

BE1- plans d'expériences et régression logistique

Tulio NAVARRO TUTUI, Filipe PENNA CERAVOLO SOARES

26 October, 2022

Exercice 01 - Plan d'expériences - Criblage

1. Étude du plan

Selon l'énoncé, chaque expérience est faite 4 fois. Donc on a 16 expériences différentes. À partir de la lecture du fichier, on réalise que on a un plan fractionnaire, avec $n = 2^{6-2} = 2^4 = 16$.

En étudiant les 16 premières lignes, on note que : $(A.B.C = E)$ et $(A.C.D = F)$. Autrement dit, Ctime est une interaction triple de Ltemp, Ltime et Lpress, et Catmos est une interaction triple de Ltemp, Lpress, Ctemp. De ce fait, les effets principaux se confondent tous avec des interactions triples. Or, comme dans le cours, on fait l'hypothèse que les interactions triples sont d'influence inenvisageable et donc négligeables. De ce fait, les effets principaux peuvent être estimés sans confusion. Cependant, ce n'est pas le cas des effets des interactions doubles.

```
silicium = read.table(file = "silicium.txt", # ./02. Segundo BE/01. Treinamento/silicium.txt
                      header = TRUE)
head(silicium)
```

```
##   Ltemp Ltime Lpress Ctemp Ctime Catmos Camber
## 1    -1    -1    -1    -1    -1    -1    167
## 2     1    -1    -1    -1     1     1     62
## 3    -1     1    -1    -1     1    -1     41
## 4     1     1    -1    -1    -1     1     73
## 5    -1    -1     1    -1     1     1     47
## 6     1    -1     1    -1    -1    -1    219
```

```
p = length(silicium) - 1
n = nrow(silicium)
x = as.matrix(silicium[, -7])
nombre = t(x) %*% x
```

```
p
```

```
## [1] 6
```

```
n
```

```
## [1] 64
```

##		Ltemp	Ltime	Lpress	Ctemp	Ctime	Catmos
##	[1,]	-1	-1	-1	-1	-1	-1
##	[2,]	1	-1	-1	-1	1	1
##	[3,]	-1	1	-1	-1	1	-1
##	[4,]	1	1	-1	-1	-1	1
##	[5,]	-1	-1	1	-1	1	1
##	[6,]	1	-1	1	-1	-1	-1
##	[7,]	-1	1	1	-1	-1	1
##	[8,]	1	1	1	-1	1	-1
##	[9,]	-1	-1	-1	1	-1	1
##	[10,]	1	-1	-1	1	1	-1
##	[11,]	-1	1	-1	1	1	1
##	[12,]	1	1	-1	1	-1	-1
##	[13,]	-1	-1	1	1	1	-1
##	[14,]	1	-1	1	1	-1	1
##	[15,]	-1	1	1	1	-1	-1
##	[16,]	1	1	1	1	1	1
##	[17,]	-1	-1	-1	-1	-1	-1
##	[18,]	1	-1	-1	-1	1	1
##	[19,]	-1	1	-1	-1	1	-1
##	[20,]	1	1	-1	-1	-1	1
##	[21,]	-1	-1	1	-1	1	1
##	[22,]	1	-1	1	-1	-1	-1
##	[23,]	-1	1	1	-1	-1	1
##	[24,]	1	1	1	-1	1	-1
##	[25,]	-1	-1	-1	1	-1	1
##	[26,]	1	-1	-1	1	1	-1
##	[27,]	-1	1	-1	1	1	1
##	[28,]	1	1	-1	1	-1	-1
##	[29,]	-1	-1	1	1	1	-1
##	[30,]	1	-1	1	1	-1	1
##	[31,]	-1	1	1	1	-1	-1
##	[32,]	1	1	1	1	1	1
##	[33,]	-1	-1	-1	-1	-1	-1
##	[34,]	1	-1	-1	-1	1	1
##	[35,]	-1	1	-1	-1	1	-1
##	[36,]	1	1	-1	-1	-1	1
##	[37,]	-1	-1	1	-1	1	1
##	[38,]	1	-1	1	-1	-1	-1
##	[39,]	-1	1	1	-1	-1	1
##	[40,]	1	1	1	-1	1	-1
##	[41,]	-1	-1	-1	1	-1	1
##	[42,]	1	-1	-1	1	1	-1
##	[43,]	-1	1	-1	1	1	1
##	[44,]	1	1	-1	1	-1	-1
##	[45,]	-1	-1	1	1	1	-1
##	[46,]	1	-1	1	1	-1	1
##	[47,]	-1	1	1	1	-1	-1
##	[48,]	1	1	1	1	1	1
##	[49,]	-1	-1	-1	-1	-1	-1
##	[50,]	1	-1	-1	-1	1	1
##	[51,]	-1	1	-1	-1	1	-1

```
## [52,] 1 1 -1 -1 -1 1
## [53,] -1 -1 1 -1 1 1
## [54,] 1 -1 1 -1 -1 -1
## [55,] -1 1 1 -1 -1 1
## [56,] 1 1 1 -1 1 -1
## [57,] -1 -1 -1 1 -1 1
## [58,] 1 -1 -1 1 1 -1
## [59,] -1 1 -1 1 1 1
## [60,] 1 1 -1 1 -1 -1
## [61,] -1 -1 1 1 1 -1
## [62,] 1 -1 1 1 -1 1
## [63,] -1 1 1 1 -1 -1
## [64,] 1 1 1 1 1 1
```

```
nombre
```

```
##      Ltemp Ltime Lpress Ctemp Ctime Catmos
## Ltemp    64    0      0      0      0      0
## Ltime     0    64      0      0      0      0
## Lpress     0     0     64      0      0      0
## Ctemp      0     0      0     64      0      0
## Ctime      0     0      0      0     64      0
## Catmos     0     0      0      0      0     64
```

