

# Prediction of Fuel Efficiency using the Auto-MPG Dataset

**Group # 14**

Syeda Shanzay Shah 23I-2016

Sarita Sangrez 23I-2088

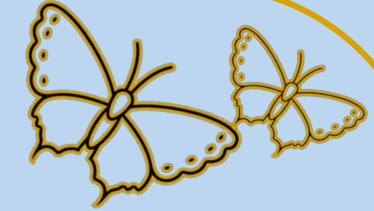
Laiba Nasir 23I-2079

CY-4A

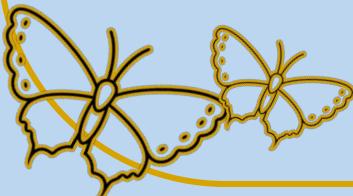


# Task 1: Introduction

---



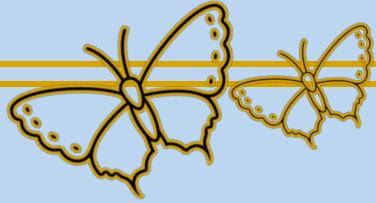
- **Topic Chosen:** Prediction of Fuel Efficiency using the Auto-MPG Dataset.
- **Objective:** To find out how different features of a car (like weight, horsepower, etc.) affect its fuel efficiency (MPG).
- **Importance of Study:** Helps in understanding which car features save fuel.
- Useful for car manufacturers and customers to make better decisions  
Promotes environmentally friendly vehicle choices.
- **Approach:** Use data analysis and basic prediction techniques on a real-world dataset.



# Task 2: Dataset Selection

---

- **Dataset Name:** Auto-MPG (Miles Per Gallon)
- **Source:** UCI Machine Learning Repository — a reliable and public source of datasets
- **Type:** Secondary data (already collected and made available online)
- **Total Records:** 398 car entries
- **Main Features** in the Dataset:
  - *mpg*: Miles per gallon (target variable)
  - *cylinders*: Number of cylinders in the engine
  - *horsepower*: Engine power
  - *weight*: Vehicle weight
  - *acceleration*: Time to reach a certain speed
  - *model year*: Year of car model
  - *origin*: Country of origin (1 = USA, 2 = Europe, 3 = Japan)



## Task 3: Descriptive statistics on dataset



# Displaying dataset:

The screenshot shows the RGui interface with two windows open: 'R Console' and 'R Editor'.

**R Console** window content:

```
> # Read the CSV file
> data <- read.csv(file_path)
>
> ?data
> data
  mpg Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
1 18.0     8        307.0       130    3504       12.0          70
2 15.0     8        350.0       165    3693       11.5          70
3 18.0     8        318.0       150    3436       11.0          70
4 16.0     8        304.0       150    3433       12.0          70
5 17.0     8        302.0       140    3449       10.5          70
6 15.0     8        429.0       198    4341       10.0          70
7 14.0     8        454.0       220    4354        9.0          70
8 14.0     8        440.0       215    4312        8.5          70
9 15.0     8        350.0       165    3693       11.5          70
10 16.0     8        304.0       150    3433       12.0          70
11 NA       4        133.0       115    3090       17.5          70
12 NA       8        350.0       165    4142       11.5          70
13 NA       8        351.0       153    4034       11.0          70
14 NA       8        383.0       175    4166       10.5          70
15 NA       8        360.0       175    3850       11.0          70
16 15.0     8        383.0       170    3563       10.0          70
17 14.0     8        340.0       160    3609        8.0          70
18 NA       8        302.0       140    3353        8.0          70
19 15.0     8        400.0       150    3761       9.5          70
```

**R Editor** window content:

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(lst 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the da
```

# ANALYSIS OF DATASET BEFORE CLEANING

Load dataset in R environment:

The screenshot shows the RGui interface with two main windows: the R Console and the R Editor.

**R Console (Left Pane):**

```
> file_path <- "C:/Users/User/Downloads/autompq.csv"
>
> # Read the CSV file
> data <- read.csv(file_path)
>
> ?data
> data
  mpg Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
1 18.0     8        307.0      130    3504       12.0        70
2 15.0     8        350.0      165    3693       11.5        70
3 18.0     8        318.0      150    3436       11.0        70
4 16.0     8        304.0      150    3433       12.0        70
5 17.0     8        302.0      140    3449       10.5        70
6 15.0     8        429.0      198    4341       10.0        70
7 14.0     8        454.0      220    4354        9.0        70
8 14.0     8        440.0      215    4312        8.5        70
9 15.0     8        350.0      165    3693       11.5        70
10 16.0     8        304.0      150    3433       12.0        70
11 NA       4        133.0      115    3090       17.5        70
12 NA       8        350.0      165    4142       11.5        70
13 NA       8        351.0      153    4034       11.0        70
14 NA       8        383.0      175    4166       10.5        70
15 NA       8        360.0      175    3850       11.0        70
16 15.0     8        383.0      170    3563       10.0        70
17 14.0     8        340.0      160    3609        8.0        70
```

**R Editor (Right Pane):**

```
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(1st 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the da
```

# Descriptive stats on MPG:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its output. The right window is the R Editor, showing the same code in a script file.

```
RGui
File Edit Packages Windows Help
R Console
> # descriptive stats for mpg
> mean(data$mpg.Y., na.rm = TRUE)
[1] 23.50207
> median(data$mpg.Y, na.rm = TRUE)
[1] 22.75
> sd(data$mpg.Y, na.rm = TRUE)      # Standard deviation
[1] 7.821489
> var(data$mpg.Y, na.rm = TRUE)      # Variance
[1] 61.17568
> min(data$mpg.Y, na.rm = TRUE)
[1] 9
> max(data$mpg.Y, na.rm = TRUE)
[1] 46.6
> range(data$mpg.Y, na.rm = TRUE)
[1] 9.0 46.6
> quantile(data$mpg.Y, na.rm = TRUE)
  0%   25%   50%   75%  100%
9.00 17.50 22.75 29.00 46.60
>
> # descriptive stats for cylinders
> mean(data$Cylinders, na.rm = TRUE)
[1] 5.463731
> median(data$Cylinders, na.rm = TRUE)
[1] 4
> |
```

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# descriptive stats for mpg
mean(data$mpg.Y., na.rm = TRUE)
median(data$mpg.Y, na.rm = TRUE)
sd(data$mpg.Y, na.rm = TRUE)      # Standard deviation
var(data$mpg.Y, na.rm = TRUE)      # Variance
min(data$mpg.Y, na.rm = TRUE)
max(data$mpg.Y, na.rm = TRUE)
range(data$mpg.Y, na.rm = TRUE)
quantile(data$mpg.Y, na.rm = TRUE)

# descriptive stats for cylinders
mean(data$Cylinders, na.rm = TRUE)
median(data$Cylinders, na.rm = TRUE)
sd(data$Cylinders, na.rm = TRUE)
var(data$Cylinders, na.rm = TRUE)
```

At the bottom of the screen, the taskbar shows various open applications including a search bar, a pinned icons bar, and system status indicators like the date and time (11:15 AM, 5/8/2025), battery level (27°C), and a notification icon (19).

# Descriptive stats on Cylinders:

The screenshot shows the RGui interface with two panes. The left pane is the R Console, and the right pane is the R Editor. The R Console contains R code and its output for calculating descriptive statistics (mean, median, sd, var, min, max, range, quantile) for 'displacement', 'horsepower', and 'weight' variables from a dataset. The R Editor pane shows the same R code for these variables.

```
RGui
File Edit View Misc Packages Windows Help
R Console
R C:\Users\User\Downloads\Lecture 22.R - R Editor
9.00 17.50 22.75 29.00 46.60
>
> # descriptive stats for cylinders
> mean(data$Cylinders, na.rm = TRUE)
[1] 5.463731
> median(data$Cylinders, na.rm = TRUE)
[1] 4
> # descriptive stats for horsepower
> mean(data$HorsePower, na.rm = TRUE)
[1] 103.7617
> median(data$HorsePower, na.rm = TRUE)
[1] 92.5
> sd(data$HorsePower, na.rm = TRUE)
[1] 37.60264
> var(data$HorsePower, na.rm = TRUE)
[1] 1413.959
> min(data$HorsePower, na.rm = TRUE)
[1] 46
> max(data$HorsePower, na.rm = TRUE)
[1] 230
> range(data$HorsePower, na.rm = TRUE)
[1] 46 230
> quantile(data$HorsePower, na.rm = TRUE)
  0%   25%   50%   75%  100%
46.0  75.0  92.5 125.0 230.0
<
+ descriptive stats for displacement
mean(data$Displacement, na.rm = TRUE)
median(data$Displacement, na.rm = TRUE)
sd(data$Displacement, na.rm = TRUE)
var(data$Displacement, na.rm = TRUE)
min(data$Displacement, na.rm = TRUE)
max(data$Displacement, na.rm = TRUE)
range(data$Displacement, na.rm = TRUE)
quantile(data$Displacement, na.rm = TRUE)

+ descriptive stats for horsepower
mean(data$HorsePower, na.rm = TRUE)
median(data$HorsePower, na.rm = TRUE)
sd(data$HorsePower, na.rm = TRUE)
var(data$HorsePower, na.rm = TRUE)
min(data$HorsePower, na.rm = TRUE)
max(data$HorsePower, na.rm = TRUE)
range(data$HorsePower, na.rm = TRUE)
quantile(data$HorsePower, na.rm = TRUE)

+ descriptive stats for weight
mean(data$Weight, na.rm = TRUE)
median(data$Weight, na.rm = TRUE)
sd(data$Weight, na.rm = TRUE)
var(data$Weight, na.rm = TRUE)
```

# Descriptive stats on Displacement:

The screenshot shows the RGui interface with two windows open. The main window is the R Console, displaying R code and its output. The R Editor window is also visible, showing additional R code. The taskbar at the bottom includes icons for File Explorer, Edge, and other applications.

R Gui

File Edit Packages Windows Help

R Console

```
> max(data$HorsePower, na.rm = TRUE)
[1] 230
> range(data$HorsePower, na.rm = TRUE)
[1] 46 230
> quantile(data$HorsePower, na.rm = TRUE)
  0%   25%   50%   75%  100%
46.0  75.0  92.5 125.0 230.0
>
> # descriptive stats for weight
> # descriptive stats for displacement
> mean(data$Displacement, na.rm = TRUE)
[1] 193.2759
> median(data$Displacement, na.rm = TRUE)
[1] 151
> sd(data$Displacement, na.rm = TRUE)
[1] 103.0871
> var(data$Displacement, na.rm = TRUE)
[1] 10626.96
> min(data$Displacement, na.rm = TRUE)
[1] 68
> max(data$Displacement, na.rm = TRUE)
[1] 455
> range(data$Displacement, na.rm = TRUE)
[1] 68 455
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
max(data$Cylinders, na.rm = TRUE)
range(data$Cylinders, na.rm = TRUE)
quantile(data$Cylinders, na.rm = TRUE)

# descriptive stats for displacement
mean(data$Displacement, na.rm = TRUE)
median(data$Displacement, na.rm = TRUE)
sd(data$Displacement, na.rm = TRUE)
var(data$Displacement, na.rm = TRUE)
min(data$Displacement, na.rm = TRUE)
max(data$Displacement, na.rm = TRUE)
range(data$Displacement, na.rm = TRUE)
quantile(data$Displacement, na.rm = TRUE)

# descriptive stats for horsepower
mean(data$HorsePower, na.rm = TRUE)
median(data$HorsePower, na.rm = TRUE)
sd(data$HorsePower, na.rm = TRUE)
var(data$HorsePower, na.rm = TRUE)
min(data$HorsePower, na.rm = TRUE)
max(data$HorsePower, na.rm = TRUE)
range(data$HorsePower, na.rm = TRUE)
quantile(data$HorsePower, na.rm = TRUE)

# descriptive stats for weight
```

# Descriptive stats on Horsepower:

The screenshot shows the RGui interface with two windows open. The main window is the R Console, displaying R code and its output. The second window is the R Editor, showing a script file with R code. Both windows are running on a Windows operating system.

**R Console Output:**

```
> min(data$Displacement, na.rm = TRUE)
[1] 68
> max(data$Displacement, na.rm = TRUE)
[1] 455
> range(data$Displacement, na.rm = TRUE)
[1] 68 455
> # descriptive stats for horsepower
> mean(data$HorsePower, na.rm = TRUE)
[1] 103.7617
> median(data$HorsePower, na.rm = TRUE)
[1] 92.5
> sd(data$HorsePower, na.rm = TRUE)
[1] 37.60264
> var(data$HorsePower, na.rm = TRUE)
[1] 1413.959
> min(data$HorsePower, na.rm = TRUE)
[1] 46
> max(data$HorsePower, na.rm = TRUE)
[1] 230
> range(data$HorsePower, na.rm = TRUE)
[1] 46 230
> quantile(data$HorsePower, na.rm = TRUE)
  0%   25%   50%   75%  100%
46.0  75.0  92.5 125.0 230.0
> |
```

**R Editor Script Content:**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
max(data$Cylinders, na.rm = TRUE)
range(data$Cylinders, na.rm = TRUE)
quantile(data$Cylinders, na.rm = TRUE)

# descriptive stats for displacement
mean(data$Displacement, na.rm = TRUE)
median(data$Displacement, na.rm = TRUE)
sd(data$Displacement, na.rm = TRUE)
var(data$Displacement, na.rm = TRUE)
min(data$Displacement, na.rm = TRUE)
max(data$Displacement, na.rm = TRUE)
range(data$Displacement, na.rm = TRUE)
quantile(data$Displacement, na.rm = TRUE)

# descriptive stats for horsepower
mean(data$HorsePower, na.rm = TRUE)
median(data$HorsePower, na.rm = TRUE)
sd(data$HorsePower, na.rm = TRUE)
var(data$HorsePower, na.rm = TRUE)
min(data$HorsePower, na.rm = TRUE)
max(data$HorsePower, na.rm = TRUE)
range(data$HorsePower, na.rm = TRUE)
quantile(data$HorsePower, na.rm = TRUE)

# descriptive stats for weight
```



Type here to search



11:18 AM  
5/8/2025



# Descriptive stats on Weight:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its output for descriptive statistics on the 'Weight' column of a dataset. The right window is the R Editor, showing the same R code. Both windows have a blue header bar with the R logo and the title of the file.

```
R Gui
File Edit Packages Windows Help
R Console
R Editor
```

```
> range(data$HorsePower, na.rm = TRUE)
[1] 46 230
> quantile(data$HorsePower, na.rm = TRUE)
  0%   25%   50%   75%  100%
46.0 75.0 92.5 125.0 230.0
> # descriptive stats for weight
> mean(data$Weight, na.rm = TRUE)
[1] 2972.674
> median(data$Weight, na.rm = TRUE)
[1] 2803.5
> sd(data$Weight, na.rm = TRUE)
[1] 847.1126
> var(data$Weight, na.rm = TRUE)
[1] 717599.8
> min(data$Weight, na.rm = TRUE)
[1] 1613
> max(data$Weight, na.rm = TRUE)
[1] 5140
> range(data$Weight, na.rm = TRUE)
[1] 1613 5140
> quantile(data$Weight, na.rm = TRUE)
  0%   25%   50%   75%  100%
1613.00 2223.75 2803.50 3608.00 5140.00
>
>
```

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
max(data$Displacement, na.rm = TRUE)
range(data$Displacement, na.rm = TRUE)
quantile(data$Displacement, na.rm = TRUE)

# descriptive stats for horsepower
mean(data$HorsePower, na.rm = TRUE)
median(data$HorsePower, na.rm = TRUE)
sd(data$HorsePower, na.rm = TRUE)
var(data$HorsePower, na.rm = TRUE)
min(data$HorsePower, na.rm = TRUE)
max(data$HorsePower, na.rm = TRUE)
range(data$HorsePower, na.rm = TRUE)
quantile(data$HorsePower, na.rm = TRUE)

# descriptive stats for weight
mean(data$Weight, na.rm = TRUE)
median(data$Weight, na.rm = TRUE)
sd(data$Weight, na.rm = TRUE)
var(data$Weight, na.rm = TRUE)
min(data$Weight, na.rm = TRUE)
max(data$Weight, na.rm = TRUE)
range(data$Weight, na.rm = TRUE)
quantile(data$Weight, na.rm = TRUE)

# descriptive stats for acceleration
```

# Descriptive stats on Acceleration:

The screenshot shows the RGui interface with two windows open. The main window is the R Console, displaying R code and its output. The secondary window is the R Editor, showing the same code. Both windows have a blue header bar with the R logo and the title of the respective window.

