Sweave

Marlena Maziarz
July 5, 2012

A Tale of Two Students

John, Without Sweave

1. Do your analysis in R
2. Write a report in word
3. Insert your tables and plots

The investigator wants to add another covariate into the main model
- or -
the dataset was updated
- or -
various other scenarios that require redoing of the analysis

4. Repeat 1-3 (depending on the project, it may be surprisingly many times)

Jane, With Sweave

1. Do your analysis and writeup in Sweave

2. Repeat 1

Huge time savings in the end, not to mention reproducibility!

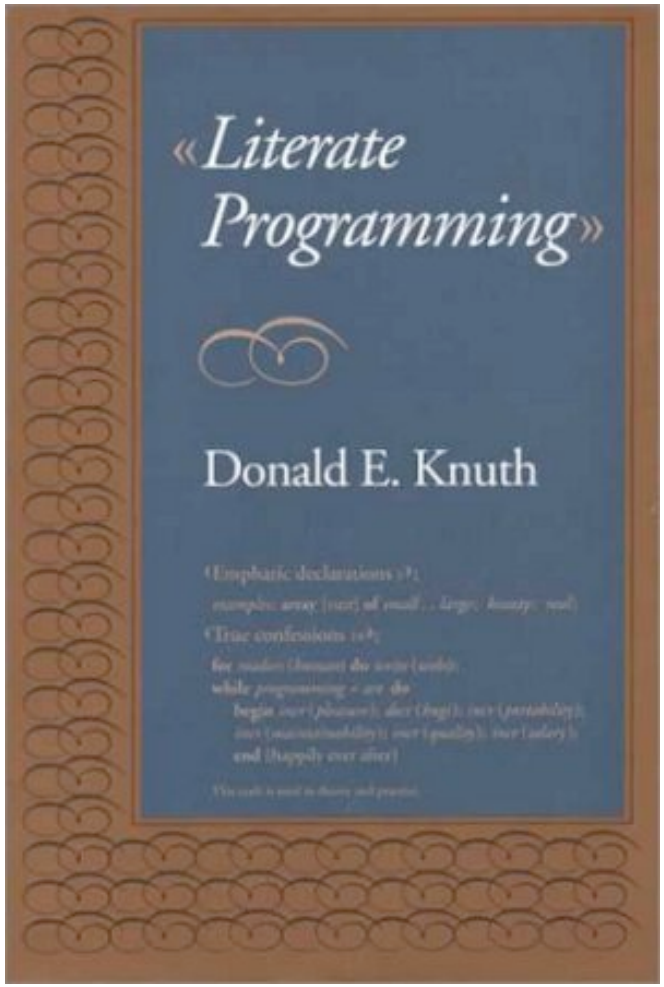
What is sweave?

- A framework for mixing text and R code for automatic generation of dynamic reports.
- Allows to regenerate a report if input data changes (as it often does).
- Very little new syntax to learn, given you know latex and R.
- Great for keeping track of your work and ensuring its reproducibility.

How does Sweave work?
Noweb and literate programming

- Noweb
  - a simple, extensible tool for literate programming
- Literate programming
  - an approach to programming introduced by Donald Knuth as an alternative to the structured programming paradigm of the 1970s
  - is the art of preparing programs for human readers

- In short - a noweb file is a simple text file which consists of a sequence of code and documentation segments, called chunks



Documentation and code chunks

Sweave-example.rnw

Compile PDF

```
\documentclass[a4paper]{article}

\title{Sweave}
\author{Marlena Maziarz}

\usepackage{Sweave}

\begin{document}
\maketitle


This is an example of a simple Sweave document.
You can write $T_EX$t as usual in an Sweave document. A code chunk sta
at the beginning of a line. Even a space in front of it will cause an
chunk:
<<label=code-chunk-1, echo = T>>=
set.seed(1)
x <- rnorm(100)
@
A code chunk ends with `@'. After that we can write text as usual. In
the mean and standard deviation of the numbers we generated in the pre
<<code-chunk-2, echo = F>>=
mean(x)
sd(x)
@
Oops!
<<code-chunk-2, echo = F>>=
round(c(mean(x), sd(x)), 2)
@
Much better.\\

Save this file as an "R noweb file" - 'Sweave-example.rnw'. To generat
PDF" in R studio. In R, it's a bit more tedious:
\begin{verbatim}
library(utils)
## This will generate a 'tex' document
Sweave('Sweave-example.rnw')
## Now compile it to PDF by
tools::texi2dvi(`Sweave-example.tex', pdf=TRUE)
## or outside R by
R CMD texi2dvi --pdf Sweave-example.tex
\end{verbatim}

That's all there's too it, Sweave, I mean.
\end{document}
```

