Régression de Poisson avec R

Tutoriel Tanagra

1 Introduction

Ce tutoriel fait suite au support de cours dédié à la Régression de Poisson [COURS]. Je reprends la trame et les données d'un traitement décrit dans l'ouvrage "Approaching Multivariate Analysis - A Practical Introduction" (2010, chapitre 13) [LIVRE]. Les auteurs effectuent les traitements sous SPSS. J'ai trouvé intéressant de pouvoir reproduire (ou pas) leurs résultats en effectuant l'analyse sous R avec l'outil `glm()` du package "stats", installé et chargé automatiquement au démarrage de R.

Nous ferons constamment référence à ces deux sources bibliographiques (LIVRE et COURS) tout au long de ce document.

2 Données

2.1 Importation des données

Le fichier de données (LIVRE, page 314) retranscrit le nombre de crise d'épilepsies (EVENTS) chez des patients, sur une année consécutivement à la fin de la prise d'un médicament (TREATMENT, 1 : haute dose, 2 : faible dose, 3 : placebo).

L'objectif est d'évaluer l'impact du médicament sur la réduction d'épisodes de crise. Il (cet impact) peut être relativisé par la consommation d'alcool du patient (ALCOHOL, binaire, 1 : oui, 0 : non) et l'estime de soi du patient (ESTEEM, numérique, plus la valeur est élevée, plus la personne est confiante).

Le fichier "**med.poissonregression.equaltimes.sav**" est au format SPSS, nous utilisons la fonction `read.spss()` de package "foreign" qu'il nous faut au préalable installer et charger.

```
#changer de répertoire
setwd("... votre dossier ...")
```

```r
#charger les données
library(foreign)
D <- read.spss("med.poissonregression.equaltimes.sav",to.data.frame=TRUE)

## re-encoding from CP1252

#structure de la base
print(str(D))

## 'data.frame':    75 obs. of  4 variables:
##  $ esteem   : num  13 15 16 15 21 10 18 17 17 16 ...
##  $ alcohol  : num  0 0 0 0 0 1 0 0 1 0 ...
##  $ treatment: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ events   : num  6 5 4 4 3 4 3 3 3 3 ...
##  - attr(*, "variable.labels")= Named chr
##   ..- attr(*, "names")= chr
##  - attr(*, "codepage")= int 1252
## NULL

#premières valeurs
print(head(D))

##   esteem alcohol treatment events
## 1     13       0         1      6
## 2     15       0         1      5
## 3     16       0         1      4
## 4     15       0         1      4
## 5     21       0         1      3
## 6     10       1         1      4
```

2.2 Variable cible EVENTS

La variable cible EVENTS retient notre attention. Nous affichons la distribution de fréquences.

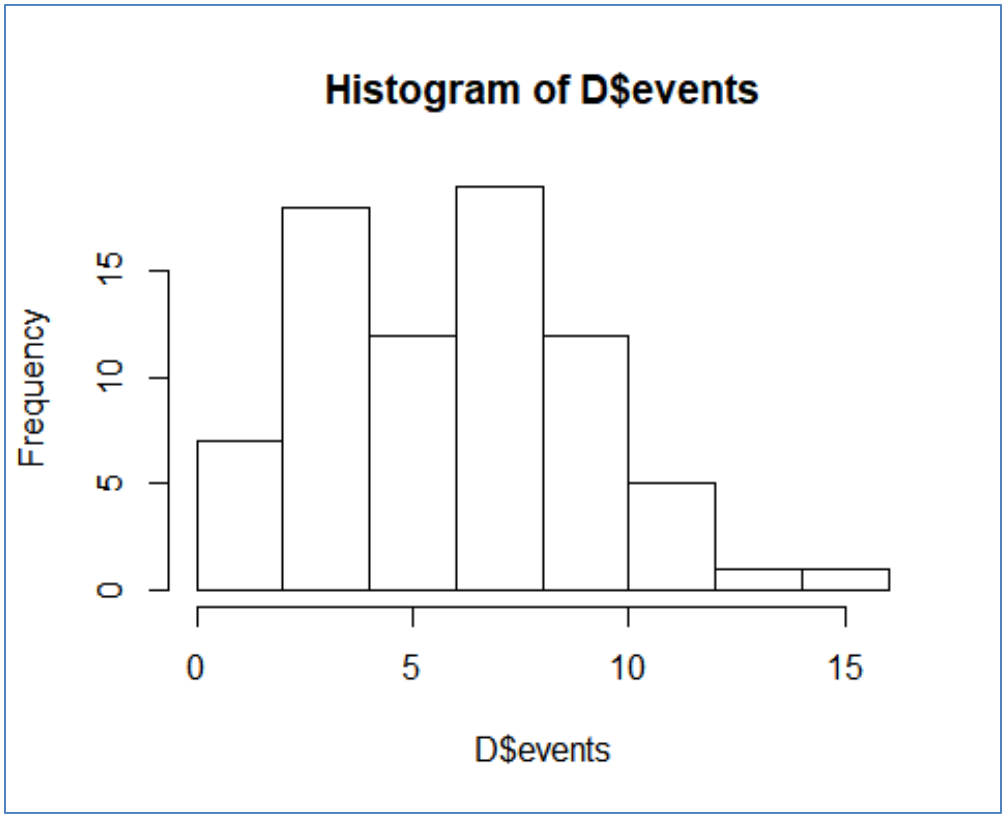
```r
#valeurs possibles evenements
print(table(D$events))

##
##  1  2  3  4  5  6  7  8  9 10 11 12 14 16
##  3  4 12  6  6  6 12  7  9  3  3  2  1  1
```

La valeur min. est 1, son nombre d'occurrence est 3, etc. La valeur 16 apparaît une seule fois.

