Lecture 28: Nov 28, 2018

R Packages

- *Motivation*
- *Creating an R Package*
  - *RStudio Cloud* and *RStudio Desktop*
- *Structure of R Packages*

James Balamuta
STAT 385 @ UIUC

Announcements

- **Group Project Final Report, Demo Video,** and **Peer Evaluation** due **Tuesday, December 18th** at **10:00 PM**
  - Details: http://stat385.stat.illinois.edu/group-projects/
- **hw09** due **Friday, November 30th** at **6:00 PM**
- **Quiz 13** covers Week 12 contents @ CBTF.
  - Window: Nov 27th - Nov 29th
  - Sign up: https://cbtf.engr.illinois.edu/sched
- Want to review your homework or quiz grades? **Schedule an appointment.**

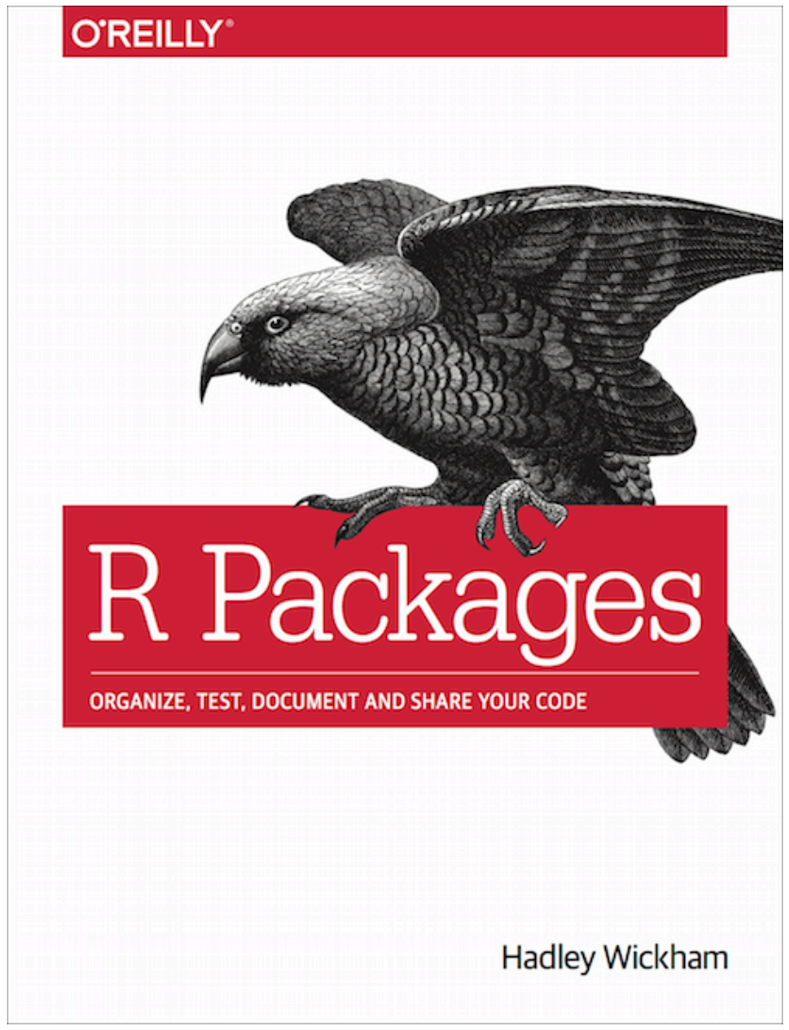
Last Time

- **OOP**
  - **O**bject **o**riented **p**rogramming emphasizes objects in a collection with interactions.
  - Focuses on **encapsulation**, **inheritance**, and **polymorphism**.
- **S3 Objects**
  - Each object has an underlying **class** attribute.
  - **Generic functions** are used to dispatch to the appropriate method function for an object.
- **Unpaired (Two-Sample) t-Test**
  - Case study in implementing a method for assessing group difference.

Lecture Objectives

- Explain the **benefits** of *R* packages
- **Structuring** code for an *R* package
- **Developing** unit tests for functions

Motivation



"Packages are the **fundamental units of reproducible** *R* code. They include reusable *R* functions, the documentation that describes how to use them, and sample data."

– Hadley Wickham in R packages

R Code vs. *R* Packages

… in the context of shipping containers …

Source

Source

Source

Source

13,468 *R* packages 

exist on Comprehensive R Archival Network (CRAN)

R Package Structure

… extending the *R* language …

Data

trees

Girth	Height	Volume

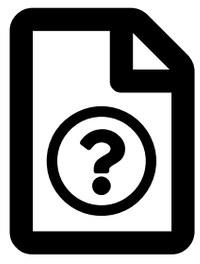
mtcars

mpg	…	gear	carb

…

Help





…

Functions

mean(x, …)
sd(x, …)
sum(x, …)
prod(x, …)
plot(x, y, …)
…

Why an *R* Package?

… why not ???

1. Code is already organized via functions.

2. Nothing already exists

3. Give back to the community: **pay it forward**

4. Portfolio of your coding and development skills.

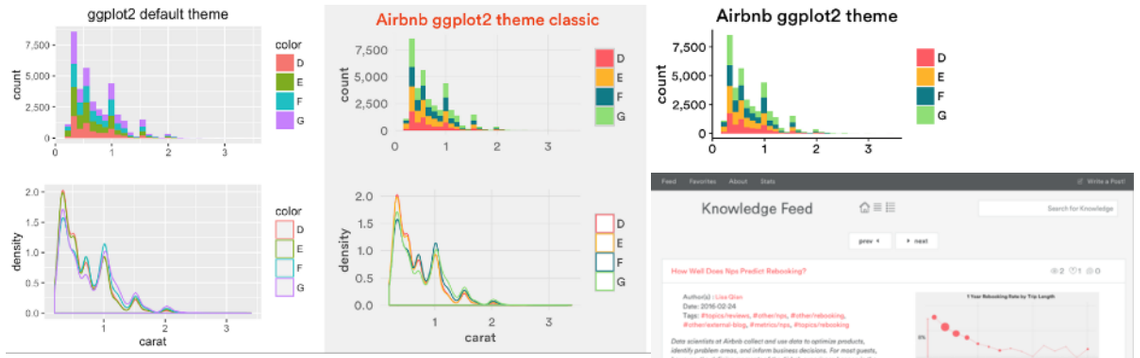
R Script vs. R Package

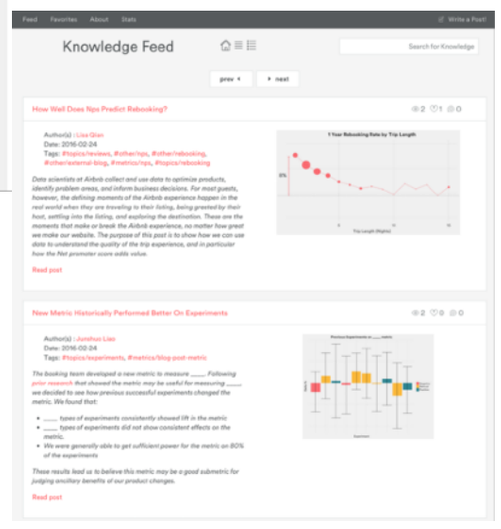
… the battle for the ages …

R Script	R Package
Focus on **one** data analysis project	Generalized to be used **on more than one** data analysis project.
Creates analysis results	Builds new *R* objects
Pollutes the Global Environment	Chunked logic through functions

Rbnb

… industry example of Airbnb designing an *R* package …



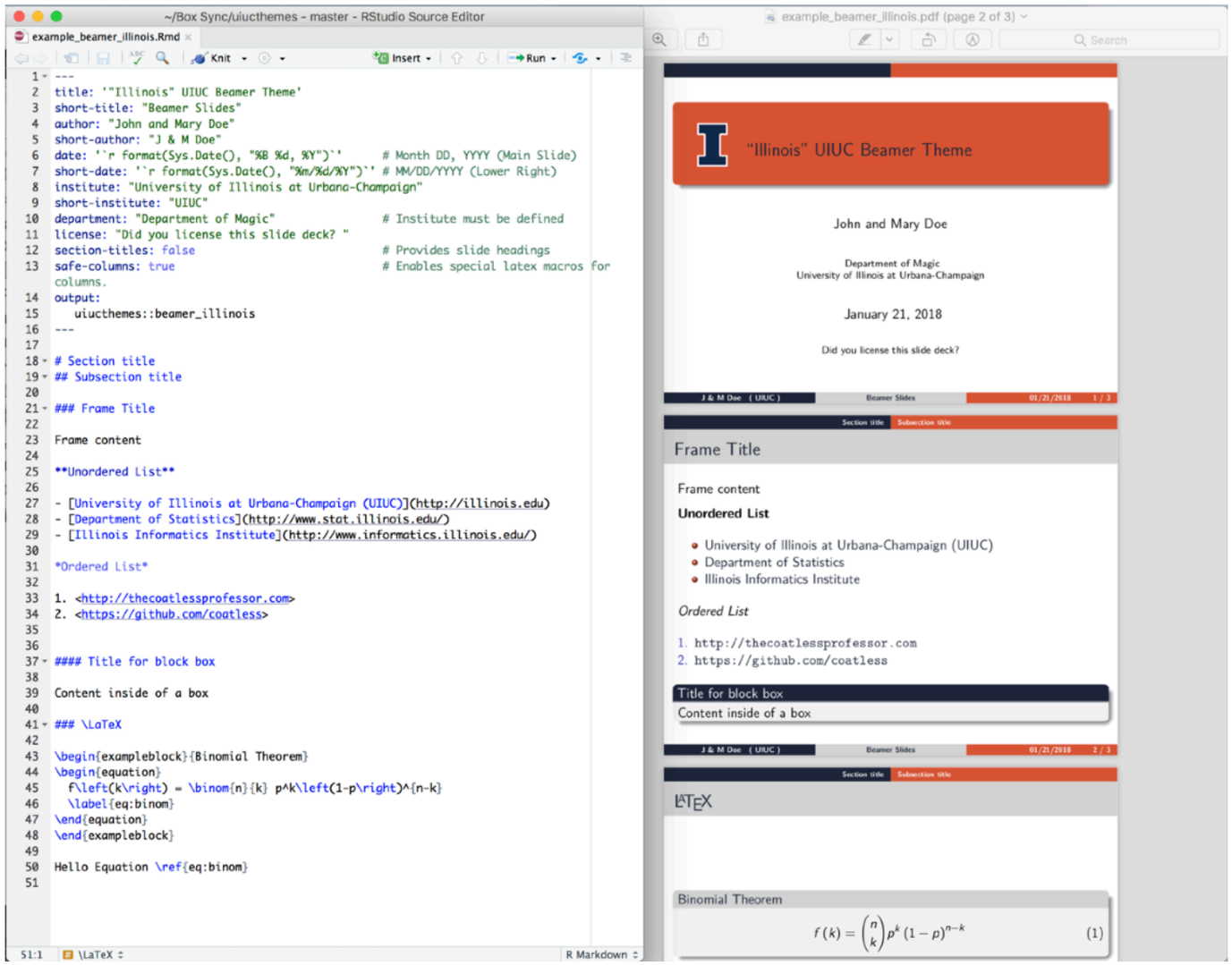


- rstudio::conf 2017 - Airbnb @ 10:47
- Using R packages and education to scale Data Science at Airbnb
- Airbnb visualization techniques
- Airbnb Knowledge Repo
- PeerJ: How R Helps Airbnb Make the Most of Its Data

Internal R Packages

… more guidelines …

[RStudio Views: Some Ideas for your Internal R Package]



