Statistische Modellierung

Kacper Bohaczyk

2024-12-22

Aufgabe 1

Das Dataset zeigt eine Auswahl an Daten über die 50 Staaten der USA an.
Bevölkerung in 100 Tausenden,
Einkommen pro Person in USD,
Analphabetenrate als Prozent der Bevölkerung,
Lebenserwartung in Jahren,
Mord und Totschlag pro 100 Tausend Einwohner,
Menschen mit High-School abschlüssen als Prozent der Bevölkerung,
Tage mit Temperatur unter 0° C in großen Städten (1931-1960),
Fläche der Staaten in Quadratmeilen.

```
?state.x77
summary(state.x77)
```

```
##    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.: 81162
##  Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432
```

```
invisible(library(lmtest))
```

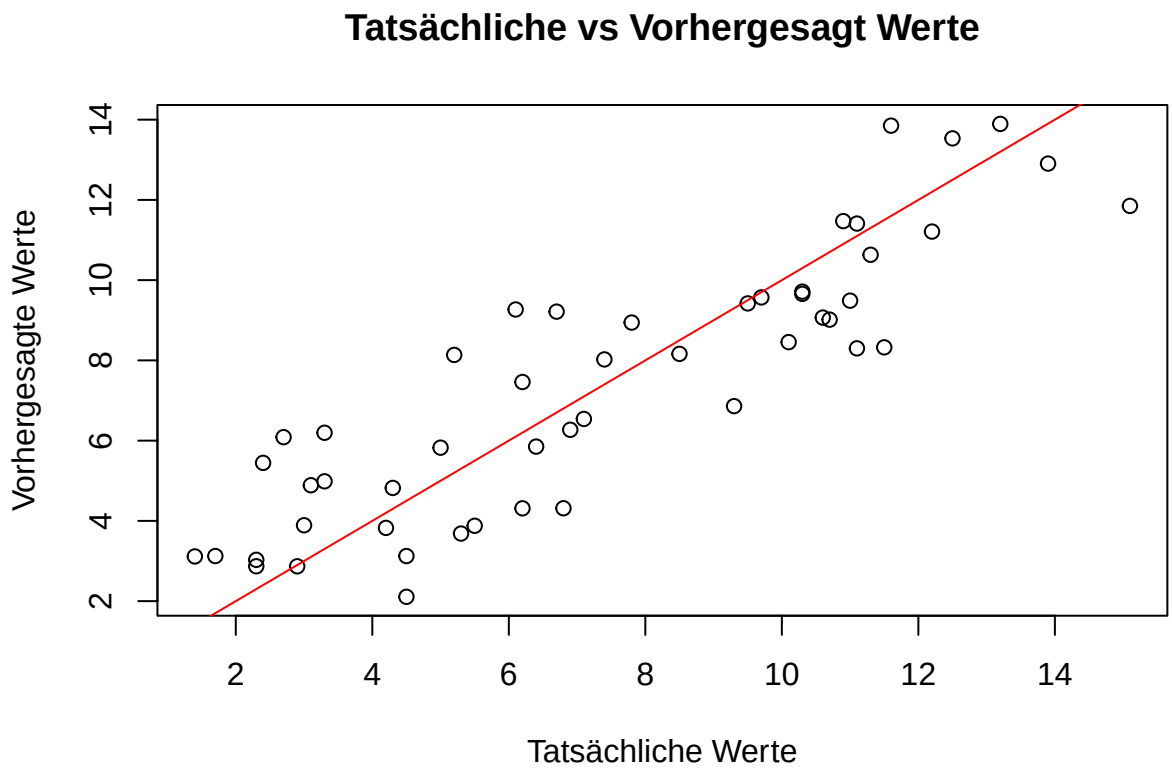
```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
