

Scientific Programming with R Reproducible Research in R (RRR)

Stephen Eglen Laurent Gatto

November 2, 2022

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:

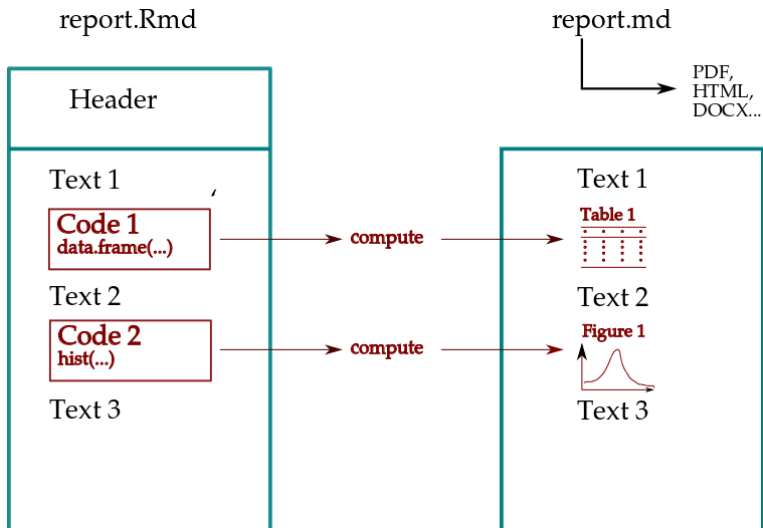
CTANGLE

converts a source file `foo.w` to a
compilable program file `foo.c`;

CWEAVE

converts a source file `foo.w` to a
prettily-printable and
cross-indexed document file `foo.tex`.

Tangling and weaving



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.
- Bioconductor has had a positive impact on genomic data analysis.

Duke cancer trials / Harvard Economics

- Technical details (37 mins, Cambridge 2010)
http://videlectures.net/cancerbioinformatics2010_baggerly_irrh/
- Wide audience, but rather narrow-sighted: 13-minute video from 60 minutes:
<https://www.youtube.com/watch?v=W5sZTNPMQRM>
- “For an economist, the five most terrifying words in the English language are: I can’t replicate your results. But for economists Carmen Reinhart and Ken Rogoff of Harvard, there are seven even more terrifying ones: I think you made an Excel error.”
[The Atlantic](#)

Approaches to reproducible research

1. Makefiles
2. Sweave / knitr
3. Build a package

1. Make and Makefiles

- Make is an automated 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 (i.e the recipes are executed) if the **target** is older than any of its **dependencies** (**prerequisites**).

```
target: prerequisites ...  
    recipe  
    ...
```

- make works backwards from the target to the prerequisites and compares creation time of files (timestamp).

Example rule

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

We can define the (a) target, (b) dependencies (c) commands to run.

A complete example

```
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

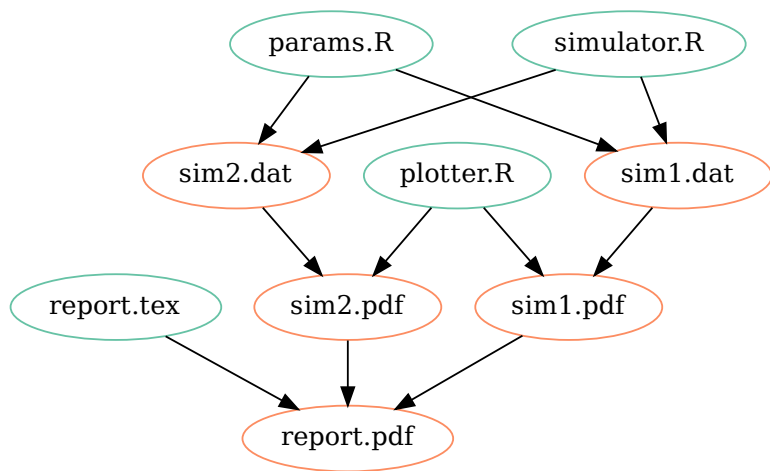
.PHONY: all clean

all: report.pdf

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

https://github.com/lgatto/spr/blob/master/rr_make/Makefile

Graphical representation



Makefile conventions

- PHONY targets: denote actions; ignore filenames with same name. PHONY targets are 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: simulator.R params.R
    Rscript $< runif > $@
```

- parallel processing `make -j8 jobs`

Further reading

- Managing Projects with GNU Make
<http://oreilly.com/catalog/make3/book/index.csp>
- The GNU make manual
<http://www.gnu.org/software/make/manual/make.html>
- Tutorial (work in progress) <https://github.com/sje30/make-tutorial/blob/master/make-tut.pdf>

2. Sweave: literate programming for R

1. Sweave is the system for mixing \LaTeX and R code in the same document.
2. Used within R often to create “vignettes” which can be dynamically run.
3. Allows you to write reports where results (tables, graphs) are automatically generated by your R code.
4. **Sweave** was the first package for mixing R/LaTeX; **knitr** is the dominant version in use today. Knitr much more useful.

Code chunks

By default we are in 'LaTeX mode', and we switch to code mode by using a code chunk.

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

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

... back to latex here.

What is markdown?

What is markdown? A light-weight markup language for generating HTML.

Example

Here is some `*markup text*` with `**bold**` and a
[link](http://www.rstudio.org).

Maths can be included $x^2 + y^2 = z^2$.

Useful also e.g. on github.

Rmarkdown is an R package that embeds R code into markdown. See [cheat sheet](#).

A full example

Estimate the value of π using the dartboard method.

[https:](https://github.com/lgatto/spr/blob/master/estimate/estimatek.Rnw)

[//github.com/lgatto/spr/blob/master/estimate/estimatek.Rnw](https://github.com/lgatto/spr/blob/master/estimate/estimatek.Rnw)

[https:](https://github.com/lgatto/spr/blob/master/estimate/estimatek.Rmd)

[//github.com/lgatto/spr/blob/master/estimate/estimatek.Rmd](https://github.com/lgatto/spr/blob/master/estimate/estimatek.Rmd)

shows how to include figures, tables, and caching feature.

See also <https://bookdown.org>

Quarto

- quarto.org is “an open-source scientific and technical publishing system build on Pandoc”.
- Supports R, Julia, Python. Can create source from markdown or Jupyter notebooks.
- Unifies approaches for different languages and different formats.
- Renders documents, web pages, presentations in large number of formats.
- See e.g. [Julia lecture notes](#).

3. Building an R package

- Building an R packages allows you to bundle your code, data, tests, vignettes into form that others can easily install and run.
- See e.g. <https://devtools.r-lib.org/>
- Or, for beginners, use **fusen**: “inflates a Rmarkdown file to magically create a package.” <https://thinkr-open.github.io/fusen/>

Summary

1. Concept of reproducible research
2. Make
3. R approaches (knitr/rmarkdown/quarto)
4. Building your own packages