Predictive Analytics Problem Set 4

Diyasha Basu

19-02-2026

Problem to demonstrate the utility of nonlinear regression over linear regression.

Get the fgl data set from "MASS" library.

```
df_1 = fgl
head(df_1)
```

```
##      RI    Na   Mg   Al    Si    K   Ca Ba   Fe type
## 1  3.01 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00 WinF
## 2 -0.39 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00 WinF
## 3 -1.82 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00 WinF
## 4 -0.34 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00 WinF
## 5 -0.58 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00 WinF
## 6 -2.04 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26 WinF
```

(a) Considering the refractive index (RI) of "Vehicle Window glass" as the variable of interest and assuming linearity of regression, run multiple linear regression of RI on different metallic oxides. From the p value, report which metallic oxide best explains the refractive index.

```
df_1subset = subset(df_1, type == "Veh")
model = lm(RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = df_1subset)
summary(model)
```

```
##
## Call:
## lm(formula = RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = df_1subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29194 -0.08582  0.00072  0.10740  0.33524
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 131.4641    47.2669   2.781  0.02388 *
## Na           -0.4333     0.3509  -1.235  0.25190
## Mg           -0.2866     1.0075  -0.285  0.78325
## Al           -0.8909     0.5550  -1.605  0.14713
## Si           -1.8824     0.4993  -3.770  0.00547 **
## K            -2.4232     0.9725  -2.492  0.03743 *
## Ca            1.5326     0.5818   2.634  0.02998 *
## Ba            0.3517     2.6904   0.131  0.89922
## Fe            3.8931     0.9581   4.063  0.00362 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2621 on 8 degrees of freedom
## Multiple R-squared:  0.9906, Adjusted R-squared:  0.9813
## F-statistic: 105.9 on 8 and 8 DF,  p-value: 2.622e-07
```

```
stargazer(model, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                 RI
## -----------------------------------------------
## Na                            -0.433
##                               (0.351)
##
## Mg                            -0.287
##                               (1.007)
##
## Al                            -0.891
##                               (0.555)
##
## Si                           -1.882***
##                               (0.499)
##
## K                            -2.423**
##                               (0.973)
##
## Ca                            1.533**
##                               (0.582)
##
## Ba                             0.352
##                               (2.690)
##
## Fe                           3.893***
##                               (0.958)
##
## Constant                     131.464**
##                              (47.267)
##
## -----------------------------------------------
## Observations                    17
## R2                             0.991
## Adjusted R2                    0.981
## Residual Std. Error       0.262 (df = 8)
## F Statistic           105.887*** (df = 8; 8)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```r
summary(model)$coefficients
```

```
##                Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 131.4640676 47.2669236  2.7813121 0.023876172
## Na           -0.4333080  0.3508773 -1.2349274 0.251895370
## Mg           -0.2866243  1.0074637 -0.2845009 0.783251988
## Al           -0.8908690  0.5550086 -1.6051446 0.147129402
## Si           -1.8823864  0.4993058 -3.7700067 0.005465591
## K            -2.4231984  0.9725295 -2.4916451 0.037426154
## Ca            1.5326244  0.5817872  2.6343387 0.029975590
## Ba            0.3517015  2.6904136  0.1307240 0.899221141
## Fe            3.8931318  0.9580806  4.0634699 0.003616000
```

Conclusion : From the p-values, it is clear that Iron (Fe) oxide has the lowest p-value and hence best explains the refractive index.

