Modèles Linéaires & GLMs
Analyse Logit & Régression de Poisson
Analyse d'un Portefeuille d'Assurance
Algorithme IRWLS avec R

Julien TOMAS ♠ *

♠ *ISFA - Laboratoire SAF* †

Version 1.0

Ce document est basé sur :

- Kaas *et al.* (2008), *Modern Actuarial Risk Theory - Using R*, Chapitre 9, Berlin Heidelberg : Springer Verlag, Second édition.

*Contact: julien.tomas@univ-lyon1.fr.

†Institut de Science Financière et d'Assurances - Université Claude Bernard Lyon 1 - 50 Avenue Tony Garnier - 69366 Lyon - France.

Table des matières

1 Modèles linéaires et modèles linéaires généralisés

1.1 Bref rappel théorique

En statistiques, la régression est utilisée pour modéliser la relation entre les variables :

- ✔ une variable de réponse aléatoire Y, aussi appelée variable dépendante, variable expliquée ou variable ajustée,
- ✔ les prédicteurs $X_1, \ldots, X_p$, aussi appelés variables indépendantes, variables explicatives, variables de contrôle, données collatérales, covariables ou régresseurs. Habituellement, ceux-ci sont supposés non aléatoires et mesurables sans erreur.

Dans les applications actuarielles, une variable aléatoire symétrique normalement distribuée avec une variance fixe ne permet pas de décrire les situations correctement.

- ✔ En effet, dans la pratique, de nombreux types de données contiennent des erreurs non-normales ;
  1. Les erreurs peuvent être fortement asymétrique. Pour la taille des sinistres, la distribution ne présente pas de queue à gauche mais une queue droite significativement épaisse, et est donc asymétrique (comme une densité gamma)
  2. Les erreurs peuvent être plus aplaties (plus concentrée dans les queues de la distribution) que le distribution normale
  3. Les erreurs peuvent être strictement délimitées (comme pour les proportions)
  4. Les erreurs peuvent ne pas avoir de valeurs négatives (comme pour les données de comptage)
- ✔ Dans la pratique, les données présentent généralement une variance supérieure à la moyenne ;
- ✔ Souvent, les données montrent un coefficient de variation σ/μ constant plutôt qu'une variance constante ;
- ✔ Les phénomènes à modéliser sont rarement additifs. Un modèle multiplicatif peut être beaucoup plus plausible. Par exemple, le remplacement d'une voiture par une voiture de 200 kg plus légère, sans changer les autres caractéristiques de l'assuré, se traduira par une réduction du nombre moyen de sinistres par un pourcentage fixe et non par un montant fixe indépendant du risque original.

Par le passé, les seuls outils disponibles pour faire face à ces problèmes étaient la transformation de la variable dépendante ou l'adoption de méthodes non paramétriques. Si la variable dépendante n'est pas normale mais d'une autre famille de la classe exponentielle, alors tous ces problèmes peuvent être résolus en utilisant les modèles linéaires généralisés introduits par Nelder and Wedderburn (1972). En probabilités et statistiques, la famille exponentielle est une classe importante de distributions en raison de sa commodité mathématique, de ces propriétés algébriques mais aussi car ces distributions sont souvent naturelles à considérer :

- ✔ Les déviations aléatoires obéissent à autre distribution que normale :
  1. Les erreurs peuvent être Poisson, utiles pour les données de comptage
  2. Les erreurs peuvent être Binomiale, utiles avec les données sur les proportions
  3. Les erreurs peuvent être Gamma, utiles avec des données montrant un coefficient de variation constant
  4. Les erreurs peuvent être exponentielles, utiles avec des données de mortalité, comme dans l'analyse de survie

✔ La moyenne de la variable aléatoire est une fonction non-linéaire des variables explicatives. Elle ne doit être linéaire que sur une certaine échelle. Si cette échelle est logarithmique, on a en fait un modèle multiplicatif au lieu d'un modèle additif. Ceci est très souvent essentiel dans les applications d'assurance.

La normalité et la constance de la variance ne sont plus nécessaires. Comme le montre le tableau 1 ci-dessous, les hypothèses des GLMs sont moins restrictives. Néanmoins, une caractéristique importante est qu'ils supposent des observations indépendantes (ou au moins non corrélées). Cette hypothèse d'indépendance est une caractéristique du modèle linéaire de régression classique, et est portée sans modification aux modèles linéaires généralisés. Une seconde hypothèse sur la structure de l'erreur est l'existence d'un terme d'erreur unique dans le modèle.

Modèles linéaires	Modèles linéaires généralisés
• variables aléatoires $Y_1, ..., Y_n$	• variables aléatoires $Y_1, ..., Y_n$
• mutuellement indépendant	• mutuellement indépendant
• $Y_i \rightarrow Normale$	• la distribution de Y_i n'est pas nécessairement normale mais doit être dans la famille exponentielle
• $\mathbb{E}[Y_i] = \mu_i = x_i^{'}\beta$ et $\mathbb{V}[Y_i] = \sigma^2$	• $g(\mathbb{E}[Y_i]) = g(\mu_i) = x_i^{'}\beta$ ou $\mu_i = g^{-1}(x_i^{'}\beta)$

TABLE 1: Modèles linéaires généralisés vs. Modèles linéaires

Les modèles linéaires généralisés sont une extension des modèles linéaires classiques. Supposons que $Y_1, \ldots, Y_n$ sont des variables aléatoires normales indépendantes avec moyenne $\boldsymbol{\mu}$:

✔ La partie systématique du modèle spécifie le vecteur $\boldsymbol{\mu}$ en fonction d'un petit nombre de paramètres inconnus $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{p-1})^\top$,

$$\mathbb{E}\big[Y_i\big] = \mu_i = \beta_0 + \sum_{j=1}^{p-1} x_{ij}\,\beta_j$$

en notation matricielle

$$\underset{n \times 1}{\mathbb{E}\big[\boldsymbol{y}\big]} = \underset{n \times p}{\boldsymbol{X}}\underset{p \times 1}{\boldsymbol{\beta}} = \underset{n \times 1}{\boldsymbol{\mu}}$$

✔ La variance des erreurs est assumée constante et indépendante.

Les modèles linéaires généralisés possèdent trois caractéristiques :

1. Il y a une composante stochastique, qui précise que les observations sont des variables aléatoires indépendantes Y_i, $i = 1, \ldots, n$ avec une densité appartenant à la famille de dispersion exponentielle. Une distribution appartient à la famille de dispersion exponentielle si sa fonction de densité peut être écrite sous la forme :

$$f_Y(y_i; \theta_i, \phi) = \exp\left\{\frac{y_i\,\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\right\}$$

pour des fonctions spécifiques $a(\cdot)$, $b(\cdot)$ et $c(\cdot)$. Le paramètre ϕ est appelé paramètre de dispersion. Il s'agit d'un paramètre de nuisance ne dépendant pas de i. Les fonctions a et c sont telles que $a_i()\phi = \phi/\omega_i$ et $c = (y_i, \phi/\omega_i)$ ou ω_i est un poids connu pour chaque observation i.

Les exemples les plus importants sont présentés dans le tableau 2 ci dessous :

Distribution de Y_i	θ_i	ϕ	$a_i(\phi)$	$b(\theta_i)$	$c(y_i, \phi)$
Normale(μ_i ;σ^2)	μ_i	σ^2	ϕ	$\frac{\theta^2}{2}$	$-\frac{1}{2}\left\{\frac{y^2}{\sigma^2} + log(2\pi\sigma^2)\right\}$
Poisson(μ_i)	$\log(\mu_i)$	1	ϕ	$\exp(\theta_i)$	$-\log y!$
Binomiale$\frac{1}{m_i}$(m_i ;μ_i)	$\log\left(\frac{\mu_i}{1-\mu_i}\right)$	$\frac{1}{\mu_i}$	ϕ	$\log(1+\exp\theta_i)$	$\log\binom{m_i}{m_i y_i}$
Gamma(μ_i ;α)	$\frac{-1}{\mu_i}$	α^{-1}	ϕ	$-\log(-\theta)$	$\alpha\log(\alpha y) - \log y - \log\Gamma(\alpha)$
Inverse Gaussienne(μ_i ;σ^2)	$\frac{-1}{2\mu_i^2}$	σ^2	ϕ	$-(-2\theta)^{1/2}$	$-\frac{1}{2}\left\{\log(2\pi\phi y^3) + \frac{1}{\phi y}\right\}$

TABLE 2: Famille de dispersion exponentielle

2. La composante systématique du modèle attribue a chaque observation un prédicteur linéaire

$$\eta_i = \beta_0 + \sum_{j=1}^{p-1} x_{ij}\beta_j$$

3. Le troisième composant d'un GLM connecte les deux premiers éléments. L'espérance μ_i de Y_i est liée au prédicteur linéaire η_i par une fonction de lien

$$\eta_i = g\big(\mu_i\big) = \beta_0 + \sum_{j=1}^{p-1} x_{ij}\,\beta_j$$

La valeur de η est différente de celle de μ (à l'exception du cas ou la fonction de lien est l'identité). La valeur de η est obtenue en transformant μ par la fonction de lien,

$$g\big(\mathbb{E}\big[Y_i\big]\big) = g\big(\mu_i\big)$$

$\mathbb{E}\big[Y_i\big]$ est donc liée une combinaison linéaire des paramètres $\boldsymbol{\beta}$ par une fonction différentiable et monotone g, qui n'est pas nécessairement l'identité. Ls prédictions sont obtenues en appliquant l'inverse de la fonction de lien, g^{-1}, appelée la fonction de réponse. Les fonctions de lien les plus fréquemment utilisées sont indiquées dans le tableau 3 ci-dessous.

log	$g(\mu_i) = \log\mu_i$
logit	$g(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right)$
probit	$g(\mu_i) = \Phi^{-1}(\mu_i)$, où $\Phi(.)$ est la fdc $\mathcal{N}(0,1)$
complementary log-log	$g(\mu_i) = \log\left(-\log(1-\mu_i)\right)$
log-log	$g(\mu_i) = \log(-\log(\mu_i))$

TABLE 3: Exemple de fonctions de lien

Un critère important dans le choix de la fonction de lien est de s'assurer que les valeurs ajustées restent dans des limites raisonnables. Par exemple, nous voulons nous assurer que le taille des sinistres soit supérieur ou égal à 0 et, si la variable dépendante est une proportion alors les valeurs ajustées devraient se situer entre 0 et 1. Dans le premier cas, une fonction de lien log est appropriée parce que la valeur ajustée sont exponentielles du prédicteur linéaire, dans le second cas, le lien logit est appropriée parce que les valeurs ajustées sont calculées comme l'inverse de $\log\big(p/(1-p)\big)$.

