Final Project

Amisha Patel

2023-08-07

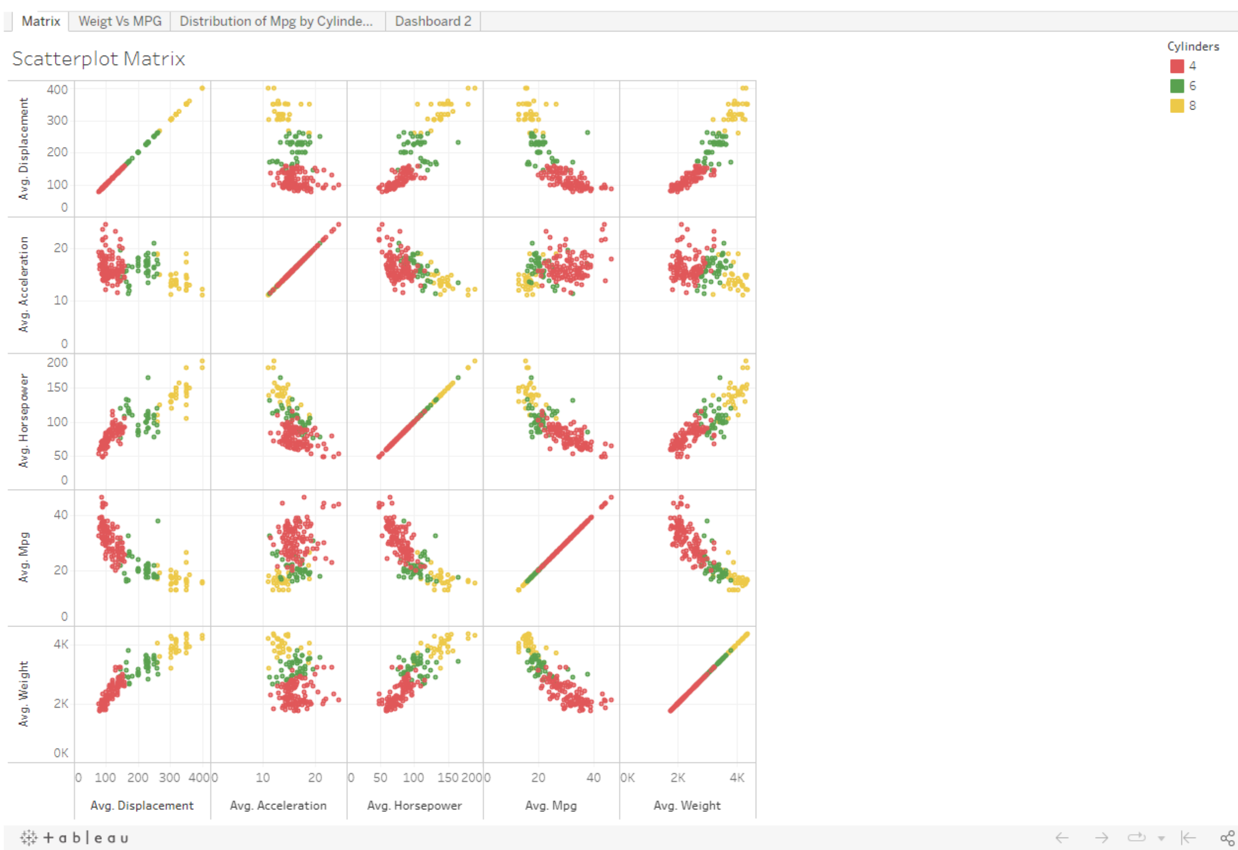
Visualizations using Tableau Public

I have generated a series of informative visualizations and assembled them into an interactive dashboard. You can access the visualizations through the Tableau Public link.

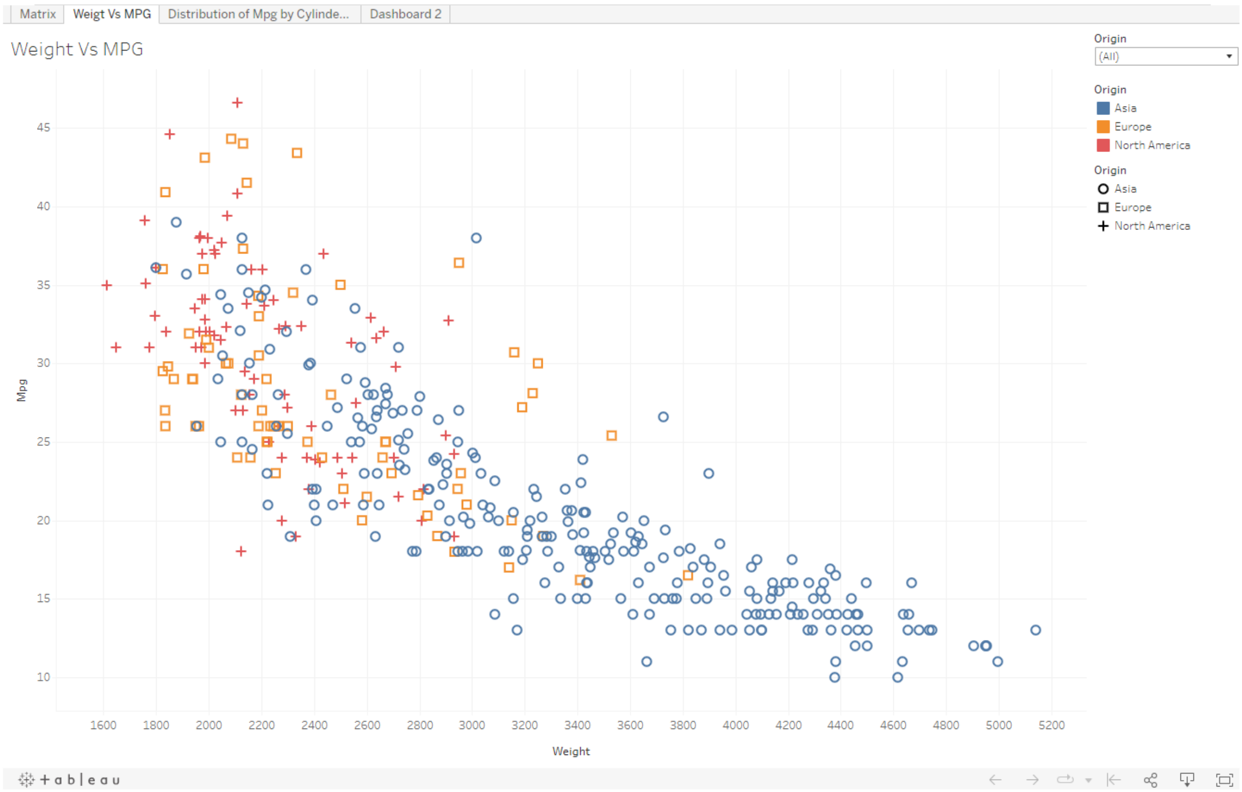
https://public.tableau.com/app/profile/amisha.patel7081/viz/Auto-Mpg/Dashboard2?publish=yes

The visualizations encompass a scatterplot matrix, a scatterplot depicting Weight versus MPG, and a box plot illustrating the distribution of MPG based on Cylinder count.To provide you with a comprehensive understanding, I've also included screenshots of these visualizations.

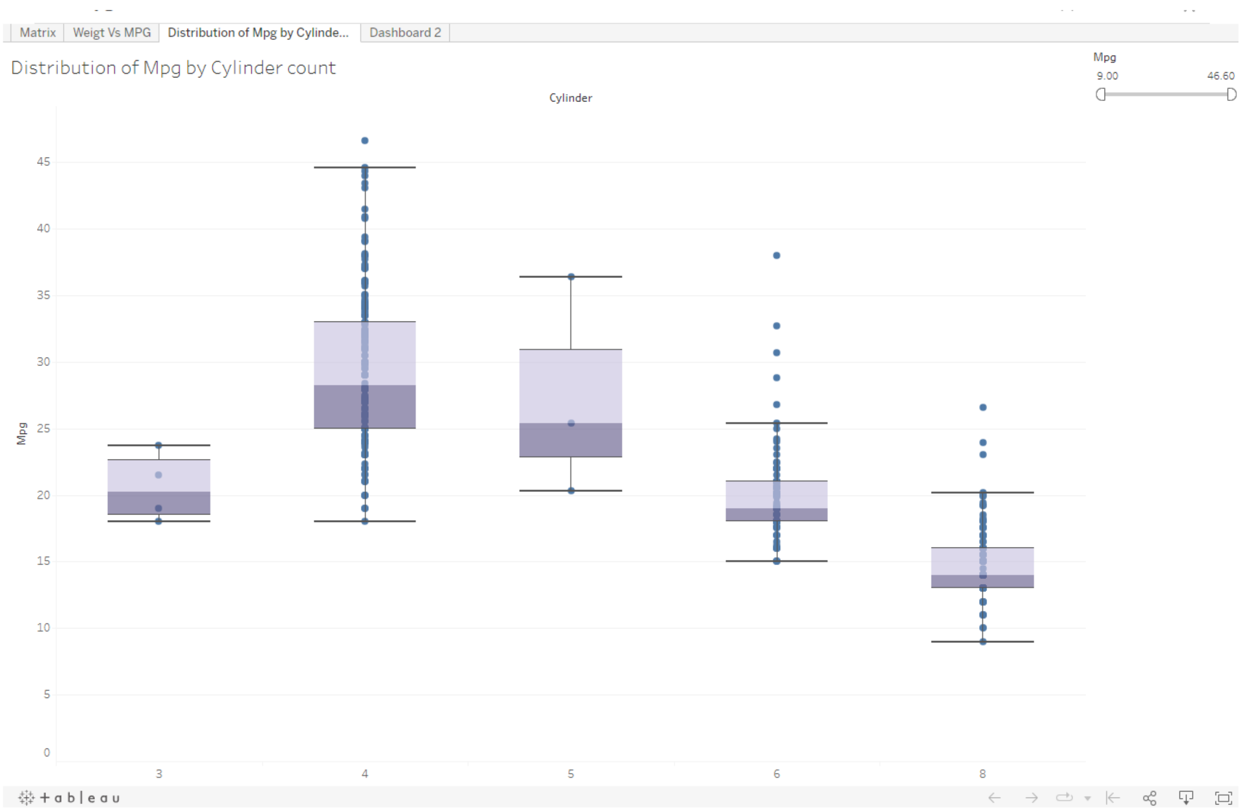
[1] Scatterplot Matrix: This visualization likely showcases a grid of scatterplots. Each scatterplot within the grid corresponds to a pair of variables from your dataset. It's a powerful way to visualize the relationships and correlations between multiple variables simultaneously.



