La régression logistique

Chapitre IV

Anouar El Ghouch

LSBA, Université catholique de Louvain, Belgium

Régression logistique binaire simple

Régression logistique binaire multiple

Régression logistique binaire simple

Introduction

Soit Y est une variable binaire $(1, 0)$ (« oui / non », « vrai / faux », ...) dont on cherche à expliquer les variations à l'aide d'une ou plusieurs variables explicatives, $X_1, \ldots, X_d$, pouvant être de *n'importe quel type*.

Soit

$$p(x) = P(Y = 1 | \mathbf{X} = \mathbf{x}),$$

où $\mathbf{X} = (X_1, \ldots, X_d)^t$.

Par définition, pour un vecteur donné $\mathbf{x} = (x_1, \ldots, x_d)^t$,

$$Y | \mathbf{X} = \mathbf{x} \sim \text{Ber}(p(\mathbf{x})).$$

Notez que,

$$\mathbb{E}(Y | \mathbf{X} = \mathbf{x}) = p(\mathbf{x}) \text{ et } \mathbb{V}\text{ar}(Y | \mathbf{X} = \mathbf{x}) = p(\mathbf{x})(1 - p(\mathbf{x})).$$

Dans le but de simplifier la compréhension, lorsque cela ne nuit pas à la généralisation des notions exposées, nous nous focaliserons (au début) sur la cas où $d = 1$.

Pourquoi pas un modèle linéaire?

Puisuqe $\mathbb{E}(Y|X = x) = p(x)$, un modèle linéaire classique simple reliant Y à X s'écrit comme

$$p(x) = \beta_0 + \beta_1 x.$$

Ce qui représenterait plusieurs défauts:

- L'incohérence entre la partie droite et gauche de cette équation puisque $p(X) \in [0, 1]$ alors que, en général, $X \in (-\infty, \infty)$.
- Pour l'inférence, les hypothèses, fondamentales et fort utiles, de normalité et d'homoscédasticité sont clairement inappropriées puisque $Y|X \sim Ber(p(X))$ et que $\mathbb{V}ar(Y|X) = p(X)(1 - p(X))$.

Si, malgré tout, un modèle linéaire est ajusté à une réponse binaire, cela donnera lieu, typiquement, à un modèle sans grand intérêt.

Exemple 1 : maladie cardiovasculaire

On a relevé l'âge (`AGE`) et la présence ("m") ou l'absence ("s") d'une maladie cardiovasculaire (`CHD`) chez 100 individus. Les données sont stockées dans le fichier chd.csv.

```
maladie <- read.csv("data/chd.csv")
head(maladie)
```

```
  AGE CHD
1  20   s
2  23   s
3  24   s
4  25   s
5  25   m
6  26   s
```

```
str(maladie)
```

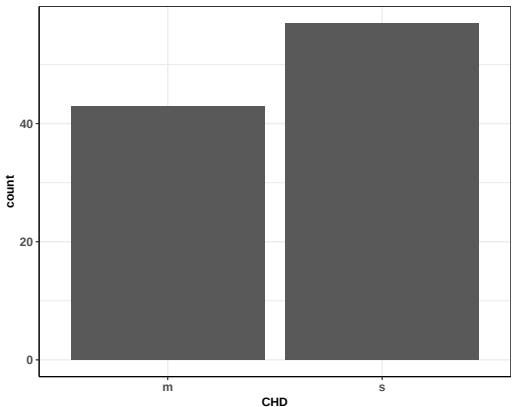
```
'data.frame': 100 obs. of  2 variables:
 $ AGE: int  20 23 24 25 25 26 26 28 28 29 ...
 $ CHD: chr  "s" "s" "s" "s" ...
```

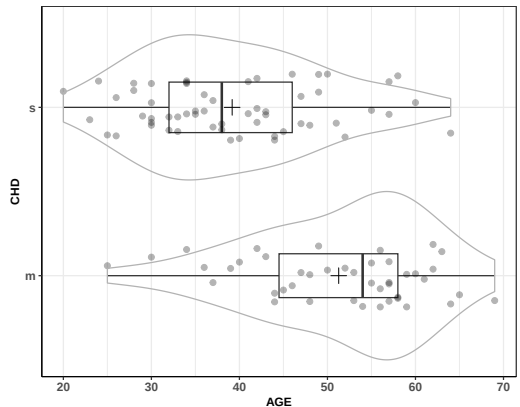
```
maladie$CHD <- factor(maladie$CHD)
```

```
summary(maladie)
```

AGE	CHD
Min. :20.0	m:43
1st Qu.:34.8	s:57
Median :44.0	
Mean :44.4	
3rd Qu.:55.0	
Max. :69.0	

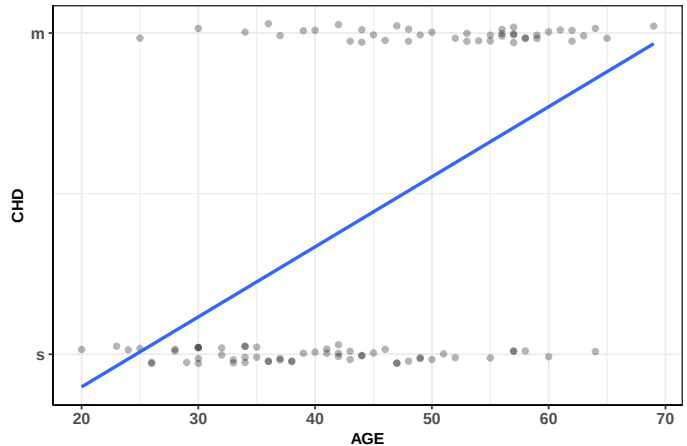
```r
ggplot(maladie, aes(CHD)) + geom_bar()
ggplot(maladie, aes(x = AGE, y = CHD)) + geom_boxplot(width = 0.3, varwidth = TRUE, outlier.shape = NA, fill = NA) +
  geom_jitter(alpha = 0.3, size = 2, width = 0, height = 0.2) + stat_summary(geom = "point", fun = "mean", size = 4, shape = 3) +
  geom_violin(fill = NA, width = 1, color = "gray70")
```





Le graphique suivant illustre ce qu'on obtient si on réalise une régression linéaire simple sur ces données.

```
ggplot(maladie, aes(x = AGE, y = ifelse(CHD == "m", 1, 0))) + geom_jitter(height = 0.03, width = 0, alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) + scale_y_continuous(breaks = c(0, 1), labels = c('s', 'm')) + labs(y = "CHD")
```



La droite de régression rate ici clairement son objectif, elle passe à côté de la majorité des données. $\rightarrow$ la droite n'explique pas les variations observées ($R^2 = 26.4\%$).

Exemple 2 : L'effet d'un insecticide

On dispose de données relatives à l'effet d'un insecticide (plus précisément, la dose appliquée de ce dernier) sur une espèce d'insecte. La variable d'intérêt est

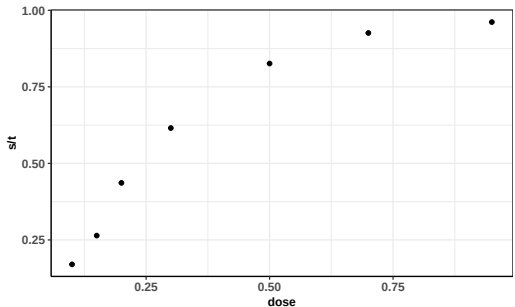
$$Y = (0, 1) = \text{ (insecte vivant, insecte mort)}$$

Les données (agrégées (A)) sont

```
dose <- c(0.1, 0.15, 0.2, 0.3, 0.5, 0.7, 0.95) # dose appliquée, mesurée en $g/cm^3$
s <- c(8, 14, 24, 32, 38, 50, 50)              # nbr. d'insectes tués (succès)
t <- c(47, 53, 55, 52, 46, 54, 52)             # nbr. total d'insectes exposés
dfA <- data.frame(dose, s, t)
```

i	dose	s	t
1	0.10	8	47
2	0.15	14	53
3	0.2	24	55
4	0.3	32	52
5	0.5	38	46
6	0.7	50	54
7	0.95	50	52

$n = 359$



Il est important de noter que, dans cet exemple, on dispose de résultats d'expérience réalisée avec répétition: chaque dose a été appliquée sur plusieurs individus à la fois.

Il faut aussi noter que ces données sont présentées ici sous forme agrégée (groupée) et non pas sous forme brute (un individu/insecte par ligne). Dans la littérature, les désignations "données binaires" et "données binomiales" (binary and binomial data, en anglais) sont souvent utilisées pour distinguer et modéliser ces deux type de données.

Dans ce qui suit, nous allons nous focaliser sur le cas de données agrégées avec un échantillon sous forme

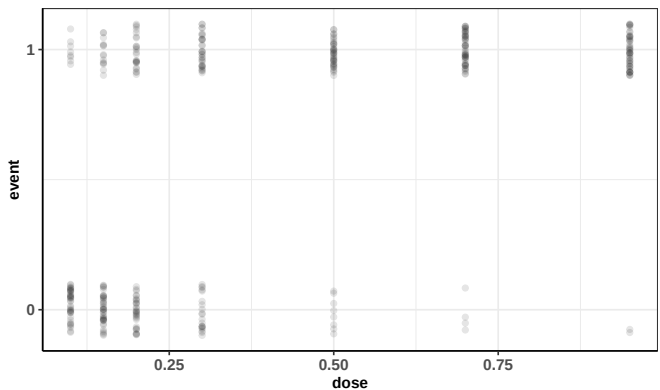
$$(\text{var. explicatives, nbr. succès, nbr. total}) = (x_i, s_i, t_i),\ i = 1, \ldots, I.$$

Les données brutes (encodées en 0 et 1) peuvent être considérées comme un cas particulier $\rightarrow$ elles s'écrivent aussi sous cette même forme, mais avec, en plus, $t_i = 1$ et $s_i = y_i \in \{0, 1\}$, $\forall i$. Avec cette convention, toutes les formules qu'on développera par la suite restent d'application dans les deux cas.

Bien entendu, il est toujours possible de transformer des données agrégées en données brutes (format long) et vice versa.

```
# données brutes (B)
dfB <- dfA |> transform(e = t - s, t = NULL) |> pivot_longer(cols = c(s, e), names_to = "event") |>
  uncount(weights = value)
```

```
# A tibble: 359 x 2
    dose event
   <dbl> <chr>
 1   0.1 s
 2   0.1 s
 3   0.1 s
 4   0.1 s
 5   0.1 s
 6   0.1 s
 7   0.1 s
 8   0.1 s
 9   0.1 e
10   0.1 e
# i 349 more rows
```



Comme expliqué ci-dessus, un modèle (linéaire) qui stipulerait que $p(x) = \beta_0 + \beta_1 x$, n'a ni de sens ni d'intérêt au vu de l'incohérence manifeste entre la partie gauche et la partie droite de cette équation.
Une façon de résoudre ce problème tout en gardant une structure simple est de relier $p(x)$ à $\beta_0 + \beta_1 x$ via une fonction de lien $g : (0,1) \to (-\infty, \infty)$ et d'écrire

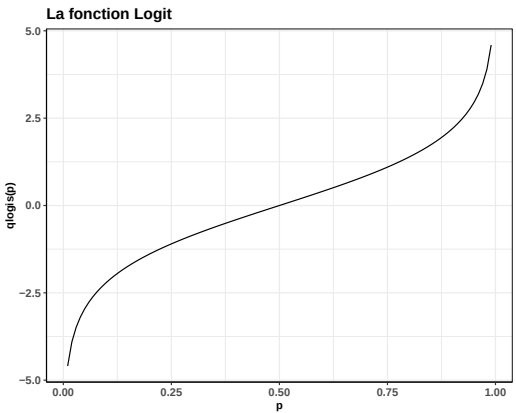
$$g(p(x)) = \beta_0 + \beta_1 x.$$

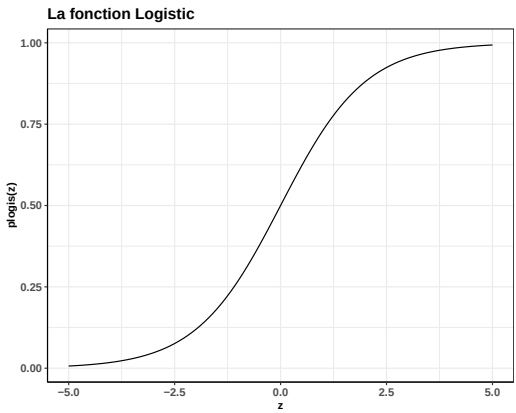
Il y a plusieurs choix possibles pour g, mais la fonction la plus utilisée, et de loin, en pratique est la fonction logit (`qlogis()` en `R`), qui s'avère être la fonction de lien canonique pour une Bernoulli/Binomial.

$$g(p) \equiv \text{logit}(p) = \log\left(\frac{p}{1-p}\right), \text{ pour } p \in (0,1).$$

L'inverse de la fonction logit est la fonction logistique (`plogis()` en `R`):

$$g^{-1}(z) \equiv \text{logistic}(z) = \frac{1}{1+e^{-z}}, \text{ pour } z \in (-\infty, \infty).$$





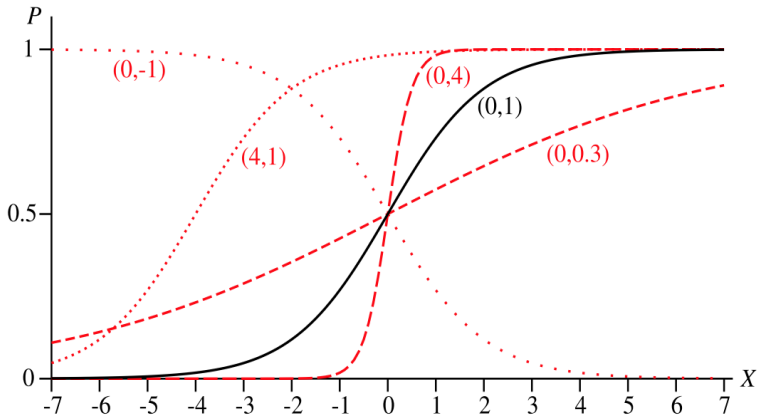
Le modèle logistique (ou logit) simple avec comme prédicteur une variable X (continue) stipule que

$$\text{logit}(p(x)) \equiv \log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x, \ \forall x.$$

Ce qui est équivalent à écrire que

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \equiv \text{logistic}(\beta_0 + \beta_1 x), \ \forall x.$$

Voici $p(X)$ pour différents valeurs de (β_0, β_1)



Quelques propriétés

» $\beta_1 = 0 \Leftrightarrow X$ et Y sont indépendantes.

