The developer toolkit

Physalia course 2023

Instructor: Jacques Serizay

Bioconductor version

To develop an R/Bioc package, developers must be careful of the weather…

https://contributions.bioconductor.org/use-devel.html#use-devel

Package authors should develop against the version of R that will be available to users when the Bioconductor devel branch becomes the Bioconductor release branch.

R has a '.y' release in x.y.z every year (typically mid-April), but Bioconductor has a .y release (where current devel becomes release) every 6 months (mid-April and mid-October).

This means that:

- from mid-October through mid-April, Bioconductor developers should be developing against R-devel
- from mid-April to mid-October, developers should use R-release for Bioconductor development.

Bioconductor version

To develop an R/Bioc package, developers must be careful of th

https://contributions.bioconductor.org/use-de

Package authors sh … s
when the Bi

… s a .y
… pril and mid-

… pril, Bioconductor developers should be
… er

- … mid-October, developers should use R-release
… ductor development.

BEWARE
R/BIOC VERSION

The (Bioc.) developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

- devtools
    - usethis
    - remotes
    - testthat

The (Bioc.) developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

- devtools
    - usethis
    - remotes
    - testthat
- BiocCheck
- Biocthis

The (Bioc.) developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

devtools

- usethis
- remotes
- testthat

BiocCheck

Biocthis

RStudio

- ➢ And many more…: https://indrajeetpatil.github.io/awesome-r-pkgtools/

Devtools and co

devtools makes package development easier by providing R functions that simplify and expedite common tasks.

Devtools and co

devtools makes package development easier by providing R functions that simplify and expedite common tasks.

- For **interactive use**, users should attach devtools and think of it as the provider of your favorite functions for package development.

```
library(devtools)
#Loading required package: usethis
load_all()
```

Devtools and co

devtools makes package development easier by providing R functions that simplify and expedite common tasks.

- For **interactive use**, users should attach devtools and think of it as the provider of your favorite functions for package development.

```r
library(devtools)
#Loading required package: usethis
load_all()
```

- For programmatic use, such as inside another package, developers should **NOT** depend on devtools, but should instead access functions via the package that is their primary home.

```r
pkgload::load_all()
```

Devtools and co

devtools makes package development easier by providing R functions that simplify and expedite common tasks.

```
devtools’ NAMESPACE:


importFrom(cli,cat_bullet)
importFrom(cli,cat_rule)
importFrom(ellipsis,check_dots_used)
importFrom(lifecycle,deprecated)
importFrom(memoise,memoise)
importFrom(miniUI,miniPage)
importFrom(pkgbuild,clean_dll)
importFrom(pkgbuild,find_rtools)
importFrom(pkgbuild,has_devel)
importFrom(pkgbuild,with_debug)
importFrom(pkgload,check_dep_version)
importFrom(pkgload,parse_deps)
importFrom(pkgload,unload)
importFrom(profvis,profvis)
importFrom(remotes,dev_package_deps)
importFrom(remotes,github_pull)
importFrom(remotes,github_release)
importFrom(remotes,install_bioc)
importFrom(remotes,install_bitbucket)
importFrom(remotes,install_cran)
importFrom(remotes,install_dev)
importFrom(remotes,install_git)
importFrom(remotes,install_github)
importFrom(remotes,install_gitlab)
importFrom(remotes,install_local)
importFrom(remotes,install_svn)
importFrom(remotes,install_url)
importFrom(remotes,install_version)
importFrom(remotes,update_packages)
importFrom(sessioninfo,package_info)
importFrom(sessioninfo,session_info)
importFrom(stats,update)
importFrom(urlchecker,url_check)
importFrom(usethis,ui_code)
importFrom(usethis,ui_done)
importFrom(usethis,ui_field)
importFrom(usethis,ui_path)
importFrom(usethis,ui_todo)
importFrom(usethis,ui_value)
importFrom(usethis,use_test)
importFrom(utils,available.packages)
importFrom(utils,contrib.url)
importFrom(utils,install.packages)
importFrom(utils,installed.packages)
importFrom(utils,modifyList)
importFrom(utils,packageDescription)
importFrom(utils,packageVersion)
importFrom(utils,remove.packages)
```

