Solution to Series 6

1. Collinearity and variable selection: In a study about infection risk controlling in US hospitals a random sample from 113 hospitals contains the following variables:

id	randomly assigned ID of the hospital
length	average duration of hospital stay (in days)
age	average age of patients (in years)
inf	averaged infection risk (in percent)
cult	number of cultures per non-symptomatic patient x 100
xray	number of X-rays per non-symptomatic patient x 100
beds	number of beds
school	university hospital 1=yes 0=no
region	geographical region 1=NE 2=N 3=S 4=W
pat mittl.	average number of patients a day
nurs mittl.	number of full-employed, trained nurses
serv	percentage of available services from a fixed list of 35 references

Read in the data from: `http://stat.ethz.ch/Teaching/Datasets/senic.dat`. Since some observations span more than a single line, you have to use scan() to read the file into R:

```
senic <-scan("http://stat.ethz.ch/Teaching/Datasets/senic.dat",
  what=list(id=0,length=0,age=0,inf=0,cult=0,xray=0,beds=0,school=0,
  region=0,pat=0,nurs=0,serv=0))}
```

Using `senic <- data.frame(senic)`; `senic <- senic[ ,-1]` you turn the object into a user friendly data frame structure. Turn the variables school and region into so-called factor variables.

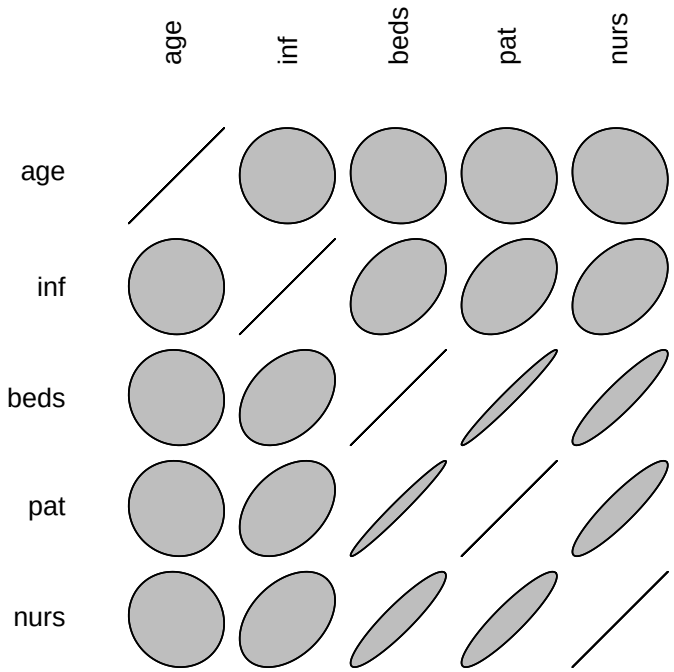
```
> senic <-scan("http://stat.ethz.ch/Teaching/Datasets/senic.dat",
   what=list(id=0,length=0,age=0,inf=0,cult=0,xray=0,
   beds=0,school=0,region=0,pat=0,nurs=0,serv=0))
>   senic <- data.frame(senic)
>   senic <- senic[ ,-1]
>   senic$school <- factor(senic$school)
>   attach(senic)
```

a) Check the correlation between these (not transformed) variables. Which variables are problematic and why? Suggest a combination of variables to improve the situation.
Checking the correlations:

```
> my.senic.00 <- senic[,c("length", "age", "inf", "region", "beds", "pat", "nurs")]
>     cor(my.senic.00[,-c(1,4)])
              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:

```
>     library(ellipse)
>     plotcorr(cor(my.senic.00[,-c(1,4)]))
```

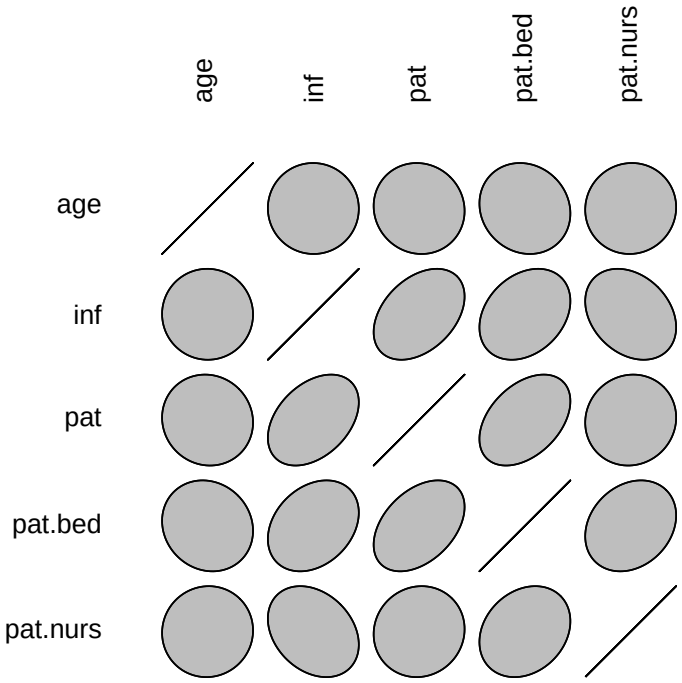


We can see that beds, pat and nurs are strongly correlated. These are all variables mainly describing the size of the hospital. For our goal it would be best to only include pat. However, for modelling workload we can include the coefficient pat/beds and for the human resource situation the coefficient pat/nurs.
New data set:

