

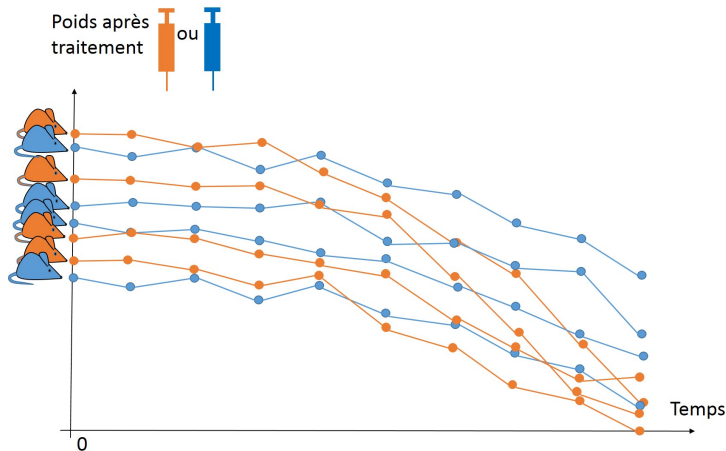
Analyse de données longitudinales ou mesures répétées dans le temps - diverses approches plus ou moins adaptées suivant les données et les objectifs de l'analyse

M.L. Delignette-Muller
VetAgro Sup - LBBE

26 janvier 2023

Cadre d'étude

Ex. mesure d'une variable quantitative continue au cours du temps sur des animaux soumis à différents traitements : effet du traitement sur la réponse au cours du temps ?



Jeu de données exemple n°1 - souris avec ou sans induction de colite par le DSS (Dextran Sodium Sulfate)

```
d1 <- read.table("DATA/DSS.txt", header = TRUE,
                 stringsAsFactors = TRUE)

str(d1)

## 'data.frame': 192 obs. of 4 variables:
## $ animal      : Factor w/ 24 levels "A1","A10","A11",...: 1 1 1 1 1 1 1
## $ traitement: Factor w/ 2 levels "avec DSS","sans DSS": 2 2 2 2 2
## $ temps       : int  0 1 2 3 4 5 6 7 0 1 ...
## $ poids       : num  25.3 25.4 24.9 25.5 25.9 ...

# Changement de l'ordre des modalités du facteur
# traitement
d1$traitement <- factor(d1$traitement,
                       levels = c("sans DSS", "avec DSS"))
```

Jeu de données exemple n°2 - souris sous DSS traitées ou non par la buprénorphine (un morphinique)

```
d2 <- read.table("DATA/buprenorphine.txt", header = TRUE,
                 stringsAsFactors = TRUE)

str(d2)

## 'data.frame': 192 obs. of 4 variables:
## $ animal      : Factor w/ 24 levels "B1","B10","B11",...: 1 1 1 1 1 1
## $ traitement: Factor w/ 2 levels "avec bupre","sans bupre": 2 2 2
## $ temps       : int  0 1 2 3 4 5 6 7 0 1 ...
## $ poids       : num  26.2 26.1 26.4 25 26 ...

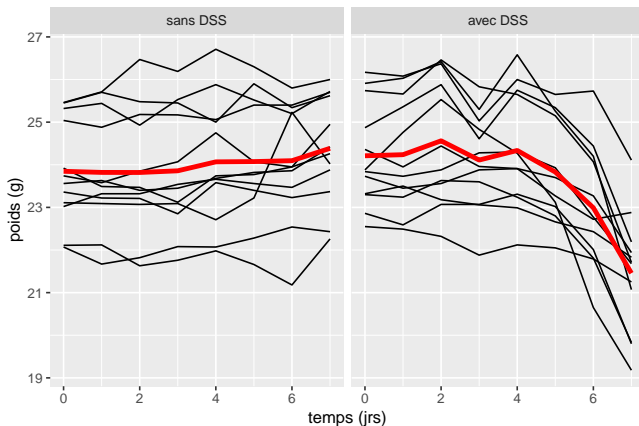
# Changement de l'ordre des modalités du facteur
# traitement
d2$traitement <- factor(d2$traitement,
                      levels = c("sans bupre", "avec bupre"))
```

Création d'une fonction utilisant le package `ggplot2` pour visualiser les données

```
require(ggplot2)
plotd <- function(d)
{
  ggplot(d, aes(x = temps, y = poids, group = animal)) +
    facet_wrap(~ traitement) + geom_line() +
    xlab("temps (jrs)") + ylab("poids (g)") +
    stat_summary(aes(group = traitement), geom = "line",
                 lwd = 1.5, fun = mean, col = "red")
}
```

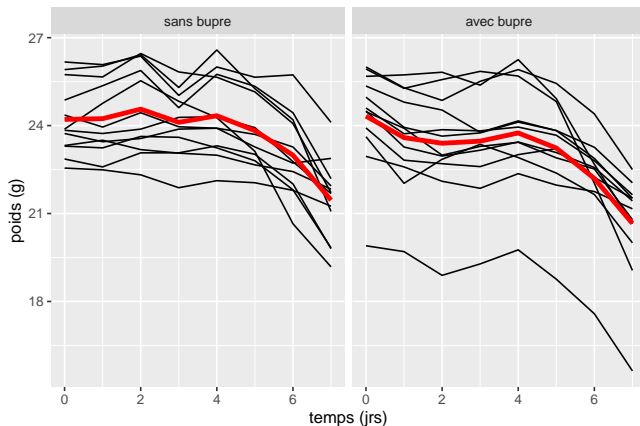
Visualisation du jeu de données exemple n°1 - DSS

```
plotd(d1)
```



Visualisation du jeu de données exemple n°2 - buprénorphine

```
plotd(d2)
```



Diversité des démarches utilisées

Consigne 1

Prenez connaissance des données à l'aide du script fourni et énoncez les méthodes que vous avez déjà vues utilisées sur ce type de données où les méthodes auxquelles vous pensez.

Démarches basiques explorées

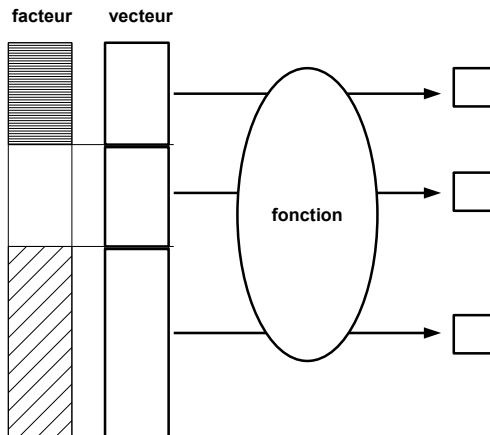
Parmi les démarches proposées nous explorerons dans un premier temps les démarches les plus basiques en réfléchissant en particulier à leurs limites.