[2] **Weight vs. MPG Scatterplot:** This scatterplot specifically focuses on the relationship between two variables: the weight of vehicles and their miles per gallon (MPG) efficiency. Scatterplots are great for identifying trends or patterns in data points between two continuous variables.

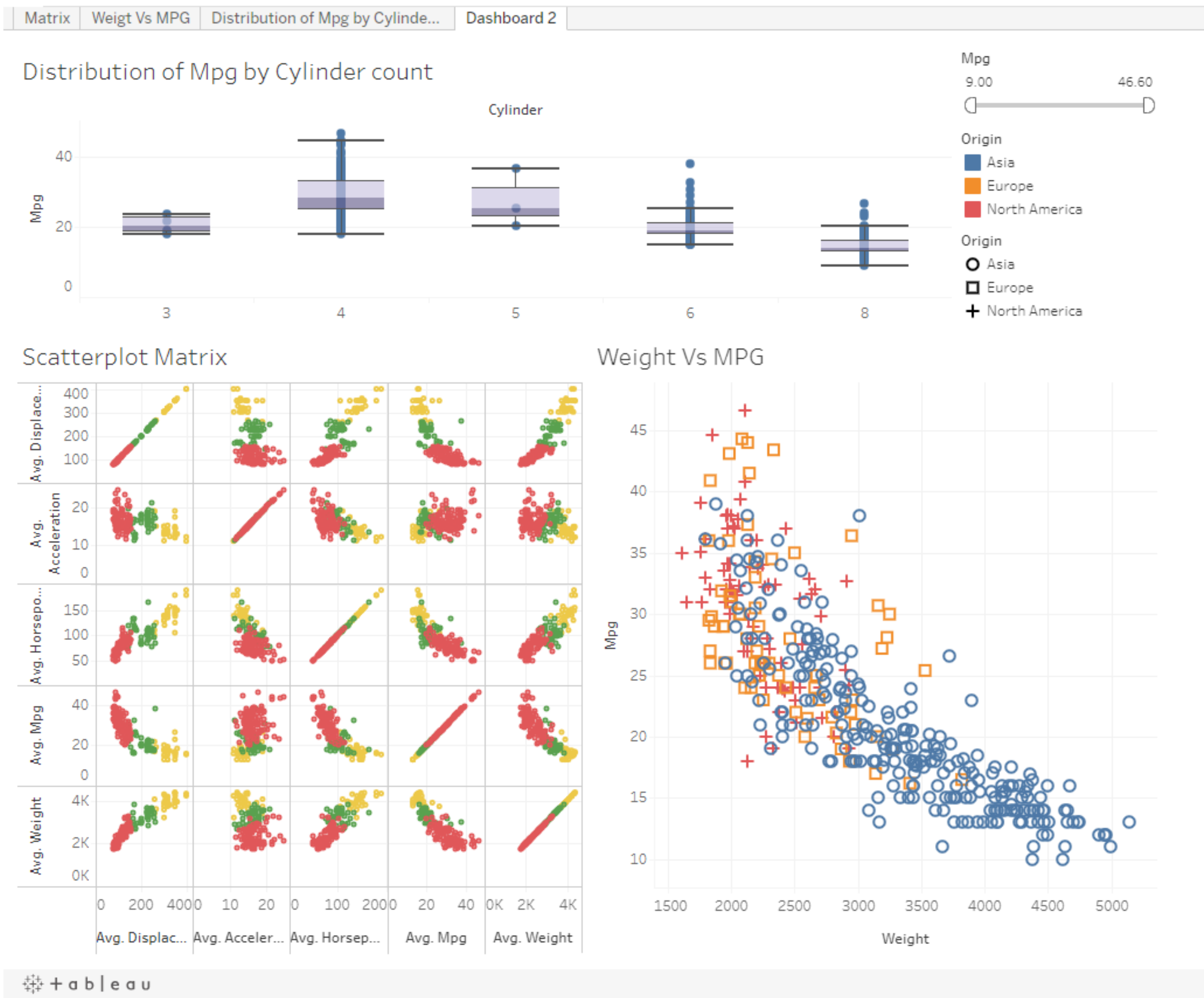


[3] **Distribution of MPG by Cylinder Count (Box Plot):** This visualization is a box plot that showcases the distribution of MPG values based on different levels of cylinder counts. Box plots are excellent for understanding the spread, central tendency, and potential outliers within different categories or groups.



---

[4] Dashboard: The dashboard provides a comprehensive view of the relationships between variables, the impact of vehicle weight on MPG, and the MPG distribution based on cylinder counts. By presenting these visualizations together, viewers can make more informed insights and comparisons from the data.



Visual plots and Charts using R

```r
# Load the data
data <- read.csv("./auto-mpg.csv", na.strings = c("?", ""))
# Here remove ? as value from table

# Check the structure of the data (column names and data types)
str(data)
```

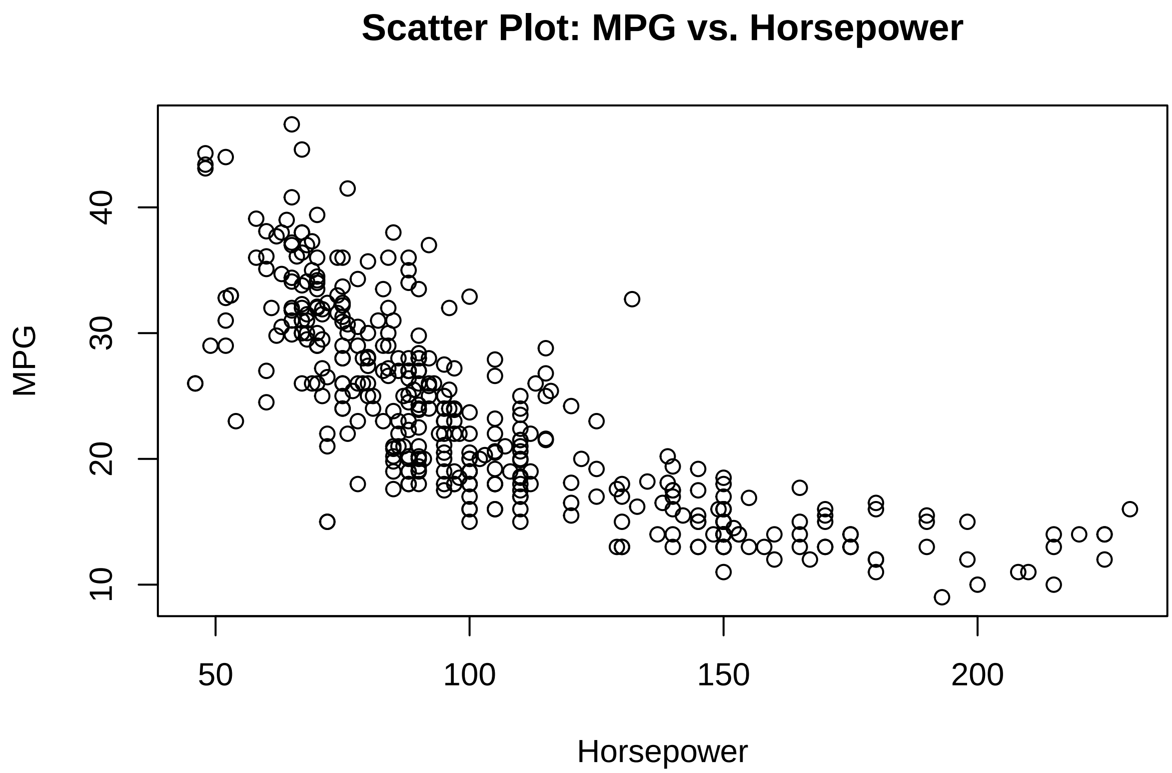
```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinder    : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : int  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
```

```
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel
```

