Exemples d'Analyses de Variance avec R

Christophe Pallier*

25 août 2002

Résumé

R est un logiciel d'analyse statistique qui fournit toutes les procédures usuelles (t-tests, anova, tests non paramétriques...) et possède des possibilités graphiques performantes pour visualiser les données. Pouvant être utilisé aussi bien en mode interactif qu'en mode « batch », R est disponible pour Linux (et autres unixes), Windows et Macintosh. De plus R est un logiciel libre, dont le code source est disponible et qui peut être recopié et diffusé gratuitement. Le but de cette note est de permettre au lecteur de découvrir les possibilités du programme en matière d'analyse de variance et de faciliter sa prise en main par de nouveaux utilisateurs. Pour cela, nous détaillons quelques exemples d'analyses de données structurées selon des plans classiques.

Le site web du logiciel R est www.r-project.org. On peut télécharger le programmes, des extensions, et également plusieurs documents, notamment dans la section "Documentation/Other/Contributed" plusieurs manuels intéressants, notamment Baron and Li [2000].

Cette note présente quelques exemples simples d'analyses de données avec R. Ces analyses sont présentées sous forme de "scripts", c'est à dire de suites de commandes qui peuvent soit être entrées interactivement sur la ligne de commande de R, soit être sauvées dans un fichier texte nommé, par exemple 'script.R'. Dans ce dernier cas, les commandes peuvent être exécutées en mode batch en se plaçant dans le répertoire contenant les données, puis en exécutant la commande *source('script.R', echo=T)* (Sous Linux, il n'est pas nécessaire de démarrer R : on peut entrer "R CMD BATCH script.R" sur une ligne de commande et les résultats sont écrits dans le fichier 'script.Rout').

Les données des exemples se trouvent dans le fichier donnees.zip.

*`www.pallier.org`. INSERM U562, SHFJ CEA, Orsay. Ce document n'est pas dans un état définitif. Tout commentaire pour l'améliorer ou me signaler des erreurs est bienvenu.

Une aide en ligne est disponible sous R : on peut rechercher de l'aide avec la fonction *help.search('mot clé')*, et obtenir de la description détaillée d'une commande en tapant ' *?nom_de_la_commande*' (par exemple *?t.test*).

1 Comparaison de n moyennes indépendantes

TAB. 1 – Données de quatres groupes indépendants

Groupe 1	Groupe 2	Groupe 3	Groupe 4
54	55	62	56
50	54	58	52
50	61	58	52
58	55	66	60
57	55	65	59
55	58	63	57
51	59	59	53
58	54	66	60
50	56	58	52
53	57	61	55

Des scores ont été recueillis pour quatre groupes indépendants de dix sujets chacun (cf. Table 1). Dans le language des plans d'expériences, le facteur « Sujet »à 40 modalité est emboité dans le facteur « Groupe »à 4 modalités.

Les données sont dans un fichier texte (`set1.dat`) sous la forme d'un tableau à deux colonnes, présentant sur chaque ligne, un label définissant le groupe, suivi du score du sujet correspondant. Le script suivant analyse ces données :

```
a<-read.table('set1.dat',col.names=c('Gr','Sc')) # lecture des données
attach(a) # permet d'accéder à ces données par les noms Gr et Sc
table(Gr) # affiche les effectifs de chacun des groupes définis par Gr
hist(Sc) # affiche un histogramme des scores
stripchart(Sc~Gr,method='stack')
plot(Sc~Gr) # affichage de "boites à moustache" de Score par Groupe
tapply(Sc,Gr,mean) # affiche les scores moyens par Groupe
barplot(tapply(Sc,Gr,mean))
tapply(Sc,Gr,summary)
l<-aov(Sc~Gr) # calcul de l'anova de Sc en fonction de Gr
summary(l) # résultat de l'anova
model.tables(l) # tailles des effets
plot(l) # graphiques diagnostiques
g1<-Sc[Gr=='G1'] # copie les scores des groupes 1, 2 et 3
```

```
g2<-Sc[Gr=='G2'] # dans les variables 'g1', 'g2' et 'g3'
g3<-Sc[Gr=='G3']
t.test(g1,g2) # comparaison G1 vs. G2, sans supposer l'homoscédasticité
t.test(g1,g2,var.equal=T) # en supposant l'homostécadicité
t.test(g1,c(g2,g3)) # comparaison G1 vs. G2_G3
pairwise.t.test(Sc,Gr) # comparaisons multiples 2 à 2
summary(aov(Sc~Gr,subset=Gr!='G4')) # anova excluant le groupe 4
wilcox.test(g1,g2) # test de Wilcoxon/Mann-Whitney
```

