Begleitskriptum zur Weiterbildung

# Gemischte Modelle in R

Prof. Dr. Guido Knapp
Email: guido.knapp@tu-dortmund.de

Braunschweig, 15.–17. April 2019

# Inhaltsverzeichnis

# Kapitel 1

# Einleitung

Dieser Weiterbildungskurs trägt den sehr allgemeinen Namen **Gemischte Modelle in R**. Die beiden wesentlichen Ziele dieses Kurses sind, die statistischen Grundlagen für die Gemischten Modelle und die Umsetzung der statistischen Analyse der Gemischten Modelle in der frei verfügbaren Statistiksoftware R (R Core Team, 2015) darzustellen.

Die Gemischten Modelle werden hier in zwei Untergruppen aufgeteilt: Gemischte Lineare Modelle und Gemischte Verallgemeinerte Lineare Modelle. Die Gemischten Linearen Modelle sind eine Erweiterung der klassischen Linearen Modelle für normalverteilte (stetige) Responsevariablen. Das klassische Lineare Modell parametrisiert den Erwartungswertvektor durch feste Effekte, für die man sich interessiert, und hat als einzige Variationsquelle den Fehlervektor, dessen Erwartungswertvektor gleich dem Nullvektor ist und dessen Kovarianzmatrix entweder die Unkorreliertheit der Fehler oder einer bis auf einen multiplikativen Faktor bekannten Korrelationsstruktur der Fehler widerspiegelt. Bei den Gemischten Linearen Modellen werden neben dem Fehlervektor weitere Variabilitätsursachen berücksichtigt. Diese weiteren Variabilitätsursachen werden durch Faktoren mit zufälligen Effekte beschrieben und induzieren dann eine Korrelationsstruktur der Responsevariablen. Für einen Faktor wird angenommen, dass die zufälligen Effekte aus einer Population möglicher Effekte gezogen werden. Wir betrachten hierarchische und kreuzklassifizierte Faktoren mit zufälligen Effekten und schauen uns Modelle für wiederholte Messungen und Longitudinaldaten an.

Die Gemischten Verallgemeinerten Linearen Modelle verknüpfen die Verallgemeinerten Linearen Modelle für nicht-normalverteilte Responsevariablen mit den Modellen mit zufälligen Effekten. Hier steht in erster Linie die Modellierung der Korrelationsstruktur für Longitudinaldaten im Vordergrund. Wir betrachten als nicht-normalverteilte Responsevariablen binäre Daten und Zähldaten. Bei der Analyse dieser Modelle ist es wichtig, den Unterschied zwischen bedingtem und marginalem Modell zu verstehen. Bei dem marginalen Modell wird eine Aussage für die Population getroffen, bei dem bedingten Modell für das Individuum. Bei der Analyse der Zähldaten betrachten wir kurz noch zwei Typen von Modellen, nämlich Modelle mit Overdispersion und Modelle mit 'zero-inflation'. Bei dem ersten Modelltyp ist die beobachtete Varianz größer als die unter der Verteilungsannahme zu erwartende, beim zweiten Modelltyp ist die beobachtete relative Häufigkeit von Nullen größer als die Wahrscheinlichkeit Nullen unter der Verteilungsannahme zu erwarten.

Die vier wesentlichen R Pakete, die hier im Bezug auf Gemischte Modelle besprochen werden, sind `lme4`, `nlme`, `gee` und `glmmADMB`. Die Datenbeispiele liegen alle als Textdateien vor, die mit der R Funktion *read.table()* eingelesen werden. Beim Einlesen der Datensätze wird beim R Code stets vorausgesetzt, dass die Datensätze im Arbeitsverzeichnis vorliegen. Mit der Funktion *setwd()* lässt sich das Arbeitsverzeichnis ändern.

Ergänzend zu diesem Skript sind Beispieldatensätze, Aufgaben und die R-Programme in diesem Skript auf der Webseite `www.statistik.tu-dortmund.de/gemischtemodelle.html` verfügbar. Diese Webseite ist nicht mit der Homepage des Autors verlinkt.

# Kapitel 2

# Gemischte Lineare Modelle

Wir betrachten zunächst einige allgemeine Aussagen für Gemischte Lineare Modelle, ohne zu tief in die mathematisch-statistische Theorie vorzustoßen. Danach betrachten wir zunächst das Modell der Einfachklassifikation mit zufälligen Effekten und erweitern dann dieses Modell mit kreuzklassifizierten bzw. hierarchischen zufälligen Effekten. Schließlich werden noch Modelle für Longitudinaldaten und *Repeated Measurements* diskutiert.

Das allgemeine Gemischte Lineare Modell ist gegeben durch

$$y = X\beta + Zu + e \ .$$

Dabei bezeichnet $y$ den $n$-dimensionalen beobachtbaren Zufallsvektor, $X$ eine bekannte Modellmatrix der festen Effekte, $\beta$ den unbekannten Vektor der festen Effekte, $Z$ eine bekannte Designmatrix der zufälligen Effekte, $u$ den nicht-beobachtbaren Zufallsvektor der zufälligen Effekte und $e$ den nicht-beobachtbaren Zufallsvektor der Fehler. Die Dimensionen der Matrizen und Vektoren sind dabei so gewählt, dass die obigen Matrizenmultiplikationen definiert sind.

Für den Zufallsvektor $u$ treffen wir die Annahme, dass er multivariat-normalverteilt ist mit Erwartungswertvektor 0 und Kovarianzmatrix $D$. Die unbekannten Parameter in der Kovarianzmatrix $D$ werden als Varianzkomponenten bezeichnet. Die zufälligen Fehler haben alle den Erwartungswert 0 und die Varianz $\sigma_e^2$. Die Fehler sind alle unkorreliert, so dass die Kovarianzmatrix von $e$ eine Diagonalmatrix $\sigma_e^2 I_n$ ist, wobei $I_n$ die $(n \times n)$-dimensionale Einheitsmatrix bezeichnet. Für die Kovarianzmatrix von $e$ kann auch eine andere positiv definite Matrix $R$ angenommen werden, wenn die Fehler nicht unkorreliert sind oder heteroskedastische Fehlervarianzen vorliegen.

Mit diesen Annahmen ergeben sich die folgenden ersten beiden Momente für den Zufallsvektor $y$:

$$\mathrm{E}(y) = X\beta \quad \text{und} \quad \mathrm{Cov}(y) = ZDZ^T + \sigma_e^2 I =: \Lambda$$

mit $Z^T$ der transponierten Matrix von $Z$.

Unter Normalverteilungsannahme gilt somit für $y$:

$$y \sim \mathcal{N}\left(X\beta, \Lambda\right).$$

Bei bekannter Kovarianzmatrix $\Lambda$ ist der gewichtete Kleinste-Quadrate-Schätzer für $\beta$ gegeben durch

$$\hat{\beta} = (X^T \Lambda^{-1} X)^{-1} X^T \Lambda^{-1} y,$$

falls $X$ vollen Spaltenrang besitzt, und $A^{-1}$ bezeichnet die inverse Matrix von $A$. Besitzt $X$ keinen vollen Spaltenrang, so muss eine geeignete Reparametrisierung der festen Effekte erfolgen.

Da die Kovarianzmatrix $\Lambda$ in der Regel nicht bekannt ist, müssen die unbekannten Varianzkomponenten geschätzt werden. Die Schätzung der Varianzkomponenten kann mit Hilfe der Maximum-Likelihood(ML)-Methode unter Ausnutzung der Normalverteilung erfolgen.

Spezifizieren wir die Kovarianzmatrix $\Lambda$ in der Form $V = \sum_{i=0}^{r} Z_i Z_i^T \sigma_i^2$, d.h. die Kovarianzmatrix hängt von $r+1$ unbekannten Varianzkomponenten ab und die $Z_i$s sind bekannt, so können die ML-Schätzgleichungen für den Vektor der festen Effekt $\beta$ und für die Varianzkomponenten $\sigma_i^2$, $i = 0, \ldots, r$, angegeben werden. Nach McCulloch und Searle (2001) sind die ML-Schätzer $\hat{\beta}$ und $\hat{V}$ gegeben durch Lösungen der Gleichungen

$$X\hat{\beta} = X(X^T \hat{V}^{-1} X)^{-} X^T \hat{V}^{-1} y$$

und

$$-\frac{1}{2}\mathrm{tr}(\hat{V}^{-1} Z_i Z_i^T) - \frac{1}{2}(y - X\hat{\beta})^T \hat{V}^{-1} Z_i Z_i^T \hat{V}^{-1}(y - X\hat{\beta}) = 0, \quad i = 0, \ldots, r.$$

Dabei bezeichnet $A^{-}$ eine beliebige g-Inverse von $A$ und $\mathrm{tr}(A)$ die Spur (= Summe der Hauptdiagonalelemente) von $A$. Die ML-Schätzgleichungen sind oft nicht explizit lösbar, sondern numerische Verfahren müssen zur Bestimmung der ML-Schätzer herangezogen werden. Mit der ML-Theorie lassen sich asymptotische Stichprobenvarianzen für die Parameterschätzer herleiten, siehe McCulloch und Searle (2001).

Bei der ML-Methode ist jedoch bekannt, dass die wahren Varianzkomponenten unterschätzt werden. Die Verzerrung ist geringer, wenn die Restricted-Maximum-Likelihood(REML)-Methode verwendet wird.

Die Idee der REML-Methode besteht darin, dass die Likelihood einer Linearkombination der Elemente von $y$ maximiert wird. Diese werden als $k^T y$ derart gewählt, so dass $k^T y$ keine festen Effekte $\beta$ mehr enthält. Wir nutzen die maximale Anzahl $n - r_X$ von linear unabhängigen Vektoren $k$ mit $k^T y$ und setzen $K = [k_1 \; k_2 \; \cdots \; k_{n-r_x}]$. Dabei bezeichnet $r_X$ den Spaltenrang der Matrix $X$. Dies führt zur ML-Schätzung für $K^T y$ anstelle von $y$, wobei $K^T X = 0$ gilt und $K^T$ vollen Zeilenrang $n - r_X$ hat. Für die REML-Schätzung betrachten wir mit $K^T X = 0$ das dimensionsreduzierte Modell

$$K^T y \sim \mathcal{N}(0, K^T V K).$$

Die REML-Methode führt nur zu einer Schätzung von $V$ und nicht zu einer Schätzung von $\beta$. Zur Schätzung für $\beta$ wird die REML-Schätzung von V in die Schätzgleichung des ML-Schätzers für $\beta$ eingesetzt. Für die REML-Schätzer der Varianzkomponenten lassen sich wiederum asymptotische Stichprobenvarianzen herleiten, siehe McCulloch und Searle (2001).

Zur Frage, ob ML oder REML benutzt werden soll, gibt es keine eindeutige Antwort. Wenn die Schätzung der Parameter im Vordergrund steht, ist wohl REML zu bevorzugen. Wie oben schon erwähnt,

sind die REML-Varianzkomponentenschätzer weniger verzerrt als die ML-Varianzkomponentenschätzer. Diese Eigenschaft beruht darauf, dass die Freiheitsgrade, die zur Schätzung der festen Effekte benötigt werden, adäquat berücksichtigt werden. Die festen Effekte $\beta$ werden bei der Schätzung der Varianzkomponenten nicht berücksichtigt. Daher sind die REML-Schätzer invariant bezüglich des Wertes von $\beta$. Die REML-Methode liefert jedoch keinen direkten Schätzer für die festen Effekte $\beta$. Sollen für gegebene Daten verschiedene Modelle angepasst werden und die Modellanpassungen miteinander verglichen werden, so ist die ML-Methode zu benutzen.

Hypothesentests über die Parameter im Gemischten Linearen Modell lassen sich mit Hilfe des Likelihood-Quotienten-Tests konstruieren. Dazu betrachtetet man im allgemeinen Modell und im Modell unter der Nullhypothese jeweils die sog. *deviance*. Die *deviance* ist definiert als -2 mal den Wert der log-Likelihoodfunktion an der Stelle der Parameterschätzungen. Bezeichne $d_a$ die *deviance* im allgemeinen Modell und $d_0$ die *deviance* im Modell unter der Nullhypothese, so ist die Teststatistik der Likelihood-Quotienten-Tests gerade $d_0 - d_a$. Diese Teststatistik ist (approximativ) $\chi^2_\nu$-verteilt, also $\chi^2$-verteilt mit $\nu$ Freiheitsgraden. Die Anzahl der Freiheitsgrade ist gleich der Anzahl der Restriktionen unter der Nullhypothese. Wird also ein Hypothesentest nur über einen Parameter durchgeführt, so ist die approximative Testverteilung gleich der $\chi^2_1$-Verteilung.

Mit Hilfe der Parameterschätzer und zugehöriger Standardfehler lassen sich in üblicher Weise Konfidenzintervalle für die Parameter konstruieren. Diese werden als Wald-Konfidenzintervalle bezeichnet. Profile-Konfidenzintervalle sind jedoch bei Gemischten Linearen Modellen zu bevorzugen. Diese werden mit Hilfe des Likelihood-Quotienten-Test basierend auf der Profile-Likelihood konstruiert. Bei der Profile-Likelihood kann der Parameter, für den das Konfidenzintervall konstruiert werden soll, variieren und die anderen Parameter in der Likelihood nehmen den Wert ihrer (RE)ML-Schätzers an. In das 95%-Profile-Konfidenzintervall gelangen dann alle Werte des interessierenden Parameters, für den der Likelihood-Quotienten-Test die zugehörige Nullhypothese, dass der betrachtete Wert der wahre Wert ist, zum 5%-Niveau nicht verworfen werden kann.

Ein zufälliger Effekt wird als ein Effekt betrachtet der aus einer Population von Effekten stammt. Diese Population von zufälligen Effekten ist eine Zusatzannahme im Vergleich zu den festen Effekten. Die *Schätzung* der zufälligen Effekte wird als Vorhersage (*prediction*) bezeichnet. Betrachten wir die Modellschreibweise

$$y \sim \mathcal{N}(X\beta, \Lambda) \quad \Lambda = ZDZ^T + \sigma_e^2 I$$

und nehmen an, dass $y$ und $u$ gemeinsam multivariat normalverteilt sind, so gilt für den bedingten Erwartungswert $\mathrm{E}(u|y)$, also den Erwartungswert der zufälligen Effekte $u$ gegeben die Daten $y$, die folgende Gleichung

$$\mathrm{E}(u|y) = DZ^T\Lambda^{-1}(y - X\beta).$$

Ein naheliegender Prädiktor für $u$ ist dann

$$\tilde{u} = DZ^T\Lambda^{-1}(y - X\beta).$$

Schätzungen der unbekannten Parameter auf der rechten Seite der obigen Gleichung liefern

$$\hat{u} = \hat{D}Z^T\hat{\Lambda}^{-1}(y - X\hat{\beta}),$$

die besten linear unverzerrten Prädiktoren (BLUPs). Für diese Prädiktoren lassen sich Stichprobenvarianzen herleiten, siehe McCulloch and Searle (2001), und somit in üblicherweise auch Konfidenzintervalle bestimmen.

Aus der gemeinsamen multivariaten Normalverteilung von $y$ und $u$ folgt zudem, dass die bedingte Verteilung von $y$ gegeben die (unbeobachtbaren) Realisierungen von $u$ auch wiederum normalverteilt ist, und zwar

$$y|u \sim \mathcal{N}\left(X\beta + Zu, \sigma_e^2 I\right).$$

Das bedeutet, dass der Beobachtungsvektor $y$ aus bedingt unabhängigen Elementen besteht.

Die marginale Verteilung von $y$ ist dann

$$y \sim \mathcal{N}\left(X\beta, ZDZ^T + \sigma_e^2 I\right),$$

wie oben schon motiviert. Die Analyse von bedingtem und marginalem Modell spielt nachher eine Rolle bei den Gemischten Verallgemeinerten Linearen Modellen.

Im Folgenden werden verschiedene Gemischte Lineare Modelle vorgestellt und die entsprechende Analyse von Beispieldatensätzen in R durchgeführt. Die Analyse der Beispieldatensätze erfolgen mit der Funktion *lmer* im R Paket `lme4` und mit der Funktion *lme* im R Paket `nlme`.

