PREDICTIVE ANALYSIS ASSIGNMEMT 4

SHRESTHA BAJAJ

2026-02-19

Problem Sheet 3

QUESTION 5

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

Get the fgl data set from “MASS” library.

```
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 ...
```

(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.

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

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

```
##
## Call:
## lm(formula = RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = veh_data)
##
## 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-squared = 0.9906

The model with all the parameter explains around 99% of variability in the response variable.
Therefore the model is a good fit.
Of all the parameters, Fe is the most significant as it has lowest p-value and highest t-value. Fe explains the most variability in RI of “Vehicle Window glass”.

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

```
simple_model = lm(RI ~ Fe, data = veh_data)
summary(simple_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
```

R-Squared = 0.2097

It means that the simple model based on predictor Fe describes only 21% (approximately) of variability in the response variable, which is significantly less than the model with all the predictors. It is because by only considering one predictor, the model looses its explanatory power.

(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)

```
quadratic_model = lm(RI ~ Fe + I(Fe^2), data = veh_data)
summary(quadratic_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)  -0.2785     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
```

The model can be improved by taking a quadratic element of predictor Fe.
The R-square of the model increases to 0.3554 from 0.2097 meaning that the updated model explains around 35% of the variability in the response variable.

Problem Sheet 4

QUESTION 1.

Problem to demonstrate multicollinearity

Consider the Credit data in the ISLR library.

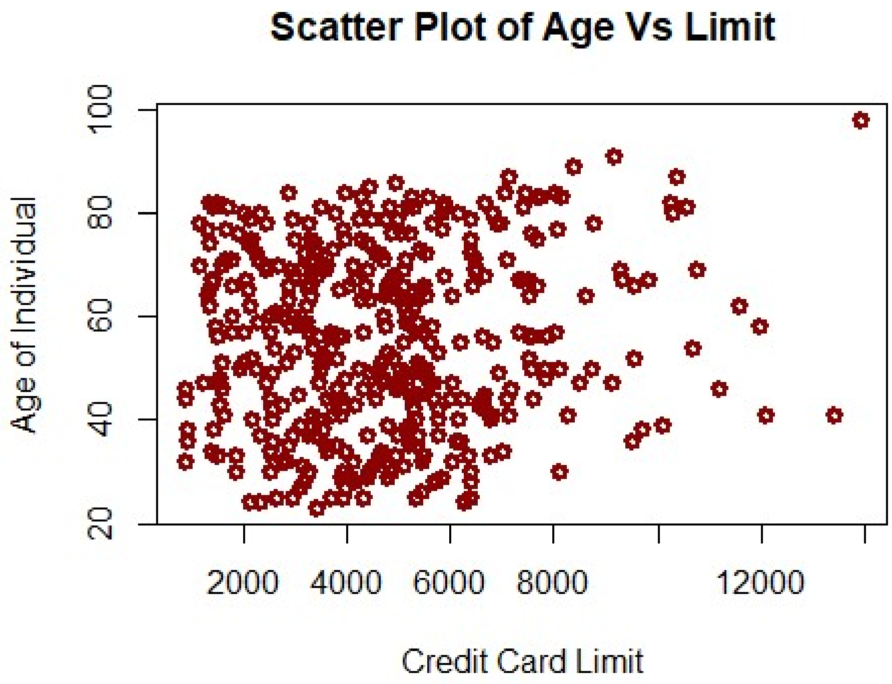
```
library(ISLR)
data(Credit)
str(Credit)
```

```
## 'data.frame':    400 obs. of  12 variables:
##  $ ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Income   : num  14.9 106 104.6 148.9 55.9 ...
##  $ Limit    : int  3606 6645 7075 9504 4897 8047 3388 7114 3300 6819 ...
##  $ Rating   : int  283 483 514 681 357 569 259 512 266 491 ...
##  $ Cards    : int  2 3 4 3 2 4 2 2 5 3 ...
##  $ Age      : int  34 82 71 36 68 77 37 87 66 41 ...
##  $ Education: int  11 15 11 11 16 10 12 9 13 19 ...
##  $ Gender   : Factor w/ 2 levels " Male","Female": 1 2 1 2 1 1 2 1 2 2 ...
##  $ Student  : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...
##  $ Married  : Factor w/ 2 levels "No","Yes": 2 2 1 1 2 1 1 1 1 2 ...
##  $ Ethnicity: Factor w/ 3 levels "African American",..: 3 2 2 2 3 3 1 2 3
1 ...
##  $ Balance  : int  333 903 580 964 331 1151 203 872 279 1350 ...
```

Choose balance as the response and 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.