Nous affichons l'histogramme de distribution.

```r
#histogramme
hist(D$events)
```



La distribution est légèrement asymétrique à droite.

2.3 Lien entre EVENTS et TREATMENT

L'objet de ce tutoriel est de s'appuyer sur la régression pour caractériser l'influence du traitement sur la survenue des crises chez les patients. Mais, avant d'utiliser le canon pour tuer la mouche comme je le dis bien souvent à mes étudiants, essayons déjà d'inspecter la relation via des statistiques descriptives simples.

Tout d'abord, voyons les effectifs des traitements prescrits.

```
#effectifs de treatment
print(table(D$treatment))

##
##  1  2  3
## 20 30 25
```

Ils sont relativement équilibrés. Sur les 75 individus étudiés, 20 ont reçu une forte dose du médicament (treatment = 1), 30 une dose modérée (2), et 25 un placebo (3), sans efficacité pharmacologique a priori.

Voyons maintenant la moyenne de l'apparition des crises selon les traitements.

```
#moyenne events selon treatment
mcond_treat <- tapply(X=D$events,INDEX = D$treatment,FUN = mean)
print(mcond_treat)

##        1        2        3
## 2.950000 6.066667 9.360000
```

Le médicament semble d'autant plus efficace que l'on augmente la dose. Nous garderons bien à l'esprit ce tableau car nous essaierons d'en retrouver les valeurs à travers les résultats de la régression que nous réaliserons par la suite.

Nous pouvons également mettre en lumière cette relation via un tableau de contingence croisant le nombre de crises avec les traitements (LIVRE, page 316).

```
#tableau croisé crises x treatment
print(table(D$events,D$treatment))

##
##      1 2 3
##   1  3 0 0
##   2  3 1 0
##   3  9 3 0
##   4  3 3 0
##   5  1 5 0
##   6  1 3 2
##   7  0 8 4
##   8  0 3 4
##   9  0 4 5
##   10 0 0 3
##   11 0 0 3
##   12 0 0 2
##   14 0 0 1
##   16 0 0 1
```

3 individus ont reçu le traitement 1 (dose forte) et n'ont connu qu'une crise, 3 autres avec le même traitement n'ont connu que deux crises, 9 ont connu 3 crises, etc. La conclusion est la même qu'avec le tableau des moyennes conditionnelles : plus la dose est réduite (allant de 1 à 3), plus les patients sont confrontés à des épisodes épileptiques. Notamment, seules les personnes ayant reçu un placebo ont connu 10 convulsions ou plus.

3 Régression EVENTS vs. TREATMENT

3.1 Codage binaire simple

La variable TREATMENT est qualitative. Nous devons la recoder. Nous optons pour un codage disjonctif simple dans un premier temps, en prenant (TREATMENT = 3, placebo) comme modalité de référence (LIVRE, page 316). De fait, nous modéliserons l'écart par rapport au placebo. On peut discuter de cette transformation car il y a une gradation dans les traitements (high, low, placebo), nous ignorons totalement cette information avec le codage disjonctif adopté ici. Nous y reviendrons plus loin (section 4).

Sous R, nous utilisons les instructions suivantes :

```
#recodage treatment - high
T1 <- rep(0,nrow(D))
T1[D$treatment==1] <- 1
print(sum(T1))

## [1] 20

#recodage treatment - low
T2 <- rep(0,nrow(D))
T2[D$treatment==2] <- 1
print(sum(T2))

## [1] 30
```

Les sommes de contrôles sont cohérentes. Heureusement ! Ces petites vérifications sont indispensables lorsque nous menons une analyse.

3.2 Régression de POISSON : EVENTS vs. TREATMENT

Nous pouvons lancer la régression avec la fonction `glm()`. L'option (family = "poisson") est primordiale pour préciser que la cible EVENTS est une variable de comptage :

```
#variable cible
y <- D$events

#régression
m1 <- glm(y ~ T1 + T2,family="poisson")
print(m1)

##
## Call:  glm(formula = y ~ T1 + T2, family = "poisson")
##
## Coefficients:
```

```
## (Intercept)           T1           T2
##      2.2364      -1.1546      -0.4336
##
## Degrees of Freedom: 74 Total (i.e. Null);  72 Residual
## Null Deviance:       122.8
## Residual Deviance: 45.84     AIC: 319.4
```

Le modèle s'écrit :

$$ln\ y = 2.2364 - 1.1546 \times T1 - 0.4336 \times T2$$

R ajoute une série d'indicateurs que nous allons expertiser maintenant. Mais auparavant, affichons les propriétés associées à l'objet généré par `glm()`.

```
#objet m1
print(attributes(m1))

## $names
##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "call"              "formula"           "terms"
## [25] "data"              "offset"            "control"
## [28] "method"            "contrasts"         "xlevels"
##
## $class
## [1] "glm" "lm"
```

Nous disposons d'une description détaillée de ces champs dans la documentation.

