Analyzing cross-over trials with the package LMMstar

Brice Ozenne

October 22, 2025

In the context of a cross-over trail, this vignette discusses

(i) how mixed model generalizes t-tests

(ii) modeling choices regarding the variance-covariance structure implied by the choice of the repetition variable (period or treatment).

We will use to the following R packages:

```
library(LMMstar)
library(ggplot2)
```

1 Illustrative dataset

The `bloodpressureL` dataset:

```
data(bloodpressureL, package = "LMMstar")
head(bloodpressureL)
```

```
  id sequence treatment period duration
1  1      ABC         A      1      1.9
2  1      ABC         B      2      2.9
3  1      ABC         C      3      4.3
4  2      ABC         A      1      1.4
5  2      ABC         B      2      2.3
6  2      ABC         C      3      3.0
```

originates from a cross-over trial comparing the impact of three formulations of a drug on the blood pressure. The study was conducted on 12 male volunteers randomly divided into tree groups (`sequence`) and receiving each of the three formulations (`treatment`) with a wash-out period of one week. The outcome variable is `duration` where a larger duration indicate a better outcome as the blood pressure is under control for a longer time. While there are not missing values in this dataset:

```
sum(is.na(bloodpressureL))
```

```
[1] 0
```

only 3 out of the 6 possible sequences of treatment have been allocated:

```
levels(bloodpressureL$sequence)
```