La performance d'une variété de modèles peut être directement comparée. Comme la déviance totale est la même dans chaque cas, nous pouvons étudier les conséquences de la modification de nos hypothèses.

1.2 Exemple avec R

Nous finissons cette section par un exemple. Supposons que nous avons les données du nombre de voyages à l'étranger d'un groupe de w_i, $i = 1, \ldots, 16$ personnes pour lesquelles le genre et le revenu sont connus :

```
> w <- c(1,2,2,1,1,1,1,4,2,4,2,3,2,1,1,2)
> Y <- c(0,1,0,8,0,0,0,30,0,1,1,38,0,0,0,26) / w
> genre <- c(0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1)
> revenu <- c(1,2,5,20,1,2,5,20,1,2,5,20,1,2,5,20)
```

Nous ajustons un modèle linéaire ordinaire et un modèle linéaire généralisé avec une distribution de Poisson et une fonction de lien logarithmique, comme suit :

```
> mod.lm <- lm(Y ~ genre + revenu, weights=w)
> mod.glm <- glm(Y ~ genre + revenu, weights=w, family=poisson)
```

Le modèle linéaire conduit au modèle suivant

```
> mod.lm$coefficients
(Intercept)       genre      revenu
 -2.4547301   1.9623425   0.5765162
```

$Y = -2.45 + 1.96 \times \texttt{genre} + 0.58 \times \texttt{revenu}$

Pour le modèle linéaire généralisé, nous obtenons

```
> exp(mod.glm$coefficients)
(Intercept)       genre      revenu
 0.05271334  1.67902501  1.28226262
```

$Y = 0.053 \times 1.68^{\texttt{genre}} \times 0.58^{\texttt{revenu}}$

Le premier modèle conduit à des nombres de voyages ajustés négatifs pour des individus de faible revenu et de genre 0.

1.3 Exercice 1.1

Décrire les matrices de design X des deux ajustements précédents. Notez qu'une constante est incluse, donc la première colonne de X est une colonne de 1. Utiliser la fonction `model.matrix()`.

Les deux matrices de design sont identiques.

```
> model.matrix(mod.lm)
   (Intercept) genre revenu
1            1     0      1
2            1     0      2
3            1     0      5
4            1     0     20
5            1     0      1
```

```
6            1     0      2
7            1     0      5
8            1     0     20
9            1     1      1
10           1     1      2
11           1     1      5
12           1     1     20
13           1     1      1
14           1     1      2
15           1     1      5
16           1     1     20
attr(,"assign")
[1] 0 1 2
```

1.4 Exercice 1.2

Regarder si `vcov(mod.lm)` correspond à la matrice d'information de Fisher estimée par maximum de vraisemblance $\widehat{\sigma}^2\left(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}\right)^{-1}$. Utiliser :

```
> mean(mod.lm$residuals^2*w)*solve(t(model.matrix(mod.lm))
+ %*%diag(w)%*%model.matrix(mod.lm))/vcov(mod.lm)
```

Ici, `solve(A)` donne l'inverse de la matrice `A`, `t(A)` donne sa transposé, `diag(w)` est une matrice diagonale avec le vecteur `w` sur sa diagonale, l'opérateur `%*%` effectue une multiplication matricielle et `/` effectue une division élément par élément. A la place de $\widehat{\sigma}^2$, quelle quantité est calculée par R pour la matrice de variance covariance ?

Une fois le modèle linéaire ajusté, nous pouvons accéder aux résultats soit en tapant le nom de l'objet dans lequel les résultats ont été stockés, ici `mod.lm`, ou en utilisant des fonctions telles que `summary()`.

```
> mod.lm

Call:
lm(formula = Y ~ genre + revenu, weights = w)

Coefficients:
(Intercept)        genre       revenu
    -2.4547       1.9623       0.5765
```

Nous pouvons aussi utiliser la fonction `str()` pour afficher la structure interne de l'objet (reproduit partiellement ci dessous)

```
> str(mod.lm)
List of 13
 $ coefficients : Named num [1:3] -2.455 1.962 0.577
  ..- attr(*, "names")= chr [1:3] "(Intercept)" "genre" "revenu"
 $ residuals    : Named num [1:16] 1.878 1.802 -0.428 -1.076 1.878 ...
  ..- attr(*, "names")= chr [1:16] "1" "2" "3" "4" ...
 $ fitted.values: Named num [1:16] -1.878 -1.302 0.428 9.076 -1.878 ...
  ..- attr(*, "names")= chr [1:16] "1" "2" "3" "4" ...
```

Ainsi, à partir de l'objet `mod.lm`, nous pouvons extraire la valeur des coefficients, des résidus, etc...

La moyenne pondérée du carré des résidus, $\widehat{\sigma}^2 = \frac{1}{n}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}})^\top \boldsymbol{W}(\boldsymbol{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}})$, peut être obtenue avec

```
> mean(mod.lm$residuals^2*w)
```

et $\left(\boldsymbol{X}^\top \boldsymbol{W}\boldsymbol{X}\right)^{-1}$ avec

```
> solve(t(model.matrix(mod.lm))%*%diag(w)%*%model.matrix(mod.lm))
```

La matrice d'information de Fisher estimée, $\widehat{\sigma}^2\left(\boldsymbol{X}^\top \boldsymbol{W}\boldsymbol{X}\right)^{-1}$, est donc obtenue avec

```
> mean(mod.lm$residuals^2*w) *
+ solve(t(model.matrix(mod.lm))%*%diag(w)%*%model.matrix(mod.lm))
```

Cependant, la matice d'information de Fisher diffère de la matrice de variance covariance calculée par R avec la fonction `vcov(mod.lm)`

```
> mean(mod.lm$residuals^2*w)*solve(t(model.matrix(mod.lm))
+ %*%diag(w)%*%model.matrix(mod.lm))/vcov(mod.lm)
            (Intercept)  genre revenu
(Intercept)      0.8125 0.8125 0.8125
genre            0.8125 0.8125 0.8125
revenu           0.8125 0.8125 0.8125
```

La différence est une constante multiplicative. La fonction de variance covariance de R calcule l'estimateur non biaisé de σ^2 qui est $\frac{n}{n-p}\widehat{\sigma}^2$. Avec $n = 16$ et $p = 3$, nous vérifions que $\frac{n-p}{n} = 0.8125$.

```
> (16-3)/16
[1] 0.8125
```

1.5 Exercice 1.3

Dans les résultats affichés par `summary(mod.glm)`, trouver où la quantité `sqrt(diag(vcov(mod.glm)))` apparait. Lorsque la fonction `diag()` est appliquée à une matrice, `diag(A)` renvoie un vecteur composé des éléments de la diagonale de la matrice `A`.

La quantité `sqrt(diag(vcov(mod.glm)))` correspond à la déviation standard des coefficients

```
> sqrt(diag(vcov(mod.glm)))
(Intercept)       genre      revenu
 0.69591912  0.20203086  0.03454757
```

La commande `summary(mod.glm)` affiche un résumé des résultats du modèle ajusté (reproduit partiellement ci dessous)

```
> summary(mod.glm)

Call:
glm(formula = Y ~ genre + revenu, family = poisson, weights = w)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8549  -0.6218  -0.3677   0.1001   1.3608

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept) -2.94289    0.69592  -4.229 2.35e-05 ***
genre        0.51821    0.20203   2.565   0.0103 *
revenu       0.24863    0.03455   7.197 6.17e-13 ***
---
```

2 Analyse logit et régression de Poisson

2.1 Exemple avec R

Supposons que les données suivantes concernant l'invalidité des individus sont disponibles :

- ✔ pour 3 catégories d'emplois, codées 1, 2 et 3
- ✔ dans 3 régions, également codées 1, 2 et 3
- ✔ pour les classes d'âge 20, 30, 40, 50 et 60

Pour chaque cellule, le nombre d'individus est donné par la variable `n` et le nombre d'invalidité par la variable `b`. Les données sont présentées ci-dessous. La probabilité d'être invalide dépendra de l'emploi occupé, de la région, et de la classe d'âge. On peut supposer qu'entre chaque décennie, la probabilité d'être invalide augmente par un facteur multiplicatif fixe et que pour chaque région/emploi le même facteur est appliqué.

