Final Project Group 53

Bhuvan Kumar Panduranga, Pratik Bhojkar

2023-11-24

```
# Import needed packages
#library(tidyverse)
library(ggplot2)
#library(tigerstats)
#library(reticulate)
library(MASS)
library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.3.2
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Part 1 : Data reading and Visualization

```
auto_mpg<-read.csv("B:/Prog for DA/auto-mpg.csv", sep = ',')
head(auto_mpg)
```

```
##   mpg cylinders 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
```

```
summary(auto_mpg)
```

```
##       mpg          cylinders      displacement     horsepower
weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.
:1613
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st
Qu.:2224
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median
:2804
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean
:2970
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd
Qu.:3608
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.
:5140
```

```
##                                                 NA's   :6
##   acceleration     model.year        origin       car.name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   Length:398
##  1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000   Class :character
##  Median :15.50   Median :76.00   Median :1.000   Mode  :character
##  Mean   :15.57   Mean   :76.01   Mean   :1.573
##  3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
##
```

```
summary(auto_mpg)
```

```
##       mpg          cylinders      displacement     horsepower
weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.
:1613
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st
Qu.:2224
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median
:2804
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean
:2970
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd
Qu.:3608
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.
:5140
##                                                 NA's   :6
##   acceleration     model.year        origin       car.name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   Length:398
##  1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000   Class :character
##  Median :15.50   Median :76.00   Median :1.000   Mode  :character
##  Mean   :15.57   Mean   :76.01   Mean   :1.573
##  3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :24.80   Max.   :82.00   Max.   :3.000
##
```

```
dim(auto_mpg)
```

```
## [1] 398   9
```

The Auto MPG set has 398 rows and 9 columns.

```
str(auto_mpg)
```

```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : 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 sst" ...
```

The data types covered are int, num and character . It has one strings.

```
#factor(data$model_year)['levels']
print("Unique model years")
```

```
## [1] "Unique model years"
```

```
unique(auto_mpg$model.year)
```

```
##  [1] 70 71 72 73 74 75 76 77 78 79 80 81 82
```

This are the years when the models were built.

```
print("Unique origin")
```

```
## [1] "Unique origin"
```

```
unique(auto_mpg$origin)
```

```
## [1] 1 3 2
```

This are the unique origins or say countries that are given numbers, for eg- 1='China', 2='japan', 3='America'. This is a categorical value.

```
print("Unique cylinders")
```

```
## [1] "Unique cylinders"
```

```
unique(auto_mpg$cylinders)
```

```
## [1] 8 4 6 3 5
```

Data Cleaning

Converting the horsepower from char to int and Car name from char to factor

```
auto_mpg$horsepower <- as.integer(as.character(auto_mpg$horsepower))

# Converting 'car.name' to a factor
auto_mpg$car.name <- as.factor(auto_mpg$car.name)

# Displaying the first few rows of the dataset
head(auto_mpg)
```

```
##   mpg cylinders 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
```

```
summary(auto_mpg)
```

```
##       mpg          cylinders      displacement     horsepower
weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.
:1613
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.: 75.0   1st
Qu.:2224
##  Median :23.00   Median :4.000   Median :148.5   Median : 93.5   Median
:2804
##  Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :104.5   Mean
:2970
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd
Qu.:3608
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.
```

```
:5140
##                                                 NA's   :6
##   acceleration     model.year        origin                car.name
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   ford pinto    :  6
##  1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000   amc matador   :  5
##  Median :15.50   Median :76.00   Median :1.000   ford maverick :  5
##  Mean   :15.57   Mean   :76.01   Mean   :1.573   toyota corolla:  5
##  3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000   amc gremlin   :  4
##  Max.   :24.80   Max.   :82.00   Max.   :3.000   amc hornet    :  4
##                                                  (Other)       :369

null_values<-colSums(is.na(auto_mpg))
null_values

##          mpg    cylinders displacement   horsepower       weight
acceleration
##            0            0            0            6            0
0
##   model.year       origin     car.name
##            0            0            0
```

##Removing the null values

```
auto_mpg <- na.omit(auto_mpg)
null_values_after_clean<-colSums(is.na(auto_mpg))
null_values_after_clean

##          mpg    cylinders displacement   horsepower       weight
acceleration
##            0            0            0            0            0
0
##   model.year       origin     car.name
##            0            0            0

dim(auto_mpg)

## [1] 392   9
```

now the data set has 392 points and 9 variables.

```
table(auto_mpg$cylinders)

##
##   3   4   5   6   8
##   4 199   3  83 103

sum(duplicated(auto_mpg))

## [1] 0
```

The cars have one of the numbers of the cylinders. This is a categorial data that we have because the are choose from one of the following categories

```r
mean_mpg_by_cylinders <- tapply(auto_mpg$mpg, auto_mpg$cylinders, mean)
mean_mpg_by_cylinders
```

```
##        3        4        5        6        8
## 20.55000 29.28392 27.36667 19.97349 14.96311
```

This is the mean of the MPG group wise. The average of MPG in unique cylinders.

```r
#minimum value of the mpg/ lowest mpg by the car
min_mpg<-min(auto_mpg$mpg)
min_mpg
```

```
## [1] 9
```

```r
#max mpg of all the cars in the dataset
max_mpg<-max(auto_mpg$mpg)
max_mpg
```

