```
---------------
Name: Econometrics
Topic: Computational Econometrics
Maintainer: Achim Zeileis
Repository: http://cran.r-project.org
---------------
Name: Finance
Topic: Empirical Finance
Maintainer: Dirk Eddelbuettel
Repository: http://cran.r-project.org
---------------
Name: MachineLearning
Topic: Machine Learning&Statistical Learning
Maintainer: Torsten Hothorn
Repository: http://cran.r-project.org
---------------
Name: gR
Topic: gRaphical models in R
Maintainer: Claus Dethlefsen
Repository: http://cran.r-project.org
```

```
R> x[[1]]
```

```
CRAN Task View
--------------
Name:       Econometrics
Topic:      Computational Econometrics
Maintainer: Achim Zeileis
Repository: http://cran.r-project.org
Packages:   bayesm, betareg, car*, Design,
            dse, dynlm, Ecdat, fCalendar,
            Hmisc, ineq, its, lmtest*, Matrix,
            micEcon, MNP, nlme, quantreg,
            sandwich*, segmented, sem,
            SparseM, strucchange, systemfit,
            tseries*, urca*, uroot, VR,
            zicounts, zoo*
            (* = core package)
```

Note that currently each CRAN task view is associated with a single CRAN-style repository (i.e., a repository which has in particular a `src/contrib` structure), future versions of **ctv** should relax this and make it possible to include packages from various repositories into a view, but this is not implemented, yet.

A particular view can be installed subsequently by either passing its name or the corresponding `"ctv"` object to `install.views()`:

```
R> install.views("Econometrics",
+                lib = "/path/to/foo")
R> install.views(x[[1]],
+                lib = "/path/to/foo")
```

An overview of these client-side tools is given on the manual page of these functions.

Writing a CRAN task is also very easy: all information can be provided in a single XML-based format called `.ctv`. The `.ctv` file specifies the name, topic and maintainer of the view, has an information section (essentially in almost plain HTML), a list of the associated packages and further links. For examples see the currently available views in **ctv** and also the vignette contained in the package. All it takes for a maintainer to write a new task view is to write this `.ctv` file, the rest is generated automatically when the view is submitted to us. Currently, there are task views available for econometrics, finance, machine learning and graphical models in R—furthermore, task views for spatial statistic and statistics in the social sciences are under development. But to make these tools more useful, task views for other topics are needed: suggestions for new task views are more than welcome and should be e-mailed to me. Of course, other general comments about the package **ctv** are also appreciated.

Achim Zeileis
Wirtschaftsuniversität Wien, Austria
Achim.Zeileis@R-project.org

Using Control Structures with Sweave

Damian Betebenner

Sweave is a tool loaded by default with the `utils` package that permits the integration of R/S with LaTeX. In one of its more prominant applications, Sweave enables literate statistical practice—where R/S source code is interwoven with corresponding LaTeX formatted documentation (R Development Core Team, 2005; Leisch, 2002a,b). A particularly elegant implementation of this is the `vignette()` function (Leisch, 2003). Another, more pedestrian, use of Sweave, which is the focus of this article, is the batch processing of reports whose contents, including figures and variable values, are dynamic in nature. Dynamic forms are common on the web where a base `.html` template is populated with user specific data drawn, most often, from a database. The incorporation of repetitive and conditional control structures into the processed files allows for almost limitless possibilities to weave together output from R/S within the confines of a LaTeX document and produce professional quality dynamic output.

Motivation for the current project came as a result of pilot data analyses for a state department of education. Dissemination of the analyses to schools in the state required the production of approximately 3,000 school reports documenting the results using school specific numbers/text, tables, and figures. The R/Sweave/LaTeX combination was deployed in various ways to produce prototype reports. This article provides details of the process so that others with similar document production needs can employ this combination of tools.

Outline of Process

The production process is quite basic in its implementation: Sweave a file using an appropriate dataframe to create a .tex that includes appropriate output. Most users of Sweave would have little problem creating such a .tex file based upon a given dataframe. Adding the capacity to deal with numerous different dataframes is possible using basic control structures available in R. In general, there are two files necessary for the mass production of reports:

`template.Rnw` A template file that provides the template into which all the relevant R output is placed.

`master.R` A master file that loops over all the recipients of the template file, creates the subset of data to be used within the template file, Sweaves the template and saves the output to a recipient specific .tex file.

