LSTAT2100 - Exercices - Série 2

Solutions

Exercice 1

Dans cet exercice, nous allons reprendre un exercice de la séance précédente, et le réaliser **en utilisant les modèles log-linéaires**.

On s'intéresse à la couleur des yeux d'individus dans un pays. Pour $n = 300$ individus pris au hasard, on a obtenu :

bleu	marron	noir	vert
48	122	95	35

(a) Peut-on dire que toutes les couleurs sont équiprobables ?

Solution:

```
Couleur <- factor(c("bleu", "vert", "marron", "noir"))
Freq <- c(48, 35, 122, 95)
data <- data.frame(Couleur, Freq)

data$Couleur <- relevel(data$Couleur, "bleu") # Ceci n'est pas obligatoire.
fit1 <- glm(Freq ~ 1, data = data, family = poisson())
fit2 <- glm(Freq ~ Couleur, data = data, family = poisson())
anova(fit1, fit2, test = "Rao")
```

```
## Analysis of Deviance Table
##
## Model 1: Freq ~ 1
## Model 2: Freq ~ Couleur
##   Resid. Df Resid. Dev Df Deviance  Rao Pr(>Chi)
## 1         3       67.4
## 2         0        0.0  3     67.4 65.8  3.3e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nous rejetons l'hypothèse selon laquelle les couleurs sont équiprobables.

(b) Peut-on dire que les yeux foncés (marron et noir) sont deux fois plus probables que les yeux clairs (bleu et vert) ?

Solution:

Nous pouvons créer une nouvelle variable binaire **Clair** qui prend la valeur 2 si les yeux sont bleus ou verts, et qui prend la valeur 1 sinon.

```
data$Clair <- data$Couleur
levels(data$Clair) <- list("1" = c("marron", "noir"), "2" = c("bleu", "vert"))
data$Clair <- relevel(data$Clair, "2")
```

Nous réalisons ensuite un modèle log-linéaire avec cette nouvelle variable comme variable explicative.

```
fit <- glm(Freq ~ Clair, data = data, family = poisson())
```

Le modèle log-linéaire stipule que

$$\mu_2 = \exp(\beta_0) \text{ (yeux clairs)}$$
$$\mu_1 = \exp(\beta_0 + \beta_1) \text{ (yeux foncés).}$$

Notons que nous cherchons à tester si $\mu_1 = 2\mu_2 \iff \exp(\beta_1) = 2 \iff \beta_1 = \log(2)$.

Il s'agit donc d'effectuer le test suivant.

$$H_0 : \beta_1 = \log(2) \text{ vs } H_1 : \beta_1 \neq \log(2)$$

Pour cela, le plus simple est d'utiliser un intervalle de confiance:

```
IC <- confint(fit)[2, ]
IC
```

```
##  2.5 % 97.5 %
##  0.713  1.219
```

Comme $log(2) = 0.693 \notin IC$, nous rejetons l'hypothèse nulle, au niveau 5%, et nous pouvons dès lors affirmer que les yeux foncés ne sont pas deux fois plus probables que les yeux clairs.

Une méthode alternative (et plus directe) est d'utiliser la fonction `wald.test` du package `aod` pour réaliser le test (une fois le package installé, regarder l'aide de la fonction `wald.test` et en particulier son argument `H0`).

```
require(aod)
wald.test(vcov(fit), coef(fit), Terms = 2, H0 = log(2))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 4.3, df = 1, P(> X2) = 0.038
```

(c) Peut-on dire que les yeux bleus et verts sont équiprobables ?

Solution:

Comme on a prit, dans la quetsion (a), "bleue" comme niveau de référence, nous pouvons directement tester cette hypothèse en testant si le coefficient correspondant à la couleur verte est zéro. (Si ni "bleue" ni "verte" n'était la référence, alors le plus simple, pour répondre à cette question, est de changer le niveau de référence à l'aide de la fonction `relevel`.)

En conséquence, il suffit de regarder le `summary` du `fit2`.

```
coef(summary(fit2))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.871	0.144	26.82	0.000
Couleurmarron	0.933	0.170	5.47	0.000
Couleurnoir	0.683	0.177	3.85	0.000
Couleurvert	-0.316	0.222	-1.42	0.155

Comme dans le cadre des tables de contingences (TP1), nous ne rejetons pas l'hypothèse que les deux couleurs sont équiprobables.