R Gui

File Edit Packages Windows Help

R Console

```
[1] 1613 5140
> quantile(data$Weight, na.rm = TRUE)
  0%   25%   50%   75%  100%
1613.00 2223.75 2803.50 3608.00 5140.00
>
> # descriptive stats for acceleration
> mean(data$acceleration, na.rm = TRUE)
[1] 15.59119
> median(data$acceleration, na.rm = TRUE)
[1] 15.5
> sd(data$acceleration, na.rm = TRUE)
[1] 2.715989
> var(data$acceleration, na.rm = TRUE)
[1] 7.376598
> min(data$acceleration, na.rm = TRUE)
[1] 8
> max(data$acceleration, na.rm = TRUE)
[1] 24.8
> range(data$acceleration, na.rm = TRUE)
[1] 8.0 24.8
> quantile(data$acceleration, na.rm = TRUE)
  0%   25%   50%   75%  100%
8.000 13.825 15.500 17.075 24.800
>
>
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
max(data$HorsePower, na.rm = TRUE)
range(data$HorsePower, na.rm = TRUE)
quantile(data$HorsePower, na.rm = TRUE)

# descriptive stats for weight
mean(data$Weight, na.rm = TRUE)
median(data$Weight, na.rm = TRUE)
sd(data$Weight, na.rm = TRUE)
var(data$Weight, na.rm = TRUE)
min(data$Weight, na.rm = TRUE)
max(data$Weight, na.rm = TRUE)
range(data$Weight, na.rm = TRUE)
quantile(data$Weight, na.rm = TRUE)

# descriptive stats for acceleration
mean(data$acceleration, na.rm = TRUE)
median(data$acceleration, na.rm = TRUE)
sd(data$acceleration, na.rm = TRUE)
var(data$acceleration, na.rm = TRUE)
min(data$acceleration, na.rm = TRUE)
max(data$acceleration, na.rm = TRUE)
range(data$acceleration, na.rm = TRUE)
quantile(data$acceleration, na.rm = TRUE)

# descriptive stats for modelyear
```



Type here to search



11:18 AM  
5/8/2025



# Descriptive stats on Model year:

RGui

File Edit Packages Windows Help

R Console

```
> range(data$acceleration, na.rm = TRUE)
[1] 8.0 24.8
> quantile(data$acceleration, na.rm = TRUE)
  0%   25%   50%   75%  100%
8.000 13.825 15.500 17.075 24.800
>
> # descriptive stats for modelyear
> mean(data$ModelYear, na.rm = TRUE)
[1] 76.07254
> median(data$ModelYear, na.rm = TRUE)
[1] 76
> sd(data$ModelYear, na.rm = TRUE)
[1] 3.635313
> var(data$ModelYear, na.rm = TRUE)
[1] 13.2155
> min(data$ModelYear, na.rm = TRUE)
[1] 70
> max(data$ModelYear, na.rm = TRUE)
[1] 82
> range(data$ModelYear, na.rm = TRUE)
[1] 70 82
> quantile(data$ModelYear, na.rm = TRUE)
  0%   25%   50%   75%  100%
 70    73    76    79    82
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
min(data$Weight, na.rm = TRUE)
max(data$Weight, na.rm = TRUE)
range(data$Weight, na.rm = TRUE)
quantile(data$Weight, na.rm = TRUE)

# descriptive stats for acceleration
mean(data$acceleration, na.rm = TRUE)
median(data$acceleration, na.rm = TRUE)
sd(data$acceleration, na.rm = TRUE)
var(data$acceleration, na.rm = TRUE)
min(data$acceleration, na.rm = TRUE)
max(data$acceleration, na.rm = TRUE)
range(data$acceleration, na.rm = TRUE)
quantile(data$acceleration, na.rm = TRUE)

# descriptive stats for modelyear
mean(data$ModelYear, na.rm = TRUE)
median(data$ModelYear, na.rm = TRUE)
sd(data$ModelYear, na.rm = TRUE)
var(data$ModelYear, na.rm = TRUE)
min(data$ModelYear, na.rm = TRUE)
max(data$ModelYear, na.rm = TRUE)
range(data$ModelYear, na.rm = TRUE)
quantile(data$ModelYear, na.rm = TRUE)
```



Type here to search



27°C 11:19 AM  
5/8/2025



# Descriptive stats on origin:

R Gui

File Edit Packages Windows Help

R Console

```
> range(data$ModelYear, na.rm = TRUE)
[1] 70 82
> quantile(data$ModelYear, na.rm = TRUE)
  0%  25%  50%  75% 100%
 70   73   76   79   82
>
> # descriptive stats for origin
> mean(data$origin, na.rm = TRUE)
[1] 1.572539
> median(data$origin, na.rm = TRUE)
[1] 1
> sd(data$origin, na.rm = TRUE)
[1] 0.8033516
> var(data$origin, na.rm = TRUE)
[1] 0.6453738
> min(data$origin, na.rm = TRUE)
[1] 1
> max(data$origin, na.rm = TRUE)
[1] 3
> range(data$origin, na.rm = TRUE)
[1] 1 3
> quantile(data$origin, na.rm = TRUE)
  0%  25%  50%  75% 100%
  1    1    1    2    3
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
min(data$acceleration, na.rm = TRUE)
max(data$acceleration, na.rm = TRUE)
range(data$acceleration, na.rm = TRUE)
quantile(data$acceleration, na.rm = TRUE)

# descriptive stats for modelyear
mean(data$ModelYear, na.rm = TRUE)
median(data$ModelYear, na.rm = TRUE)
sd(data$ModelYear, na.rm = TRUE)
var(data$ModelYear, na.rm = TRUE)
min(data$ModelYear, na.rm = TRUE)
max(data$ModelYear, na.rm = TRUE)
range(data$ModelYear, na.rm = TRUE)
quantile(data$ModelYear, na.rm = TRUE)

# descriptive stats for origin
mean(data$origin, na.rm = TRUE)
median(data$origin, na.rm = TRUE)
sd(data$origin, na.rm = TRUE)
var(data$origin, na.rm = TRUE)
min(data$origin, na.rm = TRUE)
max(data$origin, na.rm = TRUE)
range(data$origin, na.rm = TRUE)
quantile(data$origin, na.rm = TRUE)
```

# Summary of Descriptive Stats:

RGui

File Edit View Misc Packages Windows Help

K Console

```
> # =====
> # Task 3: Descriptive Statistics
> # =====
>
> library(psych)
>
> #Shows Summary of all Data - detailed descriptive stats
> describe(data)
   vars n    mean      sd median trimmed    mad min     max range
mpg.Y.    1 386  23.50    7.82  22.75  23.06  8.60  9  46.6  37.6
Cylinders  2 386   5.46    1.70   4.00   5.35  0.00  3  8.0  5.0
Displacement 3 386 193.28 103.09 151.00 183.20 90.44 68 455.0 387.0
HorsePower  4 386 103.76   37.60  92.50  99.44 28.91 46 230.0 184.0
Weight      5 386 2972.67 847.11 2803.50 2911.62 945.16 1613 5140.0 3527.0
acceleration 6 386   15.59    2.72   15.50  15.51  2.45  8  24.8 16.8
ModelYear   7 386   76.07    3.64   76.00  76.07  4.45 70  82.0 12.0
origin      8 386   1.57    0.80   1.00   1.47  0.00  1  3.0  2.0
CarName*    9 386 146.05  88.33 147.00 145.82 118.61  1 299.0 298.0
      skew kurtosis    se
mpg.Y.    0.45   -0.55  0.40
Cylinders 0.51   -1.39  0.09
Displacement 0.68   -0.82  5.25
HorsePower  1.05    0.61  1.91
Weight      0.52   -0.80 43.12
acceleration 0.36    0.37  0.14
ModelYear   0.01   -1.17  0.19
origin      0.92   -0.84  0.04
CarName*    0.03   -1.26  4.50
>
> # Quick Summary descriptive stats
> summary(data)
   mpg.Y.      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.:105.0 1st Qu.:75.0  1st Qu.:2224
Median :22.75 Median :4.000  Median :151.0  Median :92.5  Median :2804
Mean   :23.50 Mean   :5.464  Mean   :193.3  Mean   :103.8  Mean   :2973
3rd Qu.:29.00 3rd Qu.:8.000  3rd Qu.:262.0 3rd Qu.:125.0 3rd Qu.:3608
Max.   :46.60 Max.   :8.000  Max.   :455.0  Max.   :230.0  Max.   :5140
acceleration ModelYear      origin      CarName
Min.   : 8.00  Min.   :70.00  Min.   :1.000  Length:386
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
```

C:\Users\HP\Downloads\Lecture 22.R - R Editor

```
# =====
# Task 3: Descriptive Statistics
# =====

library(psych)

#Shows Summary of all Data - detailed descriptive stats
describe(data)

# Quick Summary descriptive stats
summary(data)

# descriptive stats for mpg
mean(data$mpg.Y., na.rm = TRUE)
median(data$mpg.Y., na.rm = TRUE)
sd(data$mpg.Y., na.rm = TRUE)                                # Standard deviation
var(data$mpg.Y., na.rm = TRUE)                                 # Variance
min(data$mpg.Y., na.rm = TRUE)
max(data$mpg.Y., na.rm = TRUE)
range(data$mpg.Y., na.rm = TRUE)
quantile(data$mpg.Y., na.rm = TRUE)

# descriptive stats for cylinders
mean(data$Cylinders, na.rm = TRUE)
median(data$Cylinders, na.rm = TRUE)
sd(data$Cylinders, na.rm = TRUE)
var(data$Cylinders, na.rm = TRUE)
min(data$Cylinders, na.rm = TRUE)
max(data$Cylinders, na.rm = TRUE)
range(data$Cylinders, na.rm = TRUE)
quantile(data$Cylinders, na.rm = TRUE)

# descriptive stats for displacement
mean(data$Displacement, na.rm = TRUE)
median(data$Displacement, na.rm = TRUE)
sd(data$Displacement, na.rm = TRUE)
```

# Display the default number of rows(1st 6 rows) from the dataset:

R Gui

File Edit Packages Windows Help

R Console

```
398     1           ford granada 1
399     3           toyota celica gt
400     1           dodge charger 2.2
401     1           chevrolet camaro
402     1           ford mustang gl
403     2           vw pickup
404     1           dodge rampage
405     1           ford ranger
406     1           chevy s-10
> head(data)
mpg. Y. Cylinders Displacement HorsePower Weight acceleration ModelYear 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
CarName
1 chevrolet chevelle malibu
2 buick skylark 320
3 plymouth satellite
4 amc rebel sst
5 ford torino
6 ford galaxie 500
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(1st 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the da
<
```

Type here to search

7:12 PM 5/7/2025

# Display the default number of rows(last 6 rows) from the dataset

RGui

File Edit Packages Windows Help

R Console

```
5      17      8      302      140    3449      10.5      70      1
6      15      8      429      198    4341      10.0      70      1
               CarName
1 chevrolet chevelle malibu
2       buick skylark 320
3     plymouth satellite
4        amc rebel sst
5         ford torino
6     ford galaxie 500
> tail(data)
   mpg Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
401 27      4          151       90    2950      17.3       82
402 27      4          140       86    2790      15.6       82
403 44      4          97        52    2130      24.6       82
404 32      4          135       84    2295      11.6       82
405 28      4          120       79    2625      18.6       82
406 31      4          119       82    2720      19.4       82
  origin      CarName
401 1 chevrolet camaro
402 1 ford mustang gl
403 2       vw pickup
404 1 dodge rampage
405 1      ford ranger
406 1      chevy s-10
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(first 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data, 3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data, 4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the da
```

Type here to search

7:13 PM 5/7/2025

# Dimensions:

RGui

File Edit Packages Windows Help

R Console

```
CarName
1 chevrolet chevelle malibu
2 buick skylark 320
3 plymouth satellite
4 amc rebel sst
5 ford torino
6 ford galaxie 500
> tail(data)
   mpg. Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
401 27 4 151 90 2950 17.3 82
402 27 4 140 86 2790 15.6 82
403 44 4 97 52 2130 24.6 82
404 32 4 135 84 2295 11.6 82
405 28 4 120 79 2625 18.6 82
406 31 4 119 82 2720 19.4 82
origin CarName
401 1 chevrolet camaro
402 1 ford mustang gl
403 2 vw pickup
404 1 dodge rampage
405 1 ford ranger
406 1 chevy s-10
> dim(data)
[1] 406  9
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
head(data)
# Display the first 3 rows of the dataset
head(data, 3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data, 4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)
|
# Use the 'names()' function to get the names of the columns in the da
names(data)

# Check the structure of the iris dataset
str(data)

# Use the 'summary()' function to generate a statistical summary of th
# This provides a quick overview of the data, including min, max, mean
summary(data)

# Missing Values in iris dataset
# Total number of missing values in iris dataset
<
```

# Names of variables in dataset:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its output. The right window is the R Editor, displaying R code. Both windows have standard operating system window controls (minimize, maximize, close).

**R Console**

```
3      plymouth satellite
4          amc rebel sst
5          ford torino
6          ford galaxie 500
> tail(data)
   mpg.Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
401    27         4           151        90    2950       17.3        82
402    27         4           140        86    2790       15.6        82
403    44         4           97        52    2130       24.6        82
404    32         4          135        84    2295       11.6        82
405    28         4          120        79    2625       18.6        82
406    31         4          119        82    2720       19.4        82
  origin      CarName
401     1 chevrolet camaro
402     1 ford mustang gl
403     2 vw pickup
404     1 dodge rampage
405     1 ford ranger
406     1 chevy s-10
> dim(data)
[1] 406  9
> names(data)
[1] "mpg.Y."      "Cylinders"    "Displacement" "HorsePower"   "Weight"
[6] "acceleration" "ModelYear"    "origin"        "CarName"
> |
```

**R Editor**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
```

```
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the data
names(data)

# Check the structure of the iris dataset
str(data)

# Use the 'summary()' function to generate a statistical summary of the data
# This provides a quick overview of the data, including min, max, mean
summary(data)

# Missing Values in iris dataset
# Total number of missing values in iris dataset
<
```

# Structure of dataset:

The screenshot shows the RGui interface with two windows open: the R Console and the R Editor.

**R Console (Left Window):**

```
406      31          4        119          82    2720       19.4       82
  origin      CarName
401      1 chevrolet camaro
402      1 ford mustang gl
403      2     vw pickup
404      1    dodge rampage
405      1     ford ranger
406      1     chevy s-10
> dim(data)
[1] 406  9
> names(data)
[1] "mpg.Y."      "Cylinders"   "Displacement" "HorsePower"   "Weight"
[6] "acceleration" "ModelYear"   "origin"       "CarName"
> str(data)
'data.frame': 406 obs. of  9 variables:
 $ mpg.Y.    : num  18 15 18 16 17 15 14 14 15 16 ...
 $ Cylinders : int  8 8 8 8 8 8 8 8 8 ...
 $ Displacement: num  307 350 318 304 302 429 454 440 350 304 ...
 $ HorsePower : int  130 165 150 150 140 198 220 215 165 150 ...
 $ Weight    : int  3504 3693 3436 3433 3449 4341 4354 4312 3693 3433 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 11.5 12 ...
 $ ModelYear : int  70 70 70 70 70 70 70 70 70 70 ...
 $ origin    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ CarName   : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth$>
```

**R Editor (Right Window):**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
head(data)
# Display the first 3 rows of the dataset
head(data,3)

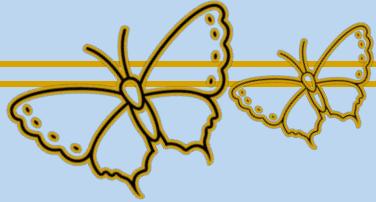
# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

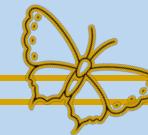
# Use the 'names()' function to get the names of the columns in the da
names(data)

# Check the structure of the iris dataset
str(data)
| 
# Use the 'summary()' function to generate a statistical summary of th
# This provides a quick overview of the data, including min, max, mean
summary(data)

# Missing Values in iris dataset
# Total number of missing values in iris dataset
<
```



## Task 4: Data Prepossessing



# Displaying dataset:

The screenshot shows the RGui interface with two windows open: the R Console and the R Editor.

**R Console (Left Window):**

```
> # Read the CSV file
> data <- read.csv(file_path)
>
> ?data
> data
  mpg Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
1 18.0     8        307.0       130    3504       12.0          70
2 15.0     8        350.0       165    3693       11.5          70
3 18.0     8        318.0       150    3436       11.0          70
4 16.0     8        304.0       150    3433       12.0          70
5 17.0     8        302.0       140    3449       10.5          70
6 15.0     8        429.0       198    4341       10.0          70
7 14.0     8        454.0       220    4354        9.0          70
8 14.0     8        440.0       215    4312        8.5          70
9 15.0     8        350.0       165    3693       11.5          70
10 16.0     8        304.0       150    3433       12.0          70
11 NA       4        133.0       115    3090       17.5          70
12 NA       8        350.0       165    4142       11.5          70
13 NA       8        351.0       153    4034       11.0          70
14 NA       8        383.0       175    4166       10.5          70
15 NA       8        360.0       175    3850       11.0          70
16 15.0     8        383.0       170    3563       10.0          70
17 14.0     8        340.0       160    3609        8.0          70
18 NA       8        302.0       140    3353        8.0          70
19 15.0     8        400.0       150    3761       9.5          70
```

**R Editor (Right Window):**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(first 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the da
```

# Missing values in dataset:

RGui

File Edit Packages Windows Help

R Console

```
1st Qu.:17.00  1st Qu.:4.000  1st Qu.:105   1st Qu.: 75.75  1st Qu.:2228  
Median :22.45  Median :4.000   Median :151    Median : 95.00  Median :2845  
Mean   :23.44  Mean   :5.505   Mean   :196    Mean   :105.36  Mean   :2986  
3rd Qu.:29.00  3rd Qu.:8.000   3rd Qu.:302   3rd Qu.:130.00  3rd Qu.:3628  
Max.   :46.60  Max.   :8.000   Max.   :455    Max.   :230.00  Max.   :5140  
NA's    :8  
acceleration  ModelYear      origin      CarName  
Min.   : 8.00  Min.   :70.00   Min.   :1.000   Length:406  
1st Qu.:13.53 1st Qu.:73.00  1st Qu.:1.000   Class  :character  
Median :15.50  Median :76.00  Median :1.000   Mode   :character  
Mean   :15.50  Mean   :75.92  Mean   :1.557  
3rd Qu.:17.07  3rd Qu.:79.00  3rd Qu.:2.000  
Max.   :24.80  Max.   :82.00  Max.   :3.000  
  
> # Missing Values in dataset  
> # Total number of missing values in dataset  
> sum(is.na(data))  
[1] 14  
> # Missing values per column in dataset  
> colSums(is.na(data))  
  mpg.Y. Cylinders Displacement HorsePower Weight acceleration  
       8          0            0           6          0          0  
ModelYear      origin      CarName  
      0          0            0  
>
```

R C:\Users\User\Downloads\Lecture 22.R - R Editor

```
str(data)  
  
# Use the 'summary()' function to generate a statistical summary of th  
# This provides a quick overview of the data, including min, max, mean  
summary(data)  
  
# Missing Values in dataset  
# Total number of missing values in dataset  
sum(is.na(data))  
# Missing values per column in dataset  
colSums(is.na(data))  
  
# Duplicate rows in  
# Number of duplicate rows in dataset  
sum(duplicated(data))  
# Shows duplicate rows in  
duplicated_data <- data[duplicated(data), ]  
duplicated_data  
  
# Assuming your dataset is called data  
data$CarName_numeric <- as.numeric(factor(data$CarName))  
  
# Check the result  
head(data$CarName_numeric)
```

# Duplicates in dataset:

RGui

File Edit Packages Windows Help



R Console

