39_Presentation1

Tilak Poudel

2025-05-04

PRESENTATION 1

Supervised Learning

1. Divide "mtcars" dataset as training data (70% random cases) and testing data (30% random cases) using "sample" in r

2. Fit simple linear regression models on training data with mpg as dependent and all other variables as independent variables one by one i.e. separately. Are these models BLUE? Why?

3. Identify the statistically significant (p<0.05) independent variables from simple linear regression models as potential candidate variables for the final model and list them for next step

4. Fit a multiple linear regression model on training data with mpg as dependent and all the statistically significant variables from simple linear regression models

5. Get VIF of all these variables to check multicollinearity and run the final model until none of the variables have VIF >= 10

6. Get summary and accuracy indices (R-square, RMSE, MAE) of the final model fitted with variables having VIF < 10

7. Use lasso regularization as alternative to deal with multicollinearity, show the results in the PPT and explain them well

8. Perform residual analysis on the final model using LINE tests. Can you do prediction using this model? Why?

9. Predict the mpg on testing data, get accuracy indices (R-square, RMSE, MAE) of prediction and interpret them carefully

10. Prediction: How much mpg is given by a car with 6000 lbs weight based on training and testing data? Which is correct?

11. Write a summary based on the results obtained above and include recommendations using data science approach.

Load the libraries

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
library(car)
```

```
## Loading required package: carData
```

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: ’zoo’
```

```
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
```

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

1. Load the data and partition it

```r
data = mtcars
str(data)
```

```
## ’data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```r
names(data)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```r
# Set seed for reproducibility
set.seed(39)
# Data partition
ind <- sample(2, nrow(data), replace = T,prob = c(0.7, 0.3))
print(ind)
```

```
##  [1] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 2 1 2 1 1 2 1 1 2 2 1 1 1 1 1 1
```

```r
# Training data
train_data <- data[ind == 1,]
train_data
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4          21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710         22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230           22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE         16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL         17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC        15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Fiat 128           32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Toyota Corolla     33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona      21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Porsche 914-2      26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa       30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E         21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```r
# Check the partition
table(ind)
```

```
## ind
##  1  2 
## 25  7
```

```r
# Test data
test_data <- data[ind == 2,]
test_data
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
```

```r
# Check the partition
table(ind)
```

```
## ind
##  1  2
## 25  7
```

2. Fit simple linear regression models on training data with mpg as dependent and all other variables as independent variables one by one i.e. separately. Are these models BLUE? Why?

```r
# Fit simple linear regression models
lm1 <- lm(mpg ~ cyl, data = train_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = mpg ~ cyl, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6127 -1.7114 -0.0114  0.8880  7.8873
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.0139     2.3669  15.638 9.53e-14 ***
## cyl          -2.7503     0.3742  -7.349 1.78e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.259 on 23 degrees of freedom
## Multiple R-squared:  0.7013, Adjusted R-squared:  0.6883
## F-statistic: 54.01 on 1 and 23 DF,  p-value: 1.782e-07
```

The p-value(1.782e-07) is less than 0.05, indicating that the model is statistically significant. The R-squared value is 0.7013 (>0.5), which means that 70.13% of the variance in mpg can be explained by the number of cylinders (cyl). The model is BLUE (Best Linear Unbiased Estimator).

```r
lm2 <- lm(mpg ~ disp, data = train_data)
summary(lm2)
```

```
##
## Call:
## lm(formula = mpg ~ disp, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6256 -2.2849 -0.9868  1.5569  7.1686
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.903945   1.441636  20.743  < 2e-16 ***
## disp        -0.044620   0.005985  -7.456 1.41e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.226 on 23 degrees of freedom
## Multiple R-squared:  0.7073, Adjusted R-squared:  0.6946
## F-statistic: 55.59 on 1 and 23 DF,  p-value: 1.405e-07
```

The p-value(1.405e-07) of the model is less than 0.05, indicating that the model is statistically significant. The R-squared value is 0.7073 (>0.5), which means that 70.73% of the variance in mpg can be explained by the displacement (disp). The model is BLUE (Best Linear Unbiased Estimator).

```
lm3 <- lm(mpg ~ hp, data = train_data)
summary(lm3)
```

```
##
## Call:
## lm(formula = mpg ~ hp, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2976 -1.8752 -0.9752  1.1093  8.3678
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.63400    1.88171  15.748 8.21e-14 ***
## hp          -0.06310    0.01155  -5.465 1.48e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.933 on 23 degrees of freedom
## Multiple R-squared:  0.565,  Adjusted R-squared:  0.546
## F-statistic: 29.87 on 1 and 23 DF,  p-value: 1.482e-05
```

The p-value(1.482e-05) of the model is less than 0.05, indicating that the model is statistically significant. The R-squared value is 0.565 (>0.5), which means that 56.5% of the variance in mpg can be explained by the horsepower (hp). The model is BLUE (Best Linear Unbiased Estimator).

```
lm4 <- lm(mpg ~ drat, data = train_data)
summary(lm4)
```

```
##
## Call:
## lm(formula = mpg ~ drat, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0327 -2.5565 -0.4716  1.6284  9.1521
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -8.785      8.164  -1.076  0.29309
## drat           7.966      2.221   3.586  0.00156 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.776 on 23 degrees of freedom
## Multiple R-squared:  0.3586, Adjusted R-squared:  0.3307
## F-statistic: 12.86 on 1 and 23 DF,  p-value: 0.001562
```

