Installation et gestion des packages sous R

Ricco.Rakotomalala
http://eric.univ-lyon2.fr/~ricco/cours

Packages
Pourquoi, comment ?

Package ?

• Un package est une bibliothèque externe

• Sous Windows → fichiers binaires pré-compilés

• Extension .zip

• Il est toujours documenté : fichier HTML (aide sous R) et PDF

Quel intérêt ?

• Un package contient des collections de fonctions utilisables sous R

• Souvent centrés sur un sujet particulier (ex. *rpart* pour les arbres de décision, etc.)

• Gestion affinée des packages : nous pouvons les installer, désinstaller, charger, décharger et mettre à jour à notre guise

Ce système permet d'augmenter considérablement la puissance de R !!!

Objectif de ce support

• Localiser un package qui nous intéresse

• L'installer (et le désinstaller) sur notre machine

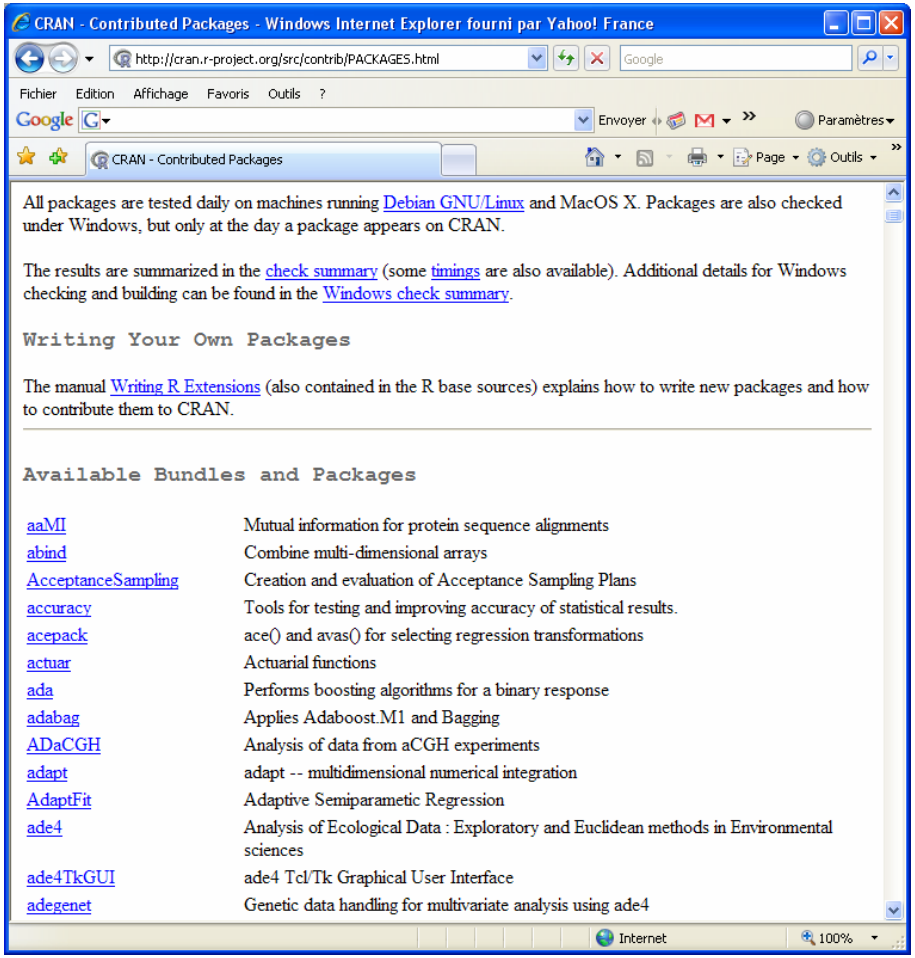
• Le charger (et le décharger)

