CEF Formation – Introduction aux modèles mixtes

Marc J. Mazerolle
Centre d'étude de la forêt-UQAT

cef
Centre d'étude de la forêt

Université du Québec
en Abitibi-Témiscamingue

Design expérimental classique

Effet fixes vs aléatoires

Effet fixe: Lorque l'expérimentateur décide des traitements appliqués

Ex. effet de CPRS, Coupe partielle et contrôle sur la croissance d'arbres malades
On veut connaître l'effet de ces traitements en particulier sur la croissance.

Ex. effet de trois fournisseurs de graines sur le temps de germination
On veut connaître le fournisseur qui offre les graines germant le plus rapidement.

Les effets dans la plupart des expériences sont fixes.

Effet fixes vs aléatoires

Effet aléatoire: L'expérimentateur sélectionne aléatoirement les traitements qui seront étudiés, parmi tous ceux disponibles.

Lorsque l'expérimentateur s'intéresse à l'effet global du traitement sur la variable réponse, en considérant tous les traitements possibles.

Cas spécial: modèles hiérarchiques (effets imbriqués)

Effets aléatoires

Ex. effet de l'origine des graines sur le temps de germination

Ici, l'expérimentateur sélectionnerait aléatoirement un nombre de fournisseurs parmi tous les fournisseurs de graines (un catalogue).

Fournisseurs: A, R, V

On s'intéresse à la variabilité entre fournisseurs plutôt qu'à la différence entre les fournisseurs A, R et V.

Effet fixes vs aléatoires

Ex. la variabilité géographique de l'âge des peuplements affligés par la tordeuse du bourgeon de l'épinette depuis la dernière décennie au Québec

Ici, l'expérimentateur sélectionnerait aléatoirement des régions du Québec et déterminerait l'âge des peuplements affligés (sélectionnés aléatoirement).

Régions: Charlevoix, Gaspésie, Outaouais

Si l'expérimentateur choisit arbitrairement des régions (Abitibi, Lac St-Jean, Côte-Nord), l'effet est fixe.

Effet fixes vs aléatoires

La différence entre effets fixes et aléatoires?

Dans l'interprétation

Pour un effet fixe, la conclusion est pertinente aux traitements étudiés dans l'expérience (Abitibi, Lac St-Jean, Côte-Nord).

Pour effet aléatoire, la conclusion est pertinente à toutes les régions du Québec (pas seulement, Abitibi, Lac St-Jean, Côte-Nord)

Effet fixes vs aléatoires

ANOVA à 1 critère (1 facteur)

Peu importe si effet fixe ou aléatoire, le calcul demeure le même.

C'est l'interprétation qui change.

Effet fixes vs aléatoires

ANOVA à 2 critères avec réplication
(2 facteurs, A et B)

On a trois scénarios possibles:

1) A fixe, B fixe = Modèle I

2) A aléatoire, B aléatoire = Modèle II

3) A aléatoire, B fixe = Modèle III

Le type d'effet (fixe vs aléatoire) détermine le terme d'erreur à utiliser dans le tableau d'ANOVA.

Effets fixes vs aléatoires

Facteur	Modèle I (A fixe, B fixe)	Modèle II (A aléatoire, B aléatoire)	Modèle III (A aléatoire, B fixe)
A	$\frac{\text{A } MS}{\text{erreur } MS}$	$\frac{\text{A } MS}{\text{A x B } MS}$	$\frac{\text{A } MS}{\text{erreur } MS}$
B	$\frac{\text{B } MS}{\text{erreur } MS}$	$\frac{\text{B } MS}{\text{A x B } MS}$	$\frac{\text{B } MS}{\text{A x B } MS}$
A x B	$\frac{\text{A x B } MS}{\text{erreur } MS}$	$\frac{\text{A x B } MS}{\text{erreur } MS}$	$\frac{\text{A x B } MS}{\text{erreur } MS}$

Effet fixes vs aléatoires

Au-delà des designs classiques....
On peut utiliser des effets aléatoires pour analyser des données qui sont groupées ou nichées.

Les modèles mixtes englobent une variété de stratégies permettant de tenir compte correctement de la structure des données dans les analyses (données nichées, mesures répétées, structure de corrélation autoregressive, etc...).

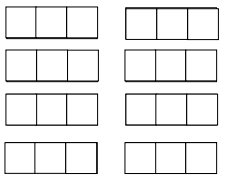
L'effet aléatoire est une variable aléatoire:

- On estime les paramètres qui décrivent la distribution de cette variable.
- L'effet aléatoire suit la distribution $N(0, \sigma^2)$.
- La moyenne de l'effet aléatoire = 0.

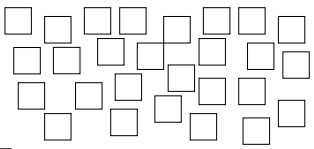
Un modèle mixte simple: blocs complets aléatoires

Blocs complets aléatoires

Rappel du design en blocs complets aléatoires:
unités expérimentales regroupées en blocs (1 unité/traitement dans chaque bloc)
e.g., vitesse de décomposition de bois mort placé dans des parcelles sans litière, avec 50% de litière et 100% litière.
un bloc = 1 site (le long de gradient latitudinal)



Pour design complètement aléatoire, on aurait besoin de 24 placettes de 24 sites différents (ANOVA standard)



Aucune litière 50% litière 100% litière

Blocs complets aléatoires

Le traitement est un facteur fixe (déterminé par l'expérimentateur).

Le bloc est un facteur aléatoire, et on suppose que les blocs sont différents entre eux.

- la variabilité intra-bloc est plus faible que la variabilité entre différents blocs.

Il n'y a pas de réplication à l'intérieur d'un même bloc (1 observation par traitement seulement):

- impossible de tester le terme d'interaction bloc x traitement.

L'ANOVA s'effectue comme une ANOVA à 2 facteurs sans réplication.

Blocs complets aléatoires

Ex.

On veut déterminer l'effet de 4 régimes d'exercices différents sur le gain de masse chez des cochons d'Inde dans des cages individuelles placées dans un grand entrepôt.

Étant donné qu'il n'y a qu'une source de chaleur à une extrémité de l'entrepôt, on pourrait s'attendre à avoir un gradient de température.

H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4$

H_a: il y a au moins une moyenne qui diffère des autres.

$\alpha = 0.05$

```
> pigs<-read.table("pigs.txt", header=TRUE)
```

Blocs complets aléatoires

```
> pigs
   Block Diet Weight
1      1    1    1.5
2      1    2    2.7
3      1    3    2.1
4      1    4    1.3
5      2    1    1.4
6      2    2    2.9
7      2    3    2.2
8      2    4    1.0
...
> pigs$Diet
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
> pigs$Block
 [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5
```

Il faut s'assurer que les variables soient reconnues comme facteurs.

```
> pigs$Diet<-as.factor(pigs$Diet)
> pigs$Block<-as.factor(pigs$Block)
```

Blocs complets aléatoires

```
> pigs$Diet
 [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
Levels: 1 2 3 4
> pigs$Block
 [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5
Levels: 1 2 3 4 5
```

Elles sont maintenant reconnues comme facteurs.

On peut ensuite effectuer l'ANOVA à blocs complets aléatoires comme on l'aurait fait pour ANOVA à 2 critères (sans interaction).

```
> block_aov<-aov(Weight~Diet+Block, data=pigs)
```

Blocs complets aléatoires

```
> summary(block_aov)
            Df Sum Sq Mean Sq F value    Pr(>F)
Diet         3 8.1535  2.7178 41.8665 1.241e-06 ***
Block        4 0.3930  0.0982  1.5135    0.2597
Residuals   12 0.7790  0.0649
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Effet de régime sur le gain de masse.

Le bloc explique une partie des données (pas beaucoup ici), mais réduit tout de même le terme d'erreur (*SSE*).

Comparons avec ANOVA à 1 critère naïve (ne tient pas compte du bloc)

```
> one_aov<-aov(Weight~Diet, data=pigs)
> summary(one_aov)
            Df Sum Sq Mean Sq F value    Pr(>F)
Diet         3 8.1535  2.7178  37.103 1.958e-07 ***
Residuals   16 1.1720  0.0732
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Blocs complets aléatoires

De façon équivalente, on peut construire un modèle de régression avec une variable catégorique pour bloc et une autre pour diète.

```
> block_lm<-lm(Weight~Diet+Block, data=pigs)
> summary(block_lm)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3650     0.1611   8.471 2.08e-06 ***
Diet2         1.4200     0.1611   8.812 1.38e-06 ***
Diet3         0.8600     0.1611   5.337 0.000177 ***
Diet4        -0.1400     0.1611  -0.869 0.401999
Block2       -0.0250     0.1802  -0.139 0.891938
Block3       -0.1500     0.1802  -0.833 0.421341
Block4       -0.0250     0.1802  -0.139 0.891938
Block5        0.2750     0.1802   1.526 0.152826
...

Residual standard error: 0.2548 on 12 degrees of freedom
Multiple R-Squared: 0.9165,     Adjusted R-squared: 0.8677
F-statistic: 18.81 on 7 and 12 DF,  p-value: 1.407e-05
```

Blocs complets aléatoires

Dans SAS:

```
proc glm data=pigs;
class Diet Block;
model Weight = Diet Block /solution;
run;
```

```
                                The GLM Procedure
Dependent Variable: Weight
                             Sum of
Source                 DF    Squares     Mean Square    F Value    Pr > F
Model                   7     8.5465      1.2209          18.81    <.0001
Error                  12     0.7790      0.0649

R-Square     Coeff Var      Root MSE    Weight Mean
0.916466      13.30483      0.254787       1.915000

Source         DF     Type III SS     Mean Square    F Value    Pr > F
Diet            3          8.1535      2.7178          41.87    <.0001
Block           4          0.3930      0.0982           1.51    0.2597
```

Notation des modèles mixtes

Modèle linéaire conventionnel (i.e., régression multiple)

$y_i = X_i\beta + \varepsilon_i$ où $X_i\beta = \beta_0 + \beta_{x_1}x_1 + ... + \beta_{x_k}x_k$

β: effets fixes

ε_i: erreurs $\varepsilon_i \sim N(0, \sigma^2)$

Modèle linéaire mixte

$y_i = X_i\beta + Z_ib_i + \varepsilon_i$ où $X_i\beta = \beta_0 + \beta_{x_1}x_1 + ... + \beta_{x_k}x_k$

β: effets fixes

$Z_ib_i = b_1x_1 + ... + b_kx_k$

b: effets aléatoires $b_i \sim N(0, \Psi)$ Ψ est une matrice variance-covariance pour effets aléatoires

