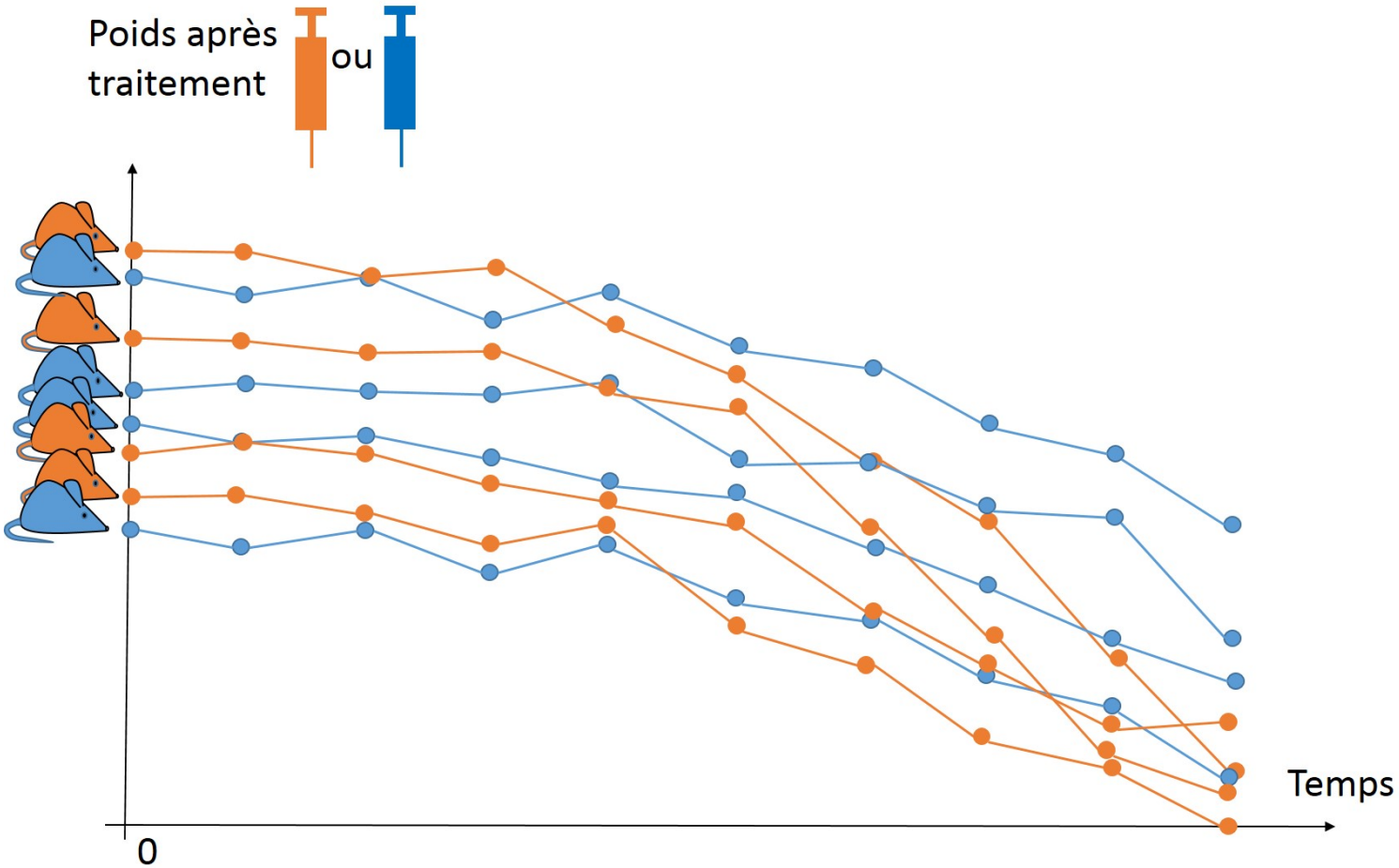
Analyse de données longitudinales ou mesures répétées dans le temps - diverses approches plus ou moins adaptées suivant les données et les objectifs de l'analyse

M. L. Delignette-Muller
VetAgro Sup - LBBE

9 janvier 2020

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)
> str(d1)

'data.frame':        192 obs. of  4 variables:
 $ animal    : Factor w/ 24 levels "A1","A10","A11",..: 1 1 1 1 1 1 1 1
 $ traitement: Factor w/ 2 levels "avec DSS","sans DSS": 2 2 2 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)

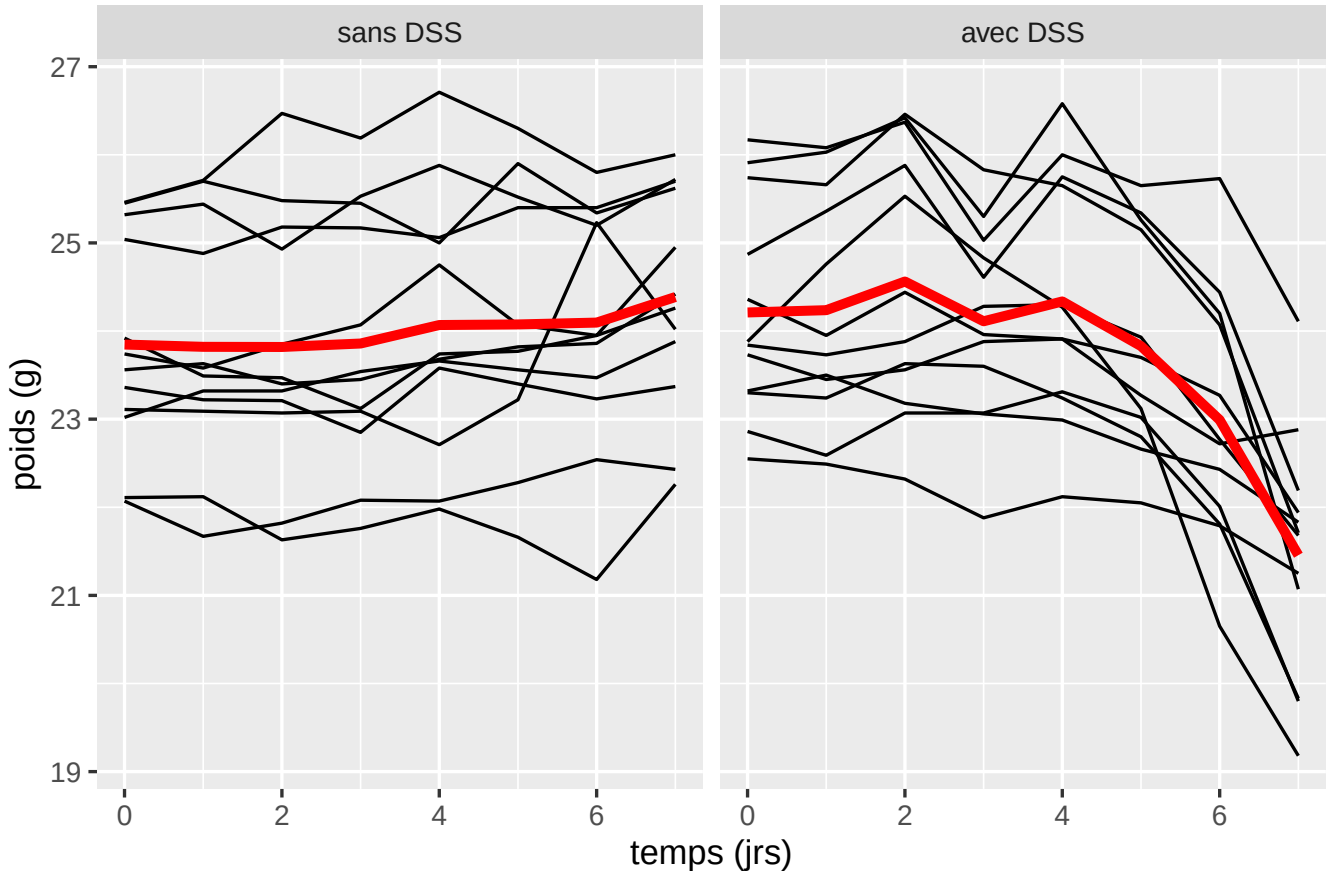
```
> d2 <- read.table("DATA/buprenorphine.txt", header = TRUE)
> str(d2)

'data.frame':        192 obs. of  4 variables:
 $ animal    : Factor w/ 24 levels "B1","B10","B11",..: 1 1 1 1 1 1 1 1
 $ traitement: Factor w/ 2 levels "avec bupre","sans bupre": 2 2 2 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)
+ {
+   plot1 <- qplot(temps, poids, data = d, geom ="line",
+                  group = animal, facets = ~ traitement,
+                  xlab = "temps (jrs)", ylab = "poids (g)")
+   plot1 + stat_summary(aes(group = traitement),
+           geom = "line", lwd = 1.5, fun.y = mean, col = "red")
+ }
> # écriture équivalente avec ggplot()
> 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.y = mean, col = "red")
+ }
```

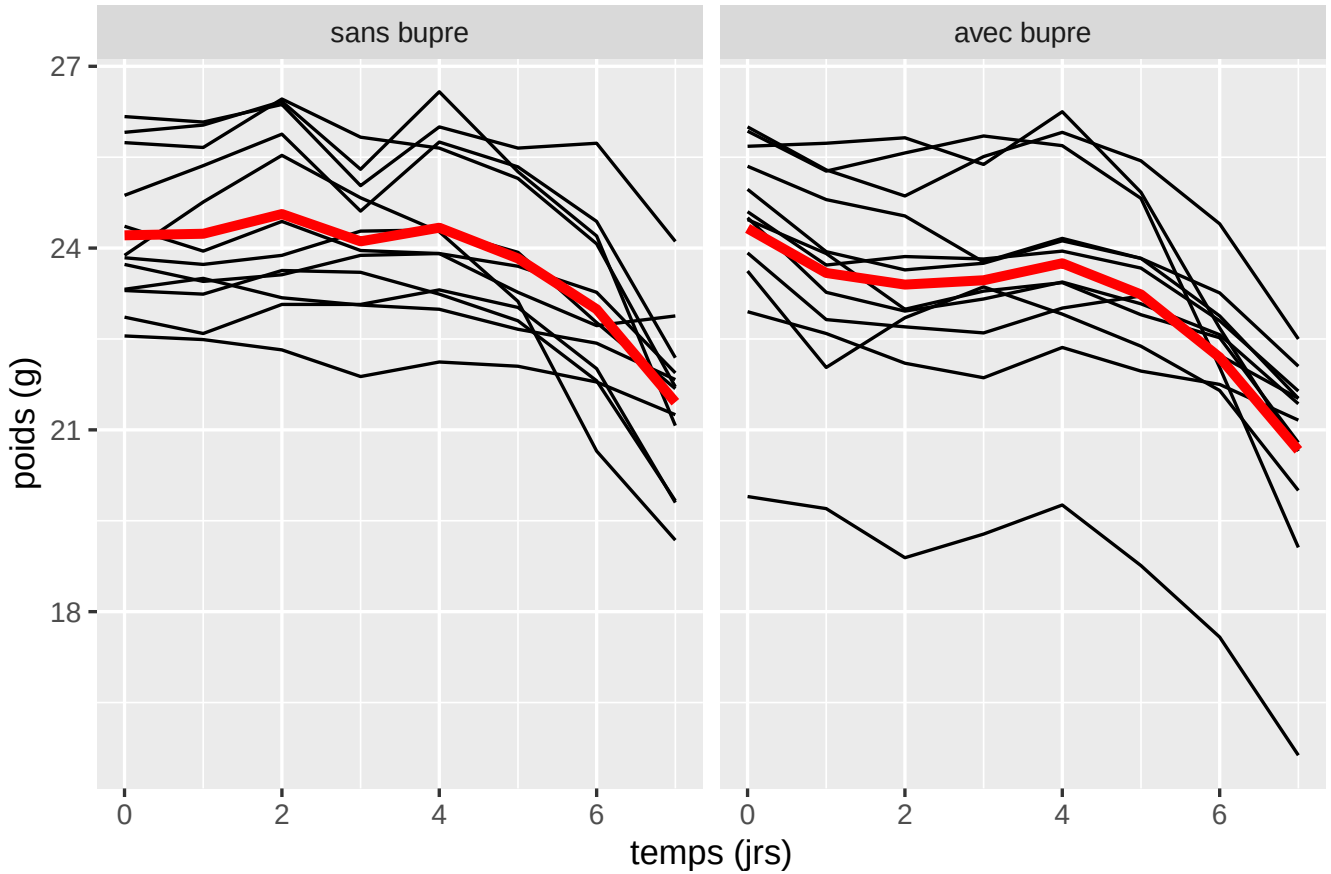
Visualisation du jeu de données exemple n°1 - DSS

```
> plotd(d1)
```



Visualisation du jeu de données exemple n°2 - buprénorphine

```
> plotd(d2)
```



Diversité des démarches utilisées

Consigne 1

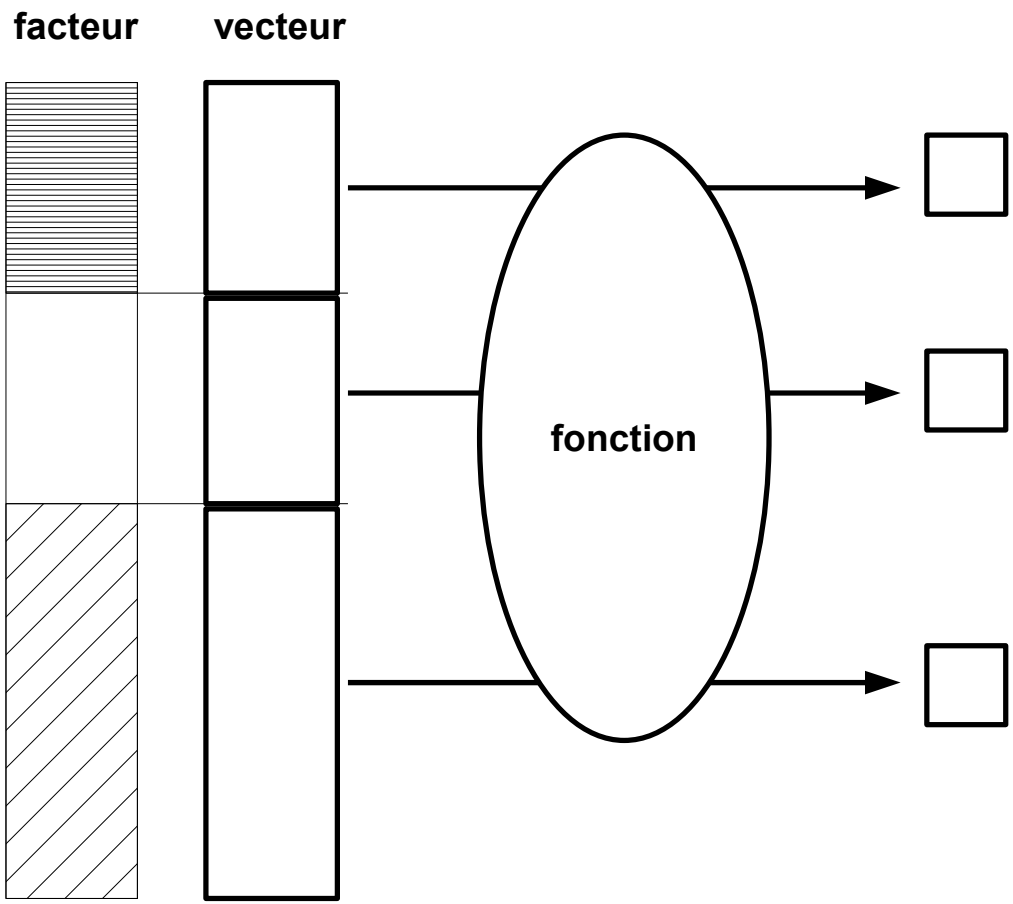
Prenez connaissance des données à l'aide du script fourni et énoncez les méthodes que vous avez déjà vues utilisées sur ce type de données où les méthodes auxquelles vous pensez.