```
> my.senic.01 <- data.frame(length, age, inf, region, pat,
     pat.bed=pat/beds, pat.nurs=pat/nurs)
>     cor(my.senic.01[,-c(1,4)])
                  age          inf         pat    pat.bed
age       1.000000000 -0.006266807 -0.05477467 -0.1096058
inf      -0.006266807  1.000000000  0.39070521  0.2897338
pat      -0.054774667  0.390705214  1.00000000  0.4151079
pat.bed  -0.109605797  0.289733778  0.41510791  1.0000000
pat.nurs  0.026954588 -0.285984796  0.05659985  0.2289331
            pat.nurs
age       0.02695459
inf      -0.28598480
pat       0.05659985
pat.bed   0.22893307
pat.nurs  1.00000000
```

Checking correlations:

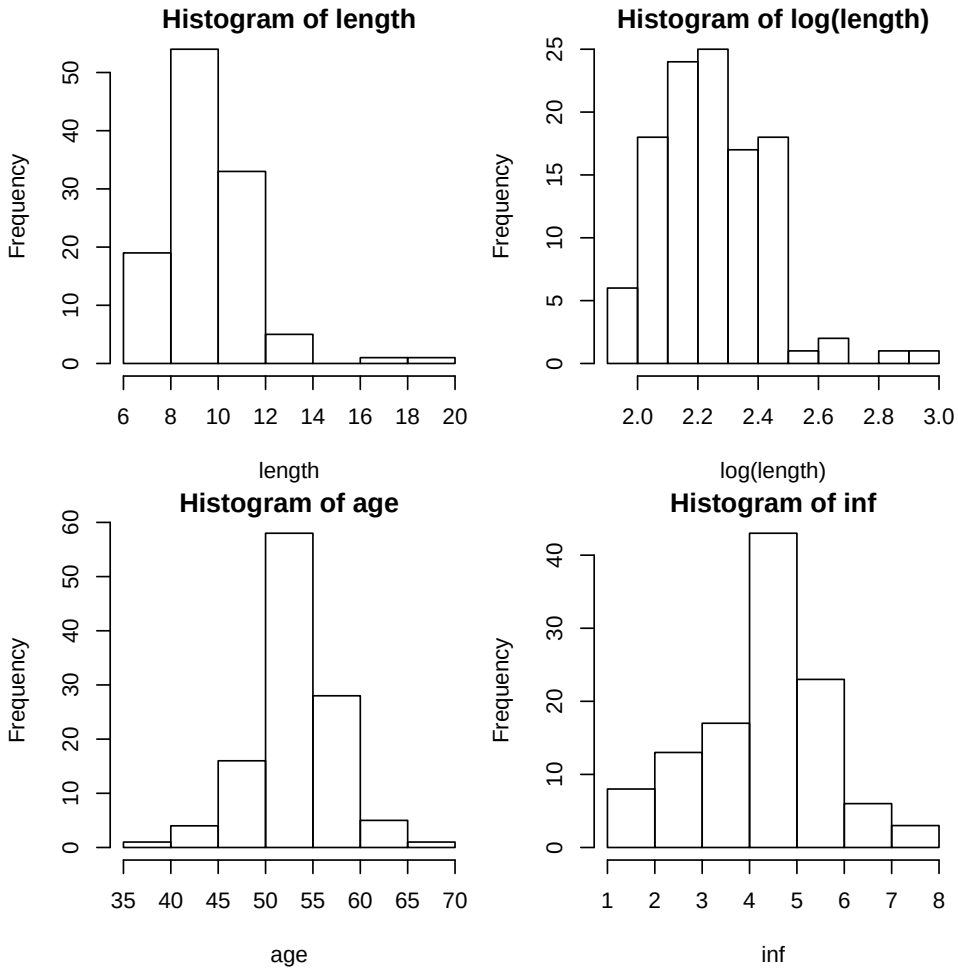
```
>     plotcorr(cor(my.senic.01[,-c(1,4)]))
```



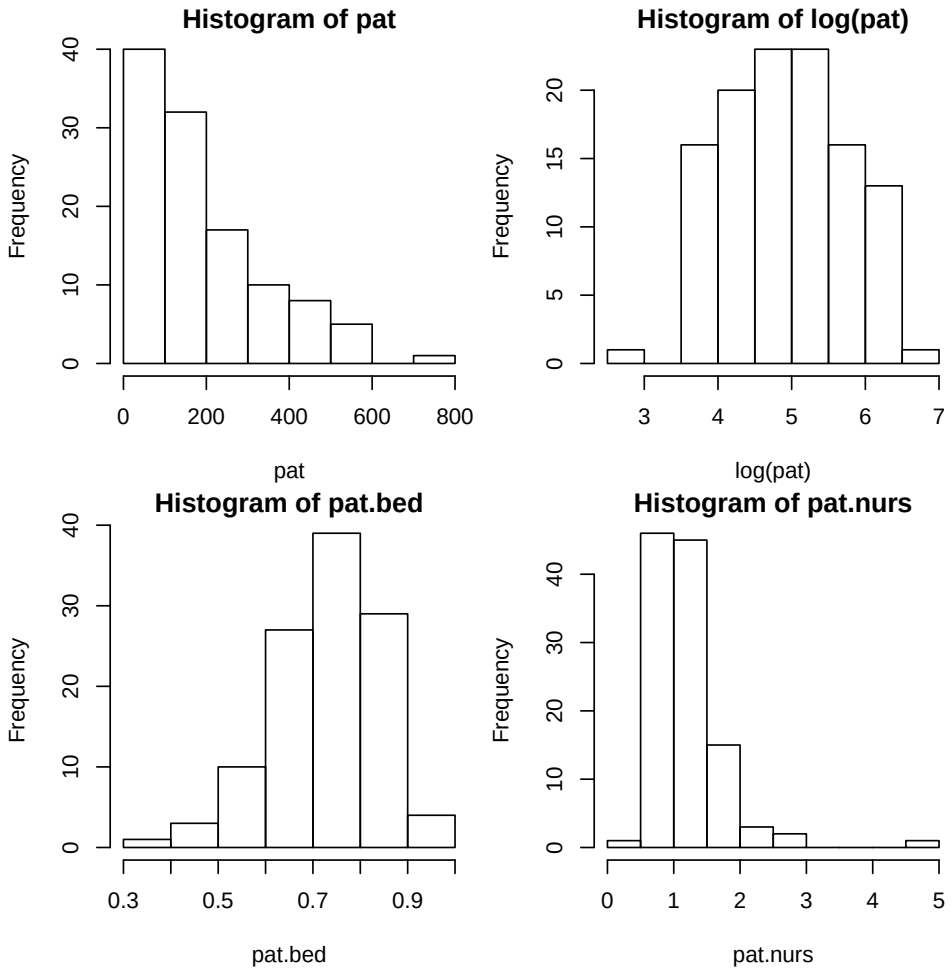
The correlations were strongly reduced. Now we check whether any transformations are necessary.

b) Perform the necessary transformations on the predictors and the response. Will there transformations be necessary for the above combinations as well?

```
> detach(senic)
> attach(my.senic.01)
> par(mfrow=c(2,2))
> hist(length)
> hist(log(length))
> hist(age)
> hist(inf)
```



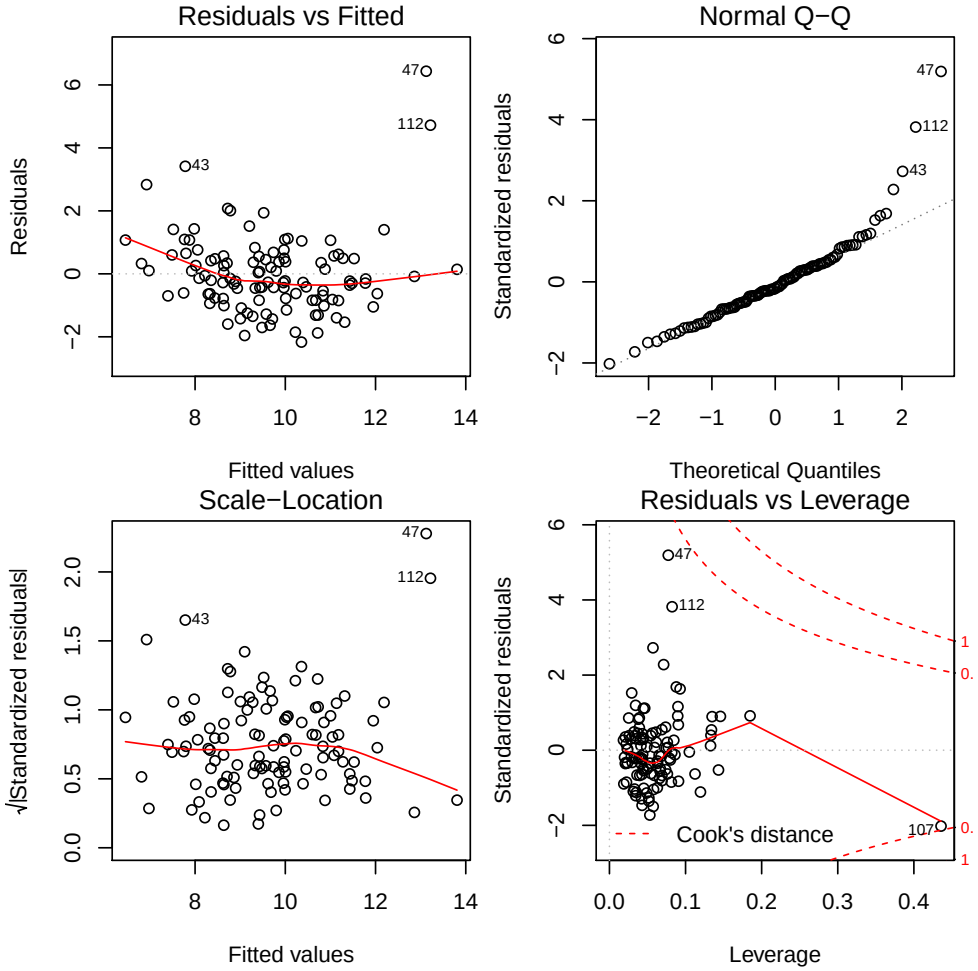