```
## [1] 46.6
```

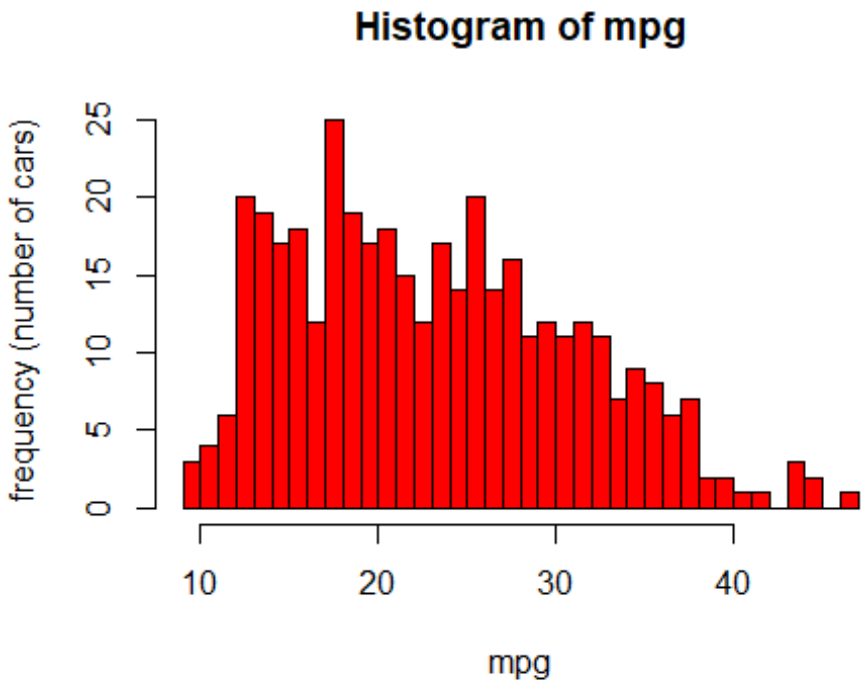
```r
# Calculate min and max MPG for each unique cylinder value
min_max_mpg_by_cylinders <- tapply(auto_mpg$mpg, auto_mpg$cylinders,
function(x) c(min(x), max(x)))

# Display the results
print(min_max_mpg_by_cylinders)
```

```
## $`3`
## [1] 18.0 23.7
##
## $`4`
## [1] 18.0 46.6
##
## $`5`
## [1] 20.3 36.4
##
## $`6`
## [1] 15 38
##
## $`8`
## [1]  9.0 26.6
```

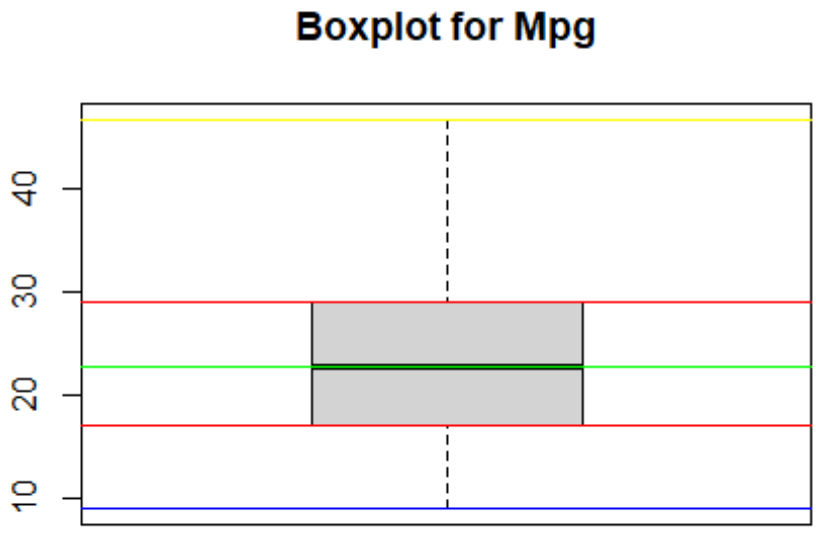
Part 2: Graphical Representation

Histogram of MPG to see the distribution of the data

```
hist(auto_mpg$mpg, breaks = 30, col = "red", main = "Histogram of
mpg",xlab="mpg", ylab = " frequency (number of cars)")
```



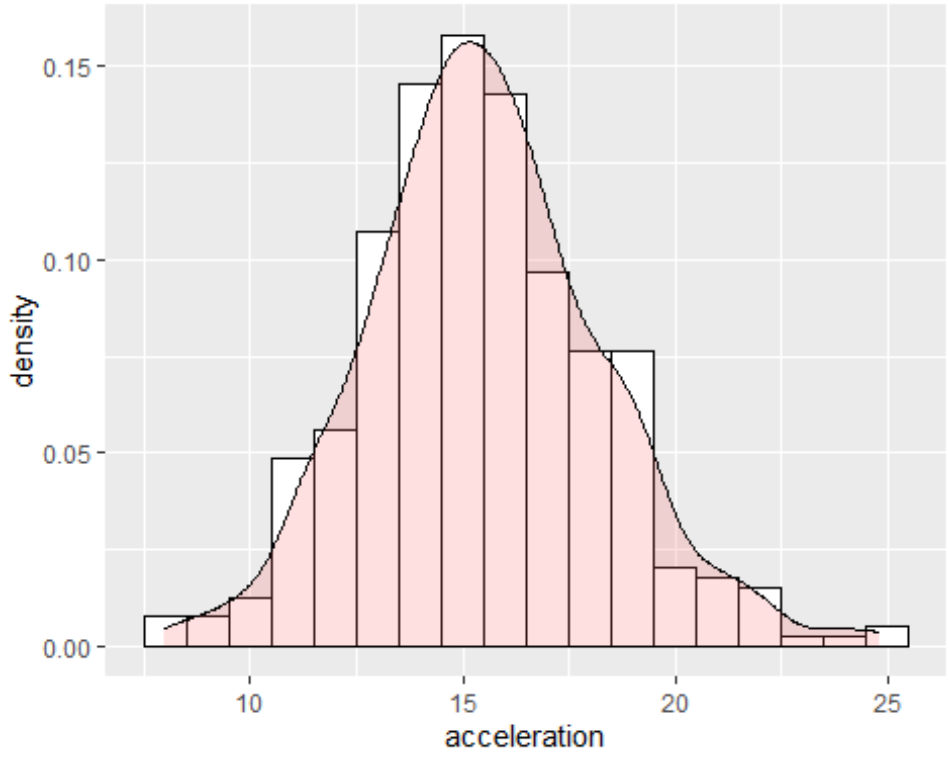
```r
par(mfrow = c(1, 1))
boxplot(auto_mpg$mpg, main= "Boxplot for Mpg")
abline(h = min(auto_mpg$mpg), col = "Blue")
abline(h = max(auto_mpg$mpg), col = "Yellow")
abline(h = median(auto_mpg$mpg), col = "Green")
abline(h = quantile(auto_mpg$mpg, c(0.25, 0.75)), col = "Red")
```



This is a box plot to check if there are any outliers. We can conclude from the observation that there are no a outlier.

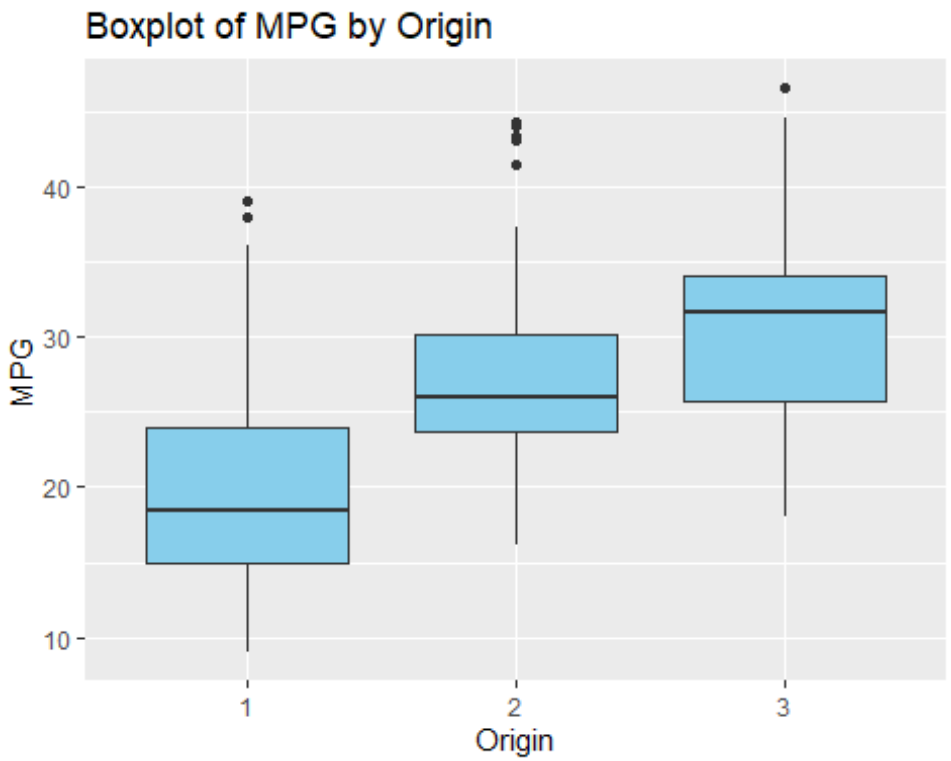
```
ggplot(auto_mpg, aes(x=acceleration)) +
 geom_histogram(aes(y=..density..), colour="black", fill="white", binwidth =
1, bins = 30)+
 geom_density(alpha=.2, fill="#FF6666")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2
3.4.0.
## ℹ Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



