (IDE) for R which has a lot of nice features for Data Science.

1. Installing R and RStudio

On your personal machine, firstly download and install R from https://www.r-project.org/ (via CRAN) and secondly download and install RStudio Desktop (Open Source Edition) from https://rstudio.com/products/rstudio/ (you MUST install R before installing RStudio).

If you are asked for a CRAN mirror, please choose https://cloud.r-project.org/.

Alternatively, you can use the versions of R and RStudio on AppsAnywhere since they have recently been updated: https://appsanywhere.coventry.ac.uk/

2. Writing and running R code in RStudio

First, we will get familiar with the RStudio IDE and run some code.

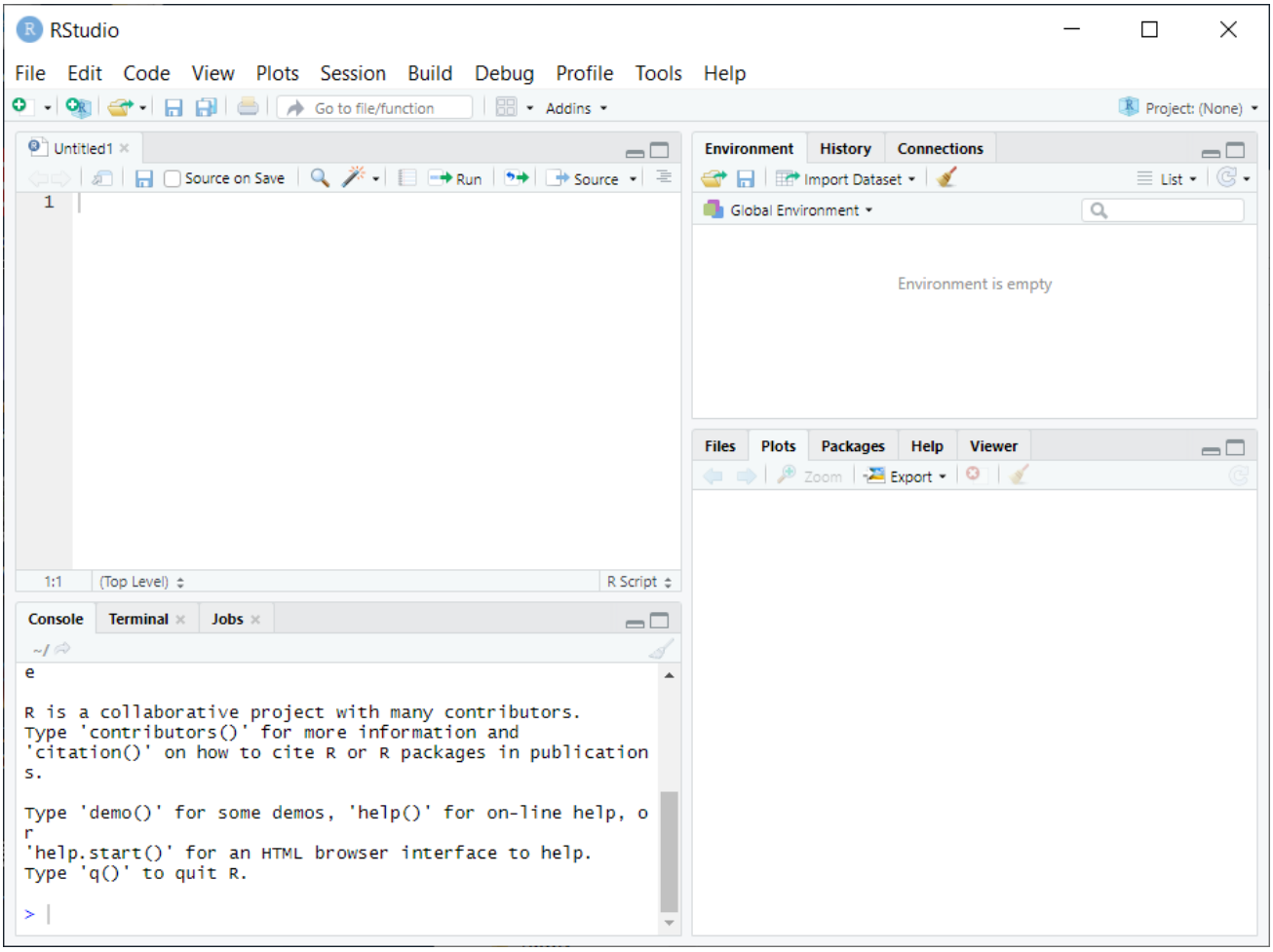
(1) Launch RStudio. Then from the File menu select New File and R Script. Notice how the IDE divides up into four window panes (see below):

Top left: file editor (can edit multiple files in different tabs)

Bottom left: R console (interactive command line and where all text output appears)

Top right: environment and history tabs

Bottom right: files, plots, packages, help and viewer tabs



(2) The console pane (bottom left) is running an R session. You can type commands directly into the console and this is also where any text output appears. The greater than sign (>) is the prompt, waiting for you to type a command. In the examples below, **don't type in the prompt**.

Type the following commands directly into the console pane (bottom left).

```
> print("Hello, World!")
[1] "Hello, World!"
> 3*(2+5)
[1] 21
> x = 10
```

```
> 3*x
[1] 30
```

In this way, we can use R as a calculator. One advantage of R is it is very efficient at calculations involving vectors and matrices. In the output you see above, the output line begins with a "[1]" indicating that the string or number that follows is the first element of a list (in the case of a string) or vector (in the case of a scalar or number).

3. Programming in R

R is a full programming language so is able to implement any structured program using sequences, selection (if) and iteration (loops). It is a curly bracket language (like C, C++, C# and Java) so blocks are delineated by curly brackets. It is dynamically typed (like Python), so you don't need to declare the type of each variable before using it (like you would have to in C++).

(1) It is much better to type code into the editor pane (top left) and save it before running it. Then if it doesn't work, you can make some changes and rerun it.

Type (or copy-and-paste) the R code below into the editor. This R code demonstrates using a comment (starting with "#"), a for loop and the if statement. The "%%" operator is the remainder when using integer division. The "cat" command prints text to the screen but does not automatically do a new line at the end.

```
# Fizzbuzz
for (i in 1:20)
{
  if (i%%15==0)
    cat("FIZZBUZZ ")
  else if (i%%3==0)
    cat("fizz ")
  else if (i%%5==0)
    cat("buzz ")
  else
    cat(i," ")
}
```

(2) From the File menu select Save to save your code as "lab01a.R" in a suitable location (perhaps a folder for the module on OneDrive). Then, use ctrl-a in the editor to select the code you want to run and click on the Run button (just above the text in the editor pane) to execute that R code in the console pane.

Exercise (optional). Write an R program using a for loop (see the syntax in the example fizzbuzz code above) to add the integers 1 up to n. For example, when $n = 10$ we should get

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

Keep your code as simple as possible. Test your code for different values of n. Do not attempt to read the value of n from the keyboard.

Write some additional R code to check your answers against the value of

$$\frac{n \times (n+1)}{2}$$

Challenge (optional). Modify your R code to add the cubes of the integers 1 up to n. For example, when $n = 3$ we should get

$$1^3 + 2^3 + 3^3 = 1 + 8 + 27 = 36$$

Use google to find a formula (involving n) that gives this sum directly.

Summary

In this minilab, we have seen how to install R and RStudio, and how to write and run some R code in RStudio.