TP R : Effets fixes et aléatoires

C. Preda

le 27 Mars 2018

Objectif du TP

L'objectif de ce TP est d'introduire les effets aléatoire dans un modèle d'analyse de la variance et plus généralement dans un modèle de régression. On fait appel à ce type d'effets (technique) dans le contexte des mesures répétées ou l'hypothèse d'indépendance des observations n'est plus valide. Nous allons illustrer cela de manière progressive à l'aide d'un exemple. Les packages qu'on va utiliser sont **nlme** et **lme4**.

Présentation du problème et des données.

Il s'agit de voir si le passage du sucre dans le sang (absorbtion) est different chez les patients obeses et chez les patients controle (non-obeses). Pour cela, on realise le plan d'expérience suivant : on forme un échantillon aléatoire de 13 patients obeses et un échantillon aléatoire de 20 patients controle. A chaque patient on administre un qunatité fixé de sucre (10mg) et on regarde ensuite la glycémie (unité de mesure non-précisée) à 8 instantes de temps différentes : à $t_0 = 0$ (avant la dose du sucre), à $t_1 = 0.5$ heures après la prise de sucre, et puis à $t_2 = 1$h, $t_3 = 1.5$h, $t_4 = 2$h, $t_5 = 3$h, $t_6 = 4$h et $t_7 = 5$h.

La base de données est disponible en format *csv* (séparateur ";") à l'adresse :

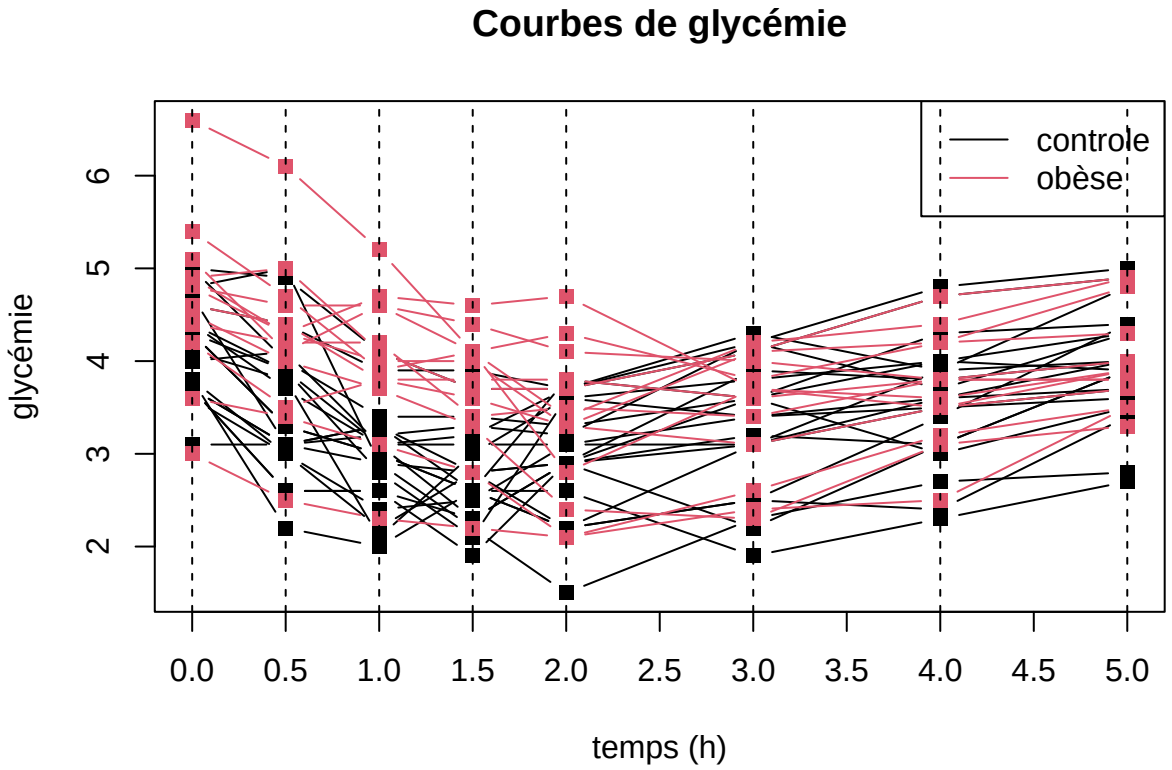
http://math.univ-lille1.fr/~preda/GIS5/glycemie.csv

Remarquez la présence d'un en-tete pour les noms de variables dont un identificateur pour chaque patient (id). Pour des raisons qui seront évidentes plus tard, n'utilisez pas cette colonne comme row.names lors de la lecture des données.

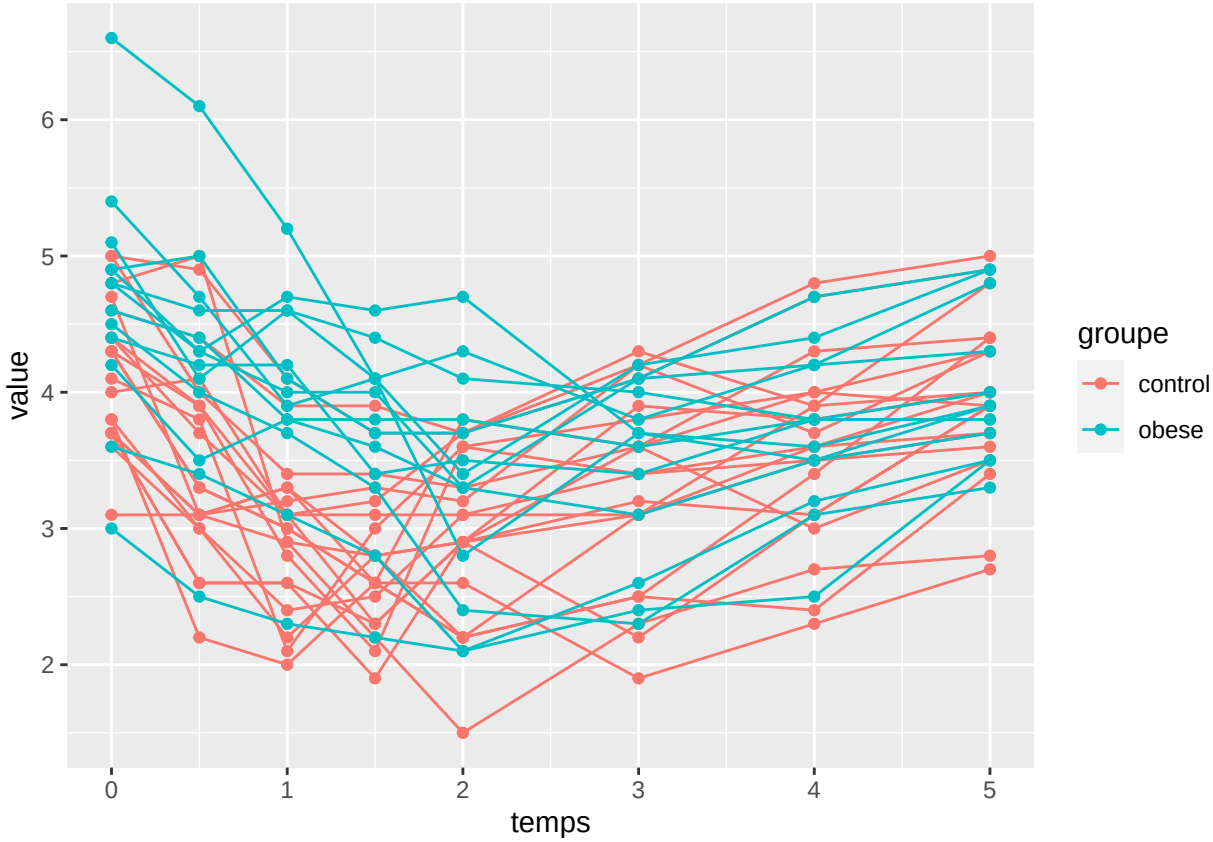
Voici quelques tâches qui vous sont démandées:

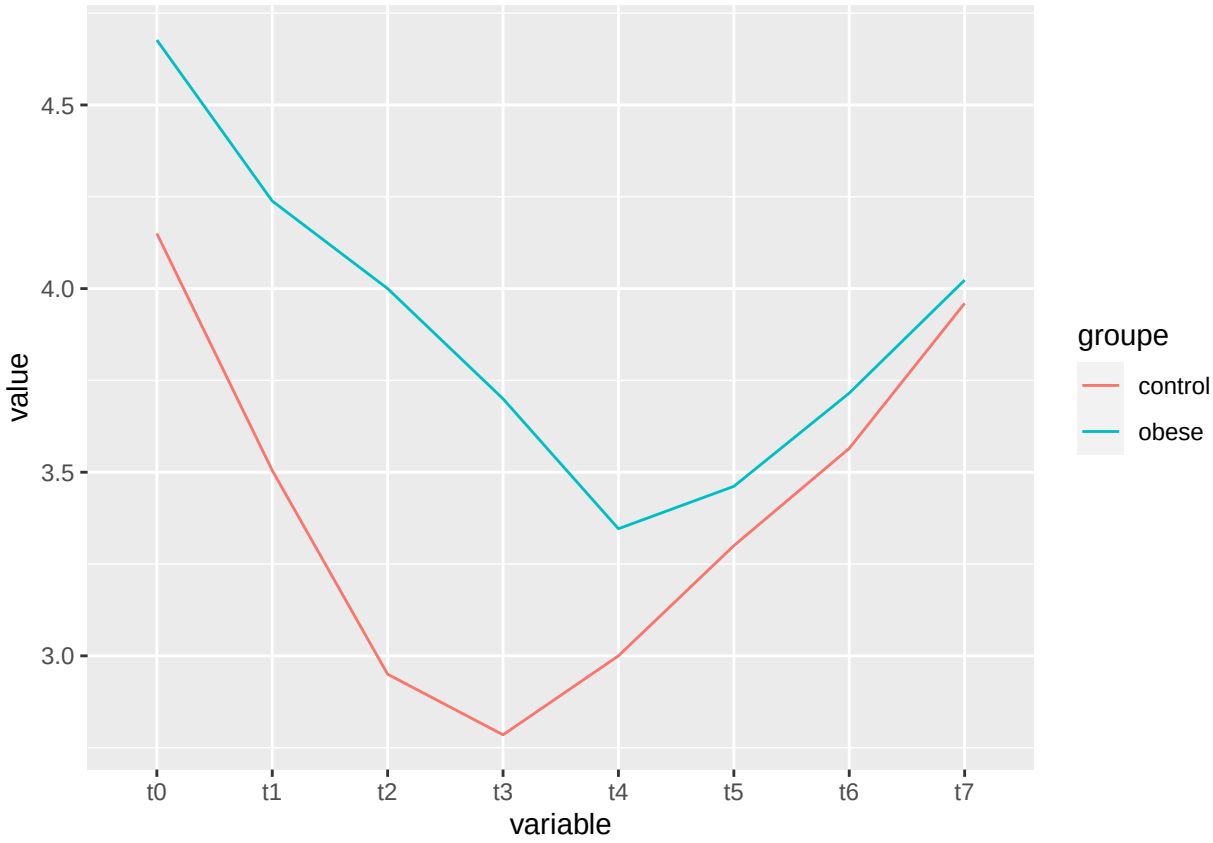
1. Statistiques descriptives pour chaque variable temps. Preciser notamment la moyenne et l'écart-type.

