

LSTAT2100 - Exercices - Série 3

Solutions

Pour ce TP nous allons utiliser les données de 200 patients issus d'un hôpital. Le jeu de données se trouve dans le fichier [dt.csv](#).

Partie 1

Pour l'instant, nous n'utiliserons que les variables suivantes:

- STA: Variable binaire indiquant si le patient est décédé (1) ou pas (0).
- AGE: L'âge du patient.

Commencez par charger le jeu de données dans R puis *examinez sa structure*.

(a) À partir de la variable AGE, créez la variable AGECAT qui correspond aux catégories d'âge suivantes:

(15 – 24] (24 – 34] (34 – 44] (44 – 54] (54 – 64] (64 – 74] (74 – 84] (84 – 94]

Quelle est la proportion de décès par catégorie d'âge? Visualiser graphiquement ces chiffres. Que pouvez-vous conclure quant à l'effet de l'âge sur la survie des patients ?

Solution:

Pour charger les données, nous allons utiliser la fonction `read.csv` et pour la structure la fonction `str` (il faut toujours examiner rapidement les données avant de commencer l'analyse.)

```
data <- read.csv("Data/dt.csv", sep = ";", header = TRUE)
str(data)
```

```
## 'data.frame':    200 obs. of  21 variables:
## $ ID : int  8 12 14 28 32 38 40 41 42 50 ...
## $ STA: int  0 0 0 0 0 0 0 0 0 0 ...
## $ AGE: int  27 59 77 54 87 69 63 30 35 70 ...
## $ SEX: int  2 1 1 1 2 1 1 2 1 2 ...
## $ RAC: int  1 1 1 1 1 1 1 1 2 1 ...
## $ SER: int  1 1 2 1 2 1 2 1 1 2 ...
## $ CAN: int  1 1 1 1 1 1 1 1 1 2 ...
## $ CRN: int  1 1 1 1 1 1 1 1 1 1 ...
## $ INF: int  2 1 1 2 2 2 1 1 1 1 ...
## $ CPR: int  1 1 1 1 1 1 1 1 1 1 ...
## $ SYS: int  142 112 100 142 110 110 104 144 108 138 ...
## $ HRA: int  88 80 70 103 154 132 66 110 60 103 ...
## $ PRE: int  1 2 1 1 2 1 1 1 1 1 ...
```

```
## $ TYP: int  2 2 1 2 2 2 1 2 2 1 ...
## $ FRA: int  1 1 1 2 1 1 1 1 1 1 ...
## $ PO2: int  1 1 1 1 1 2 1 1 1 1 ...
## $ PH : int  1 1 1 1 1 1 1 1 1 1 ...
## $ PCO: int  1 1 1 1 1 1 1 1 1 1 ...
## $ BIC: int  1 1 1 1 1 2 1 1 1 1 ...
## $ CRE: int  1 1 1 1 1 1 1 1 1 1 ...
## $ LOC: int  1 1 1 1 1 1 1 1 1 1 ...
```

Remarquez que *toutes les variables sont sous forme numérique*.

```
data$STA <- factor(data$STA)
data$AGECAT <- cut(data$AGE, breaks = c(15, 24, 34, 44, 54, 64,
74, 84, 94))
```

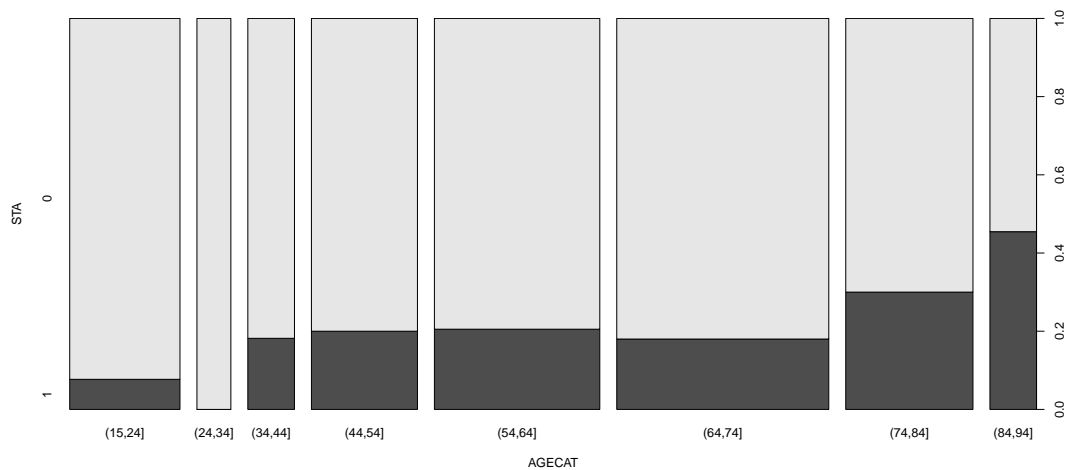
Calculons les proportions.

```
pdata1 <- prop.table(xtabs(~AGECAT + STA, data = data), 1)
pdata1
```

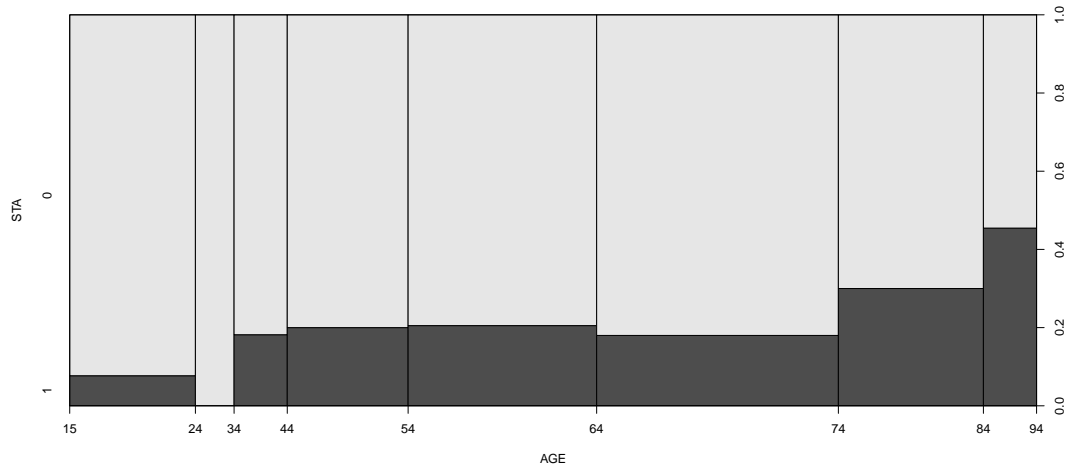
AGECAT/STA	0	1
(15,24]	0.923	0.077
(24,34]	1.000	0.000
(34,44]	0.818	0.182
(44,54]	0.800	0.200
(54,64]	0.795	0.205
(64,74]	0.820	0.180
(74,84]	0.700	0.300
(84,94]	0.545	0.455