3.3 Qualité de l'ajustement

Nous récupérons tout d'abord les valeurs ajustées ($\hat{\lambda}_i$) par le modèle (COURS, page 10).

```
#valeurs ajustées de l'endogène
lambda1 <- m1$fitted.values
print(lambda1)

##        1        2        3        4        5        6        7        8
## 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000
##        9       10       11       12       13       14       15       16
## 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000 2.950000
##       17       18       19       20       21       22       23       24
## 2.950000 2.950000 2.950000 2.950000 6.066667 6.066667 6.066667 6.066667
##       25       26       27       28       29       30       31       32
## 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667
##       33       34       35       36       37       38       39       40
```

```
## 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667
##       41       42       43       44       45       46       47       48
## 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667 6.066667
##       49       50       51       52       53       54       55       56
## 6.066667 6.066667 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000
##       57       58       59       60       61       62       63       64
## 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000
##       65       66       67       68       69       70       71       72
## 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000 9.360000
##       73       74       75
## 9.360000 9.360000 9.360000
```

Nous les utilisons pour calculer la statistique déviance (COURS, page 13).

```
#Statistique deviance
Stat_D <- 2 * sum(ifelse(y>0,y*log(y/lambda1),0)-(y-lambda1))
print(Stat_D)

## [1] 45.84243
```

C'est la statistique déviance que R appelle "*Residual deviance*" lors de l'affichage des résultats ci-dessus. Pour apprécier la qualité de l'ajustement, nous en calculons la probabilité critique avec la distribution du KHI-2. Le degré de liberté est égal à (`n - p – 1`) (`n` nombre d'observations, `p` nombre de variables dans le modèle), il est représenté par la propriété `$df.residual` dans notre calcul :

```
#p-value
print(pchisq(Stat_D,m1$df.residual,lower.tail = FALSE))

## [1] 0.9930895
```

Au risque 5%, les valeurs prédites ($\hat{\lambda}_i$) ne s'écartent pas significativement des valeurs observées (y_i). Le modèle est bien ajusté (COURS, page 13).

Nous pouvons réitérer l'opération avec la statistique de Pearson (COURS, page 12). La conclusion est identique.

```
#statistique khi2
Stat_khi2 <- sum((y-lambda1)^2/lambda1)
print(Stat_khi2)

## [1] 44.73573

#p-value
print(pchisq(Stat_khi2,m1$df.residual,lower.tail = FALSE))

## [1] 0.9951757
```

3.4 Pseudo-R2

Le R^2 ne s'interprète pas comme la proportion de variance expliquée pour la régression de Poisson, mais plutôt comme le degré d'amélioration par rapport au pire modèle représenté par la régression avec la seule constante (null model). Il résulte de la confrontation entre leurs déviances respectives. L'objet régression "**m1**" fournit les informations adéquates, l'opération est facile à réaliser sous R.

```
#Pseudo-R2
R2_1 <- 1.0 - m1$deviance / m1$null.deviance
print(R2_1)

## [1] 0.626809
```

Par rapport au modèle trivial, nous avons une réduction de 62% de la déviance.

3.5 Test de significativité globale du modèle

Le test du rapport de vraisemblance pour la significativité globale du modèle résulte également de l'opposition de ces mêmes déviances. L'écart…

```
#test du rapport de vraisemblance
LR1 <- m1$null.deviance - m1$deviance
print(LR1)

## [1] 76.99664
```

… suit une loi du KHI2 à p (nombre de variables du modèle) sous H0 (leurs coefficients sont tous nuls, $a_1 = a_2 = \ldots = a_p = 0$, le test n'inclut pas la constante) (COURS, page 16).

```
#p-value -- ddl = 2, T1 et T2
print(pchisq(LR1,2,lower.tail = FALSE))

## [1] 1.907184e-17
```

Le modèle est (largement) globalement significatif à 5%.

3.6 Test de significativité individuelle des coefficients

Nous nous appuyons sur le test de Wald pour tester la significativité individuelle des coefficients (COURS, page 17). La fonction `summary()` fournit les informations idoines.

```
#évaluation des coefficients
sm1 <- summary(m1)
print(sm1)
```

```
##
## Call:
## glm(formula = y ~ T1 + T2, family = "poisson")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.92216  -0.52184   0.02903   0.44562   1.96891
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.23645    0.06537  34.211  < 2e-16 ***
## T1          -1.15464    0.14568  -7.926 2.27e-15 ***
## T2          -0.43364    0.09883  -4.388 1.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 122.839  on 74  degrees of freedom
## Residual deviance:  45.842  on 72  degrees of freedom
## AIC: 319.37
##
## Number of Fisher Scoring iterations: 4
```

La statistique de test (**z-value**) est formée par le ratio entre le coefficient estimé (Estimate) et son écart-type (Std. Error). Elle est asymptotiquement gaussienne, nous constatons que les coefficients sont tous significativement différents de zéro au risque (α = 5%) dans notre régression puisque les probabilités critiques (p-value) [Pr(>|z|)] sont inférieures à 0.05.

