Predictive Modeling

Series 4

Exercise 4.1

The data file **fitness.rda** contains measurements of a fitness test of 31 patients. The response variable **oxy** refers to the rate of oxygen consumption which was measured by means of a complicated and expensive procedure. Predictors are **age**, **weight**, **runtime** (running time), **rstpulse** (resting pulse), **runpulse** (running/active pulse) and **maxpulse** (maximal pulse).

a) Load the data file **fitness.rda** and fit a regression model that contains all predictors. Is there any association between predictors and response variable?

R-Hint:

```r
## load data
load(".../fitness.rda")
```

b) Verify whether variable transformations are necessary by carrying out a residual analysis. **R-Hint**:

```r
source("resplot.R")
resplot(fit)
```

If necessary, adapt the model so that it does not show any systematic errors or any other violations of the model assumptions.

Optional, for the advanced reader: By using *partial residual plots* you can check whether all predictors have been included in the correct form. How do you interpret partial residual plots? **R-Hint**:

```r
## Partial residual plots
library(car)
help(crPlots)
crPlots(fit, pch = 20, layout = c(2, 3), cex.lab = 0.75)
```

c) Analyze the data with respect to pairwise correlations of the predictors.

R-Hint:

```r
library(ellipse)
plotcorr(cor(fitness[, -3]), cex.lab = 0.75, mar = c(1,
    1, 1, 1))
```

d) Check whether there is high multicollinearity by computing the VIFs.

R-Hint:

```r
library(faraway)
vif(fit)
```

e) Alleviate the multicollinearity problem by using different methods:

i) Amputation, i.e. leave out redundant variables.

ii) Create new variables that are not collinear, e.g. by considering the quotient of two correlated variables.

Save the fitted values for each of these adapted models and carry out a pairwise comparison by generating pairwise scatterplots. **R-Hints**:

```r
pairs(fit.i, fit.ii)
```

What do you observe?

f) Use the model from part e) (ii) and perform a hybrid stepwise model selection using the AIC as criterion in order to reduce the set of predictors. Consider an ANOVA-test between the model from part e) (ii) and the reduced model to decide whether the omitted variables are indeed redundant. **R-Hints**:

```r
library(MASS)
step(fit.empty, direction = "both", scope = list(lower = fit.empty,
    upper = fit.full), trace = 0)
anova(fit.reduced, fit.full)
```

g) The goal of this study is to substitute the expensive and complicated procedure required to measure the rate of oxygen consumption by a set of predictors that are inexpensive to measure. Is this possible, and if so, how would you do it?

Exercise 4.2

In a study about controlling infection risk in US hospitals, a random sample of 113 hospitals were considered along with 12 measured quantities. After loading the data file `senic.rda`, check whether `region` and `school` are recognized as a factor variables. If not, transform them into factor variables.

R-Hints:

```
str(senic)
as.factor(...)
```

By using the following variables as predictors:

- `age` : average age of patients in years
- `inf` : average infection risk in percent
- `region` : geographical region with 1 = NE , 2 = N , 3 = S and 4 = W
- `beds` : number of beds
- `pat` : average number of patients a day
- `nurs` : number of full-employed and trained nurses

and

- `length` : the average duration of hospital stay in days

as response variable, carry out a linear regression analysis and select an optimal model by following these instructions:

a) Check the correlations between these variables. Which of them are problematic and why? Is there an intuitive explanation of this problem?

b) Combine the predictors into new variables to improve the situation. In particular, substitute `beds` by `pat/beds` and `nurs` by `pat/nurs`.

c) (**Optional, for the advanced reader**)

   By using *partial residual plots* you can check whether all predictors have been included in the correct form. How do you interpret partial residual plots? **R-Hint**:

   ```
   ## Partial residual plots
   library(car)
   help(crPlots)
   crPlots(fit, pch = 20, layout = c(2, 3), cex.lab = 0.75)
   ```

   By analysing the residuals by means of the partial residual plots, explain why the variables `length`, `pat` and `pat.nurs` need to be transformed. Use a log-transformation of these variables.

d) Fit a linear regression model using the log-transformed variables `length`, `pat` and `pat.nurs`.

e) Perform a backward stepwise selection using the AIC criterion. Use the R-function step().

   Check the final model with the usual diagnostic plots.

f) Now perform a forward stepwise selection using the AIC criterion. Thus, start with the empty model. Use the R-function step(). Check also the diagnostics plots and comment on the differences with respect to c) and d).

g) Perform a hybrid stepwise selection and compare the results you have obtained either through the backward and forward stepwise selection.

h) Carry out an ANOVA-test with the models you have obtained through stepwise selection.

Exercise 4.3

So-called *Funds of Hedge Funds* (FoHF), i.e. portfolios of hedge funds, have different investment strategies with specific returns and risk properties. When such a product is evaluated it is important for the investor to choose the investment style that fits his needs. One approach to assess the investment strategy of a FoHF as an outsider is to perform a style analysis based on the returns. Using a regression model (also called multi-factor model in the financial industry) one aims to predict the returns of the FoHF on the basis of the returns of the so-called subindices of hedge funds (Long Short Equity, Fixed Income Arbitrage, Global Macro, etc.). The estimated parameters are indications for the chosen investment strategy.

Note that not all investment strategies are present due to the construction of FoHFs. The file FoHF.rda contains the monthly returns of one FoHF and the hedge fund subindices of EDHEC from January 1997 until December 2004. The individual predictors are refered to as the following measures:

- RV: Relative value
- CA: Convertible Arbitrage
- FIA: Fixed Income Arbitrage
- EMN: Equity Market Neutral
- ED: Event Driven Multistrategy
- DS: Distressed Securities
- MA: Merger Arbitrage
- LSE: Long Short Equity

- **GM**: Global Macro
- **EM**: Emerging Markets
- **CTA**: CTA / Managed Futures
- **SS**: Short Selling

Fit the following model:

$$\text{FoHF} \sim RV + CA + FIA + EMN + ED + DS + MA + LSE + GM + EM + CTA + SS$$

a) Look at the output of **summary()**. What do you conclude with respect to the investment strategy of this **FoHF** when you consider the estimated coefficients, the p-values, the global F-test and the multiple R^2?

b) Check whether this model is valid or whether any assumptions are violated. Also test whether there are problems with respect to multicollinearity and whether all predictors have been included into the model in the correct form.

c) If you have solved the previous subproblem correctly, you will have found some issues. Formulate a strategy how those can be fixed in order to obtain a valid and interpretable result. **Hint**: Creating new predictors is not helpful.

d) Perform variable selection using the BIC criterion. Implement the following search strategies, identify the best/final model and compare:

   i) Hybrid stepwise variable selection, starting with the full model.

   ```
   step(fit.full, method = "both", k = log(nrow(FoHF)))
   ```

   ii) Hybrid stepwise variable selection, starting with the empty model.

   ```
   scp <- list(lower = formula(fit.null), upper = formula(fit.full))
   step(fit.null, scope = scp, k = log(nrow(FoHF)))
   ```

   iii) All Subsets variable selection.

   ```
   step(fit.full, method = "both", k = log(nrow(FoHF)))
   ```

e) Does the ANOVA-test justifies that variables have been omitted as a consequence of the stepwise model selections?

f) **Optional**: For this dataset the Lasso is well suited. Fit the model and generate the Lasso traces which allow to identify important predictors. Choose an appropriate value for λ and retrieve the final model as well as its coefficients. See Chapter 6.6 of *Introduction to Statistical Learning* by Gareth et all.

```r
## Lasso
library(glmnet)
xx <- model.matrix(FoHF ~ ., data = FoHF)
yy <- FoHF$FoHF
cvfit <- cv.glmnet(xx, yy)
plot(cvfit)
coef(cvfit, s = "lambda.1se")
```

Result Checker

Predictive Modeling

Solutions to Series 4

Solution 4.1

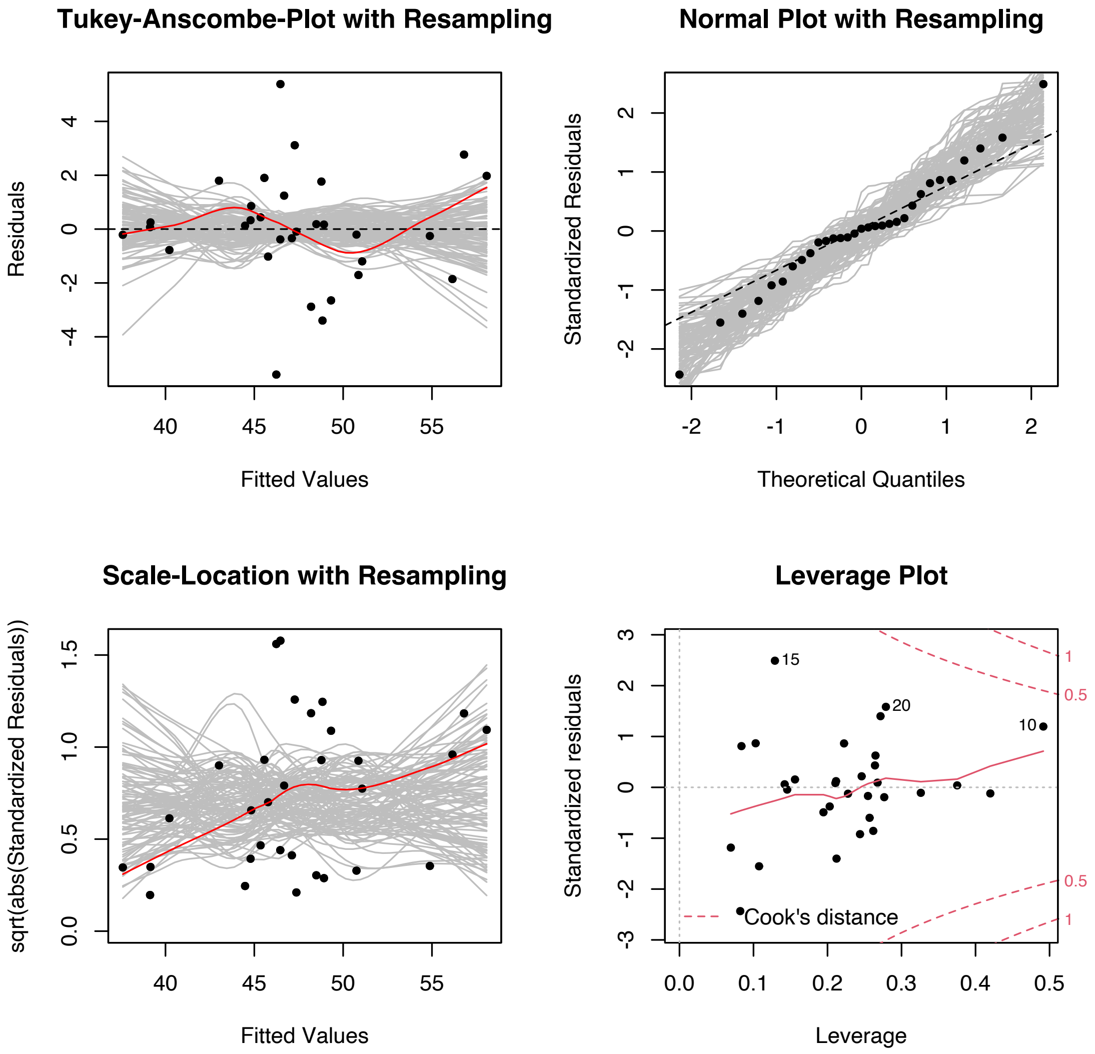
a)

```
## load data
load("Daten/fitness.rda")
## fit model
fit <- lm(oxy ~ ., data = fitness)
summary(fit)

##
## Call:
## lm(formula = oxy ~ ., data = fitness)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4026 -0.8991  0.0706  1.0496  5.3847
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
## age          -0.22697    0.09984  -2.273  0.03224 *
## weight       -0.07418    0.05459  -1.359  0.18687
## runtime      -2.62865    0.38456  -6.835 4.54e-07 ***
## rstpulse     -0.02153    0.06605  -0.326  0.74725
## runpulse     -0.36963    0.11985  -3.084  0.00508 **
## maxpulse      0.30322    0.13650   2.221  0.03601 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.317 on 24 degrees of freedom
## Multiple R-squared:  0.8487,Adjusted R-squared:  0.8108
## F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

Due to the p-value of the associated F-test (`p-value`: $9.715 \cdot 10^{-9}$), we conclude that at least one predictor is associated with the response variable.

b)

