

**Der S-PLUS-Zyklus
WS 2003/2004 bis SS 2006:**

Einführung in S-PLUS

Lineare Modelle: Regression und Varianzanalyse

Dr. Gerrit Eichner u Dr. Thorsten Schmidt

1 Einführung in die lineare Regression

Das (klassische) **Modell der (multiplen) linearen Regression** lautet in Vektor-Matrix-Notation bekanntermaßen

$$Y = X\beta + \varepsilon.$$

Dabei ist X die feste, bekannte $(n \times p)$ -Designmatrix, β der feste, unbekannte und zu schätzende p -dimensionale Parametervektor, ε ein unbekannter, n -dimensional multivariat normalverteilter Fehlervektor mit unabhängigen Komponenten sowie Y der zufällige, n -dimensionale Beobachtungs- (oder Response-)Vektor. Etwas detaillierter:

$$Y = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, X = \begin{pmatrix} x_{10} & x_{11} & \dots & x_{1,p-1} \\ x_{20} & x_{21} & \dots & x_{2,p-1} \\ \vdots & \vdots & \dots & \vdots \\ x_{n0} & x_{n1} & \dots & x_{n,p-1} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{pmatrix} \text{ und } \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix} \sim \mathcal{N}_n(0, \sigma^2 I_n).$$

Beachte:

- Es wird von *linearer* Regression gesprochen, weil der Parameter (!) β linear in das Modell eingeht.
- Für $i = 1, \dots, n$ ist also $Y_i = x'_i \beta + \varepsilon_i$, wobei der Designvektor $x'_i := (x_{i0}, x_{i1}, \dots, x_{i,p-1})$ die i -te Zeile der Designmatrix ist. Dieser geht durch eine geeignete, bekannte und für alle i gleiche Funktion aus dem Covariablenvektor z_i hervor. Einfache Beispiele: $z_i \in \mathbb{R}$ und $x'_i = (1, z_i)$ oder $x'_i = (1, z_i, z_i^2)$.
- Somit gilt: $Y_i \sim \mathcal{N}(x'_i \beta, \sigma^2)$ unabhängig für $i = 1, \dots, n$. Sind die Designvariablen stochastisch, so werden die (Y_i, x_i) als unabhängig und identisch verteilt angenommen und die Y_i als bedingt x_i unabhängig normalverteilt mit $\mathbb{E}[Y_i | x_i] = x'_i \beta$ und $\text{Var}(Y_i) = \sigma^2$.

Zur Erinnerung und zur Referenz **einige Resultate aus der Theorie der linearen Modelle** (siehe z. B. Hocking (1996), Abschnitt 3.1.1):

Der Kleinste-Quadrate-Schätzer für β ist $\hat{\beta} := (X'X)^{-1}X'Y \sim \mathcal{N}_p(\beta, \sigma^2(X'X)^{-1})$

und die gefitteten Werte sind $\hat{Y} := X\hat{\beta} = \underbrace{X(X'X)^{-1}X'}_{=: H} Y$.

Die $(n \times n)$ -Projektionsmatrix

H ist symmetrisch, idempotent und erfüllt $H(I_n - H) = 0_{n \times n}$.

Ferner ist der Residuenvektor

$$\hat{\varepsilon} := Y - \hat{Y} = (I_n - H)Y \sim \mathcal{N}_n(0, \sigma^2(I_n - H))$$

und die Residuen-Quadratsumme

$$\text{RSS} := \hat{\varepsilon}'\hat{\varepsilon} = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

erfüllt

$$\mathbb{E}[\text{RSS}/(n - p)] = \sigma^2.$$

$\hat{\beta}$ und $\text{RSS}/(n - p)$ sind

unverzerrte, Minimum-Varianz-Schätzer für β bzw. σ^2 und stochastisch unabhängig.

Außerdem ist (wegen $H(I_n - H) = 0$)

$$\text{Cov}(\hat{Y}, \hat{\varepsilon}) = 0_{n \times n}.$$

Aufgrund der Quadratesummenzerlegung

$$\begin{aligned}
SS_{Total} &:= \sum_{i=1}^n (Y_i - \bar{Y}_n)^2 \\
&= \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 + \sum_{i=1}^n (\hat{Y}_i - \bar{Y}_n)^2 \\
&=: \text{RSS} + \text{SS}_{Regression}
\end{aligned}$$

ist der multiple R^2 -Wert

$$R^2 := \frac{SS_{Total} - \text{RSS}}{SS_{Total}} \in [0, 1]$$

ein Maß für die Güte des Fits, da er der durch das Regressionsmodell erklärte relative Anteil an der Gesamtstreuung der Y_i ist.

Der multiple Korrelationskoeffizient R ist der empirische Pearsonsche Korrelationskoeffizient der (Y_i, \hat{Y}_i) und $\hat{\beta}$ maximiert letzteren für (Y_i, \tilde{Y}_i) in der Klasse $\tilde{Y} = Xa$.

Bemerkung: Die Projektionsmatrix H heißt auch “hat matrix”, da sie aus dem Beobachtungsvektor Y durch Aufsetzen eines „Hutes“ den Vektor \hat{Y} der gefitteten Werte macht.

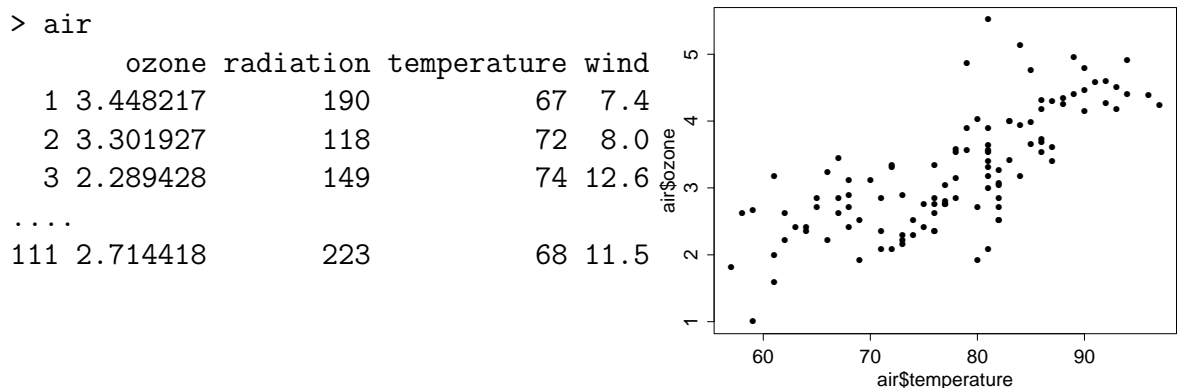
In S werden die Covariablenvektoren und der numerische Response-Vektor Y typischerweise gemeinsam in einem Data Frame zusammengefasst und die Designmatrix X durch eine Modellformel als Funktion der Covariablen beschrieben. Die Anpassung (= der Fit) des linearen Regressionsmodells wird durch die Funktion `lm()` (von “linear model”) bewerkstelligt. Ihre wichtigsten Argumente sind `formula` und `data`. Dabei spezifiziert `formula` (auf eine recht kompakte Weise) das Modell. Die Daten, sprich (Co-)Variablen werden aus dem `data` zugewiesenen Data Frame entnommen. Das Resultat der Funktion `lm()` ist ein so genanntes `lm`-Objekt (der Klasse `lm`), das neben dem Kleinste-Quadrate-Schätzer (KQS) $\hat{\beta}$ für β noch weitere diagnostisch sowie inferenzstatistisch wichtige Größen enthält.

1.1 Die einfache lineare Regression

Im Fall der einfachen linearen Regression besteht X nur aus zwei Spalten (d. h., $p = 2$) und die erste Spalte von X nur aus Einsen. Das Modell reduziert sich dann zu

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad \text{für } i = 1, \dots, n.$$

Zur Illustration verwenden wir den in S-PLUS eingebauten Datensatz `air`, der ein Data Frame ist und in seiner Komponente `ozone` Ozon-Konzentrationen enthält, die jeweils bei Temperaturen beobachtet wurden, die in der Komponente `temperature` verzeichnet sind. Hier ein Ausschnitt aus `air` und ein Streudiagramm von `ozone` gegen `temperature`:



Wir wollen den Ozon-Gehalt in Abhängigkeit von der Temperatur durch eine einfache lineare Regression modellieren, d. h., es soll das Modell

$$\text{ozone}_i = \beta_0 + \beta_1 \cdot \text{temperature}_i + \varepsilon_i \quad \text{für } i = 1, \dots, n \equiv 111$$

mit $\varepsilon_1, \dots, \varepsilon_n$ i.i.d. $\sim \mathcal{N}(0, \sigma^2)$ „gefittet“ werden. In S wird das wie folgt realisiert:

Einfache lineare Regression:

```
> lm( formula= ozone ~ temperature,
+ data= air)

Call: lm(formula = ozone ~ temperature,
data = air)

Coefficients:
(Intercept) temperature
-2.225984  0.07036344

Degrees of freedom: 111 total;
109 residual
Residual standard error: 0.5884748

> lm( ozone ~ temperature, air)
.... (Liefert genau dasselbe.)
```

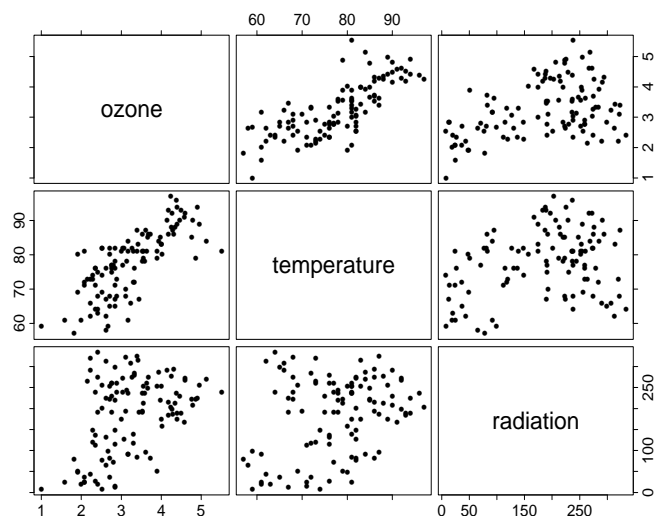
Fittet die einfache lineare Regression von ozone an temperature (beide aus dem Data Frame air). Die Tilde ~ im Argument formula bedeutet, dass die links von ihr stehende Variable von der rechts von ihr stehenden abhängen soll. Ist eine dieser Variablen nicht in dem data zugewiesenen Data Frame vorhanden, so wird sie unter den anderen benutzerdefinierten Variablen gesucht.

Resultate: Der Funktionsaufruf (Call), die Koeffizientenschätzer $\hat{\beta}_0$ ((Intercept)) und $\hat{\beta}_1$ (temperature), die Freiheitsgrade n (total) und die RSS-Freiheitsgrade $n - p$ (residual) sowie die Residuen-Standardabweichung $\hat{\sigma} = \sqrt{\text{RSS}/(n - p)}$ (Residual standard error) als Schätzer für σ .

<pre> > oz1.lm <- lm(ozone ~ temperature, + air) > summary(oz1.lm) Call: lm(formula = ozone ~ temperature, data = air) Residuals: Min 1Q Median 3Q Max -1.49 -0.4258 0.02521 0.3636 2.044 Coefficients: Value Std. Error (Intercept) -2.2260 0.4614 temperature 0.0704 0.0059 t value Pr(> t) (Intercept) -4.8243 0.0000 temperature 11.9511 0.0000 Residual standard error: 0.5885 on 109 degrees of freedom Multiple R-Squared: 0.5672 F-statistic: 142.8 on 1 and 109 degrees of freedom, the p-value is 0 Correlation of Coefficients: (Intercept) temperature -0.9926 </pre>	<p>Die Anwendung von <code>summary()</code> auf das gespeicherte <code>lm</code>-Objekt liefert detailliertere Informationen über das gefittete Modell: Den Aufruf, die "summary statistics" der Residuen (Residuals), zusätzlich zu den Koeffizientenschätzern deren geschätzte Standardabweichung (Std. Error), die Werte der t-Teststatistiken (t value) der zweiseitigen Hypothesentests $H_0 : \beta_i = 0$ (für $i = 0, 1$) samt deren p-Werten ($\Pr(> t)$). Dann die Residuen-Standardabweichung $\hat{\sigma} = \sqrt{\text{RSS}/(n-p)}$ (Residual standard error) als Schätzer für σ. Des Weiteren den multiplen R^2-Wert (Multiple R-Squared, hier: Quadrat des empirischen Pearson'schen Korrelationskoeffizienten der (Y_i, x_i)), den Wert der F-Teststatistik (F-statistic) des „globalen“ F-Tests (auch: "overall F-test") $H_0 : \beta_1 = \dots = \beta_{p-1} = 0$ mit ihren Freiheitsgraden $p-1$ und $n-p$ sowie dessen p-Wert. Zuletzt das linke untere Dreieck (ohne Diagonale) der Matrix $\hat{\sigma}^2(X'X)^{-1}$ (= geschätzte Korrelationen der Koeffizienten).</p>
---	---

1.2 Die multiple lineare Regression

Der nebenstehende Plot aller paarweisen Streudiagramme von `ozone`, `temperature` und `radiation` aus dem Data Frame `air` (erzeugt durch `pairs(air[c("temperature", "radiation", "ozone")])`) deutet darauf hin, dass eventuell auch die Strahlung einen Einfluss auf die Ozon-Werte hat. Daher soll ein Modell untersucht werden, in dem `ozone` gemeinsam durch `temperature` und `radiation` beschrieben wird.



Die Anpassung des multiplen linearen Regressionsmodells

$$\text{ozone}_i = \beta_0 + \beta_1 \cdot \text{temperature}_i + \beta_2 \cdot \text{radiation}_i + \varepsilon_i \quad \text{für } i = 1, \dots, n \equiv 111$$

(also hier für $p = 3$) mit $\varepsilon_1, \dots, \varepsilon_n$ i.i.d. $\sim \mathcal{N}(0, \sigma^2)$ ist genauso einfach wie der Fall der einfachen linearen Regression und soll anhand des folgenden Beispiels verdeutlicht werden. Die Formulierung und der Fit von noch komplizierteren Modellen, d. h., mit noch mehr Covariablen ist "straightforward" durch Erweiterung der Modellformel zu erreichen:

Multiple lineare Regression:	
<pre>> oz2.lm <- lm(ozone ~ temperature + + radiation, air); summary(oz2.lm) Call: lm(formula = ozone ~ temperature + radiation, data = air) Residuals: Min 1Q Median 3Q Max -1.183 -0.4025 -0.03355 0.2965 1.95 Coefficients: Value Std. Error t value (Intercept) -2.1530 0.4398 -4.8951 temperature 0.0643 0.0059 10.9681 radiation 0.0021 0.0006 3.4968 Pr(> t) 0.0000 0.0000 0.0007 Residual standard error: 0.5603 on 108 degrees of freedom Multiple R-Squared: 0.6112 F-statistic: 84.88 on 2 and 108 degrees of freedom, the p-value is 0</pre>	<p>Fittet eine multiple lineare Regression von <code>ozone</code> an <code>temperature</code> und <code>radiation</code> (beide aus dem Data Frame <code>air</code>) und speichert sie als ein <code>lm</code>-Objekt.</p> <p>Dabei soll die links der Tilde <code>~</code> in der Formel stehende Variable in einem multiplen linearen Modell additiv von den rechts von ihr stehenden Variablen abhängen. Dies wird durch das <code>+</code> zwischen den beiden unabhängigen Variablen erreicht.</p> <p>(Bemerkungen: Das <code>+</code> am Zeilenanfang (vor <code>radiation</code>) ist hier nur der S-Zeilenfortsetzungsprompt.</p> <p>Es sind noch weitere Verknüpfungsoperatoren für die unabhängigen Variablen zulässig, auf die wir in Abschnitt 1.3 eingehen werden.)</p> <p>Resultate: Völlig analog zu denen im einfachen linearen Modell.</p>
(Forts.: Multiple lineare Regression)	
<pre>Correlation of Coefficients: (Intercept) temperature temperature -0.9616 radiation 0.0474 -0.2941</pre>	<p>Hier ist das linke untere Dreieck (ohne Diagonale) der Matrix $\hat{\sigma}^2(X'X)^{-1}$, also die geschätzten Korrelationen der Koeffizienten, etwas „gehaltvoller“.</p>

Um an die Koeffizientenschätzwerte $\hat{\beta}' = (\hat{\beta}_0, \dots, \hat{\beta}_{p-1})$, die Residuen $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$ oder die gefitteten Werte \hat{Y}_i , $i = 1, \dots, n$, eines Fits heranzukommen, stehen die folgenden, auf `lm`-Objekte anwendbaren Funktionen zur Verfügung:

Hilfsfunktionen für die Analyse einer linearen Regression:	
<pre>> coef(oz2.lm) (Intercept) temperature radiation -2.153033 0.06433168 0.002144325</pre>	Die Funktion <code>coefficients()</code> (Abk. <code>coef()</code>) angewendet auf das <code>lm</code> -Objekt einer linearen Regression liefert die gefitteten Koeffizienten als Vektor $\hat{\beta}' = (\hat{\beta}_0, \dots, \hat{\beta}_{p-1})$ zurück.
<pre>> resid(oz2.lm) 1 2 3 0.8836053 0.5700483 -0.6375879</pre>	Die Funktion <code>residuals()</code> (Abk. <code>resid()</code>) angewendet auf ein <code>lm</code> -Objekt liefert den Residuenvektor $\hat{\varepsilon}' = (\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n)$ und die Funktion <code>fitted()</code> den Vektor der gefitteten Werte $\hat{Y}' = (\hat{Y}_1, \dots, \hat{Y}_n)$.
<pre>> fitted(oz2.lm) 1 2 3 2.564612 2.731879 2.927016</pre>	

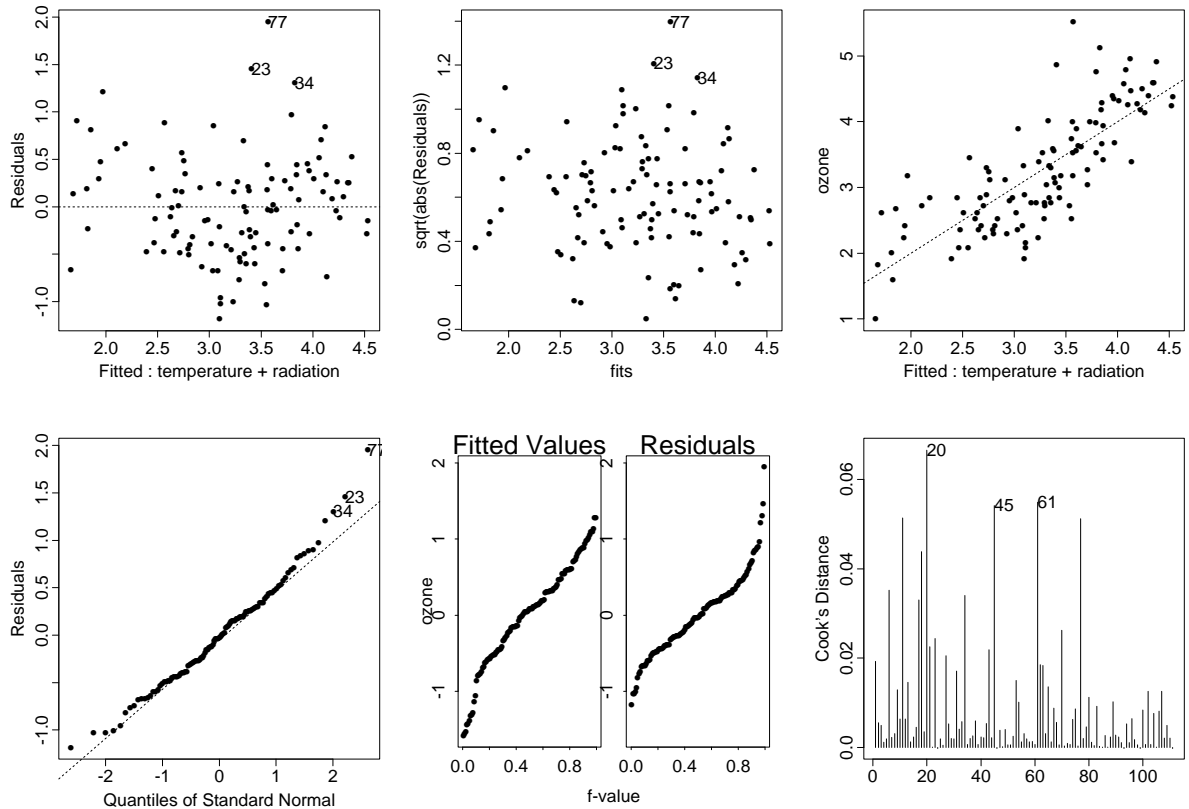
Zur qualitativen Diagnose des Fits dienen die folgenden grafischen Darstellungen:

Diagnose-Plots für die lineare Regression:	
<pre>> plot(oz2.lm)</pre>	Liefert in S-PLUS Version 6.0 für das in <code>oz2.lm</code> gespeicherte lineare Modell automatisch (!) die sechs auf der nächsten Seite oben stehenden Plots.

Von links nach rechts und von oben nach unten:

1. Die Residuen $\hat{\varepsilon}_i$ gegen die gefitteten Werte \hat{Y}_i . Voreinstellungsgemäß werden die drei extremsten Beobachtungen als potenzielle Ausreißer durch ihre Indexnummern markiert. Es sollte kein Trend oder Muster in der Punktwolke erkennbar sein.
2. Die Wurzel der Absolutwerte der Residuen $\sqrt{|\hat{\varepsilon}_i|}$ gegen die gefitteten Werte \hat{Y}_i . Auch hier werden die drei extremsten Beobachtungen durch ihre Indices markiert.
3. Die Beobachtungen Y_i gegen die gefitteten Werte \hat{Y}_i , die sich möglichst trend- und musterfrei um die (gepunktete) Identität gruppieren sollten.
4. Ein Q-Q-Plot der Residuen $\hat{\varepsilon}_i$ samt Soll-Linie, um die Plausibilität der Normalverteilungsannahme der Fehler ε_i zu beurteilen. Wiederum werden die drei extremsten Beobachtungen durch ihre Indexnummern markiert.
5. Ein Residuen-Fit-Streuungsplot ("residual-fit spread plot") der „zentrierten“ OSn der gefitteten Werte $\hat{Y}_{i:n} - \bar{Y}$ und der OSn der Residuen $\hat{\varepsilon}_{i:n}$ jeweils gegen $f_i = (i - 0.5)/n$, um die Streuungsreduktion durch die Regression zu begutachten.
6. Ein Plot der Cook-Abstände (auf die wir in Abschnitt 10.7 „Modell-Diagnose II“ eingehen) gegen ihre Indices, um potenziell einflussreiche Beobachtungen zu identifizieren. Die drei einflussreichsten sind durch ihre Indices markiert.

Diagnose-Plots des Ozondaten-Modells oz2.lm



1.3 Zur Syntax von Modellformeln

Das Konzept des Formelausdrucks (`formula`) in S erlaubt es, eine Vielzahl von Modellen für die Beziehung zwischen Variablen auf sehr kompakte Weise zu spezifizieren und für verschiedene statistische Verfahren in einheitlicher Form zu verwenden (also nicht nur für lineare Regressionsmodelle). Wir konzentrieren uns zunächst auf ihre Verwendung für stetige Design-Variablen in linearen Regressionsmodellen. Auf diskrete Design-Variablen gehen wir im Abschnitt 10.9 „Faktor-Variablen und Interaktionsterme im linearen Regressionsmodell“ ein.

In den Abschnitten 1.1 und 1.2 sind wir schon zwei einfachen Versionen von Formelausdrücken begegnet. Mittels der Funktion `formula()` kann ein solcher Formelausdruck auch als ein eigenständiges Objekt der Klasse `formula` erzeugt und gespeichert werden.

Formelobjekte:	
<pre>> oz.form1 <- formula(ozone ~ temperature) > oz.form1 ozone ~ temperature</pre>	<p>Erzeugt und speichert (im Objekt <code>oz.form1</code>) die Formel des einfachen linearen Modells</p> $\text{ozone}_i = \beta_0 + \beta_1 \text{temperature}_i + \varepsilon_i.$

Anhand von einigen Beispielformeln soll die Formelsyntax erläutert werden:

S-Formel Ausdruck:	Bedeutung:
ozone ~ 1	Das so genannte „Null-Modell“, das nur aus dem konstanten Term β_0 besteht: $\text{ozone}_i = \beta_0 + \varepsilon_i$
ozone ~ temperature + radiation	Das (bereits bekannte) Zwei-Variablen-Modell mit konstantem Term β_0 : $\text{ozone}_i = \beta_0 + \beta_1 \text{temperature}_i + \beta_2 \text{radiation}_i + \varepsilon_i$
ozone ~ -1 + temperature + radiation	Obiges Modell ohne den konstanten Term β_0 (genannt „Regression durch den Ursprung“): $\text{ozone}_i = \beta_1 \text{temperature}_i + \beta_2 \text{radiation}_i + \varepsilon_i$
ozone ~ temperature + radiation + temperature : radiation	Zwei-Variablen-Modell mit Interaktion (a : b bedeutet die Interaktion zwischen a und b; zur Erläuterung siehe Abschnitt 1.4): $\text{ozone}_i = \beta_0 + \beta_1 \text{temperature}_i + \beta_2 \text{radiation}_i + \beta_3 \text{temperature}_i \text{radiation}_i + \varepsilon_i$
ozone ~ temperature * radiation	Ist äquivalent zum Interaktionsmodell von eben und lediglich eine Abkürzung.
ozone ~ temperature * radiation * wind	Drei-Variablen-Modell mit allen drei Zweifach-Interaktionen und der Dreifach-Interaktion. Es ist äquivalent zu $\begin{aligned} \text{ozone} \sim & \text{temperature} + \text{radiation} + \text{wind} \\ & + \text{temperature} : \text{radiation} \\ & + \text{temperature} : \text{wind} \\ & + \text{radiation} : \text{wind} \\ & + \text{temperature} : \text{radiation} : \text{wind} \end{aligned}$

<pre> ozone ~ temperature * radiation * wind - temperature : radiation : wind </pre>	<p>Drei-Variablen-Modell mit allen drei Zweifach-Interaktionen, aber ohne die Dreifach-Interaktion. Es ist äquivalent zu</p> <pre> ozone ~ temperature + radiation + wind + temperature : radiation + temperature : wind + radiation : wind </pre> <p>(Mit dem Minuszeichen lässt sich <i>jeder</i> Term aus einer Modellformel wieder entfernen.)</p>
<pre> ozone ~ (temperature + radiation + wind)^2 </pre>	<p>Eine dritte Notation für das Modell von eben mit allen Zweifach-Interaktionen, aber ohne die Dreifach-Interaktion. (Mit <i>m</i> an Stelle der 2 werden alle Terme bis zur „Interaktions-Ordnung“ <i>m</i> eingebaut.)</p>

Offenbar haben die Operatoren +, -, :, * und ^m in Formeln eine spezielle, „nicht-arithmetische“ Bedeutung, wenn sie rechts vom ~ auftreten. (Dies gilt auch für den Operator /, auf dessen Sonderbedeutung wir erst in Abschnitt 10.10 „Faktor-Variablen und Interaktionsterme im linearen Regressionsmodell“ eingehen.) Andererseits dürfen die rechts wie links vom ~ in einer Formel auftretenden Variablen durch alle Funktionen transformiert werden, deren Resultat wieder als eine Variable interpretierbar ist. Dies trifft insbesondere auf alle mathematischen Funktionen wie `log()`, `sqrt()` etc. zu (wie sie z. B. bei der Anwendung Varianz stabilisierender oder linearisierender Transformationen zum Einsatz kommen; siehe Abschnitte 1.5.2 und 1.5.3).

(Aber auch Funktionen, deren Ergebnis als *mehrere* Variablen aufgefasst werden können, sind zulässig. Ein Beispiel hierfür ist die Funktion `poly()`: Sie erzeugt Basen von Orthogonalpolynomen bis zu einem gewissen Grad, die im Zusammenhang mit der polynomialen Regression eine wichtige Rolle spielen und in Abschnitt 10.9 „Polynomiale Regression“ behandelt werden.)

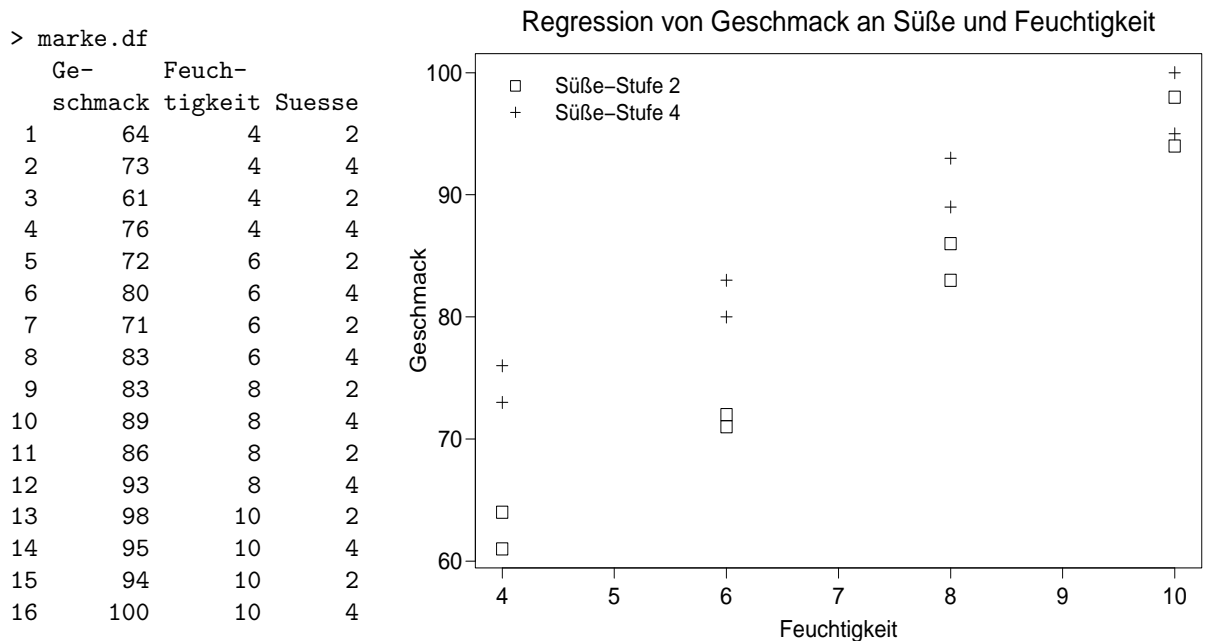
Die oben genannten Operatoren haben nur auf der „obersten Ebene“ einer Formel ihre spezielle Bedeutung, *innerhalb* eines Funktionsaufrufs in einer Formel bleibt es bei ihrer „arithmetischen“ Bedeutung. D. h., dass die zum Beispiel aus zwei Variablen *u* und *v* abgeleitete Variable $x := \log(u + v)$ in S als `log(u + v)` formuliert wird. Sollen die Operatoren jedoch auf der *obersten* Formelebene ihre arithmetische Bedeutung haben, so sind die betroffenen Terme in die Funktion `I()`, wie `Identität`, „einzupacken“. Damit also $x := u + v$ eine einzelne Covariable darstellt, ist in einer S-Formel der Ausdruck `I(u + v)` zu verwenden.

Es folgen einige Formelbeispiele mit transformierten (Co-)Variablen:

S-Formel Ausdruck:	Bedeutung:
$\log(\text{ ozone}) \sim I(\text{ temperature}^2 + \text{ sqrt}(\text{ radiation}))$	<p>Transformationen aller Variablen sind möglich; hier haben wir das Modell</p> $\log(\text{ ozone}_i) = \beta_0 + \beta_1 (\text{ temperature}_i)^2 + \beta_2 \sqrt{\text{ radiation}_i} + \varepsilon_i$
$\log(\text{ ozone}) \sim \text{ temperature}^2 + \text{ sqrt}(\text{ radiation/wind}^3)$	<p>Das Quadrat <i>einer</i> Covariablen braucht nicht in I() eingepackt zu werden und innerhalb einer Funktion agieren die arithmetischen Operatoren gemäß ihrer mathematischen Definition:</p> $\log(\text{ ozone}_i) = \beta_0 + \beta_1 (\text{ temperature}_i)^2 + \beta_2 \sqrt{(\text{ radiation}_i/\text{ wind}_i)^3} + \varepsilon_i$
$\log(\text{ ozone}) \sim I(1/\text{ wind})$	<p>Einsatz der Funktion I(), damit / die Bedeutung der Division hat:</p> $\log(\text{ ozone}_i) = \beta_0 + \beta_1 1/\text{ wind}_i + \varepsilon_i$
$\text{ ozone} \sim I(\text{ temperature} + \text{ radiation} + \text{ wind})^2$	<p>Beachte den Unterschied zum Modell mit allen Interaktionen bis zur „Interaktions-Ordnung“ 2 auf der vorherigen Seite:</p> $\text{ ozone}_i = \beta_0 + \beta_1 (\text{ temperature}_i + \text{ radiation}_i + \text{ wind}_i)^2 + \varepsilon_i$

1.4 Zur Interaktion stetiger Covariablen

An einem Beispiel soll die Bedeutung einer Interaktion zwischen zwei stetigen Variablen veranschaulicht werden. In dem unten gezeigten Data Frame `marke.df` sind die Resultate eines kontrollierten Experimentes zur Beurteilung des Geschmacks (auf einer Skala zwischen 0 und 100) einer gewissen Süßware für verschiedene Feuchtigkeits- und Süße-Stufen dokumentiert und in dem nebenstehenden Streudiagramm veranschaulicht.



Auf Grund der Markierung der Beobachtungen durch zwei verschiedene Symbole für die beiden Süße-Stufen ist in dem Streudiagramm deutlich zu erkennen, dass für verschiedene Süße-Stufen der Einfluss (!) der Feuchtigkeit auf die Geschmacksbeurteilung des Produktes ein anderer ist. D. h., der quantitative Einfluss der Covariablen „Feuchtigkeit“ auf die Response „Geschmack“ hängt vom Wert der Covariablen „Süße“ ab. Dies nennt man eine Interaktion, und um diesem Effekt Rechnung zu tragen, muss in das aufzustellende Zwei-Variablen-Modell ein geeigneter Interaktionsterm eingebaut werden.

Die einfachste Form der Interaktion stetiger Covariablen ist die multiplikative, weswegen in einem Zwei-Variablen-Modell zusätzlich zu den „Haupteffekten“ β_1 und β_2 der beiden Covariablen ein dritter „Interaktionseffekt“ β_3 für das Produkt der beiden Covariablen eingebaut wird. Das Modell lautet damit

$$\text{Geschmack}_i = \beta_0 + \beta_1 \text{Feuchtigkeit}_i + \beta_2 \text{Suesse}_i + \beta_3 \text{Feuchtigkeit}_i \text{Suesse}_i + \varepsilon_i .$$

In kompakter S-Formelsyntax: `Geschmack ~ Feuchtigkeit * Suesse`

Zunächst erstellen wir einen Fit *ohne* Interaktionsterm, was uns das folgende Resultat liefert (wobei das Argument `correlation= F` in `summary()` die Ausgabe der Koeffizientenkorrelationen unterdrückt):

```
> marke.lm <- lm( Geschmack ~ Feuchtigkeit + Suesse, marke.df)
> summary( marke.lm, correlation= F)
Call: lm(formula = Geschmack ~ Feuchtigkeit + Suesse, data = marke.df)
Residuals:
```

Min	1Q	Median	3Q	Max
-5.525	-1.85	-0.325	1.775	4.975

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	37.5250	3.3451	11.2178	0.0000
Feuchtigkeit	4.8000	0.3362	14.2773	0.0000
Suesse	3.7500	0.7518	4.9883	0.0002

Residual standard error: 3.007 on 13 degrees of freedom

Multiple R-Squared: 0.9462

F-statistic: 114.4 on 2 and 13 degrees of freedom, the p-value is 5.61e-09

Die Plots der Residuen gegen die gefitteten Werte sowie gegen jede der Covariablen und gegen den nicht im Modell befindlichen Interaktionsterm gemäß

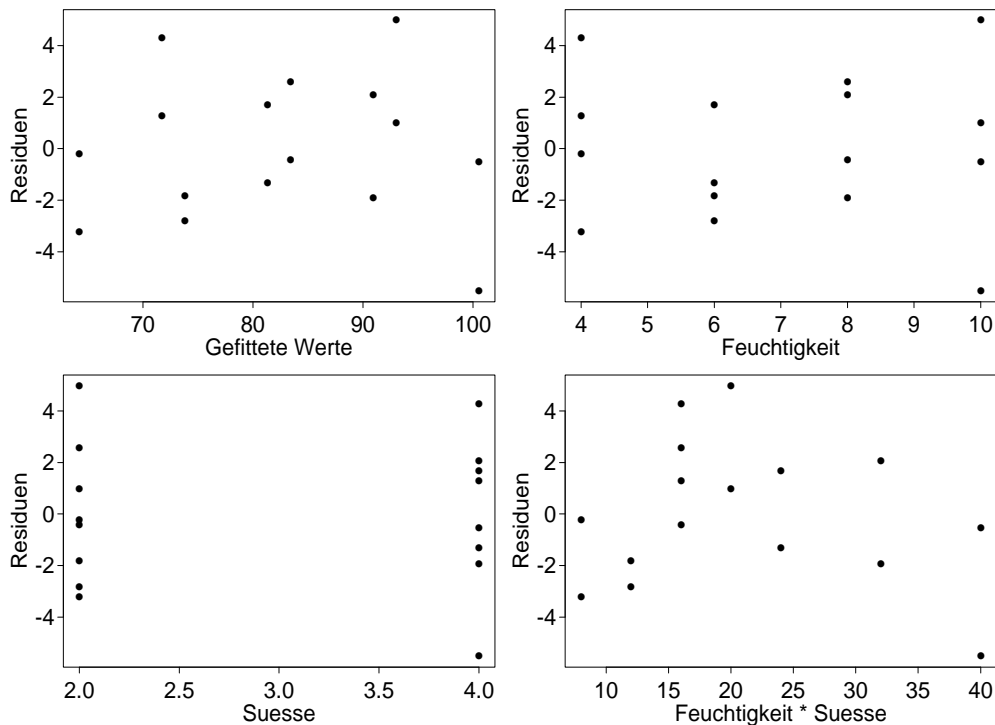
```

> plot( fitted( marke.lm), resid( marke.lm), xlab= "Gefittete Werte",
+ ylab= "Residuen")
> plot( marke.df$Feuchtigkeit, resid( marke.lm), xlab= "Feuchtigkeit",
+ ylab= "Residuen")
> plot( marke.df$Suesse, resid( marke.lm), xlab= "Suesse",
+ ylab= "Residuen")
> plot( marke.df$Feuchtigkeit * marke.df$Suesse, resid( marke.lm),
+ xlab= "Feuchtigkeit * Suesse", ylab= "Residuen")

```

ergeben wir die folgenden Bilder:

Residuen-Plots des Modells OHNE Interaktionsterm



Der Plot der Residuen gegen den Interaktionsterm (rechts unten) zeigt eine leichte Krümmung in der Punktelwolke, was den Schluss zulässt, dass eine Interaktion vorliegen könnte.

Daher fertigen wir nun den Fit *mit* Interaktionsterm an, und es zeigt sich in der Tat, dass die Anpassung an Qualität gewinnt: Die Residuen-Standardabweichung wird kleiner, der R^2 -Wert wird größer und man beachte den signifikanten Beitrag des Interaktionsterms in diesem Modell:

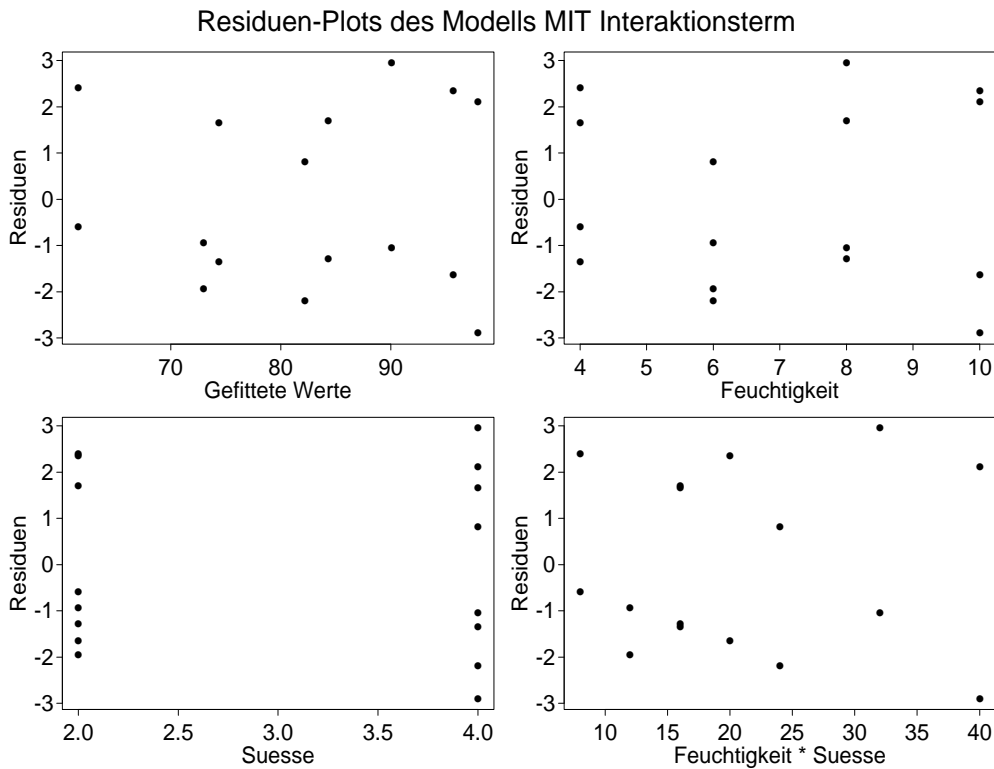
```
> marke2.lm <- lm( Geschmack ~ Feuchtigkeit * Suesse, marke.df)
> summary( marke2.lm, corr= F)
Call: lm(formula = Geschmack ~ Feuchtigkeit * Suesse, data = marke.df)
Residuals:
  Min      1Q  Median      3Q      Max
-2.9 -1.425 -0.775  1.8  2.95
```

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	19.1500	5.6275	3.4029	0.0052
Feuchtigkeit	7.4250	0.7658	9.6957	0.0000
Suesse	9.8750	1.7796	5.5491	0.0001
Feuchtigkeit:Suesse	-0.8750	0.2422	-3.6132	0.0036

Residual standard error: 2.166 on 12 degrees of freedom
 Multiple R-Squared: 0.9742
 F-statistic: 151.3 on 3 and 12 degrees of freedom, the p-value is 8.47e-10

Auch die Residuen-Plots unterstützen dieses Modell: Im Plot der Residuen gegen den Interaktionsterm (rechts unten) ist die leichte Krümmung in der Punktwolke verschwunden.



Bemerkung: Natürlich ist nicht gesagt, dass multiplikative Interaktionen die einzigen möglichen sind. Eine allgemeine Interaktion zwischen Covariablen x_1, \dots, x_k wäre von der Form $f(x_1, \dots, x_k)$, aber das wesentliche Problem dürfte darin bestehen, f richtig zu

spezifizieren. In S steht für f bei stetigen Covariablen über den Operator $:$ nur die Multiplikation zur Verfügung, was auch die einzige Form von Interaktion stetiger Covariablen zu sein scheint, die in Lehrbüchern behandelt wird.

1.5 Modelldiagnose I: Residual-Analyse und Transformationen des Modells

Nach dem, was wir am Beginn von Kapitel 1 zur Erinnerung aufgeführt haben, sind die Residuen $\hat{\varepsilon}_i := Y_i - \hat{Y}_i$, $i = 1, \dots, n$, unter den üblichen Modellannahmen zwar **normalverteilt**, aber weder **unabhängig noch identisch verteilt**, sondern mit

$$\text{Var}(\hat{\varepsilon}_i) = \sigma^2 (1 - h_{ii}) \quad \text{und} \quad \text{Cov}(\hat{\varepsilon}_i, \hat{\varepsilon}_j) = -\sigma^2 h_{ij} \quad \text{für } 1 \leq i \neq j \leq n,$$

wobei $(h_{ij})_{1 \leq i, j \leq n} = H$ die schon eingeführte Projektionsmatrix ist.

Bemerkung: Im *einfachen linearen Regressionsmodell* mit $\hat{\varepsilon}_i := Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i$ für $i = 1, \dots, n$, ist damit

$$h_{ij} = \frac{1}{n} + \frac{(x_i - \bar{x}_n)(x_j - \bar{x}_n)}{\sum_{l=1}^n (x_l - \bar{x}_n)^2} \quad \text{für } 1 \leq i, j \leq n,$$

was für die Residuen

$$\text{Var}(\hat{\varepsilon}_i) = \sigma^2 (1 - h_{ii}) = \sigma^2 \left(1 - \frac{1}{n} - \frac{(x_i - \bar{x}_n)^2}{\sum_{l=1}^n (x_l - \bar{x}_n)^2} \right)$$

liefert. Dies bedeutet, dass die Residuen zu solchen Designwerten, die weiter vom Mittelwert \bar{x}_n entfernt sind, eine kleinere Varianz haben als solche, deren Designwerte näher bei \bar{x}_n liegen. Damit haben Residual-Plots im zentralen Bereich des Designs bei erfüllten Modellannahmen grundsätzlich eine tendenziell größere Variabilität als in den Randbereichen!

1.5.1 Grafische Residual-Analyse

Trotz der eben erwähnten Abhängigkeit und der (meist leichten, aber grundsätzlich vorhandenen!) Varianz-Inhomogenität sind Plots der Residuen $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$ gegen die Designvariable(n) nützlich, um auf Grund von hierbei eventuell zu beobachtenden systematischen Strukturen

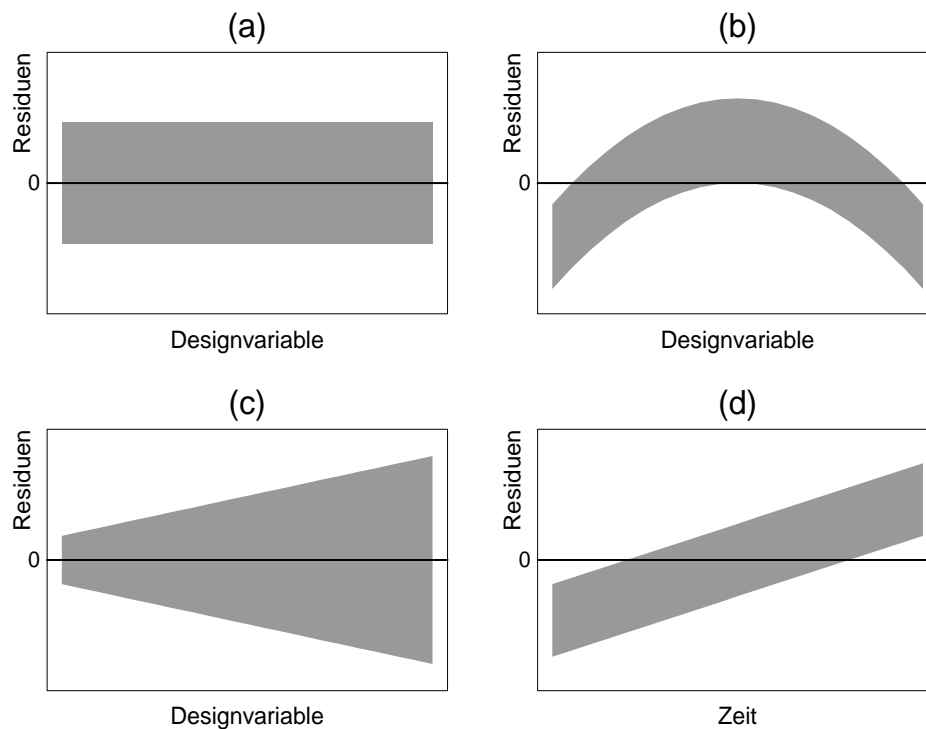
- Nichtlinearität in den Designvariablen,
- Varianz-Inhomogenität in den Störungen und/oder
- „Ausreißer“

entdecken zu können.

Die schematischen Skizzen einiger prototypischer Residual-Plots auf der nächsten Seite oben sollen dies veranschaulichen. Plot ...

- (a) zeigt einen idealtypischen Residual-Plot: Konstante Streuung im gesamten Designwertebereich, keine Krümmung der „Residuenwolke“, kein Ausreißer.
- (b) liefert durch die Krümmung ein Indiz für eine nichtlineare Beziehung zwischen der Response und der Designvariablen.
- (c) dokumentiert eine Abhängigkeit der Streuung vom Wert der Designvariablen, also Varianz-Inhomogenität. Hier: Je größer x , desto stärker streut Y um die Regressionsfunktion.