```
> par(mfrow=c(2,2))
> hist(pat)
> hist(log(pat))
> hist(pat.bed)
> hist(pat.nurs)
```



Conclusion: it might be necessary to transform the response which is the average duration of the hospital stay (continuous, not a number) and exhibits a right-skewed pattern. This suggests a log-transformation. Since we cannot be completely sure, we will check both variants.
The same goes for pat. The predictor inf is a percentage - we resign from transforming it because the range of values is rather narrow, the effect would be small.
Adjust model:

```
>     fit00 <- lm(length ~ age + inf + region + log(pat) + pat.bed +
     pat.nurs, data=my.senic.01)
>     summary(fit00)

Call:
lm(formula = length ~ age + inf + region + log(pat) + pat.bed +
    pat.nurs, data = my.senic.01)

Residuals:
    Min      1Q  Median      3Q     Max
-2.1678 -0.7796 -0.2046  0.4949  6.4366

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.36509    1.93496  -0.189  0.85070
age          0.09310    0.02779   3.350  0.00112 **
inf          0.56247    0.11309   4.974 2.55e-06 ***
region      -0.63979    0.12780  -5.006 2.22e-06 ***
log(pat)     0.47864    0.19617   2.440  0.01635 *
pat.bed      1.57915    1.34715   1.172  0.24374
pat.nurs     0.50526    0.25869   1.953  0.05344 .
---
Signif. codes:  0
```

```
>     par(mfrow=c(2,2))
>     plot(fit00)
```



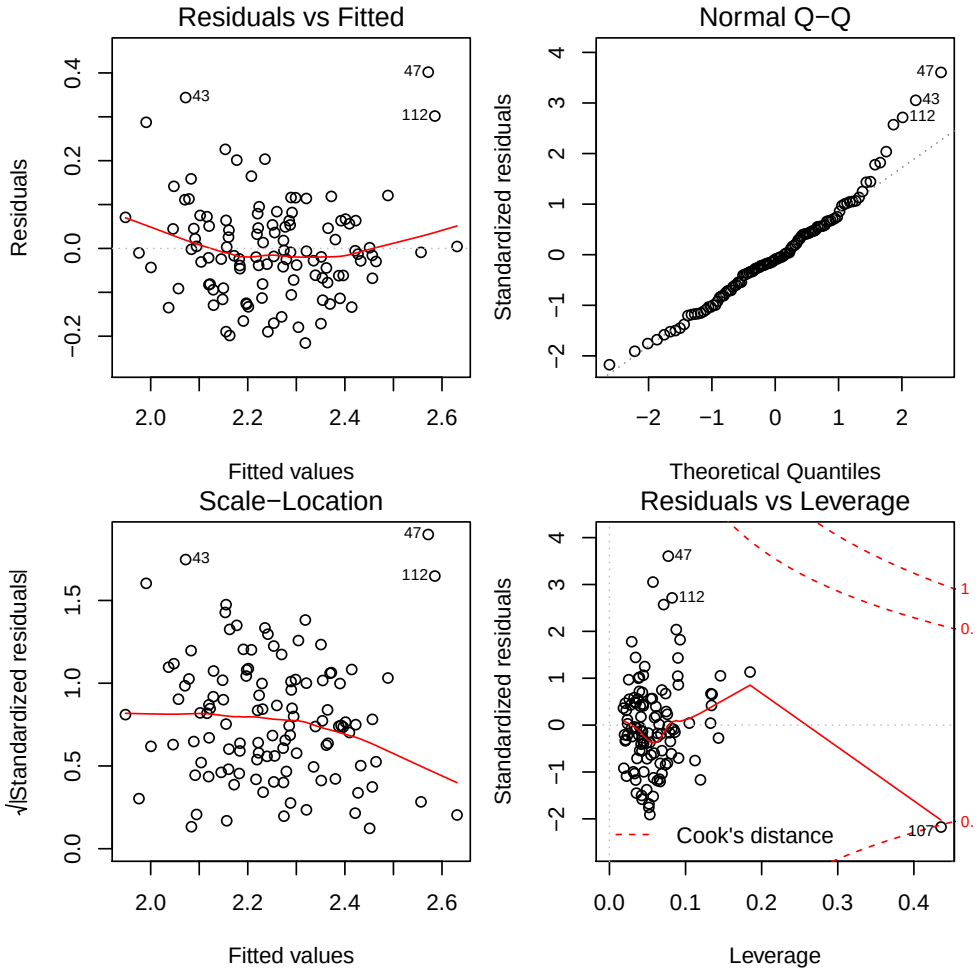
Checking the Tukey-Anscombe plot we can see that the model contains strong structural deficits. These are also visible in the normal Q-Q plot and the scale-location plot. Therefore, we use the log-transformation also on the response.

c) Find a good model! To that end, analyze the residuals, identify possible problematic observations. Decide also upon which variables to use in the model and which to remove.
Adjust model:

```
>     fit01 <- lm(log(length) ~ age + inf + region + log(pat) + pat.bed +
    pat.nurs, data=my.senic.01)
>     summary(fit01)

Call:
lm(formula = log(length) ~ age + inf + region + log(pat) + pat.bed +
    pat.nurs, data = my.senic.01)

Residuals:
     Min       1Q   Median       3Q      Max
-0.21560 -0.07203 -0.01017  0.06320  0.40182

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.347676   0.173938   7.748 5.93e-12 ***
age          0.008116   0.002498   3.249  0.00155 **
inf          0.050698   0.010166   4.987 2.41e-06 ***
region      -0.063755   0.011488  -5.550 2.13e-07 ***
log(pat)     0.050152   0.017634   2.844  0.00535 **
pat.bed      0.152480   0.121098   1.259  0.21074
```

