

Bilingualism

Data analysis

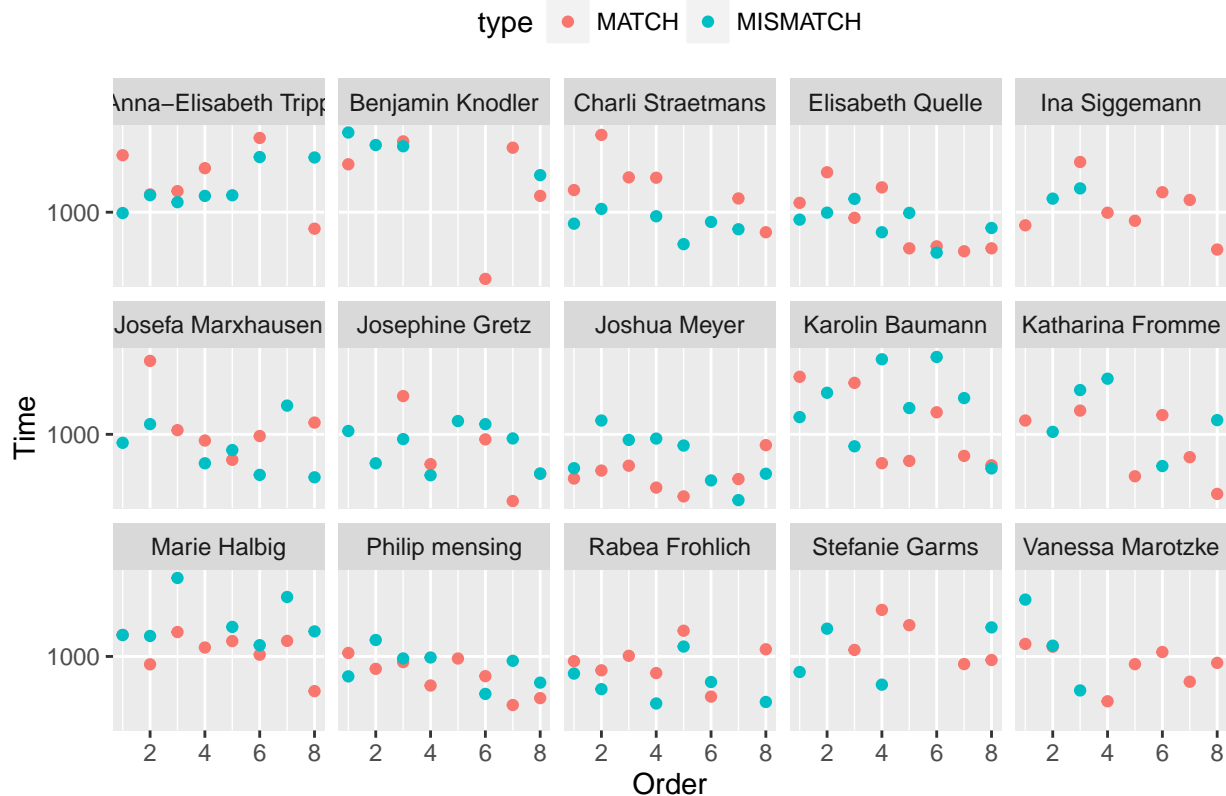
We load the data and remove the cases with NA values.

Let's have a glimpse at the data:

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

```
## Observations: 1,171
## Variables: 20
## $ Name      (fctr) Lisa Kettwig, Natalie Laibach, Tim Daldrup, Alb...
## $ CoR       (fctr) 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, ...
## $ Hand      (fctr) 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, ...
## $ EO        (fctr) 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, ...
## $ List      (fctr) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ CEF       (fctr) 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, ...
## $ SRRC      (dbl) 4.0, 4.0, 3.0, 3.8, 4.0, 4.0, 4.0, 4.0, 4.0, 4.0...
## $ PRE       (dbl) 0.83, 0.83, 0.83, 0.33, 0.78, 0.39, 1.00, 1.00, ...
## $ POST1     (dbl) 0.94, 1.00, 0.94, 0.94, 0.94, 1.00, 1.00, 1.00, ...
## $ POST2     (dbl) 0.88, 1.00, 0.88, 1.00, 0.88, 1.00, 0.83, 0.88, ...
## $ STAY      (fctr) 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, ...
## $ LEAYRS    (dbl) 0.50, 1.25, 1.00, 4.20, 5.00, 6.00, 1.50, 2.00, ...
## $ HRSD      (dbl) 0.375, 0.000, 0.350, 0.250, 0.000, 1.000, 0.375, ...
## $ RPV       (dbl) 5.0, 3.5, 5.0, 5.0, 5.0, 5.0, 4.0, 5.0, 3.5, 5.0...
## $ AMGE      (dbl) 2.5, 3.5, 4.0, 3.5, 4.5, 4.5, 4.5, 5.0, 5.0, 3.0...
## $ AMSP      (dbl) 3.6667, 3.3333, 3.5000, 4.1000, 3.3333, 4.0000, ...
## $ Order     (int) 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ Time      (int) 2095, 1842, 1187, 2808, 1075, 2210, 817, 1043, 1...
## $ ID        (int) 4, 6, 8, 9, 10, 12, 13, 14, 15, 16, 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 diagnostic 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), REML=FALSE, data = DLM)
# lrtest(lm.0, lmm.0)
AIC(lm.0, lmm.0)
```

```
##      df      AIC
## lm.0   2 1759.8
## lmm.0   4 1424.6
```