ε_i: erreurs $\varepsilon_i \sim N(0, \sigma^2)$

Notation des modèles mixtes

Modèle linéaire mixte (blocs complets aléatoires, exemple de `pigs`)

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

$$y_i = \beta_0 + \beta_{Diet}Diet + b_{Block}Block + \varepsilon_i$$

Effet fixe: Diet (β_{Diet})

Effet aléatoire: Block (b_{Block})

Plus explicitement:

$$y_i = \beta_0 + \beta_{Diet2}Diet2 + \beta_{Diet3}Diet3 + \beta_{Diet4}Diet4 + \\ + b_{Block1}Block1 + b_{Block2}Block2 + b_{Block3}Block3 + b_{Block4}Block4 + b_{Block5}Block5 + \varepsilon_i$$

Fonction de likelihood

On peut écrire le likelihood d'un modèle mixte linéaire comme suit:

$$L(\mu,\sigma \mid y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y_i-\mu}{\sigma}\right)^2} \frac{n_i}{2} e^{\left\{\frac{(y_i - X_i\beta)^T V^{-1}(y_i - X_i\beta)}{-2\sigma^2}\right\}} |V|^{-\frac{1}{2}}$$

où

V = élément nécessaire au calcul de la matrice de variance-covariance Ψ (inclut l'information à propos des effets aléatoires)

β = effets fixes

θ = ensemble de paramètres qui caractérisent la précision relative (Δ) de la matrice de variance-covariance Ψ (Δ doit satisfaire à la condition $\Psi^{-1}\sigma^2 = \Delta^T\Delta$)

σ^2 = variance résiduelle

pour un modèle mixte avec un seul effet aléatoire b, la précision relative: $\Delta = \sqrt{\frac{\sigma^2}{\sigma_b^2}}$

Techniques d'estimation

On peut maximiser le likelihood de cette fonction et trouver les estimés des paramètres (Maximum Likelihood, *ML*)

- on obtient un log-likelihood
- estimés de variance sont sous-estimés avec *ML*
- propriétés normalité asymptotique tiennent pour de gros échantillon (petits échantillons plus problématiques)

Une alternative est d'utiliser le likelihood restreint (résiduel, *REML*)

- substitue la variable réponse originale (*y*) par les résidus dans le likelihood
- on obtient un log-likelihood restreint (restricted log-likelihood)
- estimés de *REML* ont de meilleures propriétés (option par défaut pour plusieurs logiciels `lme( )`, `Proc MIXED`)

R - Blocs comme effets aléatoires

Faisons rouler un modèle mixte pour analyse en blocs avec `pigs`

```
> library(nlme)
```
package contenant fonction `lme( )`

```
> block_lme2<-lme(Weight~Diet, random=~1 | Block, data=pigs, method="REML")
```

- syntaxe similaire à celle de `glm( )`, `lm( )`
- `random = ~1 | Block` indique qu'on a un effet aléatoire pour les blocs (`~1` correspond à 1 intercepte différent par bloc)
- `method="REML"` (par défaut) indique méthode d'estimation (aussi `"ML"`)

R - Blocs comme effets aléatoires

```
> summary(block_lme2)
Linear mixed-effects model fit by REML
 Data: pigs
       AIC      BIC    logLik
  21.74701 26.38255 -4.873507

Random effects:
 Formula: ~1 | Block
        (Intercept)  Residual
StdDev:   0.091287 0.2547875
...
```

identifie la stratégie d'estimation *REML*

effets aléatoires

σ^2_{Block} = variabilité entre blocs = 0.091287^2 = 0.0083

σ^2 = variabilité intra-bloc (résiduelle) = 0.2547875^2 = 0.0649

R - Blocs comme effets aléatoires

```
...
Fixed effects: Weight ~ Diet
            Value Std.Error DF   t-value p-value
(Intercept)  1.38 0.1210372 12 11.401455  0.0000
Diet2        1.42 0.1611418 12  8.812116  0.0000
Diet3        0.86 0.1611418 12  5.336915  0.0002
Diet4       -0.14 0.1611418 12 -0.868800  0.4020

Correlation:
      (Intr) Diet2  Diet3
Diet2 -0.666
Diet3 -0.666  0.500
Diet4 -0.666  0.500  0.500

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.5276766 -0.4117586  0.1936120  0.5367776  1.6162086

Number of Observations: 20
Number of Groups: 5
```

effets fixes

De grandes valeurs ici (i.e., près de |1|) peuvent indiquer des problèmes avec le modèle.

(fortes corrélations entre variables explicatives à éviter)

SAS - Blocs comme effets aléatoires

`Proc GLM` N'EST PAS approprié pour les modèles mixtes

- l'output conventionnel par défaut (e.g., Type III SS) est incorrect car n'utilise pas les bons termes d'erreurs (il faut les spécifier explicitement avec une ligne Test ou LSmeans et E=)
  - pour des designs complexes, `Proc GLM` ne permet pas de construire des termes d'erreurs complexes
- l'inclusion d'un terme dans la ligne `random` ne change rien à l'estimation (le terme est estimé comme s'il était fixe)
  - la structure de covariance n'est pas estimée explicitement, et par conséquent, les *SE's* de certains tests sont incorrects (Estimate, comparaisons multiples)
- split-plots, mesures répétées et autres modèles mixtes devraient être analysés avec `Proc MIXED`.

SAS - Blocs comme effets aléatoires

Faisons rouler un modèle mixte pour analyse en blocs avec `pigs`

`Proc MIXED` procédure permettant d'ajuster des modèles mixtes

```
proc mixed data=pigs method=REML;
class Diet Block;
model Weight = Diet/solution ddfm=contain;
random Block;
run;
```

`method=REML` (par défaut) plusieurs autres options disponibles dont `ML`

`solution` demande les estimés des effets fixes

`random Block` spécifie que bloc est un effet aléatoire

`ddfm=contain` (par défaut) spécifie la stratégie pour déterminer les *df* associés aux effets fixes

SAS - Blocs comme effets aléatoires

Calcul des degrés de liberté du dénominateur pour les effets fixes

plusieurs options possibles et sujet de controverse en stats

- `ddfm=contain`
- `ddfm=Kenwardroger`
- `ddfm=satterthwaite`
- `ddfm=residual`
- `ddfm=betwithin`

} différents types de modèles utilisent différentes valeurs par défaut de `ddfm` dépendamment de la présence de la ligne `random` ou `repeated`

la méthode de Kenward-Roger est souvent suggérée par SAS, particulièrement pour mesures répétées

différentes approches peuvent mener à différentes conclusions (surtout avec structures de covariance complexes)

pendant un certain temps, `R` ne donnait plus de *df* et de *P*.

SAS - Blocs comme effets aléatoires

Output effets aléatoires

```
Covariance Parameter Estimates
       Cov Parm     Estimate
       Block        0.008333
       Residual      0.06492
```

σ^2_{Block} = variabilité entre blocs = 0.0083

σ^2 = variabilité intra bloc = 0.0649

ATTENTION:
si on met l'argument `covtest` sur la ligne `Proc MIXED`, SAS ajoute une colonne avec des tests d'hypothèses sur les effets aléatoires (H_o: $\sigma^2 = 0$)

```
Covariance Parameter Estimates
                          Standard       Z
Cov Parm     Estimate        Error  Value    Pr Z
Block        0.008333     0.01859   0.45  0.3270
Residual      0.06492     0.02650   2.45  0.0072
```

tests non valides à moins d'avoir un très grand nombre de niveaux pour chaque effet aléatoire

SAS - Blocs comme effets aléatoires

```
            Fit Statistics
-2 Res Log Likelihood             9.7
AIC (smaller is better)          13.7
AICC (smaller is better)         14.7
BIC (smaller is better)          13.0
```

SAS enlève une constante du log-likelihood restreint

Ceci influence les valeurs d'*AIC* et de ses homologues.

Toutefois, l'écart d'*AIC* entre deux modèles demeurera le même qu'en utilisant le log-likelihood restreint qui inclut la constante (e.g., provenant de R).

SAS - Blocs comme effets aléatoires

effets fixes (certaines versions les identifient comme effets aléatoires: c'est une ERREUR)

```
...
            Solution for Fixed Effects
                        Standard
Effect   Diet  Estimate     Error      DF    t Value    Pr > |t|
Intercept      1.2400      0.1210       4      10.24      0.0005
Diet     1     0.1400      0.1611      12       0.87      0.4020
Diet     2     1.5600      0.1611      12       9.68      <.0001
Diet     3     1.0000      0.1611      12       6.21      <.0001
Diet     4          0           .       .          .           .
...
              Type 3 Tests of Fixed Effects
                                 Num     Den
Effect         DF      DF    F Value    Pr > F
Diet            3      12      41.87    <.0001
```

Suppositions du modèle

Avant de se lancer dans l'interprétation, il faut vérifier si le modèle s'ajuste bien aux données et si les suppositions sont respectées.

Suppositions:

1) Normalité des erreurs.

2) Normalité des effets aléatoires.

3) Homogénéité des variances.

4) Indépendance des observations entre groupes différents et entre niveaux différents (effet site, parcelle, arbre).