Cette situation pourrait être abordée par une analyse logit ou probit. L'analyse logit est en fait un GLM avec un modèle log-linéaire, non pas pour la probabilité p elle-même, mais pour $p/(1-p)$. Cependant, nous souhaitons étudier un modèle multiplicatif, non pas pour les *odds-ratio* $p/(1-p)$ mais pour la probabilité binomiale p. Mais puisque les probabilités d'être invalide ne sont pas très larges, le fait de regarder p ou $p/(1-p)$ ne donnera pas une différence très grande. Le fait que le logit est le lien canonique n'est pas une raison suffisante pour supposer que la probabilité de succès soit linéaire sur cette échelle. Mais utiliser une fonction de lien log dans une modèle binomial peut être un désavantage car dans certain cas des probabilités supérieures à 1 peuvent apparaitre.

Nous voulons étudier trois modèles. Dans le premier, nous supposons des erreurs binomiales alors que dans les deux autres, nous utiliserons des erreurs de Poisson comme approximation. Nous voulons utiliser `b` comme variable dépendante, et l'expliquer avec `r` pour la région, `e` pour l'emploi et `a` pour la classe d'âge.

Dans R, nous pouvons ajuster un modèle binomial en prenant la proportion de succès `b/n` comme variable dépendante et en assignant des poids correspondant au nombre d'observations `n`. Ceci peut aussi être appliqué pour l'ajustement du modèle de Poisson. De cet manière, et en appliquant la même fonction de lien pour chacun des modèles, nous pouvons facilement comparer les modèles.

Nous lisons les données dans R

```
> n <- scan(n=45)
1: 300 248 285 215 124 432 262 288 228 118 421 204 283 263 141
16: 243 248 208 230 288 286 241 300 276 288 247 218 249 247 217
31: 130 265 231 232 420 109 228 213 224 394 116 295 296 264 354
Read 45 items
> b <- scan(n=45)
1: 21 25 27 30 19 41 19 27 30 17 64 37 65 77 41
16: 10 19 24 17 42 29 33 38 42 45 34 41 55 65 61
31: 3 18 15 12 39 7 14 15 19 39 16 36 57 50 82
Read 45 items
> r <- rep(1:3, each=15, len=45); r <- as.factor(r)
```

```
> e <- rep(1:3, each=5, len=45); e <- as.factor(e)
> a <- rep(10*(2:6), len=45); a <- as.factor(a)
```

Les variables explicatives représentent les valeurs numériques 1, 2 et 3 à la fois pour la région et les catégories d'emploi et 20, 30, 40, 50 et 60 pour les classes d'âge. Mais nous les avons traité comme des facteurs (variables catégorielles), leurs valeurs sont seulement interprétés comme le nom de la modalité, non pas comme des valeurs numériques. Elles sont codées dans la matrice de design par une variable indicatrice reflétant l'appartenance à une classe particulière.

Nous commençons par estimer un modèle binomial avec une fonction de lien log pour expliquer `b/n` à partir de `r`, `e` et `a`, avec `n` comme poids.

```
> Bin1 <- glm(b/n ~ r + e + a, weights=n, family=binomial(link = log))
```

Cette commande crée un objet de classe `"glm"` assigné à `Bin1`. En tapant le nom de l'objet, nous obtenons un résumé du contenu de l'objet ; plus de résultats peuvent être obtenus avec la commande `summary(Bin1)`, alors que la commande `str(Bin1)` affiche l'ensemble du contenu de l'objet.

```
> Bin1

Call:  glm(formula = b/n ~ r + e + a, family = binomial(link = log),
    weights = n)

Coefficients:
(Intercept)           r2           r3           e2           e3
   -2.70404     -0.01033     -0.36024      0.19389      0.86025
        a30          a40          a50          a60
    0.19818      0.38576      0.52525      0.66471

Degrees of Freedom: 44 Total (i.e. Null);  36 Residual
Null Deviance:            402.3
Residual Deviance: 36.31         AIC: 280.9
```

Après la rappel de la commande qui a produit l'objet, R affiche les estimations obtenues des coefficients. La constante (l'*intercept*) est le prédicteur linéaire pour un individu ayant toutes les modalités de référence des facteurs, dans ce cas, `1` pour la région et la catégorie de l'emplois, et `20` pour la classe d'âge. Les autres chiffres représentent ce qui doit être ajouté au prédicteur linéaire : soustraire 0.0103 pour les personnes vivant dans la région `2`, ajouter 0.8602 pour ceux ayant un emploi dans la classe `3`, et ainsi de suite.

La qualité de l'ajustement peut être évaluée à partir des données. Le modèle nul a seulement la constante comme terme du modèle. En conséquence, les degrés de liberté correspondent au nombre d'observation $3 \times 3 \times 5$ moins 1. Ce modèle a une déviance de 402. En ajoutant 8 paramètres pour les effets de la région et de la catégorie de l'emploi autre que `1` et pour les classes d'âge autre que `20`, la déviance résiduelle diminue à 36.3. L'AIC est l'*Akaike ?s Information Criterion*. Nous l'analyserons dans ce qui suit.

Pour obtenir les facteurs multiplicatifs à partir des valeurs ajustées des coefficients additifs du prédicateur linéaire, nous prenons simplement l'exponentiel des coefficients de `Bin1`

```
> exp(coef(Bin1))
(Intercept)          r2          r3          e2          e3         a30
 0.06693451  0.98971824  0.69750947  1.21396626  2.36374950  1.21917713
        a40         a50         a60
 1.47073432  1.69088735  1.94393184
```

Par exemple, pour un individu de la région 2, qui occupe l'emploi 3 et de classe d'âge 50, la probabilité estimée d'être invalide est

$$\exp(-2.70404 - 0.01033 + 0.86025 + 0.52525) = 0.067 \times 0.990 \times 2.364 \times 1.691 = 0.265$$

Ce cas correspond à la valeur ajustée de l'observation 29,

```
> fitted(Bin1)[29]
       29
0.2647755
```

2.2 Exercice 2.1

(Modèle 2) Ajuster un modèle de Poisson avec une fonction de lien log **pour expliquer** `b/n` **à partir des même variables explicatives que le Modèle 1, avec** `n` **comme poids. Assigner le résultat à un objet nommé** `Poi1`**. Afficher le modèle et l'estimation des paramètres, et les facteurs multiplicatifs à partir des valeurs ajustées.**

L'objectif est d'estimer le nombre moyen d'invalidités. Pour décrire des événements rares, la distribution de Poisson, qui n'a qu'un seul paramètre, est toujours le premier choix. En effet, on peut utiliser un processus de Poisson pour décrire l'évolution dans le temps du nombre d'invalidité. Une caractéristique du processus de Poisson est qu'il est sans mémoire : l'apparition d'une invalidité dans la seconde suivante est indépendante du passé. L'avantage d'un processus sans mémoire est sa simplicité mathématique ; l'inconvénient est qu'il n'est souvent pas réaliste. Si le modèle pour le nombre d'incapacités présente une plus grande propagation autour de la valeur moyenne, on peut utiliser la loi binomiale négative à la place.

On exécute la commande suivante

```
> print(Poi1 <- glm(formula= b/n ~ r + e + a,family=poisson(link=log), weights=n))

Call:  glm(formula = b/n ~ r + e + a, family = poisson(link = log),
    weights = n)

Coefficients:
(Intercept)           r2           r3           e2           e3
  -2.704932    -0.007434    -0.365588     0.196222     0.863137
        a30          a40          a50          a60
   0.200211     0.384574     0.519145     0.667810

Degrees of Freedom: 44 Total (i.e. Null);  36 Residual
Null Deviance:            345.4
Residual Deviance: 32.31         AIC: Inf
Il y a eu 45 avis (utilisez warnings() pour les visionner)
```

Afin d'interpréter les paramètres, nous calculons

```
> exp(Poi1$coefficients)
(Intercept)          r2          r3          e2          e3         a30
 0.06687488  0.99259404  0.69378828  1.21679662  2.37058659  1.22166070
        a40         a50         a60
 1.46898837  1.68059086  1.94996188
```

Une personne qui appartient à la classe d'âge 60 au lieu de la classe d'âge 20 a en moyenne 1.95 fois plus de *chance* d'être invalide. Nous voyons une nette augmentation de la probabilité d'être invalide avec la classe d'âge.

2.3 Exercice 2.2

(Modèle 3) Ajuster le Modèle 2 une nouvelle fois, mais maintenant traiter l'âge comme variable numérique, en remplaçant la variable explicative `a` par `as.numeric(a)`. Assigner le modèle à l'objet `Poi2` et afficher les résultats. La matrice de design X dans la composante systématique du GLM (i.e $\eta = X\beta$) qui a été utilisée en exécutant la fonction `glm()` est accessible par la commande `X1<- model.matrix(Poi1)`. Comparer les dix premières lignes `X1[1:10,]` avec celles obtenues pour le Modèle 3, et commenter sur leurs différences.

Nous exécutons la commande suivante

```
> print(Poi2 <- glm(formula= b/n ~ r + e + as.numeric(a),family=poisson(link=log),
+ weights=n))

Call:  glm(formula = b/n ~ r + e + as.numeric(a), family = poisson(link = log),
    weights = n)

Coefficients:
  (Intercept)            r2            r3            e2
    -2.840651     -0.007752     -0.365348      0.196157
           e3  as.numeric(a)
     0.863667      0.163472

Degrees of Freedom: 44 Total (i.e. Null);  39 Residual
Null Deviance:            345.4
Residual Deviance: 32.86        AIC: Inf
Il y a eu 45 avis (utilisez warnings() pour les visionner)
```

A première vue, prendre la classe d'âge comme facteur n'améliore pas le modèle. Nous verrons par la suite que le modèle `Poi2` n'est pas rejeté statistiquement contre le modèle `Poi1`. Le modèle `Poi1` ayant 3 degrés de liberté de moins alors que la différence en terme de déviance est seulement de 0.55.

