Workflow Essentials

Data Workflow in R

Dr Muhammad Saufi Abdullah
DrPH (Epidemiology) Candidate

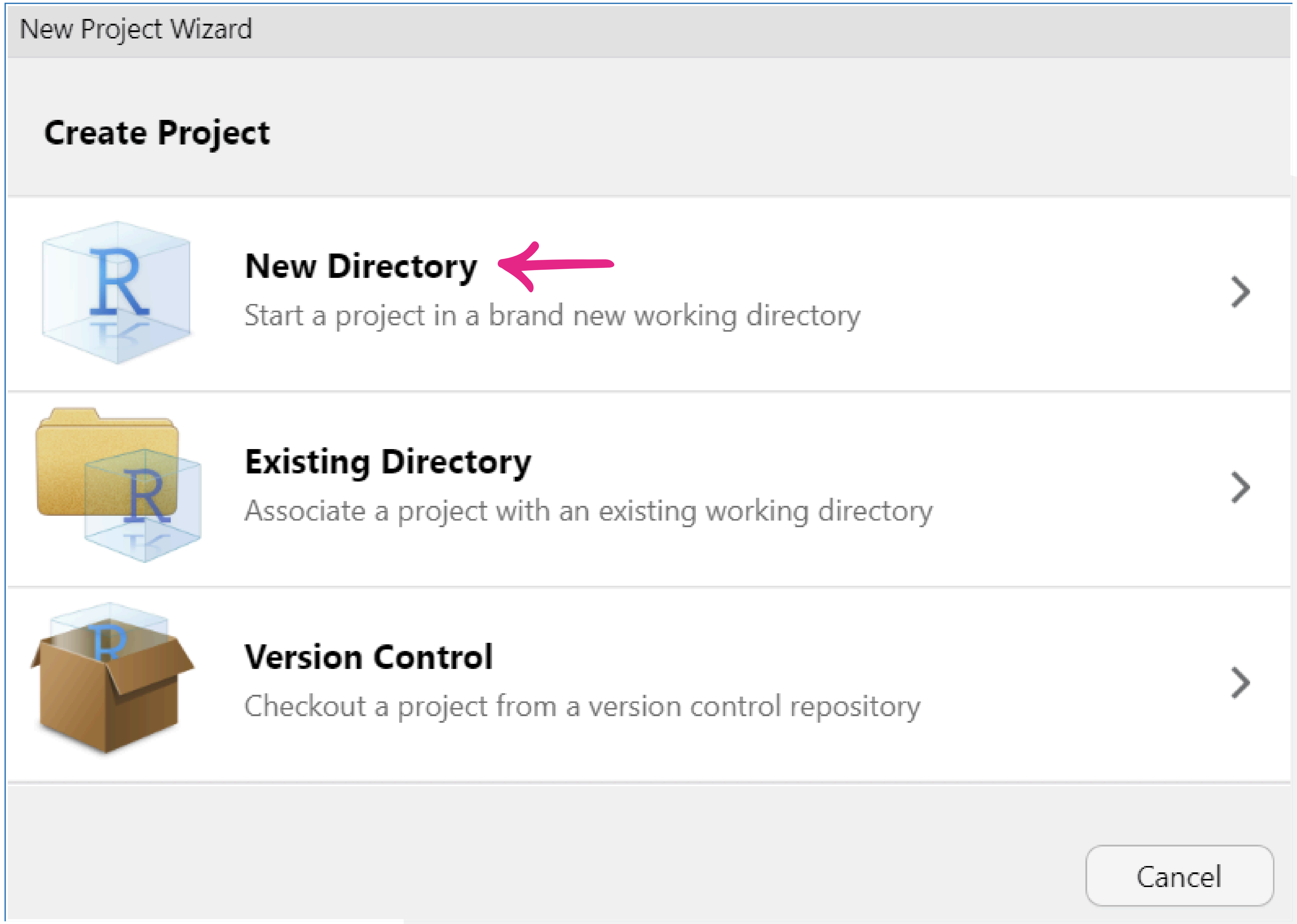
Working Directory

The first and most important thing to do before starting any work in **R** is to set the working directory.

The working directory is a folder that contains **R codes**, **R outputs**, and **datasets** that contribute to the **R project**.

The recommended method to set up a working directory is by creating a new **R project**. In **RStudio**, go to the **File** menu and choose **New Project**.