Et voici un graphe qui résume ces calculs.

```
plot(STA ~ AGECAT, data = data)
```

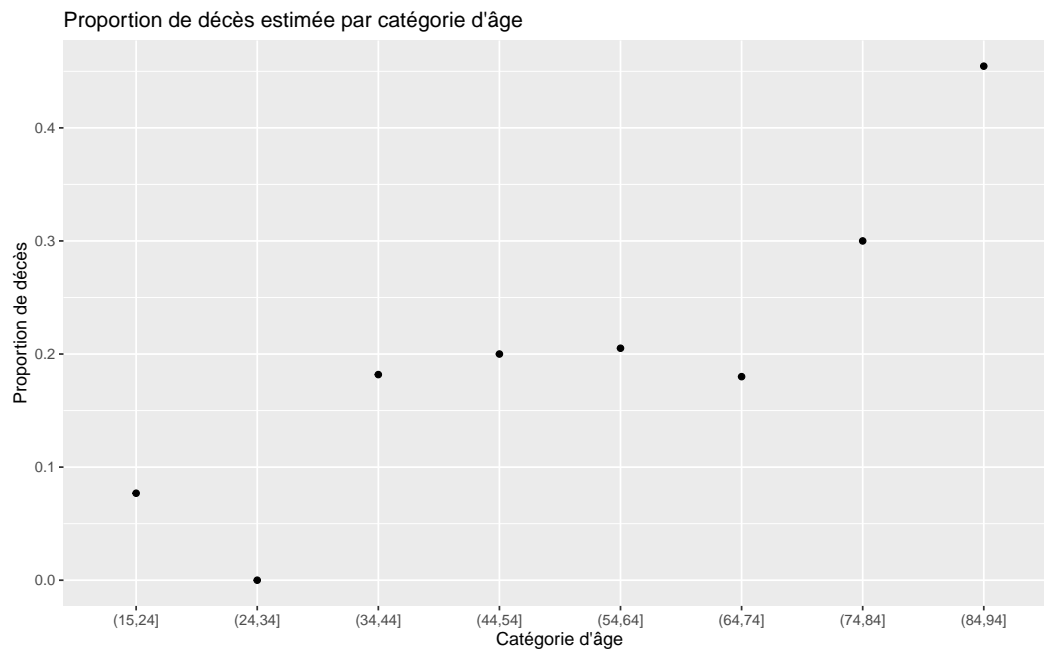


```
# ou
plot(STA ~ AGE, data = data, breaks = c(15, 24, 34, 44, 54, 64,
74, 84, 94))
```



Et voici une autre façon de représenter ces chiffres.

```
pdata1Sta1 <- subset(data.frame(pdata1), STA == "1")
ggplot(pdata1Sta1) + aes(x = AGE, y = Freq) + geom_point() +
  labs(title = "Proportion de décès estimée par catégorie d'âge",
x = "Catégorie d'âge", y = "Proportion de décès")
```

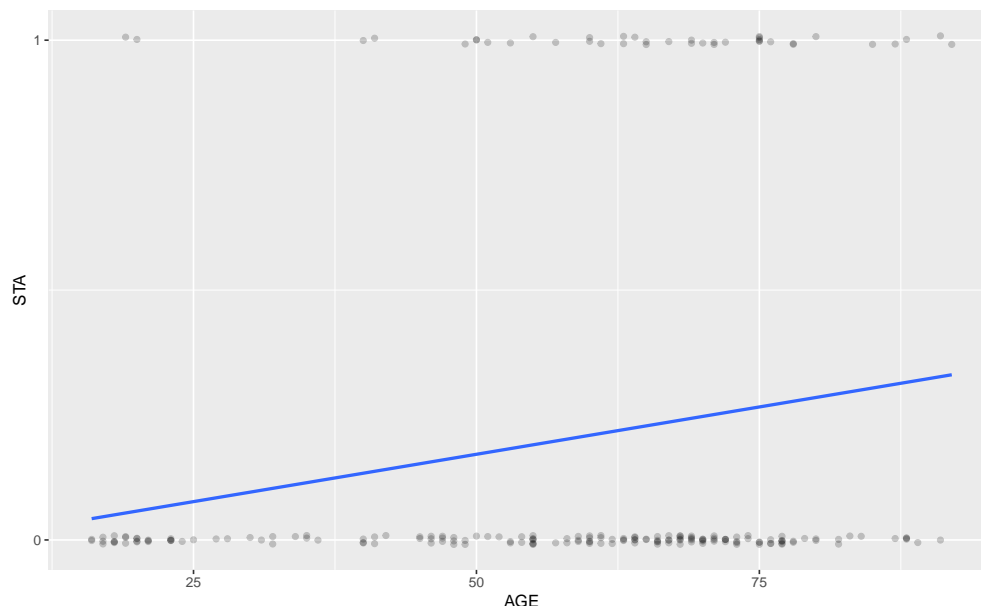


Clairement la proportion de décès tend à augmenter avec l'âge! Cette augmentation est plus prononcée dans les catégories d'âges les plus élevées (60 ans et plus).

(b) Réalisez un scatterplot de STA versus AGE. Ajouter la droite de moindre carrée à votre graphique et calculer le coefficient de détermination (classique).

Solution:

```
ggplot(data) + aes(x = AGE, y = as.numeric(as.character(STA))) +
  geom_point(alpha = 0.2, position = position_jitter(w = 0,
    h = 0.009)) + geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous(breaks = c(0, 1)) + labs(y = "STA")
```



```
with(data, cor(AGE, as.numeric(as.character(STA)))^2)
```

```
## [1] 0.0359
```

Clairement un modèle linéaire entre ces deux variables est à exclure.

(c) Écrivez l'équation du modèle logistique reliant $Y=STA$ (réponse) et $X=AGE$ (prédicteur). *L'événement de "succès" ($Y = 1$) doit être le décès.* Écrivez le log-vraisemblance de votre modèle et maximisez-le **numériquement** (sans passer par la fonction `glm()`) en utilisant, par exemple, la fonction `optim()` (voir le Help de R).

Solution:

Le modèle est $\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$, où $p(x) = P(STA = 1|Age = x) \equiv P(Y = 1|X = x)$.

Soit $p_i = P(Y_i = 1|X_i = x_i)$ et $z_i = \beta_0 + \beta_1 x_i, i = 1, \dots, n$.

