1 Kodierung qualitativer Einflussgrößen

Aufgabe 1

Der Datensatz `lesen` untersucht den Zusammenhang zwischen dem Verhalten von Grundschülern und der Anzahl Fehler bei einem Lesetest:

`Fehlerzahl`:	Anzahl Fehler beim Lesetest
`sex`:	Indikator für männlich (= 1 für männlich, 0 sonst)
`Jahrgang`:	Indikator für Klassenstufe (=1 für 3. Klasse, 0 für 4.Klasse)
`Lesezeitmin`:	Lesezeit in der Schule
`WieoftLesen`:	Wie oft wird sonst gelesen

Laden Sie den Datensatz von der Vorlesungshomepage herunter und lesen Sie diesen mit dem Befehl `read.table()` in R ein. Passen Sie ein Modell mit der (entsprechend kodierten) `Wieoftlesen` Variable als unabhängiger und der `Fehlerzahl` als abhängiger Variable an. Vergleichen Sie dabei für die Variable `WieoftLesen` die verschiedenen Kodierungsarten aus den Aufgaben 1 und 2 und interpretieren Sie die Koeffizienten.

Einlesen der Daten:

```
lesen <- read.table("../../daten/lesen.txt", header = TRUE)
table(lesen$WieoftLesen)

##
##  1  2  3  4  5
## 55 39 62 21  3

# Kategorie 5 kommt kaum vor -> mit Kat. 4 zusammenfassen
lesen$WieoftLesen[lesen$WieoftLesen == 5] <- 4
#als faktor umkodieren
lesen$WieoftLesenF <- as.factor(lesen$WieoftLesen)
```

Wichtige Funktionen:

- `model.matrix()`: extrahiert die Modellmatrix aus Modellobjekten, z.B. aus Objekten der Klasse `lm` oder erstellt die Modellmatrix, z.B. aus einer Formel (`model.matrix(Fehlerzahl ~WieoftLesenF, data = lesen)`)
- `contrasts()`: gibt die einer Faktorvariable zugeordneten Kontraste zurück (per Default Dummy-Kodierung), z.B. `contrasts(lesen$WieoftLesen)`
- `contr.treatment`, `contr.sum`: geben die Kontrastematrix für Dummy- bzw. Effekt-Kodierung zurück, z.B. `contr.sum(4)`, `contr.treatment(levels(lesen$WieoftLesen))`

Eine gute Übersicht zum Einsatz von Kontrasten in R findet sich z.B. hier: `http://www.ats.ucla.edu/stat/r/library/contrast_coding.htm`

Für die folgenden Modelle und Interpretationen sei die Variable `WieoftLesen` wie folgt kodiert:

- 1: oft
- 2: regelmäßig
- 3: gelegentlich
- 4: selten/fast nie (da mit 5 zusammengefasst)

Dummy-Kodierung:
Modell mit Dummy-Kodierung (R default):

```
# WieoftLesen dummy kodiert
lm.lesen.dummy <- lm(Fehlerzahl ~ WieoftLesenF, data = lesen)
coefs.dummy <- coefficients(lm.lesen.dummy)
summary(lm.lesen.dummy)

##
## Call:
## lm(formula = Fehlerzahl ~ WieoftLesenF, data = lesen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.726  -6.164  -1.428   4.274  29.308
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.1636     1.1752  10.350  < 2e-16 ***
## WieoftLesenF2   0.5287     1.8246   0.290  0.77235
## WieoftLesenF3   4.5622     1.6145   2.826  0.00526 **
## WieoftLesenF4  10.2947     2.1322   4.828 2.98e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 176 degrees of freedom
## Multiple R-squared:  0.1391,Adjusted R-squared:  0.1245
## F-statistic: 9.482 on 3 and 176 DF,  p-value: 7.705e-06
```

Interpretation:

- Der Intercept entspricht dem Erwartungswert (hier Mittelwert) in der Referenzkategorie (hier oft lesen)
- d.h. bei Schülern die oft lesen erwartet man im Schnitt 12.1636364 Fehler (also unterdurchschnittlich viele)

```
  mean(lesen$Fehlerzahl)

## [1] 15.22222
```

- bei Schülern die regelmäßig lesen, erwartet man, im Vergleich zu Schülern die oft lesen, im Schnitt eine um 0.5286713 geringfügig höhere Fehlerzahl
- bei Schülern die gelegentlich lesen ist die erwartete Fehlerzahl, ggü. Schülern die oft lesen, um 4.5621701 erhöht
- bei Schülern die selten oder fast nie lesen erwartet man, ggü. Schülern die oft lesen, 10.294697 Fehler mehr