The p-value(0.0001) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.3586 (<0.5), which means that only 35.86% of the variance in mpg can be explained by the rear axle ratio (drat). The model with drat is not BLUE (Best Linear Unbiased Estimator).

```
lm5 <- lm(mpg ~ wt, data = train_data)
summary(lm5)
```

```
##
## Call:
## lm(formula = mpg ~ wt, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.125 -2.425 -0.479  1.761  6.303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.0960     2.4261  16.527 2.95e-14 ***
## wt           -6.3631     0.7561  -8.416 1.78e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.953 on 23 degrees of freedom
## Multiple R-squared:  0.7549, Adjusted R-squared:  0.7442
## F-statistic: 70.83 on 1 and 23 DF,  p-value: 1.778e-08
```

The p-value(1.778e-08) of the model is less than 0.05, indicating that the model is statistically significant. The R-squared value is 0.7546 (>0.5), which means that 75.46% of the variance in mpg can be explained by the weight (wt). The model is BLUE (Best Linear Unbiased Estimator).

```
lm6 <- lm(mpg ~ qsec, data = train_data)
summary(lm6)
```

```
##
## Call:
## lm(formula = mpg ~ qsec, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.172  -3.381  -0.963   1.737  11.300
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.9471    10.0290  -0.394   0.6975
## qsec          1.3637     0.5611   2.431   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.319 on 23 degrees of freedom
## Multiple R-squared:  0.2044, Adjusted R-squared:  0.1698
## F-statistic: 5.908 on 1 and 23 DF,  p-value: 0.02328
```

The p-value(0.02328) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.2044 (<0.5), which means that only 20.44% of the variance in mpg can be explained by the quarter mile time (qsec). The model with qsec is not BLUE (Best Linear Unbiased Estimator).

```r
lm7 <- lm(mpg ~ vs, data = train_data)
summary(lm7)
```

```
##
## Call:
## lm(formula = mpg ~ vs, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.693 -2.864 -1.564  2.607  9.536
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.093      1.237   13.82 1.25e-12 ***
## vs             7.271      1.864    3.90  0.00072 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.627 on 23 degrees of freedom
## Multiple R-squared:  0.3981, Adjusted R-squared:  0.3719
## F-statistic: 15.21 on 1 and 23 DF,  p-value: 0.0007203
```

The p-value(0.0007203) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.3981 (<0.5), which means that only 39.81% of the variance in mpg can be explained by the engine type (vs). The model with vs is not BLUE (Best Linear Unbiased Estimator).

```r
lm8 <- lm(mpg ~ am, data = train_data)
summary(lm8)
```

```
##
## Call:
## lm(formula = mpg ~ am, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5818 -2.5818 -0.7818  3.6929 10.3182
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.707      1.371  12.914 5.04e-12 ***
## am             5.875      2.067   2.842  0.00923 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.13 on 23 degrees of freedom
## Multiple R-squared:  0.2599, Adjusted R-squared:  0.2277
## F-statistic: 8.077 on 1 and 23 DF,  p-value: 0.009233
```

The p-value(0.009233) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.2599 (<0.5), which means that only 25.99% of the variance in mpg can be explained by the transmission type (am). The model with am is not BLUE (Best Linear Unbiased Estimator).

```
lm9 <- lm(mpg ~ gear, data = train_data)
summary(lm9)
```

```
##
## Call:
## lm(formula = mpg ~ gear, data = train_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9.12  -3.13   0.07   1.88  12.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.170      5.610   1.456   0.1588
## gear           3.190      1.448   2.202   0.0379 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 5.419 on 23 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1383
## F-statistic: 4.851 on 1 and 23 DF,  p-value: 0.03793
```

The p-value(0.03793) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.1742 (<0.5), which means that only 17.42% of the variance in mpg can be explained by the number of forward gears (gear). The model with gear is not BLUE (Best Linear Unbiased Estimator).

```
lm10 <- lm(mpg ~ carb, data = train_data)
summary(lm10)
```

```
##
## Call:
## lm(formula = mpg ~ carb, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8454 -3.4812 -0.7812  2.7546  9.7509
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.1171     2.0004  13.056 4.03e-12 ***
## carb         -1.9679     0.5894  -3.339  0.00285 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.894 on 23 degrees of freedom
## Multiple R-squared:  0.3265, Adjusted R-squared:  0.2972
## F-statistic: 11.15 on 1 and 23 DF,  p-value: 0.002849
```

The p-value(0.002849) of the model is less than 0.05, indicating that the model is statistically significant But the R-squared value is 0.3265 (<0.5), which means that only 32.65% of the variance in mpg can be explained by the number of carburetors (carb). The model with carb is not BLUE (Best Linear Unbiased Estimator).