2. Ajuster un modèle linéaire de la variable Camber en fonction des 6 facteurs. Analyser la sortie summary du modèle

```
mod1 = lm(Camber~., data = silicium)
summary(mod1)
```

```
##
## Call:
## lm(formula = Camber ~ ., data = silicium)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -67.672 -22.703  -3.875   28.797   81.328
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  107.016      4.793   22.328 < 2e-16 ***
## Ltemp        19.453      4.793    4.059 0.000152 ***
## Ltime         2.891      4.793    0.603 0.548823
## Lpress       28.016      4.793    5.845 2.57e-07 ***
## Ctemp        -7.109      4.793   -1.483 0.143492
## Ctime       -17.234      4.793   -3.596 0.000676 ***
## Catmos      -38.734      4.793   -8.082 5.03e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.34 on 57 degrees of freedom
```

```
## Multiple R-squared:  0.6975, Adjusted R-squared:  0.6657
## F-statistic: 21.91 on 6 and 57 DF,  p-value: 3.566e-13
```

```
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: Camber
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Ltemp      1  24219   24219  16.4740 0.0001520 ***
## Ltime      1    535     535   0.3638 0.5488232
## Lpress     1  50232   50232  34.1681 2.574e-07 ***
## Ctemp      1   3235     3235   2.2003 0.1434922
## Ctime      1  19010   19010  12.9304 0.0006762 ***
## Catmos     1  96023   96023  65.3150 5.028e-11 ***
## Residuals 57  83798    1470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- R^2 est trop petite
- Ltime et Ctemp sont pas important pour le modele

3. Retrouver par le calcul le chiffre de la colonne Std Error

En analysant le summary du modèle, on observe que les paramètres Ltemp, Lpress, Ctime et Catoms sont des paramètres explicatifs du modèle. Par contre, Ltime et Ctemp n'en sont pas.

Du summary, on obtient que $\sigma = 38.34$. Donc

```
library(matlib)
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
```

```
## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```
std_error = (38.34*sqrt((inv(nombre))))[1,1]
std_error
```

```
##
## 4.7925
```

4. Estimer un modèle plus simple ne comprenant que les facteurs influents. Comparer l'estimation des coefficients avec le modèle précédent

```
silicium_adj = lm(Camber ~ Ltemp + Lpress + Ctime + Catmos, data = silicium)
summary(silicium_adj)
```

