Final Exam

Aaron Rockwell

12/16/2019

```r
#install.packages("randomForest")
library(MASS)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'

## The following object is masked from 'package:MASS':
##
##     cement

## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
library(leaps)
library(faraway)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car

##
## Attaching package: 'faraway'

## The following object is masked from 'package:olsrr':
##
##     hsb
```

```r
library(rpart)
```

```
##
## Attaching package: 'rpart'

## The following object is masked from 'package:faraway':
##
##     solder
```

```r
library(rpart.plot)
library(neuralnet)
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```r
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-1
```

```r
library(lars)
```

```
## Loaded lars 1.2
```

```r
library(C50)
library(graphics)
library(gmodels)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

Problem 1:

1.)Use the PR1_Dataset data which contains 5 continuous variables (no categorical variables), the answer the questions below: (25 pts)

a-) Fit a regression model to predict Y by using all variables. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (5pts)

```
PR1.df = data.frame(read.csv("PR1_Dataset.csv"))

PR1.reg = lm(Y~.,data=PR1.df)

PR1.reg
```

```
##
## Call:
## lm(formula = Y ~ ., data = PR1.df)
##
## Coefficients:
## (Intercept)           X1           X2           X3           X4
##    155.0304       0.3911       0.8639       0.3616      -0.8467
##          X5
##      0.1923
```

```
print("VIF:")
```

```
## [1] "VIF:"
```

```
vif(PR1.reg)
```

```
##       X1       X2       X3       X4       X5
## 3.916370 1.803353 2.812730 6.278713 1.624470
```

```
#anova(PR1.reg)
#nrow(PR1.df)

drst = rstudent(PR1.reg)
tb = qt(1-0.05/(2*40),40-6-1)

sum(abs(drst)>abs(tb))
```

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

```
drst
```

```
##           1           2           3           4           5           6
## -0.05383143  0.39576079  1.41369470  0.10185031 -0.57251916  0.03123661
##           7           8           9          10          11          12
##  0.64423184  1.41162004  1.46926590  0.47880359 -2.82764460  1.67827617
##          13          14          15          16          17          18
## -0.80964760 -0.84519108 -0.14693819  0.36486841 -0.55548943 -1.05266469
##          19          20          21          22          23          24
##  0.50573425  0.42519964  0.31323174  0.31749746  0.09492965 -0.64343598
##          25          26          27          28          29          30
##  0.44061054 -0.35207167  0.82050055 -0.35814985  0.52896891 -0.46973442
##          31          32          33          34          35          36
##  0.83520808  0.05949628 -0.61605115 -0.70881639  2.24546660 -5.21022269
##          37          38          39          40
```

```
##  0.90458819 -0.90998907  0.38887233 -0.91078750
```

```r
tb
```

```
## [1] 3.529649
```

```r
#plot(drst)

hii <- hatvalues(PR1.reg)
#hii
summary(hii)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03793 0.07857 0.11909 0.15000 0.16898 0.69026
```

```r
sum(hii>(2*6/40))
```

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

```r
(hii>(2*6/40))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##    13    14    15    16    17    18    19    20    21    22    23    24
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    25    26    27    28    29    30    31    32    33    34    35    36
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
##    37    38    39    40
## FALSE FALSE FALSE FALSE
```

Problem 1 a:

Regression model:

$$\hat{Y} = 0.3911X_1 + 0.8639X_2 + 0.3616X_3 - 0.8467X_4 + 0.1923X_5 + 155.0304$$

Using a threshold of 10 for Variance Inflation Factor, there is not significant multicolinearity in the model, with X4 at 6.278713 as highest VIF value.

The errors are normally distributed with the exception of one outlier at the 36th case of the data (-5.21022269)

There are three cases that are influencial and could be investigated further (8, 35, and 36)

b-) Use the stepwise variable selection procedure to find the best model. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (5pts)

```r
f5=lm(Y~X1+X2+X3+X4+X5, data=PR1.df)

#ols_step_both_p(f5,prem=0.05,details=TRUE)
ols_step_both_p(f5,prem=0.05,details=FALSE)
```

```
## Stepwise Selection Method
## ---------------------------
##
## Candidate Terms:
##
## 1. X1
## 2. X2
```

```
## 3. X3
## 4. X4
## 5. X5
##
## We are selecting variables based on p value...
##
## Variables Entered/Removed:
##
## - X4 added
## - X2 added
##
## No more variables to be added/removed.
##
##
## Final Model Output
## ------------------
##
##                         Model Summary
## --------------------------------------------------------------
## R                       0.918       RMSE               5.381
## R-Squared               0.842       Coef. Var          2.062
## Adj. R-Squared          0.834       MSE               28.955
## Pred R-Squared          0.795       MAE                3.693
## --------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##
##                                ANOVA
## --------------------------------------------------------------------
##                 Sum of
##                Squares        DF    Mean Square      F         Sig.
## --------------------------------------------------------------------
## Regression    5718.560         2       2859.280    98.749    0.0000
## Residual      1071.340        37         28.955
## Total         6789.900        39
## --------------------------------------------------------------------
##
##                                     Parameter Estimates
## ----------------------------------------------------------------------------------------------
##       model        Beta    Std. Error    Std. Beta      t        Sig        lower      upper
## ----------------------------------------------------------------------------------------------
## (Intercept)    222.590        15.298                 14.550    0.000    191.593    253.586
##          X4     -1.465         0.179       -0.666    -8.205    0.000     -1.827     -1.103
##          X2      0.732         0.170        0.350     4.311    0.000      0.388      1.076
## ----------------------------------------------------------------------------------------------

##
##                              Stepwise Selection Summary
## ----------------------------------------------------------------------------------------
##                      Added/                    Adj.
## Step    Variable    Removed      R-Square    R-Square     C(p)        AIC        RMSE
## ----------------------------------------------------------------------------------------
##    1       X4       addition       0.763       0.757    21.9930    267.3052    6.5079
```

```
## 2        X2        addition        0.842        0.834        4.6040        253.0265        5.3810
## ----------------------------------------------------------------------------------------------------
```

```r
PR1.bestReg = lm(Y~X2+X4, data=PR1.df)

PR1.bestReg
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.df)
##
## Coefficients:
## (Intercept)           X2           X4
##    222.5896       0.7323      -1.4652
```

```r
vif(PR1.bestReg)
```

```
##       X2       X4
## 1.544187 1.544187
```

```r
drst = rstudent(PR1.bestReg)
tb = qt(1-0.05/(2*40),40-6-1)

sum(abs(drst)>abs(tb))
```

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

```r
drst
```

```
##           1           2           3           4           5           6
## -0.24888047  0.10415686  1.21541011 -0.05566559 -0.88057529  0.37979653
##           7           8           9          10          11          12
##  0.76468574  0.25063966  1.31602192  0.32467433 -1.98824209  2.43819880
##          13          14          15          16          17          18
## -0.34911703 -0.39524065 -0.18325019  0.51258751 -0.59737944 -0.83338729
##          19          20          21          22          23          24
##  0.86136070  0.07735177 -0.14679598  0.33696864  0.40969392 -0.72229100
##          25          26          27          28          29          30
##  0.74227224 -0.63375120  1.03415357  0.20171906  0.24995594 -0.74856357
##          31          32          33          34          35          36
##  0.71392373  0.29151873 -0.93288709 -0.67210467  1.22331592 -5.70495773
##          37          38          39          40
##  0.90801931 -0.32158711  0.02975696 -0.62924405
```

```r
tb
```

```
## [1] 3.529649
```

```r
#plot(drst)

hii <- hatvalues(PR1.bestReg)
summary(hii)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02755 0.03559 0.05732 0.07500 0.07594 0.33792
```

```r
sum(hii>(2*3/40))
```

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

