# Session 1_3. Setup

## Table of contents

#### Questions

- How do you use RStudio project to manage your analysis project?
- What is the most effective way to organize directories for an analysis project?
- How to obtain additional package you need?

#### Learning Objectives

- Create an RStudio project and the directories required for storing the files pertinent to an analysis project.
- Install additional packages using the packages tab.
- Install additional packages using R code.

## Introduction

Typically, an analysis project begins with dataset files, a handful of R scripts and output files in a directory. As the project advances, complexity inevitably rises with the addition of more scripts, output files, and possibly new datasets. The complexity is further amplified when dealing with multiple versions of scripts and output files, necessitating efficient organisation. If these are not well-managed from the beginning, resuming the project after a break, or sharing the project with someone else becomes challenging and time-consuming, as we struggle to recall the project's status and navigate the directory tree.

In this session, we will first focus on an effective strategy for managing the files within a *working directory*.

What is a working directory?

A working directory in R is the default location on your computer where R looks for files to load or store any data you wish to save.

Secondly, we will also learn how to leverage RStudio projects, a feature built-in to RStudio for managing our analysis project.

## Working directory

### Structuring your working directory

For a more streamlined workflow, we suggest storing all files associated with an analysis in a specific directory, which will serve as your project's working directory. Initially, this working directory should contain four distinct directories:

- `data`: dedicated to storing raw data. This folder should ideally only house raw data and not be modified unless you receive a new dataset.
- `scripts`: for storing the R scripts you've written and utilized for analyzing the data.
- `documents`: for storing documents related to your analysis, such as a manuscript outline or meeting notes with your team.
- `output`: for storing intermediate or final results generated by the R scripts in the `scripts` directory. Importantly, if you carry out data cleaning or pre-processing, the output should ideally be stored in this directory, as these no longer represent raw data.

As your project grows in complexity, you might find it necessary to create more directories or sub-directories. Nevertheless, the aforementioned four directories should serve as the foundation of your working directory.

### Create directories

We will create the four directory under our project as recommended above. You can create them by clicking on the "New Folder" button in the file panel (bottom right) or directly from R by typing at console:

```r
dir.create("data")
dir.create("scripts")
dir.create("documents")
dir.create("output")
```

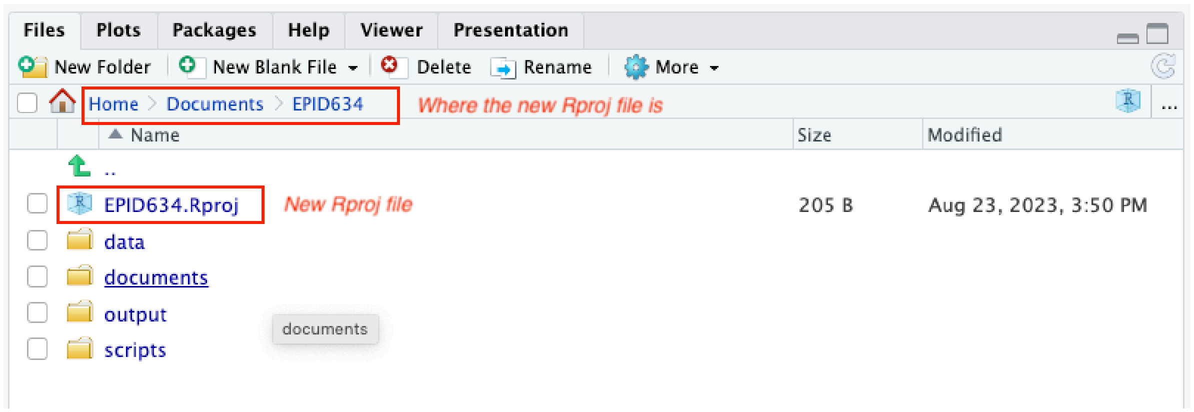
## RStudio project

RStudio project is a feature built-into RStudio for managing your analysis project. It does so by storing project-specific settings in an `.Rproj` file stored in your project's working directory. Loading these settings up into RStudio by either opening the `.Rproj` file directly or through RStudio's open project option (from the menu bar, select `File` > `Open Project...`) will automatically set your working directory in R to the location of the `.Rproj` file, essentially your project's working directory.

To create an RStudio project:

