

Introduction aux documents dynamiques avec R

Stéphane Le Vu

stephane.le-vu@santepubliquefrance.fr

DMI

13 juin 2019

① Introduction

② Pratique

Texte markdown et code R

Equations, figures et tableaux

Cache, script externe et références

Formats de sortie

Rapports itératifs

③ Exemple complet

④ Documentation

Introduction

Objectifs

- Comprendre l'intérêt d'un document dynamique
- Apprendre à écrire un fichier R Markdown
- Compiler le document en plusieurs formats

Intérêt

- Documents contenant les étapes de l'analyse
 - Code
 - Résultats
 - Illustrations
 - **Commentaires**
- Avantages
 - Automatisé, reproductible, traçable
 - Sépare structure et contenu
 - Evite les copier-coller et les documents
"résultats.final.base_provisoire.v2.3_bis"
- Limites
 - Pas de mode de révision type Word
 - Longueur du code
 - Temps de compilation

Bases de R nécessaires

- Savoir installer et charger un package

```
install.packages("le_package")  
library(le_package)
```

- Savoir que `package::objet` permet d'accéder à un objet d'un package. Par exemple, la fonction `fn1` du package `le_package` est invoquée par

```
le_package::fn1
```

Installation

- Si RStudio n'est pas installé

```
install.packages("rmarkdown")
```

- Si LaTeX n'est pas installé, pour des documents pdf (optionnel)

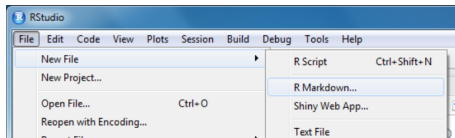
```
install.packages('tinytex')  
tinytex::install_tinytex()  
## vérifier  
tinytex::is_tinytex()
```

Exemples

- Les exemples sont dans des projets RStudio distincts
- Chacun s'ouvre dans une nouvelle session de R (En cliquant `ex*.Rproj`)
- Ils sont indiqués en rouge dans la présentation

Test installation complète

- Pour vérifier que R, RStudio et LaTeX sont bien installés
- Dans RStudio, ouvrir le menu *File* > *New File* > *R Markdown*



- Sélectionner *Documents* à gauche et *PDF* à droite
- Cliquer sur le bouton  ou Ctrl + Maj + K
- Eventuellement accepter l'installation de packages
- Cela doit produire un PDF

Outils compris dans rmarkdown

- **Markdown** : Langage balisé (*markup*), cad du texte avec des balises de format, comme du html ou du \LaTeX , mais en plus simple et lisible. Sa version RStudio s'appelle *R Markdown*, comme le package `rmarkdown` !
- **knitr** : un autre package *tricote* le code R avec le texte Markdown du fichier `.Rmd` et produit un fichier `.md`
- **Pandoc** : programme qui permet la conversion de Markdown à presque tous les autres formats, dont `.pdf` pour impression et `.html` pour écran et web



Balises

LaTeX

```
% commentaire
\documentclass{article}
\begin{document}
\section*{Titre}
\subsection*{Sous-titre}
\emph{Italique}
\textbf{gras}
\end{document}
```

Markdown

```
<!-- commentaire -->
# Titre
## Sous-titre
*Italique*
**gras**
```

Pratique

Texte markdown et code R

Premier rapport

- Préambule en YAML
- Comprend au moins le format de sortie entouré de ---

```
---  
title: "Mon premier document RMD"  
author: Richard Virenque  
date: 27/06/2019  
output: html_document  
---
```

- Ecrire ce préambule ou ouvrir **ex01**, et *Knit*

Syntaxe RMarkdown

- Permet de mettre en forme du simple texte
- Voir [ex01](#) et [aide-mémoire_rmarkdown](#)

Premier morceau de code R

- Le morceau comprend une accolade {langage nom, options} et du code, le tout entouré par trois accents graves (*backticks*)
- Le nom du morceau ne doit pas contenir de tiret bas _
- Le raccourci Ctrl + Alt + I crée un nouveau morceau (*chunk*)

```
```{r morceau1}  
x <- 4
x
x < 0
```
```

