```r
set.seed(42)
```

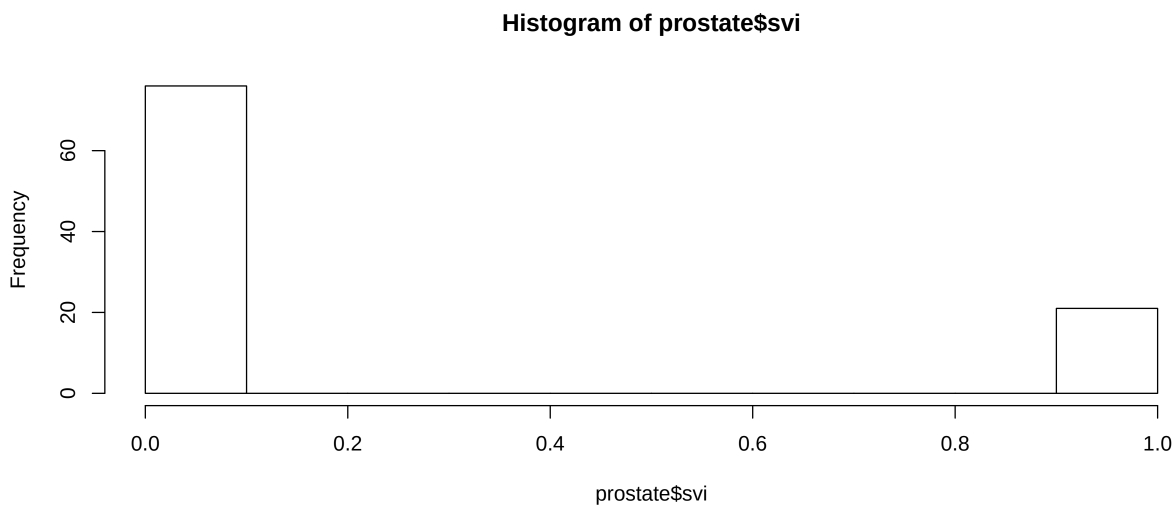
Arbeitsblatt 6

Aufgabe 1

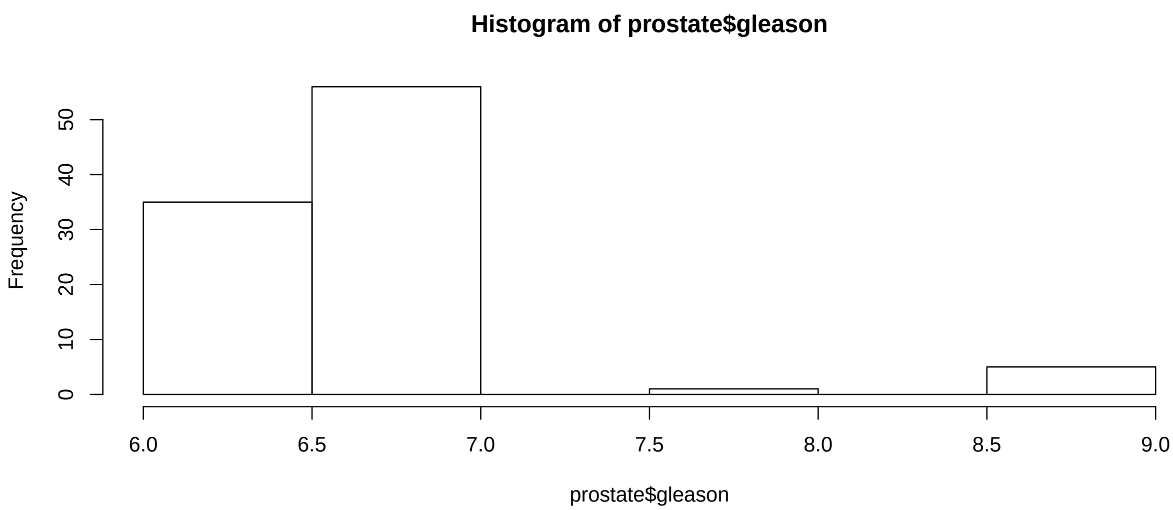
a)

```r
load("prostate.RData")

hist(prostate$age)
```

