

Aufgabe 2

Betrachten Sie den Datensatz `sprint` zu den 100m-Läufen, den Sie aus der Vorlesung kennen.

```
# Simulation der sprint Daten #
# ===== #

set.seed(123)

n1 = 10
n2 = n1
n = n1 + n2
M = 3
b = rep(rnorm(n1 + n2, 0, 0.5), each = M)
sex = c(rep(0, n1), rep(1, n2))
beta = c(rep(10.6, n1 * M), rep(11.9, n2 * M))
eps = rnorm(n, 0, 0.2)
times = beta + b + eps
athlete = rep(1:n, each = M)

sprint <- data.frame(athlete, sex = rep(sex, each = M), times)
```

- Legen Sie ein `groupedData`-Object für die Daten an (R-Paket `nlme`), das die Gruppierungsstruktur widerspiegelt. Erstellen Sie eine Graphik für die Daten.
- Fitten Sie das Modell mit einem festen Effekt für Geschlecht und zufälligen Effekten für die Athleten (Modell wie in der Vorlesung besprochen) einmal über ML-Schätzung und einmal über REML-Schätzung. Was ist der Unterschied zwischen REML- und ML-Schätzung? Vergleichen Sie die Ergebnisse. Lassen Sie sich jeweils die Schätzer für die festen und die zufälligen Effekte ausgeben.
Hinweis: Funktion `lme()` aus dem R-Paket `nlme` fittet lineare gemischte Modelle.
- Berechnen Sie nun die Schätzer der festen Effekte in R “zu Fuß”, d.h. nutzen Sie die Formel für den verallgemeinerten KQ-Schätzer und berechnen Sie die Schätzer aus den entsprechenden Matrizen und Vektoren. Zur Bestimmung der Kovarianzmatrix \mathbf{V} der Zielgröße können Sie die Ergebnisse aus dem Modellfit in (b), sowie die Formel aus dem Skript verwenden verwenden.
Hinweis: Die Funktion `getVarCov()` extrahiert den Schätzer für die Varianz-Kovarianzmatrix der zufälligen Effekte aus einem `lme`-Objekt.
- Erstellen Sie eine Graphik, welche die Zeiten für alle Athleten darstellt. Zeichnen Sie zusätzlich die Prognosen aus den festen Effekten sowie die Prognosen aus den festen und zufälligen Effekten ein.
- Generieren Sie eine zusätzliche Variable `nation`, die 4 Ausprägungen hat und die Nationalität der Läufer darstellt (bereits im Code enthalten). Schätzen Sie ein adäquates Modell für die Zeiten der Läufer, das auch die Nationalität berücksichtigt.

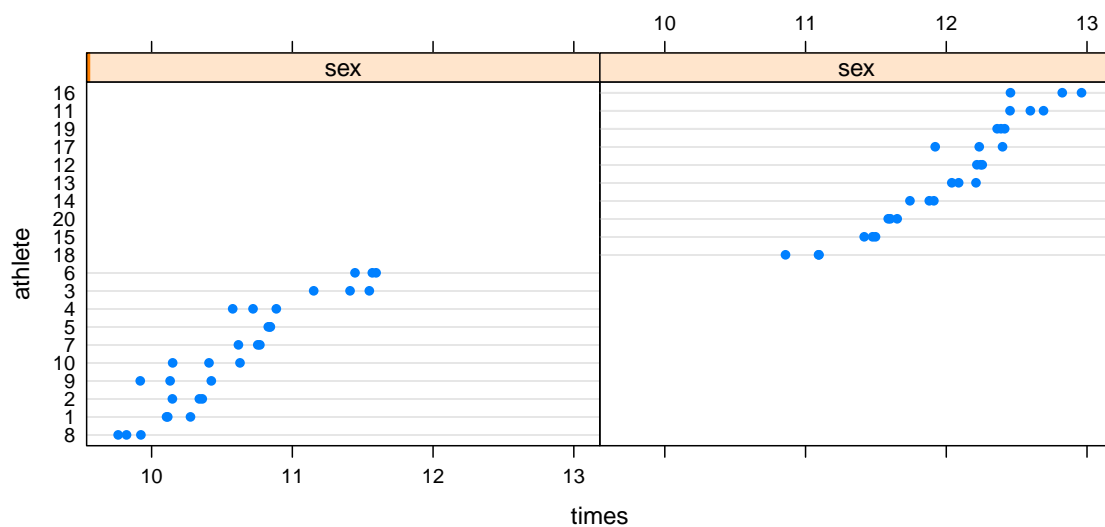
Lösung 2

- `groupedData`-Objekte:
 - `outer`: Faktoren, wie zB Geschlecht oder Behandlungsgruppe, die invariant sind bezüglich der subjektspezifischen Beobachtungen bei mehreren Faktoren, zum Beispiel: `outer=sex*nation` oder `outer=nation*sex`

