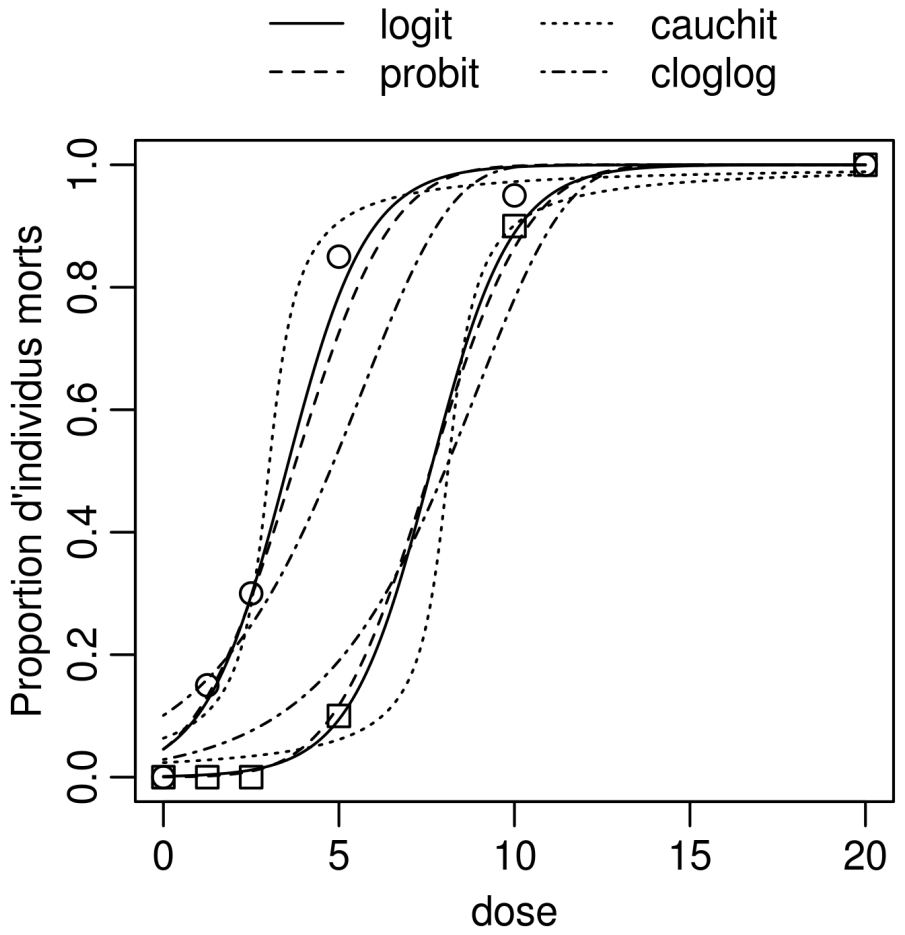
GLM : Generalized Linear Models



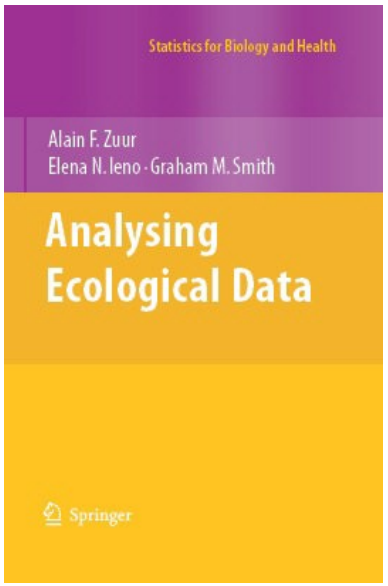
G. San Martin

gilles.sanmartin@gmail.com
Centre Wallon de Recherche Agronomique

cra-w

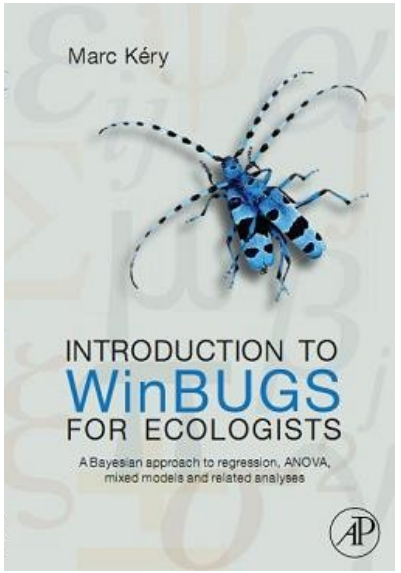
Quelques livres

Formation principalement basée sur 4 livres.
Tous ont une approche unifiée "moderne" (GLM) et certains font le lien avec les stats classiques

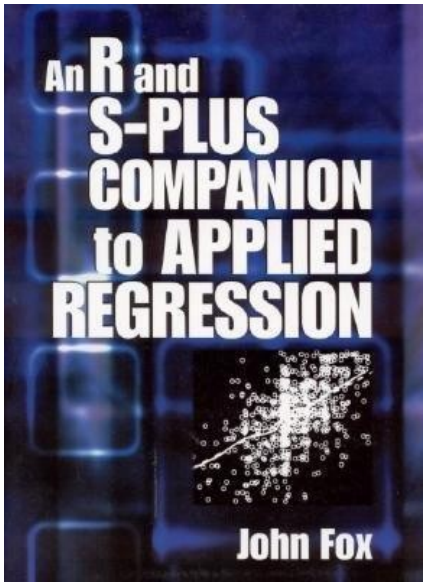


Zuur et al.
Le plus appliqué.
Pour être opérationnel le plus vite possible

Gelman & Hill
Le plus détaillé tout en restant très accessible



Kery
Analyse classique et Bayésienne de Jeux de données simulés



Fox
Le plus simple aborde des problèmes rarement abordés

Objectifs

Qu'est-ce qu'un GLM, à quoi çà sert ?

Illustrer :

Exemples des principaux types de GLM

La plupart des tests statistiques classiques sont des cas particuliers de GLM

Insister sur :

Comment interpréter les sorties du logiciel ?

Comment en faire une représentation graphique ?

Quelles sont les conditions d'application, comment les vérifier et comment solutionner les problèmes ?

GLMs : qu'est-ce que c'est ?

Régression établissant le lien entre une variable à expliquer/prédire et une ou plusieurs variables descriptives/explicatives

Paramètres à estimer

Relation linéaire additive

$$y = \alpha + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \epsilon$$

Variable "dépendante"
a expliquer :

données
continues
binaires
comptage
%

Variables "explicatives"
Quantitatives
Qualitatives
Interactions

Résidus = Erreur
parties de la variabilité
que l'on ne peut pas
expliquer avec ces variables

Distrib Normale
Distrib de Poisson
Distrib Binomiale
etc...

GLMs : qu'est-ce que c'est ?

(General) Linear Model

y = variable continue à distribution à peu près normale

Résidus : distribution Normale

Méthode d'estimation = Sum of Squares

R : lm() - SAS : PROC GLM

Generalized Linear Model

y = variable continue ou de comptage ou binaire ou %,...

Résidus : distribution Normale ou Poisson ou Binomiale,...

Fonction de lien

Méthode d'estimation = Maximum Likelihood

R : glm() - SAS : PROC GENMOD

Programme

Part 1 : (General) Linear Model (LM)

On va s'intéresser principalement aux variables explicatives
Y sera toujours une variable quantitative continue approximativement normale

1 X quantitatif = régression linéaire simple
1 X qualitatif à 2 niveaux = test de student
1 X qualitatif à n niveaux = ANOVA
Comparaisons multiples
Plusieurs X quantitatifs = régression multiple
Plusieurs X quantitatifs ou qualitatifs = ANCOVA
Interactions
Relations non linéaires

Programme

Part 2 : Generalized Linear Model (GLM)

La partie concernant les variables explicatives (x) change peu.
2 changements : distributions des résidus - fonctions de lien

On choisi la distribution des résidus a priori sur base du type de données. On vérifiera ensuite sur base des résultats si cette première idée est bonne ou pas ...

Y = données de comptage
Tables de contingence
--> distribution de Poisson

Y binaire ou % (nbre de succès/nbre d'essais)
--> distribution binomiale

Autres données continues (y compris autres %)
--> distribution gaussienne

Part 1 : General Linear Model

Régression linéaire simple : 1 x quantitatif

Il s'agit ici de trouver la meilleure droite passant par un nuage de points

Exemple : relation entre les doses de fertilisants et la production de tomates

Concepts à assimiler :

Pente, intercept, résidus

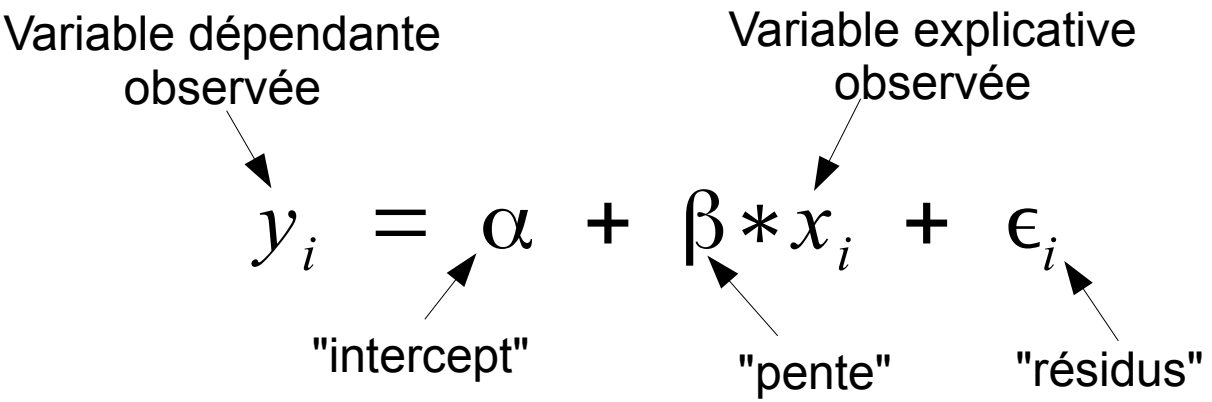
Interprétation géométrique des paramètres

R^2, % de variance expliquée

Valeurs prédites

Régression linéaire simple : 1 x quantitatif

Représentation algébrique du modèle :



Intercept :
valeur prédite de y quand x = 0

Pente (= "Slope") :
de combien augmente y quand x augmente de une unité ?

Résidus :
différence entre les valeurs observées et les valeurs prédites

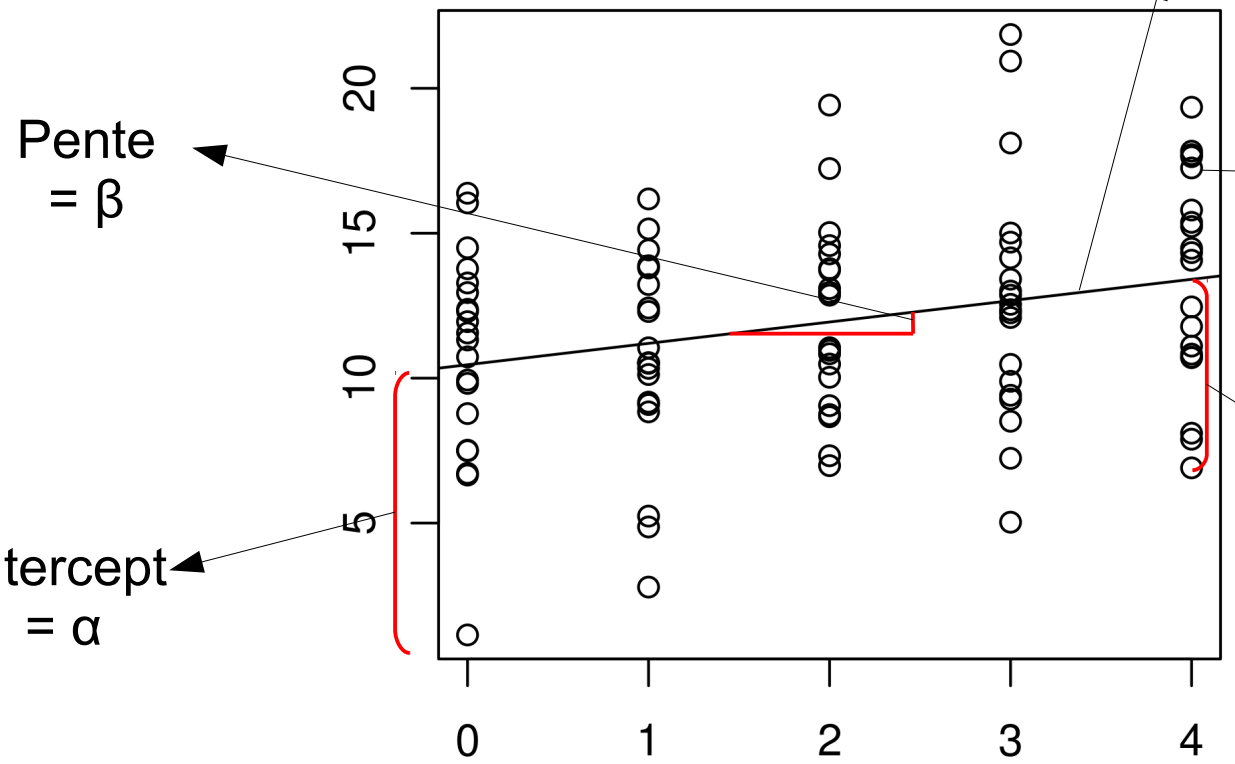
Régression linéaire simple : 1 x quantitatif

Représentation géométrique du modèle :

$$y_i = \alpha + \beta * x_i + \epsilon_i$$

Y
= poids des tomates

Droite = ŷ
Valeurs prédites
par le modèle



X
= dose fertilisant

La droite optimale est celle qui minimise les résidus
(ie la somme des carrés des résidus)

Régression linéaire simple : 1 x quantitatif

Représentation algébrique du modèle :

"Valeurs prédites par le modèle"

$$\hat{y}_i = \alpha + \beta * x_i$$

Les résidus sont la différence entre les valeurs observées et les valeurs prédites

$$\epsilon_i = y_i - \hat{y}_i$$

Les résidus suivent une distribution Normale de moyenne 0 et de variance sigma²

$$\epsilon \sim Normale\left(0, \sigma^2\right)$$

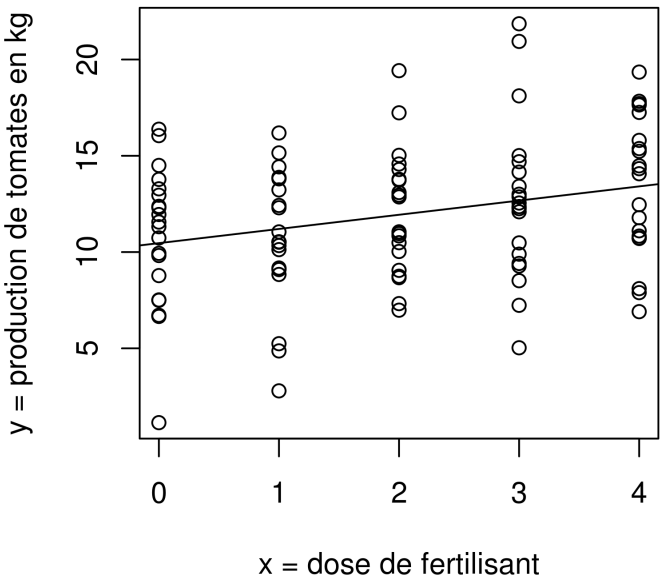
3 paramètres doivent être estimés : l'intercept, la pente et la variance des résidus

Régression linéaire simple : 1 x quantitatif

Exemple : production tomates ~ dose fertilisant

On génère des données (n=100) pour avoir un intercept (alpha) de 10 kg, une pente (beta) de 0.75 kg et une variance des résidus (sigmasq) de 16 kg².
On a 5 doses de fertilisant (0-4) et 20 observations par dose.

```
> alpha <- 10
> beta <- 0.75
> sigmasq <- 16
>
> n <- 100
> x <- rep(0:4, each = n/5)
>
> set.seed(1)
> y <- alpha + beta * x +
    rnorm(n = n, mean = 0, sd = sqrt(sigmasq))

> # autre manière de faire strictement identique :
> set.seed(1)
> y <- rnorm(n = n, mean = (alpha + beta * x) ,
    sd = sqrt(sigmasq))
```



Régression linéaire simple : 1 x quantitatif

```
> mod <- lm(y ~ x)
> summary(mod)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-9.3209 -2.4158  0.0329  2.3406  9.1842

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4621     0.6254  16.727  < 2e-16 ***
x             0.7367     0.2553   2.885  0.00481 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.611 on 98 degrees of freedom
Multiple R-squared:  0.0783,  Adjusted R-squared:  0.0689
F-statistic: 8.325 on 1 and 98 DF,  p-value: 0.004808
```

Estimation du modèle

Résumé du modèle

Intercept (alpha) estimé

Pente (beta) estimée

erreur standard des résidus

Régression linéaire simple : 1 x quantitatif

Interprétation

Quand on ne met aucun fertilisant (x=0) on estime que la production moyenne de tomates est de 10.4621 kg

Quand la dose de fertilisant augmente d'une unité, la production de tomates augmente de 0.7367 kg (dans la limite des doses testées)

On peut prédire la production de tomate en fonction de la dose de fertilisant. Par exemple on estime que pour une dose de 1.42 unités de fertilisant, on aura en moyenne une production de 10.4621 + 0.7367 * 1.42 = 11.51 kg de tomates

Autour de ces valeurs prédites, les résidus ont un écart-type estimé (erreur standard) de 3.611 kg
Attention il ne s'agit PAS de l'erreur standard des valeurs prédites !

```
> mod <- lm(y ~ x)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4621     0.6254  16.727  < 2e-16 ***
x             0.7367     0.2553   2.885  0.00481 **
(...)
Residual standard error: 3.611 on 98 degrees of freedom
Multiple R-squared:  0.0783,  Adjusted R-squared:  0.0689
F-statistic: 8.325 on 1 and 98 DF,  p-value: 0.004808
```

Régression linéaire simple : 1 x quantitatif

Inférence pour les (General) Linear Models

On teste toujours par défaut l'hypothèse nulle que chaque coefficient est = 0

Deux méthodes paramétriques principales

Dans ce cas-ci elles sont strictement équivalentes mais ce ne sera pas toujours le cas !

Méthode 1 : "Test de Wald"

coefficient / erreur standard suit une loi de Student
à n-k degrés de liberté (k = nombres de paramètres)

```
> summary(mod)
(...)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4621     0.6254  16.727  < 2e-16 ***
x             0.7367     0.2553   2.885  0.00481 **
```

Estimate +- 2*Std.Error --> intervales de confiance approximatifs
utiliser les quantiles de la loi de student pour avoir des valeurs exactes
ou `confint(mod)`

Régression linéaire simple : 1 x quantitatif

Méthode 2 : Comparaison de modèles emboîtés

Pour tester si la pente est significativement différente de 0, on compare un modèle complet y ~ a + bx avec un modèle avec une pente nulle y ~a.
La statistique utilisée ici est liée au carré des résidus (RSS : Residuals Sum of Squares) et suit une distribution de Fisher (F)

```
> mod0 <- lm(y ~ 1)
> mod1 <- lm(y ~ x)
> anova(mod0, mod1)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     99 1386.4
2     98 1277.9  1    108.56 8.3253 0.004808 **
```

Modèle avec juste 1 intercept

Modèle avec intercept (implicite) et pente

$$F = \frac{\frac{RSS_0 - RSS_1}{df_0 - df_1}}{\frac{RSS_1}{df_1}}$$

<u>Attention !!</u> : on peut aussi procéder comme suit. Ici les résultats sont identiques. Mais dans de nombreux cas cette méthode ne donne pas le résultat escompté !!!! --> très déconseillée

```
> anova(mod1)
          Df  Sum Sq Mean Sq F value   Pr(>F)
x          1  108.56  108.56  8.3253 0.004808 **
Residuals 98 1277.88   13.04
```

Régression linéaire simple : 1 x quantitatif

Extraire des informations du modèle

R permet de récupérer n'importe quelle valeur calculée afin de la manipuler

```
> # coefficients
> coef(mod)
(Intercept)           x
 10.4620597   0.7367449

> # Intervalles de confiance
> confint(mod)
                2.5 %    97.5 %
(Intercept) 9.2208763 11.703243
x           0.2300339  1.243456

> y[1:5]
[1]  7.494185 10.734573  6.657486 16.381123 11.318031

> # Valeurs prédites pour chaque x observé
> fitted(mod)[1:5]
       1        2        3        4        5
10.46206 10.46206 10.46206 10.46206 10.46206

> # Résidus pour chaque valeur observée
> resid(mod)[1:5]
         1          2          3          4          5
-2.9678749  0.2725136 -3.8045741  5.9190636  0.8559714

> # On peut vérifier la manière de calculer les résidus
> y[1:5] - fitted(mod)[1:5]
         1          2          3          4          5
-2.9678749  0.2725136 -3.8045741  5.9190636  0.8559714
```

Régression linéaire simple : 1 x quantitatif

Extraire des informations du modèle

Utiliser `str()` pour visualiser la structure d'un objet et en extraire ce que vous voulez
L'objet retourné par `summary(mod)` contient des informations supplémentaires

```
> str(summary(mod))
List of 11
 $ call          : language lm(formula = y ~ x)
 $ terms         :Classes 'terms', 'formula' length 3 y ~ x
  .. ..- attr(*, "variables")= language list(y, x)
  .. ..- attr(*, "factors")= int [1:2, 1] 0 1
  (…)

> summary(mod)$coefficients
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 10.4620597  0.6254491 16.727277 1.790848e-30
x            0.7367449  0.2553385  2.885365 4.808336e-03

> summary(mod)$coefficients[,"Std. Error"]
(Intercept)           x
  0.6254491   0.2553385

> summary(mod)$sigma
[1] 3.611032

> summary(mod)$r.square
[1] 0.07830056
```

Régression linéaire simple : 1 x quantitatif

Extraire des informations du modèle

Par exemple on peut recalculer les p valeurs des coefficients et leurs intervalles de confiance donnés par R

```
> summary(mod)
(...)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4621     0.6254  16.727  < 2e-16 ***
x             0.7367     0.2553   2.885  0.00481 **


> (tstat <- coef(mod) / summary(mod)$coefficients[,"Std. Error"])
(Intercept)           x
  16.727277    2.885365
> 2*pt(q=tstat, df = 97, lower.tail = FALSE)
 (Intercept)            x
2.440669e-30 4.818078e-03


> confint(mod)
                2.5 %    97.5 %
(Intercept) 9.2208763 11.703243
x           0.2300339  1.243456
> coef(mod) + qt(0.025, df= 98) * summary(mod)$coefficients[,"Std. Error"]
(Intercept)           x
  9.2208763   0.2300339
> coef(mod) + qt(0.975, df= 98) * summary(mod)$coefficients[,"Std. Error"]
(Intercept)           x
  11.703243    1.243456
```

Régression linéaire simple : 1 x quantitatif

Le coefficient de détermination : R^2

Le R^2 représente le % de variance expliquée par le modèle

Un $R^2 = 1$ signifie que le modèle prédit parfaitement les données, les résidus sont nuls, tous les points sont parfaitement alignés sur la droite (ou la courbe) dont la pente doit être non nulle.

Un $R^2 = 0$ signifie que le modèle n'a aucun pouvoir prédictif

Dans beaucoup de cas, le coefficient de détermination R^2 est égal au carré du coefficient de corrélation R

Attention ne pas confondre les deux.

```
> summary(mod)$r.squared
[1] 0.07830056
> cor(y, x)^2
[1] 0.07830056
```

Régression linéaire simple : 1 x quantitatif

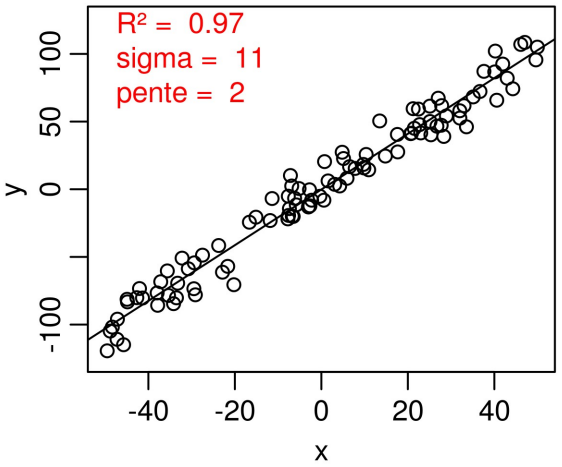
Le coefficient de détermination : R^2

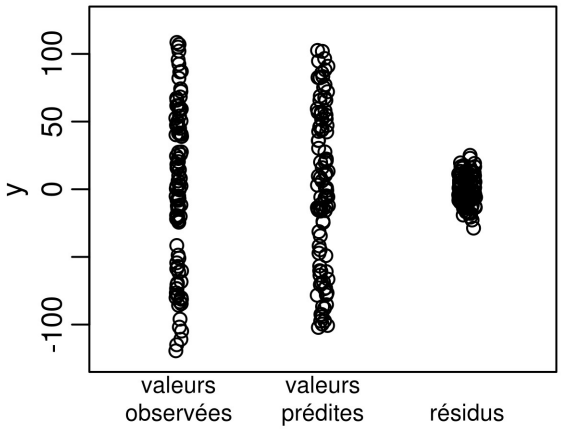
Il peut se calculer donc simplement comme le rapport entre la variance des valeurs prédites (variabilité expliquée par le modèle) et de la variance des valeurs observées (variabilité totale)

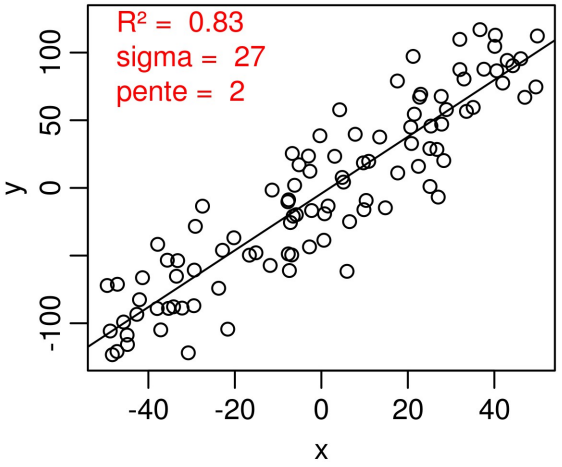
Ou encore : 1 - la variance des résidus (variabilité pas expliquée par le modèle) / la variance des valeurs observées

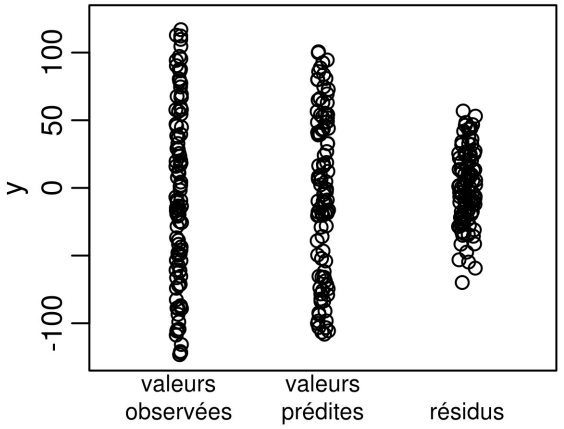
```
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.4621     0.6254  16.727  < 2e-16 ***
x             0.7367     0.2553   2.885  0.00481 **

Residual standard error: 3.611 on 98 degrees of freedom
Multiple R-squared:  0.0783,  Adjusted R-squared:  0.0689
F-statistic: 8.325 on 1 and 98 DF,  p-value: 0.004808

> var(predict(mod))/ var(y)
[1] 0.07830056

> 1-(var(resid(mod))/ var(y))
[1] 0.07830056
```

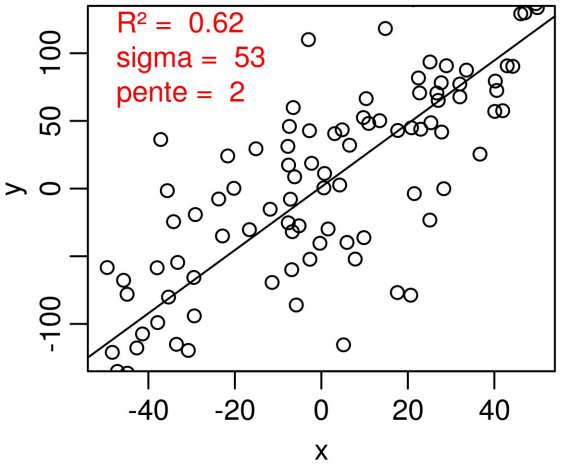
NB : en réalité le R^2 est calculé légèrement différemment :
1-[(SSres/n) / (SSy/n)]
ce qui revient au même

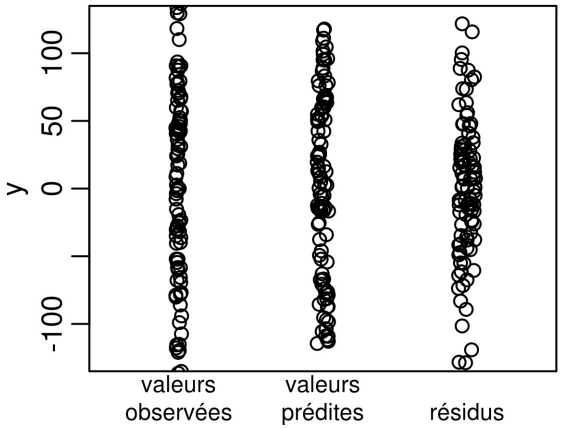








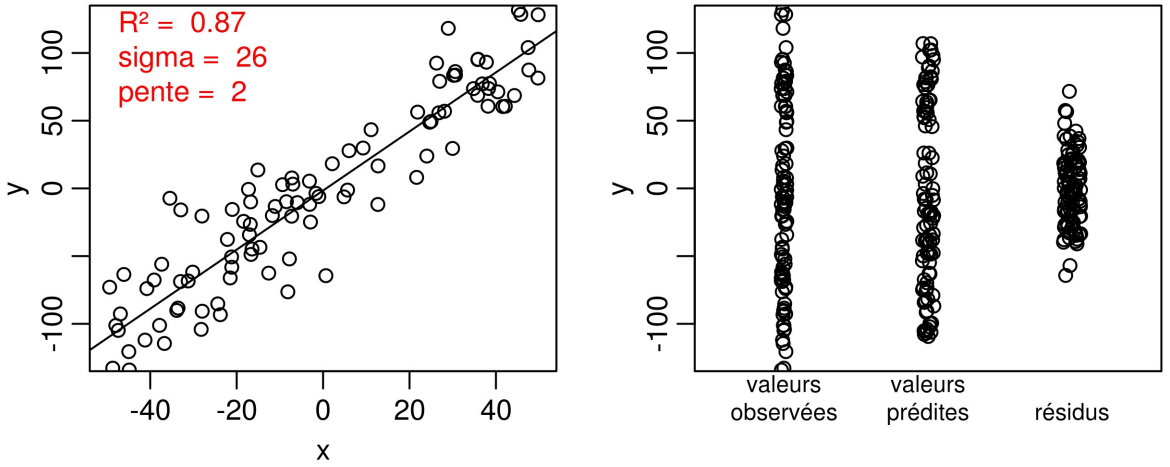


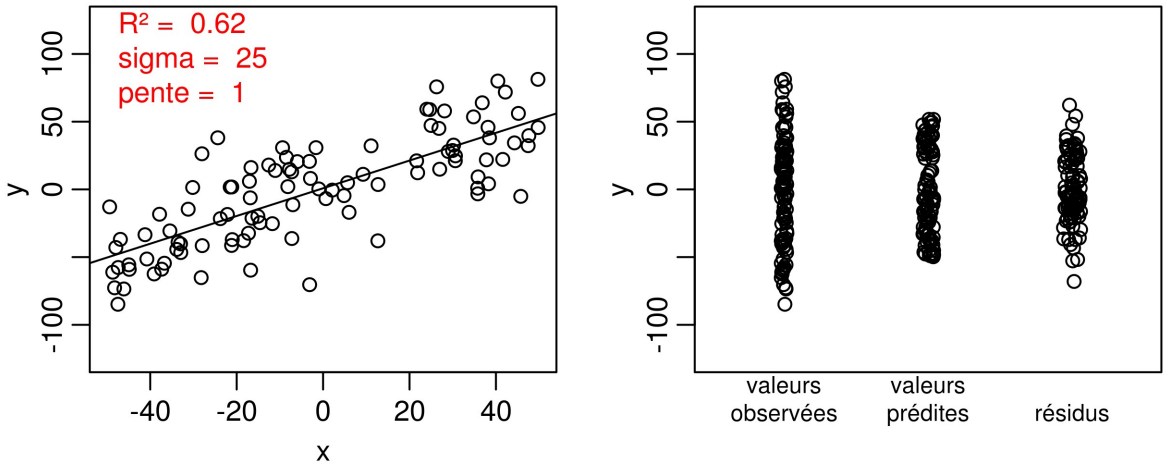


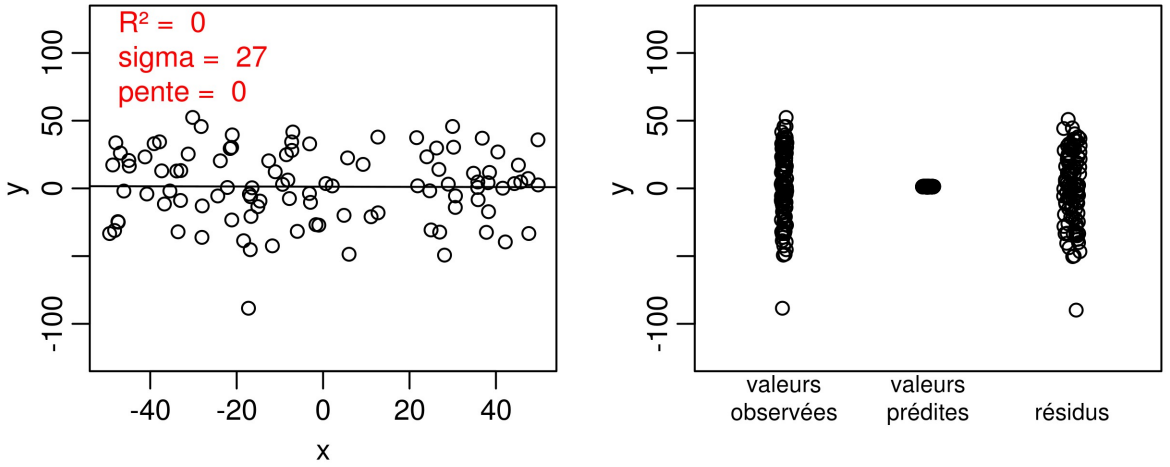
La pente est identique
La variance résiduelle change

NB, la variance des valeurs observées (= Variance totale) est égale à la somme de la variance des résidus et de la variance des valeurs prédites

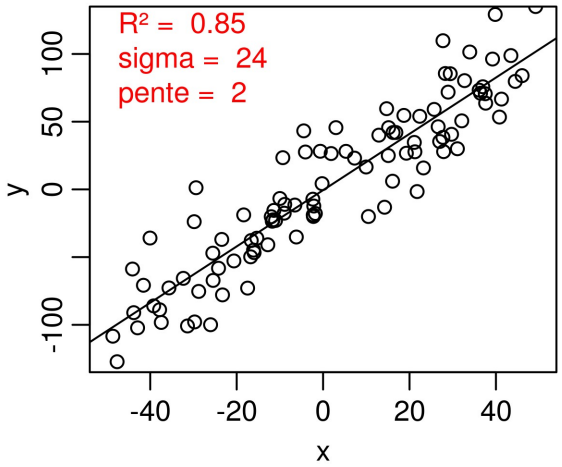
```
> var(y)
[1] 14.00439
> var(predict(mod)) + var(resid(mod))
[1] 14.00439
```

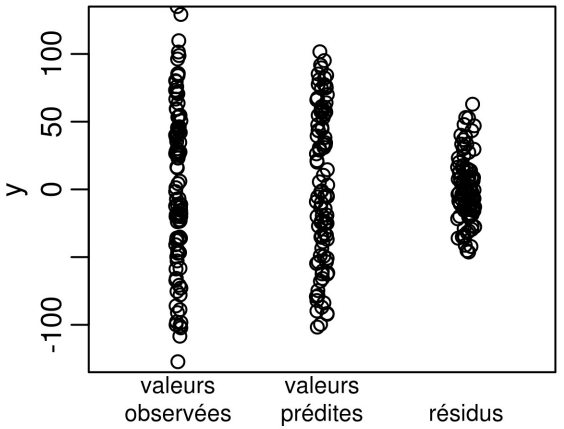


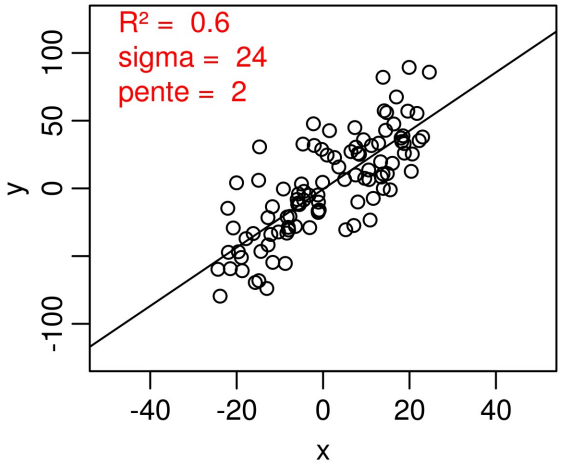


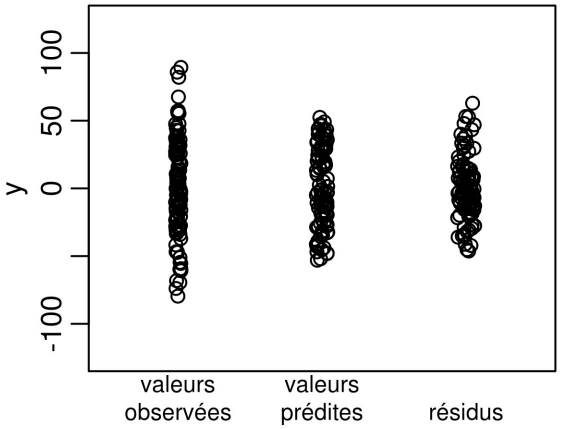


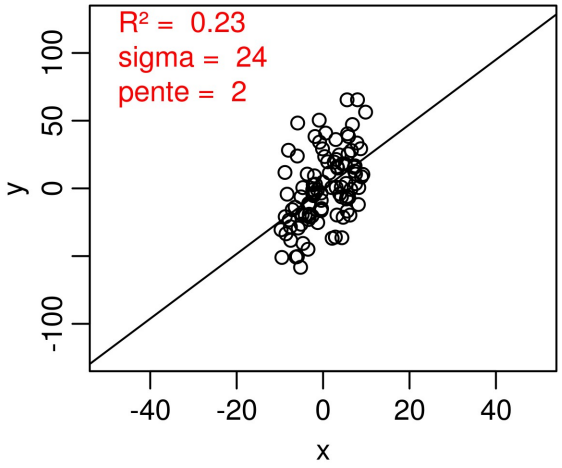
La pente change
La variance résiduelle est identique

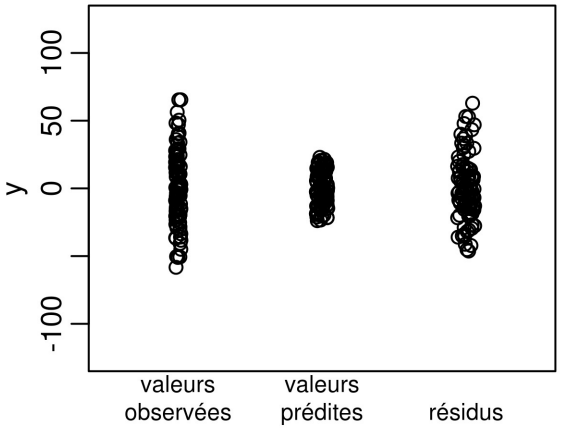












La pente est identique
La variance résiduelle est identique

Mais la gamme de valeurs de x change

Régression linéaire simple : 1 x quantitatif

Représentation graphique et valeurs prédites

Prédictions "à la main" avec un peu d'algèbre

Exemple : Quelle est la valeur prédite de y pour x = 3.45 ?

```
> (y_hat <- coef(mod)[1] + coef(mod)[2]* 3.45)
(Intercept)
   12.91871
```

On peut utiliser une série de valeurs de x dans un vecteur :

```
> (xnew <-  seq(0, 4, 0.5))
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0

> (y_hat <- coef(mod)[1] + coef(mod)[2]* xnew)
[1] 10.50290 10.85302 11.20313 11.55325 11.90337 12.25349 12.60361
[8] 12.95373 13.30384
```

Pour des problèmes plus complexes, il deviendra vite très utile de travailler avec un peu de calcul matriciel très simple...