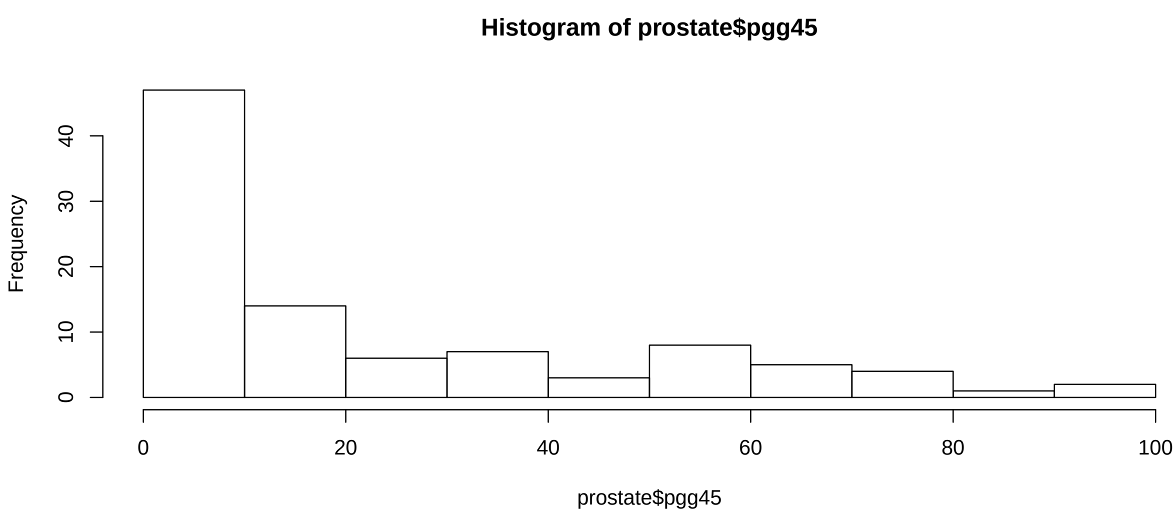

```r
hist(prostate$svi)
```



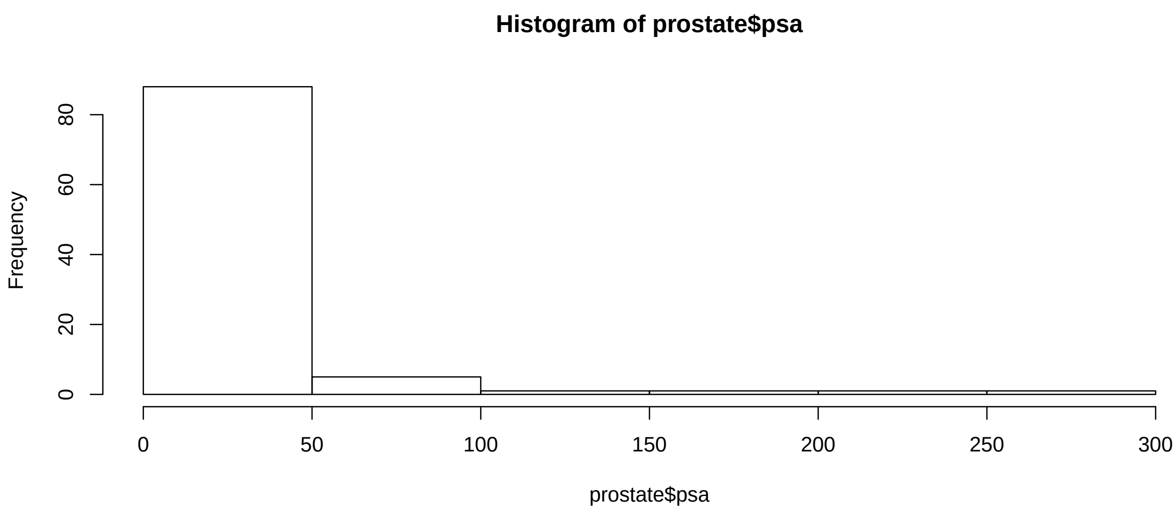

```r
hist(prostate$gleason)
```



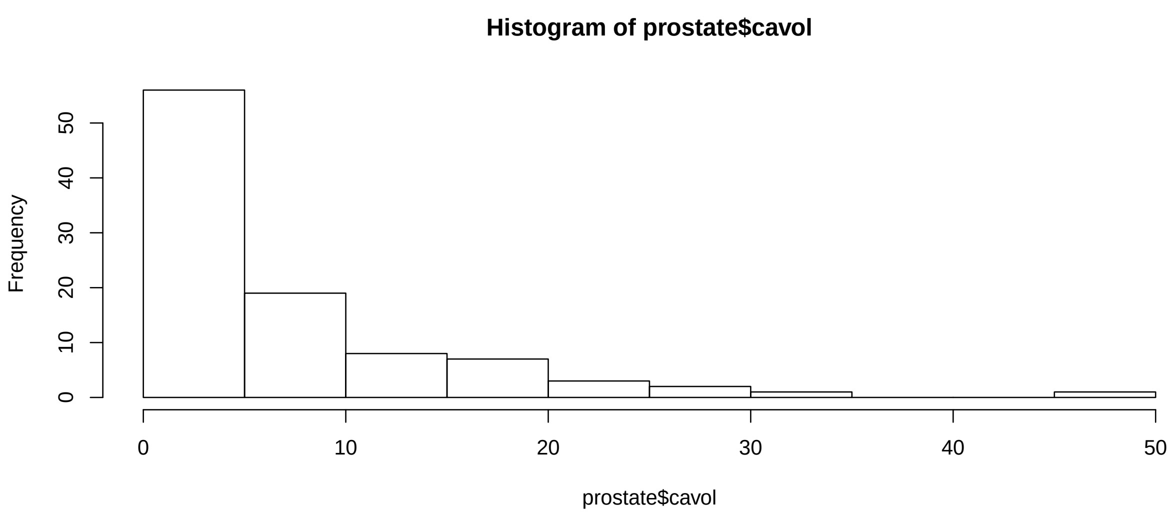

```
hist(prostate$pgg45)
```




```
hist(prostate$psa)
```




```r
hist(prostate$cavol)
```




```r
prostate$pgg45log <- log(prostate$pgg45)
prostate$pgg45log[is.infinite(prostate$pgg45log)] <- -exp(99)
prostate$cavollog <- log(prostate$cavol)
prostate$cavollog[is.infinite(prostate$cavollog)] <- -exp(99)

linreg_log <- lm(log(psa) ~ age+svi+log(gleason)+pgg45log+cavollog, data = prostate)
linreg <- lm(psa ~ age+svi+gleason+pgg45+cavol, data = prostate)
summary(linreg_log)
#>
#> Call:
#> lm(formula = log(psa) ~ age + svi + log(gleason) + pgg45log +
#>     cavollog, data = prostate)
#>
#> Residuals:
```

```
#>      Min       1Q   Median       3Q      Max
#> -1.59188 -0.44900  0.08758  0.49311  1.79603
#>
#> Coefficients:
#>                Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   4.067e+00  2.751e+00   1.478  0.14285
#> age          -1.736e-03  1.089e-02  -0.159  0.87370
#> svi           6.085e-01  2.247e-01   2.709  0.00807 **
#> log(gleason) -1.144e+00  1.382e+00  -0.828  0.40968
#> pgg45log      4.888e-44  3.067e-44   1.594  0.11441
#> cavollog      5.518e-01  8.446e-02   6.534 3.62e-09 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.7558 on 91 degrees of freedom
#> Multiple R-squared:  0.5936, Adjusted R-squared:  0.5713
#> F-statistic: 26.58 on 5 and 91 DF,  p-value: < 2.2e-16
summary(linreg)
#>
#> Call:
#> lm(formula = psa ~ age + svi + gleason + pgg45 + cavol, data = prostate)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -57.201  -7.284  -0.647   5.808 168.071
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 34.69936   48.90688   0.709   0.4798
#> age         -0.08742    0.45224  -0.193   0.8472
#> svi         26.04691   10.21595   2.550   0.0125 *
#> gleason     -4.32281    6.83508  -0.632   0.5287
#> pgg45        0.01522    0.18536   0.082   0.9348
#> cavol        2.54307    0.50782   5.008 2.68e-06 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 31.47 on 91 degrees of freedom
#> Multiple R-squared:  0.4367, Adjusted R-squared:  0.4057
#> F-statistic: 14.11 on 5 and 91 DF,  p-value: 3.3e-10
AIC(linreg_log, linreg)
#>            df      AIC
#> linreg_log  7 228.7684
#> linreg      7 952.2042
```