```
> colSums(is.na(data))
   mpg.Y. Cylinders Displacement HorsePower      Weight acceleration
      8          0           0         6            0            0
ModelYear      origin     CarName
      0          0           0

> # Duplicate rows in
> # Number of duplicate rows in dataset
> sum(duplicated(data))
[1] 5
> # Shows duplicate rows in
> duplicated_data <- data[duplicated(data), ]
> duplicated_data
   mpg.Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
9     15        8          350       165    3693      11.5          70
10    16        8          304       150    3433      12.0          70
21    15        8          350       165    3693      11.5          70
25    15        8          350       165    3693      11.5          70
27    16        8          304       150    3433      12.0          70

  origin     CarName
9     1 buick skylark 320
10    1 amc rebel sst
21    1 buick skylark 320
25    1 buick skylark 320
27    1 amc rebel sst
>
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
str(data)

# Use the 'summary()' function to generate a statistical summary of th
# This provides a quick overview of the data, including min, max, mean
summary(data)

# Missing Values in dataset
# Total number of missing values in dataset
sum(is.na(data))
# Missing values per column in dataset
colSums(is.na(data))

# Duplicate rows in
# Number of duplicate rows in dataset
sum(duplicated(data))
# Shows duplicate rows in
duplicated_data <- data[duplicated(data), ]
duplicated_data
|
# # Assuming your dataset is called data
data$CarName_numeric <- as.numeric(factor(data$CarName))

# Check the result
head(data$CarName_numeric)
```

# Convert categorial variables to numeric:

RGui - [R Console]

File Edit View Misc Packages Windows Help

[306] "vw dasher (diesel)"  
[307] "vw pickup"  
[308] "vw rabbit"  
[309] "vw rabbit c (diesel)"  
[310] "vw rabbit custom"  
> # Assuming your dataset is called data  
> data\$CarName\_numeric <- as.numeric(factor(data\$CarName))  
>  
> # Check how car names are mapped to numbers  
> levels(factor(data\$CarName))  
[1] "amc ambassador brougham"  
[2] "amc ambassador sst"  
[3] "amc concord"  
[4] "amc concord d/l"  
[5] "amc concord dl"  
[6] "amc concord dl 6"  
[7] "amc gremlin"  
[8] "amc hornet"  
[9] "amc hornet sportabout (sw)"  
[10] "amc matador"  
[11] "amc matador (sw)"  
[12] "amc pacer"  
[13] "amc pacer d/l"  
[14] "amc rebel sst"  
[15] "amc rebel sst (sw)"  
[16] "amc spirit dl"  
[17] "audi 100 ls"  
[18] "audi 100ls"  
[19] "audi 4000"  
[20] "audi 5000"  
[21] "audi 5000s (diesel)"  
[22] "audi fox"  
[23] "bmw 2002"  
[24] "bmw 320i"  
[25] "buick century"

RGui - [R Console]

File Edit View Misc Packages Windows Help

81	1	chevrolet chevelle concours (sw)	49
82	1	ford gran torino (sw)	145
83	1	plymouth satellite custom (sw)	238
84	2	volvo 145e (sw)	301
85	2	volkswagen 411 (sw)	288
86	2	peugeot 504 (sw)	216
87	2	renault 12 (sw)	258
88	1	ford pinto (sw)	160
89	3	datsun 510 (sw)	87
90	3	toyouta corona mark ii (sw)	284
91	1	dodge colt (sw)	106
92	3	toyota corolla 1600 (sw)	274
93	1	buick century 350	26
94	1	amc matador	10
95	1	chevrolet malibu	56
96	1	ford gran torino	144
97	1	dodge coronet custom	111
98	1	mercury marquis brougham	197
99	1	chevrolet caprice classic	45
100	1	ford ltd	150
101	1	plymouth fury gran sedan	226
102	1	chrysler new yorker brougham	75
103	1	buick electra 225 custom	30
104	1	amc ambassador brougham	1
105	1	plymouth valiant	240
106	1	chevrolet nova custom	63
107	1	amc hornet	8
108	1	ford maverick	152
109	1	plymouth duster	224
110	2	volkswagen super beetle	297
111	1	chevrolet impala	55
112	1	ford country	130
113	1	plymouth custom suburb	223
114	1	oldsmobile vista cruiser	211
115	1	amc gremlin	7

# Standardization:

R Gui

File Edit View Misc Packages Windows Help

R Console

```
> # View the first 10 rows
> head(data_scaled, 10)
   mpg.Y. Cylinders Displacement HorsePower      Weight acceleration
1 -0.6943935 1.449815 1.062007 0.6396594 0.6143182 -1.251712
2 -1.0770342 1.449815 1.473392 1.5481743 0.8382717 -1.430403
3 -0.6943935 1.449815 1.167245 1.1588108 0.5337423 -1.609093
4 -0.9494873 1.449815 1.033305 1.1588108 0.5301875 -1.251712
5 -0.8219404 1.449815 1.014171 0.8992351 0.5491465 -1.787783
6 -1.0770342 1.449815 2.229193 2.4047741 1.6061125 -1.966473
7 -1.2045811 1.449815 2.468370 2.9758407 1.6215167 -2.323854
8 -1.2045811 1.449815 2.334431 2.8460528 1.5717493 -2.502544
9 -1.0770342 1.449815 1.473392 1.5481743 0.8382717 -1.430403
10 -0.9494873 1.449815 1.033305 1.1588108 0.5301875 -1.251712
  ModelYear      origin
1 -1.579514 -0.7026267
2 -1.579514 -0.7026267
3 -1.579514 -0.7026267
4 -1.579514 -0.7026267
5 -1.579514 -0.7026267
6 -1.579514 -0.7026267
7 -1.579514 -0.7026267
8 -1.579514 -0.7026267
9 -1.579514 -0.7026267
10 -1.579514 -0.7026267
>
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
# Check how car names are mapped to numbers
levels(factor(data$CarName))

# View the result
print(data)

# Standardize all 4 numeric features
data_scaled <- as.data.frame(scale(data[, 1:8]))
# View the result (standardized values for all 150 rows)
head(data_scaled, 406) # View first 10 rows

# Standardize by column names
data_scaled <- as.data.frame(scale(data[, c("mpg.Y.", "Cylinders", "D.
                           "HorsePower", "Weight",
                           "acceleration", "ModelYe.

# View the first 10 rows
head(data_scaled, 10)

# Function to detect outliers using IQR method
find_outliers <- function(data) {
  Q1 <- quantile(data, 0.25)
  Q3 <- quantile(data, 0.75)
  <
```

Type here to search

8:07 PM 5/7/2025

# Outliers:

RGui

File Edit Packages Windows Help

R Console

```
> length(outliers_data_Weight)
[1] 0
> length(outliers_data_acceleration)
[1] 6
> length(outliers_data_ModelYear)
[1] 0
> length(outliers_data_origin)
[1] 0
>
>
> data[outliers_data_mpg.Y., "mpg.Y."]
Error: object 'outliers_data_mpg.Y.' not found
> data[outliers_data_Cylinders, "Cylinders"]
integer(0)
> data[outliers_data_Displacement, "Displacement"]
numeric(0)
> data[outliers_data_HorsePower, "HorsePower"]
Error: object 'outliers_data_HorsePower' not found
> data[outliers_data_Weight, "Weight"]
integer(0)
> data[outliers_data_acceleration, "acceleration"]
[1] 8.5 8.5 15.5 16.0 15.5 16.0
> data[outliers_data_ModelYear, "ModelYear"]
integer(0)
> |
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
outliers_data_acceleration <- find_outliers(data$acceleration)
outliers_data_ModelYear <- find_outliers(data$ModelYear)
outliers_data_origin <- find_outliers(data$origin)

length(outliers_data_mpg.Y.)
length(outliers_data_Cylinders)
length(outliers_data_Displacement)
length(outliers_data_HorsePower)
length(outliers_data_Weight)
length(outliers_data_acceleration)
length(outliers_data_ModelYear)
length(outliers_data_origin)

data[outliers_data_mpg.Y., "mpg.Y."]
data[outliers_data_Cylinders, "Cylinders"]
data[outliers_data_Displacement, "Displacement"]
data[outliers_data_HorsePower, "HorsePower"]
data[outliers_data_Weight, "Weight"]
data[outliers_data_acceleration, "acceleration"]
data[outliers_data_ModelYear, "ModelYear"]
data[outliers_data_origin, "origin"]
```



Type here to search



8:08 PM  
5/7/2025

17

31°C

# Correlation coefficient:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its execution results. The right window is the R Editor, showing a script file with R code.

**R Console:**

```
>  
> data[outliers_data_mpg.Y., "mpg.Y."  
Error: object 'outliers_data_mpg.Y.' not found  
> data[outliers_data_Cylinders, "Cylinders"]  
integer(0)  
> data[outliers_data_Displacement, "Displacement"]  
numeric(0)  
> data[outliers_data_HorsePower, "HorsePower"]  
Error: object 'outliers_data_HorsePower' not found  
> data[outliers_data_Weight, "Weight"]  
integer(0)  
> data[outliers_data_acceleration, "acceleration"]  
[1] 8.5 8.5 15.5 16.0 15.5 16.0  
> data[outliers_data_ModelYear, "ModelYear"]  
integer(0)  
> # Correlation Coefficient  
> # Calculate the Pearson correlation coefficient  
> # Calculate Pearson correlation matrix for selected columns  
> correlation_matrix <- cor(data[c("mpg.Y.", "Cylinders", "Displacement", "Hors$  
> print(correlation_coefficient)  
[1] 0.02509152  
>  
>  
>  
> |
```

**R Editor:**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
```

```
data[outliers_data_mpg.Y., "mpg.Y."  
data[outliers_data_Cylinders, "Cylinders"]  
data[outliers_data_Displacement, "Displacement"]  
data[outliers_data_HorsePower, "HorsePower"]  
data[outliers_data_Weight, "Weight"]  
data[outliers_data_acceleration, "acceleration"]  
data[outliers_data_ModelYear, "ModelYear"]  
data[outliers_data_origin, "origin"]  
  
# Correlation Coefficient  
# Calculate the Pearson correlation coefficient  
# Calculate Pearson correlation matrix for selected columns  
correlation_matrix <- cor(data[c("mpg.Y.", "Cylinders", "Displacement", "Hors$  
print(correlation_coefficient)  
  
## Correlation Matrix  
# Calculate the correlation matrix for the first four numeric attribu  
# Select only the relevant numeric columns
```

# Correlation Matrix:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its output. The right window is the R Editor, showing a script with comments explaining the correlation matrix calculation.

R Gui

File Edit Packages Windows Help

R Console

```
>
> # Calculate the correlation matrix
> correlation_matrix <- cor(correlation_data)
>
> # Print the result
> print(correlation_matrix)
```

	mpg.Y.	Cylinders	Displacement	HorsePower	Weight	acceleration
mpg.Y.	1	NA	NA	NA	NA	NA
Cylinders	NA	1.0000000	0.9528921	NA	0.8924826	-0.5288009
Displacement	NA	0.9528921	1.0000000	NA	0.9316949	-0.5594605
HorsePower	NA	NA	NA	1	NA	NA
Weight	NA	0.8924826	0.9316949	NA	1.0000000	-0.4306180
acceleration	NA	-0.5288009	-0.5594605	NA	-0.4306180	1.0000000
ModelYear	NA	-0.3861112	-0.4015608	NA	-0.3280744	0.3123125
origin	NA	-0.5670455	-0.6144024	NA	-0.5833515	0.2262668
	ModelYear	origin				
mpg.Y.	NA	NA				
Cylinders	-0.3861112	-0.5670455				
Displacement	-0.4015608	-0.6144024				
HorsePower	NA	NA				
Weight	-0.3280744	-0.5833515				
acceleration	0.3123125	0.2262668				
ModelYear	1.0000000	0.2135103				
origin	0.2135103	1.0000000				

R Editor

```
# Calculate Pearson correlation matrix for selected columns
correlation_matrix <- cor(data[c("mpg.Y.", "Cylinders", "Displacement",
                                "Weight", "acceleration")])
print(correlation_coefficient)

## Correlation Matrix
# Calculate the correlation matrix for the first four numeric attributes
# Select only the relevant numeric columns
correlation_data <- data[, c("mpg.Y.",
                             "Cylinders",
                             "Displacement",
                             "HorsePower",
                             "Weight",
                             "acceleration",
                             "ModelYear",
                             "origin" )]

# Calculate the correlation matrix
correlation_matrix <- cor(correlation_data)

# Print the result
print(correlation_matrix)
cor(data[, 1:8])
```

The correlation matrix returned NA values due to missing data. To fix this, we used `cor(data, use = "complete.obs")` to ignore rows with missing values when calculating correlations.

# Cleaning of dataset:

Dimensions changes as rows were deleted to handle missing values.

R Gui

File Edit Packages Windows Help

R Console

```
mpg.Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
382 27 4 151 90 2950 17.3 82
383 27 4 140 86 2790 15.6 82
384 44 4 97 52 2130 24.6 82
385 32 4 135 84 2295 11.6 82
386 28 4 120 79 2625 18.6 82
387 31 4 119 82 2720 19.4 82
origin CarName
382 1 chevrolet camaro
383 1 ford mustang gl
384 2 vw pickup
385 1 dodge rampage
386 1 ford ranger
387 1 chevy s-10
> # Display the first 4 rows of the dataset
> tail(data,4)
mpg.Y. Cylinders Displacement HorsePower Weight acceleration ModelYear
384 44 4 97 52 2130 24.6 82
385 32 4 135 84 2295 11.6 82
386 28 4 120 79 2625 18.6 82
387 31 4 119 82 2720 19.4 82
origin CarName
384 2 vw pickup
385 1 dodge rampage
386 1 ford ranger
387 1 chevy s-10
>
> # Use the 'dim()' function to obtain the dimensions of the dataset
> # This returns a vector with the number of rows and columns in dataset
> dim(data)
[1] 387 9
>
```

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
# Load the dataset into the R environment
# Set the file path
file_path <- "C:/Users/User/Downloads/autompq.csv"

# Read the CSV file
data <- read.csv(file_path)

?data
data

# Display the default number of rows(1st 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)
|
# Use the 'names()' function to get the names of the columns in the d
<
```

Before Cleaning,  
it was 406.

# Summary of dataset:

RGui

File Edit Packages Windows Help

R Console

```
$ HorsePower : int 130 165 150 150 140 198 220 150 170 160 ...
$ Weight : int 3504 3693 3436 3433 3449 4341 4354 3433 3563 3609 ...
$ acceleration: num 12 11.5 11 12 10.5 10 9 12 10 8 ...
$ ModelYear : int 70 70 70 70 70 70 70 70 70 70 ...
$ origin : int 1 1 1 1 1 1 1 1 1 1 ...
$ CarName : chr "chevrolet chevelle malibu" "buick skylark 320" "plymo...
>
> # Use the 'summary()' function to generate a statistical summary of the da...
> # This provides a quick overview of the data, including min, max, mean, an...
> summary(data)
   mpg.Y.      Cylinders      Displacement      HorsePower
Min. : 9.00  Min. :3.00  Min. : 68.0  Min. : 46.0
1st Qu.:17.25 1st Qu.:4.00 1st Qu.:105.0 1st Qu.: 75.0
Median :22.50 Median :4.00 Median :151.0 Median : 93.0
Mean :23.48 Mean :5.47 Mean :193.6 Mean :103.9
3rd Qu.:29.00 3rd Qu.:8.00 3rd Qu.:264.5 3rd Qu.:125.0
Max. :46.60 Max. :8.00 Max. :455.0 Max. :230.0
   Weight      acceleration      ModelYear      origin
Min. :1613  Min. : 8.00  Min. :70.00  Min. :1.000
1st Qu.:2224 1st Qu.:13.80 1st Qu.:73.00 1st Qu.:1.000
Median :2807 Median :15.50 Median :76.00 Median :1.000
Mean :2974 Mean :15.58 Mean :76.06 Mean :1.571
3rd Qu.:3607 3rd Qu.:17.05 3rd Qu.:79.00 3rd Qu.:2.000
Max. :5140 Max. :24.80 Max. :82.00 Max. :3.000
   CarName
Length:387
Class :character
Mode :character
```

>

C:\Users\User\Downloads\Lecture 22.R - R Editor

```
# Display the default number of rows(1st 6 rows) from the dataset
head(data)
# Display the first 3 rows of the dataset
head(data,3)

# Display the default number of rows(last 6 rows) from the dataset
tail(data)
# Display the first 4 rows of the dataset
tail(data,4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the dataset
names(data)

# Check the structure of the iris dataset
str(data)

# Use the 'summary()' function to generate a statistical summary of the dataset
# This provides a quick overview of the data, including min, max, mean, median, etc.
summary(data)

# Missing Values in dataset
<
```

Type here to search

9:27 PM  
5/7/2025

# Handling missing values:

- Techniques used:
  - Delete rows.
- Mean imputation:
- Commands (# For mean imputation):
  - `data_cleaned$HorsePower[is.na(data_cleaned$HorsePower)] <- mean(data_cleaned$HorsePower, na.rm = TRUE)`
  - `data_cleaned$acceleration[is.na(data_cleaned$acceleration)] <- mean(data_cleaned$acceleration, na.rm = TRUE)`

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying statistical summaries for various columns in a dataset:

```
Min. : 9.00  Min. : 3.00  Min. : 68.0  Min. : 46.0
1st Qu.:17.25 1st Qu.:4.00 1st Qu.:105.0 1st Qu.: 75.0
Median :22.50 Median :4.00 Median :151.0 Median : 93.0
Mean   :23.48 Mean   :5.47 Mean   :193.6 Mean   :103.9
3rd Qu.:29.00 3rd Qu.:8.00 3rd Qu.:264.5 3rd Qu.:125.0
Max.   :46.60 Max.   :8.00 Max.   :455.0 Max.   :230.0
Weight acceleration ModelYear origin
Min. :1613  Min. : 8.00  Min. :70.00  Min. :1.000
1st Qu.:2224 1st Qu.:13.80 1st Qu.:73.00 1st Qu.:1.000
Median :2807 Median :15.50 Median :76.00 Median :1.000
Mean   :2974 Mean   :15.58 Mean   :76.06 Mean   :1.571
3rd Qu.:3607 3rd Qu.:17.05 3rd Qu.:79.00 3rd Qu.:2.000
Max.   :5140 Max.   :24.80 Max.   :82.00 Max.   :3.000
CarName
Length:387
Class :character
Mode  :character
```

The right window is the R Editor, showing R code for handling missing values:

```
tail(data)
# Display the first 4 rows of the dataset
tail(data, 4)

# Use the 'dim()' function to obtain the dimensions of the dataset
# This returns a vector with the number of rows and columns in dataset
dim(data)

# Use the 'names()' function to get the names of the columns in the dataset
names(data)

# Check the structure of the iris dataset
str(data)

# Use the 'summary()' function to generate a statistical summary of the dataset
# This provides a quick overview of the data, including min, max, mean, etc.
summary(data)

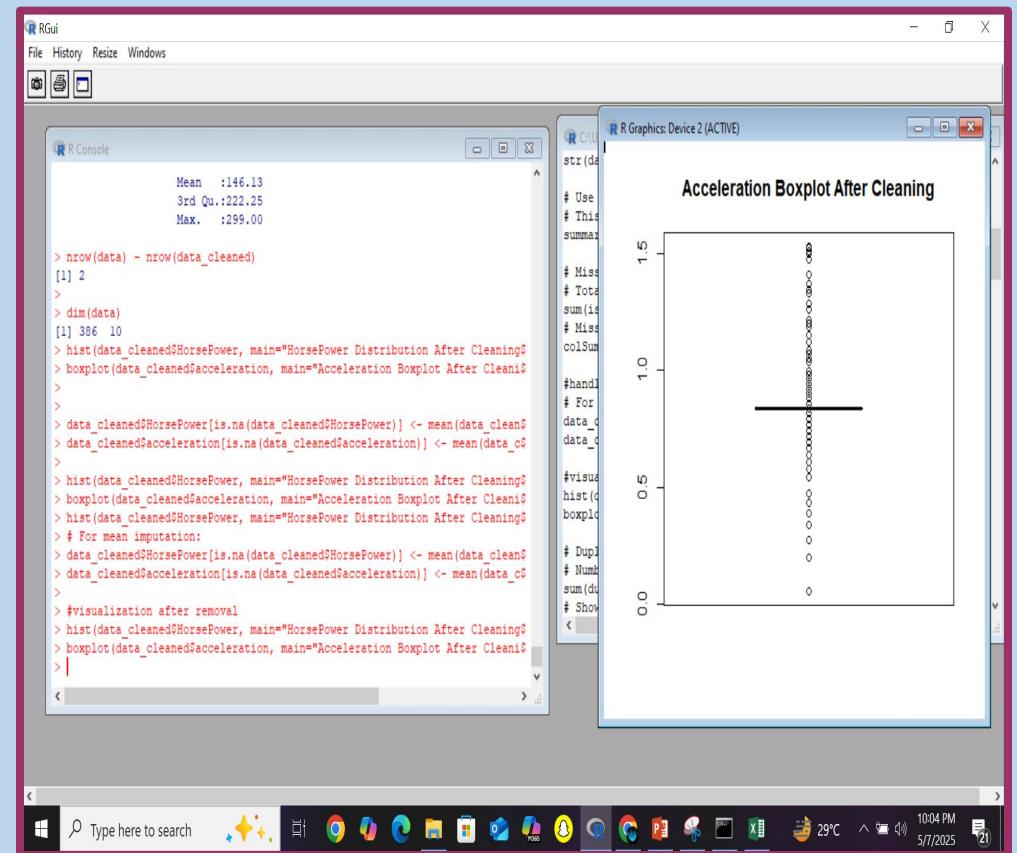
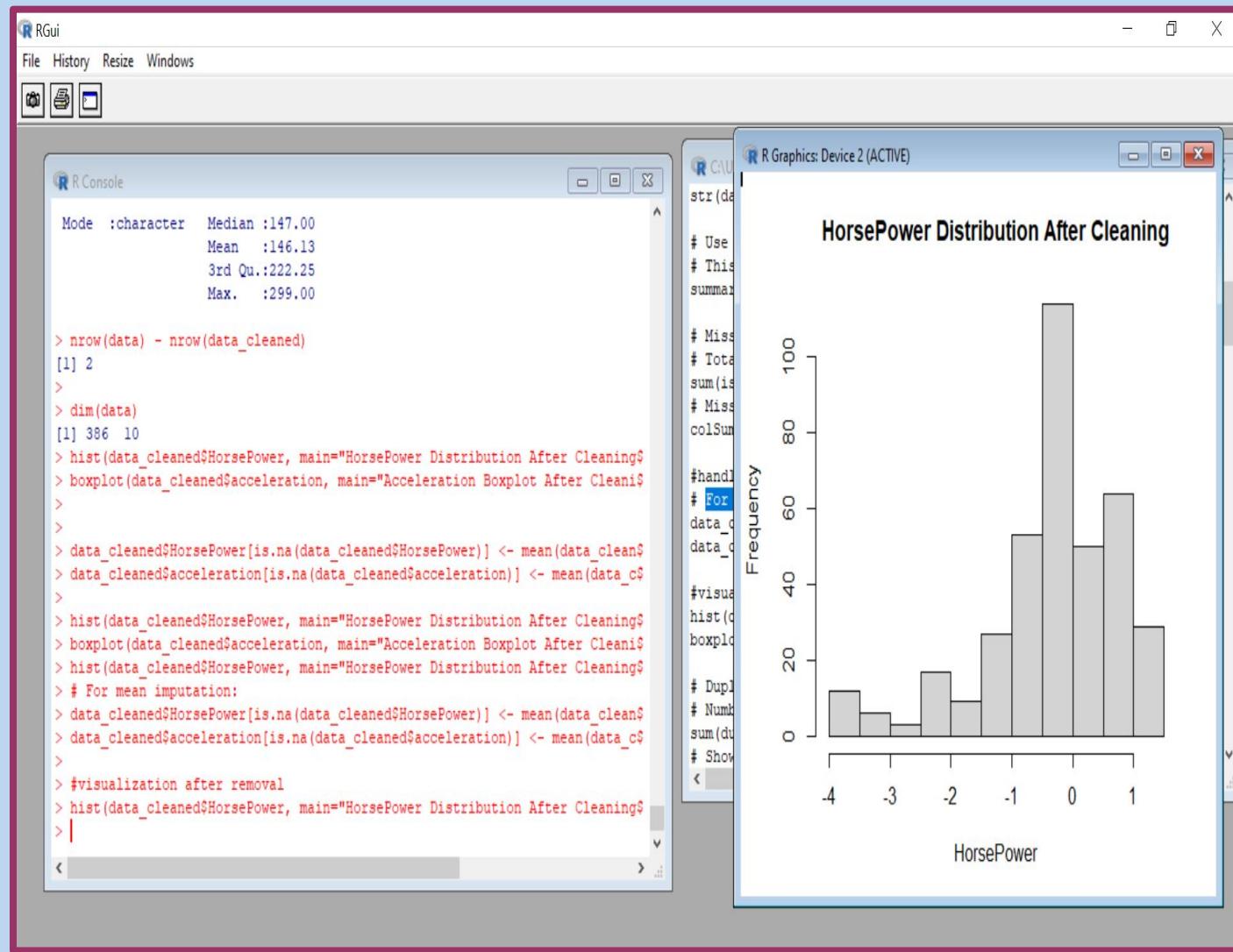
# Missing Values in dataset
# Total number of missing values in dataset
sum(is.na(data))
# Missing values per column in dataset
colSums(is.na(data))

# Duplicate rows in
```

The taskbar at the bottom of the screen shows various application icons, and the system tray indicates the date and time as 5/7/2025, 9:27 PM, with a temperature of 30°C.

# Visualization after removing missing values.

```
hist(data_cleaned$HorsePower, main="HorsePower Distribution After Cleaning", xlab="HorsePower")
```



```
boxplot(data_cleaned$acceleration,  
main="Acceleration Boxplot After  
Cleaning")
```

# Handling Duplicates:

Deleted rows having duplicate values.

The screenshot shows the RGui application window. The top menu bar includes File, Edit, View, Misc, Packages, Windows, and Help. Below the menu is a toolbar with various icons. The main area is divided into two panes: the R Console pane on the left and the R Editor pane on the right.

**R Console**

```
>
> sum(duplicated(data))
[1] 0
> # Shows duplicate rows in
> duplicated_data <- data[duplicated(data), ]
> duplicated_data
[1] mpg.Y.      Cylinders   Displacement HorsePower    Weight
[6] acceleration ModelYear   origin      CarName
<0 rows> (or 0-length row.names)
>
> |
```

**R Editor**

```
C:\Users\User\Downloads\Lecture 22.R - R Editor
# Use the 'summary()' function to generate a statistical summary of t! ^
# This provides a quick overview of the data, including min, max, mean
summary(data)

# Missing Values in dataset
# Total number of missing values in dataset
sum(is.na(data))
# Missing values per column in dataset
colSums(is.na(data))

# Duplicate rows in
# Number of duplicate rows in dataset
sum(duplicated(data))
# Shows duplicate rows in
duplicated_data <- data[duplicated(data), ]
duplicated_data

# Assuming your dataset is called data
data$CarName_numeric <- as.numeric(factor(data$CarName))

# Check how car names are mapped to numbers
levels(factor(data$CarName))

# View the result
```



# Removed Outlier by Increasing 1.5 to 2.4

The screenshot shows the RGui interface with two windows open:

- R Console:** Displays R code and its output. The code checks for outliers in various columns of the `mpg` dataset and creates corresponding data frames for each column. The output shows that all lengths are 0, indicating no outliers were found.
- R Editor:** Displays R code for outlier detection using the IQR method. It defines a function `find\_outliers` that takes a dataset as input. The function calculates the Q1 and Q3 quartiles, determines the IQR, and then finds outliers as values below  $Q1 - 2.4 \times IQR$  or above  $Q3 + 2.4 \times IQR$ . The code then applies this logic to each column of the `mpg` dataset to create separate data frames for each column. A red box highlights the line of code for calculating the upper bound: `upper_bound <- Q3 + 2.4 * IQR`.

# Standardization:

should not be done on categorial and discrete values.

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying R code and its output. The right window is the R Editor, displaying R code. The R Console output includes statistical summaries for continuous variables like mpg, Displacement, HorsePower, Weight, and acceleration. The R Editor code shows how to standardize these variables using the scale function.

```
R Gui
File Edit View Misc Packages Windows Help
R Console
Mean :146.05
3rd Qu.:222.75
Max. :299.00
> # Standardize by column names
> # Select only numeric continuous variables
> num_cols <- c("mpg.Y.", "Displacement", "HorsePower", "Weight", "accelerat$"
>
> # Standardize them
> data[num_cols] <- scale(data[num_cols])
>
> # Check summary again
> summary(data[num_cols])
   mpg.Y.   Displacement   HorsePower      Weight
Min. :-1.85413 Min. :-1.2152 Min. :-1.5361 Min. :-1.6051
1st Qu.:-0.76738 1st Qu.:-0.8563 1st Qu.:-0.7649 1st Qu.:-0.8841
Median :-0.09615 Median :-0.4101 Median :-0.2995 Median :-0.1997
Mean   : 0.00000 Mean   : 0.00000 Mean   : 0.00000 Mean   : 0.00000
3rd Qu.: 0.70293 3rd Qu.: 0.6667 3rd Qu.: 0.5648 3rd Qu.: 0.7500
Max.   : 2.95314 Max.   : 2.5389 Max.   : 3.3572 Max.   : 2.5585
   acceleration
Min.   :-2.79500
1st Qu.:-0.65029
Median :-0.03358
Mean   : 0.00000
3rd Qu.: 0.54632
Max.   : 3.39059
> |
<
```

```
R Editor
C:\Users\User\Downloads\Lecture 22.R - R Editor
# Check how car names are mapped to numbers
levels(factor(data$CarName))

# View the result
print(data)

# Standardize all 4 numeric features
data_scaled <- as.data.frame(scale(data[, 1:8]))
# View the result (standardized values for all 150 rows)
head(data_scaled, 406) # View first 10 rows

# Standardize by column names
# Select only numeric continuous variables
num_cols <- c("mpg.Y.", "Displacement", "HorsePower", "Weight", "acce.