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**.

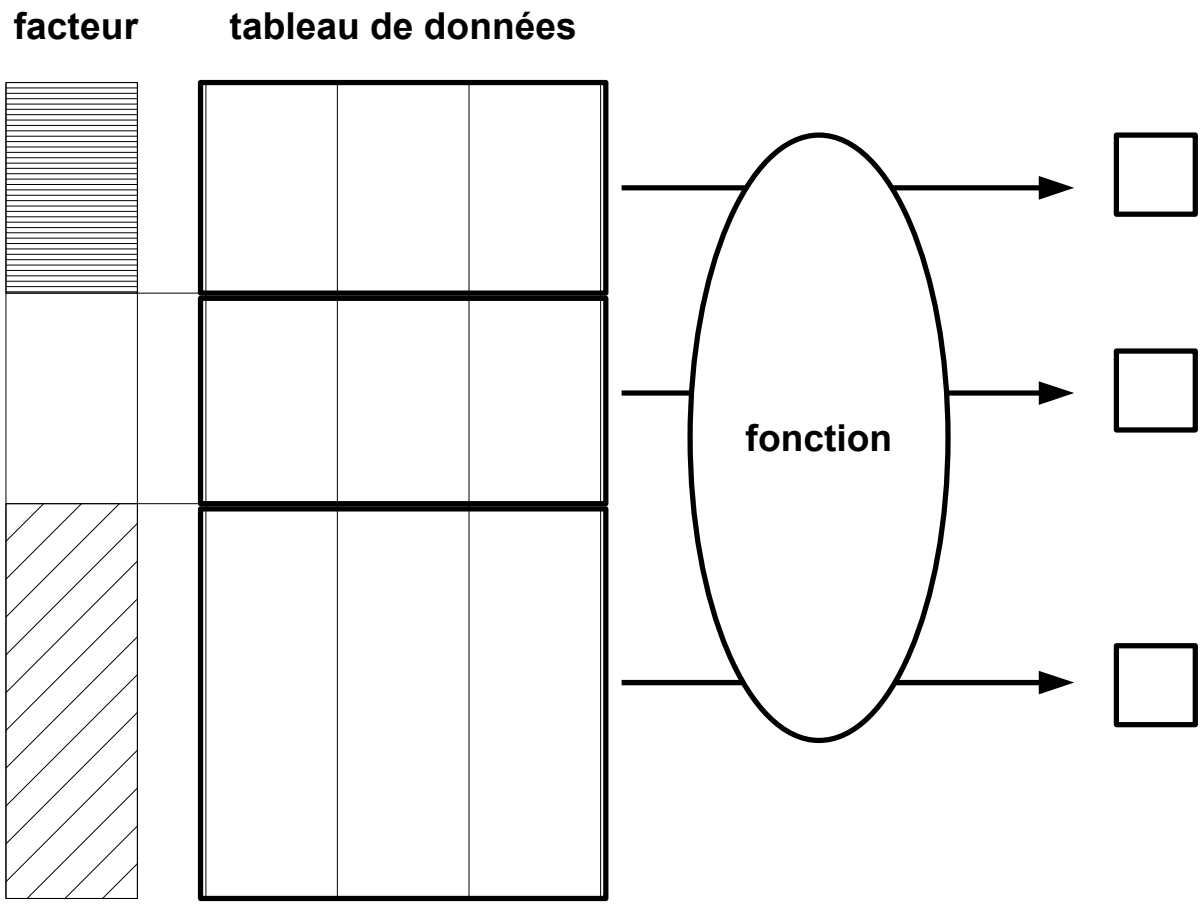
Rappel du principe de la fonction tapply

```
> 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(d1temps)
+ {
+   test <- t.test(d1temps$poids ~ d1temps$traitement,
+                  paired = FALSE, var.equal = TRUE)
+   return(test$p.value)
+ }
> ##  Essai au temps 1 pour voir si ça fonctionne
> test1tempsv1(subset(d1, temps == 0))

[1] 0.472
```

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

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

```
> by(d1, d1$temps, test1tempsv1)

d1$temps: 0
[1] 0.472
------------------------------------------------------------
d1$temps: 1
[1] 0.447
------------------------------------------------------------
d1$temps: 2
[1] 0.23
------------------------------------------------------------
d1$temps: 3
[1] 0.634
------------------------------------------------------------
d1$temps: 4
[1] 0.649
------------------------------------------------------------
d1$temps: 5
[1] 0.657
```

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$temps, test1tempsv1))
+   names(pval) <- unique(d$temps)
+   return(pval)
+ }
> ## application au jeu de données d1
> testachqtemps(d1)
```

```
       0        1        2        3        4        5        6        7
4.72e-01 4.47e-01 2.30e-01 6.34e-01 6.49e-01 6.57e-01 6.54e-02 1.92e-05
```

Application de l'approche aux deux jeux de données

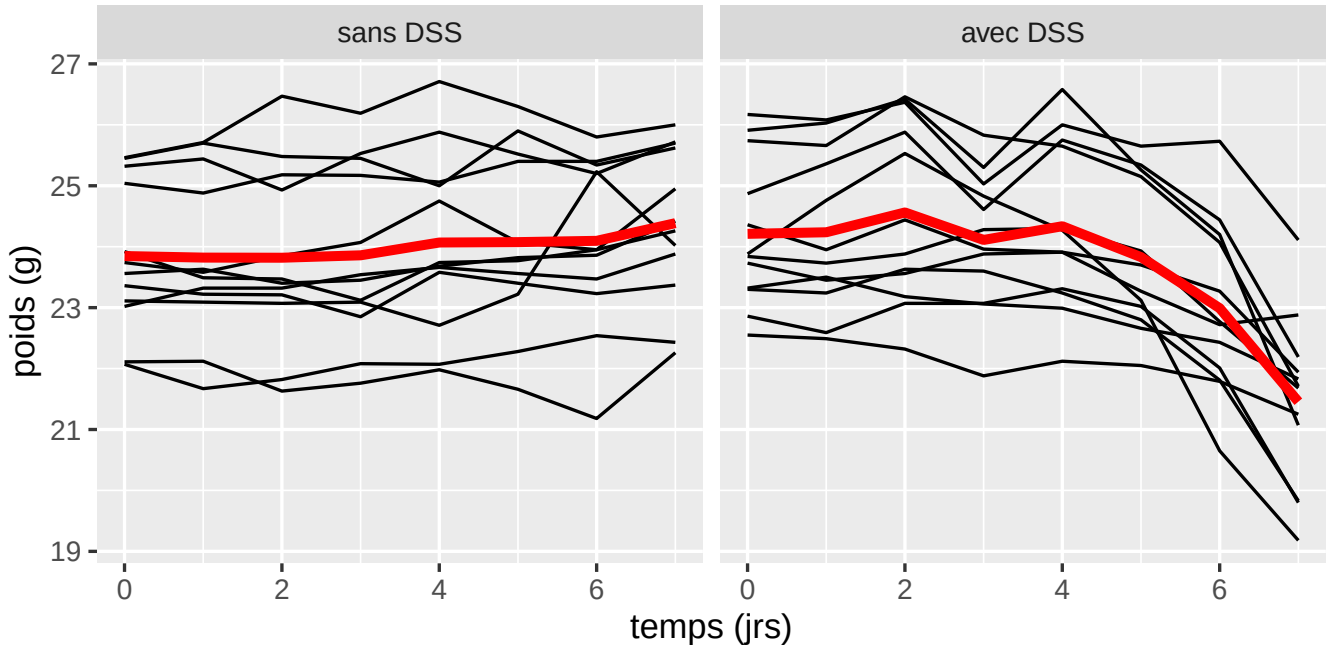
Consigne 2

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

```
> plotd(d1)
> testachqtemps(d1)
```

```
          0           1           2           3           4           5           6           7
4.72e-01 4.47e-01 2.30e-01 6.34e-01 6.49e-01 6.57e-01 6.54e-02 1.92e-05
```

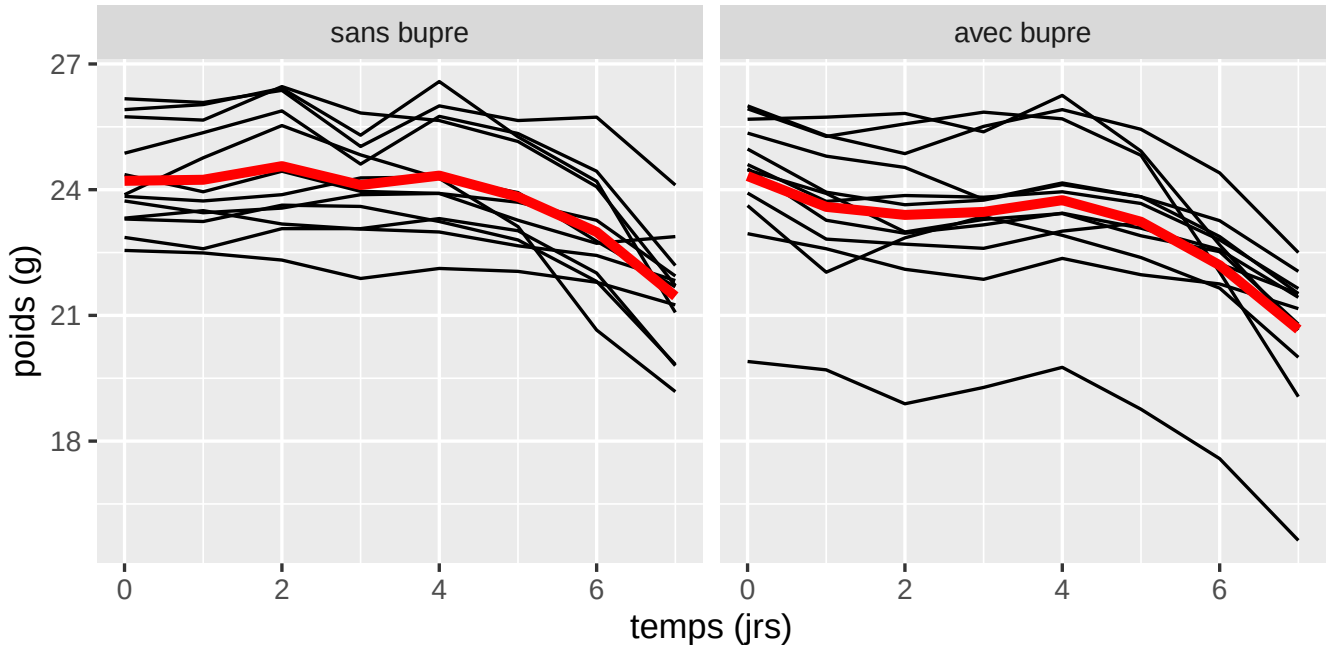


Différence significative entre les deux groupes à 7 jours (pourtant il semble que les réponses diffèrent dès J5 - J6)

Comparaison à chaque temps sur le jeu de données 2

```
> plotd(d2)
> testachqtemps(d2)
```

```
    0     1     2     3     4     5     6     7
0.851 0.303 0.103 0.300 0.374 0.345 0.215 0.243
```



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)
+ }
```

