Understanding the Relationship between Factors and CO2 Emissions in Fuel Vehicles: Analysis and Modeling

Zhiqi Pang, Xintong Yu, Laura Assylgazhina, Cindy Dai

2023-06-16

Contents

1 Introduction

The issue of carbon dioxide (CO2) emissions and their impact on climate change has become a pressing concern worldwide. In the transportation sector, fuel vehicles are significant contributors to CO2 emissions. Understanding the factors that influence CO2 emissions from fuel vehicles is crucial for promoting sustainable practices and reducing our environmental footprint.

Our data science project focuses on analyzing the factors associated with CO2 emissions from fuel vehicles. By examining variables such as vehicle year, make, engine size, cylinders, weight, transmission type, and fuel type, we aim to gain insights into the relationship between these factors and CO2 emissions. This analysis can contribute to a better understanding of the factors driving CO2 emissions in the transportation sector.

While the direct policy implications of our project may vary, the findings can still provide valuable insights for various stakeholders. Automotive manufacturers can leverage this knowledge to develop more environmentally friendly vehicles. Consumers can make informed decisions when purchasing vehicles, considering factors that influence CO2 emissions. Additionally, our project adds to the collective understanding of the environmental impact of fuel vehicles and contributes to the broader efforts toward sustainability.

By delving into the factors that contribute to CO2 emissions from fuel vehicles, our project aims to contribute to the ongoing discussions surrounding climate change mitigation and the transition to more sustainable transportation practices.

1.1 Purpose

The purpose of our project is to analyze and understand the relationships between various factors and carbon dioxide (CO2) emissions from fuel vehicles. By examining these relationships, we aim to build and test several models to determine their effectiveness in predicting CO2 emissions. Through our analysis, we seek to gain insights into the factors that significantly contribute to CO2 emissions and identify the most suitable model for accurate predictions. The findings from our project will provide valuable knowledge and contribute to the understanding of the factors influencing CO2 emissions from fuel vehicles.

1.2 The Dataset

Dataset: The dataset used in this project was derived from the Open Data from the Government of Canada (2023) [1]. It consists of information about vehicles' characteristics and their corresponding CO2 emission ratings. The data is collected from eight CSV files spanning from 2016 to 2023, with each file containing data for vehicles of a specific year.

Dataset Description: The resulting dataset comprises 8,046 rows and 14 columns. It includes the following variables:

1. Year_of_model: The model year of the vehicles ranging from 2016 to 2023.
2. Make: The brand of the vehicle.
3. Model: The model of the vehicle.
4. Vehicle_Class: The class/category of the vehicle.
5. Engine_Size: The total displacement of all cylinders in liters.
6. Cylinders: The number of engine cylinders.
7. Transmission: The type of transmission.
8. Fuel_name: The fuel type of the vehicle (E = E85, X = Regular gasoline, D = Diesel).
9. CO2_Emissions: The vehicle's tailpipe emissions of carbon dioxide in grams per kilometer (g/km) for combined city and highway driving.
10. CO2_rating: The vehicle's tailpipe emissions of carbon dioxide rated on a scale from 1 (worst) to 10 (best).
11. Weight_upto_kg: Maximum weight of the vehicle (in kilograms) referring to its Vehicle Class.
12. Auto_trans: A binary variable indicating automatic transmission (1) or manual transmission (0).
13. CO2_classes: A new variable created based on CO2_rating, categorizing vehicles into "Low emissions" (CO2 emissions <= 249 g/km) and "High emissions" (CO2 emissions > 249 g/km).
14. Fuel: Fuel types dummified into 1 for E85, 2 for Regular gasoline, and 3 for Diesel.

Data Source: The data used in this project is sourced from Open Data from the Government of Canada [1]. It is licensed under the Open Government Licence – Canada.

The dataset provides comprehensive information about various vehicle characteristics and their corresponding CO2 emissions. By exploring the relationships between these factors and building predictive models, we aim to gain insights into the factors influencing CO2 emissions and identify the most significant variables for predicting emission levels.

#1.3 Methodology

This project aimed to identify the variables influencing CO2 emissions. The analysis involved data collection, preprocessing, exploratory data analysis, model selection, model training and evaluation, contingency table analysis, assumption checking, interpretation, and reporting.

1. Preprocessing: The data underwent preprocessing to handle missing values, formatting inconsistencies, and modifying variables.

2. Exploratory Data Analysis: Descriptive statistics and data visualization were used to explore variable distributions and relationships.
3. Model Selection: Logistic regression, LDA, QDA, classification decision tree, and multinomial regression were chosen as relevant models for the analysis.
4. Model Training and Evaluation: Models were trained using a training dataset and evaluated using performance metrics and cross-validation techniques.
5. Contingency Tables and Tests: Contingency tables and chi-square tests were used to examine the relationship between CO2 emissions and categorical variables (e.g., vehicle make, vehicle class).
6. Assumptions Checking: Assumptions underlying the models, such as normality and homoscedasticity, were assessed using diagnostic tests.
7. Interpretation and Reporting: Significant variables contributing to CO2 emissions were identified, and findings were reported with relevant statistical measures.
8. Conclusion: Conclusions were drawn regarding the significant variables influencing CO2 emissions based on the analysis of selected models, contingency tables, and statistical tests.

1.4 Preprocessing

1. All 8 datasets in CSV formats were read to dataframes
2. Dataframes were merged by merged_data<-rbind(data1, data2, data3, data4, data5, data6, data7, data8)
3. Some of columns were renamed for convenience
4. Missing values were checked by sum(is.na(merged_data)): NA values were not identified
5. The values in Vehicle Class and Make were modified to one format to prevent duplicates (such as "LAMBORGHINI" and "Lamborghini"), for examle: merged_data[merged_data == "SUV - SMALL"] <- "SUV: Small" merged_data[merged_data == "TWO-SEATER"] <- "Two-seater" merged_data$Make = toupper(merged_data$Make)
6. Creating new variables to use in models:

merged_data$CO2_classes <- -ifelse(merged_data$CO2_Emissions > 249, "High", "Low")

merged_data$Auto_trans <- -ifelse(merged_data$Transmission %in% c("M6", "M7", "M5"), 0, 1) "M6", "M7", "M5" - are fully manual transmisison types, the rest - are Automatic or Semi-automatic transmission types

merged_data$Fuel[merged_data$Fuel_name == "E"] <- 1 merged_data$Fuel[merged_data$Fuel_name == "X"] <- 2 merged_data$Fuel[merged_data$Fuel_name == "D"] <- 3 Fuel types dummified into 1 for E85, 2 for Regular gasoline, and 3 for Diesel (from less associated with CO2 to more associated with CO2 regarding fuel composition)

7. Converting categorical variables to factors:

```
#read the data

mydata = read.csv("merged_data_full_6.csv")
```

```
#convert the categorical variables from character type to factor

mydata$CO2_classes = as.factor(mydata$CO2_classes)
mydata$Fuel = as.factor(mydata$Fuel)
```

```
mydata$Make = as.factor(mydata$Make )
mydata$Vehicle_Class = as.factor(mydata$Vehicle_Class)
mydata$Transmission = as.factor(mydata$Transmission)
mydata$Fuel_name = as.factor(mydata$Fuel_name)
```

1.5 Exploratory Data Analysis

In this part of our analysis, we will focus on summary information about the dataset and examine the variables individually through visualizations and contingency tables.

The tabular form of our data is shown below:

```
#display the data
head(mydata, n=4)
```

```
##   Year_of_model  Make          Model Vehicle_Class Engine_Size Cylinders
## 1          2023 ACURA        Integra     Full-size         1.5         4
## 2          2023 ACURA Integra A-SPEC     Full-size         1.5         4
## 3          2023 ACURA Integra A-SPEC     Full-size         1.5         4
## 4          2023 ACURA     MDX SH-AWD    SUV: Small         3.5         6
##   Transmission Fuel_name CO2_Emissions CO2_rating Weight_upto_kg Auto_trans
## 1          AV7         X           167          6           2700          1
## 2          AV7         X           172          6           2700          1
## 3           M6         X           181          6           2700          0
## 4         AS10         X           263          4           2041          1
##   CO2_classes Fuel
## 1         Low    2
## 2         Low    2
## 3         Low    2
## 4        High    2
```

The dataset has 8046 rows and 14 columns:

```
#check the dimension of the dataset
str(mydata)
```

```
## 'data.frame':    8046 obs. of  14 variables:
##  $ Year_of_model : int  2023 2023 2023 2023 2023 2023 2023 2023 2023 2023 ...
##  $ Make          : Factor w/ 41 levels "ACURA","ALFA ROMEO",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Model         : chr  "Integra" "Integra A-SPEC" "Integra A-SPEC" "MDX SH-AWD" ...
##  $ Vehicle_Class : Factor w/ 15 levels "Compact","Full-size",..: 2 2 2 12 13 12 12 1 1 1 ...
##  $ Engine_Size   : num  1.5 1.5 1.5 3.5 3 2 2 2 2 3 ...
##  $ Cylinders     : int  4 4 4 6 6 4 4 4 4 6 ...
##  $ Transmission  : Factor w/ 26 levels "A10","A4","A5",..: 22 22 25 12 12 12 12 12 12 12 ...
##  $ Fuel_name     : Factor w/ 3 levels "D","E","X": 3 3 3 3 3 3 3 3 3 3 ...
##  $ CO2_Emissions : int  167 172 181 263 291 232 242 230 231 256 ...
##  $ CO2_rating    : int  6 6 6 4 4 5 5 5 5 5 ...
##  $ Weight_upto_kg: int  2700 2700 2700 2041 2722 2041 2041 1600 1600 1600 ...
##  $ Auto_trans    : int  1 1 0 1 1 1 1 1 1 1 ...
##  $ CO2_classes   : Factor w/ 2 levels "High","Low": 2 2 2 1 1 2 2 2 2 1 ...
##  $ Fuel          : Factor w/ 3 levels "1","2","3": 2 2 2 2 2 2 2 2 2 2 ...
```

```r
dim(mydata)
```

```
## [1] 8046   14
```

The summary function provides a concise summary of the key statistics and information about the variables in our dataset. It includes the count, mean, minimum, maximum, and quartile values for numeric variables, as well as the frequency count for categorical variables:

```r
#summary of data

summary(mydata)
```

```
##  Year_of_model             Make         Model
##  Min.   :2016   FORD         : 734   Length:8046
##  1st Qu.:2017   CHEVROLET    : 636   Class :character
##  Median :2019   BMW          : 545   Mode  :character
##  Mean   :2019   MERCEDES-BENZ: 507
##  3rd Qu.:2021   PORSCHE      : 444
##  Max.   :2023   GMC          : 386
##                 (Other)      :4794
##                 Vehicle_Class   Engine_Size      Cylinders      Transmission
##  SUV: Small            :1499   Min.   :0.900   Min.   : 3.00   AS8    :1637
##  Mid-size              :1125   1st Qu.:2.000   1st Qu.: 4.00   AS6    : 985
##  SUV: Standard         : 976   Median :3.000   Median : 6.00   M6     : 809
##  Compact               : 891   Mean   :3.143   Mean   : 5.62   A8     : 655
##  Pickup truck: Standard: 749   3rd Qu.:3.700   3rd Qu.: 6.00   AM7    : 558
##  Subcompact            : 695   Max.   :8.400   Max.   :16.00   A9     : 557
##  (Other)               :2111                                   (Other):2845
##  Fuel_name CO2_Emissions     CO2_rating     Weight_upto_kg   Auto_trans
##  D: 192    Min.   : 94.0   Min.   : 1.000   Min.   :1300   Min.   :0.0000
##  E: 257    1st Qu.:211.0   1st Qu.: 4.000   1st Qu.:1600   1st Qu.:1.0000
##  X:7597    Median :249.0   Median : 5.000   Median :2041   Median :1.0000
##            Mean   :253.6   Mean   : 4.643   Mean   :2147   Mean   :0.8726
##            3rd Qu.:292.0   3rd Qu.: 6.000   3rd Qu.:2700   3rd Qu.:1.0000
##            Max.   :608.0   Max.   :10.000   Max.   :3628   Max.   :1.0000
##
##  CO2_classes Fuel
##  High:4001   1: 257
##  Low :4045   2:7597
##              3: 192
##
##
##
##
```

```r
colSums(is.na(mydata))
```

```
##  Year_of_model           Make          Model  Vehicle_Class    Engine_Size
##              0              0              0              0              0
##      Cylinders   Transmission      Fuel_name  CO2_Emissions     CO2_rating
##              0              0              0              0              0
## Weight_upto_kg     Auto_trans    CO2_classes           Fuel
##              0              0              0              0
```