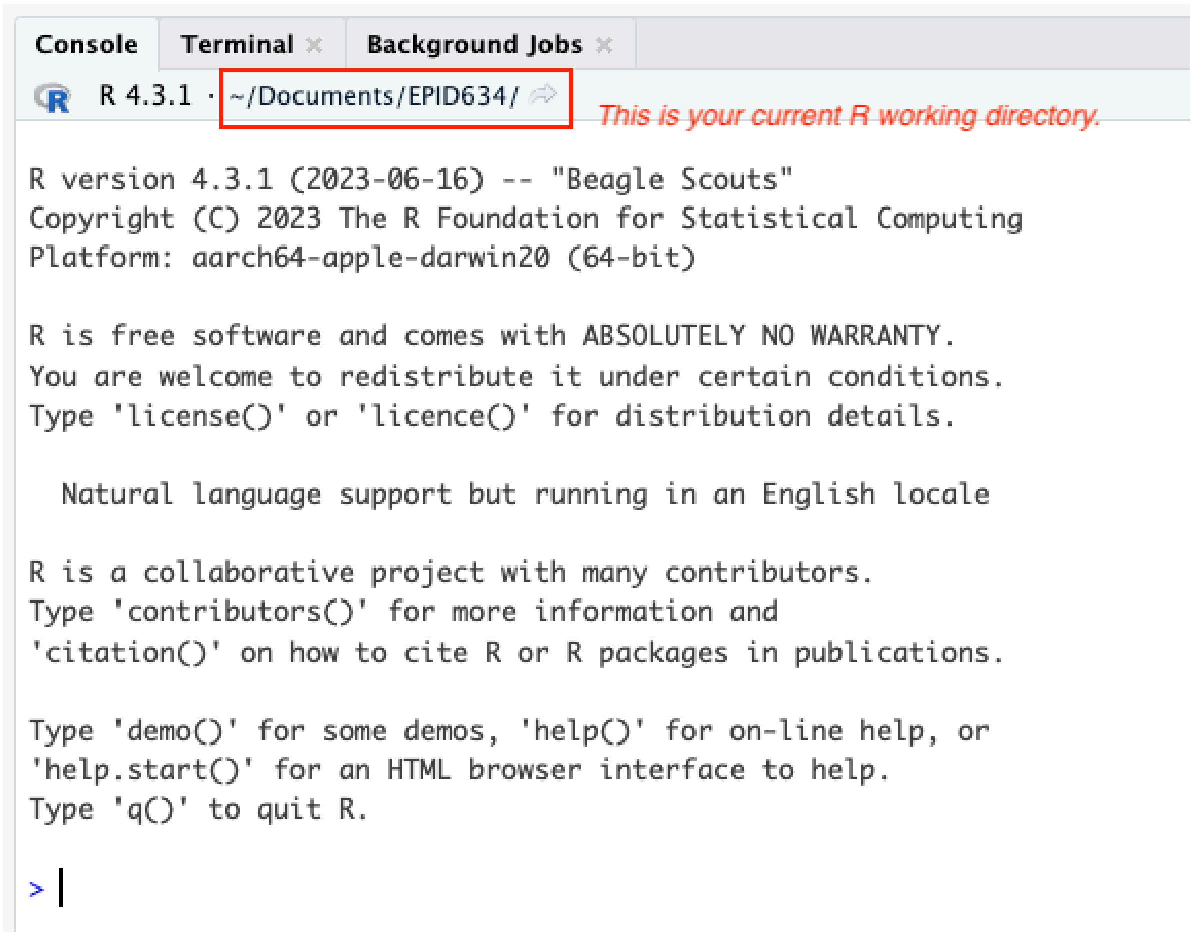
1. Start RStudio.
2. Navigate to the menu bar and select `File` > `New Project...`.
3. Choose `Existing Directory`.
4. Click `Browse...` button, and select the directory you have previously chosen as the working directory for the analysis (i.e., the directory where the 4 essential directories reside).
5. Click `Create Project` at the bottom right of the window.

Upon completion of the steps above, you will find a `.Rproj` file within your project's working directory.



```
..
EPID634.Rproj      205 B    Aug 23, 2023, 3:50 PM
data
documents
output
scripts
```

Moreover, the heading of your RStudio console should now also display the absolute path of your project's working directory, i.e., where the `.Rproj` file resides, indicating that RStudio has set this directory as your working directory in R.