```r
(hii>(2*3/40))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##    13    14    15    16    17    18    19    20    21    22    23    24
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    25    26    27    28    29    30    31    32    33    34    35    36
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##    37    38    39    40
## FALSE FALSE FALSE FALSE
```

Problem 1 b:

Best model $\hat{Y} = 0.7323X_2 - 1.4652X_4 + 222.5896$

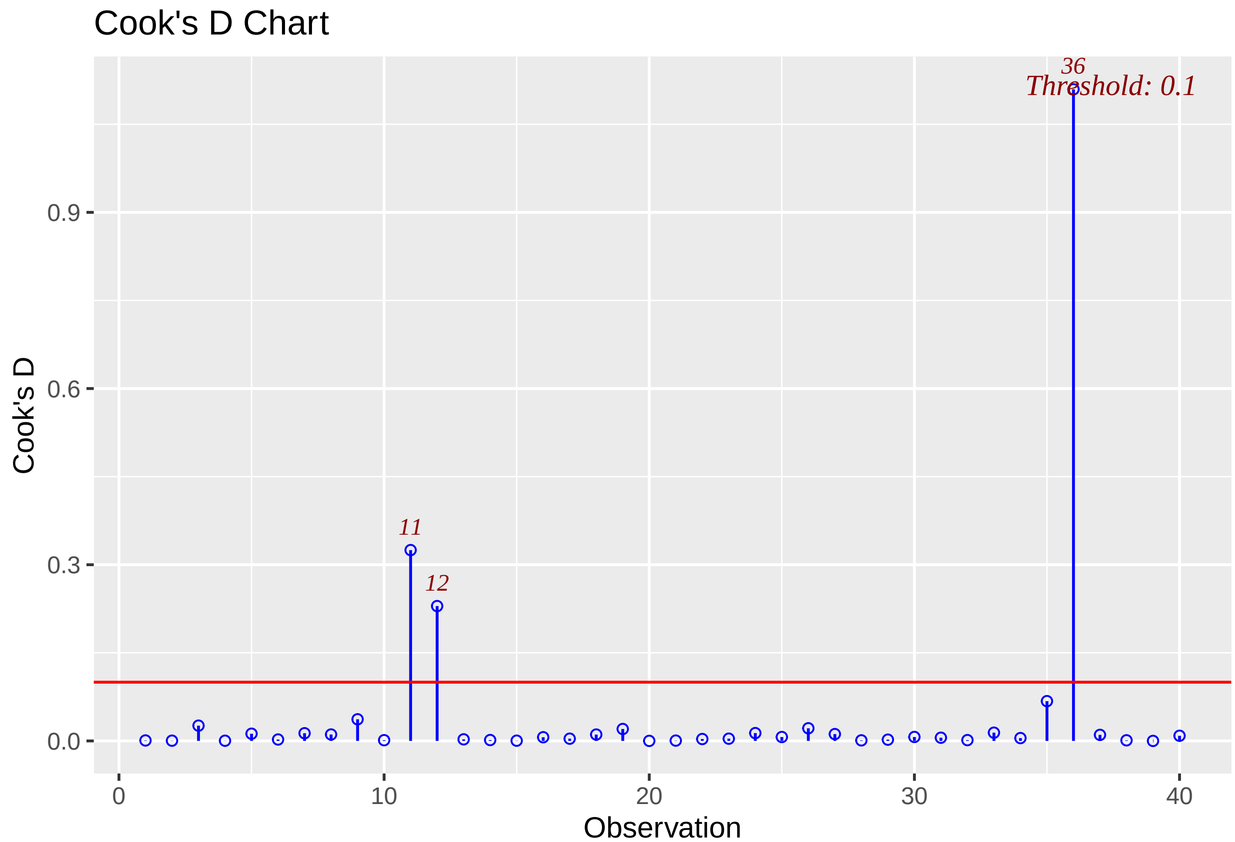
No multicolinearity present, VIF = 1.544187 (less than 10).

Case 36 is still an outlier at -5.70495773 (outside 3.529649), but the rest of the data is normally distributed.

There are 4 influencial cases in the dataset (4,8,11,36)

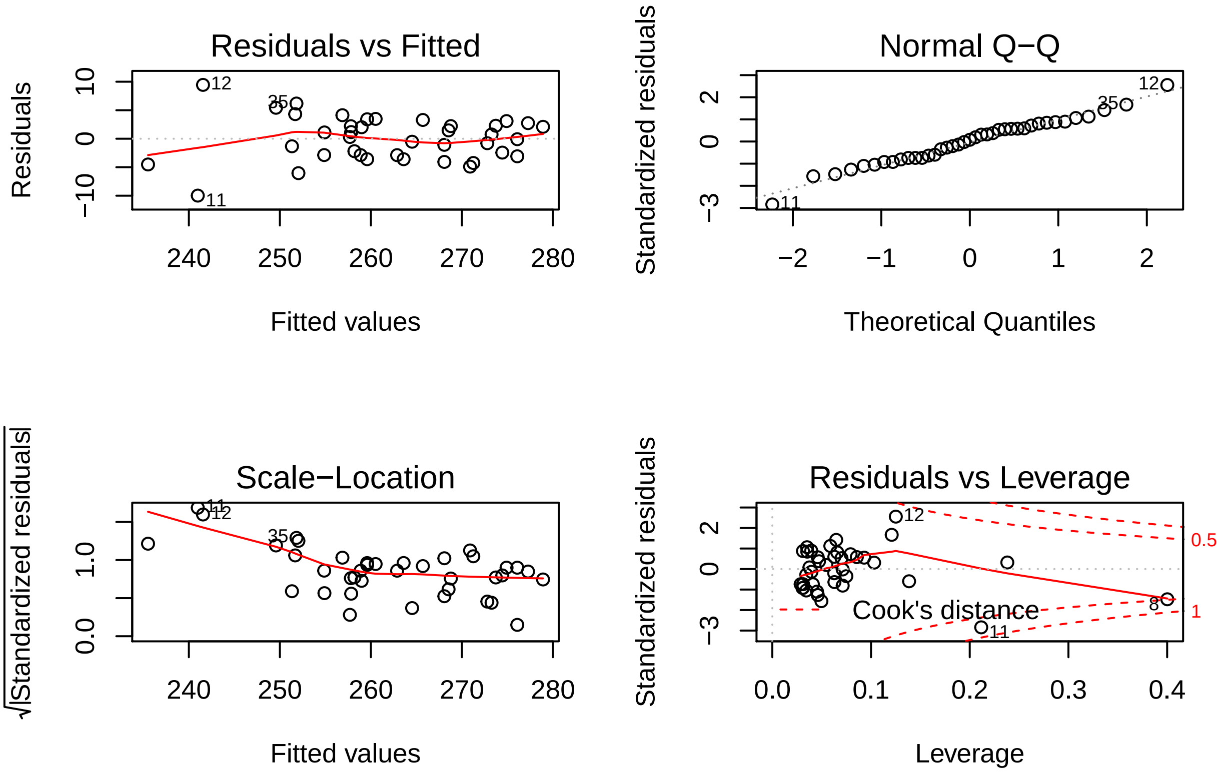
c-) Use the model built in part b, exclude the observation with the largest cook distance and refit the model and comment the model results (5pts)

```r
par(mfrow=c(2,2))
ols_plot_cooksd_chart(PR1.bestReg)
```




```r
PR1.noCook.df = PR1.df[-c(36),]
```

```r
PR1.noCook.reg = lm(Y~X2+X4, data=PR1.noCook.df)

plot(PR1.noCook.reg)
```




```r
PR1.noCook.reg
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.noCook.df)
##
## Coefficients:
## (Intercept)           X2           X4
##    211.8978       0.8233      -1.1804
```

