ASSIGNMENT 4

SIMLIN SAHA

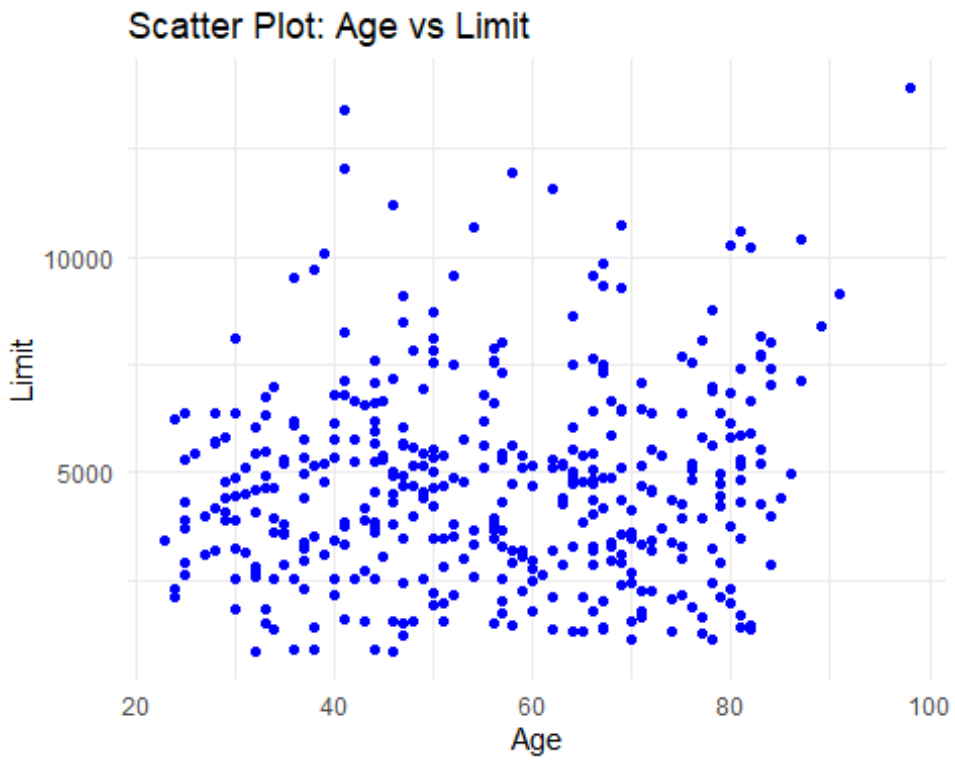
2026-02-19

1. Problem to demonstrate multicollinearity

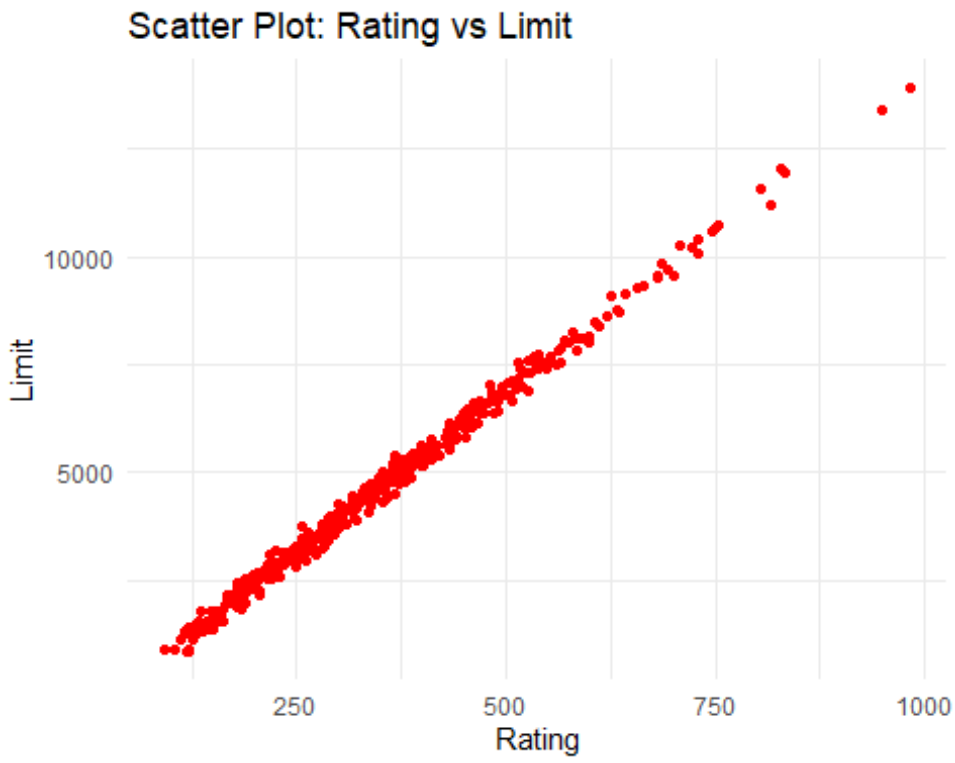
Consider the Credit data in the ISLR library. Choose balance as the response and Age, Limit and Rating as the predictors.

