FIRST STEPS WITH R

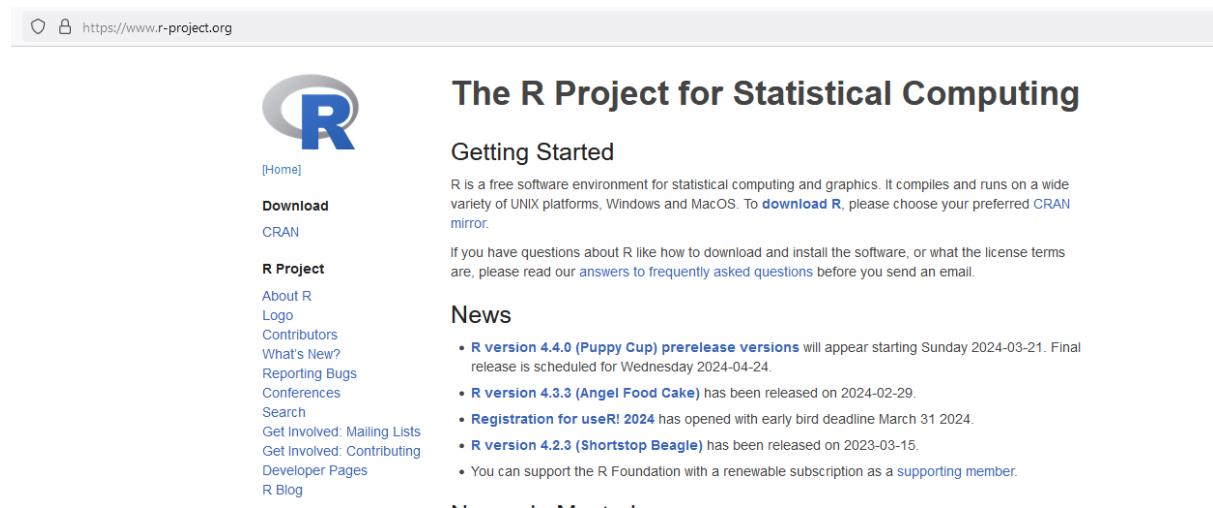## Oxford University Statistical Consulting

Part 0 – R and RStudio

If you already have R and RStudio installed on your machines, feel free to skip to **Part I**. Below are instructions on where to find R, and RStudio and how to download and install them.

We will be working on RStudio. For this you will need a fully functioning version of R installed on your machines.

TO GET A COPY OF R, VISIT:

https://www.r-project.org

You can then select either "**download R**" in the main text or "**CRAN**" on the left-hand side.

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here:

If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

0-Cloud
    https://cloud.r-project.org/   Automatic redirection to servers worldwide, currently sponsored by Rstudio
Argentina
    http://mirror.fcaglp.unlp.edu.ar/CRAN/   Universidad Nacional de La Plata
Australia
    https://cran.csiro.au/   CSIRO
    https://mirror.aarnet.edu.au/pub/CRAN/   AARNET
    https://cran.ms.unimelb.edu.au/   School of Mathematics and Statistics, University of Melbourne
    https://cran.curtin.edu.au/   Curtin University
Austria
    https://cran.wu.ac.at/   Wirtschaftsuniversität Wien
Belgium
    https://www.freestatistics.org/cran/   Patrick Wessa
    https://ftp.belnet.be/mirror/CRAN/   Belnet, the Belgian research and education network
Brazil
    https://cran-r.c3sl.ufpr.br/   Universidade Federal do Parana
    https://cran.fiocruz.br/   Oswaldo Cruz Foundation, Rio de Janeiro
    https://vps.fmvz.usp.br/CRAN/   University of Sao Paulo, Sao Paulo
    https://brieger.esalq.usp.br/CRAN/   University of Sao Paulo, Piracicaba
Bulgaria
    https://ftp.uni-sofia.bg/CRAN/   Sofia University
Canada
    https://mirror.rcg.sfu.ca/mirror/CRAN/   Simon Fraser University, Burnaby
    https://muug.ca/mirror/cran/   Manitoba Unix User Group
    https://mirror.csclub.uwaterloo.ca/CRAN/   University of Waterloo
Chile
    https://cran.dcc.uchile.cl/   Departamento de Ciencias de la Computación, Universidad de Chile

This page gives you the option to select what server to download your R copy from. All servers available have identical copies of the software but common practice is to use the server closest to you geographically as to not overload alphabetically-early servers. Scroll down to find the **UK**, if you are in the UK, where the options are to download from the **Imperial** or the **Bristol** server.

