Angewandte Regression — Musterlösungen zur Serie 5

1. a) Ohne Variablenselektion:

R-Code:

```
d.concept <- read.table("http://stat.ethz.ch/Teaching/Datasets/
             WBL/concept.dat",header=TRUE)
attach(d.concept)
model1 <- regr(gpa~iq+alter+sex+total+c1+c2+c3+c4+c5+c6, data=d.concept)
summary(model1)
```

R-Output:

```
>
Call:
regr(formula = gpa ~ iq + alter + sex + total + c1 + c2 + c3 +
    c4 + c5 + c6, data = d.concept)
Fitting function  lm

Terms:
                   coef       stcoef      signif   R2.x df p.value
(Intercept)  2.520984023  0.000000000  0.27199074     NA  1  0.5890
iq           0.073390658  0.460395291  2.09481514 0.2898  1  0.0001
alter       -0.410205964 -0.123711419 -0.70381025 0.1120  1  0.1647
sex         -0.575292611 -0.134957849 -0.73019102 0.1555  1  0.1497
total       -0.066639252 -0.393960015 -0.56869311 0.7747  1  0.2604
c1           0.188234182  0.255619950  0.84927309 0.4814  1  0.0947
c2           0.128613669  0.233979733  0.72342766 0.5174  1  0.1534
c3           0.122059074  0.200728570  0.64017323 0.5022  1  0.2057
c4          -0.003138792 -0.004392463 -0.01386129 0.5074  1  0.9780
c5           0.142543637  0.195531259  0.67029961 0.4649  1  0.1854
c6           0.018149574  0.017170523  0.05438534 0.5056  1  0.9139

St.dev.error:  1.441   on 67 degrees of freedom
Multiple R^2:  0.5903   Adjusted R-squared: 0.5291
F-statistic:   9.653   on 10 and 67 d.f.,  p.value: 9.589e-10

Factor(s) with two levels converted to 0-1 variable(s):
  sex
0 "1"
1 "2"
```

Mit multipler Regression stellt sich der iq als signifikant heraus.

b) Mit Variablenselektion:

R-Code:

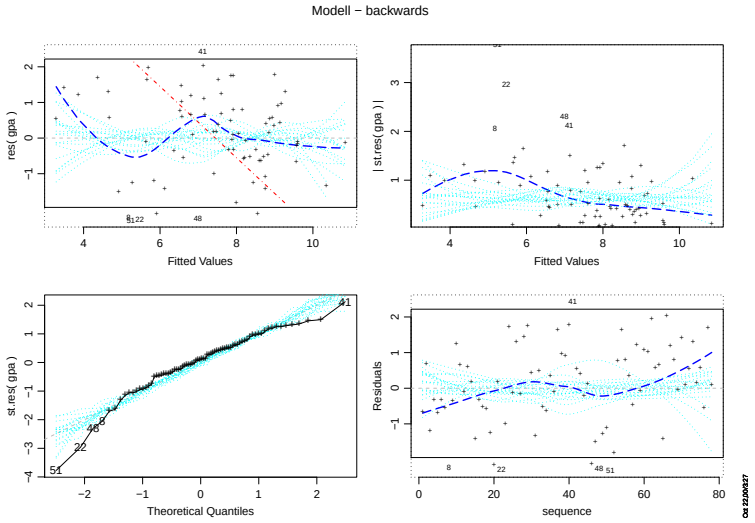
```
stepback <- step(model1,direction="backward")
formula(stepback)
modback <- regr(formula(stepback),data=d.concept)
summary(modback)
```

