Connexions between traditional tests and mixed models

Brice Ozenne

October 29, 2025

This vignette connects well-known tests (t.test, ANCOVA, Pearson's correlation, Bartlett's test, ...) with the output of a linear mixed model. The connexion typically only holds in absence of missing data.

1 Illustrative datasets

We will consider two illustrative datasets:

- data from the abeta study lifestyle and psychosicial data between patients with newly diagnosed bipolar disorder (`BD`) and matched healthy controls (`HC`) at baseline: functioning assessment test (`fast0`), quality of life (`qol0`), perceived stress score (`pss0`), ... and at 1 year follow-up (`pss1`, `fast1`, `qol1`, for `BD` only).

```
data(abetaW, package = "LMMstar")
abetaW$missingreason <- NULL
head(abetaW)
```

```
  id sex age group episode fast0 qol0 pss0 fast1 pss1 qol1 educationyears alcohol
1  1   M  30    BD       0     1   88    9     0   NA   NA             13       0
2  2   F  55    BD       1    32   87   21    NA   NA   NA             15       0
3  3   M  51    BD       0    29   86   23    31   27   79             21       1
4  4   M  38    BD       0     1   96    7     6    6  101             21       1
5  5   M  21    BD       0     3   97    1     1    5  105             12       1
6  6   M  42    BD       1    22   70   17    40   18   68             13       0
```

This dataset shows great difference in heterogeneity between the two groups, e.g.:

```
library(LMMstar)
summarize(pss0 ~ group, data = abetaW, na.rm = TRUE)
```

```
  group observed missing    mean     sd min   q1 median    q3 max
1    BD       86       1 13.2674 6.8435   1 7.25     13 17.75  29
2    HC       44       0  7.2727 5.0272   0 3.75      6 10.50  19
```

We will also use a balanced version of this dataset (equal group size):

```
abetaW.B <- do.call(rbind, by(abetaW, abetaW$group, function(iDF){
  iDF[which(!is.na(iDF$pss0))[1:44],]
}))
```

- data from the calcium dataset, a two-arm randomized clinical trial comparing bone mineral density between calcium supplement (`C`) and placebo (`P`). Visits were planned every 6 months, `bmd1` refers to the baseline measurement and `bmd2`, ..., `bmd5` refers to post-intervention measurements. `time.obs1`,..., `time.obs5` refer to the time elpased from baseline measurement in years.

```
data(calciumL, package = "LMMstar")
calciumL <- merge(by = "girl", calciumL,
                  transform(calciumL, baseline = bmd)[calciumL$visit==1,c("girl","baseline")])
calciumL <- calciumL[order(calciumL$girl,calciumL$visit),]
head(calciumL)
```

```
   girl grp dropout dropvisit visit time.obs bmd baseline
1   101   C       0        NA     1  0.00000 815      815
3   101   C       0        NA     2  0.51472 875      815
5   101   C       0        NA     3  0.98015 911      815
2   101   C       0        NA     4  1.49760 952      815
4   101   C       0        NA     5  1.99589 970      815
10  102   P       0        NA     1  0.00000 813      813
```

The corresponding wide format is

```
data(calciumW, package = "LMMstar")
calciumW$dropout <- NULL
calciumW$dropvisit <- NULL
head(calciumW)
```

```
  girl grp bmd1 bmd2 bmd3 bmd4 bmd5 time.obs1 time.obs2 time.obs3 time.obs4 time.obs5
1  101   C  815  875  911  952  970         0   0.51472   0.98015    1.4976    1.9959
2  102   P  813  833  855  881  901         0   0.51472   0.95551    1.4730    1.9521
3  103   P  812  812  843  855  895         0   0.51198   0.95825    1.4757    1.9548
4  104   C  804  847  885  920  948         0   0.51198   0.97194    1.5086    2.1136
5  105   C  904  927  952  955 1002         0   0.57495   0.97741    1.4757    1.9548
6  106   P  831  855  890  908  933         0   0.53388   1.01300    1.5907    2.1684
```

We will use the placebo group as reference:

```
calciumW$grp <- relevel(calciumW$grp, "P")
calciumL$grp <- relevel(calciumL$grp, "P")
```

and the change from baseline in bone mineral density:

```
calciumW$change2 <- calciumW$bmd2 - calciumW$bmd1
calciumW$change3 <- calciumW$bmd3 - calciumW$bmd1
calciumW$change4 <- calciumW$bmd4 - calciumW$bmd1
calciumW$change5 <- calciumW$bmd5 - calciumW$bmd1
calciumL$change <- calciumL$bmd - calciumL$baseline
```

For illustrative purpose, we will restrict both dataset to subjects with complete data:

```
calciumW.NNA <- calciumW[rowSums(is.na(calciumW))==0,]
calciumL.NNA <- calciumL[calciumL$girl %in% calciumW.NNA$girl,]
```

as the aim is to show equivalence between statistical tests when there is no missing data.

2 Test on the mean

2.1 Welch two sample t-test

A two sample t-test:

```
with(abetaW, t.test(x = pss0[group=="BD"], y = pss0[group=="HC"]))
```

```
        Welch Two Sample t-test

data:  pss0[group == "BD"] and pss0[group == "HC"]
t = 5.67, df = 112, p-value = 1.1e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 3.8988 8.0906
sample estimates:
mean of x mean of y
  13.2674    7.2727
```

is equivalent to a linear regression with a group-specific residual variance:

```
abetaW$group <- relevel(abetaW$group,"HC")
e.ttest <- lmm(pss0 ~ group, structure = IND(~group),
               data = abetaW, trace = FALSE)
model.tables(e.ttest, effects = "all")
```

```
            estimate      se      df  lower  upper    p.value
(Intercept)   7.2727 0.75788  43.009 5.7443 8.8011 2.9650e-12
groupBD       5.9947 1.05781 112.201 3.8988 8.0906 1.1399e-07
sigma         5.0272 0.54210  43.009 4.0447 6.2484         NA
k.BD          1.3613 0.18014  86.351 1.0464 1.7709 2.2090e-02
```

For comparison a linear model would estimate different standard errors, degrees of freedom, and p-values:

```
model.tables(lmm(pss0 ~ group, data = abetaW))
```

```
            estimate      se     df  lower  upper    p.value
(Intercept)   7.2727 0.94857 128.03 5.3958 9.1496 3.8629e-12
groupBD       5.9947 1.16625 128.03 3.6871 8.3023 1.0000e-06
```

as it does not account for heteroschedasticity. This makes the 'heteroschedastic linear regression' `e.ttest` a natural extension of the t-test when it comes to account for covariates.