Un petit commentaire sur le facteur de surdispersion (COURS, pages 31 et 32), le rapport entre la déviance et les degrés de liberté. Il est sensiblement inférieur à 1 ici (`45.842 / 72 = 0.63669…`) c.-à-d. indiquant une sous-dispersion. Les auteurs ne s'en émeuvent pas outre mesure : "*Here we can be satisfied that there is no evidence that the Poisson regression model is a poor fit*" (LIVRE, page 319).

Pour ma part, je dirais que les tests individuels des coefficients sont conservateurs dans ce cas (favorisent H0, le coefficient est nul). La significativité constatée n'en est que plus probante. Introduire une correction (une régression "quasi-poisson" par exemple, COURS, page 32) ne modifiera pas les coefficients estimés et réduira simplement les écarts-types, augmentant mécaniquement les "z-value" en valeur absolue.

3.7 Interprétation des coefficients

L'interprétabilité des résultats est un des atouts forts de la régression. Pour les modèles de comptage, les coefficients sont en relation directe avec les moyennes conditionnelles de la variable réponse lorsque les explicatives sont binaires. Affichons pour rappel les moyennes de EVENTS en fonction de TREATMENT (1 : haute dose, 2 : faible dose, 3 : placebo).

```
#moyennes conditionnelles de la cible
print(mcond_treat)

##        1        2        3
## 2.950000 6.066667 9.360000
```

De quelle manière ces valeurs sont-elles liées aux coefficients de la régression ?

Lecture de la constante. Prenons l'exponentielle de la constante de la régression.

```
#lecture de la constante - exponentielle
print(exp(m1$coefficients[1]))

## (Intercept)
##        9.36
```

Elle correspond à la moyenne de EVENTS pour la modalité de référence – celle qui a été omise pour la création des indicatrices – de l'explicative.

Coefficient de T1. Faisons de même pour le coefficient de T1 (TREATMENT = 1).

```
#coefficient de T1
print(exp(m1$coefficients[2]))

##        T1
## 0.3151709
```

Son exponentielle exprime le ratio entre sa moyenne conditionnelle et celle de la modalité de référence (TREATMENT = placebo) (COURS, page 20). Nous vérifions cela aisément en faisant le produit :

```
#moyenne conditionnelle de TRAITMENT = 1
print(exp(m1$coefficients[2])*mcond_treat[3])

##   T1
## 2.95
```

Coefficient de T2. Nous avons le même mécanisme pour T2.

```
#coefficient de T2
print(exp(m1$coefficients[3]))

##        T2
## 0.6481481

#vérification moyenne conditionnelle
print(exp(m1$coefficients[3])*mcond_treat[3])

##       T2
## 6.066667
```

Ainsi, tester la nullité d'un coefficient dans ce contexte revient à tester l'écart entre les logarithmes des moyennes conditionnelles (H0 : écart = 0 avec la moyenne de la modalité de référence) ou, de manière équivalente, tester le ratio entre ces moyennes (H0 : ratio = 1).

3.8 Intervalle de confiance des coefficients

Puisque nous disposons de la distribution des coefficients (normalité asymptotique) et des écarts-type, nous pouvons calculer leurs intervalles de confiance (COURS, page 18). La commande `confint.default()` fournit les bornes basses et hautes pour un niveau de confiance donné.

```
#intervalle de confiance des coefficients à 90%
print(confint.default(m1,level=0.9))

##                     5 %       95 %
## (Intercept)  2.1289179  2.3439727
## T1          -1.3942617 -0.9150185
## T2          -0.5962021 -0.2710698
```

4 Régression avec un codage différent de TREATMENT

Le codage disjonctif adopté précédemment ne rend pas compte de la progressivité du traitement (TREATMENT = 3, rien ; 2 : faible dose ; 1 haute dose). Dans cette section, nous utilisons un codage imbriqué pour traduire cette information qui peut être importante pour analyser l'efficacité graduelle du médicament (le lien avec la posologie).

Voici les nouvelles versions des indicatrices (voir le principe dans "Pratique de la Régression Logistique (version 2.0)", section 5.2.4 "Lecture et interprétation des coefficients - Variable explicative qualitative ordinale") :

```r
#s'appuyer sur un codage différent de treatment
#low
TC1 <- rep(0,nrow(D))
TC1[D$treatment<=1] <- 1
print(sum(TC1))

## [1] 20

#high
TC2 <- rep(0,nrow(D))
TC2[D$treatment<=2] <- 1
print(sum(TC2))

## [1] 50
```

Nous réalisons la régression avec ces nouvelles indicatrices.

```r
#régression
m2 <- glm(y ~ TC1 + TC2,family="poisson")
print(summary(m2))

##
## Call:
## glm(formula = y ~ TC1 + TC2, family = "poisson")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.92216  -0.52184   0.02903   0.44562   1.96891
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.23645    0.06537  34.211  < 2e-16 ***
## TC1         -0.72100    0.14981  -4.813 1.49e-06 ***
## TC2         -0.43364    0.09883  -4.388 1.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 122.839  on 74  degrees of freedom
## Residual deviance:  45.842  on 72  degrees of freedom
## AIC: 319.37
##
## Number of Fisher Scoring iterations: 4
```

Première information importante, la qualité globale de la régression est maintenue (Residual Deviance). La modification du codage change la lecture des coefficients mais ne déteint pas sur la qualité de l'ajustement.

