Die Übung am Tag der Arbeit (1.5.) entfällt. Ab 2.5. steht die Musterlösung im Netz. Wer Fragen zur Musterlösung hat kann eine E-mail an Fabian Scheipl schreiben oder einen Termin ausmachen.

Aufgabe 1:

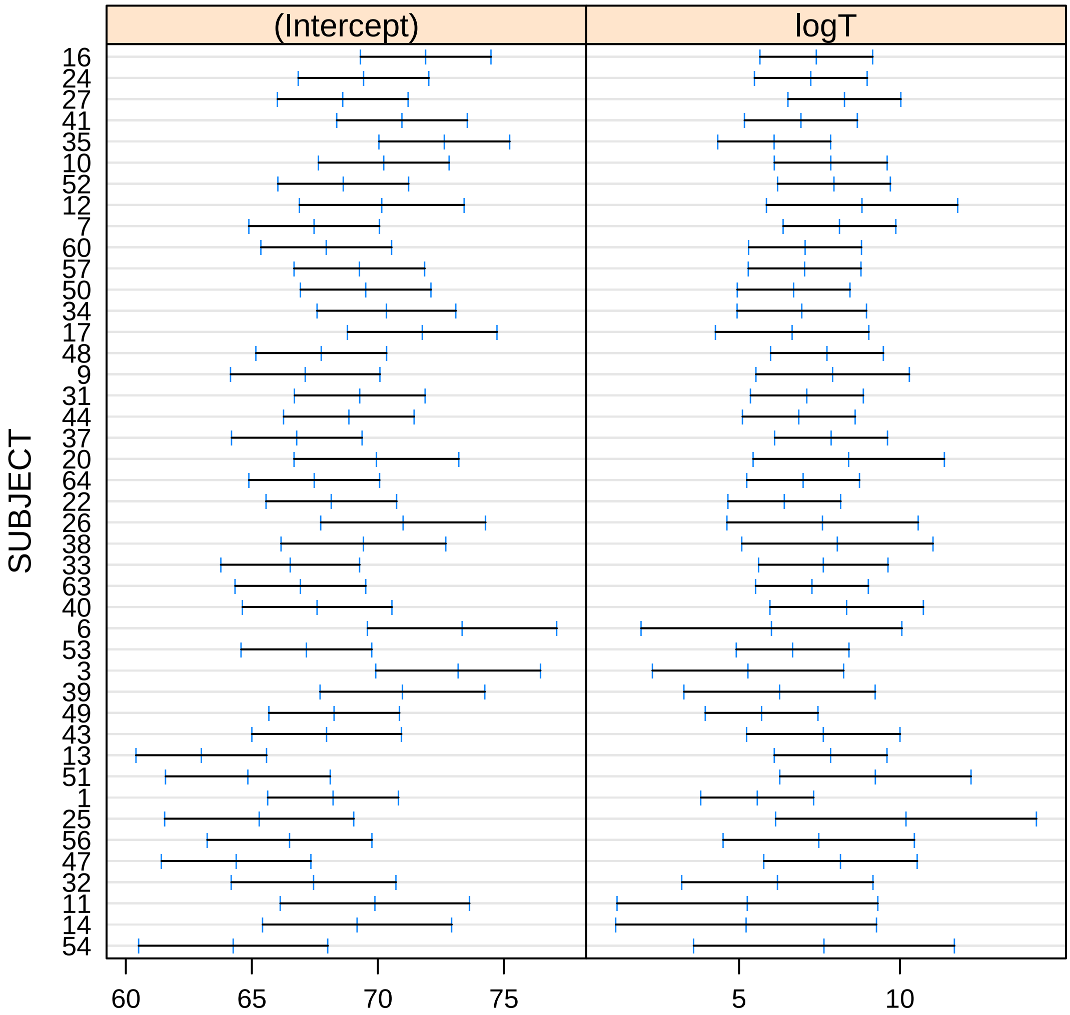
In dieser Aufgabe sollen Sie erste Erfahrungen mit dem Schätzen gemischter Modelle in R machen. Benutzen Sie den folgenden Code um den Ratten-Datensatz nach R einzulesen und für die Analyse vorzubereiten:

```
R>library("nlme")
R>url <- "http://www.statistik.lmu.de/institut/lehrstuhl/semwiso/longitudinal_ss08"
R>rats <- read.table(paste(url, "/download/rats.txt", sep = ""),
+     header = T, na.strings = ".")
R>rats$GROUP <- factor(rats$GROUP, labels = c("low", "high", "control"))
R>rats$SUBJECT <- factor(rats$SUBJECT)
R>rats$logT = log(1 + (rats$TIME - 45)/10)
R>rats <- rats[complete.cases(rats), ]
R>obs <- table(rats$SUBJECT)
R>zuvieleNAs <- levels(rats$SUBJECT)[obs < 3]
R>rats <- rats[!(rats$SUBJECT %in% zuvieleNAs), ]
```

(a) Legen Sie für `rats` ein `groupedData`-Objekt an. Berechnen Sie separate lineare Modelle mit linearem Trend in `logT` für die einzelnen Ratten. Visualisieren Sie die geschätzten subjektspezifischen Regressionskoeffizienten mit Konfidenzintervallen.