Sweave

Marlena Maziarz

June 22, 2012

This is an example of a simple Sweave document. You can write T_EXt as usual in an Sweave document. A code chunk starts with <<>>=. It has to be at the beginning of a line. Even a space in front of it will cause an error. Here is our first code chunk:

```
> set.seed(1)
> x <- rnorm(100)
```

A code chunk ends with '@'. After that we can write text as usual. In the next code chunk calculate the mean and standard deviation of the numbers we generated in the previous chunk:

```
[1] 0.1088874
```

```
[1] 0.8981994
```

Oops!

```
[1] 0.11 0.90
```

Much better.

Save this file as an "R noweb file" - 'Sweave-example.rnw'. To generate a PDF, click on "Compile PDF" in R studio. In R, it's a bit more tedious:

```
library(utils)
## This will generate a 'tex' document
Sweave('Sweave-example.rnw')
## Now compile it to PDF by
tools::texi2dvi(`Sweave-example.tex', pdf=TRUE)
## or outside R by
R CMD texi2dvi --pdf Sweave-example.tex
```

That's all there's too it, Sweave, I mean.

Documentation and code chunks

Sweave-example.rnw

Compile PDF | Run | Source

```
1  \documentclass[a4paper]{article}
2
3  \title{Sweave}
4  \author{Marlena Maziarz}
5
6  \usepackage{Sweave}
7
8  \begin{document}
9  \make
10
11
12 This is an example of a simple Sweave document.
13 You can write $T_EX$t as usual in an Sweave document. A code chunk starts with $<<>>=$. It has to be
   at the beginning of a line. Even a space in front of it will cause an error. Here is our first code
   chunk:
14 <<label=code-chunk-1, echo = T>>=
15 set.seed(1)
16 x <- rnorm(100)
17 @
18 A code chunk ends with `@'. After that we can write text as usual. In the next code chunk calculate
   the mean and standard deviation of the numbers we generated in the previous chunk:
19 <<code-chunk-2, echo = F>>=
20 mean(x)
21 sd(x)
22 @
23 Oops!
24 <<code-chunk-2, echo = F>>=
25 round(c(mean(x), sd(x)), 2)
26 @
```

Text is written as usual, no special syntax needed.

Documentation and code chunks

Sweave-example.rnw

Compile PDF | Run | Source

```
1  \documentclass[a4paper]{article}
2
3  \title{Sweave}
4  \author{Marlena Maziarz}
5
6  \usepackage{Sweave}
7
8  \begin{document}
9  \maketitle
10
11
12 This is an example of a simple Sweave document.
13 You can write $T_EX$t as usual in an Sweave document. A code chunk starts with $<<>>=$. It has to be
   at the beginning of a line. Even a space in front                                          code
   chunk:
14 <<label=code-chunk-1, echo = T>>=
15 set.seed(1)
16 x <- rnorm(100)
17 @
18 A code chunk ends with "@". After that we can wri                                         ulate
   the mean and standard deviation of the numbers we
19 <<code-chunk-2, echo = F>>=
20 mean(x)
21 sd(x)
22 @
23 Oops!
24 <<code-chunk-2, echo = F>>=
25 round(c(mean(x), sd(x)), 2)
```

R code goes in between

<<>>=

R code

@

To generate a PDF

- Save this file as an "R noweb file" - 'Sweave-example.rnw'
- In R studio - click on "Compile PDF"
- In R, it's a bit more tedious:

```
library(utils) # Sweave function is in the utils library
Sweave('Sweave-example.rnw') # generates a tex document
tools::texi2dvi(`Sweave-example.tex', pdf=TRUE) # tex to pdf
```