```r
summary(PR1.noCook.reg)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.noCook.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9810 -2.8786  0.3054  2.5058  9.4437
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 211.8978    11.3946  18.596  < 2e-16 ***
## X2            0.8233     0.1258   6.544 1.31e-07 ***
## X4           -1.1804     0.1404  -8.408 5.15e-10 ***
## ---
```

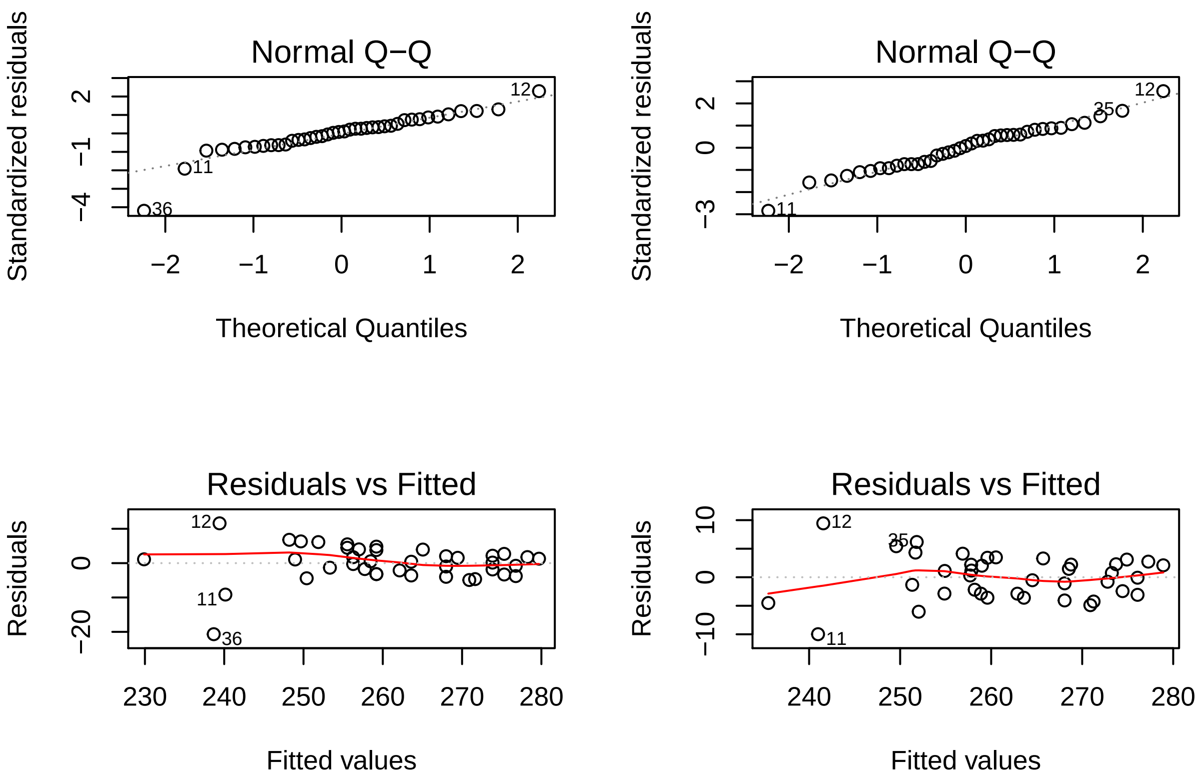
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.953 on 36 degrees of freedom
## Multiple R-squared:  0.8851, Adjusted R-squared:  0.8787
## F-statistic: 138.7 on 2 and 36 DF,  p-value: < 2.2e-16
```

Problem 1 c:

Without case 36, the QQ plot has a well-fit line, and the R^2 for the model is 0.8851 (compared to 0.8422).

d-) Use the model built in part b, fit the robust regression and compared it against the model in part c, comments on the model results. (5pts)

```
par(mfrow=c(2,2))
plot(PR1.bestReg, which = 2)
plot(PR1.noCook.reg, which=2)

plot(PR1.bestReg, which = 1)
plot(PR1.noCook.reg, which=1)
```




```
PR1.bestReg
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.df)
##
## Coefficients:
## (Intercept)           X2           X4
##    222.5896       0.7323      -1.4652
```

```r
PR1.noCook.reg
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.noCook.df)
##
## Coefficients:
## (Intercept)           X2           X4
##    211.8978       0.8233      -1.1804
```

```r
summary(PR1.bestReg)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.6799  -3.1931   0.4761   2.9719  11.5850
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 222.5896    15.2981  14.550  < 2e-16 ***
## X2            0.7323     0.1699   4.311 0.000116 ***
## X4           -1.4652     0.1786  -8.205 7.52e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.381 on 37 degrees of freedom
## Multiple R-squared:  0.8422, Adjusted R-squared:  0.8337
## F-statistic: 98.75 on 2 and 37 DF,  p-value: 1.459e-15
```

```r
summary(PR1.noCook.reg)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4, data = PR1.noCook.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9810 -2.8786  0.3054  2.5058  9.4437
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 211.8978    11.3946  18.596  < 2e-16 ***
## X2            0.8233     0.1258   6.544 1.31e-07 ***
## X4           -1.1804     0.1404  -8.408 5.15e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.953 on 36 degrees of freedom
## Multiple R-squared:  0.8851, Adjusted R-squared:  0.8787
## F-statistic: 138.7 on 2 and 36 DF,  p-value: < 2.2e-16
```

Problem 1 d:

Without case 36 (largest Cooks distance), the QQ plot for the model has a better line, and the residuals vs fitted values is a better fit.

The model without case 36, puts more weight on X2 and less on X3.

Also, the R^2 improved without case 36 from 0.8422 to .8851.

e-) Use the model built in part b, predict Y for X1=75, X2=78, X3=34, X4=18, X5=18 and calculate 95% confidence interval (5pts).

```r
predict.P1 = data.frame(cbind(X1=75, X2=78, X3=34, X4=18, X5=18))

predict(PR1.bestReg, predict.P1, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 253.3316 251.2508 255.4124
```

Problem 1 e:

Using the model from question b (X2 and X4), 253.3316 would be the predicted value with the confidence interval of 95%, the range would be 251.2508 to 255.4124.

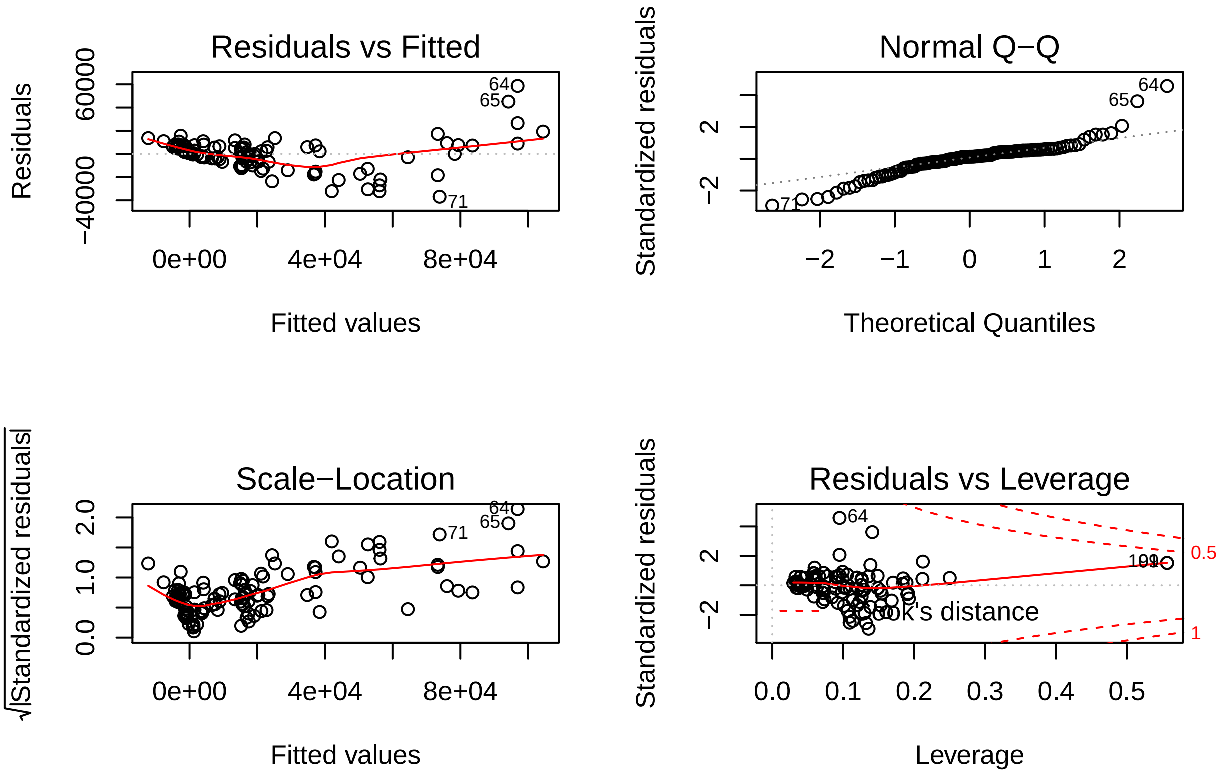
Problem 2:

2.) Use the PR2_Dataset data: X4, X5, X6, and X7 are the categorical variables, Y and remaining independent variables are continuous variables. X4 has two levels, X5 has 4, X6 has 5, and X7 has 3 levels (create dummy variables for the categorical variables). Answer the questions below: (30 pts)

a-) Fit a regression model to predict Y by using all variables. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? (10 pts)

```r
PR2.df = data.frame(read.csv("PR2_Dataset.csv"))

#PR2.df

#X4 = 1, 2
#X5 = 1, 2, 3, 4
#X6 = 1, 2, 3, 4, 5
#X7 = 1, 2, 3

Y =  PR2.df$Y
X1 = PR2.df$X1
X2 = PR2.df$X2
X3 = PR2.df$X3

X4 = as.numeric(PR2.df$X4 == 1)

X5a = as.numeric(PR2.df$X5 == 1)
X5b = as.numeric(PR2.df$X5 == 2)
X5c = as.numeric(PR2.df$X5 == 3)

X6a = as.numeric(PR2.df$X6 == 1)
X6b = as.numeric(PR2.df$X6 == 2)
X6c = as.numeric(PR2.df$X6 == 3)
X6d = as.numeric(PR2.df$X6 == 4)

X7a = as.numeric(PR2.df$X7 == 1)
X7b = as.numeric(PR2.df$X7 == 2)

PR2.refit.df = data.frame(cbind(Y,X1,X2,X3,X4,X5a,X5b,X5c,X6a,X6b,X6c,X6d,X7a,X7b))

PR2.refit.reg = lm(Y~.,data=PR2.refit.df)

par(mfrow=c(2,2))
plot(PR2.refit.reg)
```

```
## Warning: not plotting observations with leverage one:
##   63, 79

## Warning: not plotting observations with leverage one:
##   63, 79
```




```r
print("VIF:")
```

```
## [1] "VIF:"
```

```r
vif(PR2.refit.reg)
```

```
##         X1         X2         X3         X4        X5a        X5b        X5c
##   1.767314   2.750584   1.473327  24.549139  13.790226  17.189914  20.199396
##        X6a        X6b        X6c        X6d        X7a        X7b
##   5.748744   1.461505   1.099084   1.121915  20.600710   2.986026
```

```r
drst = rstudent(PR2.refit.reg)
tb = qt(1-0.05/(2*40),40-6-1)

sum(abs(drst)>abs(tb))
```

```
## [1] NA
```

```r
#drst
#tb

#plot(drst)

hii <- hatvalues(PR2.refit.reg)
#hii
summary(hii)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02966 0.05804 0.09220 0.11570 0.12644 1.00000
```

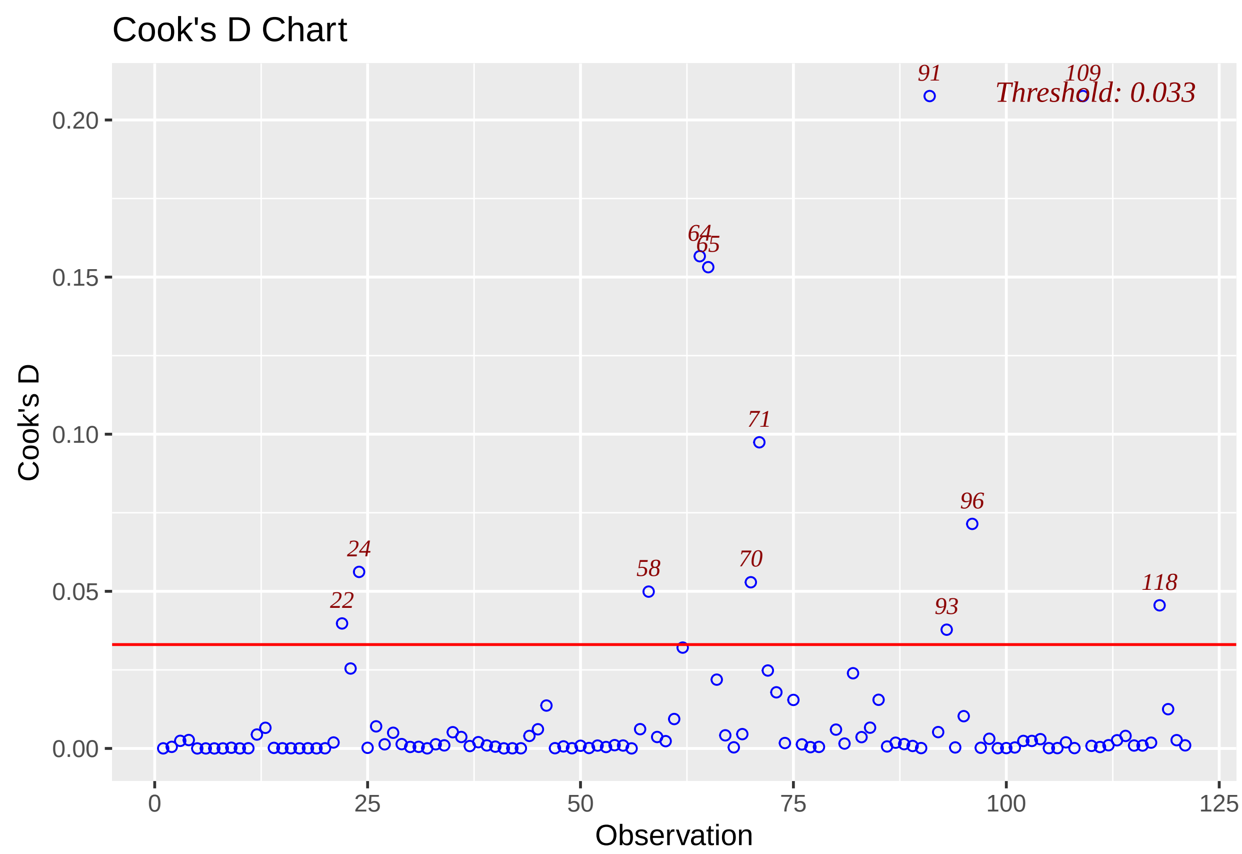
```r
sum(hii>(2*6/40))
```

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

```r
(hii>(2*6/40))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    13    14    15    16    17    18    19    20    21    22    23    24
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    25    26    27    28    29    30    31    32    33    34    35    36
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    37    38    39    40    41    42    43    44    45    46    47    48
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    49    50    51    52    53    54    55    56    57    58    59    60
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    61    62    63    64    65    66    67    68    69    70    71    72
## FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    73    74    75    76    77    78    79    80    81    82    83    84
## FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##    85    86    87    88    89    90    91    92    93    94    95    96
## FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##    97    98    99   100   101   102   103   104   105   106   107   108
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   109   110   111   112   113   114   115   116   117   118   119   120
##  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   121
## FALSE
```

```r
#print("The regression model")

#PR2.refit.reg

ols_plot_cooksd_chart(PR2.refit.reg)
```



Problem 2 a:

The regression model is:

```
PR2.refit.reg
```

```
##
## Call:
## lm(formula = Y ~ ., data = PR2.refit.df)
##
## Coefficients:
## (Intercept)           X1           X2           X3           X4
##  -2.837e+04    2.771e-02    9.661e+03    1.282e+02    2.771e+04
##         X5a          X5b          X5c          X6a          X6b
##  -3.536e+04   -6.664e+03    1.111e+04   -2.215e+03   -2.660e+03
##         X6c          X6d          X7a          X7b
##  -1.800e+03    5.194e+03    1.093e+04   -2.720e+03
```

There is multicolinearity of cases with over 10 VIF, they are: X4, X5a, X5b, X5c, X6a, and X7a.

The error residual vs fitted distribution looks exponential and might need a transformation.

The most significant Cook distance cases are 109 and 91.