Lösung:

```
R>rats <- groupedData(RESPONSE ~ logT | SUBJECT, data = rats, labels = list(x = "log(Age)",
+     y = "Distance"), units = list(x = " ", y = "[Pixels]"))
R>lmlist.lin <- lmList(RESPONSE ~ logT, rats)
R>print(plot(intervals(lmlist.lin)))
```



⇒ teilweise große Unterschiede im Ausgangsniveau, sehr große Überlappung der Intervalle für Trend.

(b) Berechnen Sie mit der Funktion `lme` ein gemischtes lineares Modell mit linearem Trend in `logT` und subjektspezifischen zufälligen Intercepts.

  (i) Wie groß ist die geschätzte Korrelation zwischen 2 Messungen an der selben Ratte?

  (ii) Wie groß ist die Korrelation zwischen 2 Messungen zum gleichen Zeitpunkt bei verschiedenen Ratten?

Lösung:

```
R>lmm.lin1 <- lme(RESPONSE ~ logT, random = ~1 | SUBJECT, data = rats)
R>summary(lmm.lin1)

Linear mixed-effects model fit by REML
 Data: rats
       AIC      BIC    logLik
  901.1729 915.0954 -446.5864
```

```
Random effects:
 Formula: ~1 | SUBJECT
        (Intercept) Residual
StdDev:    2.006254  1.19497

Fixed effects: RESPONSE ~ logT
               Value Std.Error  DF   t-value p-value
(Intercept) 68.63524 0.3683028 198 186.35547       0
logT         7.20782 0.1549328 198  46.52223       0
 Correlation:
     (Intr)
logT -0.512

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.37783156 -0.69658660 -0.03501172  0.54495942  2.84167218

Number of Observations: 242
Number of Groups: 43
```

(i) Das verwendete Modell für den Responsevektor $\boldsymbol{y}_i$ der i-ten Ratte ist

$$\boldsymbol{y}_i = \beta_0 \mathbf{1}_{n_i} + \beta_1 \boldsymbol{x}_i + b_{0i}\mathbf{1}_{n_i} + \boldsymbol{\varepsilon}_i;\ i = 1, \dots, 43$$

$$\text{mit } \boldsymbol{b}_0 \sim N_{43}(\mathbf{0}, d^2 * \boldsymbol{I}_{43});\ \boldsymbol{\varepsilon} \sim N_{242}(\mathbf{0}, \sigma^2 \boldsymbol{I}_{242})$$

$$\begin{aligned}
\Rightarrow \operatorname{Cov} \boldsymbol{y}_i &= \operatorname{Cov}(b_{0i}\mathbf{1}_{n_i} + \boldsymbol{\varepsilon}_i) \\
&= \operatorname{Cov}(b_{0i}\mathbf{1}_{n_i}) + \operatorname{Cov}(\boldsymbol{\varepsilon}_i) \quad (\boldsymbol{b}, \boldsymbol{\varepsilon} \text{ unabhängig}) \\
&= \mathbf{1}_{n_i} d^2 \mathbf{1}'_{n_i} + \sigma^2 \boldsymbol{I}_{n_i} \\
&= \begin{pmatrix} d^2+\sigma^2 & d^2 & \dots & d^2 \\ d^2 & \ddots & & \vdots \\ \vdots & & \ddots & d^2 \\ d^2 & \dots & d^2 & d^2+\sigma^2 \end{pmatrix}
\end{aligned}$$

Für die Korrelation ergibt sich daraus:

$$\begin{aligned}
\operatorname{Corr}(y_{ij_1}, y_{ij_2}) &= \frac{\operatorname{Cov}(y_{ij_1}, y_{ij_2})}{\sqrt{\operatorname{Var}(y_{ij_1}) \operatorname{Var}(y_{ij_2})}} \\
&= \frac{d^2}{\sqrt{(d^2+\sigma^2)^2}}
\end{aligned}$$

In R:

```
R>(getVarCov(lmm.lin1))
```

```
Random effects variance covariance matrix
            (Intercept)
(Intercept)      4.0251
  Standard Deviations: 2.0063
```

```
R>(d.sq <- getVarCov(lmm.lin1)[1])
```

```
[1] 4.025056
```

```
R>(sigma.sq <- (lmm.lin1$sigma)^2)
```

```
[1] 1.427953
R>(cor.withinGroup <- d.sq/(d.sq + sigma.sq))
[1] 0.7381348
```

⇒ Die Korrelation beträgt ca. 0.74.