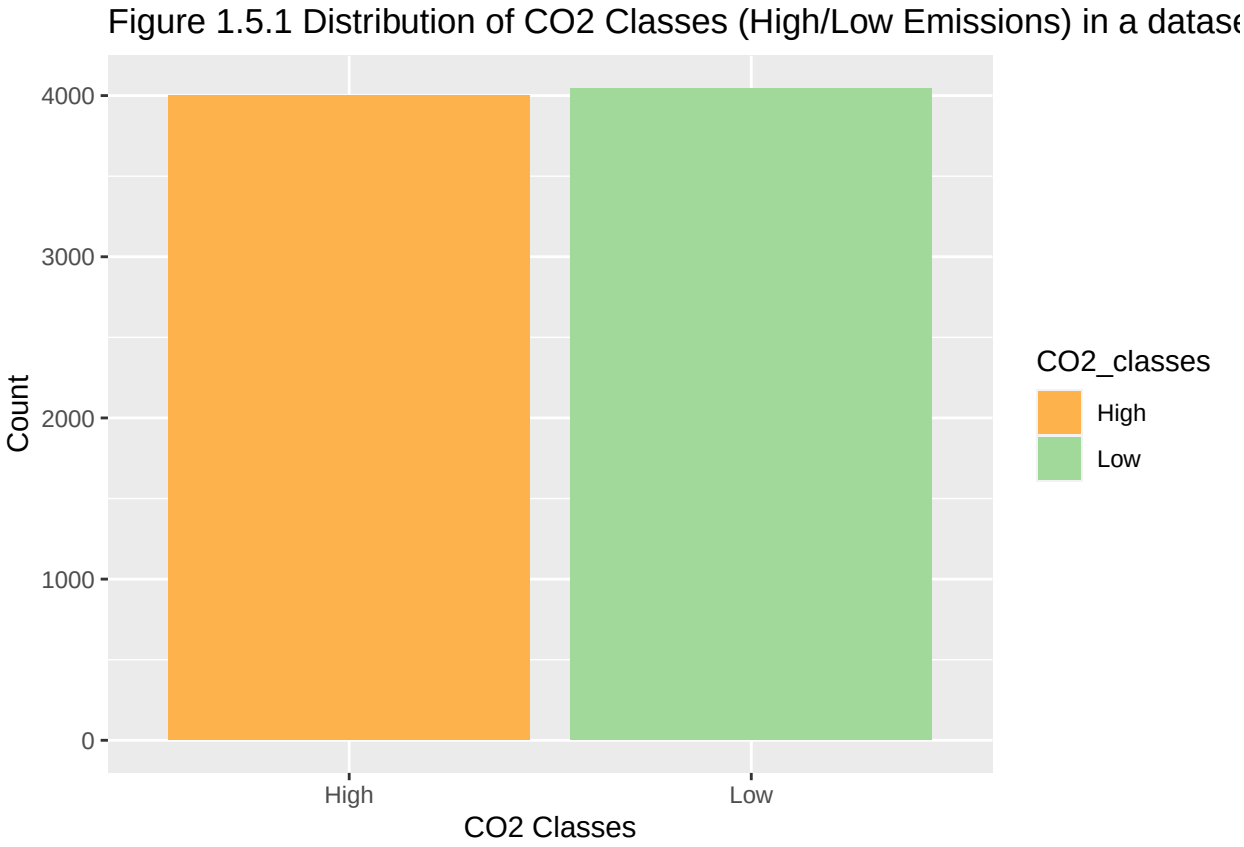
Overall distribution of classes

Based on the dataset, we observed that 49.7% (4001 vehicles) of the vehicles belong to the high emissions class, while 50.3% (1016 vehicles) belong to the low emissions class. This is due to the CO2 Class is classified based on the CO2 emission and split at median value of CO2 emission data. Therefore, the two CO2 Class are close to 50/50 distributed in the Dataset. No additional re-sampling is needed to avoid bias. This distribution is visually represented in Figure 1.5.1 below.

```
#Check the classes and their sizes

table(mydata$CO2_classes)
```

```
##
## High  Low
## 4001 4045
```

```
library(ggplot2)

# Define the colors for the CO2 classes
my_colors <- c("#feb24c", "#a1d99b")

# Plot the overall distribution of CO2 classes
ggplot(mydata, aes(x = CO2_classes, fill = CO2_classes)) +
  geom_bar() +
  scale_fill_manual(values = my_colors) +
  xlab("CO2 Classes") +
  ylab("Count") +
  ggtitle("Figure 1.5.1 Distribution of CO2 Classes (High/Low Emissions) in a dataset")
```



CO2 Class (High/Low CO2 Emissions) and Year of Model

For the High CO2 rating class, the data is distributed across the years 2016 to 2023. However, it's important to note that the data for 2023 is not complete, and the counts may not represent the full year. Excluding 2023, the highest number of vehicles in this class was manufactured in 2022, with a count of 535. This suggests that 2022 had a relatively larger number of vehicles with low CO2 emissions (high CO2 rating). On the other hand, the year 2017 had the lowest count with 479 vehicles, indicating a comparatively higher production of high CO2 emission vehicles during that year.

For the Low CO2 rating class, the year 2016 had the highest count in this class, with 589 vehicles. Since then, the number of low CO2 ratings has decreased year by year, implying a gradual reduction in the CO2 emissions of Canadian vehicles. Although the specific counts for 2023 are not available, the decreasing trend in the number of low CO2 emission vehicles suggests a positive shift towards lower emissions in more recent years.

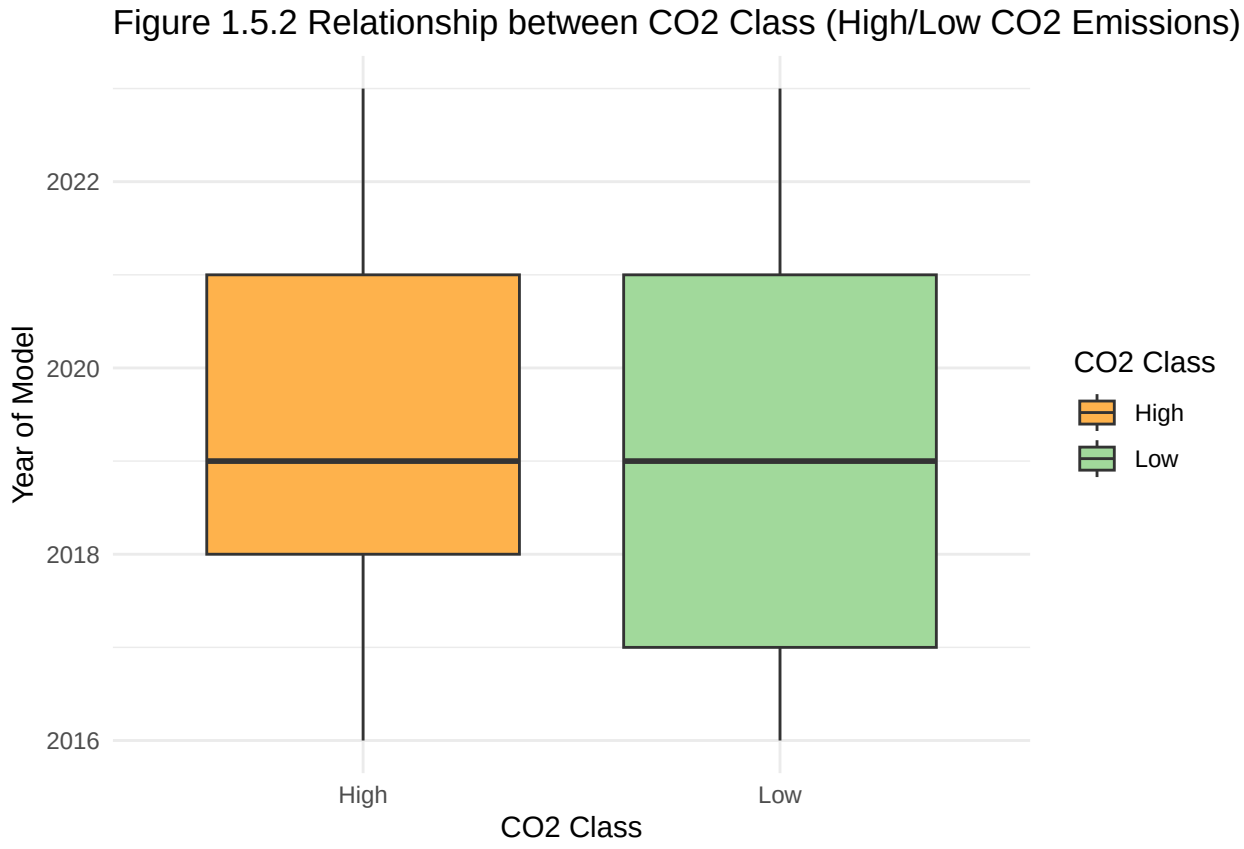
```
# Distribution of data between CO2_Class=High (High CO2 Emissions) and Year of Model
table(high$Year_of_model)
```

```
##
## 2016 2017 2018 2019 2020 2021 2022 2023
##  520  479  505  513  501  519  535  429
```

```
# Distribution of data between CO2_Class=Low (Low CO2 Emissions) and Year of Model
table(low$Year_of_model)
```

```
##
## 2016 2017 2018 2019 2020 2021 2022 2023
##  589  579  578  543  474  451  439  392
```

```r
# Custom color palette
my_colors <- c("#feb24c", "#a1d99b")

# Create the plot
ggplot(mydata, aes(x = CO2_classes, y = Year_of_model, fill = CO2_classes)) +
  geom_boxplot() +
  scale_fill_manual(values = my_colors) +
  theme_minimal() +
  labs(x = "CO2 Class", y = "Year of Model", fill = "CO2 Class") +
  ggtitle("Figure 1.5.2 Relationship between CO2 Class (High/Low CO2 Emissions) and Year of Model")
```