(b) Run a simple linear regression of RI on the best predictor chosen in (a).

```r
fit_simple = lm(RI ~ Fe, data = df_1subset)
summary(fit_simple)
```

```
##
## Call:
## lm(formula = RI ~ Fe, data = df_1subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2324 -1.0693 -0.2715  0.2907  3.7707
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5007     0.4861  -1.030   0.3193
## Fe            8.1362     4.0780   1.995   0.0645 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.759 on 15 degrees of freedom
## Multiple R-squared:  0.2097, Adjusted R-squared:  0.157
## F-statistic: 3.981 on 1 and 15 DF,  p-value: 0.06452
```

```r
stargazer(fit_simple, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                                 RI
## -----------------------------------------------
## Fe                            8.136*
##                              (4.078)
```

```
##
## Constant                          -0.501
##                                   (0.486)
##
## -----------------------------------------------
## Observations                        17
## R2                                 0.210
## Adjusted R2                        0.157
## Residual Std. Error          1.759 (df = 15)
## F Statistic                3.981* (df = 1; 15)
## ===============================================
## Note:                 *p<0.1; **p<0.05; ***p<0.01
```

(c) Can you further improve the regression of the refractive index of "Vehicle Window glass" on the predictor chosen by you in part (a)? Give the new fitted model and compare its performance with the model in (b).

```
fit_quadra = lm(RI ~ Fe + I(Fe^2), data = df_1subset)

summary(fit_quadra)

##
## Call:
## lm(formula = RI ~ Fe + I(Fe^2), data = df_1subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6215 -1.1715 -0.1345  0.5985  3.5485
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2785     0.4712  -0.591    0.564
## Fe          -12.1810    12.0408  -1.012    0.329
## I(Fe^2)      65.9600    37.0798   1.779    0.097 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.645 on 14 degrees of freedom
## Multiple R-squared:  0.3554, Adjusted R-squared:  0.2633
## F-statistic:  3.86 on 2 and 14 DF,  p-value: 0.04623
```

Comparison :

```
cat("R^2 value for simple model : ",summary(fit_simple)$r.squared)

## R^2 value for simple model :  0.2097192

cat("R^2 value for quadratic model : ",summary(fit_quadra)$r.squared)

## R^2 value for quadratic model :  0.355413
```

Conclusion : The quadratic model improves the fit compared to the simple linear model, as indicated by the higher R² and lower residual standard error. The overall model becomes statistically significant at the 5% level. However, the individual coefficients are not strongly significant, suggesting only moderate evidence of a nonlinear relationship between RI and Fe.

Problem to demonstrate multicollinearity.

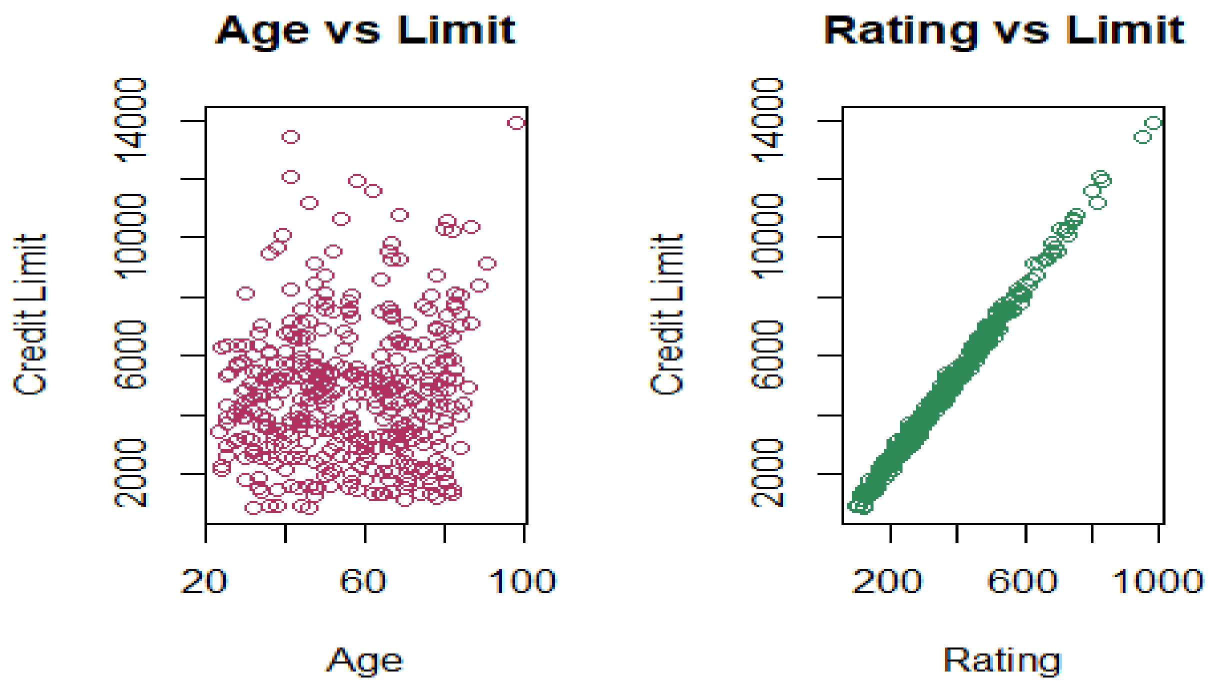
Consider the Credit data in the ISLR library. Choose balance as the response and Age, Limit and Rating as the predictors.

```
df_2 = Credit
head(df_2)
```

```
##   ID  Income Limit Rating Cards Age Education Gender Student Married
Ethnicity
## 1  1  14.891  3606    283     2  34        11   Male      No     Yes
Caucasian
## 2  2 106.025  6645    483     3  82        15 Female     Yes     Yes
Asian
## 3  3 104.593  7075    514     4  71        11   Male      No      No
Asian
## 4  4 148.924  9504    681     3  36        11 Female      No      No
Asian
## 5  5  55.882  4897    357     2  68        16   Male      No     Yes
Caucasian
## 6  6  80.180  8047    569     4  77        10   Male      No      No
Caucasian
##   Balance
## 1     333
## 2     903
## 3     580
## 4     964
## 5     331
## 6    1151
```

