Problem 1 (4.7)

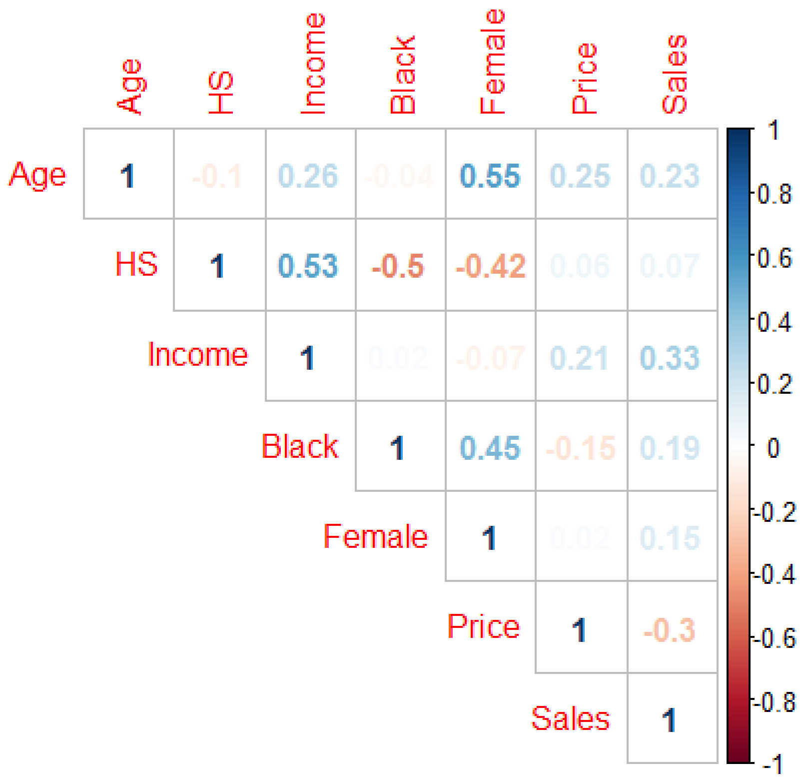
Part a.

```
> # Sales vs Price ~ Expected negative relationship, since
> # the higher the price, the less likely people are going to spend money
> # to buy cigarettes.
>
> # Sales vs Income ~ Expected positive relationship, since
> # the higher the income, the more money available to spend
>
> # Sales vs Age ~ Expected positive relationship, since
> # older people tend to consume more cigarettes compared to younger people,
> # and older people tend to have more income
>
> # Sales vs HS ~ Expected positive relationship since high school completion
> # is positively related to income. However, it is likely not to be a strong
> # relationship so no relationship might be expected since preferences for
> # smoking (and buying) cigarettes seems to be similar across different
> # education backgrounds
>
> # Sales vs Back ~ Expected positive relationship because surveys have
> # shown that African American tend to smoke more than other races
>
> # Sales vs Female ~ Expected negative relationship because surveys have
> # shown that men tend to smoke more than women.
```

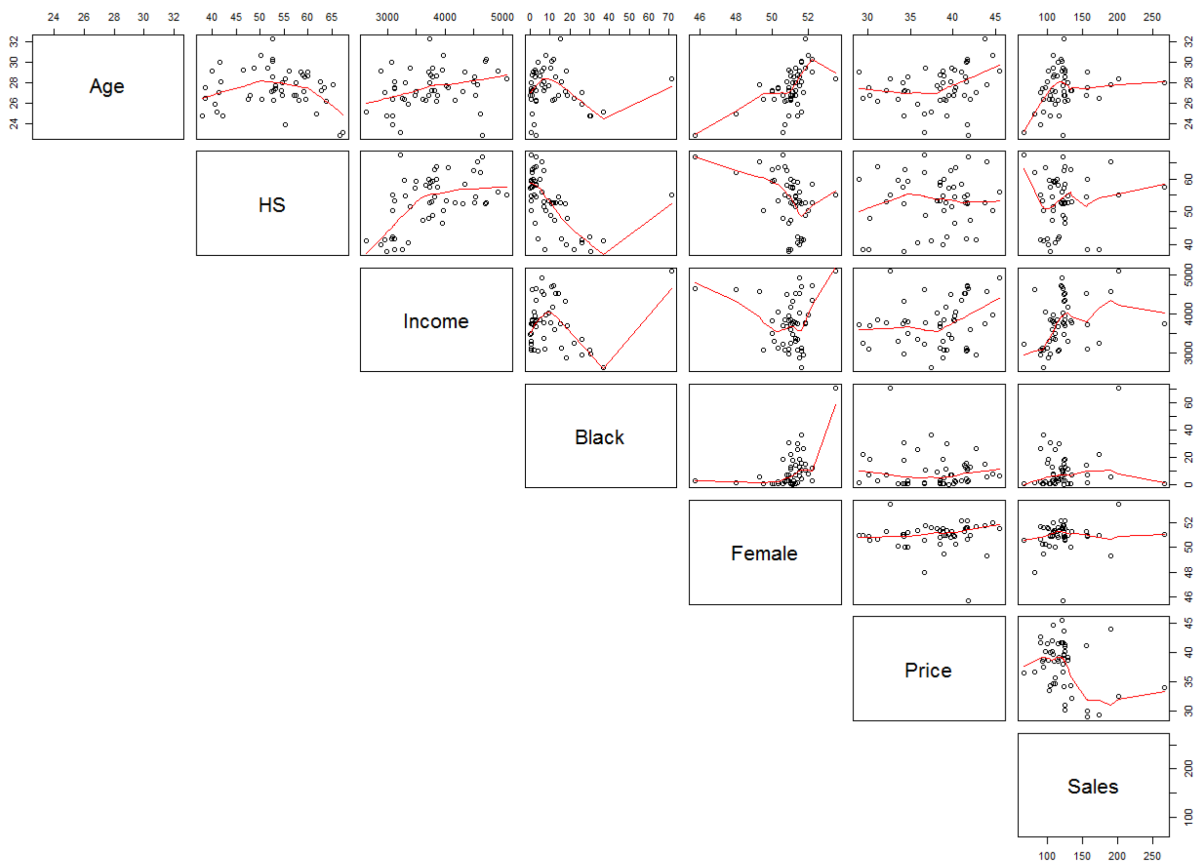
Part b.

```
> # Import data
> filename = "P088.txt"
> mydata = read.table(filename,header = T)
> corr = round(cor(mydata[-1]),2)
> corr
         Age    HS Income Black Female Price Sales
Age     1.00 -0.10   0.26 -0.04   0.55  0.25  0.23
HS     -0.10  1.00   0.53 -0.50  -0.42  0.06  0.07
Income  0.26  0.53   1.00  0.02  -0.07  0.21  0.33
Black  -0.04 -0.50   0.02  1.00   0.45 -0.15  0.19
Female  0.55 -0.42  -0.07  0.45   1.00  0.02  0.15
Price   0.25  0.06   0.21 -0.15   0.02  1.00 -0.30
Sales   0.23  0.07   0.33  0.19   0.15 -0.30  1.00
```

```
> # Plot separate
> corrplot(corr,method="number", type="upper")

> pairs(mydata[,-1], main = "Correlation coeffficients matrix and scatter plot",
+       pch = 21, lower.panel = NULL, panel = panel.smooth,cex.labels=2)
```



Correlation coefficients matrix and scatter plot



Part c.

```
> # In reality there should be no disagreement. The sign of the correlation coefficient
> # tells you the direction of the linear relationship, and the magnitude tells you the
> # strength of the linear relationship.
> # The scatter plot is a visual representation of the correlation coefficient,
> # where high correlation is shown when data points are clustered tightly together
> # around a line, and the sign is shown by the direction of the association of the
> # points (e.g. pointing down means as one variable increases, the other decreases,
> # so negative correlation).
> # So for example, price and sales have a correlation of -0.33, and this is shown
> # in the scatter plot as price increase, sales decreases.

> # However, it is also important to note that outliers might influence the correlations
> # and potentially apparent disagreement (or not make it clearly the agreement).
> # For example there seems to be a point with high value of sale. The correlation
> # coefficient for Income and Black is almost zero, but the scatter plot clearly shows
> # a negative relationship which is not captured by the correlation coefficient because
> # of the outlier with very high income.

> # Note also that there might be clear trend (e.g quadratic) but it may not be captured by
> # the correlation coefficient which captures linear relationship.
```

Part d.

```
> # For the most part the expectations in part(a) match the pairwise correlation
> # coefficients matrix and the corresponding scatter plot. For example, as expected
> # Sale and price are negatively correlated, while Sale and income are positively
> # correlated. As expected also, these relationships are the strongest. The only
> # one that didn't match the expectations is the positive relationship between
> # female and sales, where it was expected the relationship to be negative.
> # However, the relationship between female and sales (correlation = 0.15) does not
> # seem very strong. In addition, relationships that might not be linear cannot be
> # captured by the correlation coefficient, and there might also be a "third" variable
> # that might be influencing the pairwise correlations and scatter plots.
```

Part e.

```
> fit = lm(Sales ~ Age + HS + Income + Black + Female + Price, mydata)
> summary(fit)

Call:
lm(formula = Sales ~ Age + HS + Income + Black + Female + Price,
    data = mydata)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 103.34485  245.60719   0.421  0.67597
Age           4.52045    3.21977   1.404  0.16735
HS           -0.06159    0.81468  -0.076  0.94008
Income        0.01895    0.01022   1.855  0.07036 .
Black         0.35754    0.48722   0.734  0.46695
Female       -1.05286    5.56101  -0.189  0.85071
Price        -3.25492    1.03141  -3.156  0.00289 **
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.17 on 44 degrees of freedom
Multiple R-squared: 0.3208,     Adjusted R-squared: 0.2282
F-statistic: 3.464 on 6 and 44 DF,  p-value: 0.006857

> # The coefficient of Age, Income and black are positive, so they match the expectation
> # in part (a) as it is expected that an increase in these variables would lead to an
> # increase in the sales.
>
> # The coefficient of Price is negative, so they so it matches the expectation
> # in part (a) as it is expected that an increase in this variable would lead to a
> # decrease in the sales.
>
> # The coefficient of Female is negative, so they so it matches the expectation
> # in part (a) as it is expected that more females than males would lead to a decrease
> # in sales because males tend to smoke more.
>
```

```
> # The coefficient of HS is negative, so it does not match the expectation
> # of positive relationship with sale. However, it was also expected that the relationship
> # between HS and sales to be very weak or none at all, and this is consistent with the
> # small magnitude of the regression coefficient and high p-value indicating insignificant
> # effect (not significantly different than zero)
```

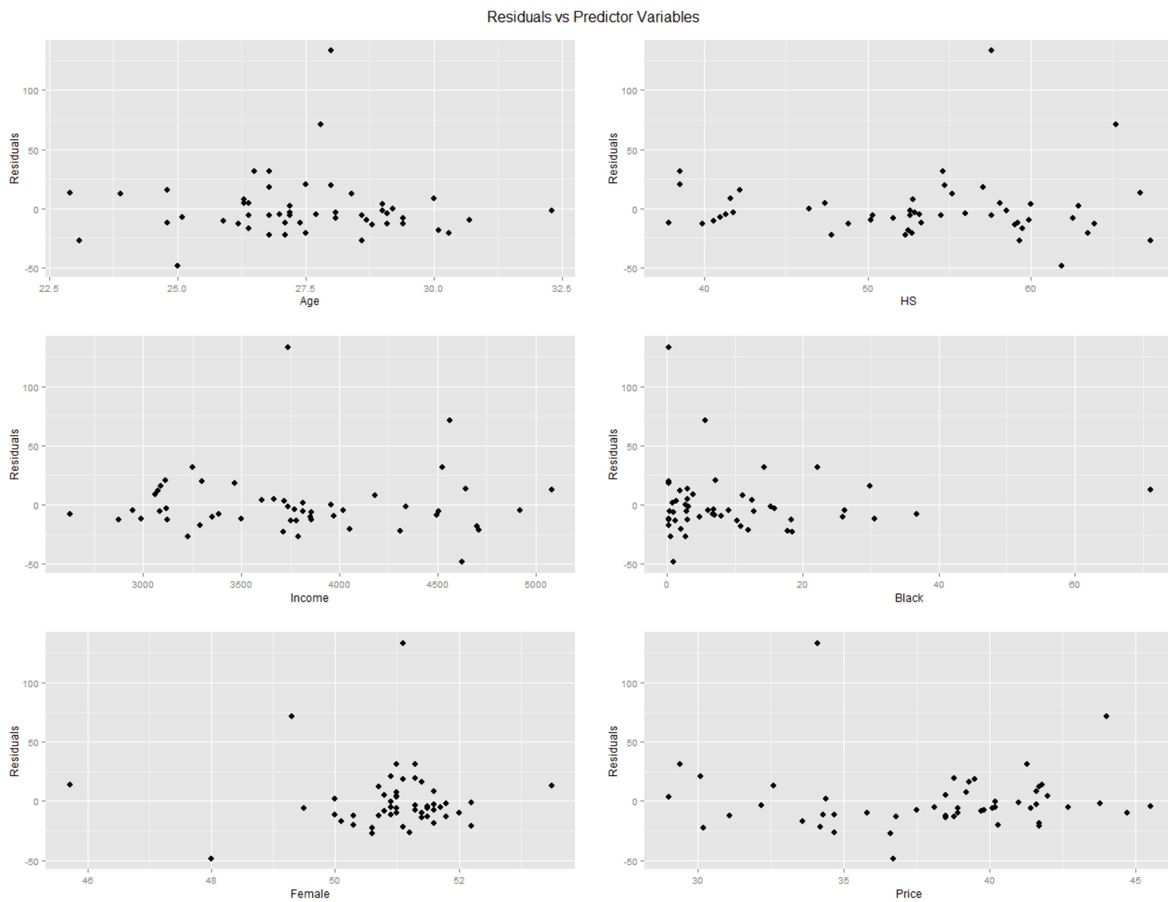
Part f.

```
> # There are differences between the pairwise correlation coefficients and the
> # correlation coefficients between Sales and each of the predictors. All of them
> # agree in terms of sign/direction except Female and HS. This can be explained by
> # the fact that the regression coefficient tells you the effect of a variable after
> # accounting for the other predictors, while the correlation coefficient measures
> # the pairwise relationship between two variables. Thus, it might be the case that
> # a variable to have an opposite effect when other variables have been taken into
> # account (e.g. female effect after controlling for income) because there might
> # be a lurking variable confounding the relationship in the pairwise correlation
> # coefficients.
> # The pairwise correlation ignores the fact that there is a more plausible lurking
> # variable giving rise to the observed correlation. So the effects of regression
> # coefficients depend on the presence of other predictors in the model. Outliers and
> # influential points might also impact the regression coefficients.
>
> # Example, the regression of sales vs only female (no other variables taken into account)
> # agrees with the sign of the pairwise correlation coefficient. But when other variables
> # are added, then the sign changes. Similar results is observed for HS.
>
> lm(Sales ~ Female ,mydata)

Call:
lm(formula = Sales ~ Female, data = mydata)

Coefficients:
(Intercept)       Female
    -93.426        4.219

> lm(Sales ~ HS ,mydata)

Call:
lm(formula = Sales ~ HS, data = mydata)

Coefficients:
(Intercept)           HS
   107.3331       0.2673

> # It is also important to point out that the p-values say for example that the
> # effect of HS  and Female after taking other predictors into account is insignificant
> # (not significantly different than zero).
```

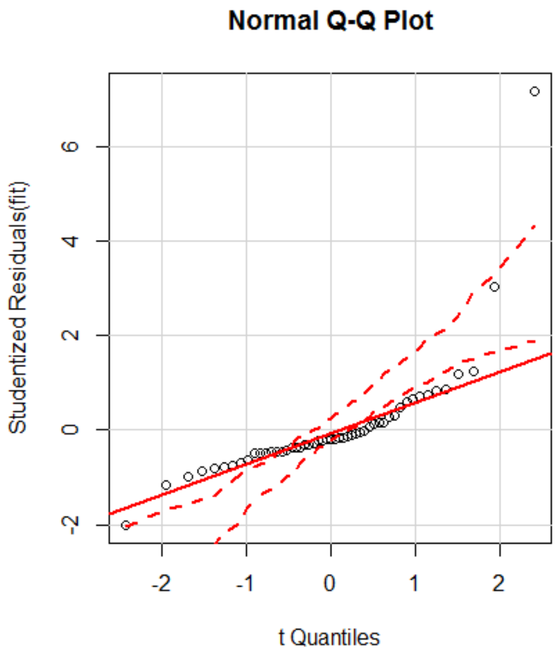
Part g.

