Solution to Series 6

1.
```
> senic.00 <-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.00 <- data.frame(senic.00)
>  senic.00 <- senic.00[ ,-1]
>  senic.00$school <- factor(senic.00$school,levels=c(1,2),labels=c("yes","no"))
>  senic.00$region <- factor(senic.00$region,levels=c(1,2,3,4),labels=c("NE","N","S","W"))
```

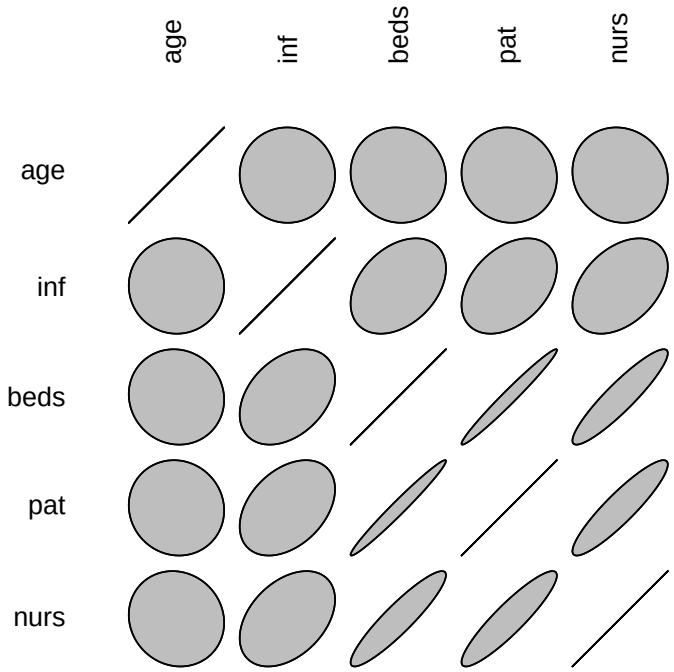
a) We check the correlations between the continuous predictors:

```
> senic.01 <- senic.00[,c("length", "age", "inf", "region", "beds", "pat", "nurs")]
> cor(senic.01[,-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 of the correlations:

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



We see that beds, pat and nurs are strongly correlated. We expected this because they all can be seen as measures of the size of a hospital. We will leave the variable pat unmodified because it is definitely a key factor to take into account when length is the response variable and change the others to solve the high-correlation problem without having to take them out of the model. For this, we will substitute beds by pat/beds and nurs by pat/nurs.
Before combining the variables, we check if beds and nurs contain zeroes:

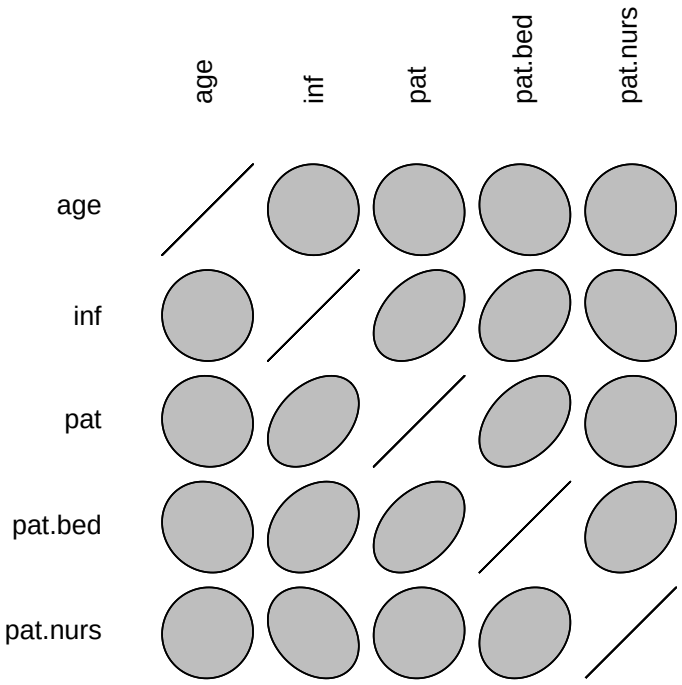
```
> any(senic.01$beds == 0)
[1] FALSE
> any(senic.01$nurs == 0)
[1] FALSE
```

Now we combine the variables and check the correlations again.

```
> senic.02 <- data.frame(length=senic.01$length, age=senic.01$age, inf=senic.01$inf,
 region=senic.01$region, pat=senic.01$pat, pat.bed=senic.01$pat/senic.01$beds,
 pat.nurs=senic.01$pat/senic.00$nurs)
> cor(senic.02[,-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
```

Graphical illustration of the correlations after modifying some variables:

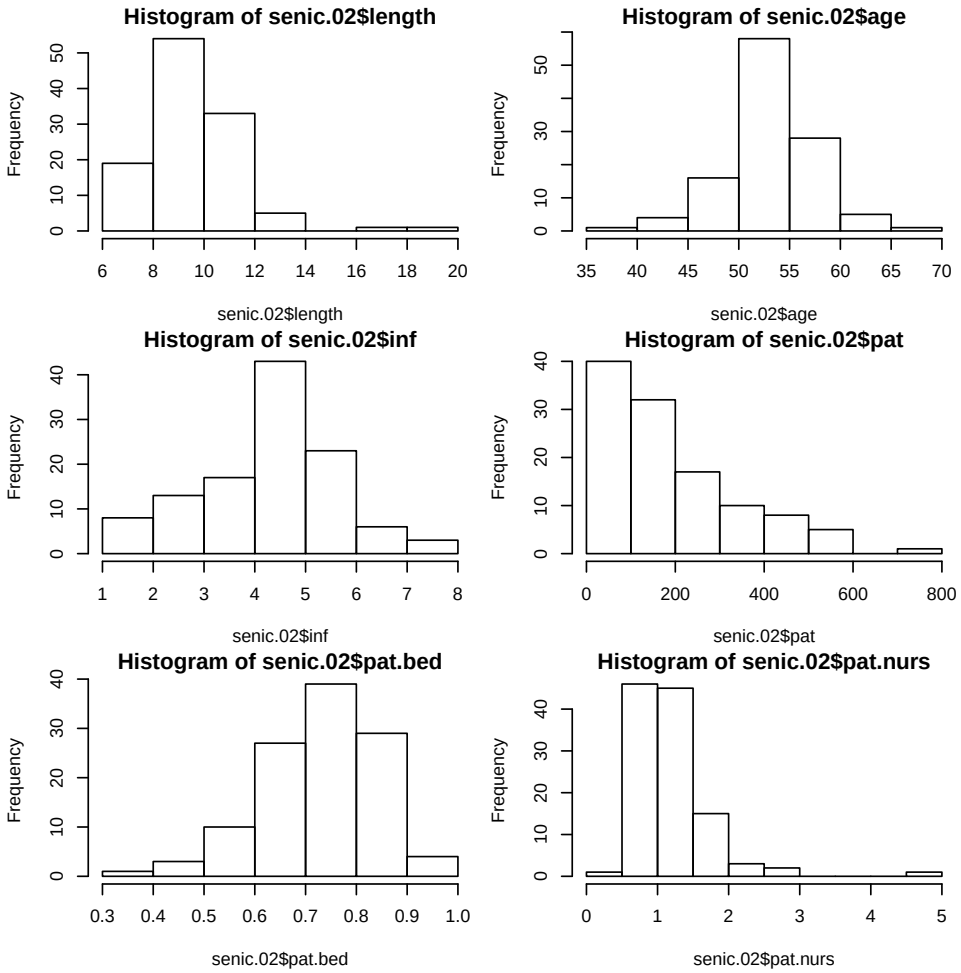
```
> plotcorr(cor(senic.02[,-c(1,4)]))
```



The correlations were strongly reduced and we still have some information about the variables `beds` and `nurs`.

b) First, we take a look at the histogram of the predictors before doing transformations:

```
> par(mfrow=c(3,2))
> hist(senic.02$length)
> hist(senic.02$age)
> hist(senic.02$inf)
> hist(senic.02$pat)
> hist(senic.02$pat.bed)
> hist(senic.02$pat.nurs)
```



The variables `length`, `pat`, `inf` (percentage) and `pat.nurs` need to be transformed. Moreover, we see that `pat.bed` is slightly left skewed. In this case, one would try to square or cube the variable to improve the situation, however, for the purpose of this question, we will not do it here and leave this as an exercise.

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

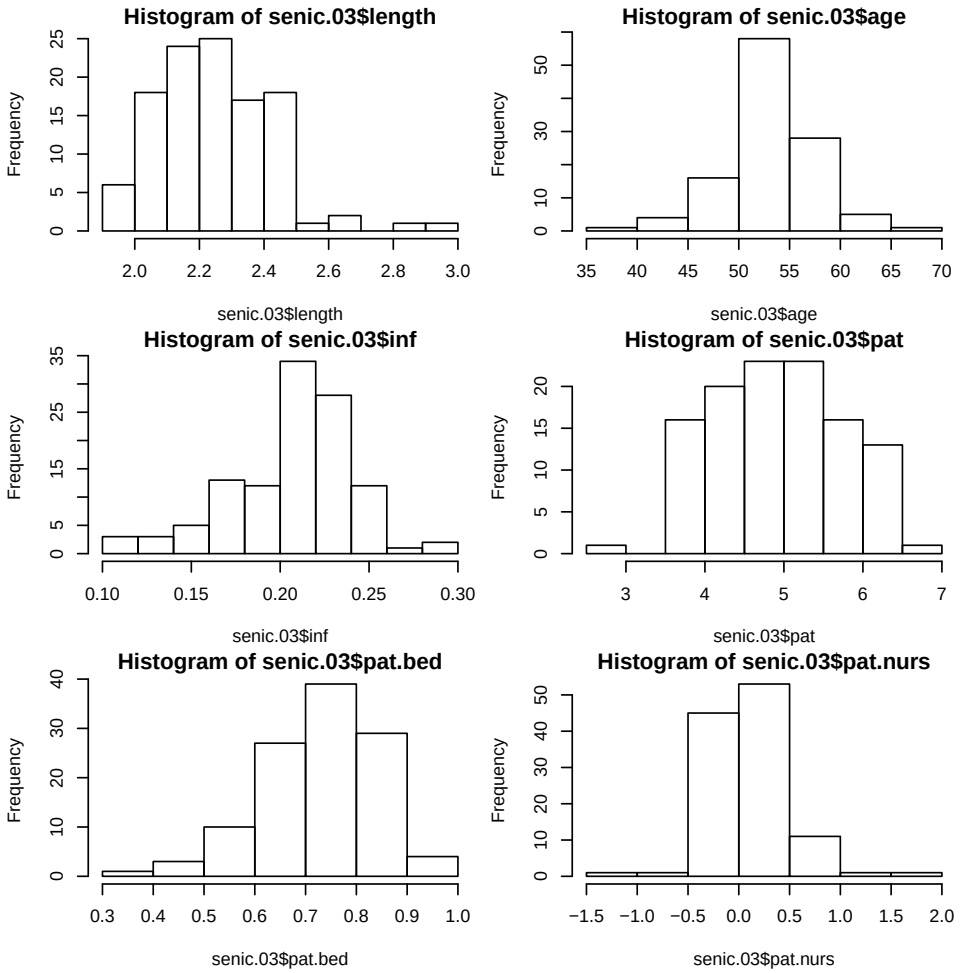
```
> any(senic.02$length == 0)
[1] FALSE
> any(senic.02$pat == 0)
[1] FALSE
```

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

```
> senic.03 <- senic.02
> senic.03$length <- log(senic.02$length)
> senic.03$inf <- asin(sqrt(senic.02$inf/100))
> senic.03$pat <- log(senic.02$pat)
> senic.03$pat.nurs <- log(senic.02$pat.nurs)
```

We look at the histograms again after applying the necessary transformations.