After choosing your nearest server you should see the following screen:

Select the version matching your operating system (e.g. "Download R for Windows"). If you selected Windows, you should be seeing this screen:

Subdirectories:

base	Binaries for base distribution. This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 4.0.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 4.0.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

In this case, click on the "**install R for the first time**" link and then select "**Download R for Windows**".

If you selected (Mac) OS , you should be seeing:

R for macOS

This directory contains binaries for the base distribution and of R and packages to run on macOS. R and package binaries for R versions older than 4.0.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting (https://cran-archive.r-project.org) accordingly.

Note: Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 4.3.3 "Angel Food Cake" released on 2024/02/29

Please check the integrity of the downloaded package by checking the signature:
pkgutil --check-signature R-4.3.3-arm64.pkg
in the *Terminal* application. If Apple tools are not avaiable you can check the SHA1 checksum of the downloaded image:
openssl sha1 R-4.3.3-arm64.pkg

Latest release:

For Apple silicon (M1-3) Macs:
R-4.3.3-arm64.pkg
SHA1-hash: 37dc051e0a241eeef8e8207b271006f767781e6c
(ca. 92MB, notarized and signed)

R 4.3.3 binary for macOS 11 (**Big Sur**) and higher, signed and notarized packages.

Contains R 4.3.3 framework, R.app GUI 1.80, Tcl/Tk 8.6.12 X11 libraries and Texinfo 6.8. The latter two components are optional and can be ommitted when choosing "custom install", they are only needed if you want to use the tcltk R package or build package documentation from sources.

For older Intel Macs:
R-4.3.3-x86_64.pkg
SHA1-hash: cfdf7e4657eb67d3a35f6c74772680aa3c6b8d4b
(ca. 94MB, notarized and signed)

macOS Ventura users: there is a known bug in Ventura preventing installations from some locations without a prompt. If the installation fails, move the downloaded file away from the *Downloads* folder (e.g., to your home or Desktop).

Note: the use of X11 (including tcltk) requires XQuartz (version 2.8.5 or later). Always re-install XQuartz when upgrading your macOS to a new major version.

This release uses Xcode 14.2/14.3 and GNU Fortran 12.2. If you wish to compile R packages which contain Fortran code, you may need to download the corresponding GNU Fortran compiler from https://mac.R-project.org/tools. Any external libraries and tools are expected to live in /opt/R/arm64 (Apple silicon) or /opt/R/x86_64 (Intel).

NEWS (for Mac GUI)

News features and changes in the R.app Mac GUI

Mac-GUI-1.80.tar.gz
SHA1-hash: 5cdd4e1ef0991cbf74a082f8ef7624697eb88fad

Sources for the R.app GUI 1.80 for macOS. This file is only needed if you want to join the development of the GUI (see also Mac-GUI repository), it is not intended for regular users. Read the INSTALL file for further instructions.

If you have an M1 processor select "**R-4.3.3-arm64.pkg**" to start your download (number of the version may have increased by the time you read this!). Otherwise, select "**R-4.3.3.pkg**". If you don't know which one you have, click on the apple symbol of your Mac, then select "About This Mac" and see there.

Once downloaded, click on .exe file (for Windows), or the .dmg file (for Mac). In both cases this will start an installation wizard. Accept the default settings when given the option and complete the installation.

TO DOWNLOAD RSTUDIO, VISIT:

https://posit.co/downloads/

You should be seeing this:
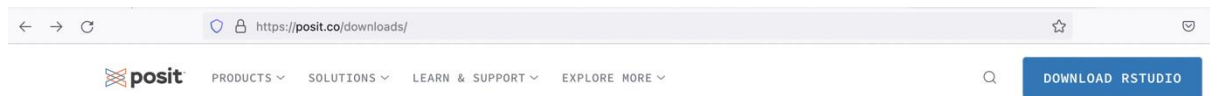
Download the RStudio IDE

The most popular coding environment for R, built with love by Posit.

