Le langage R Markdown

Christophe Lalanne & Bruno Falissard

Table des matières

1 Introduction

R Markdown offre une syntaxe simplifiée pour mettre en forme des documents contenant à la fois du texte, des instructions R et le résultat fourni par R lors de l'évaluation de ces instructions. En ce sens, il s'agit d'un outil permettant de produire des rapports d'analyse détaillés et commentés, plutôt que de simples scripts R incluant quelques commentaires.

Ce langage est basé sur Markdown. Il s'apprend très rapidement, ne nécessite rien d'autre qu'un éditeur texte et il peut être utilisé sur le forum de discussion pour formater les messages, en particulier les instructions ou les blocs d'instructions R.

R Markdown v2 est la nouvelle implémentation disponible depuis 2014 dans le logiciel RStudio. Initialement, la possibilité de transformer du texte contenant des instructions R en document HTML ou PDF reposait sur les packages knitr et markdown, mais à présent il n'existe plus qu'un seul package qui se charge de gérer la conversion des documents R Markdown (extension `.Rmd` ou `.rmd`) en documents PDF, HTML ou DOCX.

Les principales différences et les éventuels problèmes de compatibilité entre les deux versions sont décrites sur la page suivante : Migrating from R Markdown v1.

2 Les bases du langage

Markdown est un langage permettant de baliser du texte simple à l'aide de symboles prédéfinis, un peu à l'image des balises HTML, afin de produire une sortie enrichie avec des titres, des paragraphes, etc.

Il s'agit donc principalement de rédiger un document de type texte, dans un éditeur approprié (éviter MS Word), et d'insérer des symboles simples afin de signaler les mots à mettre en surbrillance ou la délimitation des titres de section, par exemple.

On trouvera une description détaillée des caractéristiques du langage sur les sites suivants :

- Markdown Cheatsheet
- The R Markdown Cheatsheet
- Élaboration et conversion de documents avec Markdown et Pandoc

Par ailleurs, RStudio fournit un petit mémento des principales commandes, cf. le petit bouton d'aide (?) apparaissant à côté du bouton de compilation `Knit HTML` (ou `Knit PDF` selon l'option activée par défaut).

FIGURE 1 – Barre d'outils R Markdown

2.1 Syntaxe et mise en forme d'un document

Les titres de sections sont préfixés d'un ou plusieurs #, selon le niveau de profondeur du titre. Par exemple, pour un titre de niveau 1, on écrira `# Titre de niveau 1`, alors que pour un titre de niveau 2 on écrira `## Titre de niveau 2` (à ne pas confondre avec le symbole de commentaire des scripts R).

Au niveau des éléments de texte, la mise en gras s'effectue en encadrant le texte par `**` et la mise en italique par `*`. Pour utiliser une police à espacement fixe (monospace), on encadrera le texte avec des deux quotes simples inversées (‘). Par exemple, pour écrire ce texte **en gras** et cette instruction R, `smp <- read.csv2("smp2.csv")`, on s'est contenté d'écrire :

```
pour écrire ce texte **en gras** et cette instruction R, `smp <- read.csv2("smp2.csv")`
```

Les instructions R sont encadrées par “‘{r} et “‘ (appelé “code chunk”), comme ci-dessous :

FIGURE 2 – Exemple d'instructions R à évaluer

2.2 Génération du document final

La génération du document final se fait en cliquant sur le bouton `Knit HTML` (ou `Knit PDF` selon l'option choisie). Dans le cas du format PDF, il est nécessaire d'avoir un système LaTeX installé sur le système. Le fichier HTML est sauvegardé dans le répertoire de travail courant et peut être affiché dans un navigateur web ou envoyé par email : les feuilles de style ou les images étant inclues avec le document HTML, il n'est pas nécessaire de les sauvegarder ou de les transmettre avec la page HTML.

Dans le cas où l'on souhaiterait générer le document à partir de R directement, sans passer par RStudio, il suffit d'utiliser la commande `render()` du package `rmarkdown`. Ce package peut être installé de la manière suivante :

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

2.3 Les options de personnalisation

Parmi les principales options de personnalisation, on distinguera

- pour le format HTML : la possibilité d'insérer une table des matières en début de document, utiliser la coloration syntaxique pour l'affichage des commandes R, modifier la feuille de style pour le rendu de la page HTML, modifier la taille des images (hauteur/largeur) et ajouter une légende ;
- pour le format PDF : la possibilité d'insérer une table des matières en début de document et de numéroter les titres de section, modifier la taille des images (hauteur/largeur) et ajouter une légende.

