

Où trouver de l'aide quand on apprend R ?

Marie VAUGOYEAU

Table of contents

Qui suis-je ?	1
Se débrouiller seul.e	2
Bonnes pratiques	2
Utilisation des messages	5
ChatGPT	6
Demander de l'aide	9
De la bonne façon	9
Au bon endroit	12
Ressources pour apprendre R	13
En français	13
En anglais	13
Blogs	14
Merci pour votre attention !	14

Ce support, produit pour [les Rencontres R à Avignon du 21 au 23 juin 2023](#), est mis à disposition selon les termes de la [Licence Creative Commons Attribution 4.0 International](#).

Qui suis-je ?



Marie VAUGOYEAU

Accompagnatrice indépendante à l'analyse de données et la formation au langage R
Dr en biologie évolutive et écologie comportementale

Partage de connaissances (Liens dispo sur linktr.ee/mstats) :

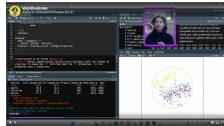
- Auteure de [Langage R et Statistiques](#), Editions ENI



- Rédactrice de la newsletter [Aime les Stats](#) et du blog [Statistiques et R](#)
- Réalisation de [directs sur Twitch](#) pour présenter des packages R et/ou des analyses



statistiques



Se débrouiller seul.e

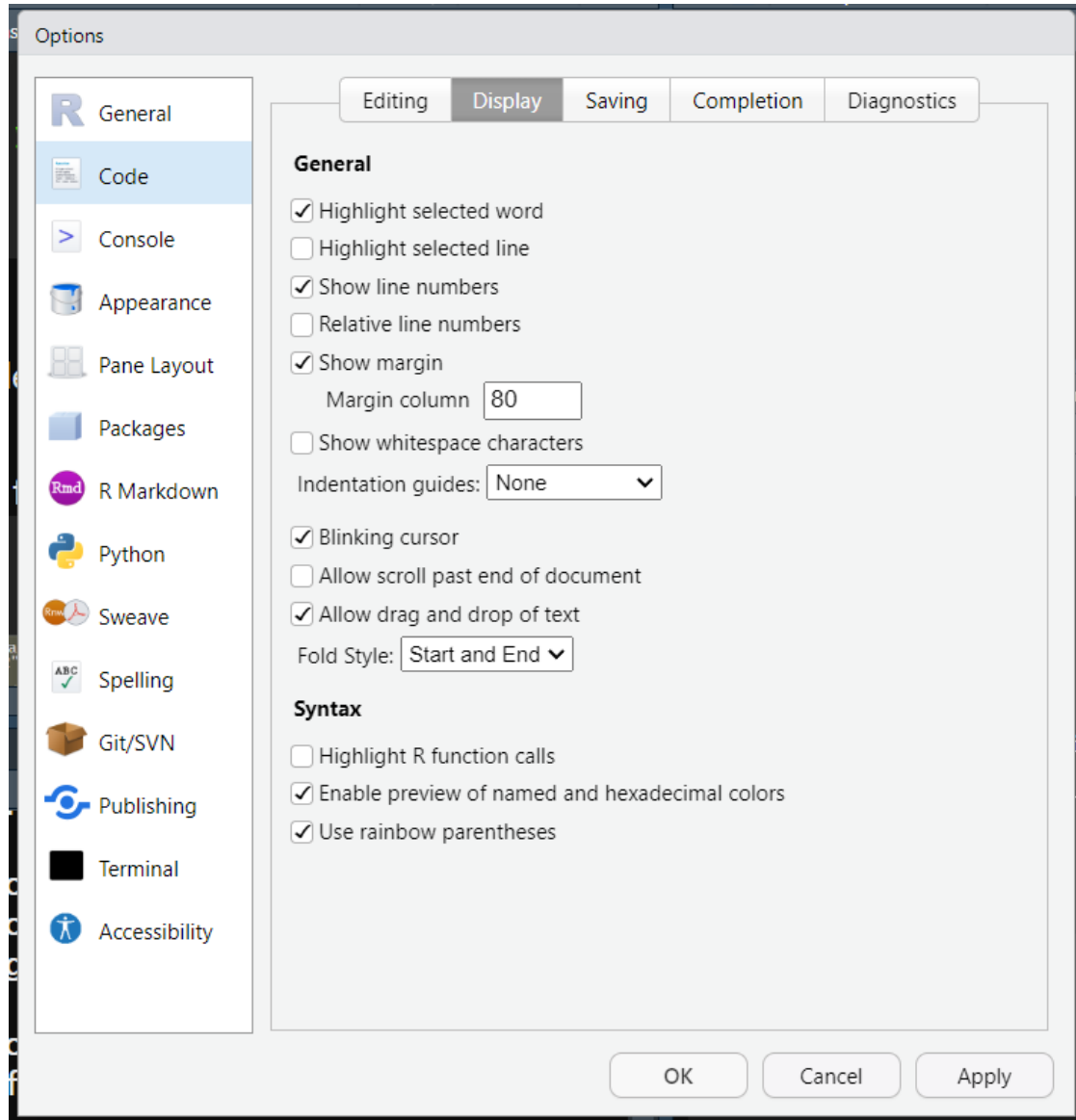
Bonnes pratiques

- Maintenir les packages, R et RStudio à jour (si possible)
- Lire les informations dans RStudio

```
45 iris |>
46   pivot_longer(- species) |>
47   ggplot() +
48   aes(y = value) +
49   geom_boxplot() +
50   facet_wrap(~ name, scales = "free.y"|
51   unmatched opening bracket '('
52 -
53
54
```

- Utiliser des parenthèses colorés pour être sûr de les fermer

Tools > Global Options > Code > Display > Use Rainbow parentheses



- Utilisation de l'auto-complétion pour éviter les fautes

```
45 pivot_longer(~ Species) |>
46 ggplot() +
47 aes(y = value) +
48 geom_bo
49 ◆ geom_boxplot {ggplot2}
50 ◆ geom_ribbon {ggplot2}
51 ● GeomBoxplot
52 ● GeomRibbon
53
54
55
56
57 facet_wrap(~ name, scales
58
```