In the special case of two groups of equal size, the standard errors estimated accounting for heteroschedasticity:

```
model.tables(lmm(pss0 ~ group, structure = IND(~group),
                 data = abetaW.B, trace = FALSE))
```

```
            estimate      se     df   lower   upper    p.value
(Intercept)  11.8636 0.98648 43.009  9.8742 13.8530 2.4425e-15
groupHC      -4.5909 1.24399 80.661 -7.0662 -2.1156 4.0523e-04
```

or ignoring it:

```
model.tables(lmm(pss0 ~ group, data = abetaW.B))
```

```
            estimate      se     df   lower   upper    p.value
(Intercept)  11.8636 0.87964 86.017 10.1150 13.6123 0.00000000
groupHC      -4.5909 1.24399 86.017 -7.0639 -2.1179 0.00039184
```

will be the same, leading to very similar p-values (degrees of freedom differ slightly).

2.2 Paired t-test

With complete data, a paired t-test:

```
t.test(calciumW.NNA$bmd2, calciumW.NNA$bmd1, paired = TRUE)
```

```
        Paired t-test

data:  calciumW.NNA$bmd2 and calciumW.NNA$bmd1
t = 13, df = 90, p-value <2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 20.229 27.529
sample estimates:
mean difference
         23.879
```

is equivalent to a LMM with an unstructured covariate pattern:

```
e.lmm2tt <- lmm(bmd ~ visit, repetition = ~visit|girl, structure = "UN",
                data = calciumL.NNA)
model.tables(e.lmm2tt)["visit2",,drop=FALSE]
```

```
       estimate     se     df  lower  upper p.value
visit2   23.879 1.8371 89.968 20.229 27.529       0
```

2.3 Comparing change

2.3.1 Using a Welch two sample t-test

With complete data, a two sample t-test comparing the change from baseline:

```
ttc <- with(calciumW.NNA, t.test(x = change2[grp=="C"], y = change2[grp=="P"]))
ttc
```

```
        Welch Two Sample t-test

data:  change2[grp == "C"] and change2[grp == "P"]
t = 2.03, df = 88.8, p-value = 0.046
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.14074 14.49659
sample estimates:
mean of x mean of y
   27.659    20.340
```

is equivalent to a LMM with a stratified unstructured covariate pattern:

```
e.lmm2tt2 <- lmm(bmd ~ visit*grp, repetition = ~visit|girl, structure = UN(~grp),
                 data = calciumL.NNA)
model.tables(e.lmm2tt2)[c("visit2","visit2:grpC"),,drop=FALSE]
```

```
            estimate     se     df    lower  upper    p.value
visit2       20.3404 2.5338 46.005 15.24013 25.441 2.6911e-10
visit2:grpC   7.3187 3.6124 88.734  0.14069 14.497 4.5767e-02
```

The estimate and standard error are exactly the same:

```
c(ttc$estimate["mean of x"] - ttc$estimate["mean of y"],
  se = ttc$stderr)
```

```
mean of x        se
   7.3187    3.6124
```

The only (small) difference lies in the estimation of the degrees of freedom.

2.3.2 Using a linear regression

Using a linear model to compare change over time:

```
eLM.change <- lm(change2 ~ grp, data = calciumW.NNA)
summary(eLM.change)$coef
```

```
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  20.3404     2.5133  8.0931 2.7975e-12
grpC          7.3187     3.6144  2.0249 4.5878e-02
```

is equivalent to the following mixed model:

```
eLMM.change <- lmm(bmd ~ visit*grp,
                   repetition =~ visit|girl, structure = UN,
                   data = calciumL.NNA)
model.tables(eLMM.change)[c("visit2","visit2:grpC"),]
```

```
            estimate     se     df    lower  upper    p.value
visit2       20.3404 2.5133 88.962 15.34654 25.334 2.8044e-12
visit2:grpC   7.3187 3.6144 88.962  0.13688 14.500 4.5880e-02
```

Here, since the linear regression assumes the same variance in both groups, we did not stratified the covariance pattern on group. The same equivalence would hold with a continuous exposure (say dose) instead of a binary exposure (here `grp`).

In presence of a covariate:

```
set.seed(1)
calciumW2.NNA <- cbind(calciumW.NNA,
                       age = round(runif(NROW(calciumW.NNA), min = 18, max = 60)))
calciumL2.NNA <- merge(calciumL.NNA, calciumW2.NNA[,c("girl","age")], by = "girl")

eLMadj.change <- lm(change2 ~ grp + age, data = calciumW2.NNA)
summary(eLMadj.change)$coef
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.17495    6.68052  1.3734 0.173121
grpC         6.99548    3.57426  1.9572 0.053495
age          0.28771    0.15982  1.8002 0.075251
```

one should specify interaction with time in the mixed model to retrieve the same results:

```
eLMMadj.change <- lmm(bmd ~ visit*grp + visit*age,
                      repetition =~ visit|girl, structure = UN,
                      data = calciumL2.NNA)
model.tables(eLMMadj.change)[c("visit2","visit2:grpC"),]
```

```
            estimate     se     df    lower  upper  p.value
visit2        9.1750 6.6805 87.966 -4.10126 22.451 0.173122
visit2:grpC   6.9955 3.5743 87.966 -0.10764 14.099 0.053497
```

