Automobiles Project

Babalola Oluwatosin

2024-03-05

Overview of Project

This project offers a comprehensive glimpse into the world of automobiles, stimulating real-world task of a junior data analyst. Following a structured approach aligning with the data analysis process which are: ASK, PREPARE, PROCESS, ANALYZE, SHARE, ACT. I am utilizing a public dataset spanning from 2010 to 2020. The analysis is conducted using the R programming language, leveraging its capabilities in data cleaning, analysis and visualization.

Ask

This phase involves asking the right question.

Prepare

1. Car's historical data was made public by Motivate International Inc. The data can be accessed here under the license
2. I checked for issues with bias or credibility in this data using the ROCCC process to check if the data is Reliabile, Original, Comprehensive, Current and Cited.
3. I downloaded the dataset and I will be considering data within 2010 and 2020. I unzipped it, changed its format to .xls(excel file) from its original CVS file.

Loading the Required Packages

```
library(tidyverse)
library(ggplot2)
library(dplyr)
library(janitor)
library(lubridate)
library(readxl)
```

Importing the Data

```
library(readxl)
CarsData <- read_excel("~/Project/CarsData.xlsx")
View(CarsData)
```

PROCESS

I used R programming in my data cleaning process.

Rows and Column number

```
dim(CarsData)

## [1] 97712    10
```

Column names and Data type

```
str(CarsData)

## tibble [97,712 × 10] (S3: tbl_df/tbl/data.frame)
##  $ model       : chr [1:97712] "I10" "Polo" "2 Series" "Yeti Outdoor" ...
##  $ year        : num [1:97712] 2017 2017 2019 2017 2017 ...
##  $ price       : num [1:97712] 7495 10989 27990 12495 7999 ...
##  $ transmission: chr [1:97712] "Manual" "Manual" "Semi-Auto" "Manual" ...
##  $ mileage     : num [1:97712] 11630 9200 1614 30960 19353 ...
##  $ fuelType    : chr [1:97712] "Petrol" "Petrol" "Diesel" "Diesel" ...
##  $ tax         : num [1:97712] 145 145 145 150 125 135 145 145 145 30 ...
##  $ mpg         : num [1:97712] 60.1 58.9 49.6 62.8 54.3 74.3 34.4 30.4
65.7 62.8 ...
##  $ engineSize  : num [1:97712] 1 1 2 2 1.2 1.8 1.5 2 1 2.1 ...
##  $ Manufacturer: chr [1:97712] "hyundi" "volkswagen" "BMW" "skoda" ...
```

Check for null values in each column

```
null_values_per_column <- colSums(is.na(CarsData))
print(null_values_per_column)

##        model         year        price transmission      mileage
fuelType
##            0            0            0            0            0
0
##          tax          mpg   engineSize Manufacturer
##            0            0            0            0
```

Check for null values in each row

```
null_values_per_row <- rowSums(is.na(CarsData))
```

Column Names

```
colnames(CarsData)
```

```
##  [1] "model"        "year"         "price"        "transmission" "mileage"
##  [6] "fuelType"     "tax"          "mpg"          "engineSize"
"Manufacturer"
```

Count the number of distinct models in the dataset

```
num_distinct_models <- CarsData %>%
  distinct(model) %>%
  nrow()
cat("Number of distinct models:", num_distinct_models, "\n")

## Number of distinct models: 195

distinct_models <- unique(CarsData$model)
print(distinct_models)

##   [1] "I10"                        "Polo"                      "2 Series"
##   [4] "Yeti Outdoor"               "Fiesta"                    "C-HR"
##   [7] "Kuga"                       "Tiguan"                    "A Class"
##  [10] "1 Series"                   "Up"                        "Golf"
##  [13] "Corsa"                      "RAV4"                      "GLA Class"
##  [16] "Aygo"                       "Q5"                        "Karoq"
##  [19] "Scala"                      "Auris"                     "Tucson"
##  [22] "A4"                         "Viva"                      "Kodiaq"
##  [25] "C Class"                    "Mondeo"                    "Citigo"
##  [28] "Yaris"                      "X4"                        "Octavia"
##  [31] "Astra"                      "Focus"                     "3 Series"
##  [34] "GLC Class"                  "Q3"                        "B-MAX"
##  [37] "C-MAX"                      "IX20"                      "X5"
##  [40] "T-Cross"                    "Shuttle"                   "Insignia"
##  [43] "Zafira"                     "A3"                        "A5"
##  [46] "SL CLASS"                   "EcoSport"                  "X1"
##  [49] "Fabia"                      "Golf SV"                   "Verso"
##  [52] "Yeti"                       "Mokka X"                   "Antara"
##  [55] "E Class"                    "4 Series"                  "Superb"
##  [58] "5 Series"                   "8 Series"                  "B Class"
##  [61] "Ka+"                        "X2"                        "GLE Class"
##  [64] "A6"                         "Mokka"                     "Passat"
##  [67] "Kamiq"                      "Adam"                      "Q7"
##  [70] "Tiguan Allspace"            "X3"                        "A1"
##  [73] "Grandland X"                "Meriva"                    "Tourneo Connect"
##  [76] "Arteon"                     "TT"                        "GLS Class"
##  [79] "Santa Fe"                   "I30"                       "S Class"
##  [82] "Ioniq"                      "Edge"                      "S-MAX"
##  [85] "SLK"                        "Crossland X"               "7 Series"
##  [88] "T-Roc"                      "Q2"                        "CL Class"
##  [91] "CLA Class"                  "6 Series"                  "V Class"
##  [94] "Scirocco"                   "i3"                        "Grand C-MAX"
##  [97] "SQ5"                        "X7"                        "Corolla"
## [100] "A7"                         "Touareg"                   "CLS Class"
```

```
## [103] "I20"                      "M Class"                "Prius"
## [106] "KA"                       "GT86"                   "Hilux"
## [109] "Galaxy"                   "M4"                     "I800"
## [112] "Kona"                     "Touran"                 "Grand Tourneo
Connect"
## [115] "Caravelle"                "Combo Life"             "GL Class"
## [118] "Avensis"                  "SQ7"                    "GLB Class"
## [121] "RS3"                      "IX35"                   "GTC"
## [124] "Land Cruiser"             "X6"                     "RS5"
## [127] "Puma"                     "CC"                     "I40"
## [130] "i8"                       "Eos"                    "Rapid"
## [133] "Amarok"                   "Beetle"                 "Supra"
## [136] "California"               "A8"                     "Z4"
## [139] "Q8"                       "S4"                     "Sharan"
## [142] "Mustang"                  "M3"                     "RS4"
## [145] "RS6"                      "Fox"                    "Cascada"
## [148] "M5"                       "Caddy Maxi Life"        "Vivaro"
## [151] "X-CLASS"                  "M6"                     "Kadjar"
## [154] "Caddy Maxi"               "Fusion"                 "Tourneo Custom"
## [157] "Tigra"                    "M2"                     "Agila"
## [160] "Zafira Tourer"            "Vectra"                 "Ranger"
## [163] "Getz"                     "R8"                     "Roomster"
## [166] "Jetta"                    "Veloster"               "S5"
## [169] "S3"                       "Z3"                     "Ampera"
## [172] "Caddy Life"               "Urban Cruiser"          "S8"
## [175] "Verso-S"                  "IQ"                     "CLK"
## [178] "PROACE VERSO"             "R Class"                "G Class"
## [181] "180"                      "Camry"                  "Caddy"
## [184] "Terracan"                 "Streetka"               "200"
## [187] "Escort"                   "Transit Tourneo"        "CLC Class"
## [190] "230"                      "A2"                     "Amica"
## [193] "RS7"                      "Accent"                 "220"
```

