ASSIGNMENT_4_MDTS4214_0733

ROUNAK SENGUPTA

2026-02-19

5.Problem to demonstrate the utility of non- regression over linear regression.

Get the fgl data set from “MASS” library.

(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.

```
library(MASS)
data(fgl)
str(fgl)
```

```
## 'data.frame':    214 obs. of  10 variables:
##  $ RI  : num  3.01 -0.39 -1.82 -0.34 -0.58 ...
##  $ Na  : num  13.6 13.9 13.5 13.2 13.3 ...
##  $ Mg  : num  4.49 3.6 3.55 3.69 3.62 3.61 3.6 3.61 3.58 3.6 ...
##  $ Al  : num  1.1 1.36 1.54 1.29 1.24 1.62 1.14 1.05 1.37 1.36 ...
##  $ Si  : num  71.8 72.7 73 72.6 73.1 ...
##  $ K   : num  0.06 0.48 0.39 0.57 0.55 0.64 0.58 0.57 0.56 0.57 ...
##  $ Ca  : num  8.75 7.83 7.78 8.22 8.07 8.07 8.17 8.24 8.3 8.4 ...
##  $ Ba  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Fe  : num  0 0 0 0 0 0.26 0 0 0 0.11 ...
##  $ type: Factor w/ 6 levels "WinF","WinNF",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
table(fgl$type)
```

```
##
##  WinF WinNF   Veh   Con  Tabl  Head
##    70    76    17    13     9    29
```

```
head(fgl)
```

```
##      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
```

```
#install.packages("stargazer")
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
veh=subset(fgl,type=="Veh")

predictors=c("Na","Mg","Al","Si","K","Ca","Ba","Fe")

summary(veh$RI)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -1.90000 -1.45000 -0.31000 -0.03647  0.32000  4.11000
```

```
summary(veh[, predictors])
```

```
##        Na              Mg              Al              Si       
##  Min.   :12.16   Min.   :3.340   Min.   :0.580   Min.   :71.36  
##  1st Qu.:13.24   1st Qu.:3.400   1st Qu.:0.910   1st Qu.:72.04  
##  Median :13.42   Median :3.530   Median :1.280   Median :72.64  
##  Mean   :13.44   Mean   :3.544   Mean   :1.201   Mean   :72.40  
##  3rd Qu.:13.64   3rd Qu.:3.650   3rd Qu.:1.380   3rd Qu.:72.70  
##  Max.   :14.32   Max.   :3.900   Max.   :1.760   Max.   :73.01  
##        K                Ca              Ba                  Fe         
##  Min.   :0.0000   Min.   :8.320   Min.   :0.000000   Min.   :0.00000  
##  1st Qu.:0.1600   1st Qu.:8.530   1st Qu.:0.000000   1st Qu.:0.00000  
##  Median :0.5600   Median :8.790   Median :0.000000   Median :0.00000  
##  Mean   :0.4065   Mean   :8.783   Mean   :0.008824   Mean   :0.05706  
##  3rd Qu.:0.5700   3rd Qu.:8.930   3rd Qu.:0.000000   3rd Qu.:0.09000  
##  Max.   :0.6100   Max.   :9.650   Max.   :0.150000   Max.   :0.37000
```

```
model1=lm(RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = veh)
summary(model1)
```

```
##
## Call:
## lm(formula = RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = veh)
##
## 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
```

```
pvals=summary(model1)$coefficients[-1, 4]
pvals=sort(pvals)

pvals
```

```
##          Fe          Si          Ca           K          Al          Na
## 0.003616000 0.005465591 0.029975590 0.037426154 0.147129402 0.251895370
##          Mg          Ba
## 0.783251988 0.899221141
```

```
best_pred=names(pvals)[1]
best_pred
```

```
## [1] "Fe"
```

Conclusion :

The metallic oxide with the smallest p-value is:

Fe :Iron

It best explains RI under linear regression assumption.

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

```r
form_simple=as.formula(paste("RI ~", best_pred))

fit_simple=lm(form_simple, data = veh)
summary(fit_simple)
```

```
##
## Call:
## lm(formula = form_simple, data = veh)
##
## 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
summary(fit_simple)$r.squared
```

```
## [1] 0.2097192
```

Conclusion:

RI= -0.5007 + 8.1362 * Fe

intercept (β_0) = −0.5007

Fe (β_1) = 8.1362 ; p-value = 0.06452

This indicates that for every one-unit increase in Fe content, the refractive index (RI) increases on average by 8.1362 units.

