Utiliser Rmarkdown

C. Laroche

04/05/2020

Contents

1 Les bases d'utilisation de R Markdown

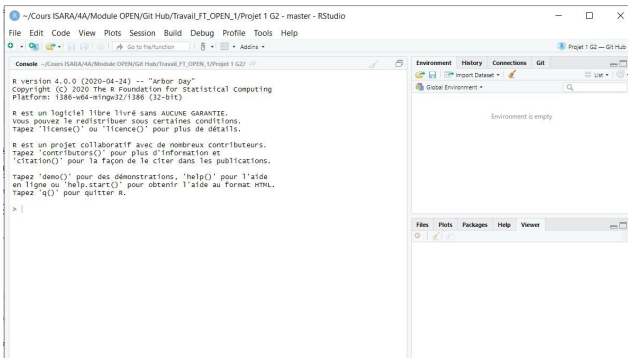
R Markdown est un type de fichier que l'on peut réaliser sous le logiciel R. Il permet de créer des fichiers qui possède à la fois tu texte, des graphiques, des images et du code. On peut soit créer un fichier "simple" qui ne nécessite pas de créer d'autres fichiers, soit de créer un multifichier c'est à dire de créer un fichier qui est composé d'autre fichiers, cela peut-être des fichier R ou des fichiers Rmarkdown par example.La réalisation de multifichier permet de collaborer sur un même projet en permettant à chacun de travailler sur un fichier différent.

La but de ce document n'est pas d'apprendre comment coder sur R mais de comprendre comment réaliser un fichier Rmarkdown ou un multifichier Rmarkdown. Pour cela nous utiliserons le language de programmation R et son environnement de développement Rstudio. On peut télécharger ces deux logiciels sur les sites web ci-après: pour R et pour Rstudio.

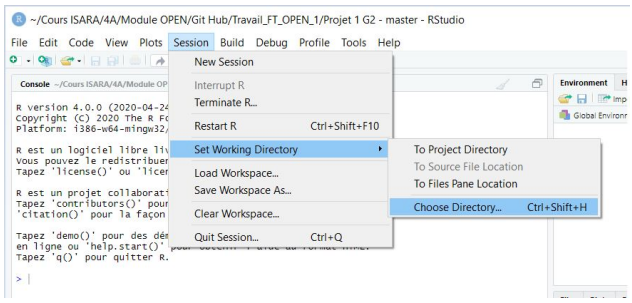
1.1 Se repérer sur Rstudio et créer un fichier Rmarkdown

1.1.1 Rstudio

Lorsque l'on ouvre Rstudio on observe souvent cette vue :



Une bonne habitude à prendre lorsque l'on travail sur R est de créerun dossier spécifique sur notre ordinateur où l'on ajoutera toute les documents dont on aura besoin. Ensuite, il faut choisir ce dossier comme répertoire de travail (en anglais: Working Directory), pour cela il faut cliquer sur l'onglet *Session*, puis sur *Set Working Diretory*, enfin *Choose Directory*, comme sur l'image ci-dessous.



Une pop up va ensuite s'ouvrir et vous pourrez choisir le dossier que vou avez créé comme répertoire de travail. Dans ce dossier il sera important que vous ajoutiez tout les fichiers que vous voudrez utiliser, par example si vous voulez mettre une image il faut qu'elle soit dans le dossier pour que vous puisiez l'utiliser dans votre fichier. Grâce à cette étape vous aurez un dossier organiser où vous pourrez stocker tout les documents annexes qui vont vous permettre de réaliser votre fichier Rmarkdown.

1.1.2 Rmarkdown

Créer un fichier Rmarkdown est très simple, il suffit de cliquer sur l'onglet *File*, puis *New File* et enfin *R Markdown*. Une fenètre va ensuite s'ouvrir, vous pourrez choisir le type de document R markdown que vous voulez créer (document, présentation, application Shiny, un document à partir d'un example) et ajouter le nom du fichier, l'auteur et le type de document final que vous souhaitez (html, pdf, word). Après avoir rempli ces différentes informations en choisissant de créer un document, vous aurrez un text par défaut en anglais vous expliquant rapidement ce qu'est un document R Markdown.

