

# LateX et divers tips

Laude

06/05/2019

## Formulations mathématiques à l'aide de LaTeX

Afin de permettre de documenter les applications statistiques, de data science ou les papiers de recherche associés, nous vous proposons ici un petit bestiaire *LaTeX* utilisable dans Rmarkdown.

Dans l'exemple de matrice suivant on remarque l'encadrement par 2 dollars pour introduire une expression *LaTeX* sur plusieurs lignes, le dollar seul permettant d'introduire une expression au sein d'une ligne.

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$$

Dont la source est :

```
$$X =  
\begin{bmatrix}  
1 & x_{1}\\  
1 & x_{2}\\  
1 & x_{3}  
\end{bmatrix}$$
```

L'idée n'étant pas ici d'expliquer la syntaxe de ces formulations, mais de permettre au lecteur ayant suivi un petit tutoriel auparavant d'avoir à sa disposition des formulations riches, courantes ou délicates à fabriquer.

## Formulations utiles à étudier

A étudier pour comprendre LaTeX, mais certaines formulations méritent aussi d'entrer dans votre cours de maths ...

### Axiomatique probabilité & Kolmogorov

Espace de probabilité

$$(\Omega, \mathcal{A}, \mathbb{P})$$

```
$$(\Omega, \mathcal{A}, \mathbb{P})$$
```

On remarque une lettre grecque et deux types de polices

$$0 \leq \mathbb{P}(A) \leq 1$$

```
$$0 \leq \mathbb{P}(A) \leq 1$$
```

On remarque le inférieur ou égal  $\leq$

$$\mathbb{P}(\Omega) = 1$$

$$\mathbb{P}(\Omega) = 1$$

Avec une famille d'événements deux à deux disjoints  $A_1, A_2, \dots$

$$\mathbb{P}(A_1 \cup A_2 \cup \dots) = \sum_{i=1}^{+\infty} \mathbb{P}(A_i)$$

$$A_1, A_2, \dots$$

$$\mathbb{P}(A_1 \cup A_2 \cup \dots) = \sum_{i=1}^{+\infty} \mathbb{P}(A_i)$$

On remarque les indices  $A_1$ , les points  $\dots$ , les sommations de 1 à l'infini  $\infty$

$$\mathbb{P}\left(\bigcup_{1 \leq k \leq n} A_k\right) = \sum_{1 \leq k \leq n} \mathbb{P}(A_k)$$

$$\mathbb{P}\left(\bigcup_{1 \leq k \leq n} A_k\right) = \sum_{1 \leq k \leq n} \mathbb{P}(A_k)$$

On remarque les parenthèses  $\left$ , le  $\cup$  en bigcup

Toujours sur l'indépendance

$$\mathbb{P}\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} \mathbb{P}(A_i)$$

$$\mathbb{P}\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} \mathbb{P}(A_i)$$

On remarque signe appartient  $\in$ , le bigcap pour l'intersection

## Matrice

Une grande matrice

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & \dots & a_{2p} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & \dots & a_{np} \end{pmatrix}$$

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & \dots & a_{2p} \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & \dots & a_{np} \end{pmatrix}$$

# Intégration

## Définition

$$\int_a^b f(x) dx = [F(x)]_a^b = F(b) - F(a)$$

$$\int_a^b f(x) dx = [F(x)]_a^b = F(b) - F(a)$$

On remarque signe intégral et sa syntaxe particulière

## Inégalité d'Hölder

$$\left| \int f(x)g(x) dx \right| \leq \left( \int |f(x)|^p dx \right)^{1/p} \left( \int |g(x)|^q dx \right)^{1/q}$$

$$\left| \int f(x)g(x) dx \right| \leq \left( \int |f(x)|^p dx \right)^{1/p} \left( \int |g(x)|^q dx \right)^{1/q}$$

## Inégalité de Minkowski

$$\left( \int |f(x) + g(x)|^p dx \right)^{1/p} \leq \left( \int |f(x)|^p dx \right)^{1/p} + \left( \int |g(x)|^p dx \right)^{1/p}$$

$$\left( \int |f(x) + g(x)|^p dx \right)^{1/p} \leq \left( \int |f(x)|^p dx \right)^{1/p} + \left( \int |g(x)|^p dx \right)^{1/p}$$

## Fonction zéta de Rieman

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

On remarque la fraction  $\frac{1}{n^s}$

## Fonction ELUs (Exponential linear units)

Souvent utilisé comme fonction d'activation dans un réseau neuronal utilisé pour la classification en lieu et place de RELU.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ a(e^x - 1) & \text{otherwise} \end{cases}$$

$a$  hyperparamètre tel que :  $a \geq 0$

```
$$f(x) =
\begin{cases}
x & \& \mbox{if } x > 0 \\
a(e^x-1) & \mbox{otherwise}
\end{cases}$$
```

$a$  hyperparamètre tel que :  $a \geq 0$

On remarque la façon de faire une grande accolade pour un système

---

## FAQ, Divers trucs et astuces

Pour conclure, ce chapitre donne divers savoir-faire complémentaires

## Comment représenter une liste de tâches

La syntaxe suivante n'est pas reconnue et interprétée par Rmarkdown, mais elle est reconnue dans d'autres environnements et c'est donc une bonne habitude que de s'y conformer.

```
- [ ] a bigger project
- [x] first subtask
- [x] follow up subtask
- [ ] final subtask
- [ ] a separate task
```

Donne :

- ☐ a bigger project
  - ☒ first subtask
  - ☒ follow up subtask
  - ☐ final subtask
- ☐ a separate task

---

## Que faire quand on a des problèmes avec le knit vers pdf ?

### Vous constatez des erreurs au moment du knit et le processus s'arrête

C'est souvent normal, pour créer un document "knitable" vers de nombreuses cibles, vous avez été dans l'obligation d'ajouter des contraintes de présentation, typiquement sur les plots ...

