

# Modèle mixte

## Motivation

Estimation d'un modèle mixte simple.

## Données

Un modèle linéaire avec effets aléatoires est simulé :

$$y_{i[j]} = \alpha_j + \beta_j x_{i[j]} + \epsilon_{i[j]}$$

Les individus, indicés par  $i$  appartiennent à des groupes, indicés par  $j$ .

```
# Paramètres
n_groupes <- 5
# Effectif moyen par groupe
n_individus_par_groupe <- 20
# Effectif réel par groupe
n_individus <- rpois(n_groupes, lambda = n_individus_par_groupe)
# Paramètres
sigma <- 1
```

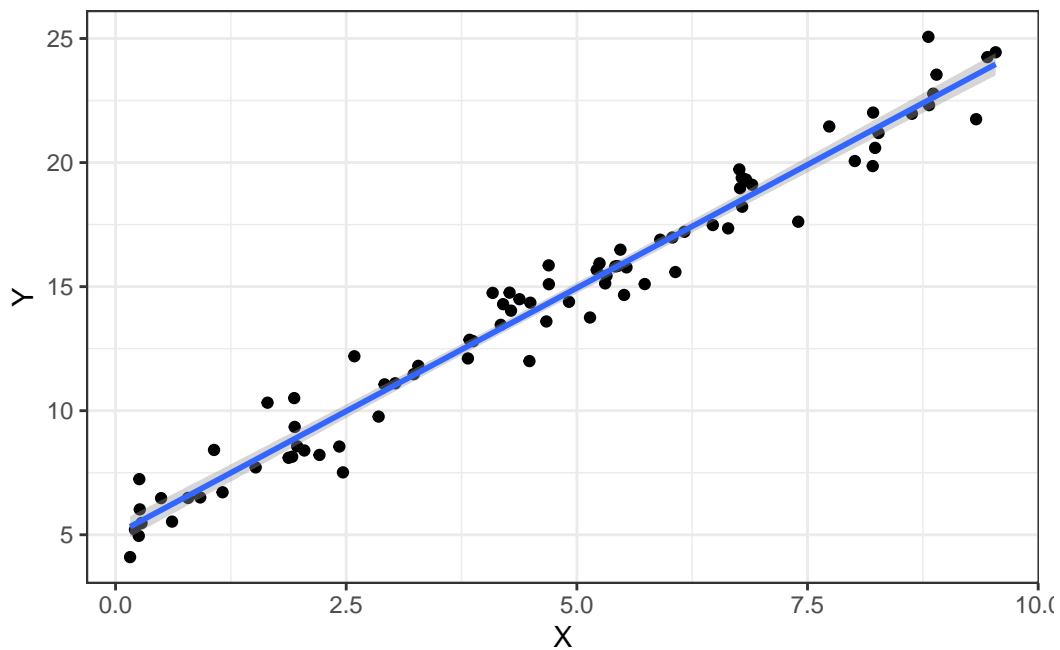
Les données sont simulées par une fonction.

```
library("tidyverse")
sim_data <- function(n_individus, n_groupes, alpha_j, beta_j) {
  n <- sum(n_individus)
  X <- runif(n, 0, 10)
  Y <- rep(alpha_j, n_individus) + rep(beta_j, n_individus) * X + rnorm(n, sd = sigma)
  seq_len(n_groupes) %>%
    rep(n_individus) %>%
    as.factor() -> groupe
  return(tibble(X, Y, groupe))
}
```

## Modèle linéaire

Les individus sont tous indépendants dans cette première version du modèle : tous les effets sont fixes.

```
alpha_j <- rep(5, n_groupes)
beta_j <- rep(2, n_groupes)
donnees <- sim_data(n_individus, n_groupes, alpha_j, beta_j)
donnees %>%
  ggplot(aes(x = X, y = Y)) +
  geom_point() +
  geom_smooth(method = "lm")
```



L'estimation utilise la fonction `lm()` du package `stats`.

```
modele_lm <- lm(Y ~ X, data = donnees)
summary(modele_lm)
```

Call:

```
lm(formula = Y ~ X, data = donnees)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.3950	-0.6214	0.0075	0.5055	2.5524

Coefficients:

Estimate	Std. Error	t value	Pr(> t )
----------	------------	---------	----------