In the following, the essential parts of each of these files is discussed in detail:

master.R

The `master.R` file implements a control structure *outside* of the Sweave process. This allows one to create a .tex file for each recipient. The key to the creation of these files is the creation of a distinct `outfile` for each unique recipient in the list. A basic `master.R` file which performs this looks like:

```
load(the.data.frame)
recip.list <- unique(the.data.frame$recip)
for (name in recip.list){
  subset.data <- subset(the.data.frame,
             the.data.frame$recip==name)
  outfile <- paste(name, "tex", sep=".")
  Sweave("template.Rnw", output=outfile)
}
```

The process starts by sourcing `master.R`. This loads a master data frame that contains the data for all the potential report recipients. A list of recipients is then derived from that dataframe. A looping structure is implemented that subsets the appropriate data file from the master data frame. Crucially, this subsetted data frame contains the recipient specific data that is used to construct the individualized report. Finally, a recipient specific .tex file is created by by Sweaving the template, `template.Rnw`.

Depending upon the number of recipients, it is entirely possible for thousands of .tex files to be generated. To avoid the files accumulating in the same directory, one can easily modify the code to place the output in other directories. For example, if one wished to create a distinct directory for each recipient, the following additions would suffice:

```
for (name in recip.list){
  subset.data <- subset(the.data.frame,
             the.data.frame$recip==name)
  outfile <- paste(name, "tex", sep=".")
  dir.create(name)
  setwd(name)
  Sweave(file.path("..", "template.Rnw"),
                   output=outfile)
  setwd("..")
}
```

This basic control structure can be tweaked in many ways. For example, a call can be made to a relational database that pulls the relevant data into R for processing. This is particularly attractive for those applications where data storage is cumbersome and better administered with a relational database.

template.Rnw

The template, `template.Rnw`, is a standard Sweave file containing LaTeX markup together with documentation and code chunks. The file, as its name suggests, is a template into which recipient specific numbers/text, tables, and figures can be included. One of the greatest challenges to producing a template is ensuring that it will produce the expected output given the range of possible values it is expected to accept.

One of the easiest ways of providing recipient specific output with the `template.Rnw` file is through the liberal use of the `\Sexpr` function. For the aforementioned school reports, the name of the school was included throughout the text, tables, and figures to provide a customized feel. In particular, the fancyhdr LaTeX package was used and the school name was included in the left header. Because school names, and string expressions in general, vary widely in their length, care must be exercised so that any place that the `\Sexpr` is utilized, the string (or number) resulting from `\Sexpr` will function in the LaTeX document for the range of values of the field that occur in the data set.

The creation of recipient specific tables is a frequently encountered (particularly with .html) means of dynamic report generation. There are a number of ways of including tables into the template.Rnw document. Perhaps the simplest way is through the use of the xtable function within David Dahl's xtable package which produces LaTeX output for a variety of R objects. A basic code chunk yielding .tex markup producing a table is given by

```
<<echo=FALSE,results=tex>>=
xtable(object, caption="table:caption",
       label="table:label")
@
```

Though suitable for a variety of tasks, the function xtable is limited in its capacity to produce complex tables (e.g., tables with multicolumn headers). Another frequently encountered difficulty occurs when tables being produced are long and possibly extend over several pages. Such long tables are not a problem with |.html| because the pages can be of nearly infinite length. However, printed pages must have the table broken into pieces according to the space available on the page. The LaTeX code produced by xtable will not yield acceptable results. To circumvent this, one can use the latex.table function provided by Roger Koenker's quantreg package together with the option for using the longtable LaTeX package. The latex.table function accepts matrix objects and outputs results to a specified file. As such, its incorporation into the Sweave document requires a slightly different two part technique than that used with xtable:

1. latex.table is called within a code chunk and produces the required LaTeX markup and saves it to an output file.

   ```
   <<echo=False>>=
   latex.table(matrix,
           file=paste(name, ":table.tex",
                      sep=""))
   @
   ```

2. The LaTeX markup for the table is inserted into the document (n.b., outside of a code chunk) at the appropriate place:

   ```
   \begin{center}
   \input{\Sexpr{name}:table.tex}
   \end{center}
   ```

As with xtable, there are still limitations to the amount of tabular complexity available with latex.table. Multicolumn headings, multirow headings, colored columns, etc. are all difficult if not impossible to produce. Depending upon ones willingness to program, however, all effects can ultimately be produced. Though tedious, frequent use of Sexpr commands outside of code chunks or cat commands within code chunks allows all LaTeX commands to be incorporated into the final .tex document upon Sweave compilation.

In many instances, particularly with tables and figures, it's necessary to produce multiple tables or figures within a single Sweave document based upon particular attributes of the recipient. For example, with the school project each school's report contained information pertinent to each grade in the school. Thus, a table was produced for each grade within the school. To implement this, a loop is placed *within* template.Rnw. This procedure, given in FAQ A.8 from Friedrich Leisch's Sweave User Manual (Leisch, 2005), is useful is many situations.

