Assignment_4_710

Protyay Chatterjee

2026-02-19

Problem Set 3: Multiple Linear Regression

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)
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)

## Warning: package 'stargazer' was built under R version 4.5.2

##
## 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

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 ...

veh <- subset(fgl, type == "Veh")
veh

##         RI    Na   Mg   Al    Si    K   Ca   Ba   Fe type
## 147 -0.31 13.65 3.66 1.11 72.77 0.11 8.60 0.00 0.00  Veh
## 148 -1.90 13.33 3.53 1.34 72.67 0.56 8.33 0.00 0.00  Veh
## 149 -1.30 13.24 3.57 1.38 72.70 0.56 8.44 0.00 0.10  Veh
## 150 -1.57 12.16 3.52 1.35 72.89 0.57 8.53 0.00 0.00  Veh
## 151 -1.35 13.14 3.45 1.76 72.48 0.60 8.38 0.00 0.17  Veh
## 152  3.27 14.32 3.90 0.83 71.50 0.00 9.49 0.00 0.00  Veh
## 153 -0.21 13.64 3.65 0.65 73.00 0.06 8.93 0.00 0.00  Veh
## 154 -1.90 13.42 3.40 1.22 72.69 0.59 8.32 0.00 0.00  Veh
## 155 -1.06 12.86 3.58 1.31 72.61 0.61 8.79 0.00 0.00  Veh
## 156 -1.54 13.04 3.40 1.26 73.01 0.52 8.58 0.00 0.00  Veh
## 157 -1.45 13.41 3.39 1.28 72.64 0.52 8.65 0.00 0.00  Veh
## 158  3.21 14.03 3.76 0.58 71.79 0.11 9.65 0.00 0.00  Veh
## 159 -0.24 13.53 3.41 1.52 72.04 0.58 8.79 0.00 0.00  Veh
## 160 -0.04 13.50 3.36 1.63 71.94 0.57 8.81 0.00 0.09  Veh
## 161  0.32 13.33 3.34 1.54 72.14 0.56 8.99 0.00 0.00  Veh
## 162  1.34 13.64 3.54 0.75 72.65 0.16 8.89 0.15 0.24  Veh
## 163  4.11 14.19 3.78 0.91 71.36 0.23 9.14 0.00 0.37  Veh

model <- lm(RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = veh)
summary(model)

##
## 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
```

```r
stargazer(model,type="text",out="data.txt")
```

```
##
## =============================================
##                         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
```

```r
#remove intercept and find smallest p value
pvals=summary(model)$coefficients[-1,4]
pvals
```

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

```r
best_predictor=names(which.min(pvals))
best_predictor
```

```
## [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
# Create formula dynamically
formula_simple = as.formula(paste("RI ~", best_predictor))

# Fit simple linear regression
model_simple = lm(formula_simple, data = veh)

# Summary
summary(model_simple)
```

```
##
## Call:
## lm(formula = formula_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

summary(model_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).

```
# Simple linear regression (already fitted earlier, but refitting for
clarity)
model_simple = lm(RI ~ Fe, data = veh)

summary(model_simple)

##
## 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
```

```r
# Store performance metrics
r2_simple = summary(model_simple)$r.squared
adjr2_simple = summary(model_simple)$adj.r.squared

# Quadratic regression model
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
```

```r
# Store performance metrics
r2_quad = summary(model_quad)$r.squared
adjr2_quad = summary(model_quad)$adj.r.squared

# Compare R-squared
r2_simple
```

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

```r
r2_quad
```

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

```r
# Compare Adjusted R-squared
adjr2_simple
```

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

```r
adjr2_quad
```

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

The quadratic model improves the regression if:

R² (quadratic) > R² (simple)

Adjusted R² 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.

```r
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.5.2

library(car)

## Warning: package 'car' was built under R version 4.5.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.5.2

data(Credit)

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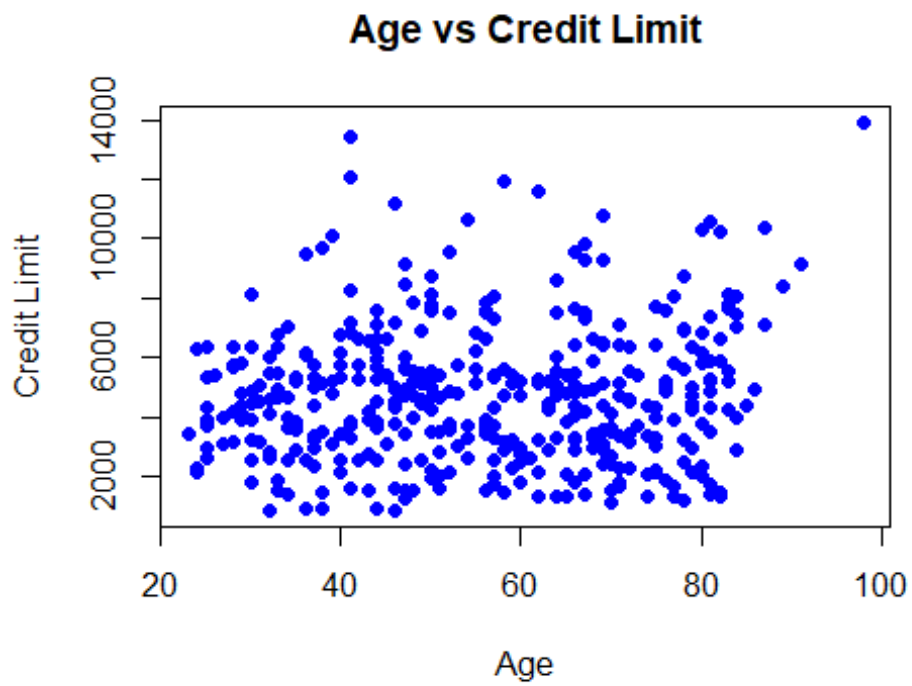
Age vs Credit Limit


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

Rating vs Credit Limit

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 = 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
```