## 2.1 Einfache Varianzanalyse

Wir betrachten zunächst den einfachsten Fall eines Gemischten Linearen Modells, die einfache Varianzanalyse mit zufälligen Effekten. Gegeben sei ein Faktor $\mathcal{A}$ mit $r$ zufälligen Effekten $a_1, \ldots, a_r$. Der Einfachheit halber konzentrieren wir uns auf ein balanciertes Design, d.h. die Anzahl der Messwiederholungen ist für jeden zufälligen Effekt identisch.

Modellstruktur:

$$y_{ij} = \mu + a_i + e_{ij}, \quad i = 1, \ldots, r,\ j = 1, \ldots, s, n = rs,$$

mit

- $y_{ij}$ – Zufallsvariable mit $\mathrm{E}(y_{ij}) = \mu$
- $\mu$ – allgemeines Mittel
- $a_i$ – zufälliger Effekt der i-ten Stufe des Faktors $\mathcal{A}$, $a_i \sim \mathcal{N}\left(0, \sigma_a^2\right)$
- $e_{ij}$ – Zufallsfehler, $e_{ij} \sim \mathcal{N}\left(0, \sigma_e^2\right)$

$a_1, \ldots, a_r, e_{11}, \ldots, e_{rs}$ sind stochastisch unabhängig.

Der Erwartungswert von $y_{ij}$ ist $\mathrm{E}(y_{ij}) = \mu$ und die Varianz $\mathrm{Var}(y_{ij}) = \sigma_a^2 + \sigma_e^2$. Für die Kovarianz zweier Zufallsvariablen gilt

$$\mathrm{Cov}(y_{ij}, y_{i'j'}) = \begin{cases} \sigma_a^2 & , \text{ falls } i = i', j \neq j', \\ 0 & , \text{ sonst.} \end{cases}$$

Dies bedeutet, dass die Zufallsvariablen innerhalb eines zufälligen Effekts korreliert sind mit

$$\text{Corr}(y_{ij}, y_{ij'}) = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2} \quad , \quad j \neq j'.$$

Diese Korrelationsstruktur wird uns noch häufiger begegnen. Sie wird als *compound symmetry* oder *exchangeable* Korrelationsstruktur bezeichnet.

**Beispiel 2.1.1** *(Verbeke and Molenberghs (1997))*
*Um die Wirksamkeit eines Antibiotikums nach zweijähriger Lagerung zu messen, werden zufällig acht Chargen (batches) aus der Menge der zur Verfügung stehenden Chargen gezogen und aus diesen Chargen werden jeweils zwei Präparate zufällig gezogen. Von Interesse ist nun die Größe der Chargenvariabilität. Die Daten aus der Datei* `Antibiotikum.txt` *sind in Tabelle 2.1 gegeben und die zugehörige ANOVA-Tafel in Tabelle 2.2 dargestellt.*

Tabelle 2.1: Konzentration der aktiven Komponente des Antibiotikums

| Charge | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Beobachtung | 40 | 33 | 46 | 55 | 63 | 35 | 56 | 34 |
| | 42 | 34 | 47 | 52 | 59 | 38 | 56 | 29 |

Tabelle 2.2: ANOVA-Tafel für die Konzentration der aktiven Komponente

| | Quadrat-summen (QS) | FG | Mittlere QS (MQS) | Erwartete MQS |
|---|---|---|---|---|
| Chargen | $SSA = 1708.4$ | 7 | $MSA = 244.062$ | $2\sigma_a^2 + \sigma_e^2$ |
| Fehler | $SSE = 32.5$ | 8 | $MSE = 4.062$ | $\sigma_e^2$ |
| Gesamt | $SSG = 1740.5$ | 15 | | |

Schaut man sich die ANOVA-Tafel in Tabelle 2.2 an, so sind erwartungstreue ANOVA-Schätzer für die beiden Varianzkomponenten $\sigma_a^2$ und $\sigma_e^2$ gegeben durch

$$\hat{\sigma}_a^2 = \frac{MSA - MSE}{2} = \frac{244.1 - 4.062}{2} = 120$$

und

$$\hat{\sigma}_e^2 = MSE = 4.062.$$

Man beachte, dass der erwartungstreue ANOVA-Schätzer für $\sigma_a^2$ nicht notwendigerweise positiv sein muss.

Analysieren wir den Datensatz in R. Die Variable *Batch* muss ein Faktor sein. Mit der Funktion *str()* können wir uns die Struktur eines R Objektes anschauen. Wir lesen den Datensatz ein, speichern ihn als Objekt `antibio` und setzen die Variable `Batch` als Faktor.

```
> antibio <- read.table("Antibiotika.txt", header=TRUE)
> antibio$Batch <- as.factor(antibio$Batch)
> str(antibio)
'data.frame':   16 obs. of  2 variables:
 $ Batch: Factor w/ 8 levels "1","2","3","4",..: 1 1 2 2 3 3 4 4 5 5 ...
 $ y    : int  40 42 33 34 46 47 55 52 63 59 ...
```

Die ersten Beobachtungen des Datensatzes sehen wie folgt aus:

```
> head(antibio)
  Batch  y
1     1 40
2     1 42
3     2 33
4     2 34
5     3 46
6     3 47
```

Nach Laden des R Pakets `lme4` können wir das Modell der Einfachklassifikation mit zufälligen Effekten in der Formelschreibweise `y ~ 1 + (1 | Batch)` modellieren. In den Klammern werden die zufälligen Effekte dargestellt. Die Standardeinstellung zur Schätzung der Varianzkomponenten ist dabei die REML-Methode.

```
> library(lme4)
> one.way  <- lmer(y ~ 1 + (1 | Batch), data=antibio)
```

Wir speichern das Ergebnis im R Objekt `one.way` ab und schauen uns das Ergebnis mit der *summary()* Funktion an.

```
> summary(one.way)
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ 1 + (1 | Batch)
   Data: antibio

REML criterion at convergence: 95

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.35132 -0.56487  0.09136  0.51635  1.12938

Random effects:
 Groups   Name        Variance Std.Dev.
 Batch    (Intercept) 120.000  10.954
```

```
 Residual                4.062   2.016
Number of obs: 16, groups:  Batch, 8

Fixed effects:
            Estimate Std. Error t value
(Intercept)   44.937      3.906   11.51
```

In dem Output wird zunächst die Formeleingabe wiederholt. Das REML-Kriterium bei Konvergenz dient im Wesentlichen als Indikator, dass die Lösung konvergiert hat. Falls es Konvergenzprobleme gibt, wird eine Warnung ausgegeben. Für die Schätzung der Varianzkomponenten wird die REML-Methode verwendet. Die Schätzung für $\sigma_a^2$ und $\sigma_e^2$ sind identisch mit den oben hergeleiteten ANOVA-Schätzern. Allgemein stimmen REML-Schätzer und ANOVA-Schätzer in balancierten (d.h. die Anzahl der Wiederholungen in allen Effekten ist gleich) Varianzkomponentenmodellen überein. Um die Stärke der Chargenvariabilität zu bestimmen, kann der Intraklassenkorrelationskoeffizient (ICC) berechnet werden, der durch $ICC = \sigma_a^2/(\sigma_a^2 + \sigma_e^2)$ bestimmt ist. In dem Beispiel wird der ICC geschätzt durch $120/124.062 = 0.967$, d.h. 96.7% der Variabilität der Antibiotikakonzentration kann durch die Chargen erklärt werden. Das allgemeine Mittel $\mu$ für die Antibiotikakonzentration wird durch $\hat{\mu} = 44.937$ geschätzt. Der Standardfehler beträgt 3.906; ein P-Wert wird nicht ausgegeben. Der Grund für die Nicht-Angabe eines P-Wertes liegt daran, dass die Bestimmung der Freiheitsgrade nicht klar ist. Es gibt verschiedene Approximationen für die Freiheitsgrade, der Entwickler des R Pakets `lme4` mag diese Approximationen aber wohl nicht.

Die Konfidenzintervalle für die drei Parameter des Modells lassen sich mit der Funktion *confint()* berechnen, die standardmäßig 95%-Konfidenzintervalle liefert. Die Profile-Konfidenzintervalle werden durch die wiederholte Anpassung einer ML-Anpassung bestimmt.

```
> confint(one.way)
Computing profile confidence intervals ...
                 2.5 %    97.5 %
.sig01        6.623338 18.490941
.sigma        1.322410  3.617427
(Intercept) 36.824888 53.050111
```

Die Konfidenzintervalle werden nicht für die Varianzen, sondern für die Standardabweichungen ausgegeben. `.sig01` ist der interne Name für die erste Standardabweichung, in unserem Falle $\sigma_a$, `.sigma` steht für $\sigma_e$, und `(Intercept)` für $\mu$. Aus dem ersten Konfidenzintervall lässt sich schließen, dass die Chargenvariabilität signifikant von Null verschieden ist.

Grafisch lassen sich die Konfidenzintervalle wie in Abbildung 2.1 darstellen. Die vertikalen Striche geben dabei die Grenzen der 50%, 80%, 90%, 95% und 99%-Konfidenzintervalle an. Dafür wird das R Paket `lattice` benötigt. Zuvor müssen mit der Funktion *profile()* die Werte aus dem *R* Objekt `one.way` extrahiert werden. Der R Code sieht wie folgt aus:

```
pr1 <- profile(one.way)
```

```
library(lattice)
xyplot(pr1, aspect=1.3, layout=c(3,1))
```

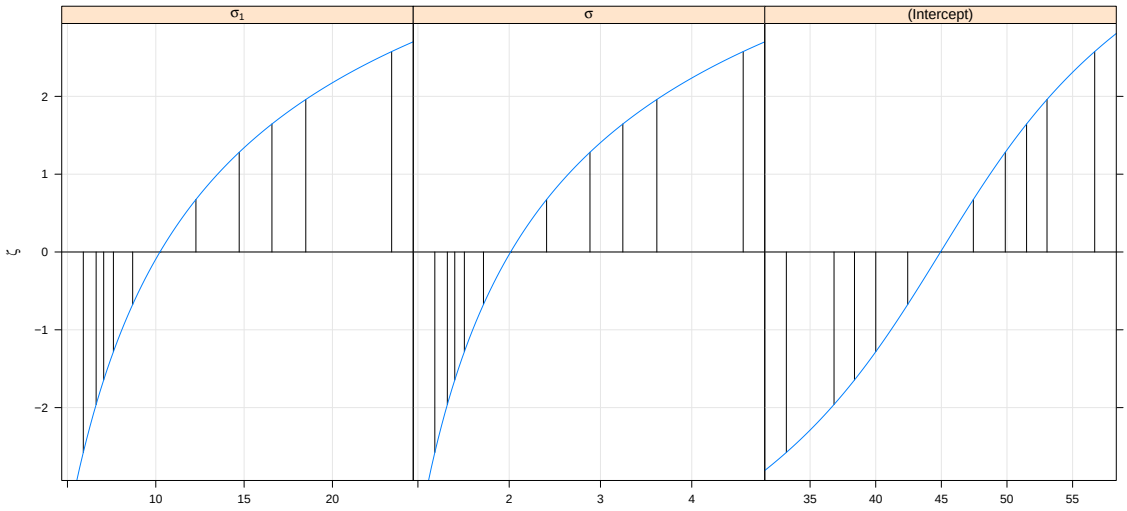


Abbildung 2.1: Profile Konfidenzintervalle für die drei Parameter im Antibiotika-Beispiel

Die besten linearen unverzerrten Prädiktoren (BLUP) für die zufälligen Effekte lassen sich mit der Funktion *ranef* extrahieren.

```
> ranef(one.way)
$Batch
  (Intercept)
1   -3.871959
2  -11.247119
3    1.536492
4    8.419974
5   15.795134
6   -8.297055
7   10.878361
8  -13.213828
```

95%-Prognoseintervalle für die BLUPs lassen sich mit

```
dotplot(ranef(one.way, condVar=TRUE))
```

darstellen, siehe Abbildung 2.2.

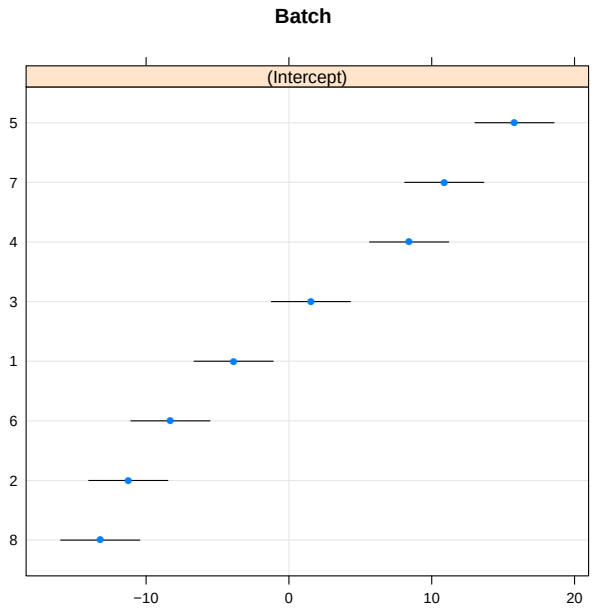


Abbildung 2.2: Prognoseintervalle für die zufälligen Effekte im Antibiotika-Beispiel

Schauen wir uns nun die Analyse des Modells an, wenn wir die ML-Methode zur Schätzung der unbekannten Parameter heranziehen. Ein Möglichkeit, dies zu tun, ist die Benutzung der Funktion *update* mit der folgenden Syntax:

```
> one.wayML <- update(one.way, REML=FALSE)
> summary(one.wayML)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: y ~ 1 + (1 | Batch)
   Data: antibio

     AIC      BIC   logLik deviance df.resid
   105.5    107.8    -49.8     99.5       13

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.3672 -0.5658  0.1044  0.5104  1.1439

Random effects:
 Groups   Name        Variance Std.Dev.
 Batch    (Intercept) 104.746  10.235
 Residual               4.062   2.016
Number of obs: 16, groups:  Batch, 8
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)   44.937      3.653    12.3
```

Wenn wir die ML-Methode nutzen, werden Statistiken zur Modellanpassung wie Akaikes Informationskriterium (AIC), Schwarz Bayes Informationskriterium (BIC), der Wert der log-Likelihood Funktion `logLik` an der Stelle der Parameterschätzungen sowie die *Deviance* (minus zweimal die log-Likelihood) an der Stelle der Parameterschätzungen angegeben. Diese Kriterien beziehen sich alle auf die Modellanpassung und werden benutzt, wenn verschiedene Modelle an dieselben Daten angepasst werden.

Wie schon erwähnt, ist die REML-Methode der ML-Methode bei der Schätzung der Varianzkomponenten im Allgemeinen vorzuziehen. Falls jedoch verschiedene Modelle für einen Datensatz miteinander verglichen werden sollen, ist es besser die ML-Methode zu benutzen.

Der ML-Schätzer für die Chargenvariabilität ist 104.746. Dieser Schätzer ist kleiner als der REML-Schätzer von 120. Die beiden Schätzer für $\sigma_e^2$ sind hier identisch. Dies ist im Allgemeinen nicht der Fall und liegt hier an dem einfachen Modell und dem balancierten Datensatz. Der Schätzer für das allgemeine Mittel ist in beiden Modellanpassungen mit 44.937 gleich, wiederum eine Konsequenz aus den einfachen Modell und dem balancierten Datensatz. Jedoch ist der Standardfehler des Schätzers für $\mu$ bei der Anpassung mit der ML-Methode mit 3.653 kleiner als 3.906, dem Standardfehler bei der Anpassung mit der REML-Methode.