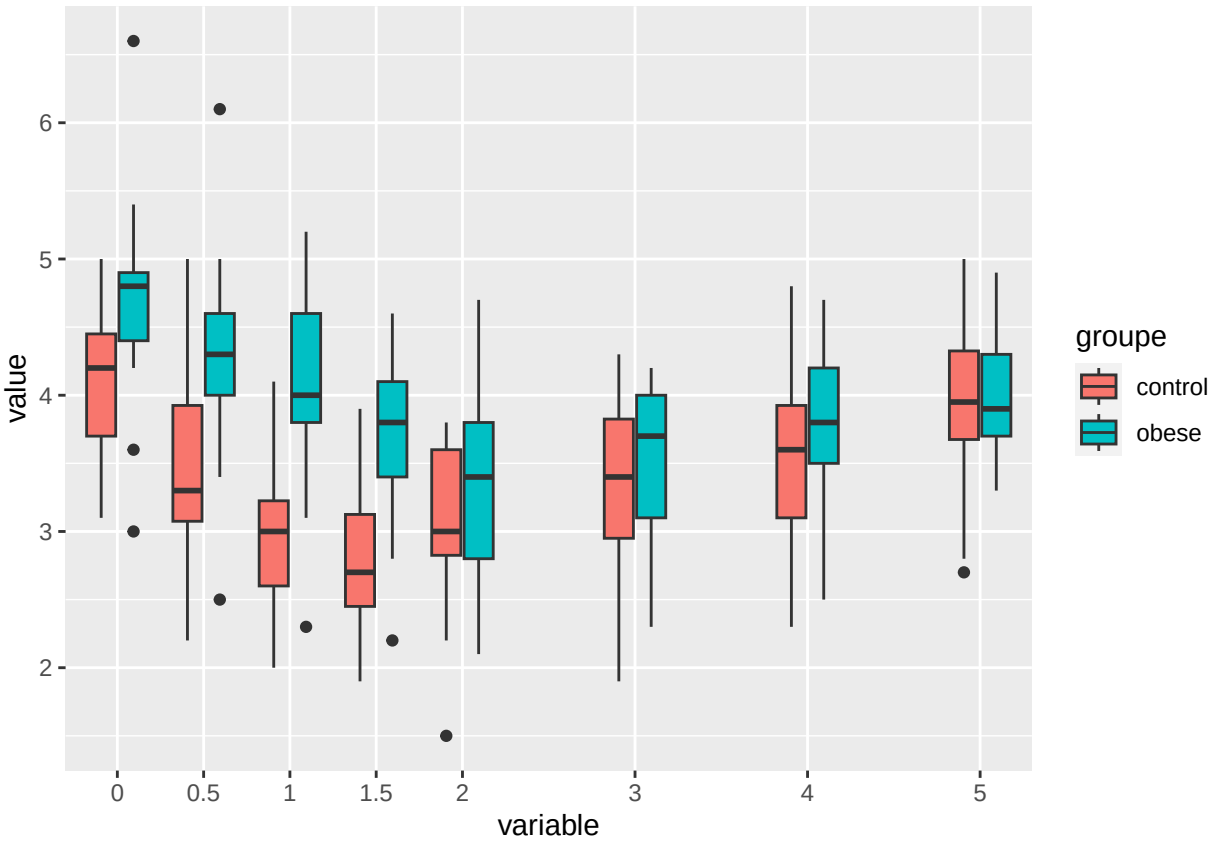
2. Représentation graphique des données. On attend quelques choses du genre :



Exemple avec ggplot2. L'important est d'avoir un dataframe au bon format où les informations pour grouper ou colorer les individus sont stockées dans une même colonne. On appelle cela le format long et il s'obtient avec la fonction melt par exemple.

