Le Modèle linéaire généralisé (glm)

Ana Fermin

Monday, March 02, 2015

Introduction : Nous souhaitons expliquer une variable Y , par une variable explicative X (ou plusieurs variables explicatives $X_1, X_2, \ldots, X_p$) lorsque Y est 0 (échec) ou 1 (succès).

Exemples :

- Banque : Y vaut 1 si l'emprunteur est bon payeur, 0 sinon. La variable X donne par exemple l'âge, la profession, le statut matrimonial, le fait d'être ou non propriétaire.
- Assurance : Y vaut 1 si l'assuré est *bon conducteur*(pas de sinistre dans l'année), 0 sinon. La variable X donne par exemple l'âge, le sexe, le degré de bonus-malus de l'année précédente, la vétustédu véhicule, le code postal.

Nous cherchos à modeliser $\mathbb{P}(Y = 1|X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p)$. Nous abordons ici différents modèles (selon le niveau d'interaction considéré entre les variables explicatives quantitatives ou qualitatives) qui appartiennent à la famille des modèles linéaires généralisés. Nous definissons ici le contexte pratique du modèle logistique avec le logiciel R. Nous presentons plusieurs exemples. Le premier exemple concerne les données de cancer de la prostate et le deuxième exemple concerne les données de default de credit banquaire.

Notons $\pi(x) = \mathbb{P}(Y = 1|X_1 = x_1, \ldots, X_p = x_p)$ et supposons que

$$\pi(x) = F(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p),$$

où F est une fonction de répartition inversible donnée et sa reciproque (inverse) est

$$F^{-1}(\pi(x)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

avec $\beta_0, \beta_1, \ldots, \beta_p$ inconnus.

- **Modèle logit** : $F(t) = \frac{\exp(t)}{1+\exp(t)}$ et $F^{-1}(t) = \log(\frac{t}{1-t})$
- **Modèle Probit** : $F(t) = \frac{1}{2\pi}\int_{-\infty}^{t} \exp(-x^2/2)dx$

Modèle Logistique

Cas simple (une seule variable explicative)

Dans ce cas, l'equation de la fonction logistique est

$$\pi(x) = \exp(\beta_0 + \beta_1 x)/(1 + \exp(\beta_0 + \beta_1 x))$$

Cas Multiple (plusieurs variables explicatives)

Dans ce cas, l'equation de la fonction logistique est

$$\pi(x) = \exp(\beta_0 + \beta_1 x + \beta_2 x + \ldots + \beta_p x)/(1 + \exp(\beta_0 + \beta_1 x + \beta_2 x + \ldots + \beta_p x))$$

Modèle Logistique Simple

Variable explicative quantitative

Considérons le cas d'une seule variable explicative quantitative X.

Exemple 1 : On a relevé l'âge et la présence(1) ou l'absence (0) d'une maladie cardiovasculaire chez 100 individus. Les données sont stockées dans le fichier `maladie_cardiovasculaire.txt`: sur une ligne donnée, la variable `AGE` fournit l'âge d'un individu tandis que la variable `CHD` prend la valeur 1 en cas de présence d'une maladie cardiovasculaire chez cet individu et la valeur 0 sinon. Les variables `ID` et `AGRP` donnent respectivement le numéro d'un individu et sa classe d'âge. On acquière les données et on en visualise les premières lignes grâce aux commandes

