

Génération de documents avec

xlsx, rmarkdown, knitr (et pandoc)

Mickaël Canouil
mickael.canouil@cnrs.fr

Génomique Intégrative et Modélisation des Maladies Métaboliques (CNRS UMR8199)
Institut de Biologie de Lille

5 février 2016




Sommaire

- 1 Générer un tableur Excel
- 2 Générer un rapport
- 3 References

Sommaire

1 Générer un tableur Excel

Le package xlsx : un peu de lecture (`read.xlsx`)

Le package xlsx propose une fonction de lecture fonctionnant sur le même principe que `read.table`, avec un argument pour sélectionner la feuille Excel .

```
R> library(xlsx)
R> formalArgs(read.xlsx)
[1] "file"          "sheetIndex"    "sheetName"     "rowIndex"
[5] "startRow"      "endRow"        "colIndex"      "as.data.frame"
[9] "header"        "colClasses"    "keepFormulas"  "encoding"
[13] "..."
```

```
R> iris2 <- read.xlsx(file = "iris.xlsx", sheetIndex = 1, header = TRUE)
R> head(iris2)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4          0.2   setosa
2          4.9         3.0          1.4          0.2   setosa
3          4.7         3.2          1.3          0.2   setosa
4          4.6         3.1          1.5          0.2   setosa
5          5.0         3.6          1.4          0.2   setosa
6          5.4         3.9          1.7          0.4   setosa
```

Le package xlsx : un peu d'écriture (`write.xlsx`)

La fonction `write.xlsx` s'utilise de la même façon que `write.table`.

```
R> library(xlsx)
R> formalArgs(write.xlsx)
[1] "x"          "file"        "sheetName" "col.names" "row.names" "append"
[7] "showNA"
```

```
R> write.xlsx(iris2, file = "iris.xlsx", row.names = FALSE)
```

	A	B	C	D	E
1	Sepal.Len	Sepal.Wid	Petal.Len	Petal.Wid	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa

Le package xlsx : les accès

- Créer ou charger un **workbook** (Excel  2007) :


```
R> wb <- createWorkbook(type = "xlsx")  
R> wb <- loadWorkbook(file = "iris.xlsx")
```

L'objet **wb** est un objet S4 (champs accessibles avec @) du package rjava.

- Créer ou charger une feuille Excel  :

```
R> sheet <- createSheet(wb, sheetName = "Sheet1")  
R> sheet <- getSheets(wb)
```

La fonction **getSheets** renvoie un objet de type **list**.

Chaque élément est une feuille Excel .

Le package xlsx : la taille ça compte

Redimensionner automatiquement les colonnes :

```
R> wb <- loadWorkbook(file = "iris.xlsx")  
R> sheet <- getSheets(wb)[["Sheet1"]]
```

```
R> autoSizeColumn(sheet, colIndex = seq(5)) # 5 colonnes dans les donnees iris
```

Ne pas oublier de sauvegarder le **workbook** avec **saveWorkbook**.

```
R> saveWorkbook(wb, file = "iris.xlsx")
```

	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa

Le package xlsx : le style c'est important

Définition et application d'un style de cellule :

```
R> titleStyle <- CellStyle(wb,
  font = Font(wb, isBold = TRUE), # texte en gras
  alignment = Alignment(horizontal = "ALIGN_CENTER") # alignement horizontal du texte
)
```

```
R> rows <- getRows(sheet, rowIndex = 1) # recupere la ligne 1
R> cells <- getCells(rows) # recupere les cellules
R> for (iCell in cells) {
  setCellStyle(iCell, cellStyle = titleStyle) # applique le style a une cellule
}
```

```
R> saveWorkbook(wb, file = "iris.xlsx")
```

	A	B	C	D	E
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5	3.6	1.4	0.2	setosa