The model explains 20.97% of the variability in refractive index.

The predictor Fe has a p-value of 0.0645, which is:

1. Not significant at the 5% level

2. Marginally significant at the 10% level

(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).

```r
simple_model = lm(RI ~ Fe, data = veh)

summary(simple_model)
```

```
##
## Call:
## lm(formula = RI ~ Fe, data = veh)
##
## 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
```

```
r2_simple = summary(simple_model)$r.squared
adjr2_simple = summary(simple_model)$adj.r.squared
```

```
model_quad = lm(RI ~ Fe + I(Fe^2), data = veh)

summary(model_quad)
```

```
##
## Call:
## lm(formula = RI ~ Fe + I(Fe^2), data = veh)
##
## 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
```

```
r2_quad = summary(model_quad)$r.squared
adjr2_quad = summary(model_quad)$adj.r.squared

r2_simple
```

```
## [1] 0.2097192
```

```
r2_quad
```

```
## [1] 0.355413
```

```
adjr2_simple
```

```
## [1] 0.1570338
```

```
adjr2_quad
```

```
## [1] 0.2633292
```

The quadratic model improves the regression if:

R^2 (quadratic) > R^2 (simple)

Adjusted R^2 increases

The quadratic term is marginally significant at 10% level ($p = 0.097$), suggesting that nonlinear regression provides a slight improvement over the simple linear model,though the improvement is not strong at the 5% significance level.

Problem Set 4: Some Potential

Problems in Multiple Linear Regression

1. Problem to demonstrate multicollinearity

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

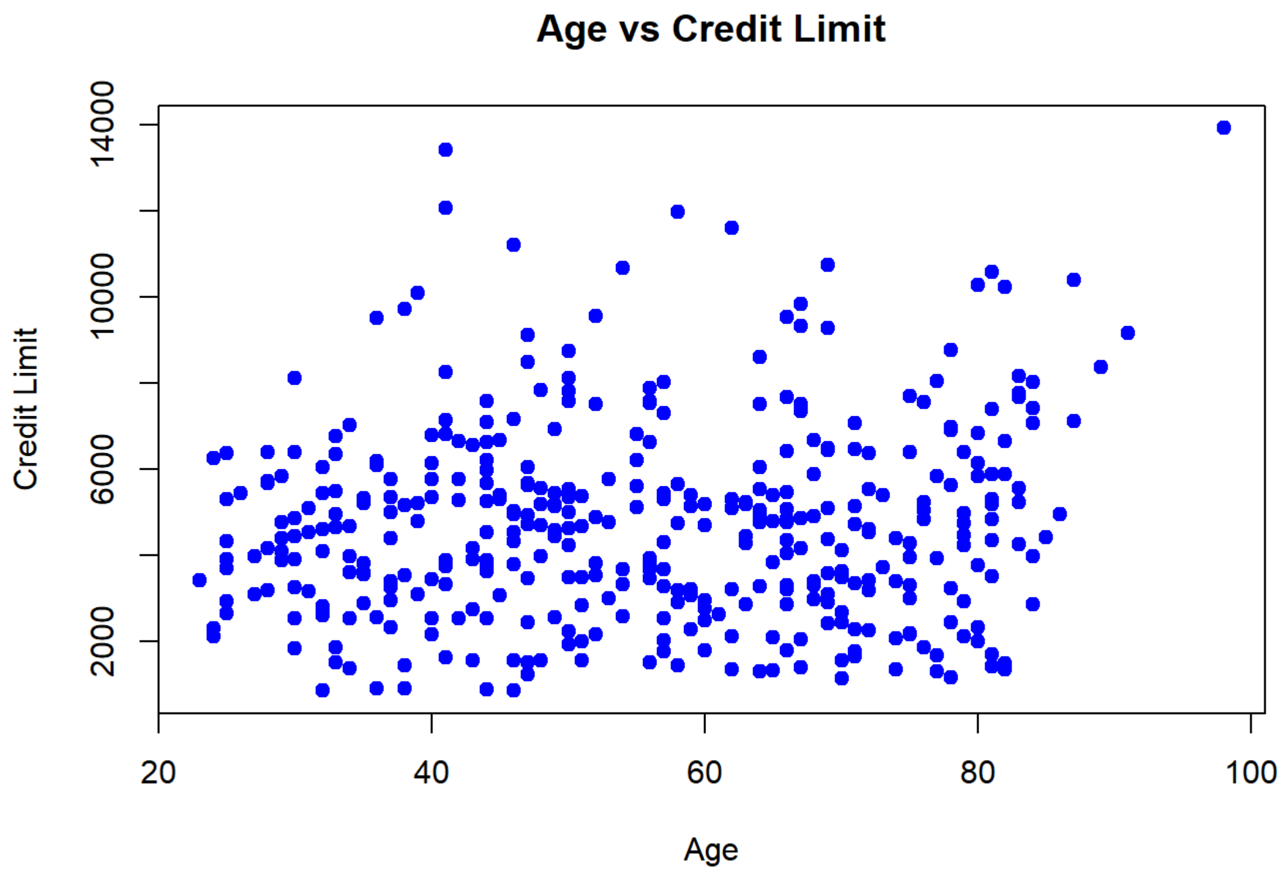
(a) Make a scatter plot of (i) Age versus Limit and (ii) Rating Versus Limit.

