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

Problem 1

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

```
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 ...
```

```
q1_data <- subset(ais,select = c('bmi','lbm','ssf','ht','wcc','hc'))
str(q1_data)
```

```
## '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.

```
library(car)
```

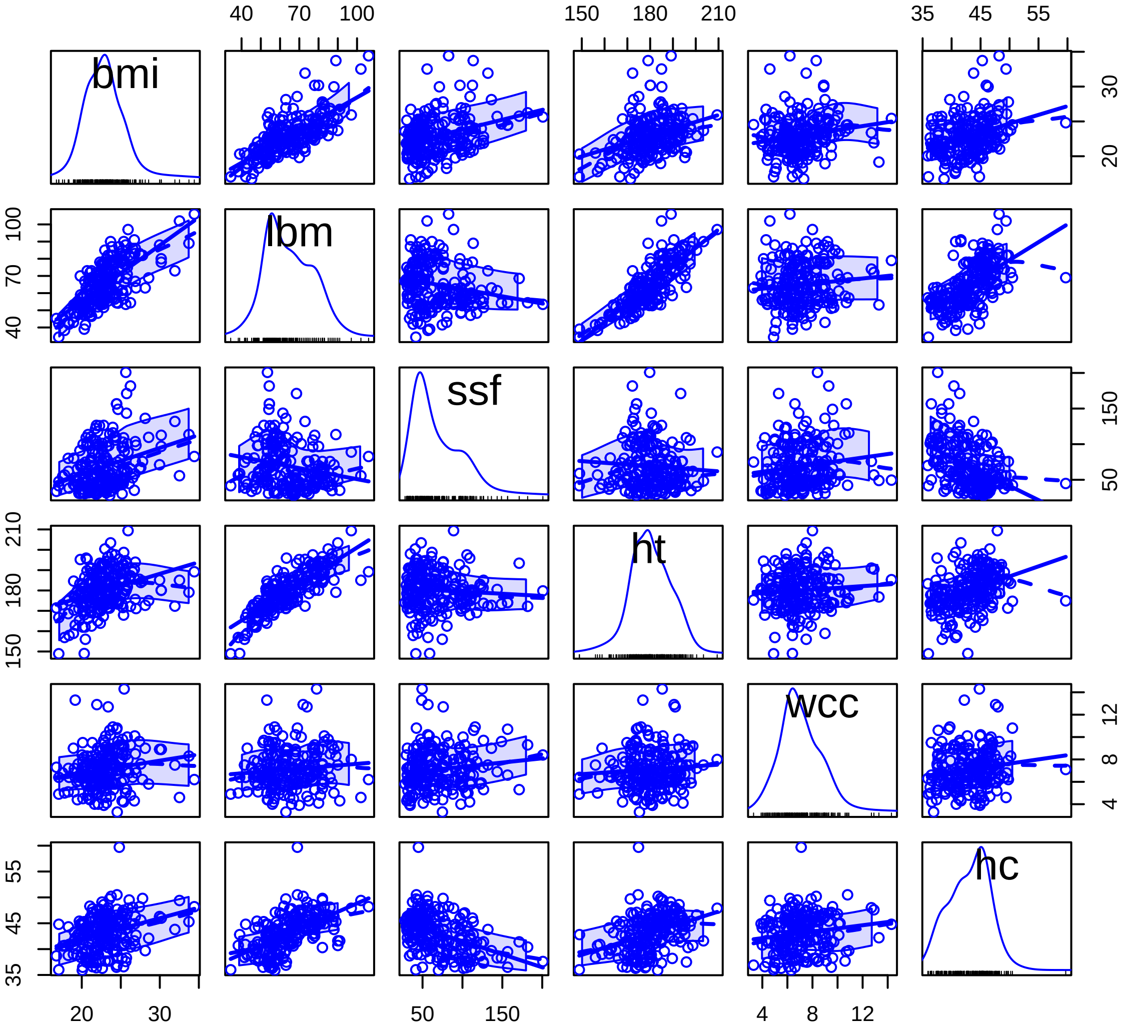
```
## Loading required package: carData
```

```
##
```

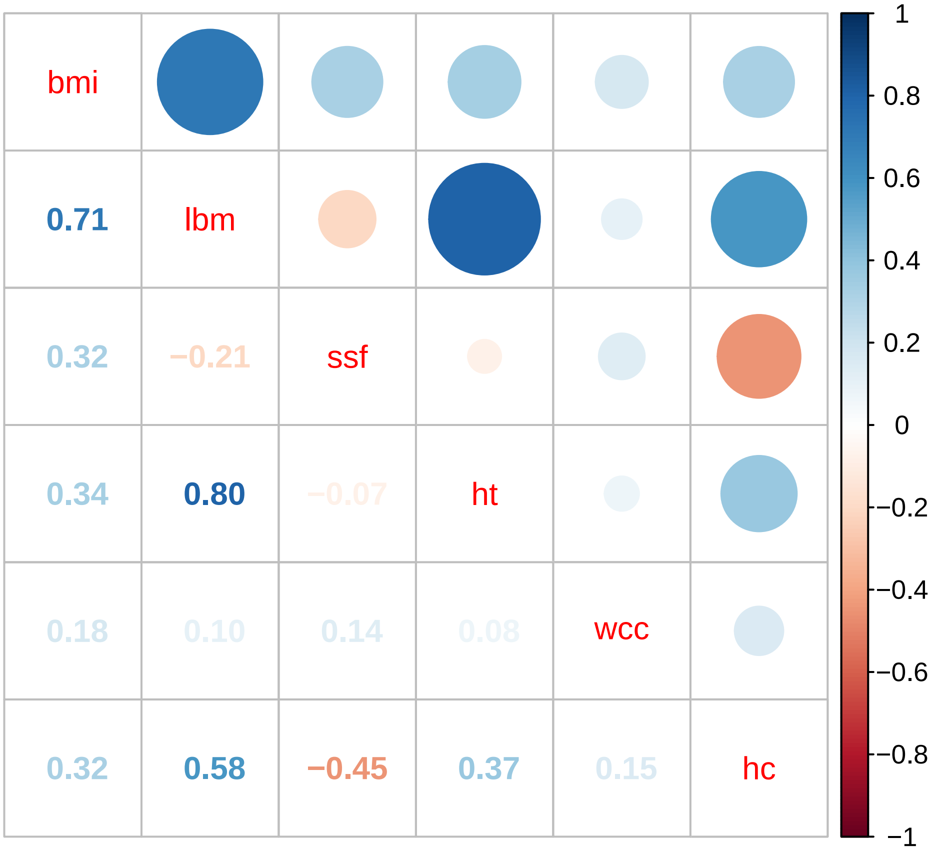
```
## Attaching package: 'car'

## The following object is masked from 'package:DAAG':
##
##     vif
```

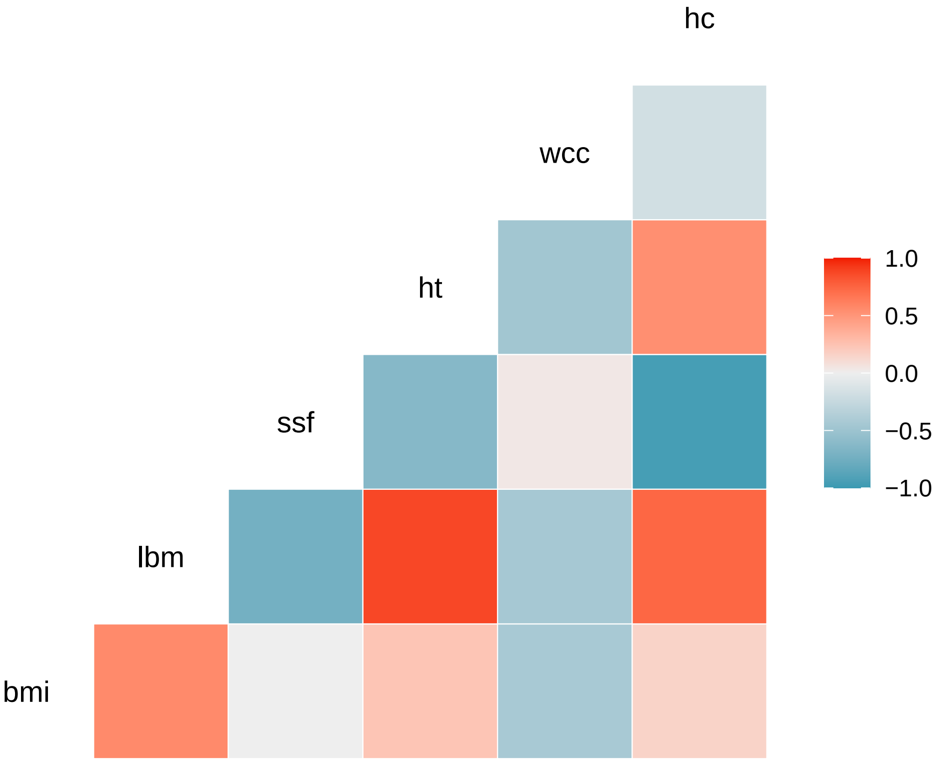
```
scatterplotMatrix(q1_data)
```



