La régression de Poisson

Chapitre III

Anouar El Ghouch

LSBA, Université catholique de Louvain, Belgium

Régression de Poisson simple

Régression de Poisson multiple

Sélection des variables et construction du modèle

Régression de Poisson pour tableaux de contingence:
Les modèles log-linéaire

Appendice

Régression de Poisson simple
Introduction
Notations et hypothèses
Estimation, prédiction et diagnostic

Régression de Poisson multiple

Sélection des variables et construction du modèle

Régression de Poisson pour tableaux de contingence:
Les modèles log-linéaire

Appendice

Régression de Poisson simple

Introduction

La régression de Poisson est un cas particulier du modèle linéaire généralisé, dans lequel la réponse (aléatoire) est caractérisée par la distribution de Poisson. Cette méthode fonctionne généralement bien lorsque les événements individuels comptabilisés sont indépendants (ou presque), et qu'il n'y a pas de limite supérieure claire pour le nombre d'événements qui peuvent se produire. Comme nous le verrons plus loin, la régression de Poisson permet également de modéliser les données de fréquences.

Dans un modèle de Poisson, comme dans n'importe quel modèle GLM, les variables explicatives peuvent être de n'importe quel type : continues ou discrètes/catégorielles ou un mélange des deux.

Lorsque toutes les variables explicatives sont catégorielles, le modèle de régression de Poisson est traditionnellement dénommé modèle log-linéaire. Nous étudierons aussi ce type de modèle dans le présent document.

Notations et hypothèses

Objectif

Expliquer une variable de comptage N, prenant des valeurs entières positives $0, 1, 2, 3, \ldots$, à l'aide d'une ou plusieurs variables explicatives. Pour rappel, une variable de comptage recense le nombre d'occurrences d'un événement d'intérêt au cours d'une certaine période de temps (ou d'espace).

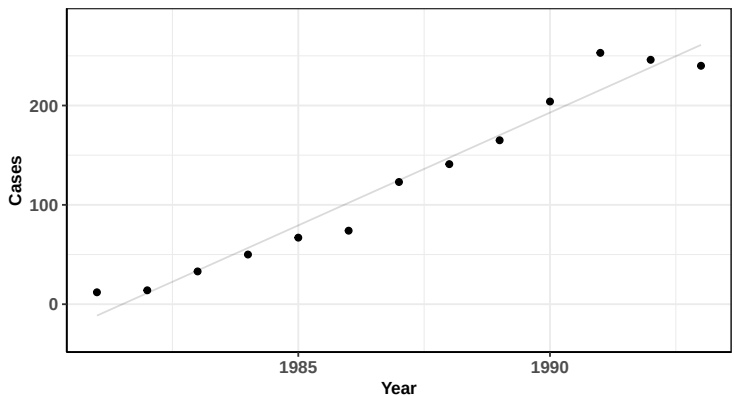
Dans un premiers temps, nous allons nous focaliser sur le cas d'une seule variable explicative continue tout en présentant une théorie générale qui couvre à la fois le cas d'une ou plusieurs variables explicatives.

Exemple (cas d'un seul prédicteur continu)

Nombre de nouveaux cas de sida en Belgique, 1981-1993 (`data = aids`)

N=Cases	Year=X
12	1981
14	1982
33	1983
50	1984
67	1985
74	1986
123	1987
141	1988
165	1989
204	1990
253	1991
246	1992
240	1993

```
Year <- c(1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990,
1991, 1992, 1993); Cases <- c(12, 14, 33, 50, 67, 74, 123, 141, 165,
204, 253, 246, 240); aids <- data.frame(Year, Cases)
```



On dispose de I observations (n_i, x_i), $i = 1, \ldots, I$, provenant d'un échantillon i.i.d. (N_i, X_i). On suppose que

$$\begin{aligned} &(1)\quad N|X \sim \text{Poisson}(\mu(X)), \text{ où } \mu(X) = E(N|X), \\ &(2)\quad \log(\mu(X)) = \beta_0 + \beta_1 X. \end{aligned}$$

Pour $i = 1, \ldots, I$, soit

$$\begin{aligned} p_i \equiv p(x_i) &:= P(N_i = n_i | X_i = x_i), \text{ et} \\ \mu_i \equiv \mu(x_i) &:= E(N_i | X_i = x_i). \end{aligned}$$

$$(1), (2) \Leftrightarrow p_i = \frac{\mu_i^{n_i}}{n_i!} e^{-\mu_i}, \text{ avec } \mu_i = \exp(\beta_0 + \beta_1 x_i).$$

Le vecteur $\boldsymbol{\beta} = (\beta_0, \beta_1)^t \in \mathbb{R}^2$ des vrais paramètres est inconnu.

Remarques

Dans la formulation ci-dessus, nous avons utilisé la fonction log comme fonction de liaison/lien entre la moyenne μ et le prédicteur linéaire $\beta_0 + \beta_1 X$. Il s'agit d'un choix et non d'une obligation!

D'un point de vue mathématique, c'est un choix naturel puisqu'il correspond à la fonction de lien canonique (pour un GLM-Poisson).

log est également le lien le plus couramment utilisé dans la pratique, car il conduit (1) à une interprétation relativement simple des paramètres estimés et (2) à des prédictions qui respectent bien l'échelle des données (systématiquement positives).

Avec la distribution de Poisson, la fonction `glm()` de R accepte comme liens les fonctions log (par défaut), identité et racine-carrée.

- $\exp(\beta_0) = E(N|X = 0)$, càd la moyenne de N (le nombre d'événements attendu) lorsque $X = 0$.
- $\exp(\beta_1) = \frac{E(N|X = x+1)}{E(N|X = x)}$ est l'effet multiplicatif, sur la moyenne de N, résultant d'une augmentation d'une unité de X.
  - $\beta_1 = 0 \Rightarrow E(N|X) = \exp(\beta_0) \Rightarrow$ X et N sont indépendantes.
  - $\beta_1 > (<)0 \Rightarrow \exp(\beta_1) > (<)1 \Rightarrow E(N|X)$ augmente (diminue) en fonction de X.
- Si la variable explicative X est binaire avec deux modalités 0 et 1, alors $\exp(\beta_1) = \frac{E(N|X = 1)}{E(N|X = 0)}$ est l'effet multiplicatif, sur la moyenne de N, résultant du changement de modalité/groupe de $\{X = 0\}$ à $\{X = 1\}$. Ce ratio est parfois appelé "rapport des taux d'incidence" (incidence rate ratio, en anglais).

De manière générale, pour d prédicteurs $X_1, \ldots, X_d$, soit $\mathbf{X} = (1, X_1, \ldots, X_d)^t$, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^t$, et $\mu(\mathbf{X}) = E(N|X_1, \ldots, X_d)$. On suppose que

$$N|\mathbf{X} \sim \text{Pois}(\mu(\mathbf{X})) \text{ et } \mu(\mathbf{X}) = \exp(\boldsymbol{\beta}^t \mathbf{X}).$$

De cette équation, on peut déduire que pour un nombre donné a,

$$\exp(a\beta_j) = \frac{\mu(x_1, \ldots, x_j + a, \ldots, x_d)}{\mu(x_1, \ldots, x_j, \ldots, x_d)}, \; j = 1, \ldots d, \text{ et}$$
$$\exp(\beta_0) = \mu(0, \ldots, 0).$$ Voir App.

- $\exp(\beta_0)$ est le nombre d'événements attendu lorsque toutes les X_j sont nulles.
- $\exp(a\beta_j)$ est le rapport de moyennes entre deux groupes différant de a unité(s) par rapport à X_j, et identiques par rapport à tous les autres prédicteurs.
- En particulier, quand X_j augmente d'une unité, tandis que les autres prédicteurs restent fixes, N change (augmente ou diminue) en moyenne par un facteur multiplicatif de $\exp(\beta_j)$.

Régression de Poisson simple

Estimation, prédiction et diagnostic

La log-vraisemblance, le Score et la Hessienne du modèle de Poisson sont

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{I} (N_i \log(\mu_i) - \mu_i - \log(N_i!))$$

$$S_j := \frac{\partial l}{\partial \beta_j} = \sum_i (N_i - \mu_i) x_{i,j}$$

$$H_{jk} := \frac{\partial S_j}{\partial \beta_k} = -\sum_i \mu_i x_{i,j} x_{i,k},$$

où $\log(\mu_i) = \boldsymbol{x}_i^t \boldsymbol{\beta} = \sum_{j=1}^{d} x_{i,j} \beta_j$, avec $\boldsymbol{x}_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,d})^t$, et $x_{i,0} = 1$.

À partir de là, nous pouvons calculer (numériquement) $\hat{\boldsymbol{\beta}} = \arg\max l(\boldsymbol{\beta})$, l'EMV de $\boldsymbol{\beta}$, faire de l'inférence sur celle-ci et effectuer des prédictions et des diagnostics, comme expliqué précédemment.

Aussi, en appliquant les définitions vues précédemment, il est facile de voir que (i) l'EMV de μ_i est $\hat{\mu}_i = \exp\left(\boldsymbol{x}_i^t \hat{\boldsymbol{\beta}}\right)$, (ii) pour le modèle saturé, l'EMV de μ_i est $\tilde{\mu}_i = N_i$, et (iii) pour le modèle nul, l'EMV de μ_i est $\hat{\mu}^0 = \overline{N} = I^{-1} \sum_i N_i$.

Nous en déduisons les statistiques suivantes.

$$D^2 = 2(l^s - l) = 2\sum_i (N_i \log(N_i/\hat{\mu}_i) - (N_i - \hat{\mu}_i)) \qquad \text{(Deviance)}$$

$$D_0^2 = 2\big(l^s - l^0\big) = 2\sum_i \big(N_i \log\big(N_i/\overline{N}\big) - (N_i - \overline{N})\big) \qquad \text{(Null Deviance)}$$

$$r_i = \frac{N_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}} \qquad \text{(Pearson residuals)}$$

$$d_i = sign(N_i - \hat{\mu}_i)\sqrt{2(N_i \log(N_i/\hat{\mu}_i) - (N_i - \hat{\mu}_i))} \qquad \text{(Deviance residuals)}$$

Enfin, il y a aussi les résidus de quantiles aléatoires, tels qu'ils ont été définis dans le chapitre précédent.

Nous allons maintenant étudier le jeu de données "aids" en mettant en pratique ce que nous avons appris jusqu'à présent.

Exemple "aids" (Slide 3)

Ajustement du modèle

```
reg1 <- glm(Cases ~ Year, family = poisson, data = aids)
## reg1 |> coef() |> exp()
## reg1 |> confint() |> exp()
```

					Exponentiated		
	Estimate	Std.Error	z.value	p.value	estimate	CI.lwr	CI.upr
(Intercept)	-397.059	15.462	-25.680	<0.001	0.000	0.000	0.000
Year	0.202	0.008	26.008	<0.001	1.224	1.206	1.243

Soit $\mu(Year)$ le nombre de nouveaux cas de sida attendu durant une année donnée $(Year)$ en Belgique. À partir des données dont on dispose, le modèle de Poisson estime que ce nombre est donné par l'équation suivante

$$\hat{\mu}(Year) = \exp(-397 + 0.2 \times Year).$$

Cette équation semble bien expliquer les variations des données comme l'atteste le pseudo-R^2 qui est ici de 90.7% .

Interprétation des paramètres

Le nombre moyen de cas de sida est estimé à $0 \approx \exp(-397)$ en l'an 0.
$\rightarrow$ *interprétation insensée*! Ceci dit, l'intercept a rarement un intérêt en soi.

Le nombre moyen de cas de sida se multiplie par $1.22 \approx \exp(0.202)$ chaque année.
$\rightarrow$ une augmentation annuelle de 22%.

Pour autant que le modèle soit correct, on peut conclure, avec un risque d'erreur de 5%, que le nombre moyen de cas de sida augmente de manière significative, dans une fourchette se situant entre 21% et 24%.

Remarque On peut rendre l'intercept "interprétable" en prenant une valeur de référence autre que 0. Pour cela, il suffit de soustraire la valeur de référence souhaitée à $Year$. Par exemple, prendre $X = Year - 1981$ comme variable explicative au lieu de $Year$. De cette manière l'exponentielle de la constante sera le nombre moyen quand $Year = 1981$.

```
glm(formula = Cases ~ I(Year - 1981), family = poisson)
```

$$\hat{\mu}(Year) = \exp(3.343 + 0.202 \times (Year - 1981))$$

Pour calculer des prédictions, et leurs intervalles de confiance, il y a la fonction (de base) `predict()`. Pour rappel, cette dernière renvoie, par défaut, les prédictions sur l'échelle du lien, càd qu'elle renvoie ici $\log(\hat{\mu}) = \hat{\beta}_0 + \hat{\beta}_1 x$. Pour avoir les prédictions sur l'échelle de la réponse, càd calculer $\hat{\mu}(x) = \exp\left(\hat{\beta}_0 + \hat{\beta}_1 x\right)$, le nombre moyen d'événements, il faut employer la transformation `exp()`, ou ajouter l'argument `type = "response"` à `predict()`.

Une autre façon pour calculer les $\hat{\mu}(x)$ est d'utiliser la fonction `predictions()` du package marginaleffects.