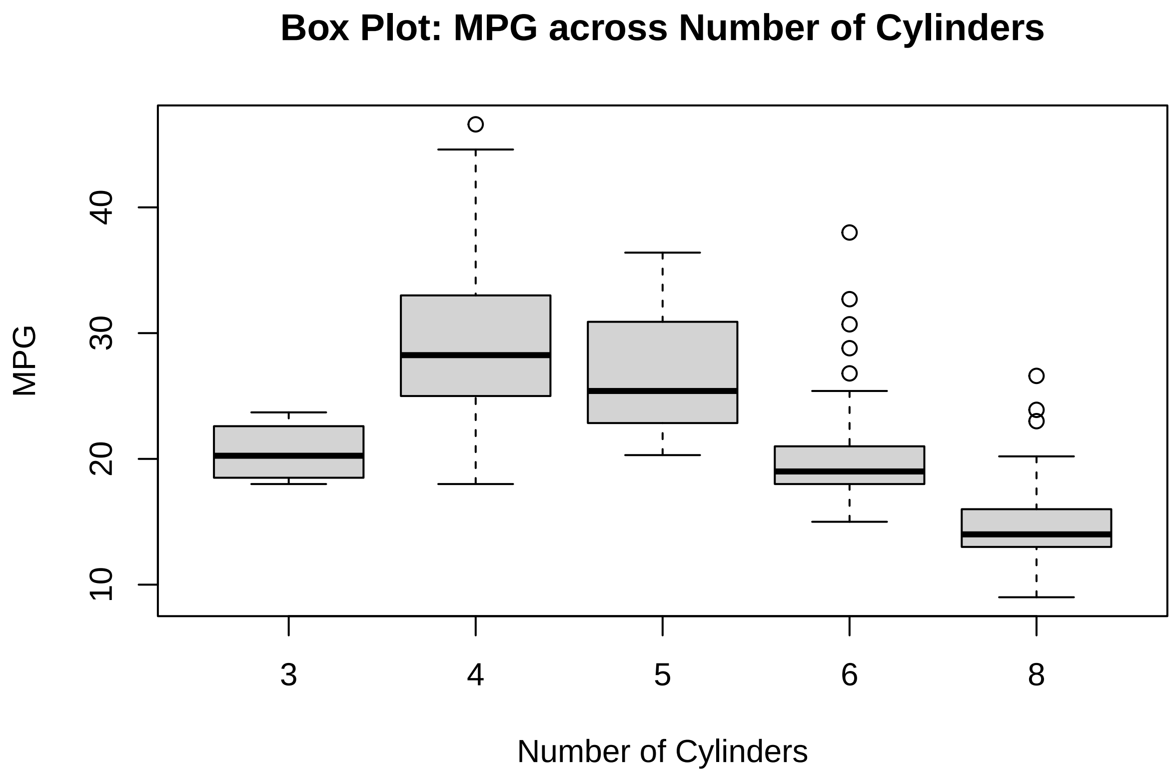
```r
# Check the first few rows of the data
head(data)
```

```
##   mpg cylinder displacement horsepower weight acceleration model.year origin
## 1  18        8          307        130   3504         12.0         70      1
## 2  15        8          350        165   3693         11.5         70      1
## 3  18        8          318        150   3436         11.0         70      1
## 4  16        8          304        150   3433         12.0         70      1
## 5  17        8          302        140   3449         10.5         70      1
## 6  15        8          429        198   4341         10.0         70      1
##                    car.name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

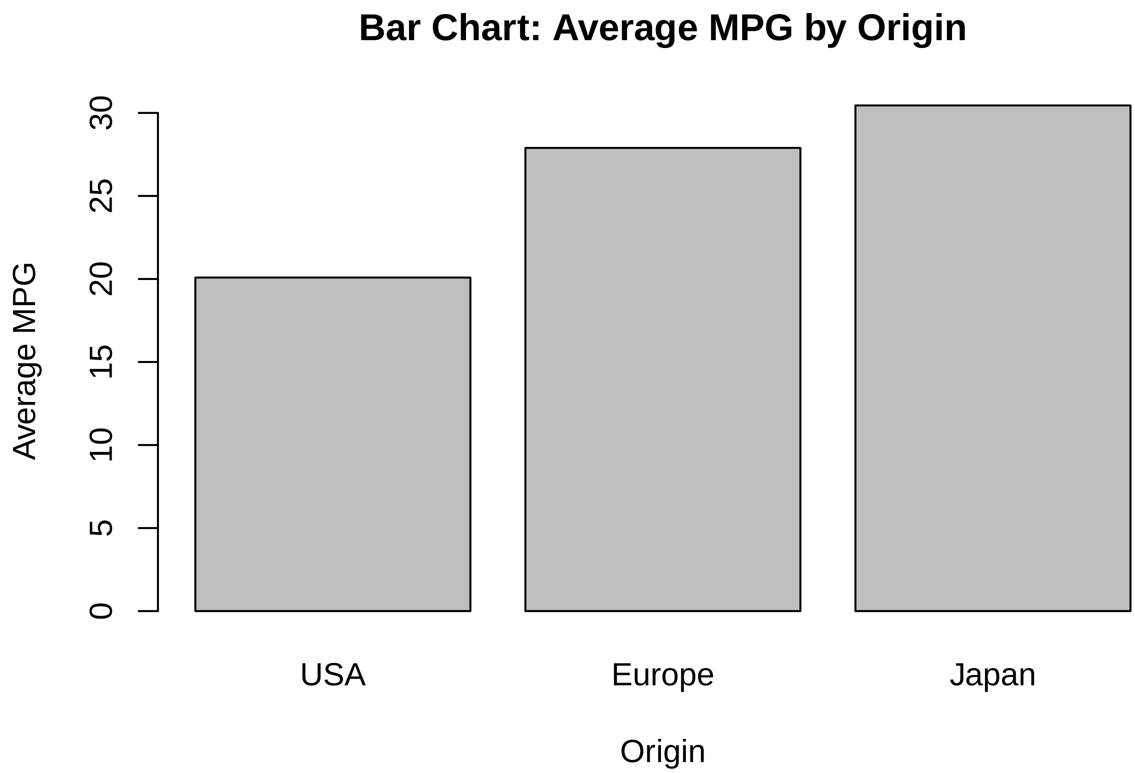
```r
# Scatter Plot: mpg vs. horsepower
# The scatter plot shows the relationship between mpg and horsepower.
# It allows us to see if there's any clear pattern or correlation between the two variables.
plot(data$horsepower, data$mpg, xlab = "Horsepower", ylab = "MPG"
     , main = "Scatter Plot: MPG vs. Horsepower")
```



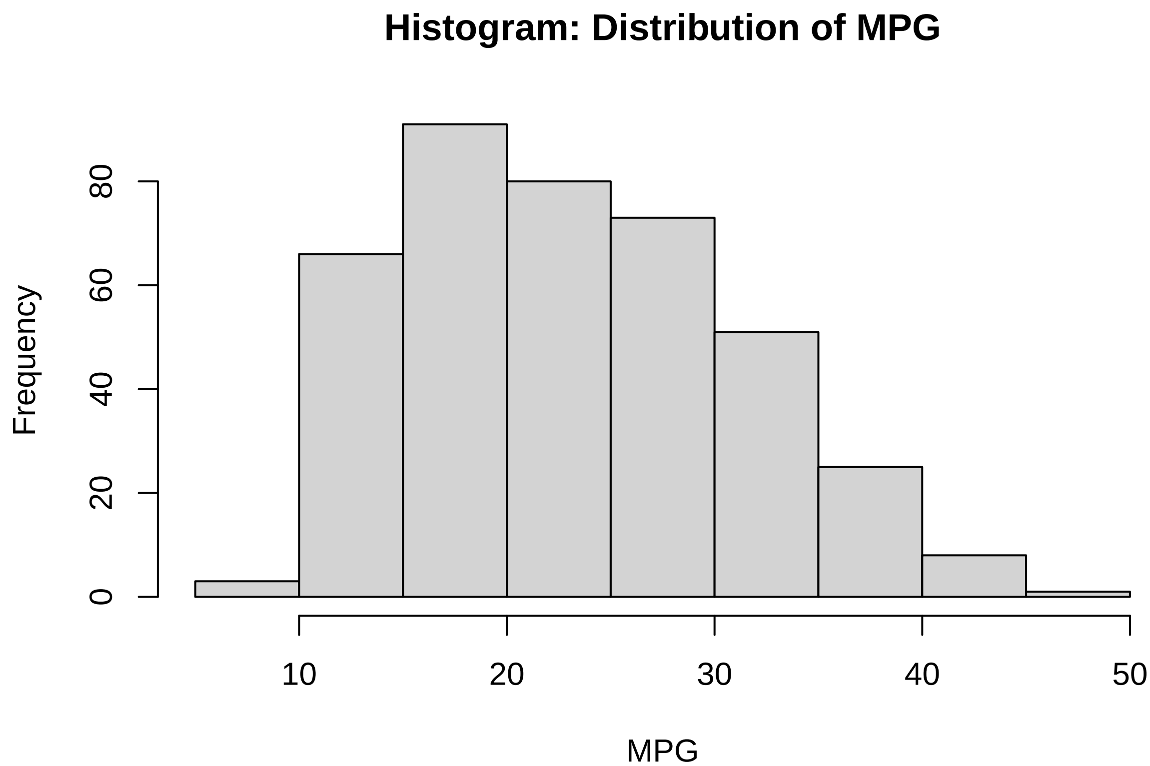