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.

```r
# Load library
library(MASS)

# Load Boston dataset
data(Boston)

# Fit multiple linear regression
model_boston <- lm(medv ~ crim + nox + black + lstat, data = Boston)

# Summary of the model
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)
```
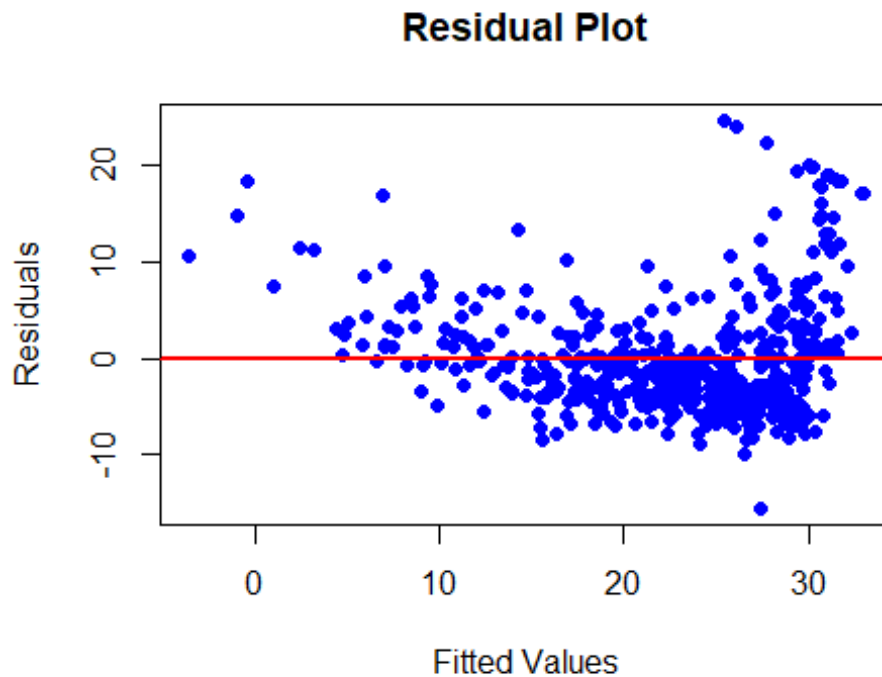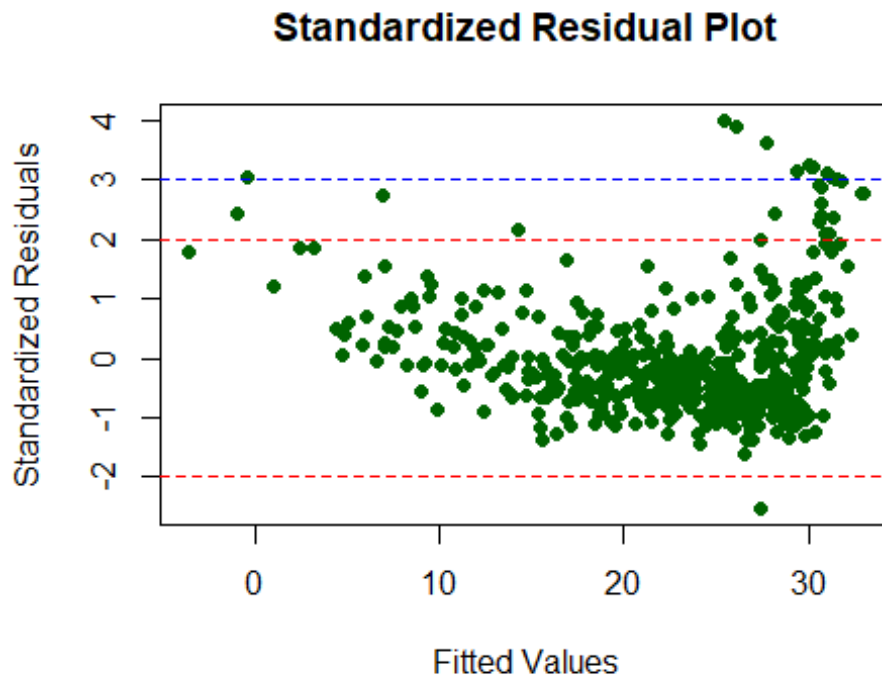
Residual Plot


```r
# Standardized residuals
std_res <- rstandard(model_boston)

# Plot standardized residuals
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)
```

