CMDA-4654

Project 1

Matthew Hill and Rohan Reddy Illapuram

April 1, 2024

Web Scraping Data and Cleaning Data

```
link = "https://kenpom.com/"
page = read_html(link)
table = page %>% html_nodes("table#ratings-table") %>%
  html_table() %>% .[[1]]
cbb_df = data.frame(table)

new_column_names = cbb_df[1, ]
names(cbb_df) = new_column_names
cbb_df = cbb_df[-1, ]

cbb_df = cbb_df[-c(7,9,11,13,15,17,19,21)]
rownames(cbb_df) = NULL
cbb_df = cbb_df[-c(41,42,83,84,125,126,167,168,209,210,251,252,293,294,335,336,377,378), ]
rownames(cbb_df) = cbb_df$Rk

cbb_df = separate(cbb_df, "W-L", into = c("Wins", "Losses"), sep = "-")

cbb_df$Rk = as.numeric(cbb_df$Rk)
cbb_df$Team = removeNumbers(cbb_df$Team)
cbb_df$Wins = as.numeric(cbb_df$Wins)
cbb_df$Losses = as.numeric(cbb_df$Losses)
cbb_df$AdjEM = as.numeric(gsub("\\+", "", cbb_df$AdjEM))
cbb_df$AdjO = as.numeric(cbb_df$AdjO)
cbb_df$AdjD = as.numeric(cbb_df$AdjD)
cbb_df$AdjT = as.numeric(cbb_df$AdjT)
cbb_df$Luck = as.numeric(gsub("\\+", "", cbb_df$Luck))
cbb_df$AdjEM.1 = as.numeric(gsub("\\+", "", cbb_df$AdjEM.1))
cbb_df$OppO = as.numeric(cbb_df$OppO)
cbb_df$OppD = as.numeric(cbb_df$OppD)
cbb_df$AdjEM.2 = as.numeric(gsub("\\+", "", cbb_df$AdjEM.2))

names(cbb_df)[names(cbb_df) == "AdjEM"] = "EM"
names(cbb_df)[names(cbb_df) == "AdjO"] = "OE"
names(cbb_df)[names(cbb_df) == "AdjD"] = "DE"
names(cbb_df)[names(cbb_df) == "AdjT"] = "Tempo"
names(cbb_df)[names(cbb_df) == "AdjEM.1"] = "SOS"
names(cbb_df)[names(cbb_df) == "AdjEM.2"] = "NCSOS"
```

Fitting a MLR

```
## Full Model with all variables
power_5 = c("ACC", "Big 12", "Big Ten", "Pac-12", "SEC")
predictive_data = cbb_df %>% filter(Conf %in% power_5)
head(predictive_data)
```

```
   Rk            Team Conf Wins Losses    EM    OE    DE Tempo   Luck   SOS
4   4          Auburn  SEC   27      8 28.04 120.4  92.4  70.0 -0.080  9.55
5   5       Tennessee  SEC   27      9 26.69 116.8  90.1  69.3 -0.026 13.43
7   7            Duke  ACC   27      9 26.51 121.6  95.1  66.4 -0.064 10.11
9   9 North Carolina  ACC   29      8 26.21 119.7  93.5  70.6 -0.038 12.21
12 12         Alabama  SEC   25     11 23.26 125.8 102.6  72.8  0.004 14.20
19 19         Clemson  ACC   24     12 19.48 117.8  98.3  66.3 -0.018 12.14
    OppO  OppD NCSOS
4  111.9 102.3  1.49
5  114.7 101.3  9.03
7  111.2 101.1 -0.02
9  112.7 100.5  6.97
```

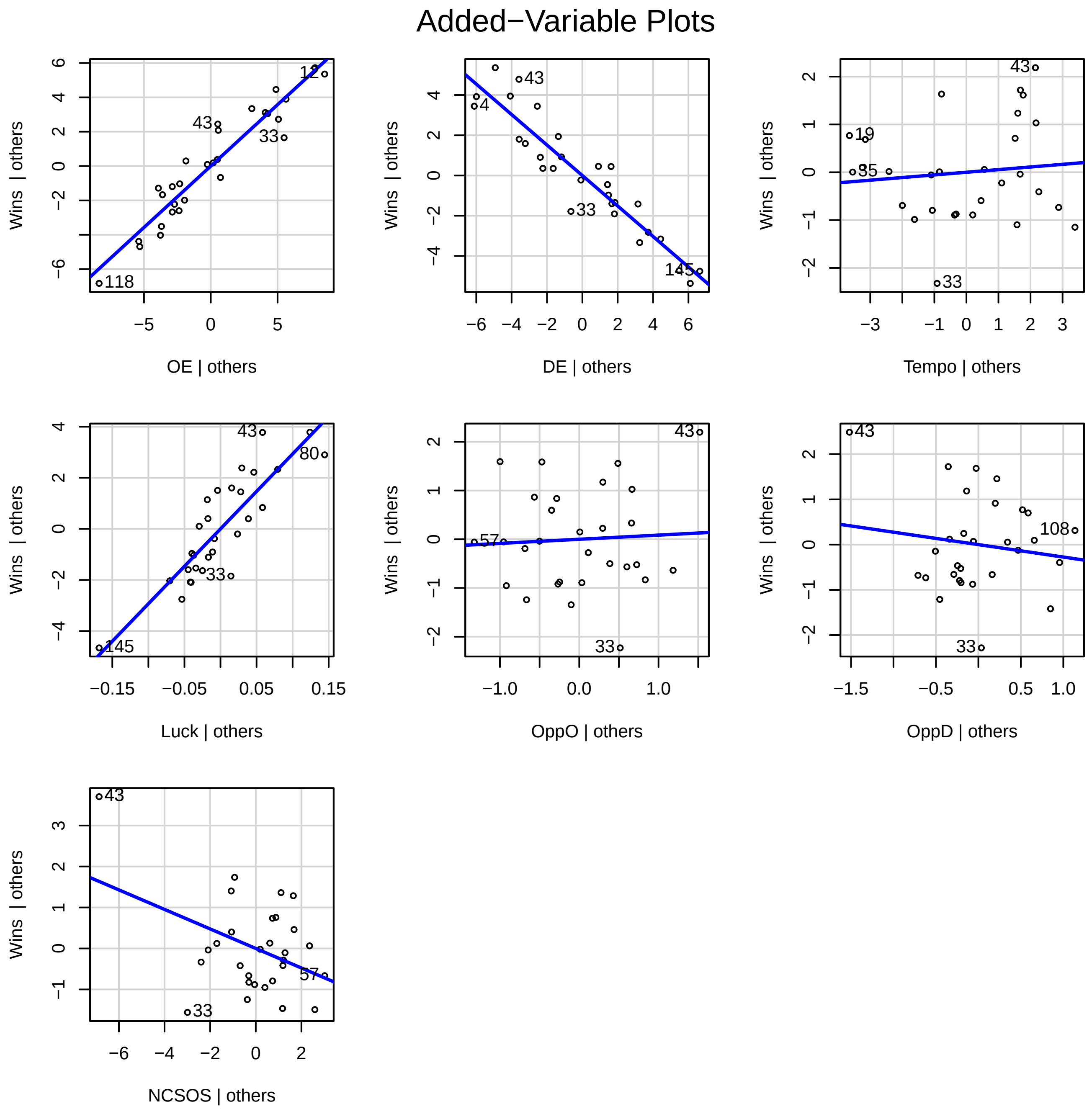
```
12 114.8 100.6  9.51
19 113.5 101.4  4.94

predictive_data = predictive_data[ ,-c(2,3)]

full_mdl = lm(Wins ~ ., data = predictive_data)
summary(full_mdl)


Call:
lm(formula = Wins ~ ., data = predictive_data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.55924 -0.60416  0.09415  0.57111  2.17198

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 15.95636   53.13600   0.300  0.76760
Rk          -0.02226    0.02179  -1.021  0.32140
Losses       0.31417    0.36963   0.850  0.40716
EM          -5.76664    7.57608  -0.761  0.45699
OE           6.52353    7.68516   0.849  0.40776
DE          -6.57568    7.66450  -0.858  0.40286
Tempo        0.09560    0.12087   0.791  0.43991
Luck        36.94904   11.71008   3.155  0.00578 **
SOS          4.18366    5.65150   0.740  0.46923
OppO        -4.33568    5.64431  -0.768  0.45294
OppD         4.25014    5.81120   0.731  0.47451
NCSOS       -0.18273    0.13092  -1.396  0.18075
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.244 on 17 degrees of freedom
Multiple R-squared:  0.9719,    Adjusted R-squared:  0.9537
F-statistic: 53.46 on 11 and 17 DF,  p-value: 5.791e-11

vif(full_mdl)

          Rk       Losses           EM           OE           DE        Tempo
   22.361931    52.054118 71590.744114 34283.735417 21213.634683     2.162818
        Luck          SOS         OppO         OppD        NCSOS
   10.765374  1264.318222   763.676802   328.828612     5.037117

## Reduced model to eliminate multicollinearity
reduced_mdl = lm(Wins ~ OE + DE + Tempo + Luck + OppO + OppD + NCSOS, data = predictive_data)
summary(reduced_mdl)


Call:
lm(formula = Wins ~ OE + DE + Tempo + Luck + OppO + OppD + NCSOS,
    data = predictive_data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.27310 -0.85619  0.00364  0.86045  2.06740

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 29.56961   46.68942   0.633   0.5334
OE           0.71474    0.05289  13.515 7.89e-12 ***
DE          -0.75783    0.06367 -11.902 8.47e-11 ***
Tempo        0.05525    0.11169   0.495   0.6260
Luck        29.32689    3.72684   7.869 1.07e-07 ***
OppO         0.08518    0.32619   0.261   0.7965
OppD        -0.27482    0.40818  -0.673   0.5081
NCSOS       -0.23819    0.11479  -2.075   0.0505 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.202 on 21 degrees of freedom
Multiple R-squared:  0.9676,    Adjusted R-squared:  0.9568
F-statistic: 89.61 on 7 and 21 DF,  p-value: 3.346e-14
```