Commentons ce script ligne par ligne. Les caractères '#' introduisent des commentaires qui sont ignorés par R. La fonction *read.table* permet d'importer, dans une variable '*a*', un tableau de données provenant d'un fichier texte. Il existe aussi une fonction (*read.csv*) qui peut lire une feuille Excel sauvée au format CSV (« comma-separated-values »). Notons que les modalités des facteurs peuvent être indiquées clairement par des labels alphabétiques plutôt que par des codes numériques ; en général, c'est une bonne idée de donner des noms courts mais clairs aux modalités des facteurs. Le format des fichiers d'entrée n'est pas crucial car R dispose de fonctions de manipulation des données qui permettent de restructurer un tableau de données, voire de générer des facteurs éventuellement absents. Toutefois, pour effectuer une anova, le format le plus efficace est celui où chaque ligne du tableau de données liste les modalités de chaque facteur (qu'il soit intra ou inter sujet) et la mesure correspondante. Les données contenues dans le tableau '*a*' peuvent être visualisées en tapant simplement '*a*'.

L'exploration des données débute par la fonction *table(Gr)*, qui recense les effectifs des différents groupes définis par le facteur *Gr*. On peut ainsi vérifier les nombres de sujets dans chaque groupe. Il est intéressant de noter que la fonction *table* accepte aussi plusieurs variables en argument (essayez '*table(Gr,Sc)*'). Les scores moyens par groupe s'obtiennent par *tapply(Sc,Gr,mean)* qui signifie « appliquer la fonction *mean* aux éléments de *Sc* regroupés par les sous-groupes définis par le facteur *Gr* ». Plus généralement, *tapply* permet d'appliquer n'importe quelle fonction sur des sous-groupes des données : par exemple *tapply(Sc,Gr,var)* utilise la fonction *var* pour afficher les variances intra-groupes.

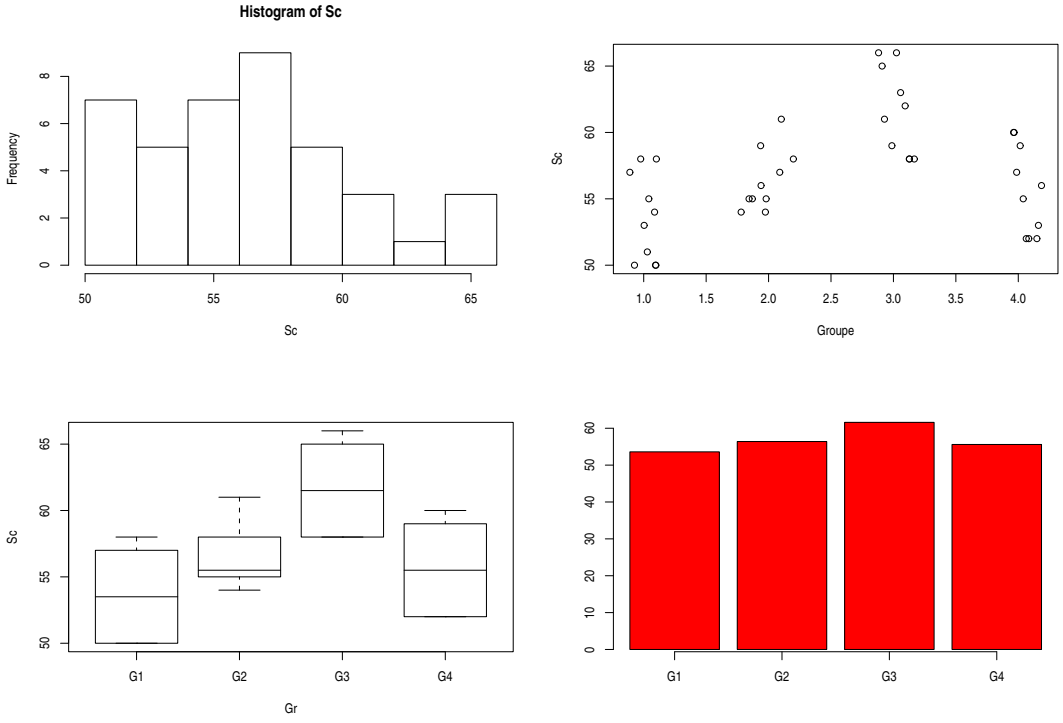
Divers graphiques sont ensuite obtenus (cf. Fig. 1). *hist(Sc)* affiche l'histogramme des scores et la ligne suivante (commande *stripchart*) présente les données brutes, groupe par groupe. Enfin, la fonction plot(Sc Gr) affiche des « boîtes à moustaches »des scores en fonctions des groupes.

