

# Mixed Model Workshop part 2

Lorna Le Stanc

10/22/2021

## Contents

<b>1</b>	<b>Goal of this workshop session</b>	<b>1</b>
<b>2</b>	<b>Needed libraries</b>	<b>1</b>
<b>3</b>	<b>Dataset for the workshop</b>	<b>2</b>
<b>4</b>	<b>Mixed models (continous outcome, discreate predictors)</b>	<b>2</b>
4.1	Data format . . . . .	2
4.2	Random structure = Structure in the data that is not explained by fixed effects . . . . .	2
4.3	Notation . . . . .	5
4.4	When not to interpret ever eveR evER eVER EVER your model . . . . .	5
4.5	How to choose your random structure . . . . .	6
4.6	Interpreting . . . . .	10
4.7	Get main effects (and p-values) using Anova from rstatix . . . . .	13
4.8	Get p-values style 2 = multcomp package corrects for multiple comparisons . . . . .	13
4.9	Get main effects using contrast coding and p-values using multcomp . . . . .	13
4.10	Get main effect using model comparison . . . . .	14

## 1 Goal of this workshop session

In this part two we will learn how to format our data to do mixed models. We will learn what is the random structure and how to interpret the output of mixed models in R.

## 2 Needed libraries

```
# to perform planned contrasts & post-hocs /\ load before tidyverse to prevent  
# masking its 'select' function  
library(multcomp)  
  
# to manipulate and plot data  
library(tidyverse)  
  
# to do mixed models  
library(lme4)  
  
# to obtain main effects from linear models and linear mixed models  
library(rstatix)
```

### 3 Dataset for the workshop

Here we will work on reaction times (continuous variable)

- Two groups (control vs test)
- Two sessions (T0 vs T1)
- 4 items per sessions (each with 5 trials) -> 20 trials per participant and session.

This is a 2x2 design, with group as a between-subjects variables and session as a within-subject variable.

The dataset was built such as :

```
# Number of subjects per groups (2 groups, control/test)
N = 30
# number of trials per subject
ntrials = 20
# reaction time at T0 for both groups (ms)
int.T0 = 950
# Retest effect
slope.retest = 100
# Training effect
slope.training = 150
# variance among subjects at T0
sd = 50
# variance among retest and training effects is sd/2
```

### 4 Mixed models (continuous outcome, discrete predictors)

Handle nested designs, repeated measures, unbalanced data sets, missing values, more than one source of structure in the residuals.

Indeed, repeated measure ANOVAs allow you to take into account one source of repetition in your data. Usually its participants.

#### 4.1 Data format

Contrary to ANOVA's, you do not use collapsed data for mixed models. Instead you use data such as there is one line per trials. You will have several lines per participant and session.

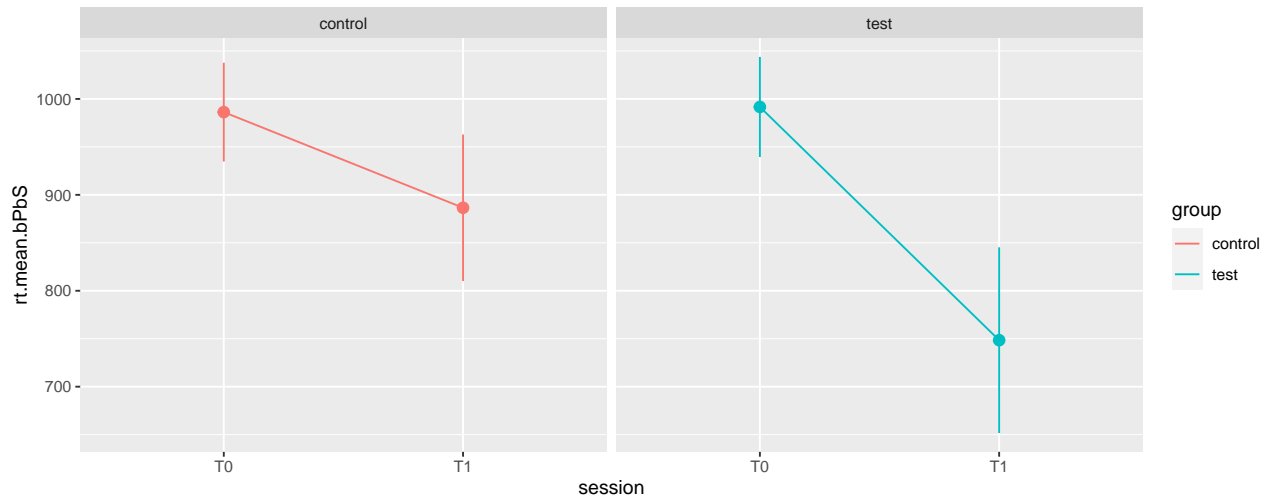
```
head(data)

## # A tibble: 6 x 6
##   participant    group session trial      rt item
##   <chr>         <chr>   <chr>  <chr>  <dbl> <chr>
## 1 sub-control001 control T0      trial1  889. item1
## 2 sub-control001 control T0      trial2  917. item1
## 3 sub-control001 control T0      trial3  881. item1
## 4 sub-control001 control T0      trial4  884. item1
## 5 sub-control001 control T0      trial5  929. item1
## 6 sub-control001 control T0      trial6  873. item2
```