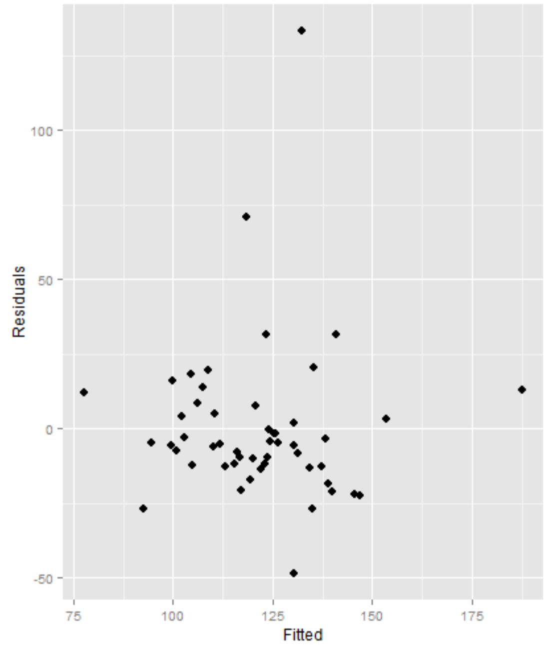
```
> # There doesn't seem anything wrong with the tests and conclusions reached in 3.15
> # because inference tests for significance and percent of variation were done based on
> # on the model (not pairwise correlations), so the effect of the predictors were taken
> # into account. However, from a model perspective, the validity of these conclusions
> # will depend on the validity of the assumptions of the model (errors, linearity, etc).

> # In order to test if there is anything wrong the tests and conclusions reached
> # in 3.15, we need to run a diagnostics to check all the assumptions hold and that there
> # no outliers or influential point that might be influencing the conclusions of the tests
>
> ## check the fit (check linearity assumption by plotting residuals vs each predictor)

> plot_vector = vector(mode="list",length=6)
>
> plot_vector[[1]] = ggplot(mydata,aes(x=mydata[[2]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[2]),y = "Residuals")
>
```

```
> plot_vector[[2]]  = ggplot(mydata,aes(x=mydata[[3]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[3]),y = "Residuals")
>
> plot_vector[[3]]  = ggplot(mydata,aes(x=mydata[[4]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[4]),y = "Residuals")
>
> plot_vector[[4]]  = ggplot(mydata,aes(x=mydata[[5]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[5]),y = "Residuals")
>
> plot_vector[[5]]  = ggplot(mydata,aes(x=mydata[[6]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[6]),y = "Residuals")
>
> plot_vector[[6]] = ggplot(mydata,aes(x=mydata[[7]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[7]),y = "Residuals")
>
> grid.arrange(plot_vector[[1]],
+              plot_vector[[2]],
+              plot_vector[[3]],
+              plot_vector[[4]],
+              plot_vector[[5]],
+              plot_vector[[6]],
+              ncol=2, main = "Residuals vs Predictor Variables")
>
> # All plots look random so assumptions about the form of the model
> # (linear in the regression parameters) is satisfied.
```



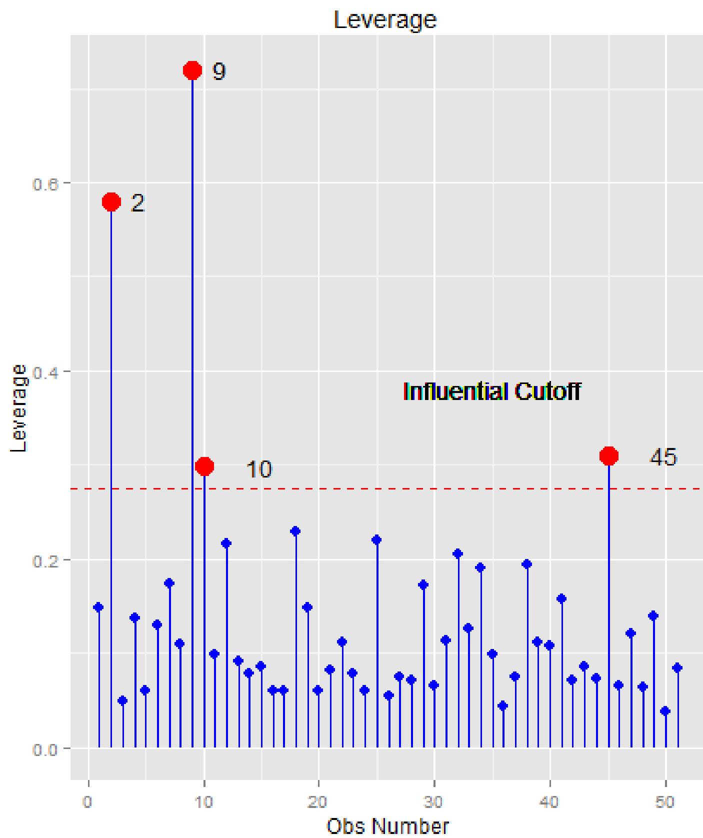
```
> ## check normality (using qq plot)
> qqPlot(fit, main = "Normal Q-Q Plot")

> ## check Checking Homoscedasticity (using residuals vs. fitted)
>
> ggplot(mydata,aes(x=fit$fitted, y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = "Fitted",y = "Residuals")
```





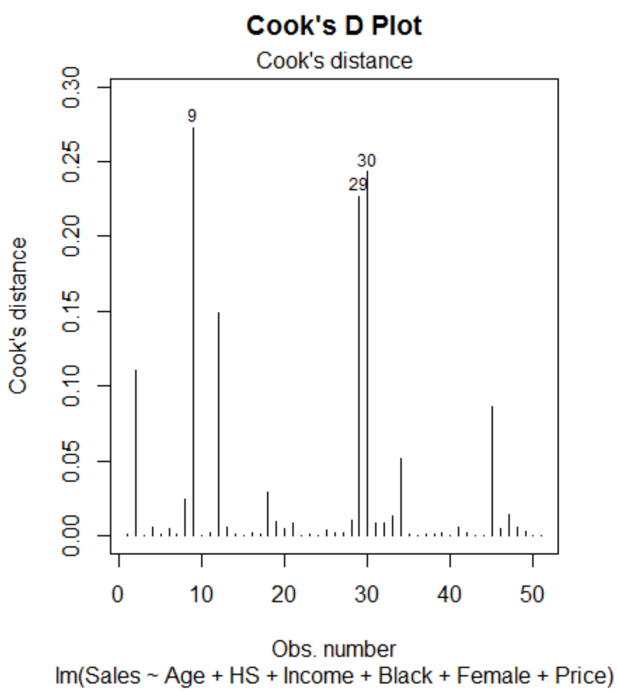

```
> # The plot shows that most points fall along the line, indicating the normality
> # assumption of errors is satisfied. However, it looks that there are a few outliers

> # The plot shows data points are random forming a parallel band, indicating the
> # common variance assumption of errors is valid (Homoscedasticity)

> ## check independence (not time series data, so ignore)
> # the Durbin-Watson
>
> ## Check multicollinearity

> vif(fit)
     Age       HS   Income    Black   Female    Price
2.300617 2.676465 2.325164 2.392152 2.406417 1.142181
>
> # The VIF < 10 for all predictors, so there is no multicollinearity problem.


> ## Compute Leverage for measuring "unusualness" of x's
> leverage =  hat(model.matrix(fit))
> mydata$leverage = leverage

> # Compute cutoff
> p=6
> n=dim(mydata)[1]
> cutoff = 2*(p+1)/n
> cutoff
[1] 0.2745098
>
```

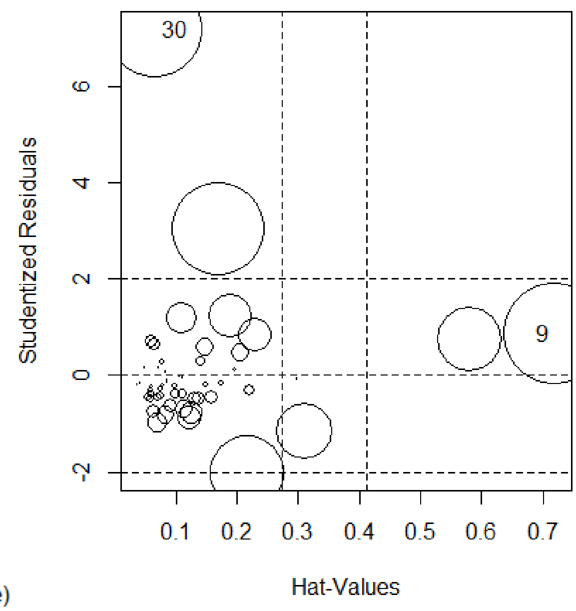
```
> # Find high leverage points
> influential = mydata["leverage"]
> influential = subset(influential,leverage> cutoff)
> influential
    leverage
2  0.5801604
9  0.7197128
10 0.2984808
45 0.3105737

> # Add observation number so can plot
> influential$obs = as.numeric(rownames(influential))
> mydata$obs = 1:n
>
> # Plot influential points
> ggplot(mydata,aes(x=obs, leverage)) +
+   geom_point(size = 3, color="blue") +
+   geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
+   geom_text(aes(35, .38, label="Influential Cutoff")) +
+   geom_segment(aes(xend=obs, yend=0), color="blue") +
+   geom_text(data =influential, aes(x=obs, y = leverage,
+                                    label = obs), hjust = -1.5) +
+   labs(title="Leverage ",
+        x = "Obs Number",
+        y = "Leverage") +
+   geom_point(data=mydata[influential$obs,], colour="red", size=5)
```




```
> # Using the rule of thumb (hii>2(p+1)/2), the observations
> # 2,9,10,45 are regarded as high leverage points
```

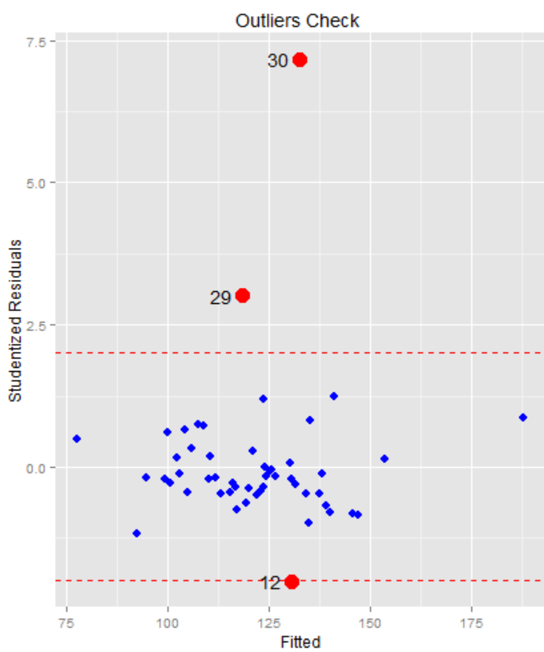
```
> ## Compute Cook distances for measuring influence

> # Cook's D plot
> cutoff = 4/(dim(mydata)[1]);
> plot(fit, which=4, cook.levels=cutoff, main = "Cook's D Plot");

> # influence plot
> library(car)    # needed for "influencePlot" function below
> influencePlot(fit)
    StudRes        Hat     CookD
9  0.859766 0.71971284 0.5222761
30 7.165222 0.06634506 0.4930253
```






```
> # point 9, 29, 30 are influential points (using cutoff 4/n)

> # The circles for each observation represent the relative size of the Cook's D
> # point 9 is high leverage and influential, 30 is an outlier with high influence

> # Outliers/high leverage/influential points

> summary(influence.measures(fit))
Potentially influential observations of
         lm(formula = Sales ~ Age + HS + Income + Black + Female + Price,      data = mydat
a) :

   dfb.1_ dfb.Age dfb.HS dfb.Incm dfb.Blck dfb.Feml dfb.Pric dffit   cov.r   cook.d hat
2   0.68   0.07    0.00   0.07     0.25    -0.64     0.17     0.87    2.56_*  0.11   0.58_*
9  -0.20   0.19    0.55  -0.01     0.98     0.04    -0.19     1.38_*  3.72_*  0.27   0.72_*
10 -0.01  -0.04   -0.02   0.03    -0.03     0.02    -0.01    -0.05    1.67_*  0.00   0.30
25 -0.01   0.01   -0.04   0.09    -0.09     0.01    -0.04    -0.16    1.49_*  0.00   0.22
29  0.58   0.54    0.53  -0.19     0.64    -0.82     0.61     1.37_*  0.37_*  0.23   0.17
30 -0.42   0.13    0.25   0.01    -0.72     0.47    -1.19_*   1.91_*  0.01_*  0.24   0.07

> # Using the R function, potential problematic points are: 2,9,10,25,29,30
```

```
> ## Studentized residuals vs fitted
> library(MASS)
> stu_res = studres(fit)
> mydata$stu_res = stu_res
> mydata$fitted = fit$fitted
>
> # Find outliers points
> outlier = mydata["stu_res"]
> outlier = subset(outlier,abs(stu_res)>2)
> outlier
      stu_res
12 -2.006893
29  3.017994
30  7.165222

> # Add observation number and fitted so can plot
> outlier$fitted = fit$fitted[outlier$obs]
> outlier$obs = as.numeric(rownames(outlier))
>
> ggplot(mydata,aes(x=fitted, stu_res)) +
+   geom_point(size = 3, color="blue") +
+   geom_hline(yintercept=2, linetype="dashed" , color = "red") +
+   geom_hline(yintercept=-2, linetype="dashed" , color = "red") +
+   geom_text(data =outlier, aes(x=fitted, y = stu_res,
+                            label = obs), hjust = 1.5) +
+   labs(title="Outliers Check ",
+        x = "Fitted",
+        y = "Studentized Residuals") +
+   geom_point(data=mydata[outlier$obs,], colour="red", size=5)
```