Exercice 2

Dans cet exercice, nous allons utiliser le jeu de données data.csv. Ce jeu de données comprend des observations liées à des clients qui ont contracté un crédit. Nous avons à notre disposition un certain nombre de variables, dont la variable Y qui nous dit si le client a pu rembourser dans les temps le crédit ($Y = 1$) ou pas ($Y = 2$) et la variable *tel* nous renseigne si la personne a un numéro de téléphone (A191 : Non, A192 : Oui).

Charger les données avec la commande suivante (cette commande suppose que votre répertoire de travail contient un répertoire `data` qui contient le fichier `data.csv`).

```
mdata <- read.csv(file = "data/data.csv", stringsAsFactors = TRUE)
# cette dernière option permet d'automatiquement transformer toute
# variable de type caractère en un facteur
```

(a) Ajustez un modèle log-linéaire saturé pour modéliser la table de contingence $tel \times Y$. Construisez des IC à 95% pour les différents paramètres. Pouvez-vous reconstruire la table de contingence à partir des coefficients estimés ? Que valent les résidus pour ce modèle ?

Solution:

Préparons d'abord les données et définissons les niveaux de références.

```
mdata$Y <- factor(mdata$Y)
levels(mdata$Y) <- c("PasDefaut", "Defaut")
levels(mdata$tel) <- c("Non", "Oui")
mdata$tel <- relevel(mdata$tel, "Oui")

tab <- xtabs(~ tel + Y, data = mdata)
tab
```

tel/Y	PasDefaut	Defaut
Oui	211	92
Non	305	142

```
data <- data.frame(tab)
data
```

tel	Y	Freq
Oui	PasDefaut	211
Non	PasDefaut	305
Oui	Defaut	92
Non	Defaut	142

Nous pouvons à présent utiliser la fonction **glm**.

```
fit <- glm(Freq ~ tel + Y + tel:Y, data = data, family = poisson())
# ou de façon identique
fit <- glm(Freq ~ tel * Y, data = data, family = poisson())
fit
```

```
##
## Call:  glm(formula = Freq ~ tel * Y, family = poisson(), data = data)
##
## Coefficients:
##    (Intercept)          telNon         YDefaut  telNon:YDefaut
##         5.3519          0.3685         -0.8301          0.0656
##
## Degrees of Freedom: 3 Total (i.e. Null);  0 Residual
## Null Deviance:       137
## Residual Deviance: 3.18e-14  AIC: 35.9
```

Nous pouvons construire des intervalles de confiance à l'aide de la fonction `confint`:

```
confint(fit)
```

	2.5 %	97.5 %
(Intercept)	5.214	5.484
telNon	0.194	0.545
YDefaut	-1.079	-0.589
telNon:YDefaut	-0.249	0.383

Et voici le tableau de contingence qu'on obtient à partir des coefficients estimés.

	Y =Defaut	YPasDefaut
$tel = Oui$	`exp(coef(fit)[1]+coef(fit)[3])`= 92	`exp(coef(fit)[1])`= 211

	Y =Defaut	YPasDefaut
$tel = Non$	`exp(sum(coef(fit)))`= 142	`exp(coef(fit)[1]+coef(fit)[2])`= 305

Mais le plus simple est d'utilser la fonction `fitted`:

```
fitted(fit)
```

```
##   1   2   3   4
## 211 305  92 142
```

Comme notre modèle est saturé (4 paramètres pour 4 observations), il 'colle' parfaitement aux données et nous voyons que les valeurs prédites sont identiques aux valeurs observées. Cela se traduit par une deviance nulle (R la calcule numériquement, c'est pourquoi on n'obtient pas exactement 0.)

```
deviance(fit)
```

```
## [1] 3.18e-14
```

Et les résidus sont nuls aussi.