(a) Make a scatter plot of (i) Age versus Limit and (ii) Rating Versus Limit. Comment on the scatter plot.

```
par(mfrow = c(1, 2))

plot(df_2$Age, df_2$Limit, main = "Age vs Limit", xlab = "Age", ylab =
"Credit Limit", col ="maroon")

plot(df_2$Rating, df_2$Limit, main = "Rating vs Limit", xlab = "Rating", ylab
= "Credit Limit" , col = "seagreen")
```



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

Comments : i. Age vs Limit

- Type of relationship: No clear linear relationship
- Direction: No clear direction (neither upward nor downward direction)
- Strength: Very weak

The points are widely scattered without any visible pattern. This suggests that Age does not strongly explain Credit Limit.

(ii) Rating vs Limit

- Type of relationship: Strong linear relationship
- Direction: Positive
- Strength: Very strong

The points lie almost perfectly along a straight upward-sloping line. This indicates that as Rating increases, Limit increases almost proportionally.

(b) Run three separate regressions: (i) Balance on Age and Limit (ii) Balance on Age, Rating and Limit (iii) Balance on Rating and Limit. Present all the regression output in a single table using stargazer. What is the marked difference that you can observe from the output?

```
model2_1 = lm(Balance ~ Age + Limit, data = df_2)

model2_2 = lm(Balance ~ Age + Rating + Limit, data = df_2)

model2_3 = lm(Balance ~ Rating + Limit, data = df_2)

stargazer(model2_1, model2_2, model2_3, type = "text", title = "Regression
Results: Balance as Response", column.labels = c("Age + Limit", "Age + Rating
+ Limit", "Rating + Limit"))

##
## Regression Results: Balance as Response
##
============================================================================
=================
##                                                  Dependent variable:
##                       -----------------------------------------------------
--------------------
##                                                        Balance
##                             Age + Limit          Age + Rating + Limit
Rating + Limit
##                                 (1)                       (2)
(3)
## -------------------------------------------------------------------------
--------------------
## Age                          -2.291***                 -2.346***
##                              (0.672)                   (0.669)
##
## Rating                                                  2.310**
2.202**
##                                                        (0.940)
(0.952)
##
## Limit                        0.173***                   0.019
0.025
##                              (0.005)                   (0.063)
(0.064)
##
## Constant                   -173.411***               -259.518***
-377.537***
##                              (43.828)                  (55.882)
(45.254)
##
## -------------------------------------------------------------------------
--------------------
## Observations                    400                       400
```

```
400
## R2                          0.750                     0.754
0.746
## Adjusted R2                 0.749                     0.752
0.745
## Residual Std. Error    230.532 (df = 397)       229.080 (df = 396)
232.320 (df = 397)
## F Statistic         594.988*** (df = 2; 397) 403.718*** (df = 3; 396)
582.820*** (df = 2; 397)
##
==============================================================================
=================
## Note:                                                              *p<0.1;
**p<0.05; ***p<0.01
```

