

Guide des bonnes pratiques en R

Marie Vaugoyeau

Table of contents

Introduction	1
Pour éviter les erreurs	1
Conseil n°1 : Maintenir les packages, R et Rstudio à jour	1
Conseil n°2 : Ne pas enregistrer les données dans un fichier <code>.RData</code> à la fermeture d'une session	2
Conseil n°3 : Apprenez à bien gérer <code>Git</code> avec <code>RStudio</code>	2
Conseil n°4 : Enregistrez régulièrement vos fichiers	2
Conseil n°5 : Attention à la casse	3

Introduction

Le but de ce document est de garder une trace des bonnes pratiques pour coder en R en suivant le cours [Initiez-vous à R pour l'analyse de données](#).

Pour éviter les erreurs

Conseil n°1 : Maintenir les packages, R et Rstudio à jour

Les packages peuvent être mis à jour dans l'onglet **Packages** puis en cliquant sur **Update** ou dans le menu **Tools > Check for Package Update**. **A faire au moins une fois par mois.**

Lorsqu'une nouvelle version de R est disponible, il est conseillé de la télécharger dans un nouveau dossier à part (ce qui est fait par défaut).

Il n'y a aucun problème à avoir plusieurs versions de R sur son ordinateur, il faut par contre

vérifier que la bonne est bien utilisée (premier texte affiché dans la console ou `sessionInfo()` ou `Tools > Global Options > R general > R version`).

Enfin, RStudio informe quand une nouvelle mise à jour est disponible.

Conseil n°2 : Ne pas enregistrer les données dans un fichier .RData à la fermeture d'une session

Comme expliqué dans la session “Ajustez l'apparence de RStudio”, je vous conseille de ne pas enregistrer les données à la fermeture de la session.

Mais si vous voulez que RStudio recharge les données en plus des fichiers ouverts, il faut dans `Tools > Global options > General > Basic > Workspace` cocher `Restore .RData into workspace at startup`.

Vous pouvez aussi sélectionner `Always` pour `Save workspace to .RData on exit`: si vous voulez que vos données soient enregistrées et restaurées automatiquement.

Ces options sont déconseillées aux personnes qui débutent car vous pouvez oublier les modifications déjà réalisées sur les données.

Imaginez que pour votre peinture, vous deviez rajouter un flacon de couleur dans le blanc, à la reprise des travaux vous pouvez vous demander si vous l'avez déjà fait pour ce pot ou si vous devez le faire maintenant.

Une bonne pratique est de relancer les scripts à chaque ouverture de RStudio afin d'être sûre des modifications réalisées sur les données.

Conseil n°3 : Apprenez à bien gérer Git avec RStudio

Pour apprendre à gérer parfaitement Git et RStudio, je vous recommande de lire les ressources :

— [Utiliser Git avec RStudio](#) du projet {utilitR} de *INSEE*

— [Utiliser GIT avec R](#) de Lino GALIANA extrait de Travail collaboratif avec R

— L'article [Travailler avec Git via RStudio et versionner son code](#) du blog de *ThinkR* par Elena SALETTE

Conseil n°4 : Enregistrez régulièrement vos fichiers

Par défaut, le fichier doit-être enregistré pour être compilé donc vous allez enregistrer au moins à chaque compilation avec `Render` mais c'est mieux de le faire plus souvent.

Par rapport à `Git`, il est conseillé de faire un `commit` (enregistrer une version d'un ou plusieurs fichier(s)) avant et après chaque développement. Par exemple, faites un `commit` avant d'importer vos données dans R et après avoir écrit le code et les avoir importées.

Conseil n°5 : Attention à la casse

R comme beaucoup de langage de programmation est sensible à la casse, c'est-à-dire qu'il fait la différence entre majuscule et minuscule. Si vous créez l'objet A puis ensuite que vous écrivez `a * 2`, R affiche une erreur en vous disant que a est un objet inconnu.

```
# assignation d'une valeur à l'objet a
a <- 2

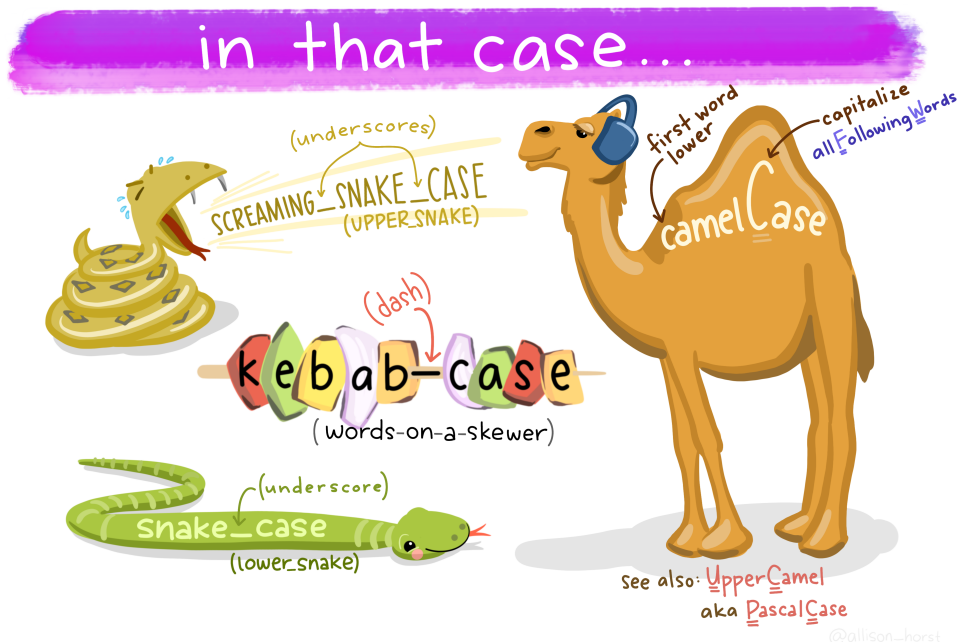
# multiplier par deux
A * 2
```

Error in eval(expr, envir, enclos): objet 'A' introuvable

```
# attention à la casse
a * 2
```

[1] 4

La bonne pratique est d'utiliser les minuscules et d'écrire en `snake_case`, c'est à dire en détachant les mots écrits en minuscules par des underscores.



Artwork by @allison_horst