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

Click on "**Download RStudio**", scroll down a bit and select the free version.
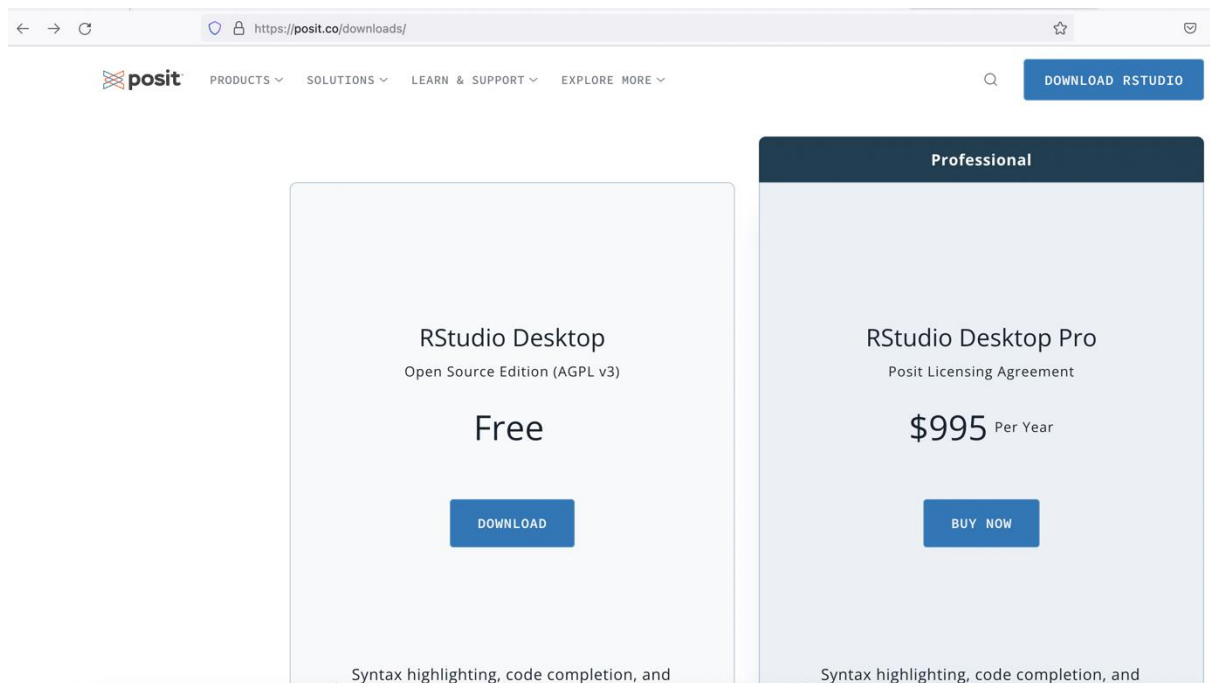
Scroll down to "Step 2" and select the download recommended for your machine.

Step 2:
Install RStudio Desktop

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

Size: 202.76MB | SHA-256: FD8EA4B4 | Version: 2022.12.0+353 |
Released: 2022-12-15

All Installers and Tarballs

RStudio requires a 64-bit operating system. If you are on a 32 bit
system, you can use an older version of RStudio.

Linux users may need to import Posit's public code-signing key prior to
installation, depending on the operating system's security policy.

OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO-2022.12.0-353.EXE ⌄	202.76MB	FD8EA4B4
macOS 10.15+	RSTUDIO-2022.12.0-353.DMG ⌄	365.70MB	FD4BEBB5
Ubuntu 18+/Debian 10+	RSTUDIO-2022.12.0-353-AMD64.DEB ⌄	131.19MB	23CAE58F

Once the download is complete run the installer to complete the installation.