» $\beta_1 > 0 \Leftrightarrow p(X)$ croît en fonction de $X \rightarrow$ association positive.

» $\beta_1 < 0 \Leftrightarrow p(X)$ décroît en fonction de $X \rightarrow$ association négative.

Interprétation des paramètres

Soit $o(x) = \dfrac{p(x)}{1-p(x)} = \dfrac{P(S|X=x)}{P(E|X=x)} =$ cote de succès (événement "$Y = 1$") quand $X = x$.

Selon l'équation de notre modèle $o(x) = \exp(\beta_0 + \beta_1 x)$, $\forall x$, donc

X	$o(X)$
0	$\exp(\beta_0)$
x	$\exp(\beta_0 + \beta_1 x) = \exp(\beta_0)\exp(\beta_1 x)$
$x+1$	$\exp(\beta_0 + \beta_1 + \beta_1 x) = \exp(\beta_0)\exp(\beta_1)\exp(\beta_1 x)$

$\Rightarrow \exp(\beta_1) = \dfrac{o(x+1)}{o(x)} =$ or entre deux groupes différant d'une unité par rapport à X.

Remarque

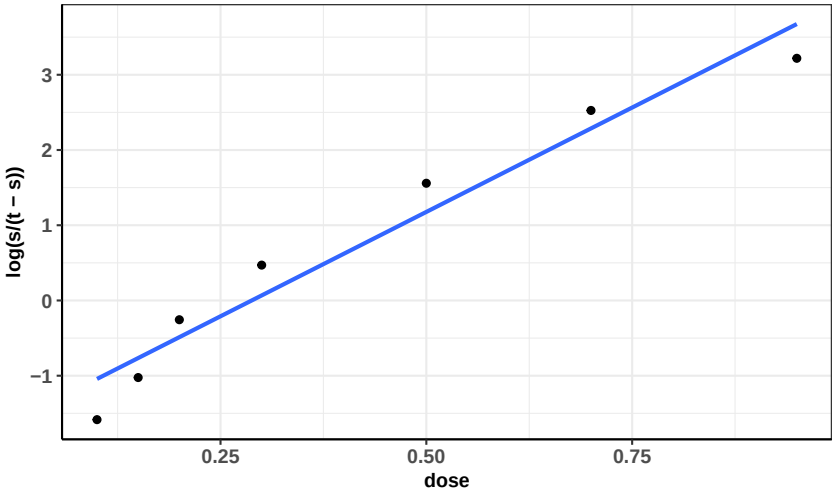
Une des complications avec la régression logistique (et plus généralement avec les GLM) est la difficulté de visuellement investiguer la "faisabilité/conformité" du modèle. Pour un modèle linéaire, il "suffit" de faire un diagramme de dispersion et voir si les points semblent plus au moins alignés.

En régression logistique, une telle façon de procéder n'est possible que si l'on dispose de *données répétées*.

Le graphique suivant illustre la démarche à suivre dans un tel cas; l'exemple traité est celui des données sur l'insecticide (Exemple 2).

Sans ou avec très peu de mesures répétées (comme c'est le cas de notre Exemple 1), un tel graphique n'est pas possible puisque dans ce cas $t_i = 1$ et $s_i = 0$ ou 1. Une manière de remédier à ce problème est de regrouper les données dont la valeur du prédicteur est relativement proche ....

```
ggplot(dfA, aes(x = dose, y = log(s/(t-s)))) + geom_point() + geom_smooth(method = "lm", se = FALSE)
```



Dans cette figure, nous pouvons voir un certain alignement des points. Cela indique qu'un modèle logistique peut convenir pour expliquer et prédire ces données.

Régression logistique binaire simple

Estimation, prédiction et diagnostic

Un modèle logistique étant un cas particulier de GLM, les techniques d'estimation, d'inférence et de diagnostic développées dans les deux derniers chapitres restent valables ici. Il en va de même pour tous les outils et fonctions `R` utilisés dans la régression de Poisson. Ces outils s'appliquent de la même manière à la régression logistique.

C'est pourquoi l'accent sera mis ci-après sur les éléments nouveaux ou nécessitant une adaptation. Pour le reste, le lecteur est renvoyé aux chapitres précédents.

On dispose de I observations, sous forme

$$(x_i, s_i, t_i) = (\text{var. explicative, nbr. succès, nbr. total}).$$

On suppose que ces observations proviennent d'un échantillon de variables indépendantes avec, pour $i = 1, \ldots, I$,

$$\begin{aligned} &(1) \quad S_i | \mathbf{X}_i = x_i \sim \text{Bin}(t_i, p_i) \\ &(2) \quad p_i = \text{logistic}(x_i^t \boldsymbol{\beta}) \equiv 1/(1 + e^{-\eta_i}), \end{aligned}$$

où

$$\begin{aligned} \eta_i &= x_i^t \boldsymbol{\beta} = \beta_0 x_{i,0} + \beta_1 x_{i,1} + \ldots \beta_d x_{i,d}, \\ x_i &= (x_{i,0}, x_{i,1}, \ldots, x_{i,d})^t = (1, x_{i,1}, \ldots, x_{i,d})^t, \\ \boldsymbol{\beta} &= (\beta_0, \beta_1, \ldots, \beta_d)^t. \end{aligned}$$

(1) implique que

$$P(S_i = s_i | \mathbf{X}_i = x_i) = C_{s_i}^{t_i} p_i^{s_i} (1 - p_i)^{t_i - s_i}.$$

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

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{I} \left(s_i \log(p_i) + (t_i - s_i) \log(1 - p_i) + \log\left(C_{t_i}^{s_i}\right) \right),$$

$$S_j := \frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{I} \left(s_i \left(\frac{1}{p_i} + \frac{1}{1 - p_i} \right) - \frac{t_i}{1 - p_i} \right) \frac{\partial p_i}{\partial \beta_j} = \sum_{i=1}^{I} (s_i - t_i p_i) \, x_{i,j},$$

$$H_{jk} := \frac{\partial S_j}{\partial \beta_k} = -\sum_{i=1}^{I} x_{i,j} x_{i,k} p_i (1 - p_i) t_i,$$

où $\dfrac{\partial p_i}{\partial \beta_j} = \dfrac{\partial p_i}{\partial \eta_i} \times \dfrac{\partial \eta_i}{\partial \beta_j} = \dfrac{e^{-\eta_i}}{(1 + e^{-\eta_i})^2} x_{i,j} = p_i (1 - p_i) x_{i,j}.$

À partir de là, nous pouvons calculer (numériquement) $\hat{\boldsymbol{\beta}} = \arg\max l(\boldsymbol{\beta})$, l'EMV de $\boldsymbol{\beta}$, faire l'inférence, et effectuer des prédictions et des diagnostics. Pour ce faire, nous aurons également besoin des quantités suivantes, qui peuvent être facilement calculées à partir des définitions fournies dans le chapitre sur les GLM.

L'EMV de p_i est $\hat{p}_i = \text{logistic}(\mathbf{x}_i^t \hat{\boldsymbol{\beta}}_M)$. Pour le modèle saturé, l'EMV de p_i est $\tilde{p}_i = s_i/t_i$, et, pour le modèle nul, l'EMV de p_i est $\hat{p}^0 = \bar{s} = \sum_i s_i / \sum_i t_i$. La déviance est

$$D^2 = 2(l^s - l) = 2\sum_{i=1}^{I} \left(s_i \log\left(\frac{s_i}{t_i \hat{p}_i}\right) + (t_i - s_i) \log\left(\frac{t_i - s_i}{t_i - t_i \hat{p}_i}\right)\right).$$

De même nous pouvons calculé la déviance-nul, $D_0^2 = 2\left(l^s - l^0\right)$, et aussi les résidus de Pearson

$$r_i = \frac{s_i - t_i \hat{p}_i}{\sqrt{t_i \hat{p}_i (1 - \hat{p}_i)}},$$

et ceux de la déviance

$$d_i = \text{sign}(s_i - t_i \hat{p}_i) \sqrt{2\left(s_i \log\left(\frac{s_i}{t_i \hat{p}_i}\right) + (t_i - s_i) \log\left(\frac{t_i - s_i}{t_i - t_i \hat{p}_i}\right)\right)}.$$

Enfin, il y a aussi les résidus quantiles randomisées (rqr), tels qu'ils ont été définis dans le chapitre sur les GLM.

Lorsque l'on dispose de données avec répétition (données Binomiale), ces résidus se révèlent très utiles pour détecter des anomalies dans le modèle étudié (choix inapproprié de la fonction de lien, termes/variables manquants ou nécessitant une transformation, etc.)

Les graphiques des résidus (typiquement, rqr ou déviance) versus les valeurs ajustées ($\hat{p}_i$) et des résidus versus chaque variable explicative (quand il y en a plusieurs) sont les plus souvent utilisés en pratique. Ces graphiques doivent montrer des points répartis aléatoirement autour de 0 sans structure ou tendance particulières et sans valeurs aberrantes.

Il convient aussi d'utiliser le QQ-plot des rqr pour repérer un écart par rapport à la distribution supposée (Bernoulli/Binomial) ou un choix inapproprié de la fonction de lien.

Pour les données sans répétition (données de Bernoulli), ces résidus sont presque toujours sans intérêt; voir ci-dessous pour plus de détails.

Exemple "insecticide" (Slide 6)

Ajustement du modèle

```
regA <- glm(s/t ~ dose, weights = t, family = binomial(link = logit), data = dfA)
#or# regA <- glm(cbind(s, e = t-s) ~ dose, family = binomial, data = dfA)
```

					Exponentiated		
	Estimate	Std.Error	z.value	p.value	estimate	CI.lwr	CI.upr
(Intercept)	-1.736	0.242	-7.173	<0.001	0.176	0.108	0.279
dose	6.295	0.742	8.482	<0.001	542.067	139.292	2585.245

Soit $o(dose)$ la cote de décès d'un insecte, prise au hasard, auquel on a administré la quantité "dose" d'insectside. Le modèle logistique estime que

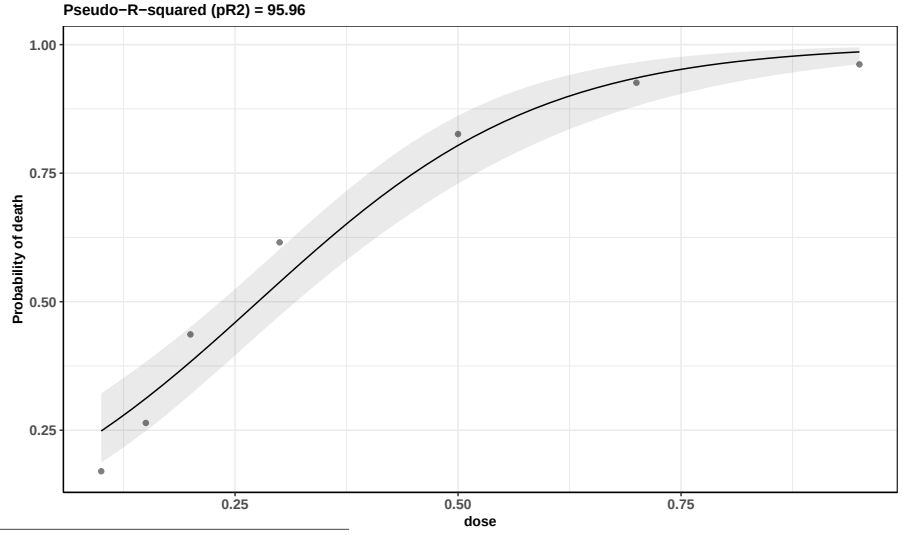
$$\hat{o}(dose) = \exp(-1.736 + 6.295 \times dose).$$

$\rightarrow$ On estime que la cote de décès se multiplie par $542 \approx exp(6.295)$ chaque fois que la dose d'insectside augmente d'une unité; soit une hausse d'environ 54100%.