Schließlich analysieren wir den Datensatz noch mit der Funktion *lme* aus dem R Paket `nlme`.

```
> library(nlme)
> one.wayNMLE <- lme(y ~ 1, data=antibio, random = ~ 1|Batch)
```

Der Aufbau der Funktion ist etwas anders. Zunächst wird in der Formel die abhängige Variable in Bezug zu den festen Effekten gesetzt. In der `random` Option wird dann die Struktur der zufälligen Effekte angegeben, wobei nach `|` die Gruppierungsvariablen stehen. Die REML-Methode ist auch hier die Standardeinstellung für die Schätzung der Varianzkomponenten. Die Ausgabe der Analyse sieht wie folgt aus:

```
> summary(one.wayNMLE)
Linear mixed-effects model fit by REML
 Data: antibio
       AIC      BIC    logLik
  101.0371 103.1613 -47.51855

Random effects:
 Formula: ~1 | Batch
        (Intercept) Residual
StdDev:    10.95445 2.015565
```

```
Fixed effects: y ~ 1
              Value Std.Error DF  t-value p-value
(Intercept) 44.9375  3.905623  8 11.50585       0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.35131912 -0.56486600  0.09135863  0.51634786  1.12937443

Number of Observations: 16
Number of Groups: 8
```

Im Gegensatz zu der Funktion *lmer* werden hier nur Schätzer für die Standardabweichungen dargestellt. Zudem werden für die Modellanpassung auch AIC und BIC Werte sowie der logLik-Wert angegeben. Alle Werte in dem obigen Output stimmen mit dem Output des *lmer*-Objektes überein. Zu erwähnen ist noch, dass für die festen Effekte die Freiheitsgrade der $t$-Verteilung und der zugehörige P-Wert angegeben werden.

Konfidenzintervalle für das allgemeine Mittel und die beiden Varianzkomponenten können mit der generischen Funktion *intervals()* bestimmt werden. Das Ergebnis sieht wie folgt aus:

```
> intervals(one.wayNMLE)
Approximate 95% confidence intervals

 Fixed effects:
               lower    est.    upper
(Intercept) 35.93112 44.9375 53.94388
attr(,"label")
[1] "Fixed effects:"

 Random Effects:
  Level: Batch
                   lower     est.    upper
sd((Intercept)) 6.430117 10.95445 18.66216

 Within-group standard error:
   lower     est.    upper
1.234790 2.015565 3.290036
```

Mit der Funktion *intervals()* werden sog. Wald-Konfidenzintervalle berechnet. Ausgenutzt wird die approximative Normalverteilung der (RE)ML-Schätzer. Die Wald-Konfidenzintervalle sind hier nicht großartig verschieden von den oben betrachteten Profile-Konfidenzintervallen.

BLUPs werden wiederum mit der Funktion *ranef()* erzeugt.

```
> ranef(one.wayNMLE)
  (Intercept)
1   -3.871959
2  -11.247119
3    1.536492
4    8.419974
5   15.795134
6   -8.297055
7   10.878361
8  -13.213828
>
```

Das Ergebnis ist identisch zu dem vom *R* Paket `lme4` erzeugten.

Abschließend in diesem Abschnitt wird auch die ML-Methode in der Funktion *lme* angewendet. Der R Code und der zugehörige Output sieht wie folgt aus:

```
> one.wayNMLE.ML <- lme(y ~ 1, data=antibio, random = ~ 1|Batch, method="ML")
> summary(one.wayNMLE.ML)
Linear mixed-effects model fit by maximum likelihood
 Data: antibio
       AIC      BIC    logLik
  105.5316 107.8493 -49.76578

Random effects:
 Formula: ~1 | Batch
        (Intercept) Residual
StdDev:    10.23455 2.015565

Fixed effects: y ~ 1
              Value Std.Error DF  t-value p-value
(Intercept) 44.9375  3.773192  8 11.90968       0

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.3671726 -0.5658247  0.1044097  0.5103755  1.1438783

Number of Observations: 16
Number of Groups: 8
```

Die Ergebnisse stimmen mit den Ergebnissen für die ML-Methode aus der *lmer*-Funktion überein. Nur der Standardfehler des festen Effekts 3.773192 ist hier größer als der Standardfehler von 3.653 berechnet mit der Funktion *lmer*.

## 2.2 Zweifache Varianzanalyse

In diesem Abschnitt betrachten wir zwei Faktoren $\mathcal{A}$ und $\mathcal{B}$. Werden alle Faktorstufenkombinationen der beiden Faktoren betrachtet, so spricht man von kreuzklassifizierten Effekten. Werden die Stufen des Faktors $\mathcal{B}$ innerhalb der Stufen des Faktors $\mathcal{A}$ betrachtet, so spricht man von hierarchischen Effekten.

### 2.2.1 Kreuzklassifizierte Effekte

Werden alle Faktorstufenkombinationen der beiden Faktoren betrachtet, so ist dies ein Modell der Zweifach-Kreuzklassifikation. Wir betrachten zunächst den Fall, dass alle Effekte der Stufen zufällig sind und zwischen den beiden Faktoren keine Wechselwirkungen auftreten.

Modellstruktur:

$$y_{ijk} = \mu + a_i + b_j + e_{ijk}, \quad i = 1, \ldots, r, j = 1, \ldots, s, k = 1, \ldots, t, n = rst,$$

mit

| | | |
|---|---|---|
| $y_{ijk}$ | – | Zufallsvariable mit $\mathrm{E}(y_{ijk}) = \mu$ |
| $\mu$ | – | allgemeines Mittel |
| $a_i$ | – | zufälliger Effekt der i-ten Stufe des Faktors $\mathcal{A}$, $a_i \sim \mathcal{N}\left(0, \sigma_a^2\right)$ |
| $b_j$ | – | zufälliger Effekt der j-ten Stufe des Faktors $\mathcal{B}$, $b_j \sim \mathcal{N}\left(0, \sigma_b^2\right)$ |
| $e_{ijk}$ | – | Zufallsfehler, $e_{ijk} \sim \mathcal{N}\left(0, \sigma_e^2\right)$ |

$a_1, \ldots, a_r, b_1, \ldots b_s, e_{111}, \ldots, e_{rst}$ sind stochastisch unabhängig.

**Beispiel 2.2.1** *(Anderson und Bancroft (1962, S. 276))*
*Untersucht wurde der Ertrag von Getreidefeldern in Abhängigkeit von drei zufällig ausgewählten Düngern (Faktor $(A)$) und zwei zufällig ausgewählten Sorten (Faktor $\mathcal{B}$). Der Ertrag wurde in pound pro Parzelle berechnet. Pro Dünger und Sorte wurden 6 Messungen gemacht. Ziel ist es, Aussagen über die Variabilität der Dünger und der Sorten zu machen. Die Daten aus der Datei* `Duenger.txt` *sind in Tabelle 2.3 dargestellt.*

Das Einlesen der Daten und die Datenvorverarbeitung erfolgt mit folgendem R Code:

```
> duenger <- read.table("Duenger.txt", header=TRUE)
> duenger$Duenger <- as.factor(duenger$Duenger)
> duenger$Sorte   <- as.factor(duenger$Sorte)
> str(duenger)
'data.frame':   36 obs. of  3 variables:
 $ Duenger: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ Sorte  : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 2 2 2 2 ...
 $ Messung: int  161 166 113 103 132 180 192 253 208 171 ...
```

Tabelle 2.3: Versuchsergebnisse für den Düngeversuch

| Dünger $i$ | | 1 | 2 | 3 |
|---|---|---|---|---|
| Sorte $j$ | $k$ | | | |
| 1 | 1 | 161 | 145 | 172 |
| | 2 | 166 | 231 | 204 |
| | 3 | 113 | 131 | 104 |
| | 4 | 103 | 158 | 135 |
| | 5 | 132 | 176 | 178 |
| | 6 | 180 | 216 | 175 |
| 2 | 1 | 192 | 232 | 227 |
| | 2 | 253 | 231 | 214 |
| | 3 | 208 | 190 | 144 |
| | 4 | 171 | 171 | 146 |
| | 5 | 196 | 242 | 186 |
| | 6 | 198 | 238 | 230 |

Das Modell der zweifachen Kreuzklassifikation ohne Wechselwirkung lässt sich mit der Funktion `lmer` analysieren, in der die zufälligen Effekte additiv verknüpft werden. Der entsprechende R Code mit der zugehörigen Analyse (ab jetzt wird die Ausgabe immer verkürzt dargestellt) sieht wie folgt aus:

```
> ertrag <- lmer(Messung ~ 1 + (1|Duenger) + (1|Sorte), data=duenger)
> summary(ertrag)
...
Random effects:
 Groups   Name        Variance Std.Dev.
 Duenger  (Intercept)   73.61   8.58
 Sorte    (Intercept)  897.70  29.96
 Residual             1133.70  33.67
Number of obs: 36, groups:  Duenger, 3; Sorte, 2

Fixed effects:
            Estimate Std. Error t value
(Intercept)   181.92      22.47   8.096
```

Der REML-Schätzer für die Varianzkomponente des Dünger ist $\hat{\sigma}_a^2 = 73.61$, für die Sorte $\hat{\sigma}_b^2 = 897.7$ und die Fehlervarianz wird mit $\hat{\sigma}_e^2 = 1133.7$ geschätzt. Der durchschnittliche Ertrag wird mit 181.92 pound pro Parzelle geschätzt.

Die 95%-Profile-Konfidenzintervalle für die wahren Varianzkomponenten sind gegeben durch:

```
> confint(ertrag)
Computing profile confidence intervals ...
```

```
                 2.5 %    97.5 %
.sig01        0.000000  40.48956
.sig02        6.593606  93.36994
.sigma       26.876110  44.10236
(Intercept) 127.517481 236.31533
```

Im 95%-Konfidenzintervall für $\sigma_a$ ist die Null enthalten, so dass zum Niveau von $\kappa = 0.025$ die Hypothese $H_0 : \sigma_a^2 = 0$ nicht verworfen werden kann. Die Variabilität der Sorte ist dagegen signifikant von Null verschieden. Bei den Varianzkomponentenmodellen kann es also vorkommen, dass der Wert Null ein möglicher Kandidat für die wahre Varianz ist, obwohl die empirische Varianz, hier des Düngers, von Null verschieden ist.

Die 95%-Profile-Konfidenzintervalle für das Dünger-Beispiel sind in Abbildung 2.3 dargestellt.

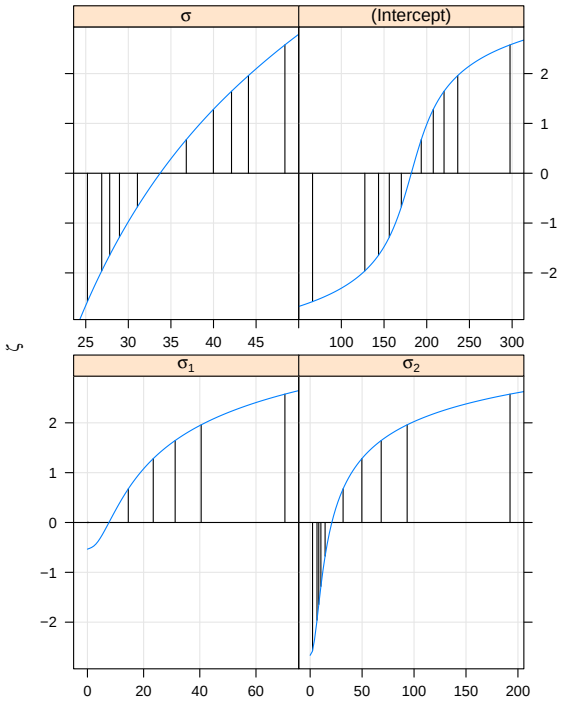


Abbildung 2.3: Profile-Konfidenzintervall-Plot für die vier Parameter im Dünger-Beispiel

Die Variabilität des Düngers ist nicht signifikant von Null verschieden, nur die Variabilität der Sorte ist signifikant. Ein Modell der Einfachklassifikation mit der Sorte als einzigen Faktor mit zufälligen Effekten sollte zur Modellanpassung daher ausreichen. Um dies zu überprüfen, kann die *anova()* Funktion genutzt werden. Zunächst werden das Modell der Zweifachklassifikation und das Modell der Einfachklassifikation mit der ML-Methode angepasst. Danach werden die beiden Modellanpassungen verglichen.

```
> ertrag1 <- lmer(Messung ~ 1 + (1|Duenger) + (1|Sorte), data=duenger, REML=FALSE)
> ertrag2 <- lmer(Messung ~ 1 + (1|Sorte), data=duenger, REML=FALSE)
> anova(ertrag1, ertrag2)
Data: duenger
Models:
ertrag2: Messung ~ 1 + (1 | Sorte)
ertrag1: Messung ~ 1 + (1 | Duenger) + (1 | Sorte)
        Df    AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
ertrag2  3 366.95 371.7 -180.47   360.95
ertrag1  4 368.66 375.0 -180.33   360.66 0.2857      1      0.593
```

Der P-Wert von 0.593 deutet daraufhin, dass es keinen signifikanten Unterschied zwischen den beiden Modellen gibt.

In den vorherigen Beispiel wurde angenommen, dass der Einfluss der zufälligen Effekte der beiden Faktoren additiv ist und keine Wechselwirkungen vorliegen. Liegen Wechselwirkungen zwischen den zufälligen Effekte vor, so betrachten wir das Modell der Zweifach-Kreuzklassifikation mit zufälliger Wechselwirkung.

Modellstruktur:

$$y_{ijk} = \mu + a_i + b_j + (ab)_{ij} + e_{ijk}, \quad i = 1, \ldots, r, j = 1, \ldots, s, k = 1, \ldots, t, n = rst,$$

mit

- $y_{ijk}$ – Zufallsvariable mit $\mathrm{E}(y_{ijk}) = \mu$
- $\mu$ – allgemeines Mittel
- $a_i$ – zufälliger Effekt der i-ten Stufe des Faktors $\mathcal{A}$, $a_i \sim \mathcal{N}\left(0, \sigma_a^2\right)$
- $b_j$ – zufälliger Effekt der j-ten Stufe des Faktors $\mathcal{B}$, $b_j \sim \mathcal{N}\left(0, \sigma_b^2\right)$
- $(ab)_{ij}$ – zufälliger Wechselwirkungseffekt zwischen der i-ten Stufe des Faktors $\mathcal{A}$ und der j-ten Stufe des Faktors $\mathcal{B}$, $(ab)_{ij} \sim \mathcal{N}\left(0, \sigma_{ab}^2\right)$
- $e_{ijk}$ – Zufallsfehler, $e_{ijk} \sim \mathcal{N}\left(0, \sigma_e^2\right)$

$a_1, \ldots, a_r, b_1, \ldots b_s, (ab)_{11}, \ldots, (ab)_{rs}, e_{111}, \ldots, e_{rst}$ sind stochastisch unabhängig.

**Beispiel 2.2.2** *(Hartung und Elpelt (2007), S. 735))*
*Um den Einfluss von Rasse in der Dosierung eines wachstumsfördernden Präparates auf die Gewichtszunahme einer Tierart zu untersuchen, werden jeweils 12 Tiere aus 3 zufällig ausgewählten Rassen mit dem wachtumsfördernden Präparat in 3 zufälligen Dosierungen verabreicht. Je 4 Tieren einer Rasse wird zufällig einer der Dosierungen verabreicht. Die Variabilität der Rasse und der Dosierungen sowie möglicher Wechselwirkungen ist von Interesse. Die Daten aus der Datei* `Gewichtszunahme.txt` *sind in Tabelle 2.4 dargestellt.*

Die Daten werden eingelesen und die Variablen `Rasse` und `Dosis` als Faktor definiert.

```
> zunahme <- read.table("Gewichtszunahme.txt", header=TRUE)
> zunahme$Rasse <- as.factor(zunahme$Rasse)
```

Tabelle 2.4: Ergebnisse der Gewichtszunahme

| Rasse $i$ | | 1 | 2 | 3 |
|---|---|---|---|---|
| Dosierung $j$ | $k$ | | | |
| 1 | 1 | 4 | 6 | 8 |
| | 2 | 4 | 5 | 7 |
| | 3 | 8 | 8 | 10 |
| | 4 | 6 | 7 | 9 |
| 2 | 1 | 6 | 9 | 11 |
| | 2 | 6 | 7 | 9 |
| | 3 | 8 | 6 | 8 |
| | 4 | 5 | 5 | 7 |
| 3 | 1 | 17 | 12 | 13 |
| | 2 | 10 | 10 | 11 |
| | 3 | 13 | 10 | 11 |
| | 4 | 8 | 9 | 10 |

```
> zunahme$Dosis <- as.factor(zunahme$Dosis)
> str(zunahme)
'data.frame':   36 obs. of  3 variables:
 $ Rasse: Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
 $ Dosis: Factor w/ 3 levels "1","2","3": 1 1 1 1 2 2 2 2 3 3 ...
 $ y    : int  4 4 8 6 6 6 8 5 17 10 ...
```

Mit `1|Rasse:Dosis)` können wir die zufällige Wechselwirkung in der Funktion *lmer* modellieren.

```
> gewicht <- lmer(y ~ 1 + (1|Rasse) + (1|Dosis) + (1|Rasse:Dosis), data=zunahme)
> summary(gewicht)
...
Random effects:
 Groups      Name        Variance Std.Dev.
 Rasse:Dosis (Intercept) 0.1898   0.4357
 Dosis       (Intercept) 5.3472   2.3124
 Rasse       (Intercept) 0.5139   0.7169
 Residual                3.6574   1.9124
Number of obs: 36, groups:  Rasse:Dosis, 9; Dosis, 3; Rasse, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)    8.417      1.441   5.841
```

Man beachte, dass die geschätzten Varianzkomponenten in der Reihenfolge $\sigma^2_{ab}$, $\sigma^2_b$ und $\sigma^2_a$ aufgeführt werden. Dies ist wichtig, um den Output der *confint()* Funktion richtig zu interpretieren. Die REML-

Schätzer für die Varianzkomponenten lauten: $\hat{\sigma}_a^2 = 3.6574$, $\hat{\sigma}_b^2 = 5.3472$ und $\hat{\sigma}_{ab}^2 = 0.1898$. Die durchschnittliche Gewichtszunahme wird mit 8.417 geschätzt.

