

Programmation statistique avec R

Une brève histoire de S et R

J. R. Lobry adapté de Deepayan Sarkar

Université Claude Bernard Lyon I – France

Biologie & Modélisation 2007-2008 (saison 1)


Table des matières

- 1 Les implémentations du langage S
- 2 Historique de R
- 3 Autour de R

Les implémentations du langage S

- 1 Les implémentations du langage S
- 2 Historique de R
- 3 Autour de R

Le langage S

-  , tout comme les logiciels commerciaux S-PLUS[®], R+ et Rpro sont des implémentations du langage de programmation appelé S.
- S a été inventé chez AT&T Bell Laboratories par John Chambers et ses collègues qui faisaient alors de la recherche en statistique sur ordinateur.
- S a ensuite longuement évolué. Deux aspects l'ont toujours distingué des autres logiciels statistiques ; il a toujours été :
 - un système *interactif*
 - un environnement de *programmation* flexible

Utilisation interactive

- S incite fortement l'utilisateur à examiner et analyser ses données de manière interactive, au contraire des logiciels classiques, tels que SAS, qui implémentent un modèle d'analyse en *différé* :
 - L'utilisateur soumet une tâche en fournissant les données et les instructions correspondant à l'analyse à effectuer
 - Le logiciel effectue l'analyse et imprime toutes les informations susceptibles d'intéresser l'utilisateur (et elles peuvent être très nombreuses)
 - L'utilisateur scrute ensuite les résultats pour extraire l'information qui l'intéresse.
- Ce n'est pas forcément un mauvais modèle, et il est utile pour les tâches répétitives. Cependant, c'était insuffisant pour les chercheurs de AT&T Bell Laboratories. L'approche de S est devenue très populaire dans les milieux académiques.

L'utilisation en différé est possible

De plus, S permet également, après une phase d'analyse interactive, d'automatiser les tâches avec la fonction `source()` qui exécute séquentiellement toutes les instructions d'un fichier texte. Supposons que dans le fichier texte `demo.r` il y ait les instructions suivantes :

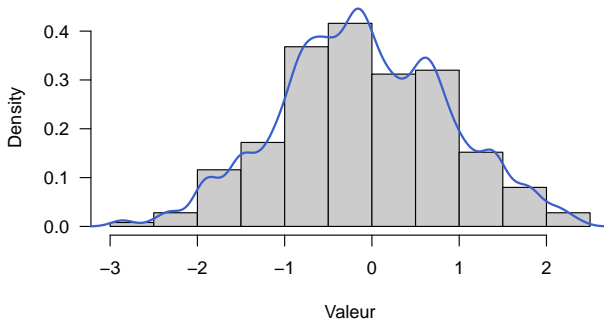
```
cat(readLines("https://pbil.univ-lyon1.fr/R/donnees/demo.r"), sep = "\n")

data <- rnorm(500)
dst <- density(data, adjust = 0.5)
hist(x = data, ylim = c(0, max(dst$y)),
     xlab = "Valeur",
     proba = TRUE, las = 1,
     col = grey(0.8),
     main = "Exemple d'histogramme")
lines(dst$x, dst$y, lwd = 2, col = "royalblue3")
```

L'utilisation en différé est possible

```
source("https://pbil.univ-lyon1.fr/R/donnees/demo.r")
```

Exemple d'histogramme



Une forte extensibilité

- Traditionnellement, les logiciels tels que SAS étaient pensés comme des *boîtes à outils* — Ils avaient un ensemble prédéfini d'analyses pouvant être faites. Bien sûr, les logiciels les plus populaires avaient un ensemble conséquent d'outils prédéfinis.
- Le principal souci lors de la conception de S, au moins au début, a été de faire en sorte qu'il soit facile pour les utilisateurs d'implémenter leurs propres techniques, plutôt que d'assurer la disponibilité de toutes les méthodes potentiellement utiles.
- Ceci a conduit à une grande quantité de bibliothèques développées par les utilisateurs sous S, dont la plupart sont disponibles sur Statlib

<http://lib.stat.cmu.edu/S/>

Frontière floue entre programmeurs et utilisateurs

On peut résumer ainsi la philosophie sous-jacente de S :

"You use S interactively, giving it tasks, looking at data, and creating objects that describe your projects. S can, and is, used in a "non-programming" style, exploiting quick interaction and graphics to look at data. This use often leads to a desire to customize your what you are doing, and S encourages you to slide into programming, perhaps without noticing."