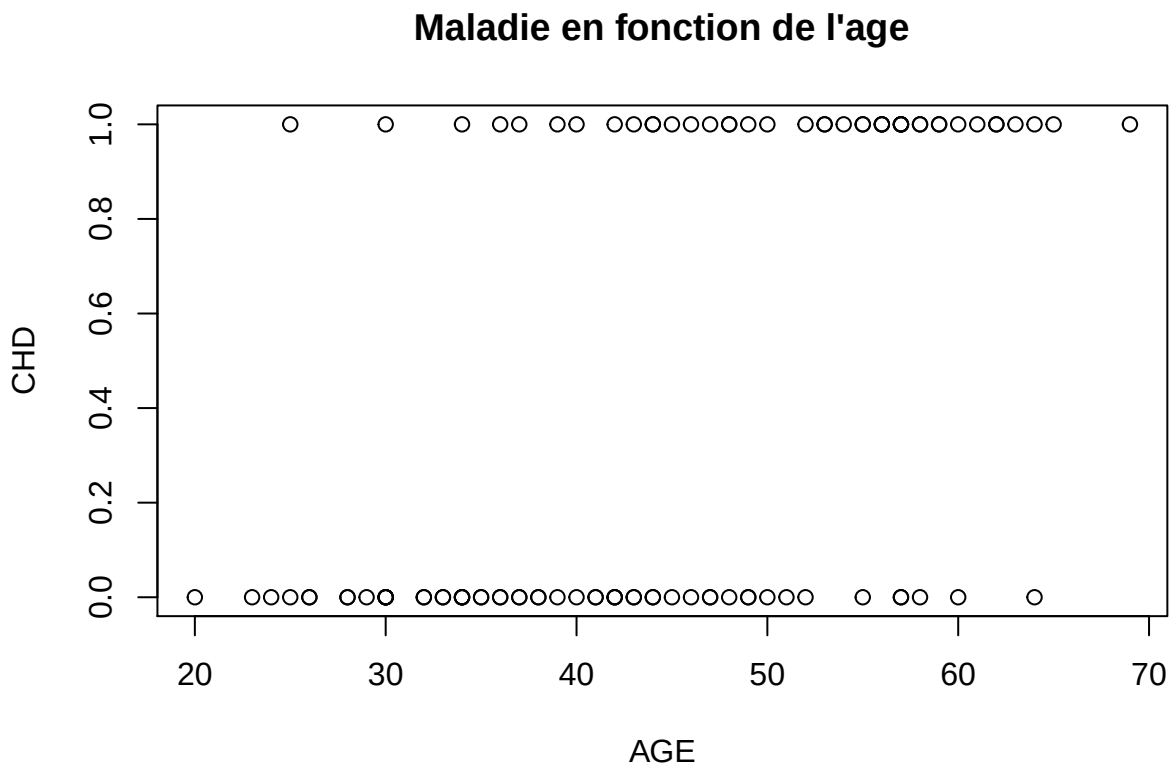
```
#Données (échantillon)
maladie= read.table('maladie_cardiovasculaire.txt',header=TRUE)
attach(maladie)
```

```
#Affichage des  observations (reagroupées en classe d'âge)
cbind(maladie [1:25,], maladie[26:50,], maladie[51:75,],maladie[76:100,])
```

```
##    ID AGRP AGE CHD ID AGRP AGE CHD ID AGRP AGE CHD  ID AGRP AGE CHD
## 1   1    1  20   0 26    3  35   0 51    4  44   1  76    7  55   1
## 2   2    1  23   0 27    3  35   0 52    4  44   1  77    7  56   1
## 3   3    1  24   0 28    3  36   0 53    5  45   0  78    7  56   1
## 4   4    1  25   0 29    3  36   1 54    5  45   1  79    7  56   1
## 5   5    1  25   1 30    3  36   0 55    5  46   0  80    7  57   0
## 6   6    1  26   0 31    3  37   0 56    5  46   1  81    7  57   0
## 7   7    1  26   0 32    3  37   1 57    5  47   0  82    7  57   1
## 8   8    1  28   0 33    3  37   0 58    5  47   0  83    7  57   1
## 9   9    1  28   0 34    3  38   0 59    5  47   1  84    7  57   1
## 10 10    1  29   0 35    3  38   0 60    5  48   0  85    7  57   1
## 11 11    2  30   0 36    3  39   0 61    5  48   1  86    7  58   0
## 12 12    2  30   0 37    3  39   1 62    5  48   1  87    7  58   1
## 13 13    2  30   0 38    4  40   0 63    5  49   0  88    7  58   1
## 14 14    2  30   0 39    4  40   1 64    5  49   0  89    7  59   1
## 15 15    2  30   0 40    4  41   0 65    5  49   1  90    7  59   1
## 16 16    2  30   1 41    4  41   0 66    6  50   0  91    8  60   0
## 17 17    2  32   0 42    4  42   0 67    6  50   1  92    8  60   1
## 18 18    2  32   0 43    4  42   0 68    6  51   0  93    8  61   1
## 19 19    2  33   0 44    4  42   0 69    6  52   0  94    8  62   1
## 20 20    2  33   0 45    4  42   1 70    6  52   1  95    8  62   1
## 21 21    2  34   0 46    4  43   0 71    6  53   1  96    8  63   1
## 22 22    2  34   0 47    4  43   0 72    6  53   1  97    8  64   0
## 23 23    2  34   1 48    4  43   1 73    6  54   1  98    8  64   1
## 24 24    2  34   0 49    4  44   0 74    7  55   0  99    8  65   1
## 25 25    2  34   0 50    4  44   0 75    7  55   1 100    8  69   1
```

On souhaite étudier la relation entre `CHD` et la variable explicative `AGE`. Les données sont représentées à l'aide d'un nuage de points, qui a l'allure suivante :

```
plot(CHD~AGE,main="Maladie en fonction de l'age")
```



