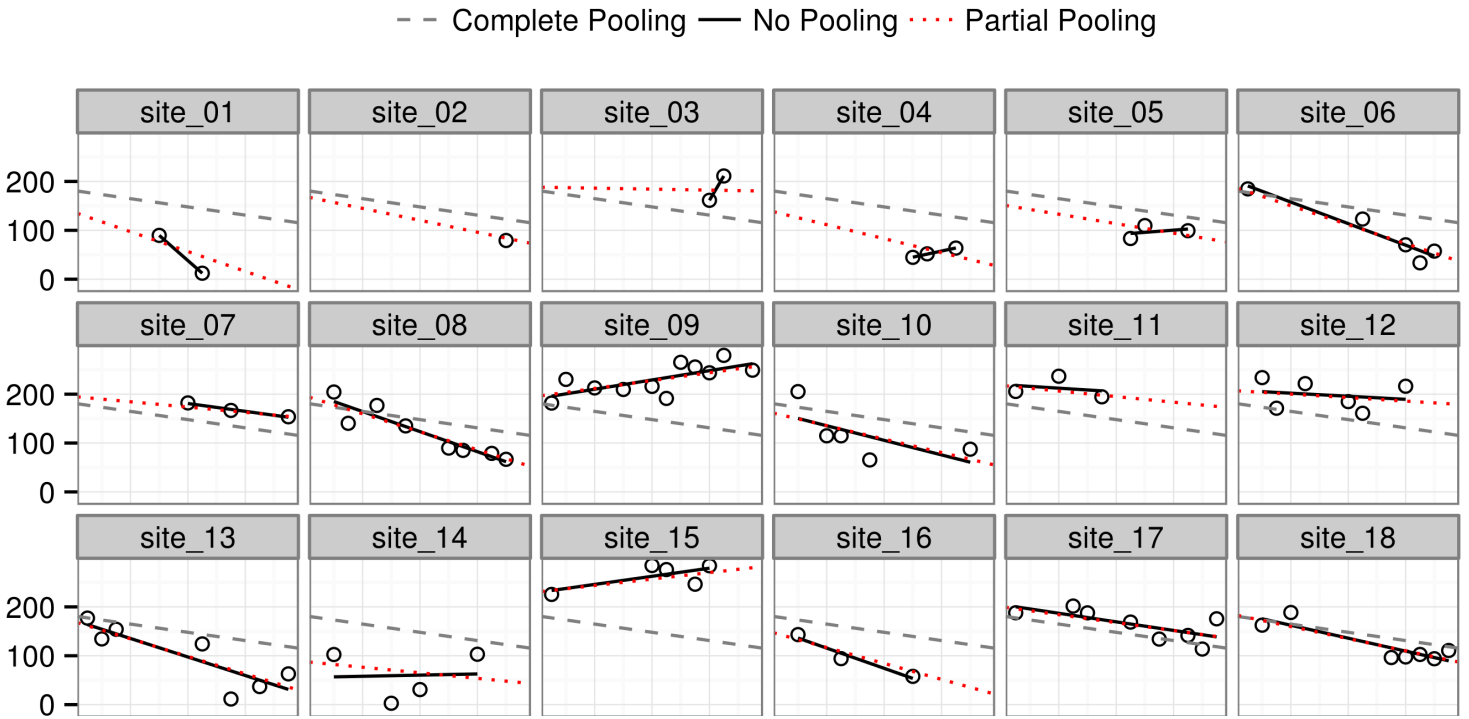
Mixed Models - Multilevel Models



G. San Martin

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

cra-w

Modèles Mixtes

Un modèle mixte est (traditionnellement) un modèle où on mélange des variables explicatives qualitatives (facteurs) de deux types :

effets "fixes" ← Ce qu'on a vu jusqu'à présent

effets "aléatoires"

On utilise des effets aléatoires typiquement quand on a des **groupes** d'observations non indépendantes :

des **sites** sur lesquels on fait plusieurs mesures

des **individus** que l'on mesure plusieurs fois au cours du temps

des "**Blocs**" délimitant des zones homogènes au sein d'un champ

des cages d'élevage dans plusieurs **chambres climatisées**

des individus que l'on a mesurés au sein de **familles** elles mêmes au sein de **populations** elles mêmes au sein de **régions**

etc...

Modèles Mixtes

Il existe ici aussi deux méthodes principales permettant d'estimer ces modèles :

1) les ANOVA mixtes

Utilisant la méthode des moindres carrés

Historiquement la première, développée au départ en agronomie

dans R : fonction `aov()`

Cette méthode a quelques avantages :

la facilité de calculer les carrés moyens des différents groupes d'observations

les inférences (p valeurs) sont "exactes" (dans certains cas précis)

C'est une méthode qui est (était?) généralement mieux connue et bien établie pour de nombreux plans expérimentaux classiques

Modèles Mixtes

1) les ANOVA mixtes

Et des désavantages ... :

Les inférences (p-valeurs) ne sont valides que pour les designs parfaitement balancés (nombre équivalent d'observations pour chaque combinaison de facteurs) et sont difficiles à étendre pour des cas non balancés

Uniquement pour des modèles à distribution Gaussienne

Pour obtenir les p-valeurs, on doit calculer un rapport de carrés moyens. Pour les ANOVAs fixes le dénominateur est toujours le carré moyen résiduel. Pour les ANOVAs mixtes le dénominateur sera différent pour chaque variables explicative et pour chaque design expérimental. L'utilisation demande donc un assez haut degré d'expertise et n'est quasi applicable pour l'utilisateur moyen qu'à des designs bien connus et pas trop complexes.

Fortement orienté vers des tests d'hypothèse nulle, pas d'intervalle de confiance, pas de paramètres estimés (ou alors assez compliqués à "extraire")...

En résumé : assez peu de Flexibilité

Modèles Mixtes

2) les "modèles mixtes"

Utilisent une méthode de "Maximum de Vraissemblance Restreinte"
(Restricted Maximum Likelihood = REML)
ie : une méthode proche du ML classique mais non biaisée pour certains paramètres

Dans R, nombreux packages, en particulier : `lme4, nlme, MCMCglmm`
Mais aussi souvent R en combinaison avec BUGS ou JAGS

Désavantages :

Les calculs impliqués sont beaucoup plus complexes mais paradoxalement l'utilisation pratique est plus simple et plus flexible
(avec cependant un risque accru de ne pas comprendre ce qu'on fait...)

Les inférences sont généralement plus approximatives à moins d'utiliser des méthodes de simulations souvent très lentes et moins "automatiques".

Plus récents et parfois moins connus mais de plus en plus utilisés et enseignés

Modèles Mixtes

2) les "modèles mixtes"

Avantages :

Beaucoup plus flexibles, modèles potentiellement très complexes et impossibles à estimer avec des méthodes classiques

Les paramètres sont mieux estimés en particulier dans les designs non balancés et dans un but prédictif

Les paramètres estimés sont souvent beaucoup plus faciles à interpréter ou même à obtenir (ea "variance components")

Modèles Gaussiens, de poisson, Binomiaux, etc...

Permettent d'inclure des structures de corrélation spatiale, temporelle, etc...

Modèles Mixtes

Différentes personnes vont souvent aborder un même modèle mixte avec une approche, un vocabulaire et des objectifs très différents selon les cas de figure (ea étude expérimentale ou observative) ou simplement la culture scientifique.

La définition même d'un effet aléatoire peut être très différente selon les personnes.

On présentera ici la définition classique en ANOVA mixte qui est une base de travail très utile en pratique
(mais il existe d'autres définitions...)

Modèles Mixtes

Effet fixe ou aléatoire ?

La question se pose uniquement pour des variables discrètes (en général qualitatives) pouvant caractériser des groupes d'observations

On considère qu'un effet est aléatoire si les valeurs possibles de cette variable (niveaux) sont un échantillonnage aléatoire parmi un grand nombre d'autres valeurs possibles.
On est pas intéressé à estimer une valeur moyenne pour chaque niveau ni à comparer ces différents niveaux entre eux.
Si on recommençait l'expérience on utiliserait normalement d'autres valeurs
Si la variable est considérée comme fixe, les résultats ne sont pas généralisables à d'autres valeurs.

En pratique, pour un effet aléatoire on va juste estimer la variance additionnelle provoquée par cet effet. Alors que pour un effet fixe, on va estimer la valeur de chaque niveau

Modèles Mixtes

Effet fixe ou aléatoire ?

Par exemple le sexe est typiquement une variable fixe.
Les niveaux de cette variable (mâle ou femelle) ne sont pas un échantillonnage parmi un grand nombre de variable possibles.

Les traitements d'une expérience comme par exemple différents types d'alimentation sont également souvent considérés comme des effets fixes même si il existe de nombreuses autres type d'alimentation que ceux considérés dans l'étude.
En effet on veut en général estimer l'effet de chaque type d'alimentation et les comparer entre eux.
On ne veut pas spécialement généraliser les conclusions à d'autres types d'alimentation.
Si on recommençait l'expérience on testerait les mêmes types d'alimentation qui sont ceux qui nous intéressent.

Modèles Mixtes

Effet fixe ou aléatoire ?

Les variables site, champ, hôpital, patient, ...sur lesquels on aurait fait plusieurs mesures sont souvent des variables aléatoires.

On a un général choisi aléatoirement quelques sites parmi un grand nombre de sites possibles.

On est en général pas intéressé à comparer les différents champs utilisés comme répétitions pour une expérience.

On veut juste estimer quelle est la variabilité supplémentaire due aux différents champs, sites, hôpitaux,...

Si on devait recommencer l'expérience on prendrait probablement d'autres sites, champs, etc... (sauf - mauvaises - raisons de facilité)

On veut en général pouvoir généraliser les conclusions à d'autres sites, champs, patients,...

Modèles Mixtes

Différentes personnes vont souvent aborder un même modèle mixte avec une approche, un vocabulaire et des objectifs très différents selon les cas de figure (ea étude expérimentale ou observative) ou simplement la culture scientifique.

On va aborder ici 3 approches différentes d'un même exemple :

On a estimé la taille des populations d'un animal sur 35 sites pendant une quinzaine d'années (mais on a pas des observations chaque année dans chaque site).

On a donc :

- 186 observations d'abondance ("y") comme variable dépendante
- l'année ("year") de 1 à 15 comme variable explicative continue
- le site (n = 35) comme variable qualitative que l'on pourra considérer comme fixe ou aléatoire selon les cas.

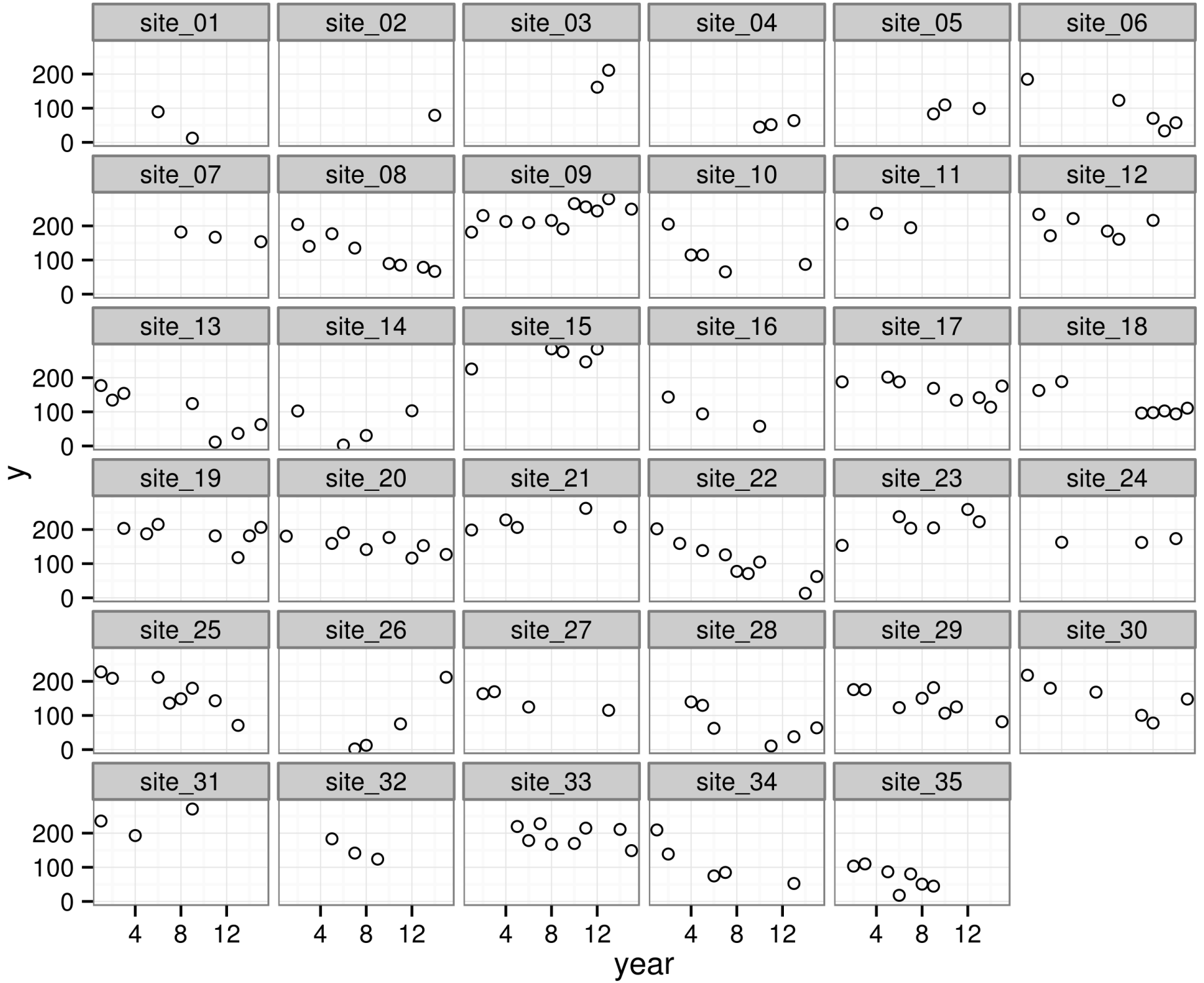
NB : une fois de plus une distribution gaussienne n'est a priori pas très adaptée à ce genre de données. On examinera le même exemple avec une distribution de Poisson plus loin

Simulation de ces données :
On y reviendra en détail plus tard...

```
# simulation du jeu de données
nsites <- 35
ny <- 15
n <- nsites * ny

# création des variables site et year
site <- paste("site", rep(sprintf("%02.0f", 1:nsites), each = ny), sep = "_")
year <- rep(1:ny, times = nsites)

# moyenne et variance des pentes et variance résiduelle
int.mean <- 200
int.sd <- 30
slope.mean <- -10
slope.sd <- 6
sigma <- 25

# Génération des pentes et des intercepts pour chaque groupe
set.seed(1)
int <- rnorm(n = nsites, mean = int.mean, sd = int.sd )
set.seed(2)
slope <- rnorm(n = nsites, mean = slope.mean, sd = slope.sd )
gamma <- c(int, slope)

X <- model.matrix (~ site + site : year - 1)
lin.pred <- X %*% gamma
set.seed(3)
y <- abs(rnorm(n = n, mean = lin.pred, sd = sigma))

d <- data.frame(y, site, year)

# On élimine une bonne partie des données pour créer un jeu de données non balancé
set.seed(234)
d <- d[c(6, 9, 29, 42, 43, 55, 56, 58, 69, 70, 73, sample(x=76:nrow(d), nrow(d)/3)),]
```

Graphique : l'utilisation du package ggplot2 (ou lattice) facilite fortement la représentation de données groupées...

```
library(ggplot2)

ggplot(data=d, aes(y = y, x = year)) +
    geom_point(shape = 1) +
    facet_wrap(~site) + theme_bw()
```


site_01
site_02
site_03
site_04
site_05
site_06
site_07
site_08
site_09
site_10
site_11
site_12
site_13
site_14
site_15
site_16
site_17
site_18
site_19
site_20
site_21
site_22
site_23
site_24
site_25
site_26
site_27
site_28
site_29
site_30
site_31
site_32
site_33
site_34
site_35
y
200
100
0
4
8
12
year

Modèles Mixtes

Approche 1

Des modèles mixtes pour prendre en compte la non indépendance des observations dans la matrice de variance covariance des résidus

Si on veut estimer la tendance globale des populations, une approche est d'estimer un modèle linéaire classique du nombre d'individus en fonction de l'année sans tenir compte du site