```
>     par(mfrow=c(3,2))
>     hist(senic.03$length)
>     hist(senic.03$age)
>     hist(senic.03$inf)
>     hist(senic.03$pat)
>     hist(senic.03$pat.bed)
>     hist(senic.03$pat.nurs)
```



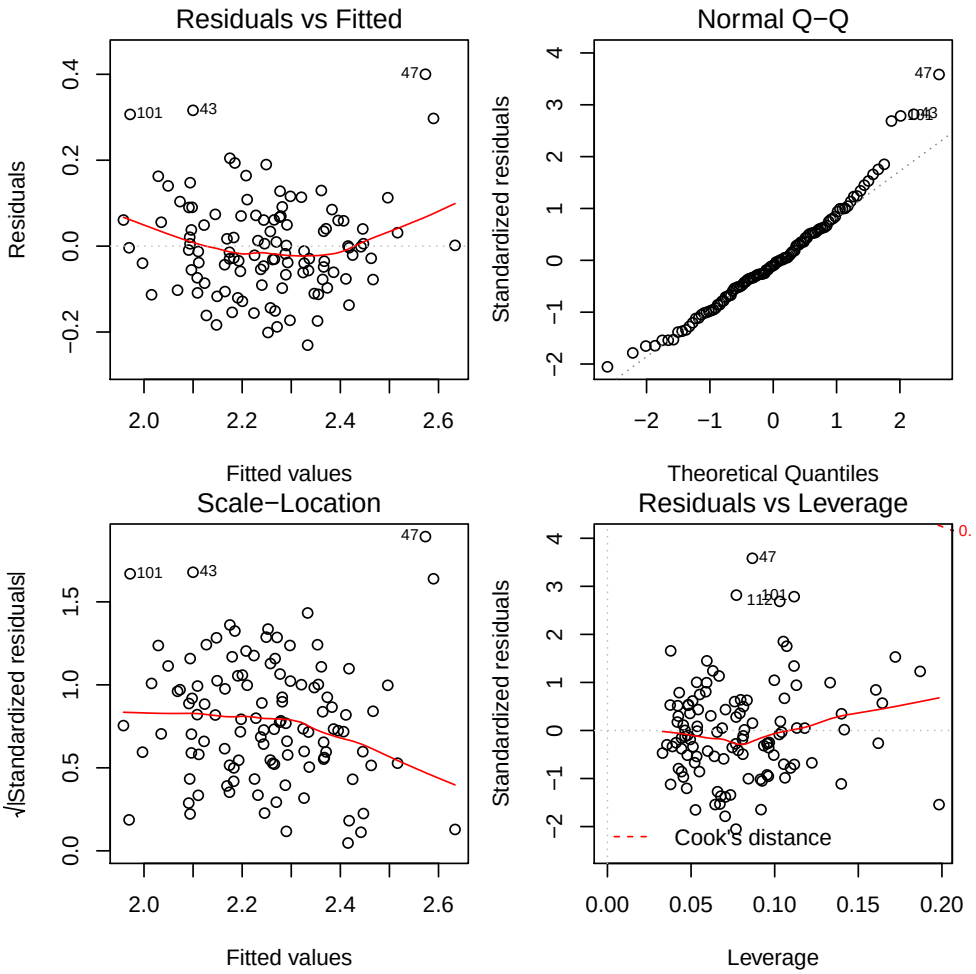
We see that the transformations improved the histograms.

c) We fit a linear regression:

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

Call:
lm(formula = length ~ age + inf + region + pat + pat.bed + pat.nurs,
    data = senic.03)

Residuals:
     Min       1Q   Median       3Q      Max
-0.23050 -0.07370 -0.01151  0.06079  0.40012

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.174393   0.181654   6.465 3.35e-09 ***
age          0.008011   0.002573   3.114  0.00238 **
inf          2.053905   0.413453   4.968 2.67e-06 ***
regionN     -0.073815   0.031465  -2.346  0.02088 *
regionS     -0.122286   0.030768  -3.975  0.00013 ***
regionW     -0.202395   0.040295  -5.023 2.12e-06 ***
pat          0.046294   0.018177   2.547  0.01233 *
pat.bed      0.102920   0.125846   0.818  0.41533
pat.nurs     0.082523   0.038341   2.152  0.03368 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1168 on 104 degrees of freedom
```

```
Multiple R-squared:   0.6,        Adjusted R-squared: 0.5692
F-statistic:  19.5 on 8 and 104 DF,  p-value: < 2.2e-16
```

```
>     par(mfrow=c(2,2))
>     plot(fit.03)
```

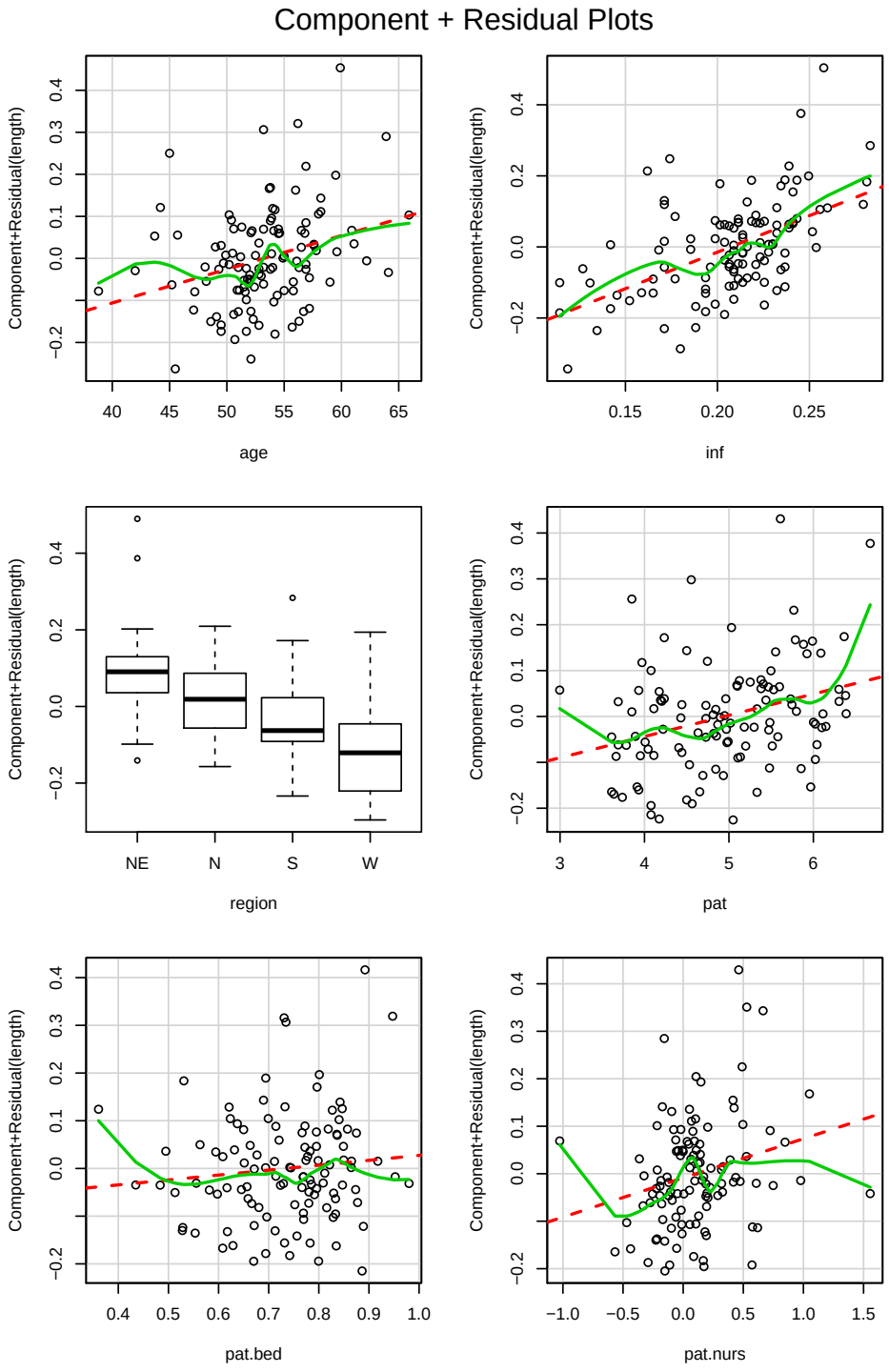


From the summary we see that `pat.beds` is not statistically significant and a variable selection is necessary (see next question).