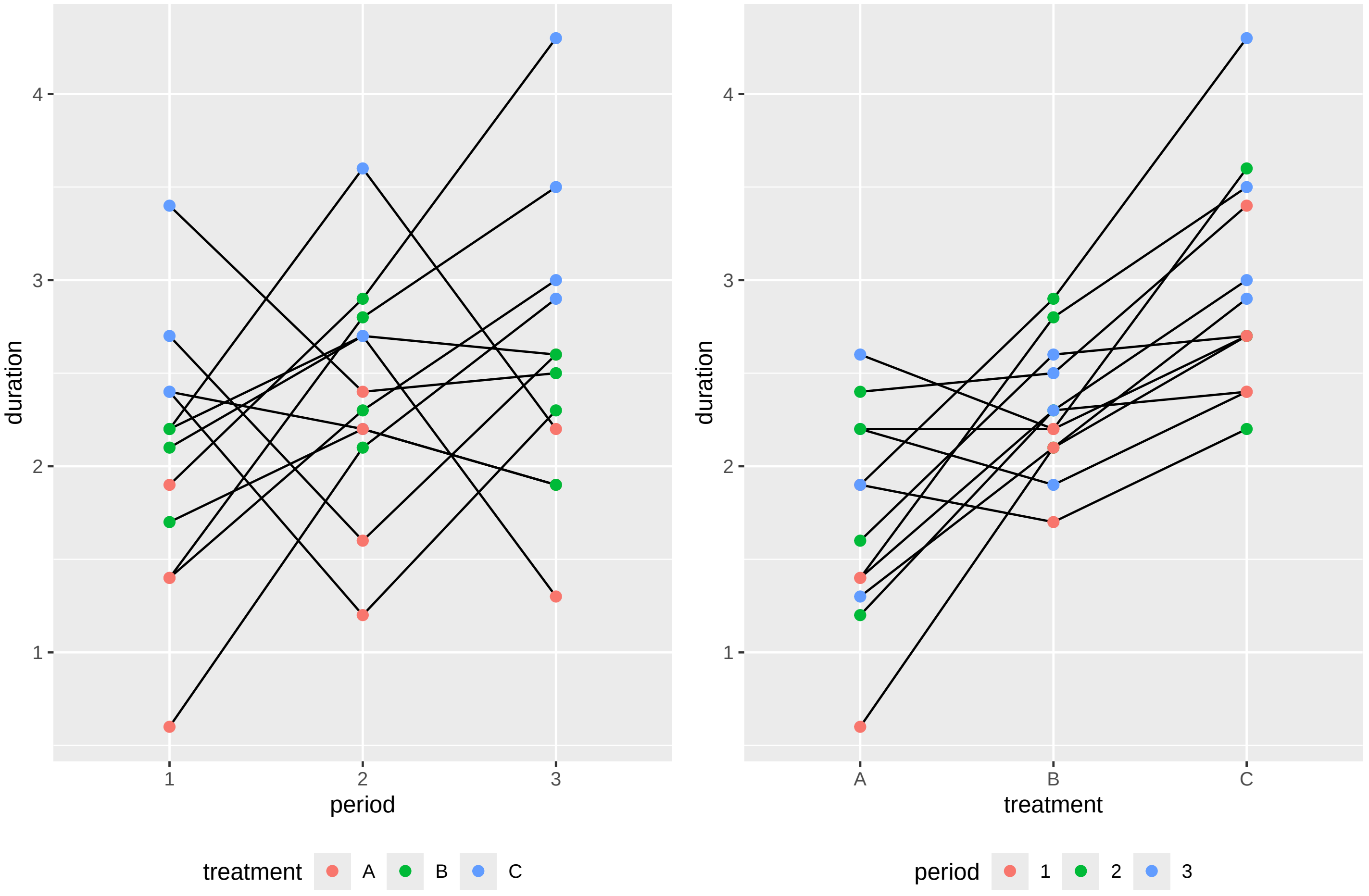
```
[1] "ABC" "BCA" "CAB"
```

A spaghetti plot provides a graphical representation of the dataset, with either period or treatment on the x-axis:

```
ggTime <- ggplot(bloodpressureL, aes(x = period, y = duration, group = id))
ggTime <- ggTime + geom_line() + geom_point(aes(color = treatment), size = 2)
ggTime
```

or

```
ggTreat <- ggplot(bloodpressureL, aes(x = treatment, y = duration, group = id))
ggTreat <- ggTreat + geom_line() + geom_point(aes(color = period), size = 2)
ggTreat
```



2 Matching the t-test results

One can use a paired t-test to assess the treatment effect when there is not missing values and no covariate to adjusted on (in particular no period effect). It is easier to carry-out with the wide format of the dataset:

```
bloodpressureW <- reshape(bloodpressureL, direction = "wide",
                          idvar = "id", timevar = "treatment",
                          v.names = c("duration","period"))
head(bloodpressureW)
```

```
   id sequence duration.A period.A duration.B period.B duration.C period.C
1   1      ABC        1.9        1        2.9        2        4.3        3
4   2      ABC        1.4        1        2.3        2        3.0        3
7   3      ABC        1.4        1        2.8        2        3.5        3
10  4      ABC        0.6        1        2.1        2        2.9        3
13  5      BCA        2.2        3        2.2        1        3.6        2
16  6      BCA        1.3        3        2.1        1        2.7        2
```

For instance we can compare drug B and A using:

```
t.test(bloodpressureW$duration.B - bloodpressureW$duration.A)
```

```

        One Sample t-test

data:  bloodpressureW$duration.B - bloodpressureW$duration.A
t = 2.9, df = 11, p-value = 0.015
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.13805 1.01195
sample estimates:
mean of x
    0.575
```

To retrieve the same results with a linear mixed model, one can use treatment as indexing the repetitions, i.e., model a treatment specific variance and correlation:

```
eTreat.lmm2tt <- lmm(duration ~ treatment, repetition = ~treatment|id, data = bloodpressureL)
model.tables(eTreat.lmm2tt)
```

```
            estimate      se     df   lower  upper    p.value
(Intercept)   1.7250 0.16703 11.002 1.35739 2.0926 5.3415e-07
treatmentB    0.5750 0.19853 10.998 0.13804 1.0120 1.4542e-02
treatmentC    1.2583 0.22578 10.998 0.76137 1.7553 1.6701e-04
```

⚠ using period as the repetition variable, i.e., modeling a period specific variance and correlation, would lead to different estimates:

```
ePeriod.lmm2tt <- lmm(duration ~ treatment, repetition = ~period|id, data = bloodpressureL)
model.tables(ePeriod.lmm2tt)
```

```
            estimate      se      df   lower  upper    p.value
(Intercept)  1.68755 0.20349  4.7145 1.15478 2.2203 5.5048e-04
treatmentB   0.58766 0.19895 14.4584 0.16223 1.0131 1.0173e-02
treatmentC   1.16557 0.19654 11.9624 0.73718 1.5939 7.0104e-05
```

As shown in appendix A.2, this mixed model considers both the variable treatment and period when deciding how much each observation contributes to the estimation of a given parameter. On one side, it makes sense that an observation taken at a period with large variance should contribute less to parameter estimation compared to an observation taken at a period with low variance. On the other side, it can be suprising that treatment B outcomes can contribute to the estimation of treatment A. This is however not the case in absence of period effects since the weights sum to 0 for treatment B and C when estimating the intercept.