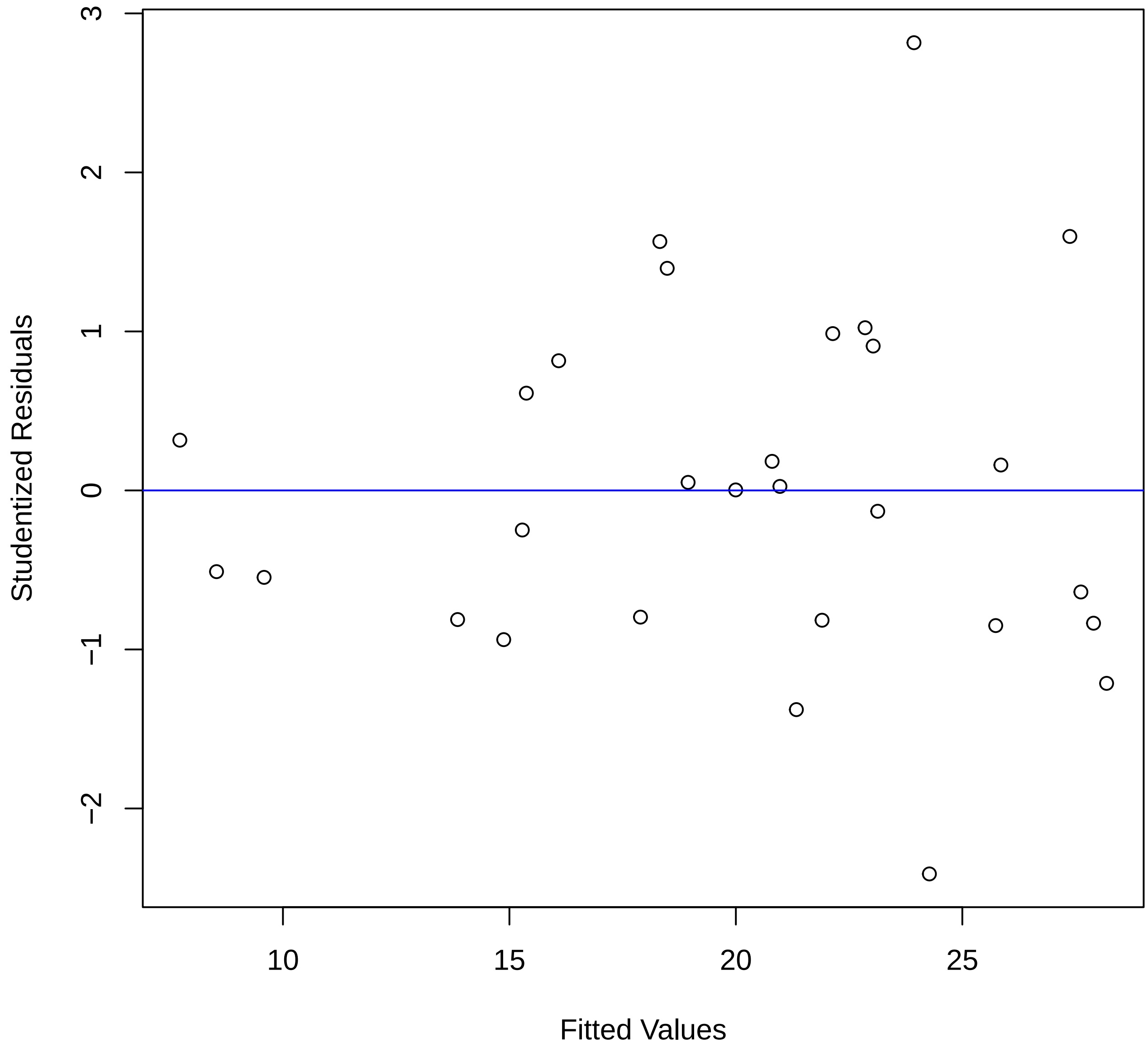
```
vif(reduced_mdl)
```

```
      OE       DE    Tempo     Luck     OppO     OppD    NCSOS
1.739551 1.568699 1.978787 1.168319 2.732746 1.738243 4.149503
```

```
avPlots(reduced_mdl)
```