John Chambers, *Programming with Data* (1998)

Un langage prestigieux : S

En 1998, l'Association for Computing Machinery (ACM) récompense John Chambers de son prix prestigieux pour les logiciels pour :

- *the S system, which has forever altered the way people analyze, visualize, and manipulate data ...*
- *le système S, qui a révolutionné la manière dont on analyse, visualise et manipule les données ...*


C'est le seul logiciel de statistique à avoir jamais eu ce prix. Les autres lauréats sont, en autres, les créateurs d'UNIX, du WWW et du langage de programmation Java.


Une implémentation commerciale : S-PLUS®

Depuis 1993, l'implémentation de S des AT&T Bell Laboratories a été exploitée par une société appelée Mathsoft (puis Insightful) qui l'a vendu, avec quelques modifications, sous le nom de S-PLUS®.



Les petits derniers


Le succès de  en tant que logiciel libre de statistiques a incité des entreprises privées a proposer une offre commerciale.

- **Rpro** est une version commerciale de  vendue par REvolution Computing (<http://www.xlsolutions-corp.com/>).
- **R+** est une version commerciale en cours de développement vendue par XL Solutions (<http://www.revolution-computing.com/>).





Historique de R

- 1 Les implémentations du langage S
- 2 Historique de R**
- 3 Autour de R

Conception de

-  a commencé au début des années 1990 comme un projet initié par Robert Gentleman et Ross Ihaka, de l'Université d'Auckland en Nouvelle Zélande, visant à fournir un environnement de statistique pour leur laboratoire. Ils étaient équipés en Macintosh, et il n'y avait pas pour ces machines de logiciel commercial répondant à leurs besoins.
- Comme ils étaient tous les deux habitués à S, ils décidèrent d'implémenter une syntaxe à la S. Une fois le logiciel suffisamment mûr pour sembler utile, ils le déposèrent sur Statlib, où il fut téléchargé par des gens qui l'utilisèrent et fournirent leur impression en retour.

Naissance de en 1995

- Encouragés par Martin Mächler, Ross et Robert décidèrent de diffuser  en tant que *logiciel libre* en juin 1995.
- La disponibilité de  en tant que logiciel libre permet aux utilisateurs d'examiner, de modifier et d'améliorer le code source de , puis de partager ces changements avec les autres.
- La seule restriction est que ces modifications doivent rester dans le domaine du libre. C'est ce qui assure, de manière pérenne, que tous les travaux futurs basés sur  resteront libres.

La licence de

licence()





```
This software is distributed under the terms of the GNU General
Public License, either Version 2, June 1991 or Version 3, June 2007.
The terms of version 2 of the license are in a file called COPYING
which you should have received with
this software and which can be displayed by RShowDoc("COPYING").
Version 3 of the license can be displayed by RShowDoc("GPL-3").
```

```
Copies of both versions 2 and 3 of the license can be found
at https://www.R-project.org/Licenses/.
```

```
A small number of files (the API header files listed in
R_DOC_DIR/COPYRIGHTS) are distributed under the
LESSER GNU GENERAL PUBLIC LICENSE, version 2.1 or later.
This can be displayed by RShowDoc("LGPL-2.1"),
or obtained at the URI given.
Version 3 of the license can be displayed by RShowDoc("LGPL-3").
```

```
'Share and Enjoy.'
```


Constitution du noyau dur

- Aussi étrange que cela puisse paraître, ce modèle de développement a très bien fonctionné pour . Avec l'arrivée d'internet, il fut alors facile pour des personnes géographiquement dispersées de collaborer. Suffisamment de personnes compétentes se sont alors intéressées à  pour l'améliorer.
- En quelques années, la quantité d'améliorations soumises devint trop importante pour être gérée par Ross, Robert et Martin seuls. En 1997, un noyau dur d'une dizaine de personnes ayant le droit de modifier les sources fut constitué. En 2000, John Chambers rallia ce noyau dur.
- Ce modèle de développement de  est toujours d'actualité. Les changements sont rapides, avec une mise à jour majeure tous les 6 mois.  a déjà dépassé S-PLUS® sur bien des points.

