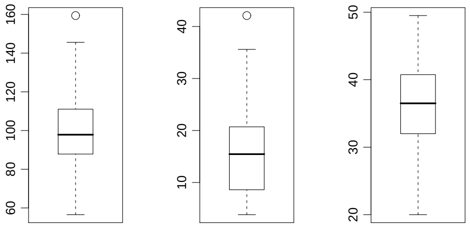
Weiterbildungs-Lehrgang in Angewandter Statistik ETHZ 2009/2011 5.10.07

Angewandte Regression — Musterlösungen zur Serie 3

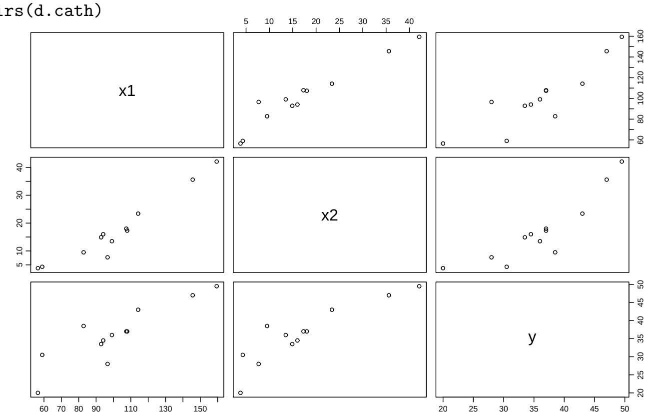
1. a) Die Variable x2 ist nicht symmetrisch. Zudem gibt es in x1 und x2 Ausreisser. R-Befehl:

```
par(mfrow=c(1,1))
boxplot(d.cath$x1, d.cath$x2, d.cath$y)
```



b) Es gibt zwischen allen Variablen einen starken linearen Zusammenhang. Insbesondere war die Abhängigkeit von x1 (Grösse) und x2 (Gewicht) zu erwarten.

```
pairs(d.cath)
```



c) Regression von y auf x1:

```
> r.cat.x1 <- lm(y ~ x1, d.cath)
> summary(r.cat.x1)

Call:
lm(formula = y ~ x1, data = d.cath)
```



```
Residuals:
    Min      1Q  Median      3Q     Max
-7.0929 -0.7298 -0.2608  1.1652  6.6879

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.12706    4.24700   2.855 0.017090 *
x1           0.23774    0.04034   5.893 0.000152 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.009 on 10 degrees of freedom
Multiple R-Squared: 0.7764,     Adjusted R-squared: 0.7541
F-statistic: 34.73 on 1 and 10 DF,  p-value: 0.0001525
```

Regression von y auf x2.

```
> r.cat.x2 <- lm(y ~ x2, d.cath)
> summary(r.cat.x2)

Call:
lm(formula = y ~ x2, data = d.cath)

Residuals:
    Min      1Q  Median      3Q     Max
-7.9676 -1.4963 -0.1386  2.0980  7.0205

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.62631    2.00264  12.796 1.59e-07 ***
x2           0.61613    0.09759   6.313 8.75e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.797 on 10 degrees of freedom
Multiple R-Squared: 0.7994,     Adjusted R-squared: 0.7794
F-statistic: 39.86 on 1 and 10 DF,  p-value: 8.755e-05
```

Modell	$\widehat{\sigma}^2$	R^2
$y = \alpha_1 + \beta_1 x^{(1)}$	16.07	0.776
$y = \alpha_2 + \beta_2 x^{(2)}$	14.42	0.799

Um die Modelle zu vergleichen, muss man z.B. $\widehat{\sigma}^2$ und R^2 betrachten. Es ist natürlich nicht sinnvoll, die Werte $\widehat{\alpha}_1, \widehat{\alpha}_2$ resp. $\widehat{\beta}_1, \widehat{\beta}_2$ miteinander zu vergleichen!

d) In beiden Modellen wird die Nullhypothese $H_0 : \beta = 0$ auf dem 5%-Niveau verworfen, denn die zugehörigen p-Werte sind < 0.05.

