CentraleSupélec

Statistiques avancées

Régression linéaire

Développement de la chenille processionnaire du pin

Auteurs :
M. Ahmed El Aamrani

Encadrants :
Mme. Sarah Lemler

Version du
23 février 2018

Table des matières

Table des figures

Introduction

Dans un contexte de bouleversement climatique, les incertitudes conçernant la faune et la flore se multiplient. Le rapport de l'Académie des Sciences présenté ce lundi 25 septembre sur les "mécanismes d'adaptation de la biodiversité aux changements climatiques" est le résultat de près de deux ans de travail qui a rassemblé de nombreux experts. Le principal constat est la migration des espèces vers le nord et l'altitude, puisqu'en effet "une augmentation de température de $1°C$ correspond en France à un décalage des zones climatiques d'environ 200 km vers le nord" [Sandra Lavorel, Académie des Sciences].

De ce fait, les espèces de milieux chauds ou tempérés migrent dans des milieux plus froids, de plus hautes latitudes ou altitudes. L'exemple le plus connu de cette migration est celui de la chenille processionnaire du pin, qui « est en train de remonter vers le nord de la France à la vitesse de 5,6 km/an », profitant d'hivers doux dans des régions où les larves ne pouvaient pas survivre auparavant.

FIGURE 1 – La chenille processionnaire du pin [www.sofraeve.fr]

Mais d'autres espèces n'arrivent pas à « migrer » suffisamment rapidement. Le cas le plus extrême est celui des arbres, générant ainsi une mauvaise adaptation de ces derniers aux populations migrantes de chenille. En se nourrissant des aiguilles des résineux, les chenilles réduisent notablement la productivité des forêts et contribuent à leur fragilisation.

C'est donc la raison pour laquelle on souhaite étudier le développement de la chenille processionnaire du pin, et plus particulièrement l'influence de caractéristiques forestières sur ce développement. Pour arriver à cette fin, nous avons observé une parcelle forestière connexe de 10 hectares. Ce terrain est considéré comme homogène par rapport aux variables étudiées. Ces variables ont été obtenues par la moyenne des valeurs mesurées.

Le jeu de données comporte 25 observations sur la chenille processionnaire du pin. La variable à expliquer est la variable x11 (log décimal du nombre de nids de processionnaires par arbre d'une placette). Elle dépend de :

— x1 ≡ altitude (en mètre)
— x2 ≡ pente (en degré)
— x3 ≡ nombre de pins dans une placette de 5 ares
— x4 ≡ hauteur de l'arbre échantillonné au centre de la placette (en mètre)
— x5 ≡ diamètre de cet arbre (en mètre)
— x6 ≡ note de densité du peuplement
— x7 ≡ orientation de la placette (1≡sud, 2≡nord)
— x8 ≡ hauteur des arbres dominants (en mètre)
— x9 ≡ nombre de strates de végetation
— x10 ≡ mélange du peuplement (1≡pas mélangé, 2≡mélangé)

Le but de cette étude est donc d'arriver à modéliser le problème à l'aide d'une régression linéaire, et d'exprimer ainsi la variable x11 en fonction des autres variables du problème.

Chapitre 1

Démarche brute par régression linéaire

1.1 Lecture des données

On commence dans un premier temps par lire le fichier de données 'pine.sup.data' auquel on enlève au préalable tous les commentaires, et on laisse uniquement les données et l'entête décrivant chaque colonne. Ensuite, on appelle la fonction :

$$read.table("pine.sup.data",\ header = TRUE)$$

Ainsi, on peut commencer par afficher la dimension de la table, ainsi que les 5 premières observations de notre jeu de données.

```
> df=read.table("pine.sup.data", header=TRUE)
> dim(df)
[1] 25 11
> head(df, 5)
    x1 x2 x3  x4   x5  x6  x7  x8  x9 x10  x11
1 1107 31 23 6.0 22.2 2.6 1.0 9.0 3.0 1.4 1.17
2 1116 34  6 2.5  6.6 1.3 1.8 3.9 1.2 1.5 0.67
3 1174 32 22 3.9 11.9 2.3 1.7 6.1 1.8 1.5 0.90
4 1131 30  6 4.7 15.3 1.5 1.5 6.5 1.4 1.3 2.32
5 1150 34 12 3.1  9.4 1.7 1.8 4.8 1.6 1.3 3.89
```

FIGURE 1.1 – Affichage des données

Cette table est en effet constituée de 25 observations, et de 11 variables comme nous avons pu le mentionner auparavant. La table a donc été correctement chargée.

La fonction *str(df)* affiche, quant à elle, le type de données dans chaque variable ainsi que l'ensemble des valeurs prises par ces variables.

```
> str(df)
'data.frame':   25 obs. of  11 variables:
 $ x1 : int  1107 1116 1174 1131 1150 1132 1258 1114 1177 1146 ...
 $ x2 : int  31 34 32 30 34 22 14 26 36 26 ...
 $ x3 : int  23 6 22 6 12 18 0 9 3 18 ...
 $ x4 : num  6 2.5 3.9 4.7 3.1 7 3.8 3.4 4 4.3 ...
 $ x5 : num  22.2 6.6 11.9 15.3 9.4 37 8.5 9.6 14.2 17.9 ...
 $ x6 : num  2.6 1.3 2.3 1.5 1.7 2.5 1 1.6 1.2 2.3 ...
 $ x7 : num  1 1.8 1.7 1.5 1.8 1.5 1.2 1.6 1.3 1.6 ...
 $ x8 : num  9 3.9 6.1 6.5 4.8 9 5.6 5.1 5.9 7.7 ...
 $ x9 : num  3 1.2 1.8 1.4 1.6 2 1 1.5 1.3 2 ...
 $ x10: num  1.4 1.5 1.5 1.3 1.3 1.5 1 1.3 1.6 1.4 ...
 $ x11: num  1.17 0.67 0.9 2.32 3.89 6 3.18 0.9 2.5 2 ...
```

FIGURE 1.2 – Ensembles des valeurs prises par les variables

Les variables $x1, x2, x3$ prennent leurs valeurs dans l'ensemble des entiers. Le reste des variables sont des variables réelles. Remarquons qu'à travers ce tableau on peut voir que $x7$ et $x10$ prennent leurs valeurs dans $\{1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$ qui est un ensemble discret. En effet, elles sont qualitatives car elles ne représentent pas une quantité. L'une correspond à un degré d'orientation et l'autre à un degré de mélange. Les deux variables sont comprises entre 1 et 2. Cependant, dans un premier temps, on suppose qu'elles sont continues.

