

Modèles Linéaires

Correction Contrôle Licence 3 MIASHS (2022-2023)

Stéphane Chrétien, Guillaume Metzler & Francesco Amato

Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

stephane.chretien@univ-lyon2.fr ; guillaume.metzler@univ-lyon2.fr ;
francesco.amato@univ-lyon2.fr ;

Exercice 1 : Régression Linéaire

On considère le

$$Y_i = \beta_0 + \beta_1 d_i + \beta_2 h_i + \beta_3 a_i + \varepsilon_i \text{ pour } i = 1 \dots 20,$$

où Y_i est une variable aléatoire représentant le nombre de cas d'une certaine pathologie dans la $i^{\text{ème}}$ région française, d_i , h_i et a_i désignent respectivement la densité urbaine moyenne, l'humidité moyenne et l'altitude moyenne de cette $i^{\text{ème}}$ région, les ε_i sont des termes d'erreur aléatoires supposés indépendants et Gaussiens de variance $\text{var}(\varepsilon_i) = \sigma^2$ pour tout $i = 1, \dots, n = 20$.

Ce modèle s'écrit sous la forme matricielle classique :

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

avec $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^\top$ et $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{20})^\top$. On a calculé

$$X^\top y = (40.7378, 527.2645, 111.4137, 22.89)^\top. \quad (1)$$

$$X^\top X = \begin{bmatrix} 20 & 226.8 & 54.1 & 12.7 \\ 226.8 & 3135.8 & 757.06 & 133.98 \\ 54.1 & 757.06 & 190.97 & 32.339 \\ 12.7 & 133.98 & 32.339 & 9.3258 \end{bmatrix}, \quad (2)$$

$$(X^T X)^{-1} = \begin{bmatrix} 0.9711 & -0.0541 & 0.0769 & -0.8115 \\ -0.0541 & 0.011 & -0.0338 & 0.0333 \\ 0.0769 & -0.0338 & 0.1281 & -0.064 \\ -0.8115 & 0.0333 & -0.064 & 0.9557 \end{bmatrix} \quad (3)$$

et

$$\varepsilon = [\varepsilon_1, \varepsilon_2]^T \quad (4)$$

où

$$\varepsilon_1 = [1.4 \quad 1.2 \quad 1.2 \quad -0.8 \quad 2.2 \quad -0.2 \quad -3.1 \quad 6.2 \quad 0.1 \quad 0]$$

et

$$\varepsilon_2 = [1 \quad -2 \quad 0.5 \quad 0.7 \quad -0.9 \quad 1.5 \quad -3 \quad -3 \quad 1.2 \quad 0.4]$$

1. Après avoir rappelé la définition de l'estimateur des moindres carrés $\hat{\beta}$ de β , calculer sa valeur.

On rappelle que l'estimateur du maximum de vraisemblance est obtenu en résolvant le problème d'optimisation suivant

$$\arg \min_{\beta \in \mathbb{R}} \|Y - \mathbf{X}\beta\|_2^2,$$

dont la solution est

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T Y.$$

Dans notre contexte, notre estimateur est donné par

$$\hat{\beta} = \begin{bmatrix} 0.9711 & -0.0541 & 0.0769 & -0.8115 \\ -0.0541 & 0.011 & -0.0338 & 0.0333 \\ 0.0769 & -0.0338 & 0.1281 & -0.064 \\ -0.8115 & 0.0333 & -0.064 & 0.9557 \end{bmatrix} \begin{bmatrix} 40.7378 \\ 527.2645 \\ 111.4137 \\ 22.89 \end{bmatrix} = \begin{bmatrix} 1.028 \\ 0.592 \\ -1.882 \\ -0.755 \end{bmatrix}$$

2. Quelles sont les propriétés statistiques connues de β (biais, matrice de variance-covariance $\text{Var}(\hat{\beta})$, optimalité...)?

L'estimateur ainsi obtenu est sans biais et il est même de variance minimale (mais c'est un point que nous n'avons pas vu en cours). En revanche nous avons vu que la variance de cette estimateur est donnée par

$$\text{Var}[\hat{\beta}] = \sigma^2(\mathbf{X}^\top \mathbf{X})^{-1},$$

où σ^2 est la variance du modèle, *i.e.* la variance de la distribution des erreurs du modèle.