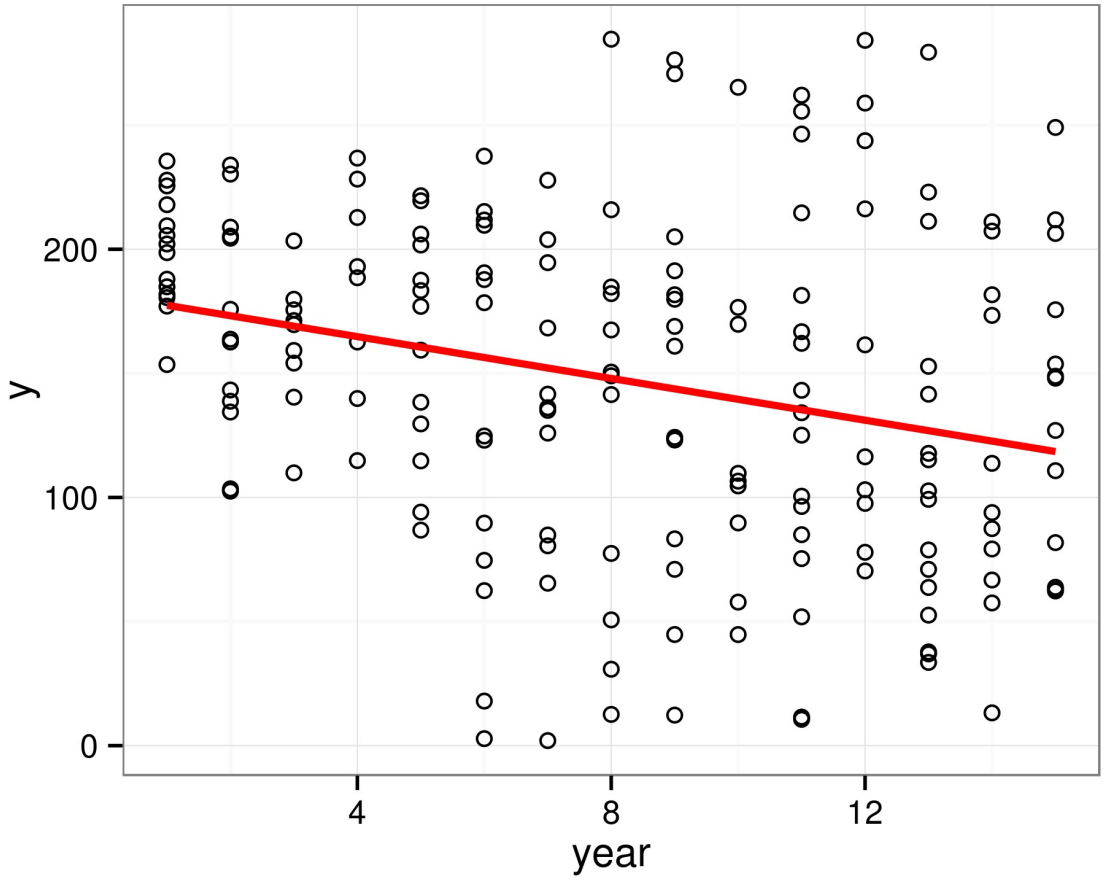
```
> modlm2 <- lm(y ~ year , data = d)
> summary(modlm2)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  181.618     10.044  18.082  < 2e-16 ***
year          -4.209      1.088  -3.869 0.000151 ***
```

Un problème avec cette approche est qu'on ne respecte pas une des hypothèses les plus importantes du modèle : l'indépendance.
En effet, les points d'un même site ne sont vraisemblablement pas indépendants

Modèles Mixtes

Approche 1

```
ggplot(data=d, aes(y = y, x = year)) + geom_point(shape = 1) +
    stat_smooth(method = "lm", se = FALSE, color = "red", lwd = 1) +
    theme_bw()
```



Modèles Mixtes

Approche 1

Un modèle mixte où on ajoute le facteur "site" comme effet aléatoire peut être vu comme une régression classique où on estime automatiquement la corrélation entre observations d'un même groupe (site ici) et où on en tient compte dans le modèle.

Le modèle linéaire classique peut s'écrire comme suit :

$$y_i = X_i\beta + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma^2 I)$$

Matrice nxn avec sigma² sur la diagonale et des 0 partout ailleurs

Le modèle mixte deviens :

$$y_i = X_i\beta + \varepsilon_i$$

$$\varepsilon \sim N(0, \Sigma)$$

Matrice nxn de variance covariance avec la variance sur la diagonale et une covariance identique pour les points au sein d'un même groupe et des 0 partout ailleurs

Approche 1

On utilise le package lme4 pour estimer le modèle mixte :

```
> library(lme4)
> mod <- lmer(y ~ year + (1|site), data = d)
> summary(mod)

Random effects:
 Groups   Name        Variance Std.Dev.
 site     (Intercept) 2785     52.78
 Residual             1483     38.51
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept) 179.8739    11.0789  16.236
year         -4.4922     0.6986  -6.431
```

"Intra-class correlation coefficient" = proportion de variance expliquée par l'effet site = 2785/(2785 + 1483) = 0.65

Matrice de variance covariance des 11 premières observations

Les observations de sites différents sont indépendantes, leur covariance est 0

```
> library(pbkrtest)
> SG <- get_SigmaG(mod)
> round(SG$Sigma, 1)[1:11, 1:11]
11 x 11 sparse Matrix of class "dgCMatrix"

 [1,] 4267.9 2785.2    .      .      .      .      .      .      .      .      .
 [2,] 2785.2 4267.9    .      .      .      .      .      .      .      .      .
 [3,]    .      .   4267.9    .      .      .      .      .      .      .      .
 [4,]    .      .      . 4267.9 2785.2    .      .      .      .      .      .
 [5,]    .      .      . 2785.2 4267.9    .      .      .      .      .      .
 [6,]    .      .      .      .      . 4267.9 2785.2 2785.2    .      .      .
 [7,]    .      .      .      .      . 2785.2 4267.9 2785.2    .      .      .
 [8,]    .      .      .      .      . 2785.2 2785.2 4267.9    .      .      .
 [9,]    .      .      .      .      .      .      .      . 4267.9 2785.2 2785.2
[10,]    .      .      .      .      .      .      .      . 2785.2 4267.9 2785.2
[11,]    .      .      .      .      .      .      .      . 2785.2 2785.2 4267.9
```

2785 + 1483 = 4268

3 observations du site 5

Variance

Covariance

Modèles Mixtes

Approche 1

On peut également avoir des structures de corrélations plus complexes.

Par exemple au sein d'un site la corrélation pourrait être plus importante pour des points plus rapprochés dans le temps.

Il pourrait également y avoir une corrélation entre les points de sites différents par exemple lorsqu'ils sont plus proches spatialement

D'autres méthodes permettent d'incorporer une structure de corrélation sans nécessairement avoir des groupes :
generalized least squares (gls)
generalized estimating equations (gee)
etc...

Modèles Mixtes

Approche 2

Des modèles mixtes comme modèles avec plusieurs termes d'erreur pour diminuer/décomposer la variance résiduelle sans estimer trop de paramètres qui ne nous intéressent pas directement

Un autre problème du modèle y ~ year est qu'il ne tient pas compte de la variabilité entre sites. Certains sites ont vraisemblablement en moyenne plus d'individus que d'autres ce qui augmente la variance résiduelle et diminue donc la précision des estimations.

On pourrait ajouter au modèle y ~ year la variable "site" comme effet fixe et sans l'intercept. Le modèle va alors estimer un intercept différent pour chaque site et une pente commune ce qui va diminuer la variance résiduelle.

Modèles Mixtes

Approche 2

```
> modlm2 <- lm(y ~ year , data = d)
> summary(modlm2)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  181.618     10.044  18.082  < 2e-16 ***
year          -4.209      1.088  -3.869 0.000151 ***

Residual standard error: 64.21 on 184 degrees of freedom
```

Modèle linéaire simple ignorant les sites

```
> modlm <- lm(y ~ site + year -1, data = d)
> summary(modlm)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
sitesite_01  84.5877    27.7363   3.050 0.002709 **
sitesite_02 141.9577    39.7531   3.571 0.000478 ***
sitesite_03 242.3775    28.6225   8.468 2.12e-14 ***
sitesite_04 104.2450    23.6282   4.412 1.95e-05 ***
(...)
sitesite_35  96.1940    15.1024   6.369 2.20e-09 ***
year         -4.4802     0.7067  -6.339 2.56e-09 ***

Residual standard error: 38.5 on 150 degrees of freedom
Multiple R-squared:  0.9542,  Adjusted R-squared:  0.9432
F-statistic: 86.82 on 36 and 150 DF,  p-value: < 2.2e-16

> logLik(modlm)
'log Lik.' -922.9502 (df=37)
```

Modèle linéaire avec un intercept par site.

La variance résiduelle est nettement moindre au prix de l'estimation de 34 paramètres supplémentaires

Modèles Mixtes

Approche 2

Cette approche (modèle fixe) n'est pas mauvaise en soi mais elle nous oblige à avoir dans le modèle un paramètre pour chaque site. Or, on est souvent peu intéressé à comparer les différents sites entre eux.
Ces sites ont été choisis au hasard parmi une population de sites possibles. Si on devait recommencer l'expérience, on utiliserait vraisemblablement des sites différents.
Ce qui nous intéresse c'est de contrôler la variabilité supplémentaire dans les données induite par l'effet "site".

Dans les modèles mixtes, on utilise les intercepts de chaque groupe pour calculer leur variance (et une valeur moyenne) et **on ne garde dans le modèle que ces deux paramètres** qui caractérisent la distribution (gaussienne) de l'effet site.
Ce sont des "**hyper-paramètres**" basés eux-mêmes sur des paramètres.
On considère aussi que les conclusions seront généralisables à d'autres sites contrairement au modèle fixe où les conclusions sont restreintes au 35 sites

Modèles Mixtes

Approche 2

On peut donc présenter les modèles mixtes comme des modèles où on a (au moins) deux termes d'erreur au lieu d'un :

1) les résidus classiques qui sont distribués selon une loi normale avec une moyenne 0 et une variance sigma y caractérisant la variabilité au sein des groupes

2) un terme d'erreur distribué selon une loi normale de moyenne 0 et de variance sigma alpha caractérisant la variabilité entre les groupes

$$y_i = \alpha + X_i\beta + \eta_{j[i]} + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma_y^2)$$

$$\eta_j \sim N(0, \sigma_\alpha^2)$$

Un résidu par observation i

Un "effet aléatoire" pour chaque groupe j

Modèles Mixtes

Approche 2

```
> library(lme4)
> mod <- lmer(y ~ year + (1|site), data = d)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year + (1 | site)
   Data: d

REML criterion at convergence: 1957.178

Random effects:
 Groups   Name        Variance Std.Dev.
 site     (Intercept) 2785     52.78
 Residual             1483     38.51
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept) 179.8739    11.0789  16.236
year         -4.4922     0.6986  -6.431

> logLik(mod)
'log Lik.' -978.589 (df=4)
```

NB : le modèle mixte est strictement identique au modèle de l'approche 1. Seule la manière de l'envisager change

σ_α Variance (et se) entre les sites (inter-sites)

σ_y Variance (et se) au sein des sites (intra-sites)

On réduit bien la variance résiduelle tout en ayant seulement 4 (hyper)-paramètres dans le modèle

NB en pratique on a quand même estimé un intercept pour chaque site ce qui pose parfois problème pour savoir combien de degrés de liberté on doit réellement considérer en pratique

Modèles Mixtes

Approche 2

On peut extraire les "effets aléatoires" qui sont donc les différences de chaque groupe d'observations (sites ici) par rapport à la valeur moyenne (intercept)

$$\eta_j \sim N\left(0, \sigma_\alpha^2\right)$$

```
> ranef(mod)
$site
         (Intercept)
site_01  -75.183682
site_02  -24.633897
site_03   49.482512
site_04  -64.115219
(…)
site_35  -77.701853

attr(,"class")
[1] "ranef.mer"
```

```
> se.ranef(mod)
$site
             [,1]
 [1,] 11.094545
 [2,] 18.334710
 [3,] 11.094545
 [4,]  7.953715
 [5,]  7.953715
 [6,]  5.078373
 [7,]  7.953715
(…)
[35,]  3.729958
```

On peut aussi extraire l'erreur standard de ces effets aléatoires avec la fonction se.ranef du package arm (A.Gelman)
Il y a aussi une copie dans le script mytoolbox.R

```
> round(mean(ranef(mod)$site$"(Intercept)"), 3)
[1] 0
> round(sd(ranef(mod)$site$"(Intercept)"), 1)
[1] 49.7
```

proche de σ_α

Modèles Mixtes

Approche 2

Ce point de vue est typiquement celui que l'on retrouve dans les essais expérimentaux en biologie/agronomie/médecine
et dans les plans expérimentaux classiques :
randomized blocs design, split-plots, latin squares, ...

On veut contrôler la variation due à des groupes d'observations créés par le design expérimental : blocs, champs, sites, portées, hopitaux, chambres climatisées, groupes d'expériences réalisées à différent moments, ...
Mais l'estimation précise des différences entre chaque groupe n'a pas d’intérêt en soi

Modèles Mixtes

Approche 3

Modèles mixtes comme modèles multiniveaux où chaque paramètre (intercept, pentes) peut être estimé de manière optimale pour chaque groupe et peut être considéré comme une variable aléatoire éventuellement modélisée avec des variables explicatives au niveau du groupe tout en pouvant utiliser des variables explicatives au niveau des observations.

C'est sans doute l'approche la plus complexe mais c'est aussi la plus flexible et celle qui offre le plus de possibilités en particulier pour des études observatives complexes.

Dans ce genre d'études on est en général pas intéressé de comparer les groupes entre eux non plus mais il peut-être néanmoins intéressant d'obtenir des estimations non biaisées pour chaque groupe
("BLUPs" : Best Linear Unbiased Predictors)

Modèles Mixtes

Approche 3

Le modèle mixte utilisé jusqu'à présent est appelé "random intercept model" : on estime un intercept pour chaque site, leur variance et leur valeur moyenne.

Un intercept pour chaque groupe j

$$y_i = \alpha_{j[i]} + X_i\beta + \varepsilon_i$$

← Un résidu par observation i

$$\varepsilon \sim N\left(0, \sigma_y^2\right)$$

$$\alpha_j \sim N\left(\mu_\alpha, \sigma_\alpha^2\right)$$

Les intercepts de chaque groupe sont une variable aléatoire de distribution normale avec juste une moyenne et une variance dans ce modèle simple.
La moyenne mu_alpha peut elle même être modélisée en fonction de variables explicatives au niveau du groupe

Modèles Mixtes

Approche 3

La manière d'écrire les modèles est en fait très proche entre l'approche 2 et l'approche 3 vues ici :

Approche 3

$$y_i = \alpha_{j[i]} + X_i\beta + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma_y^2)$$

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$$

Approche 2

$$y_i = \alpha + X_i\beta + \eta_{j[i]} + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma_y^2)$$

$$\eta_j \sim N(0, \sigma_\alpha^2)$$

Estime directement l'intercept de chaque groupe

$$\mu_\alpha = \alpha$$

$$\alpha_{j[i]} = \alpha + \eta_{j[i]}$$

Estime la différence d'intercept entre chaque groupe et la moyenne générale

Modèles Mixtes

Approche 3

`(1|site)` : l'intercept (`1`) varie par (`|`) groupe (`site`)

```
> mod <- lmer(y ~ year + (1|site), data = d)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year + (1 | site)
   Data: d

REML criterion at convergence: 1957.178

Random effects:
 Groups   Name        Variance Std.Dev.
 site     (Intercept) 2785     52.78
 Residual             1483     38.51
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept) 179.8739    11.0789  16.236
year         -4.4922     0.6986  -6.431
```

σ_α

σ_y

μ_α

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$$

$$\alpha_j \sim N(\mu_\alpha = 179.87, \sigma_\alpha^2 = 2785)$$

Modèles Mixtes

Approche 3

On peut extraire les intercepts pour chaque groupe (site)

```
> coef(mod)
$site
         (Intercept)      year
site_01    104.6902 -4.492227
site_02    155.2400 -4.492227
site_03    229.3564 -4.492227
site_04    115.7587 -4.492227
site_05    150.5484 -4.492227
(...)
site_35    102.1720 -4.492227
```

Dans ce modèle la pente est identique pour tous les groupes

Et leur valeur moyenne ("effets fixes") :

```
> fixef(mod)
(Intercept)        year
 179.873871   -4.492227 7
> colMeans(coef(mod)$site)
(Intercept)        year
 179.873871   -4.49222
```

Et on peut vérifier que : $\alpha_{j[i]} = \alpha + \eta_{j[i]}$

```
> fixef(mod)["(Intercept)"] + ranef(mod)$site$"(Intercept)"
 [1] 104.6902 155.2400 229.3564 115.7587 150.5484 (...) 102.1720
```

Modèles Mixtes

Random slope model

On peut aussi estimer un "random slope model" où l'intercept et la pente peuvent varier par site

Un intercept pour chaque groupe j

Une pente pour chaque groupe j

$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i} + \varepsilon_i$$

x_1 = "year" ici

$$\varepsilon \sim N(0, \sigma_y^2)$$

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$$

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2)$$

Modèles Mixtes

`(1+year|site)` : l'intercept (`1`) et la pente (`year`) varient par (`|`) groupe (`site`)

```
> mod <- lmer(y ~ year + (1 + year|site), data = d)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year + (1 + year | site)
   Data: d

REML criterion at convergence: 1919.426

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 site     (Intercept) 2552.93  50.527
          year          30.79   5.549  -0.31
 Residual              894.66  29.911
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept)  176.552     10.483  16.841
year          -4.233      1.149  -3.682
```