Egalement, en cherchant à opérer *summary(df)*, on a :

```
> summary(df)
       x1              x2              x3              x4              x5              x6       
 Min.   :1092   Min.   :14.00   Min.   : 0.00   Min.   :2.50   Min.   : 6.6   Min.   :1.000  
 1st Qu.:1143   1st Qu.:22.00   1st Qu.:10.00   1st Qu.:3.90   1st Qu.:11.7   1st Qu.:1.600  
 Median :1165   Median :30.00   Median :14.00   Median :4.50   Median :14.0   Median :1.900  
 Mean   :1173   Mean   :29.88   Mean   :15.52   Mean   :4.66   Mean   :15.2   Mean   :2.088  
 3rd Qu.:1186   3rd Qu.:36.00   3rd Qu.:22.00   3rd Qu.:5.60   3rd Qu.:16.4   3rd Qu.:2.500  
 Max.   :1326   Max.   :45.00   Max.   :35.00   Max.   :7.20   Max.   :37.0   Max.   :3.400  
       x7              x8              x9             x10             x11       
 Min.   :1.000   Min.   : 3.900   Min.   :1.000   Min.   :1.000   Min.   :0.010  
 1st Qu.:1.400   1st Qu.: 6.100   1st Qu.:1.600   1st Qu.:1.400   1st Qu.:0.670  
 Median :1.600   Median : 7.700   Median :2.100   Median :1.600   Median :1.000  
 Mean   :1.584   Mean   : 7.616   Mean   :2.048   Mean   :1.556   Mean   :1.439  
 3rd Qu.:1.800   3rd Qu.: 9.000   3rd Qu.:2.400   3rd Qu.:1.800   3rd Qu.:2.000  
 Max.   :2.000   Max.   :11.200   Max.   :3.000   Max.   :2.000   Max.   :6.000  
```

FIGURE 1.3 – Etude des caractéristiques de chaque variable

Dans cette table, on a le minimum et le maximum des valeurs prises par chaque variable. On a également, le 1er quartile qui est la valeur en dessous de laquelle on a 1/4 des observations, la médiane qui est la valeur en dessous de laquelle on a la moitié des observations, et le 3ème quartile qui est la valeur en dessous de laquelle on a 3/4 des observations. Les quartiles sont de ce fait les trois quantiles qui divisent l'ensemble de données en quatre groupes de tailles égales. On peut également vérifier que $x7$ et $x10$ prennent des valeurs comprises entre 1 et 2.

Ensuite, on peut calculer la moyenne et l'écart-type de chaque variable, en appelant :

apply(df,2,function(x){c(mu=mean(x),sd=sd(x))})

Cette fonction cherche à calculer ces valeurs pour chaque colonne, d'où l'argument "2" de la fonction ci-dessus. En affichant les résultats on a :

```
> apply(df,2,function(x){c(mu=mean(x),sd=sd(x))})
          x1        x2        x3       x4        x5        x6        x7       x8        x9       x10       x11
mu 1172.96000 29.880000 15.520000 4.66000 15.204000 2.0880000 1.5840000 7.616000 2.0480000 1.5560000 1.439200
sd   53.54475  8.486853  9.323447 1.20727  7.106598 0.6845436 0.2703085 1.849703 0.5694442 0.2631223 1.393799
```

FIGURE 1.4 – Moyenne et écart-type de chaque variable

On remarque d'ailleurs qu l'on obtienne les mêmes valeurs de moyenne comparées à ceux affichées avec la méthode *summary(df)*. De plus, ce sont les variables qui prennent des valeurs larges, qui ont un grand écart-type, comme $x1, x2, x3$ et $x5$.

1.2 Etude des corrélations entre variables

Pour étudier les corrélations entre chaque variable, on peut dans un premier temps afficher chaque couple de variables dans un graphe en 2 dimensions.

FIGURE 1.5 – Scatter-plot des corrélations

Juste en visualisant ce graphique, on peut s'apercevoir que $(x3, x6)$ et $(x4, x5)$ sont fortement corrélées par exemple. Le coefficient de corrélation ρ est donnée par :

$$\forall i \in \{1, \ldots, 11\} \forall j \in \{1, \ldots, 11\} \quad \rho(x_i, x_j) = \frac{Cov(x_i, x_j)}{\sqrt{Var(x_i)Var(x_j)}}$$

On remarque d'ailleurs que $\rho(x_i, x_j) = \rho(x_j, x_i)$ et $\rho(x_i, x_i) = 1$. Et on peut calculer ces coefficients par la fonction *cor(df)* :

```
> round(cor(df),3)
       x1     x2     x3     x4     x5     x6     x7     x8     x9    x10    x11
x1  1.000 -0.171 -0.070 -0.121 -0.170 -0.056 -0.183  0.020  0.041  0.174 -0.237
x2 -0.171  1.000  0.368  0.224 -0.029  0.363  0.192  0.148  0.225  0.516 -0.230
x3 -0.070  0.368  1.000  0.180 -0.052  0.984  0.065  0.589  0.807  0.606 -0.083
x4 -0.121  0.224  0.180  1.000  0.827  0.230 -0.001  0.533  0.291  0.237  0.138
x5 -0.170 -0.029 -0.052  0.827  1.000 -0.003  0.056  0.360  0.088 -0.036  0.364
x6 -0.056  0.363  0.984  0.230 -0.003  1.000  0.024  0.612  0.814  0.578 -0.047
x7 -0.183  0.192  0.065 -0.001  0.056  0.024  1.000  0.175  0.043  0.224 -0.095
x8  0.020  0.148  0.589  0.533  0.360  0.612  0.175  1.000  0.832  0.620 -0.158
x9  0.041  0.225  0.807  0.291  0.088  0.814  0.043  0.832  1.000  0.652 -0.314
x10 0.174  0.516  0.606  0.237 -0.036  0.578  0.224  0.620  0.652  1.000 -0.347
x11 -0.237 -0.230 -0.083  0.138  0.364 -0.047 -0.095 -0.158 -0.314 -0.347  1.000
```

FIGURE 1.6 – Coefficients de corrélations entre chaque couple de variables

Cette matrice est en effet symétrique, puisque comme on vient de le voir le coefficient de corrélation est lui-même symétrique, et de diagonale égale à 1. On peut également visualiser graphiquement ces coefficients par la fonction *corrplot(cor(df))* :

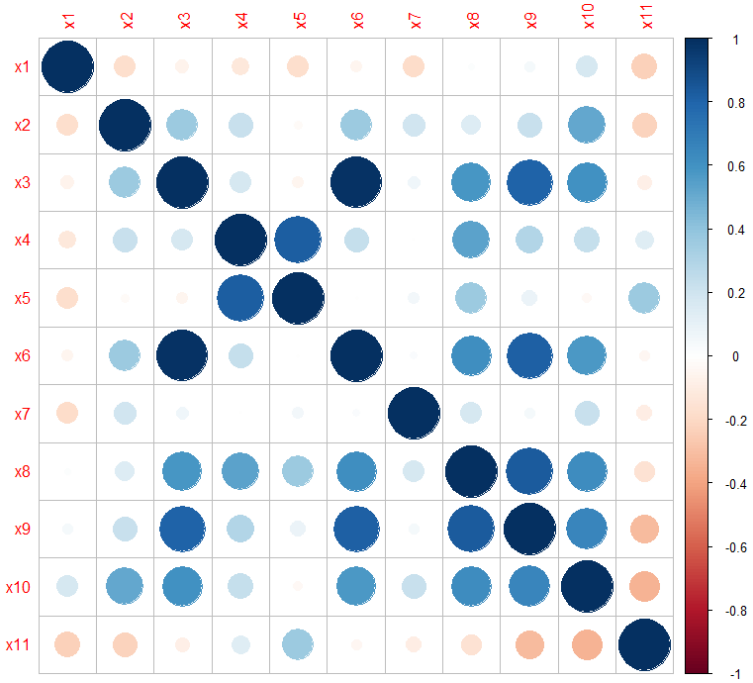

FIGURE 1.7 – Visualisation des coefficients de corrélation

On peut voir ici que les coefficients de corrélations de $x3/x6$ et $x4/x5$ sont relativement élevés et proches de 1, comme on l'a déja pressenti en affichant le scatter-plot intial des corrélations (Figure 1.5). De plus, $x8, x9, x10$ sont fortement corrélées les uns les autres.