Part I – Getting used to RStudio

The first time you open RStudio, it should look like this:
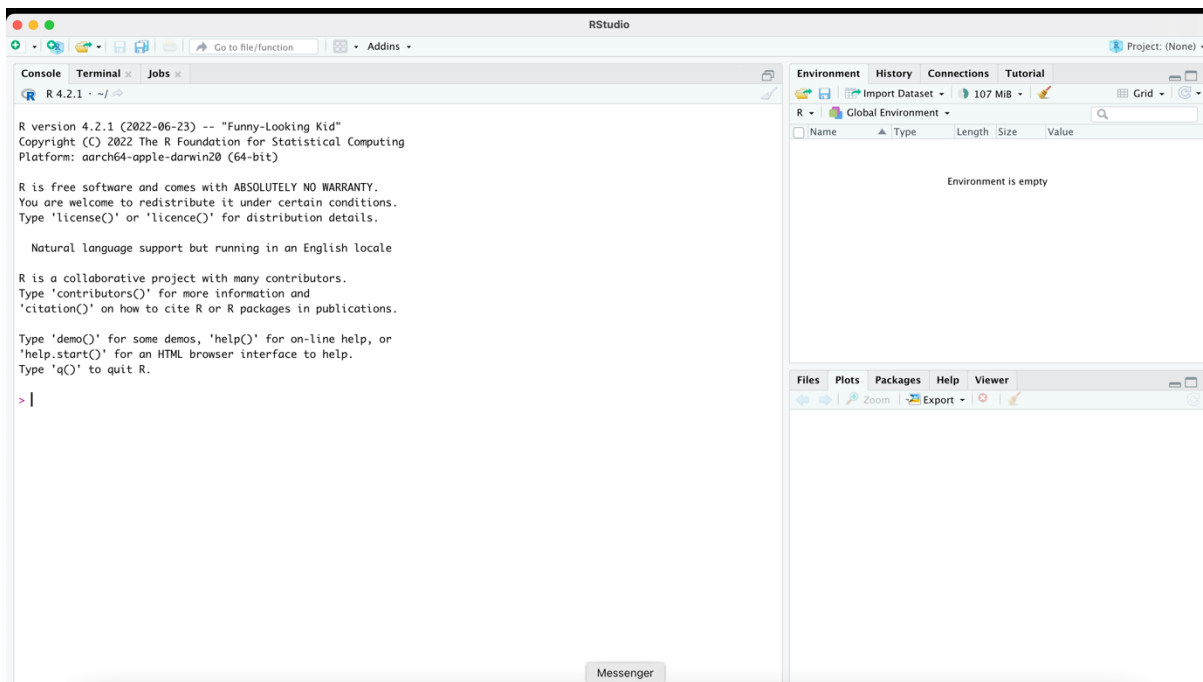
The following paragraphs are here to give you a tour of what each panel does.

The **"Console"** on the left-hand side is where R commands run. The console is where the magic happens.

The top quarter of the right-hand side has four tabs. **"Environment"** is where, for the session you are running, a set of names is associated with specific values. For example, if you name a dataset "sunflowers1", then the environment will store that name for that session so that every time you ask for "sunflowers1", R will understand you mean that particular dataset. **"History"** is a list of commands you have used in that session. **"Connections"** is for when you are working with databases. **"Tutorial"** allows you to run tutorials that will help you learn different functionalities.

The lower quarter of the right-hand side has five tabs. **"Files"** shows you the current working directory. This is where the session will look for files and will save output files. **"Plots"** is where any graphics you produce during the session will be displayed. **"Packages"** is a list of libraries downloaded on your machine that can be loaded for the session by checking the tick box. Additional packages can also be downloaded through that panel. **"Help"** contains the manual pages for all functions loaded on your session. **"Viewer"** is for local web-content, we will not be going through that on this course.

Even though the "**Console**" is where commands are run and output is presented, we prefer to write commands **USING A SCRIPT** for three reasons. A script is just a text file that you can edit, save, and send to collaborators. This is our first reason for using scripts instead of coding directly on the console. The second reason is that the script does not run the code if you press enter. It only runs it when you specifically select a code segment and click Run or press Enter (Cmd+Enter on Mac). This makes it easier to write a first draft of your code. The third reason is you can use # to add comments to your code, making it easier for yourself and others to follow what each step does.