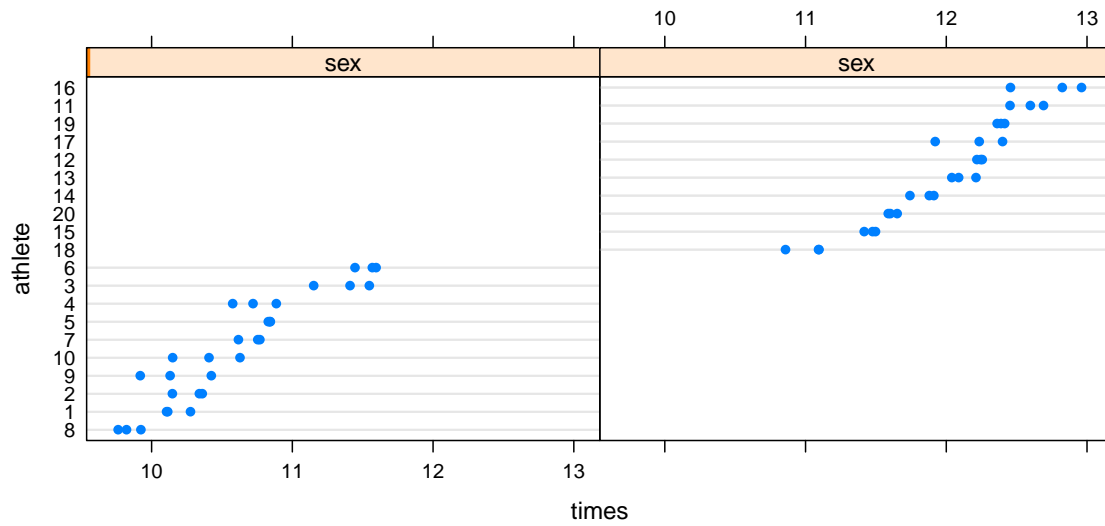
- **inner**: Faktoren, die sich innerhalb der Gruppierungsstruktur ändern und als feste Effekte modelliert werden sollen; hier zum Beispiel Art der Diät, die der Sportler am Tag des Laufes bekommen hat wenn sich die Diät über die Tage und die Sportler ändert.
- wenn man Faktoren, die sich innerhalb der Gruppierungsvariable ändern als zufällige Effekte modellieren möchte, sollte man diese als genestete Gruppierungsstruktur angeben; hier zum Beispiel Info auf welcher Bahn der Lauf stattfindet, wenn es verschiedenen Bahnen gibt, die von den Sportlern verwendet werden.

```
sprint0 <- groupedData(times ~ 1|athlete, data = sprint, outer = ~ sex)

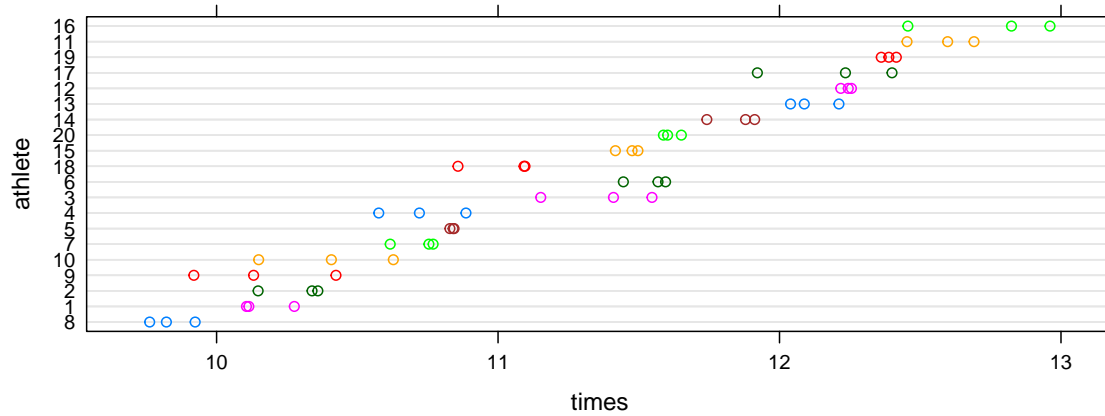
## different kinds of groupedData-plots
## ?plot.nmGroupedData
plot(sprint0, outer = ~ sex)
```



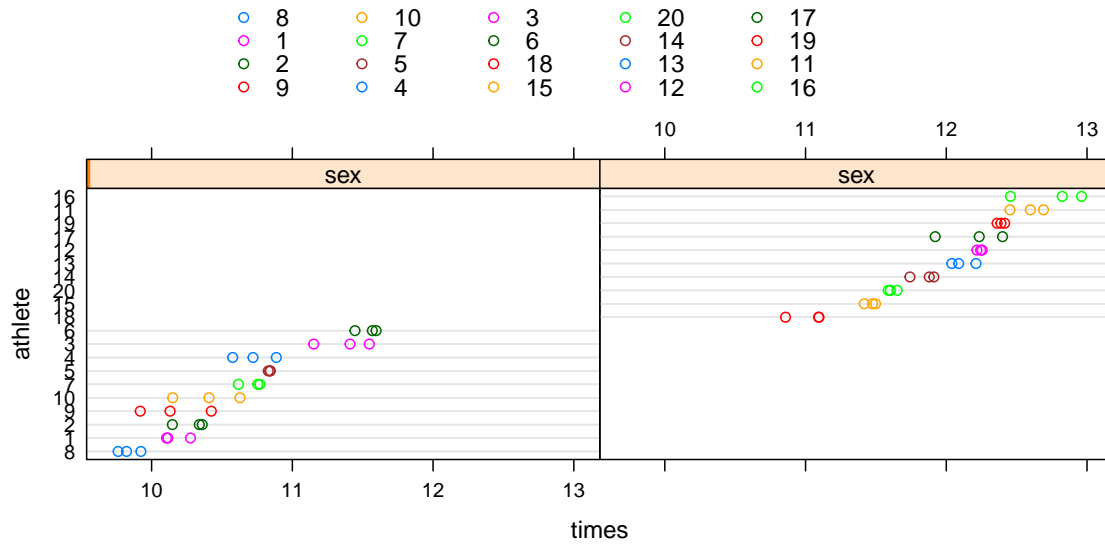
```
# use the specification of outer in the groupedData-object
plot(sprint0, outer = TRUE)
```



