Statistiques
2010-2011

TP sur le Modèle linéaire gaussien avec R

1 Les exercices

Vous traiterez les exercices suivants avec le logiciel `R`.

Exercice 1

Des photographies aériennes de champs d'orge sont analysées au photomètre, qui mesure la brillance de chaque champ. On recherche la relation entre les rendements des parcelles et la brillance, en tenant compte de l'application d'un fongicide sur certaines parcelles pour combattre le mildiou. Les données sont dans le fichier `fongicide.rda`, sous la forme : `(rendement, brillance, fongicide)`. On observe donc $(R_i, B_i, F_i)_{i=1,\dots,72}$, où i désigne l'indice de la parcelle, R_i le rendement de la parcelle i, B_i sa brillance et F_i vaut 1 si un fongicide est appliqué sur la parcelle i et 0 sinon.

1. Charger les données (cf. section 2.2) et tracer sur un même graphe les rendements en fonction de la brillance, en coloriant les points avec deux couleurs différentes en fonction du groupe auxquels ils appartiennent (i.e avec ou sans fongicide), cf. section 2.8.
2. On considère alors le modèle linéaire suivant. Pour les i tels que $F_i = 1$ (parcelle avec fongicide),
$$R_i = a_1 + b_1 B_i + \varepsilon_i \ ,$$
et pour les i tels que $F_i = 0$ (parcelle sans fongicide),
$$R_i = a_0 + b_0 B_i + \varepsilon_i \ ,$$
où a_0, a_1, b_0, b_1 sont des coefficients réels inconnus, et les ε_i sont i.i.d de loi $\mathcal{N}(0, \sigma^2)$ avec σ^2 inconnu.
   (a) Ecrire le modèle sous forme matricielle :
$$R = X\beta + \varepsilon \ ,$$
   et ajuster le modèle dans R, cf. section 2.3.
   (b) Calculer les estimations $\widehat{\beta}$, $\widehat{\sigma}$ et les *valeurs ajustées* $\widehat{m} = X\beta$, cf. section 2.7. Ajouter sur le graphique de la première question les valeurs ajustées, qui doivent former deux droites de régressions correspondant aux deux groupes (avec ou sans fongicide).
   (c) Vérifier rapidement l'allure des résidus (cf. section 2.4).
   (d) Tester l'absence d'effet du fongicide sur le rendement. (cf. section 2.5)

Exercice 2

Une expérience est destinée à étudier l'adaptation de deux variétés de moutarde à la sécheresse. On a un dispositif à 4 traitements différents dérivant de 2 variétés (Clause et Gisilba) notées A et B et de 2 intensités lumineuses (29000 lux et 8000 lux) notées X et Y. On dispose de 4 plants pour chacun des traitements AX, AY, BX et BY. Un indicateur de l'adaptation à la sécheresse

est l'apparition de racines courtes tubérisées. Le tableau suivant indique le nombre observé de ces racines.

AX :	78	79	44	77
AY :	64	96	30	20
BX :	137	85	302	315
BY :	64	67	102	47

Les données sont dans le fichier `moutarde.rda`, sous la forme `(variete, lumiere, tubercules)`. On considère un modèle d'*analyse de la variance à deux facteurs complet et équilibré*, c'est à dire qu'on observe le nombre de tubercules $Y_{i,j,k}$ pour i valant A ou B (désignant la variété), j valant X ou Y (désignant la luminosité) et $k \in \{1, \ldots, 4\}$ désignant le numéro du plant. On suppose que :

$$Y_{i,j,k} = \mu + \alpha_i + \beta_j + \gamma_{i,j} + \varepsilon_{i,j,k}.$$

où les variables $\varepsilon_{i,j,k}$ sont indépendantes, de loi $\mathcal{N}(0, \sigma^2)$, σ^2 étant inconnu.

1. Ecrire le modèle sous forme matricielle :

$$Y = X\beta + \varepsilon\ .$$

   Calculer les *valeurs ajustées* $\widehat{m} = X\beta$, et vérifier rapidement l'allure des résidus.
2. Tester, au niveau 5%, l'absence d'interaction entre les deux facteurs.
3. Tester l'absence d'effet du facteur variété sur l'adaptation à la sécheresse.

Exercice 3

On veut comparer quatre variétés de céréales sur trois types de terrains. Les mesures Y_{ijk} , $1 \leqslant k \leqslant n_{ij}$ de rendement de la variété i de céréale sur le terrain j sont dans le fichier `cereales.rda`, sous la forme `(variete, terrain, rendement)`. On considère un modèle complet d'analyse de la variance à deux facteurs.

1. Calculer les *valeurs ajustées* et vérifier rapidement l'allure des résidus.
2. Tester l'absence d'effet du facteur terrain.
3. Tester l'absence d'effet du facteur variété.