```
fit <- lm(oxy ~ ., data = fitness)
source("resplot.R")
resplot(fit)
```



At first sight, we spot two observations which show relatively large residuals, one of which is positive, the other one is negative. The assumption of constant variance seems to be acceptable at the outermost limit.

Residual Plots (optional, for advanced readers) In many applied problems, it is very interesting to understand and visualize the relation between the response Y and some arbitrary predictor X_k. However, a plot of Y versus X_k probably is deceiving, because in a multiple regression setting, all other predictors $X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_p$ will simultaneously have an effect on the response.

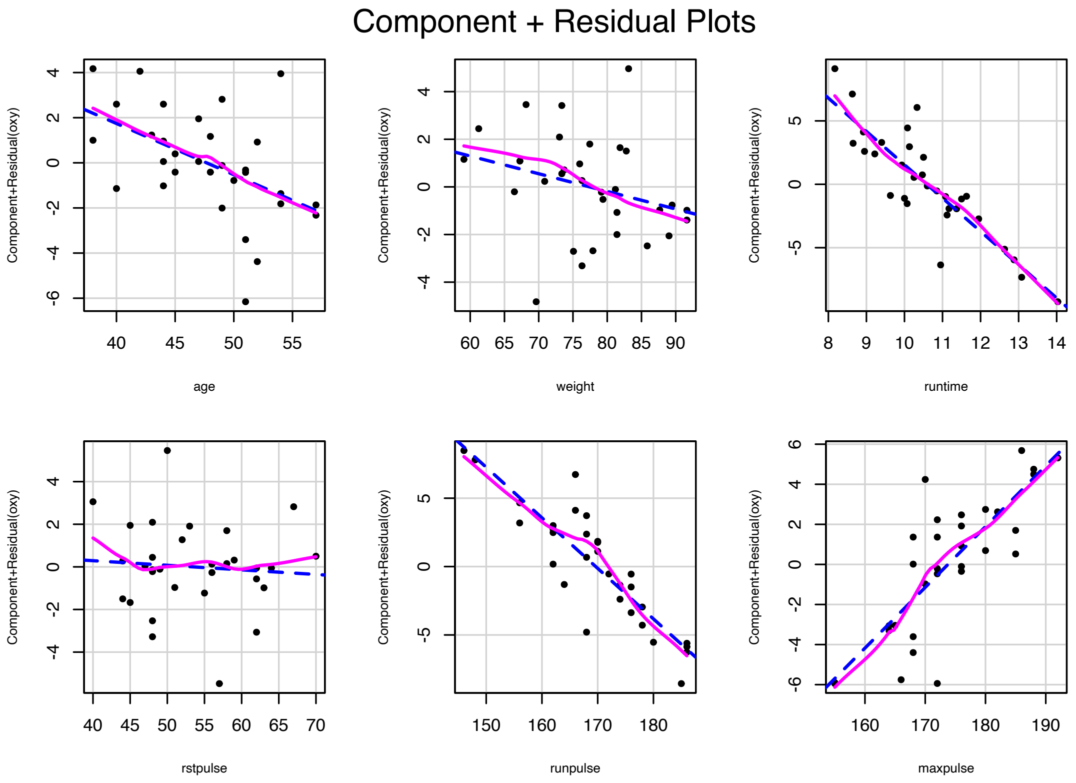
Hence, what we should aim for is displaying the relation of Y versus X_k in the presence of the other predictors. That is what the partial residual plot does. We thus generate an updated Y variable, where the effect of all other predictors is removed from the response.

Mathematically, the partial residuals for predictor X_k are:

$$\begin{aligned} Y - \sum_{j \neq k} \hat{\beta}_k X_k &= \hat{Y} + R - \sum_{j \neq k} \hat{\beta}_k X_k \\ &= \hat{\beta}_k X_k + R \end{aligned}$$

where R denotes the residuals determined for the multiple linear regression model. The residual plots are generated by plotting the partial residuals, that is for every observation i, we plot $\hat{\beta}_k X_i^{(k)} + R_i$ versus the predictor $X_i^{(k)}$.

```
## partial residual plots
library(car)

## Loading required package: carData

crPlots(fit, pch = 20, layout = c(2, 3), cex.lab = 0.75)
```



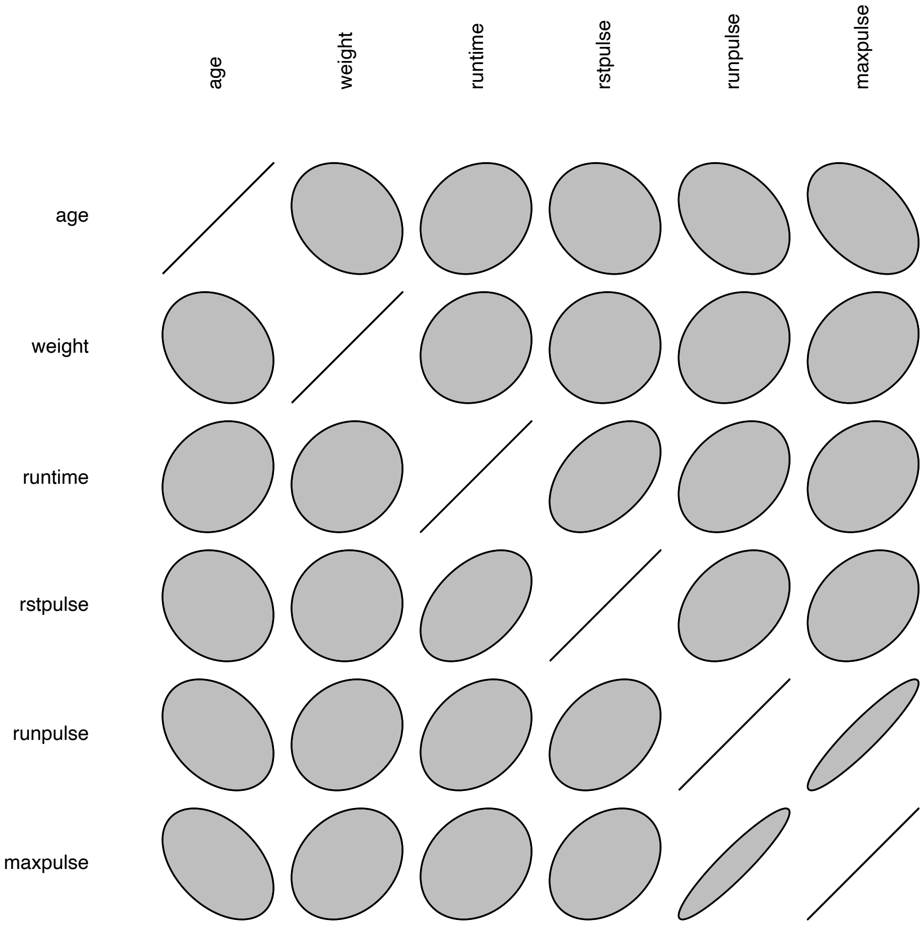
The partial residual plots are enhanced with the red dashed line that illustrates the actual fit according to the multiple linear regression model, that is $\hat{\beta}_k X_i^{(k)}$ versus $X_i^{(k)}$. The green solid line is a smoother that was added for visualizing the (true) relation between partial residual and predictor. If there appears a significant difference between their actual, linear fit and the true relation indicated by the smoother, one should improve the model. Sometimes, we can transform predictors to achieve this; at other times adding additional predictors and/or interaction terms may help.

Although the residual plots do not look perfect, the model assumptions seem to be fulfilled to some limited degree of satisfaction. On the basis of the residual plots we thus conclude that there are no systematic errors.

The two observations associated with the large residuals are responsible for deviations in the partial residual plots. In these cases, however, we would not diagnose the presence of a systematic deviation. Therefore, we conclude that the predictors have been included in the model in the correct form.

c)

```
## load data
load("Daten/fitness.rda")
par(mfrow = c(1, 1))
library(ellipse)

##
## Attaching package: 'ellipse'

## The following object is masked from 'package:car':
##
##     ellipse

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

plotcorr(cor(fitness[, -3]), cex.lab = 0.75, mar = c(1,
    1, 1, 1))
```



The analysis of the pairwise correlations should be done without the response variable. As we can see from the above plot, there is a strong positive correlation between the `running pulse` and the `maximal pulse`. The remaining variables do not show strong pairwise correlations.

d)
```
## multicollinearity
library(faraway)

## Registered S3 methods overwritten by 'lme4':
##   method                            from
##   cooks.distance.influence.merMod car
##   influence.merMod                  car
##   dfbeta.influence.merMod          car
##   dfbetas.influence.merMod          car
```

```
##
## Attaching package: 'faraway'
## The following objects are masked from 'package:car':
##
##     logit, vif
```

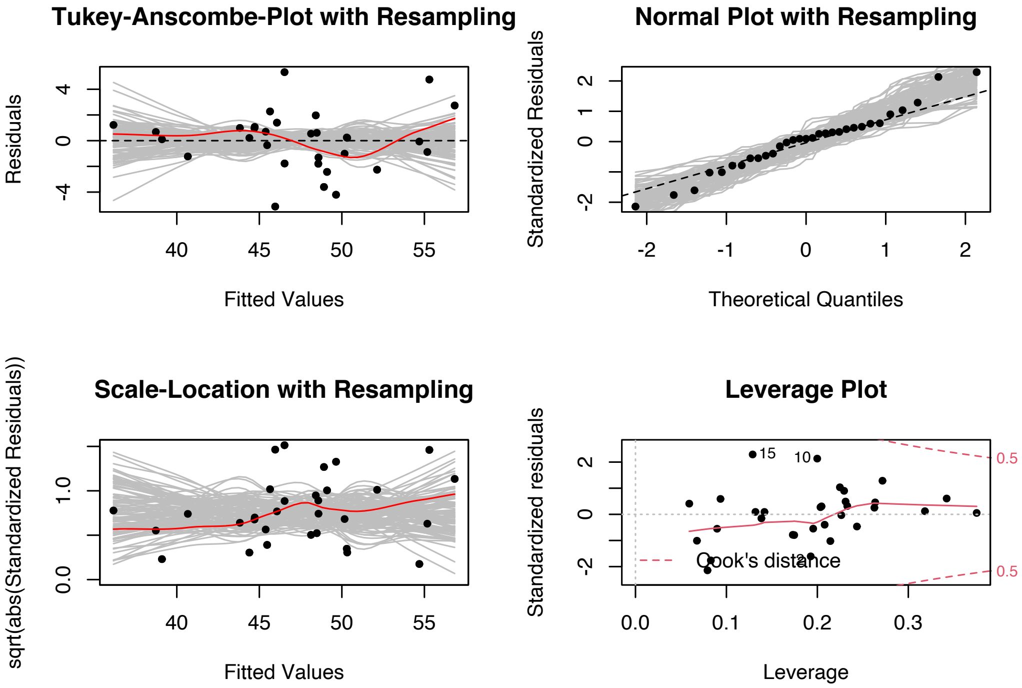
```
vif(fit)
```

```
##      age   weight  runtime rstpulse runpulse maxpulse
## 1.512836 1.155329 1.590868 1.415589 8.437274 8.743848
```

The VIFs of `runpulse` and `maxpulse` indicate the presence of critical multicollinearity. This is not surprising given the large pairwise correlation between `running pulse` and `maximal pulse`.

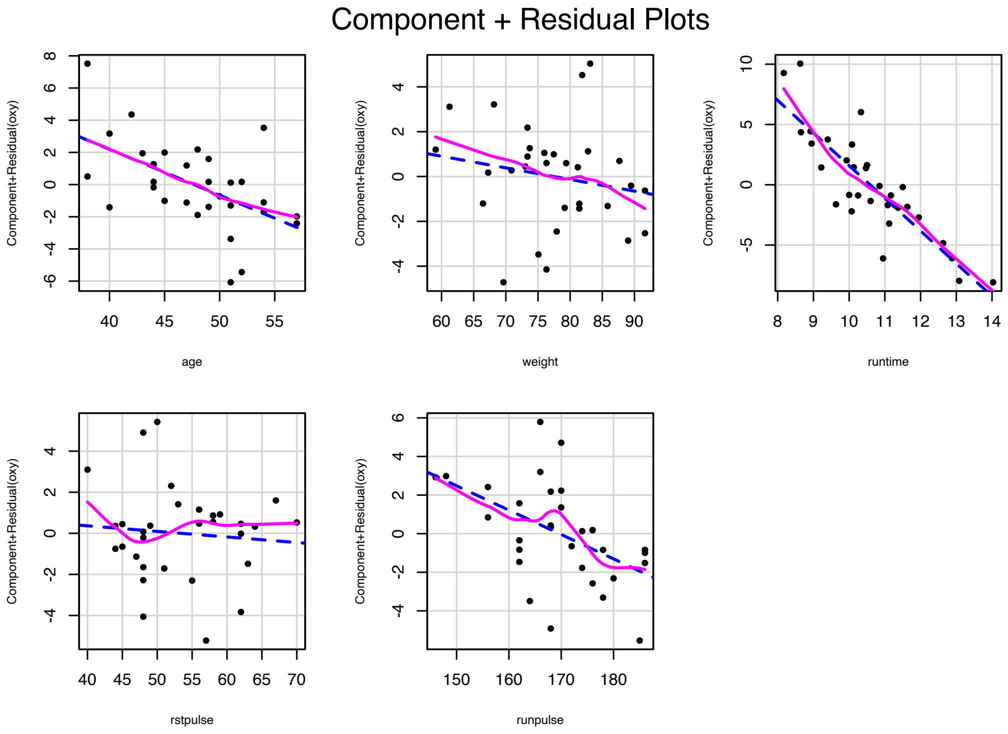
e) i) Amputation

```
## fitted values
f.o <- fitted(fit)
## Amputation - leave out maxpulse
fit <- lm(oxy ~ age + weight + runtime + rstpulse + runpulse,
    data = fitness)
resplot(fit)
```



Since the high multicollinearity stems from the large pairwise correlation between **runpulse** and **maxpulse**, one of these two variables should be excluded from the model. We recommend to leave out **maxpulse** due to background knowledge.