Comment on the scatter plot

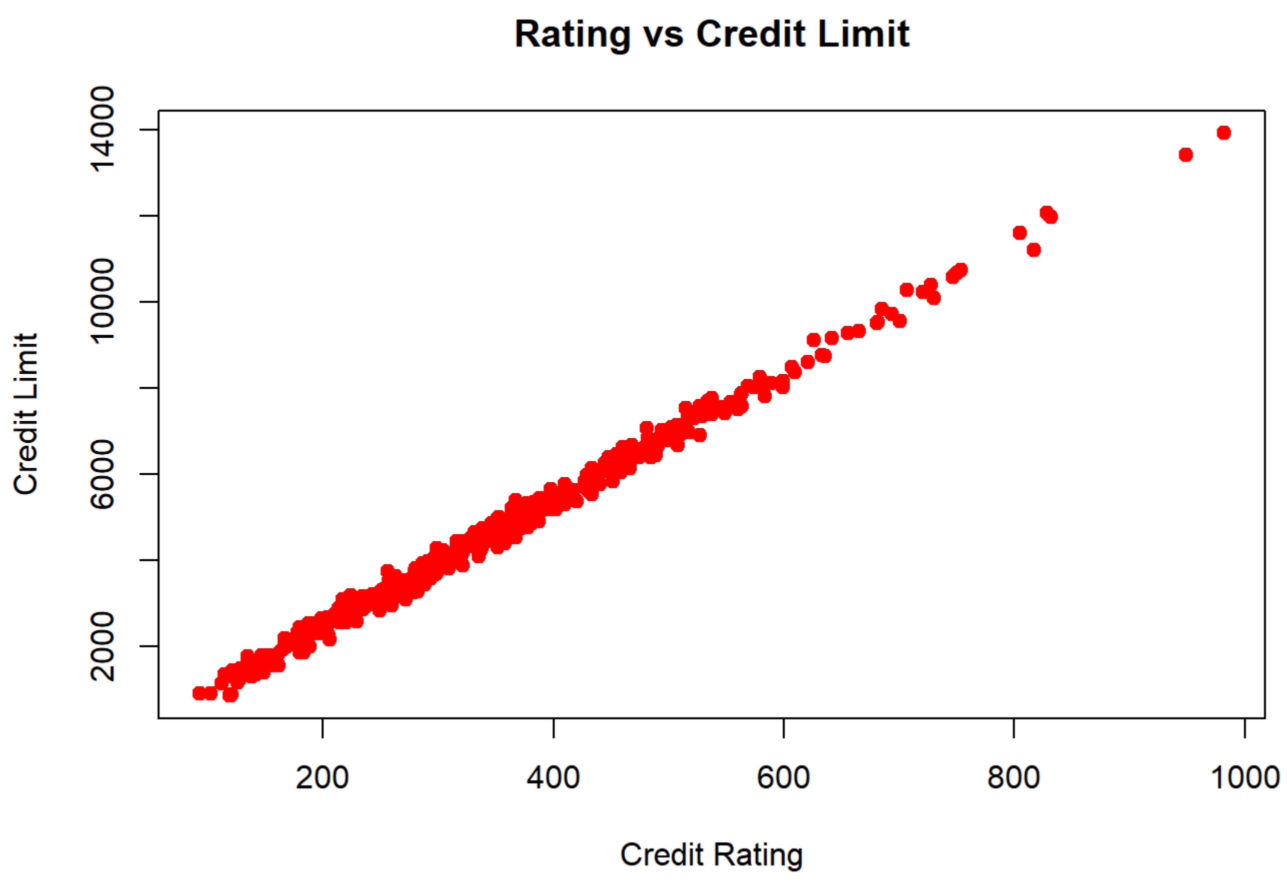
```
library(ISLR)
library(car)
```

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

```
data(Credit)

plot(Credit$Age, Credit$Limit,
     xlab = "Age",
     ylab = "Credit Limit",
     main = "Age vs Credit Limit",
     pch = 19,
     col = "blue")
```



```
plot(Credit$Rating, Credit$Limit,
     xlab = "Credit Rating",
     ylab = "Credit Limit",
     main = "Rating vs Credit Limit",
     pch = 19,
     col = "red")
```



Conclusion: i. Age vs Limit: Direction: Slight positive relationship (as Age increases, Limit tends to increase slightly).

Strength: Weak.

Type: Approximately linear, but very scattered.

The scatter plot of Age versus Limit shows a wide spread of points with no strong visible pattern. There may be a slight upward trend, but the relationship is weak and not very pronounced. This suggests that Age is not strongly associated with Credit Limit.

ii. Rating vs Limit: Direction: Strong positive relationship.

Strength: Very strong.

Type: Linear.

The scatter plot of Rating versus Limit shows points clustered tightly around an upward-sloping straight line. This indicates a very strong positive linear relationship. As Credit Rating increases, Credit 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?

```
reg1 <- lm(Balance ~ Age + Limit, data = Credit)
reg2 <- lm(Balance ~ Age + Rating + Limit, data = Credit)
reg3 <- lm(Balance ~ Rating + Limit, data = Credit)
```

```
stargazer(reg1, reg2, reg3,
          type = "text",