```
cor.ex3 <- cor(q1_data)
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 = q1_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = bmi ~ ., data = q1_data)
##
## 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 = q1_data)
##
## 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 = q1_data)
##
## 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 = q1_data)
summary(a)
```

```
## Subset selection object
## Call: regsubsets.formula(bmi ~ ., data = q1_data)
## 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)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:DAAG':
##
##     hills
```

```
stepAIC(lm1)
```

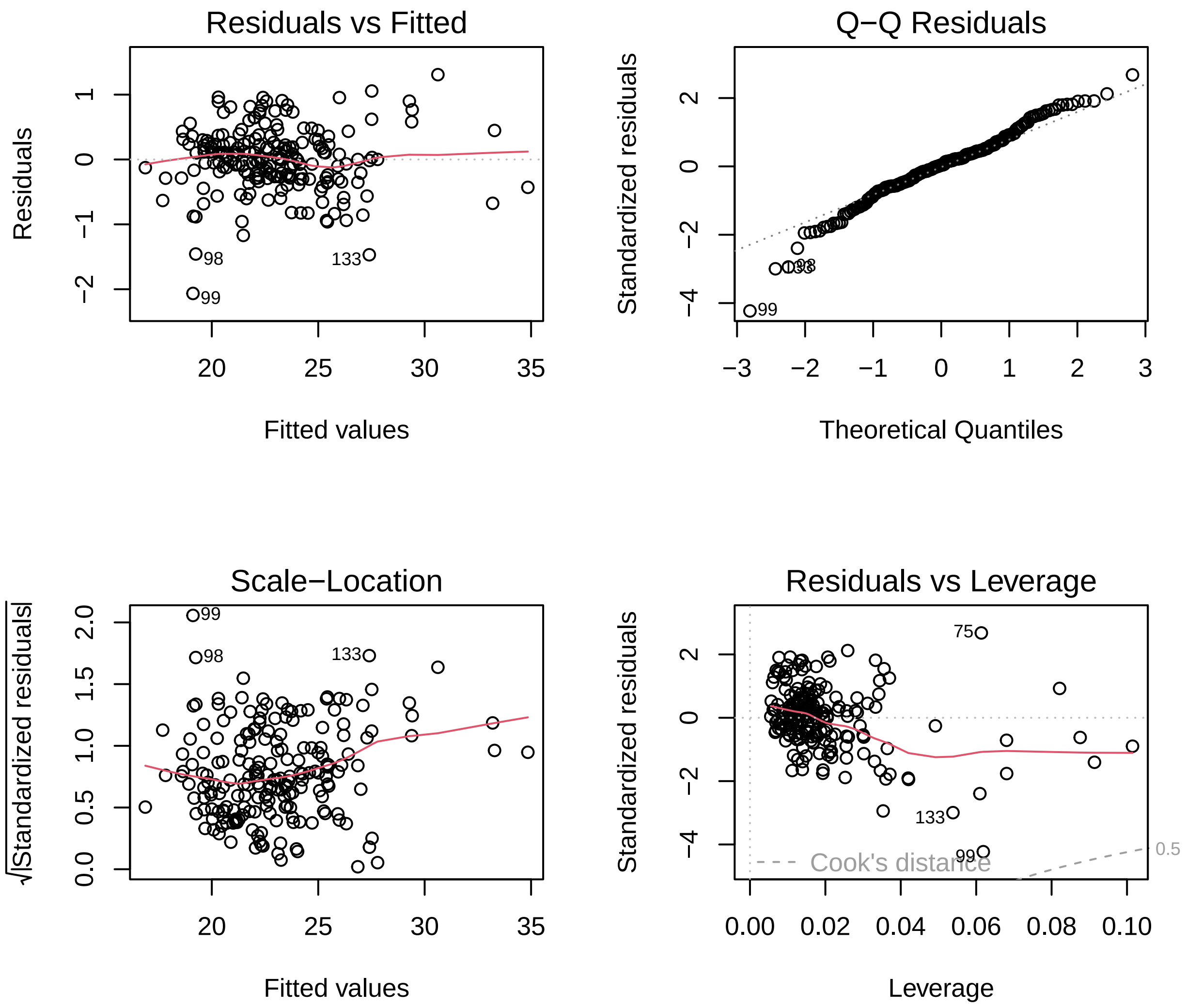
```
## Start:  AIC=-271.06
## bmi ~ lbm + ssf + ht + wcc + hc
##
##        Df Sum of Sq    RSS     AIC
## - wcc   1      0.05  49.80 -272.86
## - hc    1      0.43  50.18 -271.32
## <none>               49.75 -271.06
## - ht    1    366.24 415.99  155.92
## - ssf   1    410.23 459.98  176.23
## - lbm   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 = q1_data)
##
## 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
```

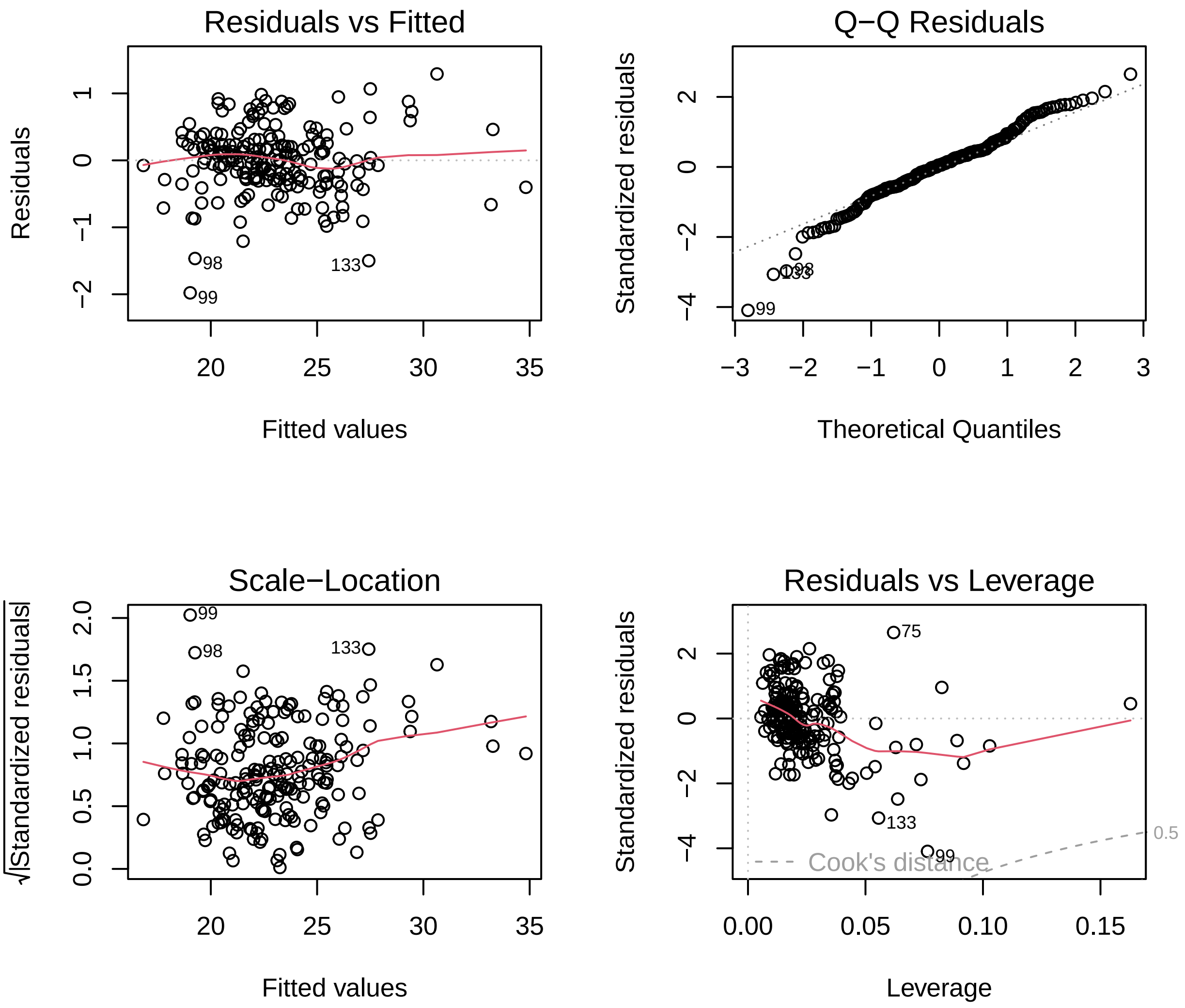
```
ncvTest(lm3)
```

```
## 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))
```

Problem 2

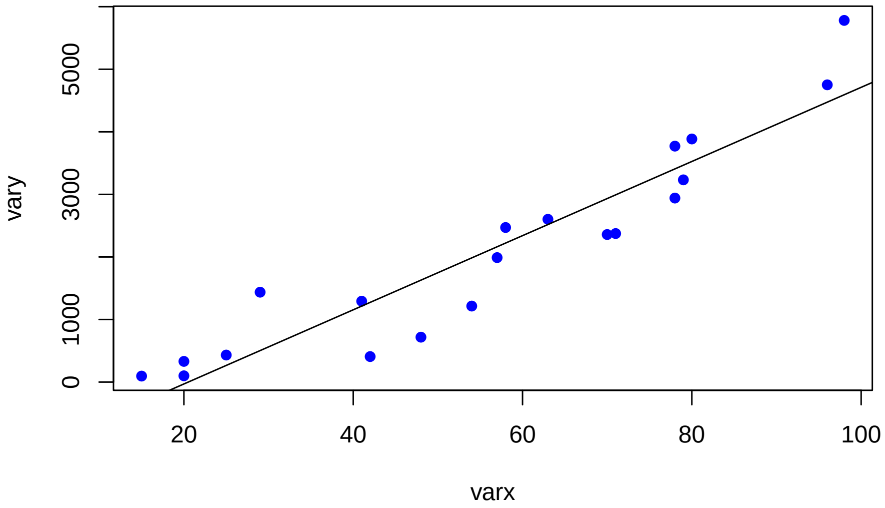
For this question use the data set PL925FPr2.csv.

(i) Read the data and plot vary as a function of varx. Fit a simple linear regression for vary as a function of varx and add the regression line to the plot. Comment. Obtain a summary for the regression and draw the diagnostic plots. Comment on the results.