> “One Ring to rule them all”
> – The Free People

The Bioc. developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

devtools

- usethis → Fastening early package development
- remotes
- testthat

BiocCheck

Biocthis

RStudio

- And many more…: https://indrajeetpatil.github.io/awesome-r-pkgtools/

The Bioc. developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

devtools

- usethis
- remotes → Installing packages from anywhere
- testthat

BiocCheck

Biocthis

RStudio

- And many more…: https://indrajeetpatil.github.io/awesome-r-pkgtools/

Some useful functions

```
>remotes::install_*                       remotes::install_local                     remotes::install_remote
                                          remotes::install_dev
remotes::install_gitlab                   remotes::install_url
remotes::install_github                   remotes::install_svn
remotes::install_cran                     remotes::install_version
remotes::install_deps                     remotes::install_bioc
remotes::install_bitbucket                remotes::install_git
```

```
>usethis::use_*                           usethis::use_description                   usethis::use_package_doc
                                          usethis::use_git_credentials               usethis::use_github_actions
usethis::use_gpl_license                  usethis::use_test                          usethis::use_git_config
usethis::use_git_ignore                   usethis::use_pipe                          usethis::use_usethis
usethis::use_version                      usethis::use_agpl_license                  usethis::use_pkgdown
usethis::use_roxygen_md                   usethis::use_tidy_upkeep_issue             usethis::use_testthat
usethis::use_github_action                usethis::use_github_links
usethis::use_github                       usethis::use_description_defaults
usethis::use_data                         usethis::use_bioc_badge
usethis::use_tibble                       usethis::use_pkgdown_github_pages
usethis::use_vignette                     usethis::use_tidy_github_actions
usethis::use_readme_md                    usethis::use_coverage
```

The Bioc. developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

devtools

- usethis
- remotes
- **testthat** → Adding unit testing in later package development

