Modèles linéaires à effets mixtes

J Gaudart, R Giorgi,JC Thalabard, D Thiam, S Whegang

Septembre 2010

Rappel

Les données sur lesquelles nous allons travailler sont des données immunologiques du paludisme. La réponse anticorps de l'enfant se mesure en concentration d'anticorps $(\mu g/l)$. Le but de cette étude est d'expliquer le comportement immunitaire de nouveaux nés de 0 à 18 mois. Dans le cadre de l'étude de la réponse anticorps des nouveaux nés nous avons utilisé une **modélisation à effets mixtes** pour prendre en compte l'**aspect longitudinal** des données. Nous allons d'abord faire un bref rappel sur les modèles à effets fixes ou modèles de régression avant d'aborder les modèles à effet mixtes.

On désigne par $Y_{i,t}$ la concentration d'anticorps de l'individu i au temps t. L'un des modèles à effets fixes les plus connus est le modèles linéaire. Supposons que la réponse anticorps de l'enfant ne dépende que du temps par exemple.

Modèle à effets fixes (modèle linéaire simple)

Le modèle linéaire simple stipule qu'il y a une dépendance linéaire entre les individus et le temps, avec un terme d'erreur $\varepsilon_{i,t}$; ce qui se traduit par l'équation suivante :

$$Y_{i,t} = \alpha + \beta.t + \varepsilon_{i,t} \tag{1}$$

$$\varepsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$$

$t = 3, 6, 9, 12, 15, 18$ α, β : effets fixes
$\varepsilon_{i,t}$: terme d'erreur aléatoire
α, β, σ : 3 paramètres à estimer

Graphiquement nous avons tous les individus qui ont le même comportement au cours du temps voir figure 1 située page 2.

Une des limites du modèle à effets fixes est qu'il suppose une indépendance entre les $Y_{i,t}$. Or ici les $Y_{i,t}$ d'un même individu présentent un certain degré de corrélation (ou dépendance). Dans ce cas le modèle à effets fixes n'est pas approprié. Pour prendre en compte la dépendance entre certains individus, on rajoute au modèle à effet fixe un effet aléatoire, le modèle ainsi obtenu est appelé modèle a effets mixtes.

Modèle à effets mixtes

Le modèle linéaire à effets mixtes est une extension du modèle linéaire qui prend en compte la variabilité liée aux individus. Ce modèle est composé d'une partie fixe et d'une partie aléatoire. La partie fixe est identique pour chaque individu et représente l'effet population. La partie aléatoire est propre à chacun des individus et traduit la variabilité liée à chaque sujet.

Pour illustrer les modèles linéaires à effets mixtes nous allons partir du modèle (voir équation (1)). La prise en compte de la variabilité se fait en ajoutant au modèle une constante aléatoire (voir ((2)).

$$Y_{i,t} = (\alpha + \alpha_i) + \beta t + \varepsilon_{i,t} \tag{2}$$

$$\varepsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$$

$$\alpha_i \sim \mathcal{N}(0, \mu^2)$$

α, β :effets fixes
$\varepsilon_{i,t}$: terme d'erreur aléatoire

α_i : effet aléatoire
α, β, σ,μ : paramètres à estimer

Sur ce modèle la constante aléatoire s'interprète comme suit : à $t = 0$ il y a un point de départ moyen de α_i; cependant chaque enfant peut varier de $\alpha_i \sim \mathcal{N}(0, \mu^2)$. Un enfant avec un effet aléatoire positif débutera plus haut que la moyenne et inversement un enfant avec une réponse anticorps négative commence plus bas que la moyenne (voir figure du milieu 1, situé 2).