(ii) Laut Modellannahme sind Beobachtungen an verschiedenen Ratten voneinander unabhängig, also auch unkorreliert.

(c) Berechnen Sie mit der Funktion `lme` ein gemischtes lineares Modell mit linearem Trend in `logT` und subjektspezifischen zufälligen Intercepts und Trends.

(i) Bestimmen Sie die Kovarianzmatrix D (s. Folie 3.26) der zufälligen Effekte. Wie groß ist die Korrelation zwischen den zufälligen Intercepts und Slopes?

(ii) Wie kann man also das Modell vereinfachen? Schätzen Sie das entsprechende Modell mit vereinfachter Kovarianzstruktur der zufälligen Effekte.
Hinweis: Die Beispiele in `?anova.lme` könnten weiterhelfen.

(iii) Vergleichen Sie die Parameterschätzungen und die gefitteten Werte der subjektspezifischen linearen Modelle und des zweiten gemischten Modells. *Hinweis:* Benutzen Sie `compareFits` und `comparePred`.

Lösung:

```
R>lmm.lin2 <- lme(RESPONSE ~ logT, random = ~1 + logT | SUBJECT,
+     data = rats)
R>summary(lmm.lin2)

Linear mixed-effects model fit by REML
 Data: rats
       AIC      BIC    logLik
  905.1729 926.0567 -446.5864

Random effects:
 Formula: ~1 + logT | SUBJECT
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev       Corr
(Intercept) 2.0062540747 (Intr)
logT        0.0002241546 0
Residual    1.1949699031

Fixed effects: RESPONSE ~ logT
               Value Std.Error  DF   t-value p-value
(Intercept) 68.63524 0.3683028 198 186.35548       0
logT         7.20782 0.1549328 198  46.52223       0
 Correlation:
     (Intr)
logT -0.512

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.37783157 -0.69658657 -0.03501174  0.54495942  2.84167221
```

```
Number of Observations: 242
Number of Groups: 43
```

(i) *R>(D <- getVarCov(lmm.lin2))*

```
Random effects variance covariance matrix
             (Intercept)       logT
(Intercept)  4.0251e+00 8.7438e-08
logT         8.7438e-08 5.0245e-08
  Standard Deviations: 2.0063 0.00022415
```

R>cov2cor(D)

```
Random effects variance covariance matrix
             (Intercept)       logT
(Intercept)  1.00000000 0.00019443
logT         0.00019443 1.00000000
  Standard Deviations: 1 1
```

$\Rightarrow$ Die Korrelation zwischen b_{0i} und b_{1i} ist ca. 0.0002.

(ii) Das Modell kann vereinfacht werden indem man unkorrelierte zufällige Effekte annimmt, d.h. D wird als Diagonalmatrix geschätzt. Äquivalente Aufrufe (äquivalent weil `rats` ein `groupedData`-Objekt ist, sonst würden nur die letzten beiden, kompliziertesten `lme`-Aufruf korrekt sein.):

```
R>lmm.lin3 <- lme(RESPONSE ~ logT, random = pdDiag(~logT), data = rats)
R>lmm.lin3 <- lme(RESPONSE ~ logT, random = pdDiag(~1 + logT),
+     data = rats)
R>lmm.lin3 <- lme(RESPONSE ~ logT, random = list(SUBJECT = pdDiag(~1 +
+     logT)), data = rats)
R>lmm.lin3 <- lme(RESPONSE ~ logT, random = list(SUBJECT = pdDiag(~logT)),
+     data = rats)
R>(summary(lmm.lin3))
```