Get the distinct values for the transmission column

```
num_distinct_trans <- CarsData %>%
  distinct(transmission) %>%
  nrow()
cat("Number of distinct transmission:", num_distinct_trans, "\n")

## Number of distinct transmission: 4

distinct_transmission <- unique(CarsData$transmission)
print(distinct_transmission)

## [1] "Manual"    "Semi-Auto" "Automatic" "Other"
```

Get the distinct values for the Fuel Type column

```
num_distinct_fuelType <- CarsData %>%
  distinct(fuelType) %>%
  nrow()
cat("Number of distinct fuelType:", num_distinct_fuelType, "\n")

## Number of distinct fuelType: 5

distinct_fuelType <- unique(CarsData$fuelType)
print(distinct_fuelType)

## [1] "Petrol"   "Diesel"   "Hybrid"   "Other"    "Electric"
```

Get the distinct values for the manufacturer column

```
num_distinct_manuf <- CarsData %>%
  distinct(Manufacturer) %>%
  nrow()
cat("Number of distinct Manufacturer:", num_distinct_manuf, "\n")

## Number of distinct Manufacturer: 9

distinct_manufacturer <- unique(CarsData$Manufacturer)
print(distinct_manufacturer)

## [1] "hyundi"     "volkswagen" "BMW"        "skoda"      "ford"
## [6] "toyota"     "merc"       "vauxhall"   "Audi"
```

Statistical Summary

```
summary(CarsData)

##     model                year          price         transmission
##  Length:97712       Min.   :1970   Min.   :   450   Length:97712
##  Class :character   1st Qu.:2016   1st Qu.:  9999   Class :character
##  Mode  :character   Median :2017   Median : 14470   Mode  :character
##                     Mean   :2017   Mean   : 16773
##                     3rd Qu.:2019   3rd Qu.: 20750
##                     Max.   :2024   Max.   :159999
##     mileage          fuelType              tax              mpg
##  Min.   :      1   Length:97712       Min.   :  0.0   Min.   :  0.30
##  1st Qu.:   7673   Class :character   1st Qu.:125.0   1st Qu.: 47.10
##  Median : 17683   Mode  :character   Median :145.0   Median : 54.30
##  Mean   : 23220                      Mean   :120.1   Mean   : 55.21
##  3rd Qu.: 32500                      3rd Qu.:145.0   3rd Qu.: 62.80
##  Max.   :323000                      Max.   :580.0   Max.   :470.80
##    engineSize     Manufacturer
##  Min.   :0.000   Length:97712
##  1st Qu.:1.200   Class :character
##  Median :1.600   Mode  :character
##  Mean   :1.665
```

```
##   3rd Qu.:2.000
##   Max.   :6.600
```

ANALYZE PHASE

This section shows the descriptive analysis.