(d) ist ein Plot der Residuen gegen ihren Index und, wenn die Daten in der Reihenfolge ihrer Indizes erhoben wurden, damit ein Plot gegen die *Zeit*. In diesem Fall impliziert er eine mögliche Abhängigkeit der Residuen voneinander.



Im einfachen linearen Regressionsmodell können äquivalent zu den obigen Plots der Residuen gegen eine Designvariable auch Plots der Residuen gegen die gefitteten Werte $\hat{Y}_1, \dots, \hat{Y}_n$ betrachtet werden. Sie unterscheiden sich lediglich in der Skala der horizontalen Achse von den erstgenannten Plots, da die \hat{Y}_i eine lineare Transformation der x_i sind, so dass der optische Eindruck der Plots derselbe ist. Außerdem sind diese Plots auch im *multiplen* Regressionsmodell (mit mehr als nur einer Designvariablen) anfertigbar und es kommt hinzu, dass die gefitteten Werte und die Residuen unter den Modellannahmen unkorreliert sind: Der Pearsonsche Korrelationskoeffizient für $(\hat{Y}_i, \hat{\varepsilon}_i)$, $i = 1, \dots, n$, ist Null, da $\text{Cov}(\hat{Y}, \hat{\varepsilon}) = 0$. Dies bedeutet, dass im Plot der Residuen gegen die gefitteten Werte insbesondere kein linearer Trend zu erkennen sein dürfte, wenn das Modell korrekt ist!

Falls die Diagnose-Plots und die Modellannahmen im Widerspruch zueinander stehen, kann das, wie oben schon angesprochen, die folgenden möglichen Ursachen haben:

1. Die funktionale Beziehung zwischen Erwartungswert der Response und der (bzw. den) Designvariable(n), d. h. die Regressionsfunktion, ist nicht korrekt formuliert (vgl. Plot (b)).

Dies wiederum kann zwei Gründe haben:

- i) Die Designvariable(n) geht (gehen) nicht in der korrekten funktionalen Form in die Regressionsfunktion ein.
- ii) Die (Mess-Skala der) Response ist nicht in der adäquaten funktionalen Form für das Modell.

2. Über den Designbereich hinweg liegt Varianz-Inhomogenität der Störungen vor (vgl. Plot (c)).
3. Die Störungen sind nicht normalverteilt.
4. Die Störungen sind korreliert (vgl. Plot (d)).

Für die jeweiligen Abweichungen von den Modellannahmen kann möglicherweise **Abhilfe** geschaffen werden, indem

- eine linearisierende Transformation auf die Designvariable(n) oder die Response angewendet wird,
- eine Varianz stabilisierende Transformation der Response vorgenommen wird,
- spezielle Kovarianz-Modelle für die Störungen angesetzt werden (worauf wir hier aber nicht eingehen).

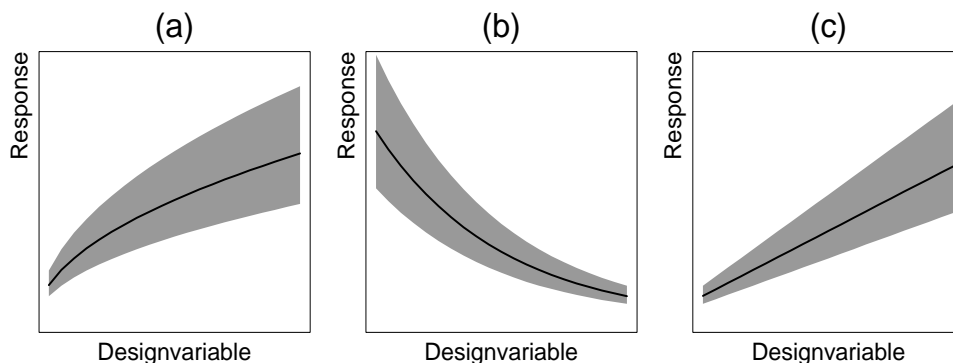
1.5.2 Varianz stabilisierende Transformationen

Wenn davon ausgegangen werden kann, dass die **Varianz eine Funktion der mittleren Response** (also des Wertes der Regressionsfunktion) ist, kann es nützlich sein, die Response zu transformieren. Wir beschränken uns hier auf eine grafische Methode zur Auswahl geeigneter Transformationen:

Auf dieser Seite unten sind drei prototypische *Regressionsplots* mit Vorschlägen für **Response-Transformationen zur Varianzstabilisierung** aufgeführt. Darunter geben wir ihre dazugehörige S-Formelsyntax unter Verwendung des Ozon-Beispiels an.

Als „Nebenwirkung“ dieser Transformationen wird häufig eine (willkommene) Linearisierung der Regressionsbeziehung und eine „Normalisierung“ der Verteilung der Störungen beobachtet.

Zusätzlich zur Transformation der Response kann es durchaus hilfreich oder sogar notwendig sein, gleichzeitig eine Design-Transformation (wie im nächsten Abschnitt beschrieben) durchzuführen, um den Modell-Fit zu verbessern. (Detaillierter wird hierauf z. B. in Box, Hunter und Hunter (1978) eingegangen.)



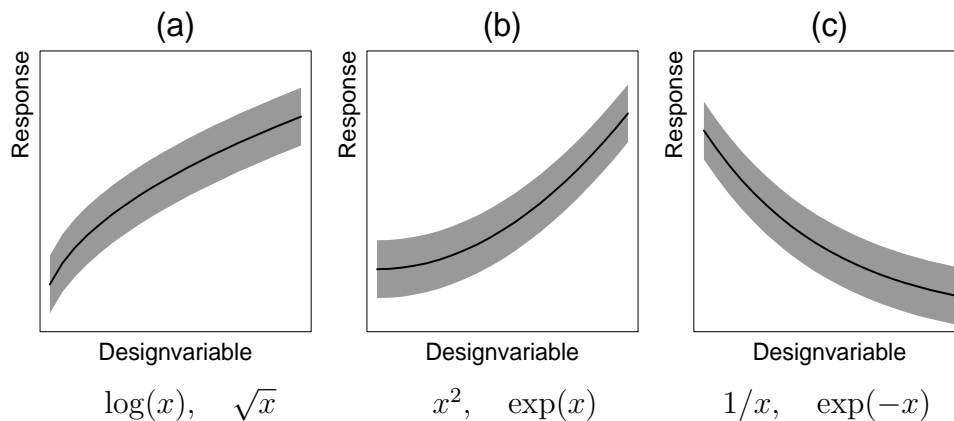
Mögliche Y-Transformationen: \sqrt{Y} , $\log(Y)$, $1/Y$.

Varianz stabilisierende Transformationen:		
Transf.	S-Formel	Modell
\sqrt{Y}	$\text{sqrt}(\text{ozone}) \sim \text{temperature}$	$\sqrt{\text{ozone}_i} = \beta_0 + \beta_1 \text{temperature}_i + \varepsilon_i$
$\log(Y)$	$\log(\text{ozone}) \sim \text{temperature}$	$\log(\text{ozone}_i) = \beta_0 + \beta_1 \text{temperature}_i + \varepsilon_i$
$1/Y$	$1/\text{ozone} \sim \text{temperature}$	$1/\text{ozone}_i = \beta_0 + \beta_1 \text{temperature}_i + \varepsilon_i$

1.5.3 Linearisierende Transformationen

Wir gehen davon aus, dass die **Varianz konstant/homogen** ist, und geben auch hier eine grafische Methode zur Bestimmung der passenden Transformation an:

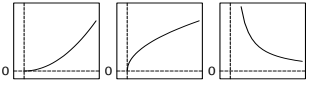
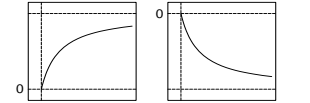
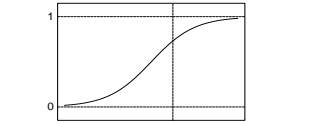
Die folgenden, prototypischen nichtlinearen Regressionsbeziehungen bei konstanter Varianz legen die darunter angegebenen **Design-Transformationen zur Linearisierung** der Beziehung zwischen Response und Design nahe. Für jede Transformation sollte ein eigener Modell-Fit durchgeführt werden, um die „beste“ Anpassung zu finden.



Es folgt die S-Formelsyntax tabelliert (wieder im Ozon-Beispiel):

Linearisierende Transformationen:		
Transf.	S-Formel	Modell
$\log(x)$	$\text{ozone} \sim \log(\text{temperature})$	$\text{ozone}_i = \beta_0 + \beta_1 \log(\text{temperature}_i) + \varepsilon_i$
\sqrt{x}	$\text{ozone} \sim \text{sqrt}(\text{temperature})$	$\text{ozone}_i = \beta_0 + \beta_1 \sqrt{\text{temperature}_i} + \varepsilon_i$
x^2	$\text{ozone} \sim \text{I}(\text{temperature}^2)$ (Polynomiale Regression ist speziell, siehe Abschnitt 10.9!)	$\text{ozone}_i = \beta_0 + \beta_1 \text{temperature}_i^2 + \varepsilon_i$
$\exp(x)$	$\text{ozone} \sim \exp(\text{temperature})$	$\text{ozone}_i = \beta_0 + \beta_1 \exp(\text{temperature}_i) + \varepsilon_i$
$1/x$	$\text{ozone} \sim \text{I}(1/\text{temperature})$	$\text{ozone}_i = \beta_0 + \beta_1 1/\text{temperature}_i + \varepsilon_i$
$\exp(-x)$	$\text{ozone} \sim \exp(-\text{temperature})$	$\text{ozone}_i = \beta_0 + \beta_1 \exp(-\text{temperature}_i) + \varepsilon_i$

Hier folgt noch eine Auflistung linearisierbarer Regressionsfunktionen aus speziellen Anwendungen samt ihrer zugehörigen Transformationen und S-PLUS-Formelsyntax, wobei \mathbf{Y} der Response-Vektor und \mathbf{x} die unabhängige Variable ist.

Spezielle linearisierbare Regressionsfunktionen:		
Linearisierbare Fkt.	Transformation und Modell	Graph & S ⁺ -Formel
$y = \theta_0 x^{\theta_1}$ (Cobb-Douglas-Funktion; Ökonomie)	$\log(y) = \log(\theta_0) + \theta_1 \log(x)$ $\log(y_i) = \beta_0 + \beta_1 \log(x_i) + \varepsilon_i$	 $\log(y) \sim \log(x)$
$y = \frac{\theta_0 x}{\theta_1 + x}$ (Bioassay- oder Dosis-Response-Modell; Biologie)	$\frac{1}{y} = \frac{1}{\theta_0} + \frac{\theta_1}{\theta_0} \frac{1}{x}$ $\frac{1}{y_i} = \beta_0 + \beta_1 \frac{1}{x_i} + \varepsilon_i$	 $1/y \sim I(1/x)$
$y = \frac{\exp(\theta_0 + \theta_1 x)}{1 + \exp(\theta_0 + \theta_1 x)}$ (Logistische Funktion)	$\log\left(\frac{y}{1-y}\right) = \theta_0 + \theta_1 x$ $\log\left(\frac{y_i}{1-y_i}\right) = \beta_0 + \beta_1 x_i + \varepsilon_i$	 $\log(y/(1-y)) \sim x$

Beachte: Bei allen obigen Modellen muss die Annahme der additiven Fehler im *transformierten* Modell gerechtfertigt sein! Wenn dies nicht der Fall ist, können zwar die Parameterschätzwerte noch vernünftig sein, aber die inferenzstatistischen Verfahren der linearen Modelle (Hypothesentests, Konfidenzintervalle usw.) sind nicht mehr gültig.

1.6 Modifizierung eines linearen Regressionsmodells

Die Analyse eines bestehenden linearen Modells kann ergeben, dass nicht alle der Design-Variablen (in der Form, in der sie im Modell auftauchen) einen signifikanten Einfluss haben. Das Entfernen eines Terms oder mehrerer Terme aus dem Modell könnte den Fit vereinfachen, ohne ihn wesentlich zu verschlechtern. Umgekehrt mag es nahe liegen, weitere Terme zu einem bestehenden Modell hinzuzufügen, um den Fit zu verbessern. In beiden Fällen ist es interessant zu quantifizieren, ob das reduzierte bzw. erweiterte Modell im Vergleich zum Ausgangsmodell einen besseren Fit liefert.

Zur Modifikation eines bestehenden Modells, das in einem `lm`-Objekt gespeichert ist, stehen in S-PLUS 6.0 mehrere Funktionen zur Verfügung. Zum einen ist dies die Funktion `update()`, die im Wesentlichen Tipp- und Rechenarbeit ersparen kann, wenn ein Modell gezielt verändert werden soll. Zum anderen sind es die Funktion `drop1()`, die ausgehend von einem bestehenden Modell die Wirkung des Weglassens einzelner, im Modell befindlicher Design-Variablen bestimmt, und die Funktion `add1()`, die dasselbe für die einzelne Hinzunahme weiterer, nicht im Modell befindlicher Design-Variablen durchführt. Das Vergleichskriterium für Ausgangs- und modifiziertes Modell, das hier zum Einsatz kommt, ist die so genannte C_p -Statistik von Mallows, die wir in Abschnitt 1.6.2 etwas ausführlicher vorstellen werden.

Doch zunächst zur Funktion `update()`, die hauptsächlich eine organisatorische Hilfe darstellt.

1.6.1 Die Funktion `update()`

Die Funktion `update()` erlaubt es, ein neues Modell ausgehend von einem bestehenden zu kreieren, indem lediglich diejenigen Argumente angegeben werden, die geändert werden sollen. Ihre zwei wesentlichen Argumente heißen `object` und `formula`, wobei `object` das Modell-Objekt, wie z. B. ein `lm`-Objekt erwartet, welches modifiziert werden soll. Das Argument `formula` erwartet die modifizierte Modellformel, wobei ein Punkt (`.`) auf der linken oder rechten Seite der Tilde `~` durch die linke oder rechte Seite der ursprünglichen Modellformel aus `object` ersetzt wird. (Ein alleiniger Punkt links der Tilde kann weggelassen werden.)

Das in Abschnitt 1.2 erstellte Modell `ozone.fit2` hätte zum Beispiel aus dem bereits in Abschnitt 1.1 generierten `lm`-Objekt `ozone.fit` wie folgt erzeugt werden können:

```
> ozone.fit2 <- update( object= ozone.fit, formula= . ~ . + radiation)
```

Oder noch kürzer, indem die Argumentbenennungen und der alleinige Punkt links der Tilde weggelassen werden:

```
> ozone.fit2 <- update( ozone.fit, ~ . + radiation)
```

Eine (hier logarithmische) Transformation der Response-Variablen bei gleichzeitigem Ausschluss des konstanten Terms β_0 aus dem Modell erreicht man z. B. durch:

```
> ozone.fit2b <- update( ozone.fit, log(.) ~ . -1)
```

In den folgenden Abschnitten werden uns noch weitere Anwendungsbeispiele der Funktion `update()` begegnen.

1.6.2 Die C_p -Statistik und “Akaike’s information criterion”

Es seien ein Ausgangsmodell der Dimension p_0 und ein reduziertes oder erweitertes Modell der Dimension p gegeben. Ein Vergleichskriterium für Ausgangs- und modifiziertes Modell ist die so genannte C_p -Statistik von Mallows mit

$$C_p = \frac{\text{RSS}}{\hat{\sigma}_0^2} + 2p - n . \quad (1)$$

Dabei ist RSS die Residuenquadratsumme im modifizierten Modell und $\hat{\sigma}_0^2 = \text{RSS}_0/(n-p_0)$ der Varianzschätzer im Ausgangsmodell.

Aus der Definition von C_p wird deutlich, dass die Modellkomplexität in Form der Dimension p und die durch RSS quantifizierte Fit-Güte antagonistisch wirken: Ein kleinerer C_p -Wert ergibt sich nur dann, wenn bei einer Dimensionsreduktion der RSS-Wert nicht zu stark ansteigt, d. h., die Fit-Güte nicht zu sehr darunter leidet, oder bei einer Modellvergrößerung der RSS-Wert hinreichend stark zurückgeht, d. h., die Fit-Güte entsprechend zunimmt. Ziel ist es, einen möglichst niedrigen Wert für C_p zu erhalten (siehe z. B. Hocking (1996)).

In S-PLUS wird eine transformierte Version von Mallows’s C_p verwendet, die dasselbe Verhalten zeigt, nämlich “Akaike’s information criterion”

$$\text{AIC}_p = \hat{\sigma}_0^2(C_p + n) = \text{RSS} + 2p\hat{\sigma}_0^2 . \quad (2)$$

Für das Verständnis der Funktionsweise der C_p -Statistik bzw. von Akaike’s information criterion ist es hilfreich, die beiden etwas genauer zu betrachten. Wegen $\hat{\sigma}^2 = \text{RSS}/(n-p)$ bzw. $\hat{\sigma}_0^2 = \text{RSS}_0/(n-p_0)$ lassen sich (1) und (2) umformen zu

$$C_p = \frac{(n-p)\hat{\sigma}^2}{\hat{\sigma}_0^2} - (n-p) + p = (n-p) \left(\frac{\hat{\sigma}^2}{\hat{\sigma}_0^2} - 1 \right) + p \quad (3)$$

und

$$\text{AIC}_p = \text{RSS}_0 \frac{n + C_p}{n - p_0} . \quad (4)$$

Es folgt noch eine tabellarische Übersicht für C_p und AIC_p in den möglichen Szenarien „Ausgangsmodell“, „reduziertes Modell“ und „erweitertes Modell“:

Modell	Mallows’s C_p	Akaike’s information criterion
Ausgangsmodell mit Dimension p_0	$C_{p_0} = p_0$	$\text{AIC}_{p_0} = \text{RSS}_0 \frac{n+p_0}{n-p_0}$
Erweitertes Modell mit Dimension $p_+ > p_0$	$C_{p_+} = \underbrace{\frac{\text{RSS}_+}{\hat{\sigma}_0^2}}_{\leq \text{RSS}_0/\hat{\sigma}_0^2} + \underbrace{2p_+ - n}_{> 2p_0}$	$\text{AIC}_{p_+} = \text{RSS}_0 \frac{n+C_{p_+}}{n-p_0}$
Reduziertes Modell mit Dimension $p_- < p_0$	$C_{p_-} = \underbrace{\frac{\text{RSS}_-}{\hat{\sigma}_0^2}}_{\geq \text{RSS}_0/\hat{\sigma}_0^2} + \underbrace{2p_- - n}_{< 2p_0}$	$\text{AIC}_{p_-} = \text{RSS}_0 \frac{n+C_{p_-}}{n-p_0}$

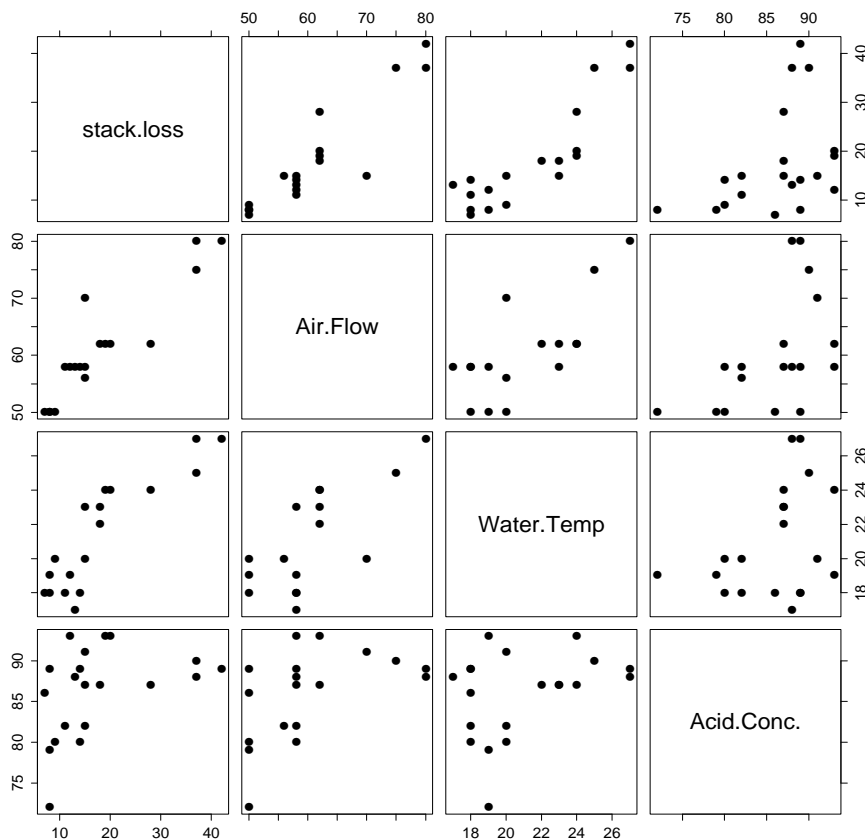
Die Berechnung der C_p -Statistik für Modelle, in denen sich p_- bzw. p_+ um genau 1 von p_0 unterscheiden, ist in den Funktionen `drop1()` bzw. `add1()` implementiert. Deren Verwendung wird in den beiden folgenden Abschnitten erläutert.

1.6.3 Das Entfernen eines Terms: `drop1()`

Zur Illustration der Funktionsweise von `drop1()` betrachten wir die eingebauten Datensätze `stack.loss` und `stack.x`, die gemeinsam Informationen über den Verlust von Ammoniak unter verschiedenen Umgebungsbedingungen in einem gewissen Fabrikationsprozess enthalten. Zusammengesetzt in einem Data Frame namens `stack.df` lassen sie sich einfach für das Fitten von linearen Regressionsmodellen verwenden:

```
> stack.df <- data.frame( stack.loss, stack.x)
> stack.df
  stack.loss Air.Flow Water.Temp Acid.Conc.
1         42      80         27          89
2         37      80         27          88
...
21        15      70         20          91
```

`pairs(stack.df)` liefert die unten stehenden paarweisen Streudiagramme, und demnach scheinen alle drei der Variablen `Air.Flow`, `Water.Temp` und `Acid.Conc.` einen Einfluss auf den `stack.loss` (Ammoniak-Verlust) zu haben.



Wir fitten das Modell

$$\text{stack.loss} = \beta_0 + \beta_1 \text{Air.Flow} + \beta_2 \text{Water.Temp} + \beta_3 \text{Acid.Conc.} + \varepsilon$$

folgendermaßen:

```
> stack.fit <- lm( stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
+ data= stack.df);      summary( stack.fit, corr= F)
Call: lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
      data = stack.df)
```

```
Residuals:      Min       1Q   Median       3Q      Max
      -7.238  -1.712  -0.4551  2.361  5.698
```

```
Coefficients:                Value Std. Error  t value Pr(>|t|)
(Intercept) -39.9197   11.8960   -3.3557  0.0038
  Air.Flow    0.7156    0.1349    5.3066  0.0001
  Water.Temp  1.2953    0.3680    3.5196  0.0026
  Acid.Conc. -0.1521    0.1563   -0.9733  0.3440
```

Residual standard error: 3.243 on 17 degrees of freedom

Multiple R-Squared: 0.9136

F-statistic: 59.9 on 3 and 17 degrees of freedom, the p-value is 3.016e-09

Bemerkung: Ist die Response-Variable die erste Spalte eines Data Frames und sollen alle restlichen Spalten in das multiple lineare Regressionsmodell eingehen, so lässt sich die Modellformel und das data-Argument im Aufruf von `lm()` einsparen, wie in nebenstehendem Beispiel der Fall.

```
> lm( stack.df)
Call: lm(formula = stack.df)
Coefficients:
(Intercept)  Air.Flow Water.Temp Acid.Conc.
 -39.91967  0.7156402   1.295286  -0.1521225

Degrees of freedom: 21 total; 17 residual
Residual standard error: 3.243364
```

Obige Summary und die (nicht gezeigten) Diagnose-Plots für `stack.fit` deuten eine gute Anpassung an, doch ist der Beitrag der Variable `Acid.Conc.` nicht signifikant (da der p -Wert des Tests $H_0 : \beta_3 = 0$ gleich 0.344 ist). Mittels der Funktion `drop1()` soll die Wirkung des Entfernens einer Variable aus `stack.fit` untersucht werden:

Entfernen von Modelltermen mittels <code>drop1()</code> :	
<pre>> drop1(stack.fit) Single term deletions Model: stack.loss ~ Air.Flow + Water.Temp + Acid.Conc. Df Sum of Sq RSS <none> 178.8300 Air.Flow 1 296.2281 475.0580 Water.Temp 1 130.3076 309.1376 Acid.Conc. 1 9.9654 188.7953 Cp 262.9852 538.1745 372.2541 251.9118</pre>	<p>Ausgehend von dem gefitteten Modell <code>stack.fit</code> bestimmt <code>drop1()</code> für jede einzelne der Modellvariablen den Effekt ihres Entfernens.</p> <p>Resultat: Das Ausgangsmodell (Model), eine (Art) ANOVA-Tabelle, in deren Zeile <code><none></code> die Residuenquadratsumme (RSS) und die AIC_p-Statistik (Cp) des Ausgangsmodells angegeben werden. Die weiteren Zeilen enthalten für die jeweils genannte Variable deren zugehörige Freiheitsgrade (df), ihren Anteil (Sum of Sq) an der Residuenquadratsumme (RSS) des um sie reduzierten Modells sowie den zum reduzierten Modell gehörenden Wert der AIC_p-Statistik (Cp).</p> <p>(Es gilt also: „RSS(ohne Variable) = RSS(<none>) + Sum of Sq(Variable)“.)</p>

Das Entscheidungskriterium für eine mögliche Verbesserung des Modells durch Entfernung eines Terms lautet wie folgt: Tritt unter den C_p -Werten, d. h. den AIC_p -Werten der reduzierten Modelle ein Wert auf, der kleiner als derjenige des Ausgangsmodells (in Zeile `<none>`) ist, so ist die Variable mit dem kleinsten dieser C_p -Werte zu entfernen. (Ist der C_p -Wert des Ausgangsmodells der kleinste, so kann das Modell durch Entfernen einer Variable nicht verbessert werden.)

In obigem Beispiel bietet sich demzufolge `Acid.Conc.` zur Entfernung an, was mittels `update()` umgesetzt wird:

```
> stack.fit2 <- update( stack.fit, ~ . - Acid.Conc.)
> summary( stack.fit2)
```

```
Call: lm(formula = stack.loss ~ Air.Flow + Water.Temp, data = stack.df)
```

```
Residuals:
```

```
    Min     1Q  Median     3Q     Max
-7.529 -1.75  0.1894  2.116  5.659
```

```
Coefficients:
```

	Value	Std. Error	t value	Pr(> t)
(Intercept)	-50.3588	5.1383	-9.8006	0.0000
Air.Flow	0.6712	0.1267	5.2976	0.0000
Water.Temp	1.2954	0.3675	3.5249	0.0024

```
Residual standard error: 3.239 on 18 degrees of freedom
```

```
Multiple R-Squared: 0.9088
```

```
F-statistic: 89.64 on 2 and 18 degrees of freedom, the p-value is 4.382e-10
```

```
Correlation of Coefficients:
```

	(Intercept)	Air.Flow
Air.Flow	-0.3104	
Water.Temp	-0.3438	-0.7819

Offenbar haben wir mit dem reduzierten Modell praktisch dieselbe Qualität des Fits erreicht wie mit dem komplizierteren Modell: Alle Design-Variablen liefern einen signifikanten Beitrag und die Residuen-Standardabweichung (`Residual standard error`) ist sogar leicht von 3.243 auf 3.239 gefallen, während der (multiple) R^2 -Wert (`R-squared`) nur leicht von 0.9136 auf 0.9088 gefallen ist. Auch die (nicht gezeigten) Diagnose-Plots unterstützen das einfachere Modell.

1.6.4 Das Hinzufügen eines Terms: `add1()`

Hier betrachten wir erneut das Ozon-Problem aus Abschnitt 1.1. Dort hatten wir eine einfache lineare Regression von `ozone` an `temperature` durchgeführt und das resultierende `lm`-Objekt mit `ozone.fit` bezeichnet. Der Data Frame `air` enthält aber noch weitere mögliche Design-Variablen, deren paarweisen Streudiagramme (nicht gezeigt) eine annähernd lineare Beziehung zwischen `ozone` und sowohl `radiation` als auch `wind` zeigen.

Wir wollen für jede der beiden Variablen den Effekt ihres Hinzufügens zum o. g. Ausgangsmodell bestimmen. Dies wird durch die Funktion `add1()` ermöglicht. Sie benötigt als erstes Argument das `lm`-Objekt mit dem gefitteten Ausgangsmodell, also `ozone.fit`. Das

zweite Argument ist eine Formel, die den Umfang der möglichen Modellergänzungen spezifiziert. Die Formel braucht *keine* „linke Seite“, d. h. Response; diese wird automatisch aus dem Ausgangsmodell entnommen. Eine Formel der Art `~ . + wind + radiation` gibt beispielsweise an, dass abwechselnd eine der Variablen `wind` und `radiation` zum Modell hinzugefügt werden soll (siehe die Tabelle unten).

Das Entscheidungskriterium für eine mögliche Verbesserung des Modells durch Hinzufügen eines Terms ist analog zu dem zur Entfernung eines Terms: Tritt unter den `Cp`-Werten, d. h. den AIC_p -Werten der erweiterten Modelle ein Wert auf, der kleiner als derjenige des Ausgangsmodells (in Zeile `<none>`) ist, so ist die Variable mit dem kleinsten dieser `Cp`-Werte hinzuzufügen. (Ist der `Cp`-Wert des Ausgangsmodells der kleinste, so kann das Modell durch Hinzufügen einer Variable nicht verbessert werden.)

Im vorliegenden Beispiel scheint es sinnvoll zu sein, die Variable `wind` hinzuzufügen (was wir hier aber nicht durchführen).

Hinzufügen von Modelltermen mittels <code>add1()</code> :			
<code>> add1(ozone.fit, ~ . + wind + radiation)</code>		Für jede der in der Formel <code>~ . + wind + radiation</code> auftretenden Variablen wird der Effekt ihres einzelnen Hinzufügens zu dem in <code>ozone.fit</code> bereits bestehenden und durch <code>.</code> symbolisierten Modell bestimmt.	
Single term additions		Resultat: Das Ausgangsmodell (<code>Model</code>), eine (Art) ANOVA-Tabelle, in deren Zeile <code><none></code> die Residuenquadratsumme (RSS) und die AIC_p -Statistik (<code>Cp</code>) des Ausgangsmodells angegeben werden. Die weiteren Zeilen enthalten für die jeweils genannte Variable die zu ihr gehörigen Freiheitsgrade (<code>df</code>), ihre Reduktion (Sum of Sq) der Residuenquadratsumme (RSS) des Ausgangsmodells sowie den zum erweiterten Modell gehörenden Wert der AIC_p -Statistik (<code>Cp</code>).	
Model:			(Es gilt also: „RSS(mit Variable) = RSS(<code><none></code>) – Sum of Sq(Variable)“.)
<code>ozone ~ temperature</code>			
	Df	Sum of Sq	RSS
<code><none></code>			37.74698
<code>wind</code>	1	5.839621	31.90736
<code>radiation</code>	1	3.839049	33.90793
		Cp	
		39.13219	
		33.98517	
		35.98575	

Zurückblickend erscheint die Wahl von `temperature` als Ausgangsvariable im einfachen linearen Modell als möglicherweise willkürlich. Eine (rein statistisch) eventuell besser begründbare Vorgehensweise ist die folgende: Ausgehend von dem auch *Null-Modell* genannten Modell, das nur β_0 enthält (also nur den so genannten “intercept term”), erlaubt uns die Funktion `add1()` mittels des C_p -Kriteriums die begründbare Auswahl einer Startvariablen:

```
> ozone.fit0 <- lm( ozone ~ 1, data= air)
> add1( ozone.fit0, ~ . + temperature + wind + radiation)
Single term additions

Model:
ozone ~ 1
```

	Df	Sum of Sq	RSS	Cp
<none>			87.20876	88.79437
temperature	1	49.46178	37.74698	40.91821
wind	1	31.28305	55.92571	59.09694
radiation	1	15.53144	71.67732	74.84855

Und siehe da: `temperature` wäre auch auf diese Weise zur ersten Wahl geworden.

Eine typische Anwendung für `add1()` ist die Auswahl geeigneter Transformationen für Design-Variablen, wie sie in Abschnitt 1.5.3 diskutiert wurden. Angenommen, für eine Covariable, die sich in ihrer „Reinform“ `x` bereits im Modell `some.model` befindet, werden mehrere Transformationen als Ersatz für `x` in Erwägung gezogen. Man kann sie alle mit Hilfe von `add1()` „durchprobieren“ lassen und die nach der C_p -Statistik (vgl. Abschnitt 1.6.2) am besten geeignete auswählen. Dazu wird `x` zunächst aus dem Modell `some.model` eliminiert und im reduzierten Modell mit den fraglichen Transformationen verglichen:

```
> some.model.minusx <- update( some.model, ~ . - x)
> add1( some.model.minusx, ~ . + x + x^2 + exp( x) + log( x) + I(1/x))
....
```

Beachte: Grundsätzlich sollten bei der Modellierung *fachspezifische* Überlegungen die entscheidende Rolle bei Auswahl und Transformation von Regressionsvariablen spielen!

1.7 Modelldiagnose II: Ausreißer, Extrempunkte, einflussreiche Punkte und Residual-Analyse

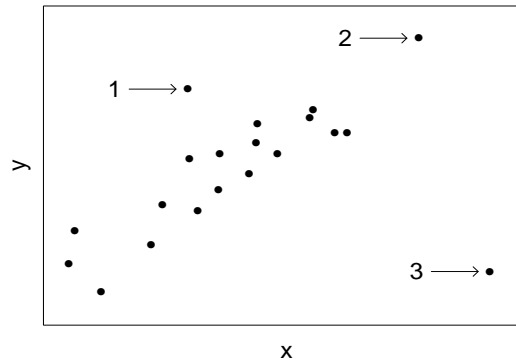
Wir liefern zunächst eine Beschreibung der in der Überschrift genannten Konzepte. Eine strenge Definition im mathematischen Sinn ist nicht existent, da es *das* Kriterium zur Bestimmung beispielsweise eines Extrempunktes nicht gibt.

„Definition“: Ein Punkt (x_i, Y_i) ist ein ...	
Ausreißer,	wenn sein Design-Wert x_i im „üblichen“ Bereich liegt (d. h. im Bereich der restlichen x_j für $j \neq i$), aber der Response-Wert Y_i „zu groß“ oder „zu klein“ ist (relativ zu den Y -Werten von Design-Punkten in der Umgebung von x_i).
Extrempunkt,	wenn sein Design-Wert x_i deutlich verschieden vom Rest der x_j , $j \neq i$ ist. (Bei der einfachen linearen Regression lässt sich das durch $ x_i - \bar{x}_n \gg 0$ feststellen.) Er wird auch „leverage point“ oder Hebelpunkt genannt.
einflussreicher Punkt,	wenn er ein Ausreißer oder Extrempunkt ist, dessen Einbeziehung in die Regression eine erhebliche Änderung der Parameterschätzwerte, der RSS oder der gefitteten Werte zur Folge hat.

Es stellt sich das Problem der Identifikation einflussreicher Punkte. Hier gibt es in beschränktem Maße nur für einfache Modelle grafisch-qualitative Verfahren und für komplexere Modelle nur quantitative Verfahren. Viele der in den folgenden Abschnitten aufgeführten quantitativen Verfahren findet man z. B. in Hocking (1996) beschrieben.

1.7.1 Grafische Identifizierung

Im Fall der einfachen linearen Regression sind obige Kriterien hinsichtlich ihres Erfülltseins oder Nicht-Erfülltseins anhand eines Streudiagramms der Y_i gegen die x_i möglicherweise noch gut beurteilbar, wie der folgende Plot veranschaulicht.



Der Punkt mit der Ziffer ...

- 1 ist offenbar ein Ausreißer bezüglich seines Y -Wertes, wohingegen sein x -Wert völlig im Rahmen der übrigen liegt.

Es stellt sich die Frage, ob hier ein Fehler in der Messung (oder Übertragung oder Dateneingabe) eines *zu kleinen* x -Wertes vorliegt, der dazu führt, dass der Y -Wert hervorsteht, denn am rechten Rand der Punktwolke würde der Y -Wert nicht sonderlich auffallen. Oder haben wir es hier mit einem korrekten x -Wert, aber einem Fehler in der Y -Messung zu tun? In jedem Fall ist eine Datenkontrolle angeraten.

Nichtsdestotrotz wird der Einfluss dieses Datums auf die gefittete Regressionsgerade nicht allzu groß sein, da es noch einige weitere Beobachtungen mit einem ähnlichen x -Wert gibt, die die Regressionsgerade von diesem einen Datum nicht zu sehr beeinflusst sein lassen werden.

- 2 ist ungewöhnlich im Hinblick auf seinen x -Wert, aber nicht so sehr hinsichtlich seines Y -Wertes. Er ist zwar ein Extrempunkt, aber kein einflussreicher Punkt, da er mit der Regressionsbeziehung der übrigen Beobachtungen in Einklang steht: Er liegt gewissermaßen „in der Verlängerung“ der Punktwolke.

Dennoch sollte geklärt werden, wie es zu der Lücke zwischen dieser Beobachtung und den restlichen Punkten gekommen ist. Außerdem stellt sich die Frage nach der Adäquatheit des Modells im „leeren“ Design-Bereich zwischen diesem x -Wert und den restlichen Design-Werten.

- 3 ist offensichtlich ein Ausreißer und ein Extrempunkt. Darüberhinaus ist er äußerst einflussreich auf die Lage der Regressionsgerade!

Es bleibt jedoch zu bedenken, dass es sich um eine korrekte Beobachtung handeln kann, die möglicherweise das Modell in Frage stellt! Das Problem hierbei ist, dass zu wenige Daten in der Umgebung von Punkt 3 vorliegen, um eine Klärung der Frage der Modell-Adäquatheit zu erreichen.

Die in obigem Plot veranschaulichte grafische Analyse ist im einfachen linearen Regressionsmodell und auch noch in einem Modell mit zwei Design-Variablen möglich, aber **für höherdimensionale Modelle ist sie viel schwieriger bzw. unmöglich**. Wie bereits erwähnt, existieren aber auch einige quantitative Hilfsmittel zur Identifizierung einflussreicher Punkte und zur Quantifizierung deren Einflusses. Diese Methoden stehen auch für komplexere Modelle zur Verfügung und sind Inhalt der nächsten Abschnitte.

1.7.2 Inferenzstatistische Residualanalyse

Wie bereits in Abschnitt 1.5 erwähnt, sind die Residuen $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$ ($i = 1, \dots, n$) unter den üblichen Modellannahmen auch **normalverteilt, allerdings weder unabhängig noch identisch**, sondern mit

$$\text{Var}(\hat{\varepsilon}_i) = \sigma^2 (1 - h_{ii}) \quad \text{und} \quad \text{Cov}(\hat{\varepsilon}_i, \hat{\varepsilon}_j) = -\sigma^2 h_{ij} \quad \text{für } 1 \leq i \neq j \leq n.$$

Um die Residuen wenigstens approximativ auf die Varianz 1 zu skalieren und damit untereinander besser vergleichbar zu machen, wird die folgende Transformation vorgeschlagen:

Intern studentisierte Residuen (auch: standardisierte Residuen):

Definition: Die intern studentisierten Residuen werden definiert durch

$$\hat{\varepsilon}_{i, \text{ int. stud.}} := \frac{\hat{\varepsilon}_i}{\hat{\sigma} \sqrt{1 - h_{ii}}} \quad \text{für } i = 1, \dots, n.$$

Verteilung: Unter der Modellannahme unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilter ε_i im linearen Regressionsmodell (der Dimension p) ist

$$(\hat{\varepsilon}_{i, \text{ int. stud.}})^2 \sim (n - p) \beta \left(\frac{1}{2}, \frac{n - p - 1}{2} \right) \quad \text{für } i = 1, \dots, n.$$

Dabei steht $\beta(r, s)$ für die Beta-Verteilung mit r und s Freiheitsgraden. Für großes n ist $\hat{\varepsilon}_{i, \text{ int. stud.}}$ approximativ $\mathcal{N}(0, 1)$ -verteilt.

Bemerkung: Ein schlecht gefittetes Datum Y_i bläht $\hat{\sigma}$ auf und reduziert $\hat{\varepsilon}_{i, \text{ int. stud.}}$ dadurch. Man könnte sagen, Y_i versucht sich vor seiner Entdeckung zu schützen. Um diesen Effekt zu kompensieren, wird eine alternative Betrachtung vorgeschlagen: Man bestimmt das Residuum zu Y_i für den Fit, den man erhält, wenn (x_i, Y_i) aus dem Datensatz *ausgeschlossen* wird. Dies sind so genannte ...

Extern studentisierte Residuen:

Definition: Die extern studentisierten Residuen werden definiert durch

$$\hat{\varepsilon}_{i, \text{ ext. stud.}} := \frac{Y_i - \hat{Y}_i^{(-i)}}{\sqrt{\widehat{\text{Var}}(Y_i - \hat{Y}_i^{(-i)})}} \quad \text{für } i = 1, \dots, n.$$

Dabei ist $\hat{Y}_i^{(-j)}$ der Wert der auf der Basis der Daten **ohne** (x_j, Y_j) gefitteten Regressionsfunktion an der Stelle x_i und $\widehat{\text{Var}}(Y_i - \hat{Y}_i^{(-i)})$ ein geeigneter Schätzer für die Varianz der Differenz $Y_i - \hat{Y}_i^{(-i)}$.

(Zur Berechnung von $\hat{\varepsilon}_{i, \text{ ext. stud.}}$ siehe „Berechnungsvereinfachung“ auf der nächsten Seite.)

Verteilung: Unter der Modellannahme unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilter ε_i im linearen Regressionsmodell (der Dimension p) ist

$$\hat{\varepsilon}_{i, \text{ ext. stud.}} \sim t_{n-p-1} \quad \text{für } i = 1, \dots, n.$$

Damit steht ein **Niveau- α -Test auf Ausreißereigenschaft** für ein *im Voraus spezifiziertes* Residuum zur Verfügung:

$$\text{Beobachtung } i \text{ ist ein Ausreißer} \iff |\hat{\varepsilon}_{i, \text{ ext. stud.}}| > t_{n-p-1; 1-\alpha/2}.$$

Problem: Um zu entdecken, ob überhaupt *irgendein* Residuum einen Ausreißer darstellt, werden in Wirklichkeit *alle* Residuen gleichzeitig miteinander verglichen und nicht ein einzelnes, vorher bestimmtes betrachtet. Das bedeutet, dass simultane Inferenz betrieben werden müsste. Erschwerend kommt hinzu, dass die Residuen nicht unabhängig sind.

Als **approximative Lösung** für dieses Problem wird eine Bonferroni-Methode vorgeschlagen (Weisberg, 1985):

$$\text{Beobachtung } i \text{ ist unter } n \text{ anderen ein Ausreißer} \iff |\hat{\varepsilon}_{i, \text{ ext. stud.}}| > t_{n-p-1; 1-\alpha/(2n)}.$$

Als vereinfachender **Kompromiss** für ein Indiz einer potenziellen Ausreißereigenschaft gilt:

$$\text{Beobachtung } i \text{ ist unter } n \text{ anderen ein Ausreißer} \iff |\hat{\varepsilon}_{i, \text{ ext. stud.}}| > 3.$$

Die gleiche Strategie wird für die intern studentisierten Residuen $\hat{\varepsilon}_{i, \text{ int. stud.}}$ verwendet.

Berechnungsvereinfachung: Man kann zeigen, dass zur Berechnung der extern studentisierten Residuen $\hat{\varepsilon}_{i, \text{ ext. stud.}}$ glücklicherweise *nicht* für jeden der n jeweils um ein Datum reduzierten Datensätze eine Regressionsfunktion zu fiten ist. Es gilt vielmehr die Beziehung

$$\hat{\varepsilon}_{i, \text{ ext. stud.}} = \frac{\hat{\varepsilon}_i}{\hat{\sigma}^{(-i)} \sqrt{1 - h_{ii}}},$$

wobei $\hat{\sigma}^{(-i)}$ der Schätzer für σ auf der Basis des um (x_i, Y_i) reduzierten Datensatzes ist, für den im linearen Regressionsmodell (der Dimension p) gilt:

$$(\hat{\sigma}^{(-i)})^2 = \frac{n-p}{n-p-1} \hat{\sigma}^2 - \frac{1}{n-p-1} \frac{\hat{\varepsilon}_i^2}{1-h_{ii}} = \frac{\hat{\sigma}^2}{n-p-1} \left(n-p - \hat{\varepsilon}_{i, \text{ int. stud.}}^2 \right).$$

1.7.3 Quantitative Identifizierung einflussreicher Punkte und Quantifizierung ihres Einflusses

Die zu Beginn von Kapitel 1 schon angesprochene Projektionsmatrix $H \equiv X(X'X)^{-1}X'$ spielt im Folgenden eine entscheidende Rolle. Ihre Elemente werden mit h_{ij} für $1 \leq i, j \leq n$ bezeichnet. Man kann zeigen, dass für die Diagonal-Elemente gilt

$$\frac{1}{n} \leq h_{ii} \leq 1, \quad i = 1, \dots, n,$$

und für die gefitteten Werte

$$\hat{Y}_i = h_{ii}Y_i + \sum_{1 \leq j \neq i \leq n} h_{ij}Y_j, \quad i = 1, \dots, n.$$

Auf Grund dieser Darstellung werden die Elemente der Projektionsmatrix H auch "leverage values" (= Hebelwerte) genannt, da einerseits die Diagonal-Elemente h_{ii} beeinflussen, wie sehr \hat{Y}_i in die Nähe von Y_i „gehebelt“ wird, und andererseits die übrigen h_{ij} für $j \neq i$ mitbestimmen, wie weit \hat{Y}_i von Y_i „weg-gehebelt“ wird.

Wir geben zwei sehr vereinfachende **Kriterien für die quantitative Identifizierung** einflussreicher Punkte an:

- Der "leverage value" oder Hebelwert h_{ii} :

Im linearen Regressionsmodell (der Dimension p) gilt (x_i, Y_i) als ein potenziell einflussreicher Punkt, falls die Abschätzung

$$h_{ii} > \frac{2p}{n}$$

erfüllt ist.

- Wie bereits gesagt, gilt für die gefitteten Werte

$$\hat{Y}_i = h_{ii}Y_i + \sum_{1 \leq j \neq i \leq n} h_{ij}Y_j.$$

Damit ist für $h_{ii} \approx 1$ auch \hat{Y}_i nahe bei Y_i . Dies liegt daran, dass im Fall $h_{ii} \approx 1$ die übrigen h_{ij} für $j \neq i$ klein (im Sinne von nahe Null) sind.

Y_i ist in diesem Fall ein Wert, der die Regressionsfunktion „in seine Nähe zwingt“, und (x_i, Y_i) somit ein potenziell einflussreicher Punkt.

Die **Quantifizierung des Einflusses eines Punktes** kann auf mehrere Arten erfolgen. Wir betrachten hier einige verschiedene Methoden, die quantifizieren, wie groß der Einfluss eines Punktes (x_i, Y_i) auf

1. die gefitteten Werte $\hat{Y}_1, \dots, \hat{Y}_n$,
2. die Parameterschätzwerte $\hat{\beta}_k$ und/oder
3. den Schätzer $\hat{\sigma}^2$ für σ^2

ist.

1.7.3.1 Einfluss eines Punktes auf $\hat{Y}_1, \dots, \hat{Y}_n$.

- **Cook's Abstand** D_i ist ein geeignet gewichteter Abstand zwischen dem *Vektor* der gefitteten Werte \hat{Y}_j beim Fit mit dem gesamten Datensatz und dem der gefitteten Werte $\hat{Y}_j^{(-i)}$ ohne das Datum (x_i, Y_i) :

$$D_i := \frac{(\hat{Y} - \hat{Y}^{(-i)})'(\hat{Y} - \hat{Y}^{(-i)})}{p \hat{\sigma}^2} = \frac{\sum_{j=1}^n (\hat{Y}_j - \hat{Y}_j^{(-i)})^2}{p \hat{\sigma}^2} = \frac{h_{ii} \hat{\varepsilon}_i^2}{p \hat{\sigma}^2 (1 - h_{ii})^2}.$$

Beachte: D_i ist groß, falls $\hat{\varepsilon}_i \gg 0$ oder falls $h_{ii} \approx 1$ ist.

