

# Installing and Maintaining R Packages

Drew Schmidt

Knoxville R Users Group  
October 3, 2014



# About This Presentation

## Downloads

This presentation is available at: <http://github.com/wrathematics/Rpackagetalk> and is licensed under a [Creative Commons Attribution 4.0 International License](#).



# Contents

- 1 Installing R Packages
- 2 Maintaining R Packages
- 3 Writing R Packages
- 4 Wrapup



# Contents

- 1 Installing R Packages
  - Why R Extensions?
  - Installing R Packages



- 1 Installing R Packages
  - Why R Extensions?
  - Installing R Packages

## R Extensions

- R is great, but limited.
- Has a great package extension system.
- R doesn't do what you want? Make it!



## R Extensions

- R is great, but limited.
- Has a great package extension system.
- R doesn't do what you want? Make it!

*"...if I dont know how to fix it, I can hire somebody else to fix it for me."* — Matt Dowle, developer of the `data.table` package.



## Terminology





## Terminology

- We call the extension a *package*.

## Terminology

- We call the extension a *package*.
- *Packages* go into a *library*.



## Terminology

- We call the extension a *package*.
- *Packages* go into a *library*.
- **This is dumb and confusing, but there's nothing we can ever do about it.**



## Example Confusion

## Example Confusion

- I wrote a `library`.

## Example Confusion

- I wrote a **library**.
- I put the library in a **package**.

## Example Confusion

- I wrote a **library**.
- I put the library in a **package**.
- I install the package ...into a library.

## Example Confusion

- I wrote a `library`.
- I put the library in a `package`.
- I install the package ...into a library.
- I load the package with `library()` ???



\*BOOM\*



- 1 Installing R Packages
  - Why R Extensions?
  - Installing R Packages

## Installing R Packages

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



## Installing R Packages

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

```
1 install.packages("devtools", lib="some/place/on/disk")
```

## Installing R Packages

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

```
1 install.packages("devtools", lib="some/place/on/disk")
```

```
1 R CMD INSTALL devtools_1.6.tar.gz -l /some/place/on/disk
```



## Repositories

- This basically assumes you're using CRAN.



## Repositories

- This basically assumes you're using CRAN.
- Lots of exciting development is happening outside of CRAN these days.



## Repositories

- This basically assumes you're using CRAN.
- Lots of exciting development is happening outside of CRAN these days.
- Other binary package repositories: Bioconductor, R-forge (Windows)





## Repositories

- This basically assumes you're using CRAN.
- Lots of exciting development is happening outside of CRAN these days.
- Other binary package repositories: Bioconductor, R-forge (Windows)
- Other source repositories: GitHub, Bitbucket, ...



## GitHub Binaries?



**Karl Broman** @kwbroman · Sep 27

.@hadleywickham @millerdl It would be great to have a service that would compile windows/mac binaries of #rstats pkgs on github.

← Reply ↻ Retweet ★ Favorite ... More



**Hadley Wickham**

@hadleywickham

+  Follow

@kwbroman @millerdl agreed. Plans are in motion



## Installing Packages from Source

To install packages from source, you need some compilers:

- **Windows:** Install [Rtools](#).
- **Mac:** Install Xcode from the app store, possibly some other things from [here](#). If you need OpenMP, god help you.
- **Linux and FreeBSD:** You're good to go.



## Installing R Packages

```
1 library(devtools)
2 install_github("hadley/devtools")
```

### Devtools Package Install Functions

<code>install_bitbucket</code>	<code>install_github</code>	<code>install_svn</code>
<code>install_deps</code>	<code>install_gitorious</code>	<code>install_url</code>
<code>install_git</code>	<code>install_local</code>	<code>install_version</code>

# Contents

- 2 Maintaining R Packages
  - Maintaining R Packages



## Maintaining R Packages

- Installing and managing R and its extensions can be pretty complicated. . .
- If printed, the [R Installation and Administration](#) manual would run over 100 pages.
- We don't have that kind of time.



## Highlights

- Problem: you want to maintain separate package libraries.
- You can always specify which library a package loads from:  
`library("foo", lib.loc="mylib/")`
- But what if you have a lot of packages to load...



## Highlights

- `.libPaths()` “gets/sets the library trees within which packages are looked for”.
- So if you have package `foo` in libraries `A` and `B`, setting `.libPaths(B)` will put `B` in the search path before `A`.
- Meaning package `foo` will load from library `B` (and `R` will look there first even if it's not installed there).
- `.Library` and `.Library.site` also exist, but are more specialized.
- `update.packages()` also exists.





# Contents

- 3 Writing R Packages
  - Package Structure
  - Getting on the GitHub
  - Getting on the CRAN



### 3 Writing R Packages

- Package Structure
- Getting on the GitHub
- Getting on the CRAN

## An Appeal: Write a Package

- Are you interested in reproducible research?
- Having functioning code a year from now?
- Collaboration?