Das Modell mit logarithmierten Kovariaten gleason, pgg45 und cavol und abhängiger Variable log(psa) ist deutlich besser als das Modell ohne Variablentransformation. Dies lässt sich zum einen am R^2-Wert ablesen, als auch über den AIC-Vergleich feststellen.

b)

```
library("MASS")
library("leaps")
```

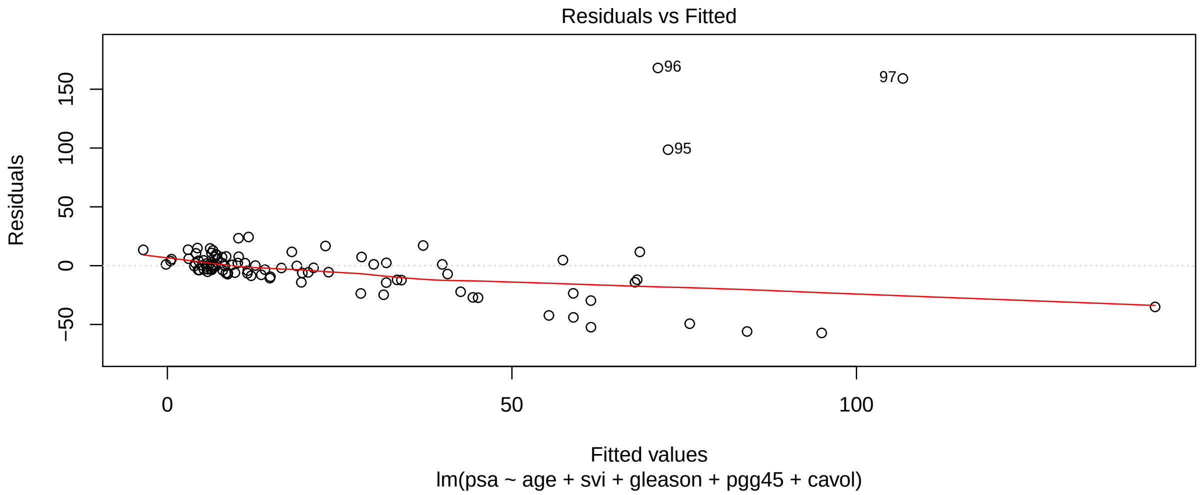
```
#> Error in library("leaps"): there is no package called 'leaps'

cat("Modellvergleich: StepAIC Backward & StepAIC Forward:")
#> Modellvergleich: StepAIC Backward & StepAIC Forward:
cat("\n")
stepAIC(linreg_log, direction = "backward")
#> Start:  AIC=-48.51
#> log(psa) ~ age + svi + log(gleason) + pgg45log + cavollog
#>
#>                Df Sum of Sq    RSS     AIC
#> - age           1    0.0145 51.999 -50.479
#> - log(gleason)  1    0.3919 52.376 -49.777
#> <none>                      51.984 -48.506
#> - pgg45log      1    1.4514 53.436 -47.834
#> - svi           1    4.1919 56.176 -42.983
#> - cavollog      1   24.3870 76.371 -13.193
#>
#> Step:  AIC=-50.48
#> log(psa) ~ svi + log(gleason) + pgg45log + cavollog
#>
#>                Df Sum of Sq    RSS     AIC
#> - log(gleason)  1    0.4061 52.405 -51.724
#> <none>                      51.999 -50.479
#> - pgg45log      1    1.4386 53.437 -49.831
#> - svi           1    4.2169 56.216 -44.915
#> - cavollog      1   24.5276 76.526 -14.996
#>
#> Step:  AIC=-51.72
#> log(psa) ~ svi + pgg45log + cavollog
#>
#>            Df Sum of Sq    RSS     AIC
#> <none>                  52.405 -51.724
#> - pgg45log  1    1.2724 53.677 -51.397
#> - svi       1    4.2570 56.662 -46.148
#> - cavollog  1   24.2135 76.618 -16.880
#>
#> Call:
#> lm(formula = log(psa) ~ svi + pgg45log + cavollog, data = prostate)
#>
#> Coefficients:
#> (Intercept)          svi     pgg45log     cavollog
#>   1.710e+00    6.127e-01    2.832e-44    5.456e-01
linreg_forward <- lm(log(psa) ~ 1, data = prostate)
stepAIC(linreg_forward, direction = "forward", scope=list(upper=linreg_log,lower=linreg_forward))
#> Start:  AIC=28.84
#> log(psa) ~ 1
#>
#>                Df Sum of Sq     RSS     AIC
#> + cavollog      1    69.003  58.915 -44.366
#> + svi           1    41.011  86.907  -6.658
#> + pgg45log      1    29.987  97.931   4.926
#> + log(gleason)  1    19.767 108.151  14.555
#> + age           1     3.679 124.239  28.007
```

