

Petite introduction à L^AT_EX en Sciences Sociales

Julio Ricardo Davalos

25 mars 2020

Introduction

Voici une petite introduction au monde merveilleux de L^AT_EX. L'idée ici n'est pas de faire un cours parfaitement carré sur l'ensemble des possibilités que ce langage nous offre mais simplement d'offrir une entrée, en douceur, dans un monde dominé par les mathématicien.ne.s et les informaticien.ne.s (i.e. relativement opaque et fermé), aux étudiant.e.s de Sciences sociales, en particulier celles et ceux qui auront besoin de faire de la quantification¹ (même basique). Il ne s'agit absolument pas d'une démarche originale, il existe déjà un livre à ce propos (ROUQUETTE [p. d.](#)). Cependant, il me semble que s'initier avec un livre de 268 pages est une tâche complexe que seul.e.s les plus sûr.e.s d'elles et eux tenteront.

Pourquoi rédiger ses rendus (mémoires, notes de synthèse etc.) en L^AT_EX ? Il y a énormément de raisons, en voici quelques-unes :

- Les logiciels de traitement de textes ont un potentiel de médiocrité conséquente pour certaines opérations. La raison à cela est qu'ils combinent à la fois le fond de votre présentation (sa structure logique, les renvois successifs...) et sa forme (marges, sauts de lignes...) selon le principe du *What You See Is What You Get*. Il est certain que cela facilite la vie dans un premier temps, mais cela se fait au prix, par exemple, de la qualité typographique qui baisse très fortement (gestion de la justification du texte, des espaces de avant/après la ponctuation/etc.). Avec L^AT_EX, la forme est séparée du fond, ce qui donne un rendu plus propre pour moins d'efforts, une fois le coût d'entrée passé bien entendu.
- Toujours à cause du même principe, la gestion de l'automatisation est complexe. Mettons que vous voulez changer un style en particulier d'un titre² ou de votre bibliographie. Il vous faudra réitérer plusieurs fois la même opération. Avec L^AT_EX, la chose se résout généralement en un (1) argument dans une fonction.
- Pour les cas qui nous intéressent ici (les résultats de quantification), on est bien souvent amené à utiliser des logiciels de traitement statistique comme R. En conséquence, une erreur dans un tableau ou un graphique sera fatale à votre présentation Word : il faudra alors changer chaque figure ou tableau incriminé. Avec L^AT_EX, il est possible de réaliser cela automatiquement.

Pour préciser ce dernier point, je m'appuierai ici sur une fonctionnalité de R qui se nomme *R Sweave*. Son intérêt est de permettre la double utilisation R-L^AT_EX sur Rstudio. Cela est possible directement via les packages L^AT_EX, mais Rstudio permet de visualiser les tableaux avant de visualiser le PDF de sortie en entier, ce qui est tout de même plus pratique. Je partirai donc du principe que tout le monde ici possède Rstudio.

Il va de soit que ce document est réalisé avec L^AT_EX.

1. Je précise tout de même que cela ne veut pas dire que les présentations sans quantification n'en ont pas besoin. On peut par exemple faire des schémas comme on peut en trouver en anthropologie, simplement je n'ai pas encore eu l'opportunité d'en réaliser.

2. Word par exemple sait maintenant gérer cette possibilité, mais vous ne serez pas à l'abri d'un bug ou d'une confusion dans votre interface graphique.

1 Installation de L^AT_EX

1.1 Installation sous Mac OS X

Le plus simple est de télécharger T_EX Live. L'ensemble des éléments nécessaires à ce dont on aura besoin ici viendra avec, notamment BibDesk pour la bibliographie. Nous y reviendrons. Voici le lien correspondant : <http://www.tug.org/mactex/>.

1.2 Installation sous GNU Linux

Sous GNU Linux, il suffit de faire la commande suivante :

```
1 sudo apt-get install texlive-full
```

1.3 Installation sous Windows

Windows n'étant pas basé sur Unix, l'installation de T_EX Live est plus fastidieuse. Je vous conseille de simplement installer MikT_EX. Je vous conseille d'installer T_EX Maker ou T_EX Works qui sont des compilateurs de fichiers T_EX afin de pouvoir faire une bibliographie : <https://www.tug.org/texworks>.