#### 4.2 Random structure = Structure in the data that is not explained by fixed effects

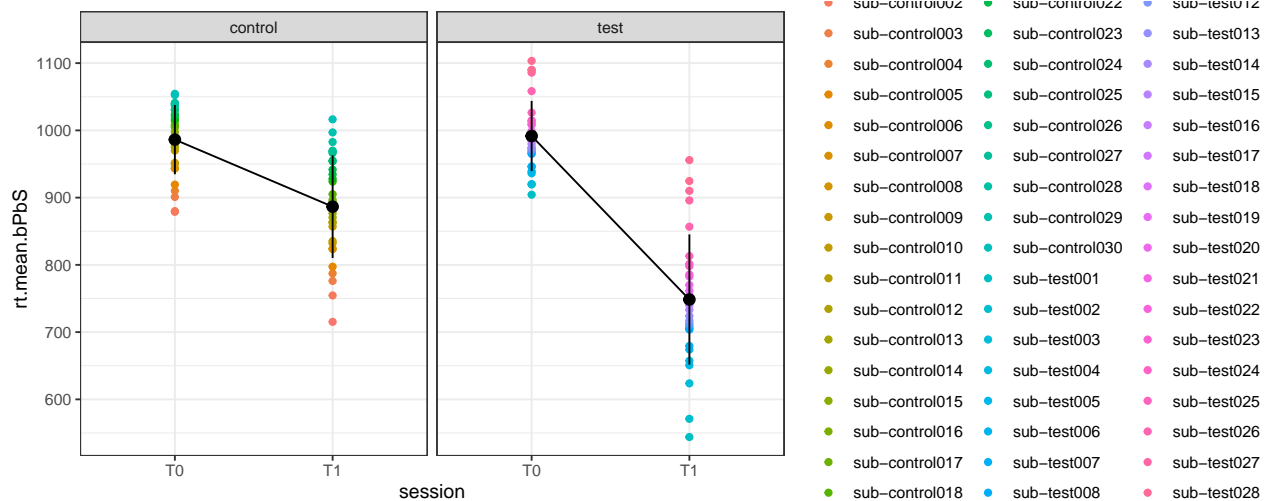
#### 4.2.1 Mean effect of session for each group -> fixed effect of session

As we've seen in part one of the workshop, reaction times are faster at T1 (retest effect for the control group and retest + training effect in the test group).



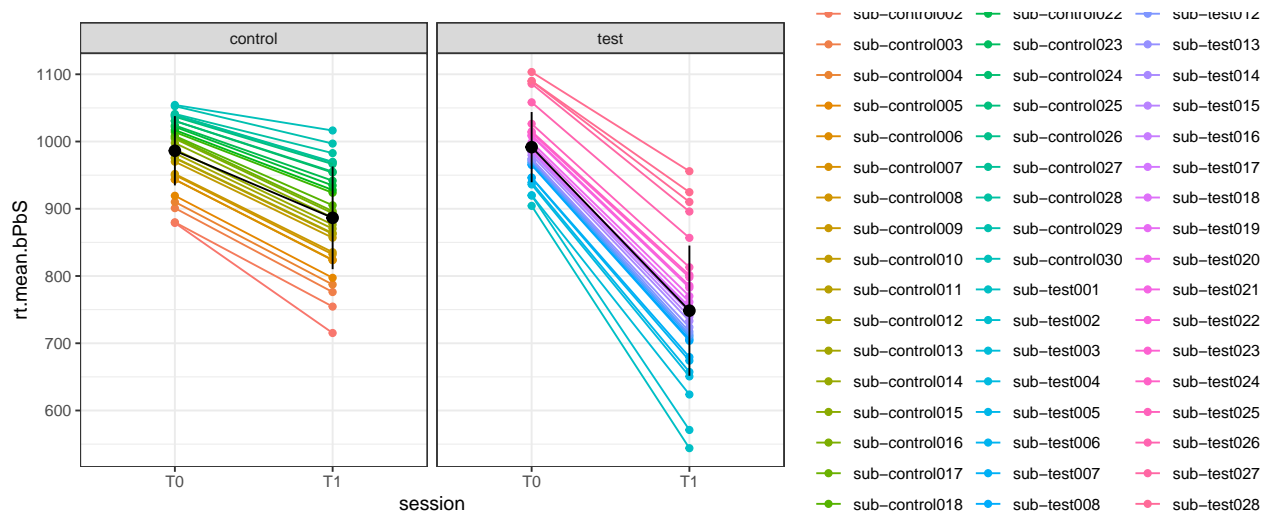
#### 4.2.2 Random intercept

Do all participants start at the same level at T0 ? -> Random intercept for participant