Régression linéaire simple : 1 x quantitatif

Calcul matriciel basique

Somme (et différence)

$$\begin{pmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \end{pmatrix} + \begin{pmatrix} 5 & 2 & 1 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 6 & 4 & 1 \\ 6 & 6 & 6 \end{pmatrix}$$

Produit scalaire

$$2 * \begin{pmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 0 \\ 6 & 8 & 10 \end{pmatrix}$$

Transposition

$$X = \begin{pmatrix} 1 & 2 & 0 \\ 3 & 4 & 5 \end{pmatrix} \Leftrightarrow X^T = X' = \begin{pmatrix} 1 & 3 \\ 2 & 4 \\ 0 & 5 \end{pmatrix}$$

```
> X <- matrix(c(1,3,2,4,0,5), 2, 3)
> Y <- matrix(c(5,3,2,2,1,1), 2, 3)
>
> X+Y
     [,1] [,2] [,3]
[1,]    6    4    1
[2,]    6    6    6


> 2*X
     [,1] [,2] [,3]
[1,]    2    4    0
[2,]    6    8   10

> t(X)
     [,1] [,2]
[1,]    1    3
[2,]    2    4
[3,]    0    5
```

Régression linéaire simple : 1 x quantitatif

Calcul matriciel basique

Produit matriciel : très utile !

$$YX' = YX^T = \begin{pmatrix} 5 & 2 & 1 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 2 & 4 \\ 0 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 28 \\ 7 & 22 \end{pmatrix}$$

Détail du calcul :

$$\begin{pmatrix} 5*1+2*2+1*0 & 5*3+2*4+1*5 \\ 3*1+2*2+1*0 & 3*3+2*4+1*5 \end{pmatrix}$$

Dans R,
produit matriciel :
%*%

```
> Y %*% t(X)
     [,1] [,2]
[1,]    9   28
[2,]    7   22
```

Régression linéaire simple : 1 x quantitatif

Calcul matriciel basique

Pourquoi se casser la tête avec cà ?
Beaucoup plus facile de calculer des valeurs prédites en particulier dans des cas plus complexes

$$\hat{Y} = X\,\hat{B} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 1*\alpha+\beta*x_1 \\ 1*\alpha+\beta*x_2 \\ 1*\alpha+\beta*x_3 \\ 1*\alpha+\beta*x_4 \\ 1*\alpha+\beta*x_5 \end{pmatrix}$$

"model matrix" avec les valeurs de x pour lesquelles on veut une prédiction et une colonne de "1" correspondant à l'intercept alpha

vecteur des coefficients de régression

calcul des valeurs prédites comme on les aurait fait "à la main"

Régression linéaire simple : 1 x quantitatif

Représentation graphique et valeurs prédites

Prédictions "à la main" avec un peu de calcul matriciel

```
> X <- cbind(1, seq(0, 4, 0.01))
> X[1:5,]
     [,1] [,2]
[1,]    1 0.00
[2,]    1 0.01
[3,]    1 0.02
[4,]    1 0.03
[5,]    1 0.04
```

Nouvelles valeurs de X pour lesquelles on veut une prédiction
La première colonne correspond à l'intercept

```
> beta <- coef(mod)
> beta
(Intercept)           x
 10.5028981   0.7002367
```

Vecteur avec les coefficients

```
> y_hat <- X %*% beta
```

$$\hat{Y} = X\,\hat{B}$$

Le produit matriciel donne les valeurs prédites

```
> V <- as.matrix(vcov(mod))
> y_hat_se <- sqrt(diag(X %*% V %*% t(X)))
```

$$V\left(\hat{Y}\right) = X\,V\left(\hat{B}\right)X'$$

Cette formule utilisant la matrice de variance-covariance des coefficients donne les erreurs standard des prédictions

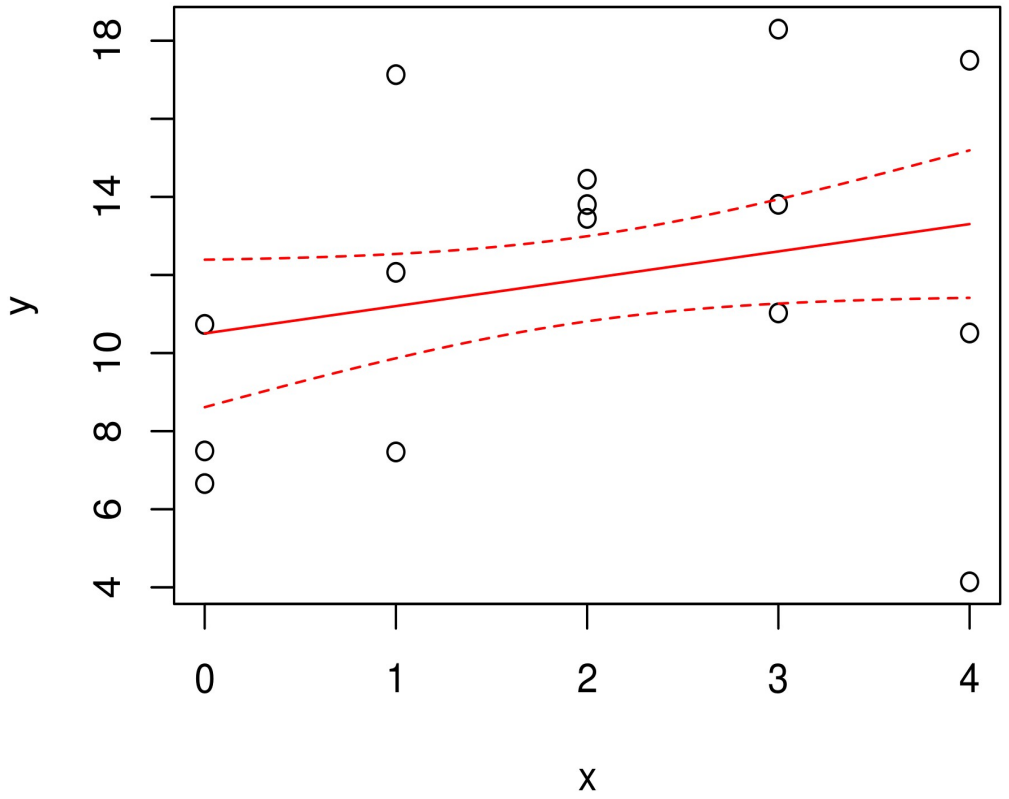
Régression linéaire simple : 1 x quantitatif

Représentation graphique et valeurs prédites

Prédictions "à la main" avec un peu de calcul matriciel

Résultats exactement identiques avec ceux obtenus via `predict()`

```
plot(y ~ x)
lines(y_hat ~ X[,2], col = "red")
lines(y_hat + y_hat_se ~ X[,2], lty = 2, col = "red")
lines(y_hat - y_hat_se ~ X[,2], lty = 2, col = "red")
```

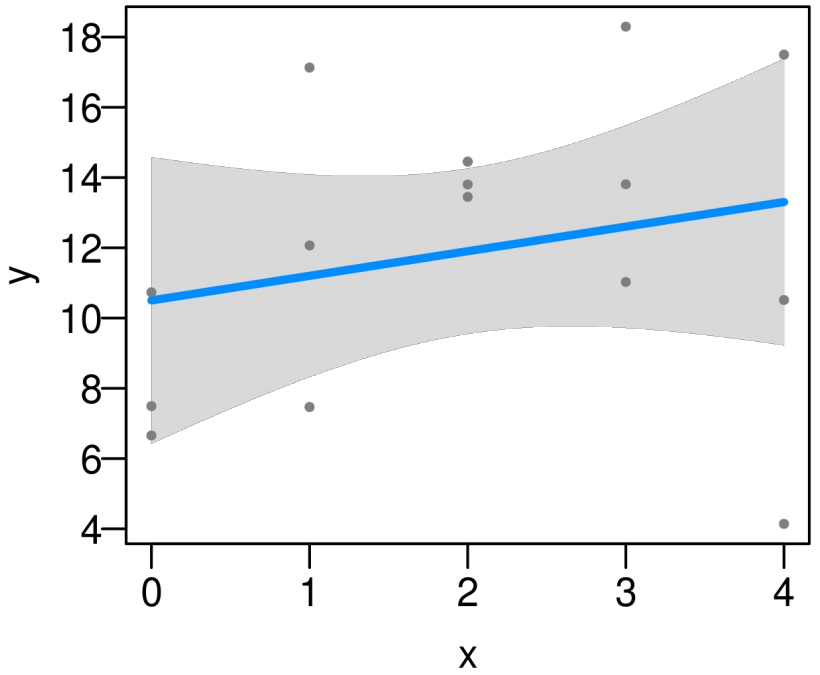


Régression linéaire simple : 1 x quantitatif

Représentation graphique et valeurs prédites

Avec visreg

```
library(visreg)
par(mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```



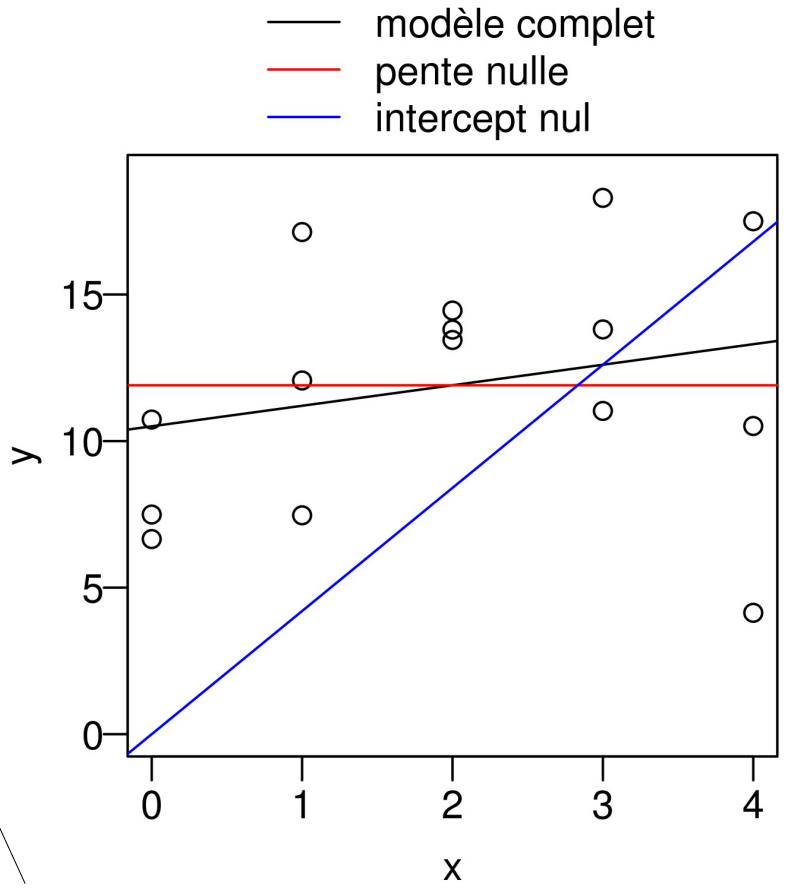
Attention, les bandes représentent les intervalles de confiance des prédictions, pas leur erreur standard

Régression linéaire simple : 1 x quantitatif

Interprétation graphique de différentes hypothèses

Modèle complet vs pente nulle vs intercept nul

```
mod0 <- lm(y ~ x) # modèle complet
mod1 <- lm(y ~ 1) # modèle à pente nulle
mod2 <- lm(y ~ -1 + x) # modèle à intercept nul

dev.new(10/2.54, 10/2.54)
par(mar = c(3,3,4,3), mgp = c(1.8, 0.5, 0), las = 1)

plot(y ~ x, ylim = c(0, 19))
abline(mod0)
abline(mod1, col = "red")
abline(mod2, col = "blue")

legend("top", legend = c("modèle complet",
              "pente nulle", "intercept nul"),
         col = c("black", "red", "blue") ,
         lty = 1, bty = "n",  xpd = NA,
         inset = -0.3)
```



`bty = "n"` : pas de cadre autour de la légende

`xpd=NA` : permet de mettre la légende en dehors de la zone de graphique

`inset` : éloigne la légende du bord (en%)

`mar` : taille des marges en nb de lignes

`mgp` : position des "ticks" et étiquettes par rapport à l'axe

`las` : orientation des étiquettes

1 x qualitatif : test de Student - ANOVA

La formulation mathématique est exactement la même mais la variable explicative est un facteur (variable qualitative) transformé en "Dummy variable"

On compare les moyennes des deux groupes ou des n groupes. On se demande si au moins une des moyennes est différente.

Exemple : Différence de production de tomates entre deux ou 3 (ou n) variétés

Concepts à assimiler :

Dummy variables, codage de variables qualitatives contrastes

1 x qualitatif : test de Student - ANOVA

Dummy variables

Il s'agit de variables composées uniquement de 0 et de 1 permettant de classer les données en groupes mutuellement exclusifs.

En général on choisi un niveau de référence (par défaut dans R : le premier dans l'ordre alphabétique) qui correspondra à l'intercept et on construit les autres variables par rapport à ce niveau.

1 x qualitatif : test de Student - ANOVA

Dummy variables

Exemple : comparaison de la production de tomates entre 2 variétés (variable qualitative à 2 niveaux)

Valeurs observées de y

Valeurs observées de x

Intercept

Var2

$$Y = X\,\hat{B} \qquad \begin{pmatrix} 21 \\ 23 \\ 16 \\ 26 \\ 34 \\ 28 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

La Col 2 indique simplement si l'observation appartient oui ou non à la variété 2

Les 3 premières observations (lignes) correspondent à la variété 1 les 3 suivantes à la variété 2

col1 correspond à α = moyenne de la variété 1 prise comme référence

Col 2 correspond à β = différence moyenne entre var 1 et var2

1 x qualitatif : test de Student - ANOVA

Dummy variables

Exemple : comparaison de la production de tomates entre 3 variétés (variable qualitative à 3 niveaux)

Les 3 premières observations (lignes) correspondent à la variété 1
les 3 suivantes à la variété 2
et les 3 dernières à la variété 3

$$Y = X\,\hat{B}$$

$$\begin{pmatrix} 21 \\ 23 \\ 16 \\ 26 \\ 34 \\ 28 \\ 10 \\ 8 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}$$

Intercept Var2 Var3

Col1 correspond à β_0 = moyenne de la variété 1 prise comme référence

Col 2 correspond à β_1 = différence moyenne entre var 1 et var 2

Col 3 correspond à β_2 = différence moyenne entre var 1 et var 3

1 x qualitatif : test de Student - ANOVA

Dummy variables

En pratique, R fait le boulot à notre place si la variable explicative est un facteur. On peut extraire la matrice utilisée avec la fonction `model.matrix()`

```
> n <- 4
> set.seed(123)
> variety1 <- rnorm(n,23,5) # mean = 23
> set.seed(12)
> variety2 <- rnorm(n,15,5) # mean = 15
>
> data <- data.frame(tomato= c(variety1,variety2),
                     variety= rep(c("variety1","variety2"), each=n))
> data
     tomato  variety
1 20.197622 variety1
2 21.849113 variety1
3 30.793542 variety1
4 23.352542 variety1
5  7.597162 variety2
6 22.885847 variety2
7 10.216278 variety2
8 10.399974 variety2
```

```
> mod <- lm(tomato ~ variety, data=data)
> model.matrix(mod)
  (Intercept) varietyvariety2
1           1                0
2           1                0
3           1                0
4           1                0
5           1                1
6           1                1
7           1                1
8           1                1
```

1 x qualitatif : test de Student - ANOVA

Dummy variables

```
> n <- 4
> set.seed(123)
> variety1 <- rnorm(n,23,5) # mean = 23
> set.seed(12)
> variety2 <- rnorm(n,15,5) # mean = 15
> set.seed(1)
> variety3 <- rnorm(n,10,5) # mean = 10
>
> data <- data.frame(tomato= c(variety1,variety2, variety3),
                     variety= rep(c("variety1","variety2", "variety3"), each=n))
> data
      tomato  variety
1  20.197622 variety1
2  21.849113 variety1
3  30.793542 variety1
4  23.352542 variety1
5   7.597162 variety2
6  22.885847 variety2
7  10.216278 variety2
8  10.399974 variety2
9   6.867731 variety3
10 10.918217 variety3
11  5.821857 variety3
12 17.976404 variety3
```

```
> mod <- lm(tomato ~ variety, data=data)
> model.matrix(mod)
   (Intercept) varietyvariety2 varietyvariety3
1            1                0                0
2            1                0                0
3            1                0                0
4            1                0                0
5            1                1                0
6            1                1                0
7            1                1                0
8            1                1                0
9            1                0                1
10           1                0                1
11           1                0                1
12           1                0                1
```

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à 2 niveaux

Un tel LM est strictement identique à un test de Student de comparaison des moyennes

Construction du jeu de données pour 2 variétés avec 4 observations par variété :

```
> n <- 4
> set.seed(123)
> variety1 <- rnorm(n,23,5) # mean = 20
> set.seed(12)
> variety2 <- rnorm(n,15,5) # mean = 15

> # on construit un jeu de données
> data <- data.frame(tomato= c(variety1,variety2),
                     variety= rep(c("variety1","variety2"), each=n))
> data
     tomato  variety
1 20.197622 variety1
2 21.849113 variety1
3 30.793542 variety1
4 23.352542 variety1
5  7.597162 variety2
6 22.885847 variety2
7 10.216278 variety2
8 10.399974 variety2
```

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à 2 niveaux

Un tel LM est strictement identique à un test de Student de comparaison des moyennes

```
> # t test
> t.test(variety1, variety2, var.equal=TRUE)
t = 2.7151, df = 6, p-value = 0.03487
```

```
> mod <- lm(tomato ~ variety, data=data)
> summary(mod)

                Estimate Std. Error t value Pr(>|t|)
(Intercept)       24.048      2.936   8.191 0.000178 ***
varietyvariety2  -11.273      4.152  -2.715 0.034867 *
```

```
> anova(mod)
Analysis of Variance Table

Response: tomato
          Df Sum Sq Mean Sq F value  Pr(>F)
variety    1 254.18 254.179   7.372 0.03487 *
Residuals  6 206.87  34.479
```

Le test de comparaison de modèles est toujours identique dans ce cas-ci

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à 2 niveaux

Un tel LM est strictement identique à un test de Student de comparaison des moyennes

```
> # t test
> t.test(variety1, variety2, var.equal=TRUE)
t = 2.7151, df = 6, p-value = 0.03487
```

On estime que le poids moyen des tomates est de 24.048 kg quand x = 0 c'est à dire pour la variété 1. La probabilité d'obtenir un tel poids par hasard (si la production réelle était nulle) est très faible : p = 0.00018 Ce test n'a pas beaucoup d'intérêt ici...

```
> mod <- lm(tomato ~ variety, data=data)
> summary(mod)

                Estimate Std. Error t value Pr(>|t|)
(Intercept)       24.048      2.936   8.191 0.000178 ***
varietyvariety2  -11.273      4.152  -2.715 0.034867 *
```

On estime que en moyenne la variété 2 produit 11.27 kg de tomates en moins que la variété 1. La probabilité d'obtenir une telle différence uniquement par hasard (si il n'y avait aucune différence) est faible : p = 0.0349

```
> anova(mod)
Analysis of Variance Table

Response: tomato
          Df Sum Sq Mean Sq F value  Pr(>F)
variety    1 254.18 254.179   7.372 0.03487 *
Residuals  6 206.87  34.479
```

Le test de comparaison de modèles est toujours identique dans ce cas-ci

1 x qualitatif : test de Student - ANOVA

Intervalles de confiance sur les paramètres

Classiquement on peut obtenir un intervalle de confiance à 95 % approximatif en prenant le paramètre +- 2* son erreur standard.
Par exemple la différence de production serait comprise entre `-11.273 - 2*4.152` et `-11.273 + 2*4.152`

Mais comme le nombre d'observations est très faible ici, il vaut mieux utiliser les quantiles de la loi de Student ou la fonction `confint()` :

```
> mod <- lm(tomato ~ variety, data=data)
> summary(mod)

                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        24.048      2.936   8.191 0.000178 ***
varietyvariety2  -11.273      4.152  -2.715 0.034867 *

> confint(mod)
                    2.5 %    97.5 %
(Intercept)      16.86422 31.232186
varietyvariety2 -21.43307 -1.113705
```

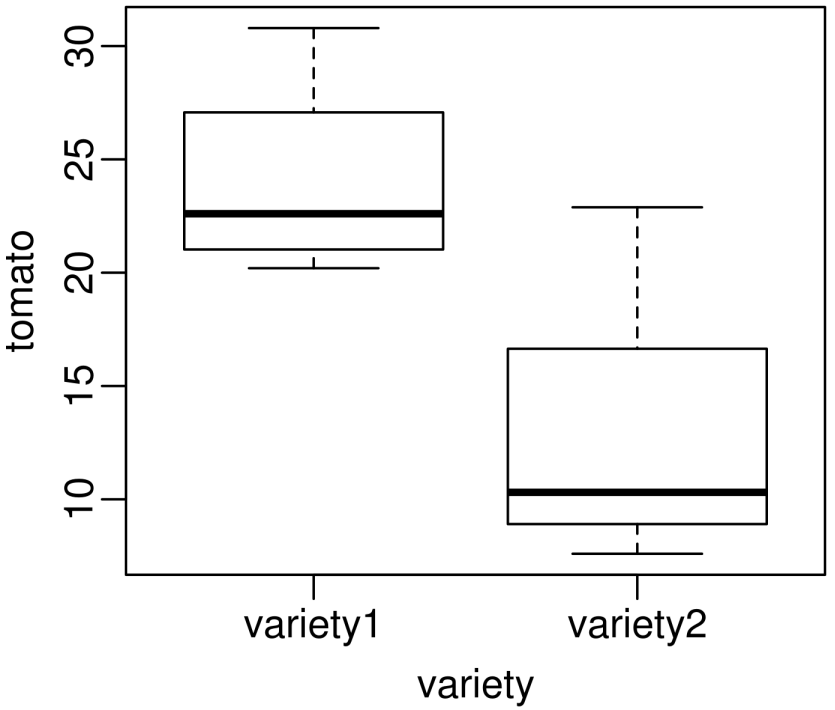
Si on répétait l'expérience 1000 fois l'estimation de la différence de production serait comprise dans 95 % des cas entre -21.4 et -1.11 kg

1 x qualitatif : test de Student - ANOVA

Représentation graphique

Si on représente les données par défaut, on obtient un boxplot. Il s'agit d'une représentation des données mais pas du modèle.

```
> par(mar= c(3,3,1,1),mgp=c(1.75,0.5, 0))
> plot(tomato ~variety , data=data)
```



1 x qualitatif : test de Student - ANOVA

Représentation graphique

On calcule les valeurs prédites et leurs erreurs standard

Avec le calcul matriciel :

```
> beta <- coef(mod)
> X <- cbind(1, c(0,1))
> X
     [,1] [,2]
[1,]    1    0
[2,]    1    1

> pred <- X %*% beta

> V <- as.matrix(vcov(mod))
> se <- sqrt(diag(X %*% V %*% t(X)))
> pred
          [,1]
[1,] 24.04820
[2,] 12.77482
> se
[1] 2.935938 2.935938
```

Matrice X pour laquelle on veut une prédiction.
La première ligne représente la variété 1
La deuxième ligne représente la variété 2

1 x qualitatif : test de Student - ANOVA

Représentation graphique

On ajoute les valeurs prédites et leurs erreurs standard sur le graphique

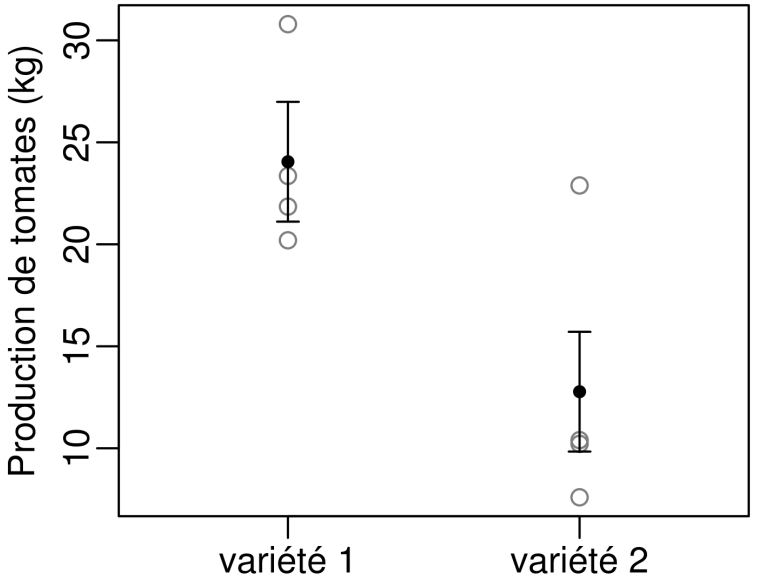
```
position <- as.numeric(data$variety)-1
plot(tomato ~ position, data=data, xaxt="n", xlim = c(-0.5, 1.5),
     col = "grey50", xlab = "", ylab = "Production de tomates (kg)" )
axis(side = 1, at = c(0,1), labels = c("variété 1", "variété 2"))
```

Ajoute les 2 points représentant les moyennes estimées au graphique existant

```
points(x = c(0,1), y = pred, pch=20 )
arrows(x0 = c(0,1), y0 = pred-se, x1 = c(0,1), y1 = pred + se,
       angle=90, length = 0.05, code = 3)
```

Ajoute les barres d'erreur

NB : On appelle parfois les valeurs prédites par le modèle : "Least Square Means"



1 x qualitatif : test de Student - ANOVA

Une variable qualitative à n niveaux

Il s'agit de ce qu'on appelle généralement une ANOVA

mais l'approche GLM met plus l'accent sur l'estimation des paramètres que sur le tableau d'analyse de la variance

```
# Génération du jeu de données : 4 variétés, 4 répétitions
n <- 4
beta0 <- 23
beta1 <- -2
beta2 <- -10
beta3 <- -12
sigma <- 4
B <- c(beta0, beta1, beta2, beta3)

variety <- rep(c("var1", "var2", "var3", "var4"), each = n)
X <- model.matrix(~ variety)

set.seed(1)
y <- X %*% B + rnorm(4*n, 0, sigma)

d <- data.frame(tomato= y, variety= variety)
> d
      tomato variety
1  20.494185    var1
2  23.734573    var1
3  19.657486    var1
4  29.381123    var1
5  22.318031    var2
(...)
```

On crée les Dummy variables à l'aide de la fonction `model.matrix()`

```
> X
   (Intercept) varietyvar2 varietyvar3 varietyvar4
1            1           0           0           0
2            1           0           0           0
3            1           0           0           0
4            1           0           0           0
5            1           1           0           0
6            1           1           0           0
7            1           1           0           0
8            1           1           0           0
9            1           0           1           0
10           1           0           1           0
11           1           0           1           0
12           1           0           1           0
13           1           0           0           1
14           1           0           0           1
15           1           0           0           1
16           1           0           0           1
```

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à n niveaux

Il s'agit de ce qu'on appelle généralement une ANOVA

mais l'approche GLM met plus l'accent sur l'estimation des paramètres que sur le tableau d'analyse de la variance

```
> mod <- lm(tomato ~ variety, data=d)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   23.317      2.045  11.403 8.52e-08 ***
varietyvar2   -1.582      2.892  -0.547 0.594356
varietyvar3   -8.145      2.892  -2.816 0.015561 *
varietyvar4  -14.073      2.892  -4.866 0.000387 ***

Residual standard error: 4.09 on 12 degrees of freedom
Multiple R-squared:  0.714,   Adjusted R-squared:  0.6425
F-statistic: 9.987 on 3 and 12 DF,  p-value: 0.001393


> mod0 <- lm(tomato ~ 1, data=d)
> mod1 <- lm(tomato ~ variety, data=d)
> anova(mod0,mod1)
Analysis of Variance Table

Model 1: tomato ~ 1
Model 2: tomato ~ variety
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     15 701.82
2     12 200.71  3    501.11 9.9871 0.001393 **
```

La comparaison de modèles emboîtés ne teste plus la même hypothèse !

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à n niveaux

Il s'agit de ce qu'on appelle généralement une ANOVA

mais l'approche GLM met plus l'accent sur l'estimation des paramètres que sur le tableau d'analyse de la variance

```
> mod <- lm(tomato ~ variety, data=d)
> summary(mod)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    23.317      2.045  11.403 8.52e-08 ***
varietyvar2    -1.582      2.892  -0.547 0.594356
varietyvar3    -8.145      2.892  -2.816 0.015561 *
varietyvar4   -14.073      2.892  -4.866 0.000387 ***

Residual standard error: 4.09 on 12 degrees of freedom
Multiple R-squared:  0.714,   Adjusted R-squared:  0.6425
F-statistic: 9.987 on 3 and 12 DF,  p-value: 0.001393
```

On estime par exemple que la variété 2 produit en moyenne, 1.58 kg en moins que la variété 1 (choisie arbitrairement comme référence) mais qu'on aurait pu obtenir le même effet par hasard (p=0.59)

```
> mod0 <- lm(tomato ~ 1, data=d)
> mod1 <- lm(tomato ~ variety, data=d)
> anova(mod0,mod1)
Analysis of Variance Table

Model 1: tomato ~ 1
Model 2: tomato ~ variety
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     15 701.82
2     12 200.71  3    501.11 9.9871 0.001393 **
```

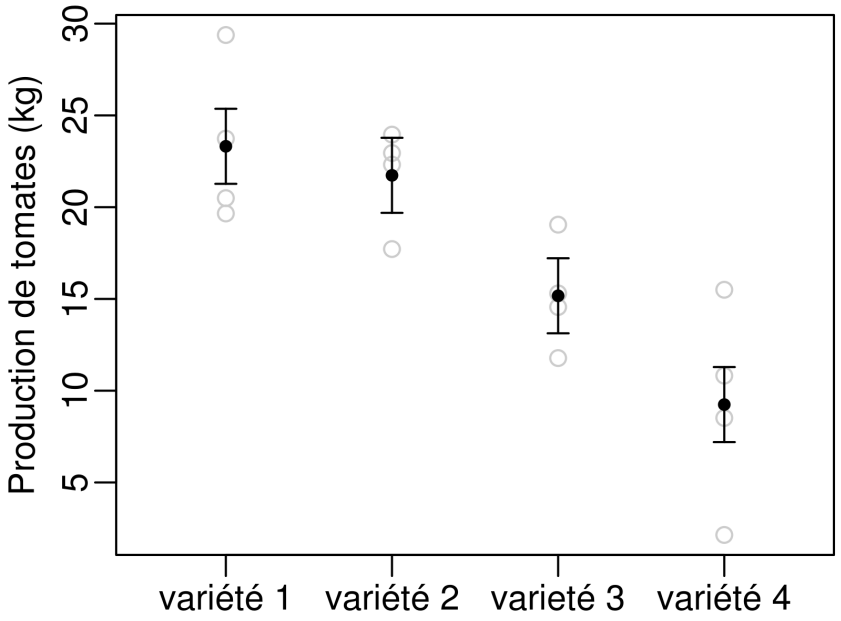
La comparaison de modèles emboîtés ne teste plus la même hypothèse ! Elle teste l'hypothèse qu'au moins une des variétés a une production différente de d'une autre variété quelconque. C'est en général cette question qu'il faut se poser en premier !!

1 x qualitatif : test de Student - ANOVA

Une variable qualitative à n niveaux

Représentation graphique

