Aufgabenblatt 1

UK Erweiterungen des linearen Modells

Cordula Eggerth

Matrikelnummer: 00750881

Kursleiter:

Prof. Dr. Marcus Hudec &

Prof. Dr. Wilfried Grossmann

Sommersemester 2019

Aufgabe 1:

Führen Sie mit dem Datensatz *mtcars* verschiedene sinnvolle regressionsanalytische Auswertungen und Visualisierungen durch.

Als Vorlage können Ihnen die Auswertungen und R-Codes aus dem Kapitel 01 dienen.

Der mtcars Datensatz stammt aus dem Motor Trend US Magazin und bietet Informationen über den Treibstoffverbrauch, die Anzahl an Zylindern und weiteren Messzahlen bezüglich Kraftfahrzeugdesign und -leistung für Automodelle (aus den Jahren 1973 und 1974).[1] Der Datensatz enthält 32 Beobachtungen.

Die Abkürzungen in den 11 Dataframe-Spalten beziehen sich auf die genannten Eigenschaften der Automodelle: [2]

- mpg … Miles per US Gallon
- cyl … Nr. of Cylinders
- disp … Displacement (in cubic inch)
- hp … Gross Horsepower
- drat … Rear Axle Ratio
- wt … Weight (1000 lbs)
- qsec … ¼ Mile Time
- vs … Engine (0=V-shaped; 1=Straight)
- am … Transmission (0=Automatic; 1=Manual)
- gear … Nr. of Forward Gears

Der Datensatz mtcars:

```
> mtcars
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
```

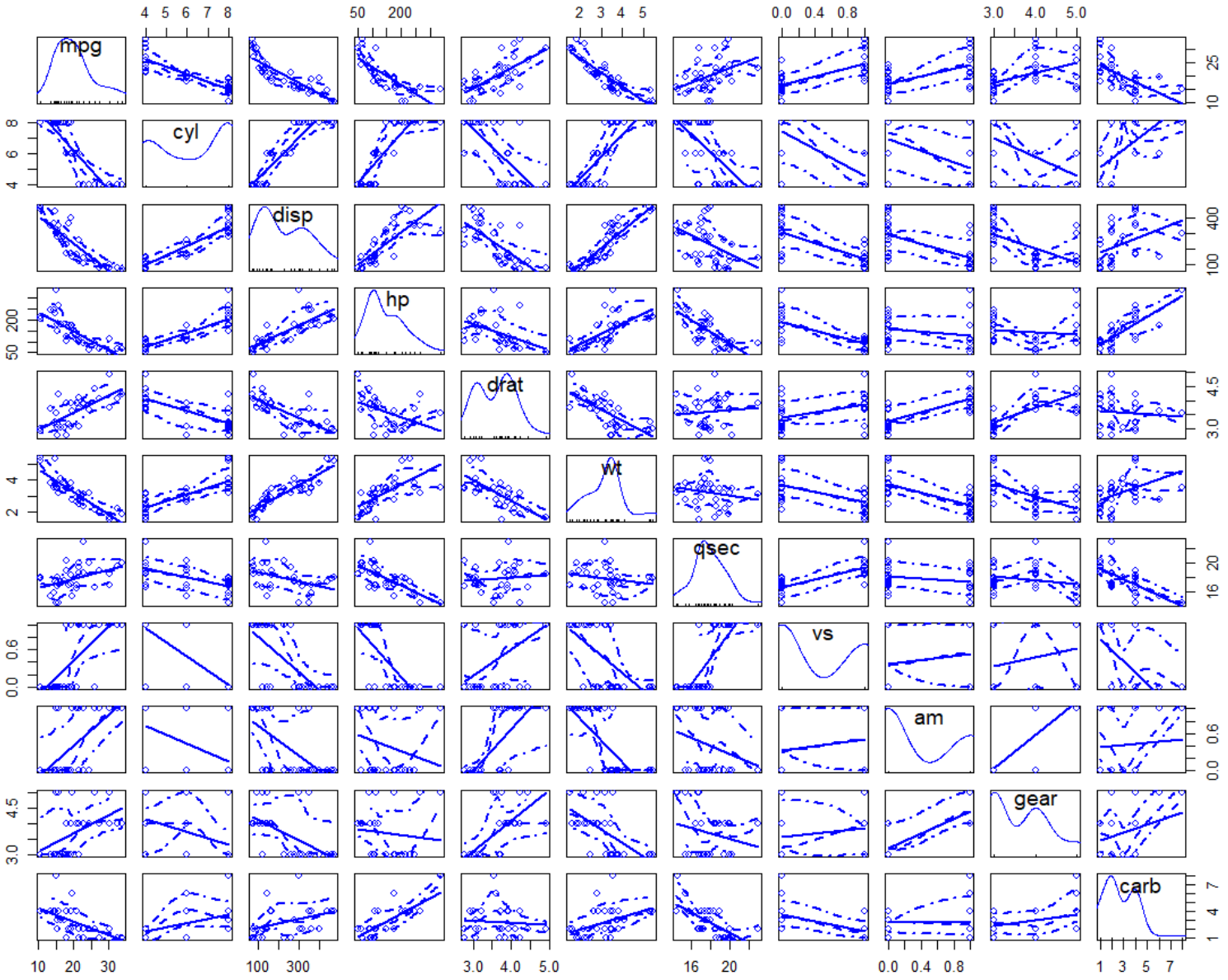
[1] Quelle: https://www.rdocumentation.org/packages/datasets/versions/3.5.3/topics/mtcars.
[2] Quelle: https://www.rdocumentation.org/packages/datasets/versions/3.5.3/topics/mtcars.

Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Zusammenfassung der deskriptiven Statistiken zu mtcars:

```
> summary(mtcars) # deskriptive zusammenfassung der daten
      mpg             cyl             disp             hp             drat             wt       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760   Min.   :1.513  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695   Median :3.325  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597   Mean   :3.217  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930   Max.   :5.424  
      qsec             vs              am              gear            carb      
 Min.   :14.50   Min.   :0.0000   Min.   :0.0000   Min.   :3.000   Min.   :1.000  
 1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
 Median :17.71   Median :0.0000   Median :0.0000   Median :4.000   Median :2.000  
 Mean   :17.85   Mean   :0.4375   Mean   :0.4062   Mean   :3.688   Mean   :2.812  
 3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
 Max.   :22.90   Max.   :1.0000   Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

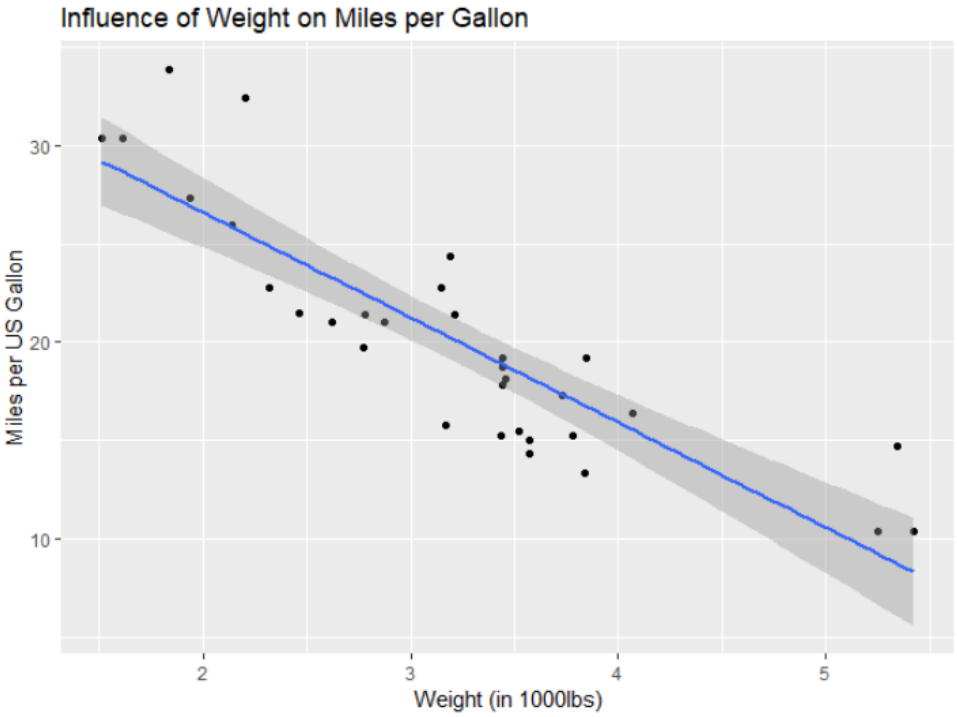
Scatterplotmatrix aller Variablen:



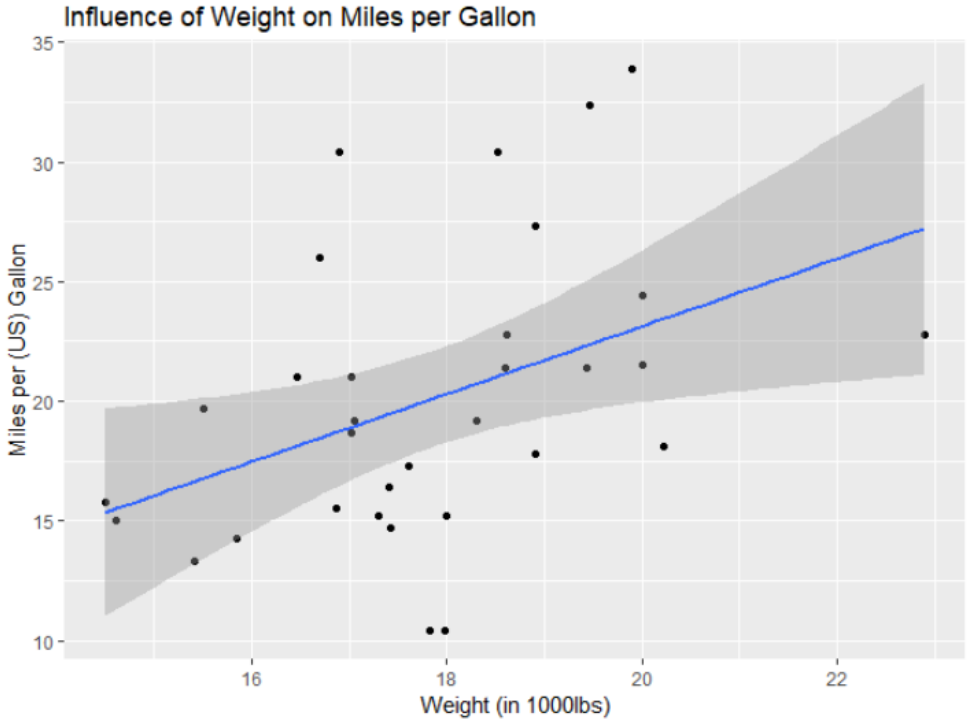
Auf den Datensatz mtcars wurden sodann verschiedene regressionsanalytische Verfahren angewandt und Plots angefertigt, die in weiterer Folge vorgestellt werden:

1.a. LINEAR REGRESSION (univariat):

Man erkläre mpg (miles per US gallon) durch die variable wt (weight in 1000lbs), wobei für die Visualisierung ggplot verwendet wurde. Der Plot zeigt in blau die Regressionsgerade und in dunkelgrau um die Regressionsgerade die Konfidenzbänder.



Man erkläre mpg durch qsec (i.e. ¼ mile time), wobei man sieht, dass die Konfidenzbänder der untersuchten Variable sehr weit sind, und die Daten (im Scatterplot) nicht linear sind:



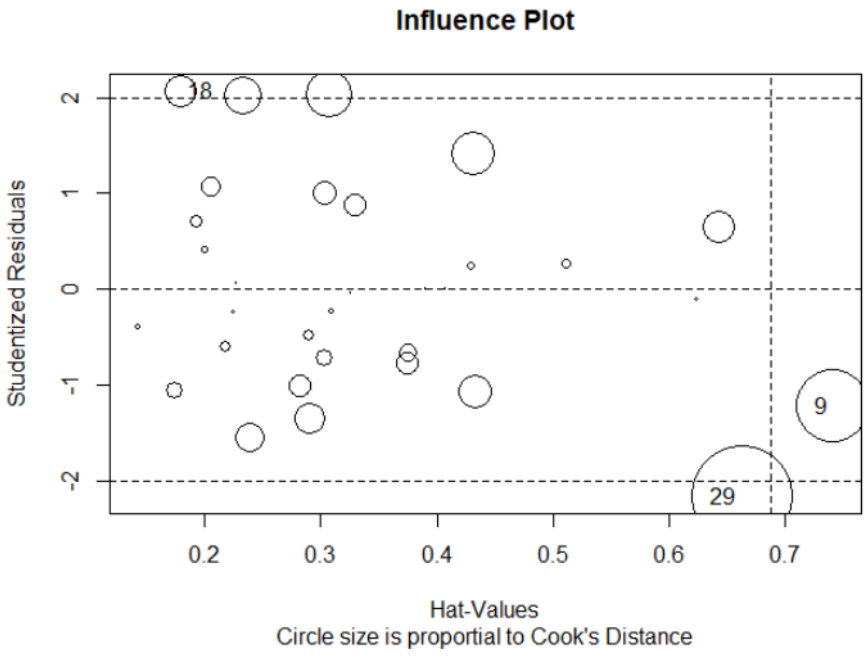
1.b. LINEAR REGRESSION (multiple):

Bei der multiplen linearen Regression wird in diesem Beispiel mpg durch die restlichen Variablen in Form eines additiven Zusammenhangs erklärt. Signifikant ist hier nur wt (weight

in 1000lbs) und R^2 (i.e. erklärter Anteil der Gesamtvariabilität) liegt bei ca. 87% in diesem Modell.

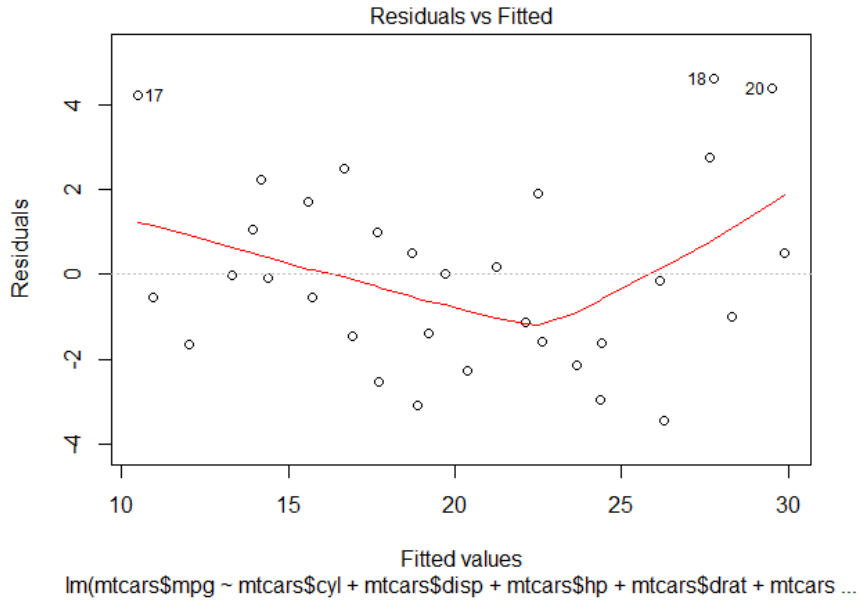
```
> summary(multiple_lm)

Call:
lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
    am + gear + carb)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4506 -1.6044 -0.1196  1.2193  4.6271

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.30337   18.71788   0.657   0.5181
cyl         -0.11144    1.04502  -0.107   0.9161
disp         0.01334    0.01786   0.747   0.4635
hp          -0.02148    0.02177  -0.987   0.3350
drat         0.78711    1.63537   0.481   0.6353
wt          -3.71530    1.89441  -1.961   0.0633 .
qsec         0.82104    0.73084   1.123   0.2739
vs           0.31776    2.10451   0.151   0.8814
am           2.52023    2.05665   1.225   0.2340
gear         0.65541    1.49326   0.439   0.6652
carb        -0.19942    0.82875  -0.241   0.8122
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.65 on 21 degrees of freedom
Multiple R-squared:  0.869,     Adjusted R-squared:  0.8066
F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

Der Influence-Plot zeigt, dass einzelne Punkte, wie z.B. 8, 9 oder 29 einen großen Einfluss auf die Residuen haben:



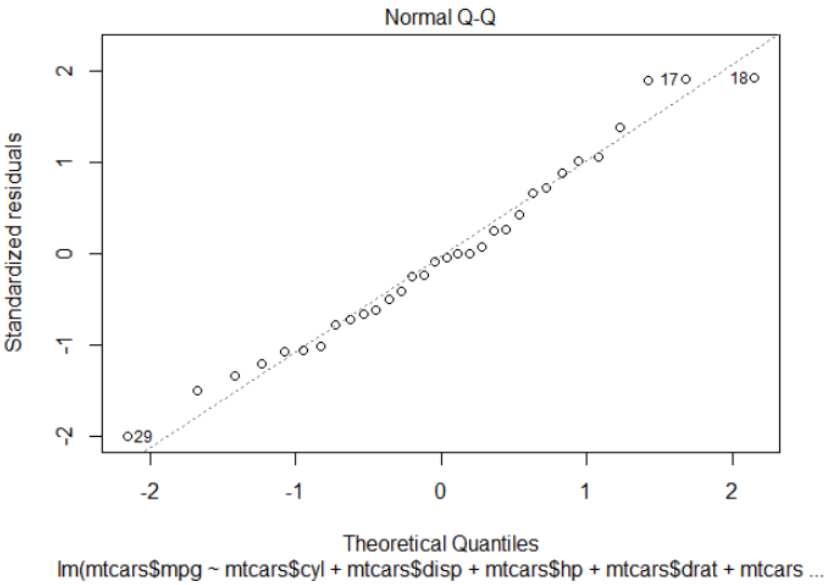
Residuen vs. gefittete Werte:

In diesem Beispiel sind die Residuen zufällig um die Nulllinie verteilt, daher ist keine starke Abweichung von der Linearität anzunehmen.