• Accéder à l'aide et la documentation

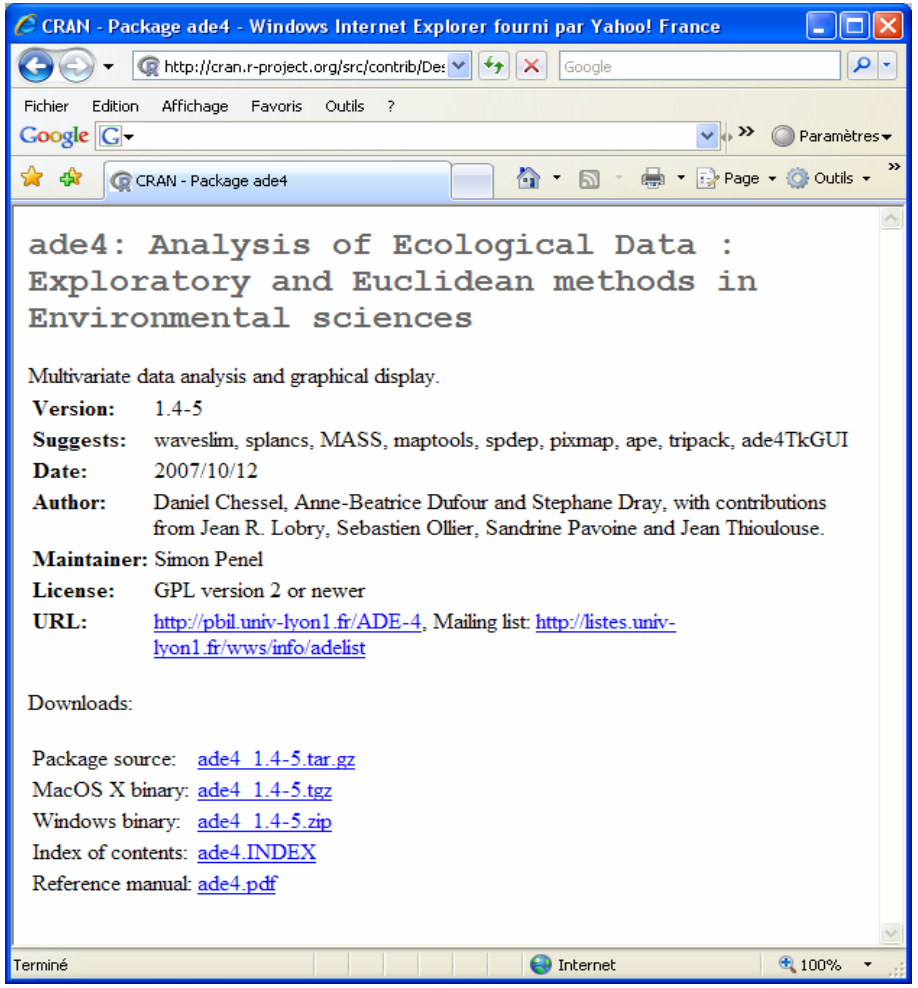
• Faire appel aux fonctions du package

Liste des packages disponibles

Disponible sur le site : http://cran.r-project.org/src/contrib/PACKAGES.html

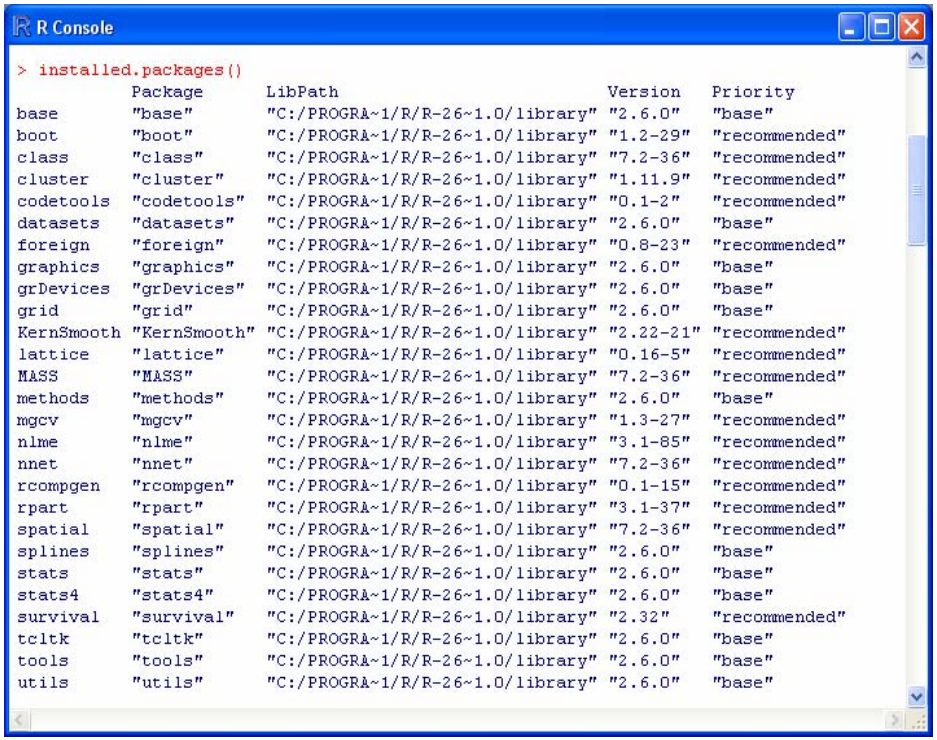


A chaque package correspond une série de fichiers, dont la documentation PDF