**Put your code in an R package.**

## Package Structure

R packages: a place to put stuff

```
|  
├─ data/  
├─ demo/  
├─ inst/  
├─ man/  
├─ R/  
├─ src/  
├─ tests/  
├─ vignettes/  
├─ DESCRIPTION  
└─ NAMESPACE
```

## Package Development Resources

- [Writing R Extensions](#) (R Core)
- [R package primer](#) (Karl Broman)
- [R packages](#) (Hadley Wickham)
- [Developing Packages with RStudio](#) (Josh Paulson)
- Packages on CRAN, Bioconductor, GitHub, R-Forge, ...

## Git and GitHub Resources

Publishing to GitHub is very easy. . . assuming you use git.

- [git/github guide](#) (Karl Broman)
- [Try Git: Code School](#)
- [GitHub Bootcamp](#)

- 3 Writing R Packages
  - Package Structure
  - Getting on the GitHub
  - Getting on the CRAN

## The CRAN

Publishing to CRAN comes at varying degrees of difficulty, largely depending on how A CERTAIN SOMEONE is feeling that day.

- All submissions must pass R CMD check.





## The CRAN

Publishing to CRAN comes at varying degrees of difficulty, largely depending on how A CERTAIN SOMEONE is feeling that day.

- All submissions must pass R CMD check.
- You must read, abide by, and acknowledge the

## The CRAN

Publishing to CRAN comes at varying degrees of difficulty, largely depending on how A CERTAIN SOMEONE is feeling that day.

- All submissions must pass R CMD check.
- You must read, abide by, and acknowledge the [CRAN Repository Policy](#) (it changes from time to time).



## The CRAN

Publishing to CRAN comes at varying degrees of difficulty, largely depending on how A CERTAIN SOMEONE is feeling that day.

- All submissions must pass R CMD check.
- You must read, abide by, and acknowledge the [CRAN Repository Policy](#) (it changes from time to time).
- Submit via the [web form](#).



## The CRAN

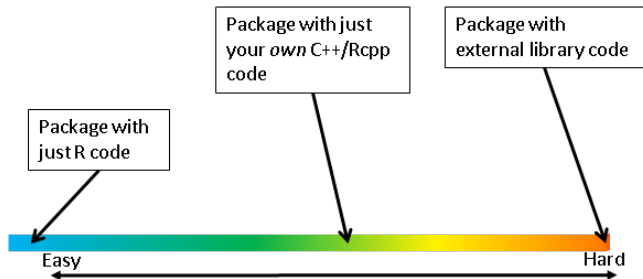
Publishing to CRAN comes at varying degrees of difficulty, largely depending on how A CERTAIN SOMEONE is feeling that day.

- All submissions must pass R CMD check.
- You must read, abide by, and acknowledge the [CRAN Repository Policy](#) (it changes from time to time).
- Submit via the [web form](#).
- Receive serenity in the knowledge that they will yell at and belittle you, but it doesn't make you a bad person. Just do what they say and move on with your life.

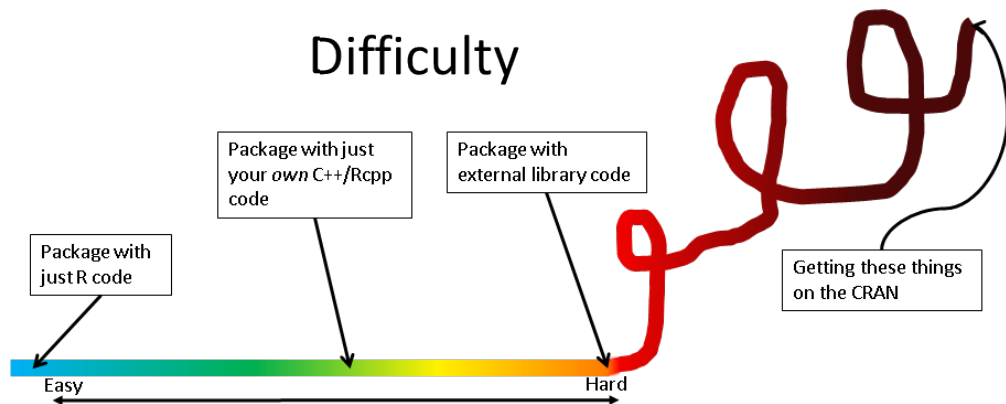


# Package Development Difficulty

## Difficulty



# Package Development Difficulty



# Getting on the CRAN



# Contents

## 4 Wrapup





## Summary

- R packages are great!
- You can easily maintain and use several libraries (e.g., multiple binary versions of packages)
- Making an R package isn't scary!
- Submitting it to CRAN is!



Thanks for coming!

# Questions?

I'm on the internet!

twitter: [@wrathematics](https://twitter.com/wrathematics)  
code: [github.com/wrathematics](https://github.com/wrathematics)  
blog: [librestats.com](https://librestats.com)