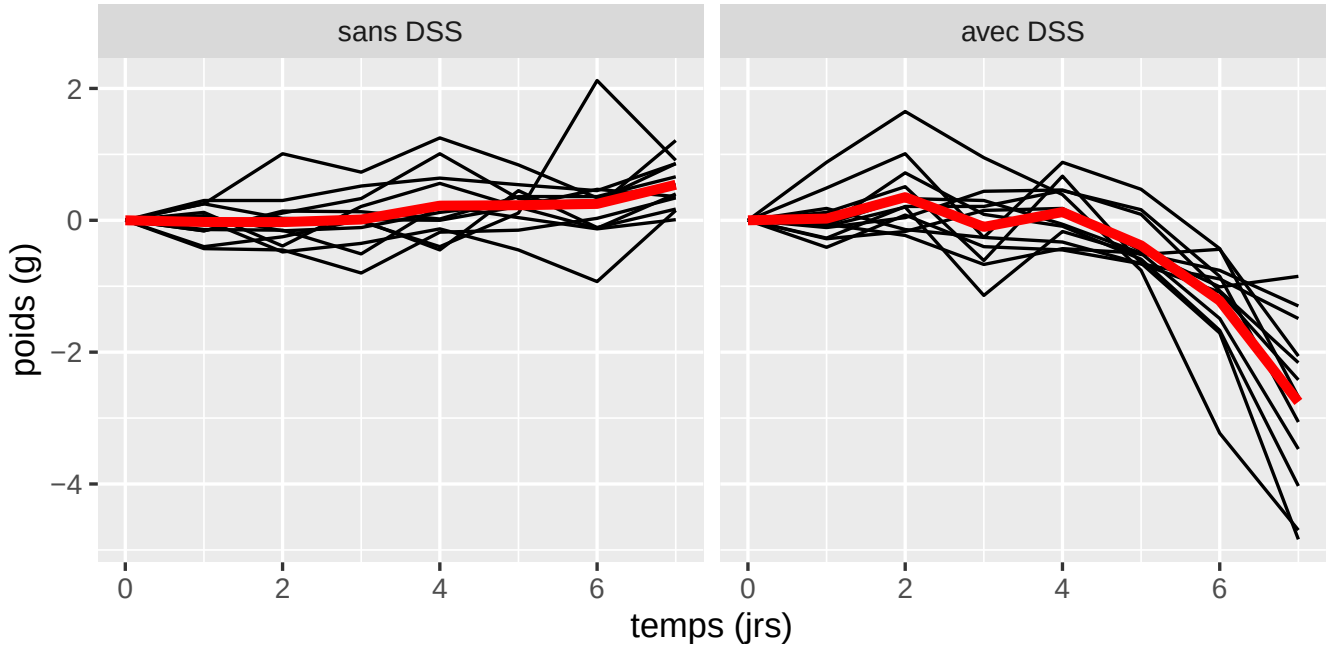
Consigne 3

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	7
NaN	6.82e-01	6.92e-02	5.92e-01	6.36e-01	4.64e-04	6.46e-05	2.42e-08

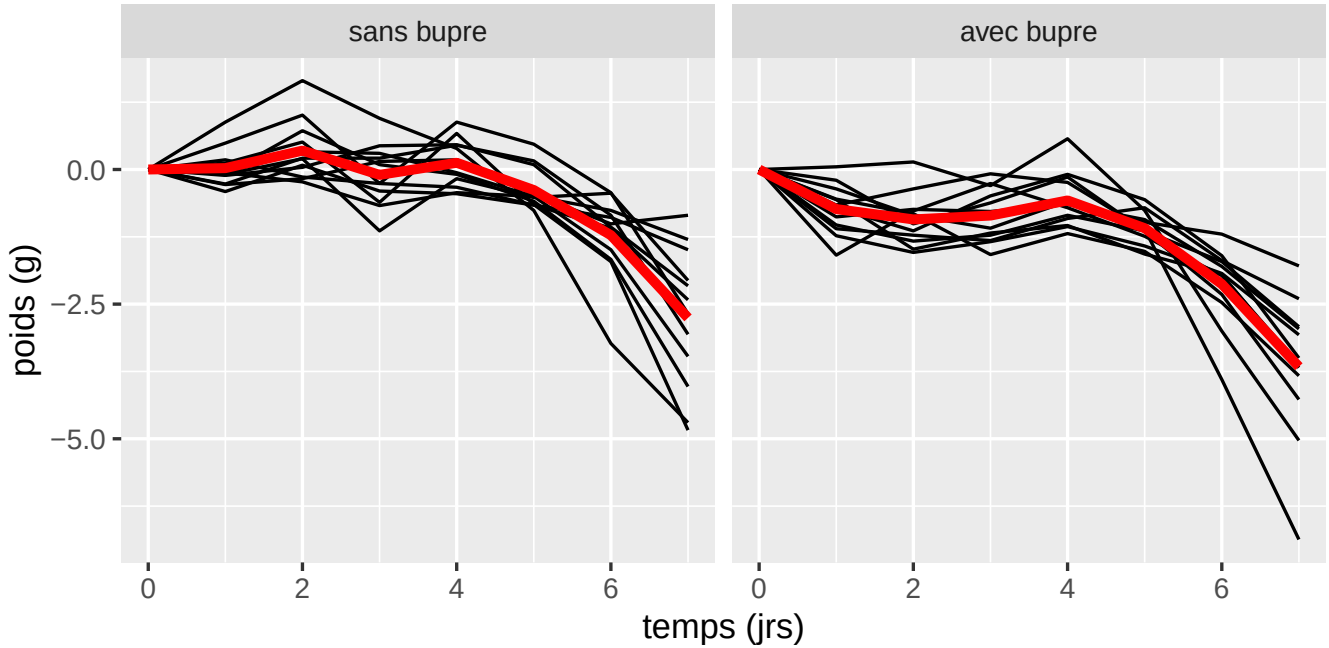


Différence significative observée à 5, 6 et 7 jours.

Analyse du jeu de données 2 modifié

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

```
       0          1          2          3          4          5          6          7
     NaN 1.69e-04 4.24e-06 2.23e-03 1.69e-03 7.84e-05 6.58e-03 1.04e-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}.

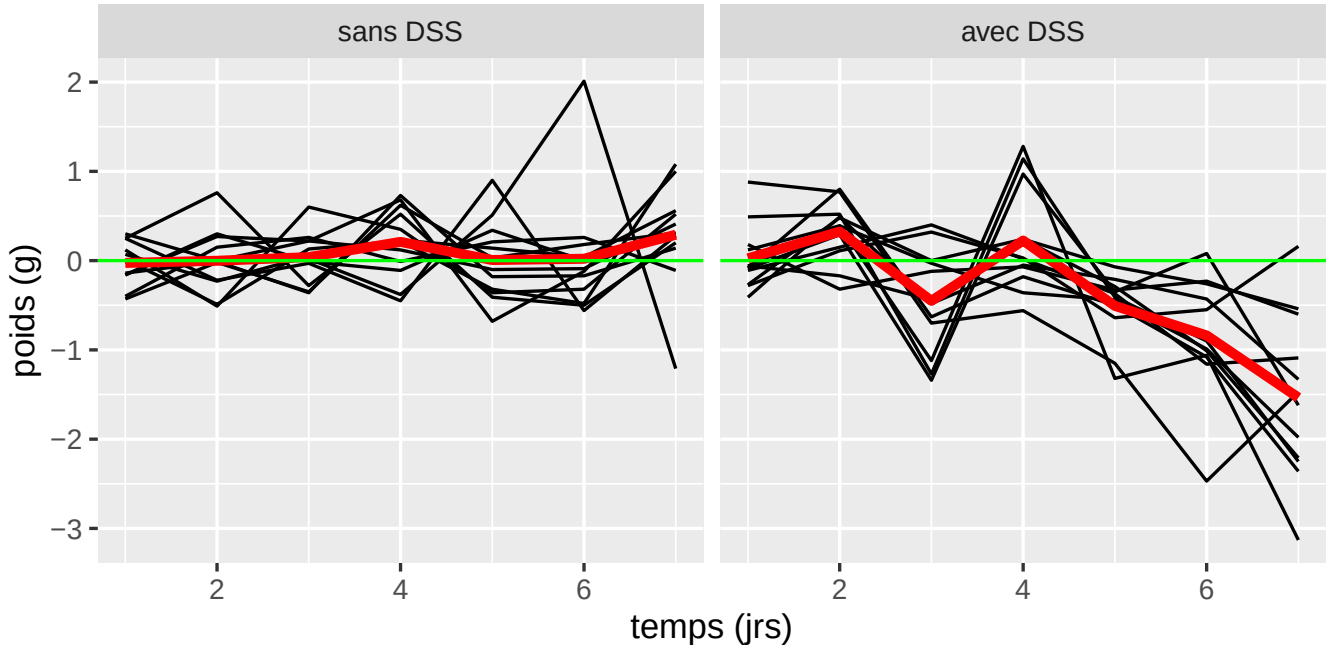
```
> require(dplyr)
> modifydbis <- function(d)
+ {
+   dsort <- arrange(d, animal, temps)
+   animu <- unique(d$animal)
+   db <- subset(dsort, temps != 0)
+   for (j in 1:length(animu))
+   {
+     animj <- animu[j]
+     dsortanimj <- subset(dsort, animal == animj)
+     db[db$animal == animj, ]$poids <- diff(dsortanimj$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        7
6.82e-01 3.03e-02 1.62e-02 9.43e-01 5.45e-03 4.63e-03 7.86e-06
```

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



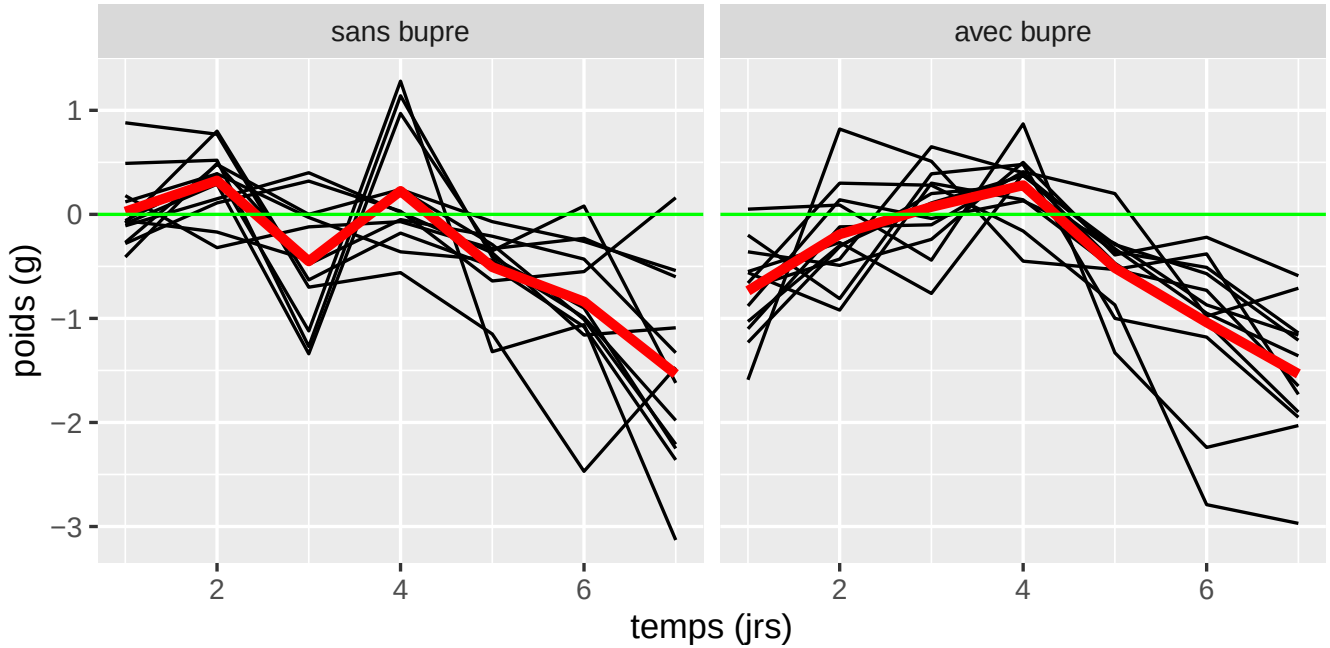
Différence significative observée à 5, 6 et 7 jours (même résultat que précédemment) mais aussi à 2 et 3 jours.

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
0.000169 0.005998 0.019228 0.782806 0.957969 0.495593 0.995969
```

```
> 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 moyennes des différences $J_i - J_{i-1}$

Une dernière méthode simple que l'on va explorer consiste à calculer la moyenne sur tous les temps des différences au temps précédent et à les comparer entre les deux groupes.

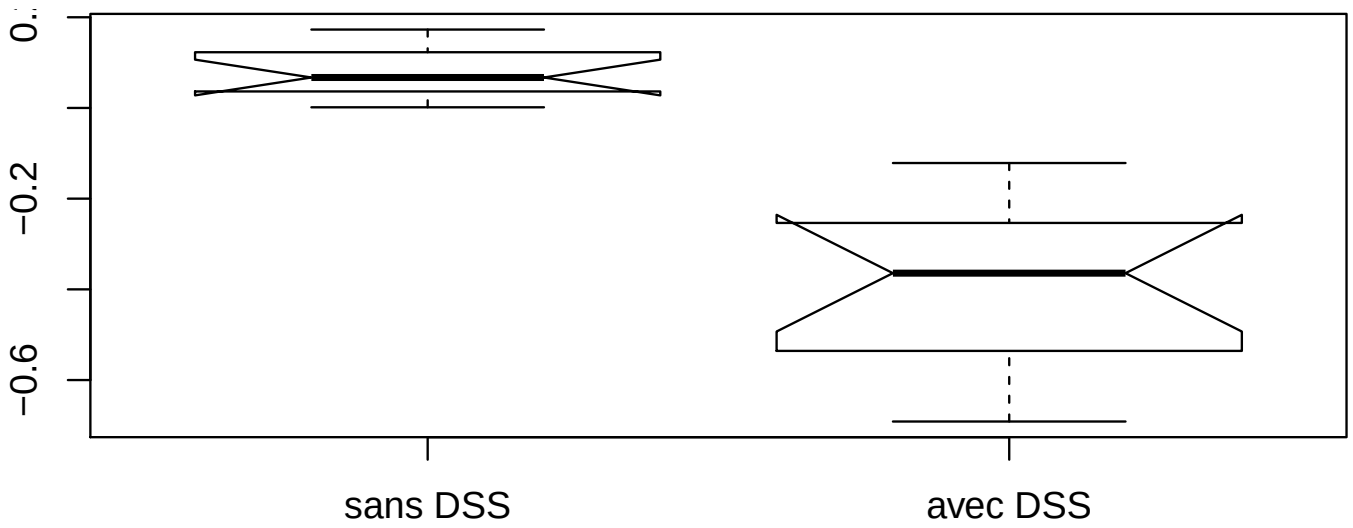
```
> # Calcul des moyennes par animal sur tous les temps
> d <- d1c
> mpoids <- tapply(d$poids, d$animal, mean)
> # récupération des noms des animaux correspondant aux moyennes
> vanim <- names(mpoids)
> # récupération du 1er indice des lignes correspondant à chaque animal
> ind <- match(vanim, d$animal)
> # récupération du vecteur des traitements correspondant à mpoids
> traitement <- d$traitement[ind]
```

Consigne 4

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

Calcul des moyennes sur tous les temps - jeu de données 1 sur différences $J_i - J_{i-1}$