#### 4.2.3 Random slope

Do all participants react in the same way to the training ? -> random session slope for participants



#### 4.2.4 Correlated random intercept and slope

Are the participants start-level linked to how much they benefit from the training ? -> correlated random intercept and slope

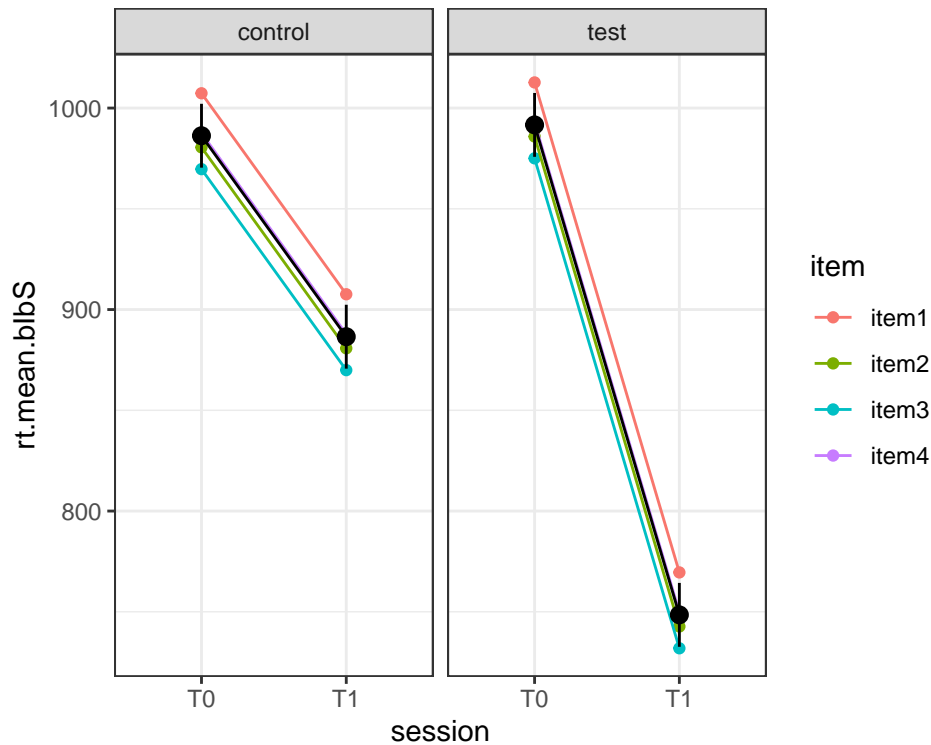
```
data.cor = data.mean%>%
  pivot_wider(names_from = session, values_from = rt.mean.bPbS)%>%
  mutate(training = T1-T0)%>%
  select(-T1,-group,-participant)

cor.test(data.cor$T0, data.cor$training)

##
## Pearson's product-moment correlation
##
## data: data.cor$T0 and data.cor$training
## t = 2.9722, df = 58, p-value = 0.004299
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.1207899 0.5653009
## sample estimates:
## cor
## 0.3635635
```

#### 4.2.5 Random structure for items.

- Are some items easier than others (-> random intercept for items) ?
- Are some items more “trainable” than others (-> random session slope for items) ?
- Do groups react in the same way to the items (-> random group slope for items) ?



### 4.3 Notation

- $(1|s)$  = random intercept
- $(0+x|s)$  = random slope
- $(1|s) + (0+x|s)$  = uncorrelated random intercept and random slope
- $(1+x|s)$  = correlated random intercept and random slope

## 4.4 When not to interpret ever ever EVER EVER your model

### 4.4.1 Lack of convergence

The process to fit a model consist of multiple iterations.

The process first attributes initial values to the parameters to estimate (here intercept and slopes). Then it computes the criteria to estimate whether the model is a good fit to the data (for example the maximum likelihood). Then it iterates this two steps by adjusting the parameters and recomputing the criteria. When the criteria reaches an appropriate value then the model is said to converge.

When a model fail to converge, it means that the iterations to find the estimates does not reach a stable solution or that it is taking too long. The estimates are not trustworthy and you should not interpret the results.