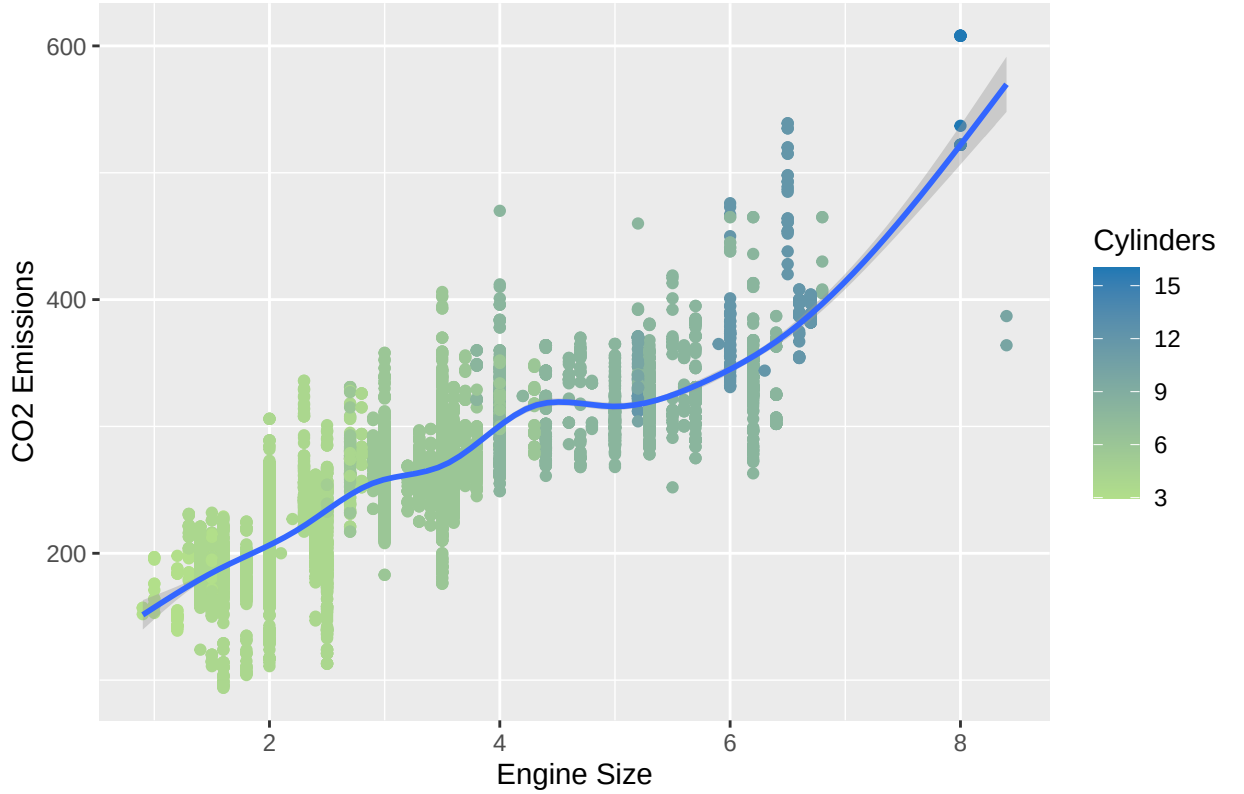
Visualizing the impact of engine size and cylinder count on CO2 emission through a scatter plot

```r
# Making plot of engine size and CO2 emissions incorporating cylinders as categorical variable
ggplot(data = mydata, aes(x = Engine_Size, y = CO2_Emissions, colour = Cylinders)) +
  geom_point() +
  geom_smooth() +
  scale_color_gradient(low = "#B2DF8A", high = "#1F78B4") +
  labs(title = "Figure 1.5.3 Impact of Engine Size and Cylinder Count on CO2 Emissions",
```

```
       x = "Engine Size",
       y = "CO2 Emissions")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

## Warning: The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

Figure 1.5.3 Impact of Engine Size and Cylinder Count on CO2 Emissions



The scatter plot demonstrates a positive correlation between engine size and CO2 emissions, indicating that larger engine sizes typically result in higher CO2 emissions. Additionally, the plot emphasizes the positive association between the number of cylinders in a vehicle and CO2 emissions. This connection is based on the principle that a higher cylinder count often corresponds to a larger engine size, as more cylinders require additional combustion space. It is worth noting that the majority of vehicles with engine sizes above 5 tend to have cylinder counts exceeding 12, potentially leading to increased CO2 emissions.

CO2 Class (High/Low CO2 Emissions) and Vehicle Class Weight

```
summary(high$Weight_upto_kg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1300    1800    2700    2419    2722    3628
```

```r
summary(low$Weight_upto_kg)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1300    1600    1800    1879    2041    3628
```

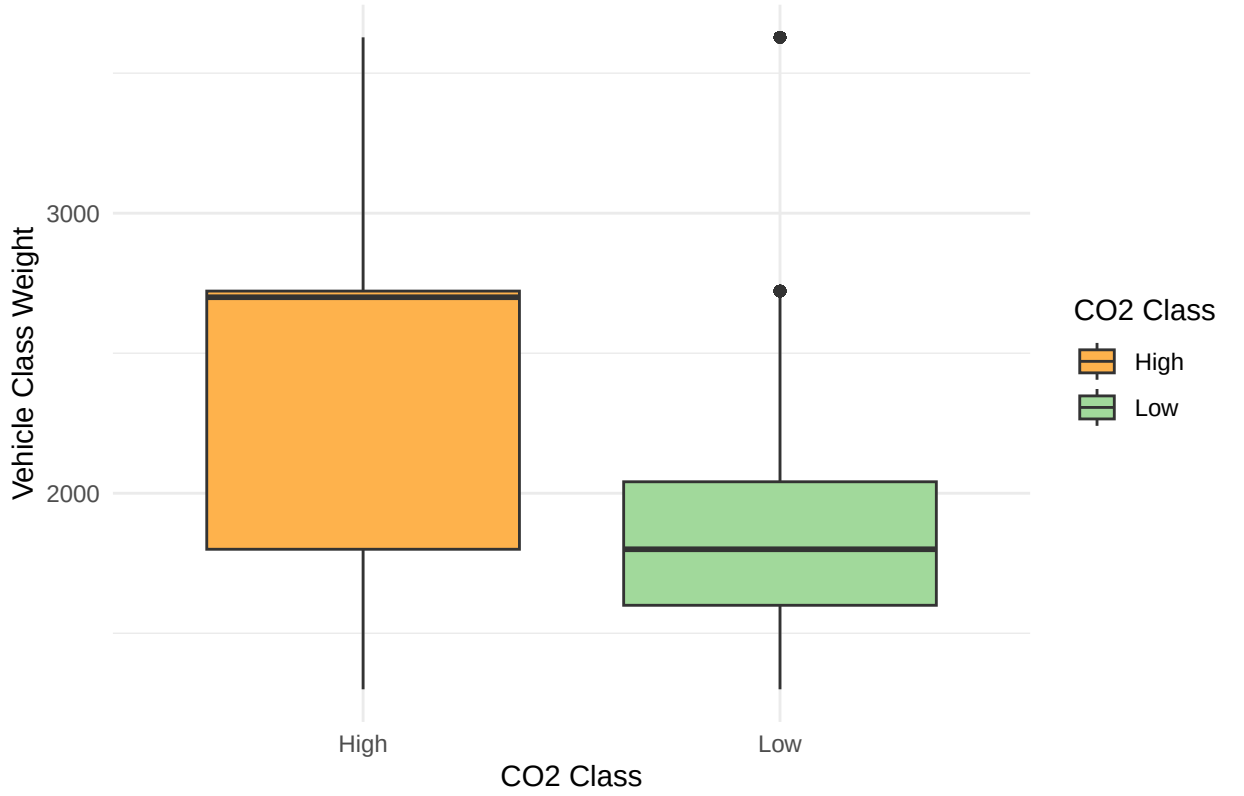
The summary statistics for the vehicle class weights indicate that the weights range from 1300 kg to 3628 kg in both the high and Low CO2 emissions class.

The average (mean) weight in the high CO2 emissions class is approximately 2419 kg. On the other hand, the average weight in the low CO2 emissions class is approximately 1879 kg. This implies that vehicles in this class generally have a lower weight compared to the high emissions class, which could be a contributing factor to their lower CO2 emissions.

In summary, the weight of the vehicle is a factor that appears to influence its CO2 emissions, with heavier vehicles typically exhibiting higher emissions, while lighter vehicles tend to have lower emissions. This can be visually explored in the Boxplots in Figure 1.5.5 below.

```r
# Custom color palette
my_colors <- c("#feb24c", "#a1d99b")

# Create the plot
ggplot(mydata, aes(x = CO2_classes, y = Weight_upto_kg, fill = CO2_classes)) +
  geom_boxplot() +
  scale_fill_manual(values = my_colors) +
  theme_minimal() +
  labs(x = "CO2 Class", y = "Vehicle Class Weight", fill = "CO2 Class") +
  ggtitle("Figure 1.5.5 Relationship between CO2 Class (High/Low CO2 Emissions) and Vehicle Class Weigh
```

Figure 1.5.5 Relationship between CO2 Class (High/Low CO2 Emissions)



CO2 Class (High/Low CO2 Emissions) and Transmission Type (Automatic/Manual)

The majority of vehicles in both the high and low emissions classes have automatic transmission. However, there are more vehicles with automatic transmission in the high emissions class compared to the low emissions class.

The type of transmission may have an impact on the CO2 emissions of vehicles, with automatic transmission potentially being associated with higher emissions. Further analysis and testing will be done to explore the relationship between transmission type and CO2 emissions in more detail.

```
# Distribution of data between CO2_Class=High (High CO2 Emissions) and Transmission Type
table(high$Auto_trans)
```

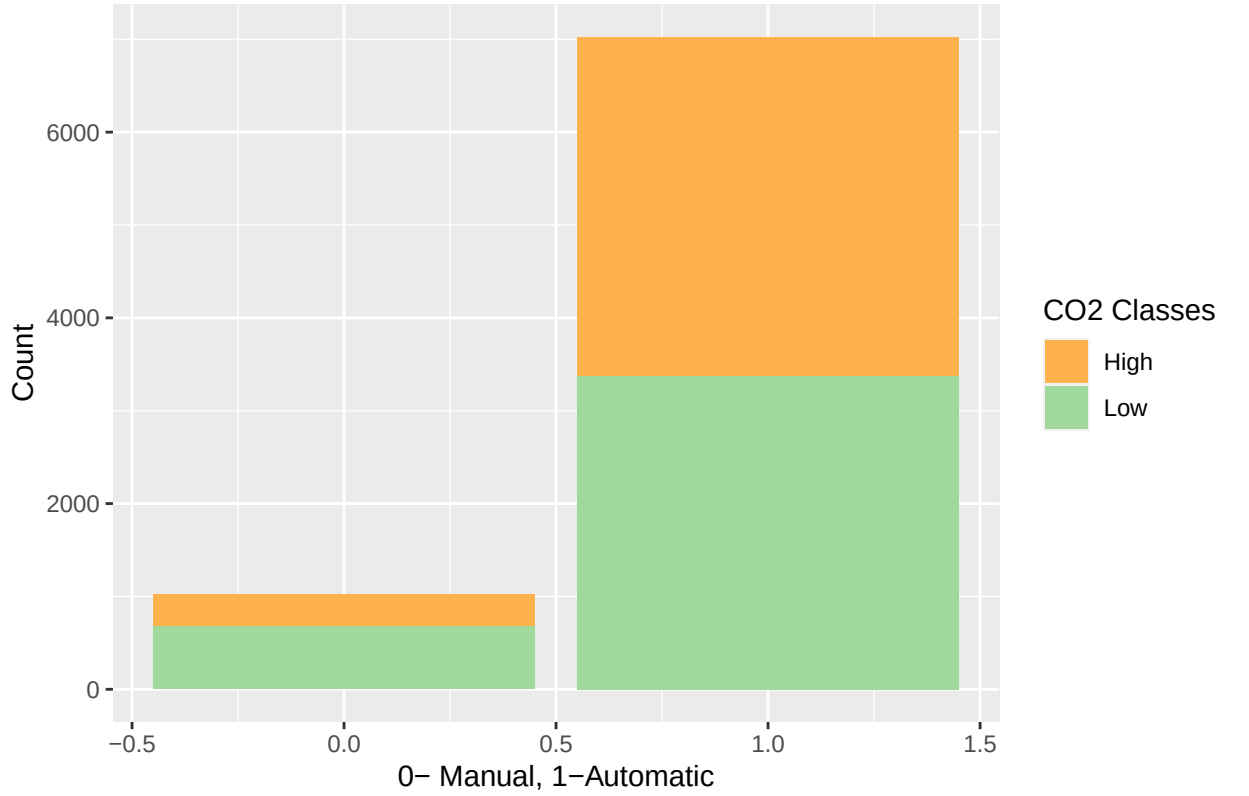
```
##
##    0    1
##  351 3650
```