Pour ce qui est des coefficients, celui TC2 est identique à T2, celui de TC1 est différent en revanche. Son exponentielle exprime le ratio de la moyenne conditionnelle de (TREATMENT = 1) *par rapport à la modalité précédente* (TREATMENT = 2.)

```
#coefficient de TC1
print(exp(m2$coefficients[2]))

##       TC1
## 0.4862637

#vérification par rapport mcond_treat[2]
print(exp(m2$coefficients[2])*mcond_treat[2])

##  TC1
## 2.95
```

Par rapport à ceux qui ont pris une faible dose du médicament (TREATMENT = 2), le nombre moyen de crises est (`1/0.4862637 =`) `2.06` fois plus faible chez les personnes qui ont reçu une plus forte dose (TREATMENT = 1). Conclusion : augmenter la dose améliore manifestement la rémission des patients puisque le coefficient est significatif.

5 Introduction des covariables dans la régression

5.1 Régression avec les explicatives disponibles

L'impact du traitement sur les crises peut être atténué ou amplifié par les caractéristiques des patients. Nous introduisons dans la régression la consommation d'alcool (ALCOHOL, 1 : oui, 0 : non) et l'estime de soi (ESTEEM).

```
#ajout de covariables
m3 <- glm(y ~ TC1 + TC2 + alcohol + esteem, data = D, family="poisson")
print(m3)

##
## Call:  glm(formula = y ~ TC1 + TC2 + alcohol + esteem, family = "poisson",
##     data = D)
##
## Coefficients:
## (Intercept)          TC1          TC2      alcohol       esteem
##     2.74912     -0.69576     -0.45741      0.00426     -0.03385
##
## Degrees of Freedom: 74 Total (i.e. Null);  70 Residual
## Null Deviance:       122.8
## Residual Deviance: 39.21     AIC: 316.7
```

```r
#résultats détaillés
sm3 <- summary(m3)
print(sm3)

##
## Call:
## glm(formula = y ~ TC1 + TC2 + alcohol + esteem, family = "poisson",
##     data = D)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.53829  -0.45949  -0.05642   0.41094   1.60403
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.74912    0.22836  12.038  < 2e-16 ***
## TC1         -0.69576    0.15038  -4.626 3.72e-06 ***
## TC2         -0.45741    0.09939  -4.602 4.19e-06 ***
## alcohol      0.00426    0.10315   0.041   0.9671
## esteem      -0.03385    0.01387  -2.441   0.0147 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 122.839  on 74  degrees of freedom
## Residual deviance:  39.214  on 70  degrees of freedom
## AIC: 316.74
##
## Number of Fisher Scoring iterations: 4
```

Nous retrouvons les résultats de SPSS (LIVRE, page 319). C'est toujours rassurant.

Pour ce qui est des variables additionnelles introduites dans le modèle :

- ALCOHOL ne semble pas peser sur les crises d'épilepsies.

- ESTEEM si, de manière négative : plus le patient a une forte estime de lui-même, moins le nombre de crises est élevé.

5.2 Sélection de variables

Retirer les variables explicatives non-pertinentes du modèle est toujours une bonne chose, ne serait-ce que pour leur lisibilité Dans ce section, nous décidons de réaliser une sélection automatique avec la commande `stepAIC()` du package "MASS" qui est installé avec R, mais que nous devons charger explicitement.

Nous optons pour la stratégie (direction = backward), le processus retire individuellement les variables tant que le critère BIC (k = log(n)) diminue (COURS, pages 24 et 25).

```r
#sélection de variables
library(MASS)
m4 <- stepAIC(m3,direction = "backward", k = log(nrow(D)))
```

```
## Start:  AIC=328.33
## y ~ TC1 + TC2 + alcohol + esteem
##
##           Df Deviance    AIC
## - alcohol  1   39.216 324.01
## <none>         39.214 328.33
## - esteem   1   45.212 330.01
## - TC2      1   60.608 345.41
## - TC1      1   62.941 347.74
##
## Step:  AIC=324.01
## y ~ TC1 + TC2 + esteem
##
##          Df Deviance    AIC
## <none>        39.216 324.01
## - esteem  1   45.842 326.32
## - TC2     1   60.657 341.14
## - TC1     1   63.032 343.51
```

ALCOHOL a été retiré. Les variables TC1, TC2 et ESTEEM sont conservées dans le modèle final.

```r
#modèle final
sm4 <- summary(m4)
print(sm4)
```

```
##
## Call:
## glm(formula = y ~ TC1 + TC2 + esteem, family = "poisson", data = D)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.53951  -0.46200  -0.04932   0.41537   1.59781
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.75299    0.20824  13.220  < 2e-16 ***
## TC1         -0.69607    0.15020  -4.634 3.58e-06 ***
## TC2         -0.45718    0.09924  -4.607 4.09e-06 ***
## esteem      -0.03403    0.01322  -2.575     0.01 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 122.839  on 74  degrees of freedom
## Residual deviance:  39.216  on 71  degrees of freedom
## AIC: 314.74
##
## Number of Fisher Scoring iterations: 4
```

Le coefficient de ESTEEM est significatif à 5%.