Prédire la probabilité de décès

```
prd <- data.frame(dose = c(0.16, 0.32, 0.79)) |> predict(regA, newdata = _, se = TRUE)
data.frame(fit = prd$fit) |> transform(lwr = fit - 1.96 * prd$se.fit,
  upr = fit + 1.96 * prd$se.fit) |> apply(2, FUN = plogis)

      fit     lwr     upr
1 0.32545 0.26198 0.39605
2 0.56916 0.50269 0.63323
3 0.96221 0.91999 0.98258

#or# data.frame(dose = c(0.16, 0.32, 0.79)) |> predictions(regA, newdata = _)
```

```
plot_predictions(regA, condition = "dose") + geom_point(data = regA$data, aes(x = dose, y = s / t), alpha = 0.5) +
  labs(y = "Probability of death", subtitle = paste("Pseudo-R-squared (pR2) =", pr2(regA) |> round(2)))
```



Utiliser `type = "link"` pour visualiser l'équation du modèle sur l'échelle linéaire

```
plot_predictions(regA, condition = "dose", type = "link", vcov = FALSE)
```

Prédiction inverse

On peut aussi utiliser le modèle pour prédire la dose nécessaire pour atteindre une probabilité de succès π donnée. Selon notre modèle, cette dose est

$$\text{dose}(\pi) = \frac{\text{logit}(\pi) - \beta_0}{\beta_1}.$$

Que l'on peut estimer par

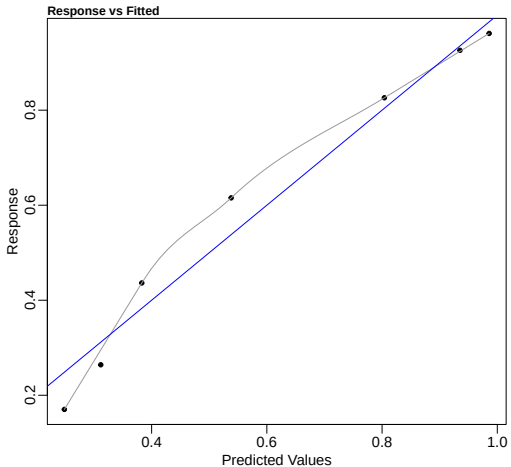
$$\widehat{\text{dose}}(\pi) = \frac{\text{logit}(\pi) - \hat{\beta}_0}{\hat{\beta}_1}.$$

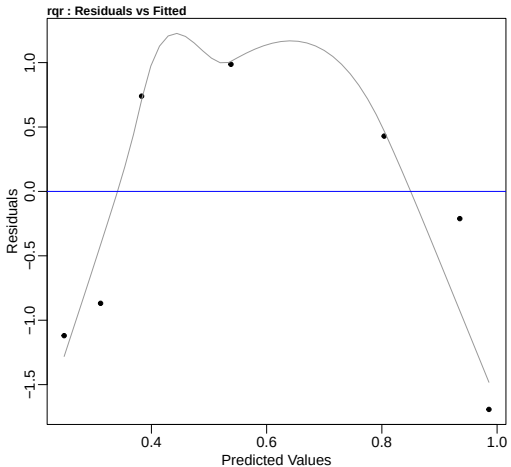
Par exemple, la dose médiane, càd la dose qui entrainerait le décès de 50% des individus, est estimée à $-\hat{\beta}_0/\hat{\beta}_1 \approx$ `0.276` . Il s'agit ici d'une estimation ponctuelle qu'on pourra compléter éventuellement par un intervalle de confiance.

```
MASS::dose.p(regA, p = 0.5)

            Dose       SE
p = 0.5: 0.27577 0.020984
```

Les résidus

```
diagnost(regA, plots = c("response", "fitted"))
```

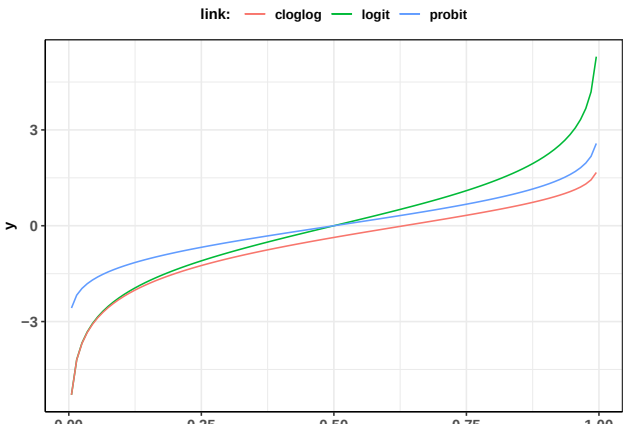




Compte tenu de la forme des résidus, nous pouvons envisager, par exemple, une autre fonction de lien ou une transformation de la variable prédictive (ici $dose$).

Pour la loi binomiale, R propose les liens suivant:

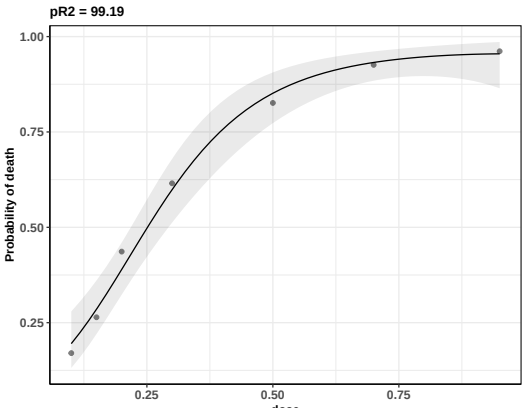
- `logit` : $g(p) = \log(p/(1-p))$, le lien par défaut
- `probit` : $g(p) = \Phi^{-1}(p)$, $\Phi =$ CDF de la $\mathcal{N}(0,1)$
- `cauchit` : $g(p) = F^{-1}(p)$, $F =$ CDF de la $Cauchit(0,1)$
- `log` : $g(p) = \log(p)$,
- `cloglog` : $g(p) = \log(-\log(1-p))$.

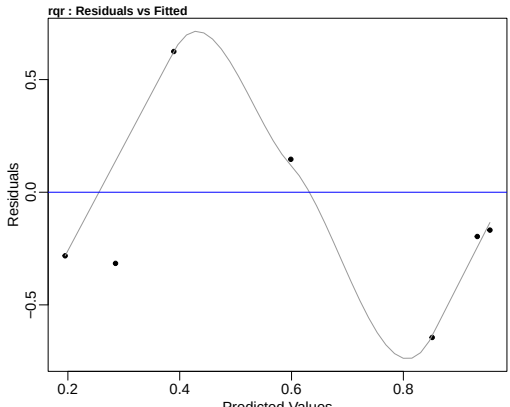


Dans ce qui suit, nous examinerons une série de modèles pour voir si nous pouvons faire mieux que `regA`.

Autres modèles possibles

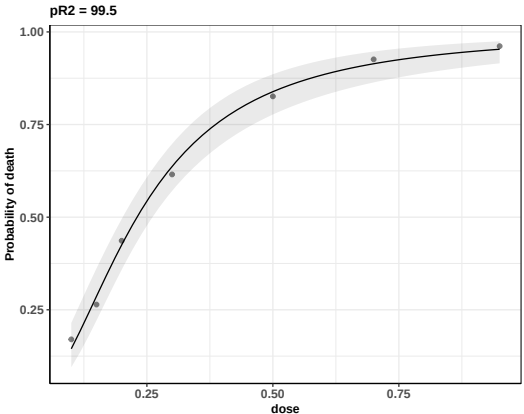
```
regA2 <- glm(s/t ~ dose + I(dose^2), weights = t, family = binomial, data = dfA)
```

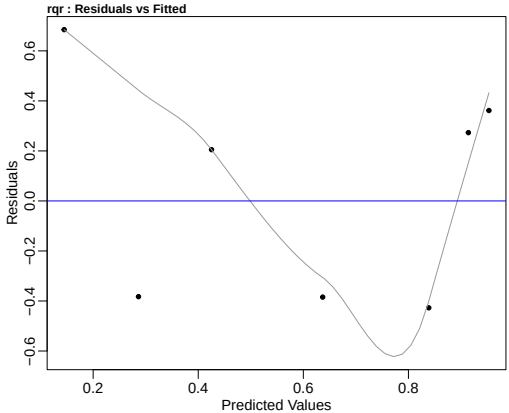




→ ce modèle semble meilleur que le précédent.

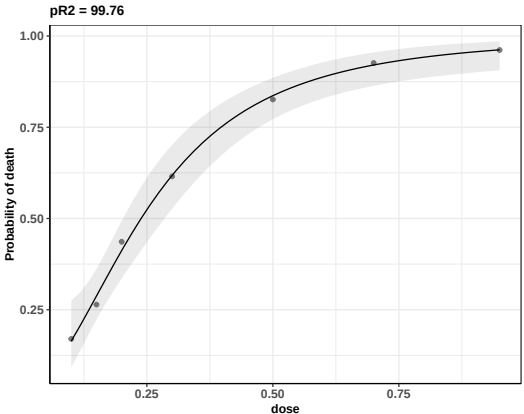
```
regA1 <- glm(s/t ~ log(dose), weights = t, family = binomial, data = dfA)
```

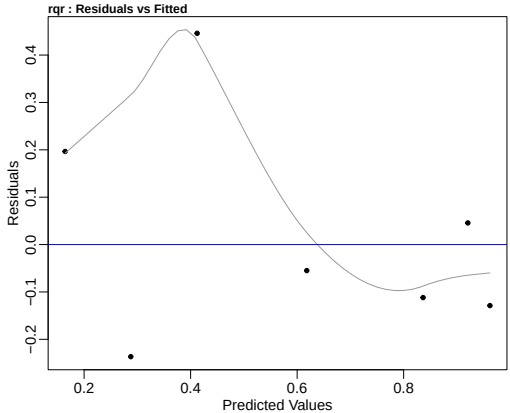




→ malgré la forme parabolique suggérée par la courbe lissée des résidus, ce modèle semble s'ajuster quasi parfaitement aux données. Il a également l'avantage d'être plus simple que le modèle quadratique (`regA2`).

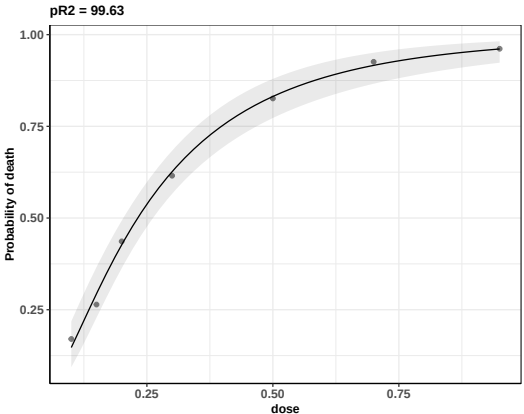
```r
regAl2 <- glm(s/t ~ log(dose) + I(log(dose)^2), weights = t, family = binomial, data = dfA)
```

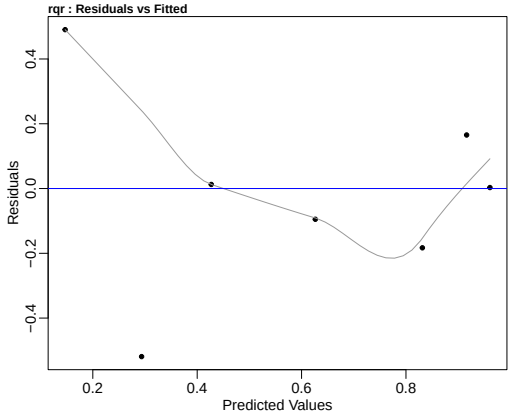




→ légère amélioration par rapport au modèle précédent (`regAl`), mais cela vaut-il vraiment la peine ?

```r
regAp <- glm(s/t ~ log(dose), weights = t, family = binomial(probit), data = dfA)
```





→ ce modèle (probit) semble légèrement meilleur que le modèle (logit) `regAl`. Cependant, d'un point de vue pratique, la régression logistique est beaucoup plus "reconnaissable/acceptable" dans les sciences de la vie.

Données de Bernoulli et les résidus

Pour les données de Bernoulli (càd les données binaires non groupées), les résidus de Pearson et de déviance ne sont pas très utiles pour effectuer des diagnostics, sauf peut-être pour détecter des valeurs aberrantes.

En effet, dans ce cas, les écarts $s_i - t_i\hat{p}_i$, qui constituent l'essentielle de ces résidus, se réduisent à $y_i - \hat{p}_i$ or ceci vaut soit $-\hat{p}_i$ ou $1 - \hat{p}_i$. Ce qui ne peut en rien refléter la qualité d'ajustement!

Voyons cela avec notre exemple de maladie cardiovasculaire; voir Slide 3.

Avant d'aller plus loin, commençons par définir le niveau de référence de la réponse à l'aide de la fonction `relevel()`. C'est ce niveau qui sera automatiquement encodé comme "0" par R lors de l'appel de la fonction `glm()`. C'est ce qu'on peut vérifier avec la fonction `contrasts()`.

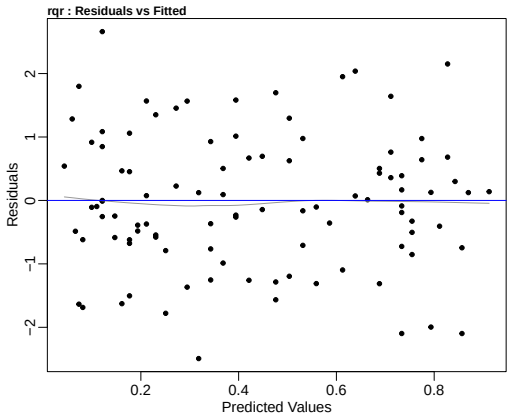
```
maladie$CHD <- relevel(maladie$CHD, "s")
contrasts(maladie$CHD)
```

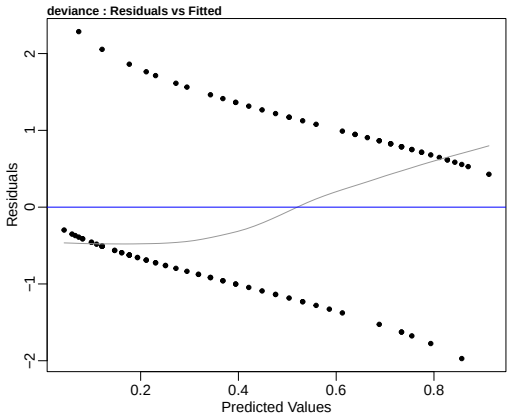
```
  m
s 0
m 1
```

De façon générale, avant d'écrire un modèle GLM, il faut s'assurer (i) que chaque variable catégorielle est bien considéré telle quelle par R, et (ii) du bon choix du niveau de référence pour chaque variable factorielle impliquée dans le modèle, et en particulier pour la réponse.