```r
# Box Plot: mpg across different number of cylinders
# The box plot displays the distribution of mpg across different numbers of cylinders.
# It helps us understand how the number of cylinders impacts the fuel efficiency of the vehicles.
boxplot(mpg ~ cylinder, data = data, xlab = "Number of Cylinders", ylab = "MPG"
        , main = "Box Plot: MPG across Number of Cylinders")
```



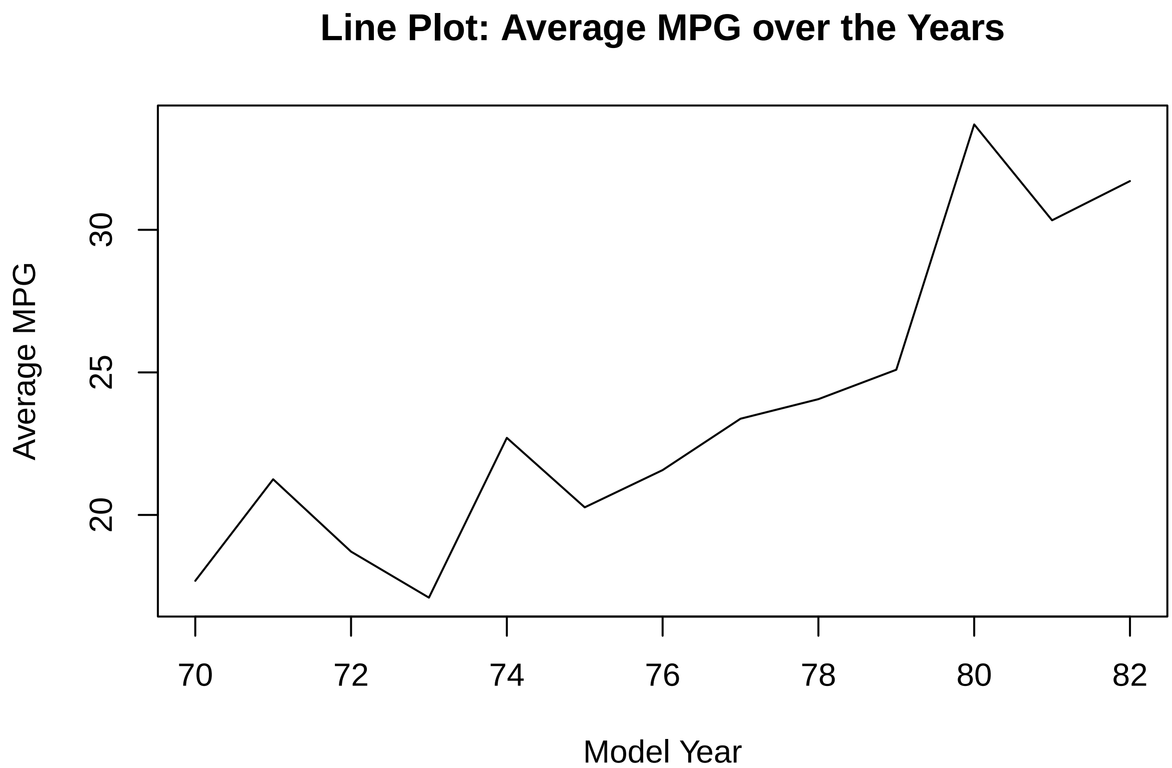

```r
# Bar Chart: Average mpg by origin
# The bar chart illustrates the average mpg for each origin (USA, Europe, and Japan),
# allowing us to compare the fuel efficiency of cars from different regions.
avg_mpg_by_origin <- tapply(data$mpg, data$origin, mean)
barplot(avg_mpg_by_origin, names.arg = c("USA", "Europe", "Japan"), xlab = "Origin"
        , ylab = "Average MPG", main = "Bar Chart: Average MPG by Origin")
```




```r
# Histogram: Distribution of mpg
# The histogram shows the distribution of mpg,
# providing insights into how fuel efficiency is distributed across the automobile models.
hist(data$mpg, breaks = "FD", xlab = "MPG", ylab = "Frequency"
     , main = "Histogram: Distribution of MPG")
```



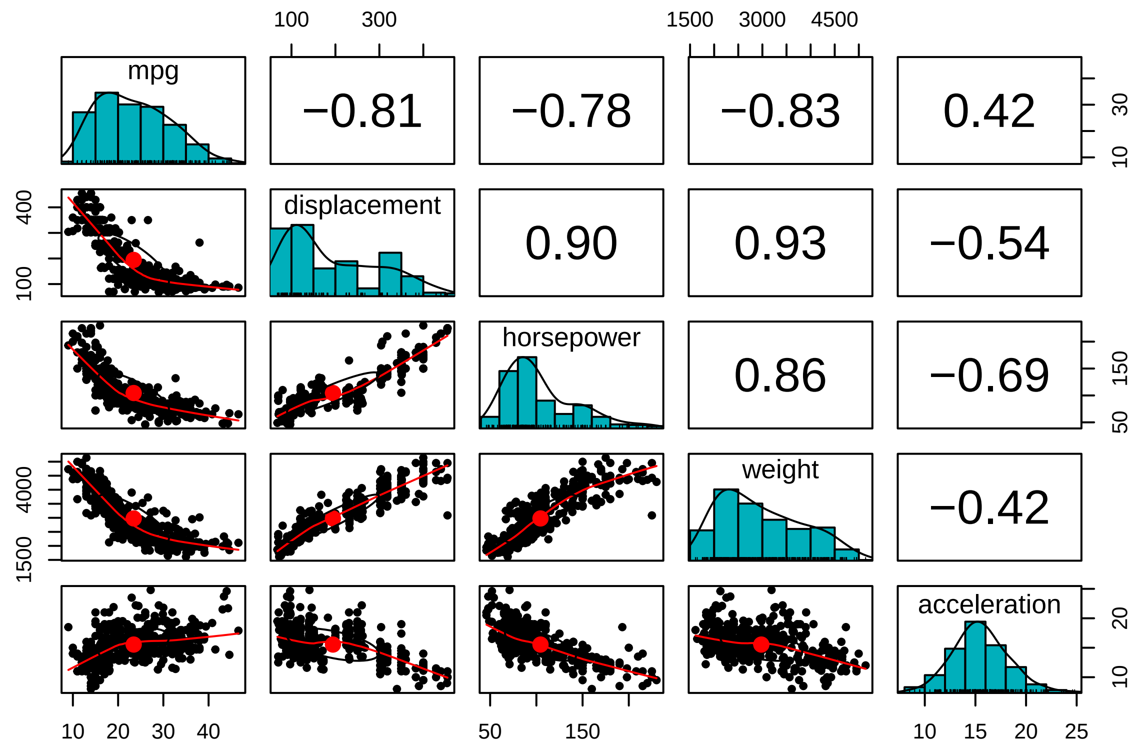

```r
# Line Plot: Average mpg over the years
# The line plot demonstrates the trend in average mpg over the years,
# helping us identify any improvements in fuel efficiency over time.
avg_mpg_by_year <- tapply(data$mpg, data$model.year, mean)
plot(names(avg_mpg_by_year), avg_mpg_by_year, type = "l", xlab = "Model Year"
     , ylab = "Average MPG", main = "Line Plot: Average MPG over the Years")
```




```r
# Take subset of data
subset_data <- subset(data, select = c(mpg, displacement, horsepower, weight, acceleration))
subset_data <- na.omit(subset_data)

# Show summary statistics of the subset data
summary(subset_data)
```

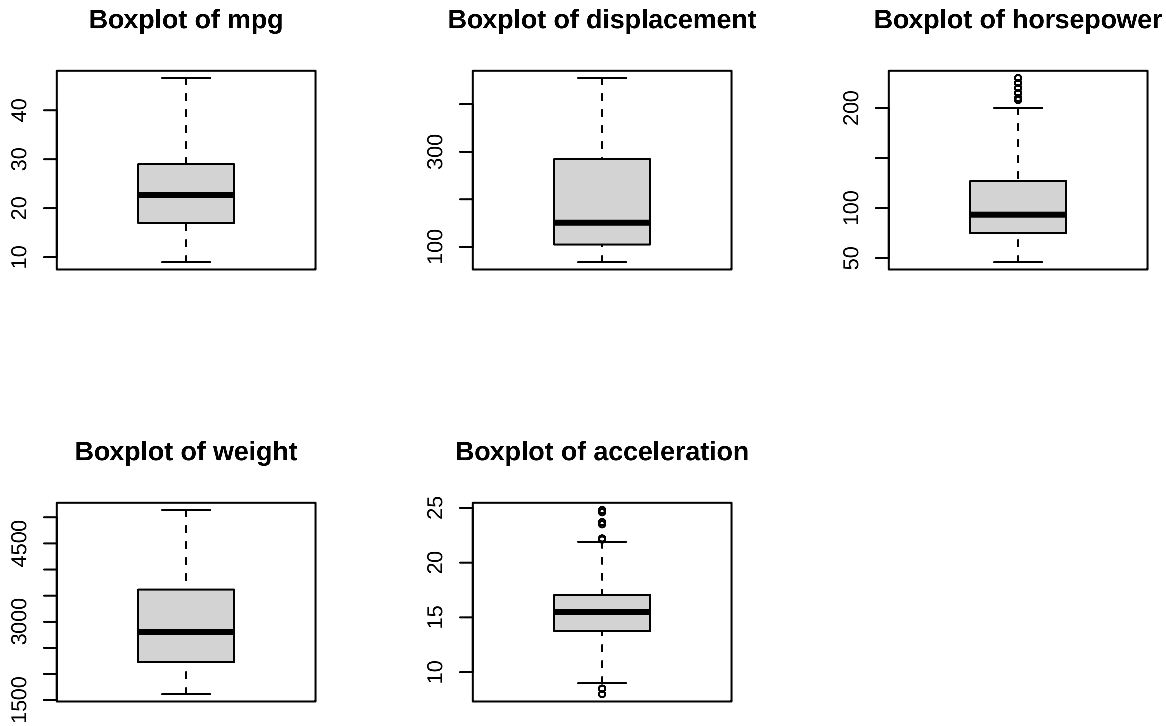
```
##       mpg         displacement     horsepower        weight      acceleration
##  Min.   : 9.00   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
##  1st Qu.:17.00   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78
##  Median :22.75   Median :151.0   Median : 93.5   Median :2804   Median :15.50
##  Mean   :23.45   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54
##  3rd Qu.:29.00   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02
##  Max.   :46.60   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80
```

```r
# Create the scatterplot matrix with correlation ellipses and histograms
pairs.panels(subset_data, method = "pearson", hist.col = "#00AFBB", density = TRUE, ellipses = TRUE)
```



The pairs.panels() function generates a matrix of scatter plots This output shows three things: the correlation between variables, the scatter plot that shows how the variables relate to each other, and the histograms that show how skewed the data are. We see that displacement and displacement are strongly correlated, and negatively correlated to the MPG. We also see that there is a multicollinearity between the independent variables.