Q-Q-Plot:

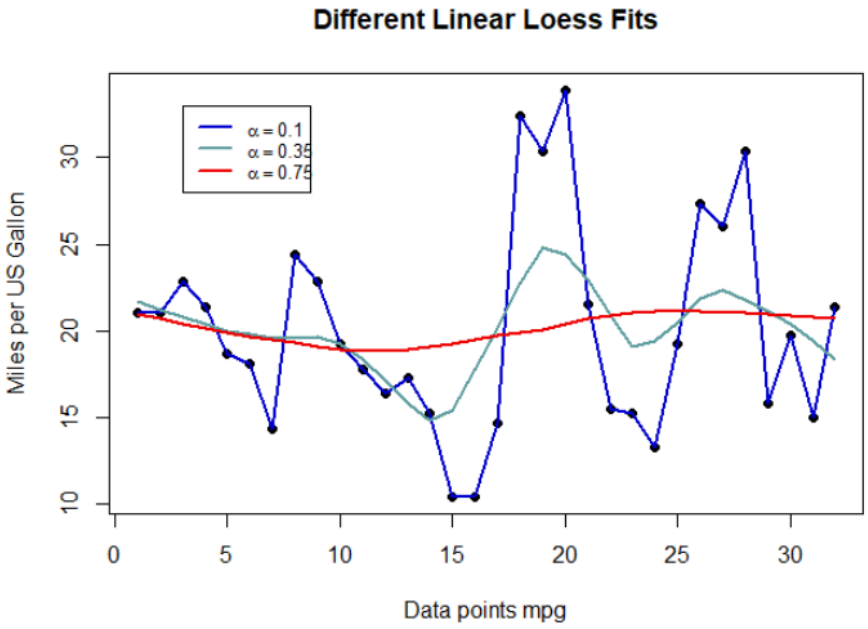
Der Q-Q-Plot vergleicht die theoretischen Quantile mit jenen, die in den Daten beobachtet wurden. Wenn die Punkte auf der Gerade der theoretischen Quantile liegen, sind die Residuen normalverteilt – im Plot weichen die Enden von der Gerade relativ stark ab, daher ist wohl keine exakte Normalverteilung der Residuen in den Daten vorhanden.



2.a. LOCAL REGRESSION:

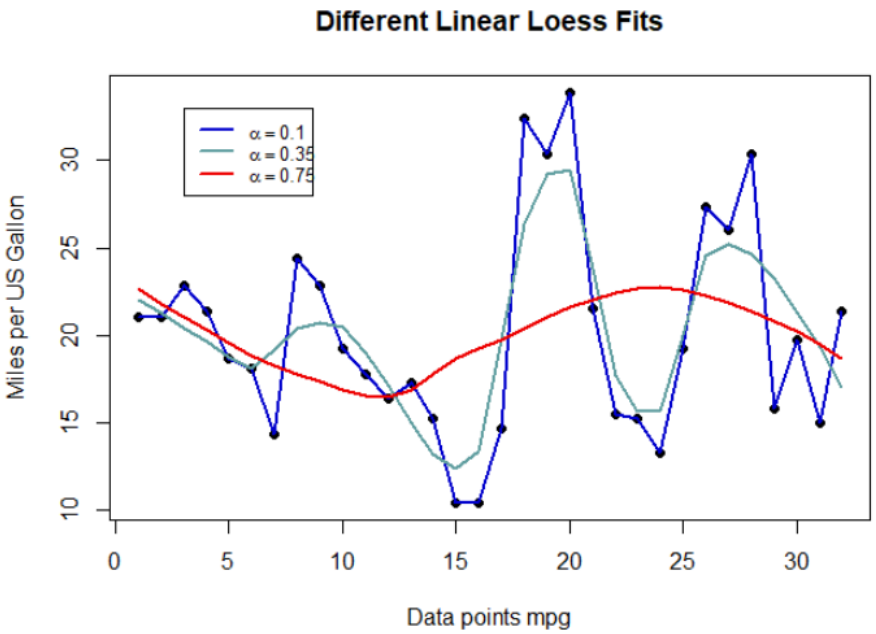
LINEAR LOESS FIT (Polynome 1. Grades):

Die lokale polynomial Regression bzw. LOESS ist ein Verfahren, bei der lokal eine bestimmte Nachbarschaft um den jeweiligen untersuchten Punkt in die Regression miteinbezogen wird. Die Größe der inkludierten Nachbarschaft richtet sich nach dem Parameter alpha, der der Spannweite-Parameter ist. Auf die Nachbarschaft wird außerdem eine Gewichtung angewandt, je nachdem wie weiter die Punkte vom untersuchten Punkt entfernt sind. Im untenstehenden Plot werden Polynome 1. Grades, also Geraden zur Modellierung verwendet.

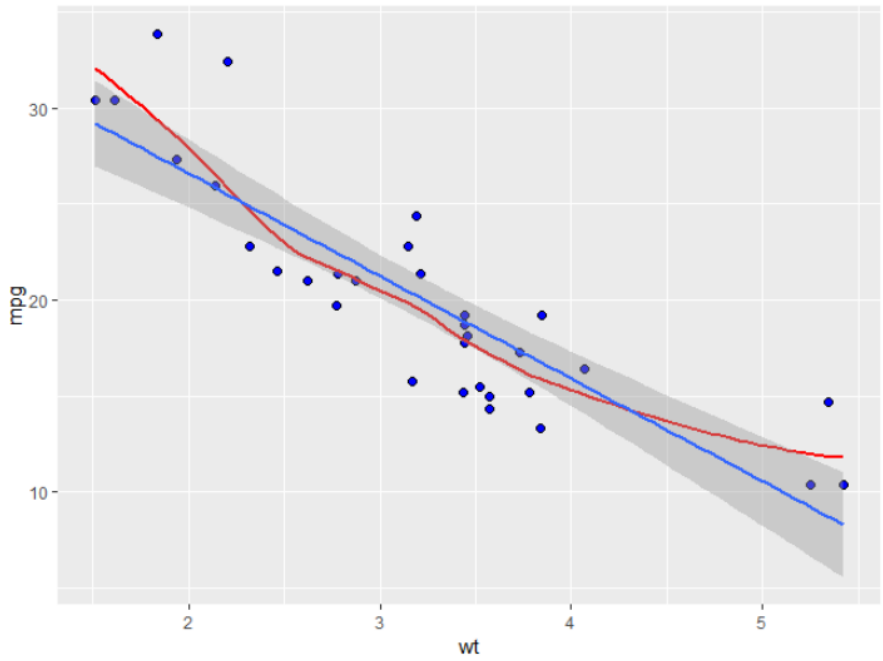


QUADRATIC LOESS FIT (Polynome 2. Grades):

Im folgenden Plot warden Polynome 2. Grades verwendet, wodurch die Kurven glatter aussehen.

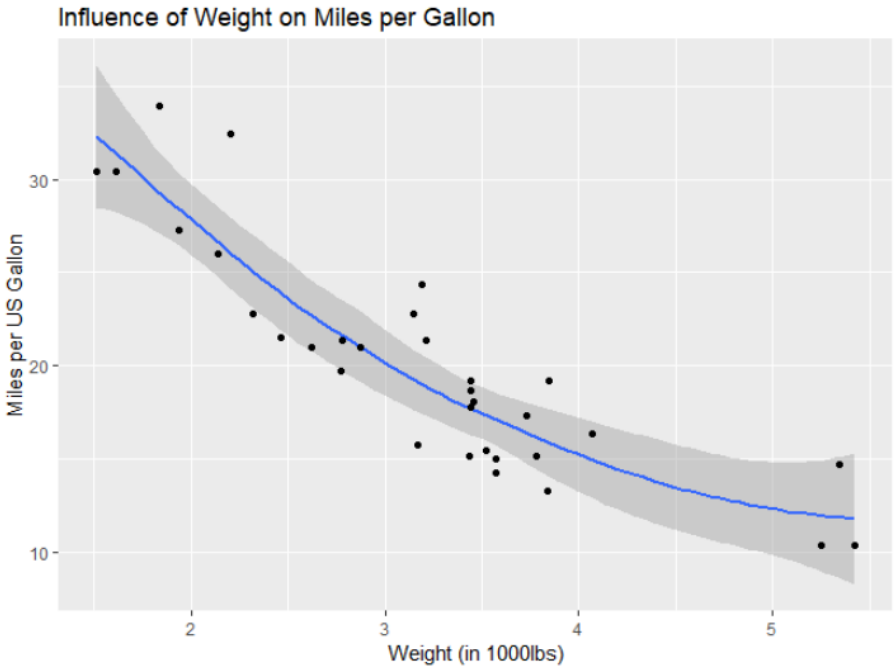


Lineare Regression (blaue Linie) vs. LOESS (rote Linie):

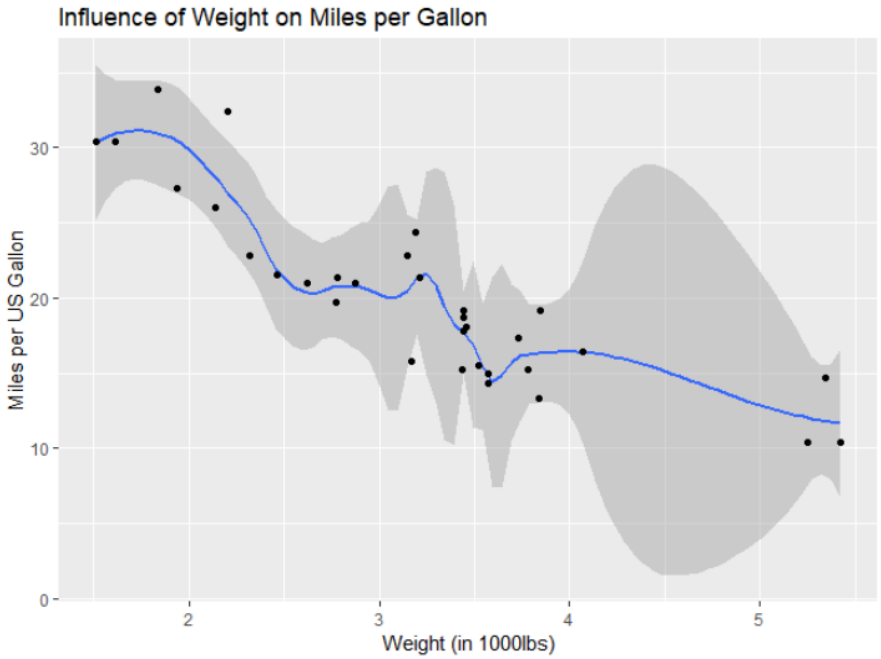


2.b. LOESS bzw. LOCAL POLYNOMIAL REGRESSION ("smoothing", univariat, mittels ggplot library)

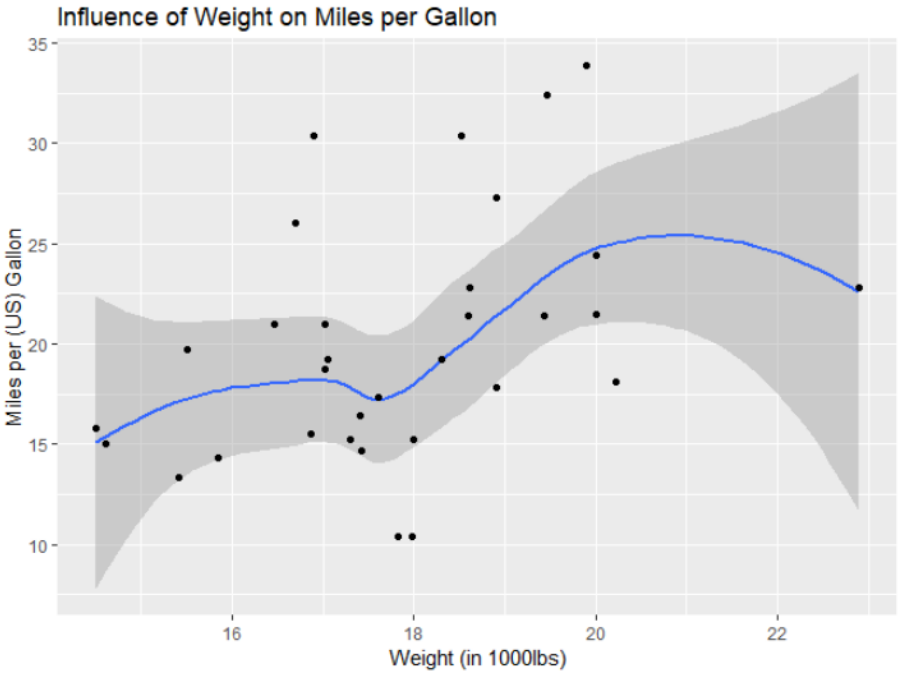
Man erkläre mpg durch wt mit einem Span-Parameter von 0.9, also unter Einbeziehung von 90% der Punkte. Wenn eine größere Anzahl von Punkten in der Umgebung des betrachteten Punktes miteinbezogen wird, erhält man, wie hier, eine glattere Kurve.



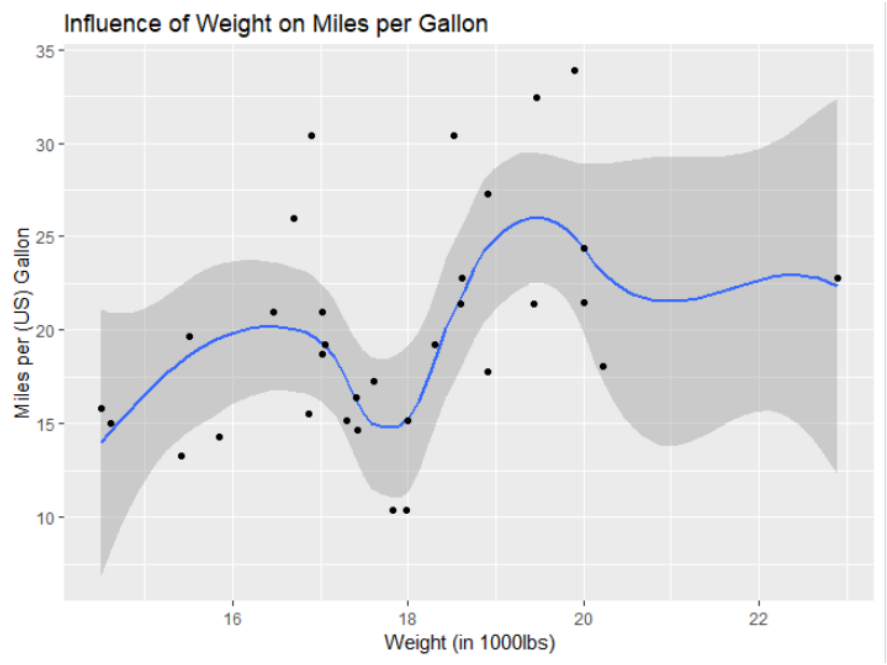
Bei einem Span-Parameter von nur 30% werden weniger Punkte in der Umgebung miteinbezogen und die Berechnung ist weniger glatt, wie es das untenstehende Bild zeigt.



Erklärung von mpg durch qsec mit Span von 0.9:

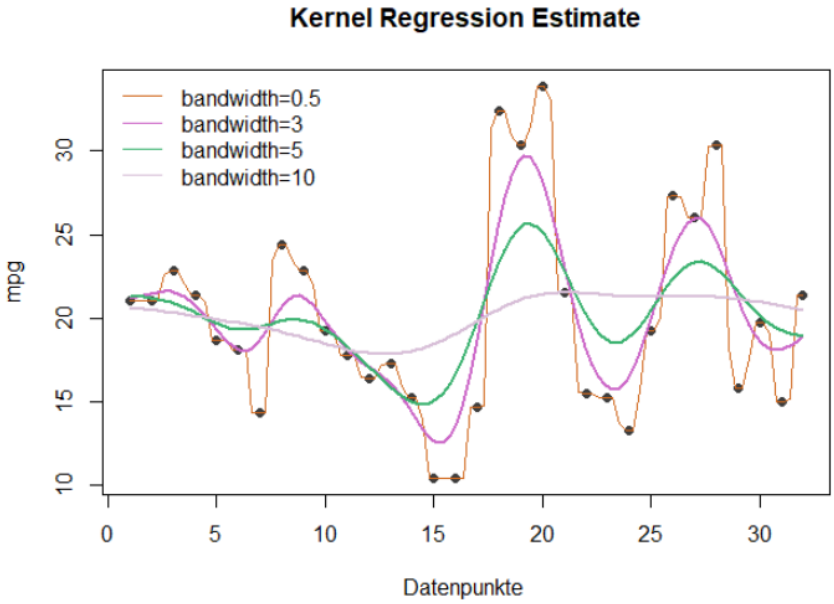


Erklärung von mpg durch qsec mit Span von 0.7:

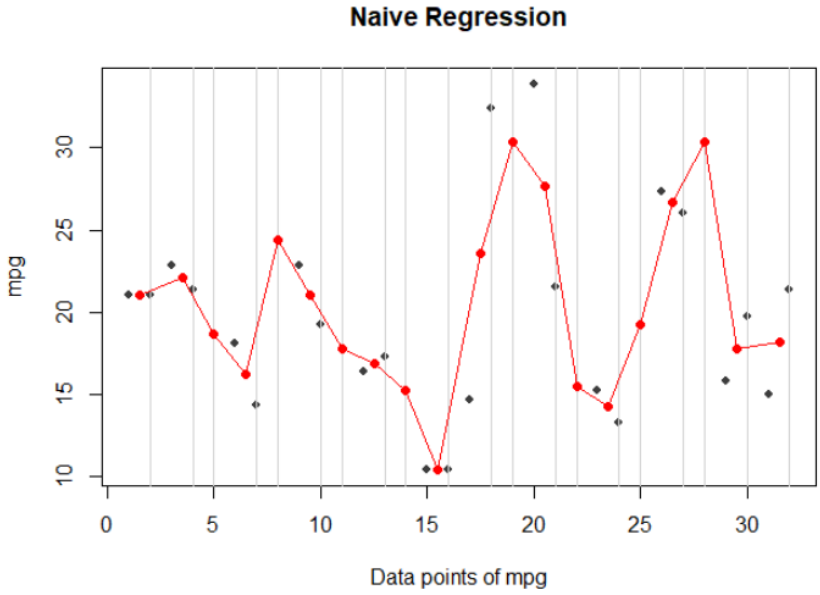


3. KERNSCHAETZUNG (KERNEL REGRESSION - nicht-parametrisch)

Der folgende Kernschätzer wurde mittels verschiedener Bandbreiten berechnet. Eine größere Bandbreite führt zu einer eher zackigen Struktur, während eine große Bandbreite zu einer sehr „glatten“ Struktur mit weniger Ausrissen führt, wie es aus der untenstehenden Grafik ersichtlich ist.