```
attach(Credit)
plot(Limit, Age, xlab = "Credit Card Limit", ylab = "Age of Individual", main
= "Scatter Plot of Age Vs Limit", col = "darkred", lwd = 3)
```



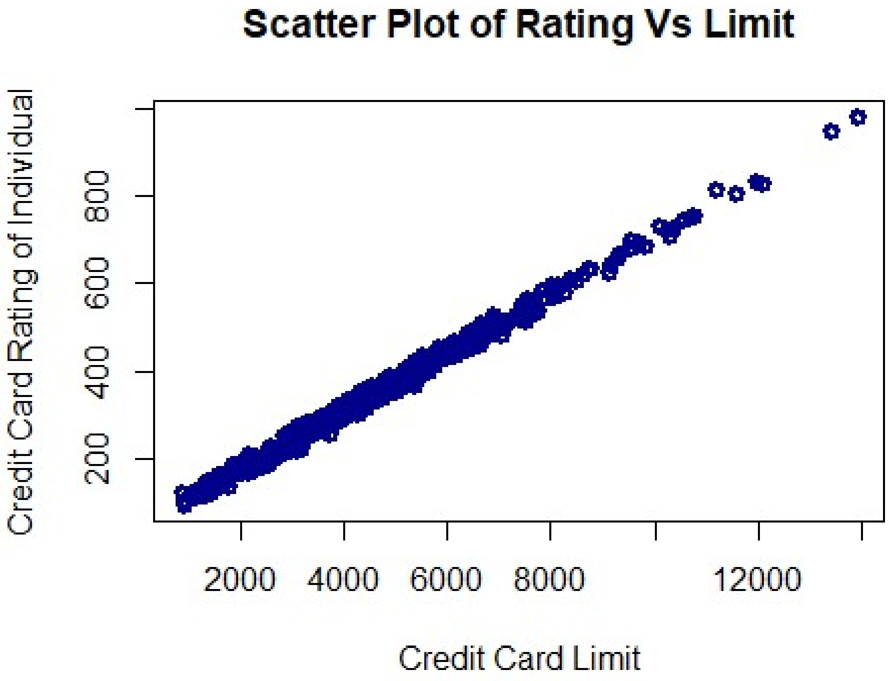
Interpretation:

- Age and Limit have appears to have no clear linear relationship.
- The points are randomly scattered without any visible upward or downward trend.
- Younger and older population both hold varying credit card limits.
- The points are clustered around 2000 - 6000 limit implying most of the population gas moderate limit.
- One observation around age ≈ 98 with high limit ≈ 14,000 but no extreme leverage patterns are evident.

```
attach(Credit)

## The following objects are masked from Credit (pos = 3):
##
##     Age, Balance, Cards, Education, Ethnicity, Gender, ID, Income,
##     Limit, Married, Rating, Student

plot(Limit, Rating, xlab = "Credit Card Limit", ylab = "Credit Card Rating of
Individual", main = "Scatter Plot of Rating Vs Limit", col = "darkblue", lwd
= 3)
```



Interpretation:

- Rating and Limit have a strong positive linear relationship.
- As Rating increases, Limit increases too.
- Very small vertical dispersion.
- A few high-limit observations (≈ 13,000–14,000) but no concerning influential outliers are evident.
- Rating and Limit appears to be highly correlated. Therefore, concern of multicollinearity exists.

(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?

```
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
```

```
age_and_limit_model = lm(Balance ~ Age + Limit, data = Credit)
age_rating_and_limit_model = lm(Balance ~ Age + Limit + Rating, data =
Credit)
rating_and_limit_model = lm(Balance ~ Limit + Rating, data = Credit)

stargazer(age_and_limit_model,age_rating_and_limit_model,rating_and_limit_mod
el,
          type = "text", dep.var.labels = "Credit Card Balance",
          column.labels = c("Age and Limit Model", "Age, Limit and Rating
Model", "Limit and Rating Model"))
```

```
## 
## 
=============================================================================
====================
##                                                             Dependent variable:
##                        -----------------------------------------------------
-----------------------
##                                                             Credit Card Balance
##                          Age and Limit Model    Age, Limit and Rating Model
Limit and Rating Model
##                                  (1)                         (2)
(3)
## ---------------------------------------------------------------------------
-----------------------
## Age                           -2.291***                   -2.346***
##                                (0.672)                     (0.669)
## 
## Limit                          0.173***                     0.019
0.025
##                                (0.005)                     (0.063)
(0.064)
## 
## Rating                                                     2.310**
2.202**
##                                                            (0.940)
(0.952)
## 
## 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
```

Interpretation:

- In the first 2 models, Age is highly significant at 1% level implying Age has a statistically strong effect on balance. Age has negative impact on Balance since the coefficients of Age in both the models is negative.
- In the first model, Limit was highly significant at 1% level but in second and third model where a new predictor Rating was introduced, Limit became statistically insignificant. This suggests, there exists multicollinearity between Rating and Limit.
  When both enter the model, Rating absorbs most explanatory power. Rating remains important even when Limit is included.This means Rating is a stronger determinant of balance than Limit.

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

```
library(car)

## Loading required package: carData

vif(age_and_limit_model)

##      Age    Limit
## 1.010283 1.010283

vif(age_rating_and_limit_model)

##        Age      Limit     Rating
##   1.011385 160.592880 160.668301

vif(rating_and_limit_model)

