Sweave, Part II: Package Vignettes

Reading, writing and interacting with R package primers in Sweave format.

by Friedrich Leisch

This is the second article in a two-part miniseries on Sweave (Leisch, 2002a), a tool that allows embedding of R code in LaTeX documents so that code, results, and descriptions are presented in a consistent way. The first article (Leisch, 2002b) introduced the Sweave file format and the R functions to process it, and demonstrated how to use Sweave as a reporting tool for literate statistical practice. This article will concentrate on how to use files in Sweave format as primers or manuals for R packages, so that users have direct access to the code examples shown and so that all code can be checked automatically for syntax errors and for consistency of the usage description with the implementation.

R package documentation

The main vehicle for documenting R packages are the help files, which are the sources, written in R documentation (Rd) format, of what you see after calling `help()` on a topic. These files, which are divided into sections, typically contain code in just two sections: usage and examples. All examples in the R help files are, by default, required to be executable such that the user can copy & paste the code to a running R process using

- the mouse,
- keyboard shortcuts if running R inside Emacs with ESS, or
- R's `example()` function.

Examples should be flagged as non-executable only if there are good reasons for this, e.g. because they require user interactivity like `identify()` and hence cannot be executed in batch mode.

The tools for package quality control, available through the `R CMD check`[1] command, test if all the examples in the package can be successfully executed. Furthermore, the code in the usage section is compared with the actual implementation to check for inconsistencies and for missing documentation.

The Rd file format was designed for reference documentation on single R objects (functions, classes, data sets, . . .). It is not intended for demonstrating the interaction of multiple functions in a package. For this task we have developed the concept of *package vignettes* — short to medium-sized documents explaining parts or all of the functionality of a package in a more informal tone than the strict format of reference help pages.

Reading vignettes

Books like Venables and Ripley (2002) that describe the use of S for data analysis typically contain a mixture of documentation, code, and output. Short documents in this style are ideally suited to explaining the functionality of a package to new users. The directory 'inst/doc' of an R source package may contain such package documentation in any format, although we recommend PDF files because of their platform independence.

We call a user guide located in 'inst/doc' a vignette only when it is a document from which the user can extract the R code and interact with it. Sweave is the only format for such documents that is currently supported by R; there may be others in the future. In short: every vignette is a user guide, but not every user guide is a vignette.

Command line interface

Starting with R version 1.8.0 there is support in base R for listing and viewing package vignettes. The `vignette()` function works similar to `data()` and `demo()`. If no argument is given, a list of all vignettes in all installed packages is returned — see the example R session in Figure 1.

Do not be surprised if this list is rather short or even empty on your computer. At present only a few of the packages on CRAN have vignettes. For Bioconductor we have made package vignettes a requirement and thus all Bioconductor packages provide one or more vignettes.

Following the title of each vignette listed in Figure 1, you will see in parenthesis a list of the formats that are available. In Figure 1 all the vignettes are available in both source and PDF format. To view the `strucchange-intro` vignette, all you need to do is to issue

```
R> vignette("strucchange-intro")
```

and the PDF file is opened in your favorite PDF reader (exactly which PDF reader depends on the platform that you use). If the source file for a vignette is available, one can easily extract the code shown in the vignette, although we have not yet fully automated the procedure. First we get the full path to the vignette directory

```
R> vigdir =
+     system.file("doc", package="strucchange")
```

and then we examine the names of the files it contains

---

[1] `R CMD xxx` is `Rcmd xxx` in the Windows version of R.

```
R> vignette()

Vignettes in package 'AnnBuilder':

AnnBuilder                AnnBuilder Basic (source, pdf)
HowTo                     AnnBuilder HowTo (source, pdf)


Vignettes in package 'Biobase':

Biobase                   Biobase Primer (source, pdf)
Bioconductor              Howto Bioconductor (source, pdf)
HowTo                     HowTo HowTo (source, pdf)
esApply                   esApply Introduction (source, pdf)


...


Vignettes in package 'strucchange':

strucchange-intro         strucchange: An R Package for Testing for
                          Structural Change in Linear Regression Models
                          (source, pdf)
...
```

Figure 1: Usage of `vignette()` to list available package vignattes.

```
R> list.files(vigdir)
[1] "00Index.dcf"
[2] "strucchange-intro.R"
[3] "strucchange-intro.Rnw"
[4] "strucchange-intro.pdf"
[5] "strucchange-intro.tex"
```

File 'strucchange-intro.Rnw' is the original Sweave file, 'strucchange-intro.R' has the extracted R code for all code chunks and could now be executed using `source()` or opened in your favorite editor. If the '.R' file is not available, we can create it in the current working directory by

```
R> library("tools")
R> vig = listFilesWithType(vigdir, "vignette")
R> Stangle(vig[1])
Writing to file strucchange-intro.R
```

where `listFilesWithType()` returns the full path to all files in `vigdir` that have type "vignette", i.e., an extension marking them as Sweave files.