```
# Boxplots are useful for understanding the central tendency, spread,
# and presence of outliers in each variable.
par(mfrow=c(2,3))
for (i in names(subset_data)) {
  boxplot(subset_data[, i], main = paste("Boxplot of", i))
}
```



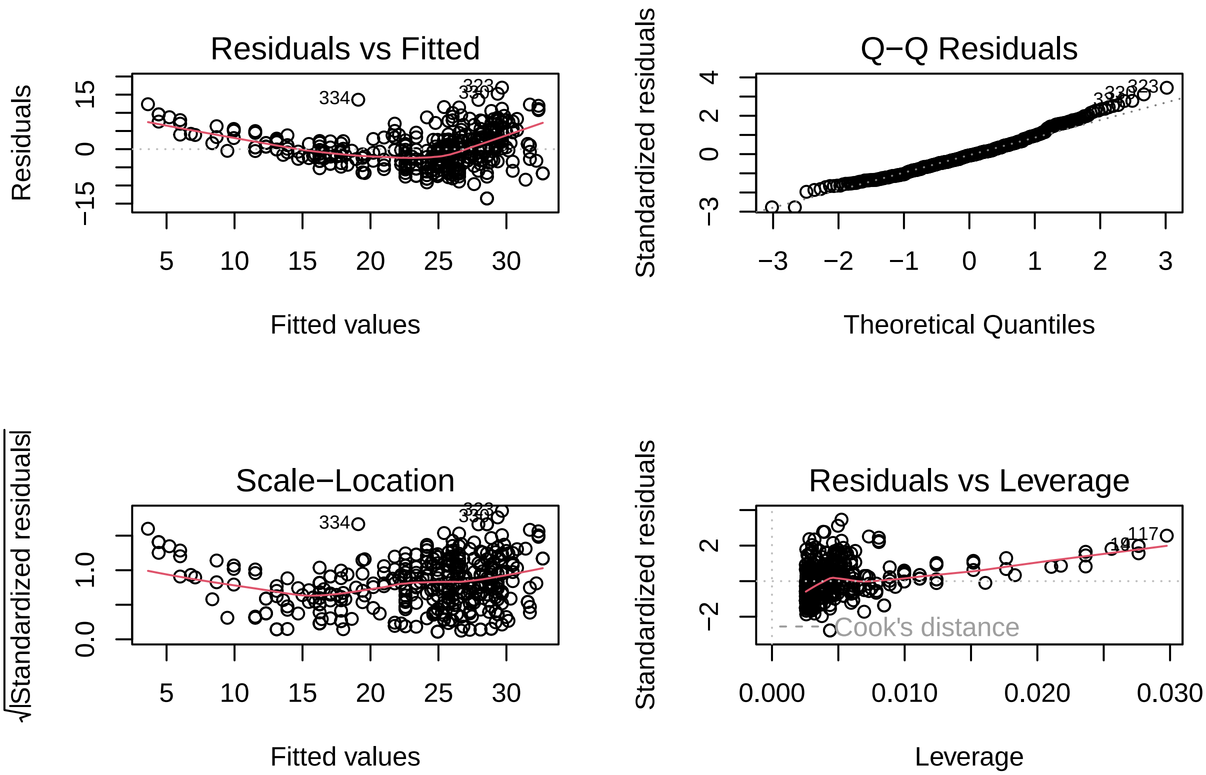
Simple linear regression and multiple linear regression

```
# First, make sure 'subset_data_First' contains only the first 300 rows
subset_data_First <- subset_data[1:300, ]

# Simple Linear Regression between mpg and different variables
slr_hors <- lm(mpg ~ horsepower, data = subset_data)
slr_dis <- lm(mpg ~ displacement, data = subset_data)
slr_wie <- lm(mpg ~ weight, data = subset_data)
slr_acc <- lm(mpg ~ acceleration, data = subset_data)

# Multiple Linear Regression
mlr <- lm(mpg ~ horsepower + displacement + weight + acceleration, data = subset_data)

# Extracting regression coefficients and summary statistics
slr_summary_hors <- summary(slr_hors)
slr_summary_dis <- summary(slr_dis)
slr_summary_wie <- summary(slr_wie)
slr_summary_acc <- summary(slr_acc)
mlr_summary <- summary(mlr)

# Extracting required information
slr_print_hors <- paste("Simple Linear Regression of mpg~horsepower:\n",
```

```r
                    "Multiple R-squared:", slr_summary_hors$r.squared, "\n",
                    "Adjusted R-squared:", slr_summary_hors$adj.r.squared, "\n",
                    "Complete Linear Regression equation:\n",
                    "mpg =", slr_hors$coefficient[1], "+", slr_hors$coefficient[2],
                    "* horsepower", "\n\n",
                    sep = "")
slr_print_dis <- paste("Simple Linear Regression of mpg~displacement:\n",
                    "Multiple R-squared:", slr_summary_dis$r.squared, "\n",
                    "Adjusted R-squared:", slr_summary_dis$adj.r.squared, "\n",
                    "Complete Linear Regression equation:\n",
                    "mpg =", slr_dis$coefficient[1], "+", slr_dis$coefficient[2],
                    "* displacement", "\n\n",
                    sep = "")
slr_print_wie <- paste("Simple Linear Regression of mpg~weight:\n",
                    "Multiple R-squared:", slr_summary_wie$r.squared, "\n",
                    "Adjusted R-squared:", slr_summary_wie$adj.r.squared, "\n",
                    "Complete Linear Regression equation:\n",
                    "mpg =", slr_wie$coefficient[1], "+", slr_wie$coefficient[2],
                    "* weight", "\n\n",
                    sep = "")
slr_print_acc <- paste("Simple Linear Regression of mpg~acceleration:\n",
                    "Multiple R-squared:", slr_summary_acc$r.squared, "\n",
                    "Adjusted R-squared:", slr_summary_acc$adj.r.squared, "\n",
                    "Complete Linear Regression equation:\n",
                    "mpg =", slr_acc$coefficient[1], "+", slr_acc$coefficient[2],
                    "* acceleration", "\n\n",
                    sep = "")


mlr_print <- paste("Multiple Linear Regression:\n",
                    "Multiple R-squared:", mlr_summary$r.squared, "\n",
                    "Adjusted R-squared:", mlr_summary$adj.r.squared, "\n",
                    "Complete Linear Regression equation:\n",
                    "mpg =", mlr$coefficients[1], "+", mlr$coefficients[2], "* horsepower \n  +",
                    mlr$coefficients[3], "* displacement +", mlr$coefficients[4], "* weight \n  +",
                    mlr$coefficients[5], "* acceleration", "\n")

# Printing the results
cat(slr_print_hors)
```

```
## Simple Linear Regression of mpg~horsepower:
## Multiple R-squared:0.605948257889435
## Adjusted R-squared:0.6049378688071
## Complete Linear Regression equation:
## mpg =39.9358610211705+-0.157844733353653* horsepower
```

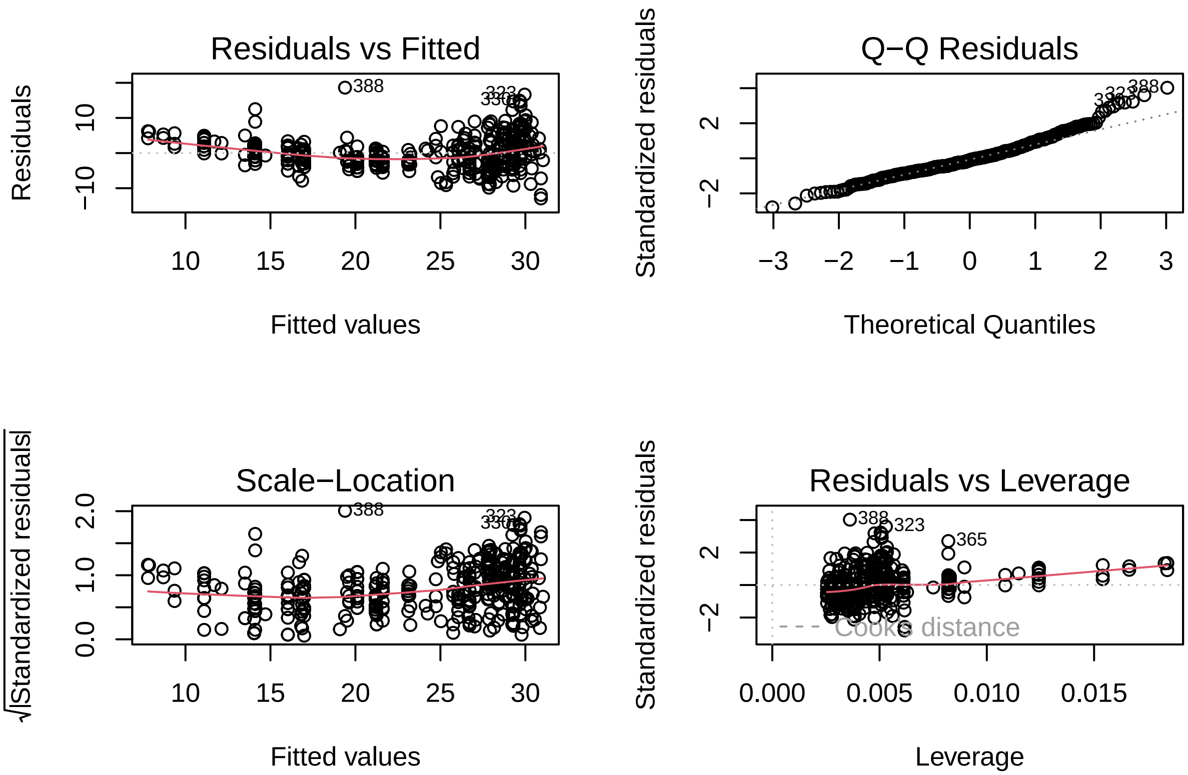
```r
# Horsepower Model
par(mfrow=c(2,2))
plot(slr_hors)
```




