STAT 210
Applied Statistics and Data Analysis
Problem List 9
(Due on Week 10)

Exercise 1

In this exercise we re-visit exercise 1 from last week. Then, we extracted the data corresponding to species 'setosa' from the `iris` data set and considered a simple linear regression of `Sepal.Width` on `Sepal.Length`. We reproduce below the summary table for this regression.

```r
iris.set <- subset(iris, Species == 'setosa')
modelA <- lm(Sepal.Width ~ Sepal.Length, data = iris.set)
summary(modelA)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length, data = iris.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72394 -0.18273 -0.00306  0.15738  0.51709
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.5694     0.5217  -1.091    0.281
## Sepal.Length   0.7985     0.1040   7.681 6.71e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2565 on 48 degrees of freedom
## Multiple R-squared:  0.5514, Adjusted R-squared:  0.542
## F-statistic: 58.99 on 1 and 48 DF,  p-value: 6.71e-10
```

We see that the intercept is marginally significant, and in this case, where we are dealing with a physical model, it makes sense that the intercept be equal to zero. The purpose of this exercise is to fit a model with intercept equal to zero and compare with the previous model.

(i) Fit a model of `Sepal.Width` as a function of `Sepal.Length`, with intercept equal to zero. To do this, add a term `-1` to the regression equation. Produce the summary table and compare the estimated value for the regression slope with the previous result. Compare also the standard error for both estimates and the R^2 for both models. Look also at the correlation matrix for the two parameters in the model including the intercept (to do this, set `corr = TRUE` in the call to `summary`.)

```r
modelB <- lm(Sepal.Width ~ -1 + Sepal.Length, data = iris.set)
summary(modelB)
```

```
##
## Call:
```

```
## lm(formula = Sepal.Width ~ -1 + Sepal.Length, data = iris.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78398 -0.18311 -0.00811  0.18176  0.53629
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Sepal.Length 0.685328   0.007244   94.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.257 on 49 degrees of freedom
## Multiple R-squared:  0.9946, Adjusted R-squared:  0.9944
## F-statistic:  8952 on 1 and 49 DF,  p-value: < 2.2e-16
```

The coefficients for the slope, standard errors and R^2 are

	ModelA	ModelB
Slope	0.7985	0.6853
Std Error (slope)	0.1040	0.0072
Std Error (res)	0.2565	0.257
R^2	0.5514	0.9946

To get the correlation matrix,

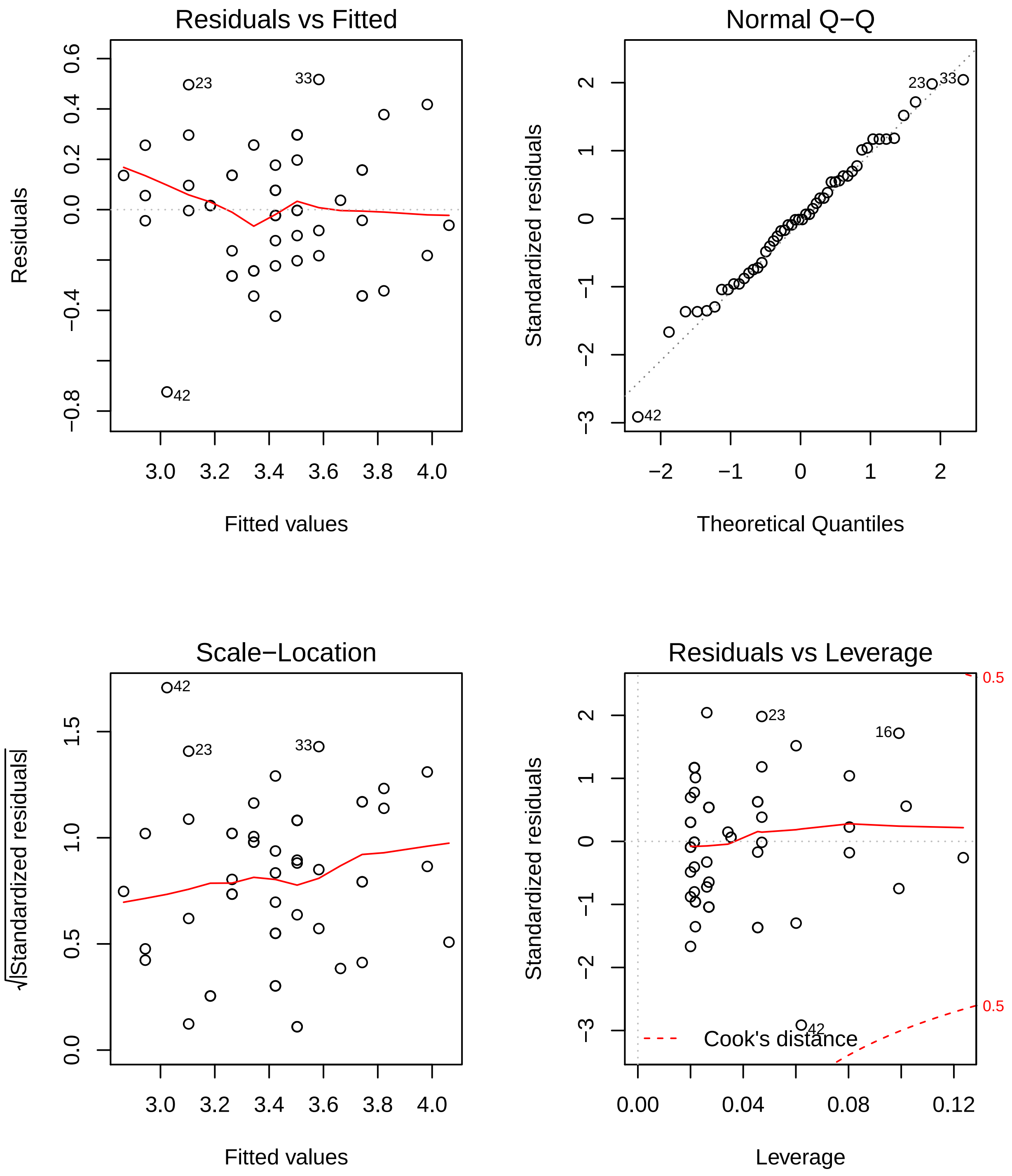
```
summary(modelA, corr = TRUE)
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length, data = iris.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72394 -0.18273 -0.00306  0.15738  0.51709
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.5694     0.5217  -1.091    0.281
## Sepal.Length   0.7985     0.1040   7.681 6.71e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2565 on 48 degrees of freedom
## Multiple R-squared:  0.5514, Adjusted R-squared:  0.542
## F-statistic: 58.99 on 1 and 48 DF,  p-value: 6.71e-10
##
## Correlation of Coefficients:
##              (Intercept)
## Sepal.Length -1.00
```

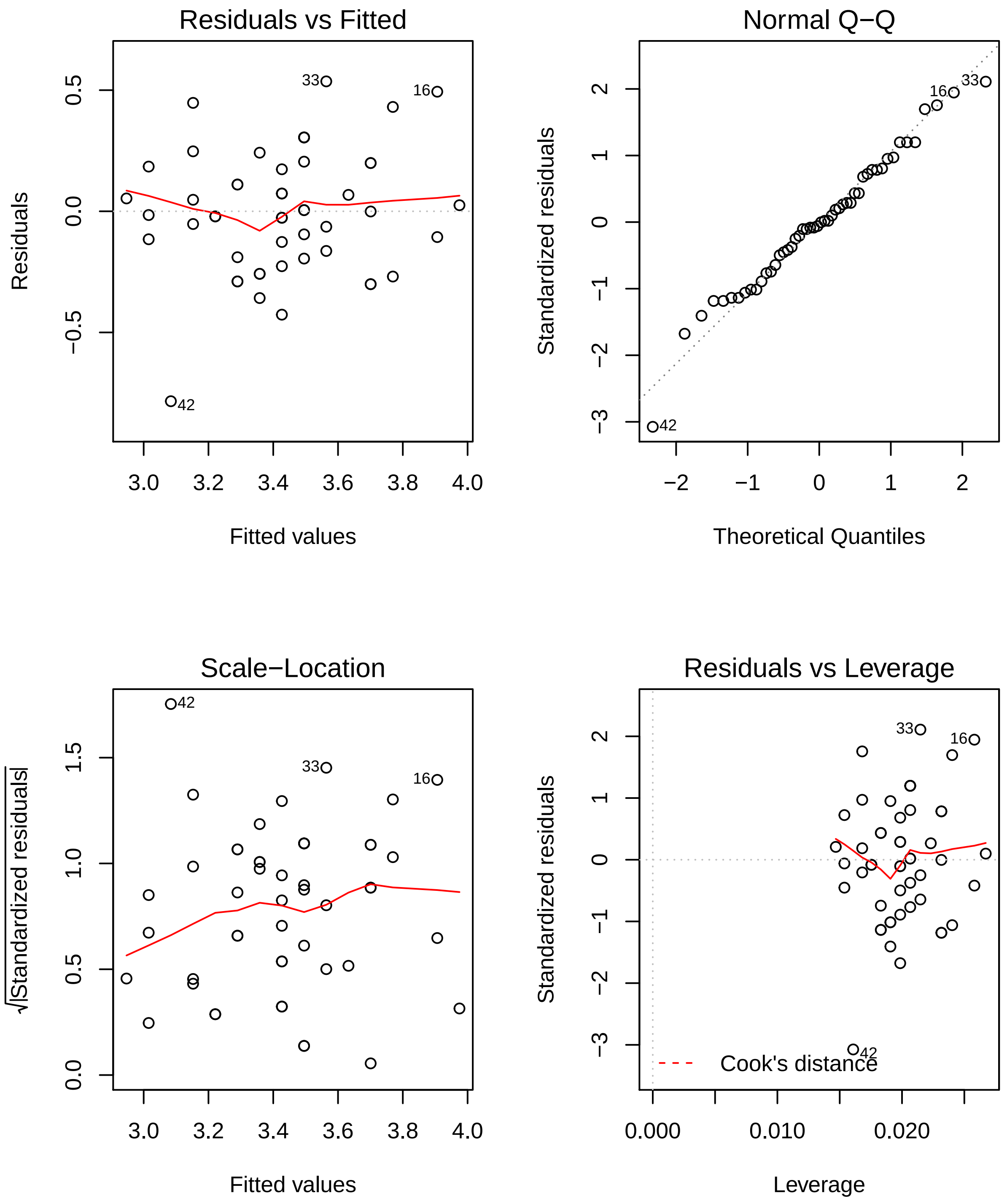
Note that is the correlation between the coefficients in the model.

(ii) Plot diagnostic graphs for both models and comment on the results.

```
par(mfrow=c(2,2))
plot(modelA)
```


Residuals vs Fitted
Residuals
0.6
0.4
0.2
0.0
−0.4
−0.8
3.0
3.2
3.4
3.6
3.8
4.0
Fitted values
23
33
42
Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−3
−2
−1
0
1
2
Theoretical Quantiles
23
33
42
Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
3.0
3.2
3.4
3.6
3.8
4.0
Fitted values
42
23
33
Residuals vs Leverage
Standardized residuals
2
1
0
−1
−2
−3
0.00
0.04
0.08
0.12
Leverage
23
16
42
0.5
0.5
Cook's distance

```
plot(modelB)
```



```
par(mfrow=c(1,1))
```

```
library(car)
ncvTest(modelA)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.07602347, Df = 1, p = 0.78276
```

```r
ncvTest(modelB)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.01510656, Df = 1, p = 0.90218
```

We now want to compare the predictive power for each type of model in this particular example. For this, we are going to use the idea behind leave-one-out cross-validation. We select one data point from the set, let's say it is the i-th one, and we fit both regression models with the remaining 49 points. With this model, we predict the value for the i-th point and now we can compare with the observed value.

