Le modèle linéaire avec R

Fabrice Rossi

31 mars 2003

1 Le modèle linéaire simple

1.1 Introduction

Le logiciel R propose une fonction `lm` qui permet d'estimer les paramètres d'un modèle linéaire. Voici un exemple d'utilisation :

exemple-simple.R

```
1  # création des données
2  base=runif(20,min=-2,max=2)
3  mydata=data.frame(x=base,y=2*base+1+rnorm(length(base),sd=0.2))
4
5  # estimation du modèle
6  model=lm(y~x,data=mydata)
7
8  # coefficients obtenus
9  print(model)
```

Certains éléments de l'exemple méritent des explications :

- le modèle linéaire est estimé de préférence à partir d'une `data.frame`. C'est le sens du paramètre `data` de `lm` ;
- on précise ce qu'on cherche à faire par l'intermédiaire d'une formule R qui est interprétée dans le contexte de la `data.frame` : dans l'exemple `x` et `y` sont donc des raccourcis pour `mydata$x` et `mydata$y` ;
- les formules de R permettent des constructions assez sophistiquées. Nous nous contenterons des constructions de base. Dans l'exemple proposé, `y~x` signifie qu'on cherche à représenter `y` comme une fonction affine de `x`. Pour obtenir une fonction linéaire de `x` (pas de terme constant), on utilise la formule `y~x+0` ;
- on peut réaliser des transformations arbitraires sur les formules, par exemple utiliser `y~log(x)` pour indiquer qu'on cherche une relation affine entre `y` et `log(x)`.

1.2 Utilisation du modèle

L'objet modèle linéaire obtenu peut être bien entendu utilisé de diverses manières. On dispose par exemple des fonctions suivantes (qui prennent pour paramètre un modèle linéaire) :

- `coef` (ou `coefficients`) : renvoie un vecteur contenant les estimateurs des coefficients du modèle linéaire
- `resid` (ou `residuals`) : renvoie un vecteur contenant les erreurs de prédiction réalisées par le modèle (les erreurs résiduelles)
- `deviance` : renvoie la somme des carrés des erreurs résiduelles

La fonction la plus utile dans la pratique est sûrement `predict` qui permet d'appliquer le modèle sur de nouvelles données pour réaliser des prédictions. Considérons la suite de l'exemple précédent :

exemple-simple.R

```
11 # nouvelles données
12 newdata=data.frame(x=seq(-2,2,length=51))
13
14 # calcul des prédictions
```

```
result=predict(model,newdata)

# dessin
plot(newdata$x,result,type="l")   # prédiction
points(mydata,col="blue") # observation
abline(1,2,col="red")    # modèle réel
```

Deux points importants méritent d'être soulignés :
- pour réaliser une prédiction, on doit transmettre à `predict` une `data.frame` contenant au moins les variables utilisées pour construire le modèle (variables explicatives) ;
- `predict` renvoie un vecteur (ou une matrice) de prédiction.

De plus, `predict` peut être utilisé pour calculer des intervalles de confiance comme dans la suite de l'exemple :

exemple-simple.R

```
# intervalles de confiance
p.conf=predict(model,newdata,interval="confidence")
p.int=predict(model,newdata,interval="prediction")
matplot(newdata$x,cbind(p.conf,p.int[,-1]),type="l")
```

Deux modes de calcul sont possibles :
- les intervalles de confiance (`confidence`) correspondent à l'intervalle auquel $E(Y|X = x)$ appartient avec une probabilité de 0.95 (on peut changer cette valeur grâce au le paramètre `level`) ;
- les intervalles de prédiction (`prediction`) correspondent à l'intervalle auquel, sachant $X = x$, Y appartient avec une probabilité de 0.95 (i.e., l'intervalle de prédiction prend en compte le bruit).

Attention, ces intervalles ne sont valables que si les hypothèses du modèle linéaire sont vérifiées (bruit gaussien, homoscédasticité, etc.).

1.3 Exercices

Exercice 1.1

On considère le modèle linéaire $Y \sim 2X + 1 + \mathcal{N}(0, \frac{1}{4})$. Vérifier par simulation que les intervalles de confiance calculés par `predict` sont satisfaisants quand on considère 20 réalisations, avec X distribué uniformément dans $[-5, 5]$.

Exercice 1.2

On considère le modèle linéaire $Y \sim 2X + 1 + \frac{|X|}{2}\mathcal{N}(0, \frac{1}{4})$. Ce modèle est hétéroscédastique, c'est-à-dire que la variance du bruit dépend de X. Comparer par simulation les intervalles de confiance réels avec ceux calculés par `predict`.

2 Régression linéaire multiple

2.1 Principe

Les formules de R permettent de chercher une relation linéaire entre une variable cible et *plusieurs* variables explicatives. Il suffit pour de faire apparaître les différentes variables dans la formule, comme dans l'exemple suivant :

exemple-deux-variables.R

```
# création des données
base.x=runif(20,min=-2,max=2)
base.y=runif(20,min=-2,max=2)
mydata=data.frame(x=base.x,y=base.y,z=2*base.x-0.5*base.y+1+rnorm(length(base.x),sd=0.2))

# estimation du modèle
model=lm(z~x+y,data=mydata)

```

```
# coefficients obtenus
print(model)
```

Il est parfois fastidieux d'écrire une formule comportant de nombreux termes. Il est conseillé d'utiliser la fonction `paste` (concaténation de chaîne de caractères) pour fabriquer la formule automatiquement. L'appel suivant, par exemple, fabrique un modèle linéaire pour 25 variables (de `x1` à `x25`) :

```
as.formula(paste("y ~ ", paste(paste("x", 1:25, sep=""), collapse= "+")))
```

On peut aussi regrouper toutes les variables explicatives sous forme d'une matrice, comme l'illustre l'exemple suivant :

exemple-deux-variables-mat.R

```
# création des données
base.x=runif(20,min=-2,max=2)
base.y=runif(20,min=-2,max=2)
M=cbind(base.x,base.y)
z=2*base.x-0.5*base.y+1+rnorm(length(base.x),sd=0.2)

# estimation du modèle
model=lm(z~M)

# coefficients obtenus
```

Cette approche est cependant moins souple.

2.2 Exercices

Exercice 2.1

Étudier les données `trees` comme suggéré par l'aide en ligne correspondant à celles-ci.

Exercice 2.2

On considère le modèle quadratique $Y \sim 2X^2 + 1 + \mathcal{N}(0, \frac{1}{4})$. Étudier expérimentalement l'estimation fournie par R grâce à la formule `y~1+x+I(x^2)`.