```r
cat(slr_print_dis)
```

```
## Simple Linear Regression of mpg~displacement:
## Multiple R-squared:0.648229400319304
## Adjusted R-squared:0.647327424422687
## Complete Linear Regression equation:
## mpg =35.1206359384039+-0.0600514278122062* displacement
```

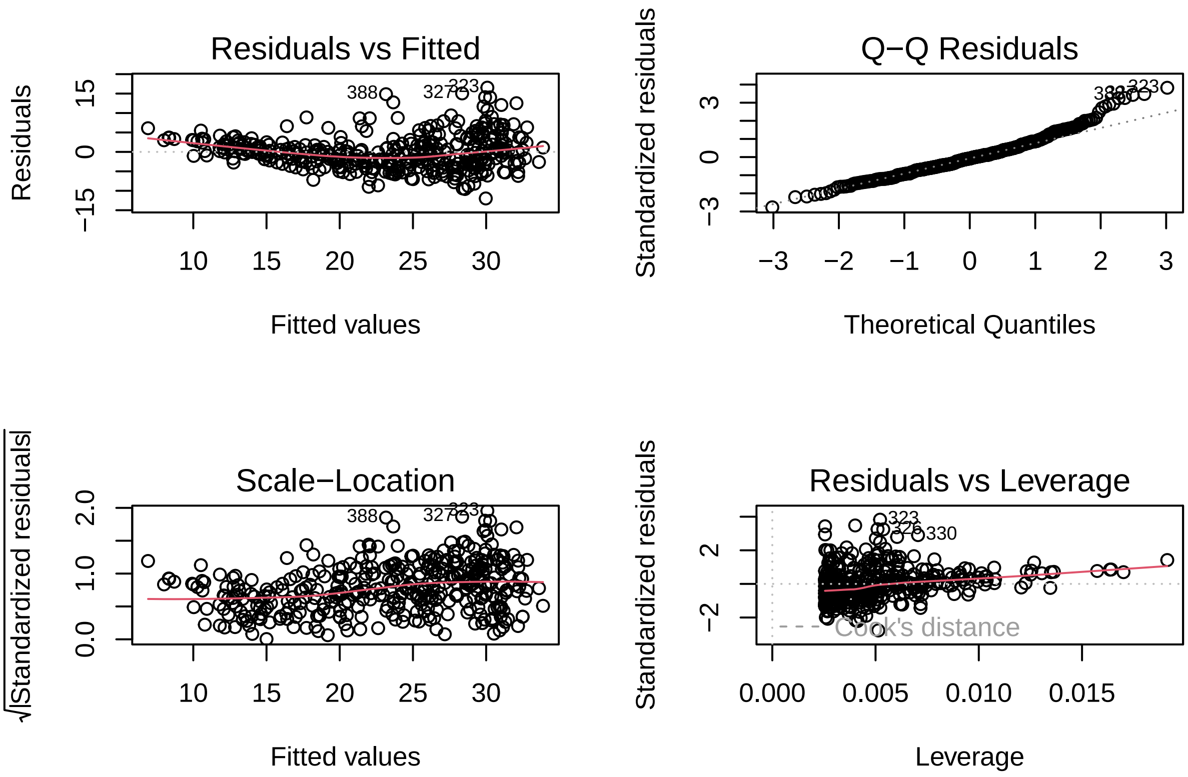
```r
#Displacement Model
par(mfrow=c(2,2))
plot(slr_dis)
```




```r
cat(slr_print_wie)
```

```
## Simple Linear Regression of mpg~weight:
## Multiple R-squared:0.692630433120625
## Adjusted R-squared:0.691842306026063
## Complete Linear Regression equation:
## mpg =46.2165245490176+-0.00764734253577959* weight
```

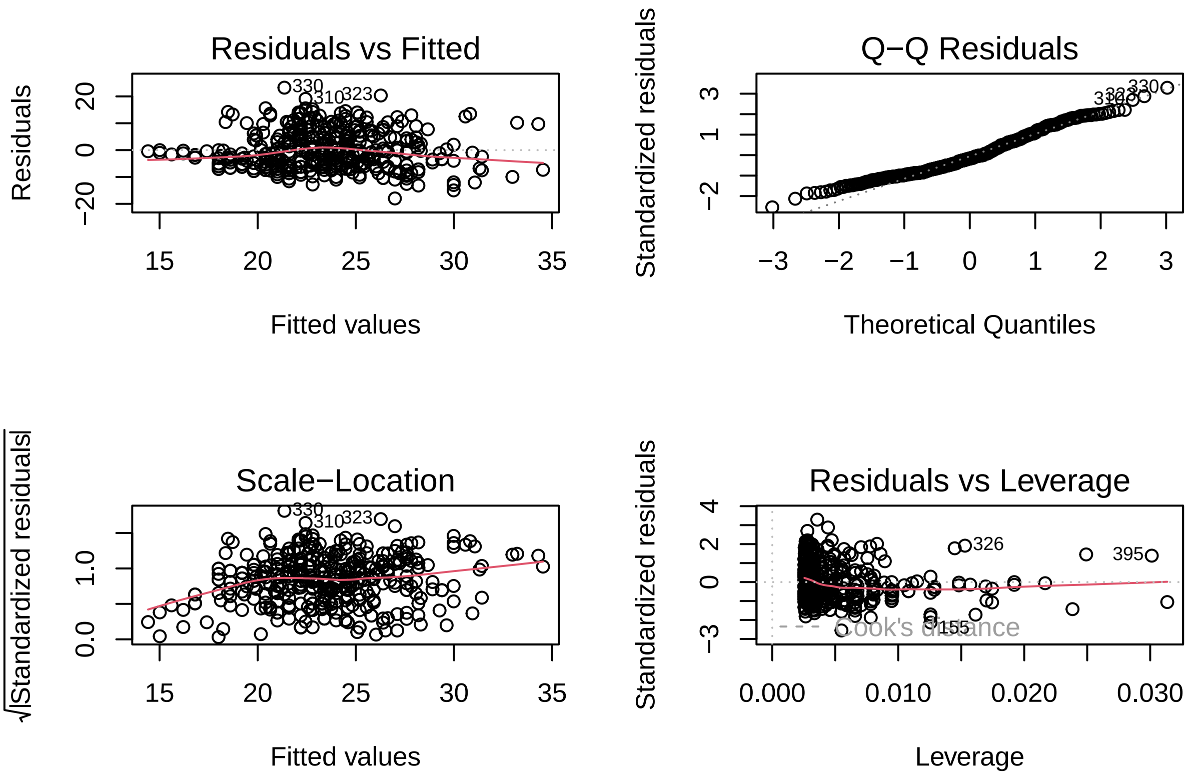
```r
# weight Model
par(mfrow=c(2,2))
plot(slr_wie)
```