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

```
# Load libraries
library(ISLR)

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

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.5.2

# Load data
data(Credit)

# Scatter Plot 1: Age vs Limit
ggplot(Credit, aes(x = Age, y = Limit)) +
  geom_point(color = "blue") +
  theme_minimal() +
  labs(title = "Scatter Plot: Age vs Limit")
```




```r
# Scatter Plot 2: Rating vs Limit
ggplot(Credit, aes(x = Rating, y = Limit)) +
  geom_point(color = "red") +
  theme_minimal() +
  labs(title = "Scatter Plot: Rating vs Limit")
```



Comments on Scatter Plots

(i) Age vs Limit

The plot shows no strong linear relationship.

Data points are widely scattered.

This indicates weak or negligible correlation between Age and Limit.

(ii) Rating vs Limit

The plot shows a very strong positive linear relationship.

Points lie almost perfectly on a straight line.

This indicates extreme correlation, suggesting serious multicollinearity risk.

(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 differ- ence that you can observe from the output?

```
# Load 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

# Regression models
model1 <- lm(Balance ~ Age + Limit, data = Credit)
model2 <- lm(Balance ~ Age + Rating + Limit, data = Credit)
model3 <- lm(Balance ~ Rating + Limit, data = Credit)

# Regression output table
stargazer(model1, model2, model3, type = "text",
          title = "Regression Results",
          dep.var.labels = "Balance",
          covariate.labels = c("Age", "Limit", "Rating"))

##
## Regression Results
##
================================================================================
================
##                                                 Dependent variable:
##                     ---------------------------------------------------------
-------------------
##                                                        Balance
##                                (1)                       (2)
(3)
## --------------------------------------------------------------------------
-------------------
## Age                          -2.291***                -2.346***
##                               (0.672)                  (0.669)
##
## Limit                                                  2.310**
2.202**
##                                                        (0.940)
(0.952)
##
## Rating                        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
```

The disappearance of statistical significance for Rating after incorporating Limit suggests a high degree of multicollinearity between the two predictors.

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

```
# Load car library
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

# Calculate VIF
vif(model1)

##      Age    Limit
## 1.010283 1.010283

vif(model2)

##        Age     Rating      Limit
##   1.011385 160.668301 160.592880

vif(model3)

##   Rating    Limit
## 160.4933 160.4933
```

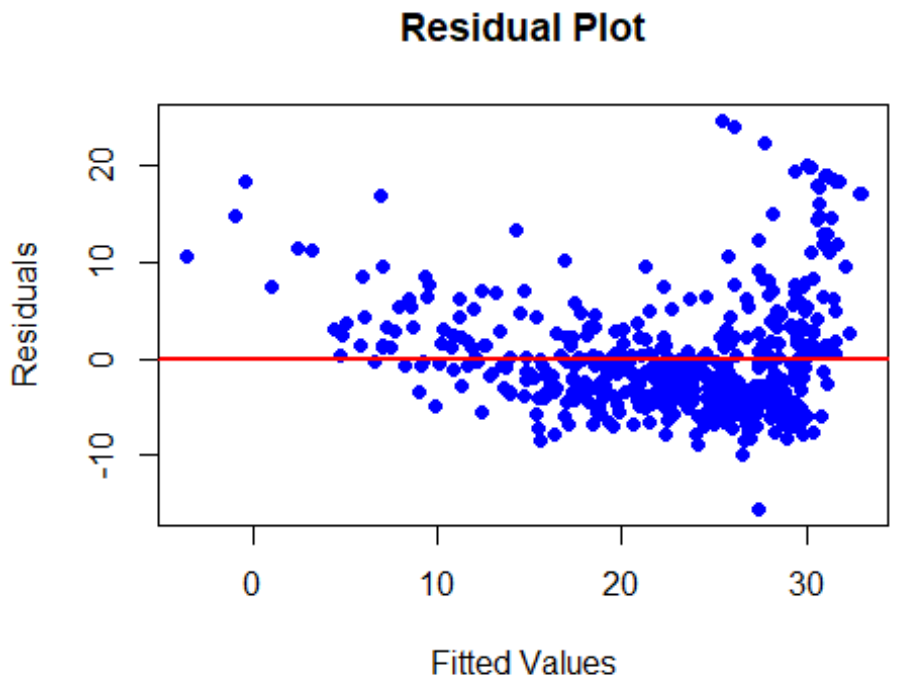
Age and Limit exhibit negligible multicollinearity, as indicated by VIF values close to 1. In contrast, Rating and Limit display severe multicollinearity, with VIF values around 160. This high degree of collinearity explains the loss of statistical significance of Rating in the multiple regression models.

