Logistic_Regression

Walid Keddad

Exercice 1 : Modèle Logistique Simple

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 "MCV.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.

chargement des données:

```
setwd("f:/ml")
data = read.csv("MCV.txt" , header = T , sep = "\t")
head(data)
```

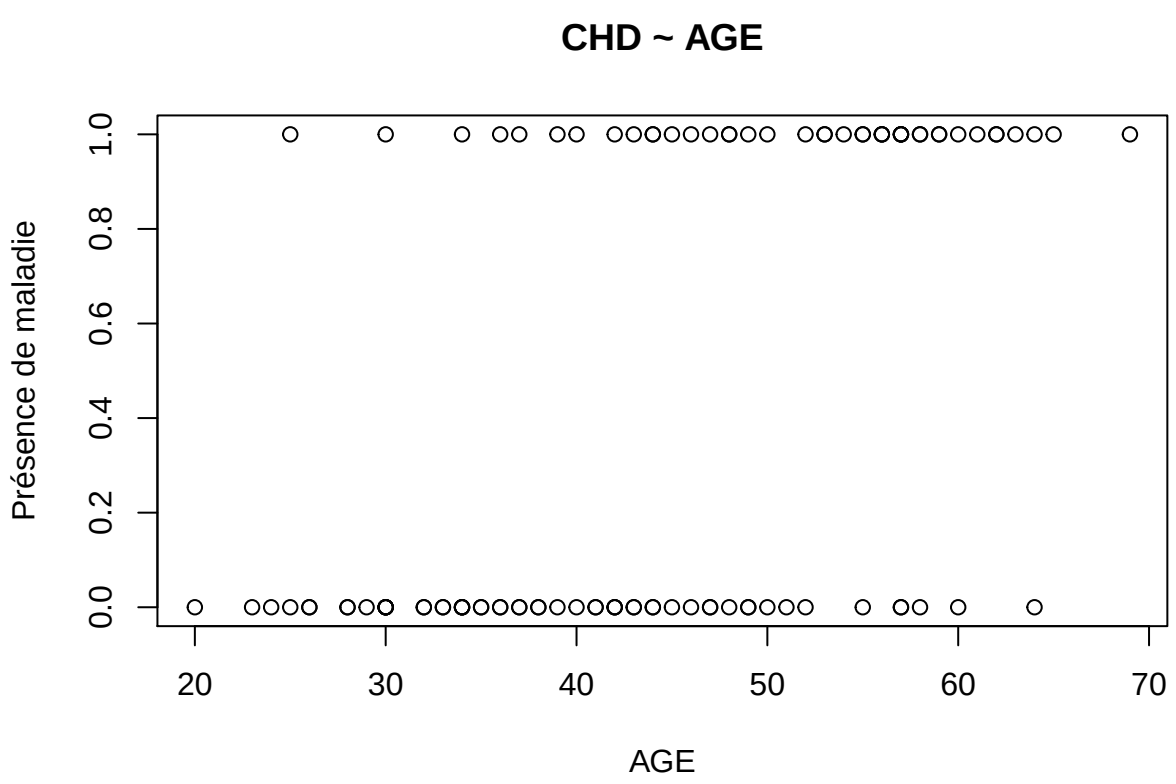
```
##   ID AGRP AGE CHD
## 1  1    1  20   0
## 2  2    1  23   0
## 3  3    1  24   0
## 4  4    1  25   0
## 5  5    1  25   1
## 6  6    1  26   0
```

```
summary(data)
```

```
##        ID              AGRP           AGE             CHD
##  Min.   :  1.00   Min.   :1.00   Min.   :20.00   Min.   :0.00
##  1st Qu.: 25.75   1st Qu.:2.75   1st Qu.:34.75   1st Qu.:0.00
##  Median : 50.50   Median :4.00   Median :44.00   Median :0.00
##  Mean   : 50.50   Mean   :4.48   Mean   :44.38   Mean   :0.43
##  3rd Qu.: 75.25   3rd Qu.:7.00   3rd Qu.:55.00   3rd Qu.:1.00
##  Max.   :100.00   Max.   :8.00   Max.   :69.00   Max.   :1.00
```

On veut étudier la relation entre **CHD** et la variable explicative **AGE** , on les représente avec un nuage de point :

```
plot(data$AGE,data$CHD,xlab = "AGE" ,ylab = "Présence de maladie" , main = "CHD ~ AGE")
```



On constate que l'age a un impact sur la présence de maladie , plus une personne est agé plus elle est la probabilté qu'elle soit malade

.