```
# Distribution of data between CO2_Class=Low (Low CO2 Emissions) and Transmission Type
table(low$Auto_trans)
```

```
##
##    0    1
##  674 3371
```

```
# Create a bar plot with custom colors
ggplot(mydata, aes(x = Auto_trans, fill = CO2_classes)) +
  geom_bar() +
  scale_fill_manual(values = c("#feb24c", "#a1d99b")) +
  labs(x = "0- Manual, 1-Automatic", y = "Count", fill = "CO2 Classes") +
  ggtitle("Figure 1.5.6 Distribution of CO2 Classes (High/Low Emissions) by Transmission Type")
```

Figure 1.5.6 Distribution of CO2 Classes (High/Low Emissions) by Transm



CO2 Class (High/Low CO2 Emissions) and Fuel Type (Diesel(D)/ E85(E) / Regular Gasoline (X))

The output below suggests that the majority of vehicles in both the high and low emissions classes use Fuel type 2 (Regular Gasoline). Fuel types 1 (E85) and 3 (Diesel) are less common in both classes.

The type of fuel used by vehicles can have an impact on their CO2 emissions. Fuel type 2 appears to be more prevalent in both high and low emissions classes, indicating its importance in determining emission levels.

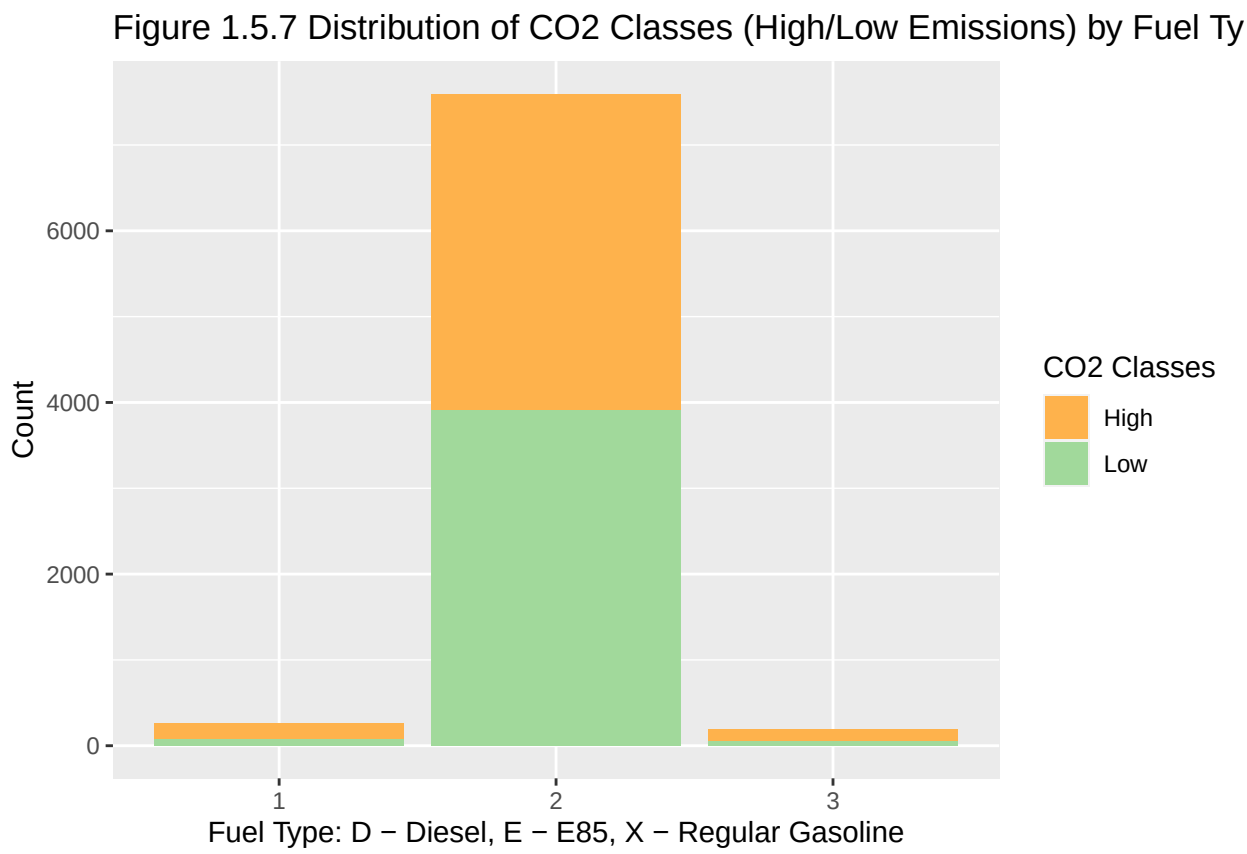
```
# Distribution of data between CO2_Class=High (High CO2 Emissions) and Fuel Type
table(high$Fuel)
```

```
##
##    1    2    3
##  179 3688  134
```

```
# Distribution of data between CO2_Class=Low (Low CO2 Emissions) and Fuel Type
table(low$Fuel)
```

```
##
##    1    2    3
##   78 3909   58
```

```r
# Create a bar plot with custom colors
ggplot(mydata, aes(x = Fuel, fill = CO2_classes)) +
  geom_bar() +
  scale_fill_manual(values = c("#feb24c", "#a1d99b")) +
  labs(x = "Fuel Type: D - Diesel, E - E85, X - Regular Gasoline", y = "Count", fill = "CO2 Classes") +
  ggtitle("Figure 1.5.7 Distribution of CO2 Classes (High/Low Emissions) by Fuel Type")
```



Two-way contingency table

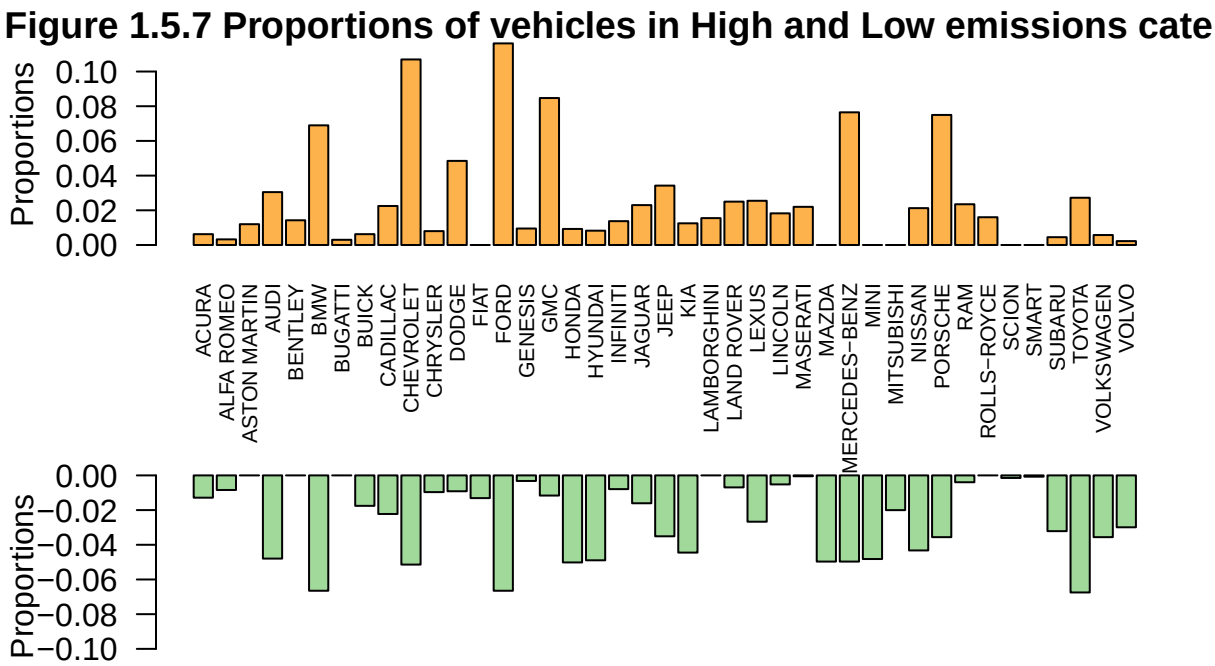
In this section, we will explore the relationship between the "Make" variable and the "CO2 Class" variable using a two-way contingency table. The "Make" variable represents the brand or manufacturer of the vehicles, while the "CO2 Class" variable categorizes the vehicles into high or low CO2 emissions classes.

```r
tab1<-table(mydata$CO2_classes, mydata$Make) #the joint table of CO2_class and Make
proportions = prop.table(tab1, margin = 1) #proportions table conditional on row
proportions
```

```
##
##                ACURA    ALFA ROMEO ASTON MARTIN         AUDI       BENTLEY
##   High 0.0062484379 0.0032491877 0.0119970007 0.0304923769 0.0142464384
##   Low  0.0128553770 0.0084054388 0.0000000000 0.0479604450 0.0000000000
##
##                  BMW      BUGATTI        BUICK     CADILLAC    CHEVROLET
##   High 0.0689827543 0.0029992502 0.0062484379 0.0224943764 0.1069732567
```

```
##   Low  0.0665018541 0.0000000000 0.0175525340 0.0222496910 0.0514215080
##
##           CHRYSLER        DODGE         FIAT         FORD      GENESIS
##   High 0.0079980005 0.0484878780 0.0000000000 0.1162209448 0.0094976256
##   Low  0.0096415328 0.0091470952 0.0131025958 0.0665018541 0.0032138443
##
##                GMC        HONDA      HYUNDAI     INFINITI       JAGUAR
##   High 0.0847288178 0.0092476881 0.0082479380 0.0137465634 0.0229942514
##   Low  0.0116192831 0.0501854141 0.0489493201 0.0079110012 0.0160692213
##
##               JEEP          KIA  LAMBORGHINI   LAND ROVER        LEXUS
##   High 0.0342414396 0.0124968758 0.0154961260 0.0249937516 0.0254936266
##   Low  0.0351050680 0.0444993820 0.0000000000 0.0069221261 0.0266996292
##
##            LINCOLN     MASERATI        MAZDA MERCEDES-BENZ         MINI
##   High 0.0182454386 0.0219945014 0.0000000000  0.0764808798 0.0000000000
##   Low  0.0051915946 0.0004944376 0.0496909765  0.0496909765 0.0482076638
##
##          MITSUBISHI       NISSAN      PORSCHE          RAM  ROLLS-ROYCE
##   High 0.0000000000 0.0212446888 0.0749812547 0.0234941265 0.0159960010
##   Low  0.0200247219 0.0432632880 0.0355995056 0.0039555006 0.0000000000
##
##              SCION        SMART       SUBARU       TOYOTA   VOLKSWAGEN
##   High 0.0000000000 0.0000000000 0.0044988753 0.0272431892 0.0057485629
##   Low  0.0014833127 0.0007416564 0.0321384425 0.0674907293 0.0355995056
##
##              VOLVO
##   High 0.0022494376
##   Low  0.0299134734
```

Based on proportions table, the visualization of Vehicles' Makes in High and Low emissions categories provided below (Figure 1.5.8).



From the barplots, we can observe that certain brands, such as FORD, CHEVROLET and BMW make relatively balanced contributions to both the High and Low emissions categories.

Some Makes, such as ASTON MARTINI, LAMBORGHINI and MASERATI have a very low or zero proportion in the Low CO2 class, indicating they are more commonly associated with higher emissions vehicles.

TOYOTA, HONDA, MAZDA, and HYUNDAI make a significant contribution to low-emission vehicles compared to high-emission vehicles among others.

In general, the proportions vary across different Make categories, suggesting that different manufacturers have different distributions of vehicles in terms of CO2 emissions.