invisible(library(ggplot2))
```

```r
modell <- lm(state.x77[,"Murder"] ~ state.x77[,"Life Exp"] + state.x77[,"Population"] + state.x77[,"Illi
predicted_values <- predict(modell)
plot(state.x77[,"Murder"], predicted_values,
     xlab = "Tatsächliche Werte", ylab = "Vorhergesagte Werte",
     main = "Tatsächliche vs Vorhergesagt Werte")
abline(0, 1,col = "red")#
```




```r
invisible(library(relaimpo))
```

```
## Loading required package: MASS
```

```
## Loading required package: boot
```

```
## Loading required package: survey
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

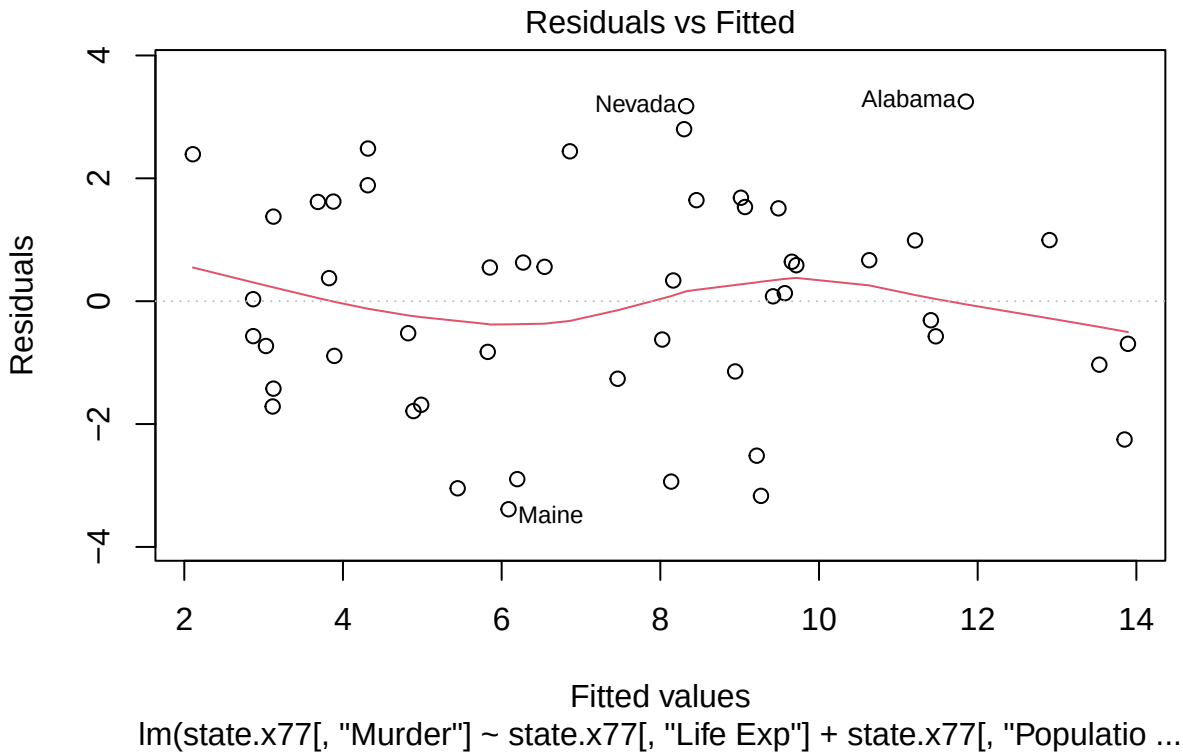
```
## Loading required package: survival
```

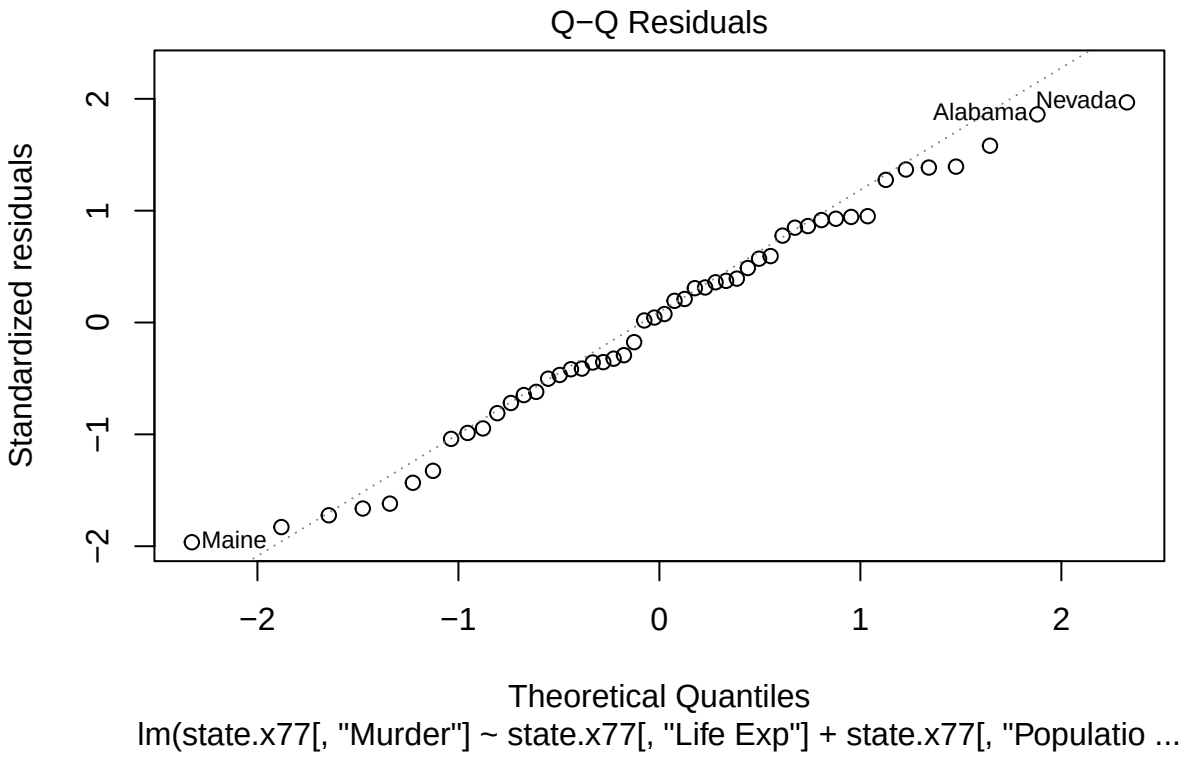
```
##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