```
data2 <- read.csv('PL925FPr2.csv')
str(data2)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ varx: int  54 20 25 48 78 58 70 29 98 57 ...
##  $ vary: num  1215 331 432 718 3771 ...
```

```
plot(vary ~ varx, data = data2, pch = 16, col = 'blue')
model1 <- lm(vary ~ varx, data = data2)
abline(model1)
```



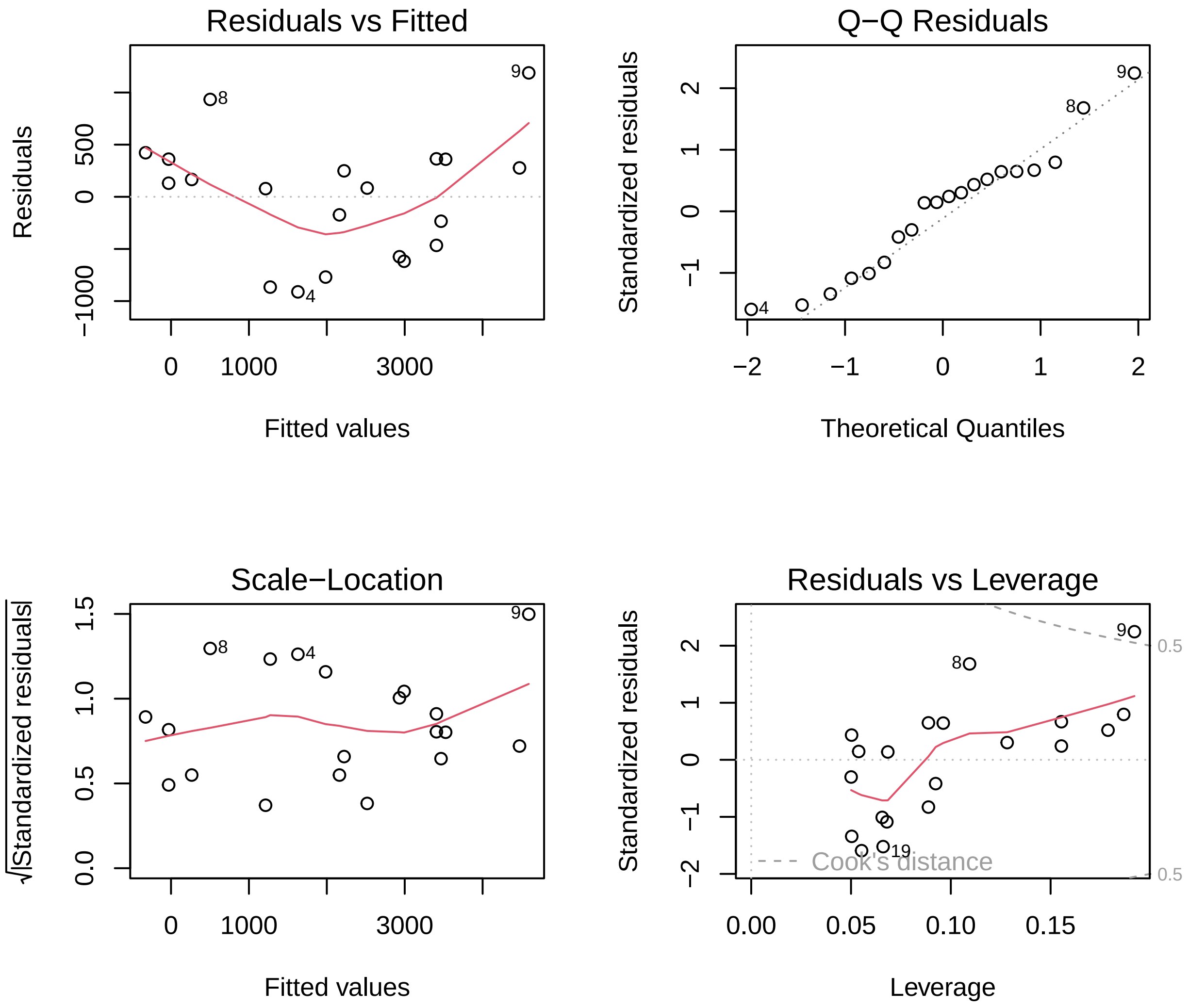
The fit of the regression line to the points is not very good. We can see that most of the points in the central part of the plot are below the line, while those at the extremes are mainly above.

```
summary(model1)
```

```
##
## Call:
## lm(formula = vary ~ varx, data = data2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -911.5 -493.1  107.0  360.2 1188.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1215.752    324.871  -3.742  0.00149 **
## varx           59.267      5.294  11.194 1.53e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 588.7 on 18 degrees of freedom
## Multiple R-squared:  0.8744, Adjusted R-squared:  0.8674
## F-statistic: 125.3 on 1 and 18 DF,  p-value: 1.532e-09
```

In the output, we see the values are not symmetric in the summary for the residuals, and the median is not close to zero. This points to a non-gaussian distribution for the residuals. This is also clear on the diagnostic plots below. On the other hand, both parameters have small p-values (they are significantly different from zero) and the model has a high R^2 value of 0.87.

```
par(mfrow=c(2,2))
plot(model1)
```



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

The first plot shows a clear U pattern in the residuals, with negative values in the center and positive values at the extremes. The quantile plot has a good fit in general, with some points at the lower extreme away from the reference line. We can check normality using the Shapiro-Wilk test on the standardized residuals:

```
shapiro.test(rstandard(model1))
```

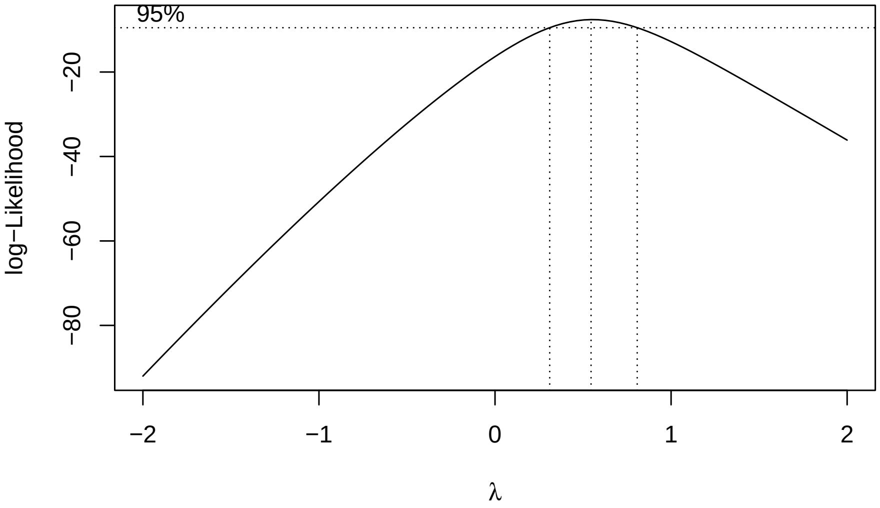
```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model1)
## W = 0.95332, p-value = 0.4203
```

The test does not reject the null hypothesis of normality.

The scale-location plot shows an oscillating pattern, although the evidence is not completely clear.

(ii) Now we are going to use the Box-Cox transformations to improve the model. To simplify this problem, you have to choose between two transformations of the output variable, a square root or a logarithm. Use the function `boxcox` on the package `MASS` with the argument set to the model you fitted in (i). If the confidence interval in the graph includes zero, choose a logarithmic transformation for `vary`. If the confidence interval in the graph includes 0.5 then choose a square root transformation.