1.2 Les fonctionnalités de R Markdown:

1.2.1 L'en tête

Le début du script ou son en tête est encadré par trois tirets, comme ci-après.

```
---
title: "Utiliser Rmarkdown"
author: "C. Laroche"
date: "04/05/2020"
output: html_document
---
```

On remarque que cette partie contient en faite les métadonnées du fichier (tire, auteur, date, format final du document). C'est ici que l'on pourra modifier la forme global du document, en ajoutant par exemple une table des matières dont on peut choisir le format. Pour ajouter une table des matière il suffit de rajouter `toc: yes` à la ligne avec une tabulation de plus du format de sortie, comme ci-après:

```
output:
  html_document:
    toc: yes
    toc_depth: 3
    toc_float: yes
---
```

La table des matières ou table of content (toc) en anglais, peut être modifié à votre convenance, de nombreuses fonctionnalité existe, comme `toc_depth` qui permet de choisir à quel niveau la table des matières doit s'arrêter, ici on va jusqu'au troisième titre, `toc_float` ne peut être utilisé que sous un format html car cela permet d'avoir un menu déroulant.

Comme R est un language de programmation, il est très important de respecter tout les espaces et saut de lignes pour que le code foncionne.

1.2.2 La partie texte du document

On différencie le texte du code R sur Rstudio par son fond blanc, contre in fond grisé pour le code. Cette partie peut-être mise en page avec des titres, des tirets, des mises en forme pour le texte ou les formules. Etudions quelques examples de fonctionnalité:

Pour écrire en italique il faut mettre * de chaque côté du texte: *italique* Pour écrire en gras il faut mettre ** de chaque côté du texte: **gras** Pour écrire en gras et en italique il faut mettre *** de chaque côté du texte: ***les deux***

`pour mettre du code au sein d'un texte` ajouter “de chaque côté (grâce à AltGr7)

`$A = \pi*r^{2}$` $= A = \pi * r^2$

`$$E = mc^{2}$$` =

$$E = mc^2$$

2 Header1

```
# Header1
```

2.1 Header 2

```
## Header 2
```

2.1.1 Header 3

```
### Header 3
```

- liste non ordonnées
  - sous-item 1
  - sous-item 2
    - sous-sous-item 1
- item 2

1. liste ordonnée
2. item 2
   i) sous-item 1

Pour pouvoir faire une liste à puce, il faut mettre une étoile et un espace avant le texte, comme ceci `*`. Lorsqu'on veut faire des sous puces, il faut rajouter une ou plusieurs tabulation devant l'étoile, celon le nombre de retrait que l'on souhaite. Pour que ces lises à puces soit numérotées, on peut remplacer l'étoile par des chiffres et pour des sous puces, on peut remplacer l'étoile par un plus ou un moins, cela channge le style de la puce.

Que ce soit pour les titre ou pour les listes à puces, il est important de bien noté qu'il faut un espace entre le texte et le dièse ou l'étoile, sinon la mise en forme ne sera pas réalisée.

`<!--commenté votre texte-->` permet de mettre un commentaire sans qu'il soit afficher sur le document final.

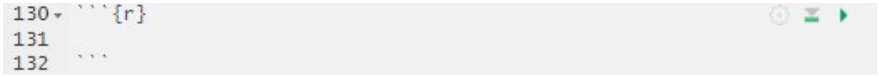
Deux méthodes pour insérer un lien: `<http://www.rstudio.com>` ou `[link](www.rstudio.com)`

Pour ajouter une image: `![Caption](Rstudio.jpg)`

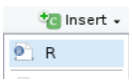
vous pouvez trouvez ces différentes fonctionnalité et d'autres encore à l'adresse: https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf

2.1.2 La partie code R du document ou chunk

Dans un document R Markdown il est possible d'insérer du code au sein d'un texte. Cela permet par exemple d'afficher des graphiques, des tableaux, ou simplement afficher une portion de code pour expliquer comment a été réalisé une anlyse sous R. L'endroit où l'on peut écrire du code dans un document R Markdown s'appel chunk et il ressemble à cela :



Pour génerer un chunk il existe plusieur manière, premièrement vous pouvez rédiger 3 apostrophe AltGr7, r entrecrochet, u saut de ligne et encore 3 apostrophe AltGr7. Deuxièmement, vous pouvez utiliser le raccourci clavier `Ctrl+Alt+i` ou cliquer sur l'icone



