

# Hausübung 4

Reutterer Maximilian, Sattler Lukas, Weinzierl Jakob

2023-11-26

Für alle Beispiele gelten folgende Aufgabenstellungen:

- Überprüfen Sie alle erforderlichen statistischen Voraussetzungen für die Gültigkeit dieses Modells mithilfe der quality plots der Residuen und gegebenenfalls Scatterplots.
- Führen Sie eine Modellselektion durch und wählen anhand statistischer Kriterien ein optimales Modell aus. Argumentieren Sie anhand Kriterien für die Signifikanz von Koeffizienten und gegebenenfalls zusätzlich von Modellen.
- Schreiben Sie das Regressionsmodell und die angepasste Modellgleichung des optimalen Modells explizit an.
- Interpretieren Sie die Werte der Koeffizienten im Sachzusammenhang.

## Datentransformation

Wählen Sie den Datensatz UN aus der library car. Filtern Sie erst 'NA' mit der Funktion na.omit. Erklären Sie dann infant mortality durch gross domestic product. Explorieren Sie die Daten, bevor Sie ein Modell anpassen.

Folgende Voraussetzungen müssen für ein lineares Regressionmodell erfüllt sein. Das Modell hat keinen systematischen Fehler. Die Fehlervarianz ist für alle Beobachtungen gleich groß (homoskedastisch). Die Komponenten des Fehlerterms sind nicht korreliert. Der Modellfehler sei normalverteilt.

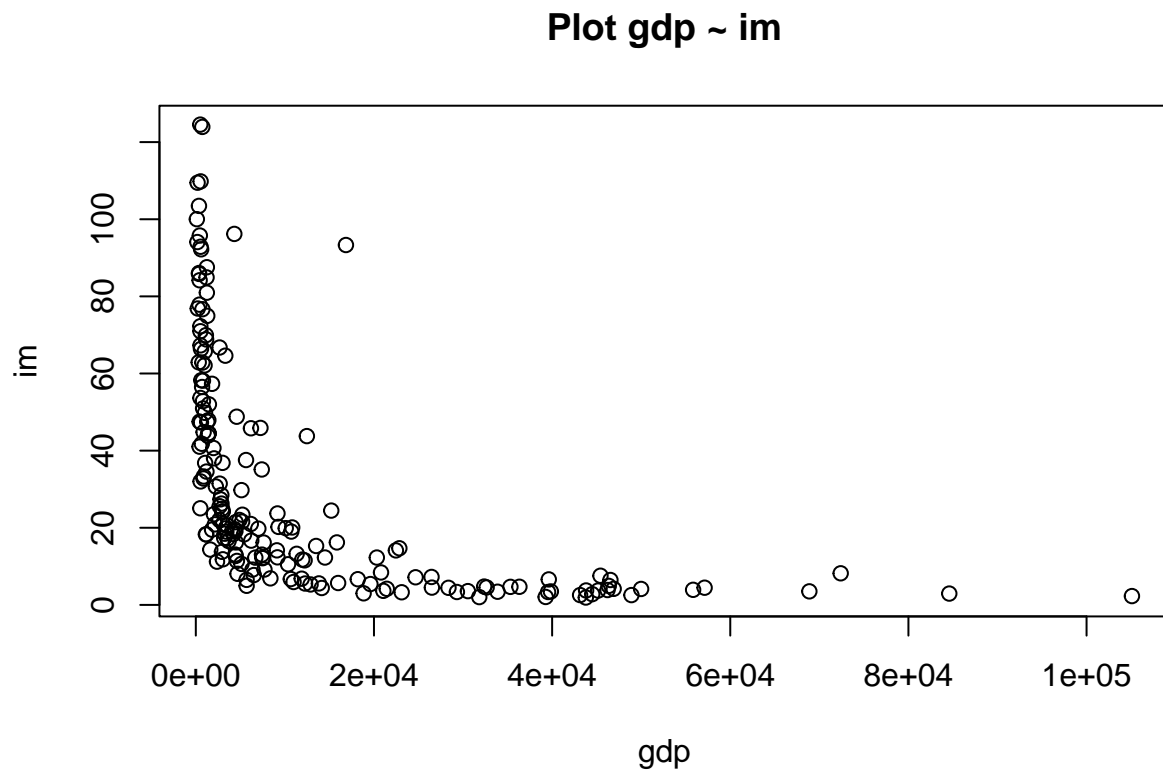
Um dies zu überprüfen, werden aus dem Datensatz "UN" zuerst alle NA-Werte verworfen, eine lineare Regression erstellt und mittels diverser Plots überprüft.

```
library(car)
summary(UN)
```

```
##           region      group      fertility      ppgdp
## Africa      :53  oecd   : 31  Min.      :1.134  Min.      : 114.8
## Asia        :50  other  :115 1st Qu.:1.754  1st Qu.: 1283.0
## Europe      :39  africa: 53  Median :2.262  Median : 4684.5
## Latin Amer:20  NA's   : 14  Mean    :2.761  Mean    :13012.0
## Caribbean  :17              3rd Qu.:3.545  3rd Qu.: 15520.5
## (Other)     :20              Max.    :6.925  Max.    :105095.4
## NA's        :14              NA's    :14    NA's    :14
##      lifeExpF      pctUrban      infantMortality
## Min.      :48.11  Min.      : 11.00  Min.      : 1.916
## 1st Qu.:65.66  1st Qu.: 39.00  1st Qu.: 7.019
## Median :75.89  Median : 59.00  Median : 19.007
## Mean    :72.29  Mean    : 57.93  Mean    : 29.440
## 3rd Qu.:79.58  3rd Qu.: 75.00  3rd Qu.: 44.477
## Max.    :87.12  Max.    :100.00  Max.    :124.535
## NA's    :14    NA's    :14    NA's     :6
```

```
df <- data.frame(UN)
invisible(na.omit(df))
```

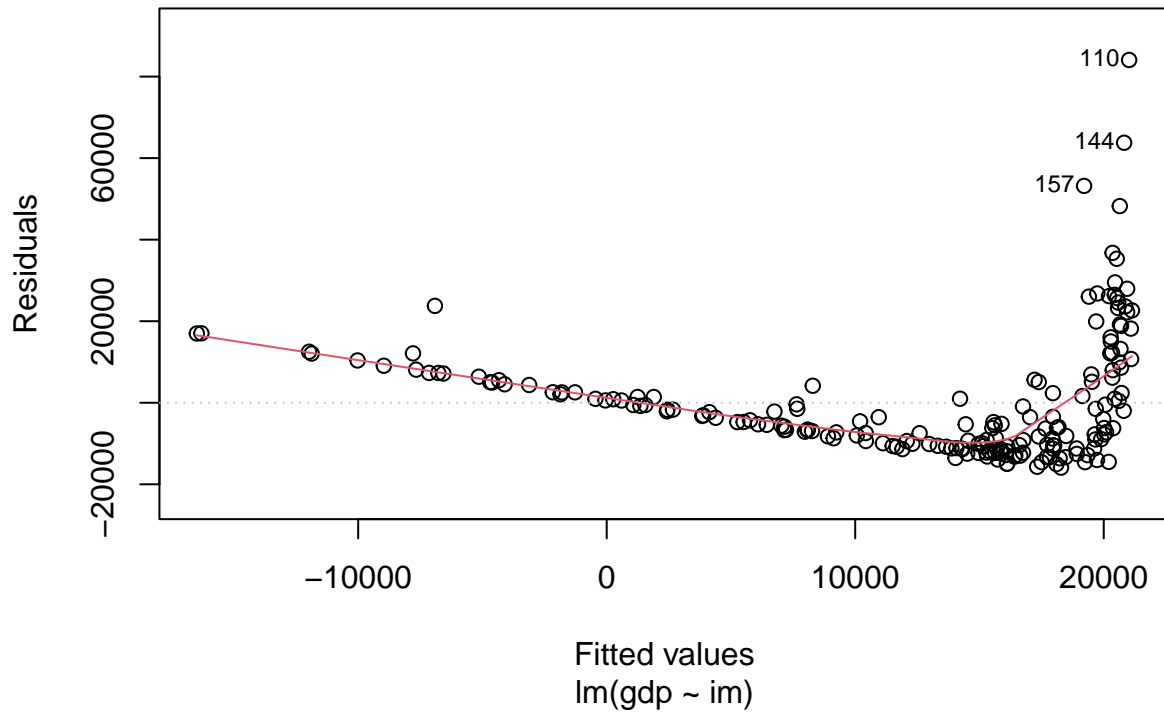
```
gdp <- df$ppgdp
im <- df$infantMortality
plot(im ~ gdp, main = "Plot gdp ~ im")
```

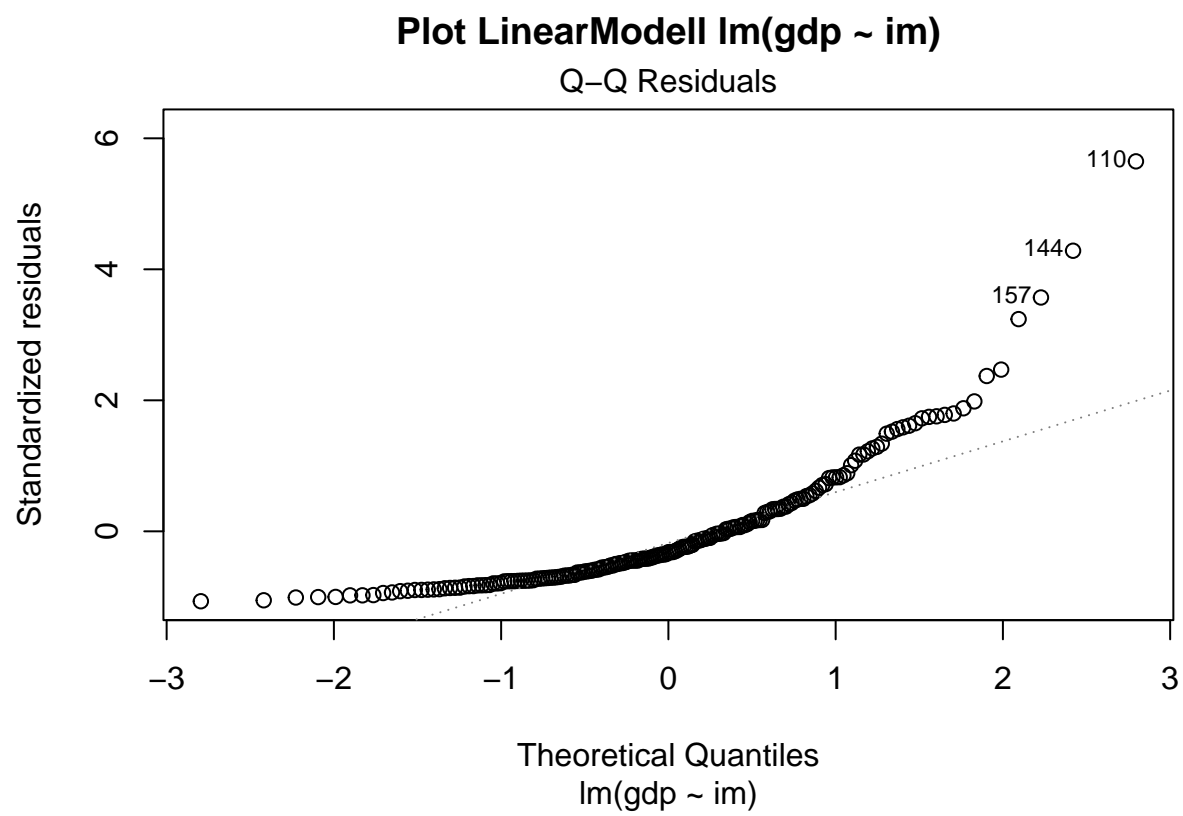


```
#boxplot(gdp~im, main = "Boxplot gdp ~ im")
fmB <- lm(gdp ~ im)
plot(lm(formula = gdp~im),main = "Plot LinearModell lm(gdp ~ im)")
```

# Plot LinearModell lm(gdp ~ im)

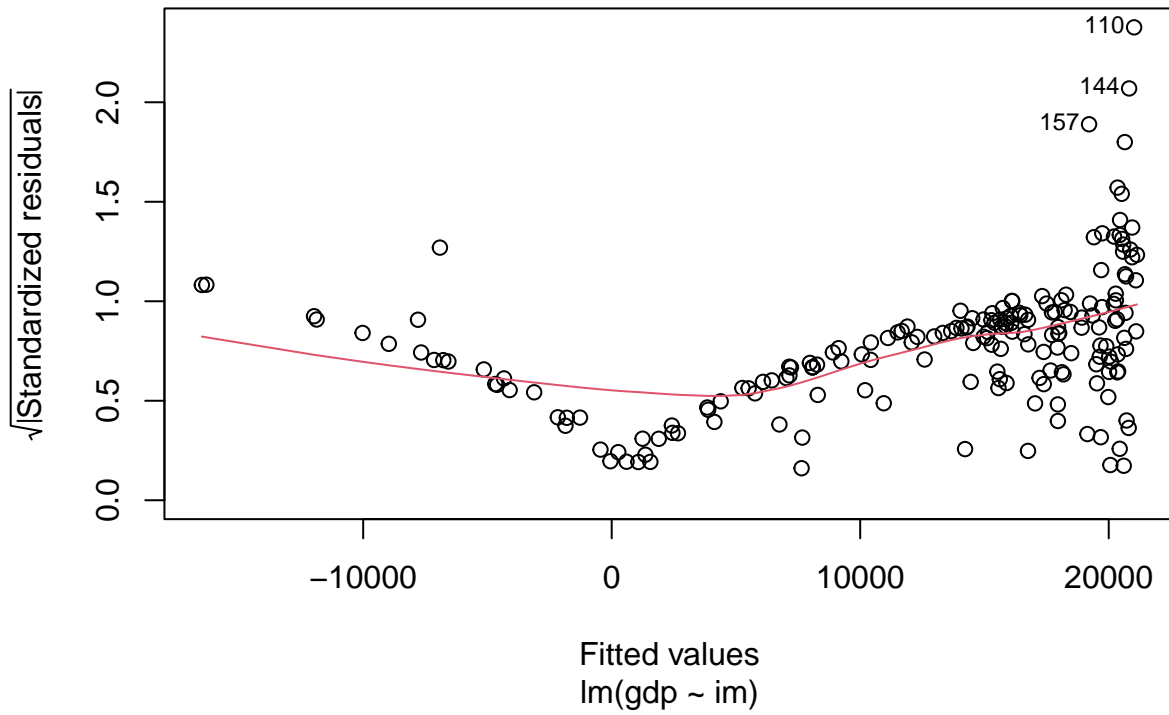
Residuals vs Fitted

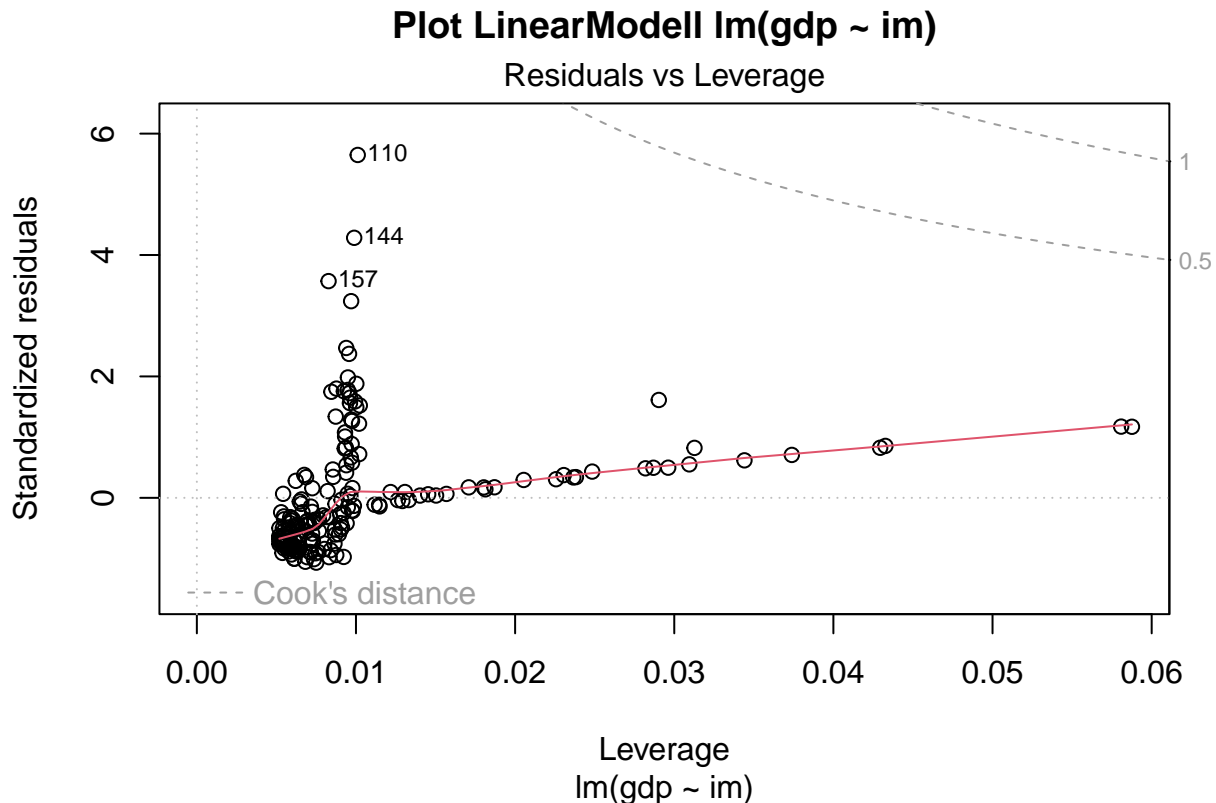




# Plot LinearModell $\ln(\text{gdp} \sim \text{im})$

Scale-Location





Auf den Plots ist zu erkennen, dass unsere Variablen diese Kriterien nicht erfüllen.

Residuals vs fitted: Überprüfung der Linearitätsannahme und Homoskedastizität (konstante Varianz der Fehler). Anhand des Modells kann man erkennen, dass die Residuen einen systematischen Fehler beinhalten. Daher ist ein lineares Modell im ersten Schritt nicht zulässig.

QQPlot: Überprüfung der Normalverteilungsannahme der Residuen. Hier kann man erkennen, dass die Residuen keiner Normalverteilung folgen, wodurch das Modell nicht gültig ist.

Scal-Location-Plot: Überprüfung der Homoskedastizität. Man kann erkennen, dass die Daten "treichterförmig auseinanderlaufen", was für eine unterschiedliche Varianz bei verschiedenen großen und geschätzten Werten bedeutet. Daher ist auch hier ein lineares Modell ungültig.

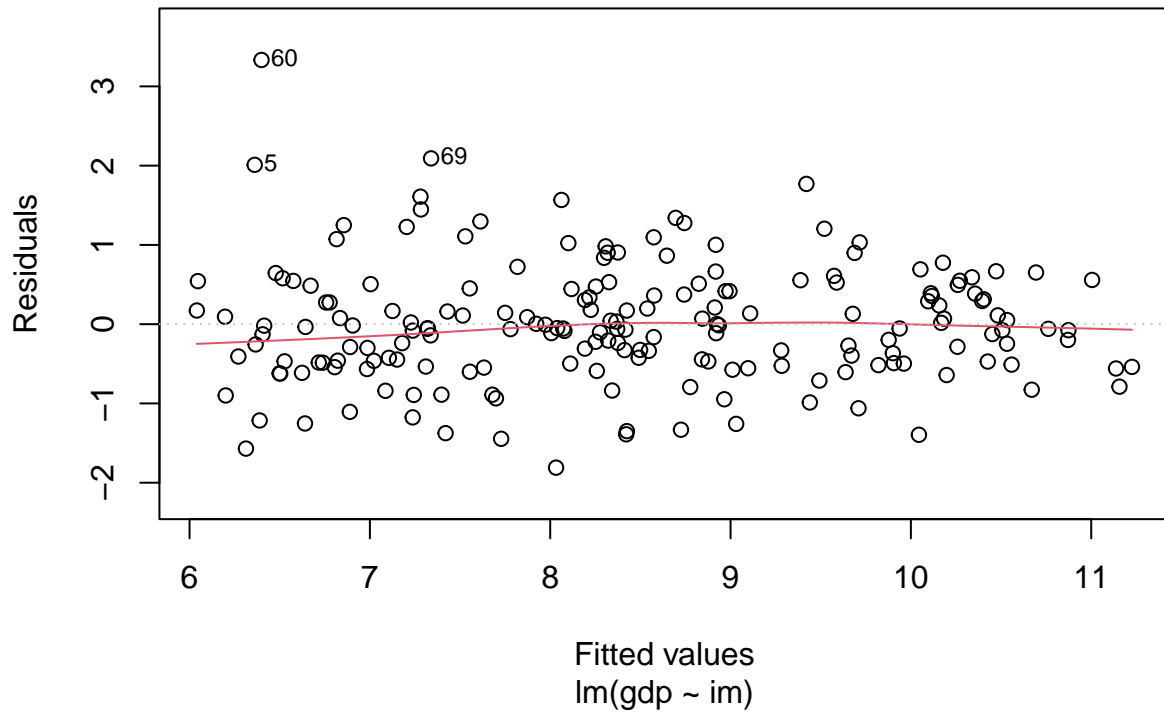
Residuals vs leverage Plot: Identifizierung von einflussreichen Datenpunkten (d.h. Punkte, die einen großen Einfluss auf die Anpassung des Modells haben). Die meisten Punkte haben eine kleine Hebelwirkung und sind konzentriert. Einige Punkte haben eine höhere Leverage und könnten daher einflussreich sein.

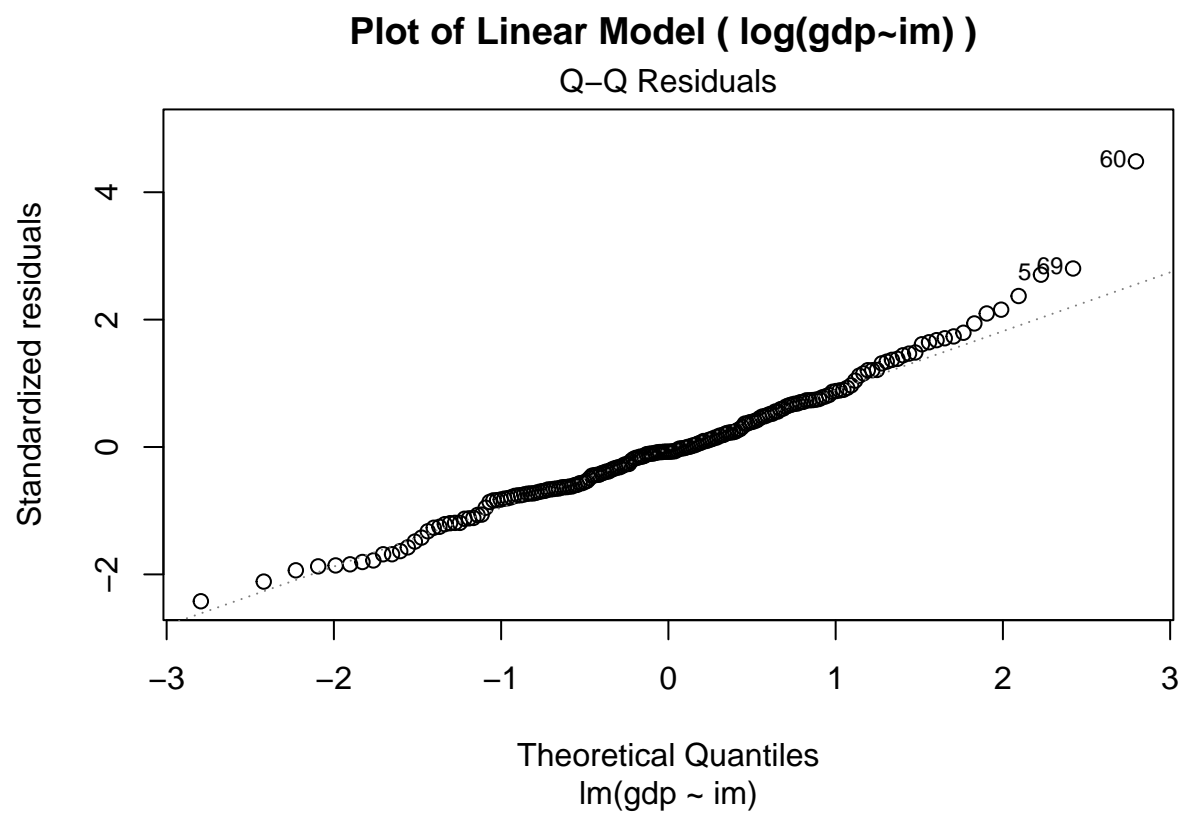
Dementsprechend werden die Daten transformiert. Auf dem Plot erkennt man, dass der Zusammenhang in etwa einer Exponentialfunktion entspricht, weswegen wir uns dazu entschieden haben, die Daten zu logarithmieren. Ausgehend davon werden dann erneut die Residuenplots analysiert.

```
gdp <- log(df$ppgdp)
im <- log(df$infantMortality)
#boxplot(gdp~im , main = "boxplot gdp~im ")
plot(lm(formula = gdp~im), main="Plot of Linear Model ( log(gdp~im) ) " )
```