1.3 Régression linéaire avec toutes les variables

On commence dans un premier temps par extraire des données, les valeurs de sortie Y qui correspondent à la colonne $x11$. On extrait également le plan d'expérience X en prenant toutes les colonnes de *df* à part la coloone $x11$, puis en rajoutant un *intercept* à gauche de la matrice. On renomme la colonne de l'intercept $x0$, pour garder une cohérence avec la notation des autres colonnes.

On note :
— n le nombre de lignes ou d'observations dans le plan d'expérience
— p le nombre de colonnes ou de variables dans le plan d'expérience
Le modèle linéaire s'écrit :

$$Y = X\theta + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I_n) \tag{1.1}$$

1.3.1 Implémentation manuelle

On peut commencer par estimer $\widehat{\theta} = (X^T X)^{-1} X^T Y$ en calculant :

```
> #Calcul de l'estimateur de theta
> theta.est=solve(t(X)%*%X)%*% t(X)%*%Y
> round(theta.est,3)
      [,1]
x0   8.472
x1  -0.004
x2  -0.037
x3  -0.030
x4  -0.529
x5   0.138
x6   2.218
x7  -0.745
x8   0.236
x9  -2.905
x10  0.172
```

FIGURE 1.8 – Calcul de l'estimateur de theta

Ainsi, nous avons ici un vecteur colonne de taille 11, ce qui est bien cohérent. Et dans chaue ligne, on a la valeur de l'estimateur correspondant à une variable donnée. Typiquement, si l'estimateur lié à x1 est nul (ce qui n'est pas le cas ici), cela veut dire simplement que le nombre de nids ne dépend pas de l'altitude de l'arbre.

On peut également calculer l'estimateur non-biaisé de σ qui est donné par

$$\widehat{\sigma} = \sqrt{\frac{1}{n-p}||Y - X\widehat{\theta}||^2}$$

```
> #Calcul de l'estimateur non-biaisé de sigma^2
> sigma.est=sqrt(sum( (X%*%theta.est -Y)^2   )/(n-p))
> round(sigma.est,3)
[1] 1.193
```

FIGURE 1.9 – Calcul de l'estimateur de l'écart-type du bruit

De plus, on sait que $\widehat{\theta} \sim \mathcal{N}(\theta, \sigma^2(X^TX)^{-1})$. Donc, on peut calculer la matrice de covariance de la loi de theta, et ainsi l'écart-type de chaque composante de theta qui est donnée par la racine de l'élement de la diagonale correspondant.

```
#Calcul de la matrice de covariance de la loi de theta
V=solve(t(X)%*%X) *sigma.est^2
#Calcul de l'écart-type de chaque composante de theta
stddev=sqrt(diag(V))
```

FIGURE 1.10 – Calcul de la matrice de covariance de la loi de theta

Le test de Student est dédié au test d'une relation affine des composantes du paramètre $L\theta = c$. En prenant L une matrice ligne de dimension p, on a :

$$T = \frac{L\widehat{\theta} - L\theta}{\widehat{\sigma}\sqrt{L(X^TX)^{-1}L^T}} \sim_{H_0} \mathcal{T}(n-p)$$

Ensuite, on pourrait calculer cette statistique pour une variable $j \in \{1, ..., 10\}$ avec :

$$(H_0) : \text{consite à supposer } \theta_j = 0$$
$$(H_1) : \text{consite à supposer } \theta_j \neq 0$$

Dans ce cas, on prend $L = (0 \dots 010 \dots 0)$ qui vaut 0 partout sauf en position j, la statistique de test pour une variable j devient :

$$T_j = \frac{\theta_j}{\sqrt{\widehat{\sigma}^2(X^TX)^{-1}_{jj}}}$$

Ce qu'on pourrait calculer sur R avec :

```
#Calcul de la statistique de test de Student pour theta_j=0 contre theta_j#0
tobs=theta.est/stddev
```

FIGURE 1.11 – Calcul de la statistique du test de Student

On peut également calculer les valeurs de sortie estimées comme suit :

$$\widehat{Y} = X\widehat{\theta}$$

1.3.2 Implémentation de la fonction 'lm' et comparaison

On commence par appeler la fonction *lm(x11∼., data=df)* qui consiste simplement à essayer d'opérer une régression linéaire dont la valeur de sortie est $x11$ et les variables sont l'ensembles des variables restantes. En affichant le résultat, on a :

```
> res=lm(x11~.,data=df)
> summary(res)

Call:
lm(formula = x11 ~ ., data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.08268 -0.76751  0.04839  0.47910  2.80973

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.471750   6.590257   1.285   0.2195
x1          -0.003987   0.005278  -0.755   0.4625
x2          -0.036608   0.041214  -0.888   0.3894
x3          -0.030087   0.166655  -0.181   0.8593
x4          -0.528646   0.470190  -1.124   0.2798
x5           0.138107   0.072770   1.898   0.0785 .
x6           2.218364   2.244948   0.988   0.3398
x7          -0.744518   1.056633  -0.705   0.4926
x8           0.235926   0.335667   0.703   0.4937
x9          -2.904804   1.182389  -2.457   0.0277 *
x10          0.172329   1.727251   0.100   0.9219
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.193 on 14 degrees of freedom
Multiple R-squared:  0.5727,    Adjusted R-squared:  0.2675
F-statistic: 1.877 on 10 and 14 DF,  p-value: 0.1367
```

FIGURE 1.12 – Résultat de la fonction lm sur le modèle brut

Dans la première partie des résultats, on a des informations (min, 1er quartile, médiane et 3ème quartile) des résidus qui sont données par :

$$\widehat{\epsilon} = Y - \widehat{Y} = Y - X\widehat{\theta}$$

D'ailleurs, en appelant $'res\$residuals'$ et $'Y - X\%*\%theta.est'$, on obtient les mêmes résultats (voir script associé au rapport).

La 2ème partie des résultats concerne l'estimation de θ, en effet dans la première colonne *'Estimate'*, on a les mêmes valeurs que $\theta.est$. Ceci est donc bien cohérent.

De plus, dans la 2ème colonne *'Std. Error'*, on a les mêmes valeurs que le vecteur *stddev* donnée par la (Fig. 1.10). Ces valeurs représentent l'écart-type de chaque régresseur θ_j pour j allant de 0 à 10.

La statistique du test de Student ou 't value' est donc le rapport de la première colonne avec le deuxième, comme on l'a démontré précédemment. Elle représente la statistique de test consistant à considérer $\theta_j = 0$ contre $\theta_j \neq 0$

Or, cette statistique de test définit une région de rejet $R_\alpha = \{|T| > c_\alpha\}$ et suit la loi de Student à $n - p$ degrés de libertés. La bilatéralité du test nous donne une p-value $pval$:

$$pval_{Student} = 2\Phi^{\mathcal{T}(n-p)}_{-|tobs|}$$

où tobs est la statistique de test observée et Φ est la fonction de répartition de la loi de Student à $n - p$ degrés de liberté prise en $-|tobs|$.