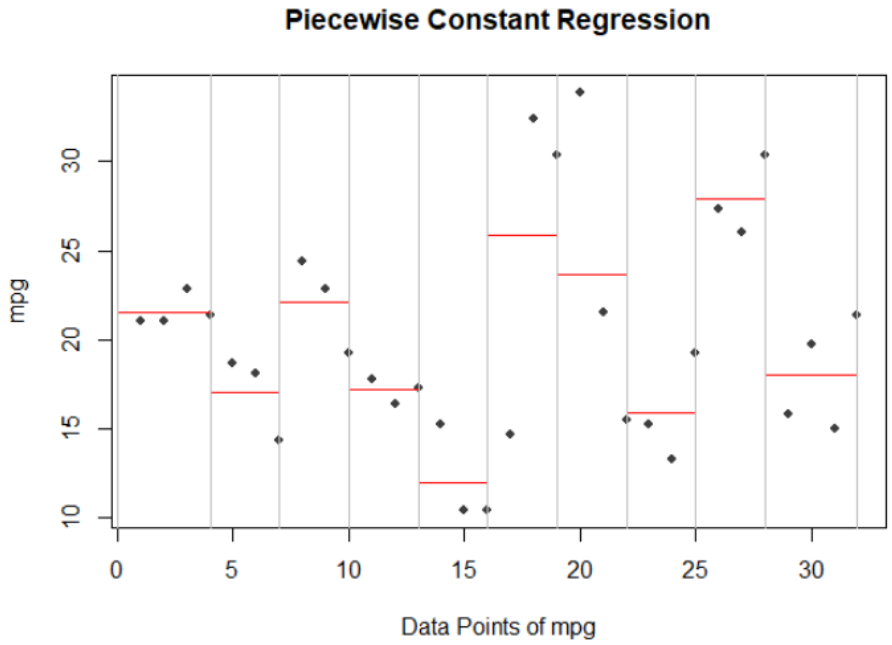


4. NAIVE REGRESSION



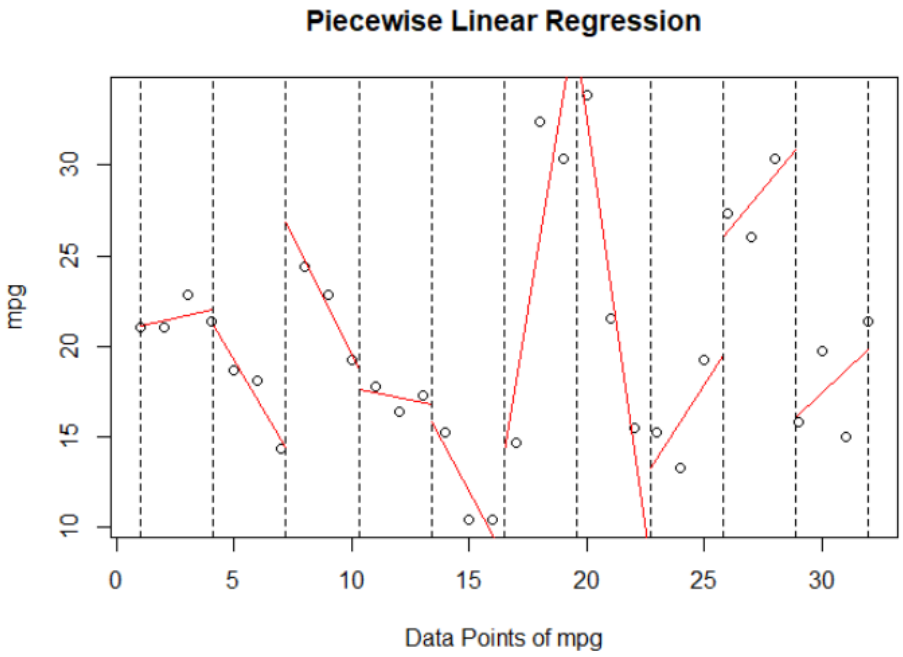
5. PIECEWISE CONSTANT REGRESSION

In einzelnen Intervallen werden jeweils konstante Geraden aus den Punkten berechnet und durch das Intervall gelegt.



6. BROKEN STICK REGRESSION mit lokalen funktionen (Piecewise Linear Regression)

Die Punktewolke wird in Intervalle eingeteilt und es wird pro Intervall eine Gerade durch die Punkte gelegt, sodass eine stückweise Struktur entsteht.



R-Code zu Aufgabe 1:

```
# AUFGABENBLATT 1
# Cordula Eggerth (00750881)

rm(list=ls())

install.packages("car")
install.packages("np")
install.packages("UsingR")
install.packages("XLConnect")
install.packages("xlsx")
install.packages("MASS")

library(car)
library(np)
library(UsingR)
library(XLConnectJars)
library(XLConnect)
require(XLConnectJars)
require(XLConnect)
library(xlsxjars)
library(xlsx)
require(xlsxjars)
require(xlsx)
library(MASS)


path <- "C:/Users/Coala/Desktop/A1_ERWEIT"


#-------------------------------------------------------------------------------------------
### AUFGABE 1 ### ---------------------------------------------------------------------------
#-------------------------------------------------------------------------------------------
# 1. Führen Sie mit dem Datensatz mtcars verschiedene sinnvolle regressionsanalytische
#    Auswertungen und Visualisierungen durch.

data("mtcars")
mtcars


summary(mtcars) # deskriptive zusammenfassung der daten
n <- nrow(mtcars)
x.werte <- 1:n
scatterplotMatrix(mtcars) # diagonalelemente zeigen histogramm der jeweiligen
                          # variable
sapply(mtcars,mean) # mittelwerte der variablen
sapply(mtcars,sd) # standardabweichung der variablen

# 1.a. LINEAR REGRESSION (univariat)

# fall: erklaere mpg (miles per US gallon) durch wt (weight in 1000lbs)
ggplot(mtcars, aes(wt, mpg)) + geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  ylab("Miles per US Gallon") +
  xlab("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")

# fall: erklaere mpg (miles per US gallon) durch qsec (1/4 mile time)
ggplot(mtcars, aes(qsec, mpg)) + geom_point() +
  geom_smooth(method = "lm", se = TRUE) +
  ylab("Miles per (US) Gallon") +
  xlab("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")
```

```
# 1.b. LINEAR REGRESSION (multiple)
#      (hier: mpg durch restliche variablen erklaeren als additives modell)
multiple_lm <- lm(mtcars$mpg ~ mtcars$cyl + mtcars$disp + mtcars$hp +
                  mtcars$drat + mtcars$wt + mtcars$qsec + mtcars$vs +
                  mtcars$am + mtcars$gear + mtcars$carb)
summary(multiple_lm)
influencePlot(multiple_lm, id.method="identify", main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance")

plot(multiple_lm$fitted.values) # plot fitted values vs. residuals

plot(multiple_lm$residuals) # qqplot


# 2.a. LOCAL REGRESSION
    # LINEAR LOESS FIT (polynome 1. grades)
alpha <- c(0.1, 0.35, 0.75)
colors <- c("mediumblue","cadetblue","red2")

plot(x.werte, mtcars$mpg, pch=16, col="black",
     xlab="Data points mpg", ylab="Miles per US Gallon",
     main = "Different Linear Loess Fits")


for(i in 1:length(alpha)){
  localregression.model <- loess(mtcars$mpg ~ x.werte, span=alpha[i], degree=1)
  lines(x.werte, localregression.model$fitted, lwd=2, col=colors[i])
}

legend(3,33,c(expression(alpha == 0.10), expression(alpha == 0.35),
              expression(alpha == 0.75)), lty=c(1,1,1), col=colors, lwd=c(2,2,2),
       cex=0.8)


   # QUADRATIC LOESS FIT (polynome 2. grades)
alpha <- c(0.1, 0.35, 0.75)
colors <- c("mediumblue","cadetblue","red2")

plot(x.werte, mtcars$mpg, pch=16, col="black",
     xlab="Data points mpg", ylab="Miles per US Gallon",
     main = "Different Linear Loess Fits")

for(i in 1:length(alpha)){
  localregression.model <- loess(mtcars$mpg ~ x.werte, span=alpha[i], degree=2)
  lines(x.werte, localregression.model$fitted, lwd=2, col=colors[i])
}

legend(3,33,c(expression(alpha == 0.10), expression(alpha == 0.35),
       expression(alpha == 0.75)), lty=c(1,1,1), col=colors, lwd=c(2,2,2),
       cex=0.8)

  # linearitaet ueberpruefen:
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(size=2, shape=21, fill="blue") +
  stat_smooth(method=loess)

  # lineare regression vs. loess
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(size=2, shape=21, fill="blue") +
  stat_smooth(method=loess, col="red",se=FALSE) +
  stat_smooth(method=lm)
```

```
# 2.b. LOESS bzw. LOCAL POLYNOMIAL REGRESSION ("smoothing", univariat)

# fall: erklaere mpg (miles per US gallon) durch wt (weight in 1000lbs)
#       span=0.9 (glattere kurve, weil groessere anzahl an punkten in
#       umgebung in berechnung miteinbezogen)
ggplot(mtcars, aes(wt, mpg)) +
  stat_smooth(span=0.9) + geom_point() +
  ylab("Miles per US Gallon") +
  xlab ("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")

#       span=0.3 (weniger glatte kurve, weil kleine anzahl von punkten
#       in umgebung in berechnung miteinbezogen)
ggplot(mtcars, aes(wt, mpg)) +
  stat_smooth(span=0.3) + geom_point() +
  ylab("Miles per US Gallon") +
  xlab ("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")

# fall: erklaere mpg (miles per US gallon) durch qsec (1/4 mile time)
#       span=0.9
ggplot(mtcars, aes(qsec, mpg)) + geom_point() +
  stat_smooth(span=0.9) + geom_point() +
  ylab("Miles per (US) Gallon") +
  xlab("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")


#       span=0.7
ggplot(mtcars, aes(qsec, mpg)) + geom_point() +
  stat_smooth(span=0.7) + geom_point() +
  ylab("Miles per (US) Gallon") +
  xlab("Weight (in 1000lbs)") +
  ggtitle("Influence of Weight on Miles per Gallon")


# 3.a. KERNSCHAETZUNG (KERNEL REGRESSION - nicht-parametrisch)
plot(x.werte, mtcars$mpg, pch=16, col="gray24",
     xlab="Datenpunkte ", ylab="mpg",
     main = "Kernel Regression Estimate")

lines(ksmooth(x.werte,mtcars$mpg,"normal"),col="chocolate")
lines(ksmooth(x.werte,mtcars$mpg,"normal",bandwidth=3),
      col="orchid3", lwd=2) # bandwidth 3
lines(ksmooth(x.werte,mtcars$mpg,"normal",bandwidth=5),
      col="mediumseagreen", lwd=2) # bandwidth 5
lines(ksmooth(x.werte,mtcars$mpg,"normal",bandwidth=10),
      col="thistle", lwd=2) # bandwidth 10
legend("topleft",
       c("bandwidth=0.5","bandwidth=3",
         "bandwidth=5","bandwidth=10"),
       lty=1,
       col=c("chocolate","orchid3","mediumseagreen",
             "thistle"),
       bty="n")
```

```
# 3.b. KERNEL REGRESSION (univariat)
# using library np from:
# https://socialsciences.mcmaster.ca/racinej/Gallery/Regression.html
# local linear model
local_linear_model <- npreg(mpg~wt,regtype="ll")
# local constant model
local_constant_model <- npreg(mpg~wt)
# plot
plot(wt,mpg,cex=0.25,col="black",pch=3)
lines(wt,fitted(local_linear_model),col="chocolate",lty=2)
lines(wt,fitted(local_constant_model),col="cadetblue",lty=1)
legend("topright", c("local linear model","local constant model"),
       lty=c(2,1), col=c("chocolate","cadetblue"), bty="n")


# 4. NAIVE REGRESSION
plot(x.werte, mtcars$mpg, pch=20, col="gray24", cex=1.2,
     xlab="Data points of mpg", ylab="mpg", main = "Naive Regression")

x.breaks = cut(x.werte, breaks=20, labels=F)
x.naive <- by(x.werte, x.breaks, mean)
x.cut   <- by(x.werte, x.breaks, max)
abline(v=x.cut, col="lightgrey", lwd=0.7)
y.naive <- by(mtcars$mpg, x.breaks, mean)
lines(x.naive, y.naive , pch=19, cex=1.0, type="o", lwd=1, col=2)



# 5. PIECEWISE CONSTANT REGRESSION
const.reg <- function(x, y, interv=10, type=mean, ...){
  plot(x, y, pch=20, col="gray24", cex=1.2, ...)
  x.breaks = cut(x, breaks=interv, labels=F)
  x.naive <- by(x, x.breaks, type)
  x.cut   <- c(0, by(x, x.breaks, max))
  y.naive <- by(y, x.breaks, type)
  for (i in 1:(length(x.breaks)-1))
    lines(c(x.cut[i], x.cut[i+1]), c(y.naive[i],y.naive[i]), col=2)
  abline(v=x.cut, col="grey", lwd=0.8)
}

const.reg(x.werte, mtcars$mpg, xlab="Data Points of mpg", ylab="mpg",
          main = "Piecewise Constant Regression")


# 6. BROKEN STICK REGRESSION mit lokalen funktionen
#    (piecewise linear)
piecewise <- function(x, y, wo=2, vertical=T, ...)
{
  if(length(wo) == 1)
    wo <- quantile(x, probs = seq(0, 1, 1/wo))
  bruch <- (cut(x, wo,labels=F))
  bruch[is.na(bruch)] <- 1
  res <- vector("list", max(bruch))
  plot(x, y, ...)
  for(i in 1:length(res)) {
    res[[i]] <- lm(y ~ x, subset = (bruch == i))
    xp <- wo[i:(i + 1)]
    yp <- xp * res[[i]]$coefficients[2] + res[[i]]$coefficients[1]
    lines(xp, yp, col=2)
  }
  if (vertical) abline(v=wo, lty=2)
  res
}

piecewise(x.werte, mtcars$mpg, 10, xlab="Data Points of mpg", ylab="mpg",
          main = "Piecewise Linear Regression")
```

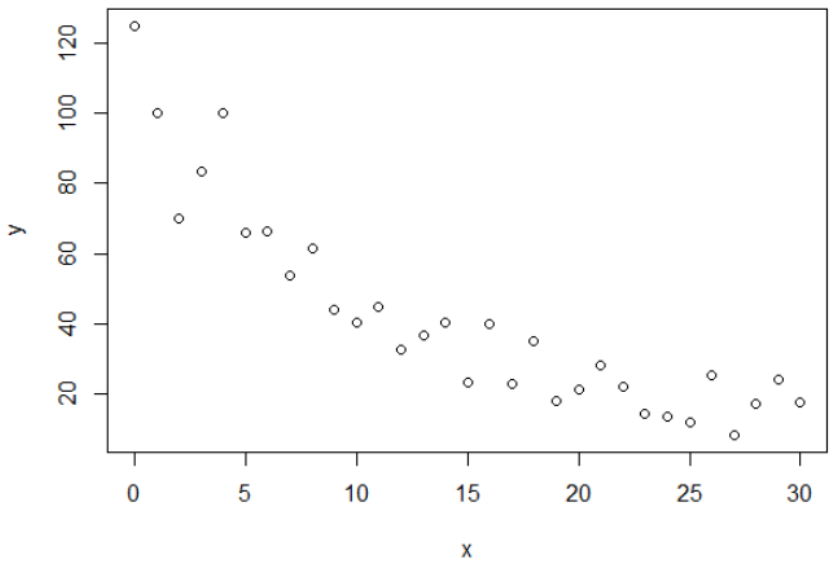
Aufgabe 2:

Führen Sie mit dem Datensatz *decay* verschiedene Modellierungen des offensichtlich nichtlinearen Zusammenhanges durch.