          title = "Regression Results Demonstrating Multicollinearity",column.labels = c("Age
+ Limit", "Age + Rating + Limit", "Rating + Limit"),
          digits = 3)
```

```
##
## Regression Results Demonstrating Multicollinearity
## ==============================================================================================
====
##                                                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 = 39
7)
## 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
```

Coefficient changes for Limit:

In Reg 1, Limit has 0.173* (highly significant).

In Reg 2, after adding Rating, Limit drops to 0.019 and becomes insignificant.

In Reg 3, Limit is also small (0.025) and insignificant.

Standard errors for Limit increase dramatically in Regression 2 compared to Regression 1.

This is a hallmark of multicollinearity: predictor coefficients become unstable, and p-values increase.

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

```
vif(reg1)
```

```
##      Age    Limit 
## 1.010283 1.010283
```

```
vif(reg2)
```

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

```
vif(reg3)
```

```
##   Rating    Limit 
## 160.4933 160.4933
```

Model 2: Balance ~ Age + Rating + Limit

VIF for Rating ≈ 160

VIF for Limit ≈ 160

Interpretation:

Extremely high VIF values indicate severe multicollinearity between Rating and Limit.

The coefficients of Rating and Limit are unstable, standard errors inflate, and significance may be misleading (as seen in the Stargazer table).