Le package xlsx : création et ajout de tableaux

Création d'un fichier Excel contenant deux tableaux dans la même feuille :

```
R> numericSummary <- data.frame(t(apply(iris[, 1:4], 2, quantile, seq(0, 1, 0.25))), check.names
  = FALSE)
R> wb <- createWorkbook(type = "xlsx")
R> sheet <- createSheet(wb, sheetName = "IRIS")
R> titleStyle <- CellStyle(wb,
  font = Font(wb, isBold = TRUE),
  alignment = Alignment(horizontal = "ALIGN_CENTER")
)
```

```
R> addDataFrame(iris, sheet, row.names = FALSE, colnamesStyle = titleStyle) # premier tableau
R> addDataFrame(numericSummary, sheet, startColumn = ncol(iris)+2, colnamesStyle = titleStyle,
  rownamesStyle = titleStyle) # second tableau
```

```
R> autoSizeColumn(sheet, colIndex = seq(ncol(iris)+2+ncol(numericSummary)+1))
R> saveWorkbook(wb, file = "iris.xlsx")
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species			0%	25%	50%	75%	100%
2	5.1	3.5	1.4	0.2	setosa		Sepal.Length	4.3	5.1	5.8	6.4	7.9
3	4.9	3	1.4	0.2	setosa		Sepal.Width	2	2.8	3	3.3	4.4
4	4.7	3.2	1.3	0.2	setosa		Petal.Length	1	1.6	4.35	5.1	6.9
5	4.6	3.1	1.5	0.2	setosa		Petal.Width	0.1	0.3	1.3	1.8	2.5
6	5	3.6	1.4	0.2	setosa							

Sommaire

2 Générer un rapport

La syntaxe Markdown : écrire en Markdown I

La syntaxe Markdown utilise des combinaisons de symboles pour mettre en forme le texte : *****, **-**, **#**, **_**, ...

Texte normal

italique et *_italique_*

****gras**** et **__gras__**

Exposant^2^

[un lien hypertexte](http://www-good.ibl.fr)

Titre 1

Titre 2

Titre 6

Equation: \$A = \pi * r^{2}\$

Image:

Texte normal

italique et *italique*

gras et **gras**

*Exposant*²

un lien hypertexte

Titre 1

Titre 2

Titre 6

Equation : $A = \pi * r^2$

Image : 

La syntaxe Markdown : écrire en Markdown II

```
* Liste à puces
  + item 1
  + item 2
```

```
■ Liste à puces
  ■ sous item 1
  ■ sous item 2
```

```
1. Liste à numéros
  + item 1
  + item 2
```


```
1 Liste à numéros
  ■ sous item 1
  ■ sous item 2
```

```
Colonne 1 | Colonne 2
-----|-----
Cellule 1 | Cellule 2
Cellule 3 | Cellule 4
```

Colonne 1	Colonne 2
Cellule 1	Cellule 2
Cellule 3	Cellule 4

Le package rmarkdown : du dans Markdown

Le script au format Markdown s'écrit dans un fichier `.rmd`.

Le code  s'écrit en ligne ``r`` ou dans un block délimité par ````\{}```` et `````.

```
```\r  
a <- 1 + 1
a + 2
```
```

```
a <- 1 + 1  
a + 2
```

```
[1] 4
```


```
```\r eval=TRUE, echo=FALSE}  
a <- 1 + 1
a + 2
```
```

```
[1] 4
```

Plusieurs options sont disponibles pour gérer les sorties graphiques et les résultats d'exécution :

`dev`, `dpi`, `fig.cap`, `fig.height`, `fig.width`, `fig.align`, `cache`,
`results`, `size`, ...

Le package rmarkdown : compiler un fichier .rmd

La compilation du code  s'effectue avec le package **knitr** (sans **Pandoc**, le Markdown est converti en html).

```
R> library(rmarkdown)
R> knit2html("example.Rmd")
R> render("example.Rmd", output_format = "html_document")
R> render("example.Rmd", output_format = "all") # tout les formats definis dans le fichier
```

```
# Un titre en Markdown
```

```
## Un titre de sous niveau
```

```
Sinon on peut faire des calculs avec **R**
comme 1 + 1 = 'r 1 + 1'.
```

```
*Vous ne croyez tout de même pas que j'ai
fait le calcul moi-même, si?*
```

```
```{r}
a <- 1 + 1
a + 2
```
```

1 Un titre en Markdown

1.1 Un titre de sous niveau


Sinon on peut faire des calculs avec **R** comme $1 + 1 = 2$.