La partie `{r}` permet de choisir ce que l'on veut afficher sur le rapport final, elle permet donc de choisir les options des blocs de code R. Lorsqu'on rajoute `echo=FALSE` cela veut dire que l'on ne veu pas afficher le code du chunk, dans le cas où on met `echo=TRUE` cela affichera le code sur le rendu, c'est la valeur qui lui est donné par défaut. L'ajout de `eval=FALSE` entraine la non exécution du code R présent dans le chunk, eval est par défaut vraie. La fonction `include`permet de cummuler les fonctions `eval` et `echo`, c'est à dire d'ajout le code R et ses résultats. Si l'on rajoute `warning=TRUE` dans les options, cela permet d'aficher les avertissements causé par ce bloc, cela permet donc de débuger plus facilement le code qui a été rédigé. L'option `message=TRUE` fait quasiment la même chose que warning sauf que cela affiche les messages générés par le chunk.

Pour plus d'options vous pouvez vous renseigner grâce à l'image suivante:

Chunk options

option	default value	description
Code evaluation		
child	NULL	A character vector of filenames. Knitr will knit the files and place them into the main document.
code	NULL	Set to R code. Knitr will replace the code in the chunk with the code in the code option.
engine	'R'	Knitr will evaluate the chunk in the named language, e.g. **engine = 'python'**. Run **names(knitr::knit_engines$get())** to see supported languages.
eval	TRUE	If **FALSE**, knitr will not run the code in the code chunk.
include	TRUE	If **FALSE**, knitr will run the chunk but not include the chunk in the final document.
purl	TRUE	If **FALSE**, knitr will not include the chunk when running **purl()** to extract the source code.
Results		
collapse	FALSE	If **TRUE**, knitr will collapse all the source and output blocks created by the chunk into a single block.
echo	TRUE	If **FALSE**, knitr will not display the code in the code chunk above it's results in the final document.
results	'markup'	If **'hide'**, knitr will not display the code's results in the final document. If **'hold'**, knitr will delay displaying all output pieces until the end of the chunk. If **'asis'**, knitr will pass through results without reformatting them (useful if results return raw HTML, etc.)
error	TRUE	If **FALSE**, knitr will not display any error messages generated by the code.
message	TRUE	If **FALSE**, knitr will not display any messages generated by the code.
warning	TRUE	If **FALSE**, knitr will not display any warning messages generated by the code.
Code Decoration		
comment	'##'	A character string. Knitr will append the string to the start of each line of results in the final document.
highlight	TRUE	If **TRUE**, knitr will highlight the source code in the final output.
prompt	FALSE	If **TRUE**, knitr will add > to the start of each line of code displayed in the final document.
strip.white	TRUE	If **TRUE**, knitr will remove white spaces that appear at the beginning or end of a code chunk.
tidy	FALSE	If **TRUE**, knitr will tidy code chunks for display with the **tidy_source()** function in the **formatR** package.

Ensuite, à l'intérieur du chunk, vous pouvez écrire du code R comme vous l'auriez fait dans un script R. Vous pouvez donc afficher des tableaux, une parte du tableau, des graphiques, des images, etc.

3 Exemple de citation

Barnier et al. (2020)

(Studio, 2016)

Grolemund et al. (2020)

BARNIER, Julien, BIAUDET, Julien, BRIATTE, François et BOUCHET-VALAT, Milan, 2020. R Markdown : les rapports automatisés. analyse-R. In : [en ligne]. 27 avril 2020. [Consulté le 13 mai 2020]. Disponible à l'adresse : https://larmarange.github.io/analyse-R/rmarkdown-les-rapports-automatises.html#options-des-blocs-de-code-r.

GROLEMUND, Garrett, ALLAIRE, J. J. et XIE, Yihui, 2020. *R Markdown: The Definitive Guide* [en ligne]. S.l. : s.n. [Consulté le 13 mai 2020]. Disponible à l'adresse : https://bookdown.org/yihui/rmarkdown/.

STUDIO, R, 2016. Introduction. In : [en ligne]. 2016. [Consulté le 13 mai 2020]. Disponible à l'adresse : https://rmarkdown.rstudio.com/lesson-1.html.