

Modèles Linéaires

Correction TD 5 : Régression Multiple : Construction de modèles et validation des hypothèses Licence 3 MIASHS

Guillaume Metzler

Institut de Communication (ICOM)

Université de Lyon, Université Lumière Lyon 2

Laboratoire ERIC UR 3083, Lyon, France

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

alejandro.rivera@univ-lyon2.fr

Résumé

Dans cette section, nous regarderons comment construire un bon modèle à partir d'un jeu de données dont nous disposons. En effet, le cadre d'étude des modèles linéaires gaussiens a imposé beaucoup d'hypothèses quant aux données mais aussi sur les résidus du modèles.

Nous verrons comment utiliser ces points là, mais aussi d'autres outils, pour tenter de construire un bon modèle à partir d'un sous ensemble des informations à notre disposition.

Plus précisément, nous allons

- Etudier la colinéarité au sein de nos données
- Tester la significativité des covariables.
- Construire un bon modèle à l'aide d'une stratégie dite *backward*, i.e. en supprimant des covariables pas à pas.
- Etudier les résidus du modèle.

Dans le présent TD, on considérera toujours le modèle de régression linéaire multiple gaussien à p covariables

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon.$$

L'étude effectuée se basera sur le même jeu de données que lors du précédent TD, à savoir *attendance*, où le but est de prédire le nombre de spectateurs à une rencontre sportive (match de baseball) à l'aide de différentes caractéristiques.

On redonne les différentes sorties associées à notre modèle

```
# Pour charger un jeu de données
data = read.csv("../data/attendance.csv", header = TRUE)
n = nrow(data)
p = ncol(data) - 1
# Régression linéaire
mymodel = lm(attendance ~ ., data)
summary(mymodel)

##
## Call:
## lm(formula = attendance ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -974.31 -280.59  -36.55   315.90 1067.98
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1239.40512   327.07553    3.789  0.000202 ***
## temperature    -4.03087     7.76298   -0.519  0.604189
## promotion      47.92031    66.49461    0.721  0.471992
## weekend         32.96119    63.49093    0.519  0.604255
## seats          0.34933     0.04671    7.479  2.6e-12 ***
## size           0.34210     0.03394   10.079 < 2e-16 ***
## rateofwins     -1.80828     2.85917   -0.632  0.527844
## rateofoppwins   4.01839     2.97794    1.349  0.178802
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 434.2 on 192 degrees of freedom
## Multiple R-squared:  0.5545, Adjusted R-squared:  0.5383
## F-statistic: 34.14 on 7 and 192 DF,  p-value: < 2.2e-16
```

Nous avons précédemment vu que toutes les covariables de ce modèle ne sont pas significatives. Nous allons donc voir comment rechercher les covariables les plus significatives pour construire notre modèle de régression.

1 Construction d'un bon modèle

La première hypothèse fondamentale pour la construction de notre modèle et la résolution de notre modèle réside dans l'*identifiabilité du modèle*, *i.e.*, dans le fait que la matrice $\mathbf{X}^\top \mathbf{X}$ soit inversible. Cela signifie que les différentes covariables doivent être **linéairement indépendantes**.

De fortes colinéarités dans les covariables peuvent entraîner une mauvaise estimation des coefficients de la régression, attribuant ainsi un poids significatif à une variable qui ne l'est pas, ou, au contraire, en attribuant un poids faible à une variable du modèle qui est pourtant importante pour la prédiction.

Pour détecter ces **colinéarités**, on emploie un critère appelé le **VIF** pour *Variation Inflation Factor* [Snee, 1981, James et al., 2013] qui sert à mesurer l'impact de la variable sur l'erreur standard σ comparé au cas où la variable aurait une corrélation nulle avec toutes les autres covariables.