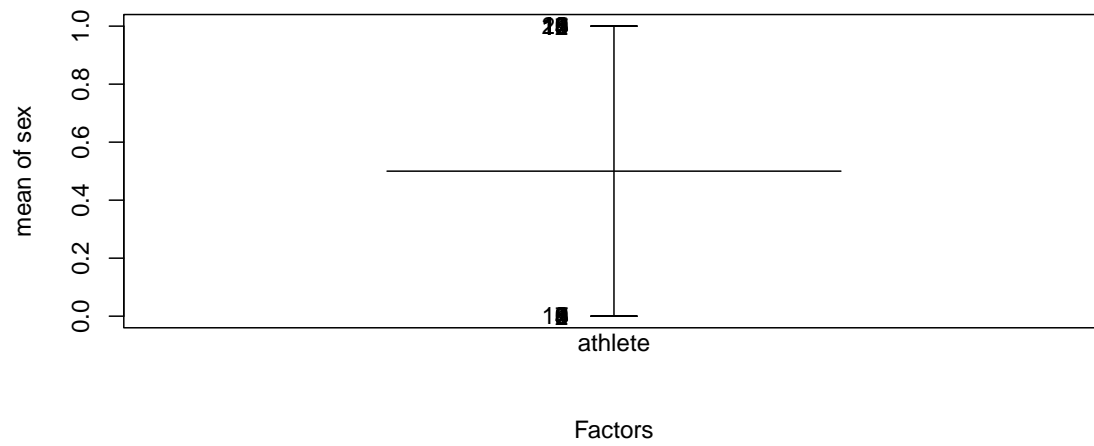
```
plot(sprint0, inner = ~ athlete)
```

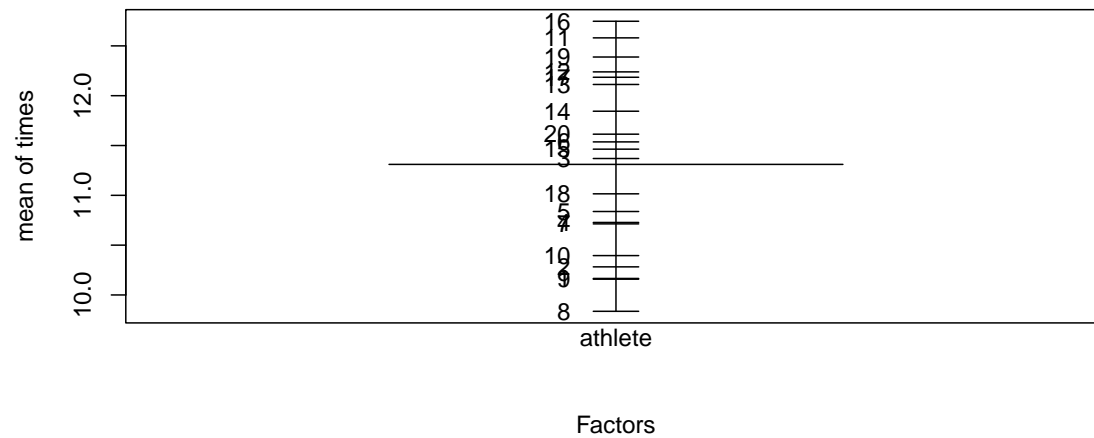


```
plot(sprint0, inner = ~ athlete, outer = ~ sex)
```



```
plot.design(sprint0)
```





```
(b) # fit the model with ML
sprintML <- lme(times ~ sex, random = ~ 1 | athlete,
               data = sprint, method = "ML")
# summary(sprintML)

# fit the model with REML
sprintREML <- lme(times ~ sex, random = ~ 1 | athlete,
                  data = sprint, method = "REML")
# summary(sprintREML)

# compare variance components
getVarCov(sprintML)

Random effects variance covariance matrix
      (Intercept)
(Intercept)      0.2553
Standard Deviations: 0.50528

getVarCov(sprintREML)

Random effects variance covariance matrix
      (Intercept)
(Intercept)      0.28443
Standard Deviations: 0.53332

# compare fixed effects
fixef(sprintML)
```

```

(Intercept)      sex
 10.602175     1.416772

fixef(sprintREML)

(Intercept)      sex
 10.602175     1.416772

# not equal in general!

# compare random effects
head(
  cbind("ML" = ranef(sprintML), "REML" = ranef(sprintREML))
)

      (Intercept) (Intercept)
1 -0.4251768    -0.4263124
2 -0.3116480    -0.3124803
3  0.7474187     0.7494149
4  0.1221639     0.1224902
5  0.2293335     0.2299460
6  0.9097690     0.9121989

```

```

(c) ##### compute fixed effects by hand #####

sprint$sex <- as.factor(sprint$sex)

# design matrices and response
X <- cbind(rep(1, 60), as.numeric(levels(sprint$sex)[sprint$sex]) )
Z <- kronecker(diag(20), rep(1,3))
y <- sprint$times

# Covariance matrices
R <- diag(rep(sprintML$sigma^2, 60)) # covariance of residuals
G <- diag(rep(getVarCov(sprintML), 20)) # covariance of random effects
V <- Z %*% G %*% t(Z) + R # covariance of response

# estimate beta
hatBeta <- solve(t(X) %*% solve(V) %*% X) %*% t(X) %*% solve(V) %*% y

