Latex and R via Sweave
An example document how to use Sweave

Peter Filzmoser
Vienna University of Technology, Austria
P.Filzmoser@tuwien.ac.at

March 12, 2008

Contents

1 General information on Sweave **1**
1.1 Sweave sources 2

2 Producing a document **2**
2.1 How it works 2
2.2 Executing the files 2

3 Examples **2**
3.1 Showing and hiding computations and results 3
3.2 Generating and including figures 4
3.3 Including tables 6

1 General information on Sweave

Sweave combines both LaTeX and R code in one document. Via R the R commands contained in the document are executed, and the computations are performed. Also figures can be produced and they can be named automatically. The result is a LaTeX file and eventually files (.pdf or .eps, or both) for the figures. Using *pdflatex* the LaTeX file is executed and the figures are automatically included. Thus, the user of Sweave has the advantage that the code for computation and figures is 100% reproducable. Moreover, it is convenient to get the figures automatically included in the final document. In that way, dynamic reports can be generated.

1.1 Sweave sources

Sweave is part of every R installation, and thus nothing additional has to be installed. The program was written by Fritz Leisch, and the key reference is

> Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 - Proceedings in Computational Statistics*, pages 575-580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

Details on Sweave, as well as manuals and examples can be found at:
`http://www.statistik.lmu.de/~leisch/Sweave/`

2 Producing a document

2.1 How it works

The document should have the extension *.Snw*, or alternatively *.Rnw*, in order to avoid any problems later on. You can start with the ususal LaTeX definitions and commands. Additionally, options for Sweave can be defined, but this is not necessary.

In the document, R commands start with `<<>>=` and end with `@`. In between there can be pages and pages of R code (but not LaTeX code). Between the brackets `<<>>` there can be options for Sweave. Important options are `<<fig=TRUE>>` for including figures or `<<echo=TRUE>>` for showing R code in the document.

2.2 Executing the files

It is clear that both LaTeX and R have to be installed on your computer to successfuly use this tool. Once the Sweave file has been written, it can be processed in R by typing into a running R session

```
Sweave("filename.Snw")
```

where `filename.Snw` is the Sweave file. If R is run in a different directory, one has to provide the full path name. This generates the file `filename.tex` which can be processed as usual with LaTeX.

3 Examples

In the following we want to demonstrate briefly how Sweave can be used. We show specific examples for including R code, results, figures, and tables.

3.1 Showing and hiding computations and results

For example, we can either show R code by typing into the document

```
<<echo=TRUE>>=
x <- rnorm(100)
xm <- mean(x)
xm
@
```

which appears in the final document as

```
> x <- rnorm(100)
> xm <- mean(x)
> xm

[1] 0.0725661
```

If, for any reason, results should not be shown in the final document but only be computed within R, we can use

```
<<echo=TRUE,results=hide>>=
x <- rnorm(100)
xm <- mean(x)
xm
@
```

which finally gives

```
> x <- rnorm(100)
> mean(x)
> xm <- mean(x)
> xm
```

without the result. If the computation should be done in R but only the result should be visible in the final document we need

```
<<echo=FALSE>>=
x <- rnorm(100)
xm <- mean(x)
xm
@
```

which indeed gives only the result

```
[1] -0.1822784
```

for the mean of out 100 standard normally distributed data points.