```
\
a <- 1 + 1
a + 2
[1] 4
```

, si ?

Le package rmarkdown : ajouter un thème (Pandoc)

Un thème peut également être appliqué :

- html : feuille de style css
- \LaTeX (pdf / tex) : document format \LaTeX
- Word  (docx) : document Word (provenant de pandoc)

```
# Un titre en Markdown

## Un titre de sous niveau

Sinon on peut faire des calculs avec **R**
comme 1 + 1 = 'r 1 + 1'.

*Vous ne croyez tout de même pas que j'ai
fait le calcul moi-même, si?*
```

```
```{r}
a <- 1 + 1
a + 2
```
```

1 Un titre en Markdown

1.1 Un titre de sous niveau

Sinon on peut faire des calculs avec **R** comme $1 + 1 = 2$.

```
a <- 1 + 1
a + 2
```

```
[1] 4
```

, si ?

Le package rmarkdown : ajouter un thème (YAML)

Les thèmes sont définis au préalable dans des fichiers séparés qu'il faut importer dans le format **YAML** au début du fichier `.rmd`

```
---  
output:  
  word_document:  
    pandoc_args: [  
      "--reference-docx=votreTheme.docx"  
    ]  
  html_document:  
    theme: cerulean <!-- plusieurs theme de couleurs sont disponibles-->  
    css: votreTheme.css  
  pdf_document:  
    template: votreTheme.tex  
  beamer_presentation:  
    template: votreTheme.tex  
---
```

- bootswatch.com : gallerie de thèmes css.
- github.com/jgm/pandoc-templates : thème pandoc par défaut.

Le package rmarkdown : les options Pandoc

Plusieurs paramètres peuvent être définis via l'entête **YAML** pour générer des documents \LaTeX :

- **lang**
- **fontsize**
- **documentclass**
- **geometry**
- **number_sections**
- **toc**
- **keep_tex**
- **pandoc_args**

```
---  
output:  
  pdf_document:  
    number_sections: true  
    toc: true  
    keep_tex: true  
    template: votreTheme.tex  
    pandoc_args: [  
      "--listings" <!-- Utilise l'environnement listings pour le code -->  
    ]  
---
```

Le package rmarkdown : \LaTeX article (pdf)

Un titre en markdown

Un titre de sous niveau

Sinon on peut faire des calculs avec **R** comme $1 + 1 = 2$.

Vous ne croyez tout de même pas que j'ai fait le calcul moi-même, si?

```
a <- 1 + 1  
a + 2
```

```
[1] 4
```

Le package rmarkdown : word (**docx**)

Un titre en markdown

Un titre de sous niveau

Sinon on peut faire des calculs avec **R** comme $1 + 1 = 2$.

Vous ne croyez tout de même pas que j'ai fait le calcul moi-même, si?

```
a <- 1 + 1
```

```
a + 2
```

```
[1] 4
```

Le package rmarkdown : `html`

Un titre en markdown

Un titre de sous niveau

Sinon on peut faire des calculs avec R comme $1 + 1 = 2$.

Vous ne croyez tout de même pas que j'ai fait le calcul moi-même, si?

```
## <- 1 + 1  
## + 2
```

```
[1] 4
```

Le package rmarkdown : L^AT_EX beamer (pdf)

Un titre en markdown

Un titre de sous niveau

Sinon on peut faire des calculs avec **R** comme $1 + 1 = 2$.

Vous ne croyez tout de même pas que j'ai fait le calcul moi-même, si?

```
a <- 1 + 1  
a + 2  
[1] 4
```

Sommaire

3 References

References



Allaire, J., McPherson, J., Xie, Y., Wickham, H., Cheng, J., and Allen, J. (2014).

rmarkdown : Dynamic Documents for R.

R package version 0.4.2.



Dragulescu, A. A. (2014).

xlsx : Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files.

R package version 0.5.7.