The influencial cases are 63, 79, 91, 109.

b-) Conduct the Breusch-Pagan for testing unequal variances and document your results (5pts)

```r
ei<-PR2.refit.reg$residuals

ei2<-ei^2
g<-lm(ei2~X1+X2+X3+X4+X5a+X5b+X5c+X6a+X6b+X6c+X6d+X7a+X7b)
summary(g)
```

```
##
## Call:
## lm(formula = ei2 ~ X1 + X2 + X3 + X4 + X5a + X5b + X5c + X6a +
##     X6b + X6c + X6d + X7a + X7b)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -795671167  -67804865  -16663653   58261496 2563545746
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.121e+08  2.180e+08  -1.432   0.1550
## X1           2.422e+02  1.297e+02   1.868   0.0645 .
## X2           8.842e+07  4.100e+07   2.157   0.0333 *
## X3           1.341e+06  3.382e+06   0.396   0.6926
## X4          -4.062e+08  3.808e+08  -1.067   0.2886
## X5a          3.900e+08  4.783e+08   0.815   0.4167
## X5b          7.909e+07  2.661e+08   0.297   0.7669
## X5c          8.957e+08  4.042e+08   2.216   0.0288 *
## X6a         -1.085e+08  1.740e+08  -0.623   0.5344
## X6b         -5.785e+07  1.042e+08  -0.555   0.5798
## X6c         -2.038e+08  3.706e+08  -0.550   0.5836
## X6d         -4.702e+07  1.452e+08  -0.324   0.7467
## X7a          4.008e+07  2.910e+08   0.138   0.8907
## X7b          2.938e+07  1.183e+08   0.248   0.8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.52e+08 on 107 degrees of freedom
## Multiple R-squared:   0.36,  Adjusted R-squared:  0.2822
## F-statistic:  4.63 on 13 and 107 DF,  p-value: 2.879e-06
```

```r
anova(g)["Sum Sq"]
```

```
##               Sum Sq
## X1        3.0467e+18
## X2        2.0680e+18
## X3        1.2670e+15
## X4        4.2349e+17
## X5a       1.1520e+18
## X5b       5.9346e+16
## X5c       5.9708e+17
## X6a       2.5647e+16
## X6b       2.9585e+16
## X6c       3.5086e+16
## X6d       1.2538e+16
## X7a       2.2316e+14
## X7b       7.6420e+15
```

```
## Residuals 1.3260e+19
```

```r
anova(PR2.refit.reg)
```

```
## Analysis of Variance Table
##
## Response: Y
##            Df     Sum Sq    Mean Sq  F value    Pr(>F)
## X1          1 4.2241e+10 4.2241e+10 232.8643 < 2.2e-16 ***
## X2          1 3.3966e+10 3.3966e+10 187.2466 < 2.2e-16 ***
## X3          1 8.3760e+03 8.3760e+03   0.0000    0.9946
## X4          1 6.3802e+09 6.3802e+09  35.1724 3.755e-08 ***
## X5a         1 6.5717e+09 6.5717e+09  36.2282 2.501e-08 ***
## X5b         1 4.0175e+08 4.0175e+08   2.2148    0.1396
## X5c         1 5.5791e+07 5.5791e+07   0.3076    0.5803
## X6a         1 2.3273e+06 2.3273e+06   0.0128    0.9100
## X6b         1 1.1569e+08 1.1569e+08   0.6378    0.4263
## X6c         1 6.8443e+06 6.8443e+06   0.0377    0.8464
## X6d         1 1.7796e+08 1.7796e+08   0.9810    0.3242
## X7a         1 3.2105e+08 3.2105e+08   1.7698    0.1862
## X7b         1 6.5469e+07 6.5469e+07   0.3609    0.5493
## Residuals 107 1.9410e+10 1.8140e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
nrow(PR2.refit.df)
```

```
## [1] 121
```

```r
SSR<-sum(anova(g)["Sum Sq"])-13260001241805215744
SSE<- 19409611507
chi.test<-(SSR/13)/((SSE/121)^2)
chi.test
```

```
## [1] 22.29725
```

```r
1-pchisq(chi.test,2)
```

```
## [1] 1.439508e-05
```

Problem 2 b:

Ho: Gamma is 0 Ha: Gamma is NOT 0

Gamma is almost zero, accept null, the error variance is constant.

c) Use weight least squares regression (perform only one iteration) document your results. (5 pts)

```r
abs.ei<-abs(PR2.refit.reg$residuals)
PR2.refit.rege<-lm(abs.ei~X1+X2+X3+X4+X5a+X5b+X5c+X6a+X6b+X6c+X6d+X7a+X7b)
si<-PR2.refit.rege$fitted.values
wi<-1/(si^2)

PR2.refit.regf<-lm(Y~X1+X2+X3+X4+X5a+X5b+X5c+X6a+X6b+X6c+X6d+X7a+X7b,weights=wi)
summary(PR2.refit.regf)
```

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

```
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5a + X5b + X5c + X6a +
##     X6b + X6c + X6d + X7a + X7b, weights = wi)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8869 -1.1530 -0.0745  0.3800  4.0710
##
## Coefficients: (5 not defined because of singularities)
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.999e+03  1.277e-11  3.131e+14   <2e-16 ***
## X1           7.897e-02  5.666e-17  1.394e+15   <2e-16 ***
## X2                  NA         NA         NA       NA
## X3                  NA         NA         NA       NA
## X4                  NA         NA         NA       NA
## X5a         -9.302e+03  1.203e+04 -7.730e-01   0.4409
## X5b         -1.961e+04  1.152e+04 -1.702e+00   0.0915 .
## X5c          1.718e+04  1.348e+04  1.274e+00   0.2053
## X6a          3.923e+03  6.819e+03  5.750e-01   0.5663
## X6b          8.663e+02  3.274e+03  2.650e-01   0.7918
## X6c                 NA         NA         NA       NA
## X6d          1.385e+03  3.211e+03  4.310e-01   0.6671
## X7a          1.388e+04  1.162e+04  1.195e+00   0.2346
## X7b                 NA         NA         NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.295 on 112 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.428e+29 on 8 and 112 DF,  p-value: < 2.2e-16
```

```r
rbind(coef(PR2.refit.regf),coef(PR2.refit.reg))
```

```
##      (Intercept)         X1       X2       X3      X4        X5a
## [1,]    3998.531 0.07897132       NA       NA      NA  -9302.263
## [2,]  -28368.194 0.02770604 9660.987 128.1552 27709.9 -35364.637
##             X5b      X5c       X6a        X6b       X6c      X6d      X7a
## [1,] -19613.304 17180.41  3922.736   866.3499        NA 1384.740 13880.96
## [2,]  -6663.521 11113.96 -2215.349 -2659.8851 -1799.712 5193.991 10927.18
##            X7b
## [1,]        NA
## [2,] -2719.675
```

Problem 2 c:

After a round of wieghted regression, 5 coefficient were not defined because of singularities, thus changing the other coefficients significantly.

