Introduction à R

MDI 343 - Apprentissage

24 février 2010

Objectifs de la séance

Le but de cette séance est d'apprendre à utiliser efficacement le langage R et son environnement de travail. Après avoir appris à effectuer les commandes de base, vous apprendrez les rudiments de la programmation en langage R (qui est très proche de nombreux autres langages de calculs). Sur le plan de l'apprentissage statistique, seront évoquées dans cette séances les thèmes suivants : régression linéaire, régression logistique, perceptron.

La séance sera d'autant plus profitable que chacun essayera de faire le maximum en autonomie. Avant de poser une question, essayez de résoudre votre problème tout seul. Vous pourrez vous aider de l'aide en ligne, ainsi que du manuel disponible à l'adresse :

`http://cran.r-project.org/doc/manuals/R-intro.html`

Pour le contenu d'apprentissage, on utilisera comme en cours le livre de référence *The Elements of Statistical Learning*, de Hastie, Tibshirani et Friedman (il y est fait référence ci-dessous sous l'acronyme HTF), disponible en ligne à l'adresse :

`http://www-stat.stanford.edu/~tibs/ElemStatLearn`

Découverte de R

Ligne de commande, aide en ligne et manuel

Basiquement, R a toutes les fonctionnalités d'une calculatrice moderne. Bien noter l'opérateur d'affectation $\leftarrow$ un peu inhabituel.

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. A l'aide du manuel d'introduction à R disponible sur internet, trouver comment effectuer les opérations de base d'algèbre linéaire.

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`.

4. Récupérer sur le site

   `http://perso.telecom-paristech.fr/~garivier/centrale/`

   le fichier relatif aux "Lois dérivées de la Gaussienne", et le lancer dans R.

Fonctions

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

5. Ecrire et tester une fonction permettant de calculer la factorielle d'un entier positif.

Graphiques

La commande de base pour les représentations graphiques est `plot`. Par défaut, elle ne relie pas les points entre eux.

6. Regarder ce que fait la commande `lines`. Représenter deux fonctions usuelles sur le même graphe.
7. Illustrer graphiquement la loi forte des grands nombres pour les variables de Bernoulli de paramètre 3/4.

Gestion des données

En plus des classiques tableaux, vecteurs et matrices, R possède deux structures de données particulièrement utiles pour manipulées des données numériques : les *lists* et surtout les *data frames*. Ces données peuvent être chargés simplement 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 HTF sont disponible sur le site indiqué plus haut.

8. Regarder dans le manuel ce que sont les data frames.
9. Exécuter les commandes suivantes, et comprendre ce qui se passe :
   `data(); attach(cars); plot(speed,dist)`
10. Grâce à la commande
    `read.table("http://www-stat.stanford.edu/ tibs/ElemStatLearn/datasets/SAheart.data",...`
    `sep=",",head=T,row.names=1)`
    récupérer les données utiles pour la régression logistique (plus bas) et regarder de quoi elles sont constituées.

Gestion des packages

Le chargement d'un package (=module complémentaire, qui ajoute des fonctionnalités au noyau de base de R) se fait par la commande `require`.

11. Récupérer sur le site

    `http://perso.telecom-paristech.fr/~garivier/centrale/`

    le fichier relatif à la régression linéaire simple, et regarder ce qu'il contient.

Programmation en R

Régression linéaire

Les régressions linéaires sont gérées par dans logiciel R par la commande `lm`. Executer les commandes suivante, comprendre ce qu'elles font et ce qu'elles renvoient.

Code 1 - Régression linéaire simple

```
1: attach(cars)
2: summary(cars)
3: plot(speed, dist)
4: reglin ← lm(dist ∼ speed)
5: summary(reglin)
```

Retrouver sans son aide tous les résultats (ou en tous cas une bonne partie) fournis par la commande `lm` sur cet exemple.

Régression logistique

12. Ecrire un programme qui calcule l'estimateur du maximum de vraisemblance dans le modèle de régression logistique grâce à l'algorithme de Newton-Raphson. Si besoin, se référer à la section 4.4 du HTF.
13. Tester la procédure sur un exemple simulé.
14. Chercher comment faire la même chose directement grâce à la commande `glm`.
15. Reprendre l'exemple du HTF concernant les maladies cardiaques en Afrique du Sud.

Perceptron

16. Ecrire une procédure qui ajuste perceptron monocouche comme décrit dans la section 11.3 du HTF.
17. Générer des données séparables dans $\mathbb{R}^2$, et illustrer graphiquement la convergence de l'algorithme. Essayer avec un exemple de données non séparables.