3. Déterminer un estimateur $\hat{\sigma}^2$ sans biais de la variance σ^2 , donner sa valeur et en déduire une estimation sans biais de la matrice de variance-covariance $\text{Var}(\beta)$.

Un estimateur sans biais de la variance des erreurs σ^2 du modèle est donné par

$$\hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^n (y_i - \hat{y}_i)^2 = \frac{1}{n-p} \sum_{i=1}^n \varepsilon_i^2,$$

où p désigne le nombre de paramètres dans le modèle de régression, qui est donc égal au nombre de variables plus un (donc ici $p = 4$ dans notre exemple).

On en déduit un estimateur sans biais de la matrice de variance covariance de l'estimateur $\hat{\beta}$:

$$\hat{\sigma}^2(\mathbf{X}^\top \mathbf{X})^{-1}.$$

On détermine cette valeur numérique à l'aide des informations contenus dans le vecteur ε qui est un vecteur de taille 20.

```
# Estimation de la variance des erreurs du modèle
res1 = c(1.4, 1.2, 1.2, -0.8, 2.2, -0.2, -3.1, 6.2, 0.1, 0)
res2 = c(1, -2, 0.5, 0.7, -0.9, 1.5, -3, -3, 1.2, 0.4)
res = c(res1, res2)
sigma2_hat = sum(res^2)/(length(res)-4)
sigma2_hat
## [1] 5.42625
```

4. Faire le test de nullité pour chaque coefficient β_0 , β_1 , β_2 et β_3 . On fera un test au risque d'erreur $\alpha = 5\%$.

On rappelle que les différents paramètres $\hat{\beta}_j$ sont distribués selon une loi normale de moyenne β_j et de variance $h_{j+1,j+1}$ où $h_{j+1,j+1}$ désigne l'élément en position $(j+1, j+1)$ dans la matrice $\sigma^2(\mathbf{X}^T \mathbf{X})^{-1}$.

Ainsi la statistique de test employé pour tester la significativité du paramètres β_j est donné par

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{h_{j+1,j+1}}}$$

Cette statistique de test, sous l'hypothèse H_0 selon laquelle $\beta_j = 0$, devient

$$\frac{\hat{\beta}_j}{\sqrt{h_{j+1,j+1}}}$$

qui suit une loi de Student à $n - p$ degrés de libertés et dont on utilisera l'estimation de σ^2 pour calculer la valeur de cette statistique de test.

Ainsi les valeurs sont données par :

```
beta = c(1.028, 0.592, -1.882, -0.755)
XtX = rbind(c(20, 226.8, 54.1, 12.7),
            c(226.8, 3135.8, 757.06, 133.98),
            c(54.1, 757.06, 190.97, 32.339),
            c(12.7, 133.98, 32.339, 9.3258))
H = sigma2_hat * solve(XtX)
h = diag(H)

# Les valeurs des statistiques de test sont les suivantes
stats_tests = beta/sqrt(h)
stats_tests

## [1] 0.4478169 2.4275882 -2.2574232 -0.3315400
```

On compare ensuite la valeur absolue de ces statistiques de tests au quantile d'ordre $1 - \alpha/2$ d'une loi de student à $20 - 4$ soit 16 degrés de liberté. Nous prendrons ici $\alpha = 5\%$

```
# Seuil critique
t_crit = qt(0.975, length(res)-4)
t_crit

## [1] 2.119905

# Résultats du test
abs(stats_tests) > t_crit

## [1] FALSE TRUE TRUE FALSE
```

Ainsi, seuls les paramètres β_1 et β_2 sont significatifs dans ce modèle.

5. Calculez le BIC pour le modèle où toutes les variables sont considérées et décrire quels calculs vous pourriez faire pour comparer tous les modèles possibles obtenus en incluant ou pas chacune des variables.

On rappelle que le BIC est défini par

$$BIC = n(\ln(2\pi) + 1) + n \ln \left(\frac{SCR}{n} \right) + (k + 1) \ln(n),$$

où n désigne le nombre d'individus, k le nombre de paramètres à apprendre (nombre de paramètre de la régression + variance des erreurs) et SCR désigne la somme des carrés des résidus. Ainsi, pour le modèle complet, nous avons

```
n = length(res)
k = 5

BIC = n*log(2*pi)+n + n*log(sum(res^2)/n) + (k+1)*log(n)
BIC
## [1] 104.094
```

Pour évaluer le BIC des autres modèles, disons partiels, il faudrait appliquer la procédure suivante

- sélectionner les variables pour l'apprentissage (les différents sous groupes possibles de variables)
- apprendre un modèle pour chaque sous groupe
- évaluer le BIC

Exercice 2 : Détection d'Outliers

On s'intéresse dans cet exercice à la construction d'une fonction d'influence pour la détection d'outliers.