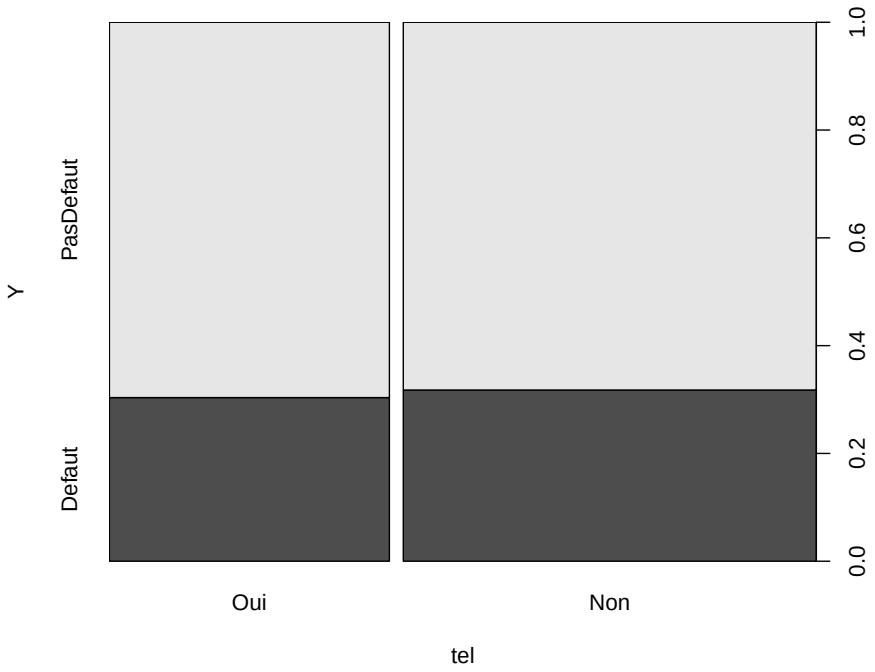
```
resid(fit)
```

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

(b) Visualisez graphiquement $\hat{p}(Y|tel)$. Que suggère votre graphique quand à l'association entre Y et tel ?

Solution:

```
spineplot(tab)
```



Nous voyons un alignement presque parfait entre la partie gauche et droite de ce graphique. Cela illustrant le fait que la distribution de `Y` ne dépend pas de la variable `tel`.

(c) Utilisez le modèle log-linéaire ajusté précédemment pour tester l'indépendance entre Y et *tel*. Proposez deux approches différentes, dont une ne nécessitant aucun calcul ou ajustement supplémentaire. Que peut-on conclure ?

Solution:

Nous remarquons que l'intervalle de confiance pour le terme d'interaction contient 0. Dès lors, nous ne pouvons pas rejeter le fait que ce terme d'interaction est égal à 0. Nous pouvons également voir cela en inspectant le `summary` du modèle et en y observant la p-valeur correspondante.

```
coef(summary(fit))
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.352	0.069	77.740	0.000
telNon	0.368	0.090	4.115	0.000
YDefaut	-0.830	0.125	-6.644	0.000
telNon:YDefaut	0.066	0.161	0.407	0.684

Une autre façon de répondre à cette question est d'effectuer un test de Score. Pour cela on peut soit utiliser la fonction `anova` ou `drop1`:

```r
fit0 <- glm(Freq ~ tel + Y, data = data, family = poisson())
anova(fit0, fit, test = "Rao")
```

```
## Analysis of Deviance Table
##
## Model 1: Freq ~ tel + Y
## Model 2: Freq ~ tel * Y
##   Resid. Df Resid. Dev Df Deviance   Rao Pr(>Chi)
## 1         1      0.166
## 2         0      0.000  1    0.166 0.166     0.68
```

```r
drop1(fit, test = "Rao")
```

```
## Single term deletions
##
## Model:
## Freq ~ tel * Y
##        Df Deviance  AIC Rao score Pr(>Chi)
## <none>       0.000 35.9
## tel:Y   1    0.166 34.1     0.166     0.68
```

Notons que nous aurions obtenu le même résultat en faisant un test chi-carré de Pearson sur la table de contingence:

```r
chisq.test(tab, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 0.2, df = 1, p-value = 0.7
```

Dans ces tests, nous échouons à rejeter l'hypothèse d'indépendance entre tel et Y.

Exercice 3

Dans cet exercice, nous allons utiliser le même jeu de données (**data.csv**) que celui utilisé dans l'Exercice 2. Nous considérons ici les variables *check_ac* et *employm*.

- *check_ac* représente le montant sur le compte à vue du client selon 4 modalités :
  - A11 : < 0
  - A12 : $[0; 200[$
  - A13 : ≥ 200
  - A14 : pas de compte.
- *employm* donne des informations sur la situation professionnelle du client :
  - A71 : sans emploi
  - A72 : < 1 an
  - A73 : $[1; 4[$
  - A74 :$[4; 7[$
  - A75 : ≥ 7 ans

(a) Créez une nouvelle vriable *check_ac1* qui regroupe les modailités "A11" et "A14" de la vraible *check_ac* en une seule modalité nommée "A11" et les modailités "A12" et "A13" en une seule modalité nommée "A12". Réalisez une table de contingence pour les vraibles *employm* et *check_ac1*. Puis construisez un modèle log-linéaire saturé sur vos données en choisissant "A11" comme référence.

Solution:

Préparons d'abord les données et définissons les niveaux de références.