Le VIF, VIF_j , d'une variable X_j , $j \in \llbracket 1, p \rrbracket$ est défini par

$$VIF_j = \frac{1}{1 - R_j^2},$$

où R_j^2 désigne le coefficient de détermination du modèle de régression consistant à estimer les valeurs de la covariable X_j à l'aide de toutes les autres covariables indépendantes X_k , $k \neq j$:

$$X_j \sim X_1 + X_2 + \dots + X_{j-1} + X_{j+1} + X_p.$$

En général, on dit qu'une variable présente des colinéarités avec les autres covariables du jeu de données lorsque le VIF est supérieur à 10. Si plusieurs covariables ont un VIF supérieur à 10, on commence par éliminer la variable avec le plus grand VIF avant de le recalculer avec les variables restantes, jusqu'à ce que toutes les covariables aient un VIF inférieur à 10.

1. Calculer le VIF de chaque variable afin de détecter d'éventuel colinéarité.

Pour cela on doit effectuer les différentes régressions pour lesquelles on extrait le coefficient de détermination permettant de calculer le VIF.

```
# Modèles de régression
model_temp = lm(temperature~. -attendance,data)
model_prom = lm(promotion~. -attendance,data)
```

```

model_weekend = lm(weekend~. -attendance,data)
model_seats = lm(seats~. -attendance,data)
model_size = lm(size~. -attendance,data)
model_rateofwins = lm(rateofwins~. -attendance,data)
model_rateofoppwins = lm(rateofoppwins~. -attendance,data)

# Calcul des VIF
VIF_temp = 1/(1-summary(model_temp)$r.squared)
VIF_prom = 1/(1-summary(model_prom)$r.squared)
VIF_weekend = 1/(1-summary(model_weekend)$r.squared)
VIF_seats = 1/(1-summary(model_seats)$r.squared)
VIF_size = 1/(1-summary(model_size)$r.squared)
VIF_rateofwins = 1/(1-summary(model_rateofwins)$r.squared)
VIF_rateofoppwins = 1/(1-summary(model_rateofoppwins)$r.squared)

VIF_temp
## [1] 1.007221

VIF_prom
## [1] 1.011979

VIF_weekend
## [1] 1.021818

VIF_seats
## [1] 1.152905

VIF_size
## [1] 1.149199


VIF_rateofwins
## [1] 1.025278

VIF_rateofoppwins
## [1] 1.011795

```

2. Le jeu de données initiale présente-t'il de fortes colinéarités ?

Les valeurs des différents VIF sont toutes inférieures à 10, on ne peut donc pas dire que le jeu de données présente des colinéarités.

3. Comparer les résultats avec ceux de la fonction VIF de 

```
library(car)

## Loading required package: carData
vif(mymodel)

##      temperature      promotion      weekend      seats      size
##      1.007221      1.011979      1.021818      1.152905      1.149199
##      rateofwins rateofoppwins
##      1.025278      1.011795
```

Lorsque nous avons extrait le bon sous-ensemble de covariables, nous pouvons maintenant regarder si ces dernières sont toutes pertinentes pour notre modèle de régression.

Une méthode consiste à regarder la performance du global lorsque l'on retire une variable de notre modèle et chercher à voir si l'impact est significatif ou non, cette méthode s'appelle la méthode *backward*. Plus précisément, partant du modèle

$$\mathbf{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p,$$

elle consiste à évaluer la performance des modèles suivants

$$\forall j \in \llbracket 1, p \rrbracket, \quad Y \sim X_1 + X_2 + \dots + X_{j-1} + X_{j+1} + X_p.$$

Le modèle initial et ses différents sous-modèles sont évalués à l'aide d'un critère performance comme

- le R^2 ajusté, que l'on cherchera à maximiser
- l'*AIC* ou le *BIC* que l'on cherchera à minimiser.

Si un des sous-modèles à un *AIC* ou *BIC* plus faible (ou un R^2 ajusté plus élevé) il est alors privilégié par rapport au modèle complet.

