

Notes on R package development

Contents

1	Prerequisites	1
2	Set up the package root manually	1
3	Use <code>usethis</code> interactively	1
4	Initiate a github repository	2

1 Prerequisites

Package development is based on the helper package `usethis`, and `here` package to set the project root. The notes are based on the information provided in [this package online reference](#)

```
library("usethis")
library("here")
```

```
## here() starts at /home/baumanph/philipp Dropbox/Philipp Baumann/repro-science-guides
```

1. Create a folder where you want to start the self-contained package
2. Navigate to this folder and open a terminal from this folder
3. Start RStudio from the terminal

```
rstudio
```

After this, RStudio opens with the working directory in the package folder.

2 Set up the package root manually

Next, you can set the project root programatically, for example using the `here` package. You can also set up a package project using the **File -> New Project...** menu in RStudio.

To initiate a hidden `.here` file that sets the project root within the self-contained project folder, you can use:

```
here::set_here()
```

After setting the project root, you can store the active project for use for the remaining session. For details, consider [this help section](#)

To create the package template folder structure and key files you can run:

```
create_package(here())
```

This creates the package template and starts the RStudio project session in a new session.

3 Use `usethis` interactively

This `r-lib` online article explains further how to setup and invoke R package development workflow functions. It provides a good overview how to integrate git version control and how to install package building tools, e.g. compilers.

1. Set up interactive mode. This will open the `.Rprofile` file. This user profile is sourced into the project workspace when opening the project, which will automatically attach the `usethis` and `devtools` packages upon startup. `.Rprofile` belongs to one the *profile* type of files and contains R code. There is an other type of startup files, *environment files*, which contain lists of environment variables to be set. After running the the below command and when restarting, you will always have these two packages attached and you can directly use the functions without having to qualify with `usethis::<fun_name>`.

```
usethis::use_usethis()
```

2. Copy the following lines into the file:

```
if (interactive()) {  
  suppressMessages(require(devtools))  
  suppressMessages(require(usethis))  
}
```

4 Initiate a github repository