En effet, pour interpréter correctement une sortie `glm()`, il faut comprendre l'encodage utilisé en interne par R faute de quoi on risque d'inverser les événements "succès" et "échec" et, au final, calculer $P(Y = 0)$ au lieu de $P(Y = 1)$. Notez que, dans sa sortie, R ne fournit aucune information à ce sujet.

```r
regM <- glm(CHD ~ AGE, family = binomial, data = maladie)
diagnost(regM, plot = "fitted")
diagnost(regM, plot = "fitted", type = "deviance")
```





$\rightarrow$ il est difficile de tirer quelque chose de pertinent des résidus de déviance (graphique de droite) en raison de leur nature discrète. Les résidus des quantiles randomisés présentent un profil plus cohérent.

Données de Bernoulli et le pR^2

Avec des données de Bernoulli, il est assez courant de voir de faibles valeurs de pR^2 *même lorsque le modèle ajusté est de très bonne qualité.*

```
pr2(regM)
```

```
[1] 21.447
```

Un examen plus approfondi de la définition de pR^2 révèle que pour que ce dernier atteigne 1 (sa valeur maximale), les p_i estimés par le modèle doivent coïncider avec les réponses individuelles ($y_i = 0, 1$), mais cet "idéal" est inaccessible, voire impossible à frôler.

Les faibles valeurs de pR^2 qu'on obtient typiquement avec des données de Bernoulli traduisent simplement la difficulté pour le modèle logistique à prédire des observations binaires individuelles.

Cela pose un problème lorsqu'il s'agit de communiquer ces pR^2 à un public habitué à voir des valeurs beaucoup plus grandes dans le cadre de la régression linéaire classique. C'est pourquoi on conseille de ne montrer ces pR^2 qu'à un public bien informé!

Remarque

Si nous effectuons la régression logistique sur les données incecteisde, mais qu'au lieu d'utiliser les données groupées comme précédemment, nous utilisons les données au format brut (`dfB`), nous obtiendrons les mêmes estimations pour les coefficients du modèle (β_0 et β_1). En effet, à une constante près ($\sum_i \log(C_{t_i}^{s_i})$), les deux log-vraisemblances (Bernouilli vs. Binomile) sont identiques.

Il convient toutefois de noter que la différence entre les deux log-vraisemblances (Bernouilli vs. Binomile) impact les autres statistiques basées sur la vraisemblance, telles que la déviance, pR^2, l'AIC, etc.

```
dfB <- dfB |> transform(event = factor(event) |> relevel("e"))
regB <- glm(event ~ dose, family = binomial, data = dfB)
pr2(regB)
```

```
[1] 29.027
```

Régression logistique binaire simple

Matrice de confusion et ROC

Une manière de quantifier la qualité d'un modèle logistique (avec données brutes ou groupées) est de se focaliser sur ses performances prédictives. Nous allons voir comment faire cela à l'aide de la matrice dite de confusion et de certains critères de performances qui en découlent.

Pour illustrer la démarche, considérant les données sur la maladie cardiovasculaire (Slide 3). Le modèle `regM` qu'on a ajusté à ces données nous permet d'estimer, pour tout individu i, dont on connait l'âge, la probabilité p_i qu'il ait la maladie. Or, notre variable d'intérêt (Y = 'CHD') est de type binaire avec deux modalités: "m" pour malade (1) et "s" pour saint (0).

Si nous voulons utiliser notre modèle pour classifier un individu dans l'une ou l'autre de ces deux catégories, càd classifier un individu comme étant négatif (0) ou positif (1), nous devons trouver une manière pour convertir nos probabilités estimées en une variable (de classification) binaire. Cela peut se faire facilement en se fixant un seuil c (cutoff ou threshold, en anglais), typiquement $c = 0.5$, et en procédant de la manière suivante.

» Si $\hat{p}_i > c$, on assigne l'individu i au groupe (1), càd le groupe d'individus positifs (ou les cas, ici les malades)

» Si $\hat{p}_i \leqslant c$, on assigne l'individu i au groupe (0), càd le groupe d'individus négatifs (ou les contrôles, ici non-malades)

Pour $c = 0.5$, cela revient à définir la variable $\hat{Y}_i = I(\hat{p}_i > 0.5)$.

```
maladie <- transform(maladie, predM = fitted(regM))
maladie <- transform(maladie,
 pCHD = ifelse(predM > 0.5, "m", "s") |> factor(levels = c("s", "m")))
```

La sortie à droite montre les 6 premières et les 6 dernières lignes de `maladie`. CHD représente les valeurs observées et pCHD représente les statuts prédits. Le modèle sera "bon" si les positifs sont classés positifs, et les négatifs sont classés négatifs.

	CHD	predM	pCHD
1	s	0.043479	s
2	s	0.059621	s
3	s	0.066153	s
4	s	0.073344	s
5	m	0.073344	s
6	s	0.081248	s
95	m	0.827449	m
96	m	0.842716	m
97	s	0.856866	m
98	m	0.856866	m
99	m	0.869939	m
100	m	0.912465	m

À partir de ces résultats, nous pouvons construire ce que l'on appelle la matrice de confusion, qui n'est rien d'autre qu'un tableau de contingence dans lequel nous comptons le nombre d'individus correctement classés dans chaque modalité, ainsi que le nombre d'individus incorrectement classés. Cette matrice prend la forme suivante.

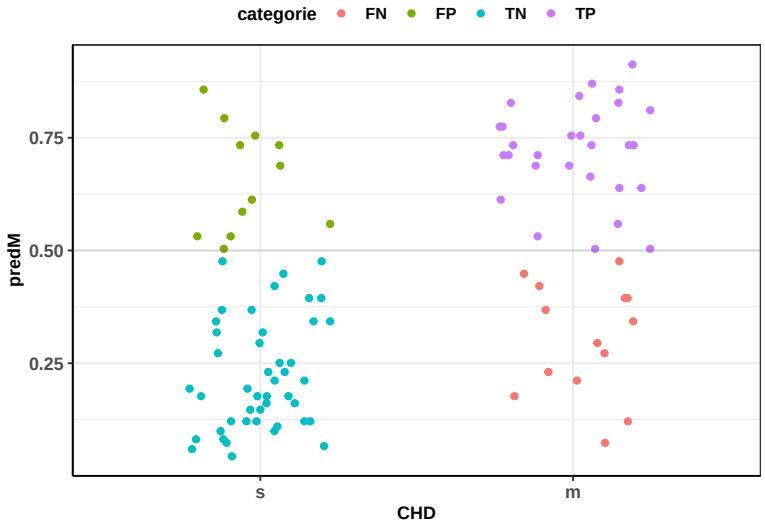
	Y (Observed response)	
pY (Predicted response)	0	1
$0: -$	TN	FN
$1: +$	FP	TP

Où l'on peut lire le nombre de

- TN (True Negative): Individus classés négatifs à juste titre.
- FP (False Positive): Individus classés positifs à tort.
- FN (False Negative): Individus classés négatifs à tort.
- TP (True Positive): Individus classés positifs à juste titre.

Le graphique suivant montre les observations et leurs catégories de classification.

```
maladie <- transform(maladie, categorie = ifelse(pCHD == "m" & CHD == "m", "TP", ifelse(pCHD == "m" & CHD == "s", "FP",
  ifelse(pCHD == "s" & CHD == "s", "TN", "FN"))))
ggplot(maladie, aes(x = CHD, y = predM, color = categorie)) + geom_jitter(height = 0, width = 0.25) +
  geom_abline(slope = 0, intercept = 0.5, alpha = 0.1)
```



Dans notre exemple la matrice de confusion est

```
xtabs(~pCHD + CHD, data = maladie)
```

```
     CHD  s  m
pCHD
s         45 14
m         12 29
```

À partir de cette matrice on peut dériver tout un tas de critères de performance. Parmi ces critères il y a :

» La précision (*accuracy*): $P(\hat{Y} = Y)$. Ce que l'on peut estimer par la proposition d'individus correctement classés.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{29 + 45}{29 + 45 + 12 + 14} = 0.74.$$

» *sensitivity* (ou TPR: *True Positive Rate*): $P(\hat{Y} = 1|Y = 1) \approx$ la proposition d'individus positifs correctement classés comme tels

$$SEN = \frac{TP}{TP + FN} = \frac{29}{29 + 14} = 0.674.$$

$\rightarrow$ 67% des malades sont correctement classés comme tels par le modèle.

» specificity (ou $1 - \text{FPR}$, FPR: *False Positive Rate*): $P(\hat{Y} = 0|Y = 0) \approx$ la proposition d'individus négatifs correctement classés comme tels

$$\text{SPE} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \frac{45}{12 + 45} = 0.789.$$

$\rightarrow$ 79% des non-malades sont correctement classés comme tels par le modèle.

Ces statistiques (et bien d'autres) quantifient ce que l'on désigne comme étant le pouvoir discriminatoire du modèle. Dans cette logique, on peut définir le modèle parfait comme celui qui présente une sensibilité et une spécificité de 100%. La précision synthétise ces deux critères, puisque

$$\text{ACC} = \text{Prevalence} \times \text{SEN} + (1 - \text{Prevalence}) \times \text{SPE},$$

où $\text{Prevalence} = P(Y = 1) \approx$ la proportion des individus positifs.

La fonction `confusionMatrix()` du package `caret` permet de calculs tous ces quantités facilement.

```r
confusionMatrix(maladie$pCHD, maladie$CHD, positive = "m")
```

```
Confusion Matrix and Statistics

          Reference
Prediction  s  m
         s 45 14
         m 12 29

               Accuracy : 0.74
                 95% CI : (0.643, 0.823)
    No Information Rate : 0.57
    P-Value [Acc > NIR] : 0.000319

                  Kappa : 0.467

 Mcnemar's Test P-Value : 0.844519

            Sensitivity : 0.674
            Specificity : 0.789
         Pos Pred Value : 0.707
         Neg Pred Value : 0.763
             Prevalence : 0.430
```

Un inconvénient de cette façon de procéder pour mesurer le pouvoir discriminatoire du modèle est le fait d'utiliser les même données pour à la fois estimer le modèle et pour en juger sa qualité, ce qui est susceptible de conduire à une surestimation de cette dernière.

Une façon de contourner ce problème est d'utiliser la méthode de validation croisée (CV) telle que décrite dans le chapitre précédent. L'approche et les calculs restent identiques, la seule différence réside dans la manière de quantifier la qualité prédictive du modèle. En effet, au lieu d'utiliser le risque quadratique (RMSE), il faut utiliser une mesure plus adapté à la nature des données et du modèle, comme par exemple la précision (accuracy).

La fonction `caret::train()` est programmée pour adapter les mesures de performance à la nature du problème (régression vs. classification).

La procédure consiste à (1) diviser les données en K Folds, (2) estimer le modèle en utilisant $K - 1$ Folds, (3) calculer la précision en utilisant le K—ième Fold restant comme test set, (4) refaire les calculs en parcourant tous les Folds, et (5) répéter l'ensemble de l'opération autant de fois que possible (disons $L = 100$ fois). Au final, nous obtenons $L \times K$ valeurs de précision, dont la moyenne est calculée pour obtenir la la "cross-validated accuracy" ($cvACC$).

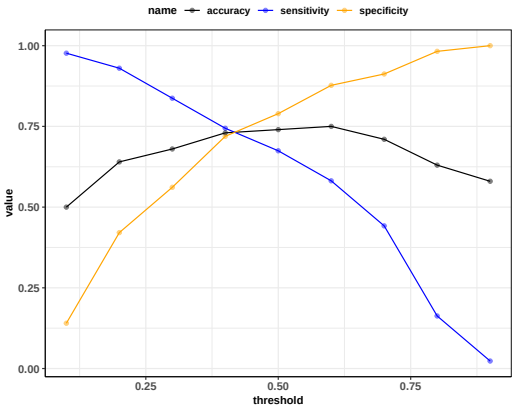
```
set.seed(1)
train(CHD ~ AGE, data = maladie, family = binomial, method = "glm",
  trControl = trainControl(method = "repeatedcv", number = 10, repeats = 100))

Summary of sample sizes: 91, 91, 90, 89, 90, 90, ...
Resampling results:

  Accuracy  Kappa
  0.73273   0.44771
```

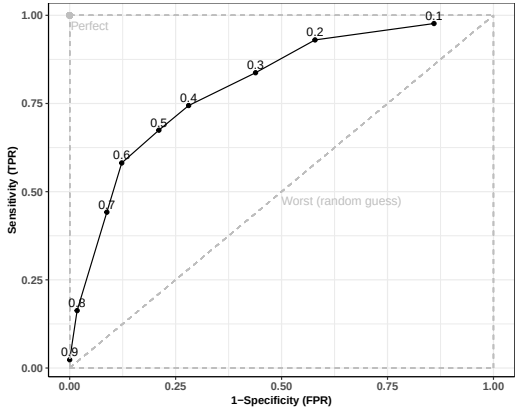
Un autre inconvénient des critères tels que la précision, la sensibilité et la spécificité (calculés avec ou sans CV) est qu'ils dépendent du seuil c initialement choisi pour effectuer les calculs. Il est possible de pousser l'analyse plus loin en variant ce seuil.