Les packages de notre configuration

Obtenir la liste des packages installés sur notre machine

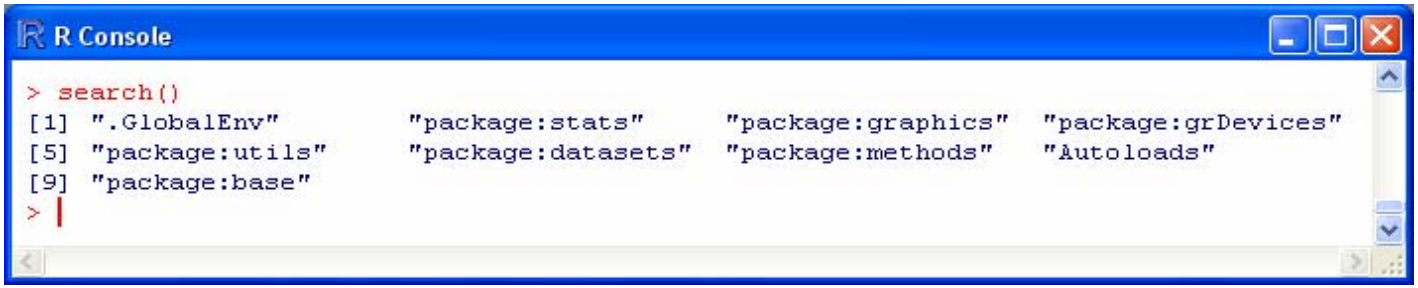


Des informations complémentaires sont fournies :

- Description succincte
- Dépendances (!)
- Etc.

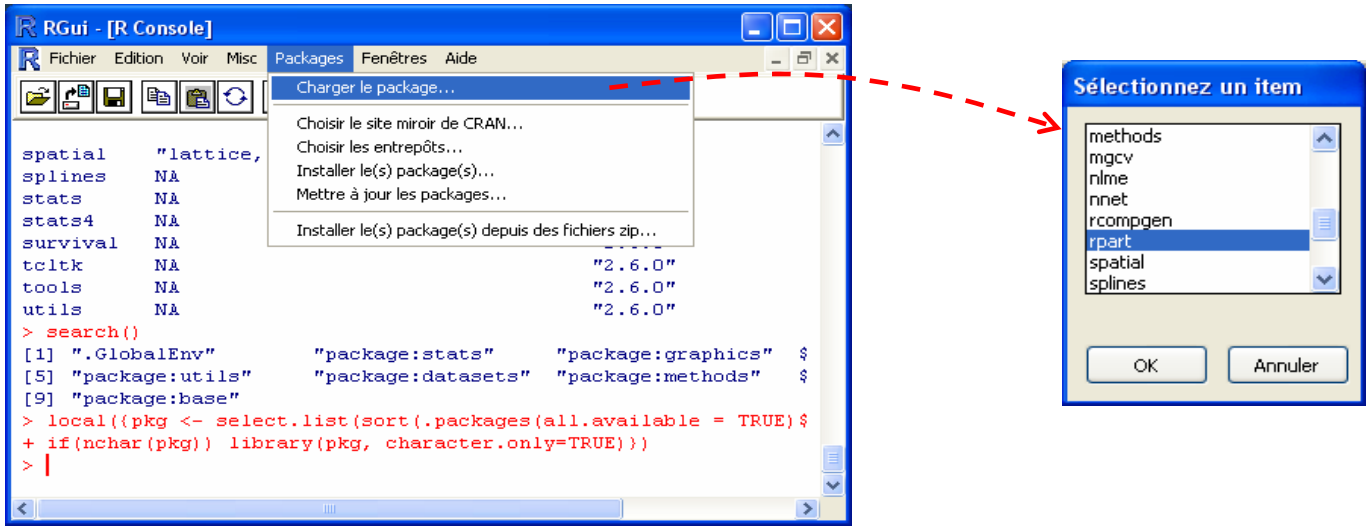
> **library()** permet aussi de lister les packages installés