Vérification des suppositions

Outils potentiels

- graphique de résidus vs valeurs prédites (homoscédasticité)
- graphique quantile-quantile (normalité des erreurs, effets aléatoires)

La vérification des suppositions est plus complexe car le modèle mixte est plus complexe

- aucun test d'ajustement formel
- plusieurs des diagnostiques ne sont pas disponibles ou difficile à calculer
- différents types de résidus

Deux niveaux de valeurs prédites

Basées uniquement sur effets fixes:

valeurs prédites basées uniquement sur les effets fixes (`fixef(block_lme)`)

```
Fixed effects: Weight ~ Diet
            Value Std.Error DF   t-value p-value
(Intercept)  1.38 0.1210372 12 11.401455  0.0000
Diet2        1.42 0.1611418 12  8.812116  0.0000
Diet3        0.86 0.1611418 12  5.336915  0.0002
Diet4       -0.14 0.1611418 12 -0.868800  0.4020
```

$\hat{y}_i = \beta_0 + \beta_{Diet2} Diet2 + \beta_{Diet3} Diet3 + \beta_{Diet4} Diet4$

$\hat{y}_1 = 1.38$

$\hat{y}_2 = 1.38 + 1.42 = 2.80$

```
> pigs[1:10,]
   Block Diet Weight
1      1    1    1.5
2      1    2    2.7
3      1    3    2.1
4      1    4    1.3
5      2    1    1.4
6      2    2    2.9
7      2    3    2.2
8      2    4    1.0
9      3    1    1.4
10     3    2    2.1
```

```
> fitted(block_lme, level=0)
   1    1    1    1    2    2    2    2    3    3    3    3    4    4    4    4
1.38 2.80 2.24 1.24 1.38 2.80 2.24 1.24 1.38 2.80 2.24 1.24 1.38 2.80 2.24 1.24
   5    5    5    5
1.38 2.80 2.24 1.24
```

Deux niveaux de valeurs prédites

Prédiction avec effets aléatoires (BLUP, best linear unbiased predictors):

somme de valeurs prédites selon effets fixes + effets aléatoires

effets aléatoires (BLUP)

```
> ranef(block_lme)
   (Intercept)
1 -0.005089051
2 -0.013570803
3 -0.055979562
4 -0.013570803
5  0.088210219
```

on a un intercepte différent pour chaque bloc (`random = ~1 | Block`)

```
> pigs[1:5,]
  Block Diet Weight
1     1    1    1.5
2     1    2    2.7
3     1    3    2.1
4     1    4    1.3
5     2    1    1.4
```

$\hat{y}_{BLUP1} = \beta_0 + \beta_{Diet2} Diet2 + \beta_{Diet3} Diet3 + \beta_{Diet4} Diet4 + b_i$

$\hat{y}_{BLUP1} = 1.38 + -0.00509 = 1.3749$

$\hat{y}_{BLUP2} = 1.38 + 1.42 + -0.00509 = 2.7949$

```
> fitted(block_lme, level=1)[1:8]
       1        1        1        1        2        2        2        2
1.374911 2.794911 2.234911 1.234911 1.366429 2.786429 2.226429 1.226429
```

Coéfficients corrigés pour effets aléatoires

On peut extraire les coéfficients qui incluent les effets aléatoires (BLUP) directement avec la fonction `coef( )`

```
> coef(block_lme, level=1)
  (Intercept) Diet2 Diet3 Diet4
1    1.374911  1.42  0.86 -0.14
2    1.366429  1.42  0.86 -0.14
3    1.324020  1.42  0.86 -0.14
4    1.366429  1.42  0.86 -0.14
5    1.468210  1.42  0.86 -0.14
```

on a seulement l'intercepte qui change (effet aléatoire sur intercepte)

les autres coéfficients demeurent les mêmes

Deux niveaux de résidus

Excluant effets aléatoires (résidus marginaux = marginal residuals):

mesurent la déviation par rapport à la moyenne de la population

```
> residuals(block_lme, level=0)[1:10]
    1     1     1     1     2     2     2     2     3     3
 0.12 -0.10 -0.14  0.06  0.02  0.10 -0.04 -0.24  0.02 -0.70
```

`residuals( )` a un argument `type =` (par défaut, `type="pearson"`)

(aussi `"response"` ou `"normalized"`)

Incluant effets aléatoires (résidus conditionnels = conditional residuals):

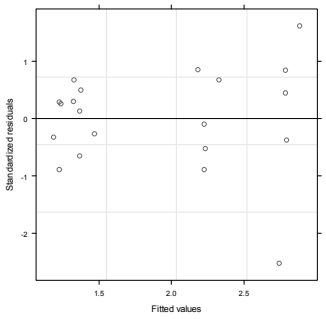
mesurent la déviation par rapport à la moyenne conditionnelle (condtionnelle aux groupes ou effets aléatoires)

```
> residuals(block_lme, level=1)[1:10]
          1           1           1           1           2           2
 0.12508905 -0.09491095 -0.13491095  0.06508905  0.03357080  0.11357080
          2           2           3           3
-0.02642920 -0.22642920  0.07597956 -0.64402044
```

Vérification des suppositions

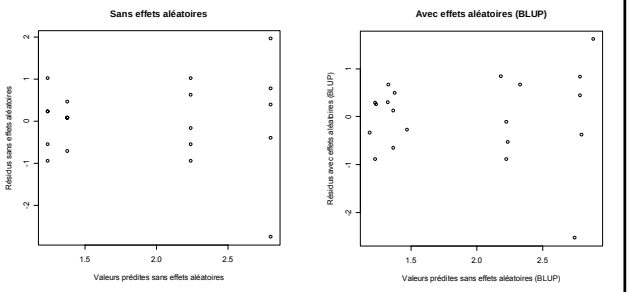
```
> plot(block_lme)
```

par défaut, montre le graphique avec le plus haut niveau d'effets aléatoires



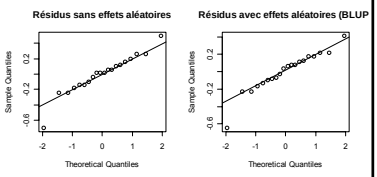
Vérification des suppositions

On peut créer nous-mêmes les graphiques en spécifiant les niveaux qui nous intéressent.

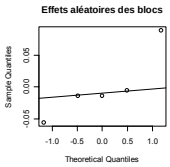


Vérification des suppositions

On peut vérifier la normalité des résidus et effets aléatoires avec `qqnorm( )`.



De façon générale, il faut un minimum de groupes pour pouvoir évaluer la normalité des effets aléatoires (5 – pas suffisant).



Vérification des suppositions

Du côté de SAS:

`Proc MIXED` peut extraire les résidus de Pearson marginaux ou conditionnels.

`Proc GLIMMIX` a une batterie de graphiques très pratiques pour vérifier le respect des suppositions.

procédure expérimentale qui n'est pas intégrée dans SAS `9.1` (maintenant dans SAS `9.2`) et qui permet d'ajuster des modèles mixtes généralisés.

il faut la télécharger sur le site de SAS et l'installer nous-même.

Vérification des suppositions

Du côté de SAS:

```
ods html;
ods graphics on;
```
on utilise ODS pour générer des graphiques en html (ou pdf)

```
proc glimmix data=pigs
      plots=(studentpanel(type=noblup)
             studentpanel(type=blup));
      class Diet Block;
      model Weight = Diet /ddfm=contain;
      random Block;
      run;
```
on demande des graphiques avec résidus de Student (avec BLUP's et sans BLUP's)

```
ods graphics on;
ods html close;
```

Vérification des suppositions

Du côté de SAS `Proc GLIMMIX`:

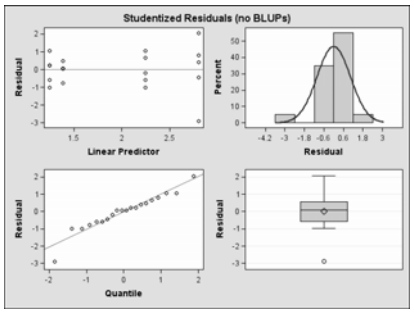
Covariance Parameter Estimates		
Cov Parm	**Estimate**	**Standard Error**
Block	0.008333	0.01859
Residual	0.06492	0.02650

estimés identiques à `Proc Mixed`

Type III Tests of Fixed Effects				
Effect	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Diet	3	12	41.87	<.0001

Vérification des suppositions

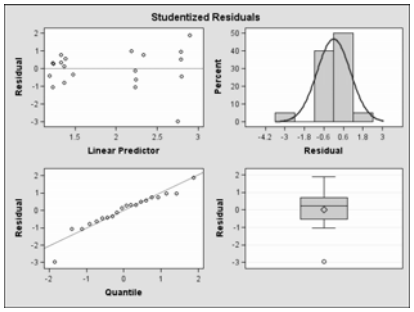
Du côté de SAS `Proc GLIMMIX`:



résidus de Student sans effets aléatoires

Vérification des suppositions

Du côté de SAS `Proc GLIMMIX`:



résidus de Student avec effets aléatoires

Blocs complets aléatoires - pigs

Le modèle semble bien s'ajuster aux données et les suppositions principales sont respectées.

On peut maintenant aller interpréter les résultats (effets fixes et effets aléatoires).

Tester des Hypothèses – effets fixes

Sur effets fixes

on peut interpréter les β's, *SE*'s et tests-*t* associés aux effets fixes (comme pour les GLM's) à partir de l'output de `summary( )`

```
...
Fixed effects: Weight ~ Diet
            Value Std.Error DF   t-value p-value
(Intercept)  1.38 0.1210372 12 11.401455  0.0000
Diet2        1.42 0.1611418 12  8.812116  0.0000
Diet3        0.86 0.1611418 12  5.336915  0.0002
Diet4       -0.14 0.1611418 12 -0.868800  0.4020
...
```

Masse est plus grande avec diète 2 que 1.

Masse est plus grande avec diète 3 que 1.

Masse n'est pas plus faible avec diète 4 que 1.

Tester des Hypothèses – effets fixes

on peut effectuer des tests-*F* séquentiels (même ordre que formule) ou marginaux (modèle complet vs modèle sans variable) avec la fonction `anova( )`

pratique pour évaluer effet global de variables catégoriques

```
> anova(block_lme, type="sequential")
             numDF denDF  F-value p-value
(Intercept)     1    12 746.5093  <.0001
Diet            3    12  41.8665  <.0001
```

Important effet global de diète sur la masse.

Tester des Hypothèses – effets fixes

les tests du ratio du likelihood (likelihood ratio statistic) sur les effets fixes ne performent pas bien avec les modèles mixtes

approximent grossièrement une distribution du X^2 (donnent valeurs de *P* plus faibles que ce qu'on devrait avoir – test anti-conservateur)

si on veut tout de même utiliser ce test, il faut choisir la méthode d'estimation par *ML* (*REML* a un terme qui change avec les effets fixes).

on garde les mêmes effets aléatoires pour les deux modèles comparés

```
> anova(block_lme2, block_lme3)
           Model df      AIC      BIC     logLik   Test  L.Ratio p-value
block_lme2     1  6 11.67371 17.64810   0.163147
block_lme3     2  3 47.49795 50.48514 -20.748974 1 vs 2 41.82424  <.0001
```

on pourrait obtenir une meilleure estimation du *P* par simulations ou bootstrap paramétrique (présenté dans mon cours d'hiver)

Tester des Hypothèses – effets fixes

Du côté de SAS:

```
 Type 3 Tests of Fixed Effects
              Num     Den
Effect         DF      DF    F Value    Pr > F
Diet            3      12      41.87    <.0001
```

ATTENTION:
en présence d'interaction(s) dans le modèle, ce test est seulement valide pour l'interaction (pour les autres termes le principe de marginalité n'est pas respecté).

l'argument `HTYPE=` de la ligne `model`, permet de spécifier le type de test d'hypothèse (1, 2, 3, par défaut `HTYPE=3`).

En présence d'interactions, préférable d'utiliser `HTYPE=1`.

