You (S)wove? Well (S)tangle now!

You (S)wove? Well (S)tangle now!

Vincent Goulet

École d'actuariat, Université Laval

TUG 2024
Prague, July 19–21, 2024

Source code
View on GitLab

Cover
Variegated, or painted, grasshopper (*Zonocerus variegatus*) in Ghana. Do not be fooled by its luxuriant colors: this species of grasshopper is considered an important agricultural pest in much of Western and Central Africa. Photo credit: © Charles J. Sharp, sharpphotography, CC BY-SA 4.0, via Wikimedia Commons, where this image was selected as picture of the day for 6 December 2018.

“Night and day to each comer
 I sang, if you please.”
 “You sang! I’m at ease;
For ’tis plain at a glance,
Now, ma’am, you must dance.”

Jean de La Fontaine
(English translation by Elizur Wright)

The software tools that had the biggest impact on the way I work:

- R
- Version control
- Literate programming with Sweave

The software tools that had the biggest impact on the way I work:

- R
- Version control
- Literate programming with Sweave

IFT-[47]902 **Programmation avec R pour l'analyse de données**

Gestion de versions avec Git

Programmation lettrée et R Markdown

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be *works of literature*. Hence, my title: "Literate Programming."

— Donald Knuth, 1982

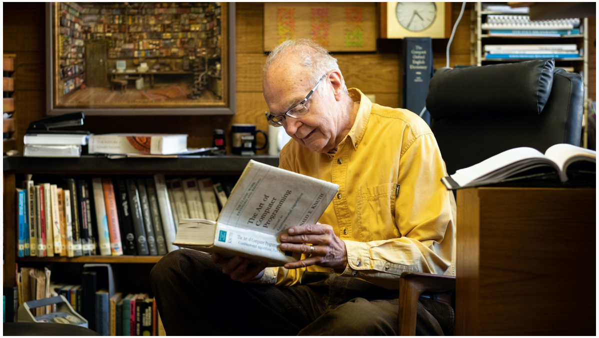


Photo: Vivian Cromwell. © Simons Foundation, via QuantaMagazine.org.

The players

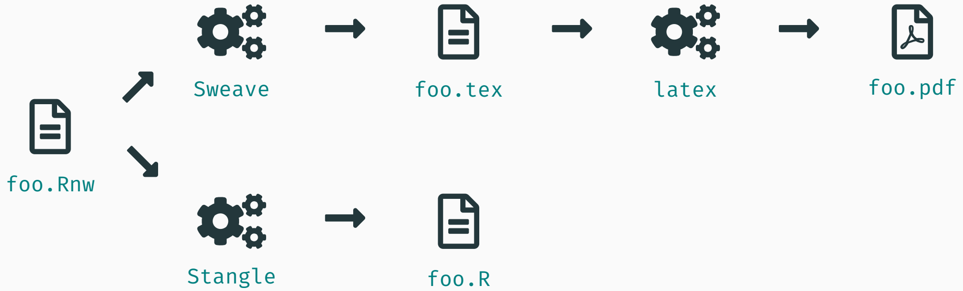
Literate programming in a nutshell

Create programs that are suitable literature for human beings.

- Combine the source code and the documentation in one file
- *Weave* procedure to extract the documentation
- *Tangle* procedure to extract the source code
- Many systems over the years: WEB (Knuth, 1984), CWEB (Knuth and Levy, 1987), doc (Mittelbach, 1989), noweb (Ramsey, 1989), Sweave (Leisch, 2002), knitr (Xie, 2012), ...

Illustration

Workflow for a web of LaTeX and R code processed using Sweave.



The set-up

Nowadays

Literate programming is a cornerstone of reproducible research.

Nowadays

Literate programming is a cornerstone of reproducible research.

Concept behind leading tools of data science.

Nowadays

Literate programming is a cornerstone of reproducible research.

Concept behind leading tools of data science.

Emphasis is often on the weave procedure.

The hook

Some non-standard situations

Maintain documentation and code together even if the code does not create the text.

Some non-standard situations

Maintain documentation and code together even if the code does not create the text.

Extract the code to include it verbatim in a document.

Some non-standard situations

Maintain documentation and code together even if the code does not create the text.

Extract the code to include it verbatim in a document.

Part of the code relies on other parts being saved as files.