barplot(tapply(Sc,Gr,mean)) affiche le diagramme en bâtons des moyennes.

L'anova est effectuée par la commande *l<-aov(Sc~ Gr)*, qui place les résultats de l'analyse dans la variable *l*. Plusieurs fonctions opèrent sur un objet de type 'aov' ; la plus importante, *summary(l)*, affiche les lignes suivantes :

```
            Df Sum Sq Mean Sq F value    Pr(>F)
Gr           3 348.80  116.27  12.182 1.184e-05 ***
```

FIG. 1 – Différents graphiques pour l'analyse des groupes indépendants



```
Residuals   36 343.60    9.54
```

On a ainsi obtenu le résultat de la comparison globale : l'effet de *Gr* est significatif au seuil 0.001 (F(3,36)=12.18). La fonction *model.tables(l)* renvoit les tailles d'effet, c'est à dire, ici, les différences entre chaque groupe et la moyenne globale (Notez qu'on aurait pu obtenir ce résultat par *tapply(Sc,Gr,mean)-mean(Sc)*). La fonction *plot(l)* fournit différents graphiques diagnostiquant de la validité du modèle (homogénéité des variances intra-groupes, distribution normale des résidus, etc...)

Le script effectue ensuite les comparaisons entre les groupes G1 et G2 d'une part (*P G1, G2* en langage VAR3), et entre G1 et l'union de G2 et G3 d'autre part (*P G1, G2 G3* en langage VAR3). Ces deux comparaisons sont « à un degré de liberté » : on peut les estimer par des tests de Student, réalisés dans R par la fonction *t.test*. Par défaut, cette fonction ne postule pas l'égalité des variances intra-groupes, contrairement à VAR3. Pour obtenir exactement le même résultat sous R que sous VAR3, il faut préciser l'option *var.equal=T* en argument de *t.test*. Pour la comparaison *P G1,G2 G3*, il suffit de créer un vecteur réunissant les scores de groupes 2 et 3, grâce à la fonction '*c(g2,g3)*' qui crée un vecteur concatenant les éléments des vecteurs *g1* et *g2*. Finalement *pairwise.test* permet d'effectuer toutes les comparaisons deux à deux entre groupes, tout en appliquant une correction pour les tests multiples. La ligne suivante montre comment l'option *subset* de *aov* permet de restreindre les données sur lesquelles s'effectue l'anova.

Parmi les tests supplémentaires que R permet d'effectuer, il y a par exemple le test de Mann-Whitney, obtenu par la procédure *wilcox.test*. Cette fonction s'utilise de façon similaire à *t.test*. Bien entendu, le script proposé n'épuise pas les possibilités d'analyses ou de graphiques. Par exemple, pour réaliser un graphique montrant les intervalles de confiances à 95% des moyennes de chaque groupe, on exécutera le code suivant :

```
m <- tapply(Sc, Gr, mean)
v <- tapply(Sc, Gr, var)
d <- qt(.95, df=10) * sqrt(v/10) # intervalles de confiance à 95%
mp <- barplot(m, ylim=c(45,65), xpd=F)
arrows(mp, m-d, mp, m+d, angle=90, code=3)
```

Si ce code paraît un peu long, il faut savoir que R permet de créer des fonctions : l'utilisateur peut écrire une fois pour toute une fonction qui affiche les moyennes avec leurs intervalles de confiance, et en la sauvant dans un fichier texte, y avoir accès pour tous nouvel ensemble de données[1] (une telle fonction,

[1] Il suffit que ce fichier soit "exécuté" avec la fonction *source*. Sous Linux/unix, le fichier .Rprofile est automatiquement exécuté au démarrage de R : c'est donc un bon endroit pour copier les fonctions qui servent souvent.

plotmeans, figure d'ailleurs dans un des modules complémentaires disponibles sur le site de R (cf. module « gregmisc »).