```
pat.nurs     0.034479   0.023254   1.483  0.14111
---
Signif. codes:  0
```

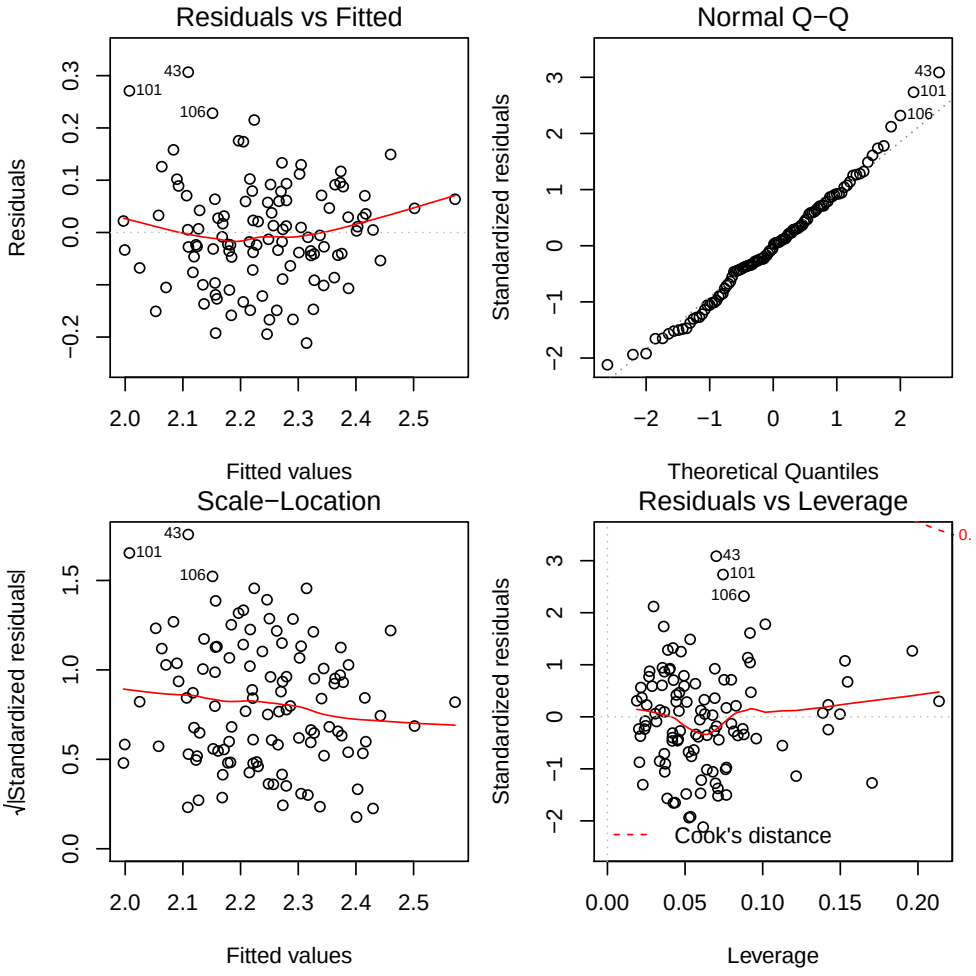
```
>     par(mfrow=c(2,2))
>     plot(fit01)
```



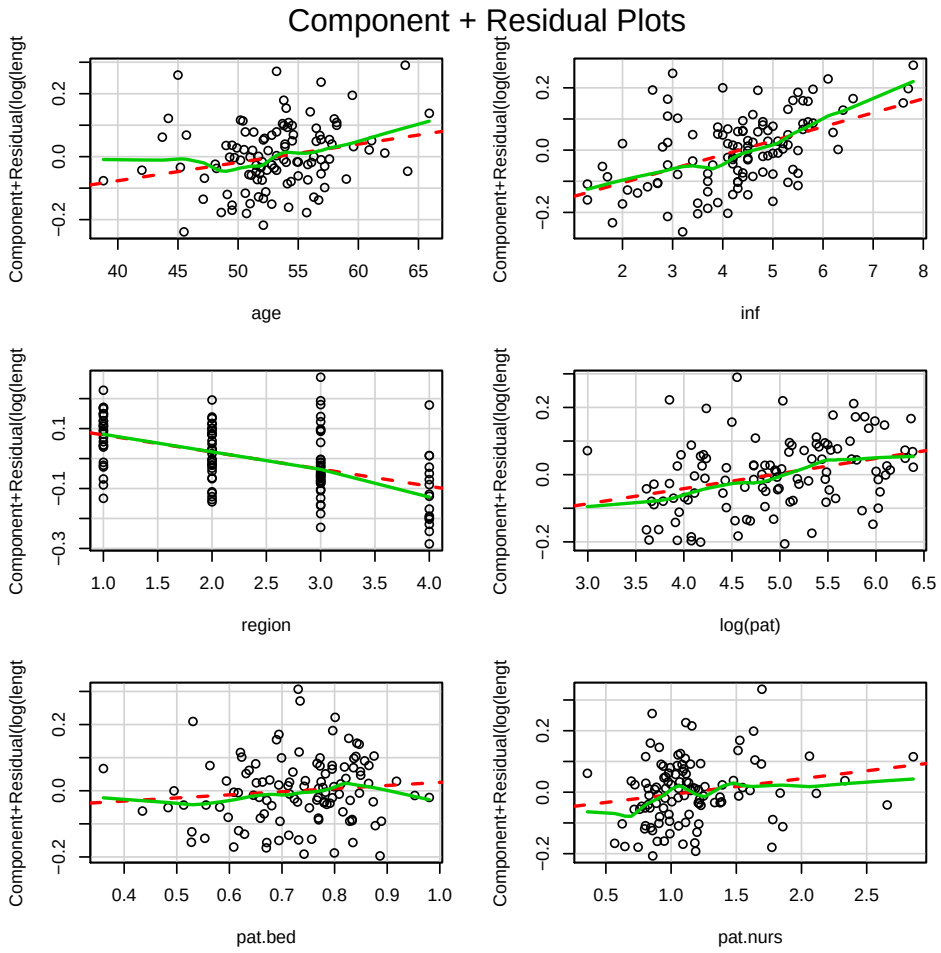
This model still is far from optimal. There are three influential points, i.e., 47, 112 (outliers) and 107 (leverage point). We remove them and check whether we get a better fit.

```
> my.senic.02 <- my.senic.01[-c(47,107,112),]
>     fit02 <- lm(log(length) ~ age + inf + region + log(pat) + pat.bed +
    pat.nurs, data=my.senic.02)
>     summary(fit02)

Call:
lm(formula = log(length) ~ age + inf + region + log(pat) + pat.bed +
    pat.nurs, data = my.senic.02)

Residuals:
      Min        1Q    Median        3Q       Max
-0.211494 -0.061278 -0.001207  0.063051  0.306647

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.523390   0.158350   9.620 5.17e-16 ***
age          0.005812   0.002256   2.577  0.01139 *
inf          0.044946   0.009148   4.913 3.38e-06 ***
region      -0.057023   0.010271  -5.552 2.21e-07 ***
log(pat)     0.044893   0.015786   2.844  0.00538 **
pat.bed      0.094130   0.108183   0.870  0.38627
```

```
pat.nurs     0.051482   0.027029   1.905  0.05960 .
---
Signif. codes:  0
> anova(fit02)
Analysis of Variance Table

Response: log(length)
           Df  Sum Sq Mean Sq F value    Pr(>F)
age         1 0.02926 0.02926  2.7576 0.0998372 .
inf         1 0.70817 0.70817 66.7319 8.305e-13 ***
region      1 0.46526 0.46526 43.8420 1.645e-09 ***
log(pat)    1 0.17230 0.17230 16.2360 0.0001073 ***
pat.bed     1 0.02059 0.02059  1.9406 0.1666002
pat.nurs    1 0.03850 0.03850  3.6280 0.0596046 .
Residuals 103 1.09305 0.01061
---
Signif. codes:  0
> par(mfrow=c(2,2))
> plot(fit02)
```




```
> library(car)
> crPlots(fit02)
```



The fit has improved but is still not perfect. Unfortunately we lack the means for further improvement.

The analysis of the partial residual plots shows a nonlinear influence of the variable age. Until the age of 55 the duration of the hospital stay seems not to increase with age, afterwards it raises markedly.

From the summary we can see that not all predictors are significant. The task of reducing the model to the necessary predictors is subject of part d), e) and f). The corresponding solution will be given then.

d) Perform a backward elimination using the AIC criterion. Use the function `step()`. Check the final model with the usual diagnostic plots.

Backward elimination:

```
> fit.back <- lm(log(length) ~ age + inf + region + log(pat) + pat.bed + pat.nurs, data=my.sen
> fit.B <- step(fit.back, direction="backward")
Start:  AIC=-493.27
log(length) ~ age + inf + region + log(pat) + pat.bed + pat.nurs

           Df Sum of Sq    RSS     AIC
- pat.bed   1   0.00803 1.1011 -494.46
<none>                  1.0931 -493.27
- pat.nurs  1   0.03850 1.1316 -491.46
- age       1   0.07046 1.1635 -488.39
- log(pat)  1   0.08583 1.1789 -486.95
- inf       1   0.25619 1.3492 -472.10
- region    1   0.32710 1.4202 -466.47