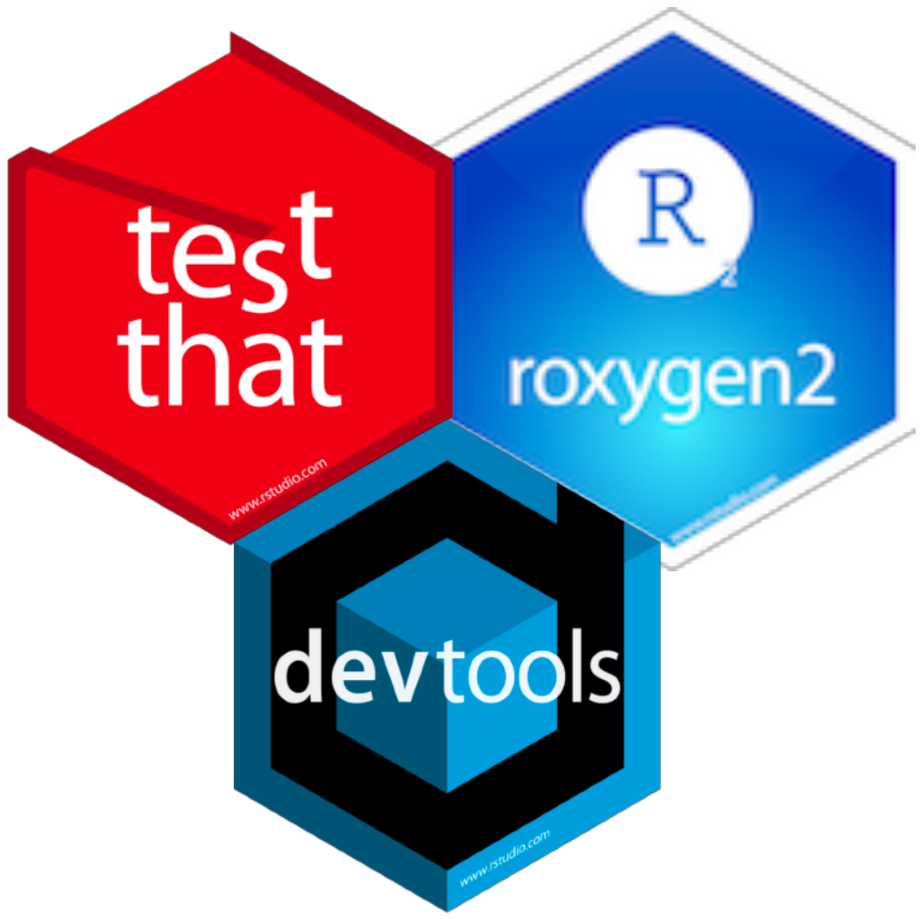
https://github.com/coatless/uiucthemes

Creating an *R* Packages

Tools to Develop Packages

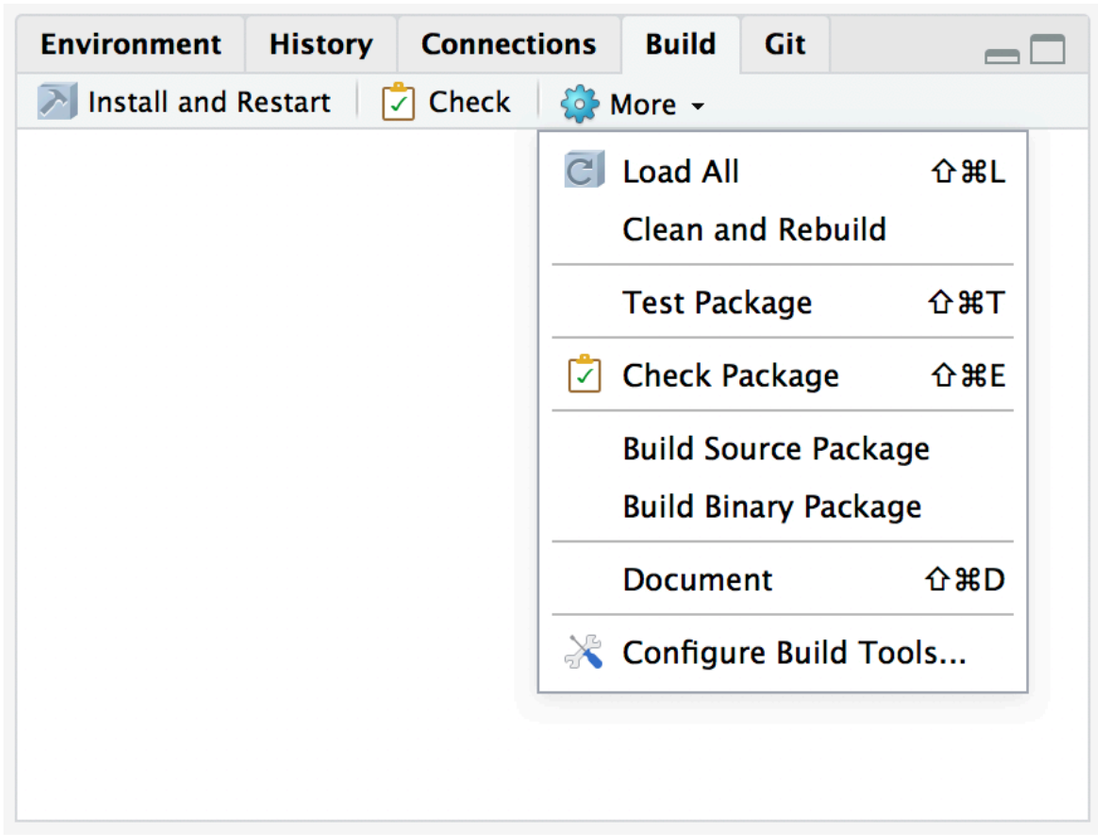
… useful *R* packages for writing *R* packages …

```
install.packages(c("devtools", "testthat", "roxygen2"))
```



Package Build Tab

… RStudio's built in project support for packages …

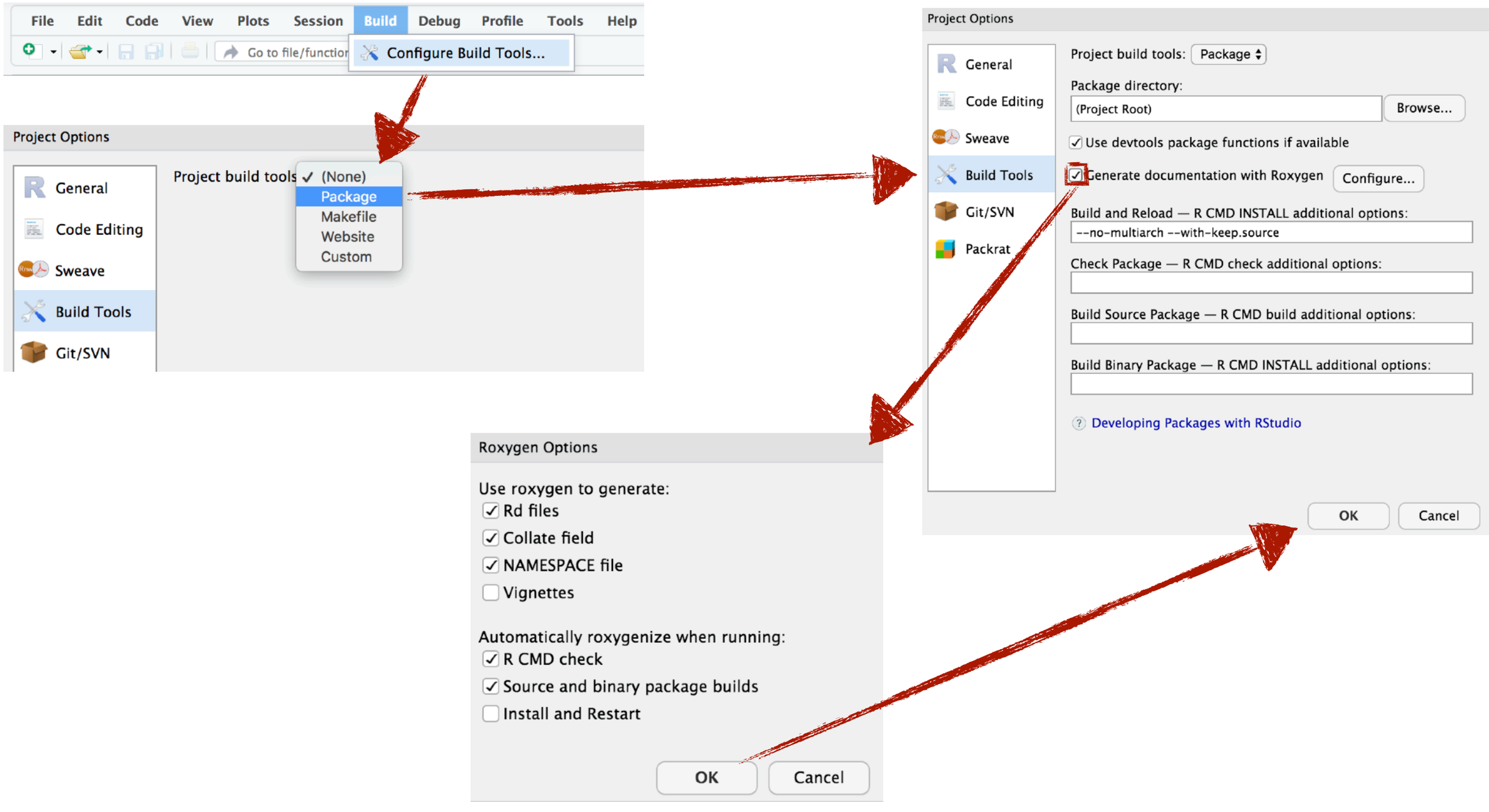


(how to enable the tab is shown next)


RStudio Cloud

Creating an *R* Package

… in RStudio **Cloud** …



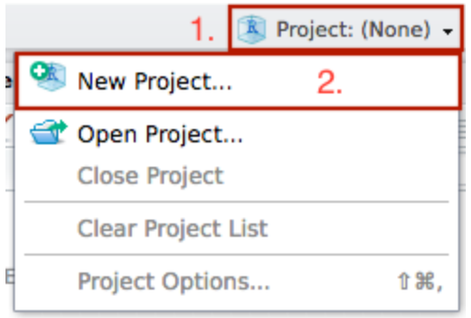
Then run in *R* ….

