

# Predictive\_PS3(q5) & PS4\_735

Srinjana Sen\_735

2026-02-19

## PREDICTIVE ANALYTICS

### Problem Set 3: Multiple Linear Regression

Q5 Problem to demonstrate the utility of non-linear regression over linear regression

Get the fgl data set from “MASS” library.

```
library(MASS)
data("fgl")

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

- (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.

```
full_model=lm(RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = veh_data)
```

```
summary(full_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

pval=summary(full_model)$coefficients[-1,4]
pval

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

```

cat("So the metallic oxide with the strongest relationship with RI is the one
with minimum p-value i.e. ",names(which.min(pval)))

## So the metallic oxide with the strongest relationship with RI is the one
with minimum p-value i.e. Fe

```

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

```

slr_model=lm(RI ~ Fe, data = veh_data)
summary(slr_model)

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

```

Thus, the fitted model is:

$$RI = -0.5007 + 8.1362 * Fe$$

(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).

```

quad_model = lm(RI ~ Fe + I(Fe^2), data = veh_data)
summary(quad_model)

##
## Call:
## lm(formula = RI ~ Fe + I(Fe^2), data = veh_data)
##
## 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) -0.5007    0.4861  -1.030   0.3193    
## Fe          8.1362    4.0780   1.995   0.0645 .  
## I(Fe^2)     0.0000    0.0000   0.0000  0.0000    
## ---        
## 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

```

```

## (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

```

By considering the quadratic term  $I(Fe^2)$  in the regression model, we obtain the non-linear relationships between Fe and RI, if any. The quadratic term is significant as the adjusted R-squared increases compared to the linear model, thus providing a better explanation of the relationship between Fe and RI.

## Problem Set 4: Some Potential Problems in Multiple Linear Regression

### Q1 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.

```

library(ISLR)

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

data(Credit)
cred_data<-Credit[,c("Balance", "Age", "Limit", "Rating")]

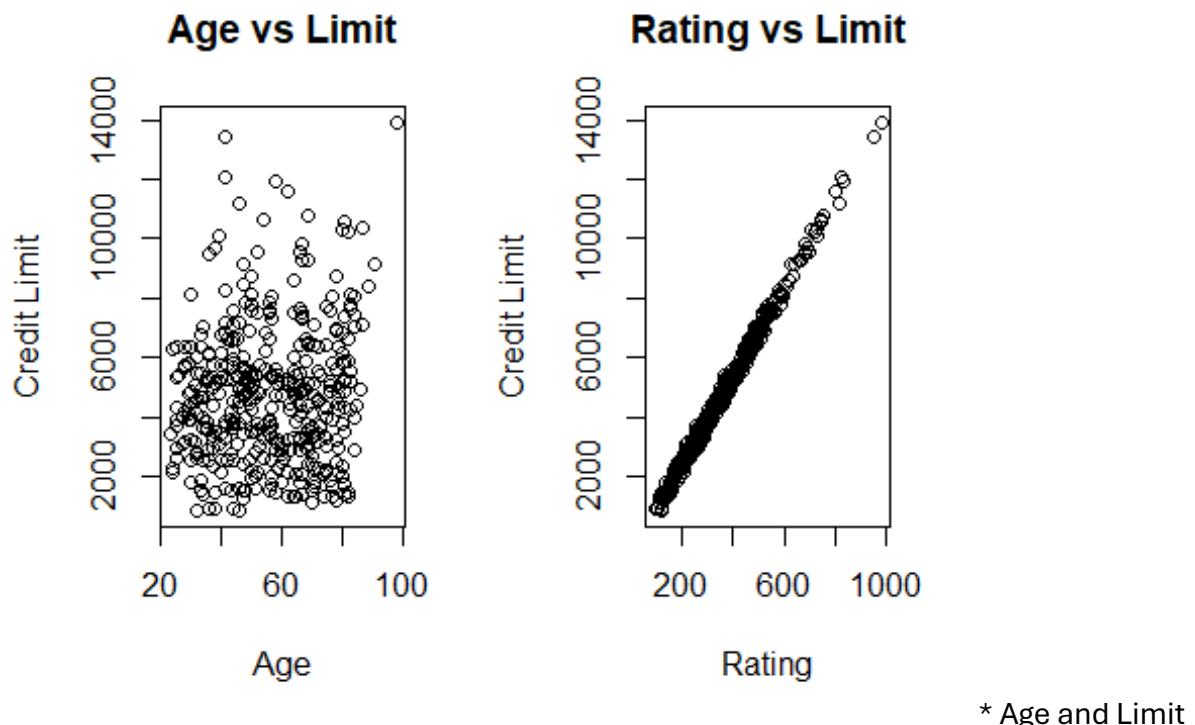
```

- (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(cred_data$Age, cred_data$Limit,
      xlab = "Age",
      ylab = "Credit Limit",
      main = "Age vs Limit")
plot(cred_data$Rating, cred_data$Limit,
      xlab = "Rating",
      ylab = "Credit Limit",
      main = "Rating vs Limit")

```



\* Age and Limit

exhibit almost no relationship

\* Rating and Limit on the other hand exhibit very strong positive linear relationship  
This suggests Limit and Rating are highly correlated, leading to multicollinearity.

(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?

