```
> #Statistik Uebung 10

# Plots in Datei schreiben
postscript(file = "./serie_10_plot.ps")
> #Aufgabe 1)

#a)
strasse ←read.table("http://stat.ethz.ch/Teaching/Datasets/strasse.dat", header=T)
par(mfrow=c(2,1))
plot(strasse$verkehr, strasse$blei, main="Streudiagramm")
#Verkehrsfluss ist diskret.

#b)
strasse.fit←lm(blei ~ verkehr, data=strasse)
abline(coef(strasse.fit))
summary(strasse.fit)

#c)
#signifikante Korrelation (Beta1!=0)
#86% Wahrscheinlichkeit für Beta0=0

#d)
plot(fitted(strasse.fit), resid(strasse.fit), main="Tukey−Anscombe Plot")
abline(h=0, lty=2)
#Residuen sind zufällig verteilt => gut!

#e)
beta←as.vector(coefficients(strasse.fit))
bleigehalt←beta[1]+beta[2]*40
bleigehalt


Call:
lm(formula = blei ~ verkehr, data = strasse)

Residuals:
    Min      1Q  Median      3Q     Max
-128.43  -63.13   24.52   69.32  125.72

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -12.842     72.143  -0.178    0.863
verkehr       36.184      3.693   9.798 4.24e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 92.19 on 9 degrees of freedom
Multiple R-Squared: 0.9143,     Adjusted R-squared: 0.9048
F-statistic: 96.01 on 1 and 9 DF,  p-value: 4.239e-06

[1] 1434.512
> # Aufgabe 2)

# Statistik Uebung 10

# Aufgabe 2)

# a)
forbes←read.table("http://stat.ethz.ch/Teaching/Datasets/forbes.dat", header=T)
par(mfrow=c(3,1))
plot(forbes$Temp, forbes$Press, main="Streudiagramm")
# Es handelt sich sicher nicht um eine Unsinnkorrelation
```



```
# das Streudiagramm verlangt nach linearer Regression
# Druck -> Log?

# b)
forbes.fit←lm(Press ~ Temp, data=forbes)
summary(forbes.fit)
abline(coef(forbes.fit))

# c)
plot(fitted(forbes.fit), resid(forbes.fit), main="Tukey−Anscombe Plot")
abline(h=0)
qqnorm(resid(forbes.fit))
# der QQ-plot sieht auch nicht schlecht aus

# d)
forbes$LogPress ← log(forbes$Press)
plot(forbes$Temp, forbes$LogPress, main="Logarithmisches Streudiagramm")
forbesLog.fit←lm(LogPress ~ Temp, data=forbes)
summary(forbesLog.fit)
abline(coef(forbesLog.fit))

Call:
lm(formula = Press ~ Temp, data = forbes)

Residuals:
     Min       1Q   Median       3Q      Max
-0.25717 -0.11246 -0.05102  0.14283  0.64994

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -81.06373    2.05182  -39.51   <2e-16 ***
Temp          0.52289    0.01011   51.74   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2328 on 15 degrees of freedom
Multiple R-Squared: 0.9944,     Adjusted R-squared: 0.9941
F-statistic:  2677 on 1 and 15 DF,  p-value: < 2.2e-16


Call:
lm(formula = LogPress ~ Temp, data = forbes)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0073622 -0.0033863 -0.0015865  0.0004322  0.0313139

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.9708662  0.0769377  -12.62 2.17e-09 ***
Temp         0.0206224  0.0003789   54.42  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00873 on 15 degrees of freedom
Multiple R-Squared: 0.995,      Adjusted R-squared: 0.9946
F-statistic:  2962 on 1 and 15 DF,  p-value: < 2.2e-16

> # e)
plot(fitted(forbesLog.fit), resid(forbesLog.fit), main="Tukey−Anscombe Plot")
abline(h=0)
# ein Ausreisser sichtbar
```

```
qqnorm(resid(forbesLog.fit))
# ein Ausreisser sichtbar

> # f)
# Ausreisser suchen
identify(fitted(forbesLog.fit), resid(forbesLog.fit))