```
> d <- d1c
> mpoids <- tapply(d$poids, d$animal, mean)
> # récupération des noms des animaux correspondant aux moyennes
> vanim <- names(mpoids)
> # récupération du 1er indice des lignes correspondant à chaque animal
> ind <- match(vanim, d$animal)
> traitement <- d$traitement[ind]
> par(mar = c(3,2,0.1,0.1)); boxplot(mpoids ~ traitement, notch = TRUE)
```



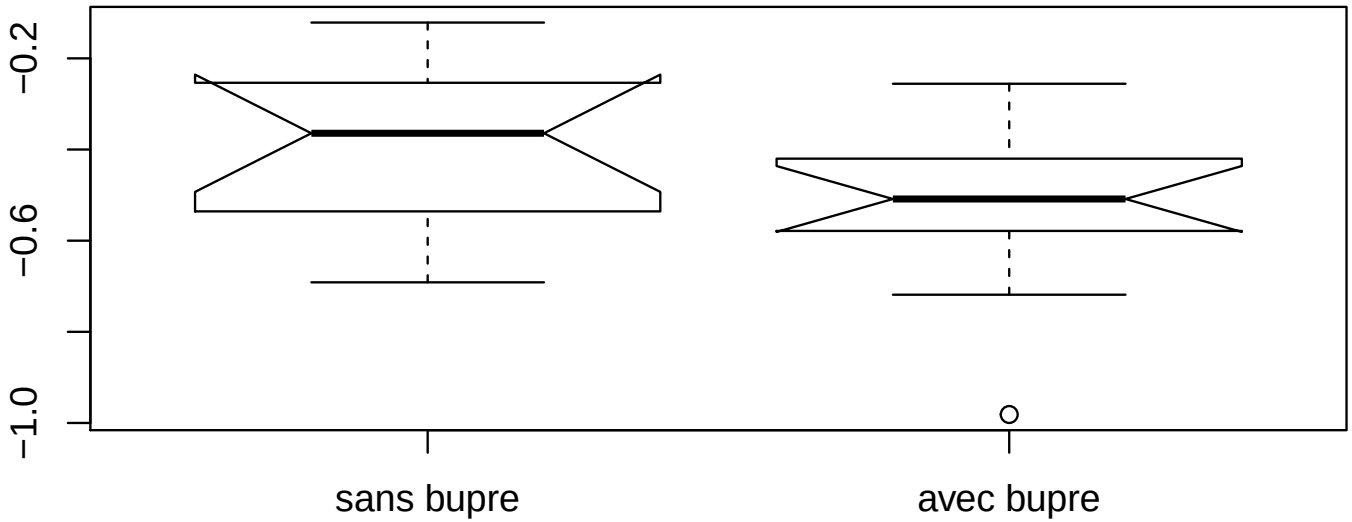
Test de comparaison des moyennes des différences au temps précédent sur le jeu de données 1

```
> t.test(mpoids ~ traitement, paired = FALSE, var.equal = FALSE)

        Welch Two Sample t-test

data:  mpoids by traitement
t = 8, df = 10, p-value = 1e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.350 0.591
sample estimates:
mean in group sans DSS mean in group avec DSS
                 0.077                -0.394
```

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

Calcul des moyennes sur tous les temps - jeu de données 2 sur différences $J_i - J_{i-1}$

```
> d <- d2c
> mpoids <- tapply(d$poids, d$animal, mean)
> # récupération des noms des animaux correspondant aux moyennes
> vanim <- names(mpoids)
> # récupération du 1er indice des lignes correspondant à chaque animal
> ind <- match(vanim, d$animal)
> traitement <- d$traitement[ind]
> par(mar = c(3,2,0.1,0.1)); boxplot(mpoids ~ traitement, notch = TRUE)
```



Test de comparaison des moyennes des différences au temps précédent sur le jeu de données 2

```
> t.test(mpoids ~ traitement, paired = FALSE, var.equal = FALSE)

        Welch Two Sample t-test

data:  mpoids by traitement
t = 2, df = 20, p-value = 0.1
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.0289  0.2879
sample estimates:
mean in group sans bupre mean in group avec bupre
                  -0.394                   -0.523
```

On conclut plus à une perte de poids moyenne significativement plus importante avec la buprénorphine, ce qui semble plus pertinente d'un point de vue biologique.

Conclusion sur l'approche calculant les moyennes sur les pertes de poids entre deux temps consécutifs

- 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)**. La méthode serait appropriée uniquement en cas d'absence d'interaction entre l'effet du temps et l'effet du traitement.

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 sur chacun des deux jeux de données.

Construction du modèle pour le jeu de données 1

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)
> m1 <- lmer(poids ~ temps + traitement:temps + (1| animal),
+            data = d1)
```

Interprétation des résultats sur le jeu de données 1

```
> summary(m1)
Linear mixed model fit by REML ['lmerMod']
Formula: poids ~ temps + traitement:temps + (1 | animal)
   Data: d1

REML criterion at convergence: 491

Scaled residuals:
   Min     1Q Median     3Q    Max
-4.342 -0.397  0.023  0.487  2.588

Random effects:
 Groups   Name        Variance Std.Dev.
 animal   (Intercept) 1.732    1.316
 Residual             0.474    0.688
Number of obs: 192, groups:  animal, 24

Fixed effects:
                      Estimate Std. Error t value
(Intercept)            24.3014     0.2836   85.70
```

Interprétation des résultats sur le jeu de données 1 (2)

```
> summary(m1)$coefficients
```

	Estimate	Std. Error	t value
(Intercept)	24.3014	0.2836	85.70
temps	0.0615	0.0301	2.04
temps:traitementavec DSS	-0.3772	0.0418	-9.03

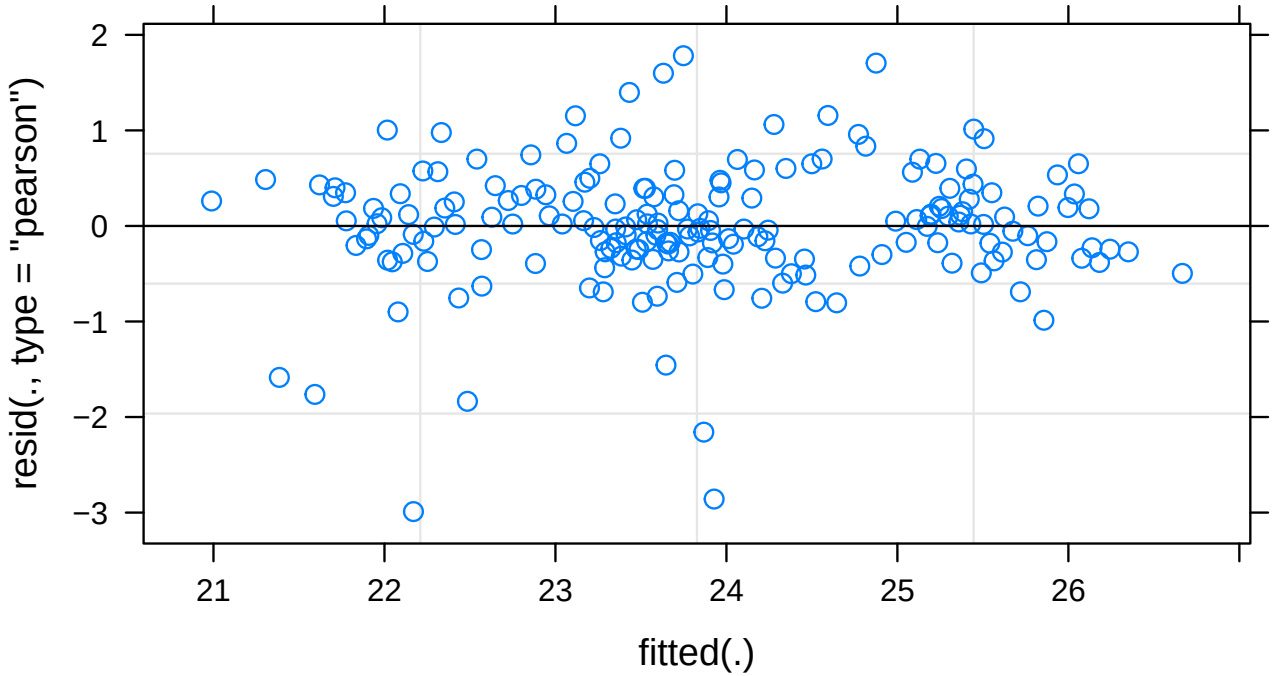
```
> confint(m1)
```

	2.5 %	97.5 %
.sig01	0.97616	1.776
.sigma	0.61731	0.765
(Intercept)	23.73773	24.865
temps	0.00153	0.120
temps:traitementavec DSS	-0.45905	-0.293

On met en évidence un effet significatif du temps et une interaction significative entre le traitement et le temps.

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

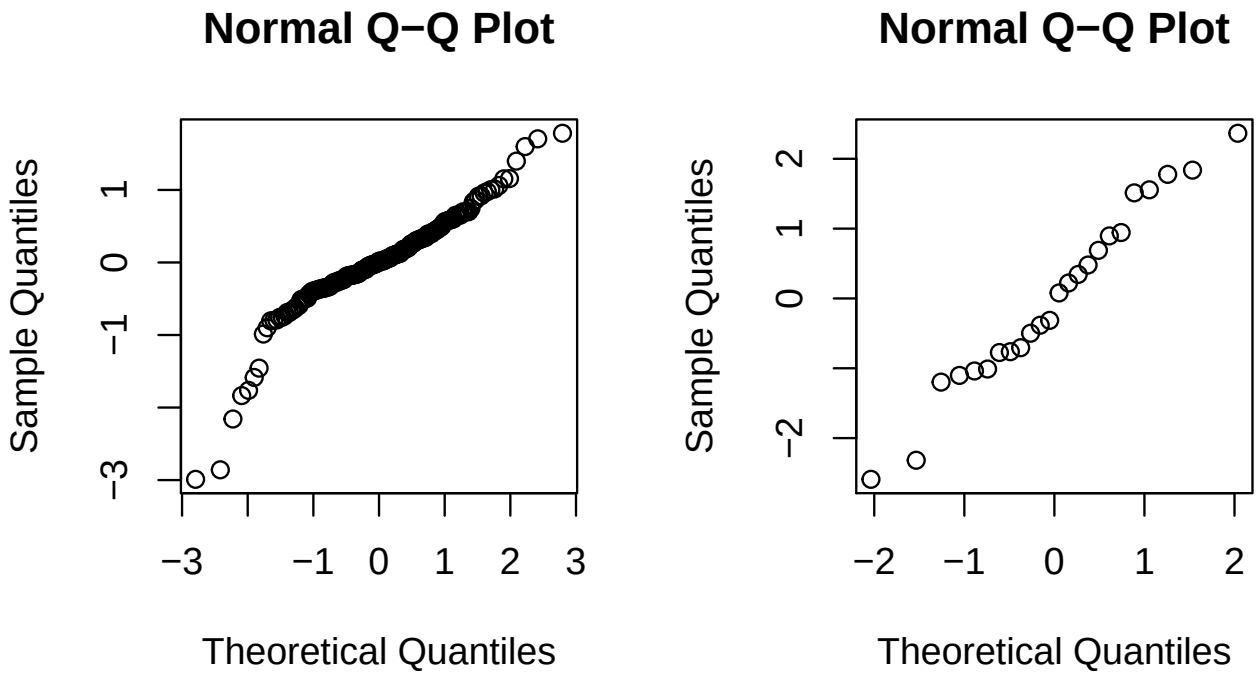
```
> plot(m1)
```



Quelques valeurs extrêmes (nettement en dessous des valeurs prédites).

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

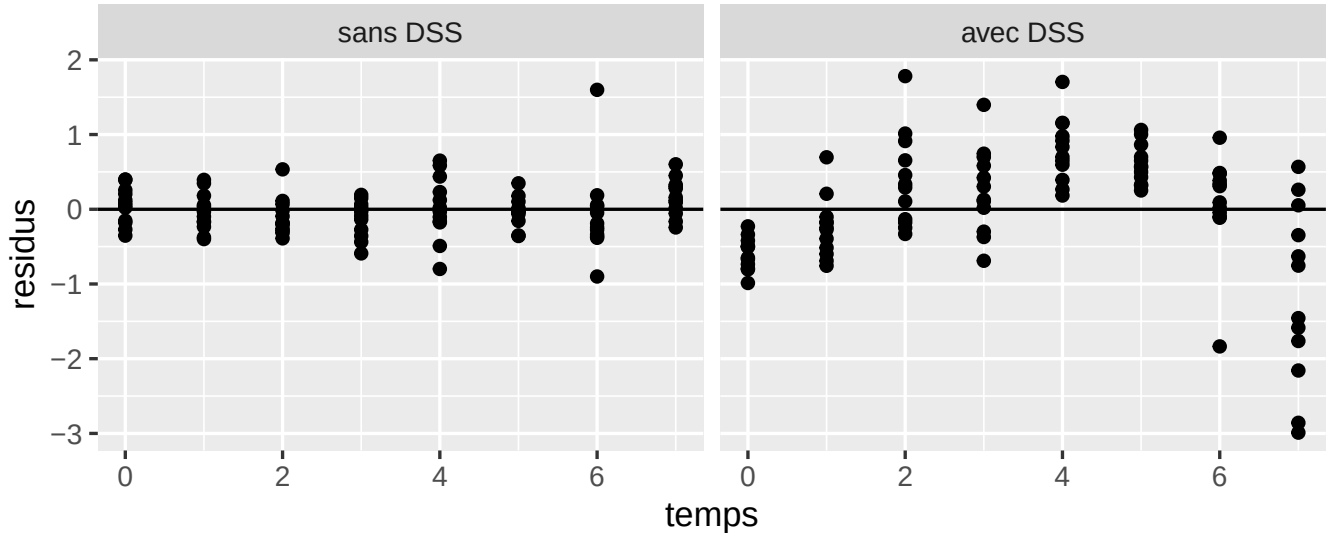
```
> par(mfrow = c(1,2))
> qqnorm(residuals(m1))
> qqnorm(ranef(m1)$animal[,1])
```



Visualisation des mêmes valeurs extrêmes.