2 Analyse d'un plan S<A2*B2> (plan factoriel)

Considérons maintenant le cas où le plan est structuré par deux facteurs de groupes, autrement dit, où il y a deux variables entre-sujets croisées entre-elles. Nous pouvons reprendre les mêmes données que précédemment, et supposer que les quatre groupes sont définits par le croisement de deux facteurs fixes A et B à deux modalités. Les modalités de ces facteurs ne sont pas décrites dans le fichier d'entrée, mais la commande *gl* de R permet facilement de générer de nouveaux facteurs (voir `?gl`) :

```
d<-read.table('set1.dat') # lecture des données
x<-d$V2 # récupère la deuxième colonne
a<-gl(2,10,40) # création du facteur A
b<-gl(2,20,40) # création du facteur B
table(a,b)
tapply(x,list(a=a,b=b),mean)
interaction.plot(a,b,x)
l<-aov(x~a*b) # ANOVA
summary(l)
model.tables(l,se=T)
t.test(x[a==1 & b==1],x[a==1 & b==2]) # P B/A1
t.test(x[a==2 & b==1],x[a==2 & b==2]) # P B/A2
```

Cette fois encore, l'analyse de variance est effectuée par la commande *aov* : La formule '*x˜a*b*' spécifie que les deux facteurs *a* et *b* ainsi que leur interaction doivent être inclues dans le modèle statistique. On obtient :

```
            Df Sum Sq Mean Sq F value    Pr(>F)
a            1  25.60   25.60  2.6822 0.1101890
b            1 129.60  129.60 13.5786 0.0007477 ***
a:b          1 193.60  193.60 20.2841 6.768e-05 ***
Residuals   36 343.60    9.54
```

On constate que l'effet de *b* et l'interaction (notée *a :b* par R) sont significatifs. Les deux lignes finales du script effectuent les tests de Student qui examinent les effets de *b* restreints à chaque modalité de *a*.

Si le plan est équilibré, c'est à dire si tous les groupes contiennent le même nombre de sujets, alors les résultats fournis par la fonction *aov* sont rigoureusement identiques à ceux de VAR3. Dans le cas des plans déséquilibrés, par contre, il faut savoir que la fonction *aov* de R calcule des sommes de carrées qui correspondent à la comparaison pondérée par les effectifs. En fait, les sommes de carrés

calculées sont séquentielles : l'effet d'un facteur est testé après avoir enlevé les effets des facteurs plus à gauche que lui dans le modèle statistique. Par conséquent, l'ordre de déclaration des facteurs dans le modèle importe : seul le test F associé au facteur le plus à gauche de la formule (e.g. a dans a*b) correspondra à celui fournit par VAR3 (pour la solution pondérée). Si le plan est déséquilibré, il faudra donc appeler la fonction *aov* autant de fois qu'il y a de facteurs et placer ceux-ci succesivement en première place. Ainsi, pour une plan déséquilibré à deux facteurs A et B, il faudra appeler examiner l'effet de *a* dans *aov(x˜a*b)* et l'effet de *b* dans *aov(x˜b*a)*. Dans tous les cas, les valeurs fournies par R correspondent à la comparaison dite «pondérée par les effectifs».

Pour des tests correspondant à des comparaisons équipondérées, il faut utiliser des sommes de carré dites de "type III" qui peuvent être calculées grâce à la librairie *car* (Fox [2002]).

```
library(car)
contrasts(a)<-contr.sum
contrasts(b)<-contr.sum
l<-aov(x~a*b)
Anova(l,type='III')
```

On constate l'importance de la formule du modèle statistique fournie à la fonction *aov*. C'est par elle qu'on spécifie le plan expérimental et, éventuellement, les sources adjointes ou termes d'erreur. Ce système permet de spécifier des modèles plus simples que le modèle complet qui est implicite dans VAR3. Par exemple, dans le cas d'un plan S<A∗B>, s'il y a des raisons a priori de supposer que les interactions sont inexistantes, on peut préférer un modèle additif *x˜a+b* au modèle non-additif *x˜a*b*.