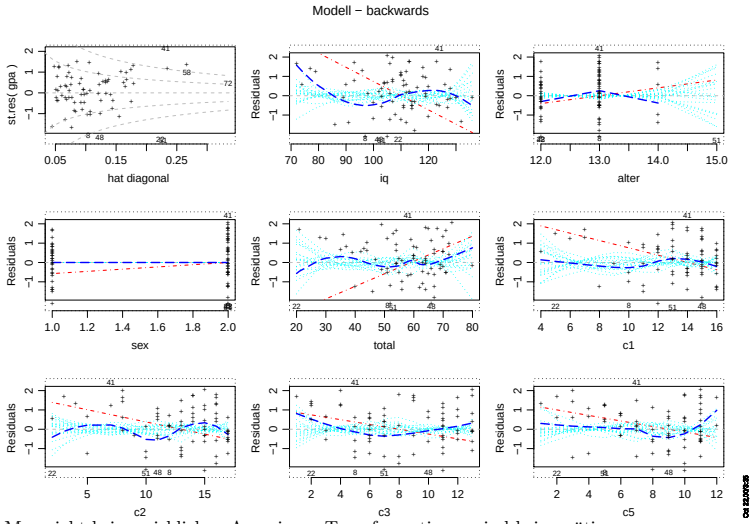

Wenn man den obigen R-Code ausführt, sieht man, dass Variablen im Modell aufgenommen werden, jedoch stellt sich immer der iq als signifikant heraus.

R-Output:

```
Call:
regr(formula = formula(stepback), data = d.concept)
Fitting function  lm

Terms:
                   coef      stcoef      signif   R2.x df p.value
(Intercept)  2.47444426  0.0000000  0.2721850     NA  1  0.5889
iq           0.07352652  0.4612476  2.2631592 0.2457  1  0.0000
alter       -0.40666273 -0.1226428 -0.7167479 0.1015  1  0.1573
sex         -0.57485722 -0.1348557 -0.7701066 0.1221  1  0.1290
total       -0.06493277 -0.3838716 -0.7434638 0.7022  1  0.1426
c1           0.18921701  0.2569546  0.9248499 0.4467  1  0.0693
c2           0.12618337  0.2295584  0.7454130 0.5008  1  0.1416
c3           0.12419305  0.2042379  0.6973449 0.4751  1  0.1686
c5           0.14315563  0.1963707  0.6998099 0.4521  1  0.1672

St.dev.error:   1.42   on 69 degrees of freedom
Multiple R^2:  0.5902   Adjusted R-squared: 0.5427
F-statistic:   12.42   on 8 and 69 d.f.,  p.value: 7.587e-11

Factor(s) with two levels converted to 0-1 variable(s):
  sex
0 "1"
1 "2"
```

c) `plot(modback)`



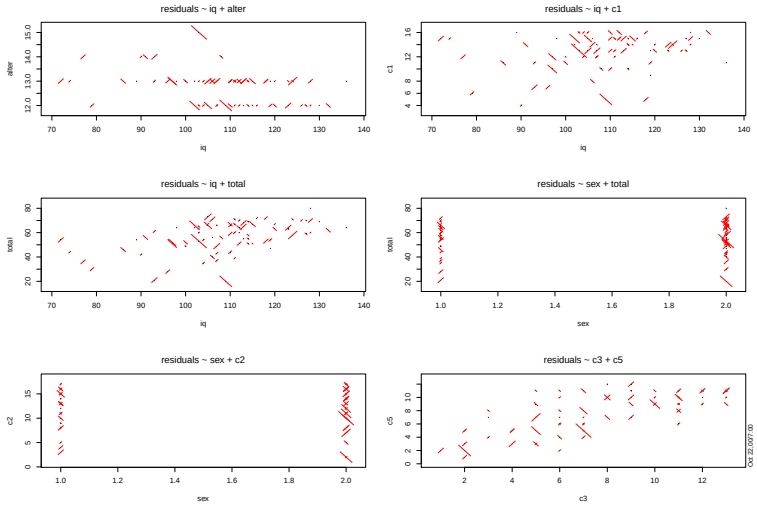


Man sieht keine wirklichen Ausreisser. Transformationen sind keine nötig.

d) Wir betrachten die folgenden Residual-plots zweier Eingangsvariablen:

```
plres2x(~ iq + alter,modback)
plres2x(~ iq + c1,modback)
plres2x(~ iq + total,modback)
plres2x(~ sex + total,modback)
plres2x( ~ sex + c2, modback)
plres2x(~ c3 + c5,modback)
```

Es



sieht nicht nach Wechselwirkungen in den ersten beiden Plots aus.