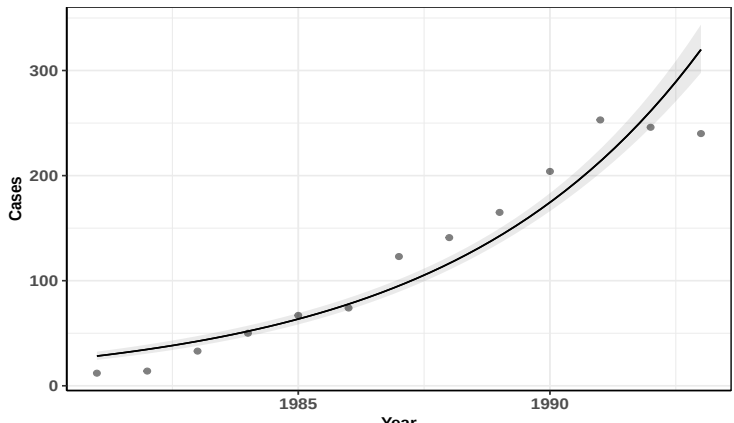
```
predictions(reg1, newdata = data.frame(Year = c(1980, 1981, 1994, 1995)))
```

```
 Estimate Pr(>|z|)   S 2.5 % 97.5 % Year
     23.1   <0.001 Inf  19.8   26.9 1980
     28.3   <0.001 Inf  24.6   32.5 1981
    391.6   <0.001 Inf 360.3  425.7 1994
    479.3   <0.001 Inf 435.4  527.8 1995
```

La courbe de régression et les résidus

Le package `marginaleffects` peut aussi être utilisé pour tracer la courbe $\hat{\mu}(Year) = \exp(-397 + 0.2Year)$ et son intervalle de confiance. La figure suivante montre ces éléments. Les points qui y figurent représentent les observations.

```
plot_predictions(reg1, condition = "Year", points = 0.5)
```



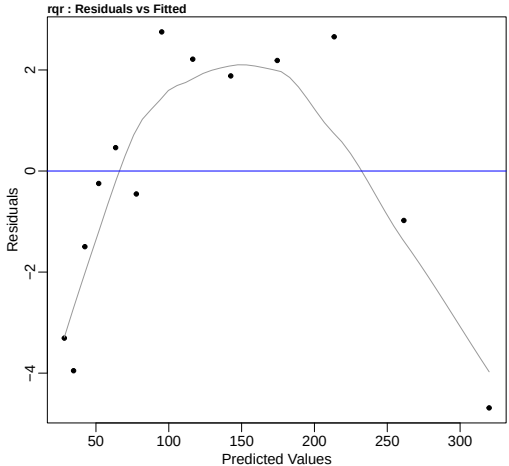
Remarque

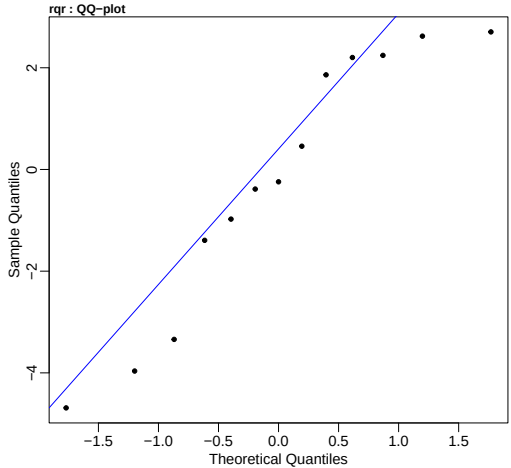
Il n'est pas nécessaire d'utiliser `marginaleffects` pour créer des graphiques de prédiction. Le code suivant montre d'autres façons d'obtenir plus ou moins le même graphique que celui ci-dessus.

```r
# base R (sans package à charger)
plot(Cases ~ Year, data = aids, pch = 19)
curve(exp(coef(reg1)[1] + coef(reg1)[2] * x), add = TRUE)

# avec le package ggplot2 (méthode 1)
ggplot(data = aids, aes(x = Year, y = Cases)) +
  geom_point(alpha= 0.5) +
  stat_function(fun=\ (x) exp(coef(reg1)[1] + coef(reg1)[2] * x)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), data = predictions(reg1), alpha = 0.1)

# avce le package ggplot2 (méthode 2)
ggplot(data = aids, aes(x = Year, y = Cases)) +
     geom_point(alpha = 0.5) +
     geom_smooth(method = "glm", method.args = list(family = "poisson"))
```

Examinons les résidus de notre modèle.

```
diagnost(reg1, plots = c("fitted", "qqplot"))
```





Il semble y avoir un problème dans l'ajustement du modèle aux données.

Modèle quadratique

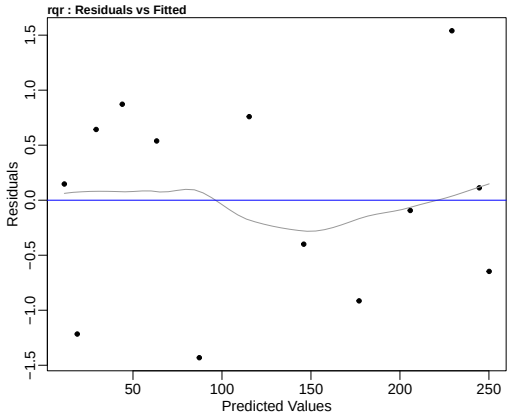
```
reg2 <- glm(Cases ~ Year + I(Year^2), family = poisson, data = aids)
summary(reg2)$coef
```

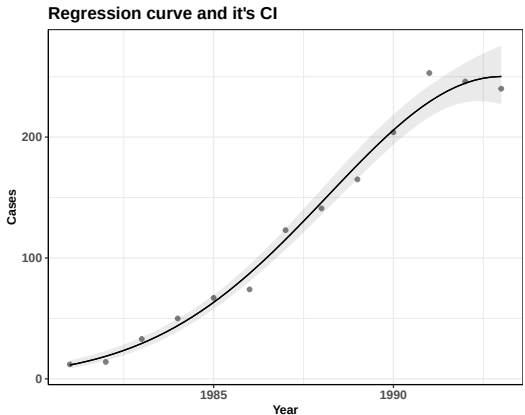
	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-84784.908	10511.963	-8.066	7.2898e-16
Year	85.087	10.573	8.048	8.4452e-16
I(Year^2)	-0.021	0.003	-8.029	9.8181e-16

Avec un pseudo-R^2 de 98.9%, ce modèle, dont l'équation est donnée par

$$\hat{\mu}(Year) = \exp\left(-84785 + 85.1 \times Year - 0.02 \times Year^2\right),$$

est significativement meilleur que le modèle simple (sans terme quadratique).





Contrairement au modèle "reg1", les résidus, qui sont ici tous compris entre -1.5 et 1.5, ne présentent pas de structure particulière. On peut dire qu'ils oscillent aléatoirement autour de 0. Aussi, le QQ-plot (non affiché ici) ne montre aucun problème.

Et pourquoi pas un modèle cubique ?

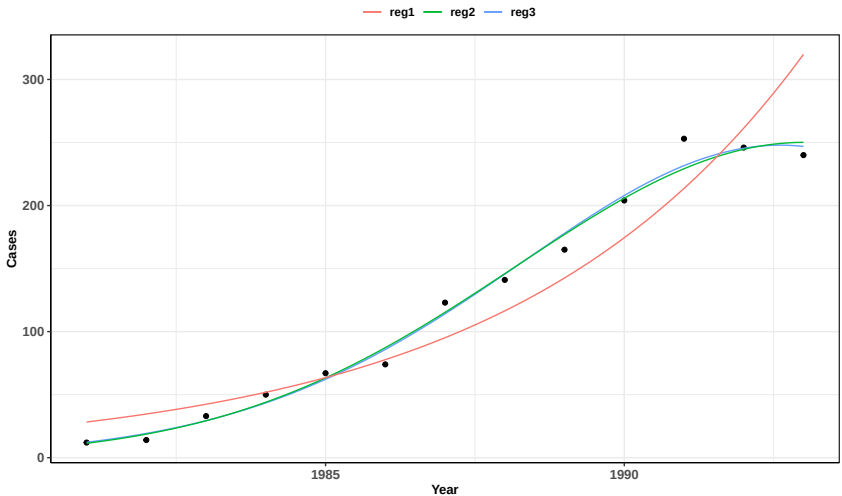
```
reg3 <- glm(Cases ~ Year + I(Year^2) + I(Year^3), family = poisson)
summary(reg3)$coef
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2983442.442	6.3537e+06	0.470	0.63867
Year	-4545.280	9.5886e+03	-0.474	0.63548
I(Year^2)	2.308	4.8240e+00	0.478	0.63231
I(Year^3)	0.000	1.0000e-03	-0.483	0.62917

On constate une très légère augmentation du pseudo-R^2 qu'est ici de 99%. Mais ce modèle est à rejeter en faveur du modèle quadratique.

reg3 est un exemple d'un modèle avec du Overfitting/Multicollinearity. En effet ce modèle est surchargé en prédicteurs corrélés les uns aux autres.

Le graphique suivant montre les trois courbes de régression correspondant aux trois modèles étudiés.



Régression de Poisson simple

Régression de Poisson multiple
- Modèle avec un prédicteur catégoriel à plus de deux niveaux
- Un prédicteur catégoriel et un continu
- Modélisation des taux: régression de Poisson avec Offset

Sélection des variables et construction du modèle

Régression de Poisson pour tableaux de contingence: Les modèles log-linéaire

Appendice

Exemple

Le fichier sp.csv contient des données qui proviennent d'une étude sur la relation entre, d'une part, le nombre d'espèces végétales (`Spe`) et, d'autre part, la biomasse totale (`Bio`) et le pH, des parcelles cultivées, codé comme "low", "mid", "high".

```
sp <- read.csv("Data/sp.csv")
head(sp, 4)
```

```
    pH    Bio Spe
1 high 0.4693  30
2 high 1.7309  39
3 high 2.0898  44
4 high 3.9258  35
```

```
str(sp)
```

```
'data.frame': 100 obs. of  3 variables:
 $ pH : chr  "high" "high" "high" "high" ...
 $ Bio: num  0.469 1.731 2.09 3.926 4.367 ...
 $ Spe: int  30 39 44 35 25 29 23 18 19 12 ...
```

```
sp <- transform(sp, pH = factor(pH,
                levels = c("low", "mid", "high")))
summary(sp)
```

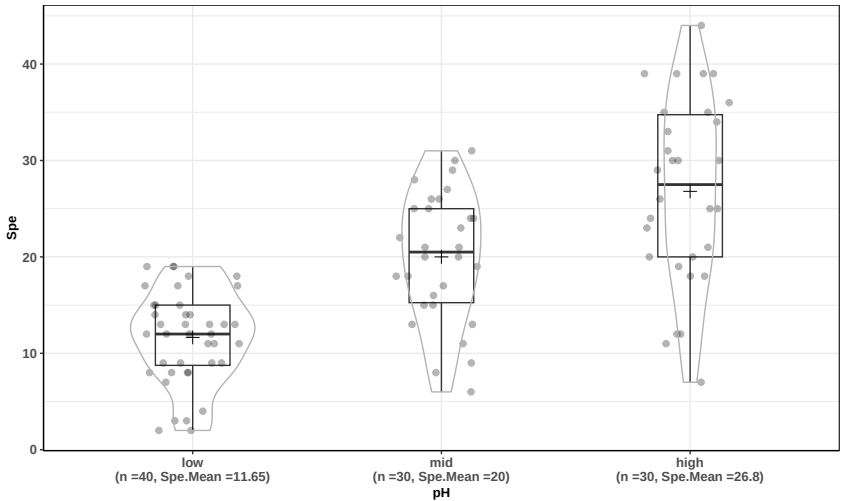
pH	Bio	Spe
low :40	Min. :0.0502	Min. : 2.0
mid :30	1st Qu.:1.4160	1st Qu.:12.0
high:30	Median :2.8622	Median :18.0
	Mean :3.3724	Mean :18.7
	3rd Qu.:4.8793	3rd Qu.:25.0
	Max. :9.9818	Max. :44.0

Régression de Poisson multiple

Modèle avec un prédicteur catégoriel à plus de deux niveaux

Commençons par étudier comment le `pH` affecte le nombre d'espèces.

```
ggplot(sp, aes(x = pH, y = Spe)) + geom_boxplot(width = 0.3, varwidth = TRUE, outlier.shape = NA, fill = NA) +
  geom_jitter(alpha = 0.3, size = 2, width = 0.2, height = 0) + stat_summary(geom = "point", fun = "mean", size = 3, shape = 3) +
  geom_violin(fill = NA, width = 0.5, color = "gray70") + scale_x_discrete(labels = paste0(levels(sp$pH), "\n(n =", table(sp$pH),
  ",", " Spe.Mean =", aggregate(Spe ~ pH, data = sp, FUN = mean)[, 2], ")"))
```



Soit $\mu(\text{pH}) = \text{E}(\text{Spe}|\text{pH})$. Nous supposons que $\text{Spe}|\text{pH} \sim \text{Pois}(\mu(\text{pH}))$. Le `pH` étant une variable nominale, on ne peut pas écrire

$$\log(\mu(\text{pH})) = \beta_0 + \beta_1 \text{pH}$$

Afin d'intégrer `pH` dans l'équation du modèle, il faut impérativement l'encoder de manière adéquate. Il y a plusieurs façons de procéder, ici nous allons utiliser le codage 0/1 (Dummy coding, en anglais).

pH	pHlow I(pH = "low")	pHmid I(pH = "mid")	pHhigh I(pH = "high")
"low"	1	0	0
"mid"	0	1	0
"high"	0	0	1

Une de ces trois variables ainsi définies est redondante: il suffit de connaitre, par exemple, les valeurs de pHmid et pHhigh pour deviner celle de pHlow puisque $\text{pHlow} = 1 - (\text{pHmid} + \text{pHhigh})$. On peut donc mettre de côté cette dernière variable et écrire (symboliquement) $\text{pH} \equiv (\text{pHmid}, \text{pHhigh})$.

Dans une telle configuration pH = "low" est notre niveau de référence. Càd le niveau auquel les autres niveaux seront comparés (voir ci-après pour plus de précisions).

L'équation du modèle s'écrit comme $\log(\mu(pH)) = \beta_0 + \beta_1 pHmid + \beta_2 pHhigh$, où

$$\log(\mu(pH)) = \beta_0 + \beta_1 I(pH = \text{"mid"}) + \beta_2 I(pH = \text{"high"}).$$

Ce qui revient à écrire

$$\log(\mu(pH)) = \begin{cases} \beta_0 & \text{si } pH = \text{"low"} \\ \beta_0 + \beta_1 & \text{si } pH = \text{"mid"} \\ \beta_0 + \beta_2 & \text{si } pH = \text{"high"} \end{cases}$$

Cette équation implique que $\exp(\beta_0) = E(Spe|pH = \text{"low"})$,

$$\exp(\beta_1) = \frac{E(Spe|pH = \text{"mid"})}{E(Spe|pH = \text{"low"})}, \text{ et } \exp(\beta_2) = \frac{E(Spe|pH = \text{"high"})}{E(Spe|pH = \text{"low"})}$$

De façon générale, pour un prédicteur catégoriel X à G niveaux “1”, “2”, …, “G”, en considérant le niveau "1" comme référence, nous avons recours à $G-1$ variables indicatrices: $X2 = I(X = "2")$, $X3 = I(X = "3")$, …, $XG = I(X = "G")$. Le modèle de Poisson s'écrit comme

$$\log(\mu(X)) = \beta_0 + \beta_1 I(X = "2") + \ldots + \beta_{G-1} I(X = "G").$$

Pour ajuster le modèle de Poisson $Spe \sim pH$ dans R, on peut procéder comme suit

```
sp <- transform(sp, pHmid = (pH == "mid") * 1, pHhigh = (pH == "high") * 1)
glm(Spe ~ pHmid + pHhigh, family = poisson, data = sp)
```

Ou, plus simplement, écrire

```
mph <- glm(Spe ~ pH, family = poisson, data = sp)
```

Mais pour qu'un tel code fonctionne et donne le résultat escompté, il est conseillé de *déclarer, dans R, tout prédicteur catégoriel comme* `factor`.

Pour rappel, par défaut R choisit toujours le premier niveau d'un factor comme référence. Pour modifier ce choix, il suffit d'utiliser la fonction `factor()` et son argument `levels` ou, plus simplement, d'utiliser la fonction `relevel()`. Voici un exemple

```
# imposer l'ordre 'high', 'mid', 'low' pour les niveaux de pH
sp$pH <- factor(sp$pH, levels = c("high", "mid", "low"))

# déclarer 'high' comme niveau de référence pour pH (càd placer 'haut' en
# première position et décaler les autres niveaux en conséquence)
sp$pH <- relevel(sp$pH, "high")
```

Voici un extrait du `summary(mph)`

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.455	0.046	53.003	0.0000e+00
pHmid	0.540	0.062	8.752	2.0885e-18
pHhigh	0.833	0.058	14.309	1.9170e-46

Il est facile de vérifier que $\hat{\beta}_0 = \log\left(\widehat{\text{moyenne}}(Spe|pH = \text{"low"})\right)$. En effet, la moyenne échantillonnale de `Spe` pour les individus ayant un `pH = "low"` est de 11.65 et
`log(11.65) = 2.45531` $= \hat{\beta}_0$.

De même, on peut vérifier que

$$\hat{\beta}_1 = \log\left(\widehat{\text{moyenne}}(Spe|pH = \text{"mid"})\right) - \hat{\beta}_0,$$
$$\hat{\beta}_2 = \log\left(\widehat{\text{moyenne}}(Spe|pH = \text{"high"})\right) - \hat{\beta}_0.$$

Il est aussi possible de réaliser des prédictions à l'aide du modèle estimé. Pour un individu i ayant un $pH = pH_i$

$$\hat{E}(Spe_i|pH = pH_i) = \exp\left(\hat{\beta}_0 + \hat{\beta}_1 I(pH_i = \text{"mid"}) + \hat{\beta}_2 I(pH_i = \text{"high"})\right)$$

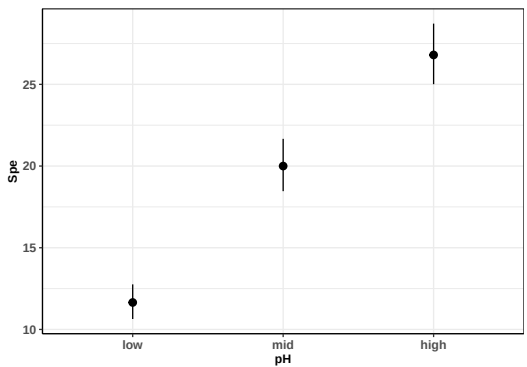
Ce qui revient simplement à calculer la moyenne empirique de Spe dans chaque niveau de pH.

Les duex sorties ci-après nous montrent ces prédictions accompagnées des intervalles de confiance correspondants.