2 Bases du langage L^AT_EX

2.1 Notion de package

Si vous êtes familier.ère avec R, la notion de package ne vous est pas étrangère. Dans le langage L^AT_EX, on a la même logique : chaque fonction est définie dans un package que l'on se doit d'appeler avant d'y faire référence. En effet, cela demanderait bien trop de mémoire de charger l'ensemble des fonctions existantes dans la documentation. La différence avec R se situe dans ladite documentation : vous trouverez toujours la réponse à toutes vos questions ! Soit dans la notice de chaque package (que l'on trouve facilement sur internet), soit au pire dans les forums. Vous pouvez également créer vos propres packages, mais ce sera pour une autre fois. Ce qu'il est important de noter donc, c'est que si vous voulez réaliser quelque chose alors quelqu'un.e l'aura très probablement fait avant vous et il vous suffira de charger le package et donc les fonctions désirées.

Contrairement à R, on appelle les packages dans le préambule du fichier T_EX, c'est-à-dire à l'extérieur de notre environnement de texte – on y reviendra. On aura ainsi comme modèle de fichier T_EX par défaut, un code de ce style :

```
1 % Les commentaires sont indiqués par un pourcentage
2
3 % On se trouve ici dans le préambule
4 \documentclass{article} % la classe de document est obligatoire, elle permet àLaTeX de
   charger des styles par défaut. Ici j'ai choisi article mais cela peut être report pour
   un mémoire, ou bien d'autres
5 % On peut créer sa propre classe
6
7 \usepackage{package1}
8 % J'appelle ici \og package1 \fg{}, on verra plus tard lesquels peuvent nous être utiles,
   comme:
9 \usepackage[noae]{Sweave} % noae permet d'obtenir les guillemets àla française
10 \begin{document}
11 % Ici on ne se trouve plus dans le préambule. L'ensemble de ce qui apparait dans le PDF se
   trouve là.
12 \end{document}
```

2.2 Notion d'environnement

Le langage \LaTeX se base sur la notion d'environnement pour fonctionner en séparant la forme du fond. Un environnement est un espace délimité dans lequel certaines règles précises et certaines fonctions ont cours. L'environnement le plus utilisé est, logiquement `document` que l'on vient de voir. Il est obligatoire pour imprimer un PDF. Pour comprendre un peu mieux, prenons l'environnement `equation`. Dans celui-ci, les outils mathématiques – contenus dans le package `amsmaths` – ont un sens.

On fait :

```
1 \documentclass{article}
2 \usepackage{amsmaths}
3
4 \begin{document}
5   \begin{equation}
6     f(x)=\sqrt{x} % racine carrée
7   \end{equation}
8 \end{document}
```

On obtient :

$$f(x) = \sqrt{x} \quad (1)$$

Si j'avais écrit la même formule en dehors de l'environnement, j'aurais eu une erreur de compilation. Par ailleurs, `equation` n'est pas le seul environnement mathématique. Utiliser les signes \$ en début et en fin d'équation permet d'en glisser une (ou un chiffre écrit avec le style « mathématique ») au milieu d'une phrase. Pour écrire n'importe quel caractère spécial comme \$, il faudra qu'il lui précède un antislash \\$.

```
1 \documentclass{article}
2 \usepackage{amsmaths}
3
4 \begin{document}
5 Une phrase lambda $f(x)=\sqrt{x}$
6 \end{document}
```

On obtient : Une phrase lambda $f(x) = \sqrt{x}$.

Cette option nous sera utile pour les nombres décimaux étant donné que les virgules et les points sont automatiquement suivies d'un espace dans \LaTeX .

2.3 Structure d'un document

L'intérêt de \LaTeX réside dans la facilité de structuration d'un texte. On a plusieurs valeurs de titre par défaut, en fonction de la classe de document bien sûr. On compte 7 niveaux :

Commande	Sens	Niveau
<code>\part</code>	Partie	-1
<code>\chapter</code>	Chapitre	0
<code>\section</code>	Section	1
<code>\subsection</code>	Sous-section	2
<code>\subsubsection</code>	Sous-sous-section	3
<code>\paragraph</code>	Paragraphe	4
<code>\subparagraph</code>	Sous-paragraphe	5

La classe `article` n'admet que les niveaux strictement supérieurs à 0. La classe `book` ou `report` commencent à -1. Cela permet d'avoir une déclinaison automatique de vos numéros de sections. Ainsi, si je décide de déplacer la section 56 en position 37, l'arborescence du PDF changera complètement

en fonction. Un couper-coller suffit à tout changer, y compris dans la table des matières

Cette dernière se génère tout simplement avec le code `\tableofcontents` que l'on complète avec la fonction `\setcounter{tocdepth}{n}` dans le préambule si l'on veut changer l'admission des niveaux par défaut. On met alors n le niveau à partir duquel on veut commencer à afficher les numéros des titres dans la table des matières.