⚠ using a random intercept model instead would lead to the same estimate but a different p-value:

```
eTreat.RI <- lmm(duration ~ treatment + (1|id), data = bloodpressureL)
model.tables(eTreat.RI)
```

```
            estimate      se     df   lower   upper    p.value
(Intercept)   1.7250 0.15192 29.483 1.41452 2.03548 2.7569e-12
treatmentB    0.5750 0.18673 22.000 0.18774 0.96226 5.4846e-03
treatmentC    1.2583 0.18673 22.000 0.87107 1.64560 8.9931e-07
```

as it makes more restrictive assumptions (homoschedasticity, equal correlation).

3 Accounting for a period effect

A natural extension of the t-test to adjust for a possible period effect on the average outcome is to consider the corresponding mixed model (i.e. treatment as repetition) and add period in the mean model:

```
eTreat.lmm <- lmm(duration ~ treatment + period, repetition = ~treatment|id,
                  data = bloodpressureL)
model.tables(eTreat.lmm)
```

```
            estimate      se      df     lower   upper    p.value
(Intercept)  1.54915 0.16604 13.9154  1.192831 1.90547 2.3014e-07
treatmentB   0.57500 0.16768  9.3858  0.198043 0.95196 7.0688e-03
treatmentC   1.25833 0.17889  9.2795  0.855515 1.66115 5.2215e-05
period2      0.19991 0.12698  4.0388 -0.151307 0.55112 1.8984e-01
period3      0.32764 0.12109  4.8784  0.014028 0.64125 4.3588e-02
```

Here because the design is balanced in term of period across treatments, we obtain the same estimates for the difference in treatment effect as if we do not adjust for period. However the estimated mean outcome under each treatment (say treatment A) now depends on all observations (and not only observations under treatment A). See appendix A.3 for details.