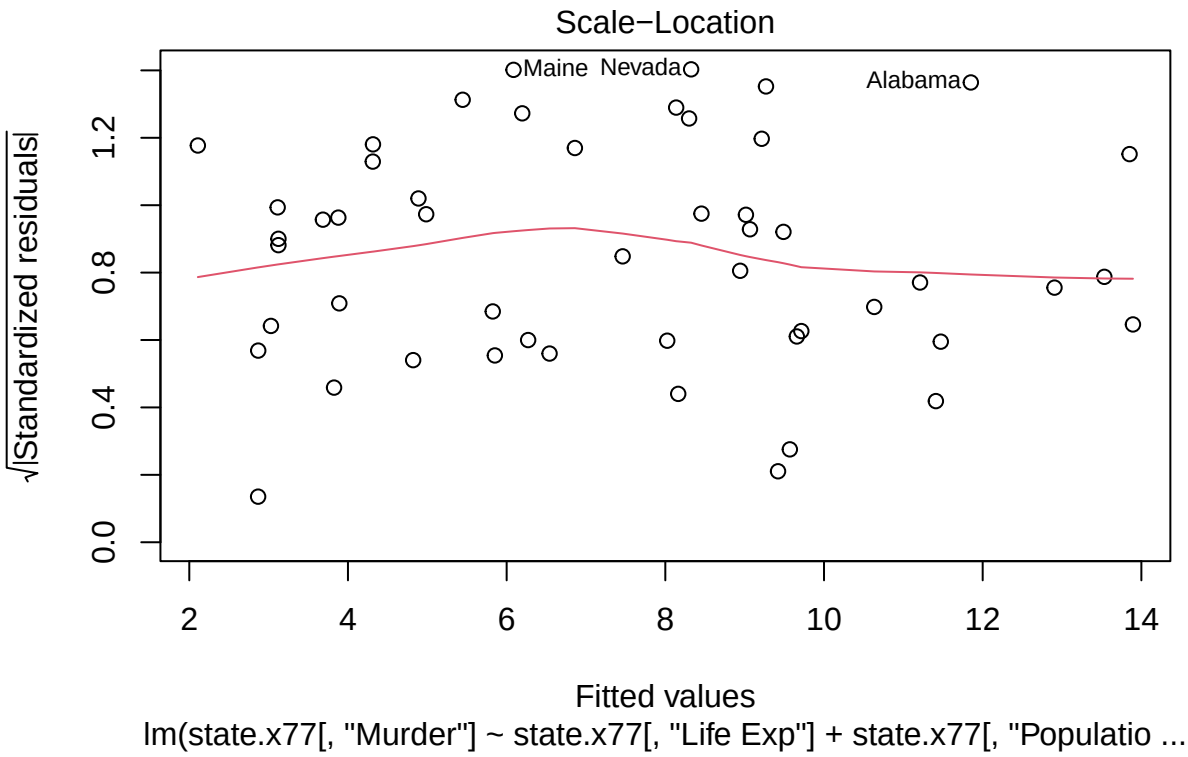
```
calc.relimp(modell, type = "lmg")
```

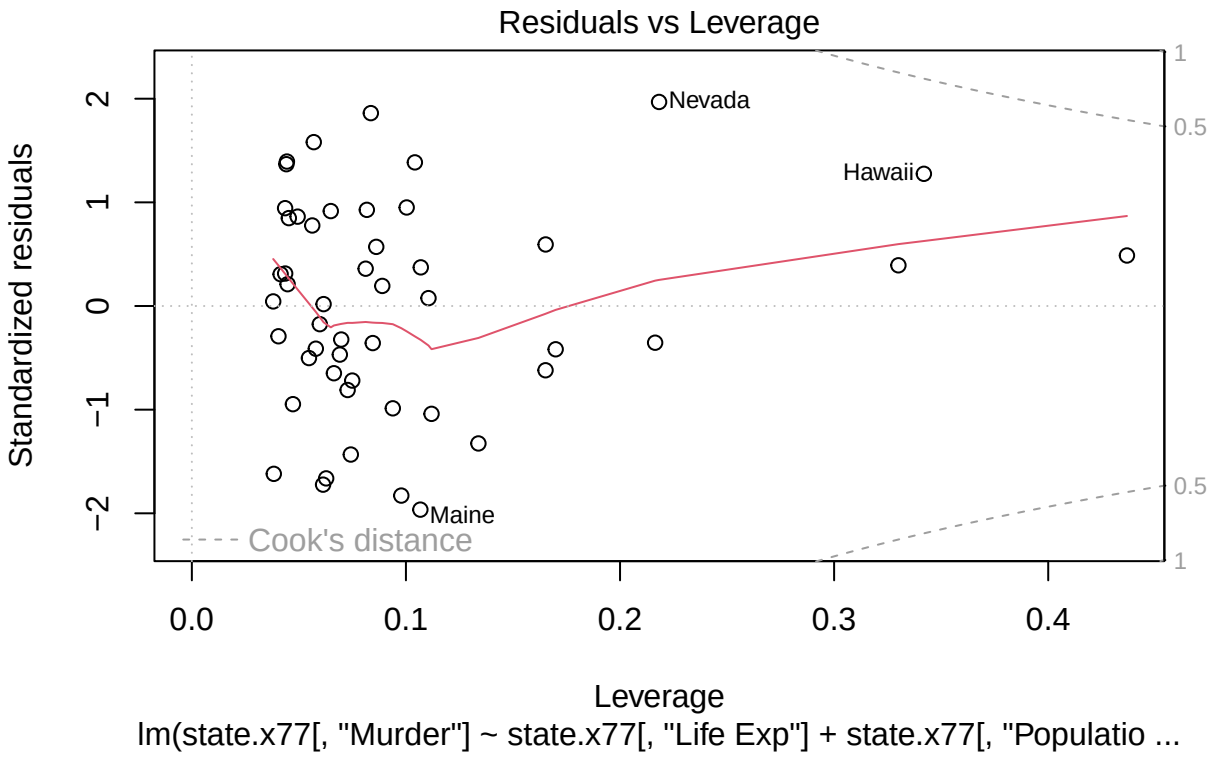
```
## Response variable: state.x77[, "Murder"]
## Total response variance: 13.62747
## Analysis based on 50 observations
##
## 4 Regressors:
## state.x77[, "Life Exp"] state.x77[, "Population"] state.x77[, "Illiteracy"] state.x77[, "Income"]
## Proportion of variance explained by model: 77.58%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##                                  lmg
## state.x77[, "Life Exp"]   0.39183297
## state.x77[, "Population"] 0.08847560
## state.x77[, "Illiteracy"] 0.27083102
## state.x77[, "Income"]     0.02463989
##
## Average coefficients for different model sizes:
##
##                                    1X           2Xs           3Xs
## state.x77[, "Life Exp"]   -2.1473010533 -1.9407471296 -1.7310151950
## state.x77[, "Population"]  0.0002841468  0.0002679824  0.0002308627
## state.x77[, "Illiteracy"]  4.2574567427  3.6154586618  2.9314818960
## state.x77[, "Income"]     -0.0013822330 -0.0003597256  0.0002550537
##                                   4Xs
## state.x77[, "Life Exp"]   -1.5659396965
## state.x77[, "Population"]  0.0002058589
## state.x77[, "Illiteracy"]  2.2650087139
## state.x77[, "Income"]      0.0004523656
```

```
plot(modell)
```