σ_α

σ_β

σ_y

μ_α

μ_β

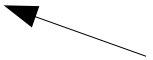
NB : ce "random slope model" est donc équivalent à un modèle avec une interaction entre "year" et l'effet aléatoire "site"

Lorsqu'il y a une pente et un intercept, le modèle estime un paramètre supplémentaire : la corrélation entre les deux

Modèles Mixtes

Pour Info : on peut estimer un modèle sans estimer la corrélation entre les effets aléatoires avec la syntaxe suivante :

```
> mod <- lmer(y ~ year + (1|site) + (0 + year|site), data = d)
> mod <- lmer(y ~ year + (1|site) + (-1 + year|site), data = d) # équivalent
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year + (1 | site) + (-1 + year | site)
   Data: d

REML criterion at convergence: 1920.209

Random effects:
 Groups   Name        Variance Std.Dev.
 site     (Intercept) 1734.67  41.649
 site.1   year          24.84   4.984
 Residual              953.63  30.881
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept)  178.682      9.140   19.55
year          -4.431      1.063   -4.17
```



la corrélation n'est plus estimée

Modèles Mixtes

Random slope model

On pourrait obtenir également un intercept et une pente pour chaque site avec un modèle entièrement fixe en ajoutant le site comme effet fixe et son interaction avec l'année

NB : On a enlevé l'intercept et l'effet year de manière à ce que modèle estime directement l'intercept et la pente de chaque site et pas leurs différences

```
> modlm <- lm(y ~ site + year:site -1, data = d)
> summary(modlm)$coefficients
                     Estimate Std. Error    t value     Pr(>|t|)
sitesite_01       244.4085523 102.767872  2.3782584 1.901444e-02
sitesite_02        79.2346694  28.502679  2.7799025 6.337962e-03
sitesite_03      -435.6348296 504.263874 -0.8639025 3.894091e-01
sitesite_04       -17.6049811 150.436761 -0.1170258 9.070403e-01
sitesite_05        73.4145280 104.576243  0.7020192 4.840614e-01
(...)
sitesite_35       124.6591588  28.086560  4.4383919 2.063550e-05
sitesite_01:year  -25.7896653  13.436292 -1.9194035 5.736863e-02
sitesite_03:year   49.7607714  40.308876  1.2344867 2.194953e-01
sitesite_04:year    6.2712571  13.194177  0.4753049 6.354558e-01
sitesite_05:year    2.2512028   9.681879  0.2325171 8.165426e-01
(...)
sitesite_35:year   -9.4616126   4.539209 -2.0844189 3.929806e-02
```

NB : on obtiendrait exactement les mêmes paramètres en estimant une régression y~year séparément pour chaque site

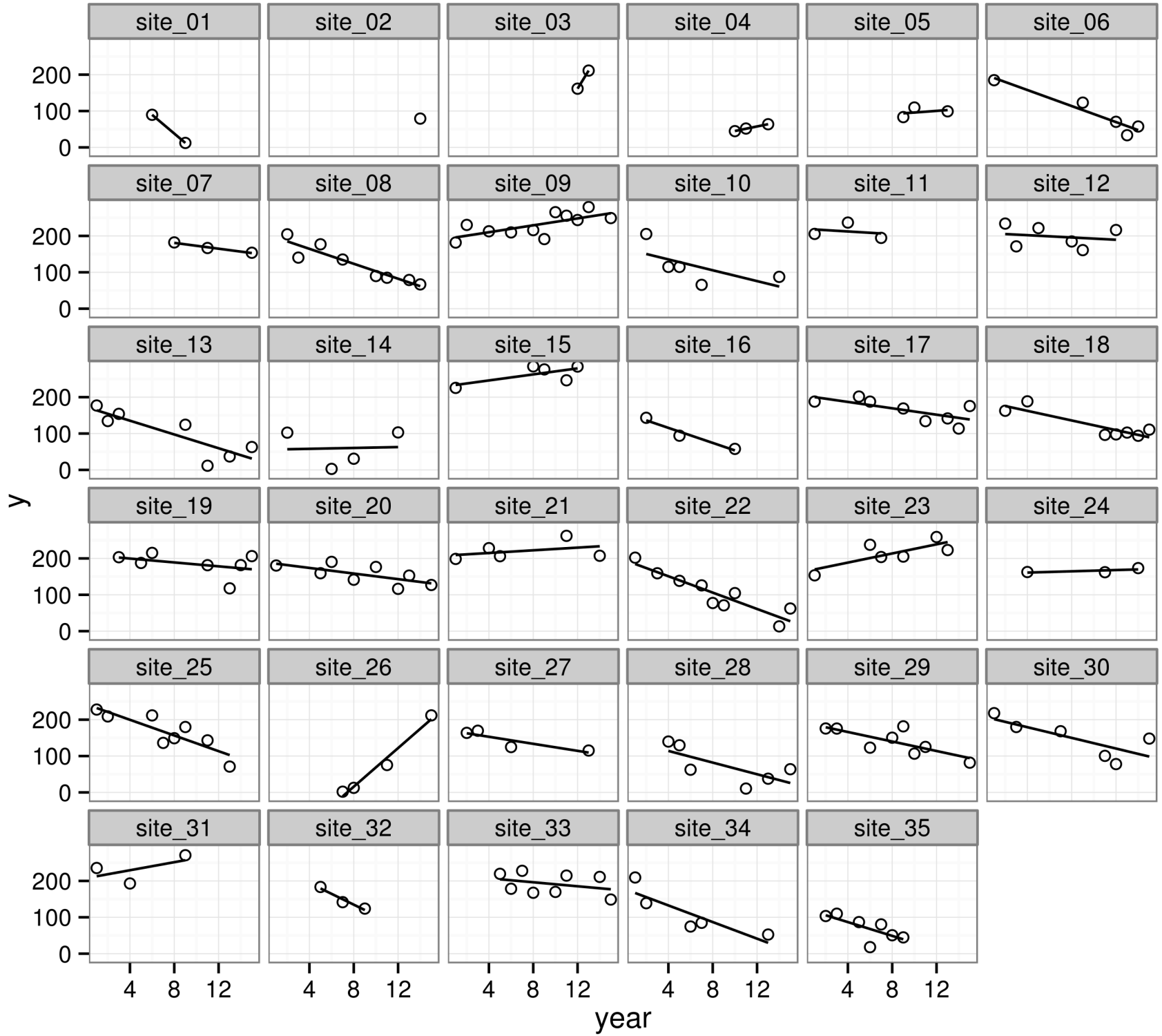
Modèles Mixtes

Random slope model

Représentation graphique du modèle fixe

NB dans ggplot facet_wrap permet de diviser le graphique en sous-graphiques (1 par site) et stat_smooth tel qu'il est spécifié va estimer une droite régression séparée pour chaque sous-graphique identique aux résultats du modèle fixe.

```
ggplot(data=d, aes(y = y, x = year)) +
    geom_point(shape = 1) +
    stat_smooth(method = "lm", se = FALSE, color = "black") +
    facet_wrap(~site) + theme_bw()
```


site_01
site_02
site_03
site_04
site_05
site_06
site_07
site_08
site_09
site_10
site_11
site_12
site_13
site_14
site_15
site_16
site_17
site_18
site_19
site_20
site_21
site_22
site_23
site_24
site_25
site_26
site_27
site_28
site_29
site_30
site_31
site_32
site_33
site_34
site_35
y
200
100
0
4
8
12
year

Modèles Mixtes

BLUPs : Best Linear Unbiased Predictors

Un des problèmes avec cette approche est que les estimations des pentes et des intercepts peuvent être très mauvais en particulier pour les sites où on a peu de données.

Dans le cas du site 2 où on a qu'un point on ne peut même pas estimer de pente.

Modèles Mixtes

BLUPs : Best Linear Unbiased Predictors

Les modèles mixtes vont avoir une meilleure estimation (les fameux "BLUPs") des pentes et des intercepts en estimant une valeur comprise entre deux cas extrêmes :

"Complete pooling" :
C'est le cas où on estime une seule pente et un seul intercept pour l'ensemble des données en ignorant l'effet site : y ~ year (tous les sites sont "poolés", rassemblés dans un seul groupe)

"No Pooling" :
C'est le cas où on estime une droite séparément pour chaque groupe : y ~ year + site + year:site

Modèles Mixtes

BLUPs : Best Linear Unbiased Predictors

Les modèles mixtes utilisent donc un "Partial Pooling" :

La pente et l'intercept d'un groupe vont voir leur valeur d'autant plus "tirée" (shrinkage) vers la valeur "Complete pooling" quand :

- le nombre d'observations du groupe diminue
- la variance intergroupe diminue
- la variance intragroupe augmente

C'est à dire pour ces deux derniers points quand la corrélation entre valeurs d'un même groupe ("intraclass correlation coefficient) diminue.

Modèles Mixtes

BLUPs : Best Linear Unbiased Predictors

Que se passe-t-il dans les cas extrèmes ?

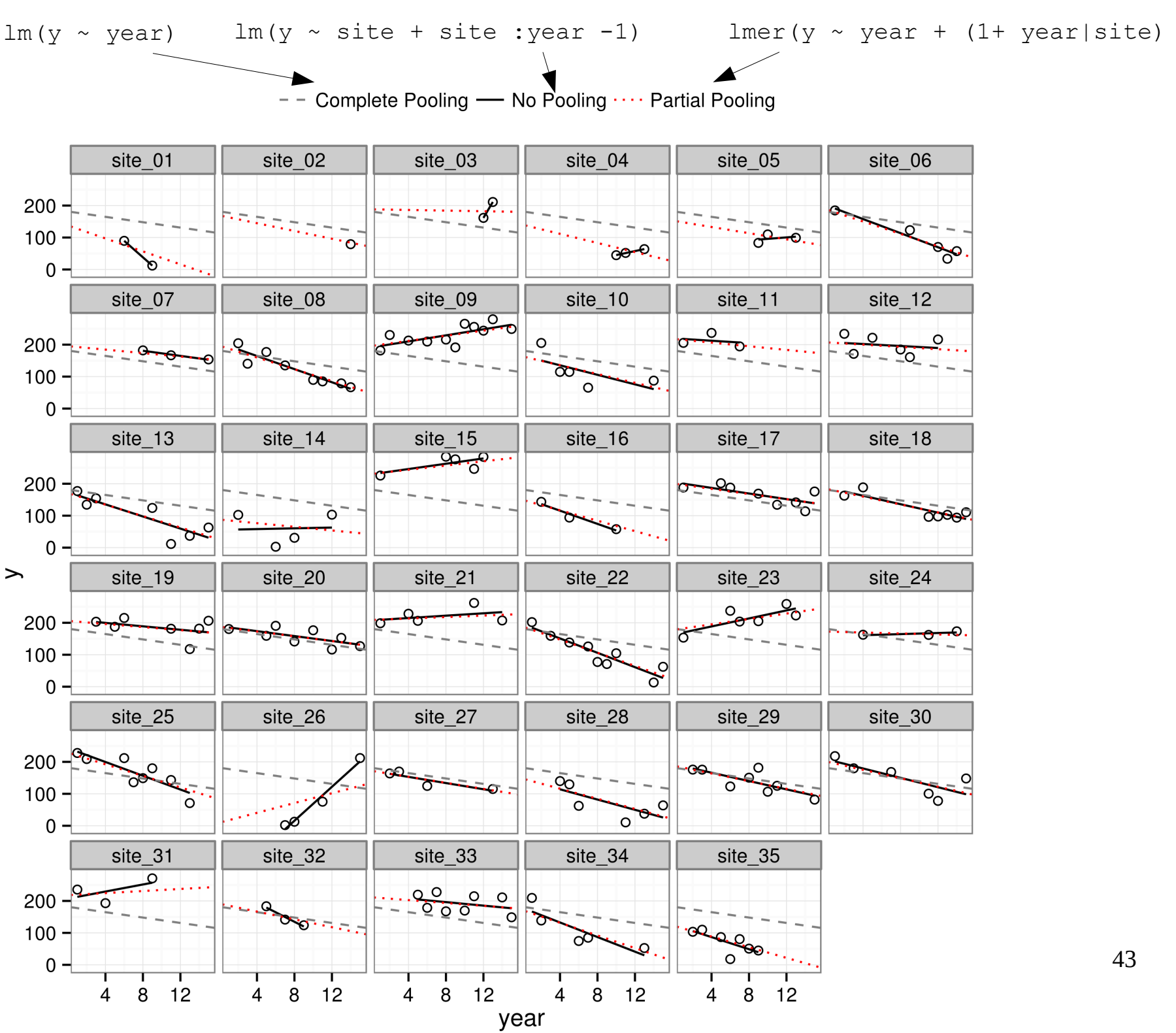
Quand on a aucune observation dans un groupe, la pente et l'intercept de ce groupe sont égaux à ceux du modèle y~year pour ce groupe

Si la variabilité entre les groupes est nulle, la corrélation entre observations d'un même groupe est nulle, les observations sont donc indépendante et le modèle entier se réduit au modèle y~year

Le modèle mixte va toujours utiliser le niveau idéal de pondération entre les deux valeurs "no pooling" et "complete pooling" en fonction des données et de leur corrélation.

code pour le graphique de la dia suivante

```
# No pooling : une droite séparée pour chaque groupe
modlm <- lm(y ~ site + site:year -1, data = d)
nopooling <- data.frame(
    int = coef(modlm)[1:nsites],
    slope = coef(modlm)[(nsites+1) : length(coef(modlm))],
    site = levels(d$site))
seint = summary(modlm)$coefficients[1:nsites, 2],
seslope = summary(modlm)$coefficients[(nsites+1) : nrow(summary(modlm)$coefficients), 2]

# complete pooling : modèle sans tenir compte des groupes
modlm2 <- lm(y ~ year , data = d)

# modèle mixte random slope
mod <- lmer(y ~ year + (1+year|site), data = d)
partialpooling <- coef(mod)$site
partialpooling$site <- row.names(partialpooling)
colnames(partialpooling) <- c("int", "slope", "site")

dev.new(18/2.54,18/2.54)
ggplot(data=d, aes(y = y, x = year)) +
    geom_point(shape = 1) +
    stat_smooth(method = "lm", se = FALSE, mapping =
                aes( linetype = "No Pooling", color = "No Pooling")) +
    geom_abline(intercept = coef(modlm2)[1], slope = coef(modlm2)[2],
                mapping = aes(linetype = "Complete Pooling", color = "Complete Pooling")) +
    geom_abline(data = partialpooling,
                aes(intercept = int, slope = slope, linetype = "Partial Pooling",
                    color = "Partial Pooling")) +
    scale_linetype_manual(name = "", values = c(2,1,3),
                          breaks = c("Complete Pooling","No Pooling", "Partial Pooling")) +
    scale_color_manual(name = "", values = c("grey50","black","red"),
                          breaks = c("Complete Pooling","No Pooling", "Partial Pooling")) +
    facet_wrap(~site) +
    theme_bw() + theme(legend.position = "top", legend.key = element_rect(color = NA))

savepng()
```


lm(y ~ year)
lm(y ~ site + site :year -1)
lmer(y ~ year + (1+ year|site)
Complete Pooling
No Pooling
Partial Pooling
site_01
site_02
site_03
site_04
site_05
site_06
site_07
site_08
site_09
site_10
site_11
site_12
site_13
site_14
site_15
site_16
site_17
site_18
site_19
site_20
site_21
site_22
site_23
site_24
site_25
site_26
site_27
site_28
site_29
site_30
site_31
site_32
site_33
site_34
site_35
y
year
0
100
200
4
8
12

code pour le graphique de la dia suivante

```
modlm <- lm(y ~ site + site:year -1, data = d)
modlm2 <- lm(y ~ year , data = d)
mod <- lmer(y ~ year + (1 + year|site), data = d)
nbobs <- data.frame(aggregate(d["y"], d["site"], length))

int <- summary(modlm)$coefficients[1:nsites, 1]
se <-  summary(modlm)$coefficients[1:nsites, 2]
l = (int-se) ;u = (int+se)
nb <- jitter(nbobs$y, amount=0.25)

dev.new(16/2.54, 16/2.54)
par(mfrow = c(2,2), mar = c(3,3,3,1), mgp = c(1.75, 0.5, 0))
plot(y = int, x= nb, pch = 20, cex = 0.6, ylim = c(-150,300),
     ylab = "Intercept \u00B1 se", xlab = "Number of observations",
     main = "No Pooling Estimates")
abline(h = (coef(modlm2)[1]))
segments(x0=nb, y0=l, x1=nb, y1=u)

int <- coef(mod)$site$"(Intercept)"
se <-  se.ranef(mod)$site[,1]
l = (int-se) ;u = (int+se)

plot(y = int, x= nb, pch = 20, cex = 0.6, ylim = c(-150,300),
     ylab = "Intercept \u00B1 se", xlab = "Number of observations",
     main = "Partial Pooling Estimates")
abline(h = (coef(modlm2)[1]))
segments(x0=nb, y0=l, x1=nb, y1=u)
```

```
n <- nrow(summary(modlm)$coefficients)
int <- summary(modlm)$coefficients[(nsites+1):n, 1]
se <-  summary(modlm)$coefficients[(nsites+1):n, 2]
l = (int-se) ;u = (int+se)
nb2 <- nb[nbobs$y>1]

plot(y = int, x= nb2, pch = 20, cex = 0.6, ylim = c(-40,40),
     ylab = "Slope \u00B1 se", xlab = "Number of observations",
     main = "No Pooling Estimates")
abline(h = (coef(modlm2)[2]))
segments(x0=nb2, y0=l, x1=nb2, y1=u)

int <- coef(mod)$site[,2]
se <-  se.ranef(mod)$site[,2]
l = (int-se) ;u = (int+se)

plot(y = int, x= nb, pch = 20, cex = 0.6, ylim = c(-40,40),
     ylab = "Slope \u00B1 se", xlab = "Number of observations",
     main = "Partial Pooling Estimates")
abline(h = (coef(modlm2)[2]))
segments(x0=nb, y0=l, x1=nb, y1=u)

savepng()
```