The solution should be self-contained

The solution should be self-contained

(`make` is otherwise always a solution)

The tale

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

* included in the manual, but not otherwise used to create the document

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

Source code — originally

`.tex` files

`.R` files

* included in the manual, but not otherwise used to create the document

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

* included in the manual, but not otherwise used to create the document

Source code — originally

`.tex` files

`.R` files

⊗ cumbersome to synchronize and maintain

⊗ manual validation of the code

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

* included in the manual, but not otherwise used to create the document

Source code — now

.Rnw files

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

* included in the manual, but not otherwise used to create the document

Source code — now

`.Rnw` files

- ✓ fewer files to maintain
- ✓ sample code split by section
- ✓ automatic validation of the code

Example 1

Textbook *Programmer avec R* (source code)

Material

manual

sample R code*

* included in the manual, but not otherwise used to create the document

Source code — now

.Rnw files

⚠ R script files need to exist at compilation time

Example 2

R programming term project

Material for students

assignment	unit tests	conf. files*

Material for TAs

solutions	grading tests shell scripts	conf. files*

* for Roger the Omni Grader

Example 2

R programming term project

Material for students

assignment

unit tests

conf. files*

Material for TAs

solutions

grading tests
shell scripts

conf. files*

Source code

one huge `.Rnw` file

* for Roger the Omni Grader

Example 2

R programming term project

Material for students

assignment | unit tests | conf. files*

Material for TAs

solutions | grading tests shell scripts | conf. files*

* for Roger the Omni Grader

Source code

one huge .Rnw file

.Rnw file = 3700 lines
.tex file = 900 lines

Example 2

R programming term project

Material for students

assignment

unit tests

conf. files*

Material for TAs

solutions

grading tests
shell scripts

conf. files*

* for Roger the Omni Grader

Source code

one huge `.Rnw` file

✓ proximity between questions and solutions

✓ unit tests executed on the solutions

Example 2

R programming term project

Material for students

assignment

unit tests

conf. files*

Material for TAs

solutions

grading tests
shell scripts

conf. files*

Source code

one huge `.Rnw` file

⚠ solution files need to exist for unit tests

* for Roger the Omni Grader

The sting

Tangle now!

Sweave evaluates all the code in a `.Rnw` file.

Tangle now!

Sweave evaluates all the code in a `.Rnw` file.

Stangle extracts all the code from a `.Rnw` file.

Tangle now!

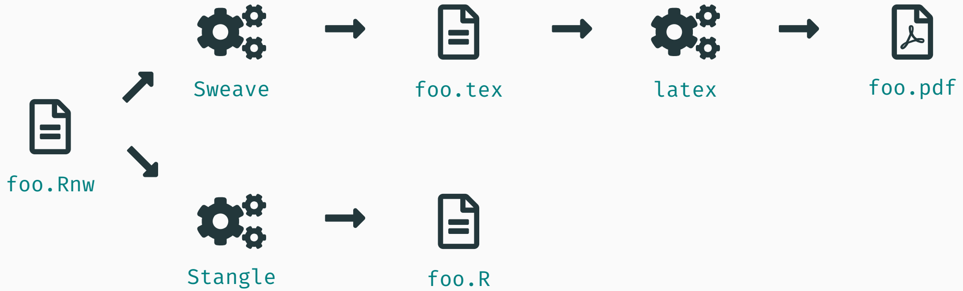
Sweave evaluates all the code in a `.Rnw` file.

Stangle extracts all the code from a `.Rnw` file.

Combine the two by calling Stangle inside Sweave!

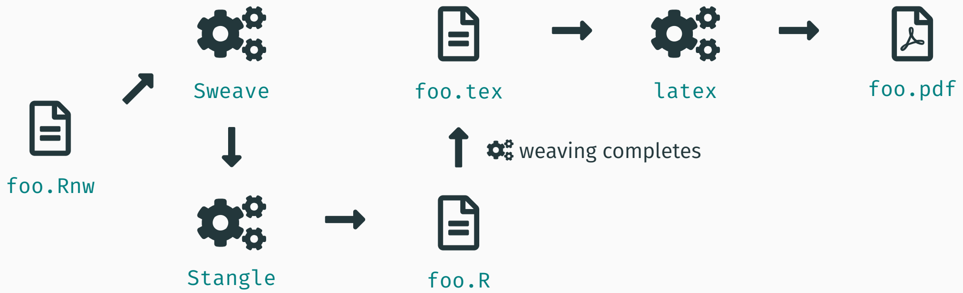
Illustration

Workflow for a web of LaTeX and R code processed using Sweave.



Illustration

Workflow using Stangle inside Sweave.