Commençons par la **comparaison à chaque temps des réponses entre les deux groupes**.

Afin de pouvoir réaliser ce genre d'analyse efficacement, voyons tout d'abord **comment automatiser les analyses répétées**.

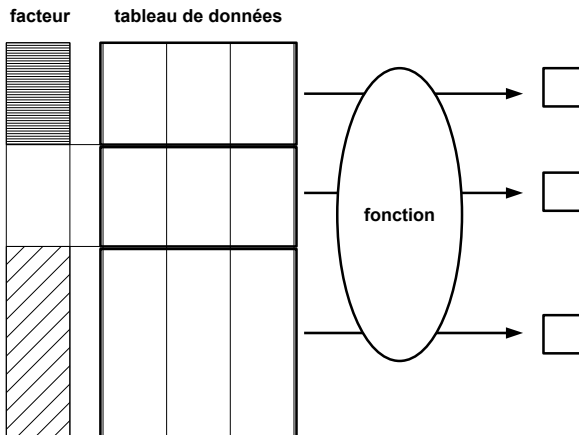
Rappel du principe de la fonction `tapply`

```
tapply(vecteur, facteur, fonction)
```



Principe de la fonction `by`

```
by(tableau_de_donnees, facteur, fonction)
```



Création d'une fonction de comparaison des deux groupes à un temps donné (test T)

Fonction qui fait le test sur un sous-jeu de données à un temps donné et renvoie la valeur de p associé

```
test1tempsv1 <- function(dltemps)
{
  test <- t.test(dltemps$poids ~ dltemps$traitement,
                 paired = FALSE, var.equal = FALSE)
  # Mieux vaut ne pas supposer a priori les variances égales
  # si on ne se donne pas la peine de le vérifier à chaque temps.
  # On pourrait ajouter dans la fonction un graphe
  # à chaque temps pour vérifier normalité et égalité des var.
  #
  return(test$p.value)
}
## Essai au temps 1 pour voir si ça fonctionne
test1tempsv1(subset(dl, temps == 0))

## [1] 0.47
```

Test T à chaque temps sur le jeu de données 1 - DSS

```
by(d1, d1$temps, test1tempv1)
```

```
## d1$temps: 0
```

```
## [1] 0.47
```

```
## -----
```

```
## d1$temps: 1
```

```
## [1] 0.45
```

```
## -----
```

```
## d1$temps: 2
```

```
## [1] 0.23
```

```
## -----
```

```
## d1$temps: 3
```

```
## [1] 0.63
```

```
## -----
```

```
## d1$temps: 4
```

```
## [1] 0.65
```

```
## -----
```

```
## d1$temps: 5
```

```
## [1] 0.66
```

```
## -----
```

```
## d1$temps: 6
```

```
## [1] 0.065
```

```
##
```

Création d'une seconde fonction pour rendre les résultats plus lisibles

Fonction qui fait le test à chaque temps et renvoie les valeurs de p associées

```
testachqtemps <- function(d)
{
  pval <- as.vector(by(d, d$temps, testltempv1))
  names(pval) <- unique(d$temps)
  return(pval)
}
## application au jeu de données dl
testachqtemps(dl)
```

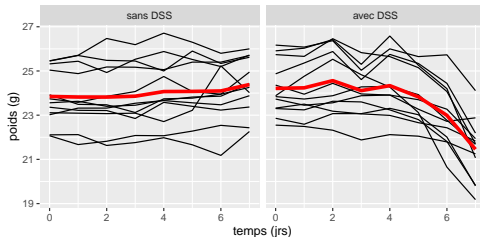
##	0	1	2	3	4	5	6
##	0.47215	0.44744	0.22959	0.63394	0.64886	0.65671	0.06537
##	7						
##	0.00002						

Application de l'approche aux deux jeux de données

Consigne 2

A l'aide du script fourni appliquez la première approche aux deux jeux de données, tentez d'en interpréter les résultats, discutez des limites de cette approche et proposez une amélioration.

Comparaison à chaque temps sur le jeu de données 1

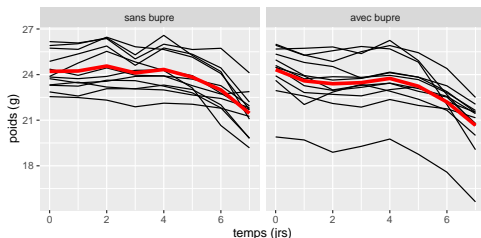


```
testachqtemps (d1)
```

```
##           0           1           2           3           4           5           6
## 0.47215 0.44744 0.22959 0.63394 0.64886 0.65671 0.06537
##          7
## 0.00002
```

Différence significative entre les deux groupes à 7 jours
(pourtant il semble que les réponses diffèrent dès J5 - J6)

Comparaison à chaque temps sur le jeu de données 2



```
testachqttemps (d2)
```

```
##      0      1      2      3      4      5      6      7
## 0.85 0.30 0.10 0.30 0.37 0.35 0.22 0.24
```

Aucune différence significative observée (pourtant on a l'impression d'en voir une au moins entre J0 et J1)

Inconvénients de l'approche

- Problème des tests répétés (faudrait-il faire une correction de type Bonferroni-Holm ?).
- En même temps perte de puissance du fait du découpage du jeu de données.
- Conditions d'utilisation des tests pas vérifiables globalement
- Problème d'interprétation (ce n'est pas parce qu'à un temps l'effet n'est pas significatif encore qu'il n'y a pas d'effet).
- Et perte de puissance du fait de la non prise en compte de l'effet animal (dépendance des réponses aux différents temps).

Tentons d'améliorer l'approche en travaillant non pas sur la réponse brute, mais sur la différence : poids au jour J - poids à J0.

Comparaison à chaque temps des différences au contrôle

Pour cela voici une fonction modifie le jeu de données en remplaçant le poids à J par poids à J - poids à J0.

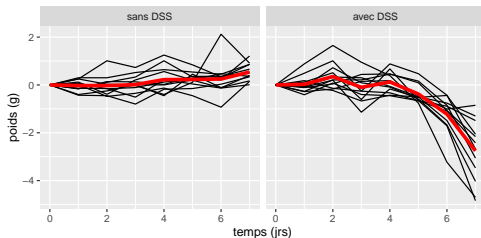
```
modifyd <- function(d)
{
  db <- d
  for (i in 1:nrow(d))
  {
    anim <- d$animal[i]
    pds.anim.t0 <- subset(d, (animal == anim) &
                          (temps == 0))$poids
    db$poids[i] <- d$poids[i] - pds.anim.t0
  }
  return(db)
}
```

Consigne 3

Utilisez cette procédure pour réanalyser les jeux de données 1 et 2.