Puisque $Y_i|X_i = x_i \sim \text{Ber}(p_i)$, avec $p_i = \frac{1}{1 + e^{-z_i}}$, la log-vraisemblance est donnée par $l = \sum_{i=1}^n l_i$, où

$$\begin{aligned} l_i &= y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \\ &= y_i \log\left(\frac{1}{1 + e^{-z_i}}\right) + (1 - y_i) \log\left(\frac{e^{-z_i}}{1 + e^{-z_i}}\right) \\ &= -(1 - y_i)z_i - \log(1 + e^{-z_i}). \end{aligned}$$

Notez que, puisque $z + \log(1 + e^{-z}) = \log(1 + e^z)$, on peut écrire cette dernière ligne comme $y_i z_i - \log(1 + e^{z_i})$.

```
nl <- function(b, x, y) {
  z <- b[1] + b[2] * x
  l <- sum(-(1 - y) * z - log(1 + exp(-z))) #ou l<-sum(y * z - log(1 + exp(z)))
  return(-l) # car la fonction optim() calcule le minimum alors qu'on cherche le max.
}
optim(par = c(0, 1), fn = nl, x = data$AGE, y = as.numeric(as.character(data$STA)))$par
```

```
## [1] -3.0592 0.0275
```

(d) Utilisez la fonction `glm` pour estimer le modèle tel que défini en (c). Comparez avec les estimations obtenues auparavant. Visualisez le modèle à l'aide d'un graphique adéquat.

Solution:

Avant de continuer, si ce n'est pas déjà fait, il est conseillé de transformer `STA` en `Factor`. Pour rappel, par défaut, R ordonne les niveaux d'un facteur par ordre alphanumérique croissant. Aussi, par défaut, le premier niveau (le plus petit) sera le niveau de référence.

```
levels(data$STA)
```

```
## [1] "0" "1"
```

`STA` n'a que deux niveaux "1" et "0". Ce dernier sera donc choisi, par R, comme référence.

Pour rappel, pour une variable aléatoire W à I catégories ("cat1", ..., "catI"), le fait que, par exemple, "cat1" soit la référence implique qu'en interne, R utilisera les variables indicatrices $I(W = \text{"cat2"}), \dots, I(W = \text{"catI"})$, à la place de W . Ainsi, dans le cas présent, R utilisera la variable indicatrice $I(STA = \text{"1"})$ à la place de `STA`. Vous pouvez vérifier cela à l'aide de la fonction `contrasts`.

```
contrasts(data$STA)
```

```
##      1
## 0 0
## 1 1
```

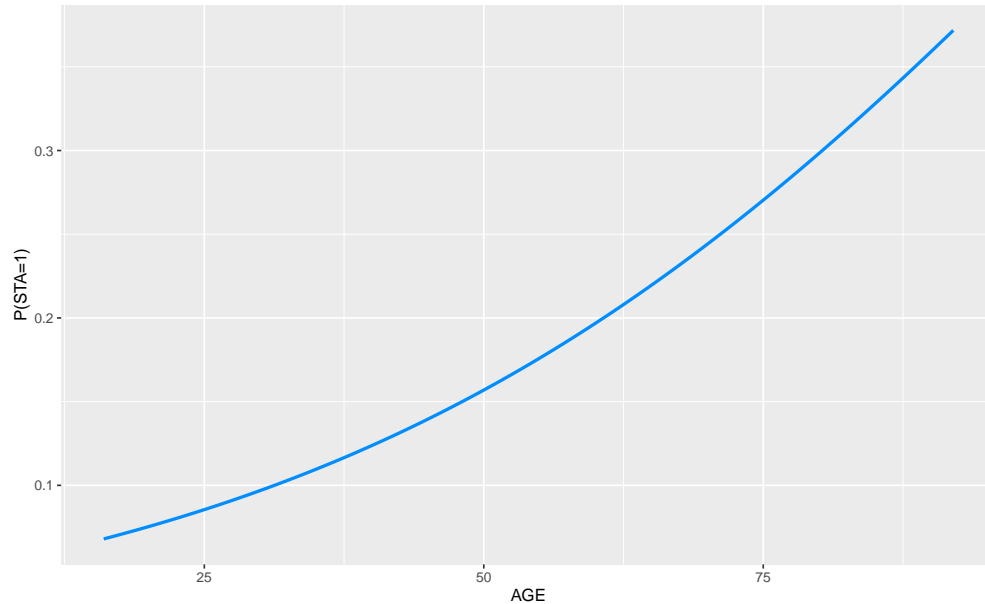
Estimons le modèle avec la fonction `glm`.

```
mdAge <- glm(STA ~ AGE, data = data, family = binomial())
summary(mdAge)
```

```
##
## Call:
## glm(formula = STA ~ AGE, family = binomial(), data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.0585      0.6961  -4.39 1.1e-05 ***
## AGE          0.0275      0.0106   2.61 0.0091 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 200.16  on 199  degrees of freedom
## Residual deviance: 192.31  on 198  degrees of freedom
## AIC: 196.3
##
## Number of Fisher Scoring iterations: 4
```

Voici un graphe qui montre la courbe estimée.

```
require(visreg)
p <- visreg(mdAge, scale = "response", band = FALSE, rug = FALSE,
  gg = TRUE) + labs(y = "P(STA=1)")
p
```



(e) Interprétez les paramètres de votre modèle. Selon ce dernier, quelle est la probabilité de mourir pour une personne âgée de 60 ans ? Construisez un IC à 95% pour cette probabilité.

Solution:

L'équation du modèle estimé est

$$\frac{\hat{p}(STA = "1"|Age)}{\hat{p}(STA = "0"|Age)} = \exp(\hat{\beta}_0 + \hat{\beta}_1 Age) = \exp(-3.059 + 0.028 Age).$$

Selon cette équation, on peut dire que la cote de décès quand $Age = 0$ (!) est $\exp(\hat{\beta}_0) = 0.047$. Et comme $\hat{\beta}_1 = 0.028 > 0$, on peut dire que cette cote (et donc la probabilité de décès) augmente avec l'âge. Plus précisément, une augmentation de l'âge d'une unité (ici d'une année), multiplie la cote de décès par $\exp(\hat{\beta}_1) = 1.028$.

Pour la prédiction, nous pouvons utiliser simplement la fonction `predict`:

```
pred <- predict(mdAge, data.frame(AGE = 60), type = "response",
  se = TRUE)
pred$fit
```

```
##      1
## 0.197
```

```
# Un interval de confiance à 95% peut être construit:
pred$fit + c(-1, 1) * qnorm(0.975) * pred$se.fit
```

```
## [1] 0.140 0.254
```

(f) Utilisez la fonction `fitted()` pour calculer les résidus de la déviance à l'aide de la formule qui définit ces derniers (voir le cours). Vérifier vos calculs à l'aide de la fonction `resid()`. Représenter ces résidus graphiquement. Que pouvez-vous en conclure ?