2.4 Multiple Student's t-test

To adjust several t-tests for multiple testing, one can use the equivalence with `lmm`. This however require to specify the structure of the data (via the argument `repetition`), i.e., at which level replicates are independent so the software can deduce the appropriate number of independent observation across t-tests:

```
e.ttest2 <- lmm(change2 ~ grp, structure = IND(~grp),
                data = calciumW, repetition = ~1|girl, trace = FALSE)
e.ttest3 <- lmm(change3 ~ grp, structure = IND(~grp),
                data = calciumW, repetition = ~1|girl, trace = FALSE)
e.ttest4 <- lmm(change4 ~ grp, structure = IND(~grp),
                data = calciumW, repetition = ~1|girl, trace = FALSE)
e.ttest5 <- lmm(change5 ~ grp, structure = IND(~grp),
                data = calciumW, repetition = ~1|girl, trace = FALSE)
```

The `anova` method is then used to specify the parameter of interest and the results combined using `rbind`:

```
e.mttest <- rbind(anova(e.ttest2, effects = "grpC=0"),
                  anova(e.ttest3, effects = "grpC=0"),
                  anova(e.ttest4, effects = "grpC=0"),
                  anova(e.ttest5, effects = "grpC=0"))
model.tables(e.mttest, method = "single-step2")
```

```
               estimate     se      df   lower  upper  p.value
change2: grpC=0   6.7507 3.3549 103.014 -1.2191 14.721 0.112799
change3: grpC=0  13.8150 4.8336  95.812  2.3321 25.298 0.014660
change4: grpC=0  12.5190 5.8369  86.835 -1.3473 26.385 0.084529
change5: grpC=0  19.0155 6.4666  86.916  3.6533 34.378 0.011440
```

Note: the `single-step2` adjustment is similar to the `single-step` adjustment of the multcomp package, i.e., a max test adjustment. But instead of relying on the density of a multivariate Student's t-distribution, which requires equal degrees of freedom, it samples in a multivariate distribution with Student's t marginal possibly based on different degrees of freedom and a Gaussian copula. Being based on random sampling, results will slightly change everytime the code is run unless the inital state of the random number generator is set to a specific value before running the code:

```
set.seed(1)
model.tables(e.mttest, method = "single-step2")
```

```
               estimate     se      df   lower  upper  p.value
change2: grpC=0   6.7507 3.3549 103.014 -1.2151 14.717 0.113439
change3: grpC=0  13.8150 4.8336  95.812  2.3379 25.292 0.014590
change4: grpC=0  12.5190 5.8369  86.835 -1.3404 26.378 0.085339
change5: grpC=0  19.0155 6.4666  86.916  3.6609 34.370 0.011640
```

The `LMMstar.options` function can be used to output the number of samples used:

```
LMMstar.options()$n.sampleCopula
```

```
[1] 1e+05
```

and change it:

```
LMMstar.options(n.sampleCopula = 1e4)
```

This whole procedure can be streamlined using the long format and the `mlmm` function:

- the argument `by` indicates how to split the data. A separate model is fitted on each split.
- the argument `effects` indicates the test to be extracted for each model.
- the argument `name.short` is a cosmetic argument: should the name of each test be the covariate value or a combination of the covariate variable and the covariate value?

```
e.mttest2 <- mlmm(change ~ grp, structure = IND(~grp), repetition = ~visit|girl,
                  data = calciumL[calciumL$visit!=1,], trace = FALSE,
                  by = "visit", effects = "grpC=0", name.short = FALSE)
set.seed(1)
model.tables(e.mttest, method = "single-step2")
```

```
                estimate     se      df   lower  upper  p.value
change2: grpC=0   6.7507 3.3549 103.014 -1.2151 14.717 0.113439
change3: grpC=0  13.8150 4.8336  95.812  2.3379 25.292 0.014590
change4: grpC=0  12.5190 5.8369  86.835 -1.3404 26.378 0.085339
change5: grpC=0  19.0155 6.4666  86.916  3.6609 34.370 0.011640
```

The function `mlmm` can be used not only to emulate multiple t-tests but also for multiple linear regressions or linear mixed models. In the special case of multiple Welch two-sample test, a dedicated function `mt.test` offers a more user friendly interface:

```
set.seed(1)
mt.test(change2 + change3 + change4 + change5 ~ grp, data = calciumW)
```

```
Argument 'data' contains 59 missing values.
        estimate     se      df   lower  upper  p.value
change2   6.7507 3.3549 103.014 -1.2151 14.717 0.113439
change3  13.8150 4.8336  95.812  2.3379 25.292 0.014590
change4  12.5190 5.8369  86.835 -1.3404 26.378 0.085339
change5  19.0155 6.4666  86.916  3.6609 34.370 0.011640
```

2.5 ANCOVA

Instead of comparing the final value or the change between groups using a Welch two sample t-test, the ANCOVA is often refered to as the superior approach to assess a treatment effect (Vickers and Altman, 2001). It regresses the group variable and the baseline value against the change:

```
model.tables(lmm(change2 ~ bmd1 + grp, data = calciumW.NNA))
```

```
             estimate       se     df      lower    upper  p.value
(Intercept) -25.742684 25.757918 88.018 -76.930991 25.44562 0.320337
bmd1          0.052948  0.029457 88.018  -0.005592  0.11149 0.075693
grpC          6.741021  3.584377 88.018  -0.382155 13.86420 0.063324
```

or the final value:

```
model.tables(lmm(bmd2 ~ bmd1 + grp, data = calciumW.NNA))
```

```
            estimate       se     df     lower   upper  p.value
(Intercept) -25.7427 25.757918 88.018 -76.93099 25.4456 0.320337
bmd1          1.0529  0.029457 88.018   0.99441  1.1115 0.000000
grpC          6.7410  3.584377 88.018  -0.38215 13.8642 0.063324
```

both leading to equivalent result. The corresponding mixed model constrains the both group to take the same baseline value. This can be specified by introducing a new covariate that only differ between groups after baseline:

```
calciumL.NNA$trt <- ifelse(calciumL.NNA$visit==1,"P",as.character(calciumL.NNA$grp))
calciumL.NNA$trt <- factor(calciumL.NNA$trt, levels = c("P","C"))
ftable(grp = calciumL.NNA$grp, trt = calciumL.NNA$trt, visit = calciumL.NNA$visit)
```

```
        visit  1  2  3  4  5
grp trt
P   P         47 47 47 47 47
    C          0  0  0  0  0
C   P         44  0  0  0  0
    C          0 44 44 44 44
```