`geom_boxplot(mapping = NULL, data = NULL, stat = "boxplot", position = "dodge2", ..., outlier.colour = NULL, outlier.color = NULL, outlier.fill = NULL, outlier.shape = 19, outlier.size = 1.5, outlier.stroke = 0.5, outlier.alpha = NULL, notch = FALSE, notchwidth = 0.5, varwidth = FALSE, na.rm = FALSE, orientation = NA, show.legend = NA, inherit.aes = TRUE)`

A box and whiskers plot (in the style of Tukey)

The boxplot compactly displays the distribution of a continuous variable. It visualises five summary statistics (the median, two hinges and two whiskers), and all "outlying" points individually.

Press F1 for additional help

- Utilisation de la page d'aide des fonctions avec l'onglet ou la fonction `help()` et de la vignette, *par exemple* `vignette("pivot")`

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"",
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
  row.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrow = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#",
  allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = FALSE,
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
  dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...)
```

Utilisation des messages

- Messages `simple message()`

```
> library(tidyverse)
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr      1.1.2      ✓ readr      2.1.4
✓ forcats    1.0.0      ✓ stringr    1.5.0
✓ ggplot2    3.4.2      ✓ tibble     3.2.1
✓ lubridate  1.9.2      ✓ tidyr      1.3.0
✓ purrr      1.0.1
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()     masks stats::lag()
i use the conflicted package to force all conflicts to become errors
```

- Messages d'avertissement `warning()`

```
> acp <- PCA(penguins |> select(where(is.numeric)))
Warning message:
In PCA(select(penguins, where(is.numeric))) :
  Missing values are imputed by the mean of the variable: you
  should use the imputePCA function of the missMDA package
```

- Messages d'erreur `error()`


