

# Quick R Package, And more

Laurent Gatto

[lg390@cam.ac.uk](mailto:lg390@cam.ac.uk)

RSS conference

4<sup>th</sup> Sept 2013

# Plan

## A minimal example

### More

- Dependencies

- Documentation

- The build/check/install cycle

- Vignettes

- Even more

### Distributing packages

## Overview

We are about to create the QuickPackage R package:

1. Create a dummy function to come into the package
2. Create the package structure
3. Update the package content
4. Build/check/install
5. Enjoy!

```
> ## This is the dummy function that will return  
> ## information about our 'QuickPackage'  
> qpf <- function()  
+   packageDescription("QuickPackage")
```

```
> ## This creates the package template  
> package.skeleton("QuickPackage", list = c("qpf"))
```

*Creating directories ...*

*Creating DESCRIPTION ...*

*Creating NAMESPACE ...*

*Creating Read-and-delete-me ...*

*Saving functions and data ...*

*Making help files ...*

*Done.*

*Further steps are described in*

*'./QuickPackage/Read-and-delete-me'.*

QuickPackage structure:

```
QuickPackage/  
|-- DESCRIPTION  
|-- man  
|   |-- qpf.Rd  
|   `-- QuickPackage-package.Rd  
|-- NAMESPACE  
|-- R  
|   `-- qfd.R  
`-- Read-and-delete-me
```

- ▶ Read and delete Read-and-delete-me
- ▶ Update DESCRIPTION
- ▶ Write a proper package documentation file or delete it (this is the only optional documentation) and update the man page for `qpf` (see next slide)

```
\name{qpf}  
\alias{qpf}  
\title{ A test function }  
\description{ Returns package information. }  
  
\usage{  
  qpf()  
}  
\value{  
  An object of class \code{packageDescription}.  
}  
\examples{  
  qpf()  
}
```



In a terminal:

1. Build the package with  
R CMD build QuickPackage
2. Check the package with  
R CMD check QuickPackage\_1.0.tar.gz
3. If any, fix errors and warnings and repeat steps 1 and 2.
4. Install the package with  
R CMD INSTALL QuickPackage\_1.0.tar.gz

```
> library("QuickPackage")  
> qpf()
```

Package: QuickPackage

Type: Package

Title: A quick and mini R package

Version: 1.0

Author: Laurent Gatto <lg390@cam.ac.uk>

Maintainer: Laurent Gatto <lg390@cam.ac.uk>

Description: Mini intro on building R packages.

License: GPL-2

Packaged: 2013-09-04 09:15:24 UTC; lgatto

Built: R 3.1.0; ; 2013-09-04 09:15:45 UTC; unix

-- File: /home/lgatto/R/x86\_64-unknown-linux-gnu-library/3

# Plan

## A minimal example

### More

- Dependencies

- Documentation

- The build/check/install cycle

- Vignettes

- Even more

## Distributing packages

## Terminology

A **package** is loaded from a **library** by the functions `library` or `require`. A library is a directory containing installed packages.

Calling `library("foo", lib.loc = "/path/to/bar")` loads the package (book) `foo` from the library `bar` located at `/path/to/bar`.

## Dependencies

In the DESCRIPTION file, should one add the package to the Depends and Imports field?

Is your user expected to use all/most of the functionality of the dependency? Would she anyway load it with `library` and add it to the search path?

## Imports specifications

Specify the imported symbols in the NAMESPACE with `import(package)`, `importFrom(package, function, class, method)`.

Also Suggests and Enhances.

## Rd

Different R objects must be documented with specific sections. All the details are available in the R-Ext reference.

Templates can be generated automatically using `prompt`, `promptData`, `promptClass`, `promptMethod` and, optionally, `promptPackage`.

## Inline documentation with the roxygen2 package.

```
##' A simple function for the QuickPackage demo.
##'
##' The function calls the \link{packageDescription}
##' function to retrieve the QuickPackage description.
##' @title The QuickPackage description
##' @return An object of class packageDescription.
##' @author Laurent Gatto
##' @examples
##' qpf()
qpf <-
function()
  packageDescription("QuickPackage")
```

```
> library("roxygen2")
> roxygenize("QuickPackage", roclets = "rd")
```

Other roclets: namespace and collate.

