

Programming for Data Science

How to create a package in R

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica



How to create a R package

- we describe how to create an R package using RStudio;
- Obviously it is possible to build R Packages with Command Line Tools, but RStudio has built-in tools that will make easier this task;
- the **devtools** package must be installed in RStudio.

```
>install.packages("devtools")
```

How to create a R package

- Open a new .R file;

```
File Edit Code View Plots Session Build Debug Profile Tools Help
File Edit Code View Plots Session Build Debug Profile Tools Help
prova.R* | Source on Save Run Source Environment History Build
1 sum <- function(x,y){  
2   return (x+y)  
3 }
```

Console ->[prova] >

```
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.
```

> |

Files Plot Packages Help Viewer

New Folder Delete Rename More

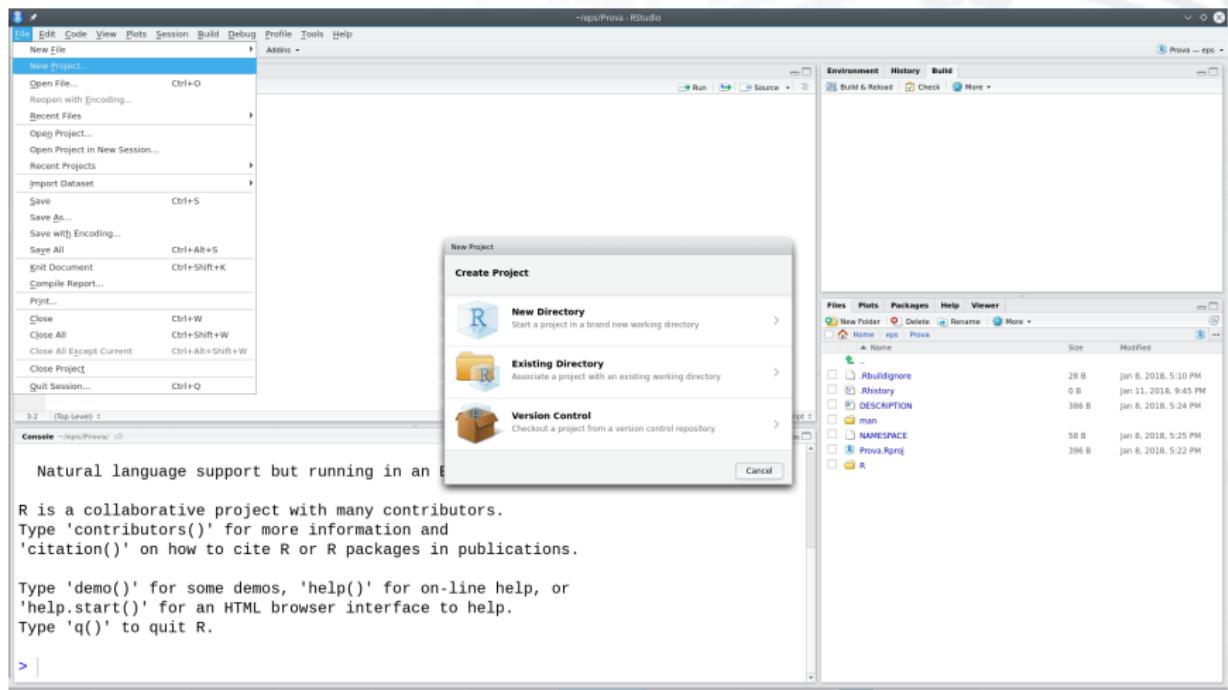
Name Size Modified

- ...
Absuldtignore 28 B jan 8, 2018, 5:10 PM
- DESCRIPTION 386 B jan 8, 2018, 5:24 PM
- man 58 B jan 8, 2018, 5:25 PM
- NAMESPACE 396 B jan 8, 2018, 5:22 PM
- R

- Write the code for your functions in this .R file. You can create one file with all of your functions or create separate files for each function. Save these files somewhere where you can easily find them.

