



HAL
open science

Dans l'air du temps : logiciel R

Sébastien Boutry

► **To cite this version:**

Sébastien Boutry. Dans l'air du temps : logiciel R. École d'ingénieur. Organisation et bonnes pratiques avec R, ENSEGID, France. 2023. hal-04235446

HAL Id: hal-04235446

<https://hal.inrae.fr/hal-04235446>

Submitted on 10 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open licence - etalab



Dans l'air du temps

Logiciel R



Sébastien Boutry

6/3/23



LICENCE OUVERTE
OPEN LICENCE

► Langage interprété

Avantages :

- flexible
- pas de temps de compilation

Inconvénients :

- la vitesse d'exécution

► Un peu d'histoire

Dans les années 1970, le langage **S** a été développé par l'entreprise *Bell Laboratories* ... que l'on retrouve dans le logiciel commercial **S-Plus**.

En 1996, deux chercheurs **Ross Ihaka** et **Robert Gentleman** (?@fig-createurs_R) (Ihaka and Gentleman 1996) proposent un nouveau langage pour les **analyses statistiques** et la **production de graphiques** se basant sur le langage **S** et **Scheme**.



Figure 1: A l'origine du projet R (source : wikipedia)

S... puis R

i le langage R

- ▶ un historique de plus de **27** ans
- ▶ logiciel **gratuit** (fait partie du projet GNU)
- ▶ langage
 - ▶ de programmation complet
 - ▶ un langage **interprété**
 - ▶ basé sur la notion de **vecteur**

i mais aussi

- ▶ un site internet The R Project for Statistical Computing
- ▶ une grande communauté
- ▶ plus de 19159 packages sur le CRAN (50 par défaut)

► Le langage R

Structure de R

- R est un **langage orienté objet**
 - on va donc manipuler des objets
 - qui sont caractérisés par
 - des **attributs** (classes, ...)
 - des **méthodes** (=fonctions)

Fonctions génériques

Certaines fonctions (ex : *print()*, *plot()*, ...) peuvent être utilisées pour différentes classes d'objets.



Les principales classes des objets

- ▶ les vecteurs (classe élémentaire)
- ▶ les facteurs (variable qualitative)
- ▶ les matrices / les tableaux (2 dimensions)
- ▶ les listes
 - ▶ les data.frames (cas particulier)
- ▶ les fonctions
- ▶ ...

Grands principes de R

- ▶ **l'affectation** : on affecte une sortie d'une commande à un objet existant ou nouveau
- ▶ **l'indexation** : simplifie nos extractions à partir des noms de colonnes ou indice mais aussi à partir de conditions logiques
- ▶ **l'approche matricielle** : on privilégie une approche itérative (par boucle) en mettant en place le calcul matriciel (gain de temps)

► Pourquoi utiliser R ?

- **mémoire** des commandes à l'aide d'un script (X au logiciel clic-bouton)
 - automatisation des tâches avec les **fonctions**
 - accès aux lignes de commandes des fonctions
- **grande diversité** de packages
 - variétés des analyses
 - souvent à la pointe des nouveaux développements
- **lisibilité** et **reproductibilité** du code et des traitements
- **gratuit** (licence GNU)
- en **constante évolution**
 - tous les 6 mois pour le logiciel R
 - tous les mois pour de nouveaux packages
- communauté R **dynamique**
 - slack (**groupe Grrr**)
 - réseaux sociaux (twitter **#rstats**)
 - blogs R-Bloggers, ThinkR, DellaData, Analyse-R, R-atique...
 - meetups (**RLadies**,)/ formations
 - frrrenchies



.jpg .gpkg
.img .json .shp
.xls .html
.rds .dat .txt .csv .grd
.zip .sql .xlsx .ods .bib .svg
.tiff .rdata .nc .gpx
.png

► Principes de la programmation

Selon Gilles Hunault, (ancien maitre de conférence de l'Univercité d'Angers),

- "Programmer, c'est **réfléchir et organiser**"
- "Programmer, c'est **choisir et expliciter** ses choix"
- "Programmer, c'est **tester, prévoir et valider**"
- "Programmer demande de l'**endurance**, de la **précision** et de la **rigueur**"



► Bibliographie

Carayon, David. 2023. *InraeThemes: Modèles Et Thèmes INRAE*. <https://davidcarayon.github.io/InraeThemes/>.

Ihaka, Ross, and Robert Gentleman. 1996. "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics* 5 (3): 299-314.