```
library(MASS)
boxcox(model1)
```



The confidence interval includes the point 1/2 but not zero, so we choose a square root transformation.

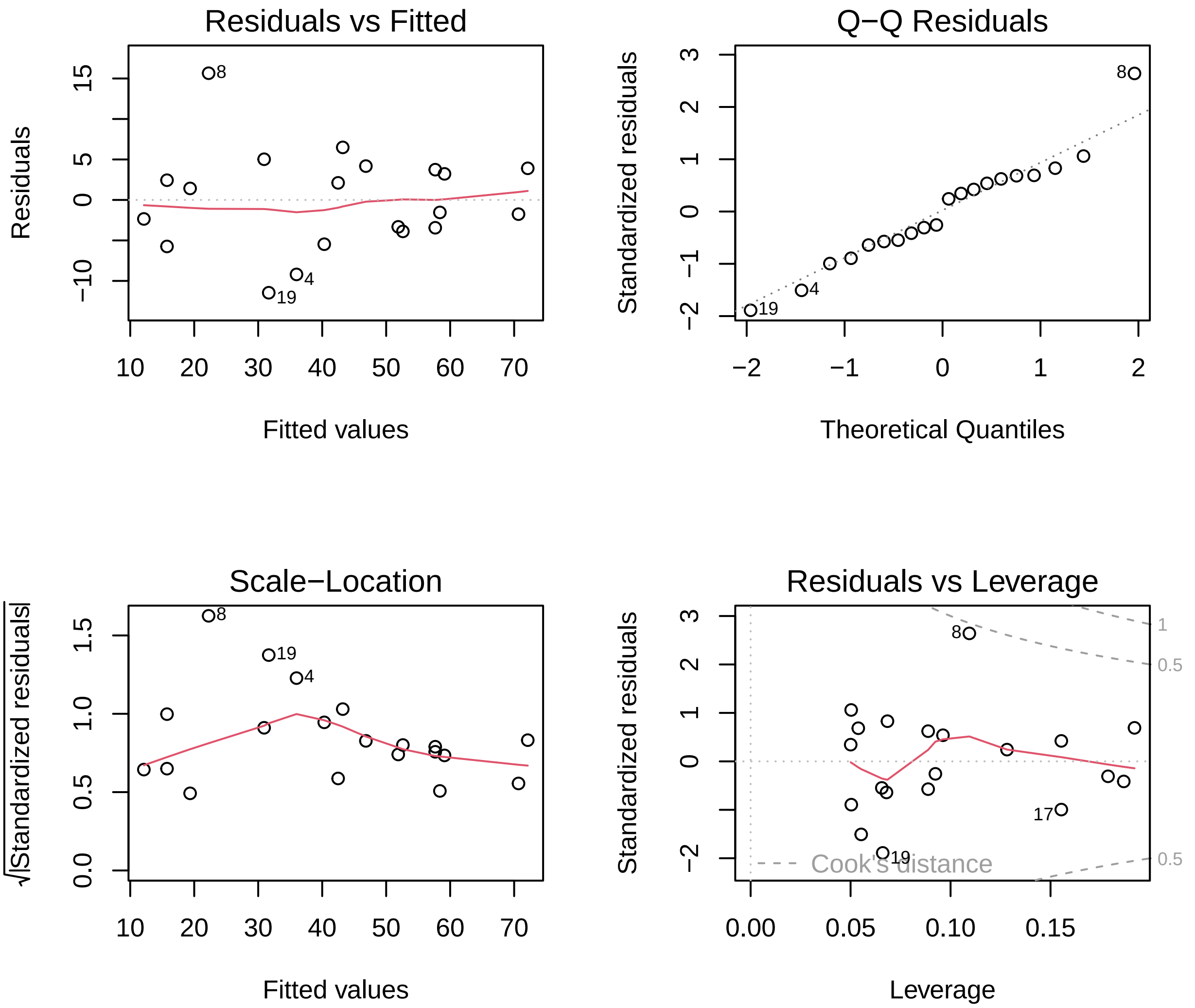
(iii) Fit a new model with the transformation that you choose in (ii). Obtain a summary of the new regression and compare with the previous one. Draw the diagnostic plots and compare with the previous results.

```
model2 <- lm(sqrt(vary) ~ varx, data = data2)
summary(model2)
```

```
##
## Call:
## lm(formula = sqrt(vary) ~ varx, data = data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.463  -3.550  -0.060   3.779  15.649
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.29074    3.46478   0.373    0.714
## varx         0.72282    0.05646  12.801 1.77e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.278 on 18 degrees of freedom
## Multiple R-squared:  0.901,  Adjusted R-squared:  0.8955
## F-statistic: 163.9 on 1 and 18 DF,  p-value: 1.77e-10
```

The quartile values for the residuals are more symmetric now, and the median is close to zero. The parameters for the model are, of course, very different since we have transformed one of the variables, but now only the slope has a small p-value. The intercept is not significantly different from zero according to the t-test. The R^2 has increased to 0.9.

```
par(mfrow=c(2,2))
plot(model2)
```



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

The plots are now much better. The residuals vs fitted plot shows no significant patters, and the residuals look symmetrically distributed. The quantile plot is good, the scale-location shows an increasing-decreasing pattern, but the variation is not large. The fourth plot is similar for both models. We check normality with the Shapiro-Wilk test

```
shapiro.test(rstandard(model2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model2)
## W = 0.95936, p-value = 0.531
```

(iv) Write down the final model in terms of the original variables. What would be the predicted `yval` for a point with `xval` = 45? Draw a scatterplot of `yval` against `xval` and add the regression line for the first model and the curve you obtained with the second regression.

The second model is

$$\sqrt{vary} = 1.291 + 0.72282 \times varx$$

Squaring both sides we get

$$\begin{aligned} vary &= \Big(1.29 + 0.7228 \times varx\Big)^2 \\ &= 1.6641 + 1.8663 \times varx + 0.5225 \times (varx)^2 \end{aligned}$$

This is the equation of the model in terms of the original variables. The predicted value for $varx = 45$ is

```
1.6641 + 1.8663*45 + 0.5225*(45^2)
```

```
## [1] 1143.71
```

This can also be obtained with the function `predict`:

```
new_data = data.frame(varx = 45)
(pr_sqy = predict(model2, new_data))
```

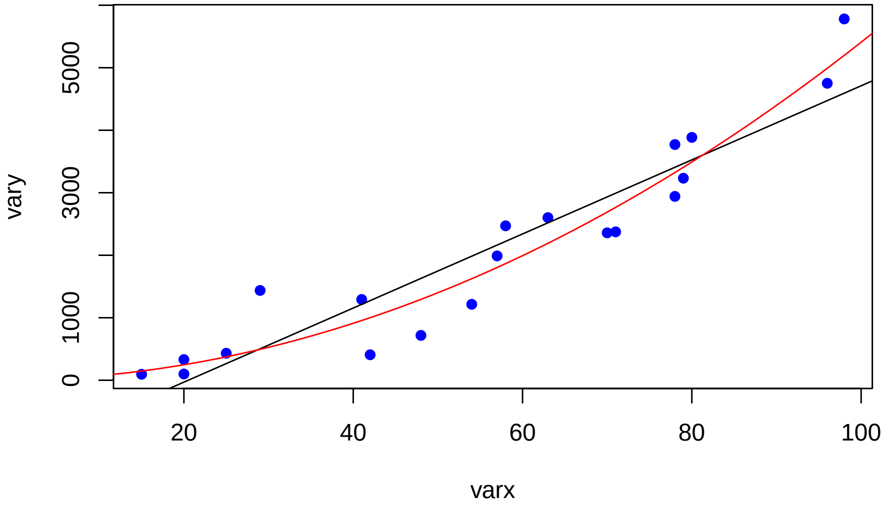
```
##        1
## 33.81746
```

```
(pr_sqy)^2
```

```
##        1
## 1143.621
```

The difference is due to rounding off errors. We now plot the data and the lines/curves for the two models.

```
plot(vary ~ varx, data = data2, pch = 16, col = 'blue')
abline(model1)
curve(1.664 + 1.865*x + 0.522*(x^2), 10, 105, add = T, col = 'red')
```



Problem 3

The data for this question is stored in the file PL925FPr3.csv. It has two variables, yvar and xvar. We want to look at yvar as a function of xvar.

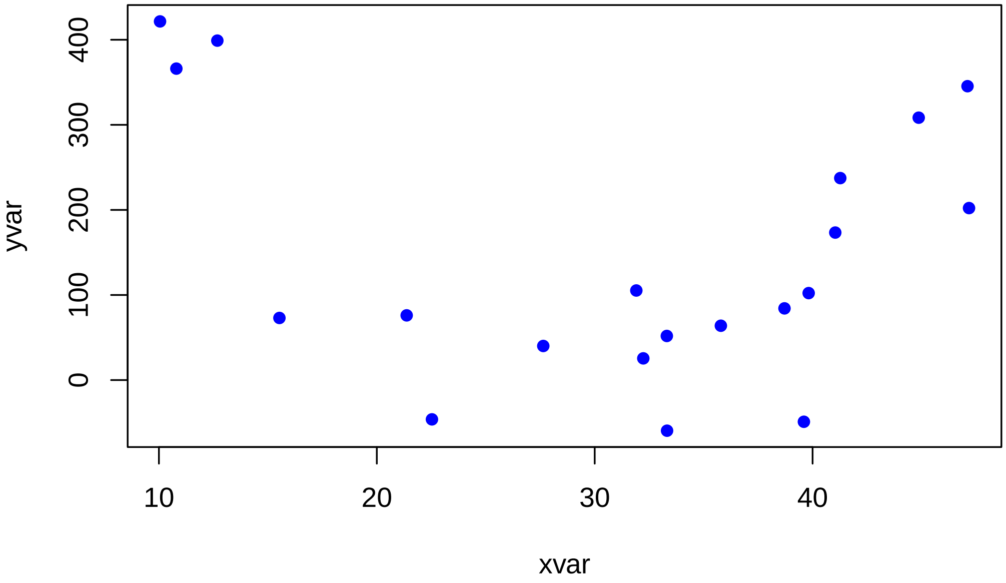
Read the data:

```
q3_data <- read.csv('PL925FPr3.csv')
str(q3_data)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ yvar: num  102 237 366 173 399 ...
##  $ xvar: num  39.8 41.3 10.8 41 12.7 ...
```

(a) Draw a graph. Do these variables seem to be linearly related? Calculate the correlation.