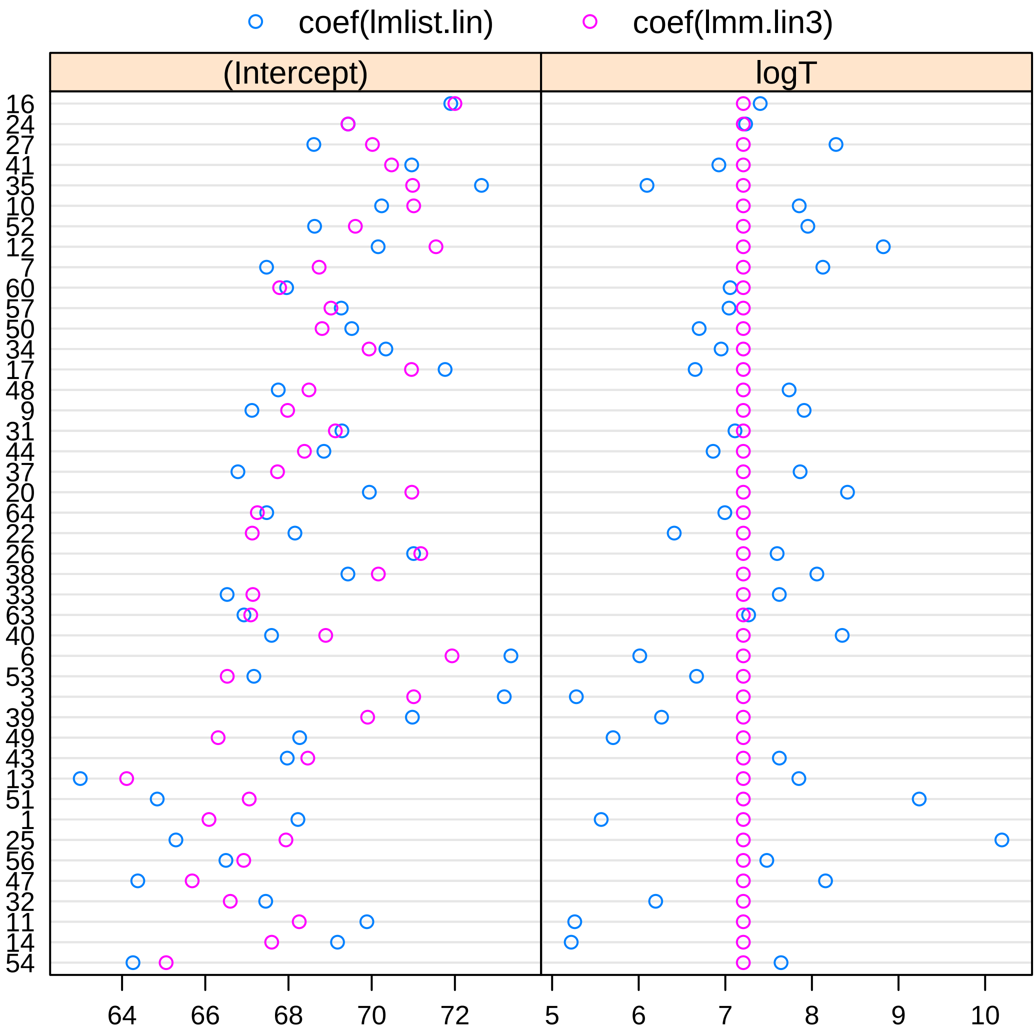
```
Linear mixed-effects model fit by REML
 Data: rats
       AIC     BIC    logLik
  903.1729 920.576 -446.5864

Random effects:
 Formula: ~logT | SUBJECT
 Structure: Diagonal
        (Intercept)         logT Residual
StdDev:    2.006254 0.0001645551  1.19497

Fixed effects: RESPONSE ~ logT
               Value Std.Error  DF   t-value p-value
(Intercept) 68.63524 0.3683028 198 186.35547       0
logT         7.20782 0.1549328 198  46.52223       0
 Correlation: 
     (Intr)
logT -0.512

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max 
-2.37783156 -0.69658659 -0.03501173  0.54495942  2.84167219 
```

```
Number of Observations: 242
Number of Groups: 43
```