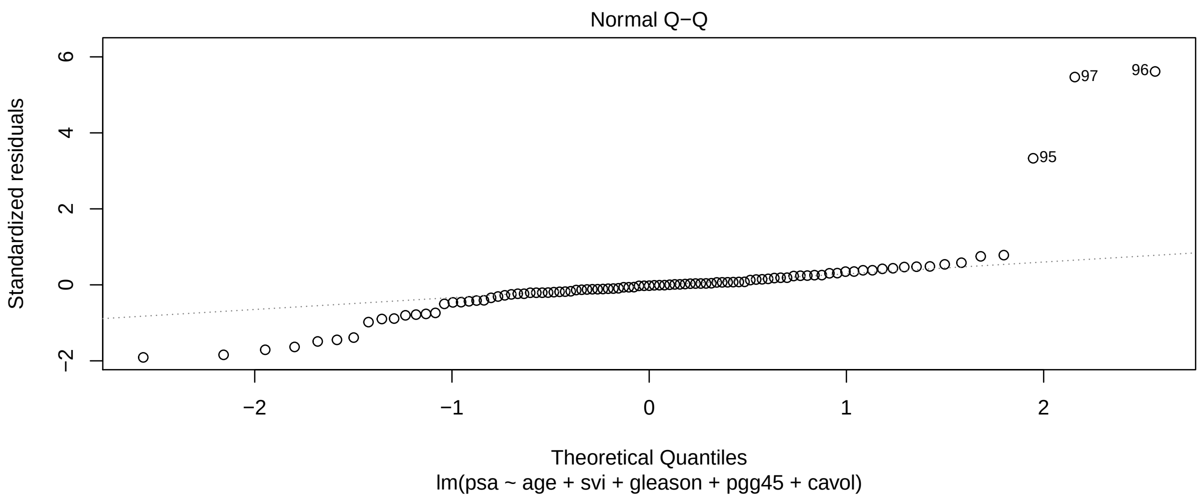
```
#> <none>                      127.918  28.838
#>
#> Step:  AIC=-44.37
#> log(psa) ~ cavollog
#>
#>                Df Sum of Sq    RSS     AIC
#> + svi           1    5.2375 53.677 -51.397
#> + pgg45log      1    2.2528 56.662 -46.148
#> <none>                      58.915 -44.366
#> + log(gleason)  1    0.5963 58.318 -43.353
#> + age           1    0.0025 58.912 -42.370
#>
#> Step:  AIC=-51.4
#> log(psa) ~ cavollog + svi
#>
#>                Df Sum of Sq    RSS     AIC
#> + pgg45log      1   1.27236 52.405 -51.724
#> <none>                      53.677 -51.397
#> + log(gleason)  1   0.23993 53.437 -49.831
#> + age           1   0.00364 53.674 -49.404
#>
#> Step:  AIC=-51.72
#> log(psa) ~ cavollog + svi + pgg45log
#>
#>                Df Sum of Sq    RSS     AIC
#> <none>                      52.405 -51.724
#> + log(gleason)  1   0.40613 51.999 -50.479
#> + age           1   0.02874 52.376 -49.777
#>
#> Call:
#> lm(formula = log(psa) ~ cavollog + svi + pgg45log, data = prostate)
#>
#> Coefficients:
#> (Intercept)     cavollog          svi     pgg45log
#>   1.710e+00    5.456e-01    6.127e-01    2.832e-44

regfit.full <- regsubsets(log(psa) ~ age+svi+log(gleason)+pgg45log+cavollog, data = prostate)
#> Error in regsubsets(log(psa) ~ age + svi + log(gleason) + pgg45log + cavollog, : konnte Funktion "re
reg.summary <- summary(regfit.full)
#> Error in summary(regfit.full): Objekt 'regfit.full' nicht gefunden

plot(reg.summary$rsq, type='l', xlab="Number of Variables", ylab="R-Squared")
#> Error in plot(reg.summary$rsq, type = "l", xlab = "Number of Variables", : Objekt 'reg.summary' nich
plot(reg.summary$cp, type='l', xlab="Number of Variables", ylab="AIC")
#> Error in plot(reg.summary$cp, type = "l", xlab = "Number of Variables", : Objekt 'reg.summary' nicht
plot(reg.summary$bic, type='l', xlab="Number of Variables", ylab="BIC")
#> Error in plot(reg.summary$bic, type = "l", xlab = "Number of Variables", : Objekt 'reg.summary' nich

cat("Bestes Modell laut regsubsets:")
#> Bestes Modell laut regsubsets:
cat("\n")
coef(regfit.full, 3)
#> Error in coef(regfit.full, 3): Objekt 'regfit.full' nicht gefunden
```