Step:  AIC=-494.46
log(length) ~ age + inf + region + log(pat) + pat.nurs
```

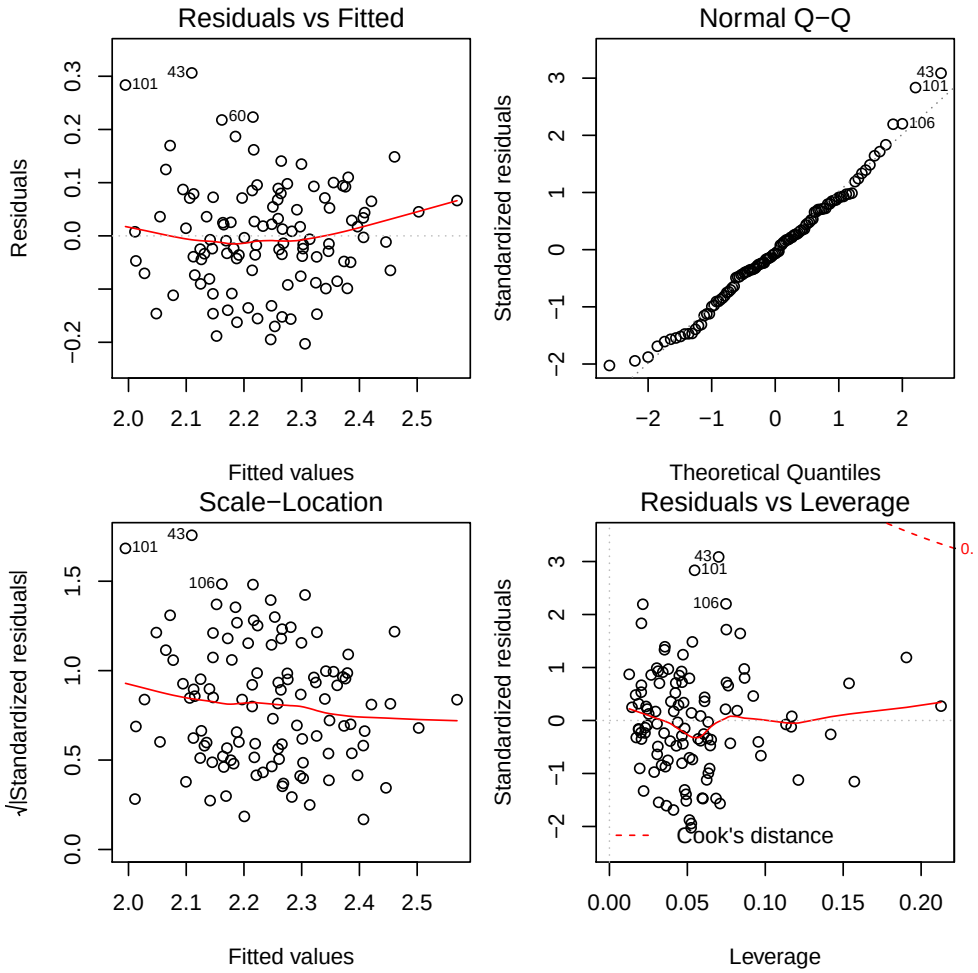
```
           Df Sum of Sq    RSS     AIC
<none>                  1.1011 -494.46
- pat.nurs  1   0.05106 1.1521 -491.47
- age       1   0.06654 1.1676 -490.01
- log(pat)  1   0.12830 1.2294 -484.34
- inf       1   0.27114 1.3722 -472.25
- region    1   0.36421 1.4653 -465.03
```

```
>     summary(fit.B)
```

```
Call:
lm(formula = log(length) ~ age + inf + region + log(pat) + pat.nurs,
    data = my.senic.02)

Residuals:
      Min        1Q    Median        3Q       Max
-0.202879 -0.064849 -0.006766  0.067493  0.306311

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.568858   0.149304  10.508  < 2e-16 ***
age          0.005622   0.002242   2.507 0.013723 *
inf          0.045903   0.009071   5.061 1.81e-06 ***
region      -0.058870   0.010037  -5.865 5.37e-08 ***
log(pat)     0.050358   0.014466   3.481 0.000731 ***
pat.nurs     0.057388   0.026132   2.196 0.030307 *
---
Signif. codes:  0
```

The backward elimination only removes the variable pat.bed from the model.

```
> par(mfrow=c(2,2))
>     plot(fit.B)
```



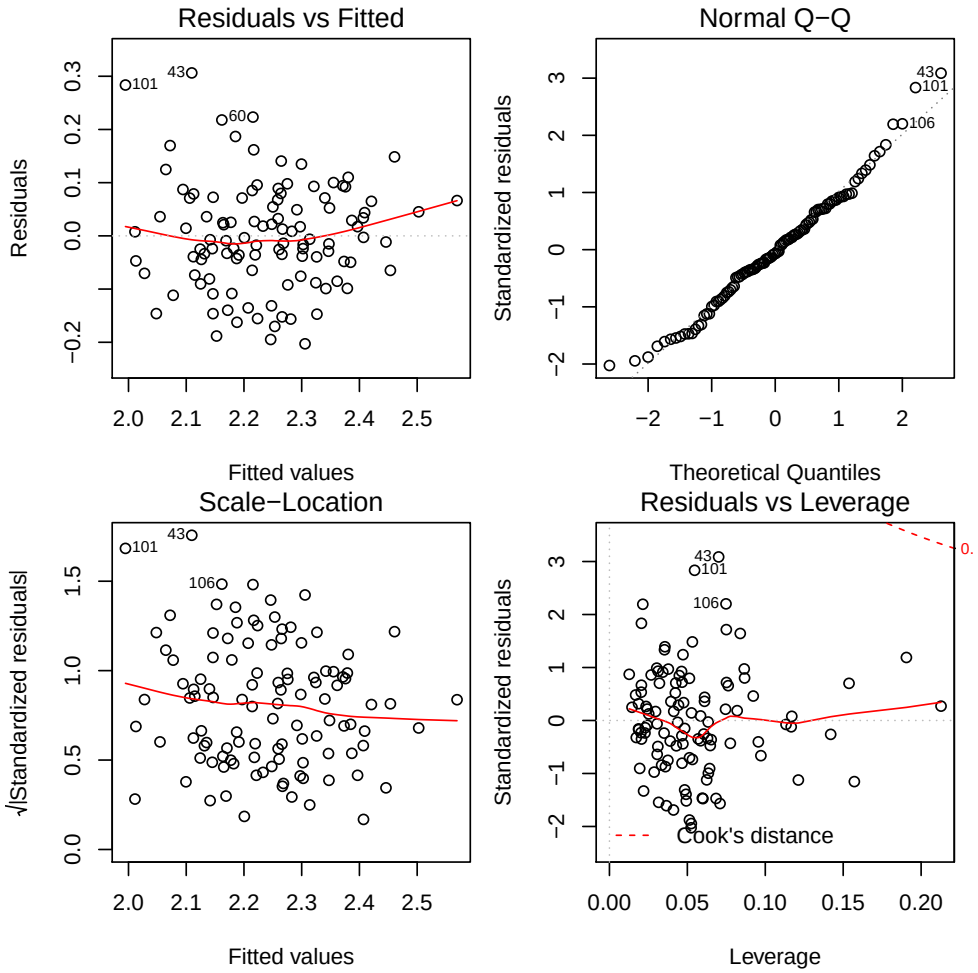
e) Now perform a forward selection using the AIC criterion. Thus, start with the empty model, i.e.: `fit.for <- lm(log(length) ~ 1, data=...)`
Use the same function as before. Check also the diagnostic plots and comment on the differences to **d)**.
Forward selection:

```
> fit.for <- lm(log(length) ~ 1, data=my.senic.02)
>     scp   <- list(lower=~1, upper=~age + inf + region + log(pat) + pat.bed + pat.nurs)
>     fit.F <- step(fit.for, scope=scp, direction="forward")
Start:  AIC=-413.07
log(length) ~ 1

            Df Sum of Sq    RSS     AIC
+ inf        1   0.69419 1.8329 -446.40
+ region     1   0.63749 1.8896 -443.05
+ log(pat)   1   0.62198 1.9051 -442.15
+ pat.bed    1   0.42539 2.1017 -431.35
<none>                   2.5271 -413.07
+ age        1   0.02926 2.4979 -412.35
+ pat.nurs   1   0.02725 2.4999 -412.27

Step:  AIC=-446.4
log(length) ~ inf

            Df Sum of Sq    RSS     AIC
+ region     1   0.46482 1.3681 -476.57
+ log(pat)   1   0.21093 1.6220 -457.85
+ pat.bed    1   0.19266 1.6403 -456.62
```

```
+ pat.nurs  1   0.14782 1.6851 -453.65
+ age       1   0.04324 1.7897 -447.03
<none>                  1.8329 -446.40