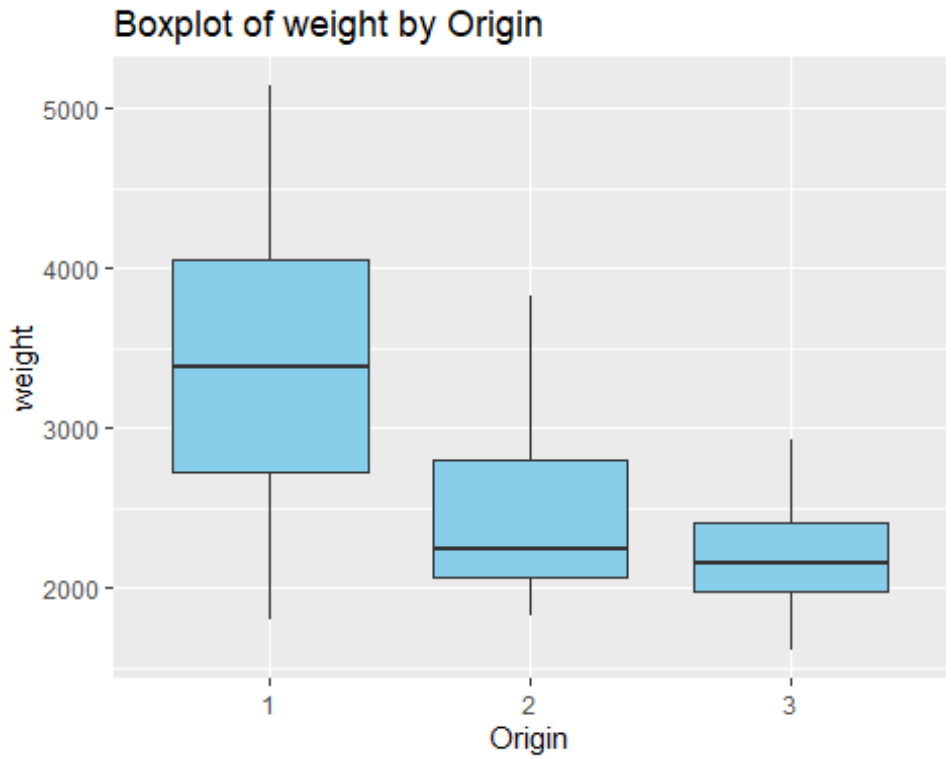
This is a histogram for the acceleration. We can conclude that this is a approximately linearly distributed.

```r
# Load ggplot2 package if not already loaded
library(ggplot2)

# Creating a boxplot to compare 'mpg' by 'origin'
ggplot(auto_mpg, aes(x = as.factor(origin), y = mpg)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Boxplot of MPG by Origin", x = "Origin", y = "MPG")
```