Nous calculons

```
> exp(Poi2$coefficients)
  (Intercept)            r2            r3            e2            e3
   0.05838765    0.99227761    0.69395536    1.21671844    2.37184126
as.numeric(a)
   1.17759269
```

La survenance d'une invalidité pour une personne vivant dans la région `3` est 30.6 % moins que pour un individu en région `1`. Une personne travaillant dans la catégorie `2` au lieu d'avoir un emploi dans la catégorie `1` a 21.6 % d'invalidité en plus en moyenne. Le nombre d'invalidités augmente par un pourcentage fixe, 17.76 %, tous les dix ans.

La matrice de design contient des variables indicatrices (des uns ou zéros) qui indique l'appartenance au groupe comme pour le modèle `Poi1` lorsque la variable `a` correspondant à la classe d'âge est considérée comme un facteur. Cependant, pour le modèle `Poi2`, la matrice de design contient les niveaux représentant les 5 différentes classe d'âge lorsque la variable est considérée comme numérique.

```
> model.matrix(Poi1)[1:10,]
   (Intercept) r2 r3 e2 e3 a30 a40 a50 a60
1            1  0  0  0  0   0   0   0   0
2            1  0  0  0  0   1   0   0   0
3            1  0  0  0  0   0   1   0   0
```

```
4            1  0  0  0  0   0   0   1   0
5            1  0  0  0  0   0   0   0   1
6            1  0  0  1  0   0   0   0   0
7            1  0  0  1  0   1   0   0   0
8            1  0  0  1  0   0   1   0   0
9            1  0  0  1  0   0   0   1   0
10           1  0  0  1  0   0   0   0   1
> model.matrix(Poi2)[1:10,]
   (Intercept) r2 r3 e2 e3 as.numeric(a)
1            1  0  0  0  0              1
2            1  0  0  0  0              2
3            1  0  0  0  0              3
4            1  0  0  0  0              4
5            1  0  0  0  0              5
6            1  0  0  1  0              1
7            1  0  0  1  0              2
8            1  0  0  1  0              3
9            1  0  0  1  0              4
10           1  0  0  1  0              5
```

2.4 Exercice 2.3

(Tous les modèles) Expliquer pourquoi la constante du Modèle 3, `Poi2`, est si différente que celle des Modèles 1 et 2, `Bin1` et `Poi1`. Pour ces trois modèles, comparer la probabilité ajustée d'être invalide pour l'observation $i = 10$. Pour cette observation, `r=1`, `e=2`, `a=60`, et `n=118`, alors que `b=17` ; ainsi la valeur observée est $17/118 = 0.144$. Indiquer comment les valeurs ajustées peuvent être obtenues en utilisant seulement les résultats affichés par les commandes `Bin1`, `Poi1` et `Poi2`.

La constante peut être interprétée comme la valeur estimée de la variable dépendante pour un individu ayant tous les facteurs au niveau de référence et toutes les variables explicatives numériques égalent à 0. La constante des Modèles 1 et 2, `Bin1` et `Poi1` est calculée en incluant le facteur relatif à la classe d'âge, au contraire du Modèle 3 `Poi2`.

La probabilité ajustée d'être invalide pour une personne ayant les caractéristiques suivantes, région = `1`, emploi = `2`, classe d'âge = `60` est obtenue par :

- ✔ Pour le cas binomial :

$$\exp\big(-2.7040 + 0 + 0.1939 + 0.6647\big) = 0.067 \times 1 \times 1.214 \times 1.944 = 0.1580$$

- ✔ Pour le cas Poisson avec classe d'âge comme facteur :

$$\exp\big(-2.7049 + 0.1962 + 0.6678\big) = 0.1587$$

- ✔ Pour le cas Poisson avec classe d'âge comme numérique :

$$\exp\big(-2.840651 + 0.196157 + 0.163472 \times 5\big) = 0.1609$$

Nous pouvons aussi obtenir ces résultats avec R. La valeur observée de la probabilité de survenance de l'invalidité pour l'observation $i = 10$ est

```
> print(p.obs <- (b/n)[10])
[1] 0.1440678
```

et les probabilités ajustées sont

```
> p.fit <- c(fitted(Bin1)[10], fitted(Poi1)[10], fitted(Poi2)[10])
> rslt <- cbind(p.fit, (1-p.obs/p.fit)*100)
> rownames(rslt) <- c("Bin1", "Poi1", "Poi2")
> colnames(rslt) <- c("Pr. Ajust.", "Erreur (%)")
> rslt
     Pr. Ajust. Erreur (%)
Bin1  0.1579566   8.792784
Poi1  0.1586745   9.205448
Poi2  0.1608743  10.446958
```

2.5 Exercice 2.4

(Comparaisons du modèle binomial et modèle de Poisson) A partir des paramètres estimés, nous pouvons voir que les modèles 1 et 2, `Bin1` et `Poi1` se ressemblent fortement : les prédictions (valeurs ajustées) générées par ces deux modèles sont identiques souvent à deux, voir trois décimales près. Vérifier cette affirmation, et expliquer pourquoi. Utiliser la commande `mean(abs(fitted(Bin1)-fitted(Poi1)))`.

La commande renvoie

```
> mean(abs(fitted(Bin1)-fitted(Poi1)))
[1] 0.0005460667
```

Par ailleurs

```
> summary(fitted(Bin1)-fitted(Poi1))
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-2.444e-03 -3.497e-04  5.470e-05 -5.655e-05  3.201e-04  1.757e-03
```

Le modèle binomial et de Poisson se ressemblent fortement. Il est bien connu que si $X \sim \mathcal{B}(n,p)$, alors la valeur ajustée de X est $\mathbb{E}[X] = np$ et la variance est $\mathbb{V}[X] = np(1-p)$. Si n est grand et p petit, de sorte que np est de taille moyenne, la distribution de Poisson de paramètre $\lambda = np$ est une bonne approximation. Si la taille de la base de données est grande et les probabilités de succès petite taille, alors le modèle binomial et de Poisson donnent des résultats sensiblement identiques. Comme n tend vers ∞ et p tend vers 0 et np reste fixe à $\lambda > 0$, la distribution binomiale (n,p) tend vers la distribution de Poisson d'espérance λ.

2.6 Exercice 2.5

(Inférence) La différence entre les modèles 2 et 3, `Poi1` et `Poi2`, est la façon dont l'âge est pris en compte. Comment l'hypothèse nulle *la fréquence d'invalidité augmente de façon exponentielle avec la classe d'âge* peut être tester contre l'hypothèse alternative *elle augmente avec la classe d'âge de façon arbitraire* ? Faire le test.

Pour tester le problème : $\mathcal{H}_0$: "M_a *tient*" contre $\mathcal{H}_1$: "M_b *tient*", où M_a et M_b sont des modèles emboités (M_a est définit en imposant des restrictions sur M_b), on effectue le test du rapport de vraisemblance (appelé aussi test de la déviance). On calcul la statistique du ratio de vraisemblance

$$\xi = 2(l_b - l_a) = 2\left[(l_s - l_a) - (l_s - l_b)\right] = \frac{(D_a - D_b)}{\phi}$$

où D_a and D_b sont les déviances des modèles M_a and M_b respectivement. Sous l'hypothèse que le nombre d'invalide suit une loi de Poisson $b_i \sim \mathcal{P}(n_i\, p_i)$, la déviance s'écrit

$$\begin{aligned}
&\text{si } b_i > 0, D_i = 2\left(b_i \ln\left(\frac{b_i}{n_i\,\widehat{p}_i}\right) - \left(b_i - n_i\,\widehat{p}_i\right)\right).\\
&\text{si } b_i = 0, D_i = 2\,n_i\,\widehat{p}_i.\\
&\text{et}\quad \text{Déviance totale} = \sum_i D_i\,.
\end{aligned}$$

ξ a asymptotiquement, sous $\mathcal{H}_0$, une distribution de χ^2 avec les degrés de liberté, m, égalent à la différence entre le nombre de paramètres des modèles M_a and M_b :

$$\xi \sim \chi^2(m).$$

Ainsi, l'hypothèse nulle $\mathcal{H}_0$ est rejetée si

$$\xi > \chi^2_{1-\alpha}(m),$$

où $\chi^2_{1-\alpha}(m)$ est le $(1-\alpha)$ quantile de la distribution de χ^2 avec m degrés de liberté.

Pour pouvoir tester l'hypothèse nulle *la fréquence d'invalidité augmente de façon exponentielle avec la classe d'âge* contre l'hypothèse *elle augmente avec la classe d'âge de façon arbitraire*, nous effectuons un test de la déviance entre les modèles `Poi2` et `Poi1`. Avec R, nous pouvons utiliser pour calculer la statistique du ratio de vraisemblance avec la fonction `anova()`

```
> anova(Poi2, Poi1, test = "Chisq")
Analysis of Deviance Table

Model 1: b/n ~ r + e + as.numeric(a)
Model 2: b/n ~ r + e + a
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        39     32.864
2        36     32.310  3  0.55392   0.9069
```

On obtient la valeur de $\chi^2_{0.95}(3)$

```
> qchisq(.95,3)
[1] 7.814728
```

L'hypothèse $\mathcal{H}_0$ n'est pas rejetée, le modèle `Poi2` n'est pas rejetée statistiquement contre le modèle `Poi1` au seuil de 95 % puisque que le modèle `Poi1` contient 3 paramètres de moins et que la valeur du $\chi^2_{0.95}(3) = 7.81$ alors que la différence entre les déviances n'est que de 0.55. Il est vraisemblable que la probabilité d'être invalide augmente de façon exponentielle plutôt que de façon arbitraire.