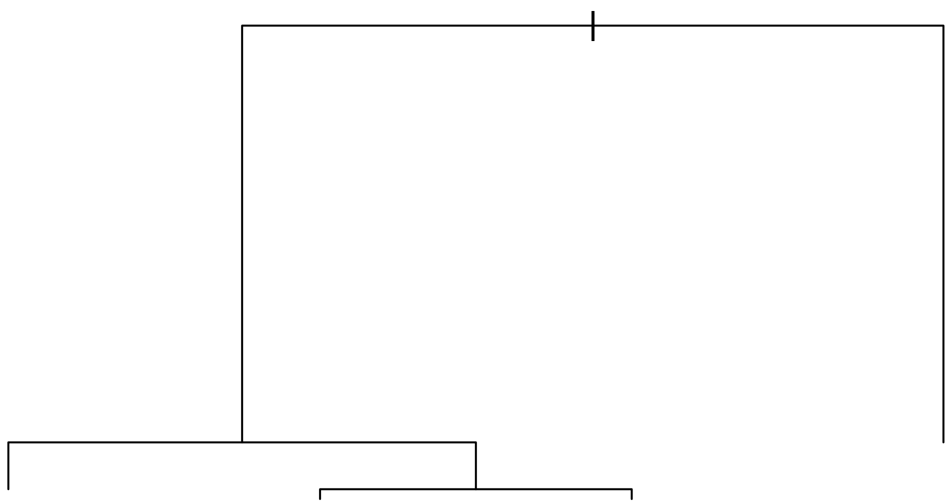
d-) Compare your model in part a against the regression tree and Neural Network Model, and calculate the SSE for each model, which method has the lowest SSE? And explain which model you will choose. (10 pts)

```r
f.q4.bestsubset<-PR2.refit.reg
an=anova(f.q4.bestsubset)
anova(f.q4.bestsubset)
```

```
## Analysis of Variance Table
```

```
##
## Response: Y
##            Df     Sum Sq    Mean Sq  F value    Pr(>F)
## X1          1 4.2241e+10 4.2241e+10 232.8643 < 2.2e-16 ***
## X2          1 3.3966e+10 3.3966e+10 187.2466 < 2.2e-16 ***
## X3          1 8.3760e+03 8.3760e+03   0.0000    0.9946
## X4          1 6.3802e+09 6.3802e+09  35.1724 3.755e-08 ***
## X5a         1 6.5717e+09 6.5717e+09  36.2282 2.501e-08 ***
## X5b         1 4.0175e+08 4.0175e+08   2.2148    0.1396
## X5c         1 5.5791e+07 5.5791e+07   0.3076    0.5803
## X6a         1 2.3273e+06 2.3273e+06   0.0128    0.9100
## X6b         1 1.1569e+08 1.1569e+08   0.6378    0.4263
## X6c         1 6.8443e+06 6.8443e+06   0.0377    0.8464
## X6d         1 1.7796e+08 1.7796e+08   0.9810    0.3242
## X7a         1 3.2105e+08 3.2105e+08   1.7698    0.1862
## X7b         1 6.5469e+07 6.5469e+07   0.3609    0.5493
## Residuals 107 1.9410e+10 1.8140e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
SSE.bestsubset = an$`Sum Sq`[14]


r.q4.tree<-rpart(Y~.,data=PR2.refit.df)
plot(r.q4.tree)
```



```
pop<-PR2.refit.df
SSE.Tree<-sum((predict(r.q4.tree)-pop$Y)^2)

max = apply(pop, 2 , max)
min = apply(pop, 2 , min)
scaled = as.data.frame(scale(pop, center = min, scale = max - min))
NN = neuralnet(Y~X1+X2+X3+X4+X5a+X5b+X5c+X6a+X6b+X6c+X6d+X7a+X7b, scaled , hidden = 6 , linear.output =

predict_testNN = compute(NN, scaled [,c(2:14)])
predict_testNN1 = (predict_testNN$net.result * (max(pop$Y) - min(pop$Y))) + min(pop$Y)
SSE.NN<-sum((pop$Y-predict_testNN1)^2)

round(data.frame(cbind(SSE.bestsubset,SSE.Tree,SSE.NN)),0)
```

```
##   SSE.bestsubset    SSE.Tree     SSE.NN
## 1    19409611507 38120812350 5034482236
```

Problem 2 d:

The model with the lowest SSE is the neural net model. Depends on the audience and what is being predicted, but I would prefer the linear regression model, because I can still consider sculpting the model down to selective variable and doing further analysis. Also, the linear regression model will be easier to explain to a diverse audience.

LM: 19409611507 Tree: 38120812350 NN: 4122397548

Problem 3:

3.) Use the PR3_Dataset data: Y is the outcome variable and indicates the number of awards earned by students at a high school in a year, X1 is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled. It is coded as 1 = "General", 2 = "Academic" and 3 = "Social", and X2 is a continuous predictor variable and represents students' scores on their math final exam. Answer the following questions: (20pts)

a-)Build a model to predict the number of awards earned by students, is the model significant? (5pts)

```
PR3.df = data.frame(read.csv("PR3_Dataset.csv"))

#PR3.df

Y = PR3.df$Y
X1a = ifelse(PR3.df$X1 == 1, 1, 0)
X1b = ifelse(PR3.df$X1 == 2, 1, 0)
X2 =PR3.df$X2

PR3.refit.df = data.frame(cbind(Y,X1a,X1b,X2))

head(PR3.refit.df)
```

```
##   Y X1a X1b X2
## 1 0   0   0 41
## 2 0   1   0 41
## 3 0   0   0 44
## 4 0   0   0 42
## 5 0   0   0 40
## 6 0   1   0 42
```

```
PR3.refit.reg = lm(data=PR3.refit.df)

summary(PR3.refit.reg)
```

```
##
## Call:
## lm(data = PR3.refit.df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7311 -0.5618 -0.1537  0.2851  4.4126
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.982998   0.382708  -5.181 5.45e-07 ***
## X1a         -0.212506   0.187433  -1.134    0.258
## X1b          0.266107   0.174482   1.525    0.129
## X2           0.047889   0.007773   6.161 4.03e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9019 on 196 degrees of freedom
## Multiple R-squared:  0.2773, Adjusted R-squared:  0.2662
## F-statistic: 25.07 on 3 and 196 DF,  p-value: 9.016e-14
```

Problem 3 a:

The model has an R^2 value of 0.2773, which shows a weak correlation of predictability, but can still say the model is significant, depending on the desired accuracy.

b-) Find the predicted number awards earned by students given the independent variables below and calculate 99% confidence interval. (5pts) X1 = 2, X2 = 75

```
predict.P3 = data.frame(cbind(X1a=0, X1b = 0, X2=75))

predict(PR3.refit.reg, predict.P3, interval = "confidence", level = 0.99)
```

```
##        fit       lwr      upr
## 1 1.608662 0.9423302 2.274994
```

```
#help(predict)
```

Problem 3 b:

Using the model, 1.608662 would be the predicted value with the confidence interval of 99%, the range would be 0.9423302 to 2.274994

c-) Fit the negative binomial model and compare it the model built in part a, which model is better? (10pts)

```
#help(glm)

PR3.refit.dfY = PR3.refit.df

PR3.refit.dfY$Y = PR3.refit.dfY$Y/(min(PR3.refit.dfY$Y)+max(PR3.refit.dfY$Y))

lmod <- glm(Y ~ X1a+X1b+X2, family = binomial, PR3.refit.dfY)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial
## glm!
```

```
#summary(lmod)
beta <- coef(lmod)
cbind(beta,exp(beta))
```

```
##                    beta
## (Intercept) -7.36124374 0.0006354077
## X1a         -0.42220481 0.6555997511
## X1b          0.76678485 2.1528334329
## X2           0.08481127 1.0885116167
```

```
summary(lmod)
```

```
##
## Call:
## glm(formula = Y ~ X1a + X1b + X2, family = binomial, data = PR3.refit.dfY)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9784  -0.3473  -0.2019   0.1410   1.5045
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -7.36124    1.71438  -4.294 1.76e-05 ***
## X1a         -0.42220    1.11485  -0.379  0.70490
## X1b          0.76678    0.82130   0.934  0.35050
## X2           0.08481    0.02946   2.879  0.00399 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 56.844  on 199  degrees of freedom
## Residual deviance: 38.101  on 196  degrees of freedom
## AIC: 64.599
##
## Number of Fisher Scoring iterations: 6
```

Problem 3 c:

Both models are trying to acheive something slightly different, with that being said, the bounds of the binomial model will ensure there is not negative amount of awards earned, but would also cap a student whose values exceeded the max.

Problem 4:

4.) Use the PR4_Dataset data, Y is a dichotomous response variable. X2, X3, and X4 are categorical variables: X2 has 3 levels, X3 and X4 have 2 levels (create dummy variables for the categorical variables). Answer the questions below: (20pts)