Le second modèle à titre illustratif est le modèle à pente aléatoire (equation (3))

$$\begin{aligned} Y_{i,t} &= \alpha + (\beta + \beta_i)t + \varepsilon_{i,t} \\ \varepsilon_{i,t} &\sim \mathcal{N}(0, \sigma^2) \\ \beta_i &\sim \mathcal{N}(0, \nu^2) \end{aligned} \tag{3}$$

α, β : effets fixes
$\varepsilon_{i,t}$: terme d'erreur aléatoire
β_i : effets aléatoires
α, β, σ,ν : paramètres à estimer

Sur ce modèle la pente aléatoire nous renseigne sur l'acquisition de la réponse immunitaire au cours du temps. Ainsi il y a une évolution moyenne de$\beta_i \sim \mathcal{N}(0, \nu^2)$ et chaque enfant varie de β_i. Il est possible d'avoir plusieurs effets aléatoires en même temps dans un seul modèle.

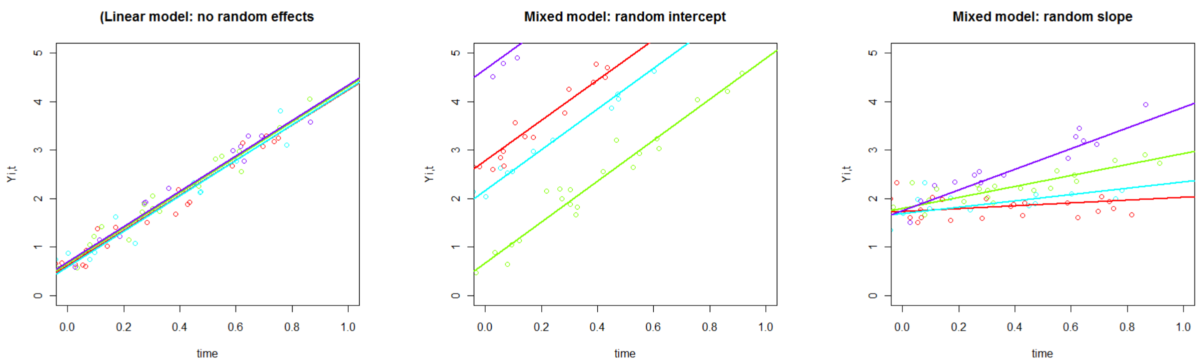


Figure 1: De gauche à droite modèle linéaire, modèle linéaire à effets mixtes avec constante aléatoire, modèle linéaire à effets mixte avec pente aléatoire,

Descriptif des données

Variable réponse

La variable d'intérêt est la concentration d'anticorps aux temps $t = 3, 6, 9, 12, 15, 18$

Covariables

- Concentration d'anticorps dans le cordon
- Concentration d'anticorps circulant dans le placenta
- Apposition placentaire

Remarque: Dans le jeu de données sur lequel nous allons travailler, les temps de mesures sont $t = (-1, 0, 3, 6, 9, 12, 15, 18)$. *Alors que ce que nous cherchons à modéliser est la réponse d'anticorps de l'enfant aux temps* $t = 3, 6, 9, 12, 15, 18$. *De plus la seule covariable présente est ap="apposition placentaire". Il va falloir effectuer quelques transformations pour créer nos covariables "concentration dans le cordon" et "concentration dans le placenta".*

```
> #==========================================================================================
> ##Chemin du repertoire de travail
>  directory<-"~/Formation R/Presentation"
>  ##Nom du fichier de données
>  name<-"immno_data.txt"
>  ##Dans un premier temps on se place dans le répertoire de travail
>  setwd(directory)
> name<-"imuno_data.txt"
>  ##Dans un premier temps on se place dans le répertoire de travail
>  setwd(directory)
>  ##Exportation des données imuno_data.txt
>  data<-read.table(name,header=TRUE)
> ##apperçu des données
>  head(data)
```

```
    id time              conc ap
1 A026   -1  529.470674832787  0
2 A026    0  473.682993027729  0
3 A026    3  125.249998802944  0
4 A026    6  10.5600797819131  0
5 A026    9  1.66330285109751  0
6 A026   12 0.747218505941409  0
```