Obtenir la liste des packages chargés

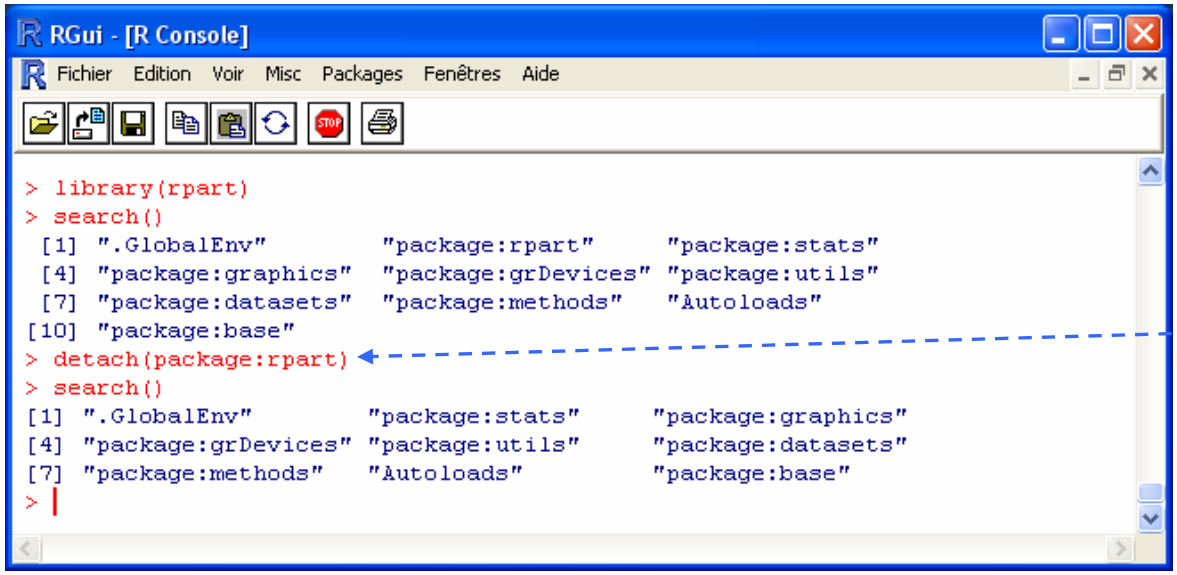


Charger un package déjà installé
(ex. le package rpart pour construire des arbres de décision)

Utiliser les menus



Utiliser la ligne de commande



On peut « décharger » un package (pour éviter les conflits entre fonctions de packages différents par ex.)

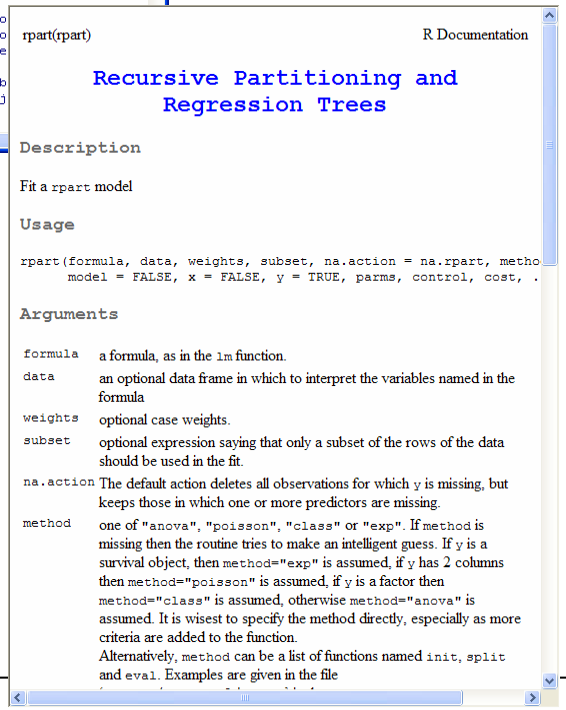
Obtenir de l'aide sur un package

Liste des fonctions, syntaxe des fonctions, etc.

(1) En chargeant la documentation PDF (le plus simple et le plus recommandé)

(2) En insérant les commandes adéquates dans R

```
#obtenir la liste des fonctions
library(help=rpart)
help(package=rpart)
```




```
#obtenir de l'aide sur une fonction du package
help(rpart,package=« rpart »)
#si le package est chargé, pas besoin de le spécifier
help(rpart)
```

Exemple d'utilisation du package rpart

Code source

```r
#charger les données dispo dans le package "rpart"
data(car.test.frame)

#aperçu des données
summary(car.test.frame)

#arbre de régression
arbre <- rpart(Price ~ ., method = "anova", data = car.test.frame)

#arbre sous forme textuelle
print(arbre)

#arbre sous forme graphique
plot(arbre)
text(arbre)
```