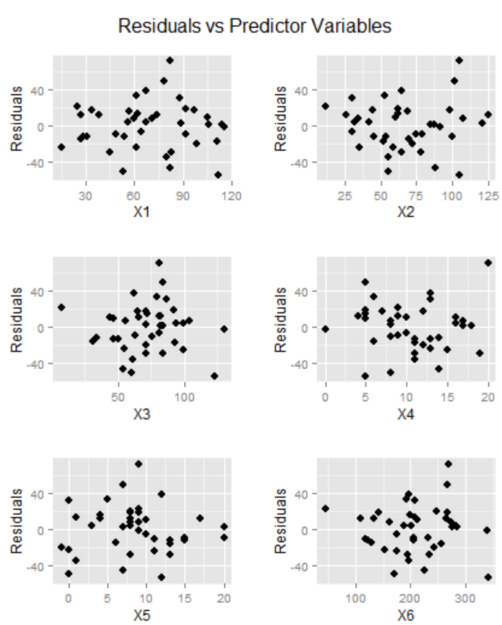



```
> # Using the rule that |studentized residuals| > 2, the observations
> # 12,29,30 are are regarded as outliers

> # Remove potential outliers and influential points described above
> # because the regression coefficients and interpretations might
> # change due to the impact of these points, potentially altering the validity of
> # the tests and conclusions reached in 3.15.
```

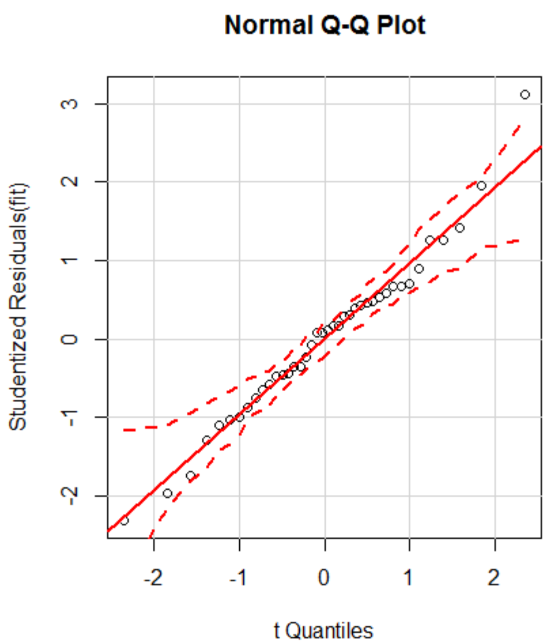
Problem 2 (4.12)

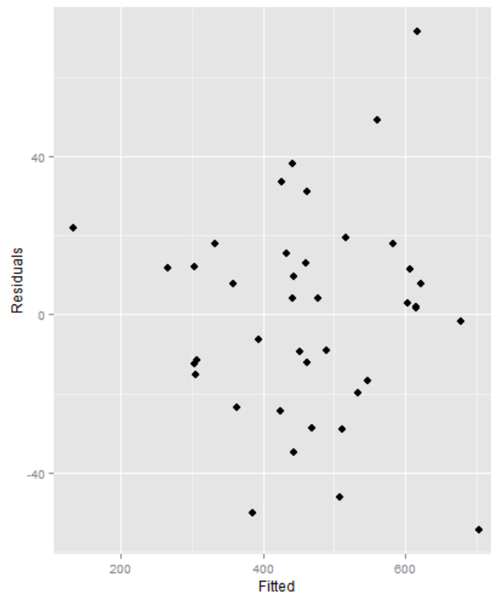
Part a.

```
> ## check the fit (check linearity assumption by plotting residuals against each predictor
)
> fit = lm(Y~.,mydata)
>
> plot_vector = vector(mode="list",length=6)
>
> plot_vector[[1]] = ggplot(mydata,aes(x=mydata[[2]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[2]),y = "Residuals")
>
> plot_vector[[2]]  = ggplot(mydata,aes(x=mydata[[3]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[3]),y = "Residuals")
>
> plot_vector[[3]]  = ggplot(mydata,aes(x=mydata[[4]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[4]),y = "Residuals")
>
> plot_vector[[4]]  = ggplot(mydata,aes(x=mydata[[5]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[5]),y = "Residuals")
>
> plot_vector[[5]]  = ggplot(mydata,aes(x=mydata[[6]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[6]),y = "Residuals")
>
> plot_vector[[6]] = ggplot(mydata,aes(x=mydata[[7]], y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = colnames(mydata[7]),y = "Residuals")
>
> grid.arrange(plot_vector[[1]],
+              plot_vector[[2]],
+              plot_vector[[3]],
+              plot_vector[[4]],
+              plot_vector[[5]],
+              plot_vector[[6]],
+              ncol=2, main = "Residuals vs Predictor Variables")
>
> # All plots look random so assumptions about the form of the model
> # (linear in the regression parameters) is satisfied.
```

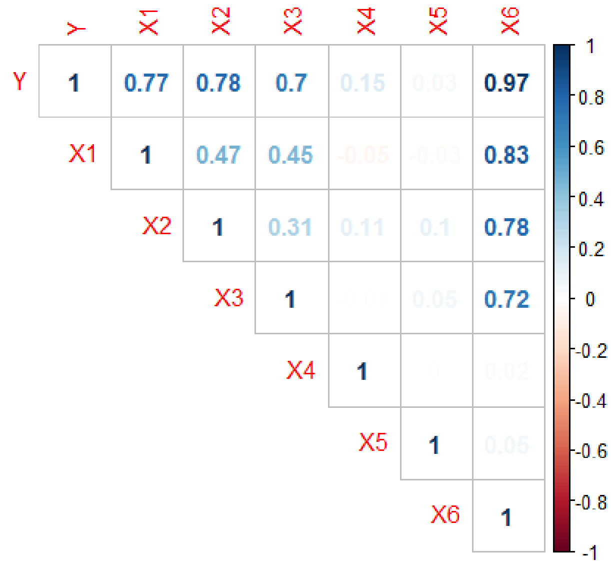


```
> ## check normality (using qq plot)
> qqPlot(fit, main = "Normal Q-Q Plot")

> ## check Checking Homoscedasticity (using residuals vs. fitted)
>
> ggplot(mydata,aes(x=fit$fitted, y = fit$resid)) +
+   geom_point(size = 3) +
+   labs(x = "Fitted",y = "Residuals")
```

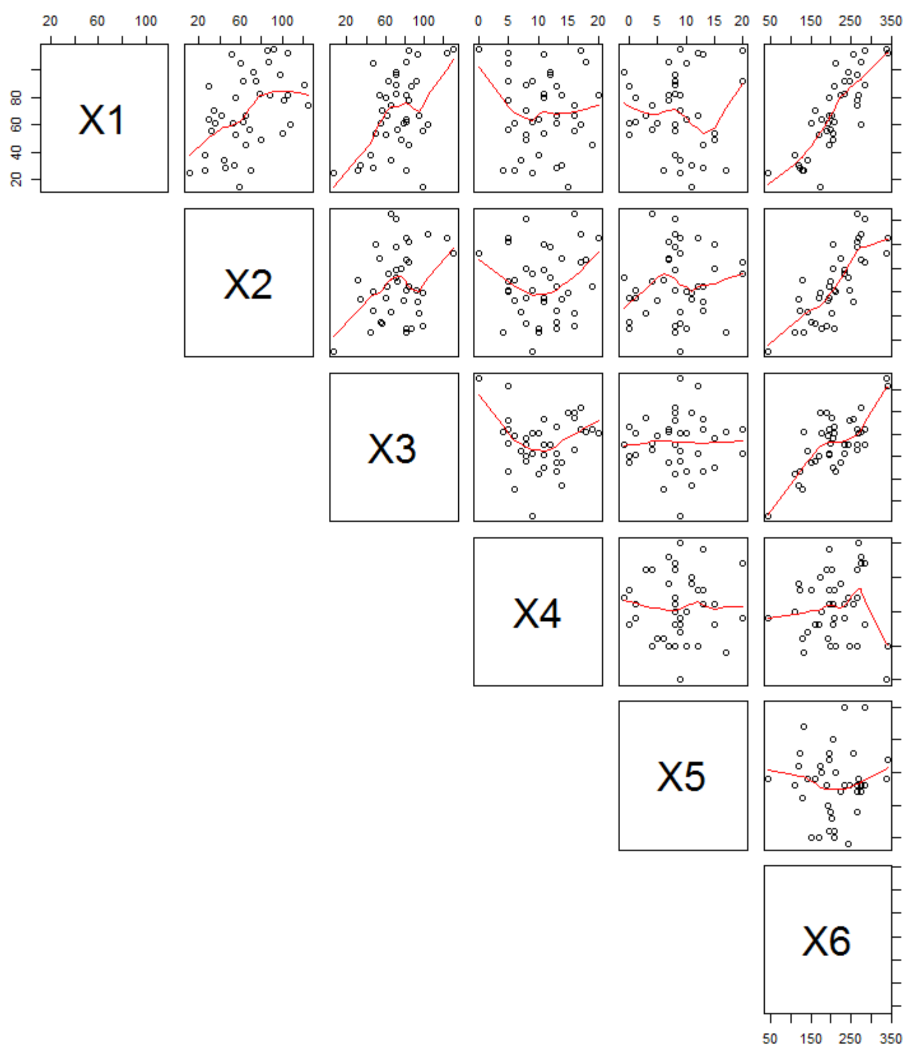





```
> # The plot (QQ) shows that most points fall along the line,
indicating the normality assumption of errors is satisfied
>
> # The plot (res vs fitted) shows data points are random forming a parallel band,
indicating the common variance assumption of errors is valid (Homoscedasticity)

> ## check independence (not time series data, ignore)
> # the Durbin-Watson

> ## Check multicollinearity
>
> corr = round(cor(mydata),2)

> pairs(mydata[,-1], main = "Correlation coefffficients matrix and scatter plot",
+       pch = 21, lower.panel = NULL, panel = panel.smooth,cex.labels = 3)

> vif(fit)
         X1          X2          X3          X4          X5          X6
1062.167136 1103.595644  764.394462    1.036287    1.023312 5310.504807
>
> # The scatter plots & correlation coefficients show strong correlation among the
> # predictors. In addition, the VIF > 10 for X1,X2 and X3, so the assumption of
> # linearly independence of each predictor is violated; there is a
> # multicollinearity problem.
```

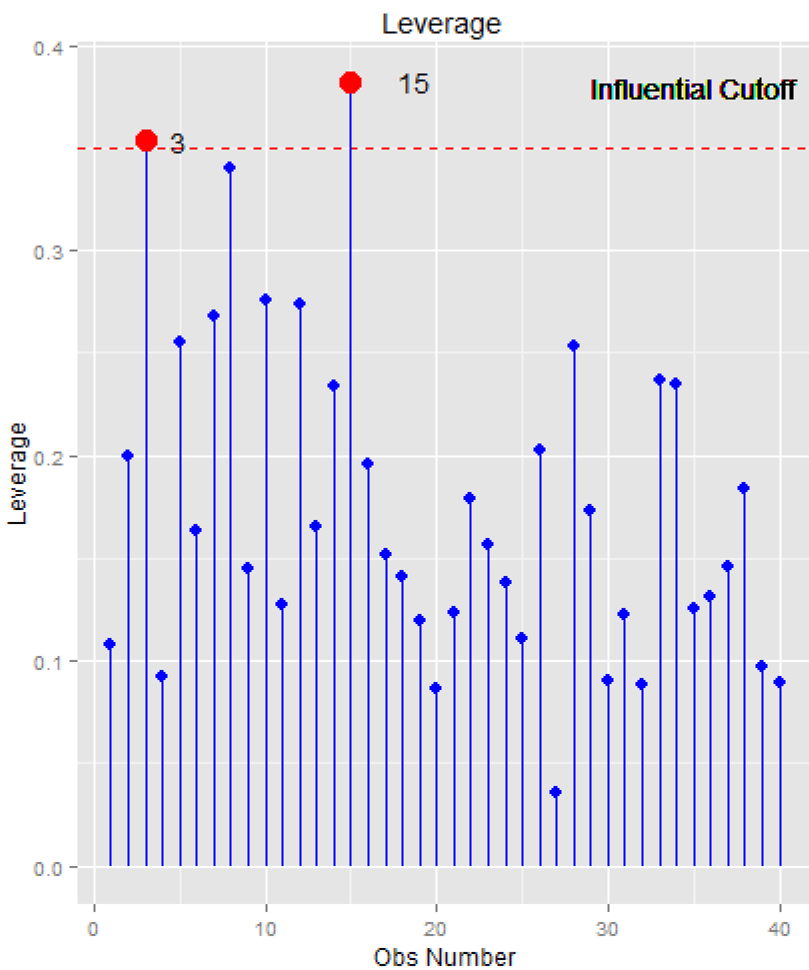


Correlation coeffficients matrix and scatter plot



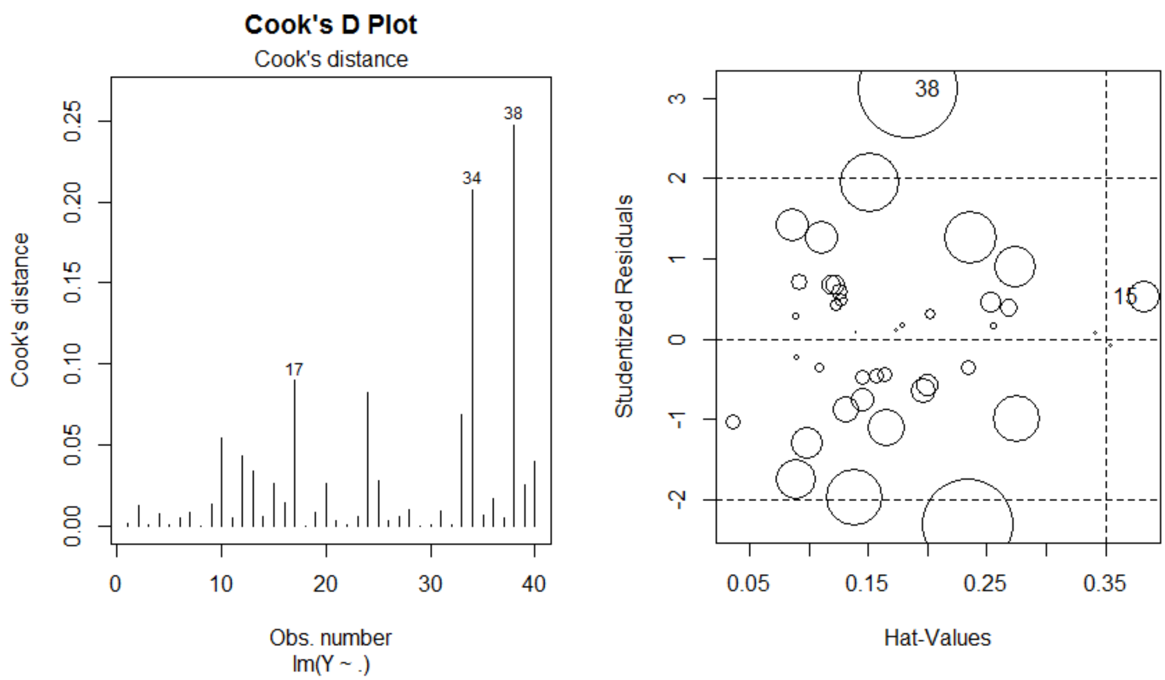
Part d.

```
> ## Compute Leverage for measuring "unusualness" of x's
> leverage =  hat(model.matrix(fit))
> mydata$leverage = leverage

> # Compute cutoff
> p=6
> n=dim(mydata)[1]
> cutoff = 2*(p+1)/n
> cutoff
[1] 0.35
>
> # Find high leverage points
> influential = mydata["leverage"]
> influential = subset(influential,leverage> cutoff)
> influential
    leverage
3  0.3535547
15 0.3823626
```



