Introduction aux modèles linéaires généralisés

Méthodologie et concepts généraux

Chapitre II

Anouar El Ghouch

LSBA, Université catholique de Louvain, Belgium

Dans un modèle de régression quelconque, l'objectif est d'expliquer et/ou de prédire une réponse (variable à expliquer, disons Y) à l'aide d'un ou plusieurs prédicteurs (variables explicatives) continus ou discrets (catégorielles).

Le choix du modèle dépend, entre autres, de la nature de la variable à expliquer.

Dans le cas de la régression linéaire classique (simple), Y est supposée être une variable continue. La validité d'un tel modèle repose sur les hypothèses suivantes:

1. $E(Y|X) = \beta_0 + \beta_1 X$;
2. $Var(Y|X) = \sigma^2$ (une constante qui ne varie pas avec X);
3. $Y|X \sim Normale$.

Ce qui est équivalent à écrire que $Y|X \sim N(\mu(X), \sigma^2)$, avec $\mu(X) := E(Y|X) = \beta_0 + \beta_1 X$.

Il arrive souvent, en pratique, que ces conditions ne soient pas remplies.
Comme, par exemple, le cas où:

- Y est une variable binaire (dichotomique).
- Y est une variable de comptage.

Dans ces deux cas (et d'autres encore) le modèle linéaire ne peut pas convenir, et ce pour de multiples raisons :

- Considérer Y comme une variable continue suppose des valeurs arbitraires (avec des décimales) entre un minimum et un maximum donnés. Ceci n'est clairement pas le cas, par exemple, d'une variable binaire où les seules valeurs possibles sont 0 ou 1 ou d'une variable de comptage où seules quelques valeurs entières (positives ou nulles) peuvent être mesurées.

» L'écriture $E(Y|X) = \beta_0 + \beta_1 X$ n'est pas compatible avec la nature de Y. Par exemple, dans le cas d'une variable binaire $E(Y|X) = P(Y = 1|X)$, et donc écrire que $E(Y|X) = \beta_0 + \beta_1 X$ est équivalent à écrire que $P(Y = 1|X) = \beta_0 + \beta_1 X$. Or, il y a une incohérence entre la partie droite et gauche de cette équation puisque $P(Y = 1|X) \in (0, 1)$ alors que, en général, $X \in (-\infty, \infty)$. Le même problème se pose dans le cas d'une variable de comptage puisque dans ce cas $E(Y|X) \geqslant 0$.

» L'homogénéité de la variance ($Var(Y|X) = \sigma^2$) n'est pas compatible non plus avec la nature de Y. Par exemple, dans le cas d'une variable binaire, $Var(Y|X) = P(Y = 1|X)(1 - P(Y = 1|X))$ ne peut pas être constante sauf dans le cas où X et Y sont indépendantes.

» Considérer Y comme suivant une loi normale suppose une distribution symétrique autour de sa moyenne, alors que, souvent en pratique, on observe des données réparties de façon fort asymétrique avec, par exemple, une tendance à prendre plus fréquemment de petites valeurs que de grandes valeurs (ou l'inverse).

Le modèle linéaire généralisé (Generalized Linear Model (GLM) en anglais) permet de remédier à ces problèmes en introduisant quelques modifications majeures aux équations et hypothèses du modèle linéaire classique :

1. Dans un GLM, la réponse ne doit pas être nécessairement une variable continue.
2. Dans un GLM, on ne suppose pas une relation linéaire "directe" entre Y et X, càd on ne suppose pas que $E(Y|X) = \beta_0 + \beta_1 X$. Au lieu de quoi, on suppose une hypothèse plus générale de type

$$g(E(Y|X)) = \beta_0 + \beta_1 X,$$

où g est une fonction dite de lien (link function, en anglais). Les fonctions de lien les plus fréquemment utilisées sont:

- la fonction identité $g(x) = x$, souvent utilisée quand $Y \sim Normale$;
- la fonction logarithmique $g(x) = \log(x)$, souvent utilisée quand $Y \sim Poisson$;
- la fonction logit $g(x) = \log(x/(1-x))$, souvent utilisée quand $Y \sim Bernoulli/Binomial$.

3. L'homogénéité de la variance n'a pas besoin d'être satisfaite.
4. Y n'a pas besoin d'être distribuée normalement, mais on suppose une distribution qui fait partie d'une famille beaucoup plus générale connue sous le nom de la famille exponentielle. Cette famille comprend, entre autres, les distributions suivantes: Bernoulli, Binomiale, Poisson, Binomiale négative, Multinomiale, Normale, Exponentielle, Gamma, Beta, et Inverse-Gaussienne.

Comme c'est typiquement le cas, ces avantages entrainent quelques "inconvénients":

» L'estimation d'un GLM ne peut pas se faire via la méthode des moindres carrés ordinaire utilisée dans le cadre de la régression linéaire classique. À la place, il faut utiliser le maximum de vraisemblance.

» Les coefficients deviennent, en général, plus difficiles à calculer et interpréter. Pour le calcul, on a recours à des méthodes numériques itératives, et pour l'interprétation on doit faire appel à des notions telles que le rapport des cotes (pour la régression logistique).

» L'inférence se base sur des approximations valides uniquement pour de grandes tailles d'échantillon (théorie asymptotique) et ne peut donc être appliquée que dans un contexte où l'on juge que l'on dispose de suffisamment d'observations.

» La visualisation, le diagnostic, l'analyse des résidus et l'analyse de la qualité d'ajustement d'un modèle GLM ne sont pas toujours des sujets faciles à traiter. On ne peut pas appliquer directement les outils classiques $\rightarrow$ *des adaptations sont nécessaires.*

En résumé, la régression linéaire généralisée est un outil qui généralise le modèle linéaire classique. Il peut être utilisé dans de nombreuses situations, afin d'analyser une réponse qui n'est pas nécessairement continue ou normalement distribuée.

Dans la suite de ce chapitre, nous examinerons de manière générale la méthodologie de la régression linéaire généralisée. Les chapitres suivants seront consacrés au cas particulier où la réponse est de type comptage (régression de poisson) ou de type binaire (régression logistique).

Une variable Y ayant une fonction de densité (ou une fonction de masse de probabilité, dans le cas discret) pouvant s'écrire sous la forme

$$f(y;\theta,\phi) = a(y,\phi)\exp\left(\frac{y\theta - b(\theta)}{\phi}\right)$$

est dite membre de la famille exponentielle à dispersion (Exponential Dispersion Family (EDF)).

Dans cette expression,

» θ est appelé le paramètre canonique ou naturel.

» $\phi > 0$ est appelé le paramètre de dispersion (dispersion or scale parameter en anglais). Souvent, ce paramètre est connu (égal à 1) ou ne présente pas d'intérêt direct (paramètre de "nuisance").

» $a(\cdot) > 0$ et $b(\cdot)$ sont deux fonctions connues. b est appelée la fonction génératrice ou la fonction de normalisation ($\int f = 1$).

Pour une variable Y provenant d'une EDF de moyenne μ et de variance σ^2, il est facile de prouver que

$$\mu = b'(\theta), \text{ et}$$
$$\sigma^2 = \phi b''(\theta).$$

$\Rightarrow$ la fonction b' est strictement croissante en $\theta \Rightarrow \theta = b'^{-1}(\mu)$.

$\rightarrow$ Une EDF peut être formulée en termes de μ (et σ^2) au lieu de θ (et ϕ).