On peut retrouver la p-valeur obtenue avec la fonction `anova()` par

```
> 1-pchisq(0.55392,3)
[1] 0.9068954
```

3 Analyser un portefeuille simple en assurance automobile

3.1 Exemple avec R

Nous allons étudier un portefeuille d'assurance automobile. Chaque cellule contient les données agrégées de conducteurs ayant des valeurs identiques pour les facteurs de risques suivant :

- ✔ genre : 1 = femme, 2 = homme,
- ✔ région : 1 = rurale, 2 = autre, 3 = urbaine,
- ✔ type de la voiture : 1 = petite, 2 = moyenne, 3 = grande,
- ✔ emploi : 1 = fonctionnaire/actuaire..., 2 = entre-deux, 3 = 'dynamique',

Pour chaque cellule, nous connaissons

- ✔ `npol` : nombre total d'assurés dans cette cellule
- ✔ `n` nombre total de sinistres observés durant l'année précédente

Les observations sont ordonnées de telle sorte que

- ✔ la variable `genre` est égale à `1` pour les 27 premières observations, et `2` pour les 27 dernières observations ;
- ✔ la variable `region` a trois modalités qui se produisent par blocs de 9 observations ;
- ✔ la variable `type` a trois modalités qui se produisent par blocs de 3 observations ;
- ✔ la variable `emploi` a trois modalités qui se répètent jusqu'à atteindre la 54ème observations.

Les modalités des variables catégorielles peuvent être reconstruits en utilisant la fonction `gl()`. La commande `gl(n,k,N)` produit un vecteur de longueur `N` contenant les catégories numérotées `1` à `n` par blocs de longueur `k`. La commande `gl(n,k,N)` est équivalente à `as.factor(rep (1: n, each = k, length = N))`.

```
> genre <- gl(2, 27, 54); region <- gl(3, 9, 54)
> type <- gl(3, 3, 54); emploi <- gl(3, 1, 54)
```

Le nombre moyen de sinistres par contrat `n/npol` dans chaque cellule est la quantité d'intérêt.

```
> npol <- scan(n=54)
1: 50 61 90 77 91 75 129 86 80 145 54 103 56 76 127 125 133 96
19: 112 114 143 98 84 109 60 123 74 114 98 113 131 148 140 140 52 74
37: 108 132 83 111 115 129 126 57 105 68 104 123 141 92 67 91 125 63
Read 54 items
> n <- scan(n=54)
1: 3 11 11 6 8 9 10 2 14 11 6 9 4 10 19 9 23 21
19: 11 12 21 8 10 16 10 28 15  5 9 10 6 17 23 21
35: 9 16 8 15 12 11 9 22 23 15 25 7 13 20 13 17 10 8 24 12
Read 54 items
```

Nous pouvons visualiser les données en les regroupant

```
> cbind(npol, n, genre, region, type, emploi)[1:10,]
      npol  n genre region type emploi
 [1,]   50  3     1      1    1      1
 [2,]   61 11     1      1    1      2
```

```
 [3,]   90 11     1      1    1      3
 [4,]   77  6     1      1    2      1
 [5,]   91  8     1      1    2      2
 [6,]   75  9     1      1    2      3
 [7,]  129 10     1      1    3      1
 [8,]   86  2     1      1    3      2
 [9,]   80 14     1      1    3      3
[10,]  145 11     1      2    1      1
```

Nous voulons lier le nombre moyen de sinistres aux facteurs de risque donné, pour établir ou analyser un système de classification pour le portefeuille. Si il s'avère que les conducteurs en milieu urbain, par exemple, ont trois fois plus de sinistres que les conducteurs en milieu rural, ils devraient, en principe, avoir une prime trois fois plus élevée. Puisque nous voulons une classification, c'est à dire, un système multiplicatif, nous allons voir si un modèle log-linéaire bien ajusté peut expliquer la fréquence des sinistres en fonction des facteurs de risque.

Pour le nombre de sinistres par contrat, il est raisonnable de supposer une distribution de Poisson. Parce que la variance dépend de la moyenne et que nous avons un modèle multiplicatif, pas un additif un, nous allons ajuster un modèle linéaire généralisé plutôt que d'un modèle linéaire ordinaire. N'ayant que des données agrégées dans chaque cellule (pas de données sur les contrats individuels), nous faisons un modèle pour le nombre moyen de sinistres par contrat, c'est à dire, pour `n/npol`. Cela signifie que

Mais cela signifie que, mis à part la proportionnalité de la variance de la variable expliquée à la moyenne, elle doit être également à diviser par `npol`. Ceci est réaliser dans la commande `glm()` en indiquant qu'il existe un poids `npol` pour chaque observation / cellule, comme suit :

```
> print(glm.1 <- glm(n/npol ~ genre+region+type+emploi, fam= poisson
+ (link=log), wei=npol))

Call:  glm(formula = n/npol ~ genre + region + type + emploi, family =
+ poisson(link = log),
    weights = npol)

Coefficients:
(Intercept)       genre2      region2      region3        type2
    -2.8401       0.1346       0.1995       0.2742       0.1026
      type3      emploi2      emploi3
     0.4773       0.3881       0.5779

Degrees of Freedom: 53 Total (i.e. Null);  46 Residual
Null Deviance:            118.4
Residual Deviance: 44.46        AIC: Inf
Il y a eu 50 avis ou plus (utilisez warnings() pour voir les 50 premiers)
```

Nous visualisons l'objet résultant de la commande `glm()`, nommé `glm.1`. Les avertissements (*warnings()*) proviennent du fait que les moyennes utilisées ne sont pas des nombres entiers dans la plupart des cas, et ne sont donc pas Poisson distribuées. Mise à part que l'AIC ne peut pas être calculé, cela ne présente pas un problème.

3.2 Exercice 3.1

A partir des résultats du modèle `glm.1`, déterminer le nombre de sinistres qu'a en moyenne un individu appartenant aux modalités les plus mauvaises. Comparer avec un individu ayant la meilleure combinaison.

Par rapport à un individu ayant la meilleur combinaison (`genre = 1`, `region = 1`, `type = 1` et `emploi = 1`), un individu ayant la plus mauvaise combinaison (`genre = 2`, `region = 3`, `type = 3` et `emploi = 3`), a en moyenne 432.32 % plus de sinistres.

$$\exp\big(0.1346 + 0.2742 + 0.4773 + 0.5779\big) \times 100 = 432.3218$$

3.3 Exercice 3.2

Tester l'hypothèse nulle qu'ajouter tous les facteurs `genre`, `region`, `type`, `emploi`, n'a en réalité aucun effet.

Dans ce cas, la différence entre la déviance du modèle nul et les modèles incorporant ces facteurs a une distribution de χ^2 avec comme degrés de liberté le nombre de paramètres estimés. A partir des résultats du modèle `glm.1`, nous pouvons voir que le modèle incluant les facteurs à $53 - 46 = 7$ paramètres de moins que le modèle nul, alors que le modèle nul a une déviance $\xi = 118.4 - 44.46 = 73.94$ plus grande. La valeur de $\chi^2_{0.95}(7)$ est

```
> qchisq(.95,7)
[1] 14.06714
```

Comme, $\xi > \chi^2_{0.95}(7)$, nous pouvons conclure que le modèle incluant les facteurs est significativement meilleur.

3.4 Exercice 3.3

En s'inspirant de la commande utilisée pour produire l'objet `glm.1`, exécuter une série de commande `glm()` à partir du script R en enlevant le dernier terme du modèle à chaque fois. Est-ce que tous les facteurs enlevés sont significatifs ? A quelle conclusion arrivez-vous pour le 'meilleur' modèle ? Regardez également le modèle avec seulement les facteurs `region`, `type` et `emploi`.

A partir du script R, nous créons les objets `glm.2`, `glm.3` et `glm.4` :

```
> glm.2 <- glm(n/npol ~ genre+region+type, fam= poisson(link=log), wei=npol)
> glm.3 <- glm(n/npol ~ genre+region, fam= poisson(link=log), wei=npol)
> glm.4 <- glm(n/npol ~ genre, fam= poisson(link=log), wei=npol)
```

Nous affichons la déviance et les degrés de liberté de chaque modèle par

```
> cbind(c(glm.2$deviance, glm.3$deviance, glm.4$deviance),
+ c(glm.2$df.residual, glm.3$df.residual, glm.4$df.residual))
          [,1] [,2]
[1,]  82.24081   48
[2,] 108.18971   50
[3,] 117.38835   52
```

La valeur de $\chi^2_{0.95}(2)$ est

```
> qchisq(.95,2)
[1] 5.991465
```

Nous voyons que

- ✔ ajouter le facteur `emploi` (modèle `glm.1`) au modèle incluant les facteurs `genre`, `region` et `type` (modèle `glm.1`) permet de faire diminuer la déviance de $82.24 - 44.46 = 37.78$ alors que le nombre de degré de liberté diminue de $48 - 46 = 2$. Le gain en terme de déviance étant supérieur à la valeur de $\chi^2_{0.95}(2)$, nous pouvons en conclure que le facteur `emploi` est significatif. Le modèle `glm.1` est 'meilleur' que le `glm.2`.
- ✔ ajouter le facteur `type` (modèle `glm.2`) au modèle incluant les facteurs `genre` et `region` (modèle `glm.3`) fait diminuer la déviance de $108.18 - 82.24 = 25.94$ alors que le nombre de degré de liberté diminue de $50 - 48 = 2$. Le facteur `type` est significatif. Le modèle `glm.2` est 'meilleur' que le `glm.3`.
- ✔ ajouter le facteur `region` (modèle `glm.3`) au modèle incluant les facteur `genre` (modèle `glm.4`) fait diminuer la déviance de $117.39 - 108.18 = 9.21$ alors que le nombre de degré de liberté diminue de $52 - 50 = 2$. Le facteur `region` est significatif. Le modèle `glm.3` est 'meilleur' que le `glm.4`
- ✔ ajouter le facteur `genre` (modèle `glm.4`) au modèle nul fait diminuer la déviance de $118.4 - 117.39 = 1.01$ alors que le nombre de degré de liberté diminue de $53 - 52 = 1$. Le gain en terme de déviance étant inférieur à la valeur de $\chi^2_{0.95}(1) = 3.84$, le facteur `genre` n'est pas significatif. Le modèle nul est 'meilleur' que le `glm.4`.