The message you get in R will be :

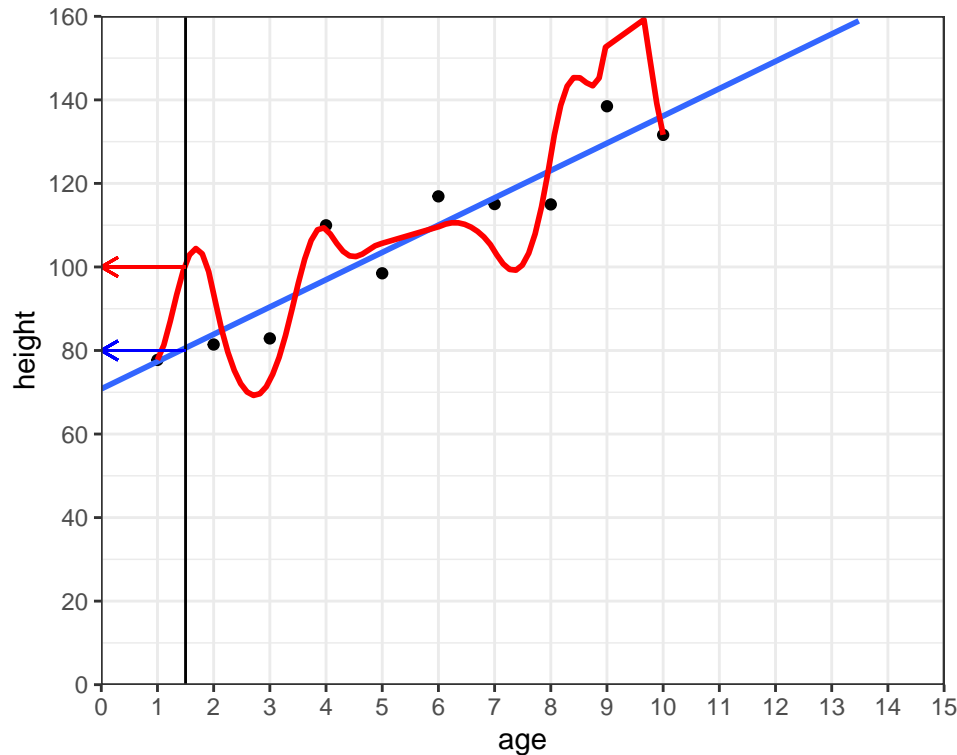
```
lme.fail = lmer(rt ~ group*session+ (1+session|participant)+ (1+group|trial),
               data = data) # failed
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 2 negative eigenvalues
```

#### 4.4.2 Overfitting

The goal of a model is to be able to generalize your data and to draw conclusion on what would be the outcome given a new set of predictors. When you over fit your data, the model is too close to the data, predicts perfectly each data point of your given data set but will not generalize to a new set of data.



In blue, the true fit to the data, in red the overfit.

If you want to know what would be the height at 1.5 years old, you draw a vertical line at  $x = 1.5$ . The  $y$  for which it crosses the line drawn by the model tells you what is the predicted age according to that model.

You can see that the red line does not give you a generalisation for the data set, the red model overfits the data.

The message you would see in R is :

```
lme.singular = lmer(rt ~ group*session+ (1|participant)+ (1+group|trial),
                    data = data)
```

```
## boundary (singular) fit: see ?isSingular
```

#### 4.5 How to choose your random structure

There is no scientific consensus on how to do that. I will give you two methods. For sure you will find other methods in the literature.

For example, for method one (starting with maximal), if you built your material selecting the items such as no item was easier then you can justify not to include item in the random structure.

#### 4.5.1 Start with maximal and simply

**4.5.1.1 How to find your maximal structure** Ask yourself what in your design could have a structure that you would want to control for.

Here we've seen previously here: participants and items.

To determine the maximal structure regarding the slopes, ask yourself what is measured repeatedly across participants and then across items.

Here participants are measured across sessions and items across groups and sessions.

Put everything correlated and don't forget interactions between random slopes.

```
lme.full = lmer(rt ~ group*session + (1+session|participant) + (1 + group*session|item),  
               data = data)
```

```
## boundary (singular) fit: see ?isSingular
```

#### 4.5.1.2 Decorrelate and simplify until convergence and no overfitting.

	maximal	(1+x1*x2 s)
uncorrelated intercept and slope		(1 s) + (0+x1*x2 s)
no interaction		(1+x1+x2 s)
uncorrelated intercept and slope, no interaction		(1 s) + (0+x1+x2 s)
no x2		(1+x1 s)
uncorrelated intercept and slope, no x2		(1 s) + (0+x1 s)
no x1		(1+x2 s)
uncorrelated intercept and slope, no x1		(1 s) + (0+x2 s)
intercept only		(1 s)

Look at your data to choose and to the variance in the random effects.