```
plot(yvar~xvar, data = q3_data, pch = 16, col = 'blue')
```



They do not seem to be linearly related. The correlation is

```
cor(q3_data$yvar, q3_data$xvar)
```

```
## [1] -0.2084462
```

which is small.

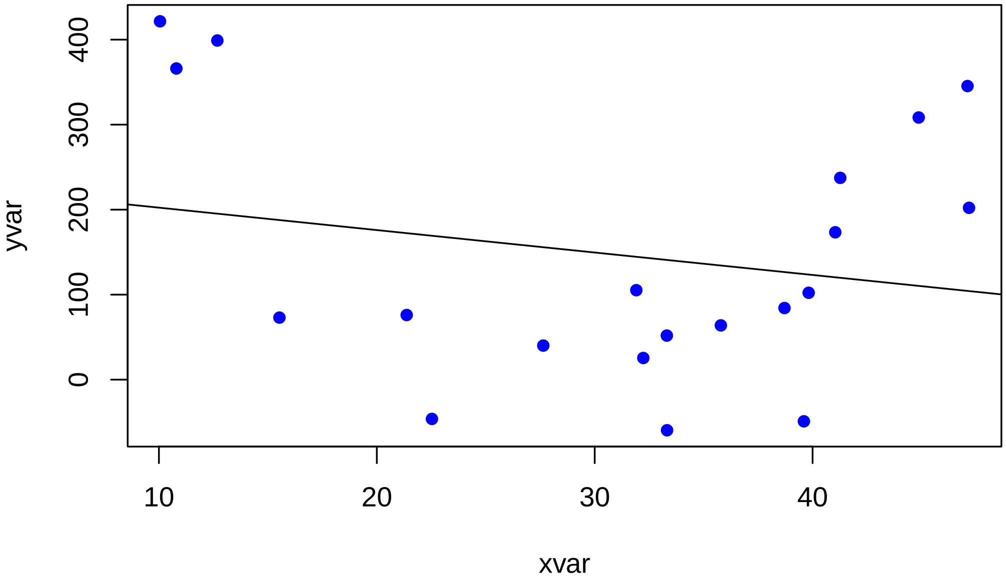
(b) Fit a linear model and comment on the results. What is the R^2?

```
modd1 <- lm(yvar ~ xvar, data = q3_data)
summary(modd1)
```

```
##
## Call:
## lm(formula = yvar ~ xvar, data = q3_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -215.47 -115.01  -40.78  129.64  241.01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  228.758     97.697   2.342   0.0309 *
## xvar          -2.639      2.919  -0.904   0.3778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 153.4 on 18 degrees of freedom
## Multiple R-squared:  0.04345,    Adjusted R-squared:  -0.009692
## F-statistic: 0.8176 on 1 and 18 DF,  p-value: 0.3778
```

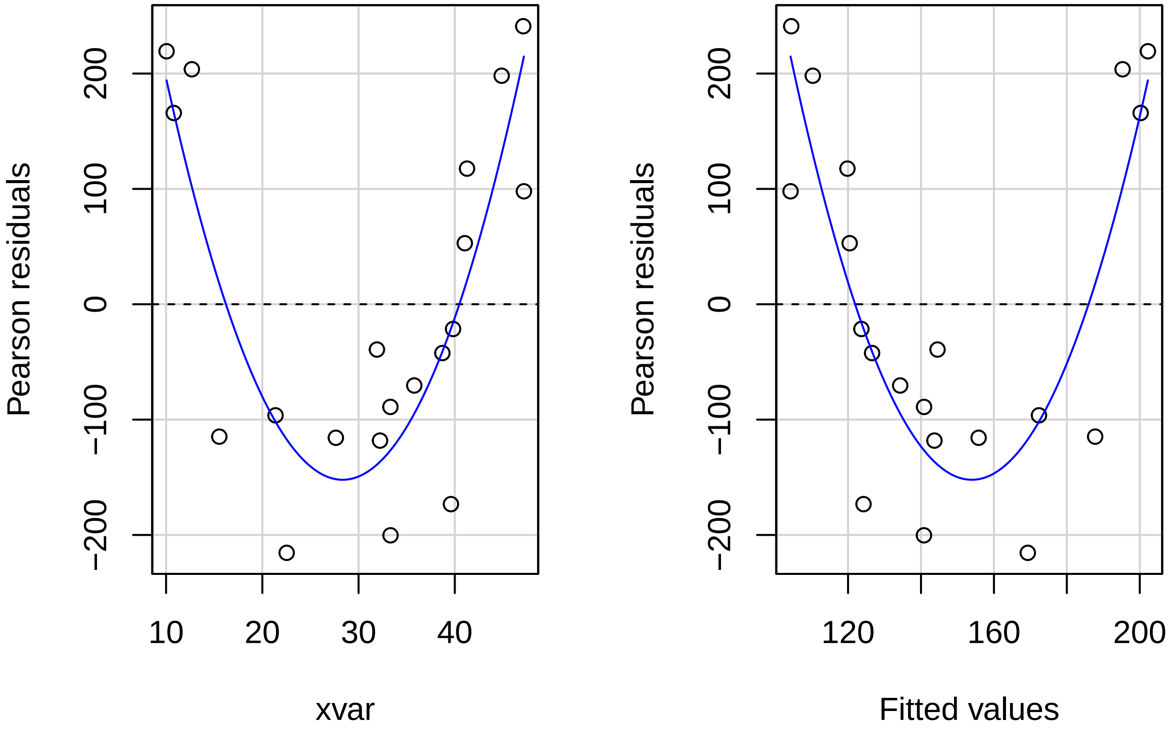
The R^2 is very small, 0.04345

```
plot(yvar~xvar, data = q3_data, pch = 16, col = 'blue')
abline(modd1)
```



(c) Draw residual plots using the function `residualPlots` in the `car` package. Comment on your results.

```
library(car)
residualPlots(modd1)
```



```
##            Test stat Pr(>|Test stat|)
## xvar          6.9428        2.375e-06 ***
## Tukey test    6.9428        3.843e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values for the tests are all small and suggest adding a quadratic term to the regression model.

(d) Fit a quadratic model to these data and comment on the results. What are the R^2 and adjusted R^2?

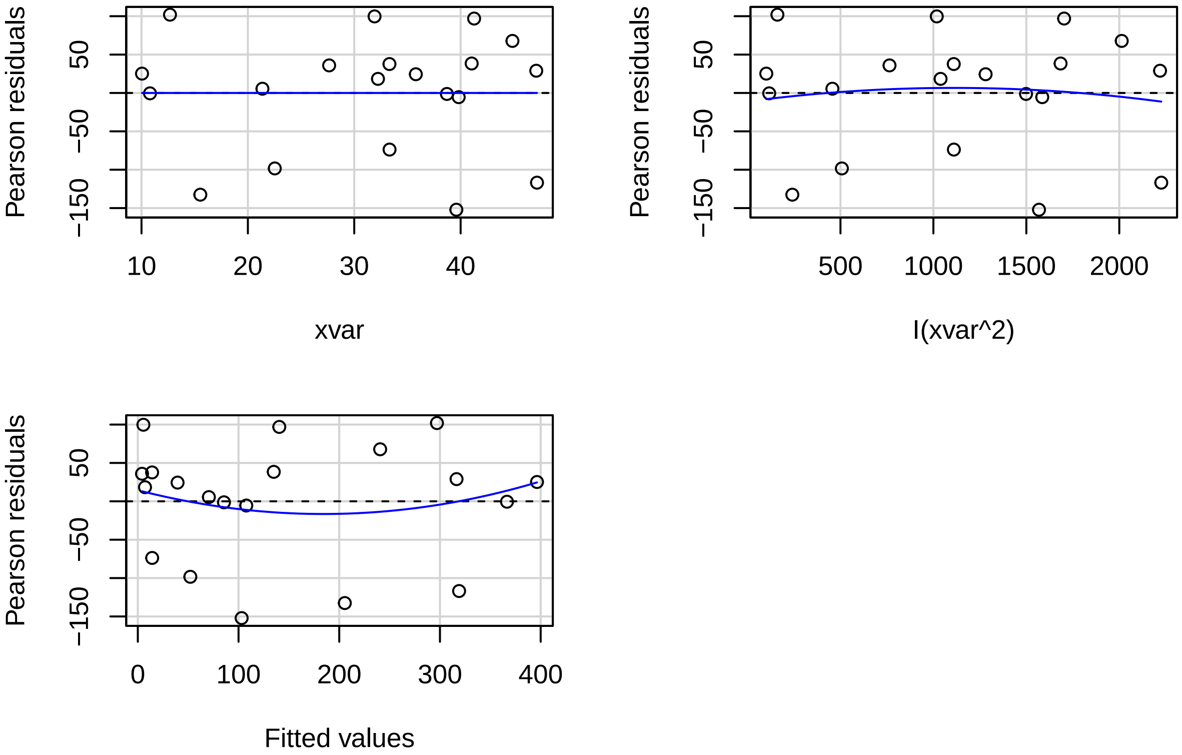
```
modd2 <- lm(yvar ~ xvar + I(xvar^2), data = q3_data)
summary(modd2)
```

```
## 
## Call:
## lm(formula = yvar ~ xvar + I(xvar^2), data = q3_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -152.12  -22.55   21.38   37.86  101.97 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  907.791    110.456   8.219 2.52e-07 ***
## xvar         -61.285      8.585  -7.139 1.66e-06 ***
## I(xvar^2)      1.034      0.149   6.943 2.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 80.61 on 17 degrees of freedom
## Multiple R-squared:  0.7506, Adjusted R-squared:  0.7213 
## F-statistic: 25.58 on 2 and 17 DF,  p-value: 7.474e-06
```

The R^2 and adjusted R^2 have now increased to 0.75 and 0.72, respectively.