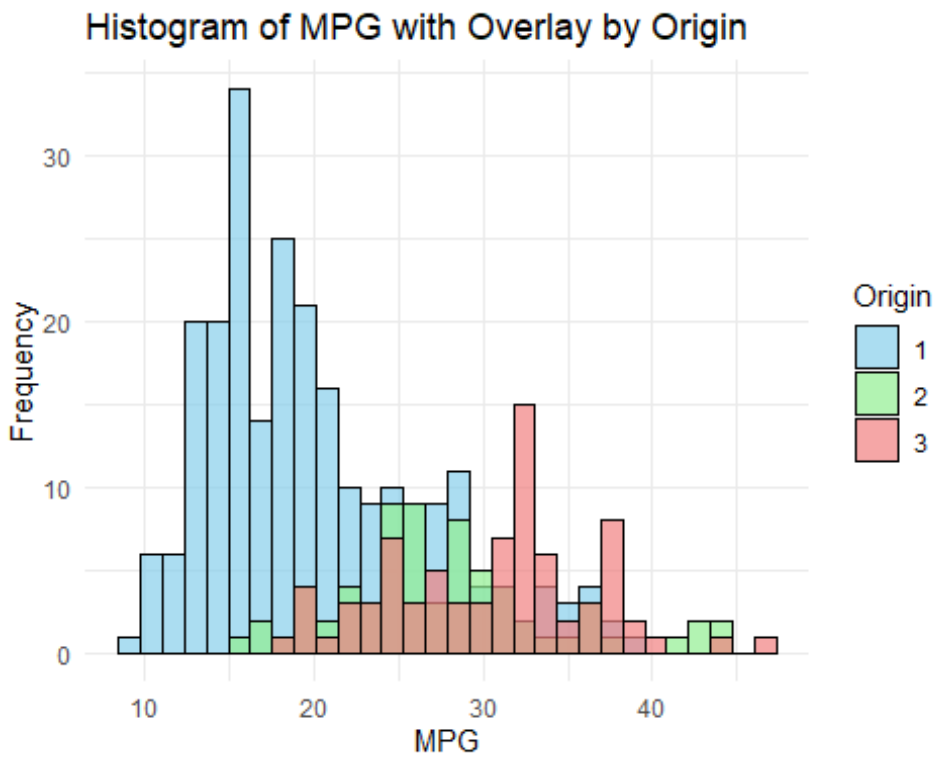
This is a box plot to see the outliers ffor the origin. We conclude that there are outliers. We can also see that MPG for the cars of origin 3 is greater. There may be many factors that may affect it.

```r
ggplot(auto_mpg, aes(x = as.factor(origin), y = weight)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Boxplot of weight by Origin", x = "Origin", y = "weight")
```



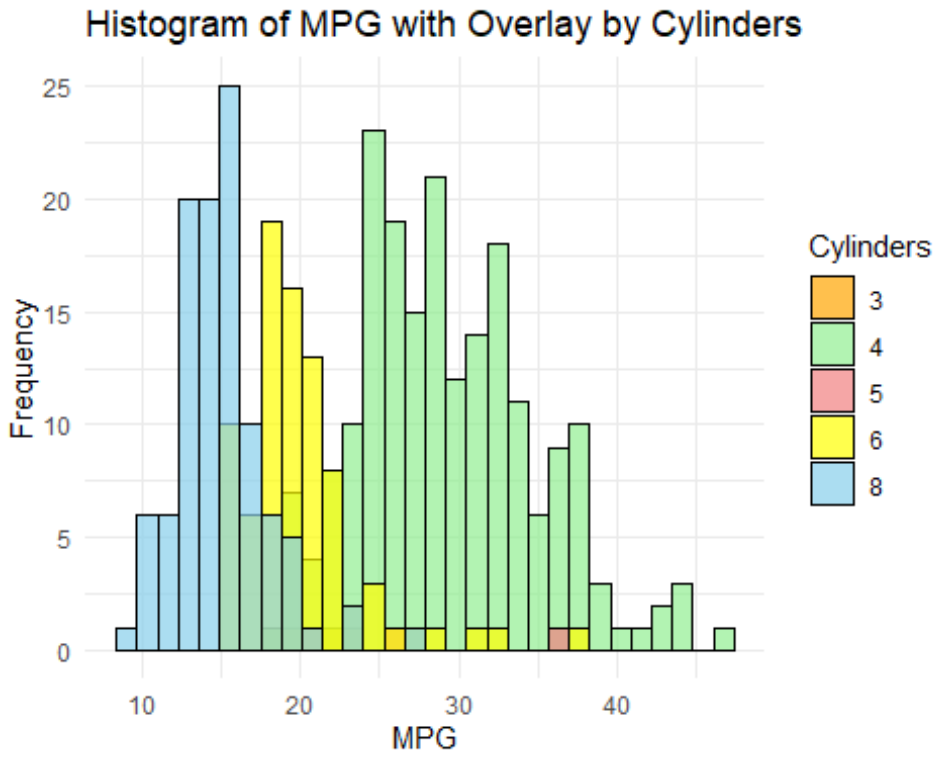
Weight is one of the reasons. The region 3 is having autos with lower weight and with higher mpg as seen above. So we can say that the weight plays imp role in determining mpg.

```r
# Creating a histogram with overlay for 'mpg' by 'origin' with border
ggplot(auto_mpg, aes(x = mpg, fill = as.factor(origin))) +
  geom_histogram(position = "identity", alpha = 0.7, bins = 30, color =
"black") +
  labs(title = "Histogram of MPG with Overlay by Origin", x = "MPG", y =
"Frequency") +
  scale_fill_manual(values = c("skyblue", "lightgreen", "lightcoral"), name =
"Origin") +
  theme_minimal()
```



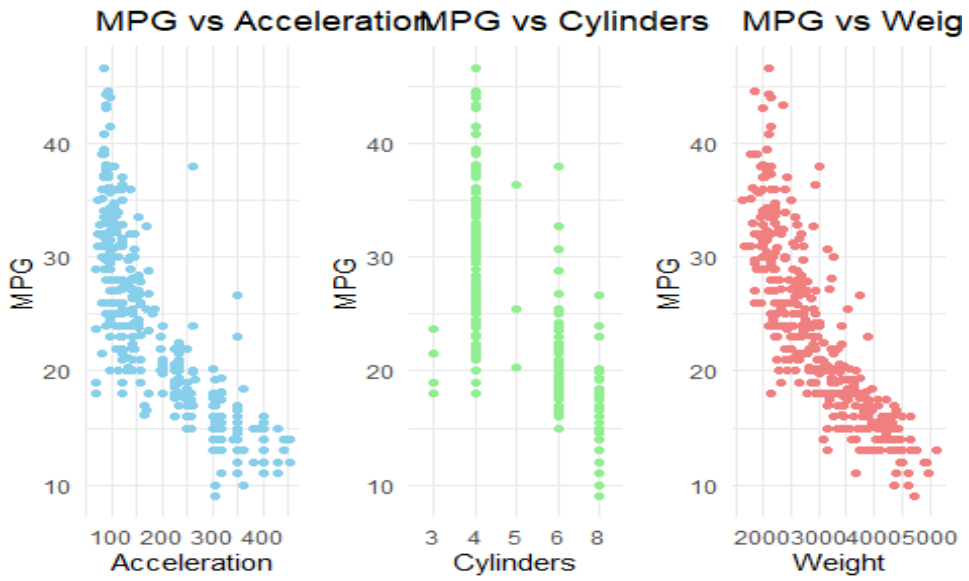
this is a overlay Histogram of MPG and origin. We see that one value is numeric and other is categorial.

```r
# Load ggplot2 package if not already loaded
library(ggplot2)
# Creating a histogram with overlay for 'mpg' by 'cylinders'
ggplot(auto_mpg, aes(x = mpg, fill = as.factor(cylinders))) +
  geom_histogram(position = "identity", alpha = 0.7, bins = 30,
color="black") +
  labs(title = "Histogram of MPG with Overlay by Cylinders", x = "MPG", y =
"Frequency") +
  scale_fill_manual(values = c("orange", "lightgreen", "lightcoral",
"yellow", "skyblue"), name = "Cylinders") +
  theme_minimal()
```



This shows mpg of cars with different cylinders.