2. Anpassen des Modells $Y_i = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + E_i$ an die Daten in catheter.dat.

```
r.cat <- lm(y ~ x1 + x2, data=d.cath)
```

```
> summary(r.cat)

Call:
lm(formula = y ~ x1 + x2, data = d.cath)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.0497 -1.2753 -0.2595  1.9095  6.9933

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 21.08527    8.77037   2.404   0.0396 *
x1           0.07681    0.14412   0.533   0.6070
x2           0.42752    0.36810   1.161   0.2753
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.94 on 9 degrees of freedom
Multiple R-Squared: 0.8056,     Adjusted R-squared: 0.7624
F-statistic: 18.65 on 2 and 9 DF,  p-value: 0.0006301
```

a) Der F-Test testet die Nullhypothese $H_0 : \beta_1 = \beta_2 = 0$. Da die Teststatistik `F-statistic`$= 18.65$ einem p-Wert von < 0.001 entspricht, wird die Nullhypothese auf dem 5%-Niveau verworfen. Es können also nicht beide Variablen *gleichzeitig* aus dem Modell entfernt werden, obwohl beide für sich allein betrachtet p-Werte aufweisen, welche viel grösser als 0.05 sind!

b) Für den Test der Nullhypothese $H_0 : \beta_1 = 0$ gegen die Alternative $H_A : \beta_1 \neq 0$ ist die Teststatistik $\mathtt{T} = 0.533$ mit einem p-Wert von 0.607.
Für den Test der Nullhypothese $H_0 : \beta_2 = 0$ gegen die Alternative $H_A : \beta_2 \neq 0$ ist die Teststatistik $\mathtt{T} = 1.161$ mit einem p-Wert von 0.275.

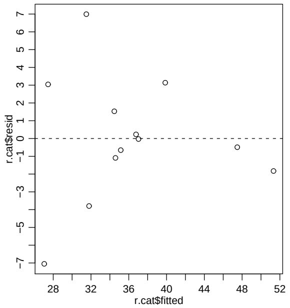
Keine der beiden Nullhypothesen kann auf dem 5%-Niveau verworfen werden.

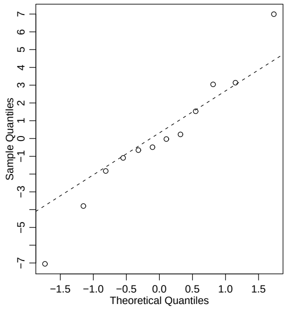
Bemerkung: Da `x1` und `x2` stark korreliert sind (siehe Aufgabe 1.b)), bewirkt das Hinzufügen von `x2`, dass `x1` nicht mehr signifikant im Modell vorkommt und umgekehrt.

c) Die drei Modelle unterscheiden sich kaum in den $\widehat{\sigma}^2$ und R^2-Werten. Man kann hier das Modell $Y_i = \beta_0 + \beta_1 \mathtt{x2}_i + E_i$ wählen, da R^2 nur unbedeutend schlechter als im vollen Modell und $\widehat{\sigma}^2$ am kleinsten ist. Dazu ist das Modell "einfacher", da weniger Parameter vorkommen.

Modell	$\widehat{\sigma}^2$	R^2
$\mathtt{y}_i = \beta_0 + \beta_1 \mathtt{x1}_i + E_i$	16.07	0.776
$\mathtt{y}_i = \beta_0 + \beta_1 \mathtt{x2}_i + E_i$	14.42	0.799
$\mathtt{y}_i = \beta_0 + \beta_1 \mathtt{x1}_i + \beta_2 \mathtt{x2}_i + E_i$	15.52	0.806

d) Der Normal Plot deutet auf eine langschwänzige Verteilung der Fehler hin. Es gibt zwei Residuen, deren Betrag sehr gross ist: 8. und 11. Beobachtung.