```
##
## Call:
## lm(formula = Camber ~ Ltemp + Lpress + Ctime + Catmos, data = silicium)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -74.953 -26.547  -4.016   25.844   85.547
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   107.016     4.816   22.222  < 2e-16 ***
## Ltemp         19.453     4.816    4.040 0.000157 ***
## Lpress        28.016     4.816    5.818 2.59e-07 ***
## Ctime        -17.234     4.816   -3.579 0.000698 ***
## Catmos       -38.734     4.816   -8.043 4.62e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.53 on 59 degrees of freedom
## Multiple R-squared:  0.6839, Adjusted R-squared:  0.6625
## F-statistic: 31.92 on 4 and 59 DF,  p-value: 3.711e-14
```

Les quatre coefficients obtenus avec ce modèle sont strictement similaires aux quatre coefficients associées aux mêmes variables explicatives (identifiés en jaune précédemment) dans le modèle précédent qui prenait en compte les six variables explicatives. De ce fait, en supprimant deux variables explicatives, aucun potentiel effet d'interaction n'a été perdu. Ceci valide la suppression des deux variables Ltime et Ctemp.

5.

5.1. Quelles sont les conditions expérimentales qui permettent de minimiser la courbure Camber ?

Pour minimiser la courbure de la plaque de silicium dans notre modèle, on veut d'une part maximiser l'influence des facteurs à influence négative sur la courbure, et d'autre part minimiser l'influence des facteurs à influence positive sur la courbure. Ainsi, on va se placer en 1 pour Ctime et Catmos et en -1 pour Ltemp et Lpress. Autrement dit, les conditions expérimentales pour minimiser la courbure sont : Ctime 29 secondes ; Catmos 26°C ; Ltemp 55 °C et Lpress 5bars.

5.2. Donner un intervalle de confiance pour la courbure moyenne en ce point de fonctionnement optimal

On cherche maintenant un intervalle de confiance pour la courbure moyenne en ce point de fonctionnement optimal.

```
frame = data.frame(Ltemp = -1, Lpress = -1, Ctime = 1, Catmos = 1)
base_prediction = data.frame(predict(silicium_adj, frame, interval="confidence", level=0.95))
base_prediction
```

```
##          fit      lwr      upr
## 1 3.578125 -17.9689 25.12515
```

```
upper_limit = 25 + 2*38.53/sqrt(64)
upper_limit
```

```
## [1] 34.6325
```

5.3. Quel est l'impact sur la courbure d'une augmentation de 5 degrés C de la température de laminage ?

On étudie enfin l'influence dans notre modèle d'une augmentation de 5°C de la température de laminage de la plaque (soit +0,5 en valeurs normalisées par rapport à l'intervalle [55C;75C]).

```
frame_augment = data.frame(Ltemp = -0.5, Lpress = -1, Ctime = 1, Catmos = 1)
adj_prediction = data.frame(predict(silicium_adj, frame_augment, interval="confidence", level=0.95))
dif = adj_prediction$fit - base_prediction$fit
dif
```