Es gibt mehrere Vorschläge, wie Cook's Abstand für die Entscheidung, ob (x_i, Y_i) ein einflussreicher Punkt ist, herangezogen werden soll: (x_i, Y_i) ist einflussreich, falls

$$D_i > \begin{cases} F_{p, n-p; 1-\alpha} & \text{für } \alpha = 0.1 \text{ (Cook) bzw. für } \alpha = 0.5 \text{ (Weisberg);} \\ 1 & \text{für } n \text{ groß; entspricht } \alpha = 0.5; \\ \text{Rest der Cook-} & \longleftarrow \text{In der Praxis verwendet.} \\ \text{Abstände} & \end{cases}$$

- **DFFITS** (= Differenzen zwischen den **FITS** (= gefittete Werte) an den Stellen x_i beim Fit mit und ohne Datum (x_i, Y_i)):

$$\text{DFFITS}_i := \frac{\hat{Y}_i - \hat{Y}_i^{(-i)}}{(\hat{\sigma}^{(-i)})^2 h_{ii}} = \sqrt{p \frac{\hat{\sigma}^2}{(\hat{\sigma}^{(-i)})^2}} D_i = \sqrt{\frac{h_{ii} \hat{\epsilon}_i^2}{(\hat{\sigma}^{(-i)})^2 (1 - h_{ii})^2}},$$

wobei D_i Cook's Abstand ist und $(\hat{\sigma}^{(-i)})^2$ wie zuvor der Schätzer für σ^2 auf der Basis des um (x_i, Y_i) reduzierten Datensatzes. Damit sind hier die analogen Kriterien wie oben bei D_i anzulegen.

1.7.3.2 Einfluss eines Punktes auf $\hat{\beta}_k$.

- Ebenfalls **Cook's Abstand** D_i : Für die p -dimensionalen Vektoren $\hat{\beta}$ und $\hat{\beta}^{(-i)}$ gilt:

$$D_i = \frac{(\hat{\beta} - \hat{\beta}^{(-i)})' X' X (\hat{\beta} - \hat{\beta}^{(-i)})}{p \hat{\sigma}^2},$$

wobei X die Design-Matrix ist und $\hat{\beta}^{(-i)}$ analog zu $\hat{Y}^{(-i)}$ zu Stande kommt.

D_i ist somit ein spezielles Abstandsmaß für die Vektoren $\hat{\beta}$ und $\hat{\beta}^{(-i)}$ (in der „ $X'X$ -Metrik“). Dies war übrigens Cook's ursprüngliche Definition für D_i .

- **DFBETAS** (= Differenz zwischen den **BETAS** (= geschätzte Parameter) beim Fit mit und ohne Datum (x_i, Y_i)).

$$\text{DFBETAS}_{ki} := \frac{\hat{\beta}_k - \hat{\beta}_k^{(-i)}}{\hat{\sigma}^{(-i)} \sqrt{c_{k+1, k+1}}} \quad \text{für } k = 0, 1, \dots, p-1,$$

wobei c_{ll} das l -te Diagonalelement von $(X'X)^{-1}$ für $l = 1, \dots, p$ ist.

Im einfachen linearen Modell ist speziell für $k = 0, 1$:

$$c_{k+1, k+1} = \begin{cases} \frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x}_n)^2} & \text{für } k = 0 \\ 1 & \text{für } k = 1 \\ \frac{1}{\sum_{i=1}^n (x_i - \bar{x}_n)^2} & \text{für } k = 1 \end{cases}.$$

Ein großer Wert für $|\text{DFBETAS}_{ki}|$ deutet auf einen großen Einfluss von Y_i auf $\hat{\beta}_k$ hin. Faustregel: $|\text{DFBETAS}_{ki}|$ gilt als „groß“, falls

$$|\text{DFBETAS}_{ki}| > \begin{cases} 1 & \text{für „kleines“ bis „mittleres“ } n \\ \frac{2}{\sqrt{n}} & \text{für „großes“ } n \end{cases}.$$

1.7.3.3 Einfluss eines Punktes auf $\hat{\sigma}^2$.

Für das Verhältnis der Varianzschätzer des reduzierten Datensatzes $(\hat{\sigma}^{(-i)})^2$ und des vollen Datensatzes $\hat{\sigma}^2$ gilt:

$$\frac{(\hat{\sigma}^{(-i)})^2}{\hat{\sigma}^2} = \frac{n-p}{n-p-1} \left(1 - \frac{\hat{\varepsilon}_i, \text{ int. stud.}}{n-p} \right) \sim \frac{n-p}{n-p-1} \beta \left(\frac{n-p-1}{2}, \frac{1}{2} \right).$$

1.7.3.4 Zusammenfassung und Umsetzung in S-PLUS.

Für einen konkreten Datensatz erscheint es sinnvoll, obige Statistiken zusammenfassend in einer Tabelle der folgenden Art zu präsentieren oder in Grafiken, in denen zusätzlich die jeweils kritischen (Schwellen-) Werte eingezeichnet sind:

Nr. <i>i</i>	Intern studentisierte Residuen	Extern	Hebel- werte h_{ii}	Cook's D_i	DF- FITS $_i$	DFBETAS $_{ki}$			$\frac{(\hat{\sigma}^{(-i)})^2}{\hat{\sigma}^2}$
						$k = 0$...	$k = p-1$	
1	1.595	1.606	0.022	0.019	0.242	5.1	...	0.005	0.986
2	1.025	1.026	0.015	0.005	0.128	372.3		0.372	1.000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
$n-1$	-0.712	-0.710	0.012	0.002	-0.079	364.8		0.361	1.005
n	0.027	0.026	0.024	0.000	0.004	3496.9	...	3.479	1.009

Die S-PLUS-Funktionen (ab Version 3.4), die die Berechnung eines Teils obiger Statistiken durchführen bzw. bei deren Berechnung behilflich sein können, heißen `lm.influence()` und `ls.diag()`. (Dabei steht `ls.diag` für "least-squares diagnostics".) Im Zusammenhang mit diesen beiden Funktionen sind ein paar Besonderheiten zu beachten:

- Damit die Anwendung von `lm.influence()` auf ein `lm`-Objekt funktioniert, **muss bei dessen Erzeugung durch `lm()` die (bisher noch nicht erwähnte) Option `model= T` verwendet worden sein.**
- Die (vermutlich noch aus den Anfängen von S-PLUS stammende) Funktion `ls.diag()` kann *nicht* auf ein `lm`-Objekt angewendet werden. Vielmehr ist für ihre Verwendung zunächst ein Modell-Fit mit der (ebenfalls etwas älteren) Funktion `lsfit()` vorzunehmen. Aus deren zahlreiche Komponenten umfassenden Resultat-Liste kann `ls.diag()` die gewünschten Statistiken berechnen. (Für Details zu `lsfit()` siehe das folgende Beispiel oder die Online-Hilfe.)

Zur Demonstration der Arbeitsweise der obigen Diagnosefunktionen betrachten wir als **Beispiel** die bereits aus Abschnitt 1.2 bekannten Ozon-Daten, die in dem Data Frame `air` gespeichert sind: Wir fitten ein multiples lineares Modell von `ozone` an `temperature` und `radiation` sowohl mittels der Funktion `lm()` als auch mit der neu kennen gelernten (alten) Funktion `lsfit()`. Letztere ist *nicht* in der Lage die Formelsyntax von S-PLUS zu verarbeiten, sondern erwartet als Argumente die Design-Matrix X (ohne 1-Spalte) und den Vektor der Response Y :

```
> oz2.fit <- lm( ozone ~ temperature + radiation, air, model= T)
> oz2.lsfrit <- lsfit( cbind( air$temperature, air$radiation), air$ozone)
```

Die Funktion `lm.influence()` angewendet auf ein `lm`-Objekt (entstanden mit `model= T`) liefert eine Liste mit drei Komponenten zurück:

- In `coefficients` steckt eine $n \times p$ -Matrix, deren i -te Zeile den Parameterschätzwert für β bei Auslassung des Datums (x_i, Y_i) enthält. Also befindet sich $\hat{\beta}_k^{(-i)}$ in Spalte $k + 1$ und Zeile i .
- In `sigma` steht der Vektor der $\hat{\sigma}^{(-i)}$.
- In `hat` befindet sich der Vektor der Diagonal-Elemente h_{ii} der Projektionsmatrix H .

Beispiel:

```
> lm.influence( oz2.fit)
$coefficients:
  (Intercept) temperature  radiation
1  -2.238428  0.06541618  0.002105840
2  -2.183174  0.06456888  0.002179350
3  -2.128840  0.06414148  0.002124914
...
111 -2.154358  0.06434979  0.002143138

$sigma:
      1      2      3 ...    111
0.5562681 0.5601893 0.5595125 ... 0.5629338

$hat:
[1] 0.022183132 0.015423319 0.011205513 ... 0.023785204
```

Die Funktion `ls.diag()` angewendet auf das Resultat der Funktion `lsfit()` liefert weitere diagnostische Statistiken in einer Liste mit neun Komponenten:

- `std.dev` ist die Residuenstandardabweichung $\hat{\sigma} \equiv \sqrt{RSS/(n - p)}$.
- In `hat` sind die Diagonal-Elemente h_{ii} der Projektionsmatrix H .
- `std.res` enthält die standardisierten, d. h. intern studentisierten Residuen $\hat{\varepsilon}_i$, int. stud.
- In `stud.res` sind die extern studentisierten Residuen $\hat{\varepsilon}_i$, ext. stud.
- `cooks` beinhaltet die Cook-Abstände D_i .
- `dfits` enthält die DFFITS.
- In `correlation` steht die geschätzte $p \times p$ -Korrelationsmatrix $(\hat{\rho}_{kl})_{1 \leq k, l \leq p}$ des Parameterschätzers $\hat{\beta}$, wobei $\widehat{\text{cor}}(\hat{\beta}_{k-1}, \hat{\beta}_{l-1}) \equiv \hat{\rho}_{kl} = c_{kl} / \sqrt{c_{kk}c_{ll}}$ und c_{kl} das (k, l) -Element von $(X'X)^{-1}$ ist. (Die Covariablen-Bezeichnungen sind jedoch verloren.)
- `std.err` beinhaltet die geschätzten Standardabweichungen von $\hat{\beta}_0, \dots, \hat{\beta}_{p-1}$, also den Vektor $\hat{\sigma} \sqrt{\text{diag}((X'X)^{-1})}$ (allerdings als $p \times 1$ -Matrix).

- `cov.unscaled` ist $(X'X)^{-1}$.

Beispiel:

```
> ls.diag( oz2.lsfitt)
$std.dev:
[1] 0.5603234

$hat:
[1] 0.022183132  0.015423319  0.011205513 ... 0.023785204

$std.res:
[1] 1.59474338  1.02529335 -1.14432206 ... 0.02657297

$stud.res:
[1] 1.606369379  1.025538867 -1.145980505... 0.026449743

$cooks:
[1] 1.923206e-02  5.489128e-03  4.946534e-03 ... 5.734827e-06

$dfits:
[1] 0.2419513684  0.1283560301 -0.1219944540 ... 0.0041285943

$correlation:
                Intercept          X1          X2
Intercept  1.00000000 -0.9616310  0.04743269
          X1 -0.96163101  1.00000000 -0.29408764
          X2  0.04743269 -0.2940876  1.00000000

$std.err:
                [,1]
Intercept  0.4398302138
          X1  0.0058653467
          X2  0.0006132217

$cov.unscaled:
                Intercept          X1          X2
Intercept  6.161587e-01 -7.901503e-03  4.074767e-05
          X1 -7.901503e-03  1.095746e-04 -3.369073e-06
          X2  4.074767e-05 -3.369073e-06  1.197726e-06
```

Mit diesen Angaben lassen sich die restlichen, in obiger Tabelle aufgeführten diagnostischen Statistiken (genauer die *DFBETAS*) leicht selbst berechnen.

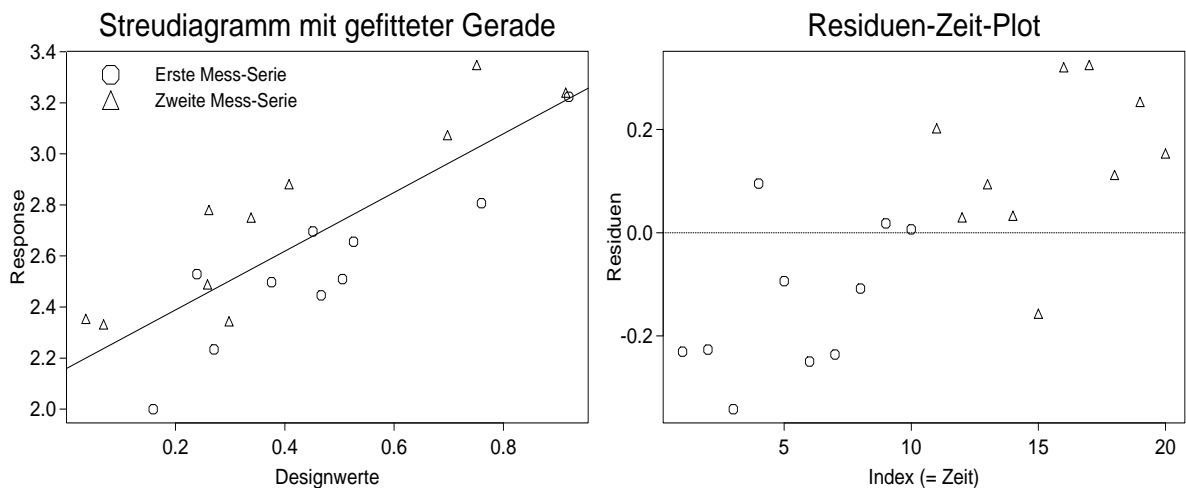
1.7.4 Zur Unabhängigkeitsannahme der Fehler

Die wesentliche **Grundannahme** in der linearen Regression ist, dass $\varepsilon_1, \dots, \varepsilon_n$ unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilt sind. Auf ihr basieren alle inferenzstatistischen Aussagen über die Parameterschätzer und die gefitteten Werte. Man weiß, die Inferenz in der Regressionsanalyse ist

- relativ robust gegenüber moderaten Abweichungen von der Normalverteilungsannahme, **aber**
- empfindlich gegenüber Abweichungen von den Annahmen der Unabhängigkeit oder der Varianzhomogenität!

Die zeitlich sequenzielle Erhebung der Y_i kann zu einer Korrelation der jeweils assoziierten Fehler führen. Dies kann mittels einer grafischen Analyse durch einen Plot der **Residuen** $\hat{\varepsilon}_i$ gegen ihre Indizes i , also **gegen die Zeit**, eventuell aufgedeckt werden.

Ein synthetisches **Beispiel** soll das veranschaulichen: Angenommen, es wurden zwei Mess-Serien für denselben Produktionsprozess bei verschiedenen Design-Werten zur Bestimmung seines Ertrages erstellt, und zwar eine frühe und eine späte. Bei der zweiten Mess-Serie war der Prozess schon „eingespielt“ und erzielte über den gesamten Design-Wertebereich hinweg tendenziell höhere Response-Werte. In dem einfachen linearen Regressionsmodell, das diesen zeitlichen Effekt nicht beachtet, wäre die im linken Plot präsentierte Regressionsgerade gefittet worden. Der Residuenplot gegen die Zeit deckt die dadurch entstandene zeitliche Korrelation der Fehler jedoch auf: Die Störungen sind positiv mit der Zeit korreliert, d. h., *nicht* unabhängig!



Natürlich sind auch andere zeitliche Abhängigkeitsstrukturen der Störungen möglich.

Tests: Es existieren auch statistische Tests, die es ermöglichen, die Unabhängigkeitsannahme zu prüfen. Einige gehen dabei von sehr konkreten Modellen für die Abhängigkeitsstruktur der Störungen aus. Ein Beispiel hierfür ist der Durbin-Watson-Test auf Korrelation, der ein autoregressives Schema 1. Ordnung für die ε_i zugrundelegt. Andere Tests sind nichtparametrischer Natur, wie beispielsweise der Runs-Test. Wir gehen hier nicht weiter auf derlei Tests ein.

1.8 Schätz- und Prognosewerte sowie Konfidenz- und Toleranzintervalle im linearen Regressionsmodell

Im Rahmen eines linearen Modells lässt sich, wie gesehen, der Parametervektor β der Regressionsfunktion recht einfach schätzen. Zumindest innerhalb des Beobachtungsbereichs des Designs existiert damit für jede Stelle ein guter Schätzer für den zugehörigen Regressionsfunktionswert: Es sind die Werte der gefitteten Regressionsfunktion an eben diesen Stellen. Außerdem lassen sich, wie wir sehen werden, unter den gemachten (Normal-) Verteilungsannahmen $(1 - \alpha)$ -**Konfidenzintervalle** für diese Regressionsfunktionswerte angeben. Letzteres gilt auch für die Komponenten des Parametervektors β .

Die Bedeutung der Regression liegt ferner darin, dass sie – vorausgesetzt das Modell ist und bleibt korrekt! – die Prognose *zukünftiger* Responses für gegebene Werte der Covariablen erlaubt: Es sind (auch dies) die Werte der gefitteten Regressionsfunktion an den gegebenen Design-Stellen. Zu diesen Prognosewerten können unter der Bedingung, dass die Verteilungsannahmen auch zukünftig gelten, $(1 - \alpha)$ -**Toleranzintervalle** berechnet werden.

Werden mehrere Design-Stellen betrachtet, so muss sowohl bei den Konfidenzintervallen als auch bei den Toleranzintervallen genau unterschieden werden, ob sie das gewählte Niveau $1 - \alpha$ jeweils **einzeln** (d. h. **punktweise**) oder **simultan** einhalten sollen. Zur Vollständigkeit und zur Erinnerung listen wir im Folgenden noch einmal die relevanten Formeln im Modell der multiplen linearen Regression auf. In Abschnitt 1.8.6 geben wir sie der Anschaulichkeit halber für das einfache lineare Regressionsmodell (die “straight line regression”) ganz explizit an.

Zur Wiederholung: Der Parametervektor $\beta' = (\beta_0, \beta_1, \dots, \beta_{p-1})$ ist p -dimensional und die Design-Matrix X ist eine $n \times p$ -Matrix. Der beste lineare unverzerrte Schätzer (Engl.: “best linear unbiased estimator”, kurz: BLUE) der Regressionsfunktion an der Stelle x ist $x'\hat{\beta}$ und der Schätzer für seine Standardabweichung $\sigma_{x'\hat{\beta}}$ (= “standard error of the fit”) lautet $\hat{\sigma}\sqrt{x'(X'X)^{-1}x}$, wobei $\hat{\sigma}^2 = \text{RSS}/(n - p)$ ist. (Zur Notation siehe nochmal den Anfang von Abschnitt 1 und nötigenfalls eine Vorlesung über lineare Modelle.)

In S-PLUS 6.0 steht die Funktion `predict()` zur Verfügung, mit deren Hilfe wir im linearen Modell eine gefittete Regressionsfunktion $x \mapsto x'\hat{\beta}$ an beliebigen Stellen auswerten können. Sie leistet auch die Berechnung entweder **punktweiser** oder **simultaner** Konfidenzintervalle (gemäß der Bonferroni- oder der Scheffé-Methode) bzw. eines **Konfidenzbandes** für die gesamte Regressionsfunktion oder eines **simultanen Konfidenzbereichs** für den Parametervektor β der Regressionsfunktion. (In S-PLUS 3.4 war `predict()` dazu noch nicht in der Lage.)

1.8.1 Schätzwerte für die Regressionsfunktion

Die Funktion `predict()` benötigt zur Auswertung einer geschätzten Regressionsfunktion als Argumente ein `lm`-Objekt und einen Data Frame, der zeilenweise die Covariablenwerte der Design-Stellen enthält, für die die Auswertung erfolgen soll. In diesem Data Frame müssen dieselben Variablennamen (d. h. Spaltennamen) auftreten wie in demjenigen, der zur Erzeugung des `lm`-Objekts verwendet wurde (allerdings auch *nur* die).

Wir werden zwei Beispielmodelle für den Ozon-Datensatz betrachten, den wir schon aus vorherigen Abschnitten kennen. Anhand dieser Modelle wird im Folgenden die Arbeitsweise von `predict()` beschrieben. Zur Übersicht und zu Referenzzwecken „erzeugen“ und dokumentieren wir diese unterschiedlich komplexen Modelle hier:

```
> oz1.lm <- lm( ozone ~ temperature + temperature^2, air)
> oz2.lm <- update( oz1.lm ~ . + radiation + radiation^2)

> summary( oz1.lm, cor= F)
....
Coefficients:
                Value Std. Error t value Pr(>|t|)
(Intercept)   5.5594   3.1356     1.7730  0.0790
temperature  -0.1358   0.0824    -1.6485  0.1022
I(temperature^2) 0.0013  0.0005     2.5090  0.0136

Residual standard error: 0.5747 on 108 degrees of freedom
Multiple R-Squared: 0.591
F-statistic: 78.03 on 2 and 108 degrees of freedom, the p-value is 0
```

```
> summary( oz2.lm, cor= F)
....
Coefficients:
                Value Std. Error t value Pr(>|t|)
(Intercept)   7.1578   2.9410     2.4338  0.0166
temperature  -0.1848   0.0775    -2.3845  0.0189
I(temperature^2) 0.0016  0.0005     3.1711  0.0020
  radiation    0.0070   0.0026     2.7471  0.0071
I(radiation^2) 0.0000   0.0000    -1.8995  0.0602

Residual standard error: 0.5343 on 106 degrees of freedom
Multiple R-Squared: 0.6531
F-statistic: 49.88 on 4 and 106 degrees of freedom, the p-value is 0
```

Offenbar hängt das Modell `oz1.lm`, obwohl sein Parametervektor β dreidimensional ist, nur von *einer* Covariablen, nämlich der Temperatur ab, da die beiden Terme `temperature` und `temperature^2` im Modell aus demselben Wert berechnet werden. Das andere Modell, `oz2.lm`, ist fünfdimensional, hängt jedoch nur von *zwei* Covariablen ab, nämlich von Temperatur und Strahlung. Auch hier werden alle Modellterme aus diesen beiden Werten abgeleitet. Will man eine geschätzte Regressionsfunktion auswerten, so braucht man also lediglich die den Modelltermen jeweils zugrundeliegenden Covariablen anzugeben, um den vollständigen Design-Vektor $x = (x_0, x_1, \dots, x_{p-1})'$ und schließlich $\hat{y}(x) = x'\hat{\beta}$ zu bestimmen.

Ist man an, sagen wir, k verschiedenen Design-Stellen $\tilde{x}_j = (\tilde{x}_{j0}, \tilde{x}_{j1}, \dots, \tilde{x}_{j,p-1})'$ für $j = 1, \dots, k$ interessiert, so ist es in S-PLUS notwendig, die dazugehörigen k Vektoren der Covariablenwerte als *Zeilen* erst in einen Data Frame zu „packen“ (dessen Variablennamen (= Spaltennamen), wie schon gesagt, so lauten müssen, wie in demjenigen, der zur Erzeugung des `lm`-Objekts verwendet wurde).

Beispiel: Wir wollen das Modell `oz2.lm` an den drei Temperatur-Strahlung-Wertepaaren (60, 200), (64, 250) und (68, 300) auswerten, haben die (marginalen) Temperatur- und Strahlungswerte jedoch in separaten Vektoren gespeichert:

```
> newtemp <- c( 60, 64, 68);      newrad <- c( 200, 250, 300)
```

Dann können diese beiden Vektoren durch Verwendung von

```
> data.frame( temperature= newtemp, radiation= newrad)
  temperature radiation
1           60      200
2           64      250
3           68      300
```

in einem Data Frame zusammengefasst werden, wie er von `predict()` benötigt wird.

Will man ein Modell, das auf mindestens zwei Covariablen beruht, auf einem vollständigen (endlichen) *Gitter* auswerten und hat man die dabei zu durchlaufenden (Marginal-) Werte der Covariablen als separate Vektoren gespeichert, dann erzeugt die Funktion `expand.grid()` aus letzteren das entsprechende Gitter (Engl.: “grid”).

Beispiel: Das Modell `oz2.lm` soll für alle Kombinationen der (marginalen) Temperaturwerte $T = \{50, 55, \dots, 105\}$ mit den (marginalen) Strahlungswerten $S = \{0, 20, \dots, 340\}$ ausgewertet werden, d. h. auf dem Gitter $T \times S$. Sind T und S als separate Vektoren á la

```
> newtemp2 <- seq( 50, 105, by= 5);      newrad2 <- seq( 0, 340, by= 20)
```

gespeichert, dann erhalten wir das gewünschte Gitter durch

```
> expand.grid( temperature= newtemp2, radiation= newrad2)
  temperature radiation
1           50         0
2           55         0
.....
12          105         0
13           50        20
14           55        20
.....
.....
204          105       320
205           50       340
.....
216          105       340
```

Offenbar durchläuft die erste Komponente des resultierenden Data Frames ihren Wertebereich vollständig, bevor die zweite Komponente auf ihren nächsten Wert „springt“. (Analog würde sich der Wert einer j -ten Komponente, falls vorhanden, erst auf den nächsten ändern, wenn sämtliche Kombinationen in den ersten $j - 1$ Komponenten durchlaufen sind, usw.)

Mit den eben eingeführten vier „Hilfsvektoren“ `newtemp`, `newrad`, `newtemp2` und `newrad2` wird `predict()` nun (auf der folgenden Seite) vorgestellt:

Schätzwerte für die Regressionsfunktion und grafische Darstellung

```
> oz1pred <- predict( oz1.lm,
+ newdata= data.frame( temperature=
+ newtemp2));      oz1pred
           1           2 ....          12
2.129198 2.155593 .... 6.113698
```

```
> attach( air)
> plot( temperature, ozone,
+ xlim=range( temperature, newtemp2),
+ ylim= range( ozone, oz1pred),
+ xlab= "Temperature",
+ ylab= "Ozone")
> lines( newtemp2, oz1pred)
> detach( "air")
```

```
> predict( oz2.lm, newdata=
+ data.frame( temperature= newtemp,
+ radiation= newrad))
           1           2           3
2.636228 2.720516 2.78531
```

```
> oz2pred <- predict( oz2.lm,
+ newdata= expand.grid(
+ temperature= newtemp2,
+ radiation= newrad2));      oz2pred
           1           2 ....          216
1.891848 1.802358 .... 6.043337
```

```
> zout <- persp( x= newtemp2,
+ y= newrad2,
+ z= matrix( oz2pred,
+           nrow= length( newtemp2)),
+ xlab= "Temperature",
+ ylab= "Radiation",
+ zlab= "Ozone",
+ zlim= range( oz2pred, air$ozone))
```

```
> points( persp(
+ x= air$temperature,
+ y= air$radiation,
+ z= air$ozone,
+ zout))
```

`predict()` wertet das Modell `oz1.lm` an den Stellen in `newtemp2` aus: Hierzu wird `newtemp2` unter dem Namen `temperature` in einen Data Frame gepackt und dieser dem `newdata`-Argument von `predict()` zugewiesen.

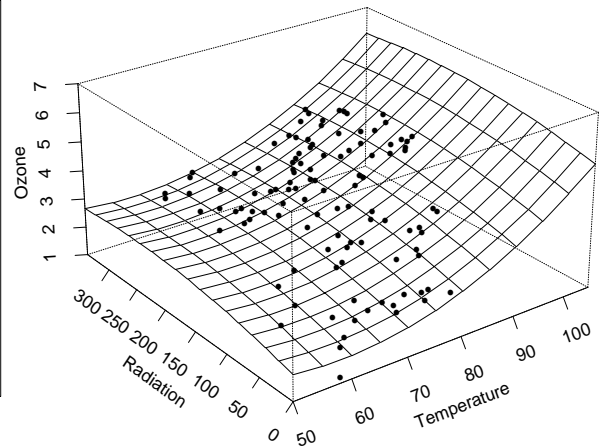
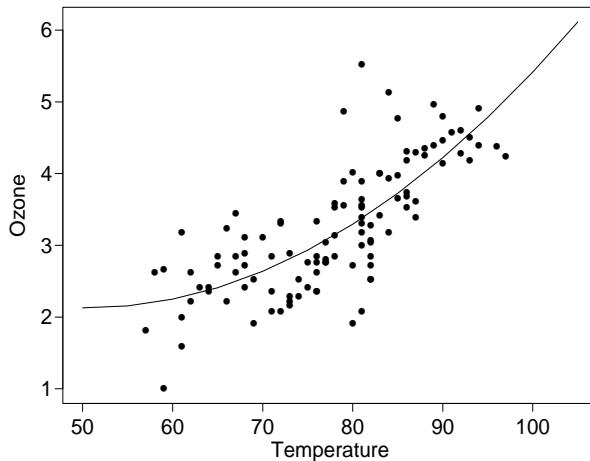
Plottet das Streudiagramm der beobachteten `temperature`- und `ozone`-Paare (x_i, y_i) so, dass sowohl diese als auch die „neuen“ Temperaturwerte \tilde{x}_j und die gefitteten Ozonwerte $\hat{y}(\tilde{x}_j)$ in die Achsenlimits passen. `lines` fügt die gefittete Regressionsfunktion als Polygonzug durch $(\tilde{x}_j, \hat{y}(\tilde{x}_j))$ hinzu. (Siehe Bild nächste Seite, oben links.)

Für das Modell `oz2.lm` benötigt `predict()` im `newdata` zugewiesenen Data Frame **alle** Covariablen des Modells (hier `temperature` und `radiation`). Die diesen Variablen zugewiesenen Vektoren (hier `newtemp` und `newrad`) müssen gleich lang sein.

Hier wird `oz2.lm` auf dem regulären Gitter `newtemp2` \times `newrad2` ausgewertet, und zwar in der Reihenfolge, in der dieses Gitter in dem durch `expand.grid()` erzeugten Data Frame (mit den Spaltennamen `temperature` und `radiation`) durchlaufen wird.

`persp()` generiert einen 3D-perspektivischen Plot von (linear interpolierten) Funktionswerten $z_{ij} = f(x_i, y_j)$ über einem 2D-Gitter. Die Argumente `x` und `y` müssen die (aufsteigend sortierten!) Marginalien dieses Gitters sein. `z` erwartet eine $(\text{length}(x) \times \text{length}(y))$ -Matrix, deren Element `z[i, j]` den Funktionswert $f(x[i], y[j])$ enthält.

Das Resultat (hier `zout`) kann dazu verwendet werden, mit Hilfe der Funktion `perspp()` weitere Punkte auf den 3D-Plot zu projizieren, wie hier, wo die Originalbeobachtungen dem Fit überlagert werden. (Bild nächste Seite, oben rechts.)



(Forts.: Schätzwerte für die Regressionsfunktion und graf. Darstellung)

<pre>> oz2hat <- predict(oz2.lm) > oz2hat 1 2 111 2.740264 2.726605 2.810383 > plot(oz2hat, air\$ozone) > abline(0, 1, lty= 2)</pre>	<p>Ohne eine Zuweisung an <code>newdata</code> werden die gefitteten Werte $\hat{y}_i (= \hat{y}(x_i))$ mit den Stellen x_i des <i>ursprünglichen</i> Designs) geliefert. Es ist also dasselbe wie <code>fitted(oz2.lm)</code> und kann z. B. für den (nicht gezeigten) Plot von y_i gegen \hat{y}_i zusammen mit der Identität als „Soll-Linie“ verwendet werden, um die Güte der Modellanpassung an die Daten qualitativ zu begutachten. (Dies entspricht dem dritten Plot, den man mit <code>plot(oz2.lm)</code> erhält.)</p>
<pre>> predict(oz2.lm, newdata= + data.frame(temperature= + newtemp, radiation= newrad), + se.fit= T) \$fit: 1 2 3 2.636228 2.720516 2.78531 \$se.fit: 1 2 3 0.1849758 0.126345 0.1183417 \$residual.scale: [1] 0.534264 \$df: [1] 106</pre>	<p>Das Argument <code>se.fit= T</code> veranlasst zusätzlich die Berechnung der Standardabweichungen (Engl.: “standard errors”) der Schätzwerte der Regressionsfunktion. Das Resultat von <code>predict()</code> ist jetzt eine Liste mit vier Komponenten. Die Komponente <code>fit</code> enthält den Vektor der Schätzwerte $\hat{y}(\tilde{x}_j)$ und die Komponente <code>se.fit</code> den Vektor der dazugehörigen Standardabweichungen. D. h., das j-te Element in <code>se.fit</code> ist gleich $\hat{\sigma} \sqrt{\tilde{x}'_j (X'X)^{-1} \tilde{x}_j}$.</p> <p>Die Komponente <code>residual.scale</code> birgt den Wert von $\hat{\sigma}$ und <code>df</code> enthält die zugehörigen Freiheitsgrade $n - p$.</p>

Vorsicht: `predict()` kann bei Modellen, deren Formeln datenabhängige Transformationen, wie `sqrt(x - mean(x))` oder `poly(x, 3)` enthalten, falsche Resultate liefern! (Was `poly(...)` bedeutet, wird in Abschnitt 1.9 noch vorgestellt.) Für solche Fälle ist die Funktion `predict.gam()` die „sichere“ Methode, worauf wir an dieser Stelle jedoch nicht näher eingehen.

1.8.2 Punktweise Konfidenzintervalle für die Regressionsfunktion

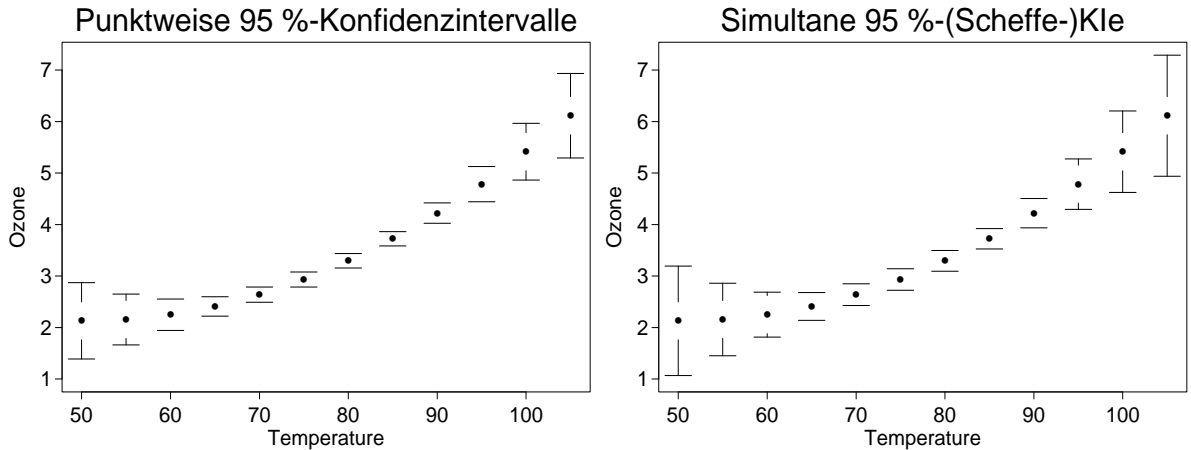
Das punktweise Konfidenzintervall (KI) zum Konfidenzniveau (KN) $1 - \alpha$ für den Wert der Regressionsfunktion an der Stelle x lautet

$$x'\hat{\beta} \pm t_{n-p;1-\alpha/2} \hat{\sigma} \sqrt{x'(X'X)^{-1}x} \quad \text{mit } \hat{\sigma} = \sqrt{\frac{\text{RSS}}{n-p}}.$$

Dabei ist $t_{n-p;1-\alpha/2}$ das $(1 - \alpha/2)$ -Quantil der t -Verteilung zu $n - p$ Freiheitsgraden.

Zur Berechnung dieses KIs ist `predict()` ebenfalls in der Lage. Eine grafische Darstellung ist allerdings nur für die einfache lineare Regression möglich bzw. im Fall von nur einer Covariablen. Hier kann eine Funktion namens `error.bar()` verwendet werden.

Punktweise Konfidenzintervalle für die Regressionsfunktion	
<pre>> oz1CIpw <- predict(oz1.lm, + newdata= data.frame(+ temperature= newtemp2), + ci.fit= T); oz1CIpw \$fit: 1 2 12 2.12920 2.15559 6.11370 \$ci.fit: lower upper 1 1.388050 2.870346 2 1.663124 2.648061 12 5.293363 6.934033 attr(,"conf.level"): [1] 0.95 > attach(oz1CIpw) > error.bar(x= newtemp2, + y= fit, + lower= ci.fit[,"lower"], + upper= ci.fit[,"upper"], + incr= F, + xlab= "Temperature", + ylab= "Ozone", + main= paste("Punktweise", + 100 * attr(ci.fit, + "conf.level"), + "%-Konfidenzintervalle")) > detach("oz1CIpw")</pre>	<p><code>predict()</code> mit dem Argument <code>ci.fit= T</code> bestimmt für die in <code>newdata</code> angegebenen Covariablenwerte \tilde{x}_j die Schätzwerte $\hat{y}(\tilde{x}_j)$ (in der Komponente <code>fit</code>) sowie Ober- und Untergrenzen der zugehörigen, punktweisen 95 %-KIs. Diese stehen in der Komponente <code>ci.fit</code> als eine zweispaltige Matrix mit den Spaltennamen <code>lower</code> und <code>upper</code>. Mit dem (nicht gezeigten) zusätzlichen Argument <code>conf.level</code> für <code>predict()</code> kann jedes andere KN gewählt werden. (Voreinstellung ist 95 %. Das aktuell verwendete KN ist als so genanntes <u>Attribut</u> der Matrix in <code>ci.fit</code> hinzugefügt und kann via <code>attr(oz1CIpw\$ci.fit, "conf.level")</code> abgefragt werden; s. u.)</p> <p>(Ohne Angabe des Arguments <code>newdata</code>, erhalte man die gefitteten Werte \hat{y}_i samt den Unter- und Obergrenzen ihrer zugehörigen KIs.)</p> <p>(<code>attach()</code> spart uns hier etwas Tipparbeit.)</p> <p><code>error.bar()</code> plottet an den in <code>x</code> angegebenen Stellen vertikale „Fehlerbalken“ um die Werte in <code>y</code>; hier also an den reellwertigen (!) Stellen \tilde{x}_j in <code>newtemp2</code> um die gefitteten Werte $\hat{y}(\tilde{x}_j)$, was nur im Fall <i>einer</i> Covariablen funktioniert. Die Unter- und Obergrenzen der Balken werden von <code>lower</code> bzw. <code>upper</code> erwartet und <code>incr= F</code> besagt, dass es sich dabei <i>nicht</i> um Inkremente bzgl. der <code>y</code>-Werte, sondern um absolute Limits handelt. Sie werden hier aus den Spalten der Komponente <code>ci.fit</code> von <code>oz1CIpw</code> genommen und liefern also punktweise KIs. Die restlichen Angaben (<code>xlab</code>, <code>ylab</code> und <code>main</code>) dienen dem Layout des Plots (nächste Seite oben links). Beachte auch, wie auf das Attribut <code>"conf.level"</code> von <code>ci.fit</code> zugegriffen wird.</p>



1.8.3 Simultane Konfidenzintervalle für die Regressionsfunktion und simultane Konfidenzbereiche für ihren Parametervektor

Der beste lineare unverzernte Schätzer der Regressionsfunktion an k Stellen $\tilde{x}_1, \dots, \tilde{x}_k$ ist $\tilde{x}'_j \hat{\beta}$ für $j = 1, \dots, k$ und zur Bestimmung von simultanen $(1 - \alpha)$ -KIn für die Regressionsfunktionswerte $\tilde{x}'_1 \beta, \dots, \tilde{x}'_k \beta$ an diesen Stellen gibt es zwei Methoden:

1. Die **Bonferroni-Methode**: Die simultanen KIe lauten:

$$\tilde{x}'_j \hat{\beta} \pm t_{n-p; 1-\alpha/(2k)} \hat{\sigma} \sqrt{\tilde{x}'_j (X'X)^{-1} \tilde{x}_j}, \quad \text{für } j = 1, \dots, k.$$

Beachte: Das $(1 - \alpha/2)$ - t -Quantil, wie es im Fall des punktweisen KIes verwendet wird, ist hier durch das $(1 - \alpha/(2k))$ - t -Quantil ersetzt! Dies muss in S-PLUS realisiert werden, indem das punktweise KN in `predict()` durch das Argument `conf.level` „von Hand“ auf $1 - \alpha/(2k)$ gesetzt wird. (Nicht gezeigt.)

2. Die **Scheffé-Methode**: Es sei d der Rang der $(p \times k)$ -Matrix $(\tilde{x}_1, \dots, \tilde{x}_k)$. D. h., d ist die Anzahl der *linear unabhängigen* Vektoren unter $\tilde{x}_1, \dots, \tilde{x}_k$ und es ist $d \leq \min\{p, k\}$. Die simultanen KIe lauten dann:

$$\tilde{x}'_j \hat{\beta} \pm \sqrt{dF_{d, n-p; 1-\alpha}} \hat{\sigma} \sqrt{\tilde{x}'_j (X'X)^{-1} \tilde{x}_j}, \quad \text{für } j = 1, \dots, k.$$

Beachte: Diese Methode ist nur für den Fall $d = p$ (also insbesondere $k > p$) in S-PLUS implementiert und wird durch das `predict()`-Argument `conf.type= "s"` (wie **s**imultan) aktiviert. (Voreinstellung ist `conf.type= "p"` wie **p**unktweise. Siehe hierzu auch den nächsten Abschnitt zum Konfidenzband.)

Bemerkungen:

- In beiden Fällen kann für $k = p$ und $\tilde{x}_j = (0, \dots, 0, 1, 0, \dots, 0)'$, dem j -ten p -dimensionalen Einheitsvektor für $j = 1, \dots, p$, ein **Konfidenzbereich** für den (ganzen) Parametervektor β angegeben werden. (Durch eine Auswahl von $k < p$ Einheitsvektoren geht dies auch für die entsprechenden Teile von β .)
- Die Intervall-Längen der Bonferroni-Methode und die der Scheffé-Methode sind unterschiedlich. Sinnvollerweise wählt man diejenige Methode, die die kürzeren Intervalle liefert, was in jeder gegebenen Situation auf einen Vergleich von $t_{n-p; 1-\alpha/(2k)}$ und $\sqrt{dF_{d, n-p; 1-\alpha}}$ hinausläuft.

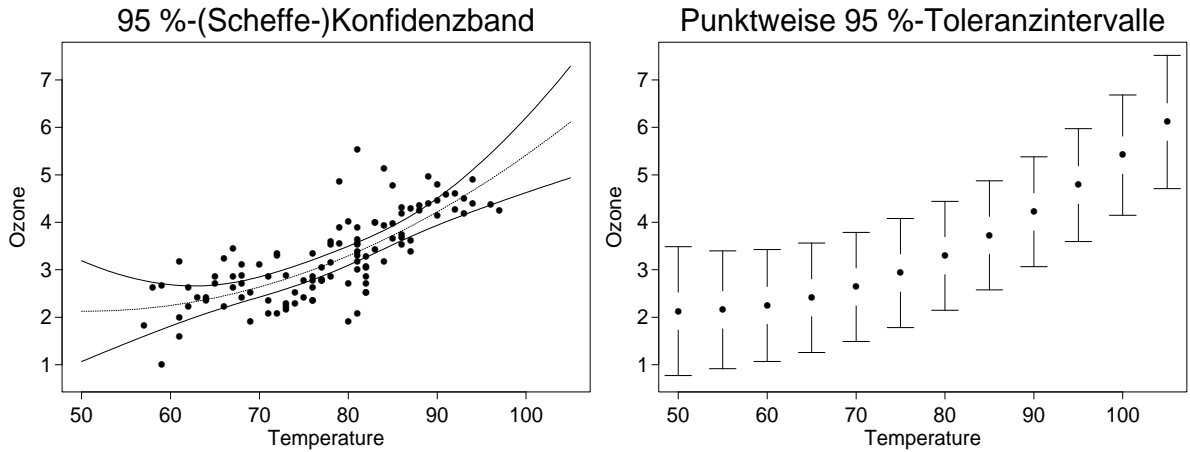
Simultane Konfidenzintervalle für die Regressionsfunktion	
<pre>> oz1CIsim <- predict(oz1.lm, newdata= + data.frame(temperature= newtemp2), + ci.fit= T, conf.type= "s"); oz1CIsim \$fit: 1 2 12 2.12920 2.15559 6.11370 \$ci.fit: lower upper 1 1.067272 3.191125 2 1.449978 2.861208 12 4.938311 7.289084 attr(,"conf.level"): [1] 0.95 > attach(oz1CIsim) > error.bar(x= newtemp2, y= fit, + lower= ci.fit[,"lower"], + upper= ci.fit[,"upper"], incr= F,....) > detach("oz1CIsim")</pre>	<p>predict() mit ci.fit= T und conf.type= "s" liefert für die in newdata angegebenen Covariablenwerte die Regressionsschätzwerte (in der Komponente fit) sowie Ober- und Untergrenzen der zugehörigen, simultanen 95 %-KIE. Diese stehen wieder als zweispaltige Matrix in der Komponente ci.fit (und sind erkennbar breiter als im punktweisen Fall).</p> <p>Mit conf.level wäre wieder jedes andere KN wählbar. (Voreinstellung ist 95 % und das aktuelle KN ist ein Attribut von ci.fit.)</p> <p>Die grafische Darstellung funktioniert völlig identisch zu der im punktweisen Fall. (Siehe Plot auf der vorherigen Seite oben rechts.)</p>

1.8.4 Ein Konfidenzband für die Regressionsfunktion

Aus der Scheffé-Methode erhält man auch das (simultane!) $(1 - \alpha)$ -Konfidenzband (KB) für die gesamte Regressionsfunktion, indem *alle* Design-Punkte betrachtet werden und $d = p$ gesetzt wird. Das KB lautet:

$$x \mapsto x' \hat{\beta} \pm \sqrt{p F_{p, n-p; 1-\alpha}} \hat{\sigma} \sqrt{x'(X'X)^{-1}x}.$$

Konfidenzband für die Regressionsfunktion	
<pre>> newtemp3 <- seq(50, 105, by= 0.5) > oz1CB <- predict(oz1.lm, newdata= + data.frame(temperature= newtemp3), + ci.fit= T, conf.type= "s") > attach(oz1CB) > plot(air\$temperature, air\$ozone, + xlim= range(air\$temperature, newtemp3), + ylim= range(air\$ozone, ci.fit), + xlab= "Temperature", ylab= "Ozone", + main= paste(100 * attr(ci.fit, + "conf.level"), + "%-(Scheffe-)Konfidenzband")) > lines(newtemp3, fit, lty= 2) > lines(newtemp3, ci.fit[,"lower"]) > lines(newtemp3, ci.fit[,"upper"]) > detach("oz1CB")</pre>	<p>Wir bestimmen Regressionsschätzwerte und ihre simultanen 95 % Scheffé-KIE für ein feines Gitter des Design-Bereichs und damit das KB. (Mit conf.level wäre ein anderes KN wählbar; Voreinstellung: 95 %.)</p> <p>Zur grafischen Darstellung werden zunächst die Originalbeobachtungen geplottet, dann (mit lines()) die gefittete Regressionsfunktion (gepunktet wg. lty= 2) und schließlich die Unter- bzw. Obergrenze des KBs (beide durchgezogen). (Siehe Plot auf der nächsten Seite oben links.)</p>



1.8.5 Punktweise und simultane Toleranzintervalle für zukünftige Response-Werte

Das punktweise $(1 - \alpha)$ -Toleranzintervall (TI) für einen *zukünftigen* Response-Wert (Prognosewert) an der Stelle x^* hat die Gestalt

$$x^{*'}\hat{\beta} \pm t_{n-p;1-\alpha/2} \hat{\sigma} \sqrt{1 + x^{*'}(X'X)^{-1}x^*}.$$

Beachte die 1 unter der Wurzel! Sie ist der Varianzbeitrag, der in der zukünftigen Response $Y^* = \beta_0 + \beta_1x^* + \varepsilon^*$ auf das Konto der *neuen* Störung ε^* geht.

Bemerkungen:

- Soll ein punktweises TI für das *arithmetische Mittel* \bar{Y}_m^* von m zukünftigen Responses Y_1^*, \dots, Y_m^* an *einer* Stelle x^* angegeben werde, so ist die 1 unter der Wurzel durch $1/m$ zu ersetzen. Dies ist *nicht* in S-PLUS implementiert. (Nicht gezeigt.)
- Für simultane TIE für die zukünftigen Responses an k *verschiedenen* Stellen x_1^*, \dots, x_k^* , von denen $d \leq \min\{k, p\}$ linear unabhängig sind, ist jedoch das Quantil $t_{n-p;1-\alpha/2}$ durch $\sqrt{dF_{d,n-p;1-\alpha}}$ zu ersetzen. Dies ist nur für $d = p$ (also $k > p$) in S-PLUS implementiert. (Nicht gezeigt.)