To start a new script, click on the green plus sign on the top left-hand side corner, just above your console.
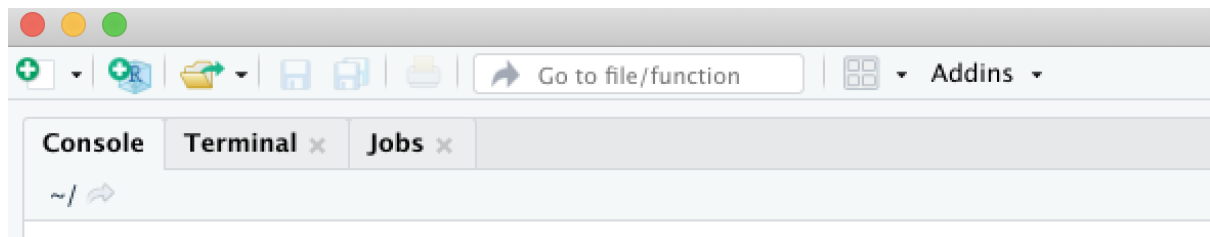
This should give you the option of an "R script" at the top. If you click on that, your console will be reduced to the bottom half of the left-hand side and your new script will be on the top half:
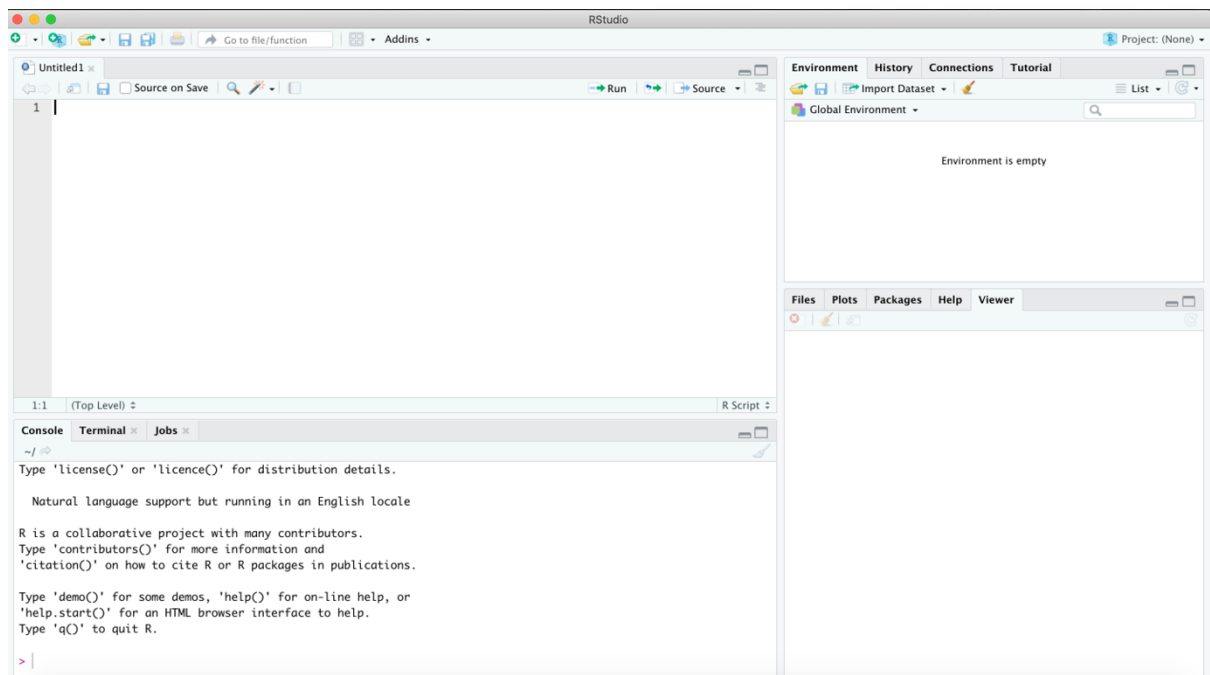
You can save your script by clicking on the little floppy disk symbol. You can do the rest of the practical using either a script or by typing directly in the console.

Part II – R projects

It is common practice to keep all files associated with an analysis in one directory. This means that in a single directory you can save your data, your scripts, and your analysis outputs. It's very convenient but also important for reproducibility: you can easily work out what dataset version you used, and also you can share the entire project with a collaborator without worrying about sending them any additional datasets or scripts - everything is in that directory.

RStudio has an easy, built-in way to create projects.