```
> iris |>
+   pivot_longer(- species) |>
+   ggplot() +
+   aes(y = value) +
+   geom_boxplot() +
+   facet_wrap(~ name, scales = "free.y")
Error in `facet_wrap()`:
! `scales` must be one of "fixed", "free_x",
  "free_y", or "free", not "free.y".
i Did you mean "free_y"?
Run `rlang::last_trace()` to see where the error occurred.
```

- Ne pas hésiter à copier-coller les messages d'erreur dans un moteur de recherche pour mieux les comprendre

ChatGPT

- Possibilité d'utiliser le package `{askgpt}`
- Définir facilement une notion




En programmation R, le pipe `%>%` est un opérateur très utile pour chaîner les opérations et faciliter la lecture du code. Il permet d'envoyer le résultat d'une expression à la fonction suivante comme premier argument. 


Voici un exemple simple pour illustrer son utilisation :

Supposons que nous avons un dataframe `df` et que nous voulons effectuer plusieurs opérations successives sur celui-ci, telles que la sélection de certaines colonnes, le filtrage des données, le tri, etc.

Sans utiliser le pipe, le code pourrait ressembler à ceci :

```
R  Copy code  
  
filtered_data <- sort(head(subset(df, select = c("col1", "col2")), 100), by
```

Avec l'utilisation du pipe, le code devient plus lisible et plus concis :