BiocCheck

Biocthis

RStudio

➢ And many more…: https://indrajeetpatil.github.io/awesome-r-pkgtools/

The Bioc. developer toolkit

https://r-pkgs.org/

https://contributions.bioconductor.org/index.html

devtools

- usethis
- remotes
- testthat

BiocCheck

Biocthis → Extra Bioconductor-specific features

RStudio

➢ And many more…: https://indrajeetpatil.github.io/awesome-r-pkgtools/

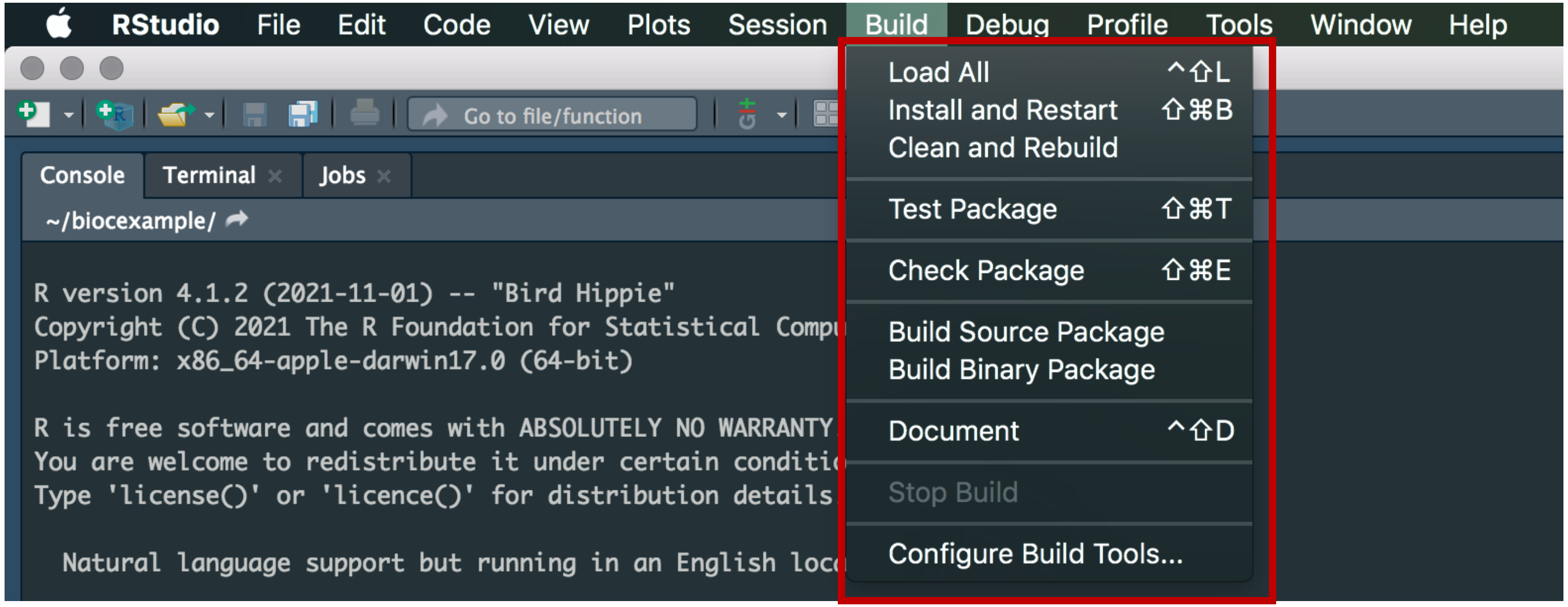
Some useful functions

```
>testthat::expect_*                        testthat::expect_snapshot                  testthat::expect_false
                                           testthat::expect_snapshot_output           testthat::expect_is
testthat::expect_match                     testthat::expect_true                      testthat::expect_type
testthat::expect_snapshot_warning          testthat::expect_that                      testthat::expect_output_file
testthat::expect_snapshot_error            testthat::expect_error                     testthat::expect_snapshot_file
testthat::expect_equal                     testthat::expect_lte                       testthat::expect_success
testthat::expect_lt                        testthat::expect_reference                 testthat::expect_output
testthat::expect_cpp_tests_pass            testthat::expect_s4_class                  testthat::expect_gte
testthat::expect_condition                 testthat::expect_setequal                  testthat::expect_gt
testthat::expect_equivalent                testthat::expect_invisible                 testthat::expect_known_output
testthat::expect_failure                   testthat::expect_null                      testthat::expect_vector
testthat::expect_visible                   testthat::expect_message                   testthat::expect_identical
testthat::expect_s3_class                  testthat::expect_snapshot_value            testthat::expect_no_match
testthat::expect_known_hash                testthat::expect_mapequal
testthat::expect_length                    testthat::expect_silent
testthat::expect_less_than                 testthat::expect_more_than
testthat::expect_equal_to_reference        testthat::expect_known_value
testthat::expect_warning                   testthat::expect_named
```

```
>biocthis::*                               biocthis::use_bioc_citation                biocthis::use_bioc_issue_template
                                           biocthis::use_bioc_github_action           biocthis::use_bioc_readme_rmd
biocthis::use_bioc_description             biocthis::use_bioc_pkg_templates           biocthis::use_bioc_coc
biocthis::bioc_style                       biocthis::biocthis_example_pkg             biocthis::use_bioc_news_md
biocthis::use_bioc_vignette                biocthis::use_bioc_support
```

Rstudio and package development

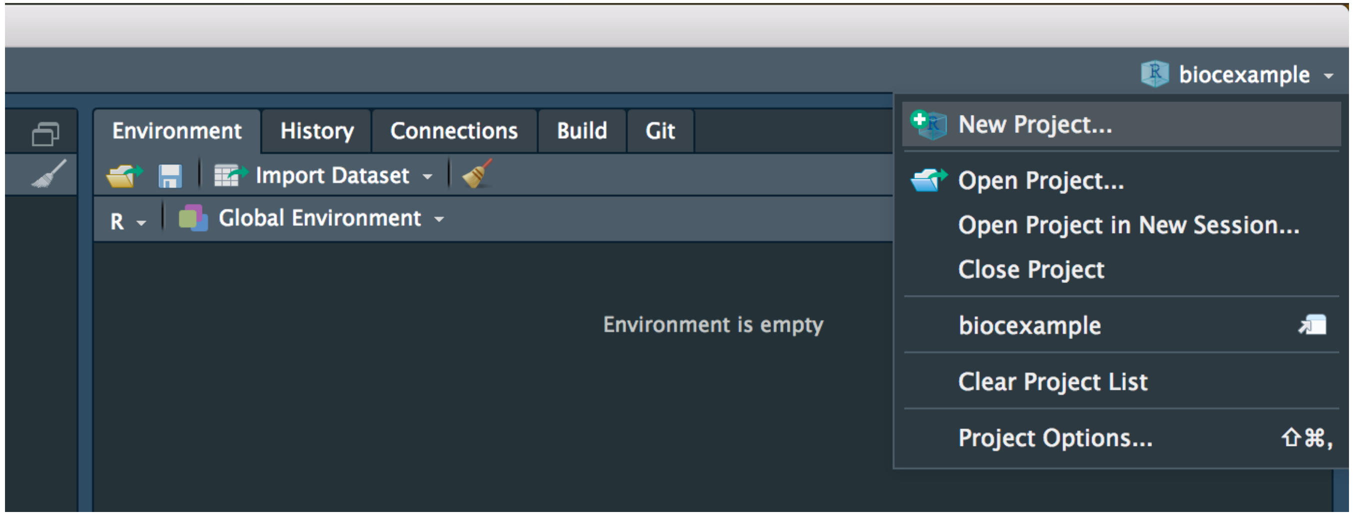
"devtools works hand-in-hand with RStudio, which we [Hadley] believe is the best development environment for most R users. To be clear, you can use devtools without using RStudio and you can develop packages in RStudio without using devtools. But there is a special, two-way relationship that makes it very rewarding to use devtools and RStudio together."