Nous allons répéter les calculs précédents pour différentes valeurs de c entre 0 et 1. Pour faciliter ces calculs, nous allons utiliser le package `pROC` et ces fonctions `roc()` et `coords()`.

```
rocM <- roc(maladie, response = CHD, predictor = predM, levels = c("s", "m"))
coords(rocM, x = seq(0.1, 0.9, 0.1), ret = c("threshold", "specificity", "sensitivity", "accuracy"))
```

```
  threshold specificity sensitivity accuracy
1       0.1     0.14035    0.976744     0.50
2       0.2     0.42105    0.930233     0.64
3       0.3     0.56140    0.837209     0.68
4       0.4     0.71930    0.744186     0.73
5       0.5     0.78947    0.674419     0.74
6       0.6     0.87719    0.581395     0.75
7       0.7     0.91228    0.441860     0.71
8       0.8     0.98246    0.162791     0.63
9       0.9     1.00000    0.023256     0.58
```





Le graphique de droite montre une courbe connue sous le nom de la courbe ROC. C'est une représentation graphique du TPR (Sensitivity) par rapport au FPR (1− Specificity) pour toutes les valeurs seuils c possibles.

Une courbe ROC ne se retrouve jamais en dessous de la diagonale. Cette dernière sert de ligne de référence puisqu'il s'agit de la courbe ROC d'un classificateur purement aléatoire dont $\text{SEN} = 1 - \text{SPE}$, $\forall c$.

Un classificateur parfait est celui dont $\text{SEN} = \text{SPE} = 1$, $\forall c$. Plus la courbe ROC est proche du coin supérieur gauche du carré, meilleur est le modèle.

D'une manière générale, lorsque le seuil c augmente, la spécificité augmente, tandis que la sensitivité diminue. Le contraire est aussi vrai.

La courbe ROC peut être utilisée pour comparer deux ou plusieurs classificateurs/-modèles en traçant leurs courbes sur le même graphique. La courbe ROC la plus éloignée de la diagonale représente le meilleur modèle.

Il sexiste plusieurs statistiques qui permettent de résumer la courbe ROC en un seul chiffre, ce qui facilite la comparaison des classificateurs/modèles. Les plus connues de ces statistiques sont l'indice de Youden (J_{max}) et l'AUC.

L'indice de Youden est définit comme

$$J_{max} = \max_{c}(SEN(c) + SPE(c) - 1)$$

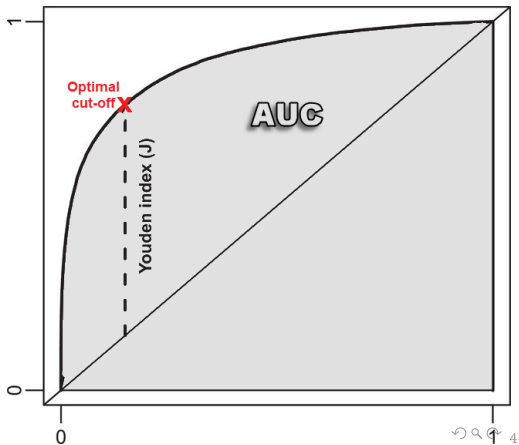
C'est une quantité qui combine la sensitivité et la spécificité en une seule mesure et a une valeur comprise entre 0 (le moins bon) et 1 (le meilleur).

Graphiquement, il s'agit de la distance verticale maximale entre la courbe ROC et la ligne diagonale (du hasard pur).

Le seuil qui correspond à ce maximum, $c_{max} = \arg\max_c(SEN(c) + SPE(c) - 1)$, est très sauvant utilisé comme le seuil optimal.

```
coords(rocM, x = "best")
```

```
  threshold specificity sensitivity
1    0.3813     0.68421      0.7907
```



L'AUC (Area Under the ROC Curve) n'est rien d'autre que l'aire sous la courbe ROC. Un classificateur/modèle parfait a une AUC de 1, tandis qu'un classificateur/modèle purement aléatoire a une AUC de 0.5.

Comme l'indice de Youden, et contrairement à la précision (accuracy), l'AUC ne dépend ni de la prévalence ni d'un quelconque seuil.

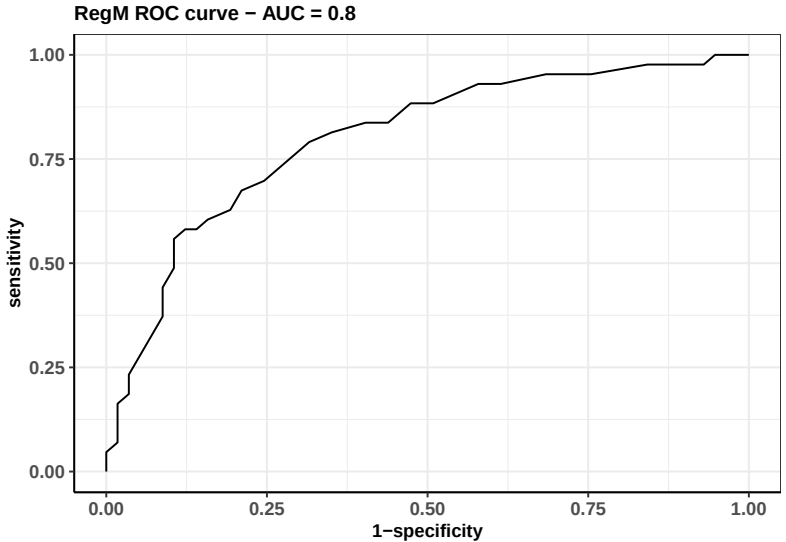
L'AUC coïncide avec l'indice de corcondance (C—index) et peut être interprétée comme la probabilité qu'un individu (A) aléatoirement sélectionné parmi les positifs ait probabilité prédite plus élevée qu'un individu (B) aléatoirement sélectionné parmi les négatifs, càd $P(\hat{p}_A > \hat{p}_B | Y_A > Y_B)$.

Voici un système de classement (subjectif) de classificateurs/modèles sur base de leur pouvoir discriminatoire tel que mesuré par l'AUC.

AUC	Qualité prédictive du modèle
$(0.95 - 1]$	Exceptionnel
$(0.85 - 0.95]$	Excellent/très bon
$(0.75 - 0.85]$	Bon
$(0.7 - 0.75]$	Acceptable
$(0.6 - 0.7]$	Médiocre
$[0.5 - 0.6]$	Pas ou quasi pas de discrimination

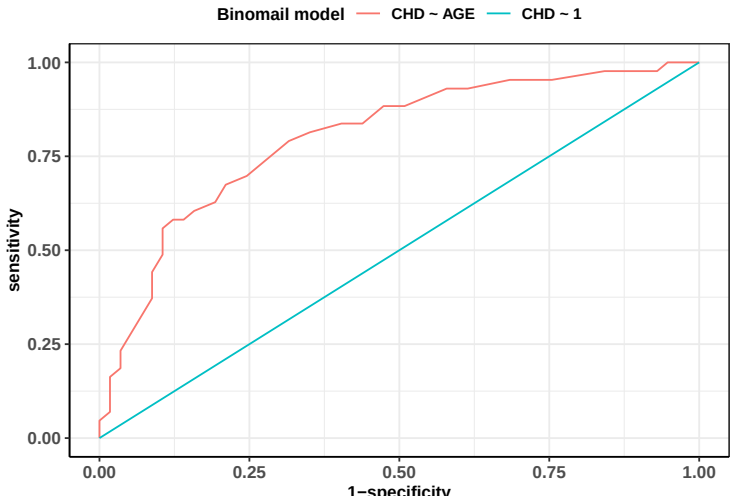
Voici la courbe ROC et l'AUC pour notre modèle logistique `regM: CHD ~ AGE`

```
ggroc(rocM, legacy.axes = TRUE) + labs(subtitle  = paste("RegM ROC curve - AUC =", round(rocM$auc, 2)))
```



Il est également possible de placer plusieurs courbes sur un même graphique. Voici un exemple.

```
Reg0 <- glm(CHD ~ 1, family = binomial, data = maladie)
roc0 <- roc(response = maladie$CHD, predictor = fitted(Reg0))
list("CHD ~ AGE"  = rocM, "CHD ~ 1" = roc0) |> ggroc(legacy.axes = TRUE) + labs(color = "Binomail model")
```



La validation croisée peut être utilisée pour obtenir une estimation plus pertinente de l'AUC. Sur le plan méthodologique, la procédure est identique à celle décrite ci-dessus pour la précission; voir Slide 43. À la fin des calculs, on obtient la "cross-validated AUC" (cvAUC). En ce qui concerne le code R, quelques modifications doivent être apportées pour effectuer les calculs.

```
fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 100,
  classProbs = TRUE, summaryFunction = twoClassSummary, savePredictions = T)

set.seed(1)
train(CHD ~ AGE, data = maladie, family = binomial, method = "glm",
  trControl = fitControl, metric = "ROC")

Summary of sample sizes: 91, 91, 90, 89, 90, 90, ...
Resampling results:

  ROC      Sens     Spec
  0.80053  0.79243  0.6537
```

Dans cette sortie, le chiffre étiqueté ROC est le cvAUC recherché.

Exemple: Les manchots de Palmer

Les données "penguins", stockées dans le fichier penguins.csv, contiennent des mesures effectuées par des chercheurs de la station Palmer sur les manchots. "penguins" contient 344 lignes/individus et 8 colonnes qui sont :

- species : l'espèce; une variable à trois modalités (Adelie, Chinstrap, Gentoo)
- island : île; une variable à trois modalités (Biscoe, Dream, Torgersen),
- bill_length_mm : la longueur du bec en millimètre (mm),
- bill_depth_mm : l'épaisseur du bec en mm,
- flipper_length_mm : la longueur des nageoires en mm,
- body_mass_g : le poids en gramme,
- sex : sexe (female, male),
- year : année d'étude (2007, 2008 ou 2009).

Comme il se doit, nous commençons notre étude par une simple analyse descriptive.

```r
penguins <- read.csv(file = "data/penguins.csv")
head(penguins)
str(penguins)

# transformer des variables en Factor
penguins <- transform(penguins, species = factor(species), island = factor(island),
                                sex = factor(sex), year = factor(year))
summary(penguins)

# les données contiennent 11 valeurs manquantes (NA); les voici
penguinsNA <- subset(penguins, !complete.cases(penguins)) |> print()

# nous considérons uniquement les lignes sans NA (n = 344 --> n = 333).
penguins <- subset(penguins, complete.cases(penguins))
```

Ici, notre objectif final est de construire un modèle permettant de prédire, aussi bien que possible, le sexe d'un manchot sur base des informations recueillies.

Avant cela nous allons étudier quelques modèles logistiques avec l'objectif de se donner le temps de mieux comprendre la méthode et de se l'approprier.

Régression logistique binaire multiple

Prédicteurs continus

Pour simplifier cet exposé, nous nous focalisons ici sur le cas d'un modèle binomial avec seulement de deux variables explicatives.

Notre objectif est de comprendre la relation entre une réponse binaire $Y \in \{1 \equiv S, 0 \equiv E\}$ et deux prédicteurs continus X et Z.

Introduisant d'abord les notations suivantes

$$p(x,z) = P(Y = 1|X = x, Z = z) \text{ et } o(x,z) = p(x,z)/(1 - p(x,z))$$

Comme pour le cas d'une seul variable explicative, le modèle logistique stipule que

$$(1) : Y|(X,Z) \sim Ber(p(X,Z))$$
$$(2) : \text{logit}(p(X,Z)) = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ$$

Selon cette équation

X	Z	$o(X,Z)$
x	z	$\exp(\beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz)$
$x+1$	z	$\exp(\beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz)\exp(\beta_1 + \beta_3 z)$

$\rightarrow$ le rapport des cotes associé à une augmentation d'une unité de la variable X, tout en maintenant Z fixe, est

$$\frac{o(x+1,Z)}{o(x,Z)} = \exp(\beta_1 + \beta_3 Z).$$

C'est l'effet de X sur la réponse Y.
À cause de l'interaction, cet effet change en fonction de la valeur prise par Z.

En cas d'absence d'interaction ($\beta_3 = 0$), l'effet de X sur Y serait le même quelque soit la valeur de Z et se résumerait à $\exp(\beta_1)$.