```
chisq.test(tab1)
```

```
## Warning in chisq.test(tab1): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab1
## X-squared = 2279.1, df = 40, p-value < 2.2e-16
```

Based on the very low p-value, we can conclude that there is a significant association between CO2_class and Vehicle_Make. In other words, the CO2 emission classes are not independent of the vehicle makes, and there is a relationship between these two variables. However, the test may not be accurate as there are small observed frequencies in some cells.

As the assumptions of the Chi-squared test are violated, we applied another statistical test that is called Fisher's Exact Test.

```
# Apply Fisher's exact test with simulate.p.value=TRUE
fisher_result <- fisher.test(tab1, simulate.p.value = TRUE)

# Print the test result
print(fisher_result)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  tab1
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

This means that under the null hypothesis (assuming independence between the variables), the observed data is unlikely to occur by chance alone. The small p-value suggests that there is a significant association between the variables in the contingency table.

Therefore, we can conclude that there is evidence to reject the null hypothesis and suggest that there is a relationship between the CO2_class and Vehicle_Make variables.

2 Analysis

2.1 Logistic regression

The code below allows to check how R dummifies the "CO2_class" variable: 1 is associated with "Low" emissions and 0 with "High" emissions.

```
#Check how R dummifies the "CO2_class" variable

contrasts(mydata$CO2_classes)
```

```
##      Low
## High   0
## Low    1
```

```
#Check the order of class

unique(mydata$CO2_classes)
```

```
## [1] Low  High
## Levels: High Low
```

The code below defines the proportion of training set (75% of the original dataset). The proportions of "High" and "Low" CO2 classes in training set reflect the proportions of these classes in the original dataset.

```r
#calculating the appropriate proportions of classes for training set

percent_75 = 3/4*dim(mydata)[1] #75% of data for training set
high_class = table(mydata$CO2_classes)[1]/dim(mydata)[1] #proportion of high class
low_class = table(mydata$CO2_classes)[2]/dim(mydata)[1] #proportion of low class

sample_high_class=round(high_class*percent_75,0) #number of high class in training set
sample_low_class=round(low_class*percent_75,0) #proportion of low class in training set

N = sample_high_class + sample_low_class #size of all training set

c(sample_high_class, sample_low_class)
```

```
## High  Low
## 3001 3034
```

Then, we split the data into training and testing sets using stratified sampling with "CO2_classes" strata and proportions defined in the code above.

```r
#splitting to train and test sets

set.seed(2023)
idx=sampling:::strata(mydata, stratanames=c("CO2_classes"), size=c(sample_low_class, sample_high_class)
train=mydata[idx$ID_unit,]
test=mydata[-idx$ID_unit,]
```

Fitting the model with variables of interest:

```r
# Get a logistic regression(Full) model based on the training set

log_model<-glm(CO2_classes~Year_of_model+Engine_Size+factor(Cylinders)+Weight_upto_kg+Auto_trans+factor
summary(log_model)
```

```
##
## Call:
## glm(formula = CO2_classes ~ Year_of_model + Engine_Size + factor(Cylinders) +
##     Weight_upto_kg + Auto_trans + factor(Fuel), family = binomial,
##     data = train)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           2.214e+02  3.743e+02   0.592    0.554
## Year_of_model        -9.785e-02  2.023e-02  -4.837 1.32e-06 ***
## Engine_Size          -1.960e+00  1.481e-01 -13.231  < 2e-16 ***
## factor(Cylinders)4   -1.353e+01  3.720e+02  -0.036    0.971
## factor(Cylinders)5   -1.386e+01  3.720e+02  -0.037    0.970
## factor(Cylinders)6   -1.471e+01  3.720e+02  -0.040    0.968
## factor(Cylinders)8   -1.755e+01  3.720e+02  -0.047    0.962
## factor(Cylinders)10 -2.797e+01  6.867e+02  -0.041    0.968
## factor(Cylinders)12 -2.617e+01  4.829e+02  -0.054    0.957
## factor(Cylinders)16 -2.247e+01  1.199e+03  -0.019    0.985
## Weight_upto_kg       -1.270e-03  8.119e-05 -15.642  < 2e-16 ***
```

```
## Auto_trans           9.197e-01  1.348e-01   6.824 8.85e-12 ***
## factor(Fuel)2       -1.975e+00  2.749e-01  -7.184 6.77e-13 ***
## factor(Fuel)3       -2.954e+00  3.690e-01  -8.005 1.19e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8366.1  on 6034  degrees of freedom
## Residual deviance: 3318.9  on 6021  degrees of freedom
## AIC: 3346.9
##
## Number of Fisher Scoring iterations: 16
```

The model results indicate that several variables have significant effects on the prediction. Specifically, 'Year_of_model', 'Engine_Size', 'Weight_upto_kg', 'Auto_trans', and 'Fuel' show statistically significant coefficients.

```
# Drop the "Cylinder" variable to get the best model based on the training set

bestmodel<-glm(CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel), family=bin
summary(bestmodel)
```

```
##
## Call:
## glm(formula = CO2_classes ~ Year_of_model + Engine_Size + Weight_upto_kg +
##     Auto_trans + factor(Fuel), family = binomial, data = train)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     2.285e+02  4.029e+01   5.672 1.41e-08 ***
## Year_of_model  -1.070e-01  1.995e-02  -5.366 8.03e-08 ***
## Engine_Size    -2.953e+00  8.140e-02 -36.271  < 2e-16 ***
## Weight_upto_kg -1.193e-03  7.975e-05 -14.955  < 2e-16 ***
## Auto_trans      9.348e-01  1.351e-01   6.919 4.54e-12 ***
## factor(Fuel)2  -2.143e+00  2.726e-01  -7.862 3.77e-15 ***
## factor(Fuel)3  -2.994e+00  3.587e-01  -8.346  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8366.1  on 6034  degrees of freedom
## Residual deviance: 3386.9  on 6028  degrees of freedom
## AIC: 3400.9
##
## Number of Fisher Scoring iterations: 7
```

After dropping the "Cylinder" variable, the logistic regression model was refitted using the remaining variables based on the training set. The updated model, referred to as the best model, includes 'Year_of_model', 'Engine_Size', 'Weight_upto_kg', 'factor(Auto_trans)', and 'factor(Fuel)' as predictor variables.

The best model demonstrates statistically significant effects for all remaining variables at the 0.05 significance level. These variables, namely 'Year_of_model', 'Engine_Size', 'Weight_upto_kg', 'factor(Auto_trans)', and 'factor(Fuel)', have a significant impact on predicting the CO2 emission class.

The goodness-of-fit measures indicate that the best model exhibits a lower residual deviance compared to the null deviance, suggesting a reasonable fit to the data. Despite the slightly higher AIC compared to the full model (3363.4 vs. 3414.1), the best model strikes a balance between model complexity and fit.

When selecting a model, it is essential to consider the significance and interpretability of the variables. The best model, by removing the non-significant "Cylinder" variable, provides a more interpretable and focused representation of the relationship between the predictor variables and the CO2 emission class.

Although the AIC did not decrease in the best model, we prioritize the significance and interpretability of the variables. By considering only the variables that have a significant impact, the best model offers a more accurate and reliable prediction of the CO2 emission class.

In summary, we select the best logistic regression model as it provides a simplified yet statistically significant prediction of the CO2 emission class based on the selected features. This model ensures a good balance between goodness of fit and simplicity, enhancing our understanding of the relationship between the predictor variables and the CO2 emission class.

```
VIF(bestmodel)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Year_of_model  1.068789  1        1.033823
## Engine_Size    1.149254  1        1.072033
## Weight_upto_kg 1.198980  1        1.094980
## Auto_trans     1.154836  1        1.074633
## factor(Fuel)   1.161835  2        1.038212
```

From the VIF values, it can be seen that there is no multicollinearity between predictor variables (all values < 5).

```
#Apply the full logistic regression model to the test set

Prob.predict1<-predict(log_model, test, type="response")
Predict1<-rep("High",dim(test)[1])
Predict1[Prob.predict1>=0.5]="Low"
Actual<-test$CO2_class
table(Predict1, Actual)
```

```
##         Actual
## Predict1 High Low
##     High  869 139
##     Low   131 872
```

```
#checking the misclassification rate of

Misc_rate_logreg = 1-mean(Predict1==test$CO2_class)
Misc_rate_logreg
```

```
## [1] 0.1342616
```

```
#Apply the best model to the test set

Prob.predict<-predict(bestmodel, test, type="response")
Predict<-rep("High",dim(test)[1])
```

```r
Predict[Prob.predict>=0.5]="Low"
Actual<-test$CO2_class
table(Predict, Actual)
```

```
##         Actual
## Predict High Low
##    High  848 117
##    Low   152 894
```

```r
#checking the misclassification rate

Misc_rate_logreg = 1-mean(Predict==test$CO2_class)
Misc_rate_logreg
```

```
## [1] 0.1337643
```

After applying the best logistic regression model to the test set, the predicted probabilities for each observation were obtained using the predict function. The Predict variable was initialized with "High" for all observations, and then modified based on the predicted probabilities. If the predicted probability of an observation belonging to the "Low" class was greater than or equal to 0.5, the corresponding element in Predict was changed to "Low". The actual class labels were stored in the Actual variable.

The contingency table, generated using the table(Predict, Actual) function, compares the predicted classes (Predict) with the actual classes (Actual) from the test set. The table shows that there were 848 observations classified as "High" and correctly predicted as "High", 894 observations classified as "Low" and correctly predicted as "Low", while 117 observations were classified as "High" but actually belonged to the "Low" class. Additionally, 152 observations were classified as "Low" but were actually "High".

The misclassification rate, calculated as 1 minus the proportion of correct predictions, was 0.1337643 or approximately 13.38%. This indicates that approximately 13.38% of the observations in the test set were misclassified by the best logistic regression model.

It is worth noting that the misclassification rate of the best model (0.1337643) is lower than that of the full model (0.1342616). This suggests that the best model, despite having a slightly higher AIC, provides improved accuracy in predicting the CO2 emission class compared to the full model.

To assess the performance and generalization ability of our logistic model, we apply 10-fold cross-validation technique. By splitting the dataset into 10 subsets (folds), it allows for training and evaluating the model 10 times using different combinations of training and validation data. This will help to estimate how well the model will perform on unseen data and provides a more reliable evaluation metric compared to a single train-test split.

```r
set.seed(2023)
folds<-createFolds(factor(mydata$CO2_classes), k=10)
```

```r
#checking Accuracy by cross-validation method for logistic regression

model_fit_log<-train(CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel), data=
model_fit_log
```

```
## Generalized Linear Model
##
## 8046 samples
```

```
##    5 predictor
##    2 classes: 'High', 'Low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 805, 804, 804, 804, 806, 805, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8725799  0.7451129
```

The logistic regression model, trained using the glm method, achieved an accuracy of approximately 0.8726 and a kappa statistic of 0.7451. These performance metrics were obtained through cross-validation with a 10-fold resampling strategy.

The model was fitted on a dataset consisting of 8,046 samples and included five predictor variables: Year_of_model, Engine_Size, Weight_upto_kg, factor(Auto_trans), and factor(Fuel). The response variable, CO2_classes, had two classes: 'High' and 'Low'.

The results indicate that the logistic regression model achieved a relatively high accuracy of 0.8726, indicating that it correctly classified approximately 87.26% of the samples. The kappa statistic of 0.7451 suggests a substantial agreement between the model's predictions and the true classifications beyond what could be expected by chance alone.

```
#Apply the best logistic regression model to the whole set

Prob.predict_whole<-predict(bestmodel, mydata, type="response")
Predict_whole<-rep("High",dim(mydata)[1])
Predict_whole[Prob.predict_whole>=0.5]="Low"
Actual_whole<-mydata$CO2_classes
table(Predict_whole, Actual_whole)
```

```
##              Actual_whole
## Predict_whole High  Low
##          High 3437  459
##          Low   564 3586
```

```
#checking the misclassification rate

Misc_rate_logreg_whole = 1-mean(Predict_whole==Actual_whole)
Misc_rate_logreg_whole
```

```
## [1] 0.1271439
```

Applying the best logistic regression model to the entire dataset, we obtained predictions for the CO2_classes. The resulting table shows the comparison between the predicted classes (High and Low) and the actual classes from the dataset.