We then retrieve the same estimate and similar (but not identical) standard errors and p-values with the following mixed model:

```
e.lmmANCOVA <- lmm(bmd ~ visit*trt, repetition = ~visit|girl, structure = UN,
                   data = calciumL.NNA)
model.tables(e.lmmANCOVA)["visit2:trtC",,drop=FALSE]
```

```
Constant values in the design matrix for the mean structure.
Coefficient "trtC" relative to interaction "visit:trt" has been removed.
            estimate     se     df   lower  upper  p.value
visit2:trtC    6.741 3.5642 88.853 -0.3411 13.823 0.061839
```

To avoid the message about the design matrix, one should 'manually' define the interaction terms:

```
calciumL.NNA$visit.trt <- ifelse(calciumL.NNA$trt == "C", calciumL.NNA$visit, "baseline")
calciumL.NNA$visit.trt <- factor(calciumL.NNA$visit.trt, levels = c("baseline",2:5))
ftable(grp = calciumL.NNA$grp, visit.trt = calciumL.NNA$visit.trt, visit = calciumL.NNA$visit)
```

```
             visit  1  2  3  4  5
grp visit.trt
P   baseline       47 47 47 47 47
    2               0  0  0  0  0
    3               0  0  0  0  0
    4               0  0  0  0  0
    5               0  0  0  0  0
C   baseline       44  0  0  0  0
    2               0 44  0  0  0
    3               0  0 44  0  0
    4               0  0  0 44  0
    5               0  0  0  0 44
```

```
e.lmmANCOVA2 <- lmm(bmd ~ visit + visit.trt, repetition = ~visit|girl, structure = UN,
                    data = calciumL.NNA)
model.tables(e.lmmANCOVA2)["visit.trt2",,drop=FALSE]
```

```
           estimate     se     df   lower  upper  p.value
visit.trt2    6.741 3.5642 88.853 -0.3411 13.823 0.061839
```

As before, in presence of a covariate:

```
summary(lm(bmd2 ~ bmd1 + grp + age, data = calciumW2.NNA))$coef
```

```
            Estimate Std. Error t value   Pr(>|t|)
(Intercept) -37.6452  26.215165 -1.4360 1.5459e-01
bmd1          1.0536   0.029064 36.2524 2.7566e-54
grpC          6.4062   3.540822  1.8093 7.3865e-02
age           0.2914   0.157689  1.8479 6.8008e-02
```

one should add the covariate along with time interactions to retrieve the same estimate and similar standard error/p-value/confindence intervals with a linear mixed model:

```
calciumL2.NNA$visit.trt <- ifelse(calciumL2.NNA$grp == "C", calciumL.NNA$visit, "1")

e.lmmANCOVAadj <- lmm(bmd ~ visit + visit.trt + visit*age, repetition = ~visit|girl,
                      structure = UN, data = calciumL2.NNA)
model.tables(e.lmmANCOVAadj)["visit.trt2",,drop=FALSE]
```

```
           estimate     se     df    lower  upper  p.value
visit.trt2   6.4062 3.5206 87.855 -0.59046 13.403 0.072223
```

A natural extension of the ANCOVA would be to relax the assumption of common residual variance between the two treatment groups:

```
model.tables(lmm(change2 ~ bmd1 + grp, data = calciumW.NNA, structure = IND(~grp)))
```

```
             estimate        se     df       lower    upper  p.value
(Intercept) -25.833272 25.805339 83.926 -77.1506784 25.48413 0.319665
bmd1          0.053052  0.029513 84.179  -0.0056359  0.11174 0.075828
grpC          6.739886  3.585265 87.584  -0.3855470 13.86532 0.063448
```

However the 'straightforward' connexion with mixed model seems lost:

```
e.lmmHANCOVA <- lmm(bmd ~ visit + visit.trt, repetition = ~visit|girl, structure = UN(~grp),
                    data = calciumL.NNA)
model.tables(e.lmmHANCOVA)["visit.trt2",,drop=FALSE]
```

```
           estimate     se     df    lower  upper  p.value
visit.trt2   6.7516 3.5654 88.326 -0.33341 13.837 0.061542
```

3 Test on the correlation

3.1 Pearson's correlation

One can retrieve Pearson's correlation:

```
cor.test(calciumW.NNA$bmd1,calciumW.NNA$bmd5)
```

```
        Pearson's product-moment correlation

data:  calciumW.NNA$bmd1 and calciumW.NNA$bmd5
t = 18.3, df = 89, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.83615 0.92551
sample estimates:
    cor
0.88901
```

using a linear mixed model moving to the long format and using an unstructured mean and covariance pattern over time:

```
eCor.lmm <- lmm(bmd ~ visit, repetition = ~visit|girl,
                structure = UN, data = calciumL.NNA)
model.tables(eCor.lmm,  effects = "correlation")["rho(1,5)",]
```

```
         estimate     se     df   lower   upper p.value
rho(1,5)  0.88901 0.0221 96.839 0.83607 0.92555       0
```

P-value and confidence interval will differ (only slightly in large samples) because `cor.test` uses an exact[1] formula for the variance after `atanh` transformation while the linear mixed model rely on the observed information matrix. In this example the observed information (default option) is more in line with `cor.test` than the expected information:

```
model.tables(eCor.lmm,  type.information = "expected", effects = "correlation")["rho(1,5)",]
```

```
         estimate       se       df   lower   upper p.value
rho(1,5)  0.88901 0.021914 17285033 0.83738 0.92492       0
```

Equivalent results can be obtained with a linear regression:

```
e.lmCor <- lm(bmd1 ~ bmd5, data = calciumW.NNA)
coef(e.lmCor)["bmd5"]/sqrt(coef(e.lmCor)["bmd5"]^2+vcov(e.lmCor)["bmd5","bmd5"]*df.residual(e.
    lmCor))
```

```
bmd5
0.88901
```