(iii) Implement the procedure described above for all fifty points in `iris.vir`. Produce a matrix with three columns, the first column (`P1`) should have the predicted value with the model including an intercept, the second column (`P2`) corresponds to the predicted value for the model without the intercept, and the third column (`O`) should be the observed value.

```r
attach(iris.set)
pred.values <- matrix(rep(0,100), ncol = 2)
for (i in 1:50) {
  xx <- Sepal.Length[-i]
  yy <- Sepal.Width[-i]
  model1 <- lm(yy ~ xx)
  model2 <- lm(yy ~ -1 + xx)
  pred.values[i,1] = predict(model1, data.frame(xx = Sepal.Length[i]))
  pred.values[i,2] = predict(model2, data.frame(xx = Sepal.Length[i]))
}
pred.values <- cbind(pred.values,Sepal.Width)
colnames(pred.values) <- c('P1', 'P2', 'O')
str(pred.values)
```

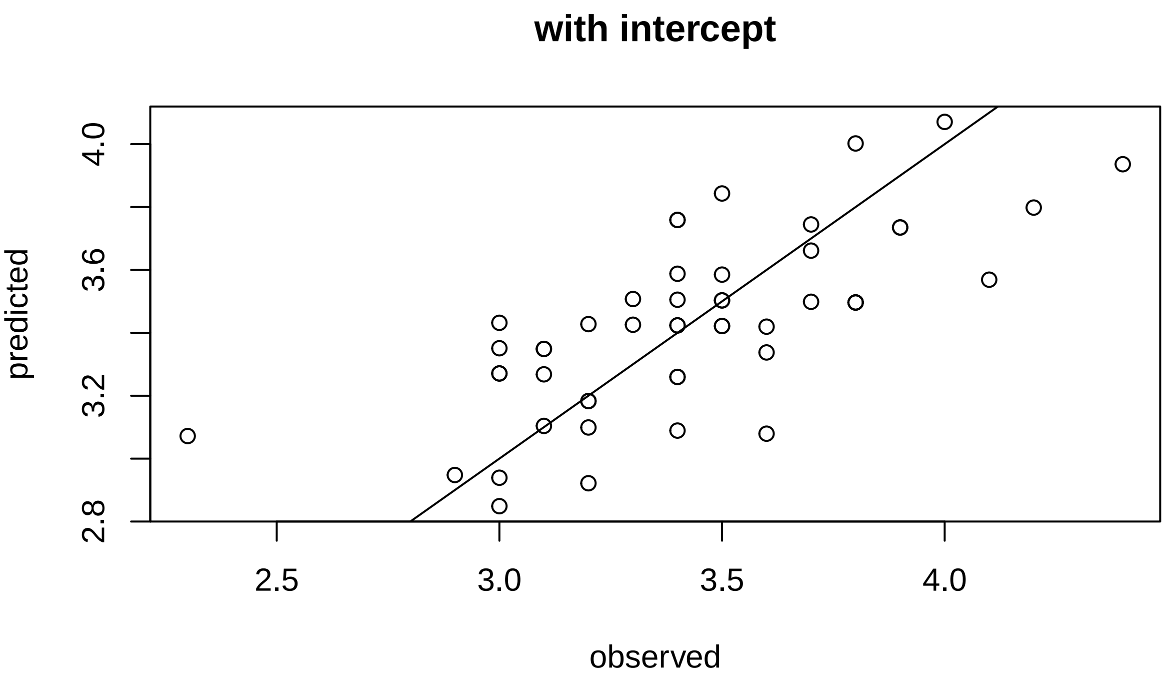
```
##  num [1:50, 1:3] 3.5 3.35 3.18 3.1 3.42 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "P1" "P2" "O"
```

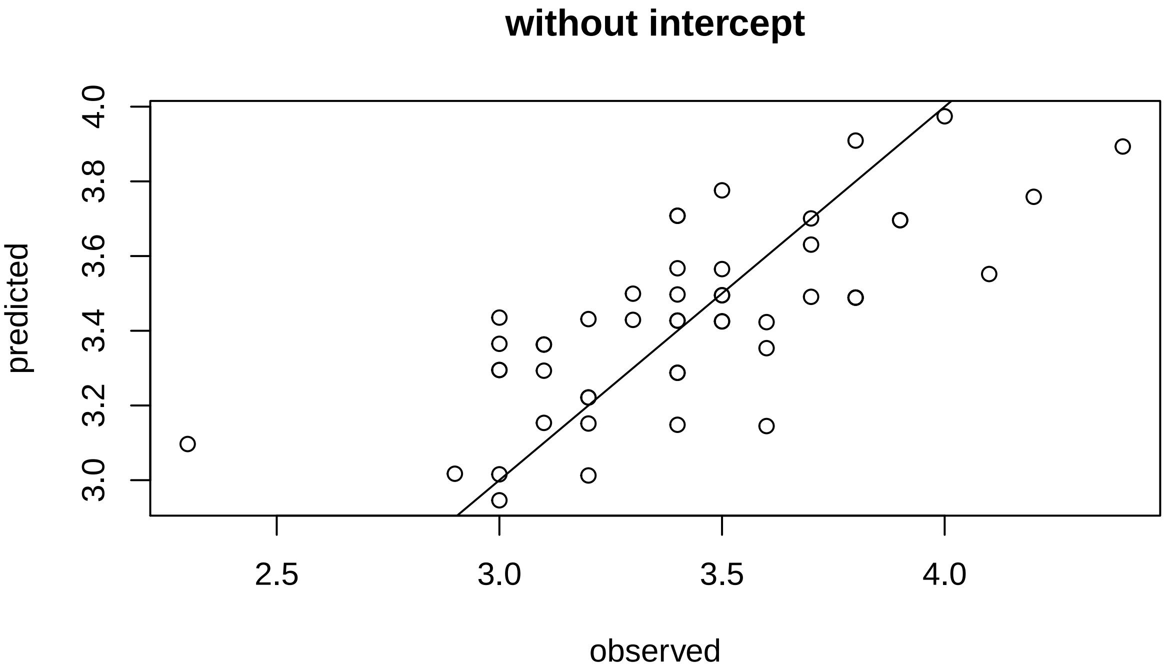
```r
head(pred.values)
```

```
##            P1       P2   O
## [1,] 3.503129 3.495073 3.5
## [2,] 3.351024 3.365070 3.0
## [3,] 3.183051 3.221419 3.2
## [4,] 3.103985 3.153408 3.1
## [5,] 3.419600 3.423130 3.6
## [6,] 3.735118 3.696049 3.9
```

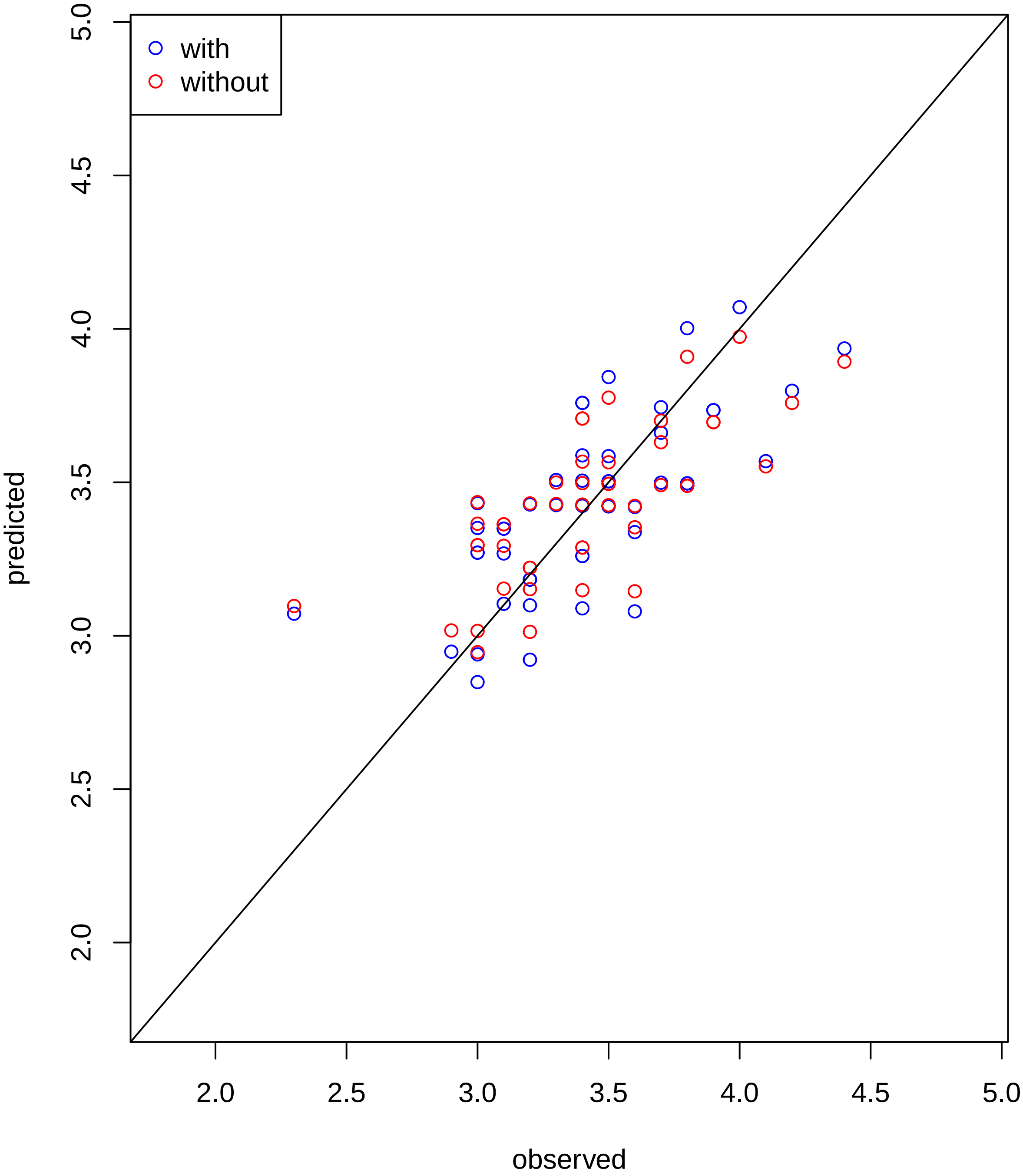
(iv) Produce graphs of `P1` against `O` and `P2` against `O`. Comment.

```r
par(mfrow=c(2,1))
plot(pred.values[,3],pred.values[,1], ylab = 'predicted',
     xlab = 'observed', main = 'with intercept')
abline(a=0,b=1)
plot(pred.values[,3],pred.values[,2], ylab = 'predicted',
     xlab = 'observed', main = 'without intercept')
abline(a=0,b=1)
```