Distribution normale

$$f(y;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right\},\ \sigma^2 > 0$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{y^2}{2\sigma^2}\right\} \exp\left\{\frac{y\mu - \mu^2/2}{\sigma^2}\right\}$$

$\rightarrow$ EDF avec $\theta = \mu$, $\phi = \sigma^2$ et $b(\theta) = \theta^2/2$. Et nous avons que

$$E(Y) = b'(\theta) = \mu.$$
$$Var(Y) = \phi b''(\theta) = \sigma^2.$$

Distribution de Poisson

$$f(y;\mu) = \frac{\mu^y}{y!}e^{-\mu},\ \mu > 0$$
$$= \frac{1}{y!}\exp\{y\log(\mu) - \mu\}$$

$\rightarrow$ EDF avec $\theta = \log(\mu)$, $\phi = 1$, et $b(\theta) = e^\theta$.

$$E(Y) = b'(\theta) = \mu.$$
$$Var(Y) = \phi b''(\theta) = \mu.$$

La distribution Binomiale/Bernoulli ($n = 1$)

$$\begin{aligned} f(y;p) &= C_n^y p^y (1-p)^{n-y},\ 0 < p < 1 \\ &= C_n^y \exp\left\{ y \log\left(\frac{p}{1-p}\right) + n \log(1-p) \right\} \end{aligned}$$

$\rightarrow$ EDF avec $\theta = \log\left(\frac{p}{1-p}\right)$, $\phi = 1$ et $b(\theta) = n\log\left(1+e^{\theta}\right)$.

$$\begin{aligned} E(Y) &= b'(\theta) = np. \\ Var(Y) &= \phi b''(\theta) = np(1-p). \end{aligned}$$

La distribution Gamma

$$f(y;\mu,\nu) = \frac{1}{\Gamma(\nu)}\left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1}\exp\left(-\frac{\nu}{\mu}y\right)I(y>0),\ \mu,\nu>0$$
$$= \frac{\nu^{\nu}}{\Gamma(\nu)}y^{\nu-1}\exp\left(\frac{-y/\mu-\log(\mu)}{\nu^{-1}}\right)$$

$\rightarrow$ EDF avec $\theta = -\frac{1}{\mu}$, $\phi = \frac{1}{\nu}$ et $b(\theta) = -\log(-\theta)$.

$$E(Y) = b'(\theta) = \mu.$$
$$Var(Y) = \phi b''(\theta) = \mu^2/\nu.$$

On dispose de n observations (y_i, x_i), $i = 1, \ldots, n$, provenant d'un échantillon i.i.d. $(Y_i, \mathbf{X}_i)$. Où Y_i est la réponse et $\mathbf{X}_i = (1, X_{i,1}, \ldots, X_{i,d})$ est le vecteur des prédicteurs pour l'individu i. Dans un GLM, on suppose que

(1) Les Y_i proviennent de la même EDF (Normale, Poisson, Bernoulli,...) avec une moyenne μ_i et une dispersion ϕ.

(2) Il existe une fonction g et un vecteur $\boldsymbol{\beta}$ tels que $g(\mu_i) = \boldsymbol{\beta}^t \mathbf{x}_i = x_{i,0}\beta_0 + x_{i,1}\beta_1 + \ldots + x_{i,d}\beta_d$, $\forall i$, avec $x_{i,0} = 1$.

g est appelée fonction de lien. Elle met en relation la valeur moyenne de la réponse et la combinaison linéaire des prédicteurs.

Théoriquement, toute fonction monotone et différentiable peut être utilisée comme fonction de lien. En pratique, nous utilisons le plus souvent les fonctions dites canoniques ou naturelles.

Lien canonique

Soit une EDF de moyenne μ et de paramètre canonique θ. Le lien canonique de cette EDF est la fonction qui transforme μ en θ; càd la fonction $b'^{-1}(\cdot)$.

Exemples

Loi	θ en fonction de μ	lien canonique (default link in R)	l'inverse du lien canonique
Normale	$\theta = \mu$	$identité : x \mapsto x$	$identité$
Poisson	$\theta = \log(\mu)$	$x \mapsto \log(x)$	e^x
Bernoullli	$\theta = \log\left(\frac{p}{1-p}\right)$	$logit : x \mapsto \log\left(\frac{x}{1-x}\right)$	$logistic : x \mapsto \frac{e^x}{1+e^x}$
Gamma	$\theta = -\frac{1}{\mu}$	$inverse : x \mapsto \frac{1}{x}$	$inverse$

Les liens canoniques simplifient sensiblement les calculs nécessaires à l'estimation des paramètres. Ils facilitent aussi l'interprétation du modèle.

Exemples de GLM avec liens canoniques

Normale (ou Gaussienne)

1. $Y_i \sim N(\mu_i, \sigma^2)$
2. $\mu_i = \beta_0 + \beta_1 x_i$

Poisson

1. $Y_i \sim Pois(\mu_i)$
2. $\log(\mu_i) = \beta_0 + \beta_1 x_i$

Bernoullli

1. $Y_i \sim Ber(p_i)$
2. $\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i$

Gamma

1. $Y_i \sim Gamma(\mu_i, \nu)$
2. $\frac{1}{\mu_i} = \beta_0 + \beta_1 x_i$

Remarque Transformer la réponse et écrire, par exemple, $\log(Y_i) = \beta_0 + \beta_1 x_i + \epsilon_i$ n'est pas la même chose que d'écrire $\log E(Y_i) = \beta_0 + \beta_1 x_i$. Cette dernière équation, qui est équivalente à $E(Y_i) = \exp(\beta_0 + \beta_1 x_i)$, est plus intéressante car elle décrit, avec exactitude, comment la moyenne de Y_i varie en fonction de x_i. □

Vraisemblance et log-vraisemblance

L'estimation des paramètres d'un GLM se fait par la méthode du maximum de vraisemblance. Voici la vraisemblance d'un GLM

$$L = \prod_{i=1}^{n} a(Y_i, \phi) \exp\left(\frac{Y_i\theta_i - b(\theta_i)}{\phi}\right).$$

$\rightarrow$ La log-vraisemblance est donnée par

$$\ell = \frac{1}{\phi} \sum_i (Y_i\theta_i - b(\theta_i)) + \sum_i \log a(Y_i, \phi), \tag{1}$$

où $\theta_i = b'^{-1}(\mu_i)$ et $\mu_i = g^{-1}(\eta_i)$, avec $\eta_i := \boldsymbol{\beta}^t \boldsymbol{x}_i$.

Notez que nous sommes ici intéressés par l'estimation de $\boldsymbol{\beta}$, le vecteur des coefficients de notre modèle.

Notez également que si nous utilisons la fonction de lien canonique, càd si $g(\cdot) = b'^{-1}(\cdot)$, nous aurons $\theta_i = \boldsymbol{\beta}^t \boldsymbol{x}_i$, et la vraisemblance, en tant que fonction des β's, devient plus simple.

Fonction de score

Le score est défini comme la dérivée partielle de la log-vraisemblance par rapport aux paramètres du modèle, ici les β_j, $j = 0, \ldots, d$.

Soit $S_j = \partial \ell / \partial \beta_j$. En utilisant la règle de dérivation en chaîne (Chain-rule, en anglais), nous avons que

$$\begin{aligned} S_j &= \frac{1}{\phi} \sum_i \left(Y_i - b'(\theta_i) \right) \frac{\partial \theta_i}{\partial \beta_j} \\ &= \frac{1}{\phi} \sum_i \left(Y_i - \mu_i \right) \frac{\partial \mu_i}{\partial \beta_j} (b'^{-1})'(\mu_i) \\ &= \frac{1}{\phi} \sum_i \left(Y_i - \mu_i \right) \frac{\partial \eta_i}{\partial \beta_j} (g^{-1})'(\eta_i)(b'^{-1})'(\mu_i) \\ &= \frac{1}{\phi} \sum_i w_i \left(Y_i - \mu_i \right) x_{i,j}, \end{aligned}$$

où $w_i = \frac{(b'^{-1})'}{g'}(\mu_i)$. À noter que $w_i = 1$ si la fonction de lien canonique est utilisée.

On appelle $\mathbf{S} = (S_0, S_1, \ldots, S_d)^t$ le vecteur score.

Pour trouver l'estimateur de maximum de vraisemblance (EMV) de $\boldsymbol{\beta}$, il faut résoudre le système d'équations $\mathbf{S} = \mathbf{0}$, càd

$$\begin{aligned} & S_j = 0, \ \forall j = 0, 1, \ldots, d \\ \Leftrightarrow & \sum_{i=1}^{n} w_i \left(Y_i - \mu_i\right) x_{i,j} = 0, \ \forall j = 0, 1, \ldots, d. \end{aligned}$$

De façon générale, une solution explicite n'existe pas.

$\rightarrow$ On fait appel à des *méthodes numériques itératives* telles que la méthode de Newton-Raphson ou la méthode de scoring.