Go to the toolbar at the top, click on **File**> **New Project…**

This would open the project wizard:
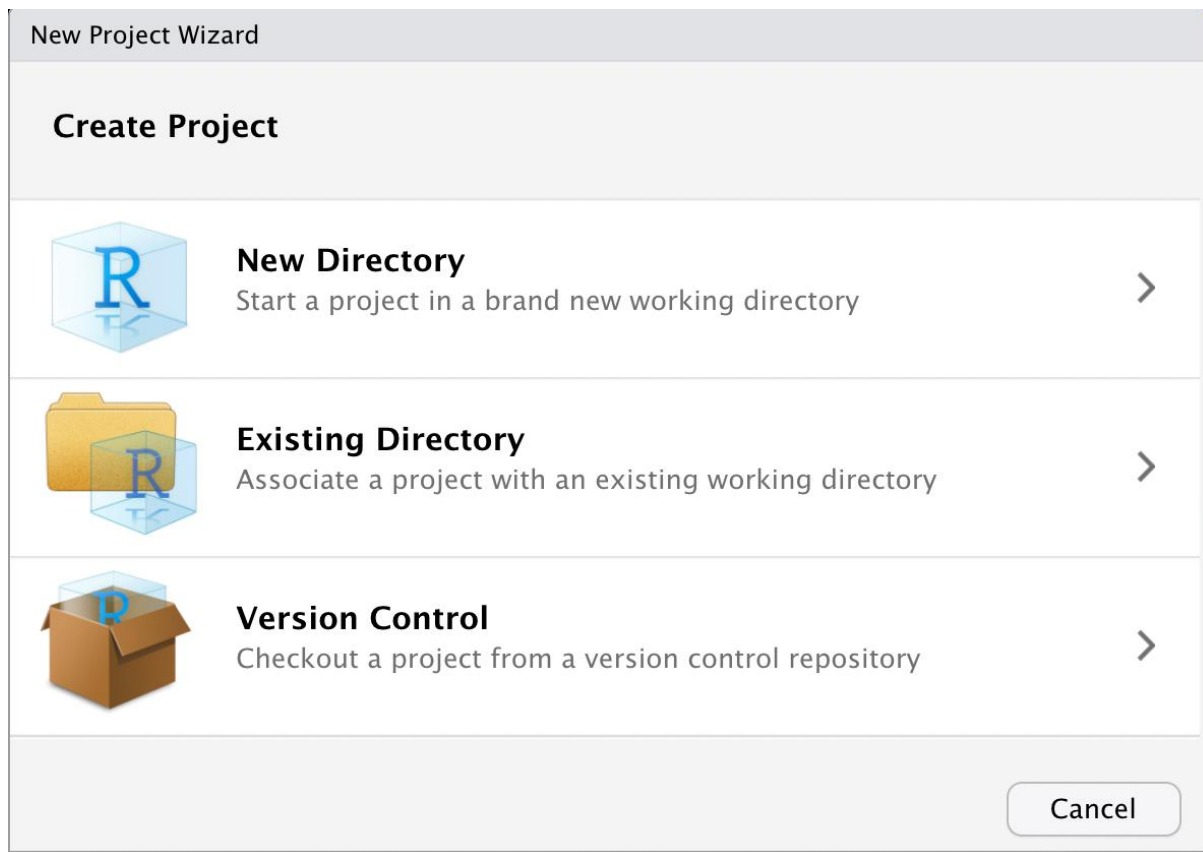
You can select "**New Directory**" if you wish to make a new directory (i.e. folder). Alternatively, if you wish to add the project into an existing directory, select "**Existing Directory**". You can then browse your directory tree to place your project in the appropriate position:
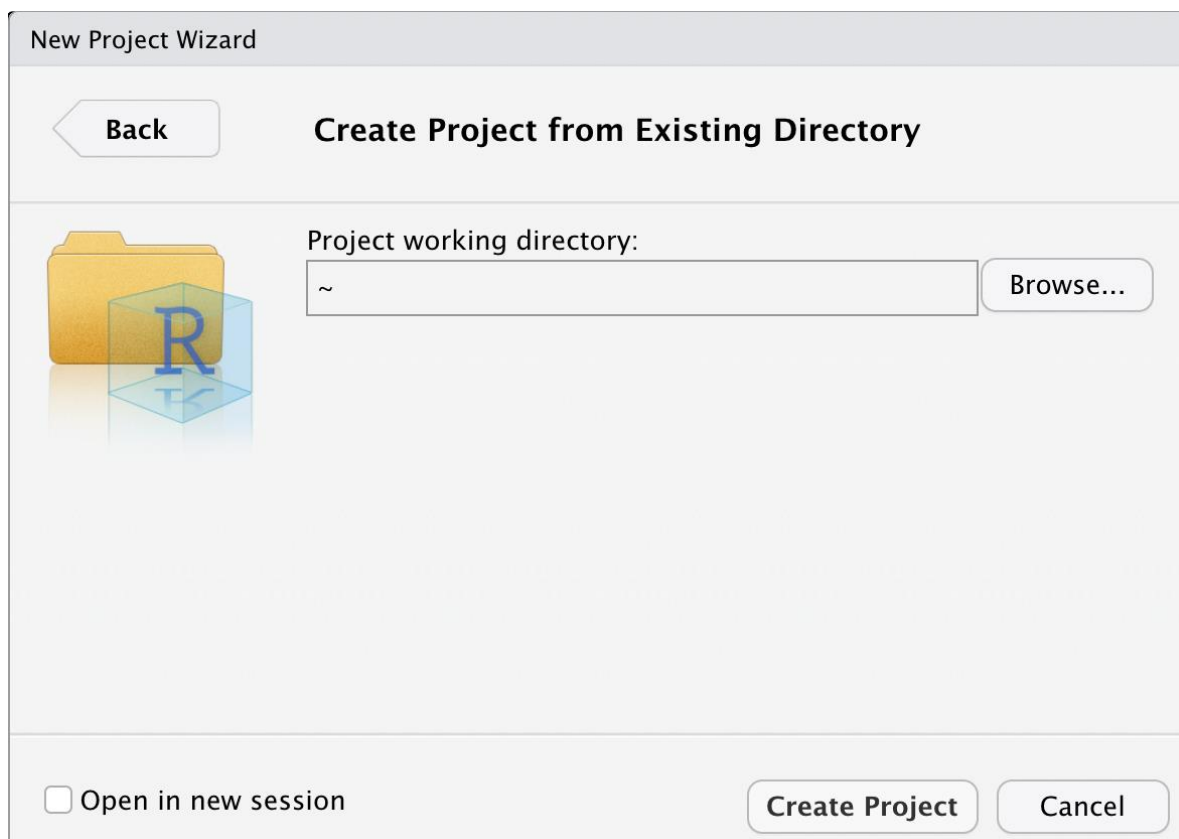
You can then click create project.