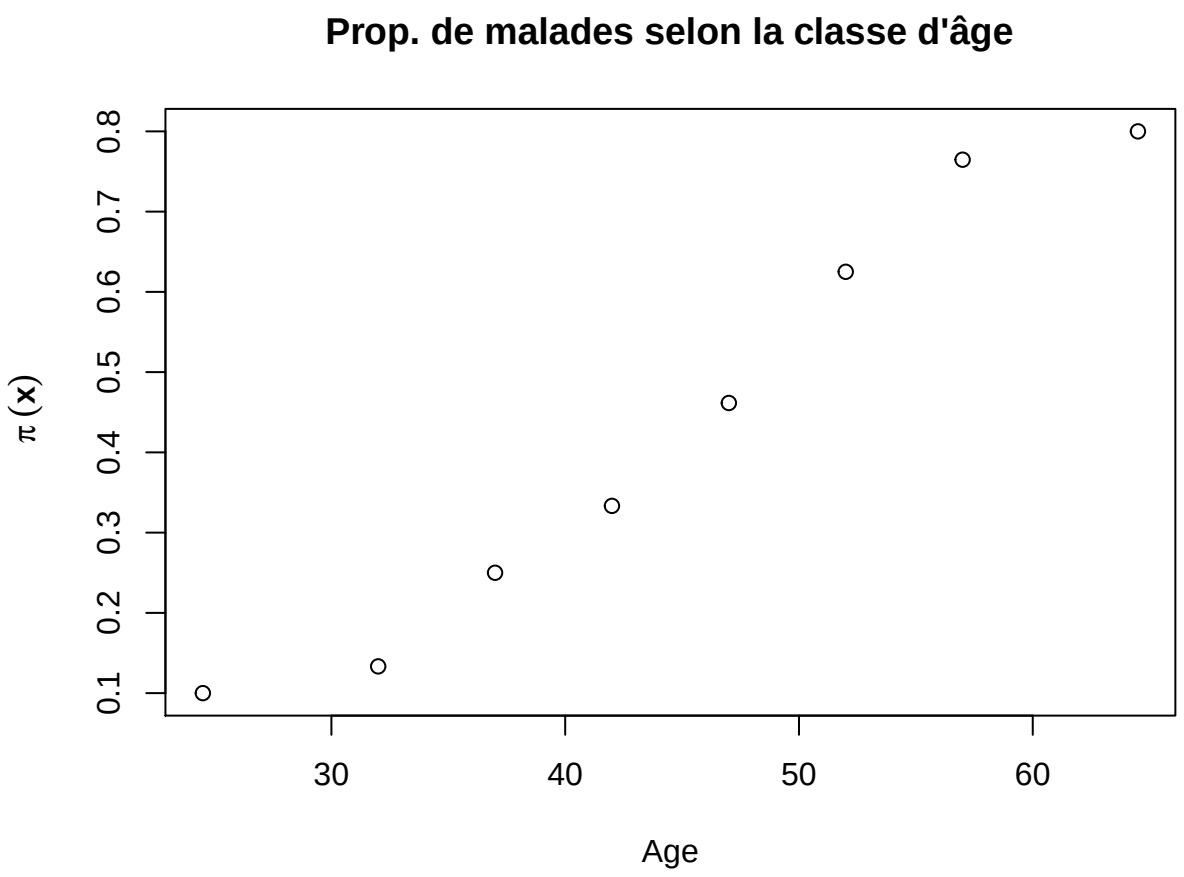
Commenter la Figure.

Calculons la proportion de malades observée selon les classes d'âge définies par la variable `AGRP`. Définir un vecteur `centre` qui donne les centres de chaque classe puis représenter le nuage de points de `p` versus `centre`. Y a-t-il une liaison entre `CHD` et `AGE` ? Quelle est la forme de ce graphique ? Quel est son intérêt comparativement au graphique précédant? Quel modèle suggérez-vous d'utiliser ?

```
#n_i : nombre de malades selon les classes d'age
n_i=tapply(X=CHD,INDEX=AGRP,FUN=sum)
#n : nombre de patiens selon les clases
n=tapply(X=CHD,INDEX=AGRP,FUN=length)
#p : proportion de maladas selon les classes d'age
p=n_i/n
#centre : centre de chaque classe
centre=c(24.5,32,37,42,47,52,57,64.5)
#Affichage du tableau
cbind(centre,n_i,n,p)
```

```
##   centre n_i  n         p
## 1   24.5   1 10 0.1000000
## 2   32.0   2 15 0.1333333
## 3   37.0   3 12 0.2500000
## 4   42.0   5 15 0.3333333
## 5   47.0   6 13 0.4615385
## 6   52.0   5  8 0.6250000
## 7   57.0  13 17 0.7647059
## 8   64.5   8 10 0.8000000
```

```r
#Plot
plot(p~centre,main="Prop. de malades selon la classe d'âge",
     xlab="Age", ylab=expression(italic(~pi)~(bold(x))))
```



Comenter la dernière Figure.