Console
Terminal
Background Jobs
R 4.3.1 · ~/Documents/EPID634/
This is your current R working directory.

```
R version 4.3.1 (2023-06-16) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

From this point forward, any R code that you execute that involves reading data from a file or saving data in a file will, by default, be directed to a path relative to your project's working directory.

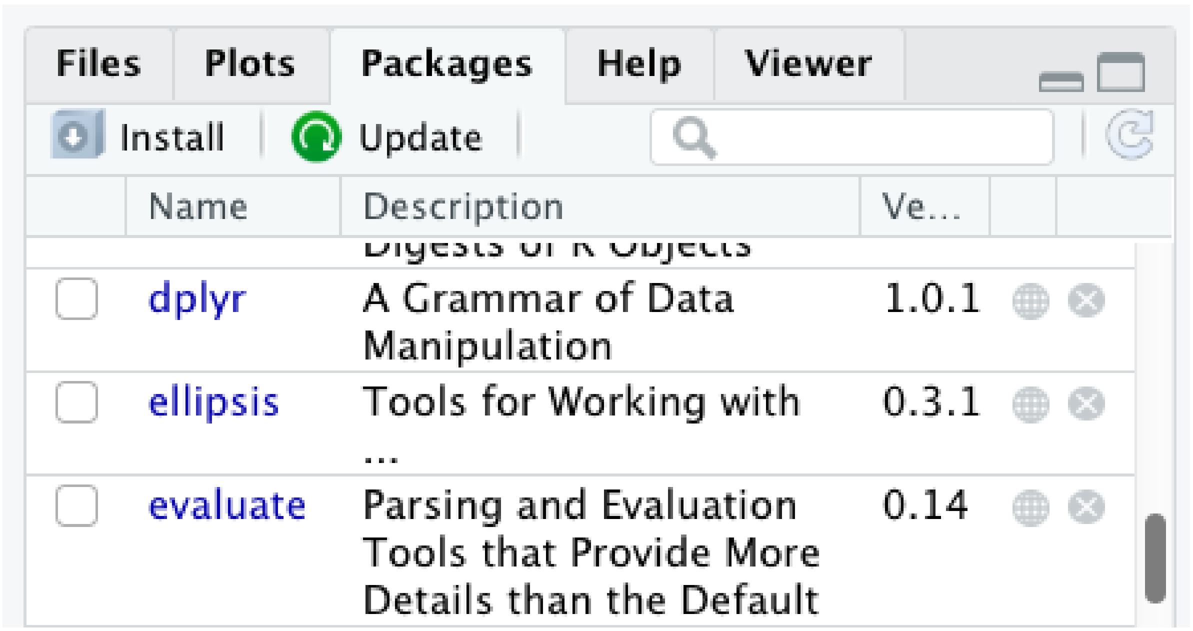
If you wish to close the project, perhaps to open another project, create a new one, or take a break from the project, you can do so by using to `File` > `Close Project` option located in the menu bar. To open the project back up, either double-click on the .Rproj file in the working directory, or open up RStudio and using the `File` > `Open Project` option in the menu bar.

## Install packages

In addition to the core R installation, there are in excess of 10,000 additional packages which can be used to extend the functionality of R. Many of these have been written by R users and have been made available in central repositories, like the one hosted at CRAN, for anyone to download and install into their own R environment. Let's install one of the package we will use in this course, `tidyverse`.

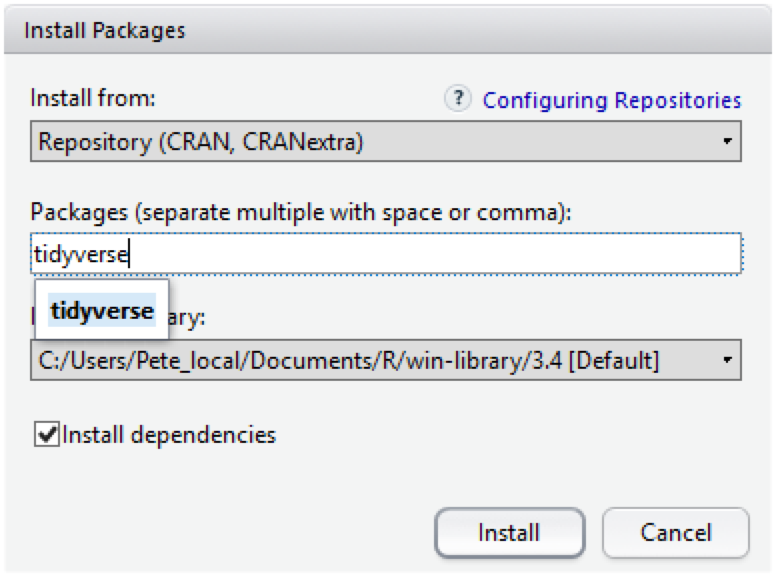
### Using the package tab

You can see if you have a package installed by looking in the ***Packages*** tab (on the lower-right by default).



| | Name | Description | Ve... |
|---|---|---|---|
| | | Digests of R Objects | |
| | dplyr | A Grammar of Data Manipulation | 1.0.1 |
| | ellipsis | Tools for Working with ... | 0.3.1 |
| | evaluate | Parsing and Evaluation Tools that Provide More Details than the Default | 0.14 |

Packages can be installed from the ***Packages*** tab: click the ***Install*** icon and start typing the name of the package you want in the text box. As you type, packages matching your starting characters will be displayed in a drop-down list so that you can select them.



Repository (CRAN, CRANextra)

tidyverse

C:/Users/Pete_local/Documents/R/win-library/3.4 [Default]

☑ Install dependencies

At the bottom of the ***Install Packages*** window is a check box to ***Install dependencies***. This is ticked by default, which is usually what you want. Packages can (and do) make use of functionality built into other packages, so for the functionality contained in the package you are installing to work properly, there may be other packages which have to be installed with them. The ***Install dependencies*** option makes sure that this happens.

### Using R code

You could also install packages using `install.packages()` function.

```
install.packages("tidyverse")
```

## Summary

- Proper organisation of the files required for your project in a working directory is crucial for maintaining order and ensuring easy access in the future.
- RStudio project serves as a valuable tool for managing your project's working directory and facilitating analysis.
- Use `install.packages()` to install packages (libraries)

  Reference
  https://datacarpentry.org/r-socialsci/instructor/00-intro.html#getting-set-up
  https://carpentries-incubator.github.io/bioc-rnaseq/02-setup.html
  https://carpentries-incubator.github.io/bioc-intro/20-r-rstudio.html#the-working-directory