Average Prices by Transmission Type

```
prices_by_transmission <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  group_by(transmission) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  arrange(desc(avg_price))

print(prices_by_transmission)

## # A tibble: 4 × 2
##   transmission avg_price
##   <chr>            <dbl>
## 1 Semi-Auto       24252.
## 2 Automatic       21749.
## 3 Other           16219.
## 4 Manual          12181.
```

Average MPG by Fuel Type

```
mpg_by_fuel <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  group_by(fuelType) %>%
  summarize(avg_mpg = mean(mpg, na.rm = TRUE)) %>%
  arrange(desc(avg_mpg))

print(mpg_by_fuel)

## # A tibble: 5 × 2
##   fuelType avg_mpg
##   <chr>      <dbl>
## 1 Electric   297.
## 2 Hybrid      89.0
## 3 Other       85.9
## 4 Diesel      58.3
## 5 Petrol      51.0
```

Average Mileage by Manufacturer

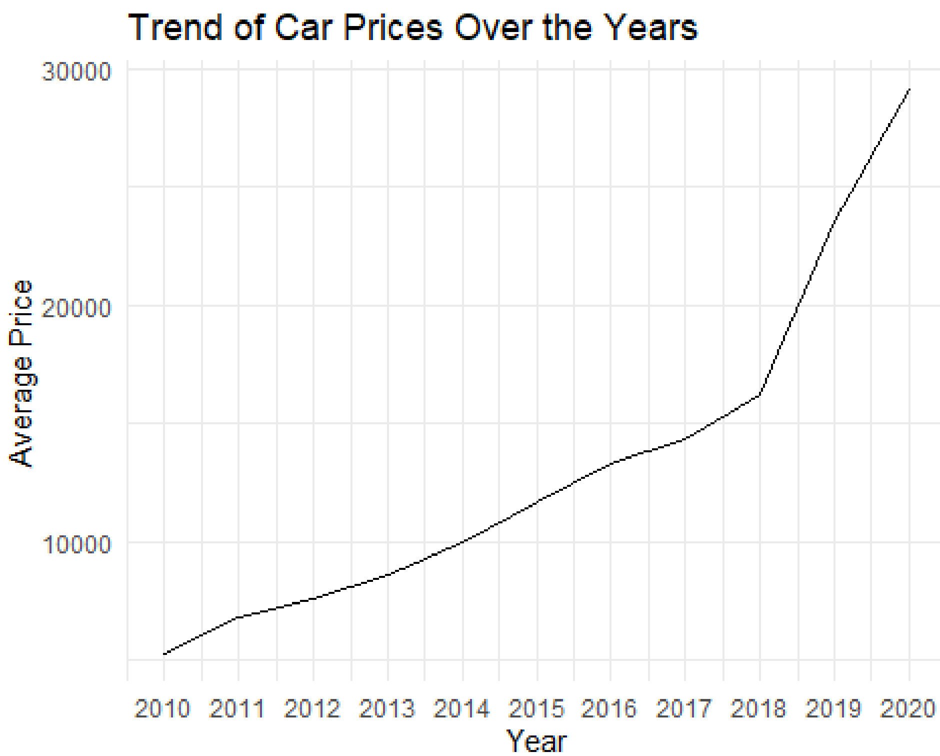
```
mileage_by_manufacturer <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  group_by(Manufacturer) %>%
  summarize(avg_mileage = mean(mileage, na.rm = TRUE)) %>%
  arrange(desc(avg_mileage))

print(mileage_by_manufacturer)
```

```
## # A tibble: 9 × 2
##   Manufacturer avg_mileage
##   <chr>              <dbl>
## 1 BMW               25004.
## 2 Audi              24318.
## 3 vauxhall          23370.
## 4 ford              22664.
## 5 toyota            21874.
## 6 volkswagen        21711.
## 7 merc              21602.
## 8 hyundi            21287.
## 9 skoda             19794.
```

SHARE PHASE

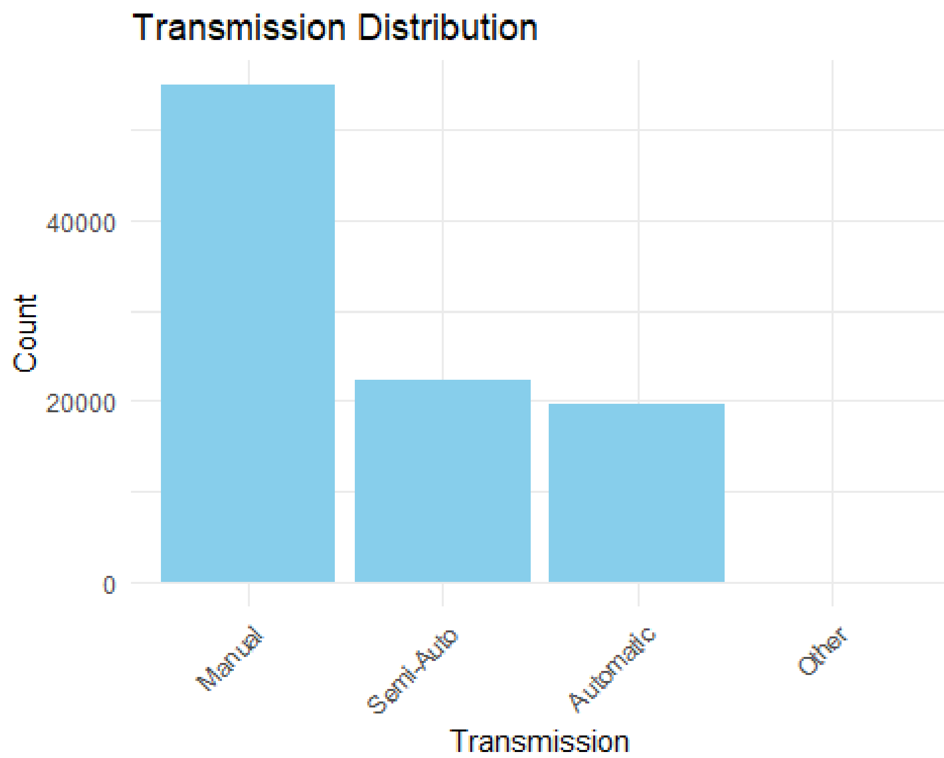
Trend of Average Car Prices Over the Years (2010-2020)

```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020)

# Calculate average price by year
price_trend <- filtered_data %>%
  group_by(year) %>%
  summarize(avg_price = mean(price, na.rm = TRUE))

ggplot(price_trend, aes(x = year, y = avg_price)) +
  geom_line() +
  labs(title = "Trend of Car Prices Over the Years",
       x = "Year",
       y = "Average Price") +
  scale_x_continuous(breaks = seq(2010, 2020, 1), labels = seq(2010, 2020,
1)) +
  theme_minimal()
```