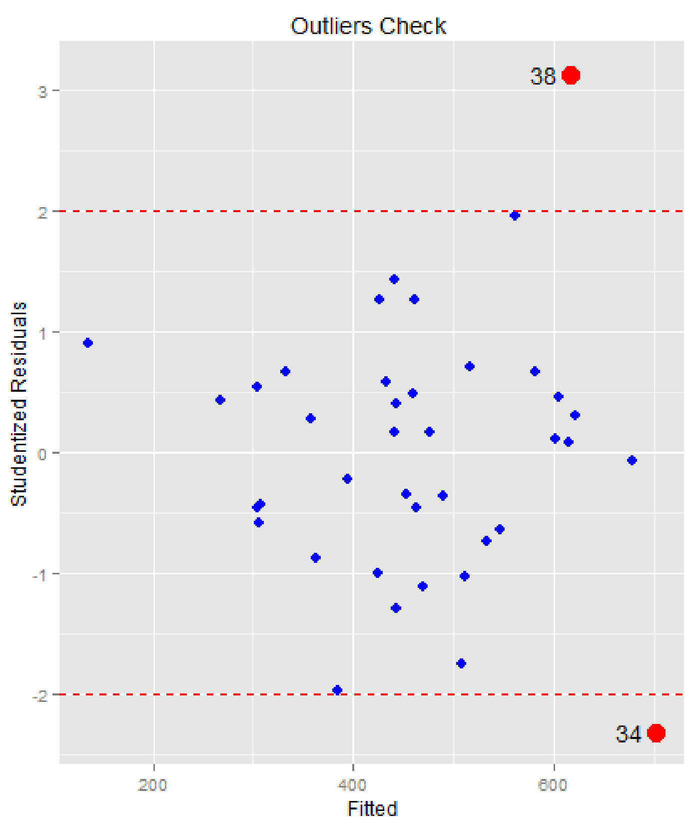

```
> # Using the rule of thumb (hii>2(p+1)/2), the observations
> # 3 and 15 are regarded as high leverage points
```

```
> ## Compute Cook distances for for measuring influence
> # Cook's D plot

> cutoff = 4/(dim(mydata)[1]);
> plot(fit, which=4, cook.levels=cutoff, main = "Cook's D Plot");

> # influence plot
> library(car)    # needed for "influencePlot" function below
> influencePlot(fit)
     StudRes       Hat     CookD
15 0.5357756 0.3823626 0.1610822
38 3.1195250 0.1837930 0.4975412
```




```
> # point 17, 34 and 38 are influential points (using cutoff 4/n)

> # The circles for each observation represent the relative size of the Cook's D
> # point 15 is high leverage, and 38 is an outlier with high influence

> # Outliers/high leverage/influential points
> summary(influence.measures(fit))
Potentially influential observations of
         lm(formula = Y ~ ., data = mydata) :

   dfb.1_ dfb.X1 dfb.X2 dfb.X3 dfb.X4 dfb.X5 dfb.X6 dffit   cov.r   cook.d hat
3   0.01   0.02   0.02   0.02   0.03   0.00  -0.02  -0.06    1.92_*  0.00   0.35
5   0.01   0.06   0.06   0.06   0.03  -0.03  -0.06   0.09    1.66_*  0.00   0.25
7   0.11   0.12   0.11   0.11  -0.09   0.03  -0.12   0.24    1.64_*  0.01   0.27
8  -0.04  -0.02  -0.02  -0.02   0.02   0.04   0.02   0.05    1.88_*  0.00   0.34
15  0.17   0.23   0.23   0.24  -0.17   0.17  -0.23   0.42    1.89_*  0.03   0.38
38 -0.49   0.54   0.55   0.54   0.95  -0.03  -0.54   1.48_*  0.24_*  0.25   0.18

> # Using the R function, potential problematic points are: 3,5,7,8,15 and 38
```

```
> ## Studentized residuals vs fitted
> library(MASS)
> stu_res = studres(fit)
> mydata$stu_res = stu_res
> mydata$fitted = fit$fitted

> # Find outliers points
> outlier = mydata["stu_res"]
> outlier = subset(outlier,abs(stu_res)>2)
> outlier
     stu_res
34 -2.315259
38  3.119525
> #rownames(mydata)[abs(stu_res) > 2]

> outlier$fitted = fit$fitted[outlier$obs]
> outlier$obs = as.numeric(rownames(outlier))
> ggplot(mydata,aes(x=fitted, stu_res)) +
+   geom_point(size = 3, color="blue") +
+   geom_hline(yintercept=2, linetype="dashed" , color = "red") +
+   geom_hline(yintercept=-2, linetype="dashed" , color = "red") +
+   geom_text(data =outlier, aes(x=fitted, y = stu_res,
+                                   label = obs), hjust = 1.5) +
+   labs(title="Outliers Check ",
+        x = "Fitted",
+        y = "Studentized Residuals") +
+   geom_point(data=mydata[outlier$obs,], colour="red", size=5)
```

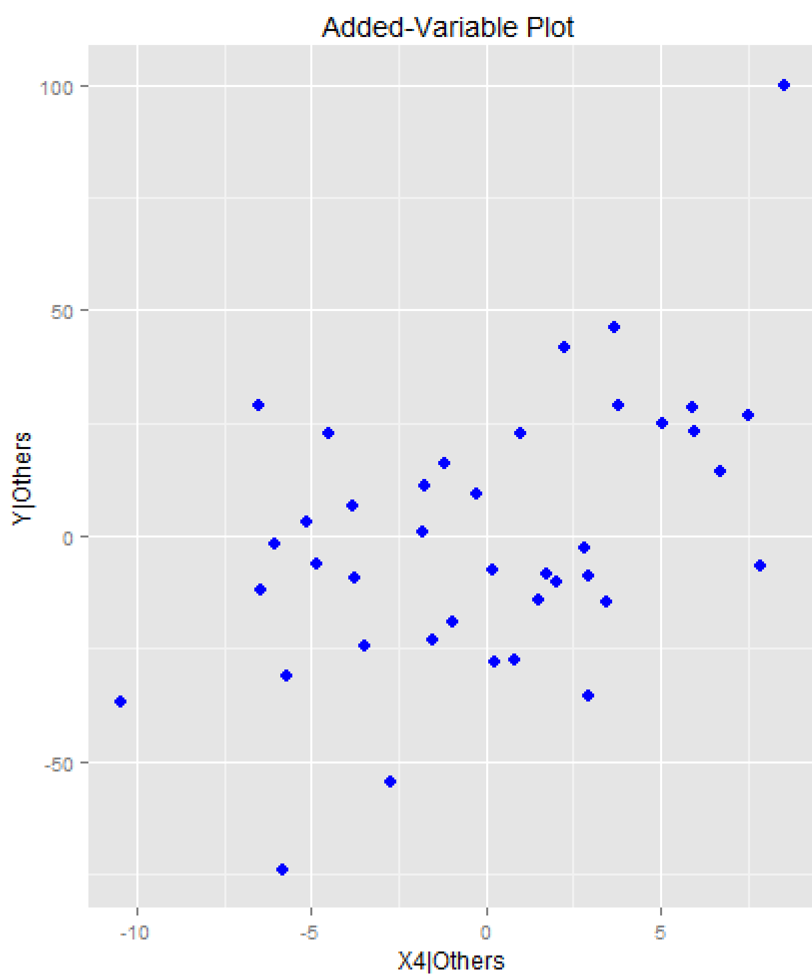



```
> # Using the rule that |studentized residuals| > 2, the observations
> # 34 and 38 are regarded as outliers
```

Problem 3 (4.13)

Part a.

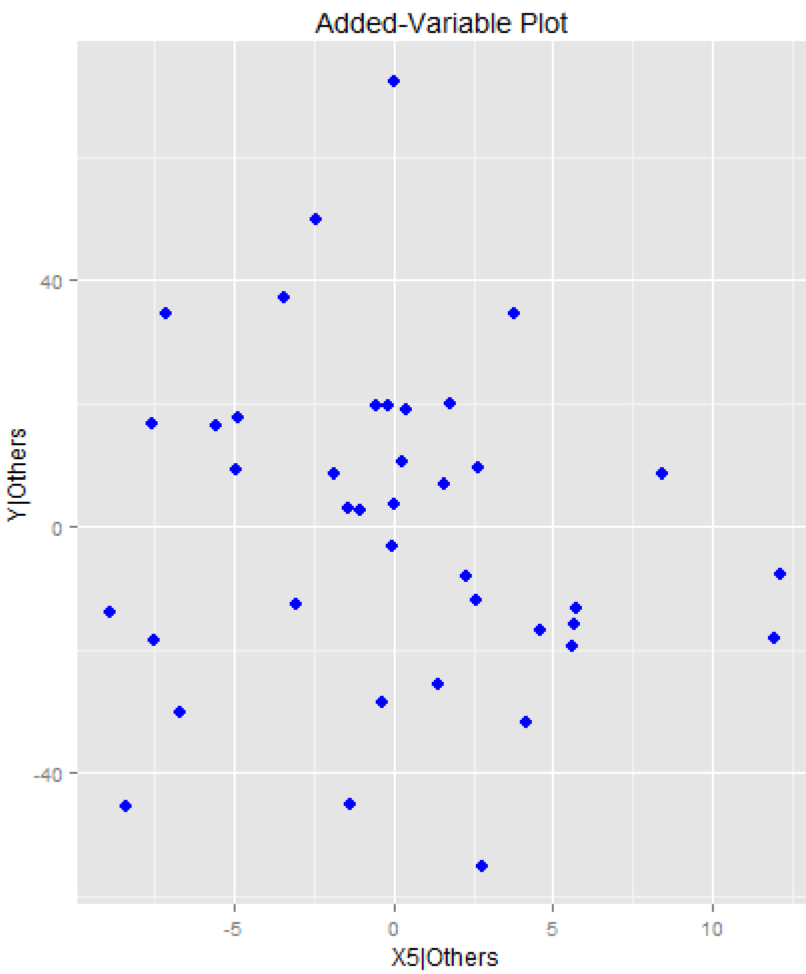
```
> fit_y = lm(Y~ X1+X2+X3,mydata)
> fit_x = lm(X4~ X1+X2+X3,mydata)
> data = data.frame(x=fit_x$res,y=fit_y$res)
>
> ggplot(data,aes(x,y)) +
+   geom_point(size = 3, color="blue") +
+   labs(title="Added-Variable Plot ",
+        x = "X4|Others",
+        y = "Y|Others")
```




```
>
> # The partial regression plots shows a linear relationship, thus
> # X4 makes a marginal contribution to y given other predictors are already
> # in the model. Conclusion: Add X4 to model
```

Part b.

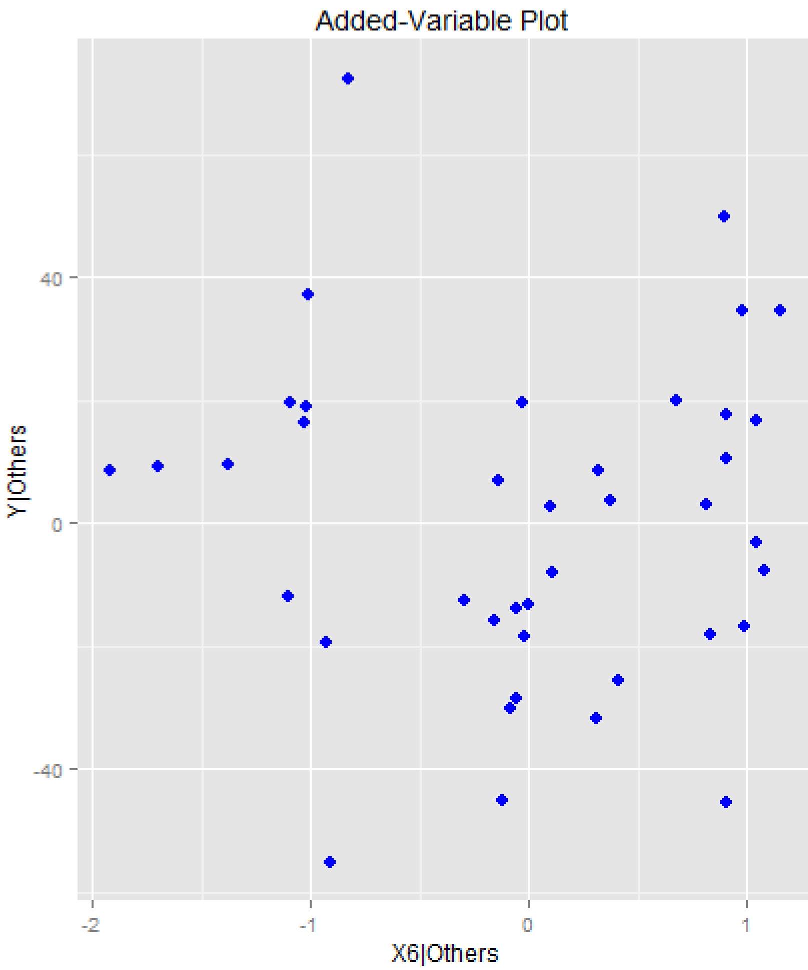
```
> fit_y = update(fit_y,.~.+X4)
> fit_x = lm(X5~ X1+X2+X3+X4,mydata)
> data = data.frame(x=fit_x$res,y=fit_y$res)
>
> ggplot(data,aes(x,y)) +
+   geom_point(size = 3, color="blue") +
+   labs(title="Added-Variable Plot ",
+         x = "X5|Others",
+         y = "Y|Others")
```




```
>
> # The partial regression plots looks random, thus
> # X5 makes no marginal contribution to y given other predictors are already
> # in the model. Conclusion: Do not add X5
```

Part c.