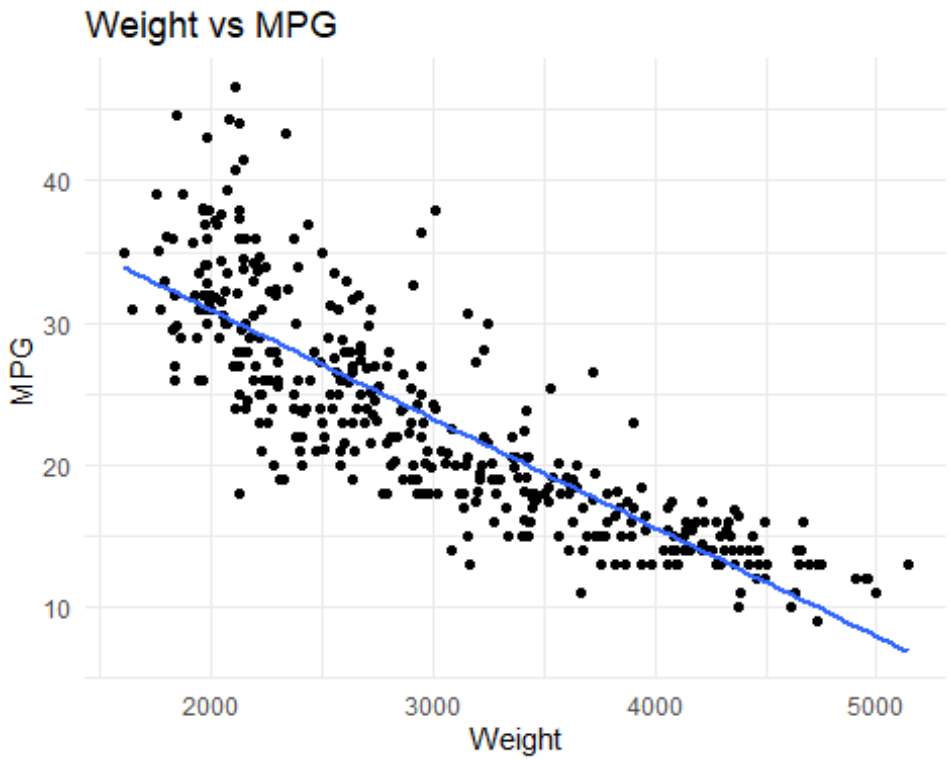
```r
# Load ggplot2 package if not already loaded
library(ggplot2)
# Scattered plot for 'mpg' vs 'acceleration'
scatter_acceleration <- ggplot(auto_mpg, aes(x = displacement, y = mpg)) +
  geom_point(color = "skyblue") +
  labs(title = " MPG vs Acceleration", x = "Acceleration", y = "MPG") +
  theme_minimal()
# Scattered plot for 'mpg' vs 'cylinders'
scatter_cylinders <- ggplot(auto_mpg, aes(x = as.factor(cylinders), y = mpg))
+
  geom_point(color = "lightgreen") +
  labs(title = " MPG vs Cylinders", x = "Cylinders", y = "MPG") +
  theme_minimal()
# Scattered plot for 'mpg' vs 'weight'
scatter_weight <- ggplot(auto_mpg, aes(x = weight, y = mpg)) +
  geom_point(color = "lightcoral") +
  labs(title = " MPG vs Weight", x = "Weight", y = "MPG") +
  theme_minimal()
# Displaying the scattered plots side by side
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

grid.arrange(scatter_acceleration, scatter_cylinders, scatter_weight, ncol =
3)
```



This are the scatterplots that are made to see the relation between mpg and different variables.

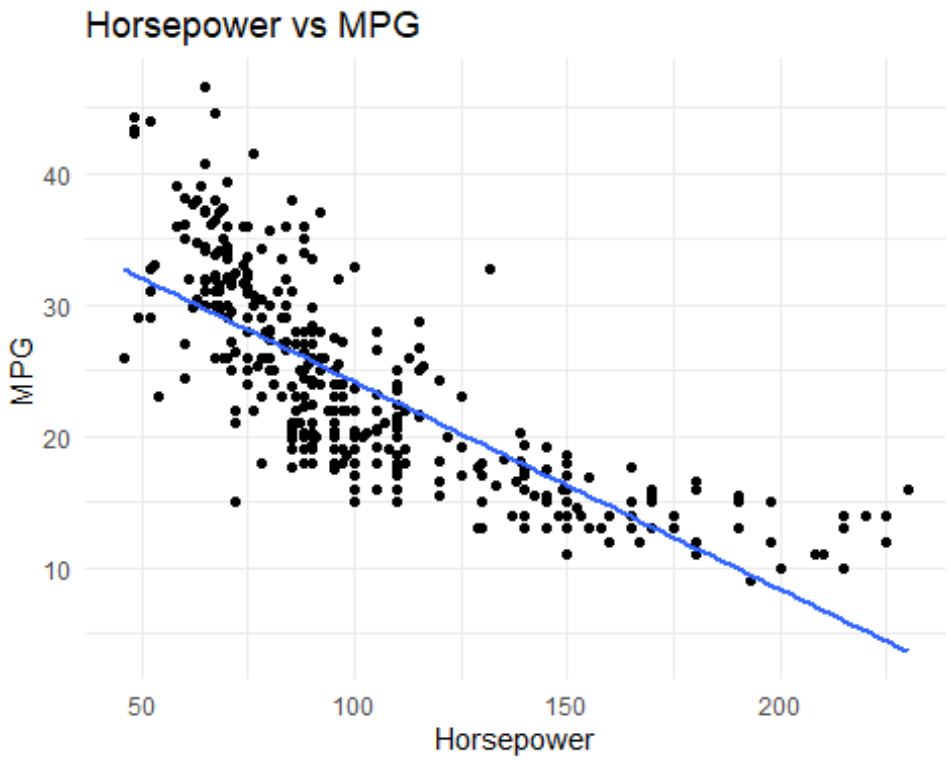
```r
# Scattered plot for weight vs mpg
ggplot(auto_mpg, aes(x = weight, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Weight vs MPG", x = "Weight", y = "MPG") +
  theme_minimal()
```