Le processus est ensuite itéré jusqu'à ce que l'on ne puisse plus retirer de covariables, *i.e.*, jusqu'à que le critère de performance ne puisse plus être amélioré.

1. Appliquer la méthode de stratégie *backward* à votre jeu de données, après avoir retiré les covariables avec le VIF.

On conserve ici toutes les covariables, on va maintenant retirer les covariables une à une pour déterminer celles qui a le moins d'impact dans le modèle et qui, en la supprimant permet une amélioration de l' R^2_{aj} .

```

# On va créer une première matrice qui va enregistrer l'AIC
# des différents modèles correspondant à la suppression d'une variable ||

# Métriques du modèle complet
Full_AIC = AIC(lm(attendance~.,data))
Full_R2 = summary(lm(attendance~.,data))$adj.r.squared

Val_AIC = matrix(0,nrow = 2,ncol = p)
colnames(Val_AIC) = names(data[,-which(names(data)=="attendance")])
rownames(Val_AIC) = c("AIC", "Adjusted_R_square")

# On va faire une première boucle pour supprimer une variable

# L'indice i va servir à compléter la table Val_AIC
i = 1
for (j in names(data)){
  if (j=="attendance"){
  }
  else {
    data_Loop = data[,-which(names(data)%in%c(j))]
    mylm <- lm(attendance~., data_Loop)
    R_adj = summary(mylm)$adj.r.squared
    Val_AIC[,i] = c(AIC(mylm), R_adj)
    i = i+1
  }
}
Val_AIC


##           temperature  promotion    weekend      seats
## AIC          3005.1180254 3005.377642 3005.1179224 3055.9759141
## Adjusted_R_square    0.5400165    0.539419    0.5400167    0.4068305
##           size  rateofwins rateofoppwins
## AIC          3089.7711643 3005.2536028 3006.7251498
## Adjusted_R_square    0.2976328    0.5397046    0.5363053

```

On voit donc, dans ce cas là, qu'il faudrait supprimer la variable *weekend* en premier. On doit ensuite itérer le processus jusqu'à ce qu'il n'y ait plus d'amélioration en terme du critère étudié.

2. Quelles sont les covariables significatives pour notre tâche.

Si on itère le processus jusqu'au but, on trouve que seules deux variables sont significatives : *size* et *seats*.

3. On pourra comparer les résultats avec le fonction *step* de 

```
step(mymodel)

## Start:  AIC=2437.26
## attendance ~ temperature + promotion + weekend + seats + size +
##      rateofwins + rateofoppwins
##
##              Df Sum of Sq      RSS      AIC
## - weekend      1      50818 36252899 2435.5
## - temperature 1      50836 36252918 2435.5
## - rateofwins   1      75420 36277502 2435.7
## - promotion    1      97926 36300008 2435.8
## - rateofoppwins 1     343325 36545406 2437.2
## <none>                        36202081 2437.3
## - seats        1    10547687 46749769 2486.4
## - size         1    19153947 55356028 2520.2
##
## Step:  AIC=2435.54
## attendance ~ temperature + promotion + seats + size + rateofwins +
##      rateofoppwins
##
##              Df Sum of Sq      RSS      AIC
## - temperature  1      54100 36306999 2433.8
## - rateofwins   1      90153 36343052 2434.0
## - promotion    1     103604 36356504 2434.1
## - rateofoppwins 1     342580 36595479 2435.4
## <none>                        36252899 2435.5
## - seats        1    10498768 46751667 2484.4
## - size         1    19310600 55563499 2518.9
##
## Step:  AIC=2433.84
## attendance ~ promotion + seats + size + rateofwins + rateofoppwins
##
##              Df Sum of Sq      RSS      AIC
## - rateofwins   1      95523 36402522 2432.4
## - promotion    1     104272 36411271 2432.4
## - rateofoppwins 1     347929 36654928 2433.8
## <none>                        36306999 2433.8
## - seats        1    10448162 46755161 2482.4
## - size         1    19487143 55794142 2517.8
##
## Step:  AIC=2432.37
## attendance ~ promotion + seats + size + rateofoppwins
```

```
##
##           Df Sum of Sq      RSS      AIC
## - promotion      1      117186 36519708 2431.0
## - rateofoppwins  1      359803 36762325 2432.3
## <none>                        36402522 2432.4
## - seats           1  10540907 46943428 2481.2
## - size            1  19396563 55799085 2515.8
##
## Step:   AIC=2431.01
## attendance ~ seats + size + rateofoppwins
##
##           Df Sum of Sq      RSS      AIC
## - rateofoppwins  1      347795 36867502 2430.9
## <none>                        36519708 2431.0
## - seats           1  10472428 46992135 2479.4
## - size            1  19293931 55813639 2513.8
##
## Step:   AIC=2430.9
## attendance ~ seats + size
##
##           Df Sum of Sq      RSS      AIC
## <none>                        36867502 2430.9
## - seats      1  10205231 47072733 2477.8
## - size       1  19610767 56478270 2514.2
##
## Call:
## lm(formula = attendance ~ seats + size, data = data)
##
## Coefficients:
## (Intercept)      seats      size
##    1267.8841     0.3399     0.3430
```