Residuals vs Fitted
Residuals
Fitted values
lm(state.x77[, "Murder"] ~ state.x77[, "Life Exp"] + state.x77[, "Populatio ...
Nevada
Alabama
Maine
4
2
0
−2
−4
2
4
6
8
10
12
14


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(state.x77[, "Murder"] ~ state.x77[, "Life Exp"] + state.x77[, "Populatio ...
Maine
Alabama
Nevada
−2
−1
0
1
2

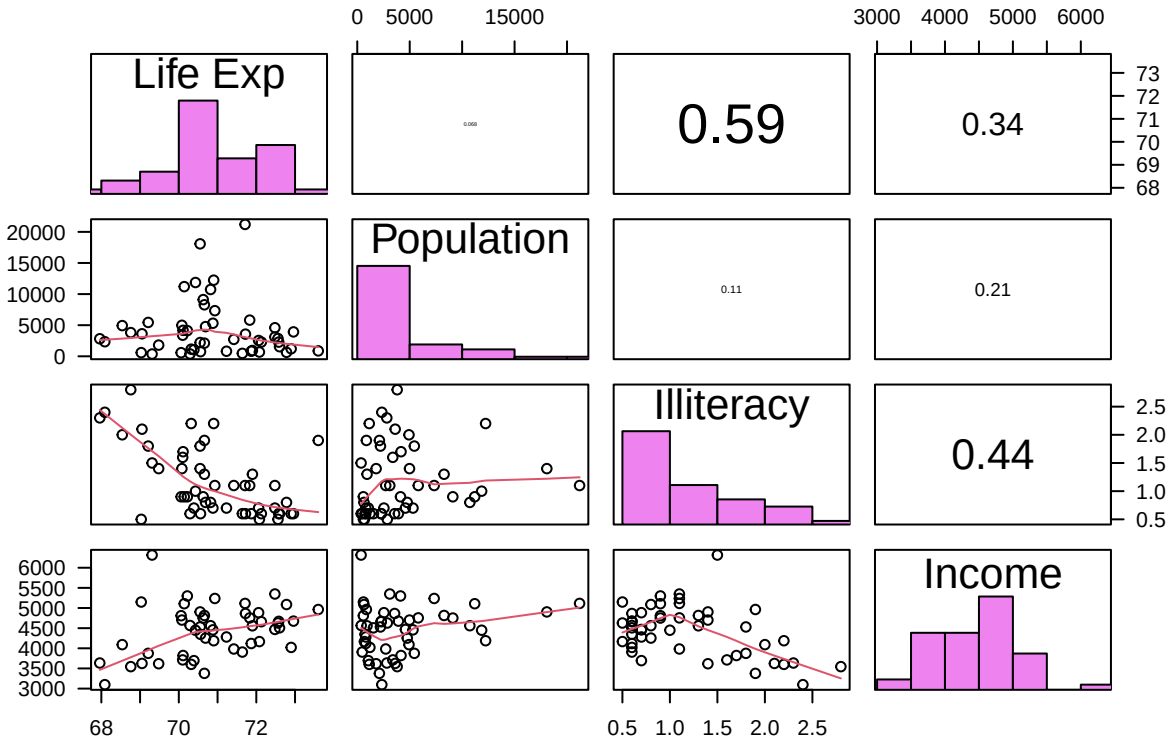

Scale−Location
Maine
Nevada
Alabama
√|Standardized residuals|
0.0
0.4
0.8
1.2
2
4
6
8
10
12
14
Fitted values
lm(state.x77[, "Murder"] ~ state.x77[, "Life Exp"] + state.x77[, "Populatio ...




```r
 ## Plottet die 4 Diagramme


panel.hist <- function(x, ...) {
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5))
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "violet", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

# Create a matrix of scatterplots
pairs(state.x77[, c("Life Exp" , "Population" , "Illiteracy", "Income")],