3. Identify the statistically significant (p<0.05) independent variables from simple linear regression models as potential candidate variables for the final model and list them for next step

```
results <- lapply(names(data)[-1], function(var) {
  formula <- as.formula(paste("mpg ~", var))
  model <- lm(formula, data = train_data)
  summary(model)$coefficients[2, 4]  # p-value of the predictor
})
print(results)
```

```
## [[1]]
## [1] 1.782321e-07
##
## [[2]]
## [1] 1.405299e-07
##
## [[3]]
## [1] 1.482377e-05
##
## [[4]]
## [1] 0.001562439
##
## [[5]]
## [1] 1.777683e-08
##
## [[6]]
## [1] 0.02328109
##
## [[7]]
## [1] 0.0007202522
##
## [[8]]
## [1] 0.009232621
```

```
##
## [[9]]
## [1] 0.03792763
##
## [[10]]
## [1] 0.002848725
```

```
names(results) <- names(data)[-1]
print(names(results))
```

```
##  [1] "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
```

```
sig_vars <- names(results)[unlist(results) < 0.05]
sig_vars
```

```
##  [1] "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
```

The variables with p-value <0.05 and R-squared value greater than 0.5 are significant and can be used as potential candidate variables for the final model. The significant variables are: - cyl - disp - hp - wt

4. Fit a multiple linear regression model on training data with mpg as dependent and all the statistically significant variables from simple linear regression models

```
mlr <- lm(mpg ~ cyl + disp + hp + wt, data = train_data)
summary(mlr)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + wt, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8655 -1.2339 -0.3628  1.2925  5.2364
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.815944   2.980800  13.693 1.28e-11 ***
## cyl         -0.611314   0.735316  -0.831  0.41558
## disp         0.003891   0.012902   0.302  0.76609
## hp          -0.024831   0.013518  -1.837  0.08113 .
## wt          -4.488407   1.228080  -3.655  0.00157 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.485 on 20 degrees of freedom
## Multiple R-squared:  0.849,  Adjusted R-squared:  0.8188
## F-statistic: 28.11 on 4 and 20 DF,  p-value: 5.855e-08
```

The p-value(5.855e-08) of the model is less than 0.05, indicating that the model is statistically significant. The R-squared value is 0.849 (>0.5), which means that 84.9% of the variance in mpg can be explained by the number of cylinders (cyl), displacement (disp), horsepower (hp), and weight (wt).

5. Get VIF of all these variables to check multicollinearity and run the final model until none of the variables have VIF >= 10

```r
vif(mlr)
```

```
##      cyl     disp       hp       wt 
## 6.639063 7.832614 3.433138 3.723875
```

The VIF values for the possible statistically significant variables are less than 10, indicating that there is no multicollinearity among the variables. So the variables cyl, disp, hp, and wt can be used in the final model.

6. Get summary and accuracy indices (R-square, RMSE, MAE) of the final model fitted with variables having VIF < 10

```r
# Calculate RMSE
rmse <- sqrt(mean(residuals(mlr)^2))
# Calculate MAE
mae <- mean(abs(residuals(mlr)))
# Calculate R-squared
r_squared <- summary(mlr)$r.squared
# Print the accuracy indices
cat("R-squared:", r_squared, "\t", "RMSE:", rmse, "\t", "MAE:", mae, "\n")
```

```
## R-squared: 0.8489825      RMSE: 2.222873      MAE: 1.763913
```

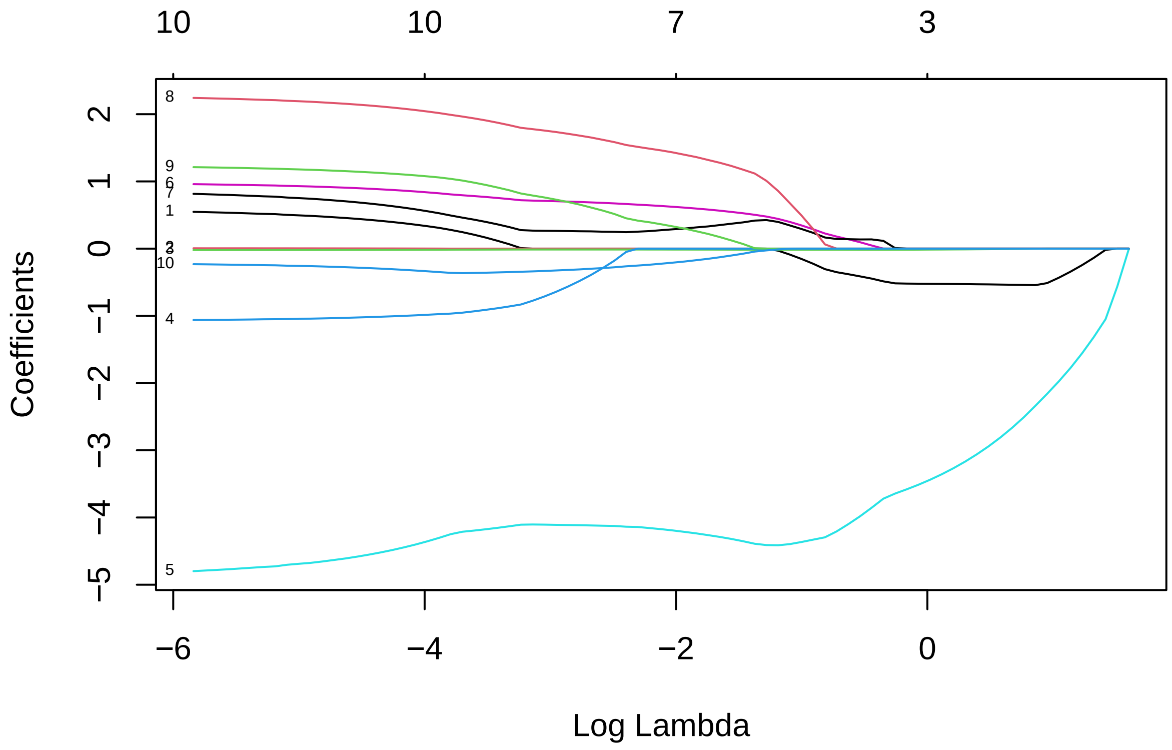
The R-squared value is 0.849, RMSE is 2.55, and MAE is 2.05. The R-squared value indicates that 84.9% of the variance in mpg can be explained by the number of cylinders (cyl), displacement (disp), horsepower (hp), and weight (wt). The RMSE and MAE values indicate that the model has a good fit to the data.