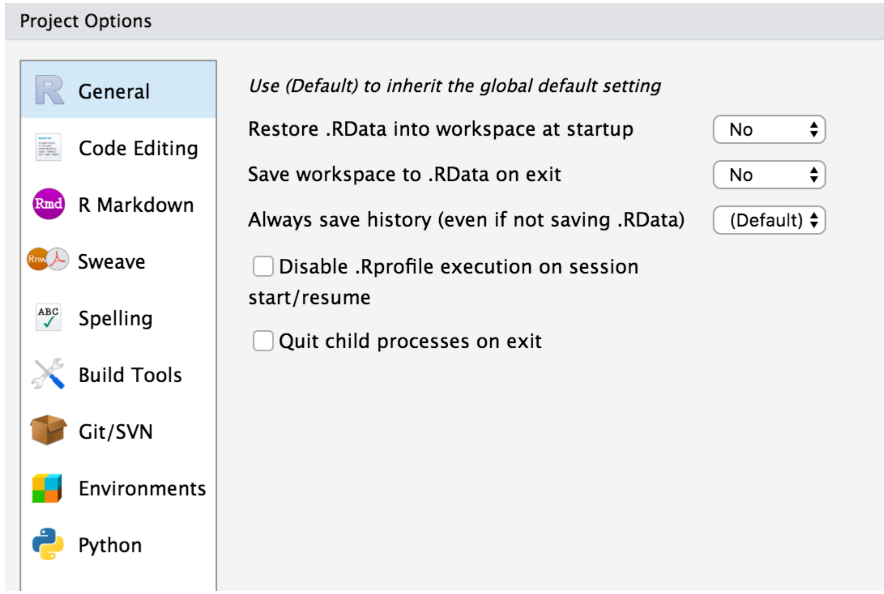
```
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Compu
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY
You are welcome to redistribute it under certain conditi
Type 'license()' or 'licence()' for distribution details

  Natural language support but running in an English loc
```

Rstudio and projects (.Rproj files)

- Projects are very “launch-able”. It’s easy to fire up a fresh instance of RStudio in a Project, with the file browser and working directory set exactly the way you need, ready for work.
- Each Project is isolated. You can have several RStudio Projects open at once and code executed in Project A does not have any effect on the R session and workspace of Project B.
- You get handy code navigation tools like F2 to jump to a function definition and Ctrl + . to look up functions or files by name.
- You get useful keyboard shortcuts and a clickable interface for common package development tasks, like generating documentation, running tests, or checking the entire package.



Environment is empty



Use (Default) to inherit the global default setting

Restore .RData into workspace at startup: No

Save workspace to .RData on exit: No

Always save history (even if not saving .RData): (Default)

☐ Disable .Rprofile execution on session start/resume

☐ Quit child processes on exit

Rstudio and projects (.Rproj files)

Working with projects ensures:

1. That you are not accidently loading additional variables when opening Rstudio after rebooting
2. That your package root is **always** correct!
3. Many more convenient stuff you don’t actually want to realize it would need to be taken care of without projects (e.g. package dependencies, versions, …)

```
> library(devtools)
> usethis::create_package()
> usethis::use_rstudio()
> usethis::proj_sitrep()
```