Tester des Hypothèses – effets fixes

Du côté de SAS:

si on a ajouté l'argument `/solution` dans la ligne `model`, on obtient:

Solution for Fixed Effects

Effect	Diet	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		1.2400	0.1210	4	10.24	0.0005
Diet	1	0.1400	0.1611	12	0.87	0.4020
Diet	2	1.5600	0.1611	12	9.68	<.0001
Diet	3	1.0000	0.1611	12	6.21	<.0001
Diet	4	0	.	.	.	.

on peut interpréter directement par rapport au niveau de référence ou on peut utiliser `estimate` ou `contrast` comme avec `Proc GLM`

Tester des Hypothèses – effets aléatoires

Sur effets aléatoires

on utilise généralement un test du ratio du likelihood pour déterminer si l'inclusion d'un effet aléatoire est importante ou non (pour mêmes effets fixes)

la statistique approxime un X^2 mais le test est conservateur (donne *P* plus grand qu'il le devrait)

pour tester les effets aléatoires, on peut utiliser soit le *ML* ou *REML* (être constant – tous par *ML*, ou tous par *REML*)

on peut comparer un modèle sans effets aléatoires vs celui avec blocs

```
> block_lm1<-lm(Weight~Diet, data=pigs)
> block_lme2<-lme(Weight~Diet, random=~1 | Block, data=pigs, method="ML")
> chi<- -2*(logLik(block_lm1)[1]- logLik(block_lme2)[1])
> chi
[1] 0.3434228
> 1-pchisq(chi, df=1)
[1] 0.5578602
```

on pourrait déterminer le *P* par simulation

Tester des Hypothèses – effets aléatoires

Du côté de SAS:

si on a ajouté l'argument `/solution` dans la ligne `random`, on obtient:

Solution for Random Effects

Effect	Block	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Block	1	-0.00509	0.07792	12	-0.07	0.9490
Block	2	-0.01357	0.07792	12	-0.17	0.8646
Block	3	-0.05598	0.07792	12	-0.72	0.4862
Block	4	-0.01357	0.07792	12	-0.17	0.8646
Block	5	0.08821	0.07792	12	1.13	0.2797

les mêmes valeurs obtenues avec `ranef( )` – ce sont les différences entre l'intercepte moyen et celui de chaque bloc

ATTENTION:
le test présenté dans SAS pour les effets aléatoires est basé sur une approximation normale et n'est valide que lorsqu'il y a un grand nombre d'observations par niveaux d'effets aléatoires (e.g., > 30).

Sélection de modèles et inférence multimodèles

On peut utiliser l'AIC_c pour effectuer la sélection de modèles pour les effets fixes et/ou aléatoires si on choisit l'estimation par *ML*.

Avec l'estimation par *REML*, on peut utiliser l'AIC_c uniquement pour sélectionner les effets aléatoires (même raison que pour test LR).

Un autre problème, c'est l'estimation de *n*: nombre total d'observations?
nombre total de groupes indépendants?
combinaison des deux?

Mieux de calculer *AIC* soi-même (SAS ne donne pas toujours la bonne mesure, utilise *n* total ou *n – p*, dépendamment de la version ou de la procédure utilisée).

Un nouvel *AIC* a été développé pour les modèles mixtes utilisant le *REML* (Vaida et Blanchard 2005: Biometrika 92:351-370)) – code R disponible.

Un autre outil: la corrélation intraclasse

Corrélation intraclasse (intraclass correlation):

une estimation de la corrélation entre observations d'un niveau donné d'un effet aléatoire

$$\rho = \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_\varepsilon^2}$$

e.g., corrélation intraclasse pour effet aléatoire de bloc

$$\rho = \frac{(0.091287)^2}{(0.091287)^2 + (0.2547875)^2} = 0.1138$$

peut aussi être interprété comme le % de variation totale expliqué par σ^2_α.

Modèles hiérarchiques

Un type de modèle mixte qui consiste à ajouter des effets aléatoires pour tenir compte de la hiérarchie des données (effets nichés)

aussi connus comme « multilevel models », « hierarchical models », « random coefficient models »

e.g., mesures répétées sur plusieurs arbres d'une même parcelle dans différents sites

`lme( )` a été conçu spécifiquement pour ce genre de modèles

modèle mixte « classique »

```
proc mixed data=pigs method=REML;
class Diet Block;
model Weight = Diet/solution;
random Block;
run;
```

modèle hiérarchique

```
proc mixed data=pigs method=REML;
class Diet Block;
model Weight = Diet/solution;
random int / subject=Block;
run;
```

Exemple de croissance d'orangers

Ex. Croissance de 5 orangers (circonférence en mm) en fonction du nombre de jours écoulés.

```
> Orange
   Tree  age circumference
1     1  118            30
2     1  484            58
3     1  664            87
4     1 1004           115
5     1 1231           120
6     1 1372           142
7     1 1582           145
8     2  118            33
9     2  484            69
10    2  664           111
...
```

chaque arbre a été mesuré plusieurs fois

Exemple de croissance d'orangers

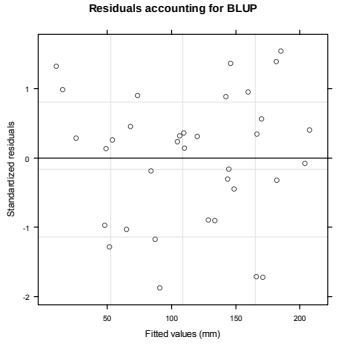
On peut essayer un modèle mixte avec Tree comme effet aléatoire:

```
> mod_orange<-lme(circumference~age, random=~1 | Tree, data=Orange)
```

Vérifions les suppositions du modèle avant d'aller interpréter les résultats.

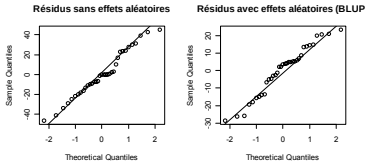
Vérification des suppositions

Bonne répartition des résidus

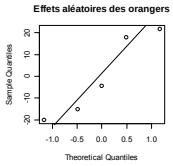


Vérification des suppositions

Normalité des résidus



Normalité des effets aléatoires?
(5 arbres seulement)



Exemple de croissance d'orangers

On peut essayer un modèle mixte avec Tree comme effet aléatoire:

```
> mod_orange<-lme(circumference~age, random=~1 | Tree, data=Orange)

> summary(mod_orange)
Linear mixed-effects model fit by REML
 Data: Orange
       AIC      BIC    logLik
  311.1669 317.1529 -151.5834

Random effects:
 Formula: ~1 | Tree
        (Intercept) Residual
StdDev:    19.73873 15.26082
```

effets aléatoires

grande variabilité entre les arbres et à l'intérieur des arbres

corrélation entre observations d'un même arbre

```
> (19.73873^2)/((19.73873^2)+(15.26082^2))
[1] 0.6258814
```

Exemple de croissance d'orangers

effets fixes

```
...
Fixed effects: circumference ~ age
               Value Std.Error DF  t-value p-value
(Intercept) 17.39965 10.423696 29  1.66924  0.1058
age          0.10677  0.005321 29 20.06585  0.0000
...
```

La circonférence des orangers augmente substantiellement avec l'âge de l'arbre.

Output identique avec SAS `proc MIXED` (voir fichier SAS).

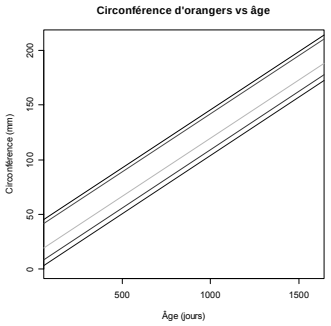
Exemple de croissance d'orangers

On peut représenter graphiquement les résultats

```
> coef(mod_orange)
  (Intercept)       age
3   -2.737898 0.1067703
1    2.395202 0.1067703
5   13.056257 0.1067703
2   35.299693 0.1067703
4   38.984996 0.1067703
```

le modèle a ajusté 1 intercepte aléatoire pour chaque arbre, effet de l'âge est constant pour tous les arbres

(conceptuellement, c'est une analyse de covariance qui considère arbre comme effet aléatoire)



Design pour modèles mixtes

Si peu de réplication à l'intérieur des effets aléatoires (1-2 obs/site), peu efficace de traiter comme effet aléatoire.

Généralement, si le design comprend des éléments nichés (hiérarchie), il est préférable d'inclure la structure des données dans le modèle, même si certains effets aléatoires ne sont pas importants.

Taille d'échantillon

Performance augmente avec taille d'échantillon (50, 100 ou +)

avec petits échantillons devrait fonctionner aussi bien que régression classique (variance plus difficile à estimer – moins précise)

Nombres de groupes

lorsqu'il y a très peu de groupes, il devient difficile d'estimer la variance entre groupe (réponse très semblable à régression classique)

si seulement 1 ou 2 groupes, on a avantage à utiliser régression classique et traiter groupe comme variable explicative

Taille d'échantillon

Nombre d'observations dans les groupes

1 ou 2 observations/groupe sont suffisantes pour ajuster le modèle (amène tout de même une information sur la variance et les coéfficients)

plus on a d'observation, meilleure sera l'estimation du coéfficient aléatoire associé à ce site (meilleure estimation de la variance)

Modèles hiérarchiques

Souvent plusieurs paramétrisations possibles et équivalentes pour un même problème

un design en split-plot peut être analysé comme un modèle hiérarchique (ou comme autre type de modèle mixte)

Allons voir un exemple de l'analyse d'un tel dispositif en agriculture...