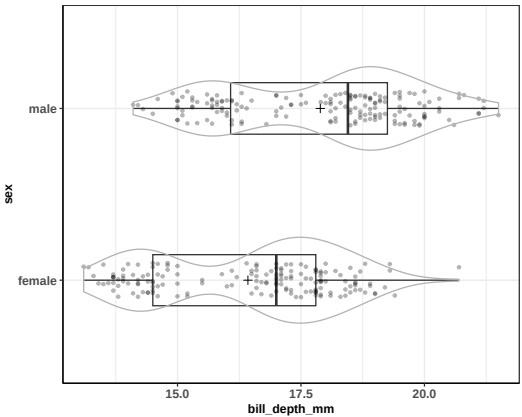
Le même raisonnement s'applique à Z:

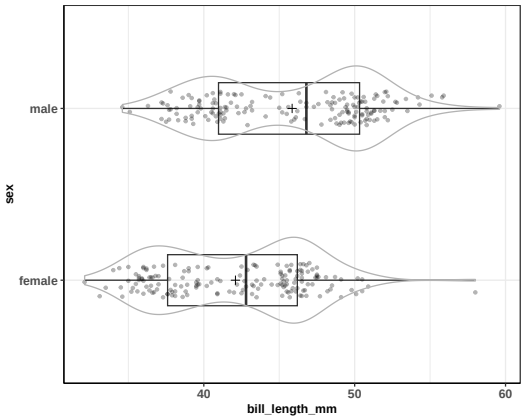
$$\frac{o(X,z+1)}{o(X,z)} = \exp(\beta_2 + \beta_3 x).$$

À titre d'exemple, considérons la régression logistique avec $Y = \mathrm{I}(sex = \text{'male'})$ comme variable dépendante, $X = bill\_depth\_mm$ et $Z = bill\_length\_mm$ comme variables explicatives.

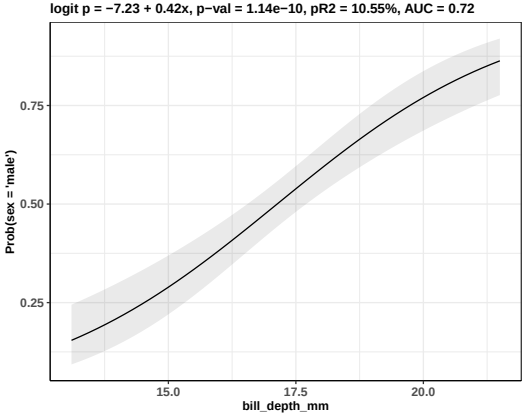
Pour commencer étudions le lien entre Y et X d'une part et entre Y et Z d'autre part.

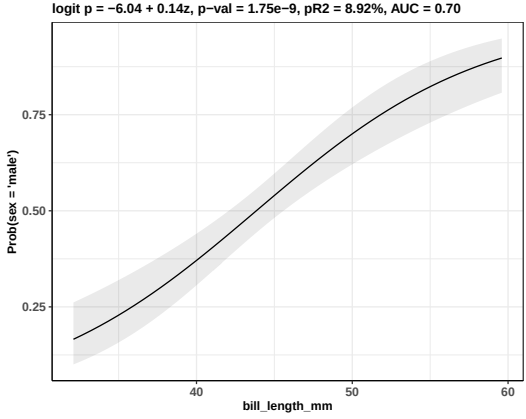
```r
pls <- ggplot() + geom_boxplot(width = 0.3, varwidth = TRUE, outlier.shape = NA, fill = NA) +
  geom_jitter(alpha = 0.3, size = 1, width = 0, height = 0.1) + stat_summary(geom = "point", fun = "mean",
  size = 2, shape = 3) + geom_violin(fill = NA, width = 0.5, color = "gray70")
ggplot(penguins, aes(x = bill_depth_mm, y = sex)) + pls$layers
ggplot(penguins, aes(x = bill_length_mm, y = sex)) + pls$layers
```





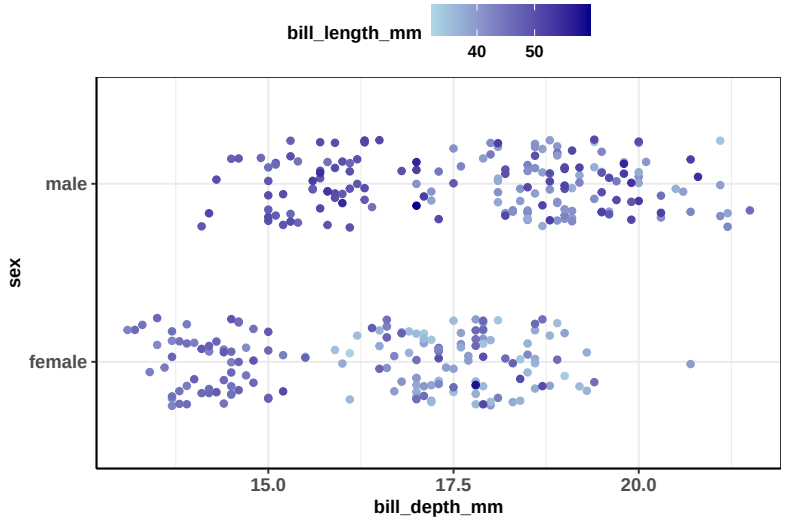
```r
# sex ~ bill_depth_mm
mbd <- glm(sex ~ bill_depth_mm, family = binomial, data = penguins)
summary(mbd) |> coef()
pr2(mbd)
roc(mbd$data$sex, fitted(mbd)) |> auc()
# sex ~ bill_length_mm
mbl <- glm(sex ~ bill_length_mm, family = binomial, data = penguins)
summary(mbl) |> coef()
pr2(mbl)
roc(mbl$data$sex, fitted(mbl)) |> auc()
```





Nous allons à présent incorporer les deux variables explicatives (bill_depth_mm et bill_length_mm) simultanément dans notre analyse.

```
ggplot(penguins, aes(x = bill_depth_mm, y = sex, color= bill_length_mm)) + geom_jitter(width = 0, height = 0.25) +
  scale_color_gradient(low = "lightblue", high = "darkblue")
```



```r
mbdbl <- glm(sex ~ bill_depth_mm*bill_length_mm, family = binomial, data = penguins)
summary(mbdbl) |> coef()
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-130.614	22.034	-5.928	3.0694e-09
bill_depth_mm	6.686	1.204	5.553	2.8141e-08
bill_length_mm	2.584	0.468	5.516	3.4741e-08
bill_depth_mm:bill_length_mm	-0.130	0.026	-5.067	4.0418e-07

Avec les notations $\texttt{bill\_depth\_mm} \equiv \text{BD}$ et $\texttt{bill\_length\_mm} \equiv \text{BL}$, l'équation du modèle logistique $sex \sim \text{BD} + \text{BL} + \text{BD} \times \text{BL}$ est

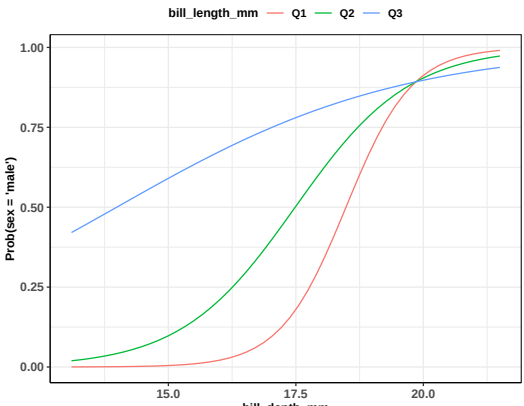
$$\text{logit}\,\hat{\text{P}}(sex = \text{'male'}|\text{BD}, \text{BL}) = -130.61 + 6.69\text{BD} + 2.58\text{BL} - 0.13\text{BD} \times \text{BL}$$

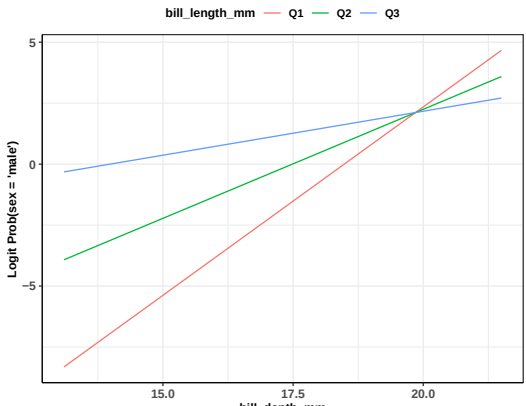
$$\Rightarrow \hat{\text{P}}(sex = \text{'male'}|\text{BD}, \text{BL}) = \frac{1}{1 + e^{130.61 - 6.69\text{BD} - 2.58\text{BL} + 0.13\text{BD} \times \text{BL}}}$$

Nous pouvons mieux apprécier le modèle et l'interpréter plus facilement en considérant les quelques graphiques suivants.

» Effet de bill_depth_mm pour différentes valeurs de bill_length_mm

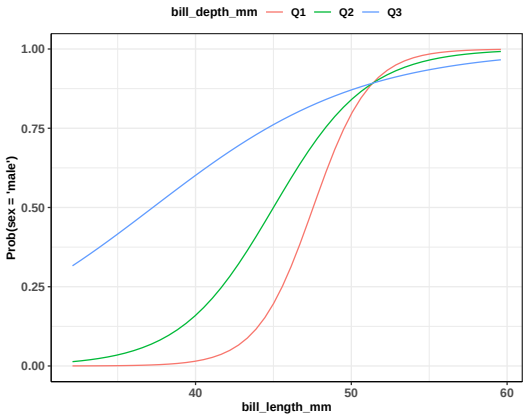
```
plot_predictions(mbdbl, condition = list("bill_depth_mm", bill_length_mm = "quartile"), vcov = FALSE) +
  labs(y = "Prob(sex = 'male')")
plot_predictions(mbdbl, condition = list("bill_depth_mm", bill_length_mm = "quartile"), vcov = FALSE, type = 'link') +
  labs(y = "Logit Prob(sex = 'male')")
```

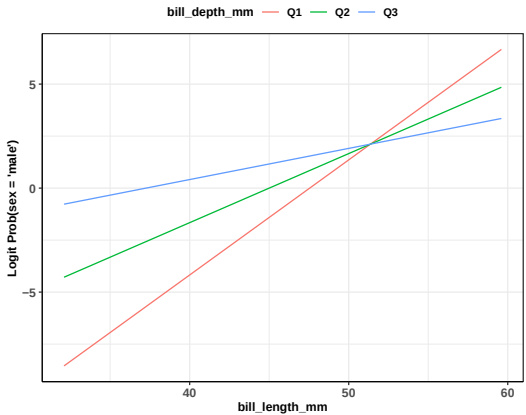




» Effet de bill_length_mm pour différentes valeurs de bill_depth_mm

```
plot_predictions(mbdbl, condition = list("bill_length_mm", bill_depth_mm = "quartile"), vcov = FALSE) +
 labs(y = "Prob(sex = 'male')")
plot_predictions(mbdbl, condition = list("bill_length_mm", bill_depth_mm = "quartile"), vcov = FALSE, type = 'link') +
  labs(y = "Logit Prob(sex = 'male')")
```





Comme conclusion, nous pouvons dire que, globalement,

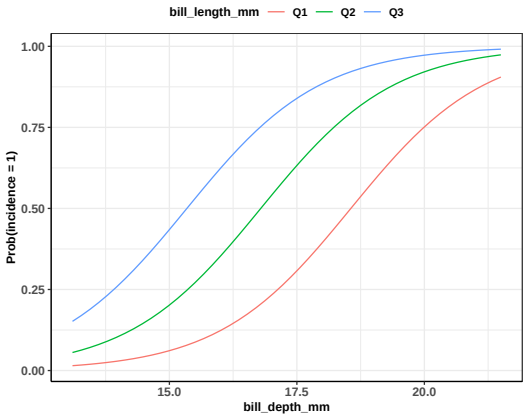
- plus l'épaisseur du bec d'un manchot est grande, plus il y a de chance qu'il soit un mâle. Ceci est d'autant plus notable chez les individus dont la taille de bec (BL) est relativement petite. Plus précisément, on estime que la cote d'être un mâle se multiplie par $exp(6.69 - 0.13 \times \text{BL})$ chaque fois que l'épaisseur du bec augmente d'un millimètre. Ce qui, à titre d'exemple, correspond à une augmentation d'environ 344 % pour les manchots avec un bec de taille 40 mm, et seulement d'environ 21 % pour ceux avec un bec de taille 50 mm.

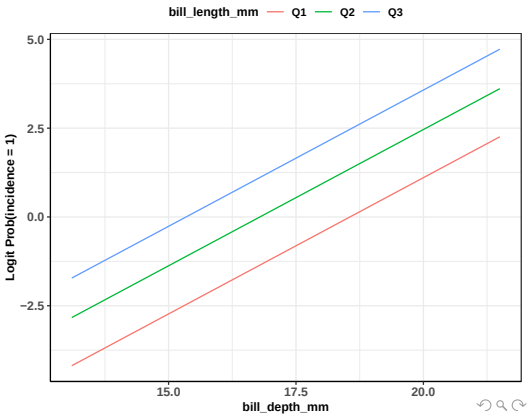
- plus la taille du bec d'un manchot est grande, plus il y a de chance qu'il soit un mâle. Ceci est d'autant plus notable chez les individus dont l'épaisseur du bec (BD) est relativement petite. Plus précisément, on estime que la cote d'être un mâle se multiplie par $\exp(2.58 - 0.13 \times \text{BD})$ chaque fois que la taille du bec augmente d'un millimètre. Ce qui, à titre d'exemple, correspond à une augmentation d'environ 88 % pour les manchots avec un bec d'épaisseur 15 mm, et seulement d'environ 12 % pour ceux avec un bec d'épaisseur 19 mm.

Remarque

On peut bien sûr ajuster un modèle sans interaction même si cette dernière est significative. L'absence d'interaction se traduit graphiquement par des droites parallèles lorsqu'on examine les logits et par des courbes identiques, mais décalées horizontalement, lorsqu'on examine les probabilités.