On calcule la proportion de malades observée selon les classes d'âge définies par la variable **AGRP** , et On crée un vecteur qui donne les centres de chaque classe

```
proportion = tapply(data$CHD,data$AGRP, mean)
proportion
```

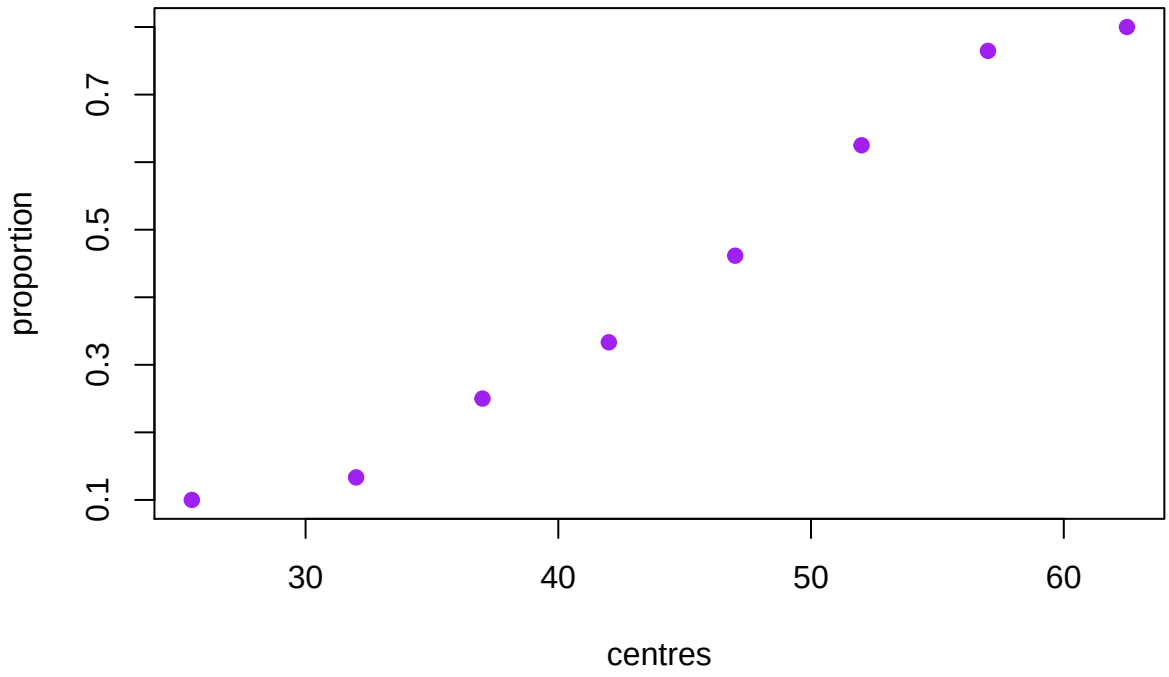
```
##         1         2         3         4         5         6         7
## 0.1000000 0.1333333 0.2500000 0.3333333 0.4615385 0.6250000 0.7647059
##         8
## 0.8000000
```

```
centres = tapply(data$AGE,data$AGRP, median)
centres
```

```
##    1    2    3    4    5    6    7    8
## 25.5 32.0 37.0 42.0 47.0 52.0 57.0 62.5
```

Nuage de points de **proportion** en fonction de **centres**

```r
plot(proportion~centres , pch = 19 , col = "purple")
```



On voit qu'il y a une relation entre **AGE** et **CHD** , et on voit aussi que le graphe a une curve **sigmoid** donc on peut appliquer un regression logistique sur ces données

.