```
predictions(mph, newdata = datagrid(
           pH = c("low", "mid", "high")))
```

```
   pH Estimate Pr(>|z|)   S 2.5 % 97.5 %
 low      11.7   <0.001 Inf  10.6   12.8
 mid      20.0   <0.001 Inf  18.5   21.7
 high     26.8   <0.001 Inf  25.0   28.7
```

```
plot_predictions(mph, condition = "pH")
```



De façon générale, un (G)LM avec que des prédicteurs catégoriels, comme notre modèle `mph`, n'a pas vraiment un intérêt direct, ni en termes d'explication ni en termes de prédiction.

En effet, les coefficients estimés et les prédictions n'apportent aucune information supplémentaire qu'on ne pouvait obtenir directement à partir des données.

L'intérêt majeur se situe ici au niveau de l'inférence qui en découle et les multitudes de généralisations et extensions possibles dont une partie sera traitée par la suite.

Concernant l'inférence, la p−valeur qui figure dans la deuxième ligne de `summary(mph)` correspond au test

$$H_0 : E(Spe|pH = \text{"mid"}) = E(Spe|pH = \text{"low"}) \text{ vs } H_1 : \text{ Non } H_0$$

Et celle de la troisième ligne correspond au test

$$H_0 : E(Spe|pH = \text{"high"}) = E(Spe|pH = \text{"low"}) \text{ vs } H_1 : \text{ Non } H_0$$

Sur base de ces valeurs on peut conclure, au seuil 5%, à une différence significative dans le nombre moyen d'espèces végétales entre les milieux à pH "mid" et à pH "low", d'une part, et entre les milieux à pH "high" et à pH "low", d'autre part.

Par contre, cette sortie nous ne permet pas de tester l'hypothèse

$$H_0 : E(Spe|pH = \text{"low"}) = E(Spe|pH = \text{"mid"}) = E(Spe|pH = \text{"high"}),$$

tout en contrôlant l'erreur de type 1.

Tester H_0 ci-dessus revient à réaliser le test suivant

$$H_0 : \beta_1 = \beta_2 = 0 \text{ vs } H_1 : \beta_1 \neq 0 \text{ ou } \beta_2 \neq 0$$

Ce qui peut être réalisé en utilisant l'un des tests classiques (Wald, LR, ou Score).

Voici, à titre d'exemple, le test LR.

```
mph0 <- glm(formula = Spe ~ 1, family = poisson, data = sp)
anova(mph0, mph, test = "LRT")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
99	488.07			
97	269.77	2	218.3	3.9587e-48

Exercice Utiliser les données sp pour tester

$$H_0 : E(Spe|pH = \text{"mid"}) = E(Spe|pH = \text{"high"}).$$

Comparer les résultats d'un test de Student classique (`t.test`) avec un test basé sur une régression Poisson (`glm`). Si vous deviez choisir, que préconiseriez-vous ?

□

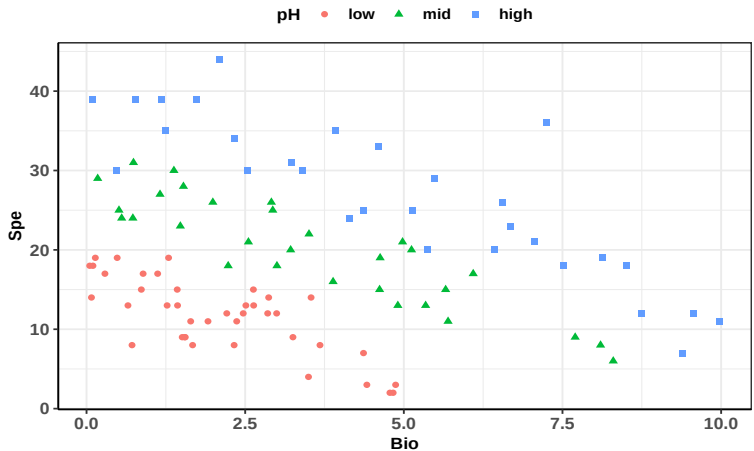
Régression de Poisson multiple

Un prédicteur catégoriel et un continu

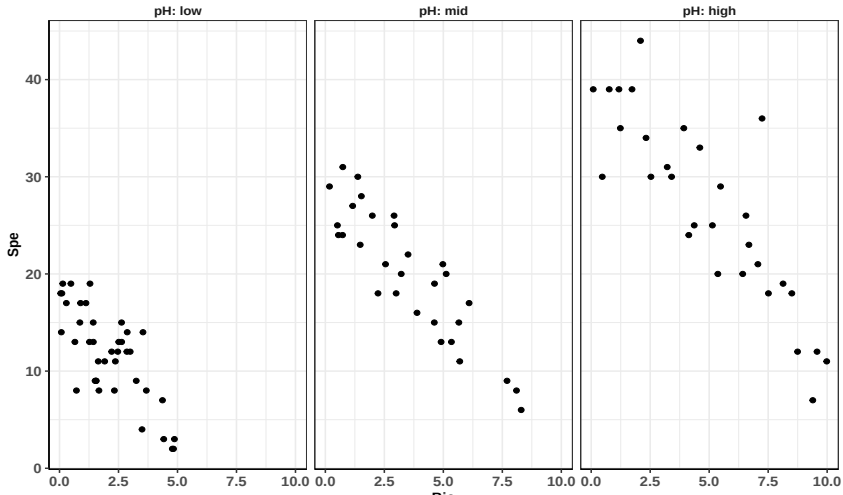
Dans notre analyse de Spe (nombre d'espèces végétales), il est naturel de vouloir aussi intégrer la biomasse (Bio) dans l'analyse.

Voici deux figures aidant à mieux apercevoir les données.

```
ggplot(sp, aes(x = Bio, y = Spe, col = pH, shape = pH)) + geom_point()
```



```
ggplot(sp, aes(x = Bio, y = Spe)) + geom_point() + facet_grid(col = vars(pH), labeller = label_both)
```



Pour réaliser une régression de Poisson sur de telles données, la démarche la plus simple consiste à séparer les individus par pH ("low", "mid", "high") et à effectuer une régression par groupe (stratified regression, en anglais). Voici les trois modèles:

```
mpHlow <- glm(Spe ~ Bio, family = poisson, data = subset(sp, pH == "low"))
mpHmid <- glm(Spe ~ Bio, family = poisson, data = subset(sp, pH == "mid"))
mpHhigh <- glm(Spe ~ Bio, family = poisson, data = subset(sp, pH == "high"))
```

	Estimate	Std. Error	z value	Pr(>|z|)
low				
(Intercept)	2.910	0.076	38.091	0.0000e+00
Bio	-0.243	0.037	-6.653	2.8729e-11
mid				
(Intercept)	3.437	0.069	50.081	0.0000e+00
Bio	-0.139	0.019	-7.165	7.7875e-13
high				
(Intercept)	3.768	0.062	61.240	0.0000e+00
Bio	-0.107	0.012	-8.577	9.7195e-18

Voici quelques remarques concernant cette démarche.

- Pour plus de précision, il serait plus intéressant d'utiliser l'ensemble des données pour effectuer l'analyse.
- La démarche appliquée ci-dessus suppose que chaque sous-population (groupe) dispose de son propre modèle (constante, pente).
- Comment peut-on comparer, par exemple, ces pentes et faire de l'inférence à ce sujet ?
- Comment peut-on, par exemple, ajuster des modèles qui partagent la même pente ou la même constante ?

La façon la plus simple de répondre à ces critiques/questions est d'intégrer ces trois modèles dans une seule et même équation dont les paramètres peuvent différer (ou pas) selon les groupes/situations.

Soit $\mu(\text{Bio}, \text{pH}) = E(Spe|\text{Bio}, \text{pH})$. On propose le modèle suivant:

$$\log(\mu(\text{Bio}, \text{pH})) = \beta_0 + \beta_1 \text{Bio} + \beta_2 I(\text{pH} = \text{"mid"}) + \beta_3 I(\text{pH} = \text{"high"}) + \beta_4 \text{Bio} \times I(\text{pH} = \text{"mid"}) + \beta_5 \text{Bio} \times I(\text{pH} = \text{"high"})$$

càd,

$$\mu(\text{Bio}, \text{"low"}) = \exp(\beta_0 + \beta_1 \text{Bio}) \quad (\text{référence})$$
$$\mu(\text{Bio}, \text{"mid"}) = \exp\left((\beta_0 + \beta_2) + (\beta_1 + \beta_4)\text{Bio}\right)$$
$$\mu(\text{Bio}, \text{"high"}) = \exp\left((\beta_0 + \beta_3) + (\beta_1 + \beta_5)\text{Bio}\right)$$

Dans la première équation, $\exp(\beta_0 + \beta_1 \text{Bio})$ est la courbe de régression de notre groupe de référence $\{\text{pH} = \text{"low"}\}$.

Dans la 2e équation, les paramètres supplémentaires β_2 et β_4 décrivent l'écart du groupe $\{\text{pH} = \text{"mid"}\}$ par rapport à la référence.

Tandis que, dans la 3e équation, β_3 et β_5 décrivent l'écart du groupe $\{\text{pH} = \text{"high"}\}$, toujours par rapport à la référence.

β_1 et (β_2, β_3) sont les "effets de base" de Bio et pH, respectivement.

Le couple (β_4, β_5) représente l'effet d'interaction entre Bio et pH, il joue un rôle très important dans cette modélisation. Il décrit comment les deux variable explicatives (Biomasse et pH) interagissent dans leurs actions sur la réponse (le nombre d'espèces végétales).

Pour voir ce que cela signifie, il faut constater que:

» l'effet du pH sur le nombre d'espèces peut être décrit par

$$\mu(\text{Bio}, \text{"mid"})/\mu(\text{Bio}, \text{"low"}) = \exp(\beta_2 + \beta_4 \text{Bio})$$
$$\mu(\text{Bio}, \text{"high"})/\mu(\text{Bio}, \text{"low"}) = \exp(\beta_3 + \beta_5 \text{Bio})$$

» l'effet de la biomasse (Bio) sur le nombre d'espèces peut être décrit par

$$\mu(\text{bio} + 1, \text{pH})/\mu(\text{bio}, \text{pH}) = \exp(\beta_1 + \beta_4 \text{I}(\text{pH} = \text{"mid"}) + \beta_5 \text{I}(\text{pH} = \text{"high"}))$$

En cas d'absence d'interaction ($\beta_4 = \beta_5 = 0$), l'effet de la biomasse sur le nombre d'espèces ne dépend pas du pH puisque, dans ce cas, $\mu(\text{bio}+1, \text{pH})/\mu(\text{bio}, \text{pH}) = \exp(\beta_1)$. De même, l'effet du pH sur le nombre d'espèces ne dépend pas de la biomasse ($\mu(\text{Bio}, \text{"mid"})/\mu(\text{Bio}, \text{"low"}) = \exp(\beta_2)$ et $\mu(\text{Bio}, \text{"high"})/\mu(\text{Bio}, \text{"low"}) = \exp(\beta_3)$). Aussi,

- les trois droites $\log(\mu(\text{Bio}, \text{"low"}))$, $\log(\mu(\text{Bio}, \text{"mid"}))$ et $\log(\mu(\text{Bio}, \text{"high"}))$ seront parallèles (pentes identiques)
- les courbes $\mu(\text{Bio}, \text{"low"})$, $\mu(\text{Bio}, \text{"mid"})$ et $\mu(\text{Bio}, \text{"high"})$ auront une forme identique, mais seront décalées (horizontalement) les unes par rapport aux autres.

Dans le modèle avec interaction ($\beta_4 \neq 0$ ou $\beta_5 \neq 0$), au moins deux des trois droites se croisent et l'interprétation des paramètres sera plus difficile car on ne pourra plus parler de l'effet de la biomasse sur la réponse, sans fixer le niveau de pH et vice-versa.