Transformation des données

```
> #==========================================================================================
>
> ##Voir la structure des variables (numériques, catégorielles,\dots)
> str(data)
```

```
'data.frame':        400 obs. of  4 variables:
 $ id  : Factor w/ 50 levels "A026","A027",..: 1 1 1 1 1 1 1 1 2 2 ...
 $ time: int  -1 0 3 6 9 12 15 18 -1 0 ...
 $ conc: Factor w/ 304 levels "0.394076915563615",..: 252 235 69 49 37 8 304 304 141 112 ...
 $ ap  : int  0 0 0 0 0 0 0 0 0 0 ...
```

Cette commande nous informe sur deux choses:

- **ap** qui est une variable binaire est exportée en numérique
- **conc** qui est une variable continue est exportée en variable factorielle

Nous allons nous intéresser à la variable d'intérêt **ap**; l'instruction suivante la met en variable catégorielle :

```
> #=========================================================================================
> ##ap en variable catégorielle :
> data$ap<-as.factor(data$ap)
> str(data)


'data.frame':        400 obs. of  4 variables:
 $ id  : Factor w/ 50 levels "A026","A027",..: 1 1 1 1 1 1 1 1 2 2 ...
 $ time: int  -1 0 3 6 9 12 15 18 -1 0 ...
 $ conc: Factor w/ 304 levels "0.394076915563615",..: 252 235 69 49 37 8 304 304 141 112 ...
 $ ap  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Nous allons nous intéresser à la variable d'intérêt **conc**. Nous remarquons qu'elle est mise en variable catégorielle, alors qu'il s'agit de données continues. Un aperçu sur les différents niveau de cette variable nous éclaire sur ce sujet.

```
> #========================================================================================
> ##Différents niveaux de la variable "conc"
> levels(data$conc)[250:304]


 [1] "5.90557583766243" "524.56879134341"  "529.470674832787" "534.032718140476"
 [5] "54.5234810135174" "54.6990816966891" "54.8958415579685" "545.457856653432"
 [9] "564.423641247055" "57.0131510676497" "58.0709989904095" "58.4029255317147"
[13] "59.9543023097441" "6.02623604017053" "6.39051063769409" "6.63721811582465"
[17] "6.88858656827532" "61.0274483313252" "651.137968630445" "7.36425716400648"
[21] "7.37498404183813" "7.80068020920365" "7.83376869654816" "7.93844323129843"
[25] "70.3237503128675" "713.775731998662" "72.029399781135"  "739.963666717544"
[29] "7449.51590255745" "765.814482665904" "783.066662188844" "783.615960464655"
[33] "7892.4186256385"  "8.23816268739339" "8.26030781928232" "8.31959101751639"
[37] "8.52546520776916" "8.68710158780084" "806.858292751755" "81.8626583750007"
[41] "82.2175351407928" "8203.89778374936" "850.84696445669"  "864.670622484403"
[45] "864.893247156763" "89.7572912187203" "894.694853740906" "9.11487706404194"
[49] "90.4011576056987" "903.593556498496" "94.9668588347593" "965.146876045932"
[53] "9776.49317702774" "980.892555963136" "Low"


> ##Quelles lignes ont une "conc" qui vaut "Low"
> which(data$conc=="Low")


 [1]   7   8  14  15  21  22  23  24  30  31  32  38  39  40  47  68  75  86  96
[20] 107 110 111 118 119 123 124 125 126 127 128 129 133 134 135 136 141 142 143
[39] 144 150 151 152 155 159 160 165 166 168 173 175 176 179 182 183 184 192 200
[58] 213 214 215 216 229 230 231 238 239 248 254 255 256 263 278 280 288 296 302
[77] 304 320 323 327 328 334 352 354 355 358 359 360 365 366 367 368 376 391


> ##Quelles mesures ont une "conc" qui vaut "Low"
> lines<-which(data$conc=="Low")
> data[lines,]$id
```