3 Analyse d'un plan S*A3 (mesures répétées)

Ce troisième exemple met en oeuvre une variable intra-sujet à trois modalités. Le fichier de données reproduit la table 2 : il contient une série de lignes contenant chacune les trois scores de chaque sujet. La première ligne du fichier contient les noms de colonnes « a1, a2 et a3 ».

Le script d'analyse est le suivant :

```
s<-read.table('set3.dat',header=T)
attach(s)
suj<-gl(10,1,30)
cond<-gl(3,10,30)
x<-c(a1,a2,a3)
tapply(x,cond,summary)
interaction.plot(cond,suj,x)
```

TAB. 2 – Mesures répétées

a1	a2	a3
316	376	354
297	373	268
331	335	332
348	335	391
311	379	369
271	338	357
334	365	351
383	389	332
300	393	374
381	379	375

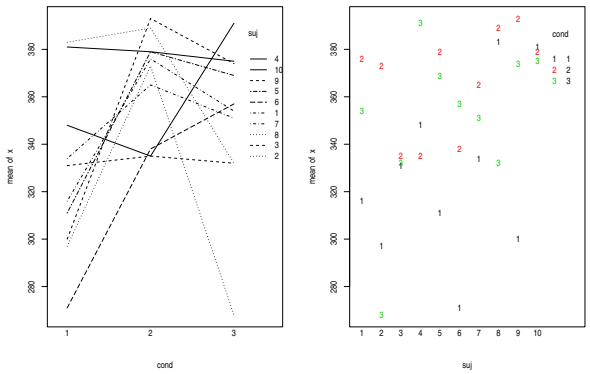
```
interaction.plot(suj,cond,x,type='p',col=1:3)
barplot(c(mean(a1),mean(a2),mean(a3)))
summary(aov(x~cond+Error(suj/cond))) # ANOVA
t.test(a1,a2,paired=T) # P A1 A2
t.test(a1,a3,paired=T) # P A1 A3
t.test(a2,a3,paired=T) # P A2 A3
```

Le paramètre *header=T* indique que la première ligne du fichier contient les noms de colonnes. Après avoir généré les deux facteurs *suj* et *cond*, on regroupe les données dans le vecteur '*x*'. On utilise ensuite la fonction *interaction.plot* pour afficher graphiquement les résultats individuels (Fig. 2).

Quand le plan présente, comme ici, un ou plusieurs facteurs intra-sujets, il faut le spécifier dans un terme *Error* du modèle statistique fournit à la fonction *aov* : si le facteur sujet est représenté par le vecteur *suj*, et les facteurs intra-sujets sont a, b, c, d, le terme passé à '*Error*' doit être '*suj/(a*b*c*d)*'. Cela permet à R de déterminer la source adjointe adéquate pour chaque source de variation. La sortie d'*aov* montre que le facteur *cond* a pour source adjointe l'interaction *suj :cond*, et que son effet est significatif :

```
Error: suj:cond
          Df  Sum Sq Mean Sq F value  Pr(>F)
cond       2  7691.4  3845.7  4.3734 0.02831 *
Residuals 18 15827.9   879.3
```

FIG. 2 – Affichage des données individuelle dans un plan S*A3



4 Analyse d'un plan S<G2>*B4

Cet exemple concerne un plan « split-plot », avec un facteur de groupe binaire A et un facteur intra-sujet B à quatre modalités : le plan est S<A2> $*B4$.

Pour la première fois dans cette note, on suppose que le fichier d'entrée contient des données brutes, essai par essai. Cela donne l'occasion de montrer que R permet également d'effectuer les prétraitements sur les données. Les premières lignes du fichier de données sont listées ci-dessous : La première colone contient un code pour le sujet, la seconde un code pour le facteur de groupe '*gr*', la troisième code le facteur répété '*lg*', et la quatrième contient une réponse binaire (1=réponse correcte, 0=réponse erronée).

```
c01 c 4 1
c01 c 3 1
c01 c 2 1
c01 c 1 1
c01 c 4 0
c01 c 1 1
c01 c 1 0
c01 c 2 1
...
```