INRAE

Dans l'air du temps

Sébastien Boutry

► Supports pour construire ce diaporama

Les cours de :

- *Frédéric Chiroleu* (chercheur au Cirad de la Réunion)
- *Vincent Goulet* "Introduction à la programmation en R"
- Ewan Gallic

Le thème :

- InraeThemes (Carayon 2023)



Bonnes pratiques

sous R



Sébastien Boutry, Maud Pierre et Eric Quinton

Invalid Date



LICENCE OUVERTE
OPEN LICENCE

► Installation



Suivre les instructions sur le site [Posit](#)

- **R**
- **RStudio** environnement de développement intégré (IDE)

► Travailler sous R

Modules (packages)

Un module rassemble une ou plusieurs fonctions ou jeux de données sur un thème ou une application. On peut trouver les packages sur le site du [RCran](#)

- par thématique [Task Views](#)
- par mots-clés

mais aussi sur la plate-forme [GitHub](#).

Un module **installé, n'est pas chargé**



Rproject

Création

New Project/New Directory or Existing Directoring

Avec les Rprojects, on ne travaille plus qu'en chemin relatif.

Créer votre arborescence de projet

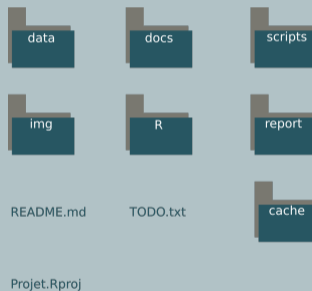


Figure 3: Structure de repertoire (source : Sébastien Boutry)

Raccourcis dans RStudio :

::: {.cell-output-display}

Action	Raccourcis
Assignment	Alt+6
Indentation	Ctrl+I
Pipe	Ctrl+Shift+M
Commenter/Décommenter	Ctrl+Shift+C
Autocomplétion	Ctrl+Espace

Visibilité du code

FiraCode : police

Sous Linux :

- ▶ `sudo apt install fonts-firacode`

Sous Windows :

- ▶ [Lien de téléchargement](#)

INRAE

Bonnes pratiques

Sébastien Boutry, Maud Pierre et Eric Quinton

Les ligatures :

=> =>>

>= >=

!= ≠

== ==

Addin RStudio : Styler (Müller and Walthert 2022)

```
install.packages("styler")
```



INRAE

Bonnes pratiques

Sébastien Boutry, Maud Pierre et Eric Quinton

► Organiser

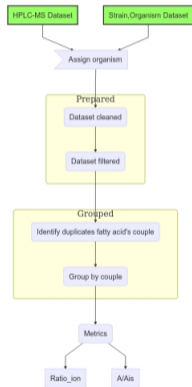
Faire des **schémas** des différentes **étapes** afin de **structurer** son code

- **fragmentation** en plusieurs scripts
- redondance de partie -> **automatisation**

Création d'un workflow

- Graphviz / Draw.io / ... / [Mermaid](#)
- Package : DiagrammeR (Iannone 2022)

```
remotes::install_github("jimhester/DiagrammeR")
```



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

script_Mathilde_29juil.R CuveMacro_Plots_RR app.R Kruskalmc.r CuveMacro_Plots_RR Acp_Rr TestMoySa701.r

```

23
24
25 #----- Saintonge
26 GAF1_70AM=subset(Sa,Campagne=="GAF1"&Frequence=="70"&Direction=="AM")
27 GAF1_AM<-tapply( GAF1_70AM$Stge, GAF1_70AM$Section, mean)
28 GAF1_70AV=subset(Sa,Campagne=="GAF1"&Frequence=="70"&Direction=="AV")
29
30
31 GAF1_AV<-data.frame()
32 for(i in 1:unique(GAF1_70AV$Section)){
33 mydataGAF1_AV=GAF1_70AV[GAF1_70AV$Section==i,]
34 GAF1_AVi[i]<-mean(na.omit(mydataGAF1_AV$Stge))
35 GAF1_AV<-rbind(GAF1_AV, GAF1_AVi)
36 }
37
38 GAF1_70_Stge=data.frame(GAF1_AV,GAF1_AM)
39
40 GAF2_70AM=subset(Sa,Campagne=="GAF2"&Frequence=="70"&Direction=="AM")
41 GAF2_AM<-tapply( GAF2_70AM$Stge, GAF2_70AM$Section, mean)
42 GAF2_70AV=subset(Sa,Campagne=="GAF2"&Frequence=="70"&Direction=="AV")
43 GAF2_AV<-tapply( GAF2_70AV$Stge, GAF2_70AV$Section, mean)
44
45 GAF2_70_Stge=data.frame(GAF2_AV,GAF2_AM)
46
47 GAFInter_70AM=subset(Sa,Campagne=="GAF inter"&Frequence=="70"&Direction=="AM")
48 GAFInter_AM<-tapply( GAFInter_70AM$Stge, GAFInter_70AM$Section, mean)
49 GAFInter_70AV=subset(Sa,Campagne=="GAF inter"&Frequence=="70"&Direction=="AV")
50 GAFInter_AV<-tapply( GAFInter_70AV$Stge, GAFInter_70AV$Section, mean)
51
52 GAFInter_70_Stge=data.frame(GAFInter_AV,GAFInter_AM)
53
54 GAF3_70AM=subset(Sa,Campagne=="GAF3"&Frequence=="70"&Direction=="AM")
55 GAF3_AM<-tapply( GAF3_70AM$Stge, GAF3_70AM$Section, mean)
56 GAF3_70AV=subset(Sa,Campagne=="GAF3"&Frequence=="70"&Direction=="AV")
57 GAF3_AV<-tapply( GAF3_70AV$Stge, GAF3_70AV$Section, mean)
58
59 GAF3_70_Stge=data.frame(GAF3_AV,GAF3_AM)
60
61 GAF3_70_Stge=