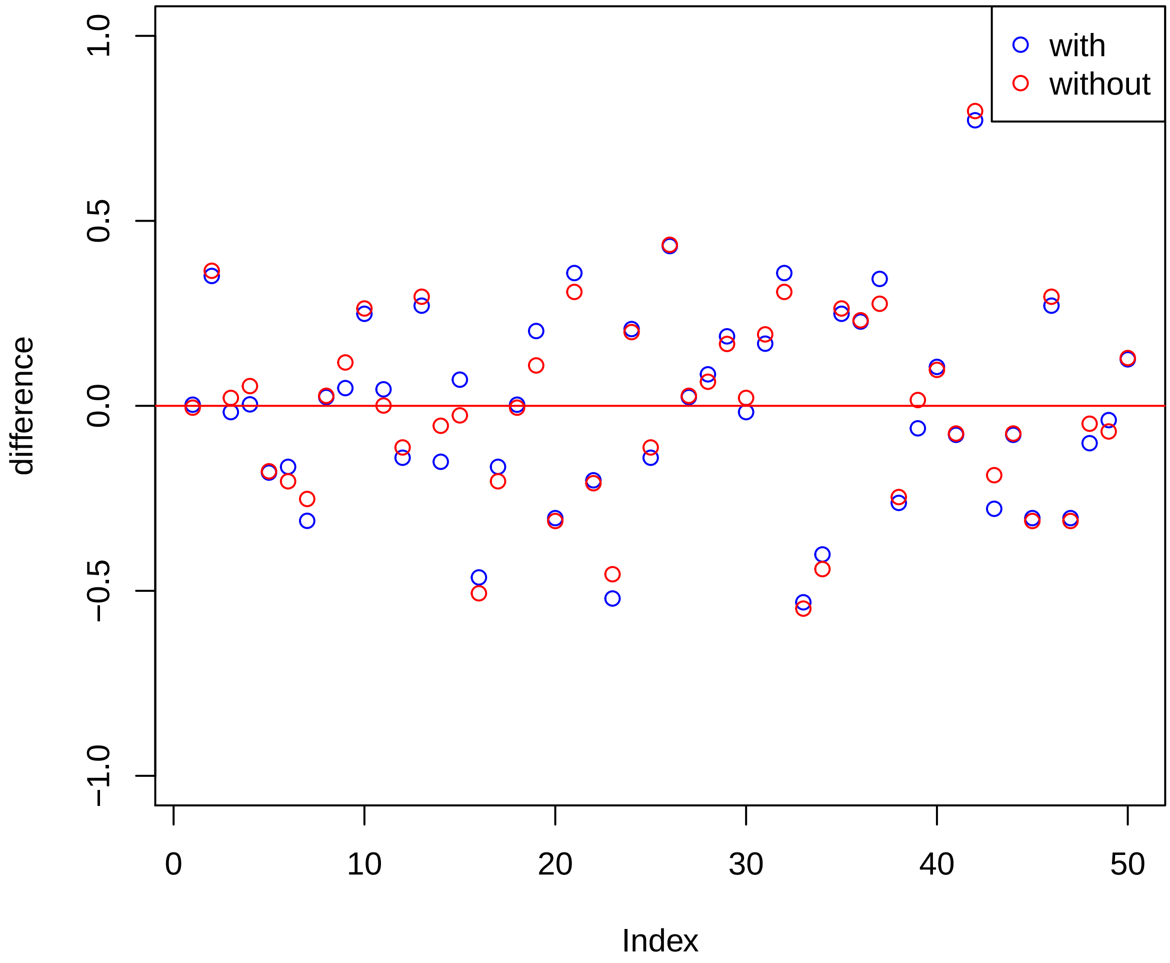


```r
par(mfrow=c(1,1))
plot(pred.values[,3],pred.values[,1], col = 'blue', ylab = 'predicted', xlab = 'observed',
     xlim = (range(pred.values)+c(-.5,+.5)), ylim = (range(pred.values)+c(-.5,+.5)))
points(pred.values[,3],pred.values[,2], col = 'red')
legend('topleft',c('with','without'),col=c('blue','red'), pch=c(1,1))
abline(0,1)
```



(v) Produce graphs of errors: P1-O and P2-O. Comment.

```
plot(pred.values[,1]-pred.values[,3], col = 'blue', ylab = 'difference', ylim = c(-1,1))
abline(h=0,col='red')
points(pred.values[,2]-pred.values[,3], col = 'red')
legend('topright',c('with','without'),col=c('blue','red'), pch=c(1,1))
```



(vi) To compare the predictive powers the models, find the average of the absolute values of the errors in both cases and comment.

```
sum(abs(pred.values[,1]-pred.values[,3]))
```

```
## [1] 10.39845
```

```
sum(abs(pred.values[,2]-pred.values[,3]))
```

```
## [1] 10.02746
```

```
mean(abs(pred.values[,1]-pred.values[,3]))
```

```
## [1] 0.2079689
```

```
mean(abs(pred.values[,2]-pred.values[,3]))
```

```
## [1] 0.2005491
```

Exercise 2

The data for this question is stored in the file `data("diamond")` in package `HH`. The dataset presents data on the price (Singapore dollars) of ladies' diamond rings and the number of carats in the ring's diamond. The data are accessed as `data(diamond)`.

```
library(HH)
data(diamond)
```

```
str(diamond)
```

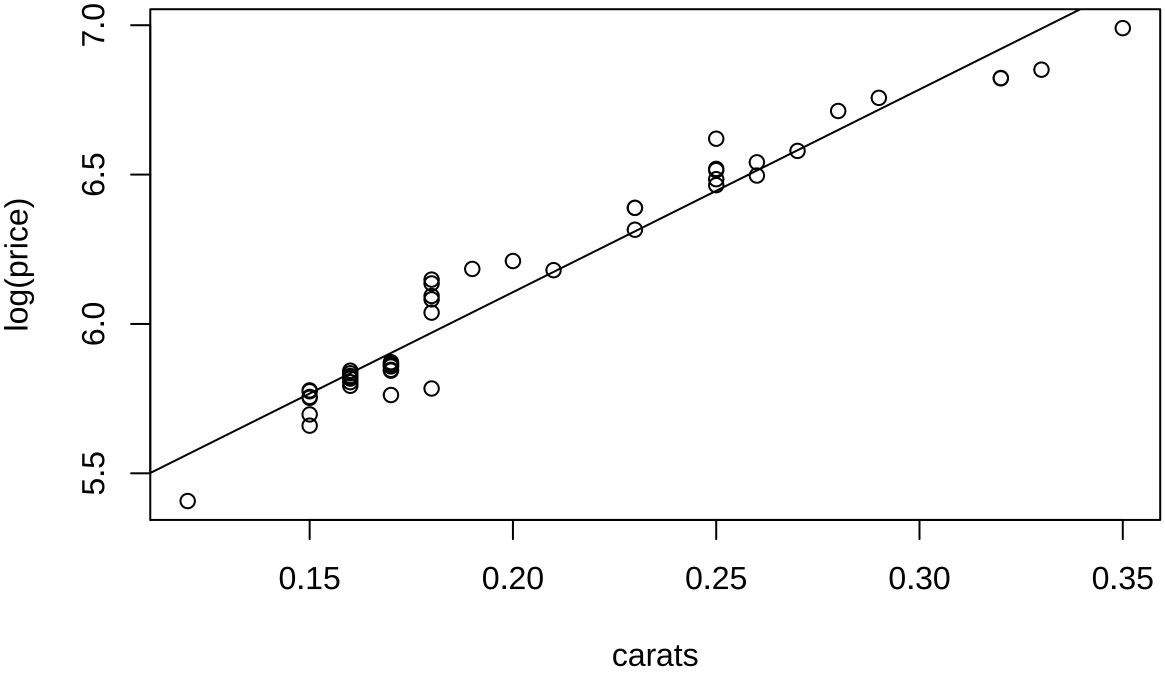
```
## 'data.frame':    48 obs. of  2 variables:
##  $ carats: num  0.17 0.16 0.17 0.18 0.25 0.16 0.15 0.19 0.21 0.15 ...
##  $ price : int  355 328 350 325 642 342 322 485 483 323 ...
```

(a) Regress the logarithm of `price` on `carats`. Do a scatterplot of these variables and include the regression line. Interpret the slope of the regression line. Write down an equation for price according to this model

```
mod1 <- lm(log(price) ~ carats, data = diamond)
summary(mod1)
```

```
##
## Call:
## lm(formula = log(price) ~ carats, data = diamond)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.186792 -0.044812 -0.005545  0.064524  0.177851
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.74891    0.04775   99.46   <2e-16 ***
## carats       6.78725    0.22547   30.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08778 on 46 degrees of freedom
## Multiple R-squared:  0.9517, Adjusted R-squared:  0.9506
## F-statistic: 906.2 on 1 and 46 DF,  p-value: < 2.2e-16
```

```
plot(log(price) ~ carats, data = diamond)
abline(mod1)
```



The slope is the rate of increase of the logarithm of the price of diamonds (in Singapure dollars) per unit increase in carat. The equation is

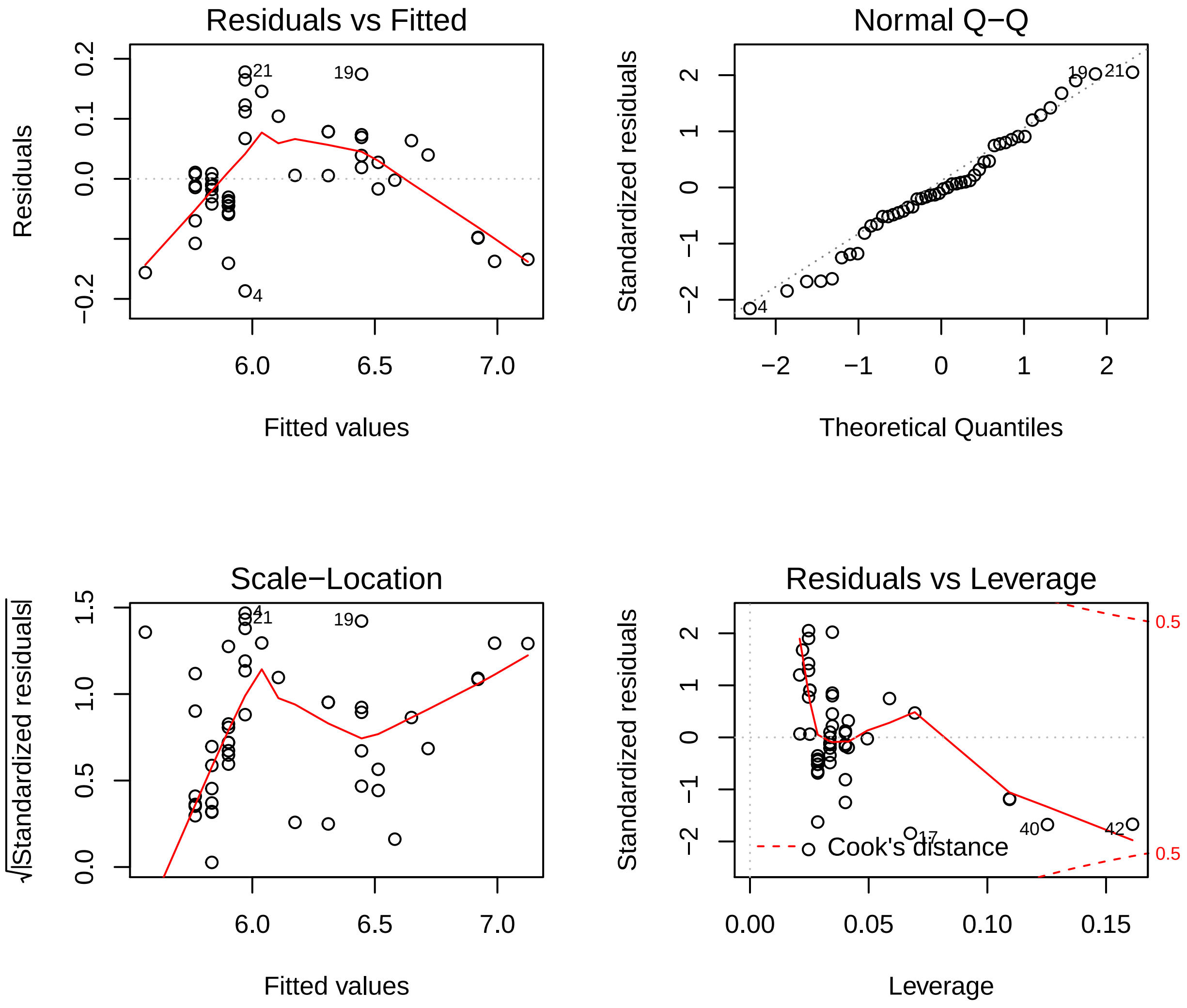
$$\log(\texttt{price}) = 4.749 + 6.787 * \texttt{carats}$$

and in Singapore dolars

$$\texttt{price} = \exp(4.749 + 6.787 * \texttt{carats}) = 115.46 * \exp(6.787 * \texttt{carats})$$

(b) Do residual plots and **discuss the results.**

```r
par(mfrow=c(2,2))
plot(mod1)
```