Le noyau dur de


La liste des ses membres est donnée dans le fichier "AUTHORS" :

```
auteurs <- readLines(file.path(R.home("doc"), "AUTHORS"))
cat(auteurs[9:25], sep = "\n")
```

```
Douglas Bates
John Chambers
Peter Dalgaard
Seth Falcon
Robert Gentleman
Kurt Hornik
Ross Ihaka
Michael Lawrence
Friedrich Leisch
Uwe Ligges
Thomas Lumley
Martin Maechler
Martin Morgan
Duncan Murdoch
Paul Murrell
Martyn Plummer
Brian Ripley
```

Et bien d'autres encore, voir aussi `contributors()`.

Avantages du modèle de développement de

- Avec une communauté active d'utilisateurs et de développeurs les bugs sont identifiés rapidement.
- Comme le code source est disponible, c'est souvent les utilisateurs eux-même qui localisent les bugs dans le code source et proposent des solutions.
- Ces points, ainsi que d'autre problèmes de développement, sont discutés dans un forum publique de sorte que tous les utilisateurs ont la possibilité de contribuer.
- Une forme d'aide pour  est fournie sous la forme d'une liste de diffusion, où les utilisateurs peuvent poser des questions et d'autres y répondre. Bien que ce soit un forum informel, il y a tant d'abonnés qu'il n'est pas rare d'avoir une réponse dans la minute qui suit.



Différences entre et S

Il y a peu de différences entre  et S.

Ces différences sont détaillées dans la FAQ

(<http://cran.r-project.org/doc/FAQ/R-FAQ.html>)

concernent :

- La portée lexicale des variables (à la Scheme) dans .
- De petites différences dans l'écriture des modèles.
- La correction de quelques comportements indésirables de S dans .

Autour de R

- 1 Les implémentations du langage S
- 2 Historique de R
- 3 Autour de R**



Facilité d'implémentation de nouvelles idées

Pour citer John Chambers, le but de S est :





To turn ideas into software, quickly and faithfully

De transformer rapidement et de manière fiable les idées en logiciels

- le langage S a beaucoup d'outils de haut niveau utilisables de sorte qu'il n'est pas nécessaire de repartir à zéro en permanence.
- les parties gourmandes en temps calcul (qui peuvent être lentes avec l'interpréteur S) peuvent, au prix d'un petit effort, être implémentées dans un langage compilé rapide tel que C ou Fortran.

Pour cette raison, de plus en plus de statisticiens utilisent  pour illustrer de façon reproductible leurs recherches. Beaucoup de packages  disponibles en sus sont écrits par les experts mondiaux de la discipline correspondante.

Le système des packages

- Le système des packages  est similaire à celui de l'archive Statlib pour S-PLUS®.
- Cependant, il est bien plus organisé, avec une structure formelle plus stricte.
- Tous les packages sont régulièrement testés avec la dernière version de  pour garantir que tous les problèmes induits par des changements dans  seront rapidement identifiés et corrigés.
- Il y a aujourd'hui, 19 Feb 2017, *10063* packages disponibles pour 



La liste des packages disponibles

```
n <- 30
dimnames(available.packages())[[1]][1:n]
```

```
[1] "A3"                "abbyR"           "abc"
[4] "ABCanalysis"      "abc.data"        "abcdeFBA"
[7] "ABCOptim"         "ABCp2"           "ABC.RAP"
[10] "abcrf"            "abctools"        "abd"
[13] "abf2"             "ABHgenotypeR"   "abind"
[16] "abjutils"         "abn"             "abodOutlier"
[19] "AbsFilterGSEA"    "AbSim"           "abundant"
[22] "ACA"              "acc"             "accelerometry"
[25] "accelmissing"     "AcceptanceSampling" "ACCLMA"
[28] "accrual"          "accrued"         "ACD"
```

Nous n'avons listé ici que les 30 premiers packages.

Prospective

- Il y a beaucoup de projets, plus ou moins avancés basés sur .
- Parmi eux on peut noter le projet Bioconductor dédié à l'analyse des données post-génomiques.
- Il y a aussi plusieurs projets de construction d'une interface utilisateur graphique multi-plateformes.
- Plusieurs projets visent à utiliser  comme un serveur Web, i.e. comme une application générant des documents sur le Web. On peut citer le projet *Rweb*
`http://www.math.montana.edu/Rweb/` qui est utilisé par exemple ici
`http://biomserv.univ-lyon1.fr/~necsulea/repro/`