Die 95%-Profile-Konfidenzintervalle sind gegeben durch:

```
> confint(gewicht)
Computing profile confidence intervals ...
                2.5 %    97.5 %
.sig01      0.0000000  2.236272
.sig02      0.5206738  5.842267
.sig03      0.0000000  2.945306
.sigma      1.4969223  2.539559
(Intercept) 5.1536220 11.679724
```

Von den drei Varianzkomponente ist nur $\sigma_b^2$ signifikant von Null verschieden.

Sind die Stufen des einen Faktors fest, so ergibt sich das folgende kreuzklassifizierte Modell mit einem festen und einem zufälligen Effekt sowie der zufälligen Wechselwirkung.

Modellstruktur:

$$y_{ijk} = \mu + \alpha_i + b_j + (\alpha b)_{ij} + e_{ijk}, \quad i = 1, \ldots, r, j = 1, \ldots, s, k = 1, \ldots, t, n = rst,$$

mit

- $y_{ijk}$ – Zufallsvariable mit $\mathrm{E}(y_{ijk}) = \mu + \alpha_i$
- $\mu$ – allgemeines Mittel
- $\alpha_i$ – fester Effekt der i-ten Stufe des Faktors $\mathcal{A}$
- $b_j$ – zufälliger Effekt der j-ten Stufe des Faktors $\mathcal{B}$, $b_j \sim \mathcal{N}\left(0, \sigma_b^2\right)$
- $(\alpha b)_{ij}$ – zufälliger Wechselwirkungseffekt zwischen der i-ten Stufe des Faktors $\mathcal{A}$ und der j-ten Stufe des Faktors $\mathcal{B}$, $(\alpha b)_{ij} \sim \mathcal{N}\left(0, \sigma_{\alpha b}^2\right)$
- $e_{ijk}$ – Zufallsfehler, $e_{ijk} \sim \mathcal{N}\left(0, \sigma_e^2\right)$

$b_1, \ldots b_s, (\alpha b)_{11}, \ldots, (\alpha b)_{rs}, e_{111}, \ldots, e_{rst}$ sind stochastisch unabhängig.

**Beispiel 2.2.3** *(Anderson und Bancroft (1952), S. 289)*
*In einem Düngeversuch wurde der Einfluss von Phosphat (Faktor $\mathcal{A}$ mit 3 festen Stufen) und 3 zufällig ausgewählten Stickstoffmengen untersucht. Pro Faktorstufenkombination wurde je zweimal das Gewicht von Graspflanzen (in Gramm) gemessen. Von Interesse ist zum einen, ob es einen Unterschied in den drei Phosphatkonzentration gibt und wie groß die Variabilität der Stickstoffmengen ist. Die Daten aus der Datei* `gras.txt` *sind in Tabelle 2.5 dargestellt.*

Das Einlesen der Daten und die Datenvorverarbeitung erfolgt mit dem folgenden R Code:

```
> gras            <- read.table("Gras.txt", header=TRUE)
> gras$Phosphat   <- as.factor(gras$Phosphat)
> gras$Stickstoff <- as.factor(gras$Stickstoff)
```

Tabelle 2.5: Versuchsergebnisse des Düngeversuchs

| | | Phosphatstufe $i$ | | |
|---|---|---|---|---|
| Stickstoffstufe $j$ | $k$ | 1 | 2 | 3 |
| 1 | 1 | 18.7 | 19.2 | 20.8 |
| | 2 | 17.5 | 21.3 | 20.5 |
| 2 | 1 | 20.8 | 18.8 | 22.0 |
| | 2 | 20.5 | 23.5 | 24.0 |
| 3 | 1 | 22.3 | 24.9 | 25.6 |
| | 2 | 22.9 | 24.2 | 27.1 |

```
> str(gras)
'data.frame':   18 obs. of  3 variables:
 $ Phosphat  : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 2 2 2 2 ...
 $ Stickstoff: Factor w/ 3 levels "1","2","3": 1 1 2 2 3 3 1 1 2 2 ...
 $ y         : num  18.7 17.5 20.8 20.5 22.3 22.9 19.2 21.3 18.8 23.5 ...
```

Die Auswertung des Datensatzes mit der *lmer()* Funktion erfolgt mit der folgenden Syntax, in der Phosphat nun ein fester Effekt ist.

```
> gewicht <- lmer(y ~ 1 + Phosphat + (1|Stickstoff) + (1|Phosphat:Stickstoff), data=gras)
> summary(gewicht)
...
Random effects:
 Groups              Name        Variance Std.Dev.
 Phosphat:Stickstoff (Intercept) 0.000    0.000
 Stickstoff          (Intercept) 5.657    2.378
 Residual                        1.566    1.251
Number of obs: 18, groups:  Phosphat:Stickstoff, 9; Stickstoff, 3

Fixed effects:
            Estimate Std. Error t value
(Intercept)  20.4500     1.4652  13.957
Phosphat2     1.5333     0.7225   2.122
Phosphat3     2.8833     0.7225   3.991

Correlation of Fixed Effects:
          (Intr) Phsph2
Phosphat2 -0.247
Phosphat3 -0.247  0.500
```

Die Varianzkomponente der Wechselwirkung $\sigma^2_{\alpha b}$ wird als Null geschätzt, so dass im Endeffekt ein Modell ohne Wechselwirkung gerechnet wird. Die Variabilität der Stickstoffmengen wird mit 5.657 geschätzt. Für die Kodierung des festen Effekts Phosphat ist zu bemerken, dass der Schätzwert

20.45 für `(Intercept)` der Schätzwert für $\mu + \alpha_1$ ist, also der ersten Stufe des Faktors Phosphat. Der Schätzwert 1.5333 für `Phosphat2` ist der Schätzwert für $\alpha_2 - \alpha_1$ und 2.8833 ist der Schätzwert für $\alpha_3 - \alpha_1$; also jeweils die Differenz der Stufen 2 und 3 zur Stufe 1.

### 2.2.2 Hierarchische Effekte

Hierarchische Effekte entstehen dadurch, dass jeweils innerhalb der zufälligen Effekte des Faktors $\mathcal{A}$ die zufälligen Effekte des Faktors $\mathcal{B}$ betrachtet werden. Wir betrachten zunächst das Modell mit zweifach-hierarchischer Struktur, indem beide Effekte zufällig sind.

Modellstruktur:

$$y_{ijk} = \mu + a_i + b_{ij} + e_{ijk}, \quad i = 1, \ldots, r, j = 1, \ldots, s, k = 1, \ldots, t, n = rst,$$

mit

| | | |
|---|---|---|
| $y_{ijk}$ | – | Zufallsvariable mit $\mathrm{E}(y_{ijk}) = \mu$ |
| $\mu$ | – | allgemeines Mittel |
| $a_i$ | – | zufälliger Effekt der i-ten Stufe des Faktors $\mathcal{A}$, $a_i \sim \mathcal{N}\left(0, \sigma_a^2\right)$ |
| $b_{ij}$ | – | zufälliger Effekt der j-ten Stufe des Faktors $\mathcal{B}$ innerhalb |
| | – | der i-ten Stufe des Faktors $\mathcal{A}$, $b_{ij} \sim \mathcal{N}\left(0, \sigma_b^2\right)$ |
| $e_{ijk}$ | – | Zufallsfehler, $e_{ijk} \sim \mathcal{N}\left(0, \sigma_e^2\right)$ |

$a_1, \ldots, a_r, b_{11}, \ldots b_{rs}, e_{111}, \ldots, e_{rst}$ sind stochastisch unabhängig.

**Beispiel 2.2.4** *(Ahrens (1967, S. 125))*
*Gemessen wurde die Konzentration von Kalzium (Messwert in %) im Blattwerk von Zuckerrüben. Es erfolgte eine zufällige Auswahl von 4 Rüben; je Rübe wurden per Zufall 3 Blätter ausgewählt und für jedes Blatt wurden zwei Messungen durchgeführt. Es interessiert die Variabilität der Zuckerrüben und die Variabilität der Blätter innerhalb der Zuckerrüben bezogen auf die Kalziumkonzentration. Die Daten aus der Datei* ***`Rueben.txt`*** *sind in Tabelle 2.6 dargestellt.*

Tabelle 2.6: Versuchsergebnisse für die Kalziumkonzentration in Zuckerrüben

| | Rübe $i$ | 1 | | | 2 | | | 3 | | | 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k$ | Blatt $ij$ | 11 | 12 | 13 | 21 | 22 | 23 | 31 | 32 | 33 | 41 | 42 | 43 |
| 1 | | 3.28 | 3.52 | 2.88 | 2.46 | 1.87 | 2.19 | 2.77 | 3.74 | 2.55 | 3.78 | 4.07 | 3.31 |
| 2 | | 3.09 | 3.48 | 2.80 | 2.44 | 1.92 | 2.19 | 2.66 | 3.44 | 2.55 | 3.87 | 4.12 | 3.31 |

Das Einlesen der Daten erfolgt in der gewohnten Weise; ebenso die Datenvorverarbeitung.

```
> rueben <- read.table("Rueben.txt", header=TRUE)
> rueben$Ruebe <- as.factor(rueben$Ruebe)
> rueben$Blatt <- as.factor(rueben$Blatt)
> str(rueben)
'data.frame':   24 obs. of  3 variables:
```

```
 $ Ruebe  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 2 2 2 2 ...
 $ Blatt  : Factor w/ 3 levels "1","2","3": 1 1 2 2 3 3 1 1 2 2 ...
 $ Messung: num  3.28 3.09 3.52 3.48 2.88 2.8 2.46 2.44 1.87 1.92 ...
```

Die ersten Daten sehen wie folgt aus.

```
> head(rueben)
  Ruebe Blatt Messung
1     1     1    3.28
2     1     1    3.09
3     1     2    3.52
4     1     2    3.48
5     1     3    2.88
6     1     3    2.80
```

Schauen wir uns mit der Funktion *xtabs* die Anzahl der Wiederholungen der Faktorstufenkombinationen an, so sehen wir, dass der Datensatz genauso wie bei der zweifachen Kreuzklassifikation konstruiert ist.

```
> xtabs(~ Ruebe + Blatt, data=rueben)
     Blatt
Ruebe 1 2 3
    1 2 2 2
    2 2 2 2
    3 2 2 2
    4 2 2 2
```

Um diesen Datensatz geeignet mit der Funktion *lmer* analysieren zu können, muss eine weitere Datenvorarbeitung geschehen. Wir müssen die hierarchische Struktur des Modells darstellen. Das geschieht mit der Variablen `sample`, die in dem folgenden R Code erzeugt wird. Mit der Funktion *xtabs* können wir uns nun die Anzahl der Faktorstufenkombinationen von `Ruebe` und `sample` anschauen. Somit haben wir die korrekte hierarchische Datenstruktur erzeugt.

```
> rueben$sample <- with(rueben, factor(Ruebe:Blatt))
> xtabs(~sample + Ruebe, data=rueben, sparse=TRUE)
12 x 4 sparse Matrix of class "dgCMatrix"
    1 2 3 4
1:1 2 . . .
1:2 2 . . .
1:3 2 . . .
2:1 . 2 . .
2:2 . 2 . .
2:3 . 2 . .
```

```
3:1 . . 2 .
3:2 . . 2 .
3:3 . . 2 .
4:1 . . . 2
4:2 . . . 2
4:3 . . . 2
```

Nun können wir mit dem schon bekannten R Code den Datensatz analysieren.

```
> two.way.nested  <- lmer(Messung ~ 1 +  (1|Ruebe) + (1|sample), data=rueben)
> summary(two.way.nested)
...
Random effects:
 Groups   Name        Variance Std.Dev.
 sample   (Intercept) 0.161060 0.40132
 Ruebe    (Intercept) 0.365223 0.60434
 Residual             0.006654 0.08157
Number of obs: 24, groups:  sample, 12; Ruebe, 4

Fixed effects:
            Estimate Std. Error t value
(Intercept)    3.012      0.324   9.295
```

Die Schätzer für die Varianzkomponenten lauten: $\hat{\sigma}_a^2 = 0.365$, $\hat{\sigma}_{ab}^2 = 0.161$ und $\hat{\sigma}_e^2 = 0.07$. Das allgemeine Mittel wird durch $\hat{\mu} = 3.012$ geschätzt.

Die obige Datenvorverarbeitung diente zur Illustration dessen, was bei `Ruebe:Blatt` passiert. Diese Schreibweise kann auch direkt in der *lmer* Funktion angegeben werden. Mit dem folgenden R Code erhalten wir dasselbe Ergebnis wie oben:

```
two.way.nested.2  <- lmer(Messung ~ 1 +  (1|Ruebe) + (1|Ruebe:Blatt), data=rueben)
```

Bei hierarchischen Modellen (*multi-level Modellen*) ist die Anwendung der Funktion *lme* im R Paket `nlme` vielleicht intuitiver. Bei der Angabe der Struktur der `random` Optionen können jeweils durch Schrägstrich / getrennt mehrere hierarchische Faktoren bwz. Gruppierungsvariablen angegeben werden.

```
> library(nlme)
> two.way.nested2 <- lme(fixed=Messung ~ 1, random = ~ 1|Ruebe / Blatt, data=rueben)
> summary(two.way.nested2)
Linear mixed-effects model fit by REML
 Data: rueben
       AIC      BIC    logLik
  10.17294 14.71492 -1.086471
```

```
Random effects:
 Formula: ~1 | Ruebe
        (Intercept)
StdDev:   0.6043356

 Formula: ~1 | Blatt %in% Ruebe
        (Intercept)   Residual
StdDev:   0.4013232 0.08157314

Fixed effects: Messung ~ 1
               Value Std.Error DF t-value p-value
(Intercept) 3.012083 0.3240437 12  9.2953       0
...
```

Das Ergebnis ist identisch zu den obigen Ergebnis mit der Funktion *lmer*

Abschließend betrachten wir noch das Modell, indem die Effekte des Faktors $\mathcal{A}$ fest und die des Faktors $\mathcal{B}$ zufällig sind.