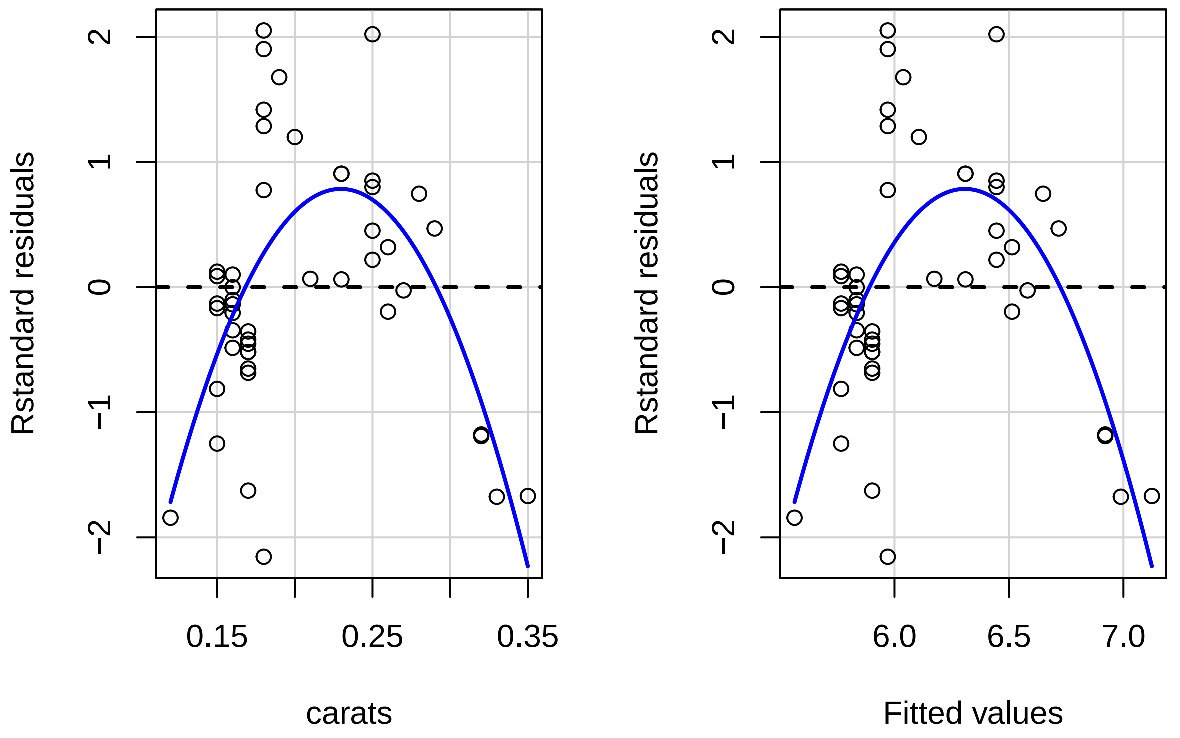


```r
par(mfrow=c(1,1))
```

The quantile plot has a good fit to the normal distribution but the other plots do not look good, residuals against fitted values shows unequal dispersion and more positive values at the center while the scale location plot shows an increasing tendency in dispersion.

A useful tool is the function `residualPlots` in the `car` package. This function plots residuals against all the regressors and also against fitted values, and adds a quadratic term. It also tests the significance of the added term and lists the p-values. In thie case, the quadratic term for carats has a small p-value.

```
residualPlots(mod1, type = 'rstandard')
```



```
##             Test stat Pr(>|Test stat|)
## carats        -5.5214        1.594e-06 ***
## Tukey test    -5.5214        3.363e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Add a quadratic term to this model and test whether the new term is significant. Compare with the previous model in terms of R^2 and AIC. Do residual plots and discuss the results. Write down an equation for price according to this model

```
mod2 <- update(mod1, ~. + I(carats^2))
summary(mod2)
```

```
##
## Call:
## lm(formula = log(price) ~ carats + I(carats^2), data = diamond)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.209897 -0.041486 -0.004652  0.036470  0.154746
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.8872     0.1605  24.225  < 2e-16 ***
## carats        14.8597     1.4726  10.091 3.91e-13 ***
## I(carats^2) -17.5370     3.1762  -5.521 1.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06852 on 45 degrees of freedom
## Multiple R-squared:  0.9712, Adjusted R-squared:  0.9699
## F-statistic: 758.8 on 2 and 45 DF,  p-value: < 2.2e-16
```

In the new model, all terms are significant. The R^2 has increased form 0.9517 to 0.9712. To calculate the AIC we use the function `stepAIC`

```
library(MASS)
stepAIC(mod2)
```

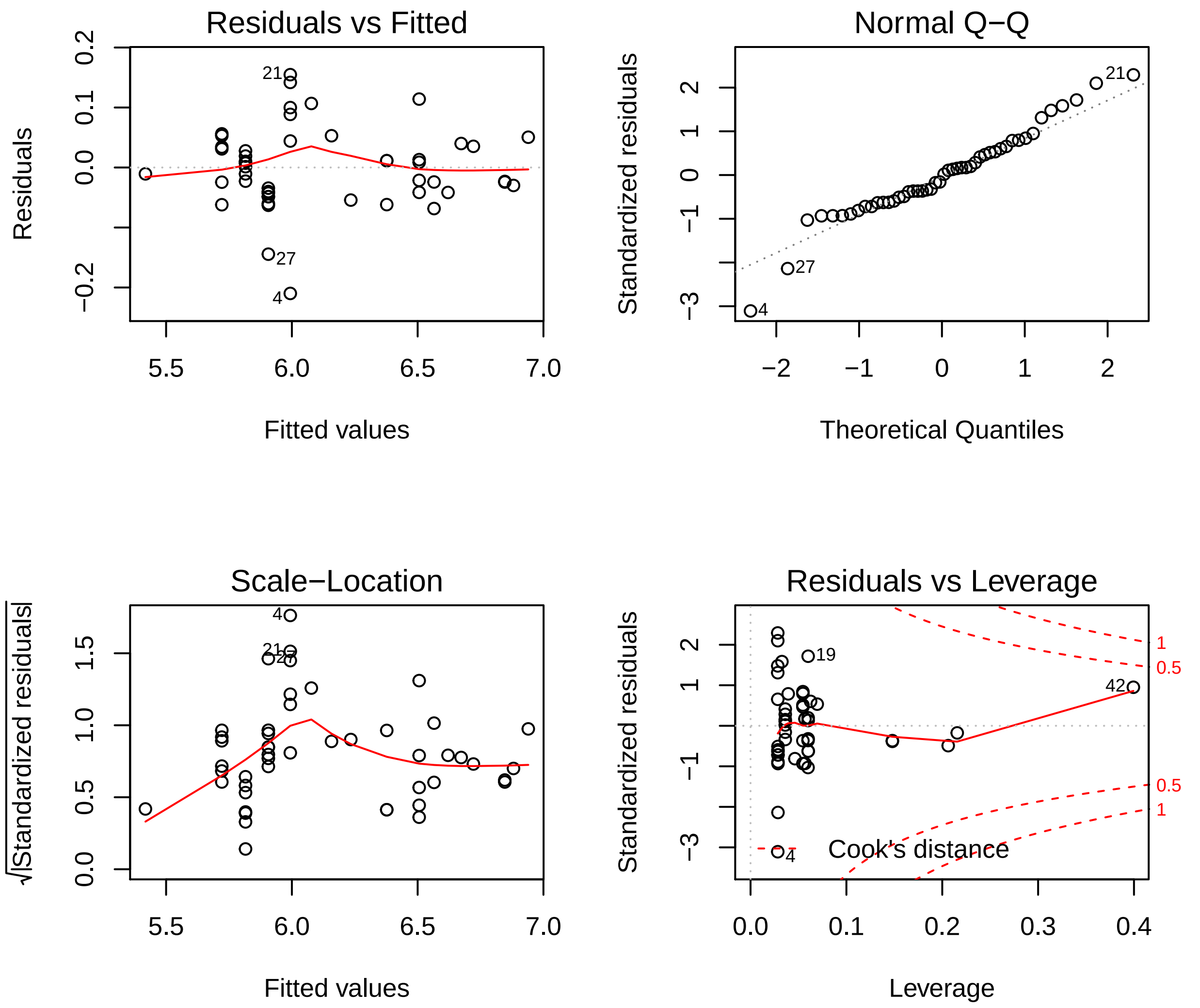
```
## Start:  AIC=-254.43
## log(price) ~ carats + I(carats^2)
##
##               Df Sum of Sq     RSS     AIC
## <none>                     0.21129 -254.43
## - I(carats^2)  1   0.14314 0.35444 -231.60
## - carats       1   0.47811 0.68940 -199.67
```

```
##
## Call:
## lm(formula = log(price) ~ carats + I(carats^2), data = diamond)
##
## Coefficients:
## (Intercept)       carats  I(carats^2)
##       3.887       14.860      -17.537
```

The first model has an AIC of -231.6, and adding the quadratic term has reduced it to -254.43.

We plot the diagnostic graphs

```
par(mfrow=c(2,2))
plot(mod2)
```



```
par(mfrow=c(1,1))
```

All the plots have improved considerably.

The equation for this model is

$$\log(\texttt{price}) = 3.887 + 14.89 * \texttt{carats} - 17.54 * \texttt{carats}^2$$

and in Singapore dollars

$$\texttt{price} = \exp(3.887 + 14.89 * \texttt{carats} - 17.54 * \texttt{carats}^2) = 48.77 * \exp(14.89 * \texttt{carats} - 17.54 * \texttt{carats}^2)$$

(d) Which model would you choose and why?

The second model. It has higher R^2, smaller AIC and the diagnostic plots are better.

Exercise 3

For this exercise we use the data set `ais` in library `DAAG`. We concentrate in six variables, `bmi`, `lbm`, `ssf`, `ht`, `wcc`, and `hc`.