Ex. – Azote et variétés d'avoine

Design en split plot

0	0.2	0.4	0	0.2	0.6
0.6	0.4	0.6	0.2	0.4	0
0	0.2	0.6	0.2	0.2	0
0.4	0.6	0.4	0	0.4	0.6
0.2	0	0	0.2	0.6	0.4
0.4	0.6	0.4	0.6	0.2	0
0	0.2	0.6	0	0.4	0.2
0.4	0.6	0.2	0.4	0.6	0
0	0.4	0	0.2	0.4	0
0.2	0.6	0.4	0.6	0.2	0.6
0	0.4	0.2	0.4	0	0.2
0.2	0.6	0.6	0	0.4	0.6

6 blocs

chaque bloc divisé en 3 sections avec application d'une variété d'avoine

- ☐ Marvellous
- ☐ Golden Rain
- ☐ Victory

chaque section divisée en sous-parcelles où on applique 1 de 4 concentration d'azote (cwt/acre)

0, 0.2, 0.4, 0.6

variété d'avoine: facteur appliqué à l'échelle du bloc (whole plot factor)

concentration d'azote: facteur appliqué à l'échelle des sections des blocs (split-plot factor)

Ex. – Azote et variétés d'avoine

Jeu de données inclus dans package `nlme`

```
> Oats[1:10, ]
 Block     Variety nitro yield
1      I     Victory   0.0   111
2      I     Victory   0.2   130
3      I     Victory   0.4   157
4      I     Victory   0.6   174
5      I Golden Rain   0.0   117
6      I Golden Rain   0.2   114
7      I Golden Rain   0.4   161
8      I Golden Rain   0.6   141
9      I  Marvellous   0.0   105
10     I  Marvellous   0.2   140
```

On s'intéresse à l'effet de variété et concentration d'azote sur la quantité d'avoine produite (boisseau/acre; 1 boisseau ~ 36 L)

Ex. – Azote et variétés d'avoine

On peut analyser ces données provenant d'un design en split-plot comme modèle hiérarchique

```
> Oats$nitro<-as.factor(Oats$nitro)
```

pour s'assurer que nitro soit considéré comme facteur

On pourrait ajuster le modèle complet

```
> mod1<-lme(yield~nitro+Variety+nitro:Variety, random=~1 |
Block/Variety, data=Oats)
```

le terme Block/Variety peut être vu comme étant la variété nichée dans le bloc (ou comme un terme Block avec une interaction Block:Variety)

Variety apparaît comme effet fixe et effet aléatoire ????????

le terme aléatoire permet de modéliser l'effet niché (tient compte de la variabilité entre parcelles d'un même bloc pour une même variété)

le terme fixe permet de modéliser l'effet de la variété sur la quantité d'avoine produite

Ex. – Azote et variétés d'avoine

Dans SAS:

Deux stratégies complètement équivalentes:

Analyse exécutée comme un split-plot classique:

```
proc mixed data=oats;
class Block Nitro Variety;
model Yield = Nitro Variety Nitro*Variety /solution;
random Block Block*Variety;
run;
```

Analyse exécutée comme un modèle hiérarchique:

```
proc mixed data=oats;
class Block Nitro Variety;
model Yield = Nitro Variety Nitro*Variety /solution;
random int /subject=Block;
random int /subject=Variety(Block);
run;
```

Ex. – Azote et variétés d'avoine

Regardons l'output du modèle général:

```
> summary(mod1)
Linear mixed-effects model fit by REML
 Data: Oats
       AIC      BIC    logLik
  559.0285 590.4437 -264.5143

Random effects:
 Formula: ~1 | Block
        (Intercept)
StdDev:    14.64496

 Formula: ~1 | Variety %in% Block
        (Intercept) Residual
StdDev:    10.29862 13.30727
...
```

Variabilité due aux blocs

$\sigma_1 = 14.64 \quad (\sigma_1^2 = 214.47)$

Variabilité de variété à l'intérieur de blocs

$\sigma_2 = 10.30 \quad (\sigma_2^2 = 106.06)$

Variabilité à l'intérieur des sections

$\sigma = 13.31 \quad (\sigma^2 = 177.08)$

Ex. – Azote et variétés d'avoine

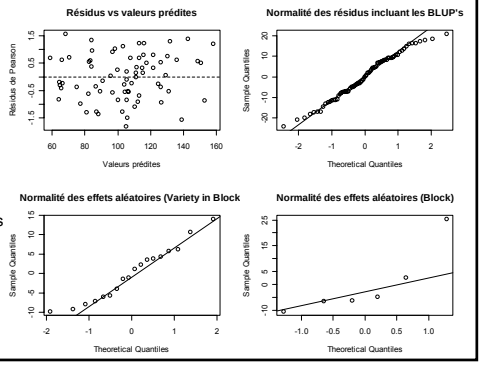
```
...
Fixed effects: yield ~ nitro + Variety + nitro:Variety
                               Value Std.Error DF   t-value p-value
(Intercept)                 80.00000  9.106958 45  8.784492  0.0000
nitro0.2                    18.50000  7.682958 45  2.407927  0.0202
nitro0.4                    34.66667  7.682958 45  4.512151  0.0000
nitro0.6                    44.83333  7.682958 45  5.835426  0.0000
VarietyMarvellous            6.66667  9.715026 10  0.686222  0.5082
VarietyVictory              -8.50000  9.715026 10 -0.874933  0.4021
nitro0.2:VarietyMarvellous  3.33333 10.865343 45  0.306786  0.7604
nitro0.4:VarietyMarvellous -4.16667 10.865343 45 -0.383482  0.7032
nitro0.6:VarietyMarvellous -4.66667 10.865343 45 -0.429500  0.6696
nitro0.2:VarietyVictory    -0.33333 10.865343 45 -0.030679  0.9757
nitro0.4:VarietyVictory     4.66667 10.865343 45  0.429500  0.6696
nitro0.6:VarietyVictory     2.16667 10.865343 45  0.199411  0.8428
...
```

Avec tous ces facteurs, ce sera plus pratique de vérifier l'effet global de chacun avec `anova( )`.

Ex. – Azote et variétés d'avoine

Avant d'interpréter les résultats, vérifions les suppositions du modèle.

Bonne répartition des résidus (homoscédasticité), normalité des résidus et des effets aléatoires.



Ex. – Azote et variétés d'avoine

```
> anova(mod1, type="sequential")
             numDF denDF   F-value p-value
(Intercept)      1    45 245.14299  <.0001
nitro            3    45  37.68561  <.0001
Variety          2    10   1.48534  0.2724
nitro:Variety    6    45   0.30282  0.9322
```

ajout des termes de façon séquentielle dans le même ordre que dans la formule

Pas d'interaction nitro:Variety.

Pas d'effet de Variety.

Effet important de concentration d'azote (nitro).

On peut ajuster un modèle avec seulement nitro comme effet fixe.

```
> mod2<-lme(yield~nitro, random=~1 | Block/Variety, data=Oats)
```

Ex. – Azote et variétés d'avoine

```
> summary(mod2)
...
Random effects:
 Formula: ~1 | Block
        (Intercept)
StdDev:    14.50597

 Formula: ~1 | Variety %in% Block
        (Intercept) Residual
StdDev:    11.03867 12.74987

Fixed effects: yield ~ nitro
               Value Std.Error DF   t-value p-value
(Intercept) 79.38889  7.132404 51 11.130734       0
nitro0.2    19.50000  4.249955 51  4.588283       0
nitro0.4    34.83333  4.249955 51  8.196164       0
nitro0.6    44.00000  4.249955 51 10.353050       0
```

contrastes de traitement

Volume d'avoine produit augmente avec la concentration d'azote.

Contrastes

Les types de contrastes par défaut sont ceux qui comparent chaque niveau à un niveau de référence.

R: niveau le plus bas

SAS: niveau le plus élevé

Si on le désire, on peut utiliser des contrastes orthogonaux (*a priori*) pour comparer des groupes de traitements entre eux.

particulièrement utiles avec des facteurs qui reflètent un gradient

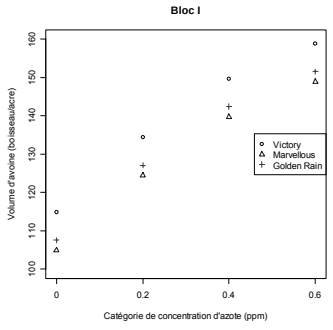
polyvalents, car on peut les utiliser avec ANOVA's, GLM's, modèles mixtes

Ex. – Azote et variétés d'avoine

On peut utiliser un graphique pour interpréter les résultats (ici, pour le bloc I)

La fonction `predict( )` permet de calculer les valeurs prédites en tenant compte ou non des effets aléatoires

Il serait préférable d'ajouter des barres d'erreurs (*SE* ou *IC*) - plus compliqué pour modèles mixtes et utilisation de simulations nécessaires



Problèmes potentiels avec modèles mixtes

Dans certains cas, l'estimation de la variance due aux effets aléatoires est difficile (problèmes d'estimation de la matrice de variance-covariance).

- se produit lorsque la variance estimée est négative (variance doit être > 0)
- lorsque la variabilité de l'effet aléatoire est 0 (variable à effet négligeable)

Un bon moyen de vérifier si ce problème existe, est de calculer les intervalles de confiance autour des estimés avec `intervals( )`.

- Des *IC*'s anormalement larges ou avec des bornes infinies sont une indication d'un tel problème ou d'une mauvaise formulation du modèle (erreur dans les niveaux).

Ex. – Gain de masse chez souris

Ex. Gain de masse chez des souris en fonction de diète et condition de cage

- cages sont considérées comme blocs (6 cages)
- chaque cage est divisée en 2 sections
- on applique une condition de séjour (4 possibles) à chaque section
- 3 souris par section (chacune reçoit 1 de 3 diètes)

Diète1 Diète3 Diète2	Diète1 Diète3 Diète2

Diète1 Diète3 Diète2	Diète1 Diète3 Diète2

condition de séjour est un facteur appliqué au niveau des blocs (whole-plot factor)

diète est un facteur appliqué au niveau des sections (split-plot factor)

Ex. – Gain de masse chez souris

Jeu de données « Mice.txt »

```
> mice[1:12, ]
   cage condition     diet gain
1     1         1   normal   58
2     1         1 restrict   58
3     1         1 suppleme   58
4     1         2   normal   54
5     1         2 restrict   46
6     1         2 suppleme   57
7     2         1   normal   49
8     2         1 restrict   50
9     2         1 suppleme   57
10    2         3   normal   57
11    2         3 restrict   53
12    2         3 suppleme   62

> mice$cage<-as.factor(mice$cage)
> mice$condition<-as.factor(mice$condition)
> mod1<-lme(gain~diet+condition+diet:condition, random=~1|cage/condition,
data=mice)
```

étant donné que chaque cage ne peut accomoder que 2 conditions de séjour (sur 4 possibles), les blocs sont incomplets

Ex. – Gain de masse chez souris

Regardons l'output et l'estimation de la variance

```
> summary(mod1)
Linear mixed-effects model fit by REML
 Data: mice
       AIC      BIC    logLik
  193.2238 210.8946 -81.61188

Random effects:
 Formula: ~1 | cage
        (Intercept)
StdDev:    1.742886

 Formula: ~1 | condition %in% cage
         (Intercept) Residual
StdDev: 0.0003216755 5.276632
```

Variabilité due aux cages

$\sigma_1 = 1.74 \quad (\sigma_1^2 = 3.04)$

Variabilité de condition dans cage

$\sigma_2 = 0.0003 \quad (\sigma_2^2 = 1.03 \times 10^{-7})$

Variabilité à l'intérieur des sections

$\sigma = 5.28 \quad (\sigma^2 = 27.84)$

une variance de 0, c'est un problème

Ex. – Gain de masse chez souris

on peut essayer de calculer les *IC*'s avec `intervals( )`

```
> intervals(mod1)
Erreur dans intervals.lme(mod1) :
  Cannot get confidence intervals on var-cov components: Non-
positive definite approximate variance-covariance
```

on a un message que la matrice de variance-covariance ne peut-être inversée (problème d'estimation)

(SAS nous informe de ce problème dans la fenêtre log, il faut la vérifier)

Ce modèle n'est pas optimal et il faudrait soit

1) opter pour un modèle plus simple (enlever condition des effets aléatoires)

2) incorporer une structure de corrélation des erreurs dans le modèle (`corStruct`) pour plus de flexibilité

3) opter pour une autre stratégie d'analyse

Modèles hiérarchiques

Retournons voir d'autres exemples de modèles hiérarchiques

extension de la régression linéaire et concept d'ANCOVA avec effet aléatoire

Exemple de croissance - OrthoFem

Ex. Croissance mesurée par la distance entre deux fissures sur le crâne (pituitaire et ptérygomaxillaire) chez 11 filles à intervalles de 2 ans (8 - 14 ans).