```
crPlots(fit, pch = 20, layout = c(2, 3), cex.lab = 0.75)
```



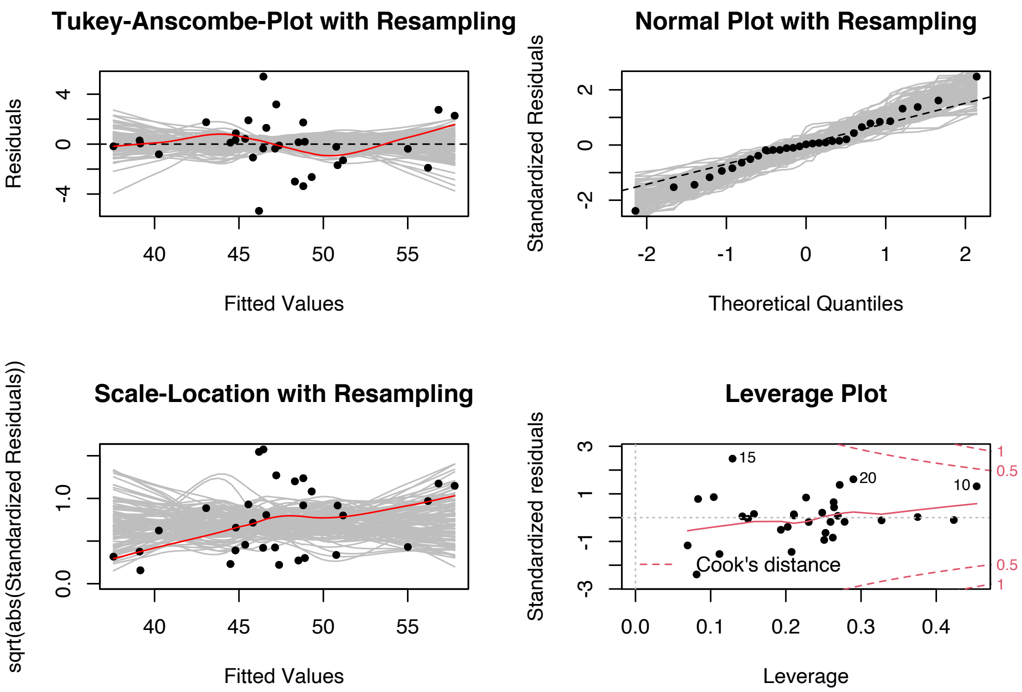
```
vif(fit)

##       age   weight  runtime rstpulse runpulse
## 1.408289 1.116150 1.578518 1.413545 1.388799

f.i <- fitted(fit)
```

The resulting model does not show any systematic error, the predictors seem to enter the model in the correct form and there is no high multicollinearity.

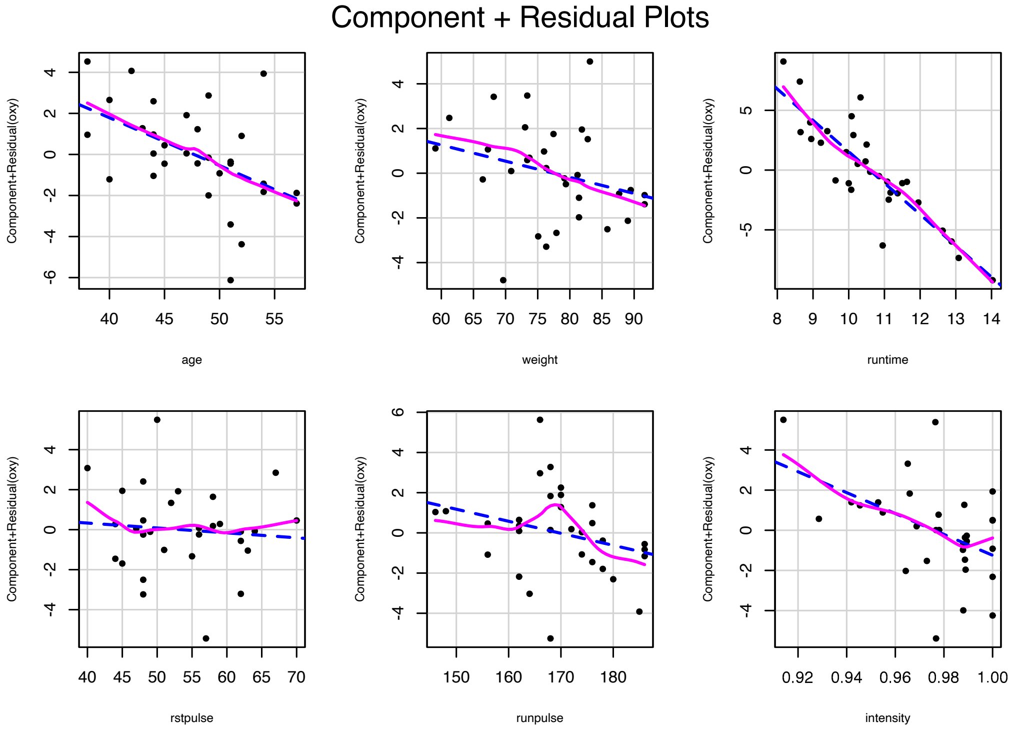
ii) Either **runpulse** or **maxpulse** need to be adjusted. We keep **runpulse** in the model and substitute **maxpulse** by creating a new variable, which is either **maxpulse - runpulse** or **runpulse/maxpulse**. We will choose as new variable the quotient **runpulse/maxpulse**.

```
my.fitness <- fitness[, -7]
my.fitness$intensity <- fitness$runpulse/fitness$maxpulse
```

```
fit <- lm(oxy ~ ., data = my.fitness)
resplot(fit)
```



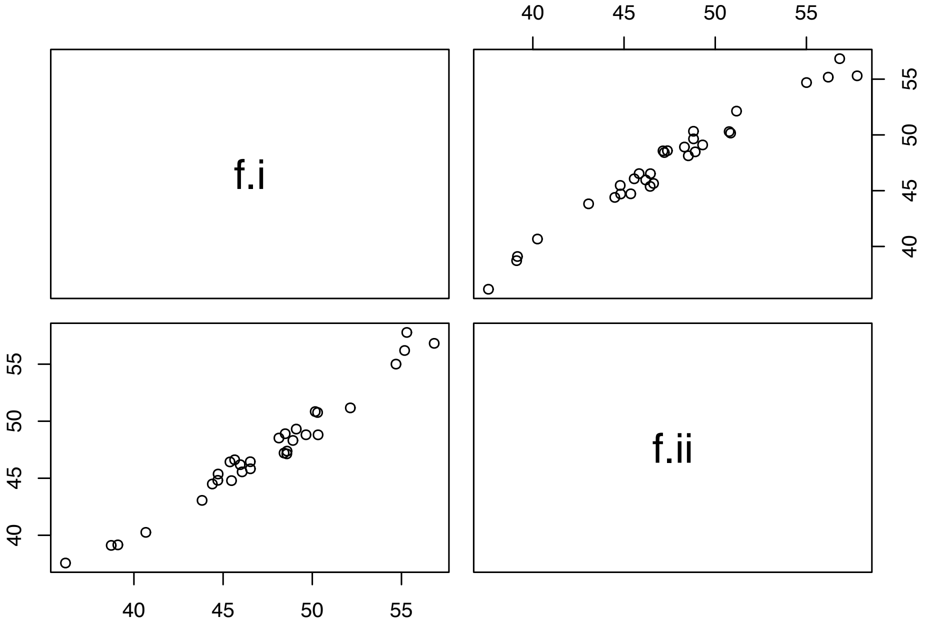
```
crPlots(fit, pch = 20, layout = c(2, 3), cex.lab = 0.75)
```



```
vif(fit)

##       age    weight   runtime  rstpulse  runpulse
##  1.500884  1.152036  1.594347  1.414005  1.961997
## intensity
##  1.615894

f.ii <- fitted(fit)
```

Pairwise comparison of fitted values

```
## comparison of the fitted values
df <- data.frame(f.i, f.ii)
pairs(df)
```



When using amputation the fitted values are notably different from the approach using a new variable. This indicates that we do lose some precision when excluding a variable from the set of predictors.

f)
```
## Hybrid stepwise selection
library(MASS)
fit.full <- lm(oxy ~ ., data = my.fitness)
fit.empty <- lm(oxy ~ NULL, data = my.fitness)
step(fit.empty, direction = "both", scope = list(lower = fit.empty,
    upper = fit.full), trace = 0)

##
## Call:
## lm(formula = oxy ~ runtime + intensity + age + weight, data = my.f
##
## Coefficients:
## (Intercept)      runtime    intensity          age
##   156.37864     -2.78418    -66.75605     -0.17542
##      weight
##    -0.07759
```

The variables **rstpulse** and **runpulse** are excluded from the model. Finally, only **runtime**, **age** and **intensity** (the quotient of **runpulse** and **maxpulse**) are considered in the model.

```
fitness.reduced <- my.fitness[, c("oxy", "age", "intensity",
    "runtime", "weight")]
fit.reduced <- lm(oxy ~ ., data = fitness.reduced)
summary(fit.reduced)
```

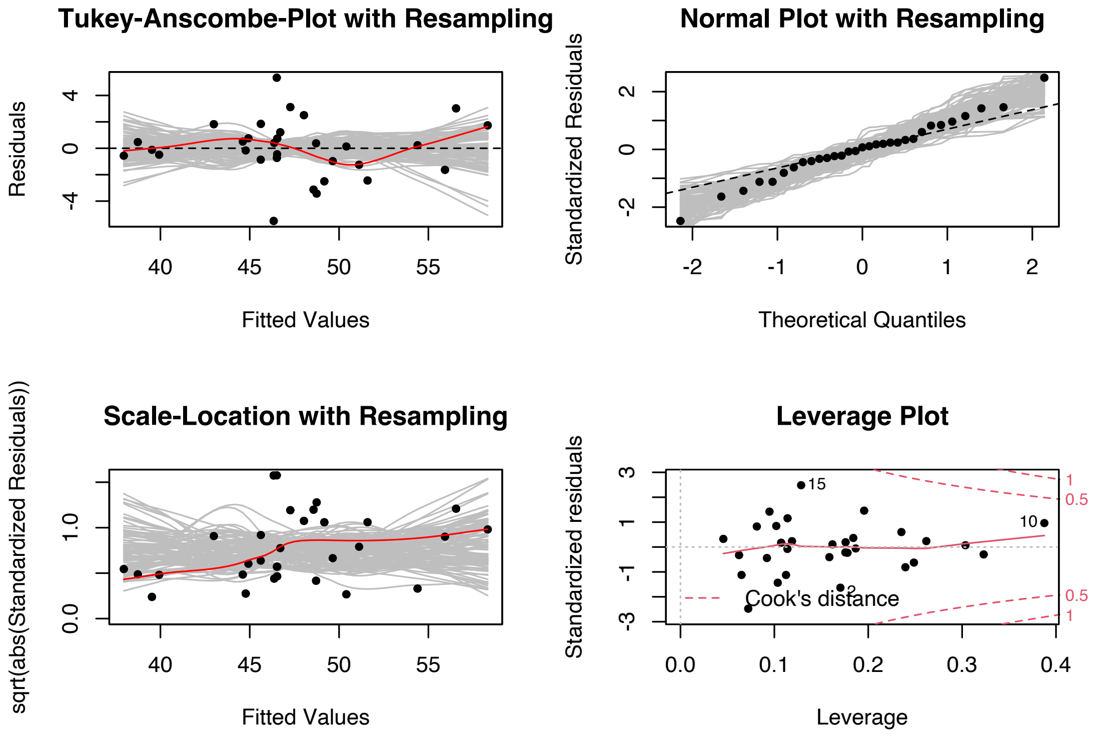
```
##
## Call:
## lm(formula = oxy ~ ., data = fitness.reduced)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5034 -0.9151  0.1373  0.9846  5.3441
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 156.37864   20.38523   7.671 3.85e-08 ***
## age          -0.17542    0.08537  -2.055  0.05008 .
## intensity   -66.75605   20.39749  -3.273  0.00301 **
## runtime      -2.78418    0.33333  -8.353 7.81e-09 ***
## weight       -0.07759    0.05348  -1.451  0.15877
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.305 on 26 degrees of freedom
## Multiple R-squared:  0.8377,Adjusted R-squared:  0.8127
## F-statistic: 33.55 on 4 and 26 DF,  p-value: 6.45e-10
```

```
anova(fit.reduced, fit.full)
```

```
## Analysis of Variance Table
##
## Model 1: oxy ~ age + intensity + runtime + weight
## Model 2: oxy ~ age + weight + runtime + rstpulse + runpulse + inte
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     26 138.18
## 2     24 131.09  2    7.0992 0.6499 0.5311
```

According to the ANOVA-test, the null hypothesis $\beta_{\text{rstpulse}} = \beta_{\text{runpulse}} = \beta_{\text{weight}} = 0$ cannot be rejected. Thus the model selection consequently omitted the corresponding predictor variables.

