

Aufgabe 2

Betrachten Sie folgendes Modell für den Datensatz mit den 100m-Läufen (vgl. Blatt 2):

$$\mathbf{y}_i = \beta_0 + \beta_1 \text{sex}_i + b_i + \boldsymbol{\varepsilon}_i, \quad i = 1, \dots, 20, \quad \mathbf{y}_i = (y_{i1}, \dots, y_{i3})'$$
$$b_i \stackrel{\text{iid}}{\sim} N(0, \tau^2), \quad \boldsymbol{\varepsilon}_i \stackrel{\text{iid}}{\sim} N(\mathbf{0}, \sigma^2 \mathbf{I}), \quad b_i, \boldsymbol{\varepsilon}_i \text{ unabhängig}$$

- (a) Fitten Sie das Modell mit REML-Schätzung.
- (b) Schätzen Sie das Modell bayesianisch.
Hinweis: Verwenden Sie die Funktion `MCMCg1mm()` aus dem R-Paket `MCMCg1mm`.
- (c) Vergleichen Sie die Schätzer für die festen Effekte zwischen dem mit REML und mit MCMC geschätzten Modell. Betrachten Sie im bayesianischen Modell Posteriori-Modus, -Erwartungswert und -Median. Vergleichen Sie die Schätzungen auch mit den wahren Werten.
Hinweis: Die Ziehungen der Posteriori-Dichte sind im Listenelement `So1` des Modells gespeichert.
- (d) Vergleichen Sie die Schätzungen sowie wahren Werte für die Varianzkomponenten τ^2 und σ^2 .
Hinweis: `mod$VCV` enthält die posteriori der Varianzkomponenten.
- (e) Vergleichen Sie die Prognosen für die zufälligen Effekte aus den beiden Modellen.
- (f) Berechnen Sie die HPD-Intervalle für die festen und zufälligen Effekte.
Hinweis: Funktion `HPDinterval()`.
- (g) Lassen Sie sich diagnostische Graphiken zu den MCMC-Ketten ausgeben.
Hinweis: `plot()`-Funktion auf `mod$So1` und `mod$VCV` anwenden.

Lösung 2

```
(a) library(nlme)

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

Linear mixed-effects model fit by REML
Data: sprint
      AIC      BIC    logLik
21.39124 29.63301 -6.695619

Random effects:
Formula: ~1 | athlete
      (Intercept)  Residual
StdDev:    0.533318 0.1430256

Fixed effects: times ~ sex
              Value Std.Error DF  t-value p-value
(Intercept) 10.602175 0.1706595 40  62.12471      0
```

```
sex1      1.416772 0.2413491 18 5.87022      0
Correlation:
(Intr)
sex1 -0.707

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-1.91519759 -0.44244911 0.06646893 0.40820115 1.78589389

Number of Observations: 60
Number of Groups: 20
```

```
(b) library(MCMCglmm)
# ?MCMCglmm

sprintMCMC <- MCMCglmm(times ~ sex, random = ~athlete, data=sprint,
                      nitt=13000, thin=10, burnin=3000,
                      pr=TRUE, # save posterior distribution of random effects
                      verbose = F)
summary(sprintMCMC)

Iterations = 3001:12991
Thinning interval = 10
Sample size = 1000

DIC: -41.54156

G-structure: ~athlete

      post.mean 1-95% CI u-95% CI eff.samp
athlete 0.3174 0.1446 0.5586 1000

R-structure: ~units

      post.mean 1-95% CI u-95% CI eff.samp
units 0.02137 0.0118 0.0301 903

Location effects: times ~ sex

      post.mean 1-95% CI u-95% CI eff.samp pMCMC
```

```

(Intercept) 10.6039 10.2202 10.9295 911.6 <0.001 ***
sex1        1.4262 0.9104 1.9016 1000.0 <0.001 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Check der autocorrelation of MCMC-Samples
# increase "thin" and "burnin" if necessary
# autocorr(sprintMCMC$Sol, lag=1)

```

```

(c) (compDF <- as.data.frame(
  rbind(c(10.6, 11.9 - 10.6, "Wahre Werte"),
        c(posterior.mode(sprintMCMC$Sol[,c("(Intercept)", "sex1")])),
        "Posteriori-Modus"),
        c(apply(sprintMCMC$Sol[,c("(Intercept)", "sex1")], MARGIN=2, FUN=mean),
        "Posteriori-Erwartungswert"),
        c(apply(sprintMCMC$Sol[,c("(Intercept)", "sex1")], MARGIN=2, FUN=median),
        "Posteriori-Median"),
        c(fixef(sprintREML), "REML-Schaetzung"))
))