On s'intéresse plus particulièrement au modèle linéaire :

$$y_i = x_i^\top \beta + \varepsilon_i \quad (5)$$

$i = 1, \dots, n$, où x_i est un vecteur colonne de \mathbb{R}^p . On voudrait pouvoir détecter ces données aberrantes dans un jeu de données.

Pour cela, notons

$$h_{i,i} = x_i^\top (X^\top X)^{-1} x_i.$$

Cette quantité est appelée "leverage score" et nous permet d'introduire la "distance de Cook" comme le nombre

$$D_i = \frac{(Y_i - x_i^\top \hat{\beta})^2}{(p+1)s^2} \frac{h_{ii}}{(1-h_{ii})^2}$$

pour $i = 1, \dots, n$, où

$$s^2 = \frac{1}{n-p-1} \sum_{i=1}^n (Y_i - x_i^\top \hat{\beta})^2 \quad (6)$$

est l'estimation non-biasée de la variance σ^2 du bruit. Cette distance nous indique la sensibilité de la régression vis à vis de la $i^{\text{ième}}$ donnée (x_i, y_i) et focalise habituellement sur la détection de la présence d'outliers.

Passons maintenant à un exemple. Un père a deux garçons, et s'inquiète de la croissance de son cadet qu'il trouve petit. Il décide de faire un modèle familial à partir des mesures de taille en fonction de l'âge de l'aîné :

age	3	4	5	7	8	9	10	11	12
taille	96	104.8	110.3	121.9	127.4	130.8	136	139.7	144.5

1. Calculer les paramètres du modèle de régression pour ces données.

Le modèle étudié ici est un modèle de régression linéaire simple, on peut donc faire le choix de déterminer les paramètres du modèle en employant la même relation qu'à l'exercice précédent, ou encore en calculant

$$\hat{\beta}_1 = \frac{Cov[x,y]}{Var[x]} \quad \text{et} \quad \hat{\beta}_0 = \mathbb{E}[Y] - \hat{\beta}_1 \mathbb{E}[X].$$

Avec nos données, nous avons

```
x = c(3,4,5,7,8,9,10,11,12)
y = c(96,104.8,110.3,121.9,127.4,130.8,136,139.7,144.5)

# Espérance de X
mx = mean(x)

# Variance de X
vx = var(x)


# Espérance de Y
my = mean(y)

# Covariance
cxy = cov(x,y)

# Estimation de beta_1 et beta_0
beta_1 = cxy/vx
beta_0 = my - beta_1*mx

beta_1
## [1] 5.229583

beta_0
## [1] 83.39542
```

On va tout de même faire la régression avec  pour vérifier nos calculs et avoir une idée de nos éventuelles outliers avec notre droite de régression.

```
data <- data.frame(x = x, y = y)

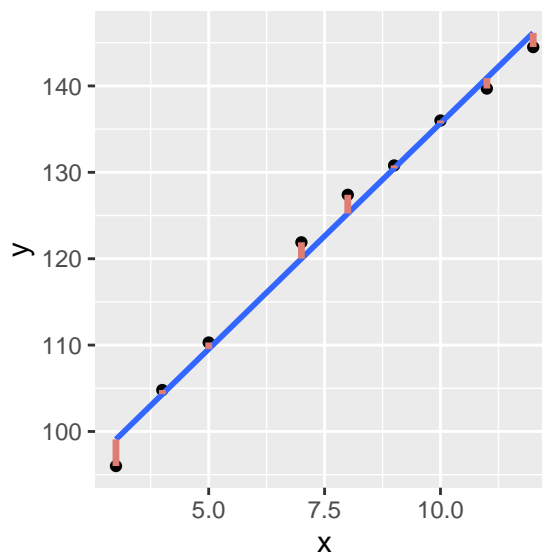
# Estimation des paramètres du modèle
mymodel <- lm(y~x,data)
coeff <- mymodel$coefficients
```

```

coeff
## (Intercept)          x
## 83.395417    5.229583

# Représentation graphique de la droite de régression et résidus
library(ggplot2)
ggplot(data, aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  geom_segment(aes(x = x, y = y, xend = x,
                  yend = coeff[1] + coeff[2]*x, col = "Residuals"),
              col = "#DF7D72", lwd= 1.2, data = data)

