Bordeaux INP
ENSCBP

Mémento R sur les plans de mélanges - AGB 2A

I Sélection de données dans une matrice

Considérons une matrice M avec trois lignes et trois colonnes du type $M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$.

$M[x, y]$ renvoie le coeffcient en ligne x et colonne y ;
$M[x,]$ renvoie la ligne x ;
$M[, y]$ renvoie la colonne y ;
$M[, x : y]$ renvoie une matrice avec les colonnes comprises entre les colonnes x et y.
Exemples :

```
> M=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow=TRUE)
> M

     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

> M[2,1]

[1] 4

> M[1,]

[1] 1 2 3

> M[,2]

[1] 2 5 8

> M[,1:2]

     [,1] [,2]
[1,]    1    2
[2,]    4    5
[3,]    7    8
```

II Ajout de lignes ou colonnes à une matrice

Commande R :
cbind(..., ...) ajoute une voire des colonnes à une matrice.
rbind(..., ...) ajoute une voire des lignes à une matrice.

```
> vecteur = c(10, 11, 12)
> matrice_col = cbind(M,vecteur)
> matrice_col
```

```
     vecteur
[1,] 1 2 3      10
[2,] 4 5 6      11
[3,] 7 8 9      12
```

```
> matrice_line = rbind(M,vecteur)
> matrice_line
```

```
        [,1] [,2] [,3]
           1    2    3
           4    5    6
           7    8    9
vecteur   10   11   12
```

III Création de la matrice du plan et représentation graphique

III.1 Simplexe centroïde (plan centré) sans contrainte

Commande R :
Création : SCD(fac =...) où fac correspond au nombre de facteurs (composants)
Représentation : DesignPoints(...)

Exemple avec $n = 2$ composants :

```
> # Charger le package mixexp.
> n=2
> plan=SCD(n)
> plan
```

III.2 Plans en réseaux sans contrainte

Commande R : SLD(fac = ..., lev =...) où fac correspond au nombre de facteurs (composants) et lev au nombre de niveaux.

Exemple avec $q = 3$ composants :

```
> # Charger le package mixexp.
> q=3
> m=2
> plan = SLD(fac =q, lev =m)
> plan
> DesignPoints(plan)
```

III.3 Construction de plans sous contraintes

Commande R : Xvert(nfac = ..., lc = c(...), uc = c(...),ndm = ..., pseudo= ...)
où nfac correspond au nombre de facteurs (composants), lc et uc correspondent aux vecteurs de contraintes respectivement inférieures et supérieures, ndm étant associé à l'ordre (privilégier 1).
pseudo = TRUE/FALSE permet d'obtenir la représentation graphique soit en pseudo-composants, soit dans le triangle initial.

Exemple avec $n = 3$ composants pour $\begin{cases} 0,1 \leq X_1 \leq 0,2 \\ 0 \leq X_2 \leq 0,6 \\ 0 \leq X_2 \leq 0,7 \end{cases}$:

```
> # Charger le package mixexp.
> M=Xvert(nfac =3, lc=c(0.1,0,0), uc=c(0.2,0.6,0.7), ndm=1,pseudo=FALSE)
>
```

IV Modèle et validation

IV.1 Ajout des réponses à la matrice du plan

On note plan la matrice du plan et y le vecteur des réponses

```
> y=c(.....)
> donnees = data.frame(plan,y)
```

IV.2 Analyse du modèle et résidus

Commande R : modele = lm(...)
residus = resid(...)
Anayse = summary(...)

A noter que la commande lm(...) fait référence au modèle linéaire (linear model) à utiliser ainsi :

```
lm(Réponse ~ Variables)
```

Exemple dans le cas d'une analyse du modèle cubique dans le cas de 2 composants avec l'ANOVA dans le cas d'une matrice du plan donnees :

```
> modele=lm(y ~ (donnees$x1 + donnee$x2)^3)
> summary(modele)
```

Les résidus associés au modèle s'obtiennent en utilisant :

```
> modele=lm(y ~ (donnees$x1 + donnee$x2)^3)
> residus = resid(modele)
> residus
```

IV.3 Résultats du modèle

Détermination du modèle quadratique dans le cas de 2 composants avec l'ANOVA dans le cas d'une matrice du plan donnees.
Il est indispensable de mettre le **"-1"** afin de tenir compte de la contrainte fondamentale des mélanges.

```
> lm(y ~ -1 + ( donnees$x1+donnees$x2 )^2 )
```

V Optimisation

La fonction d'optimisation présentée ci-dessous minimise une fonction sous contraintes éventuelles. Pour obtenir le maximum, il suffit de considérer l'opposé de la fonction. L'exemple présenté traite d'un ajustement quadratique dans le cas de 3 composants.

```
> library("nloptr")
> f = function(x) -(...*x[1]+...*x[2]+...*x[3]+...*x[1]*x[2]+...*x[1]*x[3]+...*x[2]*x[3])
> # valeurs initiales (à changer si contraintes inf)
> x0 = c( 0, 0,0)
> # lower and upper bounds of control (à changer si contraintes)
> lb = c( 0, 0,0)
> ub = c( 1, 1,1)
> # Définition de la contrainte fondamentale des mélanges
> contrainte = function(x) x[1]+x[2]+x[3]-1
> # Optimisation
> gradient = function(x) nl.grad(x,f)
> optimisation = auglag(x0=x0,fn=f,lower=lb,upper=ub,heq=contrainte)
> # Obtention de la composition optimale
> valeurs_des_xi = optimisation$par
> # Obtention du maximum
> maximum = - optimisation$value
```

VI Graphe des effets

Commande R : EffPlot(des=matrice des données, mod=...)
où mod est associé à l'ordre du polynôme d'ajustement.