# Plot of Linear Model ( $\log(\text{gdp} \sim \text{im})$ )

Residuals vs Fitted

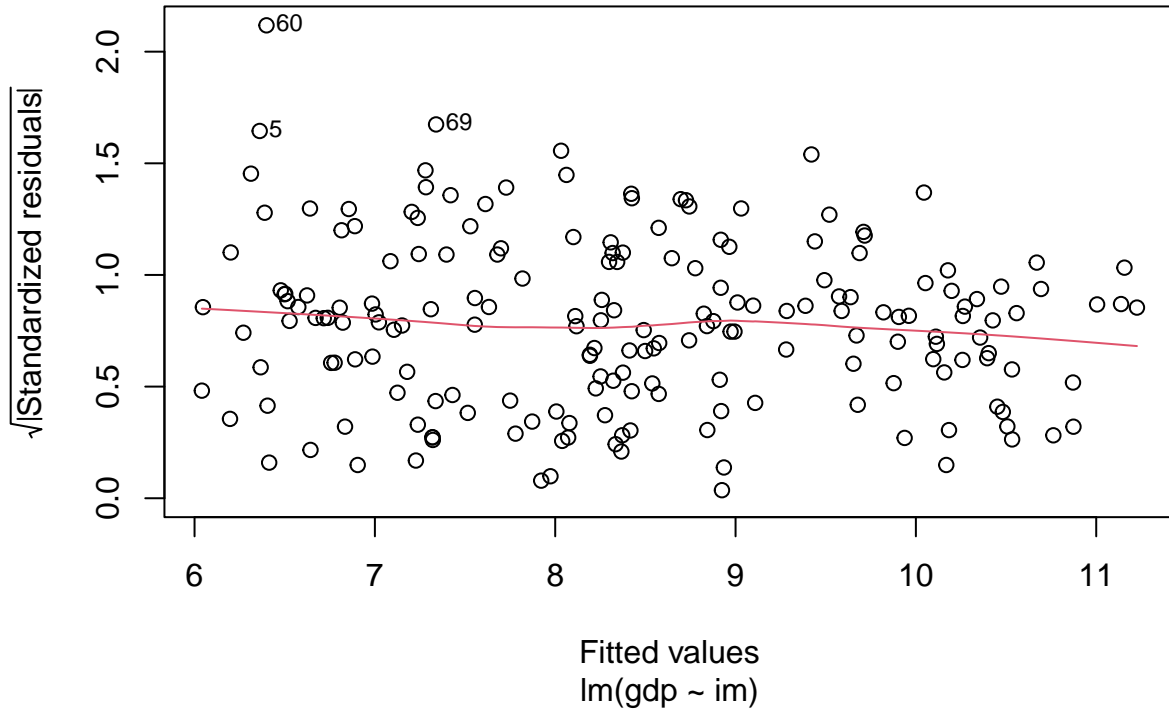


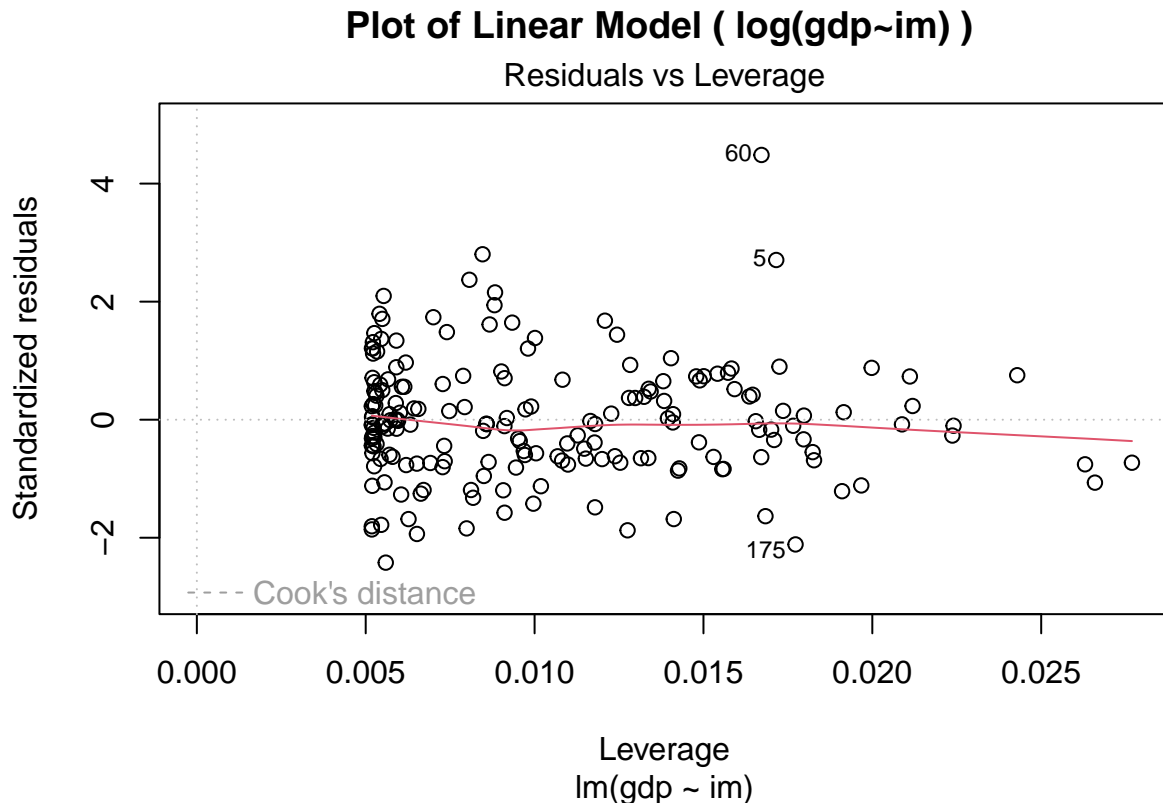




# Plot of Linear Model ( log(gdp~im) )

Scale-Location





Residuals vs fitted: Überprüfung der Linearitätsannahme und Homoskedastizität (konstante Varianz der Fehler). Das logarithmierte Modell hat keinen systematischen Fehler mehr in den Residuals und ist daher dsbbzgl. geeignet.

QQPlot: Überprüfung der Normalverteilungsannahme der Residuen. Die Residuen-Fehler approximieren in Annäherung eine Normalverteilung und ist daher dsbbzgl. geeignet.

Scal-Location-Plot: Überprüfung der Homoskedastizität. Im Scale-Location plot ist Homoskedastizität gegeben, Daher ist das Modell auch dsbbzgl. geeignet.

Residuals vs leverage Plot: Identifizierung von einflussreichen Datenpunkten (d.h. Punkte, die einen großen Einfluss auf die Anpassung des Modells haben). Die meisten Punkte sind nahe der y-Achse und haben keine große Hebelwirkung. Es sind keine Punkte ausserhalb der Cooks-Distance.

**\*\* Ergebnis \*\*** Ausgehend von den Residuenplots entschließen wir uns daher, dieses lineare Modell zu behalten, und modellieren den linearen Zusammenhang aus dem Output der Summarys des Modells. Durch den Output der Summarys erhalten wir folgende Kennzahlen zu dem Modell:

```
summary(lm(formula = gdp~im))
```

```
##
## Call:
## lm(formula = gdp ~ im)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8099 -0.4860 -0.0552  0.4436  3.3333
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.03359    0.15445   77.91  <2e-16 ***
## im          -1.24220    0.04965  -25.02  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7495 on 191 degrees of freedom
## (20 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.7662, Adjusted R-squared:  0.765
## F-statistic: 625.9 on 1 and 191 DF,  p-value: < 2.2e-16
```

```
fmA <- lm(gdp ~ im)
summary(fmA)
```

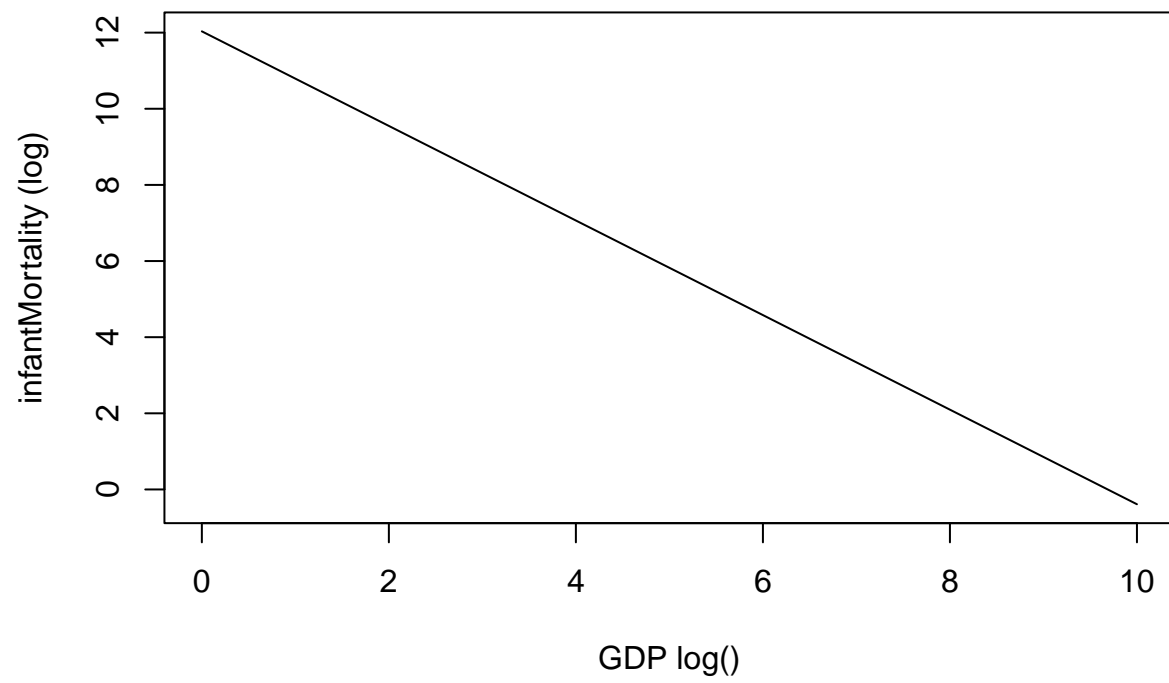
```
##
## Call:
## lm(formula = gdp ~ im)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8099 -0.4860 -0.0552  0.4436  3.3333
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.03359    0.15445   77.91  <2e-16 ***
## im          -1.24220    0.04965  -25.02  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7495 on 191 degrees of freedom
## (20 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.7662, Adjusted R-squared:  0.765
## F-statistic: 625.9 on 1 and 191 DF,  p-value: < 2.2e-16
```

Der Mittlere Fehler der Residuen beträgt -0.055. Für den Intercept bzw. Alpha wird ein Wert von 12,033 mit einem p-Wert von <2e-16 vorgeschlagen. Dieser p-Wert ist hochsignifikant. Für das Beta (Steigungskoeffizienten) wird ein Wert von -1.24 mit einem p-Wert von <2e-16 vorgeschlagen. Damit ist auch dieser Wert hochsignifikant.

Damit erhalten wir die Parameter für die Modellierung eines linearen Zusammenhangs. Mit diesem Modell können 76,5% der Varianz (R-Squared) erklärt werden. Durch das Logarithmieren haben wir ein robustes Modell erhalten, um den Zusammenhang zwischen GDP und Infant Mortality linear darstellen zu können.

$$y(i) = \alpha + \beta * x$$

```
curve(12.03359 -1.24220 * x, from =0, to = 10, n=40, xlab = "GDP log()", ylab = "infantMortality (log)")
```



## Schweiz

Wir kehren zurück zu den Variablen “Fertility”, “Agriculture”, “Education”, “Catholic” und “Infant. Mortality” aus dem R Datensatz `swiss` des R package `utils`. Passen Sie für die oben genannten Variablen ein Modell an, das Education durch die übrigen Variablen erklärt, soweit dies zulässig ist.

```
library(ggplot2)
library(GGally)
```

```
Swiss <- swiss
str(Swiss)
```

```
## 'data.frame':  47 obs. of  6 variables:
## $ Fertility      : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
## $ Agriculture    : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
## $ Examination    : int   15 6 5 12 17 9 16 14 12 16 ...
## $ Education      : int   12 9 5 7 15 7 7 8 7 13 ...
## $ Catholic       : num   9.96 84.84 93.4 33.77 5.16 ...
## $ Infant.Mortality: num   22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
```

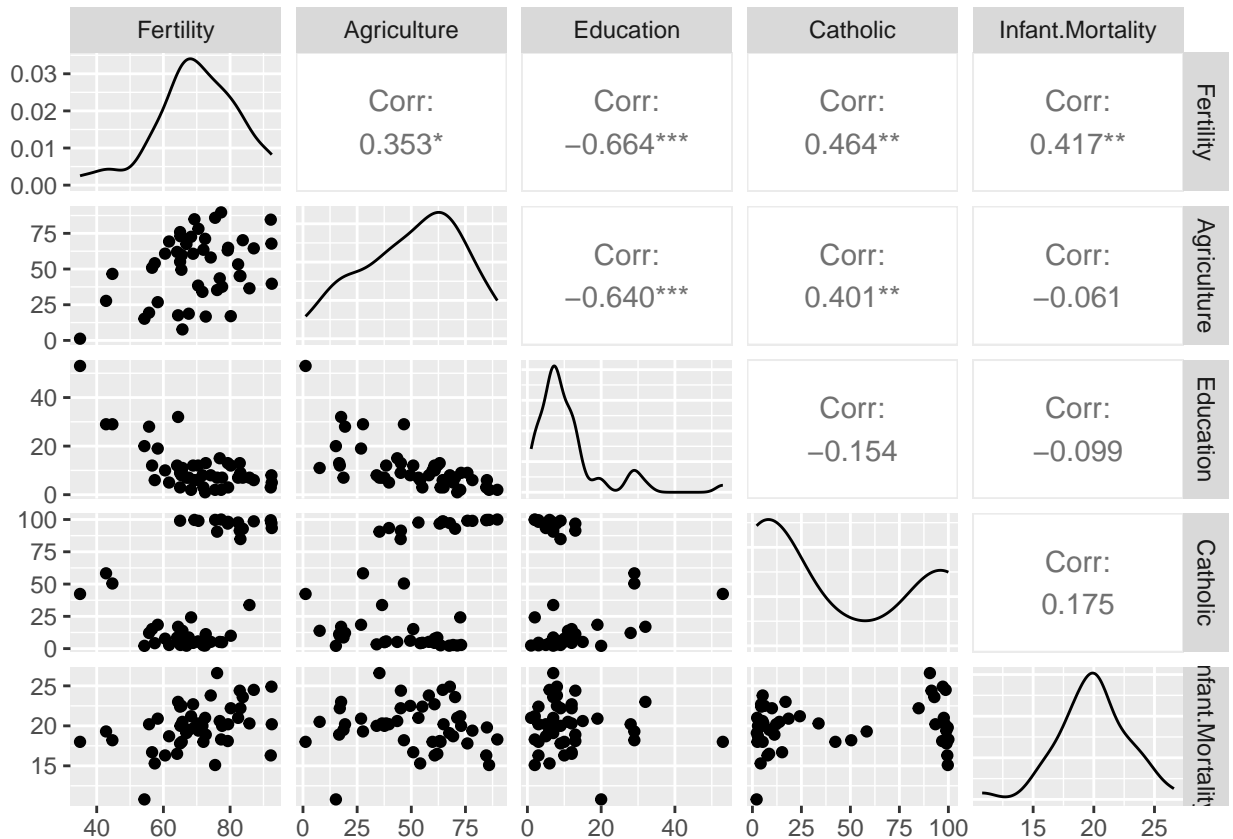
```
summary(Swiss)
```

```
##      Fertility      Agriculture      Examination      Education
## Min.   :35.00    Min.   : 1.20    Min.   : 3.00    Min.   : 1.00
## 1st Qu.:64.70    1st Qu.:35.90    1st Qu.:12.00   1st Qu.: 6.00
## Median :70.40    Median :54.10    Median :16.00   Median : 8.00
## Mean   :70.14    Mean   :50.66    Mean   :16.49   Mean   :10.98
## 3rd Qu.:78.45    3rd Qu.:67.65    3rd Qu.:22.00   3rd Qu.:12.00
## Max.   :92.50    Max.   :89.70    Max.   :37.00   Max.   :53.00
##      Catholic      Infant.Mortality
## Min.   : 2.150    Min.   :10.80
## 1st Qu.: 5.195    1st Qu.:18.15
## Median :15.140    Median :20.00
## Mean   :41.144    Mean   :19.94
## 3rd Qu.:93.125    3rd Qu.:21.70
## Max.   :100.000    Max.   :26.60
```

```
Swiss <- Swiss[,-3] #remove "Examination"
str(Swiss)
```

```
## 'data.frame':  47 obs. of  5 variables:
## $ Fertility      : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...
## $ Agriculture    : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...
## $ Education      : int   12 9 5 7 15 7 7 8 7 13 ...
## $ Catholic       : num   9.96 84.84 93.4 33.77 5.16 ...
## $ Infant.Mortality: num   22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
```

```
ggpairs(Swiss)
```



## Überprüfung der Koeffizienten mittels Scatterplot Matrix.

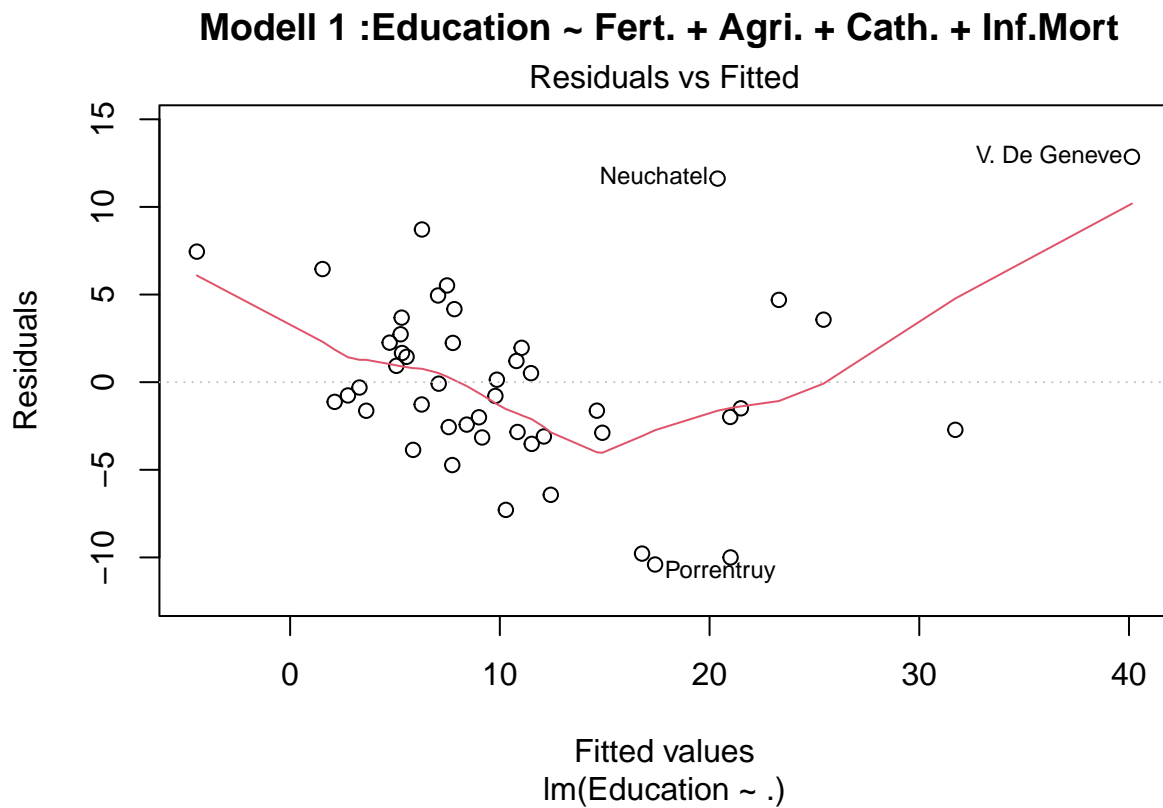
In der Scatterplot-Matrix zeigt sich, dass Education stark signifikant mit Fertility und Agriculture negativ korreliert ist. Bei Catholic und infant mortality lässt sich keine Korrelation feststellen. Es wird auch die Korrelation der Regressoren überprüft. Hier sind keine Werte über 0,5 feststellbar, weshalb wir initial alle Parameter als Regressoren modellieren und anschließend die Residuenplots überprüfen.

```
education_lm1 <- lm(Education ~ ., data=Swiss)
summary(education_lm1)
```

```
##
## Call:
## lm(formula = Education ~ ., data = Swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4029  -2.7803  -0.7571   2.4934  12.8590
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   49.99303     6.18641   8.081 4.31e-10 ***
## Fertility     -0.52070     0.07869  -6.617 5.14e-08 ***
## Agriculture   -0.22880     0.03906  -5.857 6.37e-07 ***
## Catholic       0.08333     0.02179   3.825 0.000428 ***
## Infant.Mortality 0.28437     0.30040   0.947 0.349243
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

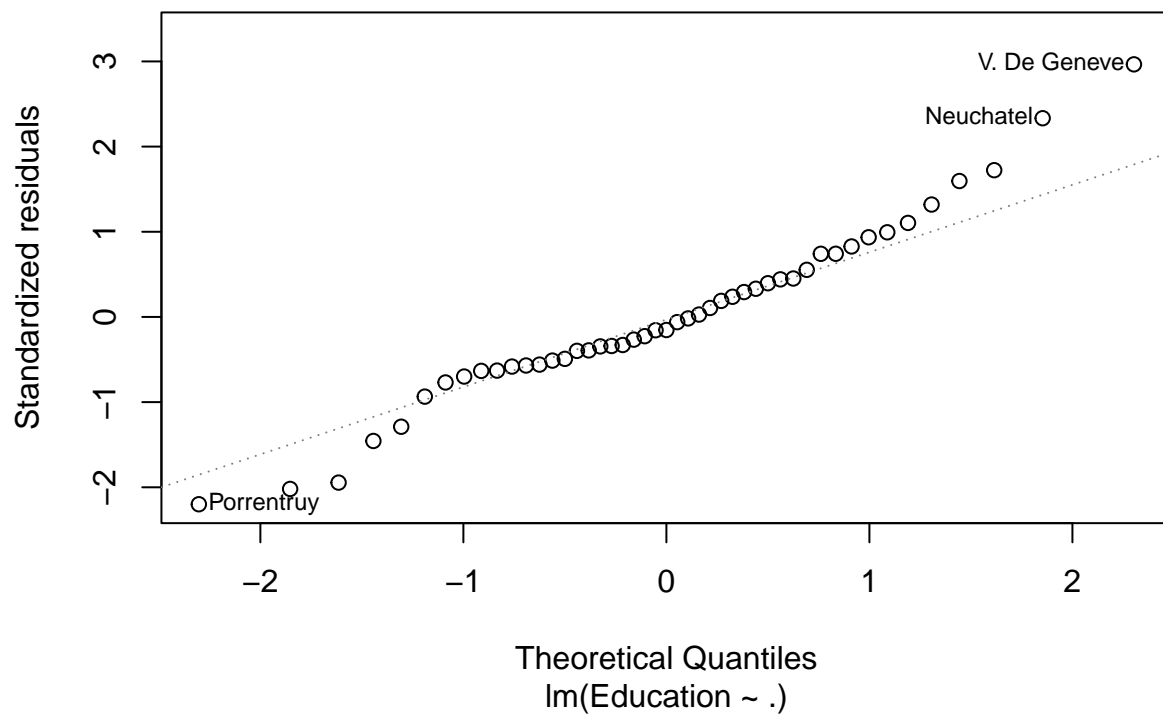
```
##
## Residual standard error: 5.224 on 42 degrees of freedom
## Multiple R-squared:  0.7305, Adjusted R-squared:  0.7048
## F-statistic: 28.46 on 4 and 42 DF,  p-value: 1.804e-11
```