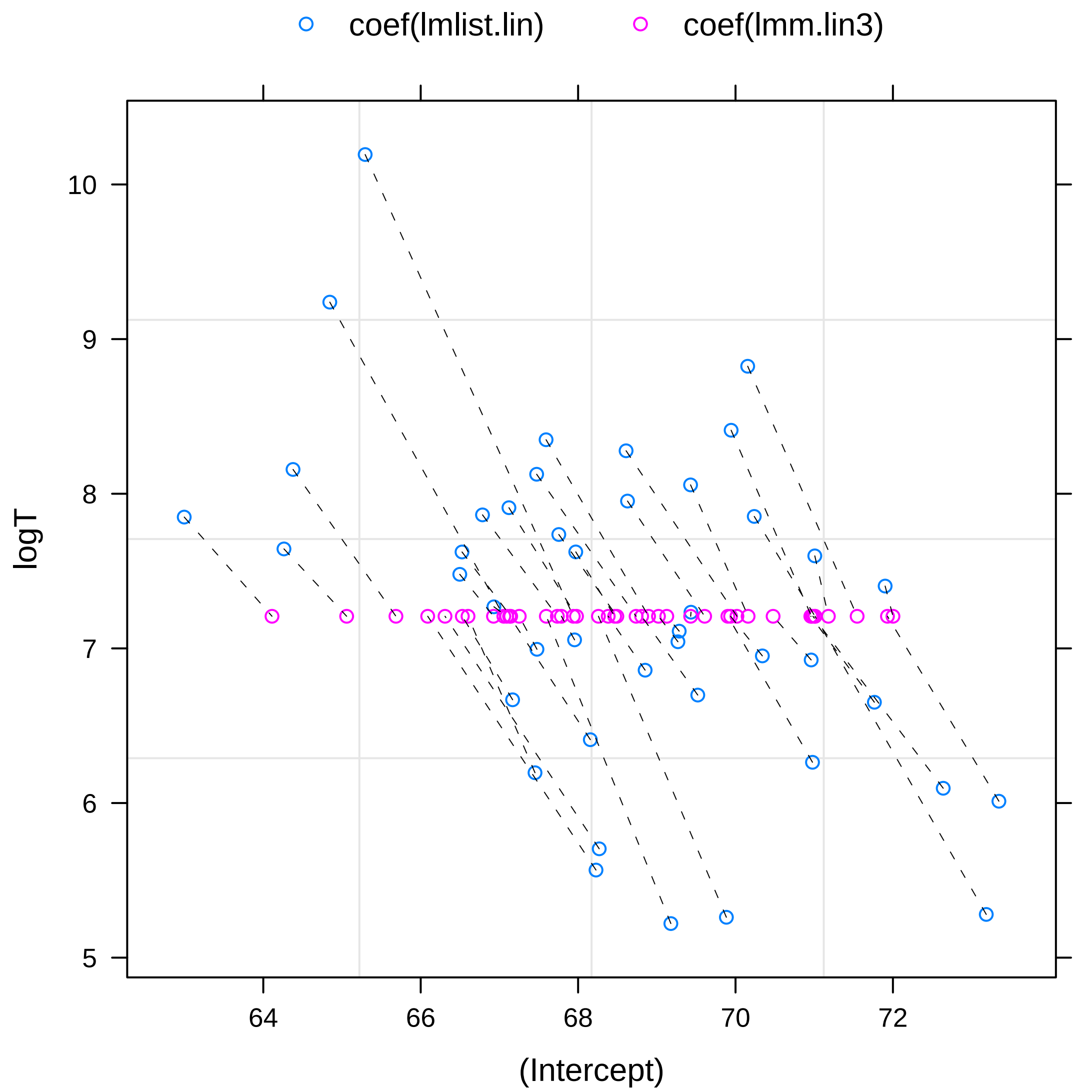
R>(getVarCov(lmm.lin3))

```
Random effects variance covariance matrix
             (Intercept)       logT
(Intercept)       4.0251 0.0000e+00
logT              0.0000 2.7078e-08
  Standard Deviations: 2.0063 0.00016456
```

