Building R Packages

What are R Packages?

R Packages

Windows and RTools

Starting Up

devtools

```
install.packages("devtools")
```

-
-
-
-

Checking Package Names

available::available

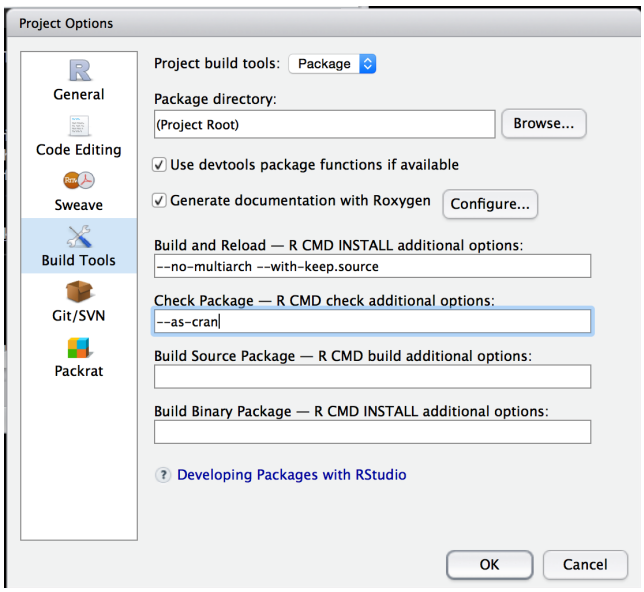
```
available::available("ggplot")
```

```
## Warning: 'BiocInstaller' and 'biocLite()' are deprecated, use the 'BiocManager'
##   CRAN package instead.

## ── ggplot ──────────────────────────────────────────────────────────────
## Name valid: ✔
## Available on CRAN: ✖
## Available on Bioconductor: ✔
## Available on GitHub:  ✔
## Bad Words: ✔
## Abbreviations: http://www.abbreviations.com/ggplot
## Wikipedia: https://en.wikipedia.org/wiki/ggplot
## Wiktionary: https://en.wiktionary.org/wiki/ggplot
## Urban Dictionary:
##   Not found.
## Sentiment:???
```

Setting Up

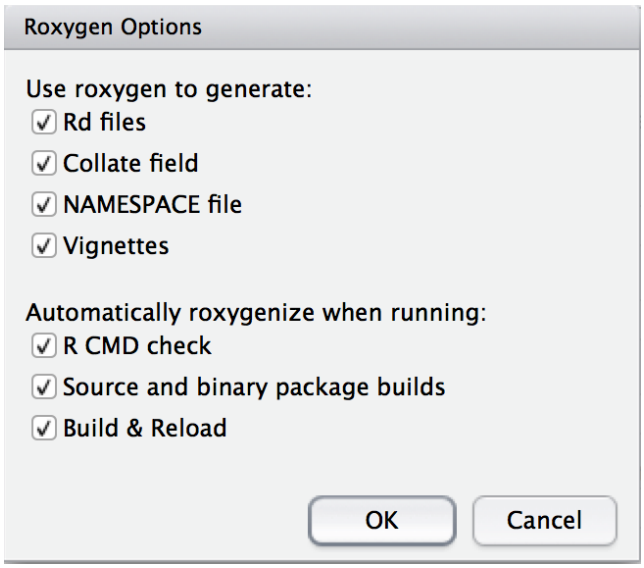
```
--as-cran
```



Setting Up

`roxygen2`

```
install.packages("roxygen2")
```



Modifying the Skeleton

```
      R/hello.R

      man/hello.Rd

          Build → Configure Build Tools → Generate Documentation
with Roxygen                              Vignettes    Build and
Reload

             --as-cran                          Build → Configure
Build Tools

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

DESCRIPTION file

.

- function names

DESCRIPTION file

-
-
-
-
-

Authors

DESCRIPTION

```
Authors@R: person(given = "John",
           family = "Muschelli",
           role = "cre",
           email = "muschellij2@gmail.com")
```

Authors@R desc

DESCRIPTION file: additional fields

- library require
-
  - base stats methods
-
  - ALL
-
  - examples vignettes

```
usethis::use_package("tidyr", type = "Imports")
usethis::use_package("dplyr", type = "Suggests")
```

Description

```
Description

            <http...>
            " "
```

Roxygen2

```
top = function(x, n) {
  xx = x[1:n, 1:n]
  hist(xx)
  print(xx)
}
```

top.R R/

Roxygen2

```
top = function(x, n) {
```

Roxygen Skeleton:

```
#' Title
#'
#' @param x
#' @param n
#'
#' @return
#' @export
#'
#' @examples
```

Roxygen Skeleton:

`@title` `@description`

```
#' @title
#' @description
#'
#' @param x
#' @param n
#'
#' @return
#' @export
#'
#' @examples
```

Roxygen Skeleton:

- `@param`
- `@return`
- `@export`
  -
- `@examples`
  `\dontrun{}`

Roxygen Skeleton:

```
#' @title Print the top of a matrix
#' @description \code{top} is a small function to not just present the first rows
#' of a matrix, but also the first number of columns
#'
#' @param x a \code{matrix}
#' @param n Number of rows and columns to display of the matrix
#'
#' @return A \code{NULL}
#' @export
#'
#' @examples
#' mat = matrix(rnorm(100), nrow = 10)
#' top(mat, n = 4)
#' \dontrun{
#'    top(mat, n = 10)
#' }
```

Functions: a little style

```
                    NULL        is.null()
                    #' @rdname
#' @inheritParams
verbose
message    cat                          suppressMessages

 ...

  do.call(FUNCTION, args = list_of_arguments)
```

NAMESPACE

NAMESPACE

- @export
  -
- @import package @import PACKAGENAME
  - ALL
  - whole package
  - library(PACKAGENAME)

NAMESPACE

- @importFrom function @import PACKAGENAME func1 func2
  -
  - pkgA A pkgB A B @import pkgA A @import pkgB B A() pkgA
  - base stats quantile graphics hist

  @importFrom graphics hist

Build and Reload

-
  - -
  - -
  - -
-

Using Data

data/ must .RData rda use_data

```
usethis::use_data(DATAOBJECT, compress = "xz")
```

DATAOBJECT.rda data

Making Data

`data-raw`

`use_data`

```
usethis::usethis::use_data_raw()
```

Documenting Data

DATAOBJECT

```
#' @title Some object to document
#'
#' @description A list containing things
#'
#' @format A list with 7 elements, which are:
#' \describe{
#' \item{x}{first thing}
#' \item{y}{second thing}
#' }
"DATAOBJECT"
```

Different kinds of data

inst/
blah.csv inst/

inst/extdata

find.package

```
find.package("readr")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.5/Resources/library/readr"
```

system.file

```
system.file("extdata", package = "readr")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.5/Resources/library/readr/extdata"
```

Different kinds of data

`file.path`

```
system.file("extdata", "challenge.csv", package = "readr")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.5/Resources/library/readr/extdata/challenge.
```

`mustWork`

```
system.file("extdata", "asdfsdf.csv", package = "readr", mustWork = TRUE)
```

```
## Error in system.file("extdata", "asdfsdf.csv", package = "readr", mustWork = TRUE): no file
```

Using the file system

```
file.path  paste
    file.exists     any    all
file.remove
                                                    tempfile(fileext
= ".csv")
dir.create            unlink
     tempdir
                    tdir = tempfile(); dir.create(tdir);
on.exit({ unlink(tdir) })
file.copy    file.rename
download.file   httr::GET
```

Vignettes

- _____

  how

- THIS IS EXACTLY WHAT A REPRODUCIBLE PAPER IS!

```
usethis::use_vignette("my-vignette")
```

-

Unit tests

```
    testthat                                                tests/testthat
            test-DESCRIPTOR.R          testthat
```

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

```
usethis::use_test("name of test")
```

General Rule: Any package issue turns into a test.

Unit tests

testthat tests/testthat test-DESCRIPTOR.R testthat

```
testthat::context("OVERALL DESCRIPTION OF TESTS IN THIS FILE")
testthat::test_that("Description of this test", { MYCODE })
testthat::expect_equal(OUTPUT, 1234.34535)
testthat::expect_identcal(OUTPUT1, OUTPUT2)
testthat::expect_true(SOME_OUTPUT)
testthat::expect_silent({ no_warning_error_code })
testthat::expect_message({ some_warn }, "a[test]regexp")
```

Continuous integration (Thoughtworks.com)

Building CI and README

```
usethis::use_git()
usethis::use_github() # must have GITHUB_PAT set up
usethis::use_github(protocol = "https") # must have GITHUB_PAT set up
usethis::use_readme_rmd()
usethis::use_appveyor()
usethis::use_travis()
```

Configuring Travis

.travis.yml

```
os:
  - linux
  - osx

warnings_are_errors: true
after_success:
  - Rscript -e 'covr::codecov(type = "all")'
```

Configuring Appveyor

appveyor.yml

```
environment:
  global:
    WARNINGS_ARE_ERRORS: 1
```

Adding to the README.Rmd

GITHUB_USERNAME/REPO

```
[![Travis-CI Build Status](https://travis-ci.com/GITHUB_USERNAME/REPO.svg?branch=master)](https
[![AppVeyor Build Status](https://ci.appveyor.com/api/projects/status/github/GITHUB_USERNAME/RE
```

```
usethis::use_badge("Travis-CI Build Status",
src = "https://travis-ci.com/GITHUB_USERNAME/REPO.svg?branch=master",
href = "https://travis-ci.com/GITHUB_USERNAME/REPO")
```

README.Rmd

S3, S4, Reference Classes

- `class(x) = "myS3Class"`
  -
  - _____
- `new`
  -
  - _____
-
  - `class$method()`
  - _____

S3, S4 Methods

- `bar <- function(y) UseMethod("bar", y)`
  - `bar.myS3Class bar(x)`
- `setGeneric("myGeneric", function(x) standardGeneric("myGeneric"))`

```
setMethod("myGeneric", signature(x = "myS3Class"), function(x, y) {
 x@slot + y
})
```

Compiled Code: C and C++

- src/
- cleanup configure
- configure make
- Makevars Makefile
- configure.win Makevars.win
- .Call

Rcpp

Questions?