

How (not to) submit a project to CRAN

David C. Sterratt
School of Informatics
`david.c.sterratt@ed.ac.uk`

March 22, 2015

Outline

CRAN

Packages

Dos

Don'ts

Further Reading

The comprehensive R archive network (CRAN)

- ▶ A tremendous boon for R users
- ▶ Allows easy installation of R software packages written by others

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

Installing package into ‘/home/dcs/share/R/x86_64-pc-linux-gnu/3.1’
(as ‘lib’ is unspecified)

also installing the dependencies ‘rstudioapi’, ‘roxygen2’

trying URL ‘http://cran.r-project.org/src/contrib/rstudioapi_0.2.tar.gz’

...

* DONE (devtools)

```
> library(devtools)
```

- ▶ Turning your code into a package, and getting the package onto CRAN is a great way of sharing and archiving your code

The CRAN maintainers

Do a fantastic job of making sure that:

- ▶ Packages compile on all platforms
- ▶ Code and documentation passes checks
- ▶ Licences are sound
- ▶ Package descriptions are adequate

However, with over 6000 packages on CRAN:

- ▶ They are busy people
- ▶ They don't have much time to help novices with the sometimes fiddly process of getting packages ready for CRAN and submitting them.
- ▶ They don't check that your algorithms work
- ▶ They don't Curate packages as do the Bioconductor repository maintainers

Overview of uploading a package to CRAN

1. Create the package, as described in the *Writing R Extensions* manual
2. Build the package R CMD build to create a .tar.gz (or .zip) package file
3. Run R CMD check --as-cran on this file, making sure that there are no WARNINGS or NOTES
4. Upload the package file to <http://cran.r-project.org/submit.html>
 - ▶ Comment might explain changes if this is a new upload
 - ▶ Read the CRAN policies before you check the box to agree to them
5. Receive email from CRAN admins (probably within a few hours or a day)
 - ▶ Package is accepted and appears on CRAN
 - ▶ Or needs more work; modify and go to 2.
6. A Windows build will appear a few days later

The package structure (*Writing R Extensions*)

- ▶ DESCRIPTION
- ▶ NAMESPACE Exposes functions to the world
- ▶ R/ The R code
- ▶ data/ Any data files (not too much)
- ▶ man/ Help files *in R documentation* (Rd) format.
- ▶ demo/ Demos, callable with `demo()`
- ▶ src/ C or FORTRAN source code
- ▶ tests/ Tests run by R CMD check
- ▶ vignettes/ "Vignettes"
- ▶ inst/ copied recursively to the installation directory
- ▶ tools/ Files needed during configuration
- ▶ exec/ Additional executable scripts the package needs, typically scripts for interpreters such as the shell, Perl, or Tcl.
- ▶ po/ Localisation

The DESCRIPTION file (*Writing R Extensions*)

```
Package: pkgname
Version: 0.5-1
Date: 2015-01-01
Title: My First Collection of Functions
Authors@R: c(person("Joe", "Developer", role = c("aut", "cre"),
  email = "Joe.Developer@some.domain.net"),
  person("Pat", "Developer", role = "aut"),
  person("A.", "User", role = "ctb",
    email = "A.User@whereever.net"))
Author: Joe Developer [aut, cre],
  Pat Developer [aut],
  A. User [ctb]
Maintainer: Joe Developer <Joe.Developer@some.domain.net>
Depends: R (>= 3.1.0), nlme
Suggests: MASS
Description: A (one paragraph) description of what
  the package does and why it may be useful.
License: GPL (>= 2)
URL: http://www.r-project.org, http://www.another.url
BugReports: http://pkgname.bugtracker.url
```

The NAMESPACE file (*Writing R Extensions*)

```
# Generated by roxygen2 (4.0.2): do not edit by hand
export(dot)
useDynLib(geometry)
```


Example of function with roxygen markup

- `devtools::document()` creates `.Rd` and `NAMESPACE` files from the following:

```
##' If {x} and {y} are matrices, calculate the dot-product  
##' along the first non-singleton dimension. If the optional argument  
##' {d} is given, calculate the dot-product along this  
##' dimension.  
##'  
##' @title Compute the dot product of two vectors  
##' @param x Matrix of vectors  
##' @param y Matrix of vectors  
##' @param d Dimension along which to calculate the dot product  
##' @return Vector with length of {d}th dimension  
##' @author David Sterratt  
##' @keywords arith math array  
##' @export  
dot <- function(x, y, d=NULL) {  
}
```

Dos: Before you write the package

Think about the LICENCE

- ▶ Free and open source licences (e.g. GPL-3, BSD) are good
- ▶ But check the licence if you are including others' code

Read the *Writing R extensions* manual on creating packages

- ▶ Package structure
- ▶ DESCRIPTION file format, including Dependencies: and Suggests:
- ▶ Interface for including code in C and FORTRAN
- ▶ How to build and check packages

Consider using roxygen2, devtools and testthat

- ▶ roxygen2 makes documenting functions easier
- ▶ devtools contains useful tools for building on Windows
- ▶ testthat helps with writing tests

Dos: Before submitting

Make sure you're using the latest R release

- ▶ New checks are often added with a new release

Run `R CMD check --as-cran`

- ▶ There must no WARNING and only one NOTE
- ▶ All functions must be documented

Check any C or FORTRAN code compiles OK on all platforms

- ▶ `devtools::build_win()` helps if you're not on Windows
- ▶ If code runs on GNU/Linux, it will probably run on Mac & vice-versa

Dos: Submitting

Submit packages through the web interface

- ▶ This is also true when resubmitting a package.