3. Für die Berechnung des Vorhersage-Intervalls $[V_0^*\langle x_0\rangle, V_0^*\langle x_1\rangle]$ benutzen wir die Formel aus dem Skript (siehe 2.4.e und 2.4.f):

$$[V_0^*\langle x_0\rangle, V_0^*\langle x_1\rangle] = [\ \hat{\eta} - q_{0.975}^{t_{n-p}} \sqrt{\hat{\sigma}^2 + (se^{(\hat{\eta})})^2}\ ,\ \hat{\eta} + q_{0.975}^{t_{n-p}} \sqrt{\hat{\sigma}^2 + (se^{(\hat{\eta})})^2}\ ]$$

wobei:

$$\begin{aligned} \hat{\eta} &= \hat{\beta}_0 + \hat{\beta}_1 \mathtt{x1}_0 + \hat{\beta}_2 \mathtt{x2}_0 \\ q_{0.975}^{t_{n-p}} &= 2.262 \qquad \textit{(mit R berechnet)} \\ \hat{\sigma}^2 &= 3.94^2 = 15.52 \qquad \textit{(siehe R-Output der Aufgabe 2)} \end{aligned}$$

Diese Formel ist in der Funktion `predict` bereits implementiert. R-Befehl:

```
r.cat <- lm(y ~ x1 + x2, d.cath) # Anpassen des Modells
round(predict(r.cat, interval = "prediction"), 1)
```

Berechnung "von Hand"

Den Standardfehler der Residuen $\hat{\sigma}$ erhalten wir aus der "Summary-Tabelle", das Quantil können wir mit `qt()` bestimmen, $\hat{\eta}$ und der Standardfehler $se^{(\hat{\eta})}$ berechnen wir mit der R-Funktion `predict()`.

R-Code für die Vorhersage-Intervalle:

```
## Vorhersage von eta und dessen Standardfehler
t.prog   <- predict(r.cat, se.fit=T)
t.eta    <- t.prog$fit
t.se.eta <- t.prog$se.fit

## Bestimmung von sigma
t.sigma  <- summary(r.cat)$sigma

## Berechnung des Vorhersage-Intervalles
t.qse   <- qt(0.975,9) * sqrt(t.sigma^2 + t.se.eta^2)
t.unten <- t.eta - t.qse
t.oben  <- t.eta + t.qse
round(cbind(t.unten,t.eta,t.oben),1)
```

R-Output:

```
   t.unten t.eta t.oben
1     27.7  37.0   46.4
2     40.4  51.3   62.3
3     25.8  35.2   44.5
4     24.9  34.5   44.0
5     30.4  39.9   49.3
6     20.5  31.8   43.1
7     27.3  36.8   46.3
8     16.5  27.0   37.6
9     25.3  34.6   43.9
10    17.1  27.5   37.8
11    22.1  31.5   41.0
12    37.3  47.5   57.7
```

Alle Intervalle sind viel grösser als ±2 cm. Das bedeutet, dass unser Modell unbrauchbar ist. Da das volle Modell nicht genügt, tun es auch die Teilmodelle nicht.

Fazit: Das Modell muss verbessert werden, insbesondere durch mehr Beobachtungen (verkleinert $se^{(\hat{\eta})}$) oder neue erklärende Variablen (verkleinert $\widehat{\sigma}^2$).