Pour évaluer sa contribution, nous pourrions également effectuer un test du rapport de vraisemblance confrontant les déviances des modèles (EVENTS ~ TC1 + TC2) et (EVENTS ~ TC1 + TC2 + ESTEEM). Sous l'hypothèse nulle d'absence de contribution, la statistique de test suit une loi du KHI-2 à 1 degré de liberté (COURS, page 16, q = 1). Voyons ce qu'il en est.

```
#Rapport de vraisemblance
LR_ESTEEM = m2$deviance - m4$deviance
print(LR_ESTEEM)

## [1] 6.626253

#probabilité critique (p-value)
print(pchisq(LR_ESTEEM,df=1,lower.tail = FALSE))

## [1] 0.01004864
```

Oui, au risque 5%, ESTEEM contribue significativement dans le modèle.

5.3 Résidus déviance

La fonction `residuals()` fournit les résidus déviance (COURS, page 27).

```
#résidus déviance
rd4 <- residuals(m4)
print(rd4)

##           1           2           3           4           5           6
##  1.40555510  1.07030541  0.62735249  0.56589179  0.35695954  0.24837037
##           7           8           9          10          11          12
##  0.18933316  0.13220710  0.13220710  0.07443788  0.01601449 -0.10283988
...
## -0.37326302 -0.37326302 -0.43653702 -0.53213718 -0.62886315 -0.68015713
##          73          74          75
## -0.72131349 -0.92599212 -1.39113675
```

La somme des carrés de ces valeurs correspond à la statistique déviance (COURS, page 28).

```
#et on a bien la stat déviance si on somme le carré
print(sum(rd4^2))

## [1] 39.21618
```

5.4 Levier

Le levier indique la contribution globale d'un point dans la régression, d'une certaine manière il caractérise également le degré d'éloignement d'un point par rapport aux autres dans l'espace de représentation (COURS, page 29).

Il fait partie des informations produites par la commande `influence.measures()`.

```
#levier
h <- influence.measures(m4)$infmat[,'hat']
print(h)

##          1          2          3          4          5          6
## 0.05636203 0.05038331 0.04910572 0.05038331 0.05584202 0.07568842
##          7          8          9         10         11         12
## 0.04944987 0.04882720 0.04882720 0.04910572 0.05038331 0.05636203
##         13         14         15         16         17         18
## 0.06129501 0.06769161 0.04944987 0.04944987 0.04910572 0.05088240
##         19         20         21         22         23         24
## 0.05276473 0.05636203 0.03464347 0.04884469 0.08527525 0.03688919
##         25         26         27         28         29         30
## 0.03688919 0.05638275 0.03329984 0.09366616 0.03397641 0.04259778
##         31         32         33         34         35         36
## 0.07577103 0.03688919 0.04226935 0.03464347 0.05036402 0.05036402
##         37         38         39         40         41         42
## 0.07577103 0.03688919 0.03464347 0.03329984 0.03329984 0.03397641
##         43         44         45         46         47         48
## 0.04259778 0.06143727 0.03329984 0.03688919 0.04259778 0.03780554
##         49         50         51         52         53         54
## 0.03780554 0.05638275 0.06053500 0.09595681 0.07567310 0.04984646
##         55         56         57         58         59         60
## 0.04024941 0.05006523 0.08173691 0.06919276 0.04061924 0.04061924
##         61         62         63         64         65         66
## 0.09568644 0.18967433 0.06053500 0.05006523 0.04320892 0.04320892
##         67         68         69         70         71         72
## 0.04061924 0.04061924 0.05855345 0.05006523 0.04399257 0.04984646
##         73         74         75
## 0.06919276 0.04320892 0.04984646
```