In terms of accuracy, the model correctly classified 3437 samples as High CO2 emissions and 3586 samples as Low CO2 emissions. However, it misclassified 459 samples as Low when they were actually High, and 564 samples as High when they were actually Low.

Calculating the misclassification rate, we find that the overall misclassification rate for the logistic regression model applied to the entire dataset is approximately 0.1271, or 12.71%. This indicates that the model's predictions for the CO2 emission classes have an error rate of 12.71%.

```r
unique(test$Auto_trans)
```

```
## [1] 1 0
```

```r
unique(mydata$Auto_trans)
```

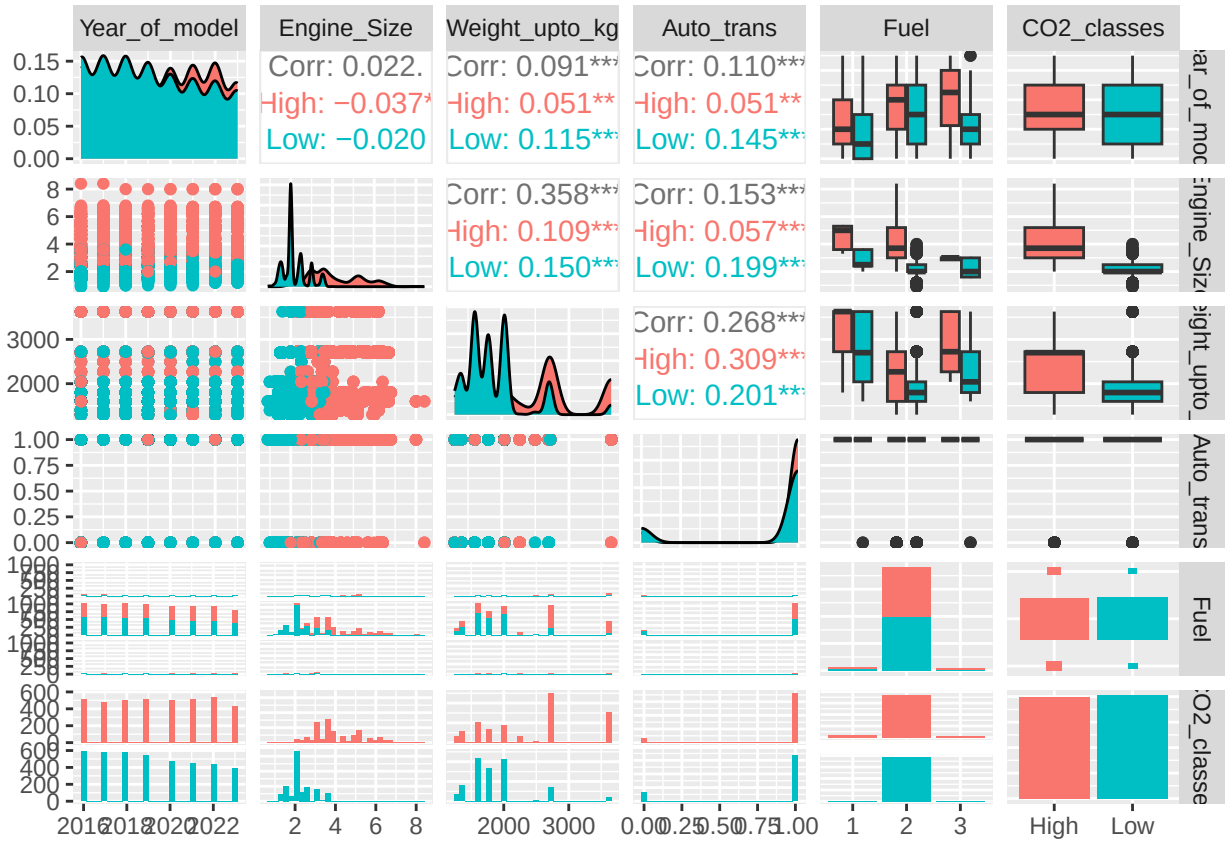
```
## [1] 1 0
```

2.2 Linear Discriminant Analysis

LDA Assumptions

```r
# Select the variables
my_variables <- c("Year_of_model", "Engine_Size",
    "Weight_upto_kg", "Auto_trans", "Fuel", "CO2_classes")

# Create a scatterplot matrix
ggpairs(mydata, columns = my_variables, mapping = aes(color = CO2_classes))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Multivariate Normality test:From the density plot in ggpairs, the variables in our dataset don't follow a multivariate distribution within each class of CO2_class. Equality of variances:From the box plots, the variances of variables are not equal across different levels of the CO2_class variable.

Using tests to check the Assumptions of Linear Discriminant Analysis

Multivariate normality

H0 (null): The variables follow a multivariate normal distribution within each class of CO2_class. Ha (alternative): The variables do not follow a multivariate normal distribution within each class of CO2_class.

An Energy Test is a statistical test that determines whether or not a group of variables follows a multivariate normal distribution.

```
# Subset the data for each CO2 class
low_emissions <- subset(mydata, CO2_classes == "Low")
high_emissions <- subset(mydata, CO2_classes == "High")

# Extract the predictor variables for each class
predictors_low <- subset(low_emissions, select = c("Year_of_model", "Engine_Size",
    "Weight_upto_kg", "Auto_trans", "Fuel"))
predictors_high <- subset(high_emissions, select = c("Year_of_model", "Engine_Size",
    "Weight_upto_kg", "Auto_trans", "Fuel"))

#perform Multivariate normality test
mvnorm.etest(predictors_low, R=100)
```

```
##
##  Energy test of multivariate normality: estimated parameters
##
## data:  x, sample size 4045, dimension 5, replicates 100
## E-statistic = 257.07, p-value < 2.2e-16
```

```
mvnorm.etest(predictors_high, R=100)
```

```
##
##  Energy test of multivariate normality: estimated parameters
##
## data:  x, sample size 4001, dimension 5, replicates 100
## E-statistic = 190.55, p-value < 2.2e-16
```

The p-value of the test is < 2.2e-16. Since this is less 0.05, we accept the null hypothesis of the test. We don't have evidence to say that the variables in our dataset follow a multivariate distribution within each class of CO2_class.

Equality of Variances

H0 (null): the variances of the predictor variable are equal across different levels of the CO2_class variable. Ha (alternative): the variances of the predictor variable are not equal across different levels of the CO2_class variable.

Multivariate Normality test: From the density plot in ggpairs, the variables in our dataset don't follow a multivariate distribution within each class of CO2_class. Equality of variances: From the box plots, the variances of variables are not equal across different levels of the CO2_class variable.

```
# Perform Levene's test for each variable
leveneTest(Year_of_model ~ mydata$CO2_classes, data = mydata)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value  Pr(>F)
## group    1  3.6008 0.05779 .
##       8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Engine_Size ~ mydata$CO2_classes, data = mydata)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  1682.2 < 2.2e-16 ***
##       8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Weight_upto_kg ~ mydata$CO2_classes, data = mydata)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  1340.5 < 2.2e-16 ***
##       8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Auto_trans ~ mydata$CO2_classes, data = mydata)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1   114.2 < 2.2e-16 ***
##       8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
leveneTest(Fuel ~ mydata$CO2_classes, data = mydata)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##         Df F value    Pr(>F)
## group    1  76.671 < 2.2e-16 ***
##       8044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-values of the test for all predictor variables are < 0.05. Therefore, we accept the null hypothesis of the test. We don't have evidence to say that the variances of the predictor variable are equal across different levels of the CO2_class variable.

LDA model

```
#get a linear discriminant analysis (LDA) model

lda_model<-lda(CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel), data=train)

lda_model
```

```
## Call:
## lda(CO2_classes ~ Year_of_model + Engine_Size + Weight_upto_kg +
##     Auto_trans + factor(Fuel), data = train)
##
## Prior probabilities of groups:
##      High       Low
## 0.4972659 0.5027341
##
## Group means:
##      Year_of_model Engine_Size Weight_upto_kg Auto_trans factor(Fuel)2
## High      2019.444    4.118894       2415.340  0.9136954     0.9233589
## Low       2019.194    2.194759       1874.528  0.8295979     0.9657218
##      factor(Fuel)3
```

```
## High    0.03432189
## Low     0.01549110
##
## Coefficients of linear discriminants:
##                          LD1
## Year_of_model  -0.0161049459
## Engine_Size    -0.9726801059
## Weight_upto_kg -0.0005136552
## Auto_trans      0.1587510743
## factor(Fuel)2  -0.5361468718
## factor(Fuel)3  -1.4318313222
```

```
summary(lda_model)
```

```
##         Length Class  Mode
## prior    2     -none- numeric
## counts   2     -none- numeric
## means   12     -none- numeric
## scaling  6     -none- numeric
## lev      2     -none- character
## svd      1     -none- numeric
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  1     -none- list
```

Variables with larger absolute coefficients are considered more important in separating the classes. In this case, variables like "Engine_Size", "factor(Fuel)3", "factor(Fuel)2" have larger absolute coefficients, indicating their greater contribution to the discriminant function.

To further assess the significance of variables of LDA model, we will use statistical test of Wilks' lambda.

```
# Extract Wilks' lambda
wilks_lambda <- lda_model$scaling^2
wilks_lambda
```

```
##                          LD1
## Year_of_model  2.593693e-04
## Engine_Size    9.461066e-01
## Weight_upto_kg 2.638417e-07
## Auto_trans     2.520190e-02
## factor(Fuel)2  2.874535e-01
## factor(Fuel)3  2.050141e+00
```

The values extracted as Wilks' lambda represent the proportion of the variance in each predictor variable that is not explained by the linear discriminant function. So, a smaller value of Wilks' lambda indicates that the predictor variable contributes more to the separation of the groups. Therefore, the "Year_of_model", "Weight_upto_kg", and "Auto_trans" variables seem to be the most important predictors in discriminating between groups, based on their Wilks' lambda values.

```
#Apply the fitted LDA model to the test set

lda.pred=predict(lda_model, test)
table(lda.pred$class, test$CO2_classes)
```

```
##
##        High Low
##   High  781  85
##   Low   219 926
```

```
#check the misclassification rate

Misc_rate_lda = 1-mean(lda.pred$class==test$CO2_classes)
Misc_rate_lda
```

```
## [1] 0.1511686
```

QDA model

```
#get a quadratic discriminant analysis (QDA) model

qda_model<-qda(CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel), data=train)
qda_model
```

```
## Call:
## qda(CO2_classes ~ Year_of_model + Engine_Size + Weight_upto_kg +
##     Auto_trans + factor(Fuel), data = train)
##
## Prior probabilities of groups:
##      High       Low
## 0.4972659 0.5027341
##
## Group means:
##      Year_of_model Engine_Size Weight_upto_kg Auto_trans factor(Fuel)2
## High      2019.444    4.118894       2415.340  0.9136954     0.9233589
## Low       2019.194    2.194759       1874.528  0.8295979     0.9657218
##      factor(Fuel)3
## High    0.03432189
## Low     0.01549110
```

```
qda_class<-predict(qda_model, test)$class
table(qda_class, test$CO2_classes)
```

```
##
## qda_class High Low
##      High  785 116
##      Low   215 895
```

```
#check the misclassification rate

Misc_rate_qda = 1-mean(qda_class==test$CO2_classes)
Misc_rate_qda
```

```
## [1] 0.1645947
```

The misclassification rate in the QDA model (0.1645947) gets increased comparing to the LDA model (0.1511686).