```

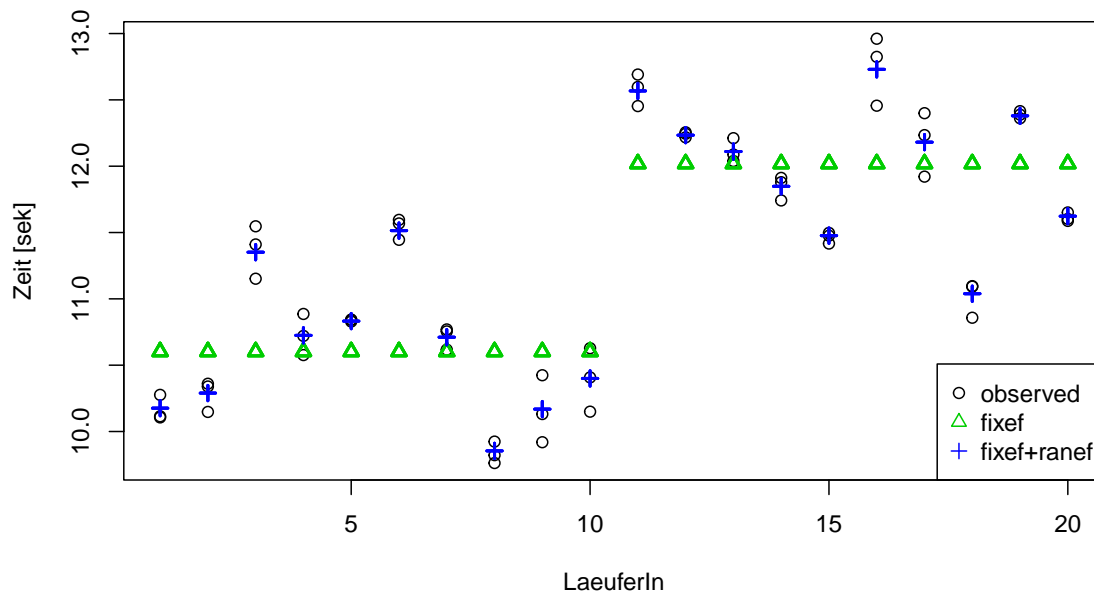
```
# compare estimates
cbind("byHand" = c(hatBeta), "mlEst" = fixef(sprintML))

      byHand      mlEst
(Intercept) 10.602175 10.602175
sex          1.416772  1.416772
```

```
(d) # Plot the original data
with(sprint, plot(times ~ athlete, xlab = "LaeuferIn", ylab = "Zeit [sek]"))

# add predictions of fixed effects
with(sprint, points(predict(sprintREML, level = 0) ~ athlete, pch = 2,
                    col = 3, lwd = 2))

# add predictions of fixed and random effects
with(sprint, points(predict(sprintREML) ~ athlete, pch = 3, col = 4, lwd = 2))
legend("bottomright", legend = c("observed", "fixef", "fixef+ranef"),
      pch = 1:3, col = c(1,3,4))
```



```
(e) # extra: nation
# -> nested design: athlete %in% nation
set.seed(42)
```

```

nationAthlete <- sample(1:4, 20, replace = TRUE)
nation <- rep(nationAthlete, each = M)
sprint$nation <- factor(nation, levels = 1:4, labels = c("A", "B", "C", "D"))

# model
fm2sprint <- lme(times ~ sex, random = ~1 | nation / athlete, data = sprint)
# summary(fm2sprint)

fixef(fm2sprint)

(Intercept)      sex1
  10.546183      1.390464

lapply( ranef(fm2sprint) , head)

$nation
  (Intercept)
A -0.36764964
B  0.18193210
C  0.09423914
D  0.09147840

$athlete
  (Intercept)
A/8  -0.3329026
A/18 -0.5381032
B/3   0.6233043
B/11  0.4488366
B/14 -0.2665663
B/15 -0.6366381

```


Aufgabe 3

Betrachten Sie erneut den Datensatz `Machines`, den Sie bereits von Blatt 1 kennen. Dieser Datensatz ist im R-Paket `nlme` enthalten.

(a) Fitten Sie folgendes Modell in R:

$$y_{ijk} = \beta_j + b_i + \varepsilon_{ijk}, \quad i = 1, \dots, 6, \quad j = 1, \dots, 3, \quad k = 1, \dots, 3$$

$$b_i \stackrel{\text{iid}}{\sim} N(0, \tau^2), \quad \varepsilon_{ijk} \stackrel{\text{iid}}{\sim} N(0, \sigma^2), \quad b_i, \varepsilon_{ijk} \text{ unabhängig}$$

Dabei ist y_{ijk} der Score für Arbeiter i an Maschine j in der k -ten Wiederholung. β_j sind also die festen Effekte der Maschinen und b_i die zufälligen Effekte der Arbeiter.

Lassen Sie sich die festen und die zufälligen Effekte ausgeben. Erstellen Sie einen Plot der Residuen.

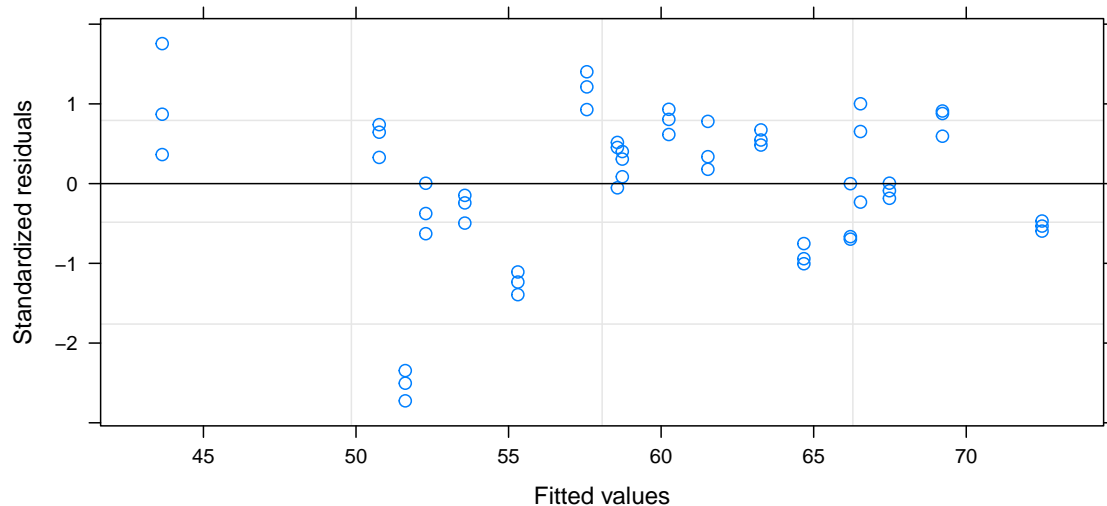
(b) Fitten Sie nun ein Modell, das zusätzlich die Interaktion von Arbeiter und Maschine berücksichtigt. Lassen Sie sich die geschätzte Kovarianzmatrix der zufälligen Effekte sowie die geschätzte Varianz der Fehler ausgeben.