- Ouvrir **ex02** et *Knit*

Options du morceau

- Principales options
 - `echo = FALSE` masque le code
 - `results = 'hide'` masque la sortie
 - `include = FALSE` masque code et sortie¹
 - `eval = FALSE` n'exécute pas le code
 - `warnings = FALSE` masque les avertissements
- Copier l'exemple `ex02` et tester ces options

1. Le code est quand même exécuté

Options globales des morceaux

- Les options peuvent être globales à tous les morceaux (voir [ex02.1](#)), avec

```
knitr::opts_chunk$set(...options...)
```

- Attention à garder la traçabilité de l'analyse (voir [ex02.2](#))

Environnement

- Le document Rmd a son propre environnement, indépendant de la session dans laquelle il est ouvert
 - Les objets doivent y être créés
 - Les packages doivent y être chargés
 - Les chemins doivent être relatifs à l'emplacement du script .Rmd
- Exécuter **ex03** et corriger

Solution de ex03

- Dans l'exemple **ex03.1**

Code inline

- Le texte Markdown peut inclure du code R, dit *inline*
- Cela s'écrit entre deux accents graves avec l'expression R précédée de `r`, comme cela : ``r 1+1 == 3``
- Essayer `ex04`

Equations, figures et tableaux

Equations

- Utilisent la syntaxe LaTeX
- Les symboles sont généralement précédés d'une barre oblique inversée `\` (*backslash*)
- Dans une phrase, l'équation est balisée par `$...$`
- Par exemple, `$ A = \pi \times r^2 $` donne $A = \pi \times r^2$
- L'équation séparée est délimitée par `$$...$$` ou `\[...\]`
- Par exemple, `$$R = \frac{A}{B}$$` donne :

$$R = \frac{A}{B}$$

- Les symboles sont documentés ici : [aide_formules_TeX](#)

Figures et images

- L'insertion de figures R est simple
 - Les figures générées par R dans un morceau s'affichent dans la sortie
 - Les images externes sont insérées depuis un morceau par `knitr::include_graphics("chemin_image.ext")` ou directement dans le texte markdown par `![légende](image)`
- Le formatage requiert des options
 - `fig.width` et `fig.height` (en *inch* ou *cm*) modulent les figures R,
 - `out.width` (en %) module figures et images
 - `fig.cap` définit une légende
- Voir **ex05**

Tableaux

- L'insertion de “beaux” tableaux est plus difficile
- Voir [ex06](#) pour les tableaux html et [ex07](#) pour les tableaux pdf

```
cars[1:3,]
```

```
##    speed dist
## 1      4     2
## 2      4    10
## 3      7     4
```

```
knitr::kable(cars[1:3,],
  format = "latex",
  booktabs = TRUE)
```

| | | speed | dist |
|--|--|-------|------|
| | | 4 | 2 |
| | | 4 | 10 |
| | | 7 | 4 |

Cache, script externe et références

Cache des morceaux

- A chaque compilation, tout le code est exécuté *de novo*
- Pour garder le résultat d'un morceau, on utilise l'option `cache = TRUE`
- Ce résultat est enregistré séparément et réutilisé sans que le morceau ne soit ré-exécuté²
- C'est dangereux si ce résultat dépend d'un élément (précédent) susceptible de changer
- Dans ce cas, on peut déclarer la dépendance par `dependson = "morceau_n-1"`
- Voir `ex08`

2. Tant qu'il n'est pas modifié

Intérêt d'un script R externe

- Plus pratique de tester son code dans un simple script R
- Evite des doublons de code
- Permet d'utiliser le même script pour plusieurs rapports
- Améliore la lisibilité du rapport

`script.R`

`rapport.Rmd`

Utilisation d'un script R externe

- ① Utiliser tout un script R (ou plusieurs scripts différents) pour alimenter un rapport Rmd, avec `source("script.R")`, par exemple pour importer des données
 - Ouvrir `ex09`, compiler `ex09.1.Rmd` et corriger
- ② Utiliser des morceaux³ du script avec `knitr::read_chunk("script.R")`
 - Les morceaux “externes” sont alors appelés par `{r nom-du-chunk-externe}`
 - Ouvrir `ex09`, compiler `ex09.2.Rmd` et corriger