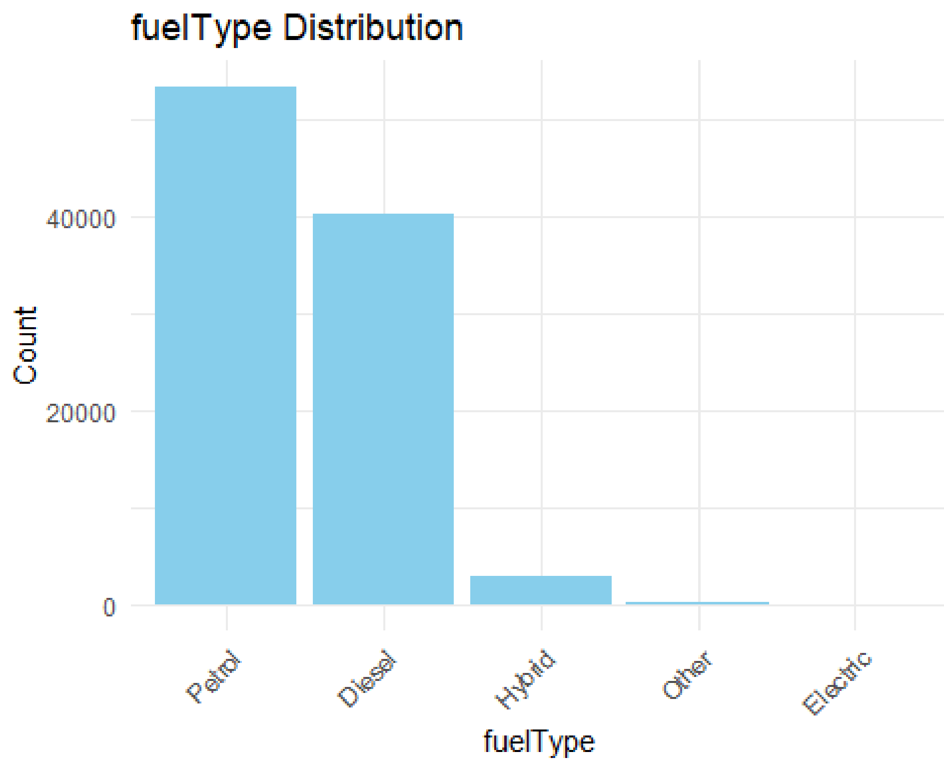
Transmission Distribution (2010-2020)

```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020)

# Summarize the counts of each transmission type within the filtered dataset
transmission_counts <- filtered_data %>%
  group_by(transmission) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

# Reorder the levels of the transmission factor variable based on the count
transmission_counts$transmission <- factor(transmission_counts$transmission,
levels = transmission_counts$transmission)

ggplot(transmission_counts, aes(x = transmission, y = count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Transmission Distribution",
       x = "Transmission",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis
labels for better readability
```



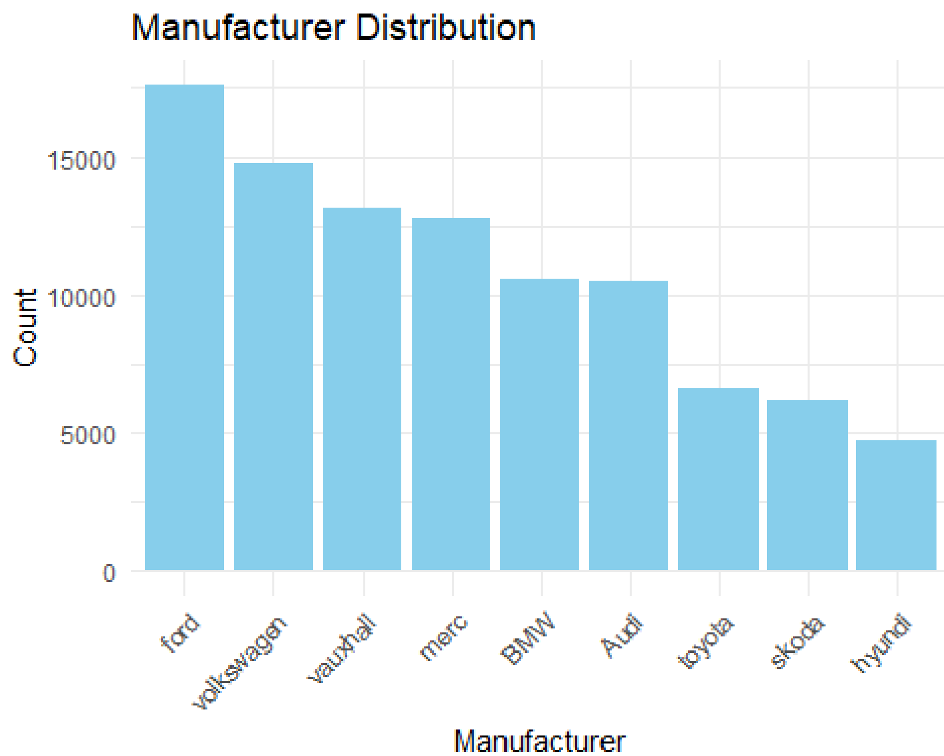
Fuel Type Distribution

```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020)

fuelType_counts <- filtered_data %>%
  group_by(fuelType) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

fuelType_counts$fuelType <- factor(fuelType_counts$fuelType, levels =
fuelType_counts$fuelType)

ggplot(fuelType_counts, aes(x = fuelType, y = count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "fuelType Distribution",
       x = "fuelType",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis
labels for better readability
```