# Standardize them
data[num_cols] <- scale(data[num_cols])

# Check summary again
summary(data[num_cols])

# Function to detect outliers using IQR method
<
```

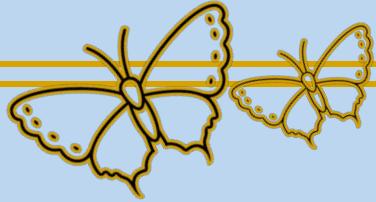
## Standardization:

- Standardize only continuous numeric variables (e.g., mpg.Y., Displacement, HorsePower, Weight, acceleration).
- These variables have meaningful magnitudes and benefit from mean = 0, SD = 1 transformation.

## Do not standardize:

- Categorical variables like origin → use one-hot encoding or factors.
- Text variables like CarName → extract brand or encode separately.
- Discrete or identifier-like variables:
  - Cylinders: Discrete, may be better as category.
  - ModelYear: Represents time, not magnitude.
  - CarName\_numeric: Just an ID, not meaningful.

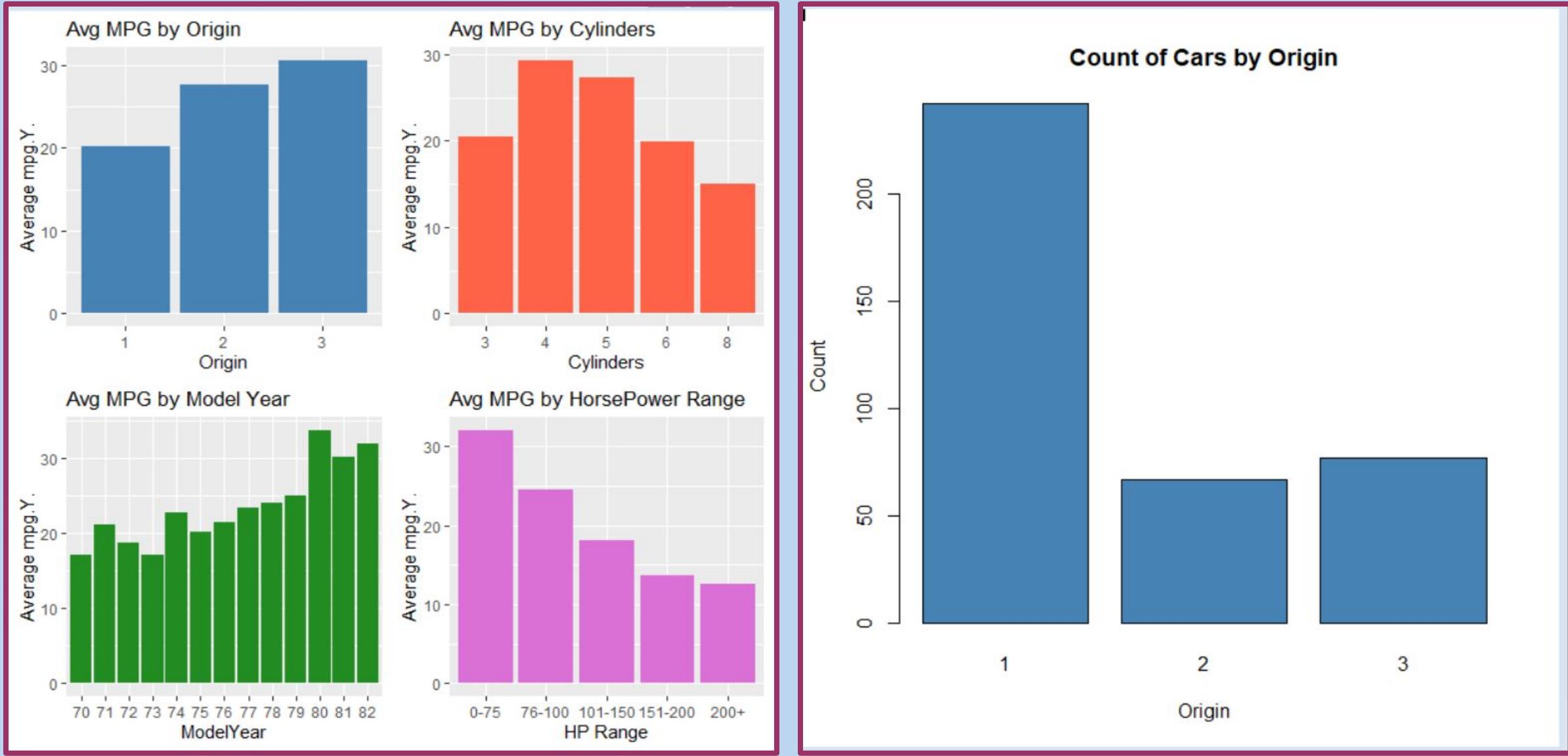
*Only standardize variables where numeric relationships are meaningful.*



## Task 5 : Exploratory Data Analysis (EDA)

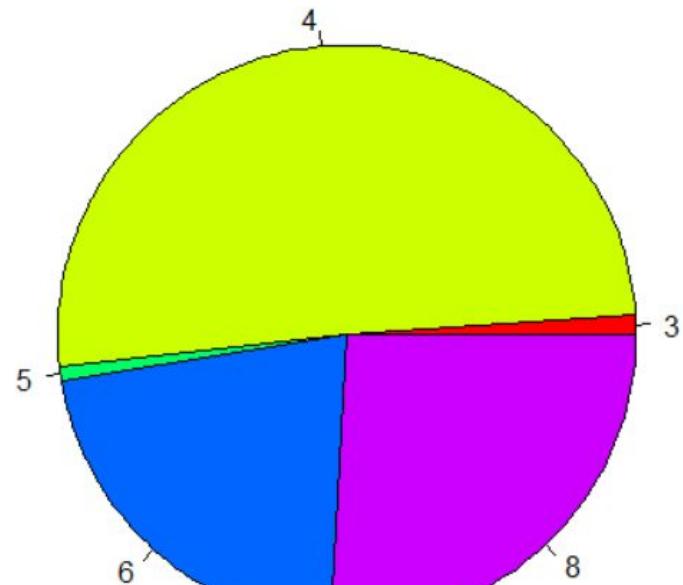


# Bar Graphs



# Pie Charts

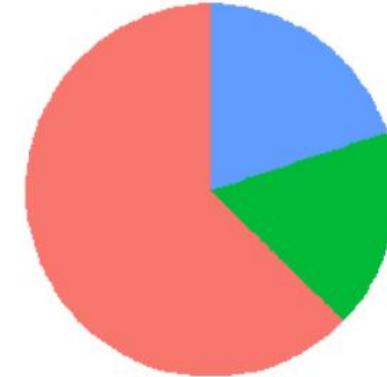
Distribution of Cylinders



Cylinders Distribution



Origin Distribution



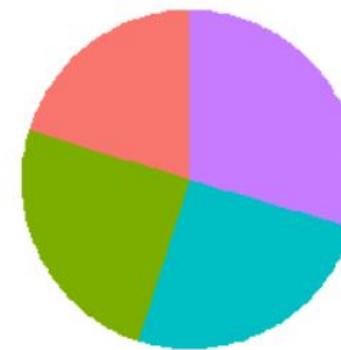
Origin