```



2. Calculer l'estimation de la variance à partir de l'estimateur non-biaisé.

On emploie à nouveau la même formule qu'à l'exercice précédent. On prendra soin, dans un premier temps, de calculer les résidus en calculant les valeurs prédites par le modèle.

```

# Prédiction du modèle
y_hat = beta_0 + beta_1*x

# Estimation des résidus
res = y-y_hat

# Estimation de la variance du modèle
s2 = sum(res^2)/(length(x)-2)

```



```

s2
## [1] 3.29222

res
## [1] -3.0841667  0.4862500  0.7566667  1.8975000  2.1679167  0.3383333  0.3087500
## [8] -1.2208333 -1.6504167


```

On note que les individus dont l'âge est égal à 3 ou 8 (et pas l'individu 9) sont les individus pour lesquels les résidus sont les plus élevés. Ils donc potentiellement représenter des outliers dans notre jeu de données.

3. Calculer l'indice de Cook pour la donnée numéro 3 et la donnée numéro 9. Attention, ici il faut comprendre les données des individus dont l'âge est égal à 3 et 9 respectivement.

Pour cela on emploie la définition de la distance de Cook décrite un peu plus haut :

$$D_i = \frac{(Y_i - x_i^\top \hat{\beta})^2}{(p+1)s^2} \frac{h_{ii}}{(1-h_{ii})^2} = \frac{\varepsilon_i^2}{(p+1)s^2} \frac{h_{ii}}{(1-h_{ii})^2}$$

Regardons cela de suite sur  pour les individus 3 et 9. Ici, il s'agit bien de supprimer les individus dont les valeurs de l'âge sont bien 3 et 9 respectivement.

```

# Matrice de design
X = cbind(c(rep(1,length(x))),x)

# Pour l'individu 3
x3 = matrix(c(1,x[1]), ncol = 1)
h3 = t(x3)%*%solve(t(X)%*%X)%*%x3
h3

##           [,1]
## [1,] 0.3833333

D3 = (res[1]^2/(3*s2))*(h3/((1-h3)^2))
D3

##           [,1]
## [1,] 0.9708255

# Pour l'individu 9
x9 = matrix(c(1,x[6]), ncol = 1)
h9 = t(x9)%*%solve(t(X)%*%X)%*%x9
h9

```

```

##           [,1]
## [1,] 0.1333333

D9 = (res[6]^2/(3*s2))*(h9/((1-h9)^2))
D9

##           [,1]
## [1,] 0.002057378

# Pour l'individu 8
x8 = matrix(c(1,x[5]), ncol = 1)
h8 = t(x8)%*%solve(t(X)%*%X)%*%x8
h8

##           [,1]
## [1,] 0.1125

D8 = (res[5]^2/(3*s2))*(h8/((1-h8)^2))
D8

##           [,1]
## [1,] 0.06796586

```

On remarque que la distance de Cook de l'individu dont l'âge est égal à 9 est plutôt faible contrairement à l'individu âgé de 3 ans et même constat pour la distance de cook de l'individu âgé de 8 ans. C'est une observation que l'on avait déjà pu faire graphiquement.

4. Recalculer le modèle de régression sans ces données.

On va maintenant supprimer les données mentionnées de notre jeu de données et refaire l'estimation du modèle.

L'apprentissage du modèle se fait exactement de la même façon que précédemment, il y a simplement moins d'individus

```

# On enlève les individus en question de notre jeu de données
newdata = data[-c(1,6),]
newx = newdata$x
newy = newdata$y

# Apprentissage du modèle

# Espérance de X
mx = mean(newx)

# Variance de X
vx = var(newx)

```

```

# Espérance de Y
my = mean(newy)

# Covariance
cxy = cov(newx,newy)

# Estimation de beta_1 et beta_0
newbeta_1 = cxy/vx
newbeta_0 = my - newbeta_1*mx

newbeta_1
## [1] 4.935156

newbeta_0
## [1] 86.18516

```

On peut à nouveau vérifier nos résultats avec [Rafin](#) de voir que les calculs effectués sont corrects et obtenir la nouvelle droite de régression.