## Package development

Building/checking/installing a package to test every changes is not practical.

- ▶ Source the updated code in your R session. But the global environment does not correspond to the package namespace.
- ▶ Load the new code directly into the package namespace using
  - ▶ The devtools package
  - ▶ ESS developer mode
  - ▶ RStudio



## Reproducible research

**Vignettes** are documents that combine text and R code. When compiled, the R code is executed and the output (text, figures, data.frame/tables) is inserted in the original document source and a pdf (or html) is generated. The syntax is originally  $\text{\LaTeX}$ . Recent support for markdown has been added. See `?Sweave` and the `knitr` package for more information.

In packages, Sweave documents (`.Rnw`) are provided in the `inst/doc` or `vignettes` directories.

These slides are written as a Sweave document and processed using `knitr`.

## Even more

- ▶ CITATION and NEWS files.
- ▶ Distributing data in the `./data` or `inst/extdata` directories.
- ▶ C/C++/Fortran code in the `./src` directory.
- ▶ Vignettes and reproducible research
- ▶ Unit testing in the `./tests` and `inst/tests` directories.

# Plan

## A minimal example

### More

- Dependencies

- Documentation

- The build/check/install cycle

- Vignettes

- Even more

## Distributing packages

# Submission

**CRAN** Read the CRAN Repository Policy<sup>1</sup>. Upload your `--as-cran checked myPackage_x.y.z.tar.gz` to `ftp://cran.R-project.org/incoming` or using `http://CRAN.R-project.org/submit.html`. Upon acceptance, your package will be installable with `install.packages("myRpackage")`.

**R-forge** Log in, register a project and wait for acceptance. Then commit you code to the svn repository. Your package will be installable with `install.packages` using `repos="http://R-Forge.R-project.org"`.

---

<sup>1</sup><http://cran.r-project.org/web/packages/policies.html>

# Submission

**Bioconductor** Make sure to satisfy submission criteria (pass check, have a vignette, use S4 if OO, make use of appropriate existing infrastructure, include a NEWS file, must **not** already be on CRAN, ...) and submit by email. Your package will then be reviewed before acceptance. Upon acceptance, an svn account will be created. Package will be installable with `biocLite("myPackage")`.

**Other** popular un-official repositories are github, bitbucket, ... and packages can be installed with `devtools::install_github,`  
`devtools::install_bitbucket.`

## References

- ▶ R package development, Robert Stojnic and Laurent Gatto  
<https://github.com/lgatto/RPackageDevelopment>
- ▶ Writing R Extensions, R Development Core Team, (get it with `help.start()`)
- ▶ This work is licensed under a CC BY-SA 3.0 License.
- ▶ Course web page and more material:  
<https://github.com/lgatto/TeachingMaterial>

```
> toLatex(sessionInfo())
```

- ▶ R Under development (unstable) (2013-06-16 r62969),  
x86\_64-unknown-linux-gnu
- ▶ Locale: LC\_CTYPE=en\_GB.UTF-8, LC\_NUMERIC=C,  
LC\_TIME=en\_GB.UTF-8, LC\_COLLATE=en\_GB.UTF-8,  
LC\_MONETARY=en\_GB.UTF-8, LC\_MESSAGES=en\_GB.UTF-8,  
LC\_PAPER=C, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C,  
LC\_MEASUREMENT=en\_GB.UTF-8, LC\_IDENTIFICATION=C
- ▶ Base packages: base, datasets, graphics, grDevices, methods,  
stats, utils
- ▶ Other packages: digest 0.6.3, knitr 1.4.1, QuickPackage 1.0,  
roxygen2 2.2.2
- ▶ Loaded via a namespace (and not attached): brew 1.0-6,  
evaluate 0.4.7, formatR 0.9, highr 0.2.1, stringr 0.6.2,  
tools 3.1.0