```
## [1] 9.726562
```

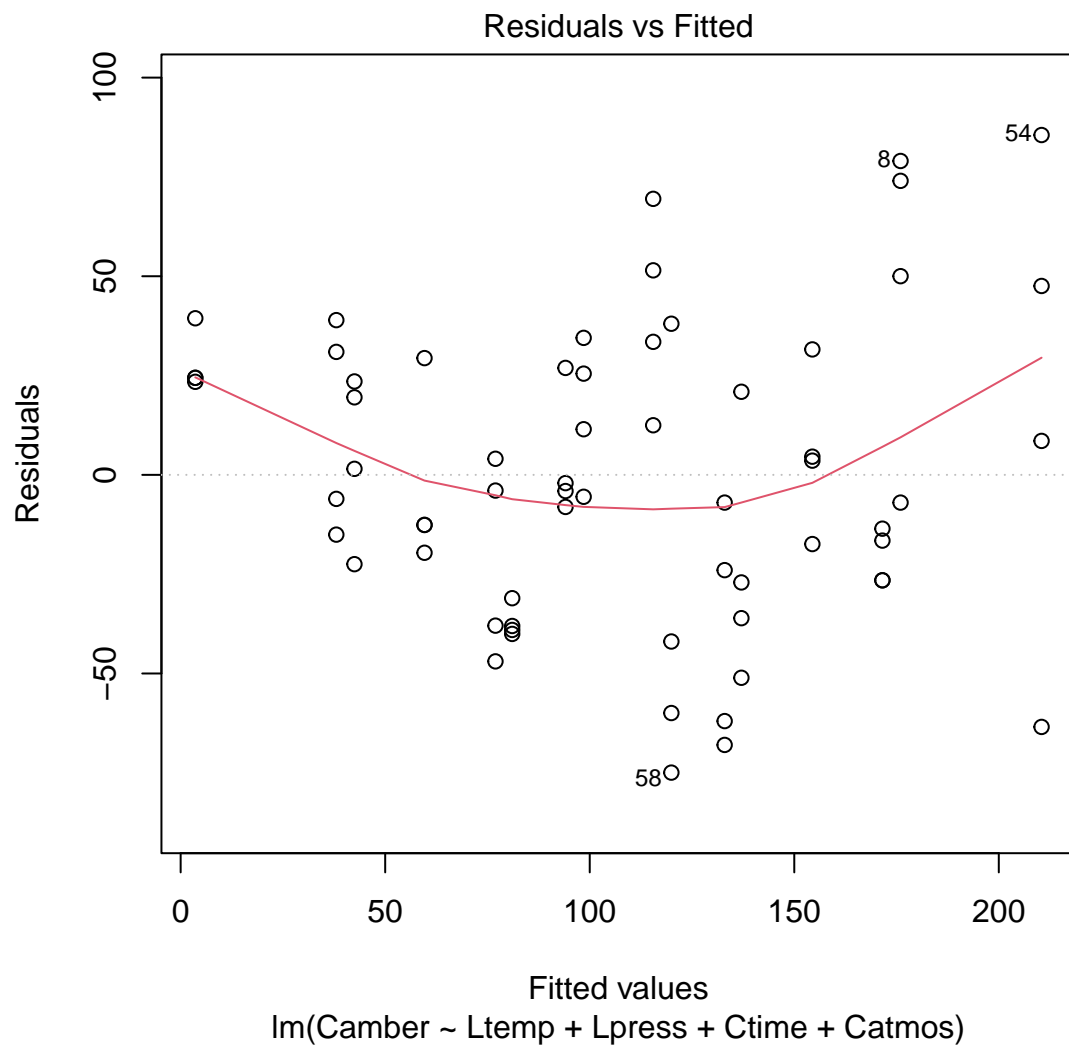
Précédemment on avait en les points minimum une courbure attendue de 3,57. Ici, après une augmentation de 5°C on obtient une courbure de 13,30. Ainsi, une telle augmentation de la température de laminage aboutit à une augmentation de 9,73 de la courbure.

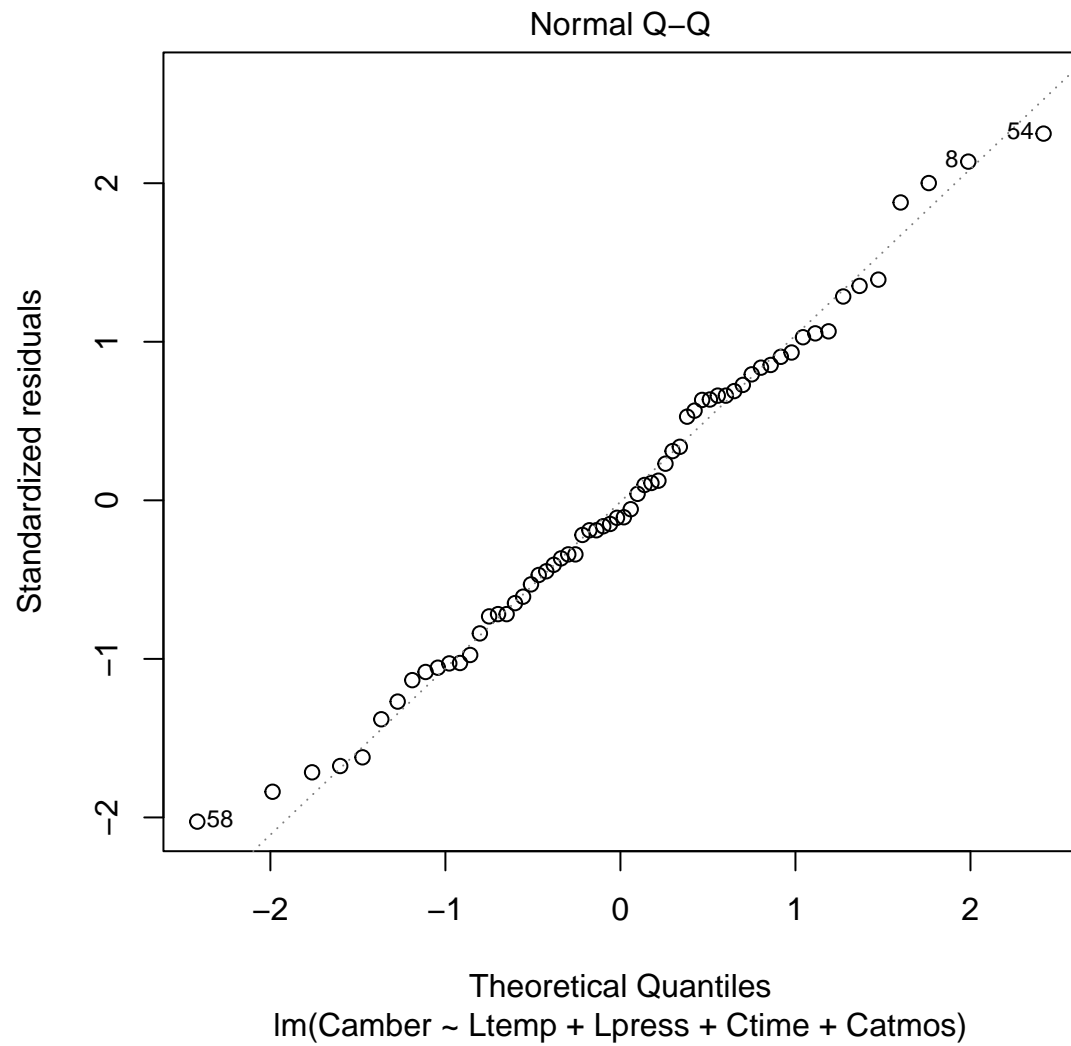
6. Les hypothèses du modèle sont-elles vérifiées ? expliquer