Read the CRAN Policies

- ▶ Really
- ▶ This small print is important
- ▶ If you say you've read them, but only skimmed them, the CRAN maintainers will smoke you out.

Keep communication with the maintainers short & to the point

- ▶ The CRAN maintainers are busy
- ▶ There's no need to send an email that just says "thank-you"

Don't forget to check before submission

On 14/03/2014 15:02, David C Sterratt wrote:

> [This was generated from CRAN.R-project.org/submit.html]

> The following package was uploaded to CRAN:

> =====

> Package: RImageJROI

...

> The maintainer confirms that he or she

> has read and agrees to the CRAN policies.

Please check before submission

* checking DESCRIPTION meta-information ... NOTE

Author field differs from that derived from Authors@R

Author: 'David C Sterratt <david.c.sterratt@ed.ac.uk>, Mikko Vihtakari <mikko.vihtakari@uit.no>'

Authors@R: 'David C Sterratt [aut, cph, cre], Mikko Vihtakari [aut,

* checking top-level files ... NOTE

Non-standard file/directory found at top level:

'RImageJROI_0.1.tar.gz'

Don't email a .tar.gz file directly to the CRAN maintainers

From: Prof Brian Ripley <ripley@stats.ox.ac.uk>

To: David Sterratt <david.c.sterratt@ed.ac.uk>

On 14/03/2014 15:29, David Sterratt wrote:

> Sorry about this - I'm using a different workflow from my normal,
> R-forge based one, and I'd forgotten to add the --as-cran option to R
> CMD check. New version attached - unless it would be better for me to
> submit it again via the web.

Have you actually read the policies: they explicitly ask you not to do that? Now several email archives need to be cleaned up

Don't incorporate code with a non-FOSS licence

From: Prof Brian Ripley <ripley@stats.ox.ac.uk>
To: David C. Sterratt <david.c.sterratt@ed.ac.uk>
Subject: CRAN package RTriangle
Date: Fri 13 Sep 2013 16:32:22 BST

This does not have a single licence for the package, and we took a closer look. As the LICENCE file includes

'This program may be freely redistributed under the condition that the copyright notices (including this entire header and the copyright notice printed when the '-h' switch is selected) are not removed, and no compensation is received.'

we do not have the rights we need of unrestricted redistribution (and some of the CRAN mirrors are commercial). So I am afraid we need to remove it from CRAN.

You could I believe avoid this by not distributing Triangle but providing an interface for those who already have it installed. You would have to ask R-forge if it is OK to distribute it from there.

Don't be offended

From: Prof Brian Ripley <ripley@stats.ox.ac.uk>

On 24/10/2013 10:23, David Sterratt wrote:

> On Fri, 2013-09-13 at 16:32 +0100, Prof Brian Ripley wrote:

>> This does not have a single licence for the package, and we took a
>> closer look. As the LICENCE file includes

>>

>> 'This program may be freely redistributed under the condition that
>> the copyright notices (including this entire header and the
>> copyright notice printed when the '-h' switch is selected) are not
>> removed, and no compensation is received.'

>>

>> we do not have the rights we need of unrestricted redistribution
>> (and some of the CRAN mirrors are commercial). So I am afraid we
>> need to remove it from CRAN.

>

> Is the problematic condition that "no compensation is received" or
> the retention of the copyright notices, or are both conditions
> problematic?

See the CRAN policies.

I should note that

- ▶ The CRAN maintainers did subsequently take a lot of time to help me finding a solution for the licence - which use to use a Creative Commons Non-Commercial licence.
- ▶ I think the work that they are doing on licensing is very valuable.

Don't rely on R-forge

- ▶ R-forge can be used to host code in a subversion repository
- ▶ It automatically builds packages (including windows binaries), making them available via the R-forge repository
- ▶ However it seems to suffer from quite a lot of downtime

Links

- ▶ The Bible of package authoring: *Writing R Extensions*
<http://cran.r-project.org/doc/manuals/r-release/R-exts.html>
- ▶ CRAN: <http://cran.r-project.org/>
- ▶ The R-Forge repository can be useful for development:
<https://r-forge.r-project.org/>
- ▶ Or you may prefer to use `install_github` from `devtools`
- ▶ <https://github.com/hadley/devtools>
- ▶ The *R Packages* book by Hadley Wickham looks to have useful information on the art of developing packages
<http://r-pkgs.had.co.nz/>