Introduction aux fonctions du langage R

Pierre Legendre
Département de sciences biologiques
Université de Montréal

Octobre, décembre 2004
Mai, septembre 2005, janvier 2006
Août 2006, novembre 2006, février 2008

R est un système (progiciel) d'analyse statistique et graphique, disponible gratuitement pour toutes les plateformes (Windows, Mac OS X, Linux). R a été créé en 1990 par Ross Ihaka et Robert Gentleman à *University of Auckland*. R est un dialecte du langage S développé en 1976 par John Chambers et ses collègues de *AT&T Bell Laboratories*. Le langage S est disponible sous la forme du logiciel S-PLUS commercialisé en 1993 par la compagnie *Insightful* (anciennement *MathSoft*). R est devenu un logiciel libre et gratuit en 1995.

R est à la fois un langage de programmation et un progiciel de fonctions statistiques. La version de base de R contient déjà un grand nombre de fonctions statistiques et graphiques permettant, par exemple, de calculer une moyenne ou une variance ou de tracer un histogramme. De nombreux chercheurs ont développé au cours des années des fonctions plus avancées qui sont disponibles à tous les utilisateurs de R. Ces fonctions sont regroupées en bibliothèques qui sont disponibles pour téléchargement sur le site du projet R (http://www.r-project.org/), ou encore sur la page Web des chercheurs ou en annexe de leurs publications. À cause de sa grande flexibilité et du fait qu'il soit multi-plateforme et gratuit, R est devenu l'un des principaux instruments de calcul statistique utilisé par les chercheurs dans tous les domaines.

1. Installer R sur un ordinateur

Voici les étapes à suivre pour installer R sur un ordinateur.

• Aller à l'adresse http://cran.r-project.org/ sur l'Internet. Ce site est logé à l'*University of Technology* (*Technische Universität*) à Vienne. Choisir un ordinateur-miroir du CRAN.

• Choisir la version désirée dans la section « *Download and Install R* ». Suivre les instructions. Pour les machines opérant sous **Windows**, choisir "base" puis télécharger l'exécutable « R-2.6.2-win32.exe" ». Pour **Mac OS X**, télécharger le disque-image « R-2.6.2.dmg ». Les bibliothèques additionnelles se trouvent dans le sous-répertoire « Packages » du répertoire « Software » à la gauche de la page principale. Ces informations seront susceptibles de changer lors de l'apparition de nouvelles versions de R. Des versions sont également disponibles pour Mac OS 9.x (version fossile), Linux et d'autres plateformes Unix.

Des manuels sont disponibles dans la section *Documentation => Manuals => Contributed documentation*, y compris le guide *R pour débutant* d'Emmanuel Paradis (Université Montpellier II). La documentation interne de R est en anglais.

2. Le langage R et ses commandes

Lorsqu'on démarre R, une fenêtre appelée « console » apparaît à l'écran. On tape une ligne de commande dans la fenêtre ; les commandes sont exécutées lorsqu'on appuie sur la touche « retour ».

Le contenu de la console peut être sauvegardé dans un fichier, par exemple « RConsole.txt ». Plus souvent, les utilisateurs de R sauvegardent leurs commandes en cours de travail par copier-coller dans un fichier texte. On peut par la suite recopier des portions de ce fichier-texte (une ou plusieurs lignes à la fois) vers la console afin d'exécuter à nouveau les commandes.

Attribuer un nom à un objet

Les objets utilisés dans le cadre de ce cours se limiteront aux vecteurs (« vector »), aux matrices (« matrix ») et aux « data.frame » (forme de matrice contenant plusieurs types de vecteurs). Dans le langage R, un nombre est un vecteur contenant un seul élément.

Pour réaliser les différentes analyses statistiques, nous utiliserons des fonctions du langage R. Certaines de ces fonctions permettent de calculer la moyenne par exemple, tandis que d'autres permettent de construire des graphiques.