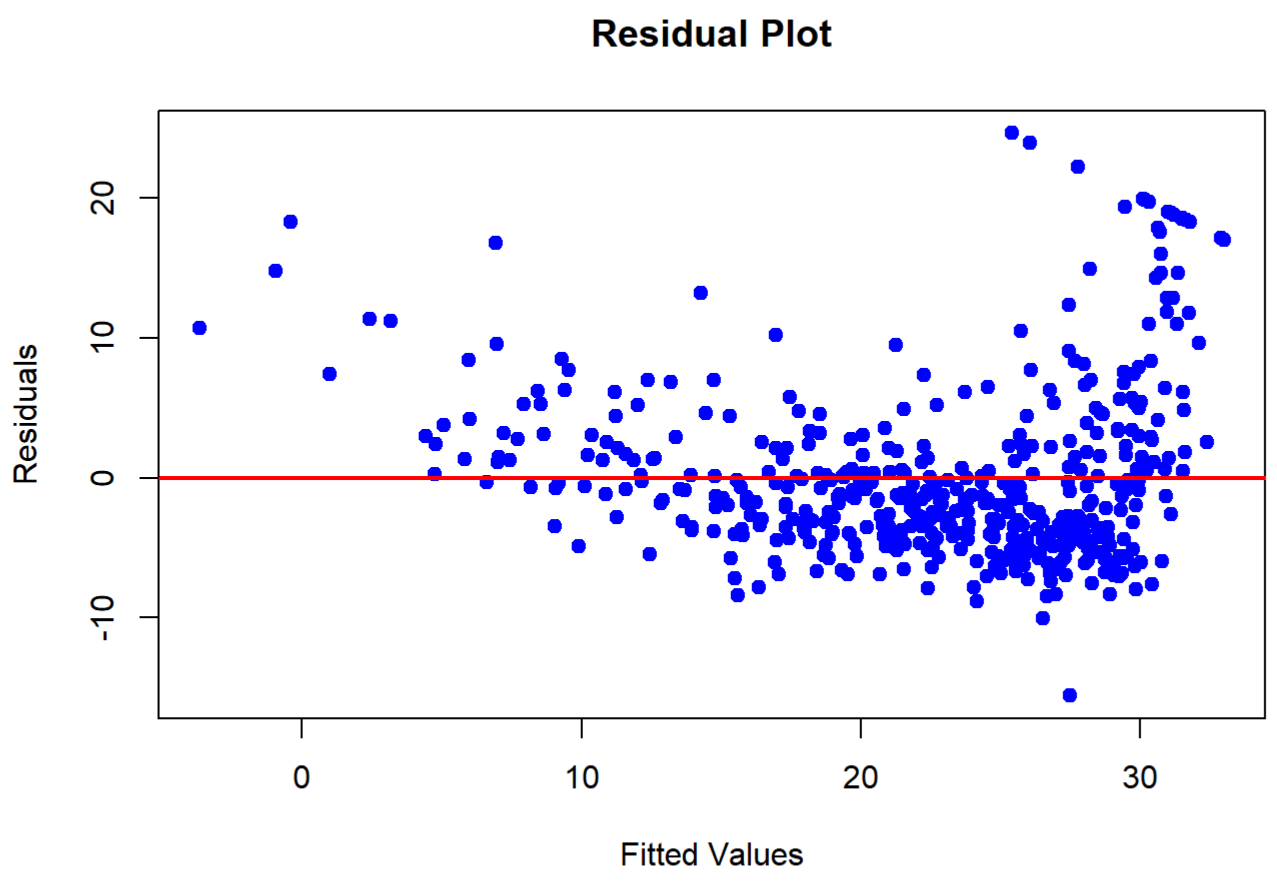
2. Problem to demonstrate the detection of outlier, leverage and influential points

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.