Standardized Residual Plot


```r
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

```r
# Leverage (hat values)
hii = hatvalues(model_boston)

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

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

Leverage Plot
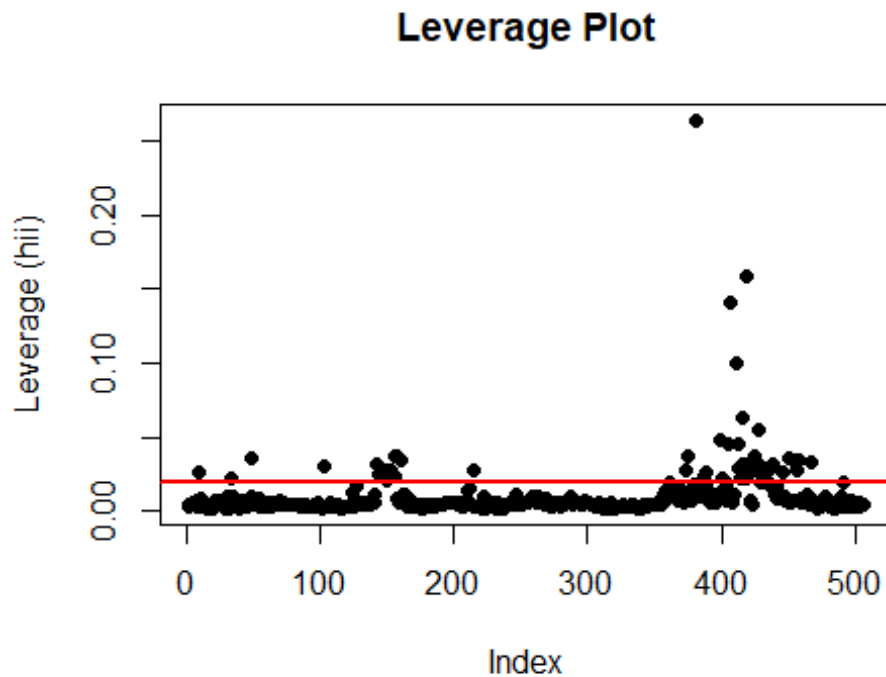

```
# Observations with high leverage
which(hii > cutoff)

##    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

# Cook's distance
cooksd = cooks.distance(model_boston)

# Plot Cook's distance
plot(cooksd,
     ylab = "Cook's Distance",
     main = "Cook's Distance Plot",
     pch = 19, col = "purple")
```
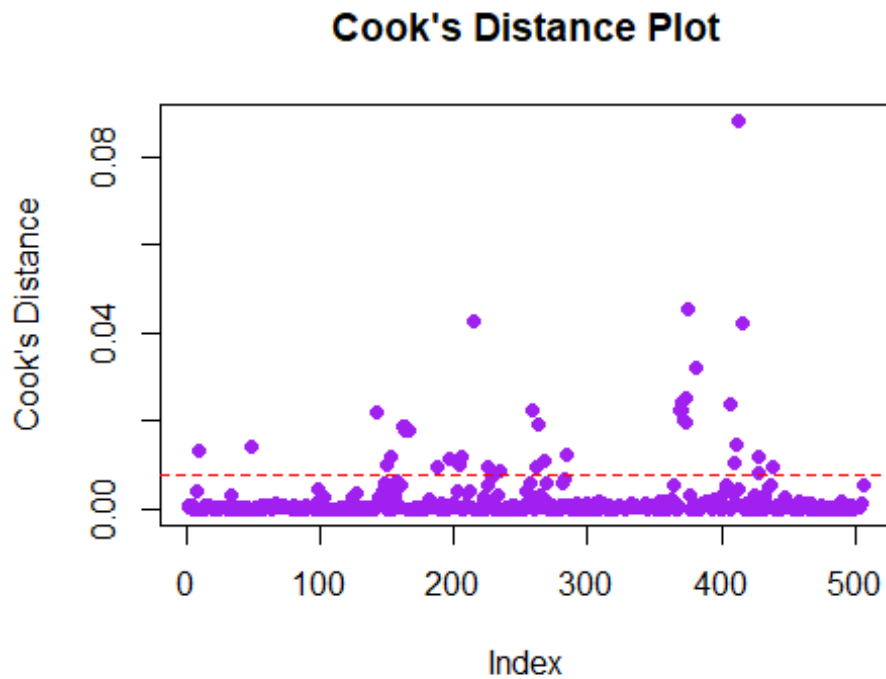
```
abline(h = 4/(n - p - 1), col = "red", lty = 2)
```

Cook's Distance Plot


```
# Identify influential points
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
```