Le modèle avec interaction entre `Bio` et `pH` s'écrit dans R comme

```
mpHBio <- glm(Spe ~ Bio * pH, family = poisson, data = sp)
summary(mpHBio)$coef
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.910	0.076	38.091	0.0000e+00
Bio	-0.243	0.037	-6.653	2.8729e-11
pHmid	0.526	0.103	5.124	2.9989e-07
pHhigh	0.858	0.098	8.742	2.2853e-18
Bio:pHmid	0.104	0.041	2.518	1.1788e-02
Bio:pHhigh	0.136	0.039	3.523	4.2617e-04

REMARQUE Dans le code ci-dessus, `Spe ~ Bio*pH` peut être remplacé par `Spe ~ Bio + pH + Bio:pH` ou encore par `Bio + pH + I(Bio * pHmid) + I(Bio * pHhigh)` . □

De la sortie ci-dessus, nous pouvons écrire

$$\log(\hat{\mu}(\text{Bio}, \text{pH})) = 2.910 - 0.243\text{Bio} + 0.526\text{I}(\text{pH} = \text{"mid"}) + 0.858\text{I}(\text{pH} = \text{"high"}) + 0.104\text{Bio} \times \text{I}(\text{pH} = \text{"mid"}) + 0.136\text{Bio} \times \text{I}(\text{pH} = \text{"high"})$$

$$\Rightarrow \log(\hat{\mu}(\text{Bio}, \text{pH})) = \begin{cases} 2.910 - 0.243\text{Bio} & \text{si pH} = \text{"low"} \\ 3.436 - 0.139\text{Bio} & \text{si pH} = \text{"mid"}, \\ 3.768 - 0.107\text{Bio} & \text{si pH} = \text{"high"} \end{cases}$$

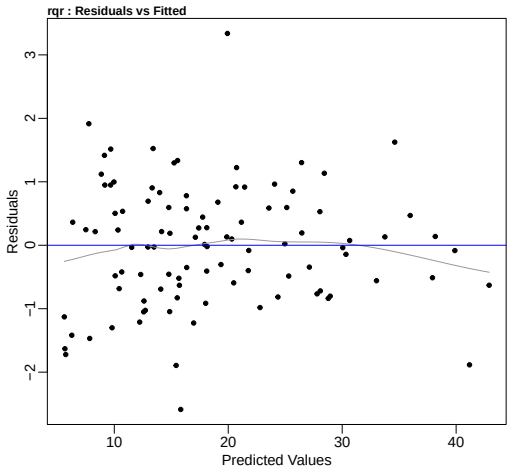
Notez que ces chiffres coïncident avec ceux obtenus avec la régression stratifiée; voir Slide 33. En considérant les trois niveaux de pH séparément, nous pouvons facilement interpréter les différents coefficients estimés. Par exemple, nous pouvons dire que

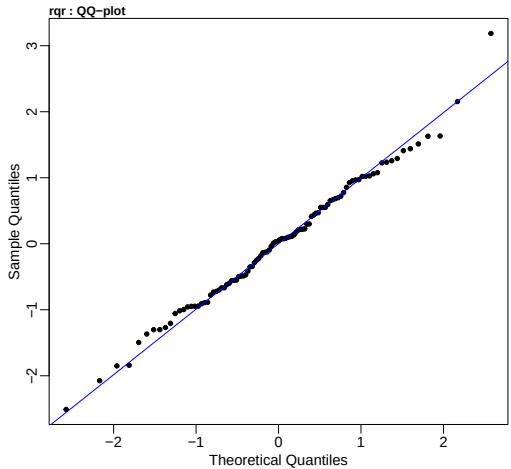
» À pH bas, le nombre d'espèces moyen est de `18.4` (`exp(2.910)`) pour une biomasse de 0 et diminue de `22%` (`exp(-0.243)-1`) par unité de biomasse.

Avec un pseudo$-R^2 =$ `81.3 %`, ce modèle semble bien expliquer les données.

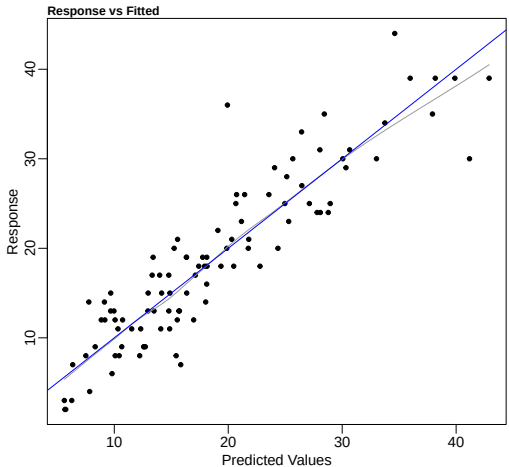
Voici quelques graphiques pour visualiser le modèle et apprécier la qualité de son ajustement aux données.

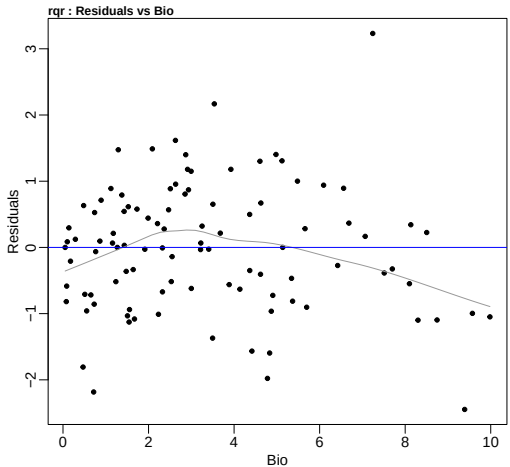
```
diagnost(mpHBio, plots = c("fitted", "qqplot"))
```



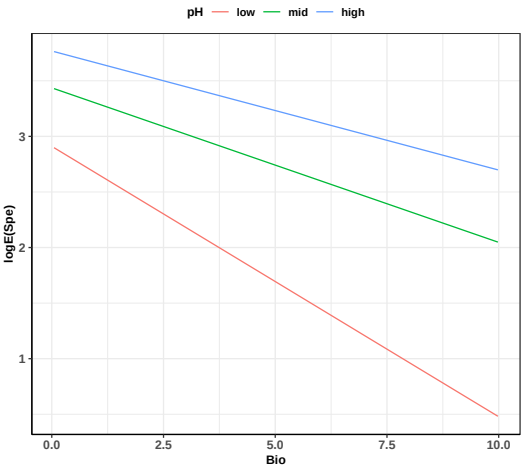


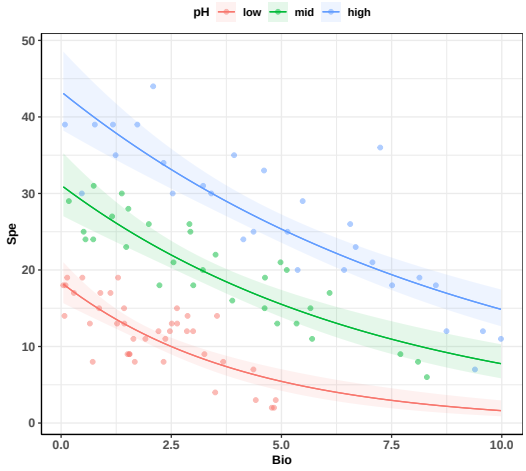
```
diagnost(mpHBio, plots = c("response", "Bio"))
```





```
plot_predictions(mpHBio, condition = list("Bio","pH"), type = "link", vcov = FALSE)
plot_predictions(mpHBio, condition = list("Bio","pH"), points = 0.5)
```





Remarque

Encore une fois, il n'est pas nécessaire d'utiliser `marginaleffects` pour créer des graphiques de prédiction. Le code suivant montre une façon de construire "manuellement" les deux derniers graphiques.

```
predL <- predict(mpHBio, se.fit = TRUE)

ggplot(sp, aes(x = Bio, y = predL$fit, color = pH)) +
  geom_line() + labs(y = "logE(Spe)")

ggplot(sp, aes(x = Bio, y = Spe, color = pH)) + geom_point(alpha = 0.5) +
  geom_line(aes(y = exp(predL$fit))) + geom_ribbon(aes(
    ymin = exp(predL$fit - 1.96 * predL$se.fit),
    ymax = exp(predL$fit + 1.96 * predL$se.fit), fill = pH, color = NULL), alpha = 0.1)
```

Globalement, nous constatons que (i) le nombre d'espèces diminue avec la biomasse; (ii) cette diminution devient plus marquée à mesure que le pH diminue; (iii) le pH a un effet positif, sur la réponse (nombre d'espèces); (iv) cet effet semble s'accentuer avec la biomasse.

Cela semble indiquer un effet d'interaction entre la biomasse et le pH. La question à se poser est de savoir si cette interaction est significative ou pas ? Pour répondre à cette question, il faut tester la nullité simultanée de β_4 et β_5, càd tester

$$H_0 : \beta_4 = \beta_5 = 0 \text{ vs } H_1 : \beta_4 \neq 0 \text{ ou } \beta_5 \neq 0$$

```
mpHBio0 <- glm(Spe ~ Bio + pH, data = sp, family = poisson)
anova(mpHBio0, mpHBio, test = "LRT")
```

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
96	104.820			
94	91.457	2	13.363	0.0012538

→ au seuil de 5%, l'interaction est significative
→ le modèle est non réductible.

Quelques remarques concernant les interactions

- Les interactions (produits) peuvent s'effectuer entre deux variables explicatives (continue-continue, continue-catégorielle, catégorielle-catégorielle) définissant ainsi des interactions dites d'ordre 2.
- La présence d'une interaction, disons entre X_1 et X_2, change l'interprétation des "effets de base" associés à ces deux variables. En effet, contrairement au cas sans interaction, il n'y a pas de sens de parler de l'effet de l'un des deux sur la réponse sans prendre en compte la valeur de l'autre.
- Il est possible de définir une interaction entre trois (ou même plus) variables, on parle alors d'interaction d'ordre 3 (three-way interaction, en anglais). Cela peut améliorer la qualité d'un modèle, mais risque de rendre sont interprétation beaucoup plus difficile.

- Si une interaction figure dans un modèle, alors ce dernier doit aussi inclure tous les termes constitutifs de cette interaction. Ainsi, un modèle avec $X_1 * X_2$ doit aussi inclure X_1 et X_2 et un modèle avec $X_1 * X_2 * X_3$ doit aussi inclure $X_1 * X_2$, $X_1 * X_3$, $X_2 * X_3$, X_1, X_2 et X_3. C'est le principe de hiérarchie des effets (hierarchical principle).
- On peut aussi formuler ce principe de la façon suivante: ne jamais supprimer les termes d'ordre inférieur avant les termes d'ordre supérieur associés.
- Ce principe, même s'il est parfois contesté, conduit à des modèles plus cohérents, du point de vue mathématique/statistique, et plus faciles à interpréter.
- Ce principe s'applique aussi au cas de la régression polynomiale. Par exemple, il est déconseillé de simplifier le modèle $\beta_0 + \beta_1 X + \beta_2 X^2$ en faveur de $\beta_0 + \beta_2 X^2$ même si β_1 s'avère être non significatif.

Modélisation des taux: régression de Poisson avec Offset

Le modèle de Poisson que nous avons considéré jusqu'à présent stipule que

$$N_i|\mathbf{X}_i \sim \text{Pois}(\mu_i), \text{ avec } \log(\mu_i) = \boldsymbol{\beta}^t\mathbf{X}_i,$$

où N est le nombre d'événements qui surviennent dans un intervalle de temps ou d'espace donné (*qui est censé rester fixe pendant l'échantillonnage*) pour une population cible bien déterminée.

Mais cela s'avère irréaliste, dans de nombreuses situations:

- On compte le nombre de nouveaux cas quotidiens d'un virus, sachant que le nombre total d'individus testés varie d'un jour à l'autre.
- nombre de plantes qui poussent dans des parcelles de différentes tailles.
- nombre d'heures d'absence par an pour des travailleurs ayant des régimes/temps de travail différents.
- le nombre annuel d'incidents enregistrés dans le secteur aéronautique, sachant que le millage varie généralement d'une année à l'autre.

Dans de pareils cas, il semble plus judicieux de modéliser le taux d'occurrence et non pas le nombre d'occurrences. Par taux, nous entendons *le nombre d'événements observés divisé par une quantité (appelons-la e) qui permet de "normaliser/standardiser"* les observations.

Ces taux sont généralement de la forme $R = N/e$, où N = nombre d'événements et e = taille de la population cible, ou taille/surface des unités d'échantillonnage, ou durée d'exposition au phénomène étudié.

Pour aller à l'essentiel, disons que nous disposons d'un échantillon i.i.d. sous la forme $(N_i, e_i, \mathbf{X}_i)$, où $N_i|\mathbf{X}_i \sim \text{Poissson}(\mu_i)$, et où notre intérêt porte sur la variable $R_i = N_i/e_i$ que nous modélisons à l'aide de l'équation $\log(E(R_i|\mathbf{X}_i)) = \boldsymbol{\beta}^t\mathbf{X}_i$, càd $\log(\mu_i/e_i) = \boldsymbol{\beta}^t\mathbf{X}_i$. De façon équivalente, nous pouvons dire que

$$(1) \quad N_i|\mathbf{X}_i \sim \text{Poissson}(\mu_i),$$
$$(2) \quad \log(\mu_i) = \boldsymbol{\beta}^t\mathbf{X}_i + \log(e_i).$$

Dans cette dernière équation le terme $\log(e_i)$ est appelé, en anglais, offset (décalage).

Il s'agit en fait d'une régression de Poisson classique mais dont la moyenne a été décalée de $\log(e_i)$. Ce dernier peut être considéré comme une variable explicative dont le coefficient est connu et égal à 1.

L'estimation et l'interprétation des paramètres restent très similaires à celles d'un modèle de Poisson classique (sans offset), sauf que l'on parlera du taux moyen et non pas du nombre moyen d'occurrences.

Exemple

Nous allons re-analyser les données de l'effet du pH ("low", "mid", "high") sur le nombre d'espèces sauf que nous allons ici considérer les données sous une forme groupée (data `dt`):

pH	TotObs	Spe
low	40	466
mid	30	600
high	30	804

`TotObs` est le nombre d'observations récoltées pour chaque niveau de pH et `Spe` est ici le nombre *total* d'espèces végétales par niveau de pH.

```
glm(Spe ~ pH + offset(log(TotObs)), family = poisson, data = dt) |> summary() |> coef()
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.455	0.046	53.003	0.0000e+00
pHmid	0.540	0.062	8.752	2.0889e-18
pHhigh	0.833	0.058	14.309	1.9183e-46

Remarquez que nous obtenons exactement les mêmes résultats qu'auparavant (voir Slide 25).

Régression de Poisson simple

Régression de Poisson multiple

Sélection des variables et construction du modèle
- Critères de comparaison
- Recherche exhaustive (best subset selection)
- Méthodes pas à pas (stepwise procedures)
- Qualité des prédictions et validation croisée

Régression de Poisson pour tableaux de contingence: Les modèles log-linéaire

Appendice

En pratique, on est souvent confronté à de nombreux modèles possibles, correspondants aux différentes combinaisons de variables explicatives (dont le nombre peut être très grand) et leurs interactions et/ou transformations possibles.

Trouver un modèle "adéquat" n'est pas une tâche facile. C'est un problème commun à toutes les méthodes de régression. En général, la sélection des variables est guidée par les principes suivants :

» On privilégie la simplicité. À pouvoirs explicatifs (ou prédictifs) sensiblement équivalents, on choisit les modèles parcimonieux pour plusieurs raisons: le modèle est plus facile à lire/intérpréter, moins sujet aux problèmes de redondance (multicolinéarité), il coûte moins puisque le nombre de variables à collecter est plus faible, il est plus flexible/robuste lorsqu'il s'agit de l'appliquer sur de nouvelles données.

» Un modèle "trop simple", qui omet des prédicteurs essentiels, ne peut capturer la pleine complexité du phénomène et risque donc de mener à des conclusions biaisées. On parle de sous-ajustement aux données (underfitting).

» À contrario, un modèle "trop complexe" (avec trop de paramètres/variables) risque de capturer des effets purement aléatoires propres au jeu de données utilisé et non généralisables. On parle de surajustement (overfitting). En général, plus un modèle est complexe, plus sa variabilité est grande (càd des estimations moins précises). En bref, il faut trouver un compromis entre biais et variance (bias-variance tradeoff en anglais), càd trouver un équilibre entre simplicité et complexité.