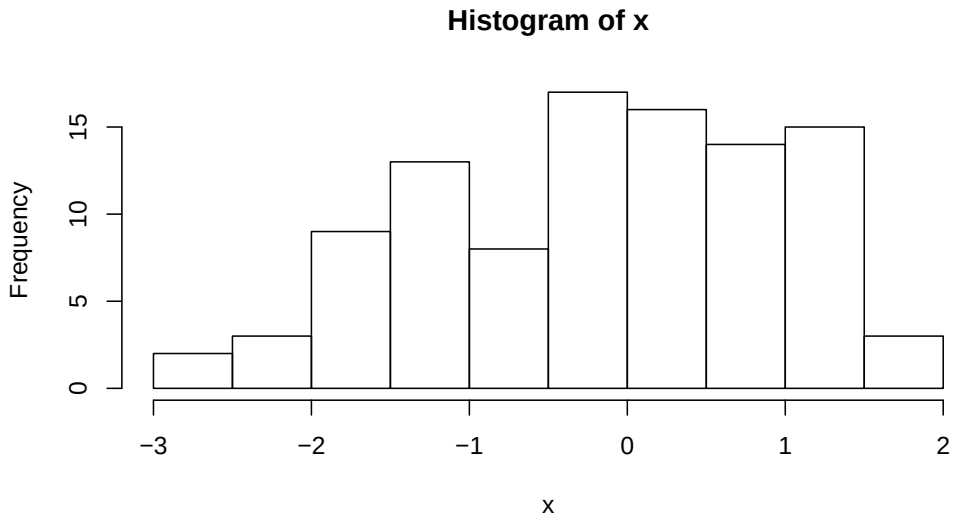
3.2 Generating and including figures

The data generated above are still available in the R object `x`. Here we want to make a histogram and include it in the final document. The simplest possibility for this purpose is by using the commands

```
\setkeys{Gin}{width=1.0\textwidth}
<<echo=TRUE,fig=TRUE,width=7,height=4>>=
hist(x)
@
```

in the document. This results in

```
> hist(x)
```



for the final document. The command `\setkeys{Gin}{width=1.0\textwidth}` is outside of the R commands and thus executed in LaTeX. This can also be defined at the beginning of the document, and it defines the scaling of the figures. Here we want to have them scaled such that the width corresponds to the textwidth, and thus the options `width=7,height=4` define only the ratio between width and height.

We can do this more sophisticated by defining a the appropriate figure environment in LaTeX. Let us first define

```
> x = runif(100, 1, 10)
> y = 2 + 3 * x + rnorm(100)
```

and compute the regression line by

```
> lm.xy = lm(y ~ x)
```

The plot of x versus y, together with the regression line is generated by

```
> plot(x, y)
> abline(lm.xy)
```

and is now shown in Figure 1.

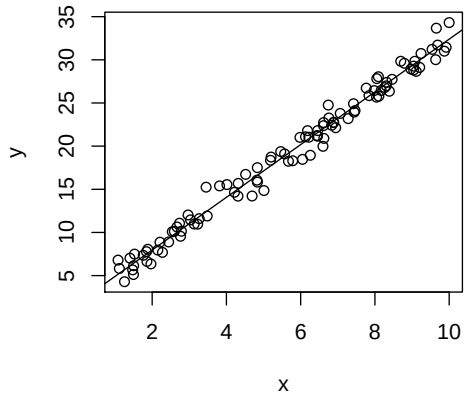


Figure 1: Regression of y on x

The complete code (LaTeX and R) is

```
\begin{figure}[htbp]
\begin{center}
\setkeys{Gin}{width=0.5\textwidth}
<<echo=FALSE,fig=TRUE,width=4,height=4>>=
plot(x,y)
abline(lm.xy)
@
\caption{Regression of $y$ on $x$}
\label{fig:regression}
\end{center}
\end{figure}
```

Here the R code has not been shown in the final document because `echo=FALSE` has been used.

3.3 Including tables

There is a nice R feature available in the *library(xtable)* which generated LaTeX code from screen output in R in form of a table. We will demonstrate this for the above regression model. The code for printing the summary statistics is obtained by

```
> library(xtable)
> xtable(summary(lm(y ~ x)))
```

The table can be included in LaTeX. However, arguments for label and captions have to be defined in *xtable* and not in the usual table environment of LaTeX. We obtain Table 1.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.8520	0.2385	7.76	0.0000
x	3.0543	0.0381	80.06	0.0000

Table 1: Summary statistics for the regression model

The complete code (LaTeX and R) is

```
\begin{center}
<<echo=FALSE,results=tex>>=
library(xtable)
xtable(summary(lm(y~x)),
caption="Summary statistics for the regression model",
label="tab:summary")
@
\end{center}
```

Now you can change the definition of x and y in the regression model, and everything changes automatically. If you additionally set random seeds you will have completely reproducable results.