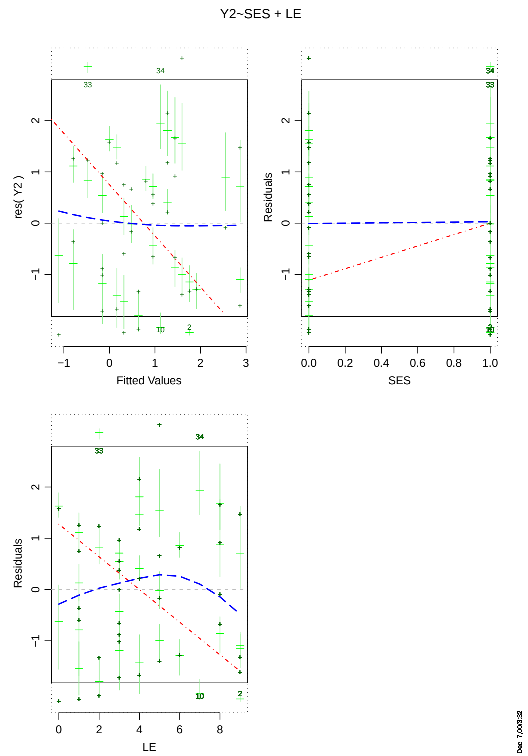
Weiterbildungs-Lehrgang in Angewandter Statistik ETHZ 2009/2011 7.12.09

Angewandte Regression — Musterlösungen zur Serie 11

1. a) Diese Aufgabe haben wir schon in der letzten Serie gemacht. Diesesmal beützen wir `regr()`.

```
 d.mental <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/mental.dat",head
d.mental$Y2 <- ordered(d.mental$Y,levels=c("Well","Mild","Moderate","Impaired"))
r.mantel <- regr(Y2~SES+LE,data=d.mental,family="ordered")
summary(r.mantel)

> Call:
regr(formula = Y2 ~ SES + LE, data = d.mental, family = "ordered")
Fitting function  polr

Terms:
      coef stcoef signif R2.x df p.value
SES -1.111 -0.560 -0.896   NA  1   0.064
LE   0.319  0.872  1.298   NA  1   0.005

Intercepts:
                   Value Std. Error
Well|Mild         -0.282      0.642
Mild|Moderate      1.213      0.661
Moderate|Impaired  2.209      0.721

Family is ordered.  Dispersion parameter taken to be 1.
AIC: NULL

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

b) Die Residuenplots



sind akzeptabel:

- die Glättungskurve im TA ist fast mit der Nullline identitisch.
- die Glättungskurve im `LE`-Residuen-Plot ist nach unten gekrümmt, aber kann als aktzeptal betrachtet werden.

2. Die Daten werden eingelesen:

```
d.kevlar <-
read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/kevlar49.dat",
          header=T)
d.kevlar$Spool <- factor(d.kevlar$Spool)
d.kevlar <- d.kevlar[d.kevlar$Stress>=24,]
```

a)
```
l.kevlar <- regr(log(Failure)~Stress+Spool, data=d.kevlar)
summary(l.kevlar)
plot(l.kevlar)

> Call:
regr(formula = log(Failure) ~ Stress + Spool, data = d.kevlar)
```

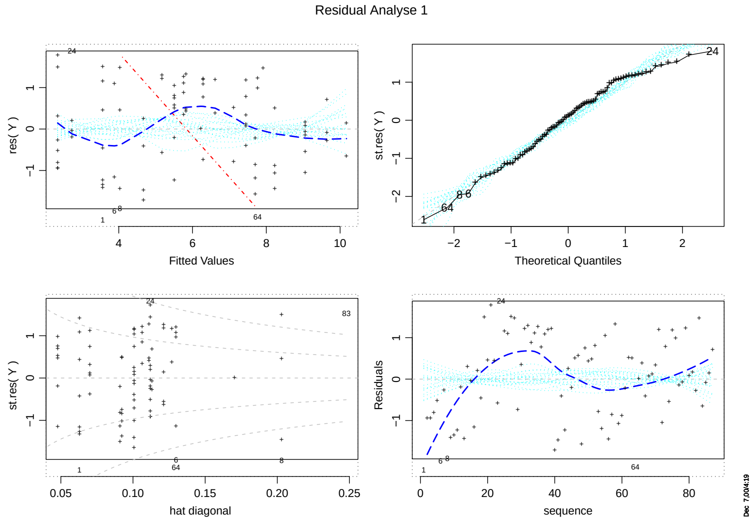
```
Fitting function  lm

Terms:
              coef stcoef signif  R2.x df p.value
(Intercept) 33.216  0.000   8.27    NA  1       0
Stress      -0.924 -0.662  -6.49 0.060  1       0
Spool           NA     NA   7.27 0.009  7       0