```
> position <- as.numeric(d$variety)-1
> plot(tomato ~ position, data=d, xaxt="n", xlim = c(-0.5, 3.5),
+      col = "grey80", xlab = "",ylab = "Production de tomates (kg)" )
> axis(side = 1, at = c(0,1,2,3), labels =
        c("variété 1", "variété 2", "varieté 3", "variété 4"))

> pred <- predict(mod, data.frame(variety =
        as.factor(c("var1", "var2", "var3", "var4"))), se.fit=TRUE)

> beta <- coef(mod)
> X <- rbind(c(1,0,0,0), c(1,1,0,0),
                 c(1,0,1,0), c(1,0,0,1))
> X
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    1    1    0    0
[3,]    1    0    1    0
[4,]    1    0    0    1

> pred <- X %*% beta

> V <- as.matrix(vcov(mod))
> se <- sqrt(diag(X %*% V %*% t(X)))

> points(x = c(0,1,2,3), y = pred, pch=20 )
> arrows(x0 = c(0,1,2,3), y0 = pred-se, x1 = c(0,1,2,3), y1 = pred + se,
    angle=90, length = 0.05, code = 3)
```

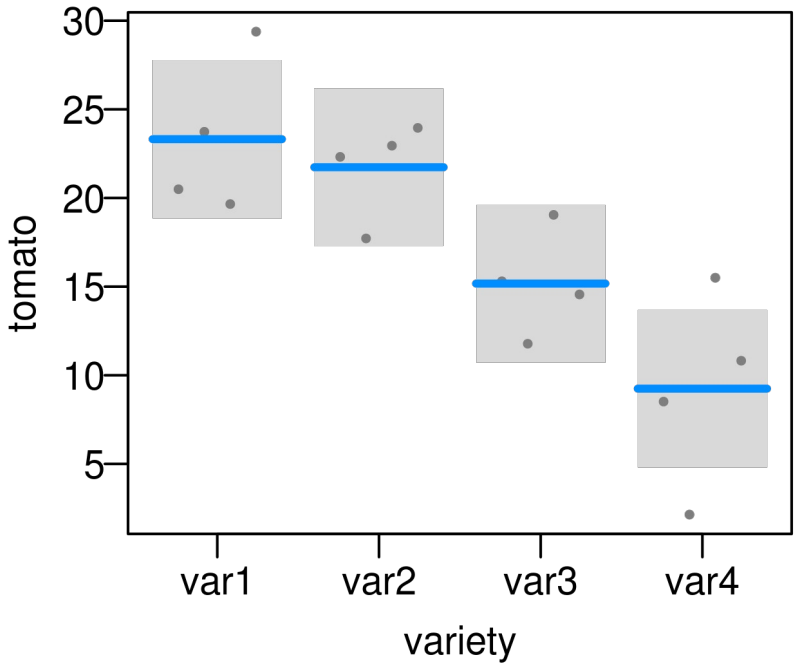


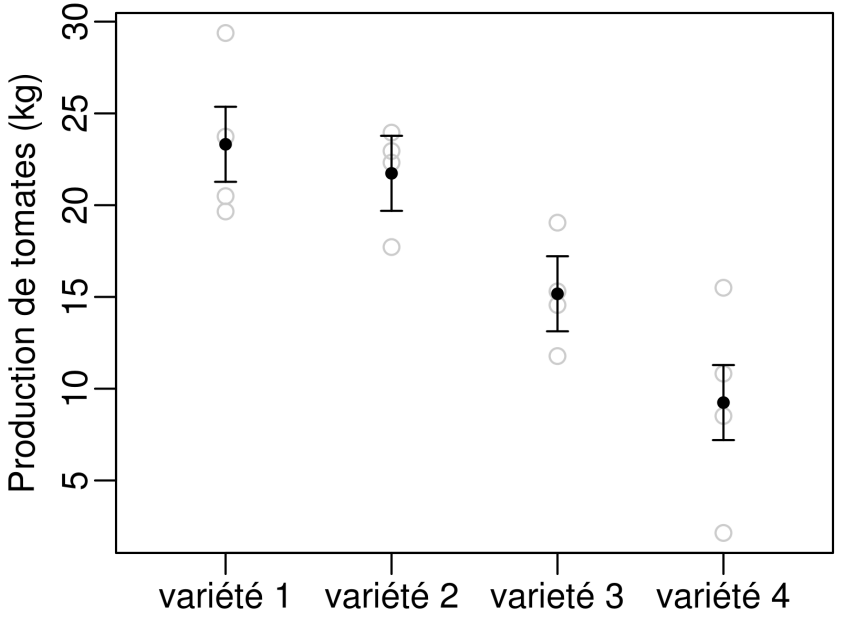
1 x qualitatif : test de Student - ANOVA

Une variable qualitative à n niveaux

Représentation graphique avec visreg

```
library(visreg)
par(mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```





1 x qualitatif : test de Student - ANOVA

Changer le niveau de référence

On peut changer le niveau du facteur utilisé pour l'intercept

```
> mod <- lm(tomato ~  variety, data=d)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   23.317      2.045  11.403 8.52e-08 ***
varietyvar2   -1.582      2.892  -0.547 0.594356
varietyvar3   -8.145      2.892  -2.816 0.015561 *
varietyvar4  -14.073      2.892  -4.866 0.000387 ***

> d$variety <- relevel(d$variety, ref = "var4")
> mod <- lm(tomato ~ variety, data=d)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.244      2.045   4.521 0.000701 ***
varietyvar1   14.073      2.892   4.866 0.000387 ***
varietyvar2   12.491      2.892   4.319 0.000997 ***
varietyvar3    5.928      2.892   2.050 0.062883 .
```

1 x qualitatif : test de Student - ANOVA

Changer le niveau de référence

On peut changer l'ordre d'affichage des niveaux du facteur

```
> d$variety <- factor(d$variety, levels =
                      c("var3", "var4", "var1", "var2"))
> mod <- lm(tomato ~ variety, data=d)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.172      2.045   7.420 8.06e-06 ***
varietyvar4   -5.928      2.892  -2.050   0.0629 .
varietyvar1    8.145      2.892   2.816   0.0156 *
varietyvar2    6.563      2.892   2.269   0.0425 *
```

Plusieurs x quantitatifs : régression multiple

Concepts à assimiler :

Effet marginal
Overfitting
Comparaison de modèles : Type I vs Type II/Type III

Plusieurs x quantitatifs : régression multiple

Exemple : on veut caractériser l'effet d'un fertilisant sur la production de tomates mais pendant l'expérience, les tomates ont été attaquées par le mildiou qui a provoqué de fortes pertes. On a donc mesuré également l'abondance du mildiou et on va le contrôler statistiquement à défaut d'avoir pu le contrôler expérimentalement.

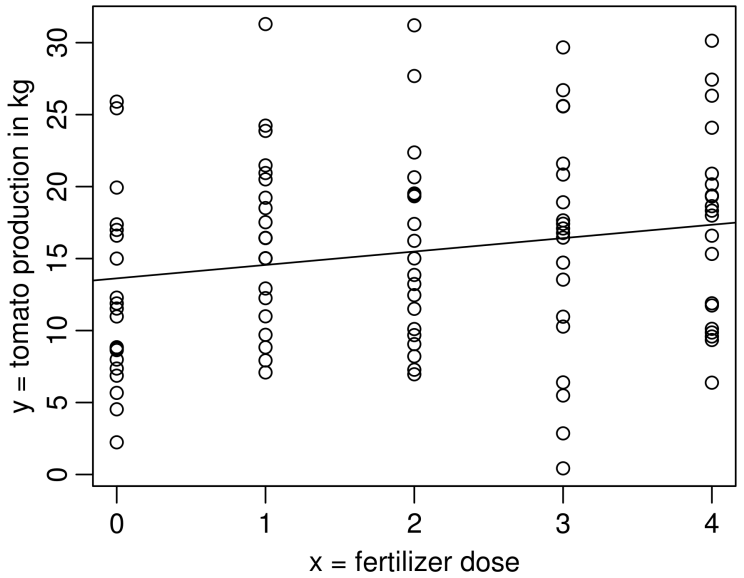
```
n <- 100
beta0 <- 25
beta1 <- 0.5
beta2 <- 5
sigma <- 5

fertilizer <- rep (0:4, each=n/5)
set.seed(1)
mildew <- runif(100,0,4)
set.seed(12)
tomato <- beta0 + beta1*fertilizer -
                  beta2*mildew + rnorm(n,0,sigma)
```

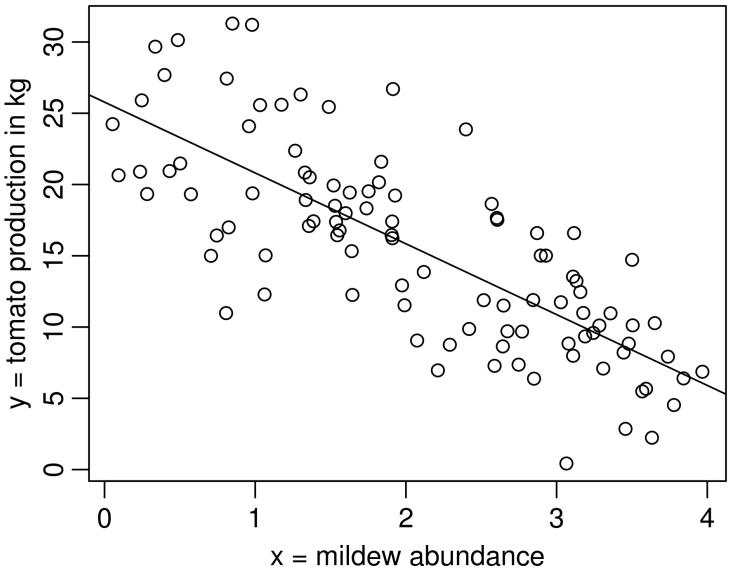
On utilise la distribution uniforme pour générer l'indice d'abondance de mildiou
Toutes les valeurs entre 0 et 4 ont la même probabilité d'être tirées

Plusieurs x quantitatifs : régression multiple

La relation tomates ~ fertilisant est assez faible et de plus les points sont fortement étalés autour de la droite (les résidus sont grands) ce qui masque la relation.
Une grande partie de ce "bruit" est dû à l'effet du mildiou...

```
par(mar = c(3,3,1,1), mgp = c(1.75, 0.6, 0))
plot(tomato ~ fertilizer, ylab =
        "y = tomato production in kg", xlab = "x = fertilizer dose")
abline(lm(tomato ~ fertilizer))
plot(tomato ~ mildew, ylab =
        "y = tomato production in kg", xlab = "x = mildew abundance")
abline(lm(tomato ~ mildew))
```





Plusieurs x quantitatifs : régression multiple

Effet marginal

Dans une régression multiple, l'effet de chaque variable explicative est estimée après avoir "enlevé", "contrôlé" l'effet des autres variables, comme si les autres variables étaient = 0

```
> model <- lm(tomato ~ fertilizer)
> summary(model)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.6255     1.1862  11.486   <2e-16 ***
fertilizer    0.9309     0.4843   1.922   0.0575 .

> model <- lm(tomato ~ fertilizer + mildew)
> summary(model)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.0389     1.1420  21.051  < 2e-16 ***
fertilizer    0.8423     0.3071   2.743  0.00726 **
mildew       -4.9417     0.4078 -12.118  < 2e-16 ***
```

Dans une régression simple, l'effet du fertilisant est limite significatif.

Si on ajoute l'effet du mildiou, les erreurs standard sont nettement plus faible et l'effet du fertilisant devient clairement significatif.

Plusieurs x quantitatifs : régression multiple

Effet marginal

Dans une régression multiple, l'effet de chaque variable explicative est estimée après avoir "enlevé", "contrôlé" l'effet des autres variables, comme si les autres variables étaient = 0

```
> model <- lm(tomato ~ fertilizer)
> summary(model)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.6255     1.1862  11.486   <2e-16 ***
fertilizer    0.9309     0.4843   1.922   0.0575 .

> model <- lm(tomato ~ fertilizer + mildew)
> summary(model)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.0389     1.1420  21.051  < 2e-16 ***
fertilizer    0.8423     0.3071   2.743  0.00726 **
mildew       -4.9417     0.4078 -12.118  < 2e-16 ***
```

Dans une régression simple, l'effet du fertilisant est limite significatif.

Si on ajoute l'effet du mildiou, les erreurs standard sont nettement plus faible et l'effet du fertilisant devient clairement significatif.

On estime que lorsque la dose de fertilisant et l'infestation de mildiou sont nuls, la production moyenne de tomates est de 24.04 kg ± 1.14 kg

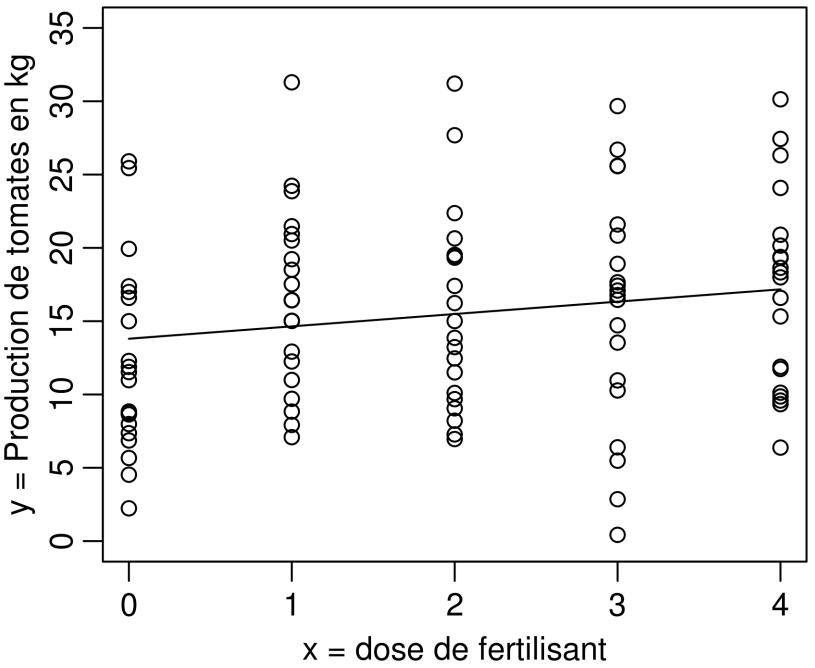
On estime que lorsque on augmente la dose de fertilisant de une unité, et que l'infestation de mildiou est 0, la production de tomate augmente de 0.84 kg ± 0.31 kg. Ici, la pente reste identique quelque soit l'infestation de mildiou (pas d'interaction)

Plusieurs x quantitatifs : régression multiple

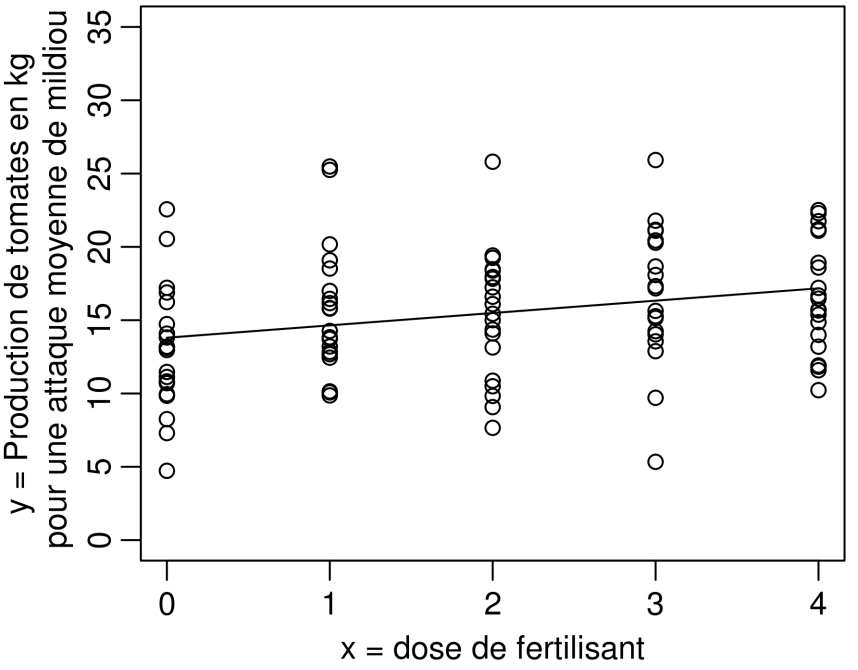
Effet marginal

Dans cet exemple, la régression multiple permet d'éliminer une partie du bruit qui est expliqué par l'effet du mildiou.

Relation tomates ~ fertilisant et données brutes



Même relation avec les données dont on a enlevé l'effet dû au mildiou
"Partial residuals"



Plusieurs x quantitatifs : régression multiple

Effet marginal

On peut estimer à la main approximativement les coefficients de la régression multiple (dans ce cas simple) en utilisant comme variable dépendante les résidus d'une régression simple

```
> model_fertilizer <- lm(tomato ~ fertilizer)
> model_mildew <- lm(tomato ~ mildew)
>
> model <- lm(tomato ~ fertilizer + mildew)
> summary(model)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.0389     1.1420  21.051  < 2e-16 ***
fertilizer    0.8423     0.3071   2.743  0.00726 **
mildew       -4.9417     0.4078 -12.118  < 2e-16 ***

> mod <- lm(resid(model_mildew) ~ fertilizer)
> summary(mod)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.6836     0.7482  -2.250  0.02667 *
fertilizer    0.8418     0.3055   2.756  0.00698 **

> mod <- lm(resid(model_fertilizer) ~ mildew)
> summary(mod)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.2304     0.9451   10.82   <2e-16 ***
mildew       -4.9389     0.4058  -12.17   <2e-16 ***
```

Les résidus de ces 2 modèles représentent les données une fois qu'on a enlevé l'effet du fertilisant ou du mildiou respectivement.

NB : degrés de liberté dans l'approche avec les résidus ne sont pas exact car ils ne prennent pas en compte l'estimation des résidus. Dans cet exemple où n est grand çà ne change pas grand chose

Plusieurs x quantitatifs : régression multiple

"Overfitting" - R^2 ajusté

Lorsque le nombre de variables explicatives augmente par rapport aux nombre de données (même si elles n'ont aucun lien avec y) :

les erreurs standard des coefficients augmentent
le R^2 augmente
la somme du carré des résidus (RSS) diminue

Lorsqu'on a autant de variables explicatives que de données même si elles n'ont aucun pouvoir explicatif, le R^2 est toujours = 1 et les RSS sont toujours = 0

Le modèle prédit alors parfaitement les données mais n'est pas généralisable à un autre jeu de données.

On ne peut donc pas utiliser le R^2 pour comparer des modèles avec des nombres de paramètres différents !

Plusieurs x quantitatifs : régression multiple

"Overfitting" - R^2 ajusté

On dit que le modèle est "surparamétrisé" ("overfitted")

Le biais du modèle est faible
Mais sa variance est très élevée : un autre jeux de données donnera des résultats (coefficients) très différents

Des variables explicatives qui on un réel lien avec la variable dépendante peuvent devenir non significatives

On applique en général des méthodes de sélection de modèle qui vont d'une manière ou d'une autre déduire les variables explicatives les plus importantes et réduire les dimensions du modèle (voir fin du module 3)

Plusieurs x quantitatifs : régression multiple

"Overfitting" - R^2 ajusté

```
> set.seed(12)
> d <- as.data.frame(matrix(runif(100, 0, 1), 10,10))
> colnames(d) <- paste0("x", 1:10)
> d$y <- 6*d$x1 + rnorm(10,0,2)
> res <- list(
+     summary(lm(y ~ x1, data=d)),
+     summary(lm(y ~ x1 + x2 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 , data=d)),
+     summary(lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10, data=d))
+ )
```

On crée 10 variables explicatives mais seule x1 explique y.
le nombre d'observations = 10

```
> # tests pour x1 dans chacun des 10 modèles
> t(sapply(res, function(x) x$coefficients[2,]))
       Estimate Std. Error    t value   Pr(>|t|)
 [1,]   5.224982   1.870292  2.7936720 0.02342528
 [2,]   5.529417   2.169264  2.5489828 0.03816023
 [3,]   6.118395   2.073810  2.9503160 0.02560246
 [4,]   4.719254   2.560075  1.8434049 0.12460084
 [5,]   5.163240   4.091366  1.2619843 0.27552470
 [6,]   3.181097   5.063452  0.6282467 0.57441594
 [7,]  -9.577013  11.699751 -0.8185656 0.49905067
 [8,] -11.459542  16.781885 -0.6828519 0.61858566
 [9,]  -7.778722        NaN        NaN        NaN
[10,]  -7.778722        NaN        NaN        NaN
```

test pour x1 dans chacun des 10 modèles

Plus on a de variables (inutiles) dans le modèle plus l'erreur standard de x1 grimpe et moins elle est significative

Plusieurs x quantitatifs : régression multiple

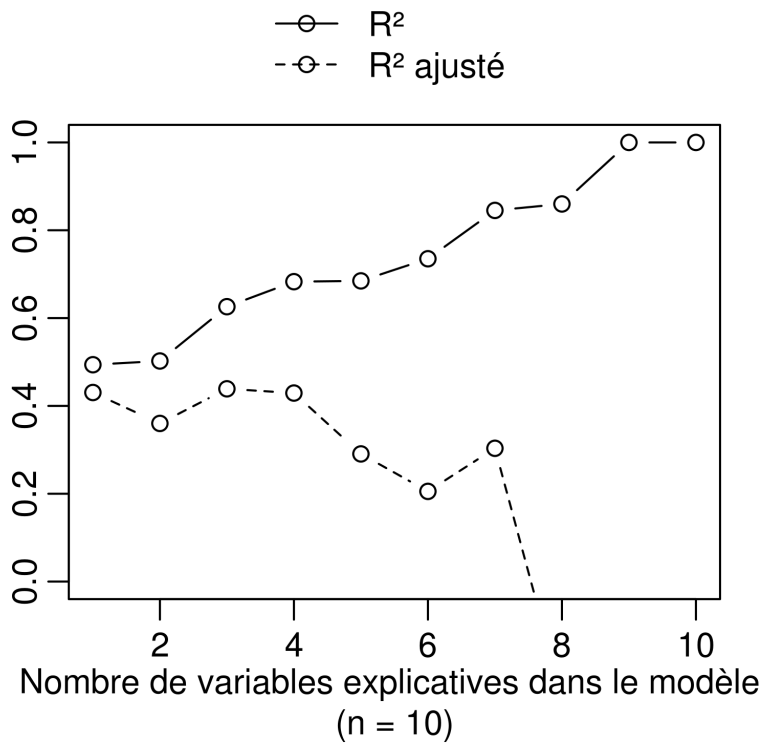
"Overfitting" - R² ajusté

Il existe une version ajustée du R² qui permet de comparer des modèles avec un nombre différent de paramètres et de données

$$R^2_{adj} = R^2 - \left(1 - R^2\right)\frac{p}{n - p - 1}$$

p représente le nombre de paramètres sans l'intercept et la variance résiduelle

```
> summary(lm(y ~ x1 + x2 + x3 + x4 + x5 , data=d))

Residual standard error: 2.203 on 4 degrees of freedom
Multiple R-squared:  0.6848, Adjusted R-squared:  0.2907
F-statistic: 1.738 on 5 and 4 DF,  p-value: 0.3062

> # extraction du R²
> (Rsq <- sapply(res, function(x) x$r.squared))
 [1] 0.4938184 0.5023192 0.6259839 0.6829472 0.6847550
0.7351300 0.8452258 0.8599036 1.0000000 1.0000000
> (Rsqadj <- sapply(res, function(x) x$adj.r.squared))
 [1]  0.4305456  0.3601247  0.4389759  0.4293050
0.2906988  0.2053899  0.3035160 -0.2608676        NaN
[10]        NaN
```



NB : l'idéal pour comparer des modèles est en général d'utiliser des méthodes par "critère d'information" (ea : AIC) que nous verrons plus loin

Plusieurs x quantitatifs : régression multiple

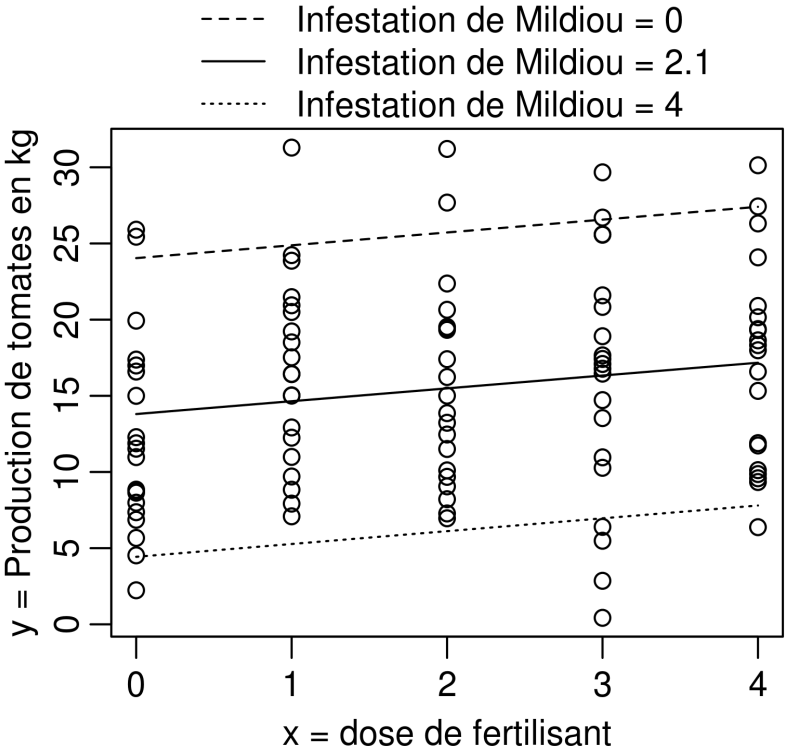
Prédiction et représentation graphique

On va en général représenter y ~ x1 en choisissant une valeur arbitraire pour les autres variables explicatives (souvent la moyenne)

```
(X <- cbind(1, c(0:4), mean(mildew)))
pred <- X %*% coef(mod)

(X <- cbind(1, c(0:4), 0))
pred0 <- X %*% coef(mod)

(X <- cbind(1, c(0:4), max(mildew)))
predmax <- X %*% coef(mod)
```

```
     [,1] [,2]  [,3]
[1,]    1    0 2.071
[2,]    1    1 2.071
[3,]    1    2 2.071
[4,]    1    3 2.071
[5,]    1    4 2.071
```

3 prédictions différentes



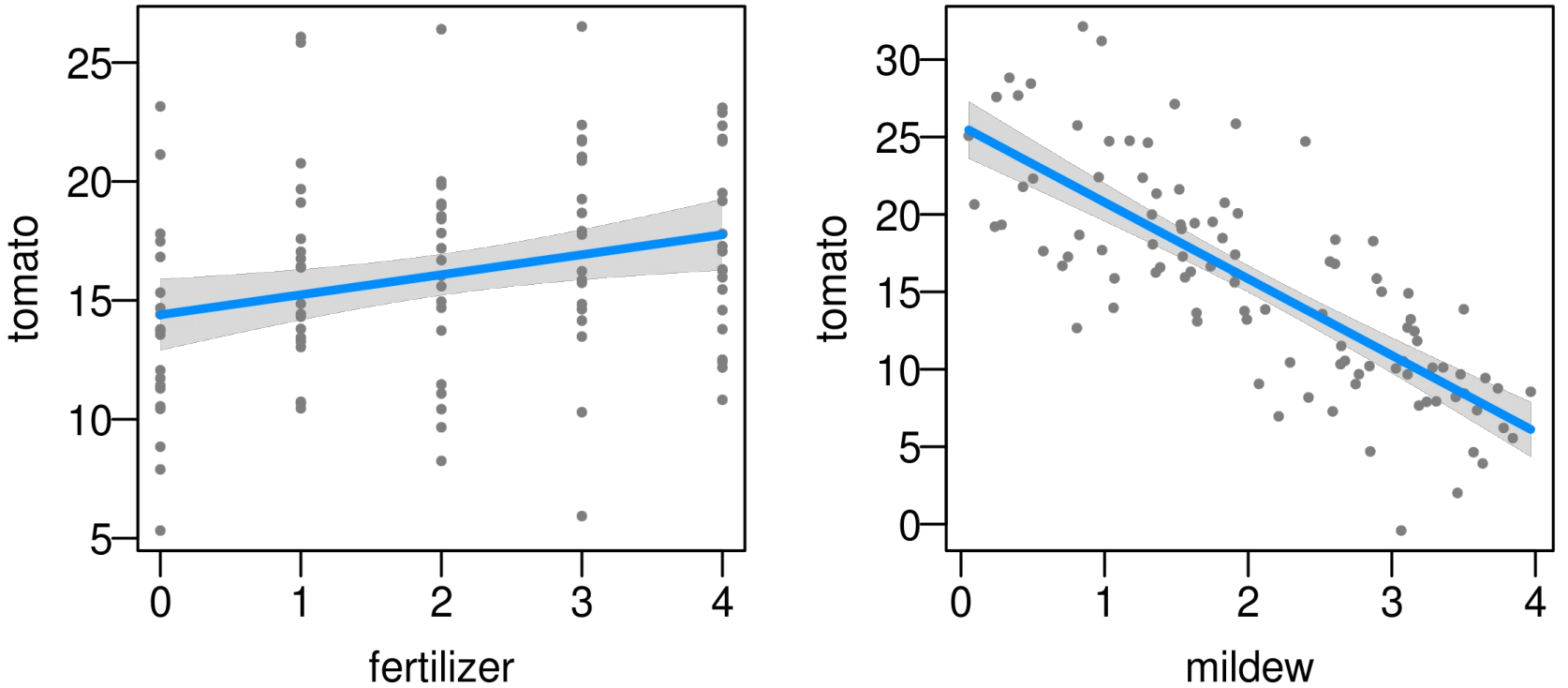

```
par(mar = c(3,4,4,2), mgp = c(1.75, 0.6, 0))
plot(tomato ~ fertilizer,
     ylab = "y = Production de tomates en kg",
     xlab = "x = dose de fertilisant")
lines(x= X[,2], y = pred, lty = 1)
lines(x= X[,2], y = pred0, lty = 2)
lines(x= X[,2], y = predmax, lty = 3)

legend(x = "top", inset = -0.3, xpd = NA, bty = "n", lty = c(2,1,3),
       legend = paste0("Infestation de Mildiou = ", round( c(0,mean(mildew), max(mildew)),1))
       )
```

Plusieurs x quantitatifs : régression multiple

Représentation graphique avec visreg

<u>Attention, les points ne sont pas les valeurs observées brutes</u>.
Il s'agit de résidus partiels. Ils représentent les valeurs observées après avoir enlevé la variation expliquée par les autres variables.

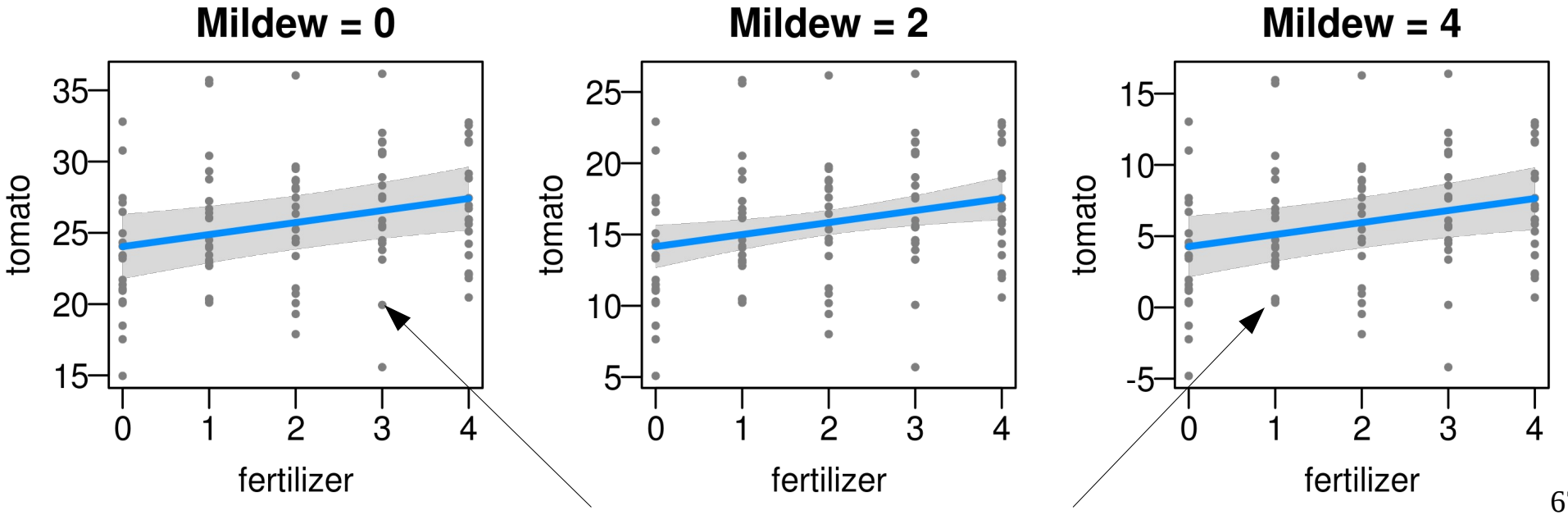
```
library(visreg)
par(mfrow = c(1,2), mar = c(3,3,1,1),
mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```



Plusieurs x quantitatifs : régression multiple

Représentation graphique avec visreg

```
par(mfrow = c(1,3), mar = c(3,3,2,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod, xvar = "fertilizer", cond = list(mildew = 0),
       main = "Mildew = 0")
visreg(mod, xvar = "fertilizer", cond = list(mildew = 2),
       main = "Mildew = 2")
visreg(mod, xvar = "fertilizer", cond = list(mildew = 4),
       main = "Mildew = 4")
```



Attention,Il s'agit de résidus partiels, pas des valeurs observées !

Plusieurs x quantitatifs : régression multiple

Comparaison de modèles : Type I vs Type II/III

Dès que l'on a plusieurs variables explicatives, il existe plusieurs manières de faire des comparaisons de modèles emboîtés.

La fonction `anova()` calcule des "Sum of Square de type I"
Elle réalise une comparaison séquentielle des modèles.
Chaque effet est testé marginalement au précédent...
En conséquence l'ordre des variables change généralement les p valeurs*
--> ce n'est clairement pas ce qu'on veut tester en général

A éviter dans la plupart des cas !

```
> anova(lm(tomato ~ fertilizer + mildew))

Response: tomato
           Df  Sum Sq Mean Sq  F value    Pr(>F)
fertilizer  1  173.31  173.31   9.1929  0.003116 **
mildew      1 2768.14 2768.14 146.8352 < 2.2e-16 ***
Residuals  97 1828.65   18.85

> anova(lm(tomato ~ mildew + fertilizer))

Response: tomato
           Df Sum Sq Mean Sq F value  Pr(>F)
mildew      1 2799.6 2799.64 148.506 < 2e-16 ***
fertilizer  1  141.8  141.80   7.522 0.00726 **
Residuals  97 1828.7   18.85
```


*sauf si les variables explicatives sont parfaitement indépendantes et que le design est parfaitement balancé

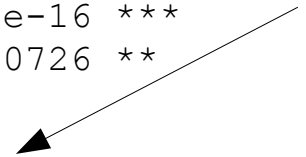
Plusieurs x quantitatifs : régression multiple

Comparaison de modèles : Type I vs Type II/III

La fonction `drop1(mod, test="F")` calcule des "Sum of Square de type II/III"
Chaque effet est testé marginalement aux autres variables.
En conséquence l'ordre des variables ne change jamais les p valeurs
--> c'est presque toujours cette approche qui est l'approche désirée

```
> drop1(lm(tomato ~ fertilizer + mildew), test = "F")

           Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>                  1828.6 296.62
fertilizer  1     141.8 1970.5 302.08   7.522 0.00726 **
mildew      1    2768.1 4596.8 386.79 146.835 < 2e-16 ***

> drop1(lm(tomato ~ mildew + fertilizer), test = "F")

           Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>                  1828.6 296.62
mildew      1    2768.1 4596.8 386.79 146.835 < 2e-16 ***
fertilizer  1     141.8 1970.5 302.08   7.522 0.00726 **
```

On peut recalculer ces valeurs avec des comparaisons de modèles emboîtés pour comprendre ce que fait cette fonction



```
> anova(lm(tomato ~ mildew), lm(tomato ~ fertilizer+ mildew))
  Res.Df    RSS Df Sum of Sq     F  Pr(>F)
2     97 1828.7  1     141.8 7.522 0.00726 **
```

test effet "fertilizer"

```
> anova(lm(tomato ~ fertilizer), lm(tomato ~ fertilizer+ mildew))
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
2     97 1828.6  1    2768.1 146.84 < 2.2e-16 ***
```

test effet "mildew"

Plusieurs x quantitatifs : régression multiple

Comparaison de modèles : Type I vs Type II/III

La fonction `Anova()` du package `car` calcule des "Sum of Square de type II" (par défaut) ou de type III (voir plus loin) et donne ici les mêmes résultats que `drop1`.
Les comparaisons de type II et III donnent les mêmes résultats, tant qu'il n'y a pas d'interaction.

