Introduction au langage R

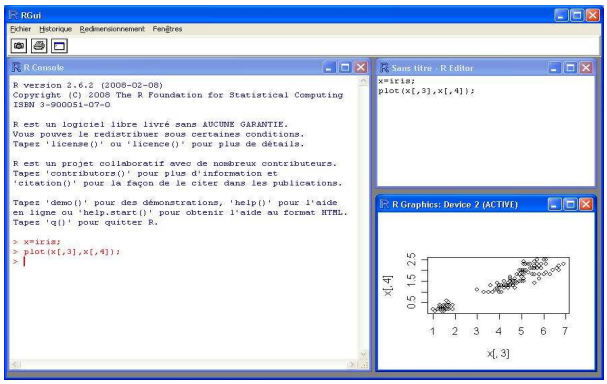
support de cours

Présentation

R est un système d'analyse statistique et graphique créé par deux chercheurs américains (Ihaka R. et Gentleman R.. 1996. (Référence: R: a language for data analysis and graphics. Journal of computational and Graphical statistics 5: 299-314). R est distribué gratuitement sous les termes de la GNU General Public Licence; son développement et sa distribution sont assurés par plusieurs statisticiens rassemblés dans le R Development Core Team. Les fichiers pour installer R sont disponibles à partir du site internet www.r-project.org du Comprehensive R Archive Network (CRAN). R comporte de nombreuses fonctions pour les analyses statistiques et graphiques (fenêtres propres ou exportées sous divers formats). Les résultats des analyses statistiques sont affichés à l'écran; certains résultats peuvent être sauvés à part, exportés dans un fichier ou utilisés dans les analyses ultérieures. Le langage R permet de programmer des boucles. Différentes fonctions statistiques peuvent être combinées dans un même programme pour réaliser des analyses plus complexes.

Environnement de travail

L'exécution sous windows de l'application R va provoquer l'affichage d'un environnement de travail graphique. Différentes fonctions sont accessibles via les menus de l'application.



Propriétés

- **R est un langage** *interprété* Les commandes tapées dans la console, sont interprétées puis directement exécutées.
- **R est un langage** *orienté objet* **.** Les variables, les données, les fonctions sont stockées dans la mémoire de l'ordinateur sous forme d'objets. Une analyse peut être faite sans qu'aucun résultat soient affichés à l'écran.
- **Les fonctions R** Les fonctions disponibles sont localisées dans des bibliothèques (packages) sauvegardées sur le disque dans un répertoire (*library*) de l'arborescence où R est installé.

Manipulation sous R

Les Types de variables:

- **Les Types de données:**
  - numérique: 10, 10.2...
  - caractère: Paris...
  - Booléean: TRUE, FALSE
  - facteur: un facteur est une variable catégorielle, de type numérique ou caractère.
- **Les Affectations simples:** L'instruction suivante affecte la valeur numérique 10 dans la variable n:
  - $n < -10$;
  - $10 -> n$;
  - $n = 10$;

  L'instruction suivante affecte la chaîne de caractères Paris dans la variable name
  - name= "Paris";
- **Les opérations élémentaires:** +;-; *; /.

Les fonctions élémentaires

Il existe de nombreuses fonctions sous R avec pour chacune un paramétrage et des sorties propres. L'aide en ligne de R est extrêmement utile pour connaître les modalités d'utilisation d'une fonction. L'aide en ligne est disponible directement par la commande

- help(nomfonction) ou
- ?nomfonction.
- Exemple: ?exp ou help(exp).

Les structures de données sous R

Il existe différentes structures de données sous R qui vont permettre de stoquer et de manipuler des données plus ou moins complexes: liste, vecteur, matrice, dataframe...

- **Vecteurs**
  - Création d'un vecteur d'éléments . Exemple: `x=c(1,2,3,4,5,6)`, (x=1:6).
  - x[i] permet d'accéder au ième élément du vecteur.
  - x[] permet d'accéder à tous les élements du vecteur.
  - Examples de fonctions associées: length(x); is.vector(x); ...
- **Matrices**
  - création d'une matrice. Exemple: `matrix(data = x, nrow = 2, ncol =3)`
  - m[i,j] permet d'accéder à l'élement de la ligne i et colonne j
  - m[i,] permet d'accéder à tous les élements de la ligne i.
  - m[,j] permet d'accéder à tous les élements de la colonne j.
  - m[,] permet d'accéder à tous les élements de la matrice m.

- Produit matriciel: $m1\% * \%m2$ effectue le produit matricielle de m1 et de m2, sous réserve que les tailles des matrices soient bien sûr compatibles.
- Exemples de quelques fonctions: is.matrix(m), nrow(m), ncol(m), dim(m), t(m) ...

- **DataFrame** Les structures de type dataframe sont particulièrement bien adaptées au stockage de données pour des traitements statistiques ultérieurs, et vont permettre de manipuler des tableaux décrivant n individus statistiques pour un ensemble de p variables quantitatives ou qualitatives. Chaque variable et/ou chaque individu statistique pourra être explicitement nommé(e) par son nom.

  Exemple de tableau de données stockées sous forme de dataframe notée tab. Les données "iris" décrivent $n = 150$ iris pour p=5 variables; 4 variables quantitatives (Sepal.Length, Sepal.Width, Petal.Length, Petal.Width), et une variable qualitative (Species).

  - tab=iris; affectation des données iris

obs	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	setosa

  - `tab[, "Petal.Length"]` permet d'accéder à la liste de toutes les valeurs de la variable Petal.Length.