Punktwise Toleranzintervalle für zukünftige Response-Werte	
<pre>> oz1TIpw <- predict(oz1.lm, + newdata= data.frame(+ temperature= newtemp2), + pi.fit= T); oz1TIpw \$fit: 1 2 12 2.12920 2.15559 6.11370 \$pi.fit: lower upper 1 0.7701931 3.488203 2 0.9145774 3.396608 12 4.7099371 7.517458 attr(,"conf.level"): [1] 0.95</pre>	<p>predict() mit dem Argument pi.fit= T bestimmt für die in newdata angegebenen Covariablenwerte x_j^* die Prognosewerte $\hat{y}(x_j^*)$ der zukünftigen Response (in der Komponente fit) sowie Ober- und Untergrenzen der zugehörigen, punktweisen 95 %-TIE. Diese stehen in der Komponente pi.fit als zweispaltige Matrix mit den Spaltennamen lower und upper.</p> <p>Mit dem (nicht gezeigten) zusätzlichen Argument conf.level für predict() kann jedes andere TN gewählt werden. (Voreinstellung ist 95 %. Das aktuell verwendete TN ist als <u>Attribut</u> der Matrix in pi.fit hinzugefügt und kann via attr(oz1TIpw\$pi.fit, "conf.level") abgefragt werden; s. u.)</p>

(Forts.: Punktweise Toleranzintervalle für zukünftige Response-Werte)	
<pre>> attach(oz1TIpw) > error.bar(x= newtemp2, y= fit, + lower= pi.fit[,"lower"], + upper= pi.fit[,"upper"], incr= F, + xlab= "Temperature", + ylab= "Ozone", + main= paste("Punktweise", + 100 * attr(pi.fit, "conf.level"), + "%-Toleranzintervalle")) > detach("oz1TIpw")</pre>	<p>Die grafische Darstellung funktioniert genauso wie für die punktweisen KIE (siehe Seite 43 unten), nur hier mit Zugriffen auf die Komponente pi.fit. (Siehe den Plot auf der vorigen Seite oben rechts. Die dortigen beiden Plots zeigen zum besseren Vergleich denselben Ausschnitt der y-Achse.)</p> <p>Beachte die im Vergleich mit KIn erheblich größere Breite der TIE!</p>

1.8.6 Die Formeln im Spezialfall der einfachen linearen Regression

Obige Resultate für lineare Regressionsmodelle werden im Folgenden anhand des **einfachen linearen Regressionsmodells**

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad \text{für } i = 1, \dots, n$$

mit unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilten (i.i.d. $\sim \mathcal{N}(0, \sigma^2)$) Fehlern $\varepsilon_1, \dots, \varepsilon_n$ und unbekannter Varianz σ^2 detaillierter aufgeführt. Hier ist die Dimension p des Modells jetzt also 2. Zur Abkürzung sei $y(x) := \beta_0 + \beta_1 x$.

1.8.6.1 Konfidenzintervalle für die Parameter der Regressionsgeraden

<u>Konfidenzintervall (KI) zum Niveau $1 - \alpha$...</u>	
... für β_0 :	$\hat{\beta}_0 \pm t_{n-2; 1-\alpha/2} \hat{\sigma} \sqrt{\frac{\sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x}_n)^2}}$
... für β_1 :	$\hat{\beta}_1 \pm t_{n-2; 1-\alpha/2} \hat{\sigma} \frac{1}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$

Bemerkung: Die angegebenen KIE halten das Niveau $1 - \alpha$ für jeden der beiden Parameter jeweils einzeln zwar exakt ein, aber die Komponenten des Vektors $\beta' = (\beta_0, \beta_1)$ werden zum Niveau $1 - \alpha$ *nicht gleichzeitig* überdeckt! Um einen Konfidenzbereich für den Parametervektor als Ganzes anzugeben, muss eine der folgenden Methoden gewählt werden:

<u>(Simultaner) Konfidenzbereich zum Niveau $1 - \alpha$ für (β_0, β_1) nach der ...</u>	
... <u>Bonferroni-Methode:</u>	
$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$:	$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \pm t_{n-2; 1-\alpha/4} \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}_n)^2}} \begin{pmatrix} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \\ 1 \end{pmatrix}$
... <u>Scheffé-Methode:</u>	
$\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$:	$\begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} \pm \sqrt{2F_{2, n-2; 1-\alpha}} \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}_n)^2}} \begin{pmatrix} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \\ 1 \end{pmatrix}$

Bemerkungen:

- Beachte den Unterschied zwischen den t -Quantilen der Bonferroni-Methode und denen der „nicht-simultanen“ KIE: In ersteren steht $\alpha/4$ anstelle von $\alpha/2$ in letzteren.
- Die Intervall-Längen der Bonferroni-Methode und die der Scheffé-Methode sind, wie gesagt, unterschiedlich. Im vorliegenden Fall läuft es auf den Vergleich der Quantile $t_{n-2;1-\alpha/4}$ und $\sqrt{2F_{2,n-2;1-\alpha}}$ hinaus.

1.8.6.2 Konfidenzintervalle für die Regressionsgerade

Die gefittete Regressionsfunktion an der Stelle x stellt gemäß Definition einen Schätzer für den Erwartungswert der Response zur „Dosis“ x dar. Bei der Angabe eines KIs für diese Werte muss unterschieden werden, ob es sich um

- das KI für den Response-Erwartungswert an *einem* x -Wert,
- k simultane KIE für die Response-Erwartungswerte an k *verschiedenen* Werten $\tilde{x}_1, \dots, \tilde{x}_k$ oder
- ein Konfidenzband für *die Regressionsfunktion als Ganzes*

handeln soll. Hier die Formeln für jede der drei obigen Situationen:

Punktweises KI zum Niveau $1 - \alpha$ für $y(x) = \beta_0 + \beta_1 x$:

$$\hat{\beta}_0 + \hat{\beta}_1 x \pm t_{n-2;1-\alpha/2} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

Bemerkung: Das KI ist umso breiter, je weiter x von \bar{x}_n entfernt ist.

Simultane KIE zum Niveau $1 - \alpha$ für $y(\tilde{x}_1), \dots, y(\tilde{x}_k)$ gemäß der Bonferroni-Methode:

$$\hat{\beta}_0 + \hat{\beta}_1 \tilde{x}_j \pm t_{n-2;1-\alpha/(2k)} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(\tilde{x}_j - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}} \quad \text{für } j = 1, \dots, k.$$

Bemerkung: Die simultanen KIE sind jeweils umso breiter, je weiter \tilde{x}_j von \bar{x}_n weg ist.

Konfidenzband zum Niveau $1 - \alpha$ für die Regressionsgerade $x \mapsto y(x)$:

$$x \mapsto \hat{\beta}_0 + \hat{\beta}_1 x \pm \sqrt{2F_{2,n-2;1-\alpha}} \hat{\sigma} \sqrt{\frac{1}{n} + \frac{(x - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

Bemerkung: Die Breite des Bandes nimmt zu, je weiter sich x von \bar{x}_n entfernt.

1.8.6.3 Toleranzintervalle zukünftiger Response-Werte

Die gefittete Regressionsfunktion an der Stelle x^* stellt auch einen Schätzer für den Erwartungswert einer *zukünftigen* Response Y^* zur Dosis x^* dar. Es können auch Toleranzintervalle (TIE) für diese Responses angegeben werden. Hierbei muss jedoch unterschieden werden, ob es sich um

- das TI für *eine* zukünftige Response Y^* an der Stelle x^* ,
- das TI für das arithmetische Mittel \bar{Y}_m^* von m zukünftigen Responses Y_1^*, \dots, Y_m^* an *ein und derselben* Stelle x^* oder
- simultane TIE für die zukünftigen Responses Y_1^*, \dots, Y_k^* an $k \geq 2$ *verschiedenen* Stellen x_1^*, \dots, x_k^*

handelt. Es folgen die Formeln für die drei genannten Szenarien:

Annahme: Der Fehler ε^* in der zukünftigen Response $Y^* = \beta_0 + \beta_1 x^* + \varepsilon^*$ ist unabhängig von $\varepsilon_1, \dots, \varepsilon_n$ und ebenfalls $\mathcal{N}(0, \sigma^2)$ -verteilt mit demselben σ^2 .

Punktweises TI zum Niveau $1 - \alpha$ für Y^* an einer Stelle x^* :

$$\hat{\beta}_0 + \hat{\beta}_1 x^* \pm t_{n-2; 1-\alpha/2} \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

Bemerkungen:

- TIE sind immer länger als KIE, was an dem von der Prognosevarianz σ^2 (also von der Varianz von ε^* in Y^*) stammenden Summanden 1 unter der Wurzel liegt.
- Falls der SPn-Umfang n hinreichend groß ist und x^* innerhalb des ursprünglich beobachteten Design-Bereichs liegt, dominiert die 1 unter der Wurzel (also die Prognosevarianz) die beiden anderen Terme, so dass die Länge der TIE *wesentlich* größer als die Länge der KIE ist.
- Durch eine Erhöhung des SPn-Umfangs n kann die TI-Länge zwar reduziert werden, aber sie ist immer von der Ordnung $t_{n-2; 1-\alpha/2} \sigma$.
- Die fälschliche Verwendung der KIE anstelle von TIE für zukünftige Responses liefert offenbar eine nicht zutreffende Präzision für die Prognose.

Annahme: Die Fehler $\varepsilon_1^*, \dots, \varepsilon_m^*$ in den zukünftigen Responses $Y_j^* = \beta_0 + \beta_1 x^* + \varepsilon_j^*$ für $j = 1, \dots, m$ sind unabhängig von $\varepsilon_1, \dots, \varepsilon_n$ und ebenfalls (untereinander) i.i.d. $\sim \mathcal{N}(0, \sigma^2)$ mit demselben σ^2 .

Punktweises TI zum Niveau $1 - \alpha$ für das arithmetische Mittel \bar{Y}_m^* von m zukünftigen Responses Y_1^*, \dots, Y_m^* an *einer* Stelle x^* :

$$\hat{\beta}_0 + \hat{\beta}_1 x^* \pm t_{n-2; 1-\alpha/2} \hat{\sigma} \sqrt{\frac{1}{m} + \frac{1}{n} + \frac{(x^* - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}}$$

Bemerkung: Es treffen alle Bemerkungen von oben zu, allerdings mit $1/m$ an Stelle der 1 unter der Wurzel. Die TIE sind also abhängig von m etwas kürzer.

Annahme: Die Fehler $\varepsilon_1^*, \dots, \varepsilon_k^*$ in den zukünftigen Responses $Y_j^* = \beta_0 + \beta_1 x_j^* + \varepsilon_j^*$ für $j = 1, \dots, k$ sind unabhängig von $\varepsilon_1, \dots, \varepsilon_n$ und ebenfalls (untereinander) i.i.d. $\sim \mathcal{N}(0, \sigma^2)$ mit demselben σ^2 .

Simultane TIe zum Niveau $1 - \alpha$ für Y_1^*, \dots, Y_k^* an $k \geq 2$ verschiedenen Stellen x_1^*, \dots, x_k^* :

$$\hat{\beta}_0 + \hat{\beta}_1 x_j^* \pm \sqrt{2F_{2,n-2;1-\alpha}} \hat{\sigma} \sqrt{1 + \frac{1}{n} + \frac{(x_j^* - \bar{x}_n)^2}{\sum_{i=1}^n (x_i - \bar{x}_n)^2}} \quad \text{für } j = 1, \dots, k.$$

Bemerkungen:

- Auch simultane TIe sind immer länger als simultane KIe, was wieder an dem von der Prognosevarianz σ^2 stammenden Summanden 1 unter der Wurzel liegt.
- Falls der SPn-Umfang n hinreichend groß ist und die x_j^* innerhalb der ursprünglich beobachteten Design-Werte liegen, dominiert die 1 unter der Wurzel die beiden anderen Terme, so dass die Länge der simultanen TIe *wesentlich* größer als die Länge der simultanen KIe ist.
- Durch eine Erhöhung des SPn-Umfangs n können die TI-Längen zwar reduziert werden, aber sie sind immer von der Ordnung $\sqrt{2F_{2,n-2;1-\alpha}} \sigma$.
- Auch hier liefert die fälschliche Verwendung von KIn anstelle von TIn für zukünftige Responses eine nicht zutreffende Präzision für die Prognosen.

1.9 Polynomiale Regression

Wie gesehen, haben einige Operatoren in der Formelsyntax (wie `*`, `-`, `:`, `/` und `^m`) zwar eine spezielle Bedeutung, wenn sie rechts vom `~` auf der obersten Ebene einer Formel auftreten, aber die Variablen in einer Formel dürfen durch alle Funktionen transformiert werden, deren Resultat wieder als eine Variable interpretierbar ist. Dies trifft insbesondere auf alle mathematischen Funktionen wie `log()`, `sqrt()` etc. zu (wie z. B. bei der Anwendung linearisierender Transformationen auf S. 20). Aber auch Funktionen, deren Ergebnis als *mehrere* Variablen aufgefasst werden können, sind zulässig. Ein Beispiel hierfür ist die Funktion `poly()`: Sie erzeugt Basen von Orthonormalpolynomen bis zu einem gewissen Grad.

Bemerkung: Die eingangs genannten Operatoren haben *innerhalb* eines Funktionsaufrufs ihre „arithmetische“ Bedeutung. D. h., die aus zwei Variablen u und v abgeleitete Variable $x := \log(u + v)$ wird in S als `log(u + v)` formuliert. Sollen die Operatoren auf der obersten Formelebene ihre arithmetische Bedeutung haben, so sind die betroffenen Terme in die Funktion `I()` „einzupacken“. Damit also $x := u + v$ eine einzelne Covariable darstellt, ist in einer S-Formel der Ausdruck `I(u + v)` zu verwenden.

Beispiele: Das auf drei Covariablen x_1, x_2, x_3 basierende (synthetische) Modell

$$Y = \begin{pmatrix} 1 & x_{11} & x_{11}^2 & \log(x_{12}) & x_{13}1_{\{x_{13} \leq 7\}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n1}^2 & \log(x_{n2}) & x_{n3}1_{\{x_{n3} \leq 7\}} \end{pmatrix}_{n \times 5} * \beta + \varepsilon$$

lautet in S: `Y ~ x1 + x1^2 + log(x2) + I(x3 * (x3 <= 7)) .`

Ein Modell, in dem eine einzelne Covariable x_1 in orthogonalpolynomialer Form bis zum Grad k auftritt, à la

$$Y = \begin{pmatrix} 1 & p_1(x_{11}) & \dots & p_k(x_{11}) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p_1(x_{n1}) & \dots & p_k(x_{n1}) \end{pmatrix}_{n \times (k+1)} * \beta + \varepsilon,$$

wobei p_1, \dots, p_k Orthonormalpolynome mit $\text{Grad}(p_j) = j$ für $j = 1, \dots, k$ sind, schreibt sich in S als `Y ~ poly(x1, degree= k) .`

Wir konzentrieren uns nun auf die Funktion `poly()`: Es sei \mathbf{x} ein `numeric`-Vektor der Länge n und $\mathbf{k} \in \mathbb{N}$. In der (von uns ausschließlich verwendeten) Form `poly(x, degree= k)` wird zu dem Vektor $\mathbf{x} \equiv x = (x_1, \dots, x_n)'$ eine Basis von **Orthonormalpolynomen** p_1, \dots, p_k bis zum maximalen Grad k erzeugt, so dass $\text{Grad}(p_j) = j$. Die Orthogonalität besteht hierbei bezüglich der Stützstellen x_1, \dots, x_n , was bedeutet, dass für alle $1 \leq s, t \leq k$ gilt:

$$\begin{aligned} (p_s(x))' p_t(x) &\equiv (p_s(x_1), \dots, p_s(x_n)) \begin{pmatrix} p_t(x_1) \\ \vdots \\ p_t(x_n) \end{pmatrix} \\ &= \begin{cases} 0 & , s \neq t \\ \|p_s\|^2 := \sum_{i=1}^n p_s(x_i)^2 = 1, & s = t \end{cases} \end{aligned}$$

Damit eine solche Basis existiert, muss $k \leq n$ sein.

Diese k Polynome werden von `poly(x, degree= k)` erzeugt und ein jedes an den Stellen x_1, \dots, x_n ausgewertet. Die Resultate werden spaltenweise zu einer $(n \times k)$ -Matrix

$$P_{(k)}(x) := \begin{pmatrix} p_1(x_1) & p_2(x_1) & \dots & p_k(x_1) \\ p_1(x_2) & p_2(x_2) & \dots & p_k(x_2) \\ \vdots & \vdots & & \vdots \\ p_1(x_n) & p_2(x_n) & \dots & p_k(x_n) \end{pmatrix}_{n \times k} \equiv (p_1(x), \dots, p_k(x))$$

zusammengefasst. Sie erfüllt offenbar $(P_{(k)}(x))' P_{(k)}(x) = I_{k \times k}$.

Beispiele: Grafische Veranschaulichung der ersten fünf Orthonormalpolynome zu zwei verschiedenen, 20-elementigen Vektoren: Es sei $n = 20$ und

- für $x = (x_1, \dots, x_n)'$ seien die x_i äquidistant mit $x_i := i$ für $i = 1, \dots, n$.
- In $y = (y_1, \dots, y_n)'$ seien die y_i *nicht* äquidistant, sondern es gelte $y_i := 1.2^{x_i}$.

In S-PLUS generieren wir diese Vektoren und die dazugehörigen Orthonormalpolynombasen durch

```
> x <- 1:20;          y <- 1.2^x
> P5x <- poly( x, 5); P5y <- poly( y, 5)
```

In `P5x` und `P5y` befinden sich nun die respektiven (20×5) -Matrizen (die mit einigen Attributen versehen sind, auf die wir hier aber nicht eingehen). Beispiel:

```
> P5x
      1          2          3          4          5
[1,] -0.36839420  0.43019174 -0.43760939  0.40514757 -0.34694765
[2,] -0.32961586  0.29434172 -0.16122451 -0.02132356  0.20086443
[3,] -0.29083753  0.17358614  0.03838679 -0.23455912  0.32260045
....
[20,] 0.36839420  0.43019174  0.43760939  0.40514757  0.34694765
```

Dass die Spalten einer jeden der Matrizen orthogonal zueinander sind, zeigen wir hier jetzt nicht, aber eine grafische Darstellung, in der für jedes $j = 1, \dots, k$ ein Polyzug durch die Punkte $(x_i, p_j(x_i))$ geplottet wird, verdeutlicht, was hinter den obigen Zahlenkolonnen steckt. Die dazu verwendete Funktion `matplot()` plottet jede Spalte einer n -zeiligen Matrix gegen jede Spalte einer anderen, ebenfalls n -zeiligen Matrix, wobei jede Kombination einen eigenen Linientyp erhält. Jede der Matrizen kann auch ein n -elementiger Vektor sein. Zur Orientierung haben wir die x - bzw. y -Elemente als Punkte (auf der Nulllinie) eingetragen, um ihre äquidistante bzw. nicht äquidistante Lage abzubilden.

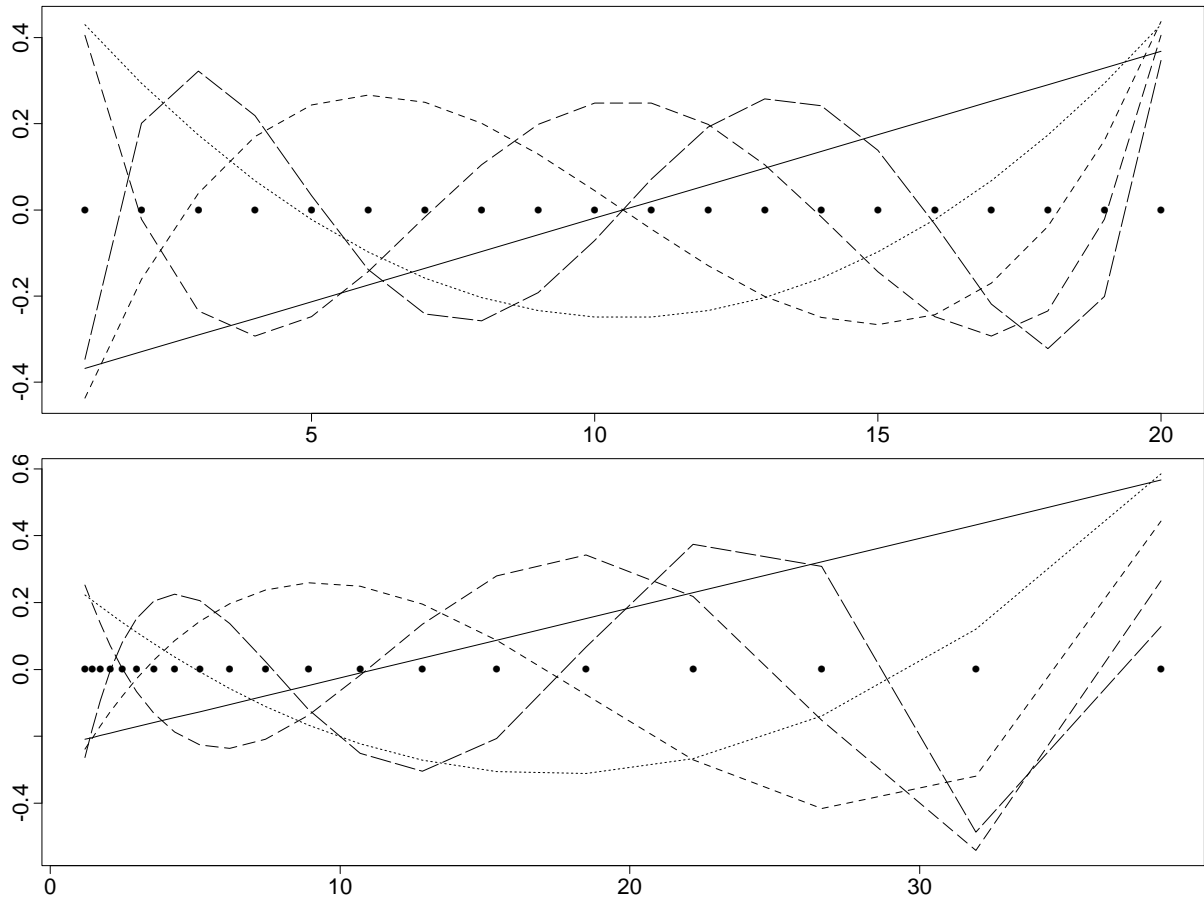
```
> matplot( x, P5x, type= "l", col= 1);  points( x, rep( 0, length( x)))
> matplot( y, P5y, type= "l", col= 1);  points( y, rep( 0, length( y)))
```

(Siehe Plots auf der nächsten Seite oben.)

Wofür sind diese Orthonormalpolynombasen gut?

Sollen in einem linearen Modell Potenzen von metrisch-stetigen, also von `numeric`-Covariablen verwendet werden, wie z. B. geschehen im Ozon-Modell auf Seite 39 mit linearen

sowie quadratischen Temperatur- und Strahlungstermen, so bekommt man es schnell mit



einer gefährlichen Eigenschaft der Monome $1, x, x^2, x^3$ etc. zu tun: Bei ihrer Verwendung werden die Spalten der Design-Matrix, da sie Potenzen voneinander sind, sehr schnell hoch korreliert und die Design-Matrix somit „fast-singulär“. Dies kann sowohl zu numerischer Instabilität des Modellfits führen als auch sehr große Varianzen und Korrelationen für die Parameterschätzer liefern.

Bei der Verwendung der orthogonalen Polynome ist beides nicht der Fall. Dies kann man ganz deutlich an der Matrix der Korrelationen der Koeffizienten (**Correlation of Coefficients**) des Fits ablesen, in der die Korrelationen der Koeffizienten der orthogonalen Modellkomponenten Null sind, wie das folgende Beispiel dokumentiert.

Anhand eines (künstlichen) **Beispiels** wird nun die Verwendung der Funktion `poly()` in einem linearen Modell erläutert. Dazu simulieren wir Realisierungen in dem Modell

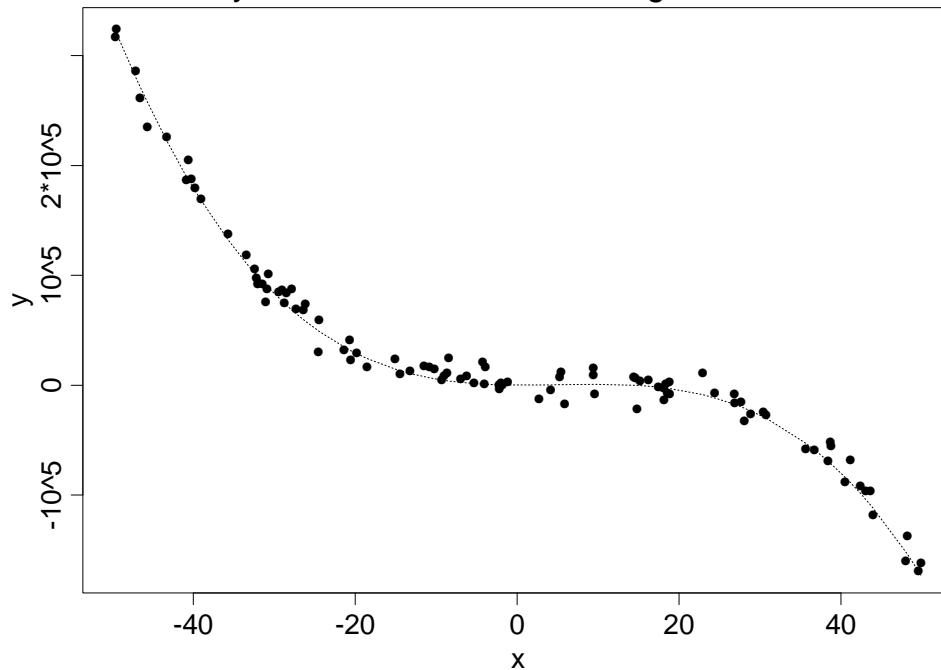
$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i = 50 - 43x_i + 31x_i^2 - 2x_i^3 + \varepsilon_i,$$

indem wir erst die x_i als Design erzeugen, dann die Werte der Regressionsfunktion $m(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$ an diesen x_i berechnen und hieraus schließlich durch zufällige, additive „Störungen“ ε_i die Response-Werte y_i generieren. In S:

```
> x <- runif( 100, -50, 50)
> m <- 50 - 43*x + 31*x^2 - 2*x^3
> y <- m + rnorm( 100, sd= 10000)
```

Das zu diesen Daten gehörende Streudiagramm der (x_i, y_i) , in dem die Regressionsfunktion $x \mapsto m(x)$ – eben ein Polynom dritten Grades – (fein) gepunktet eingezeichnet ist, findet sich auf der nächsten Seite oben.

Ein Polynom dritten Grades als Regressionsfunktion



Für den Fit eines orthogonalpolynomialen Modells dritten Grades gehen wir wie folgt vor:

```
> xy.fit <- lm( y ~ poly( x, 3));      summary( xy.fit)
Call: lm(formula = y ~ poly(x, 3))
```

Residuals:

Min	1Q	Median	3Q	Max
-23122	-5634	768.7	5641	18353

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	28764.7263	924.7141	31.1066	0.0000
poly(x, 3)1	-833114.3733	9247.1412	-90.0943	0.0000
poly(x, 3)2	212962.4433	9247.1412	23.0301	0.0000
poly(x, 3)3	-368712.0983	9247.1412	-39.8731	0.0000

Residual standard error: 9247 on 96 degrees of freedom

Multiple R-Squared: 0.9907

F-statistic: 3412 on 3 and 96 degrees of freedom, the p-value is 0

Correlation of Coefficients:

	(Intercept)	poly(x, 3)1	poly(x, 3)2
poly(x, 3)1	0		
poly(x, 3)2	0	0	
poly(x, 3)3	0	0	0

Zur Interpretation der Resultate und **Warnung**: Die in der **Coefficients**-Tabelle aufgeführten Terme `poly(x, 3)1`, `poly(x, 3)2` und `poly(x, 3)3` sind die S-Bezeichnungen für die drei im Modell befindlichen Orthonormalpolynome p_1 , p_2 und p_3 , wobei die dem Term `poly(x, 3)` angehängte Ziffer stets den Grad des jeweiligen Polynoms angibt.

Die in der Spalte **Value** stehenden Koeffizientenwerte für die Polynom-Terme sind natürlich **nicht** die Schätzwerte für die Koeffizienten β_1 , β_2 und β_3 unseres Ausgangs-

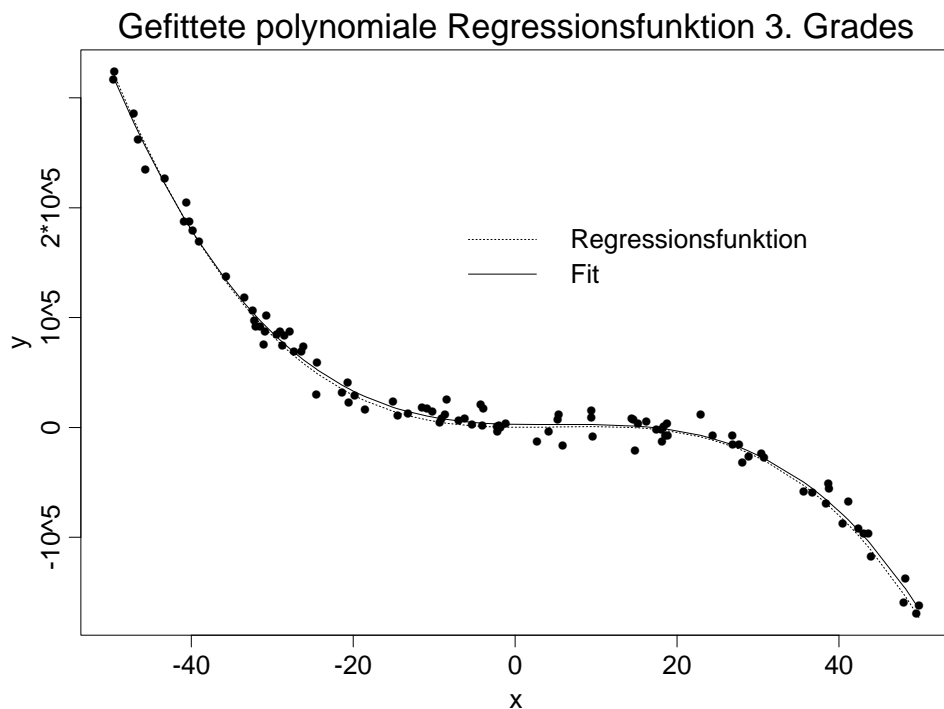
dells! Es handelt sich dabei vielmehr um die Koeffizienten der drei, passend zu den Daten generierten Orthonormalpolynome p_1 , p_2 und p_3 .

Des Weiteren ist an der Tabelle **Correlation of Coefficients** deutlich zu erkennen, dass die Koeffizienten dieser orthogonalen Komponenten unkorreliert sind.

Ist man an den Schätzwerten für β_1 , β_2 und β_3 interessiert, so müssen die erhaltenen Koeffizientenwerte also noch in eben jene umgeformt werden. Dies leistet die Funktion `poly.transform()`. Ihr Aufruf mit der verwendeten `poly()`-Funktion als erstes Argument und den Koeffizienten des Fits liefert das Gewünschte:

```
> poly.transform( poly( x, 3), coef( xy.fit))
      x^0      x^1      x^2      x^3
2721.924 -138.6709 30.68751 -1.912555
```

Wie wir sehen, liegen die Koeffizientenschätzwerte teilweise recht gut, teilweise scheinen sie allerdings weit von ihren „Zielwerten“ entfernt zu sein. Aber ein Plot der gefitteten Regressionsfunktion zeigt, dass der Fit sehr gut ist. (Die Terme niedrigeren Grades sind gegenüber denen höheren Grades schnell vernachlässigbar.)



1.10 Faktorvariablen und Interaktionsterme im linearen Regressionsmodell

Zur Erinnerung: Variablen vom Typ `factor` bzw. `ordered factor` dienen in S-PLUS der Repräsentierung von Daten des nominalen bzw. ordinalen Skalenniveaus. Wir sprechen zusammenfassend kurz von Faktoren. Sie können in Regressionsmodellen als Covariablen zur Gruppierung der Beobachtungen verwendet werden, um gewissermaßen „gruppenspezifische Submodelle“ zu fiten.

Anhand des eingebauten Datensatzes `fuel.frame` sollen zunächst die Besonderheiten bei der Verwendung von **ungeordneten** Faktoren, also nominal skalierten Covariablen in Regressionsfunktionen erläutert werden. Hier ein Ausschnitt aus `fuel.frame` und einige weitere Informationen:

```
> fuel.frame
```

		Weight	Disp.	Mileage	Fuel	Type
Eagle Summit	4	2560	97	33	3.030303	Small
Ford Escort	4	2345	114	33	3.030303	Small
...
Nissan Axxess	4	3185	146	20	5.000000	Van
Nissan Van	4	3690	146	19	5.263158	Van

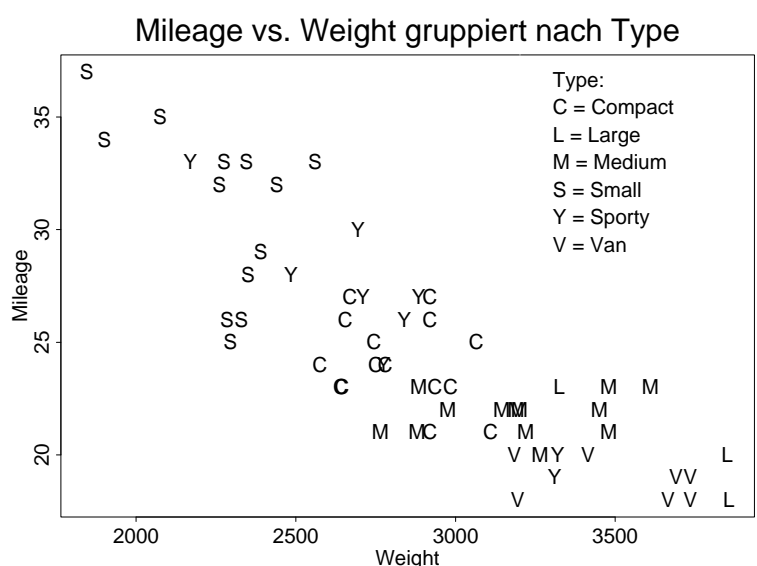
```
> sapply( fuel.frame, data.class)
```

```
Weight Disp. Mileage Fuel Type
"numeric" "numeric" "numeric" "numeric" "factor"
```

```
> table( fuel.frame$Type)
```

```
Compact Large Medium Small Sporty Van
15 3 13 13 9 7
```

Im vorliegenden Beispieldatensatz soll eine Regression der Mileage an Weight (metrisch) und Type (ungeordneter Faktor mit sechs Levels, also nominal mit sechs möglichen Ausprägungen) durchgeführt werden. In nebenstehendem Streudiagramm mit Type-spezifischen Symbolen wird deutlich, dass Type einen Einfluss auf die Beziehung zwischen Mileage und Weight haben könnte, also eine Interaktion zwischen Type und Weight vorläge.



Der Einbau von Faktoren in die Designformel eines Regressionsmodells geschieht genau wie bei metrischen (also `numeric`) Variablen. Allerdings ist zu beachten, dass die Levels eines Faktors *keine metrische* Bedeutung haben! D. h., selbst bei einer Codierung der Levels durch Zahlenwerte darf ein Faktor nicht wie eine metrisch skalierte Variable

behandelt und einfach mit einem (multiplikativen) Koeffizienten versehen werden. Stattdessen wird für einen Faktor ein ganzer Satz von Koeffizienten gefittet, und zwar *für jedes seiner Levels ein eigener Koeffizient*.

Die folgenden Abschnitte 1.10.1 und 1.10.2 geben einen Überblick über drei Möglichkeiten, einen ungeordneten Faktor und eine metrische Variable in einem Regressionsmodell zu kombinieren: Ohne Interaktion (mit rein additivem Effekt) und durch zwei mögliche Formen der Interaktion (faktoriell bzw. “crossed” oder hierarchisch bzw. “nested”).

Die erste Methode (ohne Interaktion) liefert parallele Regressiongeraden mit unterschiedlichen konstanten Termen. Die anderen beiden Methoden liefern für jedes Faktorlevel eine Regressionsgerade mit eigener Steigung und eigenem konstanten Term, allerdings in verschiedenen Parametrisierungen desselben Modells, die damit unterschiedliche Interpretationen der Beziehung zwischen den Designvariablen und der Response erlauben. Außerdem lassen sich mit diesen beiden Methoden auch Regressiongeraden mit gleichem konstantem Term und unterschiedlichen Steigungen fitten.

Bemerkung: In vielen der folgenden Parametrisierungen sind die auftretenden Koeffizienten *nicht* eindeutig bestimmt (man sagt auch „nicht identifizierbar“), weswegen S-PLUS besondere Vorkehrungen treffen muss, worauf wir aber erst in Abschnitt 1.10.4 eingehen.

1.10.1 Ein ungeordneter Faktor und eine metrische Variable ohne Interaktion: „Parallele“ Regression

Soll in unserem Beispiel der Faktor `Type` ohne Interaktion mit der metrischen Variable `Weight` in das Modell eingehen, also durch seine Levels lediglich als ein additiver Effekt auf die `Mileage` wirken, so handelt es sich um das Modell

$$\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_1 \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I.$$

Hier werden für jede Variable ihre „Haupteffekte“ gefittet: Für jedes Faktor-Level i von `Type` der Effekt α_i als Abweichung vom konstanten Term β_0 (hier: $i = 1, \dots, I = 6$) und für `Weight` die „gemeinsame“ Steigung β_1 . (Insbesondere wird also *keine* Interaktion zwischen `Type` und `Weight` modelliert, da das Level von `Type` keinen Einfluss auf den Effekt, d. h. den Koeffizienten von `Weight` hat.)

In der S-PLUS-Formelsyntax wird dies durch

$$\text{Mileage} \sim \text{Type} + \text{Weight}$$

erreicht, wobei diese Formel dasselbe wie `Mileage ~ 1 + Type + Weight` bedeutet, da der hier durch `1` explizit modellierte konstante Term gemäß Voreinstellung auch in der Formel ohne `1` in das Modell eingebaut wird.

1.10.2 Ein ungeordneter Faktor und eine metrische Variable mit Interaktionen

In Abschnitt 1.3 über die allgemeine Formelsyntax und anhand des Beispiels in Abschnitt 1.4 hatten wir das Konzept der Interaktion stetiger Designvariablen schon kennen gelernt. Auch für die Kombination eines Faktors mit einer stetigen Variablen gibt es Interaktionen. Dabei wird für *jedes* Level des Faktors ein eigener Koeffizient für die stetige (!) Variable

ermittelt. Die Parametrisierung dieses Modells kann auf zwei verschiedene Arten erfolgen, aber *hinsichtlich der Effekte der beteiligten Variablen auf die Response sind sie äquivalent*, wenngleich sich auch die Notationen in der S-PLUS-Formelsyntax unterscheiden.

1.10.2.1 Das faktorielle (= “crossed”) Modell

Zusätzlich zu den *Haupteffekten* für **Type** ($\alpha_1, \dots, \alpha_I$) und **Weight** (β_1) werden *Interaktionseffekte* zwischen diesen beiden bestimmt: γ_i als Abweichung der **Weight**-Steigung von der gemeinsamen Steigung β_1 im Faktorlevel i von **Type**. Die mathematische Formulierung lautet

$$\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_1 \text{Weight}_{ij} + \gamma_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I$$

und die S-PLUS-Formelsyntax:

$$\text{Mileage} \sim 1 + \text{Type} + \text{Weight} + \text{Type:Weight}$$

Auch hier kann die 1 weggelassen und der ganze Ausdruck sogar noch weiter durch $\text{Mileage} \sim \text{Type} * \text{Weight}$ abgekürzt werden.

Die in diesem Modell nahe liegende Zusammenfassung des konstanten Terms β_0 und der Faktor-Haupteffekte α_i zu der Parametrisierung

$$\text{Mileage}_{ij} = \beta_i + \beta_1 \text{Weight}_{ij} + \gamma_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I$$

wird in S-PLUS-Formelsyntax durch

$$\text{Mileage} \sim -1 + \text{Type} + \text{Weight} + \text{Type:Weight}$$

oder kürzer durch $\text{Mileage} \sim \text{Type} * \text{Weight} - 1$ erreicht. Dies ändert nicht die Effekte der beteiligten Variablen auf die Response, sondern nur die Parametrisierung des Modells.

1.10.2.2 Das hierarchische (= “nested”) Modell

In diesem Modell werden die Haupteffekte α_i für den Faktor **Type** gefittet und „innerhalb“ eines jeden Faktor-Levels i eine spezifische **Weight**-Steigung. Mathematisch:

$$\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I.$$

In der S-PLUS-Formelsyntax wird dies durch

$$\text{Mileage} \sim 1 + \text{Type} + \text{Weight \%in\% Type}$$

beschrieben, wobei auch wieder die 1 weggelassen werden kann und die Kurzform des Ausdrucks $\text{Mileage} \sim \text{Type} / \text{Weight}$ lautet. Beachte dabei, dass der Formeloperator */* nicht kommutativ ist (im Gegensatz zu ***)!

Auch hier sind konstanter Term β_0 und Faktor-Haupteffekte α_i zusammenfassbar zu

$$\text{Mileage}_{ij} = \alpha_i + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I,$$

was in S-PLUS-Formelsyntax mit

$$\text{Mileage} \sim -1 + \text{Type} + \text{Weight \%in\% Type}$$

oder kürzer mit $\text{Mileage} \sim \text{Type} / \text{Weight} - 1$ geschieht. Und wieder ändern sich nicht die Effekte der beteiligten Variablen auf die Response, sondern lediglich die Modellparametrisierung.

1.10.2.3 Modifikationen der beiden Interaktionsmodelle

Ein Modell aus Regressiongeraden mit gleichem konstantem Term und unterschiedlichen Steigungen ist sowohl im hierarchischen als auch im faktoriellen Modell erreichbar:

- Im faktoriellen Modell

$$\text{Mileage}_{ij} = \beta_0 + \beta_1 \text{Weight}_{ij} + \gamma_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I,$$

durch `Mileage ~ 1 + Weight + Type:Weight`

und kürzer durch `Mileage ~ Type * Weight - Type`

- Im hierarchischen Modell

$$\text{Mileage}_{ij} = \beta_0 + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I,$$

durch `Mileage ~ 1 + Weight %in% Type`

und abgekürzt durch `Mileage ~ Type / Weight - Type`

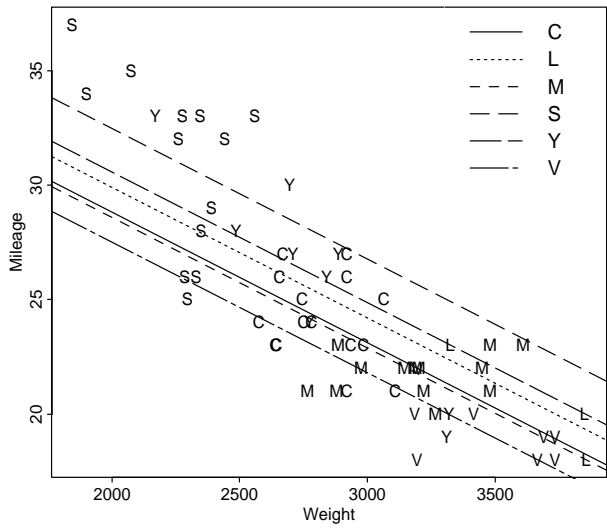
1.10.3 Die Modelle im Überblick

Es folgt eine tabellarische Zusammenfassung der mathematischen Parametrisierungen und der abgekürzten sowie „expandierten“ S-PLUS-Formeln obiger Modelle. Auf der nächsten Seite befindet sich eine (entsprechend der Tabelle angeordnete) grafische Übersicht der gefitteten Regressionsgeraden.

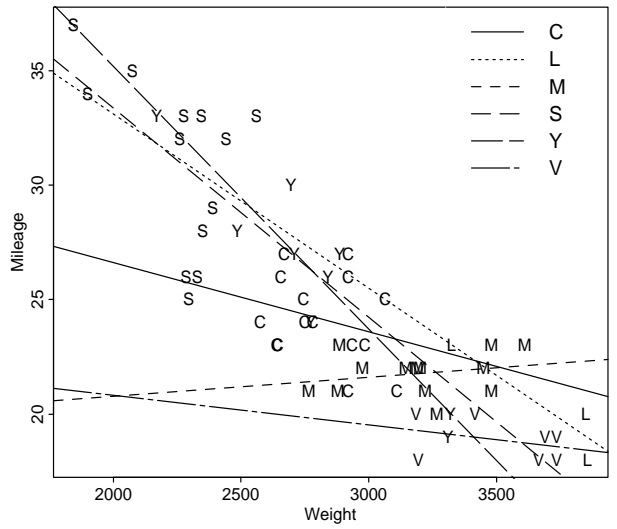
<p>Ohne Interaktion (parallele Geraden):</p> $\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_1 \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type + Weight</code> <code>Mileage ~ 1 + Type + Weight</code></p>	<p>Hierarchisch (levelspezifische Geraden):</p> $\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type / Weight</code> <code>Mileage ~ 1 + Type + Weight %in% Type</code></p>
<p>Faktoriell (levelspezifische Geraden):</p> $\text{Mileage}_{ij} = \beta_0 + \alpha_i + \beta_1 \text{Weight}_{ij} + \gamma_i \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type * Weight</code> <code>Mileage ~ 1 + Type + Weight + Type:Weight</code></p>	<p>Hierarchisch (levelspezifische Geraden):</p> $\text{Mileage}_{ij} = \alpha_i + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type / Weight - 1</code> <code>Mileage ~ -1 + Type + Weight %in% Type</code></p>
<p>Faktoriell (levelspezifische Steigungen):</p> $\text{Mileage}_{ij} = \beta_0 + \beta_1 \text{Weight}_{ij} + \gamma_i \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type * Weight - Type</code> <code>Mileage ~ 1 + Weight + Type:Weight</code></p>	<p>Hierarchisch (levelspezifische Steigungen):</p> $\text{Mileage}_{ij} = \beta_0 + \beta_i \text{Weight}_{ij} + \varepsilon_{ij}$ <p><code>Mileage ~ Type / Weight - Type</code> <code>Mileage ~ 1 + Weight %in% Type</code></p>

Beachte: Die von S-PLUS zunächst zurückgelieferten Koeffizienten sind *nicht* diejenigen aus der mathematischen Parametrisierung. Näheres dazu in Abschnitt 1.10.4.