```
plot(education_lm1 , main = "Modell 1 :Education ~ Fert. + Agri. + Cath. + Inf.Mort ")
```



# Modell 1 :Education ~ Fert. + Agri. + Cath. + Inf.Mort

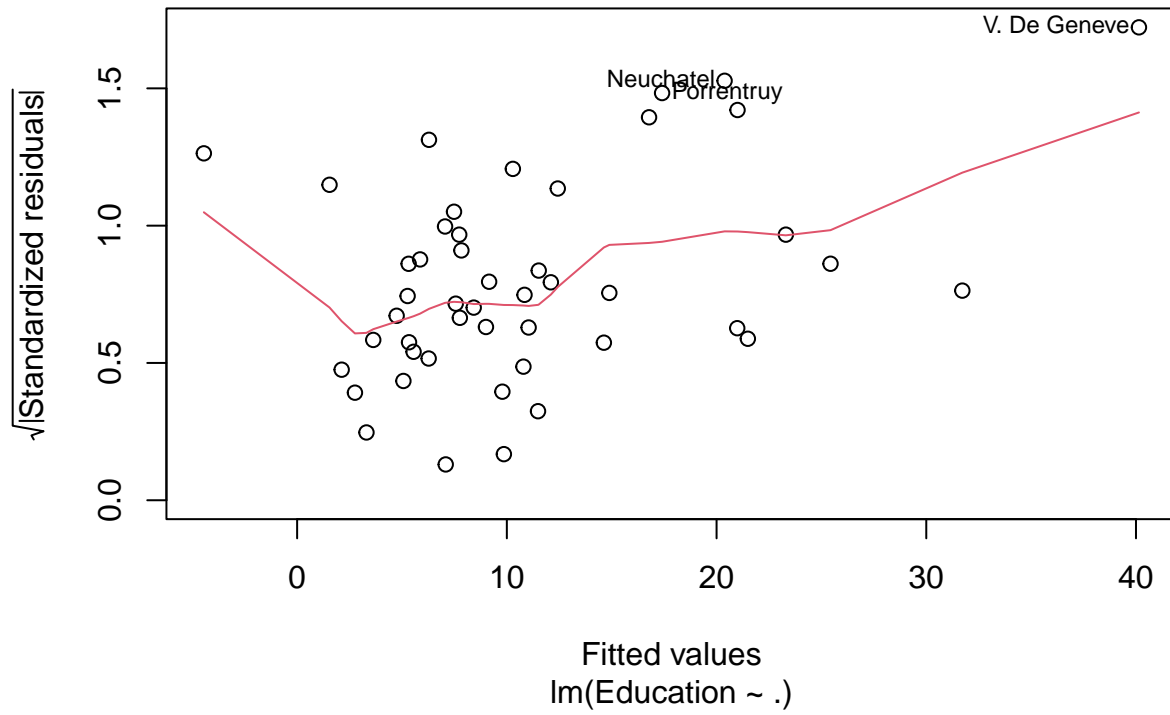
Q-Q Residuals

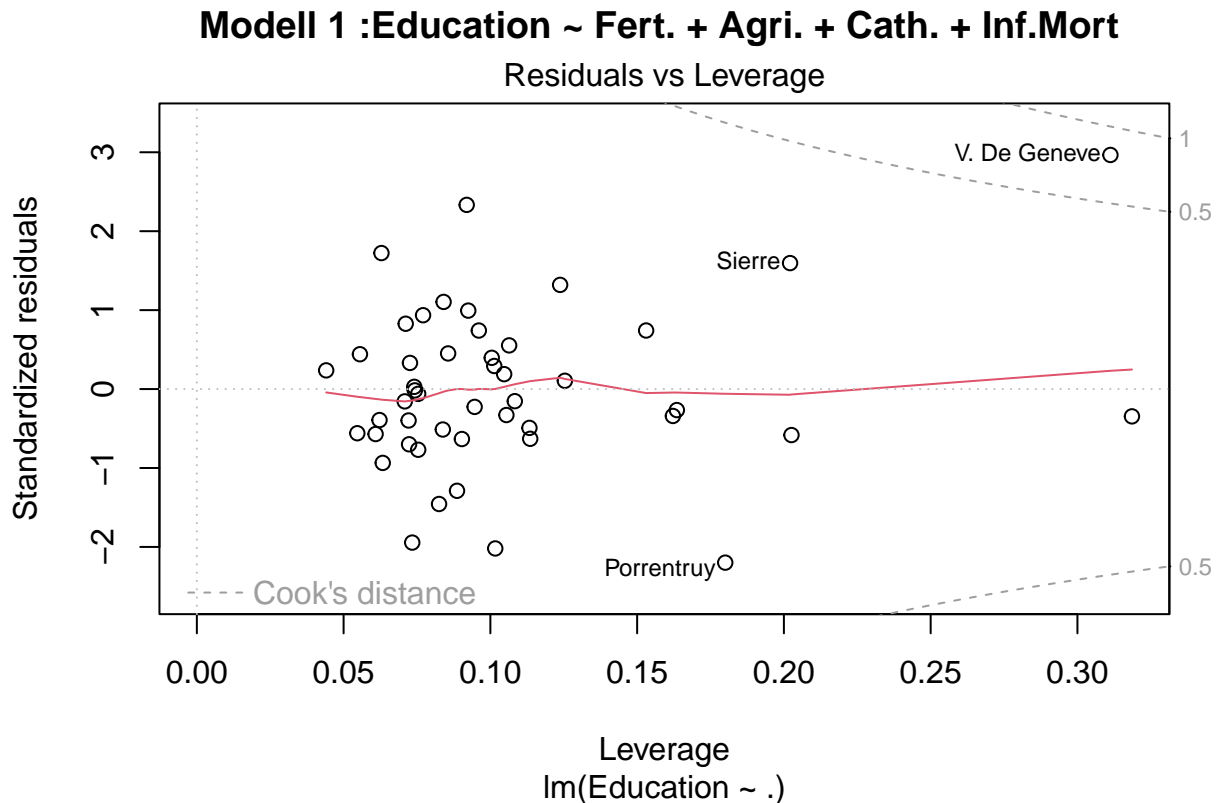




# Modell 1 :Education ~ Fert. + Agri. + Cath. + Inf.Mort

Scale-Location





##Überprüfung der Residuen

Residuals vs fitted: Überprüfung der Linearitätsannahme und Homoskedastizität (konstante Varianz der Fehler). Das Modell hat keinen systematischen Fehler mehr in den Residuals und ist daher dsbbzgl. geeignet. "V. De Geneve" ist als Ausreißer erkennbar

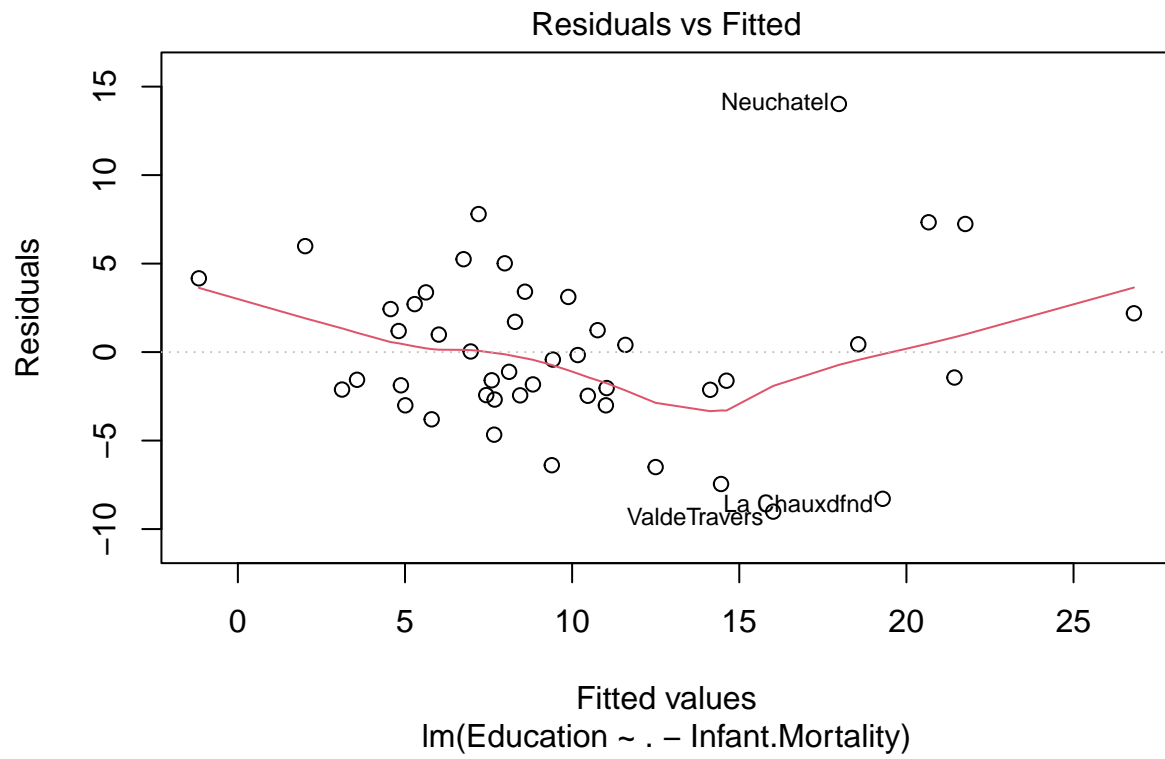
QQPlot: Überprüfung der Normalverteilungsannahme der Residuen. Die Residuen-Fehler approximieren in Annäherung eine Normalverteilung und ist daher dsbbzgl. geeignet. "V. De Geneve" ist als Ausreißer erkennbar

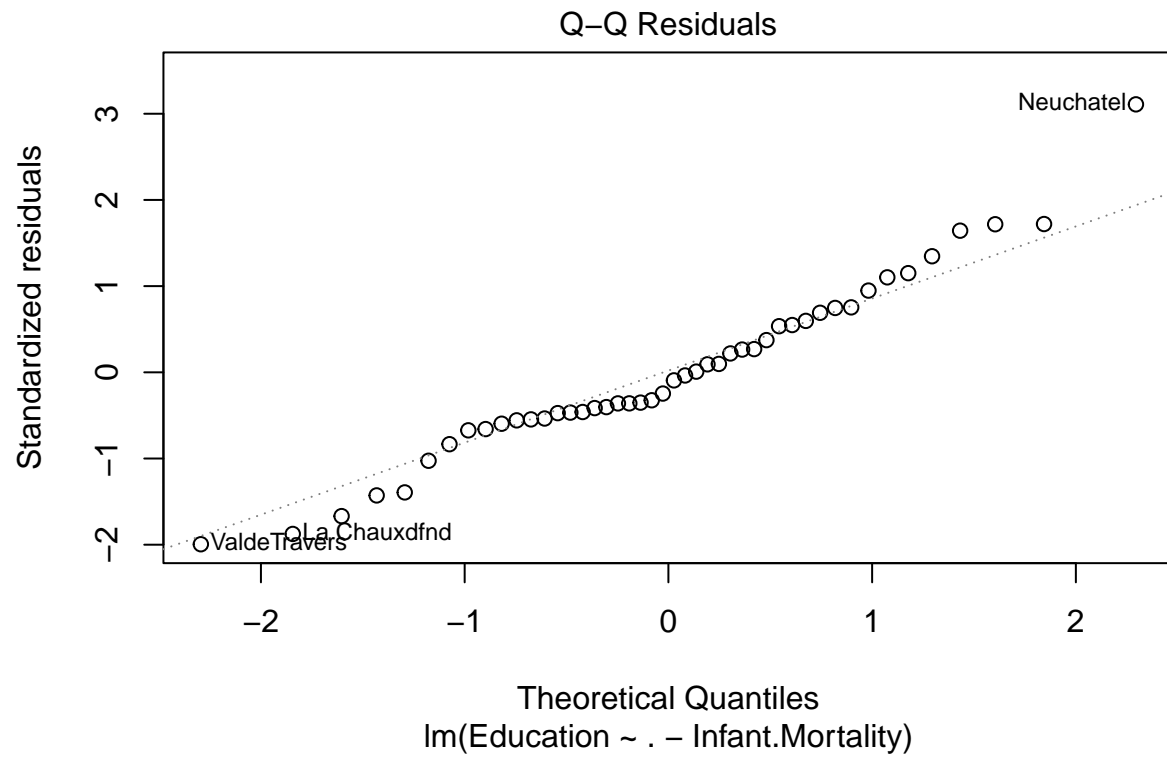
Scal-Location-Plot: Überprüfung der Homoskedastizität. Im Scale-Location plot ist Homoskedastizität gegeben. Es liegt eine Punktwolke vor. Daher ist das Modell auch dsbbzgl. geeignet.

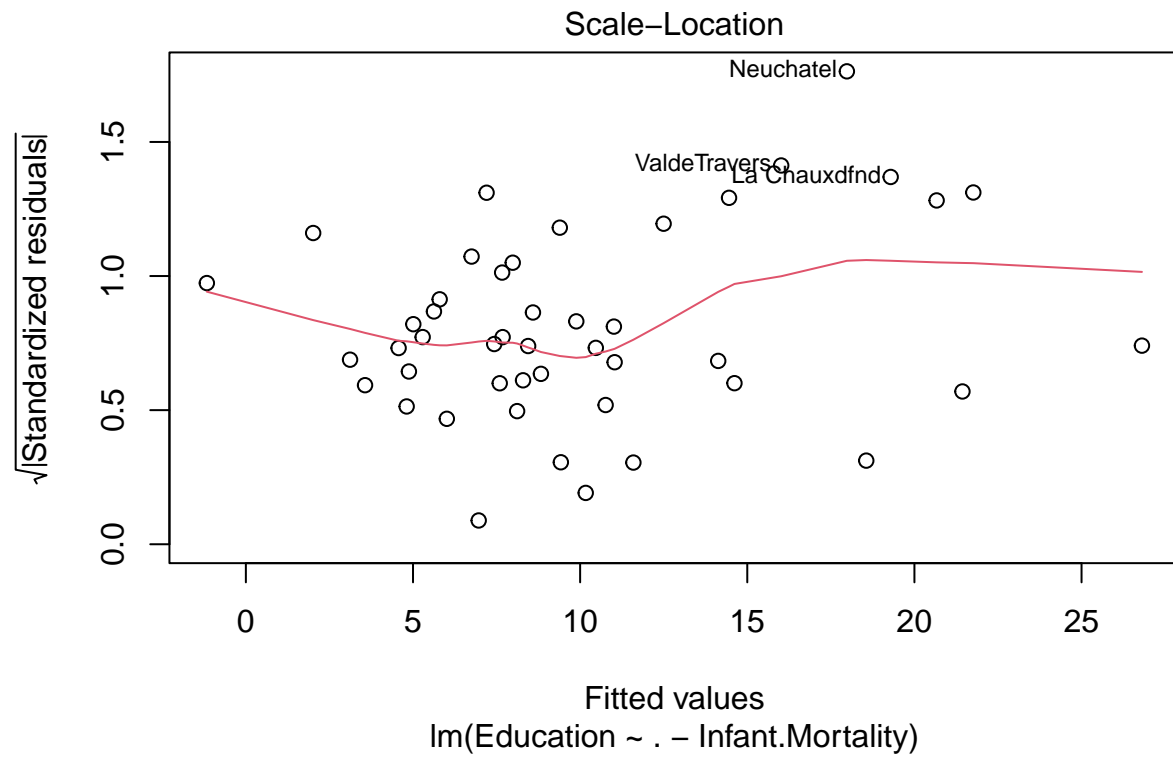
Residuals vs leverage Plot: Identifizierung von einflussreichen Datenpunkten (d.h. Punkte, die einen großen Einfluss auf die Anpassung des Modells haben). Die meisten Punkte sind nahe der y-Achse und haben keine große Hebelwirkung. "V. De Geneve" liegt ausserhalb der Cooks-Distance.

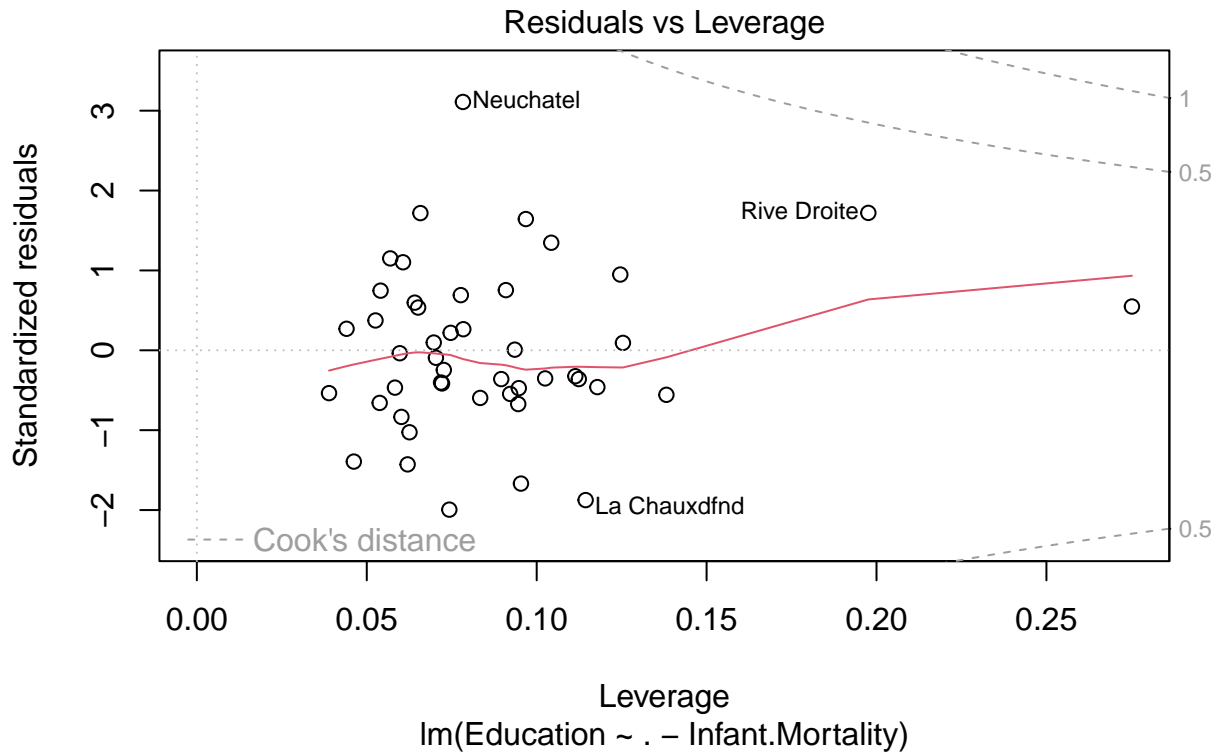
Aus der Summary erkennt man, dass alle Regressoren ausser Infant Mortality signifikant sind. Der Punkt "V. De Geneve" ist in der Residuenplot als Ausreißer erkennbar und wird daher entfernt. Infant Mortality wird ebenfalls als Regressor entfernt.

```
education_lm2 <- lm(Education ~ . - Infant.Mortality, data=Swiss[-c(45),])
plot(education_lm2)
```









```
summary(education_lm2)
```

```
##
## Call:
## lm(formula = Education ~ . - Infant.Mortality, data = Swiss[-c(45),
##    ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.0144 -2.4407 -0.7688  2.6409 14.0202
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  45.26863    4.91660   9.207 1.24e-11 ***
## Fertility    -0.38468    0.07120  -5.403 2.86e-06 ***
## Agriculture  -0.20240    0.03566  -5.676 1.16e-06 ***
## Catholic     0.06188    0.02070   2.989 0.00466 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.697 on 42 degrees of freedom
## Multiple R-squared:  0.6216, Adjusted R-squared:  0.5945
## F-statistic: 22.99 on 3 and 42 DF,  p-value: 5.776e-09
```

Wir bekommen nun mit summary die Multiple Regression (ohne Infant.Mortality und "V. De Geneve"):

Das  $R^2$  ist 62.16% und damit niedriger als jenes der ersten Anpassung, bei der Infant Mortality und “V. De Geneve” nicht entfernt wurden. Wir wählen daher `education_lm1` als Modell aus, welches mit 73.05% eine bessere Modellanpassung hat.

Das bedeutet unser Modell ist:  $\text{Education} = 49.99 + (-0.52) * \text{Fertility} + (-0.23) * \text{Agriculture} + 0.08 * \text{Catholic} + 0.28 * \text{Infant.Mortality}$  mit  $R^2 = 73.05\%$

Die Aussage dieser Formel ist, dass die Education (% der Wehrpflichtigen mit mehr als Grundschulausbildung) theoretisch bei 49.99% läge, wenn alle anderen Werte Null sind (= Intercept bzw.  $\alpha$ ). *##Interpretation*

Die Werte für Katholizismus und Säuglingssterblichkeit spielen eine untergeordnete Rolle. Es zeigt sich daher, man über die Geburtenrate und die Beschäftigungsrate in der Landwirtschaft Vorhersagen zur formalen Ausbildung der männlichen Wehrpflichtigen in der Schweiz im Untersuchungszeitraum treffen. Höhere Beschäftigung in der Landwirtschaft geht meist mit einem geringeren Urbanisierungsgrad und einer geringeren Entwicklung einer Region einher. Auch in anderen Ländern zeigt sich, dass die Geburtenrate negativ mit dem Einkommen korreliert ist und das Einkommen positiv mit der Ausbildung.

*#Ergänzung SS24*

Ausgehend von unserem einfachen linearen Regressionsmodell bilden wir mittels Ridge Regression und LASSO ebenfalls lineare Regressionsmodelle. To find a better linear model for our data, we can use Ridge Regression or LASSO. Ridge regression works by penalizing the sum of squared coefficients. This also means, the coefficients will never be fully discarded, as they won't reach zero.

LASSO uses absolute values of the coefficients to penalize them, so the values will reach zero fast(er) and then be discarded – this means, it performs feature selection.

In general, RR is used for multicollinear Data, when you want to keep all regressors, while LASSO is used when you want to perform feature selection and only keep relevant coefficients.

Cross validation is then used, to split the data into multiple chunks, and then iteratively, some chunks will be used to train the model, while others to test the trained model. The variable which is “trained” is  $\lambda$ , a factor for the penalty. We supply a range of numbers to  $\lambda$  to find the optimum.

In the end, a linear model of the parameters is received.

## Load Libraries and Datasets

```
library(glmnet)
library(dplyr)

swiss_df <- swiss
lambda.grid <- 10^seq(8, -4, length=100) # 10^10 bis 10^-2 in 100 stufen
```

## Data Preparation

```
# Data prep USA
#unabhängige variablen
X <- swiss_df %>% dplyr::select(-Examination, -Education) %>% as.matrix()
#abhängige variable
Y <- swiss_df %>% dplyr::select(Education) %>% as.matrix()

head(X)
```

	Fertility	Agriculture	Catholic	Infant.Mortality
## Courtelary	80.2	17.0	9.96	22.2
## Delemont	83.1	45.1	84.84	22.2
## Franches-Mnt	92.5	39.7	93.40	20.2
## Moutier	85.8	36.5	33.77	20.3
## Neuveville	76.9	43.5	5.16	20.6
## Porrentruy	76.1	35.3	90.57	26.6

```
head(Y)
```

	Education
## Courtelary	12
## Delemont	9
## Franches-Mnt	5
## Moutier	7
## Neuveville	15
## Porrentruy	7

```
fitRR <- glmnet::glmnet(x=X, y=Y, alpha = 0, lambda = lambda.grid) #alpha = zero->ridge regression
dim(coef(fitRR)) #zur kontrolle, zeigt welche dimensions (inkl. intercept) wir haben
```

```
## [1] 5 100
```

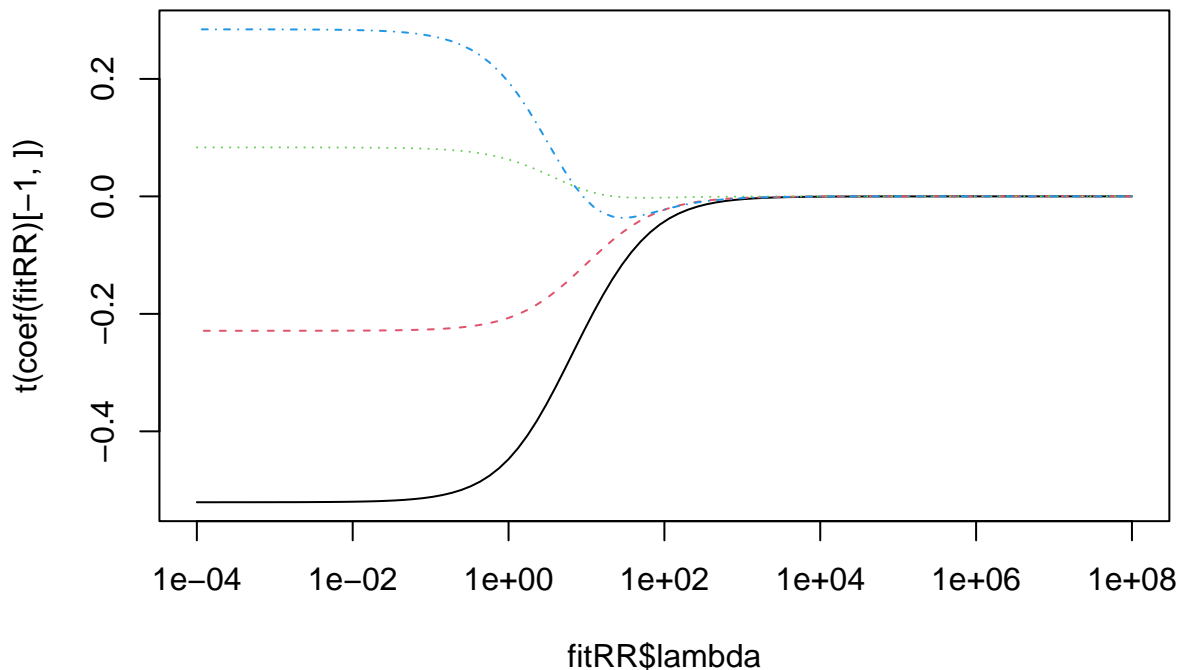
We have 5 coefficients (incl. intercept) for 100 different values of lambda.

**Daten Plotten:**

```
matplot(fitRR$lambda, t(coef(fitRR)[-1, ]), type="l", log="x", main="Ridge Regression Switzerland")
```



## Ridge Regression Switzerland

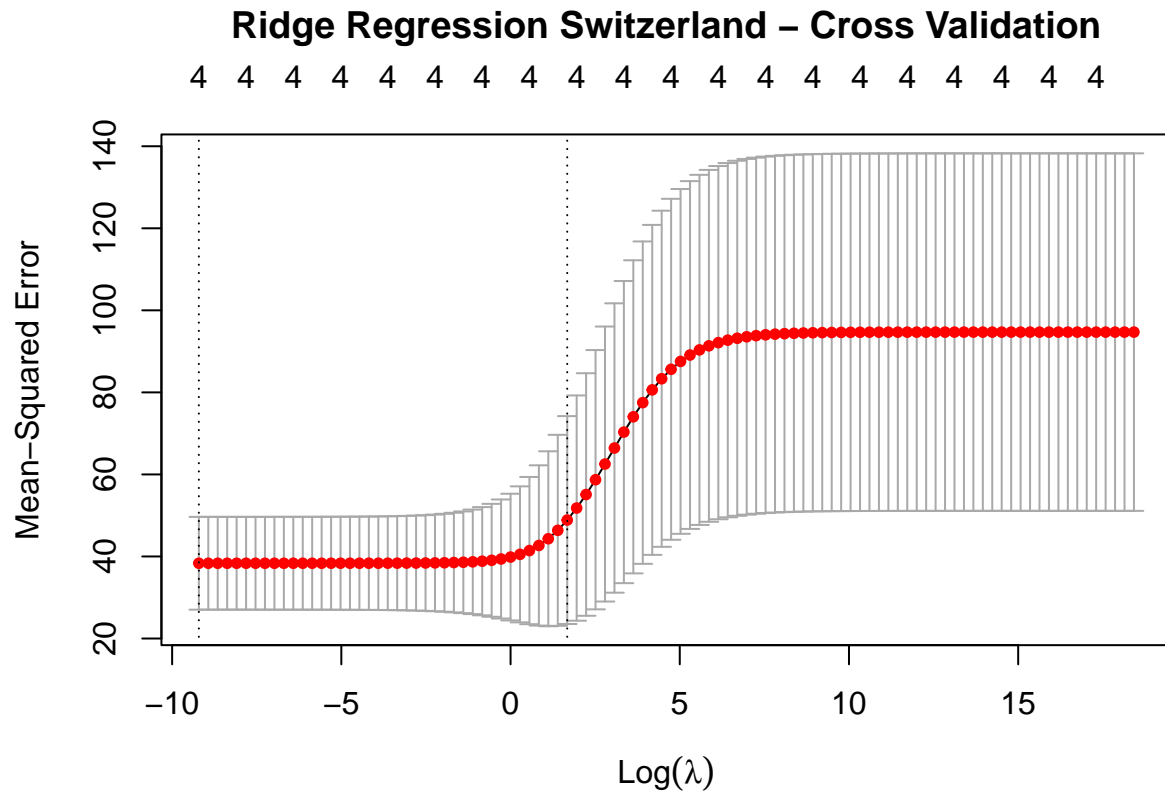


Auf der Abszisse sind die Werte für Lambda von  $10^{-2}$  bis  $10^{10}$  eingetragen, auf der Ordinate die Werte der verschiedenen Koeffizienten (ausg. der Achsenabschnitt/Intercept). Deutlich sichtbar konvergieren die Werte gegen Null, die Annäherung verlangsamt sich beim annähern an Null jedoch und wird nie erreicht, da bei Ridge Regression der Koeffizient quadriert wird.

### Cross Validation