```
R  Copy code  
  
filtered_data <- df %>%  
  subset(select = c("col1", "col2")) %>%  
  head(100) %>%  
  sort(by = "col1")
```

- **Attention** : Le web a été scrappé en septembre 2021 !



Existe t-il un pipe natif dans R ?



À partir de ma connaissance du langage R jusqu'à septembre 2021, il n'y a pas de pipe natif intégré dans le langage R lui-même. Cependant, vous pouvez utiliser des packages externes pour obtenir des fonctionnalités de pipe similaires à celles du pipe `%>%` du package `magrittr`.

- Reformuler pour chercher de l'aide ailleurs



Lorsque vous avez deux variables quantitatives que vous souhaitez analyser ensemble, vous pouvez utiliser plusieurs types de graphiques pour visualiser leur relation. Voici quelques options courantes que vous pouvez réaliser en utilisant le langage de programmation R :

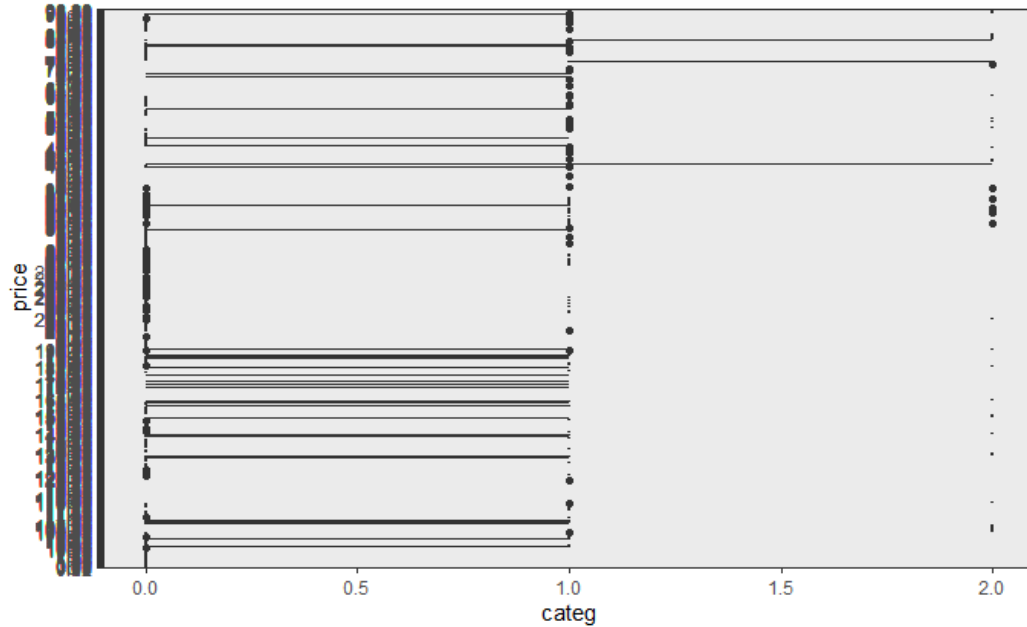
1. Diagramme de dispersion (scatter plot) : Un scatter plot est l'un des graphiques les plus simples et les plus couramment utilisés pour visualiser la relation entre deux variables quantitatives. Chaque point représente une paire de valeurs, une pour chaque variable, et leur position sur le graphique montre leur relation. Vous pouvez créer un scatter plot en utilisant la fonction `plot()` en spécifiant les deux variables que vous souhaitez comparer.
2. Nuage de points avec une ligne de tendance : Si vous souhaitez identifier une tendance ou un modèle dans les données, vous pouvez ajouter une ligne de tendance à votre scatter plot. Cela vous permettra de visualiser la relation générale entre les variables. Vous pouvez utiliser la fonction `geom_smooth()` de la bibliothèque ggplot2 pour ajouter une ligne de tendance à un scatter plot.
3. Graphique en barres (bar plot) : Si vous souhaitez comparer les valeurs moyennes ou les catégories de deux variables quantitatives, vous pouvez utiliser un graphique en barres. Vous pouvez créer un bar plot en utilisant la fonction `barplot()` ou en utilisant la bibliothèque ggplot2 avec la fonction `geom_bar()`.
4. Graphique en aires empilées (stacked area plot) : Si vous souhaitez montrer comment les valeurs de deux variables quantitatives se cumulent ou se combinent au fil du temps ou d'une autre dimension, vous pouvez utiliser un graphique en aires empilées. La bibliothèque ggplot2 offre une fonction `geom_area()` qui vous permet de créer ce type de graphique.

Demander de l'aide

De la bonne façon

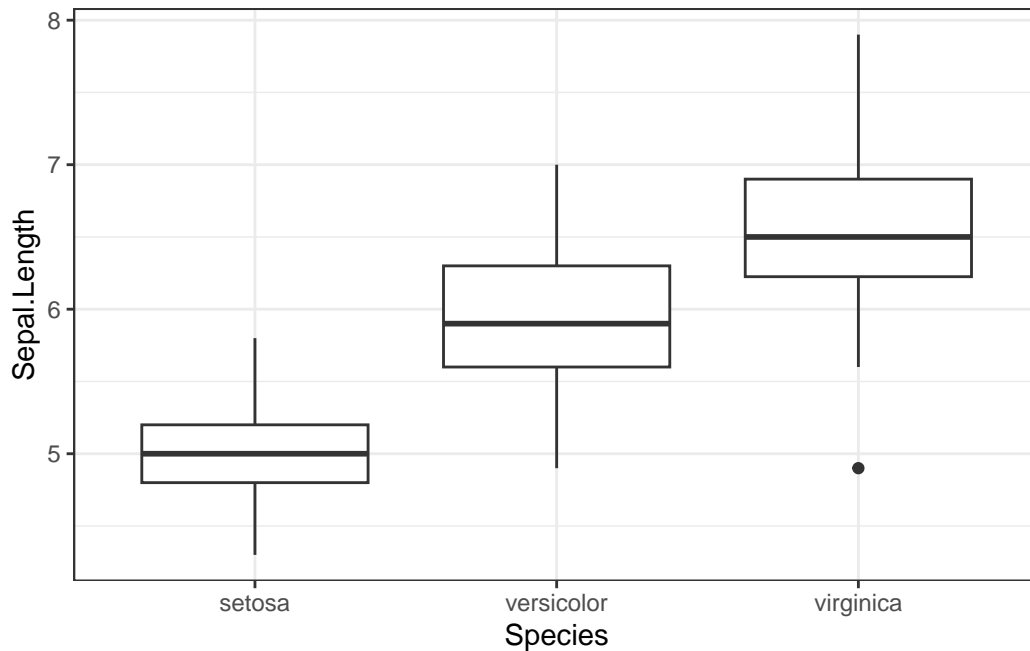
- Donner un contexte et une finalité
Personne n'est dans votre tête !

- Réaliser un exemple reproductible :
 - Ne communiquer pas vos données -> Utiliser un jeu de données qui ressemble au vôtre et est accessible dans R
 - Reproduire l'erreur avec le jeu de données -> Parfois cela suffit à se débloquer
 - Réalisation d'un graphique qui ne fonctionne pas



- Création d'un exemple reproductible