Origin	Color
1	Red
2	Green
3	Blue

HorsePower Ranges



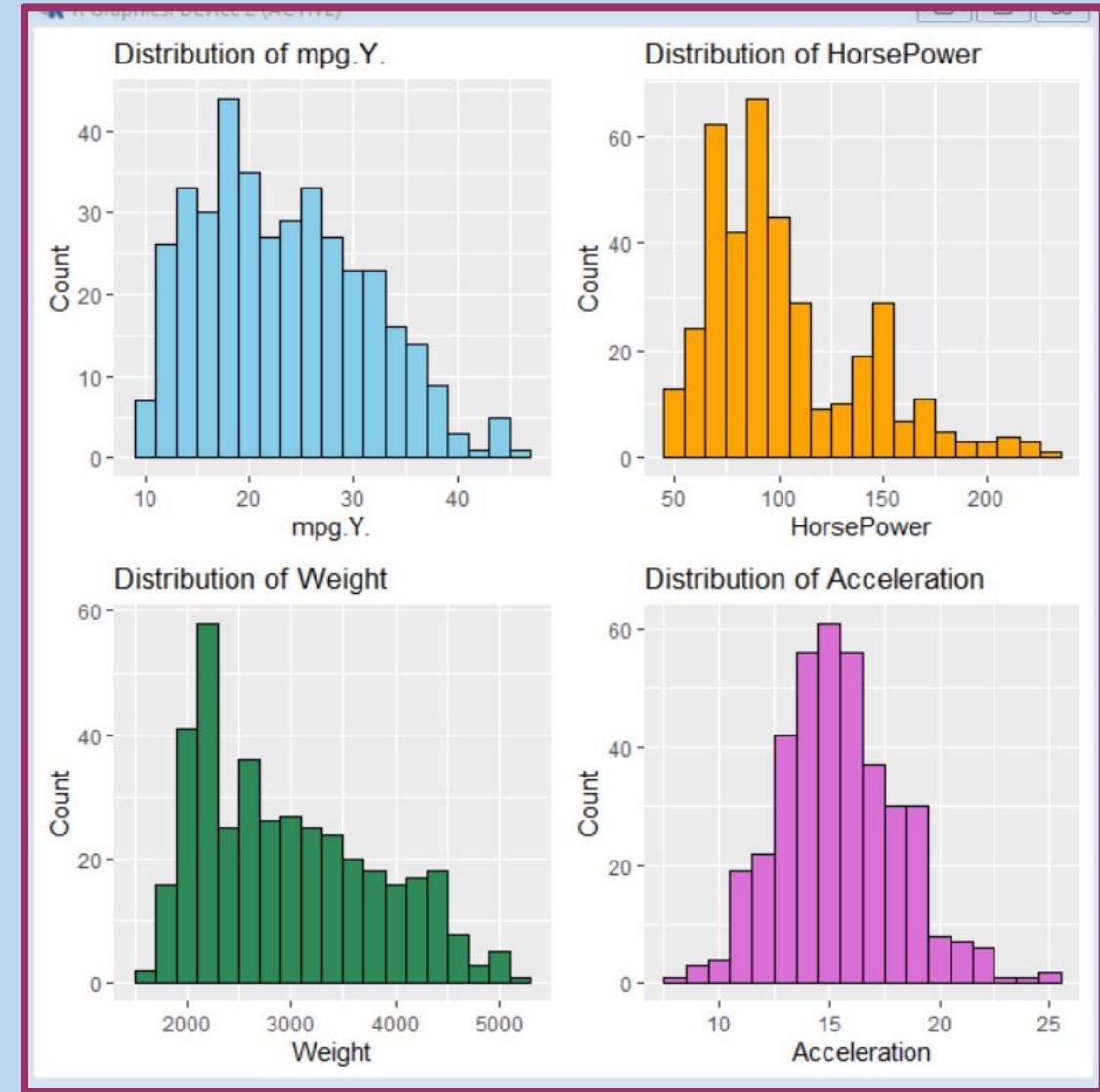
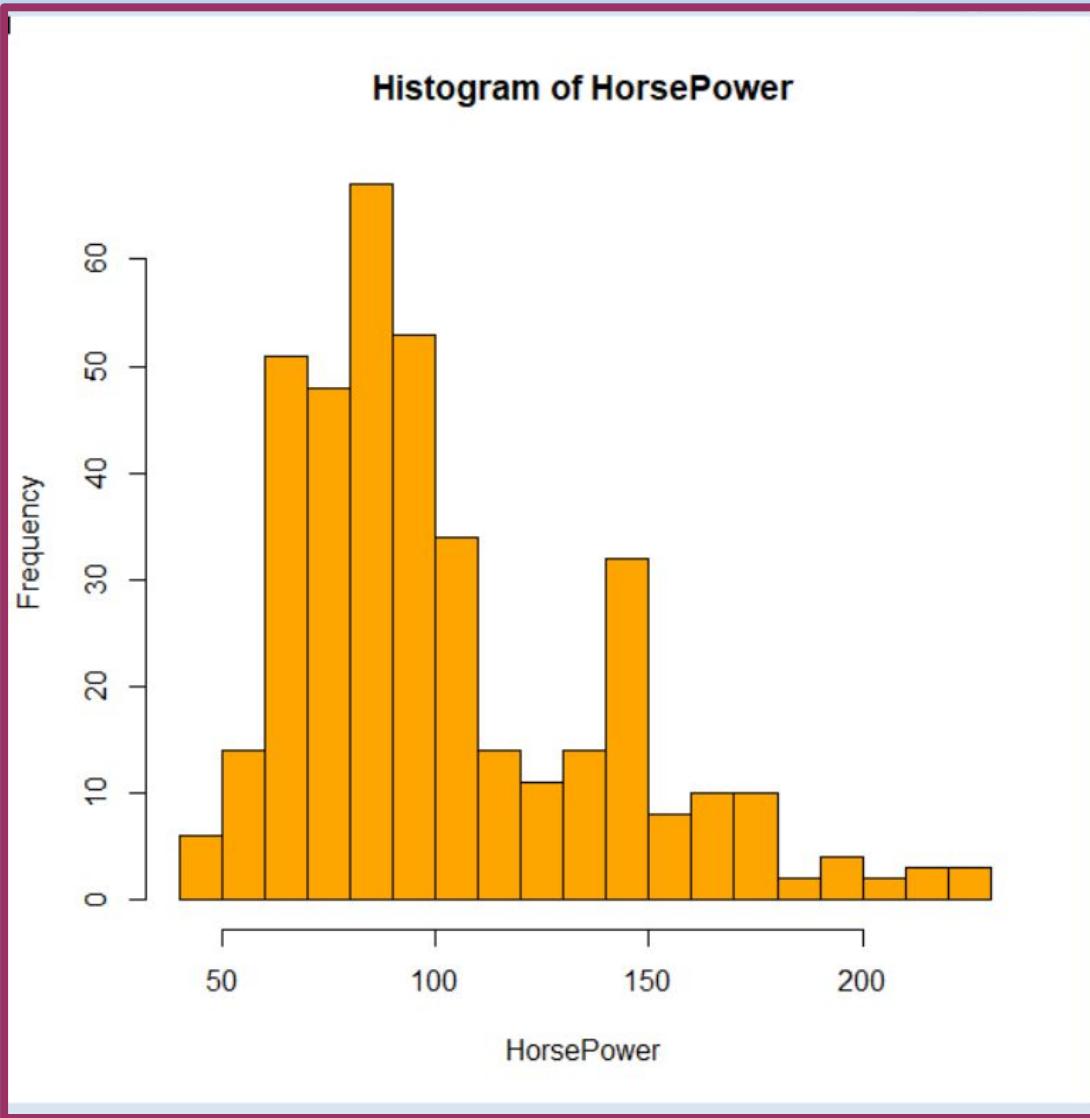
Model Year Bins



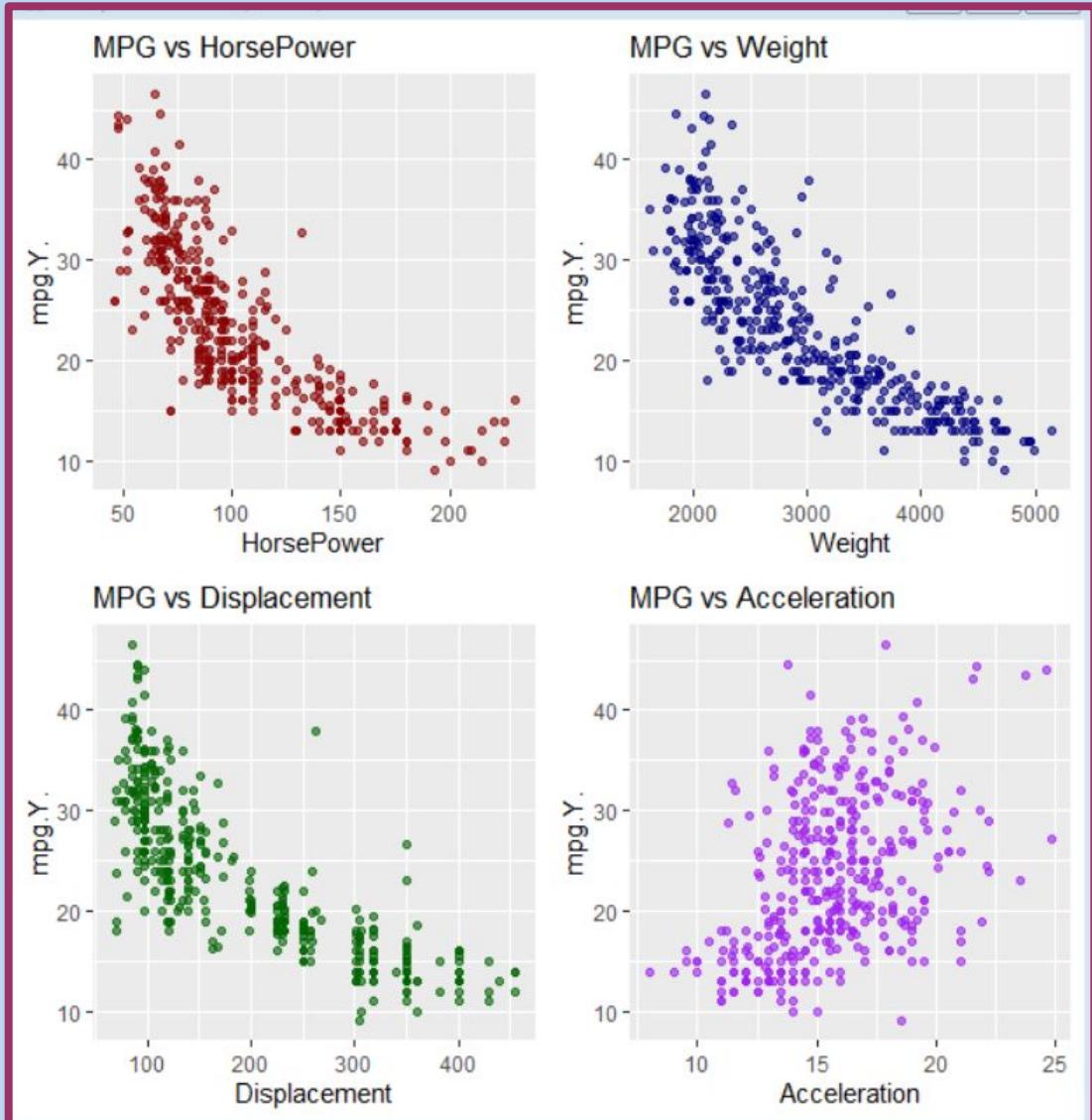
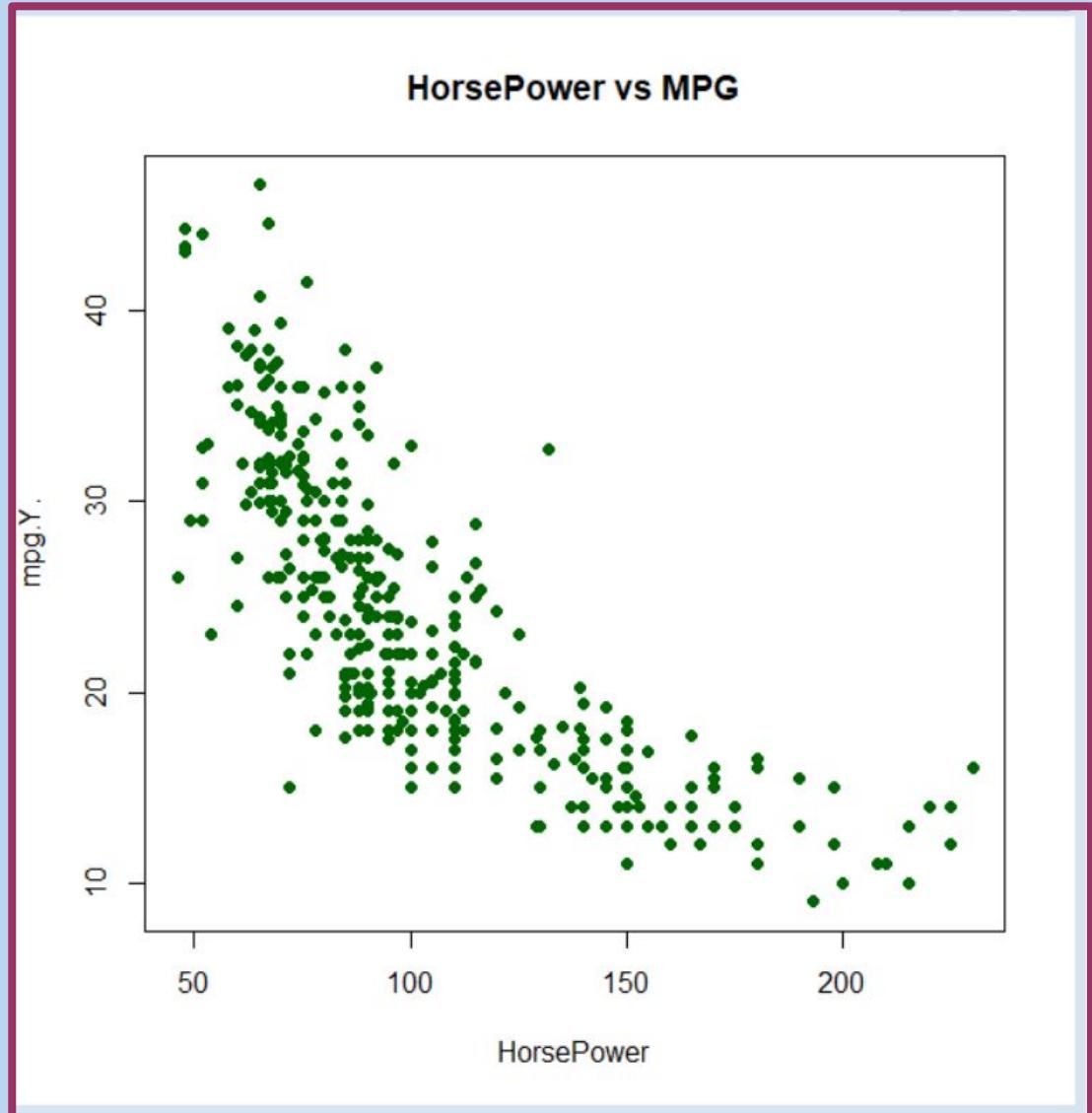
Years

Years	Color
70-72	Red
73-75	Green
76-78	Blue
79-82	Magenta

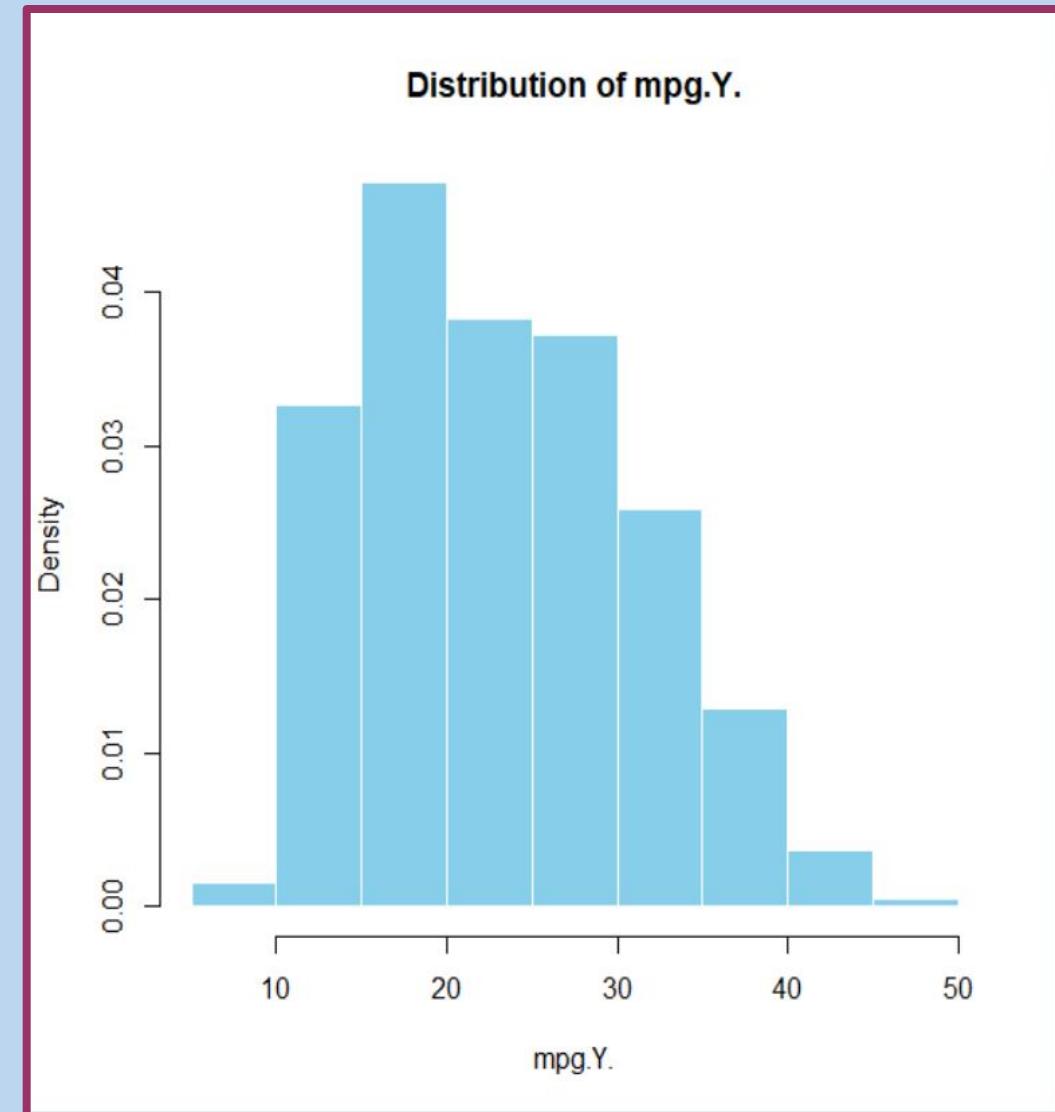
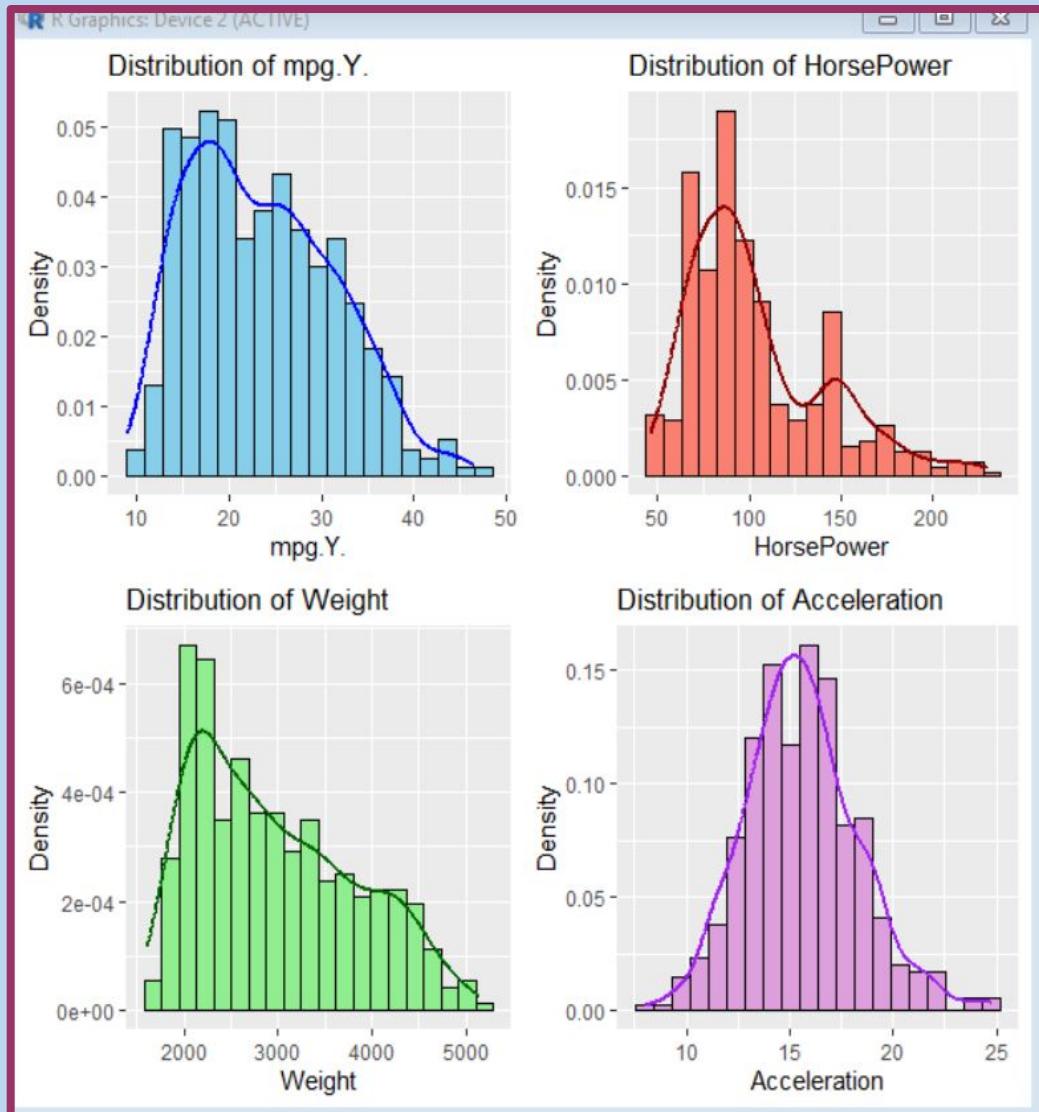
# Histogram



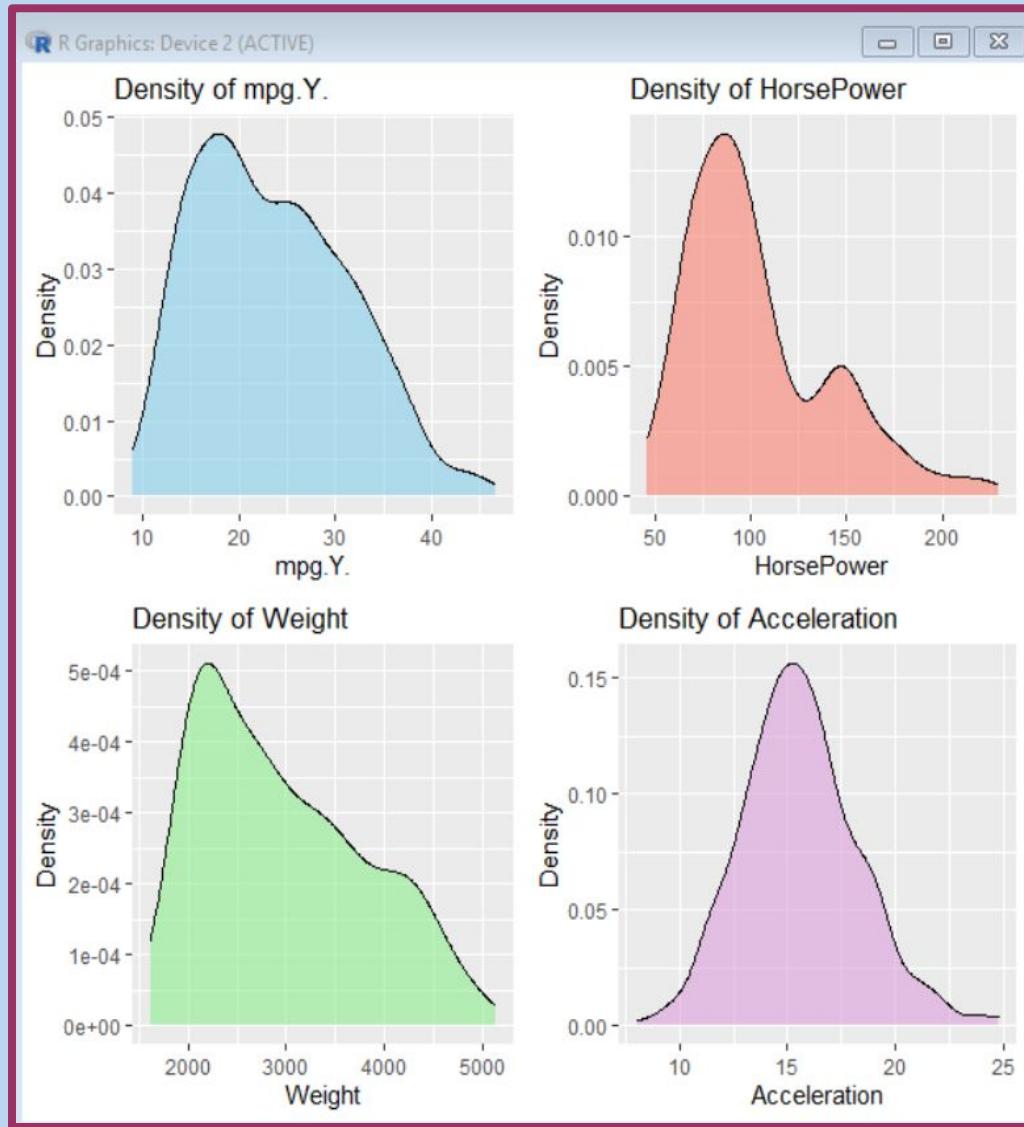
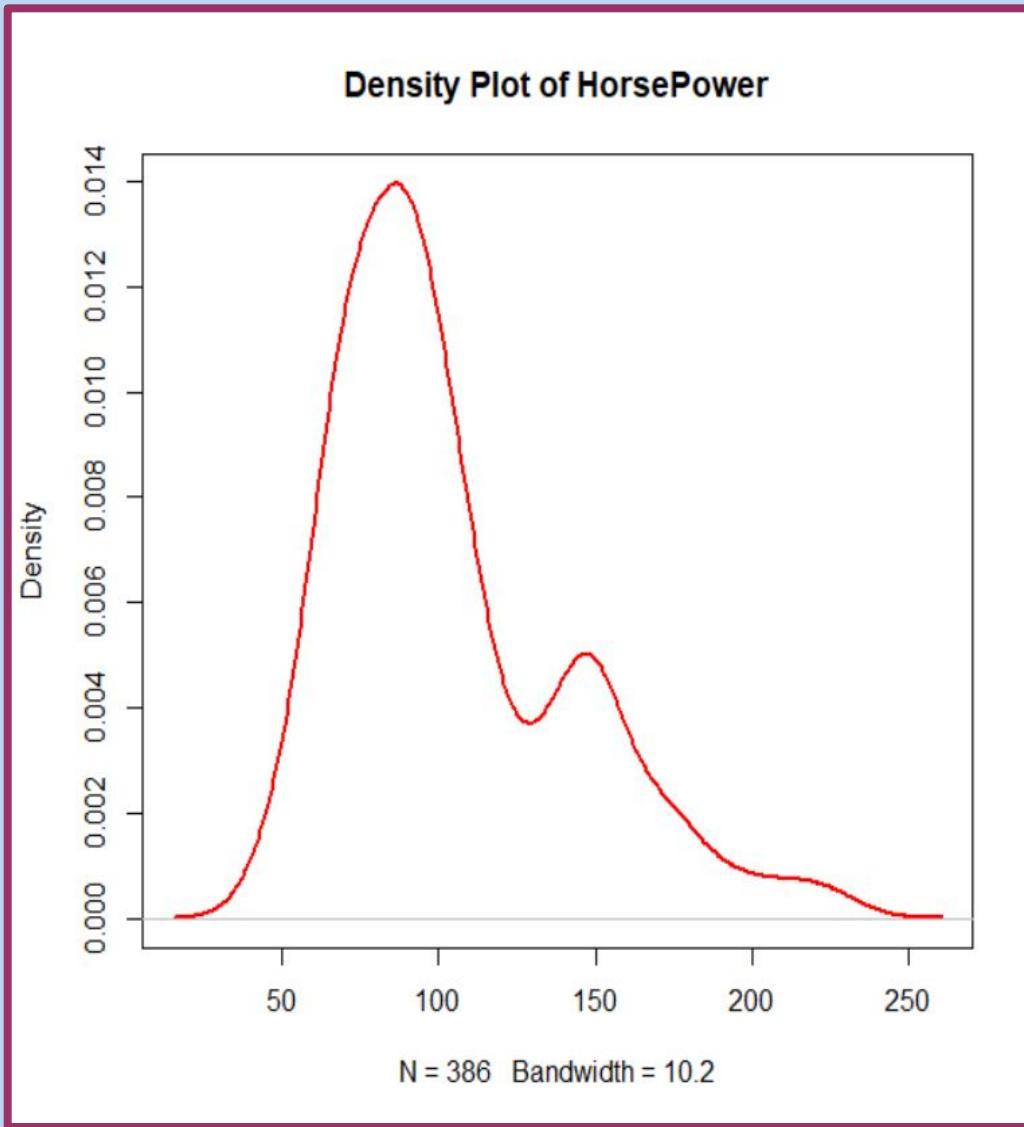
# Scatter Plots



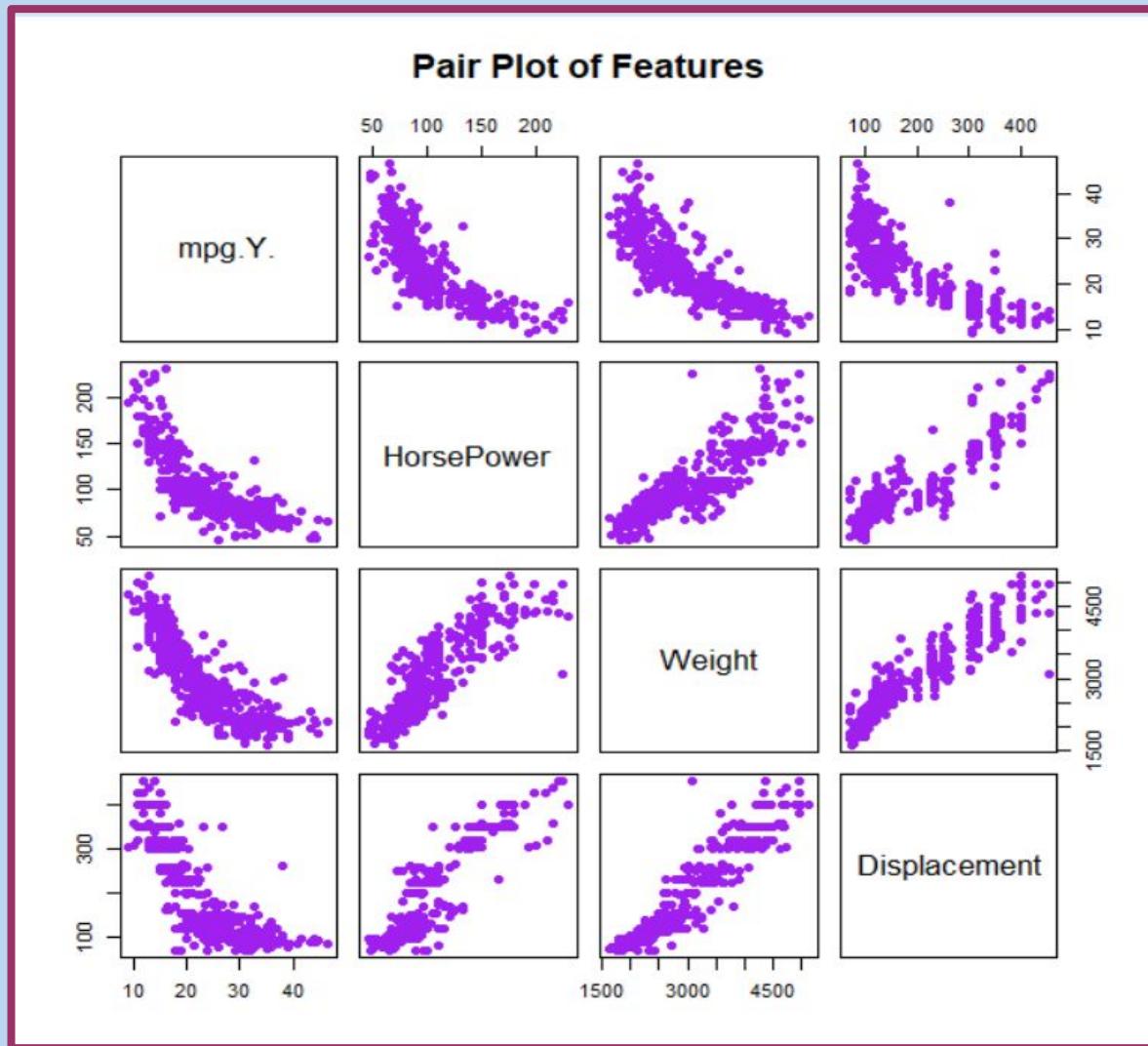
# Distribution Plot



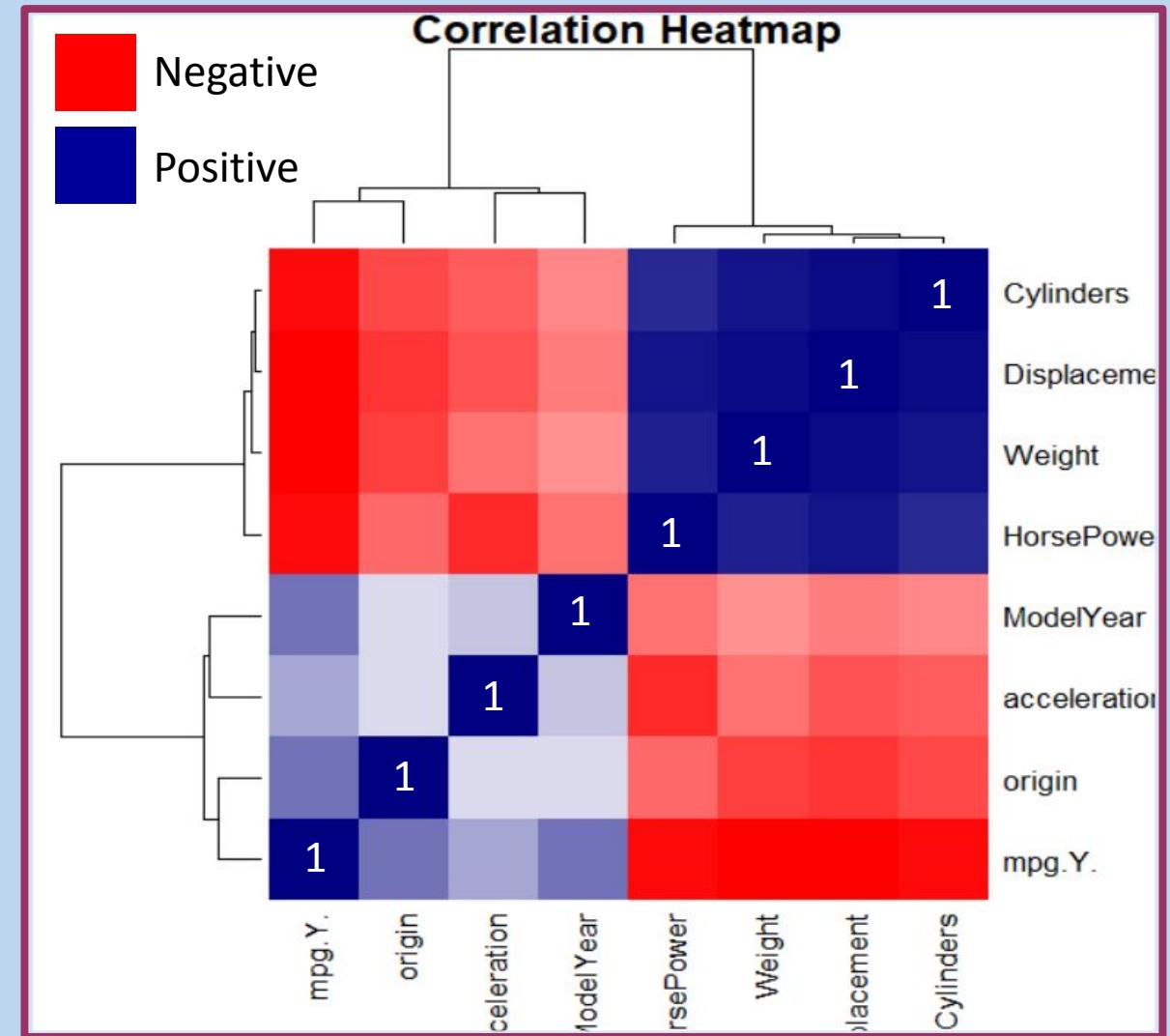
# Density Plot

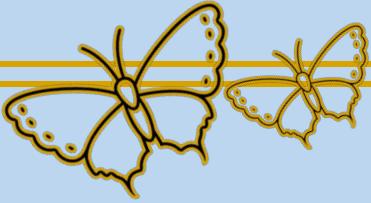


# Pair Plot



# Heatmap

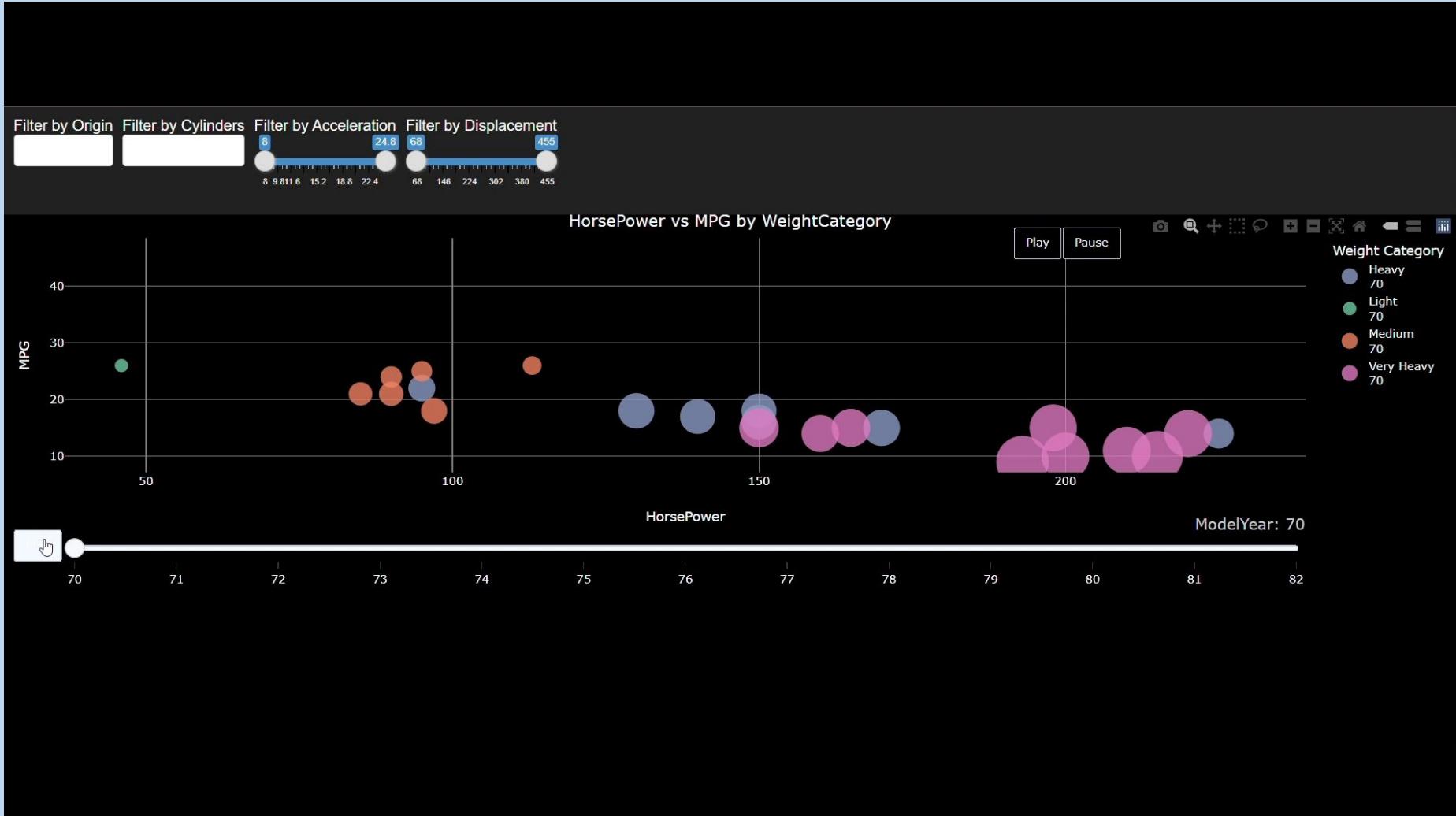


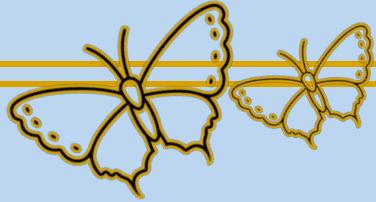


# Task 6 : Interactive Visualizations



# Interactive Scatter Plot





# Task 7 : Statistical Modeling and Inference



# Correlation Analysis:

The screenshot shows the RStudio interface with two panes. The left pane is the R Console, displaying R code and its output. The right pane is the R Editor, displaying R code for correlation analysis and simple linear regression.

**R Console Output:**