Analyse du jeu de données 1 modifié

```
d1b <- modifyd(d1); plotd(d1b); testachqtemps(d1b)
```

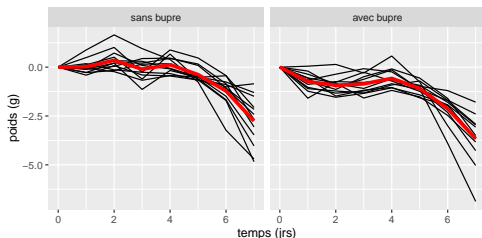


```
##      0      1      2      3      4      5      6
##      NaN 6.8e-01 7.0e-02 5.9e-01 6.4e-01 4.9e-04 6.6e-05
##      7
## 1.5e-06
```

Différence significative observée à 5, 6 et 7 jours.

Analyse du jeu de données 2 modifié

```
d2b <- modifyd(d2); plotd(d2b); testachqtemps(d2b)
```



```
##           0           1           2           3           4           5           6
##      NaN  1.9e-04  4.6e-06  2.3e-03  1.7e-03  8.7e-05  6.6e-03
##           7
##  1.0e-01
```

Différence significative observée à 1, 2, 3, 4, 5, 6 jours mais plus à 7 jours : difficile à interpréter !

Avantages et inconvénients de l'approche comparativement à la précédente

- On a gagné en puissance et on montre des effets non mis en évidence auparavant du fait de la variabilité entre les poids initiaux.
- Mais les problèmes liés aux tests répétés et au découpage des données ne sont pas réglés.
- On a toujours des problèmes d'interprétation : ici en particulier sur le jeu de données 2 on se rend bien compte que la différence aux temps 1 à 6 est due au moins en partie à la différence à J1 et on ne peut plus rien interpréter après J1 (on pourrait aussi calculer les différences $J2-J1$, $J3-J2$, ...).

Comparaison à chaque temps des différences au temps précédent

Pour cela voici une fonction modifie le jeu de données en remplaçant le poids à J_i par le poids à J_i - le poids à J_{i-1} .

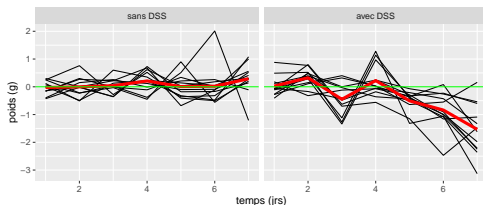
```
require(dplyr)
modifydbis <- function(d)
{
  dsort <- arrange(d, temps)
  animu <- unique(d$animal)
  db <- subset(dsort, temps != 0)
  for (j in 1:length(animu))
  {
    animj <- animu[j]
    dsortanimj <- subset(dsort, animal == animj)
    db[db$animal == animj, ]$poids <- diff(dsortanimj$poids)
  }
  return(db)
}
# ATTENTION, type de programmation qui devient un peu compliqué
# et dont le bon fonctionnement est à vérifier ligne à ligne !!!
```

Analyse du jeu de données 1 sur différences $J_i - J_{i-1}$

```
d1c <- modifydbis(d1); testachqtemps(d1c)

##           1           2           3           4           5           6
## 0.682935 0.030276 0.019348 0.942644 0.005581 0.004638
##           7
## 0.000015

plotd(d1c) + geom_hline(yintercept = 0, color = "green")
```



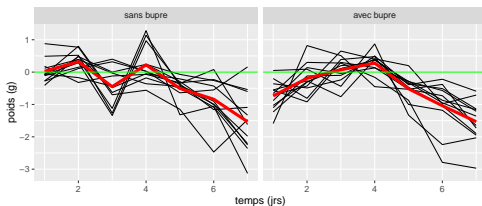
Différence significative observée à 5, 6 et 7 jours (même résultat que précédemment) mais aussi à 2 et 3 jours.

Analyse du jeu de données 2 sur différences $J_i - J_{i-1}$

```
d2c <- modifydbis(d2); testachqtemps(d2c)

##           1           2           3           4           5           6           7
## 0.00019 0.00652 0.02027 0.78348 0.95797 0.49573 0.99597

plotd(d2c) + geom_hline(yintercept = 0, color = "green");
```



Différence significative observée à 1, 2 et 3 jours mais plus après (même résultat que précédemment).

Comparaison des différences $J_{fin} - J_0$

Il s'agit d'une méthode très simple qui revient aussi à la comparaison des sommes des différences entre poids à J_i - poids à J_{i-1}

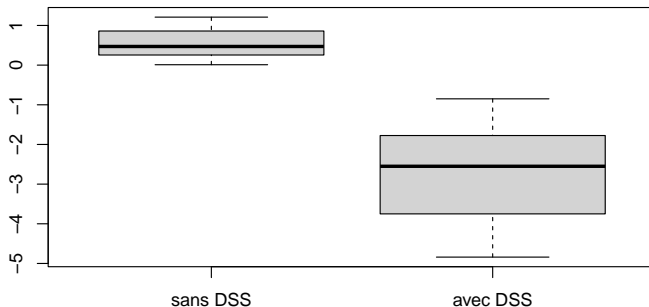
```
# Pour créer un jeu de données avec les différences Jfin - J0
calcJfinJ0 <- function(d)
{
  dsort <- arrange(d, animal)
  dsortJfin <- subset(dsort, temps == max(d$temps))
  dsortJ0 <- subset(dsort, temps == 0)
  # Création d'un petit jeu de données avec les infos utiles
  dJfinJ0 <- data.frame(
    deltapoids = dsortJfin$poids - dsortJ0$poids,
    animal = dsortJ0$animal, traitement = dsortJ0$traitement)
  return(dJfinJ0)
}
```

Consigne 4

Implémentez cette procédure pour réanalyser les jeux de données 1 et 2.

Calcul des différences $J_{fin} - J_0$ - jeu de données 1

```
ddelta1 <- calcJfinJ0(d1)
par(mar = c(3,2,0.1,0.1))
boxplot(deltapoids ~ traitement, data = ddelta1)
```



Test de comparaison des moyennes des différences

$J_{fin} - J_0$ sur le jeu de données 1

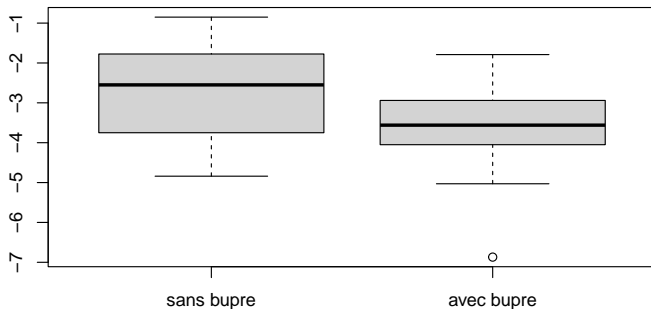
```
t.test(deltapoids ~ traitement, data = ddelta1,
       paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  deltapoids by traitement
## t = 8, df = 13, p-value = 1e-06
## alternative hypothesis: true difference in means between group sans
## 95 percent confidence interval:
##  2.4 4.1
## sample estimates:
## mean in group sans DSS mean in group avec DSS
##                0.54                -2.76
```

On conclut à une perte de poids globale moyenne significativement plus importante avec le DSS.