*Pas de panique*, il suffit de reprendre le document `.tex` intermédiaire généré, de l'ouvrir et de faire une *compilation pdf* sans tenir compte des erreurs ... un document pdf sera généré.

C'est d'ailleurs le cas avec ce document Rmarkdown ;-)

Voici l'entête YAML qui correspond à ce fichier :

On remarque :

- `keep_tex: true` -> option pour s'assurer que l'on pourra retrouver un `.tex` et le compiler (via un bouton compilation situé à la place où vous trouviez le bouton knit)
- `latex_engine: xelatex` -> ce compilateur est souvent plus permissif
- `df_print: kable` -> pour obtenir de jolies tables

## Cela peut être lié à un problème d'installation de votre contexte *LaTeX*

Essayez :

1. Désinstallez les éventuels gestionnaires de pdf de votre machine
2. installez le package bookdown, qui installe automatiquement le package tinytex R
3. utilisez le package tinytex pour installer automatiquement tinytex sur votre machine

```
tinytex::install_tinytex()  # Nécessite d'avoir installé le package bookdown
                           # installation de tinytext sur votre machine
# négligez les 2 messages d'erreur qui interviennent parfois
# arrêtez et relancez votre session R et vérifiez que tout est ok:
tinytex::is_tinytex()      # vérification que cela c'est bien installé
```

## Comment faire apparaître une table issue d'un dataset

Cette table comprend les premières lignes d'un dataset :

A table of the first 10 rows of the mtcars data.

	mpg	cyl	displacement	hp	drat	wt	qsec	vs
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1

Cette table a le code suivant :

```
knitr::kable(head(mtcars[, 1:8], 10), caption = "A table of the first 10 rows of the mtcars data.")
```

Rq: lors de certains knit, cette syntaxe ajoute une numérotation automatique

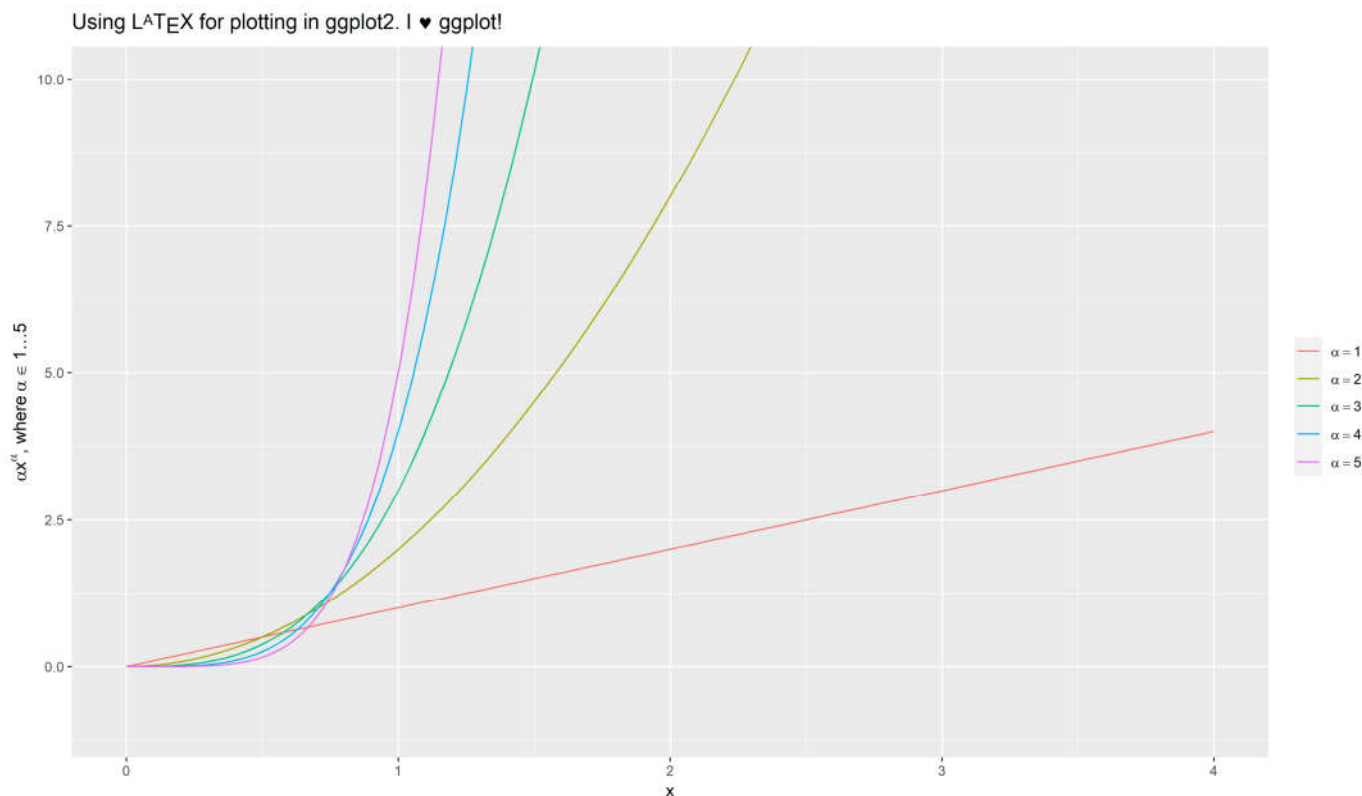
## Utiliser une expression LaTeX dans un graphe