```r
library(DAAG)
data(ais)
str(ais)
```

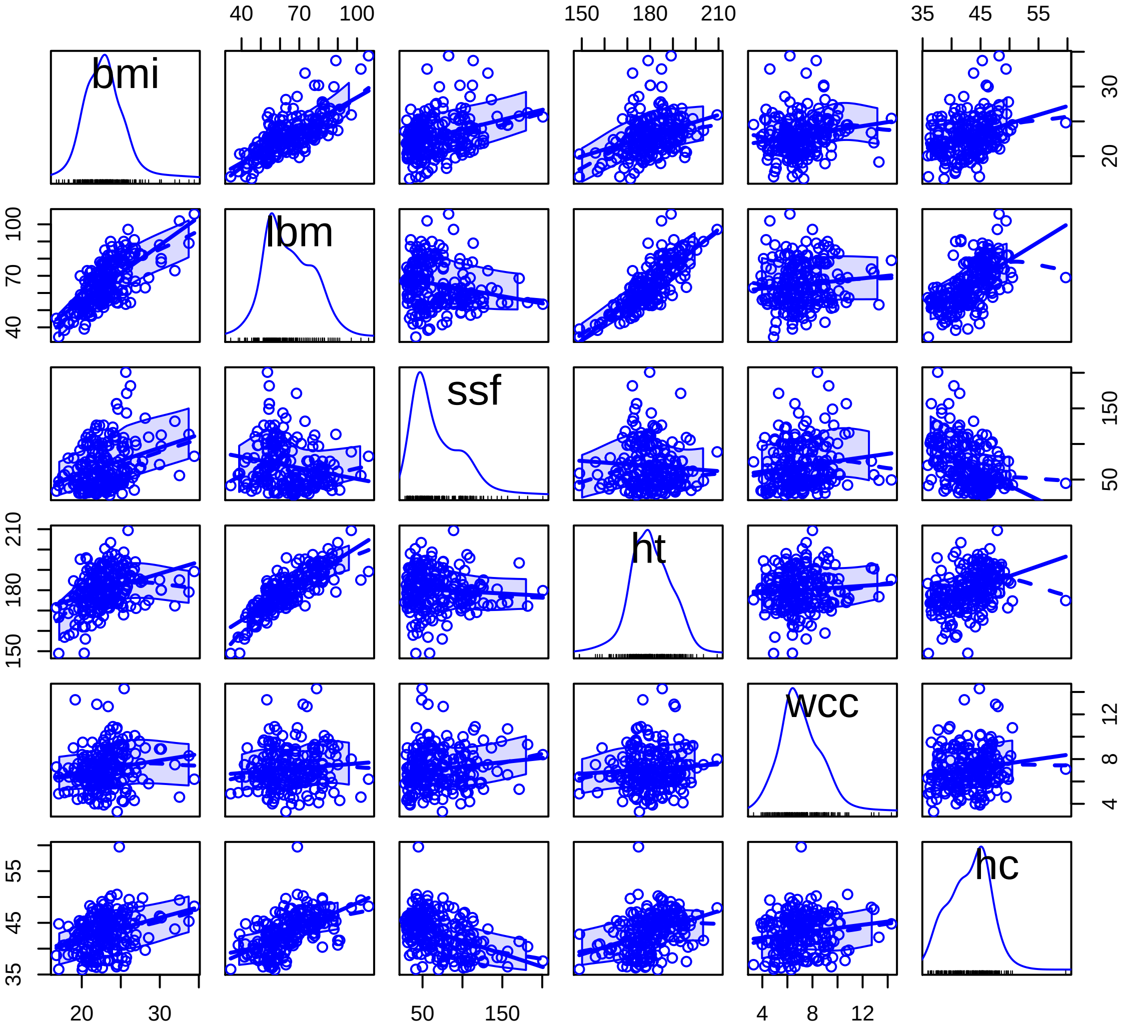
```
## 'data.frame':    202 obs. of  13 variables:
##  $ rcc   : num  3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...
##  $ wcc   : num  7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
##  $ hc    : num  37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
##  $ hg    : num  12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...
##  $ ferr  : num  60 68 21 69 29 42 73 44 41 44 ...
##  $ bmi   : num  20.6 20.7 21.9 21.9 19 ...
##  $ ssf   : num  109.1 102.8 104.6 126.4 80.3 ...
##  $ pcBfat: num  19.8 21.3 19.9 23.7 17.6 ...
##  $ lbm   : num  63.3 58.5 55.4 57.2 53.2 ...
##  $ ht    : num  196 190 178 185 185 ...
##  $ wt    : num  78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...
##  $ sex   : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sport : Factor w/ 10 levels "B_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
ex3.df <- subset(ais,select = c('bmi','lbm','ssf','ht','wcc','hc'))
str(ex3.df)
```

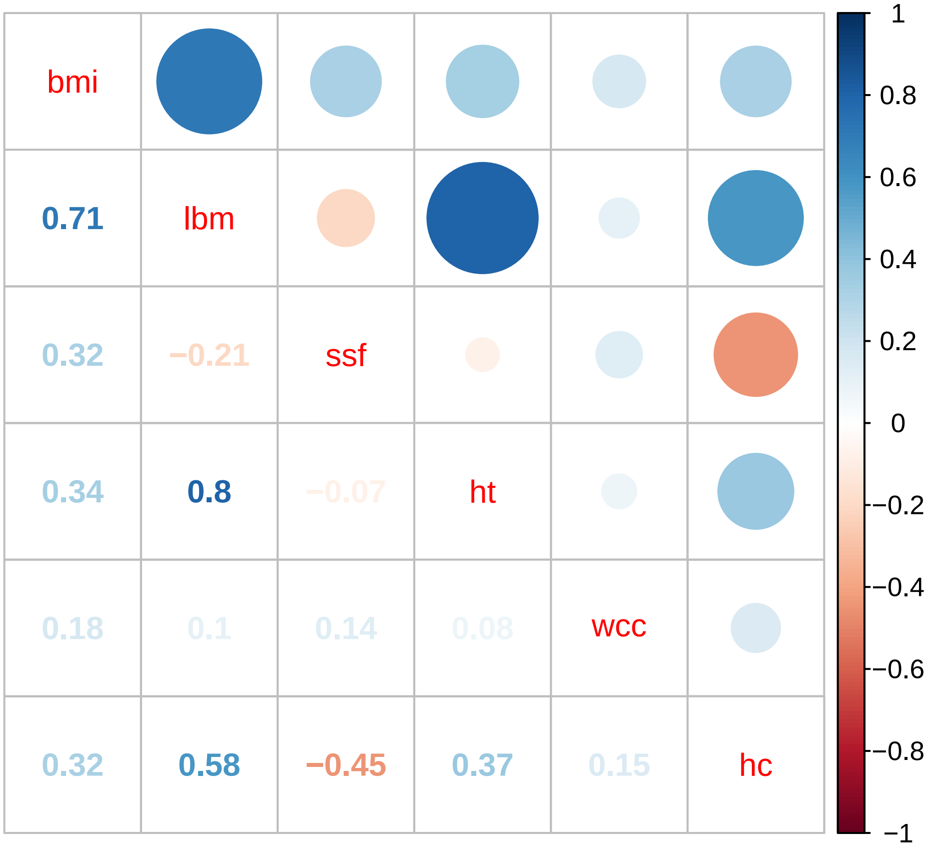
```
## 'data.frame':    202 obs. of  6 variables:
##  $ bmi: num  20.6 20.7 21.9 21.9 19 ...
##  $ lbm: num  63.3 58.5 55.4 57.2 53.2 ...
##  $ ssf: num  109.1 102.8 104.6 126.4 80.3 ...
##  $ ht : num  196 190 178 185 185 ...
##  $ wcc: num  7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...
##  $ hc : num  37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...
```

(i) Use the function `scatterplotMatrix` in the package `car` to obtain a graph matrix for the variables. Use the `corrplot.mixed` function in the package `corrplot` to draw a plot of the correlation coefficients for the six variables. Use also the `ggcorr` function in the package `GGally`. Comment.

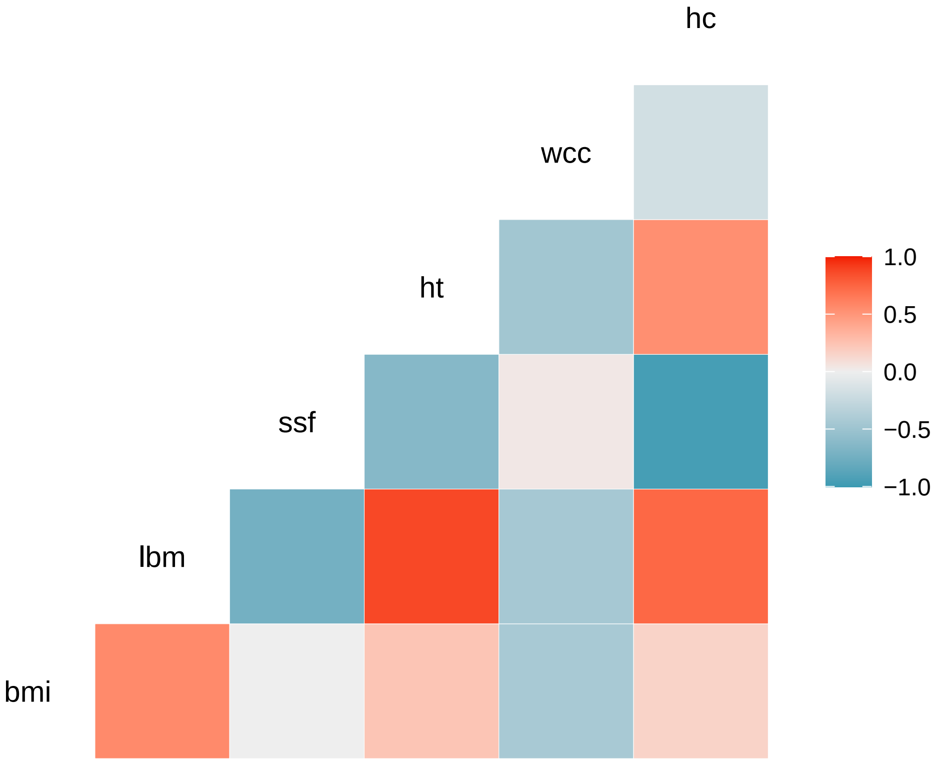
```r
scatterplotMatrix(ex3.df)
```



```
cor.ex3 <- cor(ex3.df)
corrplot::corrplot.mixed(cor.ex3)
```



```r
library(GGally)
ggcorr(cor.ex3)
```



The highest correlation corresponds to `ht` and `lbm`, with a value of 0.8.

(ii) Fit a multiple regression model for `bmi` as a function of the other variables. Print the summary table and discuss the results.

```
lm1 <- lm(bmi ~ ., data = ex3.df)
summary(lm1)
```

```
##
## Call:
## lm(formula = bmi ~ ., data = ex3.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97061 -0.27300  0.02987  0.25627  1.29559
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.599877   1.073583  37.817   <2e-16 ***
## lbm          0.321883   0.005313  60.579   <2e-16 ***
## ssf          0.050832   0.001264  40.202   <2e-16 ***
## ht          -0.237956   0.006264 -37.985   <2e-16 ***
## wcc          0.008978   0.020541   0.437    0.663
## hc           0.017608   0.013532   1.301    0.195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5038 on 196 degrees of freedom
## Multiple R-squared:  0.9698, Adjusted R-squared:  0.9691
## F-statistic:  1260 on 5 and 196 DF,  p-value: < 2.2e-16
```

The variables `wcc` and `hc` have large p-values and do not seem to be significant.

(iii) Using a stepwise procedure, select a minimal adequate model.

We choose a critical value of 0.15 for α. We remove `wcc` which has the largest p-value.

```
lm2 <- update(lm1, ~. - wcc)
summary(lm2)
```

```
##
## Call:
## lm(formula = bmi ~ lbm + ssf + ht + hc, data = ex3.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97878 -0.28435  0.02666  0.25112  1.28994
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.606591   1.071267   37.91   <2e-16 ***
## lbm          0.321894   0.005302   60.71   <2e-16 ***
## ssf          0.050961   0.001227   41.52   <2e-16 ***
## ht          -0.237976   0.006251  -38.07   <2e-16 ***
## hc           0.018793   0.013230    1.42    0.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5028 on 197 degrees of freedom
## Multiple R-squared:  0.9698, Adjusted R-squared:  0.9692
## F-statistic:  1581 on 4 and 197 DF,  p-value: < 2.2e-16
```

We now remove hc.