Donc $Pr(>|t|)$ peut se calculer avec $round(2 * pt(-abs(tobs), res\$df), 4)$. Et on obtient exactement les mêmes résultats.

Dans la 3ème et dernière partie de (Fig. 1.12) , on a :
— Residual standard error = $\widehat{\sigma} = 1.193$ (voir Fig. 1.9)
— Nombre de degrés de liberté (ndl) est $n - p = 25 - 11 = 14$ (apparaît dans le dénominateur de $\widehat{\sigma}^2$).

Ensuite, on calcule le coefficent de détermination multiple R^2 et ajustée R^2_a comme suit :

$$R^2 = \frac{||\widehat{Y} - \bar{Y}\mathbf{1}||^2}{||Y - \bar{Y}\mathbf{1}||^2}$$

$$R^2_a = 1 - \frac{n-1}{n-p}\frac{||\widehat{\epsilon}||^2}{||Y - \bar{Y}\mathbf{1}||^2}$$

avec $\bar{Y}$ la valeur moyenne de Y

Test de significativité globale

On peut aussi calculer une valeur *F-statistic*. Celle-ci correspond à la statistique de test de Fisher qui oppose le modèle i.i.d sous H_0 et le modèle complet sous H_1. Elle est donnée dans ce cas là par la formule suivante :

$$F = \frac{1}{(p-1)\widehat{\sigma}^2}(A\widehat{\theta})^T(A(X^TX)^{-1}A^T)^{-1}A\widehat{\theta}$$

avec

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \sim_{H_0} \mathcal{F}(p-1, n-p)$$

A est de taille $(p-1, p)$ et de rang $p-1$.

$\mathcal{F}$ est la loi de Fisher de paramètres $p-1$ et $n-p$.

```
> #Calcul de la statistique de Fisher (F-Statistic)
> A=diag(c(0,1,1,1,1,1,1,1,1,1,1))[2:11,]
> fstat=1/((p-1)*sigma.est^2)*t(A%*%theta.est)%*%solve(A%*%solve(t(X)%*%X)%*%t(A))%*%A%*%theta.est
> fstat
         [,1]
[1,] 1.876583
```

FIGURE 1.13 – Statistique du test de Fisher de significativité globale

Cela est donc bien cohérent avec la valeur affichée dans (Fig. 1.13). Ensuite, on peut calculer la p-value *pval* qui est donnée par :

$$pval_{Fisher} = 1 - \Phi_{fobs}^{\mathcal{F}(p-1,n-p)}$$

où Φ est la fonction de répartition de la loi de Fisher à $p-1$ et $n-p$ degrés de liberté prise en fobs.

On a le résultat suivant :

```
> round(1-pf(fstat,p-1,n-p),4)
       [,1]
[1,] 0.1367
```

FIGURE 1.14 – p-value du test de Fisher de significativité globale

Ceci est donc la même valeur qui nous a été donnée par *summary(res)*. Si on prend un risque d'erreur de première espèce $\alpha = 5\%$ ou $\alpha = 10\%$, on remarque qu'il est inférieure ou égale à la p-value du test de Fisher de significativité globale. Cela signine qu'on garde H_0 qui est le modèle i.i.d ! Que ce soit à travers ce test ou à travers la valeur de R^2 relativement faible, on peut conclure qu'une régression linéaire avec toutes les variables du modèle n'arrive pas à bien décrire le problème. D'où la nécessité d'entamer une phase de sélection de variables !

Chapitre 2

Sélection de variables

On vient donc de voir qu'un modèle linéaire avec toutes les variables ne permet pas d'expliquer le nombre de nids de pocessionnaires en fonction des autres variables. Il est donc nécessaire de pouvoir sélectionner parmi les variables, celles qui sont les plus représentatives. Dans ce rapport, on va évoquer une approche descendante *backward*, une approche ascendante *forward*, et une approche mixte *stepwise* de sélection de variables. Enfin, on verra également une recherche exhaustive qui est possible dans notre cas, en raison de la taille relativement faible des données.

2.1 Sélection descendante par critère AIC

Cette méthode, également appelée *backward selection*, procède par élimination successive de variables : à partir du modèle complet, la variable la moins influente sur le critère choisi (qui est ici AIC ou *Akaike Information Criterion)* est successivement enlevée.

Le critère AIC sélectionne le modèle (ω) qui minimise :

$$AIC(\omega) = -2 \log L(\omega) + 2|\omega|$$

où $L(\omega)$ est la vraisemblance, et $|\omega|$ la dimension du modèle. Dans le cas gaussien, on a :

$$AIC(\omega) = cte + n \log \left(\frac{SCR(\omega)}{n} \right) + 2|\omega|$$

où $SCR(\omega)$ est la somme des carrés des résidus dans le modèle ω.

On implémente cette itération en appelant la fonction "$resf = stepAIC(res)$". En ne donnant que le modèle linéaire res, les autres paramètres sont pris par défaut, en particulier le critère AIC et la recherche *backward*. La première étape déjà consiste à connaître le type d'objet de resf.

```
> #Vérification : resf est de même classe que res
> class(res)
[1] "lm"
> class(resf)
[1] "lm"
```

FIGURE 2.1 – Classe de resf

L'objet resf est donc de classe "lm" qui est le résultat d'un modèle linéaire. Un modèle linéaire est donc choisi à la fin. De même, si on appelle stepAIC, il s'affiche un ensemble d'itérations. Commençons dans un premier temps par la première itération.

```
> resf=stepAIC(res)
Start:  AIC=16.32
x11 ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

       Df Sum of Sq    RSS    AIC
- x10   1    0.0142 19.936 14.341
- x3    1    0.0464 19.968 14.381
- x8    1    0.7030 20.624 15.190
- x7    1    0.7065 20.628 15.194
- x1    1    0.8119 20.733 15.322
- x2    1    1.1227 21.044 15.694
- x6    1    1.3895 21.311 16.008
<none>              19.921 16.323
- x4    1    1.7988 21.720 16.484
- x5    1    5.1253 25.047 20.047
- x9    1    8.5882 28.509 23.284

Step:  AIC=14.34
```

FIGURE 2.2 – Première itération de backward AIC

Le critère AIC du modèle initial est donnée à une constante près par :

```
> n*log(deviance(res)/n)+2*11
[1] 16.32286
```

FIGURE 2.3 – Critère AIC du modèle initial

C'est donc bien la valeur affichée précédemment. Ensuite, dans la première ligne de la (Fig. 2.3), on essaie d'effectuer une régression linéaire sans la colonne $x10$ (ce qui revient au même que de prendre $\theta_{10} = 0$). On peut le faire manuellement en appelant :

$$resf1 = lm(x11 \sim x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9, data = df)$$

La somme des carrés des résidus de ce modèle est donnée par :

$$RSS = deviance(resf1) = sum((Y - X * c(resf1\$coefficients, 0))^2)$$

Le nombre de degrés de liberté est 1, puisqu'on enlève une seule variable $x10$. "Sum of Sq" est donné par la différence entre la somme des carrés de résidus du modèle initial (correspondant à la ligne "none") et celle du modèle de la ligne correspondante.