Here I started to simplify the items before the standard deviation for the random intercept and slopes for items was very low (see "Random effects" lines).

```
lme.full  
  
## Linear mixed model fit by REML ['lmerMod']  
## Formula: rt ~ group * session + (1 + session | participant) + (1 + group *  
##   session | item)  
##   Data: data  
## REML criterion at convergence: 20776.63  
## Random effects:  
##   Groups      Name                Std.Dev.  Corr  
## participant (Intercept)          5.172e+01  
##           sessionT1              3.725e+01 0.91  
## item        (Intercept)          1.583e+01  
##           grouptest              9.405e-04 -1.00  
##           sessionT1              1.530e-03 -0.90 0.90  
##           grouptest:sessionT1    1.922e-03 0.97 -0.97 -0.94  
## Residual                        1.656e+01  
## Number of obs: 2400, groups: participant, 60; item, 4  
## Fixed Effects:  
##           (Intercept)            grouptest            sessionT1  
##           986.272                5.373                -99.746  
## grouptest:sessionT1
```

```
##                -143.428
## optimizer (nloptwrap) convergence code: 0 (OK) ; 0 optimizer warnings; 1 lme4 warnings
lme = lmer(rt ~ group*session + (1+session|participant) + (1|item) + (0+group*session|item),
           data = data)
# singular
lme = lmer(rt ~ group*session + (1+session|participant) + (1+group+session|item),
           data = data)
# singular
lme = lmer(rt ~ group*session + (1+session|participant) + (1|item) + (0+group+session|item),
           data = data)
# singular
lme = lmer(rt ~ group*session + (1+session|participant) + (1+group|item),
           data = data)
# singular
lme = lmer(rt ~ group*session + (1+session|participant) + (1|item) + (0+group|item),
           data = data)
# singular

lme.conv = lmer(rt ~ group*session + (1+session|participant) + (1|item),
                data = data)
```

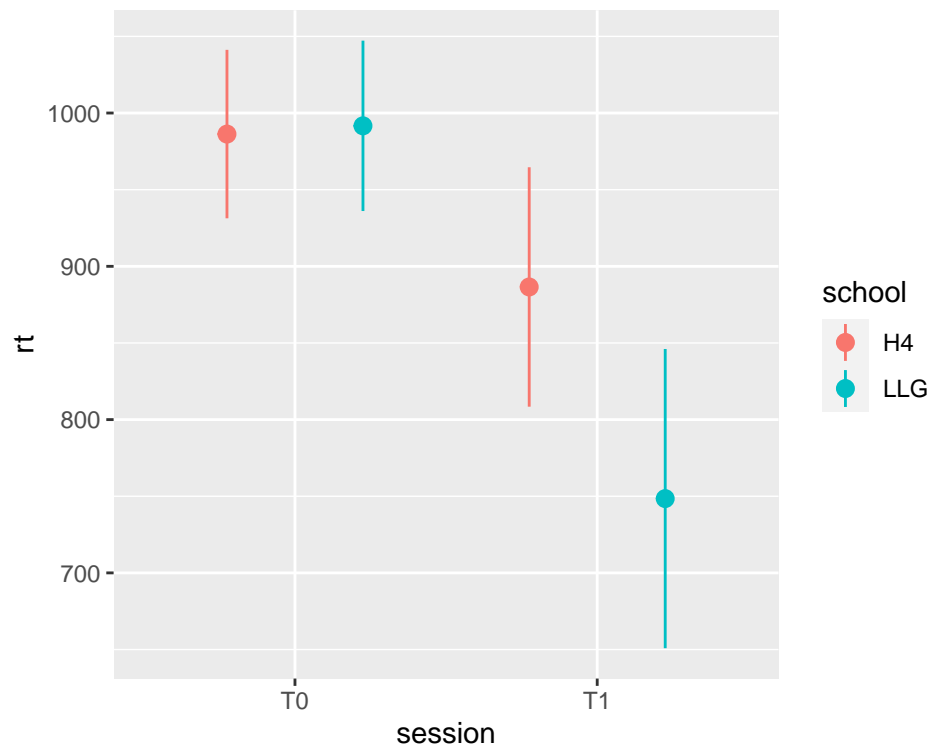
#### 4.5.2 Or think about your question

Let's now say that the two groups were instead two schools.

```
data.school = data%>%
  mutate(school = ifelse(group == "control", "H4", "LLG"))

ggplot(data.school, aes(x = session, y = rt, color = school)) +
  stat_summary(aes(y = rt, x = session), geom = "pointrange",
              fun = mean,
              fun.min = function(x) mean(x) - sd(x),
              fun.max = function(x) mean(x) + sd(x),
              position = position_dodge(width = 0.9))
```





If we are interested in the effect of training, we might specify a multilevel model, because we really care about the effect of training and want to generalize it among schools. This model would be specified as:

```
lme.trainingcontroledforschools = lmer(rt ~ session + (1|participant) +(1+session|school),
                                         data = data.school)
```

```
## boundary (singular) fit: see ?isSingular
```

```
Anova(lme.trainingcontroledforschools)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
```

```
##
```

```
## Response: rt
```

```
##           Chisq Df Pr(>Chisq)
```

```
## session 5.7201  1    0.01677 *
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
coefficients(lme.trainingcontroledforschools)$school
```

```
##      (Intercept) sessionT1
```

```
## H4      988.9584  -99.7878
```

```
## LLG      988.9584 -243.1319
```

This is particularly helpful if there are a large number of schools we are testing this hypothesis over.