4 What if there is a baseline measurement?

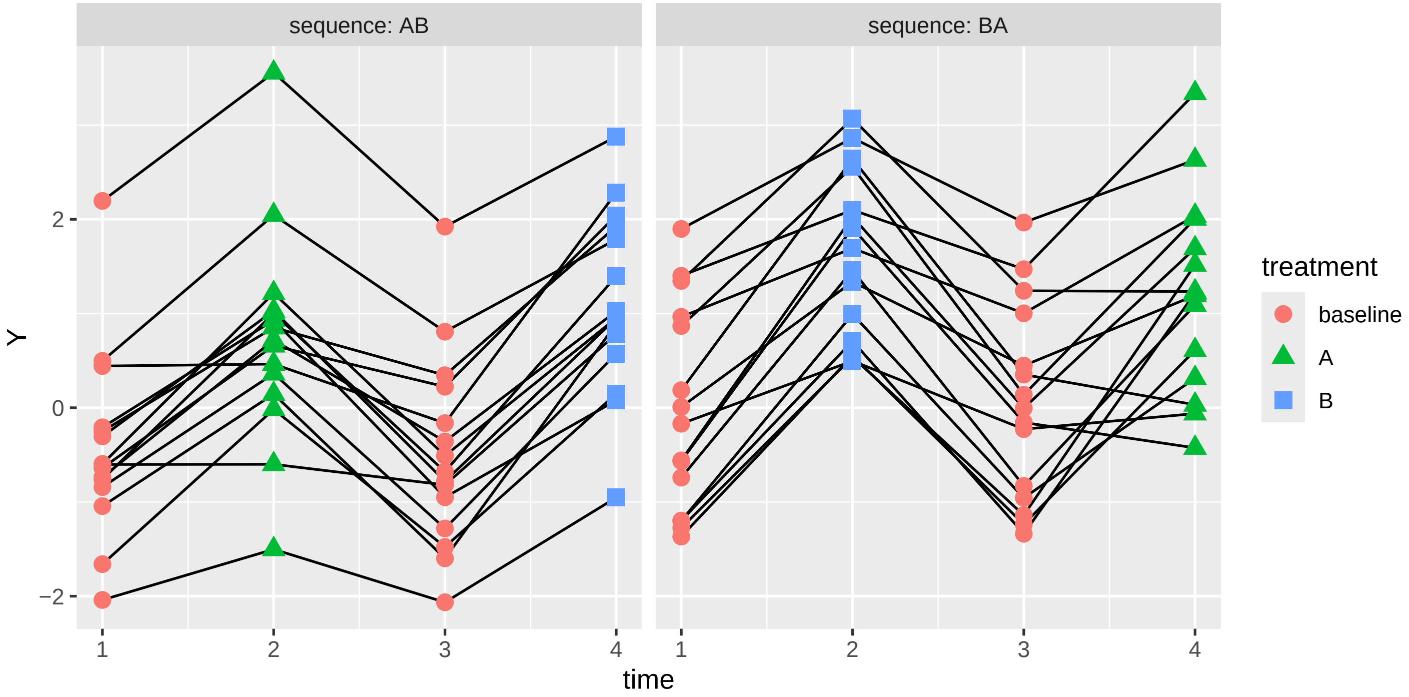
Consider now another study design where all patients have a baseline measurement before receiving each treatment. As an illustrative example we will consider the following illustrative dataset:

```
rho <- c(AB = 0.3, bb = 0.9, bA = 0.7, bB = 0.6)
SigmaBCO <- rbind(cbind(matrix(c(1,rho["bA"],rho["bA"],1),2,2),
                        matrix(c(rho["bb"],rho["bA"],rho["bB"],rho["AB"]),2,2)),
                  cbind(matrix(c(rho["bb"],rho["bB"],rho["bA"],rho["AB"]),2,2),
                        matrix(c(1,rho["bB"],rho["bB"],1),2,2)))
muBCO <- c(b1 = 0, A = 1, b2 = 0, B = 1.5)

library(mvtnorm)
set.seed(10)
n.obs <- 15
M1 <- data.frame(id = 1:n.obs, sequence = "AB",
                 rmvnorm(n.obs, mean = muBCO, sigma = SigmaBCO))
names(M1)[3:6] <- paste0("T",1:4)
M2 <- data.frame(id = n.obs+(1:n.obs), sequence = "BA",
                 rmvnorm(n.obs, mean = muBCO[c(1,4,3,2)],
                         sigma = SigmaBCO[c(1,4,3,2),c(1,4,3,2)]))
names(M2)[3:6] <- paste0("T",1:4)
dfL.BCO <- reshape(rbind(M1,M2), direction = "long",
                   idvar = "id", varying = names(M1)[-(1:2)], v.names = c("Y"), times = 1:4)
dfL.BCO$treatment <- "baseline"
dfL.BCO$treatment[dfL.BCO$time == 2 & dfL.BCO$sequence == "AB"] <- "A"
dfL.BCO$treatment[dfL.BCO$time == 4 & dfL.BCO$sequence == "AB"] <- "B"
dfL.BCO$treatment[dfL.BCO$time == 2 & dfL.BCO$sequence == "BA"] <- "B"
dfL.BCO$treatment[dfL.BCO$time == 4 & dfL.BCO$sequence == "BA"] <- "A"
dfL.BCO$treatment <- factor(dfL.BCO$treatment, levels = c("baseline","A","B"))
dfL.BCO$period <- as.character(1 + (dfL.BCO$time %in% 3:4))
dfL.BCO[dfL.BCO$id==1,]
```

```
    id sequence time        Y treatment period
1.1  1       AB    1 -0.84169  baseline      1
1.2  1       AB    2  0.36197         A      1
1.3  1       AB    3 -1.28127  baseline      2
1.4  1       AB    4  0.57369         B      2
```

```
gg.BCO <- ggplot(dfL.BCO, aes(x=time, y = Y, group = id))
gg.BCO <- gg.BCO + geom_line()
gg.BCO <- gg.BCO + geom_point(aes(color = treatment, shape = treatment), size = 3)
gg.BCO <- gg.BCO + facet_wrap(~sequence, labeller = label_both)
gg.BCO
```



