

# R for Reproducibility

## R Packages for Reproducible Workflow

Daniela Palleschi

2024-02-19

### Table of contents

<b>1</b>	<b>here package</b>	<b>1</b>
<b>2</b>	<b>pacman package</b>	<b>2</b>
<b>3</b>	<b>rbbt package</b>	<b>2</b>
<b>4</b>	<b>renv package</b>	<b>3</b>
4.1	Activate or init . . . . .	3
4.2	Snapshot . . . . .	3
4.3	Restore . . . . .	4
4.4	Upgrade . . . . .	4
4.5	Update and Hydrate . . . . .	4
<b>5</b>	<b>usethis package</b>	<b>5</b>
<b>6</b>	<b>gitcreds package</b>	<b>5</b>

This is an overview of R packages that I use to maintain a reproducible workflow.

### 1 here package

The **here** package uses file path relative to the current working directory (Müller 2020). If you're working within a RProject (please do so!) then this working directory is the folder containing the .RProj file.

```
# install here package
install.packages("here")

# read in a csv file
readr::read_csv(here::here("folder", "subfolder", "subsubfolder", "file.csv"))
# or
readr::read_csv(here::here("folder/subfolder/subsubfolder/file.csv"))
```

## 2 pacman package

The `pacman` package was designed to facilitate loading and installing packages (Rinker and Kurkiewicz 2018). I like to use the `p_load()` function, which checks if listed packages are already installed. If so, it loads them (as with `library()`). If not, it installs and then loads them (as long as they're CRAN packages!). This facilitates sharing your code across machines or package libraries.

```
# install pacman
install.packages("pacman")

# install and/or load packages
pacman::p_load(here,
               renv,
               kableExtra)
```

## 3 rbbt package

The `rbbt` package contains an RStudio Addin for better integration of RStudio, Better Bibtex, and Zotero (Dunnington and Wiernik 2023). It can be installed via GitHub:

```
# install rbbt; remotes must first be installed
remotes::install_github("paleolimbot/rbbt")
```

You can follow this [blog post](#) from Gertjan Verhoeven (May 2021) on integrating RStudio and Zotero for writing in Rmarkdown, assuming only that you have RStudio installed.

Caveat: currently some `rbbt` functions conflict with Quarto's cross-references, which also begin with `@`. For Quarto documents, I mainly use the `rbbt` Addin to insert Zotero citations while writing using a keyboard shortcut (see the blogpost for tips on how to set this up).

## 4 renv package

The **renv** package stores your project-local environment (Ushey 2023). That is, it creates a time capsule of the R and package versions that you use *within an R Project*.

The benefits:

- your output is not dependent on whichever project version you currently have stored on your machine
  - e.g., if you come back to an analysis after a long time, the output should still be the same even if a certain package version has deprecated a function your analysis uses.
- makes the project *reproducible* for collaborators as well, who will likely have some different package versions (or will have missing packages) on their machine

The hard-to-get-your-head-around (imo):

- with each new R Project with a `renv.lock` file, you need to install all the necessary packages again (even if they're on your machine)
  - this is because each **renv** doesn't look beyond your project folder!
- remembering to update the `renv.lock` file frequently!

### Running **renv** functions

Remember to always run **renv** functions *in the console*! You do not want e.g., **renv** initialising a new `renv.lock` file every time you render your documents.

### 4.1 Activate or init

- to initialise a new *project-local environment*:

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

### 4.2 Snapshot

- update `renv.lock` file

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

After updating the `renv.lock` file, remember to commit/push these changes to git (if you're using it)!

#### Version control

After updating the `renv.lock` file (i.e., running `renv::snapshot()`), remember to commit/push these changes to git (if you're using git)! Otherwise, your `renv.lock` file will be outdated compared to your output.

### 4.3 Restore

- will restore your project to the most recent `renv.lock` file versions
  - this step follows `snapshot()`, which updates the `renv.lock` file
- very useful after updating R!

```
renv::restore()
```

### 4.4 Upgrade

- to update the `renv` package:

```
# upgrade renv version
renv::upgrade()
```

### 4.5 Update and Hydrate

```
# updates all packages (stored in renv.lock) in the renv cache
renv::hydrate(update = "all")

# update should have no effect now, but doesn't hurt to check
renv::update()

# now take a snapshot with the updated packages
renv::snapshot()
```

Table 1: Summary of useful commands from the `usethis` package

command	function
<code>git_token_help()</code>	print credential situation
<code>git_sitrep()</code>	print more verbose credential information
<code>git_vaccinate()</code>	add appropriate files to <code>.gitignore</code> (remember to stage, commit, and push)
<code>browse_github()</code>	view project GitHub
<code>browse_project()</code> / <code>_package()</code>	visit project-related pages
<code>browse_github_actions()</code>	view overview of GitHub actions
<code>git_credentials()</code>	print credentials
<code>create_github_token()</code>	create a Personal Access Token (PAT)
<code>gitcreds_set()</code>	add your PAT

Table 2: Summary of useful commands from the `gitcreds` package

command	function
<code>gitcreds_get()</code>	print git/GitHub credentials
<code>gitcreds_set()</code>	set/replace credentials
<code>gitcreds_delete()</code>	delete credentials

### 💡 Version control (repeated)

After updating the `renv.lock` file (i.e., running `renv::snapshot()`), remember to commit/push these changes to git (if you're using git)! Otherwise, your `renv.lock` file will be outdated compared to your output.

## 5 `usethis` package

The `usethis` package automates project set-up and development, and is also useful when creating a package (see the package [Documentation](#) for more info) (Wickham et al. 2023).

## 6 `gitcreds` package

Has some similar functionality to `usethis` with regards to using git/GitHub in an R project (Csárdi 2022). See the package [GitHub page](#) for more information.

Csárdi, Gábor. 2022. *Gitcreds: Query 'Git' Credentials from 'r'*. <https://CRAN.R-project.org/package=gitcreds>.

Dunnington, Dewey, and Brenton M. Wiernik. 2023. *Rbbt: R Interface to the Better BiBTeX Zotero Connector*. <https://github.com/paleolimbot/rbbt>.

- Müller, Kirill. 2020. *Here: A Simpler Way to Find Your Files*. <https://CRAN.R-project.org/package=here>.
- Rinker, Tyler W., and Dason Kurkiewicz. 2018. *pacman: Package Management for R*. Buffalo, New York. <http://github.com/trinker/pacman>.
- Ushey, Kevin. 2023. *Renv: Project Environments*. <https://CRAN.R-project.org/package=renv>.
- Wickham, Hadley, Jennifer Bryan, Malcolm Barrett, and Andy Teucher. 2023. *Usethis: Automate Package and Project Setup*. <https://CRAN.R-project.org/package=usethis>.