      lower.panel = panel.smooth,
      upper.panel = panel.cor,
```

```
      diag.panel = panel.hist,
      las=1)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



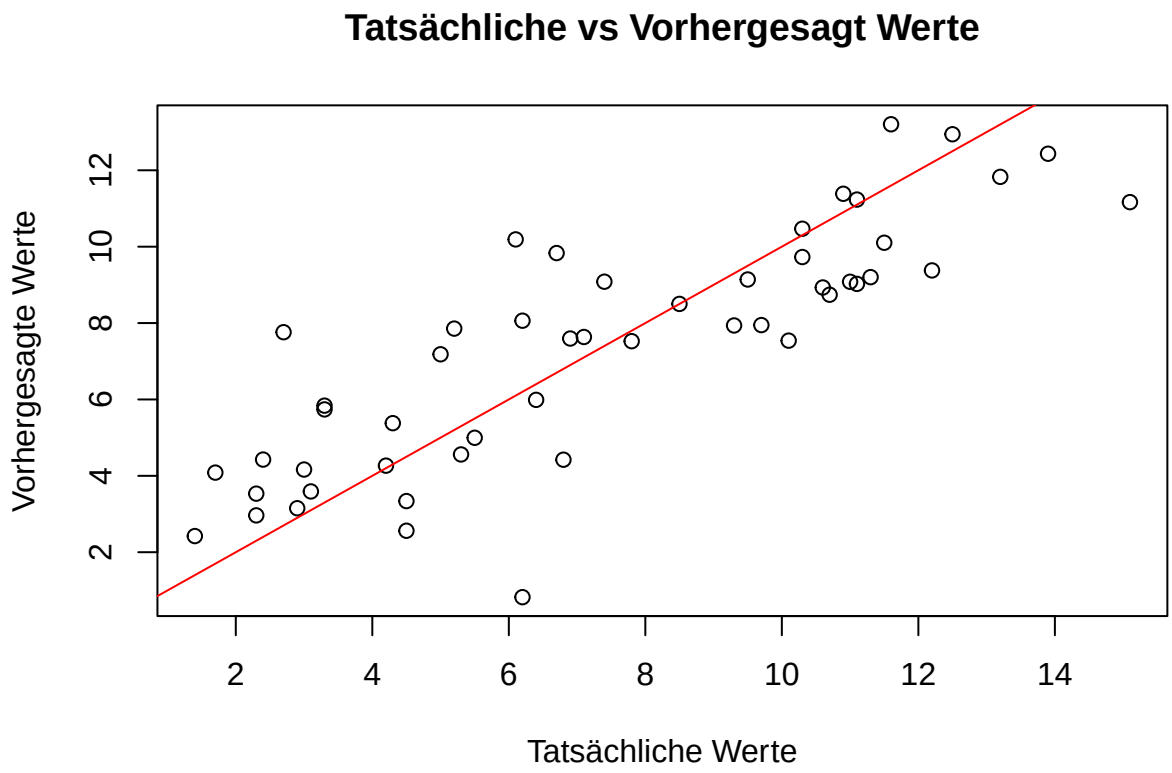
Das Modell vorhersagt sinnvoll die Daten.

Die Residuen zeigen eine Normalverteilung, bedinden sich um den Wert 0 und weisen kein Muster auf.

Zwischen Analphabetismus und Lebenserwartung gibt es Multikollinearität.

Modellselektion der relevanten erklärenden Variablen durch

```
modellNew <- lm(state.x77[,"Murder"] ~ state.x77[,"Population"] + state.x77[,"Life Exp"] + state.x77[,"
predicted_values <- predict(modellNew)
plot(state.x77[,"Murder"], predicted_values,
     xlab = "Tatsächliche Werte", ylab = "Vorhergesagte Werte",
     main = "Tatsächliche vs Vorhergesagt Werte")
abline(0, 1, col="red")#
```



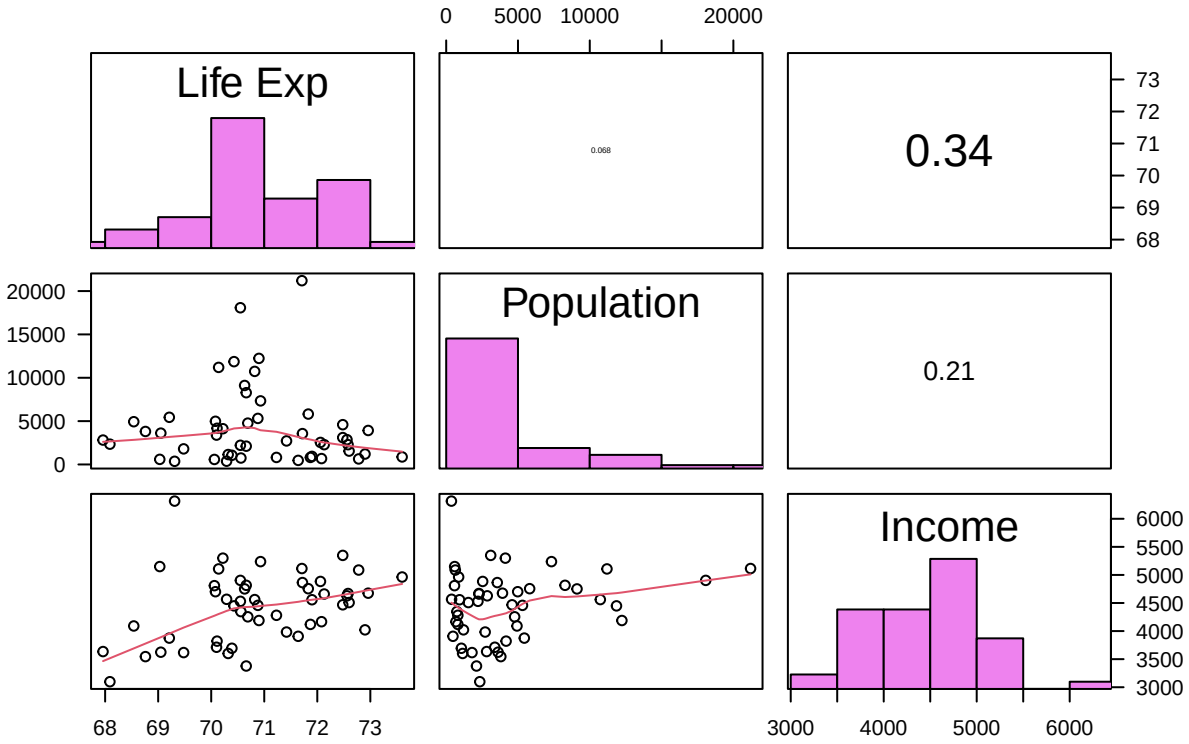