Solution:

Les résidus de la déviance sont

$$D_i = \text{sign}(y_i - \hat{p}_i) \sqrt{2 \left(y_i \log \left(\frac{y_i}{\hat{p}_i} \right) + (1 - y_i) \log \left(\frac{1 - y_i}{1 - \hat{p}_i} \right) \right)}$$

$$= \sqrt{-2 \log(\hat{p}_i)} I(y_i = 1) - \sqrt{-2 \log(1 - \hat{p}_i)} I(y_i = 0)$$

```
pr <- fitted(mdAge)
y <- data$STA
for.res <- sqrt(-2 * log(pr)) * (y == "1") - sqrt(-2 * log(1 -
  pr)) * (y == "0")
```

Vous pouvez vérifier que R utilise bien cette formule avec, par exemple,

```
glm.res <- resid(mdAge)
head(cbind(for.res, glm.res))
```

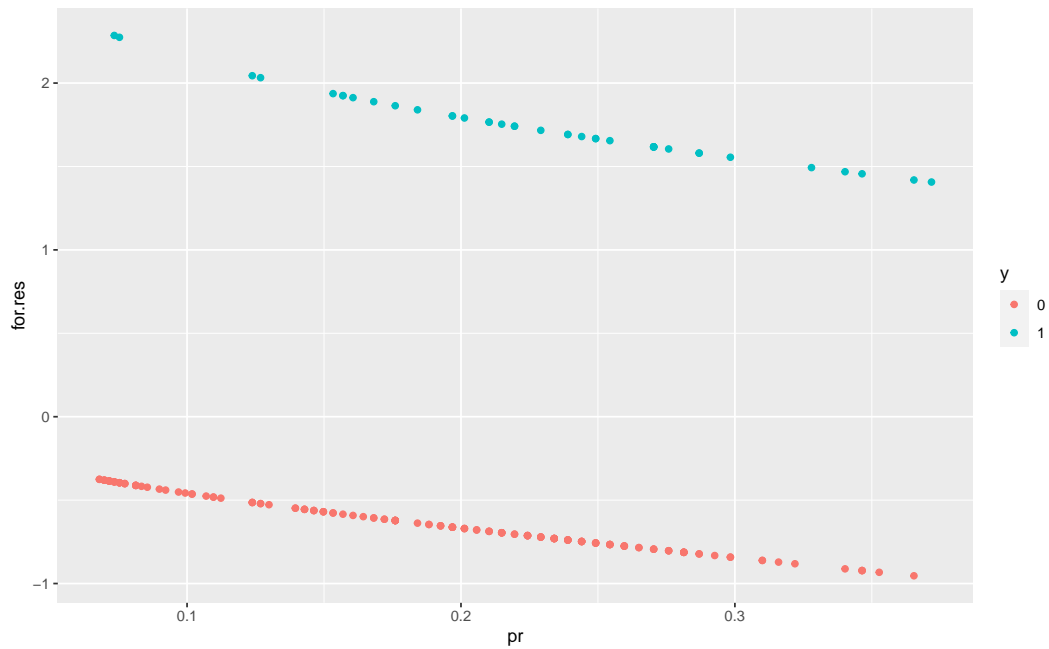
```
##   for.res glm.res
## 1 -0.434 -0.434
## 2 -0.654 -0.654
## 3 -0.813 -0.813
## 4 -0.614 -0.614
## 5 -0.912 -0.912
## 6 -0.739 -0.739
```

```
summary(abs(for.res - glm.res))
```

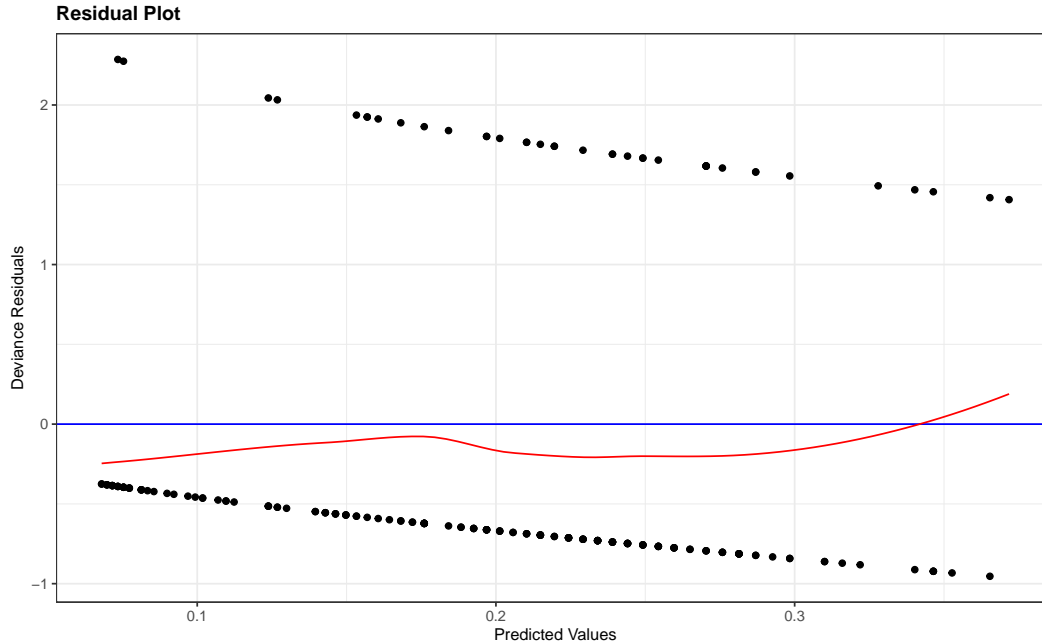
```
##      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.00e+00 0.00e+00 0.00e+00 6.13e-17 1.11e-16 2.22e-16
```

Voici le graphe des résidus versus les valeurs ajustées.

```
ggplot() + aes(x = pr, y = for.res, color = y) + geom_point()
```



```
# ou  
ggResidpanel::resid_panel(mdAge, plots = c("resid"), smoother = TRUE)
```

Contrairement au cas des données agrégées (voir le cours), ces résidus (ou les résidus de Pearson) ne sont pas d'une grande utilité pour juger de la qualité du modèle.

En conclusion, avec l'information dont on dispose, il n'est pas possible de juger de la qualité de ce modèle.

(g) Que pensez-vous de la qualité d'ajustement du modèle ? Répondez à cette question de deux façons différentes: (i) en utilisant les calculs réalisés au point (a); (ii) en utilisant la courbe ROC.