(e) Draw residual plots using the `car` function `residualPlots`. Comment on your results.

```
residualPlots(modd2)
```



```
##              Test stat Pr(>|Test stat|)
## xvar           -0.6789           0.5069
## I(xvar^2)      -1.0718           0.2997
## Tukey test      0.6964           0.4862
```

All the plots are satisfactory and the tests all have large p-values, indicating that further higher order terms may not ne necessary.

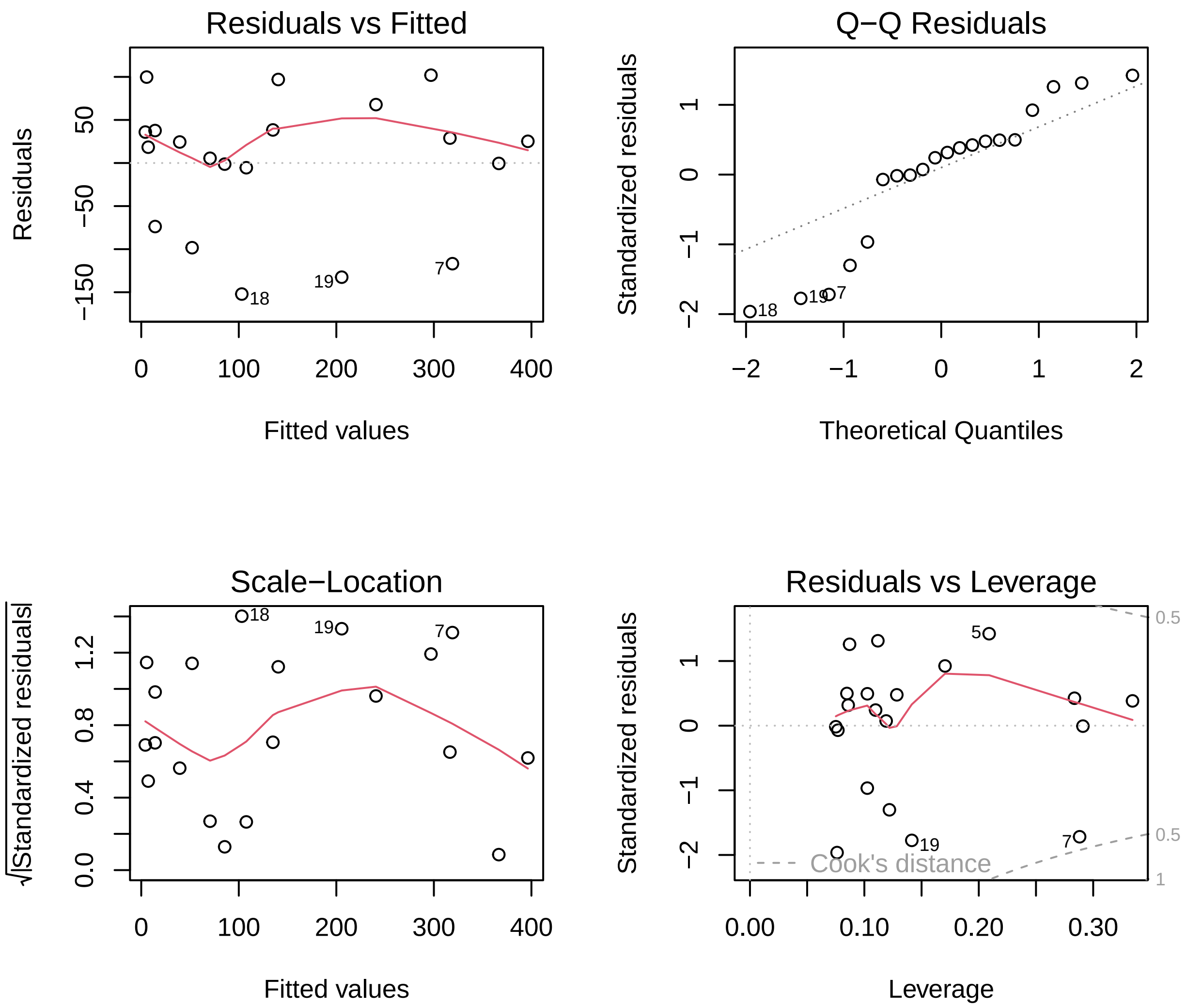
(f) Fit a cubic model to these data and comment on the results. What are the R^2 and adjusted R^2? Is this a better model than the quadratic model? Give reasons for your answer.

```
modd3 <- lm(yvar ~ xvar + I(xvar^2) + I(xvar^3), data = q3_data)
summary(modd3)
```

```
##
## Call:
## lm(formula = yvar ~ xvar + I(xvar^2) + I(xvar^3), data = q3_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -170.45  -33.87   11.51   53.08  104.74
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1207.61946  289.49411   4.171  0.00072 ***
## xvar        -101.82416   37.21510  -2.736  0.01465 *
## I(xvar^2)      2.57134    1.38131   1.862  0.08114 .
## I(xvar^3)     -0.01749    0.01563  -1.119  0.27963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.02 on 16 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7253
## F-statistic: 17.73 on 3 and 16 DF,  p-value: 2.437e-05
```

The third degree term in the cubic model has a large p-value, and is not statistically significant. Also, the p-value for the second order term has increased above 0.05, making it marginally significant. The R^2 has increased from 0.7506 to 0.7687, but the adjusted R^2 only marginally increases from 0.7213 to 0.7253.

```
par(mfrow = c(2,2))
plot(modd2)
```



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

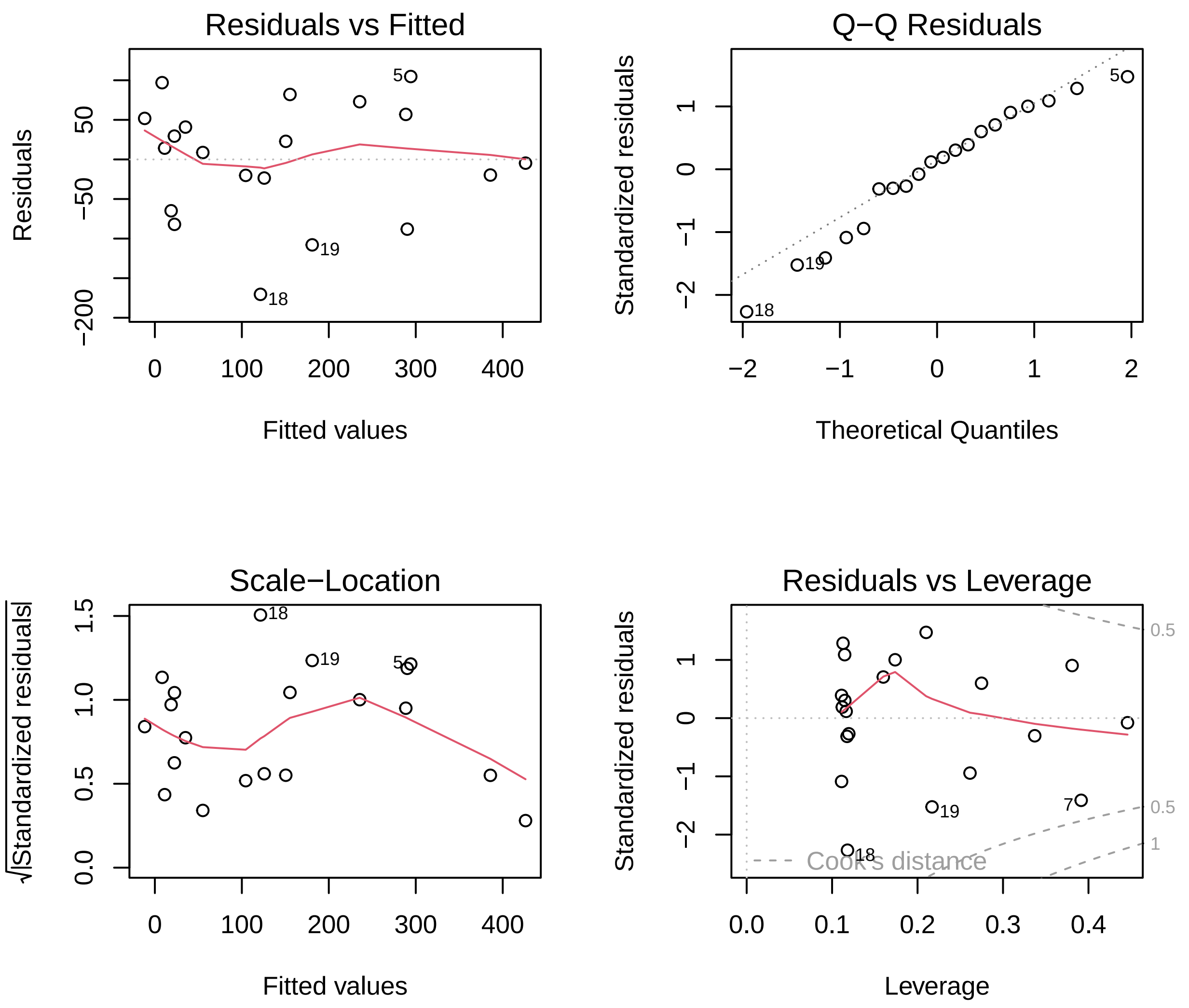
```
shapiro.test(rstandard(modd2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(modd2)
## W = 0.89878, p-value = 0.03913
```

```
ncvTest(modd2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.06459266, Df = 1, p = 0.79938
```

```
par(mfrow = c(2,2))
plot(modd3)
```



