Bilingualism

Data analysis

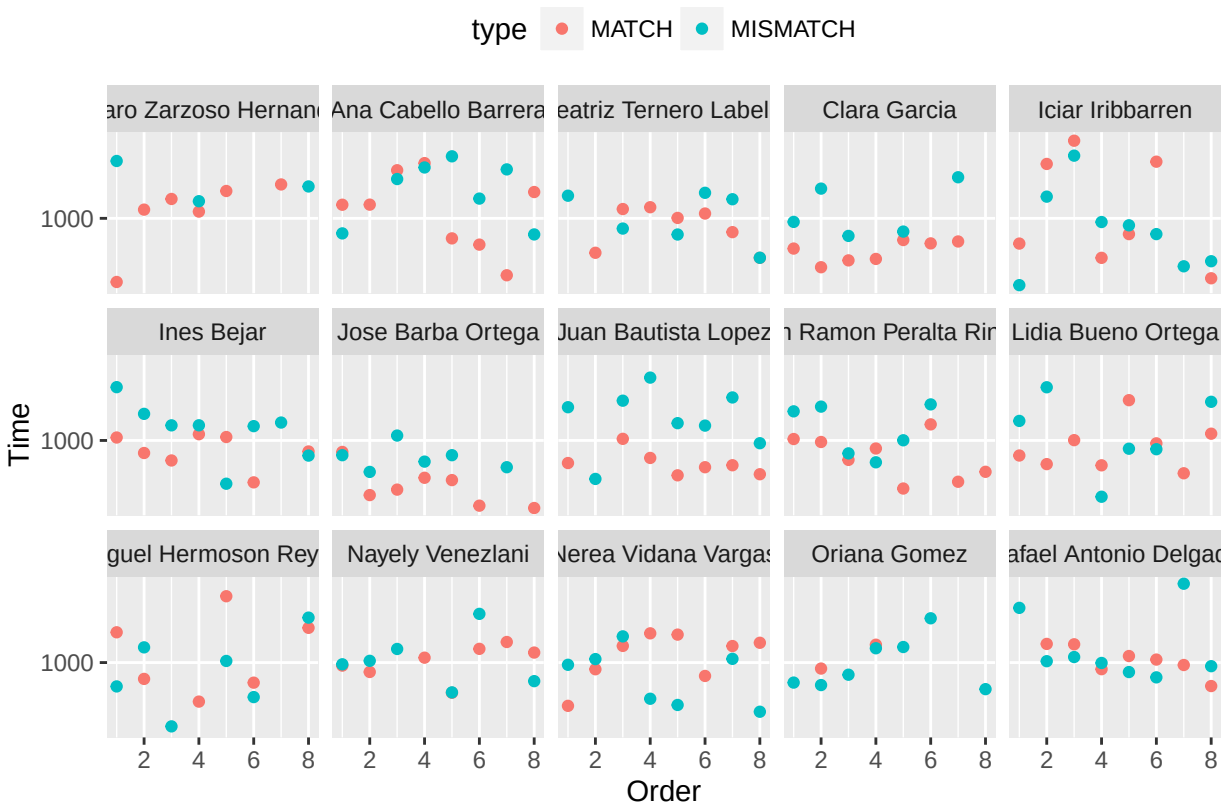
We load the data and remove the cases with NA values.