Solution:

Dans (a), on a calculé les proportions de décès pour différente catégorie d'âge.

pdata1Sta1

	AGECAT	STA	Freq
9	(15,24]	1	0.077
10	(24,34]	1	0.000
11	(34,44]	1	0.182
12	(44,54]	1	0.200
13	(54,64]	1	0.205
14	(64,74]	1	0.180
15	(74,84]	1	0.300
16	(84,94]	1	0.455

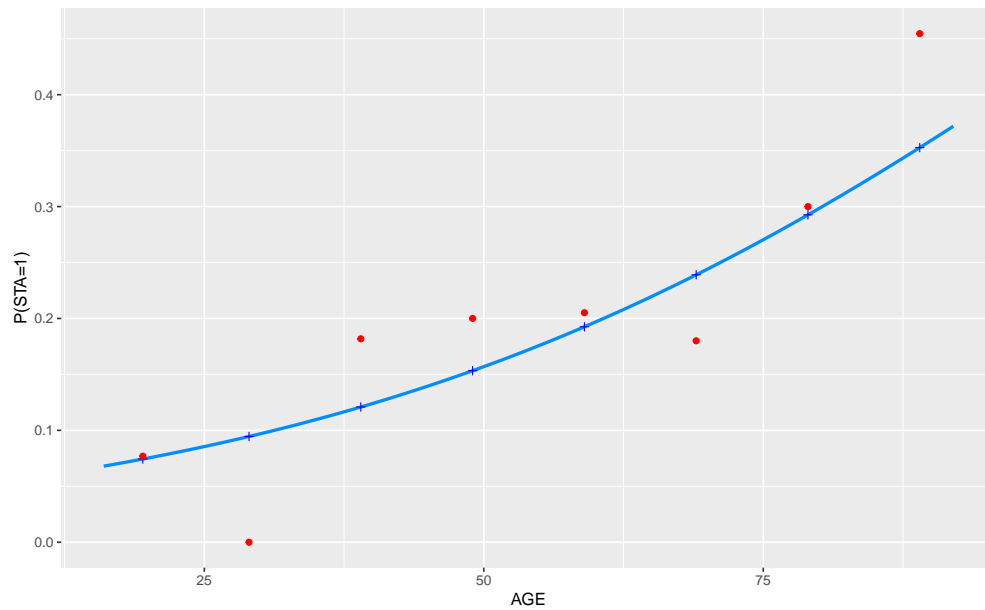
on peut comparer les chiffres qui figurent dans la dernière colonne de ce tableau avec ceux estimés par le modèle:

```
pdata1Sta1 <- data.frame(pdata1Sta1, glm.fit = predict(mdAge,
  data.frame(AGE = c(19.5, 29, 39, 49, 59, 69, 79, 89)), type = "response"))
pdata1Sta1
```

	AGECAT	STA	Freq	glm.fit
9	(15,24]	1	0.077	0.074
10	(24,34]	1	0.000	0.095
11	(34,44]	1	0.182	0.121
12	(44,54]	1	0.200	0.153
13	(54,64]	1	0.205	0.193
14	(64,74]	1	0.180	0.239
15	(74,84]	1	0.300	0.293
16	(84,94]	1	0.455	0.353

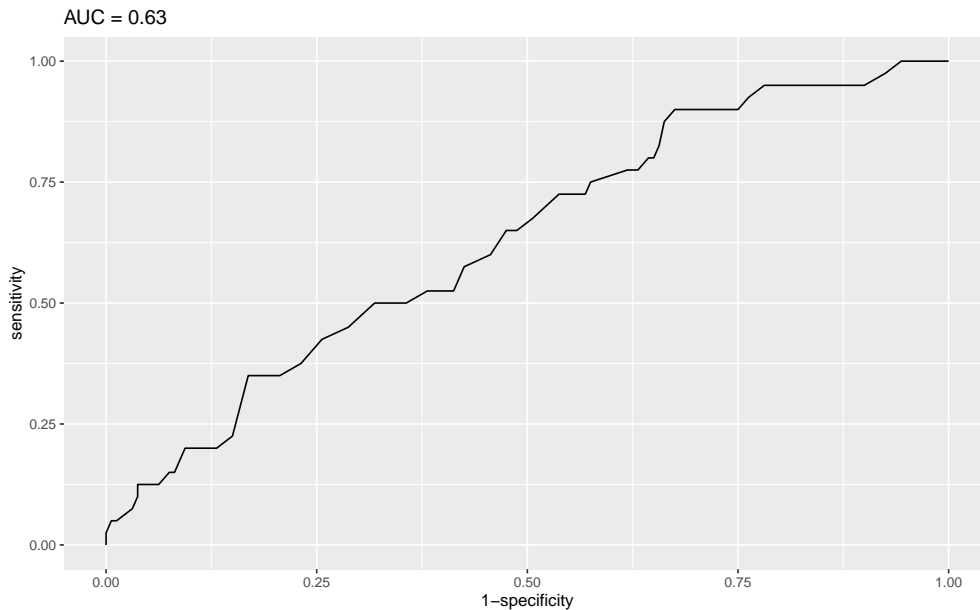
On peut représenter cela graphiquement comme suite.

```
p + geom_point(pdata1Sta1, mapping = aes(x = c(19.5, 29, 39,
49, 59, 69, 79, 89), y = glm.fit), color = "blue", shape = 3) +
geom_point(pdata1Sta1, mapping = aes(x = c(19.5, 29, 39,
49, 59, 69, 79, 89), y = Freq), color = "red")
```



Une meilleure façon d'évaluer la qualité de notre modèle est d'utiliser la courbe ROC et son AUC.

```
require(pROC)
roc <- roc(STA ~ fitted(mdAge), data = data)
ggroc(roc, legacy.axes = TRUE) + labs(title = paste("AUC =",
round(roc$auc, 2)))
```



→ on peut dire que le pouvoir discriminatif de ce modèle, tel que mesuré par l'AUC, est médiocre.

On peut aussi calculer le R^2 :

```
# Pseudo R^2
(1 - mdAge$deviance/mdAge$null.deviance) * 100
```

```
## [1] 3.92
```

Cependant, pour des données individuelles ou avec peu de répétitions, le R^2 n'est pas d'une grande aide. En effet, souvent, même pour un "très bon modèle", on peut observer une faible valeur de ce coefficient. Cela traduit simplement la difficulté de prédire (exactement) des valeurs binaires (0 ou 1) à partir d'un modèle qui n'est, à priori, pas conçu pour cela, mais plutôt conçu pour estimer des probabilités.

Partie 2

(a) On aimerait ajuster un modèle logistique avec *STA* comme variable expliquée et *CPR* comme variable explicative. En précisant vos notations et en prenant $CPR = 2$ et $STA = 0$ comme niveaux de référence,

- réalisez un tableau croisé entre les variables *STA* et *CPR*.
- donnez l'équation théorique du modèle puis
- estimez les paramètres **à la main** sans la fonction *glm* de R,
- calculez (toujours **à la main**) les intervalles de confiance et
- comparer vos résultats avec un ajustement fait avec la fonction *glm*.

Solution:

(i)

```
str(data$STA)
```

```
## Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```
str(data$CPR)

## int [1:200] 1 1 1 1 1 1 1 1 1 1 ...

data$CPR <- factor(data$CPR, levels = c("2", "1"))
str(data$CPR)

## Factor w/ 2 levels "2","1": 2 2 2 2 2 2 2 2 2 2 ...

tab <- xtabs(~CPR + STA, data = data)
tab
```

CPR/STA	0	1
2	6	7
1	154	33

(ii)

$$\ln \frac{P(STA = 1|CPR)}{1 - P(STA = 1|CPR)} = \beta_0 + \beta_1 I(CPR = 1)$$

$$\Leftrightarrow \ln \frac{p_i}{1 - p_i} = \beta_0 + \beta_i, \text{ avec } \beta_2 = 0,$$

où $p_i = P(STA = 1|CPR = i)$, $i = 1, 2$.