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

```
shapiro.test(rstandard(modd3))
```

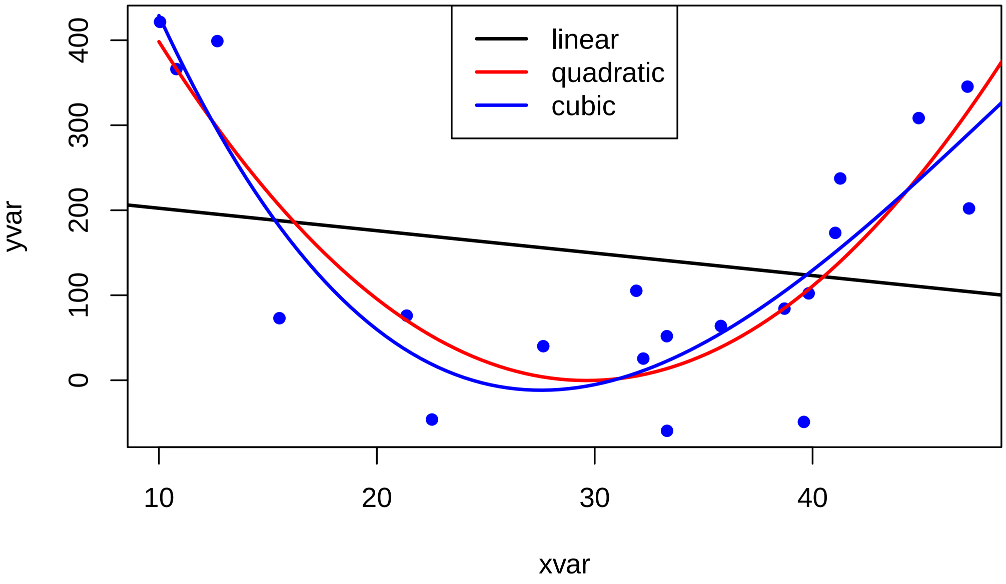
```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(modd3)
## W = 0.95789, p-value = 0.5026
```

```
ncvTest(modd3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.0005259536, Df = 1, p = 0.9817
```

(g) Draw a scatterplot of `yvar` as a function of `xvar` and add the quadratic regression curve that you fitted.

```
plot(yvar~xvar, data = q3_data, pch = 16, col = 'blue')
abline(modd1, lwd = 2)
curve(907.791 - 61.285*x + 1.034*x^2, 10, 50, add = T, col = 'red', lwd = 2)
curve(1207.63 -101.8242*x + 2.57134*x^2 - 0.01749*x^3,
      10, 50, add = T, col = 'blue', lwd = 2 )
legend('top', c('linear','quadratic','cubic'),
       col = c('black','red','blue'), lwd = 2)
```



Problem 4

Use the data set PL925FPr4.csv for this problem. This set has six variables, and we are interested in building a regression model for vary in terms of the other variables in the set.

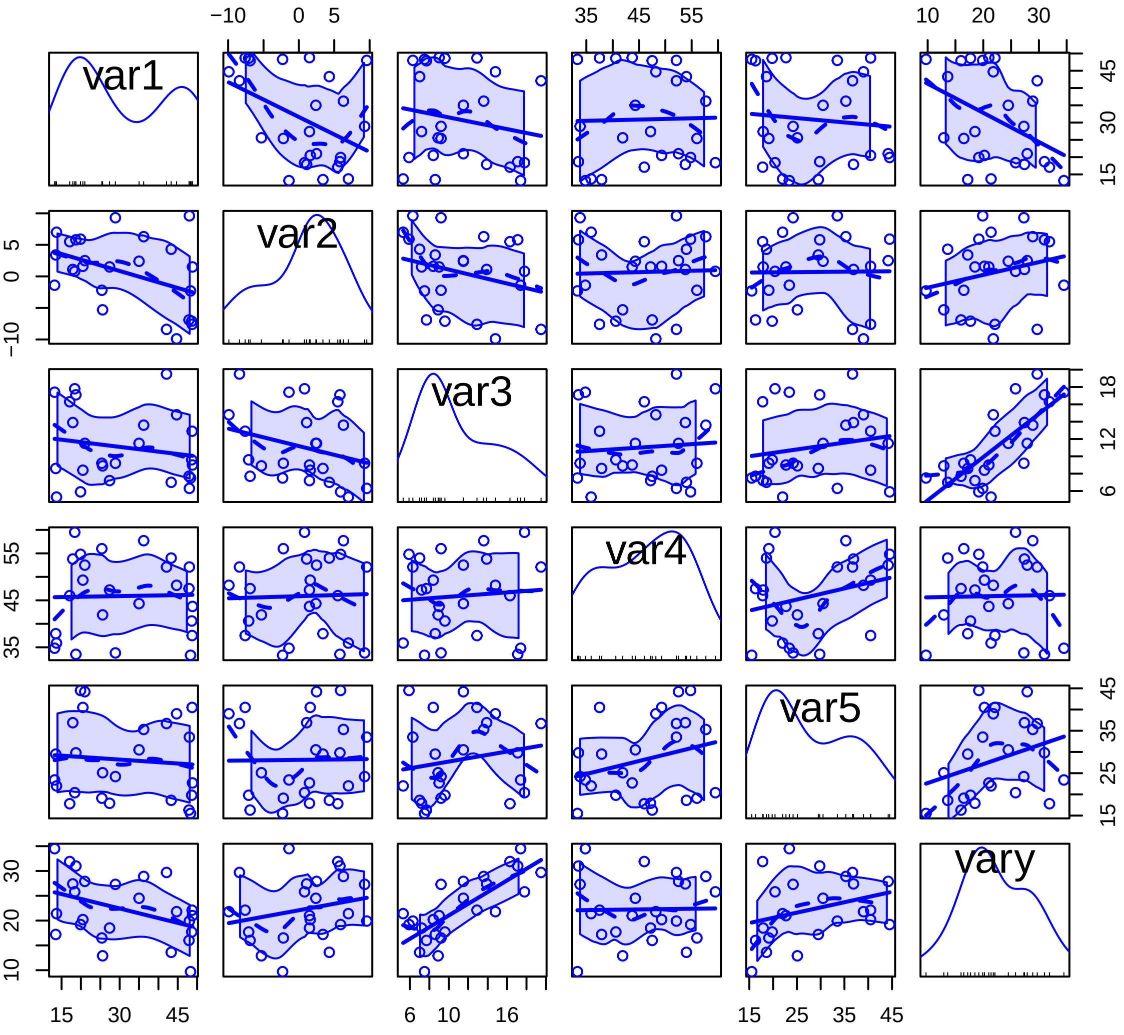
Read the data:

```
q4_data <- read.csv('PL925FPr4.csv')
str(q4_data)
```