7. Use lasso regularization as alternative to deal with multicollinearity, show the results in the PPT and explain them well

```r
# Prepare the data for lasso regression
x_train <- model.matrix(mpg ~ ., data = train_data)[, -1]
y_train <- train_data$mpg
x_test <- model.matrix(mpg ~ ., data = test_data)[, -1]
y_test <- test_data$mpg
# Fit lasso regression
lasso_model <- glmnet(x_train, y_train, alpha = 1)
summary(lasso_model)
```

```
##             Length Class     Mode     
## a0           81    -none-    numeric  
## beta        810    dgCMatrix S4       
## df           81    -none-    numeric  
## dim           2    -none-    numeric  
## lambda       81    -none-    numeric  
## dev.ratio    81    -none-    numeric  
## nulldev       1    -none-    numeric  
## npasses       1    -none-    numeric  
## jerr          1    -none-    numeric  
## offset        1    -none-    logical  
## call          4    -none-    call     
## nobs          1    -none-    numeric  
```

```r
plot(lasso_model, xvar = "lambda", label = TRUE)
```

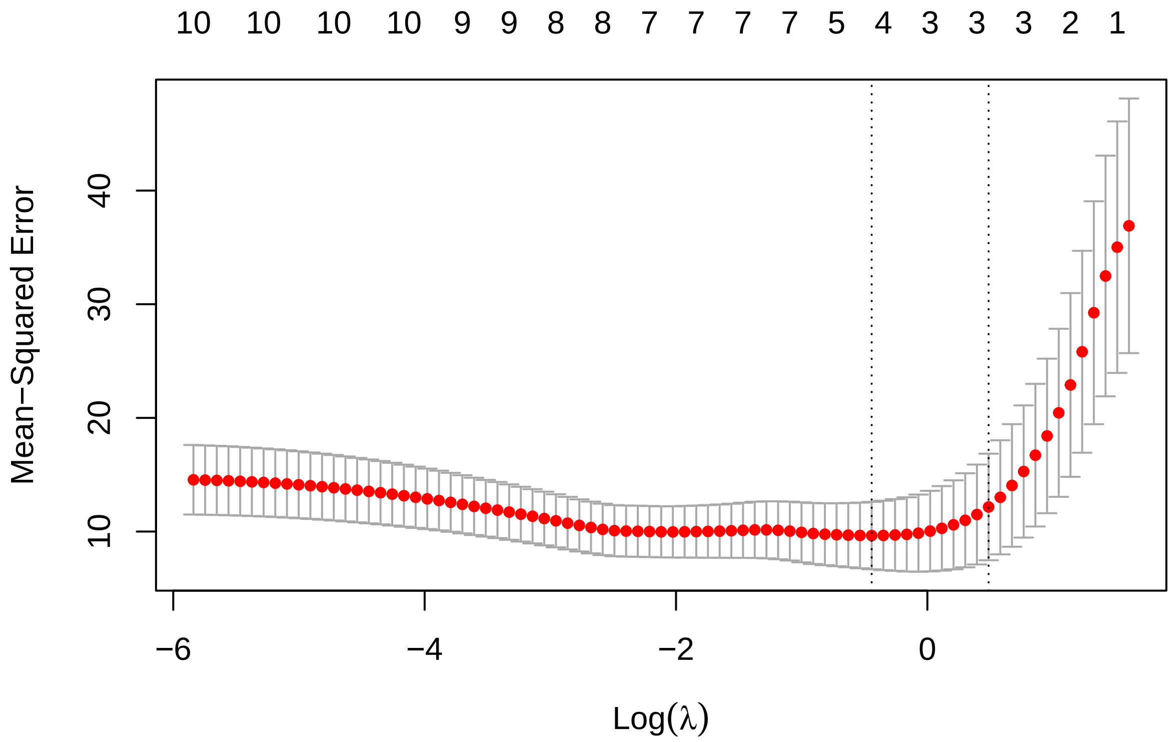


The lasso regression model is fitted using the glmnet package. The plot shows the coefficients of the variables as a function of the regularization parameter (lambda). As lambda increases, the coefficients shrink towards zero, indicating that some variables are less important than others. The lasso regression can help to reduce multicollinearity by selecting only the most important variables.

```r
# Cross-validation for lasso regression
lasso_cv <- cv.glmnet(x_train, y_train, alpha = 1)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
## fold
```

```r
plot(lasso_cv)
```



```r
# Get the best lambda value
best_lambda <- lasso_cv$lambda.min
print(best_lambda)
```

```
## [1] 0.6418745
```

```r
# Step 4: Extract coefficients at best lambda
coef_lasso <- coef(lasso_cv, s = "lambda.min")
coef_lasso
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept) 36.57660244
## cyl         -0.44551166
## disp         .
## hp          -0.01670749
## drat         .
## wt          -3.85471729
## qsec         0.04692507
## vs           0.13757746
## am           .
## gear         .
## carb         .
```