[1] assuming jointly normally distributed outcomes

Of note the confidence intervals and p-value of `cor.test` are not computed in a consistent way:

```
set.seed(7303)
X <- rnorm(10)
Y <- rnorm(10)
cor.test(X,Y)
```

```
        Pearson's product-moment correlation

data:  X and Y
t = 2.29, df = 8, p-value = 0.051
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.00016154 0.90179629
sample estimates:
    cor
0.62972
```

Here the confidence intervals do not overlap 0, i.e., suggest to reject the null hypothesis while the p-value is greater than 0.05, i.e., does not suggest to reject the null hypothesis. The corresponding mixed model estimate:

```
dfXY <- rbind(data.frame(value = X, variable = "x", id = 1:10),
              data.frame(value = Y, variable = "y", id = 1:10))
e.lmmXY <- lmm(value ~ variable, repetition = ~variable|id,
               structure = UN, data = dfXY)
model.tables(e.lmmXY, effects = "correlation")
```

```
         estimate      se     df     lower   upper  p.value
rho(x,y)  0.62972 0.20115 7.0024 -0.047159 0.91027 0.061602
```

is the same but the confidence intervals and p-value differ more substantially (due to small sample approximations). They however are consistent with respect to whether to reject the null hypothesis.

3.2 Comparing Pearson's correlations

To compare the Pearson's correlation between two groups, one can use Fisher'z test:

```
rho.C <- with(calciumW.NNA, cor(bmd1[grp=="C"],bmd5[grp=="C"]))
rho.P <- with(calciumW.NNA, cor(bmd1[grp=="P"],bmd5[grp=="P"]))
nobs.C <- sum(calciumW$grp=="C")
nobs.P <- sum(calciumW$grp=="P")
stat.fisher <- (atanh(rho.C) - atanh(rho.P))/sqrt(1/(nobs.C-3)+1/(nobs.P-3))
2*(1-pnorm(abs(stat.fisher)))
```

```
[1] 0.15261
```

and the confidence intervals suggested by Zou (2007):

```
zou.C <- tanh(atanh(rho.C) + qnorm(c(0.025,0.975))/sqrt(nobs.C-3))
zou.P <- tanh(atanh(rho.P) + qnorm(c(0.025,0.975))/sqrt(nobs.P-3))

(rho.C - rho.P) - sqrt( (rho.C-zou.C[1])^2 + (rho.P-zou.P[2])^2 )
(rho.C - rho.P) + sqrt( (rho.C-zou.C[2])^2 + (rho.P-zou.P[1])^2 )
```

```
[1] -0.15309
[1] 0.021034
```

which is implemented in the package cocor:

```
library(cocor)
cocor.indep.groups(r1.jk = rho.C, n1 = nobs.C, r2.hm = rho.P, n2 = nobs.P)
```

```
  Results of a comparison of two correlations based on independent groups

Comparison between r1.jk = 0.8597 and r2.hm = 0.917
Difference: r1.jk - r2.hm = -0.0574
Group sizes: n1 = 55, n2 = 57
Null hypothesis: r1.jk is equal to r2.hm
Alternative hypothesis: r1.jk is not equal to r2.hm (two-sided)
Alpha: 0.05

fisher1925: Fisher's z (1925)
  z = -1.4304, p-value = 0.1526
  Null hypothesis retained

zou2007: Zou's (2007) confidence interval
  95% confidence interval for r1.jk - r2.hm: -0.1531 0.0210
  Null hypothesis retained (Interval includes 0)
```

We can retrieve the same estimated difference and similar but not identical CIs/p-values using a linear mixed model with a covariance pattern stratified on group:

```
eCor2.lmm <- lmm(bmd ~ visit*grp, repetition = ~visit|girl,
                 structure = UN(~grp), data = calciumL.NNA)
model.tables(eCor2.lmm,  effects = "correlation")[c("rho(1,5):C","rho(1,5):P"),]
```

```
          estimate       se     df   lower   upper    p.value
rho(1,5):C  0.85965 0.039801 42.111 0.75492 0.92163 1.2128e-10
rho(1,5):P  0.91701 0.023456 53.835 0.85496 0.95319 7.3275e-15
```

and use a Wald test to compare the correlation coefficients:

```
set.seed(1)
summary(anova(eCor2.lmm, effects = "rho(1,5):C - rho(1,5):P = 0"), digits = 4)
```

```
           Wald F-test

       statistic       df p.value
   all    1.7165 (1,93.6)   0.193
   ------------------------------
    :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
  df: Satterthwaite approximation w.r.t. model-based se.

               Emulated Wald test (resampling parameter distribution)

                                estimate     se   df   lower upper p.value
   rho(1,5):C - rho(1,5):P = 0  -0.0574 0.0495 <NA> -0.1661  0.03   0.197
   ---------------------------------------------------------------------
    :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
  se: based on the observed information (model-based).
  Back-transformation: rho parameters with atanh (1e+05 samples).
```

The 'Wald F-test' is the usual Wald test defined by the squared difference between the two correlation coefficients divided by the squared standard error of this difference. This ratio follows, under the null hypothesis, an F-distribution which is used to obtain a p-value. The 'Emulated Wald test' attempts to provide a confidence interval for the difference compatible with the p-value. As mentionned in the litterature (Zou, 2007), a 'naive' back-transformation of the difference would not provide confidence intervals with good frequentist properties (intuitively $tanh(atanh(y) - atanh(x)) \neq y - x$). Instead samples are drawn from a bivariate Student's t distribution distribution centered around 0 and with variance-covariance matrix the inverse of the observed information on the `atanh` scale.

- `p.value`: relative frequency of a difference in simulated correlations more extreme than observed. It should be close to the p-value of the Wald F-test'.
- `se`: standard deviation of the simulated difference in correlation on the original scale
- `lower`, `upper`: quantiles of the simulated difference in correlation on the original scale after centering the simulated values on the `atanh` scale around the estimated correlation.