However, we might be interested in the school specific effects of training. In that case, they could specify school as a fixed effect:

```
lme.traininginteractionschools = lmer(rt ~ school*session + (1|participant),
                                         data = data.school)
```

```
Anova(lme.traininginteractionschools)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
```

```
##
## Response: rt
##               Chisq Df Pr(>Chisq)
## school         13.776  1  0.0002059 ***
## session        21731.005  1  < 2.2e-16 ***
## school:session  3801.554  1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 4.6 Interpreting

Back to groups and sessions.

```
summary(lme.conv)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ group * session + (1 + session | participant) + (1 | item)
## Data: data
##
## REML criterion at convergence: 20776.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.87953 -0.67405  0.00039  0.84360  2.04956
##
## Random effects:
## Groups      Name                Variance Std.Dev. Corr
## participant (Intercept) 2675.7    51.73
##              sessionT1   1387.8    37.25  0.91
## item        (Intercept)  250.8    15.84
## Residual                274.3    16.56
## Number of obs: 2400, groups: participant, 60; item, 4
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    986.272    12.343  79.904
## grouptest       5.373     13.390   0.401
## sessionT1      -99.746     6.868 -14.523
## grouptest:sessionT1 -143.428     9.713 -14.766
##
## Correlation of Fixed Effects:
##              (Intr) grptst sssnT1
## grouptest   -0.542
## sessionT1    0.688 -0.634
## grptst:ssT1 -0.486  0.896 -0.707
```

First line : tells you that you fitted a linear mixed model and the method used. REML stands for Restricted Maximum Likelihood Estimation (see <http://users.stat.umn.edu/~gary/classes/5303/handouts/REML.pdf>)

“**Formula**” line: tells you the model fitted to your data

“**Data**” line : tells you the dataset used to fit the model

“**Scaled Residuals**” : minimum, first quartile, median, third quartile and maximum of scale residuals.

“**Random effects**” : details the variance for each random intercept and slope included in the model.

Here the first *Group* is participant for which we fitted an Intercept and a random slope for session. The second *Group* is item for which we only fitted a random intercept.

The *Corr* column represents the correlation between the intercept and slope.

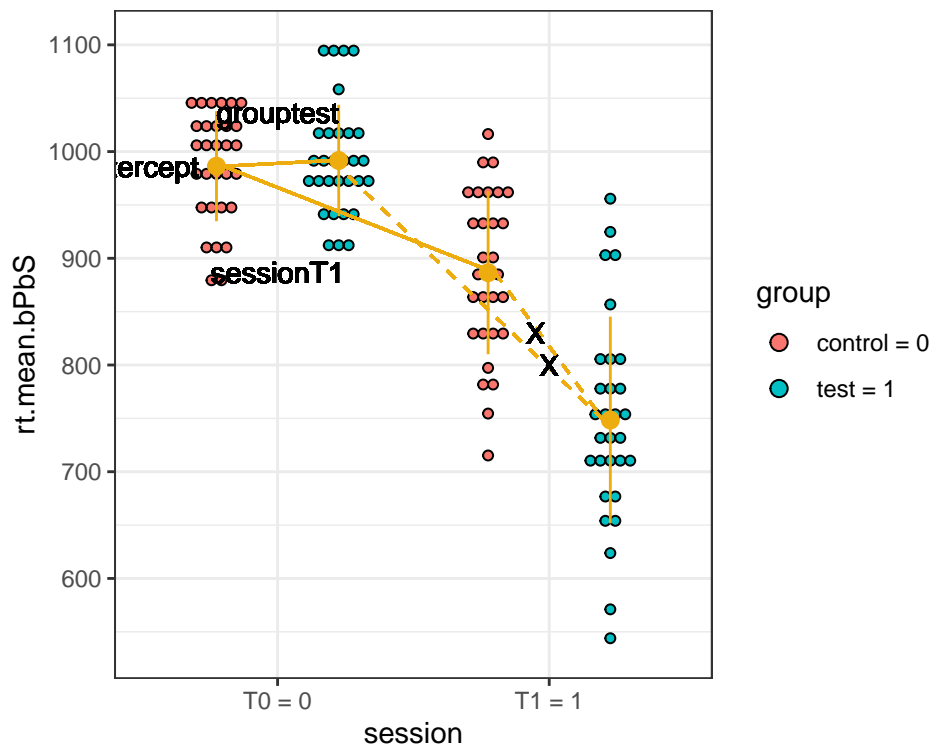
“**Number of obs**” tells you the total number of data points, the number of participants and items.

“**Fixed effects**” tells you the estimate for the Intercept, slopes and interaction as we have seen for linear models in part one of this workshop.

Note that there are no p-values in the output of linear mixed models using lme4 package because p-values are challenged by the developers of the package (and they are not alone). We will see how to get them. In the mean time, note that a t value < -2 or > 2 will be significant.

“**Correlation of Fixed Effects**” is not about the correlation of the variables. It is in fact about the expected correlation of the regression coefficients. Although this may speak to multicollinearity it does not necessarily. It is telling you that if you did the experiment again and it so happened that the coefficient for one slope got smaller, it is likely that it would affect the other slope.

As we have seen for linear models in part 1 of the workshop. The output gives you fixed effect rather than main effects and you should be careful when you write up your results and when you interpret the output of the model.



You can get the random slopes and intercepts from the model.

In our model : - for participant you will have a different intercept and slope for session for each participant. The slope for session and the interaction will be constant as they are not (and cannot be) in the random structure for participant - for item you will only have a different intercept for each item as we removed all other slopes to allow the model to converge.