a-) Fit a regression model containing the predictor variables in first-order terms and interaction terms (e.g X1*X2) for all pairs of predictor variables. (5pts)

```
PR4.df = data.frame(read.csv("PR4_Dataset.csv"))
#PR4.df

Y =  PR4.df$Y
X1 = PR4.df$X1

X2a = as.numeric(PR4.df$X2 == 1)
X2b = as.numeric(PR4.df$X2 == 2)

X3 = as.numeric(PR4.df$X3 == 1)
X4 = as.numeric(PR4.df$X4 == 1)


X1X2a = X1*X2a
X1X2b = X1*X2b
X1X3 = X1*X3
X1X4 = X1*X4

X2aX3 = X2a*X3
X2aX4 = X2a*X4

X2bX3 = X2b*X3
X2bX4 = X2b*X4

X3X4 = X3*X4

PR4.refit.df = data.frame(cbind(Y,X1,X2a,X2b,X3,X4,X1X2a,X1X2b,X1X3,X1X4,X2aX3,X2aX4,X2bX3,X2bX4,X3X4))
head(PR4.refit.df)
```

```
##   Y X1 X2a X2b X3 X4 X1X2a X1X2b X1X3 X1X4 X2aX3 X2aX4 X2bX3 X2bX4 X3X4
## 1 1 33   1   0  0  0    33     0    0    0     0     0     0     0    0
## 2 1 35   1   0  0  0    35     0    0    0     0     0     0     0    0
## 3 0  6   1   0  0  0     6     0    0    0     0     0     0     0    0
## 4 1 60   1   0  0  0    60     0    0    0     0     0     0     0    0
## 5 0 18   0   0  0  1     0     0    0   18     0     0     0     0    0
## 6 0 26   0   0  0  0     0     0    0    0     0     0     0     0    0
```

```
lmod <- glm(Y ~ ., family = binomial, PR4.refit.df)
summary(lmod)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial, data = PR4.refit.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3855  -0.8886   0.4118   0.7943   2.0273
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.995363   0.556293  -3.587 0.000335 ***
## X1           0.038728   0.015343   2.524 0.011597 *
## X2a          2.151271   0.758426   2.836 0.004561 **
## X2b          0.844992   0.810105   1.043 0.296918
## X3           1.305590   0.832973   1.567 0.117025
## X4          -1.084417   1.100962  -0.985 0.324638
## X1X2a       -0.002890   0.024113  -0.120 0.904608
## X1X2b        0.005276   0.027528   0.192 0.848009
## X1X3        -0.021077   0.022438  -0.939 0.347549
## X1X4         0.021247   0.025814   0.823 0.410451
## X2aX3       -0.388653   0.867955  -0.448 0.654312
## X2aX4        0.137603   0.958732   0.144 0.885874
## X2bX3       -0.520501   0.913169  -0.570 0.568682
## X2bX4        0.025963   1.045480   0.025 0.980187
## X3X4         0.930980   0.835249   1.115 0.265016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 212.84  on 181  degrees of freedom
## AIC: 242.84
##
## Number of Fisher Scoring iterations: 5
```

```r
beta <- coef(lmod)
cbind(beta,exp(beta))
```

```
##                     beta
## (Intercept) -1.995362935 0.1359643
## X1           0.038728057 1.0394878
## X2a          2.151271304 8.5957793
## X2b          0.844991726 2.3279586
## X3           1.305589718 3.6898644
## X4          -1.084417221 0.3380988
## X1X2a       -0.002889739 0.9971144
## X1X2b        0.005275979 1.0052899
## X1X3        -0.021077155 0.9791434
## X1X4         0.021247212 1.0214745
## X2aX3       -0.388653175 0.6779694
## X2aX4        0.137603352 1.1475203
## X2bX3       -0.520500808 0.5942229
## X2bX4        0.025963402 1.0263034
## X3X4         0.930980480 2.5369954
```

Problem 4 a:

The model can be seen in the above code.

b-) Use the likelihood ratio test to determine whether all interaction terms can be dropped from the regression model; State the alternatives, full and reduced models, decision rule, and conclusion. (5pts)

```
lmodc<-glm(Y ~ X1 + X2a +X2b+X3+X4 , family = binomial, PR4.refit.df)
anova(lmodc,lmod,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X2b + X3 + X4
## Model 2: Y ~ X1 + X2a + X2b + X3 + X4 + X1X2a + X1X2b + X1X3 + X1X4 +
##     X2aX3 + X2aX4 + X2bX3 + X2bX4 + X3X4
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       190     215.36
## 2       181     212.84  9   2.5213   0.9803
```

Problem 4 b:

Yes, all the interaction terms can be dropped from the model (>Chi = .9803)

c.) Perform the backward variable selection method to find a model where all variables are significant and Conduct the Hosmer-Lemeshow goodness of fit test for the appropriateness of the logistic regression function by forming five groups. State the alternatives, decision rule, and conclusion. (5pts)

```
#ols_step_both_p(lmod,prem=0.05,details=FALSE)

lmodc<-glm(Y ~ X1 + X2a + X2b + X3 + X4 , family = binomial, PR4.refit.df)
lmodX4 = glm(Y ~ X1 + X2a + X2b + X3 , family = binomial, PR4.refit.df)

anova(lmodc,lmodX4,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X2b + X3 + X4
## Model 2: Y ~ X1 + X2a + X2b + X3
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1       190     215.36
## 2       191     215.36 -1 -0.005474    0.941
```

```
lmodc<-glm(Y ~ X1 + X2a + X2b + X3 , family = binomial, PR4.refit.df)
lmodX3 = glm(Y ~ X1 + X2a + X2b , family = binomial, PR4.refit.df)

anova(lmodc,lmodX3,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X2b + X3
## Model 2: Y ~ X1 + X2a + X2b
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       191     215.36
## 2       192     220.57 -1  -5.2093  0.02247 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmodc<-glm(Y ~ X1 + X2a + X2b + X3 , family = binomial, PR4.refit.df)
lmodX2b = glm(Y ~ X1 + X2a + X3 , family = binomial, PR4.refit.df)

anova(lmodc,lmodX2b,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X2b + X3
## Model 2: Y ~ X1 + X2a + X3
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       191     215.36
## 2       192     218.90 -1  -3.5407  0.05988 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmodc<-glm(Y ~ X1 + X2a + X3 , family = binomial, PR4.refit.df)
lmodX2a = glm(Y ~ X1 + X3 , family = binomial, PR4.refit.df)

anova(lmodc,lmodX2a,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X3
## Model 2: Y ~ X1 + X3
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       192      218.9
## 2       193      242.0 -1    -23.1 1.538e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmodc<-glm(Y ~ X1 + X2a + X3 , family = binomial, PR4.refit.df)
lmodX1 = glm(Y ~  X2a + X3 , family = binomial, PR4.refit.df)

anova(lmodc,lmodX1,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2a + X3
## Model 2: Y ~ X2a + X3
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       192     218.90
## 2       193     232.94 -1  -14.036 0.0001793 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(ResourceSelection)
hoslem.test(lmodc$y,fitted(lmodc),g=5)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  lmodc$y, fitted(lmodc)
## X-squared = 4.0794, df = 3, p-value = 0.253
```

Problem 4 c:

Through backward selection, we will keep: X1, X2a, X3

Ho: The model is good fit Ha: Model is not a good fit. Accept Null, P value > 0.05 (P value = 0.253). The model is a good fit.

d.)Use the model developed in part c and predict probability of Y for the following two cases and calculate 95% confidence interval. (5pts)

X1 X2 X3 X4

60 1 0 0

11 2 1 1

X2a = as.numeric(PR4.df$X2 == 1)

```
dat<-data.frame(cbind(X1=60,X2a=1,X3=0))
pre1=predict(lmodc,dat,type="link",se.fit=T)
LowerCL = pre1$fit-1.96*pre1$se.fit; UpperCL = pre1$fit+1.96*pre1$se.fit
Prediction = pre1$fit
results = round(cbind(LowerCL,Prediction,UpperCL),3)
ilogit(results)
```

```
##     LowerCL Prediction   UpperCL
## 1 0.7580467  0.8899274 0.9542616
```

```
dat<-data.frame(cbind(X1=11,X2a=0,X3=1))
pre1=predict(lmodc,dat,type="link",se.fit=T)
LowerCL = pre1$fit-1.96*pre1$se.fit; UpperCL = pre1$fit+1.96*pre1$se.fit
Prediction = pre1$fit
results = round(cbind(LowerCL,Prediction,UpperCL),3)
ilogit(results)
```

```
##     LowerCL Prediction   UpperCL
## 1 0.2857736  0.4272696 0.5817594
```

Problem 4 d:

The probability in case one is 88.99271% and 42.72696% in case 2. The 95% CI

LowerCL Prediction UpperCL

0.7580467 0.8899274 0.9542616

LowerCL Prediction UpperCL

0.2857736 0.4272696 0.5817594

Problem 5:

5.) Use the PR4_Dataset data. All variables including Y are continuous variables. Fit a regression model to predict Y. Is there a Multicollinearity in the data? Are the errors Normally distributed with constant variance? Are there any influential or outlier observations? check to see if auto-correlation persists in the data set, write null and alternatives hypothesis and calculate p value. (5 pts)