Dans le langage `R`, la fonction `glm()` permet de faire differents types de régressions linéaires généralisées, ainsi que différents types de régressions non-linéaires. Aussi cette fonction permet de ajuster differents families de modèles : logit, probit, etc. Il faut en revanche avoir recours à un argument supplémentaire ,`family`, qui definit le type de modèle qu'on souhaite faire. Pour plus de details regarder l'aide grace au commande `help(glm)`.

Commençons pour ajuster une régression logistique de `CHD` en fonction de `AGE`. Commenter les résultats (tests de significativité, nombre de degrés de liberté).

```r
CHD.logit = glm(CHD~AGE, family=binomial(link="logit"))
```

L'adequation correspondant à cet ajustement est le modèle logistique donné par

$$\pi(x) = \exp(\beta_0 + \beta_1 x)/(1 + \exp(\beta_0 + \beta_1 x))$$

ou encore le modele logit donnée par

$$\log(\pi(x)/(1 - \pi(x))) = \beta_0 + \beta_1 x$$

Nous pouvons accéder à un résultat plus detaillé de la régression logistique grâce à la fonction `summary()`. Dans ce resumé on aussi quelques informations sur les statistiques d'ajustement du modèle. Le critère AIC (Akaike Information Criterion) qui est un critère de pénalisation de la log vraisemblance prenant en compte le nombre de variables explicatives. La deviance (-2log du maximun de la vraisemblance)

```
summary(CHD.logit)
```

```
##
## Call:
## glm(formula = CHD ~ AGE, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.30945    1.13365  -4.683 2.82e-06 ***
## AGE          0.11092    0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

Annalyser les résultats du tableau.

Que font les commandes suivantes ?

```
CHD.logit$deviance
```

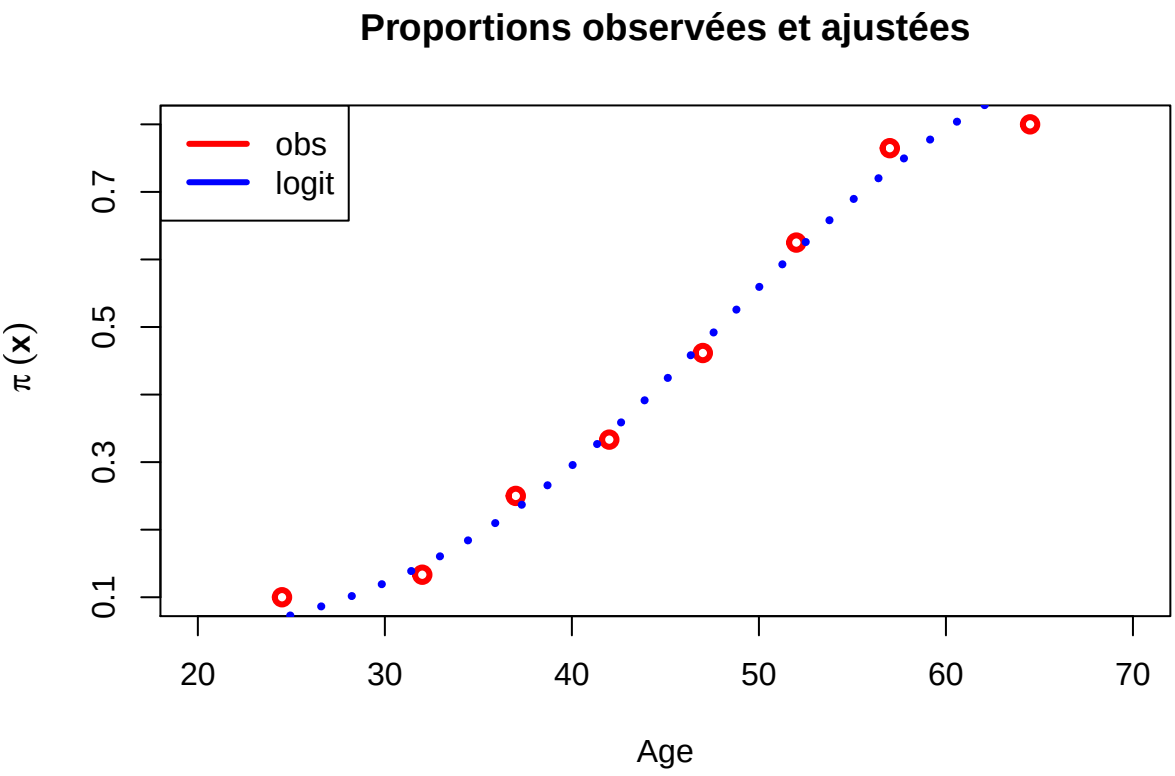
```
## [1] 107.3531
```