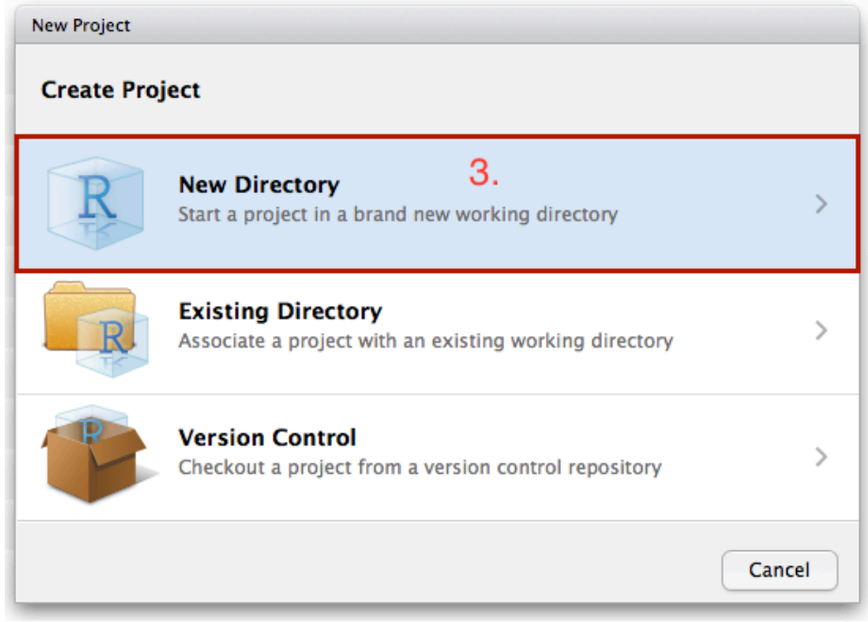
```
devtools::setup()
```

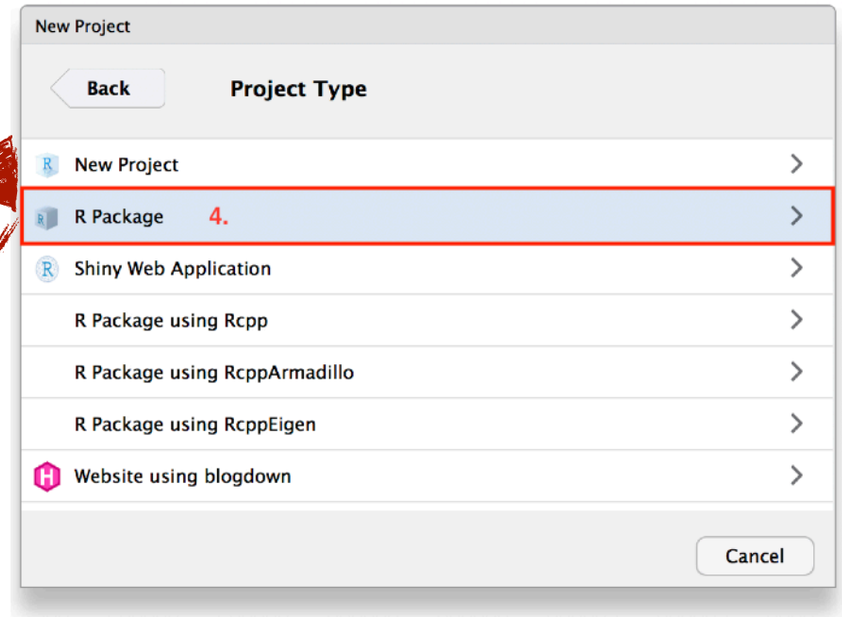
Desktop / Server

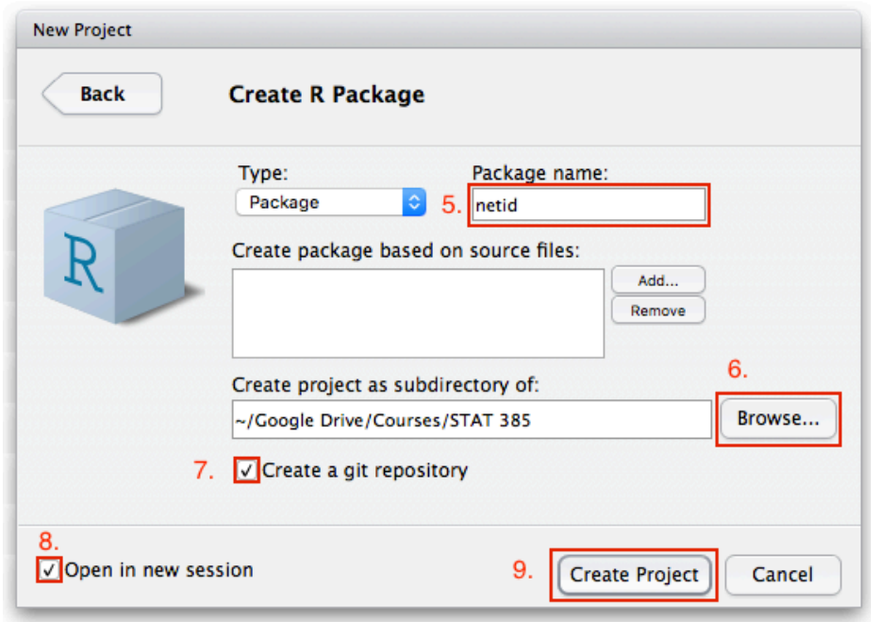
Creating an *R* Package

… in RStudio **Desktop** / **Server** …



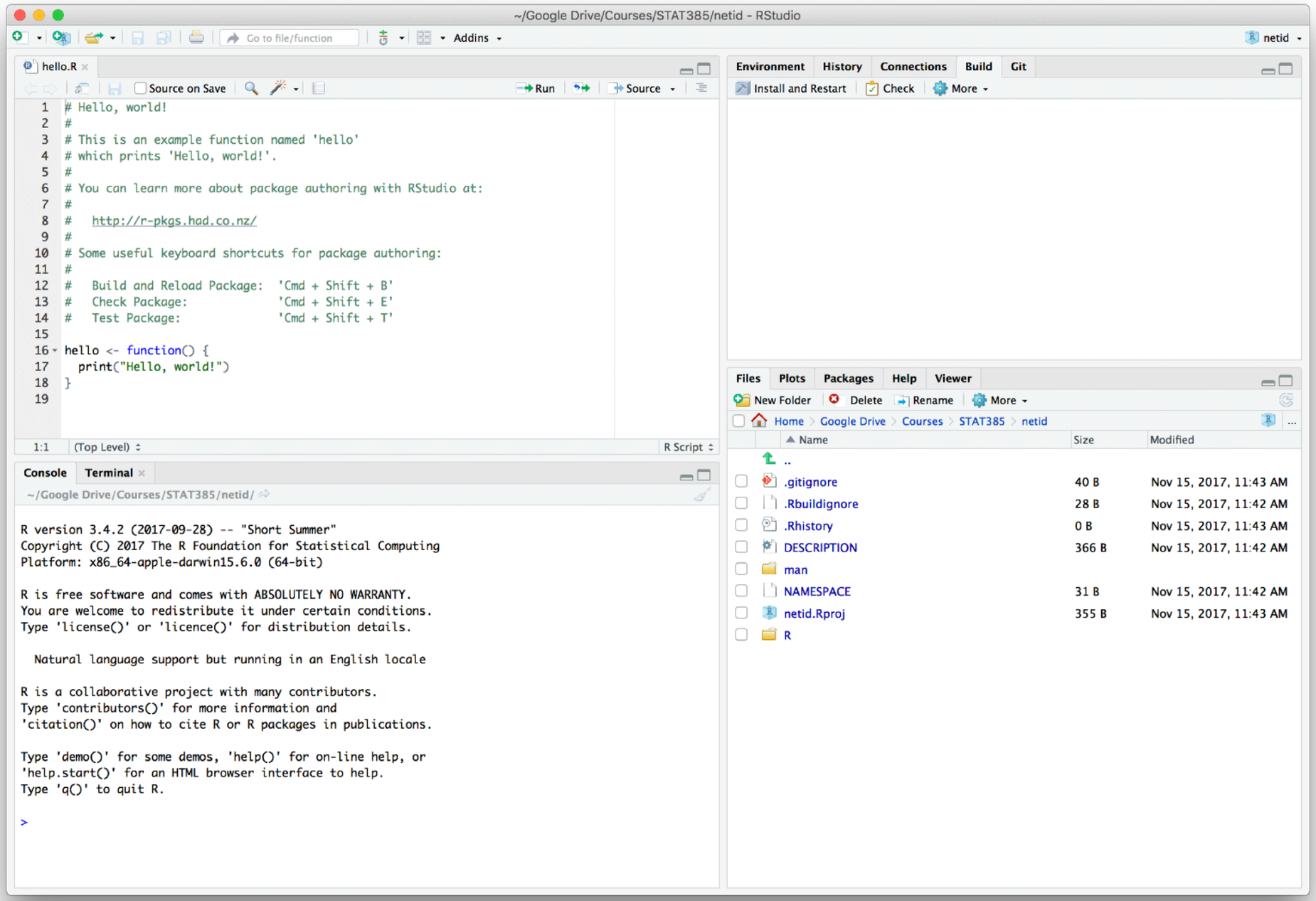






Creating an *R* Package

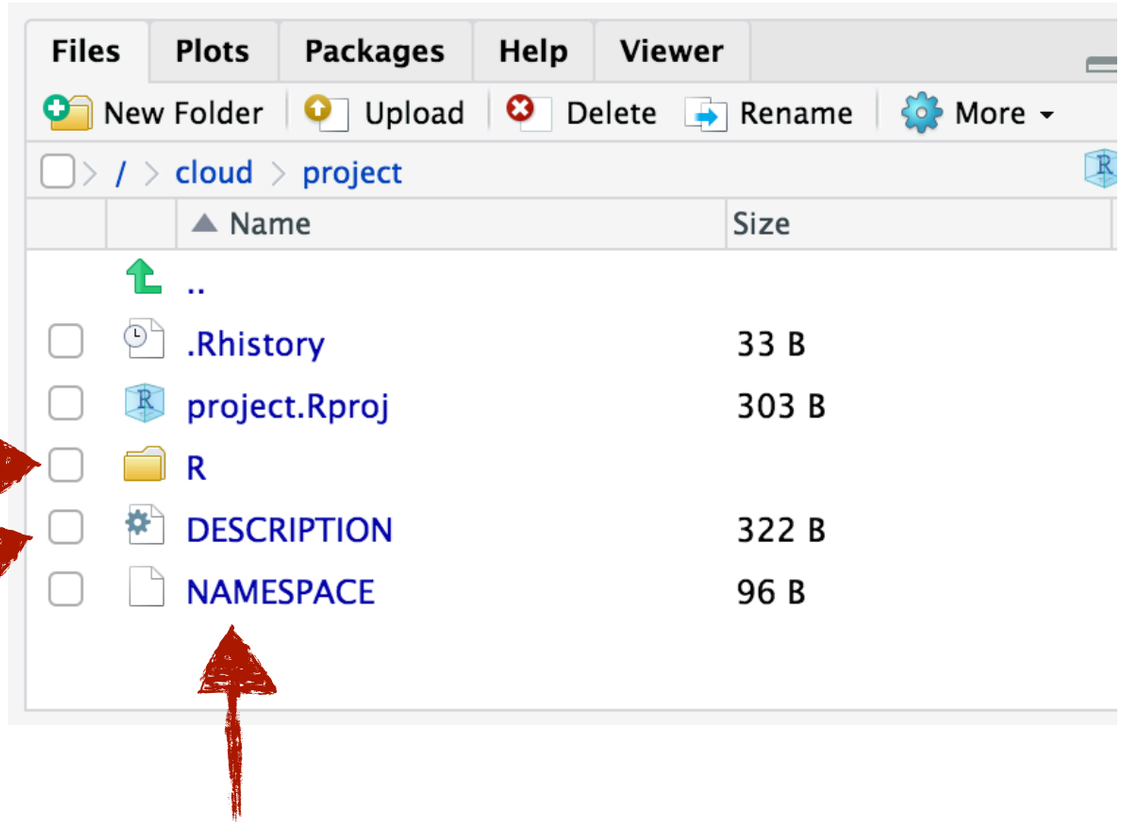
… in RStudio **Desktop** / **Server** …



Structure of *R* Packages

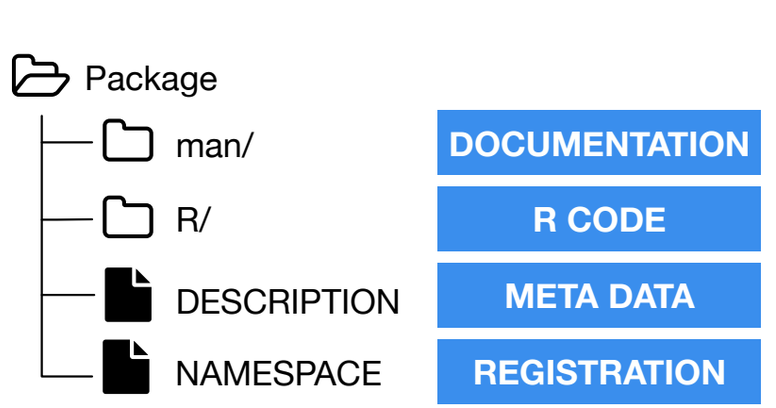
Overview

… looking inside an R package …



R Code

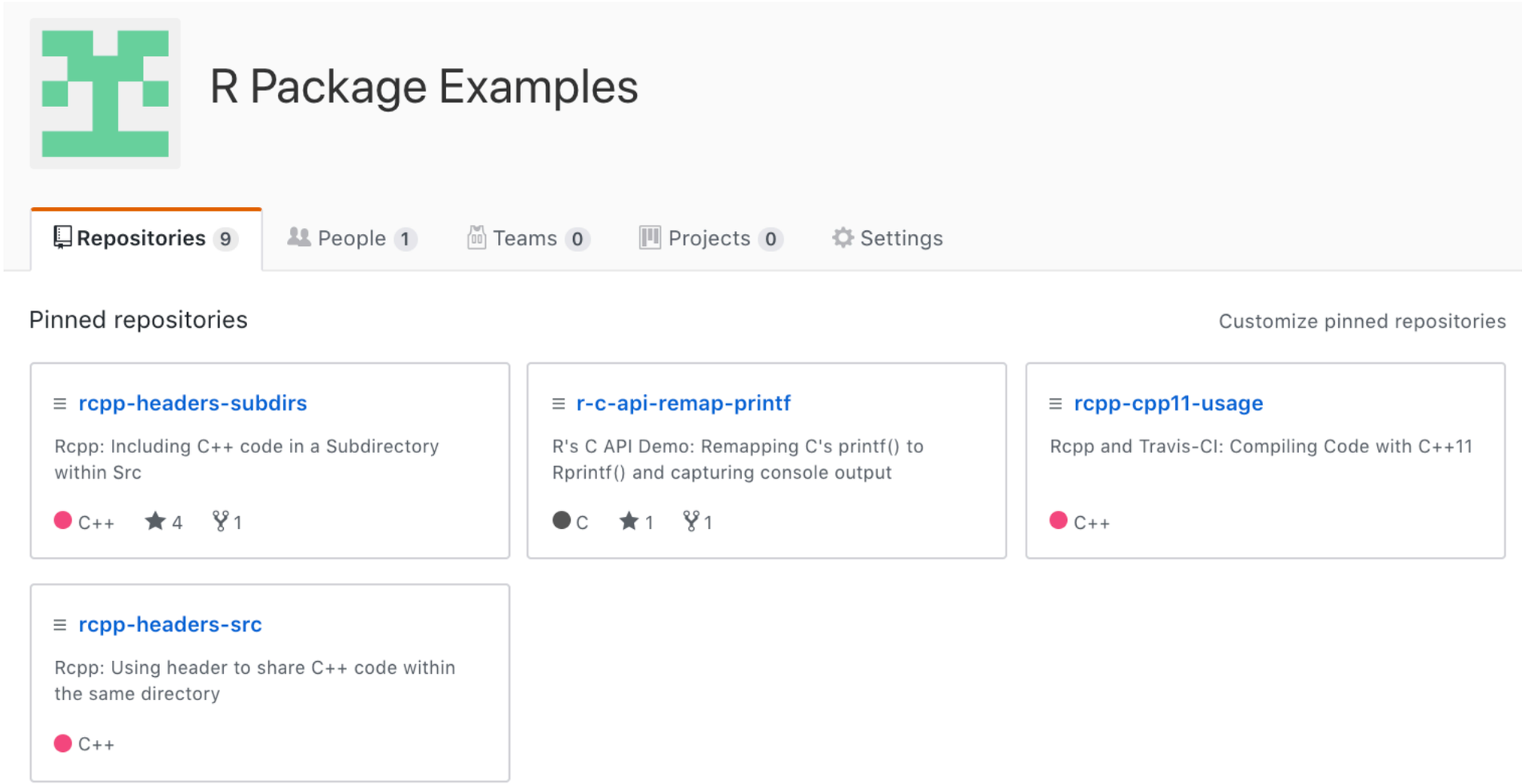
Package Metadata



Function Exports (others can use your *R* code)
Package Imports (using others *R* code)