Diagnostics du modèle pour le jeu de données 1 (3)

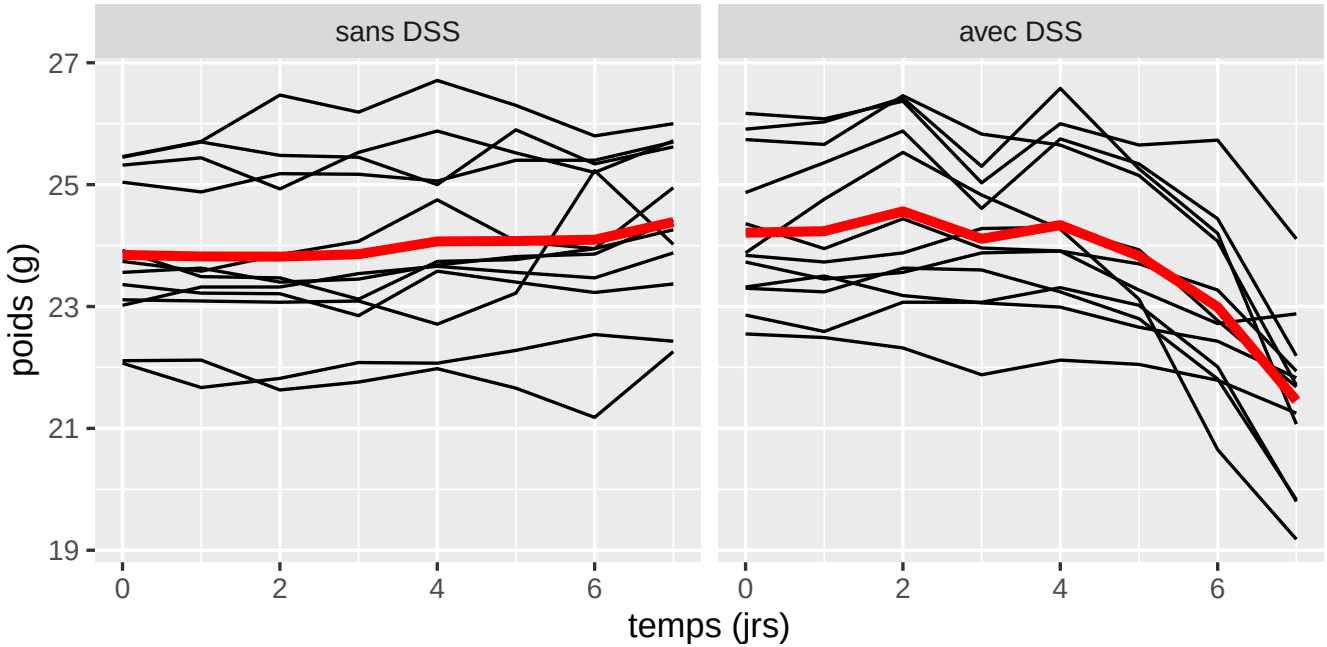
```
> d1$residus <- residuals(m1)
> ggplot(d1, aes(x = temps, y = residus)) + facet_wrap(~ traitement) +
+     geom_point() + geom_hline(yintercept = 0)
> # écriture équivalente
> # qplot(temps,residus,data=d1,facets=~traitement)+geom_hline(yinterce
```



Tendance claire sur les résidus avec DSS (modèle non linéaire).

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).



Construction du modèle pour le jeu de données 2

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

```
                              Estimate Std. Error t value
(Intercept)                    24.6532     0.3038   81.15
temps                          -0.3209     0.0388   -8.27
temps:traitementavec bupre     -0.0759     0.0533   -1.42
```

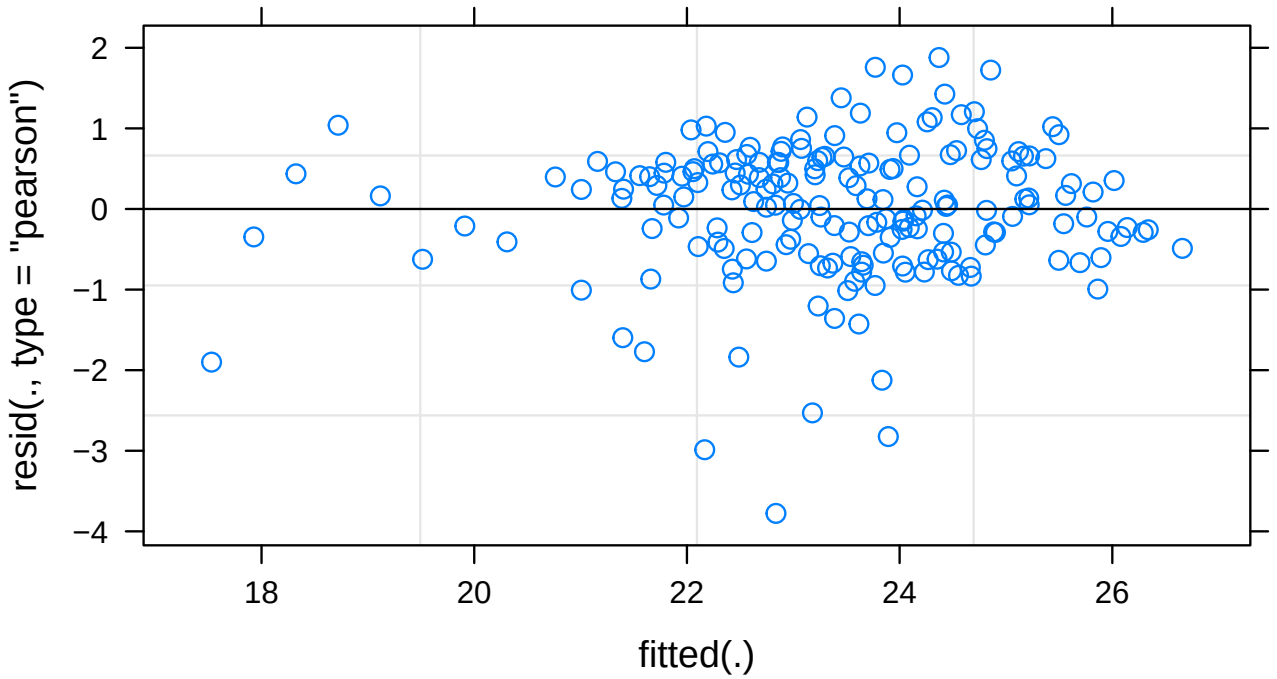
```
> confint(m2)
```

```
                                2.5 %  97.5 %
.sig01                          1.013  1.8530
.sigma                          0.803  0.9949
(Intercept)                    24.051 25.2554
temps                          -0.397 -0.2446
temps:traitementavec bupre     -0.181  0.0281
```

On met en évidence un effet significatif du temps uniquement.

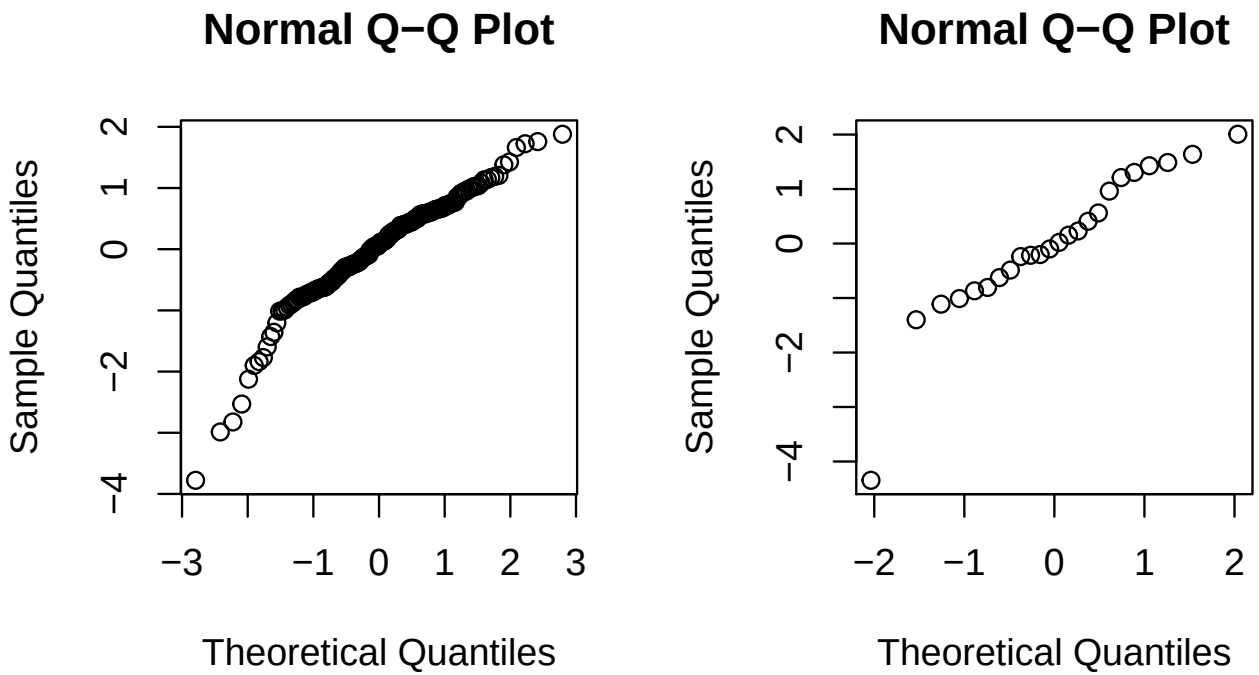
Diagnostics du modèle pour le jeu de données 2 (1)

```
> plot(m2)
```



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

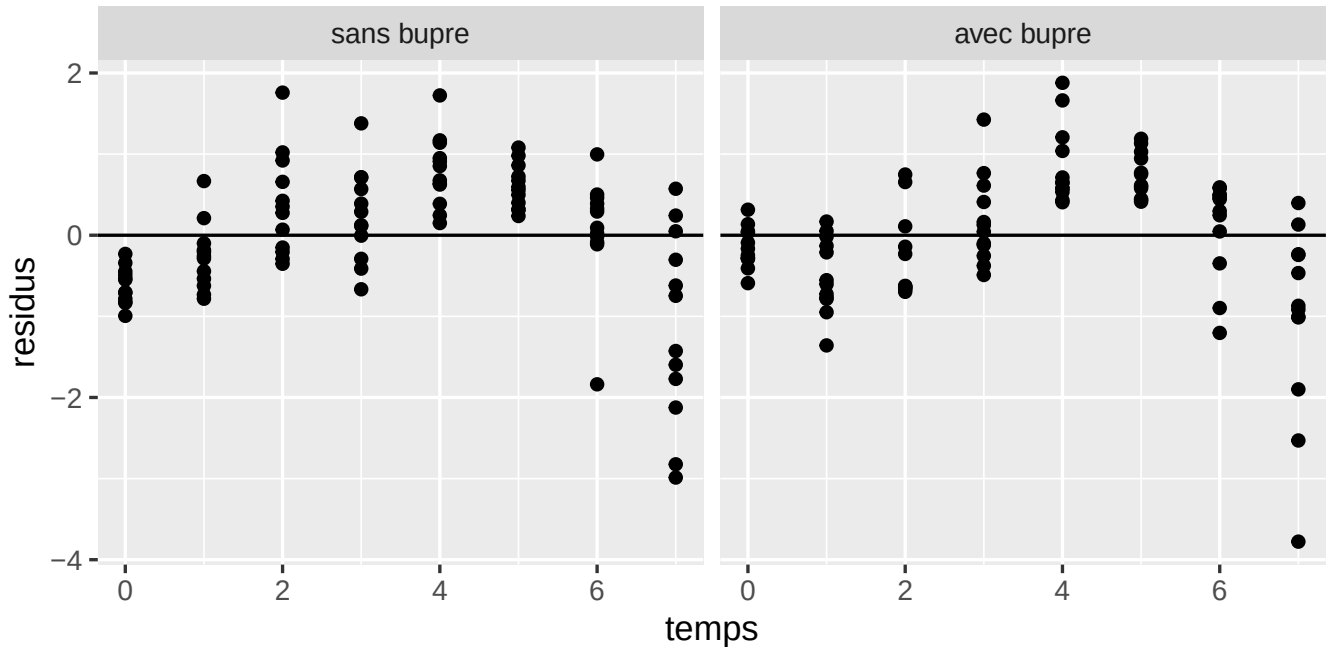
```
> par(mfrow = c(1,2))
> qqnorm(residuals(m2))
> qqnorm(ranef(m2)$animal[,1])
```