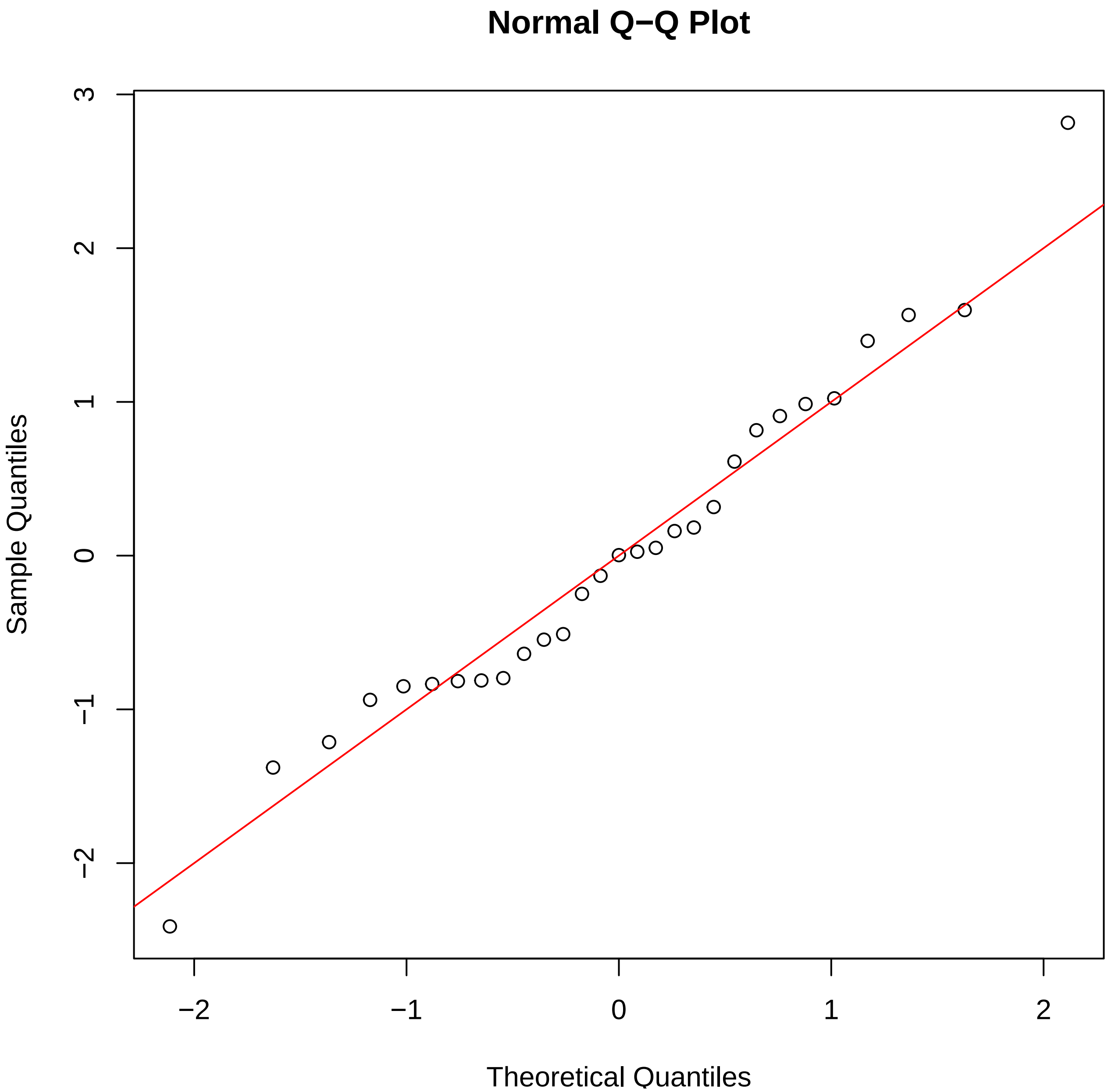

```r
## Assumptions
student_r = rstudent(reduced_mdl)
fitted_values = reduced_mdl$fitted.values

plot(fitted_values, student_r, xlab = 'Fitted Values', ylab = 'Studentized Residuals')
abline(0,0, col = 'blue')
```



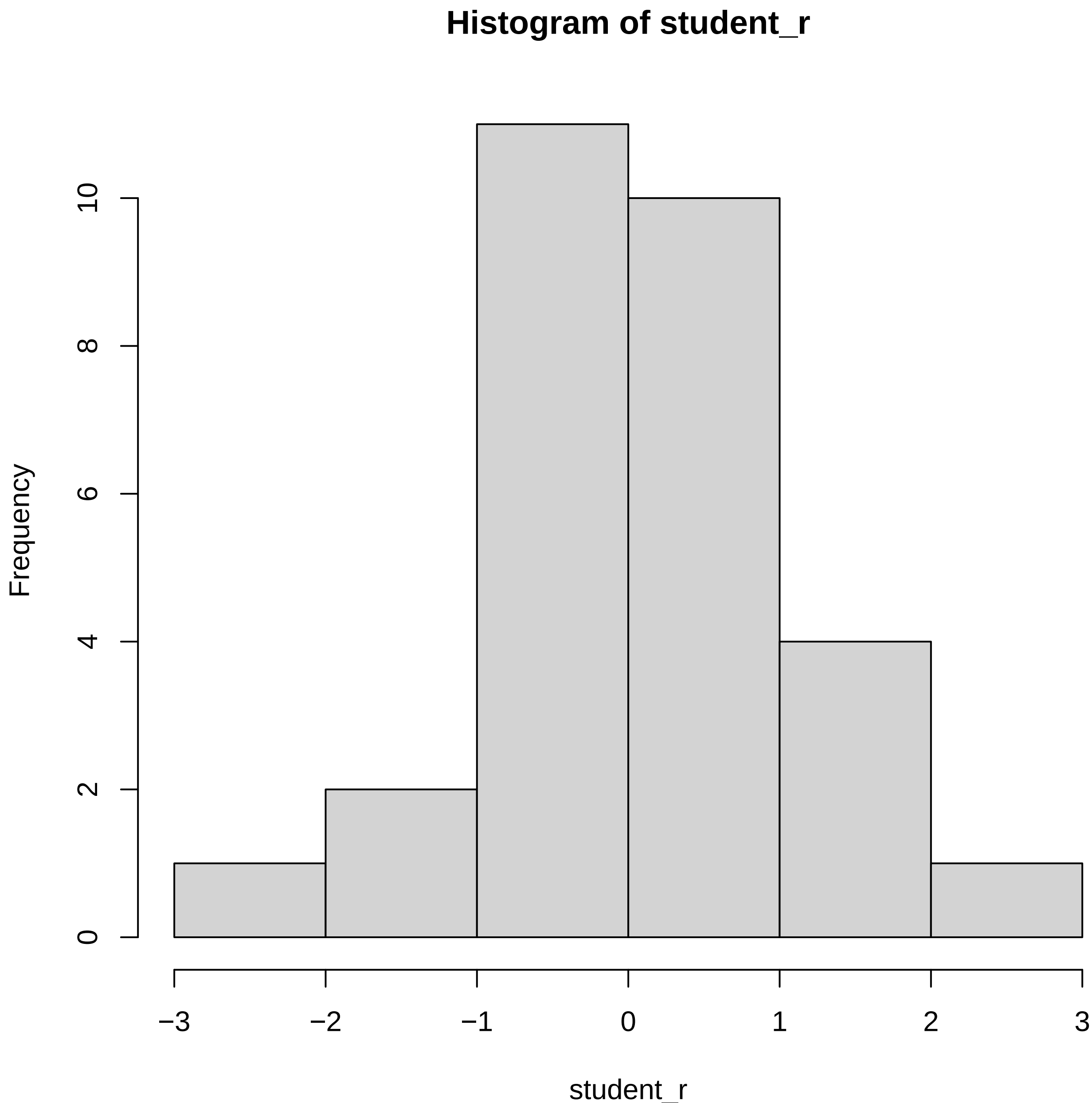

```
bptest(reduced_mdl)
```

```

	studentized Breusch-Pagan test

data:  reduced_mdl
BP = 10.979, df = 7, p-value = 0.1396
```

```
qqnorm(student_r)
abline(0,1,col='red')
```