```
lm3 <- update(lm2, ~. - hc)
summary(lm3)
```

```
##
## Call:
## lm(formula = bmi ~ lbm + ssf + ht, data = ex3.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.06486 -0.28376  0.01867  0.26263  1.30783
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 41.471878   0.883463   46.94   <2e-16 ***
## lbm          0.325392   0.004708   69.11   <2e-16 ***
## ssf          0.050275   0.001131   44.44   <2e-16 ***
## ht          -0.239281   0.006199  -38.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5041 on 198 degrees of freedom
## Multiple R-squared:  0.9695, Adjusted R-squared:  0.969
## F-statistic:  2097 on 3 and 198 DF,  p-value: < 2.2e-16
```

This is the final mode.

(iv) Fit also models using the adjusted R^2 and AIC as criteria. Select a minimal adequate model out of all these procedures. Justify your answer.

For R^2 we use the regsubsets function in the leaps package

```
library(leaps)
a <- regsubsets(bmi ~ ., data = ex3.df)
summary(a)
```

```
## Subset selection object
## Call: regsubsets.formula(bmi ~ ., data = ex3.df)
## 5 Variables  (and intercept)
##     Forced in Forced out
## lbm     FALSE      FALSE
## ssf     FALSE      FALSE
## ht      FALSE      FALSE
## wcc     FALSE      FALSE
## hc      FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          lbm ssf ht  wcc hc
## 1  ( 1 ) "*" " " " " " " " "
## 2  ( 1 ) "*" "*" " " " " " "
## 3  ( 1 ) "*" "*" "*" " " " "
## 4  ( 1 ) "*" "*" "*" " " "*"
## 5  ( 1 ) "*" "*" "*" "*" "*"
```

```
which.max(summary(a)$adjr2)
```

```
## [1] 4
```

The model has `lbm`, `ssf`, `ht`, and `hc`.

For AIC use `stepAIC` in the `MASS` package

```
library(MASS)
stepAIC(lm1)
```

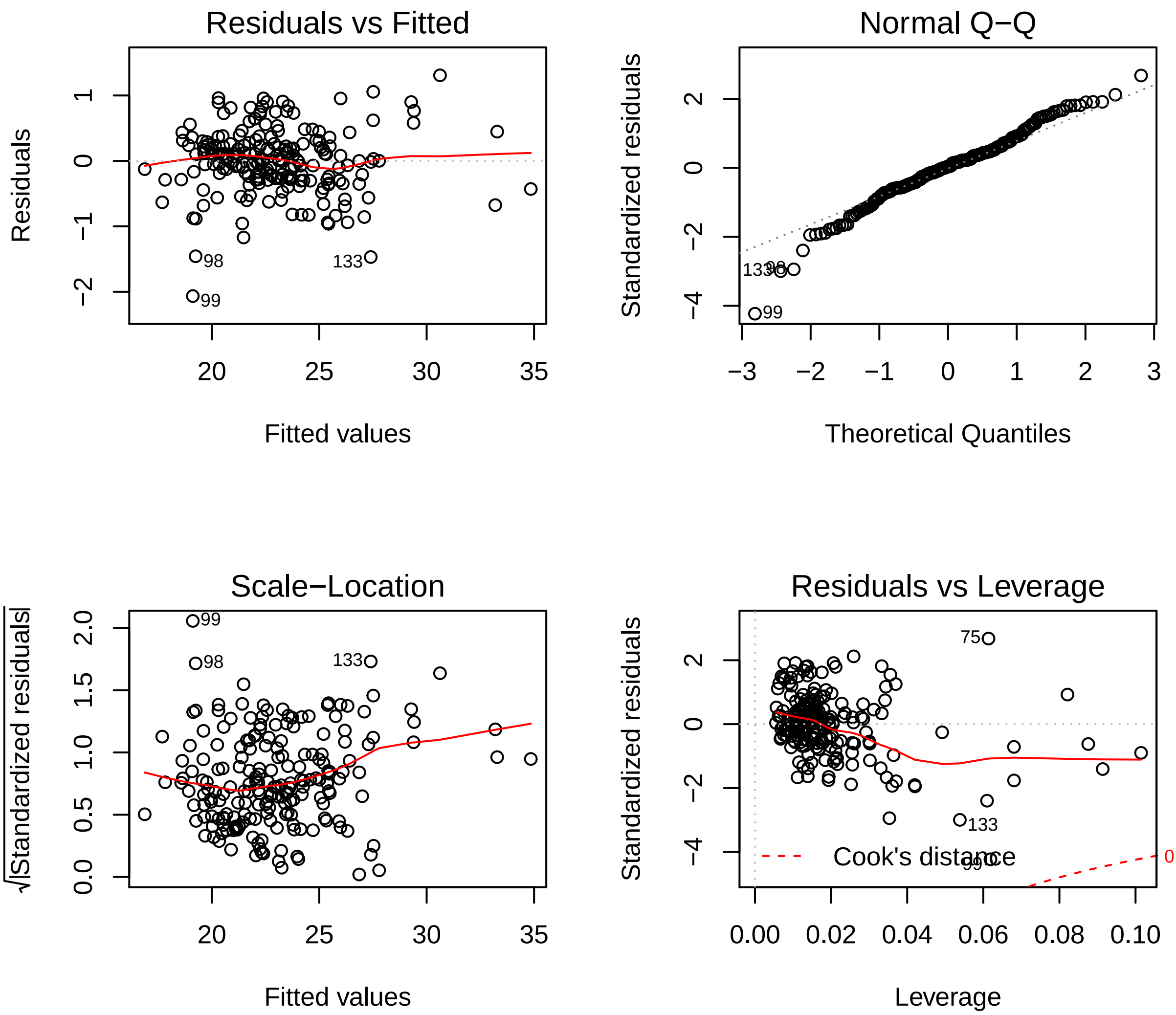
```
## Start:  AIC=-271.06
## bmi ~ lbm + ssf + ht + wcc + hc
## 
##          Df Sum of Sq    RSS     AIC
## - wcc     1      0.05  49.80 -272.86
## - hc      1      0.43  50.18 -271.32
## <none>                 49.75 -271.06
## - ht      1    366.24 415.99  155.92
## - ssf     1    410.23 459.98  176.23
## - lbm     1    931.49 981.24  329.27
## 
## Step:  AIC=-272.86
## bmi ~ lbm + ssf + ht + hc
## 
##          Df Sum of Sq    RSS     AIC
## <none>                 49.80 -272.86
## - hc      1      0.51  50.31 -272.80
## - ht      1    366.32 416.11  153.98
## - ssf     1    435.82 485.62  185.19
## - lbm     1    931.58 981.38  327.30
```

```
## 
## Call:
## lm(formula = bmi ~ lbm + ssf + ht + hc, data = ex3.df)
## 
## Coefficients:
## (Intercept)          lbm          ssf           ht           hc  
##    40.60659      0.32189      0.05096     -0.23798      0.01879
```

Both AIC and R^2 choose the second model, `lm2`. Observe, however, that the change in adjusted R^2 is 0.9692 to 0.969, and the difference in AIC is -272.86 to -272.8, a very small difference in both cases. I would keep the simpler model `lm3`.

(v) Draw the diagnostic plots for your final model and discuss them

```
par(mfrow=c(2,2))
plot(lm3)
```



```
par(mfrow=c(1,1))
```

The quantile plot has departures from the straight line at the lower end and the scale-location plots shows an increasing pattern. We use tests for normality and homoscedasticity

```
shapiro.test(rstandard(lm3))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(lm3)
## W = 0.97783, p-value = 0.002773
```

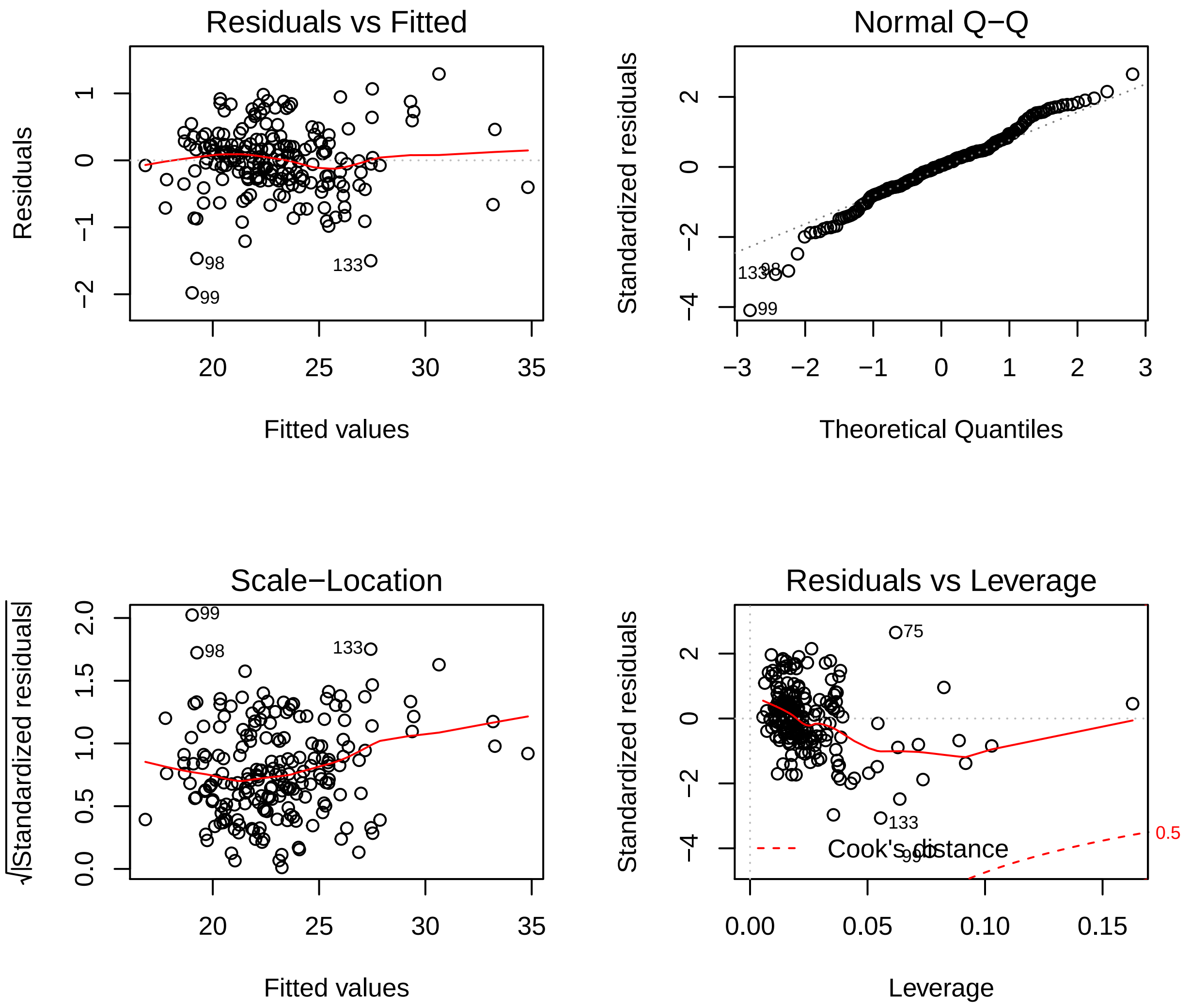
```
ncvTest(lm3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.172076, Df = 1, p = 0.14054
```

The normality test rejects the null hypothesis of a normal distribution, but the homoscedasticity assumption is not rejected.

For comparison, we plot the diagnostic graphs for the other model. The differences are small.