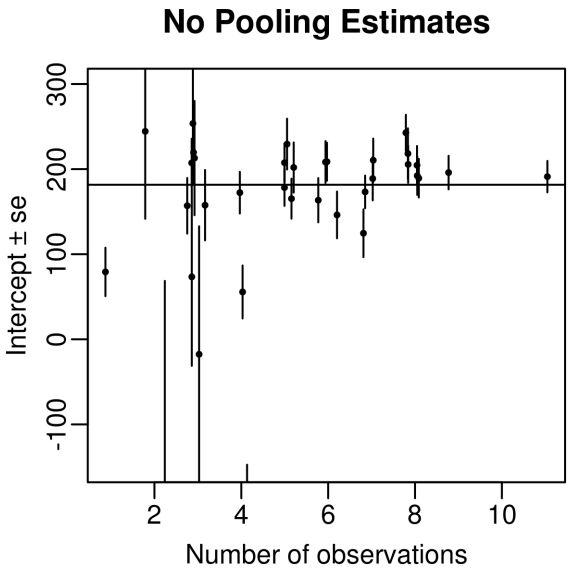
Modèles Mixtes

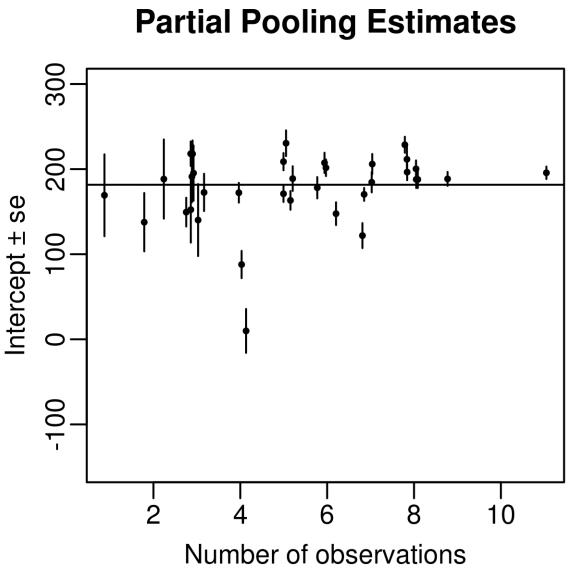
BLUPs : Best Linear Unbiased Predictors

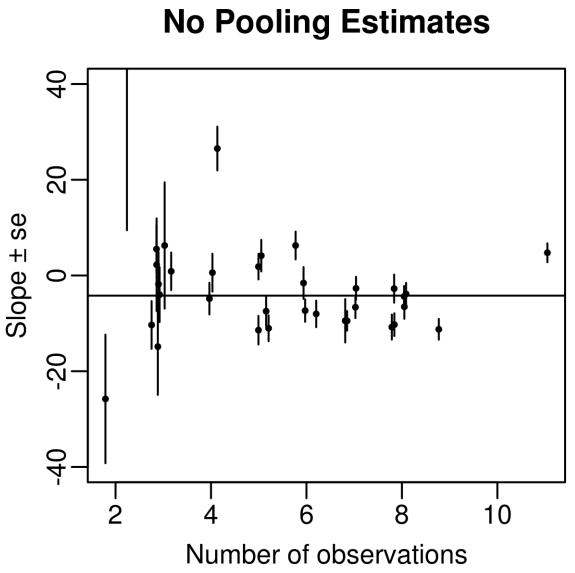
La ligne horizontale représente la valeur estimée par le modèle ignorant les sites (complete pooling)

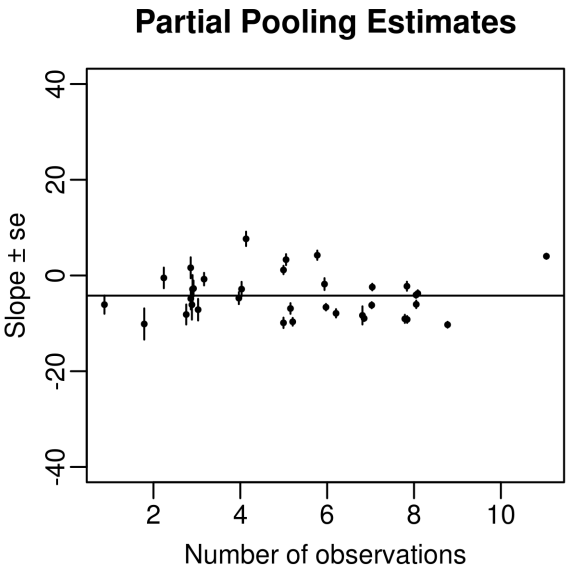
Plus le nombre d'observations dans un groupe est petit plus les estimations sont mauvaises (grande erreur standard) avec le modèle fixe (no pooling)

Les modèles mixtes (partial pooling) "tirent" les valeurs de ces groupes mal estimés vers la droite horizontale, ie la valeur du modèle ignorant les sites (complete pooling).
Et les estimations sont plus précises (erreur standard plus faible)









Modèles Mixtes

BLUPs : Best Linear Unbiased Predictors

Cette aptitude des modèles mixtes à obtenir des estimateurs tenant compte de la qualité de l'information contenue à tous les niveaux est une des raisons pour laquelle certaines personnes pensent qu'on devrait considérer pratiquement tous les variables qualitatives a priori fixes (parce qu'on veut comparer les différents niveaux) comme des effets aléatoires en particulier dans des designs non balancés/des études observatives.

Modèles Mixtes

Random slope model : simulation des données

```
nsites <- 35 ; ny <- 15 ; n <- nsites * ny

# création des variables site et year
site <- paste("site", rep(sprintf("%02.0f", 1:nsites), each = ny), sep = "_")
year <- rep(1:ny, times = nsites)

# moyenne et variance des pentes et variance résiduelle
int.mean <- 200
int.sd <- 30
slope.mean <- -10
slope.sd <- 6
sigma <- 25

# Génération des pentes et des intercepts pour chaque groupe
set.seed(1)
int <- rnorm(n = nsites, mean = int.mean, sd = int.sd )
set.seed(2)
slope <- rnorm(n = nsites, mean = slope.mean, sd = slope.sd )
gamma <- c(int, slope)

X <- model.matrix (~ site + site : year - 1)
lin.pred <- X %*% gamma
set.seed(3)
y <- abs(rnorm(n = n, mean = lin.pred, sd = sigma))

d <- data.frame(y, site, year)
# On élimine une bonne partie des données pour créer un jeu de données non balancé
set.seed(234)
d <- d[c(6, 9, 29, 42, 43, 55, 56, 58, 69, 70, 73, sample(x=76:nrow(d), nrow(d)/3)),]
```

μ_α σ_α μ_β σ_β σ_y

$$\alpha_j \sim N\left(\mu_\alpha, \sigma_\alpha^2\right)$$

$$\beta_j \sim N\left(\mu_\beta, \sigma_\beta^2\right)$$

$$\hat{y}_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i}$$

$$y_i \sim N\left(\hat{y}_i, \sigma_y^2\right)$$

Modèles Mixtes

Random slope model : avec covariance

Comme on l'a vu dans les estimations de lmer, il peut exister une corrélation (ou covariance) entre la pente et l'intercept

$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i} + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma_y^2)$$

Distribution Normale Multivariée

$$(\alpha_j, \beta_j) \sim MVN(\mu, \Sigma)$$

$$\mu = (\mu_\alpha, \mu_\beta)$$

$$\Sigma = \begin{pmatrix} \sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2 \end{pmatrix} = \begin{pmatrix} \sigma_\alpha^2 & \rho\,\sigma_\alpha\sigma_\beta \\ \rho\,\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix}$$

Matrice de Variance covariance des effets aléatoires

covariance

correlation

Random slope model : avec covariance

```
nsites <- 35 ; ny <- 15 ; n <- nsites * ny
site <- paste("site", rep(sprintf("%02.0f", 1:nsites), each = ny), sep = "_")
year <- rep(1:ny, times = nsites)

# moyenne et variance des pentes et variance résiduelle
int.mean <- 200
int.sd <- 30
slope.mean <- -10
slope.sd <- 6
sigma <- 25
cor <- -0.5
covar <- cor * int.sd * slope.sd
mu <- c(int.mean, slope.mean)
vcovmat <- matrix(c(int.sd^2, covar, covar, slope.sd^2), 2,2)

library(MASS)
set.seed(1)
effects <- mvrnorm(n = nsites, mu = mu, Sigma = vcovmat)
int <- effects[,1]
slope <- effects[,2]
gamma <- c(int, slope)

X <- model.matrix (~ site + site : year - 1)
lin.pred <- X %*% gamma
set.seed(3)
y <- abs(rnorm(n = n, mean = lin.pred, sd = sigma))
```

μ_α σ_α μ_β σ_β σ_y ρ $\sigma_{\alpha\beta}$

$$\mu = (\mu_\alpha, \mu_\beta) \qquad \Sigma = \begin{pmatrix} \sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2 \end{pmatrix}$$

$$(\alpha_j, \beta_j) \sim MVN(\mu, \Sigma)$$

$$\hat{y}_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i}$$

$$y_i \sim N(\hat{y}_i, \sigma_y^2)$$

NB : on utilise abs() pour éviter les valeurs négatives... Solution très bancale ! Il vaudrait mieux utiliser un modèle de Poisson avec lien log !!!

```
d <- data.frame(y, site, year)
# On élimine une bonne partie des données pour créer un jeu de données non balancé
set.seed(234)
d <- d[c(6, 9, 29, 42, 43, 55, 56, 58, 69, 70, 73, sample(x=76:nrow(d), nrow(d)/3)),]
```

Modèles Mixtes

Random slope model : avec covariance

```
> mod <- lmer(y ~ year + (1 + year|site), data = d)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year + (1 + year | site)
   Data: d

REML criterion at convergence: 1899.96

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 site     (Intercept) 2760.50  52.540
          year          30.72   5.543   -0.78
 Residual              939.51  30.651
Number of obs: 186, groups: site, 35

Fixed effects:
            Estimate Std. Error t value
(Intercept)  175.960     10.910  16.128
year          -7.677      1.159  -6.626

Correlation of Fixed Effects:
     (Intr)
year -0.821
```

NB : lmer estime la corrélation ("rho"), pas la covariance

NB : la corrélation est encore plus difficile à estimer que la variance des effets aléatoires --> il faut beaucoup de groupes et/ou de données par groupe pour l'estimer précisément

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

On pourrait avoir d'autres variables explicatives soit au niveau des observations soit au niveau des sites :

La température moyenne lors des inventaires est une variable explicative (predictor) au niveau des observations (elle varie pour chaque observation et chaque site) qui pourrait expliquer en partie la variation d'une année à l'autre

La taille du site est une variable explicative au niveau des sites (elle est la même pour toutes les observations d'un même site) qui pourrait expliquer en partie les différences d'intercept

De telles variables permettent d'avoir une meilleure estimation des différents paramètres

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

Mais certaines variables explicatives au niveau du site peuvent aussi être particulièrement intéressantes pour comprendre le système étudié.

Par exemple on pourrait se demander si le mode de gestion (pex pâturage vs fauchage) a une influence sur la pente, c'est à dire la tendance des populations

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

Le modèle devient alors :

$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i} + \varepsilon_i$$

$$\varepsilon \sim N(0, \sigma_y^2)$$

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$$

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2)$$

$$\mu_\beta = \gamma_0 + \gamma_1 u_{1j}$$

La pente moyenne est elle même modélisée en fonction ici de u1 qui représente le type de gestion

"dummy variable" 0/1 pour les sites fauchés

pente pour les sites pâturés fixée à -5 dans la simulation des données

différence de pente pour les sites fauchés = -10
leur pente est donc = -5-10 = -15

Simulation du jeu de données

```
nsites <- 50 ; ny <- 15 ; n <- nsites * ny

# création des variables explicatives
site <- paste("site", rep(sprintf("%02.0f", 1:nsites), each = ny), sep = "_")
year <- rep(1:ny, times = nsites)
man_site <- rep(c("grazing", "mowing"), each = nsites/2)
man <- rep(man_site, each = ny)

# moyenne et variance des pentes et variance résiduelle
int.mean <- 200   μα
int.sd <- 30      σα
sigma <- 25       σy

g0 <- -5 # pente moyenne pour les sites pâturés
g1 <- -10 # différence de pente pour les sites fauchés
U <- model.matrix(~ 1 + man_site)
slope.mean <- U %*% c(g0, g1)
slope.sd <- 6     σβ

# Génération des pentes et des intercepts pour chaque groupe
set.seed(1)
int <- rnorm(n = nsites, mean = int.mean, sd = int.sd )
set.seed(2)
slope <- rnorm(n = nsites, mean = slope.mean, sd = slope.sd )
gamma <- c(int, slope)

X <- model.matrix (~ site + site : year - 1)
lin.pred <- X %*% gamma
set.seed(3)
y <- rnorm(n = n, mean = lin.pred, sd = sigma)
y <- ifelse(y<=0, 0, y)

d <- data.frame(y, site, year, man)
set.seed(234)
d <- d[sample(x=1:nrow(d), nrow(d)/3),]
d <- d[order(d$site, d$year),]
```

$$\mu_\beta = \gamma_0 + \gamma_1 u_{1j}$$

$$\alpha_j \sim N\left(\mu_\alpha, \sigma_\alpha^2\right)$$

$$\beta_j \sim N\left(\mu_\beta, \sigma_\beta^2\right)$$

$$\hat{y}_i = \alpha_{j[i]} + \beta_{j[i]} x_{1i}$$

$$y_i \sim N\left(\hat{y}_i, \sigma_y^2\right)$$

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

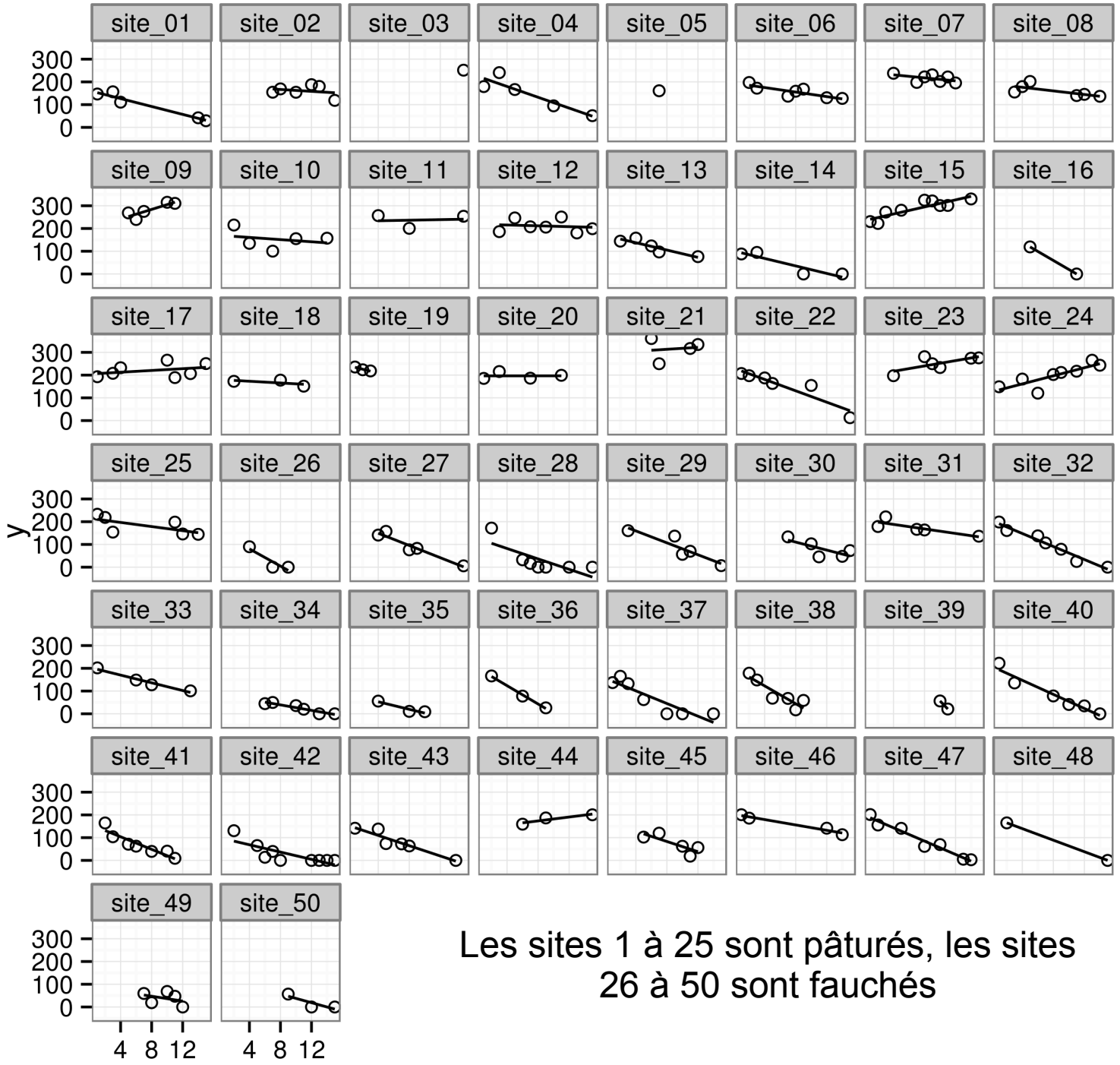
Jeu de données simulé

```
              y    site year     man
1   146.776563 site_01    1 grazing
3   156.531629 site_01    3 grazing
4   110.877139 site_01    4 grazing
14   42.181873 site_01   14 grazing
15   29.285219 site_01   15 grazing
22  154.715447 site_02    7 grazing
23  169.288856 site_02    8 grazing
25  154.488873 site_02   10 grazing
27  187.833836 site_02   12 grazing
28  180.229214 site_02   13 grazing
30  118.726169 site_02   15 grazing
45  251.548164 site_03   15 grazing
(...)
379  89.920354 site_26    4  mowing
382   0.000000 site_26    7  mowing
384   0.000000 site_26    9  mowing
394 141.215730 site_27    4  mowing
395 157.624286 site_27    5  mowing
398  76.389153 site_27    8  mowing
399  82.776707 site_27    9  mowing
405   5.956027 site_27   15  mowing
407 171.938586 site_28    2  mowing
(...)
```