Modellstruktur:

$$y_{ijk} = \mu + \alpha_i + b_{ij} + e_{ijk}, \quad i = 1, \ldots, r, j = 1, \ldots, s, k = 1, \ldots, t, n = rst,$$

mit

| | | |
|---|---|---|
| $y_{ijk}$ | – | Zufallsvariable mit $\mathrm{E}(y_{ijk}) = \mu + \alpha_i$ |
| $\mu$ | – | allgemeines Mittel |
| $\alpha_i$ | – | fester Effekt der i-ten Stufe des Faktors $\mathcal{A}$ |
| $b_{ij}$ | – | zufälliger Effekt der j-ten Stufe des Faktors $\mathcal{B}$ innerhalb |
| | – | der i-ten Stufe des Faktors $\mathcal{A}$, $b_{ij} \sim \mathcal{N}\left(0, \sigma_b^2\right)$ |
| $e_{ijk}$ | – | Zufallsfehler, $e_{ijk} \sim \mathcal{N}\left(0, \sigma_e^2\right)$ |

$b_{11}, \ldots b_{rs}, e_{111}, \ldots, e_{rst}$ sind stochastisch unabhängig.

Dieses Modell lässt sich mit der *lmer* Funktion analysieren. Die Syntax für das Rübenbeispiel lautet:

```
> two.way.nested.fe  <- lmer(Messung ~ 1 +  Ruebe + (1|sample), data=rueben)
```

## 2.3 Wiederholte Messungen und Longitudinaldaten

Wir betrachten nun Versuche mit mehreren Individuen, an denen wiederholt Messungen durchgeführt werden. Wenn diese wiederholten Messungen über die Zeit hinweg gemacht werden, so spricht man von Longitudinaldaten. Wir betrachten zunächst zwei Modelle für Longitudinaldaten, wobei die Zeit die einzige erklärende Variable sein soll. Im Beispiel später werden weitere erklärende Variablen berücksichtigt.

Modell mit zufälligem Achsenabschnitt

$$y_{ij} = \beta_0 + \beta_1 t_j + u_i + e_{ij}, \quad i = 1, \ldots, r,\ j = 1, \ldots, s, n = rs$$

mit

- $y_{ij}$ – Zufallsvariable mit $\mathrm{E}(y_{ij}) = \beta_0 + \beta_1 t_j$
- $\beta_0$ – Achsenabschnitt (Intercept)
- $\beta_1$ – Steigungsparameter
- $t_j$ – $j$-ter fester, bekannter Zeitpunkt
- $u_i$ – zufälliger Effekt der i-ten Person, $u_i \sim \mathcal{N}\left(0, \sigma_u^2\right)$
- $e_{ij}$ – Zufallsfehler, $e_{ij} \sim \mathcal{N}\left(0, \sigma_e^2\right)$

$u_1, \ldots, u_r, e_{11}, \ldots, e_{rs}$ sind stochastisch unabhängig.

Die Zufallsvariable $y_{ij}$ steht für die Beobachtung zum Zeitpunkt $t_j$ an Individuum $i$. Die Gesamtresiduen teilen sich in zwei Teile auf, der personen-spezifischen Zufallskomponente $u_i$, die über die Zeit hinweg konstant ist, und der Fehlerkomponente $e_{ij}$, die über die Zeit variiert. Für die Varianz von $y_{ij}$ gilt

$$\mathrm{Var}(y_{ij}) = \sigma_u^2 + \sigma_e^2$$

und die Beobachtungen an einer Person sind korreliert gemäß

$$\mathrm{Cor}(y_{ij}, y_{ik}) = \frac{\sigma_u^2}{\sigma_u^2 + \sigma_e^2}.$$

Diese *Intraklass-Korrelation* wird interpretiert als der Anteil der Gesamtvariabilität, der auf die Variabilität der Personen zurückzuführen ist. Ein Modell mit zufälligem Achsenabschnitt zwingt die Varianz jeder wiederholten Messung die gleiche zu sein und die Kovarianz zwischen zwei beliebigen Messungen an einer Person ebenfalls gleich zu sein. Diese Kovarianzstruktur wird als *compound symmetry* oder *exchangeable* bezeichnet. Diese Annahmen sind für Longitudinaldaten häufig nicht gerechtfertigt. Zum Beispiel sind Messungen, die zeitlich näher beianderliegen, häufig stärker korreliert als Messungen, die zeitlich weiter auseinanderliegen. Ein Modell, dass eine realistische Struktur für die Kovarianzen erlaubt, ist das Modell mit zufälligem Achsenabschnitt und zufälligem Steigungsparameter.

Modell mit zufälligem Achsenabschnitt und zufälligem Steigungsparameter

$$y_{ij} = \beta_0 + \beta_1 t_j + u_i + v_i t_j + e_{ij}$$

mit

| | | |
|---|---|---|
| $y_{ij}$ | – | Zufallsvariable mit $\mathrm{E}(y_{ij}) = \beta_0 + \beta_1 t_j$ |
| $\beta_0$ | – | Achsenabschnitt (Intercept) |
| $\beta_1$ | – | Steigungsparameter |
| $t_j$ | – | $j$-ter fester, bekannter Zeitpunkt |
| $u_i$ | – | zufälliger Effekt der i-ten Person, $u_i \sim \mathcal{N}\left(0, \sigma_u^2\right)$ |
| $v_i$ | – | zufälliger Steigungsparameter der i-ten Person, $v_i \sim \mathcal{N}\left(0, \sigma_v^2\right)$ |
| $e_{ij}$ | – | Zufallsfehler, $e_{ij} \sim \mathcal{N}\left(0, \sigma_e^2\right)$ |

$(u_i, v_i)$ sind bivariat normalverteilt mit Kovarianz $\sigma_{uv}$. Alle anderen Zufallskomponenten sind stochastisch unabhängig.

In diesem Modell gilt für die Varianz von $y_{ij}$

$$\mathrm{Var}(y_{ij}) = \sigma_u^2 + 2\sigma_{uv} t_j + \sigma_v^2 t_j^2 + \sigma_e^2,$$

die nun nicht mehr konstant für verschiedene Werte von $t_j$ ist. Für die Kovarianz zweier Beobachtungen am selben Individuum gilt:

$$\mathrm{Cov}(y_{ij}, y_{ik}) = \sigma_u^2 + \sigma_{uv}(t_j + t_k) + \sigma_v^2 t_j t_k.$$

Diese Kovarianz ist nun nicht mehr gleich für jedes Paar $t_j$ und $t_k$.

Die Varianzkomponenten können nun wieder durch die ML-Methode oder die REML-Methode geschätzt werden. Auch hier gilt die Empfehlung die REML-Methode zu benutzen, weil die ML-Methode dazu tendiert, die wahren Varianzkomponenten zu unterschätzen. Ein Modellvergleich kann wiederum mit dem Likelihood-Quotienten-Test durchgeführt werden. Wenn die Modelle durch die REML-Methode geschätzt wurden, so kann der Likelihood-Quotienten-Test für Varianzkomponenten nur durchgeführt werden, wenn in beiden Modellen dieselben festen Effekte vorliegen, siehe Longford (1993).

**Beispiel 2.3.1** *(Everitt, Hothorn (2010), Proudfoot et al. (2003))*
*Die Datei BtheB.txt enthält die Daten eines randomisierten klinischen Versuches zur Depression. Dabei wurden zwei psychotherapeutische Verfahren, 'Beat the Blues' (BtheB) und 'Treatment as Usual' (TAU), angewendet. Der Datensatz ist auch in dem* R *Paket* `HSAUR2` *verfügbar. Eine Zielvariable war Beck Depression Inventory II (BDI). BDI wurde pro Patient fünfmal gemessen, vor der Behandlung, 2 Monate, 3 Monate, 5 Monate und 8 Monate nach Behandlung. Neben dem Effekt der Behandlung (*`drug`*, yes oder no) ist auch noch die Länge der aktuellen depressiven Episode von Interesse (*`length`*, weniger oder mehr als sechs Monate).*

Lesen wir den Datensatz ein und schauen uns die ersten Beobachtungen an.

```
> BtheB <- read.table("BtheB.txt", header=TRUE)
> head(BtheB)
  drug length treatment bdi.pre bdi.2m bdi.3m bdi.5m bdi.8m
1   No    >6m       TAU      29      2      2     NA     NA
```

```
2  Yes    >6m     BtheB      32     16     24     17     20
3  Yes    <6m       TAU      25     20     NA     NA     NA
4   No    >6m     BtheB      21     17     16     10      9
5  Yes    >6m     BtheB      26     23     NA     NA     NA
6  Yes    <6m     BtheB       7      0      0      0      0
```

Die Daten liegen in der 'weiten' Form, d.h. pro Patient gibt es eine Zeile. Für die Analyse mit der *lmer* Funktion im `lme4` Paket müssen die Daten aber in der 'langen' Form vorliegen, d.h. eine Beobachtung pro Messzeitpunkt. Mit dem folgenden R Code kann die Neuanordnung der Daten gemacht werden. Man beachte dabei, dass der BDI-Wert vor der Behandlung als erklärende Variable mit in das Modell aufgenommen werden soll.

```
> BtheB$subject <- factor(rownames(BtheB))
> nobs <- nrow(BtheB)
> BtheB_long <- reshape(BtheB, idvar = "subject",
+          varying = c("bdi.2m", "bdi.3m", "bdi.5m", "bdi.8m"),
+          direction = "long")
> BtheB_long$time <- rep(c(2, 3, 5, 8), rep(nobs,4))
> head(BtheB_long)
     drug length treatment bdi.pre subject time bdi
1.2m   No    >6m       TAU      29       1    2   2
2.2m  Yes    >6m     BtheB      32       2    2  16
3.2m  Yes    <6m       TAU      25       3    2  20
4.2m   No    >6m     BtheB      21       4    2  17
5.2m  Yes    >6m     BtheB      26       5    2  23
6.2m  Yes    <6m     BtheB       7       6    2   0
```

Es ist hier zu bemerken, dass der Datensatz nun nach den Zeitpunkten und innerhalb der Zeitpunkte nach den Patienten geordnet ist und nicht umgekehrt.

Zunächst überprüfen wir, ob das Modell mit zufälligem Achsenabschnitt schon hinreichend gut an die Daten angepasst wird, oder ein Modell mit zufälligem Achsenabschnitt und zufälligem Steigungsparameter angebracht ist.

```
> library(lme4)
> # Random Intercept Model
> BtheB_lmer1 <- lmer(bdi ~ bdi.pre + time + treatment + drug + length
+                + (1 | subject), data = BtheB_long,
+                REML = FALSE, na.action = na.omit)
> # Random Intercept and
> BtheB_lmer2 <- lmer(bdi ~ bdi.pre + time + treatment + drug + length
+                + (1 + time | subject), data = BtheB_long,
+                REML = FALSE, na.action = na.omit)
> anova(BtheB_lmer1, BtheB_lmer2)
```

```
Data: BtheB_long
Models:
BtheB_lmer1: bdi ~ bdi.pre + time + treatment + drug + length + (1 | subject)
BtheB_lmer2: bdi ~ bdi.pre + time + treatment + drug + length + (1 + time |
BtheB_lmer2:     subject)
            Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
BtheB_lmer1  8 1887.5 1916.6 -935.75   1871.5
BtheB_lmer2 10 1891.0 1927.4 -935.52   1871.0 0.4542      2     0.7969
```

Der Likelihood-Quotienten-Test mit einem P-Wert von 0.7969 zeigt an, dass das einfachere Modell mit dem zufälligen Achsenabschnitt schon ausreichend gut für die Daten ist. Das angepasste Modell sieht wie folgt aus:

```
> summary(BtheB_lmer1)
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: bdi ~ bdi.pre + time + treatment + drug + length + (1 | subject)
   Data: BtheB_long
...
Random effects:
 Groups   Name        Variance Std.Dev.
 subject  (Intercept) 48.78    6.984
 Residual             25.14    5.014
Number of obs: 280, groups:  subject, 97

Fixed effects:
             Estimate Std. Error t value
(Intercept)   3.26331    2.21371   1.474
bdi.pre       0.63968    0.07789   8.212
time         -0.70476    0.14639  -4.814
treatmentTAU  2.32908    1.67036   1.394
drugYes      -2.82495    1.72684  -1.636
length>6m     0.19708    1.63832   0.120

Correlation of Fixed Effects:
            (Intr) bdi.pr time   trtTAU drugYs
bdi.pre     -0.599
time        -0.227  0.020
treatmntTAU -0.360 -0.121 -0.018
drugYes     -0.318 -0.237 -0.022  0.323
length>6m   -0.244 -0.242 -0.036 -0.002  0.157
```

Der BDI-Wert sinkt über die Zeit (`time` $= -0.70476$), ist für die Kontrolle im Vergleich zu 'Beat the Blues' größer (`treatmentTAU` $= 2.32908$) sowie größer für Episoden länger als 6 Monate (`lenght>6m` $= 0.19708$). Ob diese Werte signifikant von Null verschieden sind, könnte mit Profile-Konfidenzintervallen

beantwortet werden. Diese Konfidenzintervalle adjustieren jedoch nicht für das multiple Testproblem. Eine Möglichkeit bezüglich des multiplen Testproblems zu adjustieren, besteht darin, die Funktion *cftest()* aus dem `multcomp` Paket anzuwenden. Dazu muss nur das *lmer*-Objekt in die *cftest()*-Funktion übergeben werden.

```
> library(multcomp)
> cftest(BtheB_lmer1)

         Simultaneous Tests for General Linear Hypotheses

Fit: lmer(formula = bdi ~ bdi.pre + time + treatment + drug + length +
    (1 | subject), data = BtheB_long, REML = FALSE, na.action = na.omit)

Linear Hypotheses:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept) == 0   3.26331    2.21371   1.474    0.140
bdi.pre == 0       0.63968    0.07789   8.212 2.22e-16 ***
time == 0         -0.70476    0.14639  -4.814 1.48e-06 ***
treatmentTAU == 0  2.32908    1.67036   1.394    0.163
drugYes == 0      -2.82495    1.72684  -1.636    0.102
length>6m == 0     0.19708    1.63832   0.120    0.904
---
...
(Univariate p values reported)
```

Nur der BDI-Wert vor der Behandlung und die Zeit sind signifikant. Mit den Profile-Konfidenzintervallen für die festen Effekte kommen wir zu demselben Ergebnis.

Die Modellannahmen des Modells bezüglich der Normalverteilungsannahme für die Zufallskomponenten lassen sich mit Hilfe von Quantile-Quantile-Plots (Q-Q-Plots) überprüfen. Wir überprüfen dazu einmal die Normalverteilungsannahme für die zufälligen Achsenabschnitte und einmal für die Residuen. Die Prädiktoren erhalten wir dabei wieder mit der Funktion *ranef*. Die Residuen werden durch die Funktion *residuals* herausgeschrieben. Die Funktion *qqnorm* erstellt den Q-Q-Plots. Wenn die Punkte des Q-Q-Plots nicht zu stark von der Winkelhalbierenden abweichen, so ist die Modellannahme gerechtfertigt. Mit der Funktion *qqline* wird die Winkelhalbierende in den Q-Q-Plot eingezeichnet.

```
layout(matrix(1:2, ncol = 2))
> qint <- ranef(BtheB_lmer1)$subject[["(Intercept)"]]
> qres <- residuals(BtheB_lmer1)
> qqnorm(qint, ylab = "Estimated random intercepts", xlim = c(-3,3),
+              ylim = c(-20, 20), main = "Random intercepts")
> qqline(qint)
> qqnorm(qres, xlim = c(-3,3), ylim = c(-20, 20),
```

```
+              ylab = "Estimated residuals",
+              main = "Residuals")
> qqline(qres)
```

Die beiden Q-Q-Plots sind in Abbilung 2.4 dargestellt. Es ergeben sich keine größeren Abweichungen von der Linearität.