e) **R-Code:**

```
add1.regr(modback)
```

R-Output:

```
Single term additions

Model:
gpa ~ iq + alter + sex + total + c1 + c2 + c3 + c5
          Df Sum of Sq
 RSS     AIC F value     Pr(F)
<none>                  139.090  63.116
I(iq^2)       1     4.154 134.935  62.751  2.0935 0.1525210
I(alter^2)    1    22.476 116.614  51.368 13.1061 0.0005618 ***
I(total^2)    1     0.416 138.673  64.882  0.2040 0.6529275
I(c1^2)       1     0.054 139.035  65.085  0.0266 0.8710275
I(c2^2)       1     0.007 139.082  65.112  0.0035 0.9532760
I(c3^2)       1     6.397 132.693  61.444  3.2782 0.0746258 .
I(c5^2)       1     3.168 135.922  63.319  1.5849 0.2123615
iq:alter      1     0.099 138.991  65.060  0.0484 0.8264768
iq:sex        1     1.376 137.713  64.340  0.6795 0.4126481
iq:total      1     5.983 133.107  61.686  3.0564 0.0849307 .
iq:c1         1     0.561 138.528  64.800  0.2756 0.6013222
iq:c2         1     3.857 135.233  62.922  1.9394 0.1682753
iq:c3         1     0.936 138.153  64.589  0.4609 0.4995198
iq:c5         1     0.019 139.070  65.105  0.0094 0.9231646
alter:sex     1     0.304 138.785  64.945  0.1491 0.7006489
alter:total   1     0.850 138.240  64.638  0.4180 0.5201130
alter:c1      1     0.671 138.418  64.739  0.3297 0.5677023
alter:c2      1     1.557 137.532  64.238  0.7700 0.3833105
alter:c3      1     0.034 139.055  65.097  0.0167 0.8975330
alter:c5      1     0.293 138.796  64.951  0.1437 0.7058409
sex:total     1     7.869 131.221  60.574  4.0776 0.0474005 *
sex:c1        1     4.866 134.224  62.338  2.4650 0.1210515
sex:c2        1     8.309 130.781  60.311  4.3203 0.0414373 *
sex:c3        1     2.084 137.005  63.938  1.0346 0.3126920
sex:c5        1     4.623 134.467  62.480  2.3376 0.1309200
total:c1      1     1.584 137.506  64.223  0.7832 0.3792894
total:c2      1     0.068 139.022  65.078  0.0331 0.8561746
total:c3      1     2.410 136.680  63.753  1.1990 0.2773843
total:c5      1     0.540 138.549  64.812  0.2653 0.6081950
c1:c2         1     2.032 137.057  63.968  1.0084 0.3188515
c1:c3         1     1.281 137.809  64.394  0.6319 0.4294122
c1:c5         1     2.354 136.735  63.784  1.1708 0.2830593
c2:c3         1     0.904 138.185  64.607  0.4450 0.5069865
c2:c5         1     0.395 138.694  64.894  0.1937 0.6612543
c3:c5         1     7.054 132.036  61.056  3.6327 0.0608820 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

f) Wir nehmen das volle lineare Modell mit den signifikanten quadratischen Termen und Wechselwirkungen:

```
model2 <- regr(gpa~iq+alter+sex+total+c1+c2+c3+c4+c5+c6
          +I(alter^2)+I(c3^2)+iq:total+sex:total+sex:c2+c3:c5, data=d.concept)
summary(model2)
stepback <- step(model2,direction="backward")
formula(stepback)
```

```
modback <- regr(formula(stepback),data=d.concept)
summary(modback)
```

Mit der backward-analyse erhalten wir:

```
> Call:
regr(formula = formula(stepback), data = d.concept)
Fitting function  lm

Terms:
                      coef     stcoef     signif   R2.x df p.value
(Intercept) -1.348058e+02  0.0000000 -1.7432472     NA  1  0.0009
iq          -2.177437e-02 -0.1365953 -0.2030645 0.7985  1  0.6865
alter        2.270072e+01  6.8461658  1.8900714 0.9626  1  0.0003
sex         -2.433626e+00 -0.5709041 -0.8427025 0.7999  1  0.0972
total       -2.888571e-01 -1.7076746 -1.3323839 0.8942  1  0.0098
c1           2.213165e-01  0.3005454  1.2070279 0.4556  1  0.0188
c2           1.124524e-01  0.2045784  0.7501769 0.5029  1  0.1390
c5           1.661921e-01  0.2279705  0.9065870 0.4609  1  0.0748
I(alter^2)  -8.942918e-01 -6.9758117 -1.9272515 0.9625  1  0.0003
I(c3^2)      7.318765e-03  0.1903856  0.7930607 0.4353  1  0.1181
iq:total     1.829685e-03  1.5601484  0.9597051 0.9166  1  0.0597
sex:total    3.335942e-02  0.4776021  0.6676067 0.8105  1  0.1871

