

Setting up R development environment on github

Ronald (Ryy) Glenn Thomas

2024-12-06

Table of contents

1	Introduction	1
2	Debugging workflow	3

1 Introduction

As is often the case a useful R function needs to be shared between data scientist co-workers or between a teacher and their students. The post describes a step by step methodology for how to wrap the function in a package and share it either on github or CRAN.

Start by using the various tools in the `usethis` package to streamline the process.

```
# install.packages(c("usethis", "devtools"))
install.packages("devtools")
# library(usethis)
library(devtools)
usethis::create_package("~/sbx/t2f")
```



Figure 1: purrr

```

julia sbx/t2f  tree --charset=ascii
.
|-- DESCRIPTION
|-- NAMESPACE
`-- R

julia sbx/t2f  more DESCRIPTION
Package: t2f
Title: What the Package Does (One Line, Title Case)
Version: 0.0.0.9000
Authors@R:
  person("First", "Last", , "first.last@example.com", role = c("aut", "cre"),
    comment = c(ORCID = "YOUR-ORCID-ID"))
Description: What the package does (one paragraph).
License: `use_mit_license()`, `use_gpl3_license()` or friends to pick a
  license
Encoding: UTF-8
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.3.2
julia sbx/t2f  more NAMESPACE
#kGenerated by roxygen2: do not edit by hand

```

Next copy the R file containing the function to the R directory and add a #' roxygen comment block to the top of the file. Then call `devtools::document()` to generate the man directory and the help page.

```
devtools::document()
```

```

julia sbx/t2f  tree --charset=ascii
.
|-- DESCRIPTION
|-- NAMESPACE
|-- R
|   `-- t2f.R
`-- man
    `-- t2f.Rd

```

The next step is to set up testing.

```
usethis::use_testthat()
```

```
call inside R  
usethis::use_test("t2f")
```

This open an editor. Enter the unit tests using the `test_that` function.

```
# inside editor called from use_test()  
test_that("t2f works correctly", {  
  expect_equal(t2f(iris, "iris"), 0)  
})
```

Set up a new repository on github.

```
git init  
git add .  
git commit -m "Initial commit"
```

```
usethis::use_github()  
usethis::use_mit_license("R.G.Thomas")  
usethis::use_readme_md()  
usethis::use_code_of_conduct("rgthomas@ucsd.edu")  
usethis::use_tidy_contributing()  
usethis::use_package("kableExtra", type = "Imports")  
devtools::check()
```

2 Debugging workflow

```
git checkout -b fix-bug
```

Summary Workflow

- – Debug locally and isolate the issue.
- – Create a local branch (fix-t2f-issue) for the fix.
- `git checkout -b fix-t2f-issue`

- – Make and test the changes.
- Run `devtools::test()` to confirm all tests pass.
- Use `devtools::check()` to validate the package.
- – Push the branch to GitHub.
-
- `git add .`
- `git commit -m "Fix issue with t2f function"`
- – Open a Pull Request.
- – Merge the branch into the main branch and clean up.
- `git branch -d fix-t2f-issue`
- `git push origin --delete fix-t2f-issue`
- – Update the version number and push the final changes.

approach ensures proper testing, collaboration, and version control.