The `partialCor` method provides a more straightforward syntax to do the later test is:

```
set.seed(1)
partialCor(bmd1 + bmd5 ~ 1, data = calciumW.NNA, by = "grp", effects = "Dunnett")
```

```
      estimate     se df  lower upper p.value
C - P  -0.0574 0.0491 NA -0.165 0.029   0.195
```

The methodology is the same, except that the underlying mixed model is based on two timepoints (1 and 5) instead of all timepoints (1,2,3,4,5).

It is also possible to not use any transformation:

```
testRho <- anova(eCor2.lmm, effects = "rho(1,5):C - rho(1,5):P = 0", transform.rho = "none")
summary(testRho, print = TRUE)
```

```
               Wald F-test

      statistic      df p.value
   all     1.542 (1,3.7)   0.288
   ------------------------------
    :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
  df: Satterthwaite approximation w.r.t. model-based se.

               Hypothesis-specific Wald test

                               estimate    se  df lower upper p.value
   rho(1,5):C - rho(1,5):P = 0   -0.057 0.046 3.7 -0.19 0.076   0.288
   ------------------------------------------------------------------
    :  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 is expected to have worse small sample properties compared to using a transformation. In this example the estimated p-value is also further away from the Fisher'z test. Here the 'Hypothesis-specific Wald test' uses a Student's t-distribution to model the distribution of the ratio between the estimate and the standard error. This is exactly the square root (up to a sign) of the Wald F-test test statistic, leading to exactly the same p-value and compatible confidence intervals.

To retrieve these results with a linear regression, one should model a different variance in each group:

```
e.lmCorG <- lmm(bmd1 ~ grp + bmd5:grp, data = calciumW.NNA, structure = IND(~grp))
eI.lmCorG <- model.tables(e.lmCorG)[c("grpP:bmd5","grpC:bmd5"),]
eI.lmCorG
```

```
          estimate       se     df   lower   upper    p.value
grpP:bmd5  0.83283 0.053999 45.009 0.72407 0.94159 0.0000e+00
grpC:bmd5  0.74604 0.068412 42.008 0.60798 0.88410 7.9048e-14
```

```
eI.lmCorG$estimate/sqrt(eI.lmCorG$estimate^2+eI.lmCorG$se^2*eI.lmCorG$df)
```

```
[1] 0.91700 0.85963
```

3.3 Comparing Pearson's partial correlations

To evaluate the Pearson's correlation after regressing-out the effect of the covariate, one can do it explicitely in two steps:

```
## regress-out covariate effect
calciumW.NNA$bmd4adj <- residuals(lm(bmd4 ~ grp + grp:bmd2, data = calciumW.NNA))
calciumW.NNA$bmd5adj <- residuals(lm(bmd5 ~ grp + grp:bmd2, data = calciumW.NNA))
## evaluate correlations
with(calciumW.NNA, cor(bmd4adj[grp=="P"],bmd5adj[grp=="P"]))
with(calciumW.NNA, cor(bmd4adj[grp=="C"],bmd5adj[grp=="C"]))
```

```
[1] 0.8312
[1] 0.69114
```

or in a single step via `partialCor`:

```
ePcor <- partialCor(bmd4 + bmd5 ~ bmd2, data = calciumW.NNA, by = "grp")
print(ePcor, digits = 5)
```

```
  estimate       se     df   lower   upper    p.value
P  0.83120 0.046080 51.263 0.71275 0.90355 1.4225e-10
C  0.69114 0.080597 35.778 0.49080 0.82206 3.2049e-06
```

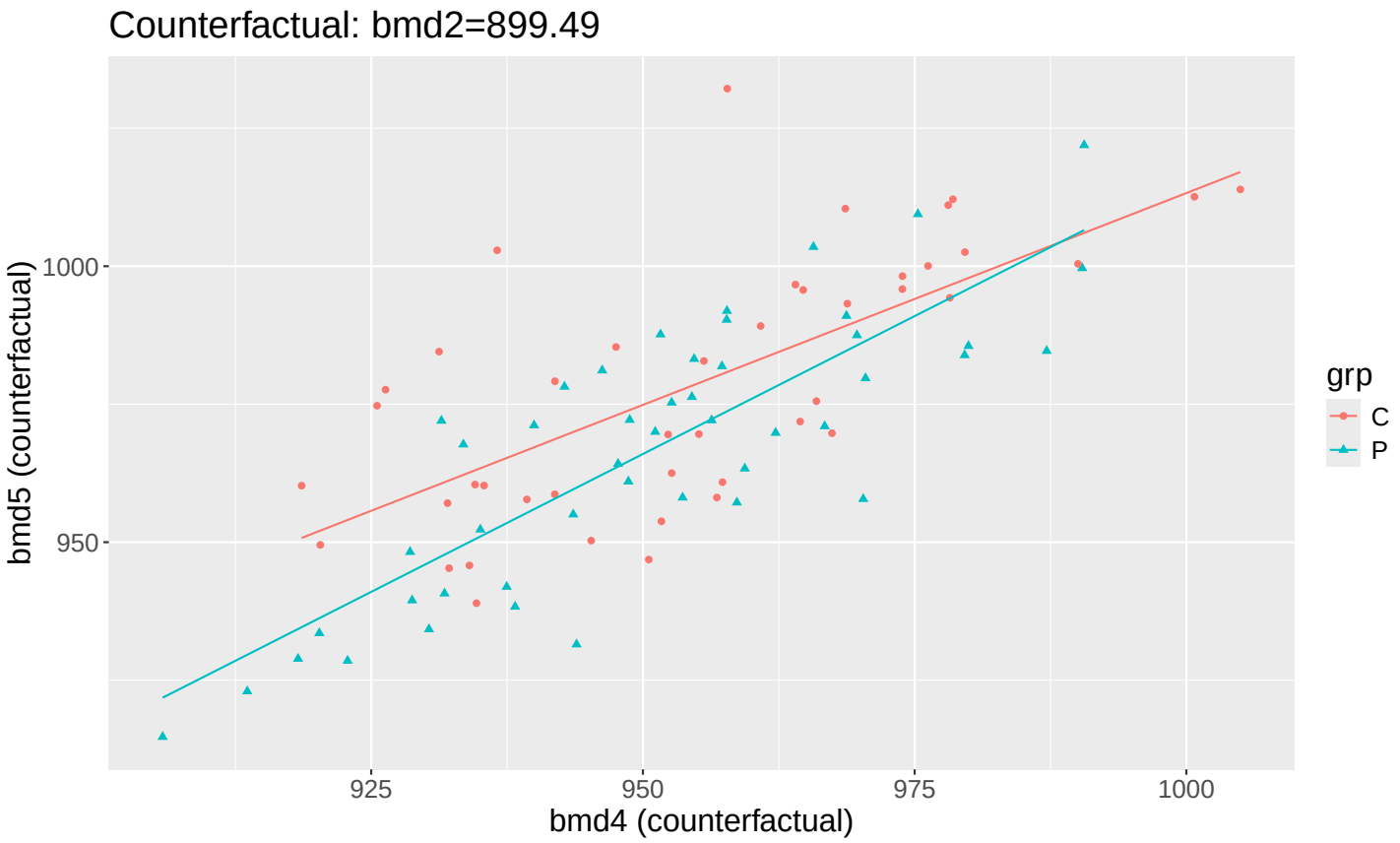
The estimates are the same but the later approach also provides confidence intervals and p.values. The function internally move the dataset to the long format and fit a linear mixed model with a mean, covariate effect, and variance specific to each measurement (here `CCvariableCC` is `bmd1` or `bmd5`).