```
 [1] A026 A026 A027 A027 A028 A028 A028 A028 A029 A029 A029 A030 A030 A030 A031
[16] A034 A035 A036 A037 A039 A039 A039 A040 A040 A041 A041 A041 A041 A041 A041
[31] A042 A042 A042 A042 A042 A043 A043 A043 A043 A044 A044 A044 A045 A045 A045
[46] A046 A046 A046 A047 A047 A047 A048 A048 A048 A048 A049 A050 A052 A052 A052
[61] A052 A054 A054 A054 A055 A055 C301 C302 C302 C302 C303 C305 C305 C306 C307
[76] C308 C308 C310 C311 C311 C311 C312 C314 C315 C315 C315 C315 C315 C316 C316
[91] C316 C316 C317 C319
50 Levels: A026 A027 A028 A029 A030 A031 A032 A033 A034 A035 A036 A037 ... C320
```

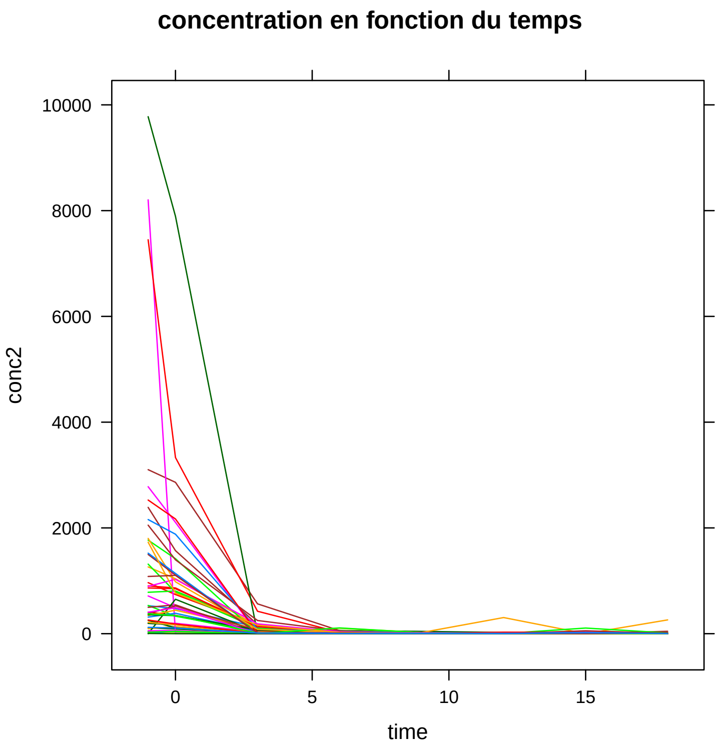
En effet sur ces données nous sommes confrontés au problème de troncatures (valeurs non mesurées en dessous d'un seuil). Nous allons effectuer quelques transformation pour gérer ce problème.

Nous avions déjà en mémoire les lignes ou il y a des troncatures (vecteur lines) Nous allons attribuer une valeur à ces "Low" en tirant un échantillon d'une loi aléatoire comprise entre 0 et le seuil (seuil détecté du laboratoire).

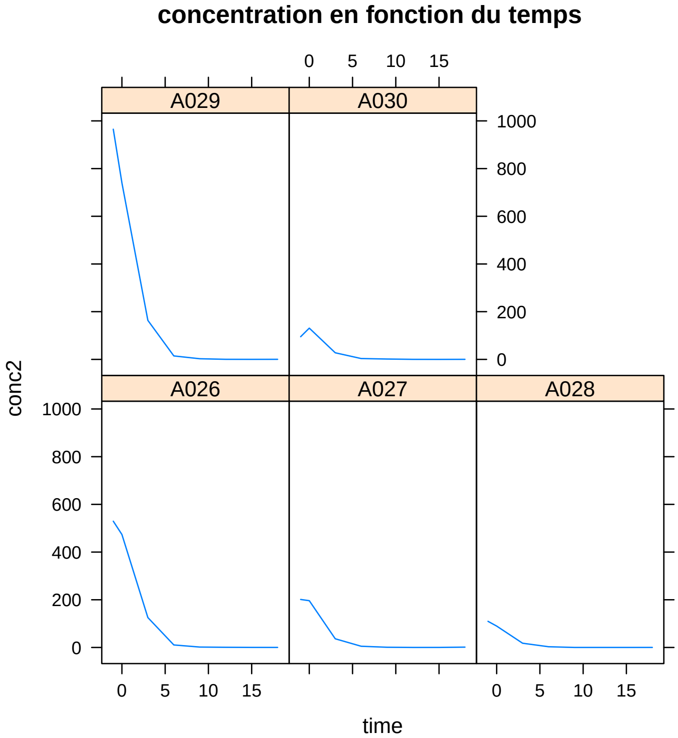
```
> #===========================================================================================
>
>
> ##Ensemble des données sans troncatures
> data_no_low<-data[-lines,]
> head(data_no_low)