Les utilisateurs de ggplot2 peuvent utiliser la syntaxe suivante très renseignée sur le web (utilisation de latex2exp::TeX)

```
library(plyr)
library(ggplot2)
library(latex2exp)
x <- seq(0, 4, length.out=100)
alpha <- 1:5
data <- mdply(alpha, function(a, x) data.frame(v=a*x^a, x=x), x)

p <- ggplot(data, aes(x=x, y=v, color=X1)) +
  geom_line() +
  ylab(TeX('$\alpha x^\alpha$, where $\alpha \in 1 \dots 5$')) +
  ggtitle(TeX('Using $\\LaTeX$ for plotting in ggplot2. I $\\heartsuit$ ggplot!')) +
  coord_cartesian(ylim=c(-1, 10)) +
  guides(color=guide_legend(title=NULL)) +
  scale_color_discrete(labels=lapply(sprintf('$\alpha = %d$', alpha), TeX))
# Note that ggplot2 legend labels must be lists of expressions, not vectors of expression
s

print(p)
```



# Eviter que le résultat de l'exécution d'un code ne soit précédée de ##

Il faut utiliser l'option de chunk comment = " ". Celle-ci peut être passée en définition par défaut en début du fichier, comme ici, ou dans les options du chunk.

```
print("sans commentaire ;-) ")
```

```
[1] "sans commentaire ;-) "
```

## Récupérer les plots effectués lors du “knit”

Dans une sous-directory, que nous avons ici nommé **figures** vous pouvez trouver vos images à condition d'avoir posé certains paramètres.

Ces paramètres peuvent être posé en entête du chunk, ou comme nous l'avons fait ici dans ce Rmarkdown au niveau général, en début de fichier (après le YAML utilisé pour le knit) :

```
library(knitr)
library(rmdformats)

## Global options
options(max.print="75")
opts_chunk$set(echo=FALSE,
               cache= FALSE,
               prompt=FALSE,
               tidy=TRUE,
               comment=NA,
               message=FALSE,
               warning=FALSE,
               dpi= 200,
               dev = c('png','svg'),
               fig.path= 'figures/')
opts_knit$set(width= '100%' )
```

Ici on a déclaré un chemin pour les images et deux types d'image (png et svg). Les images auront le nom “chunk-numéro”. Le système essaiera de s'approcher de 200 dpi (dot per inch). Les svg permettent d'obtenir une qualité parfaite (création de posters, kadémono ...)

Attention :

- seuls les vrais plots seront générés en svg (comme les plots ggplot2)
- si vous appelez une image toute faite, alors elle ne sera pas régénérée !
- si vous n'avez pas nommé le chunk, alors l'image se nommera “named-chunk-numéro.png”
- quand vous utilisez l'option fig.keep, cela change le comportement de sauvegarde (abstenez vous si vous n'avez pas compris cette option)

# Intégrer un texte issu de l'exécution de R dans le document (sans que l'on remarque que cela vient de R)

## fabriquer du texte Rmarkdown

Il suffit d'utiliser l'option `results='asis'` dans le chunk

Ce texte est généré par R Il comprend une liste à puce qui va fonctionner dans **Markdown** :

- item 1
- item 2

Le code était le suivant :

```
t <- function(x = ""){cat(x, "\n")}
t('Ce texte est généré par R, ')
t('Il comprend une liste à puce qui va fonctionner dans __Markdown__ :')
t()
t('* item 1')
t('* item 2')
t()
```

On remarque que, conformément à la syntaxe Markdown il faut encadrer chaque commande par une ligne vide pour qu'elle soit comprise comme telle.

## Insérer dans un texte le résultat d'un traitement R

Le code suivant instancie 2 variables qui vont être reprises dans le texte Rmarkdown :

```
a <- " __une chose à écrire__"
p <- pi
```

Dans un texte quelconque on peut introduire une expression R, ici : **une chose à écrire** qui est la valeur de  $\pi$  : 3.1415927

L'expression dans Rmarkdown étant :

```
Dans un texte quelconque on peut introduire une expression R, ici `r a` qui est la valeur de
$\pi$ : `r p`
```

---