```
#AIC : deviance + 2 * (nombre des paramètres à estimer)
CHD.logit$deviance +2*2
```

```
## [1] 111.3531
```

Nous allons représenter sur un même graphique les proportions selons la clase d'âge et la courbe logistique ajustée:

```
beta0=coef(CHD.logit)[1]; beta1=coef(CHD.logit)[2]
abscisse1 = seq(0,100,length=100)
plot(centre,p,col='red',lwd=3,xlim=c(20,70),ylim=c(0.1,0.8),
     xlab="Age",ylab=expression(italic(~pi)~(bold(x))))
lines(abscisse1,plogis(beta0 + beta1*abscisse1),col='blue',lwd=4,lty=3)
title("Proportions observées et ajustées")
legend("topleft", c("obs","logit"),lwd=3,col=c("red","blue"))
```



Ajuster de même le modèle probit puis comparer les deux modèles.

```
CHD.probit = glm(CHD~AGE, family=binomial(link="probit"))
summary(CHD.probit)
```

```
##
## Call:
## glm(formula = CHD ~ AGE, family = binomial(link = "probit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9713  -0.8608  -0.4499   0.8358   2.3269
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.14573    0.62460  -5.036 4.74e-07 ***
## AGE          0.06580    0.01335   4.930 8.20e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.50  on 98  degrees of freedom
## AIC: 111.5
##
```

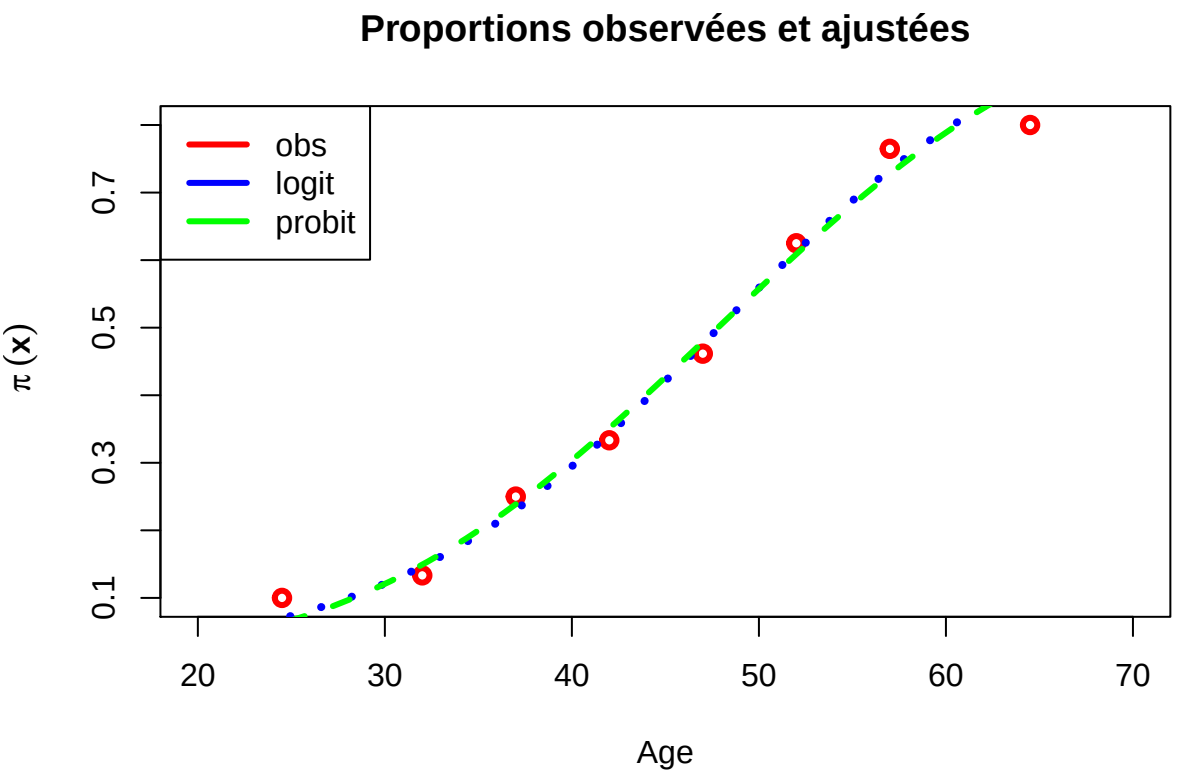
```
## Number of Fisher Scoring iterations: 4
```