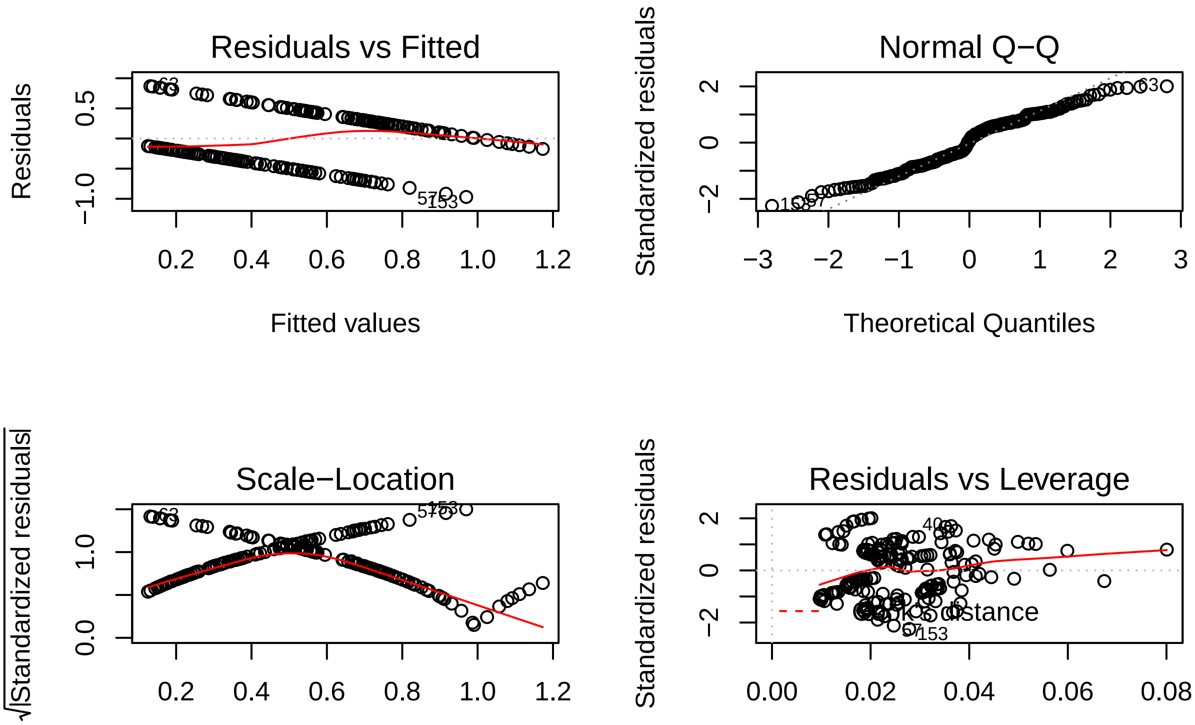
```
# I assume this problem meant to say PR5_Dataset, but included PR4 in case

par(mfrow=c(2,2))
PR4.df = data.frame(read.csv("PR4_Dataset.csv"))

PR4.reg = lm(Y~., data=PR4.df)
summary(PR4.reg)
```

```
##
## Call:
## lm(formula = Y ~ ., data = PR4.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96972 -0.35370  0.02827  0.32444  0.86832
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.722621   0.099424   7.268 9.08e-12 ***
## X1           0.006425   0.001720   3.736 0.000247 ***
## X2          -0.201265   0.037255  -5.402 1.94e-07 ***
## X3           0.143966   0.068127   2.113 0.035884 *
## X4          -0.004005   0.073692  -0.054 0.956712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4377 on 191 degrees of freedom
## Multiple R-squared:  0.247,  Adjusted R-squared:  0.2312
## F-statistic: 15.66 on 4 and 191 DF,  p-value: 4.235e-11
```

```
vif(PR4.reg)
```

```
##       X1       X2       X3       X4
## 1.076201 1.063295 1.142607 1.145966
```

```
plot(PR4.reg)
```




```r
# I assume this problem meant to say PR5_Dataset

par(mfrow=c(2,2))
PR5.df = data.frame(read.csv("PR5_Dataset.csv"))

PR5.reg = lm(Y~., data=PR5.df)
summary(PR5.reg)
```

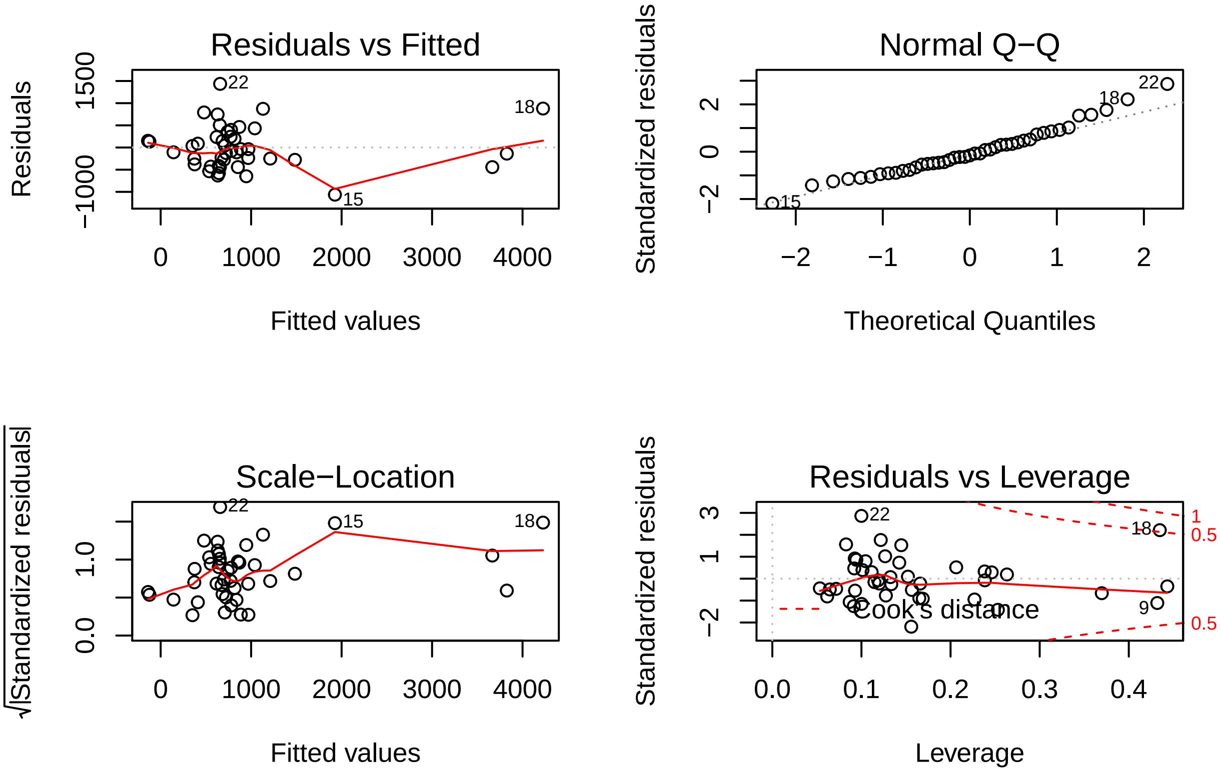
```
##
## Call:
## lm(formula = Y ~ ., data = PR5.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1063.26  -329.03   -77.92   239.84  1434.78
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.287e+03  2.171e+03  -0.593   0.5570
## X1           9.509e+03  7.828e+03   1.215   0.2324
## X2           1.889e+01  3.119e+01   0.606   0.5484
## X3           6.129e+02  8.021e+01   7.641 4.82e-09 ***
## X4          -1.670e-01  8.161e-02  -2.046   0.0481 *
## X5           6.445e-01  2.513e-01   2.564   0.0146 *
## X6          -3.102e+01  8.881e+01  -0.349   0.7289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 528.4 on 36 degrees of freedom
```

```
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.7329
## F-statistic:  20.2 on 6 and 36 DF,  p-value: 3.491e-10
```

```r
vif(PR5.reg)
```

```
##       X1       X2       X3       X4       X5       X6
## 2.656652 1.653578 1.337545 2.686929 1.367983 1.098401
```

```r
plot(PR5.reg)
```



Problem 5 Answer:

There is no multicoliniearity in the dataset (all VIF values are below 5).

The R^squared is .771, which would show as a decent prediction model.

There appears to be one outlier in cooks distance (case 18) that could be investigated further.