Soit $H_{jk} = \partial S_j / \partial \beta_k$, $j, k = 0, \ldots, d$. Nous avons que

$$\begin{aligned} H_{jk} &= \frac{1}{\phi} \sum_i \frac{\partial w_i}{\partial \beta_k} (Y_i - \mu_i) x_{i,j} - \frac{1}{\phi} \sum_i w_i \frac{\partial \mu_i}{\partial \beta_k} x_{i,j} \\ &= \frac{1}{\phi} \sum_i \frac{\partial w_i}{\partial \beta_k} (Y_i - \mu_i) x_{i,j} - \frac{1}{\phi} \sum_i \frac{w_i}{g'(\mu_i)} x_{i,j} x_{i,k}. \end{aligned}$$

$\mathbf{H} = [H_{jk}]$ est appelée la Hessienne. L'espérance (mathématique) de $-\mathbf{H}$ est appelée la matrice d'information de Fisher dont l'élément (j, k) est donné par

$$I_{jk} = -E(H_{jk}) = \frac{1}{\phi} \sum_i \frac{w_i}{g'(\mu_i)} x_{i,j} x_{i,k}.$$

Méthode de Newton-Raphson

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} - \mathbf{H}^{-1}(\boldsymbol{\beta}^{(s)}) \mathbf{S}(\boldsymbol{\beta}^{(s)}), \; s = 0, 1, 2, \ldots,$$

où $\boldsymbol{\beta}^{(s)}$ est l'estimation de $\boldsymbol{\beta}$ à l'itération s. $\mathbf{S}(\boldsymbol{\beta}^{(s)})$ et $\mathbf{H}(\boldsymbol{\beta}^{(s)})$ sont le Score et la Hessienne calculés en utilisant $\boldsymbol{\beta}^{(s)}$. $\boldsymbol{\beta}^{(0)}$, la valeur initiale de $\boldsymbol{\beta}$, doit être spécifiée pour démarrer l'algorithme. Ce dernier est stoppé lorsque l'estimation de $\boldsymbol{\beta}$ ne change plus (ou presque plus) entre deux itérations successives.

La méthode Newton-Raphson a tendance à être instable pour diverses raisons :

» Dans certain cas, $\mathbf{H}$ n'est pas (ou quasi-pas) inversible.

» La convergence est lente lorsque la valeur initiale $\boldsymbol{\beta}^{(0)}$ est "très éloignée" de $\boldsymbol{\beta}$.

» $\mathbf{H}^{-1}$ n'est pas toujours facile à calculer.

Une variante de la méthode Newton-Raphson est la méthode de scoring qui consiste à remplacer $-\mathbf{H}$ par la matrice d'information de Fisher $\mathbf{I} = [I_{jk}]$. Cette dernière est toujours inversible.

MÉTHODE DE SCORING

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} + \mathbf{I}^{-1}(\boldsymbol{\beta}^{(s)})S(\boldsymbol{\beta}^{(s)}),\ s = 0, 1, 2, \ldots$$

Notez que dans le cas de liens canoniques, les deux méthodes (Newton-Raphson et scoring) coïncident.

Dans R, les GLM sont ajustés aux données à l'aide de la fonction `glm()` :

```
glm(formula, family = dist.name(link = "link.name"), data, ...)
```

» `formula` : sert à introduire la partie linéaire du modèle sous forme d'une formule R; plus de détails sont donnés ci-après.

» `family` : sert à indiquer la distribution et la fonction de lien. Par exemple pour choisir une distribution binomiale avec un lien logit, il faut écrire `family = binomial(link = "logit")`. Pour chaque distribution, nous pouvons choisir un lien dans une liste prédéfinie. Par exemple, la distribution gaussienne accepte les fonctions identity, log et inverse. Par défaut, le lien canonique est utilisé; voir `?family` pour plus de détails.

» `data` : sert à introduire le noms du `data.frame` contenant le jeux de données à analyser.

La partie linéaire d'un modèle GLM est spécifiée dans R à l'aide d'une formule (formula) de type `Réponse ~ Prédicteur(s)`. Quelques exemples sont donnés dans le tableau suivant.

R Formula	Description
`y ~ 1`	Modèle nul, càd avec l'intercept comme unique paramètre
`y ~ x + z - 1`	Modèle avec `x` et `z` comme prédicteurs et sans l'intercept
`y ~ x + z + x:z`	Modèle avec l'intercept, `x`, `z` et leur interaction; autres écritures possibles: `y ~ x * z` ou `y ~ (x + z)^2`
`y ~ (x + z + w)^3`	Modèle avec tous les termes jusqu'à l'interaction triple; autres écritures: `y ~ x + z + w + x:z + x:w + z:w + x:z:w` ou `y ~ x*z*w`
`y ~ x + I(x^2)`	Régression polynomiale avec `x` et x^2 comme prédicteurs
`log(y) ~ exp(x)`	Régression linéaire où `log(y)` est la réponse et `exp(x)` est le prédicteur
`y ~ .`	Inclure toutes les variables dans le jeu de données, fournit en argument `data`, autres que la variable mise à la gauche du `~`.

REMARQUE Dans une formule R, *les opérations arithmétiques* classiques telles que `+`, `-`, `*`, et `^` n'ont pas leur signification habituelle. Pour qu'une opération arithmétique soit traduite de manière conventionnelle, elle doit être mise dans la fonction `I()`. Par exemple `y ~ I(x + z)` est équivalente à `y ~ w` où `w = x + z`.
□

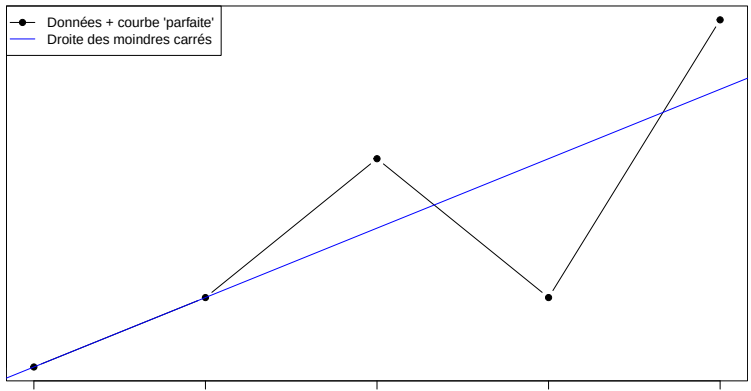
Exemple: Modèle Gaussien

```
x = 1:5; y = c(1, 2, 4, 2, 6)
dt <- data.frame(x = x, y = y)

mdg <- glm(y ~ x, family = gaussian, data = dt)
summary(mdg)

Call:
glm(formula = y ~ x, family = gaussian, data = dt)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.99e-15   1.48e+00    0.00     1.00
x           1.00e+00   4.47e-01    2.24     0.11

(Dispersion parameter for gaussian family taken to be 2)

    Null deviance: 16  on 4  degrees of freedom
Residual deviance:  6  on 3  degrees of freedom
AIC: 21.1

Number of Fisher Scoring iterations: 2
```

Exemple: Modèle Poisson

```
mdp <- glm(y ~ x, family = poisson, data = dt)
summary(mdp)

Call:
glm(formula = y ~ x, family = poisson, data = dt)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.0736     0.7676   -0.10    0.924
x             0.3508     0.1971    1.78    0.075 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5.1783  on 4  degrees of freedom
Residual deviance: 1.7602  on 3  degrees of freedom
AIC: 19.91

Number of Fisher Scoring iterations: 4
```

La déviance est un concept clé pour les modèles linéaires généralisés. La déviance vise à mesurer la "distance" séparant un modèle donné du modèle dit saturé. Ce dernier est le modèle qui s'ajuste "parfaitement" aux données, en ce sens que les valeurs ajustées $\hat{Y}_i$ sont égales aux réponses observées Y_i. La définition exacte du modèle saturé est donnée dans le Slide suivant. La figure ci-dessous montre des données avec la courbe linéaire ajustée et la courbe "parfaite".



Modèle saturé

Nous définissons le modèle saturé comme le GLM ayant la même structure que notre modèle d'intérêt mais pour lequel μ_i est libre de varier en fonction de i. Un tel modèle contient autant de paramètres à estimer que d'observations. De ce fait, le modèle saturé se caractérise par la plus grande vraisemblance parmi les GLM de même structure.