Example (simplified)

Base structure for the document and solutions.

```
\begin{document}
\maketitle

<<echo=FALSE>>=
FILE <- getSourceName()
Stangle(FILE, driver = "RtangleExtra",
        annotate = FALSE, split = TRUE)
@

Write a function \code{importStations} that...
<<importStations>>=
<<license-solutions>>
importStations <- function(file)
    ...
@

Here is an example:
<<echo=TRUE>>=
importStations("foo.csv")
@
```

Sweave uses the noweb syntax

- `<<>>=` activates code chunk mode
- `@` activates documentation mode
- `<<>>` includes a code chunk

Example (simplified)

Base structure for the document and solutions.

```
\begin{document}
\maketitle

<<echo=FALSE>>=
FILE <- getSourceName()
Stangle(FILE, driver = "RtangleExtra",
        annotate = FALSE, split = TRUE)
@

Write a function \code{importStations} that...
<<importStations>>=
<<license-solutions>>
importStations <- function(file)
    ...
@

Here is an example:
<<echo=TRUE>>=
importStations("foo.csv")
@
```

Code chunk to launch the tangling process

- `getSourceName` retrieves the name of the processed file (local function)
- more on `RtangleExtra` in a minute

Example (simplified)

Base structure for the document and solutions.

```
\begin{document}
\maketitle

<<echo=FALSE>>=
FILE <- getSourceName()
Stangle(FILE, driver = "RtangleExtra",
        annotate = FALSE, split = TRUE)
@

Write a function \code{importStations} that...
<<importStations>>=
<<license-solutions>>
importStations <- function(file)
    ...
@

Here is an example:
<<echo=TRUE>>=
importStations("foo.csv")
@
```

Code chunk for a solution

- creates the solution file `importStations.R` on tangling
- code parsed on weaving

Example (simplified)

Base structure for the document and solutions.

```
\begin{document}
\maketitle

<<echo=FALSE>>=
FILE <- getSourceName()
Stangle(FILE, driver = "RtangleExtra",
        annotate = FALSE, split = TRUE)
@

Write a function \code{importStations} that...
<<importStations>>=
<<license-solutions>>
importStations <- function(file)
    ...
@

Here is an example:
<<echo=TRUE>>=
importStations("foo.csv")
@
```

Code chunk for an example in the text

- function defined above on weaving
- displays the code and results in the document

Example (continued)

Structure for unit tests.

```
<<tests-importStations, ignore.on.tangle=TRUE>>=
source("importStations.R")
stopifnot(...)
@

<<tests-revenues, ignore.on.tangle=TRUE>>=
source("revenues.R")
stopifnot(...)
@

<<tests, ignore.on.weave=TRUE>>=
<<license-tests>>
<<tests-importStations>>
<<tests-revenues>>
@
```

We assume that other code chunks created the files `importStations.R` and `revenues.R` on tangling

Example (continued)

Structure for unit tests.

```
<<tests-importStations, ignore.on.tangle=TRUE>>=
source("importStations.R")
stopifnot(...)
@

<<tests-revenues, ignore.on.tangle=TRUE>>=
source("revenues.R")
stopifnot(...)
@

<<tests, ignore.on.weave=TRUE>>=
<<license-tests>>
<<tests-importStations>>
<<tests-revenues>>
@
```

Code chunks to define unit tests

- ignored on tangling
  (no files are created)
- executed on weaving
  (solutions are validated)

Example (continued)

Structure for unit tests.

```
<<tests-importStations, ignore.on.tangle=TRUE>>=
source("importStations.R")
stopifnot(...)
@

<<tests-revenues, ignore.on.tangle=TRUE>>=
source("revenues.R")
stopifnot(...)
@

<<tests, ignore.on.weave=TRUE>>=
<<license-tests>>
<<tests-importStations>>
<<tests-revenues>>
@
```

Code chunk to assemble the tests in one file

- created on tangling
  (one test file to rule them all)
- ignored on weaving
  (you only validate once)

The shut-out

Extending Sweave (just a little)

My R package **RWeaveExtra** provides additional Sweave drivers with extra tricks up their sleeve.

Extending Sweave (just a little)

My R package **RWeaveExtra** provides additional Sweave drivers with extra tricks up their sleeve.

- Option `ignore.on.weave` to skip a code chunk on weaving, yet tangle it as is
  - allows code in other languages!

Extending Sweave (just a little)

My R package **RWeaveExtra** provides additional Sweave drivers with extra tricks up their sleeve.

- Option `ignore.on.weave` to skip a code chunk on weaving, yet tangle it as is
  - allows code in other languages!
- Option `ignore.on.tangle` to omit a code chunk on tangling, yet weave it as is
  - avoids cluttering!

Extending Sweave (just a little)

My R package **RWeaveExtra** provides additional Sweave drivers with extra tricks up their sleeve.

- Option `ignore.on.weave` to skip a code chunk on weaving, yet tangle it as is
  - allows code in other languages!
- Option `ignore.on.tangle` to omit a code chunk on tangling, yet weave it as is
  - avoids cluttering!
- Option `extension` to specify the extension of the file name on tangling
  - sets the correct extension for other languages!

This document was typeset with the XƎLaTeX document document preparation system using the **beamer** class and the Metropolis theme. The main text is in Fira Sans, and the computer code in Fira Mono. Icons come from Font Awesome.

Movie references: The Sting (1973), The Lord of the Rings (2001–2003), You Only Live Twice (1967)