Example R Packages

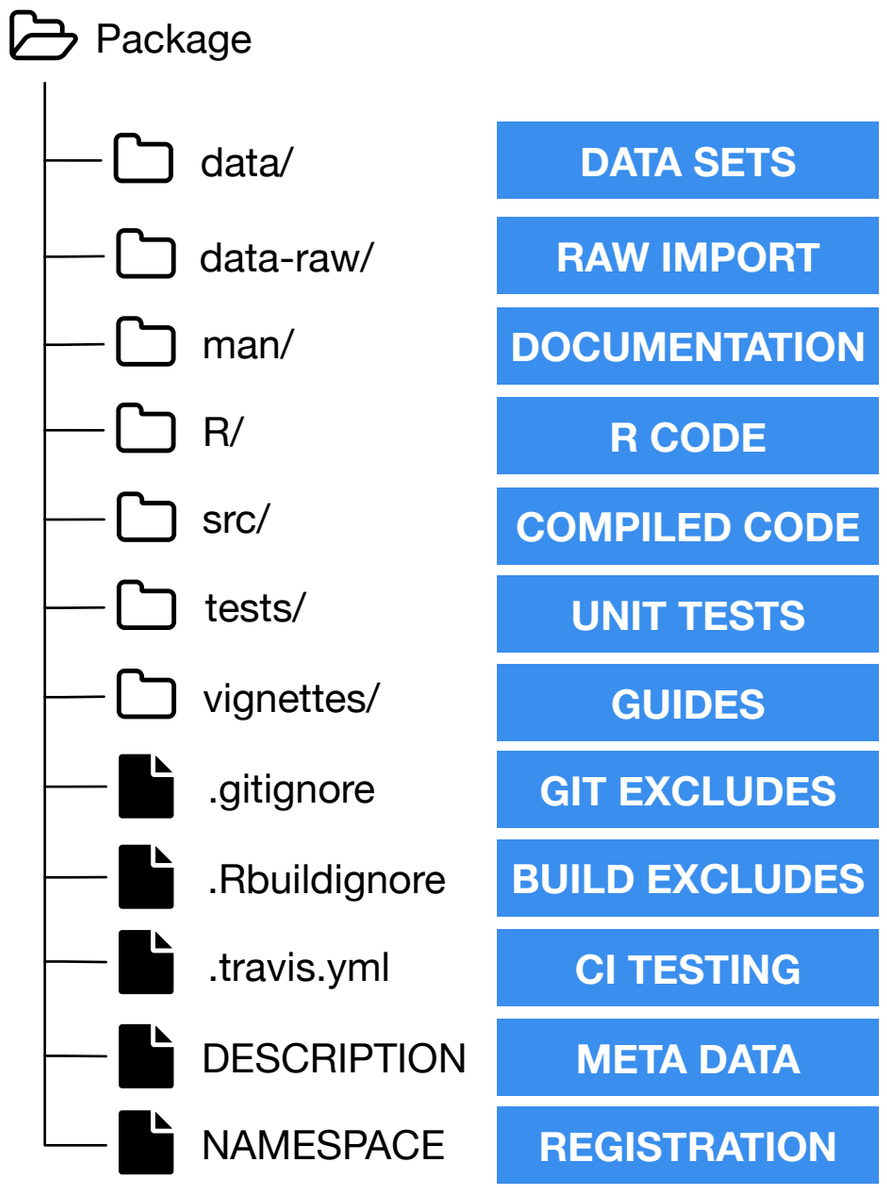
… R & D packages with narrow scope …



https://github.com/r-pkg-examples

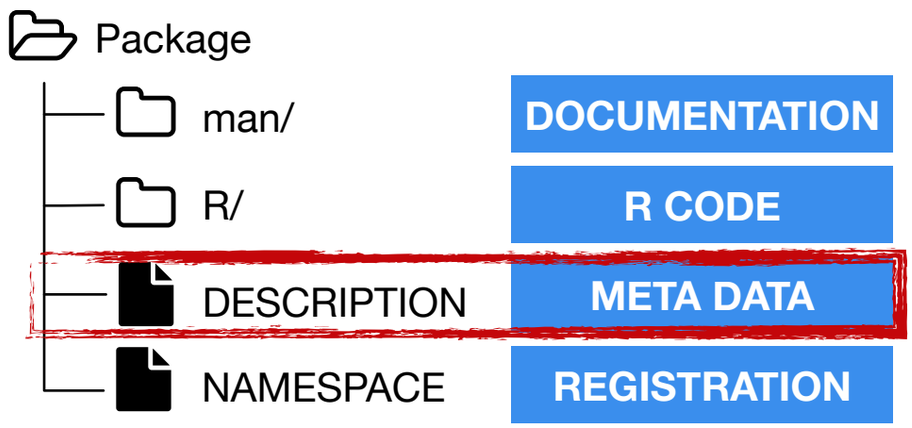
All possible options

… there are many more things to add …



Package Metadata

… editing **DESCRIPTION** …



Four spaces
if new line
used

Name of Package

```
Package: netid
Title: Sample STAT 385 Package
Version: 0.0.1
Authors@R: c(
    person("First Name", "Last Name",
           email = "netid@illinois.edu",
           role = c("aut", "cre", "cph")),
    person("James", "Balamuta",
           email = "balamut2@illinois.edu",
           role = c("aut", "cph")))
Description: Contained within are our first
    attempts at learning how to make a
    package in R.
Depends: R (>= 3.5.0)
License: GPL (>= 2)
Encoding: UTF-8
LazyData: true
```

Licensing

… choosing the right license …

James Balamuta is not a lawyer.*

---

* His advice is not a subsitute for consulting with a lawyer when licensing code. Furthermore, using his advice in relation to license is done at your own risk. He cannot be held liable for any code licensing issues that may arise.

Why License?

… provides clarity as to how others can use your code …

See Section 1.1.2: Licensing within Writing R Extensions

View differences between licenses at:

choosealicense.com
tldrlegal.com

If you want to make money or do commercial endeavors, make sure to **contact a lawyer** before choosing a license!

Specifying Contributions

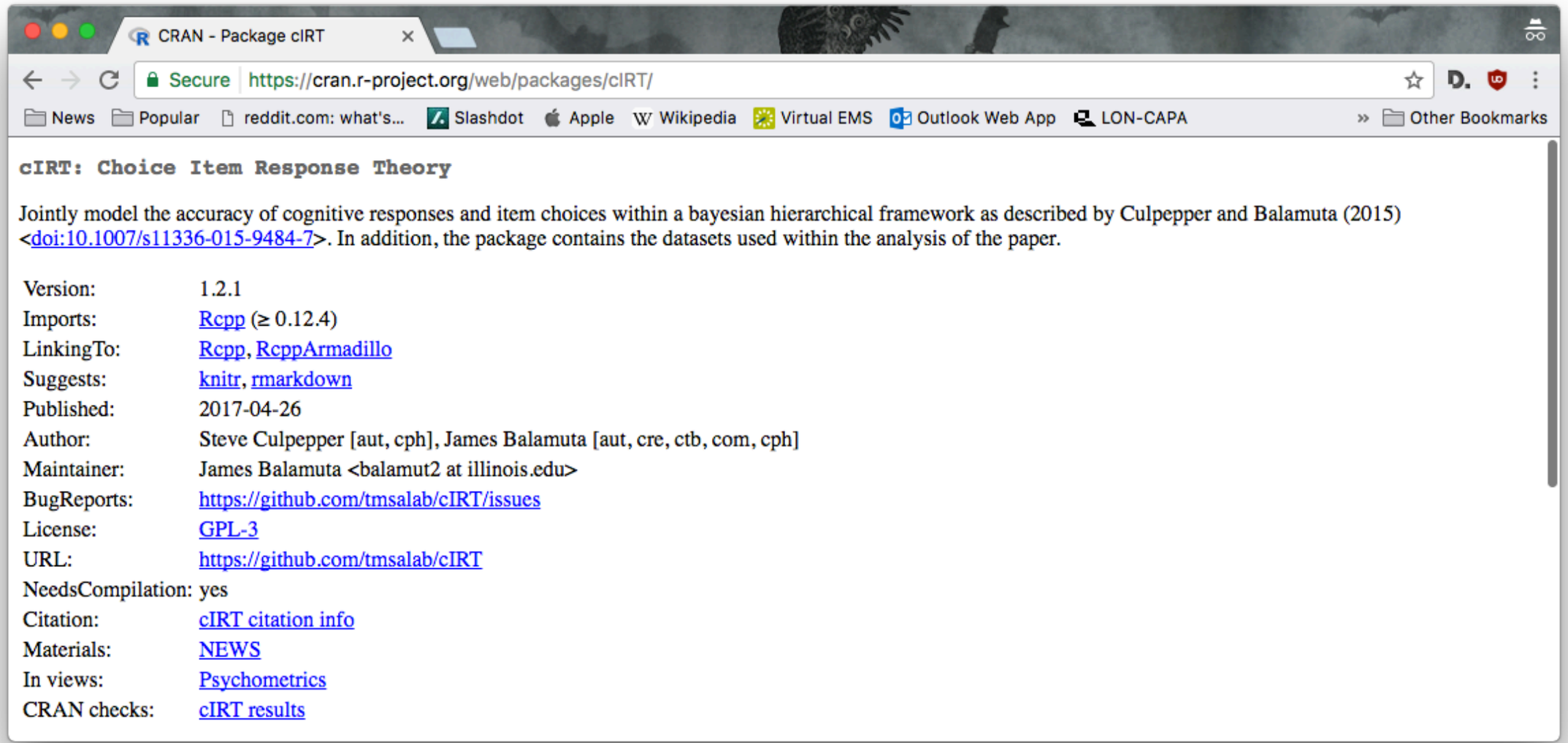
… MARC Code List for Relators …

Role	Position	Description
aut	Author	substantial contributions to the package
ctb	Contributor	authors who have made smaller contributions
cre	Creator	the package maintainer
cph	Copyright Holder	intellectual property rights holders

For more details, see **?person**

DESCRIPTION on CRAN

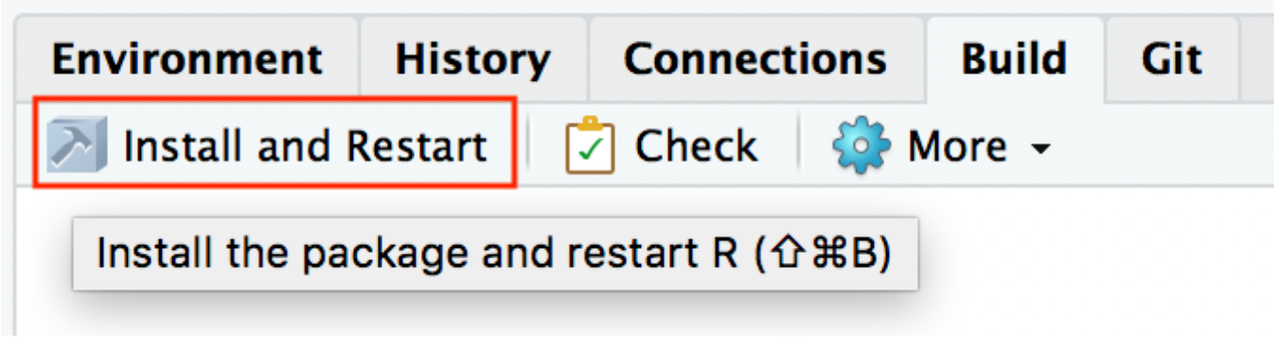
… how metadata is viewed …



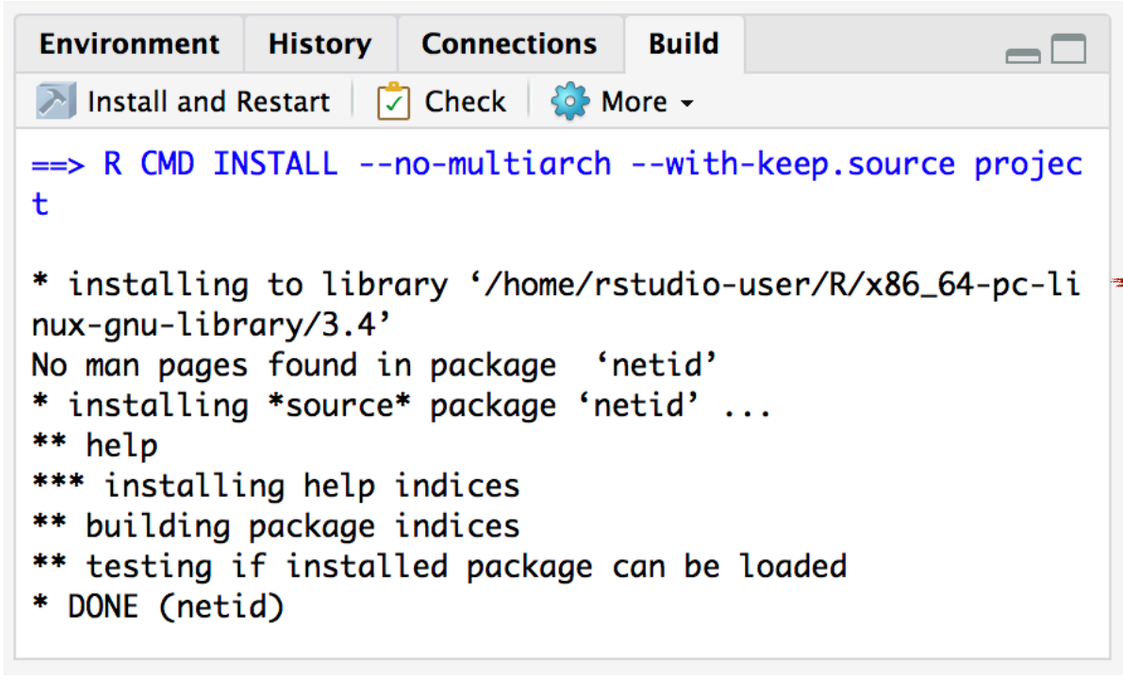
cIRT CRAN page | DESCRIPTION file source on GitHub