Let's have a glimpse at the data:

```
# tbl_df(DLM)
glimpse(DLM)
```

```
## Observations: 1,476
## Variables: 20
## $ Name   (fctr) Adriana Casa Pregal, Alvaro Vazquez Guisado, An...
## $ CoR    (int) 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, ...
## $ Hand   (fctr) 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ EO     (fctr) 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,...
## $ List   (fctr) 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ CEF    (fctr) 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,...
## $ SRRC   (int) 4, 4, 3, 3, 2, 3, 4, 4, 2, 3, 3, 3, 3, 3, 3, 5, ...
## $ PRE    (dbl) 61.10, 44.40, 11.10, 66.66, 38.90, 44.40, 100.00...
## $ POST1  (dbl) 94.4, 72.2, 100.0, 72.2, 100.0, 100.0, 100.0, 10...
## $ POST2  (dbl) 100.0, 100.0, 100.0, 100.0, 75.0, 12.5, 100.0, 1...
## $ STAY   (fctr) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ LEAYRS (dbl) 1.0, 4.0, 3.0, 2.0, 1.5, 3.0, 3.0, 1.0, 3.0, 5.0...
## $ HRSD   (dbl) 1.00, 5.00, 0.00, 1.00, 0.50, 0.00, 0.00, 2.00, ...
## $ RPV    (dbl) 3.0, 4.5, 3.5, 4.0, 2.5, 2.5, 2.0, 2.5, 4.0, 3.0...
## $ AMGE   (dbl) 4.5, 4.0, 3.5, 3.5, 3.5, 4.5, 3.5, 4.5, 3.5, 4.5...
## $ AMSP   (dbl) 4.0000, 4.0000, 3.3333, 3.3333, 3.6667, 4.3333, ...
## $ Order  (int) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ Time   (int) 1477, 2302, 1013, 591, 1137, 2087, 1573, 588, 27...
## $ ID     (int) 1, 3, 4, 5, 6, 7, 9, 10, 11, 13, 15, 18, 19, 20,...
## $ type   (fctr) MATCH, MATCH, MATCH, MATCH, MATCH, MATCH, MATCH...
```

The variable Order represents the question number. It will be treated as a factor (ie, qualitative factor) in the linear model analysis. We sample 15 persons and plot the log-transform of the response time on the questions with respect to



We construct the linear model. We log-transform time as it gives better dignostic plots. In the model we have 8 questions per each type, matched/nonmatched. Hence questions are nested within the type. We specify a model with random-effects associated to the subjects and to the questions nested in the type. These factors, subjects and questions are fully crossed, although we might have some missing data due to misclassification. In R, the linear mixed model looks as follows.

```
lm.0 <- lm(log(Time) ~ 1, data = DLM)
# LEAYRS AMSP AMGE
lmm.0 <- lmer(log(Time) ~ (1 | Name)+ (1 | type:List:Order), data = DLM)
AIC(lm.0,lmm.0)
```

```
##       df    AIC
## lm.0   2 2205.6
## lmm.0  4 1880.7
```

The log-Likelihood test confirms that the random-effects due to these two factors are statistically significant. We now focus on the fixed-effects in a forward stepwise search.

```
lmm.1 <- update(lmm.0, .~. + type)
anova(lmm.0,lmm.1)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + type
```

```
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1878 1904   -934     1868  0.11      1       0.74
```

```
lmm.2a <- update(lmm.0, .~. + HRSD)
lmm.2b <- update(lmm.0, .~.+LEAYRS)
anova(lmm.0,lmm.2a)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + HRSD
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1877 1904   -934     1867  0.74      1       0.39
```

```
anova(lmm.0,lmm.2b)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + LEAYRS
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1870 1896   -930     1860  8.04      1     0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmm.3a <- update(lmm.0, .~. + AMGE)
lmm.3b <- update(lmm.0, .~. + AMSP)
anova(lmm.0,lmm.3a)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + AMGE
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1878 1904   -934     1868     0      1       0.98
```

```
anova(lmm.0,lmm.3b)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + AMSP
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1877 1904   -934     1867  0.93      1       0.33
```

```
lmm.4 <- update(lmm.0, .~. + RPV)
anova(lmm.0,lmm.4)
```

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

```
## Data: DLM
## Models:
## object: log(Time) ~ (1 | Name) + (1 | type:List:Order)
## ..1: log(Time) ~ (1 | Name) + (1 | type:List:Order) + RPV
##        Df  AIC  BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## object  4 1876 1897   -934     1868
## ..1     5 1878 1904   -934     1868  0.31      1       0.57
```