```
beta0P=coef(CHD.probit)[1];beta1P=coef(CHD.probit)[2]
beta0/beta0P; beta1/beta1P
```

```
## (Intercept)
##     1.68783

##      AGE
## 1.685623
```

```
plot(centre,p,col='red',lwd=3,xlim=c(20,70),ylim=c(0.1,0.8),
     xlab="Age",ylab=expression(italic(~pi)~(bold(x))))
lines(abscisse1,plogis(beta0 + beta1*abscisse1),col='blue',lwd=4,lty=3)
lines(abscisse1,pnorm(beta0P + beta1P*abscisse1),col='green',lwd=3,lty=2)
title("Proportions observées et ajustées")
legend("topleft", c("obs","logit","probit"),
lwd=3,col=c("red","blue","green"))
```



Les deux modèles (logit et probit) que l'on voit dans la figure sont assez semblables. En general les modeles logit et probit fournissent des valeurs très proches. Toutefois, pour des commodités de calcul, l'expression du probit, étant pas explicite, on prefère souvent le modèle logit.

Estimer, dans chacun des deux modèles la cote d'un individu âgés de 30 ans. Commenter. Estimer le rapport de cotes correspondant à la variable `AGE`.

```r
#Rapp.cottes
Rcote.30=exp(beta0 + beta1*30); Rcote.30
```

```
## (Intercept)
##   0.1378183
```

```r
Rcote.30P=pnorm(beta0P + beta1P*30)/(1-pnorm(beta0P + beta1P*30)); Rcote.30P
```

```
## (Intercept)
##   0.1372406
```

```r
#Rapport de rapport des cotes (odds-ratio)
OR=exp(beta1);OR
```

```
##      AGE
## 1.117307
```

Modèle Logistique Multiple

Exemple 2 : Nous traitons un problème de défaut bancaire. Nous cherchons à déterminer quels clients seront en défaut sur leur dette de carte de crédit (ici `defaut` = 1 si le client fait défaut sur sa dette). La variable `defaut` est la variable réponse.

Nous disposons d'un échantillon de taille 10000 et 3 variables explicativesLes variables explicatives sont les suivantes :

- `student`: variable à 2 niveaux {0,1} (student = 1 si le client est un étudiant).
- `balance`: montant moyen mensuel d'utilisation de la carte de crédit.
- `income`: revenu du client.

```r
library(ISLR)
data(Default); attach(Default)
#Nombre de lignes
nrow(Default)
```

```
## [1] 10000
```

```r
#Nombre de colonnes
ncol(Default)
```

```
## [1] 4
```

```r
#nom de colones
names(Default)
```

```
## [1] "default" "student" "balance" "income"
```

```r
#résumé des données (un peu de stats descriptive)
summary(Default)
```

```
##  default    student       balance           income
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340
##                        Median : 823.6   Median :34553
##                        Mean   : 835.4   Mean   :33517
##                        3rd Qu.:1166.3   3rd Qu.:43808
##                        Max.   :2654.3   Max.   :73554
```

```r
#Affichage des 6 valeurs
head(Default)
```

```
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

```r
#Transformation de la variable default à 0 si Non et 1 si Yes
Default$default= as.numeric(Default$default)-1
head(Default)
```

```
##   default student   balance    income
## 1       0      No  729.5265 44361.625
## 2       0     Yes  817.1804 12106.135
## 3       0      No 1073.5492 31767.139
## 4       0      No  529.2506 35704.494
## 5       0      No  785.6559 38463.496
## 6       0     Yes  919.5885  7491.559
```

Pour illustrer un peu le problème et le jeu de donnés nous commençerons par des modèles simples et nous finaliserons par des modèles multiples.