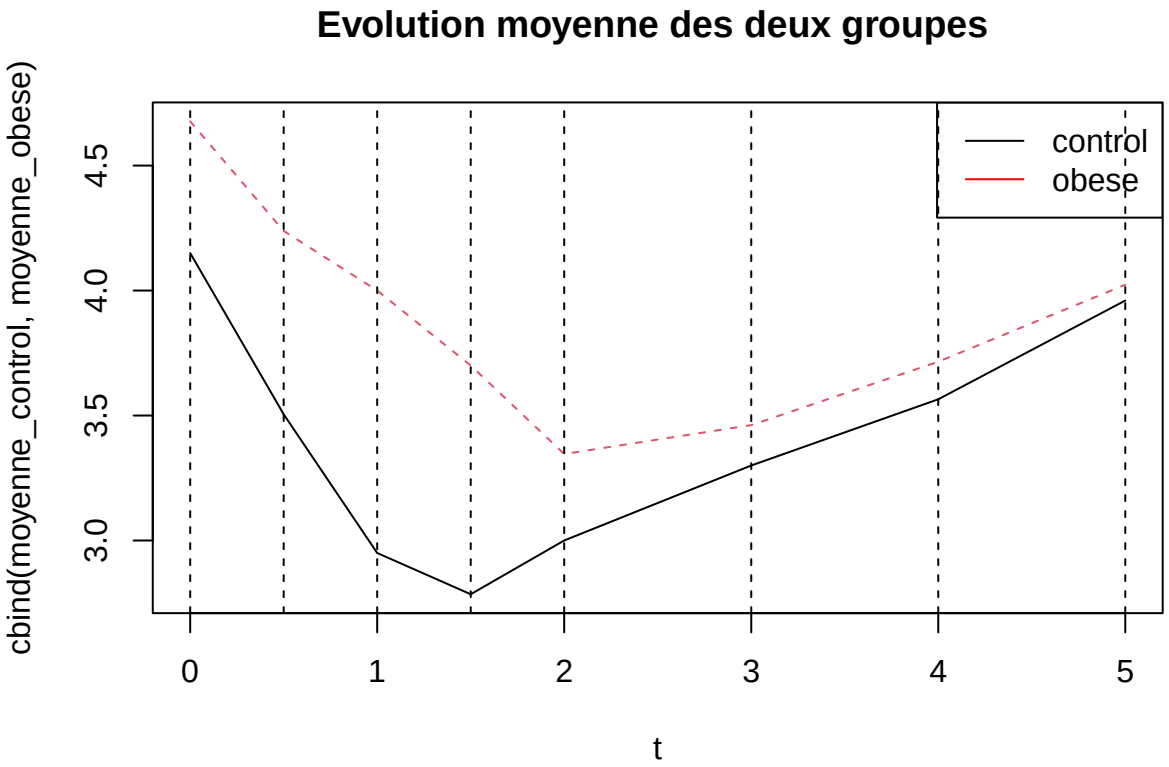

value
6
5
4
3
2
0
1
2
3
4
5
temps
groupe
control
obese


value
4.5
4.0
3.5
3.0
t0
t1
t2
t3
t4
t5
t6
t7
variable
groupe
control
obese

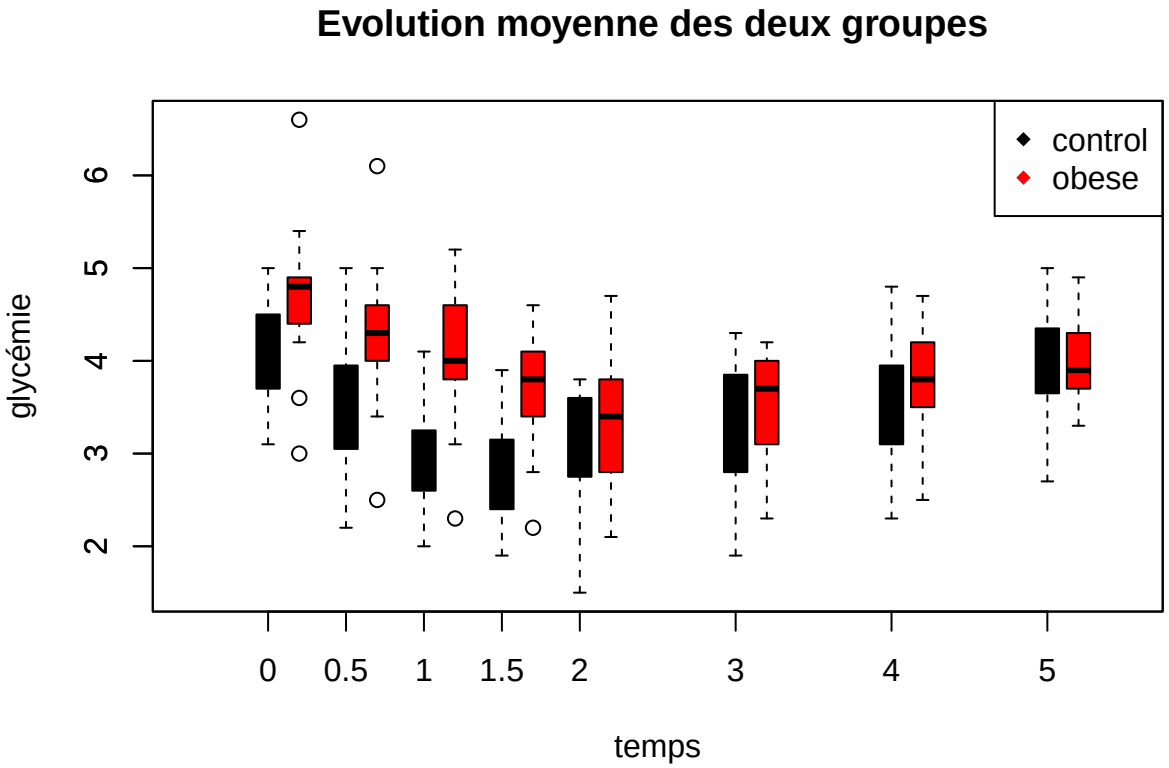


3. Comparaison des deux groupes par l'évolution moyenne de la glycémie On s'interesse à l'évolution moyenne de la glycémie par groupe.

Réaliser les graphiques suivants :



ou encore :



Pour ce dernier graphique, on utilisera surtout les parametres *boxwex*, *at*, *boxfill* et *names* de la fonction *boxplot*.

Elevons le niveau de l'analyse statistiques (et de la discussion) maintenant.

4. Les premièrs tests statistiques pour comparer les groupes. Pour chaque temps, comparer les deux groupes selon le niveaux moyen de la glycémie.

Note: Selon que l'hypothèse de normalité des données est vérifiée (à l'aide du .4942test de Shapiro - fonction *shapiro.test*, on utilisera le test de Student (fonction *t.test*) ou, dans le cas contraire, le test de Wilcoxon (fonction *wilcox.test*). Pour rappel, les tests de Student et Wilcoxon permetent de vérifier l'hypothèse nulle selon laquelle les deux groupes ont la meme esperance de la glycémie. Le test de Wilcoxon est un test non-paramétrique - c'est-à-dire que son utlisisation n'est pas conditionnée par la loi des donnéees.

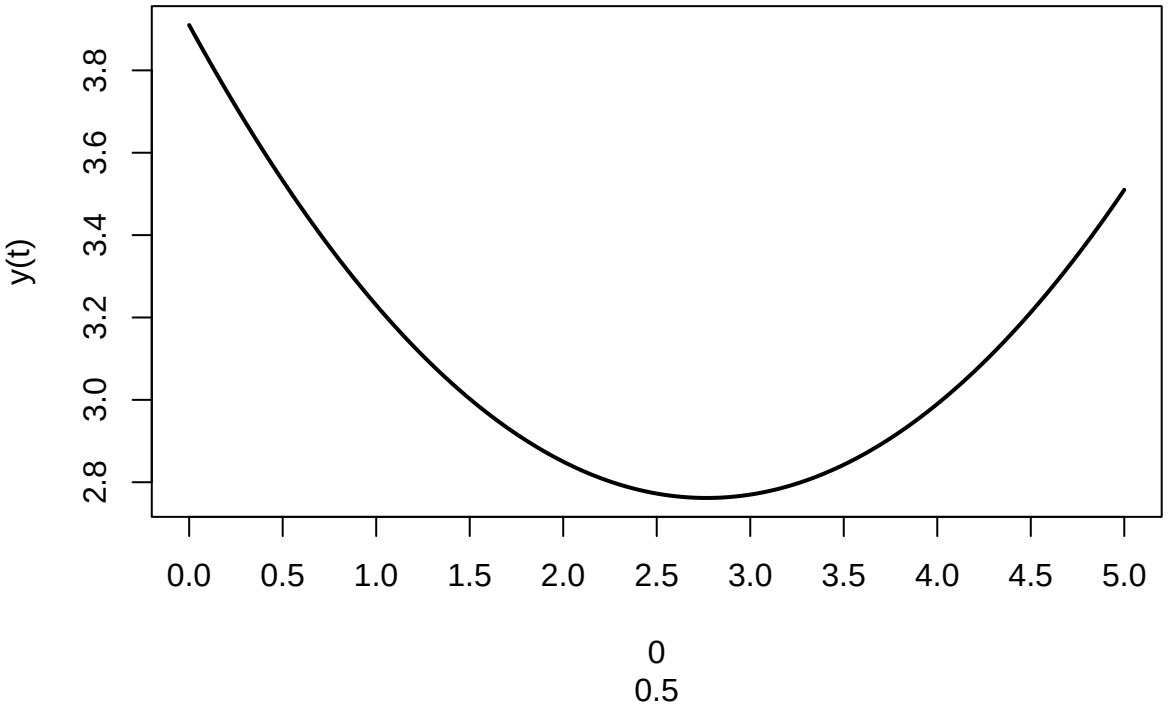
Au vue des résultats numériques (et graphiques) il y a donc des differences significatives entre les deux groupes. Alons plus en détail.

```
FALSE  ------- Temps t0
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 1.9919, df = 17.673, p-value = 0.06206
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  -0.02957173  1.08341788
FALSE sample estimates:
FALSE mean of x mean of y
FALSE  4.676923  4.150000
FALSE
FALSE  ------- Temps t1
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 2.5285, df = 23.255, p-value = 0.01868
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  0.1337633 1.3331598
FALSE sample estimates:
FALSE mean of x mean of y
FALSE  4.238462  3.505000
FALSE
FALSE  ------- Temps t2
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 4.4043, df = 20.245, p-value = 0.0002666
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  0.55308 1.54692
FALSE sample estimates:
FALSE mean of x mean of y
FALSE      4.00      2.95
FALSE
FALSE  ------- Temps t3
FALSE
FALSE   Welch Two Sample t-test
FALSE
```

```
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 4.205, df = 21.78, p-value = 0.0003724
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  0.4634702 1.3665298
FALSE sample estimates:
FALSE mean of x mean of y
FALSE     3.700     2.785
FALSE
FALSE  ------- Temps t4
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 1.3039, df = 20.922, p-value = 0.2064
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  -0.2060419  0.8983496
FALSE sample estimates:
FALSE mean of x mean of y
FALSE  3.346154  3.000000
FALSE
FALSE  ------- Temps t5
FALSE Warning in wilcox.test.default(d[[paste0("t", i - 1)]][d$groupe == "obese"], :
FALSE impossible de calculer la p-value exacte avec des ex-aequos
FALSE
FALSE   Wilcoxon rank sum test with continuity correction
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE W = 149, p-value = 0.4942
FALSE alternative hypothesis: true location shift is not equal to 0
FALSE
FALSE  ------- Temps t6
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 0.6788, df = 28.123, p-value = 0.5028
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
FALSE  -0.3033420  0.6041112
FALSE sample estimates:
FALSE mean of x mean of y
FALSE  3.715385  3.565000
FALSE
FALSE  ------- Temps t7
FALSE
FALSE   Welch Two Sample t-test
FALSE
FALSE data:  d[[paste0("t", i - 1)]][d$groupe == "obese"] and d[[paste0("t", i - 1)]][d$groupe == "conti
FALSE t = 0.31122, df = 27.889, p-value = 0.758
FALSE alternative hypothesis: true difference in means is not equal to 0
FALSE 95 percent confidence interval:
```

```
FALSE  -0.3521654  0.4783192
FALSE sample estimates:
FALSE mean of x mean of y
FALSE  4.023077  3.960000
```