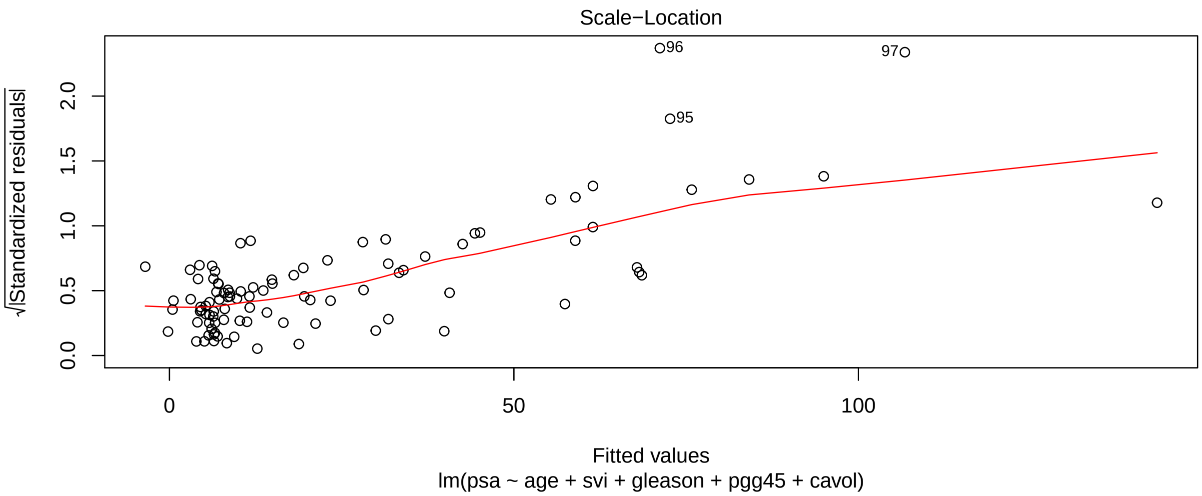
Laut StepAIC ist das Modell "log(psa) ~ svi + pgg45log + cavollog" das Beste (Sowohl backward als auch forward). variablenselektion durch Regsubsets ist nicht eindeutig. Laut BIC ist das Modell mit 2 Kovariaten das Beste, AIC und R^2 sprechen für das Modell mit 3 Kovariaten. Dieses ist das Gleiche, wie im StepAIC. Wir entscheiden uns deshalb für dieses Modell.

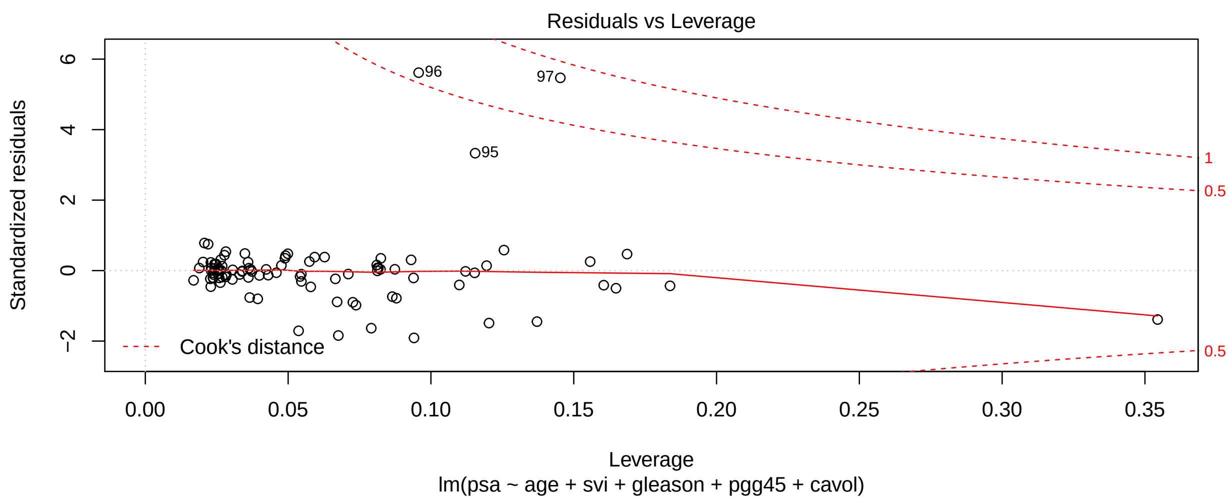
c)

```
linreg_final <- lm(log(psa) ~ svi + pgg45log + cavollog, data=prostate)
plot(linreg)
```