    id time              conc ap
1 A026   -1  529.470674832787  0
2 A026    0  473.682993027729  0
3 A026    3  125.249998802944  0
4 A026    6  10.5600797819131  0
5 A026    9  1.66330285109751  0
6 A026   12 0.747218505941409  0


> str(data_no_low)


'data.frame':        306 obs. of  4 variables:
 $ id  : Factor w/ 50 levels "A026","A027",..: 1 1 1 1 1 1 2 2 2 2 ...
 $ time: int  -1 0 3 6 9 12 -1 0 3 6 ...
 $ conc: Factor w/ 304 levels "0.394076915563615",..: 252 235 69 49 37 8 141 112 201 241 ...
 $ ap  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...


> ##Concentration qui sont différentes de "Low"
> conc_no_low<-as.numeric(as.character(data_no_low$conc))
> ##Plus petite valeur parmi les concentration qui sont différentes de "Low"
> seuil=min(conc_no_low)
> data$conc2<-rep(NA,nrow(data))
> ##La concentration pour les valeur non tronquée est la même
> data[-lines,]$conc2<-as.numeric(as.character(data[-lines,]$conc))
> ##La concentration pour les valeur tronquée vaut runif(nrow(data[lines,]),0,seuil) qui est une unifor
> data[lines,]$conc2<-runif(nrow(data[lines,]),0,seuil)
> str(data)


'data.frame':        400 obs. of  5 variables:
 $ id   : Factor w/ 50 levels "A026","A027",..: 1 1 1 1 1 1 1 1 2 2 ...
 $ time : int  -1 0 3 6 9 12 15 18 -1 0 ...
 $ conc : Factor w/ 304 levels "0.394076915563615",..: 252 235 69 49 37 8 304 304 141 112 ...
 $ ap   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ conc2: num  529.47 473.68 125.25 10.56 1.66 ...
```

Une fois ces transformations effectuées, on peut visualiser graphiquement les individus