Step:  AIC=-476.57
log(length) ~ inf + region

           Df Sum of Sq    RSS     AIC
+ log(pat)  1  0.141710 1.2264 -486.60
+ pat.nurs  1  0.098574 1.2695 -482.80
+ pat.bed   1  0.076908 1.2912 -480.94
+ age       1  0.043682 1.3244 -478.14
<none>                  1.3681 -476.57

Step:  AIC=-486.6
log(length) ~ inf + region + log(pat)

           Df Sum of Sq    RSS     AIC
+ age       1  0.074270 1.1521 -491.47
+ pat.nurs  1  0.058787 1.1676 -490.01
<none>                  1.2264 -486.60
+ pat.bed   1  0.014786 1.2116 -485.94

Step:  AIC=-491.47
log(length) ~ inf + region + log(pat) + age

           Df Sum of Sq    RSS     AIC
+ pat.nurs  1  0.051061 1.1011 -494.46
<none>                  1.1521 -491.47
+ pat.bed   1  0.020594 1.1316 -491.46

Step:  AIC=-494.46
log(length) ~ inf + region + log(pat) + age + pat.nurs

          Df Sum of Sq    RSS     AIC
<none>                 1.1011 -494.46
+ pat.bed  1 0.0080341 1.0931 -493.27
> summary(fit.F)
Call:
lm(formula = log(length) ~ inf + region + log(pat) + age + pat.nurs,
    data = my.senic.02)

Residuals:
      Min        1Q    Median        3Q       Max
-0.202879 -0.064849 -0.006766  0.067493  0.306311

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.568858   0.149304  10.508  < 2e-16 ***
inf          0.045903   0.009071   5.061 1.81e-06 ***
region      -0.058870   0.010037  -5.865 5.37e-08 ***
log(pat)     0.050358   0.014466   3.481 0.000731 ***
age          0.005622   0.002242   2.507 0.013723 *
pat.nurs     0.057388   0.026132   2.196 0.030307 *
---
Signif. codes:  0
```

We get the same result as before.

```
> par(mfrow=c(2,2))
> plot(fit.F)
```



f) Optional: Perform a stepwise selection. Start with the full model as well as with empty model and compare the results. Check the help file of `step()` on how to perform a stepwise selection. The stepwise selection gives the same result whether we are using the full model or the empty model as starting point:

```
>     step(fit.back, direction="both")
Start:  AIC=-493.27
log(length) ~ age + inf + region + log(pat) + pat.bed + pat.nurs

           Df Sum of Sq    RSS     AIC
- pat.bed   1   0.00803 1.1011 -494.46
<none>                 1.0931 -493.27
- pat.nurs  1   0.03850 1.1316 -491.46
- age       1   0.07046 1.1635 -488.39
- log(pat)  1   0.08583 1.1789 -486.95
- inf       1   0.25619 1.3492 -472.10
- region    1   0.32710 1.4202 -466.47

Step:  AIC=-494.46
log(length) ~ age + inf + region + log(pat) + pat.nurs

           Df Sum of Sq    RSS     AIC
<none>                 1.1011 -494.46
+ pat.bed   1   0.00803 1.0931 -493.27
- pat.nurs  1   0.05106 1.1521 -491.47
- age       1   0.06654 1.1676 -490.01
- log(pat)  1   0.12830 1.2294 -484.34
- inf       1   0.27114 1.3722 -472.25
- region    1   0.36421 1.4653 -465.03
```

```
Call:
lm(formula = log(length) ~ age + inf + region + log(pat) + pat.nurs,
    data = my.senic.02)

Coefficients:
(Intercept)          age          inf       region
   1.568858     0.005622     0.045903    -0.058870
   log(pat)     pat.nurs
   0.050358     0.057388
>     step(fit.for, scope=scp, direction="both")
Start:  AIC=-413.07
log(length) ~ 1

           Df Sum of Sq    RSS     AIC
+ inf       1   0.69419 1.8329 -446.40
+ region    1   0.63749 1.8896 -443.05
+ log(pat)  1   0.62198 1.9051 -442.15
+ pat.bed   1   0.42539 2.1017 -431.35
<none>                  2.5271 -413.07
+ age       1   0.02926 2.4979 -412.35
+ pat.nurs  1   0.02725 2.4999 -412.27

Step:  AIC=-446.4
log(length) ~ inf

           Df Sum of Sq    RSS     AIC
+ region    1   0.46482 1.3681 -476.57
+ log(pat)  1   0.21093 1.6220 -457.85
+ pat.bed   1   0.19266 1.6403 -456.62
+ pat.nurs  1   0.14782 1.6851 -453.65
+ age       1   0.04324 1.7897 -447.03
<none>                  1.8329 -446.40
- inf       1   0.69419 2.5271 -413.07

Step:  AIC=-476.57
log(length) ~ inf + region

           Df Sum of Sq    RSS     AIC
+ log(pat)  1   0.14171 1.2264 -486.60
+ pat.nurs  1   0.09857 1.2695 -482.80
+ pat.bed   1   0.07691 1.2912 -480.94
+ age       1   0.04368 1.3244 -478.14
<none>                  1.3681 -476.57
- region    1   0.46482 1.8329 -446.40
- inf       1   0.52151 1.8896 -443.05

Step:  AIC=-486.6
log(length) ~ inf + region + log(pat)

           Df Sum of Sq    RSS     AIC
+ age       1   0.07427 1.1521 -491.47
+ pat.nurs  1   0.05879 1.1676 -490.01
<none>                  1.2264 -486.60
+ pat.bed   1   0.01479 1.2116 -485.94
- log(pat)  1   0.14171 1.3681 -476.57
- inf       1   0.23225 1.4587 -469.53
- region    1   0.39560 1.6220 -457.85
```

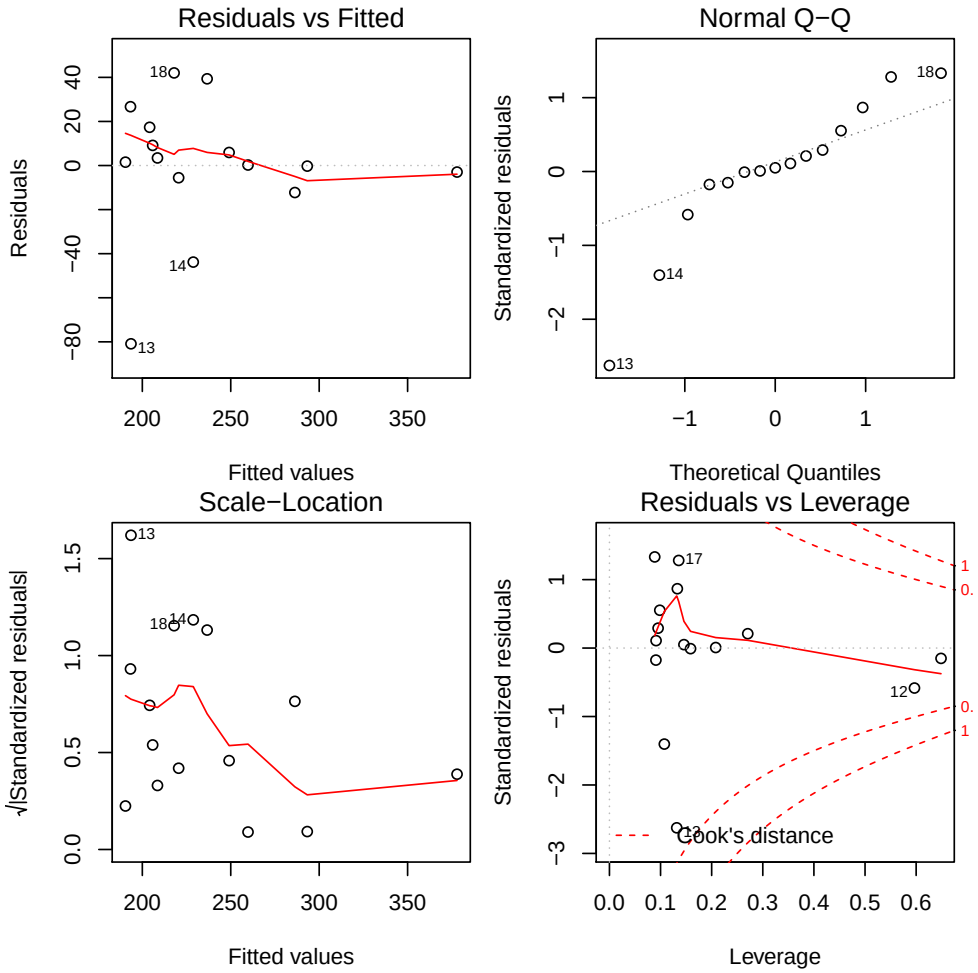
```
Step:  AIC=-491.47
log(length) ~ inf + region + log(pat) + age