3. Les morceaux doivent être nommés et délimités par un bloc de 4 tirets ----.

Solution de ex09.2

- Dans **ex10**

Insertion de références

- Des références bibliographiques au format BibTeX, RIS, EndNote, etc. peuvent être insérées par leur identifiant entre crochets : `[@identifiant]`
- La bibliographie s'ajoute à la fin du texte selon un style `cs1`, déclaré dans le préambule YAML
- Le style `cs1` correspond à un format de citation d'un journal par exemple
- Voir [ex11](#) et les [styles cs1](#)

Formats de sortie

Formats de sortie

- La sortie est définie dans le préambule YAML par output :
- Les sous-options doivent être tabulées
- Par exemple :

```
output:  
  html_document:  
    toc: yes
```

Sorties documents

- `html_document`
- `pdf_document`
- `word_document`
- aussi `odt`, `rtf`, `latex`, etc.
- Voir `ex12` et compiler `ex12.1.Rmd` avec *Knit to ...*

Sorties présentations

- `ioslides_presentation`
- `slidy_presentation`
- `revealjs::revealjs_presentation`
- `beamer_presentation` (pdf)
- `powerpoint_presentation`⁴
- Voir **ex12** et compiler `ex12.2.Rmd` avec *Knit to ...*

4. Non testé. Nécessite Pandoc v2.0.5

Autres formats

- Carnet d'analyse (R Notebook)
 - Permet d'exécuter les morceaux un par un interactivement
 - Intérêt pas évident par rapport à un ou plusieurs scripts R appelés dans un document Rmd
- Mémoire, livre
 - Pour documents à plusieurs sections/chapitres et références internes (très courant avec LaTeX)
 - Voir [bookdown](#)
- Poster
 - Plusieurs solutions à partir de LaTeX et Beamer mais en développement pour R Markdown
 - Voir [posterdown](#)
- Shiny
 - Voir [Document Shiny](#) et [App Shiny](#)
- Tutoriel interactif
 - Voir [learnr](#)

Rapports itératifs

Rapports itératifs

- Le principe est de créer des plusieurs rapports différents à partir d'un rapport type
- Le rapport type `Rmd` contient des paramètres dans le préambule `YAML` qui peuvent changer via un programme
- Dans l'exemple **ex13**, nous allons générer des rapports d'analyse par région
- Le script `ex13.R` utilise la fonction `rmarkdown::render()` pour compiler le rapport type `ex13.Rmd` pour chaque région

Exemple complet

Présentation de l'exemple

- Nous allons créer un document illustrant l'estimation de la constante Pi par simulation de Monte-Carlo
- Le principe est de tirer aléatoirement des coordonnées et d'utiliser la proportionnalité entre le nombre de points inscrits dans un cercle et π

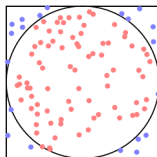


FIGURE 1 – Points inscrits dans le cercle

Instructions

- Ouvrir **ex14**
- Enregistrer votre version du fichier `ex14.Rmd` et compiler
- Utiliser le script `mcpi.r` comme script externe
- Générer un document contenant :
 - Le principe de l'analyse
 - L'image `mcpi_fig0.png`
 - Le code de l'analyse
 - Le résultat chiffré en code *inline*
 - Les figures générées
 - Des commentaires des étapes
 - La référence contenue dans `ref_mcpi.bib`

Solution

- Voir **ex15** : TODO

Documentation

Documentation complémentaire

- Générales
 - [Documentation rmarkdown officielle](#)
 - [Forum stackoverflow](#)
 - [Introduction en français](#)
 - [aide-mémoire_rmarkdown](#)
- Particulières
 - [Options des morceaux de code](#)
 - [Options YAML des sorties html](#)
 - [Options YAML des sorties pdf](#)

Notes

- Par défaut, un nouveau fichier R Markdown s'ouvre avec un code d'exemple qui peut être effacer
- Nommer les morceaux permet de débbugger plus facilement

Autres formations

- Documents complexes avec LaTeX
- Gestion bibliographie avec Zotero
- Figures programmées Tikz
- Intégration C++ dans R
- Package R
- Contrôle de version avec Git et github