4.1 First time period

If we restrict the dataset to the first period (time 1 and 2) we obtain a standard 2 arm randomized trial. To compare mixed model results with t-test tests and ANCOVA, we will use the wide format of the dataset:

```
dfW.BCO <- reshape(dfL.BCO[c("id","time","treatment","Y")], direction = "wide",
                   timevar = "time", idvar = "id", v.names = c("Y","treatment"))
```

4.1.1 Change

Performing a two-sample t-test on the change:

```
tt.change <- with(dfW.BCO, t.test(y = Y.2[treatment.2=="B"]-Y.1[treatment.2=="B"],
                                  x = Y.2[treatment.2=="A"]-Y.1[treatment.2=="A"]))
c(estimate = unname(diff(tt.change$estimate)),
  se = tt.change$stderr,
  lower = tt.change$conf.int[1],
  upper = tt.change$conf.int[1],
  df = tt.change$parameter,
  p.value = tt.change$p.value)
```

```
estimate        se     lower     upper     df.df   p.value
 0.529437  0.224916 -0.990797 -0.990797 27.165129  0.026059
```

is equivalent to a linear mixed model iwht interaction between:

```
e.lmm <- lmm(Y ~ time*sequence, repetition =~time|id,
              data = dfL.BCO[dfL.BCO$time %in% 1:2,])
model.tables(e.lmm)
```

```
                 estimate      se     df     lower    upper    p.value
(Intercept)     -1.59683 0.35438 28.004 -2.322738 -0.87092 1.0687e-04
time             1.16038 0.15904 27.995  0.834599  1.48616 6.0708e-08
sequenceBA      -0.12004 0.50117 28.004 -1.146632  0.90655 8.1244e-01
time:sequenceBA  0.52944 0.22492 27.995  0.068714  0.99016 2.5827e-02
```

4.1.2 Ancova

Controlling for baseline difference using an ANCOVA:

```
e.ANCOVA <- lm(Y.2 ~ Y.1 + treatment.2, data = dfW.BCO)
summary(e.ANCOVA)$coef
```

```
             Estimate Std. Error t value   Pr(>|t|)
(Intercept)   1.07449    0.15983  6.7226 3.2294e-07
Y.1           0.80321    0.10762  7.4631 4.9918e-08
treatment.2B  0.61000    0.22050  2.7664 1.0103e-02
```

has the same estimate of a linear mixed model with baseline constraint:

```
e0.lmm <- lmm(Y ~ treatment, repetition =~time|id,
              data = dfL.BCO[dfL.BCO$time %in% 1:2,])
summary(anova(e0.lmm, effects = c("treatmentB-treatmentA=0")))
```

```
              Hypothesis-specific Wald test

                         estimate    se   df lower upper p.value
 treatmentB-treatmentA=0     0.61 0.217 27.8 0.166 1.054 0.00882 **
 ------------------------------------------------------------------
  :  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1.
df: Satterthwaite approximation w.r.t. model-based se.
se: based on the observed information (model-based).
```

⚠ When fitting the mixed model, the variance was on purpose modeled to be time dependent instead of treatment dependent to match the ANCOVA. In many applications, however, a treatment dependent variance and correlation is preferable:

```
e0T.lmm <- lmm(Y ~ treatment, repetition =~treatment|id,
               data = dfL.BCO[dfL.BCO$time %in% 1:2,])
summary(anova(e0T.lmm, effects = c("treatmentB-treatmentA=0")))
```

```
              Hypothesis-specific Wald test

                         estimate    se   df lower upper p.value
 treatmentB-treatmentA=0    0.602 0.216 27.7 0.159 1.046 0.00954 **
 ------------------------------------------------------------------
  :  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1.
df: Satterthwaite approximation w.r.t. model-based se.
se: based on the observed information (model-based).
```

Covariates: to retrieve the same point estimate between the ANCOVA and the mixed model, covariates should be included in the mixed model with a time interaction.

