

# A primer on using Sweave/odfWeave

PKD

June 21, 2008

## Abstract

Sweave [1] allows one to mix  $\LaTeX$  with R. odfWeave [?] allows one to mix Open Document Format office documents  $\LaTeX$  with R.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	What are Sweave and odfWeave? . . . . .	1
1.2	Overview . . . . .	1
<b>2</b>	<b>Including scalar output and small output snippets</b>	<b>2</b>
<b>3</b>	<b>Including larger output from R</b>	<b>2</b>
3.1	General comments . . . . .	2
3.2	Including the R commands in the final document . . . . .	3
3.3	Including the R output in the final document . . . . .	3
3.4	Including R figures in the final document . . . . .	4
3.5	Including R tables in the final document . . . . .	5
<b>4</b>	<b>Other features</b>	<b>7</b>
4.1	Setting global options . . . . .	7
4.2	Extracting the R code . . . . .	7
4.3	Placing graphics files in separate locations . . . . .	7
4.4	Using global R figure parameter options . . . . .	7
4.5	Re-using code . . . . .	8
<b>5</b>	<b>Why use Sweave?</b>	<b>8</b>
<b>6</b>	<b>Obtaining more information</b>	<b>9</b>
<b>7</b>	<b>Citing Sweave</b>	<b>9</b>

# 1 Introduction

## 1.1 What are Sweave and odfWeave?

Sweave is a framework for mixing R computational code with  $\LaTeX$  documentation code in the one place. One file contains both the  $\LaTeX$  and R information.

## 1.2 Overview

Use Sweave like this:

- Create a  $\LaTeX$  document as usual, but give the file a `.Snw` extension rather than a `.tex` extension. For example, we may create a file called `analysis.Snw`
- In the places where output from R is required (such as figures, tables, the results of calculations), place the appropriate R code. Obviously, these commands must somehow be flagged as R commands to distinguish from the surrounding  $\LaTeX$  text (the two ways to do so are discussed in Sects. 2 and 3).
- Within R (ensuring R is operating in the appropriate folder/directory), evaluate the R code contained in the `Snw` document by typing (in R):

```
Sweave("analysis.Snw")
```

This produces a file called `analysis.tex`, which leaves all the  $\LaTeX$  in the file untouched, but replaces the R code by the corresponding R output.

- Run  $\LaTeX$  on this resulting file `analysis.tex` as normal. For example, `pdflatex analysis.tex` will produce the PDF file `analysis.pdf`

## 2 Including scalar output and small output snippets

To include small snippets of output, typically scalars, use the construct `\Sexpr{}` in the  $\LaTeX$  document, where correct R code is placed between the braces.

For example, consider the following code appearing in the `.Snw` file:

```
The data set we are using has
\Sexpr{length( len )}~observations,
comparing \Sexpr{num.supplements}~different supplements.
```

In the corresponding `.tex` file appears

```
The data set we are using has
60~observations,
comparing 3~different supplements.
```

## 3 Including larger output from R

### 3.1 General comments

In general, show larger amounts of output using the following structure:

```
...some \LaTeX\ code.
<<>>=                                (OPTIONS are given inside the << >>)
                                        (<--- R code goes in here)
@                                       (The @ signals the end of the R code)
Further \LaTeX\ code here...
```

The various options enclosed in << >> allow different output to appear in the final L<sup>A</sup>T<sub>E</sub>X document. Various options are combined, separated by commas:

```
<<echo=FALSE, results=hide, fig=TRUE>>=
hist ( rnorm(100) )
@
```

These options are explained below, but briefly:

- The R commands are not shown in the final document (`echo=FALSE`)
- The output produced by R at the command line is not shown (`results=hide`)
- The figure produced by R appears in the final document (`fig=TRUE`)

Incidentally, the order of the options does not matter; this chunk does exactly the same thing:

```
<<fig=TRUE, echo=FALSE, results=hide>>
hist ( rnorm(100) )
@
```

Importantly, the << >>= and @ constructs *must appear at the very start of a line* (that is, they cannot be indented).

### 3.2 Including the R commands in the final document

Sometimes, showing the R commands in the final document is useful (for example, to show colleagues how the data is analysed; in teaching materials to teach the analysis in R; in a document teaching R itself). Include the following in the `.Snw` document:

```
<<echo=TRUE>>=
R code in here
@
```

The default is *to show* the R code (`echo=TRUE`). The options are

- `echo=TRUE` to show the R code (the default);

- `echo=FALSE` to hide the R code.