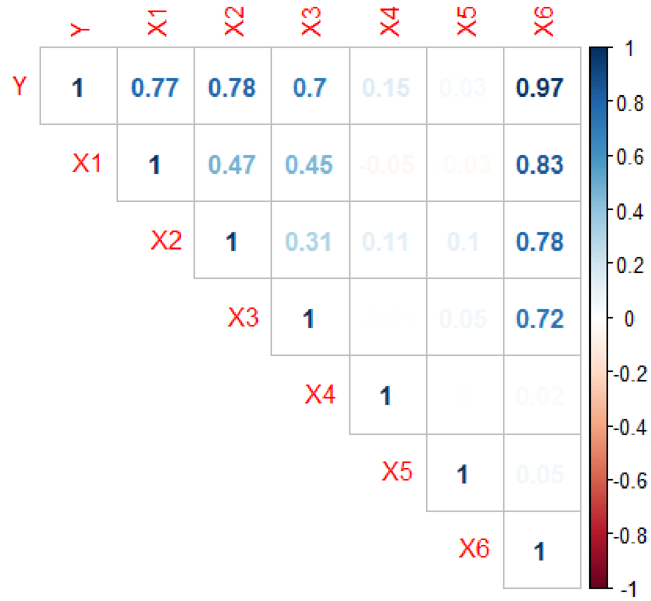
```
> fit_x = lm(X6~ X1+X2+X3+X4,mydata)
> data = data.frame(x=fit_x$res,y=fit_y$res)
>
> ggplot(data,aes(x,y)) +
+   geom_point(size = 3, color="blue") +
+   labs(title="Added-Variable Plot ",
+        x = "X6|Others",
+        y = "Y|Others")
```

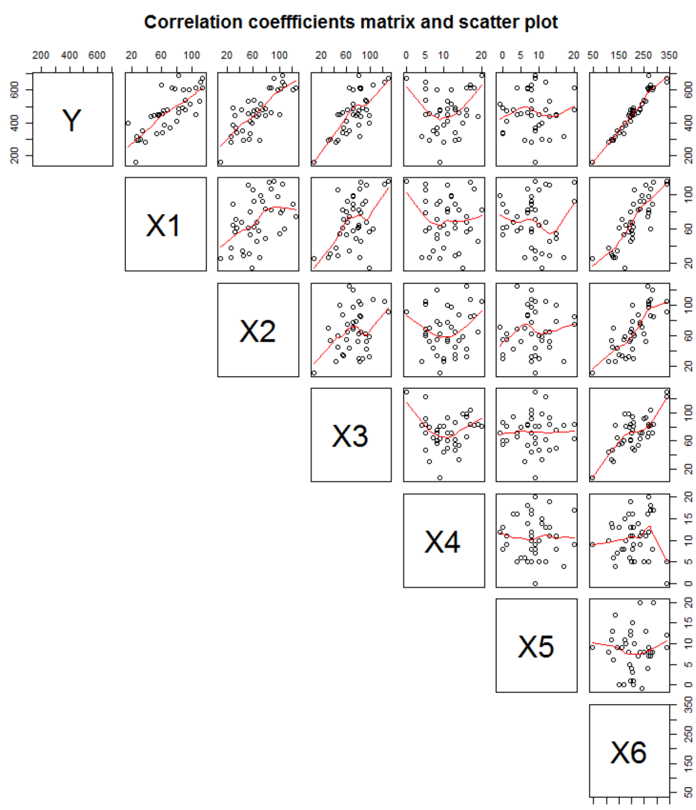



```
>
> # The partial regression plots looks random, thus
> # X6 makes no marginal contribution to y given other predictors are already
> # in the model. Conclusion: Do not add X6.
```

Part d.

```
> # Look at correlation
> corr = round(cor(mydata),2)
> corrplot(corr,method="number", type="upper")
> pairs(mydata, main = "Correlation coeffficients matrix and scatter plot",
+       pch = 21, lower.panel = NULL, panel = panel.smooth, cex.labels = 3)
```



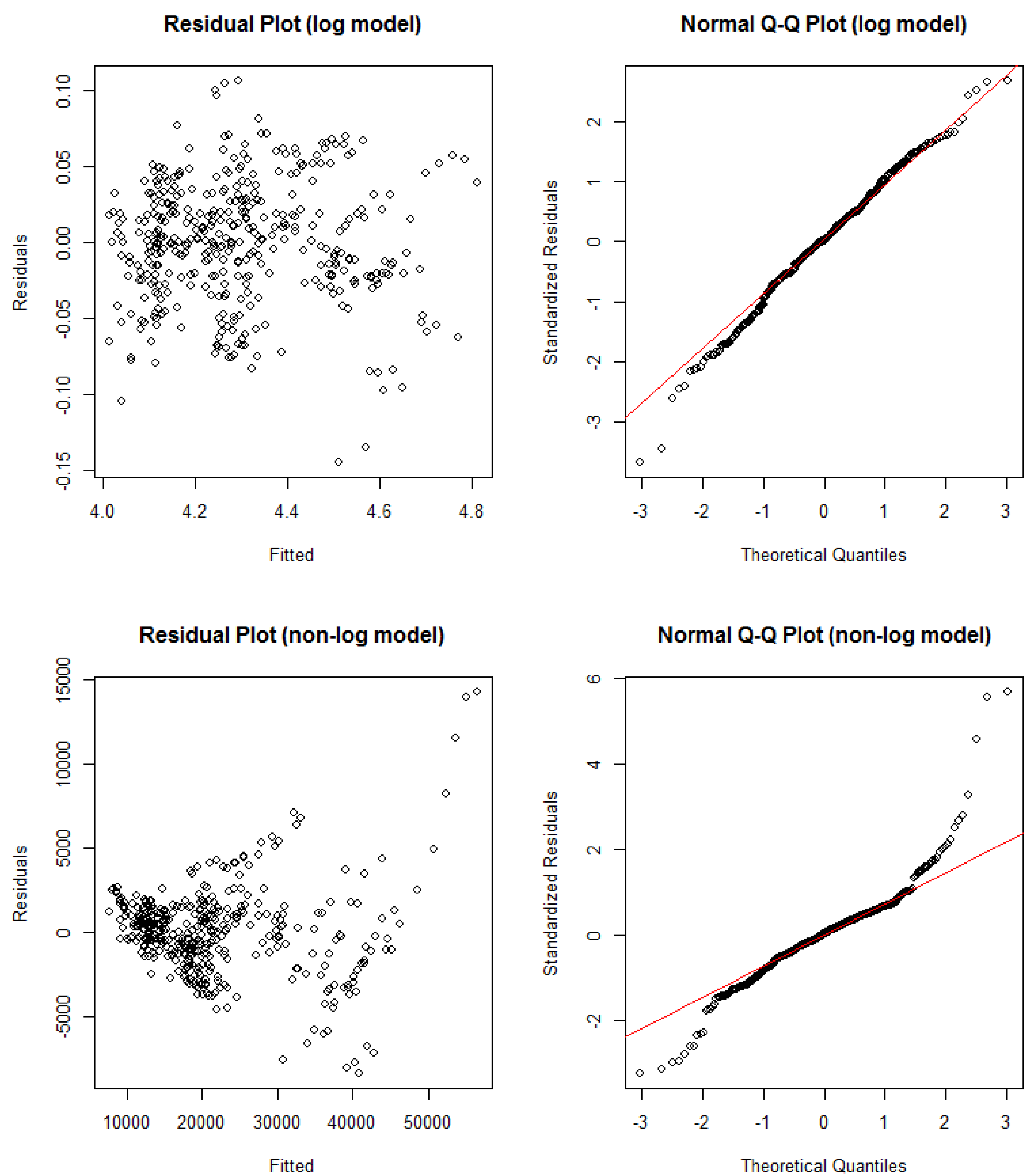


```
> # Checking VIF after adding X6 to X1+X2+X3+X4
> fitC = lm(Y~X1 + X2 + X3 + X4 + X6,mydata)
> vif(fitC)
         X1          X2          X3          X4          X6 
1059.667675 1100.074999  762.154177    1.035356 5296.041702 

> # The scatter plots and correlation coefficients shows that
> # X6 is strongly correlated with X1, X2 and X3.
> # In addition, the VIF > 10 for X1,X2 and X3, so the assumption of
> # linearly independence of each predictor is violated; there is a
> # multicollinearity problem.

> fitA = lm(Y~X1+X2+X3+X4,mydata)
> fitD = lm(Y~X4+X6,mydata)
> summary(fitA)

Call:
lm(formula = Y ~ X1 + X2 + X3 + X4, data = mydata)

Residuals:
   Min     1Q Median     3Q    Max 
-55.05 -17.03   2.83  17.08  72.40 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  28.3469    18.9141   1.499  0.14291    
X1            1.7006     0.1958   8.684 2.97e-10 ***
X2            2.0907     0.1809  11.558 1.68e-13 ***
X3            2.0209     0.2117   9.544 2.83e-11 ***
X4            3.2295     0.9654   3.345  0.00197 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.92 on 35 degrees of freedom
Multiple R-squared: 0.9551,    Adjusted R-squared:  0.95 
F-statistic: 186.3 on 4 and 35 DF,  p-value: < 2.2e-16

> summary(fitD)

Call:
lm(formula = Y ~ X4 + X6, data = mydata)

Residuals:
    Min      1Q  Median      3Q     Max 
-50.755 -17.635  -1.464  18.106  76.589 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  26.2770    18.4239   1.426 0.162183    
X4            3.4288     0.9539   3.594 0.000943 ***
X6            1.9275     0.0715  26.957  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.9 on 37 degrees of freedom
Multiple R-squared: 0.9526,    Adjusted R-squared: 0.9501 
F-statistic: 372.2 on 2 and 37 DF,  p-value: < 2.2e-16

> 
> # Thus, either use X1, X2 and X3 or X6 in the model
> # Both models (X1+X2+X3+X4 and X4+X6) show similar R-squared that
> # are very high, so either model can be used as the best possible
> # description of Y. The model X4+X6 might be preferred because
> # it is the smaller model and thus simpler.
```

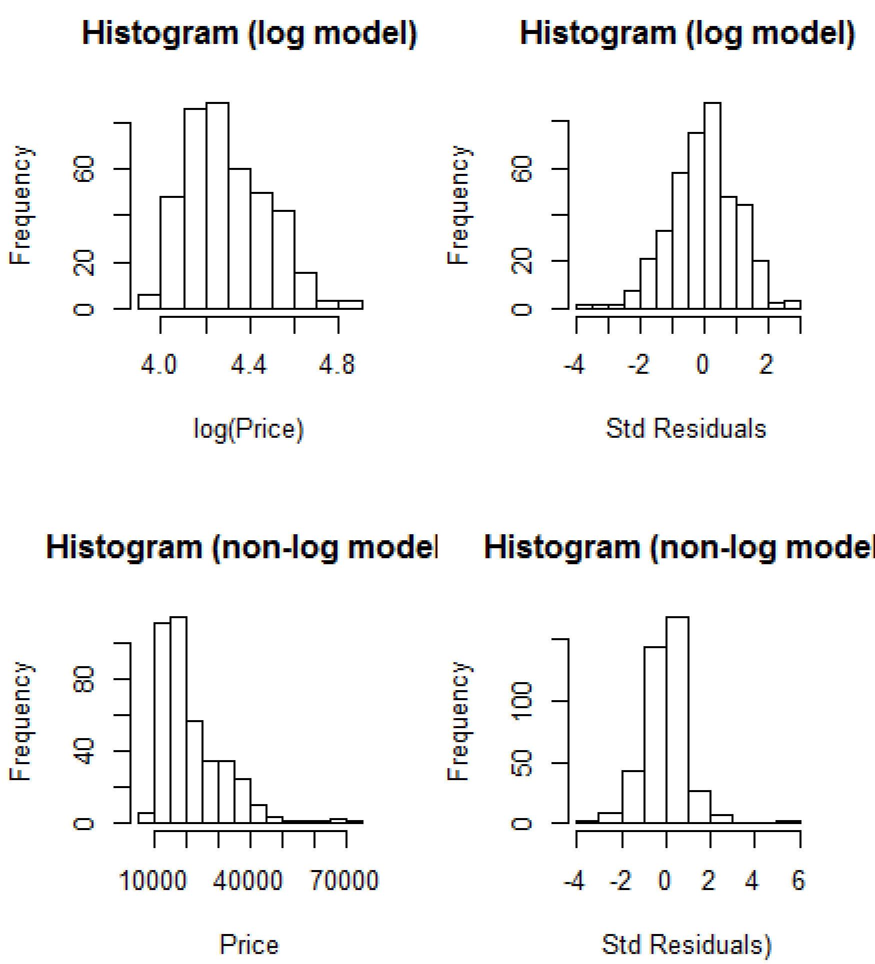
Problem 4

```
> # Import data
> filename = "Used+car+prices+%28Training+set%29.csv"
> mydata = read.csv(filename,header = T)
> # Log price
> colnames(mydata)[13] = "log_Price"
>
> # Fit log model
>
> fit = lm(log_Price ~ Mileage + Liter + Make + Type, mydata)
> summary(fit)

Call:
lm(formula = log_Price ~ Mileage + Liter + Make + Type, data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.14425 -0.02276  0.00067  0.02588  0.10642 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)    4.169e+00  1.558e-02 267.655  < 2e-16 ***
Mileage       -3.545e-06  2.499e-07 -14.184  < 2e-16 ***
Liter          9.758e-02  2.307e-03  42.291  < 2e-16 ***
MakeCadillac   1.948e-01  9.193e-03  21.188  < 2e-16 ***
MakeChevrolet -5.401e-02  7.805e-03  -6.920 1.87e-11 ***
MakePontiac   -4.133e-02  8.186e-03  -5.049 6.84e-07 ***
MakeSAAB       2.463e-01  9.733e-03  25.303  < 2e-16 ***
MakeSaturn    -4.629e-02  1.043e-02  -4.439 1.18e-05 ***
TypeCoupe     -1.337e-01  1.049e-02 -12.743  < 2e-16 ***
TypeHatchback -1.574e-01  1.220e-02 -12.907  < 2e-16 ***
TypeSedan     -1.412e-01  9.242e-03 -15.283  < 2e-16 ***
TypeWagon     -7.138e-02  1.143e-02  -6.243 1.12e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04017 on 390 degrees of freedom
Multiple R-squared: 0.9513,	Adjusted R-squared: 0.9499 
F-statistic: 692.4 on 11 and 390 DF,  p-value: < 2.2e-16


>
> # Fit non-log model
> fit2 = lm(Price ~ Mileage + Liter + Make + Type, mydata)
>
> # 4 by 4 grid
> par(mfrow=c(2,2))
>
> # Residual vs fitted log model
> plot(fit$fitted,fit$resid,
+      ylab = "Residuals",
+      xlab = "Fitted",
+      main = "Residual Plot (log model)")
> # Normal plot log model
> fit_stdres = rstandard(fit)
> qqnorm(fit_stdres,
+        ylab = "Standardized Residuals",
+        xlab = "Theoretical Quantiles",
+        main = "Normal Q-Q Plot (log model)");
> qqline(fit_stdres, col="red")
>
```

```
> # Residual vs fitted non-log model
> plot(fit2$fitted,fit2$resid,
+      ylab = "Residuals",
+      xlab = "Fitted",
+      main = "Residual Plot (non-log model)")

> # Normal plot non- log model
> fit2_stdres = rstandard(fit2)
> qqnorm(fit2_stdres,
+        ylab = "Standardized Residuals",
+        xlab = "Theoretical Quantiles",
+        main = "Normal Q-Q Plot (non-log model)")
> qqline(fit2_stdres, col="red")
>
```