Calcul des différences $J_{fin} - J_0$ - jeu de données 2

```
ddelta2 <- calcJfinJ0(d2)
par(mar = c(3,2,0.1,0.1))
boxplot(deltapoids ~ traitement, data = ddelta2)
```



Test de comparaison des moyennes des différences

$J_{fin} - J_0$ sur le jeu de données 2

```
t.test(deltapoids ~ traitement, data = ddelta2,
       paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  deltapoids by traitement
## t = 2, df = 22, p-value = 0.1
## alternative hypothesis: true difference in means between group sans
## 95 percent confidence interval:
##  -0.2  2.0
## sample estimates:
## mean in group sans bupre mean in group avec bupre
##                -2.8                -3.7
```

On ne conclut pas à une perte de poids globale moyenne significativement plus importante avec la buprénorphine.

Conclusion sur l'approche travaillant sur la perte globale de poids

- On s'est débarrassé des problèmes liés aux tests répétés et au découpage des données,
- mais **on a perdu beaucoup en terme de description de l'impact du traitement sur la réponse au cours du temps.**
- On n'utilise au final que les mesures initiales et finales (à quoi nous aurons servi les autres ?).

Conclusion sur les approches basiques explorées

Aucune des approches basiques explorées ne semble pouvoir répondre de façon satisfaisante à la problématique.

Il semble important

- de prendre en compte l'**effet aléatoire animal** (la dépendance des réponses au cours du temps),
- de faire une **analyse globale** des données,
- et de se poser la question de ce qui nous intéresse de comparer d'un **point de vue biologique**.

Pour tenter de répondre aux deux premiers objectifs, tentons d'explorer les approches possibles utilisant un modèle linéaire mixte.

Modèle linéaire mixte en considérant le temps comme une variable quantitative

Consigne 5

Construisez un modèle linéaire mixte prenant en compte l'effet aléatoire animal et les effets fixes traitement et temps en considérant le **temps comme quantitatif**.

Choisissez les termes que vous mettez dans le modèle selon des arguments biologiques (choix des interactions notamment), puis interprétez les résultats et discutez des limites de l'approche.

Regardez en particulier le graphe des résidus en fonction du temps.

A faire uniquement sur le jeu de données 2.

Construction du modèle pour le jeu de données 2

On n'a aucune raison de mettre un effet du traitement sur la constante (pas d'effet à J0) mais il est obligatoire de mettre une interaction traitement :temps.

```
require(lme4)
m2 <- lmer(poids ~ temps + traitement:temps + (1| animal),
           data = d2)
summary(m2) $coefficients
```

##	Estimate	Std. Error	t value
## (Intercept)	24.653	0.304	81.1
## temps	-0.321	0.039	-8.3
## temps:traitementavec bupre	-0.076	0.053	-1.4

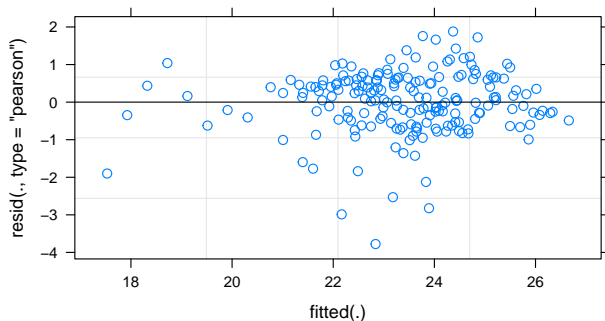
```
confint(m2)
```

##	2.5 %	97.5 %
## .sig01	1.01	1.853
## .sigma	0.80	0.995
## (Intercept)	24.05	25.255
## temps	-0.40	-0.245
## temps:traitementavec bupre	-0.18	0.028

On met en évidence un effet significatif du temps uniquement.

Diagnostics du modèle pour le jeu de données 2 (1)

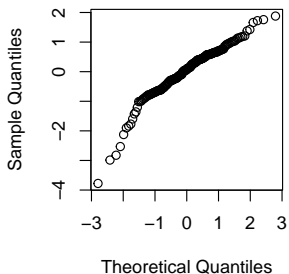
```
plot(m2)
```



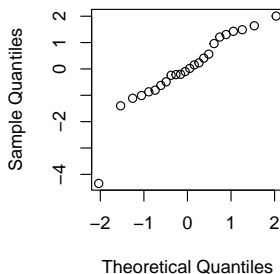
Diagnostics du modèle pour le jeu de données 2 (2)

```
par(mfrow = c(1,2))  
qqnorm(residuals(m2))  
qqnorm(ranef(m2)$animal[,1])
```

Normal Q-Q Plot



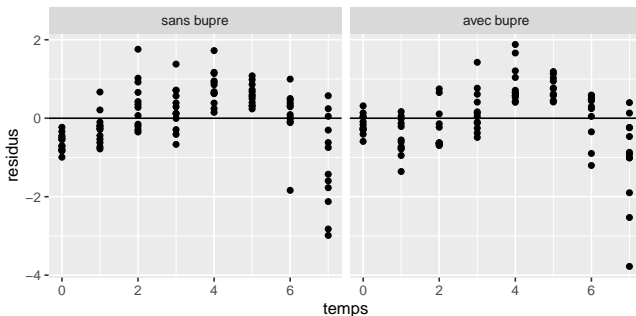
Normal Q-Q Plot



Un animal extrême (cf. QQ plot des effets aléatoires).

Diagnostics du modèle pour le jeu de données 2 (3)

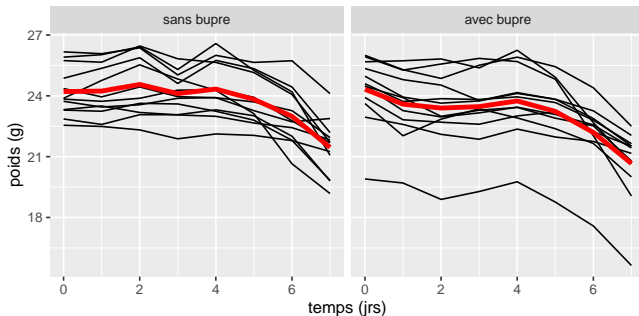
```
d2$residus <- residuals(m2)
ggplot(d2, aes(x = temps, y = residus)) + facet_wrap(~ traitement) +
  geom_point() + geom_hline(yintercept = 0)
```



Tendance nette des résidus pour les deux traitements (due à non linéarité) et hétéroscédasticité à la fin.