Avec m prédicteurs on peut envisager $2^m = \sum_{k=0}^{m} C_m^k$ modèles possibles (sans compter d'éventuelles interactions). Ce chiffre devient vite conséquent avec m:

```
2^c(1, 2, 4, 10, 12, 14, 16, 18, 20)
```

```
[1]       2       4      16    1024    4096   16384   65536  262144 1048576
```

Il est donc souvent difficile d'envisager dans la pratique une inspection exhaustive de toutes les possibilités quand m est "trop grand" .

Dans tous les cas, avant de pouvoir sélectionner un modèle, il faut disposer

1. d'un critère/métrique permettant d'évaluer objectivement l'apport de chaque prédicteur candidat au modèle. Il s'agit fondamentalement de confronter deux modèles: celui avec et celui sans le prédicteur concerné. Ce dernier est sélectionné si le modèle avec est jugé "sensiblement meilleur" que le modèle sans.
2. d'une procédure/stratégie de sélection pratique et fiable qui permet de choisir parmi les nombreux modèles envisageables, le meilleur au sens du critère adopté, sans devoir nécessairement procéder à une analyse exhaustive de toutes les possibilités.

Qu'il s'agisse du critère de comparaison ou de la procédure de sélection, il existe un large éventail d'approches dans la littérature. Le choix de l'un ou l'autre dépend de plusieurs éléments: degré de simplicité/flexibilité recherchée, la taille de l'échantillon et le nombre de prédicteurs disponibles, l'objectif de l'analyse (descriptive, prédictive ou déductive), la nature des modèles à comparer (emboités ou non emboités), les hypothèses stipulées à propos du phénomène étudié et/ou des modèles à comparer, ...

Aussi, le choix de l'une ou l'autre méthode de sélection peut aboutir à des modèles complètement différents. C'est pourquoi on parlera toujours de "meilleur" modèle par rapport à un critère donné.

Pour comparer deux modèles, il est tentant de recourir à des critères tels que la vraisemblance, la déviance ou le pseudo$-R^2$. Mais cela n'est pas toujours une bonne idée; pourquoi ?

La section suivante décrit un certain nombre d'approches parmi les plus couramment utilisées dans la pratique.

Critères de comparaison

Tests de significativité

La méthode la plus conventionnelle consiste à ajuster un modèle et à y identifier les termes qui sont statistiquement significatifs (ou non-significatifs) au regard de leurs p−valeurs, obtenues en testant la nullité des coefficients corespodants à l'aide d'un test statistique, tel que le LRT (likelihood ratio test). Les variables/termes présentant une "grande" p−valeur sont écartés, ne laissant au final que ceux présentant une "petite" p−valeur. Cette approche sera décrite plus en détail ci-dessous.

Critères d'information: AIC et BIC

Une façon simple et très populaire pour comparer des modèles consiste à utiliser les critères d'information (Information Criteria). Contrairement aux tests statistiques classiques, ces critères peuvent être utilisés pour comparer des modèles emboités ou non emboités.

L'objectif qui sous-tend l'utilisation ces critères d'information est de parvenir à un modèle comportant le moins de paramètres possible, donc facile à interpréter et à généraliser, et qui explique de manière adéquate les variations observées dans les données. Ce qu'on peut traduire comme suit: on souhaite un modèle M dont le log-vraisemblance l^M soit aussi grand que possible et le nombre p^M de paramètres (effectifs) soit aussi petit que possible. Nous pouvons combiner ces deux quantités en un seul critère, défini par,

$$IC(M) = -2l^M + k \times p^M$$

qu'il faudra minimiser. Dans cette formule, k est un chiffre positif (à choisir par l'utilisateur) dont le rôle est de pénaliser les modèles complexes et surajustés.

On trouve différents types de pénalités, les plus connus sont:

» $k = 2 \Longrightarrow$ AIC: Akaike's information criterion.

» $k = \log(\text{nbr. obs.}) \Longrightarrow$ BIC: Bayesian information criterion.

$\rightarrow$ Le BIC aura tendance à davantage pénaliser les modèles les plus complexes (càd avec plus de paramètres); notez que $\log(8) \approx 2.1$ alors que $\log(80) \approx 4.4$.

Exemple

Voici l'AIC et le BIC du modèle `mpHBio` :

» `AIC(mpHBio)` = `-2*logLik(mpHBio) + 2*6` = `565.74`

» `BIC(mpHBio)` = `AIC(mpHBio, k = log(100))` = `-2*logLik(mpHBio) + log(100)*6` = `581.37`

Le tableau suivant fournit ces critères pour les modèles `mpHBio` et `mpHBio0` .

```
cbind(AIC(mpHBio, mpHBio0),
      BIC(mpHBio, mpHBio0))
```

	df	AIC	df	BIC
mpHBio	6	565.74	6	581.37
mpHBio0	4	575.11	4	585.53

→ L'AIC et le BIC s'accordent ici sur le fait que `mpHBio` (le modèle avec interaction) est préférable à `mpHBio0` (le modèle sans interaction). De manière générale, ces deux critères ne concordent cependant pas nécessairement.

Remarques

Pour que les valeurs AIC (ou BIC) soient comparables, les modèles doivent être basés sur la même vraisemblance et avoir la même variable réponse. Vous ne pouvez pas, par exemple, comparer deux modèles dont la réponse est Y et $\log(Y)$, respectivement, ou deux modèles dont la densité est Normal pour l'un et Poisson pour l'autre.

Attention aux valeurs manquantes dans R. Lors de l'ajustement d'un modèle, R exclut automatiquement toute observation présentant une valeur manquante pour l'une des variables incluses dans le modèle. Avant d'estimer une série de modèles, il est conseillé d'éliminer toutes les observations qui ont au moins une valeur manquante pour l'une des variables d'intérêt.

L'AIC (ou le BIC) compare les modèles entre eux, mais ne nous dit pas si l'un ou l'autre modèle convient réellement aux données. En fait, le modèle ayant le plus petit AIC/BIC peut être totalement inapproprié.

Recherche exhaustive (best subset selection)

Il s'agit d'une approche naturelle, mais coûteuse en termes de calculs, qui consiste à construire tous les modèles possibles et sélectionner celui qui optimise le critère choisi (par exampel l'AIC).

Pour effectuer les calculs, nous allons utiliser la fonction `glmulti::glmulti()`. Attention, si m, le nombre de prédicteurs, est trop grand, alors il pourrait falloir des jours pour obtenir le résultat !

Exemple

Nous allons illuster cette démarche sur le jeu de données `gala` qui se trouve dans le package `faraway`. Dans ce jeu de données, on mesure 7 variables quantitatives (voir `?gala` pour une description détaillée) pour 30 îles Galapagos.

On cherche ici à construire un modèle pour expliquer la variable `Species` (nombre d'espèces végétales) à l'aide des variables géographiques à disposition (`Area`, `Elevation`, ...).

Commençons par effectuer quelques manipulations sur les données et réalisons une (mini) analyse descriptive à l'aide, entre autres, de la fonction `GGally::ggpairs()`.

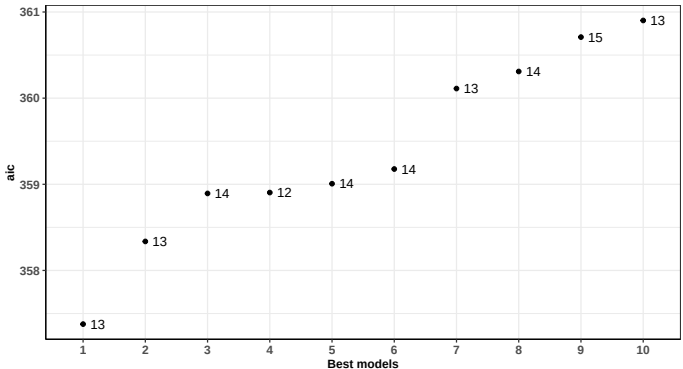
```
# run the code on the console to see the results
data(gala, package = 'faraway')
gala <- transform(gala, ElevationC = cut(Elevation, c(-Inf, quantile(Elevation,
                probs = c(0.33, 0.66)), Inf), labels = c("low", "mid", "high")), Endemics = NULL)
str(gala); summary(gala)
boxplot(gala); GGally::ggpairs(gala)
```

Appelons `glmulti()` pour effectuer la sélection. Le premier argument de `glmulti()` spécifie la variable réponse et les termes (effets principaux et/ou interactions) à utiliser dans les modèles candidats.

```
res <- glmulti(
  Species ~ (log(Area) + log(Nearest) + log(Scruz + 1) + log(Adjacent) +  log(Elevation))^2,
  family = poisson, data = gala,
  crit = aic,             # aic, bic, aicc, qaic, or qaicc
  marginality = TRUE,     # exclude models with interactions but not the corresponding main effets
  confsetsize = 10,       # number of top models to output
  plotty = FALSE, report = FALSE) # turn off any runtime outputs
```

```r
# summary of the results
summary(res)
weightable(res)
```

```r
plot(res) #or#
ggplot() + aes(factor(1:10), res@crits, label = res@K) + geom_point() + geom_text(hjust = -0.5) + labs(x = "Best models", y = "aic")
```



Dans ce graphique, les chiffres apparaissant à côté des points indiquent le nombre de termes dans chaque modèle.

```r
# The best model
mdBsb <- res@objects[[1]]
mdBsb |> summary() |> coef()
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.656	1.235	2.151	3.1506e-02
log(Area)	-0.054	0.077	-0.696	4.8625e-01
log(Nearest)	-0.918	0.219	-4.184	2.8690e-05
log(Scruz + 1)	0.407	0.203	2.003	4.5220e-02
log(Adjacent)	0.571	0.061	9.321	1.1579e-20
log(Elevation)	0.025	0.232	0.109	9.1351e-01
log(Area):log(Nearest)	-0.053	0.019	-2.732	6.2989e-03
log(Nearest):log(Scruz + 1)	-0.116	0.018	-6.487	8.7500e-11
log(Nearest):log(Adjacent)	0.023	0.007	3.316	9.1198e-04
log(Area):log(Elevation)	0.082	0.012	6.757	1.4047e-11
log(Nearest):log(Elevation)	0.252	0.047	5.403	6.5639e-08
log(Scruz + 1):log(Elevation)	-0.061	0.032	-1.868	6.1731e-02
log(Adjacent):log(Elevation)	-0.108	0.010	-10.852	1.9604e-27

Notez que la fonction `glmulti()` peut être utilisée pour sélectionner un modèle avec un nombre prédéterminé de termes. Pour ce faire, il suffit d'utiliser les arguments `minsize` et `maxsize`.

```
glmulti(Species ~ (log(Area) + log(Nearest) + log(Scruz + 1) + log(Adjacent) + log(Elevation))^2,
  data = gala, crit = aic, family = poisson, marginality = TRUE, confsetsize = 5,
  plotty = FALSE, report = FALSE, minsize = 3, maxsize = 3) |> weightable()
```

model	aic	weights
Species ~ 1 + log(Area) + log(Adjacent) + log(Adjacent):log(Area)	506.74	1.0000e+00
Species ~ 1 + log(Area) + log(Nearest) + log(Adjacent)	536.56	3.3403e-07
Species ~ 1 + log(Area) + log(Scruz + 1) + log(Adjacent)	536.62	3.2430e-07
Species ~ 1 + log(Area) + log(Adjacent) + log(Elevation)	563.27	5.3135e-13
Species ~ 1 + log(Area) + log(Scruz + 1) + log(Scruz + 1):log(Area)	682.47	6.9364e-39

Méthodes pas à pas (stepwise procedures)

C'est la procédure de sélection traditionnelle qui, malgré certaines critiques, est encore largement utilisée dans la pratique en raison de sa simplicité et de sa polyvalence.

- On commence par ajuster un modèle "complet" (avec autant de variables souhaitées) et on recherche ensuite le terme/variable le moins significatif, càd celui dont l'élimination n'impacte pas de façon sensible la qualité du modèle (telle que mesurée par le critère/métrique adopté).
- On supprime alors ce terme et on obtient ainsi un nouveau modèle "simplifié". La procédure est ensuite répétée jusqu'à ce que plus aucun terme ne puisse être supprimé du modèle (sans nuire sensiblement à la qualité de l'ajustement).

Cette démarche est connue dans la littérature comme "Backward elimination".

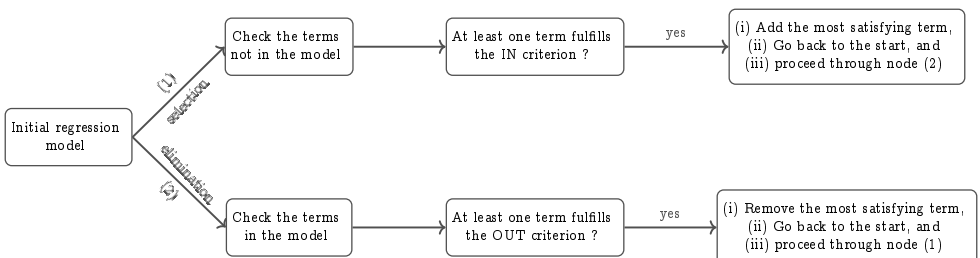
- Avec le LRT comme critère, on choisit un seuil α relativement grand, par exemple 15%, et on élimine, à chaque étape, la variable ayant la plus grande p—valeur supérieure à 15%.
- Avec l'AIC, à chaque étape, on élimine du modèle la variable qui entraîne la plus forte diminution de l'AIC.

Une stratégie similaire consiste à démarrer d'un modèle simple (avec, par exemple, seulement la constante) et à ajouter à chaque itération le terme/variable le plus significatif, parmi un ensemble prédéfini, qui améliore sensiblement la qualité de l'ajustement. On continue ainsi à ajouter des termes jusqu'à ce qu'aucune amélioration tangible ne soit détectée. Cette démarche est dite de "Forward Selection".

- Avec le LRT, comme critère de comparaison, nous pouvons, à chaque étape, incorporer la variable ayant la plus petite p—valeur inférieure à 15%.
- Avec l'AIC, à chaque étape, on ajoute au modèle la variable qui entraîne la plus forte diminution de l'AIC.

Dans tous les cas, il est fortement conseillé de ne supprimer qu'une seule variable à chaque étape et de respecter le principe de hiérarchie en éliminant, dans le cas du "backward elimination", d'abord les interactions (termes d'ordres supérieurs) avant de s'attaquer aux effets de bases (termes d'ordres inférieurs) associés. Au contraire, dans le cas du "forward selection", on ajoute les effets de base avant de s'occuper des interactions.

Le principal inconvénient de ces approches (forward ou backward) est que, une fois introduite ou supprimée, une variable n'est plus jamais questionnée même si, par exemple, elle devient redondante à cause de l'introduction, par la suite, d'autres variables qui lui sont corrélées. L'approche dite mixte (Mixed Selection, en anglais) permet de remédier à ce problème en combinant les deux précédentes méthodes (forward et backward).



La procédure est stoppée dès qu'aucun terme ne peut être ajouté ou supprimé (càd lorsque, par exemple, aucune diminution de l'AIC n'est possible).

Exemple (suite)

Expérience (1): "backward elimination" avec LRT

```
# initial model
mdaLR <- glm(Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Adjacent) +
  log(Scruz + 1) + ElevationC, family = poisson, data = gala)

# summary and dioagnostics
summary(mdaLR)
diagnost(mdaLR)
```

Dans R, la selction peut être réalisée "manuellement" en utilisant à répétition les fonctions `glm()` et `anova()`. Mais il est plus simple de recourir aux fonctions `drop1()` et `update()`. La première teste successivement tous les termes qui peuvent être supprimés d'un modèle donné. La seconde permet de mettre facilement à jour un modèle. La meilleure façon de comprendre ces outils est de les utiliser!