```
> #=================================================================================
> library(lattice)
> ##Visualisation graphique de tous les individus
> plot=xyplot(conc2~time,data=data,type='l',groups=id,main="concentration en fonction du temps")
> print(plot)
```

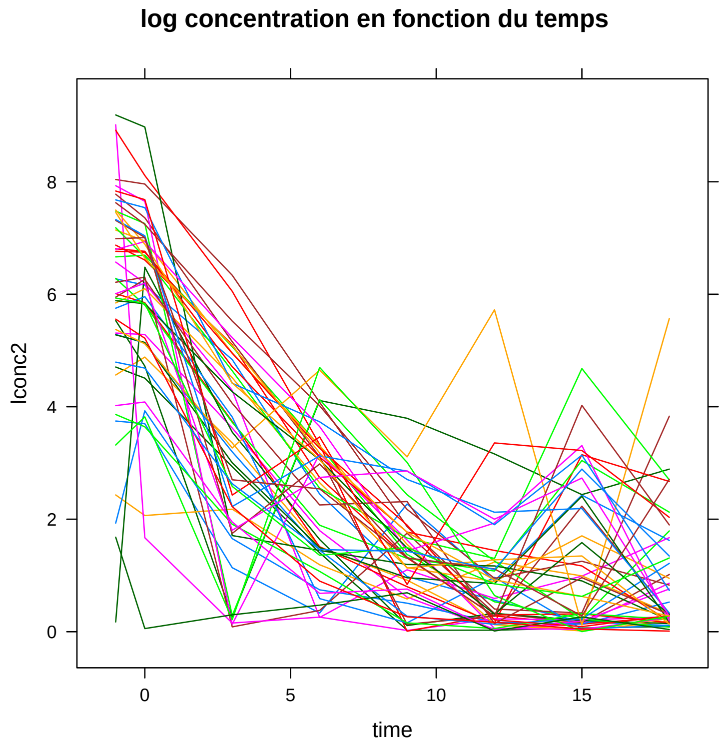



```
> #=================================================================================
> ##Visualisation graphique individu par individu (les 5 premiers)
> plot0=xyplot(conc2~time|id,data=data[1:40,],type='l',main="concentration en fonction du temps")
> print(plot0)
```



Afin de "zoomer" sur ce qui se passe de $t = 3$ à $t = 18$, nous allons procéder à une transformation logarithmique.

```
> #==========================================================================================
> ##Transformation de la variable conc2 en lconc2=log(conc2)
> data$lconc2<-log(1+data$conc2)
> ##Visualisation graphique
> plot2=xyplot(lconc2~time,data=data,type='l',groups=id,main="log concentration en fonction du temps")
> print(plot2)
```



Nous allons créer les covariables additionnelles correspondant à la concentration d'anticorps dans le cordon $(t = 0)$ et à la concentration d'anticorps séquestrés dans le placenta $(t = -1)$.

```
> #==========================================================================================
>
> ##Concentration d'anticorps circulant dans placenta (correspond à t=-1)
> data_time1<-data[data$time==-1,]$lconc2
> ##Concentration d'anticorps dans le cordon (correspond à t=0)
> data_time0<-data[data$time==0,]$lconc2
> ##Répétition de chacun des éléments
> conc_co<-rep(data_time1,each=6)
> conc_ci<-rep(data_time0,each=6)
> ###Nous travaillerons maintenant avec data2 qui est notre jeu de données à partir de $t=3$
> ##Extraction des temps supérieurs à 0
> data2<-data[data$time>0,]
> ##Rajout des covariables
> data2$conc_Co<-conc_co
> data2$conc_Ci<-conc_ci
> ##Aperçu de data2
> head(data2)
```

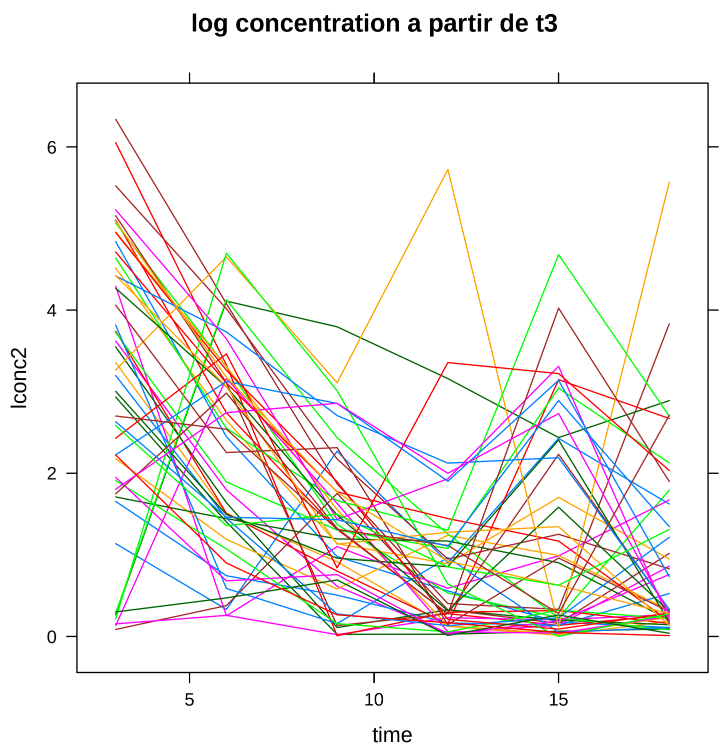
```
    id time              conc ap       conc2    lconc2  conc_Co  conc_Ci
3 A026    3  125.249998802944  0 125.2499988 4.8382641 6.273765 6.162647
4 A026    6  10.5600797819131  0  10.5600798 2.4475578 6.273765 6.162647
```