Obtenez vous le même modèle ?

Cette méthode *backawrd* est en fait une application du test effectué au TD 3 sur les modèles emboîtés.

Maintenant que l'on a obtenu notre meilleur modèle, on peut chercher à examiner les résidus pour savoir si notre modèle valide les hypothèses de notre étude.

2 Etude des résidus

Les hypothèses à vérifier concernant les résidus de notre modèle sont les suivantes

- normalité des résidus : ces derniers doivent suivre une distribution normale.
- homoscélasticité des résidus : ils doivent avoir la même variance, *i.e.* les valeurs des résidus ne doivent pas dépendre des valeurs des covariables X_j .
On dit souvent que les résidus doivent se trouver dans un «*bande centrée en 0*».
- indépendance des résidus : les valeurs des résidus doivent se trouver dans une certaine tranche de valeurs en fonction des prédictions du modèle et ne pas présenter de motifs.
On dit souvent que les résidus doivent se trouver dans un «*bande centrée en 0*» et ne pas présenter de motifs.

On va se concentrer uniquement sur le modèle que nous avons retenu à la section précédente, *i.e.*, celui obtenu après utilisation de la sélection de modèle *backward*.

Pour tester la normalité des résidus, on utilisera le test de Shapiro-wilk, ou on pourra également représenter les résidus sur la *Droite de Henry*. On préfère souvent la deuxième solution et effectuer une analyse qualitative des résidus car le test de Shapiro est très rigide.

1. Tester la normalité des résidus et conclure avec

(a) le test de shapiro

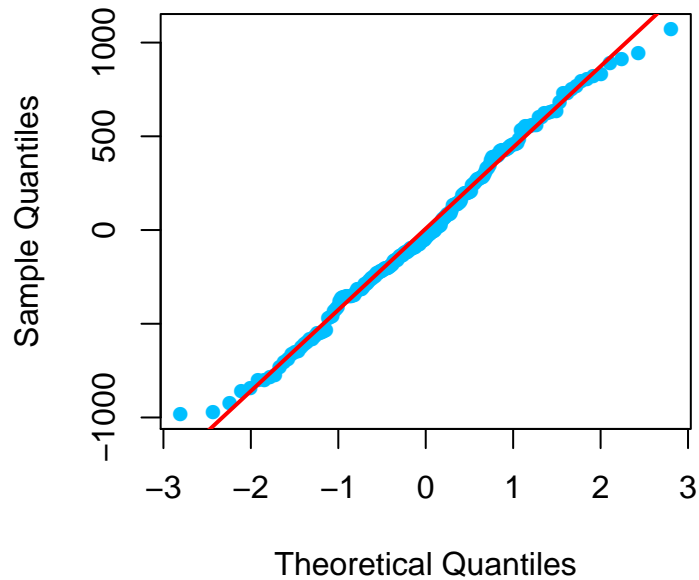
```
mymodel = lm(attendance~size + seats, data = data)
shapiro.test(mymodel$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  mymodel$residuals
## W = 0.99176, p-value = 0.3173
```