Hence, the fixed effects of type, HRSD, RPV, AMGE, AMSP are all statistically significant. How about interactions?

```
lmm.5 <- lmer(log(Time) ~ (type + HRSD + LEAYRS + RPV + AMGE + AMSP) + (1 | Name)+ (1 | type:List:Order
anova(lmm.5)
```

```
## Analysis of Variance Table of type III  with  Satterthwaite
## approximation for degrees of freedom
##        Sum Sq Mean Sq NumDF DenDF F.value Pr(>F)
## type    0.019   0.019     1  29.7    0.11 0.7459
## HRSD    0.244   0.244     1 114.8    1.40 0.2386
## LEAYRS  1.921   1.921     1 106.7   11.07 0.0012 **
## RPV     0.015   0.015     1 110.1    0.09 0.7704
## AMGE    0.002   0.002     1 107.1    0.01 0.9206
## AMSP    0.142   0.142     1 110.7    0.82 0.3671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ft <- step(lmm.5)
ft
```

```
##
## Random effects:
##                 Chi.sq Chi.DF elim.num p.value
## Name            238.76      1     kept < 1e-07
## type:List:Order 102.09      1     kept < 1e-07
##
## Fixed effects:
##        Sum Sq Mean Sq NumDF  DenDF F.value elim.num Pr(>F)
## AMGE   0.0017  0.0017     1 107.14  0.0100        1 0.9206
## RPV    0.0149  0.0149     1 111.19  0.0861        2 0.7697
```

```
## type   0.0187  0.0187     1  29.72  0.1076        3 0.7452
## AMSP   0.1973  0.1973     1 112.05  1.1374        4 0.2885
## HRSD   0.5186  0.5186     1 117.19  2.9886        5 0.0865
## LEAYRS 1.4252  1.4252     1 110.76  8.2132     kept 0.0050
##
## Least squares means:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
##  Differences of LSMEANS:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
## Final model:
## lme4::lmer(formula = log(Time) ~ LEAYRS + (1 | Name) + (1 | type:List:Order),
##     data = DLM, REML = reml.lmerTest.private, contrasts = l.lmerTest.private.contrast,
##     devFunOnly = devFunOnly.lmerTest.private)
```

There are no significant interactions!!!! Yuuu!

```
lmm.6 <- lmer(log(Time) ~ (type + HRSD + LEAYRS + RPV + AMGE + AMSP)^2 + (1 | Name)+ (1 | type:List:Ord
ft<-step(lmm.6)
ft
```

```
##
## Random effects:
##                 Chi.sq Chi.DF elim.num p.value
## Name            234.57      1     kept < 1e-07
## type:List:Order 102.70      1     kept < 1e-07
##
## Fixed effects:
##             Sum Sq Mean Sq NumDF   DenDF F.value elim.num Pr(>F)
## type:RPV    0.0001  0.0001     1 1343.94  0.0004        1 0.9835
## RPV:AMSP    0.0015  0.0015     1   98.79  0.0089        2 0.9252
## LEAYRS:AMSP 0.0025  0.0025     1  105.22  0.0145        3 0.9044
## HRSD:AMGE   0.0022  0.0022     1  103.04  0.0124        4 0.9114
## RPV:AMGE    0.0159  0.0159     1  102.26  0.0917        5 0.7626
## type:AMGE   0.0276  0.0276     1 1340.50  0.1587        6 0.6904
## AMGE:AMSP   0.0326  0.0326     1  101.90  0.1876        7 0.6658
## HRSD:AMSP   0.0309  0.0309     1  104.15  0.1777        8 0.6742
## HRSD:LEAYRS 0.0431  0.0431     1  105.52  0.2482        9 0.6193
## type:AMSP   0.0586  0.0586     1 1345.10  0.3374       10 0.5614
## type:HRSD   0.1160  0.1160     1 1369.07  0.6684       11 0.4137
## AMSP        0.2121  0.2121     1  107.56  1.2229       12 0.2713
## LEAYRS:AMGE 0.2031  0.2031     1  104.11  1.1709       13 0.2817
## AMGE        0.0104  0.0104     1  105.31  0.0601       14 0.8068
## type:LEAYRS 0.2444  0.2444     1 1336.44  1.4090       15 0.2354
## type        0.0174  0.0174     1   29.63  0.1002       16 0.7538
## LEAYRS:RPV  0.3406  0.3406     1  108.71  1.9629       17 0.1641
## HRSD:RPV    0.2140  0.2140     1  108.07  1.2335       18 0.2692
## RPV         0.0688  0.0688     1  111.66  0.3967       19 0.5301
## HRSD        0.5186  0.5186     1  117.19  2.9886       20 0.0865
## LEAYRS      1.4252  1.4252     1  110.76  8.2132     kept 0.0050
##
## Least squares means:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
```

```
##
##  Differences of LSMEANS:
##      Estimate Standard Error DF t-value Lower CI Upper CI p-value
##
## Final model:
## lme4::lmer(formula = log(Time) ~ LEAYRS + (1 | Name) + (1 | type:List:Order),
##     data = DLM, REML = reml.lmerTest.private, contrasts = l.lmerTest.private.contrast,
##     devFunOnly = devFunOnly.lmerTest.private)
```

It is evident that interactions between all factors should be included in the model as well.