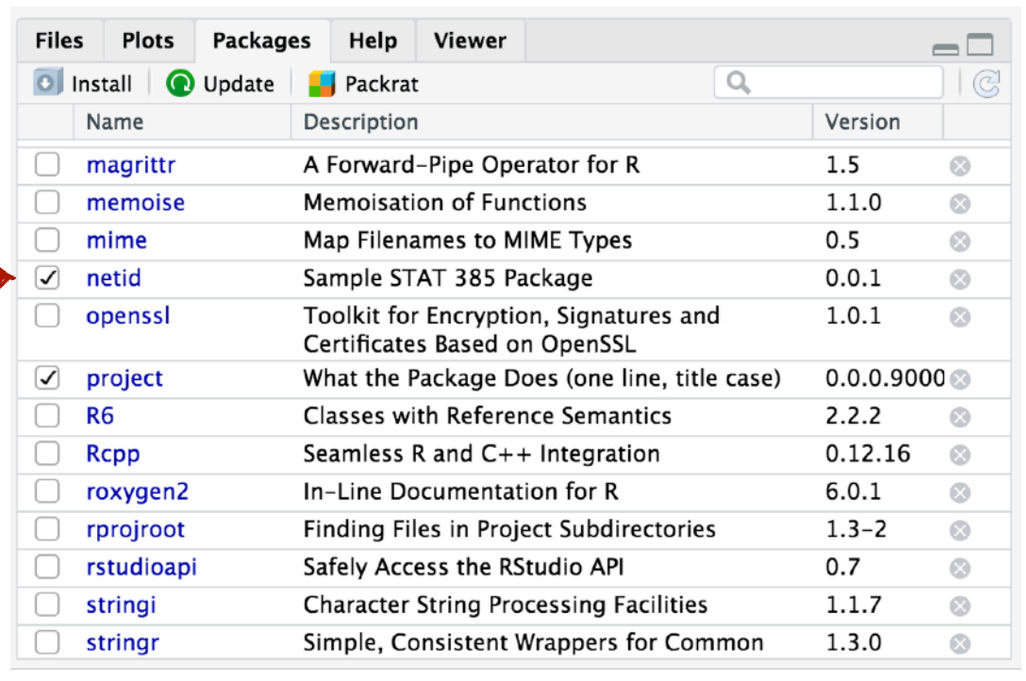
Build Package

… creates and loads the *R* package …



[CNTRL / CMD + SHIFT + B]

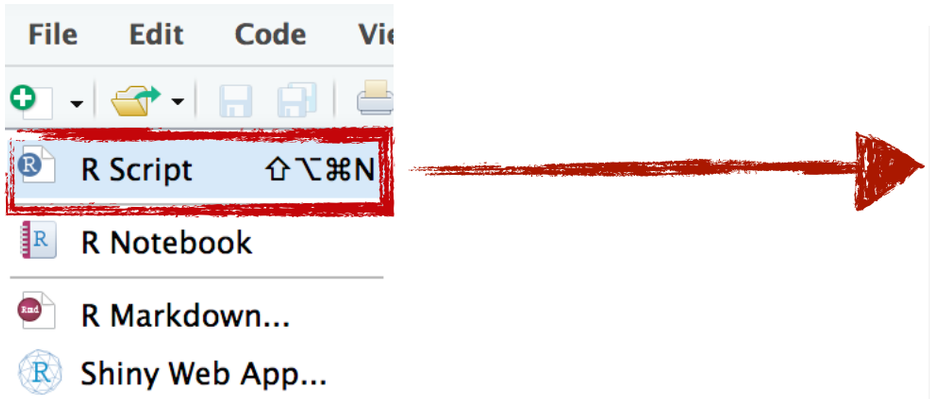


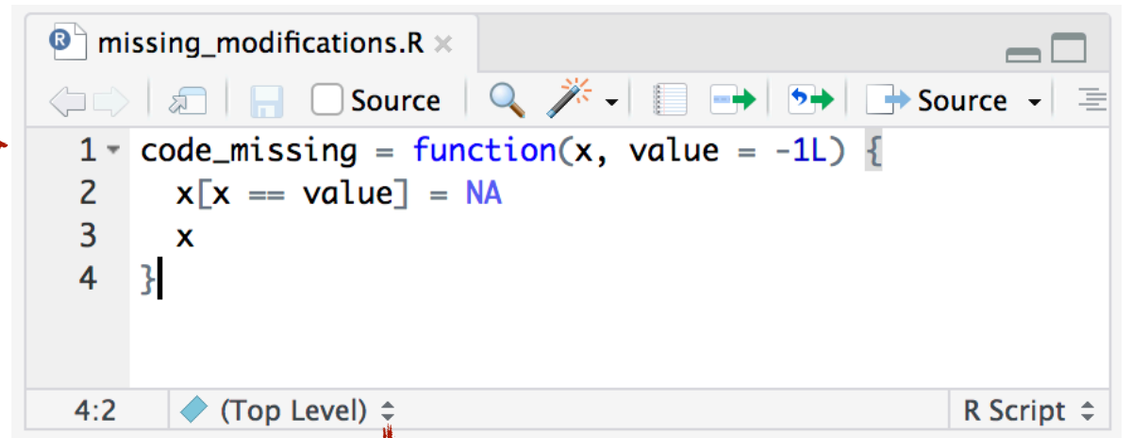


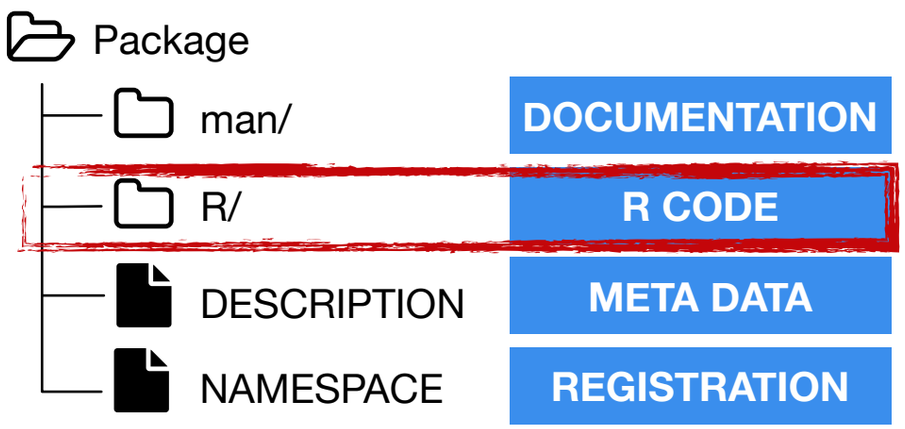
Adding an *R* Function

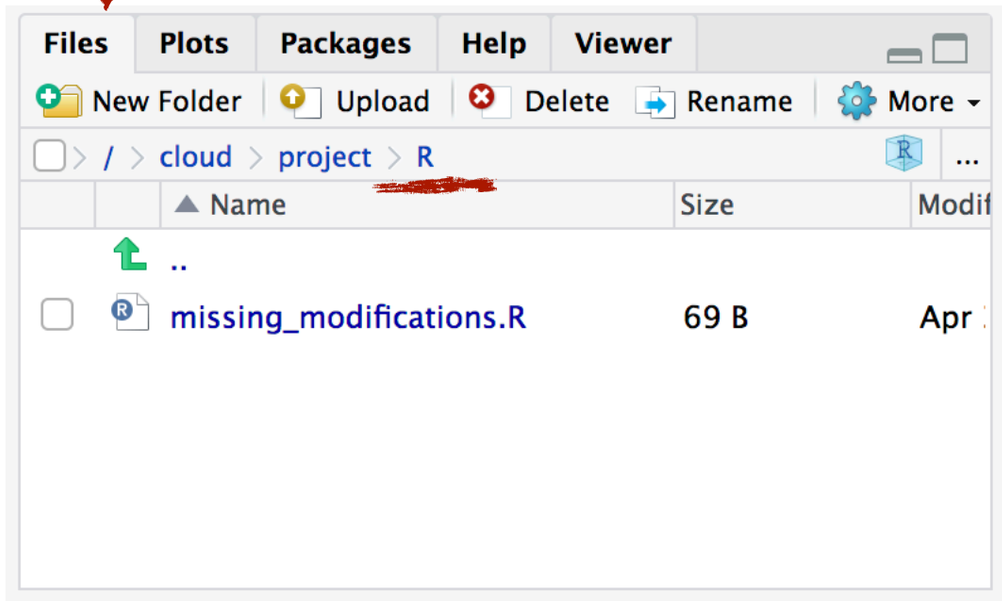
… saving a function to the **R/** directory …

