Sweave, Part I: Mixing R and LaTeX

A short introduction to the Sweave file format and corresponding R functions

by Friedrich Leisch

This is the first article in a two part mini series on Sweave (Leisch, 2002), a tool that allows to embed the R code for complete data analyses in LaTeX documents. In this issue we will introduce the Sweave file format and R functions to process it, and demonstrate how to use Sweave as a reporting tool for literate statistical practice (Rossini, 2001). The companion article scheduled for the next issue of R News will concentrate on how to use files in Sweave format to write primers or manuals for R packages that can be automatically checked for syntax errors in the code or inconsistencies between examples and implementation.

The traditional way of writing a report as part of a statistical data analysis project uses two separate steps: First, the data are analyzed, and afterwards the results of the analysis (numbers, graphs, ...) are used as the basis for a written report. In larger projects the two steps may be repeated alternately, but the basic procedure remains the same. R supports this in a number of ways: graphs can be saved as PDF, EPS, or WMF which in turn can be included in LaTeX or Word documents. LaTeX tables can be created by specifying the columns and row separators in `write.table()` or using the package **xtable**. The basic paradigm is to write the report around the results of the analysis.

The purpose of Sweave is to create dynamic reports, which can be updated automatically if data or analysis change. Instead of inserting a prefabricated graph or table into the report, the master document contains the *R code* necessary to obtain it. When run through R, all data analysis output (tables, graphs, ...) is created on the fly and inserted into a final LaTeX document. The report can be automatically updated if data or analysis change, which allows for truly reproducible research.

A small example

Sweave source files are regular noweb files (Ramsey, 1998) with some additional syntax that allows control over the final output. Noweb is a simple literate programming tool which allows to combine program source code and the corresponding documentation into a single file. These consist of a sequence of code and documentation segments, called *chunks*. Different command line programs are used to extract the code (*"tangle"*) or typeset documentation together with the code (*"weave"*).

A small Sweave file is shown in Figure 1, which contains four code chunks embedded in a simple LaTeX document. '`<<...>>=`' at the beginning of a line marks the start of a code chunk, while a '`@`' at the beginning of a line marks the start of a documentation chunk. Sweave translates this into a regular LaTeX document, which in turn can be compiled by `latex` to Figure 2.

The code chunks

The main work of Sweave is done on the code chunks. All code chunks are evaluated by R in the order they appear in the document[1]. Within the double angle brackets we can specify options that control how the code and the corresponding output are rendered in the final document. The first code chunk (lines 5–8 in Figure 1) declares that neither the R code (`echo=false`) nor output (`results=hide`) shall be included. The purpose of this chunk is to initialize R by loading packages and data, we want to hide these technical details from the reader.

Let us skip the text in lines 10–19 for the moment and go directly to the next code chunk in lines 20–22. It uses the default settings for all options (nothing is specified within the double angle brackets): both input and output are shown to the user (see Figure 2), the chunk is rendered such that it emulates the R console when the code is typed at the prompt. All input and output are automatically encapsulated in verbatim-like environments.

The next code chunk can be found at lines 30–31. It uses the package **xtable** to pretty-print the coefficient matrix of the linear regression model. By specifying `results=tex` we tell Sweave that the output of this code chunk is regular TeX code and hence needs no protection by a verbatim environment.