Le critère AIC du sous-modèle resf1 est donné par :

$$AIC_{resf1} = n \log\left(\frac{deviance(resf1)}{n}\right) + 2 * 10 = 14.341$$

Dans cette première étape, on refait exactement la même procédure en fixant d'autre variables $\theta_i = 0$. Et on remarque, que parmi les autres modèles de dimension 10, le modèle avec $\theta_{10} = 0$ est celui ayant le plus faible critère AIC. C'est donc lui qu'on choisit. Remarquons que dans cette démarche, on a tout de suite négligé la variable 10, qui est une variable qualitative. Cette variable n'est donc pas très bien décrit par un modèle continue. Ensuite, dans les prochaines étapes, on reprend le modèle choisi à la fin de l'étape précédente, et on lui retranche une variable. Remarquons que dans la 3ème itération, on a écarté la variable $x7$ qui est elle aussi une variable qualitative.

Puis on continue les itérations jusqu'à ce qu'on n'arrive plus à améliorer le modèle compte tenu de son critère AIC. En effet, dans la dernière itération, le critère AIC le plus faible correspond à la ligne "none" c'est-à-dire celle qui tient compte de toutes les variables du modèle d'entrée de l'itération. Le modèle retenu est donc un modèle où on ne considère que les variables $x2, x5, x6$ et $x9$. Il est cependant nécessaire de bien se rendre compte que ce modèle comporte aussi un terme d'intercept !

Ensuite, on peut voir que *resf$coef* nous donne accès à $\widehat{\theta}$ du modèle retenu :
— $\widehat{\theta}_0 = 2.91, \widehat{\theta}_2 = -0.05, \widehat{\theta}_5 = 0.09, \widehat{\theta}_6 = 1.80, \widehat{\theta}_9 = -2.45$
— $\widehat{\theta}_1 = \widehat{\theta}_3 = \widehat{\theta}_4 = \widehat{\theta}_7 = \widehat{\theta}_8 = \widehat{\theta}_{10} = 0$
De plus, *resf$anova* nous donne :

```
> resf$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
x11 ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Final Model:
x11 ~ x2 + x5 + x6 + x9


   Step Df   Deviance Resid. Df Resid. Dev       AIC
1                              14   19.92132 16.322864
2 - x10  1 0.01416431        15   19.93548 14.340633
3  - x3  1 0.03534687        16   19.97083 12.384921
4  - x7  1 0.81506275        17   20.78589 11.384966
5  - x8  1 0.45258449        18   21.23848  9.923465
6  - x1  1 0.59205209        19   21.83053  8.610838
7  - x4  1 1.11262480        20   22.94315  7.853593
```

FIGURE 2.4 – Résultat de resf$anova

Dans cette table, on trace l'acheminement de la démarche backward AIC du modèle initial jusqu'au modèle retenu. En effet, dans la première ligne, on a notre modèle initial tenant compte de toutes les variables de $x1$ jusqu'à $x10$. Le degré de liberté de résidu est $n - p = 25 - 11 = 14$. La somme des carrés de résidus est $SCR = 19.92$. Le critère AIC, on l'a aussi calculé et est égal à 16.32. Ensuite, dans chaque ligne correspond le résultat d'une itération de *stepAIC*.

La logique de ce tableau diffère un peu de la logique des tableaux précédents. Car dans les tableaux de *stepAIC*, dans une itération, on testait plusieurs sous-modèles, en écartant une variable mais pas plusieurs en même temps. Par contre, dans la (Fig. 2.4), on commence par négliger $x10$ (résultat de la première itération de step AIC). Le nombre de degré de liberté des résidus est $n - (p - 1) = 15$. Et on a déjà vu que SCR= 19.935 et AIC= 14.341. "Deviance résidu" représente la différence entre la SCR du modèle précédent avec le modèle correspondant. Et ainsi de suite.

Dans la 3ème ligne, on néglige $x3$ donc Df=1, et le degré de liberté est $n-(p-2) = 16$. Et la déviance résiduelle cette fois-ci de la ligne 3 est :

$$DevianceResid._3 = 19.97083 - 19.93548 = 0.03535$$

Et on continue jusqu'à obtenir le modèle retenu par la dernière itération de *stepAIC*.

Validité du modèle retenu

On commence dans un premier temps par afficher le résumé du modèle retenu :

```
> summary(resf)

Call:
lm(formula = x11 ~ x2 + x5 + x6 + x9, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5029 -0.7147 -0.2223  0.4933  2.7684

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.91039    1.10975   2.623  0.01631 *
x2          -0.05126    0.02788  -1.838  0.08090 .
x5           0.08749    0.03114   2.809  0.01083 *
x6           1.79869    0.58389   3.081  0.00590 **
x9          -2.45382    0.67443  -3.638  0.00164 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.071 on 20 degrees of freedom
Multiple R-squared:  0.5079,    Adjusted R-squared:  0.4095
F-statistic: 5.161 on 4 and 20 DF,  p-value: 0.005061
```

FIGURE 2.5 – Résultat de summary(resf)

On remarque que le $R^2 = 0.51$ est encore relativement faible. Cependant, la p-value du test de Fisher de significativité globale est de 0.005. Donc pour des risques α usuelles de 5% et 10%, on rejette l'hypothèse nulle (modèle i.i.d) contre le modèle retenu par *stepAIC*. Le modèle retenu dans ce cas est plus vraisemblable que le modèle complet.

2.2 Sélection descendante par critère BIC

Dans cette section, les mêmes étapes se succèdent. La seule différence, et que le critère BIC (Bayesian Inforamation Criterion) sélectionne dans le cas gaussien le modèle (ω) qui minimise avec les mêmes notation que la section précédente :

$$BIC(\omega) = -2\log L(\omega) + |\omega|\log(n) = cte + n\log\left(\frac{SCR(\omega)}{n}\right) + |\omega|\log(n)$$

On implémente cette approche de la manière suivante :

```
resg=stepAIC(res, k = log(n))
#Vérification : resg est de même classe que res
class(res)
class(resg)
#Propriétés de resg
resg$coef
resg$anova
#Explication des étapes de resg
n*log(deviance(resf1)/n)+10*log(n)
```

FIGURE 2.6 – Implémentation du backward BIC

En affichant *resg$anova*, on a :

```
> resg$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
x11 ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10

Final Model:
x11 ~ x2 + x5 + x6 + x9


   Step Df   Deviance Resid. Df Resid. Dev      AIC
1                          14   19.92132 29.73050
2 - x10  1 0.01416431        15   19.93548 26.52939
3  - x3  1 0.03534687        16   19.97083 23.35480
4  - x7  1 0.81506275        17   20.78589 21.13597
5  - x8  1 0.45258449        18   21.23848 18.45560
6  - x1  1 0.59205209        19   21.83053 15.92409
7  - x4  1 1.11262480        20   22.94315 13.94797
```

FIGURE 2.7 – Résultats du backward BIC

La seule différence réside donc dans la dernière colonne, qui correspond cette fois-ci au critère BIC. On voit bien que dans chaque itération, on minimise ce critère. Ceci est donc bien cohérent. Prenons le modèle resf1 défini auparavant, le critère BIC est :

```
> n*log(deviance(resf1)/n)+10*log(n)
[1] 26.52939
```

FIGURE 2.8 – Résultat d'un calcul de BIC

Ceci correspond donc bien à la dernière colonne de la 2ème ligne de (Fig. 2.7).