Accuracy in LDA, QDA models

```
# Checking Accuracy by cross-validation method for LDA
# Set up cross-validation using trainControl
folds <- createFolds(mydata$CO2_classes, k = 10)

# Train the LDA model using cross-validation
model_fit_lda <- train(
  CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel),
  data = mydata,
  trControl = trainControl(method = "cv", index = folds),
  method = 'lda'
)
model_fit_lda
```

```
## Linear Discriminant Analysis
##
## 8046 samples
##    5 predictor
##    2 classes: 'High', 'Low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 805, 804, 805, 805, 805, 805, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8552765  0.7103416
```

```
# Checking Accuracy by cross-validation method for QDA

# Set up cross-validation using trainControl
folds <- createFolds(mydata$CO2_classes, k = 10)

# Train the LDA model using cross-validation
model_fit_qda <- train(
  CO2_classes~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+factor(Fuel),
  data = mydata,
  trControl = trainControl(method = "cv", index = folds),
  method = 'qda'
)
model_fit_qda
```

```
## Quadratic Discriminant Analysis
##
## 8046 samples
##    5 predictor
##    2 classes: 'High', 'Low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 804, 804, 804, 805, 805, 804, ...
```

```
## Resampling results:
##
##   Accuracy   Kappa
##   0.8373652  0.6745237
```

```
#Apply the fitted LDA model to the whole set

lda.pred.whole=predict(lda_model, mydata)
table(lda.pred.whole$class, mydata$CO2_classes)
```

```
##
##        High  Low
##   High 3200  341
##   Low   801 3704
```

```
#check the misclassification rate

Misc_rate_lda = 1-mean(lda.pred.whole$class==mydata$CO2_classes)
Misc_rate_lda
```

```
## [1] 0.1419339
```

```
#Apply the fitted QDA model to the whole set

qda.pred.whole=predict(qda_model, mydata)
table(qda.pred.whole$class, mydata$CO2_classes)
```

```
##
##        High  Low
##   High 3179  482
##   Low   822 3563
```

```
#check the misclassification rate

Misc_rate_qda = 1-mean(qda.pred.whole$class==mydata$CO2_classes)
Misc_rate_qda
```

```
## [1] 0.1620681
```

After applying the LDA and QDA models on the whole dataset, the misclassification rate in the QDA model (0.1620681) is still higher comparing to the LDA model (0.1419339).

2.3 Classification decision tree

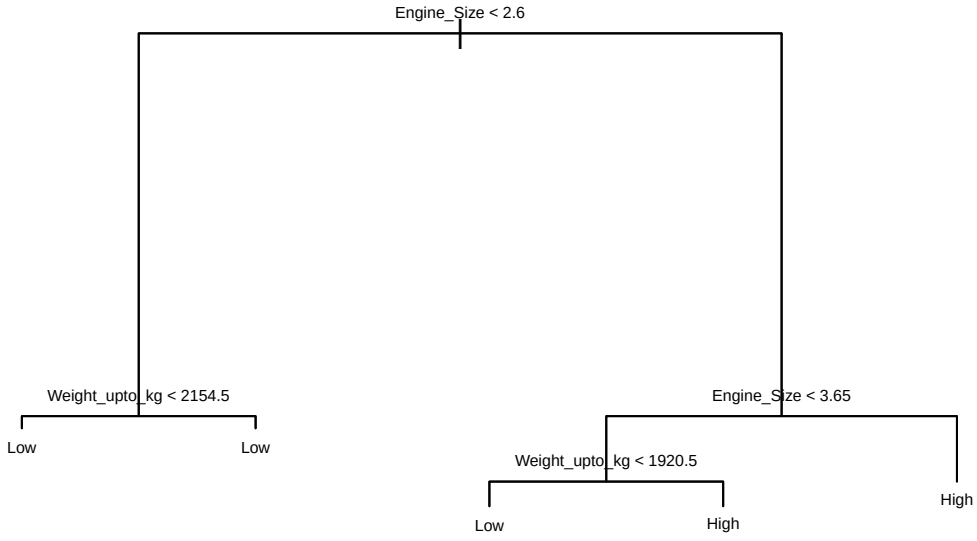
In this section, we will be performing a classification tree to predict the classes of CO2 emissions of different car models. We first fit a classification tree to the training set using the 'tree' function. The constructed decision tree uses the Year of the Model, Engine Size, Weight up to a certain kilogram limit, Auto Transmission, and Fuel type as predictors. The performance of this initial tree is then evaluated using the misclassification error rate, highlighting the proportion of instances that the model has incorrectly predicted.

To improve the model's performance, the tree was pruned using cross-validation, with the misclassification error as the criterion. The resulting pruned tree consisted of two terminal nodes and achieved a misclassification error rate of 0.12, or 12%. This indicates that approximately 12% of the observations in the dataset were misclassified by the pruned tree. The pruned tree primarily focused on the Engine_Size variable, dividing the samples into High and Low CO2 emission classes based on an engine size threshold of 2.6.

```
#fit a classification tree to the training set

tree.class<-tree(factor(CO2_classes)~Year_of_model+Engine_Size+Weight_upto_kg+Auto_trans+Fuel, train)
summary(tree.class)
```

```
##
## Classification tree:
## tree(formula = factor(CO2_classes) ~ Year_of_model + Engine_Size +
##     Weight_upto_kg + Auto_trans + Fuel, data = train)
## Variables actually used in tree construction:
## [1] "Engine_Size"    "Weight_upto_kg"
## Number of terminal nodes:  5
## Residual mean deviance:  0.5314 = 3204 / 6030
## Misclassification error rate: 0.1178 = 711 / 6035
```

```
#plot the tree

plot(tree.class)
text(tree.class, pretty=0, cex = 0.5)
```



The resulting unpruned tree consisted of six terminal nodes and exhibited a misclassification error rate of approximately 0.118, indicating that around 11.8% of the observations were misclassified. The variables Engine_Size and Weight_upto_kg played significant roles in the construction of the tree.

When applying the unpruned tree to the test set, it correctly classified 808 samples as High CO2 emissions and 948 samples as Low CO2 emissions. However, it misclassified 63 samples as Low when they were actually High and 192 samples as High when they were actually Low.

```
#apply the fitted tree to the test set

tree.pred<-predict(tree.class,test,type = "class")
table(tree.pred,test$CO2_classes)
```

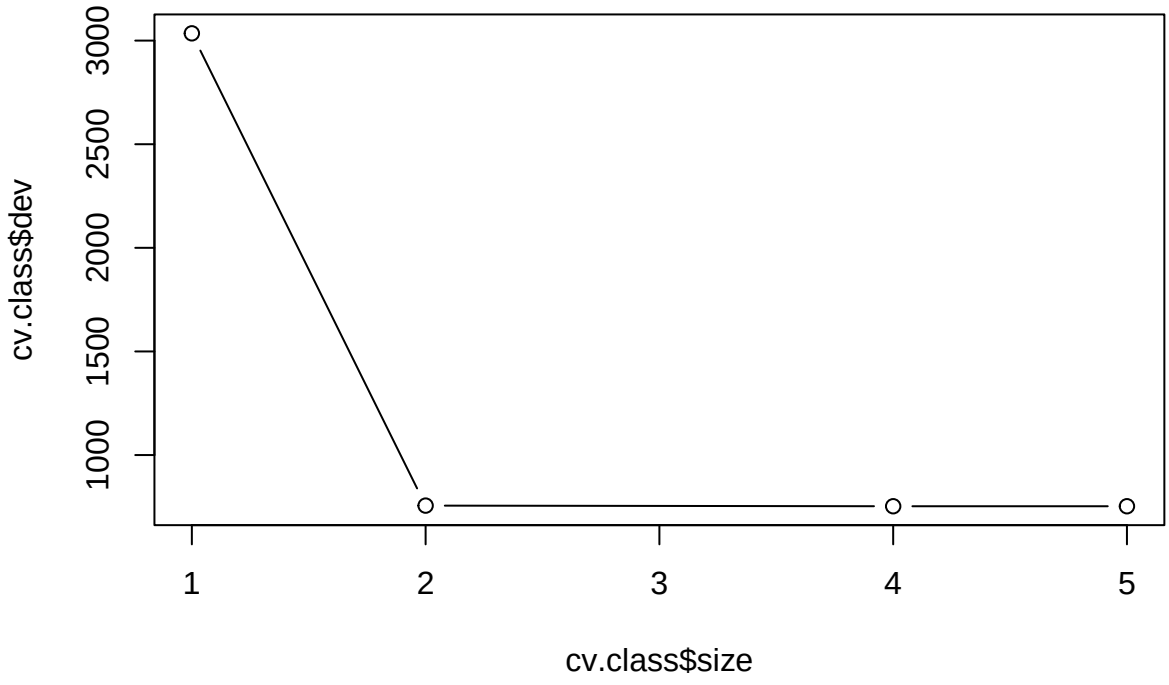
```
##
## tree.pred High Low
##      High  808  63
##      Low   192 948
```

```
#check the misclassification rate

Misc_rate_class_tree = 1- mean(tree.pred==test$CO2_classes)
Misc_rate_class_tree
```

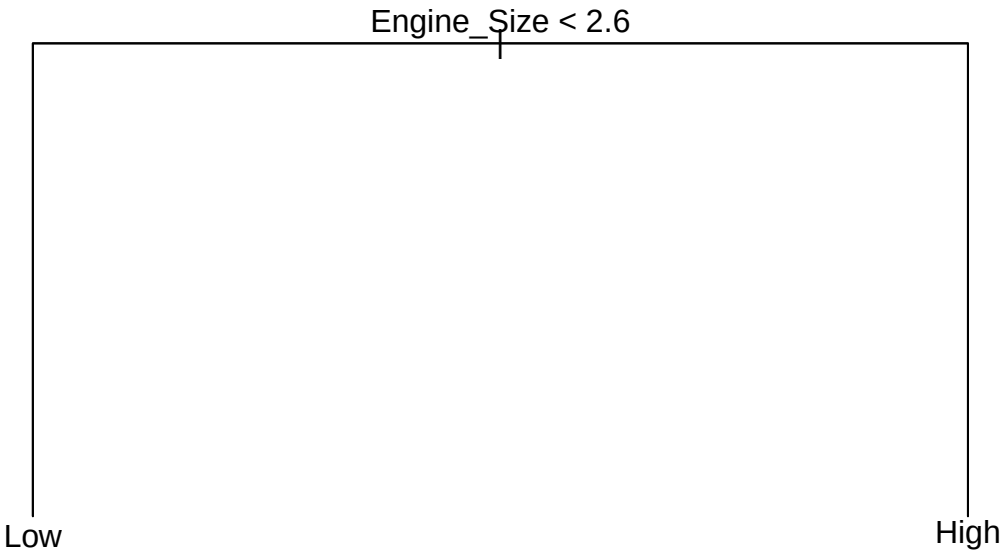
```
## [1] 0.1268026
```

To improve the model's performance, the tree was pruned using cross-validation, with the misclassification error as the criterion. The resulting pruned tree consisted of two terminal nodes and achieved a misclassification error rate of 0.127, or 12.7%. This indicates that approximately 12.7% of the observations in the dataset were misclassified by the pruned tree. The pruned tree primarily focused on the Engine_Size variable, dividing the samples into High and Low CO2 emission classes based on an engine size threshold of 2.6.

```
#Prune the tree by checking the cross-validation errors

set.seed(2023)
cv.class<-cv.tree(tree.class, FUN = prune.misclass, K=10)
plot(cv.class$size, cv.class$dev,type="b")
```




```
#Plot the pruned tree

prune.class=prune.tree(tree.class,best=2)
plot(prune.class)
text(prune.class,pretty=0)
```




```
prune.class
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
## 1) root 6035 8366 Low ( 0.49727 0.50273 )
##   2) Engine_Size < 2.6 2688 1358 Low ( 0.06957 0.93043 ) *
##   3) Engine_Size > 2.6 3347 2935 High ( 0.84075 0.15925 ) *
```

The pruned classification tree consists of two nodes.