Der Datensatz *decay* besteht ausschließlich aus numerischen Werten:

```
> decay_data
    x          y
1   0 125.000000
2   1 100.248858
3   2  70.000000
4   3  83.470795
5   4 100.000000
6   5  65.907870
7   6  66.533715
8   7  53.588087
9   8  61.332351
10  9  43.927435
11 10  40.295448
12 11  44.713459
13 12  32.533143
14 13  36.640336
15 14  40.154711
16 15  23.080295
17 16  39.867928
18 17  22.849786
19 18  35.014645
20 19  17.977267
21 20  21.159180
22 21  27.998273
23 22  21.885735
24 23  14.273962
25 24  13.665969
26 25  11.816435
27 26  25.189016
28 27   8.195644
29 28  17.191337
30 29  24.283354
31 30  17.722776
```

Zwischen x und y besteht ein nicht-linearer Zusammenhang, wenn man nur den Plot der beiden Variablen ansieht:



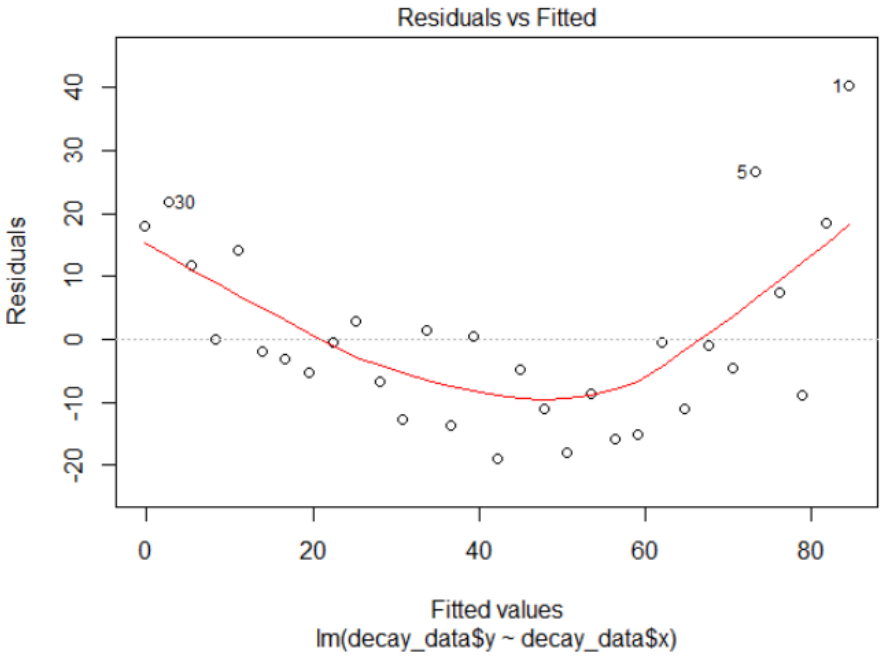
Die Korrelation zwischen x- und y-Variable ist stark negativ:

```
> cor(decay_data$x,decay_data$y)
[1] -0.8768214
```

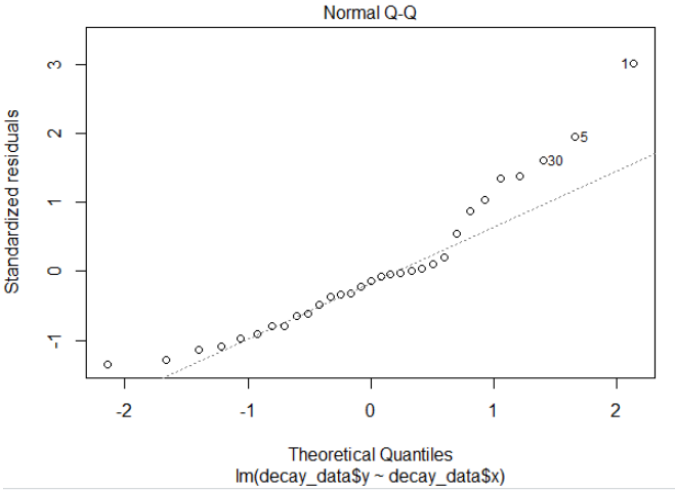
Die Anwendung des gewöhnlichen linearen Modells lm() ergibt für die Erklärung der y-Variable durch die x-Variable, dass das Intercept und die x-Variable auf dem 0.001 Level signifikant sind.

```
> summary(lm_decay)

Call:
lm(formula = decay_data$y ~ decay_data$x)

Residuals:
    Min      1Q  Median      3Q     Max
-19.065 -10.029  -2.058   5.107  40.447

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   84.5534     5.0277   16.82  < 2e-16 ***
decay_data$x  -2.8272     0.2879   -9.82 9.94e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.34 on 29 degrees of freedom
Multiple R-squared:  0.7688,    Adjusted R-squared:  0.7608
F-statistic: 96.44 on 1 and 29 DF,  p-value: 9.939e-11
```

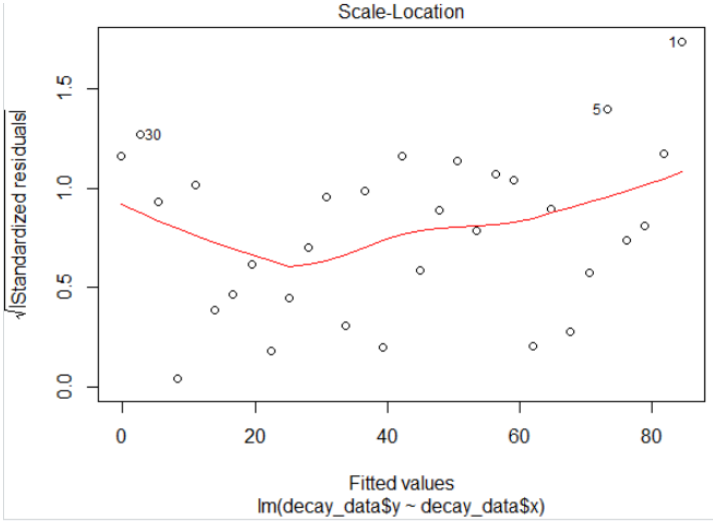
Residuals vs. Fitted Values für das gewöhnliche lineare Modell ergeben ein Bild, dass auf einen nicht-linearen Zusammenhang schließen lässt, da die Verteilung der Residuenpunkte sehr nach unten hin gekrümmt ist:

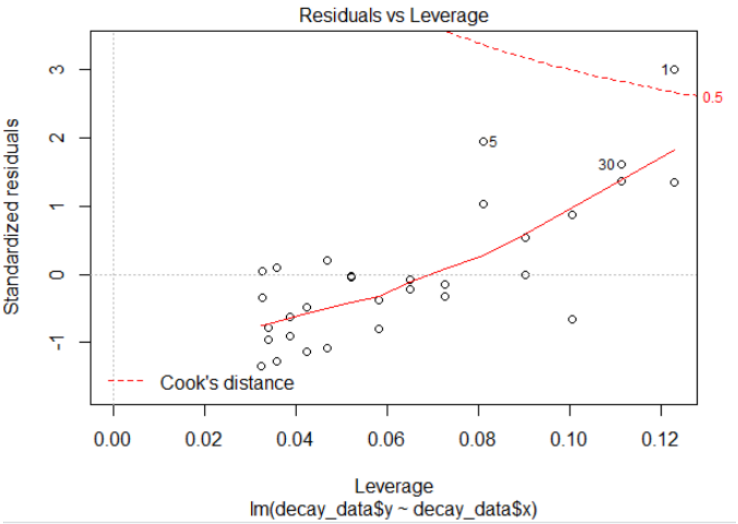


Bei der Ansicht des Q-Q-Plots fällt auf, dass die Residuen nicht normalverteilt sind, da sie von der theoretischen Quantillinie relativ stark abweichen:

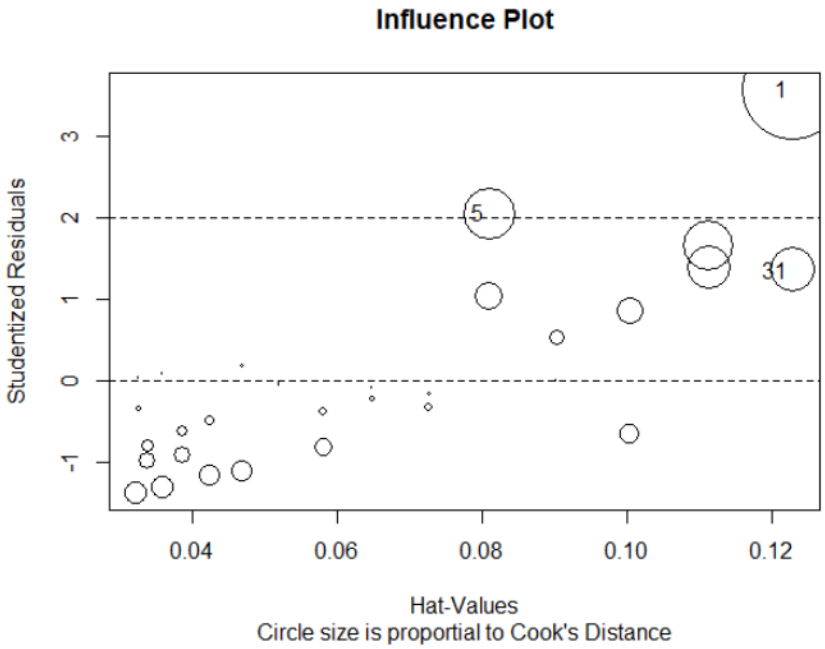


Scale-Location- und Residuals vs. Leverage Plots (Punkt 1 liegt außerhalb der Cook's Distance):

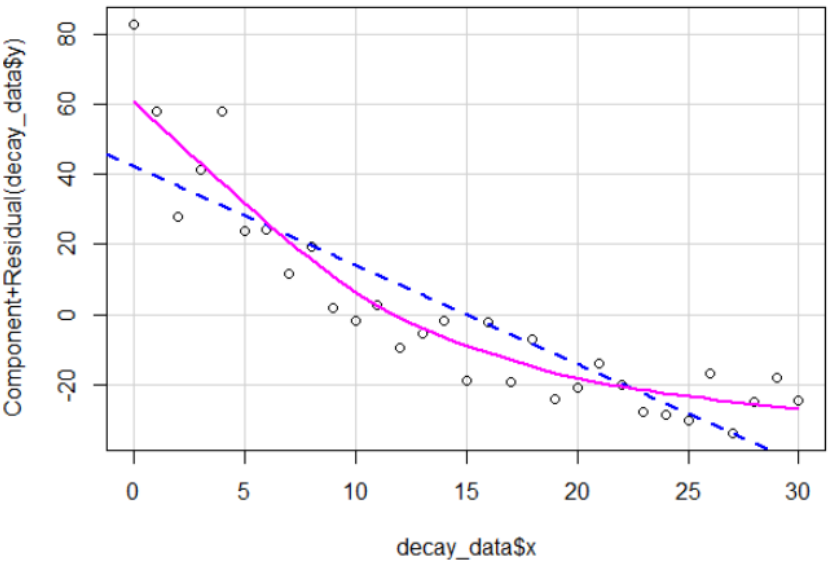




Der Influence Plot zeigt, dass es einige Punkte, wie z.B. 1, 5, oder 31 gibt, die die Residuen beeinflussen.



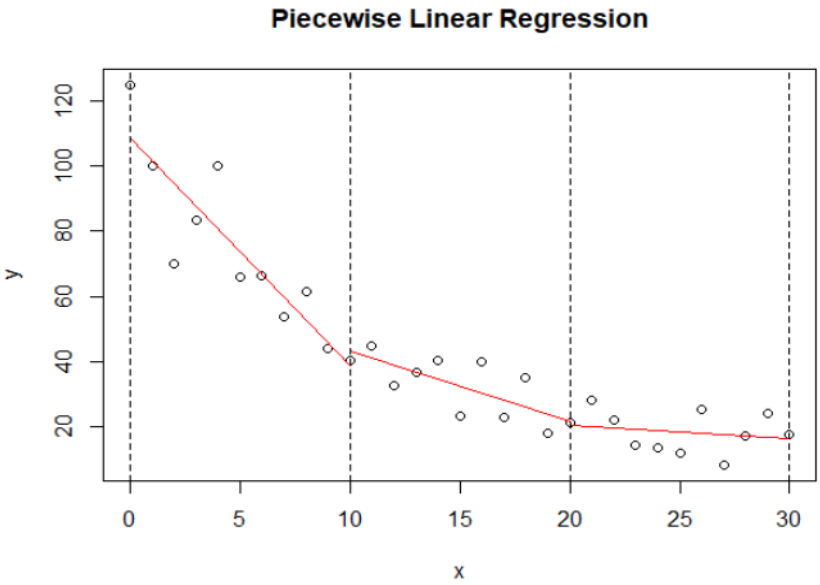
Des Weiteren wurde ein Component+Residual Plot (= C-R-Plot) angefertigt, wodurch man erkennen kann, of die Prediktoren in einer lineare Beziehung zur abhängigen Variable stehen.[3] Der partielle Residuenplot zeigt die Residuen des einen Prediktors vs. die abhängige Variable.[4] Im C-R-Plot wird eine Linie eingezeichnet, wo der beste Fit liegt.[5] Ein großer Unterschied zwischen den beiden Linien legt nahe, dass der Prediktor und die abhängige Variable keine lineare Beziehung haben, was in der untenstehenden Grafik der Fall ist:[6]



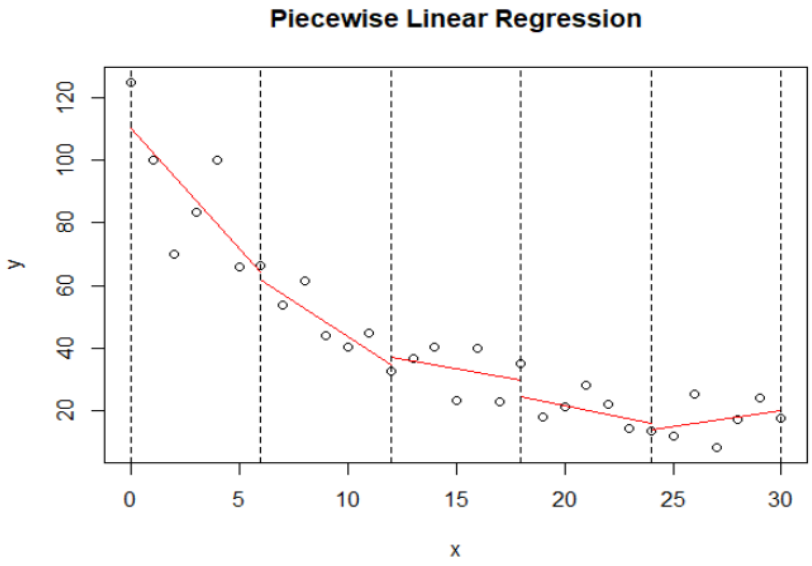
[3] http://www.vikparuchuri.com/blog/r-regression-diagnostics-part-1/.
[4] http://www.vikparuchuri.com/blog/r-regression-diagnostics-part-1/.
[5] http://www.vikparuchuri.com/blog/r-regression-diagnostics-part-1/.
[6] http://www.vikparuchuri.com/blog/r-regression-diagnostics-part-1/.

Außerdem werden nun einige andere Regressionsverfahren angewandt, aus denen ebenfalls hervorgeht, dass es sich hier um einen nicht-linearen Zusammenhang handelt:
Aus der Anwendung der Piecewise Linear Regression sieht man, dass die Steigung der jeweiligen Regressionsgeraden mit 3 bzw. 5 Intervallen sehr verschieden sind, und dann insgesamt kein linearer Zusammenhang zwischen x und y vorliegt.

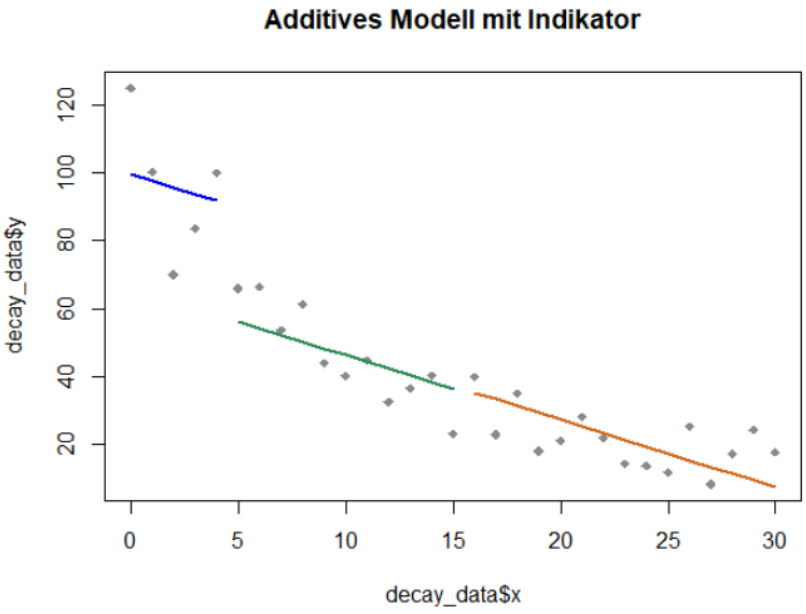
Piecewise Linear Regression mit 3 Intervallen:



Piecewise Linear Regression mit 5 Intervallen:



Danach wird ein additives Modell mit zwei Indikatoren untersucht, wobei sich zeigt, dass die Regressionsgeraden in den einzelnen von den Indikatoren abgedeckten Bereichen sehr verschieden sind und nicht ineinander übergehen. In diesem Modell ist der *indikator_5* signifikant auf dem 0.001 Level. Der Zusammenhang ist also nicht linear:



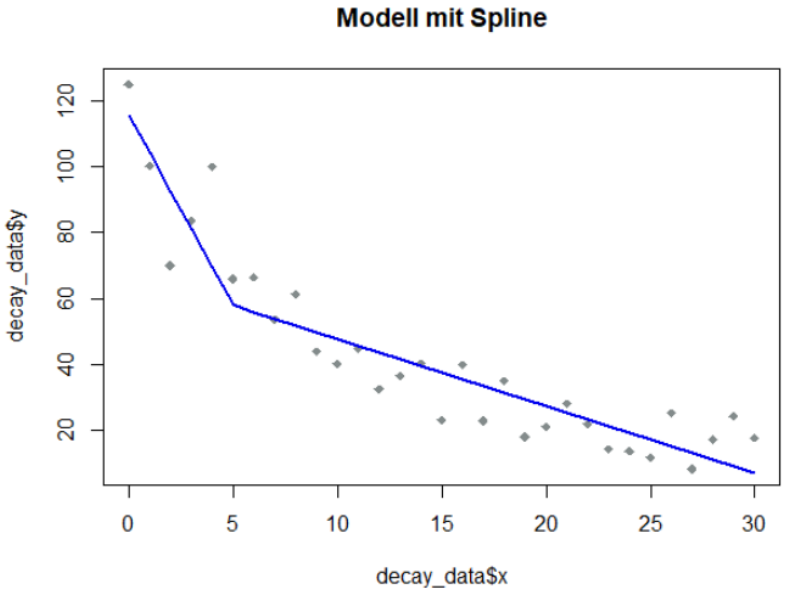

```
Call:
lm(formula = decay_data$y ~ decay_data$x + indikator_15 + indikator_5)

Residuals:
     Min       1Q   Median       3Q      Max
-25.7439  -6.0217   0.4605   7.0632  25.2699

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   66.1771     6.2665  10.560 4.35e-11 ***
decay_data$x  -1.9931     0.5366  -3.714 0.000938 ***
indikator_15   0.9370     8.1743   0.115 0.909593
indikator_5   33.5531     7.2070   4.656 7.68e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.73 on 27 degrees of freedom
Multiple R-squared:  0.8794,    Adjusted R-squared:  0.866
F-statistic: 65.61 on 3 and 27 DF,  p-value: 1.596e-12
```