2.4 Titre et derniers détails de rédaction

Dans le préambule, on renseigne également le nom de l'auteur, le titre et la date. Le code du document présent comporte les commandes suivantes :

```
1 \documentclass{article}
2 \documentclass[a4paper, 11pt]{article}
3
4 \author{Julio Ricardo Davalos}
5 \title{\textbf{Petite introduction à \LaTeX{} en Sciences Sociales}}
6
7 % packages
8 \begin{document}
9   \maketitle
10  %%% La suite
11 \end{document}
```

Le titre apparaît alors comme vous pouvez le voir plus haut, avec la date de dernière compilation. Si je veux forcer une date, il suffit d'ajouter au préambule `\date{madate}`.

Concernant la rédaction, il y a plusieurs choses à savoir :

- Plusieurs espaces à la suite produisent une seule espace.
- Une ligne vide produit un changement de paragraphe. Plusieurs lignes vides produisent le même effet. Pour redémarrer avec l'indentation, on termine un paragraphe avec `\\`.
- Un retour à la ligne est compris comme une espace.
- Un espace après un signe de ponctuation est obligatoire, mais \LaTeX comprend ce qu'il faut faire avant et insère (ou non) un espace si besoin. Par ailleurs, il vaut mieux ne pas séparer la ponctuation du mot précédent, sans quoi \LaTeX risque de faire apparaître le signe sur la ligne suivante si le mot était le dernier de la ligne.
- quatre caractères spéciaux ne s'obtiennent pas avec `\[mon caractère spécial]` :
 1. `\` : qui s'insère avec `\textbackslash`;
 2. `~` : qui s'insère avec `\textasciitilde`;
 3. `^` : qui s'insère avec `\textasciicircum`;
 4. `« »` : qui s'insère avec `\og \fg{}`.
- L'italique s'obtient en insérant : `\textit{mon texte}`.
- Le gras s'obtient en insérant : `\textbf{mon texte}`.

2.5 Environnements et packages usuels

2.5.1 Environnement tabular

Cet environnement est essentiel pour les Sciences sociales car il permet de générer des tableaux. On le place généralement dans un environnement « flottant » comme `figure` ou `table`, cela permet notamment de lui donner un titre.

```
1 \begin{table}[!h] % permet de forcer la position "here"
2   \begin{center} % je centre dans la page
3     \begin{tabular}{|c|c|} % 2 colonnes, texte centré
```

```

4 \hline %ligne horizontale
5 Case $(1,1)$ & Case $(1,2)$ \\ %saut de ligne, "&" délimite une colonne
6 \hline %ligne horizontale
7 Case $(2,1)$ & Case $(2,2)$ \\
8 \hline
9 \end{tabular}
10 \caption{Un tableau centré} % titre du table, ne marche que dans un flottant
11 \end{center}
12 \end{table}

```

Ce code donne :

Case (1,1)	Case (1,2)
Case (2,1)	Case (2,2)

TABLE 1 – Un tableau vide et centré

Je n'entre pas trop dans les détails pour gagner en concision, mais les arguments de `tabular` peuvent être « c » pour « center », « r » pour « right » et « l » pour « left ». Les lignes de séparation des colonnes ne sont pas obligatoires, ainsi, on peut supprimer la ligne du milieu en indiquant `|cc|`. On peut construire autant de lignes et de colonnes que l'on veut. On peut également faire des `tabular` dans des `tabular`, mais pas de `table` dans un `tabular`.

2.5.2 Environnement figure

Cet environnement est également essentiel car il permet d'ajouter des images, qu'elles soient en PDF ou en format image. Il nécessite le package `graphicx`.

```

1 \begin{figure}[!h] % permet de forcer la position "here"
2 \begin{center} % on centre
3 \includegraphics[width=0.4\textwidth]{Images/Image.jpg} % insertion d'une image
4 % taille = 40% de la largeur d'un paragraphe, on met le chemin de l'image
5 \end{center}
6 \caption{Une image prise au hasard} % titre, bien placé dans un flottant
7 \end{figure}

```

La ligne 3 est elle-même un flottant. Elle peut être utilisée toute seule pour insérer une image mais celle-ci sera placée là où \LaTeX calcule qu'elle a sa place optimale et elle n'aura pas de titre. L'environnement `figure` permet de résoudre ces deux soucis.

