TP Analyse des données n^o 2

L'objectif de ce TP est de faire un peu d'analyse exploratoire et multivariée, de vous initier à certains types d'objets Splus, à leur organisation en répertoires, ainsi qu'à l'éditeur de lignes.

Quelques notions de base (suite)

Dans un premier temps on apprend à éditer les lignes de commandes. Pour ce faire il est nécessaire d'appeler `Splus` avec l'option `-e`. Ensuite vous devez vous munir du T.P. n^0 0 où un récapitulatif des commandes `emacs` les plus simples est donné page 1.

Splus -e	Relancer Splus.
motif()	Ouvrir une fenêtre graphique.
	————————————————
	Structures de données et répertoires.
	————————————————
help.start()	Ouvre la fenêtre d'aide.
help(vin)	Description du jeu de données.
is.data.frame(vinj)	"vinj" est une "structure de données Splus". contenant un tableau de variables hétérogènes.
names(vinj)	Liste des noms des variables de ce tableau.
summary(vinj)	Résumé sommaire des variables.
search()	Liste des répertoires dans lesquels sont cherchés les objets.
attach(vinj)	Attache la structure de données "vinj".
attach(vinv)	Attache la structure de données "vinv".
search()	Les structures de données jouent le rôle de répertoire et prennent place dans la "liste de recherche".
find(vinv)	Où se trouve l'objet `vinv`?
ls(pos=2)	Liste les objets situés dans le deuxième répertoire attaché.
	————————————————
	Première exploration graphique d'un jeu de données
	————————————————
par(mfrow=c(2,2))	par() permet de modifier des paramètres graphiques. Ici on découpe l'écran graphique en 4 (matrice de figures 2 x 2).
hist(Anth)	Histogramme de la variable "Anthociane".
title("Histogramme",cex=0.8)	Ecriture d'un titre.
hist(Anth,nclass=40)	On demande 40 classes.
title("Histogramme en 40 classes",cex=0.8)	cex taille des caractères.
hist(A.Vv)	
plot(Anth,A.Vv)	Diagramme de dispersion de Anth et A.Vv.
title("Diagramme de dispersion",cex=0.8)	Pour imprimer un dessin, il suffit de cliquer une fois sur la commande `print` du menu `Graph` de la fenêtre graphique.
	————————————————
	Elimination d'une valeur aberrante et nouvelles utilisations de l'indiçage des matrices
	————————————————
stem(A.Vv)	Diagramme en feuilles - permet une analyse fine de la répartition de la variable.
max(A.Vv)	Maximum de la variable A.Vv.
range(A.Vv)	Etendue de la variable A.Vv.
length(A.Vv)	longueur (notée n) de la variable A.Vv.
1:length(A.Vv)	suite des entiers de 1 à n.

A.Vv==max(A.Vv)	Suite de booléens résultat de cette condition logique. (trouvez la coordonnée égale à "T" (i.e. "true" ou "vraie").
(1:length(A.Vv))[5]	sélectionnne la cinquième coordonnée de ce vecteur.
(1:length(A.Vv))[A.Vv==max(A.Vv)]	sélectionne le numéro n1 de l'individu pour lequel A.Vv. est maximum.
ls()	Quels sont les objets de mon répertoire de travail ?
n1_(mettre ici le numéro affiché)	On charge ce nombre dans n1.
ls()	Quels sont les objets de mon répertoire de travail ? (bis).
A.Vv[n1]	Vérification.
A.Vv[-n1]	Enlève l'observation n1 de la variable A.Vv.
length(A.Vv[-n1])	Vérification...
A.Vv[n1]_max(A.Vv[-n1])	On remplace la valeur maximum par la seconde plus grande valeur. (opération contestable : rapellons que le caractère "_" est un signe d'affectation). Remarquez que lors d'une affectation, vous ne voyez pas l'objet créé. Pour le voir il suffit de retaper son nom.
plot(Anthj,A.Vv)	Nouveau diagramme de dispersion.

Splus ne modifie que les objets situés dans
le répertoire situé en première position de
la liste de recherche. Quand un objet est
situé plus loin dans la liste, toute
modification entraine sa recopie en position 1
La version originale de l'objet n'est pas modifiée

---

find(A.Vv)	Où se trouve la variable A.Vv ? (réponse, une version chez vous, une autre dans la structure de données `vinv`.
rm(A.Vv)	On efface la variable modifiée située chez vous.
find(A.Vv)	Où se trouve la variable A.Vv (bis)?
A.Vv[n1]]	La valeur ancienne est retrouvée : cette variable avait été recopiée, et le répertoire où elle se trouve ne peut être modifié par vous.
is.factor(annee)	Pour savoir si cette variable est qualitative (ou encore "facteur").
summary(annee)	Résumé d'une variable quantitative.
levels(annee)	annee n'a pas de "niveaux" (ou "modalités").
annee_ factor(annee)	On déclare la variable annee comme "facteur".
is.factor(annee)	Pour savoir si cette variable est qualitative.
summary(annee)	Résumé d'une variable qualitative.
levels(annee)	La variable annee a maintenant des niveaux.
table(annee)	Effectif des modalités de cette variable.
is.factor(P.G)	La variable porte-greffe est elle qualitative?
table(annee,P.G)	Tabulation croisée de ces deux facteurs. (ces variables sont elle contrôlées ?)