```
> library(car)
> Anova(lm(tomato ~ fertilizer + mildew))
Anova Table (Type II tests)

            Sum Sq Df F value  Pr(>F)
fertilizer  141.8  1   7.522 0.00726 **
mildew     2768.1  1 146.835 < 2e-16 ***
Residuals  1828.7 97

> Anova(lm(tomato ~ mildew + fertilizer))
Anova Table (Type II tests)
            Sum Sq Df F value  Pr(>F)
mildew     2768.1  1 146.835 < 2e-16 ***
fertilizer  141.8  1   7.522 0.00726 **
Residuals  1828.7 97
```

NB : la terminologie "Type I", "Type II", "Type III" "Sum of Squares" est héritée au départ de certains logiciels (ea SAS). Ces termes sont cependant assez vagues et ne correspondent pas toujours à la même chose (en particulier type II et type III).
Cette terminologie est en général évitée dans R où on construit soi-même les modèles à comparer et où on les compare avec `anova()`

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

Concepts à assimiler

Interprétation géométrique - paramétrisation

Interactions

Syntaxe des formules de modèles

Comparaisons de modèles : Règle de marginalité et conséquences quand on ne la respecte pas (type I vs type II vs Type III tests)

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

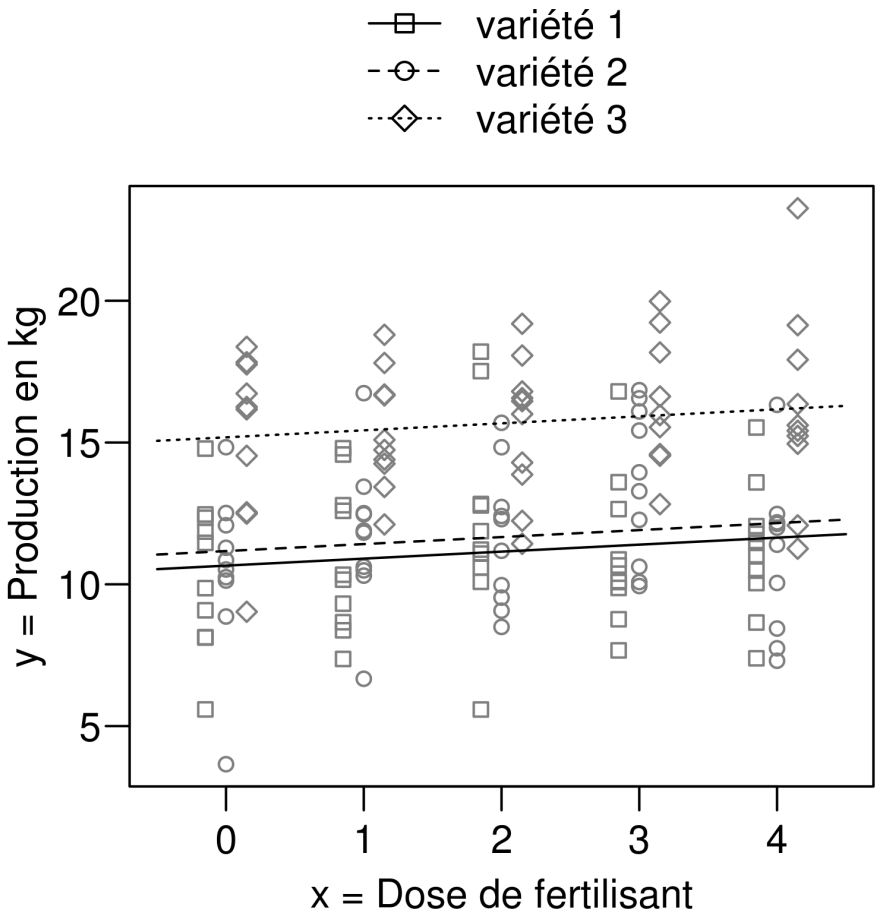
Effet de la dose de fertilisant sur 3 variétés de tomates sans interaction

Génération du jeu de données

```
> fertilizer <- rep (0:4, each=30)
> variety <- as.factor(rep(c(1, 2, 3), 50))
>
> B <- c( 10, 0.5 , 0.3, 5)
> X <- model.matrix(~ fertilizer + variety)
> set.seed(1)
> tomato <- X %*% B + rnorm(150, 0, 3)

> d <- data.frame(tomato = tomato, fertilizer=fertilizer, variety=variety)
> d
       tomato fertilizer variety
1    8.120639          0       1
2   10.850930          0       2
3   12.493114          0       3
4   14.785842          0       1
5   11.288523          0       2
6   12.538595          0       3
(...)
31  14.576039          1       1
32  10.491637          1       2
33  16.663015          1       3
34  10.338585          1       1
(...)
61  18.204853          2       1
62  11.182280          2       2
```

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

```
> mod <- lm( tomato ~ fertilizer + variety , data=d)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.6585     0.4930  21.619  < 2e-16 ***
fertilizer    0.2467     0.1559   1.582    0.116
variety2      0.5146     0.5401   0.953    0.342
variety3      4.5257     0.5401   8.380 4.05e-14 ***
```



Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

```
> mod <- lm( tomato ~ fertilizer + variety , data=d)
> summary(mod)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.6585     0.4930  21.619  < 2e-16 ***
fertilizer    0.2467     0.1559   1.582    0.116
variety2      0.5146     0.5401   0.953    0.342
variety3      4.5257     0.5401   8.380 4.05e-14 ***

Residual standard error: 2.7 on 146 degrees of freedom
Multiple R-squared:  0.3726,  Adjusted R-squared:  0.3597
F-statistic:  28.9 on 3 and 146 DF,  p-value: 9.984e-15

> drop1(mod, test = "F")
Single term deletions

Model:
tomato ~ fertilizer + variety
           Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                  1064.6 301.96
fertilizer  1     18.25 1082.9 302.51  2.5034    0.1158
variety     2    613.92 1678.6 366.26 42.0951 3.676e-15 ***
```

Intercept :
la production moyenne de la variété 1 pour une dose de fertilisant nulle est estimée à 10.66kg

fertilizer : lorsque la dose de fertilisant augmente d'une unité, la production augmente de 0.25 kg, pour la variété 1 (mais aussi pour toutes les variétés car il n'y a pas d'interaction)

variety2 et varety3 représentent la différence d'intercept.
La variété 2 produirait en moyenne 0.52 kg en plus que la variété 1 quand la dose de fertilisant est nulle mais ceci reste valable quelle que soit la dose de fertilisant car il n'y a pas d'interaction

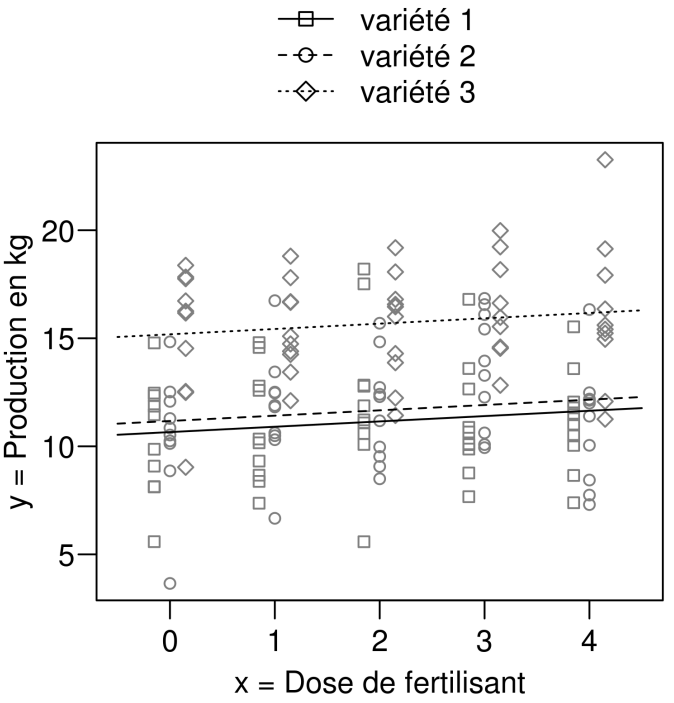
Teste l'effet global : Est-ce qu'au moins une variété a une productivité différente des autres ?

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

Représentation graphique

```
dev.new(10/2.54, 10/2.54)
par(mar = c(3,3,4,1), mgp = c(1.75, 0.6, 0), las = 1)
plot(tomato ~ fertilizer, type = "n", xlim = c(-0.5, 4.5),
     ylab = "y = Production en kg", xlab = "x = Dose de fertilisant")

points(y = d[d$variety==1, "tomato"],
        x = d[d$variety==1,"fertilizer"] - 0.15, pch = 0, cex =0.8)
points(y = d[d$variety==2, "tomato"],
        x = d[d$variety==2,"fertilizer"], pch = 1, cex =0.8)
points(y = d[d$variety==3, "tomato"],
        x = d[d$variety==3,"fertilizer"] + 0.15, pch = 5, cex =0.8)
```

```
X1 <- cbind(1, c(-0.5,4.5), 0, 0)
X2 <- cbind(1, c(-0.5,4.5), 1, 0)
X3 <- cbind(1, c(-0.5,4.5), 0, 1)

pred1 <- X1 %*% coef(mod)
pred2 <- X2 %*% coef(mod)
pred3 <- X3 %*% coef(mod)

lines(y = pred1, x = X1[,2], lty = 1)
lines(y = pred2, x = X2[,2], lty = 2)
lines(y = pred3, x = X3[,2], lty = 3)
```

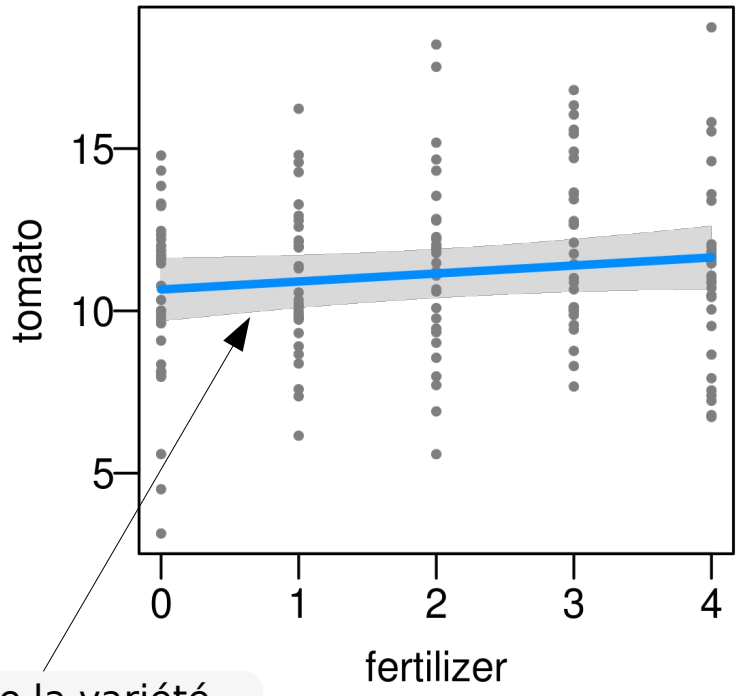


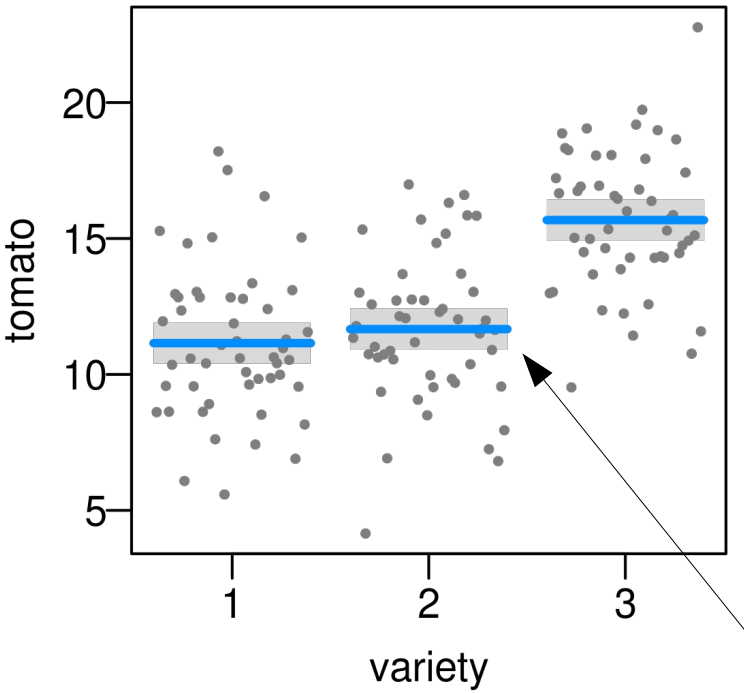

```
legend(x = "top", inset = -0.35, xpd = NA, bty = "n", lty = 1:3, pch = c(0,1,5),
       legend = c("variété 1", "variété 2", "variété 3"))
```

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

Représentation graphique avec visreg

```
library(visreg)
par(mfrow=c(1,2), mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```





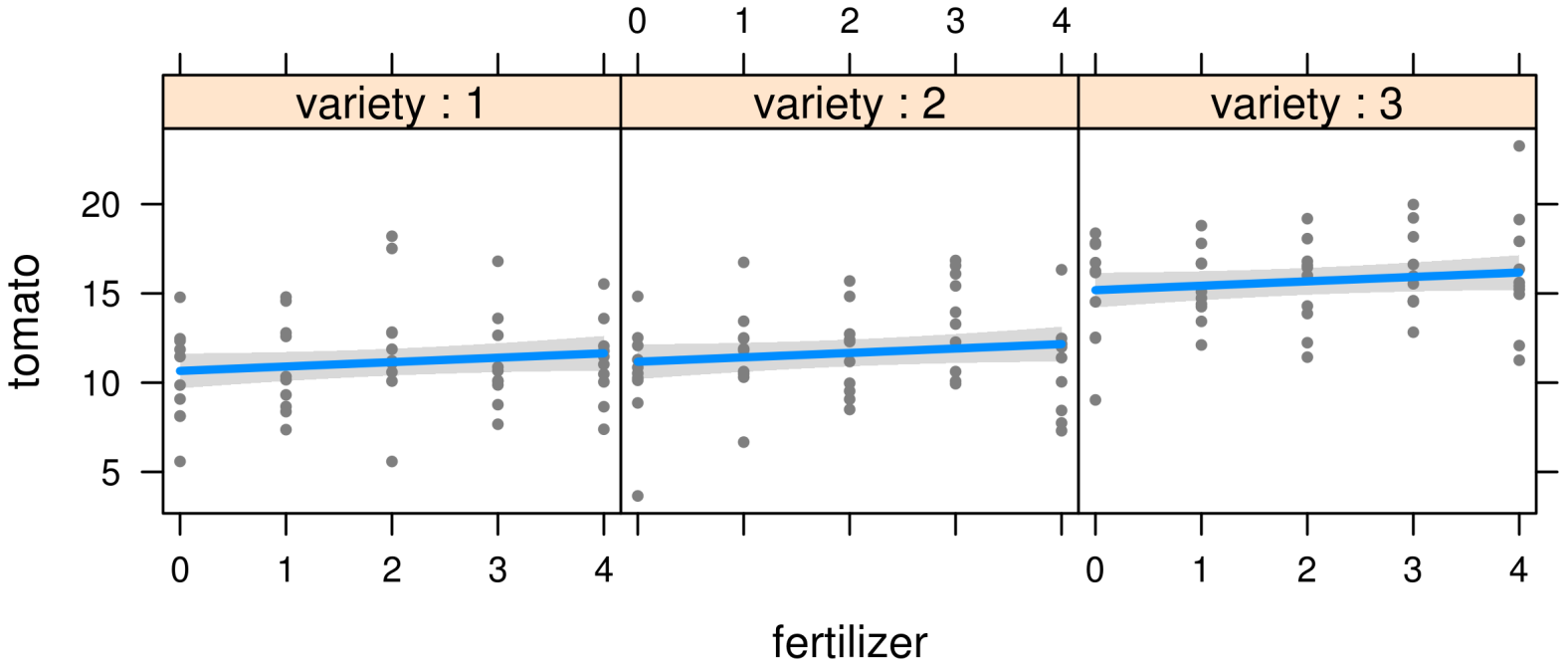
Droite de la variété 1 par défaut !

Moyenne estimée pour une quantité médiane de fertilisant par défaut !

Plusieurs x quantitatifs et/ou qualitatifs : ANCOVA

Représentation graphique avec visreg

```
par(mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod, xvar = "fertilizer", by = "variety", strip.names = TRUE)
```



Interactions

On parle d'interaction quand l'effet d'une variable explicative dépend de la valeur d'une autre variable explicative.

Pex : la relation entre la quantité de fertilisant et la production de tomates pourrait être positive pour une variété, négative pour une autre et nulle pour une troisième.

La production pourrait aussi augmenter de 1 kg quand la dose augmente de 1 unité de fertilisant pour une variété et seulement de 0.8 kg pour une autre variété.

Interactions

Effet de la dose de fertilisant sur 3 variétés de tomates avec interaction

Génération du jeu de données

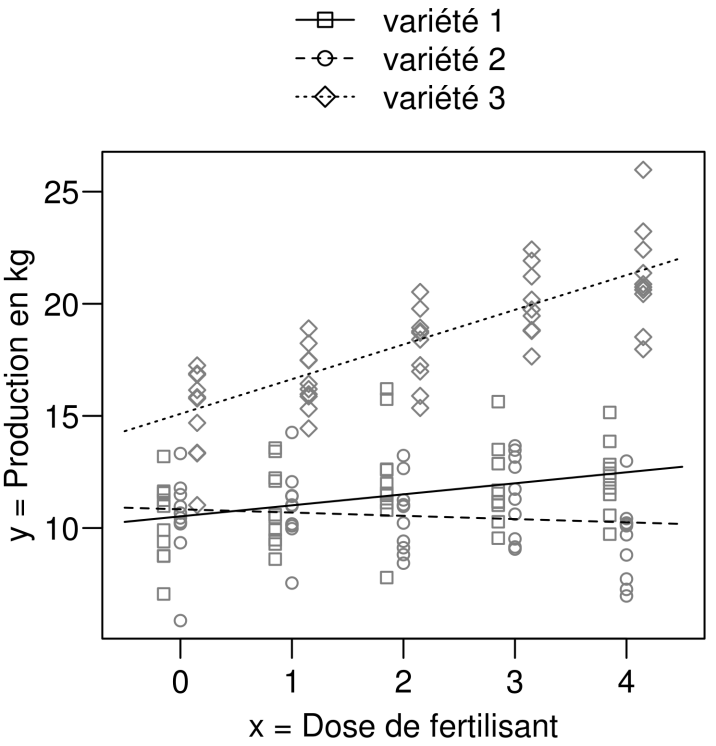
```
> fertilizer <- rep (0:4, each=30)
> variety <- as.factor(rep(c(1, 2, 3), 50))
>
> B <- c( 10, 0.7 , 0.3, 5, -0.7, 1)
> X <- model.matrix(~ fertilizer + variety + fertilizer:variety)
> X[145:150,]
    (Intercept) fertilizer variety2 variety3 fertilizer:variety2 fertilizer:variety3
145           1          4        0        0                    0                    0
146           1          4        1        0                    4                    0
147           1          4        0        1                    0                    4
148           1          4        0        0                    0                    0
149           1          4        1        0                    4                    0
150           1          4        0        1                    0                    4
> set.seed(1)
> tomato <- X %*% B + rnorm(150, 0, 2)
>
> d <- data.frame(tomato = tomato, fertilizer=fertilizer, variety=variety)
> d
       tomato fertilizer variety
1    8.747092          0       1
2   10.667287          0       2
3   13.328743          0       3
4   13.190562          0       1
5   10.959016          0       2
6   13.359063          0       3
7   10.974858          0       1
```

La colonne correspondant à l'interaction fertilisant : variété 2 contient le produit des x fertilisant * variété 2

Interactions

Ce modèle correspond à 3 droites d'intercept et de pente différents.
On estime l'intercept et la pente de la variété 1 ainsi que les différences d'intercept et de pente avec les autres variétés

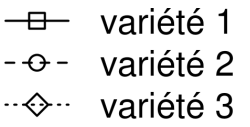
```
> mod <- lm( tomato ~ fertilizer + variety + fertilizer:variety, data=d)
> summary(mod)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          10.5183     0.4439  23.695  < 2e-16 ***
fertilizer            0.4915     0.1812   2.712   0.0075 **
variety2              0.3163     0.6278   0.504   0.6151
variety3              4.5727     0.6278   7.284 1.96e-11 ***
fertilizer:variety2  -0.6366     0.2563  -2.484   0.0141 *
fertilizer:variety3   1.0556     0.2563   4.119 6.40e-05 ***
```



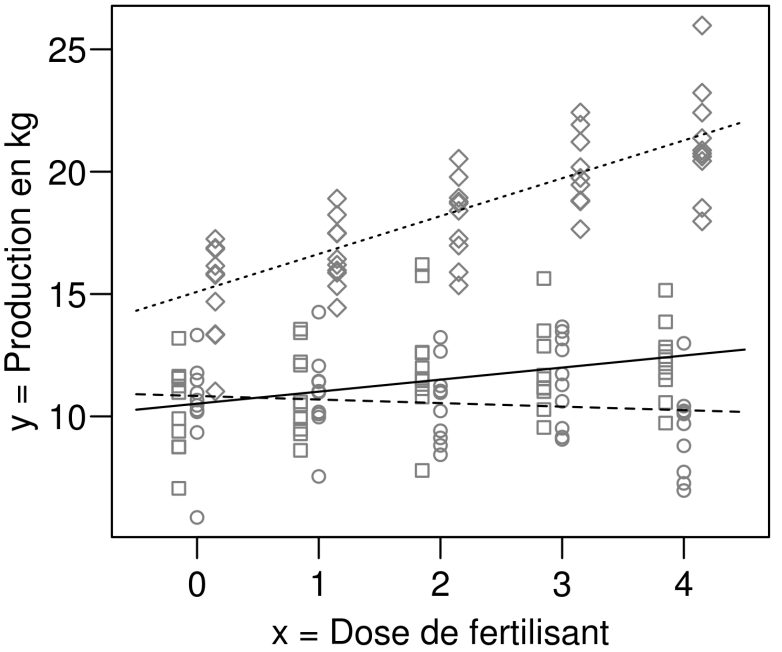
Interactions

Interprétation

```
> mod <- lm( tomato ~ fertilizer + variety + fertilizer:variety, data=d)
> summary(mod)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          10.5183     0.4439  23.695  < 2e-16 ***
fertilizer            0.4915     0.1812   2.712   0.0075 **
variety2              0.3163     0.6278   0.504   0.6151
variety3              4.5727     0.6278   7.284 1.96e-11 ***
fertilizer:variety2  -0.6366     0.2563  -2.484   0.0141 *
fertilizer:variety3   1.0556     0.2563   4.119 6.40e-05 ***
```



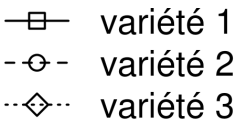
On estime que la variété 1 produit 10.52 kg de tomates pour une dose de fertilisant de 0 (Intercept) et que lorsqu'on augmente la dose de fertilisant d'une unité, la production augmente de 0.49 kg (coefficient "fertilizer") uniquement pour la variété 1.
Il est peu vraisemblable d'obtenir de telles valeurs uniquement par hasard (p<0.0001 et p = 0.0075)



Interactions

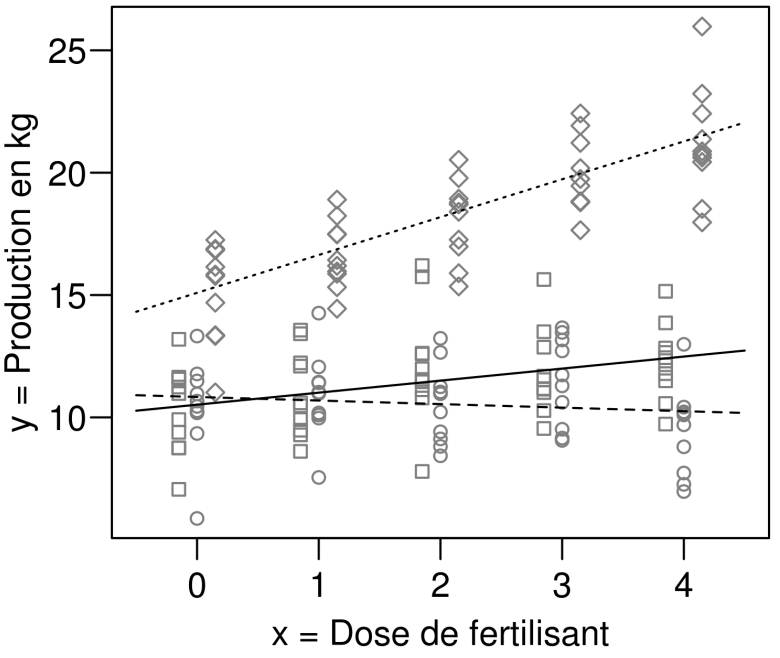
Interprétation

```
> mod <- lm( tomato ~ fertilizer + variety + fertilizer:variety, data=d)
> summary(mod)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          10.5183     0.4439  23.695  < 2e-16 ***
fertilizer            0.4915     0.1812   2.712   0.0075 **
variety2              0.3163     0.6278   0.504   0.6151
variety3              4.5727     0.6278   7.284 1.96e-11 ***
fertilizer:variety2  -0.6366     0.2563  -2.484   0.0141 *
fertilizer:variety3   1.0556     0.2563   4.119 6.40e-05 ***
```



Le coefficient "variety2" estime la différence d'intercept entre la variété 2 et la variété 1.
Le coefficient "fertilizer:variety2" estime la différence de pente entre la variété 1 et la variété 2

On estime donc que la variété 2 produit 10.52 + 0.32 kg de tomates pour une dose de fertilisant de 0 (Intercept) et que lorsqu'on augmente la dose de fertilisant d'une unité, la production de cette variété augmente de 0.49 - 0.64 kg (donc diminue de 0.15 kg).



Interactions

Interprétation

```
                  Estimate Std. Error t value Pr(>|t|)
(...)
variety2            0.3163     0.6278   0.504   0.6151
(...)
```

Le coefficient "variety2" est non significativement différent de 0 (p = 0.61).
Ça ne signifie pas qu'il n'y a pas de différence entre la variété 1 et 2 !
Ça signifie qu'il n'y a pas de différence quand la dose de fertilisant est 0 (ou du moins les données ne permettent pas de supporter une telle hypothèse)

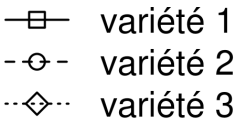
Si on centre variable "fertilizer" sur la valeur "4", l'effet "variety2" devient significatif.
On estime que quand la dose de fertilisant est 4, la variété 2 produit en moyenne 2.23 kg en moins que la variété 1

```
> d$fertilizer_c <- d$fertilizer - 4
> mod <- lm( tomato ~ fertilizer_c + variety + fertilizer_c:variety, data=d)
> summary(mod)

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           12.4842     0.4439  28.123  < 2e-16 ***
fertilizer_c           0.4915     0.1812   2.712 0.007504 **
variety2              -2.2302     0.6278  -3.552 0.000516 ***
variety3               8.7949     0.6278  14.009  < 2e-16 ***
fertilizer_c:variety2 -0.6366     0.2563  -2.484 0.014140 *
fertilizer_c:variety3  1.0556     0.2563   4.119  6.4e-05 ***
```

Interactions

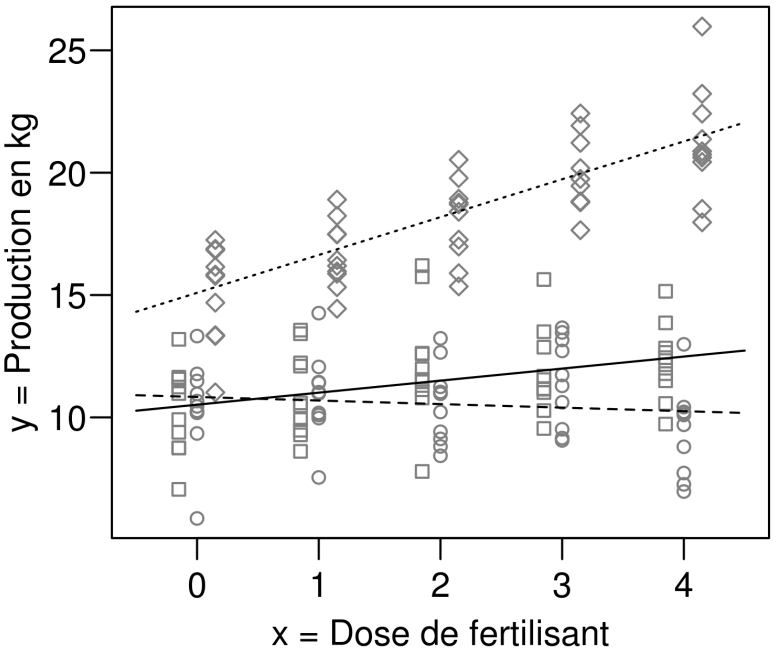
Interprétation

```
> mod <- lm( tomato ~ fertilizer + variety + fertilizer:variety, data=d)
> summary(mod)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          10.5183     0.4439  23.695  < 2e-16 ***
fertilizer            0.4915     0.1812   2.712   0.0075 **
variety2              0.3163     0.6278   0.504   0.6151
variety3              4.5727     0.6278   7.284 1.96e-11 ***
fertilizer:variety2  -0.6366     0.2563  -2.484   0.0141 *
fertilizer:variety3   1.0556     0.2563   4.119 6.40e-05 ***
```



Le coefficient "variety3" estime la différence d'intercept entre la variété 3 et la variété 1.
Le coefficient "fertilizer:variety3" estime la différence de pente entre la variété 1 et la variété 3

On estime donc que la variété 3 produit 10.52 + 4.57 kg de tomates pour une dose de fertilisant de 0 (Intercept) et que lorsqu'on augmente la dose de fertilisant d'une unité, la production de cette variété augmente de 0.49 + 1.06 kg (donc augmente de 1.55 kg).