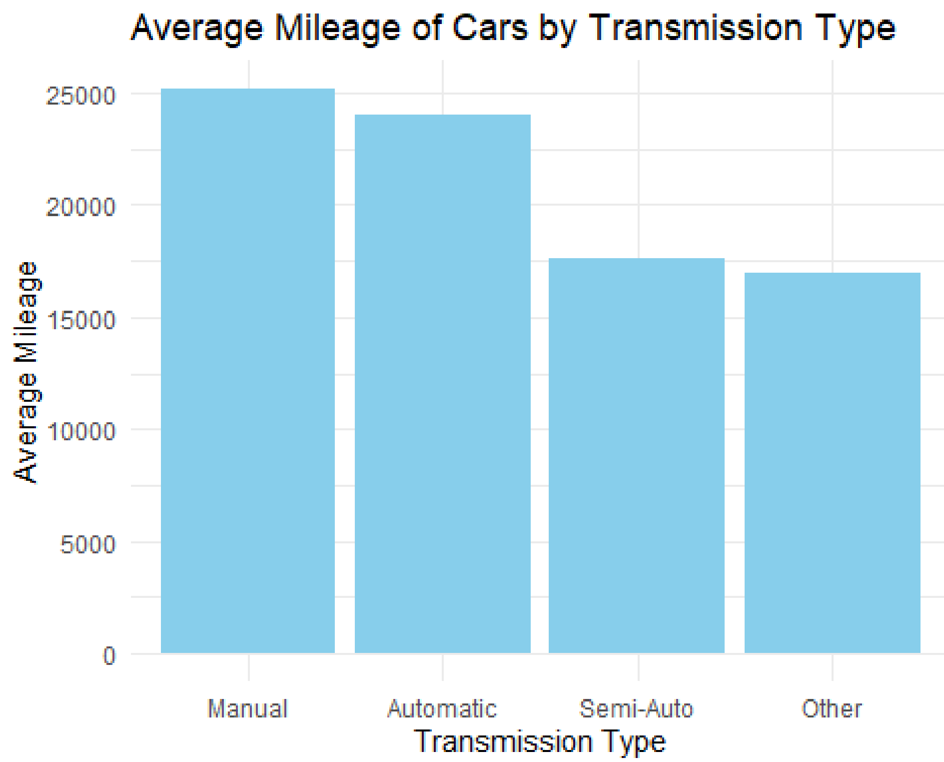
Manufacturer Distribution

```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020)

# Summarize the counts of each Manufacturer type within the filtered dataset
Manufacturer_counts <- filtered_data %>%
  group_by(Manufacturer) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

Manufacturer_counts$Manufacturer <- factor(Manufacturer_counts$Manufacturer,
levels = Manufacturer_counts$Manufacturer)

ggplot(Manufacturer_counts, aes(x = Manufacturer, y = count)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Manufacturer Distribution",
       x = "Manufacturer",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis
labels for better readability
```