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



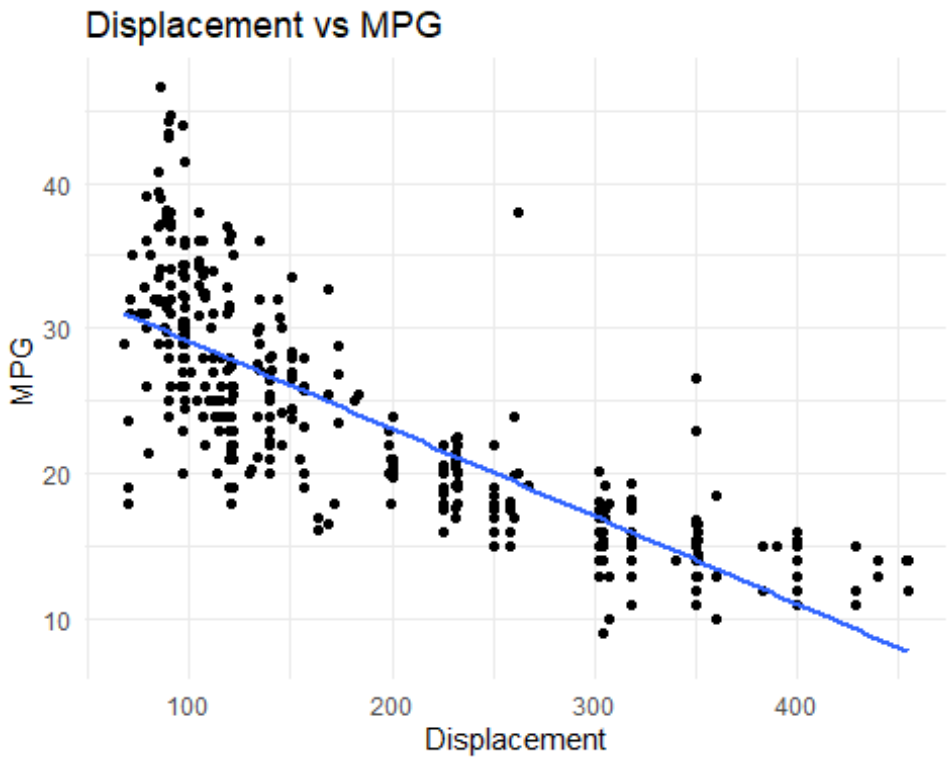
```r
# Scattered plot for horsepower vs mpg
ggplot(auto_mpg, aes(x = horsepower, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Horsepower vs MPG", x = "Horsepower", y = "MPG") +
  theme_minimal()
```

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



```r
# Scattered plot for displacement vs mpg
ggplot(auto_mpg, aes(x = displacement, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Displacement vs MPG", x = "Displacement", y = "MPG") +
  theme_minimal()
```

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



wee can see from the fighure as the hp, weight, displacement incresease the mpg decreases.

Hypothesis Test:

```r
# Converting 'origin' to a factor (if it's not already)
auto_mpg$origin <- as.factor(auto_mpg$origin)
# Performing ANOVA test
anova_result <- aov(mpg ~ origin, data = auto_mpg)
# Summarize the ANOVA results
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## origin        2   7904    3952    96.6 <2e-16 ***
## Residuals   389  15915      41
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

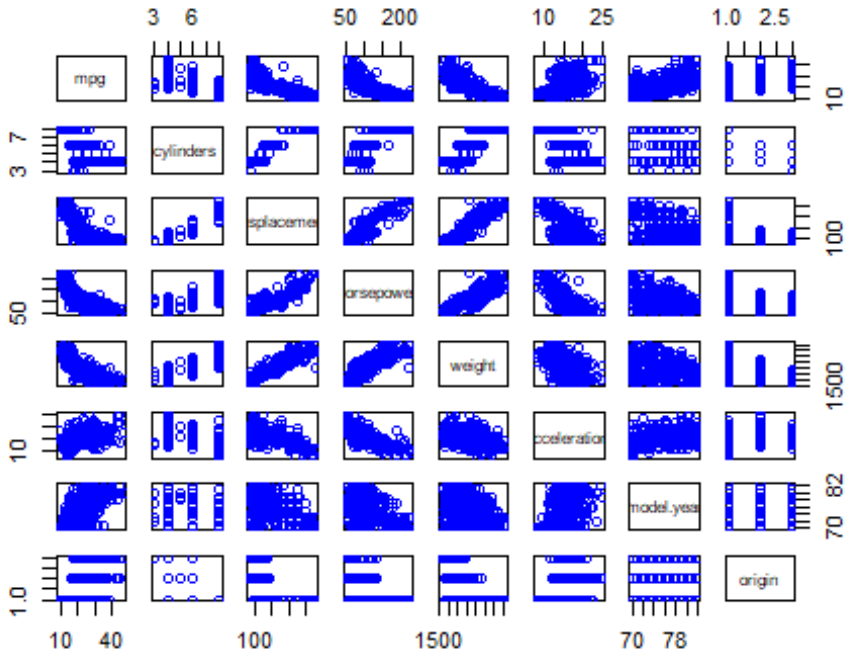
Null hypothesis: There is no significant difference in MPG between cars from different origins.

Alternative hypothesis: There is at least one significant difference in MPG between cars from different origins.

The ANOVA test performed is a valid way to test for a significant relationship between MPG and the origin of the cars. The ANOVA results show that there is a significant difference in MPG between cars from different origins (p-value < 0.001). This means that we can reject the null hypothesis and conclude that there is at least one significant relationship between MPG and origin.

Linear Regression