Souci majeur de l'approche

Lorsque le temps est considéré comme une variable quantitative continue dans un tel modèle, la relation entre la réponse et le temps est supposée linéaire ce qui n'est pas le cas ici (et est rarement le cas dans la pratique).



Modèle linéaire mixte en considérant le temps comme une variable qualitative

Consigne 6

Construisez un modèle linéaire mixte prenant en compte l'effet aléatoire animal et les effets fixes traitement et temps en considérant le **temps comme qualitatif**.

Choisissez les termes que vous mettez dans le modèle selon des arguments biologiques (choix des interactions notamment), puis interprétez les résultats et discutez des limites de l'approche sur le jeu de données 2.

Construction du modèle pour le jeu de données 2

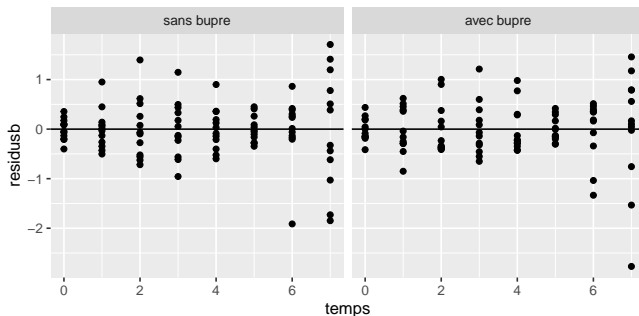
```
d2$ftemps <- as.factor(d2$temps)
levels(d2$ftemps)

## [1] "0" "1" "2" "3" "4" "5" "6" "7"

m2b <- lmer(poids ~ ftemps + traitement:ftemps + (1| animal),
             data = d2)
```


Diagnostics du modèle pour le jeu de données 2

```
d2$residusb <- residuals(m2b)
ggplot(d2, aes(x = temps, y = residusb)) + facet_wrap(~ traitement) +
  geom_point() + geom_hline(yintercept = 0)
```



On a moins de souci sur les résidus : il ne reste qu'une hétéroscédasticité (résidus plus élevés à J6 et J7).

Interprétation du modèle pour le jeu de données 2

Mais le modèle devient trop complexe pour interpréter facilement ses paramètres !

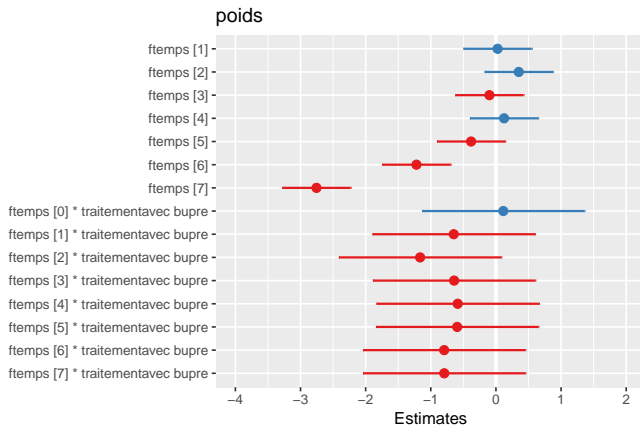
```
summary(m2b)$coefficients
```

	Estimate	Std. Error	t value
## (Intercept)	24.211	0.45	54.207
## ftemps1	0.026	0.27	0.097
## ftemps2	0.351	0.27	1.322
## ftemps3	-0.100	0.27	-0.377
## ftemps4	0.125	0.27	0.471
## ftemps5	-0.382	0.27	-1.438
## ftemps6	-1.220	0.27	-4.596
## ftemps7	-2.755	0.27	-10.379
## ftemps0:traitementavec bupre	0.113	0.63	0.179
## ftemps1:traitementavec bupre	-0.648	0.63	-1.025
## ftemps2:traitementavec bupre	-1.164	0.63	-1.843
## ftemps3:traitementavec bupre	-0.642	0.63	-1.016
## ftemps4:traitementavec bupre	-0.587	0.63	-0.929
## ftemps5:traitementavec bupre	-0.595	0.63	-0.942
## ftemps6:traitementavec bupre	-0.795	0.63	-1.259
## ftemps7:traitementavec bupre	-0.793	0.63	-1.256

Essai d'interprétation avec le package `sjPlot`

Même avec le graphe des effets estimés, on n'y voit pas bien clair !

```
require(sjPlot)
plot_model(m2b)
```

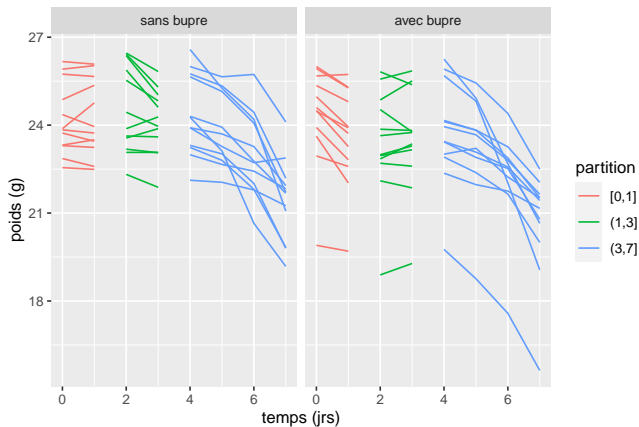


Analyse de deux sous-jeux de données

Travail sur deux sous-jeux de données sur chacun desquels le modèle linéaire mixte peut s'appliquer :

- le sous jeu de données à J0 et J1 avec le temps considéré comme qualitatif à deux modalités,
- et le sous jeu de données de J4 à J7 avec le temps considéré comme quantitatif - il semble raisonnable de supposer la relation linéaire à partir de J4 mais il faudra penser à ajouter un effet traitement sur le terme constant du fait qu'on ne part plus à J0.

Visualisation du découpage des données



Deux modèles linéaires mixtes après découpage des données

Consigne 7

Construisez un modèle linéaire mixte sur chaque sous jeu de données (J0-J1 et J4 à J7) prenant en compte l'effet aléatoire animal et les effets fixes traitement et temps en considérant le **temps comme quantitatif**.