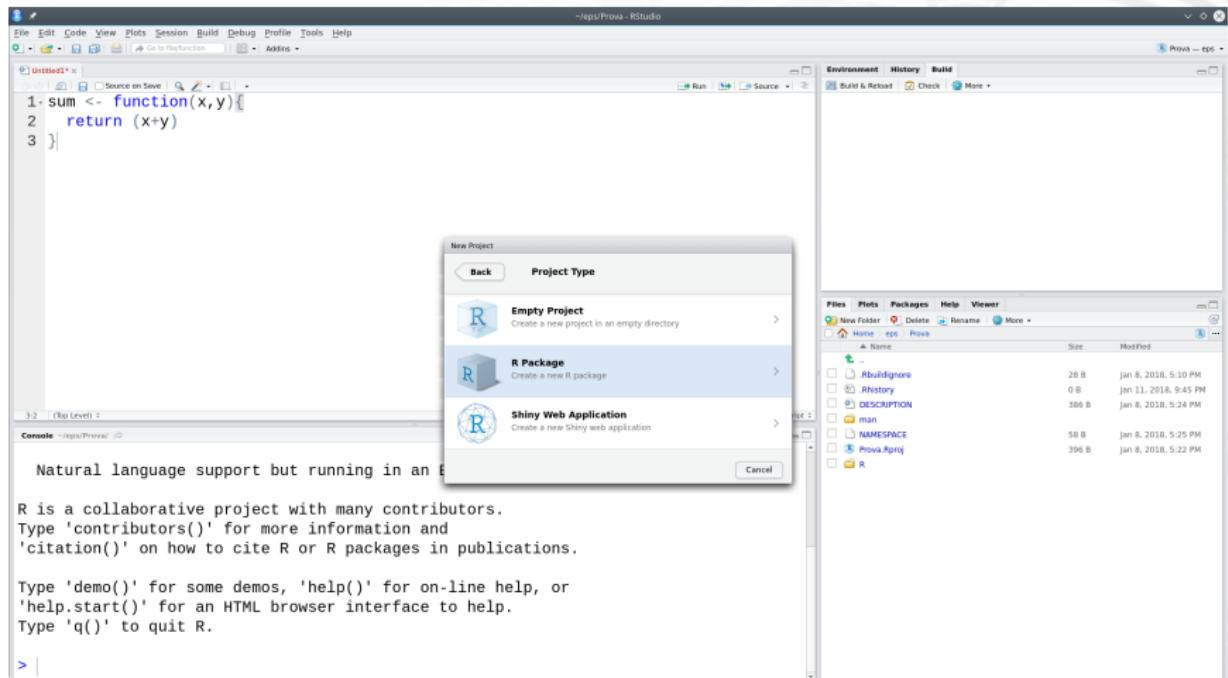
How to create a R package

- Open a new project in RStudio. Go to the **File** menu and click on **New Project**. Then select **New Directory**, and **R Package** to create a new R package.