Commençons par ajuster une régression logistique de **CHD** en fonction de **AGE** :

```r
mpg_model = glm(data$CHD~data$AGE ,"binomial")
summary(mpg_model)
```

```
##
## Call:
## glm(formula = data$CHD ~ data$AGE, family = "binomial")
##
## 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 ***
## data$AGE     0.11092    0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

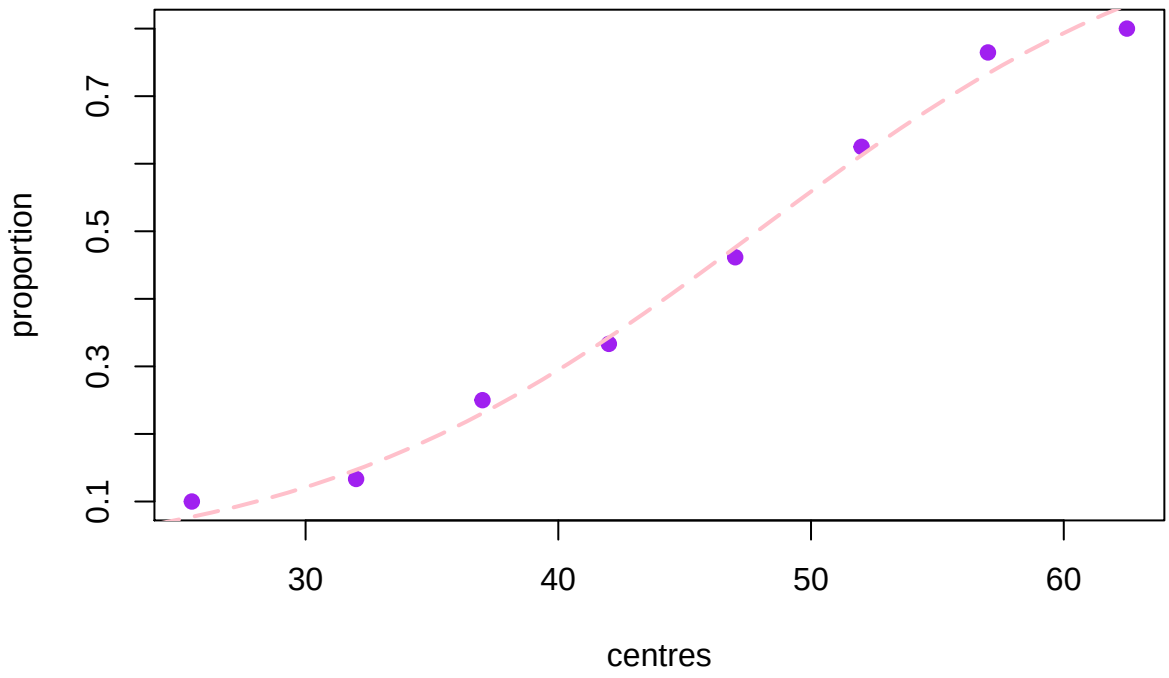
```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

Comme la valeur p de **AGE** est égale a **2.82e-06** donc la variable AGE est significative dans le model . Nombre de degrés de liberté **98**

.

Afin de mieux discerner les relations entre les différentes classes, on va représenter sur un même graphique les proportions selon la **classe d'âge** et la **courbe logistique ajustée**.

```
coefs = coef(mpg_model)
intercept = coefs[1]
age_coef = coefs[2]
point = seq(0,100,length=100)
plot(centres,proportion,col="purple",pch=19)
lines(point,plogis(intercept + age_coef*point),col='pink',lwd=2 ,lty=5 )
```