Interactions

Représentation graphique

```
dev.new(10/2.54, 10/2.54)
par(mar = c(3,3,4,1), mgp = c(1.75, 0.6, 0), las = 1)
plot(tomato ~ fertilizer, type = "n", xlim = c(-0.5, 4.5),
     ylab = "y = Production en kg", xlab = "x = Dose de fertilisant")

points(y = d[d$variety==1, "tomato"],
       x = d[d$variety==1,"fertilizer"] - 0.15, pch = 0, cex =0.8, col = "grey50")
points(y = d[d$variety==2, "tomato"],
       x = d[d$variety==2,"fertilizer"], pch = 1, cex =0.8, col = "grey50")
points(y = d[d$variety==3, "tomato"],
       x = d[d$variety==3,"fertilizer"] + 0.15, pch = 5, cex =0.8, col = "grey50")

X1 <- cbind(1, c(-0.5,4.5), 0, 0, 0, 0)
X2 <- cbind(1, c(-0.5,4.5), 1, 0, 1*c(-0.5, 4.5), 0)
X3 <- cbind(1, c(-0.5,4.5), 0, 1, 0, 1*c(-0.5, 4.5))

pred1 <- X1 %*% coef(mod)
pred2 <- X2 %*% coef(mod)
pred3 <- X3 %*% coef(mod)

lines(y = pred1, x = X1[,2], lty = 1)
lines(y = pred2, x = X2[,2], lty = 2)
lines(y = pred3, x = X3[,2], lty = 3)

legend(x = "top", inset = -0.35, xpd = NA, bty = "n", lty = 1:3, pch = c(0,1,5),
       legend = c("variété 1", "variété 2", "variété 3"))
```

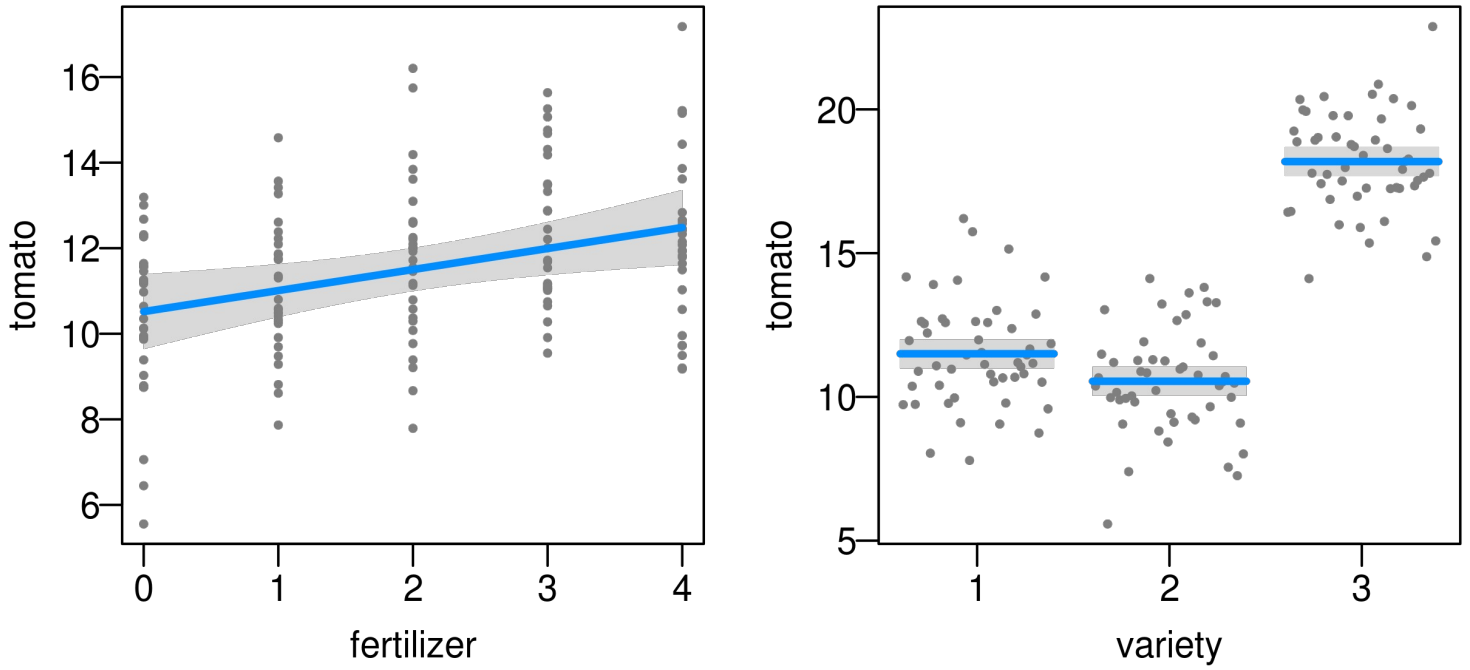
```
> X2
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1 -0.5    1    0 -0.5    0
[2,]    1  4.5    1    0  4.5    0
```

Interactions

Représentation graphique avec visreg

Pas optimal !

```
library(visreg)
par(mfrow=c(1,2), mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```

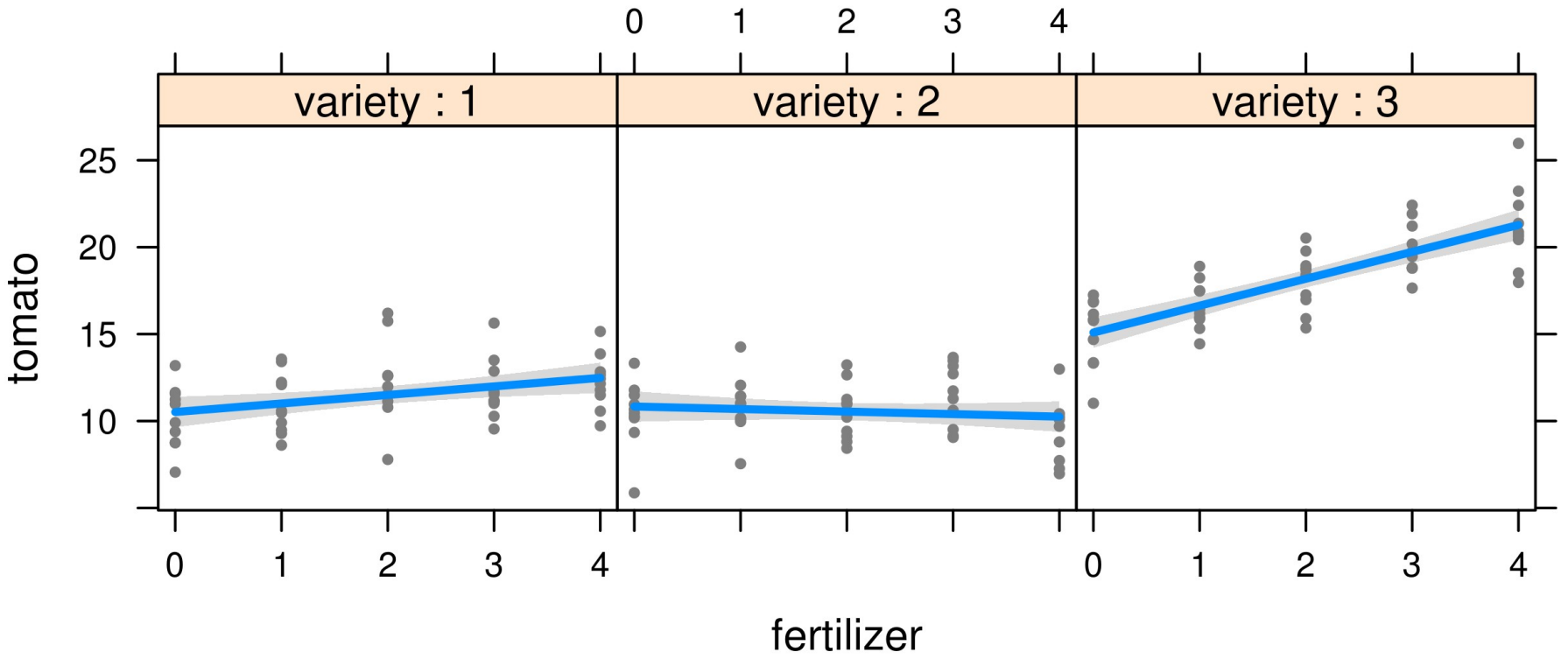


Interactions

Représentation graphique avec visreg

Mieux en spécifiant quelques arguments

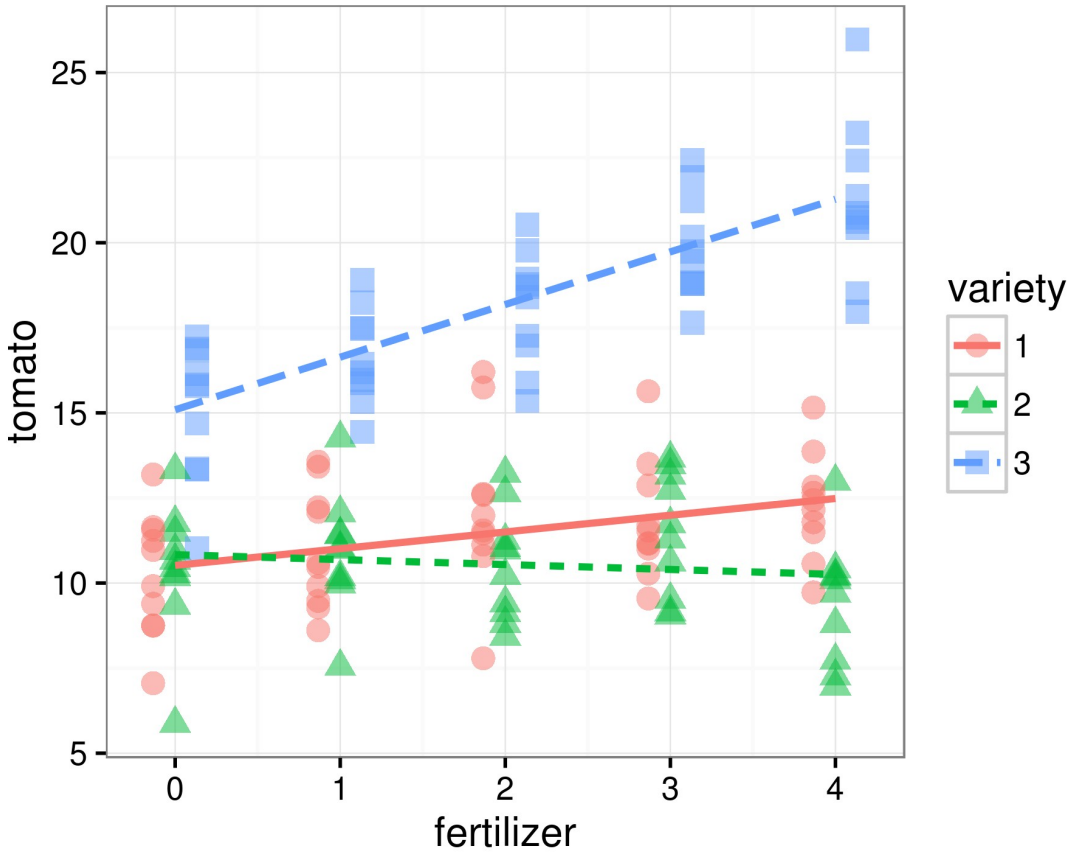
```
par(mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod, xvar = "fertilizer", by = "variety", strip.names = TRUE)
```



Interactions

Représentation graphique avec ggplot

```
library(ggplot2)

ggplot(d, aes(y = tomato, x = fertilizer,  shape = variety, color = variety)) +
    geom_point(size = 3, alpha = 0.5, position = position_dodge(0.4)) +
    stat_smooth(aes(lty = variety), method = "lm", se = FALSE) +
    theme_bw()
```



Interactions

Comparaisons multiples / Post-Hoc / Contrastes

On peut tester n'importe quelles hypothèses tout en contrôlant le risque global d'erreur de type I avec des matrices de contrastes

Exemple : on teste les 4 questions suivantes correspondant chacune à 3 hypothèses :

1) Pour quelle variétés les pentes sont-elles différentes de 0 ?
2) Quelles variétés ont-elles des pentes différentes entre elles ?
3) Quelles variétés ont des productions différentes lorsque la dose de fertilisant = 0
4) Quelles variétés ont des productions différentes lorsque la dose de fertilisant = 4

```
> X <-
+ rbind("pente var1" = c(0,1,0,0,0,0),
+       "pente var2" = c(0,1,0,0,1,0),
+       "pente var3" = c(0,1,0,0,0,1),
+
+       "pente var2 - var1" = c(0,0,0,0,1,0),
+       "pente var3 - var1" = c(0,0,0,0,0,1),
+       "pente var3 - var2" = c(0,0,0,0,-1,1),
+
+       "production var2-var1 @dose=0" = c(1,0,1,0,0,0) - c(1,0,0,0,0,0),
+       "production var3-var1 @dose=0" = c(1,0,0,1,0,0) - c(1,0,0,0,0,0),
+       "production var3-var2 @dose=0" = c(1,0,0,1,0,0) - c(1,0,1,0,0,0),
+
+       "production var2-var1 @dose=4" = c(1,4,1,0,4,0) - c(1,4,0,0,0,0),
+       "production var3-var1 @dose=4" = c(1,4,0,1,0,4) - c(1,4,0,0,0,0),
+       "production var3-var2 @dose=4" = c(1,4,0,1,0,4) - c(1,4,1,0,4,0)
+       )
```

```
                          Estimate
(Intercept)                10.5183
fertilizer                  0.4915
variety2                    0.3163
variety3                    4.5727
fertilizer:variety2        -0.6366
fertilizer:variety3         1.0556
```

Interactions

Comparaisons multiples

On peut tester n'importe quelles hypothèses tout en contrôlant le risque global d'erreur de type I

Matrice de contrastes (non indépendants) résultant :

```
> X
                                [,1] [,2] [,3] [,4] [,5] [,6]
pente var1                         0    1    0    0    0    0
pente var2                         0    1    0    0    1    0
pente var3                         0    1    0    0    0    1
pente var2 - var1                  0    0    0    0    1    0
pente var3 - var1                  0    0    0    0    0    1
pente var3 - var2                  0    0    0    0   -1    1
production var2-var1 @dose=0       0    0    1    0    0    0
production var3-var1 @dose=0       0    0    0    1    0    0
production var3-var2 @dose=0       0    0   -1    1    0    0
production var2-var1 @dose=4       0    0    1    0    4    0
production var3-var1 @dose=4       0    0    0    1    0    4
production var3-var2 @dose=4       0    0   -1    1   -4    4
```

Interactions

Comparaisons multiples

On peut tester n'importe quelles hypothèses tout en contrôlant le risque global d'erreur de type I

Comparaisons multiples avec p-valeur ajustée

NB : si on ne veut pas contrôler pour le risque global d'erreur, on peut prendre chaque coefficient estimé ± 2* Std. Error pour obtenir un intervalle de confiance approximatif à 95 %

```
> library(multcomp)
> modmc <- glht(mod, linfct = X)
> summary(modmc)

Linear Hypotheses:
                                          Estimate Std. Error t value Pr(>|t|)
pente var1 == 0                             0.4915     0.1812   2.712   0.0570 .
pente var2 == 0                            -0.1451     0.1812  -0.801   0.9447
pente var3 == 0                             1.5470     0.1812   8.536   <0.001 ***
pente var2 - var1 == 0                     -0.6366     0.2563  -2.484   0.0993 .
pente var3 - var1 == 0                      1.0556     0.2563   4.119   <0.001 ***
pente var3 - var2 == 0                      1.6922     0.2563   6.603   <0.001 ***
production var2-var1 @dose=0 == 0           0.3163     0.6278   0.504   0.9925
production var3-var1 @dose=0 == 0           4.5727     0.6278   7.284   <0.001 ***
production var3-var2 @dose=0 == 0           4.2564     0.6278   6.780   <0.001 ***
production var2-var1 @dose=4 == 0          -2.2302     0.6278  -3.552   0.0047 **
production var3-var1 @dose=4 == 0           8.7949     0.6278  14.009   <0.001 ***
production var3-var2 @dose=4 == 0          11.0251     0.6278  17.562   <0.001 ***
```

Interactions

Remarques générales sur les interactions

Les interactions sont fréquentes dans la nature mais souvent de faible intensité.
Parfois elles peuvent cependant masquer complètement des effets principaux ("main effects").

Il faut en général beaucoup plus de données pour estimer les interactions que les effets principaux

On peut avoir des interactions de deuxième niveau (A*B*C) ou plus mais on arrive rapidement à des niveaux de complexité difficilement interprétables.
Il est souvent utile dans ce cas de faire plusieurs analyses séparées pour réduire la complexité.

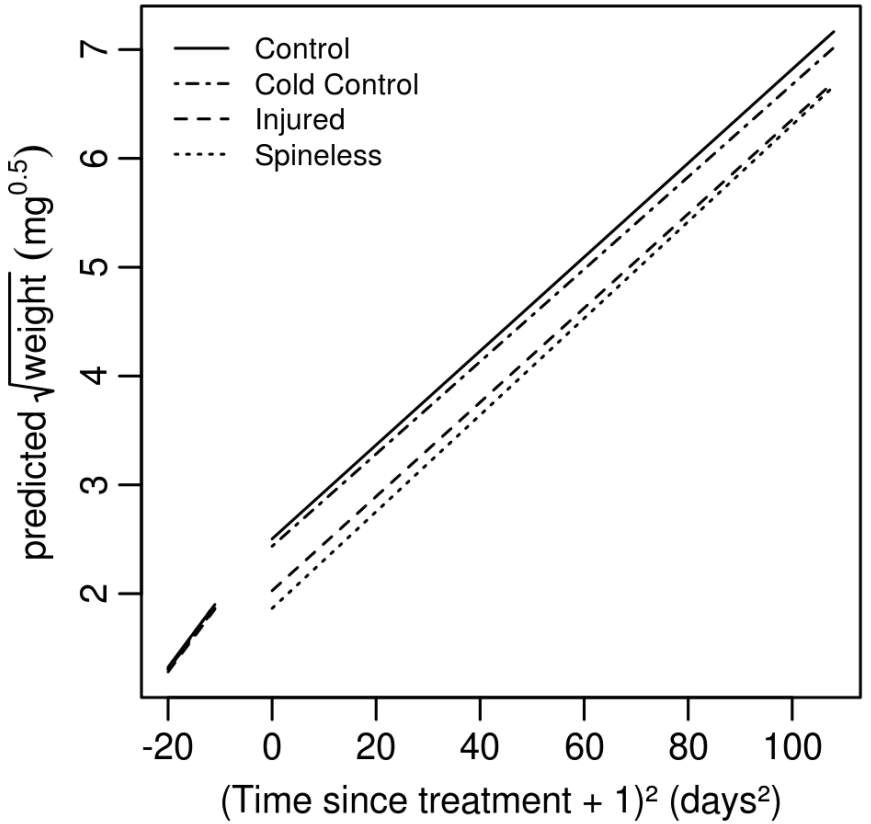
Pour des jeux de données très complexes où on peut difficilement définir a priori toutes les interactions potentielles, les analyses de Data Mining par arbres de régression (Regression Trees, Random Forests, etc...) peuvent être précieuses.

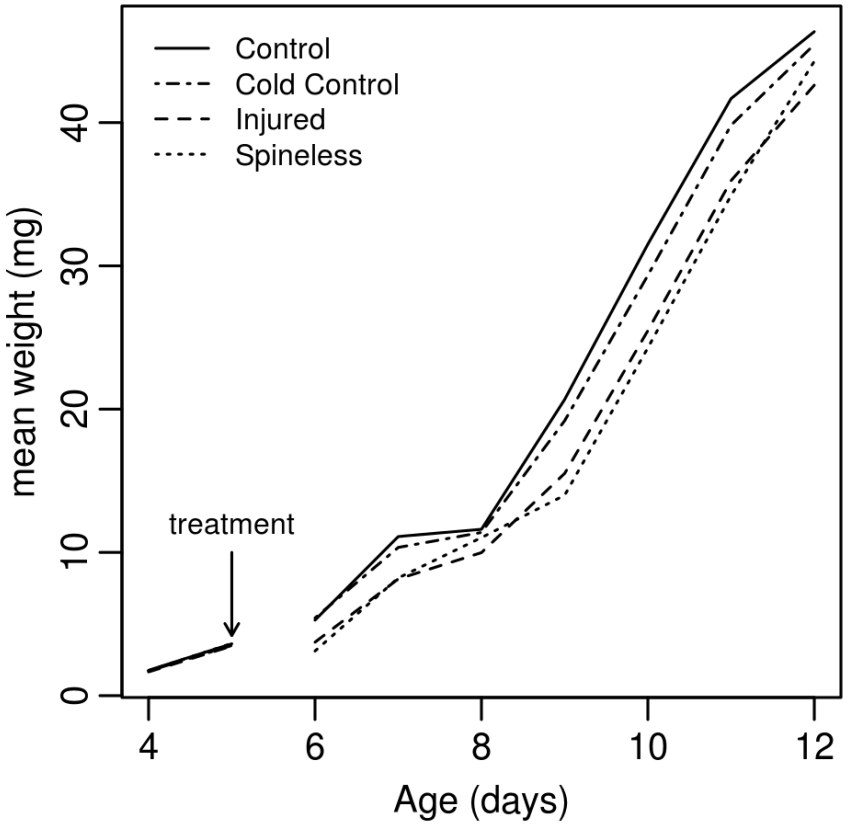
Interactions

Exemple réel d'interaction triple

poids de larves ~ temps * avant/après ablation d'épines * 4 types de traitements :

On veut savoir si il y a une différence de développement avant/après l'ablation et si cette différence dépend du type de traitement --> interaction triple (non significative ici)





Hautier et al. (in prep)

Interactions

Exemple simulé d'interaction triple : BACI design complet

On suit les populations d'une espèce chaque année sur 10 sites.
Après 10 ans on constate que les populations diminuent et on décide de mettre en place des mesures de gestion sur 5 sites.
On continue à suivre les 10 sites pendant 10 ans.

On veut savoir si l'effet de la gestion a été bénéfique par rapport aux sites non gérés.

BACI

BA : Before - After : dans ce cas avant/après la mise en place des mesures de gestion

CI : Control - Impact : sites contrôles sans gestion vs sites "impactés" avec de la gestion après une certaine date.

Interactions

Exemple simulé d'interaction triple : BACI design complet

Simulation des données

```
n <- 5
d <- data.frame(
    year = rep(-9:10, each = n*2),
    BA = factor(rep(c("before", "after"), each = 10*n*2),
                    levels = c("before", "after")),
    CI =rep(c("control", "impact"), times = 10*n*2)
)
X <- model.matrix(~ year * BA * CI, data=d)
B <- c(100, -1, 0, -5, -5, 0, 0, 3.5)

set.seed(1)
d$nb <- X %*% B + rnorm(200, 0, 5)
```

```
> d
    year     BA      CI        nb
1     -9 before control 105.86773
2     -9 before  impact 104.91822
3     -9 before control 104.82186
4     -9 before  impact 111.97640
5     -9 before control 110.64754
6     -9 before  impact  99.89766
7     -9 before control 111.43715
```

NB : La variable year correspond aux nombres d'années depuis le début de la gestion dans les sites "Impact"

Interactions

Exemple simulé d'interaction triple : BACI design complet

```
> mod <- lm( nb ~ year + BA + CI + year:BA + year:CI + BA:CI + year:BA:CI, data = d)
> summary(mod)