```
fitRR_cv <- cv.glmnet(x=X, y=Y, alpha = 0, lambda = lambda.grid)
fitRR_cv_min <- fitRR_cv$lambda.min
fitRR_cv_1se <- fitRR_cv$lambda.1se

#plotting
plot(fitRR_cv, type="l")
title("Ridge Regression Switzerland - Cross Validation", line=2.5)
```



In the row above the graph, it shows the number of regressors that are kept. So as can be seen here, none of them are discarded, which is typical of ridge regression. The red dots/line shows the Mean Squared Error when applying the training data results on the test data. Also the Standard Deviation is shown. The left dotted line shows the minimum standard deviation you can get, when applying the resulting linear model to the test data. Oftentimes, the right dotted lines is used instead. It is within one Standard Deviation of the minimum line and has a higher MSE. The pro of using it, is that it has a higher shrinkage, which means that more of the parameters are down to zero / shrunked.

## Swiss – LASSO

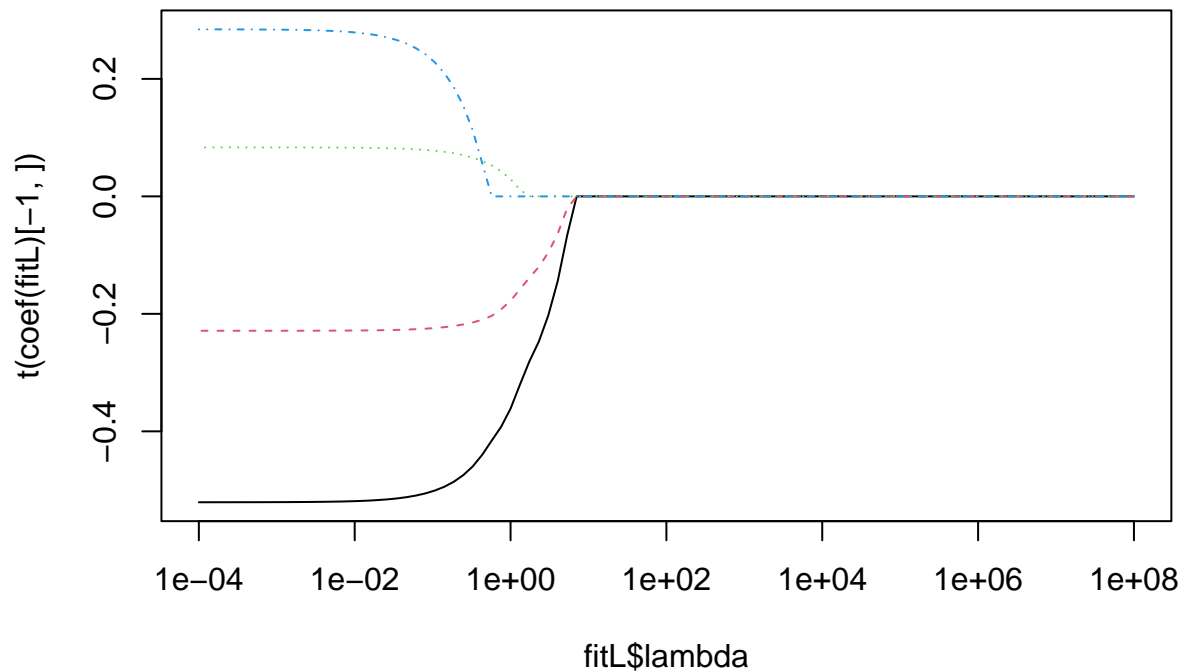
```
fitL <- glmnet(x=X, y=Y, alpha=1, lambda=lambda.grid) # alpha = 1 --> LASSO
dim(coef(fitL))
```

```
## [1] 5 100
```

### Plotten

```
#Plotten der LASSO
matplot(fitL$lambda, t(coef(fitL)[-1, ]), type="l", log="x", main="LASSO Switzerland")
```

## LASSO Switzerland



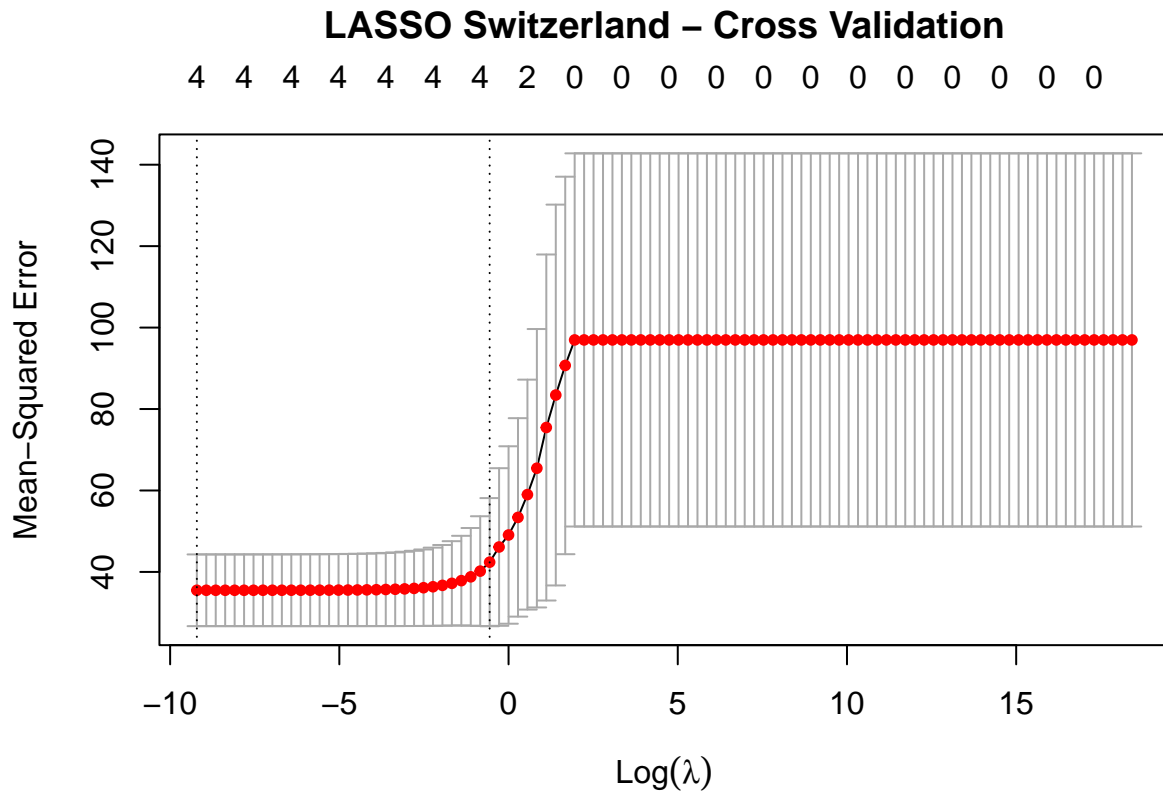
Compared to the RR, the parameters reach zero much quicker and actually reach it. This leads to those coefficients being discarded – Model selection of LASSO.

### Cross Validation Swiss LASSO

```
fitL_cv <- cv.glmnet(x=X, y=Y, alpha=1, lambda=lambda.grid) # cv = Cross Validation
```

```
fitL_cv <- cv.glmnet(x=X, y=Y, alpha = 1, lambda = lambda.grid)
fitL_cv_min <- fitL_cv$lambda.min
fitL_cv_1se <- fitL_cv$lambda.1se
```

```
#plotting
plot(fitL_cv, type="l")
title("LASSO Switzerland - Cross Validation", line=2.5)
```



## Get Coefficients for RR and LASSO Model fit

```
#for RR
RRlambda_min <- fitRR_cv$lambda.min # lamda of minimum mean cross-validated error
RRlambda_1se <- fitRR_cv$lambda.1se #largest value of lamda such that error is within 1 standard error of

fitRR_min_coef <- round(coef(fitRR)[,which(fitRR$lambda == RRlambda_min)], 3)
fitRR_1se_coef <- round(coef(fitRR)[,which(fitRR$lambda == RRlambda_1se)], 3)
```

Ridge Regression Fit, with lambda of minimum mean cross-validated error

$Education = 49,416 - 0,508 \cdot Fertility - 0,225 \cdot Agriculture + 0,080 \cdot Catholic + 0,269 \cdot InfantMortality$

Ridge Regression Fit, with largest lambda within 1 std. error of lambda.min

$Education = 41,139 - 0,354 \cdot Fertility - 0,174 \cdot Agriculture + 0,038 \cdot Catholic + 0,269 \cdot InfantMortality$

```
#coefficients for LASSO Model fit:
Llambda_min <- fitL_cv$lambda.min #lamda of minimum mean cross-validated error
Llambda_1se <- fitL_cv$lambda.1se #largest value of lamda such that error is within 1 standard error of

fitL_min_coef <- round(coef(fitL)[,which(fitL$lambda == Llambda_min)], 3)
fitL_1se_coef <- round(coef(fitL)[,which(fitL$lambda == Llambda_1se)], 3)
```

LASSO Fit, with lambda of minimum mean cross-validated error

$$Education = 49,993 - 0,521 \cdot Fertility - 0,229 \cdot Agriculture + 0,083 \cdot Catholic + 0,284 \cdot InfantMortality$$

LASSO Fit, with largest lambda within 1 std. error of lambda.min **Here, Infant Mortality is discarded as a regressor See the model selection**, with Infant Mortality and Catholic now missing.

$$Education = 30,023 - 0,203 \cdot Fertility - 0,096 \cdot Agriculture$$

## USA

Wir kehren zurück zu den Variablen “Population”, “Income”, “Illiteracy”, “Life.Exp”, “Murder”, “HS Grade” und “Frost” aus dem R Datensatz `state.x77`. Passen Sie für die oben genannten Variablen ein lineares Modell (lm) an, das “Murder” durch die übrigen Variablen erklärt, soweit dies zulässig ist. #

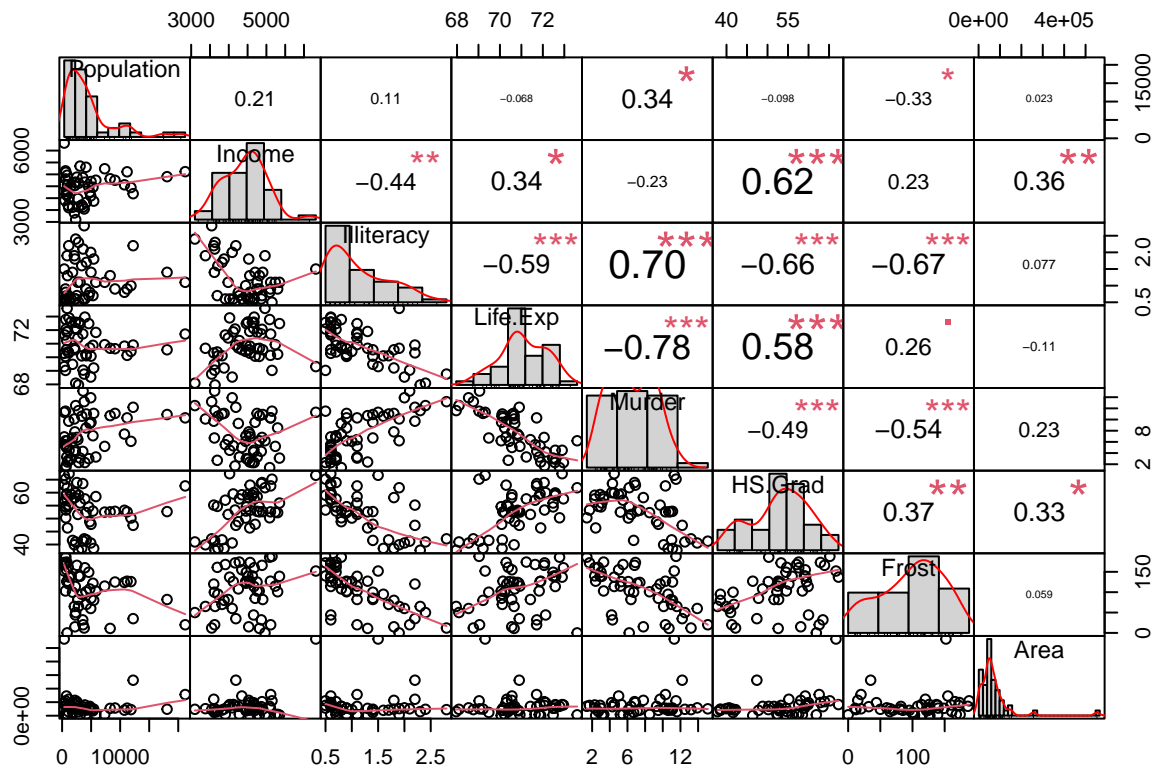
Im ersten Schritt überprüfen wir mittels Scatterplot die Korrelation der Variablen.

```
library(utils)
library("PerformanceAnalytics")

df <- data.frame(state.x77)
summary(df)
```

##	Population	Income	Illiteracy	Life.Exp
##	Min. : 365	Min. :3098	Min. :0.500	Min. :67.96
##	1st Qu.: 1080	1st Qu.:3993	1st Qu.:0.625	1st Qu.:70.12
##	Median : 2838	Median :4519	Median :0.950	Median :70.67
##	Mean : 4246	Mean :4436	Mean :1.170	Mean :70.88
##	3rd Qu.: 4968	3rd Qu.:4814	3rd Qu.:1.575	3rd Qu.:71.89
##	Max. :21198	Max. :6315	Max. :2.800	Max. :73.60
##	Murder	HS.Grad	Frost	Area
##	Min. : 1.400	Min. :37.80	Min. : 0.00	Min. : 1049
##	1st Qu.: 4.350	1st Qu.:48.05	1st Qu.: 66.25	1st Qu.: 36985
##	Median : 6.850	Median :53.25	Median :114.50	Median : 54277
##	Mean : 7.378	Mean :53.11	Mean :104.46	Mean : 70736
##	3rd Qu.:10.675	3rd Qu.:59.15	3rd Qu.:139.75	3rd Qu.: 81163
##	Max. :15.100	Max. :67.30	Max. :188.00	Max. :566432

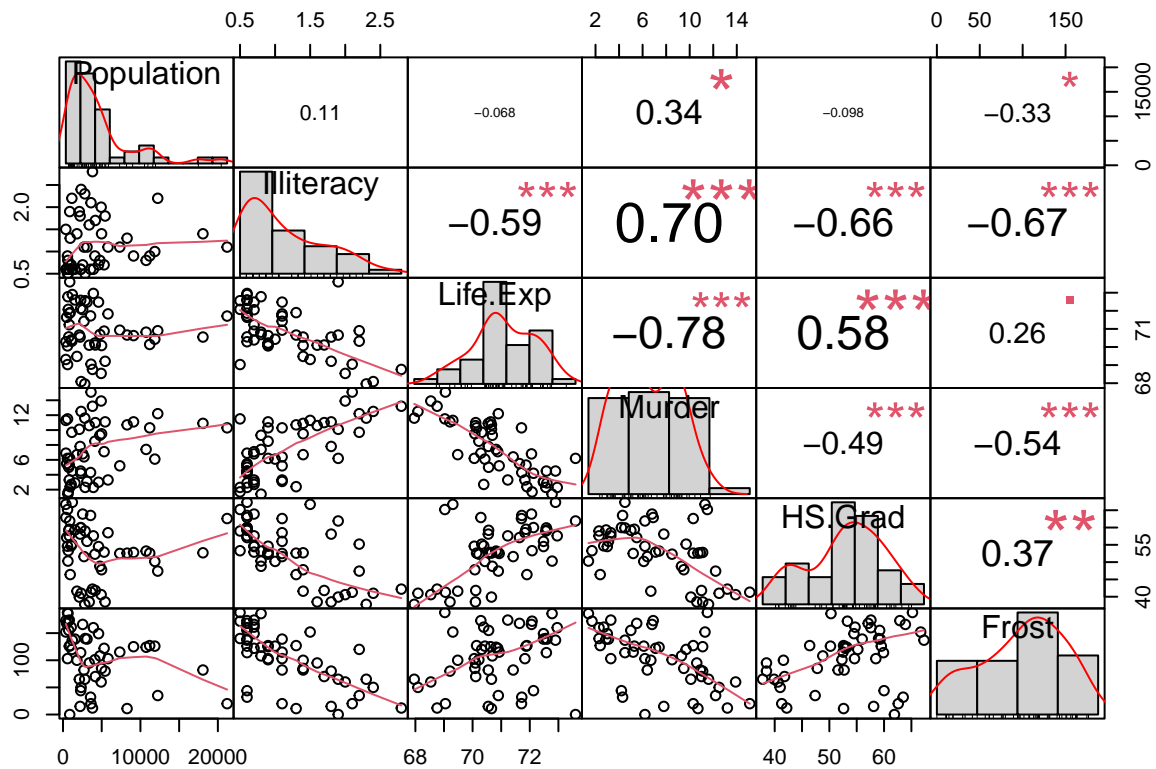
```
invisible(na.omit(df, "NA"))
#my_data <- df[, c(1,2,3,4,5,6)]
chart.Correlation(df, histogram=TRUE, pch=19)
```



```
#modell1 <- lm(formula = Murder ~ . - Area - Frost, data = df)
```

Dabei wird ersichtlich, dass alle Variablen außer Area und Income mit Murder korrelieren. Diese werden im ersten Schritt entfernt und erneut mittels Scatterplot die Korrelation der Variablen überprüft.

```
df_intititalRemoved <- df[, c(1,3,4,5,6,7)]
View(df_intititalRemoved)
chart.Correlation(df_intititalRemoved, histogram=TRUE, pch=19)
```

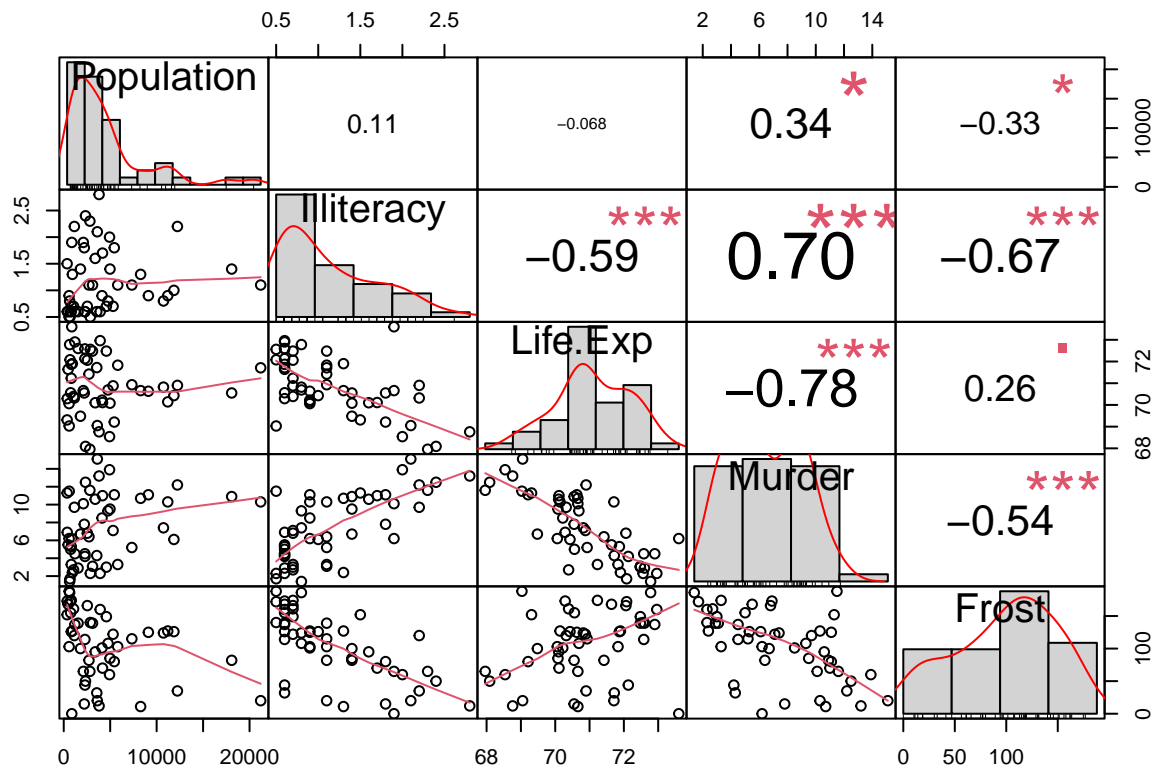


```
#modell1 <- lm(formula = Murder ~ . - Area - Frost, data = df)
```

Ausgehend davon bleiben nur die stark zu Murder korrelierten Variablen. Das Problem ist nun, dass auch einige der Kovariablen stark miteinander korreliert sind. Dabei konzentrieren wir uns vor allem auf die Werte über 0.5 bzw. um 0.8 herum. Wir haben uns dazu entschieden, HS.Grad komplett zu trimmen, da diese Variable viel höhere Korrelation zu den Kovariablen hat als zu Murder an sich hat.