           Df Sum of Sq    RSS     AIC
+ pat.nurs  1   0.05106 1.1011 -494.46
<none>                 1.1521 -491.47
+ pat.bed   1   0.02059 1.1316 -491.46
- age       1   0.07427 1.2264 -486.60
- log(pat)  1   0.17230 1.3244 -478.14
- inf       1   0.22372 1.3759 -473.95
- region    1   0.38905 1.5412 -461.47

Step:  AIC=-494.46
log(length) ~ inf + region + log(pat) + age + pat.nurs

           Df Sum of Sq    RSS     AIC
<none>                 1.1011 -494.46
+ pat.bed   1   0.00803 1.0931 -493.27
- pat.nurs  1   0.05106 1.1521 -491.47
- age       1   0.06654 1.1676 -490.01
- log(pat)  1   0.12830 1.2294 -484.34
- inf       1   0.27114 1.3722 -472.25
- region    1   0.36421 1.4653 -465.03

Call:
lm(formula = log(length) ~ inf + region + log(pat) + age + pat.nurs,
    data = my.senic.02)

Coefficients:
(Intercept)          inf       region     log(pat)
   1.568858     0.045903    -0.058870     0.050358
        age     pat.nurs
   0.005622     0.057388
```

2. Cross validation: The goal of this exercise is to make you acquainted with the cross-validation technique. Use the data set `data(houseprices)` from the package `library(DAAG)`.

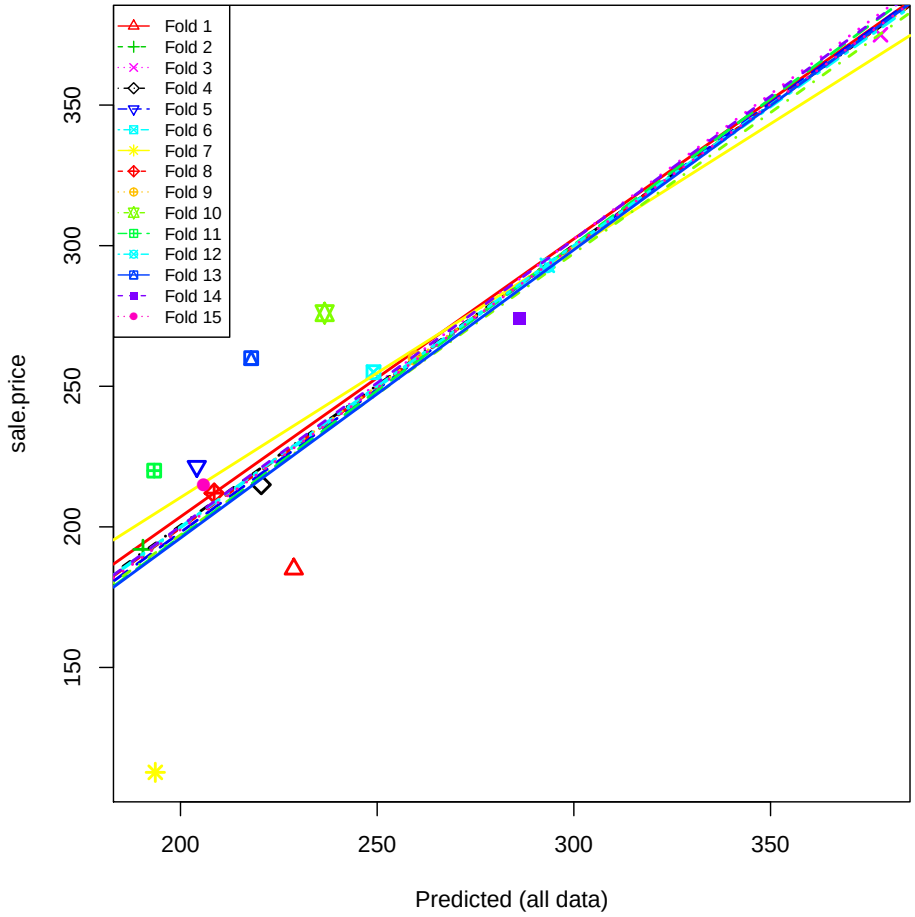
```
> head(houseprices)

   area bedrooms sale.price
9   694        4      192.0
10  905        4      215.0
11  802        4      215.0
12 1366        4      274.0
13  716        4      112.7
14  963        4      185.0
```

a) Perform a leave-one-out cross validation for the model containing both predictors as main effects: `sale.price` $\sim$ `area + bedrooms`
Is there a better model to predict the sale price? What other models are possible anyway? R hint: Use the R-function `CVlm()` from `library(DAAG)`.
Main effects model including cross validation:

```
> fit00 <- lm(sale.price ~ area + bedrooms, data=houseprices)
> summary(fit00)

Call:
lm(formula = sale.price ~ area + bedrooms, data = houseprices)

Residuals:
    Min      1Q  Median      3Q     Max
-80.897  -4.247   1.539  13.249  42.027

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -141.76132   67.87204  -2.089  0.05872 .
area           0.14255    0.04697   3.035  0.01038 *
bedrooms      58.32375   14.75962   3.952  0.00192 **
---
Signif. codes:  0