R/missing_modifications.R



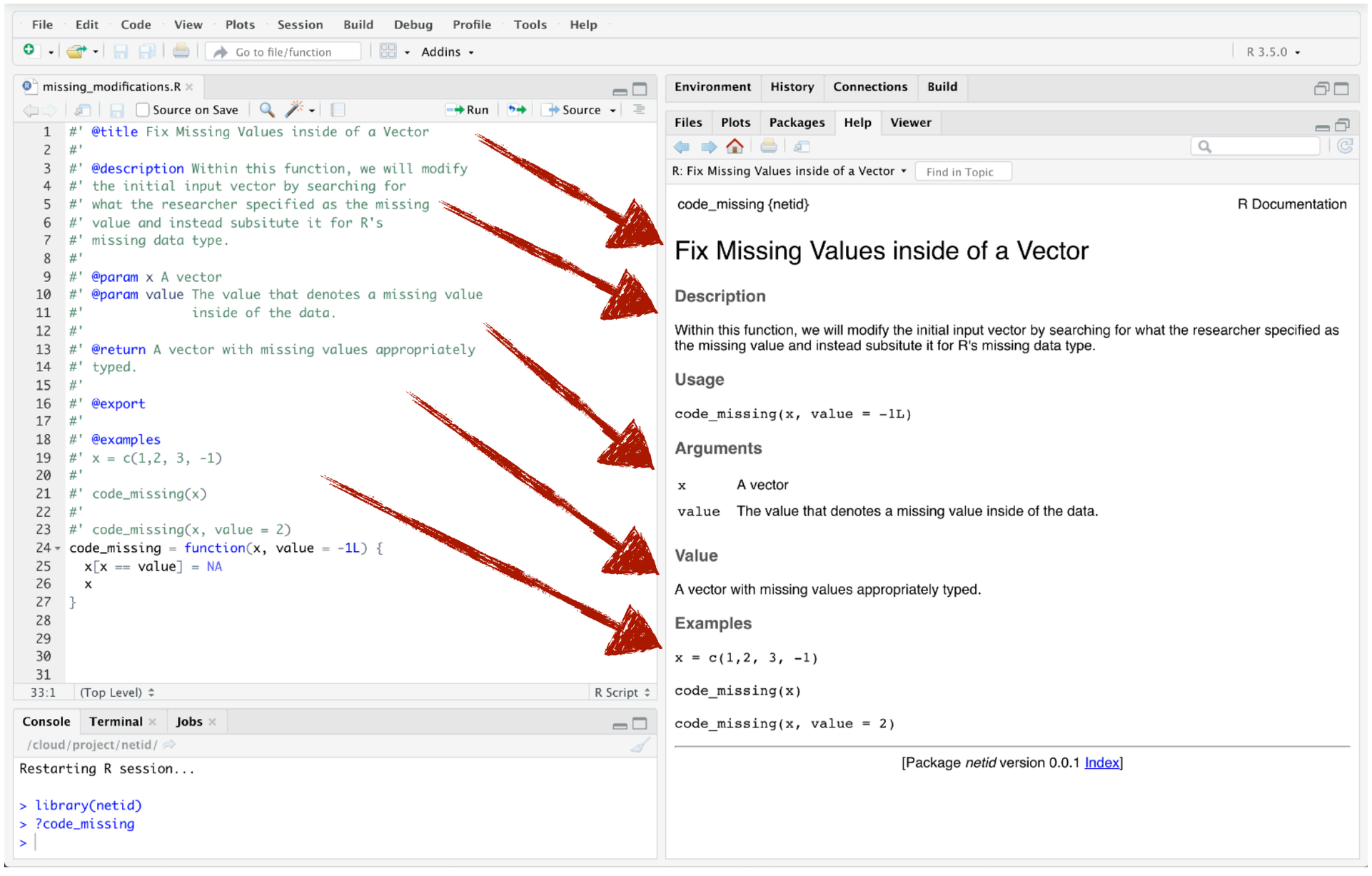






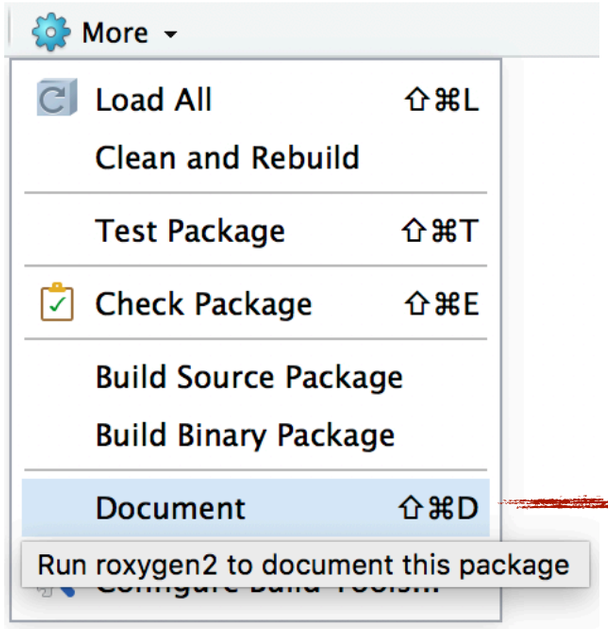
Adding Documentation

… roxygen2's inline documentation feature …

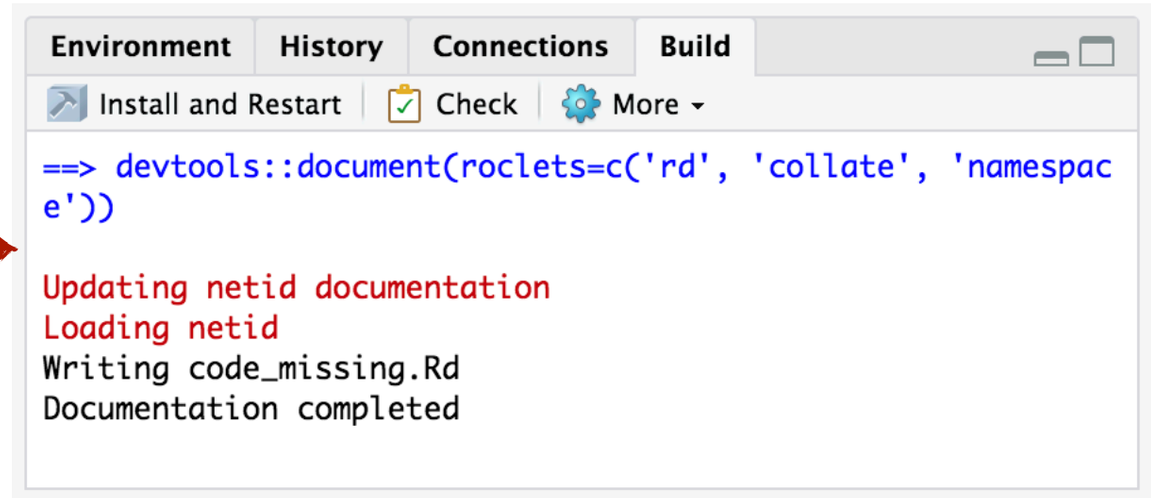


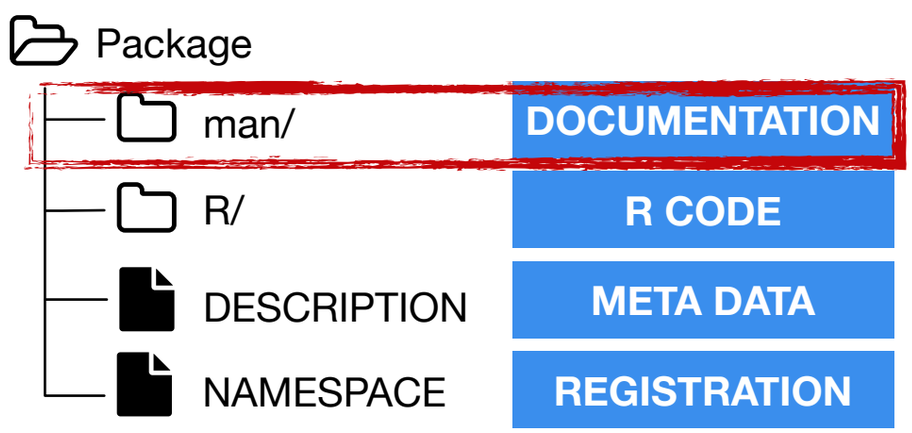
Building Documentation

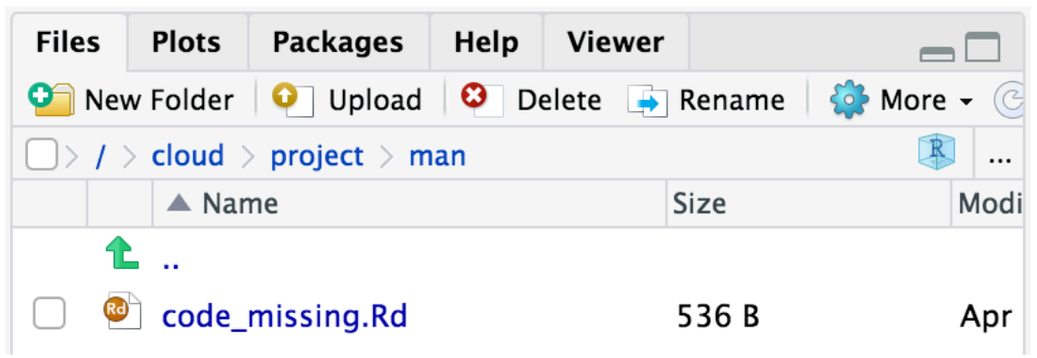
… translating roxygen2 comments to Rd files …



[CNTRL / CMD + SHIFT + D]

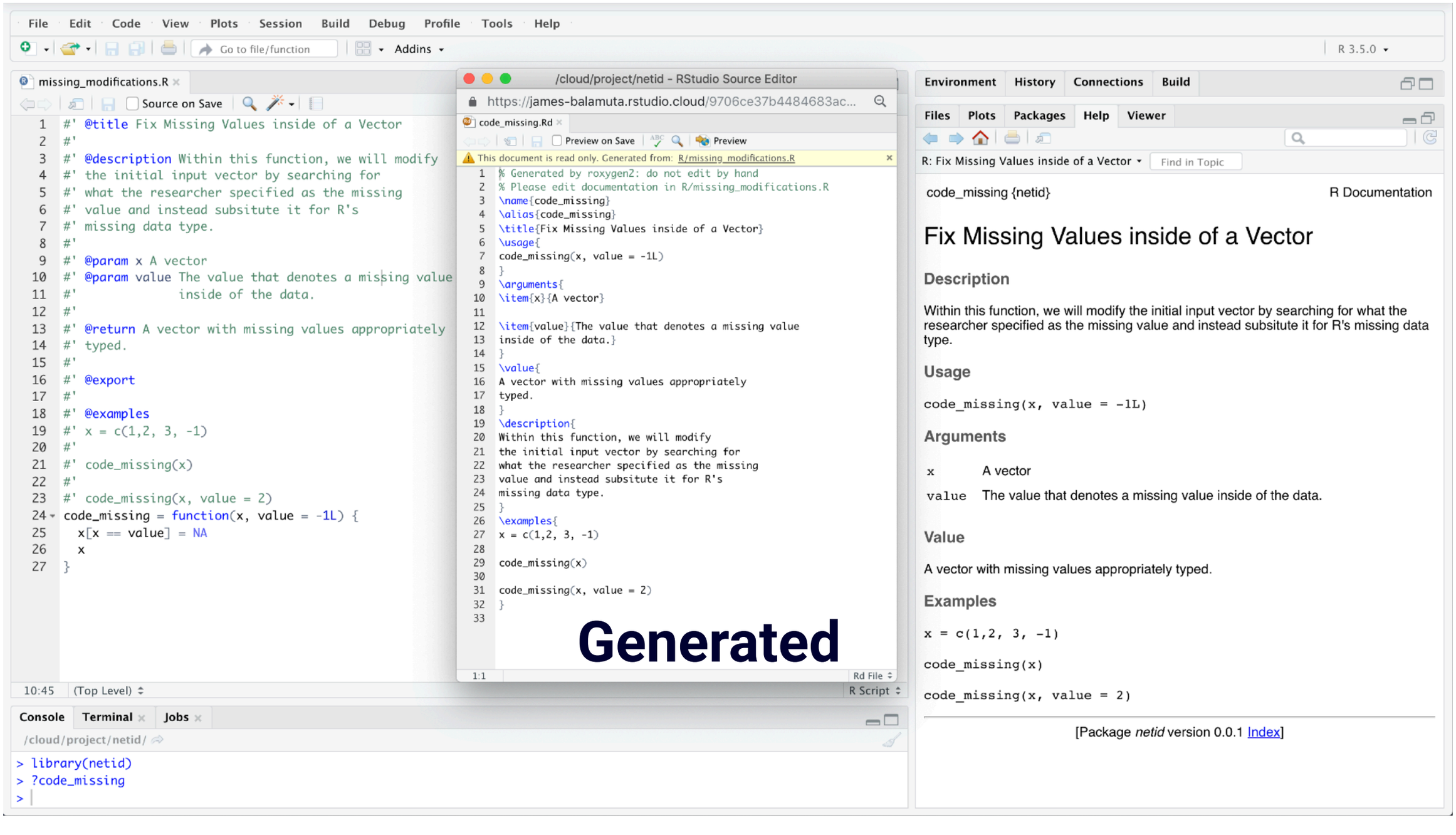






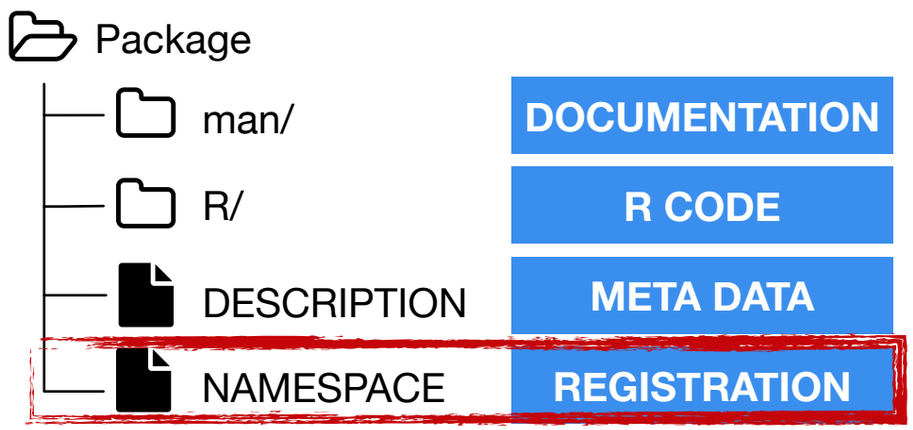
Autogenerates Rd Help Files

… roxygen2 saves time because it automatically generates Rd …



Package Namespace

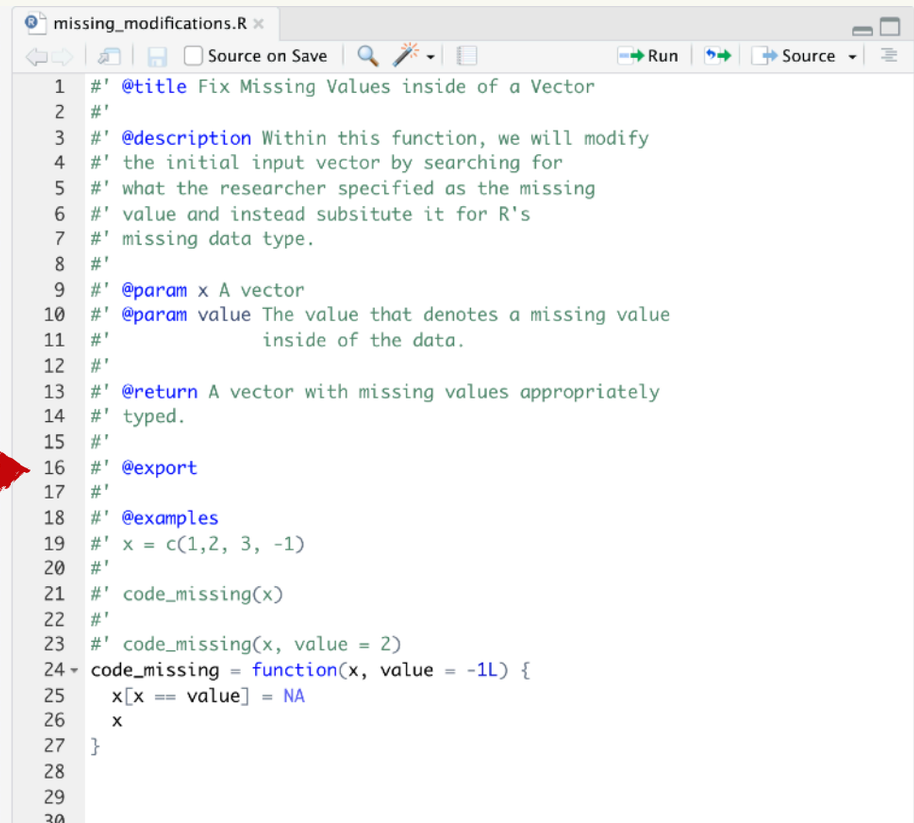
... autogenerated **NAMESPACE** ...