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           100.71122    1.23124  81.796  < 2e-16 ***
year                   -1.05920    0.23063  -4.593 7.91e-06 ***
BAafter                -2.26006    1.88781  -1.197  0.23271
CIimpact               -5.49685    1.74124  -3.157  0.00185 **
year:BAafter           -4.70638    0.32616 -14.429  < 2e-16 ***
year:CIimpact           0.08212    0.32616   0.252  0.80149
BAafter:CIimpact        2.83589    2.66977   1.062  0.28947
year:BAafter:CIimpact   3.01824    0.46127   6.543 5.33e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.684 on 192 degrees of freedom
Multiple R-squared:  0.9387,  Adjusted R-squared:  0.9364
F-statistic: 419.7 on 7 and 192 DF,  p-value: < 2.2e-16
```

Interactions

Exemple simulé d'interaction triple : BACI design complet

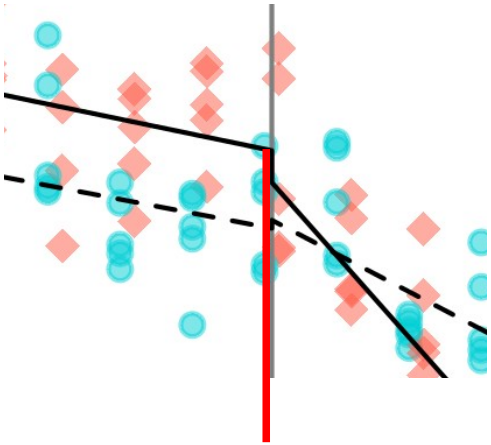
NB :

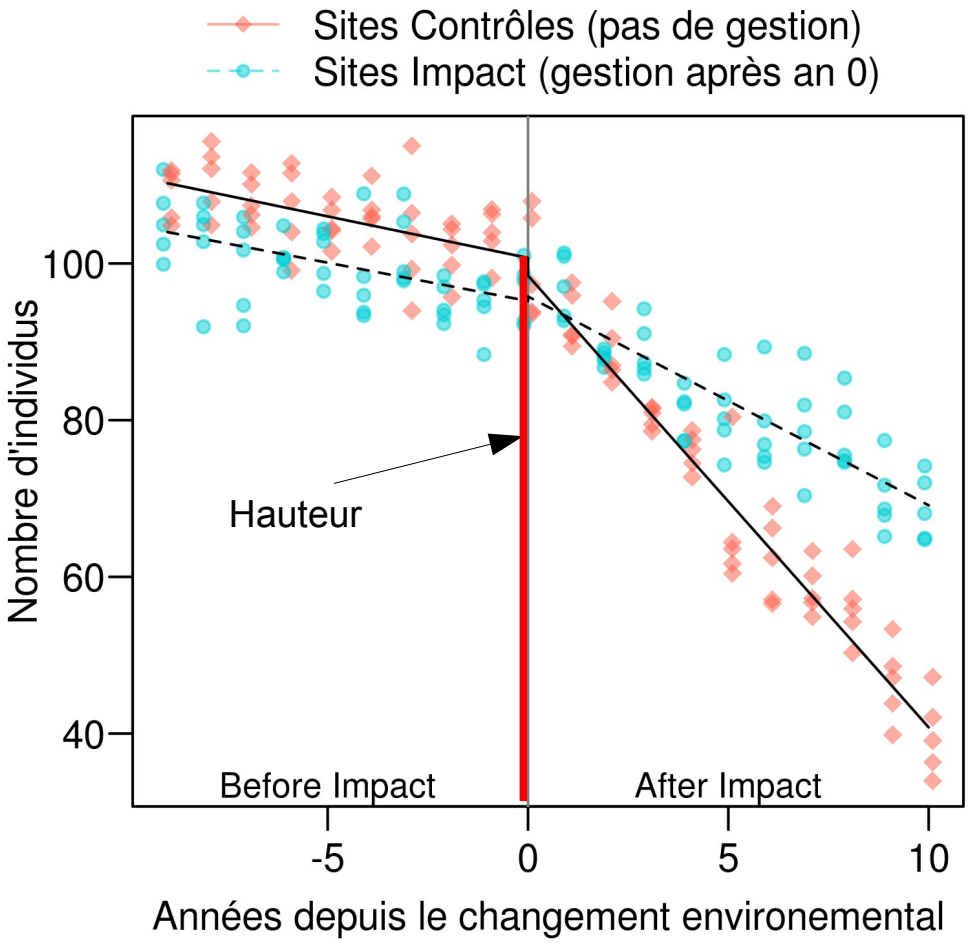
1) Pour ce genre d'études on suit les mêmes sites au cours du temps.
On devrait en tenir compte dans l'analyse
(ajout du site comme variable aléatoire --> voir plus loin, modèles mixtes)

2) Lorsqu'on compte un nombre d'individus, les résidus ne suivent en général pas une distribution normale et n'ont pas une variance homogène
--> il faudrait utiliser une distribution de Poisson --> voir plus loin GLM

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	**100.71122**	**1.23124**	**81.796**	**< 2e-16**	*******
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***

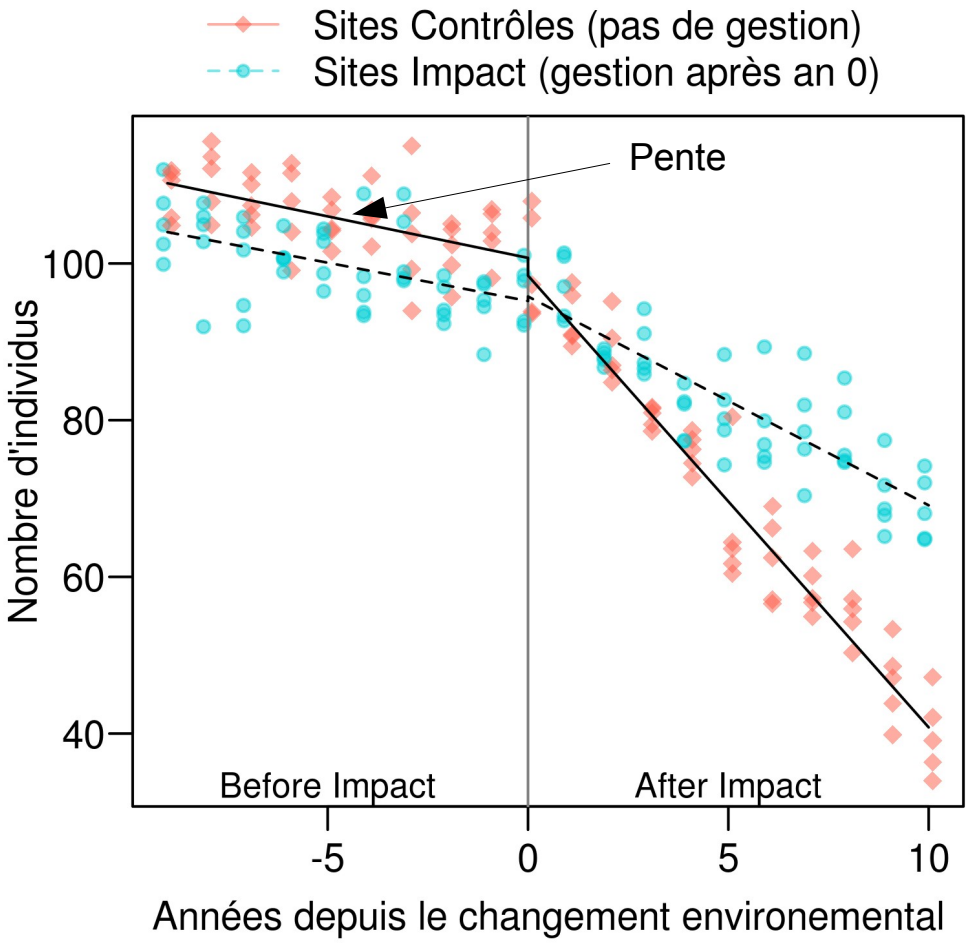




Intercept :
On estime qu'il y a en moyenne 100.71 individus dans les sites contrôles en l'an 0 (et avant la gestion mais çà n'a pas beaucoup de sens ici)

Interactions

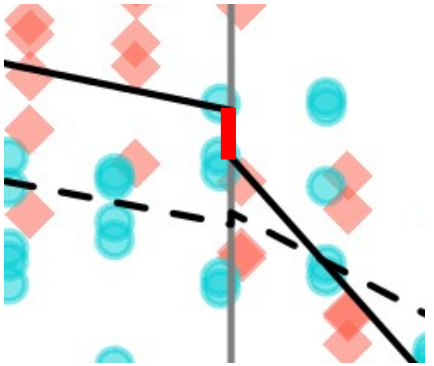
	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	**-1.05920**	**0.23063**	**-4.593**	**7.91e-06**	*******
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***

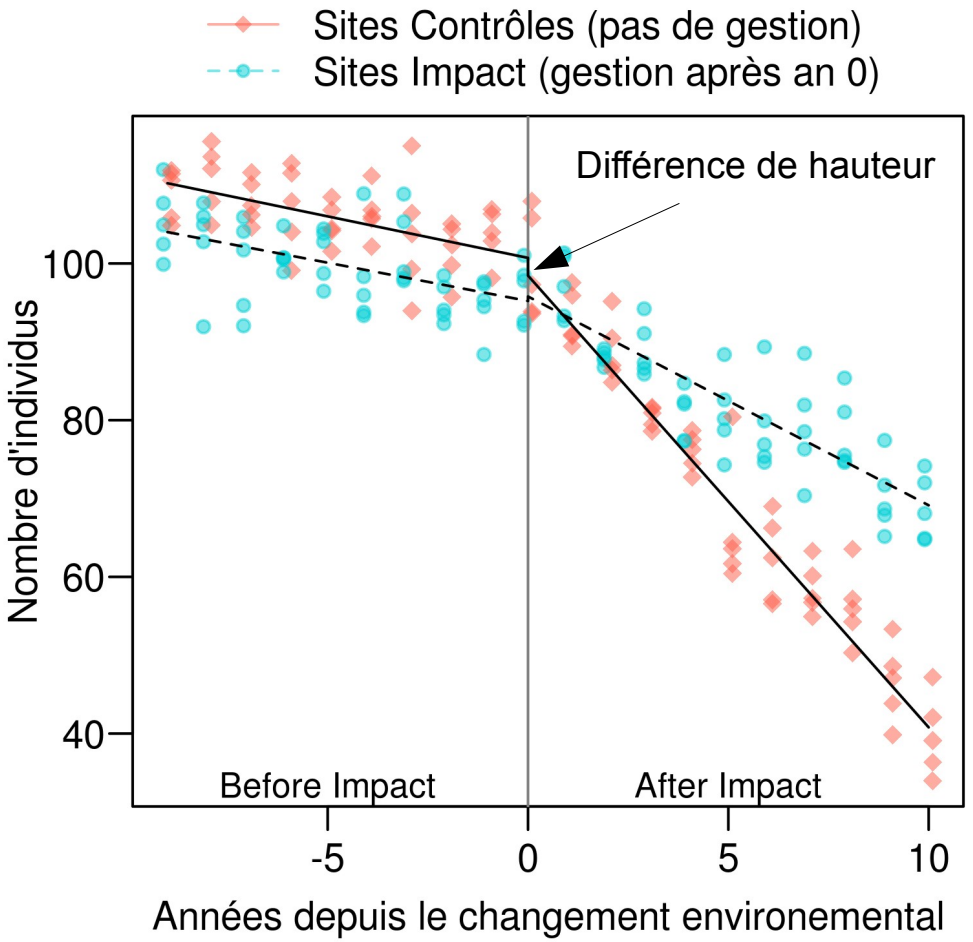


year :
On estime qu'on perdait en moyenne 1.06 individus par an sur les sites contrôles dans la période avant la gestion

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	**-2.26006**	**1.88781**	**-1.197**	**0.23271**	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***



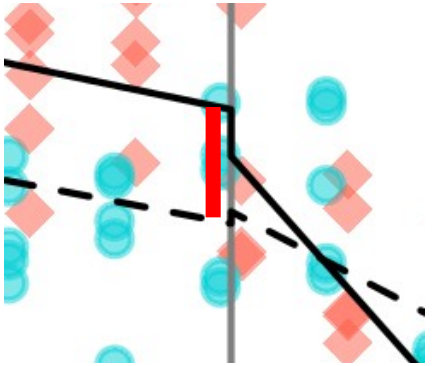


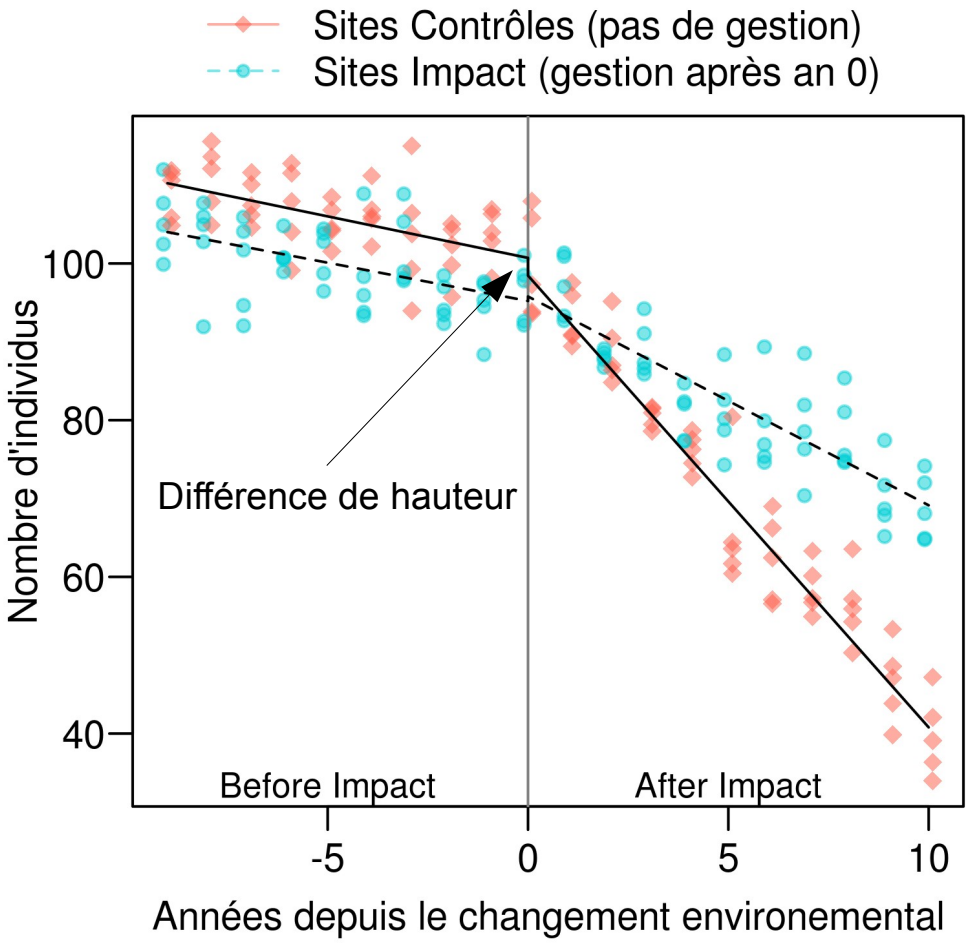
BAafter :
On estime qu'il y a en moyenne une différence de -2.26 individus en l'an 0, dans les sites contrôles entre avant et après la mise en place de la gestion.

--> ce paramètre n'a pas beaucoup de sens biologique (il n'est d'ailleurs pas significatif).
On pourrait donc dans ce cas enlever l'effet principal "BA" du modèle tout en gardant certaines de ses interactions.
On force alors les deux droites avant/après pour le contrôle à passer par le même point en l'an 0.

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	**-5.49685**	**1.74124**	**-3.157**	**0.00185**	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***



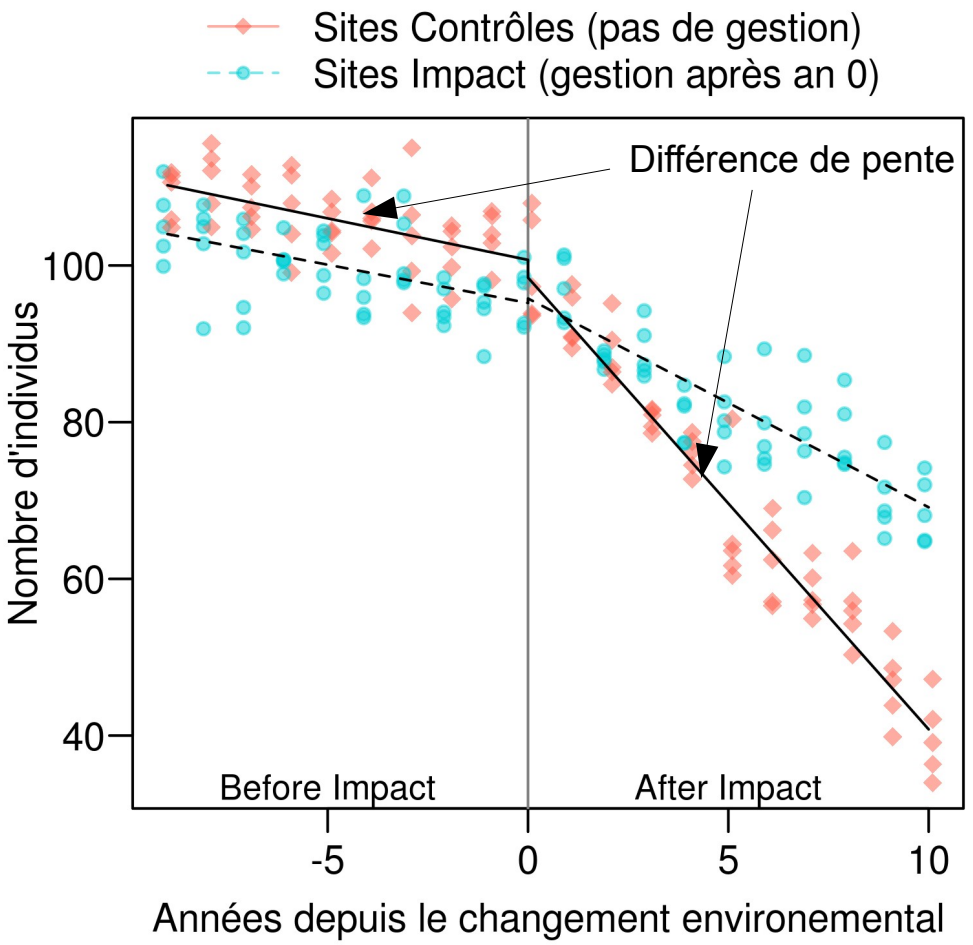


CIimpact:
On estime qu'il y a en moyenne une différence de -5.5 individus en l'an 0, entre les sites contrôle et les sites impact.

Si les sites ont bien été attribués au hasard, ceci ne devrait normalement pas arriver !!!
On pourrait imaginer qu'on a pris 5 sites d'une région pour les sites témoins et 5 sites dans une autre région pour les sites Impact/Traitement. A éviter absolument !!

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	**-4.70638**	**0.32616**	**-14.429**	**< 2e-16**	*******
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***

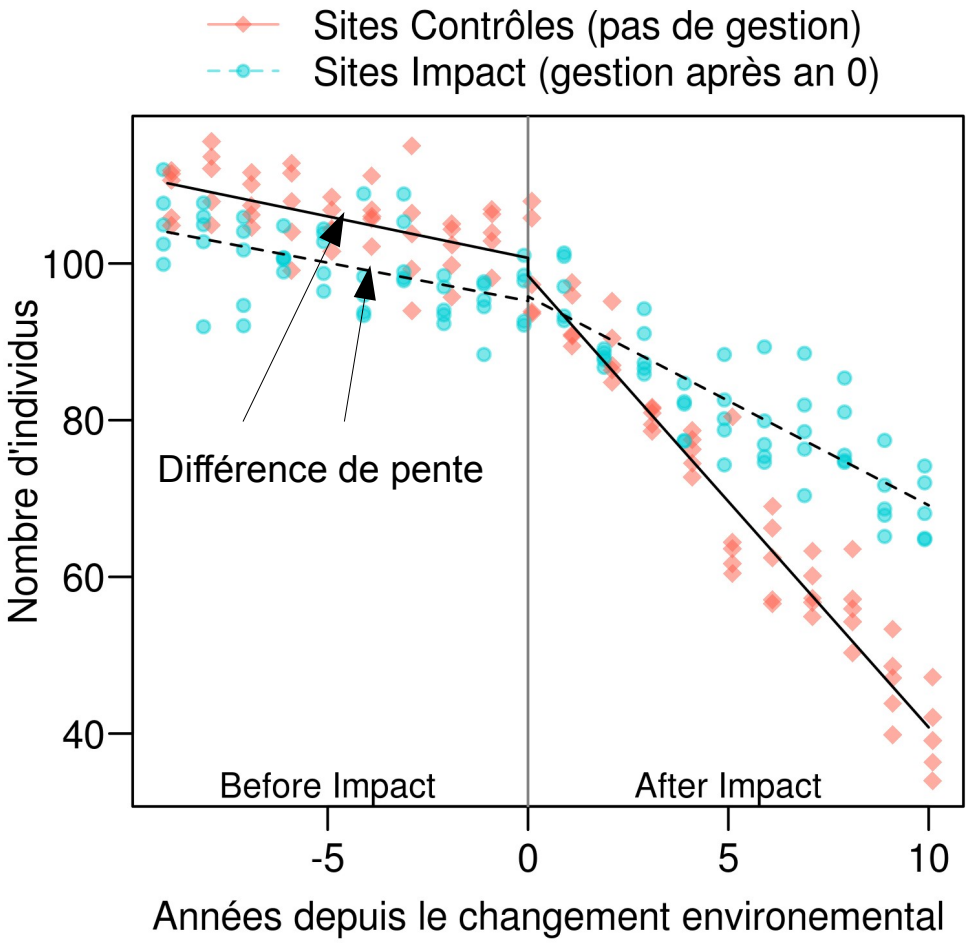


year :BAafter :
Après le début de la période de gestion, les sites contrôles ont perdu en moyenne 4.7 individus par an en plus des 1.05 qu'il perdaient déjà avant cette période. La pente de la droite contrôle est donc de -1.05 avant et de -1.05-4.7 = -5.75 après l'année 0.
year:BAafter représente donc la différence de pente entre avant et après l'année 0 pour les sites contrôle.

Il s'agit ici d'un cas (très) particulier où le début de la période d'impact coïncide avec un événement extérieur à la gestion qui a eu un impact sur l'ensemble des populations

Interactions

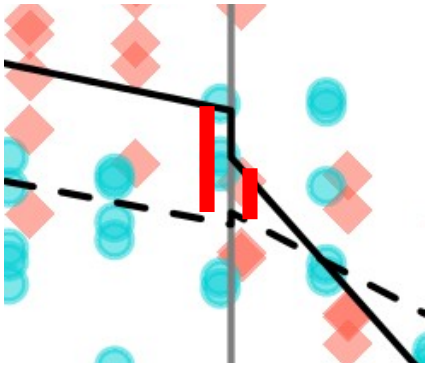
	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	**0.08212**	**0.32616**	**0.252**	**0.80149**	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***

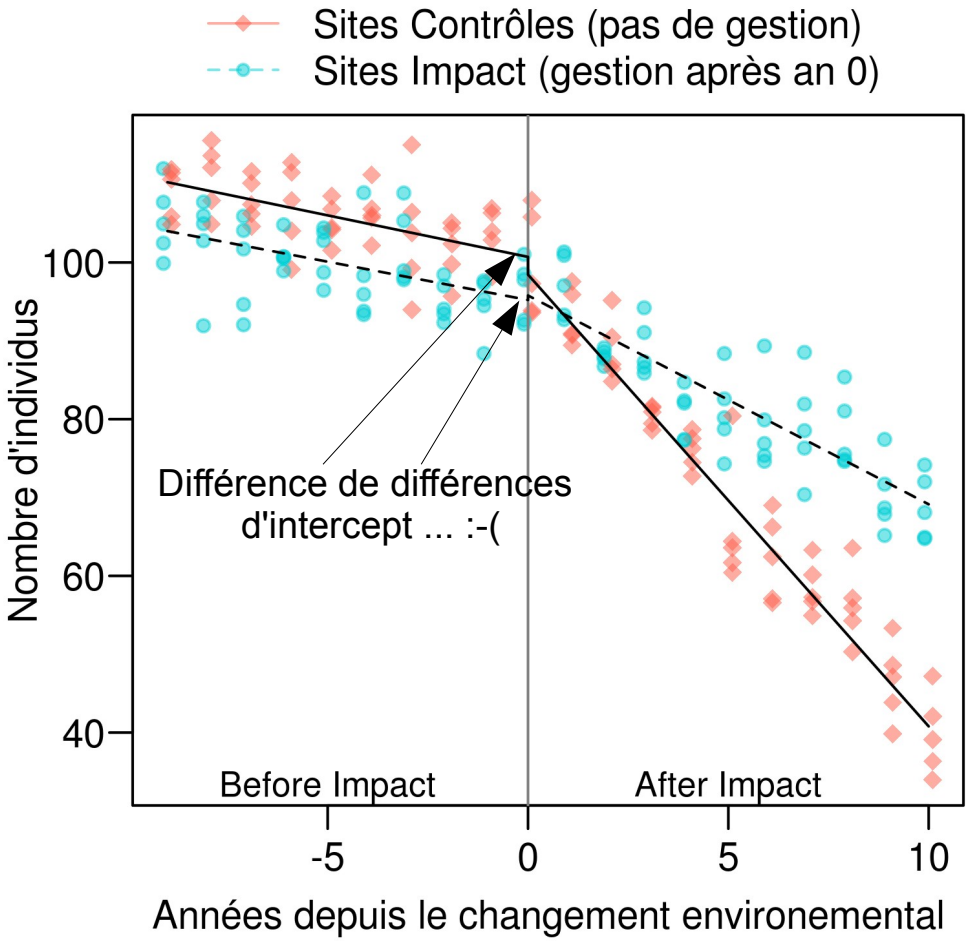


year:CIimpact :
La différence de pente avant l'Année 0 était seulement de 0.08 individus entre les sites contrôles et les sites "impact" qui à cette époque ne recevaient aucune mesure de gestion particulière. La perte annuelle d'individuas avant l'année 0 était donc de -1.05 pour les sites contrôles et de -0.97 pour les futur sites gérés.

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	**2.83589**	**2.66977**	**1.062**	**0.28947**	
year:BAafter:CIimpact	3.01824	0.46127	6.543	5.33e-10	***



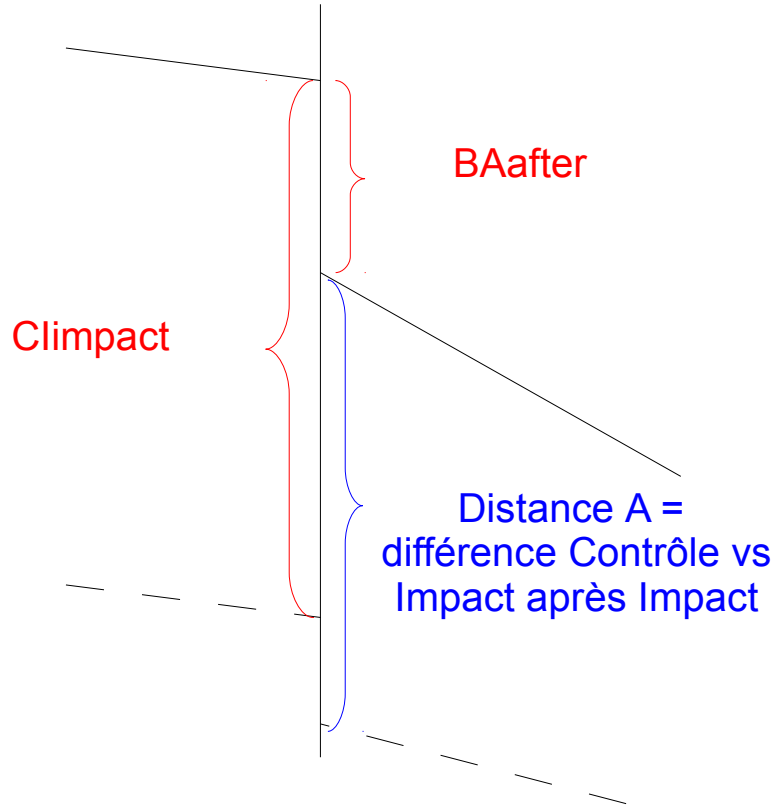


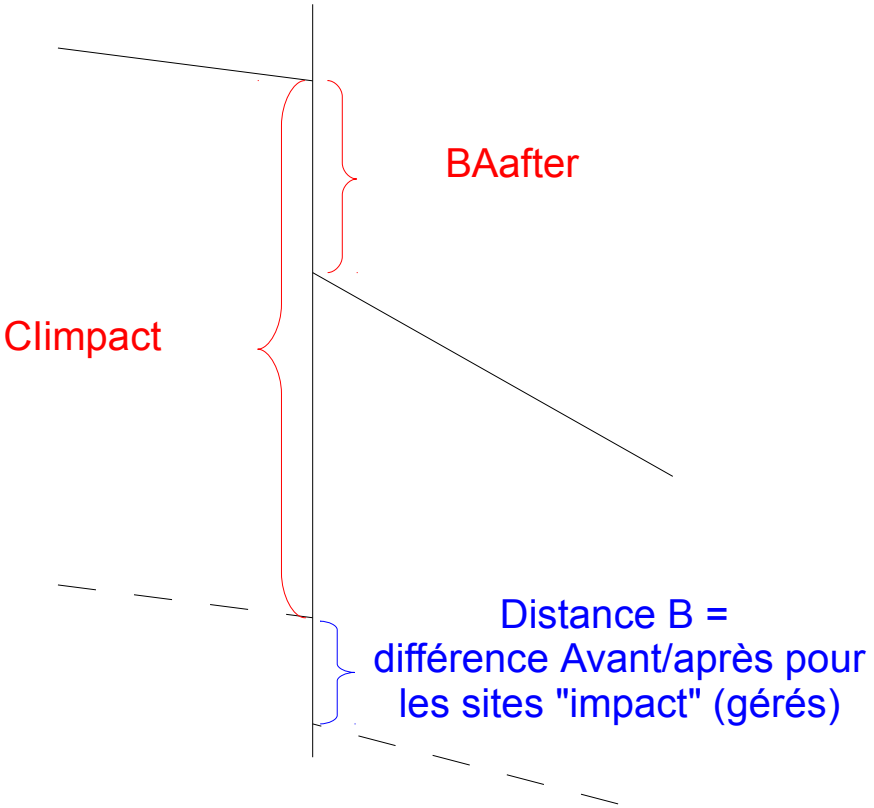
BAafter:CIimpact
Peut-être le plus difficile à cerner et dans ce cas précis sans grande signification biologique...

Ce paramètre permet de tester la question suivante : est-ce que la différence entre les sites contrôle et les sites impact est la même avant et après le début de la gestion ? Ou de manière équivalente, est-ce que les différences avant/après sont les mêmes pour les contrôles et les sites impact ?

Interactions

Deux interprétations équivanlentes de l'interaction BA x CI



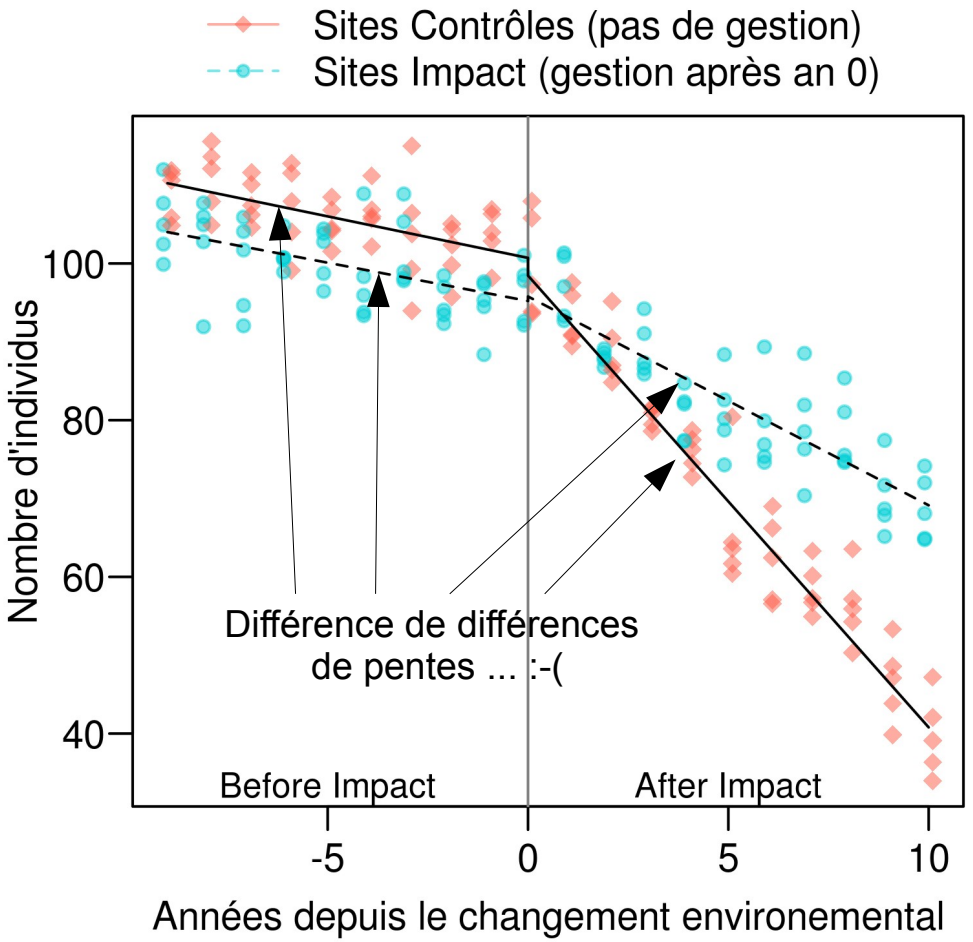


BAafter :CImpact = CImpact - Distance A

BAafter :CImpact = BAafter - Distance B

Interactions

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	100.71122	1.23124	81.796	< 2e-16	***
year	-1.05920	0.23063	-4.593	7.91e-06	***
BAafter	-2.26006	1.88781	-1.197	0.23271	
CIimpact	-5.49685	1.74124	-3.157	0.00185	**
year:BAafter	-4.70638	0.32616	-14.429	< 2e-16	***
year:CIimpact	0.08212	0.32616	0.252	0.80149	
BAafter:CIimpact	2.83589	2.66977	1.062	0.28947	
year:BAafter:CIimpact	**3.01824**	**0.46127**	**6.543**	**5.33e-10**	***



year :BAafter:CIimpact
Cette interaction permet de répondre à la question suivante (qui est la plus importante ici) : est-ce qu'il y a des différences de pentes entre avant et après les gestion, est-ce que ces différences sont du même ordre entre les sites contrôles et les sites gérés (la réponse étant clairement non) ?
Il estime la valeur suivante :
(pente impact après - pente impact avant) - (pente contrôle après - pente contrôle avant).
Pour obtenir la pente du groupe impact, à partir de l'an 0 on doit donc faire :
-1.06 - 4.71 + 0.08 + 3.02 = -2.67

```
mypalette(2,0.5,c=150)
par(mar = c(3,3,4,1), mgp = c(1.85, 0.6, 0), las = 1, cex = 0.9)
plot(y = d$nb, x = (d$year - (as.numeric(d$CI)-1.5)*0.2),
     pch = c(18, 20)[as.numeric(d$CI)],
     col = c("#FF5D4980", "#00CFD680")[as.numeric(d$CI)],
     cex = 1.1, xlab = "Années depuis le changement environemental",
     ylab = "Nombre d'individus")
abline(v=0, col = "grey50")
mtext(c("Before Impact", "After Impact"), 1, -1, at = c(-5,5) , cex = 0.8)

X1 <- cbind(1,
            c(-9:0, 0:10),
            rep(c(0,1), times = c(10,11)),
            0,
            c(-9:0, 0:10) * rep(c(0,1), times = c(10,11)),
            0,
            0,
            0)

X2 <- cbind(1,
            c(-9:0, 0:10),
            rep(c(0,1), times = c(10,11)),
            1,
            c(-9:0, 0:10) * rep(c(0,1), times = c(10,11)),
            c(-9:0, 0:10) * 1,
            rep(c(0,1), times = c(10,11)) * 1,
            c(-9:0, 0:10) * rep(c(0,1), times = c(10,11)) *1)

pred1 <- X1 %*% coef(mod)
pred2 <- X2 %*% coef(mod)

lines(y = pred1, x = X1[,2], lty = 1)
lines(y = pred2, x = X2[,2], lty = 2)

legend(x = "top", inset = -0.20, xpd = NA, bty = "n", lty = 1:2, pch = c(18,20),
       cex = 1, pt.cex = 1, col = c("#FF5D4980", "#00CFD680"),
       legend = c("Sites Contrôles (pas de gestion)", "Sites Impact (gestion après an 0)"))
```

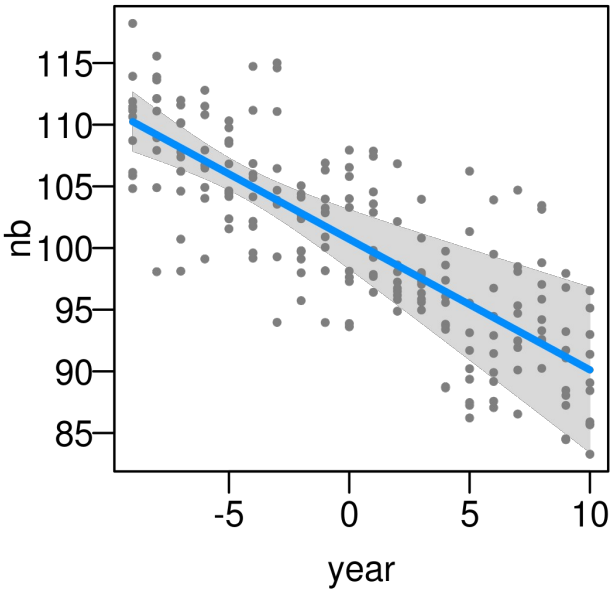
Code pour la représentation graphique...

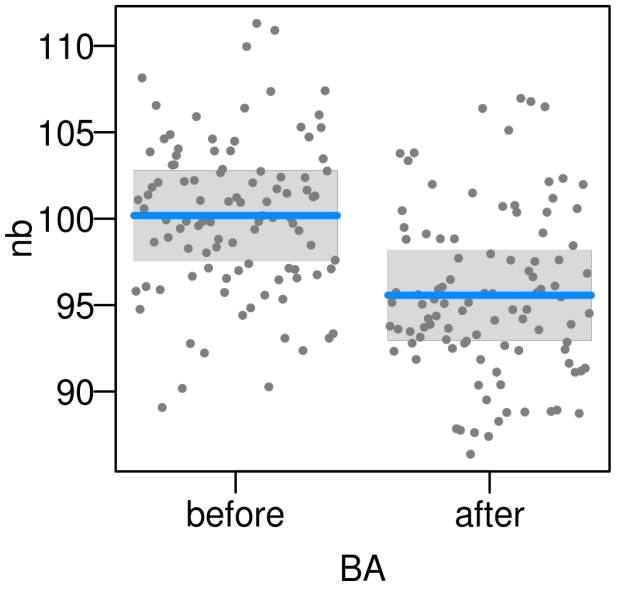
Interactions

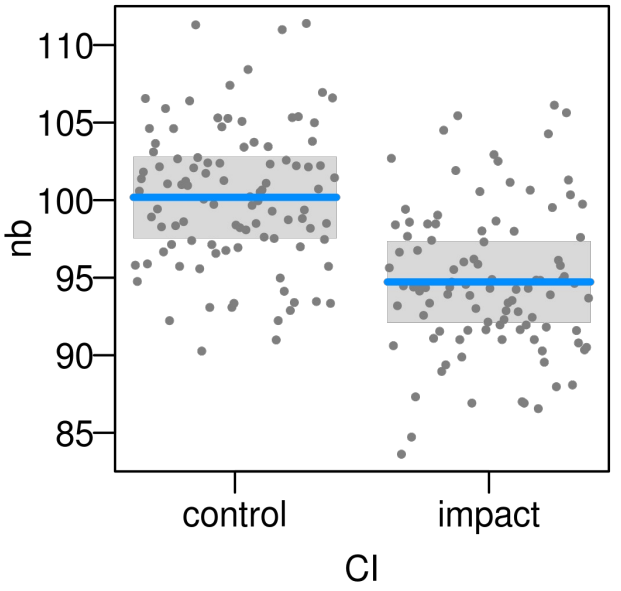
Représentation graphique avec visreg

Pas facile d'obtenir une bonne représentation...

```
library(visreg)
par(mfrow=c(1,3), mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod)
```





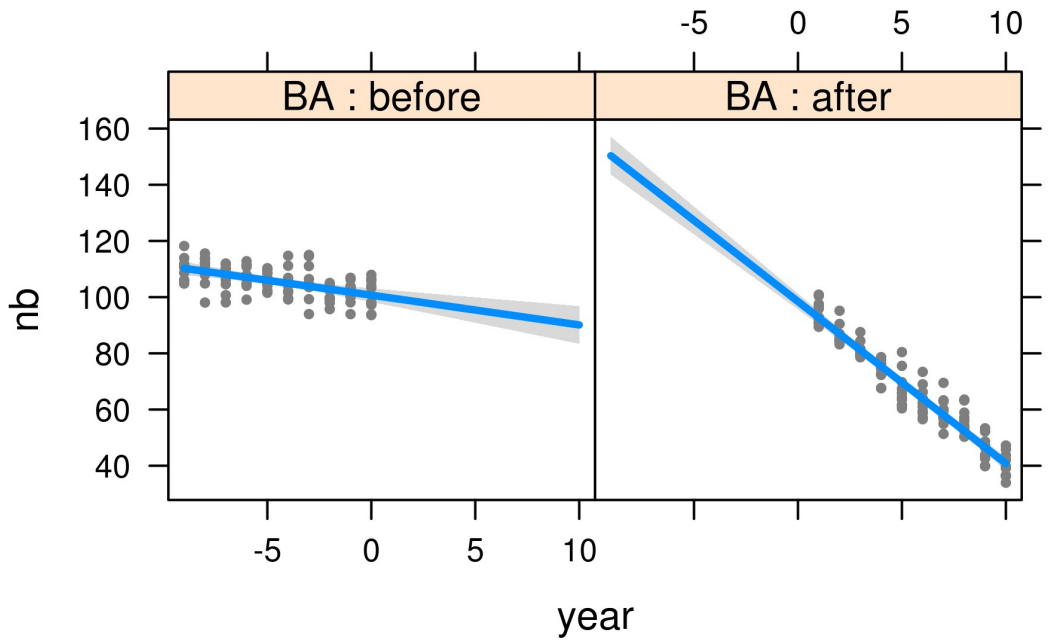


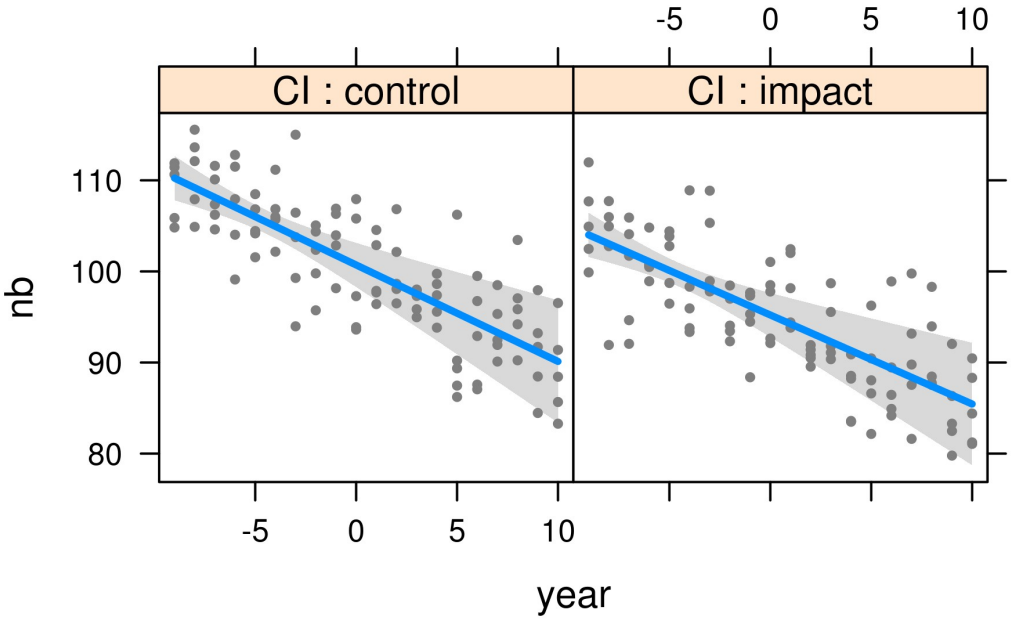
Interactions

Représentation graphique avec visreg

Pas facile d'obtenir une bonne représentation...

```
par(mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod, xvar = "year", by = "BA", strip.names = TRUE)
visreg(mod, xvar = "year", by = "CI", strip.names = TRUE)
```



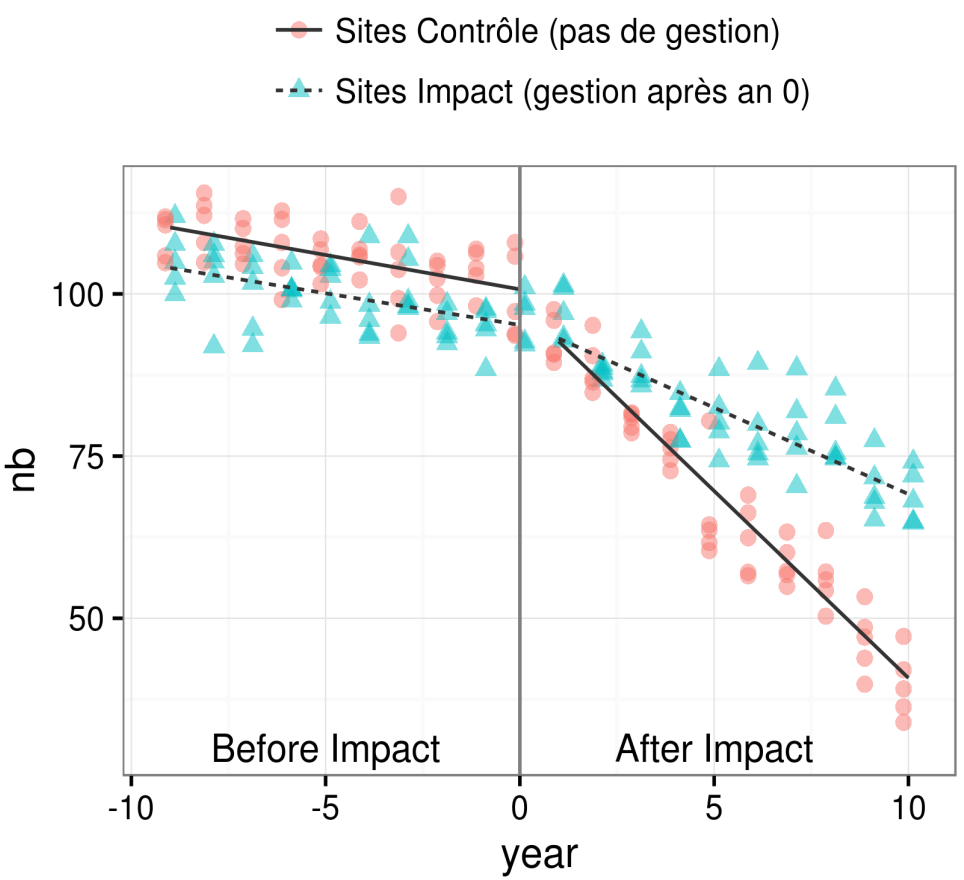


Interactions

Représentation graphique avec ggplot

Il faut ruser un peu...

```
d$treatment <- paste(d$BA, d$CI)
levels(d$CI) <- c("Sites Contrôle (pas de gestion)", "Sites Impact (gestion après an 0)")

ggplot(d, aes(x=year, y = nb, shape = CI, color = CI, group = treatment)) +
    geom_point(size = 2, alpha = 0.5,
               position = position_dodge(0.5)) +
    stat_smooth(aes(lty = CI), lwd = 0.5,
                color = "gray20",
                method = "lm", se = FALSE) +
    annotate("text", x = c(-5, 5),
             y = 30,
             label = c("Before Impact",
                       "After Impact")) +
    geom_vline(xintercept = 0, color = "gray50") +
    theme_bw() +
    theme(legend.position = "top",
          legend.direction = "vertical",
          legend.title=element_blank(),
          legend.key = element_rect(color = NA))
```



Danger : comparaison de modèles avec interaction !

Pour tester globalement des variables qualitatives à plus de 2 niveaux il faut utiliser des comparaisons de modèles emboîtés.

Typiquement, les logiciels de statistiques vous donnent un "tableau d'analyse de la variance" en vous permettant de choisir entre plusieurs "Sum of Square" : type I, type II, type III

Ces 3 approches sont toutes correctes mais testent des hypothèses parfois totalement différentes (pour une même variable explicative).

Il est parfois assez difficile de savoir quelles hypothèses sont testées exactement et il faut être particulièrement prudent avec les interprétations de tels tableaux en particulier en présence d'interactions.

Danger : comparaison de modèles avec interaction !

```
> mod <- lm( tomato ~ fertilizer + variety + fertilizer:variety, data=d)
> summary(mod)
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          10.5183     0.4439  23.695  < 2e-16 ***
fertilizer            0.4915     0.1812   2.712   0.0075 **
variety2              0.3163     0.6278   0.504   0.6151
variety3              4.5727     0.6278   7.284 1.96e-11 ***
fertilizer:variety2  -0.6366     0.2563  -2.484   0.0141 *
fertilizer:variety3   1.0556     0.2563   4.119 6.40e-05 ***
---
```

Si on utilise `drop1` comme on l'a vu précédemment, il ne teste que l'interaction du niveau le plus élevé.
`drop1` est très prudent !