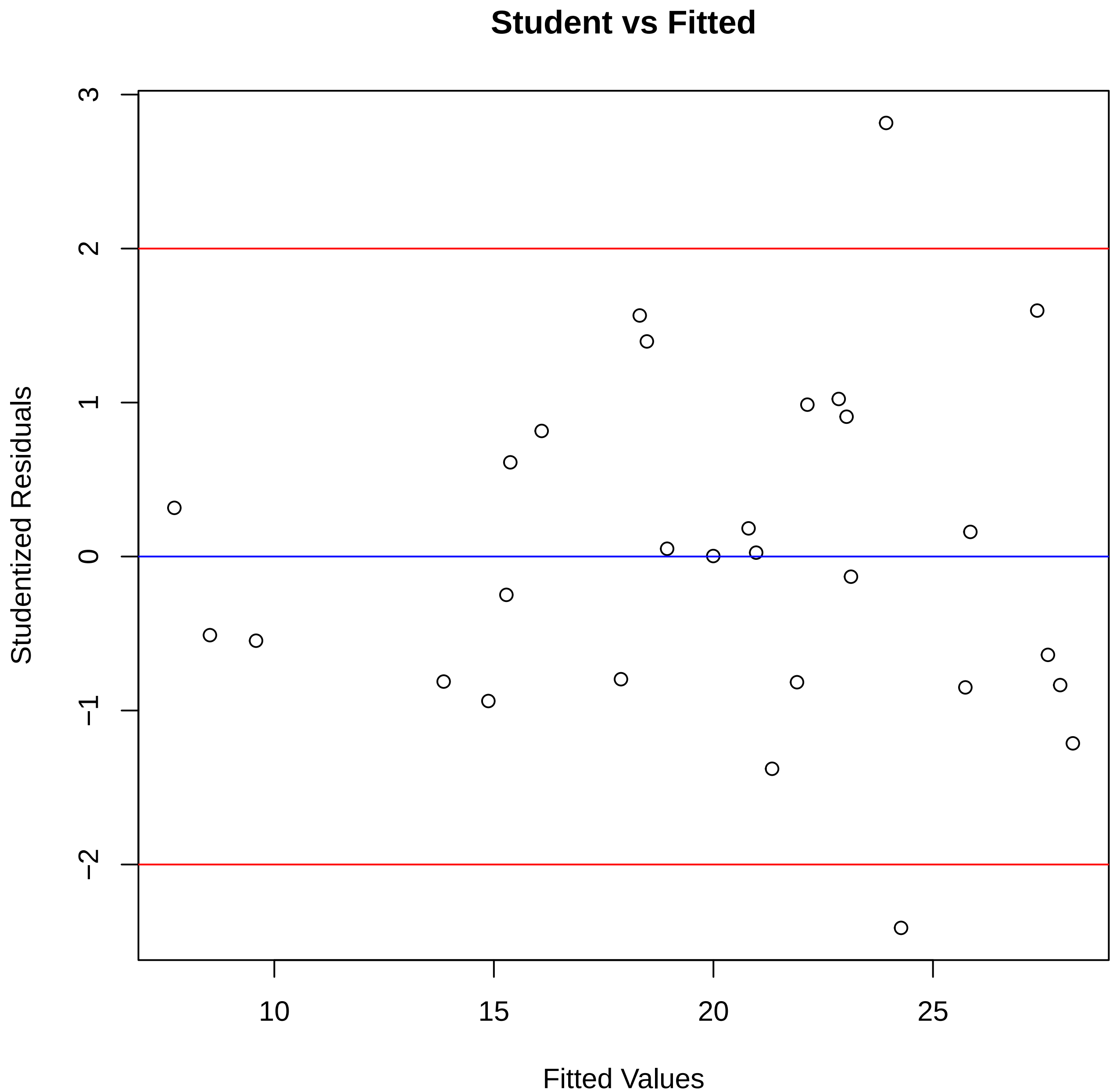

```
hist(student_r)
```



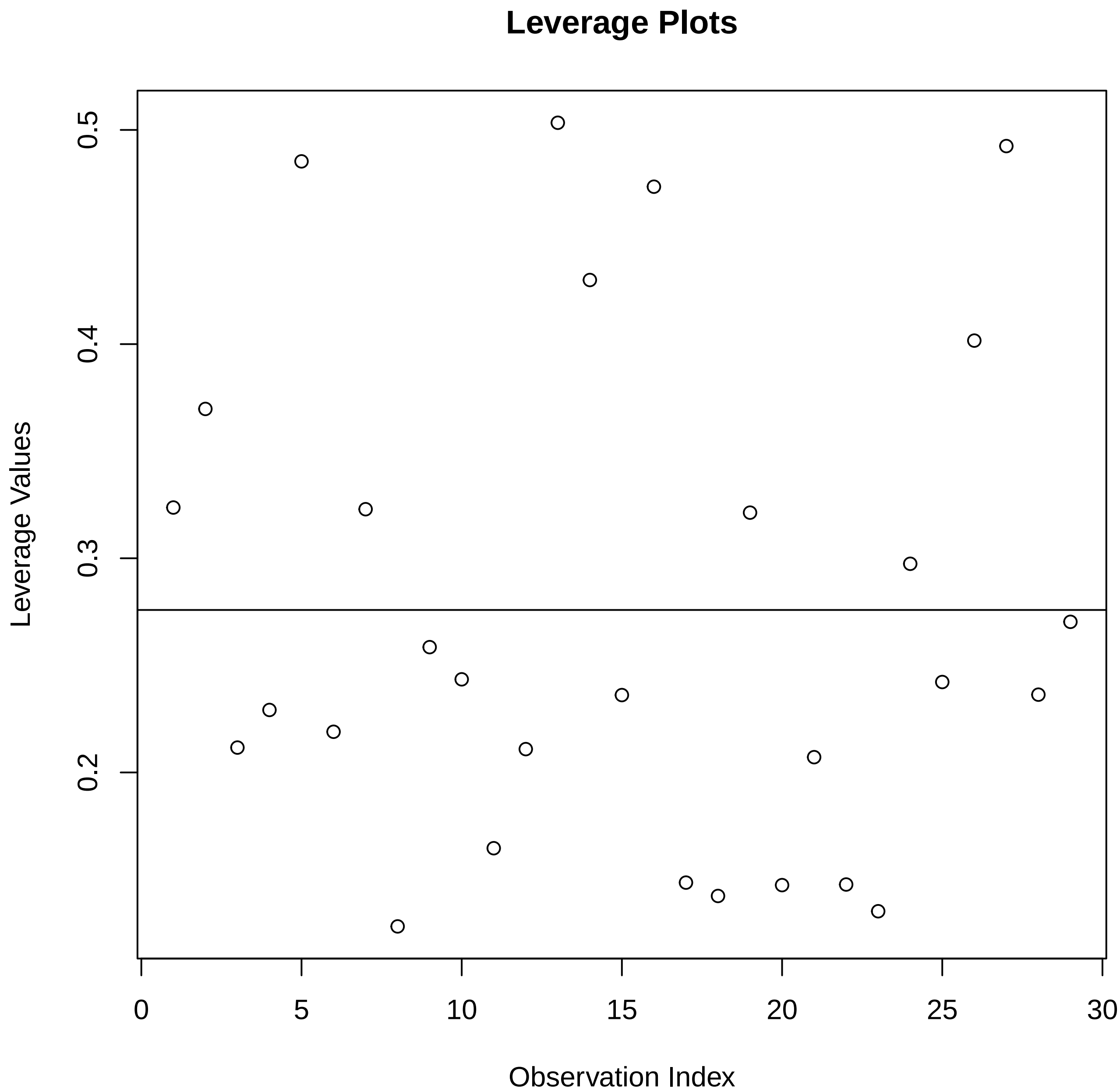

```
shapiro.test(student_r)
```

```

	Shapiro-Wilk normality test

data:  student_r
W = 0.97442, p-value = 0.6841
```

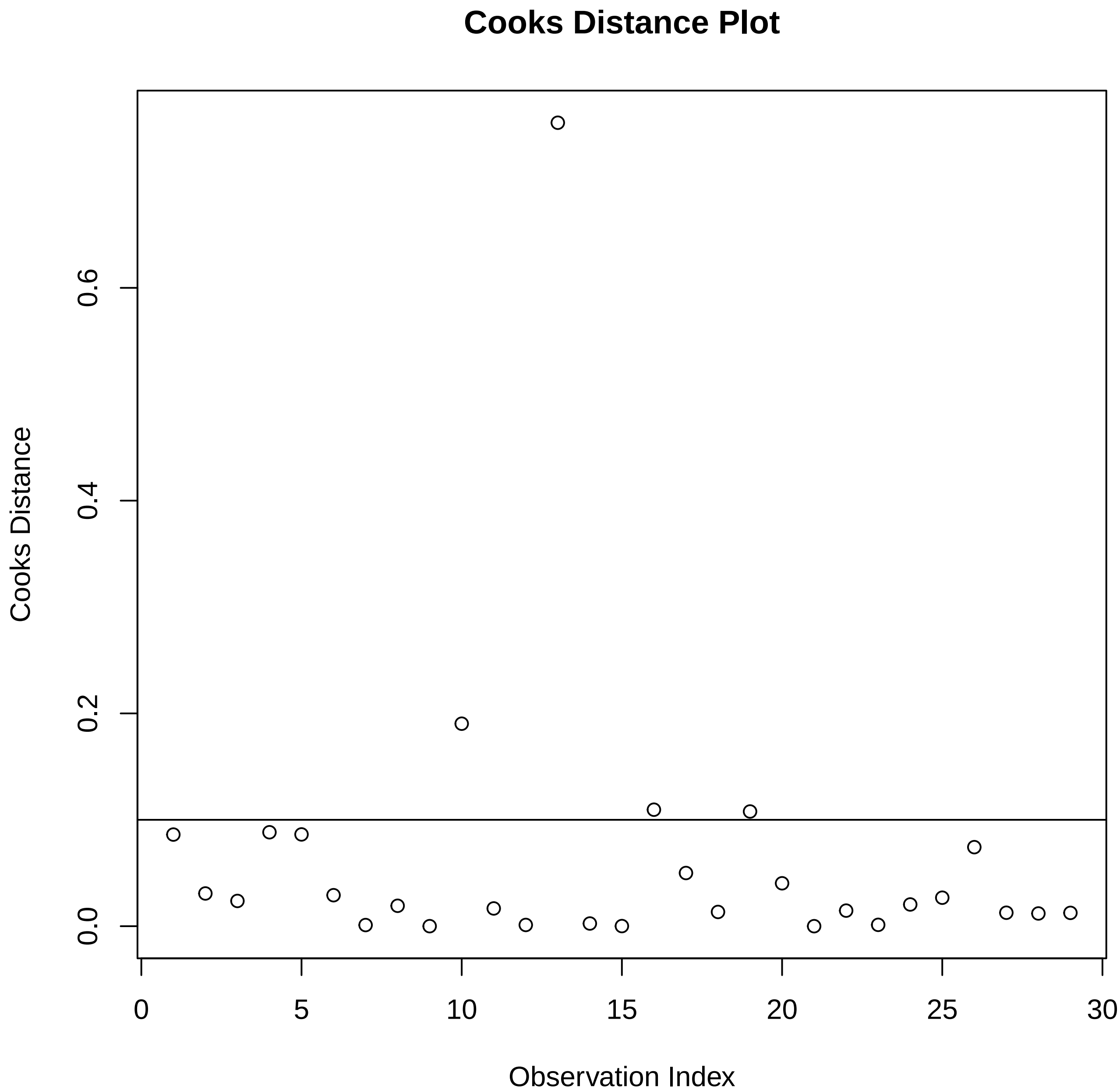
```
## Outliers
plot(fitted_values, student_r, xlab = 'Fitted Values', ylab = 'Studentized Residuals', main = 'Student vs Fitt
abline(0,0, col = 'blue')
abline(h = c(-2,2), col= 'red')
```




```
h = hatvalues(reduced_mdl)
plot(h, main = 'Leverage Plots', ylab = 'Leverage Values', xlab = 'Observation Index')
p = 4; n = length(predictive_data[,1])
cutoff = (2*p)/n
abline(h = cutoff, col = 'black')
```




```
cd = cooks.distance(reduced_mdl)
plot(cd, ylab = 'Cooks Distance', xlab = 'Observation Index', main = 'Cooks Distance Plot')
abline(h = 0.1, col = 'black')
```




```
predictive_data[which(cd > 0.6), ]

   Rk Wins Losses    EM    OE DE Tempo  Luck   SOS  OppO OppD NCSOS
43 43   26     14 16.08 115.1 99  68.1 0.014 10.84 111.8  101 -4.96

cbb_df[43, ]

   Rk        Team Conf Wins Losses    EM    OE DE Tempo  Luck   SOS  OppO OppD
43 43 N.C. State  ACC   26     14 16.08 115.1 99  68.1 0.014 10.84 111.8  101
   NCSOS
43 -4.96
```