St.dev.error:  1.251   on 66 degrees of freedom
Multiple R^2:  0.6957   Adjusted R-squared: 0.645
F-statistic:   13.72   on 11 and 66 d.f.,  p.value: 3.417e-13
```

Ausreisser: keine.
Transformationen: keine nötig

Wichtig: Datensatz mit `detach(concept)` wieder abhängen!

2. a) Schrittweise rückwärts: Wir betrachten das volle Modell

$$\log(\texttt{RUT}) = \beta_0 + \beta_1 \log(\texttt{VISC}) + \beta_2\ \texttt{ASPH} + \beta_3\ \texttt{BASE} + \beta_4\ \texttt{FINES} + \beta_5\ \texttt{VOIDS} + \beta_6\ \texttt{RUN}$$

und eliminieren schrittweise die am wenigsten signifikante Variable

```
> r.bw <- step(r.asp, direction="backward")
Start:  AIC=-129
log10(RUT) ~ log10(VISC) + ASPH + BASE + FINES + VOIDS + RUN

             Df Sum of Sq    RSS    AIC
- FINES       1    0.0039    0.3 -130.7
- BASE        1    0.0065    0.3 -130.4
<none>                      0.3 -129.1
- RUN         1       0.1    0.4 -125.8
- VOIDS       1       0.1    0.4 -121.9
- ASPH        1       0.2    0.5 -113.2
- log10(VISC) 1       0.6    0.9  -96.4

Step:  AIC=-131
log10(RUT) ~ log10(VISC) + ASPH + BASE + VOIDS + RUN

             Df Sum of Sq    RSS    AIC
- BASE        1     0.012    0.3 -131.5
<none>                      0.3 -130.7
- RUN         1       0.1    0.4 -127.7
- VOIDS       1       0.1    0.4 -121.8
- ASPH        1       0.2    0.6 -114.7
```



```
- log10(VISC) 1       0.7    1.0  -96.6

   ...

Call:
regr(formula = log10(RUT) ~ log10(VISC) + ASPH + VOIDS + RUN,
    data = d.asp)
Terms:
             coef stcoef signif  R2.x df p.value
(Intercept) -1.742  0.000  -1.31    NA  1   0.012
log10(VISC) -0.547 -0.852  -4.07 0.671  1   0.000
ASPH         0.465  0.166   2.11 0.124  1   0.000
VOIDS        0.144  0.137   1.56 0.216  1   0.004
RUN         -0.222 -0.186  -0.91 0.663  1   0.073
St.dev.error:  0.111   on 26 degrees of freedom
Multiple R^2:  0.971  Adjusted R-squared: 0.966
F-statistic:    216   on 4 and 26 d.f.,  p.value:    0
```

Das Endmodell lautet mit schrittweiser Variablenselektion rückwärts:

$$\log(\texttt{RUT}) = \beta_0 + \beta_1 \log(\texttt{VISC}) + \beta_2\ \texttt{ASPH} + \beta_3\ \texttt{VOIDS} + \beta_4\ \texttt{RUN}\,.$$

Schrittweise vorwärts: Wir betrachten das Modell $\log(\texttt{RUT}) = \beta_0$ und fügen schrittweise die signifikanteste der verbleibenden Variable hinzu.

```
> r.start <- regr(log10(RUT) ~ 1, data = d.asp)
> r.fw <- step(r.start, scope = formula(r.asp), direction="forward")
Start:  AIC=-29.9
log10(RUT) ~ 1

             Df Sum of Sq    RSS    AIC
+ log10(VISC) 1      10.5    0.6 -118.0
+ RUN         1       9.5    1.5  -89.4
+ VOIDS       1       1.9    9.2  -33.7
+ FINES       1       1.1    9.9  -31.3
<none>                      11.1  -30.0
+ BASE        1       0.4   10.7  -29.1
+ ASPH        1       0.3   10.8  -28.7

Step:  AIC=-118
log10(RUT) ~ log10(VISC)

       Df Sum of Sq    RSS    AIC
+ ASPH  1       0.1    0.5 -122.0
<none>                 0.6 -118.0
+ VOIDS 1    0.0298    0.6 -117.6
+ RUN   1    0.0231    0.6 -117.2
+ FINES 1    0.0084    0.6 -116.4
+ BASE  1    0.0055    0.6 -116.3

   ...