```
invisible(library(relaimpo))
calc.relimp(modellNew, type = "lmg")
```

```
## Response variable: state.x77[, "Murder"]
## Total response variance: 13.62747
## Analysis based on 50 observations
##
## 3 Regressors:
## state.x77[, "Population"] state.x77[, "Life Exp"] state.x77[, "Income"]
## Proportion of variance explained by model: 69.57%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
```

```
##                                   lmg
## state.x77[, "Population"] 0.10839742
## state.x77[, "Life Exp"]   0.55319097
## state.x77[, "Income"]     0.03414432
##
## Average coefficients for different model sizes:
##
##                                    1X           2Xs           3Xs
## state.x77[, "Population"]  0.0002841468  0.0002898805  0.0002487147
## state.x77[, "Life Exp"]   -2.1473010533 -2.1388329147 -2.0550437649
## state.x77[, "Income"]     -0.0013822330 -0.0008261553 -0.0002309284
```

```r
panel.hist <- function(x, ...) {
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5))
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "violet", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

# Create a matrix of scatterplots
pairs(state.x77[, c("Life Exp" , "Population" , "Income")],
      lower.panel = panel.smooth,
      upper.panel = panel.cor,
      diag.panel = panel.hist,
      las=1)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



Das Modell vorhersagt sinnvoll die Daten.

Die Residuen zeigen eine Normalverteilung, bedinden sich um den Wert 0 und weisen kein Muster auf.

Zwischen Analphabetismus und Lebenserwartung gibt es Multikollinearität.

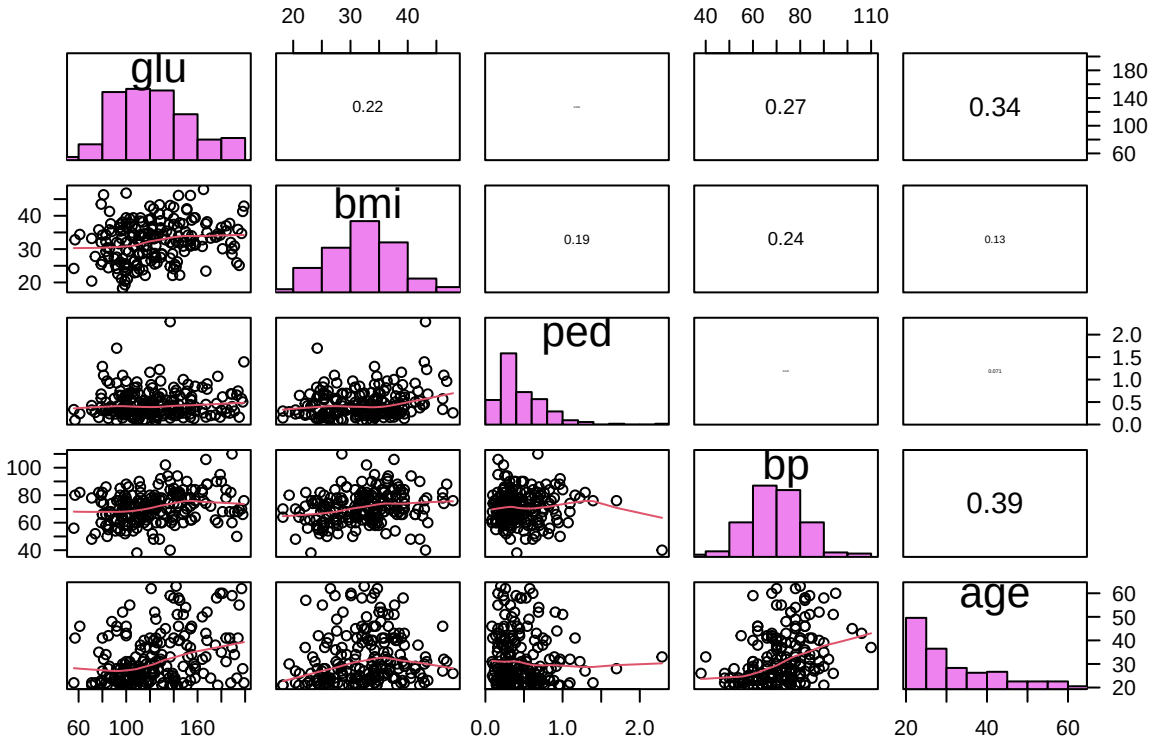
Zwischen Life Exp und Income. gibt es Multikollinearität.