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

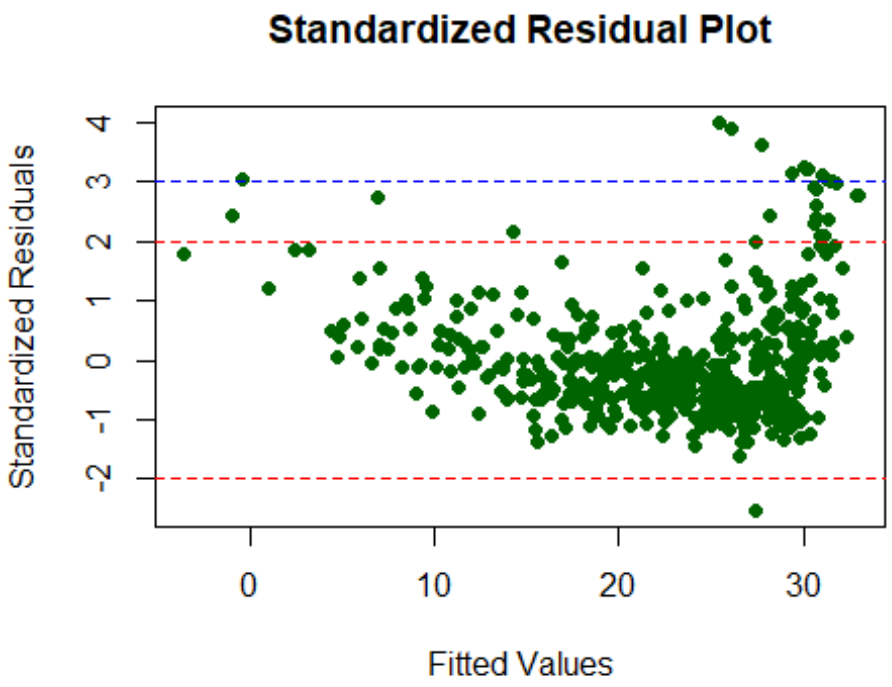
Attach “Boston” data from MASS library in R. Select median value of owner occupied homes, as the response and per capita crime rate, nitrogen oxides concentration, proportion of blacks and percentage of lower status of the population as predictors. The objective is to fit a multiple linear regression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

```
# Load library
library(MASS)

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

# 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)
```




```
# 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)
```

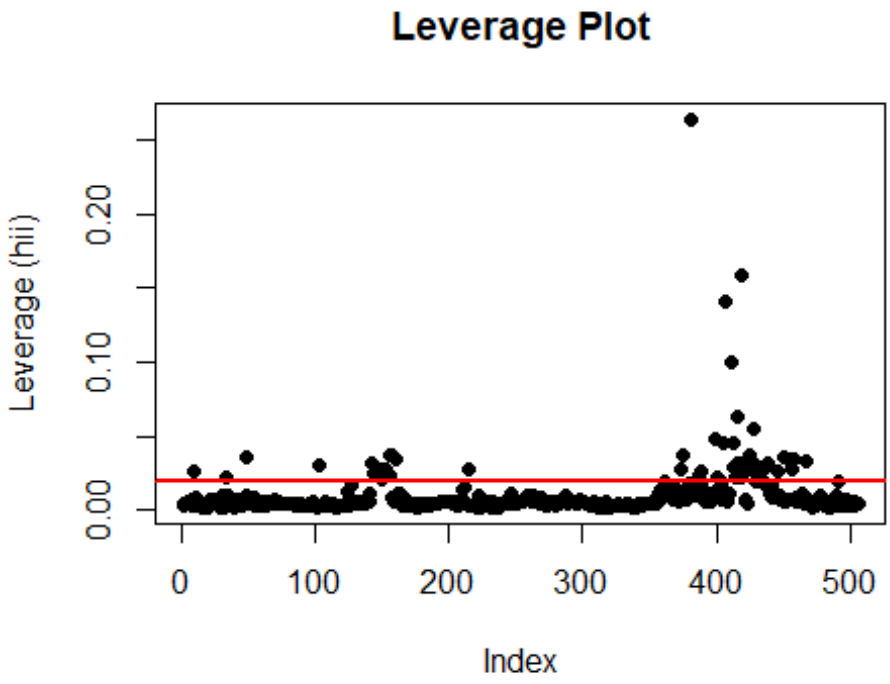



```
which(abs(std_res) > 2)  # Observations that may be outliers
```

```
##  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

