Rmd: R markdown

Table of Contents

R markdown (.Rmd) permet de générer automatiquement à partir d'un fichier écrit dans un certain format un rapport dynamique. C'est un format qui contient des balises, un peu comme du html, mais en plus simple, et qui permet de concevoir un rapport dynamique comportant :

- **du code R**: ce qui permet de montrer comment les analyses ont été faites, par exemple en termes de jeu de données, ou de fonctions utilisées.
- **les résultats du code R**: il s'agit par exemple des sorties d'un modèle de régression, ou encore d'un plot.
- **des commentaires**: qui permettent, par exemple, d'ajouter une interprétation aux résultats.

Les rapports dynamiques générés à partir d'un fichier R markdown s'intègrent dans le concept de "Recherche reproductible", puisque l'analyse statistique réalisée peut être reproduite, strictement à l'identique, par quiconque dispose du fichier.Rmd. Les rapports peuvent, en outre, être générés sous divers formats, html, pdf ou docx, par exemple. Au final le langage R markdown, est un outil parfaitement adapté à la collaboration et à la diffusion des résultats.

Installation

Prérequis

Installé R (https://www.r-project.org) (R Core Team 2018) et l'IDE RStudio (https://www.rstudio.com). RStudio n'est pas obligatoire, mais recommandé, car il est plus facile pour un utilisateur moyen de travailler avec R Markdown. Si RStudio IDE n'est pas installé ,installer Pandoc (http://pandoc.org), sinon il n'est pas nécessaire d'installer Pandoc séparément car RStudio l'a fourni.

Installer le paquetage rmarkdown dans R:

```
# Install from CRAN
install.packages('rmarkdown')

# Or if you want to test the development version,
# install from GitHub
if (!requireNamespace("devtools"))
  install.packages('devtools')
  devtools::install_github('rstudio/rmarkdown')
```

Pour générer une sortie PDF, vous devez installer LaTeX. Pour les utilisateurs de R Markdown qui n'ont pas encore installé LaTeX, installer TinyTeX (https://yihui.name/tinytex/):

```
install.packages ( "tinytex" )
tinytex::install_tinytex() # install TinyTeX
```

TinyTeX est une distribution LaTeX légère, portable, multiplate-forme et facile à entretenir. Le package compagnon R tinytex (Xie 2019 d) peut aider à installer automatiquement les packages LaTeX manquants lors de la compilation de documents LaTeX ou R Markdown en PDF, et garantit également qu'un document LaTeX est compilé pour le nombre correct de tentatives de résolution de toutes les références croisées.

Organisation du fichier Rmd

L'en-tête

L'en-tête est contenu entre deux séries de pointillés. Par défaut, il contient deux types d'éléments : le titre du document, et son format de sortie. Il est possible d'ajouter d'autres éléments, comme l'auteur, et la date de création, par exemple.

Il est encore possible d'ajouter une table des matières, ou encore un lien vers un fichier de références bibliographiques (voir cet article).

Les chunks contenant le code R

Les parties de code R sont contenues dans des blocs, appelés chunks". Ces chunks commencent et finissent par les balises `

Entre les deux, se trouve une accolade contenant la lettre r. C'est dans cette accolade, après la lettre r (il ne faut pas l'enlever) que les options vont pouvoir être passées, pour choisir de faire apparaître, ou non, le code dans le rapport dynamique, ainsi que les résultats, ou encore pour définir la taille des plots.

Le chunk de set up

Ce chunk se trouve en dessous de l'en-tête, il permet de régler les options par défaut de tous les chunks. Par exemple, on va pouvoir indiquer que l'on ne veut pas faire les messages et les warnings qui pourraient être générés lors de l'exécution des chunck. Au lieu de le faire pour tous les chunks, on peut le faire une seule fois ici.

Et si, pour un chunck donné, on veut faire apparaître les warnings et les messages, on utilisera **message=TRUE** et **warnings=TRUE** dans l'accolade du chunck concerné.

Les parties texte

Elles peuvent être insérées partout en dehors des chunks.

Il est possible de mettre en gras, ou en italique, certaines parties du texte.

Documents interactifs

Les documents R Markdown peuvent également générer un contenu interactif. Il existe deux types de documents R Markdown interactifs **HTML Widgets** ou **Shiny** (ou les deux).

Widgets HTML

Le cadre HTML Widgets est implémenté dans le package R **htmlwidgets** (Vaidyanathan et al. 2018) , qui connecte des bibliothèques JavaScript créant des applications interactives, telles que des graphiques et des tableaux interactifs. Plusieurs packages de widgets ont été développés sur la base de ce cadre, tels que DT (Xie, Cheng et Tan 2018) , des dépliants (Cheng, Karambelkar et Xie 2018) et des polygraphes (Vanderkam et al. 2018).