```r
par(mfrow=c(2,2))
plot(lm2)
```



```r
par(mfrow=c(1,1))
```

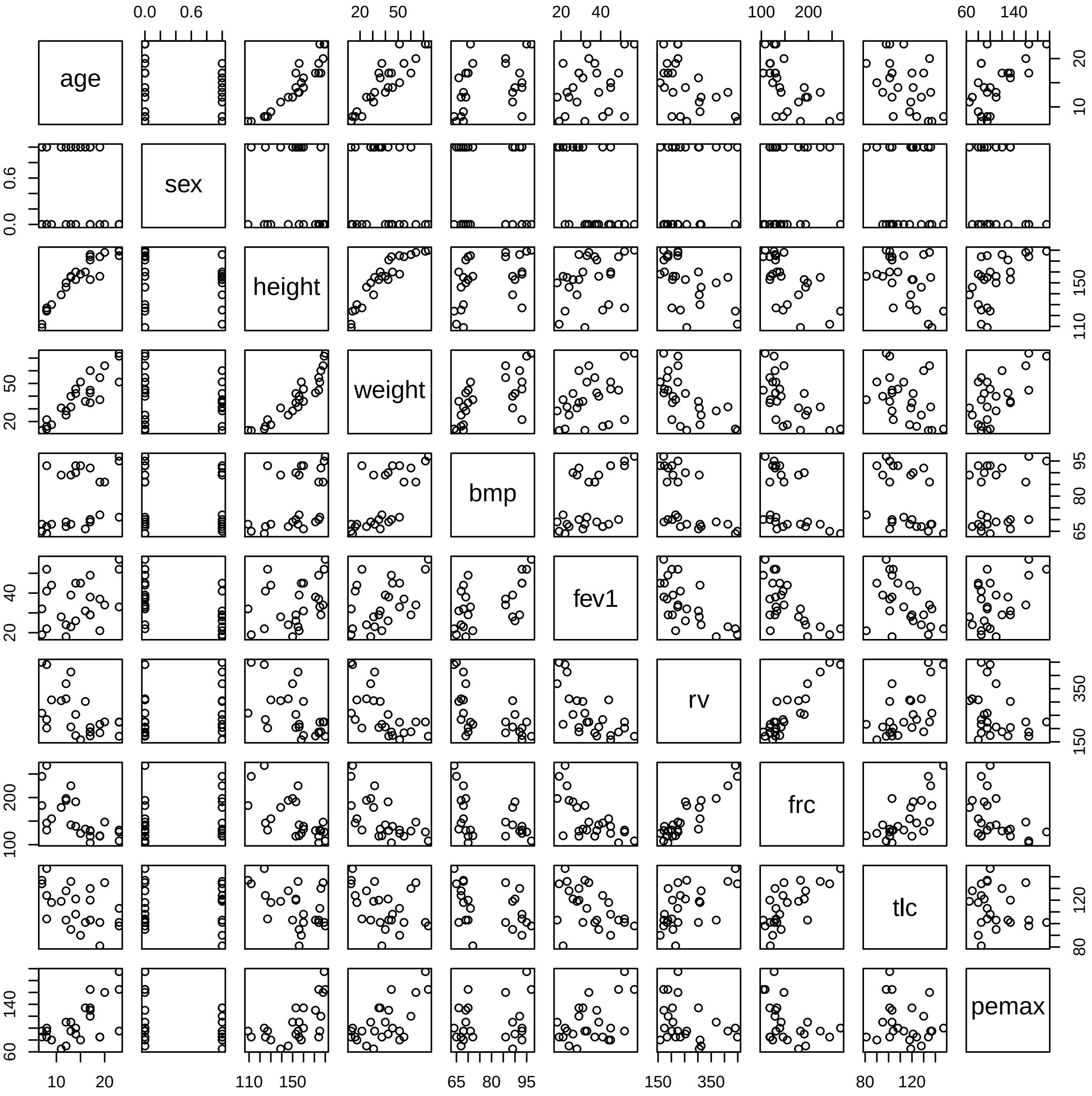
Exercise 4

For this exercise we use the data set `cystfibr` in the package `ISwR`.

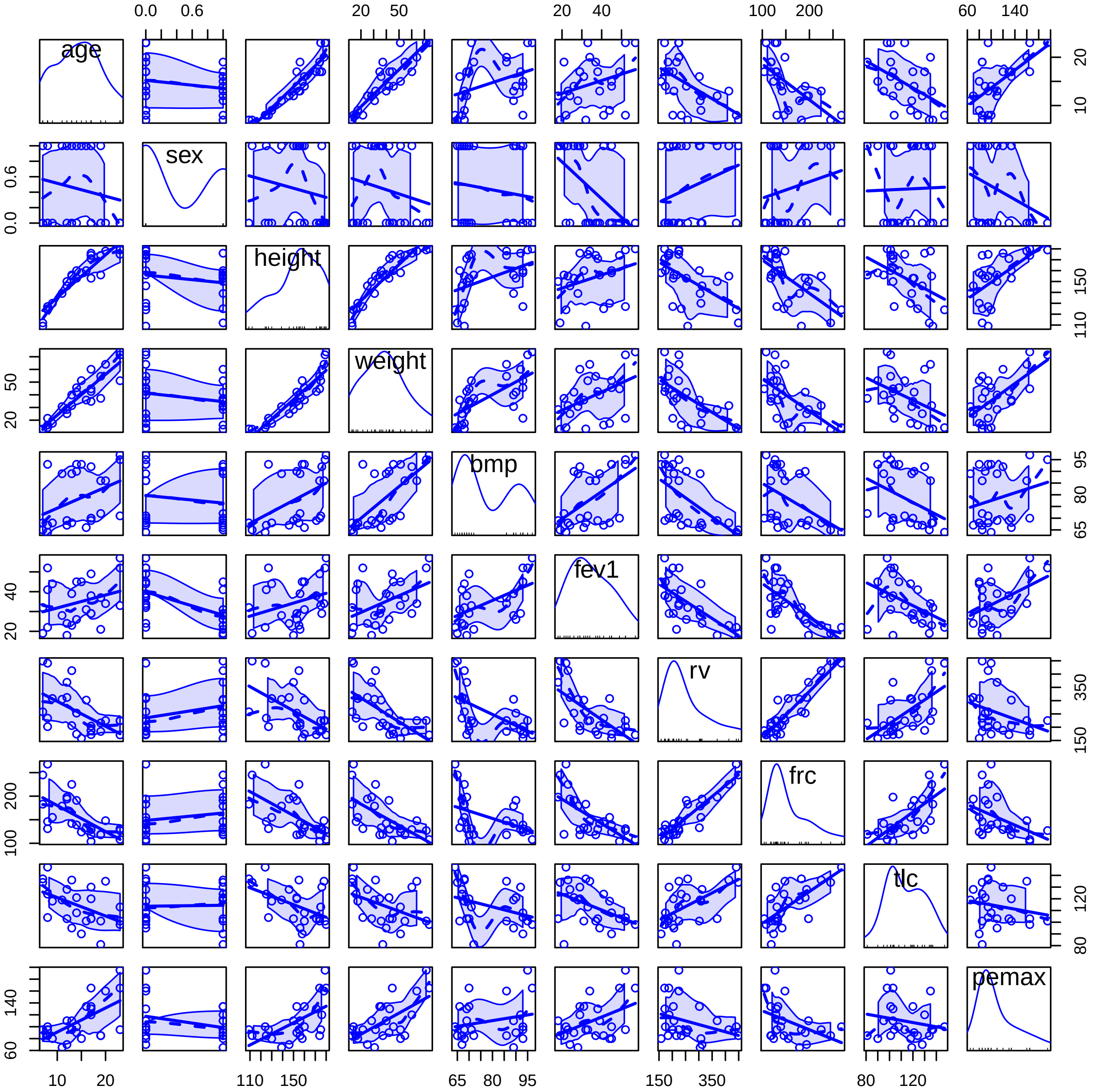
```r
library(ISwR)
library(corrplot)
data(cystfibr)
```

(i) Use the `help` to get familiar with the information contained in this set. Plot a graph matrix of the variables in the set. Use `corrplot.mixed` to get a correlation matrix graph.

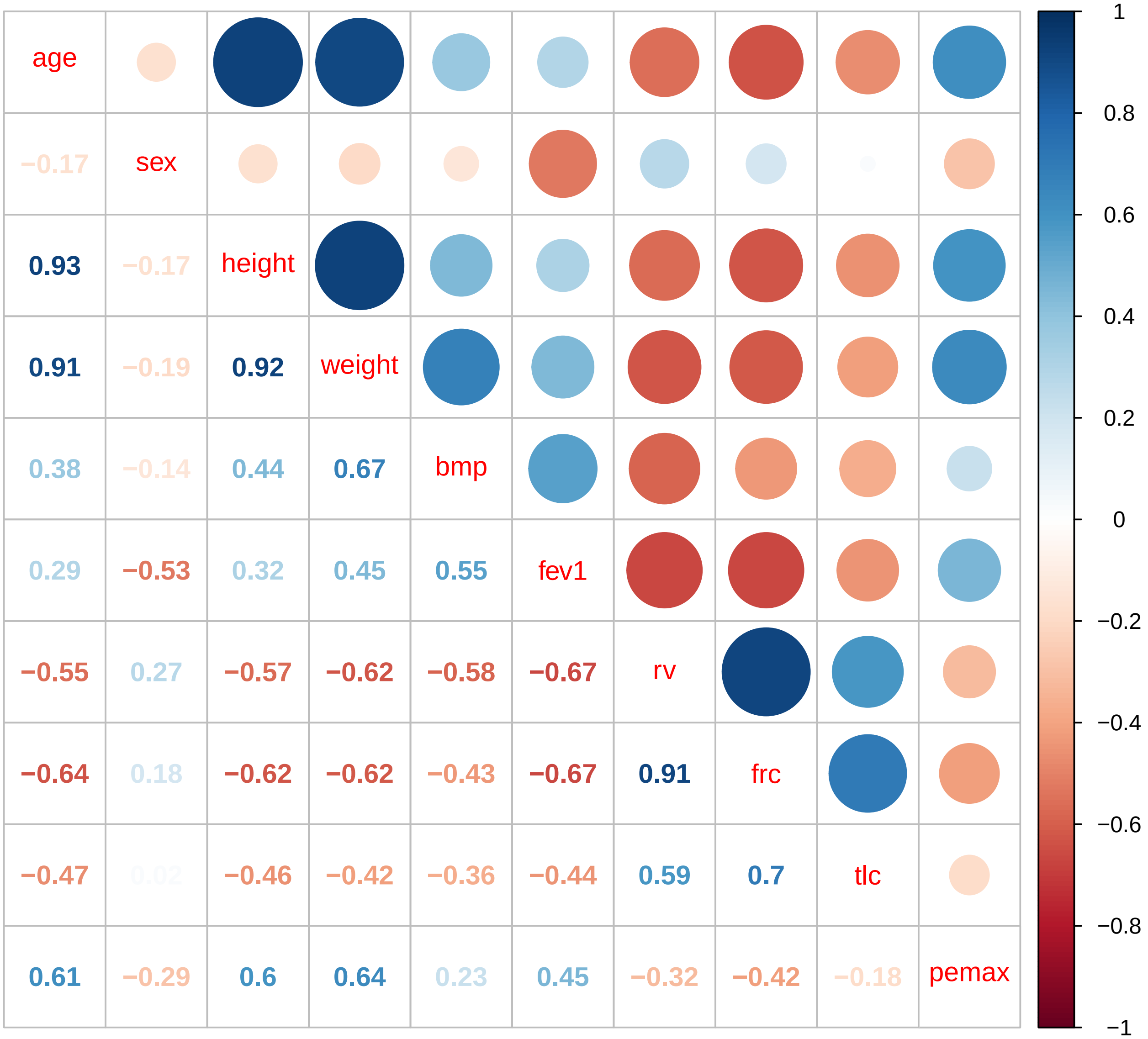
```
pairs(cystfibr)
```



```
scatterplotMatrix(cystfibr)
```



```r
cor.ex4 <- cor(cystfibr)
corrplot.mixed(cor.ex4)
```



(ii) Fit a model of `pemax` as a function of the remaining variables. Place `age` as the first terms in the regression equation (`pemax ~ age + ...`). Obtain a summary table and use `anova` to get an anova table for the regression. Observe that the tests are not the same in both tables. (This will be explained in class).