(b) ou en traçant la droite de Henry

```
res = mymodel$residuals
qqnorm(res, pch = 16, col = "deepskyblue")
qqline(res, lwd = 2, col = "red")
```

Normal Q–Q Plot



2. Pourrait-on effectuer un autre test pour tester la normalité des résidus. Si oui, lequel ?

Nous pourrions également effectuer un test de Kolmogorov-Smirnov.

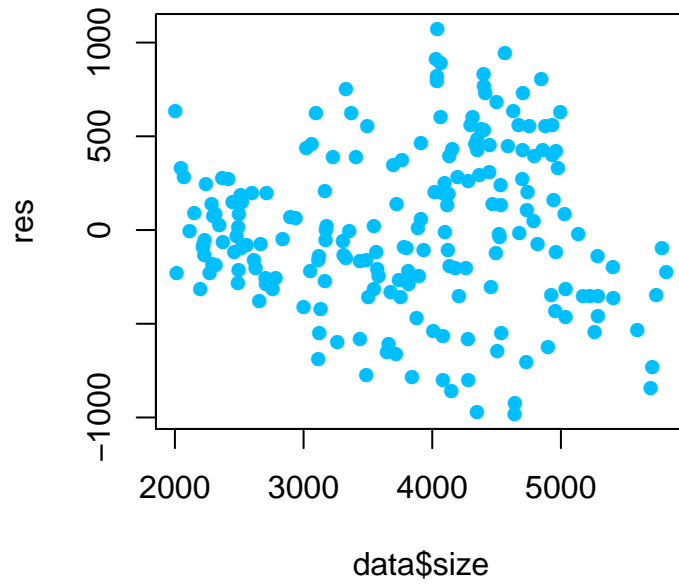
On s'intéresse maintenant à l'hypothèse d'égale variance des résidus, cette analyse se fait uniquement graphiquement et consiste à regarder si les résidus se trouvent globalement dans une bande centrée en 0.

On va donc faire un graphe (ε, X_j) avec toutes les variables X_j .

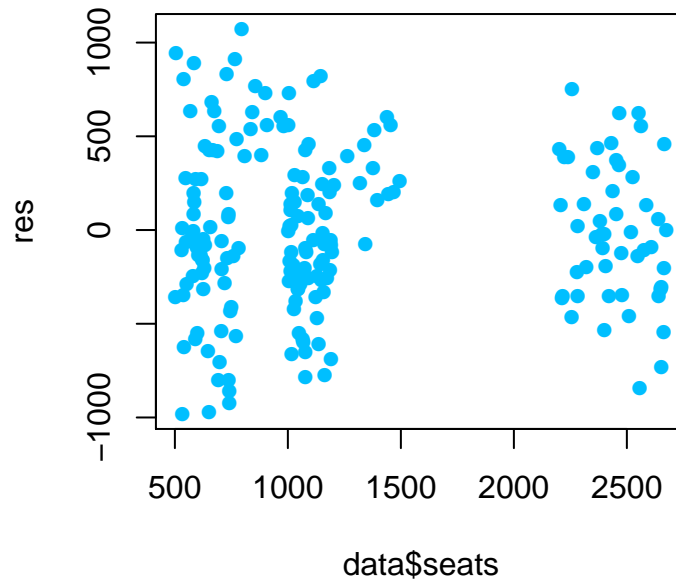
3. Effectuer l'analyse des résidus et conclure quant à l'homoscédasticité des résidus.