- outside R

```
R CMD texi2dvi --pdf Sweave-example.tex
```

```
at the beginning of a line. Even a space in front of it will cause an error. Here is our first code
chunk:
<<label=code-chunk-1, echo = T>>=
set.seed(1)
x <- rnorm(100)
@
A code chunk ends with `@'. After that we can write text as usual. In the next code chunk calculate
the mean and standard deviation of the numbers we generated in the previous chunk:
<<code-chunk-2, echo = F>>=
mean(x)
sd(x)
@
Oops!
<<code-chunk-2, echo = F>>=
round(c(mean(x), sd(x)), 2)
@
Much better.\\
```

before

at the beginning of a line. Even a space in front of it will cause an error. Here is our first code chunk:

```
> set.seed(1)
> x <- rnorm(100)
```

A code chunk ends with '@'. After that we can write text as usual. In the next code chunk calculate the mean and standard deviation of the numbers we generated in the previous chunk:

```
[1] 0.1088874

[1] 0.8981994
```

Oops!

```
[1] 0.11 0.90
```

Much better.

after

Sweave options

- Global options
  - `\SweaveOpts{opt1=value1, ..., optN = valueN}`
  - Modifies the defaults for the rest of the document
- Local options
  - `<<label, opt1 = value1, ..., optN = valueN>>=`
  - Modifies the defauls only for this chunk
- Some of the most commonly used options are:
  - `echo, eval, results, fig, width, height`

Tables

Sweave code

```
<<results = tex, echo = F>>=
library(xtable)
col.names <- c('col 1', 'col 2')
row.names <- c('row 1', 'row 2', 'row 3')
out <- matrix(1:6, nrow = 3, byrow = T)
colnames(out) <- col.names
rownames(out) <- row.names
xtable(out, digits = 2, caption = 'My first Sweave
table!')
@
```

Tex code automatically generated by Sweave

```
\begin{table}[ht]
  \begin{center}
    \begin{tabular}{rrr}
      \hline
      & col 1 & col 2 \\
      \hline
      row 1 &   1 &   2 \\
      row 2 &   3 &   4 \\
      row 3 &   5 &   6 \\
      \hline
    \end{tabular}
    \caption{My first Sweave
    table!}
  \end{center}
\end{table}
```

Figures

Sweave code

```
\begin{figure}[!h]
\begin{center}
<<fig = T, width = 7, height = 7, echo = F>>=
  # whatever you want to plot goes here (one plot
  at a time!)
  par(mfrow = c(2,1))
  hist(x, 10, main = 'histogram')
  plot(x, y, main = 'scatter plot')
@
\end{center}
\caption{What you see in this figure...}
\end{figure}
```

Tex code automatically generated by Sweave

```
\begin{figure}[!h]
  \begin{center}
    \includegraphics{auto-generated-pdf-filename}
  \end{center}
  \caption{What you see in this figure...}
\end{figure}
```

Including R code in the text

- A quick way to include dynamic numbers (or any R code that evaluates to a number) in the text is to use \Sexpr{}.
- Anything enclosed in the curly braces will be replaced with a number in the tex file.
- For example, the mean of a hundred random N(0, 1) numbers is \Sexpr{mean(rnorm(100))}.
- And if you wanted to do anything more complicated, then:

```
<<echo=F>>=
set.seed(1)
x <- rnorm(100)
@

The mean of a hundred random N(0, 1) variables is \Sexpr{x}.
```

References

- Sweave FAQ by Friedrich Leisch

  http://www.statistik.lmu.de/~leisch/Sweave/FAQ.html

- Sweave User Manual by Friedrich Leisch

  www.stat.uni-muenchen.de/~leisch/Sweave/Sweave-manual.pdf

More advanced stuff

- Stangle
- Sweave hooks
- Caching code blocks (avoid re-running long simulations or analyses)
- Consistent formatting (matching fonts in graph labels to the font in the text)
- People are excited about knitr (http://yihui.name/knitr/)