On peut remarquer que le résultat de chaque itération pour la sélection descendante est le même pour les 2 critères AIC et BIC. De ce fait, le modèle retenu est ici le même pour les 2 approches. Remarquons que ce résultat n'est pas toujours vrai, et dépend de la taille n de l'échnatillon, et la présence ou pas de la vraie loi dans les modèles (ω) en compétition.

2.3 Sélection mixte par critère AIC

Cette fois-ci, on applique une méthode mixte ou alterné *stepwise selection*. Celle-ci enchaîne les étapes ascendantes et descendantes : à partir d'un modèle initial (dans notre cas $x3 + x6$), l'algorithme étudie la positbilité d'ajouter ou enlever une variable et s'arrête lorsqu'on ne peut plus ajouter ni retrancher de variables.

```
resh=stepAIC(lm(x11~x3+x6, data=df), direction="both", scope=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10)
#Vérification : resh est de même classe que res
class(res)
class(resh)
#Propriétés de resh
resh$coef
resh$anova
```

FIGURE 2.9 – Implémentation de la sélection mixte par critère AIC

On peut afficher les résultats avec *resh$anova* :

```
> resh$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
x11 ~ x3 + x6

Final Model:
x11 ~ x6 + x9 + x5 + x2


  Step Df   Deviance Resid. Df Resid. Dev       AIC
1                          22   44.59756 20.470081
2 + x9  1 10.0483664        21   34.54919 16.087706
3 + x5  1  7.8471289        20   26.70206 11.646623
4 - x3  1  0.1183457        21   26.82041  9.757180
5 + x2  1  3.8772531        20   22.94315  7.853593
```

FIGURE 2.10 – Résultats de la sélection mixte par critère AIC

On remarque que comme prévu on commence bien à partir du modèle $x3 + x6$, puis on rajoute ou retranche des variables à chaque itération. Comme on le voit sur la figure, le critère AIC est donc bien minimisé à chaque étape.

Notons que cette démarche permet de retenir le même modèle que les 2 approches précédentes. Encore une fois, ce résultat n'est pas généralisable et dépend du jeu de données.

2.4 Sélection par recherche exhaustive

Le nombre de variables étant suffisamment faible, on souhaite faire une recherche exhaustive, grâce à la fonction regsubsets du package leaps.

2.4.1 Recherche exhaustive par critère BIC

```
library(leaps)
recherche=regsubsets(x11~., int=TRUE, nbest=1, nvmax=10, method="exhaustive", data=df)
#Interprétation
plot(recherche,scale="bic")
```

FIGURE 2.11 – Implémentation de la recherche exhaustive

Dans cette fonction, on indique la présence d'un intercept avec *int=TRUE*. On indique également le souhait d'avoir un seul meilleur modèle $nbest = 1$ correspondant à un nombre fixe de variable choisi au préalable. L'autre argument $nvmax = 10$ signifie qu'on souhaite avoir les 10 meilleurs modèles.

On peut afficher les résultats sous forme graphique :

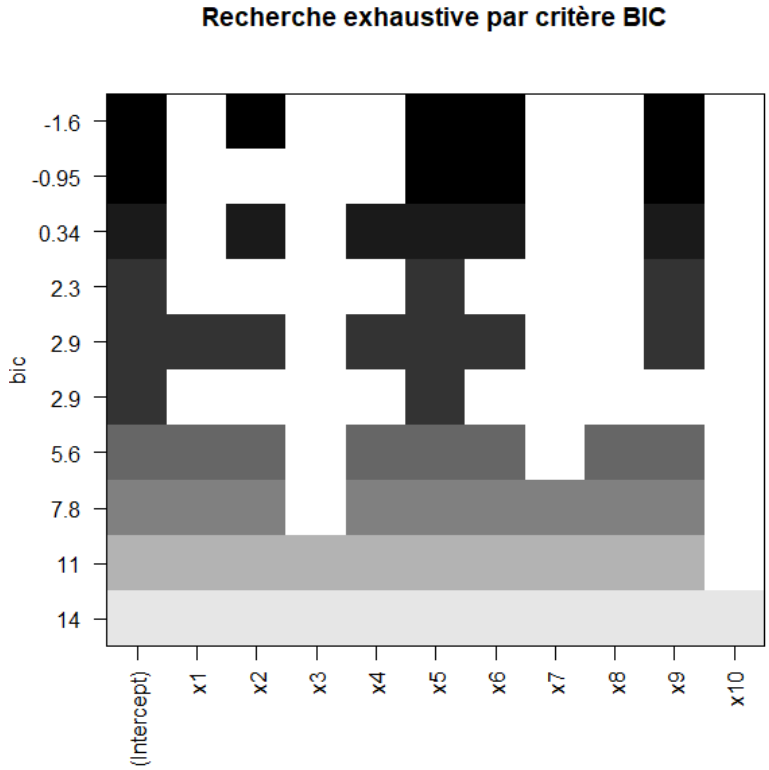

FIGURE 2.12 – Résultat de la recherche exhaustive

En effet, ici l'algorithme nous présente pour chaque nombre de variables compris entre 2 et 11 (compte tenu de l'intercept), le meilleur modèle pour le critère BIC. Et ces meilleurs modèles sont classés par leur critère BIC. En effet, dans chaque ligne on a un nombre de variable retenu différent des autres. C'est la raison pour laquelle d'ailleurs on a 10 lignes ($nvmax = 10$), et si on met $nvmax > 10$, on obtient le même résultat !

On peut aussi voir que le meilleur modèle parmi tous les modèles retenus est le modèle $x11 \sim x2 + x5 + x6 + x9$, qui est le même modèle retenu avec les approches précédentes. Néanmoins, les approches précédentes ne nous garantissent pas le meilleur modèle absolue, mais il se trouve que, dans ce cas particulier, ils permettent de le dégager.

On peut également afficher l'évolution du critère BIC en fonction de l'index

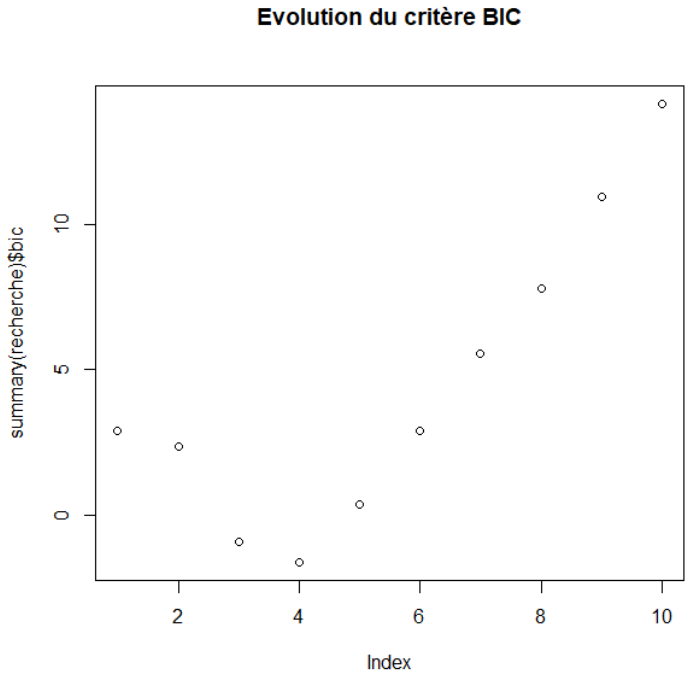

FIGURE 2.13 – Evolution du critère BIC en fonction de l'index

En effet, dans ce graphique, on affiche la valeur du critère BIC pour le meilleur modèle retenu pour un certain nombre fixé de variables entre 1 et 10 (sans tenir en compte de l'intercept). Par exmemple le meilleur modèle avec 7 variables (sans compter l'intercept) retenu par recherche exhaustive a une valeur de BIC associée égale à 5.6 (Fig. 2.12). Ceci est donc bien cohérent.