On désire analyser les taux de réponse correctes, en fonction de *gr* et *lg*. Il faut donc, dans un premier temps, les calculer, sujet par sujet, et à l'intérieur des sujets, pour chaque modalité de *lg*. Cela est réalisé par la fonction *aggregate* qui génére un nouveau tableau de données, dans lequel les moyennes se retrouvent dans la colonne nommée '*x*'.

```
a<-read.table('splitplot.dat')
names(a)<-c('suj','gr','lg','hit')
attach(a)
table(gr)
table(suj)
table(suj,lg)
b<-aggregate(hit,list(suj=suj,gr=gr,lg=lg),mean)
attach(b)
interaction.plot(lg,gr,x)
summary(aov(x~gr*lg+Error(suj/lg)))
t.test(x[gr=='c' & lg==1],x[gr=='s' & lg==1]) # P G/B1
t.test(x[gr=='c' & lg==2],x[gr=='s' & lg==2]) # P G/B2
t.test(x[gr=='c' & lg==3],x[gr=='s' & lg==3]) # P G/B3
t.test(x[gr=='c' & lg==4],x[gr=='s' & lg==4]) # P G/B4
t.test(x[gr=='c' & lg==1],x[gr=='c' & lg==2],paired=T) # P B1 B2/G1
t.test(x[gr=='s' & lg==1],x[gr=='s' & lg==2],paired=T) # P B1 B2/G1
```

La commande "*names*" permet de modifier les noms des colonnes du tableau sauvé dans la variable *a*. Rapellons que si les noms avaient été présents sur la première ligne du fichier `splitplot.dat`, il aurait suffit de passer le paramètre *header=T* à la fonction *read.table*.

Pour pouvoir évaluer les effets principaux ainsi que l'interaction des facteurs '*gr*' et '*lg*', la formule passée à *aov* est '*gr*lg*'. Le seul facteur intra-sujet étant *lg*, le terme d'erreur passé à *aov* est "*Error(suj/lg)*".

On utilise la fonction *t.test* pour effectuer les comparaisons restreintes, mais notez qu'on aurait pu utiliser l'option *subset* de *aov*. Par exemple, la ligne suivante teste l'effet de '*gr*' restraint aux données telles que '*lg==1*' :

```
summary(aov(x~gr,subset=lg==1))
```

5 Analyse d'un plan hiérarchique S<B<A>>

Dans cet exemple (extrait de Kennedy and Bush [1985], p.501), le but est de comparer deux méthodes d'enseignement (facteur A). Six groupes de 4 élèves sont formés, dont trois recoivent un enseignement suivant la méthode A1 et trois suivant la méthode A2. Les enseignements sont dispensés par six professeurs différents (facteur B). La table 3 fournit les résultats au test d'évaluation.

Le plan d'analyse est S<B<A>>, S et B étant deux facteurs aléatoires, et A fixe. Dans un tel plan, l'effet de A a pour source adjointe « B »alors que l'effet de B à pour source adjointe « S ». Le lecteur pourra se reporter à Kennedy and Bush [1985] et Abdi [1987] pour des discussions sur le choix des sources adjointes pertinentes pour différent plans expérimentaux.

TAB. 3 – Plan hiérachique

Traitement	Groupe	Scores				Moyennes
A1	B1	9	7	10	8	8.5
A1	B2	10	14	8	12	11
A1	B3	8	3	3	7	5.25
A2	B4	12	14	10	14	12.5
A2	B5	11	13	9	9	10.5
A2	B6	14	10	9	14	11.75

Dans le script d'analyse ci-dessous, comme les données sont peu nombreuses, nous les entrons directement dans une variable plutôt que de les lire dans un fichier. La fonction *aov* ne permettant pas de fournir plusieurs sources adjointes, il faut l'appeler deux fois avec chacun des termes pertinents pour effectuer l'analyse :

```
x<-scan("")
9 7 10 8 10 14 8 12 8 3 3 7
12 14 10 14 11 13 9 9 14 10 9 14

a<-gl(2,12,24)
b<-gl(6,4)

summary(aov(x~a+Error(b)))# effet de A
summary(aov(x~a/b)) # effet de B emboité dans A
```

Le première appel à *aov* fournit l'effet de A (F(1,4)=3.57 ns.), et le second fournit l'effet de B (ligne « a :b » : F(4,18)=3.8 p<.05).