L'hypothèse d'une dépendance uniquement linéaire en les quatre variables explicatives Ltemp, Lpress, Ctime et Catmos n'est pas parfaitement vérifiée. En effet, le modèle de régression obtenu a un coefficient de régression $R^2=0,6839$ assez faible.

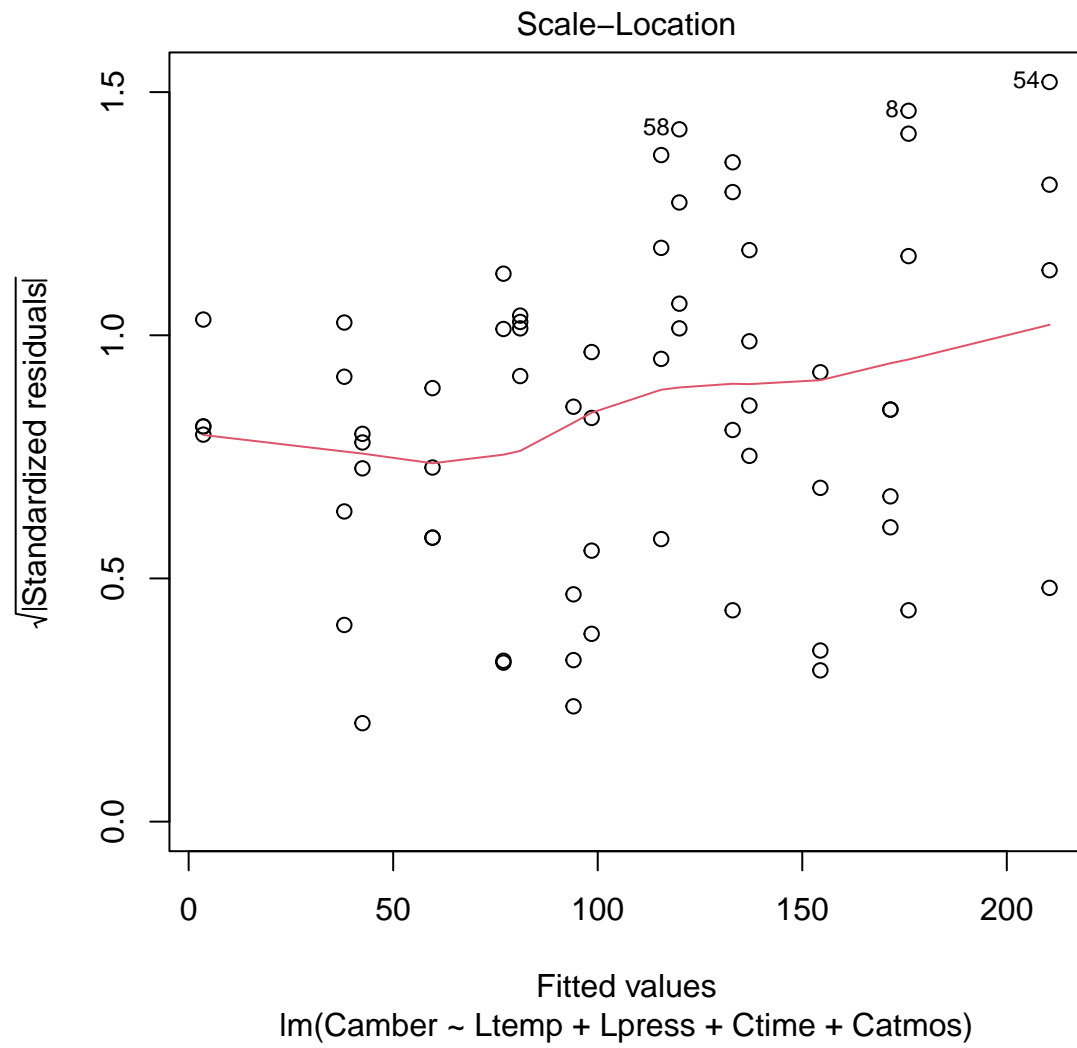
De plus, on plot la régression linéaire comme montré en haut de page suivante. Les résidus ne sont pas uniformément répartis, ce qui confirme l'existence de dépendances qui restent à expliquer.