```
# step 1
drop1(mdaLR, test = "LRT")

Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Adjacent) +
    log(Scruz + 1) + ElevationC
               Df Deviance AIC LRT Pr(>Chi)
<none>                  255 431
log(Area)       1      680 855 425  < 2e-16 ***
log(Elevation)  1      272 447  18  2.4e-05 ***
log(Nearest)    1      265 439  10   0.0015 **
log(Adjacent)   1      333 508  78  < 2e-16 ***
log(Scruz + 1)  1      255 429   0   0.8630
ElevationC      2      360 533 105  < 2e-16 ***
---
```

Dans cette sortie la ligne `<none>` représente le modèle actuel (ici `mdaLR`). Les autres lignes portent chacune le nom de la variable/effet supprimée et donnent, en quelques chiffres, des statistiques pour apprécier le bien-fondé d'une telle suppression.

Ici, la plus grande p−valeur supérieure à 15% coresponds à `log(Scruz + 1)`

→ on supprime `log(Scruz + 1)` et on recommence.

```
# step 2
mdaLR2 <- update(mdaLR, . ~ . - log(Scruz + 1))
drop1(mdaLR2, test = "LRT")

Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Adjacent) +
    ElevationC
              Df Deviance AIC LRT Pr(>Chi)
<none>                255 429
log(Area)      1      683 856 429  < 2e-16 ***
log(Elevation) 1      273 446  18  1.8e-05 ***
log(Nearest)   1      271 444  17  4.5e-05 ***
log(Adjacent)  1      339 512  84  < 2e-16 ***
ElevationC     2      367 538 113  < 2e-16 ***
---
```

Toutes les p−valeurs sont inférieures à 15% → rien à suprimer → STOP.
→ `mdaLR2` est le modèle sélectionné.

REMARQUE Contrairement à la procédure exaustive, il n'y a aucune garantie que le modèle ainsi sélectionné soit le meilleur. Et, bien entendu, le modèle final retenu change en fonction du modèle initial choisi. □

Expérience (2): même que (1) mais avec un autre modèle initial

```r
# initial model
mdbLR <- glm(Species ~ (log(Area) + log(Elevation) + log(Nearest) + log(Adjacent)
             + log(Scruz + 1))^2, family = poisson, data = gala)

# initial model: summary and dioagnostics
summary(mdbLR)
diagnost(mdbLR)
```

```r
# run the code below step by step and see the results step 1
drop1(mdbLR, test = "LRT")
# step 2
mdbLR2 <- update(mdbLR, . ~ . - log(Scruz + 1):log(Adjacent))
drop1(mdbLR2, test = "LRT")
# step 3
mdbLR3 <- update(mdbLR2, . ~ . - log(Area):log(Adjacent))
drop1(mdbLR3, test = "LRT")
# step 4
mdbLR4 <- update(mdbLR3, . ~ . - log(Area):log(Scruz + 1))
drop1(mdbLR4, test = "LRT")
```

```
# step 5
mdbLR5 <- update(mdbLR4, . ~ . - log(Scruz + 1):log(Elevation))
drop1(mdbLR5, test = "LRT")

Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Adjacent) +
    log(Scruz + 1) + log(Area):log(Elevation) + log(Area):log(Nearest) +
    log(Elevation):log(Nearest) + log(Elevation):log(Adjacent) +
    log(Nearest):log(Adjacent) + log(Nearest):log(Scruz + 1)
                           Df Deviance AIC   LRT Pr(>Chi)
<none>                            174 359
log(Area):log(Elevation)    1      226 409  51.7  6.6e-13 ***
log(Area):log(Nearest)      1      201 384  26.6  2.5e-07 ***
log(Elevation):log(Nearest) 1      227 410  53.2  3.0e-13 ***
log(Elevation):log(Adjacent) 1     308 490 133.4  < 2e-16 ***
log(Nearest):log(Adjacent)  1      187 370  13.2  0.00028 ***
log(Nearest):log(Scruz + 1) 1      218 400  43.4  4.5e-11 ***
---
```

→ `mdbLR5` est le modèle sélectionné.

Quelques remarques

- Nous ne pouvons pas comparer les deux modèles `mdaLR2` et `mdbLR5` sur la base de la p−valeur du LRT (traditionnel). Pourquoi ? Une comparaison est possible en recourant à d'autres critères ....
- Au lieu du "Backward elimination", nous pouvons utiliser la procédure "Forward Selection". Dans ce cas, la fonction `add1()` peut être utile. Voici un exemple illustrant son usage.

```
add1(glm(Species ~ 1, family = poisson, data = gala), scope = ~ (log(Area) + log(Nearest) +
     log(Scruz + 1) + log(Adjacent) + log(Elevation))^2, test = "LRT")
# the scope argument specifies the predictors/terms that may be added to the model
```

- En tant qu'outil d'inférence statistique, un test entre modèles n'est fiable que si le modèle complet et ses hypothèses sous-jacentes sont corrects et que la taille de l'échantillon est suffisante. Plus on s'éloigne de ce scénario, plus la validité du test est compromise. En outre, la multiplication des tests signifie que l'erreur de type 1 (globale) n'est pas contrôlée (mais cela n'est pas la priorité ici).
- En raison de ces inconvénients et d'autres encore, les méthodes de sélection fondées sur les tests statistiques ne sont pas les plus recommandées/utilisées.

Autres Expériences: "mixed stepwise selection" avec AIC ou BIC

Ici aussi, les fonctions `drop1()`, `add1()` et `update()` peuvent être utilisées pour effectuer les calculs. Mais un moyen encore plus commode est de se servir de la fonction `step()`.

La fonction `step()` est programmée pour utiliser l'un des critères d'information afin d'effectuer la sélection des variables de manière totalement autonome. Pour ce faire, il fait appel, en interne, aux fonctions `add1()` et `drop1()`. L'exemple suivant illustre son utilisation.

```
# simplest model
simp_model <- glm(Species ~ 1, family = poisson, data = gala)
# full model
full_model <- glm(Species ~ (log(Area) + log(Nearest) + log(Scruz + 1) +
                  log(Adjacent) + log(Elevation))^2, family = poisson, data = gala)
# the range of models to be examined
scope <- list(lower = simp_model, upper = full_model)

# Backward-Forward selection: start with the full model and go backward to the simplest one
mdAicBF <- step(full_model, scope = scope,
                  direction = "both",  # direction = "backward" ---> backward elimination
                  k = 2)  # k= 2 ---> AIC; k = log(n), with n = nobs(full_model) ---> BIC
```

```r
# Forward-Backward selection: start with the simplest model and go forward to the full one
mdAicFB <- step(simp_model, scope = scope,
                direction = "both")  # direction = "forward" ---> forward selection
```

Les deux approches ci-dessus conduisent à des modèles différents:

```r
formula(mdAicBF)

Species ~ log(Area) + log(Nearest) + log(Scruz + 1) + log(Adjacent) +
    log(Elevation) + log(Area):log(Nearest) + log(Area):log(Elevation) +
    log(Nearest):log(Scruz + 1) + log(Nearest):log(Adjacent) +
    log(Nearest):log(Elevation) + log(Scruz + 1):log(Elevation) +
    log(Adjacent):log(Elevation)

formula(mdAicFB)

Species ~ log(Area) + log(Adjacent) + log(Nearest) + log(Elevation) +
    log(Area):log(Elevation) + log(Nearest):log(Elevation) +
    log(Area):log(Nearest) + log(Adjacent):log(Elevation)
```

Notez que le modèle `mdAicBF` coïncide avec le modèle `mdBsb` (best subset model). Ici aussi, l'AIC et le BIC (avec le paramètre `k` fixé à $\log(30)$) aboutissent aux mêmes modèles (la sortie n'est pas fournie).

Qualité des prédictions et validation croisée

Si nous voulons utiliser un modèle donné pour faire des prédictions, ce qui importe est sa capacité à prédire de nouvelles observations. Si nous disposons effectivement de nouvelles données $(Y_l^{New}, \mathbf{X}_l^{New}), l = 1, \ldots, L$, nous pouvons alors facilement quantifier la qualité prédictive en calculant, par exemple,

$$\text{RMSE} = \sqrt{L^{-1} \sum_{l=1}^{L} \left(\hat{Y}_l^{New} - Y_l^{New} \right)^2}, \text{ou MAE} = L^{-1} \sum_{l=1}^{L} \left| \hat{Y}_l^{New} - Y_l^{New} \right|$$

où $\hat{Y}_l^{New} = \hat{\mu}(\mathbf{X}_l^{New})$ est la valeur prédite de Y_l^{New} calculée sur la base du modèle estimé.

Cette approche est typiquement impossible dans la pratique, car nous ne disposons que d'un seul échantillon, celui utilisé pour ajuster le modèle.

Il est tentant d'utiliser ce même échantillon pour calculer le RMSE (ou le MAE), mais cela peut conduire à une forte surestimation de la qualité prédictive du modèle en question.

La méthode de validation croisée (CV) est l'approche la plus couramment utilisée en pratique pour contourner ce problème. Cette méthode consiste à répartir aléatoirement l'échantillon en deux sous-ensembles disjoints : un sous-ensemble d'estimation/apprentissage (training set) et un sous-ensemble de validation/test (test set). En général, la taille du training set est relativement grande par rapport à celle du test set.

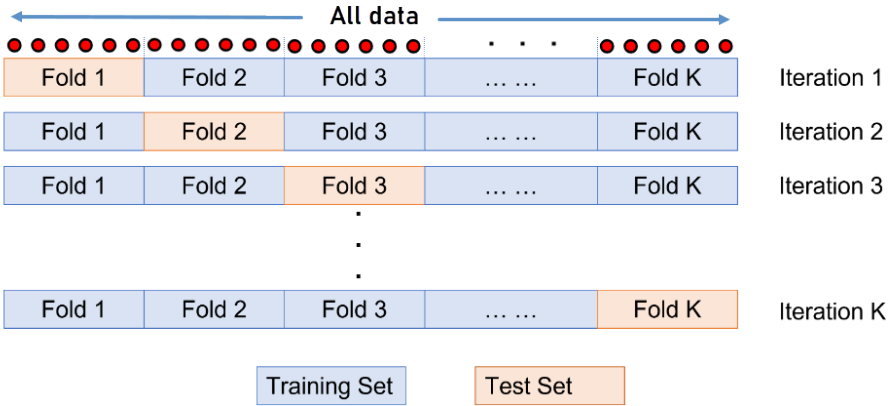
Le modèle est estimé en utilisant uniquement le training set. Les critères d'évaluation du modèle, tels que RMSE ou MAE, sont calculés en utilisant uniquement le test set. *Ce processus est généralement répété plusieurs fois* en sélectionnant au hasard les observations à placer dans le training et le test set.

Plusieurs versions de la CV existent dans la littérature, la plus connue étant le K−fold CV ou KCV (K étant un nombre entier compris entre 2 et n):

1. On dévise l'échantillon de manière alétoire en K groupes (Fold 1, Fold 2, ..., Fold K) de tailles à peu près égales.
2. On estime le modèle en utilisant toutes les observations à l'exception de celles du groupe 1 (Fold 1).
3. On utilise les observations du groupe 1 comme test set pour calculer l'erreur de prédiction ($RMSE_1$ ou MAE_1).

On répété (2) et (3), en modifiant à chaque fois le test set: test-set = Fold 2, puis test-set = Fold 3, ... Nous obtenons alors K estimations de l'erreur de prédiction: $RMSE_1, RMSE_2, \ldots, RMSE_K$. La moyenne arithmétique de ces dernières, $K^{-1} \sum_{k=1}^{K} RMSE_k$ constitue l'estimateur (final) de l'erreur de prédiction.

Le graphique suivant illustre cette approche.


All data
Fold 1
Fold 2
Fold 3
… …
Fold K
Iteration 1
Fold 1
Fold 2
Fold 3
… …
Fold K
Iteration 2
Fold 1
Fold 2
Fold 3
… …
Fold K
Iteration 3
Fold 1
Fold 2
Fold 3
… …
Fold K
Iteration K
Training Set
Test Set

Le choix de K joue un rôle important dans la KCV. Plus K est grand, plus le nombre de calculs nécessaires est important. En outre, la théorie suggère que le biais d'estimation de l'erreur de prédiction diminue en fonction de K. En revanche la variance augmente en fonction de K.

Un cas particulier de la KCV correspond au choix de $K = n$. Cette version est connue dans la littérature sous le nom de "leave-one-out CV" ou LOOCV.

L'erreur quadratique estimée par cette méthode est donnée par

$$\text{RMSE} = \sqrt{n^{-1} \sum_{i=1}^{n} \left(\hat{Y}_{-i} - Y_i \right)^2},$$

où $\hat{Y}_{-i}$ est la valeur prédite de Y_i obtenue en utilisant toutes les observations pour estimer le modèle, sauf la ième. LOOCV est souvent recommandée lorsque la taille de l'échantillon est faible et que l'objectif est d'obtenir l'estimation la moins biaisée possible de la qualité prédictive d'un (seul) modèle.

Le choix $K = 10$ est souvent recommandé comme critère de sélection lorsqu'il s'agit de choisir (de manière consistante) un modèle parmi une liste donnée.

Lorsque $K << n$, la qualité de l'estimation de l'erreur de prédiction peut davantage être améliorée en répétant la KCV un certain nombre de fois, où le fractionnement des données, en k—Flods, diffère à chaque répétition d'une exécution à l'autre. Cette méthode est connue sous le nom de "repeated k-Fold Cross-Validation" (repeated KCV). Par exemple, avec 10 répétitions d'une 4CV, on obtient 40 RMSE dont la moyenne arithmétique est l'estimation finale de l'erreur de prédiction.

Les calculs nécessaires pour produire une KCV avec (ou sans) répétition peuvent facilement être effectués à l'aide de R. Le package caret rend ces calculs encore plus simples. Voici un exemple.

```
# Select the CV version to be used; see the Help
fitControl <- trainControl(method = "repeatedcv",   # "cv", "LOOCV", ...
                           number = 5, repeats = 10,   # number of folds/repeatitions
                           savePredictions = TRUE)   # To hold-out predictions for each resample

# Perform calculations
set.seed(1)
fit <- train(Species ~ log(Area), data = gala, family = poisson, method = "glm",
             trControl = fitControl)
```

```
30 samples
 1 predictor

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 10 times)
Summary of sample sizes: 24, 25, 24, 23, 24, 25, ...
Resampling results:

  RMSE    Rsquared  MAE
  65.576  0.81687   41.231
```

Il est important de comprendre que le RMSE figurant dans la sortie ci-dessus est la moyenne des 50 RMSE calculés par l'algorithme tel que décrit ci-dessus. Dans notre cas, l'échantillon original des 30 observations est divisé de manière aléatoire en environ $6 = 30/5$ observations utilisées dans la phase de test et $24 = 30 - 6$ observations utilisées dans la phase d'apprentissage. Cette procédure est répétée 10 fois, ce qui donne les 50 RMSE.

Le code suivant peut être utile pour obtenir plus de détails sur les calculs intermédiaires effectués par la fonction `train()`.