Ce code donne :



FIGURE 1 – Une image choisie complètement au hasard

On peut également combiner les deux environnements en faisant des tableaux (avec ou sans lignes) contenant des images.

2.5.3 Environnements `itemize` et `enumerate`

Ces deux environnements sont très importants également car ils permettent de faire des listes. `itemize` permet de faire une suite de tirets (ou autre symbole que l'on choisit) et `enumerate`, une liste numérotée. Ils fonctionnent exactement de la même façon.

```
1 \begin{itemize}
2   \item point 1
3   \item point 2
4 \end{itemize}
```

Ce code nous donne :

- point 1
- point 2

2.5.4 Packages d'encoding et de typographie

Les packages permettant d'écrire en français avec les bons accents et la typographie standard sont les suivants :

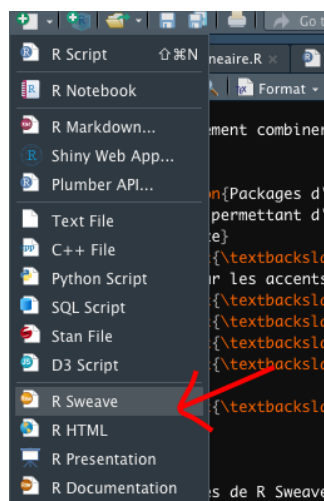
- `\usepackage[utf8]{inputenc}` : permet à \LaTeX de comprendre l'encodage d'entrée, notamment pour les accents.
- `\usepackage[french]{babel}` : permet d'avoir les typographies françaises.
- `\usepackage[T1]{fontenc}` : permet d'encoder le PDF de sortie en T1.
- `\usepackage{hyperref}` : permet de créer des liens hyper-textes.
- `\usepackage[left=cm,right=cm,top=cm,bottom=cm]{geometry}` : permet de définir les marges.
- `\usepackage{tabularx}` : variante de `tabular` avec plus d'options.

3 Bases de R Sweave

Lorsque l'on produit des graphiques et des tableaux à partir de Rstudio, on aimerait pouvoir les mettre directement dans un code \LaTeX . Il est possible d'enregistrer des sorties utilisables par \LaTeX directement et de compiler un document via une autre interface. Le problème, c'est en cas de modification d'une sortie, il faut alors être sûr.e de l'avoir placée dans le bon répertoire et avec le bon nom. R Sweave est donc là pour permettre l'automatisation de l'impression des sorties directement dans un code \LaTeX .

3.1 Interface

R Sweave est une des options de Rstudio. On le trouve comme sur l'image qui suit :



On obtient par défaut la configuration suivante :

```
1 \documentclass{article}
2
3 \begin{document}
4 \SweaveOpts{concordance=TRUE}
5
6 |
7
8
9 \end{document}
```

On retrouve donc notre préambule et notre environnement `document`.

3.2 Chunk

3.2.1 Définition et insertion

Le chunk est l'objet qui permet de faire le lien entre \LaTeX et R. Il s'insère dans \LaTeX via le code suivant :

```
1 \begin{Schunk}
2 \begin{Sinput}
3 > #une commande R
4 \end{Sinput}
5 \end{Schunk}
```

Qui se résume sur Rstudio par : "< < > > =" en début de chaque chunk et @ en fin. Il y a un raccourci en haut à droite du code dans Rstudio.

Les chunks doivent comprendre toutes les commandes du code R dont on a besoin, y compris le chargement de répertoire, des bases de données etc. Il est possible d'en imprimer le contenu, la sortie ou les deux dans le PDF \LaTeX .

3.2.2 Arguments principaux

Il dispose de différents arguments que l'on va utiliser ici :

- `eval=[T ou F]` : permet de lancer le code ou non. *Sa valeur par défaut est T.*
- `echo=[T ou F]` : permet d'imprimer ou non le code R contenu dans le chunk. *Sa valeur par défaut est T.*
- `fig=[T ou F]` : permet d'avoir une sortie de type figure ou non. Cela revient à faire `\includegraphics`, il vaut donc mieux penser à ajouter un environnement `figure` autour comme on l'a vu plus haut, même si cela n'est pas absolument nécessaire. *Sa valeur par défaut est F.*
- `results=[verbatim, tex ou hide]` : permet d'afficher une sortie R paramétrée pour être au langage \LaTeX ou verbatim. On l'utilisera ici pour les `xtable`. *Sa valeur par défaut est hide.*
- `label=` : permet de donner un nom au chunk afin de potentiellement y faire appel plus tard.
- `width=` et `height=` : permet de donner une taille à la sortie figure.