Pour l'utilisation dans lmer, les variables explicatives au niveau des groupes (man ici) sont simplement répétées pour chaque observation

Présentation graphique :

```
ggplot(data=d, aes(y = y, x = year)) +
    geom_point(shape = 1) +
    stat_smooth(method = "lm", se = FALSE,
        color = "black") +
    facet_wrap(~site) + theme_bw()
```



Les sites 1 à 25 sont pâturés, les sites 26 à 50 sont fauchés

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

analyse avec lmer

```
> mod <- lmer(y ~ year*man + (1 + year|site),
              data = d)
> summary(mod)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ year * man + (1 + year | site)
   Data: d

REML criterion at convergence: 2506.15

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 site     (Intercept) 943.19   30.71
          year         27.56    5.25    0.19
 Residual             675.75   26.00
Number of obs: 250, groups: site, 50

Fixed effects:
               Estimate Std. Error t value
(Intercept)     198.797      8.232  24.148
year             -2.029      1.242  -1.634
manmowing       -26.970     11.851  -2.276
year:manmowing   -9.467      1.755  -5.395

Correlation of Fixed Effects:
            (Intr) year   mnmwng
year        -0.180
manmowing   -0.695  0.125
year:mnmwng  0.127 -0.708 -0.197
```

Le nombre moyen pour l'ensemble des sites pâturés l'année 0 est estimé à 198.8 individus (plus précisément: loi normale de moyenne 198.8 et sd 30.71)

Pour les sites fauchés cette valeur est de 198.8 - 26.97

NB : on sait que la vraie valeur est en fait 0 dans ces données simulées

La tendance pour les sites pâturés est une variable aléatoire normale de moyenne -2.029 ± 1.24 et d'erreur standard 5.25 (vraies valeurs : -5 et 6).

On estime donc que sur les sites pâturés on perd en moyenne 2 individus par an ± 5.25 individus selon les sites

Sur les sites fauchés la tendance moyenne est par contre de -2.03 - 9.47 = -11.5 avec également une erreur standard de 5.25 (vraies valeurs = -15 et 6)

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

Une analyse similaire avec un modèle fixe où on mélange des une variable explicative "site" et des variables explicatives au niveau du site est impossible ou très difficile à interpréter

```
> modlm <- lm(y ~ year*man*site , data = d)
> summary(modlm)
Coefficients: (102 not defined because of singularities)
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  161.39442   18.90438   8.537 1.32e-14 ***
year                          -8.68432    1.99937  -4.344 2.55e-05 ***
manmowing                    -28.83730   78.27571  -0.368 0.713083
sitesite_02                   21.95606   46.98712   0.467 0.640970
sitesite_03                  220.41848   32.58425   6.765 2.71e-10 ***
(...)
year:manmowing                -0.78405    6.51630  -0.120 0.904387
year:sitesite_02               6.61015    4.33372   1.525 0.129266
year:sitesite_03                    NA         NA      NA       NA
(...)
manmowing:sitesite_02               NA         NA      NA       NA
manmowing:sitesite_03               NA         NA      NA       NA
manmowing:sitesite_04               NA         NA      NA       NA
(...)
year:manmowing:sitesite_02          NA         NA      NA       NA
year:manmowing:sitesite_03          NA         NA      NA       NA
year:manmowing:sitesite_04          NA         NA      NA       NA
year:manmowing:sitesite_05          NA         NA      NA       NA
```

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

Une analyse similaire avec un modèle fixe où on mélange des une variable explicative "site" et des variables explicatives au niveau du site est impossible ou très difficile à interpréter

```
> modlm <- lm(y ~ year*man + year*site , data = d)
> summary(modlm)
Coefficients: (4 not defined because of singularities)
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)      161.39442   18.90438   8.537 1.32e-14 ***
year              -8.68432    1.99937  -4.344 2.55e-05 ***
manmowing        -28.83730   78.27571  -0.368 0.713083
sitesite_02       21.95606   46.98712   0.467 0.640970
sitesite_03      220.41848   32.58425   6.765 2.71e-10 ***
sitesite_04       65.96492   27.28392   2.418 0.016801 *
(...)
year:manmowing    -0.78405    6.51630  -0.120 0.904387
year:sitesite_02   6.61015    4.33372   1.525 0.129266
year:sitesite_03        NA         NA      NA       NA
year:sitesite_04  -3.19386    3.06154  -1.043 0.298502
year:sitesite_05        NA         NA      NA       NA
year:sitesite_06   3.69926    3.16631   1.168 0.244508
```

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

"Two stage analysis"

Une approche possible avec des modèles fixes est de procéder en deux étapes (two stage analysis) :
On estime les pentes de chaque site avec un premier modèle puis on compare les pentes obtenues en fonction des caractéristiques des sites (le type de gestion ici).

Les désavantages de cette approche sont entre autres que les pentes seront mal estimées pour certains groupes et qu'on ne peut pas intégrer en même temps des variables explicatives au niveau des prédictions et des groupes

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

"Two stage analysis"

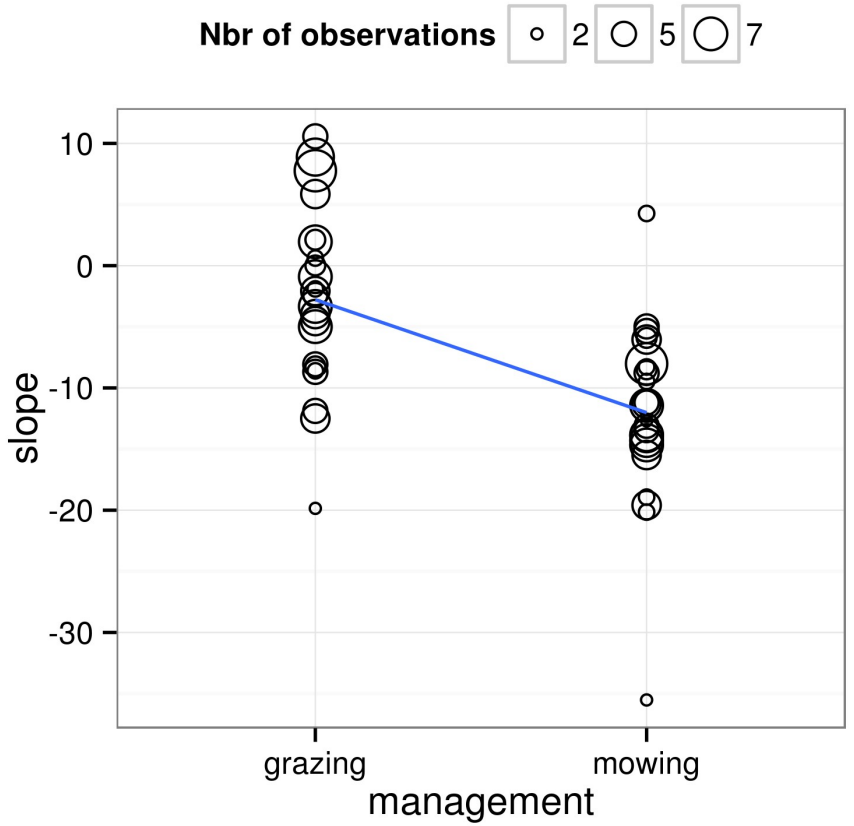
```
> modlm <- lm(y ~ -1 + site + year:site , data = d)
> summary(modlm)
Coefficients: (2 not defined because of singularities)
                   Estimate Std. Error t value Pr(>|t|)
sitesite_01      161.39442   18.90438   8.537 1.32e-14 ***
sitesite_02      183.35048   43.01644   4.262 3.54e-05 ***
sitesite_03      251.54816   26.31283   9.560  < 2e-16 ***
(...)
sitesite_01:year  -8.68432    1.99937  -4.344 2.55e-05 ***
sitesite_02:year  -2.07416    3.84494  -0.539 0.590365
sitesite_03:year        NA         NA      NA       NA
sitesite_04:year -11.87818    2.31851  -5.123 8.99e-07 ***
(...)

> d2 <- aggregate(list(nb = d$y), d[,c("site", "man")], length)
> d2$slope <- coef(modlm)[51:100]
> d2
      site     man nb        slope
1  site_01 grazing  5  -8.68431535
2  site_02 grazing  6  -2.07416138
3  site_03 grazing  1           NA
4  site_04 grazing  5 -11.87817554
5  site_05 grazing  1           NA
6  site_06 grazing  7  -4.98505706
7  site_07 grazing  7  -3.33447167
```

Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux

"Two stage analysis"

```
ggplot(d2, aes(y = slope, x = man)) +
    geom_point(aes(size = nb),shape = 1) +
    stat_smooth(method = "lm", se = FALSE, aes(group = 1)) +
    scale_size_continuous(name = "Nbr of observations", breaks = c(2,5,7)) +
    xlab("management") +
    theme_bw() +
    theme(legend.position = "top")
```



Modèles Mixtes

Approche 3 (suite) : modèles multiniveaux
"Two stage analysis"

Modèle linéaire équivalent à un test de Student

```
> mod <- lm(slope ~ man, data = d2)
> summary(mod)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -2.786      1.515  -1.839   0.0723 .
manmowing     -9.236      2.099  -4.400 6.36e-05 ***
```

On peut éventuellement utiliser l'argument weight pour donner plus de poids aux pentes estimées sur base de plus de points

```
> mod <- lm(slope ~ man, weights = nb, data = d2)
> summary(mod)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.700      1.312  -1.295    0.202
manmowing     -9.976      1.848  -5.397 2.29e-06 ***
```

Modèles Mixtes Généralisés

Comme pour les modèles linéaires classiques, on peut également utiliser d'autres distributions des résidus que la Gaussienne : Poisson, binomial, etc...

Dans le package `lme4` on utilise la fonction `glmer()` dont l'utilisation est assez similaire à celle de `glm()` et de `lmer()` combinées

`glmer` pour Generalized Linear Mixed Effects models with R

On utilise souvent l'acronyme GLMM :
Generalized Linear Mixed Models

Les intercepts et pentes sont en revanche (presque) toujours modélisés selon une distribution Normale.

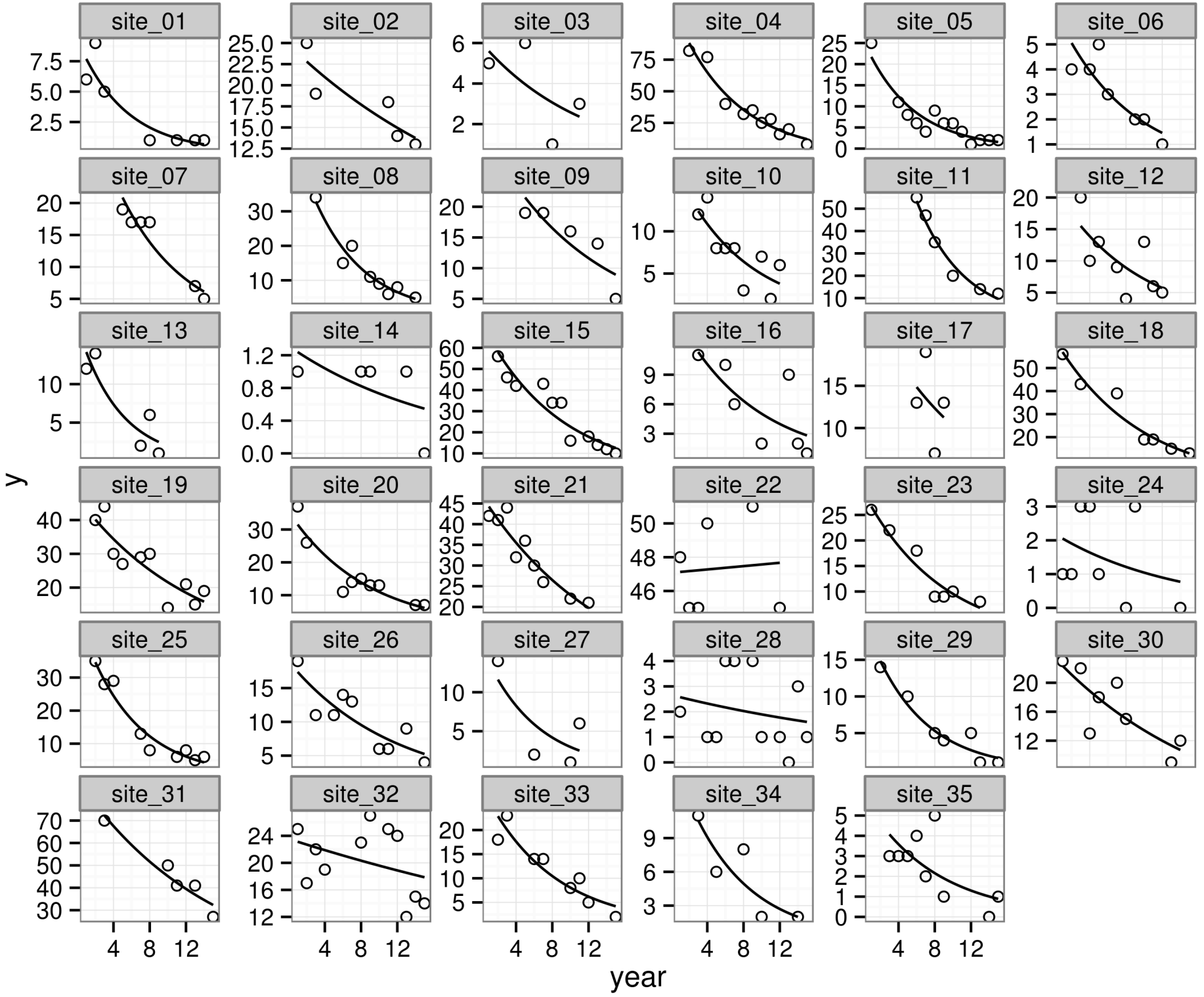
Modèles Mixtes Généralisés

Simulation d'un jeu de donnée avec une distribution de Poisson (plus réaliste dans notre exemple)

```
nsites <- 35 ; ny <- 15 ; n <- nsites * ny

# création des variables site et year
site <- paste("site", rep(sprintf("%02.0f", 1:nsites), each = ny), sep = "_")
year <- rep(1:ny, times = nsites)

# moyenne et variance des pentes et variance résiduelle
int.mean <- log(20) # une vingtaine d'individus en moyenne l'année 0
int.sd <- log(3) # une forte variation entre sites
slope.mean <- log(0.9) # 10% de diminution par an
slope.sd <- (log(1.05)) # une variation des pentes de +ou- 5 %

# Génération des pentes et des intercepts pour chaque groupe
set.seed(1)
int <- rnorm(n = nsites, mean = int.mean, sd = int.sd )
set.seed(12)
slope <- rnorm(n = nsites, mean = slope.mean, sd = slope.sd )
gamma <- c(int, slope)

X <- model.matrix (~ site + site : year - 1)
lin.pred <- exp(X %*% gamma)
set.seed(3)
y <- rpois(n = n, lambda = lin.pred)

d <- data.frame(y, site, year)
set.seed(234)
d <- d[sample(x=1:nrow(d), nrow(d)/2),]
```

Avec une échelle log, les relations sont multiplicatives...

```
ggplot(data=d, aes(y = y, x = year)) +
    geom_point(shape = 1) +
    stat_smooth(method = "glm",
        family = poisson,
        se = FALSE, color = "black") +
    facet_wrap(~site, scales = "free_y")+
    theme_bw()
```


site_01
site_02
site_03
site_04
site_05
site_06
site_07
site_08
site_09
site_10
site_11
site_12
site_13
site_14
site_15
site_16
site_17
site_18
site_19
site_20
site_21
site_22
site_23
site_24
site_25
site_26
site_27
site_28
site_29
site_30
site_31
site_32
site_33
site_34
site_35
y
year

Modèles Mixtes Généralisés

Analyse avec `glmer`

```
> mod <- glmer(y ~ year + (1 + year|site), data = d, family = poisson)
> summary(mod)
Generalized linear mixed model fit by maximum likelihood ['glmerMod']
 Family: poisson ( log )
Formula: y ~ year + (1 + year | site)
   Data: d

      AIC       BIC    logLik  deviance
1501.0770 1518.9187 -745.5385 1491.0770

Random effects:
 Groups Name        Variance Std.Dev. Corr
 site   (Intercept) 0.91807  0.95816
        year        0.00208  0.04561  0.01
Number of obs: 262, groups: site, 35

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.13155    0.16770   18.67   <2e-16 ***
year        -0.11369    0.00966  -11.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
     (Intr)
year -0.116
```

Inférence et construction du modèle

Les tests d'hypothèses et les estimations d'intervalles de confiance pour les modèles mixtes sont un sujet difficile et controversé...