```

Environment History Connections

Environment: (Unrtd) R Script

Files Plots Packages Help Viewer

New Folder Delete Rename More

R: 2010 - Alain > 2011 - ECSA - Alain

Name	Size	Modified
..		
avc apc illu.jpeg	22.1 KB	Oct 21, 2011, 9:26 AM
ACP sous R.pdf	461.7 KB	Sep 29, 2011, 5:16 PM
acp.csv	22.6 KB	Sep 28, 2011, 4:53 PM
acp2.txt	22.6 KB	Sep 28, 2011, 4:53 PM
acp_ecsa_alain.xls	110.5 KB	Sep 29, 2011, 3:35 PM
Acp_Rr	16 KB	Oct 21, 2011, 2:07 PM
Acp_R_global.r	16 KB	Oct 21, 2011, 2:14 PM
Acp_R_sans indiv illustratifs.r	17.6 KB	Oct 21, 2011, 6:33 PM
Acp_R_sans indiv illustratifs_global.r	17 KB	Oct 21, 2011, 6:33 PM
cerde corrélation global.jpeg	178.9 KB	Oct 21, 2011, 2:30 PM
cerde corrélation.jpeg	42.1 KB	Oct 21, 2011, 11:22 AM
cerde corrélation_Mortagne.jpeg	31.6 KB	Oct 21, 2011, 1:16 PM
cerdes corr 3d Mortagne.jpeg	148.7 KB	Oct 21, 2011, 1:46 PM
dendo + nuage point.jpeg	84.7 KB	Oct 21, 2011, 10:07 AM
dendo + nuage point_tot.jpeg	95 KB	Oct 21, 2011, 10:56 AM
dendo + nuage point_tot_Mortagne.jpeg	103.4 KB	Oct 21, 2011, 1:16 PM
dendo 1et2.jpeg	53.5 KB	Oct 21, 2011, 10:37 AM
dendo global.jpeg	302.4 KB	Oct 21, 2011, 2:29 PM
Eboulis vp Mortagne.jpeg	34.4 KB	Oct 21, 2011, 1:55 PM
indiv ou var suppl acp ade4.pdf	615.6 KB	Oct 20, 2011, 3:16 PM
Interprétation ACP.pdf	534.8 KB	Sep 29, 2011, 4:06 PM
nuages point 3d Mortagne.jpeg	117.4 KB	Oct 21, 2011, 1:40 PM
nuages point Global-etier vs marais.jpeg	136.4 KB	Oct 21, 2011, 2:50 PM
nuages point Global.jpeg	45.8 KB	Oct 21, 2011, 2:28 PM
Res acp etier avec indiv illustdoc	4.3 MB	Oct 21, 2011, 11:42 AM

Console Terminal

R: 2010 - Alain/2011 - ECSA - Alain

```

> setwd("R:/2010 - Alain/2011 - ECSA - Alain")
>

```

Project: (None)

► Cahier d'analyses (numérique)

- un cahier par projet
- différents formats possibles (.txt, .rmd (rmarkdown), .qmd (quarto))



► Bibliographie

Carayon, David. 2023. *InraeThemes: Modèles Et Thèmes INRAE*. <https://davidcarayon.github.io/InraeThemes/>.

Iannone, Richard. 2022. *DiagrammeR: Graph/Network Visualization*. <https://github.com/rich-iannone/DiagrammeR>.

Müller, Kirill, and Lorenz Walthert. 2022. *Styler: Non-Invasive Pretty Printing of R Code*.

<https://CRAN.R-project.org/package=styler>.

Ushey, Kevin. 2022. *Renv: Project Environments*. <https://CRAN.R-project.org/package=renv>.





Programmation dans R

Les objets dans R



Sébastien Boutry

Invalid Date



LICENCE OUVERTE
OPEN LICENCE

► Quelques recommandations

- L'**affectation**, l'utilisation de `<-` est conseillé pour affecter la sortie d'une commande dans un objet. Le signe `=` marche mais il est préférable de se le garder pour l'utilisation des arguments d'une fonction.
- **Commenter** les scripts, il faut bien sûr commenter le plus possible son script afin de faciliter la transmission.
- Choisir un **convention de nommage** des objets "snake_case" par exemple





Réflexes sous R



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*

Sébastien Boutry

Invalid Date



LICENCE OUVERTE
OPEN LICENCE