Liaison entre variables quantitatives

---

pairs(vinj)	Matrice de diagrammes de dispersion. (Ne pas considérer les "warnings" comme des erreurs)
plot(D420j,Anthj)	Diagramme de dispersion.
title(" Diagramme de dispersion ")	
cor(D420j,Anthj)	Calcul du coefficient de corrélation linéaire empirique.
summary(vinv)	Remarquer la différence de traitement de la variable quantitative "annee" et de la variable qualitative "P.G".

clear()	Effacement de la fenêtre graphique (fonction locale).
par(mfrow=c(2,3))	Découpage de la fenêtre en 6 (matrice 2×3).
cor(Degre,Anthj)	Corrélation linéaire empirique.
plot(Degre,Anthj)	Diagramme de dispersion.
par(xaxs="d",yaxs="d")	On bloque les échelles pour les dessins à venir (pour faire des comparaisons).
un_ 1:60	On définit deux ensembles d'indices (de 1 à 60 et de 61 à 150).
deux _ 61:150	
annee	Pour voir.
table(annee)	Pour voir.
table(annee[un])	Pour voir.
table(annee[deux])	Idem.
plot(Degre[un],Anthj[un])	Diagramme du premier groupe.
title("idem, années 83-85")	
plot(Degre[deux],Anthj[deux])	Diagramme du deuxième groupe.
title("années 88-89-90")	
cor(Degre[un],Anthj[un])	
cor(Degre[deux],Anthj[deux])	Conclusion ?
par(xaxs="r",yaxs="r")	On débloque les échelles.
res_lm(Anthj ~Degre)	Régression linéaire de Anthj en fonction du degré (le résultat est stocké dans l'objet `res`.
plot(Degre,Anthj)	Diagramme de dispersion (bis).
abline(res)	Représentation de la droite de régression. Remarquez que le coefficient de corrélation. mesure la qualité de la liaison *linéaire*.
	————————————
	Recherche d'une liaison non linéaire
	————————————
res_loess(Anthj~Degre)	régression "locale".
res.preplot_preplot(res)	Préparation du graphique.
plot(Degre,Anthj)	Diagramme de dispersion (ter).
par(xaxs="d",yaxs="d")	Bloquage échelles.
plot(res.preplot)	Graphique associé.
title("Résultats de loess(Anthj~Degre)	
plot(res.preplot,new=F)	(bis)
points(Degre,Anthj)	Représentation des points xi, yi. Remarquez que ce graphique "s'ajoute à un graphique existant".
title("Résultats de loess(Anthj~Degre) avec points	
plot(Degre,Anthj)	Diagramme de dispersion (iv).
res2_lm(Anthj~poly(Degre,3))	Régression polynomiale de degré 3. Les valeurs ajustées sont dans res$fitted.
xx_seq(min(Degre),max(Degre),length=100)	Génération d'une suite arithmétique de 100 nombres...
ynew_predict(res2,data.frame(Degre=xx))	Calcul des valeurs du polynomes sur les coordonnées de xx.
lines(xx,ynew)	Représentation du polynome qui s'ajoute à un graphique existant.
	————————————
	Liaison qualitatif quantitatif
	————————————
clear()	
par(mfrow=c(2,3))	
plot(annee,Anthj)	Diagrammes en boites.
plot(as.numeric(annee),Anthj)	Diagramme de dispersion.
plot(jitter(as.numeric(annee)),Anthj)	Diagramme de dispersion "secoué".
ro2(Anthj,annee)	Calcul du rapport de corrélation empirique.

Liaison qualitatif qualitatif

---

clear()	
par(mfrow=c(2,3))	
A.Tv.dis_cut(A.Tv,6)	Discrétisation.
table(A.Tv.dis)	
labb_paste("A.Tv",1:6,sep="")	Constitution de labels par collage.
labb	Visualisation.
A.Tv.dis_factor(A.Tv.dis, labels=labb)	On renomme les niveaux.
table(A.Tv.dis)	Tabulation.
A.Tv.dis	On visualise le facteur.
codes(A.Tv.dis)	Cette fonction recode en numérique.
Anth.dis_cut(Anthj,5)	Discrétisation de Anth
labb_paste("Anth",1:5,sep="")	
Anth.dis_factor(Anth.dis,labels=labb)	Nouveaux noms.
table(Anth.dis)	Tabulation.
N_table(A.Tv.dis,Anth.dis)	Tabulation croisée.
N	Pour voir.
plot(codes(A.Tv.dis),-codes(Anth.dis))	Pour comparer avec N.
plot(jitter(codes(A.Tv.dis)),-jitter(codes(Anth.dis)))	Idem après agitation.
val_chi2(N)	Calcul du χ^2 de la table de contingence N.
1-pchisq(val,5*4)	Calcul de la p-value.
xx_seq(0,max(val+2,3*5*4)),length=200)	Calcul d'une sequence de 0 a val+2.
yy_dchisq(xx,20)	Densité du χ^2 pour les valeurs de la séquence.
plot(xx,yy,type="l")	Représentation de la courbe de densité.
title("Densité sous H0 et valeurs observees")	
abline(v=0);abline(h=0); abline(v=val,lty=2)	