Recognized by **@export** in function comments

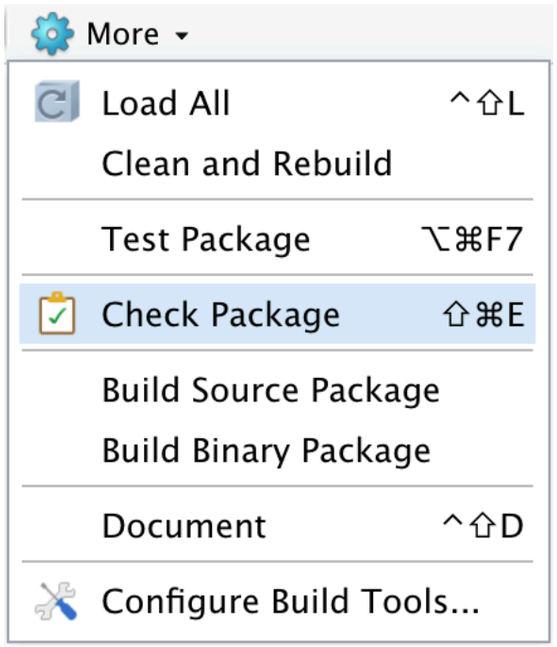
```
# Generated by roxygen2: do not edit by hand

export(code_missing)
```

export()
exposes the function

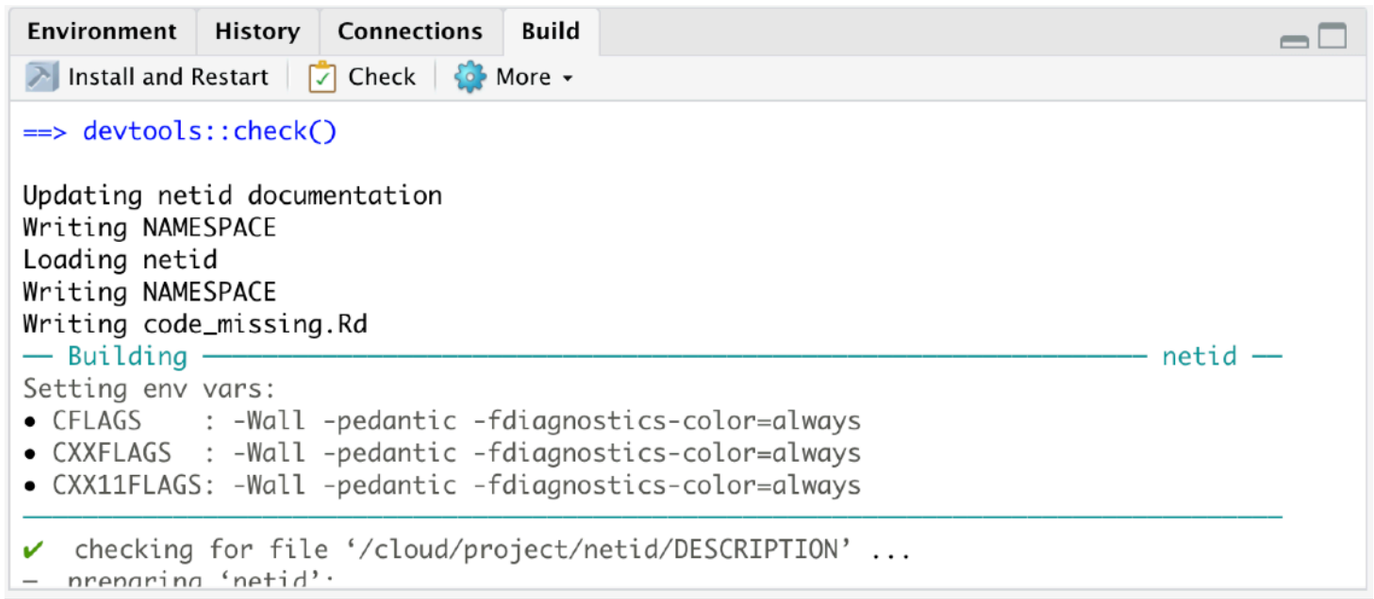


Check Package

… verifies *R* package meets a minimal criterion …

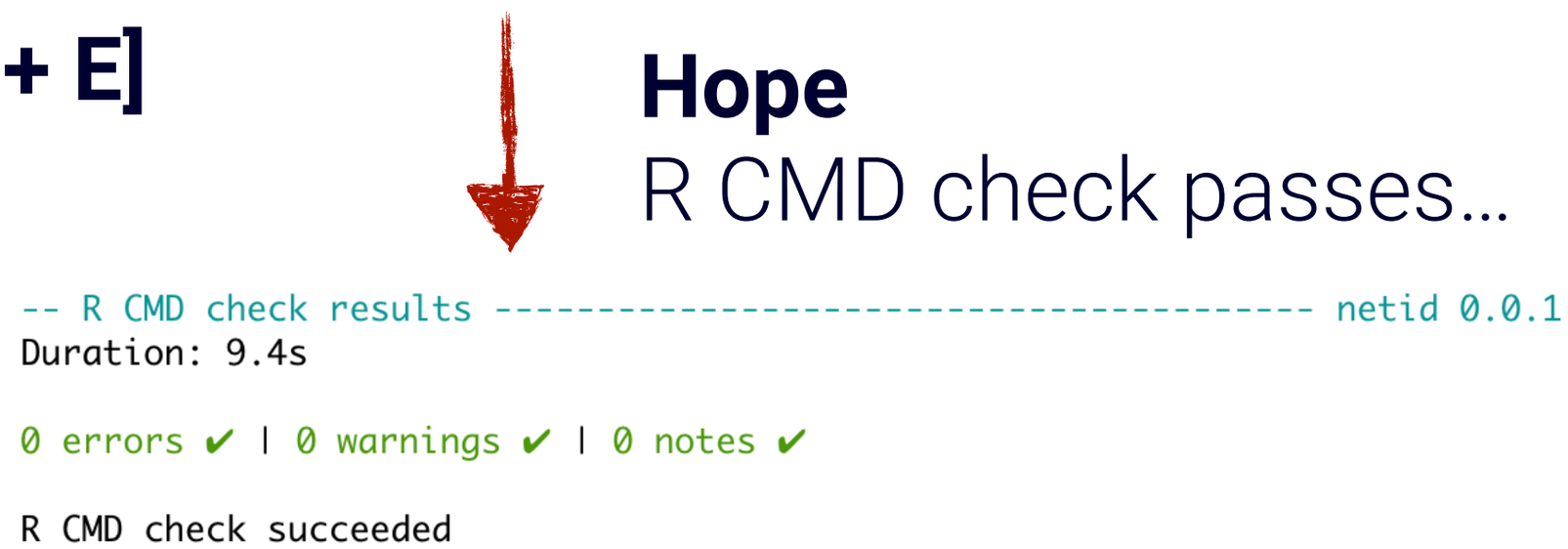


[CNTRL / CMD + SHIFT + E]



Hope

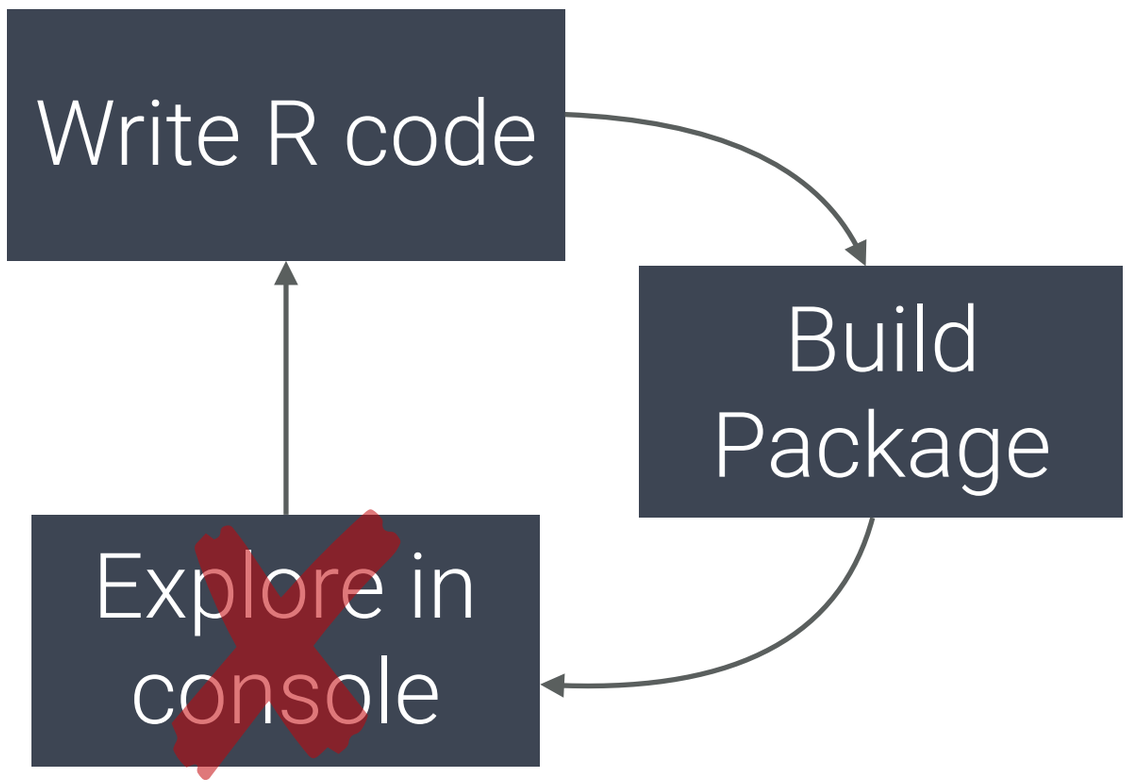
R CMD check passes…

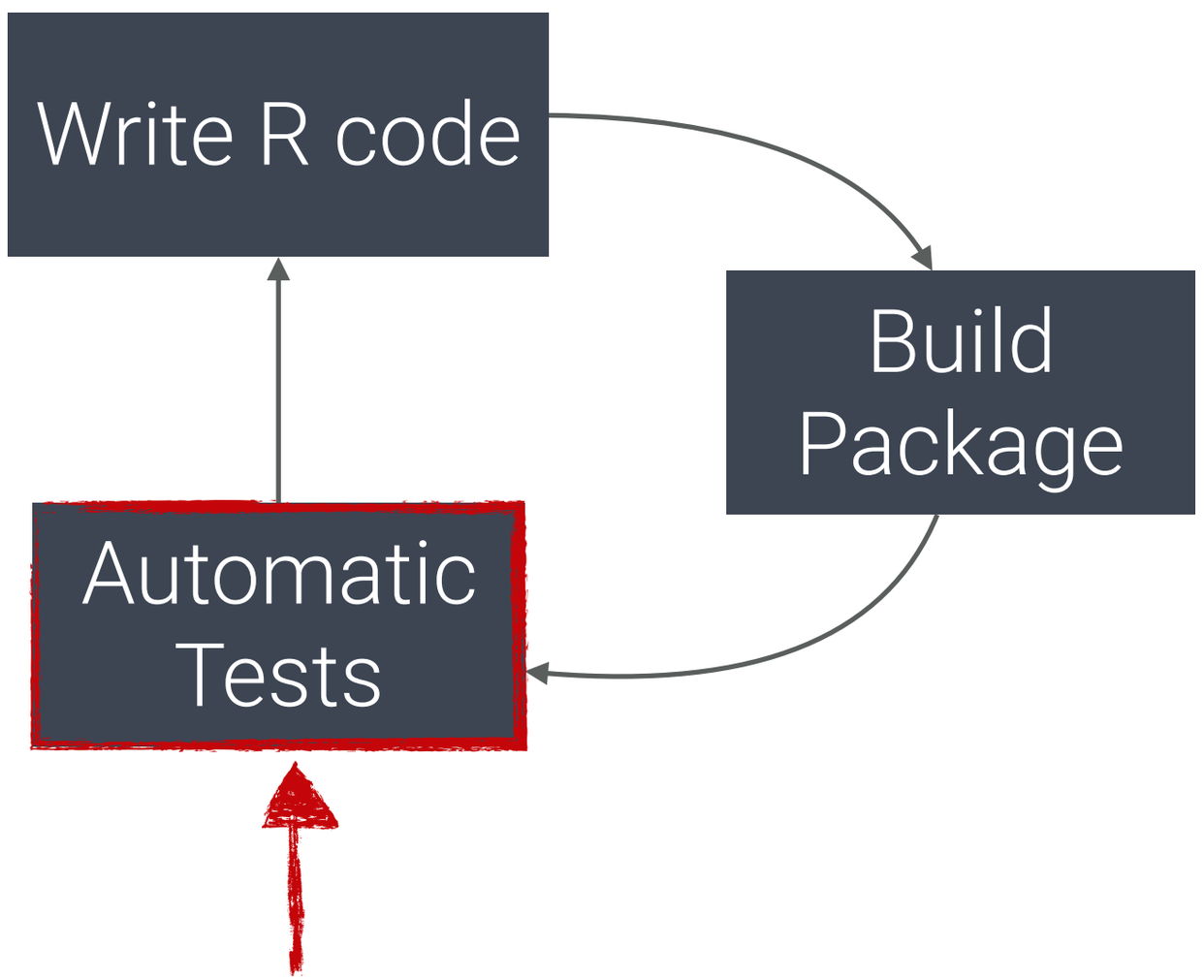


Unit Testing

Testing Functions

… naive vs. unit tests …





Definition:
Unit Testing refers to verifying the smallest parts of a piece of software, referred to as units, meet expectations set forth.