```
df_secondRemoval <- df[, c(1,3,4,5,7)]
#View(df_secondRemoval)
chart.Correlation(df_secondRemoval, histogram=TRUE, pch=19)
```

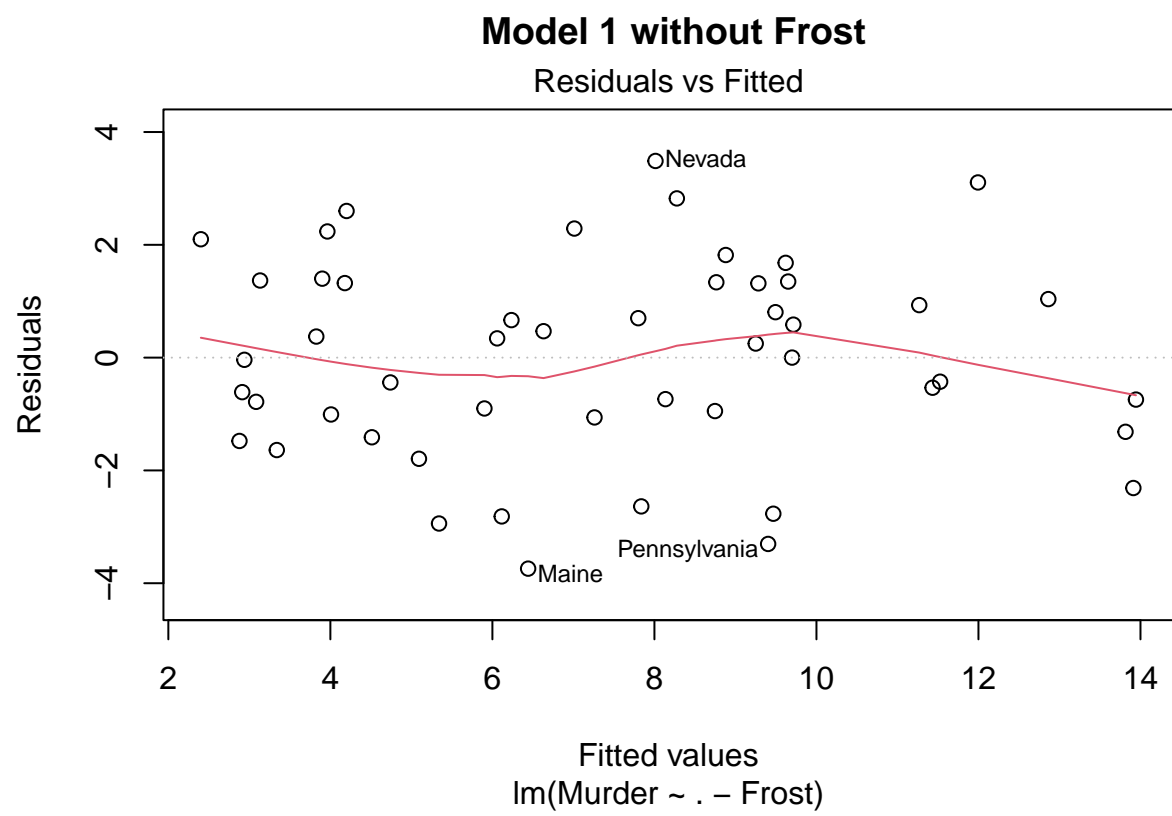


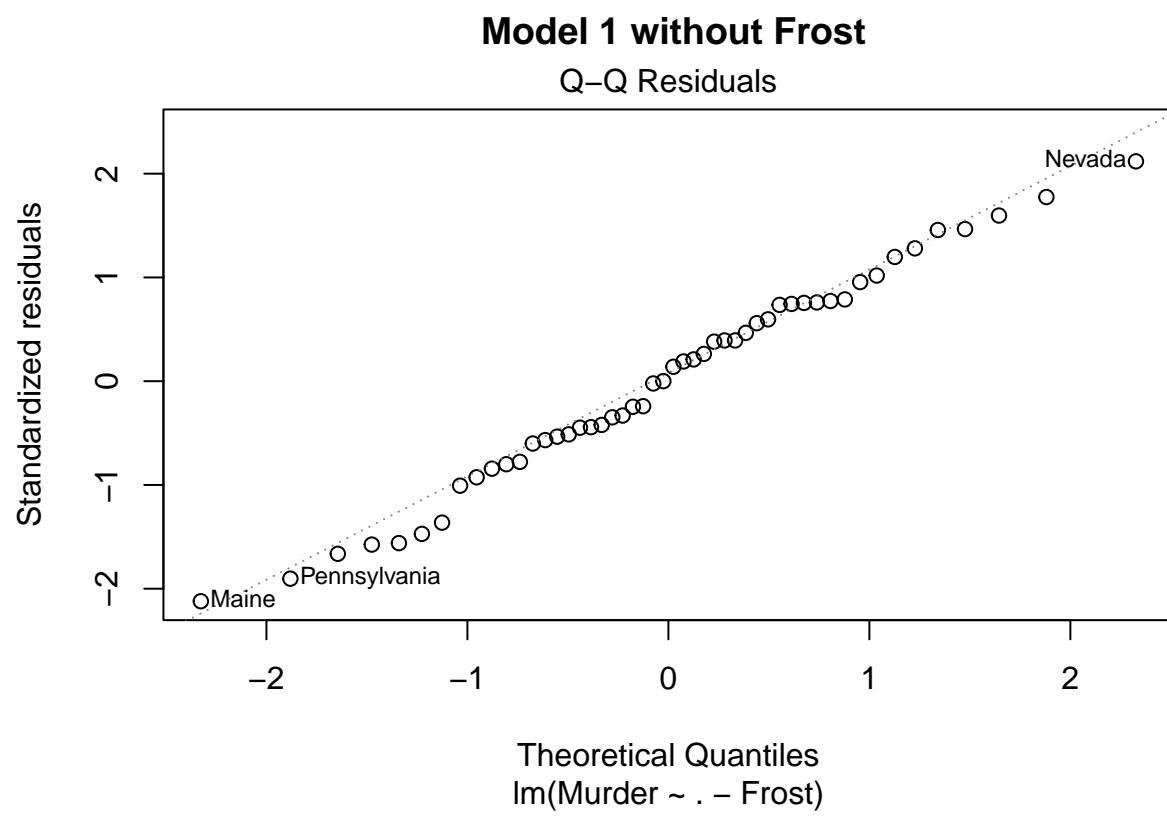


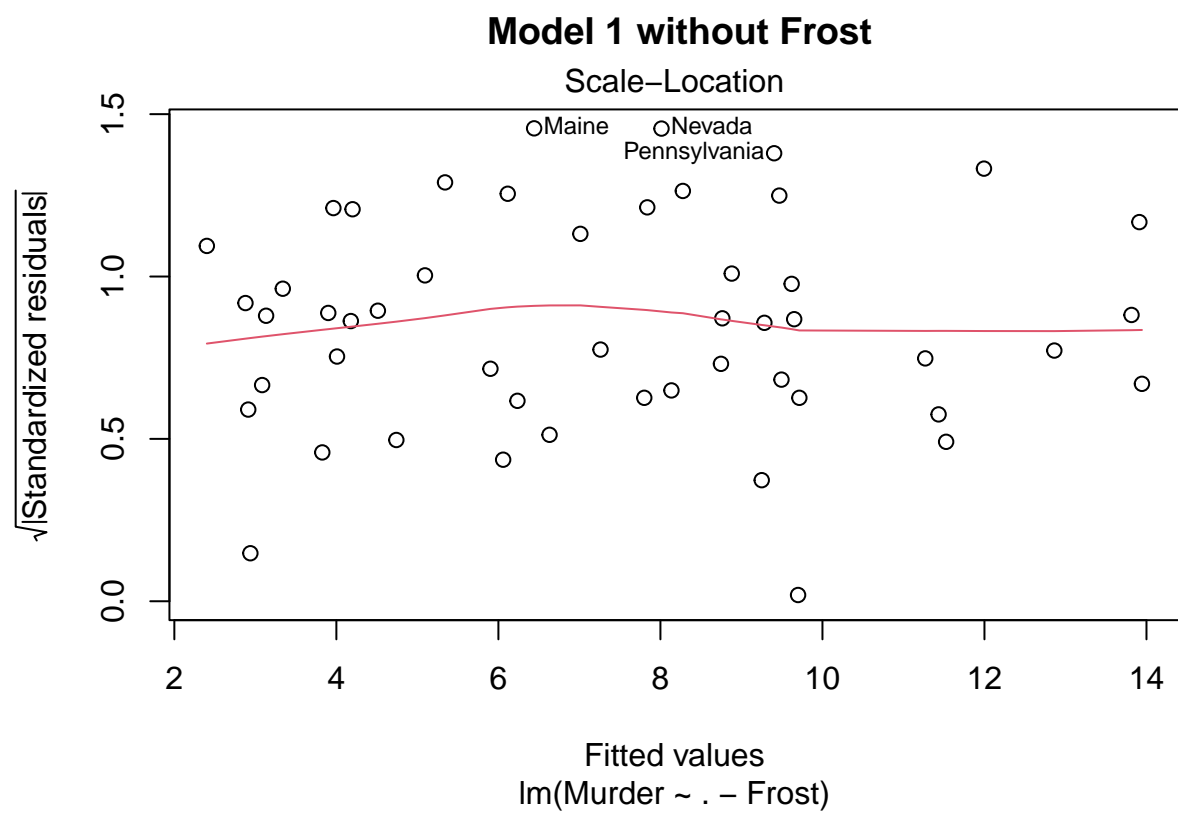
```
#modell1 <- lm(formula = Murder ~ . - Area - Frost, data = df)
```

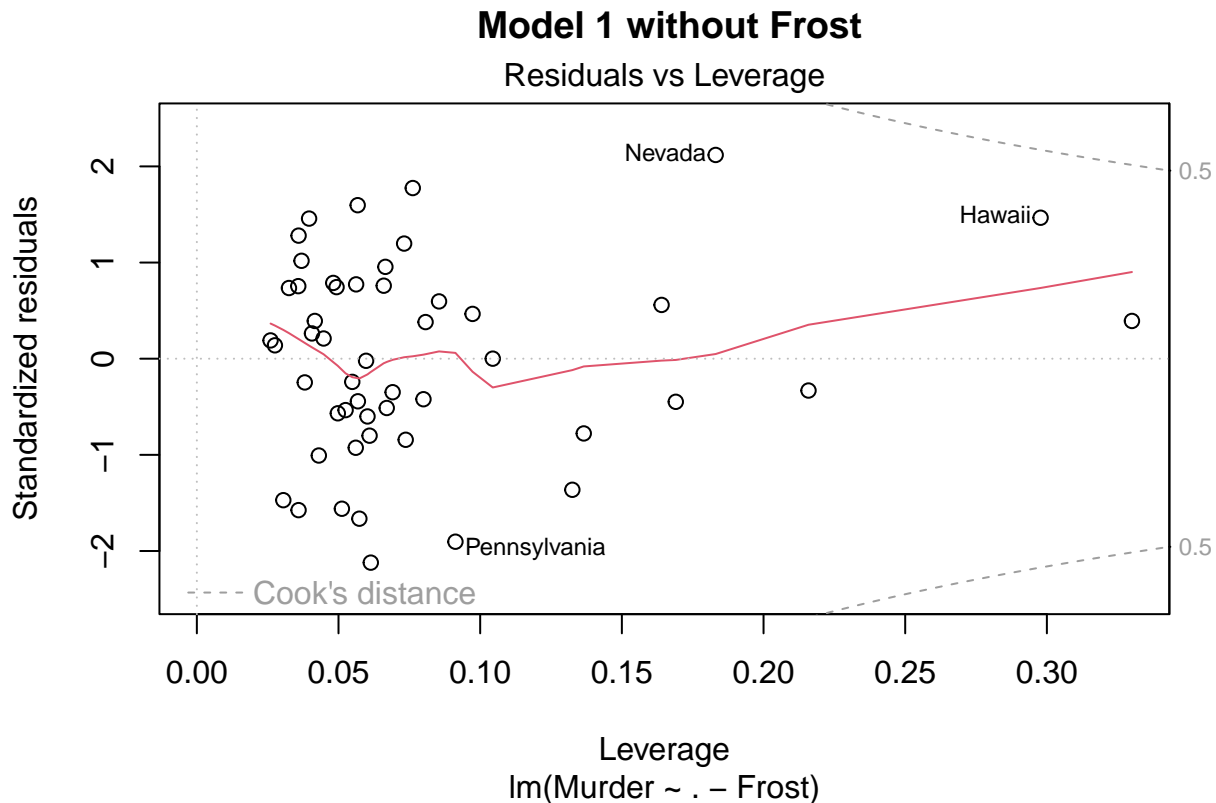
Somit bleiben nur noch mit Murder korrelierte Variablen und die Illiteracy und Frost, die Korrelation zu Kovariablen über 0.5 haben. Wir bilden also 2 Modelle, wo jeweils eine dieser Variablen verworfen wird, und analysieren anschließend die Residuenplots.

```
modell1WithoutFrost <- lm(formula = Murder ~ . - Frost , data = df_secondRemoval)
modell1WithoutIlliteracy <- lm(formula = Murder ~ . - Illiteracy , data = df_secondRemoval)
plot(modell1WithoutFrost, main = " Model 1 without Frost")
```









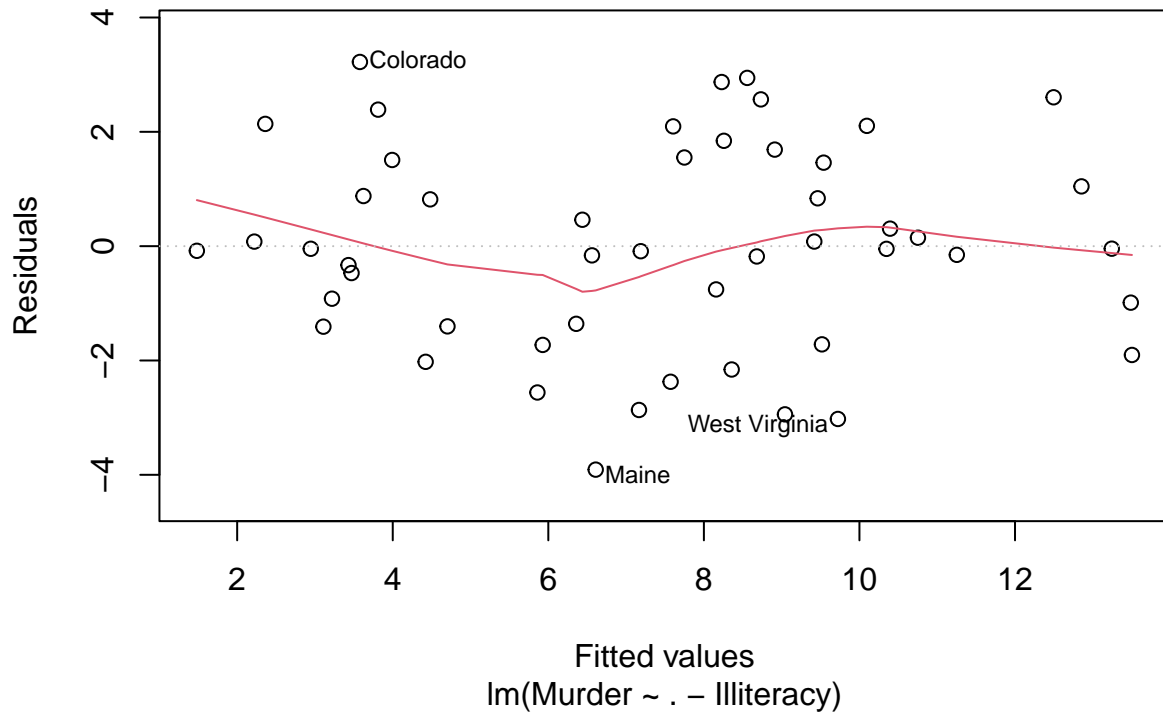
```
summary(modell1WithoutFrost)
```

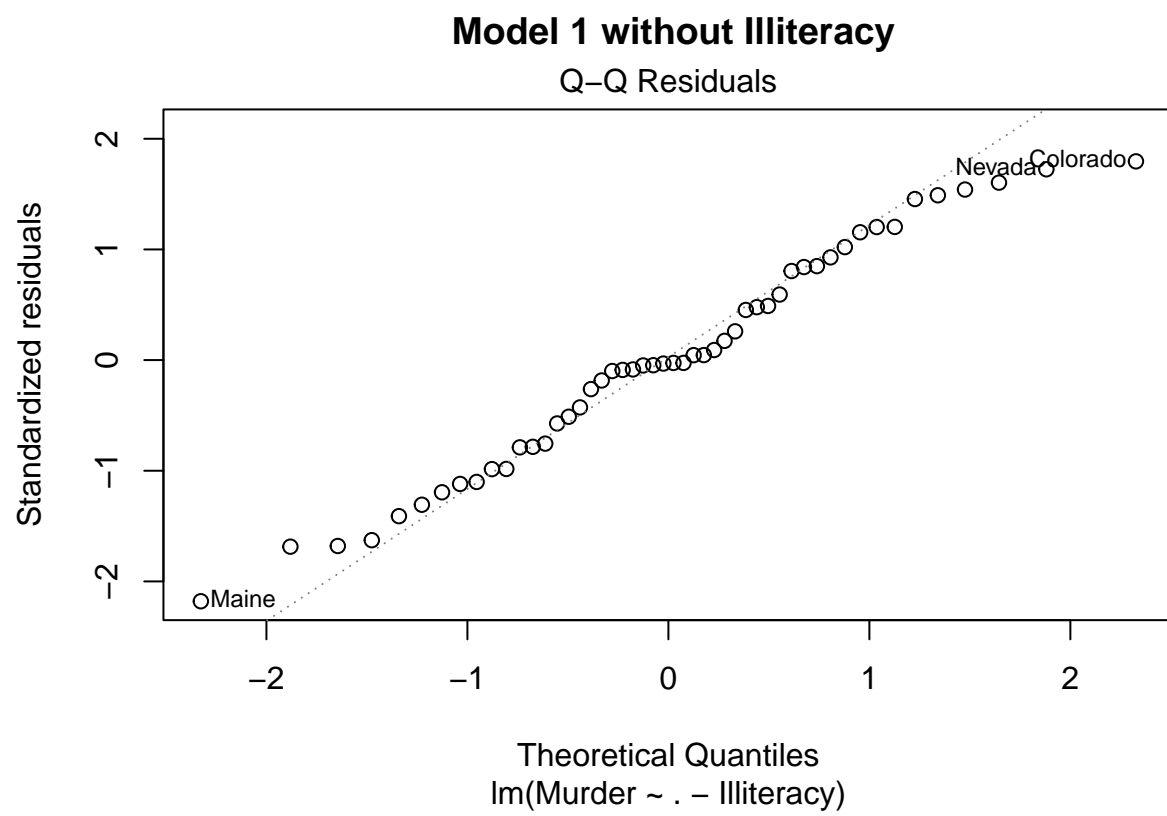
```
##
## Call:
## lm(formula = Murder ~ . - Frost, data = df_secondRemoval)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.7412 -1.0472  0.1251  1.3323  3.4869
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.131e+02  1.736e+01   6.514 4.89e-08 ***
## Population    2.220e-04  5.860e-05   3.788 0.000440 ***
## Illiteracy    2.088e+00  5.297e-01   3.941 0.000274 ***
## Life.Exp     -1.539e+00  2.397e-01  -6.423 6.71e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.821 on 46 degrees of freedom
## Multiple R-squared:  0.7716, Adjusted R-squared:  0.7567
## F-statistic: 51.81 on 3 and 46 DF,  p-value: 8.619e-15
```

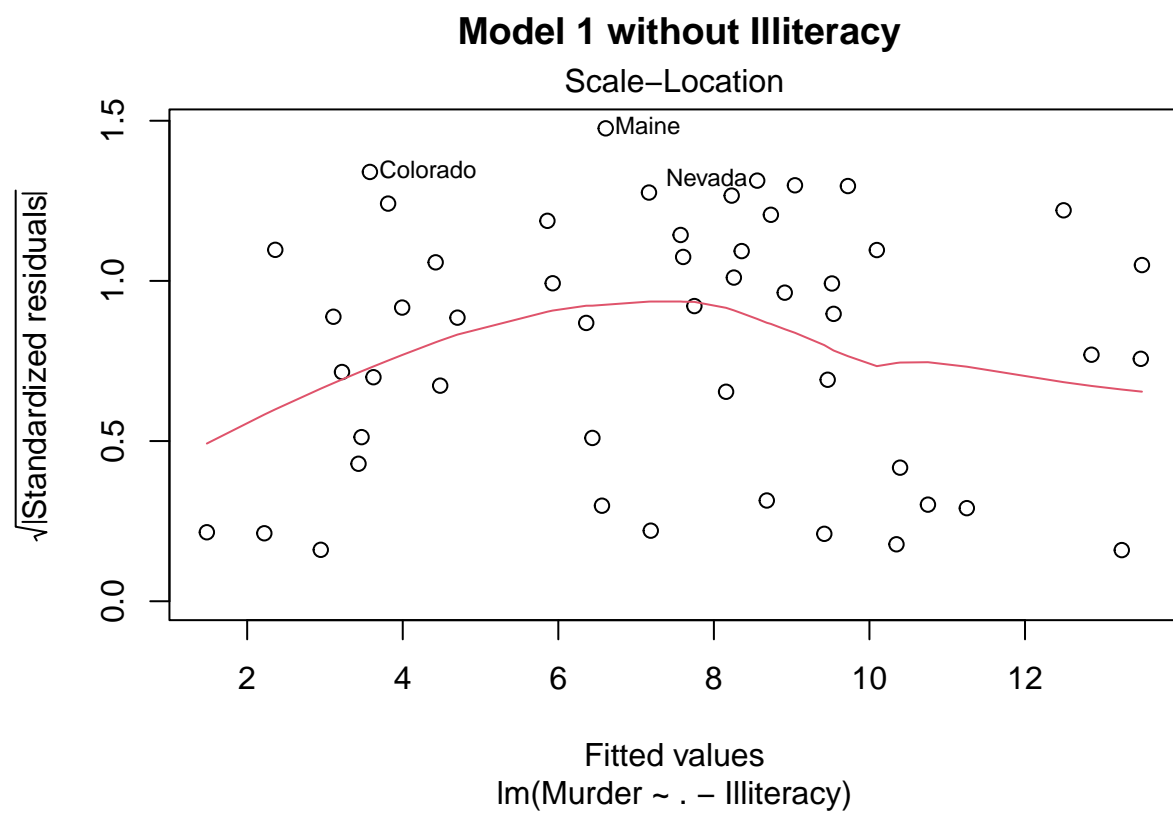
```
plot(modell1WithoutIlliteracy, main = "Model 1 without Illiteracy")
```

# Model 1 without Illiteracy

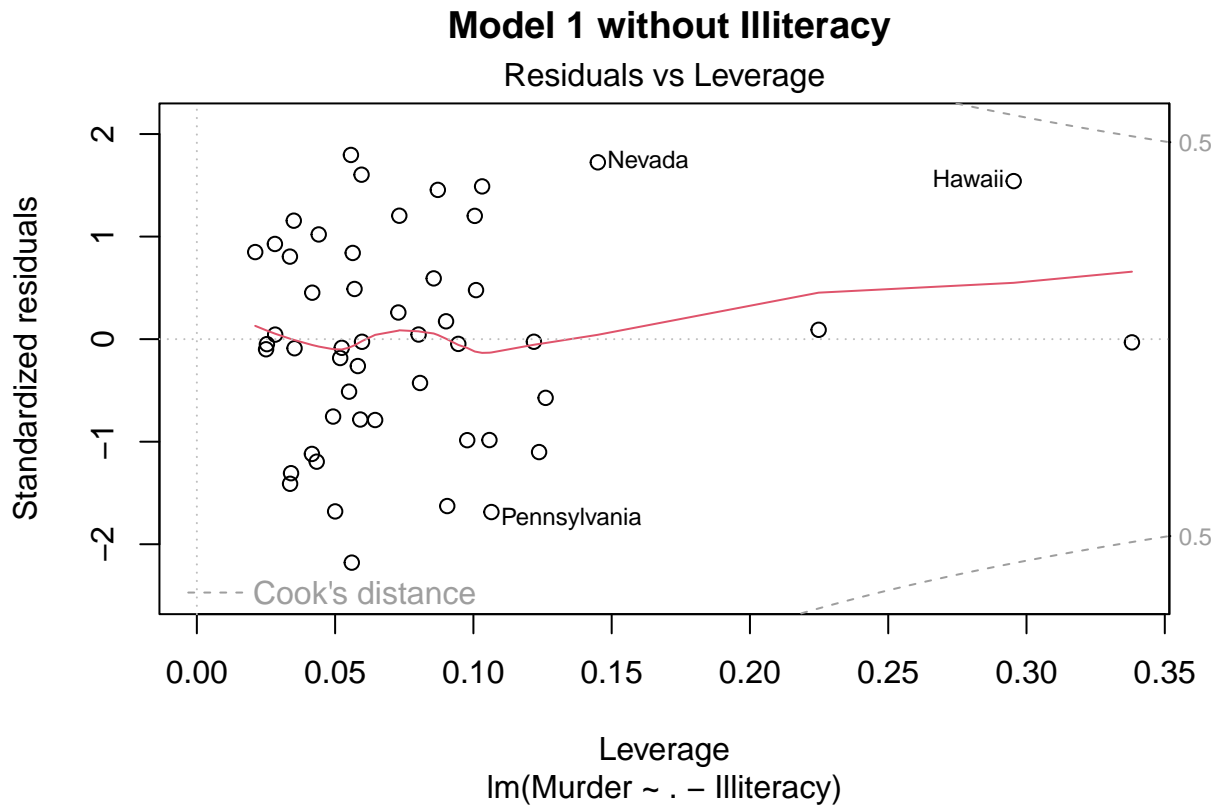
Residuals vs Fitted











```
summary(modell1WithoutIlliteracy)
```

```
##
## Call:
## lm(formula = Murder ~ . - Illiteracy, data = df_secondRemoval)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9092 -1.3924 -0.0468  1.4957  3.2221
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.435e+02  1.430e+01  10.035 3.64e-13 ***
## Population    1.652e-04  6.265e-05   2.637 0.011358 *
## Life.Exp     -1.900e+00  2.036e-01  -9.330 3.53e-12 ***
## Frost        -2.070e-02  5.562e-03  -3.721 0.000539 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.846 on 46 degrees of freedom
## Multiple R-squared:  0.7652, Adjusted R-squared:  0.7499
## F-statistic: 49.97 on 3 and 46 DF,  p-value: 1.627e-14
```

```
##Auswertung Summary und Residuenplots
```

```
#Model 1 : lm(formula = Murder ~ . - Frost, data = df_secondRemoval)
```

Residuals vs fitted: Überprüfung der Linearisitätsannahme und Homoskedastizität (konstante Varianz der Fehler). Das Modell hat keinen systematischen Fehler mehr in den Residuals und ist daher dsbbzgl. geeignet.

QQPlot: Überprüfung der Normalverteilungsannahme der Residuen. Die Residuen-Fehler approximieren in Annäherung eine Normalverteilung und ist daher dsbbzgl. geeignet.

Scal-Location-Plot: Überprüfung der Homoskedastizität. Im Scale-Location plot ist Homoskedastizität gegeben. Es liegt eine Punktwolke vor. Daher ist das Modell auch dsbbzgl. geeignet.

Residuals vs leverage Plot: Identifizierung von einflussreichen Datenpunkten (d.h. Punkte, die einen großen Einfluss auf die Anpassung des Modells haben). Die meisten Punkte sind nahe der y-Achse und haben keine große Hebelwirkung. Kein Punkt liegt ausserhalb der Cooks-Distance.

Summary: Sowohl Intercept als auch alle anderen Regressoren sind statistisch signifikant. Mit Multiple R-squared: 0.7716 können also 77% der Varianz von Murder durch die Prädiktoren erklärt werden. Mit einem p-value: 8.619e-15 ist das Modell insgesamt statistisch hoch signifikant.

```
#Model 2 : lm(formula = Murder ~ . - Illiteracy, data = df_secondRemoval)
```

Residuals vs fitted: Überprüfung der Linearisitätsannahme und Homoskedastizität (konstante Varianz der Fehler). Das Modell hat keinen systematischen Fehler mehr in den Residuals und ist daher dsbbzgl. geeignet.

QQPlot: Überprüfung der Normalverteilungsannahme der Residuen. Die Residuen-Fehler approximieren in Annäherung eine Normalverteilung und ist daher dsbbzgl. geeignet.

Scal-Location-Plot: Überprüfung der Homoskedastizität. Im Scale-Location plot ist Homoskedastizität gegeben. Es liegt eine Punktwolke vor. Daher ist das Modell auch dsbbzgl. geeignet.

Residuals vs leverage Plot: Identifizierung von einflussreichen Datenpunkten (d.h. Punkte, die einen großen Einfluss auf die Anpassung des Modells haben). Die meisten Punkte sind nahe der y-Achse und haben keine große Hebelwirkung. Kein Punkt liegt ausserhalb der Cooks-Distance.

Summary: Sowohl Intercept als auch alle anderen Regressoren sind statistisch hoch signifikant, mit Ausnahme von Population, was in diesem Modell nur signifikant ist. Mit Multiple R-squared: 0.7652 können also 76% der Varianz von Murder durch die Prädiktoren erklärt werden. Mit einem p-value: 1.627e-14 ist das Modell insgesamt hoch statistisch signifikant.

