

Tableaux avec R

Marie Plessz

11 mars 2020

Objectif : “sortir” les résultats de R (et Rstudio) pour pouvoir les communiquer dans un mémo pour soi, un mémoire ou rapport de recherche, une publication scientifique. Trois dimensions importantes du problème:

- esthétique : on veut obtenir des tableaux et graphiques faciles à lire, agréables à l’œil et harmonisés
- rigueur et reproductibilité : éviter les erreurs de copier-coller, faciliter la modification des résultats si les traitements ont été modifiés
- temps : les illustrations peuvent être très chronophages, on veut en faciliter la mise à jour; mais faire des programmes entièrement reproductibles et automatisés peut aussi prendre beaucoup de temps.

Le principe général, l’idéal, serait de sortir de R le plus tard possible (avec le tableau aussi près que possible de sa version finale), et de préparer les tableaux, autant que possible, avec des lignes de codes. Ceci est parfaitement réalisé si vous ne sortez pas du tout de R, en rédigeant la totalité de votre mémoire en Rmarkdown dans Rstudio et intégrant des “chinks” qui calculent les tableaux, illustrations, et même les chiffres cités au fil du texte. Ainsi si vos données changent vous pouvez régénérer rapidement toutes vos illustrations.

En pratique il est parfois très laborieux d’obtenir exactement le tableau qu’on veut (ou le graphique) par des lignes de commande et il est possible d’y passer/perdre énormément de temps. Je présente donc plusieurs stratégies possibles, à adapter en fonction des contextes :

- plus vous risquez d’avoir à refaire vos analyses (article qui sera révisé, rapport à reproduire tous les ans à l’identique, nouvelles données attendues bientôt), plus vous avez intérêt à investir dans les compétences pour automatiser
- à l’inverse, si vous faites un graphique vite fait, juste une fois, des techniques plus “à la main” vous feront gagner du temps.
- quand un tableau ou graphique est réussi on a tendance à s’en resservir...

Interdit : capture d’écran

On voit parfois dans des mémoires des captures d’écran (touche Impr Ecran) donc des images des tableaux des résultats. C’est moche, ça suggère que vous n’êtes pas capable de faire un tableau digne de ce nom, ça vous décrédibilise. Donc à éviter.

***Quick and dirty* : Le copier-coller**

Il est possible de faire des copiers-collers réussis depuis R et Rstudio, à condition de comprendre comment ils fonctionnent. Les fenêtres de R utilisent tous ces polices un peu désuètes qui font penser à des machines à écrire d’avant votre naissance. C’est parce qu’il s’agit de polices “à chasse fixe” - tous les caractères ont la même largeur, le i comme le m. ainsi on peut facilement aligner du texte brut : toutes les colonnes ont une largeur donnée en nombre de caractères.

Dès lors deux solutions:

copier-coller dans un traitement de texte et choisir une police à chasse fixe.

Copiez votre tableau, collez-le dans Word ou Libre-Office et appliquez une police comme **Courier new**. Les caractères vont magiquement s’aligner. si vous le faites souvent vous pouvez créer un style “tableau” dans votre traitement de texte.

C’est une bonne solution si vous devez faire un rapide mémo en .docx ou .rtf, que vous n’êtes pas à l’aise

avec le markdown, et que vous vous en fichez des intitulés des colonnes et lignes, car si vous modifiez leur contenu, vous allez tout décaler !

Attention, partez-bien du premier caractère de la première ligne quand vous sélectionnez pour copier. Attention, si vous utilisez un mode sombre dans Rstudio, il faudra aussi changer les couleurs d'arrière-plan et de police (le plus simple : appliquez le style **sans interligne** puis modifiez la police).

Copier-coller en passant par un tableur

Cela vous permettra de modifier les intitulés de colonne, puis vous pourrez revenir à votre traitement de texte.

Copiez-collez comme précédemment mais cette fois-ci dans un tableur. Si vous utilisez un mode sombre, la police est en blanc et vous ne voyez rien : sélectionnez la colonne où vous avez copié et changez la couleur de police pour du noir.

Tout est collé dans une seule colonne (élargissez-la pour vous en assurer).

Sélectionnez cette colonne puis, dans excel voici la marche à suivre:

- onglet données > convertir
- choisissez **largeur fixe**, cliquez sur suivant
- Excel essaie de deviner où sont les limites de colonnes, en plaçant des séparateurs, qui sont des lignes verticales avec flèche vers le haut. déplacez, supprimez ou créez des séparateur pour recréer vos colonnes. Cliquez sur suivant.
- paramétrez le format des colonnes si vous le souhaitez (texte est recommandé si vous avez des chiffres entre parenthèses car Excel les traite comme des nombres négatifs...). Cliquez sur terminer

Vous pouvez ensuite travailler sur votre tableau : intitulés des lignes et colonnes, nombre de décimales, supprimer des lignes inutiles...

Je passe très souvent par excel pour faire les tableaux finaux d'une publication scientifique car il est rare que je maîtrise toutes les options nécessaires pour obtenir exactement le tableau que je veux dans mon logiciel de statistiques. Mais plus on retouche "à la main", plus on a de travail si on doit refaire le tableau.

Quand je le fais, j'ai un classeur excel dédié à la publication, avec un onglet par illustration, que je nomme (Tableau1, Tableau2 etc). Le nom du classeur contient la version du document (soumission, révision1 etc) et souvent la date.

Quand le tableau est prêt on peut le coller dans le rapport à l'endroit de son choix.

Sans passer par le copier-coller: exporter les résultats depuis R

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
table(iris$Species)
```

```
##
##      setosa versicolor  virginica
##          50          50          50
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.