On peut voir qu'en partant du modèle i.i.d, et en prenant les meilleurs variables incrémentalement, le modèle tend à diminuer son critère BIC jusqu'à k=4 (on apprend des trucs intéressants sur le problème), puis augmente après (ce qui correspond à un surapprentissage). Le modèle retenu a donc bien 4 variables, ce qui correspond au minimum de la courbe.

2.4.2 Recherche exhaustive par multiples critères

On peut reprendre cette recherche exhaustive, mais cette fois-ci avec les critères BIC, R_a^2 (ou adjr2), R^2, et Cp. Cp est le critère de Mallows, et les autres sont définis plus haut. Notons que dans cette recherche, on souhaite soit minimiser BIC ou Cp, soit maximiser R_a^2 ou R^2. Le résultat est le suivant :

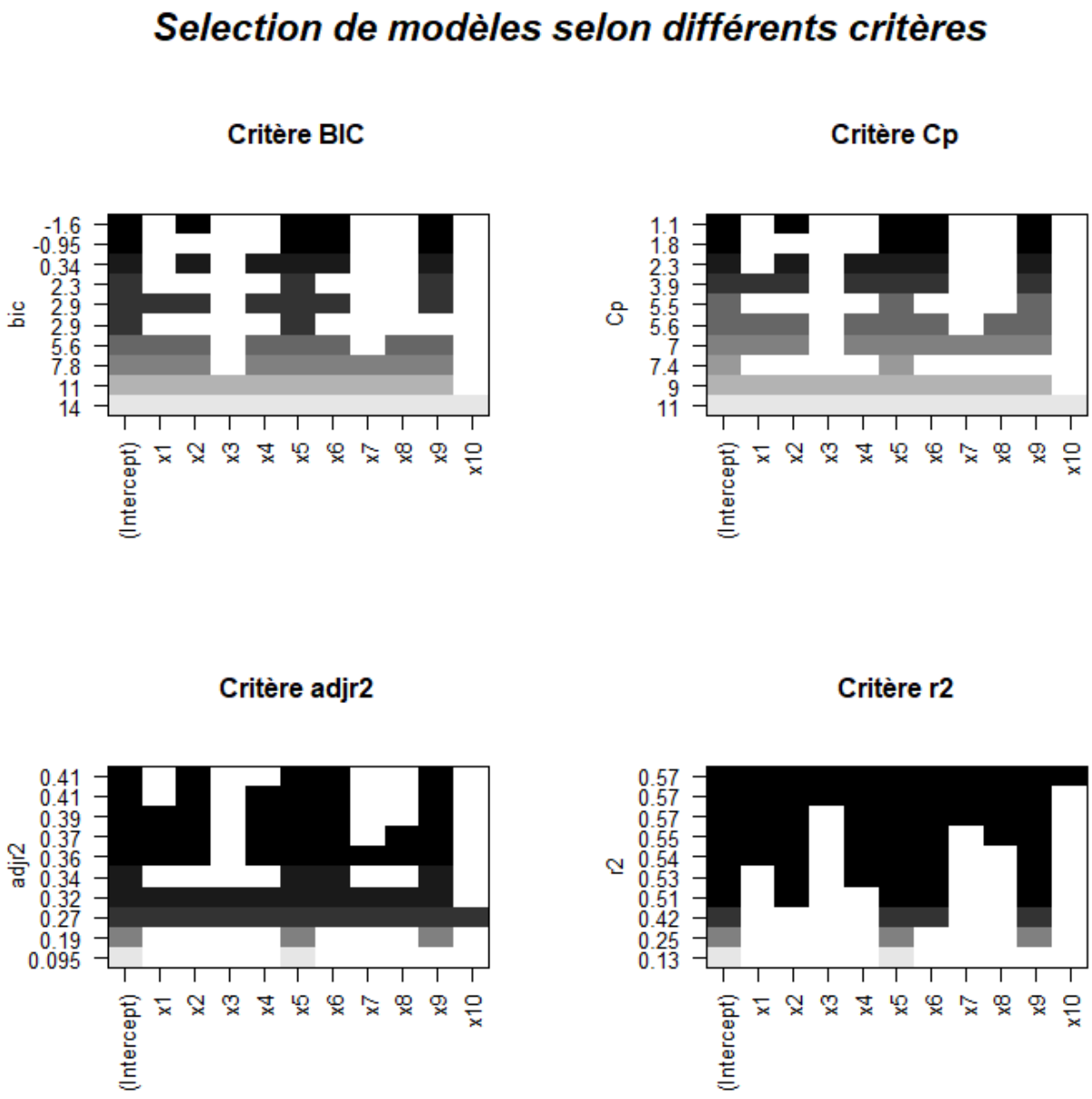

FIGURE 2.14 – Recherche exhaustive selon plusieurs critères

Suivant la direction de l'axe des ordonnées de chaque courbe, on valide le fait qu'on cherche soit à maximiser $r2$, $adjr2$; soit à minimiser BIC, Cp.

On remarque aussi que le meilleur modèle retenu pour chacun des critères BIC, Cp et $adjr2$ est le même modèle correspondant à ce qu'on a vu auparavant ($x11 \sim x2+x5+x6+x9$). Ce résultat n'est pas généralisable, et dépend donc du jeu de données. Par contre, les meilleurs modèles intermédiaires pour chaque critère diffèrent de l'un l'autre. Il n'y a en effet aucune raison qu'ils soient similaires.

Seul le critère $r2$ constitue un cas particulier, puisque comme on peut s'en attendre, $r2$ ne peut qu'augmenter si l'on ajoute un prédicteur, ce critère ne peut donc être employé pour choisir la taille d'un sous-ensemble de prédicteurs. Pour choisir la taille d'un sous-ensemble de prédicteur, on peut pénaliser $r2$ en définissant un coefficient de détermination multiple corrigé qui est $R_a^2 = adjr2$

Chapitre 3

Validation des modèles retenus

Jusqu'ici, nous avons sélectionner des modèles compte tenu de leur performance sur le jeu de données qui nous est disponible. Cependant, il n'y a aucune raison que ces modèles généralisent bien, c'est-à-dire qu'ils performent bien sur un nouvel ensemble de données. D'où, la nécessité d'étudier leur performance sur de nouvelles données. Or, comme on n'a pas d'autres données que celles disponibles, on peut procéder par validation croisée.