```
resplot(fit.reduced)
```



The residual plots do not point to any systematic errors.

g) According to our results, the rate of oxygen consumption could be modeled with the variables `running time`, `age` and `intensity`. This yields an R^2 value of approximatly 0.82, i.e., the rate of oxygen consumption can be explained rather well, however not perfectly. It is difficult to decide whether this is sufficient for practical purposes. We thus cannot conclude on the basis of our results whether this model is applicable. The trade-off between costs and loss of precision would need to be assessed further.

Solution 4.2

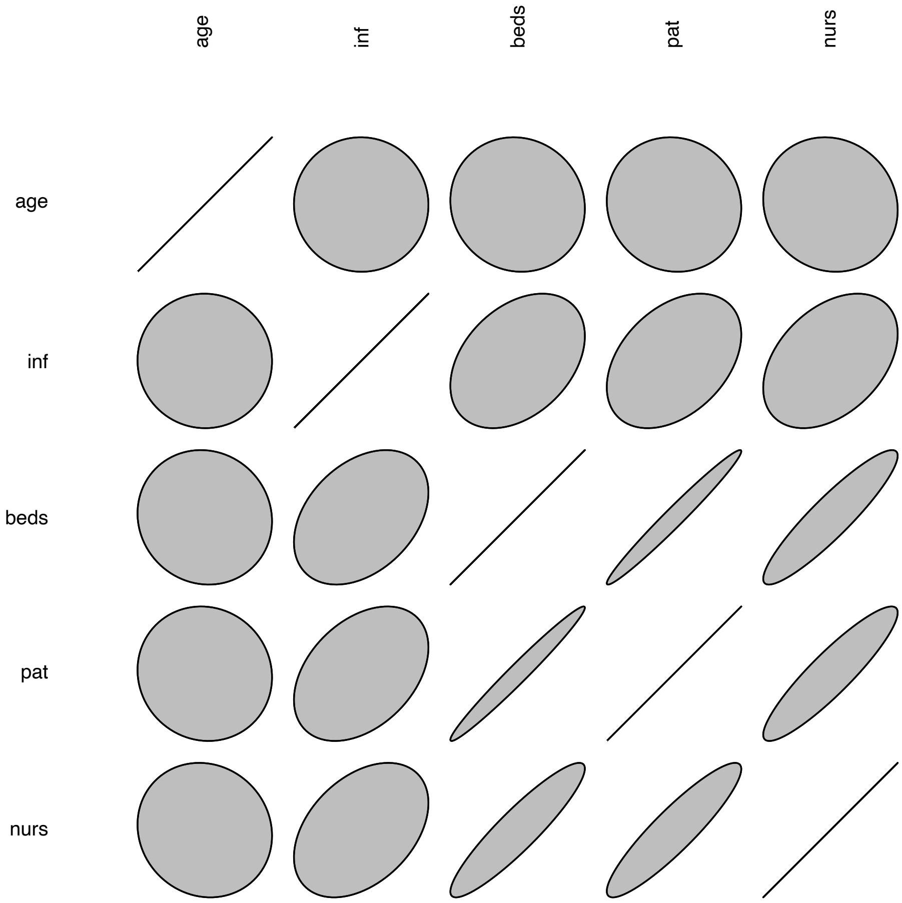
```
load("Daten/senic.rda")
str(senic)


## 'data.frame': 113 obs. of  12 variables:
##  $ id     : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ length: num  7.13 8.82 8.34 8.95 11.2 ...
##  $ age    : num  55.7 58.2 56.9 53.7 56.5 50.9 57.8 45.7 48.2 56.3 ...
##  $ inf    : num  4.1 1.6 2.7 5.6 5.7 5.1 4.6 5.4 4.3 5.3 ...
##  $ cult  : num  9 3.8 8.1 18.9 34.5 21.9 16.7 60.5 24.4 29.6 ...
##  $ xray  : num  39.6 51.7 74 122.8 88.9 ...
##  $ beds  : num  279 80 107 147 180 150 186 640 182 85 ...
##  $ school: num  2 2 2 2 2 2 2 1 2 2 ...
##  $ region: Factor w/ 4 levels "NE","N","S","W": 4 2 3 4 1 2 3 2 3 1 ..
##  $ pat   : num  207 51 82 53 134 147 151 399 130 59 ...
##  $ nurs  : num  241 52 54 148 151 106 129 360 118 66 ...
##  $ serv  : num  60 40 20 40 40 40 40 60 40 40 ...

senic$school <- as.factor(senic$school)
str(senic)

## 'data.frame': 113 obs. of  12 variables:
##  $ id    : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ length: num  7.13 8.82 8.34 8.95 11.2 ...
##  $ age   : num  55.7 58.2 56.9 53.7 56.5 50.9 57.8 45.7 48.2 56.3 ...
##  $ inf   : num  4.1 1.6 2.7 5.6 5.7 5.1 4.6 5.4 4.3 5.3 ...
##  $ cult  : num  9 3.8 8.1 18.9 34.5 21.9 16.7 60.5 24.4 29.6 ...
##  $ xray  : num  39.6 51.7 74 122.8 88.9 ...
##  $ beds  : num  279 80 107 147 180 150 186 640 182 85 ...
##  $ school: Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 1 2 2 ...
##  $ region: Factor w/ 4 levels "NE","N","S","W": 4 2 3 4 1 2 3 2 3 1 ..
##  $ pat   : num  207 51 82 53 134 147 151 399 130 59 ...
##  $ nurs  : num  241 52 54 148 151 106 129 360 118 66 ...
##  $ serv  : num  60 40 20 40 40 40 40 60 40 40 ...
```

a) We check the correlations between the continuous predictors (without response variable):

```
senic <- senic[, c("length", "age", "inf", "region", "beds",
    "pat", "nurs")]
indices_ignore <- which(is.element(colnames(senic), c("length",
    "region")))
cor(senic[, -indices_ignore])

##               age          inf        beds         pat
## age   1.000000000 -0.006266807 -0.05882316 -0.05477467
## inf  -0.006266807  1.000000000  0.36917855  0.39070521
## beds -0.058823160  0.369178549  1.00000000  0.98099774
## pat  -0.054774667  0.390705214  0.98099774  1.00000000
## nurs -0.082944616  0.402911390  0.91550415  0.90789698
##              nurs
## age  -0.08294462
## inf   0.40291139
## beds  0.91550415
## pat   0.90789698
## nurs  1.00000000
```

Graphical illustration of correlations:

```
library(ellipse)
plotcorr(cor(senic[, -indices_ignore]), cex.lab = 0.75,
    mar = c(1, 1, 1, 1))
```



We observe that **beds**, **pat** and **nurs** are strongly correlated. This is to be expected since these variables all measure the size of a hospital.

b) We will leave the variable **pat** unmodified because it is definitely a key factor to be taken into account when **length** is the response variable. We transform the other variables to solve the high-correlation problem without having to omit them in the regression model. Hence, we will substitute **beds** by **pat**/**beds** and **nurs** by **pat**/**nurs**.

Before combining the variables, we check whether **beds** and **nurs** contain zeros:

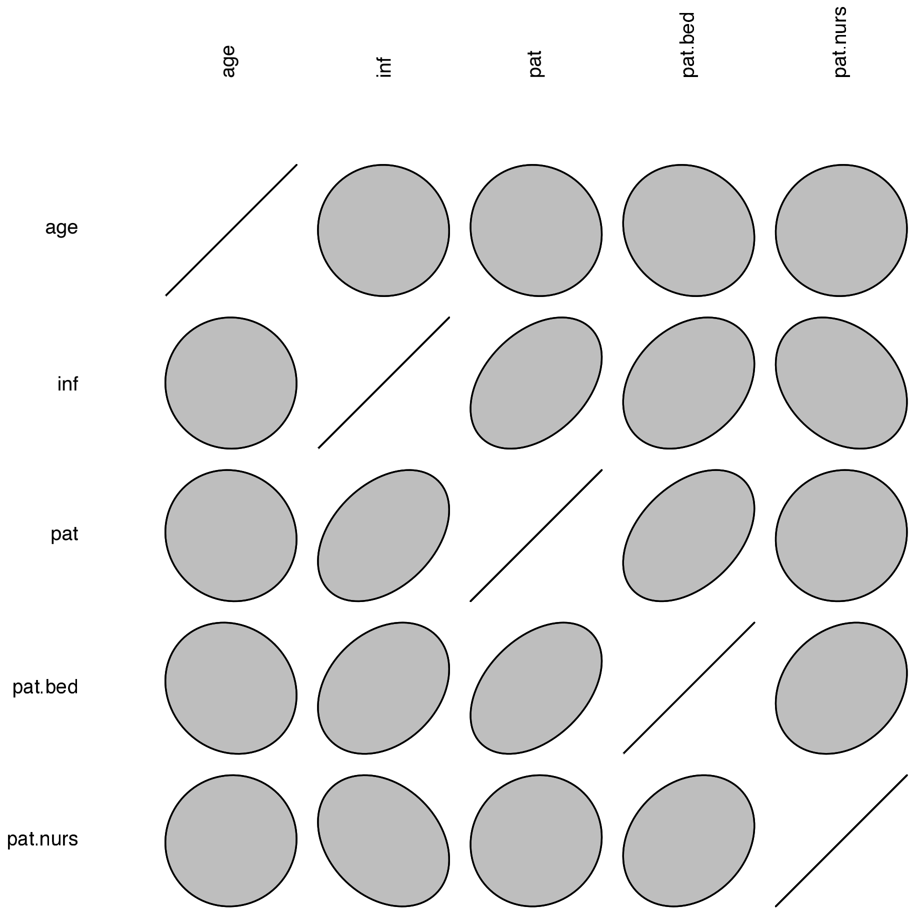
```
any(senic$beds == 0)

## [1] FALSE

any(senic$nurs == 0)

## [1] FALSE
```

Now we combine the variables to new variables and check the correlations again.

```
senic.02 <- data.frame(length = senic$length, age = senic$age,
    inf = senic$inf, region = senic$region, pat = senic$pat,
    pat.bed = senic$pat/senic$beds, pat.nurs = senic$pat/senic$nurs)
cor(senic.02[, -indices_ignore])

##                  age          inf         pat
## age       1.000000000 -0.006266807 -0.05477467
## inf      -0.006266807  1.000000000  0.39070521
## pat      -0.054774667  0.390705214  1.00000000
## pat.bed  -0.109605797  0.289733778  0.41510791
## pat.nurs  0.026954588 -0.285984796  0.05659985
##             pat.bed    pat.nurs
## age      -0.1096058  0.02695459
## inf       0.2897338 -0.28598480
## pat       0.4151079  0.05659985
## pat.bed   1.0000000  0.22893307
## pat.nurs  0.2289331  1.00000000
```

Graphical illustration of the correlations after variable transformations:

```
plotcorr(cor(senic.02[, -indices_ignore]), cex.lab = 0.75,
    mar = c(1, 1, 1, 1))
```



The correlations could be strongly reduced and we still dispose of information with respect to the variables **beds** and **nurs**.

c) **Residual Plots (optional, for advanced readers)** In many applied problems, it is very interesting to understand and visualize the relation between the response Y and some arbitrary predictor X_k. However, a plot of Y versus X_k probably is deceiving, because in a multiple regression setting, all other predictors $X_1, X_2, \ldots, X_{k-1}, X_{k+1}, \ldots, X_p$ will simultaneously have an effect on the response.

Hence, what we should aim for is displaying the relation of Y versus X_k in the presence of the other predictors. That is what the partial residual plot does. We thus generate an updated Y variable, where the effect of all other predictors is removed from the response.

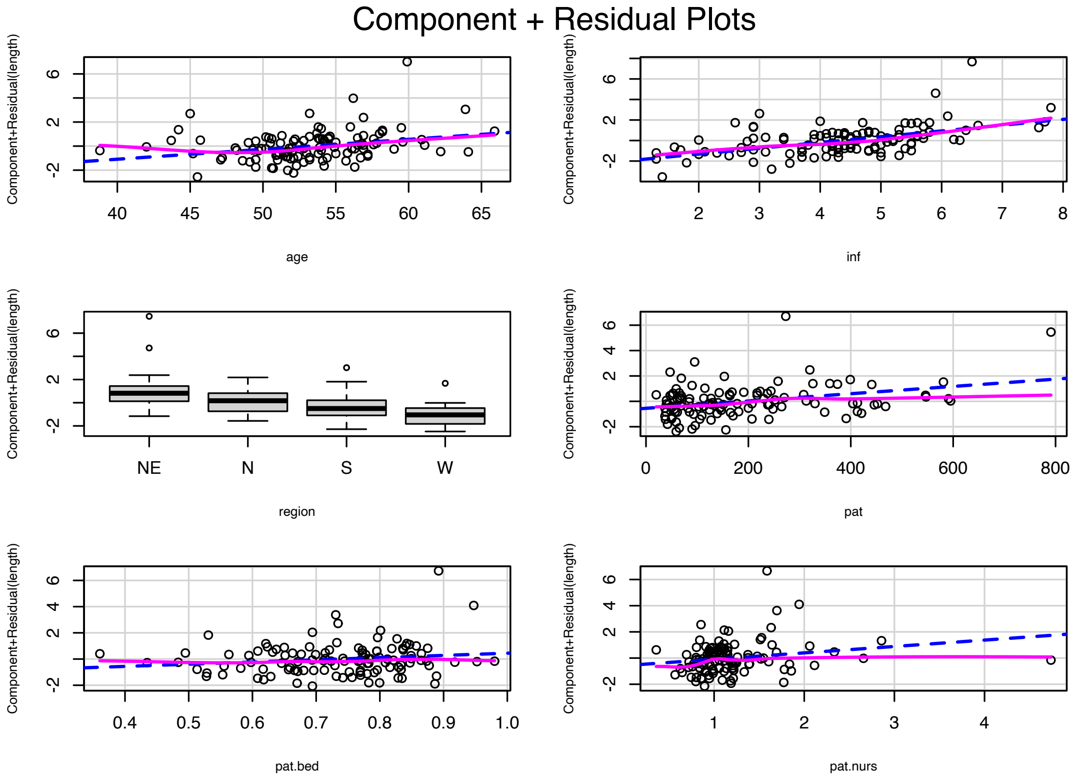
Mathematically, the partial residuals for predictor X_k are:

$$\begin{aligned} Y - \sum_{j \neq k} \hat{\beta}_k X_k &= \hat{Y} + R - \sum_{j \neq k} \hat{\beta}_k X_k \\ &= \hat{\beta}_k X_k + R \end{aligned}$$

where R denotes the residuals determined for the multiple linear regression model. The residual plots are generated by plotting the partial residuals, that is for every observation i, we plot $\hat{\beta}_k X_i^{(k)} + R_i$ versus the predictor $X_i^{(k)}$.

Let us have a look at the partial residual plots:

```r
fit.02 <- lm(length ~ age + inf + region + pat + pat.bed +
    pat.nurs, data = senic.02)
library(car)
crPlots(fit.02, cex.lab = 0.75)
```



The partial residual plots are enhanced with the red dashed line that illustrates the actual fit according to the multiple linear regression model, that is $\hat{\beta}_k X_i^{(k)}$ versus $X_i^{(k)}$. The green solid line is a smoother that was added for visualizing the (true) relation between partial residual and predictor.

If there appears a significant difference between their actual, linear fit and the true relation indicated by the smoother, one should improve the model. Sometimes, we can transform predictors to achieve this; at other times adding additional predictors and/or interaction terms may help.

The variables `length`, `pat` and `pat.nurs` need to be transformed.

We check for zeros in `pat` and `length`:

```
any(senic.02$length == 0)

## [1] FALSE

any(senic.02$pat == 0)

## [1] FALSE
```

Given that there are no zeros in these variables, we are free to transform the predictors as follows:

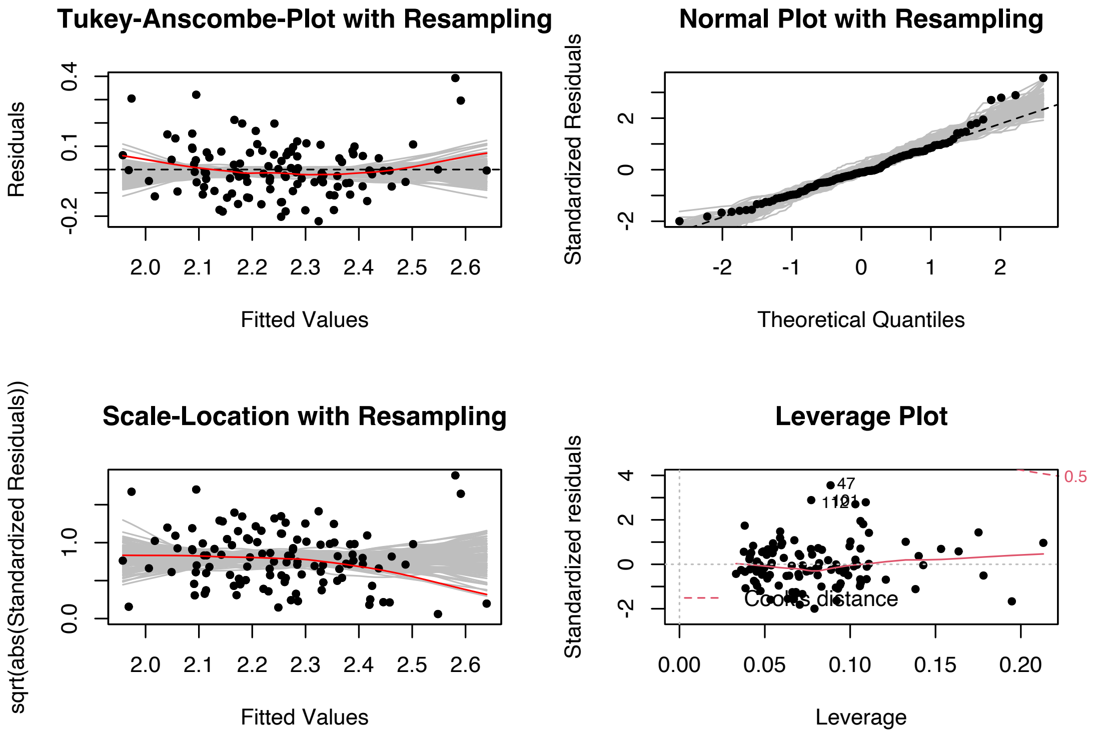
```
senic.03 <- senic.02
senic.03$length <- log(senic.02$length)
senic.03$pat <- log(senic.02$pat)
senic.03$pat.nurs <- log(senic.02$pat.nurs)
```

d) We fit a linear regression model:

```
fit.P1 <- lm(length ~ age + inf + region + pat + pat.bed +
    pat.nurs, data = senic.03)
summary(fit.P1)

##
## Call:
## lm(formula = length ~ age + inf + region + pat + pat.bed + pat.nur
##     data = senic.03)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22160 -0.07198 -0.01166  0.06382  0.39264
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.379579   0.178646   7.722 7.39e-12 ***
## age          0.007645   0.002551   2.997 0.003412 **
## inf          0.053916   0.010312   5.228 8.88e-07 ***
## regionN     -0.074073   0.031132  -2.379 0.019168 *
## regionS      -0.121379   0.030443  -3.987 0.000125 ***
## regionW      -0.200437   0.039882  -5.026 2.10e-06 ***
## pat           0.047034   0.017795   2.643 0.009485 **
## pat.bed       0.106392   0.124304   0.856 0.394020
## pat.nurs      0.073836   0.037202   1.985 0.049808 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1156 on 104 degrees of freedom
## Multiple R-squared:  0.6081,Adjusted R-squared:  0.578
## F-statistic: 20.17 on 8 and 104 DF,  p-value: < 2.2e-16
```

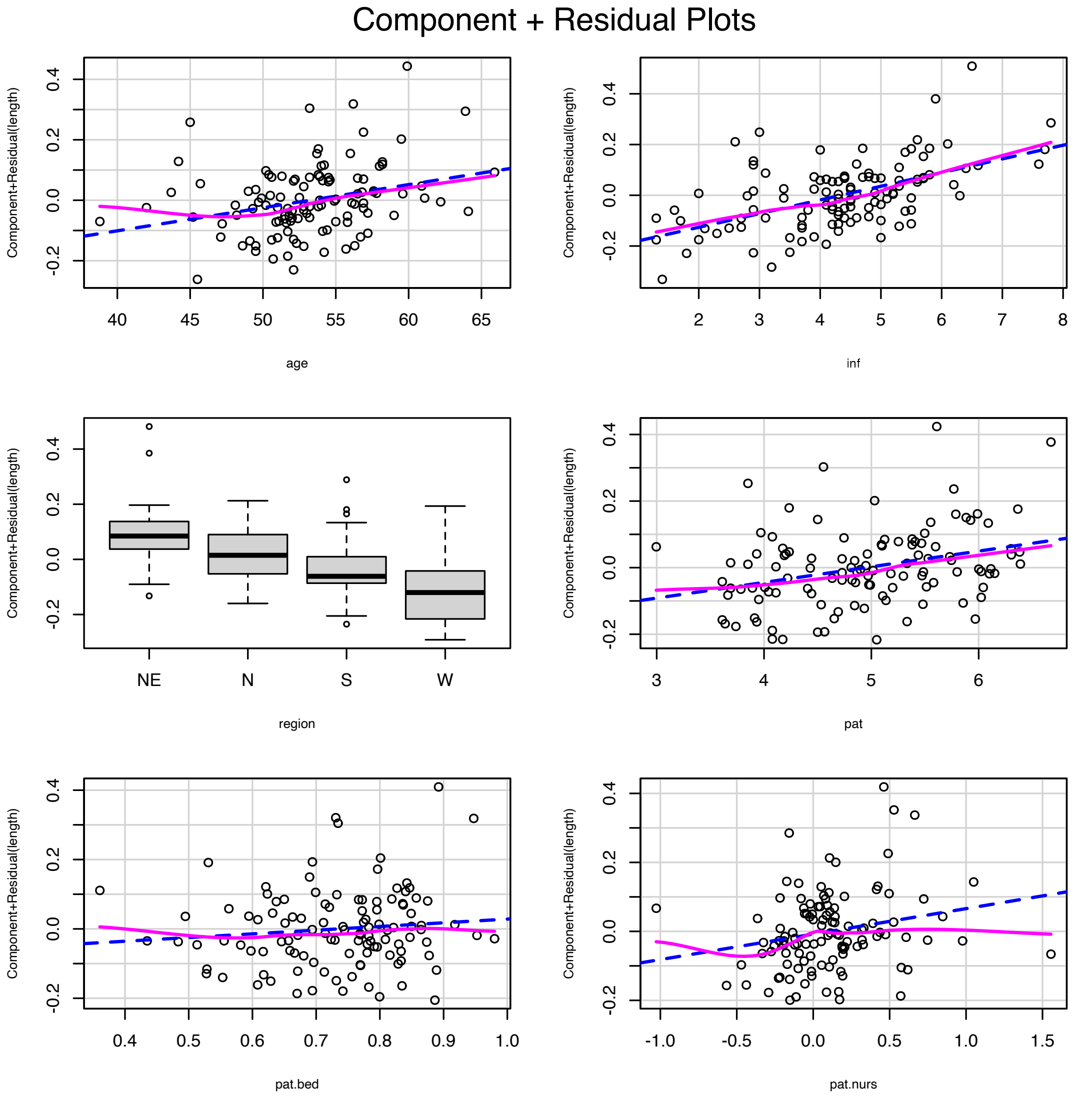
From the summary output we conclude that `pat.bed` is statistically not significant and a variable selection may be appropriate (see next question).

```
resplot(fit.P1)
```



Based on the model diagnostics plots we note that there are three outliers, i.e. observations 47, 101, and 112. However, since their Cook's distance is below 0.5, they do not significantly influence the model fit and we proceed with our analysis. The assumptions concerning linearity and constant variance seem to be satisfied. The QQ-plot does not look perfect but we may assume that the normality assumption still is fulfilled. Now we visualize our model by means of partial residual plots.

```
library(car)
crPlots(fit.P1, cex.lab = 0.75)
```



As it can be deduced from these plots, the predictor `pat.bed` does not have much explanatory power, and indeed, its p-value is rather large as we have seen in the **R** summary.

e) Backward stepwise selection:

```
fit.B <- step(fit.P1, direction = "backward")

## Start:  AIC=-478.96
## length ~ age + inf + region + pat + pat.bed + pat.nurs
##
##             Df Sum of Sq    RSS     AIC
## - pat.bed    1   0.00979 1.4000 -480.17
## <none>                   1.3902 -478.96
## - pat.nurs   1   0.05266 1.4429 -476.76
## - pat        1   0.09339 1.4836 -473.62
## - age        1   0.12006 1.5103 -471.60
## - region     3   0.41062 1.8009 -455.72
## - inf        1   0.36544 1.7557 -454.59
##
## Step:  AIC=-480.17
## length ~ age + inf + region + pat + pat.nurs
##
##             Df Sum of Sq    RSS     AIC
## <none>                   1.4000 -480.17
## - pat.nurs   1   0.06947 1.4695 -476.70
## - age        1   0.11399 1.5140 -473.33
## - pat        1   0.13498 1.5350 -471.77
## - inf        1   0.39587 1.7959 -454.03
## - region     3   0.46502 1.8651 -453.76
```

The backward stepwise selection using AIC removes only the variable `pat.bed` from the model, just as the backward stepwise selection using AIC.

f) Forward stepwise selection:

```
fit.empty <- lm(length ~ NULL, data = senic.03)
scp <- list(lower = ~NULL, upper = ~age + inf + region +
    pat + pat.bed + pat.nurs)
fit.F <- step(fit.empty, scope = scp, direction = "forward")
```

```
## Start:  AIC=-389.11
## length ~ NULL
##
##             Df Sum of Sq    RSS     AIC
## + inf        1   1.08286 2.4646 -428.27
## + pat        1   0.94180 2.6057 -421.98
## + region     3   0.98268 2.5648 -419.76
## + pat.bed    1   0.69376 2.8537 -411.70
## + age        1   0.10368 3.4438 -390.46
## + pat.nurs   1   0.07906 3.4684 -389.66
## <none>                   3.5475 -389.11
```

```
##
## Step:  AIC=-428.27
## length ~ inf
##
##             Df Sum of Sq    RSS     AIC
## + region    3   0.71923 1.7454 -461.26
## + pat.bed   1   0.30829 2.1563 -441.37
## + pat       1   0.29591 2.1687 -440.72
## + pat.nurs  1   0.28973 2.1749 -440.40
## + age       1   0.10793 2.3567 -431.33
## <none>                  2.4646 -428.27
##
## Step:  AIC=-461.26
## length ~ inf + region
##
##             Df Sum of Sq    RSS     AIC
## + pat       1  0.151470 1.5939 -469.52
## + pat.nurs  1  0.128904 1.6165 -467.93
## + age       1  0.086145 1.6592 -464.98
## + pat.bed   1  0.079078 1.6663 -464.50
## <none>                  1.7454 -461.26
##
## Step:  AIC=-469.52
## length ~ inf + region + pat
##
##             Df Sum of Sq    RSS     AIC
## + age       1  0.124380 1.4695 -476.70
## + pat.nurs  1  0.079866 1.5140 -473.33
## <none>                  1.5939 -469.52
## + pat.bed   1  0.016785 1.5771 -468.71
##
## Step:  AIC=-476.7
## length ~ inf + region + pat + age
##
##             Df Sum of Sq    RSS     AIC
## + pat.nurs  1  0.069473 1.4000 -480.17
## + pat.bed   1  0.026608 1.4429 -476.76
## <none>                  1.4695 -476.70
##
## Step:  AIC=-480.17
## length ~ inf + region + pat + age + pat.nurs
```

```
##
##           Df Sum of Sq    RSS     AIC
## <none>                 1.4000 -480.17
## + pat.bed  1 0.0097928 1.3902 -478.96
```

We obtain the same result as for the backward stepwise selection using AIC : only the predictor variable **pat.bed** has been removed from the model. Note that this occurs in this particular example and represents rather an exception.

g) Hybrid stepwise selection:

```
step(fit.P1, direction = "both")
```

```
## Start:  AIC=-478.96
## length ~ age + inf + region + pat + pat.bed + pat.nurs
##
##            Df Sum of Sq    RSS     AIC
## - pat.bed   1    0.00979 1.4000 -480.17
## <none>                   1.3902 -478.96
## - pat.nurs  1    0.05266 1.4429 -476.76
## - pat       1    0.09339 1.4836 -473.62
## - age       1    0.12006 1.5103 -471.60
## - region    3    0.41062 1.8009 -455.72
## - inf       1    0.36544 1.7557 -454.59
##
## Step:  AIC=-480.17
## length ~ age + inf + region + pat + pat.nurs
##
##            Df Sum of Sq    RSS     AIC
## <none>                   1.4000 -480.17
## + pat.bed   1    0.00979 1.3902 -478.96
## - pat.nurs  1    0.06947 1.4695 -476.70
## - age       1    0.11399 1.5140 -473.33
## - pat       1    0.13498 1.5350 -471.77
## - inf       1    0.39587 1.7959 -454.03
## - region    3    0.46502 1.8651 -453.76
##
## Call:
## lm(formula = length ~ age + inf + region + pat + pat.nurs, data =
##
## Coefficients:
## (Intercept)          age          inf      regionN
##    1.438983     0.007404     0.055360    -0.078067
##      regionS      regionW          pat     pat.nurs
##    -0.123516    -0.209690     0.052614     0.081985
```

Starting with the full model removes `pat.bed` from the model. Therefore, this method yields the same result as the models `fit.P1`, `fit.B`, and `fit.F`.