```
Weight      -0.8309833  0.8959934   0.9336499  0.8666997  1.0000000
acceleration 0.4139734 -0.4942482 -0.5268645 -0.6750382 -0.4051674
ModelYear     0.5841589 -0.3489369 -0.3679139 -0.4070861 -0.3092706
origin        0.5651836 -0.5640466 -0.6135072 -0.4550529 -0.5809339
               acceleration ModelYear      origin
mpg.Y.          0.4139734  0.5841589  0.5651836
Cylinders      -0.4942482 -0.3489369 -0.5640466
Displacement   -0.5268645 -0.3679139 -0.6135072
HorsePower     -0.6750382 -0.4070861 -0.4550529
Weight         -0.4051674 -0.3092706 -0.5809339
acceleration   1.0000000  0.2727357  0.2083816
ModelYear       0.2727357  1.0000000  0.1956379
origin          0.2083816  0.1956379  1.0000000
>
> correlation_matrix <- cor(correlation_data)
> print(correlation_matrix)
            mpg.Y. Cylinders Displacement HorsePower      Weight
mpg.Y.          1.0000000 -0.7758190 -0.8057877 -0.7809396 -0.8309833
Cylinders      -0.7758190  1.0000000  0.9517503  0.8443731  0.8959934
Displacement   -0.8057877  0.9517503  1.0000000  0.8936274  0.9336499
HorsePower     -0.7809396  0.8443731  0.8936274  1.0000000  0.8666997
Weight         -0.8309833  0.8959934  0.9336499  0.8666997  1.0000000
acceleration   0.4139734 -0.4942482 -0.5268645 -0.6750382 -0.4051674
ModelYear       0.5841589 -0.3489369 -0.3679139 -0.4070861 -0.3092706
origin          0.5651836 -0.5640466 -0.6135072 -0.4550529 -0.5809339
               acceleration ModelYear      origin
mpg.Y.          0.4139734  0.5841589  0.5651836
Cylinders      -0.4942482 -0.3489369 -0.5640466
Displacement   -0.5268645 -0.3679139 -0.6135072
HorsePower     -0.6750382 -0.4070861 -0.4550529
Weight         -0.4051674 -0.3092706 -0.5809339
acceleration   1.0000000  0.2727357  0.2083816
ModelYear       0.2727357  1.0000000  0.1956379
origin          0.2083816  0.1956379  1.0000000
>
> # Heatmap of the correlation matrix
> corrplot(correlation_matrix, method = "color", type = "upper", tl.col = "black$
```

**R Editor Content:**

```
# =====
# Task 7: Statistical Modeling and Inference
# =====

# STEP 1: Correlation Analysis
# Correlation Coefficient (single pair)
# Calculate the Pearson correlation coefficient between mpg.Y. and Weight
correlation_coefficient <- cor(data$mpg.Y., data$Weight, method = "pearson")
print(correlation_coefficient) # This now works

# Correlation Matrix (all relevant numeric columns)
correlation_data <- data[, c("mpg.Y.",
                             "Cylinders",
                             "Displacement",
                             "HorsePower",
                             "Weight",
                             "acceleration",
                             "ModelYear",
                             "origin")]

correlation_matrix <- cor(correlation_data)
print(correlation_matrix)

# Heatmap of the correlation matrix
corrplot(correlation_matrix, method = "color", type = "upper", tl.col = "black")

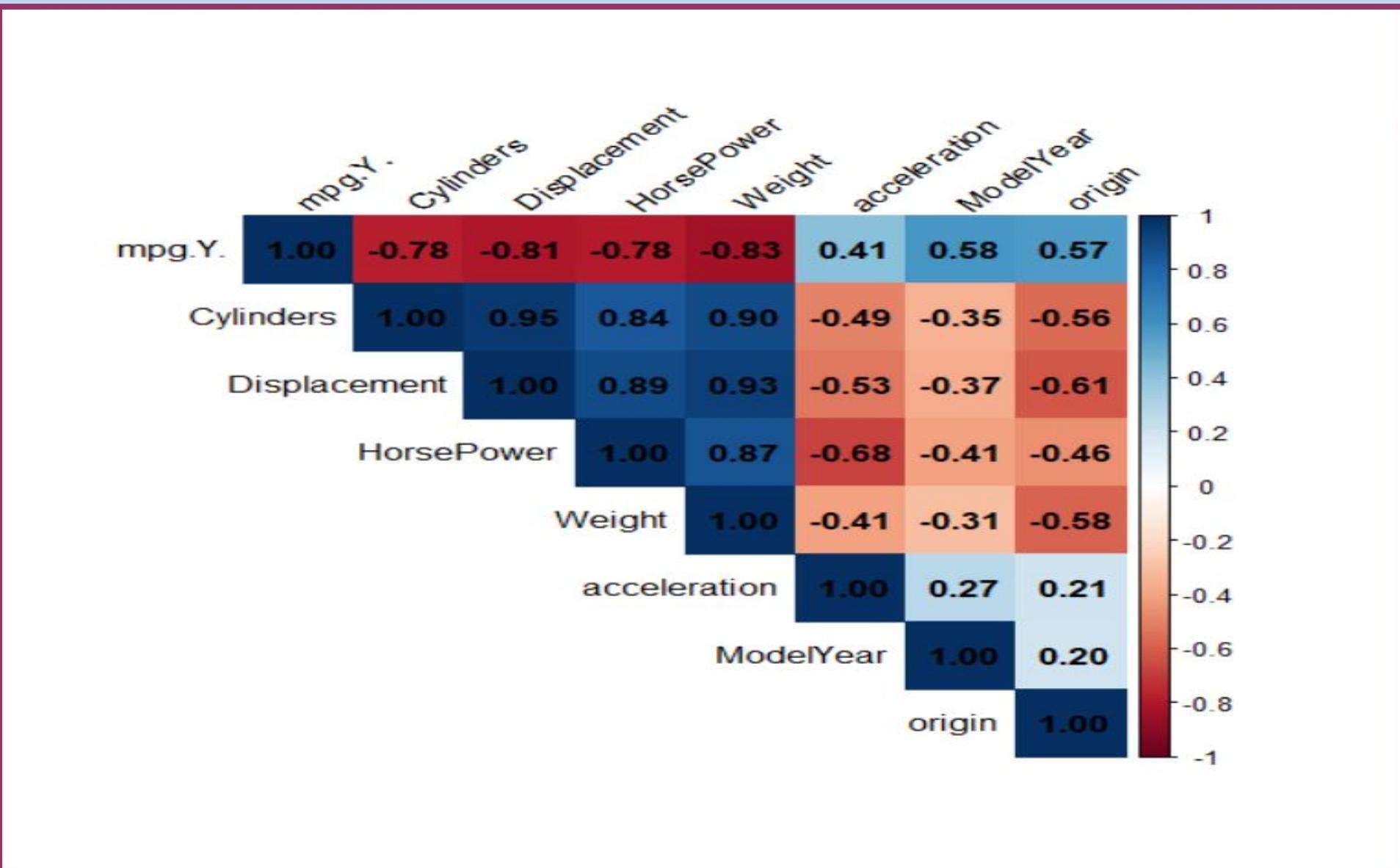
# STEP 2: Simple Linear Regression (SLR)
# Simple Linear Regression: mpg.Y. vs HorsePower
SLM <- lm(mpg.Y. ~ Cylinders, data = data)
SLM
summary(SLM) # Print the summary of the model to check
              # coefficients and statistical significance
SLM <- lm(mpg.Y. ~ acceleration, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ ModelYear, data = data)
SLM
```

- We analyzed how car attributes relate to fuel efficiency (mpg.Y.).
- Pearson correlation coefficients were calculated between mpg.Y. and other numeric variables.

## Top Findings:

- Weight and mpg.Y.: -0.83 → Heavier cars are less fuel efficient.
- HorsePower and mpg.Y.: -0.78 → More powerful engines = lower mpg.
- Model Year and mpg.Y.: +0.58 → Newer models are more efficient.
- Cylinders, Displacement, and HorsePower are highly interrelated ( $r > 0.85$ ), suggesting multicollinearity risk in modeling.

# Coefficient Correlation



# Summary

Variable	Correlation with mpg.Y.	Interpretation
Weight	-0.83	Heavier cars = lower fuel efficiency. Strong <b>negative</b> correlation.
Cylinders	-0.78	More cylinders = less fuel efficient. Strong <b>negative</b> correlation.
Displacement	-0.81	Bigger engine size = worse mileage. Strong <b>negative</b> correlation.
HorsePower	-0.78	Powerful engines = poor fuel efficiency. Strong <b>negative</b> correlation.
Acceleration	+0.41	Slight trend: quicker acceleration = better mpg, but it's weak.
ModelYear	+0.58	Newer cars tend to have <b>better mileage</b> . Medium positive correlation.
Origin	+0.56	Cars from some origins (e.g., Japan) have better mpg.

Variable Pair	Correlation	Meaning
Cylinders vs. Displacement	+0.95	More cylinders → bigger engines. VERY highly correlated.
HorsePower vs. Displacement	+0.89	More powerful cars tend to have larger engines.
Weight vs. Displacement	+0.93	Heavier cars often have bigger engines.

# Simple linear Regression:

```
RGui
File Edit Packages Windows Help
R Console
> # Step 2: Simple Linear Regression (SLR)
> # Simple Linear Regression: mpg.Y. vs HorsePower
> SLM <- lm(mpg.Y. ~ Cylinders, data = data)
> SLM

Call:
lm(formula = mpg.Y. ~ Cylinders, data = data)

Coefficients:
(Intercept) Cylinders
2.4942      -0.4565

> summary(SLM) # Print the summary of the model to check

Call:
lm(formula = mpg.Y. ~ Cylinders, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.82815 -0.40824 -0.08673  0.32650  2.28495 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.49419   0.10839   23.01 <2e-16 ***
Cylinders   -0.45650   0.01895  -24.09 <2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6318 on 384 degrees of freedom
Multiple R-squared:  0.6019, Adjusted R-squared:  0.6009 
F-statistic: 580.6 on 1 and 384 DF, p-value: < 2.2e-16

> # coefficients and statistical significance
> SLM <- lm(mpg.Y. ~ acceleration, data = data)
> SLM

Call:
lm(formula = mpg.Y. ~ acceleration, data = data)

Coefficients:
(Intercept) acceleration

```

```
R C:\Users\HP\Downloads\Lecture 22.R - R Editor
# Print the result
print(correlation_matrix)

# Alternatively, use a heatmap to visualize correlations
corrplot(correlation_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45, cex = 0.8)