```
library(tidyverse)
iris |>
  ggplot() +
  aes(x = Species, y = Sepal.Length) +
  geom_boxplot() +
  theme_bw()
```



– Mais pourquoi ça fonctionne ?

```
glimpse(iris)
```

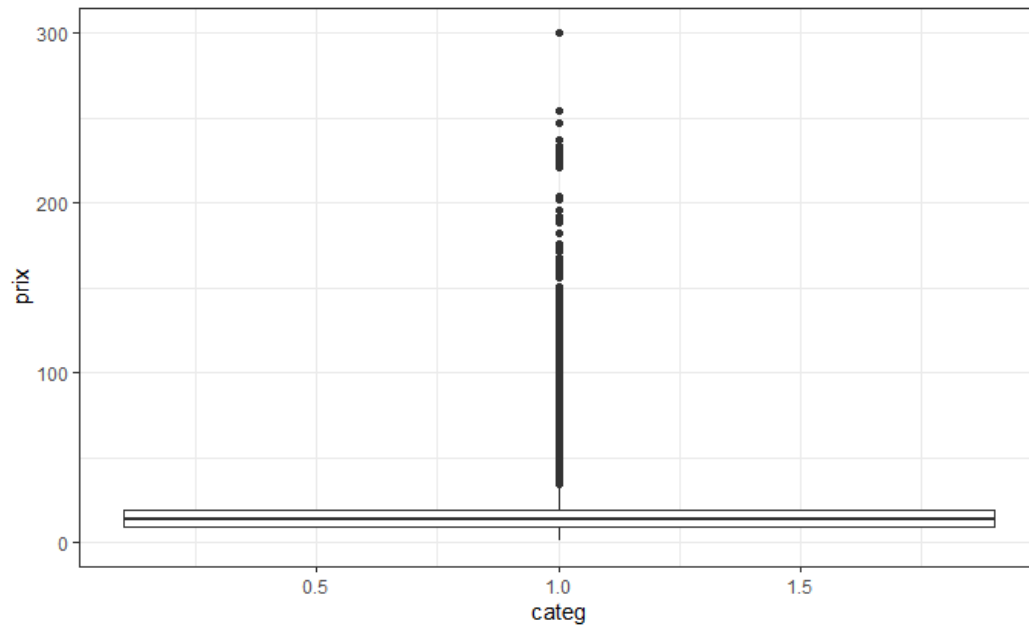
Rows: 150

Columns: 5

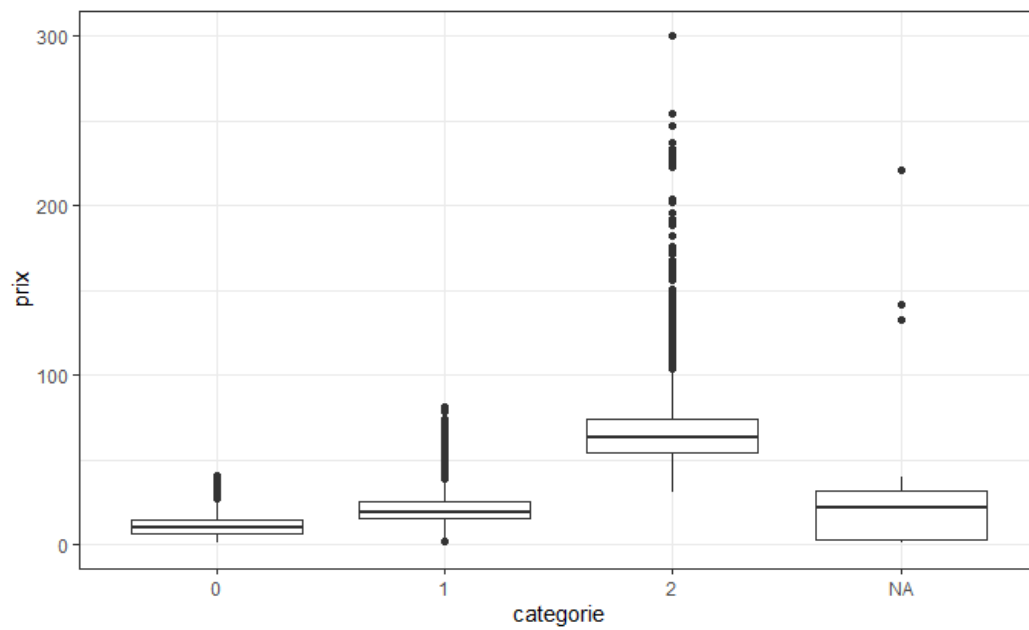
```
$ Sepal.Length <dbl> 5.1, 4.9, 4.7, 4.6, 5.0, 5.4, 4.6, 5.0, 4.4, 4.9, 5.4, 4.~
$ Sepal.Width  <dbl> 3.5, 3.0, 3.2, 3.1, 3.6, 3.9, 3.4, 3.4, 2.9, 3.1, 3.7, 3.~
$ Petal.Length  <dbl> 1.4, 1.4, 1.3, 1.5, 1.4, 1.7, 1.4, 1.5, 1.4, 1.5, 1.5, 1.~
$ Petal.Width   <dbl> 0.2, 0.2, 0.2, 0.2, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.2, 0.~
$ Species       <fct> setosa, setosa, setosa, setosa, setosa, setosa, setosa, s~
```

