

INTRODUCTION À R

Le but de cette séance est d'apprendre à utiliser efficacement le langage R et son environnement. Après avoir appris à effectuer les commandes de base, vous verrez les rudiments de la programmation en langage R, qui est très proche de nombreux autres langages adaptés au calcul matriciel comme Matlab, Octave, Scilab.

Vous pourrez vous aider de l'aide en ligne, ainsi que des documents suivants :

- *** http://zoonex2.free.fr/UNIX/48_R/all.html
- ** <http://cran.r-project.org/>
- ** http://cran.r-project.org/doc/contrib/Paradis-rdebuts_fr.pdf
- ** <http://freakonometrics.hypotheses.org/category/r>
- * http://www.burns-stat.com/pages/Tutor/R_inferno.pdf
- * <http://cran.r-project.org/doc/manuals/R-intro.html>

Remarque : Pour ceux familiers avec Matlab/Octave, une correspondance syntaxique est disponible

<http://mathesaurus.sourceforge.net/octave-r.html>

Parmi les ouvrages publiés sur R, les deux livres en français [1] et [2] peuvent être conseillés.

Enfin les éléments de correction sont disponibles sur le site http://josephsalmon.eu/index.php?page=teaching_13_14&lang=fr sous forme de fichiers R.

- DÉCOUVERTE DE R -

Ligne de commande, aide en ligne et manuel

Basiquement, R a toutes les fonctionnalités d'une calculatrice moderne.

1. Essayer quelques commandes de bases, comme par exemple :

```
a = runif(1); M <- a*matrix(c(1:3,rep(4, 3)), ncol = 3, nrow = 2); ls()
```

2. A l'aide de l'aide en ligne `help('ls')` trouver comment effacer toutes les variables à la fois.
3. Trouver à quoi sert la fonction `mosaicplot`, et lancer les exemples fournis par l'aide (penser à utiliser `require(datasets)` si besoin).

Utilisation d'un éditeur

En matière d'ergonomie, la ligne de commande R montre vite ses limites. Il est indispensable de taper son code dans un autre éditeur, puis de les exécuter grâce à la commande `source` (ou bien en utilisant l'éditeur de texte du logiciel, type Rstudio : on peut alors lancer des parties du code en sélectionnant puis en appuyant sur CTRL+ENTER, ou par le menu Code/Run).

4. Utiliser `getwd` et `setwd`
5. Trouver quelle est la définition de la médiane utilisée par R pour des vecteurs de taille impaire, en utilisant `getS3method` par exemple.

Fonctions

Les fonctions sont en R des objets comme les autres, qui sont affectées de la même façon.

6. Écrire et tester une fonction permettant de calculer la factorielle d'un entier positif (*i.e.*, une fonction qui à n renvoie $n!$). On pourra d'abord écrire une fonction récursive, puis utiliser une boucle.

Gestion des packages

Le chargement d'un package (=module complémentaire) se fait par la commande `require` ou `library`.

7. Charger la fonction `ginv` du package MASS. Pour obtenir plus de détails sur la pseudo-inverse d'une matrice (qui est encore appelée inverse généralisée ou pseudo-inverse de Moore-Penrose) on peut aller voir : <http://fr.wikipedia.org/wiki/Pseudo-inverse>. Numériquement, on pourra vérifier les quatre points définissant cette matrice sur un exemple quelconque.

Graphiques

- La commande de base pour les graphiques est `plot`. Par défaut, elle ne relie pas les points entre eux.
8. Regarder ce que font les commandes `lines` et `abline`. Représenter sin et cos sur le même graphe.
 9. Utiliser `persp` pour afficher la fonction sinus cardinale en 2D (voir aussi la fonction `contour`).
 10. Générer et afficher 100 échantillons suivant des lois normales bi-dimensionnelles (réduites), avec des moyennes (0, 0), (0, 10) et (10, 0).
 11. Illustrer graphiquement la loi forte des grands nombres pour les variables de Bernoulli $B(3/4)$.
 12. Afficher la fonction de répartition théorique et empirique pour des variables gaussiennes centrées réduites. Idem pour la densité. Pour les versions empiriques on fera évoluer le nombre de variables générées.
 13. Afficher une boîte à moustache avec la fonction `boxplot` pour un échantillon de 100 variables gaussiennes.

Gestion des données

En plus des classiques tableaux, vecteurs et matrices, R possède deux structures de données utiles pour manipuler des données numériques : `list` et `data.frame`. Ces données peuvent être chargés grâce à la commande `read.table`. R contient aussi dans sa distribution quelques jeux de données que l'on utilisera pour illustrer les algorithmes vus en cours. Par ailleurs, les données auxquelles il est fait référence dans [3] sont disponible sur le site dans les références.

14. Regarder dans le manuel ce que sont les data frames et comment en charger.
15. Exécuter les commandes suivantes, et comprendre ce qui se passe :

```
data(); attach(cars); plot(speed,dist)
heart_disease=
  read.table("http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data",
  sep=" ", head=T, row.names=1)
plot(heart_disease[,-5], pch = 21,
  bg = c("red", "green", "blue")[as.numeric(heart_disease$famhist)])
```

Quel(s) graphique(s) est-il pertinent de faire pour les représenter? Lancer le même genre de commande sur les données "iris" (qui sont déjà disponible dans R).

16. La régression linéaire est gérée grâce à la commande `lm`. Exécuter alors les commandes suivantes

```
attach(cars); summary(cars); plot(speed,dist)
reglin<-lm(dist ~ speed); abline(reglin,col = "red" ); summary(reglin)
```

BONUS : API

17. Récupérer sur le site http://josephsalmon.eu/index.php?page=teaching_13_14&lang=fr les fichiers `loisAPI.R` et `simpleRegAPI.R`; le lancer dans R et regarder ce qu'il contient.

Références

- [1] P-A. Cornillon, A. Guyader, F. Husson, N. Jégou, J. Josse, Maela Kloareg, E. Matzner-Løber, and L. Rouvière. *Statistiques avec R*. Didact Statistiques. Presses Universitaires de Rennes, 2010. 1
- [2] P-A. Cornillon and E. Matzner-Løber. *Régression avec R*. Springer, Collection Pratique R, 2011. 1
- [3] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>. 2