Pour mieux comprendre cela, revenons à la formule de la log-vraisemblance; voir (1). Considérons l'exercice qui consiste à maximiser cette quantité sans imposer de contraintes sur les μ_i, càd sans supposer que $\mu_i = g^{-1}(\boldsymbol{\beta}^t \mathbf{x}_i)$. Soit $\tilde{\mu}_i$ ($\tilde{\theta}_i$) l'EMV de μ_i (θ_i). Il est facile de voir que

$$\tilde{\mu}_i = b'(\tilde{\theta}_i) = Y_i.$$

Ainsi, la plus grande valeur que peut atteindre la log-vraisemblance est

$$\ell^S = \frac{1}{\phi} \sum_i \left(Y_i \tilde{\theta}_i - b(\tilde{\theta}_i) \right) + \sum_i \log a(Y_i, \phi).$$

Soit M un GLM quelconque dont le vecteur des coefficients β estimé par maximum de vraisemblance est $\hat{\beta}$. La log-vraisemblance maximale de M est

$$\ell^M = \frac{1}{\phi} \sum_i \left(Y_i \hat{\theta}_i - b(\hat{\theta}_i) \right) + \sum_i \log a(Y_i, \phi),$$

avec $\hat{\theta}_i = b'^{-1}(\hat{\mu}_i)$ et $\hat{\mu}_i = g^{-1}(x_i^t \hat{\beta})$.

Nous définissons la Déviance de M par

$$\begin{aligned} D^2_M &= -2\phi \{ \ell^M - \ell^S \} \\ &= 2 \sum_i \left(Y_i \left(\tilde{\theta}_i - \hat{\theta}_i \right) + \left(b(\hat{\theta}_i) - b(\tilde{\theta}_i) \right) \right). \end{aligned}$$

C'est essentiellement l'écart, mesuré en termes de vraisemblance, entre le modèle d'intérêt (M) et le modèle saturé (càd le modèle idéal qui s'ajuste "parfaitement" aux données). D^2 est une grandeur positive. *Plus elle est grande, moins bonne est la qualité d'ajustement du modèle M.* Notez que, par définition, la déviance du modèle saturé est 0.

Examinons le cas de la régression linéaire classique, càd le modèle Gaussien avec lien canonique. Dans ce cas, $b(\theta) = \theta^2/2$ et $g = b'^{-1} = \text{identité}$. Par conséquent, $\widehat{\theta}_i = \widehat{\mu}_i = \boldsymbol{x}_i^t \widehat{\boldsymbol{\beta}}$, et

$$\begin{aligned} D^2 &= 2 \sum_i \left(Y_i (Y_i - \widehat{\mu}_i) + \left(\widehat{\mu}_i^2/2 - Y_i^2/2 \right) \right) \\ &= \sum_i (Y_i - \widehat{\mu}_i)^2 \end{aligned}$$

Cette dernière quantité n'est rien d'autre que la *somme des carrés résiduels* (SCR).

$\rightarrow$ La Déviance est une généralisation du concept de somme des carrés résiduels tel qu'il est utilisé dans la régression linéaire classique.

La déviance peut également être utilisée pour définir/généraliser le concept du coefficient de détermination R^2.

Pour ce faire, notons d'abord que la valeur maximale de D^2 est atteinte pour le modèle ayant la plus petite log-vraisemblance. Ce modèle n'est autre que le *modèle sans variables explicatives*, càd le modèle dont $\mu_i = g^{-1}(\beta_0)$, $\forall i$.

Un tel modèle est appelé le modèle Nul. Sa déviance, appelée "Null deviance" (en anglais), est donnée par

$$D_0^2 = -2\phi(\ell^0 - \ell^S),$$

où $\ell^0 = \frac{1}{\phi}\sum_i \left(Y_i\hat{\theta}^0 - b(\hat{\theta}^0)\right) + \sum_i \log a(Y_i, \phi)$, avec $\hat{\theta}^0 = b'^{-1}(\overline{Y})$ et $\overline{Y} = n^{-1}\sum_i Y_i$.

Dans le cas de la régression linéaire classique,

$$D_0^2 = 2\sum_i \left(Y_i^2 - Y_i^2/2\right) - 2\sum_i \left(Y_i\overline{Y} - \overline{Y}^2/2\right) = \sum_i \left(Y_i - \overline{Y}\right)^2,$$

qui n'est rien d'autre que la *la somme des carrés totale* (SCT).

Comme SCT, D_0^2 est une mesure de la complexité des données.

Plus D_0^2 est élevé, plus Y varie, et plus il est difficile pour un modèle de saisir toutes ces variations. Au contraire, si D_0^2 est faible, les données varient peu autour de leur moyenne et sont donc faciles à modéliser.

Dans R, la déviance est renvoyée par `summary()`. R désigne la déviance comme la `"Residual deviance"` et la déviance nulle comme la `"Null deviance"` (voir Slide 24).

```
summary(mdp)$deviance
```

```
[1] 1.7602
```

```
summary(mdp)$null.deviance
```

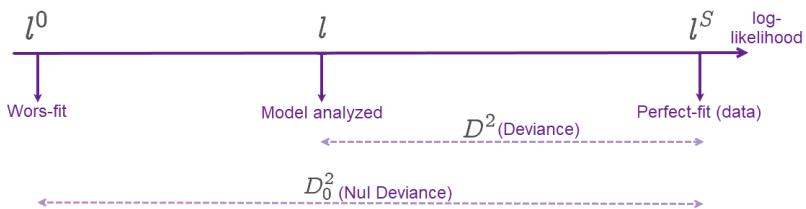
```
[1] 5.1783
```

Pseudo-R^2

Sur la base de la déviance et de la déviance nulle, il est possible d'évaluer la supériorité d'un modèle donné par rapport au modèle nul. L'idée est de prendre ce dernier comme référence ("le pire des cas") et de voir dans quelle mesure le modèle à l'étude fait mieux.

Cela peut se faire au moyen du pseudo-R^2 qui est une généralisation du coefficient de détermination classique R^2. Pour un modèle M dont la déviance est de D^2 et la log−vraisemblance est ℓ, on définit son pseudo-R^2 par

$$pR^2 = \frac{D_0^2 - D^2}{D_0^2} = \frac{\ell - \ell^0}{\ell^S - \ell^0}$$



Comme pour le R^2 classique (pour la régression linéaire), cette statistique est comprise entre 0 et 1.

Elle nous informe sur la qualité d'ajustement du modèle examiné en termes de vraisemblance. Ainsi,

$$pR^2 = 0 \Leftrightarrow \ell = \ell^0 : \text{ ajustement nul}$$
$$pR^2 = 1 \Leftrightarrow \ell = \ell^S : \text{ ajustement parfait}$$

$100 \times pR^2$ s'interprète comme le pourcentage de la déviance (nulle) expliquée par le modèle. En d'autres termes, $100 \times pR^2$ nous indique de quel pourcentage la déviance nulle a été réduite grâce au modèle étudié et à ses variables explicatives.

Remarque Comme pour un R^2 classique, ce pseudo-R^2 souffre de l'inconvénient suivant: l'ajout de prédicteurs, même non pertinents, dans le modèle entraînera toujours une augmentation (ou au moins pas de diminution) de sa valeur. □

Dans le `summary()` d'un GLM, R ne retourne pas pR^2. Ce dernier peut être calculé avec la fonction `pr2()` ; voir le document "Plan de cours".

```
# Pseudo-R-squared du modèle 'mdp'
pr2(mdp)
```

```
[1] 66.008
```

REMARQUE Pour un modèle linéaire Gaussien classique, $R^2 = pR^2$. Par exemple, pour notre modèle `"mdg"` ; voir Slide 23.

```
lm(y ~ x, data = dt) |> summary() |> _$r.squared
```

```
[1] 0.625
```

```
pr2(mdg)
```

```
[1] 62.5
```

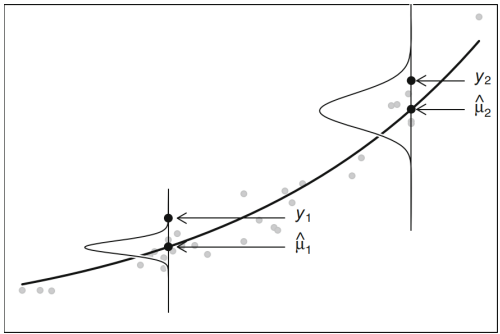
Statistique de Pearson

Il existe un certain nombre d'alternatives à la déviance, dont la plus populaire est la statistique (généralisée) de Pearson, définie par

$$X^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{\mu}_i)^2}{\widehat{Var}(Y_i)}, \text{ où } \hat{\mu}_i = g^{-1}(\hat{\beta}^t x_i).$$

Comme pour la déviance, plus X^2 est élevé, moins bonne est la qualité du modèle.