Exhaustive Model Approach

```
regit.full = regsubsets(Wins ~ OE + DE + Tempo + Luck + OppO + OppD + NCSOS, data = predictive_data, method =
output = summary(regit.full, all.best = TRUE)
criterion_mat = cbind(output$rsq, output$adjr2, output$cp, output$bic)
colnames(criterion_mat) = c('R2', 'AdjR2', 'Cp', 'BIC')
results_mat = cbind(output$outmat, round(criterion_mat, 3))
results_mat
```

```
         OE  DE  Tempo Luck OppO OppD NCSOS R2      AdjR2   Cp        BIC
1  ( 1 ) "*" " " " "   " "  " "  " "  " "   "0.584" "0.569" "244.732" "-18.695"
2  ( 1 ) "*" "*" " "   " "  " "  " "  " "   "0.859" "0.848" "68.616"  "-46.642"
3  ( 1 ) "*" "*" " "   "*"  " "  " "  " "   "0.954" "0.948" "8.876"   "-75.771"
4  ( 1 ) "*" "*" " "   "*"  " "  " "  "*"   "0.967" "0.961" "2.618"   "-81.785"
5  ( 1 ) "*" "*" " "   "*"  " "  "*"  "*"   "0.967" "0.96"  "4.343"   "-78.79"
6  ( 1 ) "*" "*" "*"   "*"  " "  "*"  "*"   "0.968" "0.959" "6.068"   "-75.799"
7  ( 1 ) "*" "*" "*"   "*"  "*"  "*"  "*"   "0.968" "0.957" "8"       "-72.525"
```

Analyzing what was found to be the best model

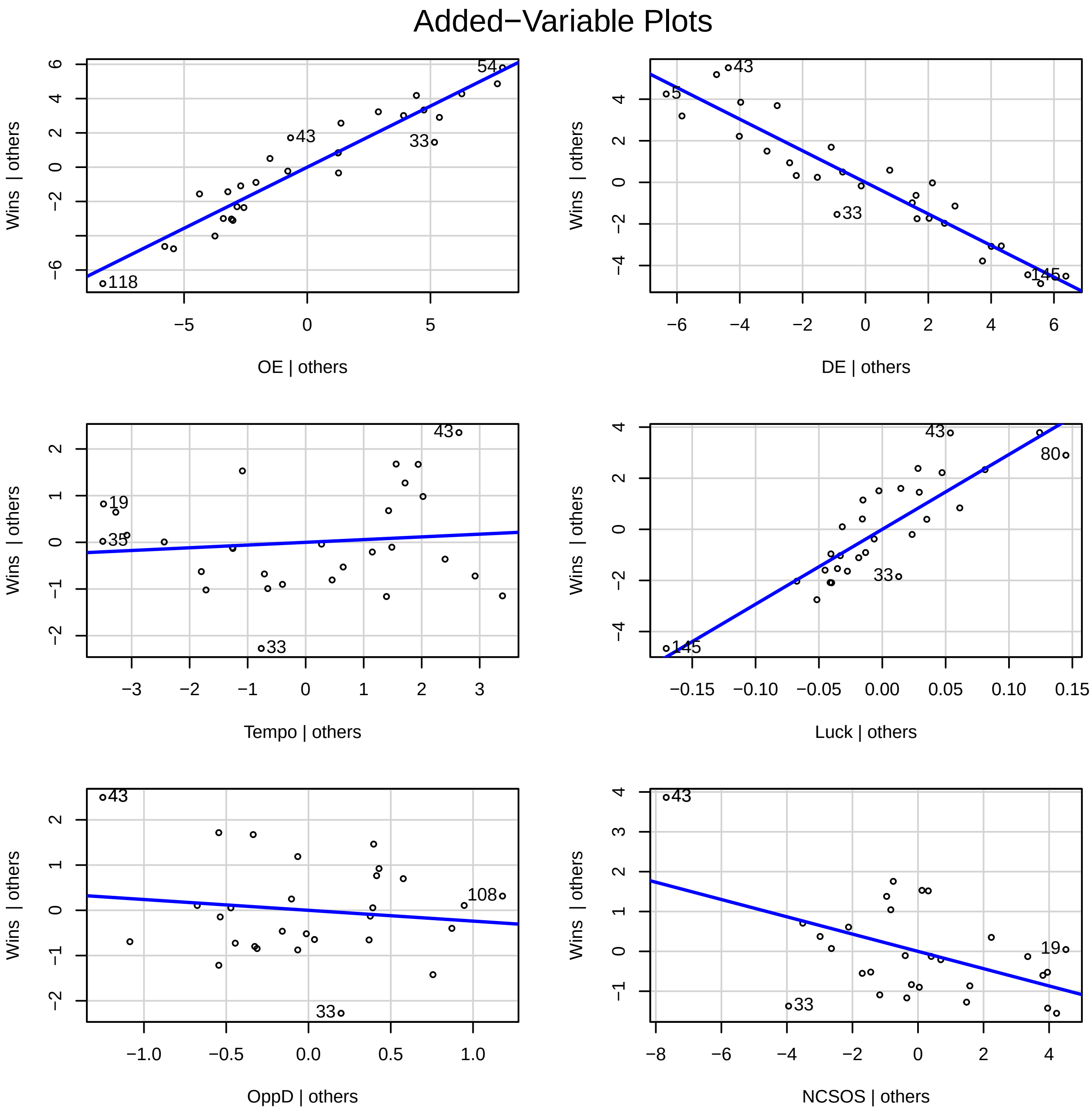
```r
best_mdl = lm(Wins ~ OE + DE + Tempo + Luck + OppD + NCSOS, data = predictive_data)
summary(best_mdl)
```

```
Call:
lm(formula = Wins ~ OE + DE + Tempo + Luck + OppD + NCSOS, data = predictive_data)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.22907 -0.83261 -0.05826  0.83629  2.19693 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 35.56792   39.77883   0.894   0.3809    
OE           0.71298    0.05133  13.889 2.29e-12 ***
DE          -0.75941    0.06203 -12.243 2.70e-11 ***
Tempo        0.05827    0.10871   0.536   0.5973    
Luck        29.29329    3.64489   8.037 5.47e-08 ***
OppD        -0.23863    0.37572  -0.635   0.5319    
NCSOS       -0.21686    0.07893  -2.748   0.0118 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.176 on 22 degrees of freedom
Multiple R-squared:  0.9675,    Adjusted R-squared:  0.9586 
F-statistic: 109.2 on 6 and 22 DF,  p-value: 3.136e-15
```