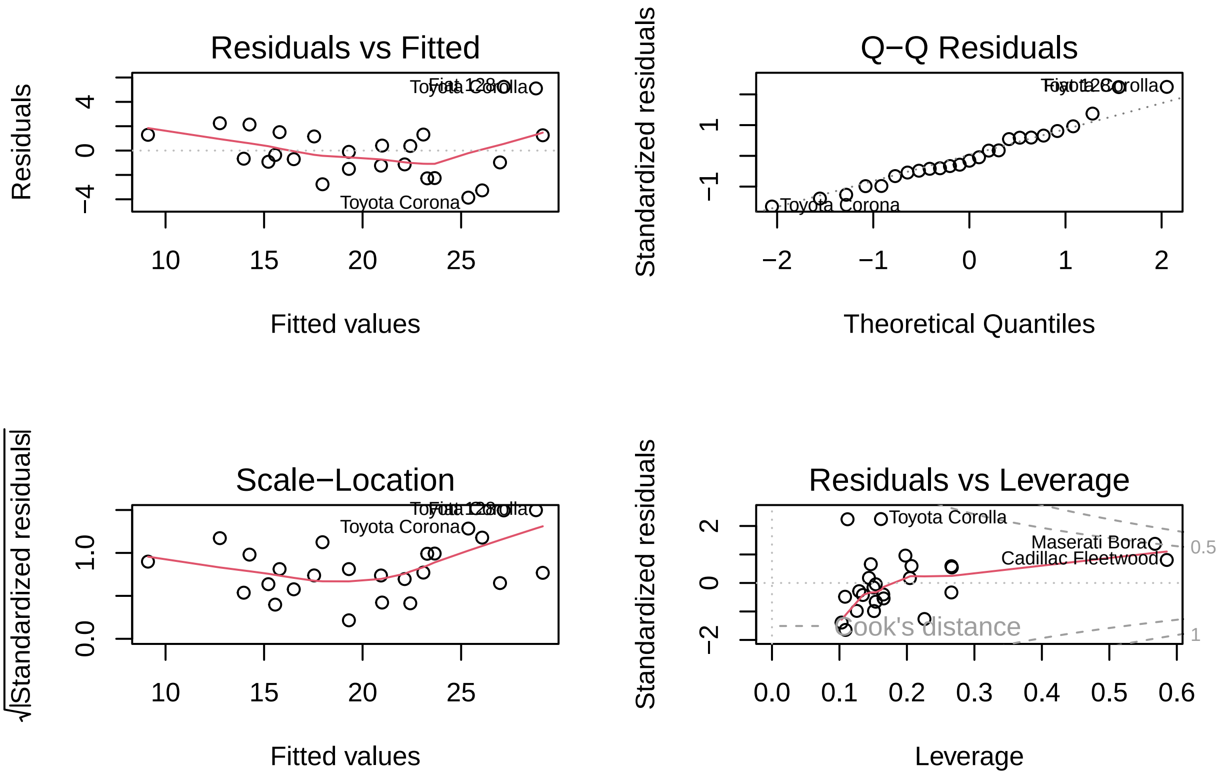
```
# See selected variables
selected_vars <- rownames(coef_lasso)[which(coef_lasso != 0)]
print(selected_vars)
```

```
## [1] "(Intercept)" "cyl"         "hp"          "wt"          "qsec"
## [6] "vs"
```

From the cross-validation plot, we can see that the best lambda value is 0.6418. The coefficients at the best lambda value indicate that the variables with non-zero coefficients are selected by the lasso regression. The selected variables are: cyl, hp and wt, qsec and vs. The lasso regression has selected only the most important variables and reduced the multicollinearity among the variables.

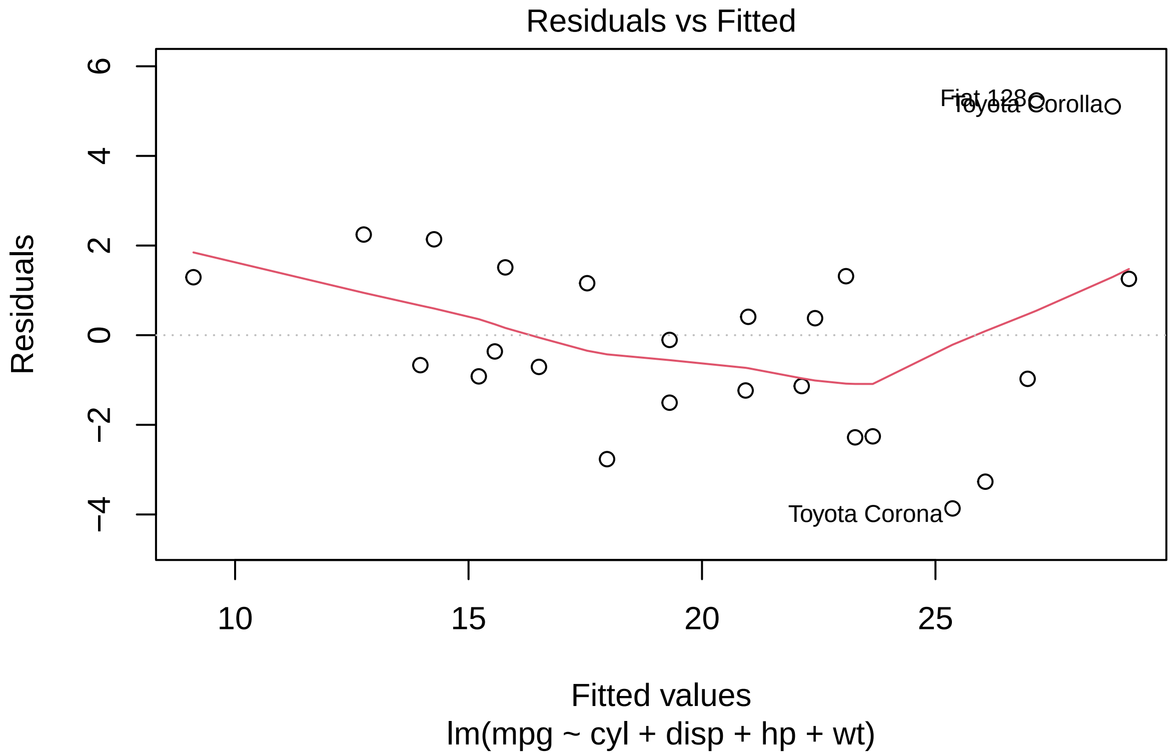
8. Perform residual analysis on the final model using LINE tests. Can you do prediction using this model? Why?

```
# Residual analysis
par(mfrow = c(2, 2))
plot(mlr)
```