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

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

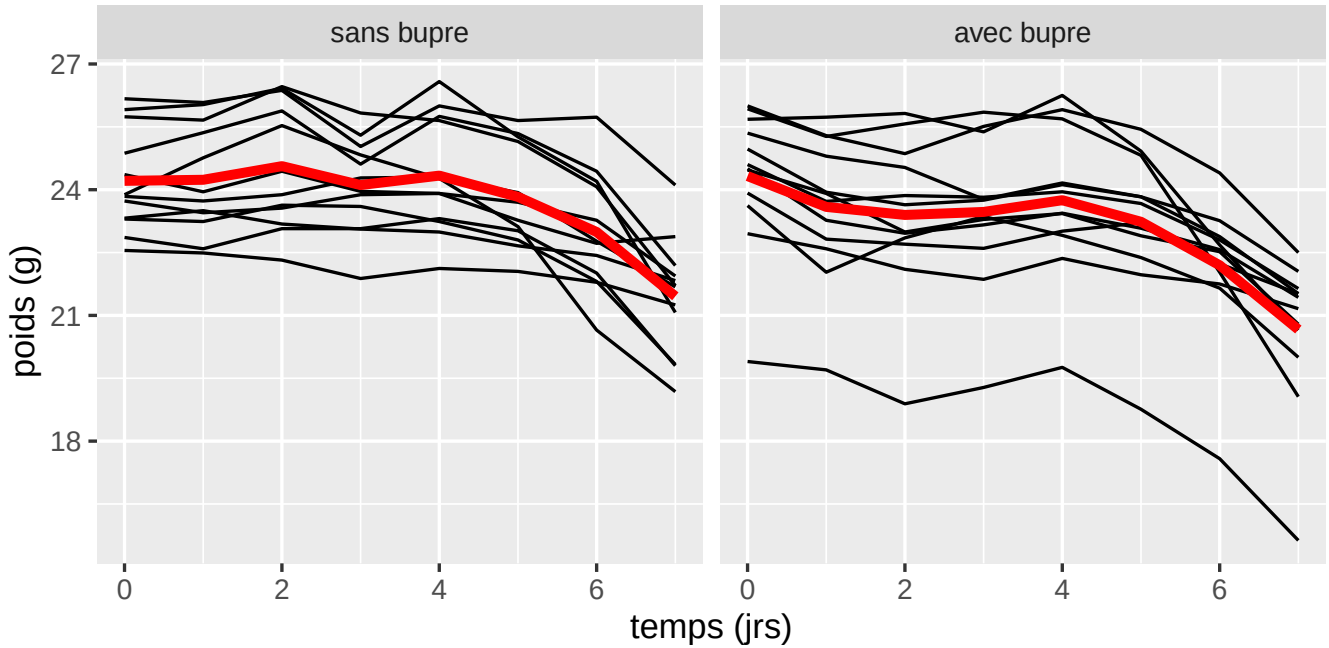
```
> d2$residus <- residuals(m2)
> ggplot(d2, aes(x = temps, y = residus)) + facet_wrap(~ traitement) +
+     geom_point() + geom_hline(yintercept = 0)
```



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

Souci majeur de l'approche retrouvé là encore

On a toujours le même souci lié au fait que la relation entre le poids et le temps n'est pas linéaire et on voit que les résidus des deux groupes diffèrent à J0 et J1.



Modèle linéaire mixte en considérant le temps comme une variable qualitative

Consigne 6

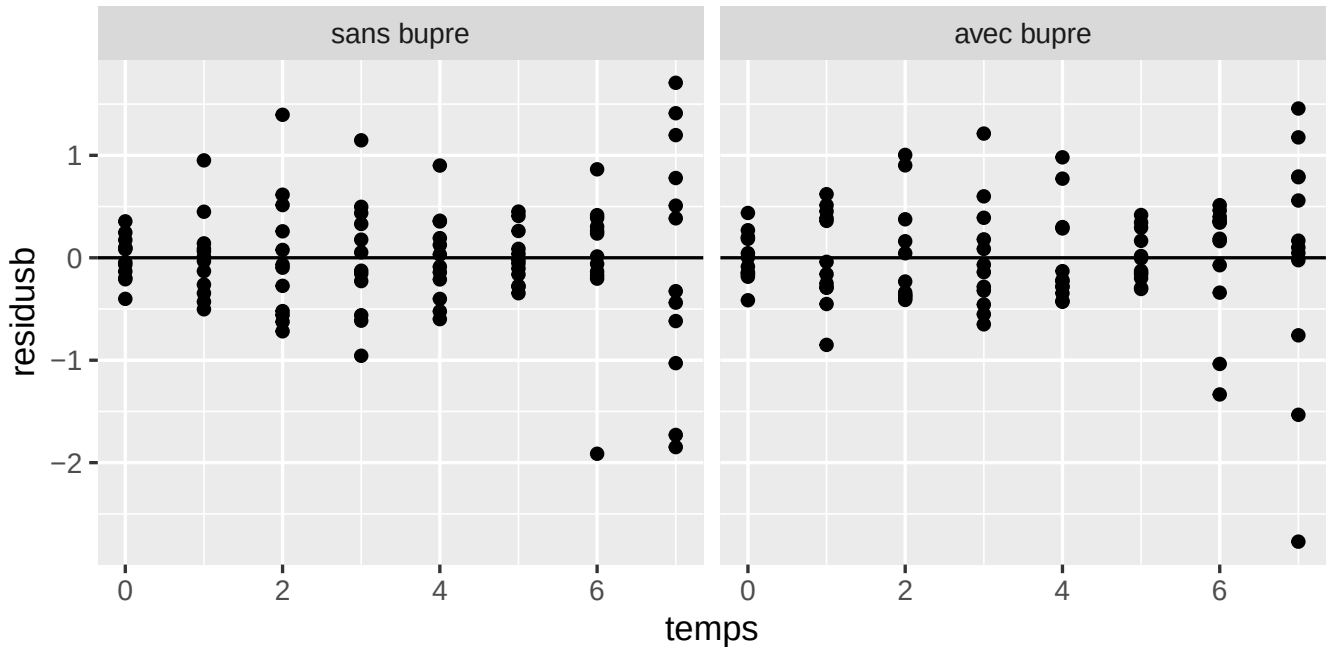
Construisez un modèle linéaire mixte prenant en compte l'effet aléatoire animal et les effets fixes traitement et temps en considérant le **temps comme qualitatif**.
Choisissez les termes que vous mettez dans le modèle selon des arguments biologiques (choix des interactions notamment), puis interprétez 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)
> 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 !

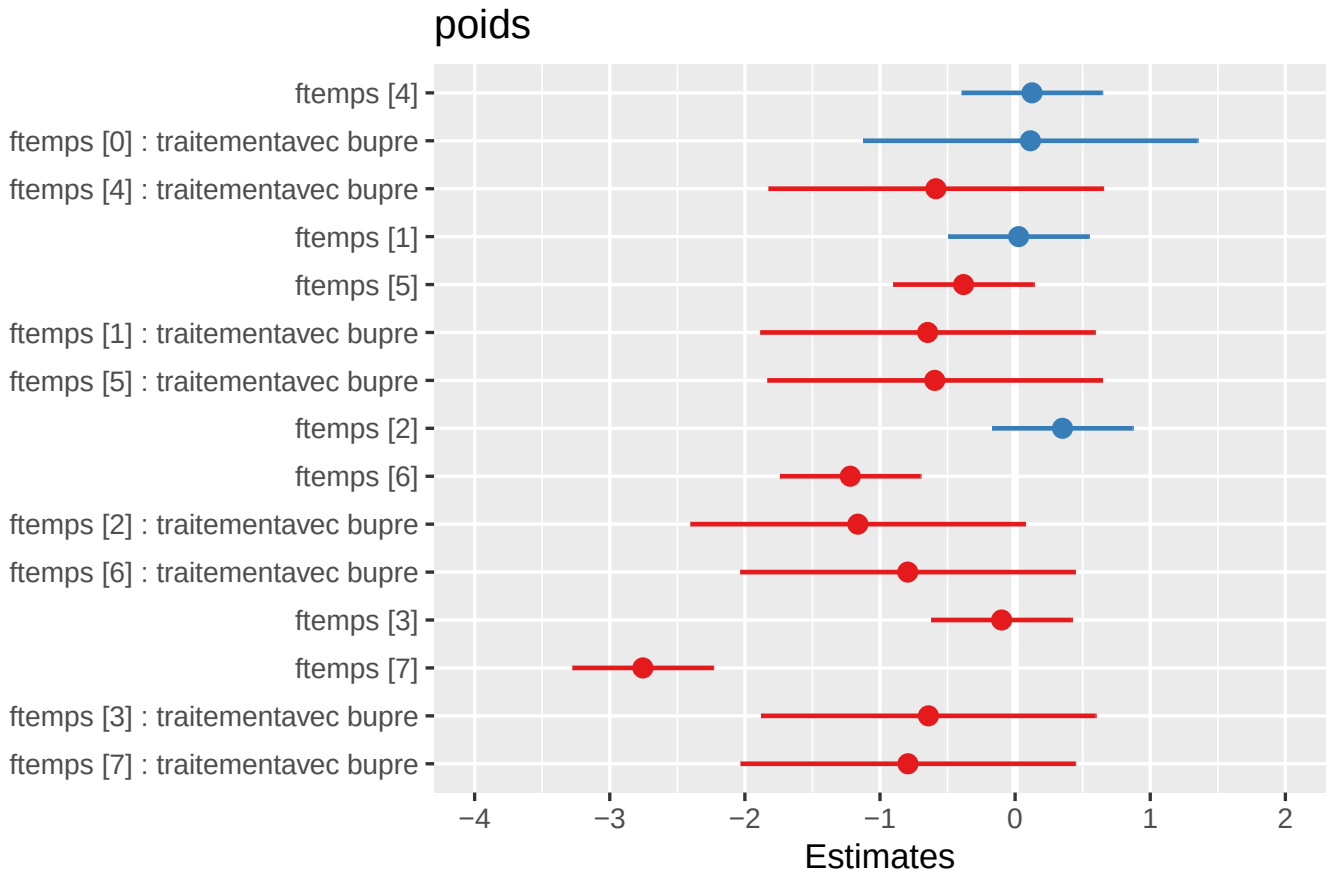
```
> summary(m2b)$coefficients
```

	Estimate	Std. Error	t value
(Intercept)	24.2108	0.447	54.2139
ftemps1	0.0258	0.265	0.0973
ftemps2	0.3508	0.265	1.3216
ftemps3	-0.1000	0.265	-0.3767
ftemps4	0.1250	0.265	0.4709
ftemps5	-0.3817	0.265	-1.4378
ftemps6	-1.2200	0.265	-4.5959
ftemps7	-2.7550	0.265	-10.3784
ftemps0:traitementavec bupre	0.1133	0.632	0.1795
ftemps1:traitementavec bupre	-0.6475	0.632	-1.0252
ftemps2:traitementavec bupre	-1.1642	0.632	-1.8433
ftemps3:traitementavec bupre	-0.6417	0.632	-1.0160
ftemps4:traitementavec bupre	-0.5867	0.632	-0.9289
ftemps5:traitementavec bupre	-0.5950	0.632	-0.9421
ftemps6:traitementavec bupre	-0.7950	0.632	-1.2588
ftemps7:traitementavec bupre	-0.7933	0.632	-1.2562