Modèle avec la variable balance (variable exlicative qualitative)

```r
#Pr(default = Yes|balance).
fit.glm1=glm(default ~ balance,data=Default,family=binomial)
fit.glm1
```

```
##
## Call:  glm(formula = default ~ balance, family = binomial, data = Default)
##
## Coefficients:
## (Intercept)      balance
##  -10.651331     0.005499
```

```
##
## Degrees of Freedom: 9999 Total (i.e. Null);  9998 Residual
## Null Deviance:       2921
## Residual Deviance: 1596  AIC: 1600
```

```
fit.glm1$coeff
```

```
##   (Intercept)       balance
## -10.651330614   0.005498917
```

```
fit.glm1$deviance
```

```
## [1] 1596.452
```

Une fois que les coefficients ont été estimés, il est simple de calculer la probabilité de défaut etant donné balance (solde moyen de carte de crédit donné . Par exemple, en utilisant les estimations des coefficients indiqués dans le tableau precedant, nous prévoyons que la probabilité de defaut pour un client qui a un balance de 1000 dollars et 2000 dollars est

```
xnew=data.frame(balance=c(1000,2000))
xnew
```

```
##   balance
## 1    1000
## 2    2000
```

```
predict.glm(fit.glm1,xnew,type="response")
```

```
##           1           2
## 0.005752145 0.585769370
```

Commenter.

Modèle avec la variable student (variable explicative qualitative)

On peut utiliser des prédicteurs qualitatives avec le modèle de régression logistique. Il est important de signaler que dans ce TP la variable student a été transformé (1 si student et 0 si non-student).

```
head(Default)
```

```
##   default student   balance    income
## 1       0      No  729.5265 44361.625
## 2       0     Yes  817.1804 12106.135
## 3       0      No 1073.5492 31767.139
## 4       0      No  529.2506 35704.494
## 5       0      No  785.6559 38463.496
## 6       0     Yes  919.5885  7491.559
```

Utiliser le tableaux de contingence suivant et estimer (avec un calcule à la main) les coefficients du modèle logistique.

```
##         student   No  Yes
## default
## 0              6850 2817
## 1               206  127
```

```
#transformation de la variable student à 0 et 1
Default$student=as.numeric(Default$student)-1
head(Default)
```

```
##   default student   balance    income
## 1       0       0  729.5265 44361.625
## 2       0       1  817.1804 12106.135
## 3       0       0 1073.5492 31767.139
## 4       0       0  529.2506 35704.494
## 5       0       0  785.6559 38463.496
## 6       0       1  919.5885  7491.559
```

```
fit.glm2=glm(default~student,data=Default,family=binomial)
summary(fit.glm2)
```

```
##
## Call:
## glm(formula = default ~ student, family = binomial, data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2970  -0.2970  -0.2434  -0.2434   2.6585
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
## student      0.40489    0.11502    3.52 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2908.7  on 9998  degrees of freedom
## AIC: 2912.7
##
## Number of Fisher Scoring iterations: 6
```

```
fit.glm2$deviance
```

```
## [1] 2908.683
```

```
fit.glm2$deviance + 2*2
```

```
## [1] 2912.683
```

```r
#AIC=Residual deviance + 2 * (nombre de parametres à estimer)
```

Question : estimer $\mathbb{P}(\text{default} = \text{Yes}|\text{student} = \text{Yes})$ et $\mathbb{P}(\text{default} = \text{Yes}|\text{student} = \text{Non})$

$$\mathbb{P}(\text{default} = \text{Yes}|\text{student} = \text{Yes}) = \exp(-3.50 + 0.40 \times 1)/(1 + \exp(-3.50 + 0.40 \times 1)) = 0.0431,$$

$$\mathbb{P}(\text{default} = \text{Yes}|\text{student} = \text{Non}) = \exp(-3.50 + 0.40 \times 0)/(1 + \exp(-3.50 + 0.40 \times 0)) = 0.0292$$

Regression logistique Multiple (avec 2 variables)

```r
fit.glm3=glm(default~ student + balance ,data=Default,family=binomial)
summary(fit.glm3)
```

```
##
## Call:
## glm(formula = default ~ student + balance, family = binomial,
##     data = Default)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4578  -0.1422  -0.0559  -0.0203   3.7435
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.075e+01  3.692e-01 -29.116  < 2e-16 ***
## student     -7.149e-01  1.475e-01  -4.846 1.26e-06 ***
## balance      5.738e-03  2.318e-04  24.750  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.7  on 9997  degrees of freedom
## AIC: 1577.7
##
## Number of Fisher Scoring iterations: 8
```