```
5 A026    9  1.66330285109751  0   1.6633029 0.9795670 6.273765 6.162647
6 A026   12 0.747218505941409  0   0.7472185 0.5580251 6.273765 6.162647
7 A026   15               Low  0   0.1922318 0.1758270 6.273765 6.162647
8 A026   18               Low  0   0.1158113 0.1095818 6.273765 6.162647
```

```
> str(data2)
```

```
'data.frame':        300 obs. of  8 variables:
 $ id      : Factor w/ 50 levels "A026","A027",..: 1 1 1 1 1 1 2 2 2 2 ...
 $ time    : int  3 6 9 12 15 18 3 6 9 12 ...
 $ conc    : Factor w/ 304 levels "0.394076915563615",..: 69 49 37 8 304 304 201 241 12 304 ...
 $ ap      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ conc2   : num  125.25 10.56 1.663 0.747 0.192 ...
 $ lconc2 : num  4.838 2.448 0.98 0.558 0.176 ...
 $ conc_Co: num  6.27 6.27 6.27 6.27 6.27 ...
 $ conc_Ci: num  6.16 6.16 6.16 6.16 6.16 ...
```

```
> #=========================================================================================
> ##Transformation de la variable conc2 en lconc2=log(conc2)
> plot3=xyplot(lconc2~time,data=data2,type='l',groups=id,main="log concentration a partir de t3")
> print(plot3)
```



Une fois notre jeu de donnée complet et transformé, nous pouvons passer à la modélisation.

- Notre nouveau jeu de donnée est **data2**.
- La nouvelle variable d'interet : **lconc2**
- Covariables: $ap, concCo, concCi$

Modélisation

- Package à télécharger: lme4
- **Syntaxe**: lmer(somme des covariables fixes+(somme des covariables aléatoires | Identifiant))

Modèle 1: modèle avec "intercept" aléatoire

```
> #===============================================================================
> ##
> library(lme4)
> mm1<-lmer(lconc2~ap+conc_Co+conc_Ci+(1|id),data=data2)
> summary(mm1)


Linear mixed model fit by REML
Formula: lconc2 ~ ap + conc_Co + conc_Ci + (1 | id)
   Data: data2
  AIC  BIC logLik deviance REMLdev
 1097 1120 -542.7     1074    1085
Random effects:
 Groups   Name        Variance   Std.Dev.
 id       (Intercept) 1.0553e-09 3.2485e-05
 Residual             2.1277e+00 1.4587e+00
Number of obs: 300, groups: id, 50

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.30061    0.32216   0.933
ap1         -0.04647    0.29829  -0.156
conc_Co      0.04231    0.06436   0.658
conc_Ci      0.18557    0.07139   2.599

Correlation of Fixed Effects:
        (Intr) ap1    conc_Co
ap1     -0.109
conc_Co -0.255 -0.317
conc_Ci -0.475  0.307 -0.707


> ##Parties fixes du modèle
> fixef(mm1)


(Intercept)         ap1     conc_Co     conc_Ci
 0.30060893 -0.04647055  0.04231183  0.18556803