Bien que les widgets HTML soient basés sur JavaScript, la syntaxe permettant de les créer dans R est souvent une syntaxe pure.

Shiny

Le package Shiny (Chang et al. 2018) construit des applications Web interactives optimisées par R. Pour appeler du code Shiny à partir d'un document R Markdown, ajouter **runtime: shiny** aux métadonnées YAML.

On peut utiliser Shiny pour exécuter le code R en réponse aux actions de l'utilisateur. Étant donné que les navigateurs Web ne peuvent pas exécuter de code R, les interactions Shiny se produisent côté serveur et reposent sur une session R en direct. En comparaison, les widgets HTML ne nécessitent pas de session en direct R pour les prendre en charge, car l'interactivité provient du côté client (via JavaScript dans le navigateur Web).

Les widgets HTML et les éléments Shiny reposent sur HTML et JavaScript. Ils fonctionneront dans

n'importe quel format R Markdown affiché dans un navigateur Web, tel que des documents HTML, des tableaux de bord et des présentations HTML5.

Compiler un document R Markdown

La méthode habituelle pour compiler un document R Markdown consiste à cliquer sur le bouton Knit. Le raccourci clavier correspondant est Ctrl + Shift + K (Cmd + Shift + K sous macOS). RStudio appelle la fonction rmarkdown::render() pour rendre le document dans une nouvelle session R:

- Le rendu d'un document Rmd dans une nouvelle session R signifie qu'aucun des objets de la session R actuelle (par exemple, ceux que ont été créés dans la console R) n'est disponible pour cette session .
- La reproductibilité est la principale raison pour laquelle RStudio utilise une nouvelle session R pour restituer les documents Rmd afin que less documents continuent de fonctionner à la prochaine ouverture de R, ou dans les environnements informatiques d'autres personnes.

Pour rendre un document dans la session R actuelle, on peut également appeler rmarkdown::render() et transmettre le chemin du fichier Rmd à cette fonction. Le deuxième argument de cette fonction est le format de sortie, qui correspond par défaut au premier format de sortie spécifié dans les métadonnées YAML (s'il est manquant, le document par défaut est html_document). Lorsque plusieurs formats de sortie sont spécifiés dans les métadonnées on peut spécifier celui ci dans le deuxième argument, par exemple, pour un document foo.Rmd avec les métadonnées:

```
output:
  html_document:
    toc:  true
  pdf_document:
    keep_tex:  true
```

On peutle rendre au format PDF via:

```
rmarkdown::render('foo.Rmd','pdf_document')
```

L'appel de fonction donne beaucoup plus de liberté (par exemple, on peut générer une série de rapports en boucle). Bien sûr, on peut démarrer une nouvelle session R propre et appeler rmarkdown::render() dans cette session. Tant qu'on ne doive interagir pas manuellement avec cette session (par exemple, en créant manuellement des variables dans la console R), les rapports doivent être reproductibles.

Un autre moyen de travailler avec les documents Rmd est les notebooks R Markdown. Avec les notebooks, on peut exécuter des fragments de code individuellement et voir les résultats directement dans l'éditeur RStudio. C'est un moyen pratique d'interagir ou d'expérimenter avec du code dans un document Rmd, car on ne doit pas compiler le document en entier. Sans utiliser les notebooks, on peut toujours exécuter partiellement des fragments de code, mais l'exécution ne se produit que dans la console R. L'interface des notebooks présente les résultats des fragments de code sous les fragments de l'éditeur, ce qui peut constituer un avantage considérable. Encore une fois, pour des raisons de reproductibilité, on peut éventuellement compiler le document dans son ensemble dans un environnement propre.

Enfin, on peut utiliser la fonction xaringan::inf_mr() pour compiler des documents Rmd:, ou de manière équivalente, l'add-in RStudio "Infinite Moon Reader". Bien entendu , cela nécessite l'installation du package xaringan (Xie 2019 e) , disponible sur CRAN. Le principal avantage de cette méthode est LiveReload: une technologie qui permet de prévisualiser en direct la sortie dès qu'on enregistre le document source, sans avoir à appuyer sur le bouton Knit . L'autre avantage est qu'il compile le document Rmd dans la session R actuelle, cette méthode ne fonctionne que pour les documents Rmd générés en HTML, y compris les documents HTML et les présentations.

Quelques paquets d'extension R Markdown, tels que bookdown et blogdown , ont leur propre manière de compiler des documents.

Il est également possible de générer une série de rapports au lieu d'un seul à partir d'un seul document source R Markdown. On peut paramétrer un document R Markdown et générer différents rapports à l'aide de différents paramètres.