Nous aurions pu voir que le facteur `genre` n'est pas significatif au seuil de 95 % à partir du résultat produit par la fonction `summary()`

```
> summary(glm.1)

Call:
glm(formula = n/npol ~ genre + region + type + emploi, family = poisson(link = log),
    weights = npol)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5677  -0.2730   0.0877   0.3617   2.1828

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.84007    0.12580 -22.576  < 2e-16 ***
genre2       0.13458    0.07646   1.760 0.078372 .
region2      0.19952    0.09626   2.073 0.038192 *
region3      0.27416    0.09638   2.845 0.004445 **
type2        0.10261    0.09895   1.037 0.299757
type3        0.47726    0.09349   5.105 3.30e-07 ***
emploi2      0.38814    0.10003   3.880 0.000104 ***
emploi3      0.57792    0.09656   5.985 2.16e-09 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 118.449  on 53  degrees of freedom
Residual deviance:  44.461  on 46  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 5
```

Nous pouvons comparer le modèle avec seulement les facteurs `region`, `type` et `emploi` au modèle incluant tous les facteurs.

```
> glm.5 <- glm(n/npol ~ region+type+emploi, fam= poisson(link=log), wei=npol)
> anova(glm.5, glm.1, test = "Chisq")
Analysis of Deviance Table

Model 1: n/npol ~ region + type + emploi
Model 2: n/npol ~ genre + region + type + emploi
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        47     47.569
2        46     44.461  1   3.1082   0.0779
```

Nous nous apercevons que le modèle `glm.5` n'est rejeté statistiquement au seuil de 95 % contre le modèle `glm.1` incluant tous les facteurs. Le gain en terme de déviance, 3.11, est inférieur à la valeur de $\chi^2_{0.95}(1) = 3.84$.

En exécutant la commande

```
> anova(glm.1)
Analysis of Deviance Table

Model: poisson, link: log

Response: n/npol

Terms added sequentially (first to last)


       Df Deviance Resid. Df Resid. Dev
NULL                      53    118.449
genre   1    1.060        52    117.388
region  2    9.199        50    108.190
type    2   25.949        48     82.241
emploi  2   37.780        46     44.461
```

les facteurs sont ajoutés un par un du premier au dernier. Nous arrivons à la conclusion similaire que le facteur genre n'est pas significatif.

3.5 Exercice 3.4

Trouver le nombre total d'assurés impliqués en utilisant `sum(npol)`, ainsi que le nombre total des sinistres. Vérifiez que la valeur ajustée obtenue par le modèle nul est égale au nombre total de sinistres divisé par le nombre d'assurés. Comparer avec deux autres façons de calculer la moyenne des sinistres `mean(n/npol)` et `weighted.mean(n/npol, npol)`. Expliquer pourquoi nous trouvons des résultats différents. Pour avoir plus d'explication, faire `?weighted.mean`.

Le nombre total d'assurés impliqués et de sinistres sont

```
> sum(npol); sum(n);  sum(n)/sum(npol)
[1] 5421
[1] 697
[1] 0.1285741
```

Nous obtenons la valeur ajustée du modèle nul en exécutant la commande

```
> glm.5 <- glm(n/npol ~ 1, fam= poisson(link=log), wei=npol)
Il y a eu 50 avis ou plus (utilisez warnings() pour voir les 50 premiers)
> fitted(glm.5)[1]
```

```
          1
0.1285741
```

Cette valeur correspond à la moyenne arithmétique pondérée qui diffère de la moyenne arithmétique.

```
> weighted.mean(n/npol, npol); mean(n/npol)
[1] 0.1285741
[1] 0.1305409
```

Les poids $w_i =$ `npol` sont nécessaires pour modéliser la fréquence des sinistres moyen d'un conducteur dans une cellule avec w_i assurés. En ne prenant pas en compte les poids, on ne tient pas compte du fait que les observations dans des cellules où le nombre d'assurés est important ont été mesurées avec beaucoup plus de précision que celles dont le nombre d'assurés est faible.

4 Implémenter l'algorithme IRWLS de Nelder et Wedderburn avec R

4.1 Bref rappel théorique

Nelder and Wedderburn (1972) ont montré qu'avec la formulation des modèles linéaires généralisés, qui englobe toute une gamme de modèles, une méthode générale permet de résoudre tous les modèles. Cette méthode est appelée la méthode des moindres carrés itérativement repondérés (ou algorithme IRWLS pour *iteratively reweighted least squares*).

Rappelons quelques formules afin de décrire le problème et la notation. L'élément aléatoire d'un GLM établit que les observations sont des variables aléatoires indépendantes Y_i, $i = 1, \ldots, n$ avec une log-densité de la forme :

$$\ell\big(\beta_1, \ldots, \beta_p | (y_i, \phi, w_i)\big) = \log f(y_i|\theta_i, \phi, w_i) = \frac{y_i\theta_i - b(\theta_i)}{\phi/w_i)} + c(y_i).$$

Ici, θ_i est le paramètre d'intérêt, déterminant la moyenne $\mu_i = \mathbb{E}\big[Y_i\big]$, ϕ est le paramètre d'échelle (paramètre de nuisance) identique pour toutes les observations, et w_i est un poids que l'on suppose connu. L'élément systématique établit que la moyenne est déterminée par un ensemble de paramètres $\beta_1, \ldots, \beta_p$ à travers un ensemble de prédicteurs linéaires η_i où $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ avec $\boldsymbol{X}$ la $n \times p$ matrice des covariables. La moyenne et les prédicteurs linéaires sont liés par une fonction de lien $g()$, $\eta_i = g(\mu_i), i = 1, \ldots, n$. L'espérance et la variance sont calculées comme la première et seconde dérivées de $b(\theta)$:

$$\mu(\theta) = \mathbb{E}\big[Y|\theta\big] = b'(\theta), \text{ et } \mathbb{V}[Y_i|\theta] = (\phi/w_i)b''(\theta).$$

Comme nous voulons maximiser la log-vraisemblance pour $\beta_1, \ldots, \beta_p$, nous cherchons la solution de l'ensemble des équations normales à remplir par les paramètres du maximum de vraisemblance estimés $\widehat{\beta}_j, j = 1, \ldots, p$:

$$\sum_{i=1}^{n} \frac{\partial}{\partial \beta_j} \ell\big(\beta_1, \ldots, \beta_p | y_i\big) = 0, \quad j = 1, \ldots, p$$

Une façon de résoudre ces équations est d'utiliser la méthode de Newton-Raphson. Avec cette technique, au lieu de résoudre $f(x) = 0$, on résout $h(x) = 0$, avec $h(x) = f(x_t) + f'(x_t)(x - x_t)$ une approximation linéaire de f à x_t. Ainsi, pour la cas unidimensionnel,

$$x_{t+1} = x_t - \big(f'(x_t)\big)^{-1} f(x_t).$$

L'algorithme de Newton-Raphson utilise (la négative de) la valeur de l'espérance du Hessien, la matrice d'information. La technique est appelé méthode de scoring de Fisher.

Nous allons montrer que le système d'équations à résoudre lors de l'"itération correspond dans ce cas à celui d'un problème particulier de régression pondérée.