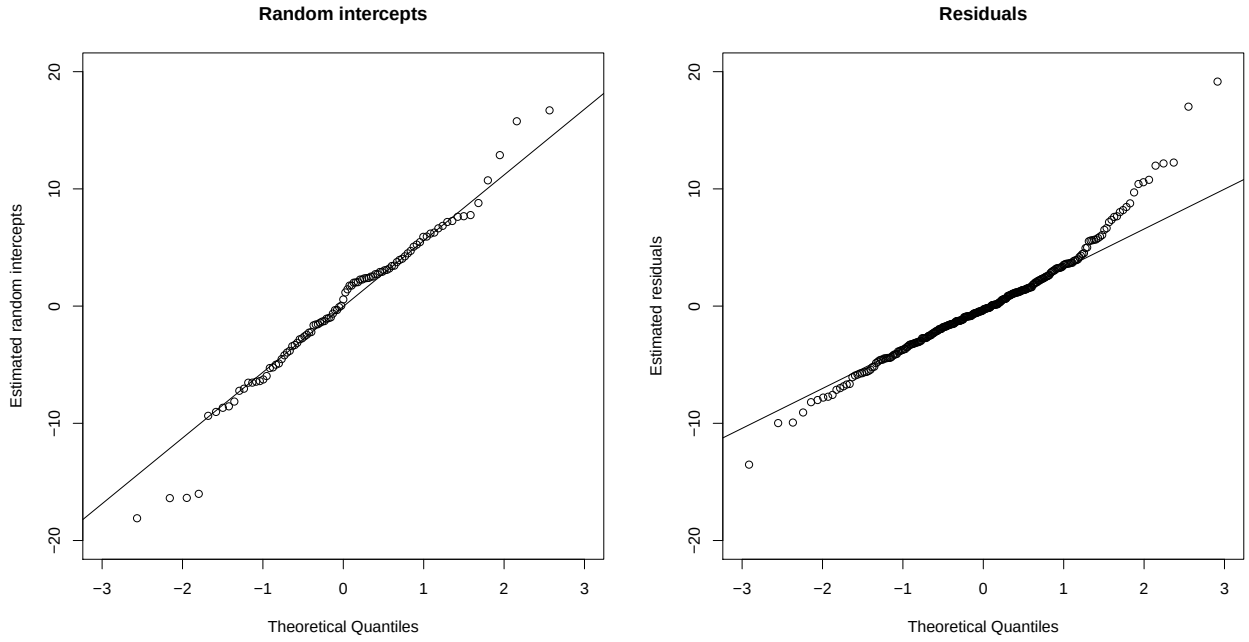


Abbildung 2.4: Quantile-Quantile-Plots für die prädiktiven zufälligen Achsenabschnitte und die Residuen im Modell mit zufälligen Achsenabschnitt für die 'Beat-the-Blues'-Daten

.

Im Folgenden betrachten wir ein Beispiel für wiederholte Messungen, bei dem die Zeit keine Rolle spielt.

**Beispiel 2.3.2** *(Faraway (2006), Crowder und Hand (1990))*
*Die Datei Vision.txt enthält die Daten eines Vesuchs zu Sehstärke von sieben Probanden. Die Zielgröße ist die Zeitdifferenz in Millisekunden zwischen einem Lichtblitz und dem Response im Cortex des Auges. Jedes Auge wird dabei mit vier verschiedenen Brillenglasstärken getestet. Ein Objekt in der Distanz der zweiten Zahl erscheint als wäre es in der Distanz der ersten Zahl.*

Lesen wir den Datensatz ein und schauen uns die ersten 10 Beobachtungen an.

```
> vision <- read.table("Vision.txt", header=TRUE)
```

```
> vision$power <- as.factor(vision$power)
> vision[1:10,]
   acuity power   eye subject
1     116   6/6  left       1
2     119  6/18  left       1
3     116  6/36  left       1
4     124  6/60  left       1
5     120   6/6 right       1
6     117  6/18 right       1
7     114  6/36 right       1
8     122  6/60 right       1
9     110   6/6  left       2
10    110  6/18  left       2
```

Um einen ersten Eindruck von den Daten zu bekommen, betrachten wir die Abbildung 2.5. Auffällig ist Proband 6, der einen großen Unterschied zwischen den beiden Augen aufweist. Außerdem könnte die dritte Messung für das linke Auge nicht korrekt sein.

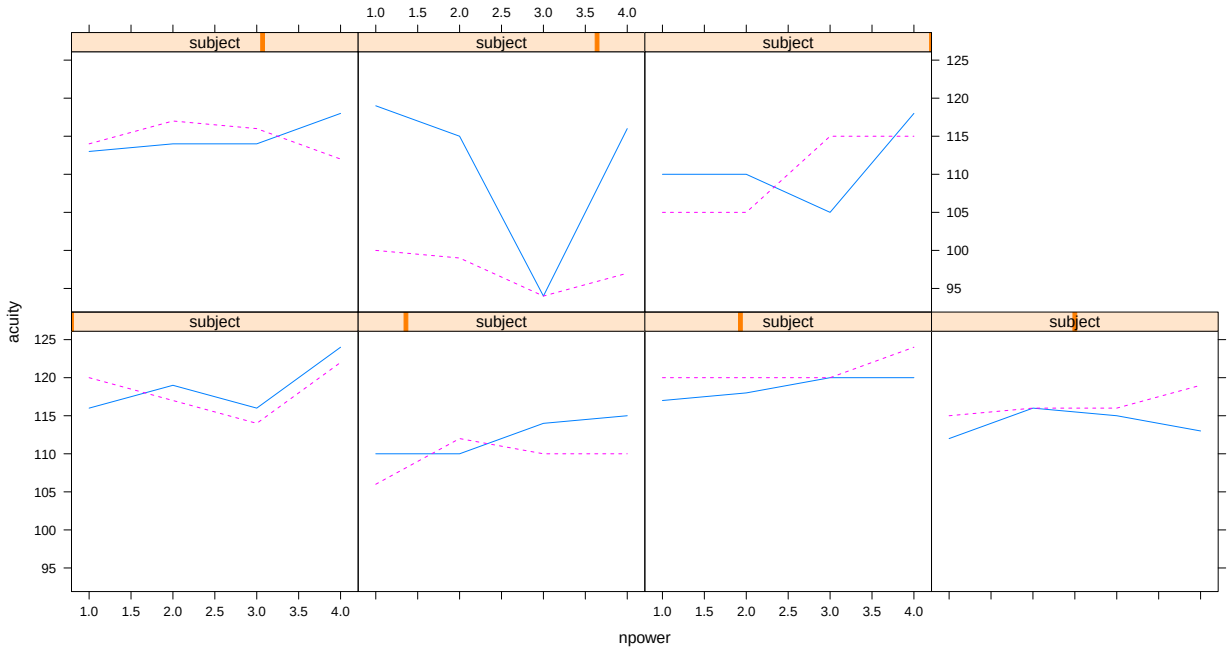


Abbildung 2.5: Sehstärkeprofile. Die durchgezogene Linie ist für das linke Auge, die gestrichelte für das rechte Auge.

Die Abbildung 2.5 wurde durch den folgenden R Code erzeugt:

```
library(lattice)
```

```
vision$npower <- rep(1:4, 14)
xyplot(acuity ~ npower|subject, data=vision, type="l", groups=eye,
lty=1:2, layout=c(4,2))
```

Wir müssen nun die Daten modellieren. Die Brillenglasstärken (`power`) sind ein fester Effekt. Die Probanden (`subject`) sollten als zufällige Effekte behandelt werden. Da wir nicht glauben, dass es eine konsistente Differenz zwischen dem rechten und dem linken Auge gibt, sollten wir den Faktor Auge hierarchisch innerhalb der Probanden modellieren. Wir betrachten also das folgende hierarchische Modell

$$y_{ijk} = \mu + p_j + s_i + e_{ik} + \epsilon_{ijk}, \quad i = 1, \ldots, 7 \text{ (subject)}, \ j = 1, \ldots 4 \text{ (power)}, \ k = 1, 2 \text{ (eye)}.$$

Der Effekt $p_j$ ist fest, die anderen Terme sind zufällige mit $s_i \sim \mathcal{N}(0, \sigma_s^2)$, $e_{ik} \sim \mathcal{N}(0, \sigma_e^2)$ und $\epsilon_{ijk} \sim \mathcal{N}(0, \sigma_\epsilon^2)$.

Die Analyse mit der Funktion *lmer* sieht wie folgt aus.

```
> mod <- lmer(acuity ~ 1 + power + (1|subject) + (1|subject:eye), data=vision)
> summary(mod)
...
Random effects:
 Groups      Name        Variance Std.Dev.
 subject:eye (Intercept) 10.27    3.205
 subject     (Intercept) 21.53    4.640
 Residual                16.60    4.075
Number of obs: 56, groups:  subject:eye, 14; subject, 7

Fixed effects:
            Estimate Std. Error t value
(Intercept) 113.4286     2.2349   50.75
power6/36    -1.7857     1.5400   -1.16
power6/6     -0.7857     1.5400   -0.51
power6/60     2.5000     1.5400    1.62
...
```

Bei den festen Effekten ist der `(Intercept)` der Schätzer für `power` von 6/18 und die anderen festen Effekteschätzer sind jeweils die Schätzer für die Differenz zu der Kategorie 6/18.

# Kapitel 3

# Gemischte Verallgemeinerte Lineare Modelle

Wir betrachten zunächst einige allgemeine Aussagen für Gemischte Verallgemeinerte Lineare Modelle. Sei $y$ der beobachtbare Zufallsvektor und $u$ der Vektor der nicht-beobachtbaren zufälligen Effekte. Es wird typischerweise angenommen, dass der Zufallsvektor $y$ aus bedingt unabhängigen Elementen besteht, die jeweils aus einer Verteilungen stammen, deren Dichte zur Exponentialfamilie gehört, z.B. Normalverteilung, Binomialverteilung oder Poisson-Verteilung.

Modellstruktur:

$$y_i|u \sim f_{Y_i|u}(y_i|u) \text{ unabhängig }, i = 1, \ldots, n$$

$$f_{Y_i|u}(y_i|u) = \exp\left\{\left[y_i\gamma_i - b(\gamma_i)\right]/\tau^2 - c(y_i, \tau)\right\}$$

Der bedingte Erwartungswert von $y_i$ steht in Beziehung zu $b(\gamma_i)$ durch $\mu_i = \partial b(\gamma_i)/\partial\gamma_i$. Wir benutzen dann eine Transformation dieses Erwartungswerts, so dass wir ein lineares Modell in festen und zufälligen Effekten erhalten:

$$\mathrm{E}[y_i|u] = \mu_i$$

$$g(\mu_i) = x_i^T\beta + z_i^T u.$$

Die Funktion $g(\cdot)$ ist bekannt und heißt *link* Funktion. Für die Normalverteilung ist die identische Abbildung die *link* Funktion, für die Binomial-Verteilung wird häufig die logit-Funktion als *link*-Funktion verwendet (logistische Regression), für Poisson- und Negative Binomial-Verteilung ist der Logarithmus die *link*-Funktion. $x_i^T$ ist der $i$-te Zeilenvektor der Modellmatrix für die festen Effekte und $\beta$ der Parametervektor der festen Effekte. $z_i^T$ ist der $i$-te Zeilenvektor der Modellmatrix für die zufälligen Effekte und $u$ der Vektor der zufälligen Effekte. Um das Modell vollständig zu spezifizieren, benötigen wir eine Verteilung der zufälligen Effekte:

$$u \sim f_U(u).$$

Wir haben bisher die bedingte Verteilung von $y$ spezifiziert. Wie sieht nun die marginale Verteilung von $y$ aus? Für den Erwartungswert gilt

$$\mathrm{E}[y_i] = \mathrm{E}[\mathrm{E}[y_i|u]] = \mathrm{E}[\mu_i] = \mathrm{E}[g^{-1}(x_i^T\beta + z_i^T u)]$$

Dieser Erwartungswert kann im Allgemeinen nicht vereinfacht werden.

Zur Illustration betrachten wir den Logarithmus als *link* Funktion, d. h. $g(\mu) = \log(\mu)$ und $g^{-1}(x) = \exp(x)$. Dann gilt

$$\mathrm{E}[y_i] = \mathrm{E}[\exp(x_i^T\beta + z_i^T u)] = \exp(x_i^T\beta)\mathrm{E}[\exp(z_i^T u))] = \exp(x_i^T\beta)M_u(z_i).$$

Dabei ist $M_u(z_i)$ die momentenerzeugende Funktion von $u$ an der Stelle $z_i$.

Nehmen wir weiterhin an, dass $u_i \sim \mathcal{N}(0, \sigma_u^2)$ und jede Zeile von $Z$ hat einen einzigen Eintrag 1 und der Rest sind Nullen. Dann gilt $M_u(z_i) = \exp(\sigma_u^2)$ und

$$\mathrm{E}[y_i] = \exp(x_i^T\beta)\exp(\sigma_u^2/2)$$

bzw.

$$\log \mathrm{E}[y_i] = x_i^T\beta + \sigma_u^2/2.$$

Für Varianzen, Kovarianzen und Korrelationen, siehe McCulloch und Searle (2001).

Die Likelihood-Funktion ist dann gegeben durch

$$L = \int \prod_i f_{y_i|u}(y_i|u) f_U(u) d_u,$$

wobei die zufälligen Effekte 'rausintegriert' werden. Die ML-Schätzer müssen in der Regel nummerisch bestimmt werden.

Alternativ können auch 'conditional' ML-Schätzer berechnet werden. Die Nutzung von marginalen und bedingten Modell diskutieren wir noch später.

Eine Alternative zur ML-Schätzung bietet der Ansatz der *generalized estimating equations* (GEEs). Dabei wird der Erwartungswert im marginalen verallgemeinerten linearen Modell als Funktion der festen Effekte dargestellt. Zum Beispiel für die logistische Regression:

$$\mathrm{logit}(\mathrm{E}[y]) = X\beta.$$

Wenn wir die Arbeitshypothese der Unabhängigkeit aller Elemente in $y$ haben, dann lautet die ML-Schätzgleichung für $\beta$

$$X^T y = X^T \mathrm{E}[y].$$

Dies ist eine unverzerrte Schätzgleichung, denn $\mathrm{E}(X^T y - X^T \mathrm{E}(y)) = 0$. Unter gewissen Regularitätsbedingungen ergeben die Lösungen der Schätzgleichung konsistente Schätzer.

Für Longitudinaldaten mit $m$ Individuen lautet die Schätzgleichung bei binären Daten

$$\sum_{i=1}^{m} X_i \mathbf{y_i} = \sum_{i=1}^{m} X_i \mathrm{E}[\mathbf{y_i}],$$

mit $X_i$ der Modellmatrix der festen Effekte für das $i$-te Individuum und $\mathbf{y_i}$ der Beobachtungsvektor des $i$-ten Individuums. Die asymptotische Varianz der Lösung $\hat{\beta}$ ist gegeben durch

$$\mathrm{Var}_\infty(\hat{\beta}) = \left(\sum_i X_i^T X_i\right)^{-1} \left(\sum_i X_i^T \mathrm{Cov}(y_i) X_i\right) \left(\sum_i X_i^T X_i\right)^{-1}$$

Diese Kovarianzmatrix kann konsistent geschätzt werden durch

$$\widehat{\mathrm{Var}}_\infty(\hat{\beta}) = \left(\sum_i X_i^T X_i\right)^{-1} \left(\sum_i X_i^T (y_i - \widehat{\mathrm{E}}[\mathbf{y_i}])(y_i - \widehat{\mathrm{E}}[\mathbf{y_i}])^T X_i\right) \left(\sum_i X_i^T X_i\right)^{-1}$$

Die Annahme der Unabhängigkeit kann zu ineffizienten Schätzern führen und andere 'working' Kovarianzmatrizen können eingebracht werden. Die GEEs lauten dann

$$\sum_i X_i W_i y_i = \sum_i X_i W_i \mathrm{E}[\mathbf{y_i}]$$

mit $W_i^{-1} = \mathrm{Cov}(\mathbf{y_i})$ der 'working' Kovarianzmatrix für $\mathbf{y_i}$. Die Varianzformeln müssen dann entsprechend angepasst werden, siehe Diggle et al. (1994).

Für die 'working' Kovarianzmatrix stehen in der Regel folgende Typen zur Verfügung:

1. **Independence**: Die 'working' Kovarianzmatrix ist die Einheitsmatrix. Die wiederholten Messungen werden als unabhängig angenommen.

2. **Exchangeable**: Die 'working' Kovarianzmatrix wird auch als *compound symmetry* bezeichnet. Hier ist die Korrelation zwischen zwei Messungen an einem Individuum immer gleich, d. h. $\mathrm{Kor}(y_{ij}, y_{ik}) = \rho$.