```
mdata$check_ac1 <- mdata$check_ac
levels(mdata$check_ac1) <- list("A11" = c("A11", "A14"), "A12" = c("A12", "A13"))
mdata$check_ac1 <- relevel(mdata$check_ac1, "A11")
mdata$employm <- relevel(mdata$employm, "A71")
tab <- xtabs(~ check_ac1 + employm, data = mdata)
tab
```

check_ac1/employm	A71	A72	A73	A74	A75
A11	28	77	175	84	147
A12	16	55	81	37	50

```
data <- data.frame(tab)
```

Nous pouvons à présent utiliser la fonction **glm** pour ajuster un modèle log-linéaire.

```
fit <- glm(Freq ~ check_ac1 * employm, data = data, family = poisson())
fit
```

```
##
## Call:  glm(formula = Freq ~ check_ac1 * employm, family = poisson(),
##     data = data)
##
## Coefficients:
##             (Intercept)             check_ac1A12               employmA72
##                   3.332                   -0.560                    1.012
##              employmA73               employmA74               employmA75
##                   1.833                    1.099                    1.658
## check_ac1A12:employmA72 check_ac1A12:employmA73 check_ac1A12:employmA74
##                   0.223                   -0.211                   -0.260
## check_ac1A12:employmA75
##                  -0.519
##
## Degrees of Freedom: 9 Total (i.e. Null);  0 Residual
## Null Deviance:       298
## Residual Deviance: 1.58e-14  AIC: 79.4
```

(b) Certains termes d'interaction sont-ils non significatifs ? Si oui, combien ? À partir de ces termes, que pouvez-vous conclure concernant l'association entre ces deux variables (il n'est pas demandé d'effectuer un calcul supplémentaire) ? Que signifie la p-valeur qui figure dans la ligne "check_ac1A12" du "summary" de votre modèle.

Solution:

```
summary(fit)
```

```
##
## Call:
## glm(formula = Freq ~ check_ac1 * employm, family = poisson(),
##     data = data)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  3.332      0.189   17.63  < 2e-16 ***
## check_ac1A12                -0.560      0.313   -1.79    0.074 .
## employmA72                   1.012      0.221    4.58  4.6e-06 ***
## employmA73                   1.833      0.204    9.00  < 2e-16 ***
## employmA74                   1.099      0.218    5.03  4.8e-07 ***
## employmA75                   1.658      0.206    8.04  8.8e-16 ***
## check_ac1A12:employmA72      0.223      0.360    0.62    0.535
## check_ac1A12:employmA73     -0.211      0.341   -0.62    0.537
## check_ac1A12:employmA74     -0.260      0.370   -0.70    0.482
## check_ac1A12:employmA75     -0.519      0.354   -1.47    0.142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2.9840e+02  on 9  degrees of freedom
## Residual deviance: 1.5765e-14  on 0  degrees of freedom
## AIC: 79.37
##
## Number of Fisher Scoring iterations: 3
```

Tous les termes d'interactions n'apparaissent pas comme étant significativement différents de 0. Néanmoins, ceux-ci ne sont que des tests marginaux. Dans la suite, nous allons tester la nullité de ces termes d'interactions simultanément.

Si on note par N la variable aléatoire qui représente les fréquences absolues ($Freq$). La p-valeur qui figure dans la ligne "check_ac1A12" correspond alors au test suivant.

$$H_0 : E(N|check\_ac1 = "A12", employm = "A71") = E(N|check\_ac1 = "A11", employm = "A71")$$

ou, en terme de probabilité,

$$H_0 : P(check\_ac1 = "A12"|employm = "A71") = P(check\_ac1 = "A11"|employm = "A71")$$

(c) Utilisez le modèle ajusté pour tester l'indépendance entre ces deux variables.

Solution:

```
fit0 <- glm(Freq ~ check_ac1 + employm, data = data, family = poisson())
anova(fit0, fit, test = "Rao")
```

```
## Analysis of Deviance Table
##
## Model 1: Freq ~ check_ac1 + employm
## Model 2: Freq ~ check_ac1 * employm
##   Resid. Df Resid. Dev Df Deviance  Rao Pr(>Chi)
## 1         4       10.1
## 2         0        0.0  4     10.1 10.2    0.038 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ou, simplement,

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

```
## Single term deletions
##
## Model:
## Freq ~ check_ac1 * employm
##                   Df Deviance  AIC Rao score Pr(>Chi)
## <none>                     0.0 79.4
## check_ac1:employm  4     10.1 81.4      10.2    0.038 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Au niveau 5%, nous rejetons l'indépendance entre les deux variables.

(d) Indépendamment du résultat du test précédent, construisez un modèle log-linéaire en considérant les deux variables indépendantes. Que signifie la p-valeur qui figure dans la ligne "check_ac1A12" du "summary" de votre modèle. Calculez le R^2 de ce modèle et comparez le au R^2 du modèle saturé.

Solution:

Voici le modèle d'indépendance (= fit0 défini plus haut).