Choisissez les termes que vous mettez dans le modèle selon des arguments biologiques (choix des interactions notamment), puis interprétez les résultats et discutez des limites de l'approche sur le jeu de données 2.

Modèle pour le jeu de données 2 J0-J1

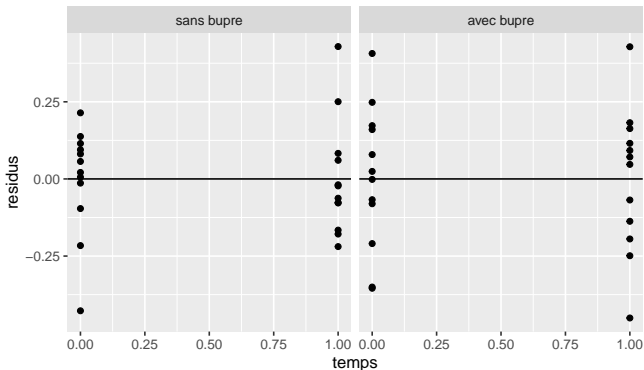
```
d2.01 <- subset(d2, temps <= 1)
m2.01 <- lmer(poids ~ temps + traitement:temps + (1| animal),
              data = d2.01)
summary(m2.01)$coefficients
```

##	Estimate	Std. Error	t value
## (Intercept)	24.267	0.30	81.8
## temps	0.024	0.12	0.2
## temps:traitementavec bupre	-0.756	0.17	-4.5

On met en évidence une interaction significative entre le traitement et le temps : perte de poids au bout d'un jour avec la buprénorphine (0.756 g en moyenne de moins que sans buprénorphine).

Diagnostics du modèle pour le jeu de données 2 J0-J1

```
d2.01$residus <- residuals(m2.01)
ggplot(d2.01, aes(x = temps, y = residus)) +
  facet_wrap(~ traitement) + geom_point() + geom_hline(yintercept = 0)
```



Modèle pour le jeu de données 2 J0-J1 avec le temps en variable qualitative

```
m2.01b <- lmer(poids ~ ftemps + traitement:ftemps + (1| animal),  
               data = d2.01)  
summary(m2.01b)$coefficients
```

##	Estimate	Std. Error	t value
## (Intercept)	24.211	0.43	56.53
## ftemps1	0.026	0.12	0.22
## ftemps0:traitementavec bupre	0.113	0.61	0.19
## ftemps1:traitementavec bupre	-0.648	0.61	-1.07

Cela ajoute un coefficient au modèle et il n'apparaît alors plus de terme significatif.

Modèle pour le jeu de données 2 après J4

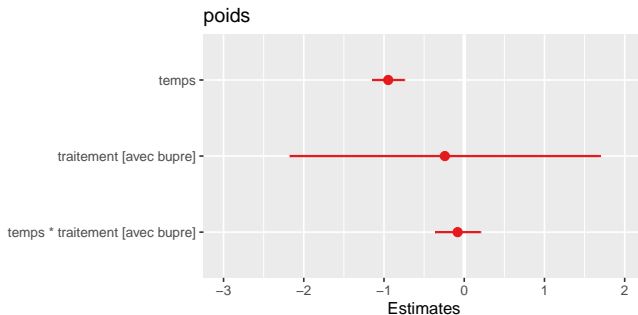
On pense à ajouter l'effet direct du traitement sur la constante.

```
d2.ap4 <- subset(d2, temps >= 4)
m2.ap4 <- lmer(poids ~ temps + traitement + traitement:temps +
              (1| animal), data = d2.ap4)
summary(m2.ap4)$coefficients
```

##	Estimate	Std. Error	t value
## (Intercept)	28.366	0.69	41.23
## temps	-0.948	0.10	-9.45
## traitementavec bupre	-0.242	0.97	-0.25
## temps:traitementavec bupre	-0.082	0.14	-0.58

Modèle pour le jeu de données 2 après J4

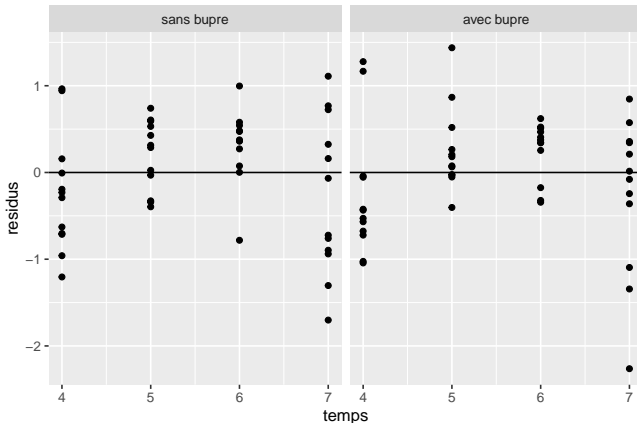
```
plot_model(m2.ap4)
```



On met en évidence uniquement un effet significatif du temps :
perte de poids après 4 jours.

Diagnostics du modèle pour le jeu de données 2 après J4

```
d2.ap4$residus <- residuals(m2.ap4)
ggplot(d2.ap4, aes(x = temps, y = residus)) + facet_wrap(~ traitement)
  geom_point() + geom_hline(yintercept = 0)
```



Conclusion sur l'utilisation du modèle linéaire mixte

Le modèle linéaire mixte ne propose pas de solution générale à l'analyse des données longitudinales mais peut être utilisé dans certains cas pour traiter tout ou partie des données observées après une réflexion sur les conditions d'utilisation du modèle, en particulier l'hypothèse de linéarité de la réponse par rapport au temps.

Partons plutôt des questions biologiques

Lorsque l'on étudie l'effet d'un traitement sur une réponse dans le temps il est important de se demander ce que l'on connaît *a priori* sur la forme attendue de la cinétique et de ce que l'on cherche à voir dans les cinétiques observées afin de définir plus précisément les objectifs de l'analyse .

Dans l'exemple du jeu de données 2, on sait que l'induction de la colite n'est pas immédiate et que la perte de poids due à la colite va démarrer aux alentours de J4. On voudrait savoir si la prise de buprénorphine, qui a pour but de limiter la douleur des animaux, influe ou non sur la perte de poids. On voudrait savoir aussi si la buprénorphine a un effet au tout début lié aux nausées fréquemment rencontrées en début de traitement par un morphinique.

Démarche simple en deux étapes

- 1 On définit des index ayant un sens biologique pour résumer l'information intéressante contenue dans chaque cinétique et on estime ces index sur chaque cinétique.
Ex. d'index :
 - **réponse** à un temps donné, **différence de réponse** entre deux temps donnés, **pente** estimée sur une partie de la cinétique,
 - **temps** pour atteindre une valeur donnée de la réponse,
 - **paramètre d'un modèle** biologique ajusté sur chaque cinétique.
- 2 Pour chaque index on a donc une valeur par animal. Il devient très facile de comparer les deux groupes pour chaque index.