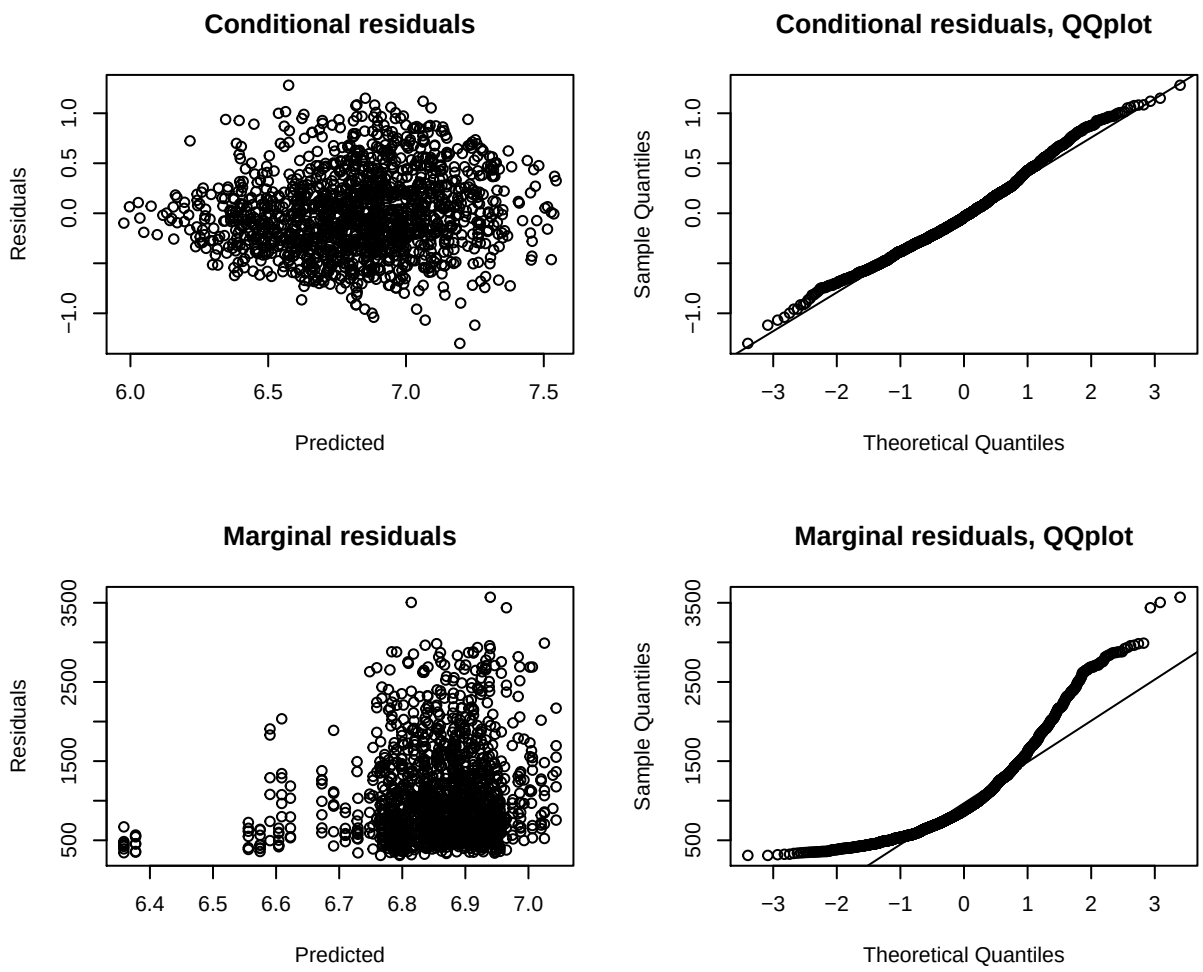
Finally, model comparison statistics using likelihood test comparisons:

```
#anova(lmm.0,lmm.1,lmm.2,lmm.3,lmm.4,lmm.5,lmm.6)
```

Why we get here that they are all insignificant instead??? Which test was used?