##    Limit   Rating
## 160.4933 160.4933
```

The Variance Inflation Factor (VIF) results clearly reveal the presence and severity of multicollinearity across the models. In the Age and Limit model, both Age and Limit have VIF values close to 1 (≈1.01), indicating virtually no multicollinearity meaning the predictors are almost completely independent of each other, and their coefficient estimates are stable and reliable. However, once Rating is introduced in the Age, Limit and Rating model, the VIF values for Limit (≈160.59) and Rating (≈160.67) increase dramatically, while Age remains near 1 (≈1.01). Similarly, in the Limit and Rating model, both variables again show extremely high VIF values (≈160.49). A VIF exceeding 10 is typically considered serious; values above 100 indicate extreme multicollinearity. This means Limit and Rating are almost perfectly linearly correlated, causing inflated standard errors and making individual coefficient estimates unstable and statistically unreliable. This explains why Limit becomes insignificant when Rating is added in the model. In contrast, Age remains unaffected because it is not strongly correlated with the other predictors. Overall, the output provides strong statistical evidence of severe multicollinearity between Limit and Rating, confirming the pattern observed in the scatter plot and regression table.

Question 2

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

Attach “Boston” data from MASS library in R.

```
library(MASS)
data(Boston)
attach(Boston)
str(Boston)

## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524
0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

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.

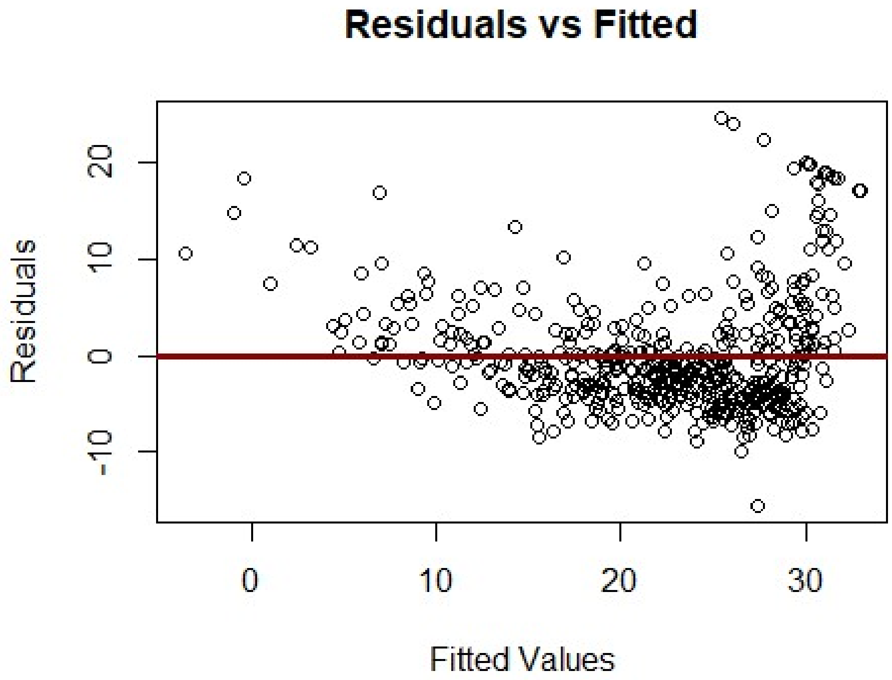
```
model = lm(medv ~ crim + nox + black + lstat, data = Boston)
summary(model)

##
## 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
```

With reference to this problem, detect outliers, leverage points and influential points if any.

```
plot(model$fitted.values, resid(model),xlab="Fitted Values",ylab="Residuals",
     main="Residuals vs Fitted")
abline(h=0,col="darkred",lwd=3)
```



From the residual plot, we can confirm the presence of extreme outliers on both sides. Though, we cannot comment on presence of potential leverage points and influential points from this graph alone.

Finding Outliers -

```
std_res = rstandard(model)
outliers = which(abs(std_res)>2)

cat("The total number of potential outliers present in the model is =",
length(outliers), "\n")

## The total number of potential outliers present in the model is = 31

cat("The outliers present are \n",outliers)

## The outliers present are
##  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
```

Finding Leverage Points -

```
diag = hatvalues(model)

n = nrow(Boston)
p = 4

#Calculating the leverage values
cutoff=3*(p+1)/n
cutoff

## [1] 0.02964427

# High leverage observations
leverage_points = which(diag>cutoff)
leverage_points

##  49 103 142 156 157 160 375 381 399 405 406 411 413 415 416 417 419 424
425 426
##  49 103 142 156 157 160 375 381 399 405 406 411 413 415 416 417 419 424
425 426
## 427 428 438 439 451 455 457 458 467
## 427 428 438 439 451 455 457 458 467

length(leverage_points)

## [1] 29
```

The model has 29 such values for which the value of predictors is very high (leverage). This 29 values may influence the model by having the potential to substantially alter coefficient estimates due to their extreme predictor values. However, they only become problematic when combined with large residuals, making them influential.

Finding Influential Points -

```
cooks_distance=cooks.distance(model)

influential_points = which(cooks_distance>1)
length(influential_points)

## [1] 0
```

We use Cook’s distance D_i which is a function of standardized residuals and elements of hat matrix to find potential influential points.

A data point is said to a influential point if D_i > 1.

Hence, in our model there are no influential points present.