Un modèle de régression quadratique L'évolution de la glycémie en fonction du temps semble une fonction quadratique, c'est à dire une courbe (parabole) en "U" :



$$y(t) = a + bt + ct^2 + \varepsilon$$

avec a, b et c des coefficients et ε une erreur aléatoire.

Estimer un modèle de régression quadratique pour chaque groupe séparement. La variable explicative est donc le temps. Il faudrait donc construire cette variable. On transformera donc ces données initiales (dites en format *large*) en format *long* :

```
FALSE     groupe id temps   Y
FALSE 1  control  1   0.0 4.3
FALSE 2  control  1   0.5 3.3
FALSE 3  control  1   1.0 3.0
FALSE 4  control  1   1.5 2.6
FALSE 5  control  1   2.0 2.2
FALSE 6  control  1   3.0 2.5
FALSE 7  control  1   4.0 3.4
FALSE 8  control  1   5.0 4.4
FALSE 9  control  2   0.0 3.7
FALSE 10 control  2   0.5 2.6
```

Ceci se réalise facilemnt grace à la fonction *reshape*. Voici le code R:

```
dlong <- reshape(
  data = d,
  varying = list(names(d)[3:10]),
```

```r
  # idvar = c("id", "groupe"),
  idvar = c("id"),
  direction = "long", v.names = "Y"
)

head(dlong) # pour voir le resultat brut !
# Arangeons un peu cela :

names(dlong) <- c("groupe", "id", "temps", "Y")
dlong <- dlong[order(dlong$id), ]
row.names(dlong) <- 1:nrow(dlong)

# mettons les vrais temps
dlong[dlong$temps == 1, c("temps")] <- 0
dlong[dlong$temps == 2, c("temps")] <- 0.5
dlong[dlong$temps == 3, c("temps")] <- 1
dlong[dlong$temps == 4, c("temps")] <- 1.5
dlong[dlong$temps == 5, c("temps")] <- 2
dlong[dlong$temps == 6, c("temps")] <- 3
dlong[dlong$temps == 7, c("temps")] <- 4
dlong[dlong$temps == 8, c("temps")] <- 5

head(dlong, 10)
```

Maintenant on peut réaliser un modèle de régression quadratique pour les controles, par exemple.

```r
mq_controle <- lm(Y ~ temps + I(temps^2), data = dlong[dlong$groupe == "control", ])
summary(mq_controle)
```

```
Call:
lm(formula = Y ~ temps + I(temps^2), data = dlong[dlong$groupe ==
    "control", ])

Residuals:
     Min       1Q   Median       3Q      Max
-1.45631 -0.44145 -0.05631  0.48336  1.47861

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.88111    0.11377  34.114  < 2e-16 ***
temps       -0.80512    0.11624  -6.926 1.05e-10 ***
I(temps^2)   0.17136    0.02249   7.618 2.27e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6525 on 157 degrees of freedom
Multiple R-squared:  0.2774,    Adjusted R-squared:  0.2682
F-statistic: 30.14 on 2 and 157 DF,  p-value: 8.377e-12
```

```r
print(shapiro.test(mq_controle$residuals)) # tester la normalité des residus
```

```
    Shapiro-Wilk normality test

data:  mq_controle$residuals
W = 0.99215, p-value = 0.5322
```

```r
library(lmtest)
print(bptest(mq_controle)) # tester homoscedasticité des residus
```

```
    studentized Breusch-Pagan test

data:  mq_controle
BP = 0.50688, df = 2, p-value = 0.7761
```

```r
print(dwtest(mq_controle)) # tester l'autocorrelation des residus
```

```
    Durbin-Watson test

data:  mq_controle
DW = 0.87339, p-value = 2.636e-13
alternative hypothesis: true autocorrelation is greater than 0
```