Observations : The key difference:

1. In model (i), Limit is highly significant.
2. In model (iii), Rating is highly significant.
3. But in model (ii) **(both included)** :

- One of them often becomes insignificant.
- Standard errors increase.
- Coefficient signs or magnitudes may shift.

This happens because Rating and Limit are highly correlated. That is multicollinearity.

(c) Calculate the variance inflation factor (VIF) and comment on multicollinearity.

```
vif_values = vif(model2_2)
print(vif_values)
```

```
##        Age     Rating      Limit
##   1.011385 160.668301 160.592880
```

Comment on multi colinearity : There is severe multi collinearity between Rating and Limit, as indicated by extremely high VIF values (≈160). Age shows no multi collinearity. The instability in the regression coefficients when both Rating and Limit are included is due to this strong linear dependence.

Problem to demonstrate the detection of outlier, leverage and influential points.

Attach “Boston” data from MASS library in R. Select median value of owner occupied homes, as the response and per capita crime rate, nitrogen oxides concentration, proportion of blacks and percentage of lower status of the population as predictors. The objective is to fit a multiple linear regression model of the response on the predictors. With reference to this problem, detect outliers, leverage points and influential points if any.

```r
df_3 = Boston
df_3subset = df_3[, c("medv", "crim", "nox", "black", "lstat")]
head(df_3subset)
```