Ces options se gèrent directement à partir du bouton situé à la droite du bouton de compilation `Knit HTML` (ou `Knit PDF` selon l'option activée par défaut).

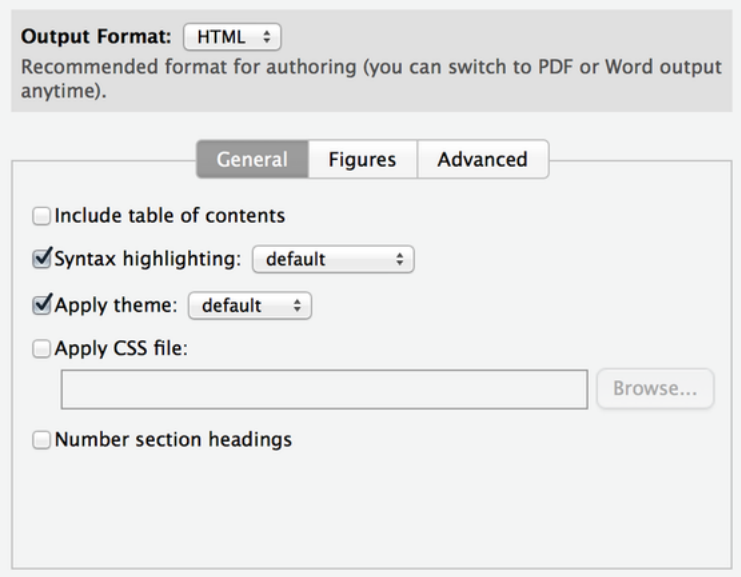

FIGURE 3 – Panneau de configuration des options de sortie HTML

2.4 Gestion des figures et des tableaux

Pour insérer une image externe, on utilise la construction suivante, en supposant que le fichier image s'appelle `fichier.png` :

```
![légende](fichier.png)
```

La gestion des figures générées directement à partir de commandes R se fait à l'aide des options dans les "code chunks". Souvent, on ajoutera une option `echo = FALSE` pour ne pas afficher les instructions R et simplement afficher le résultat produit par R. Les autres options disponibles sont celles du package knitr (par exemple, `fig.align = TRUE` pour l'alignement de la figure, seulement dans le cas HTML).

En ce qui concerne les tableaux, il est possible d'utiliser le package `xtable` qui permet de convertir au format HTML des tableaux simples (par exemple, ceux produits à l'aide de la commande `summary()` pour des data frame ou des modèles de régression). Voici un exemple d'application :

```r
n <- 100
x <- rnorm(n)
y <- 2*x + rnorm(n)
out <- lm(y ~ x)
library(xtable)
tab <- xtable(summary(out)$coef, digits=c(0, 2, 2, 1, 2))
print(tab, type="html")
```

Dans ce cas, on prendra soin de rajouter l'option `results = "asis"` puisque l'on demande explicitement à la commande `print()` de fournir un résultat de type HTML.

Il existe d'autres packages permettant des rendus plus élaborés ou plus complexes, bien que souvent réservés au format PDF, en particulier : Hmisc, texreg, stargazer, apsrtable, rapport, pander, reporttools, ou brew.

3 Pour aller plus loin

Il est conseillé de se familiariser progressivement avec l'usage de la syntaxe Markdown et les options knitr. Un bon tutoriel (en anglais) est disponible sur la page suivante : knitr in a knutshell.

Pour des documents plus complexes, il est généralement nécessaire de recourir au format PDF et d'exploiter au maximum les possibilités du package `knitr` et des options LaTeX ; voir, par exemple, Data Science with R Documenting with KnitR.

4 Références

1. Grandud, C. (2013). Reproducible Research with R and RStudio. Chapman & Hall/CRC
2. Xie, Y. (2013). Dynamic documents with R and knitr. Chapman & Hall/CRC