Diagnostic plots

```
lmm<-lmm.5
par(mfrow=c(3,2))
# plot(lm4,which=1:4)

plot(fitted(lmm, type = "response"), residuals(lmm, type = "response"),
     main = "Conditional residuals", xlab = "Predicted", ylab = "Residuals")

res <- residuals(lmm, type = "response")
qqnorm(res, main = "Conditional residuals, QQplot")
qqline(res)

lm.0 <- lm(log(Time) ~ (type + HRSD +  LEAYRS + RPV + AMGE + AMSP), data = DLM)
x <- model.matrix(lm.0)
pred <- x %*% fixef(lmm)
res <- DLM$Time - pred
plot(pred, res, main = "Marginal residuals", xlab = "Predicted", ylab = "Residuals")
qqnorm(res, main = "Marginal residuals, QQplot")
qqline(res)
```



The joint qqplot looks normal. The marginal looks less nice.

```
summary(lmm)
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite
##   approximations to degrees of freedom [lmerMod]
## Formula:
## log(Time) ~ (type + HRSD + LEAYRS + RPV + AMGE + AMSP) + (1 |
##     Name) + (1 | type:List:Order)
##    Data: DLM
##
## REML criterion at convergence: 1892.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1233 -0.6682 -0.0911  0.5855  3.0734
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Name            (Intercept) 0.0639   0.253
##  type:List:Order (Intercept) 0.0219   0.148
##  Residual                    0.1735   0.417
## Number of obs: 1476, groups:  Name, 118; type:List:Order, 32
```

```
##
## Fixed effects:
##               Estimate Std. Error        df t value Pr(>|t|)
## (Intercept)    6.86582    0.26176 112.00000   26.23   <2e-16 ***
## typeMISMATCH  -0.01860    0.05686  29.70000   -0.33   0.7459
## HRSD           0.01649    0.01392 114.80000    1.18   0.2386
## LEAYRS        -0.04409    0.01325 106.70000   -3.33   0.0012 **
## RPV           -0.00979    0.03345 110.10000   -0.29   0.7704
## AMGE          -0.00419    0.04195 107.10000   -0.10   0.9206
## AMSP           0.03898    0.04304 110.70000    0.91   0.3671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) tMISMA HRSD   LEAYRS RPV    AMGE
## typMISMATCH -0.110
## HRSD         0.061  0.001
## LEAYRS      -0.096  0.000 -0.205
## RPV         -0.581  0.000  0.047  0.030
## AMGE        -0.563  0.004  0.126  0.016  0.011
## AMSP        -0.606  0.000 -0.337 -0.095  0.325 -0.161
```

```
print(anova(lmm),signif.stars=TRUE)
```

```
## Analysis of Variance Table of type III  with  Satterthwaite
## approximation for degrees of freedom
##        Sum Sq Mean Sq NumDF DenDF F.value Pr(>F)
## type    0.019   0.019     1  29.7    0.11 0.7459
## HRSD    0.244   0.244     1 114.8    1.40 0.2386
## LEAYRS  1.921   1.921     1 106.7   11.07 0.0012 **
## RPV     0.015   0.015     1 110.1    0.09 0.7704
## AMGE    0.002   0.002     1 107.1    0.01 0.9206
## AMSP    0.142   0.142     1 110.7    0.82 0.3671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Bootstrapped confidence intervals

```
#op <- options(contrasts = c("contr.sum", "contr.poly"))
## fm01ML <- lmer(Yield ~ 1|Batch, Dyestuff, REML = FALSE)
## see ?"profile-methods"
mySumm <- function(.) { s <- sigma(.)
    c(beta =getME(., "beta"), sigma = s, sig01 = unname(s * getME(., "theta"))) }
(t0 <- mySumm(lmm)) # just three parameters
```

```
##      beta1      beta2      beta3      beta4      beta5      beta6
##  6.8658168 -0.0185968  0.0164943 -0.0440863 -0.0097866 -0.0041935
##      beta7      sigma     sig011     sig012
##  0.0389842  0.4165612  0.2528717  0.1478771
```