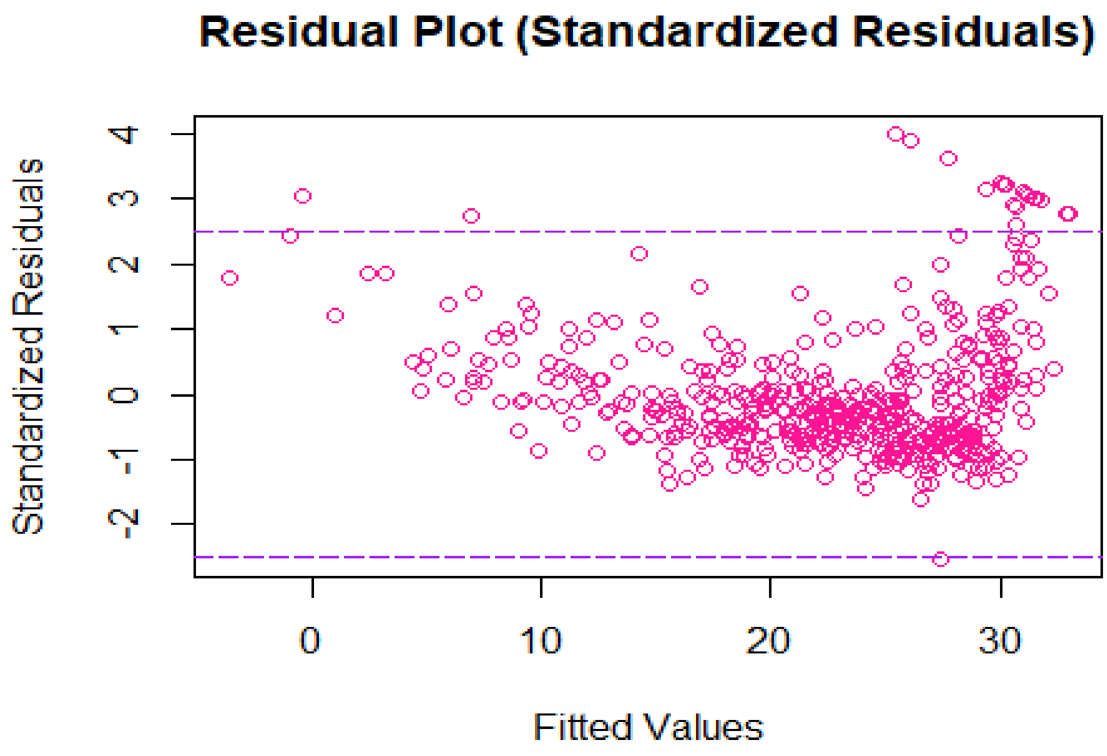
```
##   medv    crim   nox  black lstat
## 1 24.0 0.00632 0.538 396.90  4.98
## 2 21.6 0.02731 0.469 396.90  9.14
## 3 34.7 0.02729 0.469 392.83  4.03
## 4 33.4 0.03237 0.458 394.63  2.94
## 5 36.2 0.06905 0.458 396.90  5.33
## 6 28.7 0.02985 0.458 394.12  5.21
```

```r
boston_model = lm(medv ~ crim + nox + black + lstat, data = df_3subset)
summary(boston_model)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + black + lstat, data = df_3subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.564  -4.004  -1.504   2.178  24.608
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 30.053584   2.170839  13.844   <2e-16 ***
## crim        -0.059424   0.037755  -1.574    0.116
## nox          3.415809   3.056602   1.118    0.264
## black        0.006785   0.003408   1.991    0.047 *
## lstat       -0.918431   0.050167 -18.307   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.183 on 501 degrees of freedom
## Multiple R-squared:  0.5517, Adjusted R-squared:  0.5481
## F-statistic: 154.1 on 4 and 501 DF,  p-value: < 2.2e-16
```

Residual Plot :

```r
std_res = rstandard(boston_model)

plot(boston_model$fitted.values, std_res, xlab = "Fitted Values", ylab =
"Standardized Residuals", main = "Residual Plot (Standardized Residuals)" ,
col = "deeppink")

abline(h = c(-2.5, 2.5), col = "purple", lty =5)
```



Calculation for Outliers :

```
outliers = which(abs(std_res) > 2.5)

data.frame(Observation = outliers, Standardized_Residual = std_res[outliers])

##     Observation Standardized_Residual
## 162         162              2.768907
## 163         163              2.783878
## 164         164              2.988259
## 167         167              3.069974
## 187         187              3.194723
## 196         196              3.011109
## 204         204              2.897489
## 205         205              3.005419
## 215         215              2.749729
## 226         226              3.228933
## 229         229              2.591706
## 234         234              2.852734
## 258         258              3.224714
## 263         263              3.148286
## 268         268              3.605022
## 284         284              3.050345
## 369         369              3.015299
## 370         370              3.092472
## 371          371              2.970057
## 372          372              3.992694
## 373          373              3.890059
## 413          413              3.029650
## 506          506             -2.522613
```