Jeu de données « OrthoFem.txt »

```
> OrthoFem[1:8, ]
Grouped Data: distance ~ age | Subject
    distance age Subject    Sex
65      21.0   8     F01 Female
66      20.0  10     F01 Female
67      21.5  12     F01 Female
68      23.0  14     F01 Female
69      21.0   8     F02 Female
70      21.5  10     F02 Female
71      24.0  12     F02 Female
72      25.5  14     F02 Female
```

Faisons rouler ce modèle simple:

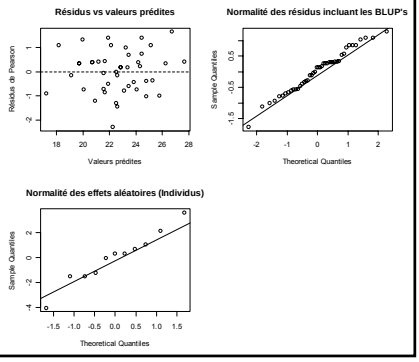
```
> mod_fm1 <- lme(distance~age, data=OrthoFem, random=~1 | Subject)
```

Exemple de croissance - OrthoFem

Bonne répartition des résidus.

Normalité des résidus.

Normalité des effets aléatoires.



Exemple de croissance - OrthoFem

```
> summary(mod_fm1)
Linear mixed-effects model fit by REML
 Data: OrthoFem
       AIC     BIC    logLik
  149.2183 156.169 -70.60916

Random effects:
 Formula: ~1 | Subject
        (Intercept)  Residual
StdDev:     2.06847 0.7800331
...
```

Variabilité entre individus

$\sigma_1 = 2.07 \quad (\sigma_1^2 = 4.28)$

Variabilité à l'intérieur des individus

$\sigma = 0.78 \quad (\sigma^2 = 0.61)$

Exemple de croissance - OrthoFem

```
...
Fixed effects: distance ~ age
               Value Std.Error DF   t-value p-value
(Intercept) 17.372727 0.8587419 32 20.230440       0
age          0.479545 0.0525898 32  9.118598       0
...
```

La taille de la fissure (croissance) augmente avec l'âge.

```
> coef(mod_fm1)
    (Intercept)       age
F10    13.36740 0.4795455
F09    15.90228 0.4795455
F06    15.90228 0.4795455
F01    16.14369 0.4795455
F05    17.35078 0.4795455
F07    17.71291 0.4795455
F02    17.71291 0.4795455
F08    18.07503 0.4795455
F03    18.43716 0.4795455
F04    19.52353 0.4795455
F11    20.97204 0.4795455
```

le modèle nous donne les estimés suivants corrigés pour les BLUP's

- intercepte différent pour chaque individu
- effet de l'âge constant pour tout le monde

on pourrait vérifier si l'hypothèse des droites parallèles est plausible

Exemple de croissance - OrthoFem

On peut observer la variabilité des estimés d'intercepte et d'âge pour chaque individu avec `lmList( )`

```
> mod_fm<-lmList(distance~age | Subject, data=OrthoFem)
```

fait un `lm( )` – modèle avec effet fixe seulement - séparé pour chaque groupe (ici, Subject)

```
> coef(mod_fm)
    (Intercept)   age
F10       13.55 0.450
F09       18.10 0.275
F06       17.00 0.375
F01       17.25 0.375
F05       19.60 0.275
F07       16.95 0.550
F02       14.20 0.800
F08       21.45 0.175
F03       14.40 0.850
F04       19.65 0.475
F11       18.95 0.675
```

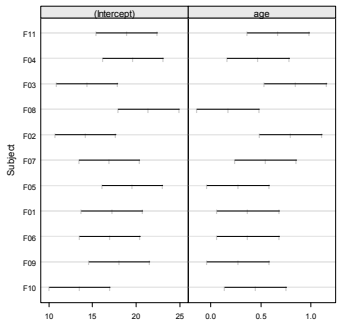
Exemple de croissance - OrthoFem

on peut représenter la variabilité des estimés lorsqu' estimés séparément par modèle avec effets fixes avec `intervals( )`

```
> plot(intervals(mod_fm))
```

on remarque que la plupart des *IC*'s se chevauchent

suggère qu'un modèle avec interaction avec âge ne serait pas nécessairement meilleur



Exemple de croissance - OrthoFem

Faisons rouler un modèle mixte ajoutant un effet aléatoire d'âge (effet de l'âge différent pour chaque individu).

```
> mod_fm2<-lme(distance~age, data=OrthoFem, random=~age | Subject)
> summary(mod_fm2)
Linear mixed-effects model fit by REML
 Data: OrthoFem
       AIC      BIC    logLik
  149.4287 159.8547 -68.71435

Random effects:
 Formula: ~age | Subject
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 1.8841866 (Intr)
age         0.1609278 -0.354
Residual    0.6682746
...
```

Exemple de croissance - OrthoFem

```
...
Fixed effects: distance ~ age
               Value Std.Error DF   t-value p-value
(Intercept) 17.372727 0.7606027 32 22.840737       0
age          0.479545 0.0662140 32  7.242353       0
...
```

On peut utiliser un test du ratio du likelihood pour comparer les deux modèles.

```
> anova(mod_fm1, mod_fm2)
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
mod_fm1     1  4 149.2183 156.1690 -70.60916
mod_fm2     2  6 149.4287 159.8547 -68.71435 1 vs 2 3.789622  0.1503
```

la statistique approxime un X^2 mais le test est conservateur (donne *P* plus grand qu'il le devrait)

Exemple de croissance - OrthoFem

On peut utiliser des simulations avec `simulate.lme( )` pour déterminer la distribution de la statistique lorsque H_0 est vraie.

```
> orthLRTsim <- simulate.lme(object=mod_fm1, m2=mod_fm2, nsim=1000)
```

`object` = modèle nul `m2` = modèle alternatif

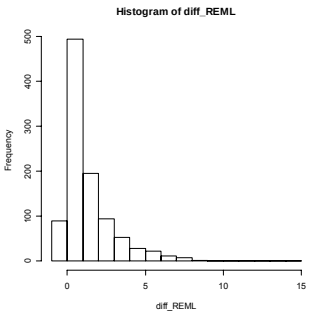
`nsim` = nombre de simulations

l'objet `orthLRTsim` est une liste contenant les valeurs de LL pour ML et REML

Exemple de croissance - OrthoFem

On peut calculer la différence de *LL* et illustrer les résultats

```
>diff_REML<- -2*(orthLRTsim$null$REML[,2]-
orthLRTsim$alt$REML[,2]

>hist(diff_REML)
```



On détermine la probabilité d'observer une valeur ≥ *LL* observé

```
>length(which(diff_REML>=3.789))/1000

[1] 0.082
```

Valeur considérablement plus petite que celle fournie dans l'output d'anova (*P* = 0.1503 - test conservateur)

Exemple de croissance - OrthoFem

```
> plot(orthLRTsim,df=c(1,2))
```

« Nominal *p*-value » = valeur obtenue en utilisant le X^2 avec *df* de 1, 2 ou mélange 1 et 2.

« Empirical *p*-value » = valeur obtenue par simulation des LRT.

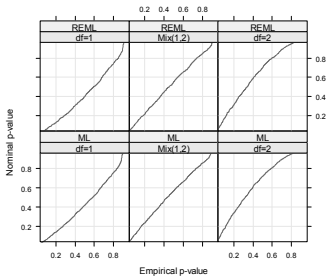
On devrait obtenir une ligne à 45º si le test fonctionne bien.

Avec *df* = 2, *P* observé > vrai *P* (conservateur)

Ici, le X^2 est un hybride entre *df* = 1 et *df* = 2.

(certains suggèrent d'utiliser $0.5\chi_1^2 + 0.5\chi_2^2$ comme statistique)

La simulation est préférable.



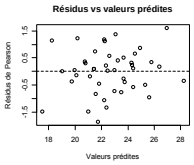
Exemple de croissance - OrthoFem

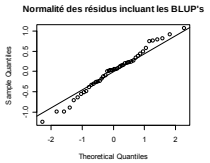
Vérification des suppositions.

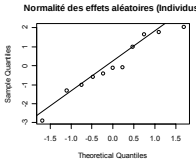
Bonne répartition des résidus.

Normalité des résidus.

Normalité des effets aléatoires.







Exemple de croissance - OrthoFem

On peut conclure que l'ajout de l'interaction âge améliore marginalement le modèle.

intercepte différent et effet d'âge différent pour chaque individu

```
> coef(mod_fm2)
    (Intercept)       age
F10    14.48469 0.3758789
F09    17.26521 0.3529593
F06    16.77334 0.3986632
F01    16.95617 0.4040977
F05    18.36221 0.3855663
F07    17.28381 0.5194036
F02    16.05413 0.6336632
F08    19.40258 0.3561659
F03    16.35669 0.6728186
F04    19.02396 0.5258844
F11    19.13721 0.6498990
```

Exercices

Particularités des modèles hiérarchiques

Avec une structure hiérarchique, on peut déterminer l'effet de variables provenant de différents niveaux sur la variable réponse.