Notez que $\widehat{Var}(Y_i)$ varie, en général, en fonction de i.
Le fait de diviser par $\widehat{Var}(Y_i)$ permet de donner plus de poids/importance aux écarts $Y_i - \hat{\mu}_i$ lorsque Y_i a une grande précision (càd petite variance).



Dans R, la statistique de Pearson peut être calculée comme suit.

```
sum(residuals(mdp, "pearson")^2)
```

```
[1] 1.6718
```

Avant de passer à la suite, il convient de noter que

Les valeurs numériques des deux statistiques (Déviance et Pearson) diffèrent généralement, mais la différence est souvent marginale, en particulier pour les échantillons de grande taille.

En pratique, la déviance est préférable, pour les GLM, pour deux raisons: (1) La déviance est basée sur la méthodologie bien rodée de la vraisemblance. (2) La déviance peut être utilisée pour effectuer des tests statistiques afin de comparer deux modèles. La section suivante traite de ce sujet.

L'inférence pour les GLM repose sur de solides propriétés propres aux estimateurs du maximum de vraisemblance. Il s'agit essentiellement de résultats asymptotiques valables pour des échantillons de "grande taille".

Les EMVs sont consistants, asymptotiquement sans biais, asymptotiquement efficaces et asymptotiquement normalement distribués. Plus précisément, la théorie du maximum de vraisemblance nous dit que, si le modèle est correct, alors

$$\hat{\boldsymbol{\beta}} \sim_a \mathbf{N}_{(d+1)}(\boldsymbol{\beta}, \mathbf{I}^{-1}),$$

où $\mathbf{I}^{-1}$ est l'inverse de la matrice d'information de Fisher dont l'élément (j, k) est

$$I_{jk} = \frac{1}{\phi} \sum_i \frac{w_i}{g'(\mu_i)} x_{i,j} x_{i,k}.$$

avec $w_i = \dfrac{(b'^{-1})'}{g'}(\mu_i)$ et $\mu_i = g^{-1}(\boldsymbol{\beta}^t \mathbf{x}_i)$. Pour rappel, la notation $\sim_a$ se lit "asymptotiquement distribué".

En remplaçant $\boldsymbol{\beta}$ par $\hat{\boldsymbol{\beta}}$ dans $\mathbf{I}$, nous obtenons $\hat{\mathbf{I}}$, un estimateur consistant de $\mathbf{I}$.

Dans R, $\widehat{\mathbf{I}}^{-1}$ est renvoyée par la fonction `vcov()`.

```
vcov(mdp)

            (Intercept)         x
(Intercept)     0.58917 -0.142500
x              -0.14250  0.038864
```

Cette matrice fournit les $\hat{\sigma}_{jk} := \widehat{Cov}\left(\hat{\beta}_j, \hat{\beta}_k\right) =$ covariance estimée entre $\hat{\beta}_j$ et $\hat{\beta}_k$.

La racine-carrée de la diagonale de cette matrice donne les $\hat{\sigma}_j := \sqrt{\widehat{Var}(\hat{\beta}_j)} =$ écart-type estimé de $\hat{\beta}_j$.

```
vcov(mdp) |> diag() |> sqrt()

(Intercept)           x
    0.76757     0.19714
```

Ce sont là les mêmes valeurs que celles figurant dans la colonne "Std. Error" du tableau des coefficients fourni par `summary()` ; voir Slide 24.

Inférence via la méthode de Wald

Pour tester $H_0 : \beta_j = \beta_j^0$ vs $H_1 : \beta_j \neq \beta_j^0$ nous pouvons utiliser la méthode dite de Wald, basée sur la statistique $Z \equiv Z(\beta_j^0) := \dfrac{\hat{\beta}_j - \beta_j^0}{\hat{\sigma}_j}$.

Sous H_0, nous nous attendons à ce que Z soit proche de 0, alors que sous H_1, nous nous attendons à ce que $|Z|$ soit plutôt grande. La normalité asymptotique de $\hat{\boldsymbol{\beta}}$ implique que, sous H_0, $Z \sim_a N(0,1)$. Càd, si effectivement $\beta_j = \beta_j^0$, alors, pour une grande taille d'échantillon, la distribution de Z peut être approximée par une $N(0,1)$.

Ces deux dernières observations conduisent à la conclusion suivante: *Soit z la valeur observée de Z. Au niveau α, on rejettera H_0, en faveur de H_1, si $|z| > z_{1-\alpha/2}$, où $z_{1-\alpha/2}$ est le quantile $1-\alpha/2$ d'une $N(0,1)$. Ce qui revient à rejeter H_0 si la p-valeur $< \alpha$, où p-valeur $= P\big(|Z| > |z| \big| H_0\big) \approx P(|N(0,1)| > |z|)$.*

Le test de Wald est communément utilisé pour *tester la nullité individuelle des paramètres* et, en même temps, pour vérifier la significativité de chaque prédicteur du modèle (en présence des autres prédicteurs).

La fonction `glm()` effectue automatiquement ces tests de nullité et renvoie les résultats (la valeur z et la p-valeur) dans le tableau généré par `summary()` ; voir Slide 24.

REMARQUE Dans certaines situations, afin d'obtenir une meilleure approximation de la p-valeur, R utilise la distribution t de Student au lieu de la normale. C'est le cas par exemple du modèle Gaussien; voir Slide 23. □

La normalité asymptotique des coefficients estimés nous permet aussi de facilement construire des intervalles de confiance (IC) asymptotiques pour les β_j. L'IC de Wald à $100(1-\alpha)\%$ est donné par

$$\hat{\beta}_j \pm z_{1-\alpha/2}\hat{\sigma}_j.$$

Notez que cet IC coïncide avec la région de non-rejet (ou région d'acceptation) de H_0 qui est donnée par,

$$\{\beta : |Z(\beta)| \leqslant z_{1-\alpha/2}\}.$$

Par ailleurs, cet IC peut être utilisé pour tester $H_0 : \beta_j = \beta_j^0$ vs $H_1 : \beta_j \neq \beta_j^0$. Il suffit pour cela de vérifier si β_j^0 se trouve ou non dans cet intervalle. Dans ce dernier cas, H_0 est à rejeter (en faveur de $H_1 : \beta_j \neq 0$) au niveau α.

Voici, à titre d'exemple, comment obtenir, à l'aide de R, de tels intervalles pour les coefficients de notre modèle mdp; voir Slide 24.

```
# IC de Wald pour beta1
coef(mdp)[2] + c(-1, 1) * qnorm(1 - 0.05/2) * sqrt(vcov(mdp)[2, 2])

[1] -0.035625  0.737145

# IC de Wald pour beta0 et beta1 à l'aide de la fonction confint.default()
confint.default(mdp, level = 0.95)

                2.5 %  97.5 %
(Intercept) -1.577987 1.43084
x           -0.035625 0.73715
```

L'argument level de `confint.default()` est facultatif et vaut par défaut 0.95.

IC et transformations

Il arrive souvent que nous soyons intéressés par la construction d'IC pour des transformations de paramètres plutôt que pour les paramètres eux-mêmes; par exemple, trouver un IC pour $exp(\beta_1)$.