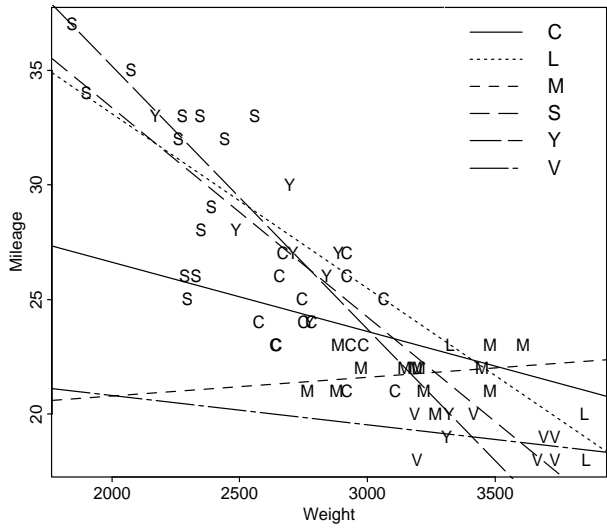
Mileage ~ Type + Weight



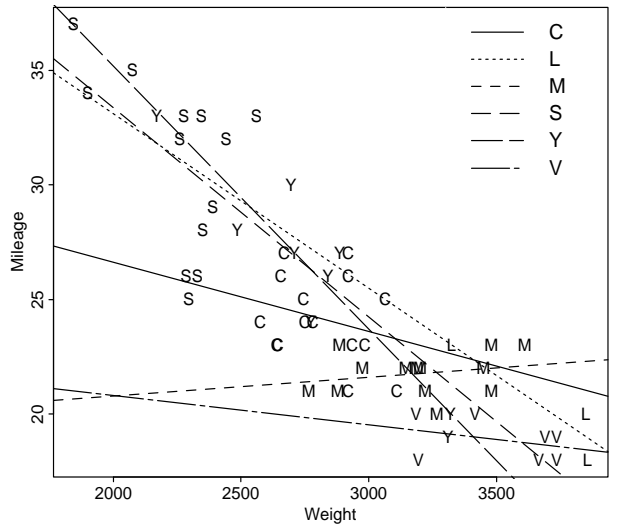
Mileage ~ Type/Weight



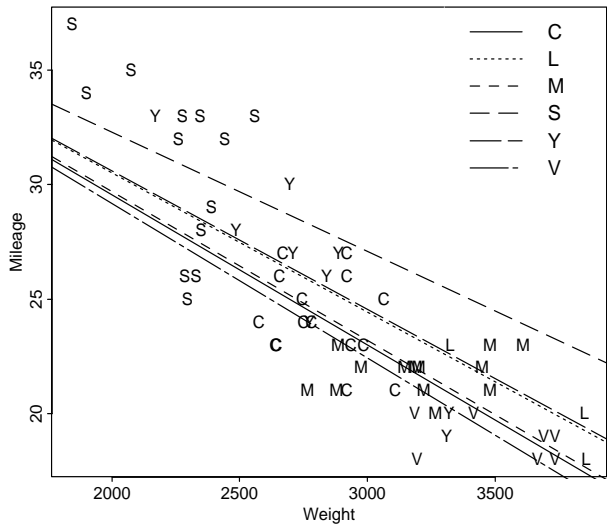
Mileage ~ Type * Weight



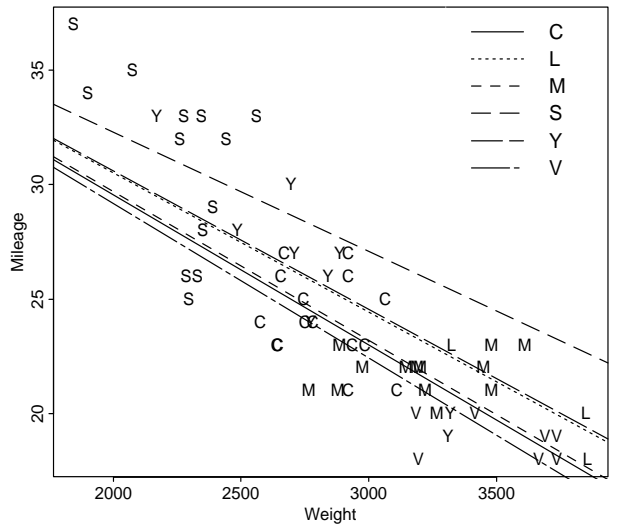
Mileage ~ Type/Weight - 1



Mileage ~ Type * Weight - Type



Mileage ~ Type/Weight - Type



1.10.4 Modell-Parametrisierung ungeordneter Faktoren durch Kontraste

Ein generelles Problem bei der Integration von Faktoren in ein Modell ist, dass sie normalerweise zu mehr Koeffizienten führen als in diesem Modell geschätzt werden können. Mit anderen Worten: Sie sind nicht „identifizierbar“, was auch *funktionale* Überparametrisierung genannt wird. Das folgende, einfache Beispiel eines linearen Modells, in dem als einzige Covariable ein Faktor auftritt, soll das Problem klar machen. Wir betrachten

$$Y_{ij} = \beta_0 + \alpha_i + \varepsilon_{ij}, \quad j = 1, \dots, n_i; \quad i = 1, \dots, I \quad \text{mit } n = \sum_{i=1}^I n_i, \quad (5)$$

wie z. B. `Mileage ~ Type` in S-PLUS-Formelnotation. In Matrix-Formulierung lautet das:

$$Y = [\mathbf{1}_n | X_a] \begin{pmatrix} \beta_0 \\ \alpha \end{pmatrix} + \varepsilon,$$

wobei $\mathbf{1}_n$ der n -dimensionale Einsen-Vektor ist,

$$Y = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{1n_1} \\ Y_{21} \\ \vdots \\ Y_{2n_2} \\ \vdots \\ \vdots \\ Y_{I1} \\ \vdots \\ Y_{In_I} \end{pmatrix}_{n \times 1}, \quad X_a = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ 1 & 0 & & & \\ 0 & 1 & & & \\ \vdots & \vdots & & & \\ 1 & & & & \\ 0 & \dots & & & \vdots \\ \vdots & & \dots & & 0 \\ & & & & 1 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}_{n \times I}, \quad \alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_I \end{pmatrix}_{I \times 1}$$

und ε analog zu Y .

Die $n \times I$ -**Inzidenzmatrix** X_a hat offensichtlich den Rang I , aber die 1-Spalte zur Codierung des konstanten Terms β_0 in der Designmatrix $X := [\mathbf{1}_n | X_a]$ ist von den Spalten von X_a linear abhängig. In Konsequenz hat die $n \times (I+1)$ -Matrix X den Rang I , sodass die $(I+1) \times (I+1)$ -Matrix $X'X$ singular ist! D. h., das zu lösende Kleinste-Quadrate-Problem führt auf ein überbestimmtes Gleichungssystem hinaus, weil das Modell überparametrisiert ist bzw. die Parameter $\beta_0, \alpha_1, \dots, \alpha_I$ nicht identifizierbar sind. (Zum Beispiel kann β_0 durch $\beta_0 + \delta$ für ein beliebiges konstantes δ ersetzt und dies durch $\alpha_i - \delta$ (für jedes $i = 1, \dots, I$) kompensiert werden.)

Die Lösung dieses Problems ist eine geeignete Reparametrisierung des Modells: Ersetze α durch $C_a \alpha^*$ mit einem Parametervektor $\alpha^* = (\alpha_1^*, \dots, \alpha_{I-1}^*)'$ und einer geeigneten $I \times (I-1)$ -**Kontrastmatrix** C_a , sodass die $n \times I$ -Matrix $X^* := [\mathbf{1}_n | X_a C_a]$ den Rang I hat. Dies ist, wie man zeigen kann, z. B. erfüllt, falls $\text{Rang}([\mathbf{1}_I | C_a]) = I$.

Man erhält dadurch ein „neues“ Modell

$$Y = X^* \begin{pmatrix} \beta_0 \\ \alpha^* \end{pmatrix} + \varepsilon \quad \text{mit invertierbarem } (X^*)'X^*.$$

Bemerkung: Wegen $\text{Rang}(C_a) = I-1$ existiert zu C_a die eindeutig bestimmte Links-Inverse $C_a^+ := (C_a' C_a)^{-1} C_a'$, sodass wir aus $\alpha = C_a \alpha^*$ gemäß $\alpha^* = C_a^+ \alpha$ die „ursprüngliche“ Parametrisierung zurückerhalten.

1.10.4.1 „Treatment“-Kontraste: Definition und Eigenschaften

Natürlich stellt sich die Frage nach der Wahl von C_a . Es gibt viele Möglichkeiten, eine Reparametrisierung vorzunehmen. S-PLUS wählt bei einem ungeordneten Faktor für C_a gemäß Voreinstellung die so genannten Helmert-Kontraste, auf die wir aber erst in Abschnitt 1.10.4.5 eingehen werden, da sie recht kompliziert sind und sich zur „Eingewöhnung“ nicht besonders eignen. Viel übersichtlicher und suggestiver sind dagegen die so genannten „**treatment**“-Kontraste, durch die der Parametervektor(-anteil) $\alpha = (\alpha_1, \dots, \alpha_I)'$ folgendermaßen als lineare Transformation eines Vektors $\alpha^* = (\alpha_1^*, \dots, \alpha_{I-1}^*)'$ ausgedrückt wird:

$$\alpha = C_a \alpha^* \quad \text{mit} \quad C_a = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}_{I \times (I-1)}. \quad (6)$$

Einige **Eigenschaften der treatment-Kontraste**:

- Die Spalten von C_a sind orthogonal zueinander und $\text{Rang}([\mathbf{1}_I | C_a]) = I$, sodass folgt: $\text{Rang}([\mathbf{1}_n | X_a C_a]) = I$.
- Die Links-Inverse C_a^+ ergibt sich wegen $C_a' C_a = \text{diag}(\mathbf{1}_{I-1})$ sofort zu

$$C_a^+ = (C_a' C_a)^{-1} C_a' = C_a' = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}_{(I-1) \times I}.$$

- Damit lautet die Beziehung zwischen den Kontrasten α^* und den Effekten α :

$$\alpha^* = C_a^+ \alpha = \begin{pmatrix} \alpha_2 \\ \vdots \\ \alpha_I \end{pmatrix}.$$

Das bedeutet faktisch, dass der Effekt α_1 aus dem Modell eliminiert (bzw. gleich Null gesetzt) wird und β_0 zur erwarteten Response des Levels 1 wird. Die Level-1-Beobachtungen werden so zu einer Art „Bezugsgruppe“ (oder „Kontrollgruppe“) interpretierbar. Der Kontrast α_i^* repräsentiert damit einen Vergleich zwischen dem Faktorlevel $i+1$ und dem Level 1; er quantifiziert demnach, wie sich die Behandlung (Engl.: „treatment“) im Level $i+1$ relativ zur Kontrollgruppe auf die erwartete Response auswirkt.

- Die Rücktransformation von α^* zu α lautet:

$$\alpha = C_a \alpha^* = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}_{I \times (I-1)} \begin{pmatrix} \alpha_1^* \\ \vdots \\ \alpha_{I-1}^* \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha_2 \\ \vdots \\ \alpha_I \end{pmatrix}.$$

1.10.4.2 Treatment-Kontraste im Beispiel der parallelen Regression

Für das Modell $Y_{ij} = \beta_0 + \alpha_i + \beta_1 x_{ij} + \varepsilon_{ij}$, $j = 1, \dots, n_i$; $i = 1, \dots, I$ aus Abschnitt 1.10.1 lautet die Modellformel in Matrix-Notation

$$Y = X\beta + \varepsilon, \quad (7)$$

wobei (mit $n = \sum_{i=1}^I n_i$)

$$Y = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{1n_1} \\ Y_{21} \\ \vdots \\ Y_{2n_2} \\ \vdots \\ \vdots \\ Y_{I1} \\ \vdots \\ Y_{In_I} \end{pmatrix}_{n \times 1}, \quad X = \begin{pmatrix} 1 & 1 & 0 & \dots & \dots & 0 & x_{11} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ & & 1 & 0 & & & x_{1n_1} \\ & & 0 & 1 & & & x_{21} \\ & & \vdots & \vdots & & & \vdots \\ & & & 1 & & & x_{2n_2} \\ & & 0 & \ddots & & \vdots & x_{31} \\ & & \vdots & & \ddots & 0 & \vdots \\ & & & & & 1 & x_{I1} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 1 & x_{In_I} \end{pmatrix}_{n \times (I+2)}, \quad \beta = \begin{pmatrix} \beta_0 \\ \alpha_1 \\ \vdots \\ \alpha_I \\ \beta_1 \end{pmatrix}_{(I+2) \times 1}$$

und ε analog zu Y .

Wie auch schon im einführenden Beispiel (5), sind in dieser Codierung die zu $\beta_0, \alpha_1, \dots, \alpha_I$ gehörenden Spalten der Designmatrix X linear abhängig, was die Matrix $X'X$ singular werden lässt. Daher kann das obige Modell so nicht gefittet werden, sondern muss zunächst mit Hilfe einer Kontrastmatrix C_a reparametrisiert werden, um die Parameterzahl zu reduzieren. Dies läuft auf eine Transformation desjenigen Teils der Designmatrix hinaus, der die Codierung der Faktorlevel-Koeffizienten enthält, also hier der Spalten zwei bis $I+1$.

Wird die treatment-Kontrastmatrix C_a aus (6) erweitert zu

$$A := \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & & & & \vdots \\ \vdots & C_a & & & \vdots \\ \vdots & & & & 0 \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & \vdots \\ \vdots & 1 & 0 & \dots & 0 & \\ & 0 & 1 & \ddots & \vdots & \\ & \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & & \dots & 0 & 1 \end{pmatrix}_{(I+2) \times (I+1)},$$

so ist $\beta = (\beta_0, \alpha_1, \dots, \alpha_{I-1}, \alpha_I, \beta_1)' = A\gamma$ mit $\gamma = (\beta_0, \alpha_1^*, \dots, \alpha_{I-1}^*, \beta_1)' = (\beta_0, \alpha_2, \dots, \alpha_I, \beta_1)'$ und β_0 sowie β_1 werden durch diese Reparametrisierung nicht beeinflusst. (Beachte, dass in Konsequenz $\alpha_1 = 0$ ist.)

Statt für Modell (7) wird nun für das reparametrisierte Modell

$$Y = XA\gamma + \varepsilon =: \tilde{X}\gamma + \varepsilon \quad (8)$$

der KQS $\hat{\gamma}$ für γ ermittelt, wobei

$$\tilde{X} \equiv XA = \begin{pmatrix} 1 & 0 & 0 & \dots & \dots & 0 & x_{11} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ & 0 & & & & & x_{1n_1} \\ & 1 & & & & & x_{21} \\ & \vdots & \vdots & & & & \vdots \\ & 1 & 0 & & & & x_{2n_2} \\ & 0 & 1 & & & & x_{31} \\ & \vdots & \vdots & & & & \vdots \\ & & 1 & & & & x_{3n_3} \\ & & 0 & \ddots & & \vdots & x_{41} \\ & & \vdots & & \ddots & 0 & \vdots \\ & & & & & 1 & x_{I1} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 1 & x_{In_I} \end{pmatrix}_{n \times (I+1)}$$

Soll S-PLUS ein lineares Modell mit ungeordneten Faktoren unter Verwendung der treatment-Kontraste fitten, so muss zunächst die Voreinstellung geändert und die Verwendung dieser für ungeordnete Faktoren aktiviert werden; das geschieht mit dem Befehl `options(contrasts= c("contr.treatment", "contr.poly"))`. Danach wird wie üblich das Regressionsmodell gefittet; hier ist das Vorgehen also wie folgt:

```
> options( contrasts= c( "contr.treatment", "contr.poly" ))
> fit1 <- lm( Mileage ~ Type + Weight, fuel.frame);      summary( fit1)
```

....

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	40.2053	3.7390	10.7530	0.0000
TypeLarge	1.0750	1.8809	0.5715	0.5701
TypeMedium	-0.2289	1.0300	-0.2223	0.8249
TypeSmall	3.6574	1.1675	3.1326	0.0028
TypeSporty	1.7407	1.0085	1.7260	0.0902
TypeVan	-1.3101	1.4233	-0.9204	0.3615
Weight	-0.0057	0.0013	-4.3583	0.0001

....

Wir stellen fest: Die fünf (!) Zeilen für die Koeffizienten des *sechs* Levels besitzenden Faktors `Type` sind mit `TypeLarge` bis `TypeVan` bezeichnet. Insbesondere existiert kein `TypeCompact`-Koeffizient. Dies ist natürlich korrekt, da ja das reparametrisierte Modell $Y = \tilde{X}\gamma + \varepsilon$ verwendet wurde, in dem nicht die Schätzwerte $\hat{\alpha}_1, \dots, \hat{\alpha}_6$ für die Faktor-Level-Koeffizienten $\alpha_1, \dots, \alpha_6$ bestimmt wurden, sondern die Schätzwerte $\hat{\alpha}_1^*, \dots, \hat{\alpha}_5^*$ für die treatment-Kontraste $\alpha_1^*, \dots, \alpha_5^*$, welche für die Levels `Large` bis `Van` die konstante Abweichung vom Level `Compact` darstellen. Die ausgegebenen Werte sind daher auch *nicht* die Schätzwerte $\hat{\alpha}_1, \dots, \hat{\alpha}_6$, sondern die Schätzwerte $\hat{\alpha}_1^*, \dots, \hat{\alpha}_5^*$. (Diese stehen dann via $\hat{\alpha} = C_a \hat{\alpha}^*$ miteinander in Beziehung.)

Konkret beschreibt hier der konstante Term $\hat{\beta}_0 = 40.2053$ zusammen mit der Steigung $\hat{\beta}_1 = -0.0057$ die Regressionsgerade zum Level `Compact` und für die anderen fünf Levels `Large` bis `Van` sind die Koeffizienten `TypeLarge` bis `TypeVan` die konstanten *Abweichungen* der `Mileage` von dieser Regressionsgeraden.

Um an den Schätzwert für β im ursprünglichen Modell (7) zu kommen, muss die Reparametrisierung aus Modell (8) rückgängig gemacht werden, was gemäß

$$\tilde{X}\gamma \equiv XA\gamma = X\beta$$

durch $\hat{\beta} = A\hat{\gamma}$ erreicht wird. Diese Re-Reparametrisierung ist implementiert durch die Funktion `dummy.coef()`:

```
> dummy.coef( fit1)
$(Intercept)":
(Intercept)
  40.2053

$Type:
Compact      Large      Medium      Small      Sporty      Van
  0  1.074954 -0.2289456  3.657357  1.740694 -1.310085

$Weight:
Weight
-0.005697258
```

Das Resultat von `dummy.coef()` bestätigt, dass $\alpha_1 = 0$ gesetzt wurde und die Level-1-Beobachtungen als eine Art Kontroll-/Bezugs-/Referenzgruppe für die übrigen Levels betrachtet werden können

Bemerkung: Die Faktorkoeffizienten $\alpha_1, \dots, \alpha_I$ werden von S-PLUS automatisch (!) in dieser Reihenfolge den Faktorlevels zugeordnet, d. h., Koeffizient α_i korrespondiert zu Level i . Wichtig zu wissen ist dabei, dass die Levels eines ungeordneten Faktors, wenn vom Benutzer nicht anders arrangiert, per Voreinstellung *alphabetisch aufsteigend sortiert* sind.

1.10.4.3 Treatment-Kontraste im faktoriellen Modell

In Modellen mit Interaktionen tritt das Identifizierbarkeitsproblem sogar noch stärker zu Tage: In der Parametrisierung des faktoriellen Modells (siehe Seite 58) sind sowohl $\beta_0, \alpha_1, \dots, \alpha_I$ linear abhängig als auch $\beta_1, \gamma_1, \dots, \gamma_I$. Auch hier wird intern eine (umfangreiche) Reparametrisierung vorgenommen. Die Umsetzung in S-PLUS ist die folgende (wobei die erneute Verwendung von `options()` unnötig ist, wenn S-PLUS seit ihrem letzten Aufruf noch nicht wieder neu gestartet wurde):

```
> options( contrasts= c( "contr.treatment", "contr.poly"))
> fit2 <- lm( Mileage ~ Type * Weight, fuel.frame);      summary( fit2)
....
Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)  32.6572    9.4757    3.4464  0.0012
  TypeLarge   15.6533   20.4596    0.7651  0.4480
  TypeMedium -13.5141   12.0409   -1.1223  0.2673
  TypeSmall   18.9650   11.6683    1.6253  0.1106
  TypeSporty  25.4513   11.1192    2.2890  0.0265
  TypeVan     -9.2723   15.4961   -0.5984  0.5524
```

```

      Weight -0.0030  0.0034  -0.9011  0.3721
TypeLargeWeight -0.0046  0.0060  -0.7705  0.4448
TypeMediumWeight  0.0038  0.0041   0.9429  0.3505
TypeSmallWeight -0.0061  0.0045  -1.3576  0.1809
TypeSportyWeight -0.0085  0.0039  -2.1462  0.0369
TypeVanWeight  0.0017  0.0048   0.3589  0.7212
....

```

Auch hier sind die präsentierten Koeffizientenschätzwerte *nicht* die, die wir in der Parametrisierung des faktoriellen Modells auf Seite 58 haben, sondern diejenigen, die in dem mittels der treatment-Kontraste reparametrisierten Modell auftreten. Also ist wieder eine Re-Reparametrisierung und damit ein Einsatz von `dummy.coef()` nötig, um die Parameter des faktoriellen Modells angeben zu können:

```

> dummy.coef( fit2)
$(Intercept)":
(Intercept)
  32.65721

$Type:
Compact   Large   Medium   Small   Sporty   Van
  0 15.65328 -13.51411 18.96499 25.45131 -9.272252

$Weight:
      Weight
-0.003021579

$"Type:Weight":
CompactWeight LargeWeight MediumWeight SmallWeight SportyWeight
  0 -0.0045878  0.003843332 -0.006112616 -0.008450301

      VanWeight
  0.001734224

```

Beachte, wie $32.657 - 0.00302 * \text{Weight}$ im Level `Compact` die „Bezugsgerade“ darstellt, von der die Geraden der anderen Levels in konstantem Term und Steigung abweichen.

1.10.4.4 Treatment-Kontraste im hierarchischen Modell

In der Parametrisierung des hierarchischen Modells (siehe Seite 58) sind $\beta_0, \alpha_1, \dots, \alpha_I$ linear abhängig und wieder schafft eine interne Reparametrisierung Abhilfe. Das hierarchische Modell mit treatment-Kontrasten liefert in S-PLUS (`options()`-Aufruf unnötig, wenn seit letztem Aufruf kein Neustart von S-PLUS):

```

> options( contrasts= c( "contr.treatment", "contr.poly"))
> fit3 <- lm( Mileage ~ Type / Weight, fuel.frame);      summary( fit3)
....
Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)  32.6572   9.4757    3.4464  0.0012
TypeLarge    15.6533  20.4596    0.7651  0.4480

```

TypeMedium	-13.5141	12.0409	-1.1223	0.2673
TypeSmall	18.9650	11.6683	1.6253	0.1106
TypeSporty	25.4513	11.1192	2.2890	0.0265
TypeVan	-9.2723	15.4961	-0.5984	0.5524
TypeCompactWeight	-0.0030	0.0034	-0.9011	0.3721
TypeLargeWeight	-0.0076	0.0049	-1.5464	0.1286
TypeMediumWeight	0.0008	0.0023	0.3546	0.7244
TypeSmallWeight	-0.0091	0.0030	-3.0401	0.0038
TypeSportyWeight	-0.0115	0.0021	-5.5601	0.0000
TypeVanWeight	-0.0013	0.0035	-0.3701	0.7130

....

Beachte, dass nur $\alpha_1, \dots, \alpha_I$ reparametrisiert wurden. (β_1, \dots, β_I haben schließlich auch kein Problem dargestellt.) Die Re-Reparametrisierung versorgt uns mit den Parametern des hierarchischen Modells:

```
> dummy.coef( fit3)
$(Intercept)":
(Intercept)
  32.65721

$Type:
Compact   Large   Medium   Small   Sporty   Van
      0 15.65328 -13.51411 18.96499 25.45131 -9.272252

$"Weight %in% Type":
CompactWeight LargeWeight MediumWeight SmallWeight SportyWeight
-0.003021579 -0.007609379 0.0008217528 -0.009134195 -0.01147188

VanWeight
-0.001287355
```

Die Notation "Weight %in% Type" symbolisiert, dass die Weight-Koeffizienten jeweils nur *innerhalb* eines jeden Levels des Faktors Type gültig sind.

Beachte: Es gilt in der Tat die Äquivalenz obiger Modelle, derzufolge sich die Parameter ineinander umrechnen lassen müssen:

$$\hat{\beta}_i(\text{aus fit3}) = \hat{\beta}_1(\text{aus fit2}) + \hat{\gamma}_i(\text{aus fit2}).$$

In S-PLUS:

```
> dummy.coef( fit3)$"Weight %in% Type" -
+ (dummy.coef( fit2)$Weight + dummy.coef( fit2)$"Type:Weight")

CompactWeight LargeWeight MediumWeight SmallWeight SportyWeight
-4.163336e-17          0 -1.409463e-18 -1.734723e-17 3.469447e-18

VanWeight
-1.084202e-18
```

Offenbar sind die Differenzen $\hat{\beta}_i(\text{aus fit3}) - (\hat{\beta}_1(\text{aus fit2}) + \hat{\gamma}_i(\text{aus fit2}))$ im Rahmen der Rechengenauigkeit Null.

- Die entsprechende Rücktransformation von α^* zu α lautet: $\alpha = C_a \alpha^* =$

$$= \begin{pmatrix} -1 & -1 & -1 & \dots & -1 & -1 \\ 1 & -1 & -1 & & \vdots & \vdots \\ 0 & 2 & -1 & & & \\ \vdots & 0 & 3 & \ddots & \vdots & \\ & \vdots & 0 & \ddots & -1 & \vdots \\ \vdots & \vdots & \vdots & \ddots & I-2 & -1 \\ 0 & 0 & 0 & \dots & 0 & I-1 \end{pmatrix}_{I \times (I-1)} \begin{pmatrix} \alpha_1^* \\ \vdots \\ \alpha_{I-1}^* \end{pmatrix} = \begin{pmatrix} -\sum_{s=1}^{I-1} \alpha_s^* \\ \alpha_1^* - \sum_{s=2}^{I-1} \alpha_s^* \\ 2\alpha_2^* - \sum_{s=3}^{I-1} \alpha_s^* \\ 3\alpha_3^* - \sum_{s=4}^{I-1} \alpha_s^* \\ \vdots \\ (I-2)\alpha_{I-2}^* - \alpha_{I-1}^* \\ (I-1)\alpha_{I-1}^* \end{pmatrix}.$$

- Eine weitere Konsequenz der Verwendung von Helmert-Kontrasten ist, dass sich die ursprünglichen Parameter (Haupteffekte) α_i zu Null addieren:

$$\sum_{s=1}^I \alpha_s = \mathbf{1}'_I \alpha = \mathbf{1}'_I C_a \alpha^* = \mathbf{0}'_{I-1} \alpha^* = 0.$$

1.10.4.6 Helmert-Kontraste im Beispiel der parallelen Regression

Wird für das Modell $Y = X\beta + \varepsilon$ aus Abschnitt 1.10.1 (bzw. Modell (7) in 1.10.4.2, Seite 63) die Helmert-Kontrastmatrix C_a verwendet und diese erweitert zu

$$A := \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & & & & \vdots \\ \vdots & C_a & & & \vdots \\ \vdots & & & & 0 \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & -1 & \dots & -1 & 0 \\ 0 & 1 & -1 & \dots & -1 & 0 \\ 0 & 0 & 2 & \dots & -1 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I-1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}_{(I+2) \times (I+1)},$$

dann ist $\beta = (\beta_0, \alpha_1, \dots, \alpha_{I-1}, \alpha_I, \beta_1)' = A\gamma$ mit $\gamma = (\beta_0, \alpha_1^*, \dots, \alpha_{I-1}^*, \beta_1)'$, sodass β_0 und β_1 auch durch diese Reparametrisierung unbeeinflusst gelassen werden.

Nun wird für das Modell $Y = XA\gamma + \varepsilon =: \tilde{X}\gamma + \varepsilon$ der KQS $\hat{\gamma}$ für γ ermittelt, wobei

$$\tilde{X} \equiv XA = \begin{pmatrix} 1 & -1 & -1 & \dots & \dots & -1 & x_{11} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ & -1 & & & & & x_{1n_1} \\ & 1 & & & & & x_{21} \\ & \vdots & \vdots & & & & \vdots \\ & 1 & -1 & & & & x_{2n_2} \\ & 0 & 2 & & & & x_{31} \\ & \vdots & \vdots & & & & \vdots \\ & & 2 & & & & x_{3n_3} \\ & & & 0 & \ddots & & \vdots \\ & & & \vdots & & -1 & \vdots \\ & & & & & I-1 & x_{I1} \\ \vdots & \vdots & \vdots & & & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & I-1 & x_{In_I} \end{pmatrix}_{n \times (I+1)}$$

Das spiegelt sich auch prompt in der folgenden Ausgabe von S-PLUS wider. (Beachte, dass die durch den Befehl `options(contrasts= c("contr.helmert", "contr.poly")`) erzielte Einstellung diejenige ist, die qua Voreinstellung nach jedem Neustart von S-PLUS gilt. D. h., der Befehl wäre nur notwendig, wenn das `contrasts`-Argument von `options()` im laufenden S-PLUS-Prozess bereits explizit geändert wurde.)

```
> options( contrasts= c( "contr.helmert", "contr.poly"))
> fit4 <- lm( Mileage ~ Type + Weight, fuel.frame);      summary( fit4)
....
Coefficients:
              Value Std. Error  t value Pr(>|t|)
(Intercept)  41.0276   3.9960   10.2673  0.0000
      Type1    0.5375   0.9404    0.5715  0.5701
      Type2   -0.2555   0.3360   -0.7603  0.4505
      Type3    0.8438   0.3842    2.1963  0.0325
      Type4    0.1230   0.1891    0.6503  0.5183
      Type5   -0.4265   0.2054   -2.0767  0.0427
      Weight  -0.0057   0.0013   -4.3583  0.0001
....
```

Zunächst stellen wir fest, dass die Zeilen für die Koeffizienten des Faktors `Type` nicht mit „Compact“, „Large“, „Medium“, „Small“, „Sporty“ und „Van“ bezeichnet werden, sondern mit `Type1` usw. (Das leuchtet ein, da die Helmert-Kontraste wenig mit den ursprünglichen Levels zu tun haben.) Des Weiteren existiert kein Koeffizient für das sechste Level von `Type`, weil ja das reparametrisierte Modell $Y = \tilde{X}\gamma + \varepsilon$ verwendet wurde. Die ausgegebenen Werte in den Zeilen `Type1` bis `Type5` sind also *nicht* die Schätzwerte $\hat{\alpha}_1, \dots, \hat{\alpha}_6$ für die Faktorlevel-Koeffizienten, sondern die Schätzwerte $\hat{\alpha}_1^*, \dots, \hat{\alpha}_5^*$ für die Helmert-Kontraste (die ja bekanntlich via $\hat{\alpha} = C_a \hat{\alpha}^*$ miteinander in Beziehung stehen).

Um an den Schätzwert für β im ursprünglichen Modell (7) zu kommen, muss die Reparametrisierung rückgängig gemacht werden, was durch $\hat{\beta} = A\hat{\gamma}$ erreicht wird und in `dummy.coef()` implementiert ist:

```
> dummy.coef( fit4)
$(Intercept)":
(Intercept)
  41.02763

$Type:
  Compact      Large      Medium      Small      Sporty      Van
-0.8223291  0.2526246 -1.051275  2.835028  0.9183649 -2.132414

$Weight:
      Weight
-0.005697258
```

Nun ist also für alle sechs Levels (von `Compact` bis `Van`) des Faktors `Type` die konstante Abweichung (von -0.82 bis -2.13) der `Mileage` von der „mittleren“ Regressionsgeraden mit der Steigung $\hat{\beta}_1 = -0.00570$ und dem konstanten Term $\hat{\beta}_0 = 41.03$ quantifiziert.

Beachte, dass sich die `Type`-Effekte zu Null summieren.

1.10.4.7 Helmert-Kontraste im faktoriellen Modell

In der Parametrisierung des faktoriellen Modells (vgl. Seite 58) sind sowohl $\beta_0, \alpha_1, \dots, \alpha_I$ linear abhängig als auch $\beta_1, \gamma_1, \dots, \gamma_I$. Das Resultat eines Fits in S-PLUS bei Reparametrisierung mit Helmert-Kontrasten ist das folgende:

```
> options( contrasts= c( "contr.helmert", "contr.poly"))
> fit5 <- lm( Mileage ~ Type * Weight, fuel.frame);      summary( fit5)
....
Coefficients:
              Value Std. Error  t value Pr(>|t|)
(Intercept)  38.8711   4.4232    8.7879  0.0000
      Type1    7.8266  10.2298    0.7651  0.4480
      Type2   -7.1136   4.2143   -1.6880  0.0979
      Type3    4.5630   2.4875    1.8343  0.0728
      Type4    4.0351   1.6292    2.4767  0.0168
      Type5   -3.0972   2.1890   -1.4149  0.1635
      Weight  -0.0053   0.0014   -3.8960  0.0003
Type1Weight  -0.0023   0.0030   -0.7705  0.4448
Type2Weight   0.0020   0.0013    1.6266  0.1104
Type3Weight  -0.0015   0.0009   -1.5923  0.1179
Type4Weight  -0.0013   0.0005   -2.4807  0.0167
Type5Weight   0.0008   0.0006    1.2697  0.2103
....
```

Auch die Bezeichnung der `Type:Weight`-Interaktionseffekte geschieht auf die für Helmert-Kontraste typische, wenig suggestive Art. Und natürlich ist wieder eine Re-Reparametrisierung nötig, um die Parameter des faktoriellen Modells identifizieren zu können:

```
> dummy.coef( fit5)
$(Intercept)":
(Intercept)
  38.87108

$Type:
 Compact   Large   Medium   Small   Sporty   Van
-6.21387  9.439407 -19.72798 12.75113 19.23744 -15.48612

$Weight:
      Weight
-0.005283773

$"Type:Weight":
 CompactWeight LargeWeight MediumWeight SmallWeight SportyWeight
  0.002262194 -0.002325607  0.006105525 -0.003850423 -0.006188107

      VanWeight
  0.003996417
```

Beachte: Sowohl die sechs `Type`-Haupteffekte als auch die sechs `Type:Weight`-Interaktionseffekte addieren sich zu Null.

1.10.4.8 Helmert-Kontraste im hierarchischen Modell

Das hierarchische Modell (vgl. Seite 58) mit linear abhängigen $\beta_0, \alpha_1, \dots, \alpha_I$ liefert in S-PLUS für Helmert-Kontraste:

```
> options( contrasts= c( "contr.helmert", "contr.poly"))
> fit6 <- lm( Mileage ~ Type / Weight, fuel.frame);      summary( fit6)
....
Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)  38.8711    4.4232    8.7879  0.0000
      Type1    7.8266   10.2298    0.7651  0.4480
      Type2   -7.1136    4.2143   -1.6880  0.0979
      Type3    4.5630    2.4875    1.8343  0.0728
      Type4    4.0351    1.6292    2.4767  0.0168
      Type5   -3.0972    2.1890   -1.4149  0.1635
TypeCompactWeight -0.0030  0.0034   -0.9011  0.3721
  TypeLargeWeight -0.0076  0.0049   -1.5464  0.1286
TypeMediumWeight  0.0008  0.0023    0.3546  0.7244
  TypeSmallWeight -0.0091  0.0030   -3.0401  0.0038
TypeSportyWeight -0.0115  0.0021   -5.5601  0.0000
  TypeVanWeight   -0.0013  0.0035   -0.3701  0.7130
....
```

Beachte, dass nur $\alpha_1, \dots, \alpha_I$ reparametrisiert wurden und nicht β_1, \dots, β_I , wie man auch an den Bezeichnungen der jeweiligen Effekte erkennen kann. Zur Re-Reparametrisierung dient wie bisher `dummy.coef()` und die Type-Haupteffekte summieren sich wieder zu Null:

```
> dummy.coef( fit6)
$(Intercept)":
(Intercept)
  38.87108

$Type:
  Compact   Large   Medium   Small   Sporty   Van
-6.21387  9.439407 -19.72798 12.75113 19.23744 -15.48612

$"Weight %in% Type":
CompactWeight LargeWeight MediumWeight SmallWeight SportyWeight
-0.003021579 -0.007609379 0.0008217528 -0.009134195 -0.01147188

VanWeight
-0.001287355
```

Beachte: Selbstverständlich gilt auch hier wieder die Äquivalenz der obigen Modelle, derzufolge sich die Parameter gemäß $\hat{\beta}_i(\text{aus fit6}) = \hat{\beta}_1(\text{aus fit5}) + \hat{\gamma}_i(\text{aus fit5})$ ineinander umrechnen lassen (in S-PLUS freilich nur im Rahmen der Rechnergenauigkeit).

1.10.5 Modell-Parametrisierung geordneter Faktoren durch Polynom-Kontraste

Für die Parametrisierung im Fall von geordneten Faktoren, d. h. von ordinalskalierten Covariablen, verwendet S-PLUS Kontrastmatrizen, die mit Hilfe von Orthonormalpolynomen erzeugt werden (eine gewöhnungsbedürftige Vorgehensweise).

Zur Erinnerung siehe Seite 61: Das Problem im Zusammenhang mit einem Faktor mit I Levels im Regressionsmodell ist, dass die Spalten seiner $n \times I$ -Inzidenzmatrix X_a und die 1-Spalte $\mathbf{1}_n$ des konstanten Terms linear abhängig sind. Die Reparametrisierung mittels einer $I \times (I - 1)$ -Kontrastmatrix C_a geschieht dergestalt, dass die Inzidenzmatrix ersetzt wird durch $X_a C_a$, sodass $\text{Rang}([\mathbf{1}_n | X_a C_a]) = I$ und das KQ-Problem somit lösbar ist. Dies ist z. B. dann garantiert, wenn die $I - 1$ Spalten von C_a zueinander und zur 1-Spalte $\mathbf{1}_n$ orthogonal sind.

Im Falle eines geordneten Faktors wird letzteres in S-PLUS durch Kontrastmatrizen erreicht, deren $I - 1$ Spalten als Orthonormalpolynome der Grade 1 bis $I - 1$ über einem Gitter von I äquidistanten Punkten interpretierbar sind. Diese zunächst etwas seltsam anmutende Strategie erlaubt es, bei der Interpretation der Kontraste und Koeffizienten des geordneten Faktors bis zu einem gewissen Grad den Charakter seiner Ordinalskalierung zu berücksichtigen.

Zur weiteren Erinnerung: Zu $z_1, z_2, \dots, z_k \in \mathbb{R}$ seien p_0, p_1, \dots, p_{k-1} Orthonormalpolynome mit $\text{Grad}(p_s) = s$ für $s = 0, 1, \dots, k - 1$. Dann gilt für $z := (z_1, \dots, z_k)$ und alle $0 \leq s, t \leq k - 1$:

$$\begin{aligned} (p_s(z))' p_t(z) &\equiv (p_s(z_1), \dots, p_s(z_k)) \begin{pmatrix} p_t(z_1) \\ \vdots \\ p_t(z_k) \end{pmatrix} \\ &= \begin{cases} 0 & , s \neq t \\ \|p_s\|^2 := \sum_{l=1}^k p_s(z_l)^2 = 1 & , s = t \end{cases} \end{aligned}$$

S-PLUS generiert die Orthonormalpolynome zur Codierung eines geordneten Faktors mit I Levels ähnlich wie jene, welche im Abschnitt 1.9 über polynomiale Regression im Zusammenhang mit der Funktion `poly()` besprochen wurden. Per Voreinstellung werden dazu $z_1 < \dots < z_I$ als äquidistant gewählt, und zwar durch $z_i := i$ für $i = 1, \dots, I$.

Definition und Eigenschaften der Kontrastmatrix eines geordneten Faktors mit I Levels: Durch $z_i := i$ (für $i = 1, \dots, I$) wird jedem Faktorlevel i ein $z_i \in \mathbb{R}$ „zugeordnet“ und dadurch ein äquidistantes Gitter $z_1 < \dots < z_I$ in \mathbb{R} definiert. Die Kontrastmatrix sei

$$C_a := \begin{pmatrix} p_1(z_1) & p_2(z_1) & \dots & p_{I-1}(z_1) \\ p_1(z_2) & p_2(z_2) & \dots & p_{I-1}(z_2) \\ \vdots & \vdots & & \vdots \\ p_1(z_I) & p_2(z_I) & \dots & p_{I-1}(z_I) \end{pmatrix}_{I \times (I-1)}$$

Dann gilt:

- Es ist $\text{Rang}(\underbrace{[\mathbf{1}_I | C_a]}_{I \times I}) = I$ und somit $\text{Rang}(\underbrace{[\mathbf{1}_n | X_a C_a]}_{n \times I}) = I$.

- Wegen $p_0 \equiv 1$ (denn $\text{Grad}(p_0) = 0$) garantiert die Orthogonalität von p_0 zu p_1, \dots, p_{I-1} , dass

$$\sum_{l=1}^I p_s(z_l) = \sum_{l=1}^I p_0(z_l) p_s(z_l) = 0 \quad \text{für jedes } s = 1, \dots, I-1$$

und somit $\mathbf{1}'_I C_a = \mathbf{0}'_{I-1}$.

- Die Orthonormalität der p_s hat zufolge, dass $C'_a C_a = \text{diag}(\{\|p_s\|^2\}_{s=1}^{I-1})$ die $(I-1) \times (I-1)$ -Einheitsmatrix ist, denn $\|p_s\|^2 \equiv 1$ für alle $s = 1, \dots, I-1$. Somit ist

$$C_a^+ = (C'_a C_a)^{-1} C'_a = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}_{(I-1) \times (I-1)}^{-1} \begin{pmatrix} p_1(z_1) & \dots & p_1(z_I) \\ p_2(z_1) & \dots & p_2(z_I) \\ \vdots & & \vdots \\ p_{I-1}(z_1) & \dots & p_{I-1}(z_I) \end{pmatrix}_{(I-1) \times I} = C'_a.$$

- Für die Kontraste α^* erhalten wir dann

$$\begin{aligned} \alpha^* = C_a^+ \alpha &\equiv \begin{pmatrix} p_1(z_1) & p_1(z_2) & \dots & p_1(z_I) \\ p_2(z_1) & p_2(z_2) & \dots & p_2(z_I) \\ \vdots & \vdots & & \vdots \\ p_{I-1}(z_1) & p_{I-1}(z_2) & \dots & p_{I-1}(z_I) \end{pmatrix}_{(I-1) \times I} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_I \end{pmatrix} \\ &= \begin{pmatrix} \sum_{l=1}^I p_1(z_l) \alpha_l \\ \sum_{l=1}^I p_2(z_l) \alpha_l \\ \vdots \\ \sum_{l=1}^I p_{I-1}(z_l) \alpha_l \end{pmatrix} \begin{matrix} \longleftarrow \text{linear} \\ \longleftarrow \text{quadratisch} \\ \vdots \\ \longleftarrow \text{Grad } I-1 \end{matrix}. \end{aligned}$$

- Die Re-Reparametrisierung lautet

$$\begin{aligned} \alpha = C_a \alpha^* &\equiv \begin{pmatrix} p_1(z_1) & p_2(z_1) & \dots & p_{I-1}(z_1) \\ p_1(z_2) & p_2(z_2) & \dots & p_{I-1}(z_2) \\ \vdots & \vdots & & \vdots \\ p_1(z_I) & p_2(z_I) & \dots & p_{I-1}(z_I) \\ \text{linear} & \text{quadrat.} & & \text{Grad } I-1 \end{pmatrix}_{I \times (I-1)} \begin{pmatrix} \alpha_1^* \\ \vdots \\ \alpha_{I-1}^* \end{pmatrix} \\ &= \begin{pmatrix} \sum_{s=1}^{I-1} \alpha_s^* p_s(z_1) \\ \vdots \\ \sum_{s=1}^{I-1} \alpha_s^* p_s(z_I) \end{pmatrix}, \end{aligned}$$

d. h., der Response-Effekt α_i des Levels i ist eine Linearkombination von Orthogonalpolynomen der Grade 1 bis $I-1$ ausgewertet an z_i (wobei $z_1 < z_2 < \dots < z_I$ äquidistant).

Bedeutung: Der Einfluss der geordneten (!) Faktorlevels lässt sich polynomial modellieren (oder z. B. auch nur linear, wenn sich $\alpha_2^*, \dots, \alpha_{I-1}^*$ alle als *nicht* signifikant verschieden von Null herausstellen sollten).

- Das reparametrisierte Modell lautet übrigens

$$Y_{ij} = \beta_0 + \alpha_1^* p_1(z_i) + \alpha_2^* p_2(z_i) + \dots + \alpha_{I-1}^* p_{I-1}(z_i) + \varepsilon_{ij}.$$

- Die Wahl der Kontraste garantiert wieder

$$\sum_{s=1}^I \alpha_s = \mathbf{1}'_I C_a \alpha^* = \mathbf{0}'_{I-1} \alpha^* = 0.$$

Ein (wenig sinnvolles) **Beispiel** für die Ausgabe von S-PLUS: Wir wandeln den Fahrzeugtyp `Type` aus `fuel.frame` um in einen geordneten Faktor `oType` mit der Level-Ordnung `Small < Compact < Sporty < Medium < Large < Van` und fitten dann die `Mileage` an `oType` und `Weight`:

```
> oType <- ordered( fuel.frame$Type, levels= c( "Small", "Compact",
+ "Sporty", "Medium", "Large", "Van"))
> fit7 <- lm( Mileage ~ oType + Weight, data= fuel.frame); summary( fit7)
....
Coefficients:
                Value Std. Error  t value Pr(>|t|)
(Intercept)  41.0276    3.9960   10.2673  0.0000
  oType.L    -2.8186    1.6887   -1.6691  0.1010
  oType.Q     0.5035    0.8112    0.6206  0.5375
  oType.C    -1.8249    0.9529   -1.9150  0.0609
  oType ^ 4    0.4055    1.0563    0.3839  0.7026
  oType ^ 5   -1.8923    0.8152   -2.3213  0.0241
  Weight     -0.0057    0.0013   -4.3583  0.0001
....
```

S-PLUS hat für den geordneten Faktor `oType` mit sechs Levels automatisch (gemäß seiner Voreinstellung) fünf polynomiale Kontraste gewählt und sie mit `oType.L`, `oType.Q`, `oType.C`, `oType ^ 4` und `oType ^ 5` benannt. Dabei deuten L, Q und C die Koeffizienten für den linearen, quadratischen bzw. kubischen Term an und die Grade 4 und aufwärts werden durch `^ n` abgekürzt.

Die Funktion `dummy.coef()` steht auch hier zur Verfügung, um α zu liefern:

```
> dummy.coef( fit7)
$(Intercept)":
(Intercept)
  41.02763

$oType:
[1]  2.8350284 -0.8223291  0.9183649 -1.0512747  0.2526246 -2.1324142

$Weight:
  Weight
-0.005697258
```

Nachrechnen bestätigt, dass sich die `oType`-Effekte, wie oben behauptet, zu Null addieren.

1.11 *F*-Tests allgemeinerer linearer Hypothesen

Im Rahmen der multiplen linearen Modelle, die uns bisher begegnet sind, haben wir unterschiedlich komplizierte Modelle kennen gelernt: Stetige Covariablen allein oder gemeinsam mit diskreten Faktor-Covariablen, ohne und mit Interaktionen. Des Weiteren sind uns inzwischen einige Möglichkeiten für die Modifikation von Modellen (`lm`-Objekten) bekannt. Allerdings können wir bei der Beurteilung, ob ein Covariablenterm einen signifikanten Einfluss auf die Response hat, bisher nur marginale Hypothesentests der Art $H_0 : \beta_i = 0$ und ihre zugehörigen *p*-Werte heranziehen (zu finden in der Ausgabe der Funktion `summary()`) bzw. Mallows C_p verwenden (gleichbedeutend mit Akaike's "information criterion" AIC und aufgetaucht in den Ausgaben der „Modellbau“-Funktionen `drop1()` und `add1()`).

Etwas allgemeinere lineare Hypothesen der Art $H_0 : \beta_{i_1} = \dots = \beta_{i_k} = 0$, in denen also mehrere Terme *gleichzeitig* auf signifikanten Einfluss getestet werden, sind jedoch auch von Interesse, z. B. wenn hierarchische Modelle miteinander verglichen werden sollen. Dabei heißen zwei Modelle M_1 und M_2 hierarchisch (auch geschachtelt bzw. Englisch "nested" genannt), wenn der Parameterraum von Modell M_1 ein Unterraum des Parameterraums von M_2 ist. Als Kurzschreibweise für diesen Sachverhalt wollen wir $M_1 \subset M_2$ verwenden und dann M_1 ein Submodell von M_2 nennen. Dies wiederum ist insbesondere dann der Fall, wenn die Menge der Covariablen von M_1 eine Teilmenge der Covariablen von M_2 ist.

Um zu prüfen, ob zwischen einem Submodell M_1 und dem (Ober-)Modell M_2 ein signifikanter Unterschied in der Beschreibung der Regressionsbeziehung zwischen Response und Covariablentermen besteht, ist im Modell M_2 die Hypothese zu testen, dass die Regressionskoeffizienten β_{i_j} derjenigen Covariablenterme von M_2 , die in M_1 nicht auftreten, alle gleich Null sind. Ein Vergleich hierarchischer Modelle im Sinne dieser Hypothese kann mit der Funktion `anova()`, die den entsprechenden *F*-Test durchführt, bewerkstelligt werden. Anhand von Beispielen wird ihre Anwendung erläutert.

1.11.1 Nur stetige Covariablen

Zunächst betrachten wir den aus früheren Abschnitten altbekannten Ozon-Datensatz, verkürzen allerdings aus Lesbarkeitsgründen die Variablenbenennungen. (Die Funktion `abbreviate()` ermittelt für die Elemente eines `character`-Vektors eindeutige Abkürzungen der (Mindest-)Länge `minlength`.)

```
> air2 <- air;   names( air2) <- abbreviate( names( air2), minlength= 2)
> air2
      oz  rd  tm  wn
1 3.448217 190 67 7.4
...
111 2.714418 223 68 11.5
```

Wir fitten das volle Modell mit allen zur Verfügung stehenden, durchweg stetigen Covariablen samt ihrer Interaktionen bis zur Ordnung drei, also

$$\begin{aligned} \mathbb{E}[\text{oz}] = & \beta_0 + \beta_1 \text{tm} + \beta_2 \text{wn} + \beta_3 \text{rd} \\ & + \beta_4 \text{tm} \cdot \text{wn} + \beta_5 \text{tm} \cdot \text{rd} + \beta_6 \text{wn} \cdot \text{rd} + \beta_7 \text{tm} \cdot \text{wn} \cdot \text{rd} . \end{aligned} \quad (9)$$

In S-PLUS:

```

> oz5.lm <- lm( oz ~ tm * wn * rd, data= air2)
> summary( oz5.lm)
....
Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept) -3.3970  2.7477    -1.2363  0.2191
           tm   0.0867  0.0376     2.3073  0.0230
           wn   0.3208  0.2263     1.4181  0.1592
           rd   0.0065  0.0150     0.4372  0.6629
        tm:wn -0.0050  0.0032    -1.5736  0.1186
        tm:rd  0.0000  0.0002    -0.2034  0.8392
        wn:rd -0.0010  0.0013    -0.7295  0.4674
    tm:wn:rd  0.0000  0.0000     0.6547  0.5141
....