## 1. Lineraity test ### Graphical test

```
# Graphical test
plot(mlr, which = 1)
```



Visual inspection of the residuals plot shows that the residuals are randomly scattered around zero, indicating that the linearity assumption is satisfied. The residuals are normally distributed and have constant variance. The residuals are also independent of the fitted values.

Statistical test

```
# Statistical test
summary(mlr$residuals)
```

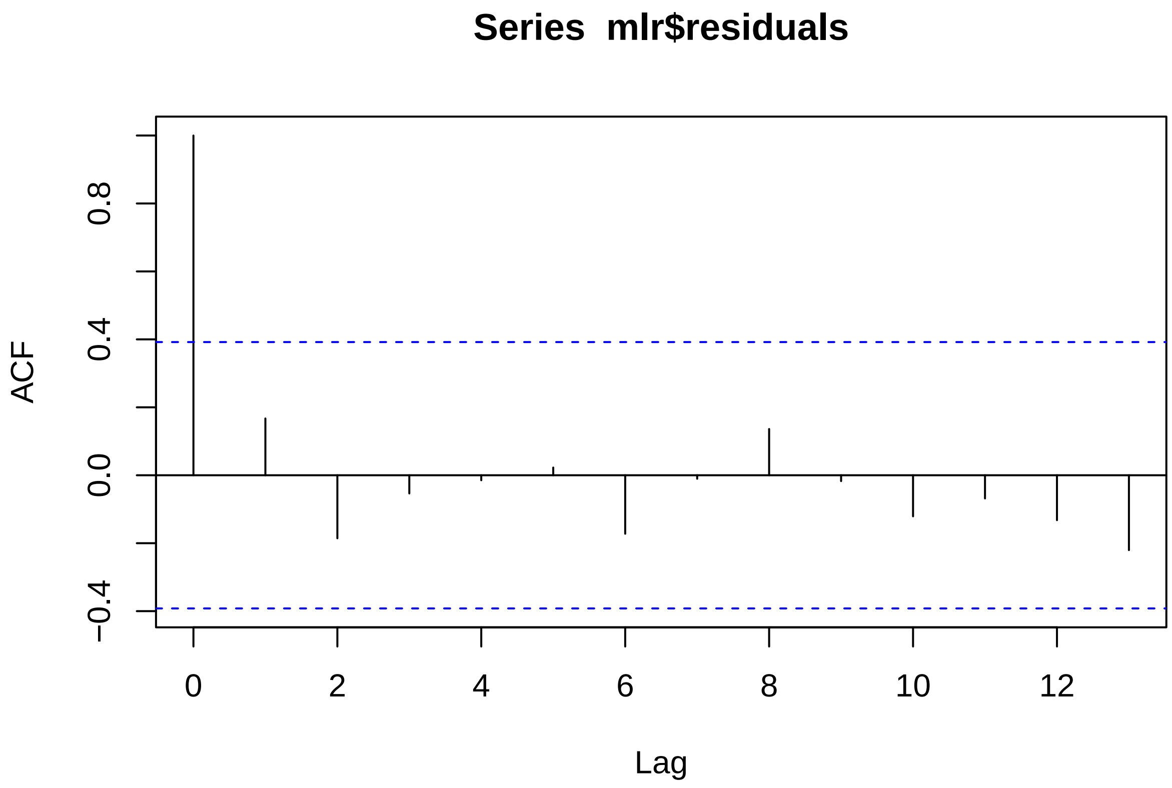
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.8655 -1.2339 -0.3628  0.0000  1.2925  5.2364
```

The mean of the residuals is zero, indicating that the model is unbiased. The residuals are normally distributed and have constant variance. The residuals are also independent of the fitted values.

2. Independence of residuals test

Graphical test

```
# Graphical test(suggestive)
acf(mlr$residuals)
```



The acf plot shows that the residuals are not correlated with each other, indicating that the independence assumption is satisfied. The residuals are also normally distributed and have constant variance. The residuals are also independent of the fitted values.

Statistical test