→Die Kontraste und die Modelmatrix haben die von der Dummy-Kodierung bekannte Struktur:

```
contr.treatment(levels(lesen$WieoftLesenF))

##   2 3 4
## 1 0 0 0
## 2 1 0 0
## 3 0 1 0
## 4 0 0 1

mm.dummy <- model.matrix(lm.lesen.dummy)
# im Folgenden geben die ersten 4 Spalten geben die Kodierung in der Modelmatrix an,
# die letzte zum Vergleich die Originalvariable WieOftLesen
head(cbind.data.frame(mm.dummy, lesen$WieoftLesenF), 10)

##    (Intercept) WieoftLesenF2 WieoftLesenF3 WieoftLesenF4 lesen$WieoftLesenF
## 1            1             0             0             1                   4
## 2            1             1             0             0                   2
## 3            1             0             0             0                   1
## 4            1             0             0             0                   1
## 5            1             0             0             0                   1
## 6            1             0             0             0                   1
## 7            1             0             1             0                   3
## 8            1             0             0             1                   4
## 9            1             0             0             1                   4
## 10           1             0             1             0                   3
```

Effekt-Kodierung:

Die Effekt-Kodierung ist in R in der Funktion `contr.sum` vorimplementiert. Hier wird per default die Letzte Kategorie in der Modell-Matrix weggelassen. Im Gegensatz zur Dummy-Kodierung bezieht sich die Interpretation aber auch nicht auf die erwartete Fehlerzahl in der Referenzkategorie.

```
## WieoftLesen effekt
contr.sum(levels(lesen$WieoftLesenF))# andere "Referenz"-Kategorie

##   [,1] [,2] [,3]
## 1    1    0    0
## 2    0    1    0
## 3    0    0    1
## 4   -1   -1   -1

lm.lesen.effect <- update(lm.lesen.dummy,
        contrasts = list(WieoftLesenF = "contr.sum"))
coefs.effect <- coefficients(lm.lesen.effect)
summary(lm.lesen.effect)

##
## Call:
## lm(formula = Fehlerzahl ~ WieoftLesenF, data = lesen, contrasts = list(WieoftLesenF = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.726  -6.164  -1.428   4.274  29.308
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    16.0100     0.6946  23.049  < 2e-16 ***
## WieoftLesenF1  -3.8464     1.0831  -3.551 0.000492 ***
```

```
## WieoftLesenF2  -3.3177     1.2068  -2.749 0.006599 **
## WieoftLesenF3   0.7158     1.0465   0.684 0.494877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 176 degrees of freedom
## Multiple R-squared:  0.1391,Adjusted R-squared:  0.1245
## F-statistic: 9.482 on 3 and 176 DF,  p-value: 7.705e-06
```

Interpretation bezieht sich nun auf den mittleren Erwartungswert über alle Ausprägungen von `WieoftLesenF`:

- Der Intercept entspricht dem mittleren Erwartungswert (hier dem Mittelwert der Gruppenmittelwerte, da keine weiteren Kovariablen im Modell)

```
group.means <- tapply(lesen$Fehlerzahl, lesen$WieoftLesenF, mean)
group.means# mittelwert von gruppe 1 entspricht Intercept in lm.lesen.dummy

##        1        2        3        4
## 12.16364 12.69231 16.72581 22.45833

mean.of.groupmeans<- mean(group.means)
mean.of.groupmeans# mittelwert der gruppenmittelwerte entspricht Intercept in lm.lesen.effect

## [1] 16.01002
```

- im Vergleich zum mittleren Erwartungswert erwartet man bei Schülern die oft lesen ca. -3.8463846 Fehler weniger
- im Vergleich zum mittleren Erwartungswert erwartet man bei Schülern die regelmäßig lesen etwa -3.3177133 Fehler weniger
- im Vergleich zum mittleren Erwartungswert erwartet man bei Schülern die gelegentlich lesen etwa 0.7157855 Fehler mehr
- die Koeffizienten geben also die Differenz der gruppenspezifischen Erwartungswerte zum mittleren Erwartungswert an

```
group.means - mean.of.groupmeans

##          1          2          3          4
## -3.8463846 -3.3177133  0.7157855  6.4483124
```