Ex.
quantité de graisse sur lièvres provenant de différents sites

variables explicatives potentielles:

taille du lièvre (au niveau de l'individu)

espèce arbustive dominante (au niveau du site)

Ex.
croissance de semis (*Abies balsamea*) dans différents quadrats sur différents sites

variables explicatives potentielles:

origine du semis (au niveau du semis)

% ouverture de la canopée au dessus du quadrat (au niveau du quadrat)

pH moyen du sol (au niveau du site)

Extensions des modèles mixtes

Deux importantes suppositions du modèle mixte/modèle hiérarchique de base:

1) la variance résiduelle est homogène
2) les résidus d'un même niveau sont indépendants

On a donc un problème si:

1) les variances ne sont pas homogènes (patron d'entonnoir)
2) les observations d'un même groupes sont corrélées (mesures répétées)

Afin de régler ces problèmes, on peut décomposer la matrice de variance-covariance intra-groupe en un produit de matrices plus simples dans le modèle mixte:

une matrice pour la structure de la variance et une autre pour la structure de corrélation

Structure de variance

Variance hétérogène (hétéroscédasticité)

la matrice de structure de variance permet de modéliser l'hétéroscédasticité des résidus en fonction de covariables

e.g., ajuster des modèles avec des variances différentes selon le groupe

(arguments `weights =` de R, `group=` de SAS)

Quelques structures possibles:

`varFixed` – variance fixe ($\sigma^2 Variable_{ij}$)

`varIdent` – variance différente par niveau de facteur ($\sigma^2 \delta^2_{s_{ij}}$; où δ = paramètre de variance pour une strate donnée)

`varPower` – variance diffère selon la covariable à une puissance 2δ ($\sigma^2 |Variable_{ij}|^{2\delta}$; où δ peut permettre à la variance d'augmenter ou diminuer avec la valeur absolue de la covariable) – non approprié lorsque la covariable a des valeurs près de 0

`varConstPower` – variance différente par niveau de facteur ($\sigma^2(\delta_1 + |Variable_{ij}|^{\delta_2})^2$; où δ_1 = est une constante positive et δ_2 permet à la variance d'augmenter ou diminuer avec la covariable) – approprié lorsque la covariable est près de 0

Ex. – Masse des rats

Ex. Masse de rats selon la diète

```
> rats[1:10, ]
    Weight Time Rat Diet
1      240    1   1    1
2      250    8   1    1
3      255   15   1    1
4      260   22   1    1
5      262   29   1    1
6      258   36   1    1
7      266   43   1    1
8      266   44   1    1
9      265   50   1    1
10     272   57   1    1
11     278   64   1    1
```

Masse en g

Chaque rat mesuré à tous les 7 jours (+ 1 obs au jour 44)

3 diètes

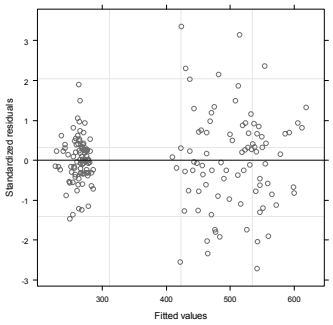
Avec un modèle de base

```
> mod1<-lme(Weight ~ Diet+Time+Diet:Time, random=~1 | Rat, data=rats)
```

Ex. – Masse des rats

Il y a nettement un patron hétérogène de la variance

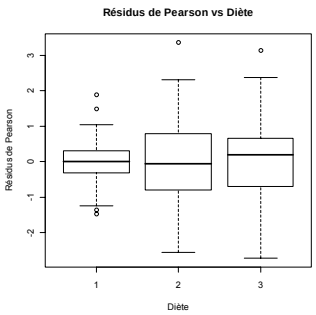
Au lieu d'essayer des transformations, essayons de modéliser l'hétérogénéité dans la variance



Ex. – Masse des rats

on peut évaluer la variation des résidus en fonction de la diète

on remarque une variabilité différente selon la diète (beaucoup plus faible pour diète 1)



Ex. – Masse des rats

Ajustons un modèle qui permet à la variance de varier selon la diète

```
> mod2<-lme(Weight~Diet+Time+Diet:Time, random=~1 | Rat,
weights=varIdent(form=~ 1 | Diet), data=rats)
```

dans l'argument `weights, varIdent( )` spécifie le type de fonction de variance

`form= ~1 | Diet` permet d'avoir une variance différente selon la diète

comme alternative, on aurait aussi pu opter de permettre à la variance d'augmenter avec les valeurs prédites

```
> mod3<-lme(Weight~Diet+Time+Diet:Time, random=~1 | Rat,
weights=varPower(), data=rats)
```

si on ne spécifie pas d'autres arguments, `varPower( )` utilise `form= ~ fitted(.)` par défaut

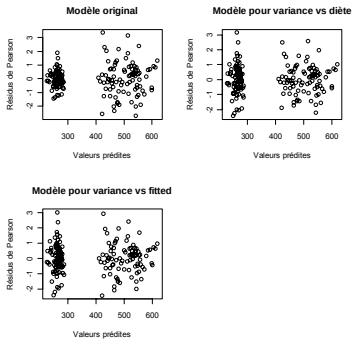
`varPower( )` est correct puisque valeurs prédites sont loin de 0 (autrement `varConstPower( )`)

Ex. – Masse des rats

la variance hétérogène avec le modèle original (`mod1`)

la variance stabilisée avec `varIdent( )` avec `mod2`

la variance stabilisée avec `varPower( )`avec `mod3`



Ex. – Masse des rats

On peut comparer l'ajustement du modèle permettant à la variance de varier avec la diète vs le modèle original

```
> anova(mod1, mod2)
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
mod1     1  8 1248.245 1273.332 -616.1226
mod2     2 10 1210.210 1241.568 -595.1049 1 vs 2 42.03536  <.0001
```

Ceci justifie nettement l'inclusion de paramètres pour estimer la variance (*P* est conservateur, et il est inférieur à ce qui est affiché).

Ex. – Masse des rats

```
> summary(mod2)
...
Random effects:
 Formula: ~1 | Rat
        (Intercept) Residual
StdDev:    36.53504 3.846144

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | Diet
 Parameter estimates:
       1        2        3
1.000000 2.242866 2.027319
```

Variabilité entre rats:

$\sigma^2_{Rats} = 36.54^2 = 1334.81$

Variabilité résiduelle

$\sigma^2 = 3.85^2 = 14.79$

Variabilité résiduelle pour Diète 1

$\sigma^2_{diète1} = \sigma^2 \delta^2_{diète1} = 14.79 \cdot 1^2 = 14.79$

Variabilité résiduelle pour Diète 2

$\sigma^2_{diète2} = \sigma^2 \delta^2_{diète2} = 14.79 \cdot 2.24^2 = 74.41$

Variabilité résiduelle pour Diète 3

$\sigma^2_{diète3} = \sigma^2 \delta^2_{diète3} = 14.79 \cdot 2.03^2 = 60.80$

Ex. – Masse des rats

```
...
Fixed effects: Weight ~ Diet + Time + Diet:Time
               Value Std.Error  DF   t-value p-value
(Intercept) 251.65165 12.942917 157 19.443194       0
Diet2       200.66549 22.537562  13  8.903602       0
Diet3       252.07168 22.510266  13 11.198076       0
Time          0.35964  0.021076 157 17.063785       0
Diet2:Time    0.60584  0.070095 157  8.643121       0
Diet3:Time    0.29834  0.063997 157  4.661759       0
```

On peut vérifier l'effet global de l'interaction avec `anova( )`

```
> anova(mod2)
            numDF denDF   F-value p-value
(Intercept)     1   157 1765.0654  <.0001
Diet            2    13   88.4327  <.0001
Time            1   157  528.9844  <.0001
Diet:Time       2   157   44.6659  <.0001
```

On conclut que l'effet de la diète diffère selon le temps écoulé.

Extensions des modèles mixtes

Mesures répétées

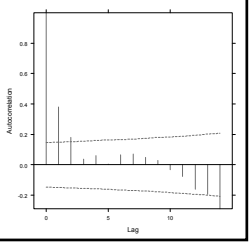
la matrice de structure de corrélation permet de modéliser la dépendance entre observations intra-groupe (analogue à analyses spatio-temporelles)

on peut utiliser la fonction `ACF( )` pour déterminer la corrélation des résidus

`plot(ACF( ) )` permet de visualiser l'autocorrelation des résidus avec des IC's autour de chaque corrélation

si les corrélations sont importantes, une bonne idée d'en tenir compte

quelques structures possibles: AR1, ARMA, compound symmetry, générale, spatiales (linéaire, exponentielle, ou autres)



Structure de corrélations

Permet d'imposer structure de corrélation particulière aux résidus

`corAR1` – structure de corrélation autorégressive d'ordre 1 (intervalles entiers; $-1 > \rho > 1$)

`corCAR1` – structure de corrélation autorégressive d'ordre 1 pour variable temporelle continue (intervalle non entiers; $\rho > 0$)

`corARMA` – structure de corrélation autorégressive avec moyenne mobile (séries temporelles)

`corCompSymm` – structure de corrélation échangeable (symétrie composée)

`corExp` – structure de corrélation spatiale exponentielle (spatiale)

`corGaus` – structure de corrélation spatiale gaussienne (spatiale)

`corSpher` – structure de corrélation spatiale sphérique (spatiale)

`corLin` – structure de corrélation spatiale linéaire (spatiale)

Ex. – Volume d'épinettes vs âge

On peut essayer d'analyser l'augmentation du volume (hauteur x diamètre^2) de 79 épinettes de Sitka mesurées à 13 reprises en fonction de l'âge sur 4 sites.

jeu de données Spruce `data(Spruce)`

```
> Spruce[1:15, ]
Grouped Data: logSize ~ days | Tree
      Tree days logSize plot
1    O1T01  152    4.51    1
2    O1T01  174    4.98    1
3    O1T01  201    5.41    1
4    O1T01  227    5.90    1
5    O1T01  258    6.15    1
6    O1T01  469    6.16    1
7    O1T01  496    6.18    1
8    O1T01  528    6.48    1
9    O1T01  556    6.65    1
10   O1T01  579    6.87    1
11   O1T01  613    6.95    1
12   O1T01  639    6.99    1
13   O1T01  674    7.04    1
14   O1T02  152    4.24    1
15   O1T02  174    4.20    1
```

tous les individus ont été mesurés aux 13 mêmes dates

intervalles entre mesures

```
> Days<-
as.numeric(rownames(table(Spruce$days)))

> diff(x=Days, lag=1)
[1]  22  27  26  31 211  27  32  28  23
34  26  35
```

`corAR1` n'est donc pas appropriée

Ex. – Volume d'épinettes vs âge

On pourrait commencer avec une structure de corrélation échangeable:

(on suppose que chaque arbre a une croissance particulière)

```
> mod1<-lme(logSize~days, random=~1 | plot,
correlation=corCompSymm(form=~1 | plot/Tree), data=Spruce)
```

effet aléatoire de `plot`

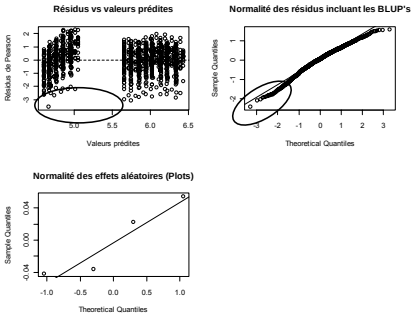
corrélation fixe pour observations d'un même arbre à l'intérieur d'un site

Ex. – Volume d'épinettes vs âge

Répartition des résidus moyenne, et il semble y avoir quelques valeurs extrêmes.