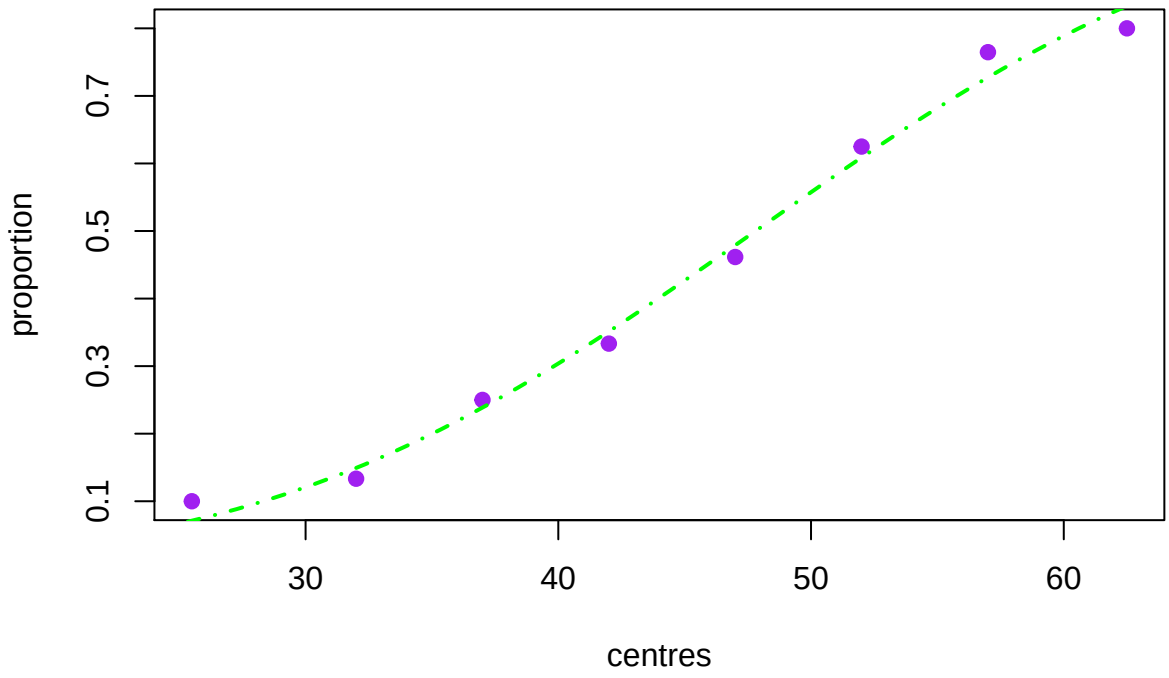


Maintenant on va ajuster le model **probit** :

```r
probit_model = glm(data$CHD~data$AGE , "binomial"(link="probit"))
summary(probit_model)
```

```
##
## Call:
## glm(formula = data$CHD ~ data$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 ***
## data$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
```

```r
intercept2 = probit_model$coefficients[1]
age_coef2 = probit_model$coefficients[2]
plot(centres,proportion,col="purple",pch=19)
lines(point,pnorm(intercept2 + age_coef2*point),col='green',lwd=2 , lty=4 )
```



On voit que les deux models **probit** et **logit** sont pas très différents, ils donnent le meme resultat

Exercice 2 : Modèle Logistique Multiple

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. La base de données Defaut est accessible à partir du package ISLR que vous devez installer au préalable. La base Defaut dispose d'un échantillon de tailles 10000 et 3 variables explicatives. Les 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.

Chargement des données :

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.2.5
```

```
def = Default
attach(def)
head(def)
```

```
##   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
```

```
summary(def)
```

```
##  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
```

Afin de faciliter le traitement, on transforme la variable default à 0 si Non et 1 si Yes

```
def$default = ifelse( def$default == "No" ,0,1)
head(def)
```

```
##   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
```

On Construit un modèle de régression logistique avec la variable **balance** comme variable explicative qualitative

```
balance_model = glm(default~balance , family = "binomial"(link = "logit"))
summary(balance_model)
```

```
##
## Call:
## glm(formula = default ~ balance, family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <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: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

Une fois que les coefficients ont été estimés, il est simple de calculer la probabilité de défaut étant donné balance (solde moyen de carte de crédit donné). En utilisant les estimations des coefficients indiqués dans le tableau précédant, on va prédire la probabilité de defaut pour un client qui a une balance de **1000**, **1500**, **2000** et **3000** dollars respectivement.

```
test = data.frame(balance=c(1000,1500,2000,3000))
result = predict.glm(balance_model , test , type = "response")
result
```

```
##           1           2           3           4
## 0.005752145 0.082947624 0.585769370 0.997115227
```

On voit que la probabilité de **défault** augmente avec l'augmentation du **balance**

Tableau de contingence des variables **default** et **student** :

```
table(student , default)
```