(iii) *R>print(plot(compareFits(coef(lmlist.lin), coef(lmm.lin3))))*

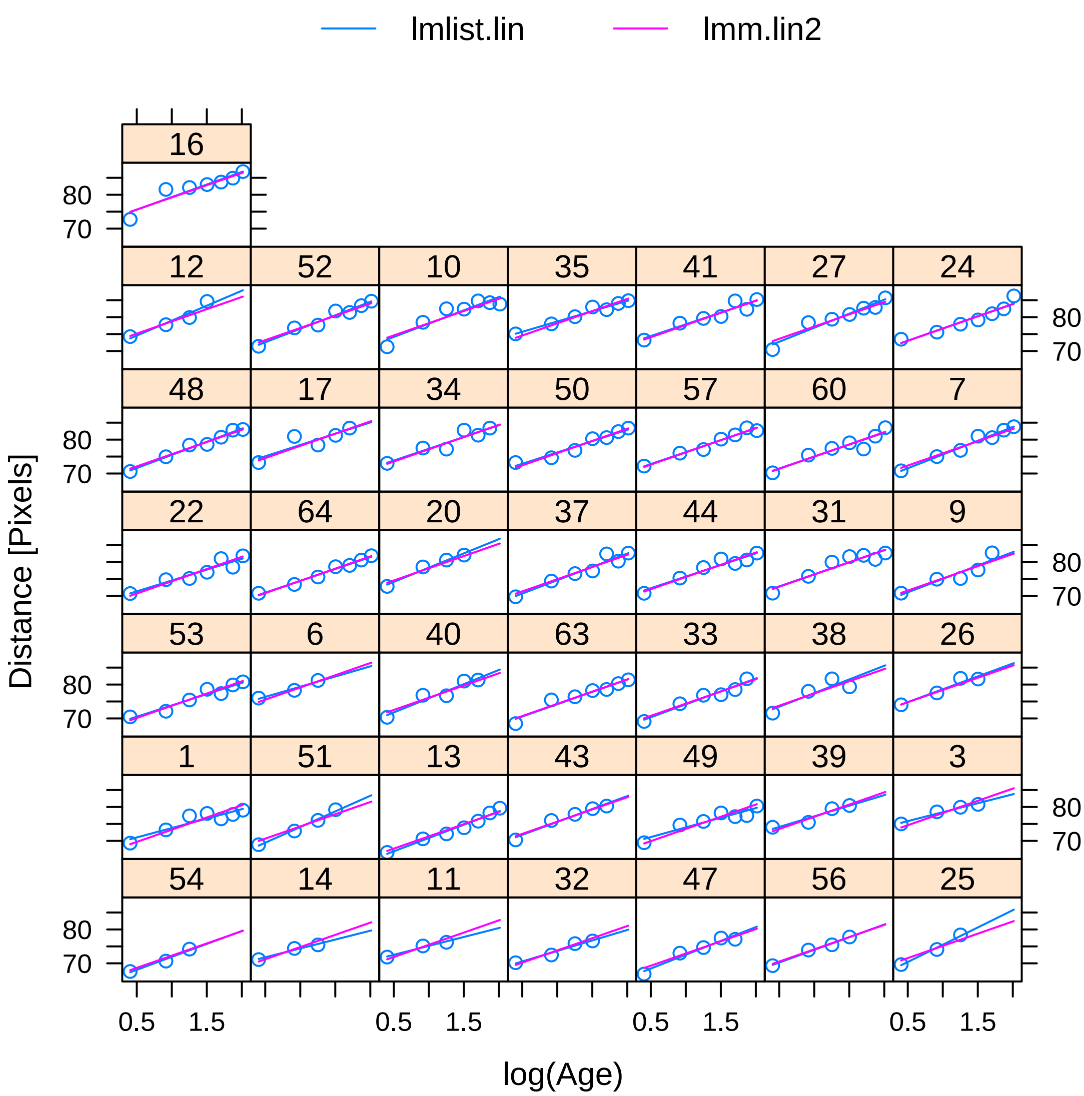


R>print(pairs(compareFits(coef(lmlist.lin), coef(lmm.lin3))))



Die subjektspezifischen Unterschiede im Trend sind mit so großer Unsicherheit behaftet (s. Teilaufgabe a)) dass sie durch das gemischte Modell (fast völlig) entfernt werden. Um trotzdem eine gute Anpassung zu erreichen müssen die subjektspezifischen Intercepts deshalb für Ratten mit überdurchschnittlich starkem Trend entsprechend größer werden (unpenalisierte Intercept-Schätzumg aus `lmList` kleiner als penalisierte Schätzung) und für Ratten mit unterdurchschnittlich starkem Trend entsprechend kleiner werden (unpenalisierte Intercept-Schätzumg aus `lmList` größer als penalisierte Schätzung), genau das sieht man sehr gut in der 2. Grafik.
Üblicherweise beobachtet man dass subjektspezifischen Unterschiede im gemischten Modell durch die Normalverteilungsannahme in Richtung des globalen Mittelwerts geschrumpft werden ($\Rightarrow$ Shrinkage-Effekt, Stabilisierung der Parameterschätzungen).

```
R>print(plot(comparePred(lmlist.lin, lmm.lin2)))
```



Die Unterschiede in den subjektspezifischen Regressionsgeraden sind teilweise recht deutlich. Da der geschätzte Trend für die Ratten im gemischten Modell für alle Tiere quasi gleich ist und deswegen die subjektspezifischen Intercepts für Ratten mit überdurchschnittlich starkem beobachteten Trend im gemischten Modell größer geschätzt werden (s.o.), sieht man hier dass die Regressionsgeraden für Tiere mit überdurchschnittlich starkem Trend durch das gemischte Modell im Ausgangsniveau nach oben verschoben werden und flacher werden, während sie für Tiere mit unterdurchschnittlich starkem Trend durch das gemischte Modell im Ausgangsniveau nach unten verschoben werden und steiler werden.

(d) Wiederholen Sie die Analyse aus b) für Modelle mit quadratischem zeitlichen Trend. Benutzen Sie das `lme4` Paket.