Régression linéaire

---

regr3	Voir les données de régression
is.data.frame(regr3)	qui sont un objet de type data.frame
motif()	Ouvre une fenêtre graphique
plot(regr3[,1],regr3[,2],xlab="travail",ylab="resistance")	Diagramme de dispersion
regr3[,1]	Pour voir.
travail	Pour voir (message d'erreur...).
find(travail)	idem
attach(regr3)	On attache la structure de données.
travail	Pour voir : maintenant les variables sont connues
find(travail)	par leur nom.
par(mfrow=c(2,2))	Matrice de figure "par ligne" (by row).
plot(travail,resistance)	Diagramme de dispersion.
res_lm(resistance ~ travail)	Régression linéaire
res	Pour voir
is.list(res)	`res` est de type liste
names(res)	avec beaucoup de composants.
res$coef	Pour voir le composant `coef`
abline(res)	droite de régression qui "s'ajoute à un graphique existant".
summary(res)	résumé des résultats de la régression
ltext("R2=...")	Remplacer les "..." par la valeur lue lors de la commande précédente. Affiche la valeur du R2 Positionement à la souris : rendre d'abord la fenêtre active en cliquant dans sa barre supérieure, puis cliquer à l'endroit où on veut écrire la valeur.
R2_1-var(res$res)/var(resistance)	Calcul direct du R2
segments(travail,res$fitted,travail,resistance)	visualisation des résidus

---

Régression polynomiale (degre 2)

---

history(lm,ed=T)	Autre moyen pour répéter des commandes avec édition Si l'éditeur choisi est `emacs`, après modification taper Ctrl X Ctrl S pour sauver, Ctrl X Ctrl C pour sortir (ou encore à la souris, utiliser le menu `file`). Générer la commande qui suit par édition.
res2_lm(resistance ~ poly(travail,2))	Ajustement polynomial (degré 2)
plot(travail,resistance)	diagramme de dispersion
lines(travail,res2$fitted)	Graphe du polynome
R2_1-var(res2$res)/var(resistance)	Calcul direct du R2 (éditer à partir de `history(var,ed=T)`).
round(R2,4)	Lecture de sa valeur (`round(x,n)` arrondi x à n décimales).
summary(res2)	Vérification.
ltext("R2= *valeur lue précédemment*")	Affichage de la valeur du R2.
segments(travail,res2$fitted,travail,resistance)	Visualisation des résidus.

Régression polynomiale (degre 9)

---

res3_lm(resistance ~ poly(travail,9))	Ajustement polynomial (degré 9.)
plot(travail,resistance)	Diagramme de dispersion
lines(travail,res3$fitted)	Graphe du polynome
summary(res3)	Résumé du résultat.
ltext("R2= *valeur lue précédemment*")	Affichage de la valeur du R2.

Modèle général additif avec lisseur spline

---

resg_gam(resistance ~(travail))	Ajustement du modèle
plot(travail,resistance)	Diagramme de dispersion
lines(travail,resg$fitted)	Graphe du polynôme
summary(resg)	Sommaire (il ne donne pas le R2)
R2_1-var(resg$res)/var(resistance)	Calcul direct
ltext("R2= *valeur lue précédemment*")	Affichage de la valeur du R2

Exemple de création de fonction et de dessin de graphe

---

f_function(x) x*sin(1/x)	fonction y=x sin(1/x)
x_seq(1/1000,pi/4,length=800)	construction d'une suite arithmétique
plot(x,f(x),type="l")	Dessin du graphe
fix(f)	Modification de la fonction dans l'éditeur (`fix()` = réparer). En cas de fonction syntaxiquement incorrecte, Splus signale qu'il faut récupérer le code en tapant fix().

Fonction calculant le R2 directement

	Le résultat `res` d'une régression est un objet de type "liste" contenant une composante `res$call` précisant quel modèle a été ajusté.
res$call	Pour voir (formule ayant généré le résultat `res`). Objet de la fonction qui suit : extraire de `res` les résidus (`res$residuals` ou `res$res`) puis la valeur de la variable à expliquer y (plus difficile..) et en tirer le R2 par la formule "directe": `1-var(residus)/var(y)` On extrait de la formule le nom de la variable à expliquer (première ligne) et on stocke son contenu dans y (2-eme ligne) Enfin on calcule le R2 en utilisant le composant `res$res` contenant les résidus.

res$call	Pour voir (formule ayant générer le résultat `res`)
calculr2_function(res)	En tête
{	Début
nomvar _ delimit(deparse(res$call), "=", "~")	Extraction du nom de la variable
y _ eval(parse(text = nomvar))	Extraction de son contenu
1 - var(res$res)/var(y)	Calcul du R2 (derniere ligne = valeur de la fonction)
}	Fin
calculr2(res)	Premier essai
calculr2(res2)	second essai
calculr2(res3)	troisième essai
calculr2(resg)	dernier essai