3.3 Quelques opérations

3.3.1 Tableaux

Créons un chunk avec quelques commandes comme suit :

```
3 > rang <- c(1,2,3,4,5,6,7,8)
4 > sexe <- c("F","F","H","F","H","H","F","F")
5 > matrice <- as.data.frame(cbind(rang, sexe))
6 > matrice$age <- c(9,92,29,2,76,25,50,15)
7 > matrice$age_reco <- ifelse(as.numeric(matrice$age) >= 50, "V","J")
```

On a maintenant une matrice. Voyons ce qu'on peut en tirer :

```
3 > library(questionr)
4 > freq(matrice$sexe)
```

On obtient, ce qui suit avec l'option `echo=T` :

```
> library(questionr)
> freq(matrice$sexe)
```

```
  n    % val%
F 5 62.5 62.5
H 3 37.5 37.5
```

Avec l'option `echo=T` et `eval=F` :

```
> library(questionr)
> freq(matrice$sexe)
```

Le résultat n'est pas satisfaisant. Voyons ce qui se passe si on crée un `xtable` :

```
3 > library(xtable)
4 > xtable(freq(matrice$sexe))
5 >
```

Avec `echo=F` et `results=tex`, on obtient :

	n	%	val%
F	5.00	62.50	62.50
H	3.00	37.50	37.50

On peut ajouter un titre et changer le nombre de décimales dans le tableau :

```
3 > library(xtable)
4 > xtable(freq(matrice$sexe),
5 +   caption = "Fréquence des sexes dans notre matrice",
6 +   digits = c(1,0,rep(1,2))) # colonne par colonne : 1 décimale, 0, 1 et 1
```

Avec `echo=F` et `results=tex`, on obtient :

	n	%	val%
F	5	62.5	62.5
H	3	37.5	37.5

Imaginons maintenant que je veuille mettre deux fois un tableau semblable, on va utiliser la propriété des flottants disant qu'on peut les encastrent.

```
1 \begin{table}[!h] %on force la position
2   \centering
3   \begin{tabular}{|c||c|}
4     \hline Sous-table 1 & Sous-table 2 \\
5     \hline
6   \begin{Schunk}
7     \begin{Sinput}
8     > print( #\print nous permet d'avoir un tabular et non un flottant, avec l'option ci-dessous
9 +   xtable(freq(matrice$sexe),
10 +     digits = c(rep(0,2),rep(1,2))), #\nombre de décimales en fonction de la colonne
11 +   floating = FALSE, #\pas flottant
12 +   hline.after = c(0,2)) #\trait après la 1ere ligne et la 3e.
13 \end{Sinput}

```



```

14 \end{Schunk}
15 &
16 \begin{Schunk}
17 \begin{Sinput}
18 > print(
19 +   xtable(freq(matrice$age_reco),
20 +         digits = c(1,0,rep(1,2))),floating = FALSE,hline.after = c(0,2))
21 \end{Sinput}
22 \end{Schunk}
23 \\
24 \hline
25 \end{tabular}
26 \caption{Deux tableaux}%On peut mettre un titre pour l'ensemble
27 \end{table}

```

On obtient alors :

Sous-table 1				Sous-table 2			
	n	%	val%		n	%	val%
F	5	62.5	62.5	J	5	62.5	62.5
H	3	37.5	37.5	V	3	37.5	37.5

TABLE 2 – Deux tableaux

Bien sur, cette option permet d'en combiner d'avantage sur le même principe.