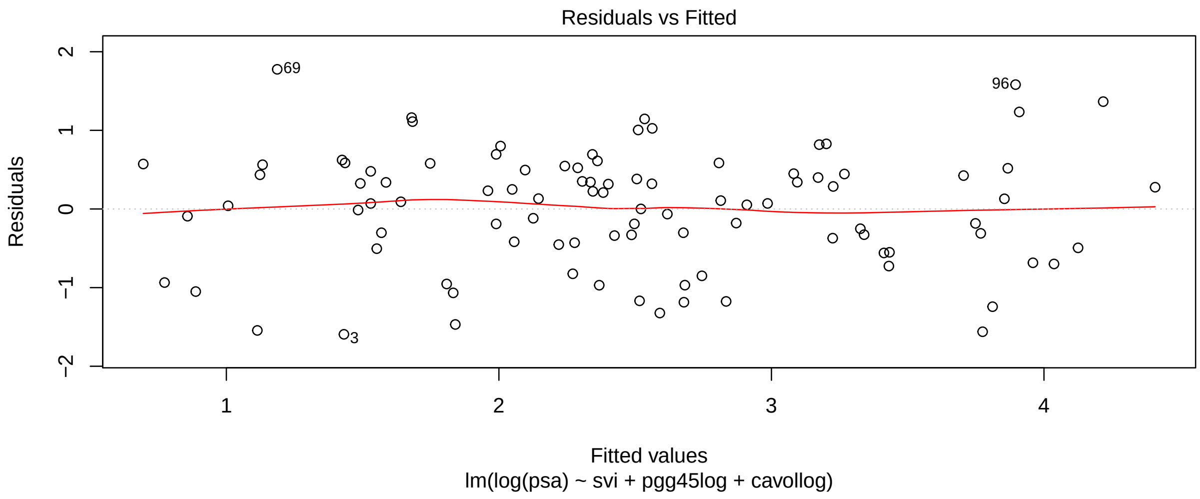


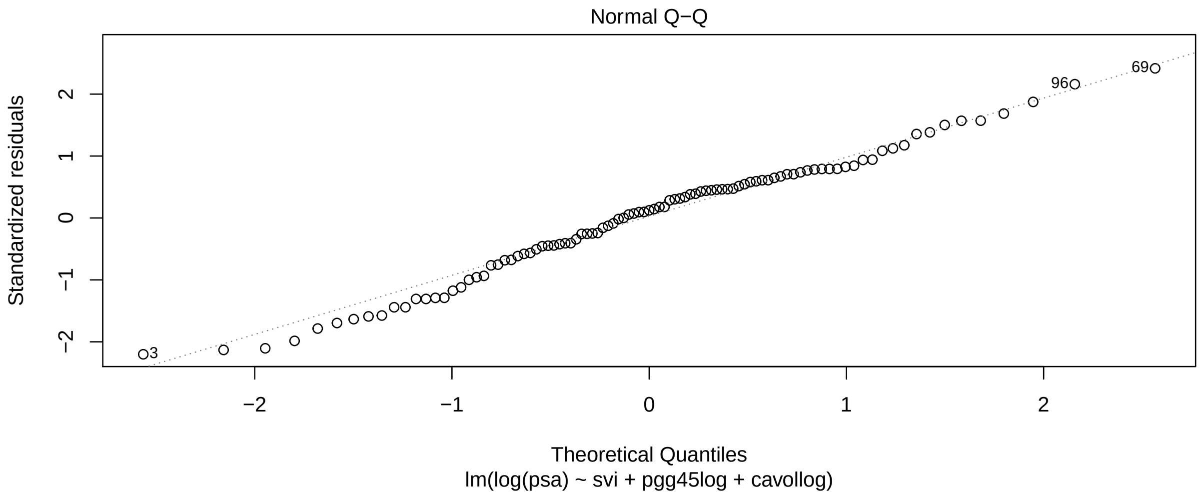


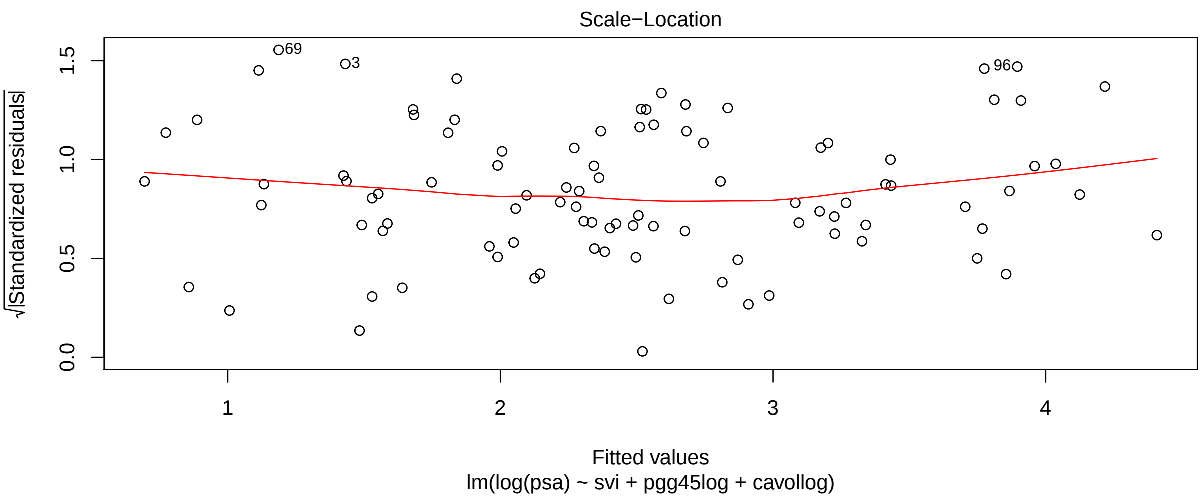


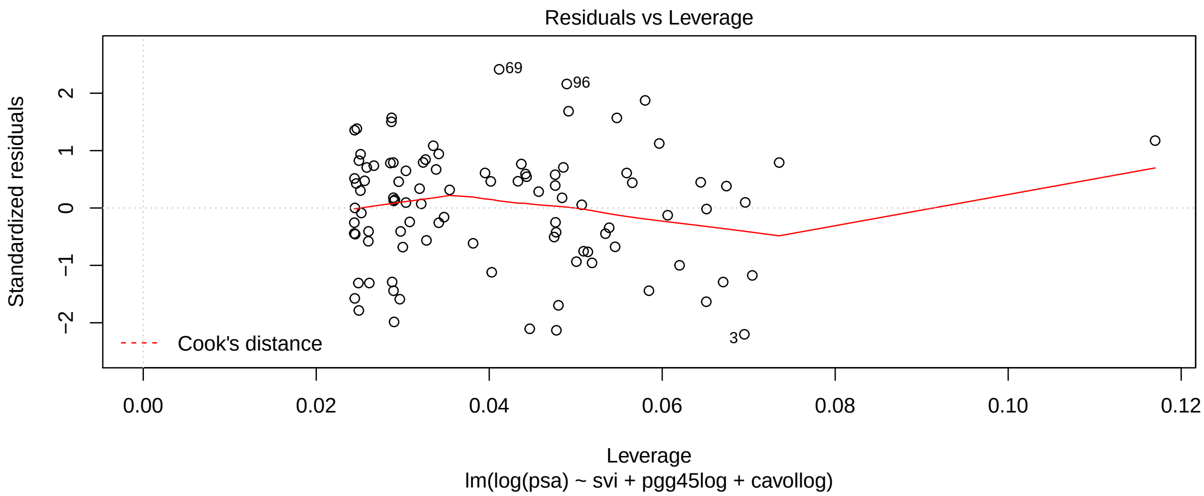



```
plot(linreg_final)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(psa) ~ svi + pgg45log + cavollog)
69
96
3


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(psa) ~ svi + pgg45log + cavollog)
69
96
3


Scale−Location
√|Standardized residuals|
Fitted values
lm(log(psa) ~ svi + pgg45log + cavollog)
69
3
96


Residuals vs Leverage
Standardized residuals
Leverage
lm(log(psa) ~ svi + pgg45log + cavollog)
Cook's distance
69
96
3
−2
−1
0
1
2
0.00
0.02
0.04
0.06
0.08
0.10
0.12

Aufgabe 2

Wir können die Unkorreliertheit einerseits beweisen durch

$$\begin{aligned}
\operatorname{Cov}(Y_i, Y_j) &= \operatorname{Cov}\left((A^T X)_i, (A^T X)_j\right) && (1)\\
&= \operatorname{Cov}\left(A^T X\right)_{ij} && (2)\\