Step:  AIC=-131
log10(RUT) ~ log10(VISC) + ASPH + VOIDS + RUN

       Df Sum of Sq    RSS    AIC
<none>                 0.3 -131.5
+ BASE  1    0.0122    0.3 -130.7
+ FINES 1    0.0096    0.3 -130.4

> summary(r.fw)
Call:
```

```
regr(formula = log10(RUT) ~ log10(VISC) + ASPH + VOIDS + RUN,
    data = d.asp)
Terms:
             coef stcoef signif  R2.x df p.value
(Intercept) -1.742  0.000  -1.31    NA  1   0.012
log10(VISC) -0.547 -0.852  -4.07 0.671  1   0.000
ASPH         0.465  0.166   2.11 0.124  1   0.000
VOIDS        0.144  0.137   1.56 0.216  1   0.004
RUN         -0.222 -0.186  -0.91 0.663  1   0.073
St.dev.error:  0.111   on 26 degrees of freedom
Multiple R^2:  0.971  Adjusted R-squared: 0.966
F-statistic:    216   on 4 and 26 d.f.,  p.value:    0
```

b) **All Subsets:** Wir berechnen alle möglichen Untermodelle und untersuchen die besten Modelle mit einer, zwei, drei, ... Variablen. Wir suchen uns diejenigen Modelle mit den kleinsten *Cp*-Werten.

```
> library(leaps)
> r.allsub <- regsubsets(log10(RUT)~log10(VISC)+ ASPH + BASE + FINES
                         + VOIDS + RUN, data=d.asp, nbest=2)
## oder
> r.allsub <- regsubsets(formula(r.asp), data=d.asp, nbest=2)

> summary(r.allsub)
Subset selection object
Call: regsubsets.formula(formula(r.asp), data = d.asp, nbest = 2)
6 Variables  (and intercept)
           Forced in Forced out
log10(VISC)     FALSE      FALSE
ASPH            FALSE      FALSE
BASE            FALSE      FALSE
FINES           FALSE      FALSE
VOIDS           FALSE      FALSE
RUN             FALSE      FALSE
2 subsets of each size up to 6
Selection Algorithm: exhaustive
         log10(VISC) ASPH BASE FINES VOIDS RUN
1  ( 1 ) "*"         " "  " "  " "   " "   " "
1  ( 2 ) " "         " "  " "  " "   " "   "*"
2  ( 1 ) "*"         "*"  " "  " "   " "   " "
2  ( 2 ) "*"         " "  " "  " "   "*"   " "
3  ( 1 ) "*"         "*"  " "  " "   "*"   " "
3  ( 2 ) "*"         "*"  " "  " "   " "   "*"
4  ( 1 ) "*"         "*"  " "  " "   "*"   "*"
4  ( 2 ) "*"         "*"  " "  "*"   "*"   " "
5  ( 1 ) "*"         "*"  "*"  " "   "*"   "*"
5  ( 2 ) "*"         "*"  " "  "*"   "*"   "*"
6  ( 1 ) "*"         "*"  "*"  "*"   "*"   "*"

> r.cp <- summary(r.allsub)$cp
> names(r.cp) <- c(t(outer(1:5, 1:2, paste, sep=".")), 6.1)
> cat("\n Cp-Wert für Modell x.y \n")