Analyse à base de deux index simples du jeu de données 2

En fonction des questions que l'on se pose sur le jeu de données 2, il peut être envisagé de résumer l'information contenue dans chaque cinétique par la différence de poids entre J1 et J0 d'une part et le gain de poids moyen quotidien après 4 jours (pente de la cinétique à partir de J4).

Consigne 8

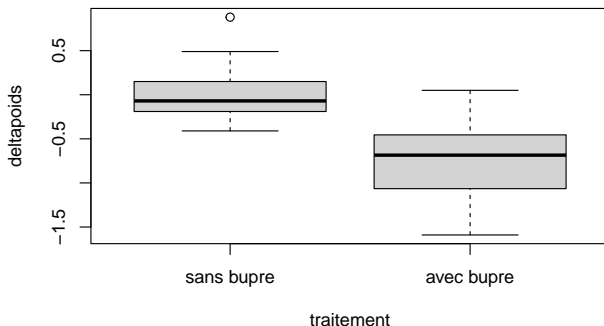
En vous inspirant des scripts fournis pour les consignes n°1 et 4 implémentez efficacement le calcul de ces deux index pour chaque cinétique puis testez l'effet de la buprénorphine sur chacun des deux index.

Calcul des différences de poids entre J1 et J0

```
calcdelta <- function(d, tempsfin, tempsdeb)
  # écrite de façon très générique
{
  dsort <- arrange(d, animal)
  dsortfin <- subset(dsort, temps == tempsfin)
  dsortdeb <- subset(dsort, temps == tempsdeb)
  # Création d'un petit jeu de données avec les infos utiles
  ddelta <- data.frame(
    deltapoids = dsortfin$poids - dsortdeb$poids,
    animal = dsortdeb$animal, traitement = dsortdeb$traitemement)
  return(ddelta)
}
dJ1J0.2 <- calcdelta(d2, tempsfin = 1, tempsdeb = 0)
```

Visualisation des différences de poids entre J1 et J0

```
boxplot(deltapoids ~ traitement, data = dJ1J0.2 )
```



Comparaison des différences de poids entre J1 et J0 entre les deux groupes

```
t.test(deltapoids ~ traitement, data = dJ1J0.2,
       paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  deltapoids by traitement
## t = 5, df = 21, p-value = 0.0002
## alternative hypothesis: true difference in means between group sans
## 95 percent confidence interval:
##  0.41 1.11
## sample estimates:
## mean in group sans bupre mean in group avec bupre
##                0.026                -0.735
```

très similaire à `summary(m.01)`

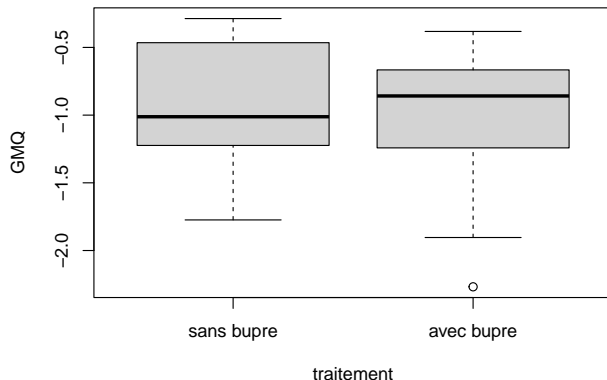
On montre un effet significatif de la buprénorphine sur la différence de poids entre J1 et J0 (perte de poids avec la buprénorphine : 0.76 g de moins que sans en moyenne).

Calcul des GMQ à partir de J4 (pentes)

```
# Fonction de calcul de l'index pour un animal
GMQapJ4 <- function(dlanimal)
{
  dfromj4 <- subset(dlanimal, temps >= 4)
  reg <- lm(poids ~ temps, data = dfromj4)
  pente <- coef(reg)[2]
  return(pente)
}
# Application de la fonction pour tous les animaux
resby <- by(d2, d2$animal, GMQapJ4)
# Récupération des résultats sous la forme d'un vecteur
gmq <- as.vector(resby)
# récupération des codes correspondants des animaux
vanim <- names(resby)
# Récupération des lers numéros de ligne associés dans le jeu initial
ind <- match(vanim, d2$animal)
# Création du vecteur traitement associé au vecteur index
trait <- d2$traitement[ind]
# on peut mettre tout dans un jeu de données
dGMQ <- data.frame(GMQ = gmq, animal = vanim, traitement = trait)
```

Visualisation des GMQ à partir de J4

```
boxplot(GMQ ~ traitement, data = dGMQ)
```



Comparaison des GMQ à partir de J4 entre les deux groupes

```
t.test(GMQ ~ traitement, data = dGMQ,
       paired = FALSE, var.equal = TRUE)

##
## Two Sample t-test
##
## data:  GMQ by traitement
## t = 0.4, df = 22, p-value = 0.7
## alternative hypothesis: true difference in means between group sans
## 95 percent confidence interval:
## -0.37  0.54
## sample estimates:
## mean in group sans bupre mean in group avec bupre
## -0.95 -1.03
```

très similaire à summary(m.ap4)

On ne montre pas d'effet significatif de la buprénorphine sur le gain de poids moyen quotidien après J4.

Calcul du temps nécessaire pour atteindre une réponse donnée

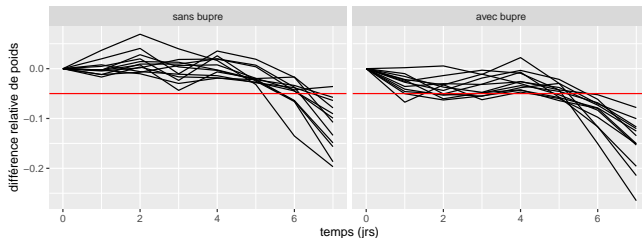
Imaginons qu'il soit intéressant d'un point de vue biologique de calculer le temps nécessaire pour atteindre une perte de poids de 5% par rapport au poids initial (en pratique on considère une perte de 15 – 20% sur quelques jours comme un point limite). Tentons de calculer cet index sur chaque cinétique.