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)
```

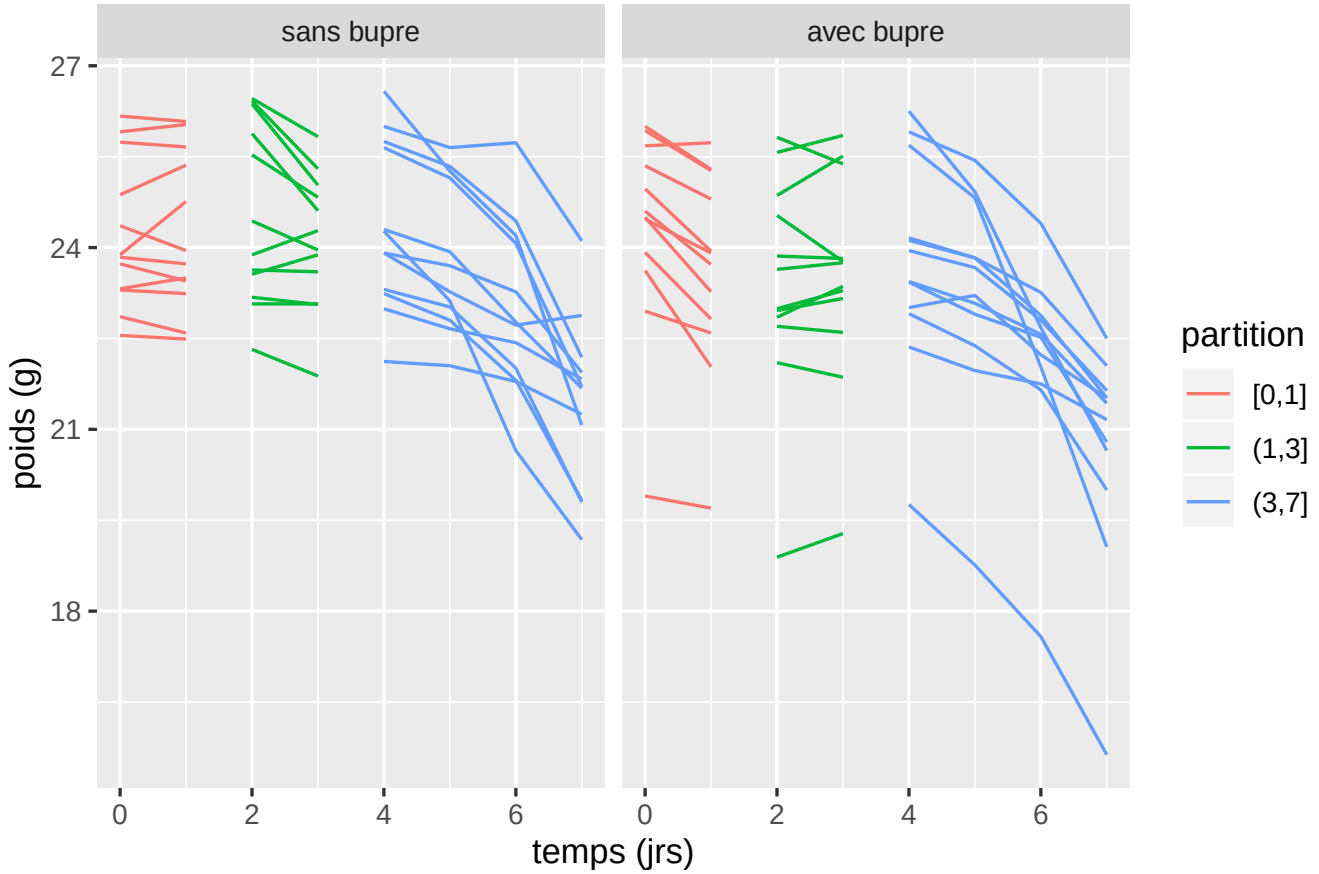


Analyse de deux sous-jeux de données

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

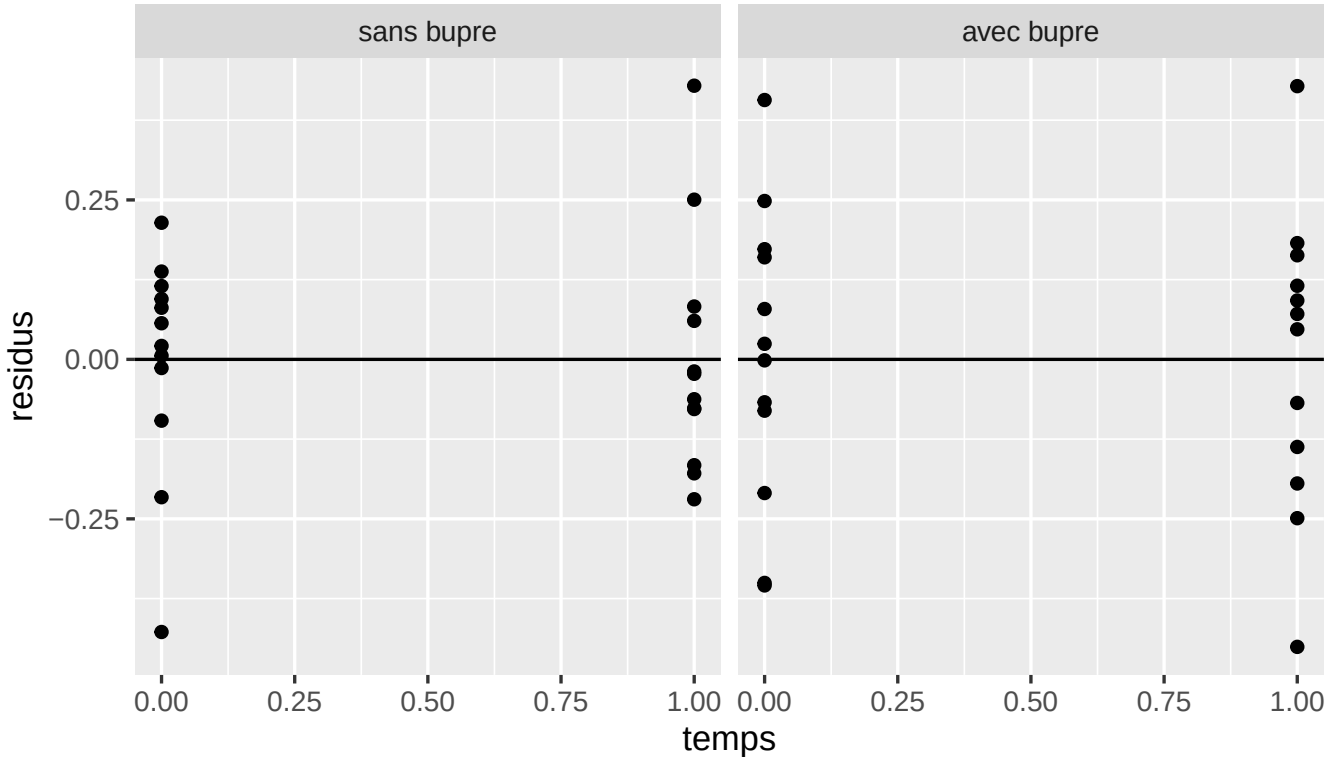
```
> d2.01 <- subset(d2, temps <= 1)
> m2.01 <- lmer(poids ~ temps + traitement:temps + (1| animal),
+               data = d2.01)
> summary(m2.01)$coefficients
```

	Estimate	Std. Error	t value
(Intercept)	24.2675	0.297	81.805
temps	0.0236	0.118	0.199
temps:traitementavec bupre	-0.7563	0.167	-4.542

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)	24.2108	0.428	56.533
ftemps1	0.0258	0.119	0.217
ftemps0:traitementavec bupre	0.1133	0.606	0.187
ftemps1:traitementavec bupre	-0.6475	0.606	-1.069

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.

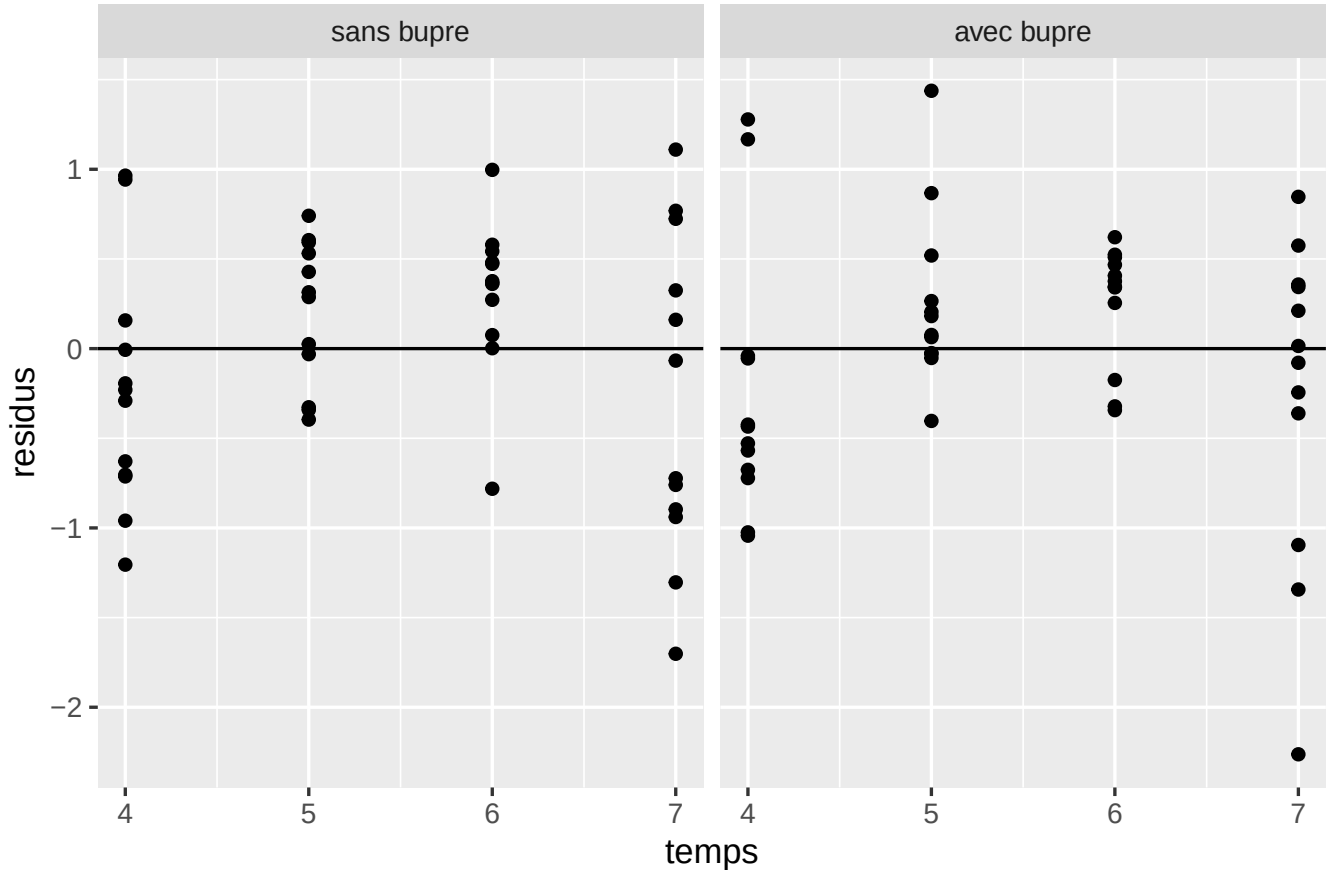
```
> d2.ap4 <- subset(d2, temps >= 4)
> m2.ap4 <- lmer(poids ~ temps + traitement + traitement:temps +
+                  (1| animal), data = d2.ap4)
> summary(m2.ap4)$coefficients
```

	Estimate	Std. Error	t value
(Intercept)	28.366	0.688	41.228
temps	-0.948	0.100	-9.446
traitementavec bupre	-0.242	0.973	-0.248
temps:traitementavec bupre	-0.082	0.142	-0.578

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

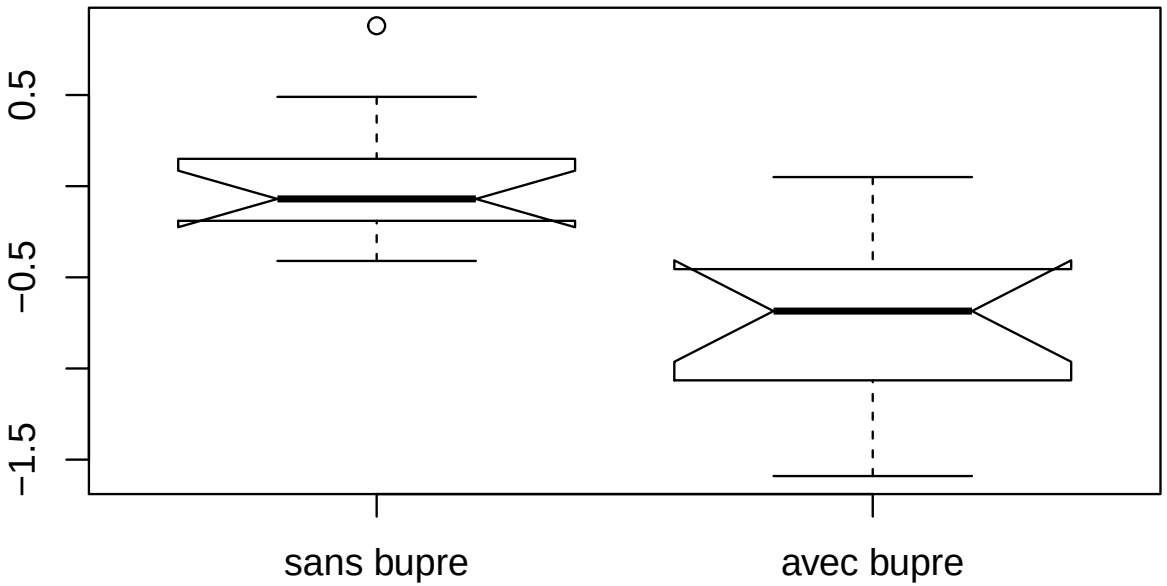
En vous inspirant des scripts fournis pour la consigne n°1 (en particulier en utilisant la fonction `by`), implémentez efficacement le calcul de ces deux index pour chaque cinétique puis testez l'effet de la buprénorphine sur chacun des deux index.

Calcul des différences de poids entre J1 et J0

```
> # Fonction de calcul de l'index pour un animal
> diffJ1J0 <- function(d1animal)
+ {
+   dj1 <- subset(d1animal, temps == 1)
+   dj0 <- subset(d1animal, temps == 0)
+   return(dj1$poids - dj0$poids)
+ }
> # Application de la fonction pour tous les animaux
> resby <- by(d2, d2$animal,diffJ1J0)
> # Récupération des résultats sous la forme d'un vecteur
> diff <- as.vector(resby)
> # récupération des codes correspondants des animaux
> vanim <- names(resby)
> # Récupération des 1ers numéros de ligne associés dans le jeu initial
> ind <- match(vanim, d2$animal)
> # Création du vecteur traitement associé au vecteur index
> traitement <- d2$traitement[ind]
```

Visualisation des différences de poids entre J1 et J0

```
> boxplot(diff ~ traitement, notch = TRUE)
```



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

```
> t.test(diff ~ traitement, paired = FALSE, var.equal = FALSE)

        Welch Two Sample t-test

data:  diff by traitement
t = 5, df = 20, p-value = 2e-04
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.411 1.111
sample estimates:
mean in group sans bupre mean in group avec bupre
                  0.0258                  -0.7350
```

très similaire à summary(m.01)

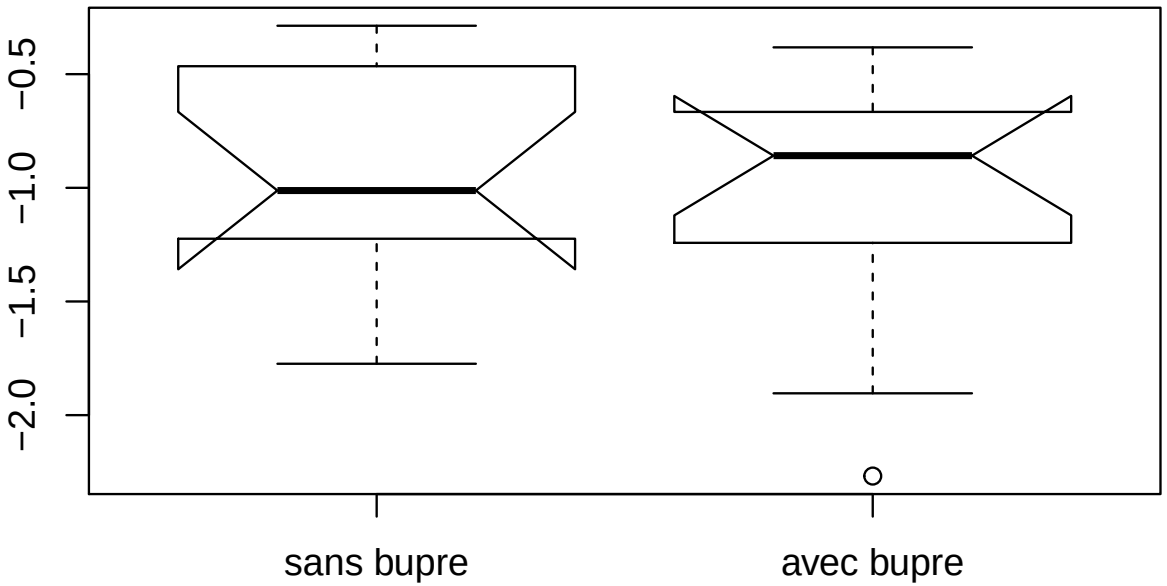
En cohérence avec ce qui avait déjà été obtenu avec diverses méthodes, on montre un effet significatif de la buprénorphine sur la différence de poids entre J1 et J0 (perte de poids avec la buprénorphine : 0.76 g de moins que sans en moyenne).

Calcul des GMQ à partir de J4 (pentes)