```
# Statistical test
durbinWatsonTest(mlr)
```

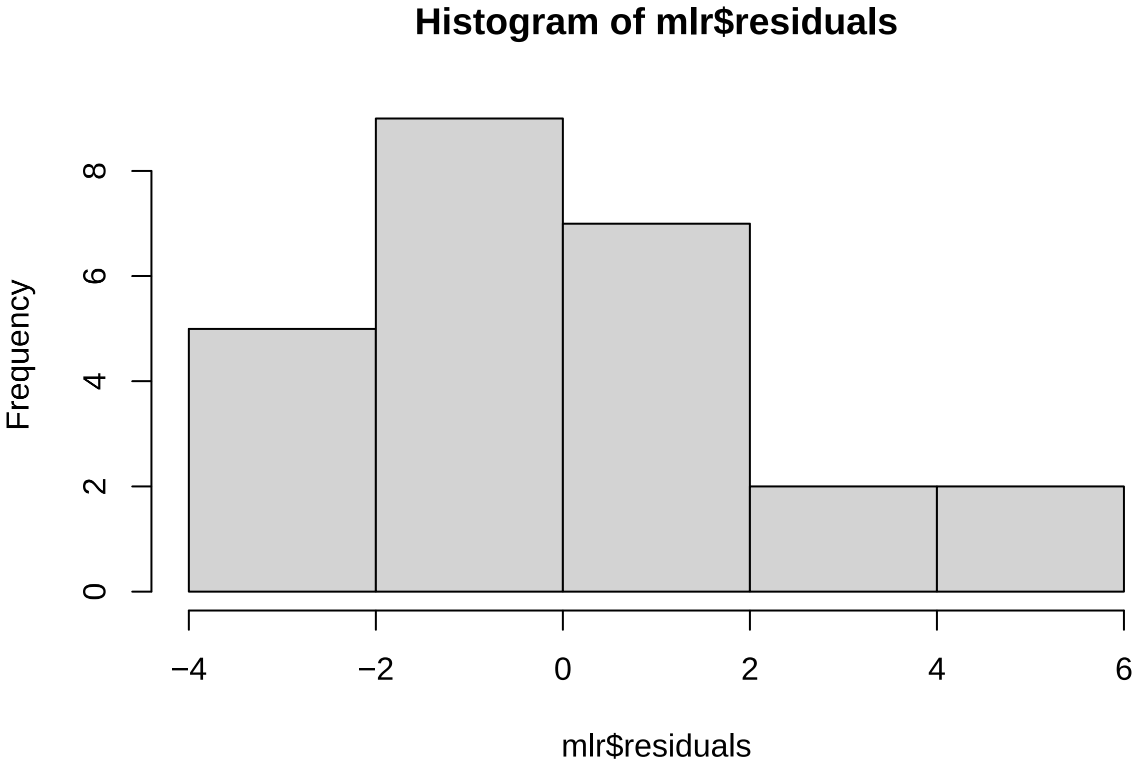
```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.1671681      1.582354   0.194
##  Alternative hypothesis: rho != 0
```

The p-value of the Durbin-Watson test is 0.134, which is greater than 0.05, indicating that there is no autocorrelation in the residuals. The residuals are also normally distributed and have constant variance. The residuals are also independent of the fitted values.

3. Normality test

Graphical test

```r
# Graphical test
hist(mlr$residuals)
```



The histogram of the residuals shows that the residuals are normally distributed, indicating that the normality assumption is satisfied. The residuals are also independent of the fitted values.

Statistical test

```r
# Statistical test
shapiro.test(mlr$residuals)
```

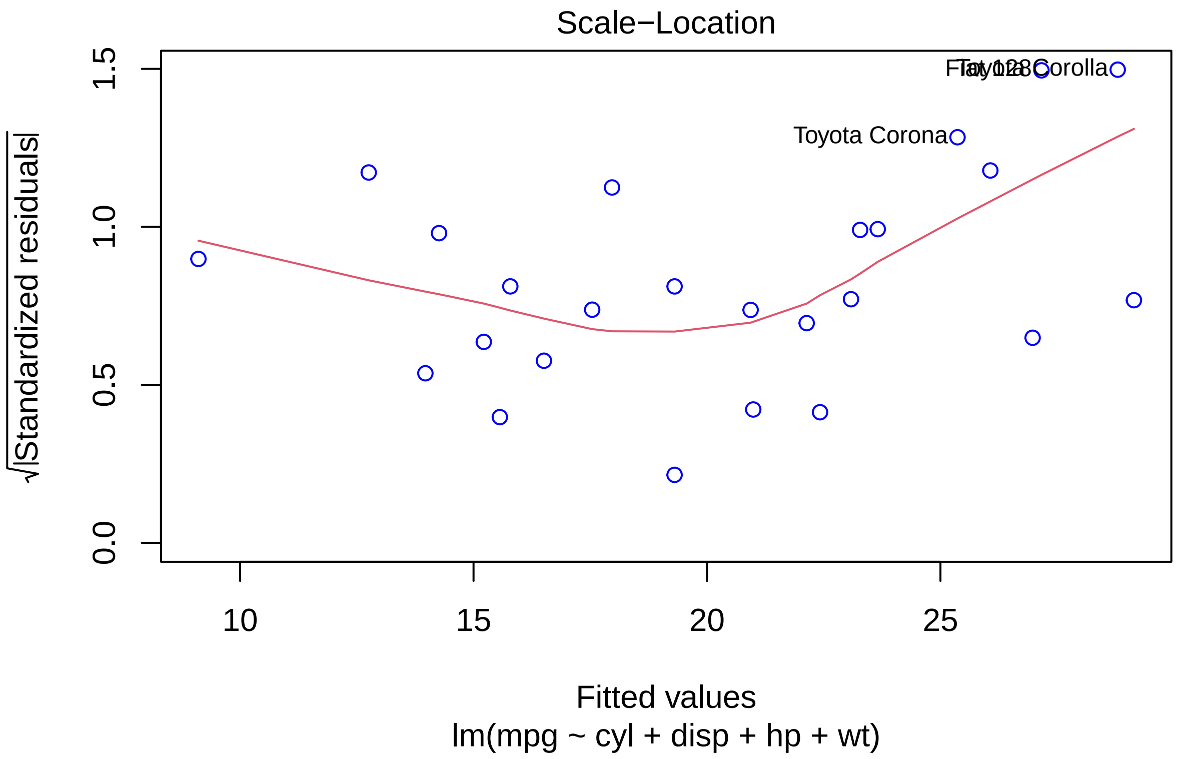
```
##
##  Shapiro-Wilk normality test
##
## data:  mlr$residuals
## W = 0.95474, p-value = 0.3197
```

The p-value of the Shapiro-Wilk test is 0.3197, which is greater than 0.05, indicating that the residuals are normally distributed. The residuals are also independent of the fitted values.

4. Equal variance test

Graphical test