```
#Conclusion und Interpretation
```

Wir haben also 2 Modelle, die beinahe gleiche Güte bzgl der Modellanpassung besitzen. Im gesellschaftlichen Kontext ist es schwierig, den gegenseitigen Ausschluss von Frost und Illiteracy aufgrund deren Multikollinearität zu interpretieren. Einfacher dagegen ist nachzuvollziehen, dass wohl eine höhere Population direkt mit einer erhöhten Mordrate zusammenhängt, und dass Illiteracy direkt mit der Mordrate korreliert, weswegen sich aus diesen Regressoren ein multiples lineares GM modellieren lässt.

Im nächsten Schritt werden wir nun mittels RIDGE und LASSO versuchen, das optimale zu ermitteln, und dieses mit unseren 2 bisherigen Modellen zu vergleichen.

```
##Ridge und LASSOO
```

```
library(glmnet)
library(ggplot2)
library(dplyr)
library(GGally)

set.seed(123)

usa <- state.x77
lambda.grid <- 10^seq(10, -2, length=100) # 10^10 bis 10^-2 in 100 stufen
```

```
#=====
#=====
```

## Bsp. 03 – USA

```
#=====
#=====
##Ridge Regression USA
```

### Data Preparation

```
# Data prep USA
usa <- as.data.frame(state.x77)
#unabhängige variablen
X <- usa %>% dplyr::select(-Murder) %>% as.matrix()
#abhängige variable
Y <- usa %>% dplyr::select(Murder) %>% as.matrix()
Y
```

##	Murder
## Alabama	15.1
## Alaska	11.3
## Arizona	7.8
## Arkansas	10.1
## California	10.3
## Colorado	6.8
## Connecticut	3.1
## Delaware	6.2
## Florida	10.7
## Georgia	13.9
## Hawaii	6.2
## Idaho	5.3
## Illinois	10.3
## Indiana	7.1
## Iowa	2.3
## Kansas	4.5
## Kentucky	10.6
## Louisiana	13.2
## Maine	2.7
## Maryland	8.5
## Massachusetts	3.3
## Michigan	11.1
## Minnesota	2.3
## Mississippi	12.5
## Missouri	9.3
## Montana	5.0
## Nebraska	2.9
## Nevada	11.5
## New Hampshire	3.3
## New Jersey	5.2
## New Mexico	9.7
## New York	10.9
## North Carolina	11.1
## North Dakota	1.4

## Ohio	7.4
## Oklahoma	6.4
## Oregon	4.2
## Pennsylvania	6.1
## Rhode Island	2.4
## South Carolina	11.6
## South Dakota	1.7
## Tennessee	11.0
## Texas	12.2
## Utah	4.5
## Vermont	5.5
## Virginia	9.5
## Washington	4.3
## West Virginia	6.7
## Wisconsin	3.0
## Wyoming	6.9

X

##	Population	Income	Illiteracy	Life Exp	HS Grad	Frost	Area
## Alabama	3615	3624	2.1	69.05	41.3	20	50708
## Alaska	365	6315	1.5	69.31	66.7	152	566432
## Arizona	2212	4530	1.8	70.55	58.1	15	113417
## Arkansas	2110	3378	1.9	70.66	39.9	65	51945
## California	21198	5114	1.1	71.71	62.6	20	156361
## Colorado	2541	4884	0.7	72.06	63.9	166	103766
## Connecticut	3100	5348	1.1	72.48	56.0	139	4862
## Delaware	579	4809	0.9	70.06	54.6	103	1982
## Florida	8277	4815	1.3	70.66	52.6	11	54090
## Georgia	4931	4091	2.0	68.54	40.6	60	58073
## Hawaii	868	4963	1.9	73.60	61.9	0	6425
## Idaho	813	4119	0.6	71.87	59.5	126	82677
## Illinois	11197	5107	0.9	70.14	52.6	127	55748
## Indiana	5313	4458	0.7	70.88	52.9	122	36097
## Iowa	2861	4628	0.5	72.56	59.0	140	55941
## Kansas	2280	4669	0.6	72.58	59.9	114	81787
## Kentucky	3387	3712	1.6	70.10	38.5	95	39650
## Louisiana	3806	3545	2.8	68.76	42.2	12	44930
## Maine	1058	3694	0.7	70.39	54.7	161	30920
## Maryland	4122	5299	0.9	70.22	52.3	101	9891
## Massachusetts	5814	4755	1.1	71.83	58.5	103	7826
## Michigan	9111	4751	0.9	70.63	52.8	125	56817
## Minnesota	3921	4675	0.6	72.96	57.6	160	79289
## Mississippi	2341	3098	2.4	68.09	41.0	50	47296
## Missouri	4767	4254	0.8	70.69	48.8	108	68995
## Montana	746	4347	0.6	70.56	59.2	155	145587
## Nebraska	1544	4508	0.6	72.60	59.3	139	76483
## Nevada	590	5149	0.5	69.03	65.2	188	109889
## New Hampshire	812	4281	0.7	71.23	57.6	174	9027
## New Jersey	7333	5237	1.1	70.93	52.5	115	7521
## New Mexico	1144	3601	2.2	70.32	55.2	120	121412
## New York	18076	4903	1.4	70.55	52.7	82	47831
## North Carolina	5441	3875	1.8	69.21	38.5	80	48798
## North Dakota	637	5087	0.8	72.78	50.3	186	69273

## Ohio	10735	4561	0.8	70.82	53.2	124	40975
## Oklahoma	2715	3983	1.1	71.42	51.6	82	68782
## Oregon	2284	4660	0.6	72.13	60.0	44	96184
## Pennsylvania	11860	4449	1.0	70.43	50.2	126	44966
## Rhode Island	931	4558	1.3	71.90	46.4	127	1049
## South Carolina	2816	3635	2.3	67.96	37.8	65	30225
## South Dakota	681	4167	0.5	72.08	53.3	172	75955
## Tennessee	4173	3821	1.7	70.11	41.8	70	41328
## Texas	12237	4188	2.2	70.90	47.4	35	262134
## Utah	1203	4022	0.6	72.90	67.3	137	82096
## Vermont	472	3907	0.6	71.64	57.1	168	9267
## Virginia	4981	4701	1.4	70.08	47.8	85	39780
## Washington	3559	4864	0.6	71.72	63.5	32	66570
## West Virginia	1799	3617	1.4	69.48	41.6	100	24070
## Wisconsin	4589	4468	0.7	72.48	54.5	149	54464
## Wyoming	376	4566	0.6	70.29	62.9	173	97203

```
fitRR <- glmnet::glmnet(x=X, y=Y, alpha = 0, lambda = lambda.grid) #alpha = zero->ridge regression
dim(coef(fitRR)) #zur kontrolle, zeigt welche dimensions (inkl. intercept) wir haben
```

```
## [1] 8 100
```

Wir haben nun 8 Coeffizienten (inkl. Intercept)

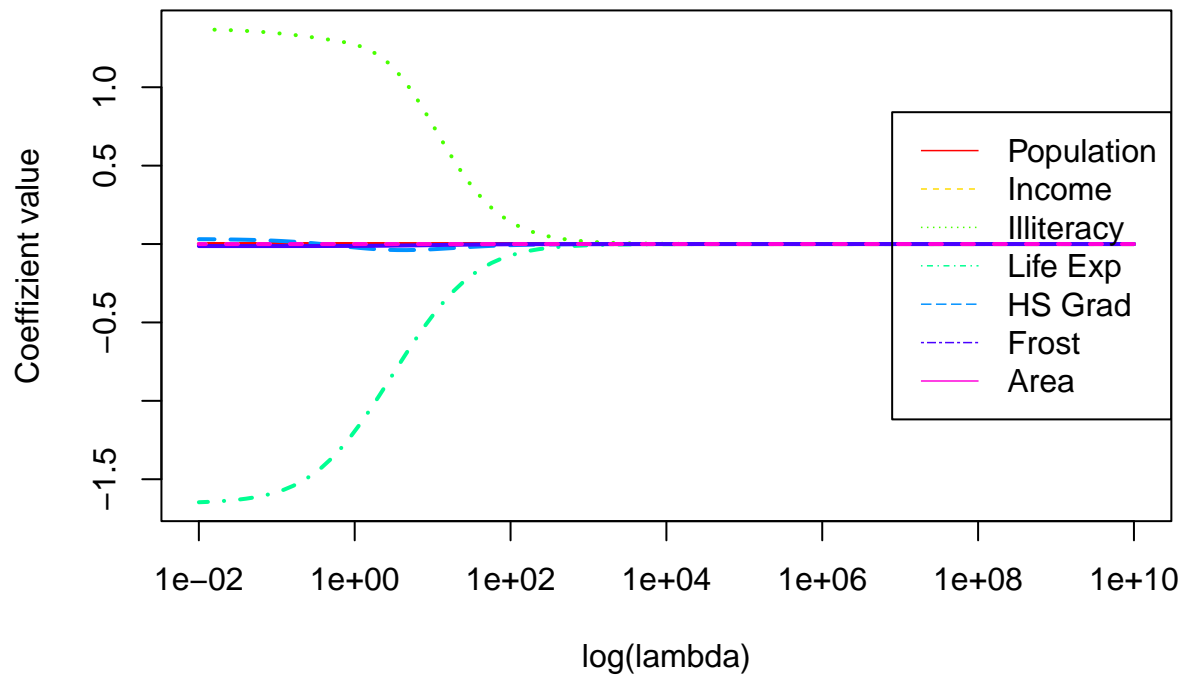
**Daten Plotten:**

```
n <- ncol(t(coef(fitRR)[-1, ]))
col_set <- rainbow(n)

matplot(fitRR$lambda, t(coef(fitRR)[-1, ]), type="l", log="x", main="Ridge Regression USA", ylab = "Coe

legend('right', legend = colnames(t(coef(fitRR)[-1, ])), col = col_set, lty = 1:n, lwd = 0.75 )
```

## Ridge Regression USA



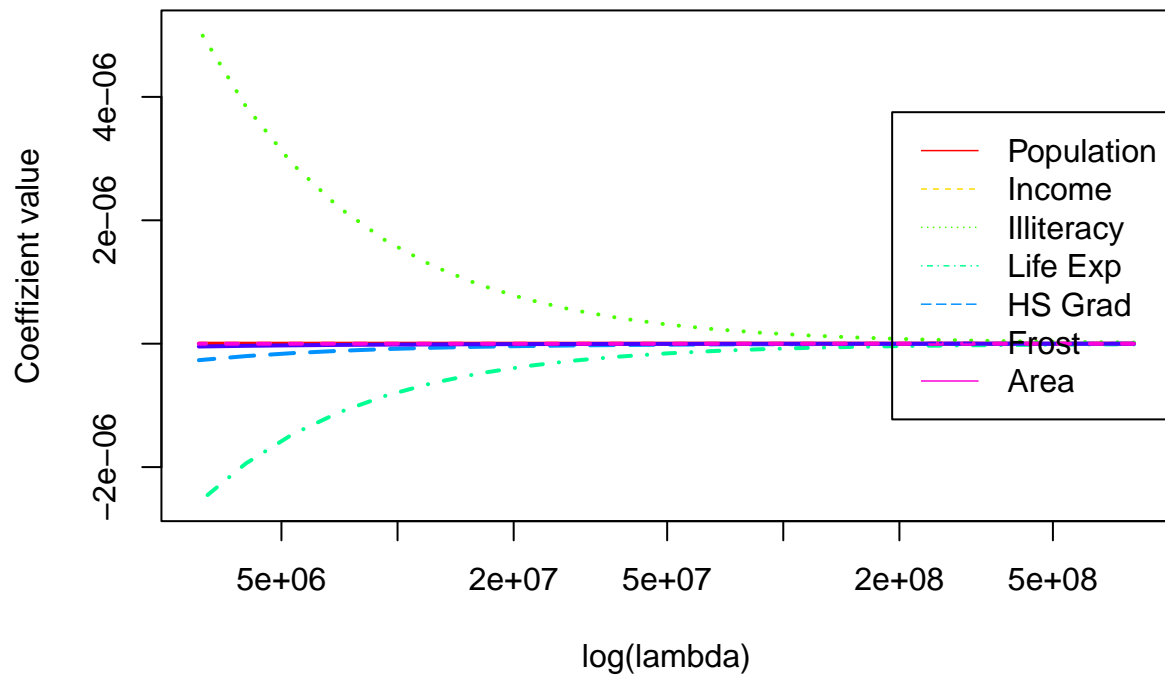
Zoomed in its possible to see, how the slopes are approaching zero very slowly.

```
n <- ncol(t(coef(fitRR)[-1, ]))
col_set <- rainbow(n)

matplot(fitRR$lambda[10:30], t(coef(fitRR)[-1, 10:30]), type="l", log="x", main="Ridge Regression USA",

legend('right', legend = colnames(t(coef(fitRR)[-1, ])), col = col_set, lty = 1:n, lwd = 0.75 )
```

## Ridge Regression USA



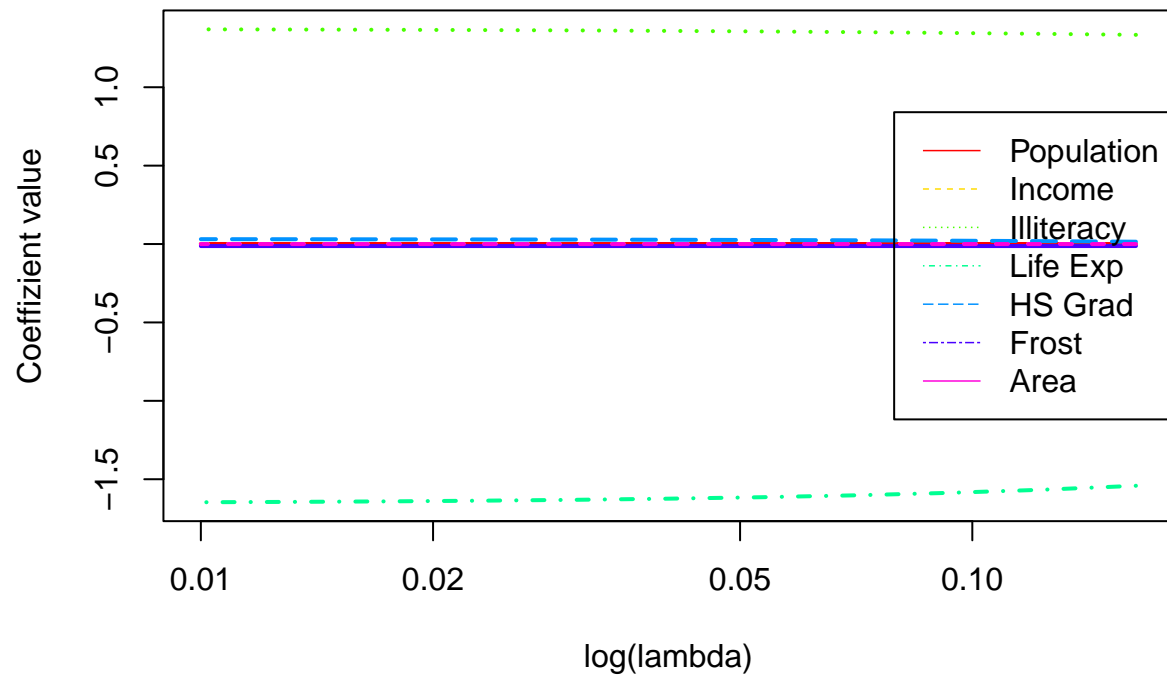
Zooming in even further, the lines are approaching parallel to zero, as they will never reach it:

```
n <- ncol(t(coef(fitRR)[-1, ]))
col_set <- rainbow(n)

matplot(fitRR$lambda[90:100], t(coef(fitRR)[-1, 90:100]), type="l", log="x", main="Ridge Regression USA")

legend('right', legend = colnames(t(coef(fitRR)[-1, ])), col = col_set, lty = 1:n, lwd = 0.75 )
```

## Ridge Regression USA

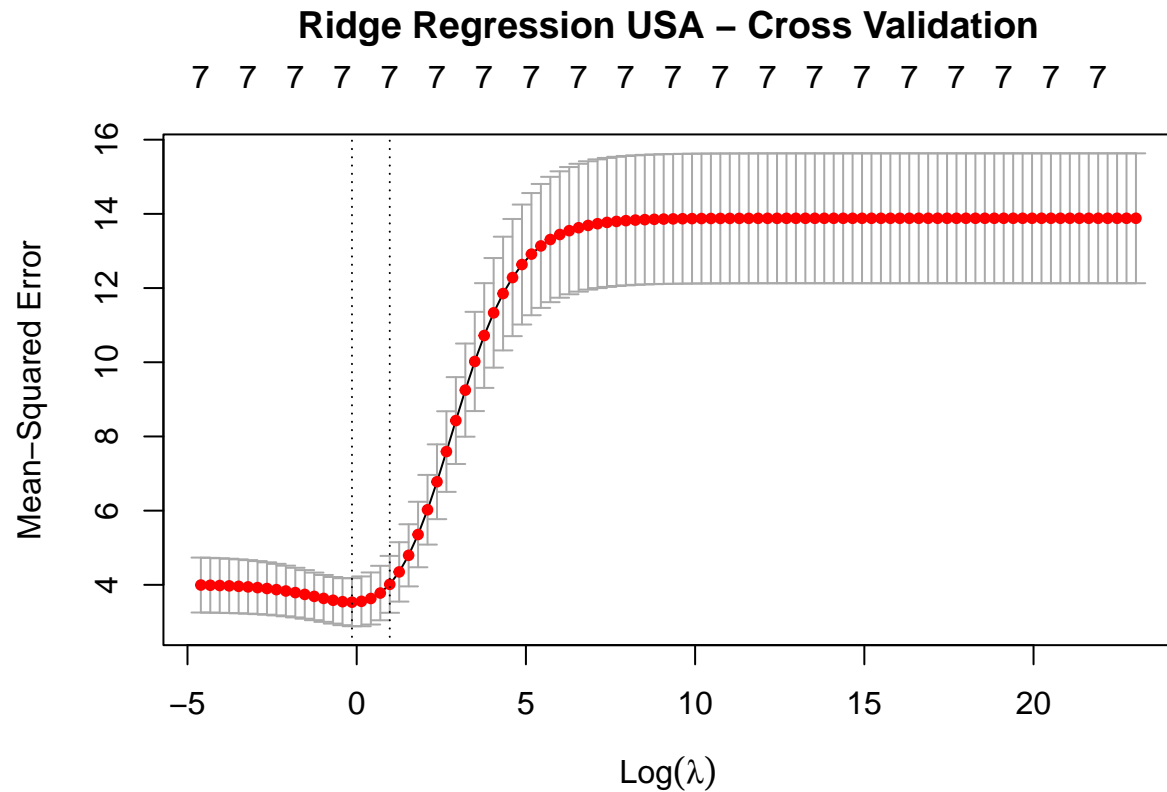


### Cross Validation

```
fitRR_cv <- cv.glmnet(x=X, y=Y, alpha = 0, lambda = lambda.grid)
fitRR_cv_min <- fitRR_cv$lambda.min
fitRR_cv_1se <- fitRR_cv$lambda.1se

#plotting
plot(fitRR_cv, type="l")
title("Ridge Regression USA - Cross Validation", line=2.5)
```





## USA – LASSO

```
fitL <- glmnet(x=X, y=Y, alpha=1, lambda=lambda.grid) # alpha = 1 --> LASSO
dim(coef(fitL))
```

```
## [1] 8 100
```

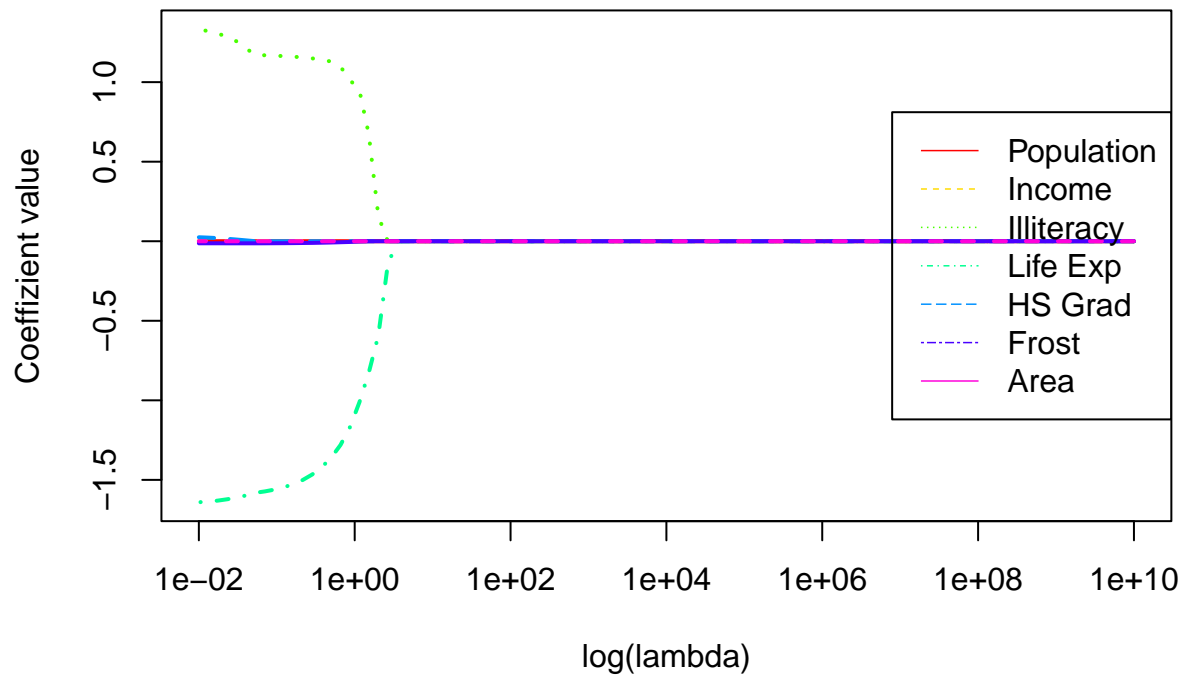
### Plotten

```
#Plotten der LASSO
n <- ncol(t(coef(fitL)[-1, ]))
col_set <- rainbow(n)

matplot(fitL$lambda, t(coef(fitL)[-1, ]), type="l", log="x", main="LASSO USA", ylab = "Coeffizient value")

legend('right', legend = colnames(t(coef(fitL)[-1, ])), col = col_set, lty = 1:n, lwd = 0.75 )
```

## LASSO USA



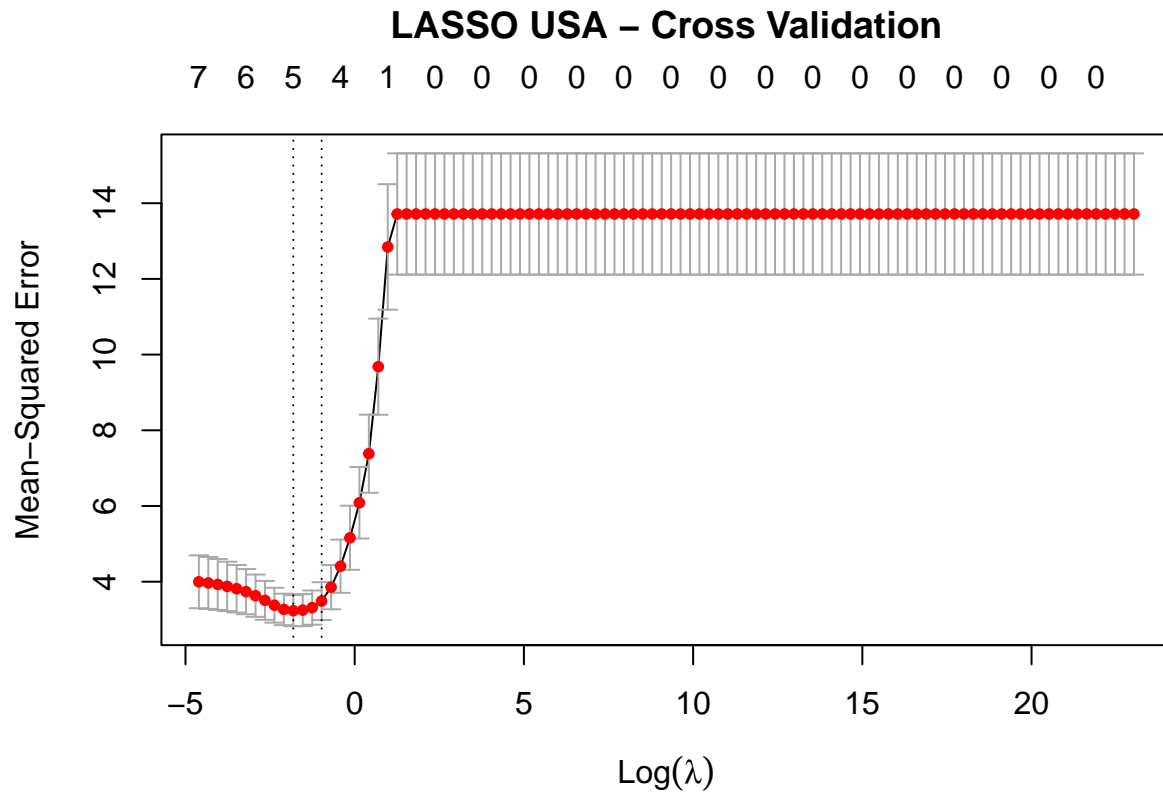
Here it is nice to see, how the slopes are moving to zero, which mean the Regressor will ‘disappear’. In comparison to RR, with LASSO, the values reach zero.

## Cross Validation USA LASSO

```
fitL_cv <- cv.glmnet(x=X, y=Y, alpha=1, lambda=lambda.grid) # cv = Cross Validation
```