Sortie texte

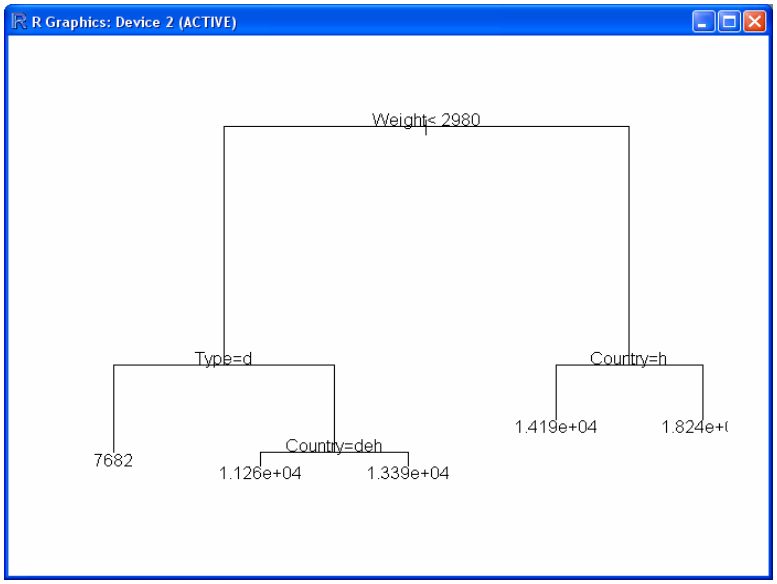
```
R Console

>
> #arbre sous forme textuelle
> print(arbre)
n= 60

node), split, n, deviance, yval
      * denotes terminal node

 1) root 60 983551500 12615.670
   2) Weight< 2980 36 283686500 10442.580
     4) Type=Small 13  21804710  7682.385 *
     5) Type=Compact,Medium,Sporty 23 106857900 12002.700
      10) Country=Japan/USA,Korea,USA 15  21272980 11261.200 *
      11) Country=France,Germany,Japan 8  61874100 13393.000 *
   3) Weight>=2980 24 274858800 15875.290
     6) Country=USA 14  47843040 14185.710 *
     7) Country=Japan,Sweden 10 131098800 18240.700 *
>
```

Sortie graphique

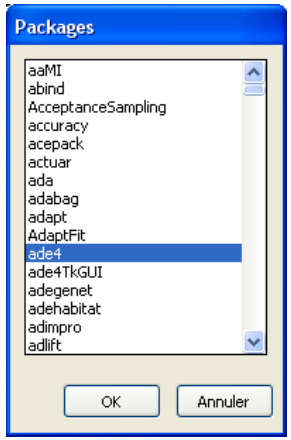


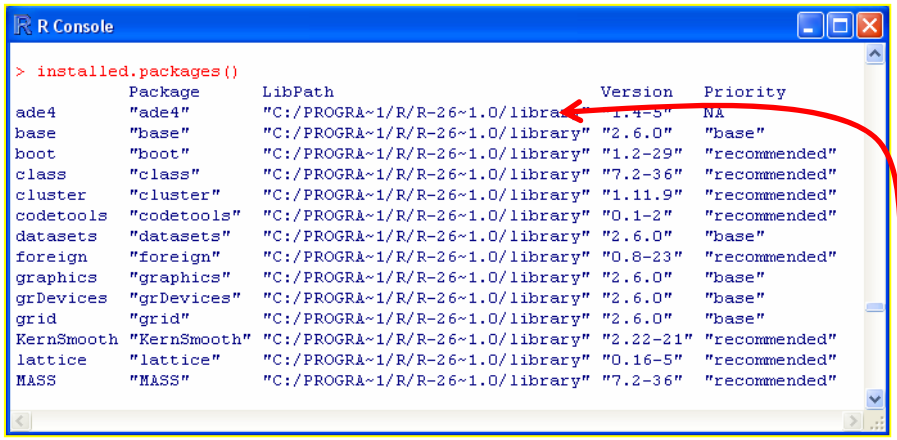
Installer un nouveau package

Ex. Package ade4 pour l'analyse de données

1 - Menu « Packages / Choisir le site miroir CRAN »
(pas trop loin si possible)

2 - Menu « Packages / Installer des packages»





3 - Menu « Packages / Mettre à jour les packages »
Pour obtenir la dernière version des packages déjà installés

Conclusion - R est merveilleux

- Il y a (presque) toujours un package adapté à nos problèmes
- Nous pouvons programmer dans R pour compléter les fonctions
- Nous pouvons programmer de nouveaux packages distribuables