 Cp-Wert für Modell x.y
> r.cp
  1.1   1.2   2.1   2.2   3.1   3.2   4.1   4.2   5.1   5.2   6.1
20.33 91.96 13.98 20.00  5.66 12.21  4.26  7.44  5.30  5.51  7.00
```

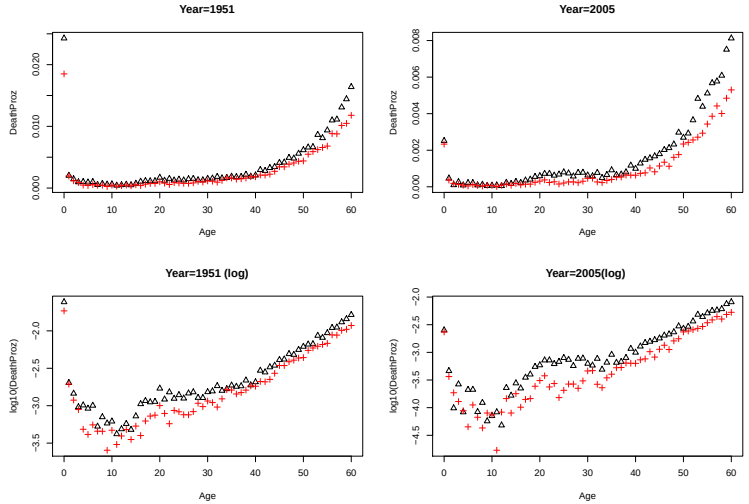
Das Endmodell lautet mit schrittweiser Variablenselektion vorwärts gleich wie das Endmodell der rückwärts Variablenselektion. Auch bei der All Subset Methode ist dieses Modell (Modell 4.1) das beste, gemessen am Cp-Wert.

Bemerkung: Das Resultat der Forward-Methode kann deutlich verschieden von der Backward-Methode sein, welche der Forward-Methode in der Regel vorzuziehen ist. Falls einige erklärende Variablen stark miteinander korreliert sind, können sich die besten Modelle je nach Verfahren unterscheiden.

3. **a)** R-Code:

```
d.mort <-
read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/mort.dat",header=TRUE)
d.mort0 <- d.mort[(d.mort$Age<=80) & ((d.mort$Year==2005) |
          (d.mort$Year==1951)),]

par(mfrow=c(2,2))
plot(DeathProz~Age,data=d.mort0[(d.mort0$Age<=60) &
    (d.mort0$Year==1951),], col=Gender+1, pch=Gender+2,main="Year=1951")
plot(DeathProz~Age,data=d.mort0[(d.mort0$Age<=60) &
    (d.mort0$Year==2005),], col=Gender+1, pch=Gender+2,main="Year=2005")
plot(log10(DeathProz)~Age,data=d.mort0[(d.mort1$Age<=60) &
    (d.mort0$Year==1951),], col=Gender+1, pch=Gender+2,
    main="Year=1951 (log)")
plot(log10(DeathProz)~Age,data=d.mort0[(d.mort0$Age<=60) &
    d.mort0$Year==2005,], col=Gender+1, pch=Gender+2,
    main="Year=2005 (log)"
```

R-Output:



Bemerkung:

- Die Sterbewahrscheinlichkeiten sind im Jahr 2005 bedeutend kleiner als im Jahr 1951.
- Wir werden sehr wahrscheinlich Schwierigkeiten bekommen in der linearen Modellierung für das ganze Specktrum, da im Alter zwischen 0 und 10 die Sterbewahrscheinlichkeiten stark sinken, zwischen 10 und 20 wieder steigen, zwischen 20 und 30 Konstant bleibend (sogar leicht fallend) und im höheren alter wieder monoton steigend.

b) Die finallen Modelle sind:

```
d.mort1 <- d.mort0[(d.mort0$Year==2005),]
```

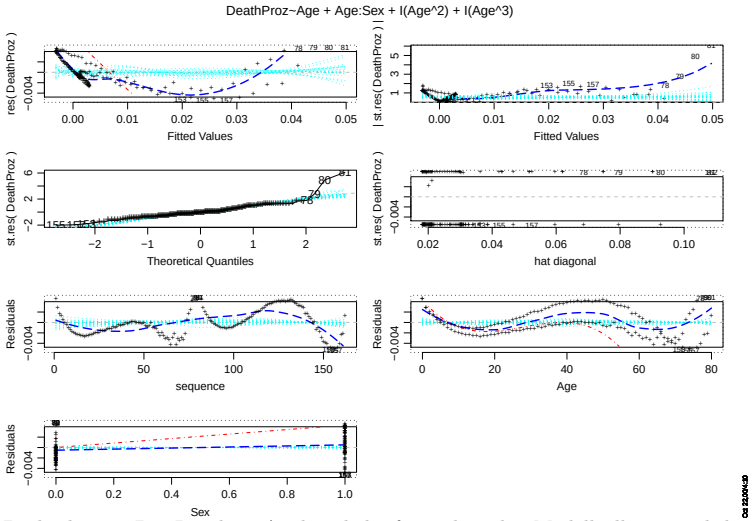
- Modell 1: für das nicht-transformiertes Modell erhalten wir:
  R-Code:

```
r.mort1 <- regr(formula=DeathProz~Age+Sex,data=d.mort1)
add1.regr(r.mort1)
r.mort1 <- regr(formula=DeathProz~Age*Sex+I(Age^2)*Sex
                +I(Age^3)*Sex,data=d.mort1)
summary(r.mort1)
```

  R-Output:

```
> add1.regr(r.mort1)
Single term additions

Model:
DeathProz ~ Age + Sex
         Df Sum of Sq      RSS      AIC F value   Pr(F)
<none>                  0.07 -2719.31
I(Age^2)  1      0.04     0.04 -2949.73 335.677 < 2e-16 ***
Age:Sex   1 0.0007654     0.07 -2720.78   3.437 0.06467 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(r.mort1)
Call:
regr(formula = DeathProz ~ Age * Sex + I(Age^2) * Sex + I(Age^3) *
    Sex, data = d.mort1)
Fitting function  lm

Terms:
                      coef      stcoef     signif   R2.x df p.value
(Intercept)  -3.932926e-03  0.00000000 -1.9056321     NA  1  0.0002
Age           9.650666e-04  1.83489999  4.2927984 0.9274  1  0.0000
Sex           1.892865e-03  0.07696326  0.6485268 0.7383  1  0.2021
I(Age^2)     -4.225177e-05 -6.64258366 -6.4476179 0.9699  1  0.0000
I(Age^3)      4.965719e-07  5.96613934  9.2252527 0.9520  1  0.0000
Age:Sex      -4.376591e-04 -0.92351696 -1.3765887 0.9537  1  0.0073
Sex:I(Age^2)  1.865965e-05  2.63731918  2.0134596 0.9763  1  0.0001
Sex:I(Age^3) -2.139609e-07 -2.13900399 -2.8107060 0.9592  1  0.0000

St.dev.error:  0.00246   on 154 degrees of freedom
Multiple R^2:  0.9619   Adjusted R-squared: 0.9602
F-statistic:   556.2   on 7 and 154 d.f.,  p.value:     0

Factor(s) with two levels converted to 0-1 variable(s):
  Sex
0 "0"
1 "1"
```

  Residuen-Analyse:



Beobachtung: Die Residuen-Analyse bekräftigt, dass das Modell alle wesentlichen Voraussetzungen nicht erfüllt.

- Modell 2: für das Log-transformiertes Modell
  R-Code:

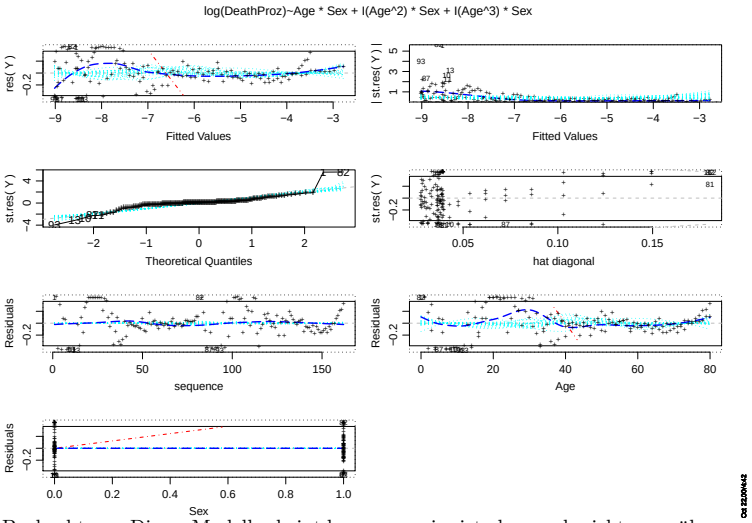
```
r.mort2 <- regr(formula=log(DeathProz)~Age+Sex,data=d.mort1)
add1.regr(r.mort2)
r.mort2 <- regr(formula=log(DeathProz)~Age*Sex+I(Age^2)*Sex
                        +I(Age^3)*Sex,data=d.mort1)
summary(r.mort2)
```

  R-Output:

```
> add1.regr(r.mort2)
Single term additions

Model:
log(DeathProz) ~ Age + Sex
         Df Sum of Sq      RSS      AIC  F value  Pr(F)
<none>                 239.994  -91.242
I(Age^2)  1    61.613  178.381 -185.370 110.5281 <2e-16 ***
Age:Sex   1     0.353  239.640  -89.720   0.4719 0.4926
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(r.mort2)

Call:
regr(formula = log(DeathProz) ~ Age * Sex + I(Age^2) * Sex +
    I(Age^3) * Sex, data = d.mort1)
Fitting function  lm

Terms:
                      coef      stcoef       signif   R2.x df p.value
(Intercept)  -8.548668e+00  0.00000000 -20.14427495     NA  1  0.0000
Age          -4.195296e-03 -0.05056388  -0.09075607 0.9274  1  0.8579
Sex          -8.569355e-02 -0.02208689  -0.14278623 0.7383  1  0.7783
```

```
I(Age^2)      1.580461e-03  1.57506566   1.17291922 0.9699  1  0.0218
I(Age^3)     -7.822034e-06 -0.59573566  -0.70671719 0.9520  1  0.1647
Age:Sex      -6.626831e-02 -0.88641573  -1.01368723 0.9537  1  0.0470
Sex:I(Age^2)  1.921939e-03  1.72195368   1.00857654 0.9763  1  0.0481
Sex:I(Age^3) -1.490134e-05 -0.94433319  -0.95199850 0.9592  1  0.0619

St.dev.error:  0.5059   on 154 degrees of freedom
Multiple R^2:  0.9354   Adjusted R-squared: 0.9324
F-statistic:   318.3   on 7 and 154 d.f.,  p.value:     0

Factor(s) with two levels converted to 0-1 variable(s):
  Sex
0 "0"
1 "1"
```

Residuen-Analyse:



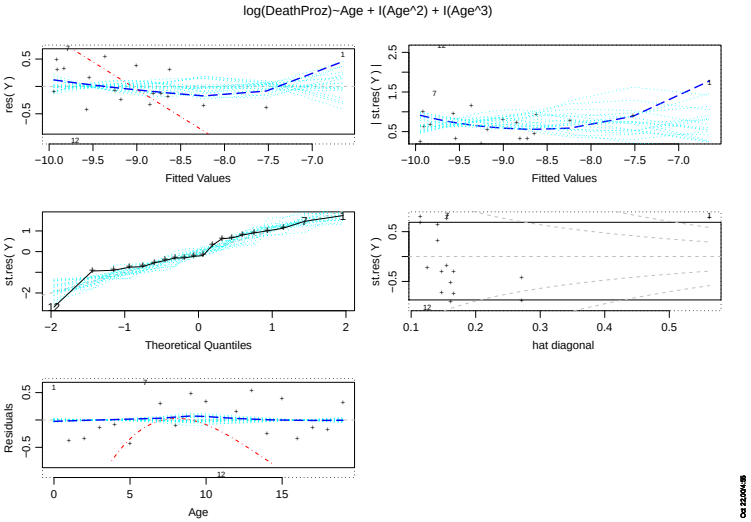
Beobachtung: Dieses Modell scheint besser zu sein, ist aber auch nicht ganz überzeugend.

Verbesserungsvorschläge:

- Mann und Frau getrennt anschauen
- Was zu Beginn betreffend den möglichen Schwierigkeiten schon vermutet wurde, hat sich in den Residuden-Analyse bestätigt. Es scheint angebracht, zusätzliche drei Modell zu machen, Alter zwischen 0 und 20, Alter zwischen 20 und 30, Alter grösser als 30.
  Als Beispiel betrachten wir den Altersbereich 0 bis 20 für Frauen (Log-Modell). Wir erhalten ein nicht fertiges Modell, das die Voraussetzungen besser erfüllt:

```
d.mort3 <- d.mort1[(d.mort1$Gender==1)&(d.mort1$Age<=19),]
r.mort3 <- regr(formula=log(DeathProz)~Age+I(Age^2)+I(Age^3),data=d.mort3)
summary(r.mort3)
```

Residuen-Analyse:



- First-Aid-Transformationen (arcsin($\sqrt{(q_x)}$))
- andere Regressionstechniken