```
# 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)
```

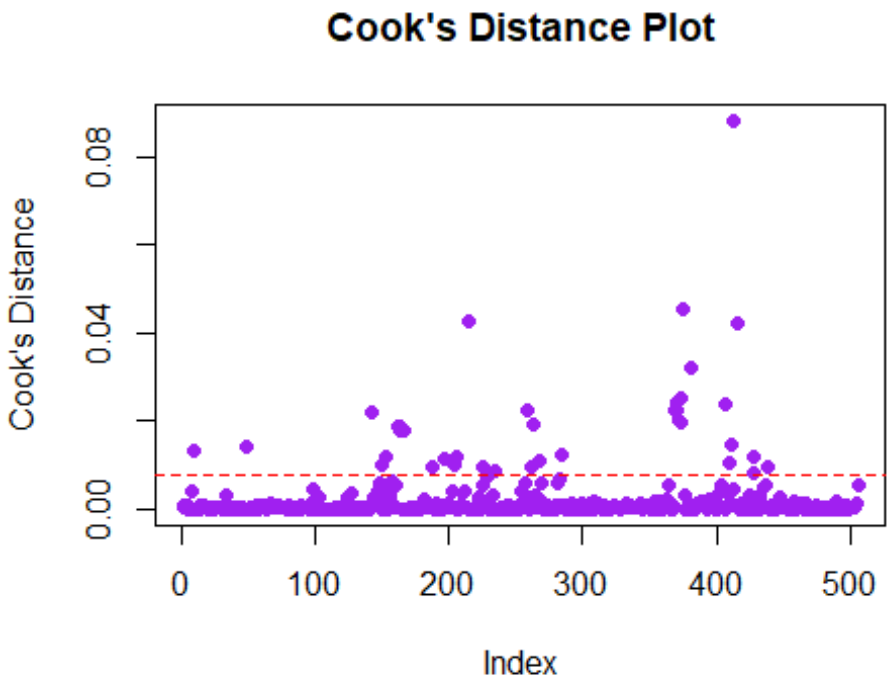



```r
# Observations with high leverage
which(hii > cutoff)
```

```
##   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
```

```r
# 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")
```

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




```r
# 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
```

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

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

(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).

```
library(MASS)
data(fgl)

# Select only Vehicle Window Glass
vw <- subset(fgl, type == "Veh")

str(vw)
```

```
## 'data.frame':    17 obs. of  10 variables:
##  $ RI  : num  -0.31 -1.9 -1.3 -1.57 -1.35 ...
##  $ Na  : num  13.6 13.3 13.2 12.2 13.1 ...
##  $ Mg  : num  3.66 3.53 3.57 3.52 3.45 3.9 3.65 3.4 3.58 3.4 ...
##  $ Al  : num  1.11 1.34 1.38 1.35 1.76 0.83 0.65 1.22 1.31 1.26 ...
##  $ Si  : num  72.8 72.7 72.7 72.9 72.5 ...
##  $ K   : num  0.11 0.56 0.56 0.57 0.6 0 0.06 0.59 0.61 0.52 ...
##  $ Ca  : num  8.6 8.33 8.44 8.53 8.38 9.49 8.93 8.32 8.79 8.58 ...
##  $ Ba  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Fe  : num  0 0 0.1 0 0.17 0 0 0 0 0 ...
##  $ type: Factor w/ 6 levels "WinF","WinNF",..: 3 3 3 3 3 3 3 3 3 3 ...
```

Variables:

Response: RI

Predictors: Na, Mg, Al, Si, K, Ca, Ba, Fe

```
# Multiple Linear Regression of RI on Metallic Oxides
model1 <- lm(RI ~ Na + Mg + Al + Si + K + Ca + Ba + Fe, data = vw)
summary(model1)
```

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

The metallic oxide with the lowest p-value is:

Iron Oxide (Fe) — p = 0.00362

Hence: Answer (a): Best predictor of RI = Fe (Iron Oxide)

```
# Simple Linear Regression of RI on Fe
model2 <- lm(RI ~ Fe, data = vw)
summary(model2)
```

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

Interpretation:

Fe is statistically significant.

However, the R^2 is moderate, indicating that the linear model does not fully capture the relationship.

```
# Non-Linear Regression (Quadratic Model)
model3 <- lm(RI ~ Fe + I(Fe^2), data = vw)
summary(model3)
```

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

```
##
## 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
```

```
summary(model2)$adj.r.squared
```

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

```
summary(model3)$adj.r.squared
```

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

The simple linear regression of RI on Fe reveals a statistically significant association; however, its explanatory capability remains limited. Incorporating a quadratic term leads to a substantial improvement in model fit, as evidenced by an increase in adjusted R^2 and a reduction in residual error. Consequently, the non-linear regression model offers a more accurate representation of refractive index variation.