```
plot(silicium_adj)
```





```
## hat values (leverages) are all = 0.078125
## and there are no factor predictors; no plot no. 5
```

Exercice 02 - Régression logistique

1. Décrire les variables

Le fichier neuralgia.txt décrit un ensemble de 60 patients soignés pour névralgie avec chacun quatre variables explicatives : l'âge (variable à modalités multiples), le sexe, (variable à deux modalités) le traitement suivi (variable à trois modalités) et la durée (variable à modalités multiples). La variable étudiée est Pain et représente la souffrance du patient : 0 s'il ne souffre pas et 1 s'il souffre. On visualise les premières lignes de ce fichier importé dans R ci-dessous

```
neuralgia = read.table(file = "neuralgia.txt", # ./02. Segundo BE/01. Treinamento/neuralgia.txt
                      header = TRUE)
head(neuralgia)
```

```
##   Treatment Sex Age Duration Pain
```

```
## 1      P  F  68      1  0
## 2      B  M  74     16  0
## 3      P  F  67     30  0
## 4      P  M  66     26  1
## 5      B  F  67     28  0
## 6      B  F  77     16  0
```

```
unique(neuralgia$Treatment)
```

```
## [1] "P" "B" "A"
```

```
unique(neuralgia$Sex)
```

```
## [1] "F" "M"
```

2. Partager le fichier en un fichier d'apprentissage (80%) et un fichier de test (20%)

```
n = nrow(neuralgia)
p = n * 0.8
u = sample(1:n,p)
donnees_apprentissage = neuralgia[u,]
donnees_test = neuralgia[-u,]
nrow(donnees_apprentissage)
```

```
## [1] 48
```

```
nrow(donnees_test)
```

```
## [1] 12
```

3. Réaliser sur le fichier d'apprentissage une régression logistique pour prédire la variable Pain

L'évènement modélisé ici est l'évènement « il y a Pain ». Soit $\pi(x)$ la probabilité de l'évènement que l'on cherche à modéliser. Alors le modèle logit consiste à écrire que :

Où sont les paramètres à estimer. La fonction Logit est la fonction définie par :

```
donnees_apprentissage$Treatment = as.factor(donnees_apprentissage$Treatment)
donnees_apprentissage$Sex = as.factor(donnees_apprentissage$Sex)
logistic_model = glm(Pain ~ Treatment + Sex + Age + Duration, family=binomial(link="logit"), data=donnees_apprentissage)
```

4. Analyser le résultat des commandes Anova