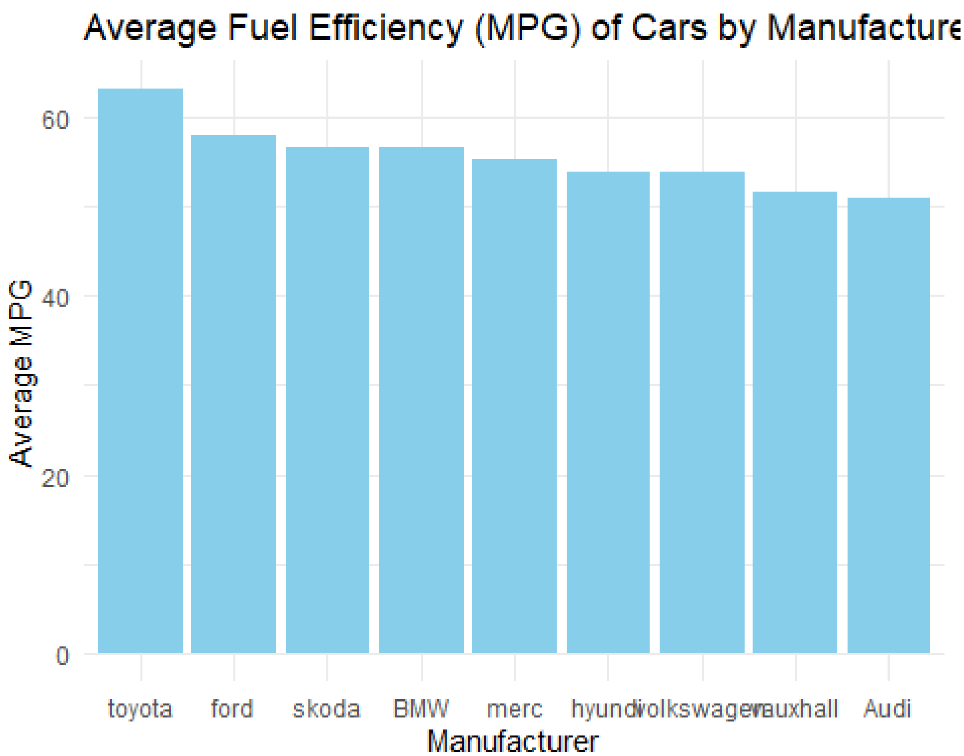
Average Mileage of Cars by Transmission Type

```
mileage_comparison <- CarsData %>%
  group_by(transmission) %>%
  summarize(avg_mileage = mean(mileage, na.rm = TRUE)) %>%
  arrange(desc(avg_mileage))

ggplot(mileage_comparison, aes(x = reorder(transmission, -avg_mileage), y =
avg_mileage)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Mileage of Cars by Transmission Type",
       x = "Transmission Type",
       y = "Average Mileage") +
  theme_minimal()
```



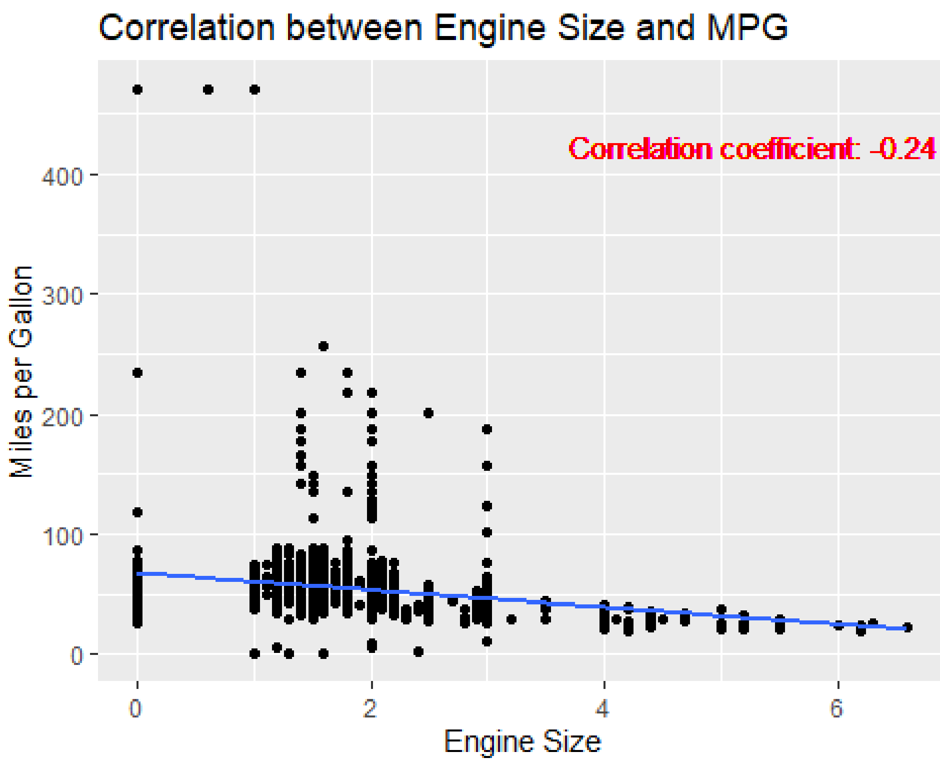
Average Fuel Efficiency of Cars by Manufacturer

```
mpg_comparison <- CarsData %>%
  group_by(Manufacturer) %>%
  summarize(avg_mpg = mean(mpg, na.rm = TRUE)) %>%
  arrange(desc(avg_mpg)) # Arrange the data in descending order based on
average mpg

ggplot(mpg_comparison, aes(x = reorder(Manufacturer, -avg_mpg), y = avg_mpg))
+
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Fuel Efficiency (MPG) of Cars by Manufacturer",
       x = "Manufacturer",
       y = "Average MPG") +
  theme_minimal()
```