```
model1 <- lm(pemax ~ ., data = cystfibr)
summary(model1)
```

```
##
## Call:
## lm(formula = pemax ~ ., data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## height       -0.4463     0.9034  -0.494    0.628
## weight        2.9928     2.0080   1.490    0.157
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

```r
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: pemax
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## age        1 10098.5 10098.5 15.5661 0.001296 **
## sex        1   955.4   955.4  1.4727 0.243680
## height     1   155.0   155.0  0.2389 0.632089
## weight     1   632.3   632.3  0.9747 0.339170
## bmp        1  2862.2  2862.2  4.4119 0.053010 .
## fev1       1  1549.1  1549.1  2.3878 0.143120
## rv         1   561.9   561.9  0.8662 0.366757
## frc        1   194.6   194.6  0.2999 0.592007
## tlc        1    92.4    92.4  0.1424 0.711160
## Residuals 15  9731.2   648.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model1)
```

```
## Anova Table (Type II tests)
##
## Response: pemax
##           Sum Sq Df F value Pr(>F)
## age        181.8  1  0.2803 0.6043
## sex         37.9  1  0.0584 0.8123
## height     158.3  1  0.2440 0.6285
## weight    1441.2  1  2.2215 0.1568
## bmp       1480.1  1  2.2815 0.1517
## fev1       648.4  1  0.9995 0.3333
## rv         653.8  1  1.0077 0.3314
## frc        254.6  1  0.3924 0.5405
## tlc         92.4  1  0.1424 0.7112
## Residuals 9731.2 15
```

(iii) The only significant variable seems to be `age`. Fit a model including only this variable and obtain the summary table. To compare the two models, use the `anova` function with argument the name of the two models. Comment on the results.

```r
model2 <- lm(pemax ~ age, data = cystfibr)
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = pemax ~ age, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.666 -17.174   6.209  16.209  51.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.408     16.657   3.026  0.00601 **
## age            4.055      1.088   3.726  0.00111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.97 on 23 degrees of freedom
## Multiple R-squared:  0.3764, Adjusted R-squared:  0.3492
## F-statistic: 13.88 on 1 and 23 DF,  p-value: 0.001109
```

```r
anova(model1, model2)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc +
##     tlc
## Model 2: pemax ~ age
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     15  9731.2
## 2     23 16734.2 -8   -7002.9 1.3493 0.2936
```

(iv) Repeat (ii) and (iii) placing `height` as the first term in the regression equation.

```r
model3 <- lm(pemax ~ height + age + sex + weight + bmp + fev1 + rv + frc + tlc, data = cystfibr)
summary(model3)
```

```
##
## Call:
## lm(formula = pemax ~ height + age + sex + weight + bmp + fev1 +
##     rv + frc + tlc, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## height       -0.4463     0.9034  -0.494    0.628
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## weight        2.9928     2.0080   1.490    0.157
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

```
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: pemax
##           Df Sum Sq Mean Sq F value   Pr(>F)
## height     1 9634.6  9634.6 14.8511 0.001562 **
## age        1  646.2   646.2  0.9960 0.334098
## sex        1  928.1   928.1  1.4305 0.250243
## weight     1  632.3   632.3  0.9747 0.339170
## bmp        1 2862.2  2862.2  4.4119 0.053010 .
## fev1       1 1549.1  1549.1  2.3878 0.143120
## rv         1  561.9   561.9  0.8662 0.366757
## frc        1  194.6   194.6  0.2999 0.592007
## tlc        1   92.4    92.4  0.1424 0.711160
## Residuals 15 9731.2   648.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model4 <- lm(pemax ~ height, data = cystfibr)
summary(model4)
```

```
##
## Call:
## lm(formula = pemax ~ height, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.876 -19.306   1.787  18.170  61.464
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -33.2757    40.0445  -0.831  0.41453
## height        0.9319     0.2596   3.590  0.00155 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.34 on 23 degrees of freedom
## Multiple R-squared:  0.3591, Adjusted R-squared:  0.3312
## F-statistic: 12.89 on 1 and 23 DF,  p-value: 0.001549
```

```
anova(model3, model4)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ height + age + sex + weight + bmp + fev1 + rv + frc +
##     tlc
## Model 2: pemax ~ height
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     15  9731.2
## 2     23 17198.0 -8   -7466.8 1.4387 0.2588
```

(v) Repeat (ii) and (iii) placing `weight` as the first term in the regression equation.

```
model5 <- lm(pemax ~ weight + height + age + sex +bmp + fev1 + rv + frc + tlc, data = cystfibr)
summary(model5)
```

```
##
## Call:
## lm(formula = pemax ~ weight + height + age + sex + bmp + fev1 +
##     rv + frc + tlc, data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## weight        2.9928     2.0080   1.490    0.157
## height       -0.4463     0.9034  -0.494    0.628
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

```
anova(model5)
```

```
## Analysis of Variance Table
##
## Response: pemax
##           Df  Sum Sq Mean Sq F value   Pr(>F)
## weight     1 10827.2 10827.2 16.6893 0.000975 ***
## height     1    36.4    36.4  0.0561 0.815975
## age        1   186.9   186.9  0.2880 0.599351
## sex        1   790.8   790.8  1.2189 0.286964
## bmp        1  2862.2  2862.2  4.4119 0.053010 .
## fev1       1  1549.1  1549.1  2.3878 0.143120
## rv         1   561.9   561.9  0.8662 0.366757
## frc        1   194.6   194.6  0.2999 0.592007
## tlc        1    92.4    92.4  0.1424 0.711160
## Residuals 15  9731.2   648.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model6 <- lm(pemax ~ weight, data = cystfibr)
summary(model6)
```

```
##
```

```
## Call:
## lm(formula = pemax ~ weight, data = cystfibr)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44.30 -22.69   2.23  15.91  48.41
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.5456    12.7016   5.003 4.63e-05 ***
## weight        1.1867     0.3009   3.944 0.000646 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.38 on 23 degrees of freedom
## Multiple R-squared:  0.4035, Adjusted R-squared:  0.3776
## F-statistic: 15.56 on 1 and 23 DF,  p-value: 0.0006457
```

```
anova(model3, model6)
```

```
## Analysis of Variance Table
##
## Model 1: pemax ~ height + age + sex + weight + bmp + fev1 + rv + frc +
##     tlc
## Model 2: pemax ~ weight
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     15  9731.2
## 2     23 16005.5 -8   -6274.2 1.2089 0.3573
```

More on this data set. Use the function `regsubsets` in the `leaps` package.

```
library(leaps)
regfit.full = regsubsets(pemax~.,cystfibr)
summary(regfit.full)
```

```
## Subset selection object
## Call: regsubsets.formula(pemax ~ ., cystfibr)
## 9 Variables  (and intercept)
##        Forced in Forced out
## age        FALSE      FALSE
## sex        FALSE      FALSE
## height     FALSE      FALSE
## weight     FALSE      FALSE
## bmp        FALSE      FALSE
## fev1       FALSE      FALSE
## rv         FALSE      FALSE
## frc        FALSE      FALSE
## tlc        FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          age sex height weight bmp fev1 rv  frc tlc
## 1  ( 1 ) " " " " " "    "*"    " " " "  " " " " " "
## 2  ( 1 ) " " " " " "    "*"    "*" " "  " " " " " "
## 3  ( 1 ) " " " " " "    "*"    "*" "*"  " " " " " "
## 4  ( 1 ) " " " " " "    "*"    "*" "*"  "*" " " " "
## 5  ( 1 ) " " " " " "    "*"    "*" "*"  "*" " " "*"
## 6  ( 1 ) "*" " " "*"    "*"    "*" "*"  "*" " " " "
## 7  ( 1 ) "*" " " "*"    "*"    "*" "*"  "*" "*" " "
## 8  ( 1 ) "*" " " "*"    "*"    "*" "*"  "*" "*" "*"
```

```
regfit.full=regsubsets(pemax~.,data = cystfibr,nvmax = 19)
reg.summary=summary(regfit.full)
names(reg.summary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

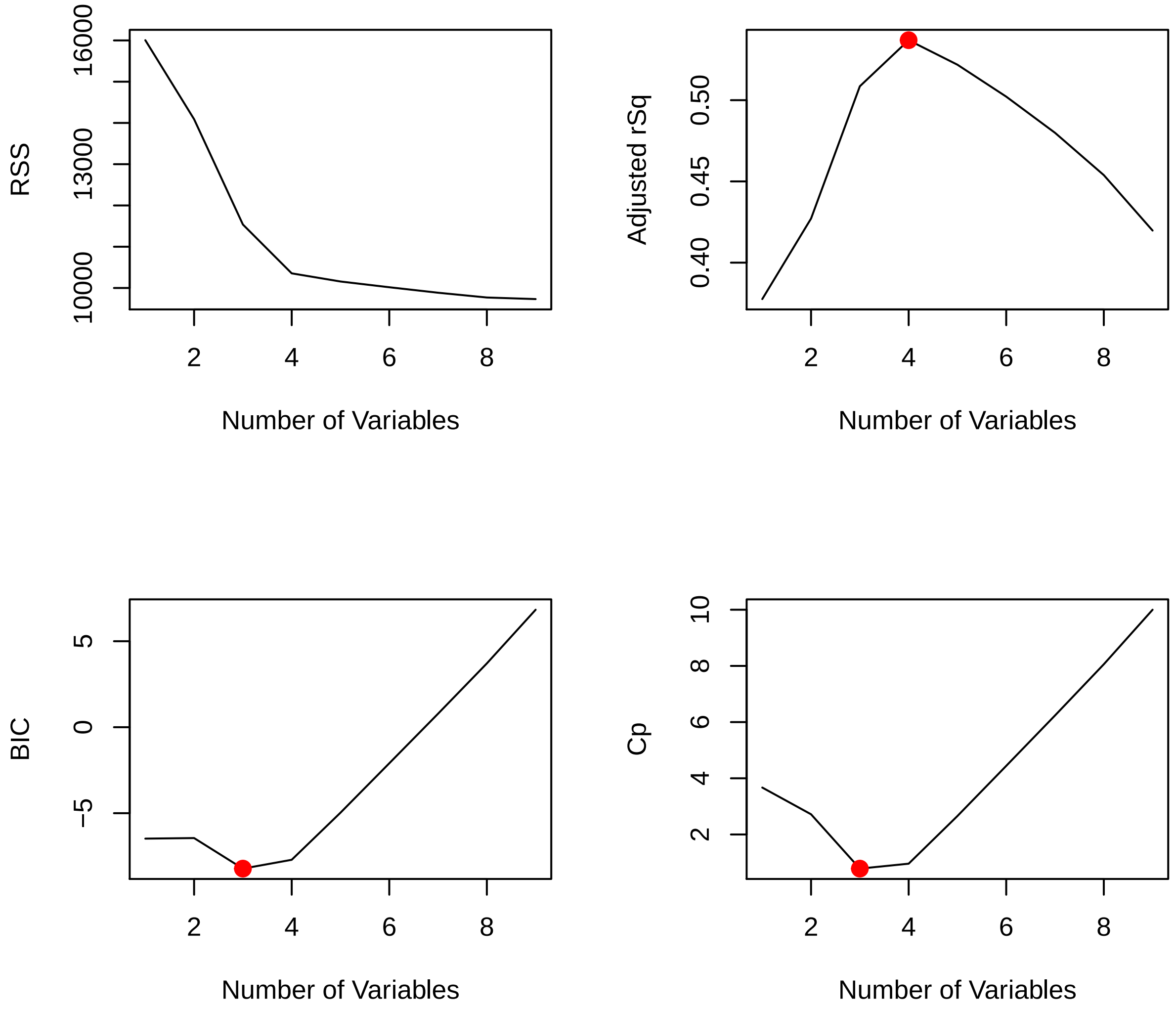
```
reg.summary$rsq
```

```
## [1] 0.4035070 0.4748731 0.5699943 0.6141043 0.6214494 0.6266394 0.6316019
## [8] 0.6359228 0.6373354
```

```
par(mfrow=c(2,2))
plot(reg.summary$rss,xlab = "Number of Variables",ylab = "RSS",type = "l")
plot(reg.summary$adjr2,xlab = "Number of Variables", ylab = "Adjusted rSq",type = "l")
which.max(reg.summary$adjr2)
```

```
## [1] 4
```

```
points(4,reg.summary$adjr2[4],col="red",cex=2,pch=20)
plot(reg.summary$bic,xlab = "Number of Variables", ylab = "BIC",type = "l")
points(3,reg.summary$bic[3],col="red",cex=2,pch=20)
plot(reg.summary$cp,xlab = "Number of Variables", ylab = "Cp",type = "l")
points(3,reg.summary$cp[3],col="red",cex=2,pch=20)
```



```
model <- lm(cystfibr$pemax~.,data = cystfibr)
summary(model)
```

```
##
## Call:
## lm(formula = cystfibr$pemax ~ ., data = cystfibr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.338 -11.532   1.081  13.386  33.405
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 176.0582   225.8912   0.779    0.448
## age          -2.5420     4.8017  -0.529    0.604
## sex          -3.7368    15.4598  -0.242    0.812
## height       -0.4463     0.9034  -0.494    0.628
## weight        2.9928     2.0080   1.490    0.157
## bmp          -1.7449     1.1552  -1.510    0.152
## fev1          1.0807     1.0809   1.000    0.333
## rv            0.1970     0.1962   1.004    0.331
## frc          -0.3084     0.4924  -0.626    0.540
## tlc           0.1886     0.4997   0.377    0.711
##
## Residual standard error: 25.47 on 15 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.4197
## F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

We will explore the problem of variable selection in next week's videos.