```
m1=lm(Balance ~ Age + Limit, data = Credit)
m2=lm(Balance ~ Age + Rating + Limit, data = Credit)
m3=lm(Balance ~ Rating + Limit, data = Credit)
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

stargazer(m1, m2, m3,
          type = "text",
          title = "Regression Models for Balance",
```

```

  column.labels = c("Age+Limit", "Age+Rating+Limit", "Rating+Limit"),
  dep.var.labels = "Balance")

## 
## Regression Models for Balance
## =====
##                                     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                        0.019
## 0.025                         (0.063)  

##                               (0.064)
## 
## Constant                     -259.518***  

## -377.537***                  (55.882)  

##                               (45.254)
## 
## -----
## Observations                 400          400
## 400
## R2                          0.754
## 0.746
## Adjusted R2                  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
```

Limit doesn't have significant effect on models 2 and 3.

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

```
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

vif(m2)

##          Age      Rating      Limit
## 1.011385 160.668301 160.592880

vif(m1)

##          Age      Limit
## 1.010283 1.010283
```

Clearly, the high VIF-s of Rating and Limit imply multicollinearity. Thus we check the VIF of model 1 where Rating is dropped. The VIF-s signify lack of multicollinearity.

## Q2 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)
bos_data=(Boston[,c("medv","crim","nox","black","lstat")])
model=lm(medv ~ crim + nox + black + lstat, data = bos_data)
summary(model)

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

```

## 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

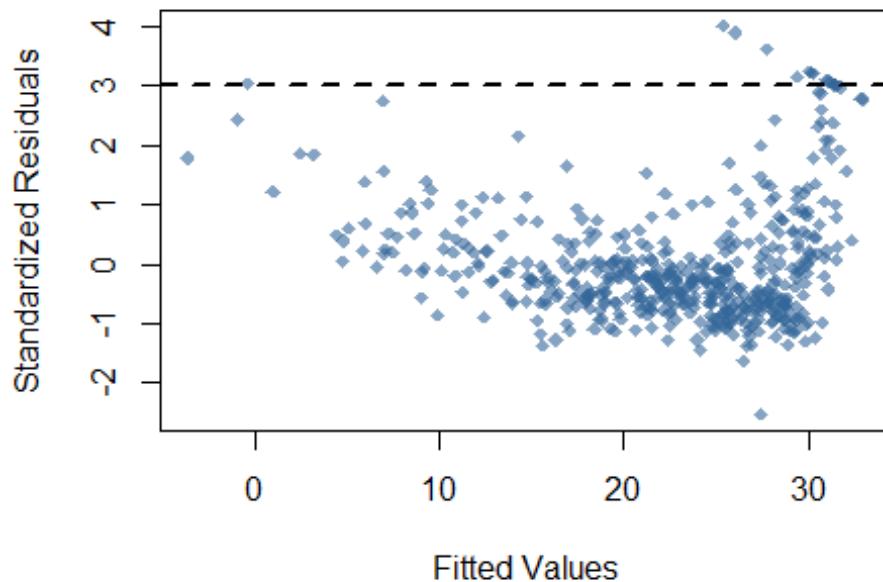
std_res = rstandard(model)
fit_val = fitted(model)

plot(fit_val, std_res,
      xlab = "Fitted Values",
      ylab = "Standardized Residuals",
      main = "Residual Plot for Outlier Detection",
      pch = 18, col = rgb(0.2, 0.4, 0.6))

abline(h = c(-3, 3), col = "black", lty = 2, lwd = 2)

```

## Residual Plot for Outlier Detection



```
outliers=which(abs(std_res) > 3)
cat("Observations identified as outliers:\n")

## Observations identified as outliers:

print(outliers)