Ein Modell mit Interaktion zwischen den jeweiligen Indikatoren und der x-Variable zeigt, dass indikator_15 mit der x-Variable interagiert. Das R^2 erhöht sich im Modell im Vergleich zum additiven Modell von ca. 88% auf 91%.

```
Call:
lm(formula = decay_data$y ~ decay_data$x * indikator_15 + decay_data$x *
    indikator_5)

Residuals:
     Min       1Q   Median       3Q      Max
-25.7439  -5.9471  -0.4697   5.9567  17.6117

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             85.0924     9.5185   8.940 2.93e-09 ***
decay_data$x            -3.8846     0.9075  -4.280 0.000241 ***
indikator_15           -38.9179    16.3650  -2.378 0.025359 *
indikator_5             24.0071    12.0400   1.994 0.057174 .
decay_data$x:indikator_15   2.8019     1.0711   2.616 0.014872 *
decay_data$x:indikator_5   -2.7932     3.1438  -0.888 0.382763
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

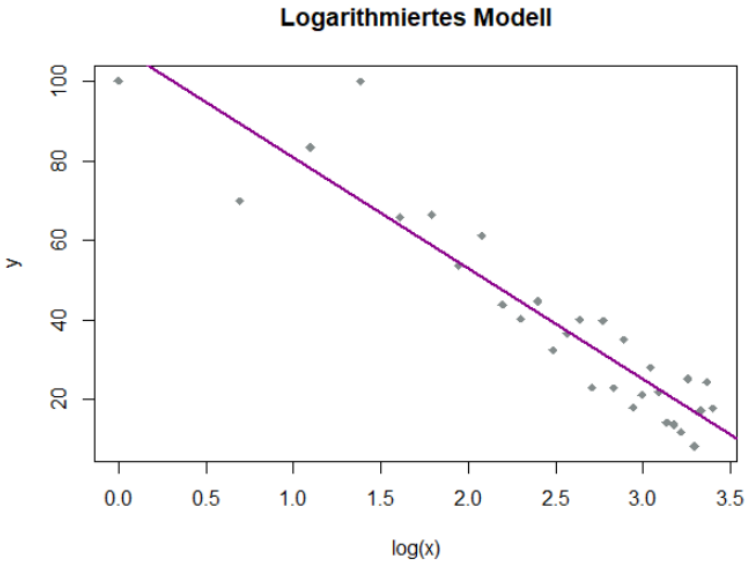
```
Residual standard error: 9.518 on 25 degrees of freedom
Multiple R-squared:  0.9122,    Adjusted R-squared:  0.8946
F-statistic: 51.92 on 5 and 25 DF,  p-value: 2.114e-12
```

Um nahtlose Übergänge zwischen den Bereichen mit unterschiedlicher Steigung zu erhalten, wird nun ein Modell mit Spline angewandt. Daraus ist ersichtlich, dass der Zusammenhang nicht-linear ist, da sich die Steigung ab x-Wert 5 sehr stark ändert. Der *spline_def*, also der definierte Spline, ist signifikant auf dem 0.001 Level.



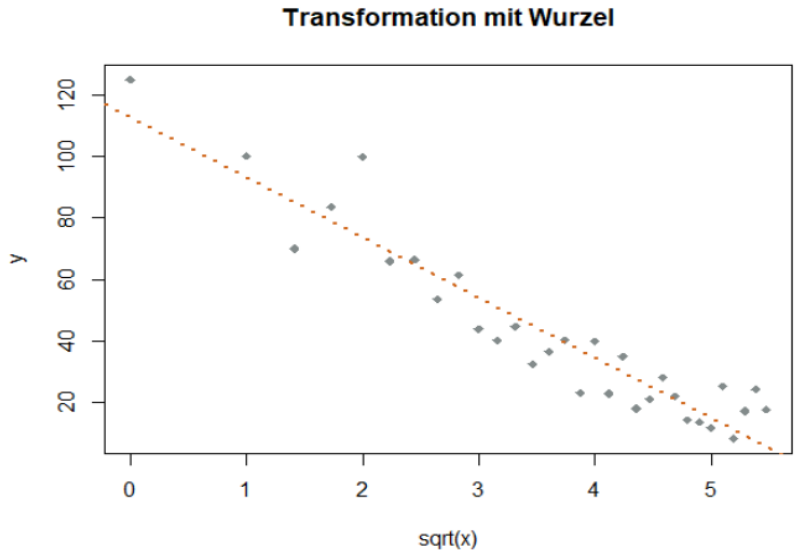

```
Call:
lm(formula = decay_data$y ~ decay_data$x + spline_def)

Residuals:
    Min      1Q  Median      3Q     Max
-22.529  -6.050  -1.040   7.047  30.473

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   115.530      7.349  15.720 2.01e-15 ***
decay_data$x  -11.501      1.782  -6.453 5.46e-07 ***
spline_def      9.455      1.929   4.902 3.62e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.7 on 28 degrees of freedom
Multiple R-squared:  0.8756,    Adjusted R-squared:  0.8667
F-statistic: 98.54 on 2 and 28 DF,  p-value: 2.127e-13
```

Durch eine logarithmische Transformation der x-Variable ergibt sich folgendes Modell:



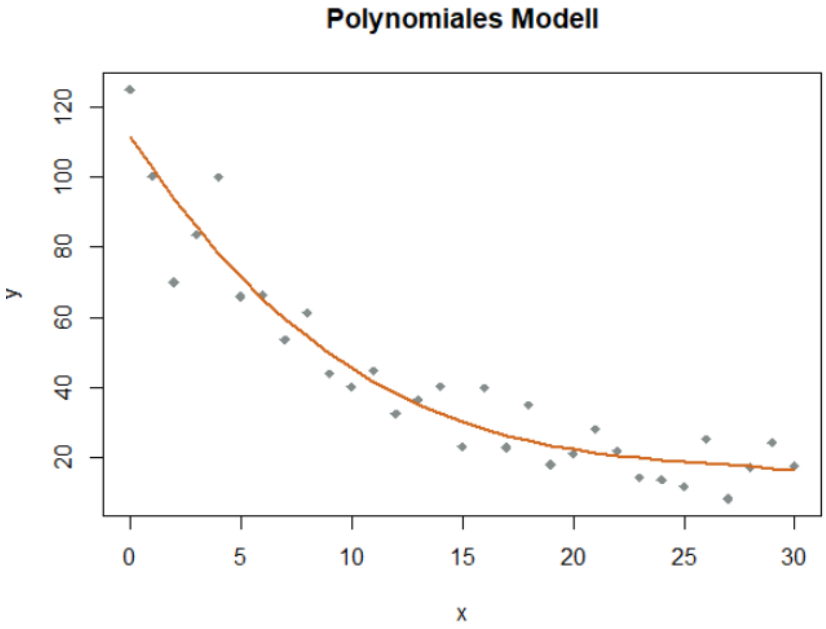

```
Call:
lm(formula = data_y ~ log_x)

Residuals:
     Min       1Q   Median       3Q      Max
-19.3970  -6.9492  -0.6671   5.2665  29.9108

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  108.705      5.324   20.42  < 2e-16 ***
log_x        -27.855      2.028  -13.73 5.78e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.288 on 28 degrees of freedom
Multiple R-squared:  0.8708,    Adjusted R-squared:  0.8661
F-statistic: 188.7 on 1 and 28 DF,  p-value: 5.778e-14
```

Transformation von x mit Wurzel:



```
Call:
lm(formula = decay_data$y ~ sqrt_x)

Residuals:
    Min      1Q  Median      3Q     Max
-15.319  -6.640  -2.952   5.214  26.169

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  113.055      4.922   22.97  < 2e-16 ***
sqrt_x       -19.612      1.271  -15.43 1.61e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.824 on 29 degrees of freedom
Multiple R-squared:  0.8914,    Adjusted R-squared:  0.8877
F-statistic: 238.1 on 1 and 29 DF,  p-value: 1.612e-15
```

Ein polynomiales Modell (mit Komponenten bis zum Grad 3) bildet gemäß R^2 die Datensituation besser ab als die Transformation mit Logarithmus oder Wurzel:




```
Call:
lm(formula = decay_data$y ~ x_zentriert + x_zentriert2 + x_zentriert3)

Residuals:
    Min      1Q  Median      3Q     Max
-23.716  -5.628  -1.134   6.767  21.664

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  30.098610   2.429549  12.389 1.19e-12 ***
x_zentriert  -2.206154   0.453987  -4.860 4.44e-05 ***
x_zentriert2  0.150589   0.022652   6.648 3.91e-07 ***
x_zentriert3 -0.004319   0.002896  -1.491    0.147
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.01 on 27 degrees of freedom
Multiple R-squared:  0.915,     Adjusted R-squared:  0.9055
F-statistic: 96.86 on 3 and 27 DF,  p-value: 1.445e-14
```

R-Code zu Aufgabe 2:

```
#------------------------------------------------------------------------
### AUFGABE 2 ### ---------------------------------------------------------
#------------------------------------------------------------------------
# 2. Führen Sie mit dem Datensatz decay verschiedene Modellierungen des
#    offensichtlichnichtlinearen Zusammenhanges durch.

decay_data <- read.table(file=paste0(path,"/decay.txt"), header=TRUE)

plot(decay_data) # nicht-linearer zusammenhang zwischen x und y

cor(decay_data$x,decay_data$y)

# lineares modell lm() anschauen
lm_decay <- lm(decay_data$y ~ decay_data$x)
summary(lm_decay)
plot(lm_decay)
outlierTest(lm_decay)
influencePlot(lm_decay, id.method="identify", main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance" )
# component residual plots um nicht-linearitaet von komponenten und
# residual plot zu analysieren
crPlots(lm_decay)


# piecewise regression, um besser die sprungstellen zu sehen:
piecewise <- function(x, y, wo=2, vertical=T, ...){
  if(length(wo) == 1)
    wo <- quantile(x, probs = seq(0, 1, 1/wo))
  bruch <- (cut(x, wo,labels=F))
  bruch[is.na(bruch)] <- 1
  res <- vector("list", max(bruch))
  plot(x, y, ...)
  for(i in 1:length(res)) {
    res[[i]] <- lm(y ~ x, subset = (bruch == i))
    xp <- wo[i:(i + 1)]
    yp <- xp * res[[i]]$coefficients[2] + res[[i]]$coefficients[1]
    lines(xp, yp, col=2)
  }
  if (vertical) abline(v=wo, lty=2)
  res
}

# 3 intervalle:
piecewise(decay_data$x, decay_data$y, 3, xlab="x", ylab="y",
          main = "Piecewise Linear Regression")
# 5 intervalle:
piecewise(decay_data$x, decay_data$y, 5, xlab="x", ylab="y",
          main = "Piecewise Linear Regression")
```

```
# ADDITIVES MODELL MIT 2 INDIKATOREN:
indikator_15 <- ifelse(decay_data$x>15, 1, 0)
indikator_5 <- ifelse(decay_data$x<5, 1, 0)
lm_decay_indikatoren <- lm(decay_data$y ~ decay_data$x + indikator_15 +
                           indikator_5)
summary(lm_decay_indikatoren)
plot(decay_data$y ~ decay_data$x, main="Additives Modell mit Indikator",
     pch=18, col="azure4")
lines(decay_data$x[decay_data$x > 15],
      predict(lm_decay_indikatoren)[decay_data$x > 15],
      col="chocolate", lwd=2)
lines(decay_data$x[decay_data$x < 5],
      predict(lm_decay_indikatoren)[decay_data$x < 5],
      col="blue2", lwd=2)
lines(decay_data$x[decay_data$x <= 15 & decay_data$x >= 5],
      predict(lm_decay_indikatoren)[decay_data$x <= 15 & decay_data$x >= 5],
      col="seagreen", lwd=2)


# MODELL MIT INTERAKTION:
lm_decay_interaktion <- lm(decay_data$y ~ decay_data$x*indikator_15 +
                                          decay_data$x*indikator_5)
summary(lm_decay_interaktion)
plot(decay_data$y ~ decay_data$x, main="Modell mit Interaktion",
     pch=18, col="azure4")
lines(decay_data$x[decay_data$x > 15],
      predict(lm_decay_interaktion)[decay_data$x > 15],
      col="chocolate", lwd=2)
lines(decay_data$x[decay_data$x < 5],
      predict(lm_decay_interaktion)[decay_data$x < 5],
      col="blue2", lwd=2)
lines(decay_data$x[decay_data$x <= 15 & decay_data$x >= 5],
      predict(lm_decay_interaktion)[decay_data$x <= 15 & decay_data$x >= 5],
      col="seagreen", lwd=2)


# MODELL MIT SPLINES interpolation:
spline_def <- ifelse(decay_data$x > 5, decay_data$x-5, 0)
model_spline <- lm(decay_data$y ~ decay_data$x + spline_def)
summary(model_spline)
plot(decay_data$y ~ decay_data$x, main="Modell mit Spline",
     pch=18, col="azure4")
lines(decay_data$x, fitted(model_spline),
      col="blue2", lwd=2)


# LOGARITHMIERTES MODELL:
log_x <- log(decay_data$x)
log_x <- log_x[2:length(log_x)] # weil log_x[1] ist -Inf
data_y <- decay_data$y[2:length(decay_data$y)]
intervalldaten <- data.frame(x = seq(from = min(log_x),
                  to = max(log_x), by = 0.10))
model_log <- lm(data_y ~ log_x)
summary(model_log)
model_log2 <- lm(data_y ~ log_x)
summary(model_log2)
plot(x=log_x, y=data_y, main="Logarithmiertes Modell",
     pch=18, col="azure4", xlab="log(x)", ylab="y")
abline(model_log, col="darkmagenta", lwd=2)
```

```
# TRANSFORMATION DURCH WURZEL:
sqrt_x <- sqrt(decay_data$x)
model_sqrt <- lsfit(sqrt_x,decay_data$y)
ls.print(model_sqrt)
model_sqrt2 <- lm(decay_data$y ~ sqrt_x)
summary(model_sqrt2)
plot(sqrt_x,decay_data$y,main="Transformation mit Wurzel",
     xlab="sqrt(x)", ylab="y", pch=18, col="azure4")
abline(model_sqrt, lty=3, col="chocolate", lwd=2)


# POLYNOMIALES MODELL
x_zentriert  <- decay_data$x-mean(decay_data$x)
x_zentriert2 <- x_zentriert^2
x_zentriert3 <- x_zentriert^3
model_poly <- lm(decay_data$y ~ x_zentriert + x_zentriert2 +
                  x_zentriert3)
summary(model_poly)

plot(decay_data$x, decay_data$y, pch=18,
     main="Polynomiales Modell",
     xlab="x",ylab="y", col="azure4")
lines(decay_data$x, fitted(model_poly), lty=1,
      col="chocolate", lwd=2)
```

Aufgabe 3:

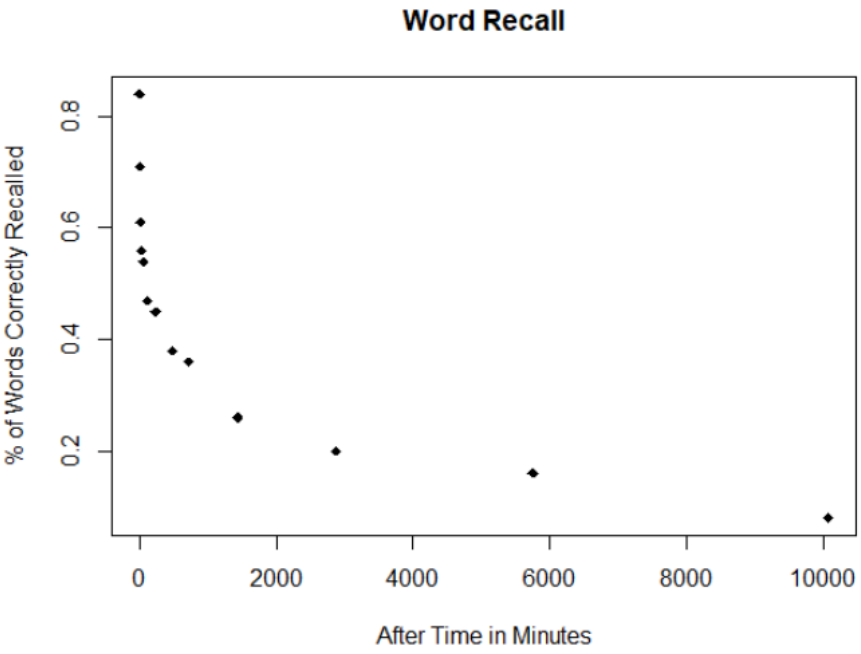
Im Excel-Sheet „*Some Datasets*“ finden Sie 5 kleine Datensätze. Führen Sie für die einzelnen Datensätze regressionsanalytische Auswertungen durch:

3.a) *WordRecall*: Check for Linearity

Anmerkung aus der Angabe:

"Data stem from a memory retention experiment in which 13 subjects were asked to memorize a list of disconnected items. The subjects were then asked to recall the items at various times up to a week later. The proportion of items (y = prop) correctly recalled at various times (x = time, in minutes) since the list was memorized were recorded."

Bereits der Plot von x- und y-Variable zeigen, dass der Zusammenhang aller Ansicht nach nicht linear ist.



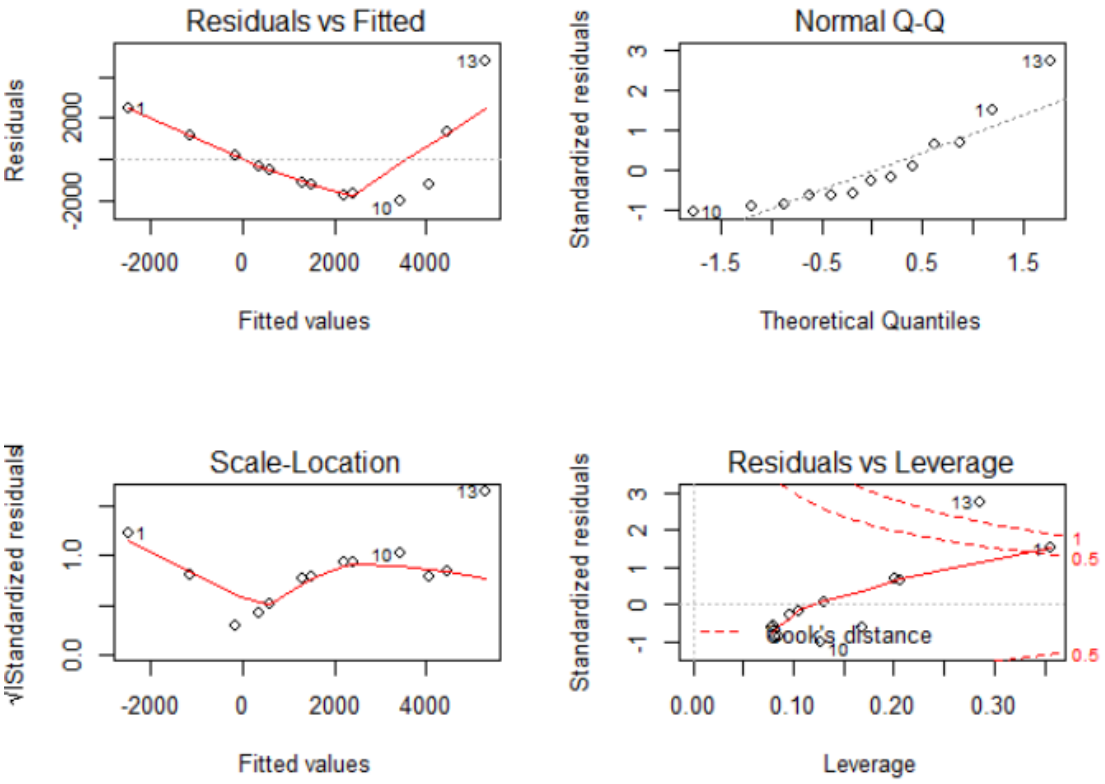
Die Korrelation zwischen *time* und *prop* ist außerdem stark negativ:

```
> cor(wordrecall_data$time, wordrecall_data$prop)
[1] -0.7555517
```

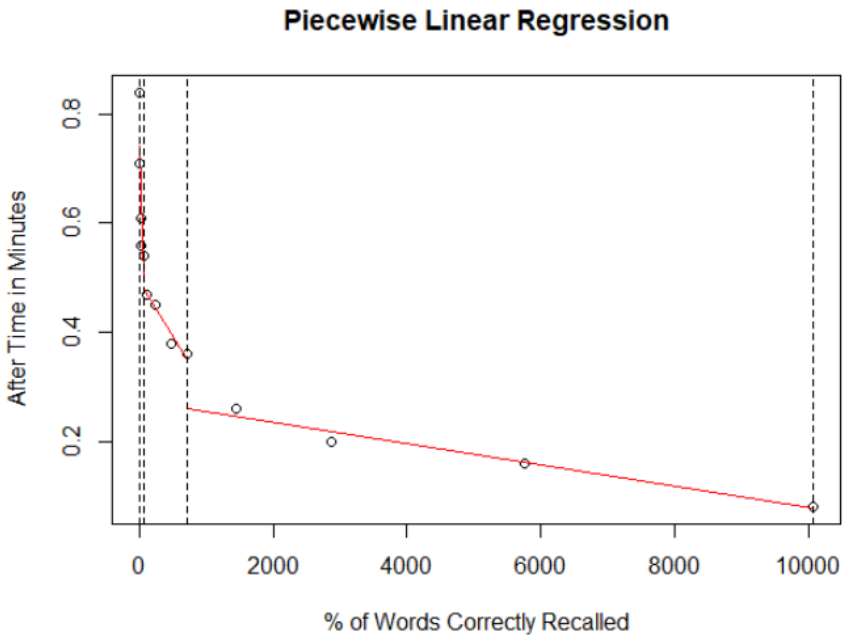
Laut linearer Regression mit lm() hat die Variable *prop* einen auf dem 0.01 Level signifikanten Einfluss auf die abhängige Variable:

```
Call:
lm(formula = wordrecall_data$time ~ wordrecall_data$prop)

Residuals:
    Min      1Q  Median      3Q     Max
-2004.8 -1258.0  -515.9  1170.9  4790.9

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)             6109       1292   4.729 0.000621 ***
wordrecall_data$prop   -10246       2678  -3.825 0.002817 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2065 on 11 degrees of freedom
Multiple R-squared:  0.5709,    Adjusted R-squared:  0.5318
F-statistic: 14.63 on 1 and 11 DF,  p-value: 0.002817
```

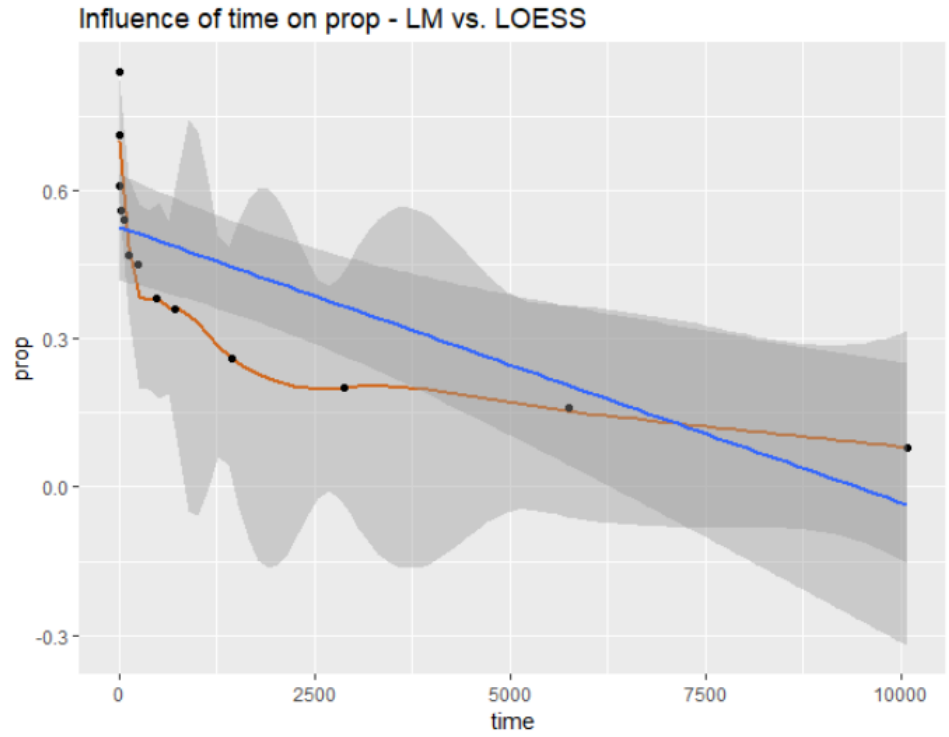
Wenn man die Diagnostic Plots begutachtet, sieht man, dass der Plot Residuals vs. Fitted einen nicht-linearen Zusammenhang andeutet, und der Plot Scale Location dies bestätigt, da die Residuen nicht zufällig als Punktewolke um die Nulllinie angeordnet sind:



Piecewise Linear Regression zeigt, dass die Steigung der Regressionsgeraden auf den einzelnen Intervallen sehr unterschiedlich ist, wodurch man sieht, dass kein linearer Zusammenhang vorliegt.

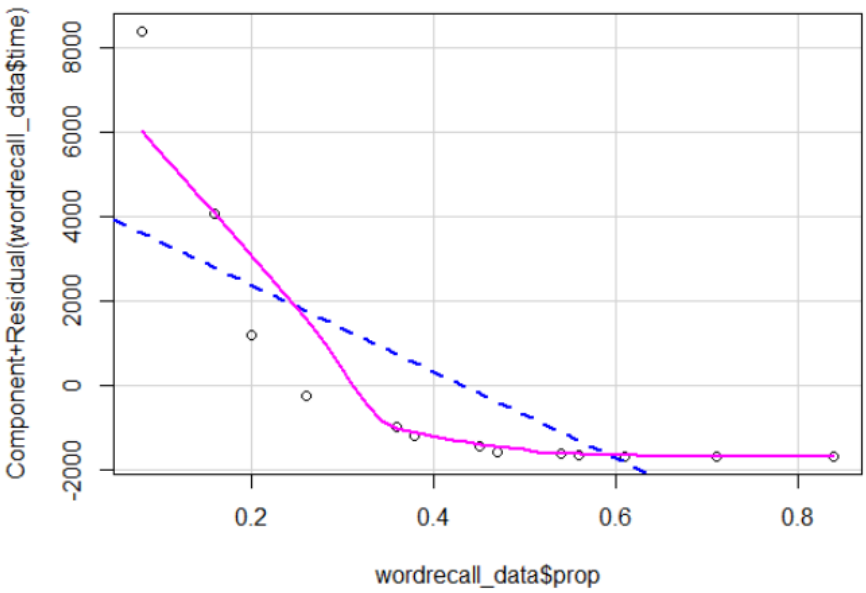


Auch der Vergleich des gewöhnlichen linearen Modells mit lokaler Regression (LOESS) ist sichtbar, dass die braune Linie (i.e. die LOESS-Linie) nicht linear ist und stark von der blauen Regressionsgerade abweicht. Daher kann man schließen, dass es keinen linearen Zusammenhang gibt.



Component+Residual Plot:

Drittens zeigt der C-R-Plot, dass die Geraden nicht übereinstimmen, was wiederum auf einen nicht linearen Zusammenhang hindeutet.

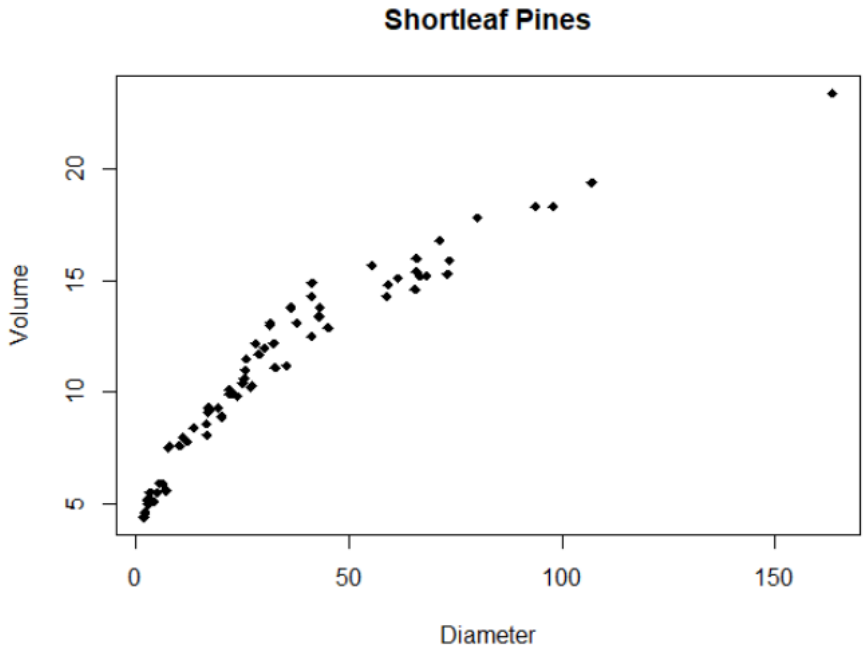


3.b.) *ShortLeaf*: Check for Linearity and Influential Observations

Anmerkung aus der Angabe:

"This is a classic data set — reported by C. Bruce and F. X. Schumacher in 1935 — concerning the diameter (x, in inches) and volume (y, in cubic feet) of n = 70 shortleaf pines. Data are used to predict the volume of the trees by means of the diameter."

Visualisierung des Datensatz:



Die Korrelation zwischen Volume und Diameter ist stark positiv:

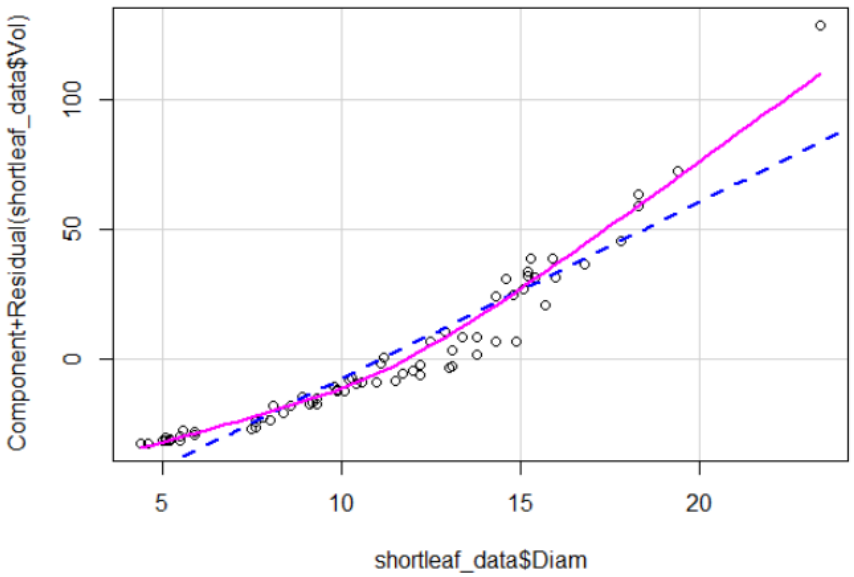
```
> cor(shortleaf_data$Vol, shortleaf_data$Diam)
[1] 0.9447509
```

Das gewöhnliche lineare Modell ergibt, dass der Regressor Diam signifikant ist auf dem 0.001 Level:

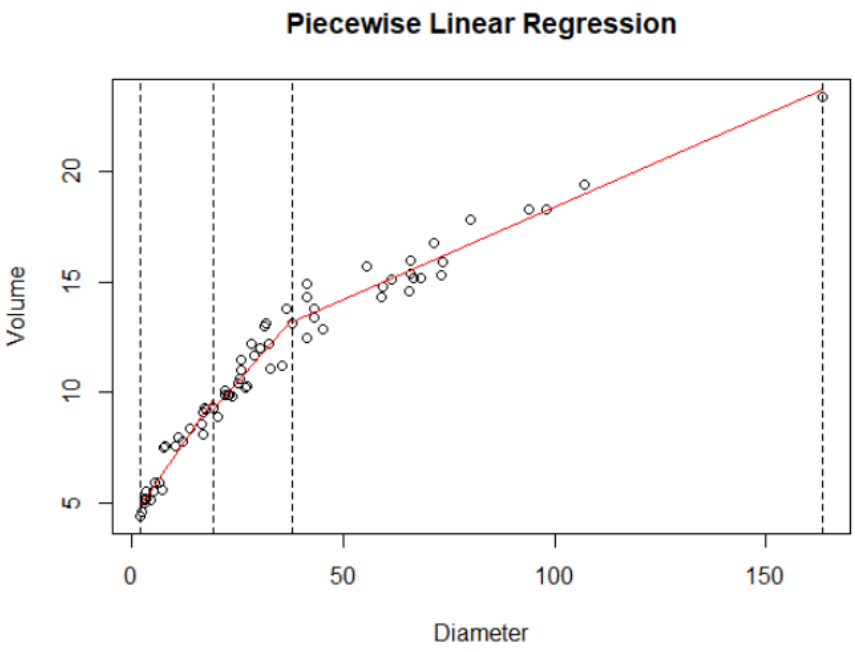
```
Call:
lm(formula = shortleaf_data$Vol ~ shortleaf_data$Diam)

Residuals:
    Min      1Q  Median      3Q     Max
-18.899  -4.768  -1.438   6.740  45.089