```
fit$resample  # dataframe of performance measurements for each resample.
fit$pred  # dataframe of predictions and fold indexes for each resample.
```

Le tableau suivant résume quelques statistiques qualitatives des différents modèles déjà abordés ("CV" = 5CV avec 10 répétitions).

	df	pR2	AIC	BIC	RMSE	MAE	CV.RMSE	CV.MAE	LOOCV.RMSE	LOOCV.MAE
mdaLR2	7	92.8	429.4	439.2	24.6	17.7	40.06	28.77	41.45	27.34
mdbLR5	12	95.0	358.9	375.7	14.4	11.3	48.93	33.56	33.61	24.97
mdAicBF	13	95.1	357.4	375.6	14.4	11.6	82.31	51.33	74.89	37.68
mdAicFB	9	93.8	396.8	409.4	18.6	14.8	43.21	30.36	33.28	25.47

Régression de Poisson simple

Régression de Poisson multiple

Sélection des variables et construction du modèle

Régression de Poisson pour tableaux de contingence: Les modèles log-linéaire

- Tableaux 2×2
- Tableaux I×J
- Tableaux I × J × K
- La sélection de modèles appliquée aux tableaux de contingence

Appendice

Il est naturel d'envisager le recours à un GLM-Poisson pour analyser les données d'un tableau de contingence.

La régression de Poisson, lorsqu'elle est utilisée pour modéliser des tableaux de contingence, est (historiquement) appelée modèle log-linéaire. C'est ce que nous allons présenter ici.

L'usage du modèle log-linéaire est approprié pour étudier les différents types de relations qui peuvent exister entre un certain nombre de variables qualitatives sans, a priori, devoir les distinguer en variables explicatives et variable à expliquer.

En règle générale, la finalité d'une modélisation log-linéaire est de trouver et d'interpréter la structure d'association qui décrit le plus fidèlement possible les données.

Tableaux 2×2

Les données observées se présentent sous la forme suivante ("Cat" pour catégorie et "Gr" pour groupe).

X|Y	$1 \equiv$ Cat1	$2 \equiv$ Cat2	
$1 \equiv$ Gr1	n_{11}	n_{12}	$n_{1.}$
$2 \equiv$ Gr2	n_{21}	n_{22}	$n_{2.}$
	$n_{.1}$	$n_{.2}$	n

En notant par N la variable de comptage, on peut, de façon équivalente, écrire ce tableau comme suit.

X	Y	N
1	1	n_{11}
1	2	n_{12}
2	1	n_{21}
2	2	n_{22}

En terme de régression, notre objectif ici est d'expliquer N (variable dépendante) à l'aide de (X, Y) (variables explicatives).

Notations

$$\mu(X,Y) = E(N|X,Y)$$
$$p_{ij} = P(X = i, Y = j),\ i = j = 1,2.$$
$$\mu_{ij} \equiv \mu(i,j) = E(N|X = i, Y = j) = np_{ij},\ i = j = 1,2.$$

Hypothèses

(1) $N|(X,Y) \sim \text{Pois}(\mu(X,Y))$,

(2) $\log(\mu(X,Y)) = \beta_0 + \beta_1^X I(X = 1) + \beta_1^Y I(Y = 1) + \beta_{11}^{XY} I(X = 1) \times I(Y = 1)$

Cette dernier équation est équivalente à

$$\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Y + \beta_{ij}^{XY}, i,j = 1,2, \text{ avec}$$

$\beta_2^X = \beta_2^Y = \beta_{12}^{XY} = \beta_{21}^{XY} = \beta_{22}^{XY} = 0$.

Notez qu'il s'agit ici d'un modèle saturé puisqu'il y a 4 paramètres $(\beta_0, \beta_1^X, \beta_1^Y, \beta_{11}^{XY})$ pour 4 observations $(n_{11}, n_{12}, n_{21}, n_{22})$.

L'équation du modèle se traduit par quatre sous-équations synthétisées dans le tableau suivant.

$\log(\mu_{ij})$	$j=1$	$j=2$
$i=1$	$\beta_0+\beta_1^X+\beta_1^Y+\beta_{11}^{XY}$	$\beta_0+\beta_1^X$
$i=2$	$\beta_0+\beta_1^Y$	β_0

Dans ce modèle, le paramètre β_{11}^{XY} représente l'effet d'interaction entre X et Y. C'est ce paramètre qui caractérise l'association entre X et Y.

En effet, il est facile de voir que

$$e^{\beta_{11}^{XY}}=\frac{p_{11}p_{22}}{p_{12}p_{21}}=or^{XY}$$ Voir App.

Et donc

$$X \perp\!\!\!\perp Y \text{ si et seulement si } \beta_{11}^{XY}=0$$

Exemple : Tabagisme

```
Whickham <- mosaicData::Whickham |>
            transform(smoker = factor(smoker, levels = c("Yes", "No")))

SmoOut <- xtabs(~ smoker + outcome, data = Whickham)
dfSmoOut <- SmoOut |> as.data.frame()
```

smoker | outcome	Alive	Dead
Yes	443	139
No	502	230

smoker	outcome	Freq
Yes	Alive	443
No	Alive	502
Yes	Dead	139
No	Dead	230

```
vcd::assocstats(SmoOut) |> summary()
```

```
                    X^2 df  P(> X^2)
Likelihood Ratio 9.2003  1 0.0024198
Pearson          9.1209  1 0.0025271
```

```
loddSmoOut <- vcd::loddsratio(SmoOut, log = FALSE)
loddSmoOut |> summary(); loddSmoOut |> confint()
```

```
                   Estimate Std. Error z value   Pr(>|z|)
Yes:No/Alive:Dead    1.4602    0.18349  7.9577 1.7523e-15
                    2.5 % 97.5 %
Yes:No/Alive:Dead 1.1414  1.868
```

```r
dfSmoOut <- transform(dfSmoOut, smoker = relevel(smoker, "No"), outcome = relevel(outcome, "Dead"))
MS <- glm(Freq ~ smoker * outcome, family = poisson(), data = dfSmoOut)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.438	0.066	82.473	0.0000e+00
smokerYes	-0.504	0.107	-4.688	2.7646e-06
outcomeAlive	0.781	0.080	9.803	1.0964e-22
smokerYes:outcomeAlive	0.379	0.126	3.013	2.5902e-03

$$\log(\hat{\mu}(\text{smoker}, \text{outcome})) = 5.438 - 0.504\mathrm{I}(\text{smoker} = \text{Yes}) + 0.781\mathrm{I}(\text{outcome} = \text{Alive}) + 0.379\mathrm{I}(\text{smoker} = \text{Yes}) \times \mathrm{I}(\text{outcome} = \text{Alive})$$

Remarque Noter que $\exp\left(\hat{\beta}_{11}^{XY}\right) = \exp(0.379) = 1.46 = \widehat{or}$. □

Les coefficients estimés s'interprètent exactement de la même manière que celle décrite auparavant.

Dans l'exemple présent, la présence d'une interaction (significative) entre les variables `smoker` et `outcome` complique un peu les choses.

Pour faciliter l'interprétation, nous pouvons splitter l'équation du modèle en deux parties :

$$\hat{\mu}(\text{No}, \text{outcome}) = \exp\left(5.438 + 0.781 \mathbb{I}(\text{outcome} = \text{Alive})\right) \quad \text{(les non-fumeurs)}$$
$$\hat{\mu}(\text{Yes}, \text{outcome}) = \exp\left(4.934 + 1.16 \mathbb{I}(\text{outcome} = \text{Alive})\right) \quad \text{(les fumeurs)}$$

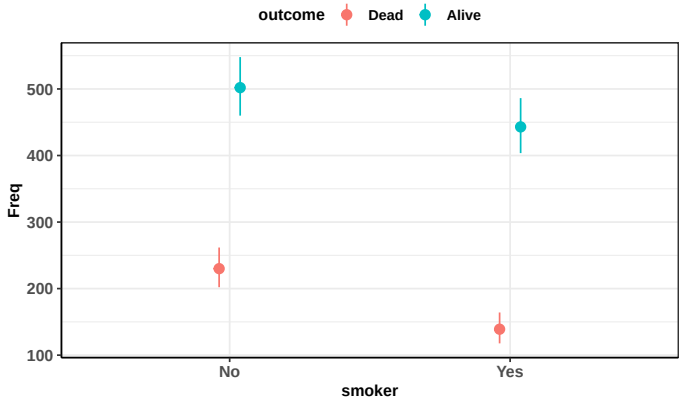
Ces équations/coefficients montrent simplement que les survivants sont majoritaires que ça soit dans le groupe des fumeurs ou des non-fumeurs avec une proportion plus importante chez les fumeurs.

C'est l'effet d'interaction (`smokerYes:outcomeAlive` : 0.379) qui, étant ici positif, suggère un "effet protecteur du tabagisme". Comme nous l'avons déjà mentionné, ce constat erroné est dû à l'omission de l'âge dans cette analyse.

Les valeurs prédites par ce modèle (saturé) coïncident avec les observations.

```
predictions(MS, newdata = datagrid(smoker = c("Yes", "No"), outcome = c("Dead", "Alive")))
plot_predictions(MS, condition = c("smoker", "outcome"))
```

smoker	outcome	Estimate	Pr(>|z|)	S	2.5 %	97.5 %
No	Dead	230	<0.001	Inf	202	262
No	Alive	502	<0.001	Inf	460	548
Yes	Dead	139	<0.001	Inf	118	164
Yes	Alive	443	<0.001	Inf	404	486



Test d'indépendance: façon GLM

Pour tester l'indépendance enter les variables `smoker` et `outcome`, il suffit de tester

$$H_0 : \beta_{11}^{XY} = 0 \quad \text{vs} \quad H_1 : \beta_{11}^{XY} \neq 0$$

Cela peut se faire de plusieurs manières.

» Test de Wald: voir la toute dernière cellule du `summary(MS)` (Slide 87).

» Test de LR:

```
drop1(MS, test = "LRT")
```

```
              Df Deviance  AIC LRT Pr(>Chi)
<none>              0.0 38.0
smoker:outcome  1      9.2 45.2 9.2   0.0024 **
```

» Test du Score (= test de Pearson):

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

```
              Df Deviance  AIC Rao score Pr(>Chi)
<none>              0.0 38.0
smoker:outcome  1      9.2 45.2      9.12   0.0025 **
```

Modèle log-linéaire

Tableaux $I \times J$

Avec les mêmes notations et hypothèses que précédemment (voir Slide 84), nous considérons ici le cas de deux variables catégorielles X et Y avec I et J modalités, respectivement. Dans ce qui suit, I (J) est la modalité de référence de X (Y). Le modèle de Poisson se présente sous la forme suivante:

$$\log(\mu(X,Y)) = \beta_0 + \sum_{i=1}^{I-1} \beta_i^X I(X=i) + \sum_{j=1}^{J-1} \beta_j^Y I(Y=j) + \sum_{i=1}^{I-1}\sum_{j=1}^{J-1} \beta_{ij}^{XY} I(X=i) \times I(Y=j).$$

Ou, de façon équivalente,

$$\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Y + \beta_{ij}^{XY}, i = 1,\ldots,I,\ j = 1,\ldots,J, \text{ avec}$$

$\beta_I^X = \beta_J^Y = 0$ et $\beta_{Ij}^{XY} = \beta_{iJ}^{XY} = 0$, $\forall i, \forall j$.

C'est un modèle saturé qui contient $I \times J$ paramètres.

On peut facilement montrer que

$$\exp\left(\beta_{ij}^{XY}\right) = \frac{p_{ij}p_{IJ}}{p_{iJ}p_{Ij}} \quad \forall i, j.$$

Il en résulte que

$$\beta_{ij}^{XY} = 0, \ \forall i, j \Leftrightarrow \ X \perp\!\!\!\perp Y \quad \text{(Pourquoi ?)}$$

Le modèle log-linéaire d'indépendance est donc le suivant

$$\log(\mu_{ij}) = \beta_0 + \beta_i^X + \beta_j^Y,$$

avec $\beta_I^X = \beta_J^Y = 0$.

Exemple : Tabagisme (Suite)

```
Whickham <- transform(Whickham, Age = cut(age, breaks = c(-Inf, 24, 65, Inf),
                                          labels = c("A1", "A2", "A3")))
AgeSmo <- xtabs(~ Age + smoker, data = Whickham)
dfAgeSmo <- AgeSmo |> as.data.frame()
```

Age | smoker	Yes	No
A1	55	72
A2	478	480
A3	49	180

Age	smoker	Freq
A1	Yes	55
A2	Yes	478
A3	Yes	49
A1	No	72
A2	No	480
A3	No	180

```
vcd::assocstats(AgeSmo) |>
  summary()
```

```
                    X^2 df   P(> X^2)
Likelihood Ratio 64.806  2 8.4377e-15
Pearson          60.889  2 5.9952e-14
```

```
loddAgeSmo <- vcd::loddsratio(AgeSmo, log = FALSE,
                              ref = dim(AgeSmo))
loddAgeSmo |>  coef(); loddAgeSmo |> confint()
```

```
             Estimate  2.5 % 97.5 %
A1:A3/Yes:No   2.8061 1.7500 4.4995
A2:A3/Yes:No   3.6582 2.6031 5.1409
```

```
dfAgeSmo <- transform(dfAgeSmo, smoker = relevel(smoker, "No"), Age = relevel(Age, "A3"))
MS <- glm(Freq ~ smoker * Age, family = poisson(), data = dfAgeSmo)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	5.193	0.075	69.671	0.0000e+00
smokerYes	-1.301	0.161	-8.075	6.7505e-16
AgeA1	-0.916	0.139	-6.571	4.9957e-11
AgeA2	0.981	0.087	11.222	3.1725e-29
smokerYes:AgeA1	1.032	0.241	4.283	1.8434e-05
smokerYes:AgeA2	1.297	0.174	7.471	7.9761e-14

REMARQUE Noter que $\exp\left(\hat{\beta}_{11}^{XY}\right) = \exp(1.032) = 2.81 = \hat{\text{or}}^{A1:A3/Yes:No}$.
De même $\exp\left(\hat{\beta}_{21}^{XY}\right) = \hat{\text{or}}^{A2:A3/Yes:No}$. □

Nous pouvons utiliser le modèle (`MS`) pour tester l'indépendance entre les variables `smoker` et `Age` en testant la nullité simultanée des coefficients d'interaction, càd tester l'hypothèse (multiple)

$$H_0 : \beta_{11}^{XY} = \beta_{21}^{XY} = 0 \quad \text{vs} \quad H_1 : \beta_{11}^{XY} \neq 0 \text{ ou } \beta_{21}^{XY} \neq 0.$$