3. **AR(1)**: Autoregressive Korrelationsmatrix mit $\mathrm{Kor}(y_{ij}, y_{ik}) = \rho^{|k-j|}, j \neq k$. Zeitlich näher beieinanderliegende Messungen sind stärker korreliert als zeitlich weiter auseinanderliegende Messungen.

4. **Unstructured**: Korrelationsmatrix mit $k(k-1)/2$ Parameter, wobei $k$ die Anzahl der wiederholten Messungen ist und $\mathrm{Kor}(y_{ij}, y_{ik}) = \rho_{jk}$. Es ist keine Struktur vorgegeben. Die Messungen können beliebig korreliert sein.

Um den Unterschied zwischen der Modellierung im marginalen und im bedingten Modell darzustellen, nutzen wir ein Beispiel aus McCulloch und Searle (2001). Sei $y_{ij} = 1$, falls das $j$-te Kind einer Frau $i$ frühgeboren wurde und $y_{ij} = 0$ sonst, und nehmen wir an, dass es eine erklärende Variable

$x_{ij}$ = 'Anzahl der alkoholischen Drinks pro Tag' gibt. Im marginalen Modell wird der marginale Erwartungswert von $y_{ij}$ direkt, z.B. durch logistische Regression, an die Daten angepasst:

$$\text{logit}(\text{E}[y_{ij}]) = \text{logit}(\text{P}(y_{ij} = 1)) = \alpha + \beta x_{ij} \ .$$

Wir modellieren hierbei den *Logit* für die Wahrscheinlichkeit einer Frühgeburt für eine Population von Frauen. Wenn wir die Korrelation bei den Frauen berücksichtigen müssen, können wir einen GEE Ansatz nutzen.

Der bedingte Ansatz dagegen berücksichtigt einen Zufallseffekt für die Frauen und spezifiziert ein bedingtes Modell derart, dass gilt

$$\text{logit}(\text{E}[y_{ij}|u]) = \text{logit}(\text{P}(y_{ij} = 1)) = \alpha + \beta x_{ij} + u_i$$

mit $u_i$ dem zufälligen Fraueneffekt. Dies entspricht der Modellierung einer bedingten Wahrscheinlichkeit einer Frühgeburt für jede Frau separat.

Wenn die Frage in dem Beispiel ist, inwieweit die Inzidenz einer Frühgeburt verringert werden kann, wenn der durchschnittliche Alkoholkonsum einer Frau gesenkt wird, so ist das marginale Modell das adäquate Modell. Wenn man jedoch an der Frage interessiert ist, wie der Alkoholkonsum die individuelle Physiologie der Frauen beeinflusst, so ist das bedingte Modell das geeignete Modell.

In den beiden folgenden Abschnitten betrachten wir die Analyse von Binärdaten und von Zähldaten.

## 3.1 Binäre Daten

**Beispiel 3.1.1** *(Everitt, Hothorn (2010), Davis (1991))*
*Die Datei Respiratory.txt enthält die Daten eines klinischen Versuches, der zwei Medikamente zur Behandlung einer Atemwegserkrankung untersucht. In jedem der zwei beteiligten Zentren wurden die Patienten zufällig der aktiven Behandlung oder Placebo zugewiesen. Während des Studienverlaufs wurde der Atemwegszustand (in den Kategorien poor oder good) bei vier monatlichen Visiten erhoben. Insgesamt umfasst der Datensatz 111 Patienten. Die Fragestellung ist nun, ob die aktive Behandlung effektiv ist und wie groß dieser Effekt ist.*

Wir lesen den Datensatz ein und schauen uns die ersten Daten an.

```
> setwd("C:/Users/knapp/Documents/Guido/GemischteModelle/Datensätze")
> respiratory <- read.table("Respiratory.txt", header=TRUE)
> head(respiratory)
    centre treatment gender age status month subject
1        1   placebo female  46   poor     0       1
112      1   placebo female  46   poor     1       1
223      1   placebo female  46   poor     2       1
334      1   placebo female  46   poor     3       1
445      1   placebo female  46   poor     4       1
2        1   placebo female  28   poor     0       2
```

Im Folgenden soll der Baselinewert auch als erklärende Variable zur Verfügung stehen. Dazu reduzieren wir den Datensatz um die Beobachtungen zum Zeitpunkt 0 und hängen an diesen neuen Datensatz die Information über den Baselinewert an. Bei der *gee()* Funktion aus dem R Paket `gee` muss die Zielvariable 1 und 0 kodiert sein. Deshalb wird die Dummy-Kodierung für die Variable `status` durchgeführt. Die Funktion *glm()* aus dem R Paket `stats` kann mit der Kodierung *poor* und *good* umgehen. Für die Funktion *glmmadmb* aus dem R Paket `glmmADMB` müssen die zufälligen Effekte Faktorvariablen sein, so dass dies hier für die Variable `subject` geändert wird.

```
# Baselinewert soll erklärende Variable werden
resp <- subset(respiratory, month > "0")
resp$baseline <- rep(subset(respiratory, month == "0")$status, rep(4,111))
# Dummy-Variable für status
resp$nstat <- as.numeric(resp$status == "good")
resp$subject <- as.factor(resp$subject)
```

Zunächst betrachten wir die logistische Regression mit der Unabhängigkeitsannahme. Dies können wir mit der Funktion *glm()* durchführen. Die Formeldarstellung ist für die festen Effekte die gewohnte. Damit eine logistische Regression gerechnet wird, muss family = `"binomial"` angegeben werden. Das Ergebnis der Analyse ist unten zu sehen. Bis auf das Geschlecht haben alle erklärenden Variablen einen Einfluss, insbesondere ist der Behandlungseffekt signifikant.

```
>
> resp_glm <- glm(status ~ centre + treatment + gender + baseline + age,
+                 data = resp, family = "binomial")
> summary(resp_glm)
...
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.310256   0.452642  -0.685  0.49307
centre             -0.671601   0.239567  -2.803  0.00506 **
treatmenttreatment -1.299216   0.236841  -5.486 4.12e-08 ***
gendermale         -0.119244   0.294671  -0.405  0.68572
baselinepoor        1.882029   0.241290   7.800 6.20e-15 ***
age                 0.018166   0.008864   2.049  0.04043 *
...
```

Mit der *gee()* Funktion aus dem R Paket `gee` können nun auch marginale Modelle mit verschiedenen Korrelationsstrukturen angepasst werden. Dazu muss die Option `corstr=` gesetzt werden. Wird wie im folgenden zunächst als Korrelationsstruktur die Unabhängigkeit benutzt, so erhalten wir dieselben Schätzer wie oben nur mit umgekehrtem Vorzeichen, weil jetzt die Analyse für die Wahrscheinlichkeit, dass der Zustand 1 (also *good*) durchgeführt wird. Oben wird die Analyse für die Wahrscheinlichkeit, dass der Status *poor* ist, durchgeführt.

```
> library(gee)
```

```
> resp_gee1 <- gee(nstat ~ centre + treatment + gender + baseline + age,
+                 data = resp, family = "binomial", id = subject,
+                 corstr = "independence", scale.fix= TRUE, scale.value = 1)
> summary(resp_gee1)
...
Model:
 Link:                      Logit
 Variance to Mean Relation: Binomial
 Correlation Structure:     Independent
...
Coefficients:
                     Estimate  Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept)         0.31025629 0.452642507  0.6854334  0.65618360  0.4728193
centre              0.67160098 0.239566599  2.8033999  0.35681913  1.8821889
treatmenttreatment  1.29921589 0.236841017  5.4856034  0.35077797  3.7038127
gendermale          0.11924365 0.294671045  0.4046670  0.44320235  0.2690501
baselinepoor       -1.88202860 0.241290221 -7.7998545  0.35005152 -5.3764332
age                -0.01816588 0.008864403 -2.0493061  0.01300426 -1.3969169
...
```

Bei der GEE-Analyse werden zwei Typen von Standardfehlern ausgegeben, die 'naiven' und die robusten. Die 'naiven' Standardfehler und 'naiven' z-Werte sind die modellbasierten Schätzwerte wie sie auch oben in der *glm* Funktion zu sehen sind. Die 'robusten' Standardfehler basieren auf dem sog. Sandwich-Schätzer, der datenbasiert ist. Die robusten Standardfehler sind viel größer als die 'naiven' Standardfehler. Dies deutet daraufhin, dass die Annahme der Unabhängigkeit nicht gerechtfertigt ist.

Daher wird in der folgenden Auswertung die *compound symmetry* oder *exchangeable* Korrelationsstruktur betrachtet. Die Analyse zeigt, dass die 'naiven' Standardfehler viel näher an den robusten Standardfehler liegen und die betrachtete Korrelationsstruktur die wahre Korrelationsstruktur besser widerspiegelt als die Unabhängigkeitsannahme.

```
> resp_gee2 <- gee(nstat ~ centre + treatment + gender + baseline + age,
+                 data = resp, family = "binomial", id = subject,
+                 corstr = "exchangeable", scale.fix= TRUE, scale.value = 1)
> summary(resp_gee2)
...
Model:
 Link:                      Logit
 Variance to Mean Relation: Binomial
 Correlation Structure:     Exchangeable
...
Coefficients:
                    Estimate Naive S.E.    Naive z Robust S.E.   Robust z
```

```
(Intercept)          0.31025629  0.6414066  0.4837124  0.65618360  0.4728193
centre               0.67160098  0.3394723  1.9783676  0.35681913  1.8821889
treatmenttreatment   1.29921589  0.3356101  3.8712064  0.35077797  3.7038127
gendermale           0.11924365  0.4175568  0.2855747  0.44320235  0.2690501
baselinepoor        -1.88202860  0.3419147 -5.5043802  0.35005152 -5.3764332
age                 -0.01816588  0.0125611 -1.4462014  0.01300426 -1.3969169
...
```

Der geschätzte Behandlungseffekt auf dem GEE Ansatz mit der *exchangeable* Korrelationsstruktur ist 1.299 und das zugehörige 95%-Konfidenzintervall mit den robusten Standardfehlern lautet

```
> se <- summary(resp_gee2)$coefficients["treatmenttreatment", "Robust S.E."]
> # Konfidenzintervall mit robustem Standardfehler
> coef(resp_gee2)["treatmenttreatment"] + c(-1,1) * se * qnorm(0.975)
[1] 0.6117037 1.9867281
```

Diese Werte sind auf der log-Odds Skala gegeben. Die Interpretation der Ergebnisse wird einfacher, wenn wir die Ergebnisse auf die Odds-Skala zurücktransformieren. Der geschätzte Behandlungseffekt ist dann $\exp(1.299) = 3.666$ mit zugehörigem 95%-Konfidenzintervall von

```
> # Als odds ausgedrückt
> exp(coef(resp_gee2)["treatmenttreatment"] + c(-1,1) * se * qnorm(0.975))
[1] 1.843570 7.291637
```

Die Chance, einen guten Atemwegszustand mit der Behandlung zu erreichen, ist um das zweifache bis siebenfache größer als die Chance, einen guten Atemwegszustand mit Placebo zu erreichen.

Alternativ können wir auch die Funktion *glmer()* aus dem R Paket `lme4` nutzen. Bei der Bestimmung der ML-Schätzer müssen nummerische Methoden angewendet werden. Dafür steht die adaptive Gauß-Hermite Quadratur zur Approximation der log-Likelihood-Funktion zur Verfügung. Über die Option `nAGQ` wird die Anzahl der Punkte pro Achse für die Evaluierung der adaptiven Gauß-Hermite Approximation gesteuert. Die Default-Einstellung `nAGQ=1` entspricht der Laplace-Approximation. Werte größer als 1 erhöhen die Genauigkeit bei der Evaluierung der Approximation auf Kosten der Rechenzeit. Bei der Einstellung `nAGQ=0` wird eine penalisierte iterativ gewichtete Kleinste-Quadrate Prozedur verwendet, die jedoch weniger genau ist, dafür aber kaum Konvergenzprobleme aufweist.

Zunächst betrachten wir das Ergebnis mit der nicht so genauen Methode an.

```
> resp_lmer <- glmer(status ~ centre + treatment + gender + age + baseline
+                    + (1 | subject), family = binomial(), data = resp, nAGQ=0)
> summary(resp_lmer)
Generalized linear mixed model fit by maximum likelihood (Adaptive
  Gauss-Hermite Quadrature, nAGQ = 0) [glmerMod]
 Family: binomial  ( logit )
```

```
...
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 3.421    1.85
Number of obs: 444, groups:  subject, 111

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.29725    0.95073  -0.313  0.75454
centre             -0.80867    0.49447  -1.635  0.10196
treatmenttreatment -1.63998    0.47595  -3.446  0.00057 ***
gendermale         -0.15315    0.61591  -0.249  0.80362
age                 0.01969    0.01842   1.069  0.28521
baselinepoor        2.35235    0.48497   4.851 1.23e-06 ***
...
```

Die Analyse erfolgt für die Modellierung der Wahrscheinlichkeit, dass ein schlechter Atemwegszustand vorliegt. Um das Ergebnis mit dem des marginalen Modells mit GEE Ansatz vergleichen zu können, muss der Schätzer mit minus 1 multipliziert werden. Der Schätzwert für den Behandlungseffekt für eine guten Atemwegszustand ist 1.6370 und somit erheblich größer als der Behandlungsschätzer im marginalen Modell. Dies ist zu erwarten, weil im marginalen Modell über die Population gemittelt wird.

Wird die Laplace-Approximation verwendet, so erhöht sich der Schätzer für den Behandlungseffekt sogar auf 2.156.

```
> resp_lmer1 <- glmer(status ~ centre + treatment + gender + age + baseline
+                   + (1 | subject), family = binomial(), data = resp, nAGQ=1)
>
> summary(resp_lmer1)
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
...
Random effects:
 Groups  Name        Variance Std.Dev.
 subject (Intercept) 3.86     1.965
Number of obs: 444, groups:  subject, 111

Fixed effects:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        -0.38176    1.03810  -0.368 0.713058
centre             -1.04333    0.54607  -1.911 0.056055 .
treatmenttreatment -2.15612    0.55445  -3.889 0.000101 ***
```

```
gendermale         -0.20182    0.67055  -0.301 0.763426
age                 0.02539    0.02008   1.265 0.205946
baselinepoor        3.06824    0.60033   5.111 3.21e-07 ***
...
```

Ein weiteres R Paket, mit dem Gemischte Verallgemeinerte Lineare Modelle angepasst werden können, ist das Paket `glmmADMB`. Die Installation des R Pakets `glmmADMB` erfolgt, indem der folgende R Code in die Konsole eingegeben wird.

```
install.packages("R2admb")
install.packages("glmmADMB",
    repos=c("http://glmmadmb.r-forge.r-project.org/repos",
            getOption("repos")), type="source")
```

Die Syntax für die Funktion *glmmadmb()* in dem Paket `glmmADMB` ist der Funktion *lme* im R Paket `nlme` ähnlich, was die Formalangabe für feste und zufällige Effekte betrifft. Das obige Beispiel im bedingten Modell wird mit der Funktion *glmmadmb()* wie folgt analysiert

```
> resp_admb <- glmmadmb(nstat ~ centre + treatment + gender + age + baseline,
+           random = ~1|subject, data = resp, family = "binomial", corStruct = "diag")
> summary(resp_admb)
...
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)          0.3817     1.0391    0.37   0.7134
centre               1.0437     0.5468    1.91   0.0563 .
treatmenttreatment   2.1573     0.5559    3.88   0.0001 ***
gendermale           0.2019     0.6713    0.30   0.7636
age                 -0.0254     0.0201   -1.26   0.2065
baselinepoor        -3.0698     0.6024   -5.10  3.5e-07 ***
...
Random effect variance(s):
Group=subject
            Variance StdDev
(Intercept)    3.865  1.966
...
```

Das Ergebnis hier entspricht im Wesentlichen dem Ergebnis mit der *glmer* Funktion für die Laplace-Approximation. Die Option `corStruct = "diag"` ist hier so zu verstehen, dass die zufälligen Effekte von Patient zu Patient unkorreliert sind.