```
summary(fit0)
```

```
##
## Call:
## glm(formula = Freq ~ check_ac1 + employm, family = poisson(),
##     data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.4005     0.1528   22.25  < 2e-16 ***
## check_ac1A12  -0.7599     0.0784   -9.70  < 2e-16 ***
## employmA72     1.0986     0.1741    6.31  2.8e-10 ***
## employmA73     1.7610     0.1632   10.79  < 2e-16 ***
## employmA74     1.0116     0.1760    5.75  9.1e-09 ***
## employmA75     1.4990     0.1667    8.99  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 298.397  on 9  degrees of freedom
## Residual deviance:  10.057  on 4  degrees of freedom
## AIC: 81.43
##
## Number of Fisher Scoring iterations: 4
```

Si on note par N la variable aléatoire qui représente les fréquences absolues ($Freq$). La p-valeur qui figure dans la ligne "check_ac1A12" correspond alors au test suivant.

$$H_0 : E(N|check\_ac1 = "A12") = E(N|check\_ac1 = "A11")$$

ou, en terme de probabilité,

$$H_0 : P(check\_ac1 = "A12") = P(check\_ac1 = "A11")$$

Ce test est réalisé ici en supposant l'indépendance entre les variables `employm` et `check_ac1` et donc en utilisant toutes les observations, quelle que soit la valeur de `employm`.

Le R^2 de ce modèle est

```
(1 - fit0$deviance / fit0$null.deviance) * 100
```

```
## [1] 96.6
```

alors que pour le modèle saturé $R^2 = 100\%$. En effet,

```
(1 - fit$deviance / fit$null.deviance) * 100
```

```
## [1] 100
```

Exercice 4

Dans cet exercice, nous allons utiliser le même jeu de données (**data.csv**) que celui utilisé dans l'Exercice 2. Nous considérons ici les variables *Y*, *tel* et *employm*.

(a) Réalisez un tableau de contingence et un mosaicplot impliquant ces trois variables. Est-ce qu'une association homogène entre ces trois variables est envisageable ? Il n'est pas demandé ici de réaliser un test.

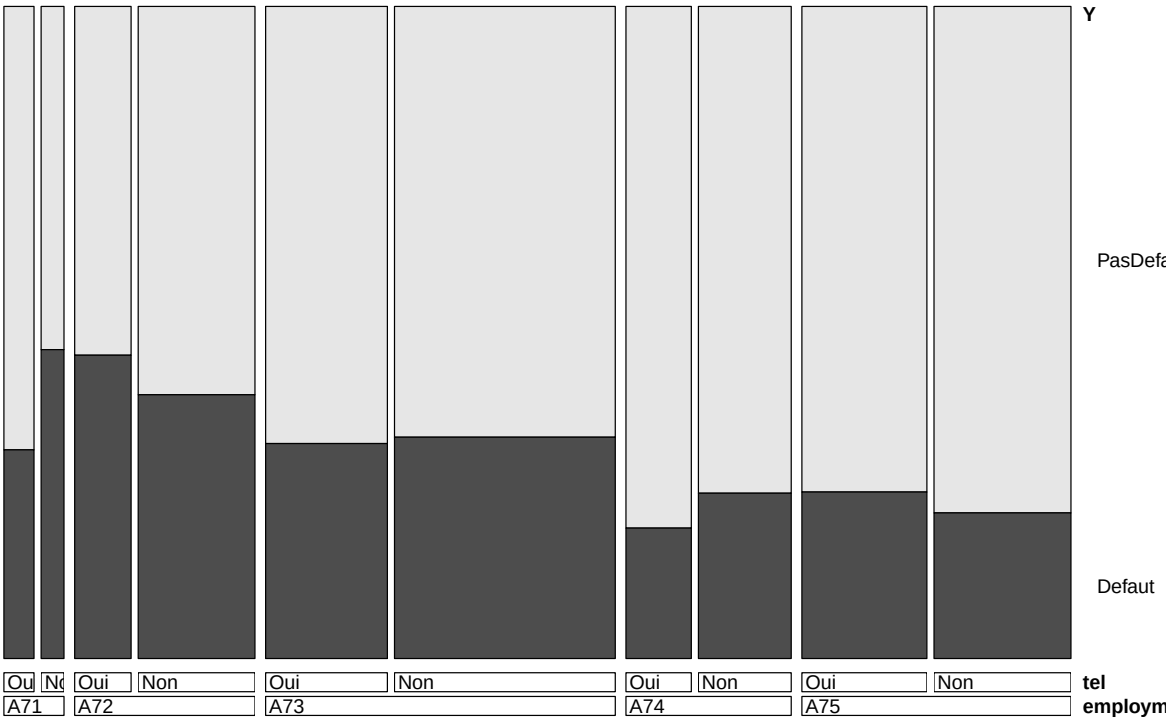
Solution:

```
tab <- xtabs(~ tel + Y + employm, data = mdata)
tab
```

tel	Y	employm	Freq
Non	Defaut	A71	9
		A72	36
		A73	56

tel	Y	employm	Freq
		A74	18
		A75	23
	PasDefaut	A71	10
		A72	53
		A73	109
		A74	53
		A75	80
Oui	Defaut	A71	8
		A72	20
		A73	30
		A74	10
		A75	24
	PasDefaut	A71	17
		A72	23
		A73	61
		A74	40
		A75	70

```
doubledecker(Y ~ employm + tel, tab)
```



Dans le graphique ci-dessus, lorsqu'on compare les blocs "A71", "A72",. . . ,"A75", mis à part "A72" et "A75", on constate, globalement, une certaine similitude dans la structure, mais il est difficile d'affirmer qu'il s'agit bien d'une association homogène. Pour voir plus clair, on peut calculer les rapports de cotes conditionnelles.

```r
loddsratio(tab, log = FALSE)
```

```
##  odds ratios for tel and Y by employm
##
##   A71   A72   A73   A74   A75
## 1.913 0.781 1.045 1.358 0.839
```

Ces chiffres confirment nos constatations, mais nous ne permettent pas de conclure.

(b) Confirmer ou infirmer votre réponse (a) à l'aide d'un test adéquat.

Solution:

```r
data <- data.frame(tab)
MS <- glm(Freq ~ Y * employm * tel, data = data, family = poisson())
drop1(MS, test = "Rao")
```

```
## Single term deletions
##
## Model:
## Freq ~ Y * employm * tel
##               Df Deviance AIC Rao score Pr(>Chi)
## <none>              0.00 144
## Y:employm:tel  4     2.28 138      2.28     0.68
```

Avec une p-valeur de presque 0.7, l'association homogène est "acceptée", au niveau 5%.

(c) Construisez un modèle log-linéaire saturé et simplifiez-le le plus possible à l'aide des tests entre modèles emboîtés. Refaites la même analyse en utilisant l'AIC ou la BIC.

Solution:

Méthode basée sur les tests :

```r
# étape 1
drop1(MS, test = "Rao")
```

```
## Single term deletions
##
## Model:
## Freq ~ Y * employm * tel
##               Df Deviance AIC Rao score Pr(>Chi)
## <none>              0.00 144
## Y:employm:tel  4     2.28 138      2.28     0.68
```

```r
# étape 2 (Voir l'output  ci-dessus; colonne Pr(>Chi))
MH <- update(MS, . ~ . - Y:employm:tel)
drop1(MH, test = "Rao")
```

```
## Single term deletions
##
## Model:
## Freq ~ Y + employm + tel + Y:employm + Y:tel + employm:tel
##             Df Deviance AIC Rao score Pr(>Chi)
## <none>            2.28 138
## Y:employm    4    20.30 148     18.01   0.0012 **
## Y:tel        1     2.29 136      0.01   0.9385
## employm:tel  4    17.27 145     15.06   0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# étape 3 (Voir l'output  ci-dessus; colonne Pr(>Chi))
MC1 <- update(MH, . ~ . - Y:tel)
drop1(MC1, test = "Rao")
```

```
## Single term deletions
##
## Model:
## Freq ~ Y + employm + tel + Y:employm + employm:tel
##             Df Deviance AIC Rao score Pr(>Chi)
## <none>            2.29 136
## Y:employm    4    20.47 146      18.2   0.0011 **
## employm:tel  4    17.43 144      15.2   0.0043 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Il n'y a plus de simplification possible. Le modèle retenu est $MC1$:

```
formula(MC1)
```

```
## Freq ~ Y + employm + tel + Y:employm + employm:tel
## <environment: 0x000001ae6109ad38>
```

c'est-à-dire le modèle d'indépendance conditionnelle entre "Y" et "tel" étant donné "employm".