```
pairs(auto_mpg[,1:8], col = "blue")
```




```
# Excluding 'carname' and any other non-numeric variables
numeric_columns <- auto_mpg[, sapply(auto_mpg, is.numeric)]
# Calculating the correlation matrix for numeric variables
correlation_matrix <- cor(numeric_columns)
# Print the correlation matrix
print(correlation_matrix)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## model.year    0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
##              acceleration model.year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
```

```
## acceleration    1.0000000  0.2903161
## model.year      0.2903161  1.0000000
```

```
print(correlation_matrix)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## model.year    0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
##              acceleration model.year
## mpg             0.4233285  0.5805410
## cylinders      -0.5046834 -0.3456474
## displacement   -0.5438005 -0.3698552
## horsepower     -0.6891955 -0.4163615
## weight         -0.4168392 -0.3091199
## acceleration    1.0000000  0.2903161
## model.year      0.2903161  1.0000000
```

Pairs function show a us the correlation between the variables.

Simple linear regression

For Linear Regression, we are going to determine what all factors are dependent on the mpg variable.

we divide the dataset into training(80) and testing data(20).

```
library(MASS)
train_index = sample(2,nrow(auto_mpg),replace=TRUE, prob = c(0.8,0.2))
auto_mpg_Training <- auto_mpg[train_index==1,]
auto_mpg_Test <- auto_mpg[train_index==2,]
dim(auto_mpg_Training)
```

```
## [1] 307   9
```

```
dim(auto_mpg_Test)
```

```
## [1] 85  9
```

Regression:Modelling the relationship between dependent variable and one or more independent variables..

```
lm_model1 <- lm(mpg ~ weight, data=auto_mpg_Training)
summary(lm_model1)
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = auto_mpg_Training)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6965 -2.8008 -0.3309  2.2101 16.3539
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.5947894  0.9081344   51.31   <2e-16 ***
## weight      -0.0077482  0.0002892  -26.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.33 on 305 degrees of freedom
## Multiple R-squared:  0.7018, Adjusted R-squared:  0.7008
## F-statistic: 717.7 on 1 and 305 DF,  p-value: < 2.2e-16
```

In summary, the linear regression model shows that there is a statistically significant relationship between weight and mpg in the auto.

```
lm_model2 <- lm(mpg ~ horsepower, data=auto_mpg_Training)
summary(lm_model2)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = auto_mpg_Training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5698  -3.2524  -0.4482   2.9233  16.9247
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.939832   0.820423   48.68   <2e-16 ***
## horsepower  -0.157916   0.007257  -21.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.962 on 305 degrees of freedom
## Multiple R-squared:  0.6083, Adjusted R-squared:  0.607
## F-statistic: 473.6 on 1 and 305 DF,  p-value: < 2.2e-16
```

In summary, the linear regression model shows that there is a statistically significant relationship between horsepower and mpg in the auto.

```
predictions <- predict(lm_model1, newdata = auto_mpg_Test)
predictions1 <- predict(lm_model2, newdata = auto_mpg_Test)
summary(predictions)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.769  20.406  25.520  24.753  30.091  32.857
```

```r
summary(predictions1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.409  22.569  25.727  24.377  28.886  31.728
```

```r
MAE(y_pred = predictions, y_true = auto_mpg_Test$mpg)
```

```
## [1] 3.250711
```

```r
MAE(y_pred = predictions1, y_true = auto_mpg_Test$mpg)
```

```
## [1] 3.637089
```

```r
MSE(y_pred = predictions, y_true = auto_mpg_Test$mpg)
```

```
## [1] 18.92616
```

```r
MSE(y_pred = predictions1, y_true = auto_mpg_Test$mpg)
```

```
## [1] 22.06039
```

multiple linear regression

```r
mlm_model <- lm(mpg ~ ., data = auto_mpg_Training[,1:8])
# Print's the summary of the multiple linear regression model
summary(mlm_model)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = auto_mpg_Training[, 1:8])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2955 -2.0724 -0.1432  1.8754 13.4500
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -16.571248   5.512434  -3.006  0.00287 **
## cylinders     -0.572920   0.355412  -1.612  0.10802
## displacement   0.026026   0.008322   3.128  0.00194 **
## horsepower    -0.022953   0.015172  -1.513  0.13138
## weight        -0.006659   0.000734  -9.073  < 2e-16 ***
## acceleration   0.021759   0.113628   0.191  0.84827
## model.year     0.775323   0.060275  12.863  < 2e-16 ***
## origin2        2.797265   0.641672   4.359 1.80e-05 ***
## origin3        2.902732   0.630444   4.604 6.14e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.325 on 298 degrees of freedom
## Multiple R-squared:  0.8281, Adjusted R-squared:  0.8235
## F-statistic: 179.5 on 8 and 298 DF,  p-value: < 2.2e-16
```

We exclude the variable called car.name. The model indicates that the contributing factors to mpg are origin, weight, model.year, displacement.

```
ypred <-predict(object = mlm_model, newdata = auto_mpg_Test[,1:8])
summary(ypred)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.319  20.572  25.267  24.383  29.083  35.703

MAE(y_pred = ypred, y_true = auto_mpg_Test$mpg)

## [1] 2.562706

MSE(y_pred = ypred, y_true = auto_mpg_Test$mpg)

## [1] 10.56341
```

MAE: Mean absolute error, measure of average mistake in a collection of prediction.
MSE:Average squared difference between estimated values and actual value.

forward stepwise egression

```
intercept_only <- lm(mpg ~ 1, data=auto_mpg_Training[,1:8])

all <- lm(mpg~., data=auto_mpg_Training[,1:8])
# performing forward step-wise regression
forward <- stepAIC (intercept_only, direction='forward',scope = formula(all))

## Start:  AIC=1271.26
## mpg ~ 1
##
##                Df Sum of Sq     RSS     AIC
## + weight        1   13454.8  5717.8  901.82
## + displacement  1   12430.2  6742.4  952.42
## + cylinders     1   11821.9  7350.7  978.94
## + horsepower    1   11661.8  7510.8  985.56
## + model.year    1    6923.1 12249.5 1135.72
## + origin        2    6139.7 13032.9 1156.75
## + acceleration  1    3182.8 15989.8 1217.53
## <none>                      19172.6 1271.26
##
## Step:  AIC=901.82
## mpg ~ weight
##
##                Df Sum of Sq    RSS    AIC
## + model.year    1   2093.76 3624.0 763.83
## + horsepower    1    273.75 5444.0 888.76
## + origin        2    196.48 5521.3 895.08
## + acceleration  1    100.97 5616.8 898.35
## + cylinders     1     95.72 5622.0 898.63
## + displacement  1     95.19 5622.6 898.66
## <none>                      5717.8 901.82
```