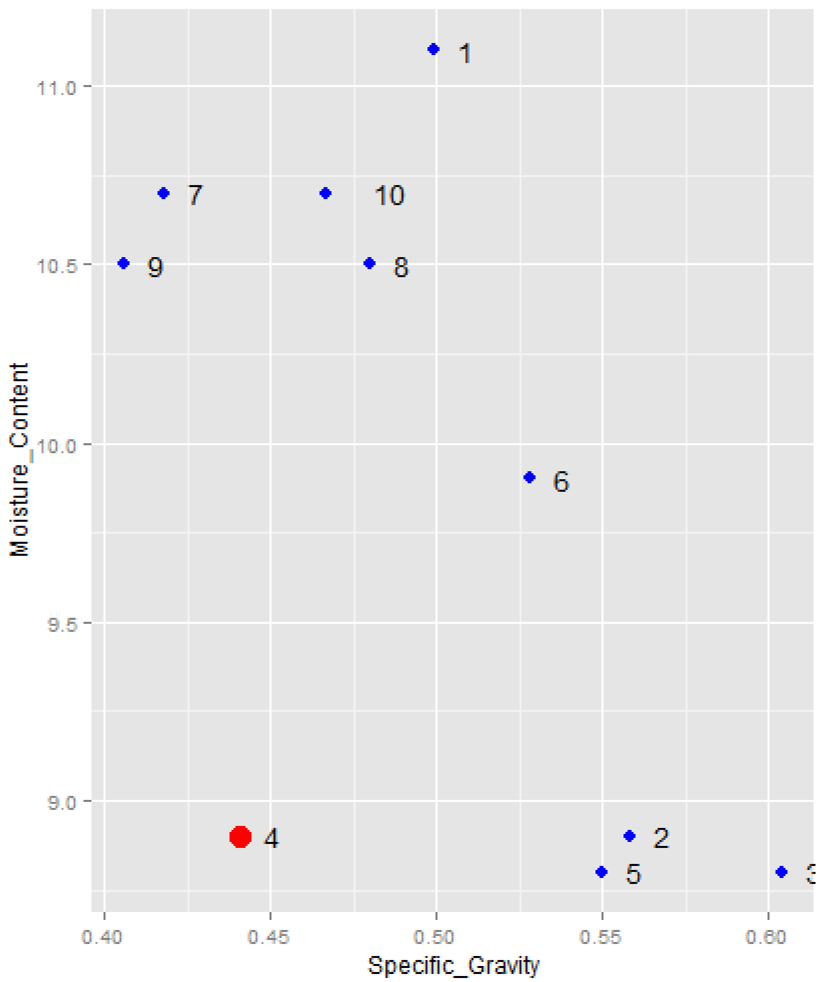
```
> # The plots for the log model look more satisfactory.
> # The residuals plot shows that points are scattered randomly
> # about zero.
> # Overall the standardized residuals seem to fit a straight
> # line with the normal scores.
>
> # On the other hand, the residual plot for the non-log model
> # shows a clear fanning pattern where the residuals increase
> # as the size of the fitted value increase.
> # The Q-Q plot also shows large deviations from the straight
> # line in the tails (distribution looks like long tails in upper)
>
> # A log transformation is clearly beneficial to make the
> # assumptions of linear regression hold because it shrinks
> # the distribution (e.g. shrinks values of Prices in such a
> # way that large values of Prices are affected much more than
> # small values are) as can be seen from the residuals and
> # Q-Q plots after applying the transformation. More specifically,
> # the log transformation is useful in stabilizing the variance
> # because it shrinks the upper tail of the data and help make the
> # variance constant, satisfying homoscedasticity, which in turns
> # helps improve normality of the response variable (price).
```



Problem 5

Part a.

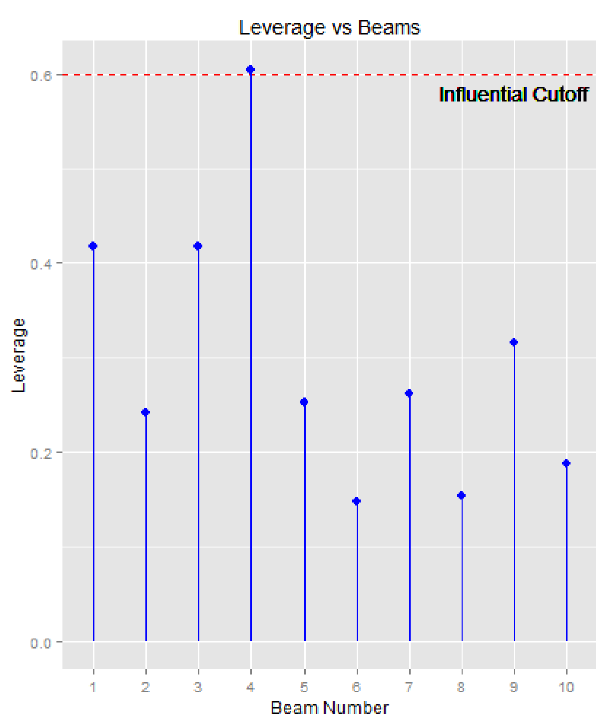
```
> ggplot(mydata,aes(x=Specific_Gravity, y = Moisture_Content)) +
+   geom_point(size = 3, color="blue") +
+   geom_text(data = mydata, aes(x=Specific_Gravity, y = Moisture_Content,
+                              label = Beam_Number), hjust = -1.5) +
+   geom_point(data=mydata[4, ], colour="red", size=5)
```




```
> # It appears that beam number 4 to be an outlier in terms of specific gravity
> # and moisture content because its value does not fall into the general pattern
> # of association between the variables. Beam 4 seems to be very in low in both
> # specific gravity and moisture content compared to the other beams.
> # More precisely, Beam 4 can be described as a bivariate outlier,
> # that is an outlier that occurs within the joint combination of two (bivariate)
> # variables (Note also that beam 1 can be also potentially be an outlier)
```

Part b.

```
> # Fit Regression
> fit = lm(Strength ~ Specific_Gravity + Moisture_Content, mydata)
>
> # Compute Leverage
> leverage =  hat(model.matrix(fit))
> mydata$leverage = leverage
>
> # Compute cutoff for influential
> p=2
> n=dim(mydata)[1]
> cutoff = 2*(p+1)/n
> cutoff
[1] 0.6
>
> # Find influential points
> names(leverage)=mydata$Beam_Number
> leverage
        1         2         3         4         5         6         7         8         9
0.4178935 0.2418666 0.4172806 0.6043904 0.2521824 0.1478688 0.2616385 0.1540321 0.3155106
       10
0.1873364
> mydata$Beam_Number[leverage > cutoff]
[1] 4
> # Plot influential points
> ggplot(mydata,aes(x=factor(Beam_Number), leverage)) +
+   geom_point(size = 3, color="blue") +
+   labs(title="Leverage vs Beams",
+        x = "Beam Number",
+        y = "Leverage") +
+   geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
+   geom_text(aes(9, .58, label="Influential Cutoff")) +
+   geom_segment(aes(xend=Beam_Number, yend=0), color="blue")
```



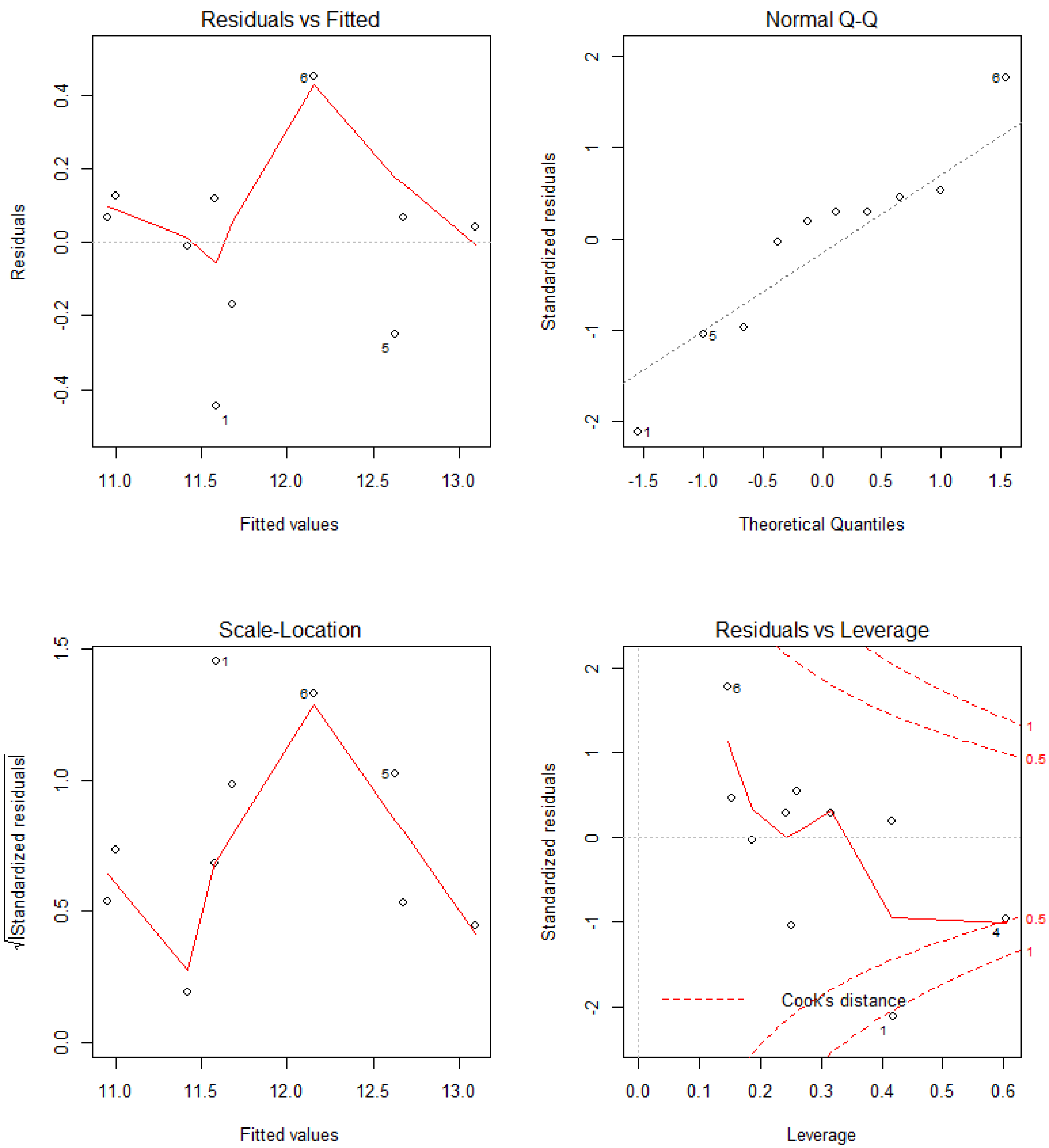

```
> # Yes, beam 4 identified as an outlier is an influential observation
> # using the rule h_ii = 0.604 > 2*(2+1)/10 = 0.6
```

Part c.

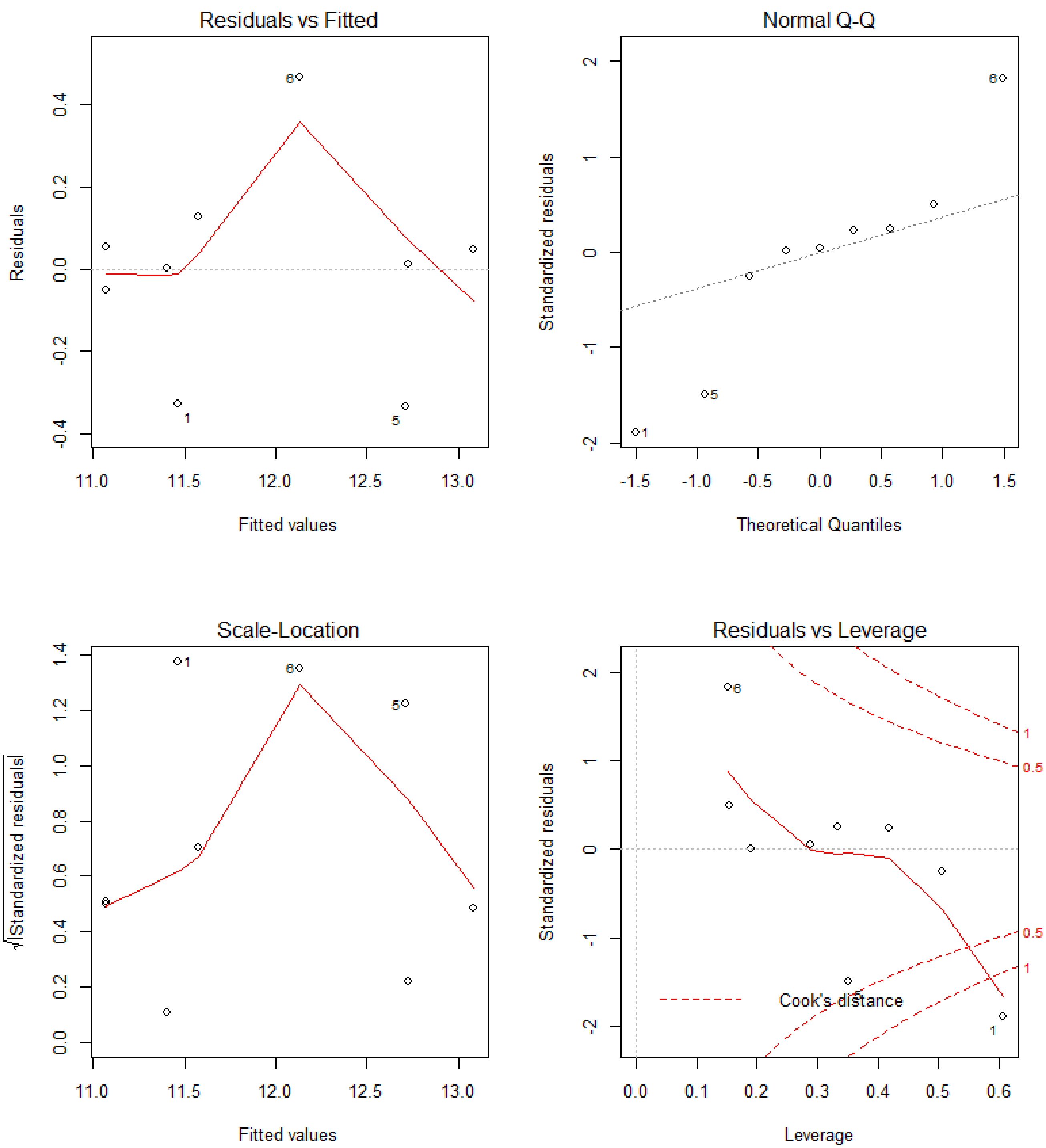
```
> # Fit regression without influential observation
> mydata2 = mydata
> mydata2 = mydata2[!(leverage > cutoff),]
>
> fit2 = lm(Strength ~ Specific_Gravity + Moisture_Content, mydata2)
>
> # Compare regressions from all data and without influential observation
> summary(fit)

Call:
lm(formula = Strength ~ Specific_Gravity + Moisture_Content,
    data = mydata)

Residuals:
     Min       1Q   Median       3Q      Max
-0.44422 -0.12780  0.05365  0.10521  0.44985

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       10.3015     1.8965   5.432 0.000975 ***
Specific_Gravity   8.4947     1.7850   4.759 0.002062 **
Moisture_Content  -0.2663     0.1237  -2.152 0.068394 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2754 on 7 degrees of freedom
Multiple R-squared:    0.9,     Adjusted R-squared: 0.8714
F-statistic:  31.5 on 2 and 7 DF,  p-value: 0.0003163

> summary(fit2)

Call:
lm(formula = Strength ~ Specific_Gravity + Moisture_Content,
    data = mydata2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.33339 -0.05037  0.01127  0.05615  0.46579

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       12.4107     2.9071   4.269  0.00527 **
Specific_Gravity   6.7992     2.5166   2.702  0.03549 *
Moisture_Content  -0.3905     0.1794  -2.177  0.07237 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.277 on 6 degrees of freedom
Multiple R-squared: 0.9108,     Adjusted R-squared: 0.8811
F-statistic: 30.65 on 2 and 6 DF,  p-value: 0.0007089

> # from stackoverflow
> percent <- function(x, digits = 2, format = "f", ...) {
+   paste0(formatC(100 * x, format = format, digits = digits, ...), "%")
+ }
>
> # percent change in magnitude
> p = percent((fit2$coeff-fit$coeff)/fit$coeff)
> names(p)=names(fit$coeff)
> p
     (Intercept) Specific_Gravity Moisture_Content
        "20.47%"         "-19.96%"         "46.65%"
```

```
> # Looking at the coefficients, it seems that the fit changed.
> # The coefficient for Specific_Gravity changed from 8.4947 to 6.799 (-20%)
> # and the coefficient for Moisture_Content changed from -0.2663 to -0.3905 (+47%).
> # The intercept also changed from 10.3015 to 12.4107 (+20%)
> # Note: percent changes refer to percent change in the magnitude of the coefficient

> # The significance of the coefficient at 0.05 level did not change, and
> # the R-squared increased slightly from 0.9 to 0.91.
> # The diagnostics plots look very similar for both models (although after removing beam 4
> # , beam 1 shows up as a very influential point in the diagnostics using Cook's D. Beam 1
> # seems to have a lower leverage than beam 4, but be more influential (higher Cook's))
> #
> # Because the influential point (beam 4) does not seem to follow the relationship
> # in terms of specific gravity and moisture content of the other beams, and also
> # has a big impact on the effect of the predictors variables and thus the prediction
> # of the strength, then the fitted equation after removing the influential
> # point should be used to predict the wood beam strength since the prediction
> # would not be heavily influenced by one data point. Further analysis and different
> # criteria for influential points should be used to address, for example, beam 1.

> par(mfrow=c(2,2))
> plot(fit)
> par(mfrow=c(2,2))
> plot(fit2)
```