3.1 Validation croisée par Leave One Out

Dans cette approche, l'échantillon est alors divisé en n parties. Puis, à tour de rôle, chaque partie est retirée de l'échantillon d'apprenstissage : l'estimation se fait sur l'échantillon tronqué, et l'erreur de prédiction est calculée sur la partie mise en réserve.

On commence par prendre 2 modèles et appeler la fonction CV du package forecast.

```
> resf1=lm(x11~x1+x2+x3+x4+x5+x6+x7+x8+x9,data=df)
> resf2=lm(x11~x1+x2+x4+x5+x6+x7+x8+x9+x10,data=df)
> CV(resf1)
       CV       AIC      AICc       BIC     AdjR2
 2.472935 16.340633 36.648326 29.748267  0.315875
> CV(resf2)
        CV        AIC       AICc        BIC      AdjR2
 2.8001099 16.3809961 36.6886884 29.7886302  0.3147696
```

FIGURE 3.1 – Résultat de la fonction CV

La fonction CV affiche donc plusieurs critères parmi les quelles on a AIC, AICc (AIC corrigé), BIC, adjr2 et CV qui est l'erreur quadratique moyenne de prédiction EQMP dans le cas d'une cross-validation par Leave One Out (LOO).

On peut remarquer d'ailleurs que les critères AIC et BIC ne donnent pas les mêmes valeurs que ceux calculés précédemment. Cependant en calculant l'écart entre les 2 valeurs

pour chacun des modèles resf1 et resf2, on a :

```
> #Comparaison avec le critère AIC de CV
> CV(resf1)[2]-(n*log(deviance(resf1)/n)+2*10)
AIC
  2
> CV(resf2)[2]-(n*log(deviance(resf2)/n)+2*10)
AIC
  2
> #Comparaison avec le critère BIC de CV
> CV(resf1)[4]-(n*log(deviance(resf1)/n)+log(n)*10)
     BIC
3.218876
> CV(resf2)[4]-(n*log(deviance(resf2)/n)+log(n)*10)
     BIC
3.218876
```

FIGURE 3.2 – Interprétation des résultats de la fonction CV

On remarque que cette différence est constante entre les 2 modèles. On peut ainsi affirmer que les critères AIC et BIC de la fonction CV sont définis à une constante près par rapport à ce qu'on a vu précédemment.

De plus, le critère CV ou PRESS est donné téoriquement par :

$$PRESS = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i}{1-h_i}\right)^2$$

où h_i est le i-ème élément diagonal de la matrice de projection $H_X = X(X^TX)^{-1}X^T$.

Dans cette expression, tous les éléments sont accessibles, plus précisément :

$$\widehat{Y} = \text{resc1\$fitted} \quad \text{et} \quad H_X = \text{lm.influence(resc1)\$hat}$$

En comparant les résultats, on a :

```
> #Calcul et comparaison de PRESS dans le cas Leave One Out
> resc1=lm(x11~.,data=df)
> resc2=lm(x11~x2+x5+x6+x9,data=df)
> CV(resc1)
        CV        AIC        AICc        BIC       AdjR2
 3.2701291 18.3228642 44.3228642 32.9493741  0.2675297
> (1/n)*sum(((Y-resc1$fitted)/(1-lm.influence(resc1)$hat))^2)
[1] 3.270129
> CV(resc2)
        CV        AIC        AICc        BIC       AdjR2
 1.8306622  9.8535930 14.5202596 17.1668479  0.4094956
> 1/n*sum(((Y-resc2$fitted)/(1-lm.influence(resc2)$hat))^2)
[1] 1.830662
```

FIGURE 3.3 – Calcul du coefficient PRESS et comparaison

Ainsi , on voit qu'on obtient des valeurs cohérentes pour les 2 modèles. C'est donc réconfortant. De plus, l'EQMP dans le modèle retenu est largement inférieur (près de la moitié) de celle du modèle complet. On peut donc conclure que dans ce cas, le modèle retenu généralise mieux que le modèle complet.

3.2 B-fold Cross-validation

Cette fois-ci l'échantillon est divisé en B=10 parties de tailles n^B égales sauf éventuellement la dernière. L'EQMP dans ce cas est donnée par :

$$\widehat{EQMP}(B) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{n^b}||Y^b - X^b\widehat{\theta}_w^{(-b)}||^2$$

Pour pouvoir calculer cette erreur, on implémente la fonction suivante :

```
#Implémentation de la K-fold cross validation
K=n
j=1
i=0
len = n%/%K
EQMP=0
for (i in 1:(K-1)){
  df_i=df[-(j:(j-1+len)),]
  model=lm(x11~., data=df_i)
  EQMP=EQMP+(sum((Y[j:(j-1+len)]-X[j:(j-1+len),]%*%model$coef)^2))
  print(1/(len)*(sum((Y[j:(j-1+len)]-X[j:(j-1+len),]%*%model$coef)^2)))
  j<-j+len
}
df_last=df[-(j:n),]
model=lm(x11~., data=df_last)
EQMP=EQMP+1/(len)*(sum((Y[j:(j-1+len)]-X[j:(j-1+len),]%*%model$coef)^2))
EQMP=EQMP/K
EQMP
```

FIGURE 3.4 – Implémentation de la fonction de validation croisée

On commence dans un premier temps par valider l'implémentation en prenant K=n, c'est-à-dire en se plaçant dans le cadre de la section précédénte. Puis, on calcule PRESS dans ce cas pour le modèle complet et le modèle retenu, et on trouve bien les mêmes valeurs (voir script).

Ensuite, on remplace K par 10. En déroulant notre algorithme, on trouve :

— Pour le modèle complet, EQMP = 2.935 (voir script)
— Pour le modèle retenu, EQMP = 1.830 (voir script)

Dans un premier temps, on remarque qu'à partir d'une validation croisée en 10 splits, l'EQMP du modèle retenu est toujours plus faible que celle du modèle complet. Le modèle retenu généralise mieux que le modèle complet. De plus, on remarque que ces valeurs sont plus grandes que celles retenues par PRESS, mais cela dépend du jeu de données.

Conclusion

Pour conclure, on vient de voir plusieurs méthodes pour sélectionner et valider le modèle. On a donc pu, par plusieurs critères différents, identifier le modèle dans lequel le log décimal du nombre de nids de processionnaires par arbre d'une placette $x11$ dépend de la pente $x2$, le diamètre de l'arbre $x5$, la densité de peuplement $x6$ et le nombre de strates de vegetation $x9$. Pour pouvoir protéger les forêts des chenilles processionnaires du pin, il faut donc agir sur ces 4 paramètres à priori.

Remarquons qu'on a pris ici le log décimal du nombre de nids, car les nombres de nids sont relativement élevés, et les écarts entre elles relativement faibles. Pour mieux proportionner les écarts entre valeurs réelles et valeurs estimées par le modèle, on a pris le log décimal pour les déserrer. Dans le modèle de sortie, on peut donc appliquer l'exponentiel aux valeurs estimées pour retoruver la grandeur intéressante qui est le nombre de nids.

D'un point de vue technique, cette étude peut encore aller plus loin si on veut tenir en compte plus précisément des variables $x7$ et $x10$, en les modélisant par des variables qualitatives. On peut par exemple représenter chaque valeur a_i de l'ensemble discret par une variable prenant 0 ou 1, selon que l'observation affiche ou non cette valeur a_i.