4.2 Multiple time periods

4.2.1 Change

A straightforward analysis with multiple time periods is to evaluate the within-individual difference between treatment of the change from baseline:

```
dfW.BCO$dY <- (dfW.BCO$Y.4-dfW.BCO$Y.3)-(dfW.BCO$Y.2-dfW.BCO$Y.1)
tt.change2 <- with(dfW.BCO, t.test(c(dY[treatment.4 == "B"],-dY[treatment.4 == "A"])))
c(estimate = unname(tt.change2$estimate),
  se = tt.change2$stderr,
  lower = tt.change2$conf.int[1],
  upper = tt.change2$conf.int[1],
  df = tt.change2$parameter,
  p.value = tt.change2$p.value)
```

```
estimate        se     lower     upper     df.df   p.value
 0.479266  0.181401  0.108260  0.108260 29.000000  0.013144
```

One can retrieve this result by introducing 2 design variables:

```
dfL.BCO$treated <- (dfL.BCO$treatment!="baseline")
dfL.BCO$periodB <- with(dfL.BCO, (period==1 & sequence=="BA") | (period==2 & sequence=="AB"))
e.lmm2tt <- lmm(Y ~ sequence:treated+(treated*periodB),
                repetition = ~time|id, data = dfL.BCO)
model.tables(e.lmm2tt)
```

```
                         estimate       se     df    lower    upper    p.value
(Intercept)              -0.43409 0.264062 28.838 -0.97429 0.106105 1.1105e-01
treatedTRUE               1.14741 0.140692 35.009  0.86179 1.433027 1.3152e-09
periodBTRUE              -0.06553 0.071101 28.992 -0.21095 0.079889 3.6432e-01
sequenceBA:treatedFALSE   0.47492 0.370826 28.037 -0.28463 1.234483 2.1079e-01
sequenceBA:treatedTRUE    0.52510 0.325697 28.013 -0.14205 1.192241 1.1812e-01
treatedTRUE:periodBTRUE   0.47927 0.181401 28.988  0.10825 0.850279 1.3146e-02
```

where the coefficient of interest in the last line, among the treatment periods (A or B) whether there is a difference between being under treatment B or treatment A.

4.2.2 Ancova

A more 'natural' linear mixed model would simply adjust the treatment effect on period and sequence effect:

```
eBCO.lmm <- lmm(Y ~ period + sequence + treatment, repetition =~time|id,
                data = dfL.BCO)
summary(anova(eBCO.lmm, effects = c("treatmentB-treatmentA=0")))
```

```
              Hypothesis-specific Wald test

                         estimate    se   df lower upper p.value
 treatmentB-treatmentA=0    0.436 0.168 27.4 0.091 0.781  0.0153 *
 -----------------------------------------------------------------
  :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
df: Satterthwaite approximation w.r.t. model-based se.
se: based on the observed information (model-based).
```

but this does not match the average individual difference adjusted for both baselines:

```
dfW.BCO$dY <- with(dfW.BCO, (Y.4-Y.2)*(treatment.4=="B")+(Y.2-Y.4)*(treatment.2=="B"))
summary(lm(dY ~ Y.1 + Y.3, data = dfW.BCO))$coef
```

```
Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.40544    0.17976  2.25551 0.032412
Y.1         -0.38831    0.46152 -0.84139 0.407522
Y.3          0.36836    0.45811  0.80409 0.428370
```

but is quite close to the average period-specific ANCOVA treatment effect:

```
e.ANCOVA2 <- lm(Y.4 ~ Y.3 + treatment.4, data = dfW.BCO)
mean(c(summary(e.ANCOVA)$coef[3,"Estimate"],summary(e.ANCOVA2)$coef[3,"Estimate"]))
```

```
[1] 0.43672
```

⚠ In many applications, however, a treatment dependent variance and correlation is preferable:

```
eBCOT.lmm <- lmm(Y ~ period + sequence + treatment, repetition =~time|id,
                 structure = CS(list(~treatment,~treatment)), data = dfL.BCO)
summary(anova(eBCOT.lmm, effects = c("treatmentB-treatmentA=0")))
```

```
              Hypothesis-specific Wald test

                         estimate    se df lower upper p.value
 treatmentB-treatmentA=0    0.414 0.172 29 0.062 0.765  0.0226 *
 ---------------------------------------------------------------
  :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
df: Satterthwaite approximation w.r.t. model-based se.
se: based on the observed information (model-based).
```

Appendix A Mixed model estimator as a weighted average

A.1 Treatment as repetition variable

Consider the linear mixed model matching the t-test results when estimating the treatment effect:

```
eTreat.lmm2tt <- lmm(duration ~ treatment, repetition = ~treatment|id, data = bloodpressureL)
coef(eTreat.lmm2tt)
```

```
(Intercept)  treatmentB  treatmentC 
     1.7250      0.5750      1.2583 
```