```
##
## Step:  AIC=763.83
## mpg ~ weight + model.year
##
##                Df Sum of Sq    RSS    AIC
## + origin        2   213.963 3410.0 749.14
## <none>                      3624.0 763.83
## + cylinders     1     4.572 3619.4 765.44
## + horsepower    1     2.970 3621.0 765.57
## + acceleration  1     2.683 3621.3 765.60
## + displacement  1     0.727 3623.3 765.76
##
## Step:  AIC=749.14
## mpg ~ weight + model.year + origin
##
##                Df Sum of Sq    RSS    AIC
## + displacement  1    44.676 3365.4 747.09
## <none>                      3410.0 749.14
## + horsepower    1     4.287 3405.8 750.76
## + acceleration  1     1.012 3409.0 751.05
## + cylinders     1     0.721 3409.3 751.08
##
## Step:  AIC=747.09
## mpg ~ weight + model.year + origin + displacement
##
##                Df Sum of Sq    RSS    AIC
## + horsepower    1    40.134 3325.2 745.41
## + acceleration  1    22.571 3342.8 747.03
## <none>                      3365.4 747.09
## + cylinders     1    21.447 3343.9 747.13
##
## Step:  AIC=745.41
## mpg ~ weight + model.year + origin + displacement + horsepower
##
##                Df Sum of Sq    RSS    AIC
## + cylinders     1   29.4305 3295.8 744.68
## <none>                      3325.2 745.41
## + acceleration  1    1.1008 3324.1 747.31
##
## Step:  AIC=744.68
## mpg ~ weight + model.year + origin + displacement + horsepower +
##     cylinders
##
##                Df Sum of Sq    RSS    AIC
## <none>                      3295.8 744.68
## + acceleration  1   0.40549 3295.4 746.64

summary(forward)
```

```
##
## Call:
## lm(formula = mpg ~ weight + model.year + origin + displacement +
##     horsepower + cylinders, data = auto_mpg_Training[, 1:8])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3119 -2.0848 -0.1394  1.8593 13.4692
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.611e+01  4.938e+00  -3.262  0.00124 **
## weight       -6.591e-03  6.389e-04 -10.315  < 2e-16 ***
## model.year    7.742e-01  5.990e-02  12.925  < 2e-16 ***
## origin2       2.798e+00  6.406e-01   4.368 1.73e-05 ***
## origin3       2.903e+00  6.294e-01   4.612 5.93e-06 ***
## displacement  2.586e-02  8.265e-03   3.129  0.00193 **
## horsepower   -2.476e-02  1.185e-02  -2.089  0.03752 *
## cylinders    -5.781e-01  3.538e-01  -1.634  0.10331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.32 on 299 degrees of freedom
## Multiple R-squared:  0.8281, Adjusted R-squared:  0.8241
## F-statistic: 205.8 on 7 and 299 DF,  p-value: < 2.2e-16
```

```
ypred_forward <-predict(object = forward, newdata = auto_mpg_Test[,1:8])
MAE(y_pred = ypred_forward, y_true = auto_mpg_Test$mpg)
```

```
## [1] 2.56686
```

```
MSE(y_pred = ypred_forward, y_true = auto_mpg_Test$mpg)
```

```
## [1] 10.59884
```

backward stepwise regression

```
intercept_only <- lm(mpg ~ 1, data=auto_mpg_Training[,1:8])

all <- lm(mpg~., data=auto_mpg_Training[,1:8])
backward <- stepAIC (all, direction='backward')
```

```
## Start:  AIC=746.64
## mpg ~ cylinders + displacement + horsepower + weight + acceleration +
##     model.year + origin
##
##                Df Sum of Sq    RSS    AIC
## - acceleration  1      0.41 3295.8 744.68
## <none>                      3295.4 746.64
## - horsepower    1     25.31 3320.7 746.99
## - cylinders     1     28.74 3324.1 747.31
## - displacement  1    108.17 3403.6 754.56
```

```
## - origin        2    305.77 3601.2 769.89
## - weight        1    910.33 4205.7 819.53
## - model.year    1   1829.70 5125.1 880.22
##
## Step:  AIC=744.68
## mpg ~ cylinders + displacement + horsepower + weight + model.year +
##     origin
##
##                Df Sum of Sq    RSS    AIC
## <none>                      3295.8 744.68
## - cylinders     1     29.43 3325.2 745.41
## - horsepower    1     48.12 3343.9 747.13
## - displacement  1    107.94 3403.7 752.58
## - origin        2    305.86 3601.7 767.93
## - weight        1   1172.79 4468.6 836.14
## - model.year    1   1841.48 5137.3 878.95
```

```
summary(backward)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     model.year + origin, data = auto_mpg_Training[, 1:8])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3119 -2.0848 -0.1394  1.8593 13.4692
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.611e+01  4.938e+00  -3.262  0.00124 **
## cylinders    -5.781e-01  3.538e-01  -1.634  0.10331
## displacement  2.586e-02  8.265e-03   3.129  0.00193 **
## horsepower   -2.476e-02  1.185e-02  -2.089  0.03752 *
## weight       -6.591e-03  6.389e-04 -10.315  < 2e-16 ***
## model.year    7.742e-01  5.990e-02  12.925  < 2e-16 ***
## origin2       2.798e+00  6.406e-01   4.368 1.73e-05 ***
## origin3       2.903e+00  6.294e-01   4.612 5.93e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.32 on 299 degrees of freedom
## Multiple R-squared:  0.8281, Adjusted R-squared:  0.8241
## F-statistic: 205.8 on 7 and 299 DF,  p-value: < 2.2e-16
```

```
ypred_backward <-predict(object = backward, newdata = auto_mpg_Test[,1:8])
MAE(y_pred = ypred_backward, y_true = auto_mpg_Test$mpg)
```

```
## [1] 2.56686
```

```
MSE(y_pred = ypred_backward, y_true = auto_mpg_Test$mpg)## [1] 10.59884
```