```
fitL_cv <- cv.glmnet(x=X, y=Y, alpha =1, lambda = lambda.grid)
fitL_cv_min <- fitL_cv$lambda.min
fitL_cv_1se <- fitL_cv$lambda.1se
```

```
#plotting
plot(fitL_cv, type="l")
title("LASSO USA - Cross Validation", line=2.5)
```



Get Coefficients for RR and LASSO Model fit (USA)

Original linear model:  $Murder = 143,5 + 0,0001652 \cdot Population - 1,900 \cdot LifeExp - 0,0270 \cdot Frost$

```
#for RR
RRlambda_min <- fitRR_cv$lambda.min #lamda of minimum mean cross-validated error
RRlambda_1se <- fitRR_cv$lambda.1se #largest value of lamdalambda such that error is within 1 standard e

fitRR_min_coef <- round(coef(fitRR)[,which(fitRR$lambda == RRlambda_min)], 3)
fitRR_1se_coef <- round(coef(fitRR)[,which(fitRR$lambda == RRlambda_1se)], 3)
```

**Ridge Regression Fit** with lambda of minimum mean cross-validated error

$Murder = 89,439 + 1,263 \cdot Illiteracy - 1,153 \cdot LifeExp - 0,025 \cdot HSGrad - 0,012 \cdot Frost$

with largest lambda within 1 std. error of lambda.min

$Murder = 64,688 + 1,097 \cdot Illiteracy - 0,787 \cdot LifeExp - 0,038 \cdot HSGrad - 0,010 \cdot Frost$

```
#coefficients for LASSO Model fit:
Llambda_min <- fitL_cv$lambda.min #lamda of minimum mean cross-validated error
Llambda_1se <- fitL_cv$lambda.1se #largest value of lamda such that error is within 1 standard error of

fitL_min_coef <- round(coef(fitL)[,which(fitL$lambda == Llambda_min)], 3)
fitL_1se_coef <- round(coef(fitL)[,which(fitL$lambda == Llambda_1se)], 3)
```

LASSO Fit, with lambda of minimum mean cross-validated error

$$Murder = 89,357 + 1,013 \cdot Illiteracy - 1,168 \cdot LifeExp - 0,005 \cdot Frost$$

LASSO Fit, with largest lambda within 1 std. error of lambda.min **Here, Infant Mortality is discarded as a regressor**

$$Education = 49,210 + 0,159 \cdot Illiteracy - 0,593 \cdot LifeExp$$

**NOTE:** The coefficients of the cross validation change each time the program runs!

## Lake Huron

Wir kehren zurück zum Datensatz “LakeHuron”. Passen Sie ein Modell an, das den Zeittrend modelliert. Überprüfen Sie alle erforderlichen statistischen Voraussetzungen für die Gültigkeit dieses Modells mithilfe der quality plots der Residuen.

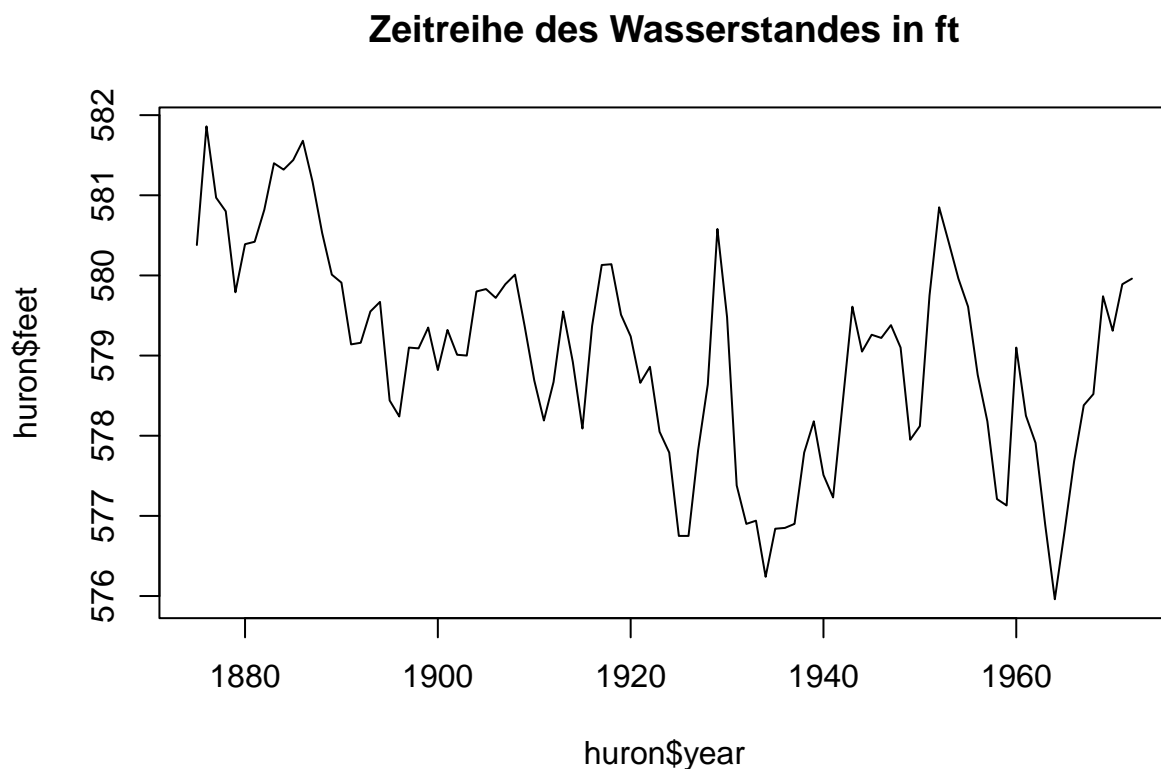
```
huron <- data.frame(feet=as.matrix(LakeHuron), date=time(LakeHuron))
huron["year"] <- 1875:1972
huron <- subset(huron, select = -c(date))
str(huron)
```

```
## 'data.frame': 98 obs. of 2 variables:
## $ feet: num 580 582 581 581 580 ...
## $ year: int 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 ...
```

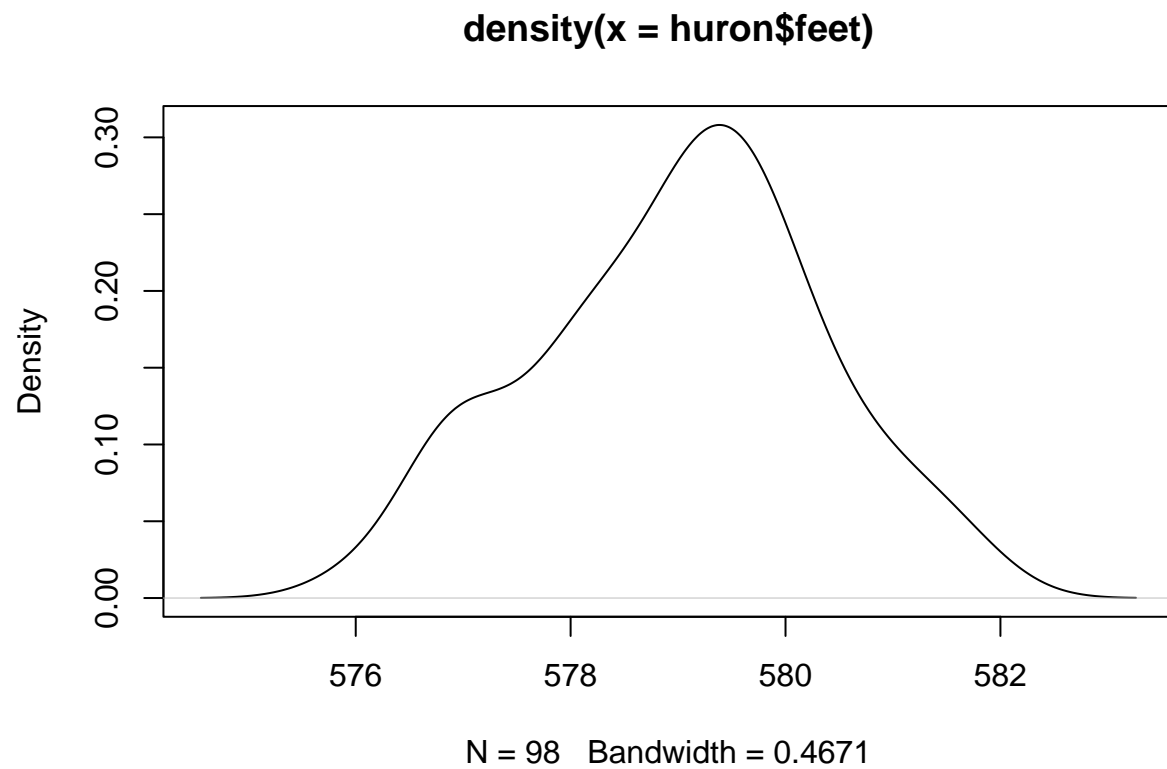
```
summary(huron)
```

```
##      feet      year
## Min.   :576.0   Min.   :1875
## 1st Qu.:578.1   1st Qu.:1899
## Median :579.1   Median :1924
## Mean   :579.0   Mean   :1924
## 3rd Qu.:579.9   3rd Qu.:1948
## Max.   :581.9   Max.   :1972
```

```
plot(x = huron$year, y=huron$feet, type="l", main="Zeitreihe des Wasserstandes in ft")
```

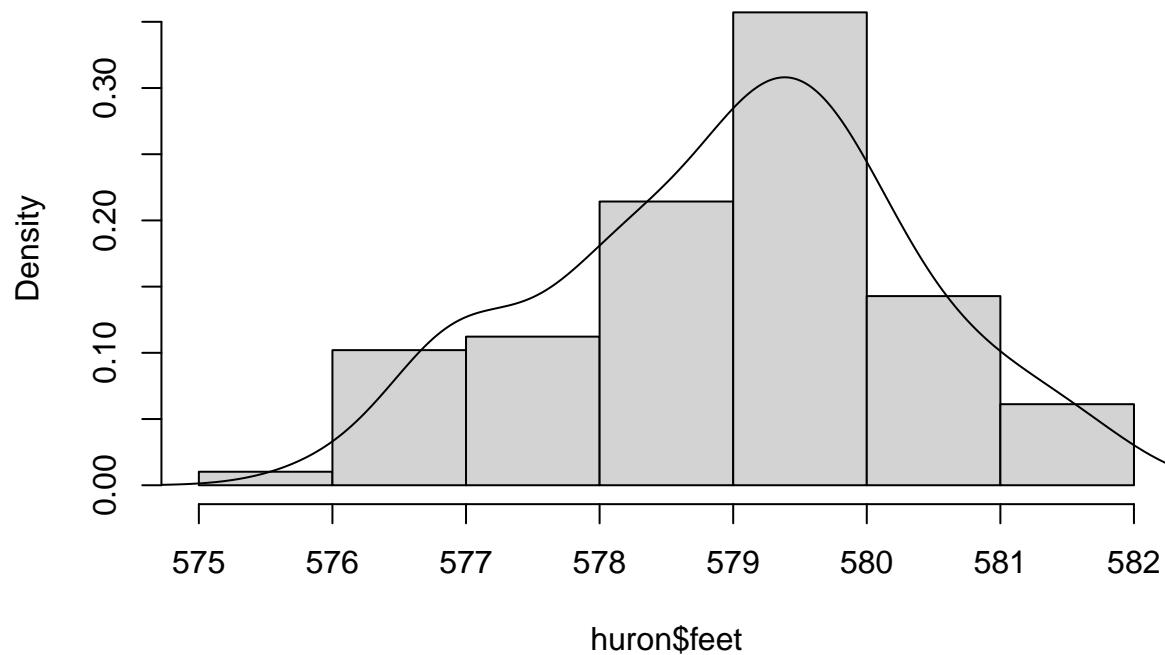


```
plot(density(huron$feet))
```



```
hist(huron$feet, freq=F, main="Dichtefunktion für Wasserstand in ft von Lake Huron (1875-1972)")  
lines(density(huron$feet))
```

## Dichtefunktion für Wasserstand in ft von Lake Huron (1875–1972)

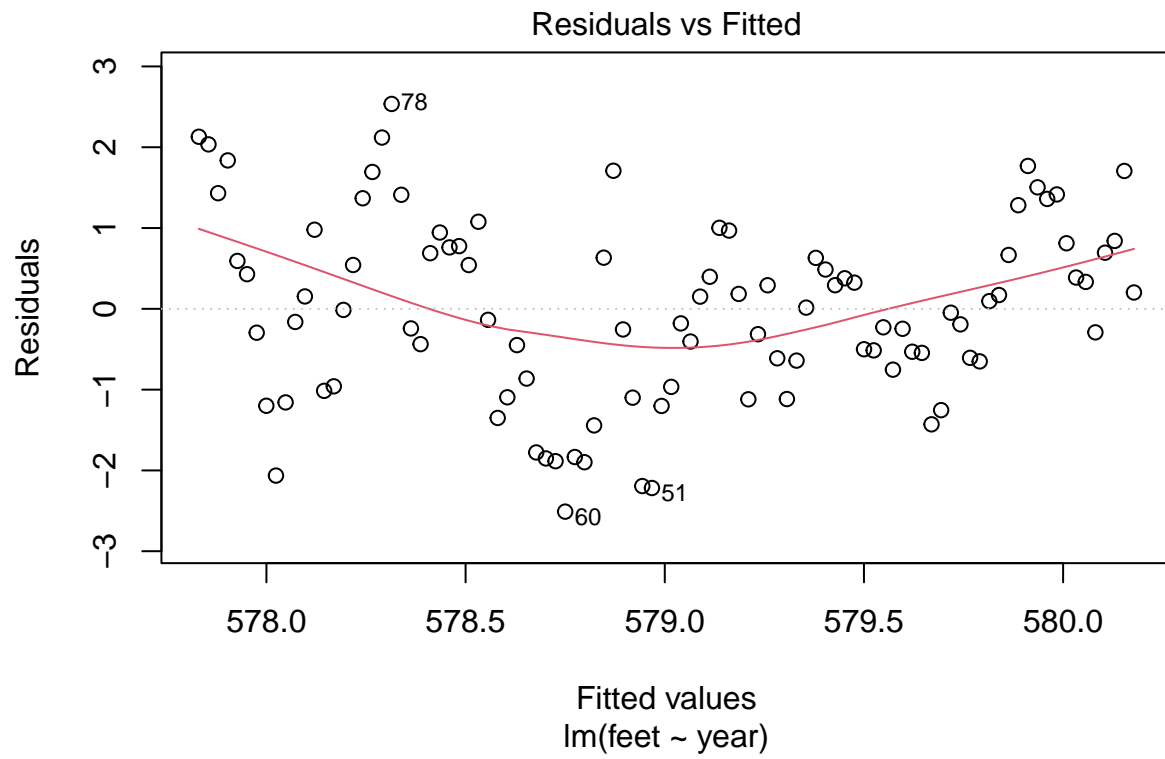


```
shapiro.test(huron$feet)
```

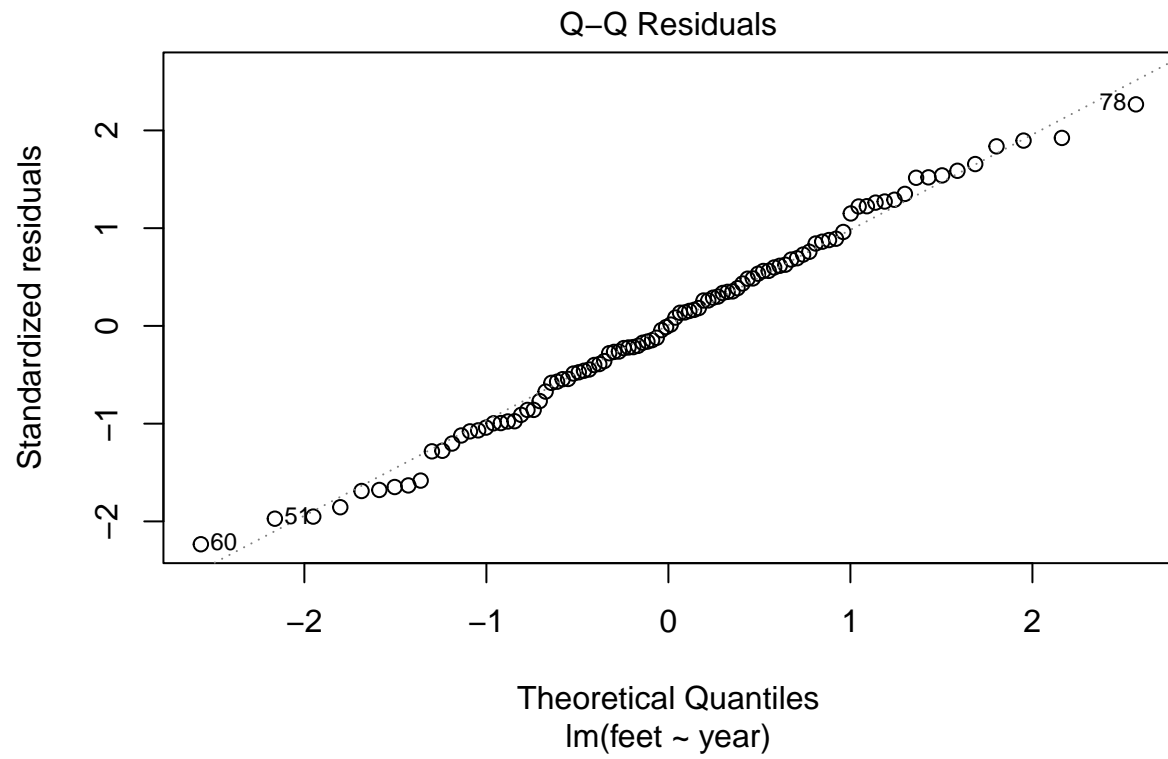
```
##  
##  Shapiro-Wilk normality test  
##  
## data:  huron$feet  
## W = 0.98492, p-value = 0.3271
```

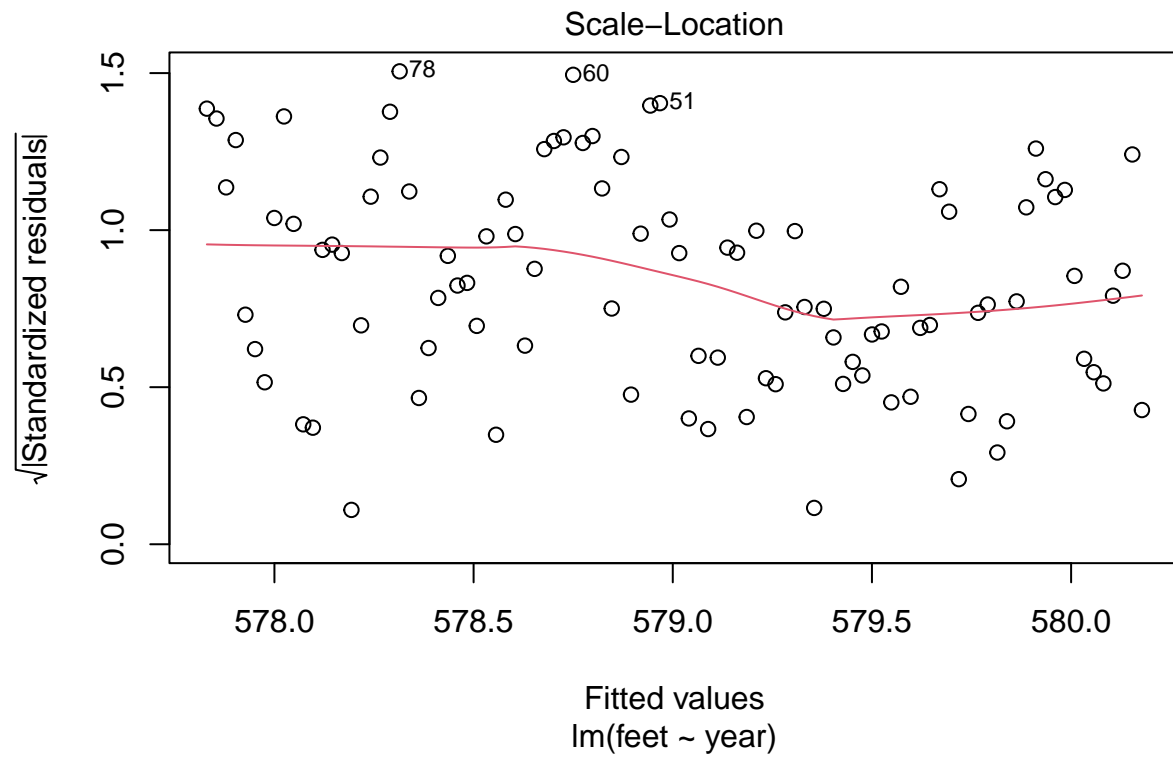
Man erkennt, dass die Daten annähernd Normalverteilt (und daher unimodal) sind. Auch der Shapiro-Wilk normality test zeigt eine Normalverteilung mit einem p-Wert von 0.3271 an.

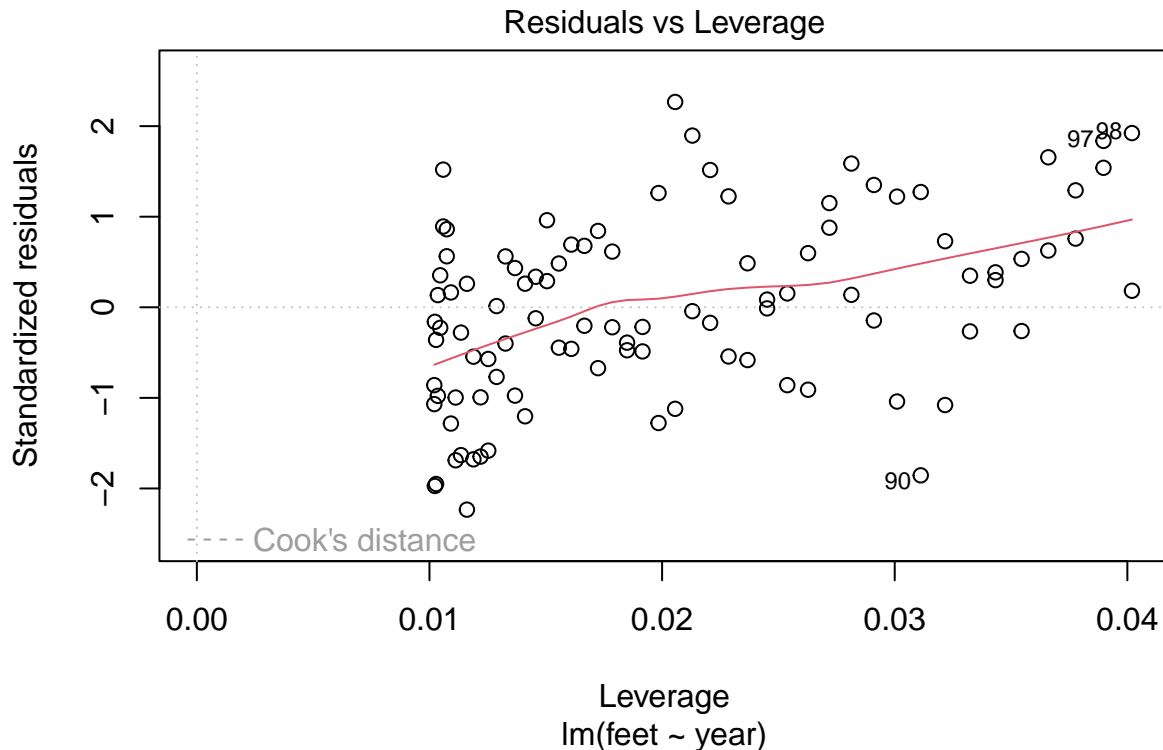
```
huron_lm1 <- lm(feet ~ year, data=huron)  
plot(huron_lm1)
```











Man sieht im Plot Residuals vs. Fitted deutlich, dass die Punkte nicht zufällig liegen, sondern Sinus-artig um die Linie verlaufen. Eine Voraussetzung der vier am Anfang genannten Voraussetzungen für ein lineares Regressionsmodell, nämlich jenes, dass die Komponenten des Fehlerterms nicht korrelieren, trifft hier nicht zu. Die anderen 3 Voraussetzungen sind hier erfüllt.

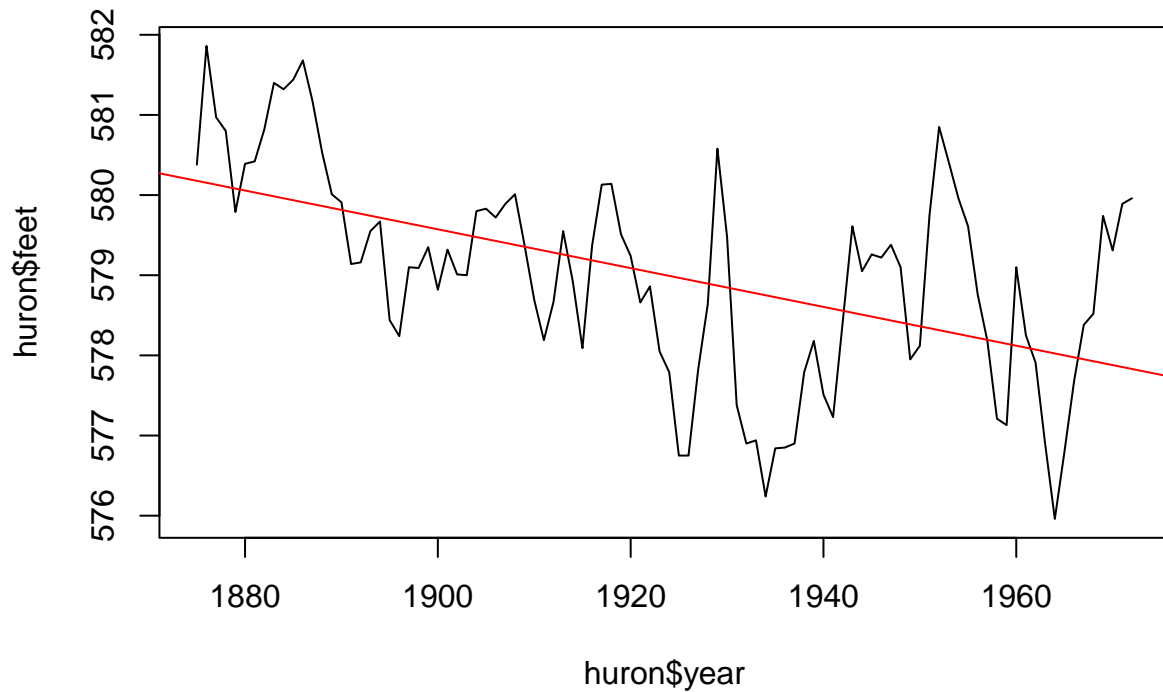
Im QQ-Plot sieht man keine Ausreißer oder gegen Normalverteilung sprechende Werte. Und keiner der Werte deutet darauf hin, als Hebel zu fungieren. Wir setzen trotzdem ein lineares Modell an, dass die Änderung des Pegels über die Zeit zeigt:

```
model <- lm(formula = feet ~ year, data=huron)
summary(model)
```

```
##
## Call:
## lm(formula = feet ~ year, data = huron)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50997 -0.72726  0.00083  0.74402  2.53565
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 625.554918   7.764293  80.568  < 2e-16 ***
## year        -0.024201   0.004036  -5.996 3.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.13 on 96 degrees of freedom
## Multiple R-squared:  0.2725, Adjusted R-squared:  0.2649
## F-statistic: 35.95 on 1 and 96 DF,  p-value: 3.545e-08
```

```
plot(huron$year, huron$feet, type="l")
abline(model, col="red")
```



Dieses Lineare Modell zeigt, dass der Wasserstand des Sees über die Jahre kontinuierlich sinkt. Der jährliche Rückgang beträgt gemäß unserem Modell 0.024 feet, also etwa 7.3 mm. Findet der Rückgang weiterhin so schnell statt, dann wäre der See in etwa 2400 Jahren ausgetrocknet.

## Pima Indians

Laden Sie den Datensatz 'Pima.tr' aus der library 'MASS'. Ermittle ein logistisches Regressionsmodell, dass das Auftreten von Diabetes ('type') durch die übrigen unabhängigen Variablen Alter (age), Anzahl der Schwangerschaften (npreg), BMI, Glukosespiegel (glu), Blutdruck (bp), familiäre Häufung von Diabetesfällen (ped) und Hautfaltendickemessung am Oberarm (skin) erklärt. Schreibe die Modellgleichung an und interpretiere die Werte der Koeffizienten im Kontext.

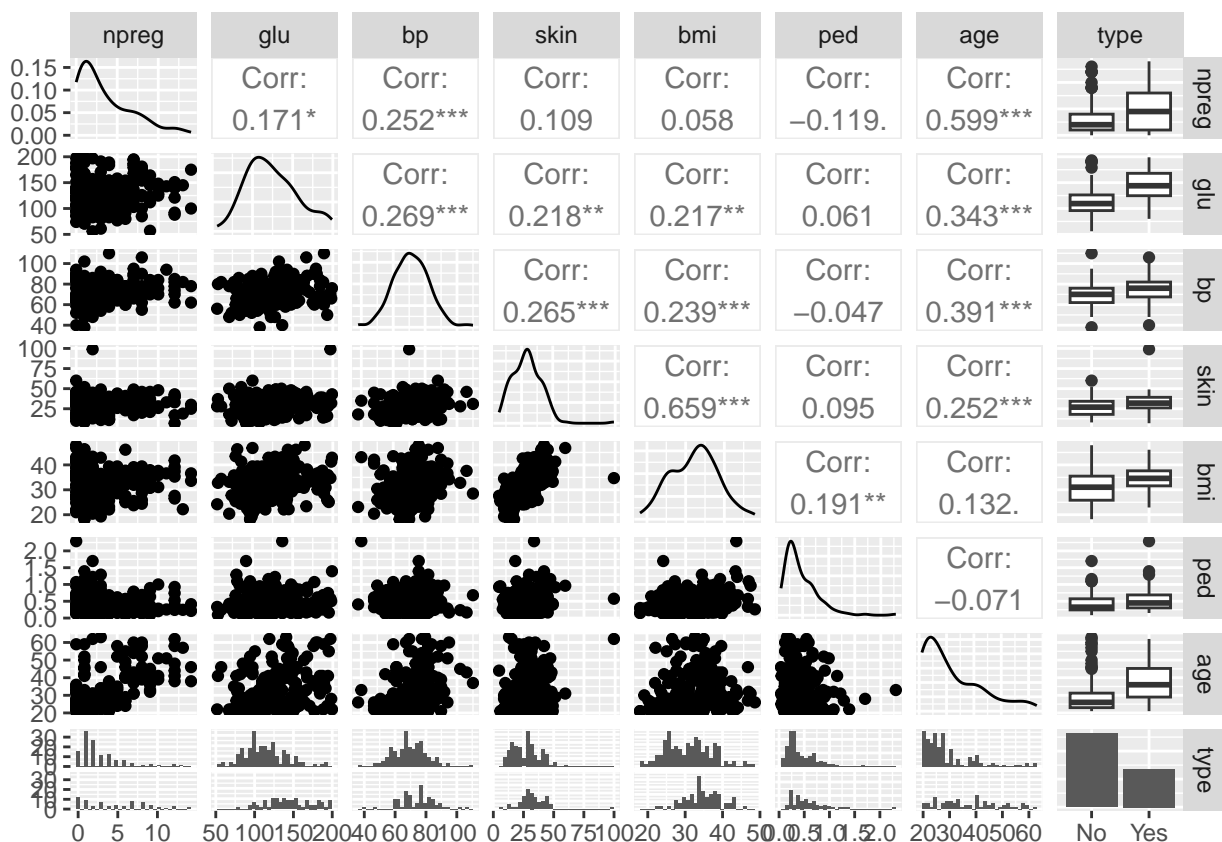
Ermitteln Sie die prädiktive Qualität des Modells mithilfe einer Receiver Operating Characteristic (ROC) Kurve. Führen Sie auch die False Positive, False Negative, True Positive und True Negative Raten in einer Tabelle (Konfusionsmatrix) an.

Eine Bevölkerung von Frauen im Alter von mindestens 21 Jahren, von Pima-Indianer-Herkunft und wohnhaft in der Nähe von Phoenix, Arizona, wurde gemäß den Kriterien der Weltgesundheitsorganisation auf Diabetes getestet. Die Daten wurden vom US National Institute of Diabetes and Digestive and Kidney Diseases gesammelt.

Wir wollen für dieses Beispiel ein logistisches Regressionsmodell entwickeln, da bei es sich bei Diabetes ja/nein um ein nominalskaliertes Kriterium handelt.

Im ersten Schritt wird die Multikollinearität mittels Scatterplotmatrix überprüft, um etwaige Fehler bei der Modellbildung im Vorhinein auszuschließen.

```
library(pROC)
library(MASS)
df <- Pima.tr
#View(df_secondRemoval)
ggpairs(df)
```



Hier wird gleich erkennbar, dass innerhalb der Variablen Multikollinearität besteht, was auch für logistische

Regression problematisch sein kann. Wie konzentrieren uns hierbei auf Werte um Bereich  $>0.5$  bzw  $0.8$ , bmi - skin (0.659) und age - npreg (0.599) sind die einzigen Variablen, deren Korrelationskoeffizienten in einen kritischen Bereich fallen. Daher haben wir uns dazu entschieden, skin und npreg aus dem Modell herauszutrimmen.

Für die Modellierung verwenden wir logit-Funktion, die log-Odds der Ergebnisse abbildet.

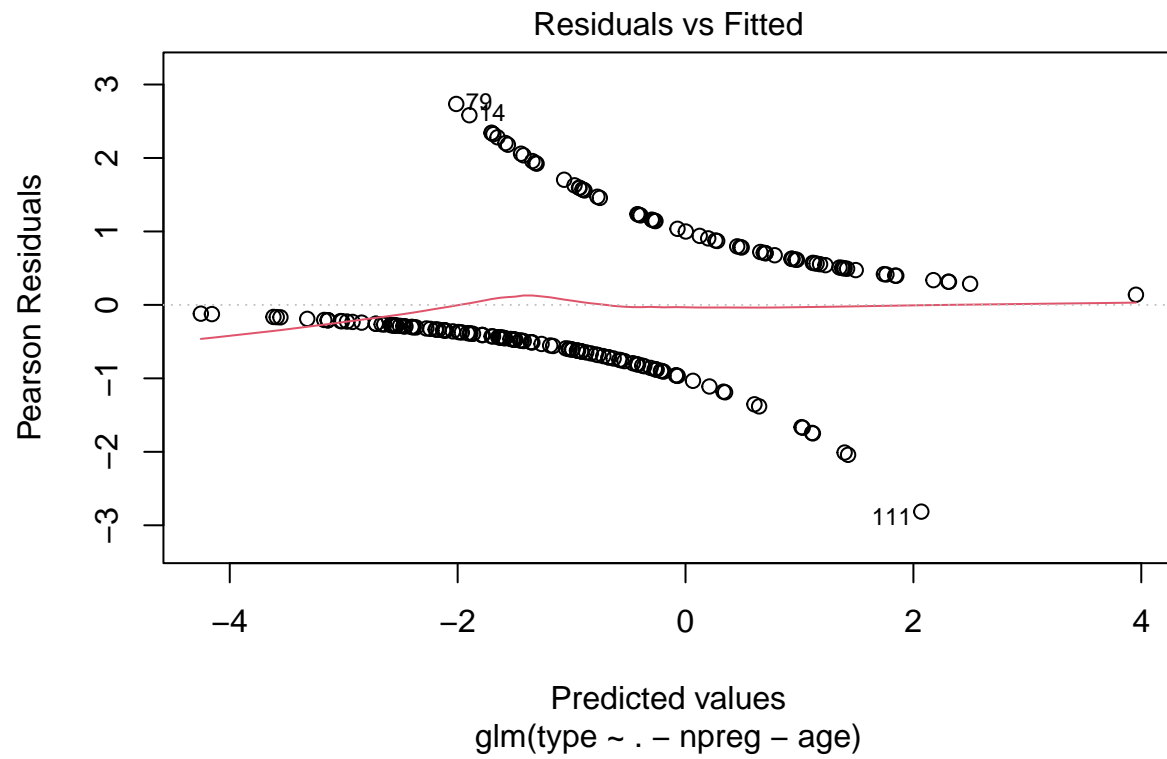
```
library(pROC)
library(MASS)

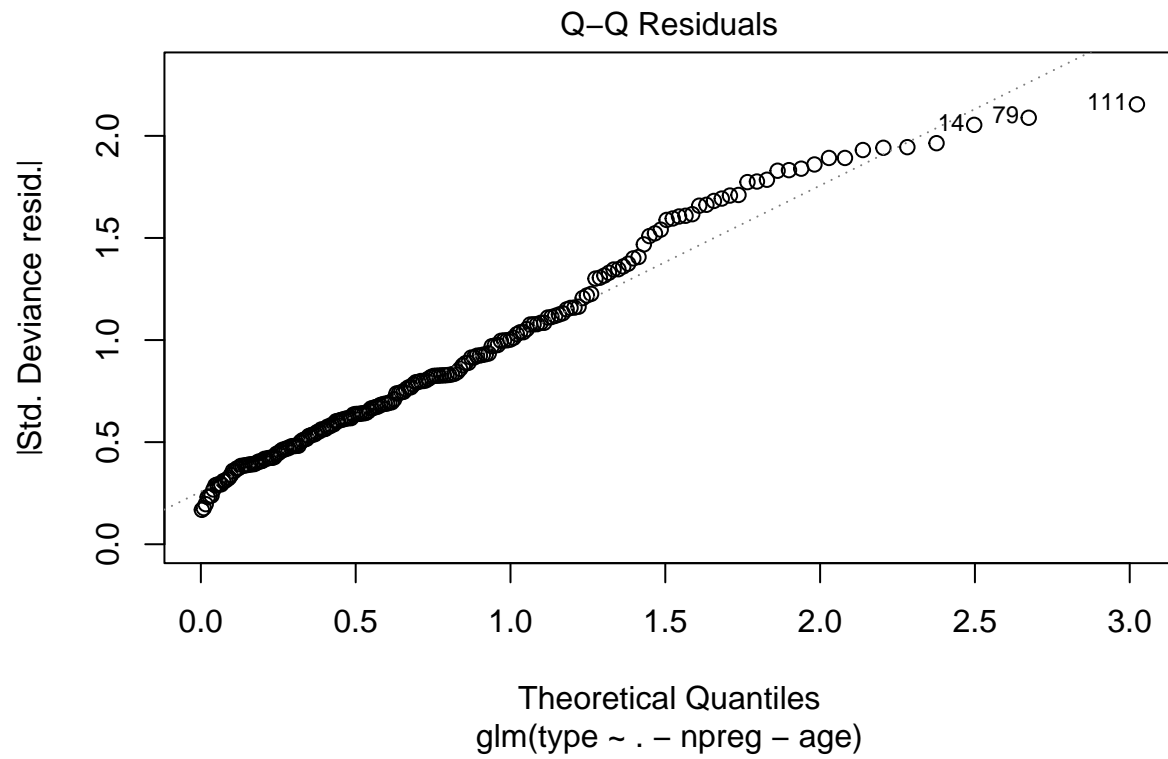
#summary(data)

df <- data.frame(Pima.tr)
modell <- glm(type ~ . - npreg - age ,data=df,family=binomial(link = "logit"))
summary(modell)

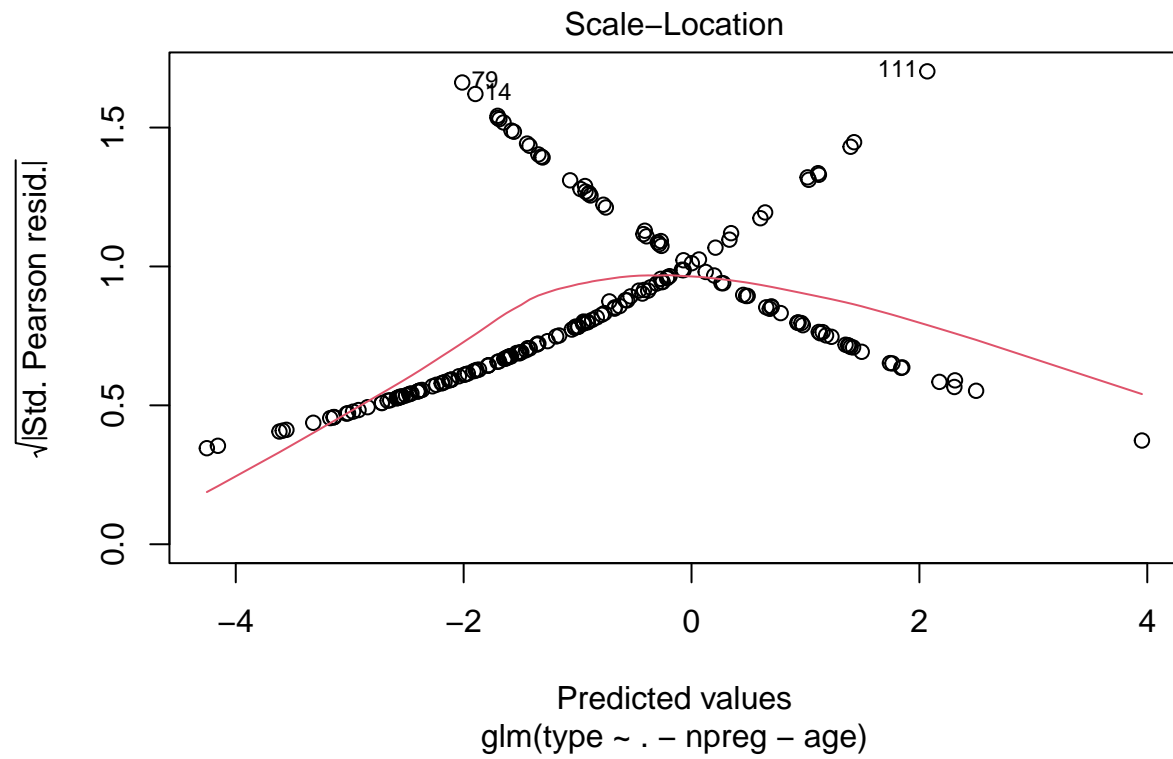
##
## Call:
## glm(formula = type ~ . - npreg - age, family = binomial(link = "logit"),
##      data = df)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.296518   1.693633  -5.489 4.04e-08 ***
## glu          0.035295   0.006566   5.376 7.63e-08 ***
## bp           0.015239   0.016793   0.907  0.3642
## skin         0.004617   0.021485   0.215  0.8298
## bmi          0.066066   0.040992   1.612  0.1070
## ped          1.459866   0.619486   2.357  0.0184 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 256.41  on 199  degrees of freedom
## Residual deviance: 191.91  on 194  degrees of freedom
## AIC: 203.91
##
## Number of Fisher Scoring iterations: 5

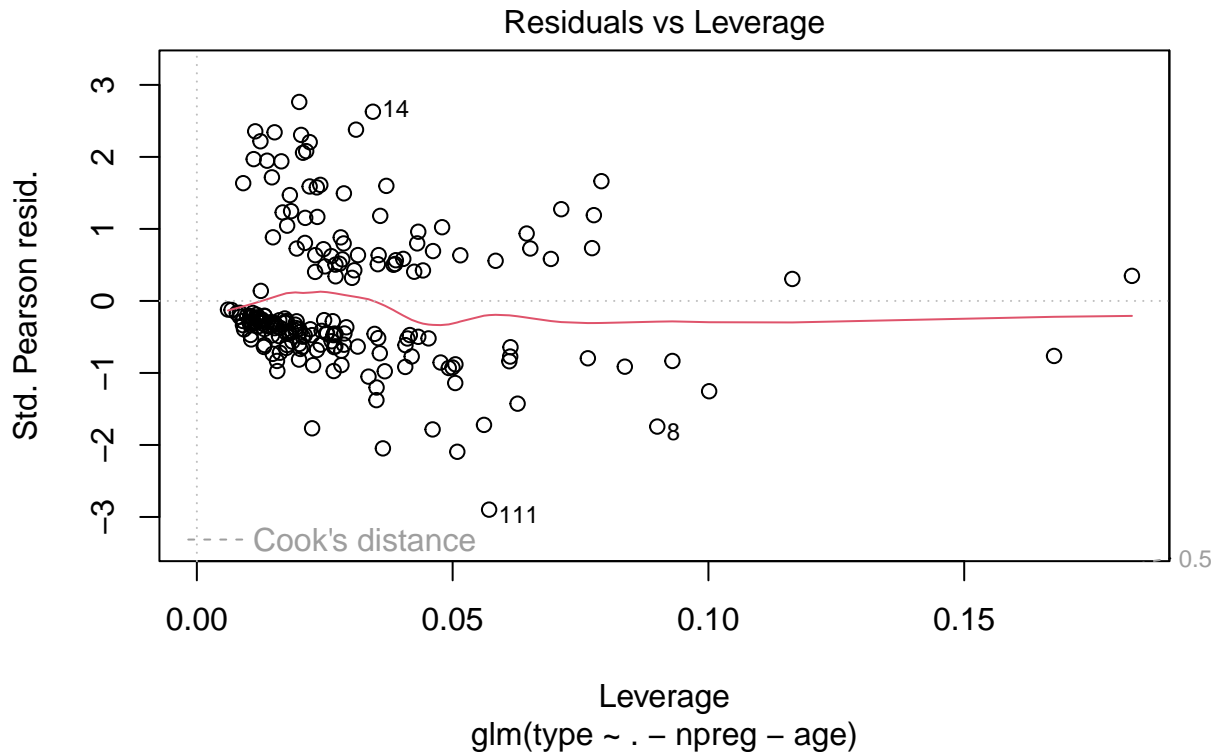
plot(modell)
```











```
#neu classification table
#data.predictions <- predict(modell, type = "response")
# classification <- data.frame(response =df$type)
```

Mittels der glm Funktion wird ein logistisches Modell aus den Prädikatorenn modelliert. Folgendes Modell wird durch die errechneten Koeffizienten beschrieben:

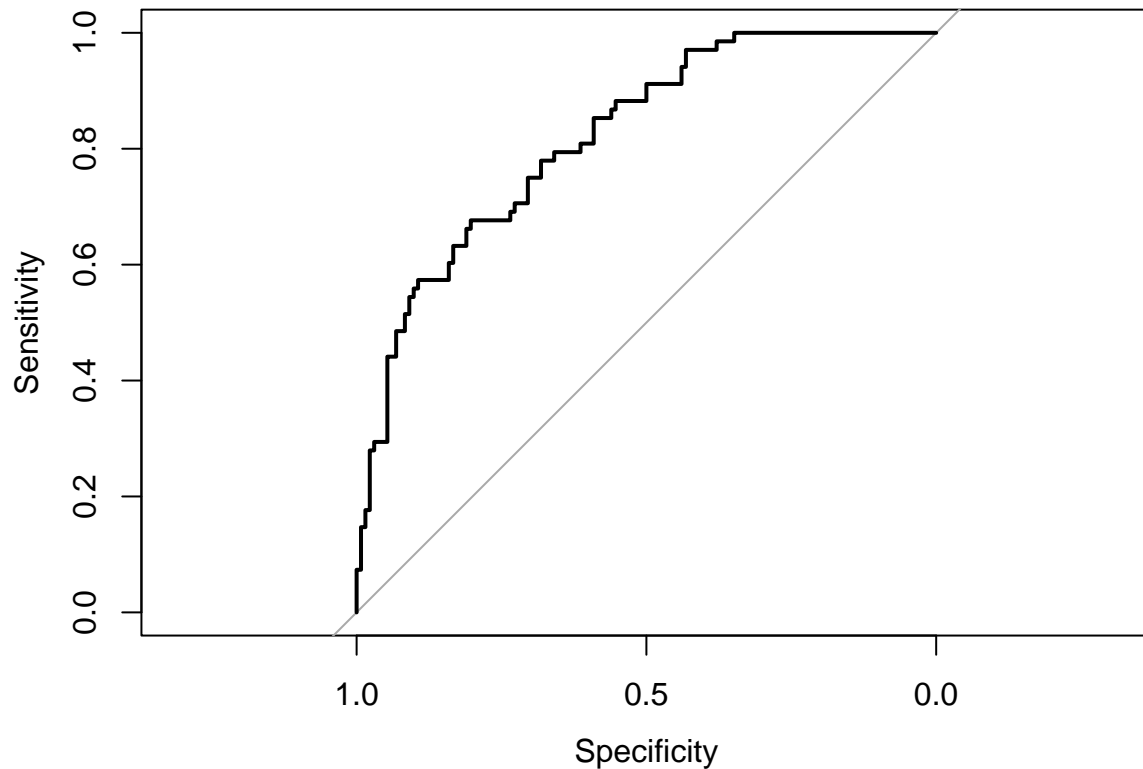
$$type = -9.296518 + (0.066066 * bmi) + (0.035295 * glu) + (0.015239 * bp) + (0.004617 * skin) + (1.459866 * ped)$$

Die Analyse der Residuenplots entfällt bei einer logistischen Regressionsanalyse. Aus der Summary können die Koeffizienten des Modells abgelesen werden. Hierbei sieht man, dass nur Glucose und der familiäre Hintergrund signifikant sind. Die prädiktive Qualität des Modells wird nun mithilfe einer ROC Kurve ermittelt. Diese stellt die Ergebnisse der Prädiktion durch das Modell grafisch dar, wobei hohe AUC Werte für eine gute Trefferquote sprechen. In unserem Fall beträgt die AUC 0.8205214, was dafür spricht, dass unser Modell eine gute Vorhersagequalität hat.

```
#neu classification table

classification <- data.frame(response =df$type)
View(classification)
predictions <- predict(modell, type = "response")
roc_curve <- roc(Pima.tr$type, predictions)
plot(roc_curve, main = "ROC Kurve -- Logistische Regression für Modell zur prädiktion von Diabetis")
```

## OC Kurve -- Logistische Regression für Modell zur prädiktion von Di



```
auc_value <- auc(roc_curve)
cat("AUC (Area Under the Curve):", auc_value, "\n")
```

```
## AUC (Area Under the Curve): 0.8205214
```

Abschließend erstellen wir eine Konfusionsmatrix und Klassifizieren unsere Daten ausgehend von unserem Modell. Als Treshold setzen wir 0.5

```
#Wir setzen als ja/nein-Kriterium
data.predictions <- predict(modell, type = "response")
classification <- data.frame(response =df$type)

threshold <- 0.5
data.pred.class <- ifelse(data.predictions >= threshold, 1, 0)
table(data.pred.class)
```

```
## data.pred.class
##    0    1
## 149   51
```

```
table(Pima.tr$type)
```

```
##
```

```
## No Yes
## 132 68
```

```
classification.matrix <- table(PRED = data.pred.class, ACTUAL = Pima.tr$type)
classification.matrix
```

```
##      ACTUAL
## PRED No Yes
##    0 119 30
##    1  13 38
```

```
colnames(classification.matrix) <- c("Neg", "Pos")
rownames(classification.matrix) <- c("Neg", "Pos")
addmargins(classification.matrix)
```

```
##      ACTUAL
## PRED Neg Pos Sum
## Neg 119 30 149
## Pos  13 38  51
## Sum 132 68 200
```

Zusammenfassend kann man aus der erstellten Tabelle (Konfusionsmatrix) folgende Informationen entnehmen:

True negative = 119 True positive = 28 False negative = 30 False positive = 13

Insgesamt ist das Modell also eher spezifisch, und klassifiziert häufiger falsch-negativ als falsch-positiv. -  
>Höhere Spezifität als Sensitivität