> ##Effets aléatoires du modèle
> ranef(mm1)


$id
       (Intercept)
A026 -3.316587e-09
A027 -5.647864e-09
A028 -1.001736e-08
A029 -2.285531e-09
A030 -6.611474e-09
```

```
A031  3.896945e-09
A032  1.276932e-08
A033  6.068174e-09
A034 -3.460576e-09
A035  2.364743e-08
A036  6.292442e-09
A037  5.155111e-09
A038  9.542062e-09
A039 -1.927960e-08
A040 -8.460035e-09
A041 -1.309138e-08
A042 -8.934435e-09
A043 -8.474245e-09
A044 -3.279843e-09
A045 -9.721935e-09
A046 -2.124879e-09
A047 -1.214079e-08
A048 -2.038816e-08
A049 -2.165526e-08
A050  3.824268e-09
A051  5.106773e-09
A052 -8.950599e-09
A053  8.486361e-09
A054 -6.818422e-09
A055 -8.679380e-09
C301  7.212596e-11
C302 -9.704503e-10
C303  3.916863e-08
C304  1.872447e-08
C305 -7.164003e-10
C306  9.679569e-09
C307  1.296227e-08
C308  3.549275e-09
C309  8.652207e-09
C310 -9.433036e-12
C311  5.617898e-09
C312  2.270607e-09
C313  7.163342e-09
C314  6.307465e-09
C315 -1.479897e-09
C316 -1.560930e-08
C317  2.214090e-09
C318 -4.556955e-09
C319 -2.069953e-09
C320  7.579916e-09
```

Modèle 2: modèle avec "pente" aléatoire

```
> #===============================================================================
> ##
> mm2<-lmer(lconc2~ap+conc_Co+conc_Ci+(1+time|id),data=data2)
> summary(mm2)
```

```
Linear mixed model fit by REML
Formula: lconc2 ~ ap + conc_Co + conc_Ci + (1 + time | id)
   Data: data2
  AIC  BIC logLik deviance REMLdev
 1043 1073 -513.6     1016    1027
```

```
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 id       (Intercept) 2.497264 1.58027
          time        0.026215 0.16191  -0.955
 Residual             1.208885 1.09949
Number of obs: 300, groups: id, 50

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.28412    0.35440   0.802
ap1         -0.04851    0.32813  -0.148
conc_Co      0.03802    0.07080   0.537
conc_Ci      0.21525    0.07853   2.741

Correlation of Fixed Effects:
        (Intr) ap1    conc_Co
ap1     -0.109
conc_Co -0.255 -0.317
conc_Ci -0.475  0.307 -0.707
```

Modèle 2: modèle avec intercept aléatoire "pente" aléatoire

```
> #===========================================================================
> ##
> mm3<-lmer(lconc2~ap+conc_Co+conc_Ci+(time|id),data=data2)
> summary(mm2)

Linear mixed model fit by REML
Formula: lconc2 ~ ap + conc_Co + conc_Ci + (1 + time | id)
   Data: data2
  AIC  BIC logLik deviance REMLdev
 1043 1073 -513.6     1016    1027
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 id       (Intercept) 2.497264 1.58027
          time        0.026215 0.16191  -0.955
 Residual             1.208885 1.09949
Number of obs: 300, groups: id, 50

Fixed effects:
            Estimate Std. Error t value
(Intercept)  0.28412    0.35440   0.802
ap1         -0.04851    0.32813  -0.148
conc_Co      0.03802    0.07080   0.537
conc_Ci      0.21525    0.07853   2.741

Correlation of Fixed Effects:
        (Intr) ap1    conc_Co
ap1     -0.109
conc_Co -0.255 -0.317
conc_Ci -0.475  0.307 -0.707
```

Sélection de modèle

```
> #===========================================================================
> a1<-anova(mm1,mm2)
> a2<-anova(mm2,mm3)
> a3<-anova(mm1,mm3)
```