- Soit $[A, B]$ un IC pour un certain paramètre θ et soit h une fonction donnée. Si h est strictement croissante, alors un IC pour $h(\theta)$ est donné par $[h(A), h(B)]$.
- Une autre façon d'obtenir un IC pour $h(\theta)$ est d'utiliser la méthode Delta. Pour rappel, cette méthode dit que, à condition que h' existe et qu'elle soit continue,

$$\hat{\theta} \sim_a N\left(\theta, \hat{\sigma}^2\right) \Rightarrow h(\hat{\theta}) \sim_a N\left(h(\theta), \hat{\sigma}^2 \times \left(h'(\hat{\theta})\right)^2\right)$$

```
# IC pour exp(beta1) par simple transformation de l'IC pour beta1
mdp |> confint.default(parm = "x") |> exp()
```

```
  2.5 % 97.5 %
x 0.965   2.09
```

```
# IC pour exp(beta1) par la méthode Delta
exp(coef(mdp)[2]) + c(-1, 1) * qnorm(1 - 0.05 / 2) * sqrt(vcov(mdp)[2, 2]) * exp(coef(mdp)[2])
```

```
[1] 0.87142 1.96887
```

Inférence via la méthode LR

Une alternative à la méthode de Wald consiste à utiliser soit la méthode LR, soit la méthode Score. Ces méthodes sont plus fiables pour de petites tailles d'échantillons et elles sont aussi, souvent, plus faciles à généraliser/appliquer à toutes sortes de problèmes d'inférence. Dans ce qui suit, nous nous pencherons sur la méthode LR, qui est la plus couramment utilisée en pratique. Pour rappel, LR est le diminutif de Likelihood Ratio (rapport de vraisemblance).

Pour simplifier la présentation, considérons un modèle avec seulement deux paramètres (β_0, β_1) et dont la fonction de vraisemblance est $L(\beta_0, \beta_1)$. Nous souhaitons tester $H_0 : \beta_1 = \beta_1^0$ vs $H_1 : \beta_1 \neq \beta_1^0$. La statistique log-LR (*profilée*) est définie par

$$G^2 \equiv G^2(\beta_1^0) = -2 \log \frac{\max_{\beta_0} L(\beta_0, \beta_1^0)}{\max_{\beta_0, \beta_1} L(\beta_0, \beta_1)} = 2\Big(\ell(\widehat{\beta}_0, \widehat{\beta}_1) - \ell(\widehat{\beta}_0(\beta_1^0), \beta_1^0)\Big),$$

où $(\widehat{\beta}_0, \widehat{\beta}_1) = \arg \max_{\beta_0, \beta_1} L(\beta_0, \beta_1)$ et $\widehat{\beta}_0(\beta_1^0) = \arg \max_{\beta_0} L(\beta_0, \beta_1^0)$.

G^2 est toujours $\geqslant 0$. Sous H_0, nous nous attendons à ce que $G^2 \approx 0$, alors que sous H_1, nous nous attendons à ce que G^2 soit plutôt grande. En plus, la théorie du maximum de vraisemblance nous dit que, sous H_0, $G^2 \sim_a \chi^2_1$.

$\rightarrow$ *Soit* g^2 *la valeur observée de* G^2. *Au niveau* α, *on rejettera* H_0, *en faveur de* H_1, *si* $g^2 > \chi^2_{1,1-\alpha}$, *où* $\chi^2_{1,1-\alpha}$ *est le quantile* $1-\alpha$ *d'une* χ^2_1. *Et la* p*−valeur*$= P\big(G^2 > g^2 \big| H_0\big) \approx P(\chi^2_1 > g^2)$.

La statistique G^2 peut être formulée de manière générale comme suit

$$G^2 = 2\big(\ell^{NR} - \ell^R\big) = \frac{D^2_R}{\phi} - \frac{D^2_{NR}}{\phi},$$

oú l^{NR} et D^2_{NR} sont la log-vraisemblance maximale et la déviance du modèle Non-Restreint (càd le modèle sans la/les contrainte(s) imposée(s) par H_0), et ℓ^R et D^2_R sont la log-vraisemblance maximale et la déviance du modèle Restreint (càd le modèle avec la/les contrainte(s) imposée(s) par H_0.). ϕ est le paramètre de dispersion. *Si ce dernier vaut* 1, *comme par exemple c'est le cas pour la loi de Bernoullli et la loi de Poisson, alors* $G^2 = D^2_R - D^2_{NR}$.

```r
mdp <- glm(y ~ x, family = poisson, data = dt)  # <--- Modèle non-restreint
mdp0 <- glm(y ~ 1, family = poisson, data = dt)  # <--- Modèle restreint

# tester H0: beta1 = 0
g2 <- summary(mdp0)$deviance - summary(mdp)$deviance
# or# g2 <- 2 * (logLik(mdp) - logLik(mdp0))

# p-value
pchisq(g2, 1, low = F)
```

```
[1] 0.064487
```

La fonction `anova()` facilite ce type de calculs.

```r
anova(mdp0, mdp, test = "LRT")
```

```
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         4       5.18
2         3       1.76  1     3.42    0.064 .
---
```

La colonne "Resid. Df" donne le nombre d'observations (ici 5) moins le nombre de paramètres estimés pour chaque modèle. "Resid. Dev" donne la déviance de chaque modèle. Les colonnes "Df" et "Deviance" contiennent les différences entre la première et la deuxième ligne. "Pr(>Chi)" donne la p−valeur (ici $P(\chi^2_1 > 3.42)$).

Comme pour la méthode de Wald, nous pouvons construire des ICs en utilisant la méthode de vraisemblance. Par exemple, un IC pour β_1 qui découle naturellement du test LR décrit plus haut est donné par $\left\{\beta : G^2(\beta) \leqslant \chi^2_1(1-\alpha)\right\}$. Ce qui n'est rien d'autre que la région de non-rejet, par la méthode LR, du test $H_0 : \beta_1 = \beta_1^0$ vs $H_1 : \beta_1 \neq \beta_1^0$. Nous pouvons montrer que cette région est effectivement un intervalle.

Les ICs ainsi définis sont généralement plus difficiles à calculer, car ils n'ont pas toujours une forme explicite, ce qui nécessite l'utilisation d'algorithmes numériques pour trouver leurs limites. Toutefois, ces intervalles sont plus performants (meilleur couverture, ne produisent pas de valeurs aberrantes, ...).

Dans R, c'est la fonction `confint()` qui permet d'obtenir les ICs construits par la méthode LR.

```
# IC LR pour beta1 à l'aide de la fonction confint()
confint(mdp, parm = "x")
```

```
Waiting for profiling to be done...
```

```
    2.5 %    97.5 %
-0.020535  0.764163
```

Le test LR s'applique aussi pour tester plusieurs paramètres simultanément. La démarche est la même: *on compare la log-vraisemblance du modèle non restreint (NR) à celle du modèle restreint (R)*; si cette différence est jugée significative alors le modèle restreint est à rejeter en faveur du modèle non restreint.

Le seul élément qui change est le nombre de degrés de liberté de la loi χ^2 utilisée comme référence: il faut utiliser la loi χ^2 à p degrés de liberté, où

$$p = \textit{nbr. param. à éstimer dans NR} - \textit{nbr. param. à éstimer dans R}$$

Exemple: Tester $H_0 : \beta_1 = \beta_2 = 0$ vs $H_1 : \beta_1 \neq 0$ ou $\beta_2 \neq 0$ dans le modèle de Poisson $Y \sim X + X^2$.

```
mdp2 <- glm(y ~ x + I(x^2), family = poisson, data = dt)  # <--- Modèle non-restreint
anova(mdp0, mdp2, test = "LRT")

Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x + I(x^2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         4       5.18
2         2       1.76  2     3.42     0.18
```

L'approche basée sur le rapport de vraisemblance peut être utilisée pour effectuer toutes sortes de tests. Voici deux autres exemples.