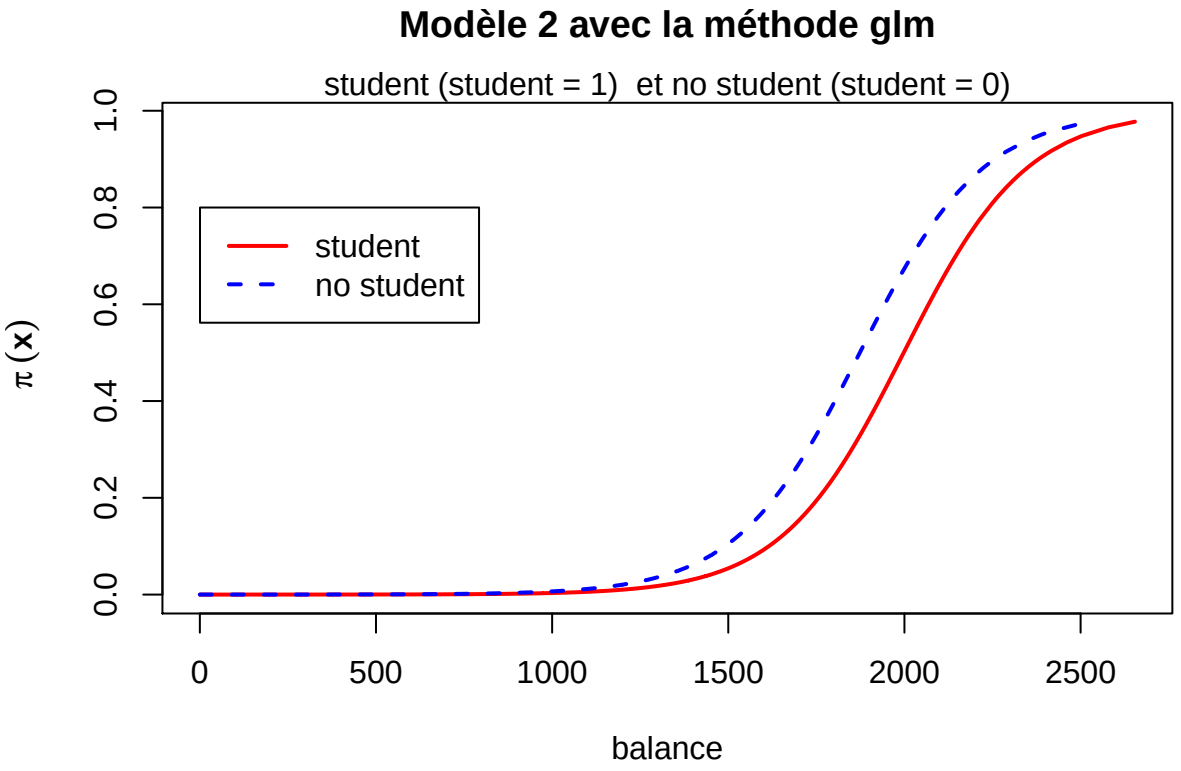
```r
#rapell de nom des variables
names(Default)
```

```
## [1] "default" "student" "balance" "income"
```

```r
#Definicion des axes pour student
ord1=predict.glm(fit.glm3,subset(Default[,c("student","balance")],(Default[,"student"]==1)),
                 type="response")
absc1=subset(Default[,"balance"],(Default[,"student"]==1))
#Definicion des axes pour no-student
```

```
ord2=predict.glm(fit.glm3,subset(Default[,c("student","balance")],(Default[,"student"]==0)),
                 type="response")
absc2=subset(Default[,"balance"],(Default[,"student"]==0))

plot(t(absc1)[order(absc1)],ord1[order(absc1)],col="red",lty=1,xlab="balance",
     ylab=expression(italic(~pi)~(bold(x))),
     main= "Modèle 2 avec la méthode glm" , type = "l", lwd=2
)
lines(t(absc2)[order(absc2)],ord2[order(absc2)],col="blue",lty=2,lwd=2)
mtext("student (student = 1)  et no student (student = 0)")
leg.txt = c("student", "no student")
legend(0.5, 0.8, leg.txt, lty = c(1,2), col = c("red","blue"), lwd=c(2,2))
```




```
fit.glm4=glm(default~balance + income + student ,data=Default,family=binomial)
summary(fit.glm4)
```

Modèle avec les 3 variables explicatives

```
##
## Call:
## glm(formula = default ~ balance + income + student, family = binomial,
##     data = Default)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## student     -6.468e-01  2.363e-01  -2.738  0.00619 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

```
AIC.glm1=fit.glm1$deviance+2*2
AIC.glm2=fit.glm2$deviance+2*3
AIC.glm3=fit.glm3$deviance+2*4
AIC.glm4=fit.glm3$deviance+2*5
cbind(AIC.glm1,AIC.glm2,AIC.glm3,AIC.glm4)
```

```
##      AIC.glm1 AIC.glm2 AIC.glm3 AIC.glm4
## [1,] 1600.452 2914.683 1579.682 1581.682
```

Quel modèle choisir ?