Rappelons que pour la moyenne, et pour les prédicteurs linéaires, avec g désignant la fonction de lien, nous avons

$$\mu(\theta_i) = b'(\theta_i)\,;\, \mathbb{V}[mu_i] = \frac{\mu_i}{\partial\theta_i};\text{ et } \eta_i = \sum_j x_{ij}\,\beta_j = g(\mu).$$

En appliquant la règle de dérivation en chaine,

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l}{\partial \theta_i}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j},$$

où

$$\begin{aligned}
\frac{\partial \ell}{\partial \theta_i} &= \frac{y_i - b'(\theta_i)}{a_i(\phi)} = \frac{y_i - \mu_i}{\phi/w_i} \\
\frac{\partial \mu_i}{\partial \theta_i} &= b''(\theta_i) = \mathbb{V}\big[\mu_i\big] \\
\frac{\partial \eta_i}{\partial \beta_j} &= x_{ij}.
\end{aligned}$$

Nous obtenons,

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{\phi/w_i}\frac{1}{\mathbb{V}\big[\mu_i\big]}\frac{\partial \mu_i}{\partial \eta_i}x_{ij}.$$

La fonction de lien $\eta_i = g(\mu_i)$ détermine

$$\frac{\partial \mu_i}{\partial \eta_i} = \frac{\partial g^{-1}(\eta_i)}{\partial \eta_i}.$$

Pour les secondes dérivées partielles, nous avons :

$$\frac{\partial^2 \ell}{\partial \beta_j \beta_k} = \frac{\partial^2 \ell}{\partial \eta_i^2}x_{ij}x_{ik}.$$

En appliquant la règle de dérivation en chaine et la règle du produit, $(f \times g)' = f' \times g + f \times g'$,

$$\frac{\partial^2 \ell}{\partial \eta_i^2} = \frac{\partial}{\partial \eta_i}\left(\frac{\partial \ell}{\partial \theta_i}\frac{\partial \theta_i}{\partial \eta_i}\right) = \left(\frac{\partial^2 \ell}{\partial \theta_i \partial \eta_i}\right)\frac{\partial \theta_i}{\partial \eta_i} + \frac{\partial \ell}{\partial \theta_i}\frac{\partial^2 \theta_i}{\partial \eta_i^2} = \frac{\partial^2 \ell}{\partial \theta_i^2}\left(\frac{\partial \theta_i}{\partial \eta_i}\right)^2 + \frac{\partial \ell}{\partial \theta_i}\frac{\partial^2 \theta_i}{\partial \eta_i^2}$$

Comme $\partial\ell/\partial\theta_i = w_i\,(y_i - \mu_i)/\phi$, sa dérivée est $\partial^2 l/\partial\theta_i^2 = -w/\phi\partial\mu_i/\partial\theta_i$. De plus $\partial\mu_i/\partial\theta_i = b''(\theta_i) = \mathbb{V}\big[\mu_i\big]$, on obtient

$$\frac{\partial^2 \ell}{\partial \eta_i^2} = \frac{w_i}{\phi}\left[-\mathbb{V}\big[\mu_i\big]\left(\frac{\partial \theta_i}{\partial \mu_i}\right)^2\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 + (y_i - \mu_i)\,\frac{\partial^2 \theta_i}{\partial \eta_i^2}\right] = \frac{w}{\phi}\left[-\frac{1}{\mathbb{V}\big[\mu_i\big]}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 + (y_i - \mu_i)\,\frac{\partial^2 \theta_i}{\partial \eta_i^2}\right].$$

Dans le cas d'une fonction de lien canonique, $\theta \equiv \eta$, alors $\partial^2\theta_i/\partial\eta_i^2 = 0$ et il ne reste que le premier terme. Dans la méthode de scoring de Fisher, l'actuelle matrice hessienne dans l'itération de Newton-Raphson est remplacée par sa valeur espérée, qui est la négative de la matrice d'information de Fisher $\mathcal{I}$. Dans ce cas là aussi le second terme disparait.

On obtient

$$\mathcal{I}_{jk} = \mathbb{E}\left[-c\frac{\partial^2 \ell}{\partial\beta_j\beta_k}\right] = \sum_{i=1}^{n} \frac{w_i}{\phi}\frac{1}{\mathbb{V}\big[\mu_i\big]}\left(\frac{\partial\mu_i}{\partial\eta_i}\right)^2 x_{ij}x_{ik} = \sum_{i=1}^{n}\frac{\omega_{ii}x_{ij}x_{ik}}{\phi} = \frac{1}{\phi}(\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X})_{jk},$$

où $\boldsymbol{\Omega}$ est un matrice diagonale avec $\omega_{ii} = \dfrac{w_i}{\mathbb{V}\big[\mu_i\big]}\left(\dfrac{\partial\mu_i}{\partial\eta_i}\right)^2$ qui dépend de μ_i. Car $\eta_i = g(\mu_i)$, on a $\partial\eta_i/\partial\mu_i = g'(\mu_i)$.

En utilisant ces poids, la liste des équations normales $\partial\ell/\partial\beta_j = 0$ peut s'écrire comme :

$$\frac{\partial\ell}{\partial\beta_j} = \sum_{i=1}^{n}\frac{\omega_{ii}\,x_{ij}\,(y_i-\mu_i)}{\phi}g'(\mu_i), j = 1,\ldots,p \text{ ou } \frac{\partial\ell}{\partial\boldsymbol{\beta}} = \frac{1}{\phi}\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{u},$$

avec $u_i = (y_i - \mu_i)g'(\mu_i)$.

En utilisant la méthode du scoring de Fisher, nous trouvons une estimation *améliorée* $\boldsymbol{b}^*$ de $\boldsymbol{\beta}$ à partir d'un précédent $\boldsymbol{b}$ comme suit :

$$\boldsymbol{b}^* = \boldsymbol{b} + \mathcal{I}^{-1}\frac{\partial\ell}{\partial\boldsymbol{\beta}} \text{ ou de façon équivalente } \mathcal{I}(\boldsymbol{b}^* - \boldsymbol{b}) = \frac{\partial\ell}{\partial\boldsymbol{\beta}}.$$

Soit $\widehat{\boldsymbol{\eta}}$ et $\widehat{\boldsymbol{\mu}}$, les vecteurs des prédicteurs linéaires et des réponses estimées lorsque le paramètre est égal à $\boldsymbol{b}$, donc

$$\widehat{\boldsymbol{\eta}} = \boldsymbol{X}\boldsymbol{b} \text{ et } \widehat{\boldsymbol{\mu}} = g^{-1}(\widehat{\boldsymbol{\eta}}),$$

alors

$$\mathcal{I}\boldsymbol{b} = \frac{1}{\phi}\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{b} = \frac{1}{\phi}\boldsymbol{X}^\top\boldsymbol{\Omega}\widehat{\boldsymbol{\eta}}.$$

, et nous pouvons réécrire les équations itératives de Fisher comme suit :

$$\mathcal{I}\boldsymbol{b}^* = \frac{1}{\phi}\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X}\boldsymbol{z},$$

où $z_i = \widehat{\eta}_i + (y_i - \mu_i)g'(\widehat{\mu}_i)$. Les éléments z_i sont appelés des pseudo réponses.

Enfin, on trouve une estimation du maximum de vraisemblance de $\boldsymbol{\beta}$ par la procédure itéative suivante :

On répète $\boldsymbol{b}^* := \left(\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^\top\boldsymbol{\Omega}\boldsymbol{z}$;
en utilisant $\boldsymbol{b}^*$, on révise les pseudo poids $\boldsymbol{\Omega}$, ainsi que les *pseudo réponses* $\boldsymbol{z}$ jusqu'à convergence.

Dans le cas spécial du modèle linéaire ordinaire, nous avons $\eta = g(\mu)$ et $\mathbb{V}\big[\mu_i\big] = 1$, donc les équations normales $\partial\ell/\partial\boldsymbol{\beta} = 0$ sont un système linéaire. La solution est celle indiquée ci-dessus en replaçant $\boldsymbol{z}$ par $\boldsymbol{y}$ et les poids $\omega_{ii} = w_i$. Aucun des itérations sont nécessaires. Par conséquent, la méthode du scoring de Fisher peut effectivement être considéré comme un algorithme des moindres carrés itérativement repondérés.

4.2 Exemple avec R

Nous voulons implémenter l'algorithme avec R. Supposons que nous souhaitons ajuster un modèle de Poisson sur 10 observations avec une constante et une variable explicative, et aucun poids. La matrice de design a un vecteur avec les éléments 1 à sa première colonne et la seconde est $(1, 2, \ldots, 10)^\top$.

L'algorithme IRWLS peut être implémenté comme suit :

✔ Initialiser (On utilise une régression pondérée pour trouver les valeurs initiales)

$\boldsymbol{X} \leftarrow$ matrice de design
$\boldsymbol{y} \leftarrow$ variable dépendante
$\boldsymbol{z} \leftarrow g(\boldsymbol{y})$ fonction de lien
$\boldsymbol{\Omega} \leftarrow$ matrice diagonale contenant les poids
$\boldsymbol{b} \leftarrow \left(\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{z}$

✔ Répéter les étapes suivantes jusqu'à la convergence ou jusqu'à ce que le nombre maximal d'itérations soit atteint

$\boldsymbol{\eta} \leftarrow \boldsymbol{X}\boldsymbol{b}$
$\boldsymbol{\mu} \leftarrow g^{-1}(\boldsymbol{\eta})$
$\omega_{ii} \leftarrow \dfrac{1}{\mathbb{V}\left[\mu_i\right]} \left(\dfrac{\partial \mu_i}{\partial \eta_i}\right)^2$
$\boldsymbol{z} \leftarrow \boldsymbol{X}\boldsymbol{b} + g'(\boldsymbol{\mu}) \times (\boldsymbol{y} - \boldsymbol{\mu})$
$\boldsymbol{S} \leftarrow \left(\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{X}\right)^{-1}$
$\boldsymbol{b} \leftarrow \boldsymbol{S}\boldsymbol{X}^\top \boldsymbol{\Omega} \boldsymbol{z}$

✔ Enfin, les résultats suivant sont produits :

$\boldsymbol{b} =$ estimation de $\boldsymbol{\beta}$
$\boldsymbol{S} =$ estimation de $\mathbb{V}\left[\boldsymbol{b}\right]$: matrice d'information de Fisher.

Pour le dernier résultat, nous utilisons le fait qu' asymptotiquement, $\boldsymbol{b} - \boldsymbol{\beta} \sim \mathcal{N}\left(0, \mathcal{I}^{-1}\right)$.