```

Wir stellen fest, dass in diesem Modell bis auf `tm` kein einziger Term einen signifikanten Einfluss zu haben scheint, denn die marginalen (!) p -Werte der übrigen Terme sind alle größer als 0.1. Es liegt nahe, die marginal nicht-signifikanten vier Interaktionsterme aus dem vollen Modell `oz5.lm` zu entfernen, weil sie es unnötig (?) zu „verkomplizieren“ scheinen. Aus Anschauungsgründen wollen wir dies mit einem Zwischenschritt (über das Modell `oz6.lm`) tun, in welchem lediglich die Dreifach-Interaktion fehlt:

```

> oz6.lm <- update( oz5.lm, ~ . - tm:wn:rd)
> oz7.lm <- update( oz6.lm, ~ . - tm:wn - tm:rd - wn:rd)

```

Das heißt, dass `oz6.lm` das Modell

$$\begin{aligned} \mathbb{E}[\text{oz}] = & \beta_0 + \beta_1 \text{tm} + \beta_2 \text{wn} + \beta_3 \text{rd} \\ & + \beta_4 \text{tm} \cdot \text{wn} + \beta_5 \text{tm} \cdot \text{rd} + \beta_6 \text{wn} \cdot \text{rd} \end{aligned} \quad (10)$$

enthält. Darin sind nun die Covariablen `tm` sowie der Interaktionsterm `tm:wn` marginal signifikant und die Covariable `wn` marginal leicht signifikant:

```

> summary( oz6.lm)
....
Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept) -1.9522  1.6326    -1.1957  0.2345
           tm   0.0664  0.0212     3.1339  0.0022
           wn   0.1912  0.1091     1.7520  0.0827
           rd  -0.0026  0.0054    -0.4854  0.6284
        tm:wn -0.0032  0.0015    -2.1224  0.0362
        tm:rd  0.0001  0.0001     1.3334  0.1853
        wn:rd -0.0001  0.0002    -0.6224  0.5350
....

```

In `oz7.lm` ist das Modell

$$\mathbb{E}[\text{oz}] = \beta_0 + \beta_1 \text{tm} + \beta_2 \text{wn} + \beta_3 \text{rd}, \quad (11)$$

wobei hier alle drei Covariablen marginal signifikant sind:

```

> summary( oz7.lm)
....
Coefficients:
              Value Std. Error t value Pr(>|t|)
(Intercept) -0.2973  0.5552    -0.5355  0.5934
           tm   0.0500  0.0061     8.1957  0.0000
           wn  -0.0760  0.0158    -4.8253  0.0000
           rd   0.0022  0.0006     3.9493  0.0001
....

```

Die Funktion `anova()` erlaubt den Vergleich mehrerer, sich nicht nur in einem Term (d. h. einer Parameterdimension) unterscheidender Submodelle. Hier werden die drei hierarchischen Modelle sequenziell paarweise miteinander verglichen und die Ergebnisse in einer Varianzanalysetabelle (ANOVA-Tabelle) zusammengefasst:

```

> anova( oz5.lm, oz6.lm, oz7.lm)
Analysis of Variance Table
Response: oz

              Terms Resid. Df      RSS
1              tm * wn * rd      103 24.61160
2 tm + wn + rd + tm:wn + tm:rd + wn:rd      104 24.71404
3              tm + wn + rd      107 27.84808

              Test Df Sum of Sq  F Value      Pr(F)
1
2              -tm:wn:rd -1 -0.102435  0.428694  0.5140900
3 -tm:wn-tm:rd-wn:rd -3 -3.134046  4.372013  0.0061177

```

Diese ANOVA-Tabelle hat für jedes Modell eine Zeile, die in der Spalte **Terms** die Modellbeschreibung enthält, sowie in den Spalten **Resid. Df** und **RSS** die Residuenfreiheitsgrade bzw. die RSS des Modells ($Df = \text{“degrees of freedom”}$). Ab der zweiten Zeile stehen in den Spalten **Test**, **Df**, **Sum of Sq**, **F Value** und **Pr(F)** die Informationen über den Vergleich des Modells der jeweiligen Zeile mit dem Modell der Zeile darüber.

Interpretation der Ergebnisse:

- In Zeile 2 steht das Ergebnis des Vergleichs von Modell (9) (= `oz5.lm`) mit Modell (10) (= `oz6.lm`): Ihre Parameterräume unterscheiden sich durch den in der Spalte **Test** aufgeführten Term `tm:wn:rd`. Dies begründet die Differenz von 1 in den Parameterdimensionen (Spalte **Df**: -1). Der Test für diesen Term, d. h., der Test der Hypothese $H_0 : \beta_7 = 0$ (im Modell (9)) liefert bei einem Wert der F -Teststatistik von **F Value** = 0.428694 kein signifikantes Ergebnis: p -Wert **Pr(F)** = 0.51409. (Dies ist hier natürlich dasselbe Resultat, wie beim t -Test der Hypothese H_0 .)

Fazit: Der Dreifach-Interaktionsterm `tm:wn:rd` kann aus dem Modell eliminiert werden (was wir aber schon durch den marginalen Test wussten).

- In Zeile 3 wird Modell (10) (= `oz6.lm`) mit Modell (11) (= `oz7.lm`) verglichen: Ihre Parameterräume unterscheiden sich durch die Terme `tm:wn`, `tm:rd` und `wn:rd` (Spalte **Terms**), was zu einer Dimensionsdifferenz von 3 führt (Spalte **Df**). Der Test der dazugehörigen Hypothese $H_0 : \beta_4 = \beta_5 = \beta_6 = 0$ (im Modell (10)) dokumentiert

mit einem p -Wert von $\Pr(F) = 0.0061177$ einen signifikanten Unterschied zwischen den beiden Modellen.

Fazit: Es können nicht alle der obigen drei β s gleichzeitig Null sein. Mit anderen Worten: Mindestens einer der drei Zweifach-Interaktionsterme hat einen signifikanten Einfluss auf die Response.

Der direkte Vergleich des vollen Modells mit unserem bisher kleinsten Submodell (Dimensionsdifferenz = 4) ist natürlich ebenfalls möglich:

```
> anova( oz5.lm, oz7.lm)
Analysis of Variance Table
Response: oz
  Terms Resid. Df      RSS      Test Df Sum of Sq
1 tm*wn*rd      103 24.61160
2 tm+wn+rd      107 27.84808 -tm:wn-tm:rd-wn:rd-tm:wn:rd -4 -3.236481

  F Value      Pr(F)
1
2 3.386184 0.01204841
```

Fazit: Offenbar sind die beiden Modelle signifikant voneinander verschieden, aber an welchem (oder welchen) der vier Interaktionsterme es nun liegt, ist so nicht erkennbar.

Wenn die Reihenfolge der Terme in der Modellformel eines `lm`-Objektes eine interpretierbare Rolle spielt (eventuell wie in unseren obigen Beispielen, in denen die Terme „von links nach rechts“ immer höhere Interaktionsordnungen darstellen), *kann* das folgende Vorgehen bei der Modellanalyse behilflich sein:

Wird die Funktion `anova()` mit nur einem `lm`-Objekt als Argument aufgerufen, so fittet sie eine Sequenz von aufsteigenden Submodellen, indem sie beim Null-Modell beginnt und in der Reihenfolge der Terme in der Modellformel des `lm`-Objektes sukzessive einen Term nach dem anderen hinzufügt. Für je zwei aufeinanderfolgende Submodelle führt sie dabei den F -Test auf Signifikanz des hinzugekommenen Terms durch. Die Resultate werden in einer ANOVA-Tabelle zusammengefasst ausgegeben. Im Fall unseres Modells in `oz5.lm` erhalten wir das Folgende:

```
> anova( oz5.lm)
Analysis of Variance Table
Response: oz
Terms added sequentially (first to last)
  Df Sum of Sq Mean Sq F Value Pr(F)
tm  1  49.46178  49.46178 206.9985 0.0000000
wn  1   5.83962   5.83962  24.4389 0.0000030
rd  1   4.05928   4.05928  16.9881 0.0000762
tm:wn  1   2.48146   2.48146  10.3849 0.0017014
tm:rd  1   0.56053   0.56053   2.3458 0.1286845
wn:rd  1   0.09206   0.09206   0.3853 0.5361709
tm:wn:rd  1   0.10244   0.10244   0.4287 0.5140900
Residuals 103  24.61160   0.23895
```

Wir erkennen, dass nach dem vierten Term (`tm:wn`) kein weiterer Term mehr einen (marginal) signifikanten Beitrag liefert. Der p -Wert des zuletzt hinzugekommenen Terms `tm:wn:rd` ist natürlich gleich dem des marginalen t -Tests für diesen Term.

Beachte: Im Allgemeinen sind diese Resultate gegenüber einer Vertauschung der Reihenfolge der Terme in der Modellformel **nicht invariant!**

Obiges Resultat legt nahe, ein Modell zu fitten, in dem nur die Terme `tm`, `wn`, `rd` und `tm:wn` auftreten, also das Modell

$$\mathbb{E}[\text{oz}] = \beta_0 + \beta_1 \text{tm} + \beta_2 \text{wn} + \beta_3 \text{rd} + \beta_4 \text{tm} \cdot \text{wn}, \quad (12)$$

z. B. durch:

```
> oz6a.lm <- update( oz6.lm, ~ . - tm:rd - wn:rd)
> summary( oz6a.lm)
```

....

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	-3.6465	1.1684	-3.1209	0.0023
tm	0.0920	0.0143	6.4435	0.0000
wn	0.2523	0.1031	2.4478	0.0160
rd	0.0023	0.0005	4.3223	0.0000
tm:wn	-0.0042	0.0013	-3.2201	0.0017

....

Wir stellen fest, dass hier alle Terme marginal signifikant sind und dass ferner der direkte Vergleich dieses Modells (12) mit dem vollen Modell (9) keinen signifikanten Unterschied zwischen den beiden liefert:

```
> anova( oz5.lm, oz6a.lm)
```

Analysis of Variance Table

Response: oz

	Terms	Resid. Df	RSS	Test Df	Sum of Sq
1	tm*wn*rd	103	24.61160		
2	tm+wn+rd+tm:wn	106	25.36662	-tm:rd-wn:rd-tm:wn:rd -3	-0.7550232

	F Value	Pr(F)
1		
2	1.053262	0.3724118

Fazit: Das im Vergleich zum vollen Modell (9) in `oz5.lm` einfachere Modell (12) in `oz6a.lm` scheint eine statistisch adäquate Beschreibung der Regressionsbeziehung zwischen Ozon, Temperatur, Wind und Strahlung zu sein und die Bedingung „so einfach wie möglich und so komplex wie (statistisch) nötig“ zu erfüllen.

1.11.2 Stetige und Faktor-Covariablen

Im Fall eines Modells, das sowohl stetige Covariablen als auch Faktor-Covariablen enthält, ist die Vorgehensweise völlig analog zu der im vorherigen Abschnitt. Dies erlaubt die Beurteilung, ob eine Faktor-Covariable (sozusagen als Ganzes und nicht nur einer ihrer levelspezifischen Koeffizienten) einen signifikanten Beitrag liefert. Wir fitten als Anschauungsmaterial auf Vorrat gleich drei (hierarchische) Modelle für den bereits bekannten Datensatz in `fuel.frame` (und zwar unter Verwendung der Treatment-Kontraste):

```
> options( contrasts= c( "contr.treatment", "contr.poly" ))
> miles.lm <- lm( Mileage ~ Weight, data= fuel.frame )
> miles1.lm <- update( miles.lm, ~ . + Type )
> miles2.lm <- update( miles1.lm, ~ . + Type:Weight )
> summary( miles2.lm )
```

....

Coefficients:

	Value	Std. Error	t value	Pr(> t)
(Intercept)	32.6572	9.4757	3.4464	0.0012
Weight	-0.0030	0.0034	-0.9011	0.3721
TypeLarge	15.6533	20.4596	0.7651	0.4480
TypeMedium	-13.5141	12.0409	-1.1223	0.2673
TypeSmall	18.9650	11.6683	1.6253	0.1106
TypeSporty	25.4513	11.1192	2.2890	0.0265
TypeVan	-9.2723	15.4961	-0.5984	0.5524
WeightTypeLarge	-0.0046	0.0060	-0.7705	0.4448
WeightTypeMedium	0.0038	0.0041	0.9429	0.3505
WeightTypeSmall	-0.0061	0.0045	-1.3576	0.1809
WeightTypeSporty	-0.0085	0.0039	-2.1462	0.0369
WeightTypeVan	0.0017	0.0048	0.3589	0.7212

....

Die marginalen p -Werte bieten ein sehr heterogenes Bild: Zwar weicht zum Beispiel der Fahrzeugtyp `Sporty` signifikant von der `Mileage-Weight`-Beziehung der Bezugsgruppe `Compact` ab, aber keiner der anderen Typen. Ist der Interaktionsterm als Ganzes denn trotzdem notwendig?

Das Resultat der Funktion `anova()`, angewandt auf die Sequenz der hierarchischen Modelle, ermöglicht die Beurteilung der Beiträge der Modellterme:

```
> anova( miles.lm, miles1.lm, miles2.lm )
```

Analysis of Variance Table

Response: Mileage

	Terms	Resid. Df	RSS	Test Df	Sum of Sq
1	Weight	58	380.8322		
2	Weight+Type	53	302.9796	+Type 5	77.85265
3	Weight+Type+Type:Weight	48	215.6139	+Type:Weight 5	87.36568

	F Value	Pr(F)
1		
2	3.466314	0.009375803
3	3.889872	0.004857684

Fazit: Jeder paarweise Vergleich liefert einen signifikanten Unterschied (und auch der hier nicht gezeigte, direkte Vergleich der Modelle `miles.1m` und `miles2.1m`). Damit wird deutlich, dass sowohl die `Type`-Koeffizienten als Ganzes als auch die `Type:Weight`-Interaktionsterme als Ganzes jeweils einen signifikanten Beitrag liefern und im Modell vertreten sein sollten.

Zu erkennen ist auch, dass die Faktor-Covariable `Type` im Modell `miles1.1m` wegen der Reparametrisierung ihrer sechs Levels für fünf (!) Parameterdimensionen verantwortlich ist (Spalte `Df`). Dasselbe gilt auch nochmal für den `Type:Weight`-Interaktionsterm `Type:Weight` im Modell `miles2.1m`.

Bemerkung: Die gewählte Reparametrisierung, also die Art der Kontraste darf diese Testergebnisse natürlich nicht beeinflussen. Um dies zu überprüfen, führen Sie (zur Übung) die obige Analyse unter Verwendung der Helmert-Kontraste durch.

2 Einführung in die Varianzanalyse

In der klassischen Varianzanalyse (“analysis of variance” = ANOVA) hat man es typischerweise mit Daten aus einem geplanten Experiment zu tun, in dem die Abhängigkeit einer metrischen Response von höchstens ordinal skalierten (also diskreten) Designvariablen untersucht wird. Genauer: Die Response wird als potenziell abhängig von den Levels bzw. Levelkombinationen der Designvariablen angesehen. Die Designvariablen werden in diesem Zusammenhang *Faktorvariablen* oder kurz *Faktoren* (auch *Behandlungen*, Englisch: “treatments”) genannt. Es ist das Ziel, die mittlere Response für jeden Level oder jede Levelkombination zu schätzen und zu untersuchen, ob die Levels der Faktoren (die Behandlungsstufen) einen signifikanten Einfluss auf die mittlere Response haben.

2.1 Die einfaktorielle Varianzanalyse (“One-way-ANOVA”)

Der einfachste Fall liegt vor, wenn nur ein Faktor als Einflussvariable für eine metrische Response betrachtet wird und auf jedem seiner, sagen wir, L Levels die Response an n_l unabhängigen Untersuchungseinheiten gemessen wird, mit $l = 1, \dots, L$. Ist jedes $n_l \geq 1$ und werden die Untersuchungseinheiten (UE_n) den Faktorlevels (den Behandlungen) zufällig (*randomisiert*) zugewiesen, haben wir es mit der einfaktoriellen Varianzanalyse (oder dem Ein-Faktor-Modell) für einen vollständigen, randomisierten Versuchsplan zu tun. Die n_l werden auch Zellhäufigkeiten genannt; sind sie alle gleich ($n_l \equiv n$), so handelt es sich um einen balancierten Versuchsplan, anderenfalls um einen unbalancierten Versuchsplan.

Formal lässt sich dies alles wie folgt als das so genannte “cell means”-Modell schreiben:

$$Y_{li} = \mu_l + \varepsilon_{li} \quad \text{für } i = 1, \dots, n_l \quad \text{und } l = 1, \dots, L, \quad (13)$$

wobei Y_{li} die Response der UE i auf Faktorlevel l und μ_l die mittlere Response auf diesem Faktorlevel l ist sowie die „Fehler“ ε_{li} , d. h. die individuellen Abweichungen von μ_l , alle unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilt sind. Insgesamt liegen also $N := \sum_{l=1}^L n_l$ Beobachtungen vor. Dieser Sachverhalt kann natürlich auch in der Matrixnotation der linearen Modelle formuliert werden:

$$Y = X\mu + \varepsilon, \quad \text{wobei}$$

$$Y = \begin{pmatrix} Y_{11} \\ \vdots \\ Y_{1n_1} \\ Y_{21} \\ \vdots \\ Y_{2n_2} \\ \vdots \\ Y_{L1} \\ \vdots \\ Y_{Ln_L} \end{pmatrix}, \quad X = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}}_{N \times L}, \quad \mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_L \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{1n_1} \\ \varepsilon_{21} \\ \vdots \\ \varepsilon_{2n_2} \\ \vdots \\ \varepsilon_{L1} \\ \vdots \\ \varepsilon_{Ln_L} \end{pmatrix} \sim \mathcal{N}_N(0, \sigma^2 I_N).$$

Bemerkung: Im Prinzip könnte die Analyse nun mittels der bereits bekannten Werkzeuge der linearen Regression durchgeführt werden. Allerdings stehen spezielle, für die ANOVA maßgeschneiderte Verfahren und Routinen zur Verfügung, die dem Modell adäquatere explorative Darstellungen der Daten sowie der inferenzstatistischen Resultate liefern.

Zur Erinnerung und als Referenz: Die Kleinste-Quadrate-Schätzer (KQS) $\hat{\mu}_1, \dots, \hat{\mu}_L$ für μ_1, \dots, μ_L in obigem Modell (13) sind die Lösung des Problems

$$\varepsilon' \varepsilon \equiv \sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \mu_l)^2 \stackrel{!}{=} \text{minimal in } \mu_1, \dots, \mu_L.$$

Dies liefert $\hat{\mu}_l = \frac{1}{n_l} \sum_{i=1}^{n_l} Y_{li} =: \bar{Y}_l$ für $l = 1, \dots, L$ und $\hat{\varepsilon}_{li} := Y_{li} - \bar{Y}_l$ als Residuen. Die Residuenquadratsumme (= “residual sum of squares” = RSS) ergibt sich also zu

$$\text{RSS} = \sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \bar{Y}_l)^2. \quad (14)$$

Die Hypothese, dass der Faktor keinen Einfluss hat, lautet formal $H_0 : \mu_1 = \dots = \mu_L$ und ist äquivalent zu $\mu_1 - \mu_L = \dots = \mu_{L-1} - \mu_L = 0$, was sich als lineare Bedingung $C\mu = 0$ (an den Parametervektor μ) formulieren lässt, wobei

$$C = \begin{pmatrix} 1 & 0 & \cdots & 0 & -1 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & -1 \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix}_{(L-1) \times L} \quad \text{und} \quad \text{Rang}(C) = L - 1.$$

Bemerkung: Der Rang der Matrix C zu obiger Hypothese H_0 wird auch Anzahl der Freiheitsgrade des Faktors (bzw. der Hypothese) genannt.

Unter H_0 lautet das Modell $Y_{li} = \mu_0 + \varepsilon_{li}$ und der KQS $\hat{\mu}_0$ für μ_0 wird als Lösung von

$$\sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \mu_0)^2 \stackrel{!}{=} \text{minimal in } \mu_0$$

ermittelt, was $\hat{\mu}_0 = \frac{1}{N} \sum_{l=1}^L \sum_{i=1}^{n_l} Y_{li} =: \bar{Y}_.$ (= “overall mean”) ergibt. Die Residuenquadratsumme unter H_0 (kurz: RSS_H) ist

$$\text{RSS}_H = \sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \bar{Y}_.)^2. \quad (15)$$

Die Theorie der linearen Modelle liefert einen F -Test für H_0 , denn

$$\frac{(\text{RSS}_H - \text{RSS}) / \text{Rang}(C)}{\text{RSS} / (N - \dim(\mu))} \sim F_{\text{Rang}(C), N - \dim(\mu)} \quad \text{unter } H_0. \quad (16)$$

Bemerkung: Offenbar dürfen hierfür nicht alle $n_l \equiv n = 1$ sein, da sonst $N - \dim(\mu) = Ln - L = 0$ ist. Es muss also mindestens ein $n_l \geq 2$ sein.

Zur Bestimmung von $\text{RSS}_H - \text{RSS}$ beachte, dass $Y_{li} - \bar{Y}_. = Y_{li} - \bar{Y}_l + \bar{Y}_l - \bar{Y}_.$ ist und sich nach Quadrieren dieser Gleichung beim Summieren über alle Indices l und i die gemischten Produkte eliminieren, so dass aus (15) mit (14) folgt:

$$\text{RSS}_H = \sum_{l=1}^L \sum_{i=1}^{n_l} \hat{\varepsilon}_{li}^2 + \sum_{l=1}^L \sum_{i=1}^{n_l} (\bar{Y}_l - \bar{Y}_.)^2 = \text{RSS} + \sum_{l=1}^L n_l (\bar{Y}_l - \bar{Y}_.)^2. \quad (17)$$

Dies liefert offenbar $RSS_H - RSS$ als Summe der gewichteten Abweichungsquadrate der Faktorlevel-Mittelwerte vom Gesamtmittel:

$$RSS_H - RSS = \sum_{l=1}^L n_l (\bar{Y}_l - \bar{Y}_{..})^2. \quad (18)$$

Die obigen Größen werden oft in einer ANOVA-Tabelle zusammengefasst dargestellt:

Quelle der Streuung (source of variation)	Freiheitsgrade (degrees of freedom)	Summe der Abweichungsquadrate (sums of squares)	Mittlere Abweichungsquadrate-summe (mean squares)	F-Statistik ($\sim F_{L-1, N-L}$ unter H_0)
Zwischen den Faktorstufen (between treatments)	$L - 1$	$RSS_H - RSS = \sum_{l=1}^L \sum_{i=1}^{n_l} (\bar{Y}_l - \bar{Y}_{..})^2$	$S_H^2 = \frac{RSS_H - RSS}{L - 1}$	$F = \frac{S_H^2}{S^2}$
Innerhalb der Faktorstufen (within treatments, error, residuals)	$N - L$	$RSS = \sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \bar{Y}_l)^2$	$S^2 = \frac{RSS}{N - L}$	
Gesamtstreuung (total variation)	$N - 1$	$RSS_H = \sum_{l=1}^L \sum_{i=1}^{n_l} (Y_{li} - \bar{Y}_{..})^2$		

Zur Illustration verwenden wir als **Beispiel** nebenstehenden Datensatz (aus Box, Hunter und Hunter (1978)), der Blutgerinnungszeiten (Koagulationszeiten) in Abhängigkeit von vier verschiedenen Diätformen A, B, C und D enthält.

Diät	Koagulationszeiten							
A	62	60	63	59				
B	63	67	71	64	65	66		
C	68	66	71	67	68	68		
D	56	62	60	61	63	64	63	59

Hier hat also der Faktor „Diät“ die vier Levels A, B, C und D und die metrische Response beinhaltet Zeitdauern. (Offenbar ist es ein unbalancierter Versuchsplan.)

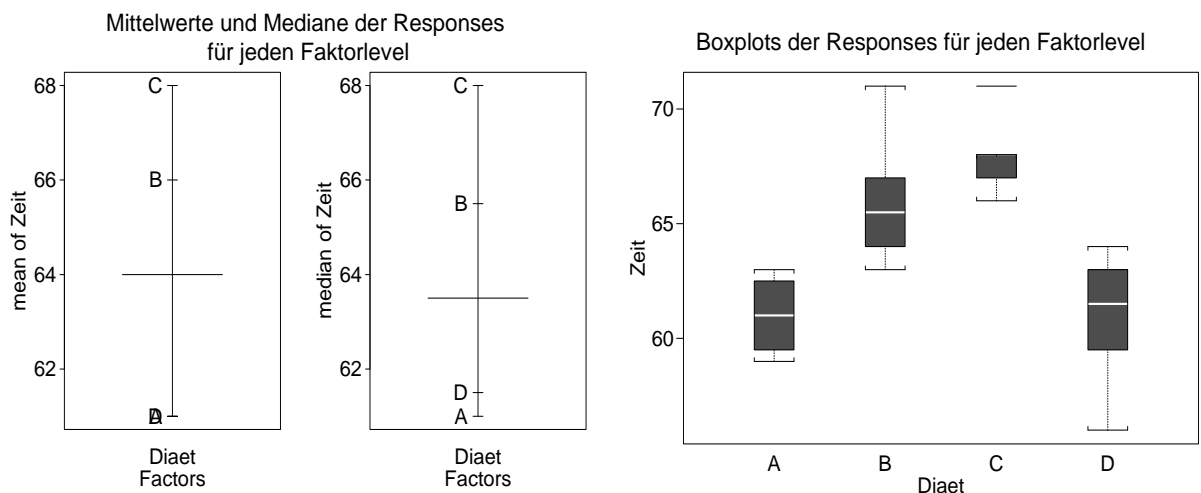
Es ist in S-PLUS nicht nötig, die in obiger Matrixnotation verwendete Designmatrix X zu spezifizieren. Vielmehr müssen der Responsevektor Y und der Vektor der zu den Y -Elementen jeweils gehörigen Faktorlevels als Spalten in einem Data Frame zusammengestellt werden. Es ist also darauf zu achten, dass die im Experiment geltende Zuordnung von beobachteten Responses und jeweiligem Faktorlevel in einer jeden Zeile des Data Frames korrekt wiedergegeben wird:

```
> Zeit <- c( 62, 60, 63, 59, 63, 67, 71, 64, 65, 66,
+ 68, 66, 71, 67, 68, 68, 56, 62, 60, 61, 63, 64, 63, 59)
> Diaet <- factor( rep( LETTERS[ 1:4], c( 4,6,6,8)));   Diaet
[1] A A A A B B B B B B C C C C C C D D D D D D D D
> Koag.df <- data.frame( Diaet, Zeit)
> Koag.df
  Diaet Zeit
```

1	A	62
2	A	60
3	A	63
...		
24	D	59

Bevor die eigentliche Varianzanalyse durchgeführt wird, sollte man sich die Daten erst einmal grafisch veranschaulichen und überprüfen, ob etwas gegen die Modellvoraussetzungen der Normalverteiltheit und Varianzkonstanz (= „Homoskedastizität“) der Fehler spricht. Dies kann mit den beiden im Folgenden beschriebenen Funktionen geschehen.

Einfaktorielle Varianzanalyse: Explorative Datenanalyse	
<pre>> plot.design(Koag.df)</pre>	Liefert einen Plot, in dem sowohl für jeden Faktorlevel (voreinstellungsgemäß) das arithmetische Mittel der zugehörigen Responses (durch einen kurzen waagrechten Strich und das Label des jeweiligen Levels) als auch das Gesamtmittel der Responses (durch einen längeren waagrechten Strich) markiert sind. Dies erlaubt eine erste Einschätzung, inwieweit die Levels die mittlere Response beeinflussen (linker Plot unten).
<pre>> plot.design(Koag.df, + fun= median)</pre>	Wie eben, aber das Argument <code>fun= median</code> veranlasst, dass Mediane als Lageparameter verwendet und markiert werden. Weicht dieser Plot von demjenigen mit den arithmetischen Mitteln nicht zu stark ab, ist das ein Hinweis darauf, dass keine Ausreißer in den Responses vorliegen (mittlerer Plot unten).
<pre>> plot.factor(Koag.df)</pre>	Erzeugt einen Plot, in dem für jeden Faktor-Level ein Boxplot der zugehörigen Responses gezeichnet ist. Dies ermöglicht eine Beurteilung sowohl der Normalverteiltheit und der Varianzhomogenität der Fehler ε über alle Levels hinweg als auch der Abhängigkeit der mittleren Responses von den Faktorlevels (rechter Plot unten).



Im vorliegenden Beispiel sprechen die Ergebnisse der EDA nicht gegen die Modellannahmen: Weder legt der Vergleich der Designplots links die Existenz von potenziellen

Ausreißern nahe, noch liefern die Faktor-Boxplots rechts deutliche Indizien gegen die Annahme der Varianzkonstanz oder der Normalverteilung.

Die Varianzanalyse wird mit dem Funktionsaufruf `aov(formula, data)` durchgeführt (wobei `aov` für “analysis of variance” steht). Das Argument `formula` spezifiziert in der bereits bekannten Formelsyntax die Beziehung zwischen Response und Design und `data` den Data Frame, aus dem die Variablen entnommen werden. Das Resultat ist ein so genanntes `aov`-Objekt. Die Erstellung der Ergebnistabelle (ANOVA-Tabelle) wird durch die Anwendung von `summary()` auf das `aov`-Objekt veranlasst:

Einfaktorielle Varianzanalyse: Die ANOVA-Tabelle					
<pre>> Koag.aov <- aov(Zeit ~ Diaet, data= Koag.df)</pre>					
<pre>> summary(Koag.aov)</pre>					
	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Diaet	3	228	76.0	13.57143	4.658471e-05
Residuals	20	112	5.6		

Hier wird das einfaktorielle Modell von `Zeit` an `Diaet` (aus Data Frame `Koag.df`) gefittet und in `Koag.aov` abgelegt. Mit `summary(Koag.aov)` erhält man die ANOVA-Tabelle. In deren Zeile `Diaet` stehen die $L - 1$ Freiheitsgrade dieses Faktors (Spalte `Df`, “degrees of freedom”), die Summe der Abweichungsquadrate der Faktorlevel-Mittelwerte vom Gesamtmittel, also $RSS_H - RSS$ (Spalte `Sum of Sq`), und deren mittlere Quadratesumme $(RSS_H - RSS)/(L - 1)$ (Spalte `Mean Sq`). In der Zeile `Residuals` stehen für die Residuen ihre $N - L$ Freiheitsgrade, ihre Quadratesumme RSS und ihre mittlere Quadratesumme $RSS/(N - L)$. Außerdem wird der Wert der F-Teststatistik (Spalte `F Value`) und der p -Wert (Spalte `Pr(F)`) des Tests auf Gleichheit der mittleren Faktorlevel-Responses angegeben, also des Tests der Hypothese $H_0 : \mu_1 = \dots = \mu_L$.

Zur Bestimmung der KQS $\hat{\mu}_1, \dots, \hat{\mu}_L$ für μ_1, \dots, μ_L ist zu sagen, dass sich das in (13) auf Seite 83 formulierte cell means-Modell auch anders parametrisieren lässt, und zwar als Faktoreffekte-Modell. Darin wird ein Gesamtmittel (oder auch „Populationsmittel“) μ_0 für die Response angesetzt, von dem sich die mittleren Responses der verschiedenen Faktorlevels durch additive Faktoreffekte α_l unterscheiden:

$$Y_{li} = \mu_0 + \alpha_l + \varepsilon_{li} \quad \text{für } i = 1, \dots, n_l \quad \text{und } l = 1, \dots, L, \quad (19)$$

wobei aus Identifizierbarkeitsgründen an diese Effekte die Bedingung $\sum_{l=1}^L n_l \alpha_l = 0$ gestellt wird. α_l wird der Effekt des l -ten Faktorlevels genannt. Offenbar ist Modell (19) ohne Nebenbedingung an die α_l überparametrisiert.

Bemerkung: Im Prinzip ist jede lineare Nebenbedingung der Art $\sum_{l=1}^L d_l \alpha_l = 0$ mit $\sum_{l=1}^L d_l \neq 0$ möglich (vgl. z. B. Seber (1977), Abschnitt 9.1.6, S. 247). Jedoch garantiert $d_l = n_l$, dass die orthogonale Zerlegung (17) auf Seite 84 stets gültig ist (vgl. z. B. Seber (1977), Abschnitt 9.1.9, S. 250f). Aus diesem Grund verwendet S-PLUS die Nebenbedingung $\sum_{l=1}^L n_l \alpha_l = 0$. Im balancierten Fall ($n_l \equiv n$) vereinfacht sich alles erheblich.

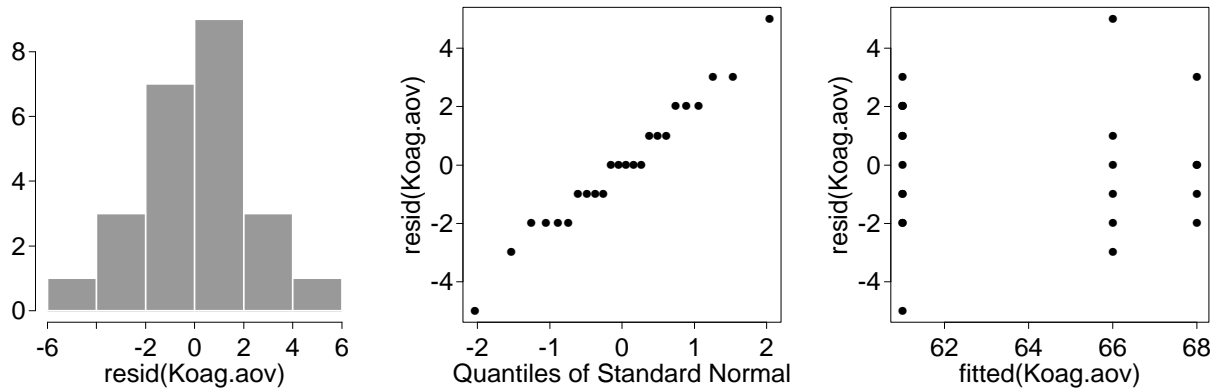
Die Funktion `aov()` fittet das einfaktorielle Modell in der Faktoreffekte-Form (19), wobei es bei ungeordneten (also nominalen) Faktoren aus Identifizierbarkeitsgründen – wie bei der linearen Regression – intern zu einer Reparametrisierung mittels der *Helmert-Kontraste* kommt (was man aber nicht merkt). Die Schätzer $\hat{\alpha}_l$ für die Effekte im Faktoreffekte-Modell oder die Schätzer $\hat{\mu}_l$ für die Faktorlevel-Mittelwerte im cell means-Modell können dann je nach Bedarf mit Hilfe der Funktion `model.tables()` ermittelt werden:

Einfaktorielle Varianzanalyse: Die Parameterschätzer	
<pre>> model.tables(Koag.aov) Tables of effects Diaet A B C D -3 2 4 -3 rep 4 6 6 8 Warning messages: Model was refit to allow projection in: model.tables(Koag.aov) > model.tables(Koag.aov, type= "means") Tables of means Grand mean 64 Diaet A B C D 61 66 68 61 rep 4 6 6 8 Warning messages: Model was refit to allow projection in: model.tables(Koag.aov, type = "means")</pre>	<p>Liefert (voreinstellungsgemäß) die Schätzer $\hat{\alpha}_l$ für die Effekte im Faktoreffekte-Modell (also Modell (19)). Man entnimmt außer den Schätzwerten für die Effekte (= Abweichungen vom Gesamtmittel) aller Levels auch die Anzahl (<code>rep</code>) der Beobachtungen auf jedem Level. Beachte: $\sum_{l=1}^L n_l \hat{\alpha}_l = 0$ ist erfüllt.</p> <p>Das Argument <code>type= "means"</code> bewirkt, dass (anstelle der Effekteschätzer) sowohl der Schätzer $\hat{\mu}_0$ für das Gesamtmittel μ_0 als auch die Schätzer $\hat{\mu}_l$ für die Faktorlevel-Mittelwerte im cell means-Modell (also Modell (13)) sowie die Anzahl (<code>rep</code>) der Beobachtungen auf jedem Level ausgegeben werden.</p>
<p>Bemerkung: Die <code>Warning messages</code> sind problemlos. Durch die Verwendung des Arguments <code>projections= T</code> im Aufruf von <code>aov()</code> würde erreicht, dass die in <code>model.tables()</code> mit "Model was refit to allow projection" gemeinte Prozedur nicht mehr nötig ist. Die Rechenzeit würde verkürzt.</p>	

Nach dem Fit des Modells können (und sollten) die Modellannahmen mit Hilfe der Residuen und der gefitteten Werte untersucht werden. Dies geschieht, wie in der linearen Regression auch, durch Histogramme und Q-Q-Plots der Residuen zur Beurteilung der Normalverteilungsannahme bzw. Plots der Residuen gegen die gefitteten Werte, um die Varianzhomogenität zu checken:

Einfaktorielle Varianzanalyse: Diagnoseplots	
<pre>> fitted(Koag.aov) > resid(Koag.aov) > hist(resid(Koag.aov)) > qqnorm(resid(Koag.aov)) > plot(fitted(Koag.aov), + resid(Koag.aov))</pre>	<p>An die gefitteten Werte (= geschätzte mittlere Responses) und die Residuen kommt man wieder durch die Funktionen <code>fitted()</code> bzw. <code>resid()</code>.</p> <p>Eine Beurteilung der Normalverteilungsannahme der Fehler ist durch ein Histogramm und ein Q-Q-Plot der Residuen möglich (siehe linker und mittlerer Plot auf der nächsten Seite oben).</p> <p>Um die Varianzhomogenität zu checken, sollten die Residuen gegen die gefitteten Werte geplottet werden (siehe rechter Plot auf der nächsten Seite oben).</p>

ANOVA-Diagnoseplots im einfaktoriellen Modell



2.2 Die zweifaktorielle Varianzanalyse (“Two-way-ANOVA”)

Hier haben wir es mit einer metrischen Response zu tun, die für die Levelkombinationen *zweier* Faktoren A und B an unabhängigen Untersuchungseinheiten gemessen wird. Wird jede Levelkombination mindestens einmal beobachtet und werden die UEn den Faktorlevelkombinationen zufällig (*randomisiert*) zugewiesen, handelt es sich um die zweifaktorielle Varianzanalyse (oder das Zwei-Faktoren-Modell) für einen vollständigen, randomisierten Versuchsplan.

Faktor A habe die Levels $j = 1, \dots, J$ und Faktor B die Levels $l = 1, \dots, L$ und die Response werde für jede Levelkombination (j, l) an $n_{jl} \geq 1$ unabhängigen UEn beobachtet. Dann lässt sich dies als cell means-Modell wie folgt schreiben:

$$Y_{jli} = \mu_{jl} + \varepsilon_{jli} \quad \text{für } i = 1, \dots, n_{jl} \quad \text{und } j = 1, \dots, J, \quad l = 1, \dots, L, \quad (20)$$

wobei Y_{jli} die Response der UE i für die Faktorlevelkombination (j, l) und μ_{jl} die mittlere Response hierfür ist. Die „Fehler“ ε_{jli} , d. h. die individuellen Abweichungen von μ_{jl} sind auch hier alle unabhängig und identisch $\mathcal{N}(0, \sigma^2)$ -verteilt. Insgesamt liegen $N := \sum_{j=1}^J \sum_{l=1}^L n_{jl}$ Beobachtungen vor. Die n_{jl} werden auch hier Zellhäufigkeiten genannt und es wird zwischen balancierten ($n_{jl} \equiv n$) und unbalancierten Versuchsplänen unterschieden.

Werden die Y_{jli} und ε_{jli} zueinander passend in N -dimensionale Vektoren $Y = (Y_{111}, \dots, Y_{JLn_{JL}})'$ bzw. $\varepsilon = (\varepsilon_{111}, \dots, \varepsilon_{JLn_{JL}})'$ „gestapelt“ und die μ_{jl} geeignet in einem JL -dimensionalen Vektor $\mu = (\mu_{11}, \dots, \mu_{JL})'$ zusammengefasst, existiert eine $N \times JL$ -Matrix X , so dass obiger Sachverhalt in Matrixnotation als $Y = X\mu + \varepsilon$ formuliert werden kann. Details sparen wir uns hier jedoch.

Bemerkung zur Notation: Zur Abkürzung von Summen und arithmetischen Mitteln werden wir von der folgenden, häufig anzutreffenden Notation Gebrauch machen: Für ein mehrfach indiziertes Schema, wie z. B. x_{jli} mit $j = 1, \dots, J$, $l = 1, \dots, L$ und $i = 1, \dots, n_{jl}$, bedeutet ein Punkt „.“ anstelle eines Indexes, dass die (marginale) Summe der x_{jli} über diesen Index gebildet wurde. Beispiele:

$$x_{.li} = \sum_{j=1}^J x_{jli} \quad \text{oder} \quad x_{.l.} = \sum_{j=1}^J \sum_{i=1}^{n_{jl}} x_{jli} \quad \text{oder} \quad x_{...} = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} x_{jli}.$$

Analog werden entsprechende (marginale) arithmetische Mittel bezeichnet:

$$\bar{x}_{\cdot li} = \frac{1}{J} \sum_{j=1}^J x_{jli} \equiv \frac{x_{\cdot li}}{J} \quad \text{oder} \quad \bar{x}_{\cdot l} = \frac{1}{n_{\cdot l}} \sum_{j=1}^J \sum_{i=1}^{n_{jl}} x_{jli} \equiv \frac{x_{\cdot l}}{n_{\cdot l}}$$

$$\text{oder} \quad \bar{x}_{\dots} = \frac{1}{n_{\dots}} \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} x_{jli} \equiv \frac{x_{\dots}}{n_{\dots}}.$$

Zur Erinnerung und als Referenz: Die Kleinste-Quadrate-Schätzer (KQS) $\hat{\mu}_{jl}$ für μ_{jl} (für $j = 1, \dots, J$ und $l = 1, \dots, L$) in obigem Modell (20) sind die Lösung von

$$\varepsilon' \varepsilon \equiv \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} (Y_{jli} - \mu_{jl})^2 \stackrel{!}{=} \text{minimal in allen } \mu_{jl}.$$

Dies liefert $\hat{\mu}_{jl} = \bar{Y}_{jl}$ und $\hat{\varepsilon}_{jli} := Y_{jli} - \bar{Y}_{jl}$ als Residuen. Die Residuenquadratsumme (RSS) ergibt sich also zu

$$\text{RSS} = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} (Y_{jli} - \bar{Y}_{jl})^2. \quad (21)$$

Die Hypothese, dass *beide* Faktoren keinen Einfluss haben, lautet $H_0 : \mu_{11} = \dots = \mu_{JL}$ und lässt sich mit der $(JL - 1) \times JL$ -Matrix $C = [I_{JL-1} | -\mathbf{1}_{JL-1}]$ vom Rang $JL - 1$ als eine lineare Bedingung $C\mu = 0$ (an den Parametervektor μ) formulieren. Unter H_0 vereinfacht sich das Modell zu $Y_{jli} = \mu_0 + \varepsilon_{jli}$ und der KQS $\hat{\mu}_0$ für μ_0 wird als Lösung von

$$\sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} (Y_{jli} - \mu_0)^2 \stackrel{!}{=} \text{minimal in } \mu_0$$

zu $\hat{\mu}_0 = \bar{Y}_{\dots}$ ermittelt. Die Residuenquadratsumme unter H_0 ist demnach

$$\text{RSS}_H = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^{n_{jl}} (Y_{jli} - \bar{Y}_{\dots})^2$$

und völlig analog zum einfaktorischen Modell erhalten wir (durch Teleskopieren gemäß $Y_{jli} - \bar{Y}_{\dots} = Y_{jli} - \bar{Y}_{jl} + \bar{Y}_{jl} - \bar{Y}_{\dots}$ und anschließendes Quadrieren sowie Summieren):

$$\text{RSS}_H - \text{RSS} = \sum_{j=1}^J \sum_{l=1}^L n_{jl} (\bar{Y}_{jl} - \bar{Y}_{\dots})^2. \quad (22)$$

Damit ist H_0 unter Verwendung von (21) und (22) mit Hilfe der F -Teststatistik in (16) testbar, wobei $\dim(\mu) = JL$ und $\text{Rang}(C) = JL - 1$.

Bemerkung: Die Formulierung weiterer, interessanter Hypothesen, die die beiden Faktoren A und B separat bzw. ihre Interaktion betreffen, lässt sich in obigem cell means-Modell (20) ebenfalls durchführen, gestaltet sich in der im Folgenden beschriebenen Parametrisierung als Faktoreffekte-Modell jedoch etwas anschaulicher. Allerdings unterscheiden sich im Zwei-Faktoren-Modell balancierter und unbalancierter Fall in ihrer formalen Komplexität stärker als im Ein-Faktoren-Modell. Der Einfachheit halber beschränken wir uns daher auf den balancierten Versuchsplan.

2.2.1 Der balancierte Versuchsplan

Im balancierten Versuchsplan ist $n_{jl} \equiv n$, so dass sich Modell (20) zu

$$Y_{jli} = \mu_{jl} + \varepsilon_{jli} \quad \text{für } i = 1, \dots, n, \quad j = 1, \dots, J, \quad l = 1, \dots, L \quad (23)$$

leicht vereinfacht und mithin $N = JLn$ ist. Es sei $n > 1$, da Inferenz sonst nicht möglich ist.

Zur Formulierung von Hypothesen hinsichtlich der separaten Einflüsse der Faktoren A und B auf die mittlere Response bzw. ihrer Interaktion ist es hilfreich, das cell means-Modell (23) als Faktoreffekte-Modell mit Interaktionsterm wie folgt zu parametrisieren:

$$Y_{jli} = \mu_0 + \alpha_j + \beta_l + (\alpha\beta)_{jl} + \varepsilon_{jli} \quad \text{für } i = 1, \dots, n, \quad j = 1, \dots, J, \quad l = 1, \dots, L, \quad (24)$$

wobei μ_0 das Gesamtmittel ist, α_j der Effekt des j -ten Levels von Faktor A, β_l der Effekt des l -ten Levels von Faktor B und $(\alpha\beta)_{jl}$ der Interaktionseffekt beim j -ten Level von Faktor A und l -ten Level von Faktor B. (Dabei hat die Notation „ $(\alpha\beta)_{jl}$ “ nichts mit einer Multiplikation zu tun, sondern soll nur darauf hinweisen, zu welchen Faktoren dieser Interaktionsterm gehört; er könnte z. B. auch γ_{jl} heißen.)

Modell (24) ist überparametrisiert und um die Identifizierbarkeit der Parameter zu gewährleisten, wird gefordert, dass sie die folgende Bedingung erfüllen:

$$\alpha. = \beta. = (\alpha\beta)_{.l} = (\alpha\beta)_{.j} = 0 \quad \text{für alle } l = 1, \dots, L \text{ und } j = 1, \dots, J. \quad (25)$$

Bemerkungen:

1. Die Beziehungen zwischen den beiden Parametrisierungen (23) und (24) unter dieser Identifizierungsbedingung lassen sich ableiten, indem die rechten Seiten von (23) und (24) gleichgesetzt und geeignet summiert werden. Zum Beispiel ist

$$\mu_{..} = LJ\mu_0 \quad \text{und somit} \quad \mu_0 = \bar{\mu}_{..}$$

oder für ein beliebiges $1 \leq j \leq J$ ist

$$\mu_{.j} = L\mu_0 + L\alpha_j \quad \text{und somit} \quad \alpha_j = \bar{\mu}_{.j} - \mu_0 = \bar{\mu}_{.j} - \bar{\mu}_{..}$$

Analog folgt $\beta_l = \bar{\mu}_{.l} - \bar{\mu}_{..}$ bzw. schließlich $(\alpha\beta)_{jl} = \mu_{jl} - \bar{\mu}_{.j} - \bar{\mu}_{.l} + \bar{\mu}_{..}$

2. Die Faktoreffekte-KQS $\hat{\mu}_0$, $\hat{\alpha}_j$, $\hat{\beta}_l$ und $\widehat{(\alpha\beta)}_{jl}$ für alle j und l ergeben sich durch Lösen der jeweiligen Minimierungsprobleme unter Beachtung der Identifizierungsbedingung. Es zeigt sich, dass man sie durch Einsetzen der cell means-KQS $\hat{\mu}_{jl} = \bar{Y}_{jl}$ in die Beziehungen zwischen den Parametrisierungen (23) und (24) erhält. Die folgende Tabelle enthält diese Beziehungen sowie die entsprechenden KQS:

Parameterbeziehungen	Kleinste-Quadrate-Schätzer
$\mu_0 = \bar{\mu}_{..}$	$\hat{\mu}_0 = \bar{Y}_{..}$
$\alpha_j = \bar{\mu}_{.j} - \bar{\mu}_{..}$	$\hat{\alpha}_j = \bar{Y}_{.j} - \bar{Y}_{..}$
$\beta_l = \bar{\mu}_{.l} - \bar{\mu}_{..}$	$\hat{\beta}_l = \bar{Y}_{.l} - \bar{Y}_{..}$
$(\alpha\beta)_{jl} = \mu_{jl} - \bar{\mu}_{.j} - \bar{\mu}_{.l} + \bar{\mu}_{..}$	$\widehat{(\alpha\beta)}_{jl} = \bar{Y}_{jl} - \bar{Y}_{.j} - \bar{Y}_{.l} + \bar{Y}_{..}$

3. Zur Interpretation von „Interaktion“ sowie der Effekte α_j und β_l :
Die folgende, aus Hocking (1996, S. 437) adaptierte Tabelle erleichtert, auf cell means, marginale und nicht-marginale Faktoreffekte zu verweisen und die Beziehungen zwischen ihnen zu verdeutlichen:

		Faktor B				
		1	2	3	4	
Faktor A	1	μ_{11}	μ_{12}	μ_{13}	μ_{14}	$\bar{\mu}_{1\cdot}$
	2	μ_{21}	μ_{22}	μ_{23}	μ_{24}	$\bar{\mu}_{2\cdot}$
	3	μ_{31}	μ_{32}	μ_{33}	μ_{34}	$\bar{\mu}_{3\cdot}$
		$\bar{\mu}_{\cdot 1}$	$\bar{\mu}_{\cdot 2}$	$\bar{\mu}_{\cdot 3}$	$\bar{\mu}_{\cdot 4}$	$\bar{\mu}_{\cdot\cdot}$

Eine Interaktion der Faktoren A und B liegt vor, wenn für zwei A-Levels $j_1 \neq j_2$ der Wirkungsunterschied zwischen ihnen, also $\mu_{j_1 l} - \mu_{j_2 l}$, vom Level l des Faktors B abhängt. (Entsprechend für zwei B-Levels $l_1 \neq l_2$ und ein A-Level j .) Ist dies *nicht* der Fall, so folgt leicht, dass $(\alpha\beta)_{jl} = 0$ für alle j, l .