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         -41.5681     3.4269  -12.13   <2e-16 ***
shortleaf_data$Diam   6.8367     0.2877   23.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.875 on 68 degrees of freedom
Multiple R-squared:  0.8926,    Adjusted R-squared:  0.891
F-statistic: 564.9 on 1 and 68 DF,  p-value: < 2.2e-16
```

Component+Residual Plot:

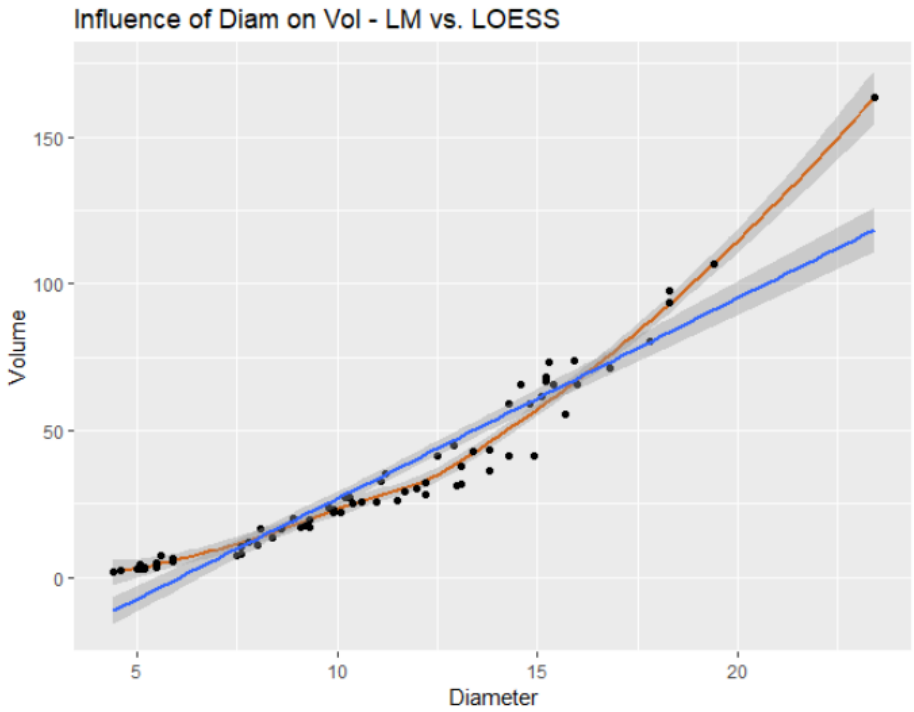
Der C-R-Plot zeigt, dass die Geraden nicht übereinstimmen, was wiederum auf einen nicht linearen Zusammenhang hindeutet.



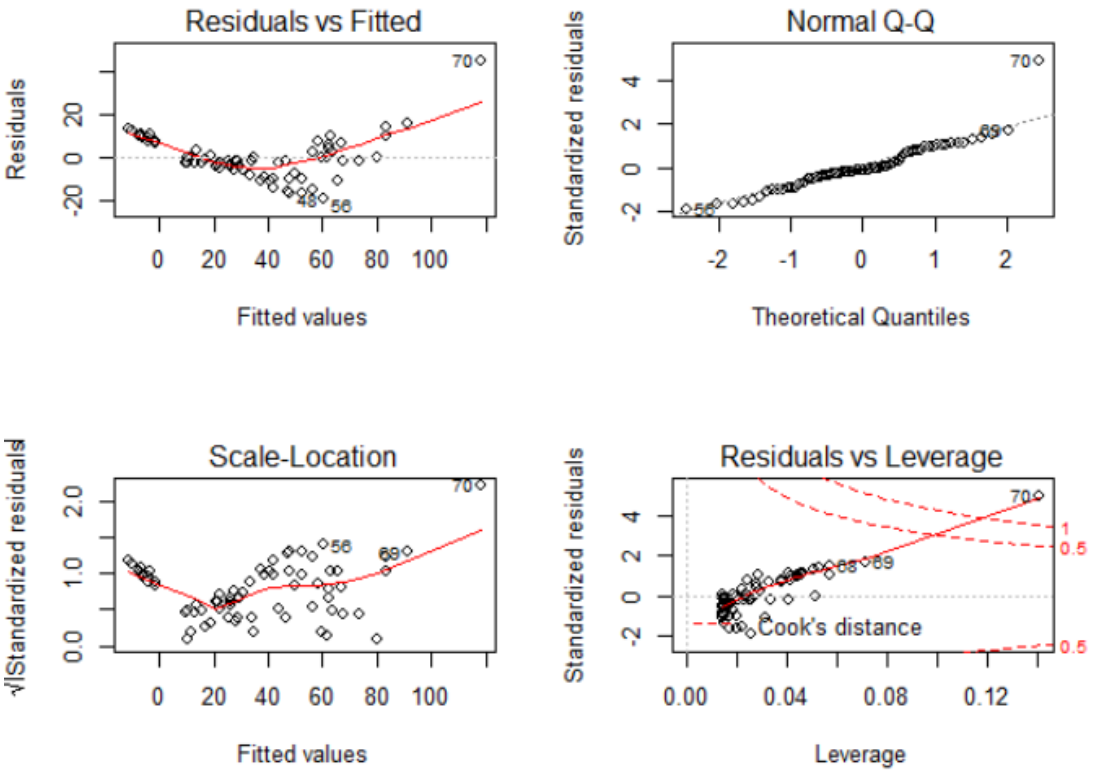
Piecewise Linear Regression zeigt, dass die Steigung der Regressionsgeraden auf den einzelnen Intervallen unterschiedlich ist, wodurch man sieht, dass kein linearer Zusammenhang vorliegt.



Auch der Vergleich des gewöhnlichen linearen Modells mit lokaler Regression (LOESS) ist sichtbar, dass die braune Linie (i.e. die LOESS-Linie) nicht linear ist und von der blauen Regressionsgerade abweicht. Daher kann man schließen, dass es keinen linearen Zusammenhang gibt.

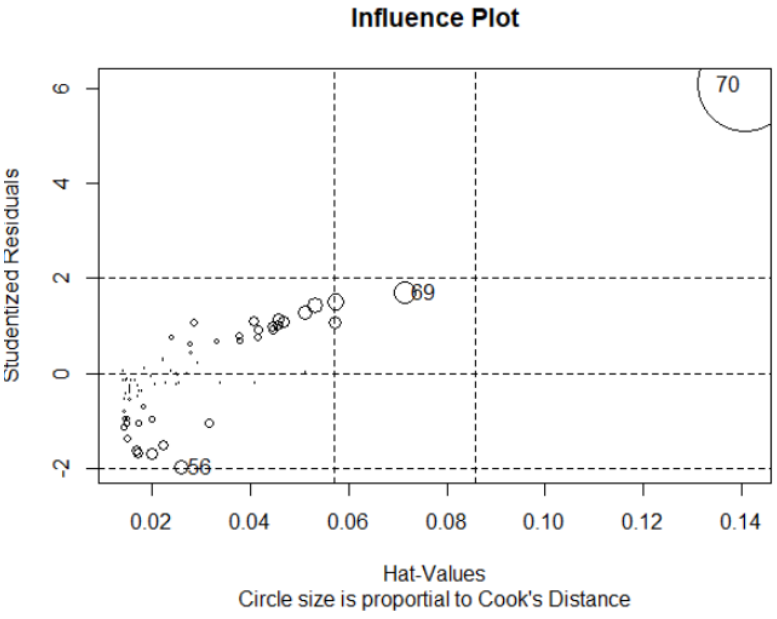


Die ***Regression Diagnostics Plots*** zu Residuals vs. Fitted und Sqrt(Standardized) Residuals vs. Fitted Values zeigen einen nicht linearen Zusammenhang. Gemäß Cook's Distance ist der Punkt 70 ein Influential Point und liegt außerhalb der Cook's Distance Linien.



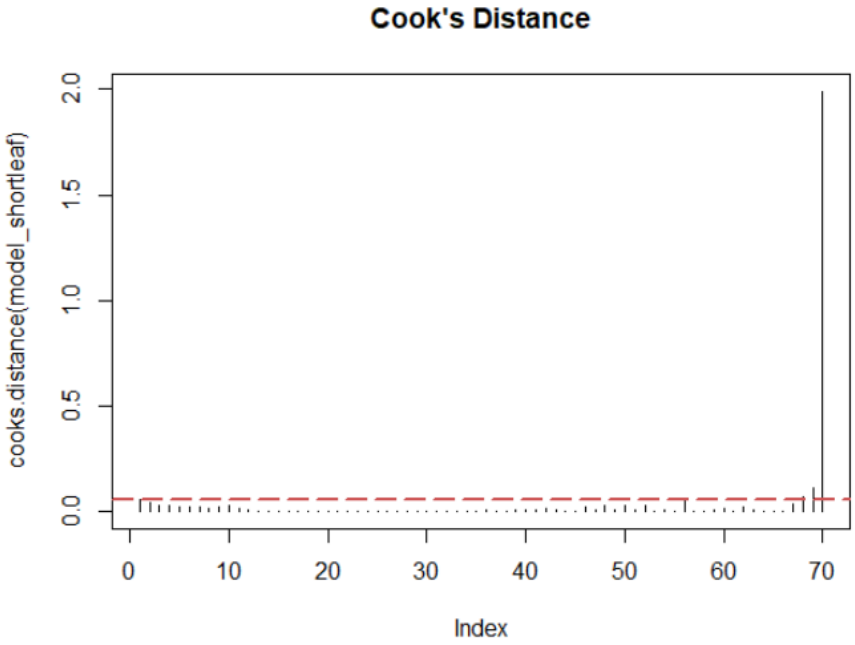
Influence Plot:

Der Influence Plot zeigt, dass die Punkte 70, 69 und 56 besonders hervorstechen und die Studentized Residuals beeinflussen.



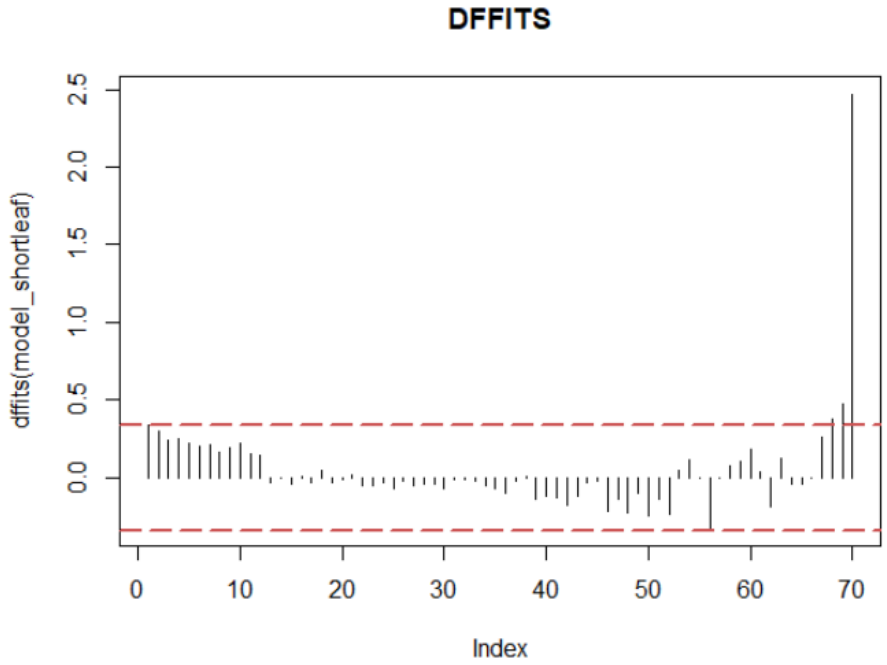
Cook's Distance Measure:

Durch Anwendung der Cook's Distance können rechts im untenstehenden Plot die einflussreichen Punkte ermittelt werden. Diese entsprechen auch u.a. den Punkten, die mittels Influence Plot oder DFFITS ermittelt werden. Hier sieht man ebenfalls, dass der Punkt 70 einflussreich ist, er im Plot stark über die strichlierte Linie hinausragt. Punkt 69 ragt leicht über die Linie hinaus.



DFFITS:

Bei DFFITS können einflussreiche Datenpunkte (hier: wieder Punkt 70 und 69) als jene Punkte, die außerhalb der strichlierten Linie liegen, festgestellt werden.



3.c.) *BirthWeight*: Use Indicator Variables

Anmerkung aus der Angabe:

"Researchers were interested in answering the research question if smoking behavior of the mother has an influence on the birth weight of a newborn child. They collected the following data (birthsmokers.txt) on a random sample of n = 32 births:

- Response (y): birth weight (Weight) in grams of baby
- Potential predictor (x1): Smoking status of mother (yes or no)
- Potential predictor (x2): length of gestation (Gest) in weeks."

Additives Modell mit Indikatorvariable:

Im additiven Modell mit Indikator für "smoke status" sind im Regressionsoutput die Variable Gestation und die Indikatorvariable I_smoke auf dem 0.001 Level signifikant. Der erklärte Anteil an der Gesamtvariabilität (R^2) liegt bei ca. 89.6%.

```
Call:
lm(formula = birthweight_data$Wgt ~ birthweight_data$Gest + I_smoke)

Residuals:
     Min       1Q   Median       3Q      Max
-223.693  -92.063   -9.365   79.663  197.507

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -2389.573    349.206  -6.843 1.63e-07 ***
birthweight_data$Gest  143.100      9.128  15.677 1.07e-15 ***
I_smoke                -244.544     41.982  -5.825 2.58e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 115.5 on 29 degrees of freedom
Multiple R-squared:  0.8964,    Adjusted R-squared:  0.8892
F-statistic: 125.4 on 2 and 29 DF,  p-value: 5.289e-15
```

Modell mit Interaktion:

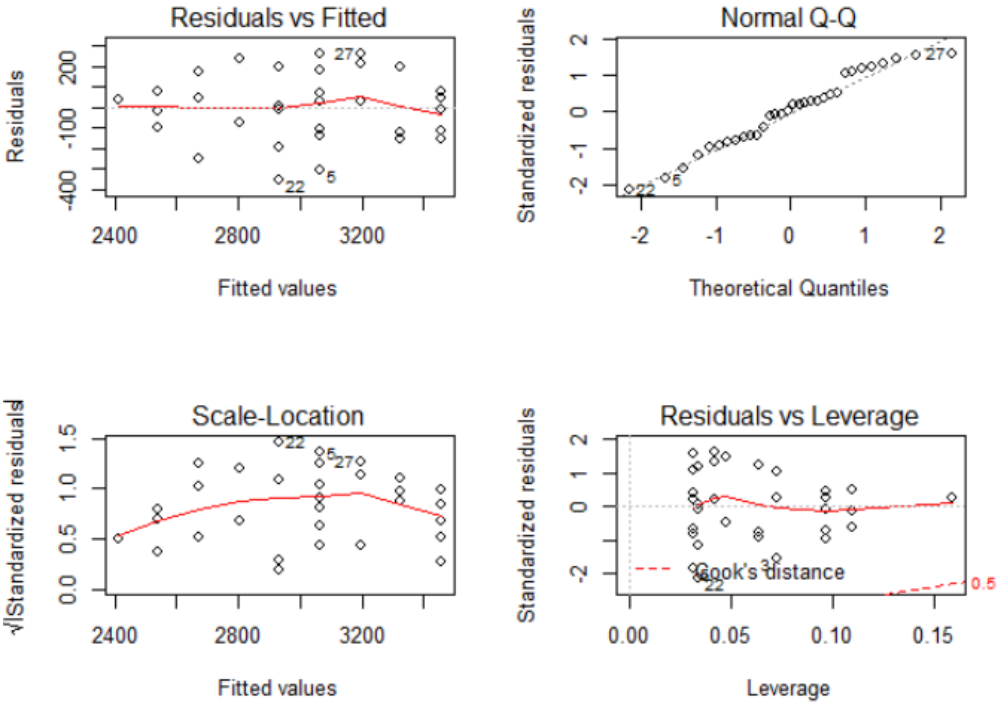
Im Modell mit Indikator für "smoke status" und Interaktion ist im Regressionsoutput die Variable Gestation auf dem 0.001 Level signifikant. Der erklärte Anteil an der Gesamtvariabilität (R^2) liegt bei ca. 89.7%, also in etwa gleich wie im additiven Modell zuvor.

```
Call:
lm(formula = birthweight_data$Wgt ~ birthweight_data$Gest * I_smoke)

Residuals:
     Min       1Q   Median       3Q      Max
-228.528  -89.560    0.273   83.629  184.529

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -2546.138    501.067  -5.081 2.22e-05 ***
birthweight_data$Gest          147.207     13.120  11.220 7.15e-12 ***
I_smoke                         71.574    716.950   0.100    0.921
birthweight_data$Gest:I_smoke   -8.178     18.515  -0.442    0.662
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 117.2 on 28 degrees of freedom
Multiple R-squared:  0.8971,    Adjusted R-squared:  0.8861
F-statistic: 81.37 on 3 and 28 DF,  p-value: 6.144e-14
```

Gesamtes Modell ohne Berücksichtigung des Indikators:

Im gesamten Modell liegt R^2, wenn man nur die Variable Gestation miteinbezieht, bei ca. 77.5%. Die Variable Gestation ist signifikant auf dem 0.001 Level.

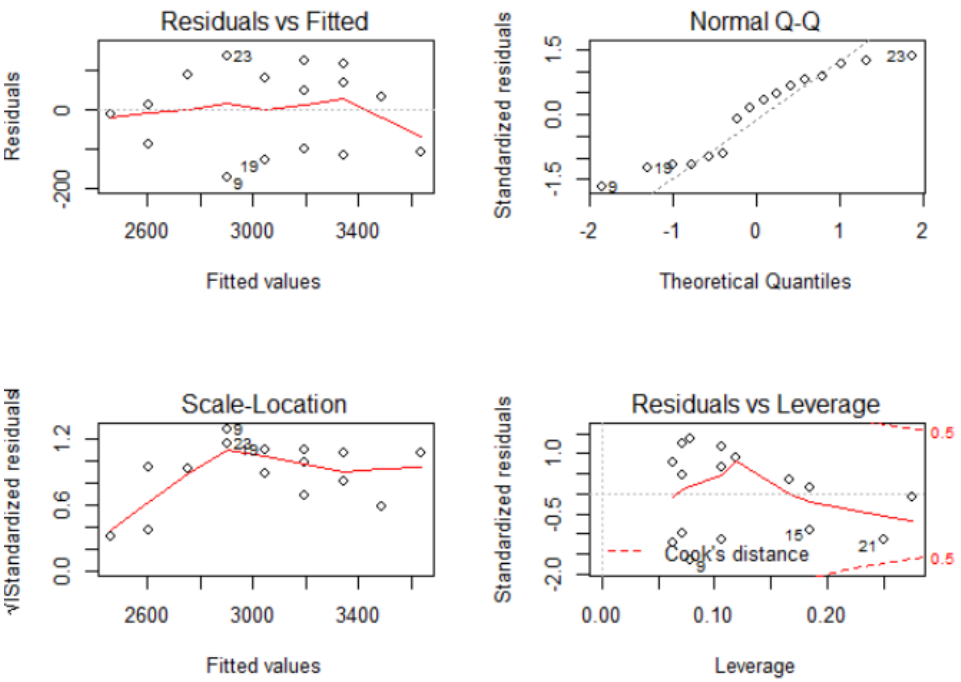
```
Call:
lm(formula = birthweight_data$Wgt ~ birthweight_data$Gest)

Residuals:
    Min      1Q  Median      3Q     Max
-354.03 -115.09   18.07  100.22  263.34

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          -2037.00     498.11  -4.089 0.000298 ***
birthweight_data$Gest   130.82      12.86  10.170 3.09e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 167.3 on 30 degrees of freedom
Multiple R-squared:  0.7752,    Adjusted R-squared:  0.7677
F-statistic: 103.4 on 1 and 30 DF,  p-value: 3.085e-11
```



Teilmodell für die Gruppe der Nicht-Raucherinnen:

Die Variable Gestation ist auf dem 0.01 Level signifikant und R^2 liegt bei 91.5%.

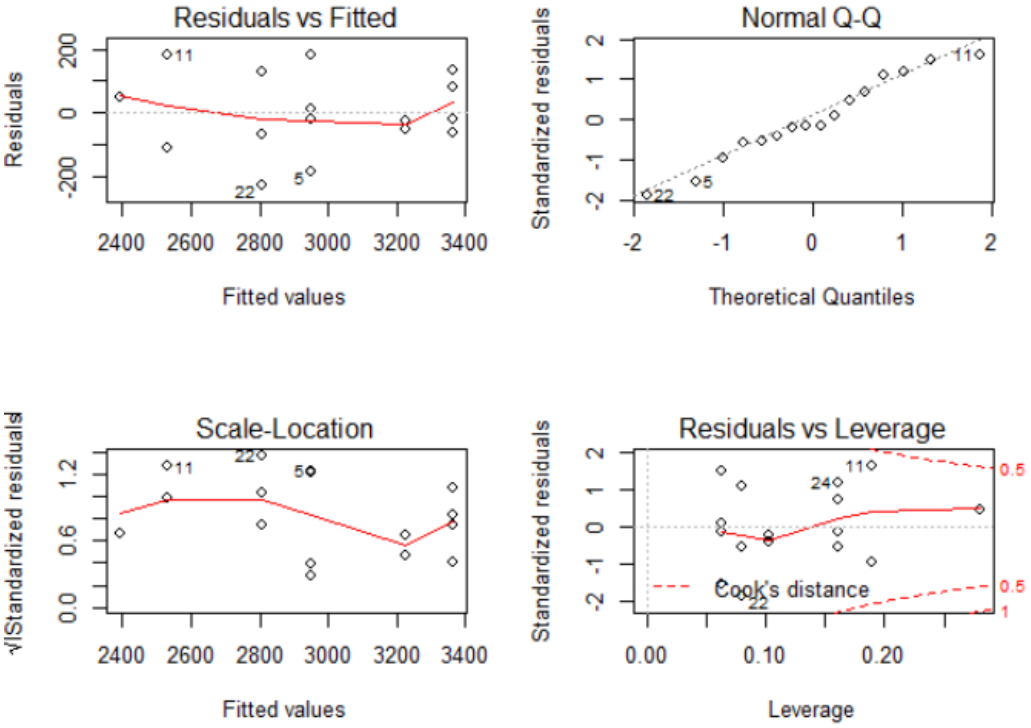
```
Call:
lm(formula = birthweight_data$Wgt ~ birthweight_data$Gest, subset = birthweight_data$Smoke ==
    "no")

Residuals:
    Min      1Q  Median      3Q     Max
-171.52 -101.59   23.28   83.63  139.48

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          -2546.14     457.29  -5.568 6.93e-05 ***
birthweight_data$Gest   147.21      11.97  12.294 6.85e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 106.9 on 14 degrees of freedom
Multiple R-squared:  0.9152,    Adjusted R-squared:  0.9092
F-statistic: 151.1 on 1 and 14 DF,  p-value: 6.852e-09
```



Teilmodell für die Gruppe der Raucherinnen:

Die Variable Gestation ist auf dem 0.01 Level signifikant und R^2 liegt bei 87.4%.