Model with all data points



Model with beam 4 removed



R-Code

```
#### Homework 4
#### From the text-book: Problems 1, 2, 3: Exercises 4.7, 4.12 (only do parts (a) and (d)) and 4.13.

#### Setup ##################################################

# Install packages if needed
# install.packages("ggplot2")
# install.packages("grid")
# install.packages("gridExtra")
# install.packages("XLConnect")
# install.packages("corrplot")
# install.packages("Hmisc")
# install.packages("car")

# Load packages
library(ggplot2)
library(grid)
library(gridExtra)
library(XLConnect)
library(corrplot)
library(Hmisc)
library(car)
library(MASS)


# My PC
main = "C:/Users/Steven/Documents/Academics/3_Graduate School/2014-2015 ~ NU/"

# Aginity
#main = "\\\\nas1/labuser169"

course = "MSIA_401_Statistical Methods for Data Mining"
datafolder = "Data"
setwd(file.path(main,course, datafolder))

#### Problem 1 ##############################################
# Import data
filename = "P088.txt"
mydata = read.table(filename,header = T)

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

### Part a

# Sales vs Price ~ Expected negative relationship, since
# the higher the price, the less likely people are going to spend money
# to buy cigarettes.

# Sales vs Income ~ Expected positive relationship, since
# the higher the income, the more money available to spend

# Sales vs Age ~ Expected positive relationship, since
# older people tend to consume more cigarettes compared to younger people,
# and older people tend to have more income

# Sales vs HS ~ Expected positive relationship since high school completion
# is positively related to income. However, it is likely not to be a strong
# relationship so no relationship might be expected since preferences for
# smoking (and buying) cigarettes seems to be similar across different
```

```
# education backrounds

# Sales vs Back ~ Expected positive relationship because surveys have
# shown that African American tend to smoke more than other races

# Sales vs Female ~ Expected negative relationship because surveys have
# shown that men tend to smoke more than women.

### Part b

corr = round(cor(mydata[-1]),2)
corr

# Plot combine correlation coefficients matrix and scatter plot
# http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/iris_plots/
panel.pearson <- function(x, y, ...) {
  horizontal <- (par("usr")[1] + par("usr")[2]) / 2;
  vertical <- (par("usr")[3] + par("usr")[4]) / 2;
  text(horizontal, vertical, format(cor(x,y), digits=2))
}

pairs(mydata[,-1], main = "Correlation coeffficients matrix and scatter plot",
     pch = 21, upper.panel=panel.pearson,lower.panel = panel.smooth)

# Plot separate
corrplot(corr,method="number", type="upper")

pairs(mydata[,-1], main = "Correlation coeffficients matrix and scatter plot",
     pch = 21, lower.panel = NULL, panel = panel.smooth, cex.labels=2)

### Part c

# No disagreement. The sign of the correlation coefficient tells you the direction
# of the linear relationship, and the magnitude tells you the strength of the
# linear relationship.
# The scatter plot is a visual representation of the correlation coefficient,
# where high correlation is shown when data points are clustered tightly together
# around a line, and the sign is shown by the direction of the association of the
# points (e.g. pointing down means as one variable increases, the other decreases,
# so negative correlation).
# So for example, price and sales have a correlation of -0.33, and this is shown
# in the scatter plot as price increase, sales decreases.

### Part d

# For the most part the expectations in part(a) match the pairwise correlation
# coefficeints matrix and the corresponding scatter plot. For example, as expected,
# Sale and price are negatively correlated, while Sale and income are positively
# correlated. As expected also, these relationships are the strongest. The only
# one that did not match the expectations is the positive relationship between female and
# sales, where it was expected the relationship to be negative. However, the relationship
# betwen female and sales (correlation = 0.15) does not seem very srong.

### Part e

fit = lm(Sales ~ Age + HS + Income + Black + Female + Price, mydata)
summary(fit)

# The coefficient of Age, Income and black are positive, so they match the expectation
# in part (a) as it is expected that an increase in these variables would lead to an
# increase in the sales.

# The coefficient of Price is negative, so they so it matches the expectation
# in part (a) as it is expected that an increase in this variable would lead to a
# decrease in the sales.
```

```
# The coefficeint of Female is negative, so they so it matches the expectation
# in part (a) as it is expected that more females than males would lead to a decrease
# in sales because males tend to smoke more.

# The coefficeint of HS is negative, so it does not match the expectation
# of positive relationship with sale. However, it was also expected that the relationship
# between HS and sales to be very weak or none at all, and this is consistent with the small
# magnitude of the regression coefficient and high p-value indicating insignficant effect
# (not signficantly different than zero)

### Part f

compare = data.frame(
  corr =corr[-7,"Sales"],
  coeff=round(summary(fit)$coeff[-1,"Estimate"],2),
  p_val=round(summary(fit)$coeff[-1,"Pr(>|t|)"],2))

compare

# There are differences between the pairwise correlation coefficeints and the
# correlation coefficients between Sales and each of the predictors. All of them
# agree in terms of sign/direction except Female and HS. This can be explained by
# the fact that the regression coefficient tells you the effect of a variable after
# accounting for the other predictors, while the correlation coefficient measures
# the pairwise relationship between two variables. Thus, it might be the case that
# a variable to have an opposite effect when other variables have been taken into
# account (e.g. female effect after controlling for income) because there might
# be a lurking variable confounding the relationship in the pairwise correlation coefficients.
# The pairwise correlation  ignores the fact that there is a more plausible lurking variable
# giving rise to the observed correlation. So the effects of regression coefficients depend
# on the presence of  other predictors in the model.

# Example, the regression of sales vs. only female (no other variables taken into account)
# agrees with the sign of the pairwise correlation coefficient. But when other variables
# are added, then the sign changes. Similar results is observed for HS.

lm(Sales ~ Female ,mydata)
lm(Sales ~ HS ,mydata)


# It is also important to point out that the p-values say for example that the
# effect of HS  and Female after taking other predictors into account is insignificant
# (not significantly different than zero).


### Part g

# In order to test if there is anything wrong the tests and conclusions reached
# in 3.15, we need to run a diagnostics to check all the assumptions hold.

# Assumptions about the predictors
# prefer cook distance since both x and y space

## check the fit (check linearity assumption by plotting residuals against each predictor)


plot_vector = vector(mode="list",length=6)

plot_vector[[1]] = ggplot(mydata,aes(x=mydata[[2]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[2]),y = "Residuals")

plot_vector[[2]]  = ggplot(mydata,aes(x=mydata[[3]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[3]),y = "Residuals")
```

```
plot_vector[[3]]  = ggplot(mydata,aes(x=mydata[[4]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[4]),y = "Residuals")

plot_vector[[4]]  = ggplot(mydata,aes(x=mydata[[5]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[5]),y = "Residuals")

plot_vector[[5]]  = ggplot(mydata,aes(x=mydata[[6]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[6]),y = "Residuals")

plot_vector[[6]] = ggplot(mydata,aes(x=mydata[[7]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[7]),y = "Residuals")

grid.arrange(plot_vector[[1]],
             plot_vector[[2]],
             plot_vector[[3]],
             plot_vector[[4]],
             plot_vector[[5]],
             plot_vector[[6]],
             ncol=2, main = "Residuals vs Predictor Variables")

# All plots look random so assumptions about the form of the model
# (linear in the regression parameters) is satisfied.

## Can also do the followig:
## check the fit (check linearity assumption by plotting partail regression/added variable plot)
library(car)
avPlots(fit)

## check normality (using qq plot)
qqPlot(fit, main = "Normal Q-Q Plot")

# Alternative code:
fit_stdres = rstandard(fit)

# To get studentized residuals
library(MASS)
stu_res = studres(fit)

qqnorm(fit_stdres,
       ylab = "Standardized Residuals",
       xlab = "Theoretical Quantiles",
       main = "Normal Q-Q Plot");
qqline(fit_stdres, col="red")

# formal test: Anderson-Darling test, Shapiro-Wilk test, Kolomogorov-Smirnov

# The plot shows that most points fall along the line, indicating the normality
# assumption of errors is satisfied. However, it looks that there are a fwew outliers

## check Checking Homoscedasticity (using residuals vs. fitted)

ggplot(mydata,aes(x=fit$fitted, y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = "Fitted",y = "Residuals")

# The plot shows data points are random forming a parallel band, indicating the
# common variance assumption of errors is valid (Homoscedasticity)

## check independence (not time series data)
# the Durbin-Watson

## Check multicollinearity
```

```
corr = round(cor(mydata[-1]),2)
corr

# Plot combine correlation coefficients matrix and scatter plot
# http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/iris_plots/
panel.pearson <- function(x, y, ...) {
  horizontal <- (par("usr")[1] + par("usr")[2]) / 2;
  vertical <- (par("usr")[3] + par("usr")[4]) / 2;
  text(horizontal, vertical, format(cor(x,y), digits=2))
}

pairs(mydata, main = "Correlation coeffficients matrix and scatter plot",
      pch = 21, upper.panel=panel.pearson,lower.panel = panel.smooth)

# Plot separate
par(mfrow=c(1,1))
corrplot(corr,method="number", type="upper")
pairs(mydata[,-1], main = "Correlation coeffficients matrix and scatter plot",
      pch = 21, lower.panel = NULL, panel = panel.smooth)

# Check
library(car)
vif(fit)

# The VIF < 10 for all predictors, so there is no multicollinearity problem.

## Compute Leverage for measuring "unusualness" of x's
leverage =  hat(model.matrix(fit))
mydata$leverage = leverage

# Can also get the leverage using:
hatvalues(fit)

# Compute cutoff
p=6
n=dim(mydata)[1]
cutoff = 2*(p+1)/n
cutoff

# Find high leverage points
influential = mydata["leverage"]
influential = subset(influential,leverage> cutoff)
influential

# Using the rule of thumb (hii>2(p+1)/2), the observations
# 2,9,10,45 are are regarded as high leverage points

# Add observation number so can plot
influential$obs = as.numeric(rownames(influential))
mydata$obs = 1:n

# Plot influential points
ggplot(mydata,aes(x=obs, leverage)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
  geom_text(aes(35, .38, label="Influential Cutoff")) +
  geom_segment(aes(xend=obs, yend=0), color="blue") +
  geom_text(data =influential, aes(x=obs, y = leverage,
                                   label = obs), hjust = -1.5) +
  labs(title="Leverage ",
       x = "Obs Number",
       y = "Leverage") +
  geom_point(data=mydata[influential$obs,], colour="red", size=5)

# Alternatively without creating obs column: Plot influential points
```

```
ggplot(mydata,aes(x=as.numeric(rownames(mydata)), leverage)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
  geom_text(aes(35, .38, label="Influential Cutoff")) +
  geom_segment(aes(xend=as.numeric(rownames(mydata)), yend=0), color="blue") +
  geom_text(aes(x = as.numeric(rownames(mydata))[mydata$leverage>cutoff],
             y = mydata$leverage[mydata$leverage>cutoff],
             label = as.numeric(rownames(mydata))[mydata$leverage>cutoff]),hjust = -1.5)+
  labs(title="Leverage ",
       x = "Obs Number",
       y = "Leverage")

## Compute Cook distances for for measuring influence

# Cook's D plot
cutoff = 4/(dim(mydata)[1]);
plot(fit, which=4, cook.levels=cutoff, main = "Cook's D Plot");
# point 9, 29, 30 are influential points (using cutoff 4/n)

# Can aslo get influence using:
cooks.distance(fit)

# influence plot
library(car)   # needed for "influencePlot" function below
influencePlot(fit)
# The circles for each observation represent the relative size of the Cook's D
# point 9 is high leverage and influential, 30 is an outlier with high influence

# Can also get studeres, hat and cook D :
influence.measures(fit)

# Outliers/high leverage/influential points
summary(influence.measures(fit))

# Using the R function, potential problematic points are: 2,9,10,25,29,30

## Studentized residuals vs fitted
library(MASS)
stu_res = studres(fit)
mydata$stu_res = stu_res
mydata$fitted = fit$fitted

# Find outliers points
outlier = mydata["stu_res"]
outlier = subset(outlier,abs(stu_res)>2)
outlier
#rownames(mydata)[abs(stu_res) > 2]

# Using the rule that |studentized residuals| > 2, the observations
# 12,29,30 are are regarded as outliers

# Add observation number and fitted so can plot
outlier$fitted = fit$fitted[outlier$obs]
outlier$obs = as.numeric(rownames(outlier))

ggplot(mydata,aes(x=fitted, stu_res)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=2, linetype="dashed" , color = "red") +
  geom_hline(yintercept=-2, linetype="dashed" , color = "red") +
  geom_text(data =outlier, aes(x=fitted, y = stu_res,
                        label = obs), hjust = 1.5) +
  labs(title="Outliers Check ",
       x = "Fitted",
       y = "Studentized Residuals") +
  geom_point(data=mydata[outlier$obs,], colour="red", size=5)
```

```
# Remove potential outliers and influential points described above
# because the regression coefficients and interpretations might
# change due to the impact of these points.

#### Problem 2 ##############################################

# Import data
filename = "P128.txt"
mydata = read.table(filename,header = T)

#### Part a

# Assumptions about the predictors
# prefer cook distance since both x and y space

## check the fit (check linearity assumption by plotting residuals against each predictor)
fit = lm(Y~.,mydata)

plot_vector = vector(mode="list",length=6)

plot_vector[[1]] = ggplot(mydata,aes(x=mydata[[2]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[2]),y = "Residuals")

plot_vector[[2]]  = ggplot(mydata,aes(x=mydata[[3]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[3]),y = "Residuals")

plot_vector[[3]]  = ggplot(mydata,aes(x=mydata[[4]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[4]),y = "Residuals")