Correlation between Engine Size and MPG

```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020)

engine_mpg_data <- filtered_data %>%
  select(engineSize, mpg)

correlation_coefficient <- cor(engine_mpg_data$engineSize,
engine_mpg_data$mpg)

ggplot(engine_mpg_data, aes(x = engineSize, y = mpg)) +
  geom_point() +
  labs(title = "Correlation between Engine Size and MPG",
       x = "Engine Size",
       y = "Miles per Gallon") +
  geom_smooth(method = "lm") +
  geom_text(aes(label = paste("Correlation coefficient:",
round(correlation_coefficient, 2))),
            x = max(engine_mpg_data$engineSize) * 0.8,
            y = max(engine_mpg_data$mpg) * 0.9,
            size = 4,
            color = "red")

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

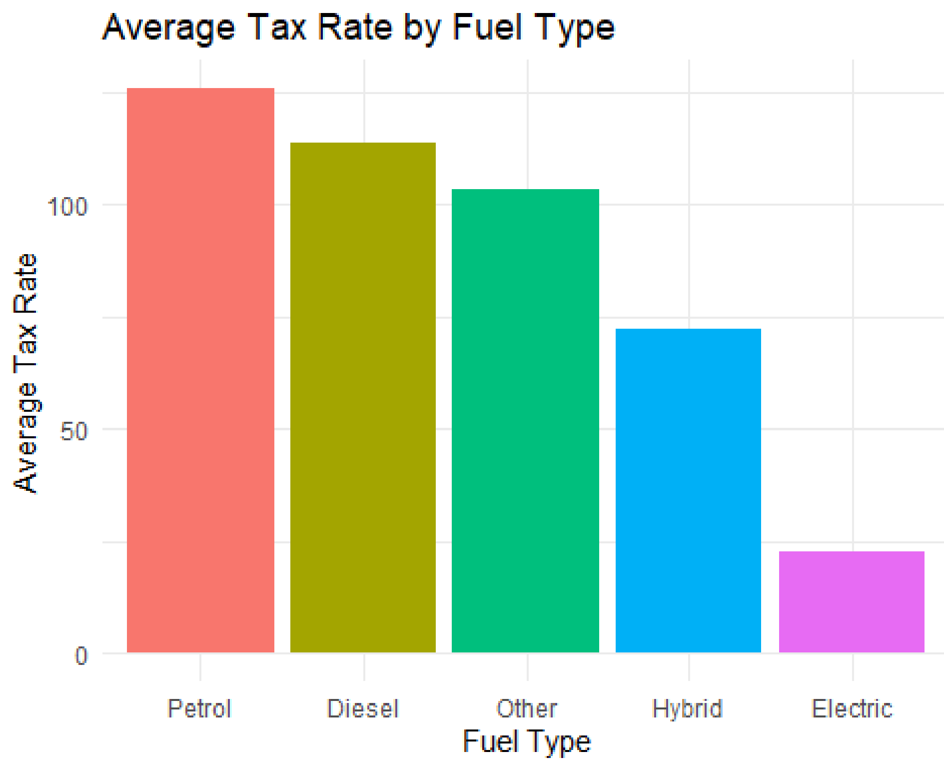


Interpretation: A correlation coefficient -0.24 indicates a weak negative correlation between engine size and miles per galon. A negative correlation between engine size and mpg suggests that larger engines tend to have lower fuel efficiency, resulting in fewer miles per gallon. However, the correlation is weak, so other factors may have a stronger influence on mpg, such as vehicle weight, driving habits, or engine technology.

Average Tax Rate by Fuel Type

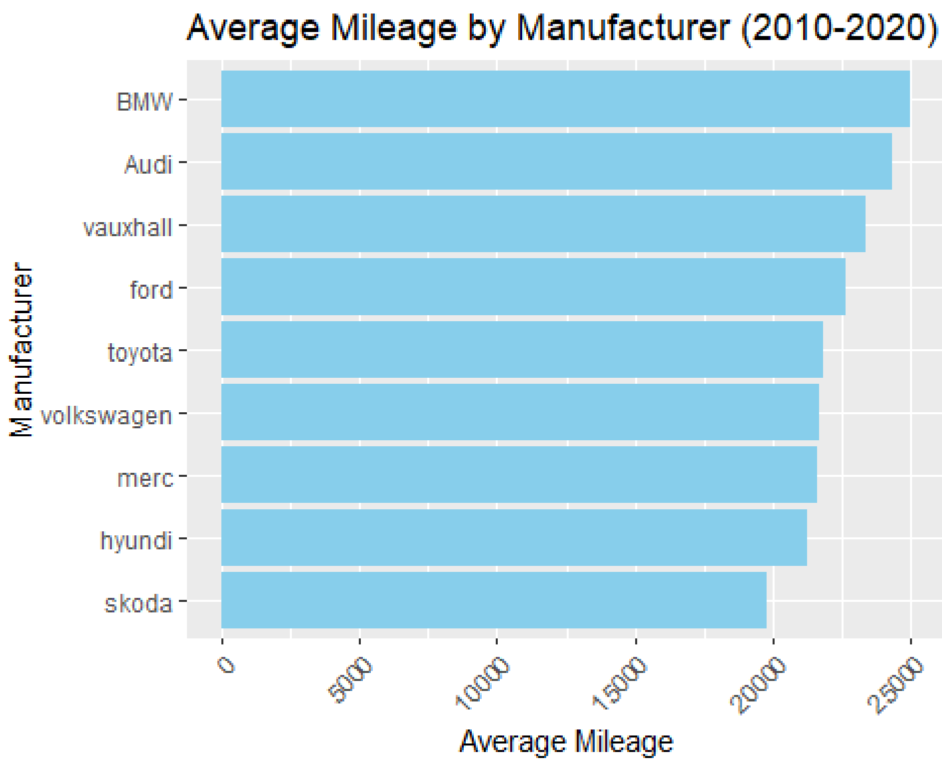
```
filtered_data <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  select(tax, fuelType)

# Convert fuel type to a factor and reorder the levels based on average tax
rate
filtered_data$fuelType <- factor(filtered_data$fuelType, levels =
unique(filtered_data$fuelType[order(filtered_data$tax, decreasing = TRUE)]))

tax_by_fuel <- filtered_data %>%
  group_by(fuelType) %>%
  summarize(avg_tax = mean(tax, na.rm = TRUE))

ggplot(tax_by_fuel, aes(x = fuelType, y = avg_tax, fill = fuelType)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Tax Rate by Fuel Type",
       x = "Fuel Type",
       y = "Average Tax Rate") +
```