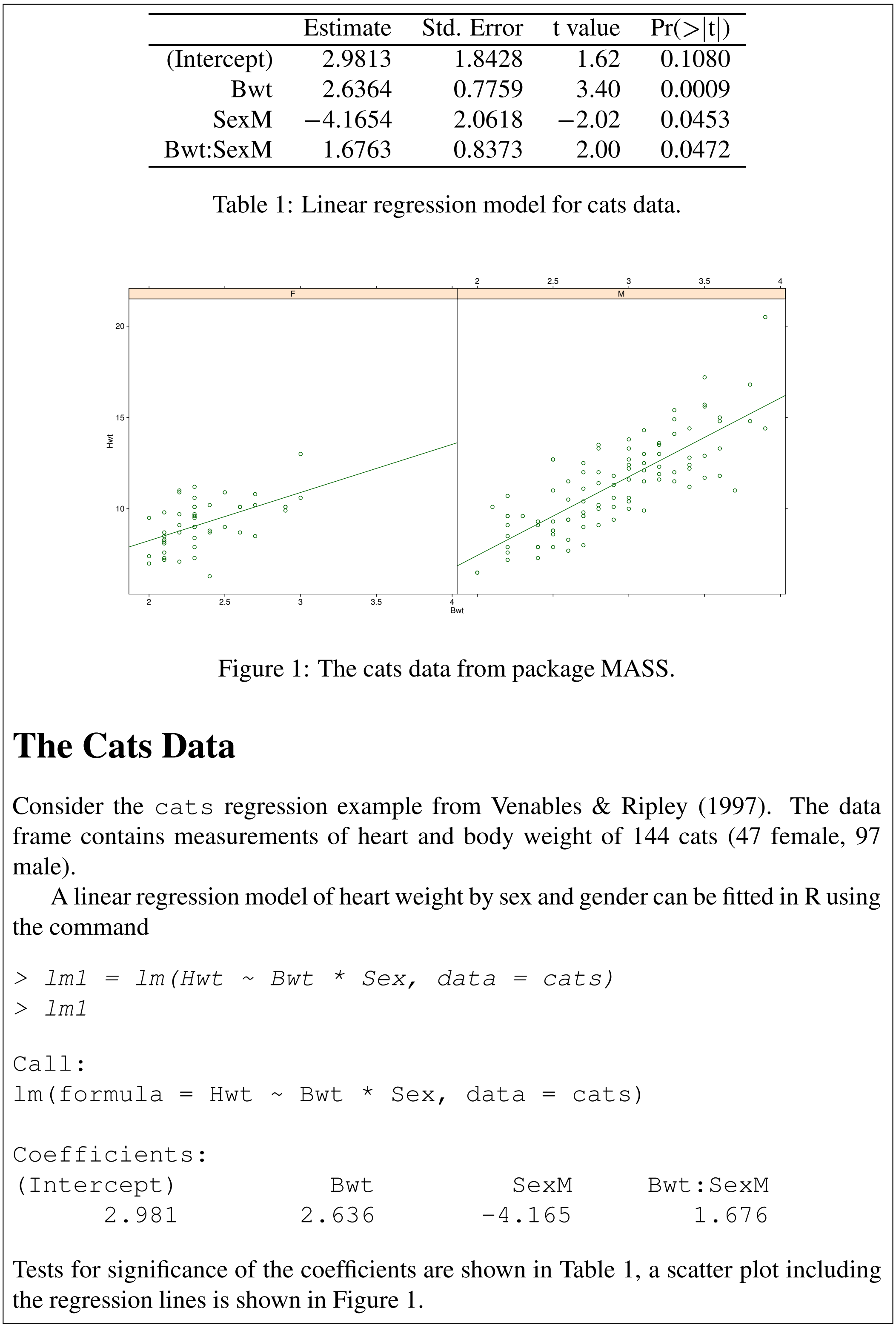
The last code chunk in lines 36–38 is marked as a figure chunk (`fig=true`) such that Sweave creates EPS and PDF files corresponding to the plot created by the commands in the chunk. Furthermore, an `\includegraphics{}` statement is inserted into the LaTeX file. Options `width` and `height` are passed to R's graphics devices and determine the size of the figure in the EPS and PDF files.

In line 28 we use `\SweaveOpts{echo=false}` to modify the default for option `echo` to the value of `false` for all code chunks following, hence the code for the last two chunks is not shown in Figure 2. It has exactly the same effect as if we had included `echo=false` within the double angle brackets of the two chunks.

[1] There are ways to suppress evaluation or re-use chunks, which is beyond the scope of this article.

```
\documentclass[a4paper]{article}

\begin{document}

<<echo=false,results=hide>>=
library(lattice)
library(xtable)
data(cats, package="MASS")
@

\section*{The Cats Data}

Consider the \texttt{cats} regression example from Venables \& Ripley
(1997). The data frame contains measurements of heart and body weight
of \Sexpr{nrow(cats)} cats (\Sexpr{sum(cats$Sex=="F")} female,
\Sexpr{sum(cats$Sex=="M")} male).

A linear regression model of heart weight by sex and gender can be
fitted in R using the command
<<>>=
lm1 = lm(Hwt~Bwt*Sex, data=cats)
lm1
@
Tests for significance of the coefficients are shown in
Table~\ref{tab:coef}, a scatter plot including the regression lines is
shown in Figure~\ref{fig:cats}.

\SweaveOpts{echo=false}

<<results=tex>>=
xtable(lm1, caption="Linear regression model for cats data.", label="tab:coef")
@

\begin{figure}
  \centering
<<fig=true,width=12,height=6>>=
lset(col.whitebg())
print(xyplot(Hwt~Bwt|Sex, data=cats, type=c("p", "r")))
@
  \caption{The cats data from package MASS.}
  \label{fig:cats}
\end{figure}

\end{document}
```

Figure 1: A minimal Sweave file: `example.Snw`.



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.9813	1.8428	1.62	0.1080
Bwt	2.6364	0.7759	3.40	0.0009
SexM	−4.1654	2.0618	−2.02	0.0453
Bwt:SexM	1.6763	0.8373	2.00	0.0472

Table 1: Linear regression model for cats data.