To keep the formatting of the R code (such as indentations, comments, etc.) use the `<<keep.source=TRUE>>` option. The default is `keep.source=FALSE` when the formatting is lost. (This is never an issue with option `echo=FALSE` since the R code is never shown anyway.)

### 3.3 Including the R output in the final document

Sometimes, you would prefer not to show the R output in the final document. Include the following in the `.Snw` document:

```
<<results=hide>>=  
R code in here  
@
```

The default is *to show* show the R output (`results=tex`). The options are

- `results=verbatim` to show the R output (the default);
- `results=hide` to hide the R output (but the R code is still executed);
- `results=tex` indicates that the output should be interpreted as  $\LaTeX$  instructions (see Sect. 3.5 for a powerful example).

### 3.4 Including R figures in the final document

Figures are often included in final documents. Include the following in the `.Snw` document:

```
<<fig=TRUE>>=  
R code in here that produces a plot  
@
```

This construct does the following:

- The R code produces the plot using the given R commands
- The picture file is saved (in both pdf and eps formats by default)
- The text between (and including) the `<< >>=` and `@` is replaced by a `\includegraphics` command which loads the file saved by Sweave.

As an example, consider the following code in the `.Snw` file:

In the Sweave file:

```
\begin{figure}
  \centering
  <<fig=TRUE>>=
  x <- seq(-2, 2, length=100)
  y <- x^2
  plot( y ~ x, type="l")
  @
  \caption{A quadratic}
  \label{FG:quadratic}
\end{figure}
```

In the resulting L<sup>A</sup>T<sub>E</sub>X file:

```
\begin{figure}
  \centering
  \includegraphics{example-001.eps}
  \caption{A quadratic}
  \label{FG:quadratic}
\end{figure}
```

The picture is created in the files `example-001.eps` `example-001.pdf` automatically by R.

- `fig=TRUE` informs R to perform the above steps to include a figure in the L<sup>A</sup>T<sub>E</sub>X document;
- `fig=FALSE` informs R not to perform the above steps; no figure will be produced (the default).

### 3.5 Including R tables in the final document

Commonly, tables appear in documents. There are two main ways to do so.

**Using `Sexpr`** A table of information can be expressed by repeatedly using `Sexpr{}` in the `.Snw` file as follows:

```
\begin{table}
  \centering
  \caption{Mean and standard deviation of ages for
           males and females in the sample}
  \label
  \begin{tabular}{cc}
    Statistic & Males & Females\\
    \hline
    Mean      & \Sexpr{signif( mean(Ages[Gender=="Male"]), 3)}
              & \Sexpr{signif( mean(Ages[Gender=="Female"]), 3)}\\
    Std dev   & \Sexpr{signif( sd(Ages[Gender=="Male"]), 3)}
              & \Sexpr{signif( sd(Ages[Gender=="Female"]), 3)}
  \end{tabular}
\end{table}
```

**Creating the table automatically** Two R functions exist that automatically create tables from R objects: `xtable` in package `xtable`, and `latex` in package `HMisc` I have used `xtable` but not `latex` so can only comment on `xtable`

Create a table automatically in the `.tex` document by placing a command like this in the `.Snw` document:

```
<<echo=FALSE,results=tex>>=
  xtable( sum.info, digits=c(1,3,3) )
@
```

This code takes the data frame `sum.info` and formats it as a  $\LaTeX$  table ready for direct inclusion into the  $\LaTeX$  document.

See an example to understand how it works.

```
> library(xtable)
> Ages <- c(23, 34, 26, 29, 20, 30, 24, 35, 26, 28)
> Gender <- c( rep("Male", 4), rep("Female",6) )
> Ages
[1] 23 34 26 29 20 30 24 35 26 28
> Gender
[1] "Male" "Male" "Male" "Male" "Female" "Female" "Female" "Female"
[9] "Female" "Female"
> tapply(Ages,Gender,mean)
Female Male
27.16667 28.00000
> tapply(Ages,Gender,sd)
Female Male
5.154286 4.690416
> sum.info <- rbind( tapply( Ages, Gender, mean ), tapply(Ages, Gender, sd) )
> rownames(sum.info) <- c("Mean", "Std dev")
> sum.info
      Female      Male
Mean  27.16667 28.00000
Std dev 5.154286 4.690416
> xtable(sum.info, digits=c(0,3,3) )
% latex table generated in R 2.6.0 by xtable 1.5-1 package
% Mon Nov 19 11:33:56 2007
\begin{table}[ht]
\begin{center}
\begin{tabular}{rrr}
\hline
& Female & Male \\
\hline
Mean & 27.167 & 28.000 \\
Std dev & 5.154 & 4.690 \\
\hline
\end{tabular}
\end{center}
\end{table}
```

The output of the `xtable` command is a table all prepared for direct inclusion in a  $\LaTeX$  document. Magic!