```
cat(slr_print_acc)
```

```
## Simple Linear Regression of mpg~acceleration:
## Multiple R-squared:0.179207050156255
## Adjusted R-squared:0.177102452848963
## Complete Linear Regression equation:
## mpg =4.83324980484383+1.19762418773205* acceleration
```

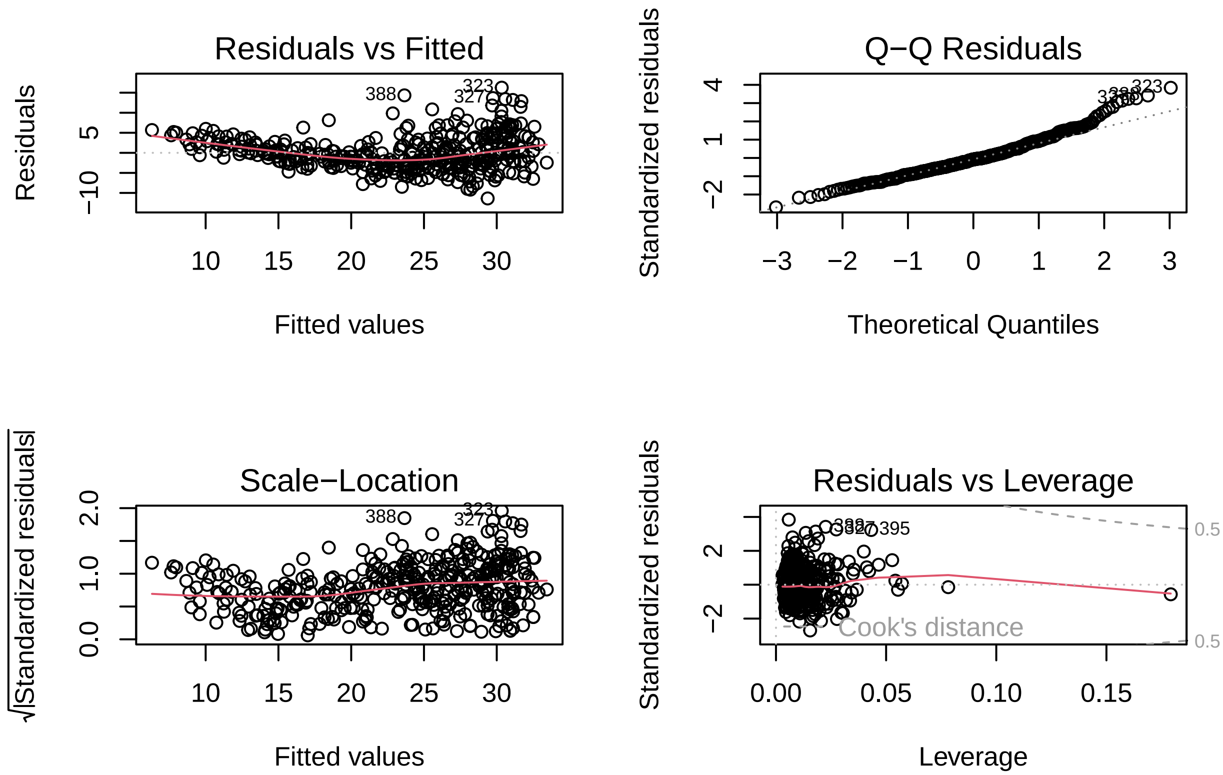
```
# Acceleration Model
par(mfrow=c(2,2))
plot(slr_acc)
```




```r
cat(mlr_print)
```

```
## Multiple Linear Regression:
##  Multiple R-squared: 0.70698118657199
##  Adjusted R-squared: 0.703952568345344
##  Complete Linear Regression equation:
##  mpg = 45.251139699335 + -0.0436077308860245 * horsepower
##   + -0.00600087098453362 * displacement + -0.00528050779763585 * weight
##   + -0.0231479993429443 * acceleration
```

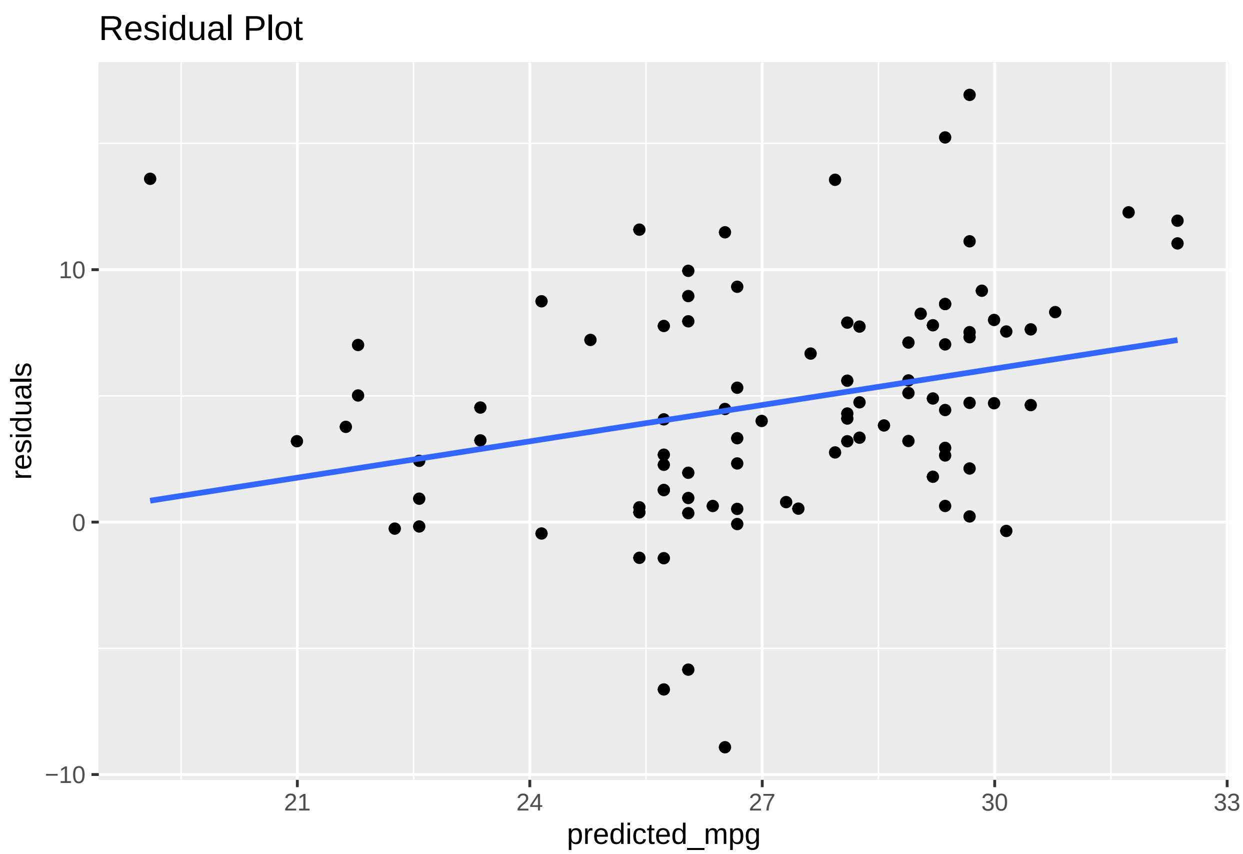
```r
# Multiple Model
par(mfrow=c(2,2))
plot(mlr)
```



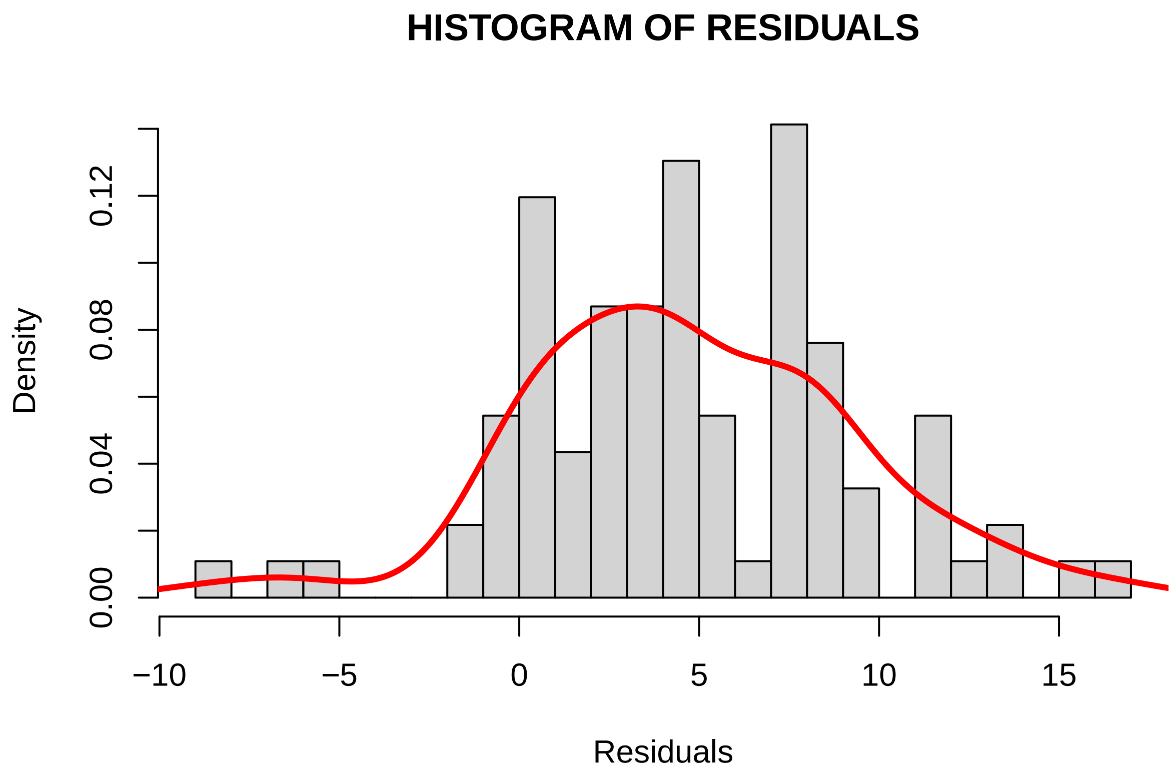
Predictions

```r
# First, make sure 'subset_data_Last' contains remaining 98 samples
subset_data_Last <- subset_data[301:398, ]

#HORSEMODEL

#Predict MPG for the remaining 98 samples
predicted_mpg <- predict(slr_hors, newdata = subset_data_Last)