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

Model Logit avec **student** comme variable explicative :

```
student_model = glm(default~student , "binomial"(link = "logit"))
summary(student_model)
```

```
##
## Call:
## glm(formula = default ~ student, family = binomial(link = "logit"))
##
## 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 ***
## studentYes   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
```

```r
student_model$coefficients[1] # -3.5041
```

```
## (Intercept)
##   -3.504128
```

```r
student_model$coefficients[2] # 0.4048
```

```
## studentYes
##  0.4048871
```

```r
# p(default = yes , student = yes ) = e(-3.50 + 0.40 * 1) / 1 + (e(-3.50 + 0.40 * 1))

# p(default = yes , student = no ) = e(-3.50 + 0.40 * 0) / 1 + (e(-3.50 + 0.40 * 0))
```

Maintenant on construit un modèle de régression logistique multiple avec les 2 variables explicatives **student** et **balance**.

```r
student_balance_model = glm(default~student + balance , family = "binomial")
summary(student_balance_model)
```

```
##
## Call:
## glm(formula = default ~ student + balance, family = "binomial")
##
## 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 ***
## studentYes  -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
```

Un modèle de régression logistique multiple avec les 3 variables explicatives **student** et **balance** et **income**.

```r
student_balance_income_model = glm(default~student + balance + income , family = "binomial")
summary(student_balance_income_model)
```

```
##
## Call:
## glm(formula = default ~ student + balance + income, family = "binomial")
##
## 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 ***
## studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## ---
## 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
```

Exercice 3 : Modèle linéaire généralisé

Supposons que nous partons d'une partie du jeu de données " mtcars" intégré dans R. Les données ont été extraites du magazine 1974 de Motor Trend US et comprennent la consommation de carburant et 10 aspects de la conception et de la performance automobile pour 32 automobiles (modèles 1973- 74). Nous utiliserons "vs" comme variable de résultat, "mpg" comme prédicteur continu, et "am" comme prédicteur catégorique (dichotomique ou binaire).

Chargement des données :

```
data("mtcars")
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
```

```
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
attach(mtcars)
```

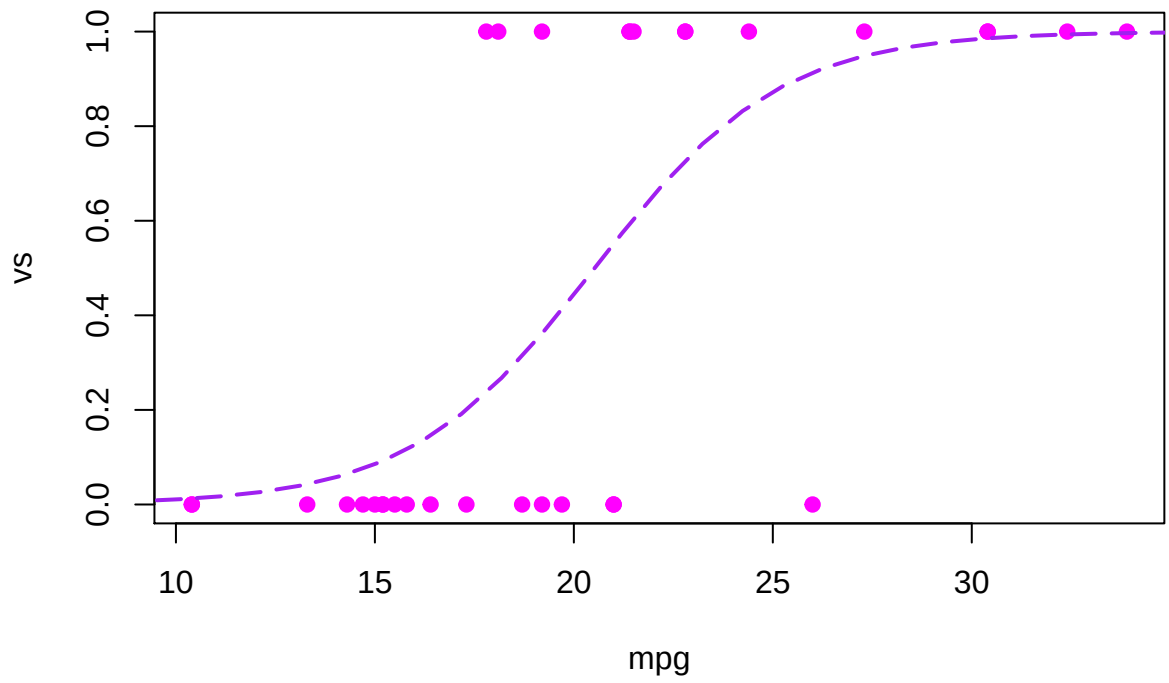
On va créer un modèle logistique où on considére **mpg** est la variable prédictive continue et **vs** est la variable de résultat qualitative binaire.

```
mpg_model = glm(vs~mpg , "binomial")
summary(mpg_model)
```

```
##
## Call:
## glm(formula = vs ~ mpg, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2127  -0.5121  -0.2276   0.6402   1.6980
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.8331     3.1623  -2.793  0.00522 **
## mpg           0.4304     0.1584   2.717  0.00659 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 25.533  on 30  degrees of freedom
## AIC: 29.533
##
## Number of Fisher Scoring iterations: 6
```