Einkommen ist ein unwichtiger Parameter

Aufgabe 2

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```

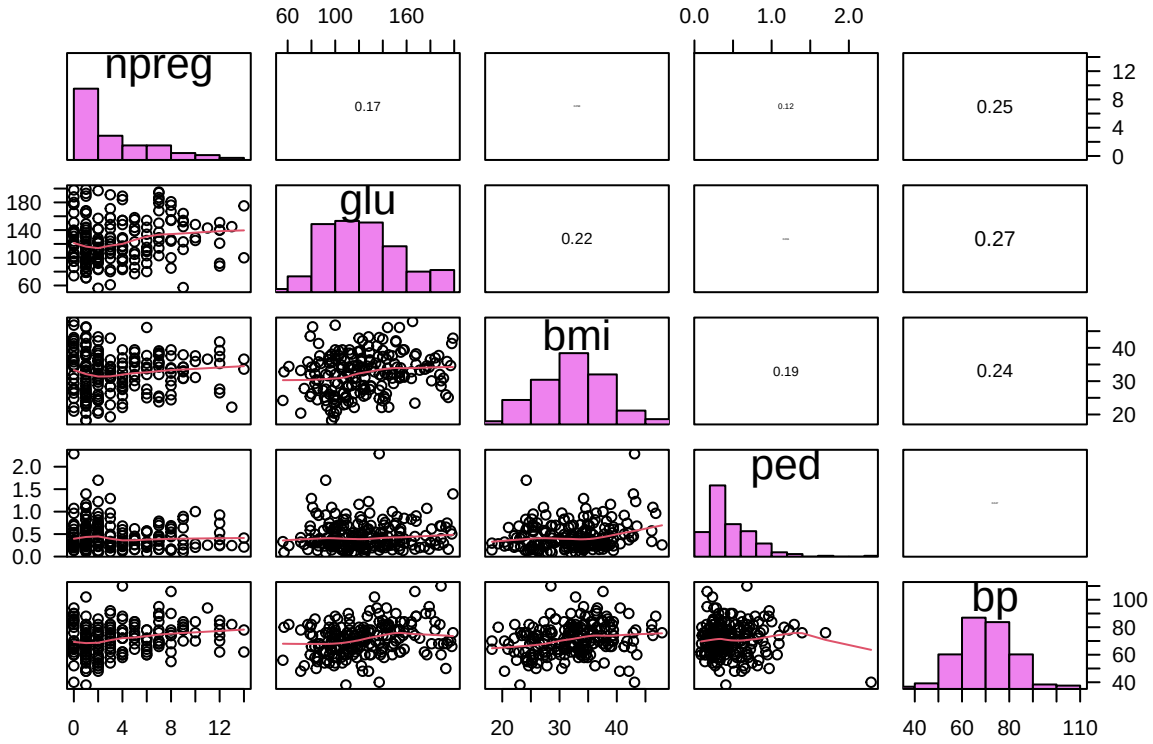



```
##
## Call:
## glm(formula = type ~ skin + glu + bmi + ped + bp + age + npreg,
##     family = binomial(link = "logit"), data = Pima.tr)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.773062   1.770386  -5.520 3.38e-08 ***
## skin        -0.001917   0.022500  -0.085  0.93211
## glu          0.032117   0.006787   4.732 2.22e-06 ***
## bmi          0.083624   0.042827   1.953  0.05087 .
```

```
## ped          1.820410   0.665514   2.735  0.00623 **
## bp          -0.004768   0.018541  -0.257  0.79707
## age          0.041184   0.022091   1.864  0.06228 .
## npreg        0.103183   0.064694   1.595  0.11073
## ---
## 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: 178.39  on 192  degrees of freedom
## AIC: 194.39
##
## Number of Fisher Scoring iterations: 5
```