Coefficients for factors:
$Spool
     1      2      3      4      5      6      7      8
 0.000 -2.192 -3.027  0.531 -1.726 -1.878 -3.415 -0.585

St.dev.error:   1.1   on 78 degrees of freedom
Multiple R^2:  0.819   Adjusted R-squared:  0.8
F-statistic:   44.1   on 8 and 78 d.f.,  p.value:    0
```

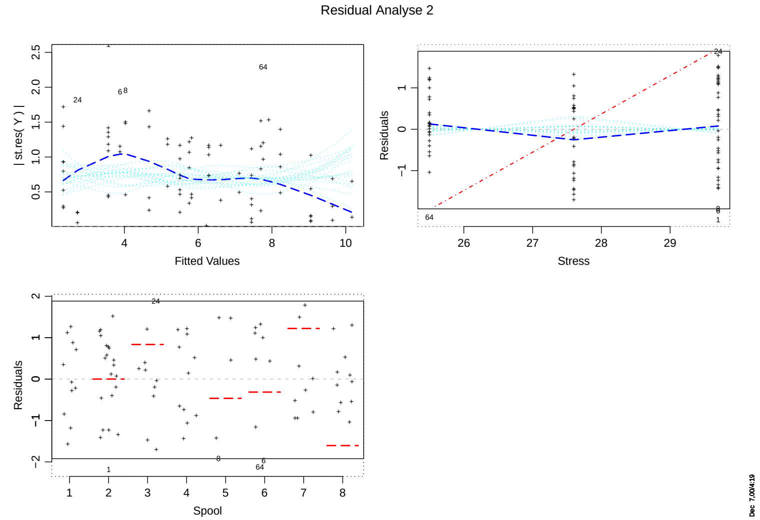
Alle Eingangsvariablen sind signifikant.

Residuenanalyse:

- TA: die Glättungskurve ist eine Wellenlinie
- Streuungsdiagramm: die Glättungskurve ist fallend
- QQ-Plot: der QQ-Plot passt nicht optimal, ist eher kurzschwänziger
- Leverage-Plot: wenige Daten haben einen grösseren Hebelarm.
- Unabhängigkeit: ist verletzt
- Eingangsvariablen-Residuen: akzeptabel

Die Annahmen nicht erfüllt.



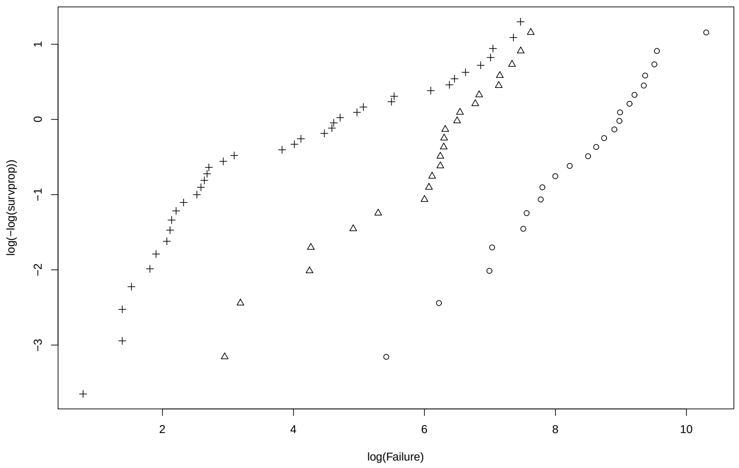


b) Die empirische Survival-Funktion $\hat{S}(x)$ lässt sich zb wie folgt bestimmen

```
d.kevlar$index <- 1
g.kelvar <- split(d.kevlar,d.kevlar$Stress)
h.kevlar1 <- NULL
h.kevlar1<-cumsum(g.kelvar$'29.7'$index)/length(g.kelvar$'29.7'$index)
h.kevlar1<-c(h.kevlar1,cumsum(g.kelvar$'27.6'$index)/
           length(g.kelvar$'27.6'$index))
h.kevlar1<- c(h.kevlar1,cumsum(g.kelvar$'25.5'$index)/
            length(g.kelvar$'25.5'$index))
d.kevlar$survprop <- 1-h.kevlar1
```

Den Plot machen wir mit