Test File

Test No NAs

Expectation 1
Expectation 2

Test NAs Present

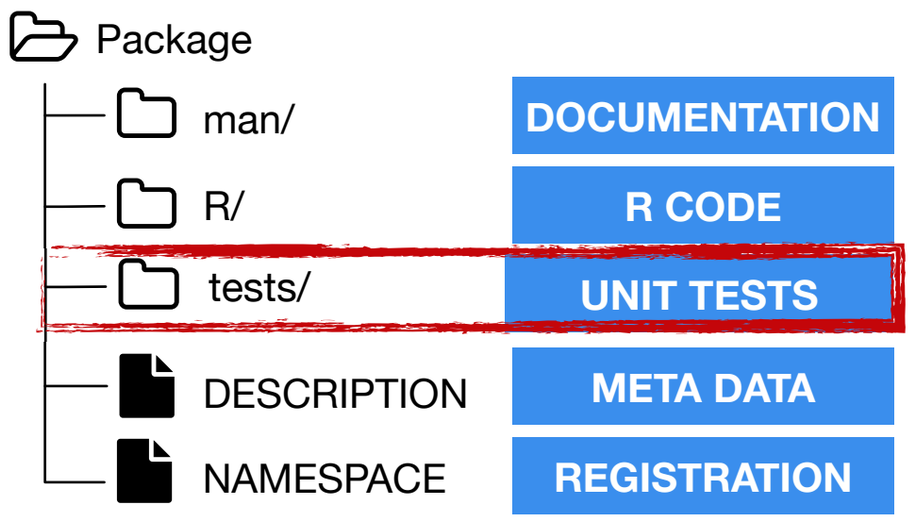
Expectation 1

… framework for writing unit tests …

Function	Description
expect_equal()	Expectation: is the object equal to a
expect_true() expect_false()	Expectation: is the object true/false?
expect_output() expect_error() expect_message() expect_warning()	Expectation: does code produce output/ message/warning/error?
test_package() test_dir()	Run all tests in package or directory

http://testthat.r-lib.org/reference/index.html

Test Example




```
# Enable testthat
usethis::use_testthat()

# Create a test file
usethis::use_test("missing-mods")

## Inside:
## tests/testthat/
## test-missing-mods.R
```

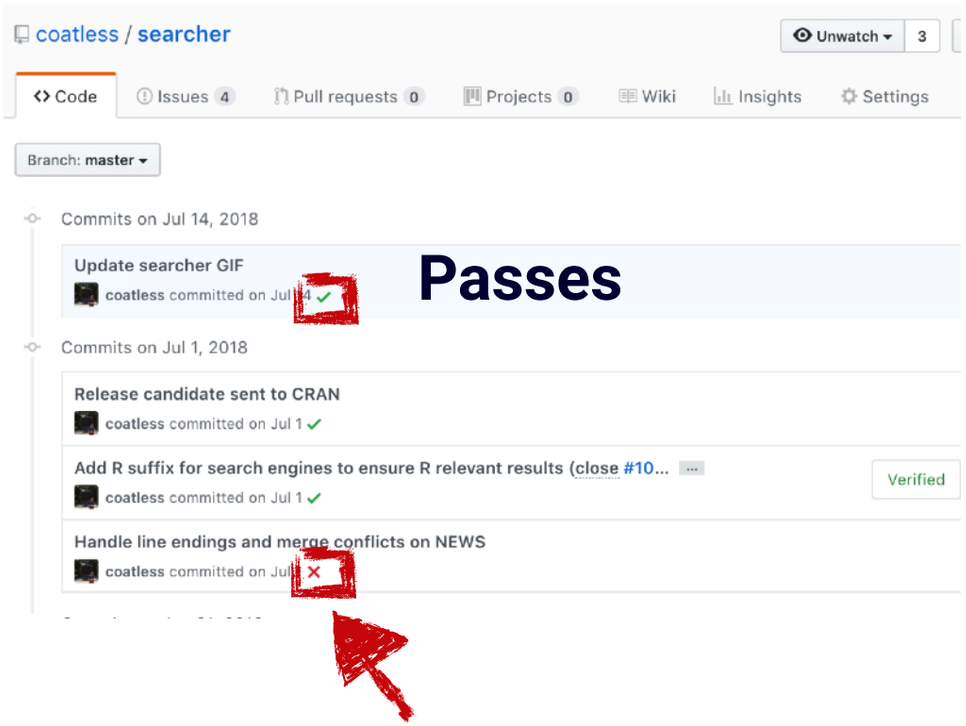
```
context("Missing mods")

test_that("Check coding", {
  x = c(1, 2, 1, -1, 0)
  expect_equal(
     code_missing(x, -1),
     c(1, 2, 1, NA, 0)
  )
})
```

Automatic Test Suites

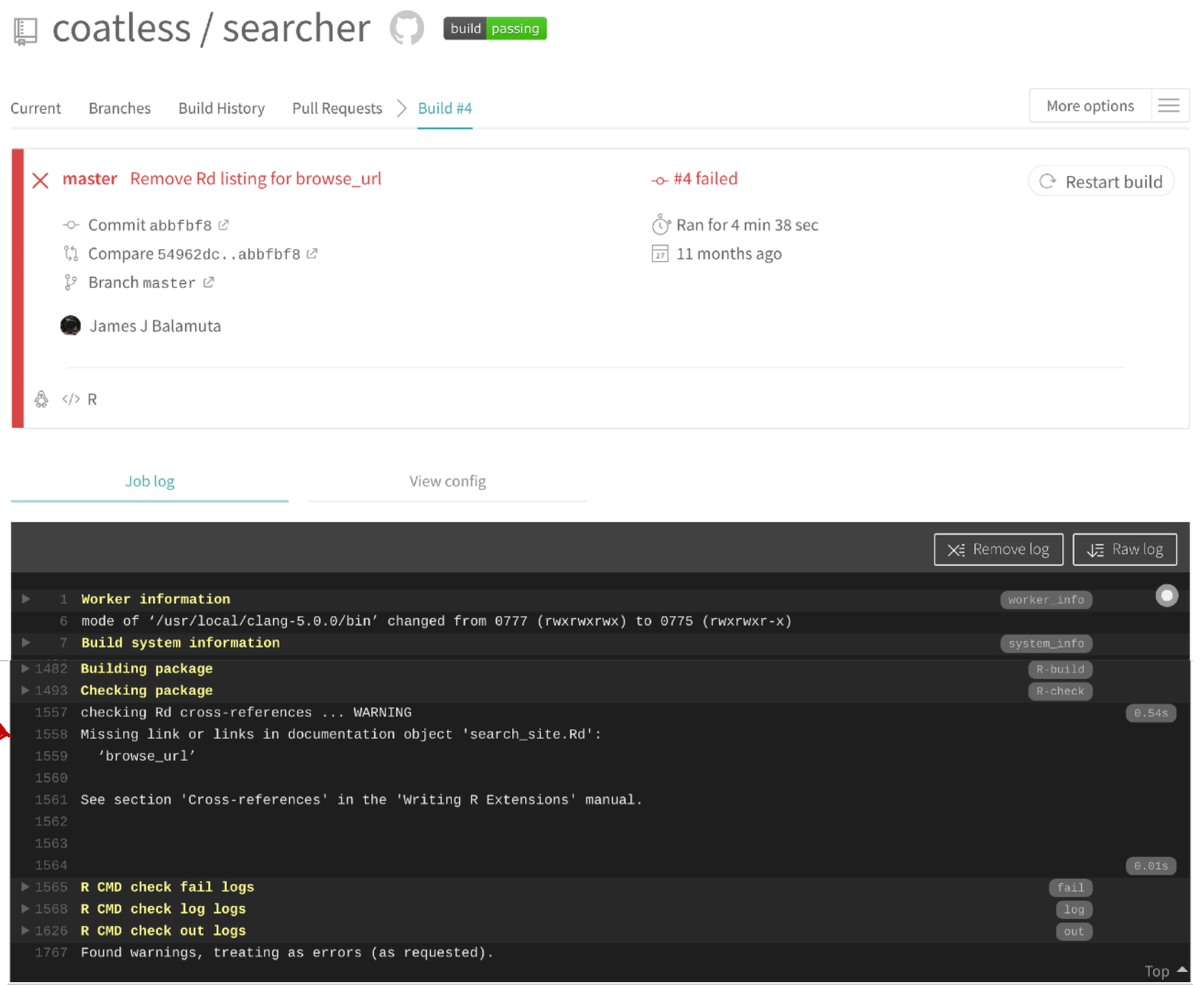
- GitHub: Enables version control and integrates with…
- **Travis-CI**: Continuous integration
  - e.g. run code every for each commit
- **Codecov**: Coverage of Tests
  - e.g. what parts of the code are tested vs. not?

GitHub + Travis-CI

… instant breakage information …

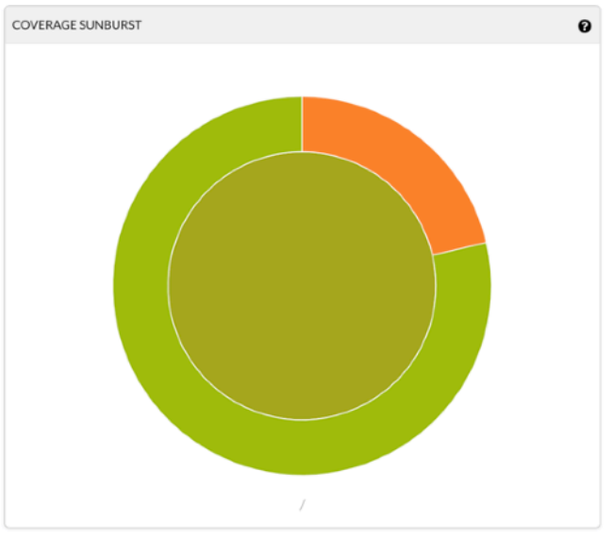


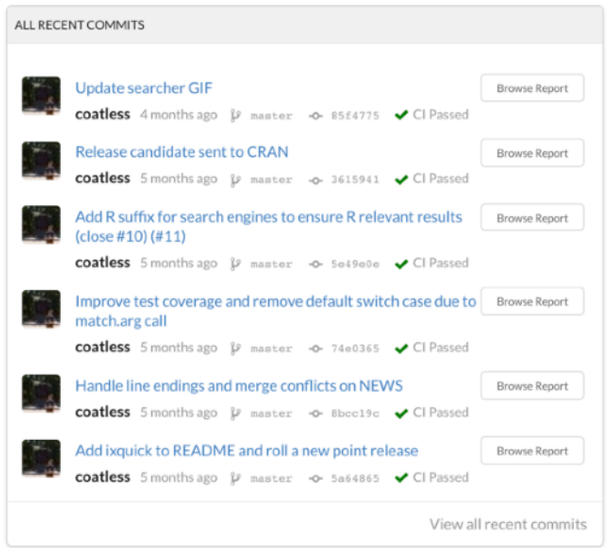
Error'd

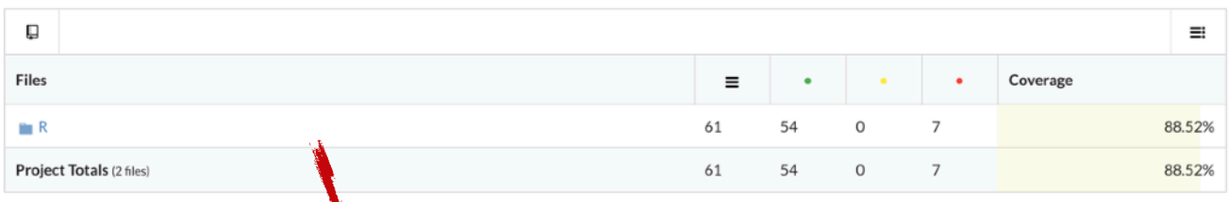


GitHub + Code Coverage

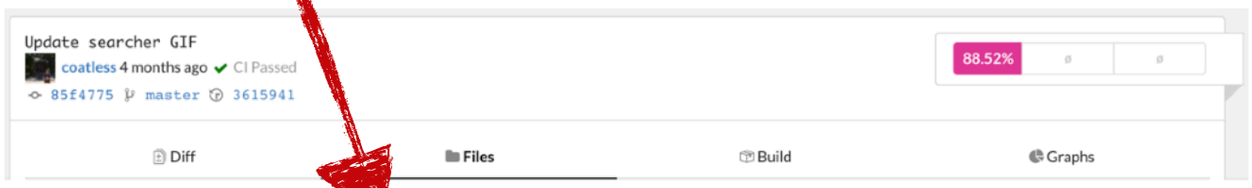
… what parts of code have unit tests …







Files					Coverage
R	61	54	0	7	88.52%
Project Totals (2 files)	61	54	0	7	88.52%



/ R

Files					Coverage
search-functions.R	48	44	0	4	91.66%
utilities.R	13	10	0	3	76.92%
Folder Totals (2 files)	61	54	0	7	88.52%
Project Totals (2 files)	61	54	0	7	88.52%

```
54  search_site = function(query,
55                         site = c(
56                           "google",
57                           "bing",
58                           "stackoverflow",
59                           "so",
60                           "github",
61                           "gh",
62                           "duckduckgo",
63                           "ddg",
64                           "bitbucket",
65                           "bb",
66                           "ixquick"
67                         ),
68                         rlang = TRUE) {
69  1    site = tolower(site)
70  1    site = match.arg(site)
71
72       switch(
73  1      site,
74  1      google         = search_google(query, rlang),
75         stackoverflow  = ,
76         # empty case carried below
77  1      so             = search_stackoverflow(query, rlang),
78         github         = ,
79         # empty case carried below
80         gh             = search_github(query, rlang),
81         bitbucket      = ,
82         # empty case carried below
83         bb             = search_bitbucket(query, rlang),
84         bing           = search_bing(query, rlang),
85         duckduckgo     = ,
86         # empty case carried below
87         ddg            = search_duckduckgo(query, rlang),
88  1      ixquick        = search_ixquick(query, rlang)
89       )
90  }
```

Red: No test covers code

Green: Unit test evaluates code

Acknowledgements

- Hadley Wickham's book aptly title R Packages
- Writing Extensions for R

Recap

- **Motivation**
  - R packages lower the bare to sharing code
  - Provide means to easily re-use functions in other analyses
- **Creating an R Package**
  - RStudio has built in package templates
  - Possible to use **devtools::create_package()** and **package.skeleton()**
- **Structure of R Packages**
  - Packages require **NAMESPACE** + **DESCRIPTION** files **R/** + **man/** folders
- **Unit Testing**
  - Standardize testing and incorporate it into an automatic framework