The estimates correspond to a Generalized Least Squared (GLS) estimator defined by:

- a block diagonal covariance matrix with element

```
Omega1 <- sigma(eTreat.lmm2tt,
                cluster = "all", simplify = TRUE)
Omega1[1:3,1:3]
```

```
           [,1]       [,2]     [,3]
[1,]  0.3347727 -0.0072727 0.047727
[2,] -0.0072727  0.1236364 0.162727
[3,]  0.0477273  0.1627273 0.372424
```

- a design matrix with element:

```
X1 <- model.matrix(eTreat.lmm2tt)
head(X1,3)
```

```
  (Intercept) treatmentB treatmentC
1           1          0          0
2           1          1          0
3           1          0          1
```

The corresponding projector weight each observation:

- proportionally to the sample size for treatments related to the regression parameter
- 0 otherwise:

```
P1 <- solve(t(X1) %*% solve(Omega1) %*% X1) %*% t(X1) %*% solve(Omega1)
vecP1 <- apply(round(P1,4), MARGIN = 1, FUN = table, y = bloodpressureL$treatment)
vecP1
```

```
$`(Intercept)`
        A  B  C
0       0 12 12
0.0833 12  0  0
total   1  0  0
```

```
$treatmentB
         A  B  C
-0.0833 12  0  0
0        0  0 12
0.0833   0 12  0
total   -1  1  0
```

```
$treatmentC
         A  B  C
-0.0833 12  0  0
0        0 12  0
0.0833   0  0 12
total   -1  0  1
```

We can verify that we retrieve the mixed model estimates:

```
(P1 %*% bloodpressureL$duration)[,1]
```

```
(Intercept)  treatmentB  treatmentC 
     1.7250      0.5750      1.2583 
```

A.2 Period as repetition variable

Consider the same linear mixed model but with period as repetition variable:

```
ePeriod.lmm2tt <- lmm(duration ~ treatment, repetition = ~period|id, data = bloodpressureL)
coef(ePeriod.lmm2tt)
```

```
(Intercept)  treatmentB  treatmentC
    1.68755     0.58766     1.16557
```

The estimates correspond to a Generalized Least Squared (GLS) estimator defined by:

- a block diagonal covariance matrix with elements

```
Omega2 <- sigma(ePeriod.lmm2tt,
               cluster = "all", simplify = TRUE)
Omega2[1:3,1:3]
```

```
         [,1]     [,2]    [,3]
[1,] 0.229440 0.082455 0.01444
[2,] 0.082455 0.249826 0.11704
[3,] 0.014440 0.117040 0.36480
```

- a design matrix with element:

```
X2 <- model.matrix(eTreat.lmm2tt)
X2[1:3,1:3]
```

```
  (Intercept) treatmentB treatmentC
1           1          0          0
2           1          1          0
3           1          0          1
```

The weigthing of the observations is less intuitive as all treatments contribute, to various extends, to each regression parameter.

```
P2 <- solve(t(X2) %*% solve(Omega2) %*% X2) %*% t(X2) %*% solve(Omega2)
vecP2 <- apply(round(P2,4), MARGIN = 1, FUN = table, y = bloodpressureL$treatment,
               simplify = FALSE)
```

```
$`(Intercept)`               $treatmentB                  $treatmentC
        A B C                        A  B C                       A  B C
-0.0156 0 4 0                -0.1052  4 0 0               -0.1078  4 0 0
-0.0129 0 0 4                -0.102   4 0 0               -0.0734  4 0 0
-0.0123 0 4 0                -0.0429  4 0 0               -0.0688  4 0 0
-0.0053 0 0 4                -0.0306  0 0 4               -0.0332  0 4 0
0.0182  0 0 4                -0.0026  0 0 4               0.0026   0 4 0
0.0279  0 4 0                0.0332   0 0 4               0.0306   0 4 0
0.0611  4 0 0                0.0688   0 4 0               0.0429   0 0 4
0.0922  4 0 0                0.0734   0 4 0               0.102    0 0 4
0.0967  4 0 0                0.1078   0 4 0               0.1052   0 0 4
total   1 0 0                total   -1 1 0               total   -1 0 1
```