From the model diagnostics plots we note that there are four outliers ,i.e., observations 43, 47, 101, and 112. However, since their Cook's distance is below 0.5, they don't significantly influence our fit and we proceed with our analysis. The assumptions of linearity, normality, non-constant variance and uncorrelated errors seem to be satisfied.

Now we visualise our model with partial residual plots.

```
>     library(car)
>     crPlots(fit.03)
>
```



As it can be seen in the plots, the predictor `pat.bed` don't have much explanatory power, and indeed, its p-value is also large.

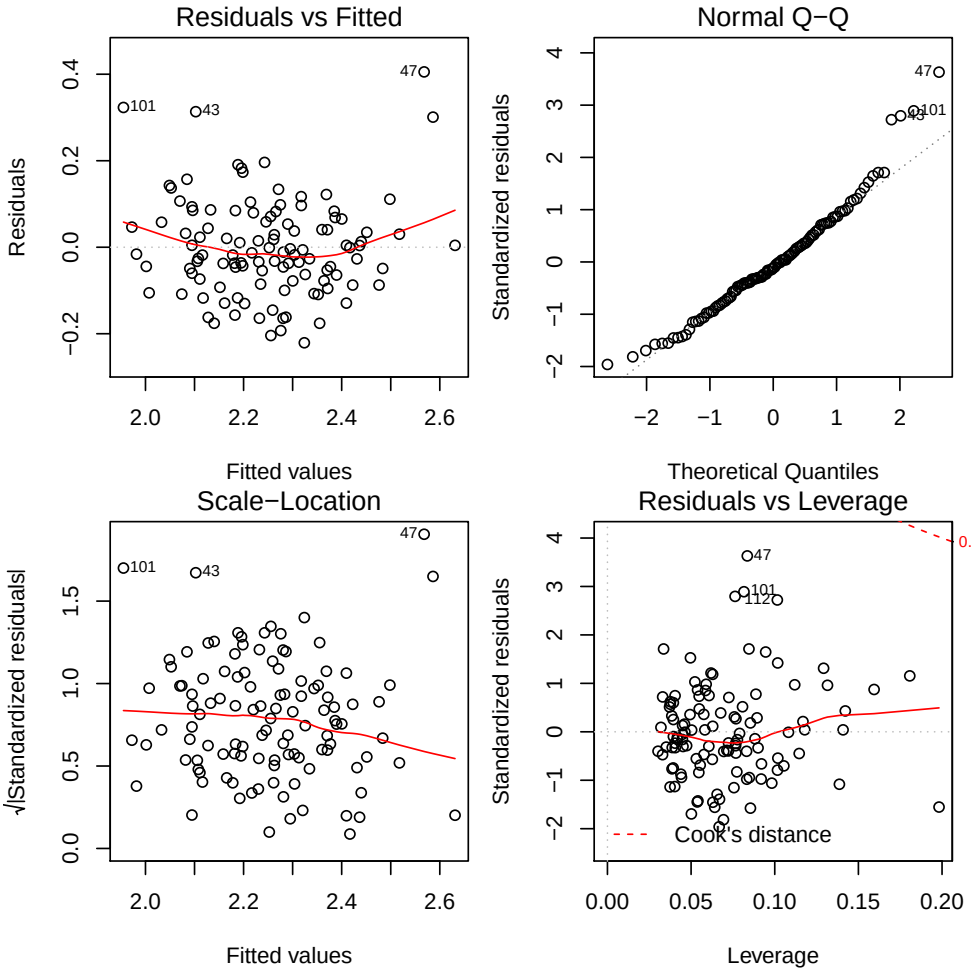
Now, we perform backwards elimination using `fit.03` as our starting model. We remove the variable `pat.bed`:

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

Call:
lm(formula = length ~ age + inf + region + pat + pat.nurs, data = senic.03)

Residuals:
     Min       1Q   Median       3Q      Max
-0.22091 -0.07352 -0.01293  0.06529  0.40542
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.225928   0.170106   7.207 9.10e-11 ***
age          0.007785   0.002554   3.049  0.00291 **
inf          2.113288   0.406384   5.200 9.89e-07 ***
regionN     -0.077614   0.031071  -2.498  0.01404 *
regionS     -0.124277   0.030623  -4.058 9.53e-05 ***
regionW     -0.211367   0.038711  -5.460 3.21e-07 ***
pat          0.051527   0.016987   3.033  0.00305 **
pat.nurs     0.090813   0.036918   2.460  0.01553 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1166 on 105 degrees of freedom
Multiple R-squared: 0.5974,        Adjusted R-squared: 0.5706
F-statistic: 22.26 on 7 and 105 DF,  p-value: < 2.2e-16
```

Note that the F-statistic increased. Since the rest of the variables are statistically significant, `pat.bed` is the only predictor that is left out of the model.
Now we look at the residuals of model `fit.P1`:

```
> par(mfrow=c(2,2))
> plot(fit.P1)
```



Model diagnostics plots look similar to the ones of the fit containing all predictors. The assumptions of linearity, constant variance, uncorrelated errors, and normality of the errors seem to be fine.

d) Backward elimination:

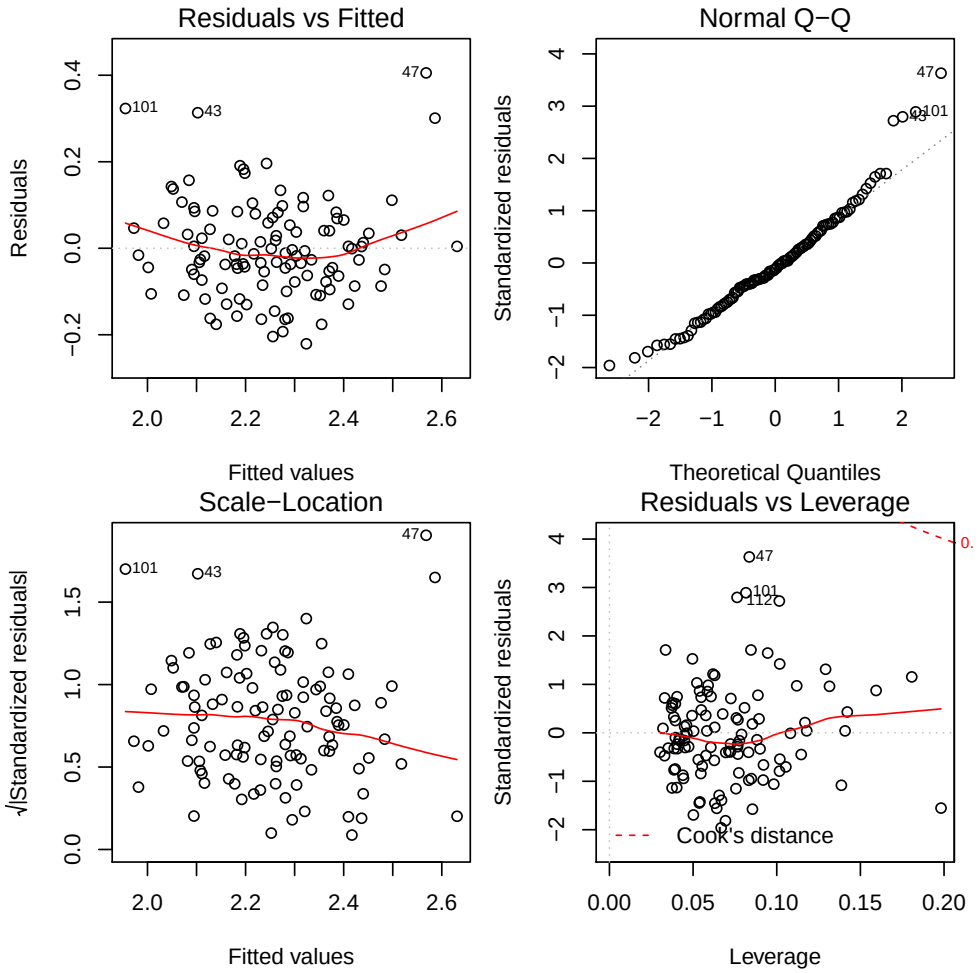
```
> fit.B <- step(fit.03, direction="backward")
```

```
Start:  AIC=-476.65
length ~ age + inf + region + pat + pat.bed + pat.nurs

           Df Sum of Sq    RSS     AIC
- pat.bed   1   0.00913 1.4281 -477.93
<none>                  1.4190 -476.65
- pat.nurs  1   0.06321 1.4822 -473.73
- pat       1   0.08850 1.5075 -471.81
- age       1   0.13231 1.5513 -468.58
- inf       1   0.33671 1.7557 -454.59
- region    3   0.41948 1.8385 -453.39

Step:  AIC=-477.93
length ~ age + inf + region + pat + pat.nurs

           Df Sum of Sq    RSS     AIC
<none>                  1.4281 -477.93
- pat.nurs  1   0.08230 1.5104 -473.60
- pat       1   0.12514 1.5533 -470.44
- age       1   0.12641 1.5545 -470.34
- inf       1   0.36780 1.7959 -454.03
- region    3   0.47346 1.9016 -451.57
```

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

```
Call:
lm(formula = length ~ age + inf + region + pat + pat.nurs, data = senic.03)

Residuals:
     Min       1Q   Median       3Q      Max
-0.22091 -0.07352 -0.01293  0.06529  0.40542

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.225928   0.170106   7.207 9.10e-11 ***
age          0.007785   0.002554   3.049  0.00291 **
inf          2.113288   0.406384   5.200 9.89e-07 ***
regionN     -0.077614   0.031071  -2.498  0.01404 *
regionS     -0.124277   0.030623  -4.058 9.53e-05 ***
regionW     -0.211367   0.038711  -5.460 3.21e-07 ***
pat          0.051527   0.016987   3.033  0.00305 **
pat.nurs     0.090813   0.036918   2.460  0.01553 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1166 on 105 degrees of freedom
Multiple R-squared: 0.5974,        Adjusted R-squared: 0.5706
F-statistic: 22.26 on 7 and 105 DF,  p-value: < 2.2e-16
```

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



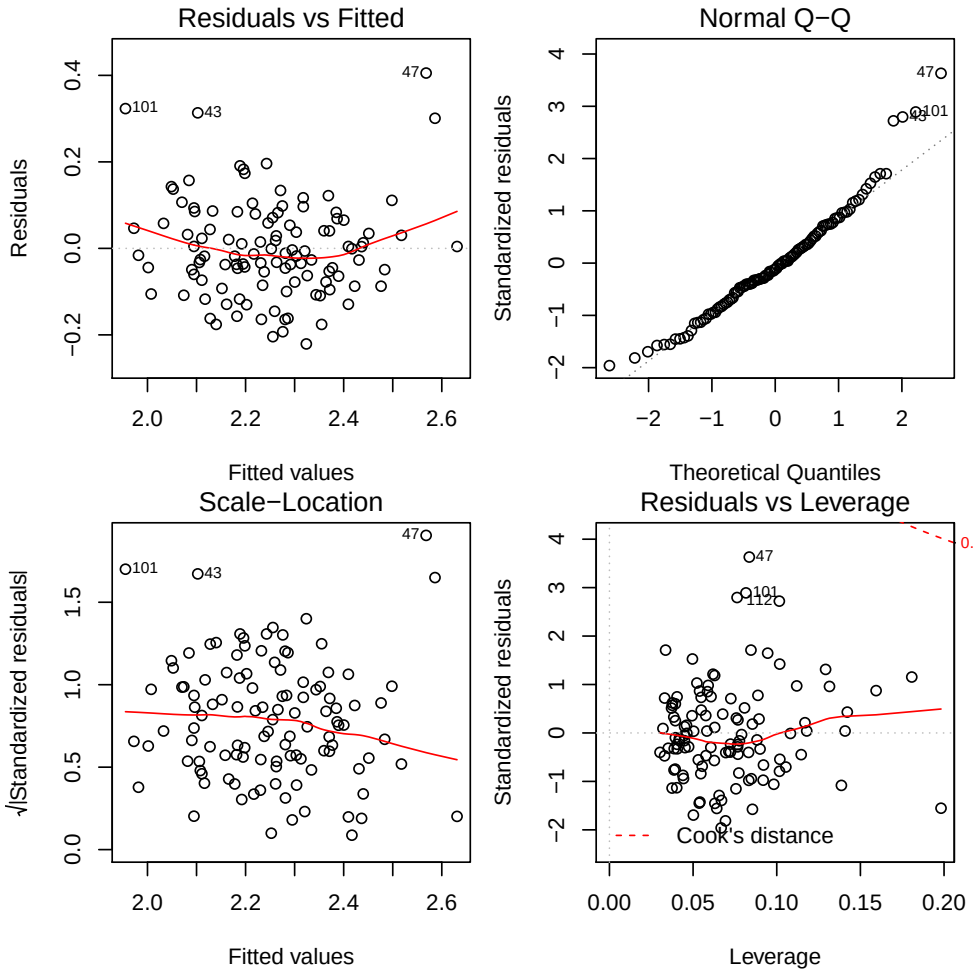
The backward elimination using AIC only removes the variable `pat.bed` from the model, just as the backward elimination using the p-values did.

e) Forward selection:

```
> fit.empty <- lm(length ~ 1, data=senic.03)
>     scp   <- list(lower=~1, upper=~age + inf + region + pat + pat.bed + pat.nurs)
>     fit.F <- step(fit.empty, scope=scp, direction="forward")
Start:  AIC=-389.11
length ~ 1

           Df Sum of Sq    RSS     AIC
+ inf       1   0.99451 2.5529 -424.29
+ 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=-424.29
length ~ inf

           Df Sum of Sq    RSS     AIC
+ region    3   0.74920 1.8037 -457.54
+ pat.nurs  1   0.33606 2.2169 -438.24
+ pat.bed   1   0.32383 2.2291 -437.61
+ pat       1   0.31243 2.2405 -437.04
+ age       1   0.12482 2.4281 -427.95
<none>                 2.5530 -424.29
```

```
Step:  AIC=-457.54
length ~ inf + region

           Df Sum of Sq    RSS     AIC
+ pat       1   0.15630 1.6474 -465.78
+ pat.nurs  1   0.15435 1.6494 -465.65
+ age       1   0.10000 1.7037 -461.99
+ pat.bed   1   0.08173 1.7220 -460.78
<none>                  1.8037 -457.54

Step:  AIC=-465.78
length ~ inf + region + pat

           Df Sum of Sq    RSS     AIC
+ age       1  0.137033 1.5104 -473.60
+ pat.nurs  1  0.092917 1.5545 -470.34
<none>                  1.6474 -465.78
+ pat.bed   1  0.017882 1.6296 -465.02

Step:  AIC=-473.6
length ~ inf + region + pat + age

           Df Sum of Sq    RSS     AIC
+ pat.nurs  1  0.082298 1.4281 -477.93
+ pat.bed   1  0.028215 1.4822 -473.73
<none>                  1.5104 -473.60

Step:  AIC=-477.93
length ~ inf + region + pat + age + pat.nurs

          Df Sum of Sq    RSS     AIC
<none>                 1.4281 -477.93
+ pat.bed  1 0.0091256 1.4190 -476.65
> summary(fit.F)
Call:
lm(formula = length ~ inf + region + pat + age + pat.nurs, data = senic.03)

Residuals:
     Min       1Q   Median       3Q      Max
-0.22091 -0.07352 -0.01293  0.06529  0.40542

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.225928   0.170106   7.207 9.10e-11 ***
inf          2.113288   0.406384   5.200 9.89e-07 ***
regionN     -0.077614   0.031071  -2.498  0.01404 *
regionS     -0.124277   0.030623  -4.058 9.53e-05 ***
regionW     -0.211367   0.038711  -5.460 3.21e-07 ***
pat          0.051527   0.016987   3.033  0.00305 **
age          0.007785   0.002554   3.049  0.00291 **
pat.nurs     0.090813   0.036918   2.460  0.01553 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1166 on 105 degrees of freedom
Multiple R-squared: 0.5974,        Adjusted R-squared: 0.5706
F-statistic: 22.26 on 7 and 105 DF,  p-value: < 2.2e-16
```

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



We get the same result as when we performed a backward elimination using AIC and when using p-values (only predictor `pat.bed` has been taken out of the model). Note that this happened in our particular example and is not always the case.

f)

```
> step(fit.03, direction="both")
Start:  AIC=-476.65
length ~ age + inf + region + pat + pat.bed + pat.nurs

           Df Sum of Sq    RSS     AIC
- pat.bed   1   0.00913 1.4281 -477.93
<none>                 1.4190 -476.65
- pat.nurs  1   0.06321 1.4822 -473.73
- pat       1   0.08850 1.5075 -471.81
- age       1   0.13231 1.5513 -468.58
- inf       1   0.33671 1.7557 -454.59
- region    3   0.41948 1.8385 -453.39

Step:  AIC=-477.93
length ~ age + inf + region + pat + pat.nurs

           Df Sum of Sq    RSS     AIC
<none>                 1.4281 -477.93
+ pat.bed   1   0.00913 1.4190 -476.65
- pat.nurs  1   0.08230 1.5104 -473.60
- pat       1   0.12514 1.5533 -470.44
- age       1   0.12641 1.5545 -470.34
- inf       1   0.36780 1.7959 -454.03
```

```
- region    3   0.47346 1.9016 -451.57

Call:
lm(formula = length ~ age + inf + region + pat + pat.nurs, data = senic.03)

Coefficients:
(Intercept)          age          inf      regionN
   1.225928     0.007785     2.113288    -0.077614
    regionS      regionW          pat     pat.nurs
  -0.124277    -0.211367     0.051527     0.090813
```

Starting with the full model leaves `pat.nurs` out the model. Therefore, this method yields to the same result as models `fit.P1`, `fit.B`, and `fit.F`.

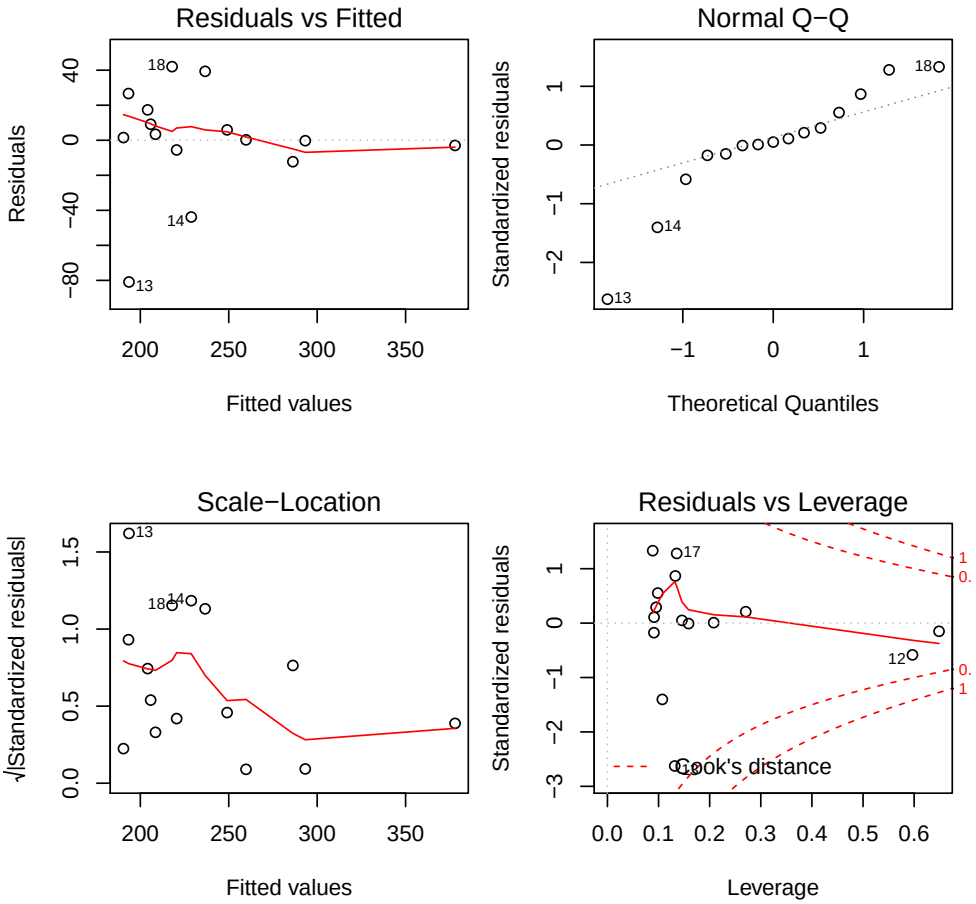
```
> step(fit.empty, scope=scp, direction="both")
Start:  AIC=-389.11
length ~ 1