In the **New Project Wizard**, select **New Directory**.



New Project Wizard

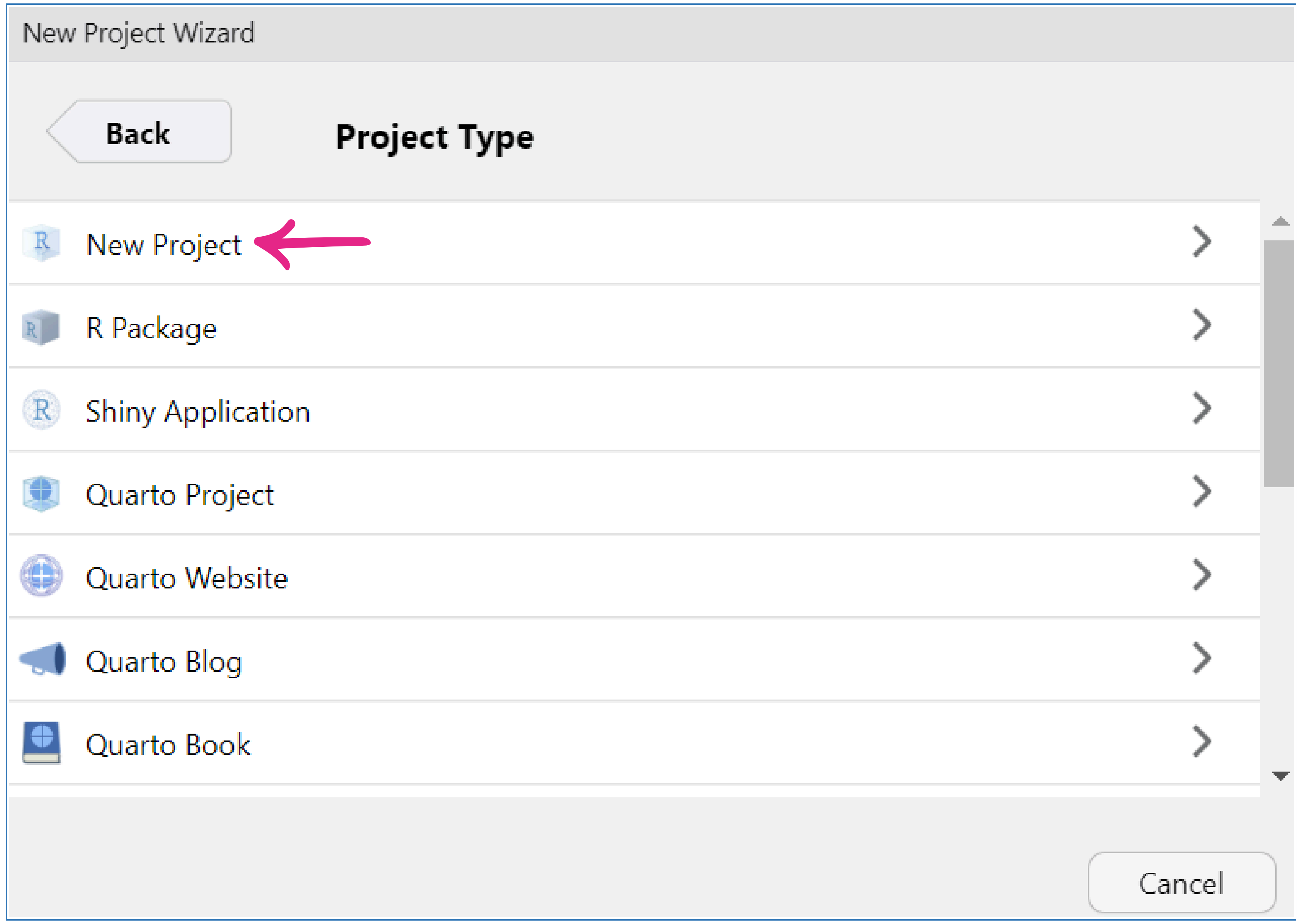
Create Project

New Directory
Start a project in a brand new working directory

