MUSÉUM NATIONAL D'HISTOIRE NATURELLE

SEMIN-R

Données sous R : stockage et échange

Julio PEDRAZA ACOSTA

UMR 5202 Origine Structure et Evolution de la Biodiversité
Département Systématique et Evolution, MNHN
pedraza@mnhn.fr

Données sous R : Stockage et Echange

Julio Pedraza Acosta

Juin, 2008

Fil d'Ariane

- Types de Format
  sur quoi on travaille
- Les scripts
  Garder son code au chaud
- Format texte standard
  Les données dans leur plus simple appareil
- Autres formats
  sinon ça serait trop facile
- Bonux
  vers l'infini et au-dela

Types de Format

Les scripts :

fichier en format texte contenant des COMMANDES

extension monScript.R monScript.r

fonction de lecture : source()

Les données en format R :

fichier ASCII ou binaire contenant des objets R (vecteurs,matrix, data.frame, listes, etc...)

extension ‘mesDonnees.RData’ or ‘mesDonnees.rda’

fonctions : save() load()

Les données en format texte :

fichiers texte simples (’.tab’ ou ’.txt’) ou csv contenant un tableau de données

fonctions read.table() write.table()...

Les données en format perso et autres fantaisies

fichiers avec une structuration particulière

fonctions : soit fonction spécifique soit "manos a la obra"

les Scripts

Les scripts sont une suite de commandes R dans un fichier texte.

ex : 0.MonMegaScript.r

Il a, par convention, l'extension '.r' ou '.R'

Il peut être appelé avec la fonction source()

ex : 1.AppelDeMonScript.r

```
source(file, local = FALSE, echo = verbose, print.eval = echo,
verbose = getOption("verbose"), prompt.echo = getOption("prompt"),
max.deparse.length = 150, chdir = FALSE, encoding = getOption("encoding"),
continue.echo = getOption("continue"), skip.echo = 0)
```

le format R

R dispose d'un format propre de stockage de ces objets.

Il enregistre des objets par la fonction save()

- soit en binaire
- soit en "clair" : save(...,ASCII =TRUE)

Dans un fichier RData on peut enregistrer un ou plusieurs objets.

- soit save(objet1,objet2...., file="./fichierDonnees.RData")
- soit save(list= /liste avec les noms des objets à enregistrer/, file="./fichierDonnees.RData")

Pour charger les objets on utilise la fonction load()

ex : 2.FormatR.r

compress dans save() permet de "gzipper" le fichier. load() le lira sans problème

en format Text et CSV

R peut utiliser des tableaux dans des fichiers en format texte

Ils sont lus suivant deux paramètres :

- un délimiteur qui sépare les valeurs (sep=...)
- un encapsuleur de chaînes de caractères (quote=...)

On distingue deux types par conventions suivant le séparateur des colonnes :

- les fichiers texte (’.txt’) où sep est un espace
- les fichiers csv (comma separated values) où sep est ’,’ ou ’ ;’

la séparation de lignes se fait par des retours chariots.

Format très pratique car faciles de modification avec des éditeurs de texte / tableurs et des nombreux logiciels peuvent les traiter.

en format Text et CSV : Lecture

Une fonction read.table() avec des nombreux paramètres

read.table(file, header = FALSE, sep = "", quote = ’ " ’, dec = ".", row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#", allowEscapes = FALSE, flush = FALSE, stringsAsFactors = default.stringsAsFactors(), encoding = "unknown")

Plusieurs fonctions qui ne changent que par leurs paramètres par défaut :

- read.csv
- read.csv2
- read.delim
- read.delim2

en format Text et CSV : Ecriture

Je vous le donne en mil...

write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ", eol = "(slash)n", na = "NA", dec = ".", row.names = TRUE, col.names = TRUE, qmethod = c("escape", "double"))

Et comme pour read.table(), on a write.csv() et write.csv2()

Note : un paramètre append qui permet d'ajouter l'objet à la fin du fichier

Avec ce type de fichier il est très facile d'importer/exporter des vecteurs, data.frames, matrices.
ex : 3.FormatTexteCsv.r

Pour les listes c'est une autre histoire...

Autres formats

C’est ici que les choses deviennent d-R-ôles.
Il faut créer un "connecteur" sur le fichier soit en lecture soit en ecriture avec file()

```
fichierLu <- file("nomFichier.txt")
```

Il faut utiliser des fonctions de lecture comme scan() et read.lines() puis ecrire du code (arghhh) pour découper les valeurs à extraire et les affecter à un objet

Pour créer un fichier au format voulu il faut créer les lignes voulues et utiliser la fonction cat() ou write.lines()

et fermer le connecteur avec close()

```
close(fichierLu)
```

ex : 4.LectureContourBeeji.r

Bonux : Plusieurs fichiers

Pour travailler avec plusieurs fichiers on peut s'aider de :

- des boucles pour traiter les fichiers un par un
- getwd() et setwd() pour récupérer et changer de répertoire courant
- list.files() pour lister des fichiers dans un répertoire.
- des expressions régulières pour filtrer les fichiers

"Y manos a la obra"

ex : 5.LectureRepertoirePoints.r

Lancée des tomates

- Types de Format
- Scripts
- RData
- Texte et Csv
- Autres
- Bonux

Questions ?