           Df Sum of Sq    RSS     AIC
+ inf       1   0.99451 2.5529 -424.29
+ 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=-424.29
length ~ inf

           Df Sum of Sq    RSS     AIC
+ region    3   0.74920 1.8037 -457.54
+ pat.nurs  1   0.33606 2.2169 -438.24
+ pat.bed   1   0.32383 2.2291 -437.61
+ pat       1   0.31243 2.2405 -437.04
+ age       1   0.12482 2.4281 -427.95
<none>                  2.5529 -424.29
- inf       1   0.99451 3.5475 -389.11

Step:  AIC=-457.54
length ~ inf + region

           Df Sum of Sq    RSS     AIC
+ pat       1   0.15630 1.6474 -465.78
+ pat.nurs  1   0.15435 1.6494 -465.65
+ age       1   0.10000 1.7037 -461.99
+ pat.bed   1   0.08173 1.7220 -460.78
<none>                  1.8037 -457.54
- region    3   0.74920 2.5530 -424.29
- inf       1   0.76104 2.5648 -419.76

Step:  AIC=-465.78
length ~ inf + region + pat

           Df Sum of Sq    RSS     AIC
+ age       1   0.13703 1.5104 -473.60
+ pat.nurs  1   0.09292 1.5545 -470.34
<none>                  1.6474 -465.78
+ pat.bed   1   0.01788 1.6296 -465.02
- pat       1   0.15630 1.8037 -457.54
```

```
- inf       1   0.30551 1.9529 -448.56
- region    3   0.59308 2.2405 -437.04

Step:  AIC=-473.6
length ~ inf + region + pat + age

           Df Sum of Sq    RSS     AIC
+ pat.nurs  1   0.08230 1.4281 -477.93
+ pat.bed   1   0.02822 1.4822 -473.73
<none>                  1.5104 -473.60
- age       1   0.13703 1.6474 -465.78
- pat       1   0.19334 1.7037 -461.99
- inf       1   0.29348 1.8039 -455.53
- region    3   0.54539 2.0558 -444.76

Step:  AIC=-477.93
length ~ inf + region + pat + age + pat.nurs

           Df Sum of Sq    RSS     AIC
<none>                  1.4281 -477.93
+ pat.bed   1   0.00913 1.4190 -476.65
- pat.nurs  1   0.08230 1.5104 -473.60
- pat       1   0.12514 1.5533 -470.44
- age       1   0.12641 1.5545 -470.34
- inf       1   0.36780 1.7959 -454.03
- region    3   0.47346 1.9016 -451.57

Call:
lm(formula = length ~ inf + region + pat + age + pat.nurs, data = senic.03)

Coefficients:
(Intercept)          inf      regionN      regionS
   1.225928     2.113288    -0.077614    -0.124277
    regionW          pat          age     pat.nurs
  -0.211367     0.051527     0.007785     0.090813
```

Doing stepwise starting with the empty model yields the same result than doing stepwise starting with the full model, backward elimination and forward elimination. Note that this is not always the case: applying these methods with different data could give us different results.

2. **a)** We first fit the main-effects model:

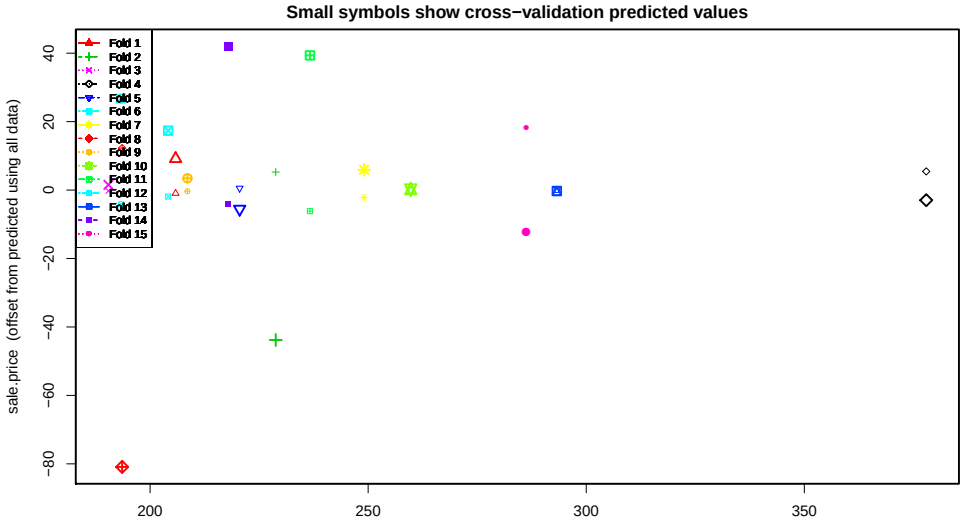
```
> library(DAAG)
> fit00 <- lm(sale.price ~ area + bedrooms, data=houseprices)
> summary(fit00)
Call:
lm(formula = sale.price ~ area + bedrooms, data = houseprices)

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 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.06 on 12 degrees of freedom
Multiple R-squared: 0.731,        Adjusted R-squared: 0.6861
F-statistic:  16.3 on 2 and 12 DF,  p-value: 0.0003792
> par(mfrow=c(2,2))
> plot(fit00)
```



From the diagnostic plots, we see that constant variance and normality assumptions are violated. We use the function `CVlm` to do a leave-one-out cross-validation.

