Writing Package Vignettes

Duncan Murdoch

Department of Statistical and Actuarial Sciences
University of Western Ontario

November 29, 2013

Outline

1. Why Write Packages?
2. What are Vignettes?
3. Mechanics of Writing Vignettes
4. How to Write a Good Vignette

Outline

1. Why Write Packages?

2. What are Vignettes?

3. Mechanics of Writing Vignettes

4. How to Write a Good Vignette

R is mostly packages!

- R ships with 13 base packages, and 15 recommended packages.
- CRAN contains about 5000 packages, Bioconductor has about 1600.
- There are other repositories (R-forge, Omegahat, rforge.net, etc.) and packages not in these repositories.

What is in a package?

- Permanent R objects: functions, data, etc.
- Help pages and vignettes documenting these objects.
- External code in C, C++, Fortran, Objective C, etc. to implement some of the functions, or link to external libraries or programs.
- Tests to help to keep the code working as R evolves.

Packages, libraries, repositories?

- We use

  ```
  > library(foo)
  ```

  to load the package and put it on the search list, but a package is not a library.
- A library is a collection of packages installed on your system. Use

  ```
  > dir.create("newlib")
  > .libPaths("newlib")
  ```

  to create a new one, and add it as the first place to look.
- A repository is a collection of packages like CRAN, usually available online. Use install.packages() to install a package from a repository into your library.

Not everyone uses packages

Packages are great, but they aren’t the only ways to save code and data. There are also

- binary images using save(), save.image() or q("yes")
- R scripts in plain text files

What's wrong with saving your workspace?

- Saved images are hard to work with: they are black boxes outside of R.
- It is very easy to save more than you intended, and get bloated saves, and unintended interactions.
- It is easy to forget how some objects were created.

Working with scripts and vignettes

- Scripts are easy to transport and edit on any platform.
- It is easy to see what's there (if you format your code nicely...)
- You can have a permanent record of how research results were produced.

But...

- It is hard to re-use parts of scripts.
- Cut and paste is error prone.
- It is hard to remember which earlier part of a script needs to be re-executed, and which doesn't.

So why packages?

- Packages combine the good aspects of saved images and scripts.
- R packages can be distributed to others.
- R tools support quality control checks.

So why packages?

- Packages combine the good aspects of saved images and scripts.
- R packages can be distributed to others.
- R tools support quality control checks.

Today's talk isn't about packages, it's about vignettes!

Outline

1. Why Write Packages?
2. What are Vignettes?
3. Mechanics of Writing Vignettes
4. How to Write a Good Vignette

What are Vignettes?

They are not easy to define.

What are Vignettes?

They are not easy to define.

- Up to the release of R 3.0.0, I could say that vignettes are Sweave documents included in a package and intended to document that package.

What are Vignettes?

They are not easy to define.

- Up to the release of R 3.0.0, I could say that vignettes are Sweave documents included in a package and intended to document that package.
- Now they don't need to be Sweave! We can have vignettes from **knitr** (Xie, 2013) or other packages, and they can produce PDF or HTML output.

What are Vignettes?

They are not easy to define.

- Up to the release of R 3.0.0, I could say that vignettes are Sweave documents included in a package and intended to document that package.
- Now they don't need to be Sweave! We can have vignettes from **knitr** (Xie, 2013) or other packages, and they can produce PDF or HTML output.

My current definition

> *Vignettes are documentation in R packages that may include R code. The code is run when the vignette is “woven”, and the code can be extracted when the vignette is “tangled”. The vignette source is in the* `vignettes` *directory of the package.*

Vignettes are considered to be part of R’s help system. The help system will display their title, and links to the PDF, source, and extracted R code. This is based on some special “meta-data” included in the source.

Outline

1 Why Write Packages?

2 What are Vignettes?

3 Mechanics of Writing Vignettes

4 How to Write a Good Vignette

Vignettes are usually Sweave documents

If you put a `.Rnw` file in the `vignettes` directory of a package, R will treat it as a vignette:

1. It will check for indexing and other meta-data.
2. It will process the file to convert it to a PDF (or HTML) file.
3. It will include the PDF file in the package when built. (Prior to 3.0.2, the vignette was included at installation time, not build time.)

Meta-data in Vignettes

Meta-data looks like a LaTeX macro in a comment:

```
%\VignetteIndexEntry{About the tables package}
%\VignetteDepends{MASS}
%\VignetteEngine{knitr::knitr}
```

It is processed by R and ignored by LaTeX.

Non-Sweave vignettes

- To create a vignette using an engine other than Sweave, put the `%\VignetteEngine{pkg::driver}` line in the source file.
- Even if the vignette is not intended to be processed into LaTeX code, use this format. E.g. **knitr** allows vignettes to be written in Markdown, and puts the meta-data in a Markdown comment:

```
<!--
%\VignetteEngine{knitr::knitr}
%\VignetteIndexEntry{An R Markdown ...}
-->
```

Outline

1 Why Write Packages?

2 What are Vignettes?

3 Mechanics of Writing Vignettes

4 How to Write a Good Vignette

What are vignettes for?

Vignettes document packages:

- They are the best way to present an overall introduction to a package.

What are vignettes for?

Vignettes document packages:

- They are the best way to present an overall introduction to a package.
- They can give technical details about how a package was implemented.

What are vignettes for?

Vignettes document packages:

- They are the best way to present an overall introduction to a package.
- They can give technical details about how a package was implemented.
- They can demonstrate particular aspects of the package.

What are vignettes for?

Vignettes document packages:

- They are the best way to present an overall introduction to a package.
- They can give technical details about how a package was implemented.
- They can demonstrate particular aspects of the package.

What are vignettes not good for?

- It is not a good idea to include reference documentation in a vignette as **`tables`** (Murdoch, 2013) does. R has strong support to ensure that the regular help pages in packages are consistent with the code; there is no support to make sure the vignette stays consistent.

What are vignettes not good for?

- It is not a good idea to include reference documentation in a vignette as **`tables`** (Murdoch, 2013) does. R has strong support to ensure that the regular help pages in packages are consistent with the code; there is no support to make sure the vignette stays consistent.
- It is not a good idea to break up the documentation into too many small pieces (as **`grid`** does). Users will find it inconvenient to have to switch between too many files.

What are vignettes not good for?

- It is not a good idea to include reference documentation in a vignette as **`tables`** (Murdoch, 2013) does. R has strong support to ensure that the regular help pages in packages are consistent with the code; there is no support to make sure the vignette stays consistent.
- It is not a good idea to break up the documentation into too many small pieces (as **`grid`** does). Users will find it inconvenient to have to switch between too many files.
- HTML is a different medium than PDF, so it might make sense to have a lot of small HTML vignettes, with links to each other; these are very new and I don't have much experience using them.

What are vignettes not good for?

- It is not a good idea to include reference documentation in a vignette as **`tables`** (Murdoch, 2013) does. R has strong support to ensure that the regular help pages in packages are consistent with the code; there is no support to make sure the vignette stays consistent.
- It is not a good idea to break up the documentation into too many small pieces (as **`grid`** does). Users will find it inconvenient to have to switch between too many files.
- HTML is a different medium than PDF, so it might make sense to have a lot of small HTML vignettes, with links to each other; these are very new and I don't have much experience using them.

References

Duncan Murdoch. *tables: Formula-driven table generation*, 2013. R package version 0.7.64, on CRAN.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, 2013. URL `http://yihui.name/knitr/`. ISBN 978-1482203530.