(c) Löschen Sie zufällig 10 Beobachtungen aus dem Datensatz und berechnen Sie die Modelle aus (a) und (b) auf den unbalancierten Daten. Was fällt Ihnen auf wenn Sie die festen Effekte dieser beiden Modelle vergleichen?

Lösung 3

```
(a) # random intercept for each worker
fm1Machine <- lme(score ~ - 1 + Machine, random=~1|Worker, data=Machines)
# summary(fm1Machine)

# plot of residuals
plot(fm1Machine)
```



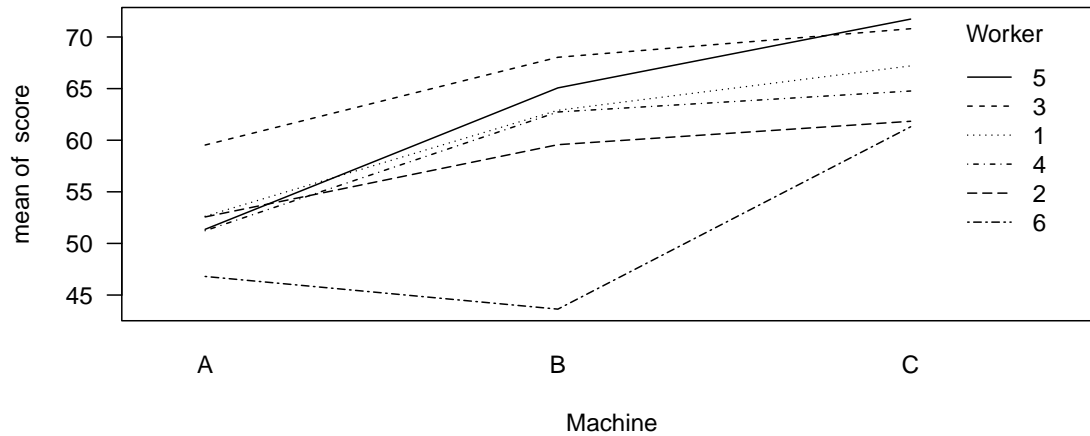
```
fixef(fm1Machine)

MachineA MachineB MachineC
52.35556 60.32222 66.27222
```

```
ranef(fm1Machine)

(Intercept)
6 -8.70711058
2 -1.59425968
4 -0.06931564
1 1.21035769
3 6.21174760
5 2.94858062
```

```
(b) ## generate an interaction plot:
## averaging the scores for each worker on each machine type,
## plot these averages against the machine type,
## and join the points for each worker.
with(Machines, interaction.plot(Machine, Worker, score, las=1))
```



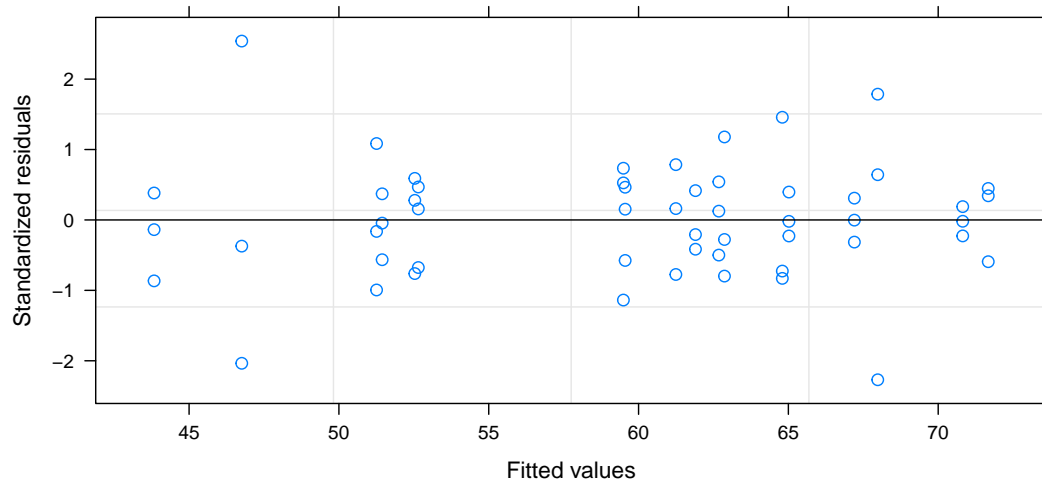
```
## if there were no interactions, the lines in the plots would be
## approximately parallel
```

- Erste Möglichkeit für eine Interaktion:

```

# # add random interaction term for machine within each worker
fm2Machine <- lme(score ~ - 1 + Machine, random=~1|Worker/Machine,
                  data=Machines)
# summary(fm2Machine)
plot(fm2Machine)