Notez comment les commandes suivantes de R utilisées dans cette exemple, avec une boucle naïvement implémentée, reproduisent la description ci-dessus :

```
> X <- cbind(rep(1,10),1:10)
> y <- c(14,0,8,8,16,16,32,18,28,22)
> z <- log(y+(y==0))
> Omega <- diag(10)
> b <- solve(t(X) %*% Omega %*% X) %*% t(X) %*% Omega %*% z
> cat("Start:", b[1], b[2], "\n")
```

```
> for (it in 1:5){        # 5 iterations suffisent
+         eta <- as.vector(X %*% b)
+         mu <- exp(eta)                                  # eta = g(mu) = log(mu)
+         Omega <- diag(mu)                          # (g?(mu))?(-2)/V(mu) = mu?2/mu
+         z <- X %*% b + (y-mu)/mu        # d eta/d mu = g?(mu) = 1/mu
+         S <- solve(t(X) %*% Omega %*% X)
+         b <- S %*% t(X) %*% Omega %*% z
+         cat("it =", it, b[1], b[2], "\n")
+         }
```

Dans R, `solve(A)` produit l'inverse de la matrice `A`. Nous obtenons les valeurs $\boldsymbol{b}$ suivantes

```
Start: 1.325353 0.215799
it = 1 1.949202 0.1414127
it = 2 1.861798 0.1511522
it = 3 1.859193 0.1514486
it = 4 1.859191 0.1514489
it = 5 1.859191 0.1514489
```

On peut se rassurer en comparant les résultats obtenus avec la fonction `glm()` :

```
> g <- glm(y ~ c(1:10), poisson)
> coef(g)
(Intercept)     c(1:10)
  1.8591909   0.1514489
```

4.3 Exercice 4.1

Pourquoi utilisons-nous `y==0` dans l'initialisation ?

Rappelons que `y <- c(14,0,8,8,16,16,32,18,28,22)`. Une observation de la variable dépendante est égale 0. Il y a un problème lorsque nous appliquons la fonction de lien puisque `log(0) = Inf`. Pour prévenir ce problème, nous remplaçons cette observation par 1.

4.4 Exercice 4.2

Dans un modèle de Poisson avec une fonction de lien standard, quelles sont les valeurs des pseudo réponses et pseudo poids ? Supposer que le vecteur $\boldsymbol{\beta}$ est connu, quelles sont la moyenne et la variance des pseudo réponses $\boldsymbol{z}$?

La solution des équations normales est équivalente à celle d'une procédure des moindres carrés itérativement repondérés avec les poids suivants

$$\omega_{ii} = \frac{w_i}{\mathbb{V}\big[\mu_i\big]}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2$$

et les pseudo réponses suivantes

$$z_i = \widehat{\eta}_i + (y_i - \mu_i)g'(\widehat{\mu}_i).$$

Sachant que la valeur espérée μ_i de Y_i est liée au prédicteur linéaire η_i par la fonction de lien, et que dans le cas d'un modèle de Poisson, la fonction de lien standard est le logarithme,

$$\eta_i = g(\mu_i) = \log(\mu_i),$$

nous avons

$$\partial \eta_i / \partial \mu_i = g'(\mu_i) = 1/\mu_i$$

Donc $z_i = \widehat{\eta}_i + (y_i - \widehat{\mu}_i)/\widehat{\mu}_i$. De plus,

$$\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 = g'(\mu_i)^{-2} = \mu_i^2 \text{ et } \mathbb{V}\big[\mu_i\big] = \partial \mu_i / \partial \theta_i = \mu_i.$$

En conséquence, les pseudo poids correspondent à une matrice diagonale avec éléments $\omega_{ii} = w_i \times \mu_i^2/\mu_i = w_i \times \mu_i$. En supposant que $\boldsymbol{\beta}$ est connu et que $z_i = \eta_i + (y_i - \mu_i)/\mu_i$,

$$\begin{aligned}
\mathbb{E}\big[z_i\big] &= \eta_i \\
\mathbb{V}\big[z_i\big] &= \mathbb{V}\big[y_i\big]/\mu_i^2 \\
&= \mu_i/w_i \times 1/\mu_i^2 \\
&= 1/(\mu_i \times w_i) \\
&= 1/\omega_{ii}
\end{aligned}$$

Les pseudo poids dans la procédure itérative sont inversement proportionnels à la variance des pseudo réponses.

4.5 Exercice 4.3

Comparer `vcov(g)` et `S`. De même `model.matrix(g)` et `X`. Enfin comparer les erreurs standards des paramètres estimés affichés par `summary(g)` et `sqrt(diag(S))`.

La fonction `vcov()` appliquée à un objet de classe `glm` renvoie la matrice de variance-covariance des paramètres du modèle ajusté. Ceci est équivalent au paramètre de dispersion ϕ multiplié par la matrice de covariance $\left(\boldsymbol{X}^\top \Omega \boldsymbol{X}\right)^{-1}$. Dans le cas d'un modèle de Poisson, le paramètre de dispersion est égal à 1, donc la commande `vcov(g)` donne le même résultat que `S` $= \left(\boldsymbol{X}^\top \Omega \boldsymbol{X}\right)^{-1}$.

```
> vcov(g)
             (Intercept)       c(1:10)
(Intercept)  0.043787682 -0.0056110540
c(1:10)     -0.005611054  0.0008370081
> S
              [,1]          [,2]
[1,]  0.043787687 -0.0056110546
[2,] -0.005611055  0.0008370081
```

La fonction `model.matrix()` renvoie la matrice de design. On vérifie que la matrice de design `X` du modèle correspond à celle obtenue par la commande `model.matrix(g)`.

```
> model.matrix(g)
   (Intercept) c(1:10)
1            1       1
2            1       2
3            1       3
4            1       4
5            1       5
6            1       6
7            1       7
8            1       8
9            1       9
10           1      10
attr(,"assign")
[1] 0 1
> X
      [,1] [,2]
 [1,]    1    1
 [2,]    1    2
 [3,]    1    3
 [4,]    1    4
 [5,]    1    5
 [6,]    1    6
 [7,]    1    7
 [8,]    1    8
 [9,]    1    9
[10,]    1   10
```

Enfin, la commande `sqrt(diag(S))` renvoie la déviation standard des paramètres ajustés par le modèle. On vérifie que le résultat correspond à `summary(g)$coefficients[,2]`

```
> summary(g)$coefficients[,2]
(Intercept)     c(1:10)
 0.20925506  0.02893109
> sqrt(diag(S))
[1] 0.20925508 0.02893109
```

4.6 Exercice 4.4

La procédure itérative s'arrête après 5 itérations, ce qui est suffisant dans notre exemple. Comment le code devrait être modifié après un nombre fixe d'itérations ou qu'un état "convergence" est été atteint, en définissant que l'ensemble des coefficients exhibent un changement relatif de 10^{-6} ou moins ? Utiliser `break` pour quitter une boucle et définir l'état "convergence" comme `abs(a-old.a) / (abs(a) + 0.1) < tol` pour un une quantité `a`.

Dans l'exemple, la procédure itérative s'arrête après 5 itérations. Nous modifions la procédure itérative en fixant un nombre d'itérations plus élevé (par exemple 20), en rajoutant un objet `old.b` au début de la boucle et une condition d'arrêt avec la fonction `if()` et la commande `break` pour quitter la boucle :

```
> for(it in 1:20){
+         old.b <- b
+         eta <- X %*% b
+         mu <- exp(eta)
+         diag(Omega) <- mu
+         z <- X %*% b + (y-mu)/mu
+         S <- solve(t(X) %*% Omega %*% X)
+         b <- S %*% t(X) %*% Omega %*% z
+         if (all((abs(b - old.b)/abs(old.b + 1e-6) < 1e-6))) break
+         cat("it=", it, b[1], b[2], "\n")
+         }
it= 1 1.949202 0.1414127
it= 2 1.861798 0.1511522
it= 3 1.859193 0.1514486
it= 4 1.859191 0.1514489
```

4.7 Exercice 4.5

Si nous souhaitons ajuster un modèle linéaire ordinaire avec une fonction de lien identité, quels changements devons-nous apporter au code R ? Quels sont les résultats ?

Lorsque nous utilisons une distribution Noramle avec une fonction de lien standard (identité), l'algorithme IRWLS renvoie les estimations obtenus par les moindres carrés ordinaires.

```
> X <- cbind(rep(1,10),1:10)
> y <- c(14,0,8,8,16,16,32,18,28,22)
> z <- y
> Omega <- diag(10)
> b <- solve(t(X) %*% Omega %*% X) %*% t(X) %*% Omega %*% z
> cat("Start:", b[1], b[2], "\n")
Start: 3.2 2.363636
> for(it in 1:20){
+         old.b <- b
+         eta <- X %*% b
+         mu <- eta
+         diag(Omega) <- 1
```

```
+         z <- X %*% b + (y-mu)
+         S <- solve(t(X) %*% Omega %*% X)
+         b <- S %*% t(X) %*% Omega %*% z
+         if (all((abs(b - old.b)/abs(old.b + 1e-6) < 1e-6))) break
+         cat("it=", it, b[1], b[2], "\n")
+         }
```

Il n'y a pas d'itérations. On peut vérifier que les résultats correspondent à ceux obtenus avec les moindres carrés ordinaires :

```
> lm(y ~ c(1:10))

Call:
lm(formula = y ~ c(1:10))

Coefficients:
(Intercept)      c(1:10)
      3.200        2.364
```

Références

Kaas, R., Goovaerts, M. J., Dhaene, J., and Denuit, M. (2008). *Modern Actuarial Risk Theory – Using R*. Berlin Heidelberg : Springer Verlag, second edition.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society*, **135**, 370–384.