Graphical user interface

The simplest way to access vignettes is probably through the HTML help system. If you execute `help.start()` and click your way to a package containing vignettes, then you will see, at the beginning of the package's table of contents, a link to an index of all the available vignettes. In addition there is a link to the directory containing the vignettes so that, for example, you could use your browser to examine the source files.

A more advanced interface to package vignettes is available in the Bioconductor package **tkWidgets**, available from http://www.bioconductor.org.

Function `vExplorer()` lists all available vignettes in a nice point & click menu. For example, after selecting the **strucchange** vignette the upper left window shown in Figure 2 is opened. The PDF version of the vignette can be opened by clicking on the "View PDF" button. Each code chunk of the vignette has a button on the left side of the window, clicking on the button shows the code in the "R Source Code" text field. The code can be executed and the resulting output is shown in the "Results of Execution" area.

The most powerful feature of this kind of interface is that the S code in the source code field can be modified by the user, e.g., to try variations of the pre-fabricated examples. To modify the example, one simply edits the code in the "R Source Code" area and presses the "Execute Code" button again.

Dynamic statistical documents and their user interfaces are an open research area, see also Buttrey et al. (2001) and Sawitzki (2002) for other approaches.

Writing vignettes

Once the Sweave file is written (we cannot do that for you), it is almost trivial to include it in an R package and make it available to users by the tools described above. Say you have written a file 'foo.Rnw' to be used as a vignette for package **foo**. First you need to add some meta-information to the file along the lines of

```
% \VignetteIndexEntry{An R Package for ...}
% \VignetteDepends{foo, bar, ...}
% \VignetteKeyword{kwd1}
% \VignetteKeyword{kwd2}
```
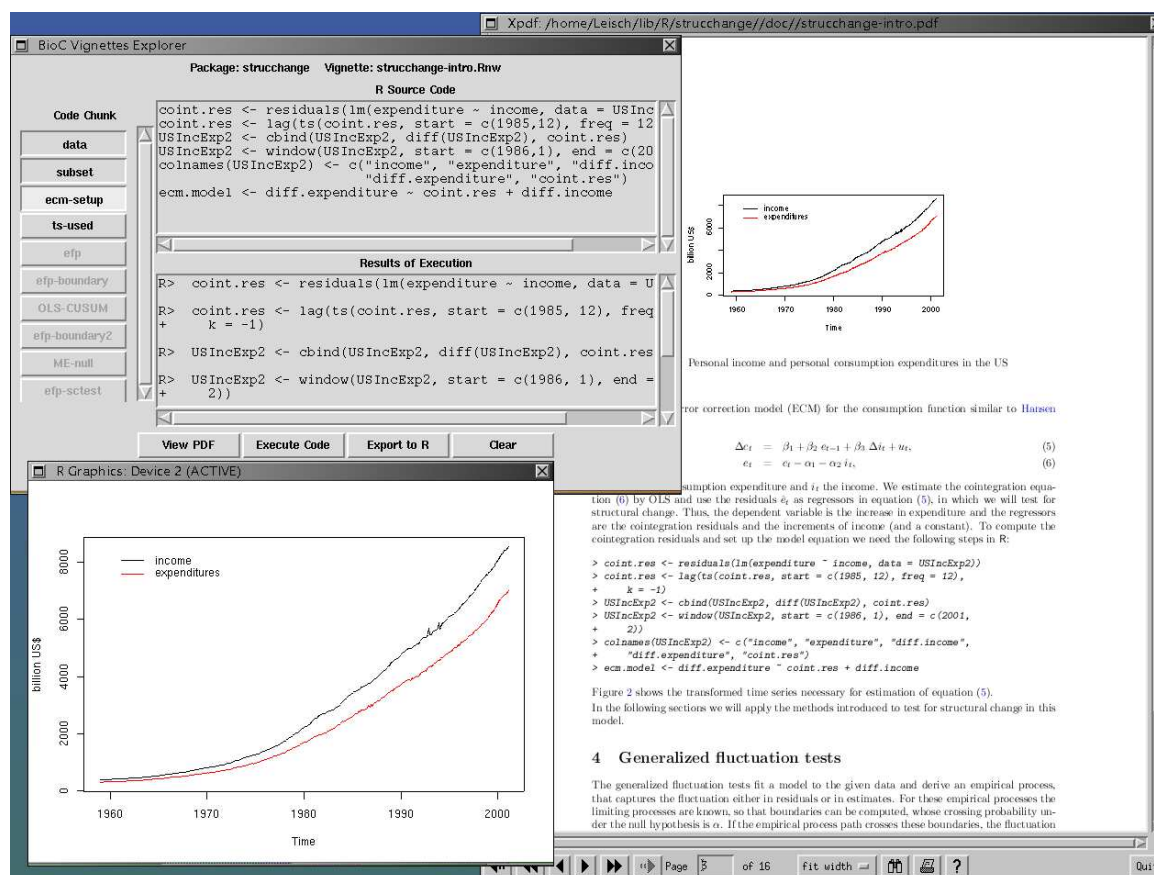
Figure 2: Screenshot of `vExplorer()` showing the vignette from package **strucchange**: main controls for code chunk execution (upper left), currently active R graphics window (lower left) and a pdf viewer (right).