  De nombreux exemples de données sont disponibles sous R, accessibles par l'intermédiare de structures "dataframe" définies par défaut dans le logiciel R. Pour y accéder, il suffit de taper le nom du dataframe (ici iris) contenant les données (affectation `tab=iris;` La commande `data()` permet de lister l'ensemble des tableaux de type dataframe accessibles sous R.

- **Les structures à champs:** certaines structures peuvent contenir des valeurs de types différents; les valeurs sont alors stockées dans des champs. On accède aux différents champs d'une variable par le caractère $ suivi du nom du champ. Exemple: La variable x contient deux champs c1 et c2. Ces différents champs seront accessibles par la commande `x$c1` et `x$c2`.
  La commande `attributes()` appliquée à un objet permet d'obtenir la liste des champs de cet objet.

Lecture de données externes

Lecture de fichiers au format texte: R peut lire des données stockées dans un fichier texte (ASCII) à l'aide de la fonction read.table. Cette fonction retourne un objet de type data frame. La fonction `read.table` possède un grand nombre de paramètres `help(read.table)`.

Exemple: la commande `t=read.table('ex.txt',header=TRUE,row.names=1)` permet d'affecter les données contenues dans le fichier texte "ex.txt" dans une structure de type dataframe.

Création de fonctions

La syntaxe de déclaration d'une fonction sous R est la suivante : La fonction notée `foo()` prends k arguments x1,x2,...,xk et retourne `yout` en sortie.
`foo<-function(x1,x2,...,xk) {...; return(yout); }`
Cette fonction sauvegardée dans un fichier foo.R pourra être chargée dans l'environnement par la commande `source("foo.R")`.

Les boucles

La syntaxe de réalisation d'un boucle est la suivante :
`for (i in 1:10) { ....; }`

Graphiques sous R

Les graphiques sous R: R permet de réaliser de nombreux graphiques. La commande `demo(graphics)` permet d'obtenir une illustration sur un ensemble de graphique. Un graphique s'affiche dans une fenêtre graphique windows. Par défaut, une seule fenêtre graphique est ouverte.

- La commande `plot(x,y)` permet d'afficher y en fonction de x, pour x et y vecteurs de même taille.
- La fonction hist permet de calculer et d'afficher l'histogramme associé à un échantillon (cf. `help(hist)` pour plus de précisions). L'affectation `z=hist(x)` permet de récupérer dans la variable z un ensemble d'informations sous forme de champs; `z$mids` donne les centres des classes, `z$counts` le nombre d'individus par classe.
- La commande `x11()` permet de créer une fenêtre graphique supplémentaire dans laquelle pourra s'afficher le prochain graphique. La commande `split.screen` permet de partitionner un graphique actif.
- La commande `par()` permet d'adapter certains paramètres de la fenêtre graphique. `help(par)` pour plus de détails sur cette fonction.

Quelques fonctions et exemples

- `help(nom)`: affiche l'aide de la fonction *nom*. `help(mean)` affiche l'aide de la fonction `mean()`.
- `seq(1,10)`: génére 10 nombres entre 1 et 10. `help(seq)` pour plus de détails.
- `mean(x)`: calcul de moyenne pour x vecteur, ou matrice.
- `sd(x)`: calcul d'écart-type pour x vecteur ou matrice.
- `rnorm(n)`: génération de n nombres aléatoire N(0,1)
- `cor(m)`: Calcul les corrélations d'un tableau de données, m matrice (ou dataframe).
- `plot(d)`: toutes les relations graphiques entre les variables du dataframe sont visualisées.
- `median()`: calcul de médiane.
- `min()`: calcul de min
- `max()`: calcul de max.
- `lm()`: modèle linéaire (estimation des paramètres)
- `predict.lm` (`predict`): modèle linéaire (prédiction)
- `glm()`: modéle linéaire généralisé
- `predict.glm` (`predict`): modèle linéaire généralisé (prédiction)
- `solve()`:inversion de matrice
- `t()`: transpose
- :...
- :...

Prise en Main

Exercice 1

Génération de données et estimation par moindres carrés ordinaires

- Déclarer deux paramètres, notées n et p, qui contiendront respectivement le nombre d'observations et le nombre de variables étudiées. On prendra ici $n = 50$, $p = 2$
- Déclarer une matrice notée X de dimension $n \times p$ en affectant la valeur zéro à tous les éléments de la matrice.
- La matrice X contient à présent les observations de p variables suivant une loi normale centrée réduite. Donner les instructions adéquates
- Génerer à l'aide de R une variable cible Y telle que $Y = 5 + 2X_1 - 3X_2 + \epsilon$ avec $\epsilon \sim \mathcal{N}(0, 1)$
- On suppose à ce stade que X et Y sont connus. Proposer les instructions permettant de retrouver l'estimation des paramètres du modèle proposé dans la question précédente à l'aide de produits matriciels.
- Programmer une fonction `mco()` qui prenne en entrée les arguments X et Y et dont le résultat est l'estimation des paramètres par la méthode des moindres carrées ordinaires
- Programmer une fonction `mco()` qui prenne en entrée les arguments X et Y et dont le résultat double : l'estimation des paramètres par la méthode des moindres carrées ordinaires et la valeur des erreurs quadratiques entre les valeurs de la variable initiale Y et la variables estimées $\hat{Y}$.

Exercice 2

Graphiques et statistique sous R

- Analyser les données `Orange` en vous aidant de la commande `help(Orange)`
- Affecter dans deux paramètres n et p le nombres d'observations et de variables
- A l'aide de la commande `demo(graphics)`, créer trois graphiques pertinents sur ces données