What this process will do is set your working directory to that folder automatically and unless specified otherwise all scripts and outputs will be saved there. You can add any dataset files in the project directory, and also create subfolders to organise your output (e.g. "figures", "scripts" or "reports").

Part III – R as a glorified calculator

You can run basic operations on R, just like you would if you were using a scientific calculator. You can run these on the console, but we recommend using a script for everything you do. Here are some examples of what you can do.

The basic operators: `+  -  /  *`
Try for example `6*7`, or `39+3`, or `126/3`

Logical operators: `<,>,  ==,  !=`
This is where you can ask R to run a logical test. Something like: "is 7 equal to 6" or "is 11 greater than 10"
Try `7==6`, `11>10`, or `2^3!=8`

Powers: `^`
This is your standard exponentiation. For example, 2 to the power of 3 would be "`2^3`"

Brackets: `()`
Brackets are used to prioritise operations.
For example: `(5+7)*12` versus `5+7*12`

Logarithms : `log()` and `log10()`
Standard logarithms for R are base e, for base 10 use "log10()"
Try: `log(100)` and `log10(100)`

Exponential: `exp()`
This raises `e` to the power of the value in brackets. Exp(17) is the equivalent of e^{17}

Square root: `sqrt()`
This calculates the square root of a number. This is equivalent to raising that number to the power of ½. Try `sqrt(9)` and `9^1/2`

Absolute value: `abs()`
Absolute values calculate the distance between a number and zero. As it is a distance, it is always a positive value.

Try `abs(5)` and `abs(-5)`