Achtung: R benutzt bei Verwendung der Effektkodierung die letzte Kategorie als 'Referenz'. Dies ist im Prinzip (bei Effektkodierung) egal, weil sich der Effekt der weggelassenen Kategorie aus den Schätzern der anderen Kategorien berechnen lässt. Allerdings muss man bei der Interpretation die Kodierung genau kennen, da die Belabelung im Falle der Effektkodierung wegfällt:

```
# moegliche Umkodierung der WieoftLesen Variable, Kategorie 1 als "Referenz"
lesen$WieoftLesenF2 <- factor(lesen$WieoftLesenF, levels = 4:1, labels = 4:1)
# aktualisieren des Models
lm.lesen.effect2 <- update(lm.lesen.dummy,
        formula = .~.-WieoftLesenF + WieoftLesenF2,
        contrasts = list(WieoftLesenF2 = "contr.sum"))
summary(lm.lesen.effect2)
```

```
##
## Call:
## lm(formula = Fehlerzahl ~ WieoftLesenF2, data = lesen, contrasts = list(WieoftLesenF2 = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.726  -6.164  -1.428   4.274  29.308
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     16.0100     0.6946  23.049  < 2e-16 ***
## WieoftLesenF21   6.4483     1.4371   4.487  1.3e-05 ***
## WieoftLesenF22   0.7158     1.0465   0.684   0.4949
## WieoftLesenF23  -3.3177     1.2068  -2.749   0.0066 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 176 degrees of freedom
## Multiple R-squared:  0.1391,Adjusted R-squared:  0.1245
## F-statistic: 9.482 on 3 and 176 DF,  p-value: 7.705e-06
```

→Die Belabelung der Koeffizienten gibt weiterhin 1,2,3 an, bezieht sich nun jedoch auf Kategorien 4,3,2.

Split-Kodierung:

Bei der Splitkodierung verwendet man alle Faktorstufen einer geordeneten (ordinalen) Variable. Theoretisch müsste man den Intercept in diesem Fall weglassen, allerdings ist die resultierende Modell-matrix identisch mit einer Modellmatrix bei der Intercept enthalten und das erste Level des Faktors weggelassen wird:

```
# definiere Kontraste-Matrix
contr.split <- lower.tri(matrix(1, nrow = 4, ncol = 4), diag = TRUE) * 1
contr.split
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    1    1    0    0
## [3,]    1    1    1    0
## [4,]    1    1    1    1
```

```
# man kann die selbst definierte Kontrastematrix der lm Funktion wie ueblich
# ueber das contrasts Argument uebergeben
#(Achtung: erste Spalte der Kontrastematrix weglassen)
lm.lesen.split <- update(lm.lesen.dummy,
        contrasts = list(WieoftLesenF = contr.split[, -1]))
coefs.split <- coefficients(lm.lesen.split)
summary(lm.lesen.split)
```

```
##
## Call:
## lm(formula = Fehlerzahl ~ WieoftLesenF, data = lesen, contrasts = list(WieoftLesenF = contr.split[,
##     -1]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.726  -6.164  -1.428   4.274  29.308
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    12.1636     1.1752  10.350  < 2e-16 ***
## WieoftLesenF1   0.5287     1.8246   0.290  0.77235
## WieoftLesenF2   4.0335     1.7813   2.264  0.02477 *
## WieoftLesenF3   5.7325     2.0954   2.736  0.00686 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 176 degrees of freedom
## Multiple R-squared:  0.1391,Adjusted R-squared:  0.1245
## F-statistic: 9.482 on 3 and 176 DF,  p-value: 7.705e-06
```

Interpretation (Vorsicht: auch hier ist die Belabelung irreführend):

- Der Intercept entspricht wie bei der Dummykodierung dem Mittelwert der Schüler die oft lesen (allgemein dem Erwartungswert der Schüler in der ersten Kategorie)
- die weiteren Koeffizienten geben dann an, um wie viel die erwartete Fehlerzahl steigt/sinkt, wenn man statt eines Schüllers der oft liest, einen Schüler der regelmäßig liest, bzw. statt eines Schülers der regelmäßig liest, einen Schüler der nur gelegentlich liest betrachtet, etc.
- So erwartet man, dass ein Schüler der selten bis fast nie liest (Kategorie 4, Koeffizient zu `WieoftLesenF3`) ca. 5.7325269 Fehler mehr macht, als ein Schüler der gelegentlich liest (Kategorie 3, Koeffizient zu `WieoftLesenF2`)
- Dementsprechend ergeben die aufsummierten Koeffizienten in diesem einfachen Fall die Gruppenmittelwerte

```
group.means# Gruppenmittelwerte

##        1        2        3        4
## 12.16364 12.69231 16.72581 22.45833

cumsum(coefs.split)# kumulativ aufsummierte Koeffizienten

##   (Intercept) WieoftLesenF1 WieoftLesenF2 WieoftLesenF3
##      12.16364      12.69231      16.72581      22.45833
```