La fonction « c » (« combine ») qui permet de combiner des nombres pour former un vecteur. Par exemple, si on veut construire un vecteur « vec1 » contenant les nombres 1,2 et 3, la ligne de commande est la suivante :

```
> vec1 <- c(1, 12, 7)
```

Le « > » en début de ligne est le caractère de sollicitation de R. Ce signe indique que R est prêt à recevoir une commande. Pour donner un nom à un objet, il suffit d'utiliser l'opérateur « assigner ». Cet opérateur est constitué de « < » suivi de « - » :

```
> vec1 <- c(1, 12, 7)
```

Une commande peut être écrite avec ou sans espaces. « = » est un autre opérateur d'assignation :

```
> vec1 = c(1, 12, 7)
```

Il est important de comprendre la différence entre le nom d'un objet et la fonction qui permet de le calculer. Par exemple, supposons qu'on désire calculer la moyenne des valeurs qui se trouvent dans un vecteur (objet) « toto ». On importe l'objet dans le langage R à l'aide de « read.table » :

```
> toto <- read.table("essai1.txt", header=T, row.names=1)
```

« read.table » est une fonction qui permet d'importer dans R des données qui se trouvent dans un fichier (son nom est « essai1.txt » dans cet exemple) situé sur le disque dur de l'ordinateur. « toto » est le nom que l'on assigne, dans la tâche R en cours, au tableau de données lues à partir du fichier. Puis on calcule la moyenne des valeurs qui se trouvent dans « toto » :

```
> moyenne.de.ces.valeurs <- mean(toto)
```

« mean » est une fonction de R; elle permet de calculer la moyenne des valeurs qui se trouvent dans l'objet « toto » que l'on fournit comme argument à la fonction. Le résultat se trouvera dans un nouvel objet appelé « moyenne.de.ces.valeurs ». Lors de l'appel d'une fonction, les arguments sont placés entre parenthèses, y compris le nom des objets utilisés par la fonction.

Important : « toto » et « moy1 » sont des noms choisis de façon arbitraire par l'utilisateur. Ils désignent des objets dans la tâche R en cours. Par contre, « read.table » et « mean » sont des fonctions qui permettent de réaliser des calculs. On ne peut ni changer ni traduire leur nom.

Le nom d'un objet **commence obligatoirement par une lettre** ; il peut comporter des lettres, des chiffres (0-9) et des points (.), mais aucune espace. Il faut savoir aussi que, pour les noms d'objets, R distingue les majuscules des minuscules ; ainsi, x et X pourront servir à nommer des objets distincts (voir le manuel *R pour les débutants* par Emmanuel Paradis).

Information sur les fonctions de R

On peut obtenir des informations sur les fonctions de deux façons différentes :

```
> ?nom-de-la-fonction
```

Une fenêtre contenant une description de la fonction apparaît à l'écran. Par exemple, pour obtenir une description de la fonction « mean », on tape :

```
> ?mean
```

Si on ne connaît pas le nom de la fonction, on peut utiliser la commande « help.search ». Cette commande fouille les bibliothèques R de votre ordinateur et produit une liste de toutes les fonctions de R dont la description contient l'expression ou le terme indiqué ; le nom de la bibliothèque R où se trouve cette fonction est aussi indiqué. On peut taper par exemple :

```
> help.search("median")
```

Les fenêtres d'aide en ligne sont en anglais.

Notes et commentaires

Il est important d'ajouter des commentaires au fichier texte contenant les commandes sauvegardées, afin de se rappeler plus tard de l'objectif des calculs ou encore la raison de telle ou telle façon de faire. Une ligne de commentaires commence par le caractère « # ». Exemple :

```
> # Calcul d'une régression multiple
```

Lorsqu'on les copie dans la console R, les lignes de commande ou les sections de lignes commençant par « # » ne sont pas exécutées par R.

Importer un fichier de données en vue d'une analyse par R

On pourrait taper à l'écran toutes les données nécessaires au calcul, mais il est beaucoup plus pratique de les inscrire à l'avance dans un fichier texte ou un tableur, puis de les importer dans R. La fonction « read.table » permet d'importer des données **depuis un fichier texte** et de créer un objet de type « data.frame ». (Pour des tableaux très grands, il est préférable d'employer la fonction « scan » plutôt que « read.table ».) Exemple :

```
> toto <- read.table("nom-du-fichier-texte", header=T, row.names=1)
```

« toto » est le nom attribué dans la console R au **data.frame** contenant les données. Le nom-du-fichier-texte est le nom du fichier texte sur le disque dur de votre ordinateur, y compris l'extension si le nom du fichier en comporte une (attention : certaines versions de MS Windows n'affichent pas automatiquement les extensions à l'écran).