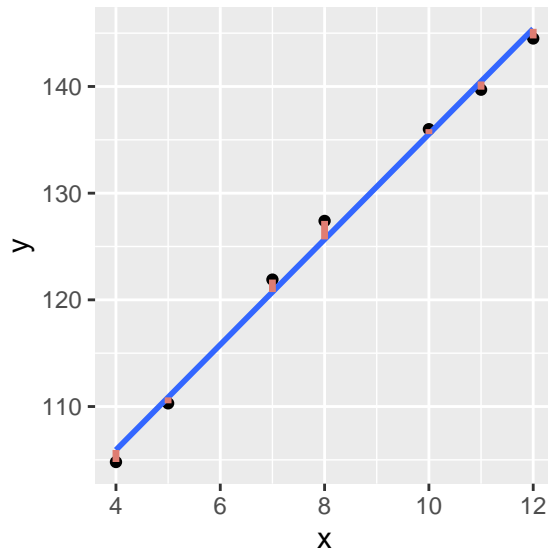
```

# Estimation des paramètres du modèle
newmymodel <- lm(y~x,newdata)
newcoeff <- newmymodel$coefficients
newcoeff

## (Intercept)          x
## 86.185156      4.935156

# Représentation graphique de la droite de régression et résidus
library(ggplot2)
ggplot(newdata, aes(x=x, y=y)) +
  geom_point() +
  geom_smooth(method=lm,se=FALSE) +
  geom_segment(aes(x = newx, y = newy, xend = newx,
                  yend = newcoeff[1] + newcoeff[2]*newx, col = "Residuals"),
              col = "#DF7D72", lwd= 1.2, data = newdata)

```



5. Comparer l'erreur de prédiction pour une nouvelle donnée (6, 115.3) pour
 — le modèle avec les données 3 et 9

Pour ce premier calcul, on utilisera les paramètres du modèle estimés à la question 1.

On commence par calculer la valeur prédite par le modèle, puis nous pourrions calculer l'erreur de prédiction.

```
x_new = 6
y_new = 115.3

# Valeur prédite par le modèle
y_hat = beta_0 + x_new*beta_1

# Calcul de l'erreur de prédiction
res = y_new - y_hat
res
## [1] 0.5270833
```

- le modèle sans les données 3 et 9

Ici nous utiliserons les paramètres du modèle estimés à la question 4.

On commence par calculer la valeur prédite par le modèle, puis nous pourrions calculer l'erreur de prédiction.

```
x_new = 6
y_new = 115.3
```

```

# Valeur prédite par le modèle
y_hat = newbeta_0 + x_new*newbeta_1
y_hat
## [1] 115.7961
# Calcul de l'erreur de prédiction
res = y_new - y_hat
res
## [1] -0.4960938

```

On remarque ainsi que l'erreur du modèle (en valeur absolue!) est plus importante en utilisant les paramètres estimés avec les individus 3 et 9.

On se propose d'aller un petit peu plus loin en regardant ce qu'il se passe si on cherche à évaluer l'erreur de prédiction pour l'individu âgé de 6 ans mais en utilisant un modèle dont les paramètres sont estimées en excluant les données aberrantes (données pour lesquelles $x = 3$ et $x = 8$)

```

newdata = data[-c(1,5),]
newx = newdata$x
newy = newdata$y

# Apprentissage du modèle

# Espérance de X
mx = mean(newx)

# Variance de X
vx = var(newx)

# Espérance de Y
my = mean(newy)

# Covariance
cxy = cov(newx,newy)

# Estimation de beta_1 et beta_0
newbeta_1 = cxy/vx
newbeta_0 = my - newbeta_1*mx

newbeta_1
## [1] 4.946649
newbeta_0

```

```

## [1] 85.87062

# Estimation de l'erreur sur notre nouvel individu

x_new = 6
y_new = 115.3

# Valeur prédite par le modèle
y_hat = newbeta_0 + x_new*newbeta_1
y_hat

## [1] 115.5505

# Calcul de l'erreur de prédiction
res = y_new - y_hat
res

## [1] -0.2505155

```

Dans ce cas, on constate, qu'après avoir supprimé les deux outliers de notre jeu de données, l'erreur d'estimation est encore plus faible que les deux erreurs précédentes.

Ainsi, en supprimant les outliers de notre jeu de données, nous sommes en mesure d'améliorer les capacités prédictives de notre modèle.