```



```

fixef(fm2Machine)

MachineA MachineB MachineC
52.35556 60.32222 66.27222

lapply( ranef(fm2Machine), head)

$Worker
  (Intercept)
6 -7.51429458
2 -1.37585675
4 -0.05981986
1  1.04454677
3  5.36077966
5  2.54464477

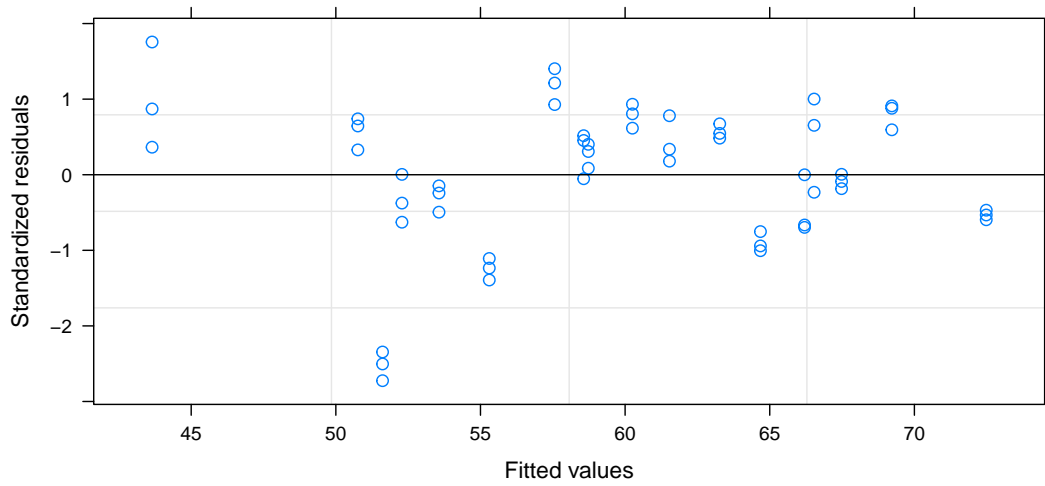
$Machine
  (Intercept)
6/A  1.9162774
6/B -8.9757070
6/C  2.4869652
2/A  1.5525655
2/B  0.6068543

```

2/C -2.9966316

- Zweite Möglichkeit für eine Interaktion:

```
# add correlation between random interactions,  
# do not model random effects for Worker  
fm3Machine <- lme(score ~ - 1 + Machine, random = ~ Machine - 1 | Worker,  
                 data = Machines)  
# summary(fm3Machine)  
plot(fm1Machine)
```



```
fixef(fm3Machine)
```

```
MachineA MachineB MachineC  
52.35556 60.32222 66.27222
```

```
raneff(fm3Machine)
```

```
MachineA MachineB MachineC  
6 -5.5916013 -16.5838056 -5.0300737  
2 0.1838844 -0.8033247 -4.2822823  
4 -1.0238706 2.3284558 -1.4146283  
1 0.3119890 2.5532237 0.9302963  
3 6.9692218 7.7793500 4.4735105  
5 -0.8496233 4.7261008 5.3231775
```

```
# get covariance matrix of b_i
```

```
(D <- getVarCov(fm3Machine))
```

```
Random effects variance covariance matrix
```

```

MachineA MachineB MachineC
MachineA 16.641 28.245 11.147
MachineB 28.245 74.396 29.184
MachineC 11.147 29.184 19.268
Standard Deviations: 4.0793 8.6253 4.3895

```

```
cov2cor(D) # get correlation
```

```
Random effects variance covariance matrix
```

```

MachineA MachineB MachineC
MachineA 1.00000 0.80275 0.62250
MachineB 0.80275 1.00000 0.77083
MachineC 0.62250 0.77083 1.00000
Standard Deviations: 1 1 1

```

```
### extra
```

```
# plot of original data and predictions of model (a) and model (b)
```

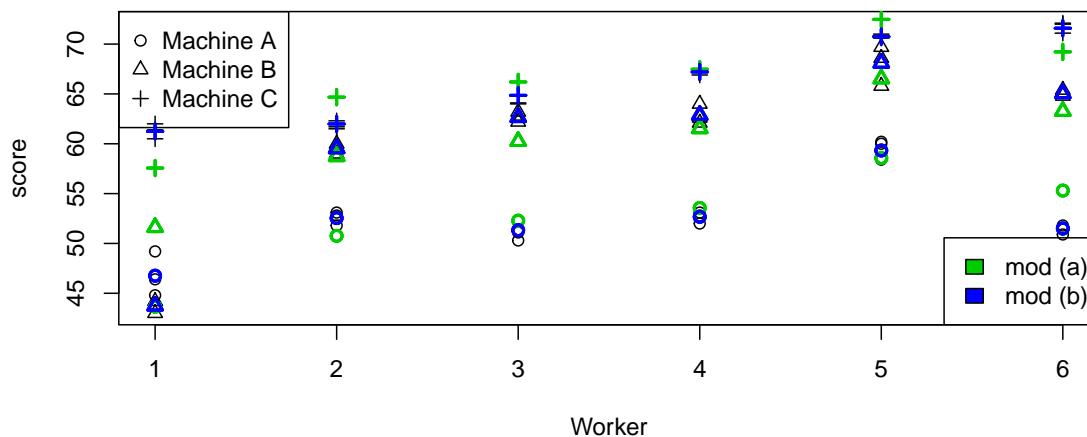
```
with(Machines, plot.default(score ~ Worker, pch=as.numeric(Machine)))
```

```
legend("topleft", legend=c("Machine A","Machine B","Machine C"), pch=1:3)
```

```
with(Machines, points(predict(fm1Machine) ~ Worker,
                       pch=as.numeric(Machine), col=3, lwd=2))
```

```
with(Machines, points(predict(fm3Machine) ~ Worker,
                       pch=as.numeric(Machine), col=4, lwd=2))
```

```
legend("bottomright", legend=c("mod (a)","mod (b)"), fil=3:4)
```



```

(c) # delete some observations to obtain unbalanced data
dim(Machines)

[1] 54 3

MachinesUnb <- Machines[sort(sample(1:54, 44)), ]

fm1MachineUnb <- lme(score ~ - 1 + Machine,
                    random = ~ 1 | Worker, data = MachinesUnb)
# summary(fm1MachineUnb)

fm3MachineUnb <- lme(score ~ - 1 + Machine,
                    random = ~ Machine - 1 | Worker, data = MachinesUnb)
# summary(fm3MachineUnb)

# fixed effects in balanced design
fixef(fm1Machine)

MachineA MachineB MachineC
52.35556 60.32222 66.27222

fixef(fm3Machine)

MachineA MachineB MachineC
52.35556 60.32222 66.27222

# fixed effects in unbalanced design
fixef(fm1MachineUnb)

MachineA MachineB MachineC
52.44653 60.09082 67.11095

fixef(fm3MachineUnb)

MachineA MachineB MachineC
52.40978 60.34640 66.89916

```

Aufgabe 4

Betrachten Sie erneut den Datensatz `Oxide`, den Sie bereits von Blatt 1 kennen. Dieser Datensatz ist im R-Paket `nlme` enthalten.