Zusammenfassung

→Die modellierten Erwartungswerte sind in allen Modellen gleich, es ändert sich nur die Interpretation der Koeffizienten.

```
## dummy kodierung
coefs.dummy

##   (Intercept) WieoftLesenF2 WieoftLesenF3 WieoftLesenF4
##    12.1636364     0.5286713     4.5621701    10.2946970

# erwartungswerte
c(coefs.dummy[1], coefs.dummy[1] + coefs.dummy[-1])

##   (Intercept) WieoftLesenF2 WieoftLesenF3 WieoftLesenF4
##      12.16364      12.69231      16.72581      22.45833

## effekt kodierung
coefs.effect

##   (Intercept) WieoftLesenF1 WieoftLesenF2 WieoftLesenF3
##    16.0100210    -3.8463846    -3.3177133     0.7157855

# erwartungswerte
c(coefs.effect[1] + coefs.effect[-1], sum(coefs.effect[1], -coefs.effect[-1]))

## WieoftLesenF1 WieoftLesenF2 WieoftLesenF3
##      12.16364      12.69231      16.72581      22.45833

## split kodierung
coefs.split

##   (Intercept) WieoftLesenF1 WieoftLesenF2 WieoftLesenF3
##    12.1636364     0.5286713     4.0334988     5.7325269

# erwartungswerte
cumsum(coefs.split)

##   (Intercept) WieoftLesenF1 WieoftLesenF2 WieoftLesenF3
##      12.16364      12.69231      16.72581      22.45833
```

→Bei Aufnahme weiterer Kovariablen ins Modell würden die modellierten Erwartungswerte nicht mehr den Gruppenmittelwerten entsprechen. Die Interpretation der Koeffizienten bliebe aber erhalten im Bezug auf den Erwarungswert bei festhalten der restlichen Kovariablen (ceteris paribus)

Add-On :
Wie man bei der Effektkodierung und nun bei der Split-Kodierung gesehen hat, kann die standardmäßig implementierte Belabelung verwirrend sein. Alternativ kann man deshalb die Modelmatrix selbst generieren und diese zur Modellanpassung übergeben:

```
# siehe Uebung fuer das Schema zur Kodierung der zu erzeugenden Dummy-Variablen
# hier wird in einem Schritt die Modelmatrix erzeugt:
mm.split <- (matrix(1:4, nrow = nrow(lesen), ncol = 4, byrow = TRUE) <= lesen$WieoftLesen) * 1
# und die Spaltennamen angepasst
colnames(mm.split) <- paste0("WieoftLesen", 1:4)
lesen.neu <- cbind.data.frame(Fehlerzahl = lesen$Fehlerzahl, mm.split)
# passe Modell mit allen Variablen im Datensatz an, lasse Intercept weg
lm.lesen.split2 <- lm(Fehlerzahl ~ . - 1, data = lesen.neu)
summary(lm.lesen.split2)
```

```
##
## Call:
## lm(formula = Fehlerzahl ~ . - 1, data = lesen.neu)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.726  -6.164  -1.428   4.274  29.308
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## WieoftLesen1  12.1636     1.1752  10.350  < 2e-16 ***
## WieoftLesen2   0.5287     1.8246   0.290  0.77235
## WieoftLesen3   4.0335     1.7813   2.264  0.02477 *
## WieoftLesen4   5.7325     2.0954   2.736  0.00686 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 176 degrees of freedom
## Multiple R-squared:  0.7664,Adjusted R-squared:  0.7611
## F-statistic: 144.4 on 4 and 176 DF,  p-value: < 2.2e-16
```

```
# pruefe ob koeffizientenschaetzer identisch
all(coefficients(lm.lesen.split2) == coefficients(lm.lesen.split))
```

```
## [1] TRUE
```

→Die Belabelung entspricht hier den tatsächlichen Faktorstufen
→Äquivalentes Vorgehen könnte man bei der Effektkodierung oder anderen selbst definierten Kodierung anwenden