```

	(Intercept)	sex1	V3
1	10.6	1.3	Wahre Werte
2	10.642257324331	1.43382438530028	Posteriori-Modus
3	10.6038745404202	1.42621426540009	Posteriori-Erwartungswert
4	10.6079848336376	1.42349880036636	Posteriori-Median
5	10.6021746089827	1.41677171992047	REML-Schaetzung

```

(d) (compDF2 <- as.data.frame(
  rbind(c(0.5^2, 0.2^2, "Wahre Werte"),
        c(apply(sprintMCMC$VCV, MARGIN=2, FUN=mean),
        "Posteriori-Erwartungswert"),
        c(getVarCov(sprintREML)[1,1], sprintREML$sigma^2,
        "REML-Schaetzung"))
))

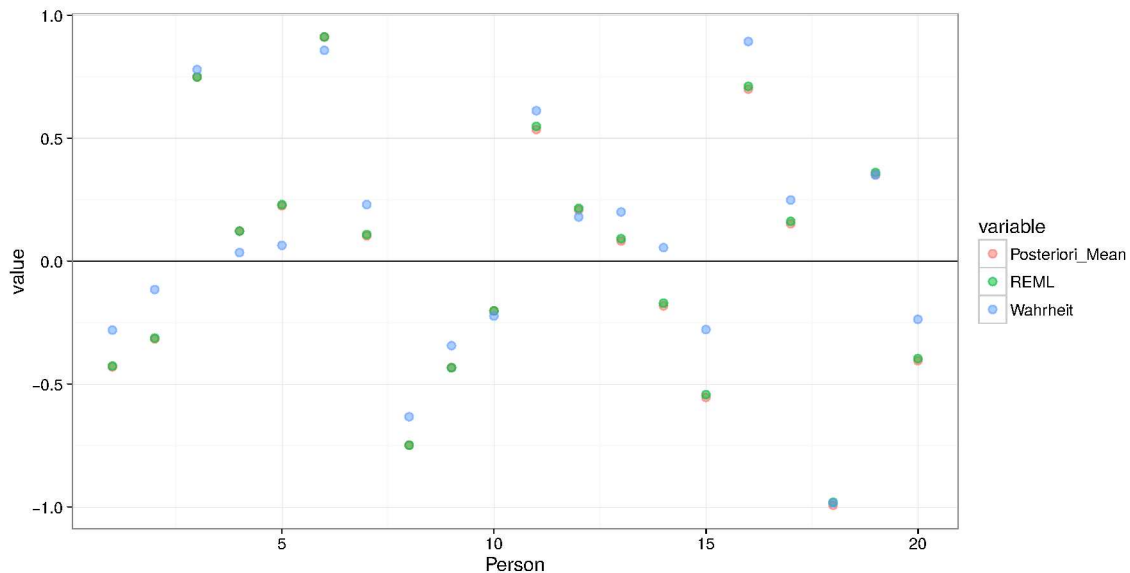
```

	athlete	units	V3
1	0.25	0.04	Wahre Werte
2	0.317426672054975	0.021365035057419	Posteriori-Erwartungswert
3	0.284428041095321	0.0204563217196773	REML-Schaetzung

```
(e) library(ggplot2)
library(reshape2)

compDF3 <- data.frame(
  Posteriori_Mean = apply(sprintMCMC$$sol[, -c(1:2)], MARGIN=2, FUN=mean),
  REML = random.effects(sprintREML)$`(Intercept)`,
  Wahrheit = b[seq(1, n*M, by=M)],
  Person = 1:n
)

ggplot(melt(compDF3, id.vars="Person"),
  aes(x=Person, y=value, colour=variable)) +
  geom_point(size=2, alpha=0.5) + theme_bw() + geom_hline(yintercept=0)
```



```
(f) HPDinterval(sprintMCMC$$sol, prob = 0.95)
```

	lower	upper
(Intercept)	10.22015658	10.92949873
sex1	0.91041054	1.90163367
athlete.1	-0.78443984	-0.06361628
athlete.2	-0.71625548	0.03685465
athlete.3	0.35642714	1.13774704
athlete.4	-0.24265980	0.50311964
athlete.5	-0.12771303	0.60424825
athlete.6	0.49733651	1.27746263
athlete.7	-0.23737001	0.51510433

```
athlete.8 -1.11408207 -0.36277242
athlete.9 -0.87042778 -0.09513543
athlete.10 -0.62188213 0.15436758
athlete.11 0.17136713 0.91772489
athlete.12 -0.18415833 0.57188337
athlete.13 -0.29254120 0.47097758
athlete.14 -0.56608559 0.18843430
athlete.15 -0.92018109 -0.15448950
athlete.16 0.33547326 1.08787801
athlete.17 -0.23296793 0.51894525
athlete.18 -1.34682382 -0.58310812
athlete.19 -0.04102732 0.71149007
athlete.20 -0.75265293 -0.02604758
attr(,"Probability")
[1] 0.95
```

```
(g) # plot(sprintMCMC£Sol) # fixed and random effects
# plot(sprintMCMC£VCV) # "units": estimation of error variance
```