Literate Programming and Reproducible Research

S J Eglen

28 October 2010

Outline

Literate Programming

From the web page describing his book *Literate Programming*, Donald E Knuth writes:
"Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language. The main idea is to treat a program as a piece of literature, addressed to human beings rather than to a computer. The program is also viewed as a hypertext document, rather like the World Wide Web. (Indeed, I used the word WEB for this purpose long before CERN grabbed it!) . . . "
`http://www-cs-faculty.stanford.edu/~uno/lp.html`

Tangling and Weaving:

- CWEB: system for documenting C, C++, Java:

```
CTANGLE
    converts a source file foo.w to a compilable program file :
CWEAVE
    converts a source file foo.w to a prettily-printable and
    cross-indexed document file foo.tex.
```

`http://sunburn.stanford.edu/~knuth/cweb.html`

What is Reproducible Research (RR)?

- Gentleman et al (2004) advocate RR:

  Buckheit and Donoho (35) , referring to the work and philosophy of Claerbout, state the following principle: "An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and that complete set of instructions that generated the figures."

`http://genomebiology.com/2004/5/10/R80`

- Bioconductor packages are good examples of reproducible research.
- This article is also good background reader for open software development.

Approaches to RR

1. Makefiles
2. Sweave

Make and Makefiles

- Make is an automatated build system, designed to avoid costly recomputation.
- *make* examines a **Makefile**, which contains a set of rules describing dependencies among files.
- A rule is run if the **target** is older than any of its **dependencies**.
- Older: compare creation time of files.
- Example:

```
res.txt: param1.dat param2.dat
        simulation param1.dat param2.dat > res1.dat
        post-process res1.dat > res.txt
```

- Commands to be run should be indented with a TAB.

A complete Makefile file:rr_make/

```
report.pdf: report.tex sim1.pdf sim2.pdf
        texi2pdf report.tex

sim1.dat: params.R simulator.R
        Rscript simulator.R rnorm > sim1.dat

sim2.dat: params.R simulator.R
        Rscript simulator.R runif > sim2.dat

sim1.pdf: sim1.dat plotter.R
        Rscript plotter.R sim1.dat

sim2.pdf: sim2.dat plotter.R
        Rscript plotter.R sim2.dat
```

Graphical description of dependencies

Blank: draw dependencies here

Makefile conventions

- PHONY targets: denote actions; ignore filenames with same name. PHONY targe ts are always out of date, and so always run.

```
.PHONY: all clean
all: report.pdf

clean:
        rm -f report.pdf report.log report.aux
        rm -f sim1.* sim2*
```

command	action
make	check first rule
make all	rebuild everything
make clean	remove files that can be rebuilt
touch file	update timestamp, preserving contents

Makefile: next steps

- variables
- implicit rules
- saving space:

```
sim2.dat: params.R simulator.R
      Rscript simulator.R runif > sim2.dat

sim2.dat: params.R simulation.R
      Rscript simulator.R runif > $@
```

- parallel processing *make -j2 job*
- Further reading:

http:
//linuxdevcenter.com/pub/a/linux/2002/01/31/make_intro.html

Makefile: example lab work

- In the lab session, download rr$_{\mathrm{make}}$.tar
- Experiment with remaking report after changing parameters.
- Add a new plot to the report, using sim3 – sampling N numbers from rgamma with new parameters (stored in params.R). You will need to edit simulator.R too.

Sweave: literate programming for R

- Sweave is the system for mixing latex and R code in the same document.
- Used within R often to create "vignettes" which can be dynamically run.
- Allows you to write reports where results (tables,graphs) are automatically generated by your R code.

Sweave: including code chunks

- An example code chunk: by default we are in 'LaTeX mode'.

```
We can then test the procedure a few
times, using the default number
of darts, 1000:

<<>>=
replicate(9, estimate.pi())
@
```

Sweave: including graphs

- Automatically creates filenames, e.g. `estimate-001.pdf`
- By default will generate .ps and .pdf; so change options:

```
 \SweaveOpts{echo=T,pdf=T,eps=F,eval=T,keep.source=T}

\setkeys{Gin}{width=0.6\textwidth}
\begin{center}
<<fig=TRUE>>=
r <- 1; n <- 50; par(las=1)
plot(NA, xlim=c(-r,r), ylim=c(-r,r), asp=1, bty='n',
     xaxt='n', yaxt='n', xlab='', ylab='')
axis(1, at=c(-r,0,r)); axis(2, at=c(-r,0,r))
symbols(x=0, y=0, circles=r, inch=F, add=T)
...
rect(-r, -r, r, r, border='blue', lwd=2)
@
\end{center}
```

Sweave: including tables

- Use the *xtable* package from CRAN.
- Example from that package:

```
<<echo=FALSE>>=
library(xtable)
data(tli)
@

<<label=tab1,echo=FALSE,results=tex>>=
     ## Demonstrate data.frame
     tli.table <- xtable(tli[1:20,])
     digits(tli.table)[c(2,6)] <- 0
     print(tli.table)
@
```

Sweave: including inline computation

```
In this case the number of darts within
the circle is \Sexpr{d}, and so the estimated
value is $\pi \approx \Sexpr{4*d/n}$.
```

Sweave: a full example

- Example application: estimate the value of π using the

dartboard method.

- estimate.Rnw
- See handout of estimate.Rnw and estimate.pdf
- For nice ways to customize Sweave output (as in estimate.Rnw:

```
http://www.stat.auckland.ac.nz/~stat782/downloads/
Sweave-customisation.pdf
```

- Compiling the document with make:

```
estimate.pdf: estimate.Rnw
        R CMD Sweave estimate.Rnw
        pdflatex estimate.tex
```

Sweave: issues and next steps.

- If you edit .tex, Sweave code is re-run. Compare with Makefiles, which offer finer-level control.
- Tedious to keep running with long calculations. cacheSweave package will help to cache results.

- FAQ available:

```
http://www.stat.uni-muenchen.de/~leisch/Sweave/FAQ.html
```

- *odfWeave* and *RHTML* packages allow for output to OpenOffice and HTML.
- matrices/ data frames can be export, e.g. using xtable package.

Other approaches to RR

- R packages: truly reproducible research. R packages allow you to include code , data, documentation, vignettes.
- Org babel: Only Emacs users need apply. Key advantage: allows many different languages to be included in one document, with textual communication between those programs.

Extra handouts

1. report.pdf
2. estimate.Rnw
3. estimate.pdf