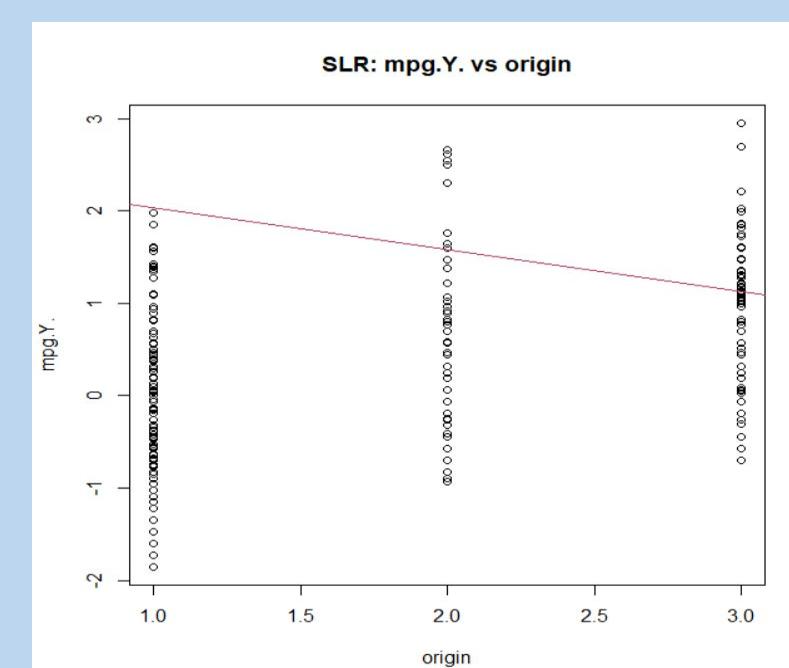
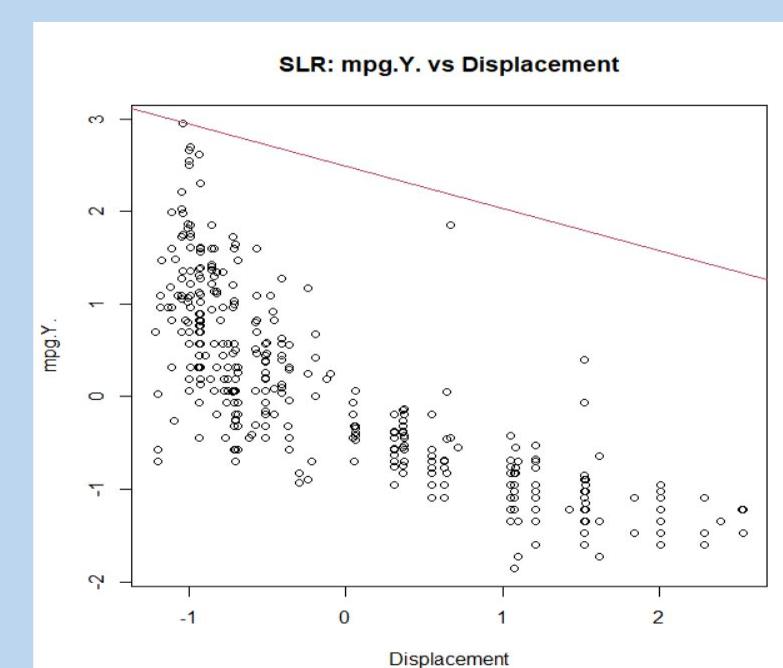
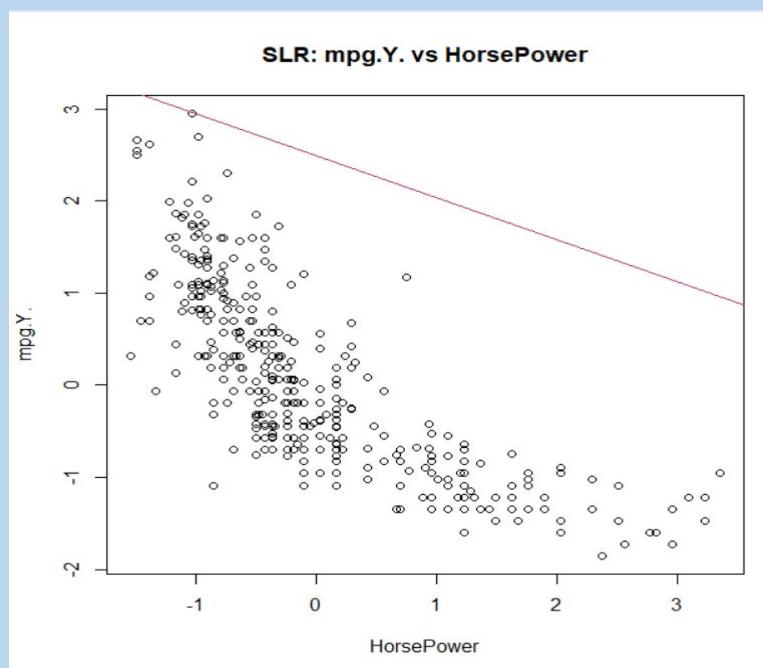
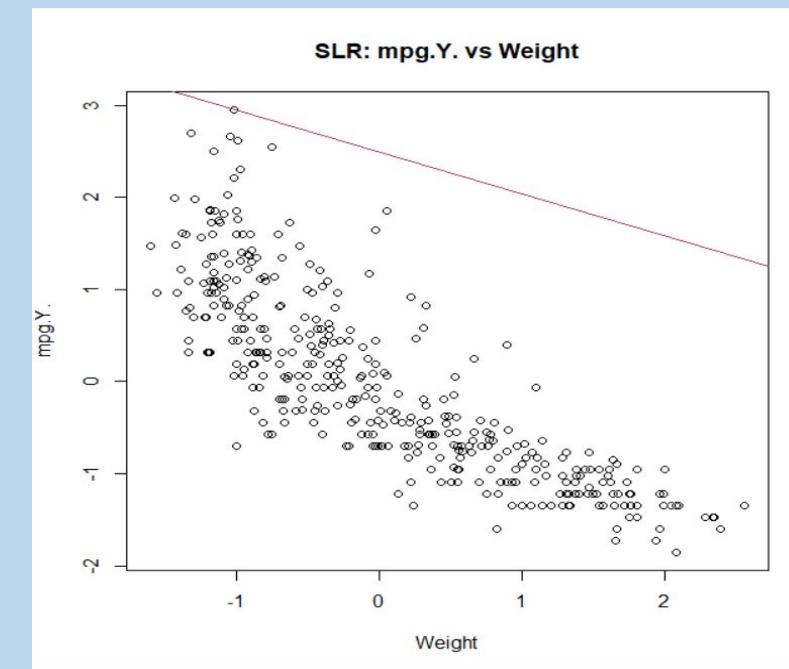
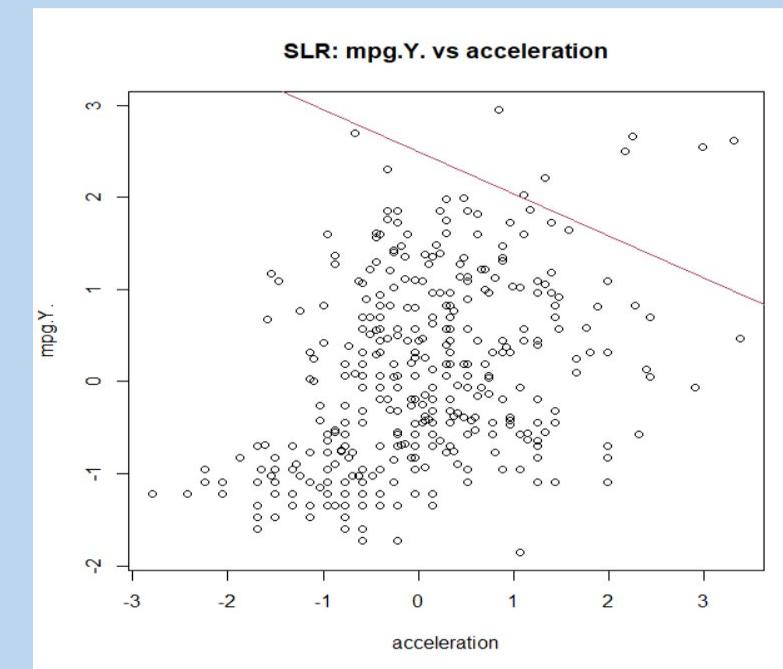
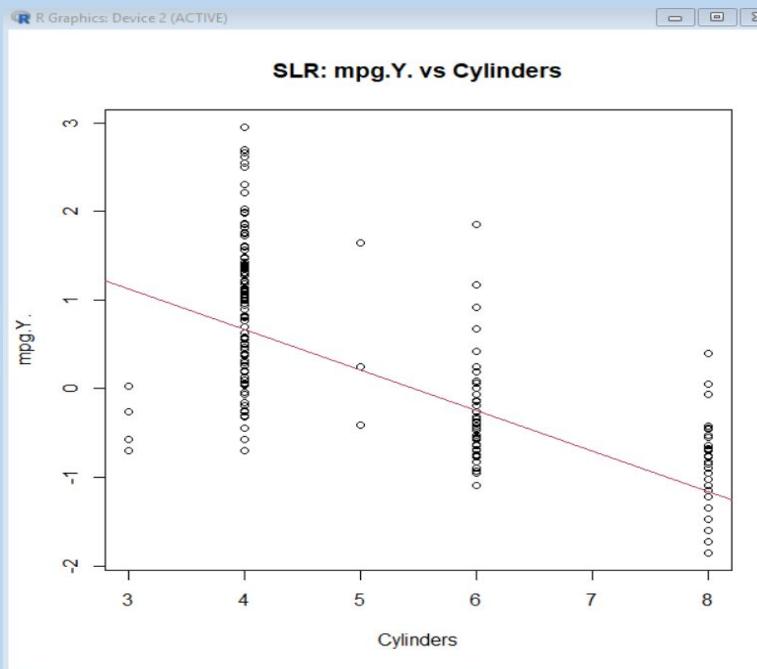
# Step 2: Simple Linear Regression (SLR)
# Simple Linear Regression: mpg.Y. vs HorsePower
SLM <- lm(mpg.Y. ~ Cylinders, data = data)
SLM
summary(SLM) # Print the summary of the model to check
# coefficients and statistical significance
SLM <- lm(mpg.Y. ~ acceleration, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ ModelYear, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ Weight, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ HorsePower, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ Displacement, data = data)
SLM
summary(SLM)
SLM <- lm(mpg.Y. ~ origin, data = data)
SLM
# Print the summary of the model
summary(SLM)

# CAN BE MODIFIED FOR OTHER VARIABES
# Plot the regression line
plot(data$Cylinders, data$mpg.Y.,
      main = "SLR: mpg.Y. vs Cylinders",
      xlab = "Cylinders",
      ylab = "mpg.Y.",
```

A series of simple linear regressions were run to predict mpg using different variables. **Weight** had the highest predictive power ( $R^2 = 0.693$ ) with a strong negative impact on mpg. **Displacement**, **HorsePower**, and **Cylinders** also showed strong negative relationships ( $R^2$  between 0.605–0.652). **Acceleration** had the weakest relationship ( $R^2 = 0.181$ ) and a positive coefficient. **ModelYear** and **Origin** positively influenced mpg, with  $R^2$  values of 0.351 and 0.322, respectively. All predictors were statistically significant ( $p < 0.001$ ).

# Simple Linear Regression Summary:

Predictor	Estimate (Slope)	R <sup>2</sup> Value	Adjusted R <sup>2</sup>	Significance (p-value)	Strength of Relationship
Cylinders	-0.4565	0.6019	0.6009	< 2e-16	Strong Negative
Acceleration	0.4140	0.1714	0.1692	< 2e-16	Weak Positive
Model Year	0.1607	0.3412	0.3395	< 2e-16	Weak Positive
Weight	-0.8310	0.6905	0.6897	< 2e-16	Strong Negative
Horsepower	-0.7809	0.6099	0.6089	< 2e-16	Strong Negative
Displacement	-0.8058	0.6493	0.6484	< 2e-16	Strong Negative
Origin	0.7035	0.3194	0.3177	< 2e-16	Moderate Positive



# Multiple linear regression model:

The screenshot shows the RGui interface with two windows open. The left window is the R Console, displaying the command history and output of a multiple linear regression model. The right window is the R Editor, showing the source code for the script.

R Console Output:

```
> MLM <- lm(mpg ~ Cylinders + Displacement + HorsePower + Weight + acceleration + ModelYear + origin, data = data)
>
> # Summary of the model to see coefficients and statistics
> summary(MLM)

Call:
lm(formula = mpg ~ Cylinders + Displacement + HorsePower +
    Weight + acceleration + ModelYear + origin, data = data)

Residuals:
    Min      1Q Median      3Q     Max 
-9.467 -2.146 -0.100  1.825 13.058 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.770e+01  4.670e+00 -3.791 0.000174 ***
Cylinders   -4.073e-01  3.225e-01 -1.263 0.207442  
Displacement 1.885e-02  7.561e-03  2.494 0.013062 *  
HorsePower   -1.851e-02  1.381e-02 -1.340 0.181129  
Weight       -6.468e-03  6.499e-04 -9.953 < 2e-16 ***
acceleration 7.443e-02  9.882e-02  0.753 0.451760  
ModelYear    7.571e-01  5.136e-02 14.740 < 2e-16 ***
origin       1.402e+00  2.784e-01  5.037 7.3e-07 *** 
---
Signif. codes:  0 '****' 0.001 '***' 0.01 '**' 0.05 '*' 0.1 ' ' 1

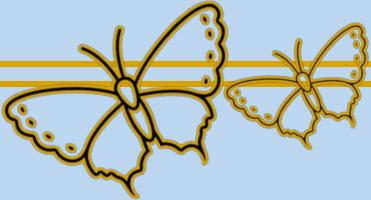
Residual standard error: 3.315 on 384 degrees of freedom
(14 observations deleted due to missingness)
Multiple R-squared:  0.8239, Adjusted R-squared:  0.8207 
F-statistic: 256.7 on 7 and 384 DF, p-value: < 2.2e-16
```

R Editor Content:

```
loads\Lecture 22.R - R Editor
~ HorsePower, data = data)
~ Displacement, data = data)
~ origin, data = data)
ary of the model

linear regression model
~ Cylinders + Displacement + HorsePower + Weight + a
model to see coefficients and statistics
```

A multiple linear regression was run to predict mpg. The model explained **82.07%** of the variance ( $R^2 = 0.8207$ ). **Weight**, **ModelYear**, and **Origin** were highly significant predictors ( $p < 0.001$ ), with **Weight** showing a strong negative effect and **ModelYear** a moderate positive one. **Displacement** was also significant ( $p = 0.013$ ), while **Cylinders**, **HorsePower**, and **Acceleration** were not statistically significant.



# Conclusion



- **MPG** (Miles Per Gallon) is a measure of a vehicle's fuel efficiency, the higher the MPG, the more distance a car can travel using less fuel.
- Using **Multiple Linear Regression**, we analyzed how various car features affect MPG.
- **Variables included:**

- Cylinders
- Displacement
- HorsePower
- Weight
- Acceleration
- ModelYear
- Origin

A1 : X ✓ fx v mpg(Y)

A	B	C	D	E	F	G	H	I	J	K
1	mpg(Y)	Cylinders	Displacem	HorsePow	Weight	acceleratio	ModelYear	origin	CarName	
2	18	8	307	130	3504	12	70	1	chevrolet chevelle malibu	
3	15	8	350	165	3693	11.5	70	1	buick skylark 320	
4	18	8	318	150	3436	11	70	1	plymouth satellite	
5	17	8	302	140	3449	10.5	70	1	ford torino	
6	15	8	429	198	4341	10	70	1	ford galaxie 500	
7	14	8	454	220	4354	9	70	1	chevrolet impala	
8	16	8	304	150	3433	12	70	1	amc rebel sst	
9	15	8	383	170	3563	10	70	1	dodge challenger se	
10	14	8	340	160	3609	8	70	1	plymouth 'cuda 340	
11	15	8	400	150	3761	9.5	70	1	chevrolet monte carlo	
12	14	8	455	225	3086	10	70	1	buick estate wagon (sw)	
13	22	6	198	95	2833	15.5	70	1	plymouth duster	
14	18	6	199	97	2774	15.5	70	1	amc hornet	
15	21	6	200	85	2587	16	70	1	ford maverick	



## Key Findings:

- Cylinders:  $-0.4565 \rightarrow$  Each additional cylinder in the engine decreases MPG by approximately 0.4565. (Interpretation: Strong Negative Relationship — more cylinders = less fuel efficiency.)
- Acceleration:  $0.4140 \rightarrow$  For each additional unit of acceleration, MPG increases by about 0.4140. (Interpretation: Weak Positive Relationship — slightly better MPG with faster acceleration.)
- Model Year:  $0.1607 \rightarrow$  Each year increase in model year improves MPG by 0.1607. (Interpretation: weak Positive — newer cars tend to be more fuel-efficient.)
- Weight:  $-0.8310 \rightarrow$  For every additional unit of weight, MPG drops by 0.8310. (Interpretation: Very Strong Negative — heavier cars burn more fuel.)
- Horsepower:  $-0.7809 \rightarrow$  Each unit increase in horsepower reduces MPG by 0.7809. (Interpretation: Strong Negative — more power, more fuel consumption.)
- Displacement:  $-0.8058 \rightarrow$  For every extra unit of engine displacement, MPG decreases by 0.8058. (Interpretation: Very Strong Negative — bigger engines are less efficient.)
- Origin:  $0.7035 \rightarrow$  Vehicles from different origins (regions) increase MPG by about 0.7035 per unit change. (Interpretation: Moderate Positive — cars from some origins (e.g., Japan) may be more fuel-efficient.)

## CONCLUSION:

The multiple linear regression model developed to predict miles per gallon (MPG) based on various vehicle characteristics—including weight, horsepower, displacement, cylinders, acceleration, and model year, demonstrates strong predictive performance.

With an  $R^2$  value of **0.8184**, the model explains approximately **82%** of the **variability in MPG**, indicating a strong linear relationship between the predictors and the response variable.

Among the predictors, Weight and Cylinders showed the most significant individual impact on fuel efficiency. The model suggests that **lighter** and **newer** vehicles generally offer better MPG.

Overall, this regression analysis provides valuable insights into the key factors affecting a car's fuel economy.