Betrachten wir einen Faktor separat: Unter Einflusslosigkeit von, z. B., A wird verstanden, dass – egal, welches B-Level l gegeben ist – die A-Levels stets dieselbe Wirkung haben und diese, wenn überhaupt, nur von l abhängt: $\mu_{jl} = \theta(l)$. Es folgt $\mu_{jl} - \bar{\mu}_{\cdot l} = 0$ für alle j, l . Nun muss jedoch unterschieden werden:

Besteht *keine* A-B-Interaktion (ist also $(\alpha\beta)_{jl} \equiv 0$), so ist Einflusslosigkeit von A äquivalent zu $\alpha_j = 0$ für alle j , d. h., zu $\bar{\mu}_{j\cdot} \equiv \bar{\mu}_{\cdot\cdot}$. (Analog folgt, dass Einflusslosigkeit von B dann äquivalent zu $\beta_l = 0$ für alle l ist, d. h., zu $\bar{\mu}_{\cdot l} \equiv \bar{\mu}_{\cdot\cdot}$.)

Liegt hingegen A-B-Interaktion vor, ist $\alpha_j \equiv 0$ *nicht* äquivalent zu $\mu_{jl} - \bar{\mu}_{\cdot l} \equiv 0$ und bedeutet dann lediglich, dass Faktor A *im Durchschnitt über alle B-Levels* (also im marginalen Sinne) keinen Einfluss hat. (Entsprechendes gilt für $\beta_l \equiv 0$.)

Kommen wir nun zurück zur Frage nach interessanten Hypothesen. Im Faktoreffekte-Modell lassen sich (z. B.) die drei folgenden angeben:

- $H_{AB} : (\alpha\beta)_{jl} = 0$ für alle $j = 1, \dots, J$ und $l = 1, \dots, L$.
Bedeutung: Keine A-B-Interaktion, d. h., der Einfluss des einen Faktors hängt nicht vom anderen Faktor ab.
- $H_A : \alpha_1 = \dots = \alpha_J = 0$.
Bedeutung: Falls H_{AB} erfüllt: Faktor A ohne Einfluss. Falls H_{AB} nicht erfüllt: Faktor A im Durchschnitt über alle B-Levels ohne Einfluss (= ohne marginalen Einfluss).
- $H_B : \beta_1 = \dots = \beta_L = 0$.
Bedeutung: Analog zu H_A .

Obschon sich jede der vier Hypothesen (H_0 und die obigen drei) bei angemessener Interpretation separat betrachten lässt, ist es nahe liegend, erst H_0 , dann H_{AB} und schließlich H_A oder/und H_B zu testen. Kann H_0 nicht verworfen werden, brauchen die übrigen gar nicht in Betracht gezogen zu werden. Wird H_{AB} verworfen, müssen H_A und H_B im marginalen Sinne interpretiert werden. Falls H_{AB} nicht verworfen werden kann, sind H_A und H_B äquivalent zu den „strengeren“ Hypothesen $H_{A_0} : \mu_{jl} = \mu_{jL}$ für alle j, l bzw. $H_{B_0} : \mu_{jl} = \mu_{jL}$ für alle j, l .

Jede der drei umseitigen Hypothesen lässt sich (mit Hilfe der Parameterbeziehungen von Seite 91 unten) unter Verwendung einer geeigneten Hypothesenmatrix C als eine lineare Bedingung an den cell means-Vektor $\mu = (\mu_{11}, \dots, \mu_{JL})'$ schreiben. Um eine solche Hypothese $H_* : C\mu = 0$ zu testen, müssen – wie üblich – die Residuenquadratsumme RSS_{H_*} unter dieser Hypothese H_* und der Rang von C bestimmt werden, damit die bekannte F -Teststatistik aus (16) zur Anwendung kommen kann (mit der RSS aus (21) und im balancierten Fall in leicht einfacherer Form).

Bemerkung: Der Rang der Matrix C wird je nach Hypothese Freiheitsgrad des Faktors bzw. des Interaktionseffektes (oder kurz: Freiheitsgrad der Hypothese) genannt.

Wir tabellieren die für die betrachteten vier Hypothesen notwendigen Ergebnisse – für den balancierten Fall – (eine Herleitung findet sich z. B. in Seber (1977, Abschnitt 9.2.2)):

Hypothese	$SS_* := RSS_{H_*} - RSS$	Rang(C) (= Freiheitsgrade)
H_0	$SS_0 := n \sum_{j=1}^J \sum_{l=1}^L (\bar{Y}_{jl\cdot} - \bar{Y}_{\dots})^2$	$JL - 1$
H_{AB}	$SS_{AB} := n \sum_{j=1}^J \sum_{l=1}^L (\widehat{(\alpha\beta)})_{jl}^2$	$(J - 1)(L - 1)$
H_A	$SS_A := nL \sum_{j=1}^J \hat{\alpha}_j^2$	$(J - 1)$
H_B	$SS_B := nJ \sum_{l=1}^L \hat{\beta}_l^2$	$(L - 1)$

Bemerkung: Die Interpretation der in den obigen Tests auftretenden Größen wird durch die folgenden Überlegungen erleichtert: Die Abweichung einer Beobachtung Y_{jli} vom Gesamtmittelwert \bar{Y}_{\dots} (also vom KQS im Modell ohne jegliche Faktoreinflüsse $Y_{jli} = \mu_0 + \varepsilon_{jli}$) kann wie folgt in die Abweichung des Faktorstufen-Mittelwertes vom Gesamtmittelwert und die Abweichung der Beobachtung vom Faktorstufen-Mittelwert zerlegt werden:

$$Y_{jli} - \bar{Y}_{\dots} = \bar{Y}_{jl\cdot} - \bar{Y}_{\dots} + \underbrace{Y_{jli} - \bar{Y}_{jl\cdot}}_{\equiv \hat{\varepsilon}_{jli}}$$

Weiter wird $\bar{Y}_{jl\cdot} - \bar{Y}_{\dots}$ (= Abweichung des Faktorstufen-Mittelwertes vom Gesamtmittelwert) in die Anteile der Faktoreffekte und des Interaktionseffekts zerlegt:

$$\bar{Y}_{jl\cdot} - \bar{Y}_{\dots} = \underbrace{\bar{Y}_{j\cdot\cdot} - \bar{Y}_{\dots}}_{\text{A-Effekt}} + \underbrace{\bar{Y}_{\cdot l\cdot} - \bar{Y}_{\dots}}_{\text{B-Effekt}} + \underbrace{\bar{Y}_{jl\cdot} - \bar{Y}_{j\cdot\cdot} - \bar{Y}_{\cdot l\cdot} + \bar{Y}_{\dots}}_{\text{AB-Interaktionseffekt}} \equiv \hat{\alpha}_j + \hat{\beta}_l + \widehat{(\alpha\beta)}_{jl}$$

Entsprechend der obigen Zerlegungen kann die Gesamtstreuung in den Beobachtungen

$$SS_{Total} = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^n (Y_{jli} - \bar{Y}_{\dots})^2 \quad (26)$$

in Anteile der Effekte, des Interaktionseffekts und der (durch das Modell nicht weiter erklärbaren) Residuen zerlegt werden, da sich beim Summieren der quadrierten Zerlegung die gemischten Produkte eliminieren. (Dies gilt jedoch nur, da wir im balancierten Fall sind; im Allgemeinen ist es falsch!) Man erhält:

$$SS_{Total} = \underbrace{nL \sum_{j=1}^J \hat{\alpha}_j^2}_{\equiv SS_A} + \underbrace{nJ \sum_{l=1}^L \hat{\beta}_l^2}_{\equiv SS_B} + \underbrace{n \sum_{j=1}^J \sum_{l=1}^L \widehat{(\alpha\beta)}_{jl}^2}_{\equiv SS_{AB}} + \underbrace{\sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^n \hat{\varepsilon}_{jli}^2}_{\equiv RSS} \quad (27)$$

Diese “sums of squares” werden üblicherweise wieder in einer ANOVA-Tabelle dokumentiert:

Streuungsquelle (source of variation)	Freiheitsgrade (degrees of freedom)	Summe der Abweichungsquadrate (sums of squares)	Mittlere Abweichungsquadratsumme (mean squares)	F-Stat. (H_*)
Zwischen den Faktorstufen (between treatments)	$JL - 1$	$SS_0 = n \sum_{j=1}^J \sum_{l=1}^L (\bar{Y}_{jl} - \bar{Y}_{...})^2$	$S_0^2 = \frac{SS_0}{JL - 1}$	$\frac{S_0^2}{S^2}$ (H_0)
Faktor-A-Haupteffekte (A main effects)	$J - 1$	$SS_A = nL \sum_{j=1}^J \hat{\alpha}_j^2$	$S_A^2 = \frac{SS_A}{J - 1}$	$\frac{S_A^2}{S^2}$ (H_A)
Faktor-B-Haupteffekte (B main effects)	$L - 1$	$SS_B = nJ \sum_{l=1}^L \hat{\beta}_l^2$	$S_B^2 = \frac{SS_B}{L - 1}$	$\frac{S_B^2}{S^2}$ (H_B)
A-B-Interaktionseffekte (AB interactions)	$(J-1)(L-1)$	$SS_{AB} = n \sum_{j=1}^J \sum_{l=1}^L \widehat{(\alpha\beta)}_{jl}^2$	$S_{AB}^2 = \frac{SS_{AB}}{(J-1)(L-1)}$	$\frac{S_{AB}^2}{S^2}$ (H_{AB})
Innerhalb der Faktorstufen (residuals, ...)	$JL(n-1)$	$RSS = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^n (Y_{jli} - \bar{Y}_{jl})^2$	$S^2 = \frac{RSS}{JL(n-1)}$	
Gesamtstreuung (total variation)	$JLn - 1$	$SS_{Total} = RSS_{H_0} = \sum_{j=1}^J \sum_{l=1}^L \sum_{i=1}^n (Y_{jli} - \bar{Y}_{...})^2$		

Wir verwenden als **Beispiel** wieder einen Datensatz aus Box, Hunter und Hunter (1978): Nebenstehend sind Überlebenszeiten (in Einheiten zu zehn Stunden) angegeben, die im Rahmen eines Tierexperiments für die Kombination von drei Giftstoffen I, II und III und vier verschiedenen Behandlungsstoffen A, B, C und D ermittelt wurden. Für jede Giftstoff-Behandlungsstoff-Kombination wurden vier Tiere verwendet, also vier Wiederholungen durchgeführt. (Offenbar ein balancierter Versuchsplan.)

Gift	Behandlung			
	A	B	C	D
I	0.31	0.82	0.43	0.45
	0.45	1.10	0.45	0.71
	0.46	0.99	0.63	0.66
	0.43	0.72	0.76	0.62
II	0.36	0.92	0.44	0.56
	0.29	0.61	0.35	1.02
	0.40	0.49	0.31	0.71
	0.23	1.24	0.40	0.38
III	0.22	0.30	0.23	0.30
	0.21	0.37	0.25	0.36
	0.18	0.38	0.24	0.31
	0.23	0.29	0.22	0.33

Der Vektor Y aller Responses und die zwei Vektoren der dazugehörigen Faktorlevels sind als Spalten in einem Data Frame zusammenzufassen, und zwar dergestalt, dass jede Zeile des Data Frames eine beobachtete Response und die hierfür gültige Faktorlevel-Kombination enthält. Dabei ist die Funktion `fac.design()` behilflich, die aus verschiedenen Faktorlevel-Informationen den Versuchsplan eines Experiments als Data Frame nach-

bildet (siehe unten): Zunächst erzeugen wir eine Liste, deren Komponenten die Namen der Faktoren haben (hier „Behandlung“ bzw. „Gift“) und die jeweiligen Faktorlevels enthalten (hier A, B, C, D bzw. I, II, III). Die Funktion `fac.design()` generiert aus der notwendigen Angabe der Levels-Anzahlen (`levels`) und jener (optionalen) Liste (`factor.names`) einen Data Frame, der alle Levelkombinationen der beiden Faktoren als „Grund-Design“ enthält (hier 12-zeilig; nicht separat gezeigt). Um die (balancierte!) Wiederholung einer jeden Levelkombination zu erreichen, steht das Argument `replications` zur Verfügung, das angibt, wie oft das Grund-Design wiederholt und einfach untereinander gehängt wird (hier vier Mal, weswegen 48 Zeilen resultieren):

```
> Faktornamen <- list( Behandlung= LETTERS[ 1:4], Gift= c( "I", "II", "III"))
> Gift.design <- fac.design( levels= c( 4,3), factor.names= Faktornamen,
+ replications= 4);      Gift.design
  Behandlung Gift
1           A   I
2           B   I
3           C   I
4           D   I
5           A  II
6           B  II
7           C  II
8           D  II
9           A III
10          B III
11          C III
12          D III
13          A   I
.....
48          D  III
```

Nun muss noch der Vektor der Response-Werte (als Komponente `Survival`) an den Design Data Frame angefügt werden. Dabei ist darauf zu achten, dass die Reihenfolge der Vektorelemente zum Arrangement des Versuchsplans in `Gift.design` passt:

```
> Survival <- c(
+ 0.31, 0.82, 0.43, 0.45,    0.36, 0.92, 0.44, 0.56,    0.22, 0.30, 0.23, 0.30,
+ 0.43, 0.72, 0.76, 0.62,    0.23, 1.24, 0.40, 0.38,    0.23, 0.29, 0.22, 0.33)
> Gift.df <- data.frame( Gift.design, Survival);      Gift.df
  Behandlung Gift Survival
1           A   I      0.31
2           B   I      0.82
3           C   I      0.43
4           D   I      0.45
5           A  II      0.36
.....
8           D  II      0.56
9           A III      0.22
.....
12          D III      0.30
13          A   I      0.45
.....
48          D III      0.33
```

Wie in Abschnitt 2.1 werden für die explorative Datenanalyse wieder `plot.design()` und `plot.factor()` verwendet, aber nun kommt auch noch eine Methode zur Entdeckung möglicher Interaktionen zum Einsatz:

- `plot.design()` erlaubt eine erste Beurteilung, wie die Levels des einen Faktors auf die über die Levels des anderen Faktors gemittelte Response wirken. Konkret werden dazu die marginalen Mittelwerte $\bar{Y}_{j..}$ für $j = 1, \dots, J$ auf einer vertikalen Skala markiert, ebenso $\bar{Y}_{.l}$ für $l = 1, \dots, L$ sowie der Gesamtmittelwert $\bar{Y}_{...}$. Dies geschieht in einem gemeinsamen Koordinatensystem (siehe linker oberer Plot auf nächster Seite). (Dabei ist $\bar{Y}_{j..}$ der KQS für $\bar{\mu}_{j.}$, $\bar{Y}_{.l}$ für $\bar{\mu}_{.l}$ und $\bar{Y}_{...}$ für $\bar{\mu}_{..} = \mu_{0.}$)
- `plot.factor()` erlaubt darüber hinaus die Beurteilung der Homoskedastizitätsannahme, da für jedes Level des einen Faktors ein Boxplot aller dazugehörigen (über die Levels des anderen Faktors gepoolten) Response-Werte geplottet wird. Konkret wird für jedes $j = 1, \dots, J$ ein Boxplot für $\{Y_{jli} : l = 1, \dots, L, i = 1, \dots, n\}$ gezeichnet, wobei diese J Boxplots in ein gemeinsames Koordinatensystem kommen; ebensolches geschieht für jedes $l = 1, \dots, L$ für $\{Y_{jli} : j = 1, \dots, J, i = 1, \dots, n\}$ (vgl. mittlere Plots auf nächster Seite).

Mithin lassen sich durch die beiden obigen Methoden die Faktoreffekte separat begutachten, allerdings im marginalen Sinne.

- Für die Darstellung potenzieller Interaktionseffekte steht ebenfalls ein exploratives Werkzeug zur Verfügung: Der so genannte Interaktionsplot. Zu seiner Motivation ist zu beachten, dass unter der Hypothese keiner Interaktion (also unter $H_{AB} : (\alpha\beta)_{jl} \equiv 0$) gemäß der entsprechenden Beziehung auf Seite 91 unten gilt:

$$\mu_{jl} = \bar{\mu}_{j.} + \bar{\mu}_{.l} - \mu_{...}$$

Daraus folgt für feste $l \neq l'$ und beliebiges j (bzw. für feste $j \neq j'$ und beliebiges l):

$$\mu_{jl} - \mu_{jl'} = \bar{\mu}_{.l} - \bar{\mu}_{.l'} \quad (\text{bzw.} \quad \mu_{jl} - \mu_{j'l} = \bar{\mu}_{j.} - \bar{\mu}_{j'.})$$

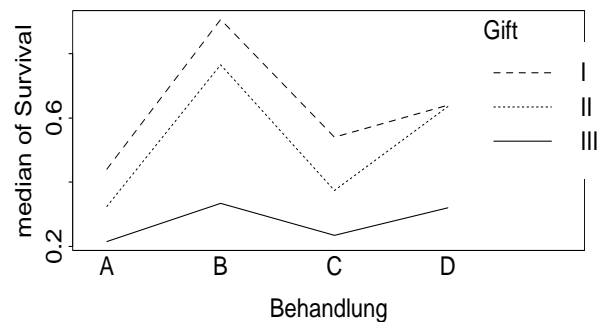
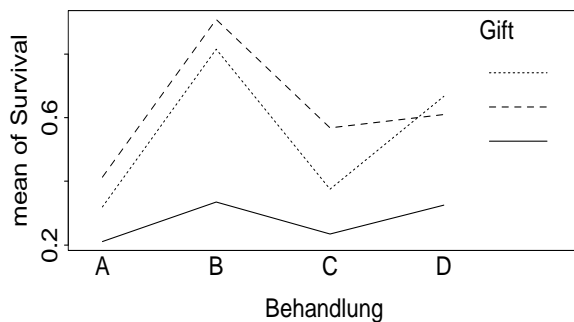
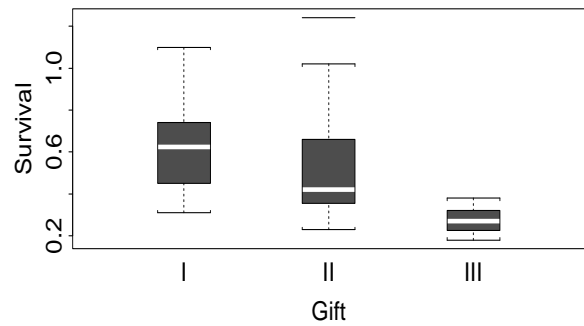
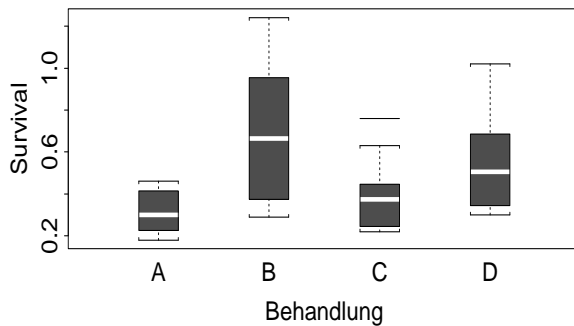
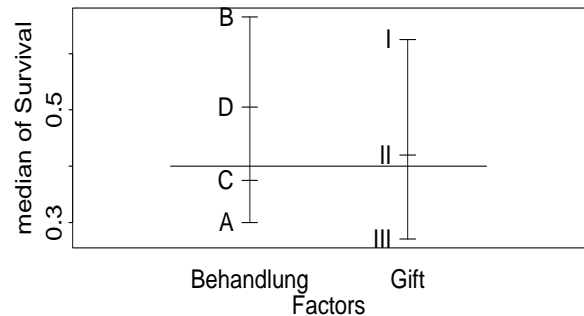
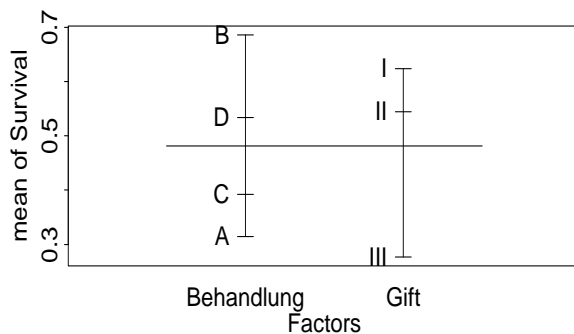
Expressis verbis: Der Wirkungsunterschied zwischen den B-Faktorlevels l und l' (also der Abstand von μ_{jl} zu $\mu_{jl'}$) ist entlang der „j-Achse“ konstant (nämlich gleich $\bar{\mu}_{.l} - \bar{\mu}_{.l'}$). Die Wirkungsprofile $(\mu_{1l}, \dots, \mu_{Jl})$ und $(\mu_{1l'}, \dots, \mu_{Jl'})$ des Faktors A sind also „parallel“ für zwei verschiedene B-Levels, und damit *alle* Profile von Faktor A. Eben dies gilt analog für die Profile $(\mu_{j1}, \dots, \mu_{jL})$ des Faktors B.

Dieser Sachverhalt sollte sich in einem Plot widerspiegeln, in dem für jedes Level $l = 1, \dots, L$ des Faktors B die empirischen Profile $(\bar{Y}_{1l}, \dots, \bar{Y}_{Jl})$ des Faktors A gegen $j = 1, \dots, J$ gezeichnet werden. (Es kommen also die KQS \bar{Y}_{jl} für μ_{jl} zum Einsatz.) Indem jedes der L A-Profil durch einen eigenen Polygonzug repräsentiert wird, sollte unter H_{AB} – in etwa – Parallelität dieser Polygonzüge zu beobachten sein. Völlig analog gilt dies für einen Plot der J B-Profil.

Diese Darstellung ist ein so genannter Interaktionsplot. Er wird mit Hilfe der Funktion `interaction.plot()` erstellt. Als ihr erstes Argument erwartet sie den Vektor des Faktors, dessen Profile gezeichnet werden sollen, d. h., dessen Levels auf der waagrechten Achse abzutragen sind, als zweites den Vektor des Faktors, für dessen Levels die Profile des ersten Faktors gebildet werden, und als drittes den Vektor der Response-Werte.

Die folgende Tabelle demonstriert im vorliegenden Gift-Beispiel die Erstellung der drei oben beschriebenen und unten abgebildeten, explorativen Plot-Typen:

Zweifaktorielle Varianzanalyse: Explorative Datenanalyse	
<pre>> plot.design(Gift.df) > plot.design(Gift.df, fun= median) > plot.factor(Gift.df) > attach(Gift.df) > interaction.plot(Behandlung, Gift, + Survival) > interaction.plot(Behandlung, Gift, + Survival, fun= median) > detach("Gift.df")</pre>	<p>Liefert den Mittelwert basierten Designplot (links oben) bzw. Median basierten Designplot (rechts oben). Erstellt die zwei Faktorplots (mittlere „Zeile“). Der Interaktionsplot links unten zeigt die zuvor beschriebenen, Mittelwert basierten Profile; derjenige rechts unten Median basierte, was durch fun= median erreicht wird. (attach() erleichtert nur die Notation.)</p>



Das qualitative Fazit der explorativen Datenanalyse lautet:

- Die Designplots zeigen für die Behandlung-Levels A und C kürzere und für D und B längere mittlere bzw. mediane Überlebenszeiten. Die Gift-Levels I, II und III verringern in dieser Reihenfolge die mittleren bzw. medianen Überlebenszeiten.

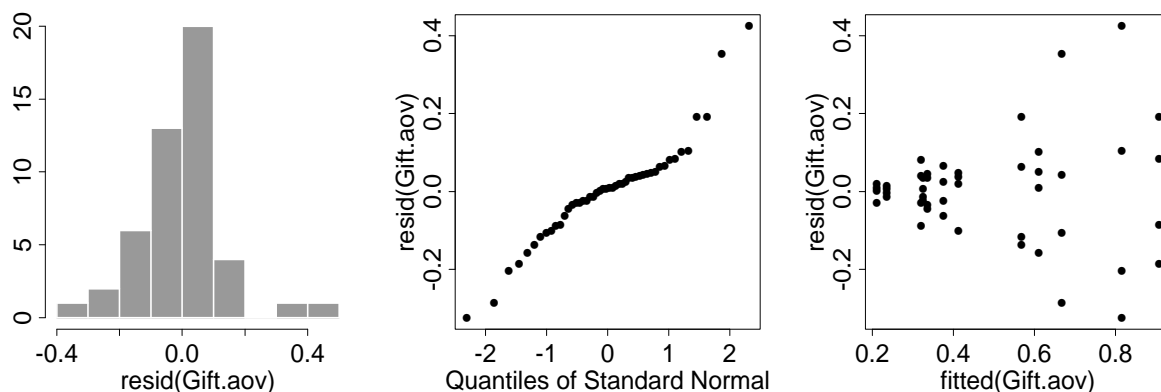
Der Vergleich von “mean”- und “median”-Designplots zeigt keine starken Unterschiede. Lediglich im Gift-Level II deutet der Unterschied zwischen arithmetischem Mittel und Median auf eine schiefe Verteilung oder (mindestens) einen potenziellen Ausreißer hin.

- Die Boxplots deuten klar auf eine größere Response-Variabilität bei höherem Response-Median hin (und somit auf eine mögliche Varianzhomogenität).
- Die nicht parallelen Verläufe in den Interaktionsplots zeigen stärkere Effekte der Behandlung bei den Giften I und II (welche diejenigen mit den höheren mittleren Überlebensdauern bzw. deren Medianen sind) als bei Gift III, was auf eine Interaktion der beiden Faktoren hindeutet.

Das Zwei-Faktoren-Modell wird als Modell (24) durch die Funktion `aov()` gefittet, wobei es bei nominalen Faktoren intern (wie `stets`) zu einer Reparametrisierung mittels der *Helmert-Kontraste* kommt, allerdings ohne, dass es sich für uns bemerkbar macht. `summary()` liefert eine ANOVA-Tabelle und danach ist Modelldiagnose angesagt.

Zwei-Faktoren-Modell: Die ANOVA-Tabelle					
<pre>> Gift.aov <- aov(Survival ~ Gift * Behandlung, data= Gift.df) > summary(Gift.aov)</pre>					
	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Gift	2	1.063904	0.5319521	23.64815	0.00000028
Behandlung	3	0.965367	0.3217889	14.30526	0.00000269
Gift:Behandlung	6	0.265996	0.0443326	1.97083	0.09576069
Residuals	36	0.809800	0.0224944		
<p>Es wird das zweifaktorielle Modell mit Interaktion (24) von <code>Survival</code> an <code>Gift</code> und <code>Behandlung</code> (aus <code>Gift.df</code>) gefittet und in <code>Gift.aov</code> abgelegt. Mit <code>summary(Gift.aov)</code> erhält man eine ANOVA-Tabelle, in der, wie an der Zeilenbenennung zu erkennen, die Zerlegung (27) dargestellt ist (aber ohne SS_0): In der Spalte <code>Df</code> stehen die Freiheitsgrade der zwei Faktoren (also $J-1$ bzw. $L-1$), der Interaktion ($((J-1)(L-1))$) und der <code>RSS</code> ($N-JL$). Unter <code>Sum of Sq</code> stehen die Summen der Abweichungsquadrate aus (27) ($= SS_* \equiv RSS_{H_*} - RSS$ bzw. <code>RSS</code>). Daneben, in Spalte <code>Mean Sq</code>, befinden sich die mittleren Summen dieser Quadrate (d. h. $S_*^2 = SS_*/\text{Freiheitsgrade}$ bzw. $S^2 = RSS/(N-JL)$). Es folgen die Werte der F-Teststatistiken S_*^2/S^2 (<code>F Value</code>) der Hypothesen „Kein Effekt“ (also H_A sowie H_B) bzw. „Keine Interaktion“ (H_{AB}) und deren p-Werte (<code>Pr(F)</code>). (Der Test von H_0 wird nicht dokumentiert.)</p>					
Zwei-Faktoren-Modell: Diagnoseplots					
<pre>> fitted(Gift.aov) > resid(Gift.aov) > hist(resid(Gift.aov)) > qqnorm(resid(Gift.aov)) > plot(fitted(Gift.aov), + resid(Gift.aov))</pre>			<p>An die gefitteten Werte (= geschätzte mittlere Responses für jede Levelkombination) und die Residuen kommt man mit <code>fitted()</code> bzw. <code>resid()</code>, was die Beurteilung der Normalverteilungsannahme der Fehler durch ein Histogramm und einen Q-Q-Plot der Residuen sowie die Prüfung der Varianzhomogenität durch einen Plot der Residuen gegen die gefitteten Werte ermöglicht. (Nächste Seite oben.)</p>		

ANOVA-Diagnoseplots im zweifaktoriellen Modell



Der Plot der Residuen gegen die gefitteten Werte offenbart eine deutliche Varianzheterogenität, was die inferenzstatistischen Ergebnisse zweifelhaft erscheinen lässt. Hier wäre eine varianzstabilisierende Transformation angezeigt (Stichwort „Box-Cox-Transformation“), worauf wir aber nicht näher eingehen, sondern dazu auf die Literatur verweisen (wie z. B. auf Seber (1971, ch. 6.7), auf Neter, Wasserman und Kuttner (1990, pp. 149-150), auf Fox (2002, ch. 3.4) oder die Originalarbeit “An analysis of transformations” von Box und Cox im J. R. Stat. Soc. B (26), 1964).

Zur expliziten Ermittlung von $\hat{\alpha}_j$, $\hat{\beta}_l$ und $\widehat{(\alpha\beta)}_{jl}$ als Schätzwerte für die Effekte dient schließlich wieder `model.tables()`:

Zwei-Faktoren-Modell: Die Parameterschätzer	
<pre>> model.tables(Gift.aov) Tables of effects Gift I II III 0.1427 0.06271 -0.2054 Behandlung A B C D -0.1675 0.2042 -0.08917 0.0525 Gift:Behandlung Dim 1 : Gift Dim 2 : Behandlung A B C D I -0.0444 0.0790 0.0323 -0.0669 II -0.0569 0.0665 -0.0802 0.0706 III 0.1012 -0.1454 0.0479 -0.0037</pre>	<p>Liefert (voreinstellungsgemäß) die KQS für die Effekte im Faktoreffekte-Modell (24):</p> <p>$\hat{\alpha}_1, \dots, \hat{\alpha}_J$</p> <p>$\hat{\beta}_1, \dots, \hat{\beta}_L$</p> <p>$\widehat{(\alpha\beta)}_{11} \dots \widehat{(\alpha\beta)}_{1L}$ \vdots $\widehat{(\alpha\beta)}_{J1} \dots \widehat{(\alpha\beta)}_{JL}$</p> <p>Wie man schnell nachrechnet, ist die Identifizierungsbedingung (25) bis auf Rundungsfehler erfüllt.</p>
<p>Bemerkung: Die Warning messages, wie sie durch <code>model.tables()</code> auf S. 88 ausgegeben wurden, sind hier nicht aufgetreten, da wir, ohne es auf der vorherigen Seite zu dokumentieren, das Argument <code>projections= T</code> im Aufruf von <code>aov()</code> verwendet haben. Dasselbe gilt für die Ausgabe auf der nächsten Seite.</p>	

(Forts.: Zwei-Faktoren-Modell: Die Parameterschätzer)	
<pre>> model.tables(Gift.aov, type= + "means") Tables of means Grand mean 0.4817 Gift I II III 0.6244 0.5444 0.2763 Behandlung A B C D 0.3142 0.6858 0.3925 0.5342 Gift:Behandlung Dim 1 : Gift Dim 2 : Behandlung A B C D I 0.4125 0.9075 0.5675 0.6100 II 0.3200 0.8150 0.3750 0.6675 III 0.2100 0.3350 0.2350 0.3250</pre>	<p>Das Argument <code>type= "means"</code> er- wirkt, dass KQS für das cell means- Modell (23) ausgegeben werden:</p> <p>$\bar{Y}_{..}$ ($= \hat{\mu}_0$) für das Gesamtmittel $\bar{\mu}_{..}$ ($= \mu_0$; "Grand mean"),</p> <p>$\bar{Y}_{1..}, \dots, \bar{Y}_{J..}$ für die marginalen (!) Faktorlevel-Mittelwerte $\bar{\mu}_{1.}, \dots, \bar{\mu}_{J.}$ (wobei $\bar{\mu}_{j.} = \mu_0 + \alpha_j$),</p> <p>$\bar{Y}_{.1}, \dots, \bar{Y}_{.L}$ für die marginalen (!) Faktorlevel-Mittelwerte $\bar{\mu}_{.1}, \dots, \bar{\mu}_{.L}$ (wobei $\bar{\mu}_{.l} = \mu_0 + \beta_l$) sowie</p> <p>$\bar{Y}_{11.} \dots \bar{Y}_{1L.}$ $\vdots \quad \quad \quad \vdots$ für die cell means $\bar{Y}_{J1.} \dots \bar{Y}_{JL.}$ $\mu_{jl} = \mu_0 + \alpha_j + \beta_l + (\alpha\beta)_{jl}.$</p>

2.2.2 Genau eine Beobachtung pro Levelkombination

Wir gehen hier nicht ins Detail, sondern machen nur einige Bemerkungen zum Fall $n = 1$ und liefern eine paar Beispiele:

- Im Modell *mit* Interaktionsterm ($Y \sim A * B$) ist keine Inferenz möglich, da $RSS = 0$. Es existiert allerdings ein approximatives Modell, innerhalb dessen ein exakter Test auf „Keine Interaktion“ möglich ist (vgl. Hocking (1996), Abschnitt 13.1.5) und somit geklärt werden kann, ob für vorliegende Daten nicht ein rein additives Modell (also $Y \sim A + B$) ausreichend ist.

Die S-PLUS-Funktion `aov()` fittet zwar auch im Fall $n = 1$ das zweifaktorielle Modell mit Interaktion ohne zu murren, aber in der ANOVA-Tabelle als Resultat von `summary()` finden sich korrekterweise keine Inferenzresultate:

Beispiel: Das Argument `subset` der Funktion `aov()` erlaubt es, eine Teilmenge der Zeilen des an `data` übergebenen Data Frames zu spezifizieren, um alle Berechnungen nur auf Basis der Daten jener Zeilen durchzuführen. Hier werden nur die ersten 12 Zeilen von `Gift.df` ausgewählt und demnach ein Versuchsplan mit genau einer Beobachtung für jede Faktorlevel-Kombination. Dafür wird dann ein Modell mit Interaktion gefittet:

```
> Gift.B1I.aov <- aov( Survival ~ Gift * Behandlung, data= Gift.df,
+ subset= 1:12)
```

```
> summary( Gift.B1I.aov)
              Df Sum of Sq   Mean Sq
      Gift      2   0.20895 0.1044750
  Behandlung    3   0.25030 0.0834333
Gift:Behandlung  6   0.08485 0.0141417
```

- Im Modell *ohne* Interaktionsterm ($Y \sim A + B$), also dem rein additiven Modell, ist Inferenz bezüglich der Faktoreffekte möglich und die `summary` des `aov`-Objektes enthält auch die jeweiligen p -Werte:

```
> Gift.B1A.aov <- aov( Survival ~ Gift + Behandlung, data= Gift.df,
+ subset= 1:12)
> summary( Gift.B1A.aov)
              Df Sum of Sq   Mean Sq  F Value      Pr(F)
      Gift      2   0.20895 0.1044750  7.387743 0.02408797
Behandlung    3   0.25030 0.0834333  5.899823 0.03193500
Residuals    6   0.08485 0.0141417
```

- Die Tatsache, nur eine Beobachtung pro Faktorlevel-Kombination zu haben und deswegen keinen Interaktionstest durchführen zu können, rechtfertigt nicht die Annahme des Fehlens einer Interaktion! Diese Annahme muss durch fachliche Überlegungen begründet sein. Die Interaktionsplots erlauben aber wenigstens eine qualitative Einschätzung des Vorhandenseins oder Fehlens von Interaktion. Im Modell des nächsten Abschnitts wird implizit davon ausgegangen, dass keine Interaktion vorhanden ist.

2.3 Das einfache, randomisierte Blockexperiment

Bei den bisherigen Betrachtungen sind wir davon ausgegangen, dass die in den Analysen betrachtete Population an Untersuchungseinheiten (UEn) homogen ist und sich Unterschiede, wenn überhaupt, dann nur auf Grund der Behandlungsstufen manifestieren. Gelegentlich werden die Behandlungen aber auf bekanntermaßen inhomogene Mengen von UEn angewendet. Und wie überall in der Statistik ist ein (Behandlungs-)Effekt schwer zu entdecken, wenn die Variabilität der Response relativ zur Größe des Effektes zu groß ist; man sagt, dass der Effekt „maskiert“ wird. Konkret im Fall der Versuchspläne kann eine große Varianz innerhalb der Faktorlevels (= Behandlungsstufen) den Einfluss der Faktoren maskieren. Ist jedoch eine Störgröße bekannt, die diese Variabilität mitverantwortet, aber selbst nur von nachrangigem Interesse ist, kann dies wie folgt zu einer Verbesserung der Analyse genutzt werden. Wir beschränken uns hier auf den Fall *eines* interessierenden Faktors mit L Levels und einer Störgröße mit J verschiedenen Ausprägungen:

Angenommen, man hat $J \cdot L$ UEn, so dass für jede Störgrößen-Ausprägung genau L UEn vorliegen. Dann teilt man die UEn in J Gruppen, genannt Blöcke, gleicher Störgrößen-Ausprägung ein. (Daher wird die Störgröße auch Blockbildungsfaktor (“blocking factor”) genannt.) Innerhalb eines jeden Blocks kann nun jeder der L Faktorlevels an genau einer UE zur Anwendung kommen. Werden die UEn innerhalb der Blöcke zufällig (also *randomisiert*) den Faktorlevels zugewiesen, so wird dies als einfaches, randomisiertes Blockexperiment bezeichnet. (Beachte: Eine randomisierte Zuweisung von UEn zu den Blöcken ist

nicht möglich, da Störgrößen-Ausprägungen unveränderliche Eigenschaft der UEn sind!)

Das einfache Blockexperiment lässt sich als ein Zwei-Faktoren-Effekte-Modell ohne Interaktion mit einer Beobachtung pro Levelkombination parametrisieren:

$$Y_{jl} = \mu_0 + \alpha_j + \beta_l + \varepsilon_{jl} \quad \text{für } j = 1, \dots, J \quad \text{und} \quad l = 1, \dots, L, \quad (28)$$

wobei μ_0 wieder das Gesamtmittel ist, α_j der Effekt des j -ten Levels des Blockbildungsfaktors und β_l der Effekt des l -ten Levels des Behandlungsfaktors. Die Effekte müssen auch hier wieder Identifizierbarkeitsbedingungen genügen, wie

$$\alpha_{\cdot} = 0 \quad \text{und} \quad \beta_{\cdot} = 0.$$

Die Inferenz bezüglich eines Einflusses des Behandlungsfaktors läuft in diesem Modell also auf den Test der Hypothese $H_A : \alpha_1 = \dots = \alpha_J = 0$ im zweifaktoriellen Modell ohne Interaktion hinaus.

Es zeigt sich ferner, dass Parameterbeziehungen und dazugehörige KQS von Modell (28) analog zu denen sind, die in der Tabelle auf Seite 91 unten aufgelistet sind, jedoch ohne Interaktionsterme und ohne dritten Index. Daraus folgt, dass die Abweichung einer Beobachtung Y_{jl} vom Gesamtmittelwert $\bar{Y}_{\cdot\cdot}$ (also vom KQS im Modell ohne jeglichen Faktor- oder Blockeinfluss) wie folgt zerlegt werden kann:

$$Y_{jl} - \bar{Y}_{\cdot\cdot} = \underbrace{\bar{Y}_{j\cdot} - \bar{Y}_{\cdot\cdot}}_{\text{Blockeffekt}} + \underbrace{\bar{Y}_{\cdot l} - \bar{Y}_{\cdot\cdot}}_{\text{Behandlungseffekt}} + \underbrace{Y_{jl} - \bar{Y}_{j\cdot} - \bar{Y}_{\cdot l} + \bar{Y}_{\cdot\cdot}}_{\text{RSS}} \equiv \hat{\alpha}_j + \hat{\beta}_l + \hat{\varepsilon}_{jl}.$$

Entsprechend zerlegt sich die Gesamtstreuung der Beobachtungen in Anteile der Block- und Faktoreffekte und der (durch das Modell nicht erklärten) Residuen, da sich beim Summieren der quadrierten Gleichung die gemischten Produkte eliminieren. Man erhält:

$$SS_{Total} = \sum_{j=1}^J \sum_{l=1}^L (Y_{jl} - \bar{Y}_{\cdot\cdot})^2 = \underbrace{L \sum_{j=1}^J \hat{\alpha}_j^2}_{\equiv SS_{Block}} + \underbrace{J \sum_{l=1}^L \hat{\beta}_l^2}_{\equiv SS_{Beh.}} + \underbrace{\sum_{j=1}^J \sum_{l=1}^L \hat{\varepsilon}_{jl}^2}_{\equiv RSS}. \quad (29)$$

Diese “sums of squares” werden in der ANOVA-Tabelle des einfachen Blockexperiments dokumentiert. (Offenbar sind SS_{AB} und RSS aus (27) zur RSS in (29) „verschmolzen“.) Die in der Tabelle auf Seite 91 unten aufgeführten “sums of squares” SS_A und SS_B nebst ihrer Freiheitsgrade kommen wieder in der F -Teststatistik (16) zum Einsatz.

Als **Beispiel** dient uns erneut ein Datensatz aus Box, Hunter und Hunter (1978): Er dokumentiert den Ertrag eines gewissen Penicillin-Produktionsprozesses für vier verschiedene Verfahrenstypen (= Faktorlevels, Behandlungsstufen) A, B, C und D. Als eine weitere Einflussgröße wurde die Sorte des verwendeten Rohstoffs “corn-steep liquor” ausgemacht, wovon es fünf verschiedene Sorten gibt. Hier stellt sich der Faktor „Sorte“ als Störvariable dar, da man in erster Linie am Einfluss des Faktors „Verfahrenstyp“ auf den Penicillin-Ertrag interessiert ist. Die Sorte dient somit zur Blockbildung für die registrierten Ertragsmengen.

Von jeder der fünf verschiedenen Sorten “corn-steep liquor” wurde eine Ladung in vier Teile unterteilt, die randomisiert den Verfahrenstypen zugewiesen wurden. Es konnte dann jeweils genau ein Durchlauf des Produktionsprozesses durchgeführt und der Ertrag bestimmt werden. Wir haben es also mit einem einfachen, randomisierten Blockexperiment zu tun, dessen Ergebnisse in der folgenden Tabelle zu sehen sind:

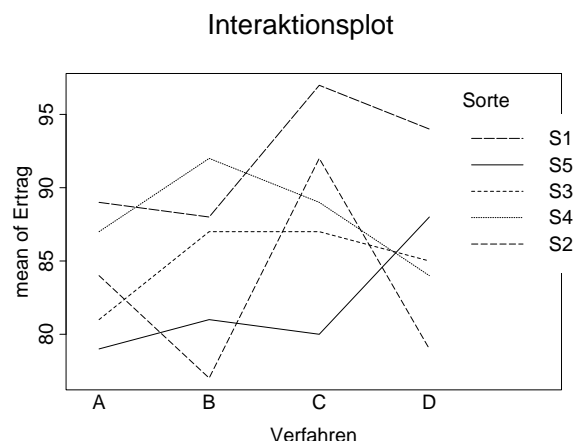
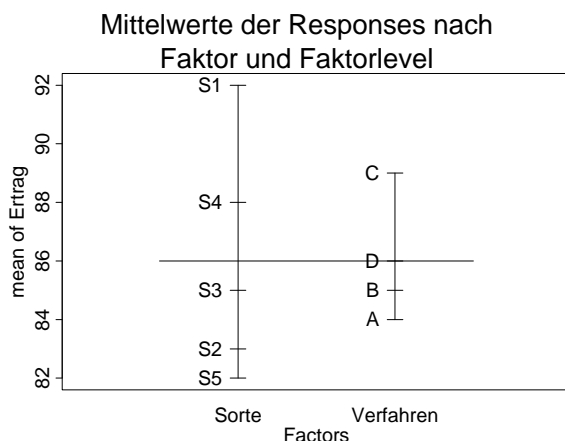
		Sorten (= Blöcke)				
		Sorte 1	Sorte 2	Sorte 3	Sorte 4	Sorte 5
Ver- fah- rens- typ	A	89	84	81	87	79
	B	88	77	87	92	81
	C	97	92	87	89	80
	D	94	79	85	84	88

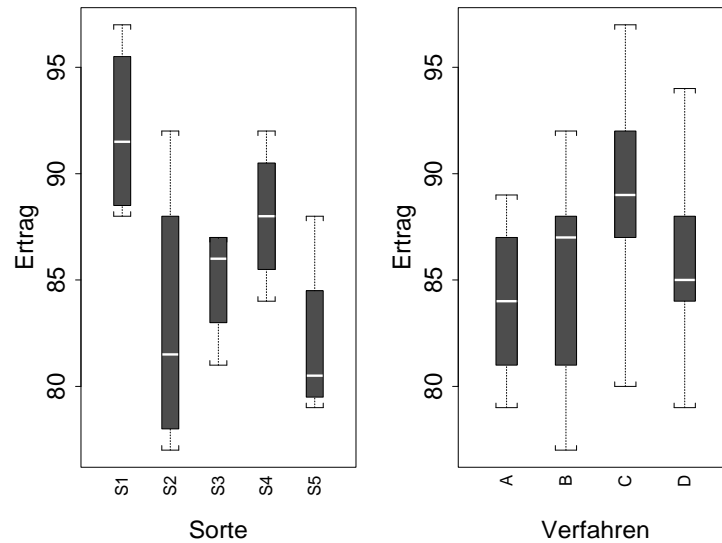
Zunächst sind (wie in Abschnitt 2.2.1 auf Seite 95) das Design des Experiments und die Daten so in einem Data Frame zusammenzustellen, dass die gewünschten Levelkombinationen der beiden Faktoren „Sorte“ und „Verfahren“ in ihrer aufgetretenen Häufigkeit sowie die beobachteten Erträge repräsentiert sind:

```
> Faktornamen <- list( Sorte= paste( "S", 1:5, sep= ""), Verfahren=
+ LETTERS[ 1:4])
> Pen.design <- fac.design( levels= c( 5,4), factor.names= Faktornamen)
> Ertrag <- c( 89, 84, 81, 87, 79, 88, 77, . . . . , 85, 84, 88)
> Pen.df <- data.frame( Pen.design, Ertrag);      Pen.df
  Sorte Verfahren Ertrag
1     S1         A     89
2     S2         A     84
3     S3         A     81
4     S4         A     87
5     S5         A     79
6     S1         B     88
. . . .
20    S5         D     88
```

Als zweites sollte sich eine explorative Analyse der Daten anschließen. Die Erklärungen auf Seite 96 treffen auch hier zu (nur eben mit $n = 1$ und ohne dritten Index):

Einfaches Blockexperiment: Explorative Datenanalyse	
<pre>> plot.design(Pen.df) > attach(Pen.df) > interaction.plot(Verfahren, + Sorte, Ertrag) > detach("Pen.df") > plot.factor(Pen.df, rotate= T)</pre>	<p>Liefert den folgenden Design-, Interaktions- und darunter die Faktorplot(s) (wie schon für das zweifaktorielle Modell auf S. 96 beschrieben). (Das Argument <code>rotate= T</code> von <code>plot.factor()</code> ist nützlich bei langen Faktorlevel-Bezeichnungen; siehe nächste Seite.)</p>





Qualitatives Fazit:

- Der Vergleich der beiden Faktorplots (diese Seite oben) zeigt, dass die Streuung der Response innerhalb der Verfahren-Levels im rechten Plot (wo über die Sorte-Blöcke gepoolt ist) größer ist als die Response-Streuung innerhalb der Sorte-Blöcke (bis auf Block „S2“). Dies ist ein Indiz dafür, dass die Störvariable Sorte eine größere Variabilität induziert als die interessierende Variable Verfahren.
- Denselben Eindruck vermittelt der Designplot (vorherige Seite links unten), in dem die mittleren Erträge zwischen den Sorten stärker variieren als zwischen den Verfahren.
- Keine Interaktion – hier zwischen dem Blockbildungsfaktor Sorte und dem Behandlungsfaktor Verfahren – läge vor, wenn die Verfahren-Profile (Polygonzüge) im Interaktionsplot idealisierterweise parallel verliefen, d. h., der „profildefinierende Faktor“ (hier: Sorte) einen rein additiven Effekt hätte. Dies scheint hier nicht der Fall zu sein!