3.3.2 Graphes

Toutes les sortes de graphes sont sorties en figure par Rstudio. On utilise donc l'environnement adéquat :

```

1 \begin{figure}[!h]% on fixe
2   \centering
3   \caption{Un graphique}
4 \end{figure}

```

On a alors, avec l'option `fig=T` et `echo=T`, mais aussi `width=10` et `height=5` :

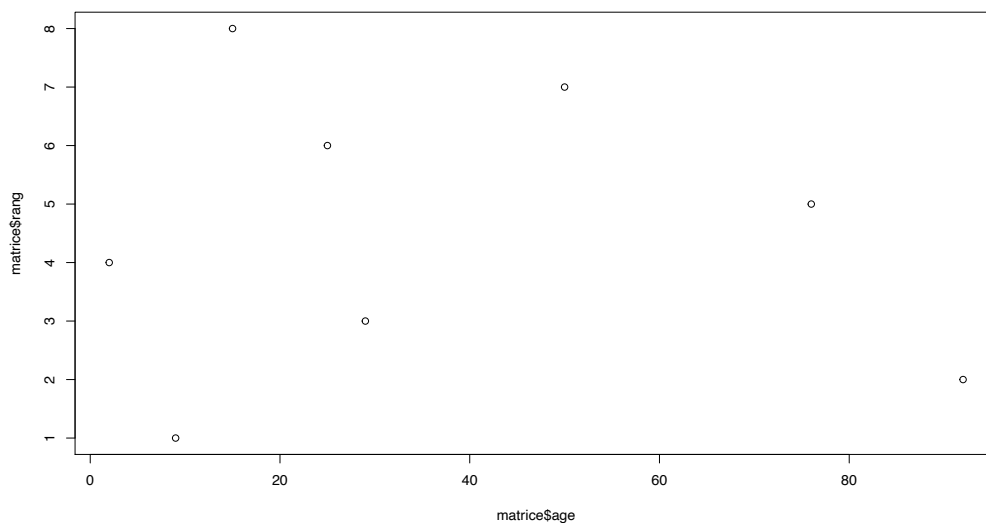


FIGURE 2 – Un graphique

3.3.3 Enregistrement des figures

On peut enregistrer l'ensemble des figures produites au format PDF dans le répertoire dans lequel se trouve le fichier R Sweave. Pour cela, on rentre la ligne suivante dans l'environnement document.

```
1 \begin{document}
2 \SweaveOpts{prefix.string = [Nom de dossier/préfixe du fichier], eps = FALSE, pdf = TRUE}
3 \end{document}
```

C'est à cela que peuvent servir les noms de figure dans le Chunk.

3.3.4 Incorporation des résultats dans le texte

On peut également utiliser la fonction `\Sexpr{}`. Cela s'utilise comme suit :

```
1 \begin{Schunk}
2 \begin{Sinput}
3 > x <- mean(matrice$age)
4 \end{Sinput}
5 \end{Schunk}
6 La moyenne d'âge de l'échantillon est de \Sexpr{x} ans.
```

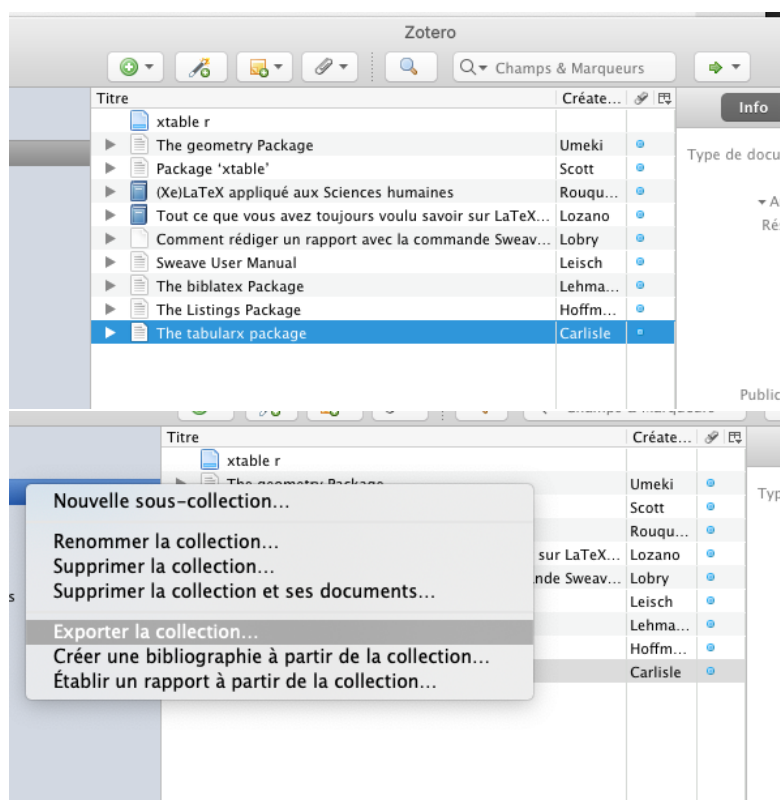
On obtient : « La moyenne d'âge de l'échantillon est de 37.25 ans. ».