```
> res <- CVlm(houseprices, sale.price ~ area + bedrooms, m=15, printit=FALSE,
              plotit="Residual")
```

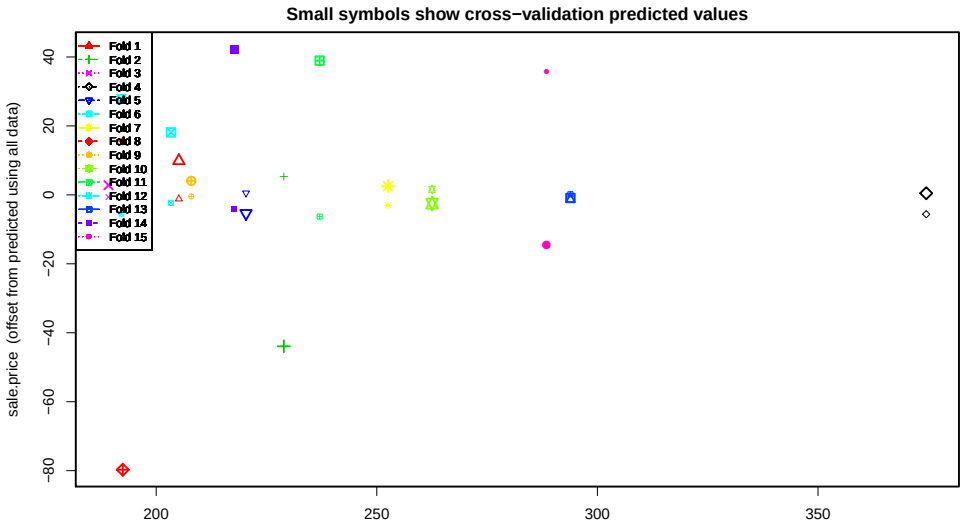
```
> RSS <- mean((res$sale.price-res$cvpred)^2)
> RSS
[1] 1187.989
```



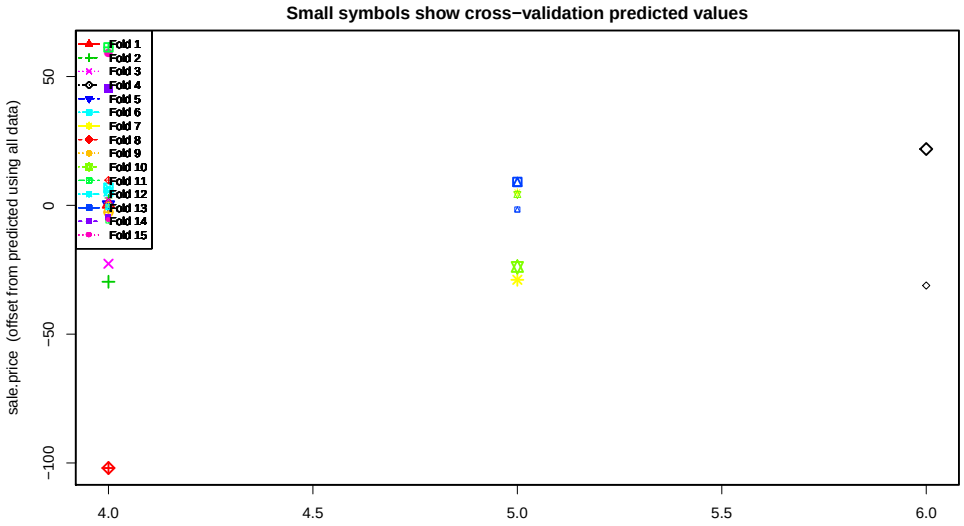
Just using these variable, there are three more models: the interaction model and two models with just one predictor. We compare the cross-validation MSE for each:

```
> res <- CVlm(houseprices, sale.price ~ area * bedrooms, m=15, printit=FALSE,
            plotit="Residual")
```

```
> RSS <- mean((res$sale.price-res$cvpred)^2)
> RSS
[1] 1335.892
```

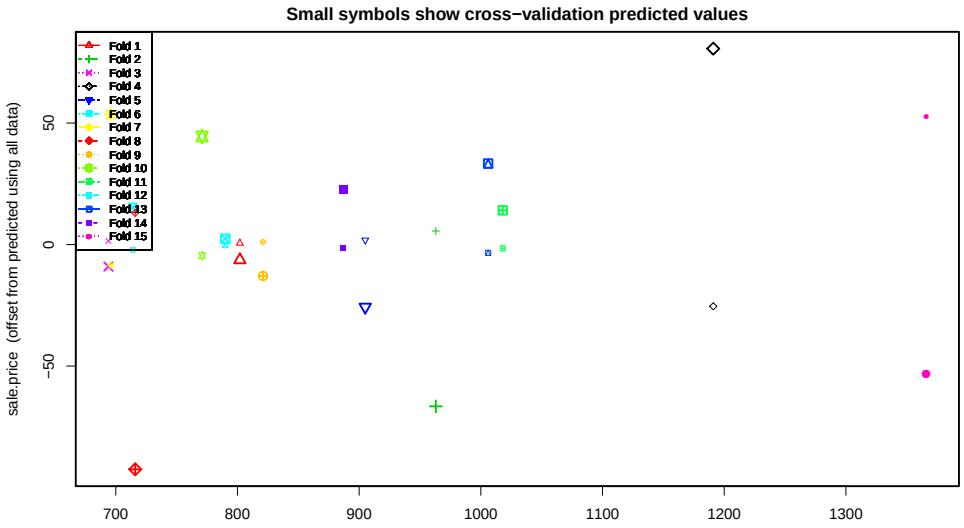



```
> res <- CVlm(houseprices, sale.price ~ bedrooms, m=15, printit=FALSE,
            plotit="Residual")
> RSS <- mean((res$sale.price-res$cvpred)^2)
> RSS
[1] 2022.884
```




```
> res <- CVlm(houseprices, sale.price ~ area, m=15, printit=FALSE,
              plotit="Residual")
> RSS <- mean((res$sale.price-res$cvpred)^2)
> RSS
```

```
[1] 3247.338
```



All the other models are considerably worse: The cross-validation MSE rises from 1188 to 1336, 2023, and 3247 respectively.

b) "By hand" cross validation:

```
> crossvalidation <- function(form) {
   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(form, data=dat.red)

       ## Predict the i-th observation
       pred[i] <- predict(fit.red, newdata=dat[i,])
     }
```

```
  ## compute the mean square prediction error
  return(mean((houseprices$sale.price-pred)^2))
 }
> crossvalidation(sale.price ~ area + bedrooms)
[1] 1187.989
> crossvalidation(sale.price ~ area * bedrooms)
[1] 1335.892
> crossvalidation(sale.price ~ bedrooms)
[1] 2022.884
> crossvalidation(sale.price ~ area)
[1] 3247.338
```