```
  theme_minimal() +
  theme(legend.position = "none")
```



Average Mileage by Manufacturer

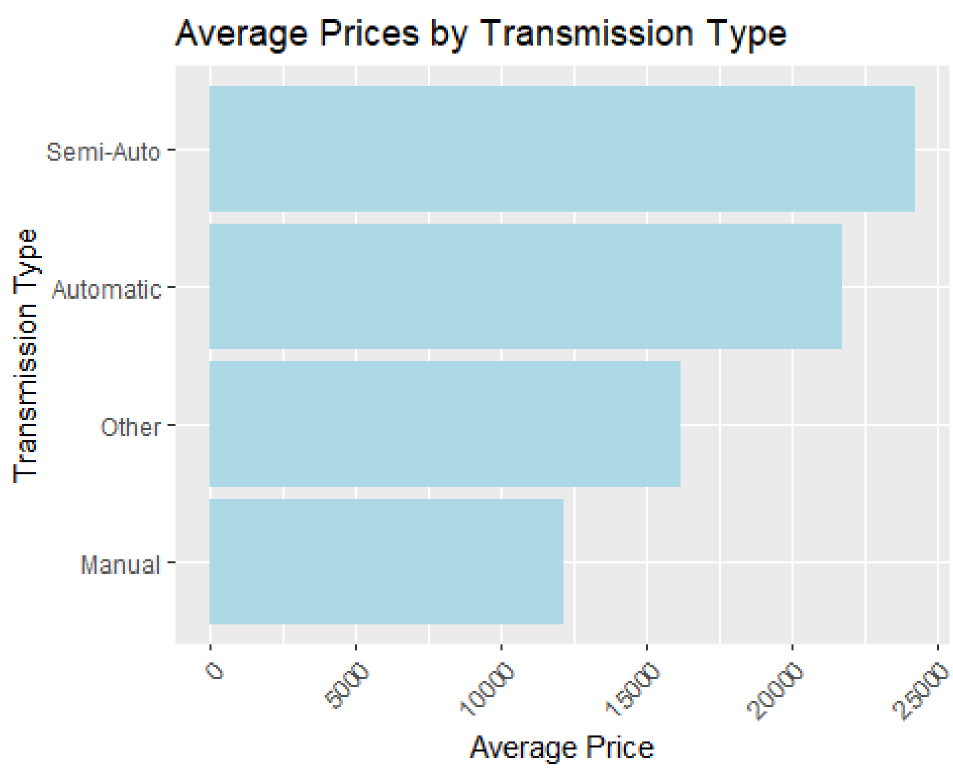
```
mileage_by_manufacturer <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  group_by(Manufacturer) %>%
  summarize(avg_mileage = mean(mileage, na.rm = TRUE)) %>%
  arrange(desc(avg_mileage))

# Create a bar plot
ggplot(mileage_by_manufacturer, aes(x = reorder(Manufacturer, avg_mileage), y
= avg_mileage)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Mileage by Manufacturer (2010-2020)",
       x = "Manufacturer",
       y = "Average Mileage") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()  # Rotate the x-axis labels for better readability
```



Average Mileage by Manufacturer (2010-2020)
BMW
Audi
vauxhall
ford
toyota
volkswagen
merc
hyundi
skoda
Manufacturer
0
5000
10000
15000
20000
25000
Average Mileage

Average Prices by Transmission Type

```r
prices_by_transmission <- CarsData %>%
  filter(year >= 2010 & year <= 2020) %>%
  group_by(transmission) %>%
  summarize(avg_price = mean(price, na.rm = TRUE)) %>%
  arrange(desc(avg_price))

ggplot(prices_by_transmission, aes(x = reorder(transmission, avg_price), y =
avg_price)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Average Prices by Transmission Type",
       x = "Transmission Type",
       y = "Average Price") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()  # Rotate the x-axis labels for better readability
```



ACT

Conclusion

1. Price Dynamics: The analysis revealed a notable upward trend in car prices over the decade, indicating market inflation or increased demand for automobiles during the period.
2. Market Dominance: Ford's dominance as the manufacturer with the highest number of car sales was cemented throughout time, highlighting the power of its brand and its attractiveness to consumers.
3. Transmission Type Trends: Manual transmission vehicles remained prevalent in the market, potentially due to factors such as cost, driving experience, or market demand.
4. Transmission Analysis: Automobiles with manual transmissions generally performed better in terms of mileage, suggesting that this type of transmission may be preferred in some markets or driving situations.
5. Transmission Pricing: Semi-Automatic transmission vehicles commanded the highest average prices.
6. Fuel Type Preference: Petrol-powered cars dominated the market share, indicating consumer preferences during the analyzed period.
7. Fuel Efficiency Insights: Bigger engine sizes showed a negative correlation with fuel efficiency, evidenced by lower miles per gallon (mpg). This implies that vehicles with bigger engines tend to consume more fuel per mile traveled.
8. Manufacturer Mileage: Toyota has emerged as a pioneer in fuel efficiency, reporting the highest average mpg among industry manufacturers, a reflection of its emphasis on designing vehicles with minimal environmental impact.
9. Tax Trends: Petrol-powered vehicles has the highest tax rates which is indicative of the government's policies and the dynamics of the industry of automobiles.