```
$ price      : chr  "7.2" "25.99" "14.53" "8.99" ...
$ categ      : int   0 1 1 0 1 1 0 1 0 1 ...
```

– Il faut que le format des prix soit continue



– Il faut que la catégorie soit un facteur



Au bon endroit

- Forum : developpez.net, [forum du CIRAD...](#)

- Slack : [GRRR](#)
- Contacter directement une personne (gestionnaire du package, créateur.trice de contenus...)
Attention à bien mettre les formes !
 La plus part des développeur.r.se.s de packages ont prévu une procédure pour déclarer un bug
- Gardez en mémoire que les personnes vous aident **bénévolement** et **gentiment** donc gardez le sourire même si cela peut prendre du temps !
- Pensez à **remercier**

Ressources pour apprendre R

En français

- Cours sur OpenClassRoom (en cours de reprise)
- [Cours *R pour scientifique*](#) offert par le Département de mathématiques et de statistique de l'Université Laval (Québec, Canada)
- [Le projet *{utilitR}*](#) de l'INSEE
- [R pour les débutants](#) par Emmanuel PARADIS
- [guide-R](#) par Joseph LARMARANGE
- [R, Bonnes pratiques](#) écrit par Christophe GENOLINI
- [frrenchies](#)

En anglais

- [The tidyverse style guide](#)
- [Page des *Posit Cheat Sheets*](#)
- [rOpenSci Packages: Development, Maintenance, and Peer Review](#) par rOpenSci
- [R for Data Science](#) écrit par Hadley Wickham et Garrett Grolemund

- [From Data To Viz](#)

Blogs

- [ROpenSci](#) (eng)
- [RStudio](#) (eng)
- [Della Data](#) par [Claire Della Vedova](#) (fr)
- [R-atique](#) par [Lise Vaudor](#) (fr)
- [ThinkR](#) & leur [blogs participatif](#) (niveau plus avancé généralement) (fr)
- [R-bloggers](#) (eng)
- [R Weekly](#) (eng)

Merci pour votre attention !

N'hésitez pas à me contacter en cas de questions, commentaires sur marie.vaugoyeau@gmail.com

Repas R-Ladies demain midi