Traçant avec la fonction plot le graphe des données et du modèle régression logistique

```r
plot(mpg , vs , col=6 , pch = 19)
intercept = mpg_model$coefficients[1]
coef_mpg = mpg_model$coefficients[2]
point = seq(0,100,length.out = 100)
lines(point,plogis(intercept + coef_mpg*point),col='purple',lwd=2 ,lty=5 )
```



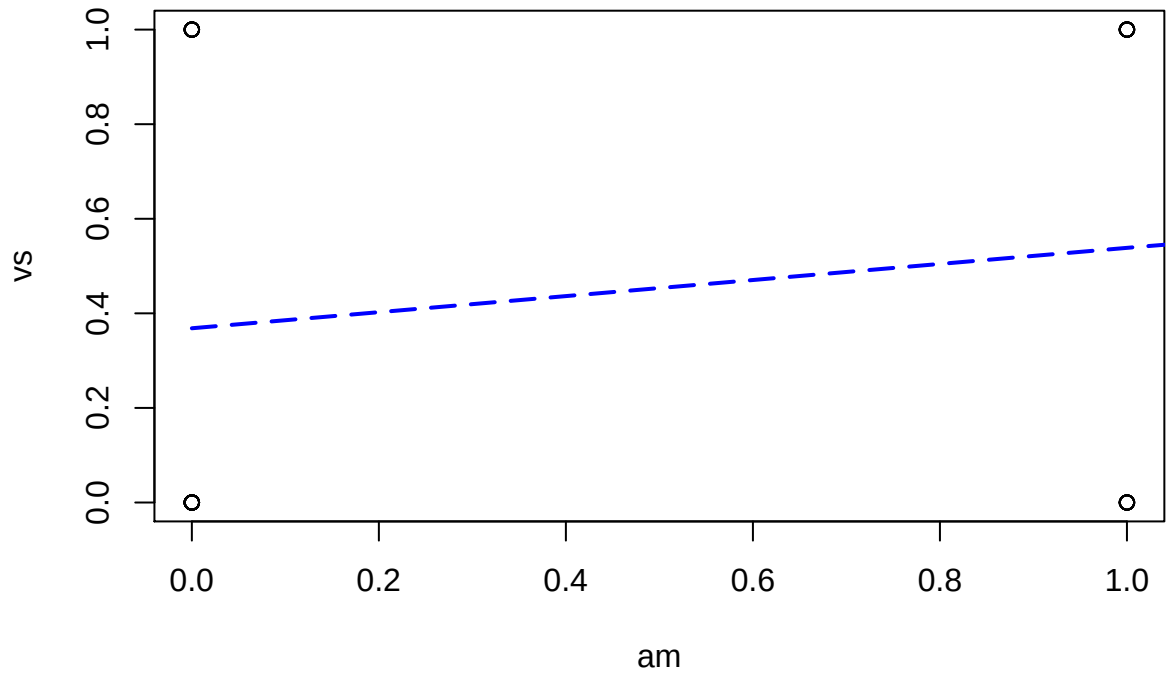
On va refaire la même chose avec la variable **am** comme variable prédective

```r
am_model = glm(vs ~ am , family = "binomial"(link = logit))
summary(am_model)
```

```
##
## Call:
## glm(formula = vs ~ am, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2435  -0.9587  -0.9587   1.1127   1.4132
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.5390     0.4756  -1.133    0.257
## am            0.6931     0.7319   0.947    0.344
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 42.953  on 30  degrees of freedom
## AIC: 46.953
##
## Number of Fisher Scoring iterations: 4
```

```
plot(am , vs )
am_intercept = am_model$coefficients[1]
coef_am = am_model$coefficients[2]
lines(point,plogis(am_intercept + coef_am * point) , col ="blue" , lwd = 2 , lty = 5)
```



