TP Analyse des données n^0 1 : Initiation à Splus

Premier contact avec le logiciel Splus : on propose ici une suite de commandes à taper sous Splus. On fera l'effort de comprendre la syntaxe du langage. On utilisera l'éditeur de ligne, qui permet d'utiliser des commandes de type `emacs` pour rappeler et éditer les lignes de commandes du langage. On rappelle que sous UNIX, les minuscules et les majuscules sont des caractères différents.

On commence par donner ici quelques commandes de emacs utiles pour éditer les lignes:

Rappel de la commande précédente (peux se répéter)	Ctrl p	(previous)
Annulation de commande (en cas de problème)	Ctrl g	
Caractère précédent	Ctrl b	(back)
Caractère suivant	Ctrl f	(forward)
Début de ligne	Ctrl a	
Fin de ligne	Ctrl e	(end)
Ligne suivante	Ctrl n	(next)
Effacement caractère courant	Ctrl d	(delete)
Effacement fin de la ligne	Ctrl k	(kill)

L'objectif de ce TP n'est pas de vous enseigner le logiciel Splus, mais de vous familiariser avec sa structure, de manipuler des objets et des fonctions, et d'avoir un apperçu de ses possibilités graphiques.

Avant de commencer

Splus -e	Lancer Splus (avec l'option éditeur de ligne)
q()	Quitter Splus Un répertoire .Data se crée dans votre répertoire personnel (c'est votre répertoire de travail par défaut)
cp ~carlier/.Data/.First .Data	Copie un fichier de commande initial dans votre répertoire de travail

Quelques notions de base

Splus -e	Relancer Splus
help.start()	Ouvre la fenêtre d'aide

La fenêtre d'aide se découpe en trois zones, une liste de catégories de fonction (en haut à gauche), une liste de fonction (en haut à droite) et un endroit pour taper les sujets (`topic` en bas à gauche). Cliquer sur la catégorie `regression` puis sur la fonction `lm`. En utilisant la fenêtre `topic`, taper distribution puis cliquer sur la ligne `dnorm`. Pour terminer, iconifier la fenêtre graphique.

Création et manipulation d'objets Splus

ls()	Liste les objets Splus situés dans votre répertoire de travail.
1:50	Génère la suite des entiers de 1 à 50 : où est passé cette suite ?
ls()	Quelque chose de nouveau ?
x_1:50	Le signe souligné est un caractère d'*affectation*.

Pour affecter une valeur à un objet, on peut utiliser le signe _ ou la suite des deux signes > _ Le signe = est réservé aux options de certaines commandes, le signe == étant lui utilisé pour les tests de valeur.

ls()	What news ?
x	Qu'y a t'il dans `x` ?
x_seq(0,5,length=10)	On définit un nouveau vecteur `x`.
x	Contrôle
y_2*x	On définit le vecteur `y`.
y	Contrôle
plot(x,y)	Message d'erreur `Device not active !`
motif()	Ouverture d'une fenêtre graphique.

	Premières représentations graphiques
plot(x,y)	Que se passe t'il ?
par(mfrow=c(2,2))	On découpe la fenêtre graphique en une matrice de figures. à 2 lignes et 2 colonnes (**m**atrice de **f**igures *by* **row**).
plot(x,y)	On recommence.
title("Sans option")	Ecriture d'un titre.
plot(x,y,type="l")	Avec l'option **l**ine.
title("Avec l'option 'l'")	Ecriture d'un titre.
plot(x,y,type="b")	Avec l'option **b**oth (les deux).
title("Avec l'option 'b'")	Ecriture d'un titre.
plot(x,y,type="n")	Avec l'option **n**ull.
text(x,y,letters[1:10])	mais on ajoute quelquechose.
	Création d'une fonction
`f_function(x) cos(1/(x+10^(-5)))`	Définition d'une fonction.
y_f(x)	Calcul de y pour les valeurs prédéfinies de x.
plot(x,y,type="b")	Représentation du graphe de la fonction.
x_seq(-3,3,by=0.01)	Nouvelle séquence de valeurs de x.
length(x)	Pour voir.
range(x)	idem (`range` signifie *étendue*).
max(x)	idem
min(x)	idem
plot(x,f(x),type="l")	Graphe de la fonction.
x_seq(-4,4,length=1000)	Points en abcisse plus serrés.
plot(x,f(x),type="l")	Graphe de la fonction.
?dnorm	Aide sur la fonction `dnorm` (densité de la loi normale).
plot(x,dnorm(x),type="l")	Graphe de la fonction.
plot(x,dnorm(x),type="l")	On remet ça.
q_qnorm(0.975)	Calcul du quantile associé à p=0.975 .
round(q,2)	Pour voir (arrondi avec deux décimales).
zz_seq(4,q,length=100)	Abcisses de point pour tracer la partie haute du polygone.
xx_c(q,4,zz,q)	Calcul des abcisses des points du polygone. La fonction c() sert à mettre ensemble des réels ou vecteurs pour construire un vecteur.
yy_c(0,0,dnorm(zz),0)	Calcul des ordonnées des points du polygone.
polygon(xx,yy)	Tracé du polygone.
	Simulation du schéma d'urne.
urne_c(rep("blanche",10),rep("noire",10))	Définition d'une urne contenant 10 blanches et 10 noires.
urne	Pour voir ce qu'il y a dans l'urne.
urne_sample(urne,20)	On remue l'urne.
sample(urne,1)	Un tirage au hasard.
urne	L'urne ne peut pas changer car on ne fait aucune affectation.
sample(urne,5)	On tire 5 boules sans remise.
sample(urne,20)	On tire 20 boules sans remise (nouvelle permutation).
sample(urne,30)	On essaye de tirer 30 boules sans remise, ça ne marche pas (pourquoi?)
sample(urne,30,rep=T)	On tire trente boules avec remise.
plot(c(0,200),c(0,1),type="n", xlab="nombre de tirages", ylab="Proportion de blanches")	Préparation de la fenêtre graphique. (attention, commande découpée sur plusieurs lignes)
title("Proportion des blanches en fonction du nombre de tirages")	