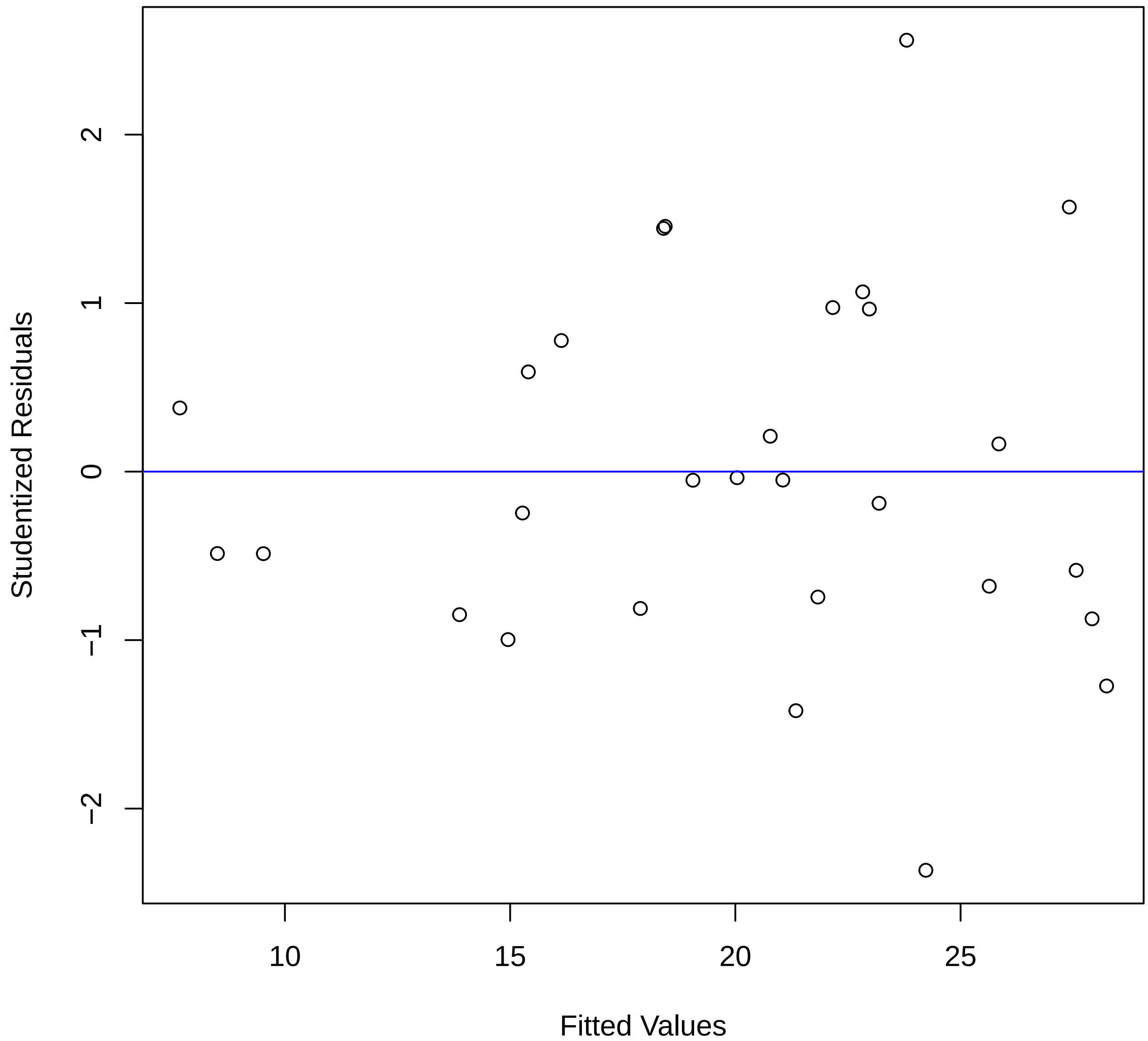
```r
vif(best_mdl)
```

```
      OE       DE    Tempo     Luck     OppD    NCSOS 
1.711360 1.554499 1.957497 1.166926 1.537902 2.048443 
```

```r
avPlots(best_mdl)
```



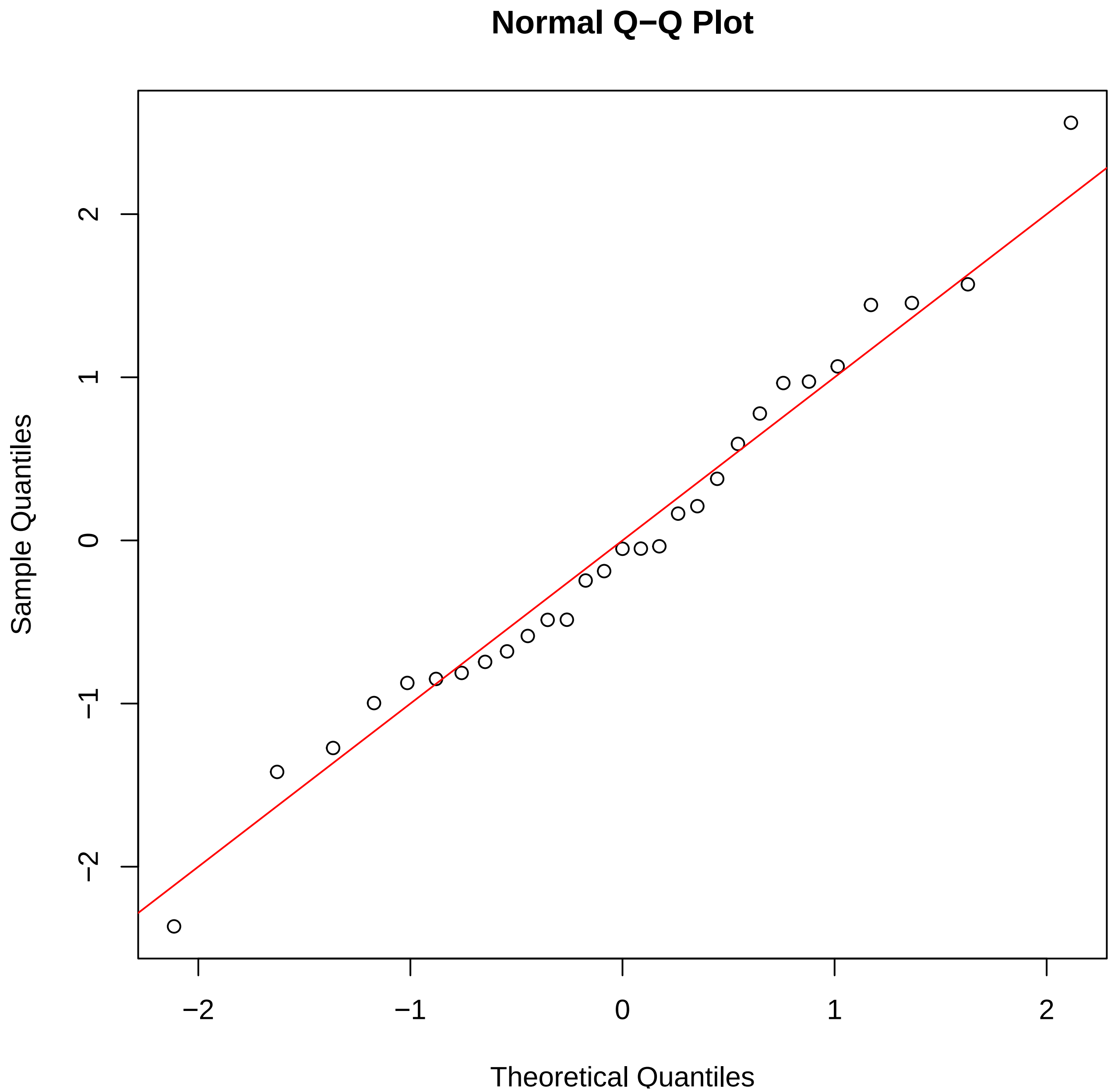

```r
## Assumptions
student_r = rstudent(best_mdl)
fitted_values = best_mdl$fitted.values

plot(fitted_values, student_r, xlab = 'Fitted Values', ylab = 'Studentized Residuals')
abline(0,0, col = 'blue')
```