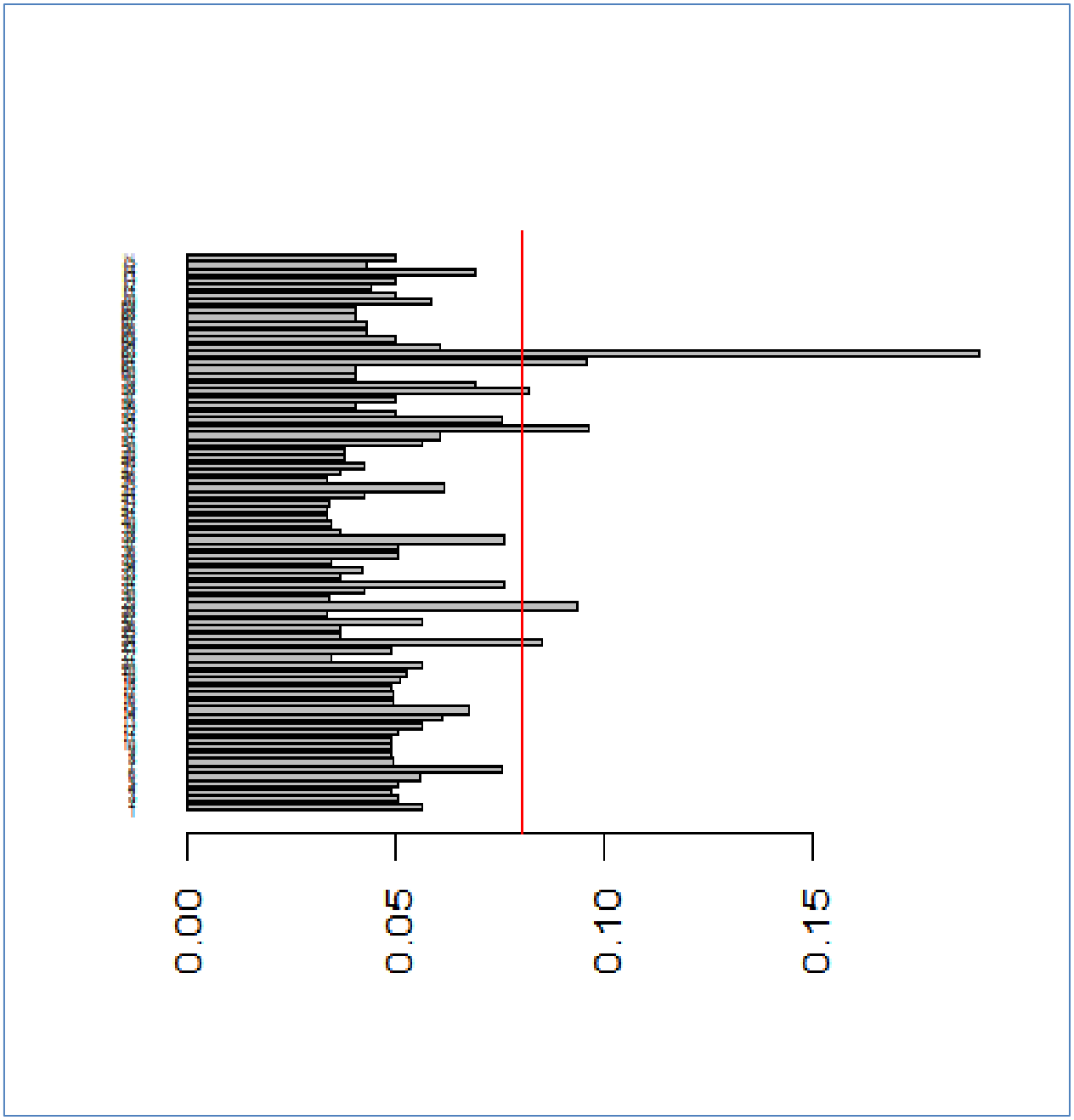
La somme des leviers est égale au nombre de paramètres estimés (nombre de coefficients du modèle). Une règle empirique de détection des points atypiques consiste à identifier les observations qui présentent un levier supérieur à 2 fois la moyenne des valeurs (2 x nombre de paramètres / nombre d'observations).

```
#seuil levier
seuil_h <- 2*(length(m2$coefficients))/nrow(D)
print(seuil_h)

## [1] 0.08
```

Ainsi, nous pouvons distinguer les points potentiellement à problème.

```
#identification de points à problème
par(las=2)
barplot(h,cex.names = 0.35,horiz=TRUE)
abline(v=seuil_h,col='red')
```



L'individu n°62 est visiblement singulier (*ce n'est pas très visible ici, mais il s'agit bien de l'observation n°62*). Voyons sa description :

```
#qui est le point n°62 ?
print(D[62,c('esteem','treatment','events')])

##    esteem treatment events
## 62      7         3     12
```

Il a une faible estime de soi (ESTEEM varie entre 7 et 23), on lui a donné un placebo (TREATMENT = 3), ce qui se ressent au niveau du nombre de crises (rappelons que EVENTS varie entre 1 et 16) durant l'année suivant son (faux) traitement.

5.5 Résidus standardisés

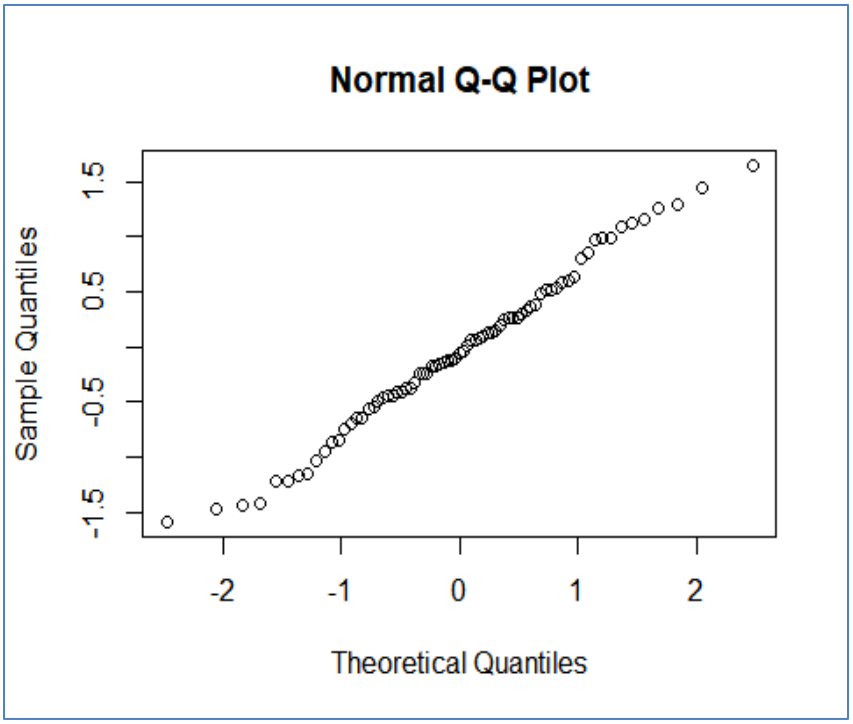
Le fait que le point n°62 ait un levier fort par rapport aux autres points ne veut pas dire qu'il est problématique et qu'il faudrait le retirer automatiquement de la régression. Voyons la qualité de sa modélisation en calculant le résidu standardisé [que l'on pouvait également obtenir avec le commande `rstandard()`] :

```
#résidus standardisé
rs4 <- rd4/sqrt(1-h)
print(rs4)

##           1           2           3           4           5           6
##  1.44692217  1.09833178  0.64334734  0.58070990  0.36736406  0.25833936
##           7           8           9          10          11          12
##  0.19419550  0.13555798  0.13555798  0.07633574  0.01643384 -0.10586658
##          13          14          15          16          17          18
## -0.16854095 -0.23245366 -0.44861912 -0.44861912 -0.55897235 -1.16209080
...
##          61          62          63          64          65          66
## -0.05186467 -0.11564723 -0.13829542 -0.17906391 -0.24678038 -0.24678038
##          67          68          69          70          71          72
## -0.38108290 -0.38108290 -0.44990752 -0.54597997 -0.64316960 -0.69777010
##          73          74          75
## -0.74764284 -0.94667026 -1.42716087
```