4 Gestion de la bibliographie

Je ne pourrai pas être exhaustif sur cette question, tant les options de bibliographie sont nombreuses dans le langage \LaTeX . Je me contenterai de ce que j'ai fait pour le présent document.

4.1 Importation et exportation sur Zotero

Ajoutons d'abord nos fichiers dans Zotero et exportons les dans notre dossier source



4.2 Ajout dans le fichier $\text{T}_{\text{E}}\text{X}$

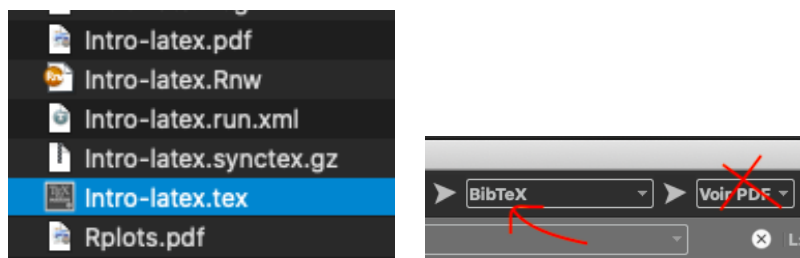
On ajoute les commandes qui suivent :

```
1 % Dans le préambule
2 \usepackage[style=author]{biblatex} % package de bibliographie, on cite l'auteur
3 \addbibresource{Intro-latex.bib} % notre fichier bib exporté
4
5 \begin{document}
6 \nocite{*} % cette commande indique à toutes les entrées d'apparaître dans la bibliographie.
   \cite{clé} permet de citer une entrée particulière. On trouve la clé dans le fichier bib
   que vous pouvez ouvrir avec bibDesk
7 \printbibliography % on imprime la bibliographie
8 \end{document}
```

Notre fichier est prêt. Pour plus de précisions, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ fait plusieurs compilations en fonction des fonctions qui sont demandées. Il y a plusieurs types de compilation et celle qui gère la bibliographie passe par un processus différent de la compilation $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$, il s'agit de $\text{BibT}_{\text{E}}\text{X}$ ou Biber. Malheureusement, Rstudio n'est capable de gérer ni l'un ni l'autre. Il nous faut donc le remplacer.

4.3 Compilation $\text{BibT}_{\text{E}}\text{X}$

On doit maintenant ouvrir le fichier à extension `.tex` situé dans le même dossier que le document Sweave avec un compilateur dont j'évoquais l'existence plus haut (1). On peut ensuite compiler avec $\text{BibT}_{\text{E}}\text{X}$. Ne compilez pas avec un autre processus.



Une fois la compilation terminée. On peut revenir à Rstudio et compiler le PDF. Nous avons enfin une bibliographie complète. Je vous conseille de placer vos références à la fin dans votre texte étant donné que l'opération est fastidieuse, sinon il faudra compiler plusieurs fois le bib.

Références

- CARLISLE, David (p. d.). “The tabularx package”. In : (), p. 12.
- HOFFMANN, Jobst (p. d.). “The Listings Package”. In : (), p. 62.
- LEHMAN, Philipp, Philip KIME et Moritz WEMHEUER (p. d.). “The biblatex Package”. In : (), p. 322.
- LEISCH, Friedrich (p. d.). “Sweave User Manual”. In : (), p. 15.
- LOBRY, Jean R (p. d.). *Comment rédiger un rapport avec la commande Sweave de R*. URL : <http://pbil.univ-lyon1.fr/R/pdf/tdr78.pdf>
- LOZANO, Vincent (2008). *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander 1.0 ou comment utiliser LaTeX quand on n'y connaît goutte*. OCLC : 494797566. Cergy-Pontoise : In Libro Veritas. ISBN : 978-2-35209-149-3.
- ROUQUETTE, Maïeul (p. d.). *(Xe)LaTeX appliqué aux Sciences humaines*. URL : <http://tug.ctan.org/info/latex-sciences-humaines.pdf>.
- SCOTT, David (p. d.). “Package ‘xtable’”. In : (). URL : <http://xtable.r-forge.r-project.org/>.
- UMEKI, Hideo (p. d.). “The geometry Package”. In : (). URL : <http://www.texdoc.net/texmf-dist/doc/latex/geometry/geometry.pdf>.