En général on procède en deux étapes (en particulier pour les données expérimentales):
D'abord on choisi la structure de la partie aléatoire du modèle, ensuite on s'intéresse à la partie fixe et on ne touche plus à la partie aléatoire

Mais on part aussi souvent d'un modèle simple que l'on complexifie progressivement en fonction des données, de leur quantité et de notre capacité à interpréter un modèle plus complexe

NB : Il n'existe vraiment pas de règles absolues dans ce qui va suivre mais plutôt une série de pratiques plus ou moins largement répandues et sujettes à controverses...

Inférence et construction du modèle

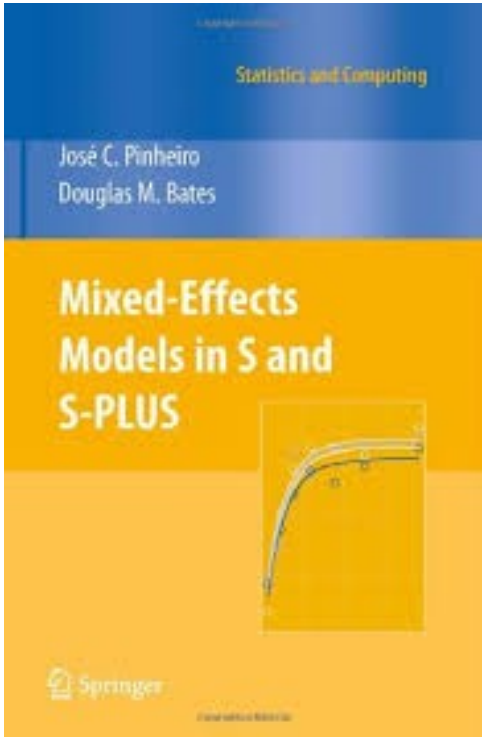
Quelques ressources utiles en particulier pour ce qui concerne les problèmes de tests d'hypothèses

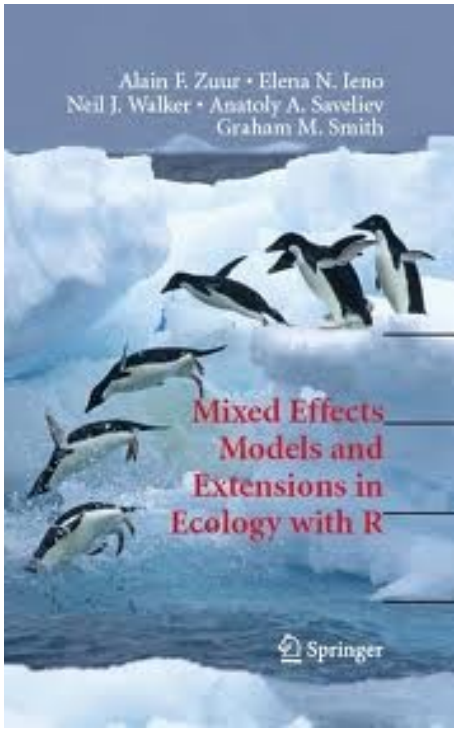
?pvalues (aide du package lme4)

Synthèse très très utile des échanges sur les mailing lists de R et autres sources :
http://glmm.wikidot.com/faq

Inférence et construction du modèle

Quelques ressources utiles en particulier pour ce qui concerne les stratégies de construction des modèles (ea de la partie aléatoire)





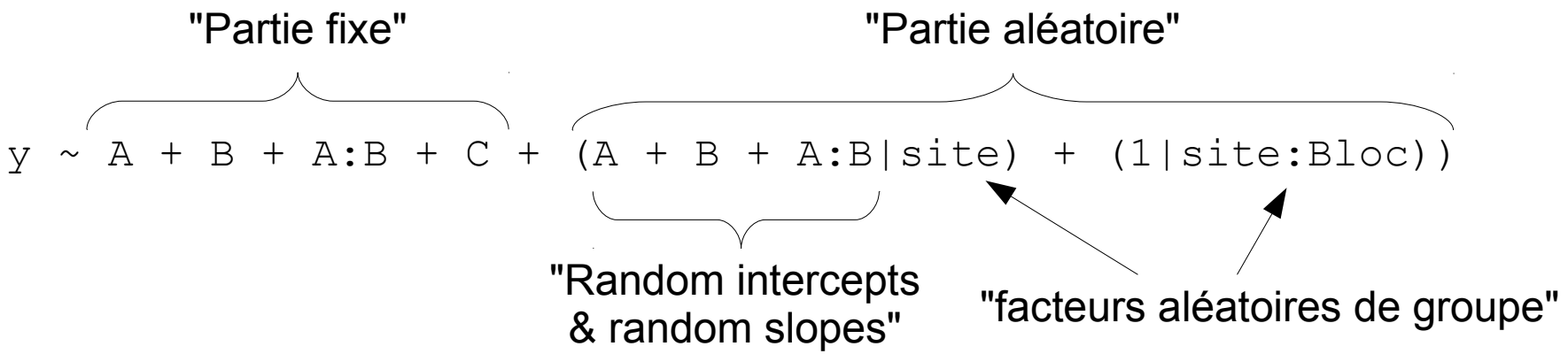
Un bel exemple concret et bien documenté : Lancelot, R., Lesnoff, M., Tillard, E., Mcdermott, J.J., 2000. Graphical approaches to support the analysis of linear-multilevel models of lamb pre-weaning growth in Kolda (Senegal). Prev. Vet. Med. 46, 225-247

http://forums.cirad.fr/logiciel-R/viewtopic.php?t=245&highlight=graphical+approaches++support+analysis

Version originale aussi sur ResearchGate

Inférence et construction du modèle

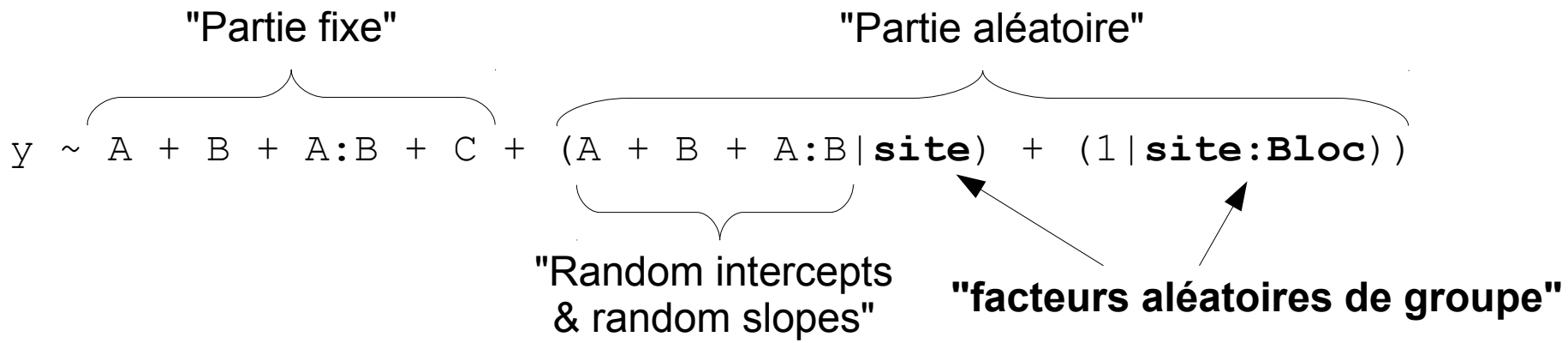
Construction du modèle



NB1 : la terminologie utilisée ici est sans doute tout sauf rigoureuse mais on trouve très peu d'explications précises sur la syntaxe de lme4, ce qu'on va tenter de faire ici tant bien que mal ...

NB2 : on a rarement des modèles aussi complexes (en particulier avec A:B comme "random interaction")

Inférence et construction du modèle

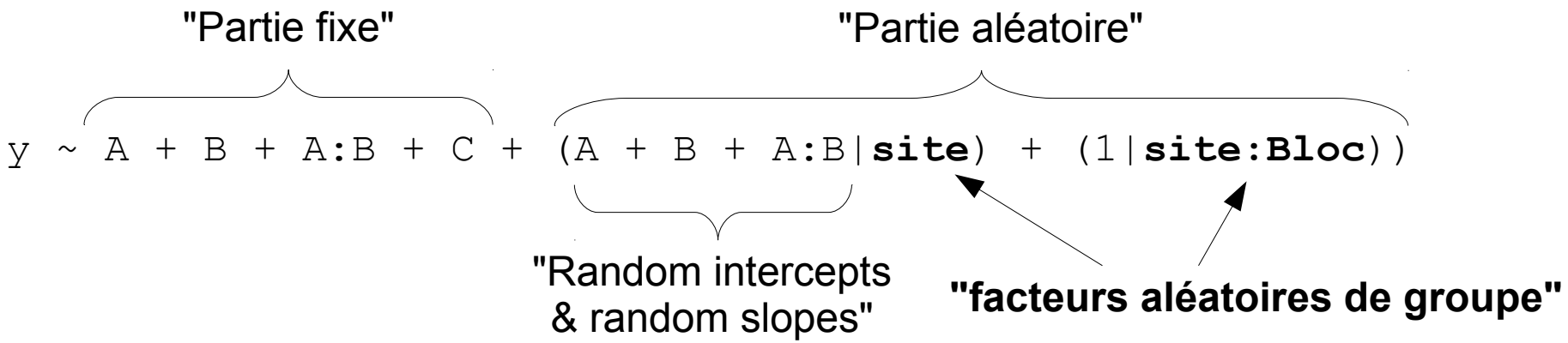


Facteurs aléatoires de groupe

Site et **bloc** sont les facteurs classiquement considérés comme aléatoires et dans ce cas Bloc est hiérarchisé à l'intérieur de Site (le bloc 1 du site 1 n'est pas présent dans les autres sites).

De manière générale il est **fortement recommandé** de donner un nom différent à chaque niveau hiérarchisé (pex bloc 1, 2, 3 dans le site 1 puis 4, 5, 6 dans le site 2) pour les distinguer des facteurs aléatoires croisés.

Inférence et construction du modèle

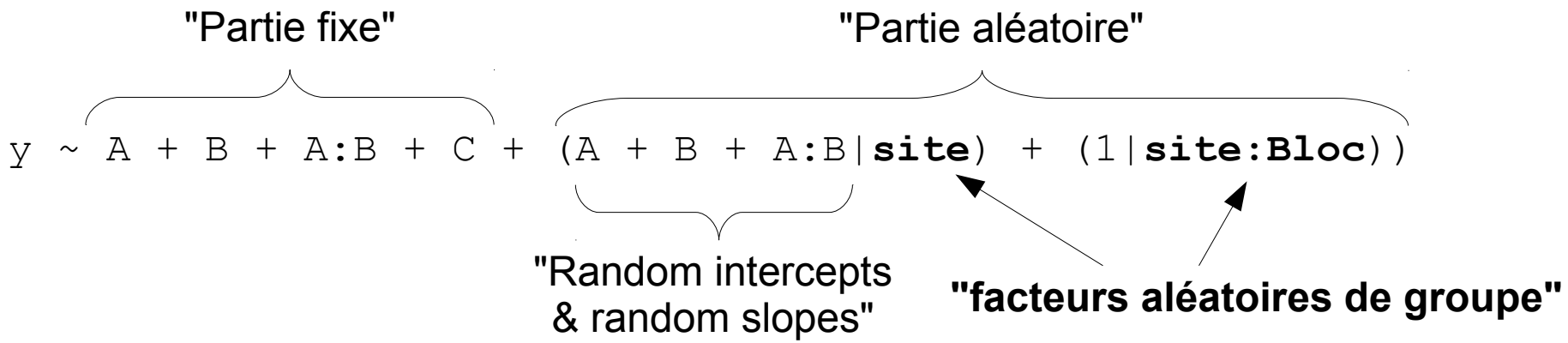


Facteurs aléatoires de groupe

On peut avoir des facteurs aléatoire de groupes croisés et éventuellement leur interaction :

```
(1|G1)  + (1|G2) + (1|G1:G2)
```

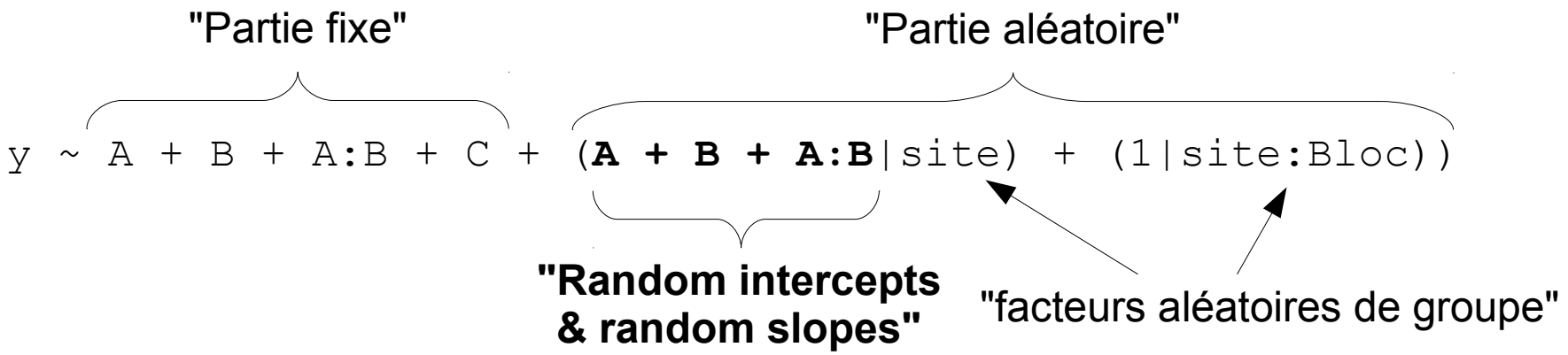
Inférence et construction du modèle



Facteurs aléatoires de groupe

L'ajoute de `(1|site)` et `(1|site:Bloc)` dans la formule induit une estimation des écarts (= random effects, fonction `ranef()`) de chaque site et chaque bloc par rapport à l'intercept càd la moyenne et surtout la variance de ces effets (variance components, fonction `VarCorr()`)

Inférence et construction du modèle



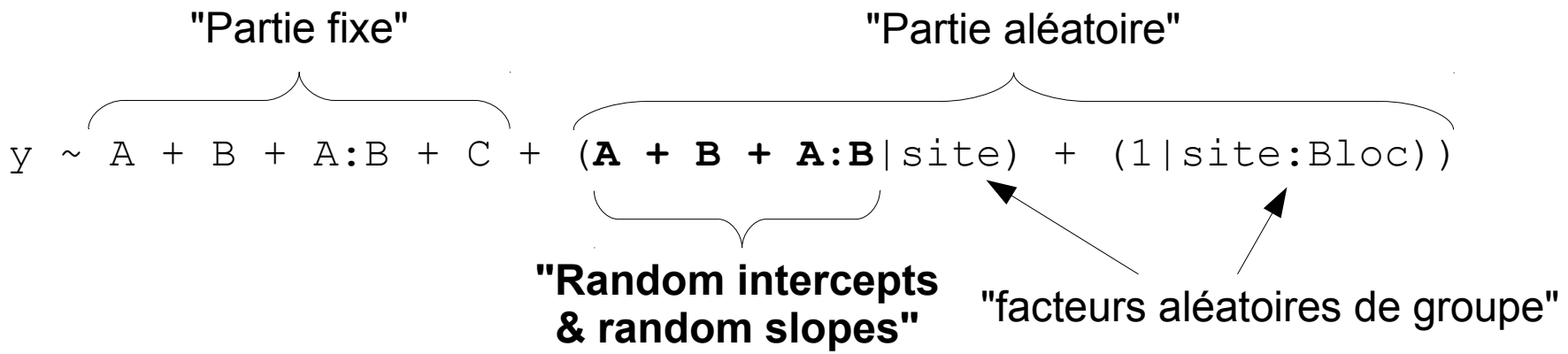
Random slopes

L'ajout de `A + B + A:B` dans `(A + B + A:B|site)` indique que l'on veut faire varier l'effet de A, B et leur interaction pour chaque site.