```
> drop1(mod, test = "F")
                   Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                          472.94 184.25
fertilizer:variety  2     146.1 619.03 220.63  22.242 3.822e-09 ***
```

Danger : comparaison de modèles avec interaction !

Pour obtenir les SS de type I :

```
> anova(mod)
                     Df  Sum Sq Mean Sq F value    Pr(>F)
fertilizer            1  119.49  119.49  36.383 1.302e-08 ***
variety               2 1732.82  866.41 263.805 < 2.2e-16 ***
fertilizer:variety    2  146.10   73.05  22.242 3.822e-09 ***
Residuals           144  472.94    3.28
-
```

Le type I teste les effets séquentiellement : chaque variable est testée après avoir enlevé l'effet de la précédente.

L'ordre des variables change donc les résultats sauf quand les variables sont parfaitement indépendantes (pex design expérimental balancé)

Danger : comparaison de modèles avec interaction !

Pour obtenir les SS de type II :

```
> library(car)
> Anova(mod)
                    Sum Sq  Df F value    Pr(>F)
fertilizer          119.49   1  36.383 1.302e-08 ***
variety            1732.82   2 263.805 < 2.2e-16 ***
fertilizer:variety  146.10   2  22.242 3.822e-09 ***
Residuals           472.94 144
```

Le type II teste toutes les variables après avoir enlevé l'effet de toutes les autres variables. L'ordre des variables ne change donc rien.

Le type II teste les effets principaux après avoir enlevé l'interaction (respect des règles de marginalité)

NB : dans ce cas, Type I et Type II donnent les mêmes résultats car on a un design parfaitement balancé et des variables explicatives parfaitement indépendantes

Danger : comparaison de modèles avec interaction !

Pour obtenir les SS de type III :

```
> library(car)
> Anova(mod, type = "III")
Anova Table (Type III tests)

Response: tomato
                     Sum Sq  Df  F value    Pr(>F)
(Intercept)        1843.91   1 561.4360 < 2.2e-16 ***
fertilizer           24.15   1   7.3545  0.007504 **
variety             217.37   2  33.0919 1.496e-12 ***
fertilizer:variety  146.10   2  22.2421 3.822e-09 ***
Residuals           472.94 144
```

NB : Type II et Type III donnent les mêmes résultats quand il n'y a pas d'interaction (ici les résultats sont donc différents).
L'interaction du plus haut niveau donne toujours les mêmes résultats avec les 3 méthodes

Le type III ne respecte pas les règles de marginalité ce qui a des conséquences parfois inatendues.
Pex si vous changez l'ordre des variétés (ou leur nom) les tests pour `fertilizer` et `variety` vont changer ! (ce qui n'est pas le cas avec le type II)
Pour fertilizer on teste la pente de la variété de référence uniquement
Pour variety on teste les différences entre variété uniquement quand la dose est 0

Danger : comparaison de modèles avec interaction !

Il est de manière générale déconseillé d'utiliser ces tableaux à moins de comprendre quelle hypothèse est évaluée par chaque test.

En général, si l'interaction AxB est significative, il n'y a que peu d'intérêt à tester si A ou B sont significatifs.
Le fait que A soit significatif va dépendre des valeurs de B et/ou de la position de l'intercept.
Si AxB est significatif, A et B sont importants pour prédire y

En général, il vaut mieux respecter la règle de marginalité :
dans les comparaisons de modèles, si on teste un effet principal (main effect) alors il faut ignorer également les interactions qui comprennent cet effet.

Les tests de type II respectent cette règle de marginalité alors que les tests de type III ne le font pas.

Danger : comparaison de modèles avec interaction !

Conclusions

Ne pas utiliser les type I (`anova(model)`) :
l'ordre des variables change les résultats (sauf designs expérimentaux balancés)

Ne pas utiliser le type III (défaut dans de nombreux logiciels) à moins de savoir ce que vous faites

En présence d'interactions, ne pas interpréter les effets principaux qui les composent (comme `drop1` le fait)

Si on veut quand même un tableau complet d'analyse de la variance (souvent assez pratique) utiliser plutôt les tests de type II (`car::Anova`)

Si vous voulez être certain de ce que vous faites : construisez vous-même les modèles correspondant aux hypothèses que vous voulez tester et comparez les avec `anova(model1, model2)`

Generalized Linear Models

Generalized Linear Models

Les Generalized Linear Models (GLM) sont une généralisation des General Linear Models (LM) vus jusqu'à présent.

Le côté "X" (= variables explicatives = "linear predictor") de l'équation ne change pas

Deux choses en plus :

1) la **distribution** qui peut être autre que normale
2) une **fonction de lien** établissant la relation entre la variable dépendante et la combinaison de variables explicatives (=linear predictor)

La méthode d'estimation n'est pas la même (**Maximum de vraisemblance** vs Moindres carrés)

Les tests d'hypothèse nulle sont légèrement différents mais s'interprètent de la même manière.

Generalized Linear Models

Dans un lm(), le modèle prédit directement les valeurs de $\hat{Y}$.
Y suit toujours une distribution Gaussienne de moyenne $X\beta$ et variance σ^2

$\hat{Y}$ = Valeurs prédites de Y

$$\hat{Y} = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots$$

"linear predictor"

$$\hat{Y} = X\beta$$

Notation matricielle équivalente

Valeurs observées de Y

$$Y \sim Normale(X\beta, \sigma^2)$$

$$Y \sim X\beta + \varepsilon \quad avec\ \varepsilon \sim Normale(0, \sigma^2)$$

Notation équivalente

Dans un glm(), le modèle prédit g($\hat{Y}$) où g() est une "fonction de lien" qui transforme Y (pex : `log`, `logit`,...).
$g^{-1}()$ est une fonction qui inverse le résultat de g() (pex `exp`, `plogis`)
Y peut suivre d'autres distributions : Gaussienne, Poisson, Binomiale,...

g() = "link function"

$$g(\hat{Y}) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots$$

$\hat{Y} = g^{-1}(g(\hat{Y}))$

$$g(\hat{Y}) = X\beta$$

$g^{-1}()$ = "inverse link function"

$$Y \sim Distribution(g^{-1}(X\beta), autres\ paramètres\ facultatifs)$$

Generalized Linear Models

GLM à distribution Gaussienne (=Normale)

Ce sont les modèles vus jusqu'à présent mais estimés avec la méthode du maximum de vraisemblance (comme tous les GLM) au lieu de la méthode des moindre carrés.

Les GLM gaussiens donnent les mêmes résultats que les LM moyennant l'hypothèse que les résidus ont réellement une distribution gaussienne.

Les LM n'ont pas besoin de cette hypothèse pour l'estimation des paramètres, mais uniquement pour l'inférence.

C'est une des raisons pour lesquelles il est recommandé d'utiliser `lm()` et non pas `glm()` quand vous voulez estimer des modèles à distribution normale
On le fera ici juste pour la comparaison ...

Generalized Linear Models

GLM à distribution Gaussienne (=Normale)

On l'utilise pour modéliser des variables continues (parfois après transformation des y).

Fonction de lien par défaut g() = identity

Fonction inverse $g^{-1}()$ = identity

La fonction identité ne change rien et donc l'inverse non plus, on peut donc résumer ce modèle comme :

$$Y \sim \text{Normale}\ (X\beta, \sigma^2)$$

Ce qui est strictement identique (dans le cas gaussien) à :

$$Y \sim X\beta + Normale(0, \sigma^2)$$

Generalized Linear Models

GLM à distribution de Poisson

On l'utilise pour modéliser des **entiers positifs**.
Typiquement les **données de comptage** suivent une distribution de Poisson. Lorsque les nombres comptés sont grands, cette distribution tend vers une distribution normale.

On l'utilisera également pour modéliser des **tables de contingence** (association entre variables qualitatives)

Fonction de lien par défaut g() = log

Fonction inverse $g^{-1}()$ = exp

$$log(\hat{Y}) = X\beta$$
$$Y \sim Poisson(\ exp(X\beta))$$

Generalized Linear Models

GLM à distribution de Poisson

Dans une distribution de Poisson, on estime pas la variance, elle est fixée comme étant égale à la moyenne $X\beta$ avec $\varphi=1$:

$$var(Y) = \varphi * X\beta$$

Le paramètre φ est appelé paramètre de surdispersion ("overdispersion") ou "scale parameter".
Il est fixé à 1 dans la distribution de Poisson

Dans certains cas la variance est plus élevée que celle prévue par la loi de Poisson.
φ est en réalité > 1 alors qu'il est fixé à 1

Ce qui a des conséquence dramatiques sur les inférences.
Il faudra vérifier systématiquement si cette hypothèse est respectée !

Generalized Linear Models

GLM à distribution Binomiale

On l'utilise pour modéliser une autre forme de données de comptage:

1) une **proportion** p de la forme nombre de succès/nombre d'essais

2) des **données binaires** qui sont un cas particulier du cas précédent où le nombre d'essais N = 1

Lorsque N est grand et que p est proche de 0.5, cette distribution tend vers une distribution normale.

Fonction de lien par défaut g() = logit(p) = log(p/(1-p))
Fonction inverse $g^{-1}()$ = invlogit(a) = exp(a) / (1 + exp(a)

Nom officieux
Dans R : fonction `plogis()`

$$logit(\hat{Y}) = X\beta$$

Proportion

$$Y \sim Binomiale(invlogit(X\beta), N)$$

N = nombre total d'observations

$$Y \sim Binomiale(\exp(X\beta) / (1 + \exp(X\beta), N)$$

GLM à distribution Binomiale

Ici aussi la variance est une fonction de la moyenne avec un paramètre de surdispersion φ fixé à 1 :

$$\text{Var}(Y) = \varphi\, Np(1-p)$$

La surdispersion aura également des conséquences importantes sur les inférences.
Pour des données binaires, φ est toujours à peu près = 1

Generalized Linear Models

GLM à distribution Binomiale

Attention : toutes les données en % ou en proportion n'ont pas une distribution binomiale.
Pex : % de surface d'une feuille attaquée par un champignon (= rapport de 2 variables quantitatives continues).

Dans ce cas on fait typiquement un modèle gaussien, on examine les résidus et si la distribution n'est pas gaussienne, on fait des transformations de variables
pex : `asin(sqrt(y))`, `logit(y+n)`,...

Ou on utilise la donnée brute (sans ratio, pex surface de champignon) et un offset (pex surface de la feuille) comme moyen de standardisation.

voir pex discussion ici : Warton, D.I., Hui, F.K., 2011. The arcsine is asinine: the analysis of proportions in ecology. Ecology 92, 3–10.

Autres distributions pour les GLM

Les distributions Gaussienne, de Poisson et Binomiale permettent de modéliser une grande variété de problèmes courants et ce sont de loin les plus utilisées.
On se concentrera sur ces distributions ici.

Certaines autres distributions sont proches des distribution de Poisson et Binomiale mais avec un paramètre de dispersion en plus : distribution **Négative Binomiale**, **Beta Binomiale**, ...

D'autres distributions sont adaptées à d'autres types de donnes comme les analyses de survie (temps écoulé jusqu'au décès) : (distribution **Gamma**, **exponentielle**, etc...)

Generalized Linear Models

Autres distributions pour les GLM

Les modèles **multinomiaux** sont une généralisation des modèles Binomiaux permettant de modéliser des variables qualitatives avec plus de 2 niveaux.
On peut souvent obtenir des résultats similaires avec des analyses de tables de contingence avec une distribution de Poisson

Il existe aussi la possibilité d'utiliser des "**mixtures**" de distributions.

Par exemple les "**Zero Inflated Poisson models**" (ZIP) utilisent une combinaison de distribution Binomiale et de Poisson pour modéliser des données de comptage avec un excès de 0 par rapport à la distribution de Poisson.

NB : En cas d'excès de 0 pour des données de comptage on peut aussi souvent soit transformer la variable en présence/absence soit regrouper les unités de mesures pour augmenter le nombre moyen d'individus

Autres fonctions de lien

Pour une même distribution, il existe parfois la possibilité d'utiliser d'autres fonctions de lien que la fonction par défaut ("canonic link").

Les fonctions de lien par défaut correspondent typiquement à la relation qu'on est susceptible d'observer avec certains types de données
(pex : courbe sigmoïde pour des données binomiales)

En pratique on peut essayer différentes fonction de lien et on **compare la qualité d'estimation des données** (déviance, le R^2, le RMSE,...) des différents modèles pour choisir la fonction qui ajuste le mieux les données.

Mais dans la grande majorité des cas, on se contente de la fonction canonique par défaut qui est généralement la plus adaptée.

Generalized Linear Models

Autres fonctions de lien

Liste des familles et des fonctions de lien disponibles pour la fonction glm() de R. Les fonction de lien par défaut sont indiquées parla lettre C.

Les familles quasi, quasibinomial et quasipoisson ne sont pas à proprement parler des distribution, mais des méthodes permettant d'estimer le degré de surdispersion dans certains modèles. Taper ?family pour plus d'info.

family	identity	log	logit	probit	cloglog	inverse	1/mu^2	sqrt	cauchit
gaussian	C	x				x			
poisson	x	C						x	
binomial		x	C	x	x				x
Gamma	x	x				C			
inverse.gaussian	x	x				x	C		
quasi	C	x	x	x	x	x	x	x	
quasibinomial		x	C	x	x				x
quasipoisson	x	C						x	

Autres fonctions de lien

Si on veut savoir à quoi correspond une fonction de lien et son inverse, on peut taper par exemple :

```
> make.link("cloglog")
$linkfun
function (mu)
log(-log(1 - mu))
<environment: namespace:stats>

$linkinv
function (eta)
pmax(pmin(-expm1(-exp(eta)), 1 - .Machine$double.eps), .Machine$double.eps)
<environment: namespace:stats>
(...)
```

= -exp(-exp(eta))+1 sauf quand on est très proche de 0 ou 1

On peut utiliser les deux slots directement comme une fonction :

```
> make.link("cloglog")$linkfun(0.4)
[1] -0.671727
> make.link("cloglog")$linkinv(-0.671727)
[1] 0.4
> -exp(-exp(-0.671727))+1
[1] 0.4
```

Generalized Linear Models

Inférence

```
> modglm <- glm(y ~ x, family = poisson)
> summary(modglm)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2208     0.1527   7.993 1.32e-15 ***
x            -0.1908     0.0290  -6.578 4.77e-11 ***
```

Test de Wald : rapport entre un paramètre et son erreur standard : suit asymptotiquement une loi normale.

Paramètre +- 1.96 * se donne l'intervalle de confiance à 95 %

```
> drop1(modglm, test = "Chisq")

       Df Deviance    AIC    LRT Pr(>Chi)
<none>      79.884 193.16
x       1  131.832 243.10 51.948  5.7e-13 ***

> Anova(modglm) # même résultat que drop1
```

Test de rapport de vraisemblance : Comparaison de modèles emboîtés Equivalent au test F de l'Anova. Suit asymptotiquement une distribution Chi carré.

L'approximation est normalement meilleure que dans le test de wald

Le test de Wald et les comparaisons de modèles emboîtés ne donnent plus les même résultats contrairement aux modèles linéaires classiques

On peut aussi utiliser la fonction anova pour comparer deux modèles emboîtés

Inférence : Test de Rapport de vraissemblance

Likelihood Ratio Test = LRT

La statistique utilisée pour calculer la déviance peut être utilisée pour comparer n'importe quels modèles emboîtés et suit asymptotiquement une distribution Chi carré avec pour degrés de liberté la différence de nombre de paramètres entre les deux modèles.

$$LRT = 2*(\log(L_{modèle\,1}) - \log(L_{modèle\,2})) = Deviance_{modèle\,1} - Deviance_{modèle\,2}$$

Generalized Linear Models

Test de Rapport de vraisemblance

$$LRT = 2*(\log(L_{modèle\,1}) - \log(L_{modèle\,2})) = Deviance_{modèle\,1} - Deviance_{modèle\,2}$$

```
> modglm <- glm(y ~ x, family = poisson)
> nullmod <- glm(y ~ 1, family = poisson)

> # anova(nullmod, modglm, test = "Chisq")
> anova(nullmod, modglm, test = "LRT") # equivalent
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ x
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        74    131.832
2        73     79.884  1   51.948  5.7e-13 ***


> (dfnullmod <- length(coef(nullmod)))
[1] 1
> (dfmodglm <- length(coef(modglm)))
[1] 2
> LR <- (deviance(modglm) - deviance(nullmod))

> 1-pchisq(q = abs(LR), df = abs(dfmodglm - dfnullmod))
[1] 5.699885e-13
```

Inférence

NB : toutes les inférences paramétriques pour les GLM sont donc approximatives en particulier quand on a peu de données.

C'est une tendance générale : plus on va vers des modèles complexes (modèles mixtes, GAM,...) moins les inférences sont fondées sur des bases mathématiques solides.

Il faut se rappeler qu'il ne 'agit que d'un outil d'aide à la décision (est-ce que j'ai suffisamment de données pour estimer un paramètre?)

"*All models are false but some models are useful*" (G. Box)

"*It is better to be able to say something approximate about the right model rather than something very precise about the wrong model*" (S.Wood)

Est-ce que le modèle prédit bien les observations ?

Pseudo R^2

On peut calculer une sorte de R^2 pour les GLMs :
R^2 d'une régression entre les valeurs prédites et observées.

Il représente le % de variance expliquée par le modèle
(NB : forcément pour un modèle binaire il sera souvent assez faible même pour un très bon modèle ...)
voir fonctions `pseudoRsq()` et `diagplot()` dans `mytoolbox.R`

Compris entre 0 et 1

Quand $R^2 = 1$ le modèle prédit parfaitement les données
Mais çà ne veut pas dire que le résultat est transposable à d'autres jeux de données (overfitting)

Est-ce que le modèle prédit bien les observations ?

Root Mean Squared Error - RMSE

Racine carrée de la moyenne des résidus au carré

~ erreur moyenne de la prédiction (mêmes unités que y)

```
RMSE <- sqrt(mean((obs-fit)^2))
```

Interprétation biologique facile !

Mean Absolute Error - MAE

Moyenne des résidus en valeur absolue

= erreur moyenne de la prédiction (mêmes unités que y)

```
MAE <- mean(abs(obs-fit))
```

Deviance

Interprétation biologique difficile !

Écart de vraisemblance entre le modèle et un modèle prédisant parfaitement les données.

Pas d'unités, interprétation biologique plus difficile !

Quand RMSE, MAE ou Deviance = 0, le modèle prédit parfaitement les données

Mais çà ne veut pas dire que le résultat est transposable à d'autres jeux de données (overfitting)

Est-ce que le modèle prédit bien les observations ?

Overfitting

NB : quand le nombre de paramètres (nombre de variables explicatives) augmente :

le R^2 ne peut qu'augmenter
les RMSE, MAE et déviance ne peuvent que diminuer

Quand le nombre de paramètres = nombre d'observations
$R^2 = 1$
RMSE = MAE = déviance = 0
Toujours !

<u>On ne peut pas les utiliser pour comparer des modèles qui n'ont pas le même nombre de paramètres !</u>
--> utiliser des AIC dans ce cas

Est-ce que le modèle prédit bien les observations ?

AIC = Aikaike Information Criterion

NB : On reverra plus loin en détail l'utilisation et la philosophie derrière les AIC

La déviance ne tient pas compte du problème de sur-ajustement des données (overfitting) quand on a trop de variables explicatives/paramètres à estimer par rapport au nombre de données.

L'AIC est un critère d'information basé sur la vraisemblance qui pénalise les modèles avec trop de paramètres à estimer.
Il existe aussi une version corrigée pour les petits échantillons : l'AICc

$$AIC = -2 \log(L) + 2k$$

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

Avec :
k = nombre de paramètres
n = nombre de données
L = vraisemblance du modèle

NB : On devrait toujours utiliser l'AICc dont la valeur tend vers celle de l'AIC quand n est grand

Est-ce que le modèle prédit bien les observations ?

AIC = Aikaike Information Criterion

L'AIC seul est inutile. Il s'agit d'une mesure relative

Pour un jeu de données et un ensemble de modèles pour ce jeu de données, ceux qui ont un AIC plus petits sont "meilleurs que les autres" (voir + loin pour + de détails)

Les AIC permettent de comparer également des modèles non emboîtés (ie dont les paramètres ne sont pas un sous-ensemble des paramètres de l'autre modèle)

ATTENTION :
Les AIC (comme la déviance) ne peuvent comparer que des modèles basés exactement sur le même jeu de données du côté des variables dépendantes (Y).
On ne peut donc pas comparer des modèles avec des transformations différentes des Y ou avec des nombres de données différentes (attentions aux NA dans les x!)

Est-ce que le modèle prédit bien les observations ?

Area Under the Receiver Operating Curve
AUC ou AUROC

Pour les données binaires uniquement !
Estimation de la capacité du modèle à discriminer entre les présences et les absences.

AUC = 1
le modèle a un pouvoir discriminant parfait.
Toutes les présences prédites sont en réalité des présences
Aucune des présences prédites ne sont en réalité des absences

AUC =0.5
Le modèle ne prédit pas mieux les présences que le hasard
AUC = 0
Toutes les présences prédites sont des absences (çà n'arrive jamais)

Mêmes problèmes d'overfitting que R^2, RMSE, MAE, deviance,....

Est-ce que le modèle prédit bien les observations ?

Area Under the Receiver Operating Curve
AUC ou AUROC

Règles arbitraires :
AUC 0.5 - 0.7 : mauvais
AUC 0.7-0.8 : acceptable
AUC 0.8-0.9 : bon
AUC>0.9 : excellent

Également sensible à l'overfitting...
--> Idéalement évaluée sur un jeu de données indépendant...

Calcul : proportion de présences observées dont la probabilité prédite est plus grande que celle des absences observées (pour chaque paire possible)
voir fonctions `AUC` et `diagplot` ds `mytoolbox.R`

GLM gaussien

On simule un jeu de données avec une distribution Normale (=Gaussienne)
Deux manières différentes mais strictement équivalentes de générer les y

```
n <- 10
beta0 <- 25
beta1 <- 1.5
beta2 <- -2
sigma <- 5

fertilizer <- rep (0:4, each=n)
set.seed(1)
mildew <- runif(n*5,0,4)
set.seed(2)

tomato <- beta0 + beta1*fertilizer + beta2*mildew + rnorm(n*5,0,sigma)

set.seed(2)
tomato2 <- rnorm(n = n*5,
                 mean = beta0 + beta1*fertilizer - beta2*mildew,
                 sd = sigma)

> all(tomato == tomato2)
[1] TRUE
```

$$y = \alpha + \beta * x + \epsilon$$

$$\epsilon \sim Normale(0, \sigma^2)$$

$$y \sim Normale(\alpha + \beta * x, \sigma^2)$$

GLM gaussien

```
> modlm <- lm(tomato ~ fertilizer + mildew)
> summary(modlm)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  25.3986     2.0847  12.183 3.77e-16 ***
fertilizer    0.9925     0.5705   1.740   0.0885 .
mildew       -1.5484     0.7484  -2.069   0.0441 *
---
Residual standard error: 5.7 on 47 degrees of freedom
Multiple R-squared:  0.1301,  Adjusted R-squared:  0.09304
F-statistic: 3.513 on 2 and 47 DF,  p-value: 0.03785
```

```
> modglm <- glm(tomato ~ fertilizer + mildew,
                                  family = gaussian)
> summary(modglm)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)  25.3986     2.0847  12.183 3.77e-16 ***
fertilizer    0.9925     0.5705   1.740   0.0885 .
mildew       -1.5484     0.7484  -2.069   0.0441 *
---
(Dispersion parameter for gaussian family taken to be 32.49223)

    Null deviance: 1755.4  on 49  degrees of freedom
Residual deviance: 1527.1  on 47  degrees of freedom
AIC: 320.85

Number of Fisher Scoring iterations: 2
```

```
> summary(modlm)$sigma
[1] 5.700195
> summary(modglm)$dispersion
[1] 32.49223
> sqrt(summary(modglm)$dispersion)
[1] 5.700195
```

Plus de R² mais une "Null Deviance" et "Residual Deviance" (voir plus loin)

GLM de Poisson

On a suivi pendant 15 ans l'évolution d'une population sur un site en comptant le nombre d'individus dans 5 quadrats.
Typiquement ce genre de données de comptage suit une distribution de Poisson.

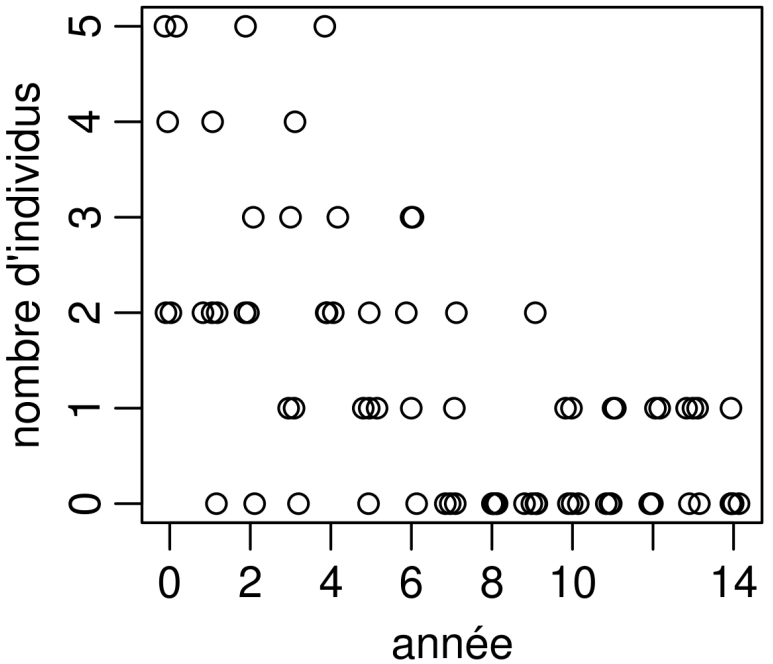
```
n <-5
x <- rep(0:14, each = n)
lambda <- 1 - 0.15 * x
```

$$\lambda = \alpha + \beta * x$$

```
set.seed(123)
y <- rpois(n*15, exp(lambda))
```

$$y \sim Poisson\left(g^{-1}(\lambda)\right) = Poisson\left(e^{\alpha+\beta*x}\right)$$

```
plot(y ~ jitter(x),
     ylab = "nombre d'individus",
     xlab = "année")
```



GLM de Poisson

Résultats et interprétation

```
> modglm <- glm(y ~ x, family = poisson)
> summary(modglm)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3668  -1.0030  -0.5046   0.7459   2.1631

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2208     0.1527   7.993 1.32e-15 ***
x            -0.1908     0.0290  -6.578 4.77e-11 ***
---

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 131.832  on 74  degrees of freedom
Residual deviance:  79.884  on 73  degrees of freedom
AIC: 193.15

Number of Fisher Scoring iterations: 5
```

log link par défaut

GLM de Poisson

Résultats et interprétation

```
> modglm <- glm(y ~ x, family = poisson)
> summary(modglm)

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3668  -1.0030  -0.5046   0.7459   2.1631

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.2208     0.1527   7.993 1.32e-15 ***
x            -0.1908     0.0290  -6.578 4.77e-11 ***
---

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 131.832  on 74  degrees of freedom
Residual deviance:  79.884  on 73  degrees of freedom
AIC: 193.15

Number of Fisher Scoring iterations: 5
```

log link par défaut

L'année 0, il y avait en moyenne exp(1.2208) = 3.39 individus par quadrat.

Un an plus tard, il y avait exp(1.2208 - 0.1908*1) = 2.80 individus par quadrat.

10 ans plus tard : exp(1.2208 - 0.1908*10) = 0.50 individus par quadrat

soit une diminution de (0.47 - 3.39)*100/3.39 = - 86.14 %

GLM de Poisson

Résultats et interprétation

La relation n'est donc pas linéaire au cours du temps mais peut s'interpréter de manière multiplicative

```
> modglm <- glm(y ~ x, family = poisson)
> summary(modglm)
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.2208     0.1527   7.993 1.32e-15 ***
x             -0.1908     0.0290  -6.578 4.77e-11 ***
```

Un an plus tard, il y avait
exp(1.2208 - 0.198*1)
= exp(1.2208) * exp(-0.198)

Autrement dit, en un an, le nombre moyen est multiplié par exp(-0.198) = 0.82 soit une diminution de 18 %
par rapport à l'année précédente

Et en effet :
(2.78 - 3.39)*100/3.39 = -17.994 %

Et au bout de 10 ans :

```
> 1-exp(-0.198*10)
[1] 0.8619308
```

```
> slopes <- round(seq(-0.4,0.4,0.05),2)
> data.frame(
+     slopes = slopes,
+     expslopes = round(exp(slopes),2),
+     pct = 100*(round(exp(slopes),2)-1)
+     )
   slopes expslopes pct
1   -0.40      0.67 -33
2   -0.35      0.70 -30
3   -0.30      0.74 -26
4   -0.25      0.78 -22
5   -0.20      0.82 -18
6   -0.15      0.86 -14
7   -0.10      0.90 -10
8   -0.05      0.95  -5
9    0.00      1.00   0
10   0.05      1.05   5
11   0.10      1.11  11
12   0.15      1.16  16
13   0.20      1.22  22
14   0.25      1.28  28
15   0.30      1.35  35
16   0.35      1.42  42
17   0.40      1.49  49
```

```
> Intercepts <- 0:7
> data.frame(Intercepts =
                  Intercepts,
+            expIntercepts =
                  exp(Intercepts))
  Intercepts expIntercepts
1          0      1.000000
2          1      2.718282
3          2      7.389056
4          3     20.085537
5          4     54.598150
6          5    148.413159
7          6    403.428793
8          7   1096.633158
```

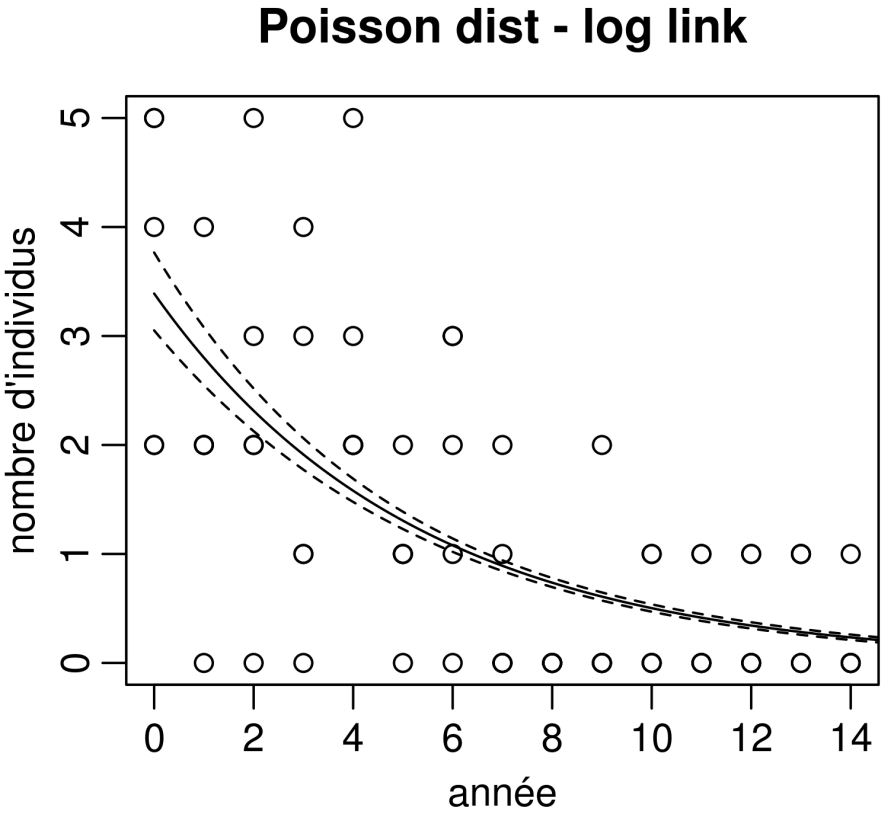
GLM de Poisson

Représentation graphique

Les prédictions et erreurs standards sont sur l'échelle log. Il faut les rétro-transformer pour obtenir les valeurs sur l'échelle d'origine.
Attention : Il ne faut PAS rétro-transformer les erreurs standard elles-mêmes mais bien les bornes : valeurs prédites +- se

NB : la fonction predict() transforme les erreurs standards et ne donnera pas les mêmes résultats...

```
X <- cbind(1, seq(0, 15, 0.1))
pred <- X %*% coef(modglm)
se <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwr <- exp(pred - se)
upr <- exp(pred + se)
pred <- exp(pred)

plot(y ~x, ylab = "nombre d'individus",
     xlab = "année",
     main = "Poisson dist - log link")
lines(pred, x = seq(0, 15, 0.1))
lines(lwr, x = seq(0, 15, 0.1), lty = 2)
lines(upr, x = seq(0, 15, 0.1), lty = 2)
```

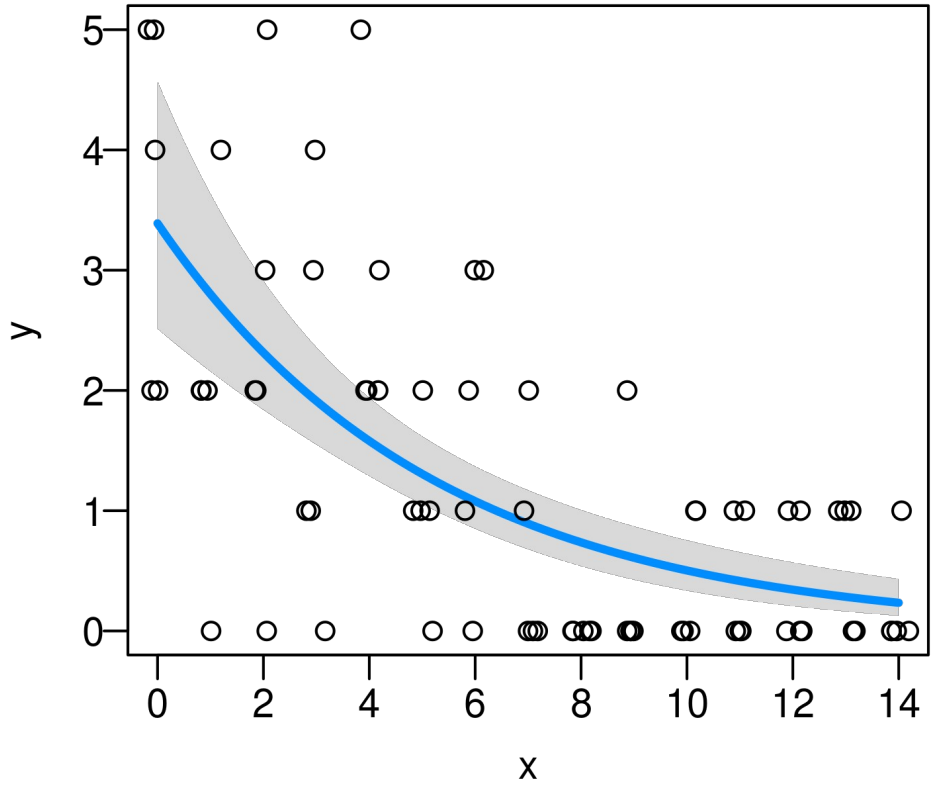


GLM de Poisson

Représentation graphique avec visreg

NB : ici on a pas affiché les résidus partiels. on a ajouté dans un deuxième temps les valeurs réellement observées sur le graphique

```
library(visreg)
par( mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(modglm, scale = "response", rug = FALSE, ylim = c(0,5))
points(y ~ jitter(x))
```

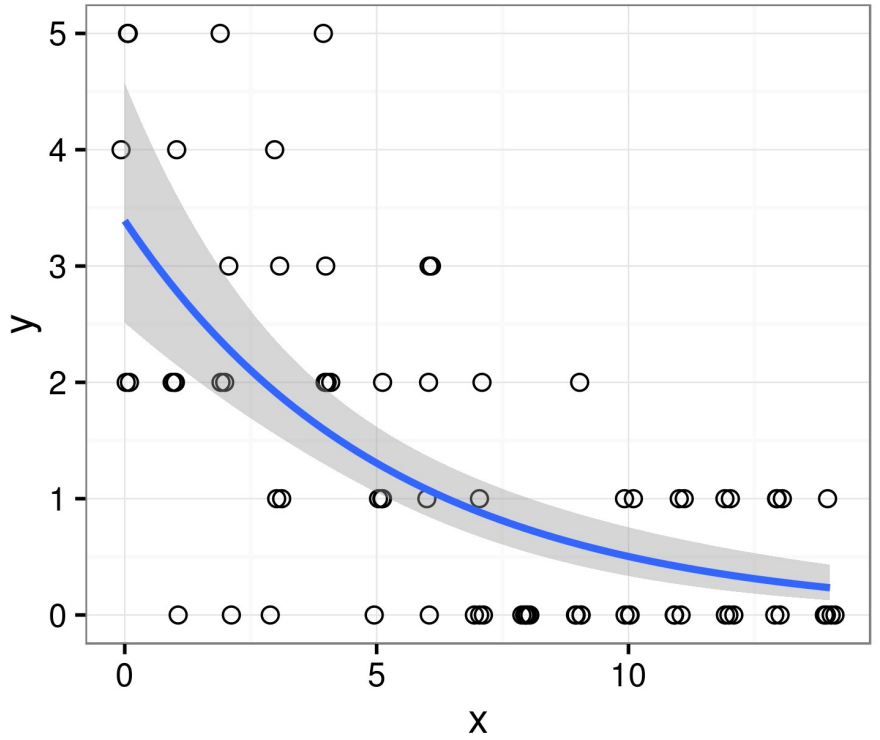


GLM de Poisson

Représentation graphique avec ggplot

NB : ggplot n'utilise pas le modèle que l'on a créé mais peut lui même estimer une régression de poisson

```
d <- data.frame(y,x)
ggplot(d, aes(x=x, y = y)) +
    geom_point(size = 2, shape = 1,  position = position_jitter(width = 0.3, height = 0)) +
    stat_smooth(method = "glm", se = TRUE, method.args = list(family = poisson)) +
    theme_bw()
```



GLM de Poisson

Différentes fonction de lien + modèles linéaires

```
> modglm <- glm(y ~ x, family = poisson)
> modglm2 <- glm(y ~ x ,
              family = poisson(link = "sqrt"))

> modglm3 <- glm(y ~ x ,
              family = poisson(link = "identity"))
Erreur : impossible de trouver un jeu de coefficients
correct : prière de fournir des valeurs initiales
De plus : Message d'avis :
In log(y/mu) : production de NaN
> modglm3 <- glm(y ~ x ,
     family = poisson(link = "identity"), start=c(3,-0.2))

> modlm <- glm(y ~ x)
> modlm2 <- glm(log(y+1) ~ x)
```

Modèle de Poisson avec une fonction de lien racine carrée

Le lien identité n'est à priori pas très adapté ea car il peut prédire des valeurs négatives.
Il faut ici lui fournir des valeurs de départ sans quoi l'algorithme n'arrive pas à estimer les paramètres

Modèles linéaires classiques
Une approche fréquente pour ce genre de données (mais pas forcément conseillée) est de transformer les y avec un log. Comme il y a des 0 on est obligé d'utiliser log(y+1)

GLM de Poisson

Différentes fonction de lien + modèles linéaires

```
> deviance(modglm) ; deviance(modglm2) ; deviance(modglm3) ; deviance(modlm)  ;
deviance(modlm2)