Lösung:

```
R>lmlist.qu <- lmList(RESPONSE ~ poly(logT, 2), rats)
R>try(lmm.qu1 <- lme(RESPONSE ~ poly(logT, 2), random = ~poly(logT,
+     2), data = rats))

Error in lme.formula(RESPONSE ~ poly(logT, 2), random = ~poly(logT, 2),:
```

```
  nlminb problem, convergence error code = 1
  message = iteration limit reached without convergence (9)

R>detach(package:nlme)
R>library(lme4)
R>lmm.qu1 <- lmer(RESPONSE ~ logT + I(logT^2) + (logT + I(logT^2) |
+     SUBJECT), data = rats)
R>summary(lmm.qu1)

Linear mixed-effects model fit by REML
Formula: RESPONSE ~ logT + I(logT^2) + (logT + I(logT^2) | SUBJECT)
   Data: rats
 AIC   BIC logLik MLdeviance REMLdeviance
 904 935.4   -443        883          886
Random effects:
 Groups   Name        Variance   Std.Dev.   Corr
 SUBJECT  (Intercept) 4.0202e+00 2.0050e+00
          logT        2.8100e-09 5.3010e-05 0.000
          I(logT^2)   6.9074e-10 2.6282e-05 0.000 0.868
 Residual             1.3815e+00 1.1754e+00
number of obs: 242, groups: SUBJECT, 43

Fixed effects:
            Estimate Std. Error t value
(Intercept)  67.7137     0.4930  137.36
logT          9.1905     0.7261   12.66
I(logT^2)    -0.8536     0.3057   -2.79

Correlation of Fixed Effects:
          (Intr) logT
logT      -0.733
I(logT^2)  0.669 -0.978

R>lmm.qu2 <- lmer(RESPONSE ~ logT + I(logT^2) + (1 | SUBJECT) +
+     (-1 + logT + I(logT^2) | SUBJECT), data = rats)
R>summary(lmm.qu2)

Linear mixed-effects model fit by REML
Formula: RESPONSE ~ logT + I(logT^2) + (1 | SUBJECT) + (-1 + logT + I(logT^2) | SUBJECT)
   Data: rats
 AIC   BIC logLik MLdeviance REMLdeviance
 900 924.4   -443        883          886
Random effects:
 Groups   Name        Variance   Std.Dev.   Corr
 SUBJECT  (Intercept) 4.0110e+00 2.0028e+00
 SUBJECT  logT        1.1593e-09 3.4049e-05
          I(logT^2)   6.9105e-10 2.6288e-05 -0.636
 Residual             1.3821e+00 1.1756e+00
number of obs: 242, groups: SUBJECT, 43; SUBJECT, 43

Fixed effects:
            Estimate Std. Error t value
(Intercept)  67.7137     0.4928  137.40
logT          9.1906     0.7263   12.65
I(logT^2)    -0.8537     0.3058   -2.79
```

```
Correlation of Fixed Effects:
          (Intr) logT
logT      -0.734
I(logT^2)  0.670 -0.978
```

```
R>(D1 <- VarCorr(lmm.qu1))

$SUBJECT
3 x 3 Matrix of class "dpoMatrix"
              (Intercept)          logT     I(logT^2)
(Intercept)  4.020195e+00 -4.769948e-09 -2.371984e-09
logT        -4.769948e-09  2.810015e-09  1.209903e-09
I(logT^2)   -2.371984e-09  1.209903e-09  6.907374e-10

attr(,"sc")
   scale
1.175362
```