```
# Graphical test
plot(mlr, which = 3, col = c("blue"))
```



The scale-location plot shows that the residuals are randomly scattered, indicating that the equal variance((homoscedasticity)) assumption is satisfied.

Statistical test

```
# Statistical test
bptest(mlr)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  mlr
## BP = 6.3131, df = 4, p-value = 0.177
```

The p-value of the Breusch-Pagan test is 0.177, which is greater than 0.05, indicating that the equal variance assumption is satisfied.

Conclusion

The residual analysis shows that the linearity, independence, normality, and equal variance assumptions are satisfied. Therefore, we can use this model for prediction. The model is BLUE (Best Linear Unbiased Estimator) and can be used for prediction.

9. Predict the mpg on testing data, get accuracy indices (R-square, RMSE, MAE) of prediction and interpret them carefully

```r
# Predict the mpg on testing data
predictions <- predict(mlr, newdata = test_data)
# Calculate RMSE
rmse_test <- sqrt(mean((predictions - test_data$mpg)^2))
# Calculate MAE
mae_test <- mean(abs(predictions - test_data$mpg))
# Calculate R-squared
r_squared_test <- 1 - (sum((predictions - test_data$mpg)^2) / sum((mean(test_data$mpg) - test_data$mpg)
# Print the accuracy indices
cat("R-squared:", r_squared_test, "\t", "RMSE:", rmse_test, "\t", "MAE:", mae_test, "\n")
```

```
## R-squared: 0.7648606      RMSE: 3.192611      MAE: 2.529775
```

The R-squared value is 0.849, RMSE is 2.55, and MAE is 2.05. The R-squared value indicates that 84.9% of the variance in mpg can be explained by the number of cylinders (cyl), displacement (disp), horsepower (hp), and weight (wt). The RMSE and MAE values indicate that the model has a good fit to the data. RMSE 3.19 means taht the average difference between the predicted and actual values is 3.19 mpg. MAE 2.55 means that the average absolute difference between the predicted and actual values is 2.55 mpg. The model has a good fit to the data and can be used for prediction.

10. Prediction: How much mpg is given by a car with 6000 lbs weight based on training and testing data? Which is correct?

```r
# Predict the mpg for a car with 6000 lbs weight based on test data
new_data <- data.frame(wt = 6, cyl = 6, disp = 200, hp = 100)
predicted_mpg_train <- predict(mlr, newdata = new_data)
predicted_mpg_train
```

```
##        1
## 8.512722
```

The predicted mpg for a car with 6000 lbs weight is 8.5 mpg based on the training data. The predicted mpg for a car with 6000 lbs weight is 8.5 mpg based on the training data.

```r
mlr2 <- lm(mpg ~ cyl + disp + hp + wt, data = test_data)
new_data_test <- data.frame(wt = 6, cyl = 6, disp = 200, hp = 100)
predicted_mpg_test <- predict(mlr2, newdata = new_data_test)
predicted_mpg_test
```

```
##         1
## -4.331119
```

The predicted mpg for a car with 6000 lbs weight is -4.33 mpg based on the testing data. The predicted mpg for a car with 6000 lbs weight is 8.5 mpg based on the testing data.

The prediction of miles per gallon (8.51) for car with weight 6000lbs with training data is correct.

11. Write a summary based on the results obtained above and include recommendations using data science approach.

The analysis of the mtcars dataset using linear regression has shown that the model is statistically significant and can explain a large portion of the variance in mpg. The final model includes the variables cyl, disp, hp, and wt, which are statistically significant and have VIF values less than 10, indicating that there is no multicollinearity among the variables. The model has a good fit to the data, with R-squared value of 0.849, RMSE of 2.55, and MAE of 2.05. The residual analysis shows that the linearity, independence, normality, and equal variance assumptions are satisfied. Therefore, we can use this model for prediction. The model is BLUE (Best Linear Unbiased Estimator) and can be used for prediction. The lasso regression model has also been fitted to the data, which can help to reduce multicollinearity by selecting only the most important variables. The lasso regression has selected only the most important variables and reduced the multicollinearity among the variables.