&= \left(A^T \operatorname{Cov}(X)\, A\right)_{ij} && (3)\\
&= \left(A^T \Sigma A\right)_{ij} && (4)\\
&= \Lambda_{ij} && (5)\\
&= \begin{cases} 1, & i = j, \\ 0, & \text{sonst.} \end{cases} && (6)
\end{aligned}$$

Alternativ zur besseren Vorstellung. Sei e_i der i-te Einheitsvektor, dann erhalten wir:

$$\begin{aligned}
\operatorname{Cov}(Y_i, Y_j) &= \operatorname{Cov}\left((A^T X)_i, (A^T X)_j\right) && (7)\\
&= \operatorname{Cov}\left(e_i^T A^T X, e_j^T A^T X\right) && (8)\\
&= e_i^T \operatorname{Cov}\left(A^T X, A^T X\right) e_j && (9)\\
&= e_i^T A^T \operatorname{Cov}(X, X)\, A e_j && (10)\\
&= e_i^T A^T \Sigma A e_j && (11)\\
&= e_i^T \Lambda e_j && (12)\\
&= \Lambda_i e_j && (13)\\
&= \Lambda_{ij} \qquad = \begin{cases} 1, & i = j, \\ 0, & \text{sonst.} \end{cases} && (14)
\end{aligned}$$

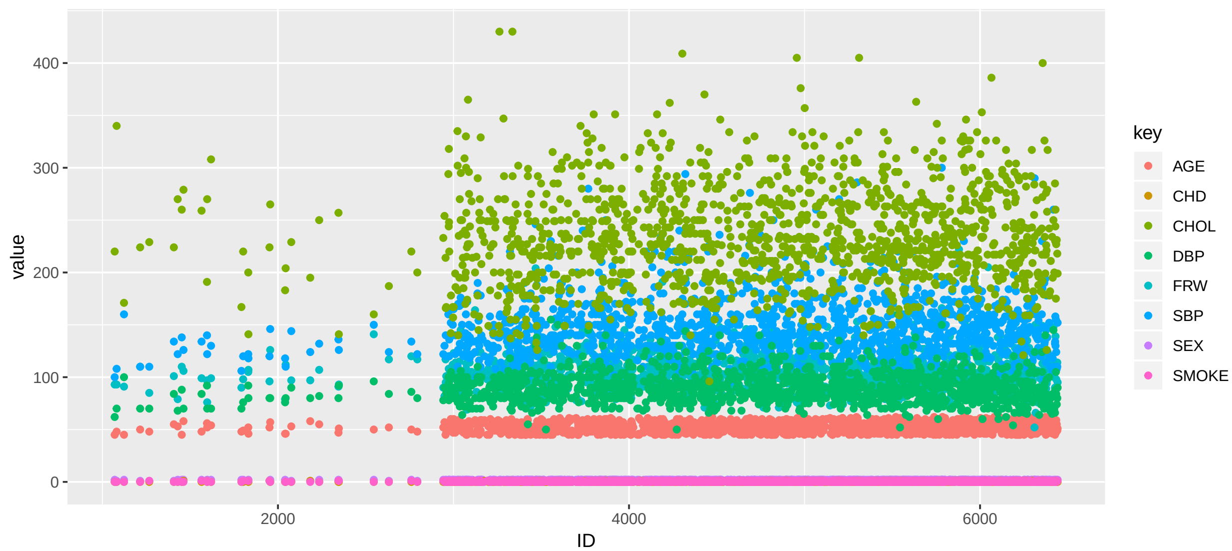
Aufgabe 3