```
plot(~log(Failure)+log(-log(survprop)),data=d.kevlar,
     pch=as.numeric(as.factor(Stress)))
```



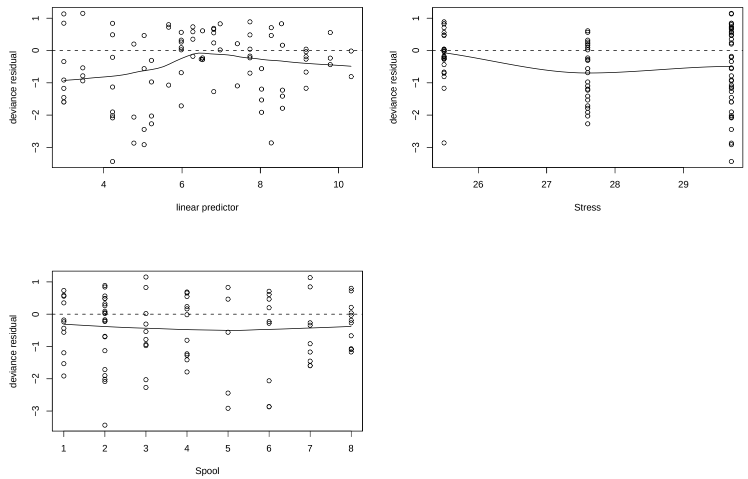
c)
```
r.kevlar <- survreg(Surv(Failure,rep(1,nrow(d.kevlar)))~Stress+Spool,
            data=d.kevlar)
summary(r.kevlar)

> Call:
survreg(formula = Surv(Failure, rep(1, nrow(d.kevlar))) ~ Stress +
    Spool, data = d.kevlar)
              Value Std. Error      z        p
(Intercept) 31.087     1.5214  20.43 8.42e-93
Stress      -0.835     0.0532 -15.71 1.21e-55
Spool2      -2.051     0.3181  -6.45 1.14e-10
Spool3      -2.812     0.3708  -7.58 3.37e-14
Spool4       0.534     0.3622   1.47 1.41e-01
Spool5      -1.245     0.4589  -2.71 6.67e-03
Spool6      -1.507     0.3866  -3.90 9.69e-05
Spool7      -3.296     0.3853  -8.55 1.19e-17
Spool8      -0.618     0.3617  -1.71 8.74e-02
Log(scale)  -0.167     0.0868  -1.92 5.44e-02

Scale= 0.846

Weibull distribution
Loglik(model)= -631   Loglik(intercept only)= -709
Chisq= 155 on 8 degrees of freedom, p= 0
Number of Newton-Raphson Iterations: 7
n= 87
```
Alle Variablen sind signifikant.

d) Wir fassen zusammen (siehe dazu auch Skript 18.1.d ybd 18.1.g):

Lineare-Regression aus a):	$\log(\texttt{Failure}) = \beta_o + \beta_1\texttt{Stress} + \beta_2\texttt{Spool} + \epsilon,$
	$\epsilon \sim$ Standard-Normalverteilung
Weilbull-Regression:	$\log(\texttt{Failure}) = \beta_o + \beta_1\texttt{Stress} + \beta_2\texttt{Spool} + \epsilon,$
	$\epsilon \sim$ Standard-Gumbel-Regression

dh, die beiden Regression unterscheiden sich in der Annahme der Fehlerverteilung.

e) Die Residuenplots
```
par(mfrow=c(2,2))
scatter.smooth(r.kevlar$linear.predictors, residuals(r.kevlar,
               type='deviance')[,"median"], xlab='linear predictor',
               ylab='deviance residual')
abline(h=0, lty=2)
scatter.smooth(d.kevlar$Stress, residuals(r.kevlar, type='deviance')[,"median"],
               xlab='Stress', ylab='deviance residual')
abline(h=0, lty=2)
scatter.smooth(d.kevlar$Spool, residuals(r.kevlar, type='deviance')[,"median"],
               xlab='Spool', ylab='deviance residual')
abline(h=0, lty=2)
```
Die



Glättungskurven sind bedeutend glätter als bei a).

3. a) Da wir am Mittelwert interessiert sind, benützen wir für die Regressionformel Y$\sim$1:
```
library(MASS)
d.ertrag <- c(35.6, 34.9, 36.0, 30.2, 36.2, 35.6, 35.8, 35.9, 36.1)

r.ertrag <- rlm(Y~1,data.frame(Y=d.ertrag))
```

```
(s.ertrag <- summary(r.ertrag))
s.mu <- s.ertrag$coefficients[1,1]
s.stdev <- s.ertrag$coefficients[1,2]
t.qt <- qt(0.975,length(d.ertrag)-1)
(r.intervall <- s.mu+c(-1,1)*t.qt*s.stdev)

>
Call: rlm(formula = Y ~ 1, data = data.frame(Y = d.ertrag))
Residuals:
     Min       1Q   Median       3Q      Max
-5.53414 -0.13414  0.06586  0.26586  0.46586

Coefficients:
            Value    Std. Error t value
(Intercept)  35.7341   0.1610   221.9895

Residual standard error: 0.3944 on 8 degrees of freedom


> [1] 35.36294 36.10535
```

Das klassische 95%-Konfidenzintervall ist viel grösser als das robuste Konfidenzintervall.

```
t.test(d.ertrag)$conf.int

33.68898 36.59990
```

b)

```
plot(d.ertrag)
abline(h=s.mu, lty=1)
abline(h=mean(d.ertrag), lty=2)
```