

Using *renv*

R code that works forever, everywhere

Urtzi Enriquez-Urzelai

2025-11-10

Outline

1 Introduction

- Why would you want to use *renv*?
- Test
- Test

2 Usage

- Initialize *renv*
- Installing and tracking packages

Code re-usability

Some nice introduction.

Test

Some nice introduction.

- more
- info
- here

First: block

This is an important warning message that appears at the bottom of the slide.

Then: alertblock

This is an important warning message that appears at the bottom of the slide.

Finally: exampleblock

This is an important warning message that appears at the bottom of the slide.

Test

Some nice introduction.

```
library(renv)  
activate()
```

- more
- info
- here

Is the same as

```
renv::activate()
```

Before initializing *renv*

```
.libPaths()
```

```
## [1] "/home/urtzi/.R"          "/usr/local/lib/R/site-library"
## [3] "/usr/lib/R/site-library"    "/usr/lib/R/library"
```

Before initializing *renv*

```
.libPaths()
```

```
## [1] "/home/urtzi/.R"          "/usr/local/lib/R/site-library"
## [3] "/usr/lib/R/site-library"    "/usr/lib/R/library"
```

And since we are using the system-wide library, we can load the installed packages. E.g.:

```
"cowsay" %in% rownames(installed.packages())
```

```
## [1] TRUE
```

Initialize *renv*

This will initialize the project, register the “essential” packages of a bare bone project, and create the `renv.lock` file

```
renv::init()
```

```
## - Linking packages into the project library ... Done!
## The following package(s) will be updated in the lockfile:
##
## # CRAN -----
## - renv    [* -> 1.1.5]
##
## The version of R recorded in the lockfile will be updated:
## - R        [* -> 4.5.0]
##
## - Lockfile written to "~/Documents/programming/R/using_renv/presentation/demo_renv_project/renv.lock".
```

After the initialization

```
list.files()  
## [1] ".renv"      ".Rprofile"   "renv.lock"
```

After the initialization

```
list.files()
```

```
## [1] ".renv"      ".Rprofile" "renv.lock"
```

And now, our library paths have changed!

```
.libPaths()
```

```
## [1] "/home/urtzi/Documents/programming/R/using_renv/presentation/demo_renv_project/.renv/library/linux-debian-trixie/R-4.5/x86_64-pc-linux-gnu/9a444a72"  
## [2] "/home/urtzi/.cache/R/renv/sandbox/linux-debian-trixie/R-4.5/x86_64-pc-linux-gnu/9a444a72"
```

Inspect the environment

But now we cannot use system-wide installed packages. Look at this:

```
"cowsay" %in% rownames(installed.packages())  
## [1] FALSE
```

Installing a new package

Let's install a package in our renv environment:

```
renv::install("cowsay")
```

```
## The following package(s) will be installed:  
## - cowsay [1.2.2]  
## - crayon [1.5.3]  
## - rlang [1.1.6]  
## These packages will be installed into "~/Documents/programming/R/using_renv/presentation/demo_renv_project/."  
##  
## # Installing packages -----  
## - Installing crayon ...          OK [linked from cache]  
## - Installing rlang ...           OK [linked from cache]  
## - Installing cowsay ...          OK [linked from cache]  
## Successfully installed 3 packages in 5.1 milliseconds.
```

Check if package is available

```
"cowsay" %in% rownames(installed.packages())
```

```
## [1] TRUE
```

Snapshot to save the state

Use `renv::snapshot()` to record the installed package in the lockfile:

```
renv::snapshot()
```

```
## - The lockfile is already up to date.
```

Now it's tracked!

```
renv::status()
```

```
## No issues found -- the project is in a consistent state.
```

The package is now in your `renv.lock` file and will be restored when others use `renv::restore()`.