> par(mfrow=c(2,2))
> plot(fit00)
```

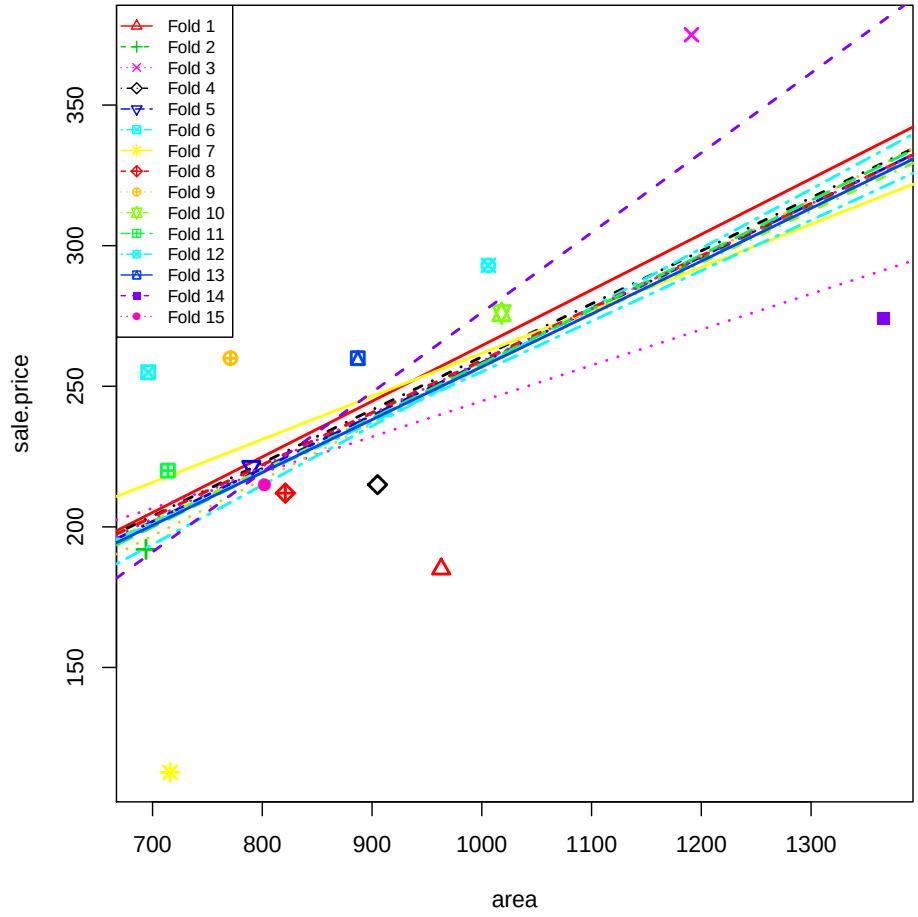



```
pdf
  2
> CVlm(houseprices, sale.price ~ area + bedrooms, m=15)
> OverallMS
Overall ms
      1188
```

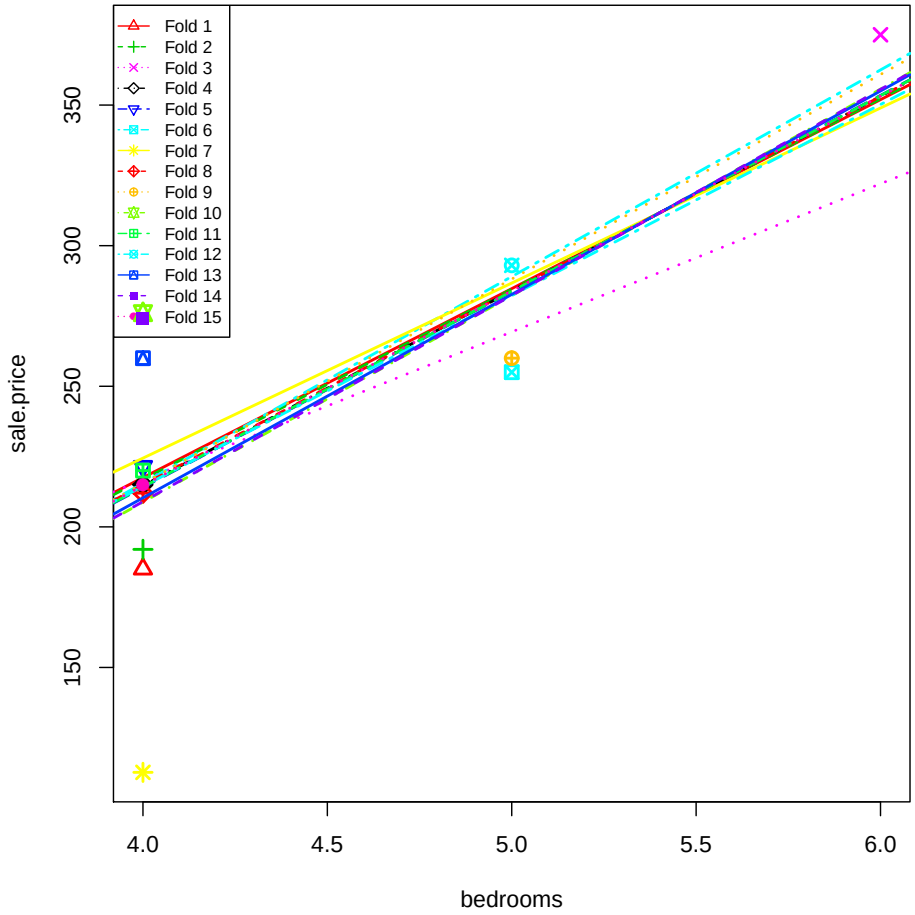


Now we can compare this model with the two other models containing each only one predictor:

```
pdf
  2
> CVlm(houseprices, sale.price ~ area , m=15)
> OverallMS
Overall ms
      3247
```

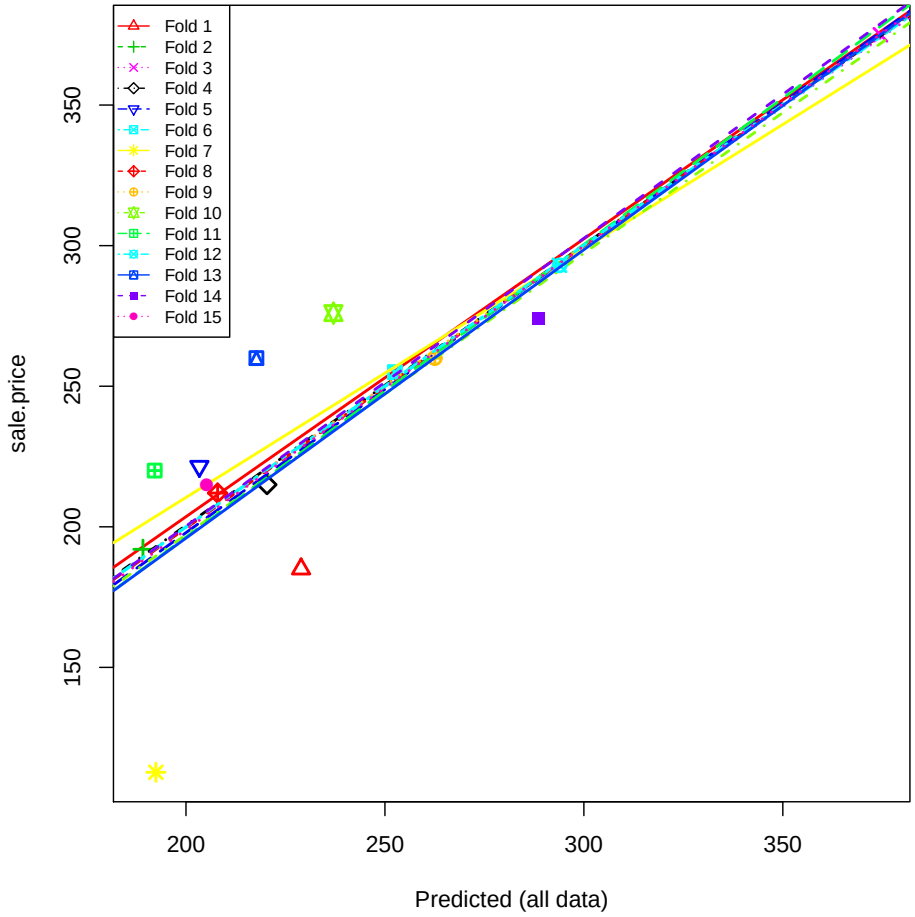



```
pdf
  2
> CVlm(houseprices, sale.price ~  bedrooms, m=15)
> OverallMS
Overall ms
      2023
```



Both single-predictor models are considerably worse: The mean squared prediction error raises from 1188 to 2023 resp. 3247. Next we could try the model including an interaction:

```
pdf
  2
> CVlm(houseprices, sale.price ~  area * bedrooms, m=15)
> OverallMS
Overall ms
      1336
```



The mean squared prediction error is 1336. Therefore, the main-effects model is the "best" model for this prediction.

b) Optional exercise for advanced users: Instead of using the function `CVlm(data, formula, fold.number, ...)` you could also perform the cross validation "by hand" using a `for`-loop. "By hand" cross validation:

```
> oos.pred <- c()
> dat      <- houseprices
> for (i in 1:nrow(dat))
   {
     ## Reduce the data-set: exclude the i-th observation
     dat.red <- dat[-i,]

     ## Fit a regession on the smaller data-set
     fit.red <- lm(sale.price ~ area + bedrooms, data=dat.red)

     ## Predict the i-th observation
     oos.pred[i] <- predict(fit.red, newdata=dat[i,])
   }
> ## compute the mean square prediction error
> mean((houseprices$sale.price-oos.pred)^2)
[1] 1188
```

We get 1188, as with the function `CVlm` from above.