```
mbdblsi <- glm(sex ~ bill_depth_mm + bill_length_mm, family = binomial, data = penguins)
```





Régression logistique binaire multiple

Prédicteurs catégoriels

Nous disposons d'une variable binaire $Y \in \{1 \equiv \text{Succès}, 2 \equiv \text{Échec}\}$ et d'une varaible X prenant ces valeurs dans $\{1, 2, \ldots, I\}$ où la valeur i correspond à une catégorie donnée (disons, Gri). Nous cherchons à étudier le lien entre Y et X.

Sous forme groupée, les données qui nous intéressent se présenter comme

X	Y	Freq
1	1	n_{11}
1	2	n_{12}
⋮	⋮	⋮
I	1	n_{I1}
I	2	n_{I2}

ou

X|Y	$1 \equiv$ Succès	$2 \equiv$ Échec
$1 \equiv$ Gr1	n_{11}	n_{12}
⋮	⋮	⋮
$I \equiv$ GrI	n_{I1}	n_{I2}

Notations

$$p(x) = P(Y = 1|X = x), \text{ et } o(x) = \frac{p(x)}{1 - p(x)}, \text{ pour } x = 1, 2, \ldots, I.$$

Par la suite nous allons aussi utilisé les raccourcis suivants

$$p_i \equiv p(i), \text{ et } o_i \equiv o(i), \text{ pour } i = 1, \ldots, I.$$

Hypothèses

(1) $Y|X \sim Ber(p(X))$,
(2) $logit(p(X)) = \beta_0 + \beta_1 I(X = 1) + \ldots + \beta_{I-1} I(X = I - 1)$.

Cette dernier équation est équivalente à

$$\log(o_i) = \beta_0 + \beta_i, i = 1, \ldots, I, \text{ avec } \beta_I = 0.$$

Ces équations supposent que "I" est le niveau/groupe de référence.

Il est facile de voir que l'équation du modèle implique que

$$e^{\beta_0} = o(I) = \frac{p_I}{1-p_I}, \text{ et } e^{\beta_i} = \frac{o(i)}{o(I)} = \frac{p_i/(1-p_i)}{p_I/(1-p_I)}, \; i = 1, \ldots, I.$$

Ainsi, pour $i = 1, \ldots, I-1$, affirmer que $\beta_i = 0$ est équivalent à dire $p_i = p_I$, càd que la probabilité de succès dans le groupe i (défini par la modalité i de X) est la même que celle dans le groupe de référence I.

En conséquence, dire que $\beta_i = 0$, $\forall i = 1, \ldots, I-1$ est équivalent à dire que les probabilités de succès $p_i = P(Y = 1|X = i)$ sont identiques quel que soit i. Cela revient à dire que les I groupes, définis par les modalités de X, sont homogènes au regard de Y.

Conclusion : Pour effectuer un *test d'homogénéité (ou d'indépendance)* dans un tableau $I \times 2$, croisant les I modalités de X et les deux modalités de Y, il suffit de tester

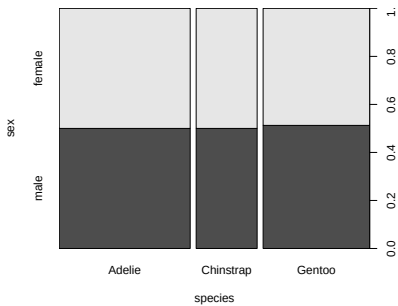
$$H_0 : \beta_1 = \cdots \beta_{I-1} = 0,$$

où β_i sont les coefficients du modèle logistique $Y \sim X$.

À titre d'exemple, considérons la régression logistique avec $Y = \mathrm{I}(sex = \text{'male'})$ comme variable dépendante, $X = species$ comme variable explicative. Il ne s'agit pas ici de vouloir prédire le sexe, mais de comprendre s'il y a ou non une différence entre les espèces des manchots quant à la proportion des mâles et des femelles.

```
xtabs(~ species + sex, data = penguins) |> print() |> proportions("species")
plot(sex ~ species, data = penguins)
```

species/sex	female	male	Total
Adelie	50.0% (73)	50.0% (73)	100.0% (146)
Chinstrap	50.0% (34)	50.0% (34)	100.0% (68)
Gentoo	48.7% (58)	51.3% (61)	100.0% (119)



Cette analyse descriptive met en évidence l'homogénéité des espèces quant à la répartition mâle-femelle. Voyons ce que la régression logistique peut révéler.

L'équation théorique du modèle logistique est

$$\text{logit}\, P(sex = \text{'male'}|species) = \beta_0 + \beta_1 I(species = \text{'Chinstrap'}) + \beta_2 I(species = \text{'Gentoo'})$$

On estime le modèle et on calcule l'exponentielle des coefficients ainsi que leurs intervalles de confiance.

```
msp <- glm(sex ~ species, family = binomial, data = penguins)
```

						IC.exp.coef	
	Estimate	Std. Error	z value	Pr(>|z|)	exp.coef	2.5 %	97.5 %
(Intercept)	0.00000	0.16552	0.00000	1.00000	1.0000	0.72243	1.3842
speciesChinstrap	0.00000	0.29363	0.00000	1.00000	1.0000	0.56163	1.7805
speciesGentoo	0.05043	0.24705	0.20413	0.83825	1.0517	0.64786	1.7084

Comme il est de coutume, la colonne "Pr(>|z|)" contient les p−valeurs asymptotiques des tests de la nullité des coefficients. Ces tests ont ici une interprétation bien particulière.

En effet

$$\beta_0 = 0 \iff P(sex = \text{'male'}|Adelie) = P(sex = \text{'female'}|Adelie)$$
$$\beta_1 = 0 \iff P(sex = \text{'male'}|Chinstrap) = P(sex = \text{'male'}|Adelie)$$
$$\beta_2 = 0 \iff P(sex = \text{'male'}|Gentoo) = P(sex = \text{'male'}|Adelie)$$

Au seuil de 5%, nous ne pouvons pas rejeter ces hypothèses (prises séparément) $\rightarrow$ il n' y a pas de différence significative entre les proportions comparées.

Soit $H_0 : \ \beta_1 = \beta_2 = 0$ vs $H_1 : \ \beta_1 \neq 0$ ou $\beta_2 \neq 0$. Cette hypothèse signifie que la proportion des mâles est la même chez les trois espèces.

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

Model:
sex ~ species
        Df Deviance AIC Rao score Pr(>Chi)
<none>          462 468
species  2      462 464    0.0486     0.98
```

Comme attendu, l'hypothèse d'homogénéité n'est pas à rejeter. Notez qu'on obtient le même résultat si l'on effectue un test d'indépendance classique.

```
summary(tb)
```

Nous disposons d'une variable binaire $Y \in \{1 \equiv \text{Succès}, 2 \equiv \text{Échec}\}$ et deux varaible catégoriels $X \in \{1, 2, \ldots, I\}$ et $Z \in \{1, 2, \ldots, K\}$. Nous cherchons à étudier le lien entre Y et (X, Z).

Nous reprenons les mêmes notations et hypothèses que celles utilisées précédemment. En particulier,

$$p(x, z) = P(Y = 1 | X = x, Z = z), \quad p_{ik} = p(i, k) \text{ et } o_{ik} = \frac{p_{ik}}{1 - p_{ik}}.$$

Le modèle logistique $Y \sim X + Y + X \times Z$ s'écrit comme

$$\text{logit}(p(X, Z)) = \beta_0 + \sum_{i=1}^{I-1} \beta_i^X I(X = i) + \sum_{k=1}^{K-1} \beta_k^Z I(Z = k) + \sum_{i=1}^{I-1} \sum_{k=1}^{K-1} \beta_{ik}^{XZ} I(X = i) I(Z = k)$$

$$\Leftrightarrow \log(o_{ik}) = \beta_0 + \beta_i^X + \beta_k^Z + \beta_{ik}^{XZ}, \; i = 1, \ldots, I, k = 1, \ldots, K, \text{ avec}$$

$$\beta_I^X = \beta_K^Z = \beta_{Ik}^{XZ} = \beta_{iK}^{XZ} = 0, \forall i, k.$$

Admettons que le modèle énoncé ci-dessus soit correct. En analysant attentivement l'équation précédente, en suivant le même raisonnement que pour la régression de Poisson, on peut établir une correspondance entre la nullité des coefficients et le type d'association liant les variables étudiées.

Le tableau suivant résume tous les scénarios possibles et fait le parallèle avec la régression Poisson.

Description	Rég. Poisson	Rég. Logistique	
ass. homo.	$\text{Freq} \sim X \times Y + Y \times Z + X \times Z$	$Y \sim X + Z$	$\beta_{ik}^{XZ} = 0$
$Y \perp\!\!\!\perp Z \mid X$	$\text{Freq} \sim X \times Y + X \times Z$	$Y \sim X$	$\beta_{ik}^{XZ} = \beta_{k}^{Z} = 0$
$Y \perp\!\!\!\perp X \mid Z$	$\text{Freq} \sim Y \times Z + X \times Z$	$Y \sim Z$	$\beta_{ik}^{XZ} = \beta_{i}^{X} = 0$
$Y \perp\!\!\!\perp (X, Z)$	$\text{Freq} \sim Y + X \times Z$	$Y \sim 1$	$\beta_{ik}^{XZ} = \beta_{i}^{X} = \beta_{k}^{Z} = 0$

Notez que, à la différence de la régression de Poisson, avec la régression logistique on ne peut étudier que l'association entre Y (la réponse) et les variables explicatives du modèle (ici X et Z). Mais, on ne peut rien dire sur l'association entre ces derniers.

À titre d'exemple, avec les données `penguins`, nous pouvons tester l'indépendance partielle entrer sex et $(species, year)$, en utilisant le modèle logistique, de la manière suivante

```
anova(glm(sex ~ 1, family = binomial, data = penguins),
      glm(sex ~ species * year, family = binomial, data = penguins), test = "Rao")

Analysis of Deviance Table

Model 1: sex ~ 1
Model 2: sex ~ species * year
  Resid. Df Resid. Dev Df Deviance    Rao Pr(>Chi)
1       332        462
2       324        462  8   0.0499 0.0499        1
```

On obtient le même résultat avec la régression Poisson

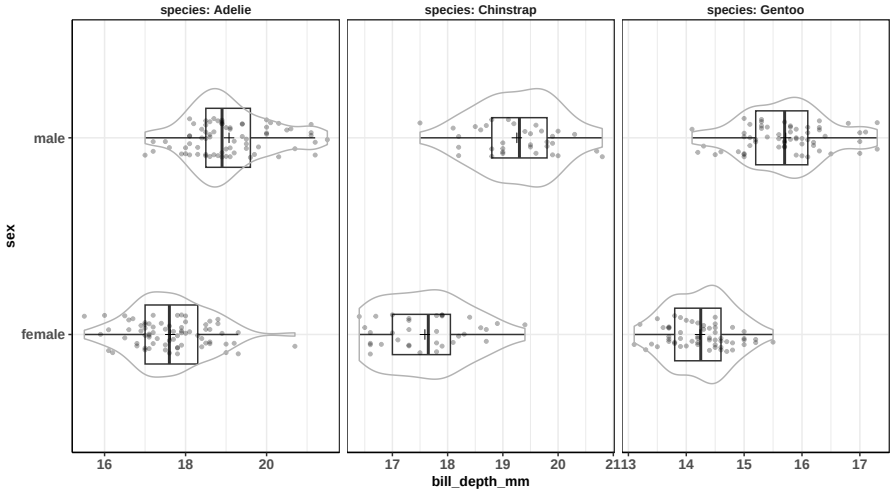
```
dftb <- xtabs(~ species + sex + year, data = penguins) |> data.frame()
anova(glm(Freq ~ sex + species * year, family = poisson, data = dftb),
      glm(Freq ~ sex * species * year, family = poisson, data = dftb), test = "Rao")
```

Régression logistique binaire multiple

Prédicteurs catégoriels et continus

Dans un modèle logistique, on peut bien évidemment combiner des variables explicatives continues et discrètes. Nous allons illustré cela en considérant la variable *sex* comme réponse, et bill_depth_mm et *species* comme prédicteurs.

```
ggplot(penguins, aes(x = bill_depth_mm, y = sex)) + pls$layers + facet_grid(cols = vars(species), lab = "label_both", scales = "free")
```



Avec les abréviations BD = bill_depth_mm, SP = species, 'Ad' = 'Adelie', 'Ch' = 'Chinstrap' et 'Ge' = 'Gentoo', nous allons ajuster aux données le modèle

$$\text{logit P}(sex = \text{'male'}|\text{BD}, \text{SP}) = \beta_0 + \beta_1 \text{BD} + \beta_2 \text{I}(\text{SP} = \text{'Ch'}) + \beta_3 \text{I}(\text{SP} = \text{'Ge'}) + \beta_4 \text{BD} \times \text{I}(\text{SP} = \text{'Ch'}) + \beta_5 \text{BD} \times \text{I}(species = \text{'Ge'}).$$

Par espèce ('Ad', 'Ch', 'Ge'), cela revient à écrire que

$$\begin{cases} \text{logit P}(sex = \text{'male'}|\text{BD}, \text{'Ad'}) & = \beta_0 + \beta_1 \text{BD} \\ \text{logit P}(sex = \text{'male'}|\text{BD}, \text{'Ch'}) & = (\beta_0 + \beta_2) + (\beta_1 + \beta_4)\text{BD} \\ \text{logit P}(sex = \text{'male'}|\text{BD}, \text{'Ge'}) & = (\beta_0 + \beta_3) + (\beta_1 + \beta_5)\text{BD} \end{cases}$$

L'interaction entre sex et BD est représentée ici par le couple (β_4, β_5). Suivant le même logique qu'auparavant, nous pouvons voir que