Normalité des résidus correcte pour la plupart des données, mais déviation dans l'une des queues.

Normalité des effets aléatoires (peu d'observations pour Plot).



Ex. – Volume d'épinettes vs âge

```
> summary(mod1)
Linear mixed-effects model fit by REML
 Data: Spruce
       AIC      BIC    logLik
  817.7945 842.4568 -403.8973

Random effects:
 Formula: ~1 | plot
        (Intercept)  Residual
StdDev:  0.08906093 0.6888212

Correlation Structure: Compound symmetry
 Formula: ~1 | plot/Tree
 Parameter estimate(s):
      Rho
0.8047096
...
```

Variabilité entre sites (faible variation entre sites)

$\sigma_1 = 0.09 \quad (\sigma_1^2 = 0.008)$

Variabilité intra site

$\sigma = 0.69 \quad (\sigma^2 = 0.47)$

Corrélation entre observations d'un arbre

$\rho = 0.80$

Ex. – Volume d'épinettes vs âge

```
...
Fixed effects: logSize ~ days
               Value  Std.Error   DF  t-value p-value
(Intercept) 4.140124 0.08704219 1022 47.56457       0
days        0.003322 0.00005070 1022 65.51744       0
```

CONCLUSION:
le volume augmente avec le nombre de jours écoulés
(chaque jour entraîne une augmentation de 0.003 unités sur échelle log – *ca*. 1)

```
> intervals(mod1, levels=0.95)
Approximate 95% confidence intervals

  Fixed effects:
                 lower       est.       upper
(Intercept) 3.969322342 4.140124181 4.310926021
days        0.003222341 0.003321832 0.003421323
attr(,"label")
[1] "Fixed effects:"
...
```

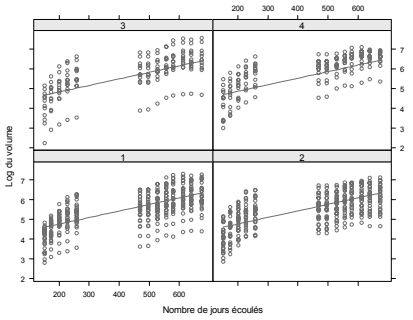
bonne précision

Ex. – Volume d'épinettes vs âge

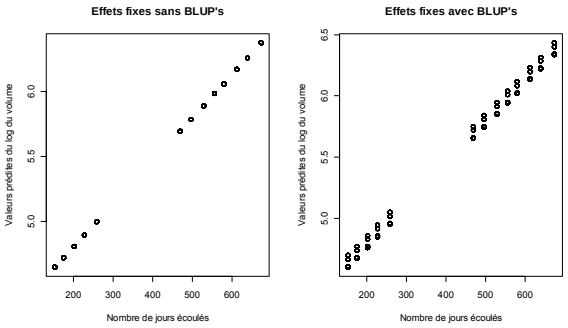
```
...
Random Effects:
  Level: plot
                       lower       est.     upper
sd((Intercept)) 0.003480982 0.08906093 2.278624

Correlation structure:
        lower      est.     upper
Rho 0.7456299 0.8047096 0.8525089
attr(,"label")
[1] "Correlation structure:"

 Within-group standard error:
    lower      est.     upper
0.6038097 0.6888212 0.7858016
```

peu de variabilité entre sites, mais *IC* plus large que les autres (moins précis – mais semble correct)

On peut illustrer les résultats pour chaque site à l'aide de `plot(augPred( ))`



On peut illustrer les résultats avec/sans effets aléatoires



Inclure ou exclure un effet aléatoire?

Pinheiro et Bates (2000) suggèrent que si la variabilité associée à un effet aléatoire est suffisamment faible, on peut l'exclure du modèle.

Les tests formels (`anova( )` et/ou simulations) peuvent être utiles, mais on peut aussi baser le choix sur la variabilité relative

Ex. comparer l'effet aléatoire associé à l'intercepte vs l'effet fixe de l'intercepte de `mod1`:

```
> (0.0890568)/abs(fixef(mod1)[1])*100
(Intercept)
   2.151066
```

variabilité relative de l'intercepte aléatoire est de 2.15%

Dans leur exemple, effet aléatoire enlevé lorsqu'il était très faible (0.006%)

On peut aussi utiliser la même stratégie pour une covariable qui apparaît comme effet fixe et effet aléatoire

Inclure ou exclure un effet aléatoire?

On peut tester si la structure de corrélation améliore réellement le modèle

```
> mod2<-lme(logSize~days, random=~1 |plot, data=Spruce)

> anova(mod1, mod2)
     Model df       AIC       BIC     logLik   Test  L.Ratio p-value
mod1     1  5  817.7945  842.4568  -403.8973
mod2     2  4 2144.7897 2164.5195 -1068.3949 1 vs 2 1328.995  <.0001
```

les deux modèles ont été solutionnés par REML, correct pour tester effets aléatoires (pas fixes)

le modèle plus simple s'ajuste moins bien aux données (logLik beaucoup plus petit)

on peut conclure que la structure de corrélation est appropriée ici puisque le LRT nous donne P < 0.0001 (et qu'il est en réalité plus petit qu'indiqué = conservateur)

Autre outils pour tester corrélations

La fonction `ACF( )` permet d'estimer l'autocorrélation empirique des résidus

appropriée pour déterminer la corrélation entre observations d'un même groupe prises à intervalles réguliers (pas le cas pour l'exemple d'épinettes...)

Pour le jeu de données de distance entre deux fissures du crâne de jeunes filles (`OrthoFem`)

```
> OrthoFem[1:5, ]
  distance age Subject    Sex
65       21.0   8     F01 Female
66       20.0  10     F01 Female
67       21.5  12     F01 Female
68       23.0  14     F01 Female
69       21.0   8     F02 Female
```

mesure à intervalle de 2 ans

```
> mod_fm2 <- lme( distance~age, data=OrthoFem, random=~age | Subject)
```

Autre outils pour tester corrélations

```
> ACF(mod_fm2)
  lag          ACF
1   0  1.000000000
2   1 -0.468000206
3   2 -0.254171422
4   3 -0.002390113
```

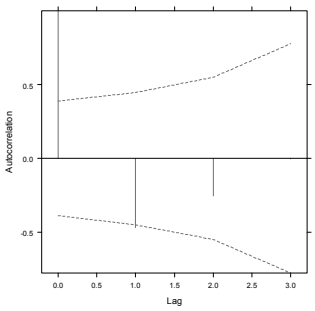
montre l'autocorrélation à différents lags

Pour visualiser l'autocorrélation

```
> plot(ACF(mod_fm2), alpha=0.01)
```

`alpha` = permet d'ajouter un intervalle de confiance à 1 - alpha%

une corrélation excédant les limites suggère la présence de corrélations entre les résidus



Extensions... mesures répétées

dans certains cas, effets aléatoires et structures de corrélation peuvent être redondants...

e.g., structure compound symmetry et effet aléatoire pour le même niveau d'organisation

toujours préférable d'organiser les données en ordre de la séquence des mesures pour chaque individu

```
ID     Mesure       Traitement
1      1            Low
1      2            Low
1      3            Low
1      4            Low
2      1            Med
2      2            Med
...
```

si vous modélisez la corrélation, R et SAS utilise les données adjacentes pour un même individu (si l'ordre n'est pas correct, la corrélation ne l'est pas non plus...)

Extensions des modèles mixtes

Modèles mixtes linéaires généralisés (generalized linear mixed models, GLMM's)

modèles mixtes avec variable réponse suivant une distribution autre que normale:

Poisson (valeurs entières positives, 0, 1, 2, 3, ... ∞)

e.g., nombre de parasites sur des individus, nombre d'individus/placette

binomiale (variable dichotomique – 0,1 ou proportion)

e.g., probabilité de germination, probabilité de reproduction

gamma (variable continue positive > 0)

e.g., temps écoulé avant germination, temps écoulé avant bris

Pour en savoir plus...

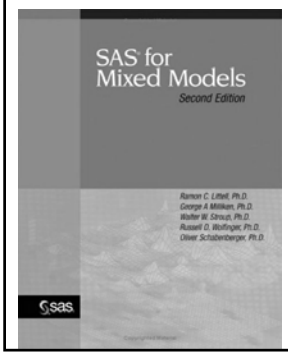
Cours BIO8093: analyse statistique de données complexes
offert à partir de l'UQAT (Rouyn-Noranda) à la session d'hiver

(disponible via vidéoconférence à l'UQAM et Laval entente CREPUQ)

Contenu du cours

- tests de randomisation et bootstrap
- régression robuste aux valeurs extrêmes
- maximum de vraisemblance, sélection de modèles et inférence multimodèles
- GLM's avancés: régression binomiale négative et gamma, régression logistique ordinale et multinomiale
- analyses de mesures répétées (GEE's)
- modèles mixtes linéaires et modèles mixtes linéaires généralisés
- simulations de Monte Carlo
- introduction aux analyses temporelles

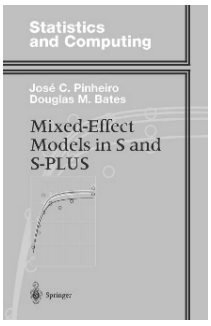
À absolument consulter



Littell, R. C., G. A. Milliken, W. W. Stroup, R. D. Wolfinger, and O. Schabenberger. 2006. SAS for mixed models, 2nd edition, Cary, NC, USA.

Bon livre avec plusieurs bons trucs de programmation et exemples de `MIXED` et `GLIMMIX`.

À absolument consulter



Pinheiro, J. C., and D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer Verlag, New York.

Un des meilleurs textes sur les modèles mixtes linéaires (2 premiers chapitres sont bien écrits) avec une multitude d'exemples clairs (utilise `lme( )`).

À absolument consulter

Gelman, A., and J. Hill. 2007. Data analysis using regression and multilevel/hierarchical models. Cambride University Press, New York, USA

Très bon texte sur les GLM's, simulations et modèles mixtes (utilise `lmer( )`).

Bonnes analyses!