for(i in 1:200) {	On commence l'écriture d'une boucle.
tirage_sample(urne,i,rep=T)	On tire *i* boules avec remise.
proportion.blanche_sum(tirage=="blanche")/i	Calcul de la proportion de blanches.
points(i,proportion.blanche)	On trace un point de coordonnées *i* et *p* .
}	Fin de boucle.
	Attention. Si on met plusieurs instructions sur une ligne, il faut les séparer par un point-virgule.
	————————————————
	Manipulation de vecteurs et de matrices Différents indiçages
	————————————————
note	Un exemple de matrice.
x_1:20	Un exemple de vecteur.
x	Pour voir.
dim(x)_c(4,5)	On donne des dimensions à `x`.
x	Pour voir.
nomi_letters[1:4]	Définition de noms de lignes.
nomi	Pour voir
nomj_LETTERS[1:5]	Définition de noms de colonnes.
nomj	Pour voir
dimnames(x)_list(nomi,nomj)	On atribue des "noms de dimensions" à `x`.
x	Pour voir.
x+2	Qu'obtient-on?
2*x +1	Idem ?
ls()	Qu'y a-t'il dans mon répertoire de travail.
y_1:5	Définition du vecteur `y`.
x%*%y	Produit entre la matrice x et le vecteur colonne `y`.
z_c(1,2,2,1)	Définition du vecteur `z`.
x%*%z	Produit entre la matrice x et le vecteur colonne `z`. Que se passe t'il ?
z %*%x	Produit entre le vecteur "ligne" `z` et la matrice `x`.
x[1,]	Extraction de la première ligne.
x[,2]	Extraction de la seconde colonne.
x[c(4,2,1),]	Extraction des lignes 4, 2 et 1.
sort(x[,1])	On ordonne les éléments de la première colonne.
order(x[,1])	Numéro des lignes pour ranger la première colonne par ordre croissant.
x[order(x[,1]),]	Permutation des lignes pour ranger la première colonne par ordre croissant.
salaire_round(runif(25,4500,20000))	Variable salaire fictive .
sexe_sample(c("M","F"),25,rep=T)	Variable sexe fictive .
cbind(salaire,sexe)	Pour voir.
mean(salaire)	Moyenne du salaire.
sexe=="M"	vecteurs de booléens (T ou True = vrai, F ou False = Faux).
salaire[sexe=="M"]	Sélection des salaires des hommes.
mean(salaire[sexe=="M"])	Calcul de la moyenne du salaire des hommes.
tapply(salaire,sexe,mean)	Calcul de la moyenne des salaires par sexe.

Exercice En utilisant la commande list, définir un vecteur $(3, -2, 7, 5)$. Fabriquer aussi un vecteur contenant des noms ('bleu', 'Paris',...). Construire une matrice contenant vos couleurs d'yeux, vos ages et vos tailles. Faire la moyenne et l'écart-type des variables quantitatives.
Construire une matrice $(3, 8)$, donner ses valeurs et sous-espaces propres, trouver sa racine, la transposer, calculer la norme (euclidienne) de chacune de ses colonnes, donner sa décomposition en valeurs singulières.

References

[1] M. Baumgartner (1994) "Une introduction à Splus" École Fédérale Polytechnique de Lausanne. (disponible sous corail à l'adresse `/app/doc/stat/Sintro.ps` au CICT).

[2] A. Carlier et A. Croquette (1992) "Introduction aux langages S et Splus" (disponible sous corail à l'adresse `/app/doc/stat/cours_s.ps` au CICT).

[3] A. Carlier et A. Croquette (1994) Nouvelles fonctions pour l'anlayse multidimensionnelle sous Splus : la bibliothèque `multidim`. (disponible sous /users/DAS/carlier/SPLUS/TEX/plaquette.ps au CICT).

[4] W.N. Venables & B.D. Ripley (1994) *Modern Applied Statistics with Splus* Statistical Computing, Springer Verlag.