

LaTeX & R

Comment rendre son travail reproductible

Philippe MICHEL

21 décembre 2023

Tout travail scientifique nécessite beaucoup de travail & de rigueur. De plus il doit être reproductible. La présentation des résultats se fait par un rapport (article, thèse...) qui comporte du texte & des résultats présentés le plus souvent sous forme de tableaux ou de figures & issus de calculs réalisés via un tableur ou un logiciel de statistique. La première solution pour rédiger ce texte est de réaliser les calculs puis de faire des copier/coller des résultats dans le rapport final. Simple, efficace. Mais si vous devez reprendre votre travail six mois plus tard ? On a ajouté dix cas dans l'échantillon, il faut refaire tous les calculs ? Votre travail ponctuel devient une surveillance trimestrielle ? D'expérience il est impossible de refaire des calculs & d'obtenir les mêmes résultats. Certains disent s'en sortir avec des fichiers de scripts bien commentés, rangés dans les bons dossiers & reprendre les copier/coller à chaque nouveauté ? personnellement je n'y suis jamais arrivé sans nuits blanches ou jurons.

Une meilleure solution consiste à n'avoir qu'un seul document qui comporte dans le texte de votre rapport, mise en page comprise, (en LaTeX ou Markdown par ex. voire du Org-mode) des bouts de code (R, python...) pour tous les calculs. C'est le principe développé au début par Xihui XIE dans les packages `knitr` (document LaTeX) ou `Rmarkdown` puis Quarto (document markdown saveur GitHub).



FIGURE 1 – La chaîne de production Quarto

Le document comporte des `chunks` c'est à dire du code (R ou Python par ex.). Une première compilation va interpréter ce code & en sortir un résultat en LaTeX (un tableau...). une seconde compilation classique va générer le document final. Plusieurs formats de sortie sont possibles grâce à l'utilisation de Pandoc. Si actuellement le format de sortie

habituel est l'html la sortie en pdf est bien entendu possible voire du docx. On peut aussi générer des présentations (via Beamer ou autre), des sites web statiques ou des manuels complexes. L'exemple suivant incorpore un diagramme de corrélation dans le document :

```
\section{Contrôles}
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

““{r}
#| label: fig-cor-iris
#| fig-cap: Diagramme de corrélation

rm(list = ls())
data(iris)
plot(iris)
““
```

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

La sortie de la première compilation sera :

```
\section{Contrôles}

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

\begin{figure}[H]
  \centering \includegraphics{tt_files/figure-pdf/setup-1.pdf}
}
\end{figure}

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.
```

Les avantages sont nombreux. tout d'abord, et c'est le plus important, si vous devez reprendre votre travail il suffira de relancer une compilation¹ sur les nouvelles données. En suivant quelques règles (avoir des chunks très courts, nommés...) la gestion du document

1. Ce n'est pas toujours aussi simple, des changements de version des logiciels ou packages peuvent introduire des erreurs ou des différences de résultats dans les calculs. Il existe des méthodes internes dans R pour prévenir ces problèmes ou on utilisera un container.

ou des erreurs, le suivi etc. sont simples & le code bien lisible. On peut choisir au coup par coup entre la puissance de l'html/css pour avoir des documents interactifs ou privilégier la qualité typographique de \LaTeX à partir du même document (enfn presque, ce n'est pas toujours aussi simple).