On peut le voir comme une interaction entre les effets fixes A, B et A:B et l'effet aléatoire site (cette interaction étant elle-même un effet aléatoire).

Le modèle va donc estimer la variance des "pentes" A, B et A:B

Inférence et construction du modèle

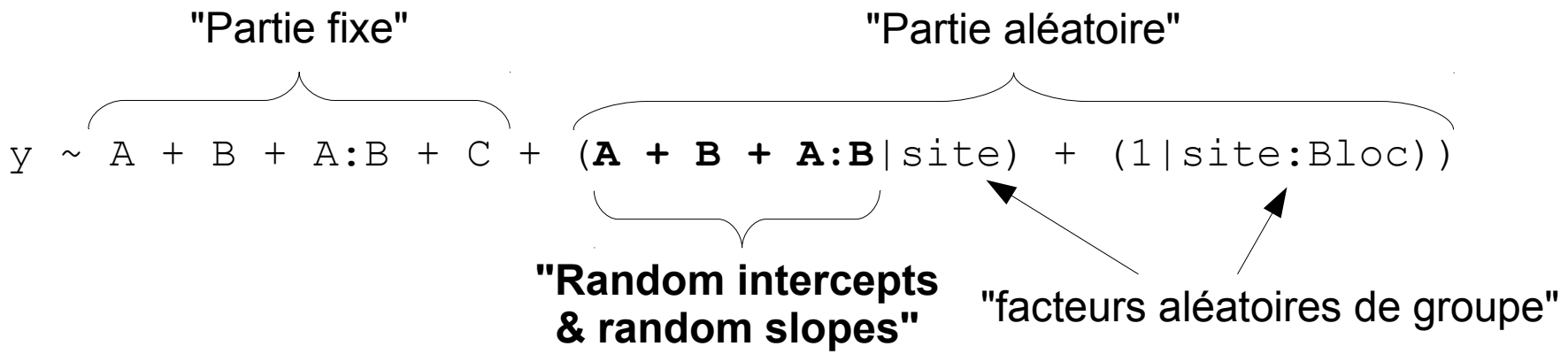


Random slopes

A et B sont souvent des variables continues mais ce n'est pas obligatoire. Il faut par contre que A et B soient des variables explicatives qui varient à l'intérieur des sites, au niveau des observations.
A ne peut par exemple pas être la surface du site.

NB : les modèles avec beaucoup de "random slopes" peuvent vite devenir très compliqués à estimer (et a interpréter). On se limite en général aux questions qui nous intéressent ou aux structures flagrantes dans les données

Inférence et construction du modèle

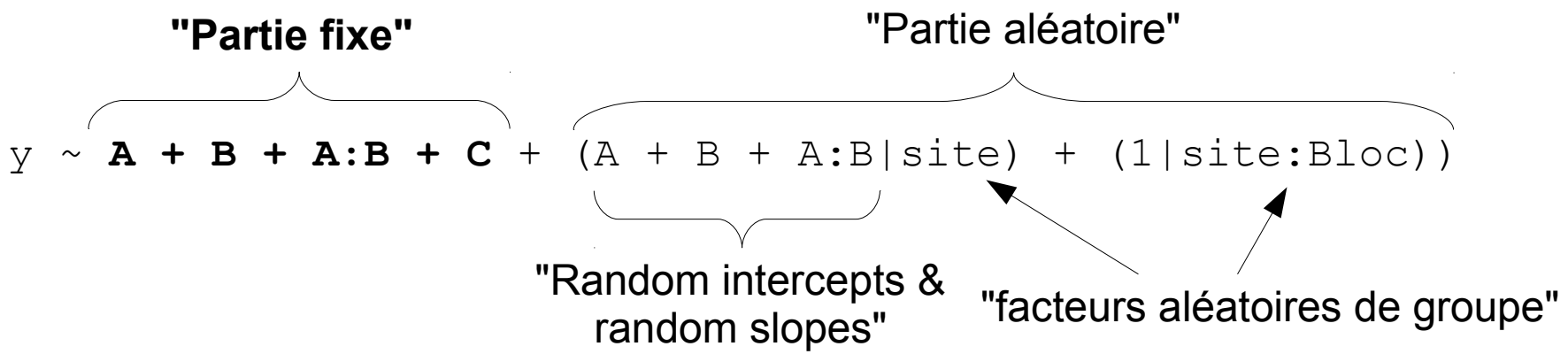


Random slopes

Le modèle estime par défaut les corrélations entre les random slopes et intercepts par groupe. Pour fixer les corrélation à 0 il faudrait écrire sans doute quelque chose comme :

```
(1 | site) + (-1 + A | site) + (-1 + B | site) + (-1 +
A:B | site)
```

Inférence et construction du modèle

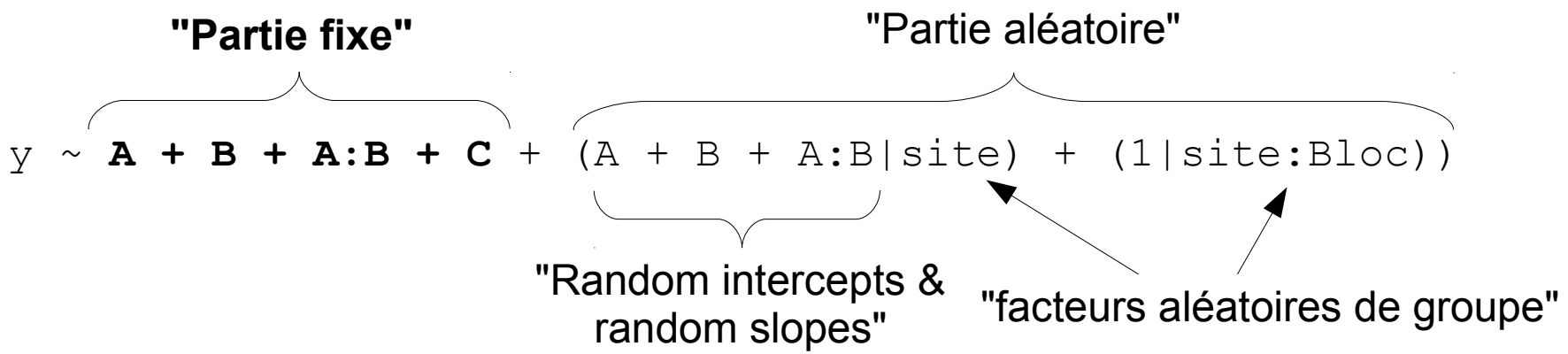


"Partie fixe"

A, B et A:B représentent ici la moyenne des valeurs estimées pour chaque site (moyenne d'une loi normale pour chaque paramètre).

C peut être de deux types :
1) une variable explicative au niveau des observations que l'on ne désire pas faire varier d'un site à l'autre. Par exemple on C peut être un indicateur du sexe d'individus mesurés et on estime que la différence mâle femelle sera du même ordre sur tous les sites (et tous les blocs).
2) une variable explicative au niveau du site que l'on ne peut donc pas faire varier au sein de chaque site. Par exemple C peut être la surface du site et modifiera l'intercept uniquement dans ce cas.

Inférence et construction du modèle



"Partie fixe"

On peut aussi avoir une interaction entre une variable explicative au niveau du site et au niveau des observations (qui varie pour chaque site) :

`A + C + A:C + (A|site)`

On a vu le cas où A était l'année d"observation et C le mode gestion du site. L'interaction A:C implique qu'on estime une pente moyenne différente pour les sites ayant des modes de gestion différents.

Inférence et construction du modèle

Construction de la partie aléatoire

En général on ne fait pas de tests d'hypothèse sur les composantes de la partie aléatoire.

Les variables aléatoire de groupe sont choisies en fonction de la manière dont les données ont été récoltées et du choix de traiter une variable qualitative comme fixe ou aléatoire.

Dans le pire des cas, la variabilité intergroupe est nulle et le modèle reviens à un modèle fixe classique mais même dans ce cas on recommande en général de ne pas retirer une variable qui décrit clairement la structure des données ("sacrificial pseudoreplication")

Inférence et construction du modèle

Construction de la partie aléatoire

Lorsque le nombre de groupe est faibles (pex <5) il est parfois difficile d'estimer correctement une variance qui peut alors tomber à 0 même si la variance réelle est >0.

Les avis divergent dans ce cas. Certains recommandent de traiter alors le groupe comme effet fixe, d'autres recommandent de le garder néanmoins comme facteur aléatoire de groupe

Inférence et construction du modèle

Construction de la partie aléatoire

Le choix d'ajouter une "random slope" ou pas (avec ou sans corrélation) peut dépendre :

des données (ea exploration graphique pour voir si la pente varie d'un groupe à l'autre, régressions séparées pour chaque groupe, ...)

de la quantité des données disponibles (plus on a de random slopes avec potentiellement des corrélations plus il faut des données de qualité pour les estimer)

de l'objectif de l'étude et de la facilité d'interprétation d'un modèle plus complexe

Très souvent les modèles mixtes sont de simples modèles "random intercept" avec un ou deux facteurs aléatoires de groupe :

```
A + B + C + D + E + (1|site)
```

Inférence et construction du modèle

Construction de la partie aléatoire

La fonction lmList (package lme4) facilite l'estimation de lm ou glm séparés pour des groupes de données

```
> (modcor <- lmList(y ~ year | site, data=dcor))
Call: lmList(formula = y ~ year | site, data = dcor)
Coefficients:
        (Intercept)        year
site_01   282.21278 -20.2107057
site_02    89.23243          NA
site_03  -385.46959  37.9555092
site_04  -113.89734  12.3283342
site_05    53.24114  -0.1731834
site_06   251.25488 -13.5102207
(…)
```

Dataset avec corrélation

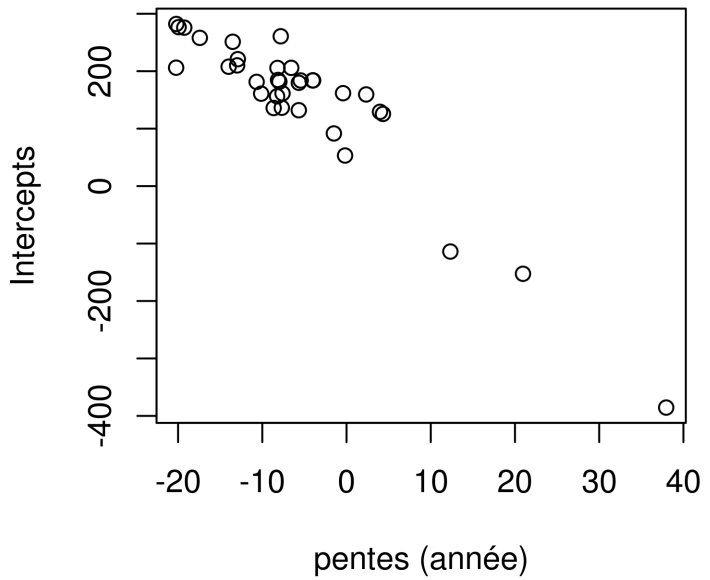
```
> (modpois <- lmList(y ~ year | site, data=d, family = poisson))
Call: lmList(formula = y ~ year | site, data = d, family = poisson)
Coefficients:
        (Intercept)         year
site_01   2.2293777 -0.189011220
site_02   3.2114767 -0.042158092
site_03   1.8087913 -0.086095827
site_04   4.7840756 -0.152423728
site_05   3.2666961 -0.188614485
(..)
```

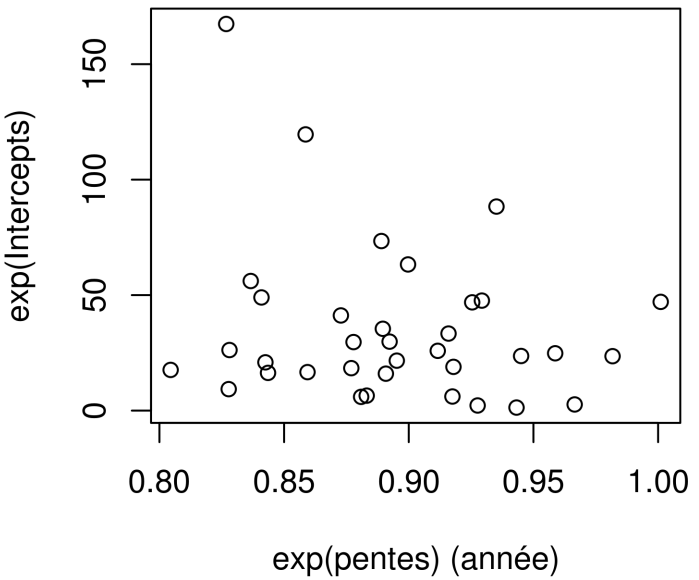
Dataset avec distribution de poisson mais sans corrélation entre les pentes et les intercept

Inférence et construction du modèle

Construction de la partie aléatoire

```
dev.new(8,4)
par(mfrow=c(1,2))
plot(coef(modcor)[,1] ~ coef(modcor)[,2],
     ylab = "Intercepts", xlab = "pentes (année)")
plot(exp(coef(modpois)[,1]) ~ exp(coef(modpois)[,2]),
     ylab = "exp(Intercepts)", xlab = "exp(pentes) (année)")
```





On visualise bien la corrélation négative entre les pentes et les intercepts.
On voit aussi qu'il y a des intercepts négatifs, preuve que le modèle n'est pas adapté et/ou que certaines estimations ne sont pas très bonnes

On voit ici qu'il y a une assez forte variabilité dans les pentes et les intercepts mais vraisemblablement pas de corrélation

Inférence sur les effets fixes

```
> lmm <- lmer(y ~ year*man + (1 + year|site), data = d2)
> summary(lmm)
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 site     (Intercept) 943.19   30.71
          year         27.56    5.25    0.19
 Residual             675.75   26.00
Fixed effects:
               Estimate Std. Error t value
(Intercept)     198.797      8.232  24.148
year             -2.029      1.242  -1.634
manmowing       -26.970     11.851  -2.276
year:manmowing   -9.467      1.755  -5.395
```

Pas de p-valeurs pour les modèles gaussiens

```
> glmm <- glmer(y ~ year + (1 + year|site), data = d, family = poisson)
> summary(glmm)
Random effects:
 Groups Name        Variance Std.Dev. Corr
 site   (Intercept) 0.91807  0.95816
        year        0.00208  0.04561  0.01
Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.13155    0.16770   18.67   <2e-16 ***
year        -0.11369    0.00966  -11.77   <2e-16 ***
```

Wald tests approximatifs pour les GLMM comme dans les GLM

Inférence

Inférence sur les effets fixes

Pourquoi pas de p-valeurs pour les modèles Gaussiens ?

Les auteurs de lme4 avancent 2 arguments :

1) dans de nombreux modèles, en particulier en cas de designs non balancés, on ne sait pas si la distribution des tests de wald est réellement une Student (ou une distribution de Fischer pour les comparaisons de modèles)

2) il est souvent difficile d'estimer les degrés de liberté corrects (combien a-t-on réellement de paramètres, combien de données réellement indépendantes dispose-t-on?)

Voir http://glmm.wikidot.com/faq pour plus de détails

Inférence

Inférence sur les effets fixes

Il existe de nombreuses possibilités pour l'inférence.