## 3.2 Zähldaten

**Beispiel 3.2.1** *(Everitt, Hothorn (2010), Thall und Vail (1990))*
*In einer klinischen Studie wurden 59 Patienten, die an Epilepsie leiden, entweder der aktiven Behandlungsgruppe (antiepileptisches Medikament) oder Placebo zufällig zugewiesen. Die Anzahl epileptischer Anfälle in vier verschiedenen Zwei-Wochen-Perioden wurden für jeden Patienten nach der Randomisierung aufgezeichnet. Darüber hinaus wurde die Anzahl epileptischer Anfälle 8 Wochen vor der Randomisierung erhoben, die als Baselinewert mit in die Analyse einfließen soll. Die Fragestellung ist, ob die Einnahme des antiepileptischen Medikaments die Anzahl epileptischer Anfälle im Vergleich zu Placebo senkt. Die Daten sind in der Datei Epilepsy.txt gegeben.*

Wir lesen den Datensatz ein, schauen uns die Struktur des Datensatzes sowie die ersten Beobachtungen an. Wir sehen, dass die Daten in 'langer' Form gegeben sind, d.h. pro Person und Periode gibt es eine Zeile in dem *data.frame*.

```
> epilepsy <- read.table("Epilepsy.txt", header=TRUE)
> str(epilepsy)
'data.frame':   236 obs. of  6 variables:
 $ treatment   : Factor w/ 2 levels "placebo","Progabide": 1 1 1 1 1 1 1 1 1 1 ...
 $ base        : int  11 11 11 11 11 11 11 11 6 6 ...
 $ age         : int  31 31 31 31 30 30 30 30 25 25 ...
 $ seizure.rate: int  5 3 3 3 3 5 3 3 2 4 ...
 $ period      : int  1 2 3 4 1 2 3 4 1 2 ...
 $ subject     : int  1 1 1 1 2 2 2 2 3 3 ...
> head(epilepsy)
    treatment base age seizure.rate period subject
1     placebo   11  31            5      1       1
110   placebo   11  31            3      2       1
112   placebo   11  31            3      3       1
114   placebo   11  31            3      4       1
2     placebo   11  30            3      1       2
210   placebo   11  30            5      2       2
```

Die Anzahl epileptischer Anfälle sind Zähldaten und es gibt zwei Verteilungen, die zur Modellierung von Zähldaten genutzt werden können: Poisson-Verteilung und Negative Binomialverteilung. Die Poisson-Verteilung hat die restriktive Eigenschaft, dass Erwartungswert und Varianz identisch sind. Bei der Negativen Binomialverteilung ist die Varianz um den Faktor $1/p$ größer als der Erwartungswert der Verteilung, wobei $p$ die Wahrscheinlichkeit für das Auftreten des interessierenden Ereignisses ist.

Schauen wir uns in dem Datensatz die arithmetischen Mittel und empirischen Varianzen für jede Behandlung zu jedem Zeitpunkt an. Dies kann mit dem folgenden R Code geschehen.

```
> itp <- interaction(epilepsy$treatment, epilepsy$period)
```

```
> tapply(epilepsy$seizure.rate, itp, mean)
  placebo.1 Progabide.1   placebo.2 Progabide.2   placebo.3 Progabide.3
   9.357143    8.580645    8.285714    8.419355    8.785714    8.129032
  placebo.4 Progabide.4
   7.964286    6.709677
> tapply(epilepsy$seizure.rate, itp, var)
  placebo.1 Progabide.1   placebo.2 Progabide.2   placebo.3 Progabide.3
  102.75661   332.71828    66.65608   140.65161   215.28571   193.04946
  placebo.4 Progabide.4
   58.18386   126.87957
```

Wir sehen, dass die empirischen Varianzen zum Teil erheblich größer als die zugehörigen arithmetischen Mittel sind.

Zunächst betrachten wir die Analyse im Generalisierten Linearen Modell mit der Funktion *glm()*. Die Beobachtungsphase in unserem Beispiel beträgt 2 Wochen, so dass für jede Beobachtung der Offset log(2) gesetzt werden kann. Dieser Offset wird in die Modellformel als erklärende Variable aufgenommen und als Offset gekennzeichnet. Bei der Schätzung der festen Effekte wird der zugehörigen Parameter für den Offset fest auf 1 gesetzt. In dem folgenden R Code wird die Variable `treatment` kurz in `trt` umbenannt. Die log-lineare Poisson-Regression wird berechnet, indem `family = "poisson"` gesetzt wird.

```
> # Offset log(2)
> per <- rep(log(2), nrow(epilepsy))
> epilepsy$period <- as.numeric(epilepsy$period)
> names(epilepsy)[names(epilepsy) == "treatment"] <- "trt"
> fm <- seizure.rate ~ base + age + trt + offset(per)
> epilepsy_glm <- glm(fm, data=epilepsy, family = "poisson")
> summary(epilepsy_glm)
...
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1306156  0.1356191  -0.963   0.3355
base          0.0226517  0.0005093  44.476  < 2e-16 ***
age           0.0227401  0.0040240   5.651 1.59e-08 ***
trtProgabide -0.1527009  0.0478051  -3.194   0.0014 **
...
(Dispersion parameter for poisson family taken to be 1)
...
```

Unter der Unabhängigkeitsannahme ergibt sich ein signifikanter Schätzer für den Behandlungseffekt von $-0.153$ mit $p = 0.0014$. Der Schätzer ist negativ, d.h. die aktive Behandlung führt zu signifkant weniger epileptischen Anfällen im Zwei-Wochen-Zeitraum als Placebo. Ist diese Aussage korrekt?

In diesem Beispiel scheint die Annahme der Poisson-Verteilung für die Verteilung der Anzahl epileptischer Anfälle nicht gerechtfertigt zu sein, weil die empirischen Varianzen größer als die entsprechenden empirischen Mittel sind. Anders ausgedrückt bedeutet dies, dass die empirischen Varianzen größer als die zu erwartenden Varianzen der Poisson-Verteilungen sind, wenn die Parameter der Poisson-Verteilungen durch die jeweiligen arithmetischen Mittel geschätzt werden. Diese Datensituation nennt man 'Overdispersion'. Das Ignorieren dieser Overdispersion führt zu zu kleinen Standardfehlern. Sei $V(\mu)$ die Varianz der Poisson-Verteilung, so lässt sich ein Overdispersionsparameter $\phi$ in das Generalisierte Lineare Modell mit aufnehmen, so dass die Varianz als $\phi V(\mu)$ modelliert wird. In der Funktion *glm* kann man Overdispersion berücksichtigen, indem `family = "quasipoisson"` gesetzt wird.

```
> # Overdispersion in GLM
> epilepsy_glm1 <- glm(fm, data=epilepsy, family = "quasipoisson")
> summary(epilepsy_glm1)
...
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.130616   0.305959  -0.427   0.6698
base          0.022652   0.001149  19.714   <2e-16 ***
age           0.022740   0.009078   2.505   0.0129 *
trtProgabide -0.152701   0.107849  -1.416   0.1582
...
(Dispersion parameter for quasipoisson family taken to be 5.08961)
d{verbatim}
```

Der Dispersionsparameter wird auf 5.09 geschätzt und die Standardfehler sind um den Faktor $\sqrt{5.09} = 2.26$ größer als im Modell ohne Overdispersion. Die Schätzung der festen Effekte ändert sich nicht. Aufgrund des größeren Standardfehlers ist die aktive Behandlung jedoch nicht mehr signifikant.

Alternativ können die Anzahlen der epileptischen Anfälle auch durch die Negative Binomialverteilung modelliert werden. Dies geschieht mit der Funktion *glm.nb()* aus dem R Paket `MASS`.

```
> library(MASS)
> # Negative Binomial
> epilepsy_glm_nb <- glm.nb(fm, data=epilepsy)
> summary(epilepsy_glm_nb)
...
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.152338   0.270606  -0.563   0.5735
base          0.026875   0.001760  15.269   <2e-16 ***
age           0.018109   0.008249   2.195   0.0281 *
trtProgabide -0.188918   0.101701  -1.858   0.0632 .
```

```
...
(Dispersion parameter for Negative Binomial(2.4035) family taken to be 1)
...
```

Mit der Modellierung durch die Negative Binomialverteilung ergibt sich ein absolut größerer Schätzer für den Behandlungseffekt. Der Standardfehler für den Behandlungseffekt ist in etwa genauso groß wie der Standardfehler für den Behandlungseffekt im Poisson-Modell mit Overdispersion. Der Behandlungseffekt im Modell mit Negativer Binomialverteilung ist zum 5%-Niveau nicht signifikant.

Bisher haben wir nur die Overdispersion im Blick gehabt und nicht die Korrelationsstruktur innerhalb der Patienten. Die Berücksichtigung der Korrelationsstruktur schauen wir uns nur im marginalen Modell an. Die Berücksichtigung der Korrelationsstruktur im bedingten Modell erfolgt mit der Funktion *glmer* aus dem R Paket `lme4` wie bei den binären Daten.

Wir betrachten zunächst die Unabhängigkeitsstruktur und setzen `scale.fix = TRUE`, d.h. modellieren keinen Dispersionsparameter, sondern setzen ihn mit der nachfolgenden Option `scale.value` fest auf 1.

```
> epilepsy_gee1 <- gee(fm, data=epilepsy, family = "poisson", id=subject,
>                      corstr = "independence", scale.fix = TRUE, scale.value = 1)
> summary(epilepsy_gee1)
...
Model:
 Link:                      Logarithm
 Variance to Mean Relation: Poisson
 Correlation Structure:     Independent
...
Coefficients:
                Estimate   Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept)  -0.13061561 0.1356191185 -0.9631062 0.365148155 -0.3577058
base          0.02265174 0.0005093011 44.4761250 0.001235664 18.3316325
age           0.02274013 0.0040239970  5.6511312 0.011580405  1.9636736
trtProgabide -0.15270095 0.0478051054 -3.1942393 0.171108915 -0.8924196
...
```

Das Ergebnis für die festen Effekte und die 'naiven' Standardfehler stimmen mit den Ergebnissen der obigen Poisson-Regression ohne Overdispersion überein. Die 'naiven' Standardfehler sind jedoch viel kleiner als die robusten Standardfehler, so dass von keiner guten Modellanpassung ausgegangen werden kann.

Die Annahme der Unabhängigkeitsstruktur wird im folgenden durch die 'Exchangeable'-Korrelationstruktur ersetzt; Overdispersion wird weiterhin nicht berücksichtigt.

```
> epilepsy_gee2 <- gee(fm, data=epilepsy, family = "poisson", id=subject,
>                      corstr = "exchangeable", scale.fix = TRUE, scale.value = 1)
```

```
> summary(epilepsy_gee2)
...
Model:
 Link:                      Logarithm
 Variance to Mean Relation: Poisson
 Correlation Structure:     Exchangeable
...
Coefficients:
                Estimate   Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept)  -0.13061561 0.2004416507 -0.6516391 0.365148155 -0.3577058
base          0.02265174 0.0007527342 30.0926122 0.001235664 18.3316325
age           0.02274013 0.0059473665  3.8235638 0.011580405  1.9636736
trtProgabide -0.15270095 0.0706547450 -2.1612270 0.171108915 -0.8924196
...
```

Mit diesen Ansatz erhalten wir, bei unveränderten Schätzern für die festen Effekte, 'naive' Standardfehler, die nun näher an den robusten Standardfehlern sind, jedoch immer noch erheblich von diesen entfernt.

Schießlich können wir auch noch die Overdispersion berücksichtigen, indem wir `scale.fix = FALSE` setzen. Der Wert für `scale.value` wird dann ignoriert.

```
> epilepsy_gee3 <- gee(fm, data=epilepsy, family = "poisson", id=subject,
                       corstr = "exchangeable", scale.fix = FALSE, scale.value = 1)
> summary(epilepsy_gee3)
...
Coefficients:
                Estimate Naive S.E.    Naive z Robust S.E.   Robust z
(Intercept)  -0.13061561 0.45219954 -0.2888451 0.365148155 -0.3577058
base          0.02265174 0.00169818 13.3388301 0.001235664 18.3316325
age           0.02274013 0.01341735  1.6948302 0.011580405  1.9636736
trtProgabide -0.15270095 0.15939823 -0.9579840 0.171108915 -0.8924196

Estimated Scale Parameter:  5.089608
...
```

Die 'naiven' Standardfehler sind nun ähnlich zu den robusten Standardfehlern. Es ist offensichtlich, dass es keinen Behandlungseffekt gibt.

Abschließend wollen wir noch einen weiteren Aspekt der Modellierung bei Zähldaten ansprechen, die 'zero-inflated' Poisson-Regression im bedingten Modell. Bei diesem Modell wird berücksichtigt, dass die Wahrscheinlichkeit, Nullen zu beobachten, größer ist als die unter der entsprechenden Poisson-Verteilung zu erwartende. Dies wird durch einen Parameter $\pi$ erreicht, der die Wahrscheinlichkeit angibt, wie viele Nullen mehr erwartet werden. Eine 'zero-inflated' Poisson-Regression kann mit

der Funktion *glmmadmb()* im R Paket `glmmADBM` durchgeführt werden. Die Option `zeroInflation` muss dabei auf `TRUE` gesetzt werden.

```
> epilepsy$subject <- as.factor(epilepsy$subject)
> epilepsy_admb <- glmmadmb(fm, random = ~1|subject, data = epilepsy,
+                  family = "poisson", corStruct = "diag", zeroInflation=TRUE)
> summary(epilepsy_admb)
...
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.0335     0.3825   -0.09    0.930
base           0.0261     0.0026   10.02   <2e-16 ***
age            0.0124     0.0118    1.06    0.291
trtProgabide  -0.2562     0.1445   -1.77    0.076 .
...
Random effect variance(s):
Group=subject
            Variance StdDev
(Intercept)   0.2367 0.4865

Zero-inflation: 0.045915  (std. err.:  0.018475 )
...
```

Die Schätzer für die Anzahl zusätzlicher Nullen beträgt 0.05. Auch in diesem Modell ist der Behandlungseffekt zum 5%-Niveau nicht signifikant.

# Literaturverzeichnis


Ahrens, H. (1967). *Varianzanalyse*. Akademie-Verlag, Berlin.

Anderson, R.L., Bancroft, T.A. (1952). *Statistical Theory ind Research*. McGraw-Hill, New York.

Crowder, M.J., Hand, D.J.(1990). *Analysis of Repeated Measures*. Chapman & Hall, London.

Davis, C.S. (1991). Semi-parametric and non-parametric methods for the analysis of repeated measurements with applications to clinical trials. *Statistics in Medicine*, 10,1959–1980.

Diggle, P., Liang, K.-Y., Zeger, S.L. (1994). *Longitudinal Data Analysis*. Oxford University Press, Oxford.

Duncan, A.J. (1974). *Quality Control and Industrial Statistics*. Richard D. Irwin, INC. Homewood.

Everitt, B.S., Hothorn, T. (2010). *A Handbook of Statistical Analyses using R*, 2nd Edition. CRC Press, Boca Raton, FL.

Faraway, J.J. (2006). *Extending the Linear Model with R*. Chapman & Hall, Boca Raton, FL.

Hartung, J., Elpelt, B. (2007). *Multivariate Statistik*, 3. Auflage. Oldenbourg, München.

Linder, A. (1969). *Planen und Auswerten von Versuchen*. Birkhäuser, Basel/Stuttgart.

Longford, N.T. (1993). *Random Coefficient Models*. Oxford University Press, Oxford, UK.

McCulloch, C., Searle, S.R. (2001). *Generalized, Linear, and Mixed Models*. Wiley, New York.

Proudfoot, J., Goldberg, D., Mann, A., Everitt, B.S., Marks, I, Gray, J.A. (2003). Computerized, interactive, multimedia cognitive-behavioural programm for anxiety and depression in general pratice. *Psychological Medicine*, 33, 217–227.

R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Searle, S.R. (1971). *Linear Models*. Wiley & Sons, New York.

Thall„P.F., Vail, S.C. (1990). Some covariance models for longitudinal count data with overdispersion. *Biometrics*, 46, 657-671.

Verbeke, G., Molenberghs, G. (1997). *Linear Mixed Models in Practice*. Springer, New York.