Avant d'importer les données, il faut indiquer à R où se trouve le fichier. Il faut aller dans le menu « File », « Tools » ou « Misc. », selon la version de R, puis choisir « Change Working Directory ». On doit naviguer sur le disque et choisir le nom du **dossier** contenant le fichier à importer. Dans certaines versions de R, il faut ouvrir le dossier en question pour en faire le dossier de travail. On peut aussi importer dans la console R des fichiers qui ne se trouvent pas dans le « Working Directory » par la commande suivante :

```
> toto <- read.table( file.choose() )
```

Cette commande ouvre une boîte de dialogue qui permet de sélectionner un fichier à lire.

Parallèlement, on peut sauvegarder un objet R sous la forme d'un fichier texte par la commande « write.table ». Exemple :

```
> write.table(toto,file="nom-de-fichier.txt")
```

L'argument « header » de « read.table » sert à indiquer s'il y a des noms en en-tête des colonnes du fichier; header=T (*true*) indique qu'il y a des noms de colonnes, header=F (*false*) qu'il n'y en a pas. L'argument « row.names » permet d'indiquer dans quelle colonne se trouvent les noms des objets (lignes). Si, par exemple, les noms d'objets se trouvent dans la colonne 8, il faut écrire « row.names=8 ». Les valeurs qui se trouvent sur une même ligne sont séparées par des espaces ou des tabulateurs. Par défaut, R reconnaît le point comme marque des décimales. On devra préciser « dec = "," » si la virgule marque les décimales dans le fichier à importer.

Les arguments « header » et « row names » ne sont pas nécessaires si le fichier est structuré comme suit : les noms d'objets se trouvent en colonne 1 et il y a des en-têtes aux colonnes, **sauf pour la colonne des noms d'objets**. R est capable de reconnaître les fichiers qui comportent un élément de moins à la première ligne (en-têtes) qu'aux lignes suivantes et il les importe sans précisions supplémentaires. Par exemple, le fichier « fich.txt » suivant :

Esp1	Esp2	Esp3	Esp4	Esp5	Esp6	Prof	Corail	Sable	Autre	
Site1	1	0	0	0	0	0	1	0	1	0
Site2	0	0	0	0	0	0	2	0	1	0
Site3	0	1	0	0	0	0	3	0	1	0
Site4	11	4	0	0	8	1	4	0	0	1
Site5	11	5	17	7	0	0	5	1	0	0
Site6	9	6	0	0	6	2	6	0	0	1
Site7	9	7	13	10	0	0	7	1	0	0
Site8	7	8	0	0	4	3	8	0	0	1
Site9	7	9	10	13	0	0	9	1	0	0
Site10	5	10	0	0	2	4	10	0	0	1

sera importé correctement par la commande

```
> recif <- read.table("fich.txt")
```

sans qu'il soit nécessaire de préciser les arguments « header » et « row names ». Il faut tout de même toujours vérifier si les données ont été importées correctement. Pour visualiser les données dans la console R, il suffit de taper le nom de l'objet qui les contient :

> recif

et appuyez sur retour. R affichera les données à l'écran.

On peut éditer et modifier le contenu d'un objet de type data.frame, matrix ou vector par la commande :

> fix(nom-de-l'objet-R) par exemple : fix(recif)

3. Les bibliothèques de R

R permet de réaliser des calculs à l'aide de fonctions préprogrammées. Ces fonctions sont stockées dans des bibliothèques. La commande

> .Library

fournit le chemin d'accès aux bibliothèques déjà chargées dans la machine. Si on désire utiliser une fonction d'une bibliothèque qui est présente dans la machine mais qui n'est pas active, il faut l'activer. On peut utiliser la commande **Load package…** (Windows) ou **Package Manager** (OS X) dans le menu **Packages**, ou encore taper la commande :

library(nom-de-la-bibliothèque) par exemple : library(MASS)

ou encore

require(nom-de-la-bibliothèque) par exemple : require(MASS)

Dans cet exemple, MASS est le nom de la bibliothèque de Venables et Ripley correspondant au contenu de leur livre *Modern Applied Statistics with S*. Cette bibliothèque, écrite d'abord pour le langage S, a été traduite en R par le Prof. Brian Ripley.

On peut apprendre l'existence de bibliothèques contenant des fonctions préprogrammées qui sont disponibles sur le disque dur grâce à la commande « help.search ». Par exemple :

> help.search("RDA")

vous informera que la bibliothèque « vegan » contient une fonction pour l'analyse canonique de redondance (RDA).