A major difficulty with control structures within the template is the variable nature of the output. In the example with school reports, some rural schools have grades 1 through 12. Thus, if a table or figure is produced for each grade, one must take account of the fact that so many tables or figures will result in the .tex document. The simplest way to accommodate different numbers of loop iterations within the template is to create a \newpage each time the loop completes. Thus, for example, each figure for each grade receives a page to itself. This doesn't always look acceptable for especially parsimonious tables or figures. However, combining multiple tables on a single page can often cause overfull hbox and vbox situations to occur. A workaround to this difficulty is to use the LaTeX \scalebox and \resizebox commands. These commands allow great flexibility in shrinking output to fit a given page. In general, a good deal of planning and experimentation is necessary to produce a template that works well for all situations.

The extent to which one wishes to make the reports specific to a particular recipient can be extended far beyond expressing the recipient specific information in tables and figures. For reports where the recipient is an individual, text within the report can be made gender specific through the use of if else statements based upon values in the subsetted data frame that the template is using to construct the recipient specific report. Supposing the gender code in the subsetted data frame is 1 for males and 2 for females, one could incorporate he/she (or him/her) pronouns into the text using a simple if else statement such as

```
\Sexpr{if (data.frame$gender==1) "he"
         else "she"}
```

The amount of customization is limited only by the amount of work put into constructing the template.

One is not limited to the basic LaTeX classes (e.g., article.cls or report.cls) when producing the template. The beamer.cls (Tantau, 2004) is Sweave compatible and provides extensive functionality to produce rich, full color .pdf output with extensive hyperlink capabilities. Figure 1 presents a two page example demonstrating some possibilities of the R/LaTeX/Sweave combination for dynamic .pdf report generation using the beamer.cls. Other

