MANGO SOLUTIONS

How to move like Hadley

Adnan Fiaz
Data Scientist
afiaz@mango-solutions.com
@tapundemek

Who are Mango?

Founded in 2002

Offices in Chippenham and London

Specialise in the provision of complex analysis solutions, consulting, world class training, and application development

Organisers of a range of data science focused events and organisations

Who am I?

- Data Scientist
- Previously @ KLM
- Like to…
  - Make paper helicopters with `library(SixSigma)`
  - Read waitbutwhy.com

Why build a package?

- Fundamental unit shareable code
- Everything (code, data, docs) in one location
- Simplify loading of code
- Facilitate reproducible analysis

Why use devtools?

- Base R facilitates package building
- **devtools** makes it easier
- Integrates with RStudio
- Integrates with other packages (**roxygen2**, **testthat**)

Package structure

Package structure
devtools::create

- Minimum components
  - DESCRIPTION
  - R directory
  - man directory
  - NAMESPACE

- RStudio → New Project → R Package

DESCRIPTION file

- Basic package information
- Package dependencies
- License

R directory

- ALL R code
  - No subdirectories

- Good practices
  - One file per function but don't go crazy
  - Meaningful filenames
  - Use consistent coding style

Document package
devtools::document

- Most important part of package
- Generated with **roxygen2**
- Each file requires a *roxygen header*
- Also manages NAMESPACE file

R CMD check

devtools::check

- Series of checks demanded by CRAN
- Even without CRAN still good practice
- The check with **devtools**:
  - Generates documentation
  - Checks DESCRIPTION
  - Checks dependencies
  - Errors/Warnings/Notes

Share package
devtools::build

- R CMD check passed?
- Ready to share package → Build package
- Create single file

- Windows: RTools
- Install with install.packages

Interactive development
devtools::load_all

- When developing package code → check → build → install can be tedious
- To check your code changes quickly
- *load_all()* will source all your R code

Tests

devtools::use_testthat

- Tests are another form of documentation
  - Ensure requirements are satisfied
  - Make your code robust to changes
  - Track bugs

- Life made easy with **testthat**

Tests

devtools::test

- Write tests
  - Context
  - Testcases
  - Expectations

- Then run *test()*

orange
NO PULP
EXTRA PULP
NEVER FROM CONCENTRATE
PRICE CHOPPED!
2/$5

Extra bits...

Version control

- Track changes
- Collaborate with others
- Git/SVN/BitBucket
- Upload package directory to GitHub

Continuous integration
devtools::use_travis

- Ensure that a package is checked and tested on a regular basis
- Automatically run tests / R CMD check
- Setting up Travis CI with Github

A package website

- **pkgdown** builds package website
- Easier to read/navigate than repo or README
- pkgdown::build_site()
- Setting up GitHub Pages

Extra extra bits...

- Add data
- Add C++
- Add vignettes
- Test coverage
- http://r-pkgs.had.co.nz/
- https://www.mango-solutions.com/data-science/training/courses.html

Adnan Fiaz

afiaz@mango-solutions.com