4. a) Die Ladung hat einen signifikanten Einfluss. Ein 95%-Vertrauensintervall für den Koeffizienten β_2 ist $[0.641, 1.56]$.
R-Code:

```
# Daten Einlesen
d.spreng <-read.table("http://stat.ethz.ch/Teaching/Datasets/NDK/spreng.dat",
                       header=T)
# Struktur anschauen
str(d.spreng)
# Faktoren definieren
d.spreng[,"stelle"] <- factor(d.spreng[,"stelle"])
# Regression
t.lm <- lm(log10(ersch)~log10(dist)+log10(ladung)+stelle, data=d.spreng)
summary(t.lm)
```

R-Output:

```
Call:
lm(formula = log10(ersch) ~ log10(dist) + log10(ladung) + stelle,
    data = d.spreng)

Residuals:
      Min        1Q    Median        3Q       Max
-0.383070 -0.091936 -0.009373  0.088866  0.393094



Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.20568    0.24163   9.128 2.23e-14 ***
log10(dist)   -1.27025    0.12237 -10.381  < 2e-16 ***
log10(ladung)  1.10002    0.23121   4.758 7.62e-06 ***
stelle2        0.18108    0.07675   2.359   0.0205 *
stelle3        0.03668    0.06659   0.551   0.5831
stelle4        0.12743    0.07663   1.663   0.0999 .
stelle5       -0.11905    0.05483  -2.171   0.0326 *
stelle6        0.30141    0.05812   5.186 1.36e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1609 on 88 degrees of freedom
Multiple R-Squared: 0.8018, Adjusted R-squared: 0.786
F-statistic: 50.86 on 7 and 88 DF,  p-value: < 2.2e-16
```

95%-Vertrauensintervall:

```
confint(t.lm)
```

R-Output:

```
                     2.5 %      97.5 %
(Intercept)    1.72549115  2.68586312
log10(dist)   -1.51343471 -1.02707503
log10(ladung)  0.64054122  1.55950533
stelle2        0.02854504  0.33361037
stelle3       -0.09565493  0.16902151
stelle4       -0.02484878  0.27970544
stelle5       -0.22801323 -0.01007958
stelle6        0.18590438  0.41690634
```


b) Die Formeln für *signif* und *stcoef* findet man im Skript, 3.1.l und 3.1.m.

$$signif_j = \frac{T_j}{q_{0.975}^{t_k}} = \frac{\hat{\beta}_j}{\mathrm{se}^{\beta_j} q_{0.975}^{t_k}}$$

$$stcoef_j = \hat{\beta}_j \frac{sd(X^{(j)})}{sd(Y)}$$

R-Code:

```
t.lm.coeff.rest <- summary(t.lm)$coefficients[2:3,]
t.df <- summary(t.lm)$df[2]

t.tratio <- t.lm.coeff.rest[,1]/(t.lm.coeff.rest[,2]*qt(0.975,df=t.df))
t.tratio <- t.lm.coeff.rest[,3]/(qt(0.975,df=t.df))

t.var <- c(var(log(d.spreng$dist)),var(log(d.spreng$ladung)))
t.signif <- t.lm.coeff.rest[1:2,1]*sqrt(t.var/var(log(d.spreng$ersch)))

cbind(t.tratio,t.signif)
```

R-Output:

```
              t.tratio   t.signif
log10(dist)   -5.223520 -0.6338627
log10(ladung)  2.394051  0.2326299
```

c) R-Code:

```
# Regression
t.r <- regr(log10(ersch)~stelle+log10(dist)+log10(ladung), data=d.spreng)
t.r
```

R-Output:

```
Call:
regr(formula = log10(ersch) ~ stelle + log10(dist) + log10(ladung),
    data = d.spreng)

Terms:
                   coef     stcoef    signif       R2.x df p.value
(Intercept)    2.205677  0.0000000  4.593381         NA  1       0
stelle               NA         NA  6.571966 0.04654250  5       0
log10(dist)   -1.270255 -0.6338627 -5.223520 0.22281471  1       0
```

```
log10(ladung)  1.100023  0.2326299  2.394051 0.02943579  1       0



Coefficients for factors:
$stelle
          1           2           3           4           5           6
 0.00000000  0.18107770  0.03668329  0.12742833 -0.11904641  0.30140536


St.dev.error:  0.1609   on 88 degrees of freedom
Multiple R^2:  0.8018  Adjusted R-squared: 0.786
F-statistic:   50.86   on 7 and 88 d.f.,  p.value:     0
```