```

(Intercept)  5.01300    0.21583    23.23   <2e-16
X            1.98669    0.04041    49.16   <2e-16

(Intercept) ***
X            ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.983 on 81 degrees of freedom
Multiple R-squared:  0.9676,    Adjusted R-squared:  0.9672
F-statistic: 2416 on 1 and 81 DF,  p-value: < 2.2e-16

```

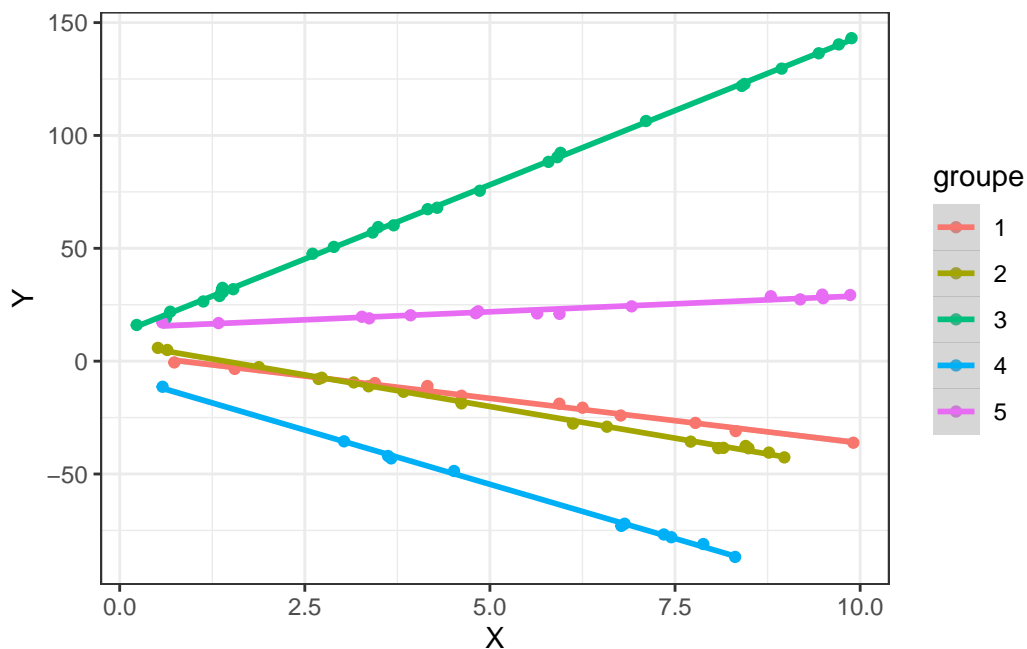
## Effets aléatoires

Les individus appartiennent à des groupes pour lesquels à la fois l'ordonnée à l'origine et la pente du modèle sont des effets aléatoires, distribués normalement autour des valeurs du modèle précédent.

```

alpha_j <- rnorm(n_groupes, mean = 5, sd = 5)
beta_j <- rnorm(n_groupes, mean = 2, sd = 5)
donnees <- sim_data(n_individus, n_groupes, alpha_j, beta_j)
donnees %>%
  ggplot(aes(x = X, y = Y, color = groupe)) +
  geom_point() +
  geom_smooth(method = "lm")

```



L'estimation est faite avec le package lme4.

```
library("lme4")
modele <- lmer(Y ~ (1|groupe) + (X||groupe), data = donnees)
summary(modele)
```

Linear mixed model fit by REML ['lmerMod']

Formula:

$Y \sim (1 \mid \text{groupe}) + ((1 \mid \text{groupe}) + (0 + X \mid \text{groupe}))$

Data: donnees

REML criterion at convergence: 301.7

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.35173	-0.66095	-0.02947	0.61926	2.24267

Random effects:

Groups	Name	Variance	Std.Dev.
groupe	(Intercept)	3.8859	1.9713
groupe.1	(Intercept)	66.6735	8.1654
groupe.2	X	62.9988	7.9372
Residual		0.8947	0.9459

Number of obs: 83, groups: groupe, 5

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	6.402	3.765	1.701

optimizer (nloptwrap) convergence code: 0 (OK)

unable to evaluate scaled gradient

Model failed to converge: degenerate Hessian with 1 negative eigenvalues