## 4 Other features

### 4.1 Setting global options

Suppose you never wish to show the R code in your final document. You need to state `<<echo=FALSE, results=hide>>=` for every chunk of R code. Instead, place the following somewhere in your  $\LaTeX$  document:

```
\SweaveOpts{echo=FALSE, results=hide}
```

This command sets the options for all subsequent code chunks. If it is placed in the preamble, it applies for *all* code chunks in the document.

### 4.2 Extracting the R code

After creating your `Snw` file, all the R commands that appear in `<< >>=... @` constructs (but not those appearing in `\Sexpr{...}` constructs) can be extracted by typing (in R):

```
Stangle("analysis.Snw")
```

This produces a file `analysis.R` in the current directory/folder which can be run in R in the usual manner (such as `source("analysis.R")`). This can be useful for locating errors in the R code.

### 4.3 Placing graphics files in separate locations

If your document contains many figures, you may wish to save them in a separate directory/folder. Use the `SweaveOpts` command (Sect. 4.1) as follows:

```
SweaveOpts{prefix.string=section1/sect1}
```

Any graphics files created will be placed in a directory/folder called `section1` and the file names will start with `sect1`; that is, the figures will be called `sect1-001.eps` and placed in directory/folder `section1` for example.

### 4.4 Using global R figure parameter options

If you wish to set graphics parameters throughout your document, use, for example,

```
options(SweaveHooks=list(fig=function() par(las=2, fg="wheat")))
```

All figure will now execute these R `par` commands before the figure is created.

## 4.5 Re-using code

You can re-use code if you wish by *naming* code chunks. Consider this code chunk:

```
<<example,eval=FALSE>>=  
hist( rnorm(10) )  
hist( rnorm(100) )  
hist( rnorm(1000) )  
hist( rnorm(10000) )  
@
```

This code would produce four histograms of 10, 100, 1000 and 10 000 random numbers from a normal distribution. But the code is not evaluated (because of the option `eval=FALSE`), but it *is* shown in the output document (because the option `echo=TRUE` is the default). The very first option in the `<< >>=` is the *name* of the chunk. The chunk name *must* be listed first inside the `<< >>=`. To reuse this chunk, refer to it by this name. Later in the document we may have:

```
<<fig=TRUE, echo=FALSE>>=  
par( mfrow=c(2,2))  
<<example>>  
@
```

The first line of code (that is, `par( mfrow=c(2,2) )`) creates a two-by-two grid on which to place the four plots (otherwise, Sweave will only show the last plot in the final document). Then, the chunk named `example` is run, and therefore the plots produced. This time, plots are produced (since `fig=TRUE`) but the code itself is not shown (since `echo=FALSE`).

## 5 Why use Sweave?

**For data analysis** When analysing data for clients, one can write a report and implement the comments immediately in the analysis. (For example, you may note a transformation of the data is required; you can follow this comment with the R commands to do exactly that.) The comments and instructions are kept together.

**For research** It allows reproducible research: keeping track of what was done is easy by having the R code commented by the  $\LaTeX$  surrounding it.

**For teaching** In teaching, it allows one to teach R by showing exactly what to do and what happens by actually embedding the actual R code into the document, and the exact output produced appears in the final product.

**For convenience** If anything is updated—the data file, the R code you have used to analyse the data—you only need to re-run `Sweave` and  $\LaTeX$  to re-generate a new version of the document.



**For accuracy** You know the results of computations are reported accurately with no typos, since the results are computed and placed in the document without human intervention. (Of course, whether those computations are appropriate is another matter entirely.)

## 6 Obtaining more information

More information can be found as follows:

- In R, type `vignette(package="Sweave")`
- Much information, including publication by the author of Sweave, is located at <http://www.stat.uni-muenchen.de/~leisch/Sweave>

## 7 Citing Sweave

To see how to cite Sweave, use the usual R commands:

```
citation("Sweave")
```

## References

- [1] Friedrich Leisch (2002). Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis, *Compstat 2002 — Proceedings in Computational Statistics*, 575–580, Wolfgang Haerdle and Bernd Roenz, Physica Verlag, Heidelberg, ISBN 3-7908-1517-9.