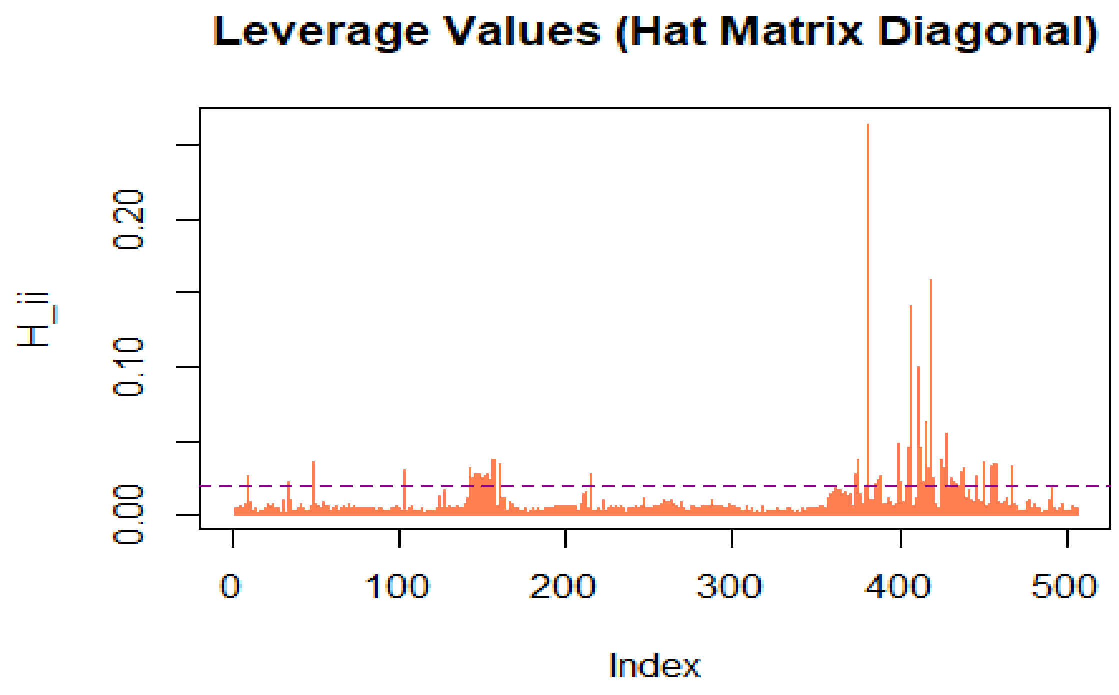
Consistency Check : The graphical method (residual plot) and the numerical method (|standardized residual| > 2.5) identify the same observations as outliers. Hence, the two approaches are consistent.

Leverage Points :

```
n = nrow(Boston)
p = length(coef(boston_model))

H = hatvalues(boston_model)

threshold_lev = 2*p/n
threshold_lev

## [1] 0.01976285

leverage_points = which(H > threshold_lev)

leverage_points

##   9  33  49 103 142 143 144 145 146 147 148 149 150 151 152 153 154 155
156 157
##   9  33  49 103 142 143 144 145 146 147 148 149 150 151 152 153 154 155
156 157
## 160 215 374 375 381 386 387 388 399 401 405 406 411 412 413 414 415 416
417 418
## 160 215 374 375 381 386 387 388 399 401 405 406 411 412 413 414 415 416
417 418
## 419 420 424 425 426 427 428 430 431 432 433 434 435 437 438 439 446 451
455 456
## 419 420 424 425 426 427 428 430 431 432 433 434 435 437 438 439 446 451
455 456
## 457 458 467 491
## 457 458 467 491
```

Plot :

```
plot(H, type = "h", main = "Leverage Values (Hat Matrix Diagonal)", ylab =
"H_ii" , col= "coral" )

abline(h = threshold_lev, col = "darkmagenta", lty = 2)
```