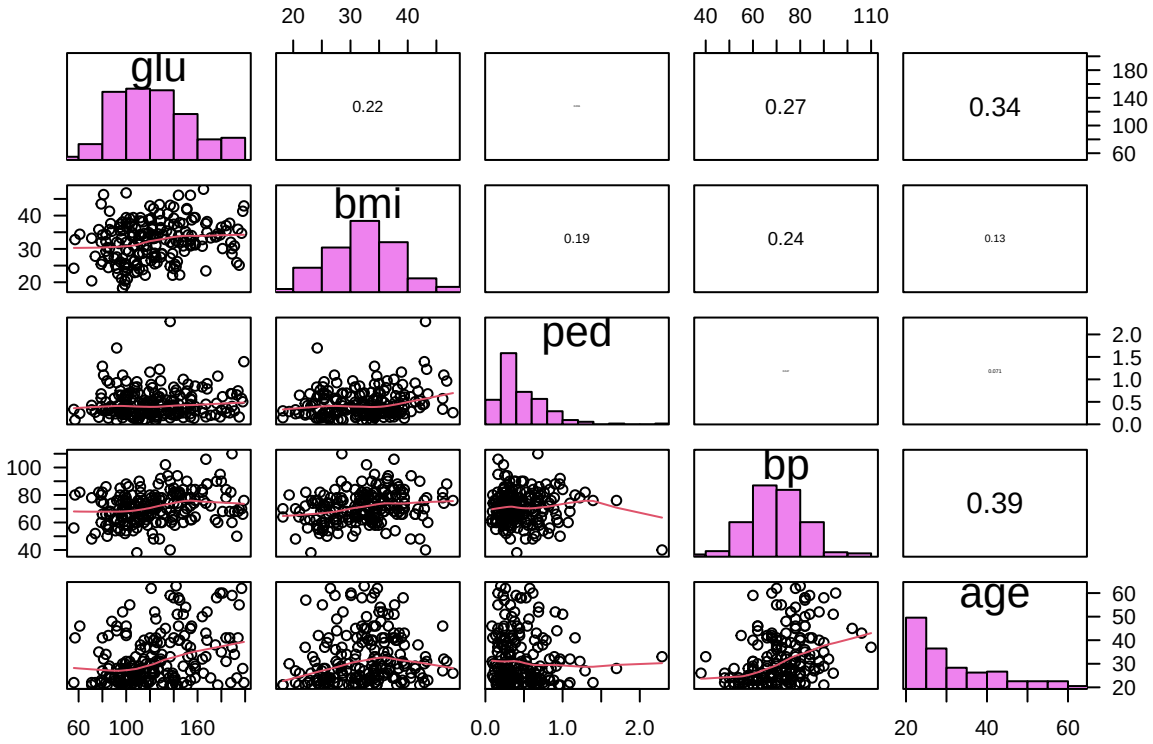
```
##  (Intercept)         skin          glu          bmi          ped           bp
## -9.773061533 -0.001916632  0.032116823  0.083623912  1.820410367 -0.004767542
##          age        npreg
##  0.041183529  0.103183427
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```




```
##
## Call:
## glm(formula = type ~ skin + glu + ped + bp + npreg, family = binomial(link = "logit"),
##     data = Pima.tr)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.861770   1.448189  -5.429 5.68e-08 ***
## skin         0.027694   0.017406   1.591  0.11159
## glu          0.034495   0.006607   5.221 1.78e-07 ***
## ped          1.833900   0.623578   2.941  0.00327 **
## bp           0.006463   0.017734   0.364  0.71554
## npreg        0.160033   0.054022   2.962  0.00305 **
## ---
## 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: 185.39  on 194  degrees of freedom
## AIC: 197.39
##
## Number of Fisher Scoring iterations: 5

##  (Intercept)         skin          glu          ped           bp        npreg
## -7.861769961  0.027693775  0.034494923  1.833900379  0.006462532  0.160033092
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```




```
##
## Call:
## glm(formula = type ~ glu + bmi + ped + bp + age, family = binomial(link = "logit"),
##     data = Pima.tr)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.762937   1.689986  -5.777 7.61e-09 ***
## glu          0.031584   0.006752   4.677 2.90e-06 ***
## bmi          0.078722   0.032814   2.399  0.01644 *
## ped          1.729202   0.660093   2.620  0.00880 **
## bp          -0.005174   0.018245  -0.284  0.77672
## age          0.060535   0.018901   3.203  0.00136 **
## ---
## 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: 181.00  on 194  degrees of freedom
## AIC: 193
##
## Number of Fisher Scoring iterations: 5

##  (Intercept)          glu          bmi          ped           bp          age
## -9.762936794  0.031583880  0.078722179  1.729202382 -0.005174239  0.060534977
```

Erstes Modell mit Kolinearität: f = -9.773+0.103*npreg+0.032*glu-0.005*bp-0.002*skin+0.084*bmi+1.820*ped+0.041*age

Zweites Modell (ohne age & bmi): f = -7.861769961 + skin*0.027693775 + glu*0.034494923 + ped*1.833900379 + bp*0.006462532 + npreg*0.160033092

Drittes Modell (ohne npreg & skin): f = -9.762936794 + glu* 0.031583880 + bmi*0.078722179 + ped*1.729202382 + bp-*0.005174239 + age*0.060534977

Das zweite Model ist l am sinnvollsten, da es am wenigsten Kolinearität aufweist.

Die einzelnen Daten wirken sich beim zweiten Modell so aus:

Pro mm Tricep Fettschicht dicke erhöt sich die Wahrscheinlichkeit um 28%

Pro Miligramm Plama Glucose pro Deziliter steigt die Diabetis Wahrscheinlickeit um 35%

Pro Erhöhung der “diabetes pedigree function.” um 1 steigt die Wahrscheinlichkeit um 625%

Pro Erhöhung des Blutdrucks um 1 mmHg steigt die Wahrscheinlichkeit um 0.6%

Pro Schwangerschaft erhöt sich das Risisko um 17%

Diese Werte im Kontext bedeuten, dass der mit Abstand wichtigste Wert der der DPF, also wie stark man für Diabetes anfällig ist.

In der Summary sieht man, dass sich die Wichtigkeit der Daten stark ändert wenn man die Kolinearität entfernt.