6 Régression linéaire

Le fichier `simple_regress.tab` contient 2 colonnes de chiffres, correspondant respectivement à des variables *x* et *y*. On souhaite effectuer la regréssion linéaire de *y* en fonction de *x*.

```
a<-read.table("simple_regress.tab",col.names=c('x','y'))

# do the linear regression & print the outcome
l=lm(y~x)
summary(l)
```

```
# graphics
par(ask=F)
plot(x,y,pch=21,bg=3)
abline(l$coeff)

# misc. diagnostic plots
plot(l)
```

La fonction *lm* permet aussi d'effectuer des regressions linéaires multiples :

```
a<-read.table("multi_regress.tab",col.names=c('x','y','z'))
attach(a)

# graphics
pairs(a)

# do the linear regression & print the outcome
l1=lm(z~x+y)
summary(l1)

l2=lm(z~y+x)
summary(l2)

# are the regressors correlated?
cor.test(x,y)
```

7 Analyse de covariance

L'analyse de covariance regroupe l'analyse de variance et la régression. Supposons qu'on veuille comparer les scores obtenus à un test par trois groupes de sujets indépendants. Supposons également qu'avant ce test, les sujets avaient participé à des prétests dont les résultats prédisaient, en partie, les résultats au test final. Pour comparer les trois groupes, il peut être intéressant de prendre en compte les scores du prétest, pour « ajuster »les scores du tests.

Les données présentées dans la table 4 (extraite de Kennedy and Bush [1985], p.411) sont sauvées dans un fichier à deux colonnes, la première contenant les scores du prétest, et la seconde les score du test. Le script d'analyse est le suivant :

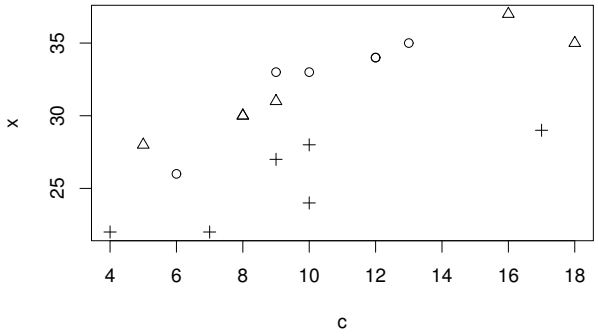
```
a<-read.table('set4.dat',col.names=c('c','x'))
attach(a)
g<-gl(3,6,18)
plot(c,x,pch=codes(g))
summary(aov(x~c+g))
```

La variable *c* contient les résultats du prétest, et la variable *g* définit les groupes. L'argument *pch* passé à la fonction *plot* (Fig. 3) permet de représenter les données

TAB. 4 – Données pour l'analyse de covariance

Group 1		Group 2		Group 3	
Pretest	Test	Pretest	Test	Pretest	Test
12	34	18	35	10	28
6	26	8	30	4	22
9	33	16	37	10	24
13	35	5	28	17	29
12	34	9	31	9	27
10	33	8	30	7	22

FIG. 3 – Graphique pour l'analyse de covariance



des trois groupes avec des symboles différents. La fonction *aov* R implémente le modèle linéaire général : elle permet donc de spécifier aussi bien des variables continues que des variables discrètes dans le modèle. Pour notre analyse de covariance, le modèle statistique pertinent est *x˜c+g*. On remarquera que ce modèle est formellement identique au modèle non-additif pour l'analyse factorielle à deux facteurs. Toutefois, dans le cas de l'analyse de covariance, la première variable n'est pas un facteur, mais une grandeur continue. Le résultat de cette analyse révèle que l'effet de *g* après avoir pris en compte celui de *c*, est significatif (F(2,14)=27.4).

```
            Df  Sum Sq Mean Sq F value    Pr(>F)
c            1 157.503 157.503  56.873 2.704e-06 ***
```

```
g            2 151.504  75.752  27.353 1.459e-05 ***
Residuals   14  38.771   2.769
```

8 Définition de fonctions

Dans certains cas, les psychologues aiment bien appliquer des « censures »aux données, c'est à dire qu'ils rejetent les points qui sont « trop éloignés »de la moyenne, considérant que ces données sont aberrantes. Cela fournit l'occasion de montrer avec quel facilité R permet de créer une nouvelle fonction qui calcule la moyenne après suppression des données situées à plus de deux écart-types :

```
clmean <- function (x) {
  keep <- (x-mean(x))/sd(x) < 2
  mean(x[keep])
}
```

Cette fonction prend comme argument un vecteur qu'elle copie dans la variable '*x*'. Elle évalue, pour chaque élément de *x*, si sa distance à la moyenne de *x*, divisée par l'écart-type de *x*, est inférieur à 2. Le résultat est un vecteur de valeurs logiques (TRUE ou FALSE) de même longeur que *x*, placée dans la variable *keep*. Ensuite, l'expression *x[keep]* extrait de *x* les éléments pour lesquels *keep* est vrai (TRUE). Par conséquent la dernière expression *mean(x[keep])* calcule la moyenne désirée.