Soit le modèle $Y \sim X_1 + X_2$. Nous souhaitons déterminer si l'effet de X_1 et celui de X_2 sur Y peuvent être considérés comme identiques. Un tel test peut être exprimé comme $H_0 : \beta_1 = \beta_2$ vs $H_0 : \beta_1 \neq \beta_2$.

```
dt12 <- data.frame(x1 = 1:5, x2 = c(14, 44, 27, 48, 35), y = c(1, 2, 4, 2, 6))
mdp12 <- glm(y ~ x1 + x2, family = poisson, data = dt12)
mdp120 <- glm(y ~ I(x1 + x2), family = poisson, data = dt12)
anova(mdp120, mdp12, test = "LRT")

Analysis of Deviance Table

Model 1: y ~ I(x1 + x2)
Model 2: y ~ x1 + x2
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         3       4.98
2         2       1.23  1     3.76    0.053 .
---
```

Un autre exemple fréquent est de vérifier si un paramètre peut être fixé à une valeur particulière (autre que 0). Par exemple, tester si $\beta_1 = 0.8$ dans le modèle $Y \sim X$. Pour cela, nous pouvons utiliser la fonction `offset()`. Dans un modèle de régression, un offset désigne tout simplement un terme dont le coefficient est connu pour être 1 et qui ne doit donc pas être estimé.

```
mdp08 <- glm(y ~ offset(0.8 * x), family = poisson, data = dt)
anova(mdp08, mdp, test = "LRT")

Analysis of Deviance Table

Model 1: y ~ offset(0.8 * x)
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         4       6.24
2         3       1.76  1     4.48    0.034 *
---
```

Comme mentionné auparavant, une autre approche pour faire de l'inférence sur des GLM est *la méthode du Score (appelée aussi la méthode de Rao)*. Brièvement, dans un modèle avec $\beta \in \mathbb{R}^d$ comme paramètres, pour tester $H_0 : \beta = \beta^0$ vs $H_1 : \beta \neq \beta^0$, la statistique du score est donnée par

$$U^2 \equiv U^2(\beta^0) = S^t(\beta^0) I^{-1}(\beta^0) S(\beta^0),$$

où S est le score et I est la matrice d'information de Fisher.

Sous H_0, nous nous attendons à ce que $U^2 \approx 0$, alors que sous H_1, nous nous attendons à ce que U^2 soit plutôt grande. En plus, la théorie du maximum de vraisemblance nous dit que, sous H_0, $U^2 \sim_a \chi^2_d$.

```
anova(mdp0, mdp, test = "Rao")

Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance  Rao Pr(>Chi)
1         4       5.18
2         3       1.76  1     3.42 3.33    0.068 .
---
```

L'un des principaux objectifs d'un modèle statistique est de prédire la réponse (ou, plus précisément, la réponse moyenne) en fonction des valeurs (nouvellement) observées des variables prédictives.

Soit $\boldsymbol{x} = (x_0, x_1, \ldots, x_d)^t$, où $x_0 = 1$, une valeur donnée de $\mathbf{X}$. Sur base du modèle GLM, nous pouvons estimer l'espérance conditionnelle de $Y|\mathbf{X} = \boldsymbol{x}$ par $\hat{\mu} = g^{-1}(\hat{\eta})$, où $\hat{\eta} = \hat{\boldsymbol{\beta}}^t \boldsymbol{x} = \sum_{j=0}^{d} \hat{\beta}_j x_j$, et $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_d)^t$ est le vecteur des coefficients estimés.

Le calcul de $\hat{\eta}$ et/ou de $\hat{\mu}$ peut facilement être effectué dans R à l'aide de la fonction `predict()` dont la syntaxe générale est la suivante.

```
predict(object, newdata, type, se.fit, ...)
```

- `object` : objet de type glm (résultat de l'appel à la fonction `glm()`).
- `newdata` : `data.frame` avec les $\boldsymbol{x}$ pour lesquels Y est à prédire.
- `type` : type de prédiction. *Par défaut* `type = "link"` *ce qui renvoi* $\hat{\eta}$. Pour obtenir $\hat{\mu}$, il faut utiliser `type = "response"` .
- `se.fit` : calculer ou non des écarts types des prédictions. Par défaut `se.fit = FALSE` .

```r
dt12New <- expand.grid(x1 = c(1.5,2.3), x2 = c(14.5, 35))

# estimer eta = beta0 + beta1*x1 + beta2*x2
predict(mdp12, dt12New)
```

```
      1       2       3       4
0.73527 1.07172 0.30660 0.64305
```

```r
# estimer mu = exp(eta)
predict(mdp12, dt12New, type = "response")
#or# predict(mdp12, dt12New)  |> exp()
```

```
     1      2      3      4
2.0861 2.9204 1.3588 1.9023
```

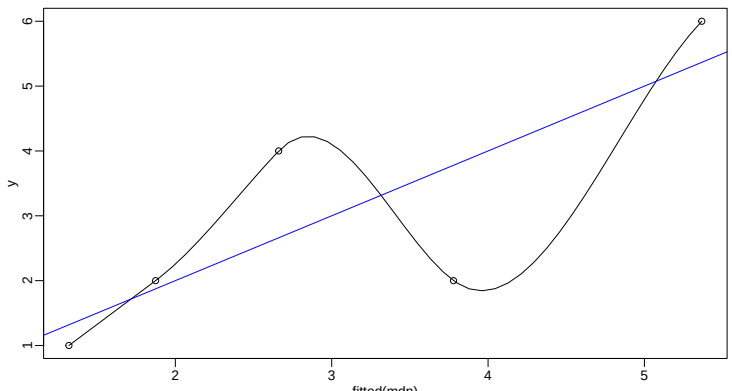
On peut aussi faire des prédictions sur les x_i observées; on obtient ainsi $\hat{y}_i := \hat{\mu}_i = g\left(\hat{\boldsymbol{\beta}}^t \mathbf{x}_i\right)$, les valeurs ajustées (fitted values, en anglais) qui interviennent dans la formulation des résidus.

```r
fitted(mdp12)
# or# predict(mdp12, type = 'response')
```

```
     1      2      3      4      5
1.7082 1.3891 3.0184 2.9629 5.9213
```

En comparant les valeurs ajustées aux valeurs observées, on peut se faire une idée sur la qualité du modèle. Cette comparaison se fait graphiquement à l'aide du diagramme de points $(y_i, \hat{y}_i)$ auquel nous ajoutons une courbe de lissage et la droite $y = x$. Nous reviendrons plus loin sur l'analyse des résidus.

```
scatter.smooth(fitted(mdp), y)
abline(a = 0, b = 1, col = "blue")
```



Intervalle de confiance pour les prédictions

Le calcul des ICs pour η peut se faire "manuellement" à partir des écarts types renvoyés lorsque `se.fit = TRUE`. Cette dernière commande donne

$$\hat{\sigma}^2 := \sum_{j=0}^{d} x_j^2 \widehat{Var}(\hat{\beta}_j) + 2 \sum_{j=0}^{d} \sum_{k:k>j}^{d} x_j x_k \widehat{Cov}(\hat{\beta}_j, \hat{\beta}_k).$$

Puisque, $\hat{\eta} \sim_a N(\eta, \hat{\sigma}^2)$, un IC pour η est donné par $\hat{\eta} \pm z_{1-\alpha/2}\hat{\sigma}$.

À partir de là, il est possible de construire un IC pour $\mu = g^{-1}(\eta)$ en suivant l'une des deux approches décrites dans le Slide 41:

- Par transformation directe: $\left[g^{-1}(\hat{\eta} - z_{1-\alpha/2}\hat{\sigma}), g^{-1}(\hat{\eta} + z_{1-\alpha/2}\hat{\sigma})\right]$.
- Par la méthode Delta: $g^{-1}(\hat{\eta}) \pm z_{1-\alpha/2}\hat{\sigma}\left|(g^{-1})'(\hat{\eta})\right|$.

Dans le cas, par exemple, de la régression de Poisson avec le lien canonique, $g = \log$, $g^{-1} = \exp$ et $(g^{-1})' = \exp$.

```r
# IC pour mu par simple transformation de l'IC pour eta
prd.eta <- predict(mdp12, dt12New, se.fit = TRUE)
cbind(fit = prd.eta$fit) |> transform(lwr = fit - 1.96 * prd.eta$se.fit,
  upr = fit + 1.96 * prd.eta$se.fit) |> exp() |> cbind(dt12New)
```

```
     fit     lwr    upr  x1   x2
1 2.0861 0.64944 6.7006 1.5 14.5
2 2.9204 0.96114 8.8736 2.3 14.5
3 1.3588 0.46798 3.9453 1.5 35.0
4 1.9023 0.86874 4.1654 2.3 35.0
```

```r
# IC pour mu par la méthode Delta
prd.mu <- predict(mdp12, dt12New, type = "response", se.fit = TRUE)
cbind(fit = prd.mu$fit) |> transform(lwr = fit - 1.96 * prd.mu$se.fit,
  upr = fit + 1.96 * prd.mu$se.fit) |> cbind(dt12New)
```

```
     fit       lwr    upr  x1   x2
1 2.0861 -0.348211 4.5203 1.5 14.5
2 2.9204 -0.325202 6.1660 2.3 14.5
3 1.3588 -0.089585 2.8072 1.5 35.0
4 1.9023  0.411350 3.3932 2.3 35.0
```

La fonction `predictions()` du package marginaleffects peut être utilisée pour simplifier davantage les calculs.

```
predictions(mdp12, newdata = datagrid(x1 = c(1.5, 2.3), x2 = c(14.5, 35)))
```

```
  x1   x2 Estimate Pr(>|z|)   S 2.5 % 97.5 %
 1.5 14.5     2.09   0.2168 2.2 0.649   6.70
 1.5 35.0     1.36   0.5729 0.8 0.468   3.95
 2.3 14.5     2.92   0.0587 4.1 0.961   8.87
 2.3 35.0     1.90   0.1078 3.2 0.869   4.17
```

Avant de pouvoir utiliser correctement un modèle, que ce soit à des fins d'inférence ou de prédiction, les hypothèses sur lesquelles il repose doivent être vérifiées à l'aide de tests statistiques ou des outils de diagnostic graphique reposant sur l'analyse des résidus. C'est cette dernière approche qui est la plus couramment utilisée dans la pratique et que nous allons aborder ici.