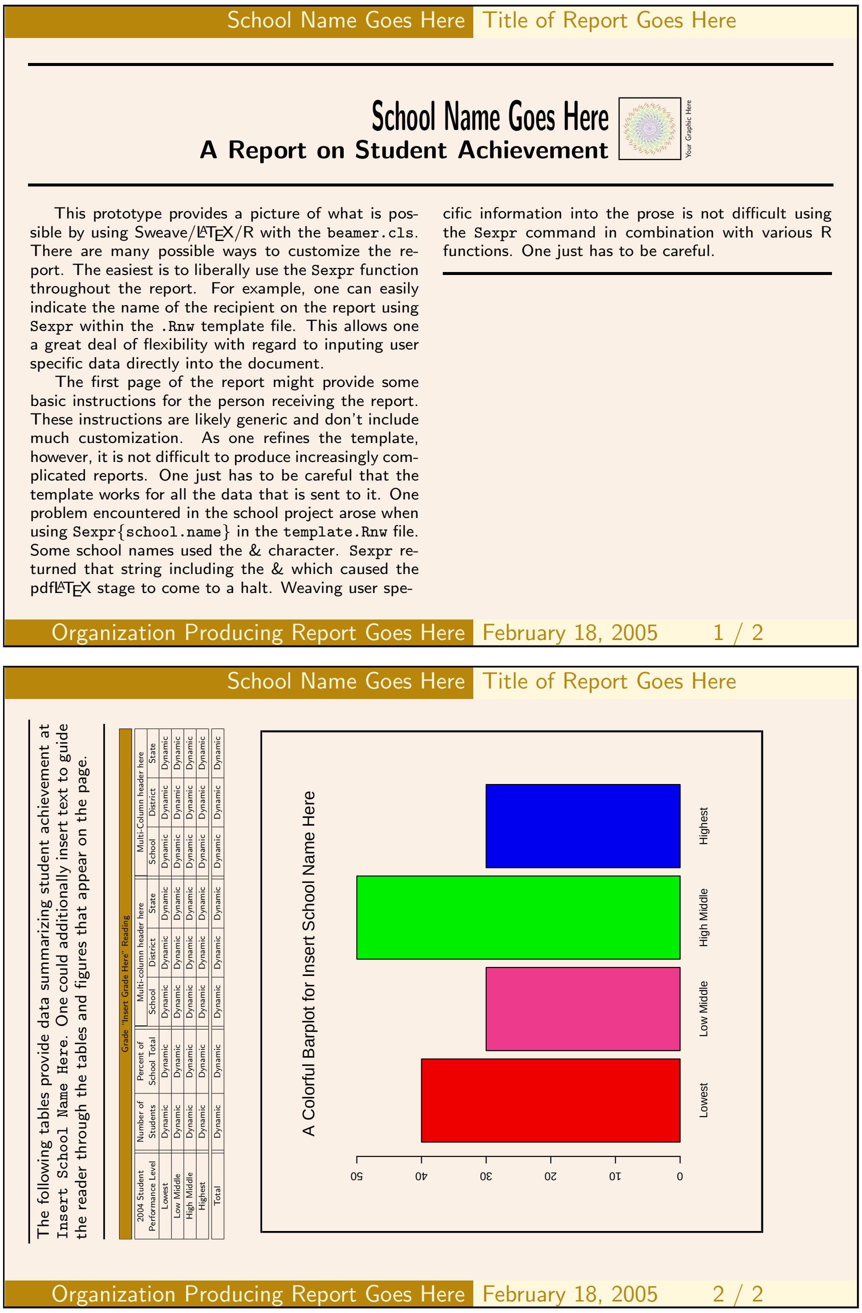

Figure 1: A basic two-page example demonstrating dynamic `.pdf` using the R/LaTeX/Sweave combination.

LaTeX classes, packages and font families are available that can be combined to yield unlimited possibilities.

Final Compilation

Upon Sweave completion, `master.R` yields numerous `.tex` files that need to be compiled. For the school project, approximately 3,000 `.tex` files were created. In addition, there were, in most cases, three `.pdf` figures created for each school. All of these files can be compiled from within R using the `texi2dvi` command available in the `tools` package. The following loop which can be placed into `master.R` accomplishes this task:

```
for (name in recip.list){
     outfile <- paste(name, "tex", sep=".")
     texi2dvi(outfile, pdf=TRUE, clean=TRUE)
}
```

`texi2dvi` will run LaTeX and BibTeX the appropriate number of times to resolve any reference and citation dependencies. The result is compiled versions of all the `.tex` files generated from the earlier loop containing the Sweave process. It is also possible to perform this task outside of R using a script or batch file to perform the same operations.

Conclusions

For batch production of dynamic .pdf or .ps documentation, the R/Sweave/LaTeX combination is powerful and nearly limitless in its capabilities. A major benefit to producing reports using this combination of tools is how closely Sweave integrates the processing power of R with the typesetting capability of LaTeX. The key to producing dynamic reports for a large number of recipients is the use of iterative control structures in R. This article provides the author's "homebrew" code. Other, more elegant, solutions are likely possible.

A next step after report generation is report distribution. In theory, given the appropriate server configuration, it should be possible to electronically distribute the reports to the appropriate recipient based upon, for example, an email address contained in the database. I would appreciate learning how others have addressed this and similar problems.

Acknowledgments

I would like to thank Bill Oliver for introducing me to R and for his gracious help. In addition, I would like to thank one of the reviewers for providing particularly insightful improvements to my code. Finally, I would like to express gratitude to the developers/contributors of R, Sweave, and LaTeX, without their efforts none of this would be possible.

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, Heidelberg, Germany, 2002a. Physika Verlag. ISBN 3-7908-1517-9. URL http://www.ci.tuwien.ac.at/~leisch/Sweave. 40

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/Rnews_2002-3.pdf. 40

F. Leisch. Sweave, Part II: Package Vignettes. *R News*, 3(2):21–24, October 2003. URL http://cran.r-project.org/doc/Rnews/Rnews_2003-2.pdf. 40

F. Leisch. *Sweave User Manual*, 2005. 42

R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL http://www.R-project.org. 3-900051-07-0. 40

T. Tantau. *User's Guide to the Beamer Class*. Sourceforge.net, 2004. URL http://latex-beamer.sourceforge.net. 42

Damian Betebenner
Lynch School of Education
Educational Research, Measurement and Evaluation
Boston College
Damian.Betebenner@bc.edu

The Value of R for Preclinical Statisticians

Bill Pikounis and Andy Liaw
Merck & Co., Inc.

Participation on the R-help mailing list has shown R to be widely used across organizations of all types the world over. This article discusses one corner where R has become an indispensable tool for effective statistical practice: a preclinical pharmaceutical environment at the authors' place of employment.

Preclinical is defined here as a portion of the research and development process for prescription medicines. In this article that portion is defined to start with fundamental discovery, and to end up where a potential product is first put into human subjects in clinical trials. There is a wide variety of knowledge sought across this spectrum, including: biological targets believed to be important influences on a given disease; chemical entities that affect such targets; effectiveness and safety in animals; formulation of product itself. This brief listing is only intended to provide a broad overview. In reality, there are many more areas that could be included based on desired granularity. A complementary term of "nonclinical," which is also commonly used, is perhaps more appropriate here; however, we will stay with the phrase preclinical for brevity.

The hallmark diversity of preclinical research presents unlimited opportunities for statisticians. Data are collected everywhere in the process, and in large quantities. We have the liberty to choose methods of processing and analysis. This is a different environment than clinical biostatistics, where SAS is the dominant tool for various reasons. Below we present examples and discussion on how and why we use R and find it invaluable.