(a) l'effet de BD peut être décrit par

$$\frac{\text{o}(sex = \text{'male'}|\text{BD} + 1, \text{SP})}{\text{o}(sex = \text{'male'}|\text{BD}, \text{SP})} = \exp\Big(\beta_1 + \beta_4 \text{I}(\text{SP} = \text{'Ch'}) + \beta_5 \text{I}(\text{SP} = \text{'Ge'})\Big).$$

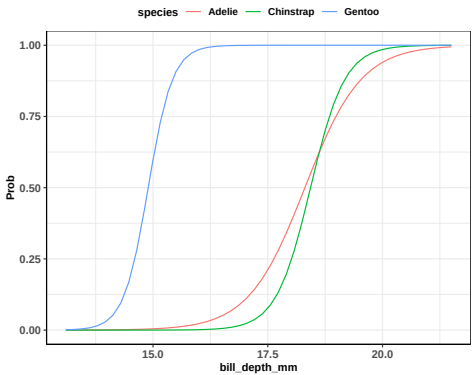
(b) l'effet de SP peut être décrit par

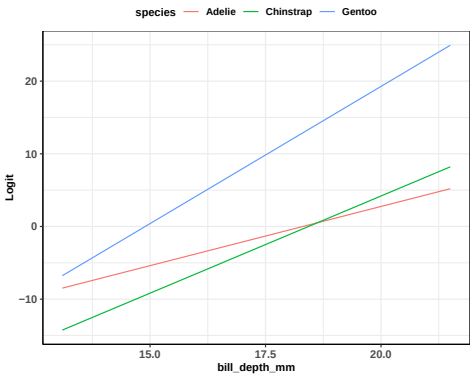
$$\frac{\text{o}(sex = \text{'male'}|\text{BD}, \text{SP})}{\text{o}(sex = \text{'male'}|\text{BD}, \text{'Ad'})} = \exp\Big((\beta_2 + \beta_4 \text{BP})\text{I}(\text{SP} = \text{'Ch'}) + (\beta_3 + \beta_5 \text{BP})\text{I}(\text{SP} = \text{'Ge'})\Big).$$

```r
mbdsp <- glm(sex ~ bill_depth_mm * species, family = binomial, data = penguins)
summary(mbdsp) |> coef()
```

```
                                Estimate Std. Error z value   Pr(>|z|)
(Intercept)                   -29.8243    5.11610 -5.8295 5.5594e-09
bill_depth_mm                   1.6290    0.27943  5.8296 5.5563e-09
speciesChinstrap              -19.4716   12.74869 -1.5273 1.2668e-01
speciesGentoo                 -26.4110   11.63370 -2.2702 2.3194e-02
bill_depth_mm:speciesChinstrap  1.0456    0.69180  1.5114 1.3068e-01
bill_depth_mm:speciesGentoo     2.1465    0.75603  2.8392 4.5226e-03
```

```r
plot_predictions(mbdsp, condition = list("bill_depth_mm", "species"), vcov = FALSE) + labs(y = "Prob")
plot_predictions(mbdsp, condition = list("bill_depth_mm", "species"), vcov = FALSE, type = 'link') + labs(y = "Logit")
```





$\rightarrow$ En bref, nous constatons que plus l'épaisseur du bec d'un manchot est grande, plus il y a de chance qu'il soit un mâle. Cela est d'autant plus remarquable chez les "Gentoo" où le rapport des cotes associé à une augmentation de l'épaisseur du bec d'un millimètre est estimé à $\exp(1.63 + 2.15) = 34.82$ alors qu'il est de seulement $\exp(1.63) = 5.10$ chez les "Adelie" et de $\exp(1.63 + 1.05) = 14.58$ chez les "Chinstrap".

Cela semble indiquer une d'interaction entre les variables $\mathrm{bill\_depth\_mm(BD)}$ et $\mathit{species}(\mathrm{SP})$. La question à se poser est de savoir si cette interaction est significative ou pas ? Pour répondre à cette question, il faudra tester la nullité simultanée de β_4 et β_5, càd tester

$$\mathrm{H}_0 : \beta_4 = \beta_5 = 0 \ \text{ vs } \ \mathrm{H}_1 : \beta_4 \neq 0 \text{ ou } \beta_5 \neq 0$$

Pour réaliser ce test, nous allons utiliser la test de rapports de vraisemblance.

```
drop1(mbdsp, test = "LRT")$"Pr(>Chi)"[2]
```

```
[1] 0.0031393
```

$\rightarrow$ à 5%, l'interaction est significative.

Régression logistique binaire multiple

Sélection de modèles et prédiction

Comme indiqué précédemment, notre objectif final ici est de construire un modèle qui nous permettrait de prédire, le plus précisément possible, le sexe d'un manchot en utilisant les données à notre disposition. Pour ce faire, nous allons tenter de construire un modèle prédictif simple et performant.

Dans la lignée de ce qui a été décrit dans le chapitre sur la régression de Poisson, nous pouvons effectuer la sélection de à l'aide de différentes méthodes. Nous nous limiterons ici aux deux méthodes décrites dans le code suivant.

```
# Exhaustive search using AIC
mexhAic <- glmulti(sex ~ (species + bill_length_mm + bill_depth_mm + flipper_length_mm +
  body_mass_g)^2, family = binomial, data = penguins, crit = aic, marginality = TRUE,
  plotty = FALSE, report = FALSE, confsetsize = 5)

# Exhaustive search using BIC
mexhBic <- glmulti(sex ~ (species + bill_length_mm + bill_depth_mm + flipper_length_mm +
  body_mass_g)^2, family = binomial, data = penguins, crit = bic, marginality = TRUE,
  plotty = FALSE, report = FALSE, confsetsize = 5)
```

model	aic	df
sex ~ 1 + species + bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g + flipper_length_mm:bill_length_mm + flipper_length_mm:bill_depth_mm + body_mass_g:bill_length_mm + body_mass_g:flipper_length_mm	130.07	11
sex ~ 1 + species + bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g + flipper_length_mm:bill_length_mm + body_mass_g:bill_length_mm + body_mass_g:bill_depth_mm + species:bill_depth_mm	130.20	12
sex ~ 1 + species + bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g + flipper_length_mm:bill_length_mm + flipper_length_mm:bill_depth_mm + body_mass_g:bill_length_mm + body_mass_g:bill_depth_mm + body_mass_g:flipper_length_mm	130.78	12
sex ~ 1 + species + bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g + flipper_length_mm:bill_length_mm + body_mass_g:bill_length_mm + body_mass_g:bill_depth_mm + body_mass_g:flipper_length_mm + species:bill_depth_mm	131.11	13
sex ~ 1 + species + bill_length_mm + bill_depth_mm + flipper_length_mm + body_mass_g + bill_depth_mm:bill_length_mm + flipper_length_mm:bill_length_mm + flipper_length_mm:bill_depth_mm + body_mass_g:bill_length_mm + body_mass_g:flipper_length_mm	131.76	12

model	bic	df
sex ~ 1 + species + bill_length_mm + bill_depth_mm + body_mass_g + bill_depth_mm:bill_length_mm	161.85	7
sex ~ 1 + species + bill_length_mm + bill_depth_mm + body_mass_g	162.15	6
sex ~ 1 + species + bill_length_mm + bill_depth_mm + body_mass_g + body_mass_g:bill_length_mm	164.15	7
sex ~ 1 + species + bill_length_mm + bill_depth_mm + body_mass_g + bill_depth_mm:bill_length_mm + body_mass_g:bill_depth_mm	166.79	8
sex ~ 1 + species + bill_length_mm + bill_depth_mm + body_mass_g + species:body_mass_g	167.14	8

```r
mexhAic1 <- mexhAic@objects[[1]]
mexhBic2 <- mexhBic@objects[[2]]

rocmexhAic1 <- roc(response = mexhAic1$data$sex, predictor = fitted(mexhAic1))
rocmexhBic2 <- roc(response = mexhBic2$data$sex, predictor = fitted(mexhBic2))

# Specificity, Sensitivity and Accuracy using the best cut-off
list(rocmexhAic1, rocmexhBic2) |> sapply(FUN = coords, x = "best",
  ret = c("threshold", "specificity", "sensitivity", "accuracy")) |> t()
```

```
     threshold specificity sensitivity accuracy
[1,] 0.56915   0.93939     0.92262     0.93093
[2,] 0.62452   0.95758     0.90476     0.93093
```

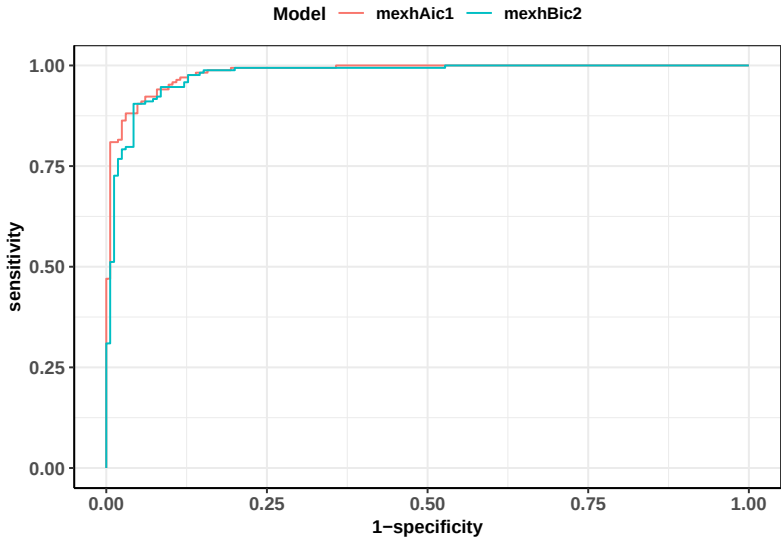
```r
# AUC
list(rocmexhAic1, rocmexhBic2) |> sapply(FUN = auc) |> round(3)
```

```
[1] 0.984 0.977
```

```r
# cvAUC
fitControl <- trainControl(method = "repeatedcv", number = 10, repeats = 100, classProbs = TRUE, summaryFunction = twoClassSummary)
c(train(formula(mexhAic1), data = mexhAic1$data, family = binomial, method = "glm", trControl = fitControl)$results$ROC,
train(formula(mexhBic2), data = mexhBic2$data, family = binomial, method = "glm", trControl = fitControl)$results$ROC) |> round(3)
```

```
[1] 0.975 0.974
```

```
list("mexhAic1" = rocmexhAic1, "mexhBic2" = rocmexhBic2) |> ggroc(legacy.axes = TRUE) + labs(color = "Model")
```



À pouvoir prédictif presque équivalent, nous nous en tiendrons ici à la simplicité et choisissons le modèle $mexhBic2$: $sex \sim species + bill\_length\_mm + bill\_depth\_mm + body\_mass\_g$. Le modèle choisi avec l'AIC est beaucoup plus complexe, pour un gain de performance, en termes prédictifs, insignifiant.

Voici d'autres aspects intéressants concernant le modèle $mexhBic2$:

» L'absence d'effets d'interaction facilite son interprétation.

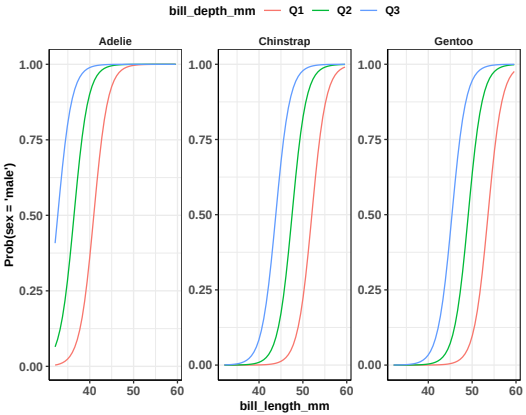
» Les résidus ne révèle rien d'alarmant; voir `diagnost(mexhBic2)`.

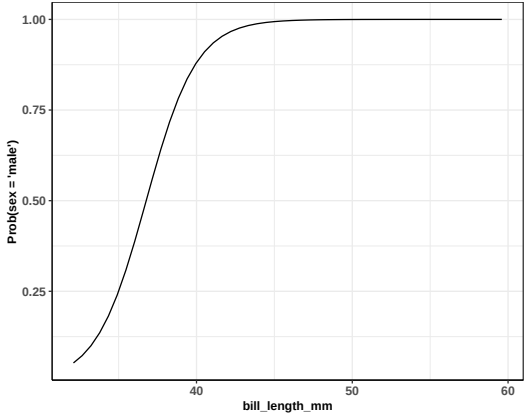
» Tous les termes du modèle sont fortement significatifs; voir `summary(mexhBic2)$coef`.

L'absence d'interaction nous permet de visualiser l'effet de chaque prédicteur en fixant les valeurs des autres sans altérer notre perception. Ceci est illustré dans les graphiques suivants.

REMARQUE Par défaut, `plot_predictions()` fixe les prédicteurs du modèle non inclus dans " `condition` " à leurs moyennes (pour les variables numériques) ou modes (pour les factors) □

```
plot_predictions(mexhBic2, condition = list("bill_length_mm", bill_depth_mm = "quartile","species"), vcov = FALSE) +
  labs(y = "Prob(sex = 'male')")
plot_predictions(mexhBic2, condition = "bill_length_mm", vcov = FALSE) + labs(y = "Prob(sex = 'male')")
```





Enfin, voici les prédictions pour le sexe des observations manquantes.

```
ifelse(predictions(mexhBic2, newdata = penguinsNA)$estimate > 0.62452, "male", "female")
```

```
 [1] NA       "female" "male"   "female" "female" "female" "female" "female"
 [9] "female" "female" NA
```