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)
```

```
## 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 1425 1445   -708    1417
## ..1     5 1423 1449   -707    1413    3.2      1      0.073 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## 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 1425 1445   -708    1417
## ..1     5 1425 1450   -707    1415    1.57      1      0.21
```

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

```
## 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 1425 1445   -708    1417
## ..1     5 1426 1452   -708    1416    0.22      1      0.64
```

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

```
## 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 1425 1445   -708    1417
## ..1     5 1426 1452   -708    1416    0.13      1      0.72
```

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

```
## 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 1425 1445   -708    1417
## ..1     5 1422 1447   -706    1412    4.79      1      0.029 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## 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 1425 1445   -708    1417
## ..1     5 1425 1450   -708    1415  1.54     1     0.21
```

```
lmm.5 <- lmer(log(Time) ~ (type + HRSD + LEAYRS + RPV + AMGE + AMSP) + (1 | Name) + (1 | type:List:Order), data = DLM)
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    0.516    0.516     1  30.6     3.24  0.082 .
## HRSD     0.099    0.099     1   93.2     0.62  0.432
## LEAYRS    0.149    0.149     1   93.1     0.94  0.336
## RPV       0.006    0.006     1   94.3     0.04  0.849
## AMGE      0.157    0.157     1   97.1     0.99  0.322
## AMSP      0.743    0.743     1  96.4     4.67  0.033 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

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

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

Finally, model comparison statistics using likelihood test comparisons:

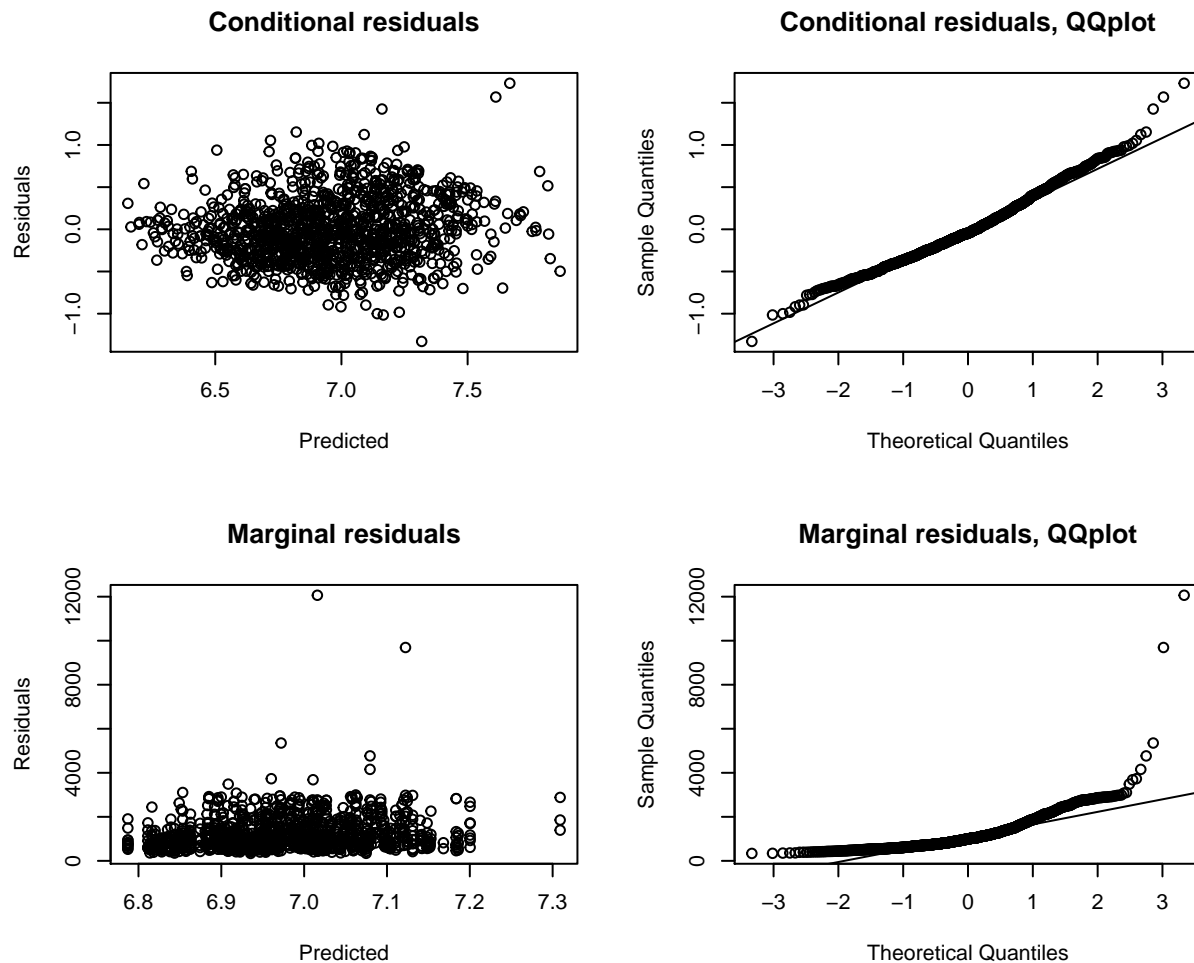
Diagnostic plots

```
lmm <- lmer(log(Time) ~ (type + HRSD + LEAYRS + AMSP) + (1 | Name) + (1 | type:List:Order), data = DLM)
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 + 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 + AMSP) + (1 | Name) + (1 |
## type:List:Order)
## Data: DLM
##
## REML criterion at convergence: 1432
##
## Scaled residuals:
```

```
##      Min      1Q Median      3Q      Max
## -3.332 -0.660 -0.114  0.579  4.341
##
## Random effects:
## Groups          Name          Variance Std.Dev.
## Name            (Intercept) 0.0739   0.272
## type:List:Order (Intercept) 0.0246   0.157
## Residual                0.1591   0.399
## Number of obs: 1171, groups: Name, 100; type:List:Order, 32
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   7.36754    0.20739 105.40000   35.53 <2e-16 ***
## typeMISMATCH   0.10870    0.06053  30.50000    1.80  0.082 .
## HRSD          -0.02520    0.03574  94.30000   -0.71  0.482
## LEAYRS         0.00876    0.00877  94.90000    1.00  0.320
## AMSP          -0.11146    0.05208  97.20000   -2.14  0.035 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) tMISMA HRSD   LEAYRS
## typMISMATCH -0.141
## HRSD         0.114  0.000
## LEAYRS       0.071 -0.004  0.063
## AMSP        -0.946 -0.001 -0.244 -0.254
```

```
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    0.513   0.513     1  30.5     3.23  0.082 .
## HRSD     0.079   0.079     1  94.3     0.50  0.482
## LEAYRS   0.159   0.159     1  94.9     1.00  0.320
## AMSP     0.729   0.729     1  97.2     4.58  0.035 *
## ---
## 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      sigma
## 7.3675413 0.1086982 -0.0252010 0.0087612 -0.1114650 0.3988773
```

```
##      sig011      sig012
## 0.2718926 0.1567349

## 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
##    2.697    0.027    2.798
```

```
## 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*  7.3675413  0.00794649  0.1792210
## t2*  0.1086982 -0.00634568  0.0536769
## t3* -0.0252010 -0.00107333  0.0327269
## t4*  0.0087612  0.00058002  0.0087451
## t5* -0.1114650 -0.00044604  0.0473585
## t6*  0.3988773 -0.00199437  0.0084120
## t7*  0.2718926 -0.00407914  0.0239525
## t8*  0.1567349 -0.00402359  0.0266982
```

```
## 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      sigma sig011
## 1 7.2917 0.013805 0.0233221 0.0071846 -0.09519 0.40257 0.26172
## 2 7.4127 0.099467 -0.0370188 0.0038531 -0.11539 0.39719 0.26288
## 3 7.5113 0.149329 -0.0710650 0.0067951 -0.14158 0.39878 0.25503
## 4 7.5112 0.146625 -0.0090036 0.0165607 -0.17772 0.40092 0.27571
## 5 7.3472 0.115968 -0.0125328 0.0028576 -0.12034 0.40998 0.26882
```

```

## 6 7.5067 0.108057 -0.0397049 0.0028788 -0.14622 0.38625 0.24133
##   sig012
## 1 0.13685
## 2 0.12175
## 3 0.19673
## 4 0.11721
## 5 0.12828
## 6 0.18064

## ----- 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%   ( 7.008, 7.711 )   ( 6.932, 7.760 )   ( 6.975, 7.803 )
## 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%   ( 0.0098, 0.2202 )   ( 0.0093, 0.2237 )   (-0.0063, 0.2081 )
## 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))

## Warning in h(t): NaNs produced

```



```
## Warning in h(t[fins]): NaNs produced

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 100 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boo01, type = c("norm", "basic", "perc"),
##       index = 2, h = log, hdot = function(.) 1/., hinv = exp)
##
## Intervals :
## Level      Normal      Basic      Percentile
## 95% (      NaN,      NaN ) ( 0.0565, 1.3806 ) ( 0.0086, 0.2093 )
## Calculations on Transformed Scale; Intervals on Original Scale
## Some basic intervals may be unstable
## Some percentile intervals may be unstable

## 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% (-0.0883, 0.0400 ) (-0.0901, 0.0382 ) (-0.0886, 0.0397 )
## 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    6.9752149  7.8029307
## beta2   -0.0062700  0.2080913
## beta3   -0.0886181  0.0396993
```

```
## beta4 -0.0096483 0.0272793
## beta5 -0.2031443 -0.0083827
## sigma 0.3791179 0.4129117
## sig011 0.2197555 0.3320774
## sig012 0.1044786 0.2018474
```

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

Histogram of t