All of these should be in LaTeX comments as we have not defined them as proper LaTeX commands. The index entry is used for the listings of `vignette()` or `vExplorer()`; frequently it is the same as the title of the document (or an abbreviated version thereof). Note that it is directly used in text and HTML files and hence should not contain any TeX markup. The dependency information is analogous to the `Depends` field of a package 'DESCRIPTION' file and lists packages needed to execute the code in the vignette. The list of `\VignetteXXX` meta-information specifications will probably get longer in the near future, especially for versioning etc.

Once this is done all you have to do is create a subdirectory 'inst/doc' in yor package source tree and copy 'foo.Rnw' to it. All the rest is taken care of by the R package management system, e.g.

- `R CMD check` will extract the code from the vignette and test that it can be executed successfully.

- `R CMD build` will run `Sweave()` and `pdflatex` on the vignette to create the PDF version.

- The package installation mechanism creates an index of all vignettes in the package and links it into the HTML help system.

Note that even code chunks with option `eval=FALSE` are tested by `R CMD check`; if you want to include code that should not be tested in a vignette, move it to a normal LaTeX verbatim environment. The reason for this policy is because users should be able to rely on code examples being executable exactly as shown in the vignette.

By including the PDF version in the package sources it is not necessary that the vignettes can be compiled at install time, i.e., the package author can use private LaTeX extensions or BibTeX files. Only the R code inside the vignettes is part of the checking procedure; typesetting manuals is not part of package quality control.

For more details see the manual "Writing R Extensions", which features a section on package vignettes.

In general it is assumed that package *authors* run `R CMD build` on their machine (and may safely assume that only they do that). `R CMD check` on the other hand should be runnable by *everybody*, e.g., CRAN runs a check on all 250+ packages on a daily basis (the results are available at `http://cran.r-project.org/src/contrib/checkSummary.html`). Bioconductor has opted for a stricter policy such that even building packages (including running `latex` on vignettes) should be re-

producible on every machine which has the necessary tools installed.

Acknowledgements

`vignette()` and most of `R CMD check` were written by Kurt Hornik. `vExplorer()` and its helper functions were written by Jeff Gentry and Jianhua Zhang as part of the Bioconductor project. I want to thank them and Robert Gentleman for helpful ideas and discussions.

Bibliography

S. E. Buttrey, D. Nolan, and D. T. Lang. An environment for creating interactive statistical documents. In E. J. Wegman, A. Braverman, A. Goodman, and P. Smyth, editors, *Computing Science and Statistics*, volume 33. Interface Foundation of North America, Fairfax Station, VA, USA, 2001. 22

F. Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In W. Härdle and B. Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physika Verlag, Heidelberg, Germany, 2002a. URL http://www.ci.tuwien.ac.at/~leisch/Sweave. ISBN 3-7908-1517-9. 21

F. Leisch. Sweave, part I: Mixing R and LATEX. *R News*, 2(3):28–31, December 2002b. URL http://CRAN.R-project.org/doc/Rnews/. 21

G. Sawitzki. Keeping statistics alive in documents. *Computational Statistics*, 17:65–88, 2002. 22

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S. Fourth Edition*. Springer, 2002. URL http://www.stats.ox.ac.uk/pub/MASS4/. ISBN 0-387-95457-0. 21

Friedrich Leisch
Institut für Statistik & Wahrscheinlichkeitstheorie
Technische Universität Wien, Austria
Friedrich.Leisch@R-project.org