Nous pouvons classer les problèmes potentiels liés aux GLM en quatre types:

- Distribution: choix inapproprié de la distribution de la réponse. Par exemple, supposer à tort que les données proviennent d'une distribution normale.
- Lien: choix inapproprié de la fonction de lien. Par exemple, utiliser la fonction d'identité à la place du logarithme.
- Linéarité: la partie linéaire du modèle est mal spécifiée: des termes/variables qui manquent ou qui nécessitent une transformation préalable.
- Outliers: Les valeurs aberrantes qui ne suivent pas le même schéma/modèle que la majorité des données.

Dans ce qui suit, nous nous concentrerons sur le troisième point (Linéarité), qui est souvent le plus problématique dans la pratique.

Pour détecter ces éventuels problèmes, plusieurs types de résidus existent dans la littérature des GLM, parmi lesquels il y a:

Résidus de réponse (response residuals) Ce n'est rien d'autre que $Y_i - \hat{\mu}_i$. En général, ces résidus ne conviennent pas à l'évaluation d'un GLM en raison de l'hétérogénéité de la variance.

Résidus de Pearson

$$r_i = \frac{Y_i - \hat{\mu}_i}{\sqrt{\widehat{Var}(Y_i)}}, \text{ avec } \hat{\mu}_i = g^{-1}(x_i^t \hat{\beta}).$$

Notez que $\sum_i r_i^2 = X^2$ (stat. de Pearson).

Pour un GLM fitté dans R, par exemple notre modèle `mdp`, les résidus de Pearson sont calculés à l'aide de `residuals(mdp, type = "pearson")`.

Résidus de la Déviance À partir de la Déviance, on définit les résidus

$$d_i = \text{sign}(Y_i - \hat{\mu}_i)\sqrt{2\Big(Y_i\Big(\tilde{\theta}_i - \hat{\theta}_i\Big) + \Big(b(\hat{\theta}_i) - b(\tilde{\theta}_i)\Big)\Big)},$$

où $\text{sign}(x) = I(x > 0) - I(x < 0)$. Les résidus de la déviance sont les plus utilisés en pratique. Notez que $D^2 = \sum_i d_i^2$.

Pour obtenir ces résidus dans R, il suffit de taper `residuals(mdp)`, où `mdp` est le nom du modèle ajusté.

Résidus de quantiles randomisés (RQR) Pour comprendre le concept qui sous-tend la définition de ces résidus, nous avons besoin de la propriété suivante.

Soit Y une va, F une fonction de distribution cumulée (cdf), U une va $\text{Uniform}(0,1)$ indépendante de Y, et Z la va définie par $Z = F(Y-) + U \times (F(Y) - F(Y-))$. Si F est la cdf de Y, alors Z suit une distribution $\text{Uniform}(0,1)$.

Dans le cadre d'un GLM, notons $\hat{F}_i(\cdot) \equiv F(\cdot; \hat{\mu}_i, \phi)$ la CFD, estimée à partir de l'échantillon, de la famille exponentielle stipulée (Poisson, Normale, etc.). Soit $U_i, i = 1, \ldots, n$, est un échantillon i.i.d. qui provient d'une $\text{Uniform}(0,1)$. Les RQR sont définis par

$$q_i = \hat{F}_i(Y_i-) + U_i \times \left(\hat{F}_i(Y_i) - \hat{F}_i(Y_i-)\right), \; i = 1, \ldots, n,$$

Si le modèle s'ajuste correctement aux données, ces résidus devraient se comporter comme s'il s'agissait d'un échantillon aléatoire provenant d'une $\text{Uniform}(0,1)$. Ce qui est équivalent à dire que $\Phi^{-1}(q_i), i = 1, \ldots, n$, *est un échantillon aléatoire d'une* $N(0,1)$.

À titre d'exemple, voici comment calculer "manuellement", dans R, les RQR pour le modèle `mdp`.

```
mu <- fitted(mdp)
set.seed(1)
{ppois(y - 1, mu) + runif(5, 0, 1) * (ppois(y, mu) - ppois(y - 1, mu))} |> qnorm()

[1] -0.35598  0.10436  0.86433 -0.65161  0.20991
```

La fonction `statmod::pqresiduals()` peut être utilisée pour facilement calculer ces résidus.

```
set.seed(1)
statmod::qresiduals(mdp)

[1] -0.35598  0.10436  0.86433 -0.65161  0.20991
```

D'une part, le graphique des résidus (Pearson ou Déviance ou quantiles randomisés) en fonction des valeurs ajustées ($\hat{\mu}_i$) et les graphiques des résidus en fonction de chacune des variables explicatives potentielles (lorsqu'il y en a plusieurs) sont les outils de diagnostic les plus fréquemment utilisés pour repérer d'éventuels *problèmes de linéarité ou de lien.*

Ces graphiques doivent montrer *des points répartis aléatoirement autour de* 0 *sans structure ou tendance particulières et sans valeurs extrêmes.*

D'autre part, lorsque la taille de l'échantillon est relativement grande, le QQ-plot des résidus peut être utilisé pour vérifier l'adéquation de la distribution choisie. À cet égard, les RQR sont les plus appropriés.

Si une anomalie est détectée, il n'est pas toujours facile d'en identifier la cause exacte (Distribution, Lien, ...). Dans tous les cas, l'un des remèdes suivants ou une combinaison de ceux-ci peut être envisagé:

- transformer les prédicteurs (ou la réponse) à l'aide de fonctions telles que $\log(.)$, $\sqrt{.}$, fonctions de puissance, ...,
- ajouter ou supprimer des prédicteurs et/ou des interactions,
- changer la fonction de lien, par exemple, dans le cas du Poisson, remplacer (`link = log` par (`link = sqrt` ou `link = identity`, ...),
- changer la distribution utilisée en faveur d'une autre distribution de la famille exponentielle ou d'un autre modèle plus souple (par exemple, estimation par la méthode dite de quasi-vraisemblance).

Pour faciliter la procédure de diagnostic, nous ferons appel à la fonction `diagnost()` qui figure dans le document "Plan de cours". Examinons le code suivant.

```
# simulated data
n <- 100
set.seed(2)
data <- data.frame(x1 = runif(100), x2 = runif(n)) |> transform(y = rpois(100, exp(1.2 + 2*x1 - 5*x2^2)))

# Model mdra1: missing the covariate x2
mdra1 <- glm(y ~ x1, family = poisson, data = data)
diagnost(mdra1, type = "rqr") # type = "rqr" --> randomized quantile residuals. These are the default residuals.
                              # other options: type = "deviance", "pearson", ...

# Model mdra2: the quadratic term is missing
mdra2 <- glm(y ~ x1 + x2, family = poisson, data = data)
diagnost(mdra2)

# Model mdra3: misspecified distribution
mdra3 <- glm(y ~ x1 + I(x2^2), family = gaussian, data = data)
diagnost(mdra3)

# Model mdra4: correced model
mdra4 <- glm(y ~ x1 + I(x2^2), family = poisson, data = data)
diagnost(mdra4)
```