## 167 187 196 205 226 258 263 268 284 369 370 372 373 413
## 167 187 196 205 226 258 263 268 284 369 370 372 373 413
```

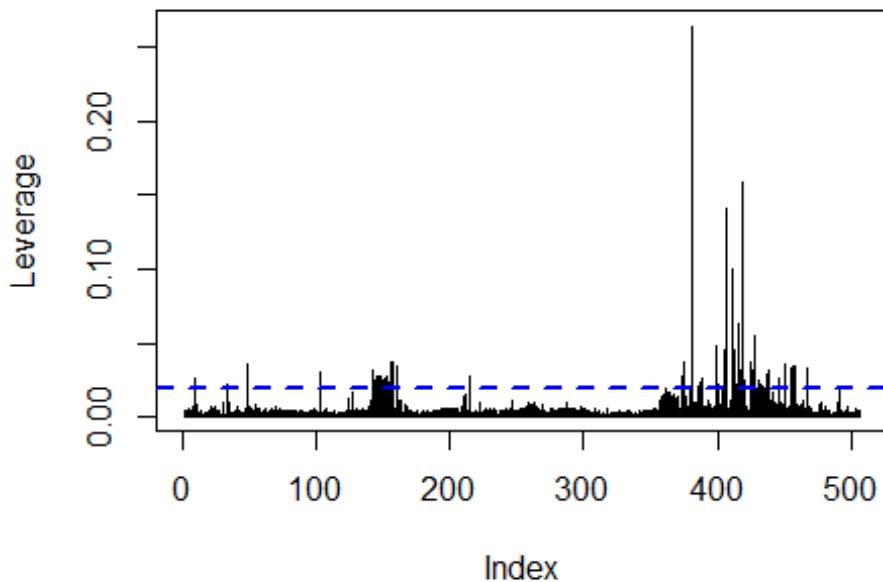
By looking at the graph, any points falling above the dashed line correspond exactly to the indices and are identified as outliers.

```
p = 4
n = nrow(Boston)
leverage_cutoff = 2 * (p + 1) / n

lev = hatvalues(model)

plot(lev, type = "h",
      main = "Leverage Values by Observation",
      ylab = "Leverage", xlab = "Index")
abline(h = leverage_cutoff, col = "blue", lty = 2, lwd = 2)
```

## Leverage Values by Observation



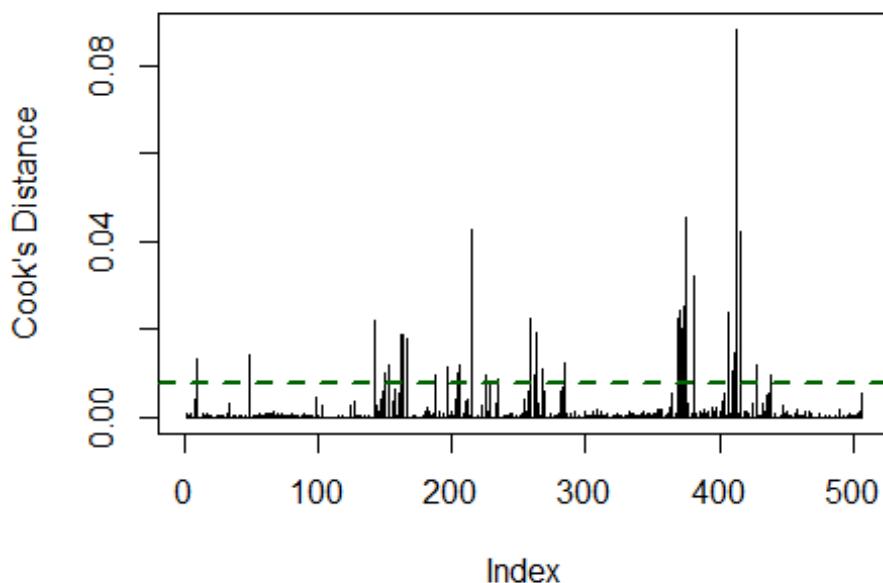
```
high_lev = which(lev > leverage_cutoff)
cat("Number of high leverage points detected:", length(high_lev), "\n")
## Number of high leverage points detected: 64
```

By looking at the graph, any observation falling above the dashed line correspond exactly to the indices and are identified as leverage points.

```
cooks_d = cooks.distance(model)
cooks_cutoff = 4 / n

plot(cooks_d, type = "h",
      main = "Cook's Distance by Observation",
      ylab = "Cook's Distance", xlab = "Index")
abline(h = cooks_cutoff, col = "darkgreen", lty = 2, lwd = 2)
```

## Cook's Distance by Observation



```
influential_points=which(cooks_d > cooks_cutoff)
cat("Number of influential points detected:", length(influential_points),
"\n")
## Number of influential points detected: 37
```

By looking at the graph, any points falling above the dashed line correspond exactly to the indices and are identified as influential points.