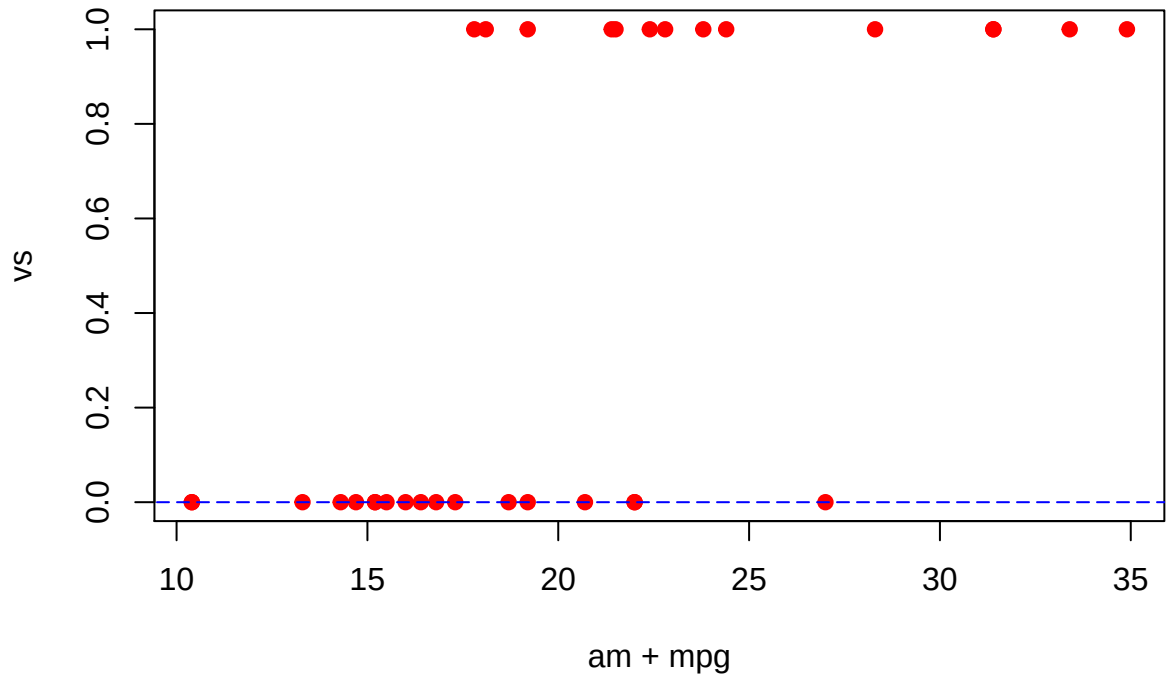
Construisant maintenant le modèle de régression avec **mpg** comme variable prédictive continue, **am** comme variable prédictive dichotomique et **vs** comme variable de résultat qualitative binaire (dichotomique).

```
multi_model = glm(vs ~ am + mpg , family = "binomial"(link = logit))
summary(multi_model)
```

```
##
## Call:
## glm(formula = vs ~ am + mpg, family = binomial(link = logit))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.05888  -0.44544  -0.08765   0.33335   1.68405
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.7051     4.6252  -2.747  0.00602 **
## am           -3.0073     1.5995  -1.880  0.06009 .
## mpg           0.6809     0.2524   2.698  0.00697 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 20.646  on 29  degrees of freedom
## AIC: 26.646
##
## Number of Fisher Scoring iterations: 6
```

```
plot (am + mpg , vs , col = "red" , pch = 19)
multi_intercept = multi_model$coefficients[1]
am_coef = multi_model$coefficients[2]
mpg_coef = multi_model$coefficients[3]
lines(point , plogis(multi_intercept + (am_coef * point) + (mpg_coef * point)) , col = "blue" , lty = 5
```