```
drop1(MS, test = "LRT")
```

```
           Df Deviance   AIC  LRT Pr(>Chi)
<none>              0.0  52.7
smoker:Age  2     64.8 113.6 64.8  8.5e-15 ***
```

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

```
           Df Deviance   AIC Rao score Pr(>Chi)
<none>              0.0  52.7
smoker:Age  2     64.8 113.6      60.9    6e-14 ***
```

Modèle log-linéaire

Tableaux $I \times J \times K$

Considérons une table de contingence à trois variable X, à valeurs dans $\{1, \ldots, I\}$, Y, à valeurs dans $\{1, \ldots, J\}$, et Z, à valeurs dans $\{1, \ldots, K\}$.

Pour chaque niveau k ($k = 1, \ldots, K$) de Z, on dispose d'une table (partielle) $I \times J$

Z	X|Y	1	2	...	J
	1	n_{11k}	n_{12k}	...	n_{1Jk}
	2	n_{21k}	n_{22k}	...	n_{2Jk}
k	⋮	⋮	⋮	⋮	⋮
	I	n_{I1k}	n_{I2k}	...	n_{IJk}

Sous format d'un dataframe classique, les données se présentent comme

X	Y	Z	N
⋮	⋮	⋮	⋮
i	j	k	n_{ijk}
⋮	⋮	⋮	⋮

Soit $\mu_{ijk} \equiv \mu(i,j,k) = E(N|X=i, Y=j, Z=k)$. En suivant le même principe et les mêmes notations que ci-avant, le modèle saturé est donné par

$$\log(\mu_{ijk}) = \beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY} + \beta_{ik}^{XZ} + \beta_{jk}^{YZ} + \beta_{ijk}^{XYZ},$$

avec les contraintes usuelles sur les paramètres ($\beta_I^X = \beta_J^Y = \beta_K^Z = \beta_{Ij}^{XY} = \beta_{iJ}^{XY} = \beta_{Ik}^{XZ} = \beta_{iK}^{XZ} = \beta_{Jk}^{YZ} = \beta_{jK}^{YZ} = \beta_{Ijk}^{XYZ} = \beta_{iJk}^{XYZ} = \beta_{ijK}^{XYZ} = 0$, $\forall i,j,k.$).

Par exemple, pour $I = J = K = 2$, le modèle se réduit aux 8 équations figurant dans le tableau suivant.

Z	X|Y	$j=1$	$j=2$
$k=1$	$i=1$	$\beta_0 + \beta_1^X + \beta_1^Y + \beta_1^Z + \beta_{11}^{XY} + \beta_{11}^{XZ} + \beta_{11}^{YZ} + \beta_{111}^{XYZ}$	$\beta_0 + \beta_1^X + \beta_1^Z + \beta_{11}^{XZ}$
	$i=2$	$\beta_0 + \beta_1^Y + \beta_1^Z + \beta_{11}^{YZ}$	$\beta_0 + \beta_1^Z$
$k=2$	$i=1$	$\beta_0 + \beta_1^X + \beta_1^Y + \beta_{11}^{XY}$	$\beta_0 + \beta_1^X$
	$i=2$	$\beta_0 + \beta_1^Y$	β_0

Dans ce cas particulier ($I = J = K = 2$), on peut facilement montrer que

$$e^{\beta_{11}^{XY}} = or^{XY|Z=2} \text{ et } e^{\beta_{111}^{XYZ}} = \frac{or^{XY|Z=1}}{or^{XY|Z=2}}$$ ▸ Voir App

Et donc,

$$\beta_{111}^{XYZ} = 0 \Leftrightarrow \text{Association homogène entre } X, Y, \text{ et } Z$$
$$\beta_{111}^{XYZ} = 0 \text{ et } \beta_{11}^{XY} = 0 \Leftrightarrow X \perp\!\!\!\perp Y|Z.$$

La même analyse s'applique aux autres termes d'interaction:

$$\beta_{111}^{XYZ} = 0 \text{ et } \beta_{11}^{XZ} = 0 \Leftrightarrow X \perp\!\!\!\perp Z|Y$$
$$\beta_{111}^{XYZ} = 0 \text{ et } \beta_{11}^{YZ} = 0 \Leftrightarrow Y \perp\!\!\!\perp Z|X.$$

Ce qui implique, entre autres, que

$$\beta_{111}^{XYZ} = 0, \beta_{11}^{XY} = 0 \text{ et } \beta_{11}^{XZ} = 0 \Leftrightarrow X \perp\!\!\!\perp (Y, Z)$$
$$\beta_{111}^{XYZ} = 0, \beta_{11}^{XY} = 0, \beta_{11}^{XZ} = 0 \text{ et } \beta_{11}^{YZ} = 0 \Leftrightarrow X \perp\!\!\!\perp Y \perp\!\!\!\perp Z.$$

Tous les cas possibles sont résumés dans le tableau suivant, qui s'applique quelle que soit la valeur de I, J et K.

Notation	Description	Équation du modèle $\log(\mu_{ijk})$
	ind. mutu.	
(X, Y, Z)	$X \perp\!\!\!\perp Y \perp\!\!\!\perp Z$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z$
	ind. part.	
(Y, XZ)	$Y \perp\!\!\!\perp (X, Z)$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ik}^{XZ}$
(X, YZ)	$X \perp\!\!\!\perp (Y, Z)$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{jk}^{YZ}$
(Z, XY)	$Z \perp\!\!\!\perp (X, Y)$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY}$
	ind. cond.	
(XZ, YZ)	$X \perp\!\!\!\perp Y \mid Z$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ik}^{XZ} + \beta_{jk}^{YZ}$
(XY, XZ)	$Y \perp\!\!\!\perp Z \mid X$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY} + \beta_{ik}^{XZ}$
(XY, YZ)	$X \perp\!\!\!\perp Z \mid Y$	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY} + \beta_{jk}^{YZ}$
(XZ, XZ, YZ)	ass. homo.	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY} + \beta_{ik}^{XZ} + \beta_{jk}^{YZ}$
(XYZ)	saturé	$\beta_0 + \beta_i^X + \beta_j^Y + \beta_k^Z + \beta_{ij}^{XY} + \beta_{ik}^{XZ} + \beta_{jk}^{YZ} + \beta_{ijk}^{XYZ}$

Dans le tableau précédent, on note qu'il est possible de "passer" d'un type d'association à l'autre en confrontant des modèles log-linéaires emboîtés découlant les uns des autres par l'ajout ou la suppression d'un ou de plusieurs termes d'interaction.

Par exemple, nous pouvons tester l'indépendance mutuelle entre les variables `smoker`, `outcome`, et `Age`, en se basant sur la modélisation log-linéaire, en utilisant le code suivant.

```
dfSmoOutAge <- xtabs(~smoker + outcome + Age, data = Whickham) |>
  as.data.frame()
# mdèle saturé
MS <- glm(Freq ~ smoker * outcome * Age, family = poisson(), data = dfSmoOutAge)
# modèle d'indépendance mutuelle
M0 <- glm(Freq ~ smoker + outcome + Age, family = poisson(), data = dfSmoOutAge)
# test
anova(M0, MS, test = "Rao")
```

Tous les modèles de ce tableau respectent le principe de la hiérarchie tel que défini auparavant (si un terme d'interaction figure dans un modèle, il en va de même pour tous les termes d'ordre inférieur faisant intervenir les variables concernées.).

En général, seuls les modèles log-linéaires hiérarchiques sont d'intérêt.

Modèle log-linéaire

La sélection de modèles appliquée aux tableaux de contingence

Lorsqu'on analyse un tableau de contingence, l'objectif est souvent soit de tester une hypothèse prédéfinie concernant la dépendance entre les variables étudiées (comme dans l'exemple du Slide précédent), soit d'identifier, de façon générale, la structure d'association qui décrit les données de manière fiable et qui est aussi simple que possible à comprendre/interpréter. Dans le second cas, on peut utiliser les techniques de sélection de modèle/variable que nous avons déjà abordées. En voici une illustration.

"Backward elimination" avec test de Pearson

```
# run the code below line by line and see the results
drop1(MS, test = "Rao")
# modèle d'ass. homo.
MH <- update(MS, . ~ . - smoker:outcome:Age)
drop1(MH, test = "Rao")
# modèle d'ind. cond. (outcome, smoker)|age
MCiSmoOut <- update(MH, . ~ . - smoker:outcome)
drop1(MCiSmoOut, test = "Rao")  # --> no simplification is possible. STOP.
```

Selon cette procédure, le modèle (minimal) à choisir est celui de l'indépendance conditionnelle (MCiSmoOut): smoker $\perp\!\!\!\perp$ outcome | Age, dont la formule est

```
Freq ~ smoker + outcome + Age + smoker:Age + outcome:Age
```

"Forward-Backward selection" avec le BIC

```
MFBstepBic <- step(M0, scope = list(lower = M0, upper = MS), direction = "both",
                   k = log(12), trace = 0)

formula(MFBstepBic)

Freq ~ smoker + outcome + Age + outcome:Age + smoker:Age + smoker:outcome
```

Selon cette procédure, le modèle (minimal) à choisir est celui d'association homogène (MH).

Nous arrivons à la même conclusion (càd au modèle (MH)) si, au lieu de cette procédure, nous effectuons une recherche exhaustive, à l'aide de la fonction `glmulti()`.

```
glmulti(Freq ~ smoker * outcome * Age, family = poisson, data = dfSmoOutAge, crit = bic,
        marginality = TRUE, confsetsize = 10, plotty = FALSE, report = FALSE)
```

Ce que nous avons appris ci-haut peut être appliqué à des tableaux de contingence de taille plus importante (avec plus de variables). Voici un exemple (simulé) d'un tableau de dimension $2 \times 2 \times 2 \times 3$ correspondant à une étude sur un échantillon de 1000 automobilistes et 4 facteurs : *Profess* utilisation professionnelle de la voiture (oui/non), *Vitesse* respect de la vitesse maximale (oui/non), *Sexe* (H/F), et *Age* âge de la voiture (< 2 ans, entre 2 et 5 ans, > 5 ans).

Sexe	Age	Profess | Vitesse	non	oui
F	<2	non	11	33
		oui	20	52
	>5	non	38	61
		oui	32	46
	2-5	non	32	28
		oui	47	19
H	<2	non	12	22
		oui	70	51
	>5	non	25	33
		oui	47	70
	2-5	non	43	30
		oui	115	63

La méthode de sélection par recherche exhaustive et le critère BIC nous donnent le modèle suivant (Faites l'exercice vous-même).

```
                     Estimate Std. Error   z value   Pr(>|z|)
(Intercept)         2.7011384    0.17366 15.554209 1.4898e-54
Professoui          0.6391290    0.17652  3.620727 2.9378e-04
Vitesseoui          0.7013977    0.16960  4.135720 3.5384e-05
SexeH              -0.1288732    0.17966 -0.717308 4.7318e-01
Age>5               0.7762079    0.19046  4.075489 4.5918e-05
Age2-5              0.7938986    0.19197  4.135486 3.5420e-05
Professoui:Vitesseoui -0.2842755    0.13883 -2.047655 4.0594e-02
Professoui:SexeH    0.8278110    0.13704  6.040517 1.5362e-09
Vitesseoui:SexeH   -0.2795480    0.13537 -2.065037 3.8920e-02
Professoui:Age>5   -0.6532603    0.17556 -3.720971 1.9846e-04
Professoui:Age2-5  -0.4545287    0.17924 -2.535809 1.1219e-02
Vitesseoui:Age>5   -0.0089871    0.16659 -0.053948 9.5698e-01
Vitesseoui:Age2-5  -0.8628663    0.16464 -5.241037 1.5968e-07
SexeH:Age>5        -0.1760886    0.16737 -1.052072 2.9277e-01
SexeH:Age2-5        0.4138888    0.17074  2.424083 1.5347e-02
```

De cette sortie, nous pouvons, par exemple, constater que (1) la majorité des conducteurs professionnels sont des hommes, (2) les professionnels et les hommes ont tendance à ne pas respecter les limitations de vitesse, (3) contrairement aux autres catégories, les conducteurs de voitures récentes (< 2 ans) ont tendance à respecter les limitations de vitesse.

Régression de Poisson simple

Régression de Poisson multiple

Sélection des variables et construction du modèle

Régression de Poisson pour tableaux de contingence: Les modèles log-linéaire

Appendice

Interprétation des paramètres (Slide 7)

- Cas univarié: $\mu(X) = \exp(\beta_0 + \beta_1 X)$

X	$\mu(X)$
0	$\exp(\beta_0)$
x	$\exp(\beta_0 + \beta_1 x)$
$x + a$	$\exp(\beta_0 + a\beta_1 + \beta_1 x)$

$\rightarrow \exp(a\beta_1) = \dfrac{\mu(x+a)}{\mu(x)}$, en particulier, pour $a = 1$, $\exp(\beta_1) = \dfrac{\mu(x+1)}{\mu(x)}$.

- Cas bivarié: $\mu(\mathbf{X}) = \mu(X_1, X_2) = \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)$

X_1	X_2	$\mu(X_1, X_2)$
0	0	$\exp(\beta_0)$
x_1	x_2	$\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$
$x_1 + a$	x_2	$\exp(\beta_0 + a\beta_1 + \beta_1 x_1 + \beta_2 x_2)$

$\rightarrow \exp(a\beta_1) = \dfrac{\mu(x_1 + a, x_2)}{\mu(x_1, x_2)}$, en particulier, pour $a = 1$, $\exp(\beta_1) = \dfrac{\mu(x_1 + 1, x_2)}{\mu(x_1, x_2)}$.

À partir des équation figurant dans la ligne $(i = 1)$ du tableau du Slide 85, nous pouvons écrire

$$\begin{aligned}\beta_1^Y + \beta_{11}^{XY} &= \log\left(\frac{\mu_{11}}{\mu_{12}}\right) = \log\left(\frac{p_{11}}{p_{12}}\right) \\ &= \log\frac{P(Y=1|X=1)}{P(Y=2|X=1)} = \log\frac{P(Y=1|X=1)}{1-P(Y=1|X=1)} \\ &= \log o(Y=1|X=1)\end{aligned}$$

De même, à partir de la ligne $(i = 2)$, $\beta_1^Y = \log o(Y=1|X=2)$.

On en déduit que

$$\beta_{11}^{XY} = \log\frac{o(Y=1|X=1)}{o(Y=1|X=2)} = \log or^{XY}$$

Interaction et indépendance: tableau $2 \times 2 \times 2$ (Slide 98)

À partir du tableau du Slide 97, en suivant la même démarche que celle utilisée pour le cas d'un tableau 2×2, il est facile de voir que

» dans le sous-tableau $(K = 1)$, on a

$$\beta_{11}^{XY} + \beta_{111}^{XYZ} = \log \mathrm{or}^{XY|Z=1}$$

» dans le sous-tableau $(K = 2)$, on a

$$\beta_{11}^{XY} = \log \mathrm{or}^{XY|Z=2}$$

On en déduit que

$$\beta_{111}^{XYZ} = \log \frac{\mathrm{or}^{XY|Z=1}}{\mathrm{or}^{XY|Z=2}}$$