Méthode basée sur l'AIC :

On aboutit à la même conclusion si on refait l'analyse ci-dessus en regardant la colonne "AIC" au lieu de regarder la colonne "Pr(>Chi)". Un moyen plus commode/rapide est de se servir de la fonction `step`.

```
maic <- step(MS, direction = "backward")
```

```
## Start:  AIC=144
## Freq ~ Y * employm * tel
##
##                 Df Deviance AIC
## - Y:employm:tel  4     2.28 138
## <none>                 0.00 144
##
## Step:  AIC=138
## Freq ~ Y + employm + tel + Y:employm + Y:tel + employm:tel
##
```

```
##               Df Deviance AIC
## - Y:tel        1     2.29 136
## <none>               2.28 138
## - employm:tel  4    17.27 145
## - Y:employm    4    20.30 148
##
## Step:  AIC=136
## Freq ~ Y + employm + tel + Y:employm + employm:tel
##
##               Df Deviance AIC
## <none>               2.29 136
## - employm:tel  4    17.43 144
## - Y:employm    4    20.47 146
```

```
formula(maic)
```

```
## Freq ~ Y + employm + tel + Y:employm + employm:tel
## <environment: 0x000001ae6109ad38>
```

Méthode basée sur le BIC :

On aboutit aussi à la même conclusion si on utilise la BIC. Pour vérifier cela, il suffit de faire tourner le code suivant (voir l'aide de la fonction `drop1` et son argument `k`):

```
n <- nrow(data)
mbic <- step(MS, k = log(n), direction = "backward")
```

```
## Start:  AIC=164
## Freq ~ Y * employm * tel
##
##                 Df Deviance AIC
## - Y:employm:tel  4     2.28 154
## <none>                 0.00 164
##
## Step:  AIC=154
## Freq ~ Y + employm + tel + Y:employm + Y:tel + employm:tel
##
##               Df Deviance AIC
## - Y:tel        1     2.29 151
## <none>               2.28 154
## - employm:tel  4    17.27 157
## - Y:employm    4    20.30 160
##
## Step:  AIC=151
## Freq ~ Y + employm + tel + Y:employm + employm:tel
##
##               Df Deviance AIC
## <none>               2.29 151
## - employm:tel  4    17.43 154
## - Y:employm    4    20.47 158
```

```
formula(mbic)
```

```
## Freq ~ Y + employm + tel + Y:employm + employm:tel
## <environment: 0x000001ae6109ad38>
```

Sachez que le fait d'aboutir à la même conclusion n'est, en général, pas la règle et qu'il est fréquent que des mèthodes de sélections différentes aboutissent à des modèles différents.

Exercice 5

Le but de cet exercice est de montrer que l'indépendance deux à deux n'implique pas l'indépendance mutuelle.

Pour cela, considérant l'expérience qui consistent à lancer deux pièces (indépendamment l'une de l'autre). On définit les variables suivantes:

- X la variable aléatoire binaire prenant la valeur 1 si *la première* pièce montre face et 0 sinon,
- Y la variable aléatoire binaire prenant la valeur 1 si *la deuxième* pièce montre face et 0 sinon.
- Z la variable aléatoire prenant la valeur 1 si les deux pièces montrent le même coté et 0 sinon; i.e., $Z = I(X = Y)$.

Les deux variables X et Y sont supposées être indépendantes.

(a) Montrer que X et Z sont indépendantes. On en déduit que X, Y et Z sont indépendantes deux à deux.

Solution:

Montrons que $P(X = 1, Z = 1) = P(X = 1) \times P(Z = 1)$.

Nous avons que

$$\begin{aligned} P(X = 1, Z = 1) &= P(X = 1, Y = X) \\ &= P(X = 1, Y = 1) \\ &= P(X = 1)\, P(Y = 1) \text{ par l'indépendance X et Y} \\ &= 0.5 \times 0.5 = 0.25 \end{aligned}$$

D'autre part,

$$\begin{aligned} P(X = 1) \times P(Z = 1) &= 0.5 \times (P(X = 1, Y = 1) + P(X = 0, Y = 0)) \\ &= 0.5 \times (0.5 \times 0.5 + 0.5 \times 0.5) = 0.25 \\ &= P(X = 1, Z = 1). \end{aligned}$$

De la même façon, nous pouvons facilement montrer tous les autres cas, à savoir que

$$P(X = i, Z = j) = P(X = i, Z = j),\ i, j = 0, 1$$

On en déduit que X et Z sont indépendantes. Et par symétrie, on peut aussi affirmer que Y et Z sont indépendantes. On en conclut l'indépendance deux à deux de ces trois variables.