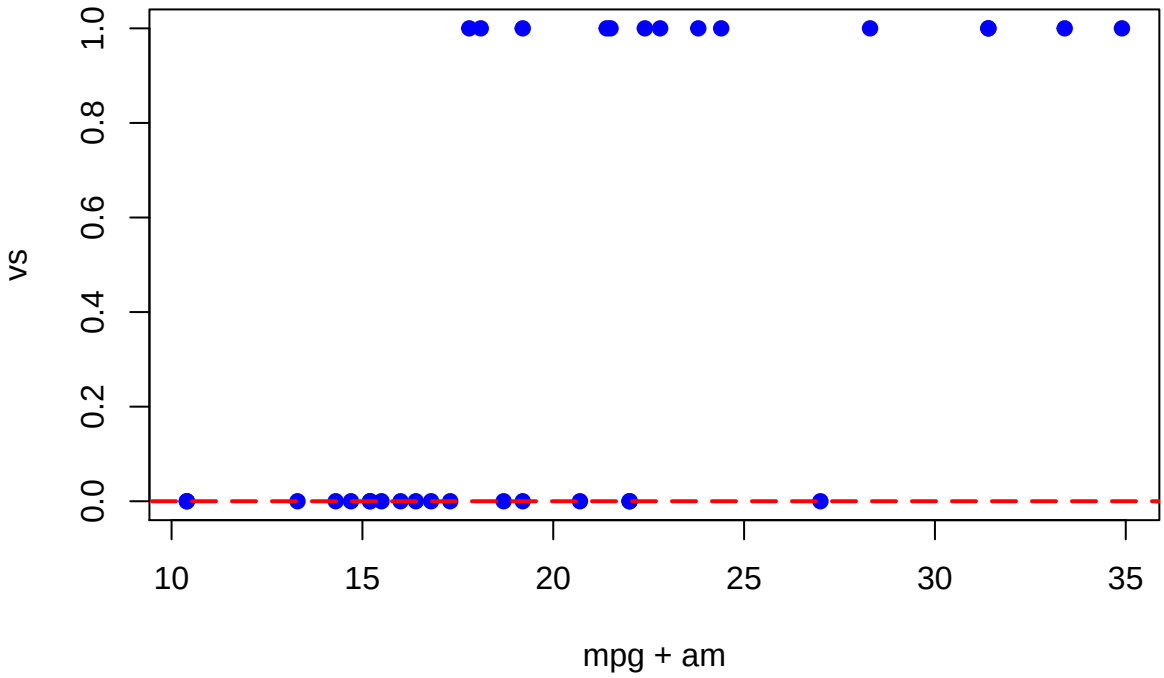


Comparant les résultats avec le model probit :

```r
probit_model = glm(vs ~ mpg + am , data = mtcars , family = "binomial"(link = "probit"))
summary(probit_model)
```

```
##
## Call:
## glm(formula = vs ~ mpg + am, family = binomial(link = "probit"),
##     data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.98526  -0.42430  -0.03027   0.32445   1.72772
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.4735     2.4437  -3.058  0.00223 **
## mpg           0.3998     0.1335   2.994  0.00275 **
## am           -1.8390     0.9126  -2.015  0.04389 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 20.347  on 29  degrees of freedom
## AIC: 26.347
##
## Number of Fisher Scoring iterations: 7
```

```r
plot(mpg + am , vs , col=4 , pch = 19)
intercept = probit_model$coefficients[1]
coef_mpg = probit_model$coefficients[2]
coef_am = probit_model$coefficients[3]
point = seq(0,100,length.out = 100)
lines(point,plogis(intercept + coef_mpg*point + coef_am*point),col='red',lwd=2 ,lty=5 )
```



On contate que les résultats sont similaires **logit** et **probit**