95%-Vertrauensintervall:

```
confint(t.r)
```

R-Output:

```
confint(t.r)
                    2.5 %      97.5 %
(Intercept)    1.72549115  2.68586312
stelle2        0.02854504  0.33361037
stelle3       -0.09565493  0.16902151
stelle4       -0.02484878  0.27970544
stelle5       -0.22801323 -0.01007958
stelle6        0.18590438  0.41690634
log10(dist)   -1.51343471 -1.02707503
log10(ladung)  0.64054122  1.55950533
```

Bei Variablen mit nur einem Freiheitsgrad, zeigt die Spalte **signif** im Output der `regr` Funktion den *t-ratio* $\tilde{T}_j = T/q_{0.975}^{t_k}$. Die Null Hypothese $\beta_j = 0$ wird verworfen, wenn der t-ratio grösser als 1 ist. Ein 95%-Vertrauensintervall für β_j ist $\hat{\beta}_j \cdot (1 \pm 1/\tilde{T}_j)$. Für Faktoren und andere Variablen wird ein F-Test durchgeführt. 95%-Vertrauensintervall von Hand:

```
conf.int <- cbind(t.r$coefficients[8]*(1-1/(t.r$testcoef$signif[4])),
                  t.r$coefficients[8]*(1+1/(t.r$testcoef$signif[4])))
```

R-Output:

```
                     [,1]     [,2]
log10(ladung) 0.6405412 1.559505
```

d) Mit der Funktion `regr` kann man das Ergebnis schon im R-Output in der Spalte `p-value` ablesen. Fuer den Faktor `stelle` ist der F-Test signifikant (der P-Wert ist gleich 0). Mit der Funktion `lm` braucht man die Funktion `drop1(t.lm,test="F")`. Man bekommt das gleiche Resultat.

e) Wir schätzen also ein Modell mit der Interaktion zwischen den Faktor `stelle` und der Variable `log10(dist)`. Die Interaktion ist signifikant (P-Wert=0.0221). Die Effekte der Variable `log10(dist)` und des Faktors `stelle` sind nicht additiv, d.h. für die verschiedenen Stellen ist der Effekt der Variable `log10(dist)` auf die Zielvariable `log10(ersch)` verschieden.
R-Code:

```
# mit lm
t.lmi<-lm(log10(ersch)~log10(dist)*stelle+log10(ladung),data=d.spreng)
drop1(t.lmi,test="F")
```

R-Output:

```
Single term deletions

Model:
log10(ersch) ~ log10(dist) * stelle + log10(ladung)

                  Df Sum of Sq     RSS     AIC F value    Pr(F)
<none>                            1.95 -348.09
log10(ladung)      1      0.57    2.52 -325.60 24.1173 4.48e-06 ***
log10(dist):stelle  5      0.33    2.28 -343.16  2.7937  0.02211 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R-Code:

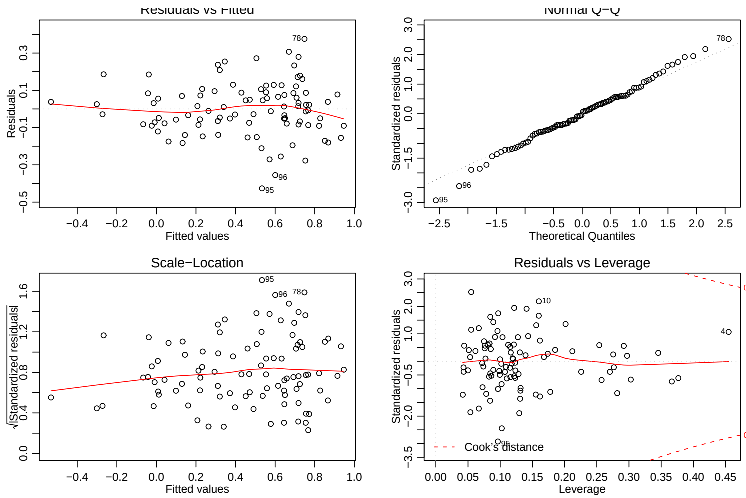
```
# mit regr
t.ri<-regr(log10(ersch)~log10(dist)*stelle+log10(ladung),data=d.spreng)
t.ri
```

R-Output:

```
Call:
regr(formula = log10(ersch) ~ log10(dist) * stelle + log10(ladung),
    data = d.spreng)

Terms:
                        coef     stcoef    signif       R2.x df p.value
(Intercept)         3.619701  0.0000000  3.735213         NA  1  0.0000
log10(dist)        -1.994272 -0.9951506 -4.034218 0.63548337  1  0.0000
stelle                    NA         NA  1.790195 0.94481436  5  0.0367
log10(ladung)       1.131367  0.2392584  2.469097 0.07206478  1  0.0000
log10(dist):stelle        NA         NA  2.012014 0.94416161  5  0.0221

Coefficients for factors:
$stelle
        1         2         3         4         5         6
 0.000000  0.655570 -1.856983 -1.101441 -1.570463 -2.142267
```

```
$`log10(dist):stelle`
1 2 3 4 5 6
0 0 0 0 0 0


St.dev.error:  0.1533   on 83 degrees of freedom
Multiple R^2:  0.8304  Adjusted R-squared: 0.8058
F-statistic:   33.86   on 12 and 83 d.f.,  p.value:     0
```

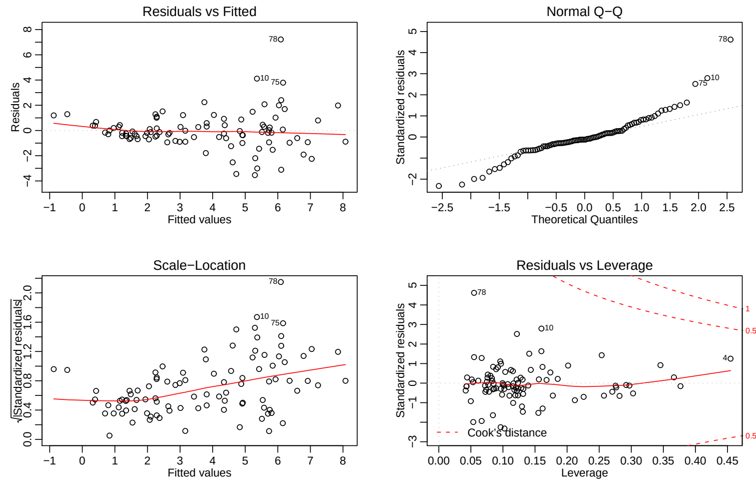
f) Der Tukey-Anscombe-Plot weist eine kleine Trichterform aus, d.h. die Annahme gleicher Varianzen für die Fehler könnte in Frage gestellt werden. Der Normalplot sieht gut aus.



R-Code:

```
#mit lm
par(mfrow=c(2,2))
plot(t.lmi)
# mit regr
plot(t.ri)
```

g) Mit der unlogarithmierten Zielgrösse `erch` weist der Tukey-Anscombe-Plot eine klare Tricherform auf, d.h. hier ist die Annahme gleicher Varianzen für die Fehler klar nicht plausibel. Der Normal Plot deutet auf eine langschwänzige Verteilung hin.



R-Code:

```
# mit lm
t.lmi2<-lm(ersch~log10(dist)*stelle+log10(ladung),data=d.spreng)
plot(t.lmi2)
# mit regr
t.ri2<-regr(ersch~stelle*log10(dist)+log10(ladung),data=d.spreng)
plot(t.ri2)
```