On représente les différents graphes

```
plot(data$size, res, pch = 16, col = "deepskyblue")
```



```
plot(data$seats,res, pch = 16, col = "deepskyblue")
```



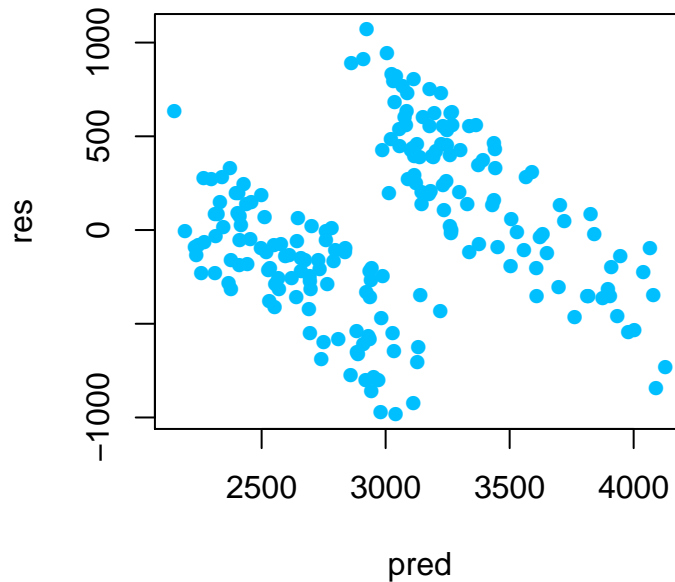
L'hypothèse d'homoscédasticité ne semble pas être contredite par ces deux graphiques.

Enfin, pour tester l'indépendance, on effectuera un graphe (ε, \hat{y}) .

4. Effectuer l'analyse des résidus et conclure quant à l'hypothèse d'indépendance.

A nouveau, on fera une analyse graphique des résidus

```
pred = mymodel$fitted.values
plot(pred, res, pch = 16, col = "deepskyblue")
```



La notion d'indépendance n'est ici pas évidente étant donnée la courbe des résidus. On ne peut donc pas dire que les hypothèses sont vérifiées.

Remarque En réalité, les résidus ε_i ont rarement la même variance au sens strict du terme. Ainsi pour étudier l'indépendance des résidus, on va déjà faire en sorte qu'ils aient la même variance. On va donc étudier des résidus **normalisés** r_i définis par

$$r_i = \frac{\varepsilon_i}{\hat{\sigma} \sqrt{1 - H_{i,i}}},$$

où $H_{i,i}$ désigne le i -ème élément sur la matrice $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})\mathbf{X}^\top$, qui est associée à la variance du i -ème résidu.

Le problème est que ces résidus normalisés font apparaître ε_i et $\hat{\sigma}$ qui ne sont pas indépendants (car la définition de $\hat{\sigma}$ met en jeu ε_i). On considère alors une autre transformation de ces résidus que l'on appelle les résidus *studentisés*, \tilde{r}_i , définis par

$$\tilde{r}_i = \frac{\varepsilon_i}{\hat{\sigma}_{(i)} \sqrt{1 - H_{i,i}}},$$

où

$$\hat{\sigma}_{(i)} = \frac{1}{n - p - 2} \sum_{\substack{i=1 \\ i \neq j}}^n \varepsilon_i^2.$$

Il s'agit donc d'une estimation de la variance qui ne prend pas en compte le i -ème résidu.

Ces nouveaux résidus suivent alors une loi de Student et on vérifiera alors qu'un ratio $1 - \alpha$ (on prend en général $\alpha = 0.05$) des valeurs des résidus studentisés \tilde{r}_i , se trouvent dans l'intervalle $[-2, 2]$.