```
step(fit.empty, scope = scp, direction = "both")

## Start:  AIC=-389.11
## length ~ NULL
##
##            Df Sum of Sq    RSS     AIC
## + inf       1   1.08286 2.4646 -428.27
## + pat       1   0.94180 2.6057 -421.98
## + region    3   0.98268 2.5648 -419.76
## + pat.bed   1   0.69376 2.8537 -411.70
## + age       1   0.10368 3.4438 -390.46
## + pat.nurs  1   0.07906 3.4684 -389.66
## <none>                  3.5475 -389.11
##
## Step:  AIC=-428.27
## length ~ inf
##
##            Df Sum of Sq    RSS     AIC
## + region    3   0.71923 1.7454 -461.26
## + pat.bed   1   0.30829 2.1563 -441.37
## + pat       1   0.29591 2.1687 -440.72
## + pat.nurs  1   0.28973 2.1749 -440.40
## + age       1   0.10793 2.3567 -431.33
## <none>                  2.4646 -428.27
## - inf       1   1.08286 3.5475 -389.11
##
## Step:  AIC=-461.26
## length ~ inf + region
##
##            Df Sum of Sq    RSS     AIC
## + pat       1   0.15147 1.5939 -469.52
## + pat.nurs  1   0.12890 1.6165 -467.93
## + age       1   0.08614 1.6592 -464.98
## + pat.bed   1   0.07908 1.6663 -464.50
## <none>                  1.7454 -461.26
## - region    3   0.71923 2.4646 -428.27
## - inf        1   0.81941 2.5648 -419.76
##
## Step:  AIC=-469.52
## length ~ inf + region + pat
##
##            Df Sum of Sq    RSS     AIC
## + age       1   0.12438 1.4695 -476.70
## + pat.nurs  1   0.07987 1.5140 -473.33
## <none>                  1.5939 -469.52
## + pat.bed   1   0.01678 1.5771 -468.71
## - pat       1   0.15147 1.7454 -461.26
## - inf       1   0.35905 1.9529 -448.56
## - region    3   0.57478 2.1687 -440.72
##
## Step:  AIC=-476.7
## length ~ inf + region + pat + age
##
##            Df Sum of Sq    RSS     AIC
## + pat.nurs  1   0.06947 1.4000 -480.17
## + pat.bed   1   0.02661 1.4429 -476.76
## <none>                  1.4695 -476.70
## - age       1   0.12438 1.5939 -469.52
## - pat       1   0.18970 1.6592 -464.98
## - inf       1   0.33436 1.8039 -455.53
## - region    3   0.53057 2.0001 -447.86
##
## Step:  AIC=-480.17
## length ~ inf + region + pat + age + pat.nurs
##
##            Df Sum of Sq    RSS     AIC
## <none>                  1.4000 -480.17
## + pat.bed   1   0.00979 1.3902 -478.96
## - pat.nurs  1   0.06947 1.4695 -476.70
## - age       1   0.11399 1.5140 -473.33
## - pat       1   0.13498 1.5350 -471.77
## - inf       1   0.39587 1.7959 -454.03
## - region    3   0.46502 1.8651 -453.76
##
## Call:
## lm(formula = length ~ inf + region + pat + age + pat.nurs, data =
##
```

```
## Coefficients:
## (Intercept)          inf      regionN      regionS
##    1.438983     0.055360    -0.078067    -0.123516
##     regionW          pat          age     pat.nurs
##   -0.209690     0.052614     0.007404     0.081985
```

Hybrid stepwise selection starting with the empty model yields the same result as the hybrid stepwise starting with the full model, backward stepwise selection and forward stepwise selection. Note that this is in general not the case: applying these methods with different data could actually lead to different results.

h) The ANOVA-test yields

```
anova(lm(length ~ age + inf + region + pat + pat.nurs, data = senic.(
    lm(length ~ age + inf + region + pat + pat.bed + pat.nurs,
        data = senic.03))

## Analysis of Variance Table
##
## Model 1: length ~ age + inf + region + pat + pat.nurs
## Model 2: length ~ age + inf + region + pat + pat.bed + pat.nurs
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    105 1.4000
## 2    104 1.3902  1 0.0097928 0.7326  0.394
```

We conclude that the predictor variable **pat.bed** can indeed be dropped from the model because of the p-value associated with the F-statistic. We further note that the p-value associated with the F-statistic is in this case identical to the p-value associated with the t-statistic which we have observed in **R**-output of the full regression model.

Solution 4.3

a)

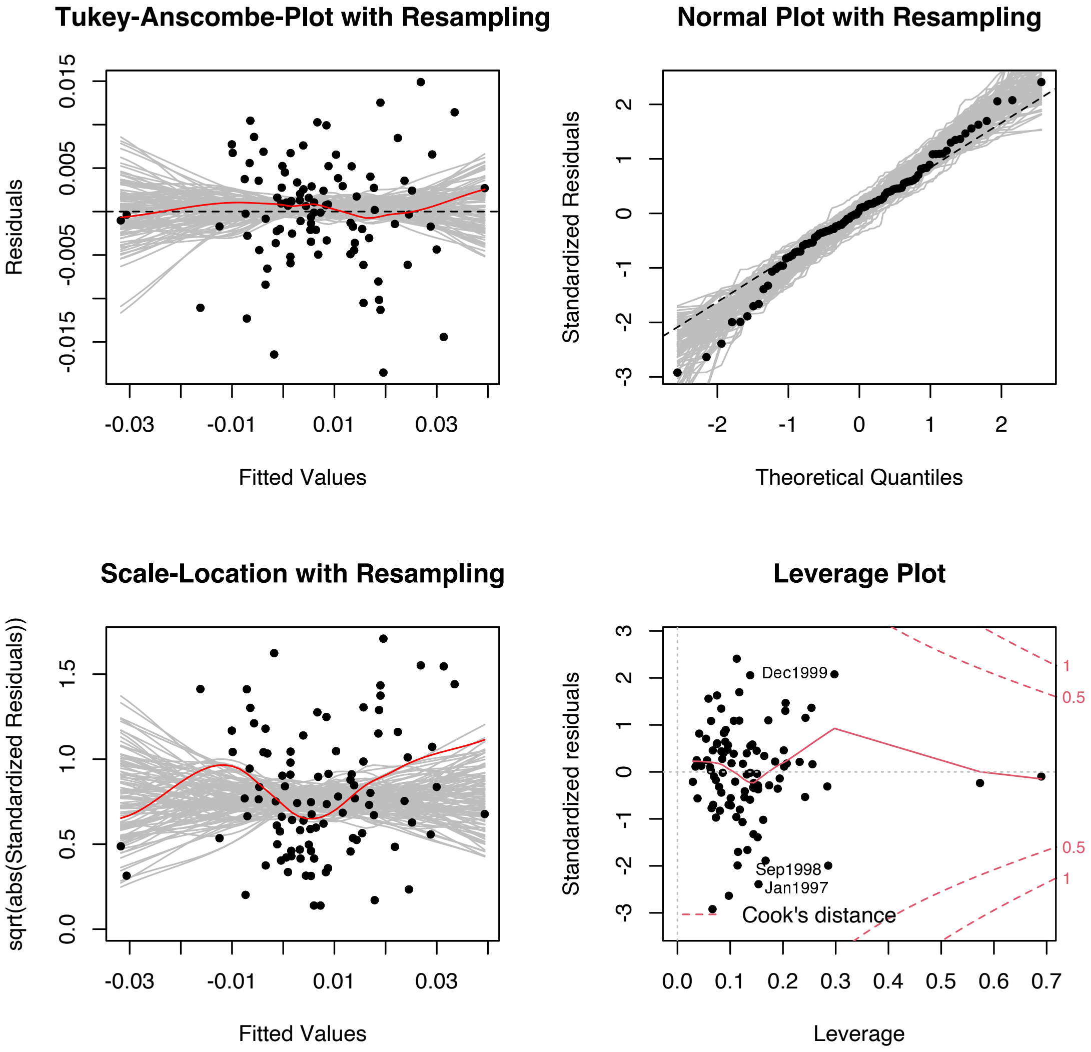
```
## Load data
load("Daten/FoHF.rda")
## Fit model with all variables
fit <- lm(FoHF ~ ., data = FoHF)
summary(fit)

##
## Call:
## lm(formula = FoHF ~ ., data = FoHF)
##
## Residuals:
```

```
##        Min         1Q     Median         3Q        Max 
## -0.0185186 -0.0031189  0.0004069  0.0035469  0.0148925 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.002256   0.001299  -1.736   0.0862 .  
## RV          -0.388854   0.171151  -2.272   0.0257 *  
## CA           0.238653   0.104522   2.283   0.0250 *  
## FIA          0.363010   0.087832   4.133 8.51e-05 ***
## EMN          0.184766   0.197475   0.936   0.3522    
## ED           0.314914   0.215792   1.459   0.1482    
## DS          -0.007699   0.124324  -0.062   0.9508    
## MA          -0.028413   0.169406  -0.168   0.8672    
## LSE          0.153636   0.099548   1.543   0.1266    
## GM           0.127093   0.086897   1.463   0.1474    
## EM           0.049183   0.035065   1.403   0.1645    
## CTA          0.159225   0.037304   4.268 5.20e-05 ***
## SS           0.032630   0.023424   1.393   0.1673    
## ---
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.006563 on 83 degrees of freedom
## Multiple R-squared:  0.8076,Adjusted R-squared:  0.7798 
## F-statistic: 29.03 on 12 and 83 DF,  p-value: < 2.2e-16
```

Only four variables are significant at the 5 % level in the summary output; two variables are associated with a very small p-value. The multiple R^2 is relatively large having a value of around 80 %. The global F-test is highly significant. In other words, the return of the **FoHF** is explained rather well and not all subindices may be necessary to predict the return of the **FoHF**. Therefore, we can assume that the **FoHF** does not invest in all subindices.

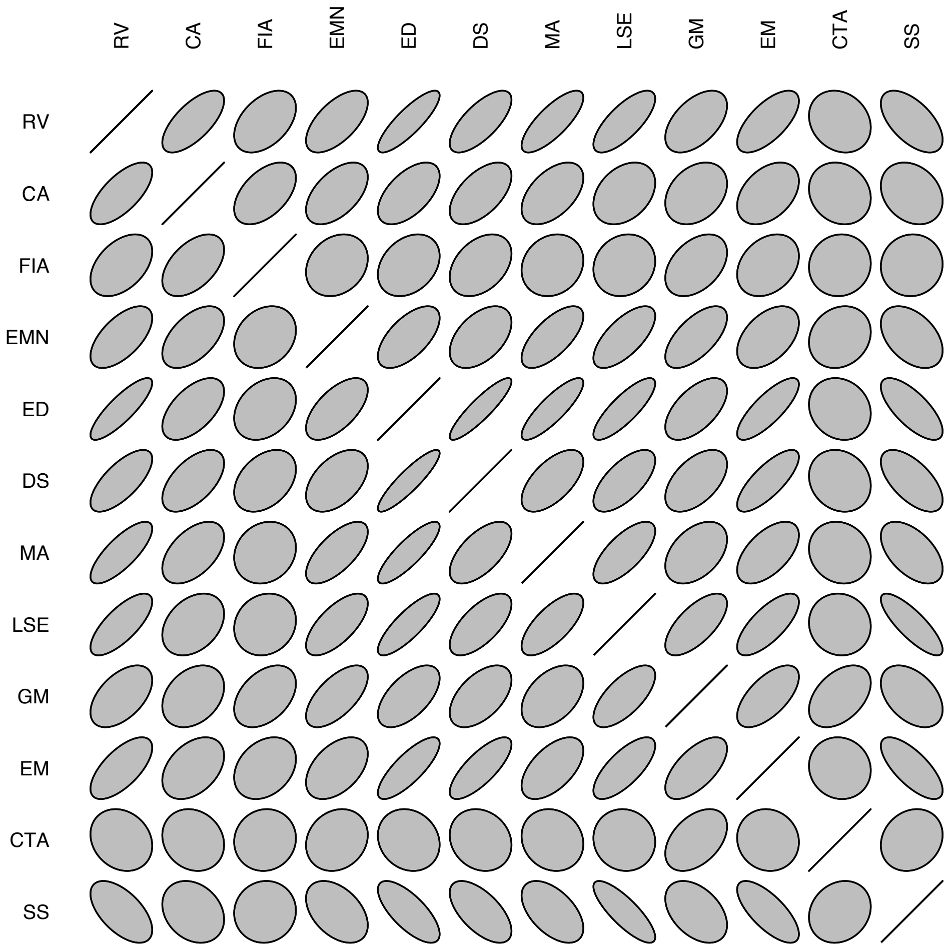
b)

```
par(mfrow = c(2, 2))
source("resplot.R")
resplot(fit)
```



The residual plots do not point to any systematic errors in the model. The normality assumption seems to be satisfied. The smoother in the scale location plot does show some deviations from the horizontal line. It is generally known that finance data shows volatility, i.e. the (conditional) variance is not necessarily constant over time. However, in this case the deviations are not very pronounced, so that the constant variance assumption seems to be satisfied and we can proceed with the analysis. Furthermore, we notice two points with high leverage. Since their Cook's distance is rather small, we tolerate them.