Calcul par interpolation linéaire du temps nécessaire pour atteindre une perte de 5% du poids initial

```
tempsperte <- function(dlanimal) {  
  interp <- approx(x = dlanimal$temps, y = dlanimal$poids,  
                  method = "linear", n = 100)  
  seuil <- dlanimal$poids[dlanimal$temps == 0] * 0.95  
  i <- which(interp$y <= seuil)[1]  
  xseuil <- interp$x[i]  
  # met NA si le seuil n'est jamais atteint  
  return(xseuil)  
}  
resby <- by(d2, d2$animal, tempsperte)  
(tseuil <- as.vector(resby))  
  
## [1] 6.58 6.72 5.66 6.51 6.36 5.66 6.15 5.73 6.15 6.15 NA  
## [12] 5.23 5.37 5.73 0.78 1.77 5.80 5.09 4.88 1.06 1.84 5.30  
## [23] 2.62 1.77  
  
vanim <- names(resby)  
ind <- match(vanim, d2$animal)  
traitement <- d2$traitement[ind]
```

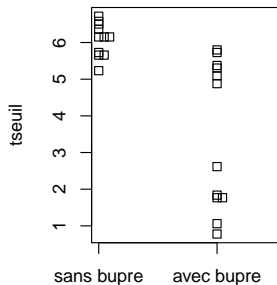
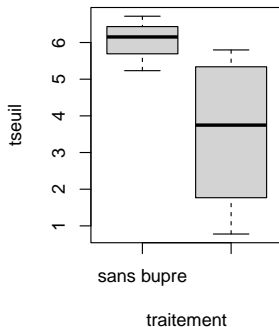

Représentation graphique associée des données

```
d2$poidsinit <- numeric(nrow(d2))  
for (i in 1:nrow(d2)) {  
  d2$poidsinit[i] <- subset(d2, (animal == d2$animal[i]) &  
    (temps == 0))$poids  
}  
d2$diffpoidsrel <- (d2$poids - d2$poidsinit) / d2$poidsinit  
ggplot(d2, aes(x = temps, y = diffpoidsrel, group = animal)) +  
  facet_wrap(~ traitement) + xlab("temps (jrs)") +  
  ylab("différence relative de poids") + geom_line() +  
  geom_hline(yintercept = -0.05, col = "red")
```



Visualisation de ces temps

```
par(mfrow = c(1, 2))  
boxplot(tseuil ~ traitement)  
stripchart(tseuil ~ traitement, vertical = TRUE, method = "stack")
```



Difficultés rencontrées sur ce type d'index

- On a souvent des données censurées (sur certaines cinétiques le seuil n'est pas encore atteint au dernier temps d'observation).
- Les cinétiques ne sont souvent pas strictement monotones : on peut passer en dessous du seuil puis remonter au-dessus.
- Les distributions d'un tel index peuvent être assez éloignées d'une loi normale et notamment multi-modales.

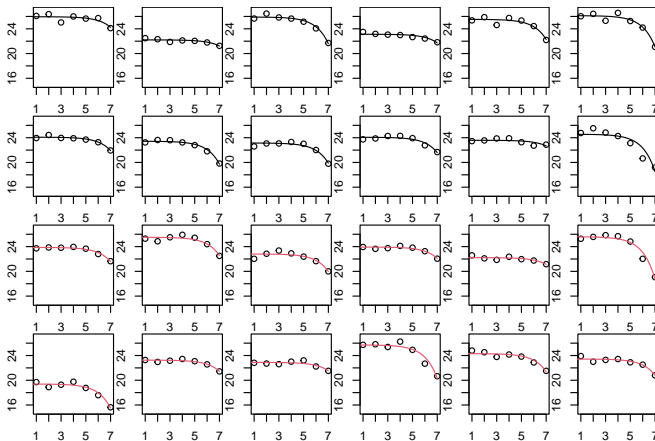
Ajustement d'un modèle paramétrique sur chaque cinétique à partir de J1

```
## Définition d'un modèle exponentiel décroissant
form <- as.formula("poids ~ P0 - b * (exp(temps) - 1)")

## Fonction d'ajustement par régression non linéaire utilisant nls()
fitmnl <- function(dlanimal)
{
  dapJ1 <- dlanimal[dlanimal$temps >= 1, ]
  # Estimation de valeurs initiales pour les paramètres
  P0start <- dapJ1$poids[dapJ1$temps == 1]
  bstart <-
    (P0start - dapJ1$poids[dapJ1$temps == 7]) / (exp(7) - 1)
  mnl <- nls(form, start = list(P0 = P0start, b = bstart),
    data = dapJ1)
  par <- coef(mnl)
  plot(dapJ1$temps, dapJ1$poids, ylim = c(15, 27))
  x <- seq(0, 7, length.out = 100)
  lines(x, par["P0"] - par["b"] * (exp(x) - 1),
    col = dapJ1$traitemement[1])
  return(par["b"])
}
```

Application de la fonction sur chaque cinétique

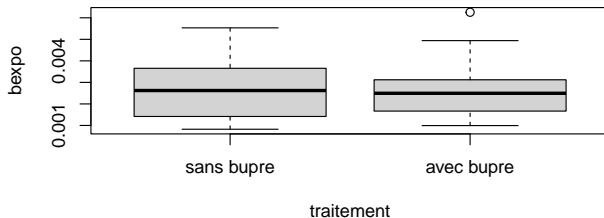
```
par(mfrow = c(4, 6), mar = c(2, 2, 0, 0))  
resby <- by(d2, d2$animal, fitmnl)
```



Comparaison entre groupes d'un des paramètres estimés

Imaginons que le paramètre b du modèle soit d'intérêt biologique.

```
bexpo <- as.vector(resby)
vanim <- names(resby)
ind <- match(vanim, d2$animal)
traitement <- d2$traitement[ind]
boxplot(bexpo ~ traitement)
```



Limites des approches par modélisation non linéaire

Bien entendu une approche par modélisation non linéaire n'a d'intérêt que si un même modèle décrit bien l'ensemble des cinétiques, qu'il est pertinent d'un point de vue biologique, et que l'un ou plusieurs de ces paramètres contient l'information que l'on souhaite extraire des cinétiques et comparer entre groupes.

Avec un modèle non linéaire il est tout à fait possible de réunir les deux étapes (estimation de l'index pour chaque cinétique puis comparaison des index entre groupes) en une seule (techniquement plus complexe mais statistiquement plus pertinent).

Conclusion

Ce cours n'avait pas pour but de vous proposer une méthode unique permettant d'analyser des données longitudinales dans tous les cas (je ne connais malheureusement pas de telle méthode),

mais de vous montrer sur un exemple simple la **diversité des approches** possibles, les **limites** de chacune d'elles, et la **nécessité de se poser en amont la question de ce que l'on souhaite extraire comme information biologique de chaque cinétique** si l'on veut avoir une chance d'aboutir à des conclusions pertinentes.

A vous de bâtir ensuite une méthode adaptée (inspirée ou non de celles précédemment proposées) pour **répondre à vos objectifs**, en faisant des **hypothèses biologiques pertinentes**.