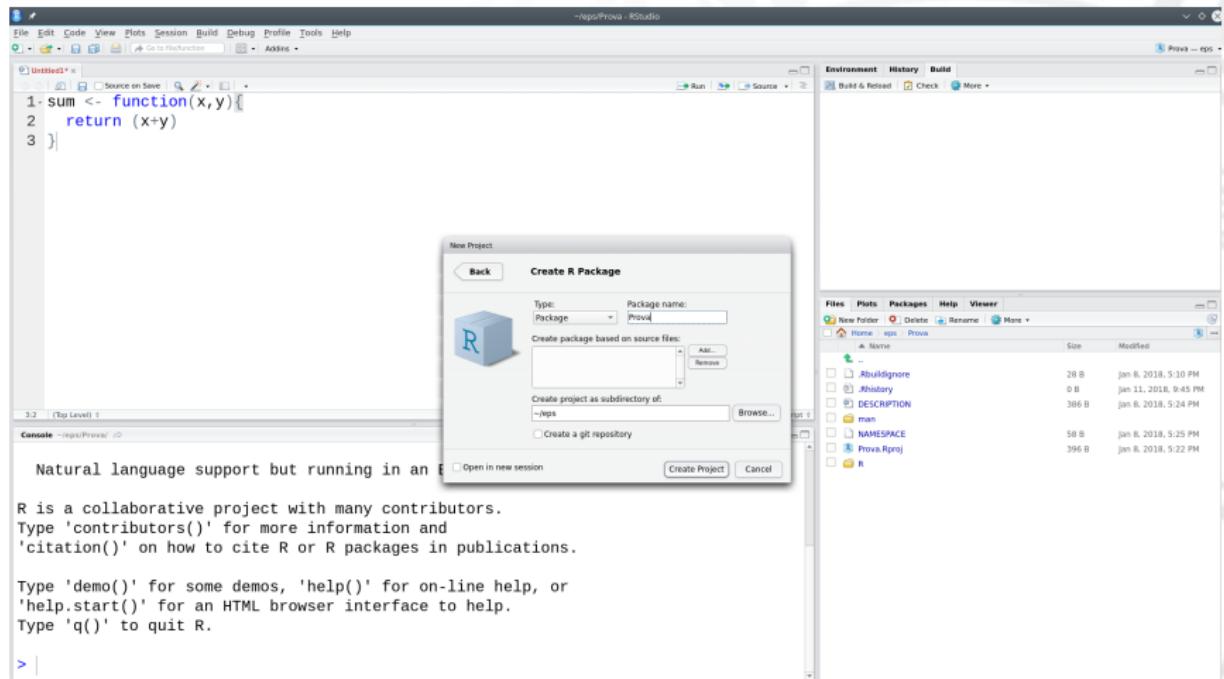
How to create a R package

- Open a new project in RStudio. Go to the **File** menu and click on **New Project**. Then select **New Directory**, and **R Package** to create a new R package.



How to create a R package

- Type the name of your package, then upload the .R file you created in step 1 under **Create package based on source files**. Click **Create project**.



How to create a R package

- Documentation can be automatically created using **roxygen2** package.

The screenshot shows the RStudio interface with the following details:

- Code Editor:** The file `prova.R` contains R code with roxygen2 documentation blocks. The code defines a function `sum` that takes two numeric vectors `x` and `y`, returns their sum, and includes a description, parameters, author information, and examples.
- Environment:** The environment pane shows "Environment is empty".
- File Explorer:** The sidebar shows the project structure: `Prova` (containing `RBuildIgnore`, `DESCRIPTION`, `main`, `NAMESPACE`, `Prova.Rproj`, and `R`).
- Console:** The console output includes:
 - 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.

```
> install.packages("roxygen2")
```

How to create a R package

- The functions must be commented using the following tags:

```
#' @title My sum function
#' @description a simple sum function
#' @param x is a numeric vector
#' @param y is a numeric vector
#' @author Name Family name, myemail [at] somewhere [dot] org, Affiliation
#' @return the sum of x and y
#'
#' @examples
#' \dontrun{
#'   sum(x,y)
#' }
#'
#' @export
sum <- function(x,y){
  return (x+y)
}
```

How to create a R package

- To enable the generation of package documentation using **roxygen2**

The screenshot shows the RStudio interface with the following details:

- Code Editor:** The file `prova.R` contains the following R code:

```
1 #' @title My sum function
2 #' @description a simple sum function
3 #' @param x is a numeric vector
4 #' @param y is a numeric vector
5 #' @author Name Family name, myemail [at] somewhere [dot] org, Affiliation
6 #' @return the sum of x and y
7 #
8 #' @examples
9 #' \dontrun{
10 #'   sum(x,y)
11 #' }
12 #
13 #' @export
14 sum <- function(x,y){
15   return (x+y)
16 }
```
- Project Options Dialog:** A modal dialog titled "Project Options" is open, specifically the "Roxygen Options" tab. It includes sections for "Build and Reload" (with options like `-mroxygen`), "Check Package" (with options like `-mroxygen`), "Build Source Package" (with options like `-mroxygen`), and "Build Binary Package" (with options like `-mroxygen`). It also has sections for "Use roxygen to generate:" (checkboxes for `Rd files`, `Collate file`, `NAMESPACE file`, and `Vignettes`) and "Automatically roxygenize when running:" (checkboxes for `R CMD check`, `Source and binary package builds`, and `Build & Reload`).
- Environment View:** Shows the global environment with objects like `deckeRseq`, `parser`, `deckeRseq --> GitLab`, `Prova`, `GitLab`, `Prova -> Downloads`, and `Prova-master`.
- Console:** Displays the message "Natural language support but running in an R environment". Below it, a note says "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."
- Spectacle Screenshot Capture Utility:** A small window is visible in the bottom-left corner.