2 Le modèle linéaire gaussien avec le logiciel R

2.1 Démarrage

Deux possibilités :

- vous lancez R dans un terminal en tapant `R`. Dans une autre fenêtre, vous éditez votre script dans un éditeur de texte, et vous copiez-collez les lignes nécessaires. Vous pouvez également charger un fichier dans R avec la commande `source(‘‘mon_fichier’’)`,
- ou bien vous éditez vos programmes en R avec `emacs`, muni du package ESS (présent sur les machines de la fac). Pour que la syntaxe R soit reconnue, il faut que la ligne suivante soit dans votre fichier `.emacs`, qui doit être dans votre dossier “home” :
  `(global-font-lock-mode t)` L'avantage de cette solution est que R peut être lancé dans emacs, et on peut alors charger les lignes une par une, sélectionner un ensemble de lignes de codes etc.

2.2 Charger les données

Tout d'abord, on peut charger des données en R avec la commande `load` si ces données sont au format R. L'objet chargé a alors un nom, on peut récupérer directement l'objet avec la commande `get` :
`datafong <- get(load("fongicide.rda"))`
(Remarque : on peut aussi lire un fichier de données sous forme de tableau dans un fichier texte avec la commande `read.table`). Les données de ce TP venant d'un fichier `.rda` ont le format de tableau de données (`data.frame`) et leurs colonnes ont un nom. Par exemple, la colonne brillance dans l'exemple ci-dessus est accessible par `datafong$brillance` On peut, pour plus de facilité d'accès, rendre ces vecteurs accessibles par leur nom directement avec la commande `attach(datafong)`.

2.3 Construire le modèle : la commande `lm`

La commande `lm` sert à construire le modèle linéaire désiré sur nos données, en utilisant des formulations symboliques. On l'utilise sous la forme `out <- lm(formula,data)`, où `formula` est une description symbolique du modèle et `data` est notre tableau de données. Un résumé de l'ajustement est obtenu avec `summary(out)`.

Il reste à expliquer la forme du modèle donné par `formula`. Un modèle typique a la forme `réponse` $\sim$ `termes` où `réponse` est le vecteur des réponses (ce qu'on veut expliquer, par exemple le rendement ci-dessus) et `termes` est une suite de termes qui spécifie un prédicteur linéaire pour la réponse. Une spécification de la forme `premier + second` indique tous les termes du tableau `premier` avec tous ceux du tableau `second`, en éliminant les doublons éventuels. Une spécification de la forme `premier:second` demande de considérer les interactions de `premier` et `second`. Enfin, `premier*second` demande de prendre en compte à la fois les effets individuels et les effets croisés de `premier` et `second`. Cela revient au même que de considérer `premier + second + premier:second`. Une formule contient toujours un terme constant implicite. Pour l'enlever, on peut écrire `-1`, comme dans `y` $\sim$ `x - 1`. Donnons quelques exemples pour bien comprendre. Dans ce qui suit, `fongicide` est une variable catégorielle, `rendement`, `brillance`, `x` et `t` sont quantitatives.

- `rendement` $\sim$ `fongicide + brillance:fongicide` désigne le modèle :

$$R_i = \mu + a_0 \ \mathbb{1}_{F_i=0} + a_1 \ \mathbb{1}_{F_i=1} + b_0 \ \mathbb{1}_{F_i=0} B_i + b_1 \ \mathbb{1}_{F_i=1} B_i + \varepsilon_i \ .$$

  Comme ce modèle n'est pas identifiable, R ajoute une contrainte d'identifiabilité entre a_0 et a_1.
- `rendement` $\sim$ `fongicide + brillance:fongicide-1` désigne le modèle :

$$R_i = a_0 \ \mathbb{1}_{F_i=0} + a_1 \ \mathbb{1}_{F_i=1} + b_0 \ \mathbb{1}_{F_i=0} B_i + b_1 \ \mathbb{1}_{F_i=1} B_i + \varepsilon_i \ .$$

  Ce modèle-ci est identifiable.
- `rendement` $\sim$ `brillance` désigne le modèle :

$$R_i = a + bB_i + \varepsilon_i \ .$$

- `x` $\sim$ `t+I(t^2)` désigne le modèle :

$$x_i = a_0 + a_1 t_i + a_2 t_i^2 + \varepsilon_i \ .$$

  Notons que dans ce cas, on doit utiliser `I()` pour que `R` interprète `t^2` comme une opération arithmétique, et non comme une opération symbolique, i.e un terme `t*t`.

Remarquons enfin qu'il est souhaitable, lorsqu'on commence à étudier un modèle, d'examiner rapidement si le modèle est meilleur que le modèle "trivial" où les observations sont gaussiennes i.i.d de moyenne inconnue, cf. section 2.5.

2.4 L'examen des résidus

Si `outV` est un modèle donné par `lm`, on obtient les résidus standardisés avec `rstandard(outV)` et les valeurs ajustées avec `fitted(outV)`.