```
library(car)
```

```
## Loading required package: carData
```

```
anova(logistic_model, test="Chisq") # test du Chi carré
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Pain
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                47      66.208
## Treatment  2  14.9723      45      51.236 0.0005608 ***
## Sex        1   9.1083      44      42.128 0.0025445 **
## Age        1   9.3657      43      32.762 0.0022108 **
## Duration   1   0.3529      42      32.409 0.5524950
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(logistic_model, test.statistic = "LR", type = 'III') # test de maximum de vraisemblance
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: Pain
##          LR Chisq Df Pr(>Chisq)
## Treatment 18.5223  2 9.505e-05 ***
## Sex        9.2846  1 0.002311 **
## Age        7.0862  1 0.007768 **
## Duration   0.3529  1 0.552495
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(logistic_model, test.statistic = "Wald", type = 'III') # test de Wald-Wolfowitz
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: Pain
##          Df   Chisq Pr(>Chisq)
## (Intercept) 1  5.0178  0.025088 *
## Treatment    2 10.3335  0.005703 **
## Sex          1  6.7053  0.009613 **
## Age          1  4.4415  0.035075 *
## Duration     1  0.3398  0.559920
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Avec le test du Chi carré, la durée du traitement n'a aucune influence sur la douleur du patient. Le traitement employé, le sexe et l'âge du patient ont eux une influence.

Dans le test de maximum de vraisemblance, la durée du traitement n'a toujours pas d'influence et le traitement employé, l'âge et le sexe du patient ont une influence. Notons néanmoins que l'influence estimée de ces trois variables explicatives est différente du test précédent. Ici, le type de traitement appliqué a une plus forte influence ainsi que l'âge, le sexe ayant une influence du même ordre de grandeur.

Pour le Wald-Wolfowitz, la même conclusion sur la non influence de la durée du traitement. De la même manière, les influences estimées du type de traitement, de l'âge et du sexe du patient sont différentes des analyses précédentes.

```
summary(glm(Pain ~.,family=binomial(link="logit"),data=donnees_apprentissage))

##
## Call:
## glm(formula = Pain ~ ., family = binomial(link = "logit"), data = donnees_apprentissage)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.7376  -0.3942  -0.1277   0.4730   2.5452
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -22.32938    9.96824  -2.240  0.02509 *
## TreatmentB   -0.45480    1.25016  -0.364  0.71601
## TreatmentP    3.84195    1.28268   2.995  0.00274 **
## SexM          2.70905    1.04618   2.589  0.00961 **
## Age           0.28407    0.13479   2.107  0.03507 *
## Duration     -0.02613    0.04483  -0.583  0.55992
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 66.208  on 47  degrees of freedom
## Residual deviance: 32.409  on 42  degrees of freedom
## AIC: 44.409
##
## Number of Fisher Scoring iterations: 6
```

A la lecture de ce summary, on conclut encore une fois sur la large non-influence (en rouge) de la durée du traitement sur la douleur ressentie par le patient. Le sexe et l'âge du patient sont tout deux influents. Concernant le traitement, on peut préciser à la lecture du summary que le traitement B n'a aucune influence sur Pain alors que le traitement P en a une grande (en vert).