```
> # Fonction de calcul de l'index pour un animal
> GMQapJ4 <- function(d1animal)
+ {
+   dfromj4 <- subset(d1animal, 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$animal,GMQapJ4)
> # Récupération des résultats sous la forme d'un vecteur
> gmq <- as.vector(resby)
> # récupération des codes correspondants des animaux
> vanim <- names(resby)
> # Récupération des 1ers numéros de ligne associés dans le jeu initial
> ind <- match(vanim, d2$animal)
> # Création du vecteur traitement associé au vecteur index
> traitement <- d2$traitement[ind]
```

Visualisation des GMQ à partir de J4

```
> boxplot(gmq ~ traitement, notch = TRUE)
```



Comparaison des GMQ à partir de J4 entre les deux groupes

```
> t.test(gmq ~ traitement, paired = FALSE, var.equal = TRUE)

        Two Sample t-test

data:  gmq by traitement
t = 0.4, df = 20, p-value = 0.7
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.375  0.539
sample estimates:
mean in group sans bupre mean in group avec bupre
                  -0.948                   -1.030
```

très similaire à summary(m.ap4)

En cohérence avec ce qui avait déjà été obtenu précédemment, 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

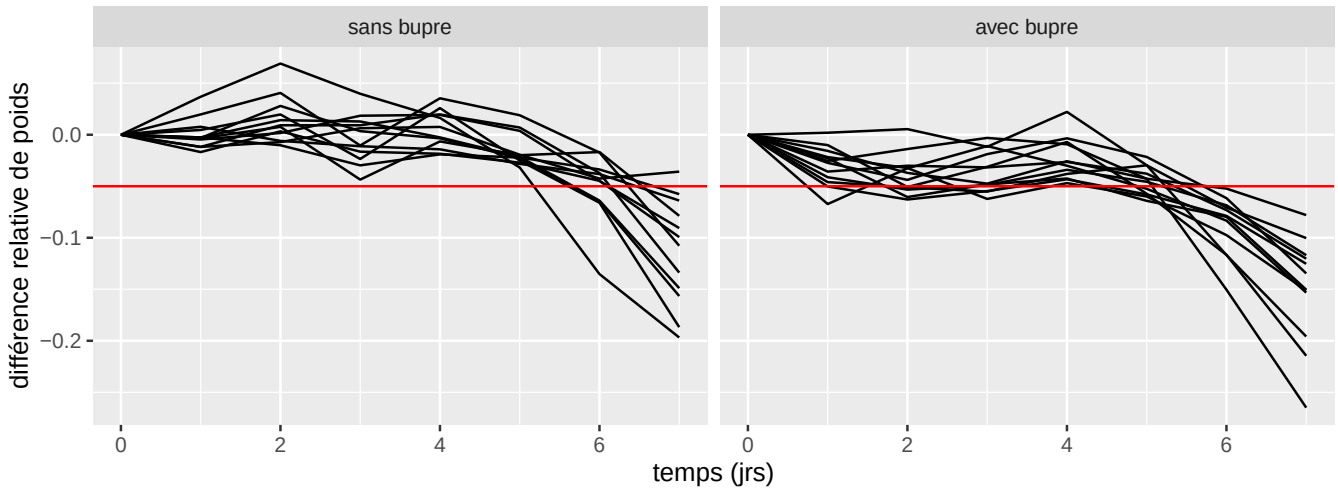
```
> tempsperte <- function(d1animal) {
+   interp <- approx(x = d1animal$temps, y = d1animal$poids,
+                    method = "linear", n = 100)
+   seuil <- d1animal$poids[d1animal$temps == 0] * 0.95
+   i <- 1
+   ni <- length(interp$x)
+   while ((interp$y[i] > seuil) & (i < ni) ) {
+     i <- i + 1
+   }
+   xseuil <- interp$x[i]
+   # met 7 si le seuil n'est jamais atteint
+   return(xseuil)
+ }
> resby <- by(d2, d2$animal,tempsperte)
> (tseuil <- as.vector(resby))

 [1] 6.576 6.717 5.657 6.505 6.364 5.657 6.152 5.727 6.152 6.152 7.000 5.232
[13] 5.374 5.727 0.778 1.768 5.798 5.091 4.879 1.061 1.838 5.303 2.616 1.768

> vanim <- names(resby)
> ind <- match(vanim, d2$animal)
> traitement <- d2$traitement[ind]
```

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

```
> tempsperte <- function(d1animal) {
+   interp <- approx(x = d1animal$temps, y = d1animal$poids,
+                    method = "linear", n = 100)
+   seuil <- d1animal$poids[d1animal$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] 6.576 6.717 5.657 6.505 6.364 5.657 6.152 5.727 6.152 6.152    NA 5.232
[13] 5.374 5.727 0.778 1.768 5.798 5.091 4.879 1.061 1.838 5.303 2.616 1.768

> vanim <- names(resby)
> ind <- match(vanim, d2$animal)
> traitement <- d2$traitement[ind]
```

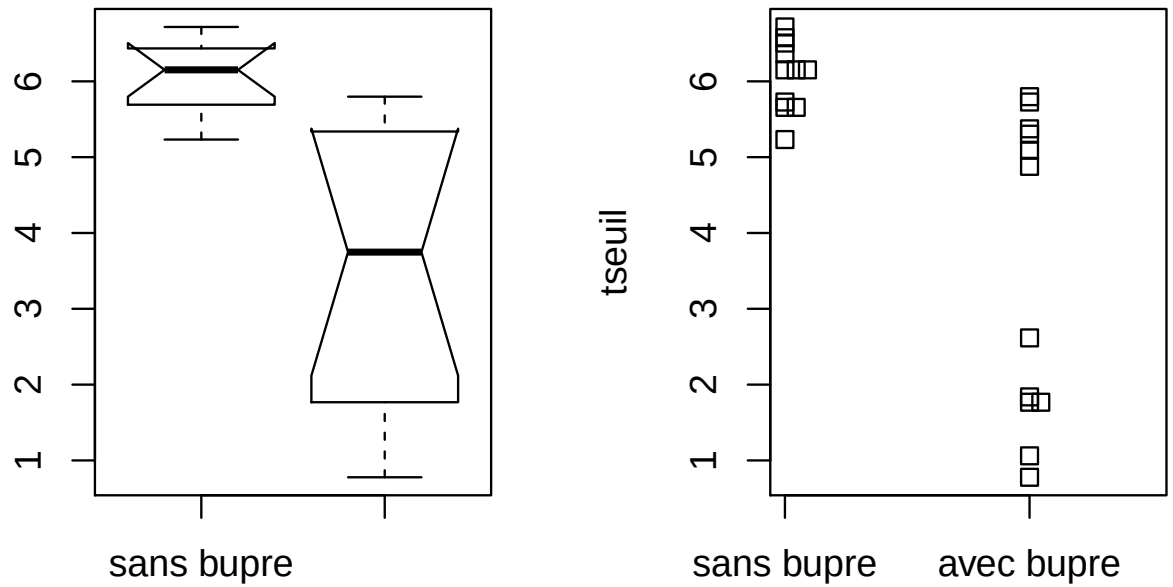
Représentation graphique associée des données

```
> 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, notch = TRUE)
> 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.

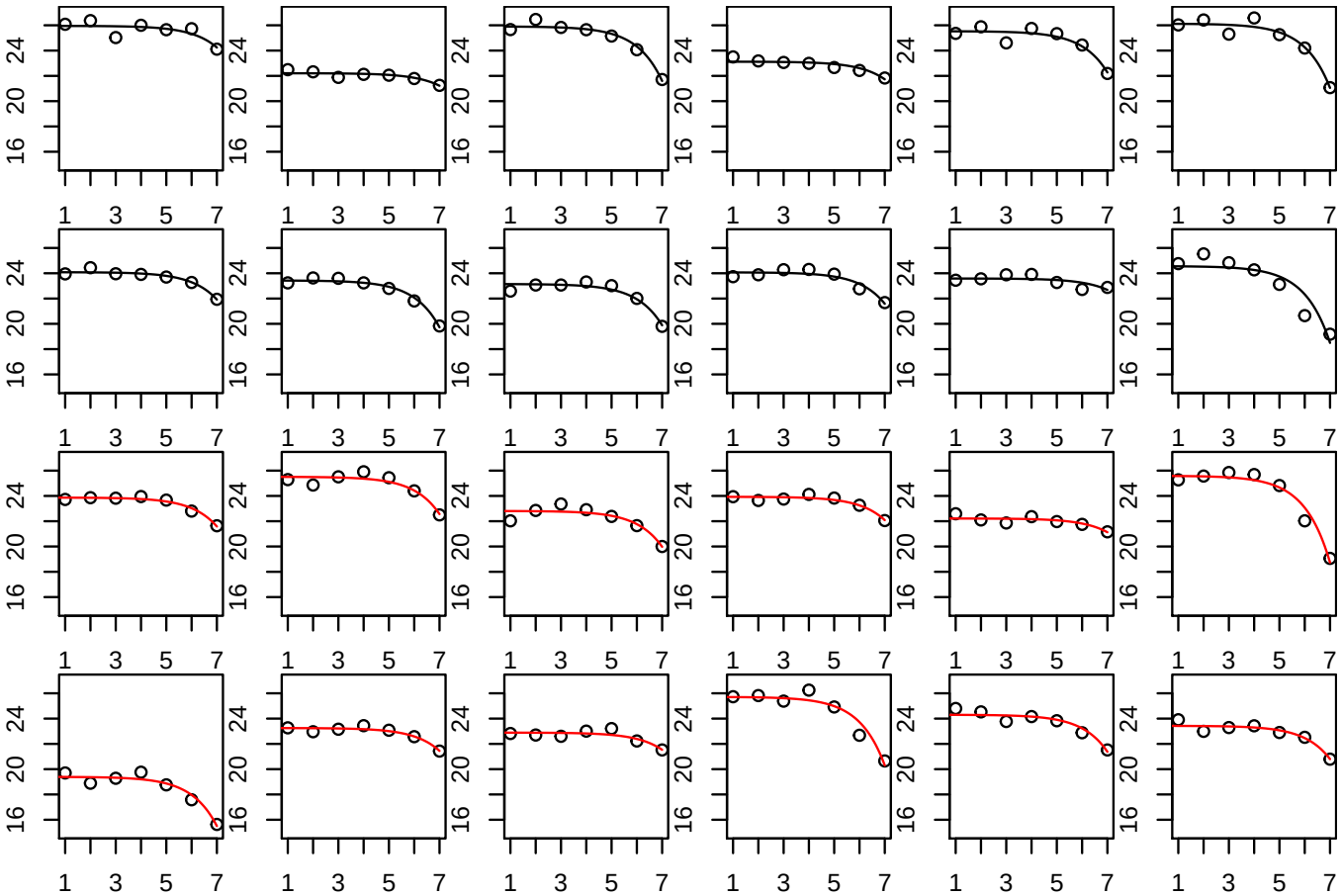
Ajustement d'un modèle paramétrique sur chaque cinétique

Imaginons qu'un modèle soit adapté pour décrire chaque cinétique à partir de J1 (ex. ici avec un modèle exponentiel décroissant)

```
> ## Définition du modèle exponentiel décroissant
> form <- as.formula("poids ~ P0 - b * (exp(temps)  - 1)")
> ## Fonction d'ajustement du modèle sur une cinétique par
> ## régression non linéaire (fonction nls)
> fitmnl <- function(d1animal)
+ {
+   dapJ1 <- d1animal[d1animal$temps >=1, ]
+   # Estimation de valeurs initiales pour les paramètres
+   P0start <- dapJ1$poids[dapJ1$temps == 1]
+   bstart <- (P0start - dapJ1$poids[dapJ1$temps == 7]) / (exp(7) - 1)
+   mnl <- nls(form, start = list(P0 = P0start, b = bstart), data = dap
+   par <- coef(mnl)
+   plot(dapJ1$temps, dapJ1$poids, ylim = c(15, 27))
+   x <- seq(0, 7, length.out = 100)
+   lines(x, par["P0"]- par["b"] * (exp(x)  - 1), col = dapJ1$traitemen
+   return(par["b"])
+ }
```

Application de la fonction sur chaque cinétique

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

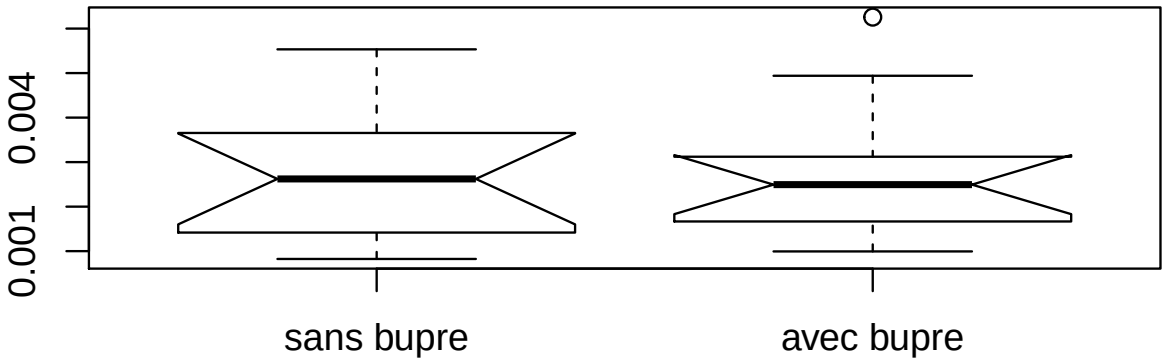


Comparaison entre groupes d'un des paramètres estimés

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, notch = TRUE)
> t.test(bexpo ~ traitement, paired = FALSE, var.equal = TRUE)$p.value
```

```
[1] 0.946
```



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**.