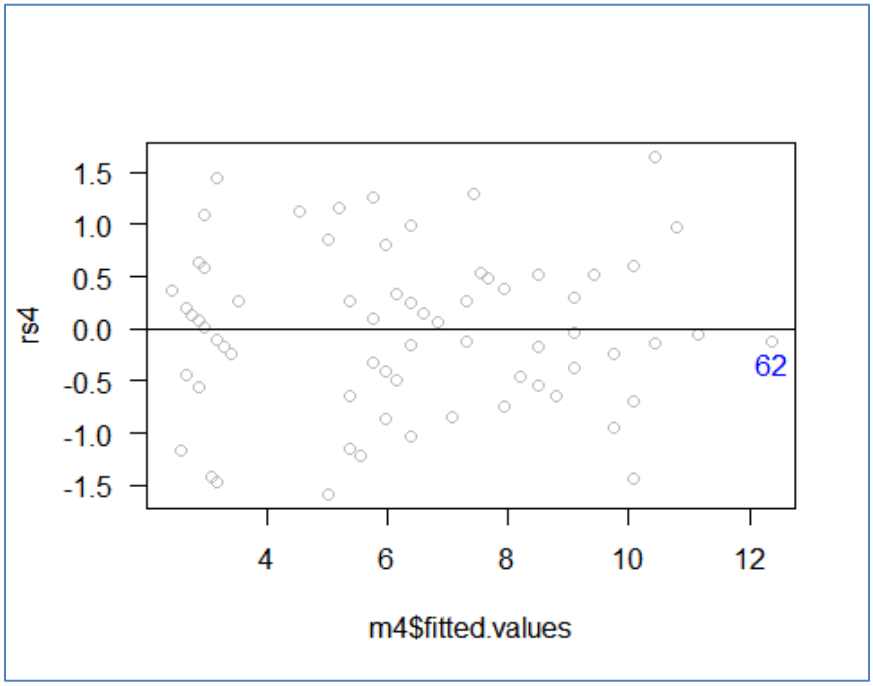
Le résidu standardisé est asymptotiquement gaussien, nous le vérifions avec le graphique `qqnorm()`. Les points forment approximativement une droite, c'est très bon signe pour la modélisation (LIVRE, page 320, les auteurs recourent plutôt à un histogramme de fréquences, approchée avec une courbe de densité normale, la conclusion est la même).

```
#quantile quantile plot – confrontation à la loi normale
qqnorm(rs4)
```



En les croisant avec les valeurs prédites par le modèle ($\hat{\lambda}_i$)…

```
#graphique avec le point n°62
par(las=1)
plot(m4$fitted.values,rs4,col='gray')
abline(h=0)
text(m4$fitted.values[62],rs4[62],labels = '62', col='blue',adj=c(0.5,1.5))
```



… on se rend compte que le point n°62 est bien approximé par la régression.

Plutôt qu'atypique, il serait plutôt emblématique du modèle. Donner un morceau de sucre a un gars qui est déjà psychologiquement au trente-sixième dessous n'aide pas vraiment à le soulager de sa maladie. Le modèle est en accord avec ce constat.

6 Conclusion

La régression de Poisson est une technique prédictive qui permet de modéliser une variable de dénombrement. Dans ce tutoriel, nous montrons sa mise en œuvre sous R sur un exemple réaliste tiré d'un ouvrage (Dugard et al., 2010) où l'analyse avait été menée avec le logiciel SPSS. Sans surprise, nous avons d'une part retrouvé les résultats à l'identique, d'autre part, nous avons pu étendre l'étude en expérimentant des variantes telles qu'un codage tenant compte du caractère ordinal du facteur explicatif TREATMENT.

7 Références

- **[LIVRE]** P. Dugard, J. Todman, H. Staines, "Approaching Multivariate Analysis - A Practical Introduction", Second Edition, Routledge, 2010.
- **[COURS]** R. Rakotomalala, "Régression de Poisson - Diapos", mai 2019.