

Gérer ses documents efficacement avec la programmation lettrée

R à Québec 2019

15 mai 2019

Gérer ses documents efficacement avec la programmation lettrée

R à Québec 2019

Vincent Goulet

Professeur titulaire

École d'actuariat, Université Laval



Vincent Goulet, 2019

© 2019 par Vincent Goulet. « Gérer ses documents efficacement avec la programmation lettrée — R à Québec 2019 » est mis à disposition sous licence **Attribution-Partage dans les mêmes conditions 4.0 International** de Creative Commons. En vertu de cette licence, vous êtes autorisé à :

- **partager** — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats ;
- **adapter** — remixer, transformer et créer à partir du matériel pour toute utilisation, y compris commerciale.

L'Offrant ne peut retirer les autorisations concédées par la licence tant que vous appliquez les termes de cette licence.

Selon les conditions suivantes :



Attribution — Vous devez créditer l'œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'œuvre. Vous devez indiquer ces informations par tous les moyens raisonnables, sans toutefois suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son œuvre.



Partage dans les mêmes conditions — Dans le cas où vous effectuez un remix, que vous transformez, ou créez à partir du matériel composant l'œuvre originale, vous devez diffuser l'œuvre modifiée dans les mêmes conditions, c'est-à-dire avec la même licence avec laquelle l'œuvre originale a été diffusée.

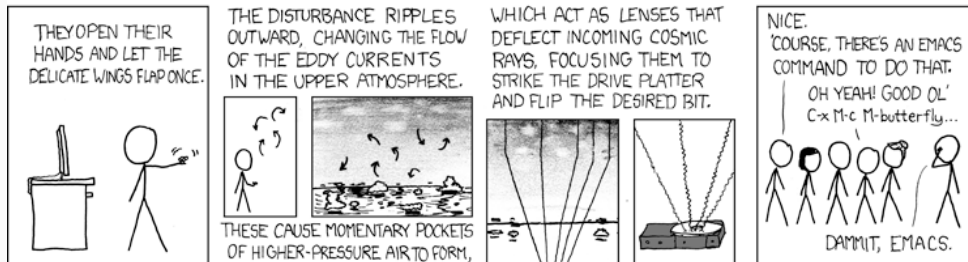
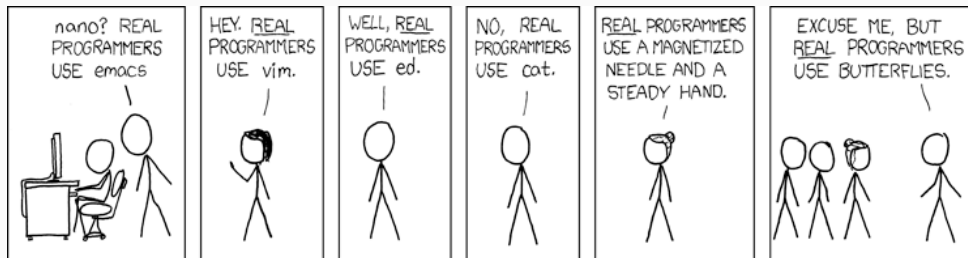
Code source

 [Voir sur GitLab](#)

Couverture

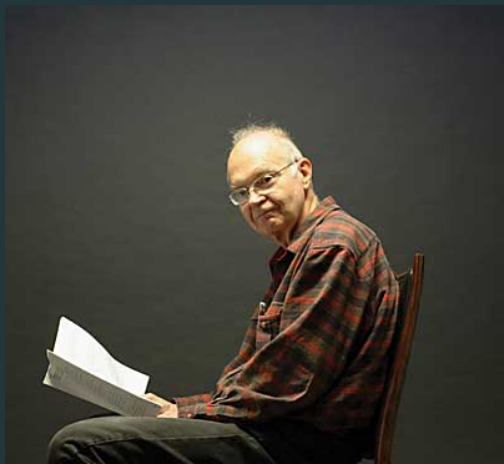
Lémur catta (*Lemur catta*) dans le parc national de l'Isalo, Madagascar.

Crédit photo : © Bernard Gagnon **CC BY-SA 3.0 non transposé**, via [Wikimedia Commons](#).



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



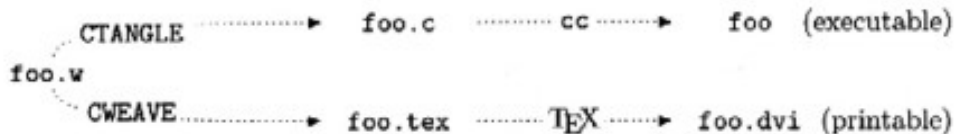
Ce qui a le plus contribué à changer mes méthodes de travail depuis 20 ans

- S-PLUS et R
- Gestion de versions
- Sweave et la programmation lettrée

Programmation lettrée

Une (autre) idée de Donald Knuth lorsqu'il a créé T_EX pour rédiger *The Art of Computer Programming*.