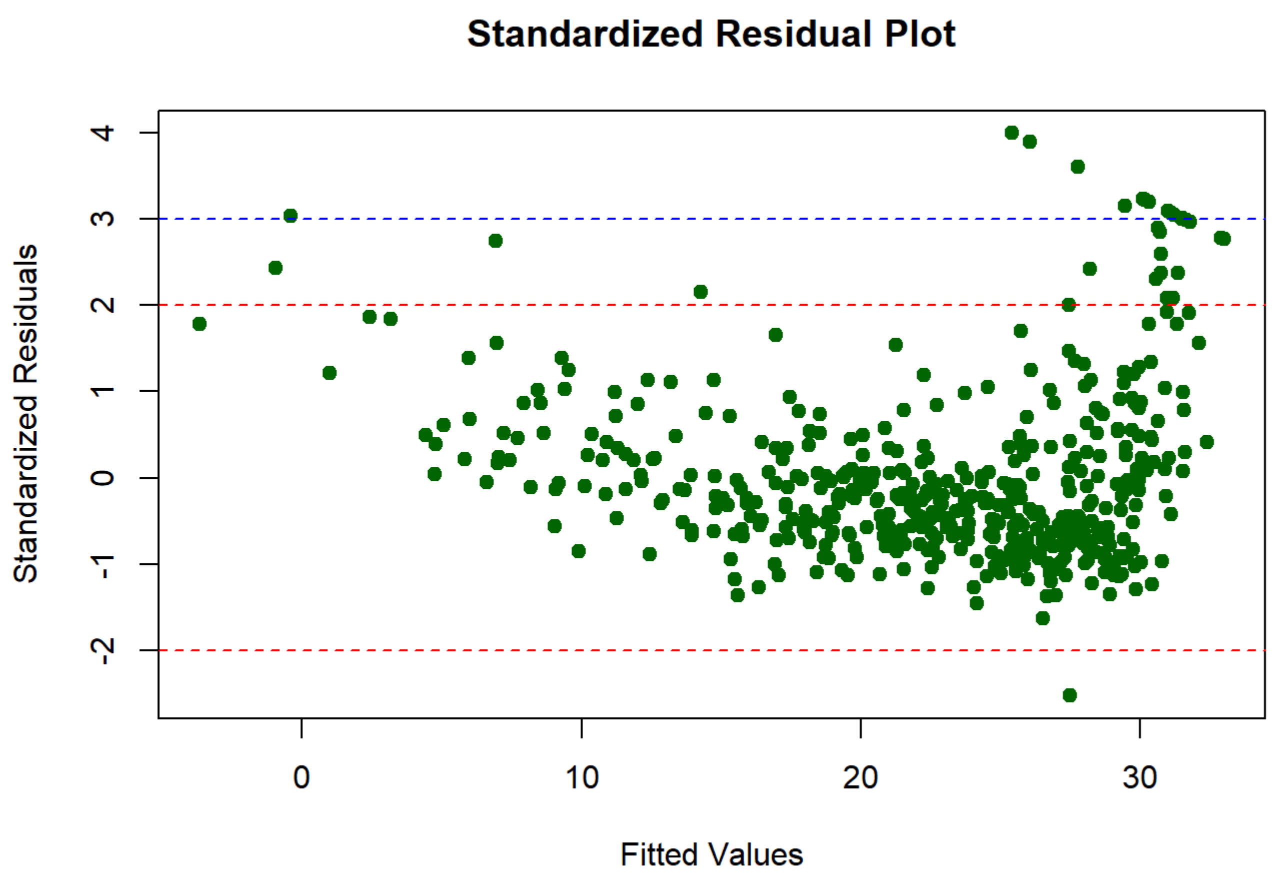
```
library(MASS)
data(Boston)

model_boston <- lm(medv ~ crim + nox + black + lstat, data = Boston)

summary(model_boston)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + black + lstat, data = Boston)
##
## 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
```

```
plot(model_boston$fitted.values, resid(model_boston),
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residual Plot",
     pch = 19, col = "blue")

abline(h = 0, col = "red", lwd = 2)
```



```
std_res <- rstandard(model_boston)
plot(model_boston$fitted.values, std_res,
     xlab = "Fitted Values",
     ylab = "Standardized Residuals",
     main = "Standardized Residual Plot",
     pch = 19, col = "darkgreen")

abline(h = c(-2, 2), col = "red", lty = 2)
abline(h = c(-3, 3), col = "blue", lty = 2)
```