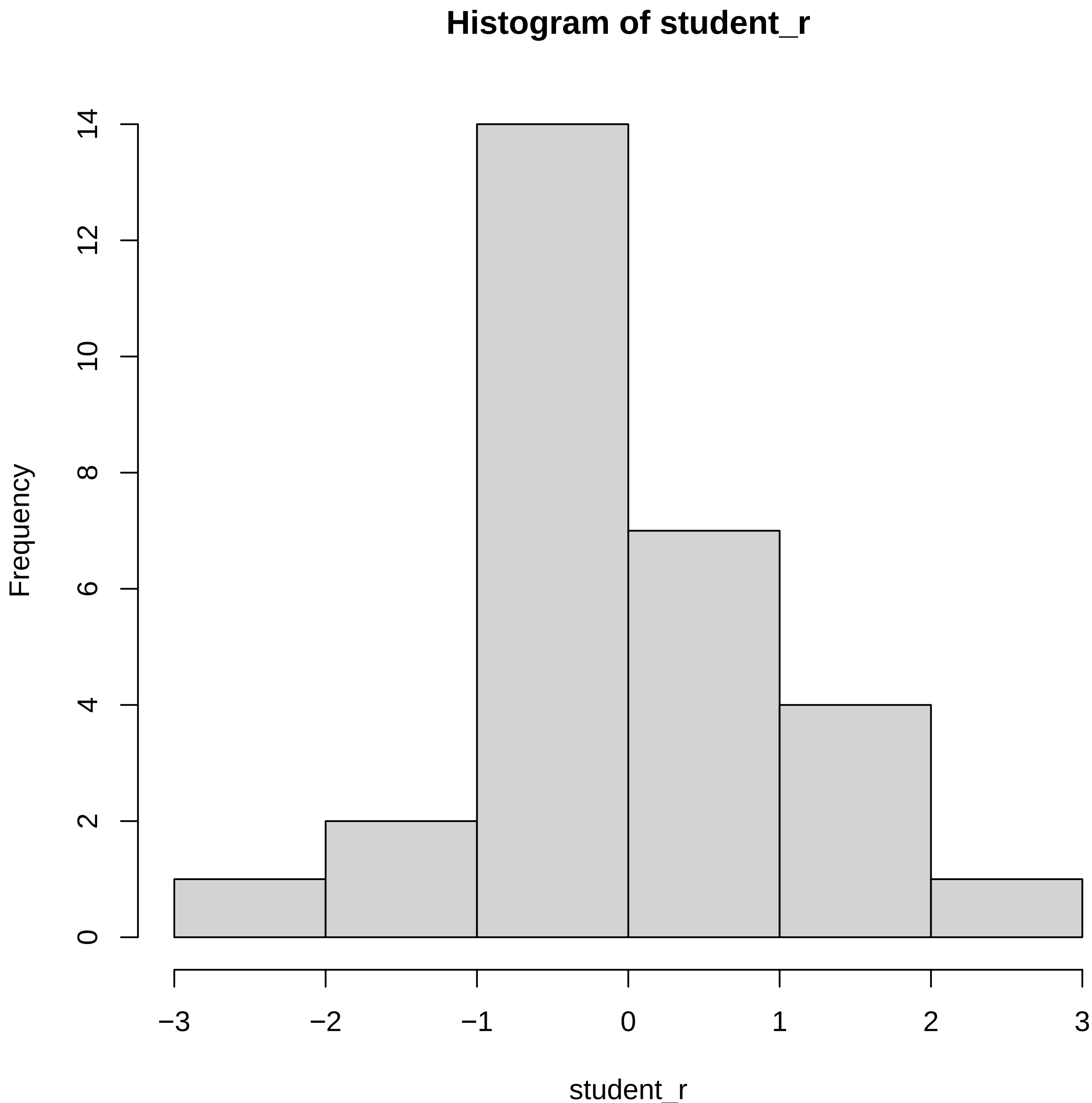

```
bptest(best_mdl)
```

```
	studentized Breusch-Pagan test

data:  best_mdl
BP = 9.9207, df = 6, p-value = 0.128
```

```
qqnorm(student_r)
abline(0,1,col='red')
```




```
hist(student_r)
```




```
shapiro.test(student_r)
```

```
	Shapiro-Wilk normality test

data:  student_r
W = 0.98252, p-value = 0.897
```

Ridge regression

Ridge regression is a regularization technique(Method in statistics used to reduce error caused by overfitting of data) for linear regression models. Used to get rid of overfitting in training data we use for our model. It is also know as L2 regularization. Problem that is solved using this regression is "Multicollinearity". In this technique of regilarization we add a bias into the model for decreasing model's variance.

Residual Sum Squares formula for linear regression is given by
$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Where:
n is the number of data points in the dataset.

y_i is the observed value of the dependent variable for data point
$\hat{y}_i$ is the predicted value of the dependent variable for data point i based on the regression model.
Where as by adding the regularization term according to Ridge regression we would get
$RSS_{ridge} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$

λ is the regularization parameter (also known as the ridge parameter or penalty parameter) that controls the strength of the regularization.}
p is the number of predictor variables (features) in the regression model.
β_j represents the coefficients (weights) associated with each predictor variable.