Ce qui est équivalent à écrire

$$\ln \frac{p_2}{1 - p_2} = \beta_0, \text{ et } \ln \frac{p_1}{1 - p_1} = \beta_0 + \beta_1$$

(iii)

D'après les deux dernières équations, on peut estimer ces paramètres à l'aide de

$$\hat{\beta}_0 = \ln \frac{\hat{p}_2}{1 - \hat{p}_2}, \text{ et } \hat{\beta}_1 = \ln \frac{\hat{p}_1/(1 - \hat{p}_1)}{\hat{p}_2/(1 - \hat{p}_2)}$$

Les \hat{p}_i , $i = 1, 2$, sont donnés par

```
p <- prop.table(tab, 1)[, 2]
```

et voici les $\hat{\beta}_i$

```
o <- p/(1 - p)
or <- o[2]/o[1]
beta <- log(c(o[1], or))
names(beta) <- c("beta0", "beta1")
beta
```

```
## beta0 beta1
## 0.154 -1.695
```

(iv)

Un IC pour p_2 est donné par

```

cip2 <- prop.test(7, 7 + 6, conf.level = 0.95, correct = FALSE)$conf.int[1:2]
cip2

```

```
## [1] 0.291 0.768
```

⇒ un IC pour β_0 est donné par

```
log(cip2/(1 - cip2))
```

```
## [1] -0.888 1.197
```

Un IC pour $\log(OR) = \beta_1$ est donné par

```
log(or) + c(-1, 1) * qnorm(1 - 0.05/2) * sqrt(sum(1/tab))
```

```
## [1] -2.848 -0.541
```

```

# ou
confint(vcd::loddsratio(tab, log = TRUE))

```

	2.5 %	97.5 %
2:1/0:1	-2.85	-0.541

(v)

```

m <- glm(STA ~ CPR, data = data, family = binomial)
coef(summary(m))

```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.154	0.556	0.277	0.782
CPR1	-1.695	0.588	-2.880	0.004

```
confint.default(m)
```

	2.5 %	97.5 %
(Intercept)	-0.936	1.245
CPR1	-2.848	-0.541

On obtient exactement les mêmes estimations. Les intervalles de confiance (calculés à la main et à l'aide de `glm`) sont similaires, mais pas identiques puisque la fonction `glm` utilise la formule classique $\hat{\theta} \pm 1.96 \times \hat{\sigma}$.

Remarque:

Une autre façon d'estimer ce modèle est la suivante.

```
dtab <- data.frame(tab)
m <- glm(STA ~ CPR, weights = Freq, data = dtab, family = binomial)
coef(summary(m))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.154	0.556	0.277	0.782
CPR1	-1.695	0.588	-2.880	0.004

```
confint.default(m)
```

	2.5 %	97.5 %
(Intercept)	-0.936	1.245
CPR1	-2.848	-0.541

(b) Que deviennent les paramètres estimés du modèle si l'on change le niveau de référence pour CPR? Répondez **sans calculs supplémentaires** puis vérifiez à l'aide de R.

Solution:

Il suffit d'interchanger "1" et "2" dans les formules ci-dessus. Ainsi, en utilisant les mêmes notations qu'auparavant,

$$\tilde{\beta}_0 = \ln \frac{\hat{p}_1}{1 - \hat{p}_1} = \hat{\beta}_1 + \hat{\beta}_0 = -1.54, \text{ et } \tilde{\beta}_1 = \ln \frac{\hat{p}_2/(1 - \hat{p}_2)}{\hat{p}_1/(1 - \hat{p}_1)} = -\hat{\beta}_1 = 1.7$$

En effet,

```
data$CPR <- relevel(data$CPR, "1")
m <- glm(STA ~ CPR, data = data, family = binomial)
coef(summary(m))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.54	0.192	-8.03	0.000
CPR2	1.70	0.588	2.88	0.004

```
data$CPR <- relevel(data$CPR, "2") # On remet le niveau de référence à 2 pour la suite.
```

(c) Réalisez un tableau croisé entre les variables STA et RACE (RAC dans le fichier de données; 1 = white, 2 = black, 3 = other). Prenez $RAC = 1$ comme niveau de référence. Calculez les deux log-OR de votre table; vous devez utiliser les groupes de références tel qu'indiqué ci-dessus. Ajustez ensuite un modèle logit avec STA comme variable expliquée et RAC comme variable explicative. Commentez.

Solution:

```
str(data$RAC)
```

```
## int [1:200] 1 1 1 1 1 1 1 1 2 1 ...
```

```
data$RAC <- factor(data$RAC)
```

```
str(data$RAC)
```

```
## Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 2 1 ...
```

```
tab <- xtabs(~RAC + STA, data = data)
```

```
tab
```

RAC/STA	0	1
1	138	37
2	14	1
3	8	2

```
vcd::loddsrcatio(tab, log = TRUE, ref = c(1, 1))
```

```
## log odds ratios for RAC and STA
```

```
##
```

```
## 1:2 1:3
```

```
## -1.32 -0.07
```

```
m <- glm(STA ~ RAC, data = data, family = binomial)
```

```
coef(summary(m))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.32	0.185	-7.110	0.000
RAC2	-1.32	1.052	-1.258	0.208
RAC3	-0.07	0.812	-0.086	0.931

Nous observons à nouveau le lien entre les cotes et les β 's du modèle logistique.

Les résultats suggèrent que pour $RAC = 2$, la probabilité de décès est moindre par rapport à la référence ($RAC = 1$). La même remarque s'applique à $RAC = 3$ mais la différence (par rapport à la référence) est plus faible. Notez que les p-valeurs dépassent largement le seuil de 5%, ce qui indique que les diminutions (par rapport à la référence) sont statistiquement non significatives (à 5%).

(d) Utilisez le modèle ajusté au point précédent pour tester l'indépendance entre les variables STA et RAC.

Solution:

```
drop1(m, test = "Rao")
```

```
## Single term deletions
##
## Model:
## STA ~ RAC
##           Df Deviance AIC Rao score Pr(>Chi)
## <none>           198 204
## RAC      2         200 202      1.81      0.4
```

à 95%, on conclut à l'indépendance entre ces deux variables. On aboutit à la même conclusion avec un test classique (de Pearson)

```
summary(tab)
```

```
## Call: xtabs(formula = ~RAC + STA, data = data)
## Number of cases in table: 200
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 1.8, df = 2, p-value = 0.4
##  Chi-squared approximation may be incorrect
```

(e) Ajustez un modèle logit avec STA comme variable expliquée et CRN et AGE comme variables explicatives. Prenez $CRN = 2$ comme référence et incluez l'interaction entre CRN et AGE dans votre modèle. Cette interaction est-elle significative ? Rectifier le modèle si besoin. Utilisez le modèle rectifié pour calculer la probabilité de mourir pour une personne âgée de 30 ans dont $CRN = 2$.

Solution:

```
str(data$CRN)
```

```
## int [1:200] 1 1 1 1 1 1 1 1 1 1 ...
```

```
data$CRN <- factor(data$CRN, levels = c("2", "1"))
str(data$CRN)
```

```
## Factor w/ 2 levels "2","1": 2 2 2 2 2 2 2 2 2 2 ...
```

```
m <- glm(STA ~ CRN * AGE, data = data, family = binomial)
coef(summary(m))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.275	2.191	0.126	0.900
CRN1	-3.573	2.322	-1.539	0.124
AGE	-0.009	0.032	-0.277	0.782
CRN1:AGE	0.038	0.034	1.118	0.263

La p-valeur du test de Wald (visible dans le summary) suggère que l'interaction n'est pas significative. Simplifions dès lors le modèle.


```
m <- glm(STA ~ CRN + AGE, data = data, family = binomial())
coef(summary(m))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-2.010	0.870	-2.31	0.021
CRN1	-1.020	0.515	-1.98	0.048
AGE	0.025	0.011	2.33	0.020

Nous pouvons utiliser la fonction `predict()` pour calculer la probabilité demandée:

```
predict(m, data.frame(CRN = "2", AGE = 30), type = "response")
```

```
##      1
## 0.221
```

Partie 3

En plus de la variable `AGE`, nous souhaitons ici expliquer la variable `STA` à l'aide des variables `CPR`, `CAN`, `INF`, ainsi que la variable `RAC` qu'on vous demande de *recoder* de façon à ce qu'elle soit dichotomique (1 = white, 0 = black or other); prenez "0" comme référence. Pour `CPR`, `CAN` et `INF`, prenez "2" comme niveau de référence.

(a) Écrivez à la main l'équation complète d'un modèle logistique, sans interactions, incluant les variables citées ci-dessus et estimez les paramètres de ce modèle.

Solution:

Soit `RACR` la variable dichotomique avec "1" = "white", "0" = "black or other". L'équation d'un modèle logistique sans interaction avec cette variable et les autres est

$$\log \frac{P(STA = 1 | AGE, CAN, CPR, INF, RACR)}{1 - P(STA = 1 | AGE, CAN, CPR, INF, RACR)} \\ = \beta_0 + \beta_1 AGE + \beta_2 I(CAN = "1") + \beta_3 I(CPR = "1") + \beta_4 I(INF = "1") + \beta_5 I(RACR = "1")$$

Estimons ce modèle.

```
data$RACR <- data$RAC
levels(data$RACR) <- list(`0` = c("2", "3"), `1` = "1")
data$CAN <- relevel(factor(data$CAN), "2")
data$INF <- relevel(factor(data$INF), "2")
m1 <- glm(STA ~ AGE + CAN + CPR + INF + RACR, data, family = binomial)
coef(summary(m1))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.301	1.125	-1.157	0.247
AGE	0.027	0.012	2.331	0.020
CAN1	-0.195	0.612	-0.318	0.750

	Estimate	Std. Error	z value	Pr(> z)
CPR1	-1.652	0.620	-2.665	0.008
INF1	-0.689	0.379	-1.817	0.069
RACR1	0.329	0.683	0.481	0.630

(b) Utilisez les fonction `logLik()` et `pchisq()` pour réaliser un test LR pour tester le modèle actuel (H_1) versus un modèle avec seulement l'intercept (H_0). Vous ne devez utiliser ni la fonction `anova()` ni la fonction `drop1()`. Écrivez explicitement vos hypothèses H_0 et H_1 . Que concluez-vous ?

Solution:

$$H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0 \quad \text{vs} \quad H_1 : \beta_1 \neq 0 \text{ ou } \beta_2 \neq 0 \text{ ou } \beta_3 \neq 0 \text{ ou } \beta_4 \neq 0 \text{ ou } \beta_5 \neq 0$$

```
m0 <- glm(STA ~ 1, data = data, family = binomial())
lr <- -2 * (logLik(m0) - logLik(m1))
pval <- pchisq(lr, df = 5, lower.tail = FALSE)
pval
```

```
## 'log Lik.' 0.00125 (df=1)
```

⇒ Au moins une des variables explicatives a un effet significatif sur STA.

Remarque: Avec `anova` cela donne

```
anova(m0, m1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: STA ~ 1
## Model 2: STA ~ AGE + CAN + CPR + INF + RACR
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      199      200
## 2      194      180  5      20    0.0013 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Simplifiez le modèle actuel en supprimant toutes les variables non-significatives à 5%. Effectuez cette simplification étape par étape en supprimant une seule variable à la fois. Écrivez l'équation de votre modèle ainsi estimé.

Solution:

On commence par supprimer CAN puisque c'est la variable avec la plus grande p-valeur supérieure à 5%.

```
m2 <- glm(STA ~ AGE + CPR + INF + RACR, data, family = binomial)
coef(summary(m2))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.466	0.998	-1.469	0.142
AGE	0.027	0.012	2.322	0.020
CPR1	-1.647	0.620	-2.657	0.008
INF1	-0.683	0.379	-1.804	0.071
RACR1	0.333	0.682	0.488	0.625

```
# puis RACR
m3 <- glm(STA ~ AGE + CPR + INF, data, family = binomial)
coef(summary(m3))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.248	0.885	-1.41	0.158
AGE	0.028	0.011	2.46	0.014
CPR1	-1.631	0.616	-2.65	0.008
INF1	-0.697	0.378	-1.85	0.065

```
# puis INF
m4 <- glm(STA ~ AGE + CPR, data, family = binomial)
coef(summary(m4))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.57	0.862	-1.82	0.069
AGE	0.03	0.011	2.66	0.008
CPR1	-1.78	0.607	-2.94	0.003

$$\log \frac{\hat{P}(STA = 1|AGE, CAN, CPR, INF, RACR)}{1 - \hat{P}(STA = 1|AGE, CAN, CPR, INF, RACR)} = -1.57 + 0.03AGE - 1.78I(CPR = "1")$$

(d) Maintenant que vous n'avez que des variables explicatives significatives, complétez le modèle ainsi construit en ajoutant toutes les interactions. Peut-on simplifier ce dernier? Utilisez la BIC pour répondre à cette dernière question.