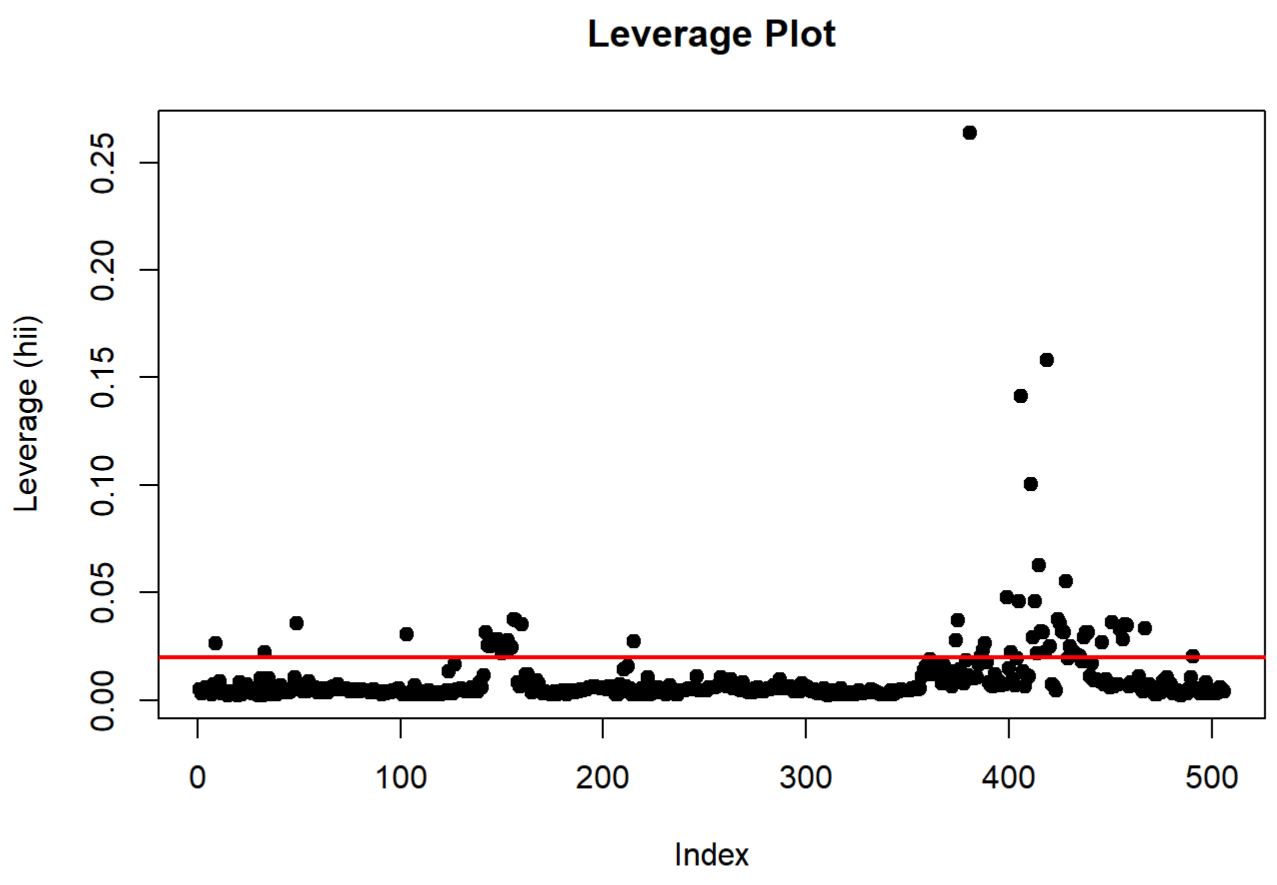


```
which(abs(std_res) > 2)  # Observations that may be outliers
```

```
##  99 162 163 164 167 187 196 204 205 215 225 226 229 234 257 258 262 263 268 281
##  99 162 163 164 167 187 196 204 205 215 225 226 229 234 257 258 262 263 268 281
## 283 284 369 370 371 372 373 375 410 413 506
## 283 284 369 370 371 372 373 375 410 413 506
```