2.5 Tests de sous-hypothèse linéaire : la commande `anova`

Elle permet de faire un test de sous-hypothèse linéaire, à partir de modèles et sous-modèles construits par `lm`. Par exemple :

```
outV <- lm(rendement ~ fongicide + brillance:fongicide-1)
outW <- lm(rendement ~ brillance)
anova(outW,outV)
```

teste l'absence d'effet du fongicide, c'est à dire l'appartenance au modèle `rendement ~ brillance` et renvoie la *table d'analyse de la variance* suivante :

```
 Analysis of Variance Table

Model 1: rendement ~ brillance
Model 2: rendement ~ fongicide + brillance:fongicide - 1
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     70 7.6731
2     68 7.0674  2   0.60573 2.9141 0.06106 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Que l'on interprète ainsi, dans le cadre d'un test de $m \in W$ contre $m \in V \setminus W$, avec W de dimension q et V de dimension p :

```
 Analysis of Variance Table

Model 1: W
Model 2: V
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	$n-q$	$\|Y-\Pi_W Y\|^2$				
2	$n-p$	$\|Y-\Pi_V Y\|^2$	$p-q$	$\|\Pi_V Y-\Pi_W Y\|^2$	$\frac{\|\Pi_V Y-\Pi_W Y\|^2/(p-q)}{\|Y-\Pi_V Y\|^2/(n-p)}$	p-valeur

Remarquons enfin qu'il est souhaitable, lorsqu'on commence à étudier un modèle, d'examiner rapidement si le modèle est meilleur que le modèle "trivial" où les observations sont gaussiennes i.i.d de moyenne inconnue. On peut faire ceci :

```
out.trivial <- lm(rendement ~ 1)
outV <- lm(rendement ~ fongicide + brillance:fongicide-1)
anova(out.trivial,outV)
```

2.6 Une alternative à `lm` : la commande `aov`

C'est un substitut à `lm`, qui en réalité utilise `lm` avec des paramètres différents des paramètres par défaut. Elle est particulièrement adaptée lorsque l'on veut effectuer une analyse de la variance à un ou plusieurs facteurs. La fonction `model.tables` donnera les estimations des "effets" de chaque facteur. Par exemple :

```
datacer <- get(load("../donnees/cereales.rda"))
attach(datacer)
aovV <- aov(rendement ~ terrain * variete)
```

```
model.tables(aovV)
```
renvoie une table contenant les estimations des coefficients devant chaque terme du modèle d'analyse de la variance complet sur les données `cereales.rda`.

2.7 L'estimation des paramètres

Si `outV` est un modèle donné par `lm`, la valeur de $\|Y - \Pi_V Y\|^2$ peut être obtenue avec `deviance(outV)` (ou bien avec `sum(residuals(outV)^2)`), et la valeur de $n - dim(V)$ par `df.residual(outV)`. Il n'y a plus qu'à faire le rapport pour obtenir l'estimateur sans biais de σ^2.

Les autres paramètres estimés sont obtenus avec `coef(outV)` et des intervalles de confiances (non simultanés !) avec `confint(outV)`. Pour construire des intervalles de confiance simultanés, il est utile de connaître la matrice des variances-covariances estimées des paramètres, qui est obtenue avec `vcov(outV)`. En effet, cette matrice est $\widehat{\sigma}^2({}^tXX)^{-1}$, et on peut montrer qu'avec probabilité $1-\alpha$, on a, pour tout $i = 1, \ldots, p$,

$$|\widehat{\beta}_i - \beta_i| \leqslant \sqrt{p\widehat{\sigma}^2({}^tXX)^{-1}_{ii} f_{1-\alpha,p,n-p}}\ ,$$

où p désigne la dimension de V et $f_{1-\alpha,p,n-p}$ le quantile d'ordre $(1-\alpha)$ de la loi de Fisher de paramètres $(p, n-p)$.

2.8 Graphiques

- ouvrir une fenêtre : `x11()`
- tracer un nuage de points (liés entre eux ou non) : `plot()`
- ajouter un graphe sur un graphe existant : `lines()` pour des points liés et `points()` pour des points non liés.
- inscrire une légende : `legend()`

Exemple : Dans `plot()`, l'option `type='l'` lie les points par des segments de droite, `type='p'` ne les lie pas. L'option `pch` précise la forme des points et `col` attribue la couleur. Par exemple, si x, y, et z sont trois vecteurs de même longueur, où z contient un facteur, la commande suivante tracera un uage de points de coordonnées (x_i, y_i) et chaque point sera colorié en bleu ou en rouge selon le niveau z_i.

```
plot(x,y,pch="+",col=c("blue","red")[unclass(z)])
legend("bottomright",pch="+",col=c("blue","red"),legend=c("sans fongicide","avec
fongicide"))
```

Remarquons que les niveaux d'un facteur sont ordonnés, et représentés intérieurement par des entiers, auxquels on accède par `unclass()`. On peut connaître les niveaux d'un facteur, et leur ordre, avec `levels()`.