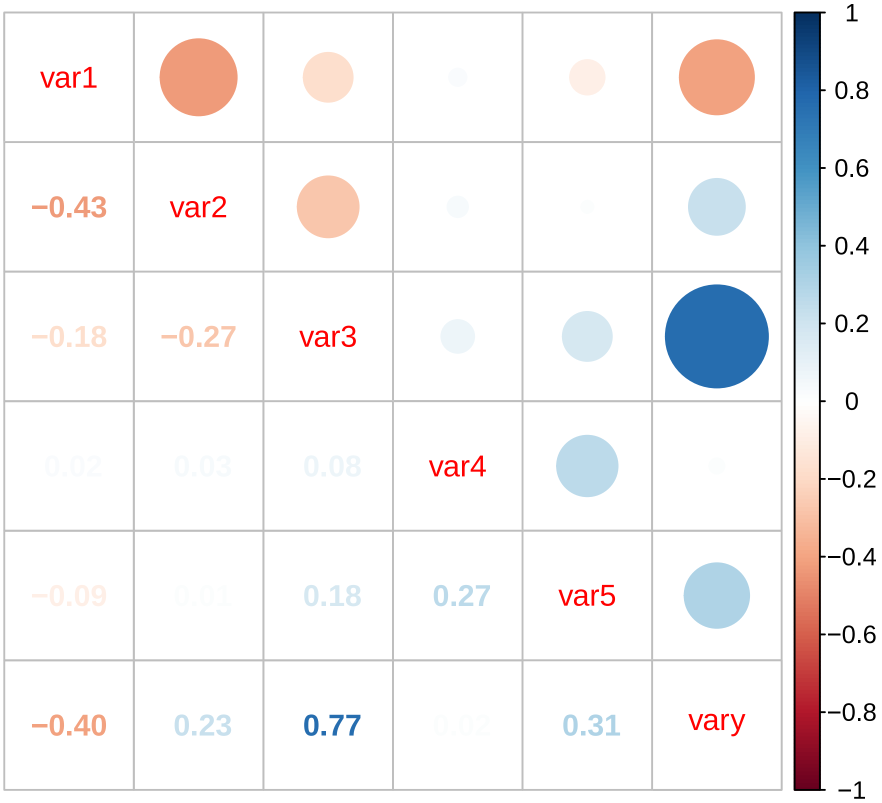
```
## 'data.frame':    25 obs. of  6 variables:
##  $ var1: num  18.7 21 25.6 36.2 27.4 42.1 25.4 43.3 28.9 13.7 ...
##  $ var2: num  5.8 2.5 -5.3 6.3 1.5 -8.4 -2.2 4.3 9.3 7 ...
##  $ var3: num  17.1 11.5 8.9 13.6 7.2 19.5 9.2 7 9.2 5.3 ...
##  $ var4: num  33.5 52.5 41.9 57.7 47.2 52.1 56 54 33.8 35.9 ...
##  $ var5: num  29.8 44.2 25.1 35.3 17.9 36.7 19.1 18.6 24.2 22 ...
##  $ vary: num  31 27.9 12.9 28.9 18.5 29.7 16.5 13.6 27.3 21.4 ...
```

(a) Start by using the function scatterplotMatrix in the car library to obtain a matrix of scatterplots for the variables in the data set. Calculate and plot the correlation matrix. Comment on what you observe.

```
library(car)
scatterplotMatrix(q4_data)
```



```r
corr_q3 = cor(q4_data)
corrplot::corrplot.mixed(corr_q3)
```



(b) Fit a complete linear regression model and find a minimal adequate model using backward elimination with a critical p-value of 0.05.

```r
model1 <- lm(vary ~ ., data = q4_data)
summary(model1)
```

```
##
## Call:
## lm(formula = vary ~ ., data = q4_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5635 -0.8469  0.4364  2.1604  4.1582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.58959    4.30643   1.995  0.06063 .
## var1        -0.02153    0.05338  -0.403  0.69116
## var2         0.50754    0.12948   3.920  0.00092 ***
## var3         1.31763    0.16273   8.097  1.4e-07 ***
## var4        -0.08604    0.07672  -1.122  0.27605
## var5         0.12028    0.06831   1.761  0.09437 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.98 on 19 degrees of freedom
## Multiple R-squared:  0.8304, Adjusted R-squared:  0.7857
## F-statistic:  18.6 on 5 and 19 DF,  p-value: 9.883e-07
```

Drop `var1` since it has the largest p-value

```r
model2 <- update(model1, . ~ . -var1)
summary(model2)
```

```
##
## Call:
## lm(formula = vary ~ var2 + var3 + var4 + var5, data = q4_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7979 -0.9570  0.2861  2.0758  4.4342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.74819    3.68788   2.101   0.0485 *
## var2         0.53376    0.10961   4.870 9.28e-05 ***
## var3         1.33927    0.15038   8.906 2.14e-08 ***
## var4        -0.08871    0.07482  -1.186   0.2496
## var5         0.12174    0.06677   1.823   0.0833 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.917 on 20 degrees of freedom
## Multiple R-squared:  0.8289, Adjusted R-squared:  0.7947
## F-statistic: 24.23 on 4 and 20 DF,  p-value: 1.995e-07
```

Drop `var4` since it has the largest p-value

```r
model3 <- update(model2, . ~ . -var4)
summary(model3)
```

```
##
## Call:
## lm(formula = vary ~ var2 + var3 + var5, data = q4_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9796 -1.2958  0.1173  1.9682  5.3620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.33766    2.33019   1.862 0.076729 .
## var2         0.52852    0.11058   4.780 0.000101 ***
## var3         1.33151    0.15169   8.778  1.8e-08 ***
## var5         0.10137    0.06515   1.556 0.134654
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.945 on 21 degrees of freedom
## Multiple R-squared:  0.8169, Adjusted R-squared:  0.7907
## F-statistic: 31.23 on 3 and 21 DF,  p-value: 6.262e-08
```

Drop `var5` since it has the largest p-value

```r
model4 <- update(model3, . ~ . -var5)
summary(model4)
```

```
##
## Call:
## lm(formula = vary ~ var2 + var3, data = q4_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.073 -1.666  0.710  1.627  4.623
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.6971     1.8256   3.669  0.00135 **
## var2          0.5393     0.1139   4.736 9.99e-05 ***
## var3          1.3755     0.1538   8.946 8.79e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.039 on 22 degrees of freedom
## Multiple R-squared:  0.7958, Adjusted R-squared:  0.7772
## F-statistic: 42.86 on 2 and 22 DF,  p-value: 2.576e-08
```

This is the minimal adequate model. All variables have small p-values.

(c) Find a minimal adequate model using AIC and the function `stepAIC` in the `MASS` package. Compare with the model in (b)

```r
library(MASS)
stepAIC(model1)
```

```
## Start:  AIC=59.74
## vary ~ var1 + var2 + var3 + var4 + var5
##
##        Df Sum of Sq    RSS    AIC
## - var1  1      1.45 170.18 57.950
## - var4  1     11.17 179.91 59.339
## <none>              168.74 59.737
## - var5  1     27.53 196.27 61.515
## - var2  1    136.46 305.20 72.552
## - var3  1    582.26 750.99 95.063
##
## Step:  AIC=57.95
## vary ~ var2 + var3 + var4 + var5
##
##        Df Sum of Sq    RSS    AIC
## - var4  1     11.96 182.14 57.648
## <none>              170.18 57.950
## - var5  1     28.28 198.46 59.793
## - var2  1    201.78 371.96 75.498
## - var3  1    674.88 845.06 96.013
##
## Step:  AIC=57.65
## vary ~ var2 + var3 + var5
##
```

```
##        Df Sum of Sq    RSS    AIC
## <none>              182.14 57.648
## - var5  1     21.00 203.14 58.376
## - var2  1    198.15 380.30 74.052
## - var3  1    668.35 850.49 94.173
```

```
##
## Call:
## lm(formula = vary ~ var2 + var3 + var5, data = q4_data)
##
## Coefficients:
## (Intercept)         var2         var3         var5
##      4.3377       0.5285       1.3315       0.1014
```

This procedure gives a different model, that includes three variables, `var2` and `var3`, the variables in the previous model, and `var5`. This is `model3` in the sequence of models we developed in (b). Observe that the AIC for `model3` is 57.648, while for `model4` it is 58.376.

(d) Find a minimal adequate model maximizing the R^2. Compare with the models in (b) and (c).

We use the function `regsubsets` in the library `leaps`:

```r
library(leaps)
q <- regsubsets(as.matrix(q4_data[,-6]), q4_data[,6])
summary(q)
```

```
## Subset selection object
## 5 Variables  (and intercept)
##      Forced in Forced out
## var1     FALSE      FALSE
## var2     FALSE      FALSE
## var3     FALSE      FALSE
## var4     FALSE      FALSE
## var5     FALSE      FALSE
## 1 subsets of each size up to 5
## Selection Algorithm: exhaustive
##          var1 var2 var3 var4 var5
## 1  ( 1 ) " "  " "  "*"  " "  " "
## 2  ( 1 ) " "  "*"  "*"  " "  " "
## 3  ( 1 ) " "  "*"  "*"  " "  "*"
## 4  ( 1 ) " "  "*"  "*"  "*"  "*"
## 5  ( 1 ) "*"  "*"  "*"  "*"  "*"
```

```r
summary(q)$adjr2
```

```
## [1] 0.5696106 0.7772205 0.7907378 0.7947056 0.7857357
```

```r
max(summary(q)$adjr2)
```

```
## [1] 0.7947056
```

```r
which.max(summary(q)$adjr2)
```

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

The maximum adjusted R^2 is attained by the model described in the fourth row of the table, which includes all variables except `var1`. This is `model2`, and has an adjusted R^2 of 0.7947056.

(e) How many different models do you have? What would you do if you had to select one of them?

There are three models: `model4`, selected by backward elimination, `model3` selected by AIC, and `model2` selected by maximizing the R^2.

model	AIC	adjusted R^2
`model2`	57.950	0.79471
`model3`	57.648	0.79074
`model4`	58.376	0.77722