How to create a R package

- Documentation is generated by clicking on **Build→ Documentation**

The screenshot shows the RStudio interface with the following details:

- File menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Build menu (highlighted):** Load All, Build and Reload, Clean and Rebuild, Test Package, Check Package, Build Source Package, Build Binary Package, Doc Generate (highlighted), Stop Build, Configure Build Tools...
- Console output:** Shows the command `devtools::document(roclets=c('rd', 'coolate', 'namespace'))` being run, followed by messages: "Updating Prova documentation", "Loading Prova", "Writing NAMESPACE", and "Documentation completed".
- File browser:** Shows the directory structure for the package: `Prova.Rproj` (containing `DESCRIPTION`, `NAMESPACE`, `man`, and `R` subfolders).
- Code editor:** Shows the R code for the `sum` function, which includes documentation blocks for parameters `x` and `y` and a value block for the sum.
- Environment tab:** Shows the current environment variables.
- History tab:** Shows the history of R commands run.
- Build tab:** Shows the build status and logs.

How to create a R package

- Package can be built and loaded by clicking on **Build**→ **Build and Reload**

The screenshot shows the RStudio interface with the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Left Panel:** A code editor showing R code for a package named 'sum'. The code includes a function definition for 'sum' and a help page for '?sum'. A context menu is open over the code, with "Build and Reload" highlighted.
- Right Panel:** An "Environment" tab showing the package structure and files: .Rbuildignore, DESCRIPTION, man, NAMESPACE, Prova.Rproj, and R. A log window displays the process of building the package, including converting help, preparing, installing help indices, and finding HTML links.
- Bottom Panel:** A "Console" window showing the R session starting, loading the Prova package, and attaching it. It also shows the masking of the 'sum' function from the base package.

How to create a R package

- Package can be built and loaded by clicking on **Build→ Build and Reload**

The screenshot shows the RStudio interface with the following components:

- Code Editor:** The left pane displays the file `prova.R` containing the code for a package. The code includes roxygen2 documentation blocks for functions `sum` and `sum2`.
- Console:** The bottom-left pane shows the R session starting and loading the package.
- Output:** The right pane shows the build log for the package "Prova". It includes messages about preparing, installing help indices, and building package indices, followed by a "DONE" message.
- Help Viewer:** The bottom-right pane shows the generated documentation for the `sum` function, detailing its usage, arguments, and value.

```
prova.R x sum.Rd x
1 % Generated by roxygen2: do not edit by hand
2 % Please edit documentation in R/prova.R
3 \name{sum}
4 \alias{sum}
5 \title[My sum function]{}
6 \usage{
7 sum(x, y)
8 }
9 \arguments{
10 \item{x}{is a numeric vector}
11 
12 \item{y}{is a numeric vector}
13 }
14 \value{
15 the sum of x and y
16 }

Console ->eps/Prova/ <->
Restarting R session...

> library(Prova)

Attaching package: 'Prova'

The following object is masked from 'package:base':

  sum

> ?sum
>

Environment History Build
Build & Reload Check More
converting help for package Prova
** preparing package for lazy loading
** help
*** installing help indices
  finding HTML links ...      sum
                                         html
done
** building package indices
** testing if installed package can be loaded
* DONE (Prova)

Files Plots Packages Help Viewer
R: My sum function <-> Find in Topics
sum (Prova) R Documentation

My sum function

Description
a simple sum function

Usage
sum(x, y)

Arguments
x is a numeric vector
y is a numeric vector

Value
the sum of x and y

Author(s)
Name Family name, myemail [at] somewhere [dot] org. Affiliation

Examples
#> sum(1, 2)
```