```
coefficients(lme.conv)

## $participant
##           (Intercept) grouptest  sessionT1 grouptest:sessionT1
## sub-control001      879.9592  5.372883 -165.019172          -143.4277
```

## sub-control002	883.9951	5.372883	-131.624450	-143.4277
## sub-control003	903.8898	5.372883	-129.260115	-143.4277
## sub-control004	912.6744	5.372883	-126.692433	-143.4277
## sub-control005	921.4211	5.372883	-125.161558	-143.4277
## sub-control006	944.2994	5.372883	-121.151176	-143.4277
## sub-control007	944.3629	5.372883	-120.704460	-143.4277
## sub-control008	950.6581	5.372883	-118.716568	-143.4277
## sub-control009	952.4016	5.372883	-117.282077	-143.4277
## sub-control010	969.8348	5.372883	-112.792466	-143.4277
## sub-control011	974.7847	5.372883	-111.930051	-143.4277
## sub-control012	975.5498	5.372883	-111.498932	-143.4277
## sub-control013	981.7233	5.372883	-110.867270	-143.4277
## sub-control014	987.1863	5.372883	-110.077514	-143.4277
## sub-control015	996.1508	5.372883	-108.389848	-143.4277
## sub-control016	1002.2549	5.372883	-107.798387	-143.4277
## sub-control017	1004.4411	5.372883	-106.844122	-143.4277
## sub-control018	1006.7414	5.372883	-100.711765	-143.4277
## sub-control019	1013.2249	5.372883	-89.033344	-143.4277
## sub-control020	1016.1343	5.372883	-87.757897	-143.4277
## sub-control021	1020.7842	5.372883	-85.635856	-143.4277
## sub-control022	1023.1809	5.372883	-81.119239	-143.4277
## sub-control023	1029.8654	5.372883	-75.731777	-143.4277
## sub-control024	1030.0678	5.372883	-74.623187	-143.4277
## sub-control025	1036.3261	5.372883	-69.744884	-143.4277
## sub-control026	1038.2846	5.372883	-69.226556	-143.4277
## sub-control027	1038.8862	5.372883	-68.843600	-143.4277
## sub-control028	1041.2474	5.372883	-58.800589	-143.4277
## sub-control029	1052.3715	5.372883	-55.440971	-143.4277
## sub-control030	1055.4554	5.372883	-39.899682	-143.4277
## sub-test001	894.4826	5.372883	-209.481081	-143.4277
## sub-test002	909.8330	5.372883	-197.839888	-143.4277
## sub-test003	914.5834	5.372883	-152.413014	-143.4277
## sub-test004	930.5406	5.372883	-141.499504	-143.4277
## sub-test005	933.6295	5.372883	-138.025694	-143.4277
## sub-test006	940.8201	5.372883	-128.736803	-143.4277
## sub-test007	941.7795	5.372883	-124.542717	-143.4277
## sub-test008	959.7505	5.372883	-117.985310	-143.4277
## sub-test009	960.9981	5.372883	-116.908599	-143.4277
## sub-test010	963.6971	5.372883	-115.710566	-143.4277
## sub-test011	965.9312	5.372883	-114.863354	-143.4277
## sub-test012	968.5109	5.372883	-113.243573	-143.4277
## sub-test013	969.3382	5.372883	-107.653594	-143.4277
## sub-test014	974.7739	5.372883	-103.788243	-143.4277
## sub-test015	978.8932	5.372883	-101.395585	-143.4277
## sub-test016	980.8237	5.372883	-96.941082	-143.4277
## sub-test017	983.1634	5.372883	-93.926563	-143.4277
## sub-test018	983.4231	5.372883	-91.760556	-143.4277
## sub-test019	989.8391	5.372883	-90.779113	-143.4277
## sub-test020	994.3065	5.372883	-86.640125	-143.4277
## sub-test021	1003.2694	5.372883	-82.955971	-143.4277
## sub-test022	1004.6797	5.372883	-81.360947	-143.4277
## sub-test023	1008.7987	5.372883	-73.381043	-143.4277
## sub-test024	1010.6730	5.372883	-71.888254	-143.4277
## sub-test025	1021.3781	5.372883	-70.639054	-143.4277