We can verify that we retrieve the mixed model estimates:

```
(P2 %*% bloodpressureL$duration)[,1]
```

```
(Intercept)  treatmentB  treatmentC
    1.68755     0.58766     1.16557
```

A.3 Treatment as repetition variable, adjusted for period

We now consider the linear mixed model similar to the t-test but adjusting for period:

```
eTreat.lmm <- lmm(duration ~ treatment + period, repetition = ~treatment|id,
                  data = bloodpressureL)
coef(eTreat.lmm)
```

```
(Intercept)  treatmentB  treatmentC     period2     period3
    1.54915     0.57500     1.25833     0.19991     0.32764
```

As before we extract the residual-variance covariance matrix and the design matrix:

```
Omega3 <- sigma(eTreat.lmm,
               cluster = "all", simplify = TRUE)
X3 <- model.matrix(eTreat.lmm)
```

to understand how the GLS estimator weight each observation:

```
P3 <- solve(t(X3) %*% solve(Omega3) %*% X3) %*% t(X3) %*% solve(Omega3)
vecP3 <- apply(round(P3,4), MARGIN = 1, FUN = table, y = bloodpressureL$treatment)
vecP3
```

```
$`(Intercept)`
        A B C
-0.0828 0 4 0
-0.0497 0 0 4                $treatmentB                  $treatmentC
-0.0134 0 4 0                         A  B  C                      A  B  C
-0.0048 0 0 4                -0.0833 12  0  0             -0.0833 12  0  0
0.0546  0 0 4                0        0  0 12             0        0 12  0
0.0631  4 0 0                0.0833   0 12  0             0.0833   0  0 12
0.0876  4 0 0                total   -1  1  0             total   -1  0  1
0.0962  0 4 0
0.0992  4 0 0
total   1 0 0
```

We can verify that we retrive the mixed model estimates:

```
(P3 %*% bloodpressureL$duration)[,1]
```

```
(Intercept)  treatmentB  treatmentC     period2     period3
    1.54915     0.57500     1.25833     0.19991     0.32764
```

A.4 Period as repetition variable, adjusted for period

We now consider the linear mixed model similar to the t-test but adjusting for period:

```
ePeriod.lmm <- lmm(duration ~ treatment + period, repetition = ~period|id,
                  data = bloodpressureL)
coef(ePeriod.lmm)
```

```
(Intercept)  treatmentB  treatmentC     period2     period3 
    1.31867     0.74657     1.39742     0.35833     0.55000 
```

As before we extract the residual-variance covariance matrix and the design matrix:

```
Omega4 <- sigma(ePeriod.lmm,
               cluster = "all", simplify = TRUE)
X4 <- model.matrix(ePeriod.lmm)
```

to understand how the GLS estimator weight each observation:

```
P4 <- solve(t(X4) %*% solve(Omega4) %*% X4) %*% t(X4) %*% solve(Omega4)
vecP4 <- apply(round(P4,4), MARGIN = 1, FUN = table, y = bloodpressureL$treatment)
vecP4
```

```
   $`(Intercept)`              $treatmentB                $treatmentC
           A B C                        A B C                        A B C
   -0.0442 0 4 0               -0.1389  4 0 0               -0.1329  4 0 0
   -0.0385 0 0 4               -0.0738  4 0 0               -0.0849  4 0 0
   0.0032  0 4 0               -0.0477  0 0 4               -0.0417  0 4 0
   0.0035  0 0 4               -0.0373  4 0 0               -0.0321  4 0 0
   0.035   0 0 4               0.006    0 0 4               -0.006   0 4 0
   0.0353  4 0 0               0.0321   0 4 0               0.0373   0 0 4
   0.0407  4 0 0               0.0417   0 0 4               0.0477   0 4 0
   0.041   0 4 0               0.0849   0 4 0               0.0738   0 0 4
   0.1739  4 0 0               0.1329   0 4 0               0.1389   0 0 4
   total   1 0 0               total   -1 1 0               total   -1 0 1
```

We can verify that we retrieve the mixed model estimates:

```
(P4 %*% bloodpressureL$duration)[,1]
```

```
(Intercept)  treatmentB  treatmentC     period2     period3 
    1.31867     0.74657     1.39742     0.35833     0.55000 
```