Figure 1: The cats data from package MASS.

The Cats Data

Consider the `cats` regression example from Venables & Ripley (1997). The data frame contains measurements of heart and body weight of 144 cats (47 female, 97 male).

A linear regression model of heart weight by sex and gender can be fitted in R using the command

```
> lm1 = lm(Hwt ~ Bwt * Sex, data = cats)
> lm1

Call:
lm(formula = Hwt ~ Bwt * Sex, data = cats)

Coefficients:
(Intercept)          Bwt         SexM     Bwt:SexM
      2.981        2.636       -4.165        1.676
```

Tests for significance of the coefficients are shown in Table 1, a scatter plot including the regression lines is shown in Figure 1.

Figure 2: The final document is created by running `latex` on the intermediate file 'example.tex' created by `Sweave("example.Snw")`.

Using S objects in text

Let us now return to the text paragraph in lines 13–16. It contains three `\Sexpr{}` statements. Sweave replaces them by the value of the corresponding S expression, which should be a simple character string (or something that can be coerced to a string by `as.character()`). In the example we use it to avoid hard-coding the size of the data set. If the number of observations changes we do not need to change anything in our Sweave file, we simply re-run `Sweave()` and `latex` and the report is up-to-date.

Writing Sweave files

The Emacs text editor offers a perfect authoring environment for Sweave, especially for people who already use Emacs for writing LaTeX documents and interacting with R. ESS (*Emacs speaks statistics*, Rossini et al., 2003) allows to connect an Sweave file to a running R process while writing the document. Code chunks can be sent to R and evaluated using simple keyboard shortcuts or popup menus. Syntax highlighting, automatic indentation and keyboard shortcuts depend on the location of the pointer: in documentation chunks Emacs behaves as if editing a standard LaTeX file, when the pointer moves to a code chunk the mode switches automatically to S programming.

However, it is not necessary to use Emacs, Sweave is a standalone system, the noweb source files for Sweave can be written using any text editor. Even the noweb syntax in not a necessity, because Sweave is highly configurable. Currently there are two syntaxes available, the noweb syntax described above and a LaTeX-based syntax. In LaTeX syntax the first code chunk of the example looks like

```
\begin{Scode}{echo=false,results=hide}
  library(lattice)
  library(xtable)
  data(cats, package="MASS")
\end{Scode}
```

Processing Sweave files

Sweave is contained in the standard R package **tools** (R version 1.5.0 or higher). The Sweave file 'example.Snw' can be processed using the R commands

```
> library(tools)
> Sweave("example.Snw")
Writing to file example.tex
Processing code chunks ...
 1 : term hide
 2 : echo term verbatim
 3 : term tex
 4 : term verbatim eps pdf

You can now run LaTeX on example.tex
```

Sweave shows a status line per code chunk indicating which options are active. The companion command

```
R> Stangle("example.Snw")
Writing to file example.R
```

can be used to extract the code of all chunks into an R source file.

Resources and summary

The example Sweave file used in this article can be found as 'example-3.Snw' at the Sweave homepage http://www.ci.tuwien.ac.at/~leisch/Sweave, where you also find a manual and more examples.

Sweave is already used for a wide variety of applications: Reports for medical statistical consulting that can be updated automatically when new data arrive or data change; lecture notes for statistics classes with executable examples; and manuals with embedded examples for R packages that can tested as part of the `R CMD check` suite, so-called package *vignettes*. The last application will be the topic of the second part of this article in the next issue of R News.

Bibliography

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, 2002. URL http://www.ci.tuwien.ac.at/~leisch/Sweave. ISBN 3-7908-1517-9. 28

N. Ramsey. *Noweb man page*. University of Virginia, USA, 1998. URL http://www.cs.virginia.edu/~nr/noweb. version 2.9a. 28

A. Rossini. Literate statistical analysis. In K. Hornik and F. Leisch, editors, *Proceedings of the 2nd International Workshop on Distributed Statistical Computing, March 15-17, 2001, Technische Universität Wien, Vienna, Austria*, 2001. URL http://www.ci.tuwien.ac.at/Conferences/DSC-2001/Proceedings/. ISSN 1609-395X. 28

A. J. Rossini, R. M. Heiberger, R. Sparapani, M. Mächler, and K. Hornik. Emacs speaks statistics: A multiplatform, multi-package development environment for statistical analysis. *Journal of Computational and Graphical Statistics*, 2003. (Accepted for publication). 31

Friedrich Leisch
Department of Statistics & Decision Support Systems
University of Vienna, Austria
leisch@R-project.org