```
model.tables(attr(ePcor,"lmm")$model$P, effects = "all")
```

```
                     estimate        se     df     lower     upper    p.value
(Intercept)          48.18469 36.685766 45.007 -25.70392 122.07330 1.9569e-01
CCvariableCCbmd5     63.38119 24.518113 44.996  13.99907 112.76332 1.3048e-02
CCvariableCCbmd4:bmd2  1.00328  0.041053 45.007   0.92060   1.08597 0.0000e+00
CCvariableCCbmd5:bmd2  0.95059  0.049349 45.013   0.85119   1.04998 0.0000e+00
sigma                20.36981  2.147167 41.333  16.46485  25.20091         NA
k.bmd5                1.20207  0.099628 53.146   1.01798   1.41946 3.0659e-02
rho(bmd4,bmd5)        0.83120  0.046080 51.263   0.71275   0.90355 1.4225e-10
```

A graphical display of the two measurements, after regressing-out the covariate effects, and their relationship (under linearity assumption) can also be produced:

```
plot(ePcor)
```



A nearly identical estimate can be obtained using a linear regression, adjusting on group:

```
e.lmpCorG <- lmm(bmd4 ~ grp + bmd5:grp + bmd2:grp,
                 data = calciumW.NNA, structure = IND(~grp))
eI.lmpCorG <- model.tables(e.lmpCorG)[c("grpP:bmd5","grpC:bmd5"),]
eI.lmpCorG
```

```
estimate       se     df   lower   upper    p.value
grpP:bmd5  0.69147 0.069727 44.009 0.55094 0.83199 8.6198e-13
grpC:bmd5  0.62242 0.101649 41.008 0.41714 0.82771 2.8871e-07
```

and back-transforming the estimated regression parameter:

```
eI.lmpCorG$estimate/sqrt(eI.lmpCorG$estimate^2+eI.lmpCorG$se^2*eI.lmpCorG$df)
```

```
[1] 0.83117 0.69110
```

3.4 Correlation between changes

In some studies, one is interested in studying the relation between two evolutions. Say the change from baseline in quality of life vs. functioning assessment test:

```
abetaW$dqol <- abetaW$qol1 - abetaW$qol0
abetaW$dfast <- abetaW$fast1 - abetaW$fast0
abetaW.NNA <- abetaW[!is.na(abetaW$dqol) & !is.na(abetaW$dfast),]
```

One can evaluate their correlation:

```
cor.test(abetaW.NNA$dqol, abetaW.NNA$dfast)
```

```
        Pearson's product-moment correlation

data:  abetaW.NNA$dqol and abetaW.NNA$dfast
t = -4.27, df = 110, p-value = 4.2e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.52570 -0.20575
sample estimates:
     cor
-0.37692
```

or estimate the regression coefficient of one change against the other:

```
model.tables(lmm(dqol ~ dfast, data = abetaW.NNA))
```

```
            estimate      se     df    lower    upper    p.value
(Intercept)  1.34601 0.85087 110.02 -0.34022  3.03224 1.1654e-01
dfast       -0.49231 0.11535 110.02 -0.72091 -0.26371 4.1977e-05
```

To retrieve the same results using a linear mixed model, one should move the dataset to the very long format, where each type of measurement is treated as a separate outcome:

```
abetaL.NNA <- reshape(abetaW.NNA[,c("id","qol0","qol1","fast0","fast1")], direction = "long",
                      idvar = "id", varying = 2:5,
                      timevar = "type", times = c("qol0","qol1","fast0","fast1"), v.names = c(
   "value"))
abetaL.NNA <- abetaL.NNA[order(abetaL.NNA$id),]
rownames(abetaL.NNA) <- NULL
head(abetaL.NNA)
```

```
  id  type value
1  3  qol0    86
2  3  qol1    79
3  3 fast0    29
4  3 fast1    31
5  4  qol0    96
6  4  qol1   101
```

One can then jointly model the association between all type of measurement using an unstructured residual variance-covariance matrix:

```
e.lmm4 <- lmm(value ~ type,
              repetition = ~type|id, structure = "UN",
              data = abetaL.NNA)
sigma.lmm4 <- sigma(e.lmm4)
sigma.lmm4
```

```
        fast0   fast1    qol0    qol1
fast0 132.471  95.090 -97.958 -72.709
fast1  95.090 102.301 -75.656 -72.360
qol0  -97.958 -75.656 143.759  91.321
qol1  -72.709 -72.360  91.321 114.957
```

Deduce the residual covariance matrix for the change:

```
Mcon <- cbind(c(-1,1,0,0),c(0,0,-1,1))
sigmeChange.lmm4 <- t(Mcon) %*% sigma.lmm4 %*% Mcon
dimnames(sigmeChange.lmm4) <- replicate(2,c("dfast","dqol"), simplify = FALSE)
sigmeChange.lmm4
```

```
        dfast    dqol
dfast  44.592 -21.953
dqol  -21.953  76.075
```

and retrieve the corrrelation and regression coefficients:

```
cov2cor(sigmeChange.lmm4)[1,2]
sigmeChange.lmm4[1,2]/sigmeChange.lmm4[1,1]
```

```
[1] -0.37692
[1] -0.49231
```

The uncertainty can be quantified using a delta method:

```
estimate(e.lmm4, function(p){
  Sigma.change <- t(Mcon) %*% sigma(e.lmm4, p = p) %*% Mcon
  c(cor = cov2cor(Sigma.change)[1,2],
    beta = Sigma.change[1,2]/Sigma.change[1,1])
})
```

```
     estimate       se     df    lower    upper    p.value
cor  -0.37692 0.081429 12.075 -0.55421 -0.19962 0.00057192
beta -0.49231 0.114833 12.561 -0.74127 -0.24334 0.00095359
```

The standard error for the regression coefficient is close to the linear model one but the degrees of freedom seem grossly underestimated. One can set the argument `df` to `FALSE` when calling `estimate` to use a Gaussian instead of a Student's t distribution.

4 Test on the variance

4.1 Comparing variances

We can emulate a F-test comparing the variance between two populations:

```
var.test(x = calciumW.NNA[calciumW.NNA$grp=="C","bmd1"],
         y = calciumW.NNA[calciumW.NNA$grp=="P","bmd1"])
```

```
        F test to compare two variances

data:  calciumW.NNA[calciumW.NNA$grp == "C", "bmd1"] and calciumW.NNA[calciumW.NNA$grp == "P", "bmd
F = 0.666, num df = 43, denom df = 46, p-value = 0.18
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.36801 1.21107
sample estimates:
ratio of variances
           0.66559
```

using an heteroschedastic linear regression with a parameter for the residual standard deviation in the reference group (σ) and a parameter for the ratio in standard deviation between the two groups (k):

```
eVar2.lmm <- lmm(bmd1 ~ grp, structure = IND(~grp), data = calciumW.NNA)
coef(eVar2.lmm, effects = "variance")
```

```
   sigma      k.C
66.87928  0.81584
```

This leads to the following modeled group-sepecific residual standard deviations:

```
coef(eVar2.lmm, effects = "variance", transform.k = "sd")
```

```
sigma.P sigma.C
 66.879  54.563
```

Testing whether the k parameter is 1, i.e. its log is 0:

```
summary(anova(eVar2.lmm, effects = "variance"))
```

```
               Hypothesis-specific Wald test

                 estimate    se   df lower upper p.value
 variance: k.C=1    0.816 0.122 88.6 0.606 1.099   0.178
 ---------------------------------------------------------
  :  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).
Back-transformation: k parameters with exp.
```

leads to a similar p-value comapred to `var.test`. The estimate differs as `anova` returns the ratio of the residual standard deviations instead of the ratio of the residual variances. The later can be obtained using:

```
summary(anova(eVar2.lmm, effects = "variance", transform.k = "logsquare"))
```

```
               Hypothesis-specific Wald test

                estimate  se   df lower upper p.value
 variance: k.C=1    0.666 0.2 88.6 0.367 1.208   0.178
 ---------------------------------------------------------
  :  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).
Back-transformation: k parameters with exp.
```

closely matching the output of `var.test`. This test is the special case of the Bartlett test:

```
bartlett.test(bmd1 ~ grp, data = calciumW.NNA)
```

```
        Bartlett test of homogeneity of variances

data:  bmd1 by grp
Bartlett's K-squared = 1.8, df = 1, p-value = 0.18
```

which generalizes to more than two variances:

```
bartlett.test(age ~ sex.group,
              data = transform(abetaW.NNA, sex.group = paste0(sex,group)))
```

```
        Bartlett test of homogeneity of variances

data:  age by sex.group
Bartlett's K-squared = 1.68, df = 3, p-value = 0.64
```

An F-test from the corresponding heteroschedastic linear regression leads to the same results:

```
eVar4.lmm <- lmm(age ~ sex.group, structure = IND(~sex.group),
                 data = transform(abetaW.NNA, sex.group = paste0(sex,group)))
summary(anova(eVar4.lmm, effects = "variance"))
```

```
               Wald F-test

                     statistic       df p.value
   variance: sex.group     0.566 (3,53.1)    0.64
   ------------------------------------------------
    :  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1.
```

```
df: Satterthwaite approximation w.r.t. model-based se.
Multiple testing adjustment: joint test.

              Hypothesis-specific Wald tests

                   estimate    se   df lower upper p.value
 variance: k.FHC=1    1.143 0.241 34.3 0.681 1.919   0.874
           k.MBD=1     0.92 0.157 68.5 0.606 1.398   0.935
           k.MHC=1    1.131  0.22 47.2 0.701 1.822   0.876
 ---------------------------------------------------------
  :  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).
Multiple testing adjustment: max test (1e+05 samples).
Back-transformation: k parameters with exp.
```

Note that the test statistic of `anova` multiplied by its (numerator) degree of freedom `0.566*3` leads to the test statistic of `bartlett.test`.

⚠ when considering variance of time instead of variance between groups the equivalence is typically lost as `lmm` can account for within-subject correlation (argument `structure` set to `UN`) while `bartlett.test` cannot.

References

Vickers, A. J. and Altman, D. G. (2001). Analysing controlled trials with baseline and follow up measurements. *Bmj*, 323(7321):1123–1124.

Zou, G. Y. (2007). Toward using confidence intervals to compare correlations. *Psychological methods*, 12(4):399.