Solution:

```
m5 <- glm(STA ~ AGE * CPR, data, family = binomial)
BIC(m5, m4)
```

	df	BIC
m5	4	202
m4	3	200

```
# ou
drop1(m5, k = log(200))
```

	Df	Deviance	AIC
	NA	181	202
AGE:CPR	1	184	200

→ on supprime l'interaction $AGE \times CPR$. Notez qu'au niveau $\alpha = 5\%$, cette interaction n'est pas significative.

```
coef(summary(m5))
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-6.765	4.150	-1.630	0.103
AGE	0.119	0.070	1.701	0.089
CPR1	3.723	4.215	0.883	0.377
AGE:CPR1	-0.094	0.071	-1.330	0.183

Par la suite nous continuons avec le modèle (m4).

(e) Utilisez le modèle que vous avez choisi pour prédire la probabilité **de survie** pour deux patients avec $AGE = 25$ et $AGE = 80$ et un $CPR = "1"$. Même question avec un $CPR = "2"$. Accompagnez vos calculs des des intervalles de confiance à 95%.

Solution:

```
nwdt <- data.frame(AGE = c(25, 80, 25, 80), CPR = c("1", "1",
  "2", "2"))
prm <- predict(m4, new = nwdt, se = TRUE, type = "response")
fit <- prm$fit
lwr <- prm$fit - 1.96 * prm$se.fit
upr <- prm$fit + 1.96 * prm$se.fit
data.frame(nwdt, fit = 1 - fit, lwr = 1 - upr, upr = 1 - lwr)
```

AGE	CPR	fit	lwr	upr
25	1	0.932	0.871	0.992
80	1	0.728	0.626	0.829
25	2	0.696	0.414	0.977
80	2	0.310	0.050	0.570

(f) Quel est, selon le modèle que vous avez choisi, l'effet de l'âge sur la probabilité de mourir? Répondez à l'aide d'un **graphique** adéquat. Même question cette fois-ci pour l'effet de la variable CPR.

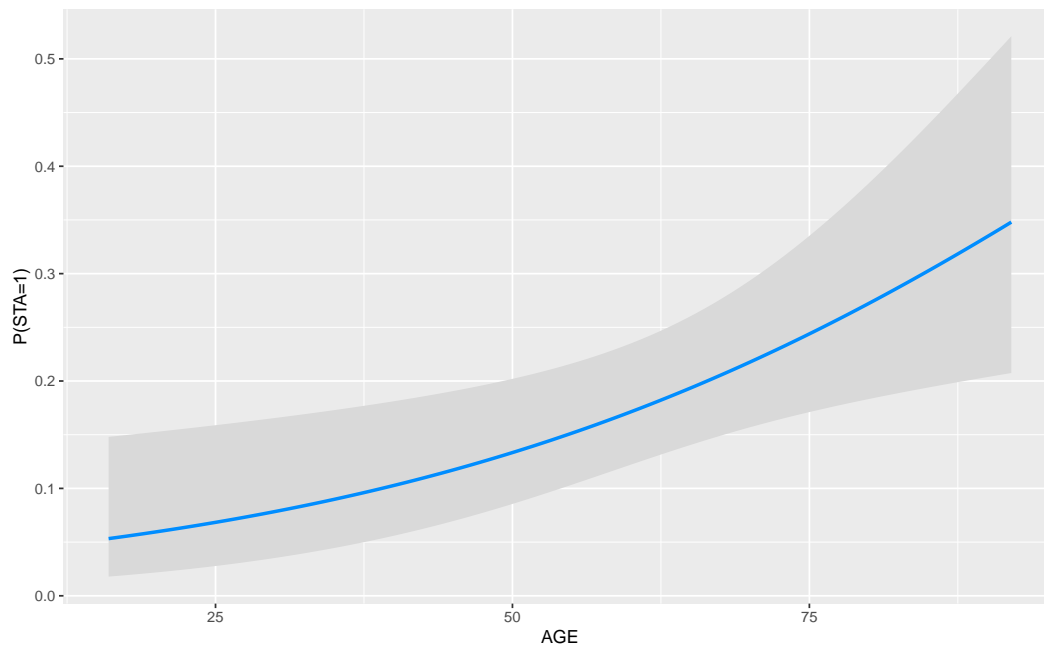
Solution:

Dans le modèle (4), la variable AGE a un effet positif. Cet effet, qu'on peut quantifier par

$$\exp(0.0296) = 1.03 = \frac{o(STA = 1|AGE = x + 1, CPR)}{o(STA = 1|AGE = x, CPR)}$$

On peut visualiser cela facilement à l'aide de `visreg`.

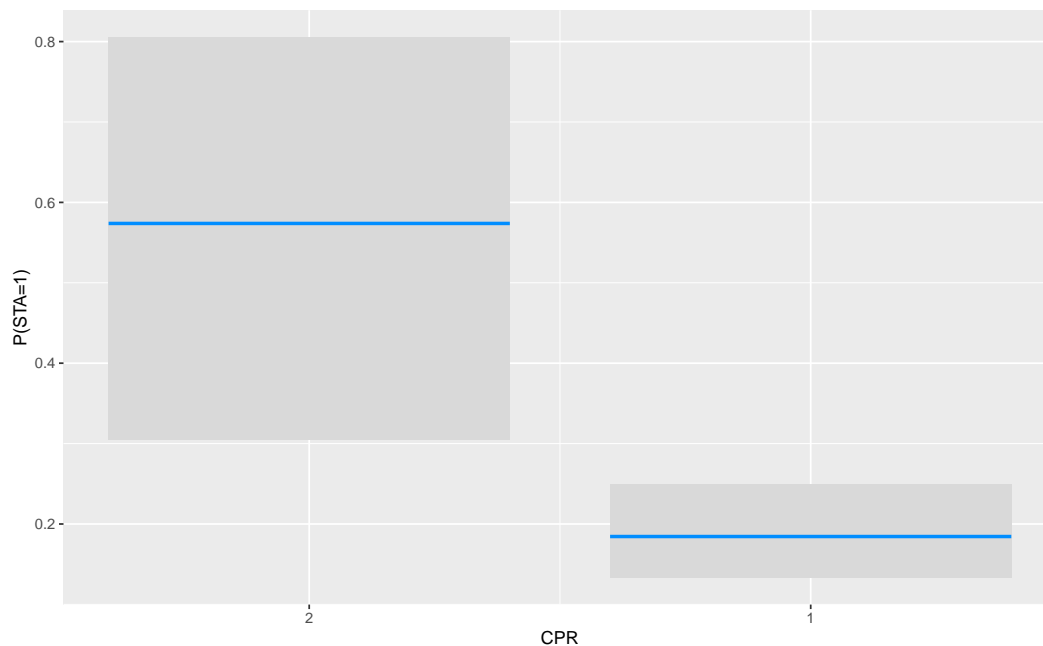
```
visreg(m4, "AGE", scale = "response", overlay = TRUE, rug = FALSE,  
       gg = TRUE) + labs(y = "P(STA=1)")
```



Pour CPR,

$$\exp(-1.78) = 0.168 = \frac{o(STA = 1|AGE, CPR = "1")}{o(STA = 1|AGE, CPR = "2")}$$

```
visreg(m4, "CPR", scale = "response", overlay = TRUE, rug = FALSE,  
       gg = TRUE) + labs(y = "P(STA=1)")
```



On peut voir les deux effets dans un seul graphe.

```
visreg(m4, "AGE", by = "CPR", scale = "response", overlay = TRUE,
  rug = FALSE, band = FALSE, gg = TRUE) + labs(y = "P(STA=1)")
```