Une fois cette définition sauvée dans un fichier, et celui-ci lu par la fonction *source*, la fonction *clmean* devient alors disponible et peut-être employée en lieu et place de la fonction *mean*.

9 Conclusion

Résumons la mise en oeuvre d'une analyse de variance sous R : il faut tout d'abord créer autant de vecteurs qu'il y a de facteurs dans le plan (y compris un vecteur pour le facteur sujet qui indispensable pour les plans avec mesures répétés). Ces vecteurs ont tous la même longueur : celle du vecteur de données. Si l'on place en colonne les vecteurs correspondant aux facteurs et le vecteur de données : les valeurs des facteurs sur une ligne définissent les modalités correspondantes au point de donnée placé sur la même ligne. Si le fichier d'entrée ne contient pas les données exactement dans le « bon »format, les fonctions *aggregate*, *gl* et *stack* s'avéreront souvent utiles à cette étape.

Dans une seconde étape, on utilise les fonctions *table* et *tapply*, ainsi que les fonctions graphiques pour explorer les données.

Finalement l'anova est effectuée en appelant la fonction *aov* avec la formule adéquate. Les comparaisons à un degré de liberté sont effectuées par des appels à la fonction *t.test*. Les autres comparaisons se font en rappelant la fonction *aov* sur des sous-ensembles des données avec l'argument *subset*.

Les exemples de plans présentés dans cet article recouvrent une grande partie des analyses qui se rencontrent en psychologie expérimentale. Ils ne sont en rien des cadres figés qu'il faut reproduire pour chaque analyse, mais plutôt des exemples que l'utilisateur devra adapter à ses propres besoins.

Une limitation des plans présentés, mis à part le cas du plan hiérarchique, est qu'ils ne comportent que des facteurs fixes hormis le facteur sujet. Pour les plans comportant des facteurs aléatoires, la fonction *aov* est insuffisante, et l'on devra alors utiliser le module « nlme »décrit dans Pinheiro and Bates [2000].

Appendice

Téléchargement de R

R peut être librement téléchargé sur le site `http\unskip\penalty\@M://cran.r-project.org`. En plus du système de base et des documentations associées, il est fortement conseillé de télécharger aussi les diverses documentations dans la section « contributed docs »(`http\penalty\@M://cran.r-project.org/other-docs.html`) notamment :

- *R pour les débutants* par Emmanuel Paradis
- *Introduction au système R* par Yves Brostaux.
- *Notes on the use of R for psychology experiments and questionnaires* par J. Baron et Y. Li.

Dans le menu "Package sources", plusieurs modules d'extensions présentent également de l'intérêt pour les psychologues : foreign, gregmisc, multcomp, nlme, VR.

Finalement, afin que le lecteur puisse reproduire les analyses présentées dans cet article, les fichiers de données sont disponibles par simple demande par email auprès de l'auteur (`pallier@lscp.ehess.fr`).

Références

Herve Abdi. *Introduction au traitement statistique des données expérimentales*. Presses Universitaires de Grenoble, Grenoble, 1987.

Jonathan Baron and Yuelin Li. Notes on the use of R for psychology experiments and questionnaires. disponible à http ://www.psych.upenn.edu/ baron/rpsych.htm, 31 decembre 2000.

Julian Faraway. *Practical Regression and Anova using R*. available at www.r-project.org, 2002.

John Fox. *An R and S-PLUS Companion to Applied Regression*. SAGE, 2002.

J. J. Kennedy and A. J. Bush. *An Introduction to the Design and Analysis of Experiments in Behavioral Research*. University Press of America, Lanham, MD, 1985.

A. Krause and M. Olsen. *The Basics of S and S-PLUS*. Springer, 1997.

J. C. Pinheiro and D. M. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000.

An introduction to R. R development core team, 1.1.1 edition, 2000.

P. Spector. *An introduction to S and S-PLUS*. Duxbury Press, 1994.