(b) Montrer que X, Y et Z ne sont pas mutuellement indépendants.

Solution:

Il nous suffit de trouver un exemple qui contredit l'affirmation comme quoi ces variables sont mutuellement indépendantes. Prenons par exemple la probabilité suivante

$$\begin{aligned} P(X=1,Y=0,Z=0) &= P(X=1,Y=0) \\ &= P(X=1)\times P(Y=0) \\ &= 0.5\times 0.5 = 0.25. \end{aligned}$$

Alors que

$$\begin{aligned} P(X=1)\times P(Y=0)\times P(Z=0) &= 0.5^2\times P(Z=0) \\ &= 0.5^2\times (P(X=1)P(Y=0)+P(X=0)P(Y=1)) \\ &= (0.5)^3 = 0.125. \end{aligned}$$

Dès lors, $P(X=1,Y=0,Z=0)\neq P(X=1)\times P(Y=0)\times P(Z=0)$ et nous n'avons pas l'indépendance mutuelle entre X, Y et Z.

Exercice 6

(a) Montrer que

$$\text{(a) } X \perp\!\!\!\perp Y|Z \text{ et } \text{ (b) } X \perp\!\!\!\perp Z|Y \iff \text{ (c) } X \perp\!\!\!\perp (Y,Z)$$

Solution:

(i) Montrons que (c) $\implies$ (a).

Nous supposons donc (c) $p(x|y,z)=p(x)$ et nous devons prouver (a) $p(x|y,z)=p(x|z)$. Pour cela, il suffit de montrer que $p(x|z)=p(x)$. Ce qui est vrai. En effet,

$$\text{(c)} \iff p(x,y,z)=p(x)p(y,z) \implies \sum_y p(x,y,z)=\sum_y p(x)p(y,z) \implies p(x,z)=p(x)p(z).$$

(ii) Montrons que (c) $\implies$ (b).

Il suffit pour cela de suivre exactement la même démarche que celle utilisée pour prouver (i).

(iii) Montrons que (a) et (b) $\implies$ (c).

Nous supposons donc (a) $p(x|y,z)=p(x|z)$ et (b) $p(x|y,z)=p(x|y)$ et nous devons prouver (c) $p(x|y,z)=p(x)$. Pour cela, il suffit encore une fois de montrer que $p(x|z)=p(x)$. Ce qui est vrai. En effet,

$$\begin{aligned} \text{(a) et (b)} &\implies p(x|z)=p(x|y) \implies p(x,z)p(y)=p(x,y)p(z) \\ &\implies \sum_y p(x,z)p(y)=\sum_y p(x,y)p(z) \implies p(x,z)=p(x)p(z). \end{aligned}$$

(b) Montrer que

$$\text{(a) } X \perp\!\!\!\perp (Y,Z) \text{ et } \text{ (b) } Y \perp\!\!\!\perp (X,Z) \iff \text{ (c) } X \perp\!\!\!\perp Y \perp\!\!\!\perp Z$$

Solution:

(i) Montrons que (c) $\implies$ (a).

Nous supposons donc (c) $p(x,y,z) = p(x)p(y)p(z)$ et nous devons prouver (a) $p(x,y,z) = p(x)p(y,z)$. Pour cela, il suffit de montrer que $p(y,z) = p(y)p(z)$. Ce qui est vrai. En effet,

$$\text{(c)} \implies \sum_x p(x,y,z) = \sum_x p(x)p(y)p(z) \implies p(y,z) = p(y)p(z).$$

(ii) Montrons que (c) $\implies$ (b).

Il suffit pour cela de suivre exactement la même démarche que celle utilisée pour prouver (i).

(iii) Montrons que (a) et (b) $\implies$ (c).

Nous supposons donc (a) $p(x,y,z) = p(x)p(y,z)$ et (b) $p(x,y,z) = p(y)p(x,z)$ et nous devons prouver (c) $p(x,y,z) = p(x)p(y)p(z)$. Pour cela, il suffit encore une fois de montrer que $p(y,z) = p(y)p(z)$. Ce qui est vrai. En effet,

$$\text{(a) et (b)} \implies p(x)p(y,z) = p(y)p(x,z) \implies \sum_x p(x)p(y,z) = \sum_x p(y)p(x,z) \implies p(y,z) = p(y)p(z).$$