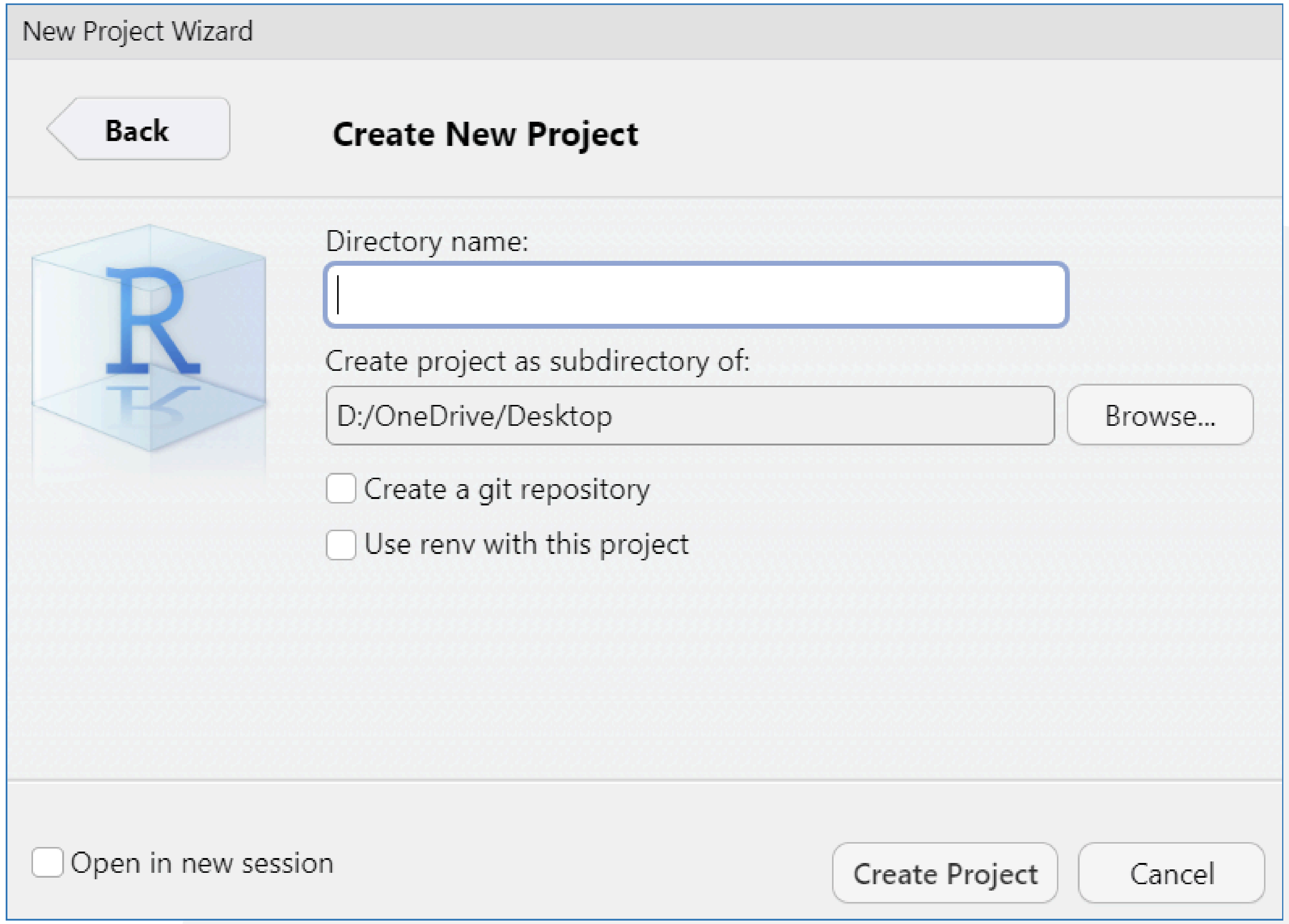
Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Working Directory



Then choose **New Project**.



Directory name:

Create project as subdirectory of: D:/OneDrive/Desktop

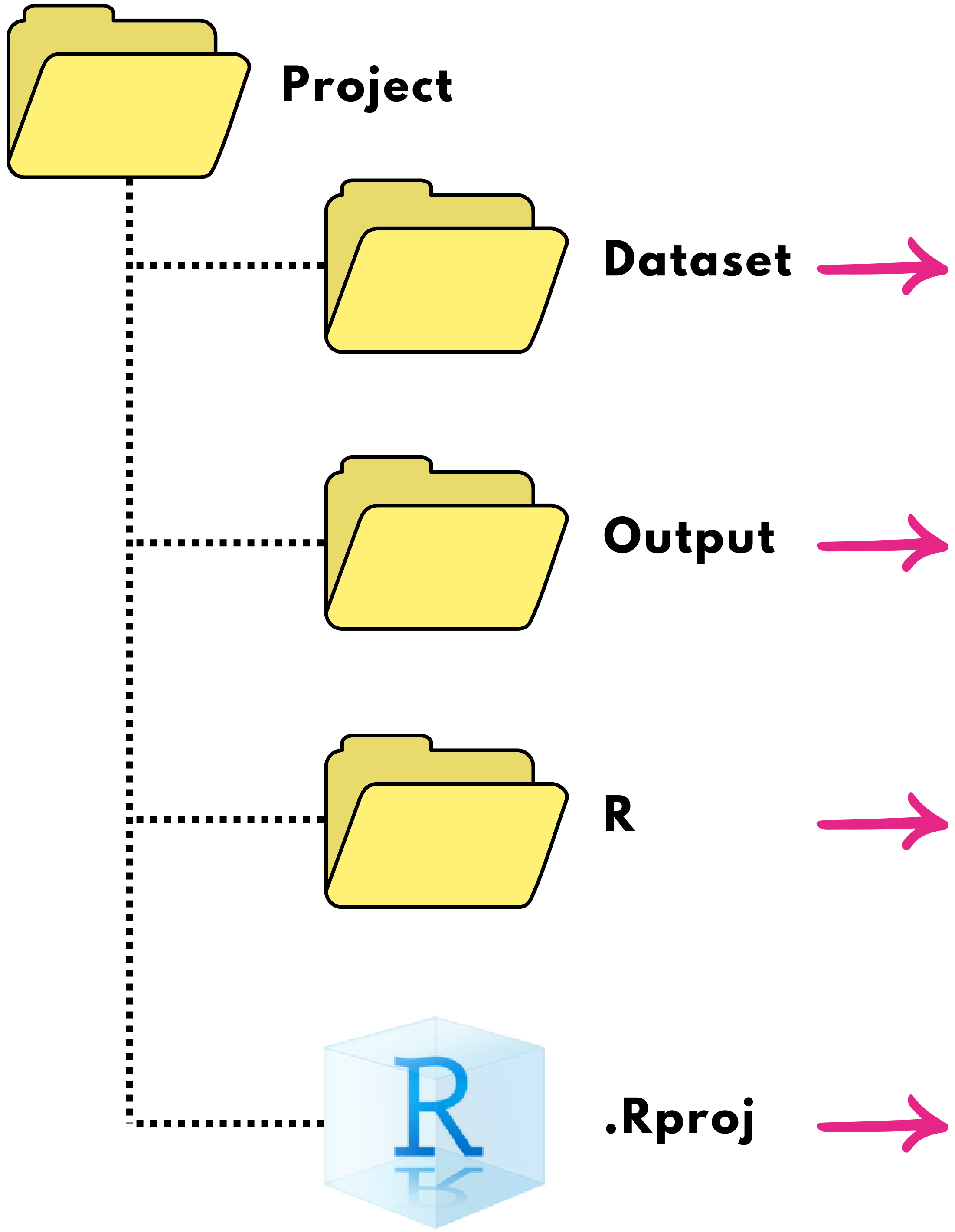
□ Create a git repository

□ Use renv with this project

□ Open in new session

Name your **R project** and choose the location where the folder or working directory containing this **R project** will be located.

It is advisable to create subfolders within your working directory to properly manage your files and other **R objects**.