On peut télécharger des bibliothèques additionnelles, comme « vegan », à partir du WWWeb : Clients Windows : menu **Packages** => **Get CRAN Packages** => **Ready-to-use Binary**. Clients Mac OS X : allez dans le menu **Packages & Données** et cliquer sur **Installateur de Package** => **CRAN (binaires)** => **Acquérir Liste**. Il suffit de choisir parmi la liste des bibliothèques disponibles. Les bibliothèques suivantes seront utilisées dans les travaux pratiques du cours : 'ade4', 'ape', 'cclust', 'geoR', 'labdsv', 'mapdata', 'maps', 'mvpart', 'rgl', 'spdep' et 'vegan'. La bibliothèque 'packfor' de même que des fonctions en langage R, disponibles sur la page du cours, seront également utilisées.

4. Analyses statistiques en R

Presque toute la statistique contemporaine a été programmée en langage R. Jetez par exemple un coup d'œil aux fonctions de la bibliothèque « stat » dont on trouve la liste dans le fichier INDEX de cette bibliothèque. Il suffit d'appeler ces fonctions préprogrammées pour réaliser les calculs. Le fichier d'aide de chaque fonction présente des exemples d'utilisation.

Des exemples d'application aux statistiques de base sont fournis dans le fichier *Travaux_Pratiques_en_R*. Dans la première section de ce fichier, on trouve des calculs de statistiques descriptives (moyenne, médiane, variance, etc.) ainsi que des commandes permettant la production d'un histogramme de distribution de fréquence et de diagrammes de dispersion.

5. Sauvegarde des résultats

Pour sauvegarder le contenu du fichier console, il suffit d'aller dans **File => Save** ou **Save as…** Sauvegardez ce fichier dans votre dossier de travail et non sur le bureau. Plus souvent, on copie-colle les résultats qui nous intéressent dans un fichier texte ou une page de traitement de texte. Dans la plupart des cas, il est inutile de conserver les résultats intermédiaires des calculs puisqu'on peut produire à nouveau ces résultats en faisant exécuter les commandes R qu'on aura sauvegardées dans un fichier.

On peut sauvegarder les graphiques produits par R dans des formats qui varient selon la machine: pdf, metafile, postscript, jpeg, etc. en Windows ; pdf en OS X. Il faut activer la fenêtre graphique en cliquant dans la fenêtre, puis faire **File => Save** ou **Save as…** On peut sauvegarder en d'autres formats par des commandes tapées dans la console R. Une description de ces formats est fournie par :

```
> ?Devices
```

6. Conseils pratiques

Plutôt que de sauvegarder le fichier console à la fin d'une session, il est préférable de copier les lignes de commande dans un fichier texte au fur et à mesure du travail. Utilisez l'un des éditeurs ASCII (= texte) gratuits comme Notepad++ pour Windows ou TextWrangler pour Mac OS X.

Inscrivez beaucoup de commentaires dans votre fichier de travail. Vous serez ainsi capable de vous relire. Si les lignes de commentaires commencent par le caractère « # », vous pourrez faire exécuter à nouveau toutes vos commandes en copiant-collant une partie de votre fichier texte, commentaires compris, dans la fenêtre R.

Évitez les accents, les espaces et les virgules dans les **noms d'objets** et les **noms de variables** à l'intérieur d'un tableau : R ne sait pas gérer ces caractères.

7. Les opérateurs du langage R

Tableau 1. Les opérateurs du langage R. Certains sont présentés avec une espace dans le tableau ; il faut les écrire sans espace dans le langage R.

Opérateur	**Fonction**
Opérateurs arithmétiques	
+	Addition
-	Soustraction
*	Multiplication (pour les vecteurs et les matrices, ceci est le produit Hadamard, soir le produit élément par élément)
/	Division (pour les matrices, cette commande produit la division élément par élément)
% / %	Division entière de deux nombres
% %	Modulo (reste de la division entière)
*% * %*	Produit scalaire de deux matrices
^	Exposant
Opérateurs de comparaison	
>	Supérieur à
<	Inférieur à
>=	Supérieur ou égal à
<=	Inférieur ou égal à
==	Égal (opérateur logique)
! =	Inégal, différent (opérateur logique)
Opérateurs logiques	
& *ou* &&	« et » logique
| *ou* ||	« ou » logique
!	« non » logique
Opérateurs d'assignation	
<- *or* =	Assignent une valeur à un nom d'objet