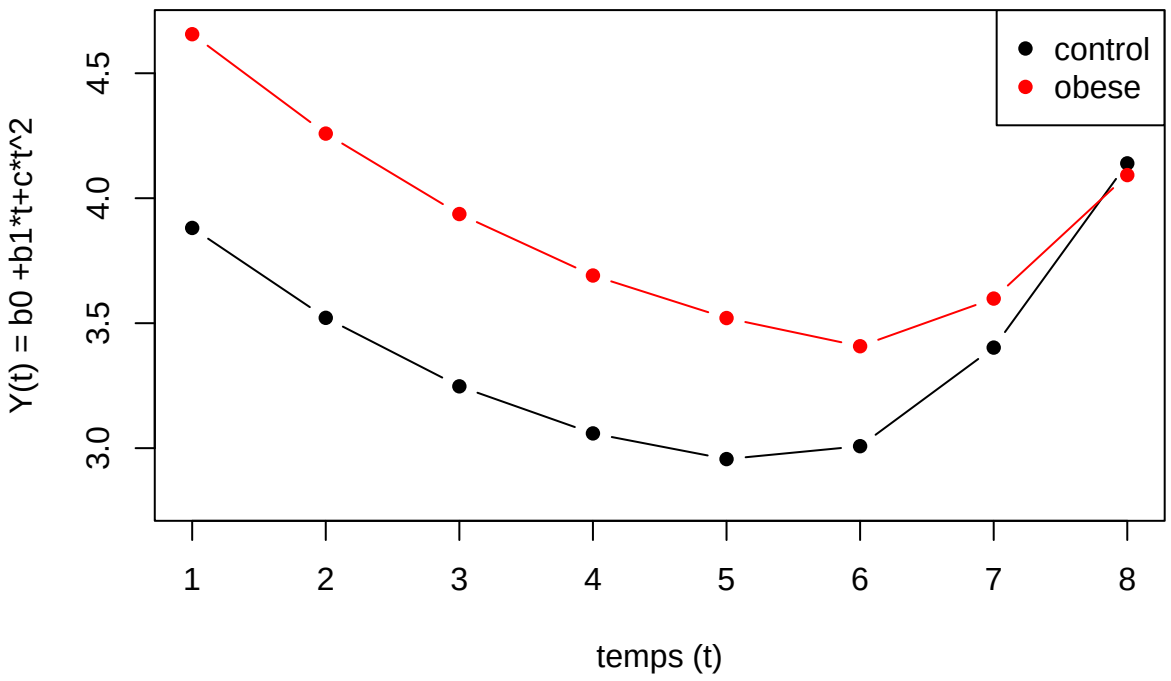
On a donc un problème d'autocorrelation des residus! Corrélationn due au temps!

Réaliser le meme modèle pour le groupe des obseses et comparer les deux modèles à l'aide des coefficients et des leurs intervalles de confiance. Tracer les deux fonctions de régression sur le meme graphique.

```r
mq_obese <- lm(Y ~ temps + I(temps^2), data = dlong[dlong$groupe == "obese", ])
summary(mq_obese)
```

```
Call:
lm(formula = Y ~ temps + I(temps^2), data = dlong[dlong$groupe ==
    "obese", ])

Residuals:
     Min       1Q   Median       3Q      Max
-1.75853 -0.29633  0.02538  0.41733  1.94380

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.65620    0.15373  30.288  < 2e-16 ***
temps       -0.87119    0.15706  -5.547 2.34e-07 ***
I(temps^2)   0.15169    0.03039   4.991 2.51e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7108 on 101 degrees of freedom
Multiple R-squared:  0.2429,    Adjusted R-squared:  0.2279
F-statistic:  16.2 on 2 and 101 DF,  p-value: 7.893e-07
```

```r
# tracage des deux fonctions de regression:

plot(predict(mq_controle, newdata = data.frame(temps = t)),
  type = "b", col = "black", pch = 16,
  ylim = c(min(c(moyenne_control, moyenne_obese)), max(c(max(c(moyenne_control, moyenne_obese))))),
  main = "Fonctions de régression pour les deux groupes",
  ylab = "Y(t) = b0 +b1*t+c*t^2", xlab = "temps (t)"
)

lines(predict(mq_obese, newdata = data.frame(temps = t)), type = "b", col = "red", pch = 16)

legend("topright", c("control", "obese"), pch = c(16, 16), col = c("black", "red"))
```



Comparaison des fonctions de régression:

```
print(summary(mq_controle)$coefficients)
```

```
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  3.8811149 0.11377062 34.113508 1.677238e-74
temps       -0.8051201 0.11623791 -6.926485 1.045965e-10
I(temps^2)   0.1713587 0.02249273  7.618401 2.269600e-12
```

```
print(summary(mq_obese)$coefficients)
```

```
              Estimate Std. Error   t value    Pr(>|t|)
(Intercept)  4.6561992 0.15372838 30.288481 1.73557e-52
temps       -0.8711851 0.15706222 -5.546751 2.34290e-07
I(temps^2)   0.1516886 0.03039248  4.990989 2.51214e-06
```

Interpréter le modèle :

```
mq <- lm(Y ~ groupe * (temps + I(temps^2)), data = dlong)
summary(mq)
```

```
Call:
lm(formula = Y ~ groupe * (temps + I(temps^2)), data = dlong)

Residuals:
     Min       1Q   Median       3Q      Max
-1.75853 -0.42139 -0.00511  0.44205  1.94380

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.88111    0.11786  32.931  < 2e-16 ***
groupeobese           0.77508    0.18777   4.128 4.95e-05 ***
```

```
temps                   -0.80512    0.12041  -6.686 1.41e-10 ***
I(temps^2)               0.17136    0.02330   7.354 2.55e-12 ***
groupeobese:temps       -0.06606    0.19185  -0.344    0.731
groupeobese:I(temps^2) -0.01967    0.03712  -0.530    0.597
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6759 on 258 degrees of freedom
Multiple R-squared:  0.3274,    Adjusted R-squared:  0.3143
F-statistic: 25.11 on 5 and 258 DF,  p-value: < 2.2e-16
```

Écrire ce modèle et comparer avec les deux modèles précédents. Qu'observez-vous ?