```
## Check for large multicollinearity
library(ellipse)
par(mfrow = c(1, 1))
plotcorr(cor(FoHF[, -1]), cex.lab = 0.75, mar = c(1, 1,
    1, 1))
```

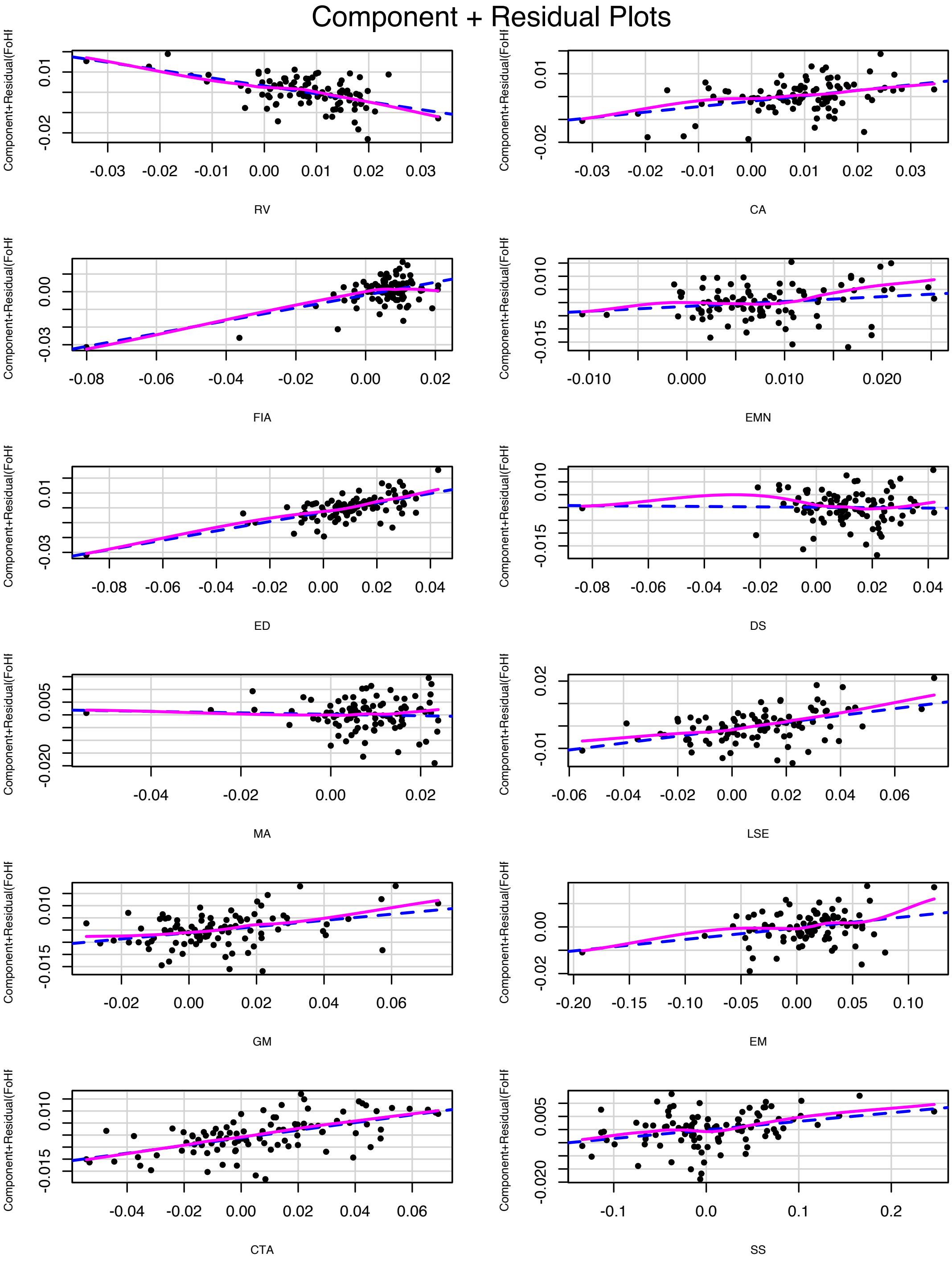


We check for high multicollinearity of the predictors by plotting the pairwise correlations and by computing the VIFs.

```
library(faraway)
vif(fit)

##         RV         CA        FIA        EMN         ED
##   6.387024   2.982646   2.271113   3.672017  29.973694
##         DS         MA        LSE         GM         EM
##   9.404810   8.001994  10.046374   5.699120   4.255477
##        CTA         SS
##   2.232320   4.972861
```

The VIF of the variables **ED** and **LSE** are larger than 10 and thus beyond the threshold of what can be tolerated.

```
library(car)
crPlots(fit, pch = 20, layout = c(6, 2), cex.lab = 0.75)
```



The partial residual plots allow us to check whether all predictors have been included in the model in the correct form. The plots show some deviations but these are neither very strong nor systematic.

c) We noticed that there is a problem in the data due to multicollinearity. The possible therapeutic measures are limited in this case: since the `FoHF` can invest in all of the given subindices, we cannot amputate some of them. Similarly, creating new variables in this context does not make sense and we cannot transform the predictors either - the `FoHF` invests in the subindices which contribute directly and linearly to the return of the `FoHF`. The multicollinearity problem is caused by some highly correlated subindices.

   It is possible, however, to perform a variable selection which will hopefully alleviate the multicollinearity problem. From the summary output, we conclude that the `FoHF` does not invest in all subindices. In the following subproblem, we shall try to achieve a reduction of the model size so that the final model will only contain those subindices the `FoHF` does in fact invest in.

d) i) Stepwise variable selection, starting with the full model.

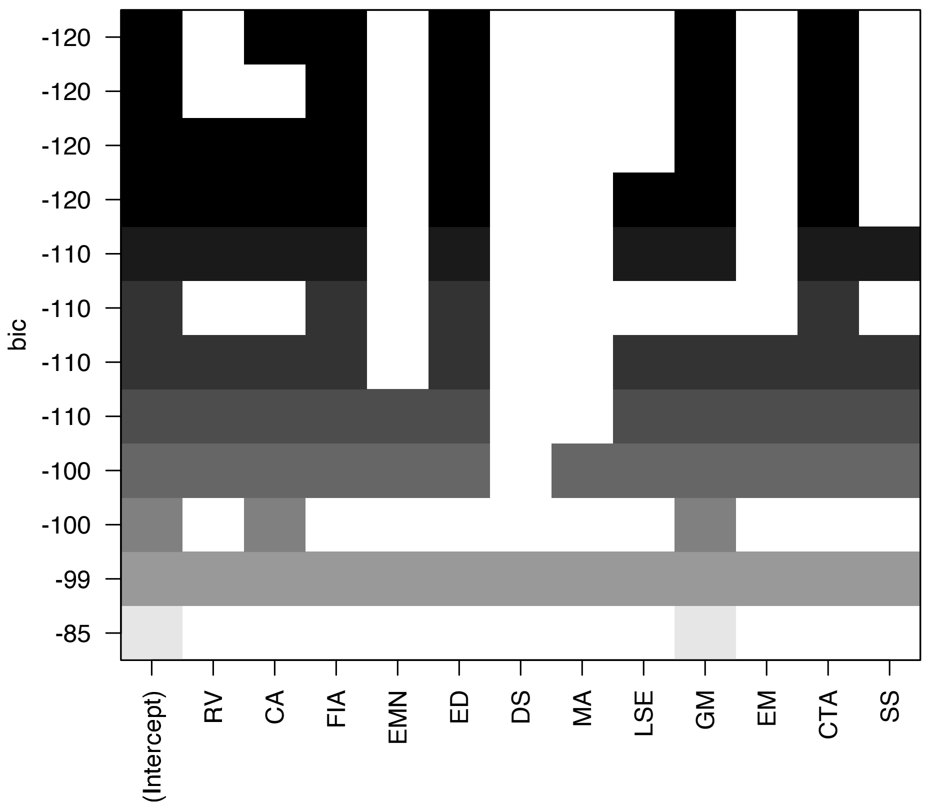
```
## Variable selection with BIC, starting with the full
## model - hybrid stepwise selection
fit.bic.01 <- step(fit, method = "both", k = log(nrow(FoHF)))

## Start:  AIC=-919.68
## FoHF ~ RV + CA + FIA + EMN + ED + DS + MA + LSE + GM + EM + CT
##     SS
##
##        Df  Sum of Sq       RSS     AIC
## - DS    1 0.00000017 0.0035753 -924.24
## - MA    1 0.00000121 0.0035763 -924.21
## - EMN   1 0.00003771 0.0036128 -923.24
## - SS    1 0.00008358 0.0036587 -922.03
## - EM    1 0.00008474 0.0036598 -922.00
## - ED    1 0.00009173 0.0036668 -921.81
## - GM    1 0.00009214 0.0036672 -921.80
## - LSE   1 0.00010260 0.0036777 -921.53
## <none>               0.0035751 -919.68
## - RV    1 0.00022234 0.0037974 -918.45
## - CA    1 0.00022456 0.0037996 -918.40
## - FIA   1 0.00073576 0.0043108 -906.28
## - CTA   1 0.00078472 0.0043598 -905.20
##
## Step:  AIC=-924.24
## FoHF ~ RV + CA + FIA + EMN + ED + MA + LSE + GM + EM + CTA +
##     SS
##
##        Df  Sum of Sq       RSS     AIC
## - MA    1 0.00000108 0.0035763 -928.78
## - EMN   1 0.00003761 0.0036129 -927.80
## - SS    1 0.00008344 0.0036587 -926.59
## - EM    1 0.00008500 0.0036603 -926.55
## - GM    1 0.00009213 0.0036674 -926.36
## - LSE   1 0.00010811 0.0036834 -925.95
## <none>               0.0035753 -924.24
## - RV    1 0.00022710 0.0038024 -922.89
## - CA    1 0.00022724 0.0038025 -922.89
## - ED    1 0.00023020 0.0038055 -922.82
## - FIA   1 0.00073934 0.0043146 -910.76
## - CTA   1 0.00079410 0.0043693 -909.55
##
## Step:  AIC=-928.78
## FoHF ~ RV + CA + FIA + EMN + ED + LSE + GM + EM + CTA + SS
##
##        Df  Sum of Sq       RSS     AIC
## - EMN   1 0.00003909 0.0036154 -932.30
## - SS    1 0.00008398 0.0036603 -931.11
## - EM    1 0.00008759 0.0036639 -931.02
## - GM    1 0.00010058 0.0036769 -930.68
## - LSE   1 0.00010832 0.0036846 -930.48
## <none>               0.0035763 -928.78
## - CA    1 0.00024101 0.0038173 -927.08
## - RV    1 0.00026057 0.0038369 -926.59
## - ED    1 0.00035144 0.0039278 -924.34
## - CTA   1 0.00079349 0.0043698 -914.11
## - FIA   1 0.00079685 0.0043732 -914.03
##
## Step:  AIC=-932.3
## FoHF ~ RV + CA + FIA + ED + LSE + GM + EM + CTA + SS
##
##        Df  Sum of Sq       RSS     AIC
## - EM    1 0.00007430 0.0036897 -934.91
## - SS    1 0.00010742 0.0037228 -934.05
## - GM    1 0.00012492 0.0037403 -933.60
## <none>               0.0036154 -932.30
## - LSE   1 0.00018537 0.0038008 -932.06
## - RV    1 0.00025580 0.0038712 -930.30
## - ED    1 0.00035729 0.0039727 -927.82
## - CA    1 0.00040461 0.0040200 -926.68
## - FIA   1 0.00075873 0.0043741 -918.57
## - CTA   1 0.00088516 0.0045006 -915.84
##
## Step:  AIC=-934.91
## FoHF ~ RV + CA + FIA + ED + LSE + GM + CTA + SS
##
##        Df  Sum of Sq       RSS     AIC
## - SS    1 0.00005369 0.0037434 -938.09
## - LSE   1 0.00014269 0.0038324 -935.83
## <none>               0.0036897 -934.91
## - GM    1 0.00024280 0.0039325 -933.36
## - RV    1 0.00026091 0.0039506 -932.92
## - CA    1 0.00037752 0.0040672 -930.12
## - ED    1 0.00058092 0.0042706 -925.44
## - CTA   1 0.00081755 0.0045073 -920.26
## - FIA   1 0.00083132 0.0045210 -919.97
##
## Step:  AIC=-938.09
## FoHF ~ RV + CA + FIA + ED + LSE + GM + CTA
##
##        Df  Sum of Sq       RSS     AIC
## - LSE   1 0.00009111 0.0038345 -940.34
## <none>               0.0037434 -938.09
## - RV    1 0.00024437 0.0039878 -936.58
## - GM    1 0.00027617 0.0040196 -935.82
## - CA    1 0.00038539 0.0041288 -933.24
## - ED    1 0.00052919 0.0042726 -929.96
## - CTA   1 0.00083822 0.0045816 -923.25
## - FIA   1 0.00092714 0.0046706 -921.41
##
## Step:  AIC=-940.34
## FoHF ~ RV + CA + FIA + ED + GM + CTA
##
##        Df  Sum of Sq       RSS     AIC
## - RV    1 0.00016854 0.0040031 -940.78
## <none>               0.0038345 -940.34
## - CA    1 0.00031176 0.0041463 -937.40
## - GM    1 0.00072123 0.0045558 -928.36
## - CTA   1 0.00074886 0.0045834 -927.78
## - ED    1 0.00083966 0.0046742 -925.90
## - FIA   1 0.00085130 0.0046858 -925.66
##
## Step:  AIC=-940.78
## FoHF ~ CA + FIA + ED + GM + CTA
##
##         Df  Sum of Sq       RSS     AIC
## <none>                0.0040031 -940.78
## - CA    1 0.00019591 0.0041990 -940.76
## - GM    1 0.00066084 0.0046639 -930.67
## - ED    1 0.00071917 0.0047222 -929.48
## - FIA   1 0.00073423 0.0047373 -929.18
## - CTA   1 0.00089226 0.0048953 -926.03
```

ii) Stepwise variable selection, starting with the empty model.