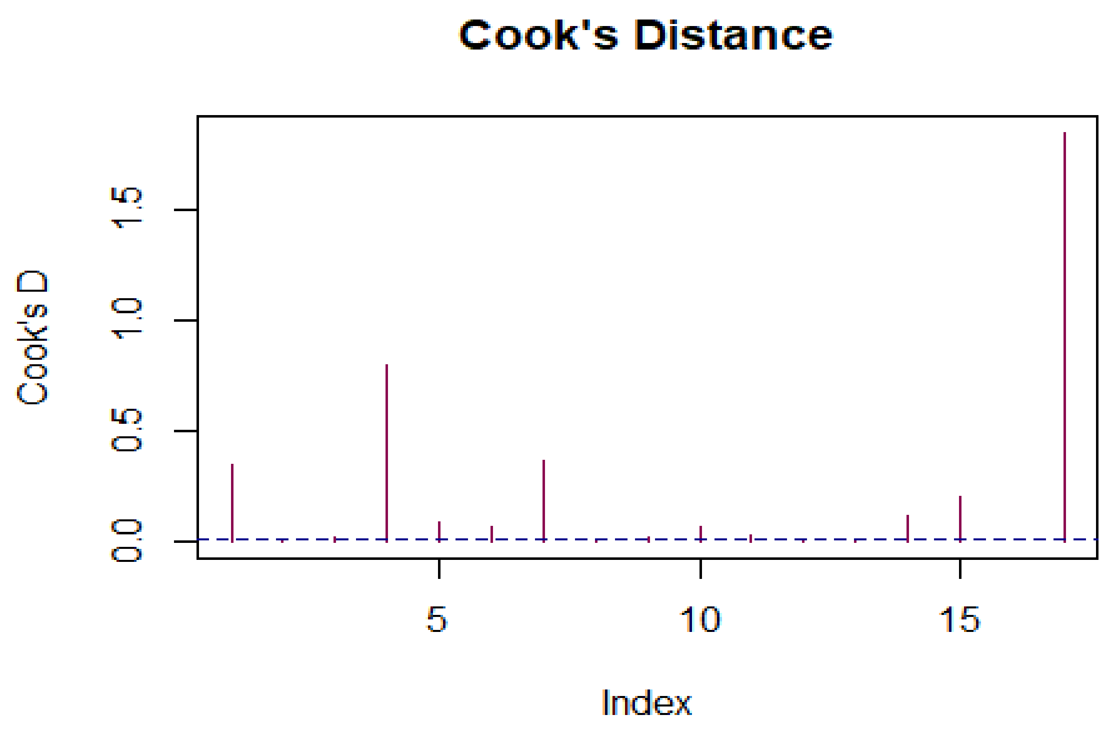
Comment : The leverage plot shows that a small number of observations exceed the leverage threshold, indicating the presence of high leverage points. Most observations have low leverage. These high leverage points may potentially influence the regression model.

Influential Points (Cook's Distance) :

```
cooks_dis = cooks.distance(model)

thre_cook = 4/n
thre_cook

## [1] 0.007905138

inf_points = which(cooks_dis > thre_cook)

inf_points

## 147 149 150 151 152 153 155 156 157 160 161 163
##   1   3   4   5   6   7   9  10  11  14  15  17
```

Plot :

```
plot(cooks_dis, type = "h", main = "Cook's Distance", ylab = "Cook's D", col=
"deeppink4")

abline(h = thre_cook, col = "blue4", lty = 2)
```



Comment : The Cook's distance plot indicates the presence of a few influential observations, with one observation having particularly high influence. Most observations have negligible influence. The highly influential points may substantially affect the regression coefficients and should be examined further.

Effect of Removing Influential Points :

```
model_red = lm(medv ~ crim + nox + black + lstat, data = Boston[-inf_points,
])

summary(model_red)

##
## Call:
## lm(formula = medv ~ crim + nox + black + lstat, data = Boston[-inf_points,
##     ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.645  -3.985  -1.529   2.173  24.498
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.916753   2.181633  13.713   <2e-16 ***
## crim        -0.057530   0.037984  -1.515   0.1305
## nox          4.018540   3.083138   1.303   0.1931
## black        0.006753   0.003424   1.972   0.0491 *
## lstat       -0.933039   0.051488 -18.121   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.206 on 489 degrees of freedom
## Multiple R-squared:  0.5525, Adjusted R-squared:  0.5489
## F-statistic:   151 on 4 and 489 DF,  p-value: < 2.2e-16
```

Comparison to original model : After removing influential observations, the regression coefficients and overall model fit remain broadly similar. Although some minor changes in magnitude and significance occur, the general conclusions remain unchanged. Therefore, the regression model is reasonably robust to influential observations.