Project

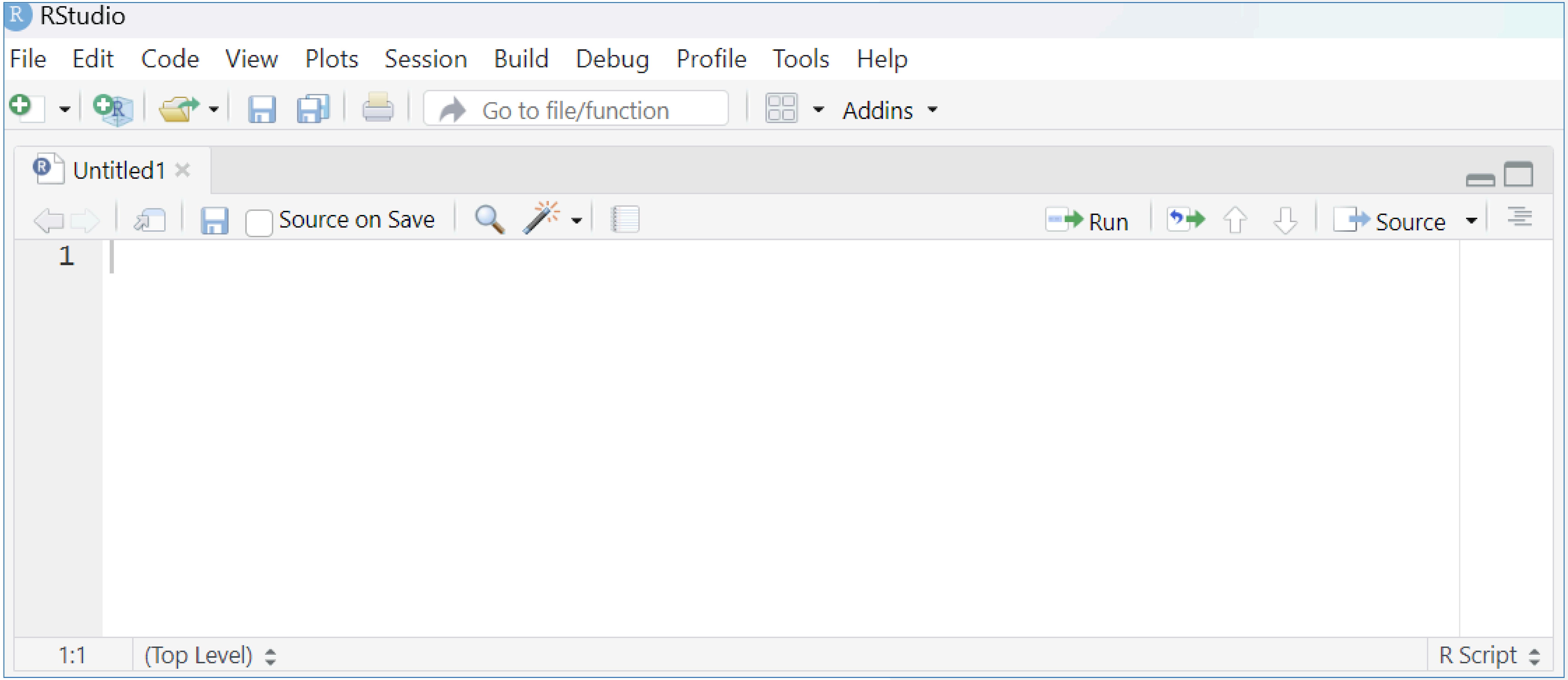
Dataset → *Stores all the raw data files needed for analysis.*

Output → *Keeps the results of the analysis, such as plots, tables, etc.*

R → *Contains R scripts used for data analysis.*

.Rproj → *Once a new **R project** is created, a file with the extension "**.Rproj**" will appear inside the folder. You can double-click this file to directly open the project.*

R Script



In **RStudio**, go to the **File** menu, then select **R Script**. The **Source** pane with a blank **R script** will appear as shown below.

R Packages

R packages are collections of functions and data sets developed by the community, enhancing the power of **R** by improving existing base **R** codes and functions or adding new ones. There are two methods for installing R packages:

Method 1: Using R Code

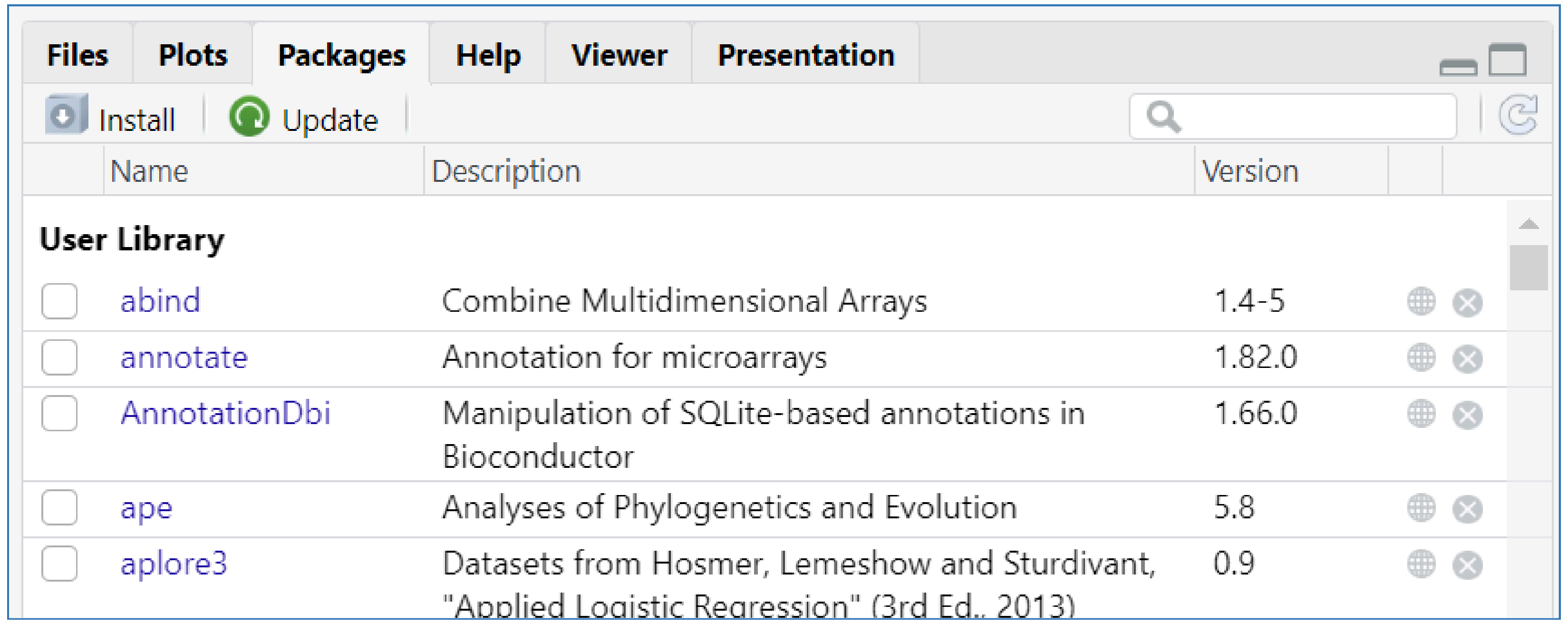
Type the following code to install an **R** package, for example, the `**tidyverse**` package:

In R script:

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

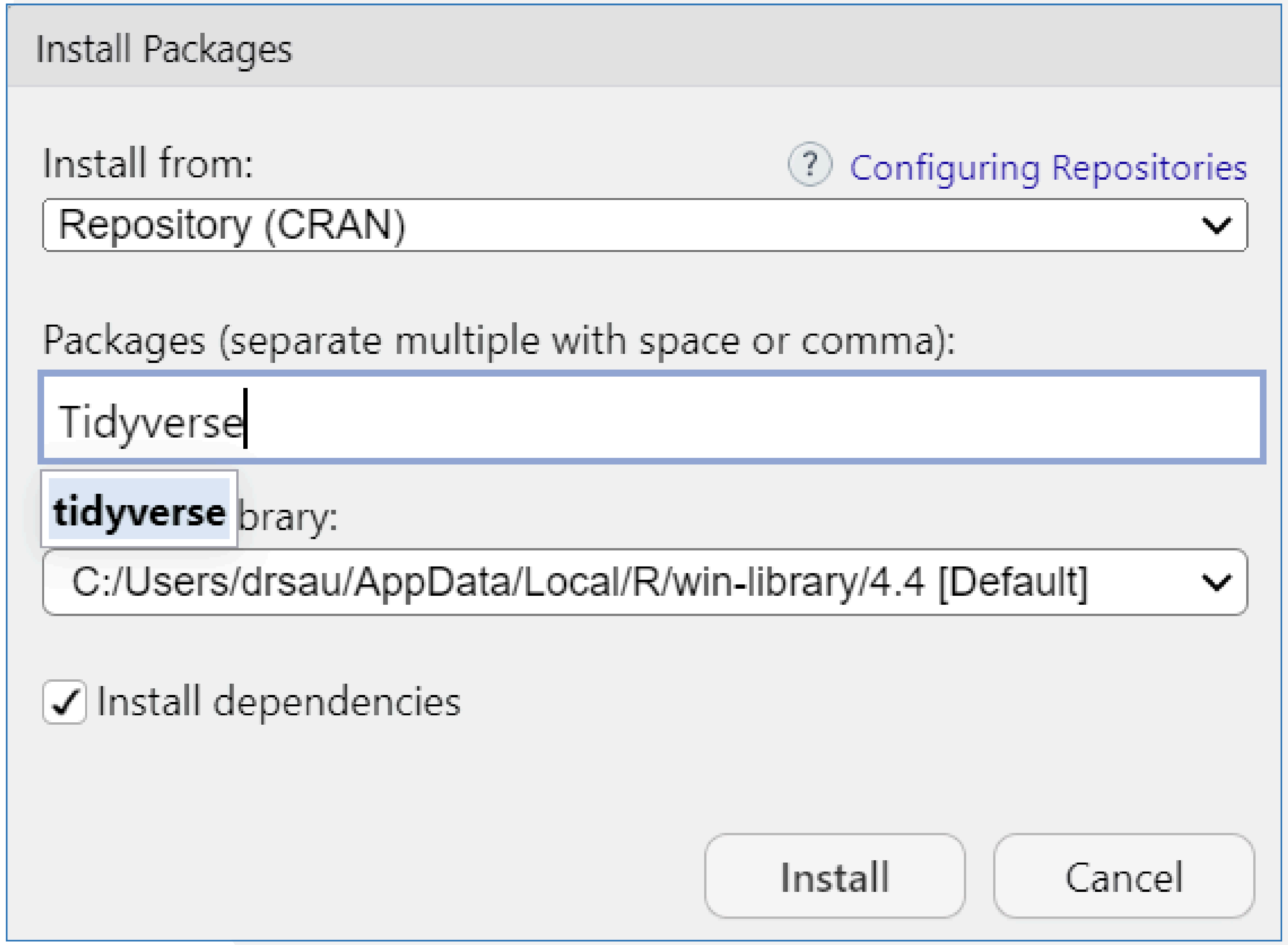
Method 2: Using the Packages Tab

In the **Packages** tab, click `**Install**` and choose the package you want to install.



User Library

	Name	Description	Version
□	abind	Combine Multidimensional Arrays	1.4-5
□	annotate	Annotation for microarrays	1.82.0
□	AnnotationDbi	Manipulation of SQLite-based annotations in Bioconductor	1.66.0
□	ape	Analyses of Phylogenetics and Evolution	5.8
□	aplore3	Datasets from Hosmer, Lemeshow and Sturdivant, "Applied Logistic Regression" (3rd Ed., 2013)	0.9



Install from: Repository (CRAN)

Packages (separate multiple with space or comma): Tidyverse

brary: C:/Users/drsau/AppData/Local/R/win-library/4.4 [Default]

☑ Install dependencies

R Packages

Loading the Packages

To use functions available in a package, you need to load the particular package:

In R script:

```
library(tidyverse)
```

*Installing packages only needs to be done once. However, loading the packages is required whenever you start a new **R script**.*

Read Data

To begin data analysis, we need to import data into **R**. The commonly used term for this process is *"**reading the data.**"* There are various types of data that require different packages to facilitate reading from different formats.

Comma-Separated Value (.csv) Files

The `**read_csv()**` function from the `**readr**` package, which is part of the `**tidyverse**`, can be used to read **CSV** files.

In R script:

```
library(readr)
mydata <- read_csv("data.csv")
```

Microsoft Excel (.xlsx) Files

To read **Excel** files, you can use the `**read_excel()**` function from the `**readxl**` package.

In R script:

```
library(readxl)
mydata <- read_excel("data.xlsx")
```

SPSS (.sav) Files

The `**haven**` package provides the `**read_sav()**` function to read **SPSS** files.

In R script:

```
library(haven)
mydata <- read_sav("data.sav")
```

Stata (.dta) Files

The `**haven**` package also supports reading **Stata** files using the `**read_dta()**` function.

In R script:

```
library(haven)
mydata <- read_dta("data.dta")
```

Keyboard Shortcuts

Execute Line

Ctrl Enter ↵

Save Your Script

Ctrl S

Undo Change

Ctrl Z

Select Full Script

Ctrl A

Search And Replace

Ctrl F

Go To End Of Line

End

Go To Start Of Line

Home

Open A File

Ctrl O

Assignment Operator

Alt —

Pipe Operator

Ctrl ⇧ Shift M

Insert Code Chunk

Ctrl Alt I

Show All Keyboard Shortcuts

Alt ⇧ Shift K

Thank You