```
## sub-test026      1052.4658  5.372883  -57.574664          -143.4277
## sub-test027      1079.3312  5.372883  -45.020353          -143.4277
## sub-test028      1083.9495  5.372883  -35.746264          -143.4277
## sub-test029      1085.2037  5.372883  -23.143477          -143.4277
## sub-test030      1099.2908  5.372883   -6.534954          -143.4277
##
## $item
##      (Intercept) grouptest sessionT1 grouptest:sessionT1
## item1    1007.3013  5.372883   -99.746          -143.4277
## item2     980.5172  5.372883   -99.746          -143.4277
## item3     969.7163  5.372883   -99.746          -143.4277
## item4     987.5528  5.372883   -99.746          -143.4277
##
## attr("class")
## [1] "coef.mer"
```

#### 4.7 Get main effects (and p-values) using Anova from rstatix

```
Anova(lme.conv)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: rt
##              Chisq Df Pr(>Chisq)
## group          946.59  1 < 2.2e-16 ***
## session       1246.38  1 < 2.2e-16 ***
## group:session  218.04  1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#### 4.8 Get p-values style 2 = multcomp package corrects for multiple comparisons

```
summary(glht(lme.conv))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lmer(formula = rt ~ group * session + (1 + session | participant) +
##      (1 | item), data = data)
##
## Linear Hypotheses:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) == 0      986.272    12.343  79.904 <1e-05 ***
## grouptest == 0         5.373     13.390   0.401  0.967
## sessionT1 == 0        -99.746     6.868 -14.523 <1e-05 ***
## grouptest:sessionT1 == 0 -143.428     9.713 -14.766 <1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

#### 4.9 Get main effects using contrast coding and p-values using multcomp

See part one of workshop for more details on contrast coding.

```
data.contrast = data %>%
  mutate(session.contrast = case_when(
    session == "T0" ~ -0.5,
    session == "T1" ~ 0.5
  ))%>%
  mutate(group.contrast = case_when(
    group == "control" ~ -0.5,
    group == "test" ~ 0.5
  ))

lme.conv.contrast = lmer(rt ~ group.contrast*session.contrast + (1+session|participant) + (1|item),
  data = data.contrast)
summary(glht(lme.conv.contrast))
```

```
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lmer(formula = rt ~ group.contrast * session.contrast + (1 +
## session | participant) + (1 | item), data = data.contrast)
##
## Linear Hypotheses:
##
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) == 0      903.228    11.940  75.647 < 1e-04 ***
## group.contrast == 0    -66.341    17.873  -3.712 0.000578 ***
## session.contrast == 0  -171.460     4.857 -35.304 < 1e-04 ***
## group.contrast:session.contrast == 0 -143.428     9.713 -14.766 < 1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

It does not always make sense to contrast code all your variables. Sometimes it makes sense to have a reference group.

Cf main +post-hocs vs proper asked question.

## 4.10 Get main effect using model comparison

When you compare two models (using same data and same random structure !!!), the output of the anova tells you whether one model explains the data better than the other.

There is a rule in modeling : choose the parsimonious model if the more complex one does not explain your data better.

Comparing models answer to the question “Does adding this factor significantly increase the understanding of the data ?”

Therefore comparing models can be used to know whether or not a factor significantly explains a part of your data. Some researchers use this to say there is a main effect of the factor.

### 4.10.1 Is there an interaction ?

```
lme.int = lmer(rt ~ group*session - group:session+ (1+session|participant) + (1|item),
  data = data)
lme.conv = lmer(rt ~ group*session + (1+session|participant) + (1|item),
  data = data)
anova(lme.conv, lme.int)
```

```
## refitting model(s) with ML (instead of REML)

## Data: data
## Models:
## lme.int: rt ~ group * session - group:session + (1 + session | participant) + (1 | item)
## lme.conv: rt ~ group * session + (1 + session | participant) + (1 | item)
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## lme.int      8 20909 20955 -10446    20893
## lme.conv     9 20817 20869 -10400    20799 93.606  1 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This gets more complicated if you have more than two levels in a factor because removing the factor removes all levels and therefore several interactions. You cannot test them separately.

#### 4.10.2 Is there a main effect of session ?

```
lme.int = lmer(rt ~ group*session - group + (1+session|participant) + (1|item),
              data = data)
lme.conv = lmer(rt ~ group*session + (1+session|participant) + (1|item),
               data = data)
anova(lme.conv, lme.int)
```

```
## refitting model(s) with ML (instead of REML)

## Data: data
## Models:
## lme.conv: rt ~ group * session + (1 + session | participant) + (1 | item)
## lme.int: rt ~ group * session - group + (1 + session | participant) + (1 | item)
##          npar    AIC    BIC logLik deviance Chisq Df Pr(>Chisq)
## lme.conv     9 20817 20869 -10400    20799
## lme.int      9 20817 20869 -10400    20799      0  0
```