- Combiner un programme et sa documentation dans un même fichier dans un format facile à consulter pour un **humain**



- Plusieurs systèmes au fil du temps : **WEB** [↗](#) (Knuth, 1984), **CWEB** [↗](#) (Knuth et Levy, 1987), **noweb** [↗](#) (Ramsey, 1989), **Sweave** [↗](#) (Leisch, 2002), **knitr** [↗](#) (Xie, 2012)

Exemple de document combinant \LaTeX et code R

foo.Rnw

L'utilisateur de R interagit avec l'interprète en entrant des commandes à l'invite de commande:

```
<<echo=TRUE>>=
```

```
2 + 3
```

```
@
```

La commande `\verb=exp(1)=` donne `\Sexpr{exp(1)}`,
la valeur du nombre e .

Après traitement par Sweave dans R

`Sweave("foo.Rnw") → foo.tex`

L'utilisateur de R interagit avec l'interprète en entrant des commandes à l'invite de commande:

```
\begin{Schunk}
```

```
\begin{Sinput}
```

```
> 2 + 3
```

```
\end{Sinput}
```

```
\begin{Soutput}
```

```
[1] 5
```

```
\end{Soutput}
```

```
\end{Schunk}
```

La commande `\verb=exp(1)=` donne 2.71828182845905,
la valeur du nombre e .

Processus en un coup d'oeil



R Markdown : même idée

- Texte en format Markdown plutôt que \LaTeX
- Traitement avec knitr plutôt qu'avec Sweave
- Sortie dans plusieurs formats différents, dont HTML



Nous pouvons faire mieux avec la programmation lettrée

Atelier d'introduction à la science des données ([code source](#))

Matériel pédagogique



diapositives



fichier de script

Atelier d'introduction à la science des données ([code source](#))

Matériel pédagogique



diapositives



fichier de script

Source



fichier unifié

- 💡 gestion facilitée
- 💡 validation du script



Étudiez le fichier `exemple-diapos.Rnw` livré avec ces diapositives. Il fournit la structure de base d'un fichier combinant le texte de diapositives et le code destiné à être distribué séparément dans un fichier de script.

Second exemple

Travail pratique de programmation en R

Matériel à gérer



énoncé



solutions



tests unitaires

Second exemple

Travail pratique de programmation en R

Matériel à gérer



énoncé



solutions



tests unitaires

Source



fichier unifié

💡 proximité entre questions
et solutions

💡 tests exécutés sur solutions



Étudiez le fichier `exemple-travail.Rnw` livré avec ces diapositives. Il fournit la structure de base d'un fichier combinant l'énoncé d'un travail de programmation, les solutions et les tests unitaires.

- Sweave exécute tout le code R d'un fichier `.Rnw`

Astuce

- Sweave exécute tout le code R d'un fichier `.Rnw`
- Stangle extrait tout le code R d'un fichier `.Rnw`

- Sweave exécute tout le code R d'un fichier `.Rnw`
- Stangle extrait tout le code R d'un fichier `.Rnw`
- Combiner les deux et exécuter Stangle à l'intérieur de Sweave!

```
<<echo=FALSE>>=
## nom du présent fichier avec et sans l'extension
file <- "foo.Rnw"
prefix <- paste0(sub(".Rnw", "", file), "-")

## extraction des scripts
Stangle(file, annotate = FALSE, split = TRUE)

## changement du nom des fichiers de script
scripts <- list.files(pattern = paste0("^", prefix, "[^0-9].*\\.R$"))
file.rename(scripts, sub(prefix, "", scripts))
@
```

Limitations et inconvénients

- Il ne faut pas oublier de nommer les blocs de code (*chunks*)
- Compliqué d'éviter que du code ne soit **pas** exécuté
(non, `eval = FALSE` n'est pas la solution)
- Pas fameux pour des applications Shiny
(justement à cause de la puce précédente)
- Stangle laisse trainer des fichiers de type `foo-001.R`

Ce document a été produit par le système de mise en page \LaTeX avec la classe **beamer** et le thème Metropolis. Le texte est composé en Fira Sans, les mathématiques en Arev Math et le code informatique en Fira Mono. Les icônes proviennent de la police Font Awesome.