```
## alternatively:
mySumm2 <- function(.) {
   c(beta=fixef(.),sigma=sigma(.), sig01=sqrt(unlist(VarCorr(.))))
}

set.seed(101)
## 3.8s (on a 5600 MIPS 64bit fast(year 2009) desktop "AMD Phenom(tm) II X4 925"):
system.time( boo01 <- bootMer(lmm, mySumm, nsim = 100) )
```

```
##    user  system elapsed
##   3.343   0.045   3.491
```

```
## to "look" at it
require("boot") ## a recommended package, i.e. *must* be there
```

```
## Loading required package: boot
```

```
boo01
```

```
##
##
## Call:
## bootMer(x = lmm, FUN = mySumm, nsim = 100)
##
##
## Bootstrap Statistics :
##        original      bias    std. error
## t1*   6.8658168  0.01168254    0.270918
## t2*  -0.0185968 -0.00587139    0.054028
## t3*   0.0164943 -0.00126382    0.012478
## t4*  -0.0440863  0.00061497    0.014273
## t5*  -0.0097866 -0.00147984    0.033308
## t6*  -0.0041935 -0.00509108    0.046196
## t7*   0.0389842  0.00475658    0.035388
## t8*   0.4165612 -0.00240025    0.008428
## t9*   0.2528717 -0.00278447    0.022555
## t10*  0.1478771  0.00049576    0.022308
```

```
## note large estimated bias for sig01
## (~30% low, decreases _slightly_ for nsim = 1000)

## extract the bootstrapped values as a data frame ...
head(as.data.frame(boo01))
```

```
##    beta1     beta2     beta3     beta4     beta5      beta6
## 1 6.8977  0.038316 0.0199094 -0.046500  0.039625 -0.0315851
## 2 7.1404  0.042897 0.0099772 -0.064994 -0.038469 -0.0567358
## 3 6.8939  0.054903 0.0338754 -0.062929 -0.048686  0.0079933
## 4 7.2393 -0.054703 0.0286202 -0.048149 -0.053508 -0.0619453
## 5 6.8599  0.030648 0.0261376 -0.043729  0.022013 -0.0494519
## 6 6.4638  0.023468 0.0186225 -0.041355  0.025094  0.0263772
```

```
##      beta7   sigma  sig011  sig012
## 1 0.010765 0.39892 0.24442 0.14176
## 2 0.051691 0.42215 0.24766 0.11884
## 3 0.035298 0.42099 0.27182 0.12920
## 4 0.035571 0.40450 0.22969 0.11215
## 5 0.040262 0.41129 0.21814 0.14413
## 6 0.082190 0.42774 0.22477 0.12946
```

```
## ------ Bootstrap-based confidence intervals ------------

## warnings about "Some ... intervals may be unstable" go away
##   for larger bootstrap samples, e.g. nsim=500

## intercept
(bCI.1 <- boot.ci(boo01, index=1, type=c("norm", "basic", "perc")))# beta
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boo01, type = c("norm", "basic", "perc"),
##     index = 1)
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   ( 6.323,  7.385 )   ( 6.319,  7.431 )   ( 6.301,  7.412 )
## Calculations and Intervals on Original Scale
## Some basic intervals may be unstable
## Some percentile intervals may be unstable
```