(a) Fitten Sie folgendes Modell in R:

$$y_{ijk} = \beta_0 + b_{i0} + b_{ij} + \varepsilon_{ijk}, \quad i = 1, \dots, 8, \quad j = 1, \dots, 3, \quad k = 1, \dots, 3$$
$$b_{i0} \stackrel{\text{iid}}{\sim} N(0, \tau_0^2), \quad b_{ij} \stackrel{\text{iid}}{\sim} N(0, \tau_1^2), \quad \varepsilon_{ijk} \stackrel{\text{iid}}{\sim} N(0, \sigma^2), \quad b_{i0}, b_{ij}, \varepsilon_{ijk} \text{ unabhängig}$$

Dabei ist y_{ijk} die Dicke der Oxidschicht gemessen an der k -ten Stelle an der j -ten Scheibe in der i -ten Parzelle. Lassen Sie sich die festen und die zufälligen Effekte ausgeben.

(b) Erstellen Sie einen Plot der Residuen.

Lösung 4

```
(a) # library(nlme)
# ?Oxide
fm1Oxide <- lme( Thickness ~ 1, random = ~ 1 | Lot/Wafer, data = Oxide )
# summary(fm1Oxide)

## fixed and random effects
fixef(fm1Oxide)

(Intercept)
  2000.153

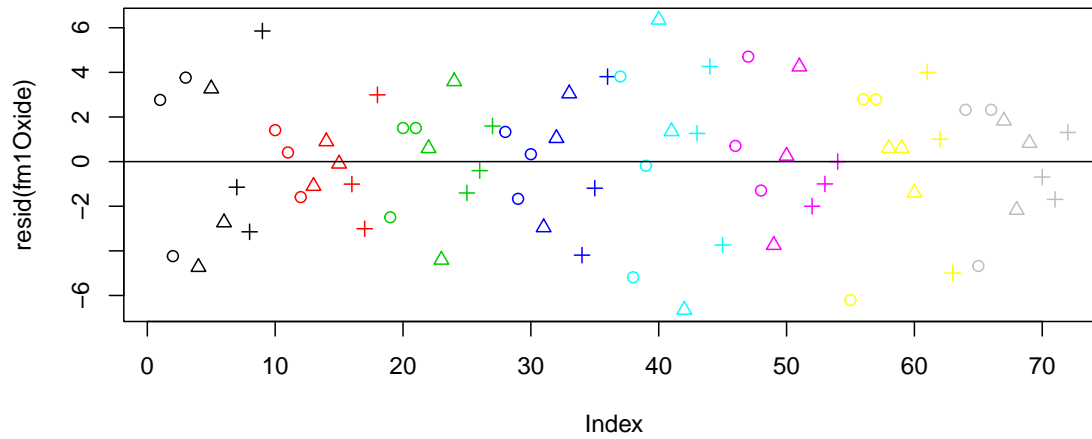
lapply( ranef(fm1Oxide) , head)

$Lot
  (Intercept)
1  -3.4634693
2 -11.2216405
3   0.8690159
4  -4.4710240
5  13.4634497
6  19.4080224

$Wafer
  (Intercept)
1/1   6.5459926
1/2 -11.9589391
1/3   4.4567261
```

```
2/1 0.6585929
2/2 -0.8337403
2/3 -2.9230068
```

```
(b) plot(resid(fm1Oxide), col=Oxide$Lot, pch=as.numeric(Oxide$Wafer))
abline(h=0)
```



Aufgabe 5

Betrachten Sie nun die blutdruck Daten.

```
# Simulation der SBD Daten #
# ===== #

set.seed(123)

Ji <- rep(6, 30)
n <- sum(Ji)
I <- length(Ji)
ID <- rep(1:I, Ji)
gender <- 1 * (ID > 15)
dosis <- rep(c(10, 20, 40, 80, 120, 160), I)
RI <- rnorm(I, 10)
RS <- rnorm(I, sd = 0.09)
eps <- rnorm(n, sd = 2)

fun <- function(x){
  -2 * sqrt(x) - x/20
}

SBD <- 150 + fun(dosis) - 2 * gender - dosis * gender/100 +
  RI[ID] + RS[ID] * dosis + eps

blutdruck <- data.frame(ID = ID, gender = gender, dosis = dosis, SBD = SBD)
```

(a) Fitten Sie ein lineares Modell, das die Abhängigkeit in den Daten ignoriert:

$$\text{SBD}_i = \beta_0 + \beta_1 \text{gender}_i + \beta_2 \text{dosis}_i + \beta_3 \text{gender}_i \text{dosis}_i + \varepsilon_i, \quad i = 1, \dots, 180, \quad \varepsilon_i \text{ unabh.}$$

(b) Fitten Sie nun ein Modell mit zufälligen Intercepts:

$$\text{SBD}_{ij} = \beta_0 + \beta_1 \text{gender}_i + \beta_2 \text{dosis}_{ij} + \beta_3 \text{gender}_i \text{dosis}_{ij} + b_i + \varepsilon_{ij}, \quad i = 1, \dots, 30, \quad j = 1, \dots, 6$$

$$b_i \stackrel{\text{iid}}{\sim} N(0, \tau^2), \quad \varepsilon_{ij} \stackrel{\text{iid}}{\sim} N(0, \sigma^2), \quad b_i, \varepsilon_{ij} \text{ unabhängig}$$

(c) Fügen Sie dem Modell aus (b) zufällige Steigungen über die Dosis hinzu. Lassen Sie sich die geschätzte Kovarianzmatrix der festen sowie die der zufälligen Effekte ausgeben.