Est-ce un modèle valide ?

```
shapiro.test(mq$residuals) # p-value = 0.6341 ok - normalité
```

```
    Shapiro-Wilk normality test

data:  mq$residuals
W = 0.99546, p-value = 0.6341
```

```
bptest(mq) # p-value = 0.3461 ok - homoscédasticité
```

```
    studentized Breusch-Pagan test

data:  mq
BP = 5.6093, df = 5, p-value = 0.3461
```

```
dwtest(mq$residuals ~ dlong$temps) # p-value < 2.2e-16 NON ! residus autocoréllés (y(t) est corellé ave
```

```
    Durbin-Watson test

data:  mq$residuals ~ dlong$temps
DW = 0.72787, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

Réalisons un modèle mixte basé sur mq. On commence avec la variante la plus simple : intercept aléatoire :

```
library(nlme)

mq_mixte1 <- lme(Y ~ groupe * (temps + I(temps^2)), random = ~ 1 | id, data = dlong, method = "ML")
summary(mq_mixte1)
```

```
Linear mixed-effects model fit by maximum likelihood
  Data: dlong
       AIC      BIC    logLik
  420.2566 448.8642 -202.1283

Random effects:
 Formula: ~1 | id
        (Intercept)  Residual
StdDev:    0.495311 0.4485368

Fixed effects:  Y ~ groupe * (temps + I(temps^2))
                            Value  Std.Error  DF   t-value p-value
(Intercept)              3.881115 0.13715062 227 28.298194  0.0000
groupeobese              0.775084 0.21851591  31  3.547038  0.0013
```

```
temps                   -0.805120 0.08082548 227 -9.961216  0.0000
I(temps^2)               0.171359 0.01564022 227 10.956285  0.0000
groupeobese:temps       -0.066065 0.12877561 227 -0.513024  0.6084
groupeobese:I(temps^2) -0.019670 0.02491885 227 -0.789366  0.4307
 Correlation:
                        (Intr) gropbs temps  I(t^2) grpbs:
groupeobese             -0.628
temps                   -0.463  0.291
I(temps^2)               0.386 -0.242 -0.962
groupeobese:temps        0.291 -0.463 -0.628  0.604
groupeobese:I(temps^2) -0.242  0.386  0.604 -0.628 -0.962

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.57947754 -0.61445174 -0.01557856  0.59624584  3.12335932

Number of Observations: 264
Number of Groups: 33
```

```
## visualiser les effets aléatoirs (alpha_i)
fixed.effects(mq_mixte1)
```

```
          (Intercept)            groupeobese                  temps
           3.88111489             0.77508430            -0.80512013
           I(temps^2)      groupeobese:temps groupeobese:I(temps^2)
           0.17135866            -0.06606499            -0.01967011
```

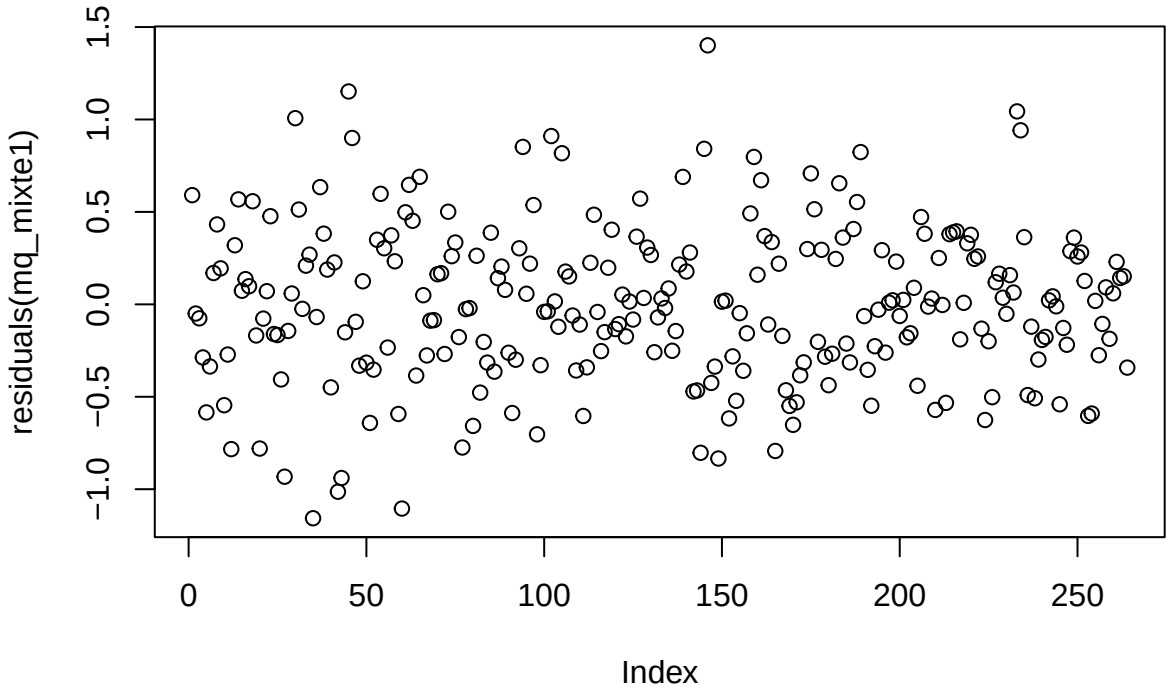
```
## validité du modele
shapiro.test(mq_mixte1$residuals)
```

```
    Shapiro-Wilk normality test

data:  mq_mixte1$residuals
W = 0.99467, p-value = 0.06364
```

```
dwtest(residuals(mq_mixte1) ~ dlong$temps)
```

```
    Durbin-Watson test

data:  residuals(mq_mixte1) ~ dlong$temps
DW = 1.4638, p-value = 5.672e-06
alternative hypothesis: true autocorrelation is greater than 0
```

```
plot(residuals(mq_mixte1))
```



Réalisons un modèle mixte basé sur mq avec intercept et pente aléatoires.

```
mq_mixte2 <- lme(Y ~ groupe * (temps + I(temps^2)), random = ~ temps | id, data = dlong)

summary(mq_mixte2)
```

```
Linear mixed-effects model fit by REML
  Data: dlong
       AIC      BIC    logLik
  442.1909 477.7205 -211.0955

Random effects:
 Formula: ~temps | id
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 0.58225632 (Intr)
temps       0.09931774 -0.479
Residual    0.41944721

Fixed effects:  Y ~ groupe * (temps + I(temps^2))
                          Value  Std.Error  DF    t-value p-value
(Intercept)             3.881115 0.14933072 227  25.990063  0.0000
groupeobese             0.775084 0.23792192  31   3.257725  0.0027
temps                  -0.805120 0.07795026 227 -10.328640  0.0000
I(temps^2)              0.171359 0.01445872 227  11.851578  0.0000
groupeobese:temps      -0.066065 0.12419464 227  -0.531947  0.5953
groupeobese:I(temps^2) -0.019670 0.02303643 227  -0.853870  0.3941
 Correlation:
                       (Intr) gropbs temps  I(t^2) grpbs:
groupeobese            -0.628
temps                  -0.496  0.311
I(temps^2)              0.327 -0.206 -0.923
groupeobese:temps       0.311 -0.496 -0.628  0.579
```