Test de Wald approximatif

L'approche pragmatique la plus simple et très utile en pratique est de considérer que les coefficients avec un t (ou un z pour les GLMM) >2 ou < -2 sont estimés correctement

(approximation normale, coef +- 2* se)

Inférence

Intervalles de confiance sur les paramètres par parametric bootstrap

Parametric Bootstrap = Fake data simulation :
On génère un nouveau jeu de données sur base des paramètres estimés par le modèle qu'on réanalyse avec un modèle identique. On recommence un grand nombre de fois --> on obtient la distribution des paramètres du modèle (et de toute autre valeur dérivée)

Méthode implémentée dans la fonction `confint()`

Intervalles de confiance sur les paramètres par parametric bootstrap

```
> (CI <- confint(lmm, method = "boot", nsim = 500))
Computing bootstrap confidence intervals ...
                                2.5 %      97.5 %
sd_(Intercept)|site         18.3520618  41.3101463
cor_year.(Intercept)|site  -0.2150056   0.8659323
sd_year|site                3.7786824   6.7256251
sigma                      23.2950131  28.8533389
(Intercept)               181.8152892 213.1364457
year                       -4.4023085   0.5055391
manmowing                 -50.4917731  -1.6039852
year:manmowing            -13.1080706  -6.1892560

> (CIglmm <- confint(glmm, method = "boot", nsim = 500,
                                        .progress="txt", PBargs=list(style=3)))
Computing bootstrap confidence intervals ...
  |=========================================                              |   57%
> CIglmm
                                2.5 %      97.5 %
sd_(Intercept)|site         0.69375367  1.16536570
cor_year.(Intercept)|site -0.47018112  0.51898676
sd_year|site                0.02634053  0.06283656
(Intercept)                 2.80876748  3.45586960
year                       -0.13288359 -0.09258169
```

Environs 30s de temps de calcul sur mon ordinateur

Paramètres permettant d'obtenir une barre de progression avec %

Temps de calcul beaucoup plus long ! ~5 minutes

Intervalles de confiance sur les paramètres par parametric bootstrap

Il peut être utile d'estimer le temps de calcul sur un petit nombre de simulations puis d'extrapoler

```
> system.time(CI <- confint(glmm, method = "boot", nsim = 10))
Computing bootstrap confidence intervals ...
   user  system elapsed
  7.013   0.156   7.238
```

7 secondes pour 10 simulations --> ~350 secondes pour 500 simulations

Inférence

Intervalles de confiance sur les paramètres par parametric bootstrap

`confint()` utilise la fonction `bootMer()` qui elle-même utilise `simulate()`. On peut utiliser ces fonctions directement pour des utilisations plus spécifiques

Fonction qui extrait tous les paramètres du modèle à passer comme argument à bootMer

```
> f <- function(m) {return(c(getME(m, "theta")*sigma(m), sigma = sigma(m), fixef(m)))}
> b <- bootMer(lmm, FUN = f, nsim = 200)
> b$t[1:5,]
     site.(Intercept) site.year.(Intercept) site.year    sigma (Intercept)      year
[1,]        0.7628365             0.18118504 0.1130103 29.31086    215.5557 -1.696027
[2,]        0.8103744             0.03499826 0.1556781 28.92839    201.3385 -0.378512
[3,]        0.8346103             0.06042532 0.1735899 25.91201    200.6700 -1.890946
[4,]        1.2545207             0.01279252 0.1878786 25.24632    200.0308 -2.814163
[5,]        1.1962980             0.03384101 0.1878064 27.18566    206.2372 -2.848562
     manmowing year:manmowing
[1,] -42.58491     -10.415632
[2,] -40.22099     -11.362464
[3,] -29.93819      -9.436540
[4,] -54.17401      -7.652619
[5,] -23.29053      -7.766572
```

b$t contient 200 lignes correspondant aux 200 simulations avec les valeurs de chaque paramètre extraites au moyen de la fonction f

Intervalles de confiance sur les paramètres par parametric bootstrap

On peut calculer les Intervales de Confiance comme le fait la fonction `confint` par défaut :

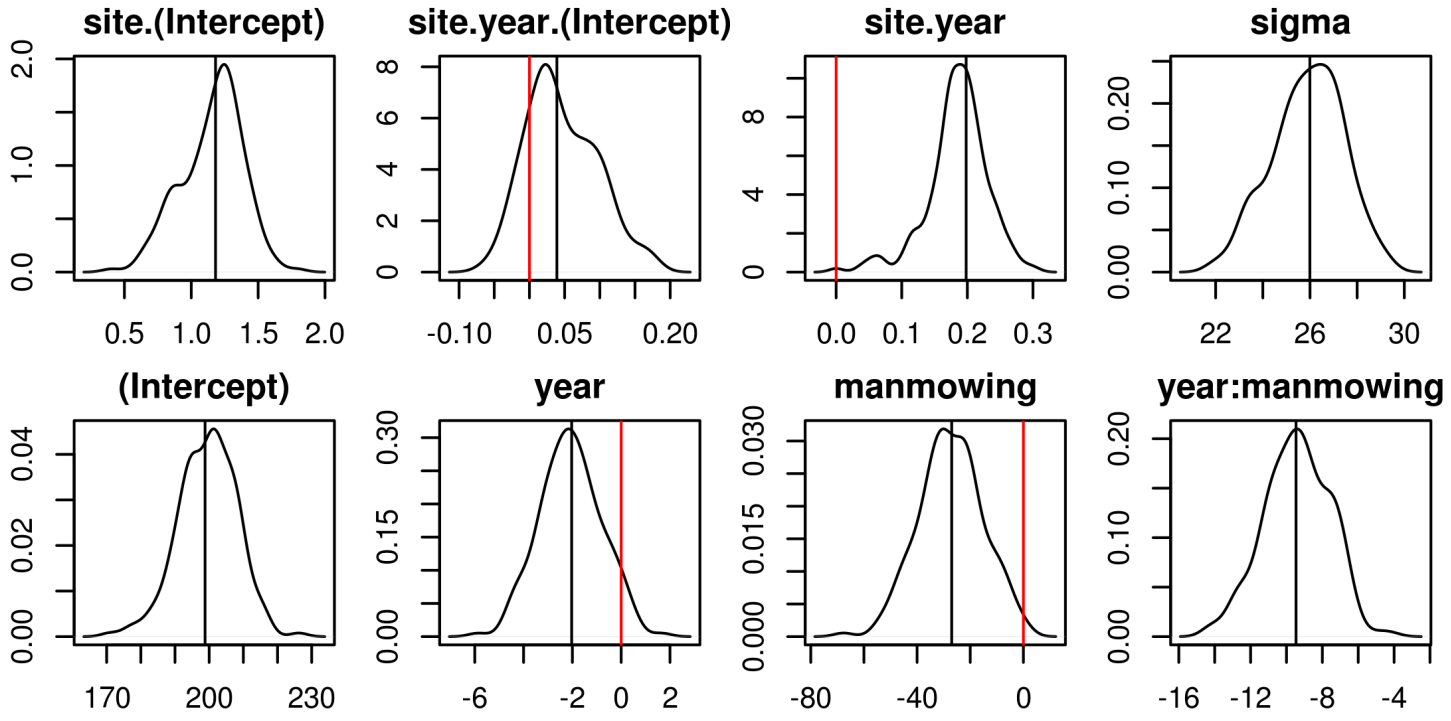
```
> (CI <- t(apply(b$t, 2, quantile, probs = c(0.025, 0.975), na.rm = TRUE)))
                              2.5%       97.5%
site.(Intercept)         17.148457  40.5042944
site.year.(Intercept)    -1.312645   3.8827935
site.year                 2.765144   6.5161568
sigma                    23.571833  28.8219678
(Intercept)             182.485979 213.4782949
year                     -4.276997   0.8320507
manmowing               -47.787670  -2.5228191
year:manmowing          -13.002529  -5.9137359
```

Inférence

Intervalles de confiance sur les paramètres par parametric bootstrap

Représentation graphique des distributions

```
dev.new(14/2.54, 7/2.54)
par(mfrow = c(2,4), mar = c(2,2,2,1))
for(i in 1:ncol(b$t)) {
    plot(density(b$t[,i]), main = colnames(b$t)[i],
         xlab = "", ylab = "")
    abline(v = 0, col = "red")
    abline(v = b$t0[i])
}
```



Inférence

Tests de rapport de vraisemblance pour comparer les modèles.

NB : on ne peut comparer la vraisemblance de modèles estimés avec la méthode REML lorsque leur partie fixe est différente.

La fonction anova réestime automatiquement le modèle avec l'option REML=FALSE. Attention cependant si vous devez faire le test vous-même.

```
> mod1 <- lmer(y ~ year*man + (1 + year|site), data = d2)
> mod2 <- lmer(y ~ year+man + (1 + year|site), data = d2)
> mod3 <- lmer(y ~ year + (1 + year|site), data = d2)
> mod4 <- lmer(y ~ man + (1 + year|site), data = d2)

> anova(mod2, mod1) # teste l'interaction
     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mod2  7 2560.5 2585.2 -1273.3   2546.5
mod1  8 2538.7 2566.9 -1261.3   2522.7 23.839      1  1.048e-06 ***

> anova(mod3, mod2) # teste man en absence d'interaction
     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mod3  6 2565.5 2586.6 -1276.8   2553.5
mod2  7 2560.5 2585.2 -1273.3   2546.5 6.9715      1   0.008282 **

> anova(mod4, mod2) # teste year en absence d'interaction
     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mod4  6 2587.2 2608.3 -1287.6   2575.2
mod2  7 2560.5 2585.2 -1273.3   2546.5 28.682      1  8.527e-08 ***
> logLik(mod4)
'log Lik.' -1281.537 (df=6)
> logLik(update(mod4, REML=FALSE))
'log Lik.' -1287.61 (df=6)
> logLik(refitML(mod4)) # idem mais plus rapide
```

Inférence

Tests de rapport de vraisemblance pour comparer les modèles.

Mêmes résultats avec `drop1`

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

          Df    AIC    LRT   Pr(Chi)
<none>       2538.7
year:man  1 2560.5 23.839 1.048e-06 ***
```

```
> drop1(update(lmm, ~.-year:man), test = "Chisq")

       Df    AIC     LRT   Pr(Chi)
<none>    2560.5
year    1 2587.2 28.6824 8.527e-08 ***
man     1 2565.5  6.9715  0.008282 **
```

Inférence

Tests de rapport de vraisemblance pour comparer les modèles.

Mêmes résultats avec ma fonction `Anova.lmer`
(qui se trouve dans mytoolbox.R)

```
> source("/home/gilles/stats/R/mytoolbox.R")
> Anova.lmer(lmm)
         LR Chisq df p(>Chisq)
year     28.68242  1    0.0000
man       6.97149  1    0.0083
year:man 23.83863  1    0.0000
```

Inférence

Tests de rapport de vraisemblance avec simulation

Les tests de rapports de vraisemblance sont également asymptotiques ie valides uniquement pour un grand nombre de données.
Les tests pour les effets fixes donnent des p-valeurs trop petites

On peut utiliser la fonction `simulate` pour générer des données sous l'hypothèse nulle c'est à dire sur base du modèle ne contenant pas la variable explicative que l'on veut tester.

Inférence

Tests de rapport de vraisemblance avec simulation

```
lrboot <- function(mod1,mod2) {
    ysim <- simulate(mod2)
    L1 <- logLik(refit(mod1,ysim))
    L2 <- logLik(refit(mod2,ysim))
    2*(L1-L2)
}

simulp <- function (mod1,mod2,nsimul=500) {
    obslr <- 2*(logLik(mod1)-logLik(mod2))[1]
    lrdist <- replicate(nsimul,lrboot(mod1,mod2))
    mean(lrdist >= obslr)
}
```

Attention à utiliser la ML au lieu de REML

```
> mod2 <- lmer(y ~ year+man + (1 + year|site), data = d2, REML = FALSE)
> mod3 <- lmer(y ~ year + (1 + year|site), data = d2, REML = FALSE)
> anova(mod3, mod2) # teste man en absence d'interaction
Data: d2
Models:
mod3: y ~ year + (1 + year | site)
mod2: y ~ year + man + (1 + year | site)
     Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
mod3  6 2565.5 2586.6 -1276.8   2553.5
mod2  7 2560.5 2585.2 -1273.3   2546.5 6.9715      1   0.008282 **

> simulp(mod1 = mod2, mod2 = mod3, nsimul = 500)
[1] 0.01
```

NB : le deuxième modèle doit obligatoirement être le modèle réduit

Inférence

Tests de rapport de vraisemblance avec simulation

On peut obtenir les p valeurs basées sur les simulations dans la fonction `Anova.lmer`

```
> system.time(res <- Anova.lmer(lmm, nsimul=500))
   user  system elapsed
280.126   0.016 280.652

> res
               LR df p(>Chisq) simul p
year     28.68242  1  8.53e-08   0.000
man       6.97149  1   0.00828   0.014
year:man 23.83863  1  1.05e-06   0.000
```

Près de 5 minutes pour 500 simulation

NB : les Tests de Rapport de vraisemblance ne testent pas les mêmes hypothèses que les tests de wald ou autres tests sur les paramètres

Inférence

Tests F avec correction de Kenward-Roger

Un des problèmes des tests de F est d'estimer les degrés de libertés du dénominateur.
La méthode de Kenward-Roger permet d'estimer ces degrés de liberté (pour des modèles Gaussiens uniquement).

```
> mod1 <- lmer(y ~ year*man + (1 + year|site), data = d2)
> mod2 <- lmer(y ~ year+man + (1 + year|site), data = d2)
> mod3 <- lmer(y ~ year + (1 + year|site), data = d2)
> mod4 <- lmer(y ~ man + (1 + year|site), data = d2)

> library(pbkrtest)
> KRmodcomp(mod1, mod2)
        stat    ndf    ddf F.scaling   p.value
Ftest 28.837  1.000 44.945         1 2.668e-06 ***

> KRmodcomp(mod2, mod3)
        stat    ndf    ddf F.scaling  p.value
Ftest 11.353  1.000 43.845         1 0.001579 **

> KRmodcomp(mod2, mod4)
        stat    ndf    ddf F.scaling   p.value
Ftest 37.703  1.000 47.051         1 1.644e-07 ***
```

Tests F avec correction de Kenward-Roger

Le package afex permet d'obtenir automatiquement un tableau similaire au tableau d'analyse de la variance classique

```
> library(afex)
> options(contrasts=c('contr.treatment', 'contr.poly'))
> modafex <- mixed(y ~ year*man + (1 + year|site), data = d2)
Contrasts set to contr.sum for the following variables: man, site
Fitting 5 (g)lmer() models:
[.....]
Obtaining 4 p-values:
[....]
Message d'avis :
In mixed(y ~ year * man + (1 + year | site), data = d2) :
  Numerical variables NOT centered on 0
(i.e., likely bogus results if in interactions): year
> modafex
       Effect     stat ndf     ddf F.scaling p.value
1 (Intercept) 963.6029   1 41.8370         1  0.0000
2        year  58.8535   1 44.9454         1  0.0000
3         man   5.1026   1 41.8370         1  0.0292
4    year:man  28.8370   1 44.9454         1  0.0000
```

NB : Tests de Type III par défaut
modifiable avec l'argument type

Inférence

La même fonction du package afex permet aussi de faire les tests de rapport de vraisemblance avec ou sans parametric bootstrap de manière similaire à Anova.lmer (avec la possibilité supplémentaire de faire des tests de type III)

```
> modafex <- mixed(y ~ year*man + (1 + year|site), data = d2, type = 2, method = "LRT")
> modafex
    Effect df.large df.small chisq df      p
1     year        7        6 28.68  1 0.0000
2      man        7        6  6.97  1 0.0083
3 year:man        8        7 23.84  1 0.0000

> lmm <- lmer(y ~ year*man + (1 + year|site), data = d2)
> Anova.lmer(lmm)
               LR df p(>Chisq)
year     28.68242  1  8.53e-08
man       6.97149  1   0.00828
year:man 23.83863  1  1.05e-06

> modafex <- mixed(y ~ year*man + (1 + year|site), data = d2, type = 2,
             method = "PB", args.test= list(nsim=500))
> modafex
    Effect    stat p.value
1     year 28.6824   0.002
2      man  6.9715   0.006
3 year:man 23.8386   0.002

> res <- Anova.lmer(lmm, nsimul = 500)
> res
               LR df p(>Chisq) simul p
year     28.68242  1  8.53e-08   0.000
man       6.97149  1   0.00828   0.016
year:man 23.83863  1  1.05e-06   0.000
```

NB : afex fait ici 499 simulations auxquelles elle ajoute la valeur observée

Inférence

Il est également possible d'obtenir des tests de Wald avec les degrés de liberté de Kenward Roger avec le package car

```
> library(car)
> Anova(lmm, test="F")
Note: method with signature ‘sparseMatrix#ANY’ chosen for function ‘kronecker’,
 target signature ‘dgCMatrix#ngCMatrix’.
 "ANY#sparseMatrix" would also be valid
Analysis of Deviance Table (Type II Wald F tests with Kenward-Roger df)

Response: y
              F Df Df.res    Pr(>F)
year     59.008  1 44.951 1.013e-09 ***
man      11.803  1 43.977  0.001302 **
year:man 28.837  1 44.945 2.668e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


> Anova(lmm, type=3, test="F")
Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger df)

Response: y
                    F Df Df.res    Pr(>F)
(Intercept) 575.1032  1 39.443 < 2.2e-16 ***
year          2.6437  1 43.512   0.11118
man           5.1026  1 41.837   0.02916 *
year:man     28.8370  1 44.945 2.668e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Inférence

Inférence sur les effets aléatoires

Dans certains cas on veut malgré tout avoir des inférences sur les effets aléatoires.
C'est assez compliqué en pratique car on veut tester si les variances sont différentes de 0 alors que leur valeur minimale est 0 ("testing on the boundary")

On peut obtenir des intervalles de confiance avec confint

On peut utiliser des tests de rapport de vraisemblance (avec REML = TRUE) dont les p valeurs seront trop élevées.
Attention on ne peut pas comparer la vraisemblance d'un modèle obtenu avec `lmer` et d'un autre avec `lm`
Les simulations vues précédemment pour les LRT donneront des p valeurs plus précises
Des tests par permutation peuvent dans certains cas être utiles si ils sont bien fait