```
Call:
lm(formula = birthweight_data$Wgt ~ birthweight_data$Gest, subset = birthweight_data$Smoke ==
    "yes")

Residuals:
    Min      1Q  Median      3Q     Max
-228.53  -64.86  -19.10   93.89  184.53

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -2474.56     553.97  -4.467 0.000532 ***
birthweight_data$Gest   139.03      14.11   9.851 1.12e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 126.6 on 14 degrees of freedom
Multiple R-squared:  0.8739,    Adjusted R-squared:  0.8649
F-statistic: 97.04 on 1 and 14 DF,  p-value: 1.125e-07
```



3.d.) *Anti-Depressiva*: Use Indicator Variables

Anmerkung aus der Angabe:

"Some researchers were interested in comparing the effectiveness of three treatments for severe depression. For the sake of simplicity, we denote the three treatments A, B, and C. The researchers collected the following data (depression.txt) on a random sample of n = 36 severely depressed individuals:

- y … measure of the effectiveness of the treatment for individual i
- possible predictor age (in years) of individual
- TRT the person has received."

Modell mit Indikatorvariablen (für TRT Variable; additiv):

In diesem Modell sind *age* und der Indikator für TRT signifikant und R^2 liegt bei 75.2%.

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        36.0431     4.3678   8.252 1.57e-09 ***
antidepr_data$age   0.6659     0.0737   9.035 1.93e-10 ***
indikator_TRT      -5.1255     1.3020  -3.937 0.000403 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.374 on 33 degrees of freedom
Multiple R-squared:  0.7515,    Adjusted R-squared:  0.7364
F-statistic: 49.89 on 2 and 33 DF,  p-value: 1.057e-10
```

Modell mit Indikatorvariablen und Interaktion:

In diesem Modell sind die Interaktion zwischen *age* und Indikator und der Indikator für TRT signifikant und R^2 liegt bei 87.1%.

```
Call:
lm(formula = antidepr_data$y ~ antidepr_data$age * indikator_TRT)

Residuals:
    Min      1Q  Median      3Q     Max
-9.8834 -2.1760 -0.3219  2.9215  8.5588

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     68.65977    6.77125  10.140 1.61e-11 ***
antidepr_data$age               -0.06779    0.14473  -0.468    0.643
indikator_TRT                  -20.71270    3.00840  -6.885 8.61e-08 ***
antidepr_data$age:indikator_TRT  0.35217    0.06448   5.461 5.19e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.657 on 32 degrees of freedom
Multiple R-squared:  0.8714,    Adjusted R-squared:  0.8593
F-statistic: 72.25 on 3 and 32 DF,  p-value: 2.434e-14
```

Gesamtes Modell:

In diesem Modell ist die Variable *age* signifikant und R^2 liegt bei 63.5%.

```
Call:
lm(formula = antidepr_data$y ~ antidepr_data$age)

Residuals:
     Min       1Q   Median       3Q      Max
-15.8916  -5.7463  -0.4105   4.7013  16.4607

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      25.33935    4.08258   6.207 4.65e-07 ***
antidepr_data$age  0.67619    0.08797   7.687 6.15e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.613 on 34 degrees of freedom
Multiple R-squared:  0.6347,    Adjusted R-squared:  0.624
F-statistic: 59.08 on 1 and 34 DF,  p-value: 6.155e-09
```

TRT Gruppe A:

In diesem Modell ist die Variable *age* signifikant und R^2 liegt bei 56.5%.

```
Call:
lm(formula = antidepr_data$y ~ antidepr_data$age, subset = antidepr_data$TRT ==
    "A")

Residuals:
    Min      1Q  Median      3Q     Max
-6.4223 -2.5643  0.4802  3.4463  6.3150

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       47.51559    4.30679  11.033 6.41e-07 ***
antidepr_data$age  0.33051    0.09175   3.602  0.00483 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.419 on 10 degrees of freedom
Multiple R-squared:  0.5648,    Adjusted R-squared:  0.5212
F-statistic: 12.98 on 1 and 10 DF,  p-value: 0.00483
```

TRT Gruppe B:

In diesem Modell ist die Variable *age* signifikant und R^2 liegt bei 79%.

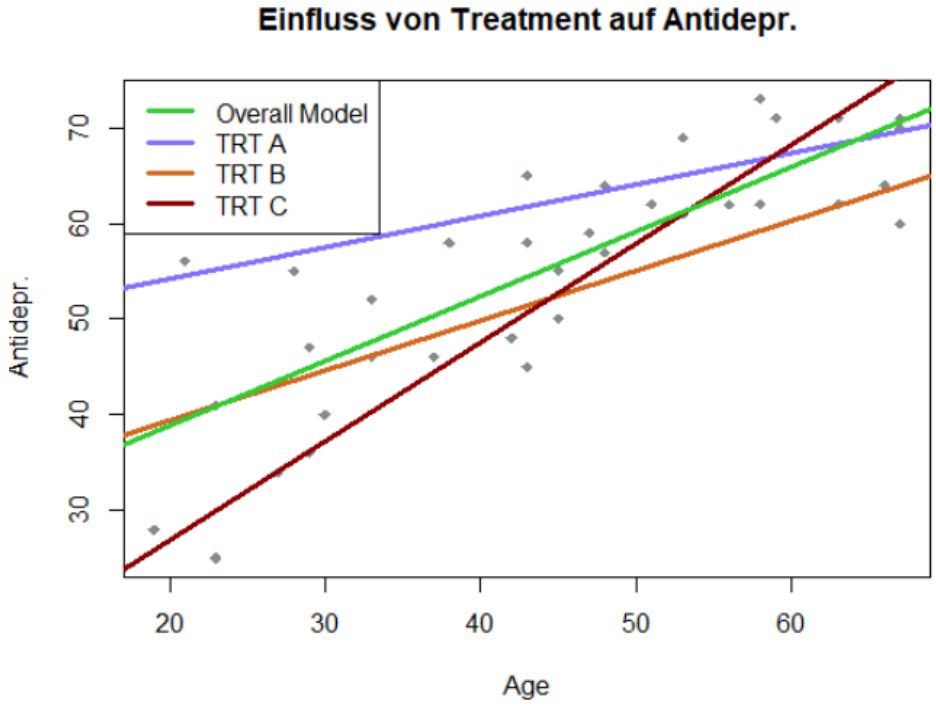
```
Call:
lm(formula = antidepr_data$y ~ antidepr_data$age, subset = antidepr_data$TRT ==
    "B")

Residuals:
    Min      1Q  Median      3Q     Max
-6.4366 -3.1860  0.2779  2.7548  6.5634

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       28.91821    3.92523   7.367  2.4e-05 ***
antidepr_data$age  0.52368    0.08539   6.133 0.000111 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.019 on 10 degrees of freedom
Multiple R-squared:   0.79,     Adjusted R-squared:  0.769
F-statistic: 37.61 on 1 and 10 DF,  p-value: 0.0001108
```

TRT Gruppe C:

In diesem Modell ist die Variable *age* signifikant und R^2 liegt bei 96.8%.

```
Call:
lm(formula = antidepr_data$y ~ antidepr_data$age, subset = antidepr_data$TRT ==
    "C")

Residuals:
    Min      1Q  Median      3Q     Max
-4.9794 -2.2394 -0.1463  2.3871  4.2192

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.21138    2.77048   2.242   0.0488 *
antidepr_data$age  1.03339    0.05982  17.275 8.94e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.246 on 10 degrees of freedom
Multiple R-squared:  0.9676,    Adjusted R-squared:  0.9643
F-statistic: 298.4 on 1 and 10 DF,  p-value: 8.94e-09
```

Überblick über die Regressionsgerade des Gesamtmodells und der einzelnen Gruppen:



R-Code zu Aufgabe 3:

```
#-----------------------------------------------------------------------
### AUFGABE 3 ### -------------------------------------------------------
#-----------------------------------------------------------------------
# Im Excel-Sheet „Some Datasets" finden Sie 5 kleine Datensätze.
# Führen Sie für die einzelnen Datensätze regressionsanalytische
# Auswertungen durch:

# a) WordRecall:      Check for Linearity

wordrecall_data <- read.xlsx(file=paste0(path,"/Some Datasets.xlsx"),
                             sheetName=1,
                             startRow=2, endRow=15,
                             colIndex=c(2:3),
                             colNames=TRUE, rowNames = FALSE)
wordrecall_data <- wordrecall_data[ ,-(3:4)]
wordrecall_data

plot(wordrecall_data$time, wordrecall_data$prop, pch=18,
     main="Word Recall", xlab="After Time in Minutes",
     ylab="% of Words Correctly Recalled")

cor(wordrecall_data$time, wordrecall_data$prop)

# lm
model_wordrecall <- lm(wordrecall_data$time ~ wordrecall_data$prop)
summary(model_wordrecall)
crPlots(model_wordrecall)

par(mfrow=c(2,2))
plot(model_wordrecall)

# piecewise linear regression
piecewise(wordrecall_data$time, wordrecall_data$prop, 3,
          xlab="% of Words Correctly Recalled",
          ylab="After Time in Minutes",
          main = "Piecewise Linear Regression")

# local regression (loess)
ggplot(wordrecall_data, aes(time, prop)) +
  stat_smooth(span=0.5,method=loess, col="chocolate") +
  geom_point() +
  ylab("prop") +
  xlab ("time") +
  ggtitle("Influence of time on prop - LM vs. LOESS") +
  stat_smooth(method=lm, se = TRUE)
```

```
# b) ShortLeaf: Check for Linearity and Influential Observations

shortleaf_data <- read.xlsx(file=paste0(path,"/Some Datasets.xlsx"),
                            sheetName=2,
                            startRow=1, endRow=71,
                            colIndex=c(1:2),
                            colNames=TRUE, rowNames = FALSE)
shortleaf_data <- shortleaf_data[ ,-(3:4)]
shortleaf_data

plot(shortleaf_data$Vol, shortleaf_data$Diam, pch=18,
     main="Shortleaf Pines", xlab="Diameter",
     ylab="Volume")

cor(shortleaf_data$Vol, shortleaf_data$Diam)

# lm
model_shortleaf <- lm(shortleaf_data$Vol ~ shortleaf_data$Diam)
summary(model_shortleaf)
crPlots(model_shortleaf)

# piecewise linear regression
piecewise(shortleaf_data$Vol, shortleaf_data$Diam, 3,
          xlab="Diameter",
          ylab="Volume",
          main = "Piecewise Linear Regression")

# local regression (loess)
ggplot(shortleaf_data, aes(Diam, Vol)) +
  stat_smooth(span=0.5,method=loess, col="chocolate") +
  geom_point() +
  ylab("Volume") +
  xlab ("Diameter") +
  ggtitle("Influence of Diam on Vol - LM vs. LOESS") +
  stat_smooth(method=lm, se = TRUE)

par(mfrow=c(2,2))
plot(model_shortleaf)
influence.measures(model_shortleaf)

par(mfrow=c(1,1))
influencePlot(model_shortleaf, id.method="identify", main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance" )


# COOK'S DISTANCE MEASURE
plot(cooks.distance(model_shortleaf), type="h",
     main="Cook's Distance")
abline(h=4/length(influences), col="indianred3", lty=5, lwd=2)

# DDFITS
influences <- lm.influence(model_shortleaf)$hat
plot(dffits(model_shortleaf), type = "h", main = "DFFITS")
abline(h=2*sqrt(length(model_shortleaf$coef)/length(influences)),
       col="indianred3", lty=5, lwd=2)
abline(h=-2*sqrt(length(model_shortleaf$coef)/length(influences)),
       col="indianred3", lty=5, lwd=2)
```

```
# c) BirthWeight: Use Indicator Variables

birthweight_data <- read.xlsx(file=paste0(path,"/Some Datasets.xlsx"),
                              sheetName=3,
                              startRow=1, endRow=33,
                              colIndex=c(1:3),
                              colNames=TRUE, rowNames = FALSE)
birthweight_data <- birthweight_data[ ,-(4:5)]
birthweight_data

# additives modell mit indikator fuer "smoke status":
I_smoke <- ifelse(birthweight_data$Smoke == "yes", 1, 0)
model_I_smoke <- lm(birthweight_data$Wgt ~ birthweight_data$Gest + I_smoke)
summary(model_I_smoke)

# modell mit interaktion zwischen weight und indikator
model_interaktion_I <- lm(birthweight_data$Wgt ~ birthweight_data$Gest*I_smoke)
summary(model_interaktion_I)

# overall LM: (ohne beachtung von smoke status)
model_birthweight <- lm(birthweight_data$Wgt ~ birthweight_data$Gest)
summary(model_birthweight)
par(mfrow=c(2,2))
plot(model_birthweight)

# mit indikatorvariablen (fuer smoke status)
model_smokeYes <- lm(birthweight_data$Wgt ~ birthweight_data$Gest,
                     subset=birthweight_data$Smoke=="yes")
summary(model_smokeYes)
par(mfrow=c(2,2))
plot(model_smokeYes)

model_smokeNo <- lm(birthweight_data$Wgt ~ birthweight_data$Gest,
                    subset=birthweight_data$Smoke=="no")
summary(model_smokeNo)
par(mfrow=c(2,2))
plot(model_smokeNo)

par(mfrow=c(1,1))
plot(birthweight_data$Wgt ~ birthweight_data$Gest,
     pch=18, col="azure4",
     main="Einfluss von Smoke Status auf Weight",
     xlab="Gestation", ylab="Weight")
abline(model_birthweight, col="lightslateblue", lwd=3)
abline(model_smokeNo, col="chocolate", lwd=3)
abline(model_smokeYes, col="darkred", lwd=3)
legend("topleft", legend=c("Smoke: no", "Smoke: yes", "Overall"),
       col=c("chocolate", "darkred", "lightslateblue"), lwd=3)


# d) Anti-Depressiva: Use Indicator Variables

antidepr_data <- read.xlsx(file=paste0(path,"/Some Datasets.xlsx"),
                           sheetName=4,
                           startRow=1, endRow=37,
                           colIndex=c(1:3),
                           colNames=TRUE, rowNames = FALSE)
antidepr_data <- antidepr_data[ ,-(4:5)]
antidepr_data
```

```
# overall LM: (ohne beachtung von smoke status)
model_antidepr <- lm(antidepr_data$y ~ antidepr_data$age)
summary(model_antidepr)
par(mfrow=c(2,2))
plot(model_antidepr)

# mit indikatorvariablen (fuer smoke status)
model_antidepr_TRTA <- lm(antidepr_data$y ~ antidepr_data$age,
                          subset=antidepr_data$TRT=="A")
summary(model_antidepr_TRTA)
par(mfrow=c(2,2))
plot(model_antidepr_TRTA)

model_antidepr_TRTB <- lm(antidepr_data$y ~ antidepr_data$age,
                          subset=antidepr_data$TRT=="B")
summary(model_antidepr_TRTB)
par(mfrow=c(2,2))
plot(model_antidepr_TRTB)

# mit indikatorvariablen (fuer smoke status)
model_antidepr_TRTA <- lm(antidepr_data$y ~ antidepr_data$age,
                          subset=antidepr_data$TRT=="A")
summary(model_antidepr_TRTA)
par(mfrow=c(2,2))
plot(model_antidepr_TRTA)

model_antidepr_TRTB <- lm(antidepr_data$y ~ antidepr_data$age,
                          subset=antidepr_data$TRT=="B")
summary(model_antidepr_TRTB)
par(mfrow=c(2,2))
plot(model_antidepr_TRTB)

model_antidepr_TRTC <- lm(antidepr_data$y ~ antidepr_data$age,
                          subset=antidepr_data$TRT=="C")
summary(model_antidepr_TRTC)
par(mfrow=c(2,2))
plot(model_antidepr_TRTC)


par(mfrow=c(1,1))
plot(antidepr_data$y ~ antidepr_data$age,
     pch=18, col="azure4",
     main="Einfluss von Treatment auf Antidepr.",
     xlab="Age", ylab="Antidepr.")
abline(model_antidepr_TRTA, col="lightslateblue", lwd=3)
abline(model_antidepr_TRTB, col="chocolate", lwd=3)
abline(model_antidepr_TRTC, col="darkred", lwd=3)
abline(model_antidepr, col="limegreen", lwd=3)
legend("topleft", legend=c("Overall Model", "TRT A", "TRT B", "TRT C"),
       col=c("limegreen", "lightslateblue", "chocolate", "darkred"),
       lwd=3)

# indikatorvariable im additiven modell
indikator_TRT <- as.numeric(antidepr_data$TRT)
model_indik_TRT <- lm(antidepr_data$y ~ antidepr_data$age + indikator_TRT)
summary(model_indik_TRT)

# indikatorvariable im modell mit interaktion
model_indik_interaktion <- lm(antidepr_data$y ~ antidepr_data$age*indikator_TRT)
summary(model_indik_interaktion)
```

Literaturquellen:

- Folien und R-Codes zu den bisher vorgetragenen Kapiteln aus UK Erweiterungen des linearen Modells (Prof. Marcus Hudec).
- Kernel Regression Examples Using np (Jeffrey Racine, McMaster University Ontario (Canada), https://socialsciences.mcmaster.ca/racinej/Gallery/Regression.html).
- R Regression Diagnostics (Vik Paruchuri, DataQuest), http://www.vikparuchuri.com/blog/r-regression-diagnostics-part-1/).