Allerdings ist im einfachen Blockexperiment die Interpretation dieser Plots mit Vorsicht zu genießen, da hinter den Knoten eines jeden Polygonzuges jeweils nur *eine einzige* Beobachtung steckt. Ein Interaktionseffekt kann somit durch die Streuung der Daten leicht suggeriert, aber auch genauso gut maskiert werden.

Um obiges Modell (28) zu fitten, wird wieder die Funktion `aov()` verwendet, wobei es bei nominalen Faktoren intern (wie zuvor) zu einer Reparametrisierung mittels der Helmert-Kontraste kommt. Zur expliziten Ermittlung von $\hat{\alpha}_j$ und $\hat{\beta}_l$ als Schätzwerte für die Effekte dient auch hier `model.tables()`.

Einfaches Blockexperiment: Die ANOVA-Tabelle					
<pre>> Pen.aov <- aov(Ertrag ~ Sorte + Verfahren, data= Pen.df)</pre>					
<pre>> summary(Pen.aov)</pre>					
	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Sorte	4	264	66.00000	3.504425	0.0407462
Verfahren	3	70	23.33333	1.238938	0.3386581
Residuals	12	226	18.83333		
Es wird das additive zweifaktorielle Modell von Ertrag an Sorte und Verfahren (aus Pen.df) gefittet und in Pen.aov abgelegt. <code>summary(Pen.aov)</code> liefert die ANOVA-Tabelle: In der Zeile für den Blockbildungsfaktor Sorte stehen die Freiheitsgrade (Df)					

$J - 1$, die Summe SS_{Block} der Abweichungsquadrate der Blockmittelwerte vom Gesamtmittel (**Sum of Sq**), die mittlere Blockquadratsumme $SS_{Block}/(J - 1)$ (**Mean Sq**) sowie der F -Test (**F Value**) samt p -Wert (**Pr(F)**) auf Einfluss des Blockbildungsfaktors. In der Zeile für den Behandlungsfaktor **Verfahren** stehen die analogen Größen, insbesondere die Summe $SS_{Beh.}$ der Abweichungsquadrate der Behandlungsmittelwerte vom Gesamtmittel und der F -Test samt p -Wert auf Einfluss des Behandlungsfaktors. In der Zeile **Residuals** stehen die Residuenquadratsumme RSS mit ihren $(J - 1)(L - 1)$ Freiheitsgraden und die mittlere Residuenquadratsumme $RSS/(J - 1)(L - 1)$.

Einfaches Blockexperiment: Die Parameterschätzer

<pre>> model.tables(Pen.aov) Tables of effects Sorte S1 S2 S3 S4 S5 6 -3 -1 2 -4 Verfahren A B C D -2 -1 3 0</pre>	<p>Liefert (voreinstellungsgemäß) die KQS für die Effekte in (28):</p> <p>$\hat{\alpha}_1, \dots, \hat{\alpha}_J$ (Identifizierungsbedingung ist, evtl. bis auf Rundungsfehler, erfüllt.)</p> <p>$\hat{\beta}_1, \dots, \hat{\beta}_L$</p>
<pre>> model.tables(Pen.aov, type= + "means") Tables of means Grand mean 86 Sorte S1 S2 S3 S4 S5 92 83 85 88 82 Verfahren A B C D 84 85 89 86</pre>	<p>Wegen <code>type= "means"</code> werden die KQS für das cell means-Modell $Y_{jl} = \mu_{jl} + \varepsilon_{jl}$ ausgegeben:</p> <p>$\bar{Y}_{..}$ (= $\hat{\mu}_0$) für das Gesamtmittel $\bar{\mu}_{..}$ (= μ_0; "Grand mean"),</p> <p>$\bar{Y}_{1.}, \dots, \bar{Y}_{J.}$ für die marginalen (!) Blocklevel-Mittelwerte $\bar{\mu}_{1.}, \dots, \bar{\mu}_{J.}$ (wobei $\bar{\mu}_{j.} = \mu_0 + \alpha_j$),</p> <p>$\bar{Y}_{.1}, \dots, \bar{Y}_{.L}$ für die marginalen (!) Faktorlevel-Mittelwerte $\bar{\mu}_{.1}, \dots, \bar{\mu}_{.L}$ (wobei $\bar{\mu}_{.l} = \mu_0 + \beta_l$).</p>
<p>Bemerkung: Die Warning messages, wie sie durch <code>model.tables()</code> auf S. 88 ausgegeben wurden, sind hier nicht aufgetreten, da wir, ohne es auf der vorherigen Seite zu dokumentieren, das <code>aov</code>-Argument <code>projections= T</code> verwendet haben.</p>	

Einfaches Blockexperiment: Diagnoseplots

<pre>> fitted(Pen.aov) > resid(Pen.aov) > hist(resid(Pen.aov)) > qqnorm(resid(Pen.aov)) > plot(fitted(Pen.aov), + resid(Pen.aov))</pre>	<p>An die gefitteten Werte (= geschätzte mittlere Responses für jede Levelkombination) und die Residuen kommt man mit <code>fitted()</code> bzw. <code>resid()</code>, was die Beurteilung der Normalverteilungsannahme der Fehler durch ein Histogramm und einen Q-Q-Plot der Residuen sowie die Prüfung der Varianzhomogenität durch einen Plot der Residuen gegen die gefitteten Werte ermöglicht. (Alle nicht gezeigt.)</p>
---	---

2.4 Zwei nicht-parametrische Mehrstichprobentest für Lokationsprobleme

Für die Inferenzstatistik der ein- oder mehrfaktoriellen Modelle ist die Normalverteilung der Daten eine zentrale Eigenschaft. Wenn die Normalverteilungsannahme **nicht** gerechtfertigt erscheint, stehen analog zu den Zweistichproben-Szenarien verschiedene nicht-parametrische (Rang-)Verfahren zur Verfügung:

Für $L > 2$ unabhängige Stichproben ist der Kruskal-Wallis-Test (als Verallgemeinerung von Wilcoxon's Rangsummentest für den Fall $L = 2$) das nicht-parametrische Pendant zur einfaktoriellen Varianzanalyse und in der Funktion `kruskal.test()` implementiert.

Das einfache Blockexperiment, welches ein geeignet interpretiertes zweifaktorielles Modell ist und den Fall von $L > 2$ verbundenen Stichproben widerspiegelt (indem die Stichproben durch die Blockbildung verbunden werden), untersucht man nicht-parametrisch mit Hilfe des Friedman-Tests, der in `friedman.test()` realisiert ist.

2.4.1 Der Kruskal-Wallis-Test für unabhängige Stichproben

Es liegen $L > 2$ unabhängige Stichproben (SPn) unabhängiger Zufallsvariablen (ZVn) X_{li} ($i = 1, \dots, n_l, l = 1, \dots, L$) vor und es soll getestet werden, ob ihre L Verteilungsfunktionen (VF) F_1, \dots, F_L gleich sind. Unter der Hypothese gleicher VFn stammen alle $N := \sum_{l=1}^L n_l$ ZVn X_{li} aus derselben Verteilung und der N -dimensionale Vektor $(R_{11}, \dots, R_{1n_1}, \dots, R_{l1}, \dots, R_{Ln_L})$ der Ränge R_{li} der *gepoolten* Stichproben ist uniform auf der Menge Σ_N der N -Permutationen verteilt. Daher sollte jede der L SPn-Rangsummen $R_l := \sum_{i=1}^{n_l} R_{li}$ einen zu ihrem SPn-Umfang n_l in etwa proportionalen Anteil an der konstanten Gesamtsumme $\sum_{l=1}^L R_l = N(N+1)/2$ haben, also $R_l \approx n_l/N * N(N+1)/2 = n_l(N+1)/2$. Es macht daher Sinn, die (quadratischen) „Abstände“ der SPn-Rangsummen von ihren „erwarteten“ Werten zu messen. (Zur etwas formaleren Vorgehensweise siehe unten.)

SP	Zufallsvariablen	VF		Ränge im SPn-Pool	Zeilen- Σ
1	X_{11}, \dots, X_{1n_1}	$\overset{\text{u.i.v.}}{\sim} F_1$		R_{11}, \dots, R_{1n_1}	$R_1.$
\vdots		\vdots			\vdots
l	$X_{l1}, \dots, \dots, X_{ln_l}$	$\overset{\text{u.i.v.}}{\sim} F_l$	\Rightarrow	$R_{l1}, \dots, \dots, R_{ln_l}$	$R_l.$
\vdots		\vdots			\vdots
L	$X_{1L}, \dots, \dots, X_{Ln_L}$	$\overset{\text{u.i.v.}}{\sim} F_L$		$R_{1L}, \dots, \dots, R_{Ln_L}$	$R_L.$
Gesamtsumme:					$N(N+1)/2$

Annahmen: Die X_{li} sind für $i = 1, \dots, n_l$ und $l = 1, \dots, L$ unabhängig und für $l = 1, \dots, L$ sind X_{l1}, \dots, X_{ln_l} u.i.v. $\sim F_l \equiv F(\cdot - \theta_l)$ mit stetigem F und unbekanntem θ_l .

Zu testen zum Signifikanzniveau α :

$$H_0 : \theta_1 = \dots = \theta_L \quad \text{gegen} \quad H_1 : \theta_j \neq \theta_l \text{ für mindestens ein Paar } j \neq l.$$

Teststatistik:

$$H_N \equiv H_{n_1, \dots, n_L} := \frac{12}{N(N+1)} \sum_{l=1}^L \frac{1}{n_l} \left(R_l - \frac{n_l(N+1)}{2} \right)^2,$$

wobei $N := \sum_{l=1}^L n_l$ und $R_l := \sum_{i=1}^{n_l} R_{li}$ ist sowie R_{li} der *Rang* von X_{li} unter den gepoolten X_{11}, \dots, X_{Ln_L} ist. Die Verteilung der Kruskal-Wallis-Statistik H_N unter H_0 ist (für kleine Werte von n_l und L) bekannt. Kombinatorische Überlegungen zeigen, dass für jedes l gilt:

$$\mathbb{E}[R_l] = \frac{n_l(N+1)}{2} \quad \text{und} \quad \text{Var}(R_l) = \frac{n_l(N+1)(N-n_l)}{12} \quad \text{unter } H_0.$$

Des Weiteren ist jedes R_l asymptotisch normalverteilt:

$$Z_{n_l} := \frac{R_l - \mathbb{E}[R_l]}{\sqrt{\text{Var}(R_l)}} \xrightarrow{n_l \rightarrow \infty} \mathcal{N}(0, 1) \quad \text{unter } H_0, \text{ falls } \frac{n_l}{N} \rightarrow \lambda_l > 0,$$

woraus folgt: $Z_{n_l}^2 \xrightarrow{n_l \rightarrow \infty} \chi_1^2$. Allerdings sind die R_l und damit die $Z_{n_l}^2$ *nicht* unabhängig, denn $\sum_{l=1}^L R_l = N(N+1)/2$, weswegen für die (geeignet gewichtete) Summe der $Z_{n_l}^2$ nicht L sondern $L-1$ Freiheitsgrade für die asymptotische χ^2 -Verteilung resultieren:

$$H_N = \sum_{l=1}^L \frac{N-n_l}{N} Z_{n_l}^2 \xrightarrow{N \rightarrow \infty} \chi_{L-1}^2 \quad \text{unter } H_0, \text{ falls } \frac{n_l}{N} \xrightarrow{N \rightarrow \infty} \lambda_l > 0 \text{ für } l = 1, \dots, L.$$

`kruskal.test()` verwendet stets diese χ^2 -Approximation. Treten Bindungen zwischen den X_{li} *verschiedener* SPn (also verschiedener l -Werte) auf, so wird die Methode der Durchschnittsränge verwendet, was den Erwartungswert von H_N nicht, wohl aber ihre Varianz beeinflusst. (Bindungen *innerhalb* von SPn, also für dasselbe l , spielen keine Rolle, da diese die Rangsummen R_l nicht beeinflussen.) In diesem Fall wird eine durch einen Faktor modifizierte Statistik H_N^{mod} an Stelle von H_N verwendet. H_N^{mod} hat dieselbe χ^2 -Asymptotik wie H_N . (Wir gehen auf diesen Sachverhalt hier nicht ein, sondern verweisen z. B. auf Büning und Trenkler (1994).)

Entscheidungsregel für konkrete Daten x_{11}, \dots, x_{Ln_L} auf Basis des p -Wertes:

$$\text{Verwirf } H_0 \iff p\text{-Wert} \leq \alpha,$$

wobei für die Berechnung des p -Wertes die χ^2 -Approximation zur Anwendung kommt: Ist h_N der realisierte Wert von H_N (bzw. von H_N^{mod} , wenn Bindungen vorliegen), dann ist

$$p\text{-Wert} = 1 - F_{\chi_{L-1}^2}(h_N).$$

Beispiel anhand des Datensatzes der Koagulationszeiten auf Seite 85 in Abschnitt 2.1: Die Funktion `kruskal.test()` erwartet als erstes Argument den Vektor der X -Werte und als zweites Argument einen Faktorvektor, der die SPn-Zugehörigkeit charakterisiert:

```
> kruskal.test( Koag.df$Zeit, Koag.df$Diaet)
```

```
Kruskal-Wallis rank sum test
```

```
data: Koag.df$Zeit and Koag.df$Diaet
Kruskal-Wallis chi-square = 17.0154, df = 3, p-value = 7e-04
alternative hypothesis: two.sided
```

Die Ausgabe ist vollständig selbsterklärend.

2.4.2 Der Friedman-Test für durch Blockbildung verbundene Stichproben

Zur Beschreibung des Szenarios erinnern wir an die Ausführungen zu Beginn von Abschnitt 2.3: Wir wollen $L \geq 3$ verschiedene Behandlungen untersuchen, deren Effekte durch eine Störgröße mit $J \geq 2$ Ausprägungen maskiert werden könnten, und haben insgesamt $J \cdot L$ UEn so zur Verfügung, dass in jeder Behandlungsgruppe zu jeder Störgrößen- ausprägung genau eine UE existiert. Die ZV X_{jl} ist dann die Beobachtung in Behandlung l im (bezüglich der Störgröße homogenen) Block j und habe die VF F_{jl} . Es soll getestet werden, ob $F_{j1} = \dots = F_{jL}$ für jedes $j = 1, \dots, J$ gilt.

Unter der Hypothese $F_{j1} = \dots = F_{jL}(=: F_j)$ für jedes $j = 1, \dots, J$ stammen die L unabhängigen ZVn X_{j1}, \dots, X_{jL} aus derselben Verteilung F_j und der L -dimensionale Vektor (R_{j1}, \dots, R_{jL}) der Ränge R_{jl} im Block j ist uniform auf der Menge Σ_L der L -Permutationen verteilt. Jede Blockrangsumme R_j ist außerdem stets gleich $L(L+1)/2$ und die Summe $R_{..}$ über alle Ränge der J SPn daher gleich $JL(L+1)/2$. Daher sollten unter der genannten Hypothese alle L Behandlungsrangsummen $R_{.l} := \sum_{j=1}^J R_{jl}$ etwa denselben Wert haben und dieser in etwa gleich $R_{..}/L = J(L+1)/2$ sein. Es macht daher Sinn, die (quadratischen) „Abstände“ der Behandlungsrangsummen von ihren „erwarteten“ Werten zu messen. (Die formalere Vorgehensweise folgt unten.)

Block	Zufallsvariablen				\Rightarrow	Ränge pro Block				Block- Rang- Σ
	Behandlung					Behandlung				
	1	2	...	L		1	2	...	L	
1	X_{11}	X_{12}	...	X_{1L}		R_{11}	R_{12}	...	R_{1L}	$L(L+1)/2$
\vdots	\vdots	\vdots	\vdots	\vdots		\vdots	\vdots	\vdots	\vdots	\vdots
j	X_{j1}	X_{j2}	...	X_{jL}		R_{j1}	R_{j2}	...	R_{jL}	$L(L+1)/2$
\vdots	\vdots	\vdots	\vdots	\vdots		\vdots	\vdots	\vdots	\vdots	\vdots
J	X_{J1}	X_{J2}	...	X_{JL}		R_{J1}	R_{J2}	...	R_{JL}	$L(L+1)/2$
						$R_{.1}$	$R_{.2}$...	$R_{.L}$	$JL(L+1)/2$
						Behandlungsrang- Σ				

Annahmen: Die X_{jl} sind für $j = 1, \dots, J$ und $l = 1, \dots, L$ unabhängig und $X_{jl} \sim F_{jl} \equiv F(\cdot - \alpha_j - \theta_l)$ mit stetigem F sowie unbekanntem α_j und θ_l .

Zu testen zum Signifikanzniveau α :

$$H_0 : \theta_1 = \dots = \theta_L \quad \text{gegen} \quad H_1 : \theta_l \neq \theta_k \text{ für mindestens ein Paar } l \neq k.$$

Teststatistik:

$$F_J := \sum_{l=1}^L \left(R_{.l} - \frac{J(L+1)}{2} \right)^2,$$

wobei $R_{.l} := \sum_{j=1}^J R_{jl}$ ist sowie R_{jl} der Rang von X_{jl} in Block j , also unter X_{j1}, \dots, X_{jL} ist. Die Verteilung der Friedman-Statistik F_J unter H_0 ist (für kleine Werte von J und L) bekannt. Kombinatorische Überlegungen zeigen, dass für jedes j und l gilt:

$$\mathbb{E}[R_{jl}] = \frac{L+1}{2}, \quad \text{Var}(R_{jl}) = \frac{L^2-1}{12} \quad \text{und} \quad \text{Cov}(R_{jl}, R_{jk}) = -\frac{L+1}{12} \text{ für } l \neq k$$

sowie

$$\mathbb{E}[F_J] = L-1 \quad \text{und} \quad \text{Var}(F_J) = \frac{2(L-1)(J-1)}{J} \quad \text{unter } H_0.$$

Des Weiteren ist unter H_0 jedes $R_{.l}$ asymptotisch normalverteilt:

$$Z_l := \frac{R_{.l} - \mathbb{E}[R_{.l}]}{\sqrt{\text{Var}(R_{.l})}} \xrightarrow{J \rightarrow \infty} \mathcal{N}(0, 1) \quad \text{unter } H_0,$$

wobei

$$\mathbb{E}[R_{.l}] = \frac{J(L+1)}{2} \quad \text{und} \quad \text{Var}(R_{.l}) = \frac{J(L^2-1)}{12}.$$

Daraus folgt: $Z_l^2 \xrightarrow{J \rightarrow \infty} \chi_1^2$. Allerdings sind die $R_{.l}$ und damit die Z_l^2 *nicht* unabhängig, denn $\sum_{l=1}^L R_{.l} = JL(L+1)/2$, weswegen für die (geeignet gewichtete) Summe der Z_l^2 nicht L sondern $L-1$ Freiheitsgrade für die asymptotische χ^2 -Verteilung resultieren:

$$F_J = \sum_{l=1}^L \frac{L-1}{L} Z_l^2 \xrightarrow{J \rightarrow \infty} \chi_{L-1}^2 \quad \text{unter } H_0.$$

`friedman.test()` verwendet stets diese χ^2 -Approximation. Treten Bindungen zwischen den X_{jl} *innerhalb* eines Blocks (also für verschiedene l -Werte bei gleichem j) auf, so wird die Methode der Durchschnittsränge verwendet, was den Erwartungswert von F_J nicht, wohl aber ihre Varianz beeinflusst. In diesem Fall wird eine durch einen Faktor modifizierte Statistik F_J^{mod} an Stelle von F_J verwendet. F_J^{mod} hat dieselbe χ^2 -Asymptotik wie F_J . (Wir gehen auf diesen Sachverhalt hier nicht ein, sondern verweisen z. B. auf Büning und Trenkler (1994).)

Entscheidungsregel für konkrete Daten x_{11}, \dots, x_{JL} auf Basis des p -Wertes:

$$\text{Verwirf } H_0 \iff p\text{-Wert} \leq \alpha,$$

wobei für die Berechnung des p -Wertes die χ^2 -Approximation zur Anwendung kommt: Ist f_J der realisierte Wert von F_J (bzw. von F_J^{mod} , wenn Bindungen vorliegen), dann ist

$$p\text{-Wert} = 1 - F_{\chi_{L-1}^2}(f_J).$$

Beispiel anhand des Penicillin-Datensatzes (siehe Seite 102 in Abschnitt 2.3): Auch die Funktion `friedman.test()` erwartet als erstes Argument den Vektor der X -Werte und als zweites Argument einen Faktorvektor, der die Behandlung charakterisiert, aber als drittes Argument noch einen Faktor, der die Blockzugehörigkeit angibt:

```
> friedman.test( Pen.df$Ertrag, Pen.df$Verfahren, Pen.df$Sorte)
```

```
Friedman rank sum test
```

```
data: Pen.df$Ertrag and Pen.df$Verfahren and Pen.df$Sorte
Friedman chi-square = 3.4898, df = 3, p-value = 0.3221
alternative hypothesis: two.sided
```

Auch diese Ausgabe ist vollständig selbsterklärend.

2.5 Multiple Mittelwertvergleiche

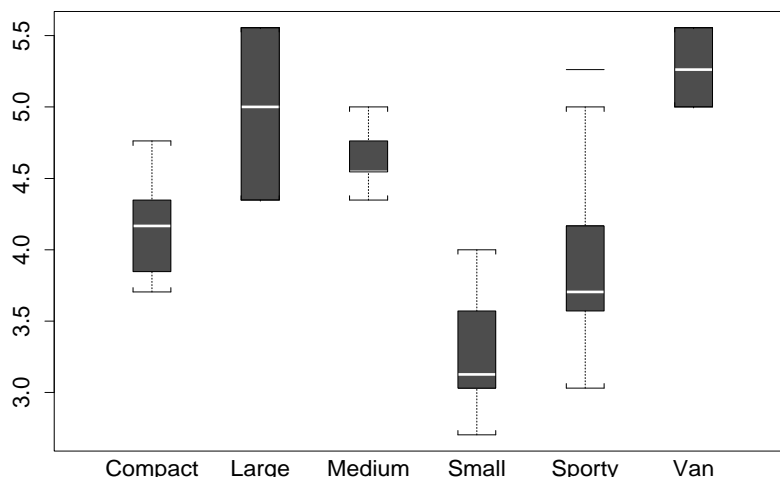
Im Anschluss an eine ANOVA, die einen signifikanten Einfluss der Levels eines Faktors zu Tage gefördert hat, ist man häufig daran interessiert herauszufinden, *welche* Levels sich signifikant unterscheiden. Dies läuft auf *multiple* Mittelwertvergleiche (Engl.: “multiple comparisons”, kurz: MC) hinaus, wofür zahlreiche Verfahren entwickelt wurden, alldieweil es sehr unterschiedliche Möglichkeiten gibt, solche Vergleiche durchzuführen.

Wir beschränken uns hier auf die Vorstellung zweier Standardverfahren für die *einfaktorielle* ANOVA, die in S-PLUS 6 in der Funktion `multicomp()` realisiert sind. (Diese Funktion ist allerdings im Stande, erheblich mehr zu leisten, als das, worauf wir hier eingehen können. Das Manual, an dessen Darstellung wir uns hier stark anlehnen, und die Online-Hilfe von S-PLUS liefern weiter gehende Informationen.) `multicomp()` liefert nicht einfach nur die Signifikanzergebnisse, die in einer „Familie“ von multiplen Vergleichen im Einzelnen erzielt wurden, sondern überdies die *simultanen* Konfidenzintervalle für alle betrachteten Mittelwertdifferenzen (bzw. allgemeiner für die den Vergleichen zugrundeliegenden linearen Kontrasten). Diese Konfidenzintervalle liefern erheblich mehr (interpretierbare) Informationen, als die alleinige Dokumentation von „Signifikant“- oder „Nicht signifikant“-Aussagen für einzelne Vergleiche.

Aus Platz- und Zeitgründen verzichten wir auf jegliche Darstellung des theoretischen Hintergrunds, sondern verweisen auf die Literatur. Zum Beispiel bietet Hsu (1996), eine gute Einführung in und einen umfangreichen Überblick über das Thema. Wir belassen es hier bei Anwendungsbeispielen.

Der eingebaute Data Frame `fuel.frame` enthält in seiner Komponente `Fuel` den Spritverbrauch zahlreicher Automobile in Gallonen pro 100 Meilen. Die Faktorkomponente `Type` kategorisiert die Autos in verschiedene Fahrzeugtypen. Nach `Type` gruppierte Boxplots der Spritverbräuche deuten auf Unterschiede im mittleren Spritverbrauch zwischen einigen der Faktorlevels hin, aber nicht zwischen allen:

```
> boxplot( split( fuel.frame$Fuel, fuel.frame$Type))
```



Eine einfaktorielle ANOVA dient der Klärung:

```
> fuel.aov <- aov( Fuel ~ Type, data= fuel.frame)
> summary( fuel.aov)
```

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Type	5	24.23960	4.847921	27.22058	1.220135e-13
Residuals	54	9.61727	0.178098		

Die ANOVA bestätigt einen hochsignifikanten Einfluss von `Type` auf den mittleren Spritverbrauch. Doch welche *paarweisen* Unterschiede zwischen den Fahrzeugtypen sind dafür verantwortlich bzw. sind signifikant? Und wie groß sind diese Unterschiede? Solche Fragen erfordern die Analyse paarweiser Differenzen mit Hilfe der Funktion `multicomp()`. Die Funktion, mit den für *unsere Zwecke* wesentlichen Argumenten, hat die folgende Syntax:

```
multicomp( x, focus= NULL, comparisons= "mca", alpha= 0.05, bounds="both",
          error.type= "fwe", control= NULL, plot= F, labels= NULL)
```

Zur Bedeutung der Argumente:

- `x`: Ein `aov`-Objekt (oder allgemeiner sogar ein `lm`-Objekt).
- `focus`: Zeichenkette, die den Faktor spezifiziert, für dessen Levels die multiplen Vergleiche durchgeführt werden sollen. Falls, wie in der Voreinstellung, `focus= NULL` ist, wird der erste Faktor im Modell in `x` als der „`focus`-Faktor“ verwendet. Damit ist dieses Argument bei einem einfaktoriellem `aov`-Objekt eigentlich unnötig.
- `comparisons`: Zeichenkette, die angibt, welche multiplen (Standard-)Vergleiche durchzuführen sind (und demnach, welche paarweisen Differenzen zu berechnen sind):
 "mca" für *alle* paarweisen Vergleiche (Voreinstellung);
 "mcc" für die paarweisen Vergleiche zwischen einem Kontrolllevel (spezifiziert durch das Argument `control`; siehe unten) und den übrigen Levels (des `focus`-Faktors).
- `alpha`: Das simultane Niveau der Konfidenzgrenzen der paarweisen Differenzen ist $1 - \alpha$, falls, wie voreingestellt, `error.type= "fwe"` ist (siehe unten). Falls `error.type= "cwe"` ist, beträgt das Niveau der Konfidenzgrenzen jeder *einzelnen* Differenz $1 - \alpha$. Voreinstellung: `alpha= 0.05`.
- `bounds`: Zeichenkette, die den Typ der zu berechnenden Konfidenzgrenzen angibt:
 "both" (Voreinstellung) führt zu abgeschlossenen Intervallen.
 "lower" (bzw. "upper") liefert untere (obere) Schranken für die betrachteten Differenzen; diese sind „schärfer“, also größer (kleiner) als die unteren (oberen) Grenzen der Konfidenzintervalle.
- `error.type`: Zeichenkette, die den Fehlertyp der Konfidenzgrenzen spezifiziert. Die Voreinstellung "fwe" (= "family-wise error") garantiert *simultane* Konfidenzgrenzen, d. h., die Wahrscheinlichkeit, dass *alle* Schranken *gleichzeitig* (sozusagen als Familie) eingehalten werden, ist mindestens $1 - \alpha$. Der Wert "cwe" (= "comparison-wise error") liefert *vergleichsbezogene* Konfidenzgrenzen, d. h., die Wahrscheinlichkeit, dass *eine vorbestimmte* Schranke eingehalten wird, ist mindestens $1 - \alpha$.
- `control`: Nur von Bedeutung, wenn `comparisons= "mcc"` ist. Dann ist es die Nummer des Faktorlevels, der die Kontrollgruppe dient. (Beachte die alphabetische Reihenfolge der Levels bei (ungeordneten) Faktoren!)
- `plot`: Logischer Wert, der angibt, ob auch gleich ein Plot der berechneten Konfidenzgrenzen angefertigt werden soll. Voreinstellung: `FALSE`. (Alternativ kann das Ergebnisobjekt von `multicomp()` auch an die Funktion `plot()` übergeben werden.)

- **labels**: Zeichenkettenvektor, der die Labels der Tabelle bzw. des Plots der Konfidenzgrenzen enthält. Per Voreinstellung werden für die Standardvergleiche (möglichst) sinnvolle Bezeichnungen aus den Faktorlevel-Bezeichnungen abgeleitet.

2.5.1 Alle paarweisen Vergleiche (= “All-pairwise multiple comparisons”, MCA)

Ist man an allen paarweisen Vergleichen (Engl.: “all-pairwise multiple comparisons”, kurz: MCA) interessiert, so ist `multicomp()` in ihrer „Voreinstellungsversion“ zu nutzen. Sie liefert alle paarweisen Mittelwertvergleiche für die Levels des durch `focus` spezifizierten Faktors des ihr übergebenen `aov`-Objektes (hier `fuel.aov`; siehe unten). Das Argument `focus` wäre hier nicht nötig, da für `fuel.aov` als *ein*faktoriellem `aov`-Objekt klar ist, welcher der „`focus`-Faktor“ nur sein kann. Das Ergebnis ist ein `multicomp`-Objekt, das hier in `fuel.mca` gespeichert wird.

Dessen Ausgabe dokumentiert, dass (gemäß Voreinstellung) simultane 95 %-Konfidenzgrenzen für die “specified linear combinations”, also hier für alle paarweisen Differenzen zwischen den `Fuel`-Mittelwerten der `Type`-Levels berechnet wurden, und zwar mittels Tukeys Methode (bzw., da die Levelgruppengrößen verschieden sind, gemäß der Tukey-Kramer-Methode; siehe Hsu (1996)). Der `critical point` ist das verwendete, methodenabhängige Quantil in den durch `Estimate ± (critical point) × Std.Error` berechneten Konfidenzintervallen für die Differenzen. Die (Vierfach-)Sternchen markieren die Intervalle, die nicht die Null enthalten. Damit gelten die entsprechenden Mittelwerte als signifikant voneinander verschieden, und zwar gemäß Tukeys HSD-Methode (= Methode der “honestly significant differences”).

```
> fuel.mca <- multicomp( fuel.aov, focus= "Type");      fuel.mca
```

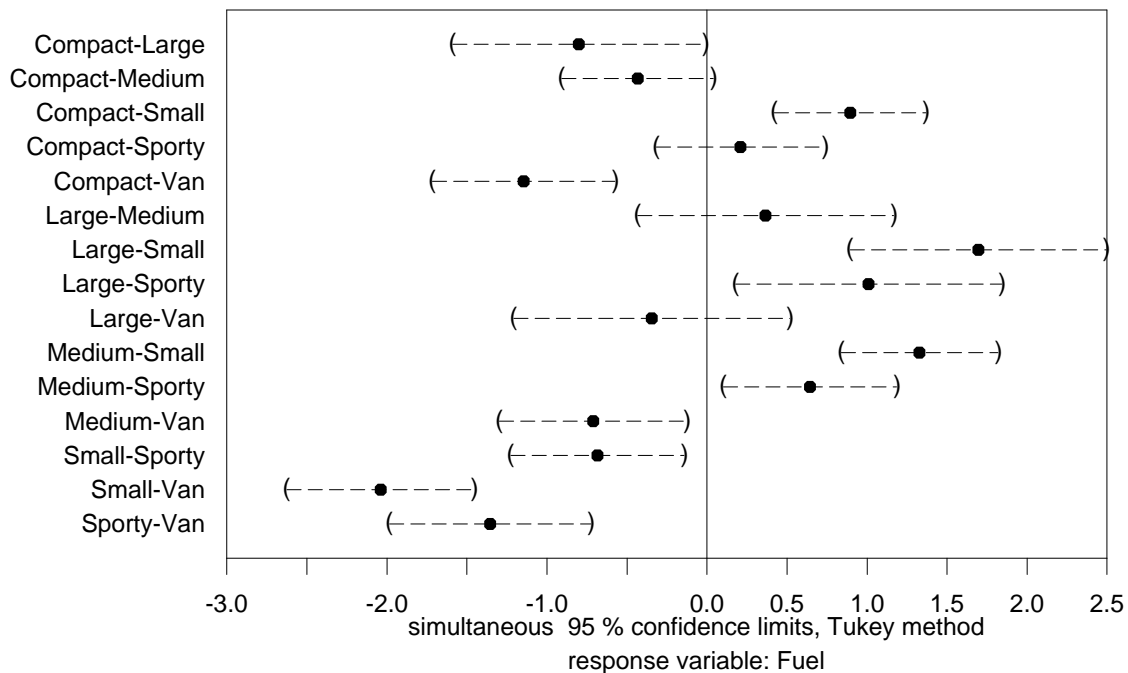
```
95 % simultaneous confidence intervals for specified
linear combinations, by the Tukey method
critical point: 2.9545
response variable: Fuel
intervals excluding 0 are flagged by '****'
```

	Estimate	Std.Error	Lower Bound	Upper Bound	
Compact-Large	-0.800	0.267	-1.590	-0.0116	****
Compact-Medium	-0.434	0.160	-0.906	0.0387	
Compact-Small	0.894	0.160	0.422	1.3700	****
Compact-Sporty	0.210	0.178	-0.316	0.7360	
Compact-Van	-1.150	0.193	-1.720	-0.5750	****
Large-Medium	0.366	0.270	-0.432	1.1600	
Large-Small	1.690	0.270	0.896	2.4900	****
Large-Sporty	1.010	0.281	0.179	1.8400	****
Large-Van	-0.345	0.291	-1.210	0.5150	
Medium-Small	1.330	0.166	0.839	1.8200	****
Medium-Sporty	0.644	0.183	0.103	1.1800	****
Medium-Van	-0.712	0.198	-1.300	-0.1270	****
Small-Sporty	-0.684	0.183	-1.220	-0.1440	****
Small-Van	-2.040	0.198	-2.620	-1.4600	****
Sporty-Van	-1.360	0.213	-1.980	-0.7270	****

Schlussfolgerung: Auf einem Konfidenzniveau von 95 % gilt die folgende „Global“-Aussage über die (hier 15) betrachteten Mittelwertpaare: „Es besteht kein signifikanter Unterschied zwischen `Compact` und `Medium`, `Compact` und `Sporty`, `Large` und `Medium` sowie `Large` und `Van`, und es bestehen signifikante Unterschiede in allen anderen Paaren.“

Eine grafische Darstellung der simultanen Konfidenzintervalle erhält man mit Hilfe von `plot()`, angewendet auf das `multicomp`-Objekt:

```
> plot( fuel.mca)
```



In obigem Plot markiert die senkrechte Linie die Null, so dass die Intervalle, welche diese Linie nicht schneiden, gerade diejenigen Konfidenzintervalle in der Tabelle sind, die die Sternchen haben.

2.5.2 Multiple Vergleiche mit einer Kontrolle (= “Multiple comparisons with a control”, MCC)

Wenn Konfidenzgrenzen zu einem gegebenen Niveau $1 - \alpha$ simultan eingehalten werden sollen, muss die Fehlerwahrscheinlichkeit α auf diese Intervalle „verteilt“ werden. Je mehr Intervalle es sind, umso kleiner ist für jedes einzelne die zur Verfügung stehende Fehlerwahrscheinlichkeit, was zu umso größeren Intervallen führt. Damit α also nicht „verschwendet“ wird, ist es wichtig, nur die wirklich interessierenden Intervalle zu betrachten. Als Belohnung für diese Sparsamkeit erhält man engere Konfidenzintervalle.

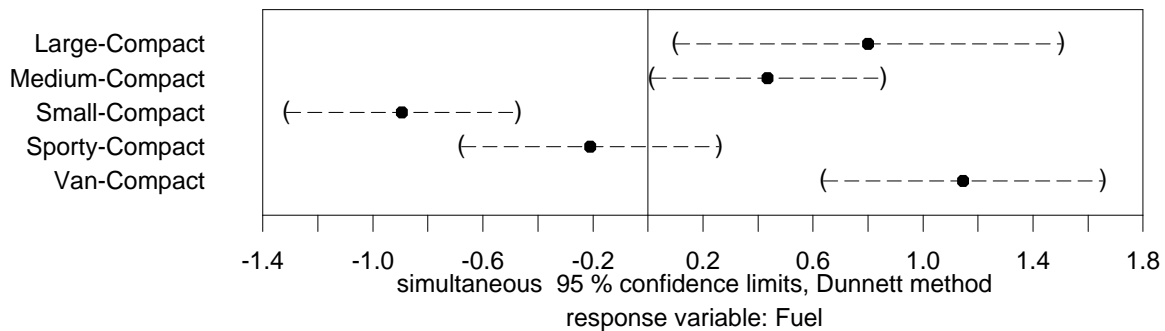
Ein typisches Anwendungsbeispiel hierfür ist der Vergleich verschiedener Gruppen (Treatments) mit einer Kontrolle, aber nicht untereinander. Dies erfordert für einen Faktor mit k Levels dann lediglich $k - 1$ paarweise Vergleiche (im Gegensatz zu $k(k - 1)/2$ für alle paarweisen Vergleiche). Man spricht dann von multiplen Vergleichen mit einer Kontrolle (Engl.: “multiple comparisons with a control”, kurz: MCC).

Wir bedienen uns wieder des Spritverbrauch-Datensatzes, tun aber nun so, als fungierte der Typ `Compact` als Kontrollgruppe. Der Funktion `multicomp()` teilen wir dies durch das Argument `comparisons= "mcc"` mit. Da die Faktorlevels von `Type` alphabetisch sortiert sind, ist `Compact` der erste Level, so dass wir durch `control= 1` diese Gruppe zur Kontrolle machen. Mit `plot= T` veranlassen wir ferner, dass auch sofort eine grafische Darstellung der Konfidenzgrenzen angefertigt wird (siehe nächste Seite):

```
> fuel.mcc <- multcomp( fuel.aov, focus= "Type", comparisons= "mcc",
+ control= 1, plot= T);      fuel.mcc
```

```
95 % simultaneous confidence intervals for specified
linear combinations, by the Dunnett method
critical point: 2.6255
response variable: Fuel
intervals excluding 0 are flagged by '****'
```

	Estimate	Std.Error	Lower Bound	Upper Bound	
Large-Compact	0.800	0.267	0.0994	1.500	****
Medium-Compact	0.434	0.160	0.0139	0.854	****
Small-Compact	-0.894	0.160	-1.3100	-0.474	****
Sporty-Compact	-0.210	0.178	-0.6770	0.257	
Van-Compact	1.150	0.193	0.6380	1.650	****



Die Ausgabe dokumentiert, dass simultane 95 %-Konfidenzintervalle für die “specified linear combinations”, also hier für alle paarweisen Differenzen zwischen dem **Fuel**-Mittelwert des **Type**-Levels **Compact** und den anderen Levels berechnet wurden, und zwar mittels der Methode von Dunnett (siehe Hsu (1996)). Der **critical point** ist wieder das für die Konfidenzintervalle der Differenzen verwendete Quantil. Er ist kleiner als derjenige für alle paarweisen Vergleiche und führt daher zu kürzeren Konfidenzintervallen, was man durch den Vergleich der Längen der ersten fünf Intervalle des vorigen Abschnitts mit den hiesigen bestätigt findet. Die (Vierfach-)Sternchen markieren die Konfidenzintervalle, die nicht die Null enthalten. Damit gelten die entsprechenden Mittelwerte als signifikant voneinander verschieden.

Schlussfolgerung: Auf einem Konfidenzniveau von 95 % gilt die folgende „Global“-Aussage über die (hier 5) betrachteten Vergleiche der Mittelwerte der Levels **Large**, **Medium**, **Small**, **Sporty** und **Van** mit der Kontrollgruppe **Compact**: „Es besteht kein signifikanter Unterschied zwischen **Sporty** und **Compact**, aber es bestehen signifikante Unterschiede zwischen allen anderen und **Compact**.“

2.5.3 Einseitige Schranken für multiple Vergleiche

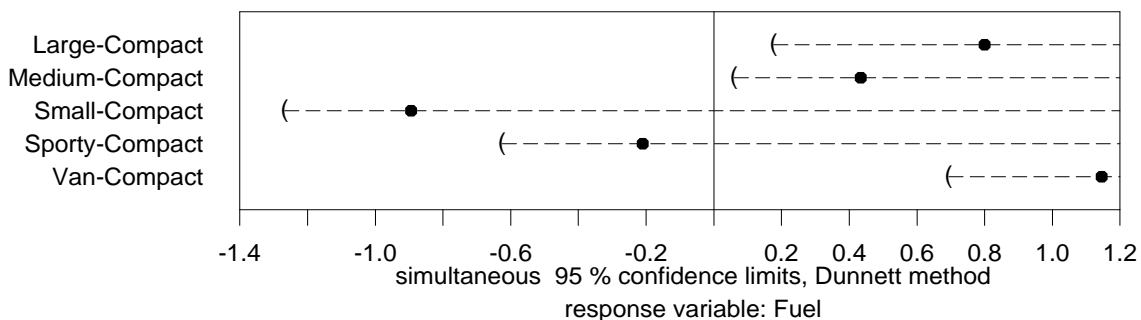
Häufig ist es nicht notwendig, beidseitige Konfidenzgrenzen, also Konfidenzintervalle anzugeben, sondern einseitige (obere oder untere) Schranken sind ausreichend. Zumal wenn die Überlegenheit (oder Unterlegenheit) verschiedener Treatments im Vergleich zu einer Kontrolle untersucht werden soll. Damit sind einseitige multiple Vergleiche und ihre simultanen Konfidenzschranken gefragt.

Anhand des Beispiels aus dem vorherigen Abschnitt wollen wir demonstrieren, wie die Funktion `multicomp()` das Gewünschte liefert: Der einzige Unterschied ist die Verwendung des Arguments `bounds= "lower"`, was zur Berechnung *unterer* Schranken für die Differenzen der Levelmittelwerte zum Compact-Mittelwert führt.

```
> fuel2.mcc <- multicomp( fuel.aov, focus= "Type", comparisons= "mcc",
+ bounds= "lower", control= 1, plot= T);      fuel2.mcc
```

```
95 % simultaneous confidence bounds for specified
linear combinations, by the Dunnett method
critical point: 2.3332
response variable: Fuel
bounds excluding 0 are flagged by '****'
```

	Estimate	Std.Error	Lower Bound	
Large-Compact	0.800	0.267	0.1770	****
Medium-Compact	0.434	0.160	0.0606	****
Small-Compact	-0.894	0.160	-1.2700	
Sporty-Compact	-0.210	0.178	-0.6250	
Van-Compact	1.150	0.193	0.6950	****



Die Ausgabe ist analog zu den vorherigen zu lesen.

Ist die Null außerhalb eines Konfidenzbereichs, so ist – hier – seine untere Schranke größer als Null und somit seine zugehörige Mittelwertdifferenz signifikant größer als Null. Dies bedeutet, dass der entsprechende Levelmittelwert signifikant größer als der Compact-Mittelwert ist.

Schlussfolgerung: Auf einem Konfidenzniveau von 95 % gilt die folgende „Global“-Aussage über die (hier 5) betrachteten Vergleiche der Mittelwerte der Levels **Large**, **Medium**, **Small**, **Sporty** und **Van** mit der Kontrollgruppe **Compact**: „Die Levels **Small** und **Sporty** haben keinen signifikant größeren mittleren Spritverbrauch als **Compact**, aber die Levels **Large**, **Medium** und **Van** haben signifikant höhere mittlere Verbräuche als **Compact**.“

Literatur

- [1] Agresti, A.: *Categorical Data Analysis*. John Wiley, New York, 1990. (2. Auflage, 2002: ~100 €)
- [2] Agresti, A.: *An Introduction to Categorical Data Analysis*. John Wiley, New York, 1996. (~98 €)
- [3] Box, G. E. P., Hunter, W. G., Hunter, J. S.: *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. John Wiley, New York, 1978. (~98 €)
- [4] Büning, H., Trenkler, G.: *Nichtparametrische statistische Methoden*. 2., völlig neu überarb. Ausg., Walter-de-Gruyter, Berlin, 1994. (~35 €, paperback)
- [5] Cleveland, W. S.: *The Elements of Graphing Data*. Wadsworth Advanced Books and Software, Monterey/Californien, 1985. (Hobart Pr., revised ed., 1994: ~56 €)
- [6] Collett, D.: *Modelling Survival Data in Medical Research*. 2nd ed., Chapman & Hall, London, 2003. (~56 € paperback)
- [7] Cox, D. R., Hinkley, D. V.: *Theoretical Statistics*. Chapman & Hall, London, 1974. (~65 €)
- [8] Fleiss, J. L., Levin, B., Paik, M. C.: *Statistical Methods for Rates and Proportions*. 3rd Edition, John Wiley, New York, 2003. (~100 €)
- [9] Fox, J.: *An R and S-PLUS Companion to Applied Regression*. Sage Publications, Thousand Oaks, 2002. (~45 € paperback)
- [10] Harrell, Jr., F. E.: *Regression Modeling Strategies. With Applications to Linear Models, Logistic Regression, and Survival Analysis*. Corr. 2nd Printing, Springer-Verlag, New York, 2002. (~97 €)
- [11] Hocking, R. R.: *Methods and Applications of Linear Models: Regression and the Analysis of Variance*. 2nd ed., John Wiley, New York, 2003. (~100 €)
- [12] Hsu, J. C.: *Multiple Comparisons*. Chapman & Hall/CRC, London, 1996. (91 €)
- [13] Mathai, A. M., Provost, S. B.: *Quadratic Forms in Random Variables: Theory and Applications*. Marcel Dekker, Inc., New York, 1992. (? €)
- [14] McCullagh, P., Nelder, J. A.: *Generalized Linear Models*, 2nd ed., Chapman & Hall, London, 1989. (~93 €)
- [15] Neter, J., Wasserman, W., Kutner, M. H.: *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs*. 3rd ed., Richard D. Irwin, Inc., 1990. (≥ 42 €, gebraucht)
- [16] Snedecor, G. W., Cochran, W. G.: *Statistical Methods*. Iowa State University Press, Ames, Iowa, 1980. (~71 €)
- [17] Stute, W.: *Kaplan-Meier Integrals*. Handbook of Statistics, Vol. 23, Elsevier, 2004.

- [18] Therneau, T. M., Grambsch, P. M.: *Modeling Survival Data: Extending the Cox Model*. 2nd printing, Springer-Verlag, New York, 2001. (~86 €)
- [19] Tukey, J. W.: *Exploratory Data Analysis* Addison-Wesley, Reading/Massachusetts, 1977. (≥ 104 €)
- [20] Venables, W. N., Smith, D. M.: *An Introduction to R*. Network Theory Ltd., 2002. (KOSTENLOS! Siehe <http://www.network-theory.co.uk>)
- [21] Venables, W. N., Ripley, B. D.: *Modern Applied Statistics with S-PLUS*. 3rd ed., Springer-Verlag, New York, 2000. (~75 paperback)
- [22] Venables, W. N., Ripley, B. D.: *Modern Applied Statistics with S*. 4th ed., Corr. 2nd printing, Springer-Verlag, New York, 2003. (~68 €)
- [23] Weisberg, S.: *Applied Linear Regression*. 3rd ed., John Wiley, New York, 2005. (~82 €)

Bemerkungen:

1. Aus Neter, Wasserman & Kutner (1990) stammt der SMSA-Datensatz.
2. Circa-Preise laut [Amazon.de](http://www.amazon.de) im Juli 2006.