Detect Leverage points

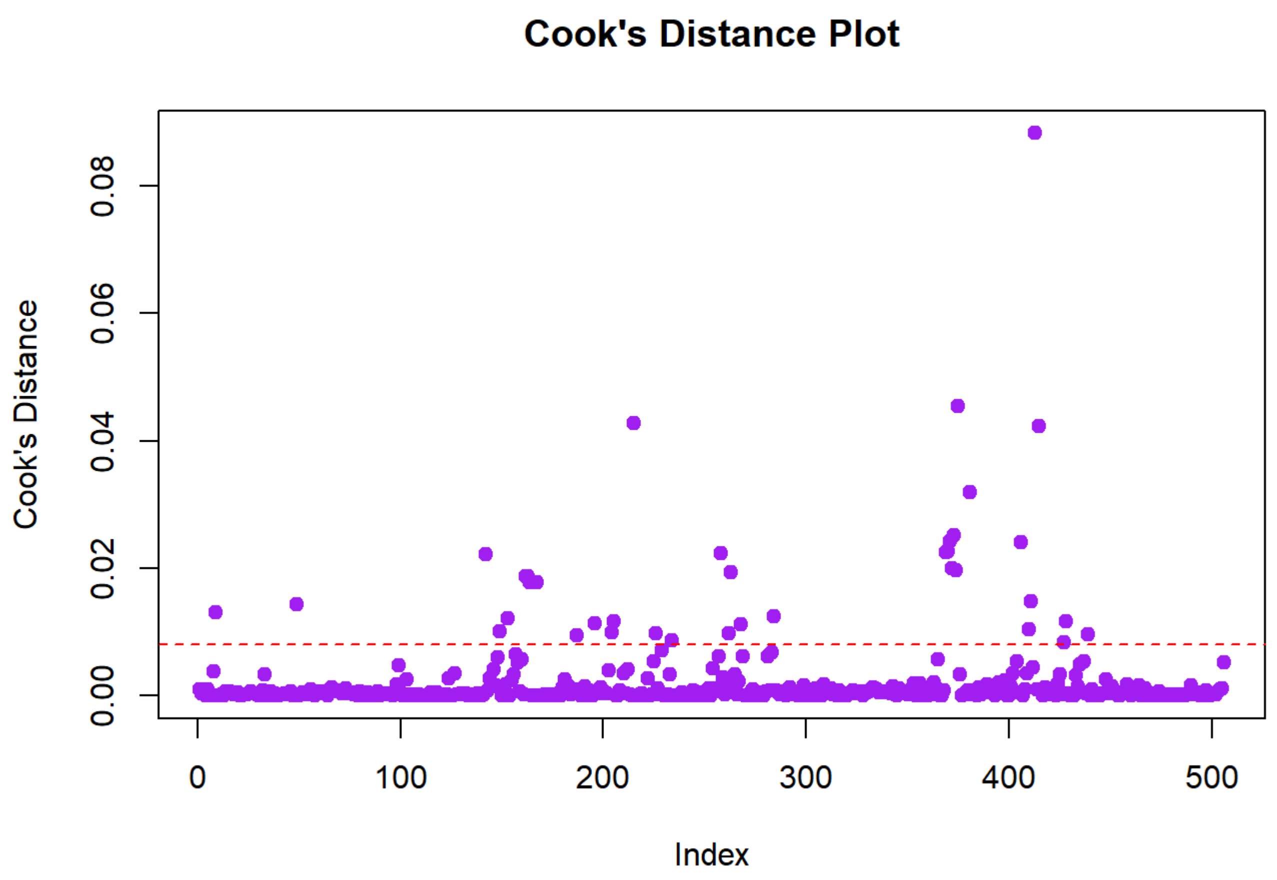
```
hii = hatvalues(model_boston)

plot(hii,
     ylab = "Leverage (hii)",
     main = "Leverage Plot",
     pch = 19)

n = nrow(Boston)
p = length(coef(model_boston)) - 1
cutoff = 2 * (p + 1) / n
abline(h = cutoff, col = "red", lwd = 2)
```



```
which(hii > cutoff)
```

```
##   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
```

```
cooksd = cooks.distance(model_boston)

plot(cooksd,
     ylab = "Cook's Distance",
     main = "Cook's Distance Plot",
     pch = 19, col = "purple")

abline(h = 4/(n - p - 1), col = "red", lty = 2)
```



```
which(cooksd > 4/(n - p - 1))
```

```
##   9  49 142 149 153 162 163 164 167 187 196 204 205 215 226 234 258 262 263 268
```

```
##   9  49 142 149 153 162 163 164 167 187 196 204 205 215 226 234 258 262 263 268
```

```
## 284 369 370 371 372 373 374 375 381 406 410 411 413 415 427 428 439
```

```
## 284 369 370 371 372 373 374 375 381 406 410 411 413 415 427 428 439
```