# Calculate residuals
residuals <- subset_data_Last$mpg - predicted_mpg

residuals <- residuals[!is.na(residuals)]
predicted_mpg <- predicted_mpg[!is.na(predicted_mpg)]

# Create a residual plot
ggplot(data.frame(predicted_mpg, residuals), aes(predicted_mpg, residuals)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  labs(title="Residual Plot ")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
#Create a histogram of residuals
hist(residuals,prob=T,breaks=20,main="HISTOGRAM OF RESIDUALS",xlab="Residuals")
lines(density(residuals),col="red",lwd=3)
```




```
### Calculate and display the comparison #####
actual_mpg <- subset_data_Last$mpg
actual_mpg <- actual_mpg[!is.na(actual_mpg)]

comparison_df <- data.frame(Actual_MPG = actual_mpg, Predicted_MPG = predicted_mpg)
print(comparison_df)
```

```
##     Actual_MPG Predicted_MPG
## 303       34.5      28.88673
## 304       31.8      29.67595
## 305       37.3      29.04457
## 306       28.4      25.72984
## 307       28.8      21.78372
## 308       26.8      21.78372
## 309       33.5      25.72984
## 310       41.5      27.93966
## 311       38.1      30.46518
## 312       32.1      28.88673
## 313       37.2      29.67595
## 314       28.0      25.72984
## 315       26.4      26.04552
## 316       24.3      25.72984
## 317       19.1      25.72984
## 318       34.3      27.62397
## 319       29.8      25.72984
## 320       31.3      28.09751
```

```
## 321        37.0       25.41415
## 322        32.2       28.09751
## 323        46.6       29.67595
## 324        27.9       23.36216
## 325        40.8       29.67595
## 326        44.3       32.35931
## 327        43.4       32.35931
## 328        36.4       29.36026
## 329        30.0       29.36026
## 330        44.6       29.36026
## 332        33.8       29.36026
## 333        29.8       30.14949
## 334        32.7       19.10036
## 335        23.7       24.15139
## 336        35.0       26.04552
## 338        32.4       28.57104
## 339        27.2       26.67690
## 340        26.6       26.67690
## 341        25.8       25.41415
## 342        23.5       22.57294
## 343        30.0       26.67690
## 344        39.1       30.78087
## 345        39.0       29.83380
## 346        35.1       30.46518
## 347        32.3       29.36026
## 348        37.0       29.67595
## 349        37.7       30.14949
## 350        34.1       29.20242
## 351        34.7       29.99164
## 352        34.4       29.67595
## 353        29.9       29.67595
## 354        33.0       28.25535
## 356        33.7       28.09751
## 357        32.4       28.09751
## 358        32.9       24.15139
## 359        31.6       28.25535
## 360        28.1       27.30828
## 361        30.7       27.93966
## 362        25.4       21.62587
## 363        24.2       20.99449
## 364        22.4       22.57294
## 365        26.6       23.36216
## 366        20.2       26.04552
## 367        17.6       26.51906
## 368        28.0       26.04552
## 369        27.0       26.04552
## 370        34.0       26.04552
## 371        31.0       26.51906
## 372        29.0       26.67690
## 373        27.0       25.72984
## 374        24.0       25.41415
## 376        36.0       28.25535
## 377        37.0       29.20242
## 378        31.0       29.20242
```

```
## 379        38.0      29.99164
## 380        36.0      28.88673
## 381        36.0      26.04552
## 382        36.0      28.09751
## 383        34.0      28.88673
## 384        38.0      29.36026
## 385        32.0      29.36026
## 386        38.0      29.36026
## 387        25.0      22.57294
## 388        38.0      26.51906
## 389        26.0      25.41415
## 390        22.0      22.25725
## 391        32.0      24.78277
## 392        36.0      26.67690
## 393        27.0      25.72984
## 394        27.0      26.36121
## 395        44.0      31.72793
## 396        32.0      26.67690
## 397        28.0      27.46613
## 398        31.0      26.99259
```

```r
# Calculate the Mean Squared Error (MSE) to evaluate the prediction accuracy
mse <- mean((actual_mpg - predicted_mpg)^2)
cat("Mean Squared Error (MSE):", mse, "\n")
```

```
## Mean Squared Error (MSE): 42.96081
```

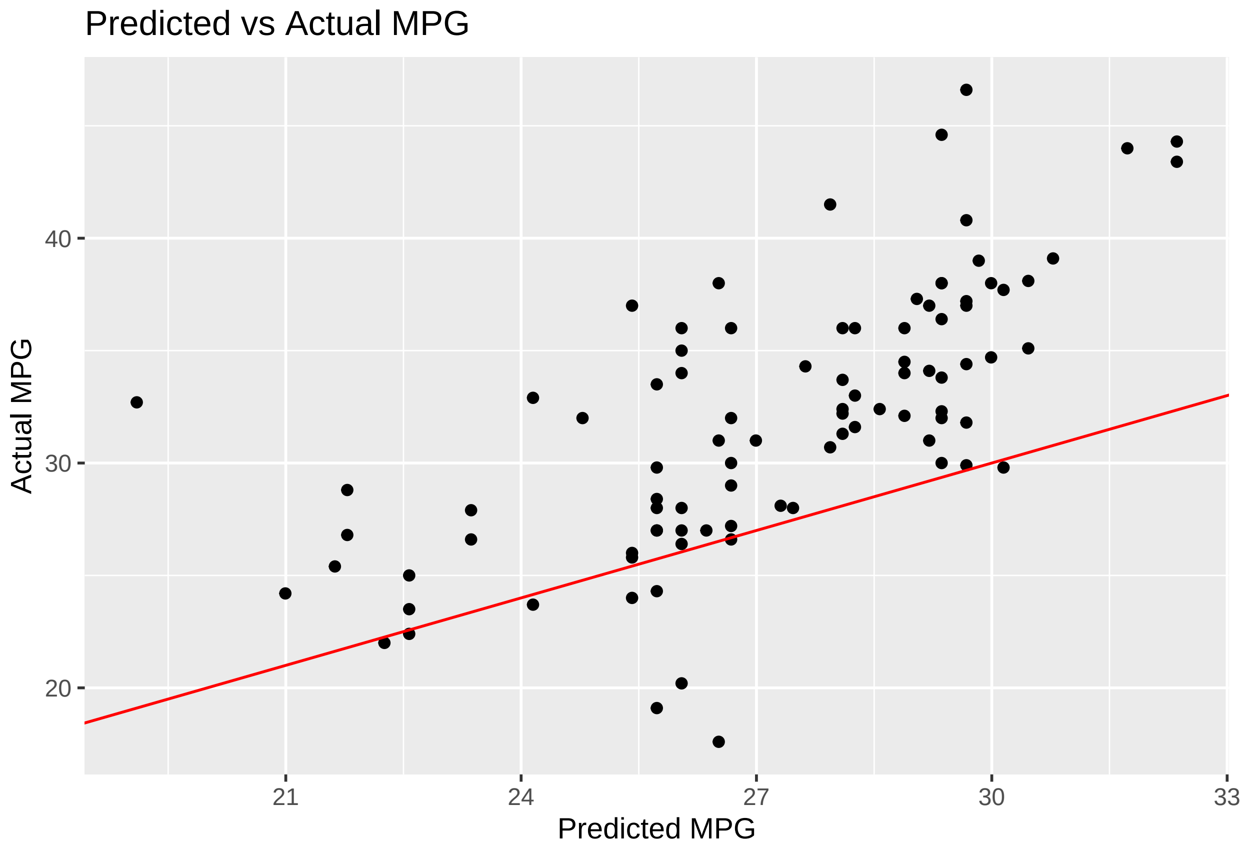
```r
# Calculate the Root Mean Squared Error (RMSE)
rmse <- sqrt(mse)
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 6.55445
```

```r
# Calculate the Mean Absolute Error (MAE)
mae <- mean(abs(actual_mpg - predicted_mpg))
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 5.293952
```

```r
# Visualize differences
ggplot(data.frame(predicted_mpg, actual_mpg)) +
  geom_point(aes(predicted_mpg, actual_mpg)) +
  geom_abline(color="red") +
  labs(title="Predicted vs Actual MPG",
       x="Predicted MPG",
       y="Actual MPG")
```



To evaluate the accuracy of the predictions, the following metrics have been calculated:

Mean Squared Error (MSE): 42.96081 Root Mean Squared Error (RMSE): 6.55445 Mean Absolute Error (MAE): 5.293952 Interpreting the metrics:

Mean Squared Error (MSE): The MSE measures the average squared difference between the predicted and actual values. A lower MSE indicates better predictive performance. In this case, the MSE is 42.96081, which means, on average, the squared difference between the predicted and actual MPG values is 42.96081.

Root Mean Squared Error (RMSE): The RMSE is the square root of the MSE, and it represents the average absolute difference between the predicted and actual values. It is a widely used metric for regression models. The RMSE here is 6.55445, indicating that, on average, the difference between the predicted and actual MPG values is approximately 6.55445.

Mean Absolute Error (MAE): The MAE measures the average absolute difference between the predicted and actual values. Like the RMSE, it is a common metric for regression models. The MAE value of 5.293952 means that, on average, the absolute difference between the predicted and actual MPG values is 5.293952.

In summary, the regression model's accuracy can be assessed using these metrics. Lower values for MSE, RMSE, and MAE indicate better performance, as they imply that the predictions are closer to the actual values. The specific context and requirements of the application will determine whether these accuracy levels are satisfactory or if further improvements are needed.