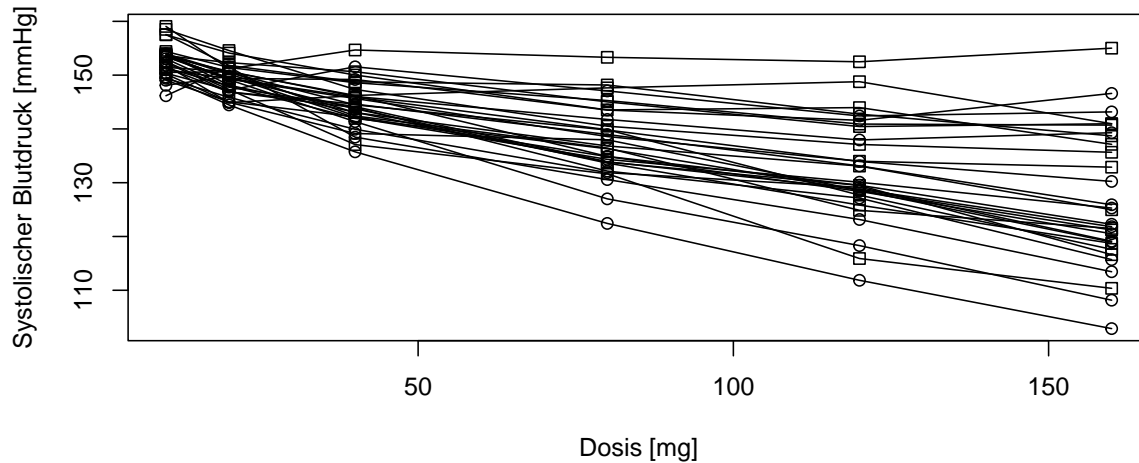
Lösung 5

```
### Plot Data ###
plot(0, xlim = c(10, 160), ylim = c(min(SBD), max(SBD)), xlab = "Dosis [mg]",
     ylab = "Systolischer Blutdruck [mmHg]", type = "n")
for (i in 1:I){
  data1 <- blutdruck[(ID == i),]
```

```

points(data1$dosis, data1$SBD, pch = data1$gender)
lines(data1$dosis, data1$SBD)  ##, lty = data1$gender + 1
}

```



```

(a) lm <- lm(SBD ~ gender + dosis + gender * dosis, data=blutdruck)
# summary(lm)

```

```

(b) lmm1 <- lme(SBD ~ gender + dosis + gender * dosis,
               random = ~ 1 | ID, data=blutdruck)
# summary(lmm1)

##### extra:
## compare estimates of fixed effects
cbind(coef(lm), fixef(lmm1))

               [,1]      [,2]
(Intercept) 153.81288362 153.81288362
gender       -2.58605855 -2.58605855
dosis        -0.15363045 -0.15363045
gender:dosis -0.02535405 -0.02535405

## compare variances of estimated of fixed effects
vcov(lm)

               (Intercept)      gender      dosis      gender:dosis

```

```
(Intercept)  1.50659257 -1.50659257 -0.0133574187  0.0133574187
gender       -1.50659257  3.01318514  0.0133574187 -0.0267148373
dosis       -0.01335742  0.01335742  0.0001863826 -0.0001863826
gender:dosis  0.01335742 -0.02671484 -0.0001863826  0.0003727652
```

```
vcov(lmm1)
```

```
              (Intercept)      gender      dosis  gender:dosis
(Intercept)  2.514644043 -2.514644043 -6.438893e-03  6.438893e-03
gender       -2.514644043  5.029288085  6.438893e-03 -1.287779e-02
dosis       -0.006438893  0.006438893  8.984501e-05 -8.984501e-05
gender:dosis  0.006438893 -0.012877785 -8.984501e-05  1.796900e-04
```

```
(c) lmm <- lme(SBD ~ gender + dosis + gender * dosis,
              random = ~ 1 + dosis|ID, data=blutdruck)
```

```
lmm
```

```
Linear mixed-effects model fit by REML
```

```
Data: blutdruck
```

```
Log-restricted-likelihood: -472.2735
```

```
Fixed: SBD ~ gender + dosis + gender * dosis
```

```
(Intercept)      gender      dosis  gender:dosis
153.81288362 -2.58605855 -0.15363045 -0.02535405
```

```
Random effects:
```

```
Formula: ~1 + dosis | ID
```

```
Structure: General positive-definite, Log-Cholesky parametrization
```

```
              StdDev      Corr
(Intercept)  0.17665062 (Intr)
dosis        0.07396025  0.934
Residual     2.35183115
```

```
Number of Observations: 180
```

```
Number of Groups: 30
```

```
# summary(lmm)
```

```
# covariance matrix of fixed effects
```

```
lmm$varFix # or equivalent: vcov(lmm)
```

	(Intercept)	gender	dosis	gender:dosis
(Intercept)	0.1706375882	-0.1706375882	-0.0006807744	0.0006807744
gender	-0.1706375882	0.3412751765	0.0006807744	-0.0013615489
dosis	-0.0006807744	0.0006807744	0.0003855270	-0.0003855270
gender:dosis	0.0006807744	-0.0013615489	-0.0003855270	0.0007710539

covariance matrix of random effects

`getVarCov(lmm)`

Random effects variance covariance matrix

	(Intercept)	dosis
(Intercept)	0.031205	0.0122050
dosis	0.012205	0.0054701

Standard Deviations: 0.17665 0.07396