```
groupeobese:I(temps^2) -0.206  0.327  0.579 -0.628 -0.923

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.77476306 -0.59537050 -0.03175758  0.54802084  2.82725593

Number of Observations: 264
Number of Groups: 33
```

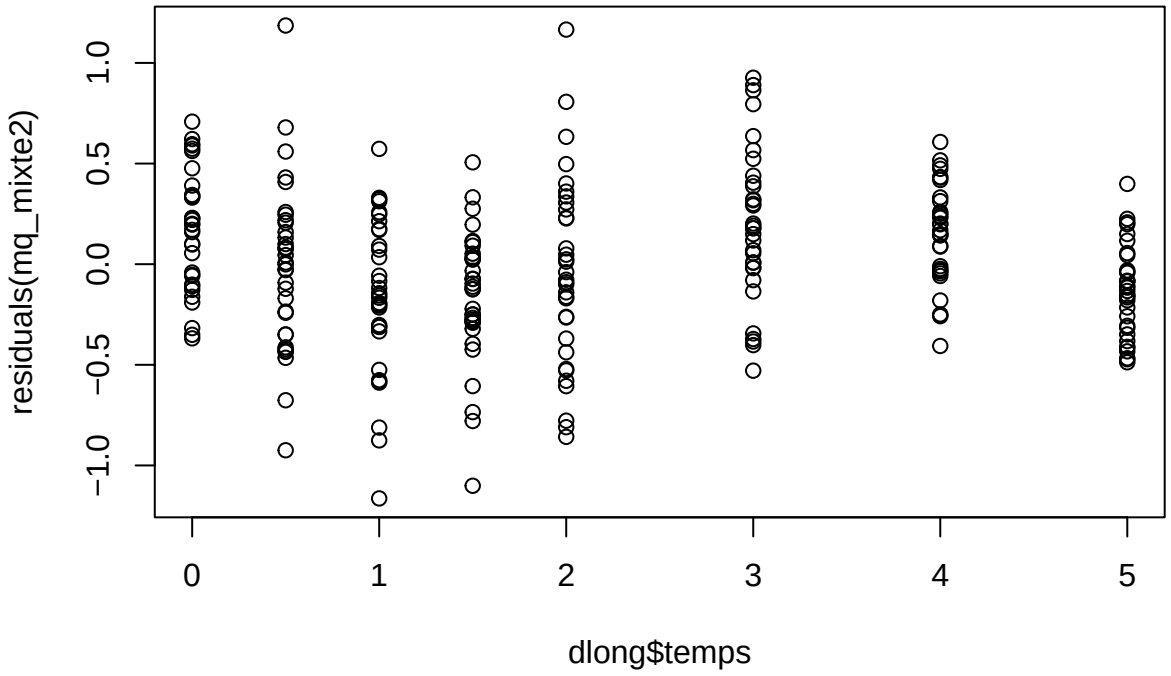
```
dwtest(residuals(mq_mixte2) ~ dlong$temps)
```

```
    Durbin-Watson test

data:  residuals(mq_mixte2) ~ dlong$temps
DW = 1.7414, p-value = 0.01668
alternative hypothesis: true autocorrelation is greater than 0
```

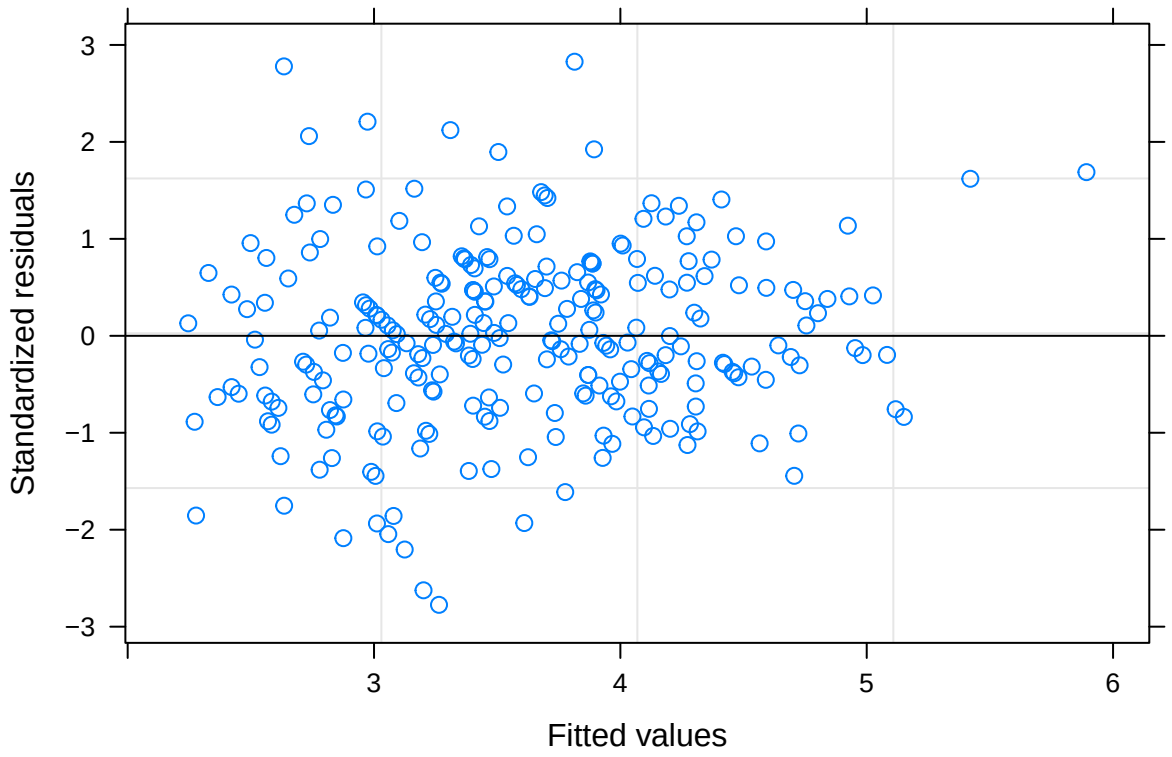
```
bptest(residuals(mq_mixte2) ~ dlong$temps)
```

```
    studentized Breusch-Pagan test

data:  residuals(mq_mixte2) ~ dlong$temps
BP = 2.8746, df = 1, p-value = 0.08998
```

```
plot(residuals(mq_mixte2) ~ dlong$temps)
```




```
plot(mq_mixte2)
```


Standardized residuals
3
2
1
0
−1
−2
−3
3
4
5
6
Fitted values