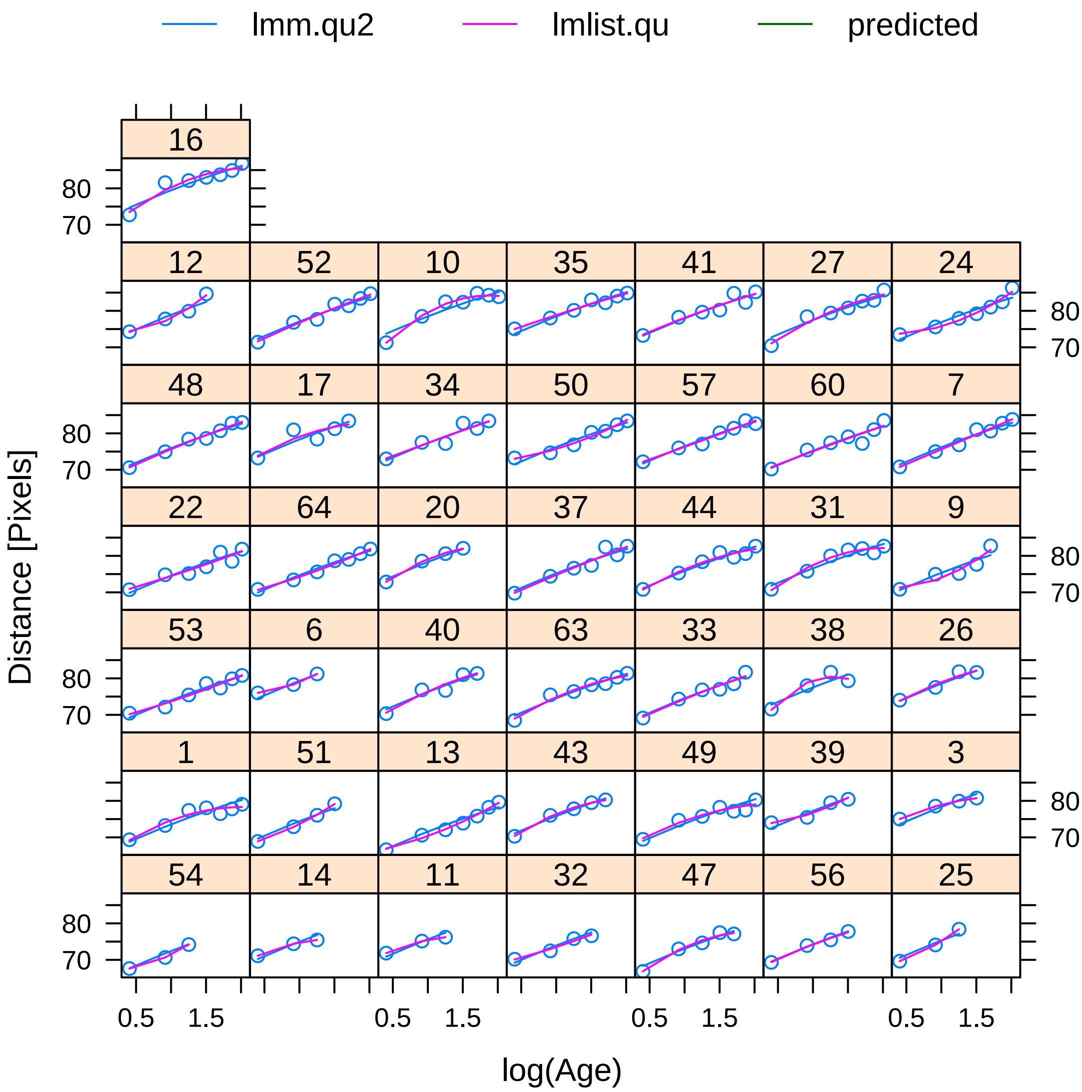
```
R>cov2cor(D1[[1]])

3 x 3 Matrix of class "dpoMatrix"
              (Intercept)          logT     I(logT^2)
(Intercept)  1.000000e+00 -4.487823e-05 -4.501235e-05
logT        -4.487823e-05  1.000000e+00  8.684396e-01
I(logT^2)   -4.501235e-05  8.684396e-01  1.000000e+00
```

```
R>(D2 <- VarCorr(lmm.qu2))

$SUBJECT
1 x 1 Matrix of class "dpoMatrix"
            (Intercept)
(Intercept)    4.011037

$SUBJECT
2 x 2 Matrix of class "dpoMatrix"
                   logT     I(logT^2)
logT       1.159320e-09 -5.688573e-10
I(logT^2) -5.688573e-10  6.910476e-10

attr(,"sc")
   scale
1.175625
```

```
R>cov2cor(D2[[2]])

2 x 2 Matrix of class "dpoMatrix"
                logT  I(logT^2)
logT       1.0000000 -0.6355469
I(logT^2) -0.6355469  1.0000000
```

Mit `nlme::lme()` kann man das quadratische Modell nicht einmal dann schätzen wenn man die numerisch günstigere Formulierung mit `poly` für den quadratischen Trend benutzt um eine orthogonale Designmatrix für den Zeittrend zu erzeugen. Die `lme4` Bibliothek ist stabiler/schneller und hat eine (meiner Meinung nach) etwas intuitivere Syntax für die zufälligen Effekte, verfügt aber (noch?) nicht über ausgefeilte Diagnoseplots wie `nlme`. Da `nlme` und `lme4` teilweise gleichlautende Funktionen benutzen kommt es zu Konflikten wenn beide Pakete gleichzeitig geladen sind, deswegen sollte man tunlichst `detach` benutzen (s.o.) um das Paket das gerade nicht benötigt wird zu entfernen.

```
R>library(nlme)
R>lmerlmList.comparePred <- function(lmm, lmlist, primary, level = 0,
+     length.out = 51) {
+     args <- list(object = lmlist, primary = primary, level = level,
+         length.out = length.out)
+     val1 <- do.call("augPred", args)
+     val1 <- val1[val1$.type == "original", ]
+     temp1 <- cbind(val1[, 1:2], fitted(lmlist), .type = rep(deparse(substitute(lmlist)),
+         nrow(val1)))
+     temp2 <- cbind(val1[, 1:2], fitted(lmm), .type = rep(deparse(substitute(lmm)),
+         nrow(val1)))
+     names(temp1) <- names(temp2) <- names(val1)
+     val <- rbind(temp2, temp1, val1)
+     class(val) <- c("comparePred", "augPred", class(val))
+     attr(val, "labels") <- attr(val1, "labels")
+     attr(val, "units") <- attr(val1, "units")
+     attr(val, "formula") <- attr(val1, "formula")
+     val
+ }
R>print(plot(lmerlmList.comparePred(lmm.qu2, lmlist.qu, primary = "logT")))
```


lmm.qu2
lmlist.qu
predicted
16
12
52
10
35
41
27
24
48
17
34
50
57
60
7
22
64
20
37
44
31
9
53
6
40
63
33
38
26
1
51
13
43
49
39
3
54
14
11
32
47
56
25
80
70
0.5
1.5
Distance [Pixels]
log(Age)