# ergibt nur numeric(0) (unter Linux mit Tcl/Tk interface)?
fitted(forbesLog.fit)
resid(forbesLog.fit)

numeric(0)
       1        2        3        4        5        6        7        8
3.040183 3.036059 3.110299 3.120610 3.141233 3.151544 3.172166 3.176291
       9       10       11       12       13       14       15       16
3.182477 3.180415 3.227847 3.248469 3.349518 3.330958 3.374265 3.399012
      17
3.405199
            1             2             3             4             5
-5.710930e-03 -1.586458e-03 -1.238101e-03  4.322241e-04  7.621790e-04
            6             7             8             9            10
-9.467980e-04  1.293818e-03  1.346461e-03 -3.590507e-03 -1.944677e-03
           11            12            13            14            15
-3.386315e-03  3.131388e-02  3.469950e-05 -7.362188e-03 -5.591092e-03
           16            17
-1.822755e-03 -2.003440e-03
> # Wert an Stelle 12 ist es, diesen entfernen
plot(forbes$Temp[-12], forbes$Press[-12], main="Streudiagramm")

forbes.fit<-lm(Press[-12] ~ Temp[-12], data=forbes)
summary(forbes.fit)
abline(coef(forbes.fit))

plot(fitted(forbes.fit), resid(forbes.fit), main="Tukey-Anscombe Plot")
abline(h=0)
qqnorm(resid(forbes.fit))

# logarithmisch
plot(forbes$Temp[-12], forbes$LogPress[-12], main="Logarithmisches Streudiagramm")
forbesLog.fit<-lm(LogPress[-12] ~ Temp[-12], data=forbes)
summary(forbesLog.fit)
# R^2 ist besser als bei der nicht logarithmischen Regression
abline(coef(forbesLog.fit))

plot(fitted(forbesLog.fit), resid(forbesLog.fit), main="Tukey-Anscombe Plot")
abline(h=0)
qqnorm(resid(forbesLog.fit))

qqnorm(apply(forbes, 1, "mean"))
# unsere Idee um eine bessere Antwort auf Aufgabe a) zu haben

Call:
lm(formula = Press[-12] ~ Temp[-12], data = forbes)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21493 -0.09546 -0.01500  0.09049  0.27793

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -80.667294   1.419984  -56.81   <2e-16 ***
Temp[-12]     0.520738   0.006997   74.42   <2e-16 ***
---
```



```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1608 on 14 degrees of freedom
Multiple R-Squared: 0.9975,     Adjusted R-squared: 0.9973
F-statistic:  5538 on 1 and 14 DF,  p-value: < 2.2e-16


Call:
lm(formula = LogPress[-12] ~ Temp[-12], data = forbes)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0048082 -0.0014595  0.0004546  0.0020358  0.0031219

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.9517662  0.0231021   -41.2 5.16e-16 ***
Temp[-12]    0.0205186  0.0001138   180.2  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.002616 on 14 degrees of freedom
Multiple R-Squared: 0.9996,     Adjusted R-squared: 0.9995
F-statistic: 3.249e+04 on 1 and 14 DF,  p-value: < 2.2e-16

> # Postscript-File schliessen
dev.off()
null device
          1
> # Aufgabe 3
# 1) & 2)
# beide Aussagen für sich alleine stimmen nicht. Viel Kühe und/oder Fläche aber
ergibt viel Einkommen, dies ist signifikant: P_Value=4.47E-05<0.05, also bezieht
 sich der P_Value-Wert auf beta_1 und beta_2 (da bei cows "..")?

# 3)
# 1) & 2) => verwerfen (wie sehen die Resultate aus falls für z.B. beta_1 0 gesc
hätzt wird?, dann könnte man H_0: beta_1=0 nicht verwerfen, jedoch H_0: beta_2=0
 schon).

# 4)
# 17 (20-3)

# 5)
# \bar{x} \pm sigma/sqrt(n-freiheitsgrade)
# Estimate_cows \pm ( t_17,0.975 / sqrt(17) ) * {Std. Error}_cows
# 32.5690 \pm ( 2.11 / sqrt(17) ) * 3.7276 => keines der Intervalle

# 6)
# 285.4572 + 0*32.5690 + 2.1384*100 = 499.297 ~ 499 => 498

# 7)
# nicht ohne weiteres klar. Wahrscheinlich aber nicht signifikant, da bei Berück
sichtigung beider Variablen das Ergebniss nur sehr knapp signifikant ist (4.47E-
05)

>
```