[1] 79.88419
[1] 82.33936
[1] 87.91184
[1] 93.07524
[1] 16.719
```

On compare la qualité d'ajustement du modèle (est-ce qu'il passe plus ou moins bien par les points observés?)
Le modèle de Poisson avec lien log est bien le meilleur modèle
Ici : déviance et AIC les plus faibles, R^2 le plus élevé.
Le nombre de paramètres est identique (sauf pour les modèles gaussiens) on peut donc utiliser déviance et R^2

```
> AIC(modglm) ; AIC(modglm2) ; AIC(modglm3) ; AIC(modlm) ; AIC(modlm2)

[1] 193.155
[1] 195.6101
[1] 201.1826
[1] 235.0348
[1] 106.2701
```

ATTENTION ! On ne peut pas comparer la déviance ou les AIC du dernier modèle car les Y ne sont pas les mêmes (transformation log)

```
> pseudoRsq(modglm) ; pseudoRsq(modglm2) ; pseudoRsq(modglm3) ; pseudoRsq(modlm) ;
pseudoRsq(modlm2)

[1] 0.433
[1] 0.428
[1] 0.392
[1] 0.392
[1] 0.38
```

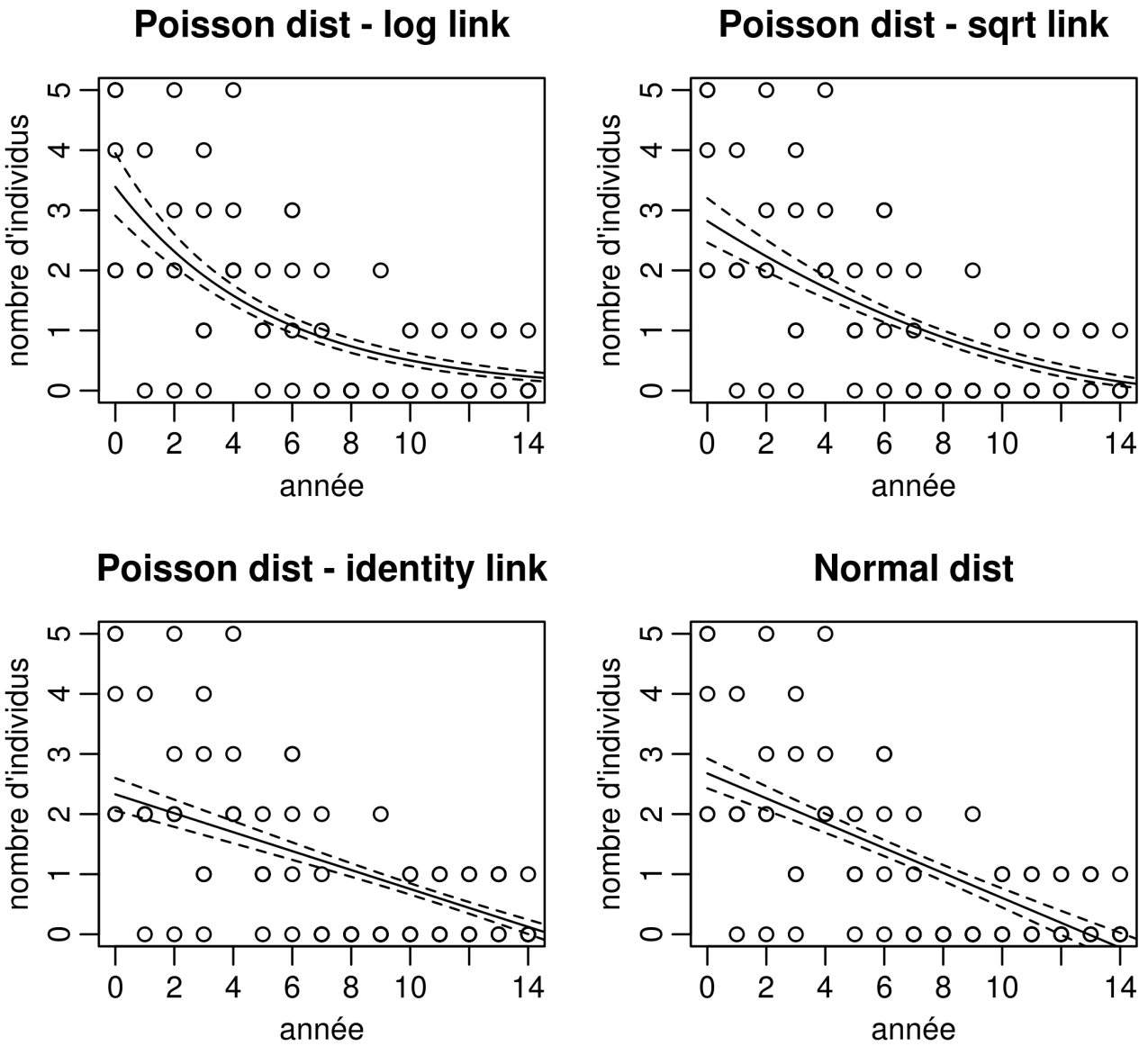
On ne peut pas non plus directement comparer le pseudo-R^2 tel quel car pour le dernier modèle, les valeurs prédites le sont sur l'échelle log alors que pour tous les autres on travaille sur l'échelle d'origine.
Mais on peut calculer le pseudo-R^2 après rétro-transformation qui lui est comparable :

```
> summary(lm(I(exp(modlm2$model[,1])-1) ~ I(exp(fitted(mod))-1)))$r.squared
[1] 0.4120851
```

GLM de Poisson

Différentes fonction de lien + modèles linéaires

Représentation graphique des 4 premiers modèles



Code pour les graphiques de la dia précédente

```
par(mfrow = c(2,2), mar = c(3,3,3,1),
     mgp = c(1.75, 0.6, 0))

modglm <- glm(y ~ x , family = poisson)
X <- cbind(1, seq(0, 15, 0.1))
pred <- X %*% coef(modglm)
se <- sqrt(diag(X %*% vcov(modglm) %*% t(X)))
lwr <- exp(pred - se)
upr <- exp(pred + se)
pred <- exp(pred)

plot(y ~x, ylab = "nombre d'individus",
     xlab = "année",
     main = "Poisson dist - log link")
lines(pred, x = seq(0, 15, 0.1))
lines(lwr, x = seq(0, 15, 0.1), lty = 2)
lines(upr, x = seq(0, 15, 0.1), lty = 2)


modglm2 <- glm(y ~ x ,
     family = poisson(link = "sqrt"))
pred <- X %*% coef(modglm2)
se <- sqrt(diag(X %*% vcov(modglm2) %*% t(X)))
lwr <- (pred - se)^2
upr <- (pred + se)^2
pred <- (pred)^2

plot(y ~x, ylab = "nombre d'individus",
     xlab = "année",
     main = "Poisson dist - sqrt link")
lines(pred, x = seq(0, 15, 0.1))
lines(lwr, x = seq(0, 15, 0.1), lty = 2)
lines(upr, x = seq(0, 15, 0.1), lty = 2)
```

```
modglm3 <- glm(y ~ x ,
     family = poisson(link = "identity"),
     start=c(3,-0.2))
pred <- X %*% coef(modglm3)
se <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwr <- (pred - se)
upr <- (pred + se)
pred <- (pred)

plot(y ~x, ylab = "nombre d'individus",
     xlab = "année",
     main = "Poisson dist - identity link")
lines(pred, x = seq(0, 15, 0.1))
lines(lwr, x = seq(0, 15, 0.1), lty = 2)
lines(upr, x = seq(0, 15, 0.1), lty = 2)


mod <- lm(y ~ x)
X <- cbind(1, seq(0, 15, 0.1))
predlm <- X %*% coef(mod)
selm <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwrlm <- (predlm - selm)
uprlm <- (predlm + selm)
predlm <- (predlm)

plot(y ~x, ylab = "nombre d'individus",
     xlab = "année",
     main = "Normal dist")
lines(predlm, x = seq(0, 15, 0.1))
lines(lwrlm, x = seq(0, 15, 0.1), lty = 2)
lines(uprlm, x = seq(0, 15, 0.1), lty = 2)
```

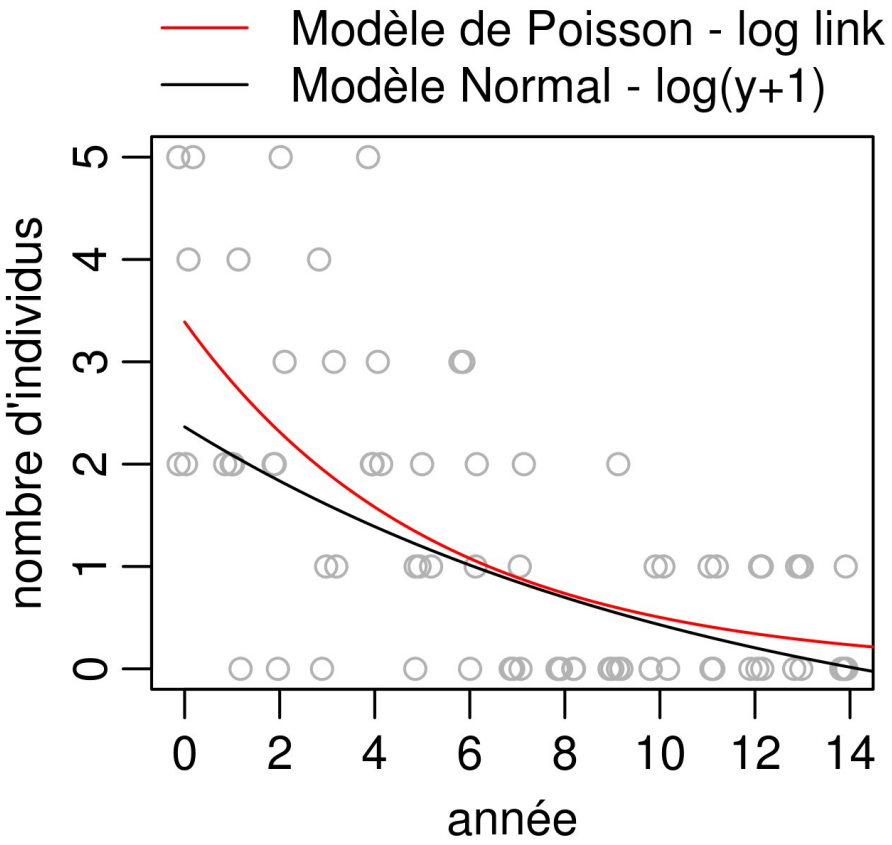
GLM de Poisson

Différentes fonction de lien + modèles linéaires

Comparaison modèle de Poisson vs modèle gaussien avec transformation des y en log(y+1)

Le fait d'ajouter 1 n'est pas anodin surtout sur des abondances faibles...

```
mod <- lm(log(y+1) ~ x)
X <- cbind(1, seq(0, 15, 0.1))
predlm <- X %*% coef(mod)
selm <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwrlm <- exp(predlm - selm)-1
uprlm <- exp(predlm + selm)-1
predlm <- exp(predlm)-1

modglm <- glm(y ~ x , family = poisson)
pred <- X %*% coef(modglm)
se <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwr <- exp(pred - se)
upr <- exp(pred + se)
pred <- exp(pred)

dev.new(8.5/2.54, 8/2.54)
par(mfrow = c(1,1), mar = c(3,3,3,1),
     mgp = c(1.75, 0.6, 0))
set.seed(2)
plot(y ~ jitter(x), ylab = "nombre d'individus",
     xlab = "année", col = "grey70")
lines(pred, x = seq(0, 15, 0.1), col = "red")
lines(predlm, x = seq(0, 15, 0.1))
```




```
legend(x = "top", inset = -0.3, xpd = NA, bty = "n", lty = 1, col = c("red", "black") ,
       legend = c("Modèle de Poisson - log link", "Modèle Normal - log(y+1)"))
```

GLM binomial

Exemple classique d'écotoxicologie :

On a exposé des mâles et des femelles d'une espèce d'insecte à 6 doses croissantes d'un pesticide.

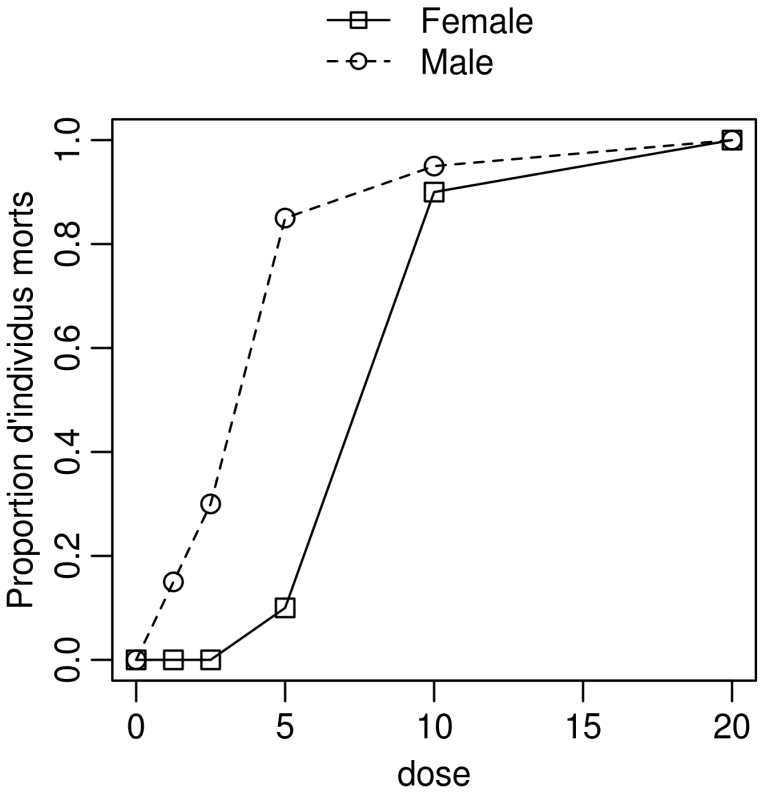
A chaque dose on a testé 20 mâles et 20 femelles (N = 20) et on a regardé combien étaient morts après 2 jours (= nombre de "succès")

GLM binomial

Génération du jeu de données

```
nb = 20
d <- data.frame(
    sex = rep(c("M", "F"), each = 6),
    dose = rep(c(0, 1.25, 2.5, 5, 10, 20), 2),
    n = nb)

X <- cbind(1, d$sex, d$dose)
Beta <- c(-9, 3, 0.8)
linpred <- X %*% Beta
invlogit <- function(x) {return(exp(x)/(1+exp(x)))}
prop <- invlogit(linpred)

set.seed(2)
d$dead <- rbinom(12, p = prop, size = nb)
d$prop <- d$dead/d$n
```

```
> d
   sex  dose  n dead prop
1    M  0.00 20    0 0.00
2    M  1.25 20    3 0.15
3    M  2.50 20    6 0.30
4    M  5.00 20   17 0.85
5    M 10.00 20   19 0.95
6    M 20.00 20   20 1.00
7    F  0.00 20    0 0.00
8    F  1.25 20    0 0.00
9    F  2.50 20    0 0.00
10   F  5.00 20    2 0.10
11   F 10.00 20   18 0.90
12   F 20.00 20   20 1.00
```




```
dev.new(9/2.54, 9/2.54)
par(mar = c(3,3,3,1), mgp = c(1.75, 0.6, 0), cex = 0.9)
plot(prop ~ dose, data=d, pch = c(0,1)[as.numeric(d$sex)],
     cex = 1.2, ylab = "Proportion d'individus morts")
lines(prop ~ dose, data=d[d$sex == "F",], lty = 1)
lines(prop ~ dose, data=d[d$sex == "M",], lty = 2)
legend(x = "top", inset = -0.25, xpd = NA, bty = "n",
       lty = 1:2, pch = 0:1 ,
       legend = c("Female", "Male"))
```

GLM binomial

Modèle

Deux syntaxes possibles donnant des résultats identiques:
1) y = proportion de morts, et N dans l'argument "weights"
2) y = matrice à deux colonnes avec nombre de vivants et nombre de morts

```
> mod <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial)
> mod2 <- glm(cbind(dead, n-dead) ~ sex + dose, data=d, family = binomial)
> summary(mod)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6042     1.0208  -6.469 9.83e-11 ***
sexM          3.5689     0.7931   4.500 6.81e-06 ***
dose          0.8682     0.1236   7.025 2.15e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 236.7523  on 11  degrees of freedom
Residual deviance:   6.7239  on  9  degrees of freedom
AIC: 28.658

Number of Fisher Scoring iterations: 6
```

GLM binomial

Modèle

On peut vérifier qu'il n'y a pas d'interaction

```
> mod3 <- glm(prop ~ sex * dose, weights = n, data=d, family = binomial)
> anova(mod, mod3, test = "Chisq")

Analysis of Deviance Table

Model 1: prop ~ sex + dose
Model 2: prop ~ sex * dose
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         9     6.7239
2         8     6.4366  1  0.28732   0.5919
```

GLM binomial

Interprétation (1)

Rappel : par défaut on travaille sur l'échelle logit

logit = log(p/(1-p)

invlogit = exp(p) / (1+exp(p))

```
invlogit <- function(x) {return(exp(x)/(1+exp(x)))}

> mod <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial)
> summary(mod)

            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6042     1.0208  -6.469 9.83e-11 ***
sexM          3.5689     0.7931   4.500 6.81e-06 ***
dose          0.8682     0.1236   7.025 2.15e-12 ***
```

GLM binomial

Interprétation (1)

Rappel : par défaut on travaille sur l'échelle logit
logit = log(p/(1-p) ; invlogit = exp(p) / (1+exp(p))

```
invlogit <- function(x) {return(exp(x)/(1+exp(x)))}
> mod <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial)
> summary(mod)

            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6042     1.0208  -6.469 9.83e-11 ***
sexM          3.5689     0.7931   4.500 6.81e-06 ***
dose          0.8682     0.1236   7.025 2.15e-12 ***
```

La proportion estimée de morts pour les femelles à une dose de 0 est de invlogit(-6.6042) = 0.0013
Pour les mâles cette proportion est de invlogit(-6.6042+3.5689) = 0.046
Lorsque la dose passe de 0 à 1, la proportion de mortalité passe de 0.0013 à invlogit(-6.6042 + 0.8682*1) = 0.0032 pour les femelles et de 0.046 à invlogit(-6.6042 + 3.5689+ 0.8682*1) = 0.10 pour les mâles.

GLM binomial

Interprétation (1)

NB : p/(1-p) représente un "odds ratio"
C'est la probabilité qu'un événement se produise / la probabilité qu'il ne se produise pas.
Lorsque ce ratio est > 1 il est plus probable que l'événement se produise
Lorsque ce ratio est <1, il est plus probable qu'il ne se produise pas

Sur l'échelle log, on a des valeurs positives lorsqu'un événement est plus probable et des valeurs négatives pour un événement moins probable et 0 quand on a autant de chances qu'il se produise que l'inverse (p = 0.5)

```
invlogit <- function(x) {return(exp(x)/(1+exp(x)))}
> mod <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial)
> summary(mod)

             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6042     1.0208  -6.469 9.83e-11 ***
sexM          3.5689     0.7931   4.500 6.81e-06 ***
dose          0.8682     0.1236   7.025 2.15e-12 ***
```

GLM binomial

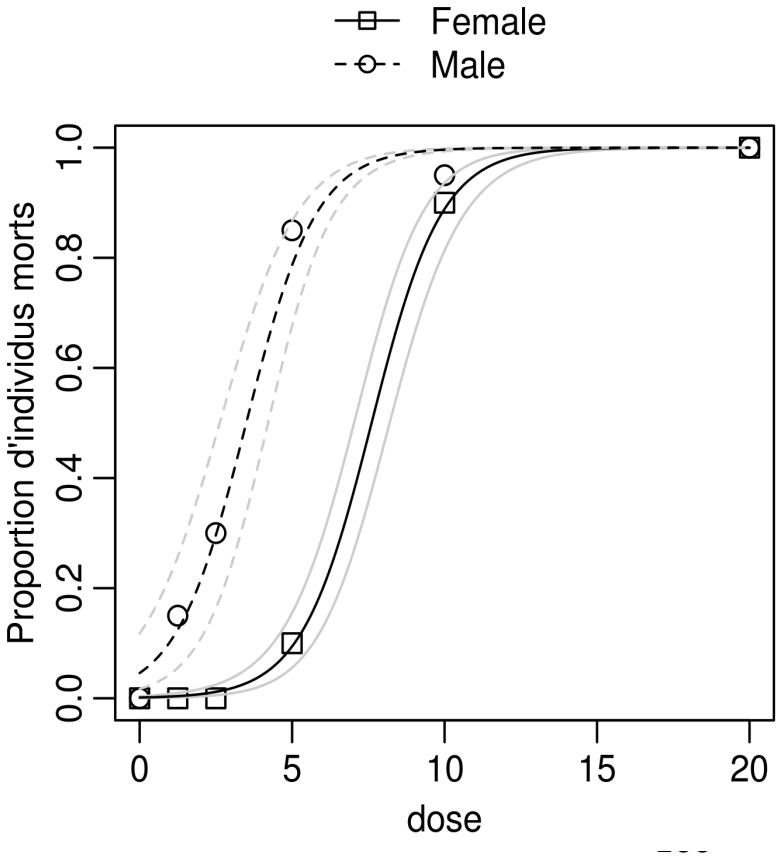
Interprétation (2)

Le signe peut s'interpréter directement : il y a plus de mortalité chez les mâles à la dose 0 (et aux autres doses) que chez les femelles et la mortalité augmente quand la dose augmente chez les femelles (et chez les mâles aussi puisqu'il n'y a pas d'interaction)

La sigmoide a une pente maximale pour p = 0.5.

A cet endroit la relation est à peu près linéaire avec une pente = à beta/4 (pour le lien logit uniquement!)

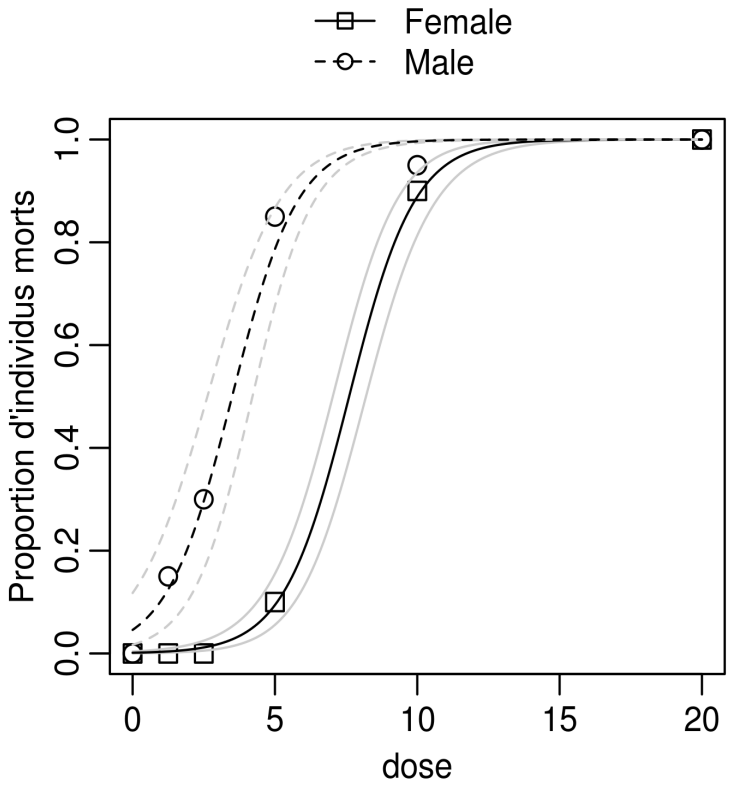
Soit ici :
0.8682 /4 = 0.21
Quand la proportion de morts est à 0.5, elle passe à 0.5 + 0.21 si on augmente la dose d'une unité



GLM binomial

Représentation graphique

```
X <- cbind(1, 0, seq(0, 20, 0.1))
pred <- X %*% coef(mod)
se <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwr <- invlogit(pred - se)
upr <- invlogit(pred + se)
pred <- invlogit(pred)

XM <- cbind(1, 1, seq(0, 20, 0.1))
predM <- XM %*% coef(mod)
seM <- sqrt(diag(X %*% vcov(mod) %*% t(X)))
lwrM <- invlogit(predM - se)
uprM <- invlogit(predM + se)
predM <- invlogit(predM)

dev.new(9/2.54, 9/2.54)
par(mar = c(3,3,3,1), mgp = c(1.75, 0.6, 0), cex = 0.9)
plot(prop ~ dose, data=d, pch = c(0,1)[as.numeric(d$sex)
    cex = 1.2, ylab = "Proportion d'individus morts")
lines(y = lwr, x = X[,3], lty = 1, col = "grey80")
lines(y = upr, x = X[,3], lty = 1, col = "grey80")
lines(y = pred, x = X[,3], lty = 1, col = "black")

lines(y = lwrM, x = X[,3], lty = 2, col = "grey80")
lines(y = uprM, x = X[,3], lty = 2, col = "grey80")
lines(y = predM, x = X[,3], lty = 2, col = "black")

legend(x = "top", inset = -0.25, xpd = NA, bty = "n", lty = 1:2, pch = 0:1 ,
       legend = c("Female", "Male"))
```



GLM binomial

Représentation graphique avec visreg

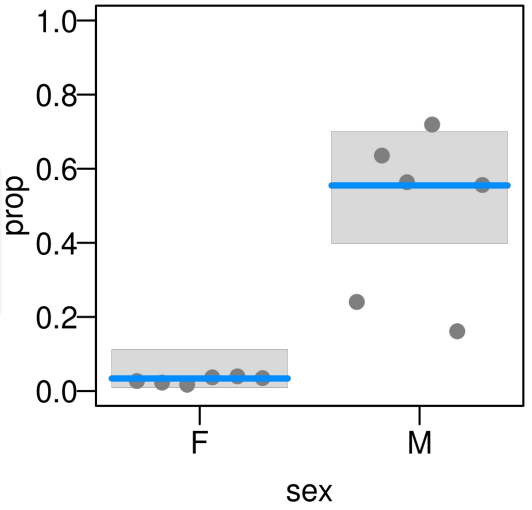
Facile mais pas idéal...

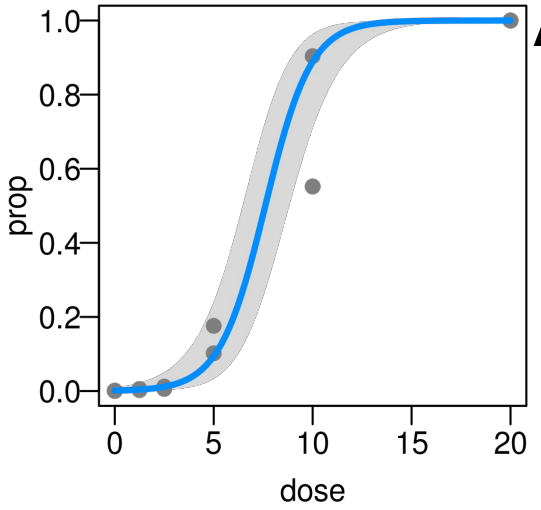
```
par( mfrow = c(1,2), mar = c(3,3,1,1), mgp = c(1.8, 0.5, 0), cex = 0.9)
visreg(mod, scale = "response", partial = TRUE, points.par = list(cex =1),
       ylim = c(0,1))
visreg(mod, scale = "response", cond=list(sex = "M", dose = 10),
       partial = TRUE, points.par = list(cex =1),ylim = c(0,1))
```

A

B

Par défaut :
mortalité moyenne
à une dose médiane
(3.75 l/ha ici)

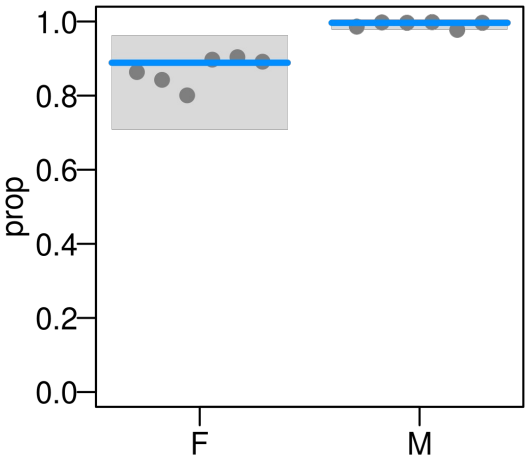


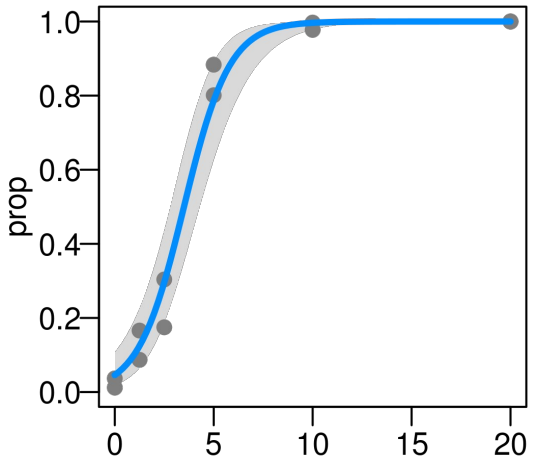


A

Par défaut :
mortalité vs dose
pour les Femelles
(groupe le plus fréquent
ou par ordre alphabétique
en cas d'égalité)

Ici on a demandé :
mortalité moyenne
à une dose de 10 l/ha





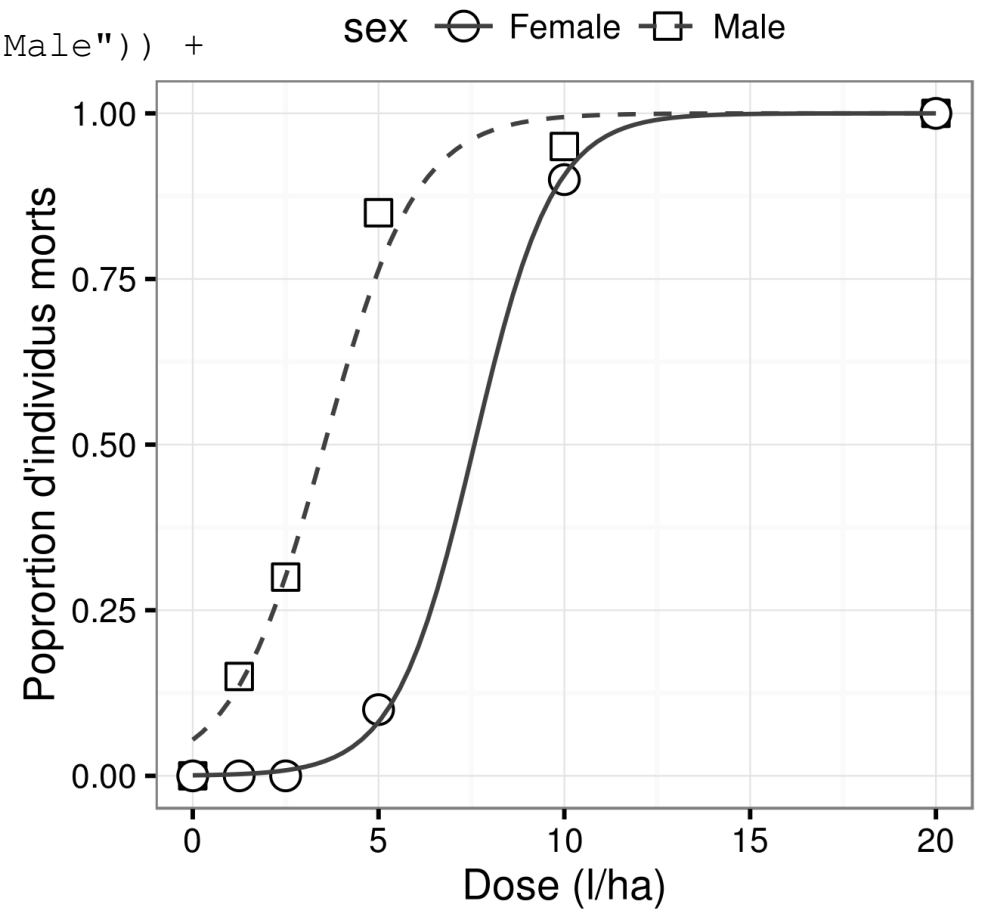
B

ici on a demandé
mortalité vs dose
pour les Males

GLM binomial

Représentation graphique avec ggplot

```
ggplot(d, aes(y = prop, x = dose, group = sex, lty = sex)) +
    geom_point(aes(shape = sex), size = 3, fill = "white") +
    stat_smooth(method = "glm", method.args = list(family = binomial),
                se = FALSE, color = "gray25", lwd = 0.5) +
    ylab("Poprortion d'individus morts") +
    xlab("Dose (l/ha)") +
    scale_shape_manual(values = c(21, 22),
                       labels = c("Female", "Male")) +
    scale_linetype_manual( values = c(1, 2),
                           labels = c("Female", "Male")) +
    theme_bw(10) +
    theme(legend.position = "top",
          legend.margin = unit(0, "line"),
          legend.key = element_rect(color = NA))
```



GLM binomial

Autres fonctions de lien

```
> mod <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial)
> mprobit <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial(link = "probit"))
> mcauchit <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial(link = "cauchit"))
> mcloglog <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial(link = "cloglog"))
Messages d'avis :
1: glm.fit: l'algorithme n'a pas convergé
2: glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> mlog <- glm(prop ~ sex + dose, weights = n, data=d, family = binomial(link = "log"))
Erreur : impossible de trouver un jeu de coefficients correct : prière de fournir
des valeurs initiales

> compmod <- function(models) {
+     data.frame(
+         deviance = sapply(models, deviance),
+         AICc = sapply(models, aic)[2,],
+         QAICc = sapply(models, aic)[4,],
+         PsRsq = sapply(models, pseudoRsq),
+         disp = sapply(models, overdisp)[1,]
+         )
+ }
>
> compmod(list(mod, mprobit, mcauchit, mcloglog))
   deviance     AICc    QAICc PsRsq      disp
1  6.723949 31.65751 28.00706 0.996 1.5852427
2  9.083086 34.01665 21.99738 0.991 3.0202049
3  9.021243 33.95481 38.66909 0.994 0.6411044
4 26.159992 51.09356 27.71492 0.942 3.0065462
```

Typiquement, on devrait prendre le log de la dose quand on utilise le lien cloglog

Le premier modèle (logit) est le meilleur (AIC, Deviance les plus faibles)

GLM binomial

Autres fonctions de lien

```
X <- cbind(1, 0, seq(0, 20, 0.1))
pred <- invlogit(X %*% coef(mod))
predprobit <- make.link("probit")$linkinv(X %*% coef(mprobit))
predcauchit <- make.link("cauchit")$linkinv(X %*% coef(mcauchit))
predcloglog <- make.link("cloglog")$linkinv(X %*% coef(mcloglog))

dev.new(9/2.54, 9/2.54)
par(mar = c(3,3,3,1), mgp = c(1.75, 0.6, 0), cex = 0.9)
plot(prop ~ dose, data=d, pch = c(0,1)[as.numeric(d$sex)], cex = 1.2,
     ylab = "Proportion d'individus morts")

lines(y = pred, x = X[,3], lty = 1, col = "black")
lines(y = predprobit, x = X[,3], lty = 2, col = "black")
lines(y = predcauchit, x = X[,3], lty = 3, col = "black")
lines(y = predcloglog, x = X[,3], lty = 4, col = "black")


legend(x = "top", inset = -0.25, xpd = NA, bty = "n", lty = 1:4,
      ncol = 2,
     legend = c("logit", "probit", "cauchit", "cloglog"))

X <- cbind(1, 1, seq(0, 20, 0.1))
pred <- invlogit(X %*% coef(mod))
predprobit <- make.link("probit")$linkinv(X %*% coef(mprobit))
predcauchit <- make.link("cauchit")$linkinv(X %*% coef(mcauchit))
predcloglog <- make.link("cloglog")$linkinv(X %*% coef(mcloglog))

lines(y = pred, x = X[,3], lty = 1, col = "black")
lines(y = predprobit, x = X[,3], lty = 2, col = "black")
lines(y = predcauchit, x = X[,3], lty = 3, col = "black")
lines(y = predcloglog, x = X[,3], lty = 4, col = "black")
```

4 fonctions de lien fonctionnent ici. Elles sont toutes très similaires (sigmoïde)