```
## Variable selection with BIC, starting with the
## empty model - hybrid stepwise selection
fit.null <- lm(FoHF ~ 1, data = FoHF)
scopi <- list(lower = formula(fit.null), upper = formula(fit))
fit.bic.02 <- step(fit.null, scope = scopi, k = log(nrow(FoHF)))

## Start:  AIC=-816.23
## FoHF ~ 1
##
##         Df Sum of Sq       RSS     AIC
## + GM    1 0.0116463 0.0069343 -906.29
## + ED    1 0.0072492 0.0113314 -859.15
## + EMN   1 0.0071500 0.0114306 -858.31
## + DS    1 0.0066117 0.0119689 -853.89
## + EM    1 0.0065705 0.0120101 -853.56
## + FIA   1 0.0059473 0.0126333 -848.70
## + LSE   1 0.0058787 0.0127020 -848.18
## + RV    1 0.0055859 0.0129947 -846.00
## + CA    1 0.0047059 0.0138748 -839.71
## + MA    1 0.0043282 0.0142525 -837.13
## + CTA   1 0.0027357 0.0158449 -826.96
## + SS    1 0.0020455 0.0165351 -822.87
## <none>               0.0185806 -816.23
##
## Step:  AIC=-906.29
## FoHF ~ GM
##
##         Df Sum of Sq       RSS     AIC
## + CA     1 0.0013440 0.0055904 -922.41
## + FIA    1 0.0012867 0.0056477 -921.43
## + DS     1 0.0008103 0.0061241 -913.66
## + ED     1 0.0007262 0.0062081 -912.35
## + RV     1 0.0004910 0.0064434 -908.78
## + MA     1 0.0004643 0.0064700 -908.38
## + EMN    1 0.0004195 0.0065148 -907.72
## <none>               0.0069343 -906.29
## + EM     1 0.0002708 0.0066636 -905.55
## + CTA    1 0.0000318 0.0069025 -902.17
## + LSE    1 0.0000280 0.0069064 -902.11
## + SS     1 0.0000009 0.0069334 -901.74
## - GM     1 0.0116463 0.0185806 -816.23
##
## Step:  AIC=-922.41
## FoHF ~ GM + CA
##
##         Df Sum of Sq       RSS     AIC
## + FIA    1 0.0004944 0.0050959 -926.73
## + CTA    1 0.0003730 0.0052174 -924.47
## <none>               0.0055904 -922.41
## + DS     1 0.0001376 0.0054527 -920.24
## + ED     1 0.0001021 0.0054882 -919.61
## + EM     1 0.0000447 0.0055457 -918.61
## + MA     1 0.0000393 0.0055511 -918.52
## + SS     1 0.0000330 0.0055574 -918.41
## + EMN    1 0.0000191 0.0055712 -918.17
## + RV     1 0.0000025 0.0055878 -917.89
## + LSE    1 0.0000024 0.0055880 -917.88
## - CA     1 0.0013440 0.0069343 -906.29
## - GM     1 0.0082844 0.0138748 -839.71
##
## Step:  AIC=-926.73
## FoHF ~ GM + CA + FIA
##
##         Df Sum of Sq       RSS     AIC
## + CTA    1 0.0003737 0.0047222 -929.48
## <none>               0.0050959 -926.73
## + ED    1 0.0002006 0.0048953 -926.03
## + MA    1 0.0001505 0.0049454 -925.05
## + DS    1 0.0001452 0.0049507 -924.95
## + EMN   1 0.0001411 0.0049548 -924.87
## + EM    1 0.0000704 0.0050255 -923.51
## + LSE   1 0.0000383 0.0050577 -922.89
## - FIA   1 0.0004944 0.0055904 -922.41
## + SS    1 0.0000056 0.0050904 -922.27
## + RV    1 0.0000055 0.0050904 -922.27
## - CA    1 0.0005517 0.0056477 -921.43
## - GM    1 0.0063437 0.0114396 -853.67
##
## Step:  AIC=-929.48
## FoHF ~ GM + CA + FIA + CTA
##
##          Df Sum of Sq       RSS     AIC
## + ED    1 0.0007192 0.0040031 -940.78
## + DS    1 0.0004934 0.0042289 -935.51
## + LSE   1 0.0003982 0.0043240 -933.37
## + EM    1 0.0003709 0.0043513 -932.77
## + MA    1 0.0003524 0.0043698 -932.36
## <none>              0.0047222 -929.48
## + SS    1 0.0001590 0.0045633 -928.20
## + EMN   1 0.0001447 0.0045775 -927.91
## - CTA   1 0.0003737 0.0050959 -926.73
## + RV    1 0.0000481 0.0046742 -925.90
## - FIA   1 0.0004951 0.0052174 -924.47
## - CA    1 0.0008029 0.0055252 -918.97
## - GM    1 0.0035251 0.0082473 -880.52
##
## Step:  AIC=-940.78
## FoHF ~ GM + CA + FIA + CTA + ED
##
##          Df  Sum of Sq       RSS     AIC
## <none>               0.0040031 -940.78
## - CA    1 0.00019591 0.0041990 -940.76
## + RV    1 0.00016854 0.0038345 -940.34
## + EMN   1 0.00004912 0.0039539 -937.40
## + EM    1 0.00002789 0.0039752 -936.88
## + LSE   1 0.00001528 0.0039878 -936.58
## + SS    1 0.00001019 0.0039929 -936.46
## + DS     1 0.00000805 0.0039950 -936.41
## + MA     1 0.00000154 0.0040015 -936.25
## - GM     1 0.00066084 0.0046639 -930.67
## - ED     1 0.00071917 0.0047222 -929.48
## - FIA    1 0.00073423 0.0047373 -929.18
## - CTA    1 0.00089226 0.0048953 -926.03
```

iii) All Subsets variable selection.

```
## All Subsets Search
library(leaps)
out <- regsubsets(FoHF ~ ., nvmax = 12, data = FoHF)
plot(out)
```



```
coef(out, 5)
```

```
##   (Intercept)            CA           FIA            ED
## -0.001856729  0.175665074  0.298491812  0.265442447
##           GM           CTA
##  0.246942244  0.153503336
```

All three variable selection methods yield the same model. It contains the subindices **CA**, **FIA**, **ED**, **GM** and **CTA**. All Subsets search shows that there are three alternative models with almost identical BIC values - they contain 4, 6 and 7 predictors respectively.

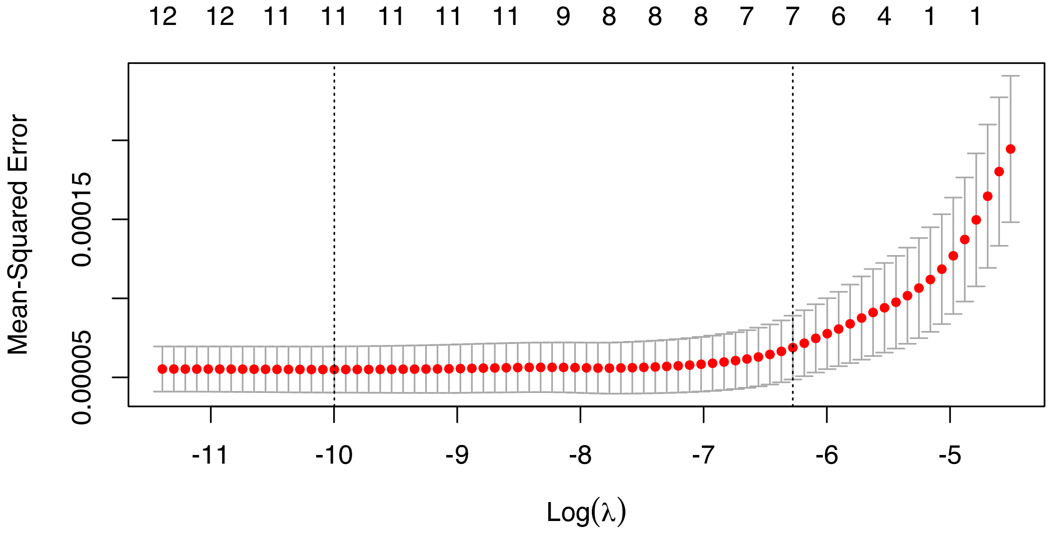
e)

```
anova(lm(FoHF ~ CA + FIA + ED + GM + CTA, data = FoHF),
    lm(FoHF ~ ., data = FoHF))

## Analysis of Variance Table
##
## Model 1: FoHF ~ CA + FIA + ED + GM + CTA
## Model 2: FoHF ~ RV + CA + FIA + EMN + ED + DS + MA + LSE + GM + E
##     SS
##   Res.Df       RSS Df  Sum of Sq      F Pr(>F)
## 1     90 0.0040031
## 2     83 0.0035751  7 0.00042798 1.4194 0.2085
```

Due to the p-value of the associated F-statistic, removing those variables is indeed justified.

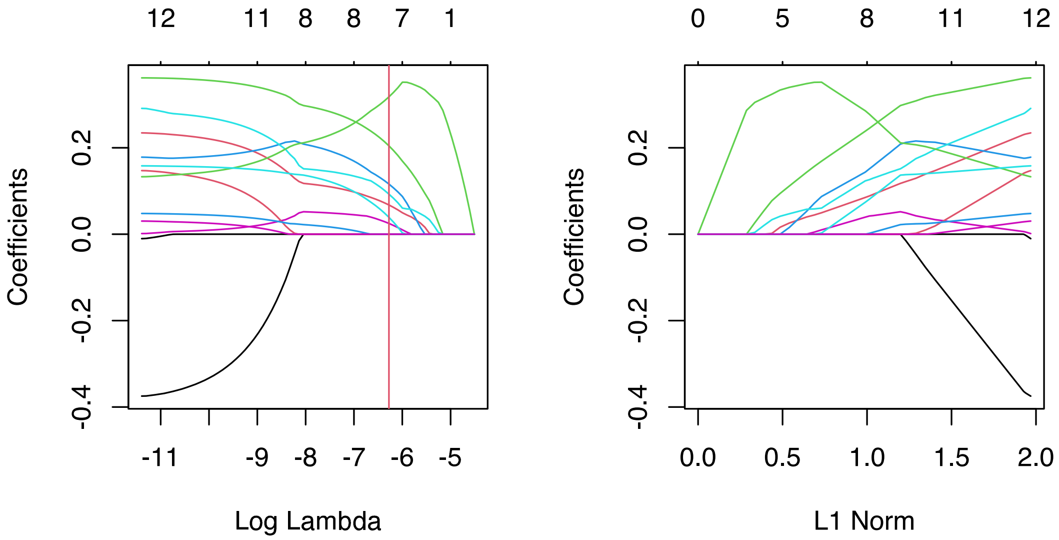
f)

```
## Lasso
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-1

xx <- model.matrix(FoHF ~ ., data = FoHF)
yy <- FoHF$FoHF
cvfit <- cv.glmnet(xx, yy)
plot(cvfit)
```



```
coef(cvfit, s = "lambda.1se")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) 0.0002196824
## (Intercept) .
## RV          .
## CA          0.0676192479
## FIA         0.2044571892
## EMN         0.1160512736
## ED          0.0927171169
## DS          0.0249532900
## MA          .
## LSE         .
## GM          0.3178869630
## EM          .
## CTA         0.0390495342
## SS          .
```

```
fit.lasso <- glmnet(xx, yy)
par(mfrow = c(1, 2))
plot(fit.lasso, label = TRUE, xvar = "lambda")
abline(v = log(cvfit$lambda.1se), col = 2)
plot(fit.lasso, label = TRUE)
```



Cross validation yields a model with 7 predictors. However, these are not identical to the ones chosen by the best BIC fit with 7 predictors. In general, the Lasso is a suitable tool as it can handle multicollinearity of predictor variables and perform variable selection. Both of these aspects are necessary here since the subindices are collinear and we know that the **FoHF** is not invested in all possible subindices. Therefore, the Lasso solution should also be considered next to the one from the variable selection with BIC.