plot_vector[[4]]  = ggplot(mydata,aes(x=mydata[[5]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[5]),y = "Residuals")

plot_vector[[5]]  = ggplot(mydata,aes(x=mydata[[6]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[6]),y = "Residuals")

plot_vector[[6]] = ggplot(mydata,aes(x=mydata[[7]], y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = colnames(mydata[7]),y = "Residuals")

grid.arrange(plot_vector[[1]],
       plot_vector[[2]],
       plot_vector[[3]],
       plot_vector[[4]],
       plot_vector[[5]],
       plot_vector[[6]],
       ncol=2, main = "Residuals vs Predictor Variables")

# All plots look random so assumptions about the form of the model
# (linear in the regression parameters) is satisfied.

## Can also do the followig:
## check the fit (check linearity assumption by plotting partail regression/added variable plot)
library(car)
avPlots(fit)

## check normality (using qq plot)
qqPlot(fit, main = "Normal Q-Q Plot")

# Alternative code:
fit_stdres = rstandard(fit)
```

```
# To get studentized residuals
library(MASS)
stu_res = studres(fit)

qqnorm(fit_stdres,
      ylab = "Standardized Residuals",
      xlab = "Theoretical Quantiles",
      main = "Normal Q-Q Plot");
qqline(fit_stdres, col="red")

# formal test: Anderson-Darling test, Shapiro-Wilk test, Kolomogorov-Smirnov

# The plot shows that most points fall along the line, indicating the normality
# assumption of errors is satisfied

## check Checking Homoscedasticity (using residuals vs. fitted)

ggplot(mydata,aes(x=fit$fitted, y = fit$resid)) +
  geom_point(size = 3) +
  labs(x = "Fitted",y = "Residuals")

# The plot shows data points are random forming a parallel band, indicating the
# common variance assumption of errors is valid (Homoscedasticity)

## check independence (not time series data)
# the Durbin-Watson

## Check multicollinearity

corr = round(cor(mydata),2)
corr

# Plot combine correlation coefficients matrix and scatter plot
# http://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/iris_plots/
panel.pearson <- function(x, y, ...) {
  horizontal <- (par("usr")[1] + par("usr")[2]) / 2;
  vertical <- (par("usr")[3] + par("usr")[4]) / 2;
  text(horizontal, vertical, format(cor(x,y), digits=2))
}

pairs(mydata, main = "Correlation coeffficients matrix and scatter plot",
     pch = 21, upper.panel=panel.pearson,lower.panel = panel.smooth)

# Plot separate
par(mfrow=c(1,1))
corrplot(corr,method="number", type="upper")
pairs(mydata[,-1], main = "Correlation coeffficients matrix and scatter plot",
     pch = 21, lower.panel = NULL, panel = panel.smooth,cex.labels = 3)

# Check
library(car)
vif(fit)

# The scatter plots and correlation coefficients show strong correlation among the
# predictors. In addition, the VIF > 10 for X1,X2 and X3, so the assumption of
# linearly independence of each predictor is violated; there is a
# multicollinearity problem.

#### Part d

## Compute Leverage for measuring "unusualness" of x's
leverage =  hat(model.matrix(fit))
mydata$leverage = leverage

# Can also get the leverage using:
hatvalues(fit)
```

```
# Compute cutoff
p=6
n=dim(mydata)[1]
cutoff = 2*(p+1)/n
cutoff

# Find high leverage points
influential = mydata["leverage"]
influential = subset(influential,leverage> cutoff)
influential

# Using the rule of thumb (hii>2(p+1)/2), the observations
# 3 and 15 are are regarded as high leverage points

# Add observation number so can plot
influential$obs = as.numeric(rownames(influential))
mydata$obs = 1:n

# Plot influential points
ggplot(mydata,aes(x=obs, leverage)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
  geom_text(aes(35, .38, label="Influential Cutoff")) +
  geom_segment(aes(xend=obs, yend=0), color="blue") +
  geom_text(data =influential, aes(x=obs, y = leverage,
                          label = obs), hjust = -1.5) +
  labs(title="Leverage ",
      x = "Obs Number",
      y = "Leverage") +
  geom_point(data=mydata[influential$obs,], colour="red", size=5)

# Alternatively without creating obs column: Plot influential points
ggplot(mydata,aes(x=as.numeric(rownames(mydata)), leverage)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
  geom_text(aes(35, .38, label="Influential Cutoff")) +
  geom_segment(aes(xend=as.numeric(rownames(mydata)), yend=0), color="blue") +
  geom_text(aes(x = as.numeric(rownames(mydata))[mydata$leverage>cutoff],
             y = mydata$leverage[mydata$leverage>cutoff],
             label = as.numeric(rownames(mydata))[mydata$leverage>cutoff]),hjust = -1.5)+
  labs(title="Leverage ",
      x = "Obs Number",
      y = "Leverage")

## Compute Cook distances for for measuring influence

# Cook's D plot
cutoff = 4/(dim(mydata)[1]);
plot(fit, which=4, cook.levels=cutoff, main = "Cook's D Plot");
# point 17, 34 and 38 are influential points (using cutoff 4/n)

# Can aslo get influence using:
cooks.distance(fit)

# influence plot
library(car)   # needed for "influencePlot" function below
influencePlot(fit)
# The circles for each observation represent the relative size of the Cook's D
# point point 15 is high leverage, and 38 is an outlier with high influence

# Can also get studeres, hat and cook D :
influence.measures(fit)

# Outliers/high leverage/influential points
summary(influence.measures(fit))
```

```
# Using the R function, potential problematic points are: 3,5,7,8,15 and 38

## Studentized residuals vs fitted
library(MASS)
stu_res = studres(fit)
mydata$stu_res = stu_res
mydata$fitted = fit$fitted

# Find outliers points
outlier = mydata["stu_res"]
outlier = subset(outlier,abs(stu_res)>2)
outlier
#rownames(mydata)[abs(stu_res) > 2]

# Using the rule that |studentized residuals| > 2, the observations
# 34 and 38 are are regarded as outliers

# Add observation number and fitted so can plot
outlier$fitted = fit$fitted[outlier$obs]
outlier$obs = as.numeric(rownames(outlier))

ggplot(mydata,aes(x=fitted, stu_res)) +
  geom_point(size = 3, color="blue") +
  geom_hline(yintercept=2, linetype="dashed" , color = "red") +
  geom_hline(yintercept=-2, linetype="dashed" , color = "red") +
  geom_text(data =outlier, aes(x=fitted, y = stu_res,
                               label = obs), hjust = 1.5) +
  labs(title="Outliers Check ",
       x = "Fitted",
       y = "Studentized Residuals") +
  geom_point(data=mydata[outlier$obs,], colour="red", size=5)


#### Problem 3 ##############################################

# Import data
filename = "P128.txt"
mydata = read.table(filename,header = T)

# For addedplots can use the following:
library(car)
avPlots(fit)

## Part a
fit_y = lm(Y~ X1+X2+X3,mydata)
fit_x = lm(X4~ X1+X2+X3,mydata)
data = data.frame(x=fit_x$res,y=fit_y$res)

ggplot(data,aes(x,y)) +
  geom_point(size = 3, color="blue") +
  labs(title="Added-Variable Plot ",
       x = "X4|Others",
       y = "Y|Others")

fit_y = update(fit_y,.~.+X4)

# The partial regression plots shows a linear relationship, thus
# X4 makes a magringal contribution to y given other preidcors are already
# in the model. Conclusion: Add X4 to model

## Part b
fit_x = lm(X5~ X1+X2+X3+X4,mydata)
data = data.frame(x=fit_x$res,y=fit_y$res)

ggplot(data,aes(x,y)) +
```

```
  geom_point(size = 3, color="blue") +
  labs(title="Added-Variable Plot ",
       x = "X5|Others",
       y = "Y|Others")

# The partial regression plots looks random, thus
# X5 makes no magringal contribution to y given other preidcors are already
# in the model. Conclusion: Do not add X5

## Part c
fit_x = lm(X6~ X1+X2+X3+X4,mydata)
data = data.frame(x=fit_x$res,y=fit_y$res)

ggplot(data,aes(x,y)) +
  geom_point(size = 3, color="blue") +
  labs(title="Added-Variable Plot ",
       x = "X6|Others",
       y = "Y|Others")

# The partial regression plots looks random, thus
# X6 makes no magringal contribution to y given other preidcors are already
# in the model. Conclusion: Do not add X6.

## Part d

# Look at correlation
corr = round(cor(mydata),2)
corr
corrplot(corr,method="number", type="upper")
pairs(mydata, main = "Correlation coeffficients matrix and scatter plot",
      pch = 21, lower.panel = NULL, panel = panel.smooth, cex.labels = 3)

# Checking VIF
fitC = lm(Y~X1 + X2 + X3 + X4 + X6,mydata)
vif(fitC)

# The scatter plots and correlation coefficients shows that
# X6 is strongly correlated with X1, X2 and X3.
# In addition, the VIF > 10 for X1,X2 and X3, so the assumption of
# linearly independence of each predictor is violated; there is a
# multicollinearity problem.

fitA = lm(Y~X1+X2+X3+X4,mydata)
fitD = lm(Y~X4+X6,mydata)
summary(fitA)
summary(fitD)

# Thus, either use X1, X2 and X3 or X6 in the model
# Both models (X1+X2+X3+X4 and X4+X6) show similar R-squared that
# are very high, so either model can be used as the best possible
# description of Y. The model X4+X6 might be preferred because
# it is the smaller model and thus simpler.

#### Problem 4 ################################################

# Import data
filename = "Used+car+prices+%28Training+set%29.csv"
mydata = read.csv(filename,header = T)

# Look at data
names(mydata)
head(mydata)
nrow(mydata)
summary(mydata)

# Log price
```

```
colnames(mydata)[13] = "log_Price"

# Fit log model

fit = lm(log_Price ~ Mileage + Liter + Make + Type, mydata)
summary(fit)

# Fit non-log model
fit2 = lm(Price ~ Mileage + Liter + Make + Type, mydata)

# 4 by 4 grid
par(mfrow=c(2,2))

# Residual vs fitted log model
plot(fit$fitted,fit$resid,
    ylab = "Residuals",
    xlab = "Fitted",
    main = "Residual Plot (log model)")
# Normal plot log model
fit_stdres = rstandard(fit)
qqnorm(fit_stdres,
      ylab = "Standardized Residuals",
      xlab = "Theoretical Quantiles",
      main = "Normal Q-Q Plot (log model)");
qqline(fit_stdres, col="red")

# Residual vs fitted non-log model
plot(fit2$fitted,fit2$resid,
    ylab = "Residuals",
    xlab = "Fitted",
    main = "Residual Plot (non-log model)")
# Normal plot non- log model
fit2_stdres = rstandard(fit2)
qqnorm(fit2_stdres,
      ylab = "Standardized Residuals",
      xlab = "Theoretical Quantiles",
      main = "Normal Q-Q Plot (non-log model)")
qqline(fit2_stdres, col="red")


# The plots for the log model look more satisfactory.
# The residuals plot shows that points are scattered randomly
# about zero.
# Overall the standardized residuals seem to fit a straight
# line with the normal scores.

# On the other hand, the residual plot for the non-log model
# shows a clear fanning pattern where the residuals increase
# as the size of the fitted value increase.
# The Q-Q plot also shows large deivations from the straight
# line in the tails (distribution looks like long tails in upper)

# A log transformation is clearly benefitical to make the
# assumptions of linear regression hold because it shrinks
# the distribution (e.g. shrinks values of Prices in such a
# way that large values of Prices are affected much more than
# small values are) as can be seen from the residuals and
# Q-Q plots after applying the transformation. More specifically,
# the log transformation is useful in stabilizing the variance
# because it shirnks the upper tail of the data and help make the
# variance constant, satisfying homoscedasticity, which in turns
# helps improve normality of the response variable (price).

par(mfrow=c(2,2))
hist(mydata$log_Price, main="Histogram (log model)", xlab = "log(Price)")
hist(fit_stdres, main="Histogram (log model)", xlab = "Std Residuals")
```

```
hist(mydata$Price, main="Histogram (non-log model)", xlab = "Price")
hist(fit2_stdres, main="Histogram (non-log model)", xlab = "Std Residuals)")


#### Problem 5 ###############################################

# Import data
filename = "wood_beams.csv"
mydata = read.csv(filename,header = T)


ggplot(mydata,aes(x=Specific_Gravity, y = Moisture_Content)) +
  geom_point(size = 3, color="blue") +
  geom_text(data = mydata, aes(x=Specific_Gravity, y = Moisture_Content,
                           label = Beam_Number), hjust = -1.5) +
  geom_point(data=mydata[4, ], colour="red", size=5)


### Part a

# It appears that beam number 4 to be an outlier in terms of specific gravity
# and moisture content because its value does not fall into the general pattern
# of association between the variables. Beam 4 seems to be very in low in both
# specific gravity and moisture content compared to the other beams.
# More precisely, Beam 4 can be described as a bivariate outlier, that is an outlier
# that occurs within the joint combination of two (bivariate) variables.

### Part b

# Fit Regression
fit = lm(Strength ~ Specific_Gravity + Moisture_Content, mydata)

# Compute Leverage
leverage =  hat(model.matrix(fit))
mydata$leverage = leverage

# Compute cutoff for influential
p=2
n=dim(mydata)[1]
cutoff = 2*(p+1)/n
cutoff

# Find influential points
names(leverage)=mydata$Beam_Number
leverage
mydata$Beam_Number[leverage > cutoff]

# Plot influential points
ggplot(mydata,aes(x=factor(Beam_Number), leverage)) +
  geom_point(size = 3, color="blue") +
  labs(title="Leverage vs Beams",
      x = "Beam Number",
      y = "Leverage") +
  geom_hline(yintercept=cutoff, linetype="dashed" , color = "red") +
  geom_text(aes(9, .58, label="Influential Cutoff")) +
  geom_segment(aes(xend=Beam_Number, yend=0), color="blue")

# Yes, beam 4 identified as an outlier is an influential observation
# using the rule h_ii = 0.604 > 2*(2+1)/10 = 0.6

### Part c

# Fit regression without influential observation
mydata2 = mydata
mydata2 = mydata2[!(leverage > cutoff),]
```

```
fit2 = lm(Strength ~ Specific_Gravity + Moisture_Content, mydata2)

# Compare regressions from all data and without influential observation
summary(fit)
summary(fit2)

# from stackoverflow
percent <- function(x, digits = 2, format = "f", ...) {
  paste0(formatC(100 * x, format = format, digits = digits, ...), "%")
}

# percent change
p = percent((fit2$coeff-fit$coeff)/fit$coeff)
names(p)=names(fit$coeff)
p

# plots of regressions
par(mfrow=c(2,2))
plot(fit)
par(mfrow=c(2,2))
plot(fit2)

# Looking at the coefficients, it seems that the fit changed.
# The coefficeint for Specific_Gravity changed from 8.4947 to 6.799 (-20%)
# and the coefficient for Moisture_Content changed from -0.2663 to -0.3905 (+47%).
# The intercept also changed from 10.3015 to 12.4107 (+20%)
# The significance of the coefficient at 0.05 level did not change, and
# the R-squared increased slightly from 0.9 to 0.91.
# The diagnostics plots look very similar for both models

# Because the influential point (beam 4) does not seem to follow the relationship
# in terms of specific gravity and moisture content of the other beams, and also
# has a big impact on the effect of the predictors variables and thus the prediction
# of the strength, then the fitted equation after removing the influential
# point should be used to predict the wood beam strength since the prediction
# would not be heavily influenced by one data point.
```