```
data = read.csv('framingham.csv', sep=';')
df = data.frame(data)
```

a)

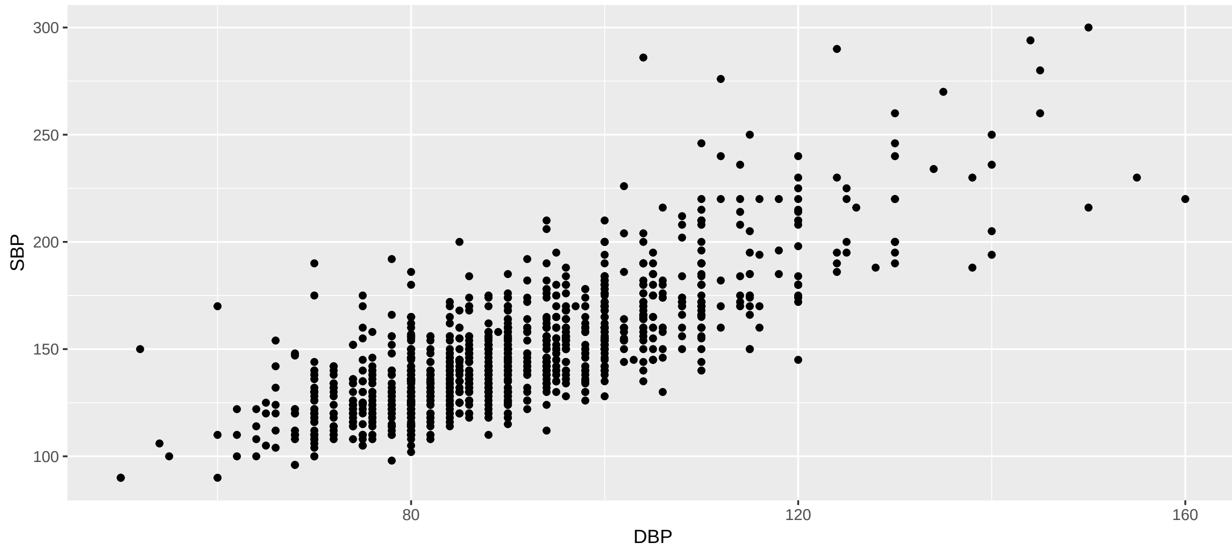
Einmal der gesamte Datensatz.

```
library(ggplot2)
library(tidyr)
ggdf = gather(df, key='key', value='value', c(3,4,5,6,7,8,9,10))
gg = ggplot(
  data = ggdf,
  mapping = aes(
    x = ID,
    y = value,
    color = key
  )
)
gg = gg + geom_point()
gg
```



Nun beschränkt auf DBP als X_1 und SBP als X_2.

```
df = data.frame(DBP = df$DBP, SBP = df$SBP)
ggdf = gather(df, key='key', value='value')
gg = ggplot(
  data = df,
  mapping = aes(
    x = DBP,
    y = SBP
  )
)
gg = gg + geom_point()
gg
```




```
x1 = df$DBP
x2 = df$SBP
sol = round(cor(x1, x2), 3)
sol
#> [1] 0.789
```

Wir erhalten eine Korrelation von 0.789 und somit sind DBP und SBP positiv korreliert.