5. Réaliser maintenant une procédure forward pour le critère AIC

On effectue maintenant une procédure forward pour construire le modèle basé sur les variables les plus significatives. Celui ci fonctionne en ajoutant à chaque itération une variable au modèle. La variable ajoutée sera celle qui minimisera le critère AIC. On arrête les itérations quand toutes les variables auront été ajoutées ou bien quand toutes les variables restantes à ajouter dépassent un certain seuil (5%) du critère. Dans notre cas, on obtient les itérations et le résultat (en vert) affichés ci dessous :

```
logistic_model_2=glm(Pain ~ 1,family=binomial,data=donnees_apprentissage)
next_step <- step(logistic_model_2, direction="forward", scope=list(upper=~(Treatment + Sex + Age + Dur
```

```
## Start: AIC=68.21
## Pain ~ 1
##
##           Df Deviance   AIC
## + Treatment  2   51.236 57.236
## + Sex        1   57.811 61.811
## + Age        1   60.330 64.330
## + Duration   1   63.379 67.379
## <none>       66.208 68.208
##
## Step: AIC=57.24
## Pain ~ Treatment
##
##           Df Deviance   AIC
## + Age      1   41.797 49.797
## + Sex      1   42.128 50.128
## <none>     51.236 57.236
## + Duration 1   49.312 57.312
##
## Step: AIC=49.8
## Pain ~ Treatment + Age
##
##           Df Deviance   AIC
## + Sex      1   32.762 42.762
## <none>     41.797 49.797
## + Duration 1   41.694 51.694
##
## Step: AIC=42.76
## Pain ~ Treatment + Age + Sex
##
##           Df Deviance   AIC
## <none>     32.762 42.762
## + Duration 1   32.409 44.409
```

```
next_step$anova
```

```
##           Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1              NA      NA      47    66.20841 68.20841
## 2 + Treatment -2 14.972255      45    51.23615 57.23615
## 3   + Age    -1  9.439503      44    41.79665 49.79665
## 4   + Sex    -1  9.034557      43    32.76209 42.76209
```

```
Anova(next_step,test.statistic="LR",type = 'III')
```

```
## Analysis of Deviance Table (Type III tests)
##
## Response: Pain
##           LR Chisq Df Pr(>Chisq)
```

```
## Treatment 19.5341 2 5.731e-05 ***
## Age       9.3657 1 0.002211 **
## Sex       9.0346 1 0.002649 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Le modèle obtenu par cette procédure confirme les analyses faites précédemment. Il sélectionne les variables explicatives Traitement, Age et Sex avec Traitement qui a la plus grande influence car sélectionné en premier.

6. A l'aide du fichier test, comparer les matrices de confusions pour les deux modèles

```
logistic_model_reduit = glm(Pain ~ Treatment + Sex + Age, family=binomial(link="logit"), data=donnees_appr
predict = exp(predict(logistic_model, newdata = donnees_test))/(1+exp(predict(logistic_model, newdata=don
predict_reduit = exp(predict(logistic_model_reduit, newdata =donnees_test))/(1+exp(predict(logistic_model
table(predict > 0.5, donnees_test$Pain)
```

```
##
##          0 1
## FALSE 8 1
## TRUE  1 2
```

```
table(predict_reduit > 0.5, donnees_test$Pain)
```

```
##
##          0 1
## FALSE 8 1
## TRUE  1 2
```

Même résultat !!

7. On se fixe un modèle. Etudier la sensibilité des qualités prédictives à l'échantillon

```
calculer_diff_vector <- function(n,p,neuralgia){

  correct_pred = vector("numeric",50)

  for (i in 1:50) {
    u = sample(1:n,p)
    donnees_apprentissage = neuralgia[u,]
    donnees_test = neuralgia[-u,]
    donnees_apprentissage$Treatment = as.factor(donnees_apprentissage$Treatment)
    donnees_apprentissage$Sex = as.factor(donnees_apprentissage$Sex)
    logistic_model = glm(Pain ~ Treatment + Sex + Age + Duration, family=binomial(link="logit"), data=d
    predict = exp(predict(logistic_model, newdata = donnees_test))/(1+exp(predict(logistic_model, newdat
    results = table(predict > 0.5, donnees_test$Pain)
    tryCatch({
```

```

        correct_pred[i] = (results[1,1]+results[2,2])/(n-p)
    },
    error = function(e){
        correct_pred[i] = (results[1,1])/(n-p)
        print('Only 1 class was predicted')
    })
}

return (correct_pred)
}

base_case = calculer_diff_vector(n,n * 0.8,neuralgia)
alt_case_1 = calculer_diff_vector(n,n * 0.9,neuralgia)

## [1] "Only 1 class was predicted"
## [1] "Only 1 class was predicted"
## [1] "Only 1 class was predicted"

alt_case_2 = calculer_diff_vector(n,n * 0.7,neuralgia)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

alt_case_3 = calculer_diff_vector(n,n * 0.6,neuralgia)

## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# aver = c(mean(base_case), mean(alt_case_1), mean(alt_case_2), mean(alt_case_3))
successes = c(mean(alt_case_1), mean(base_case), mean(alt_case_2), mean(alt_case_3))
proportion = c(0.9, 0.8, 0.7, 0.6)
plot(proportion,successes, type="b", col='blue')

```