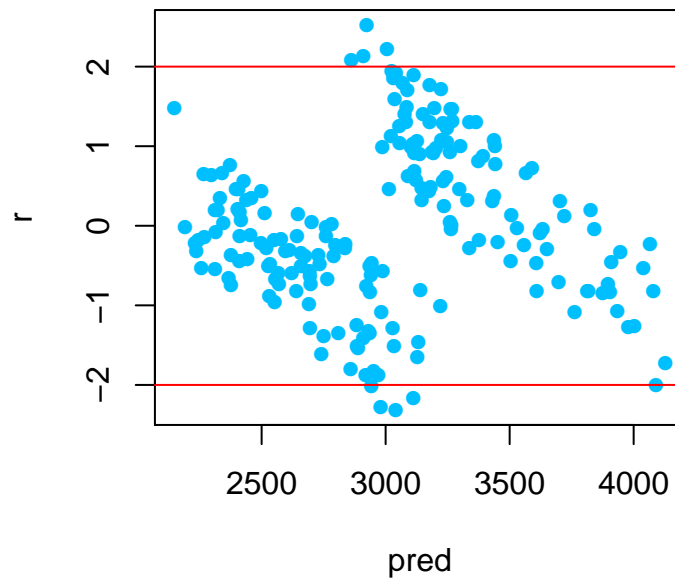
5. Effectuer l'analyse de ces résidus studentisés et conclure.

```
n = nrow(data)
p = 2

# Calcul de la matrice H
X = cbind(c(rep(1,n)), data$seats,data$size)
H = X%%solve(t(X)%%X)%%t(X)

# Calcul des sigma_i
sigma2_i = rep(0,n)
for (i in 1:n){
  sigma2_i[i] = sum(mymodel$residuals[-i]^2)/(n-p-2)
}

# On calcule les résidus studentisés
r = mymodel$residuals/sqrt(sigma2_i*(1-diag(H)))
plot(pred,r, pch = 16, col = "deepskyblue")
abline(h = 2, col = "red")
abline(h = -2, col = "red")
```



3 Etude sur un autre jeu de données

Un institut de biologie effectue une série d'étude concernant le taux de cholestérol. Pour cela, elle a procédé à une étude sur 30 personnes sélectionnées aléatoirement dans la population. Les données de cette étude sont données ci-dessous.

```
# Création des variables
```

```
Cholesterol=c(354, 190, 405, 263, 451, 302, 288, 385, 402, 365,  
             209, 290, 346, 254, 395, 435, 543, 345, 298, 237,  
             421, 498, 348, 123, 283, 319, 361, 298, 271, 246)
```

```
Poids=c(84, 73, 65, 70, 77, 69, 63, 72, 79, 75,  
        47, 89, 65, 57, 59, 65, 80, 69, 59, 78,  
        103, 80, 87, 61, 63, 69, 77, 79, 81, 70)
```

```
Age=c(46, 21, 52, 30, 57, 25, 28, 36, 57, 44,  
      24, 31, 52, 25, 60, 23, 67, 54, 38, 47,  
      39, 21, 45, 84, 53, 59, 37, 98, 45, 56)
```

```
Taille=c(180, 190, 160, 155, 165, 170, 175, 180, 150, 165,
         160, 165, 165, 170, 165, 167, 165, 156, 169, 179,
         185, 168, 156, 182, 190, 165, 147, 166, 188, 170)

data<-data.frame(Cholesterol,Poids,Age,Taille)
```

Ce jeu de données contient les informations suivantes :

- le taux de cholestérol chez l'individu en $mg.L^1$,
- la masse de l'individu en kg ,
- l'âge de l'individu en année,
- la taille de l'individu en cm .

Les biologistes cherchent à déterminer s'il existe un lien entre le taux de cholestérol chez l'individu et les caractéristiques physiques de ce dernier, *i.e.*, en fonction de son *âge*, *poids* et de sa *taille*.

A l'aide d'un modèle de régression linéaire multiple, aider les biologistes à déterminer le meilleur modèle et examiner les résidus pour valider votre modèle.

Références

- [James et al., 2013] James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning*, volume 112. Springer.
- [Snee, 1981] Snee, R. (1981). Who invented the variance inflation factor ?