```
## Residual standard deviation - original scale:
(bCI.2  <- boot.ci(boo01, index=2, type=c("norm", "basic", "perc")))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boo01, type = c("norm", "basic", "perc"),
##     index = 2)
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   (-0.1186,  0.0932 )   (-0.1116,  0.0982 )   (-0.1354,  0.0744 )
## Calculations and Intervals on Original Scale
## Some basic intervals may be unstable
## Some percentile intervals may be unstable
```

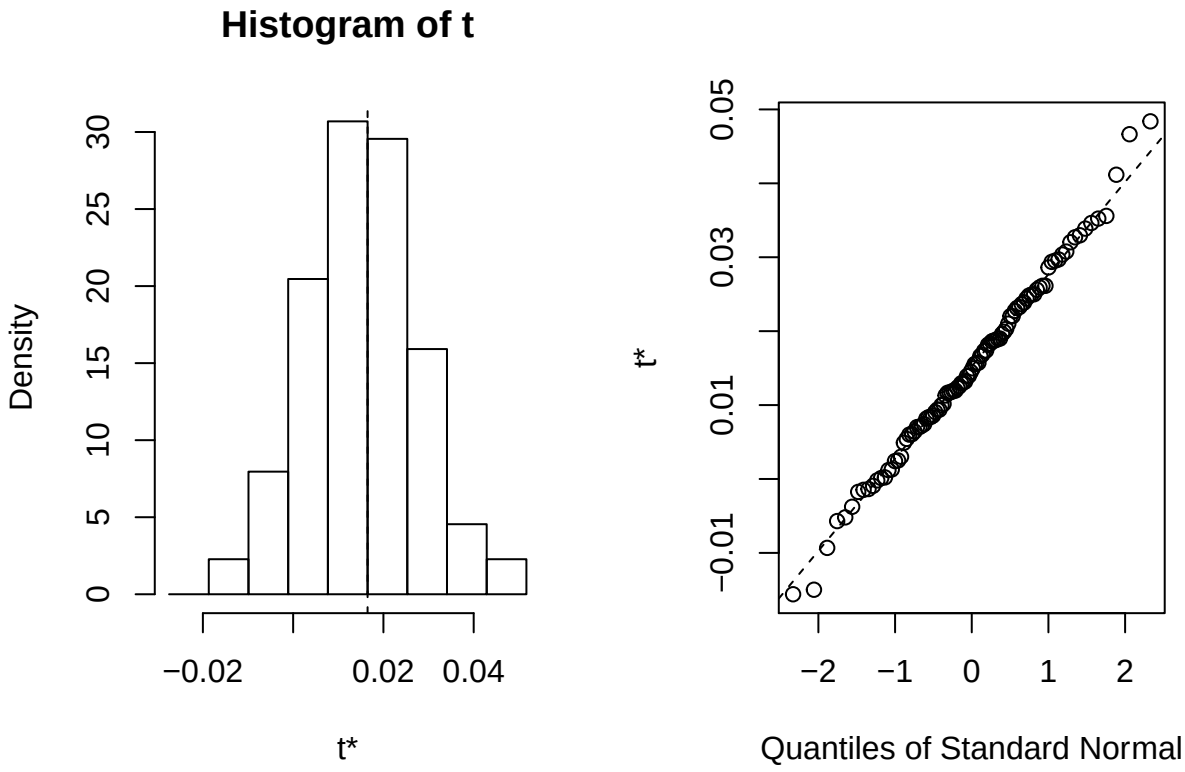
```
## Residual SD - transform to log scale:
#(bCI.2L <- boot.ci(boo01, index=2, type=c("norm", "basic", "perc"),
#                   h = log, hdot = function(.) 1/., hinv = exp))

## Among-batch variance:
(bCI.3 <- boot.ci(boo01, index=3, type=c("norm", "basic", "perc"))) # sig01
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boo01, type = c("norm", "basic", "perc"),
##     index = 3)
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   (-0.0067,  0.0422 )   (-0.0106,  0.0448 )   (-0.0118,  0.0435 )
## Calculations and Intervals on Original Scale
## Some basic intervals may be unstable
## Some percentile intervals may be unstable
```

```
## Extract all CIs (somewhat awkward)
bCI.tab <- function(b,ind=length(b$t0), type="perc", conf=0.95) {
    btab0 <- t(sapply(as.list(seq(ind)),
                      function(i)
                      boot.ci(b,index=i,conf=conf, type=type)$percent))
    btab <- btab0[,4:5]
    rownames(btab) <- names(b$t0)
    a <- (1 - conf)/2
    a <- c(a, 1 - a)
    pct <- stats:::format.perc(a, 3)
    colnames(btab) <- pct
    return(btab)
}
bCI.tab(boo01)
```

```
##            2.5 %    97.5 %
## beta1   6.300536  7.412378
## beta2  -0.135416  0.074415
## beta3  -0.011772  0.043539
## beta4  -0.069761 -0.017079
## beta5  -0.081277  0.045057
## beta6  -0.110438  0.075605
## beta7  -0.031018  0.118504
## sigma   0.394989  0.432395
## sig011  0.200435  0.293182
## sig012  0.111632  0.206172
```

```
## Graphical examination:
plot(boo01,index=3)
```


Histogram of t
Density
0
5
10
15
20
25
30
−0.02
0.02
0.04
t*
t*
0.05
0.03
0.01
−0.01
−2
−1
0
1
2
Quantiles of Standard Normal