At the root node, there are a total of 6035 observations. The deviance is 8366, and the predicted class is "Low" with a probability of 0.49727 and "High" with a probability of 0.50273.

The first split is based on the "Engine_Size" variable. If the engine size is less than 2.6, there are 2688 observations. The predicted class is "Low" with a probability of 0.06957 and "High" with a probability of 0.93043. This is a terminal node.

If the engine size is greater than 2.6, there are 3347 observations. The predicted class is "High" with a probability of 0.84075 and "Low" with a probability of 0.15925. This is also a terminal node.

In summary, the pruned tree suggests that for engine sizes less than 2.6, the predicted class is "Low", while for engine sizes greater than 2.6, the predicted class is "High".

```
summary(prune.class)
```

```
##
```

```
## Classification tree:
## snip.tree(tree = tree.class, nodes = 2:3)
## Variables actually used in tree construction:
## [1] "Engine_Size"
## Number of terminal nodes:  2
## Residual mean deviance:  0.7115 = 4292 / 6033
## Misclassification error rate: 0.1193 = 720 / 6035
```

The classification tree was pruned to have two terminal nodes, and the only variable used in constructing the pruned tree was "Engine_Size." The pruned tree achieved a residual mean deviance of 0.7115, indicating a reasonably good fit to the data. However, the misclassification error rate is 0.119, suggesting that approximately 11.9% of the observations are misclassified by the pruned tree. Therefore, the pruned tree provides an accuracy of 88.1%, correctly classifying 88.1% of the observations.

```
#Apply the fitted tree model to the whole set
tree.pred.whole <- predict(prune.class, mydata, type = "class")
table(tree.pred.whole, mydata$CO2_classes)
```

```
##
## tree.pred.whole High  Low
##            High 3733  723
##            Low   268 3322
```

```
#check the misclassification rate

Misc_rate_pruned_class_tree = 1- mean(tree.pred.whole==mydata$CO2_classes)
Misc_rate_pruned_class_tree
```

```
## [1] 0.1231668
```

When applying the pruned tree to the entire dataset, it correctly classified 3,733 samples as High CO2 emissions and 3,322 samples as Low CO2 emissions. However, it misclassified 723 samples as Low when they were actually High and 268 samples as High when they were actually Low. The overall misclassification rate for the pruned classification tree on the entire dataset was approximately 0.123, or 12.3%.

In conclusion, the classification tree models demonstrated the potential to predict the CO2 emission class based on the provided features of the vehicles. The pruned tree, with its simplified structure, achieved a misclassification rate of around 12.3% on both the test set and the entire dataset. Further improvements to the model's accuracy could be explored through feature engineering, ensemble methods, or other advanced techniques.

2.4 Multinomial regression

We employ multinomial regression methodology to re-exam the CO2 emission using CO2 Rating as the responder, which has 10 level of classes, instead of CO2 Class, which has 2 level of classes. Both CO2 rating and CO2 Class were classified based on the CO2 emission data but binning differently. CO2 rating is based on evenly binning on the original CO2 emmission data. WE hope to see if refine binning will increase the accuracy of the prediction and to provide any threshold value of some major contributors

A mulinomail cumulative logit regression model is suitable for this scenario to handle a multi-class classification problems. Since we have a set of ordered class and also we would like to know the cumulative

probability, we selected cumulative logit regression model. Here, a linear relationship between the predictors and the log-odds of the class probabilities are expected and assumed.

Step 1.Selet the best-fit cumulative logit model

First, the predictors are normalized based on their mean and standard deviation to a similar scale to facilitate the multinomial regression with better convergent.

```
# Load the VGAM package
library(VGAM)
library(caret)
set.seed(2023)

#read the data
mydata = read.csv("merged_data_full_6.csv")
mynewdata <- mydata[, c("Year_of_model", "Engine_Size", "Cylinders", "CO2_rating","Weight_upto_kg","Auto

# Standardize numerical predictors
means <- colMeans(mynewdata[, c("Year_of_model", "Engine_Size", "Cylinders","Weight_upto_kg")])
sds <- apply(mynewdata[, c("Year_of_model", "Engine_Size", "Cylinders","Weight_upto_kg")], 2, sd)
mynewdata_scaled <- mynewdata
mynewdata_scaled[, c("Year_of_model" , "Engine_Size" , "Cylinders", "Weight_upto_kg")] <- scale(mynewda

# Ordered the responder
mynewdata_scaled$CO2_rating <- ordered(mynewdata_scaled$CO2_rating, levels = 1:10)

# Split the data into train and test sets
trainData <- mynewdata_scaled[idx$ID_unit, ]
testData <- mynewdata_scaled[-idx$ID_unit, ]
#head(train,n=4)
#head(trainData,n=4)
```

```
# Fit the cumulative regression model
mn_model <- vglm(CO2_rating ~ Year_of_model + Engine_Size + Cylinders + Weight_upto_kg + factor(Auto_tra
summary(mn_model)
```

```
##
## Call:
## vglm(formula = CO2_rating ~ Year_of_model + Engine_Size + Cylinders +
##     Weight_upto_kg + factor(Auto_trans) + factor(Fuel), family = cumulative(parallel = TRUE),
##     data = trainData)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept):1       -9.23979    0.23432 -39.433  < 2e-16 ***
## (Intercept):2       -6.59551    0.19545 -33.746  < 2e-16 ***
## (Intercept):3       -3.31454    0.17249 -19.216  < 2e-16 ***
## (Intercept):4       -0.59592    0.16563  -3.598 0.000321 ***
## (Intercept):5        1.88167    0.16696  11.270  < 2e-16 ***
## (Intercept):6        3.16854    0.16930  18.716  < 2e-16 ***
## (Intercept):7        4.61981    0.17831  25.908  < 2e-16 ***
## (Intercept):8        5.56076    0.19498  28.520  < 2e-16 ***
## (Intercept):9        6.62811    0.23940  27.686  < 2e-16 ***
## Year_of_model        0.30603    0.02511  12.187  < 2e-16 ***
```

```
## Engine_Size          1.47211    0.07435  19.800  < 2e-16 ***
## Cylinders            1.51977    0.07186  21.150  < 2e-16 ***
## Weight_upto_kg       0.51255    0.02985  17.171  < 2e-16 ***
## factor(Auto_trans)1 -0.22442    0.07505  -2.990 0.002788 **
## factor(Fuel)2        0.99598    0.15210   6.548 5.82e-11 ***
## factor(Fuel)3        1.13956    0.21477   5.306 1.12e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  9
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]),
## logitlink(P[Y<=3]), logitlink(P[Y<=4]), logitlink(P[Y<=5]), logitlink(P[Y<=6]),
## logitlink(P[Y<=7]), logitlink(P[Y<=8]), logitlink(P[Y<=9])
##
## Residual deviance: 15125.38 on 54299 degrees of freedom
##
## Log-likelihood: -7562.688 on 54299 degrees of freedom
##
## Number of Fisher scoring iterations: 8
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):9'
##
##
## Exponentiated coefficients:
##       Year_of_model         Engine_Size           Cylinders      Weight_upto_kg
##           1.3580174           4.3584077           4.5711557           1.6695454
## factor(Auto_trans)1       factor(Fuel)2       factor(Fuel)3
##           0.7989783           2.7073888           3.1253793
```

From the cumulative logit model summary result, we can draw several conclusions:

a. In the model summery, the null hypothesis is that the predictor has no effect on the response variable (CO2_rating), and the alternative hypothesis is that there is a significant effect. The p-value represents the probability of observing a test statistic as extreme as the one calculated, assuming the null hypothesis is true.

All of the predictor variables (Year_of_model, Engine_Size, Cylinders, Weight_upto_kg, Auto_trans, and Fuel) have extremely low p-values (<=0.01), indicating strong evidence against the null hypothesis. Therefore, we conclude that all of these variables are significantly associated with the response variable (CO2_rating).

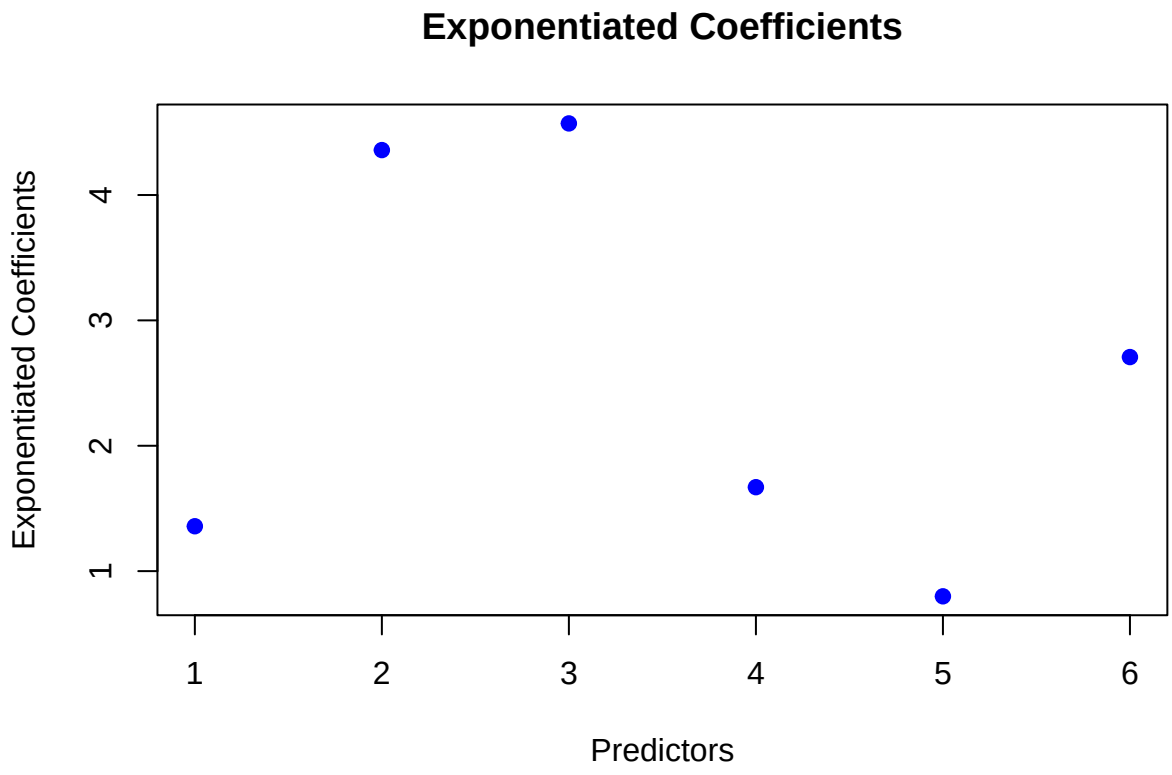
b. Model fit: The residual deviance and log-likelihood provide information about the goodness-of-fit of the model. A lower residual deviance indicates a better fit to the data. The log-likelihood represents the maximum value of the likelihood function, and a higher value indicates a better fit. However, without additional context or comparison with alternative models, it is challenging to make a definitive assessment of model fit.We will conduct further check in Step 2.

c. Hauck-Donner effect: Unfortunately Hauck-Donner effect was detected in '(Intercept 9), which suggests that the estimated coefficients are either highly sensitive to the order of the data, or data has some strong correlation /separation. This is not a desirable property. Since the result was not ensuring the stability of the coefficient estimates, these problematic intercept need to be further deal with.

However, we tried a few things either re-started the fit from a desire value for intercept 9 or re-grouping category. Couldn't get it resolved. We didn't see a strong separation in the predictors by previous sections' plotting. Therefore, we leave the warning as it is.

Then, the next question you may ask is, which predictor/predictors have largest contribution for CO2-rating. We plotted the Exponentiated coefficients as follow to present it.

```
# Extract the coefficients from the model
coefficients <- coef(mn_model)
predictor_indices1 <- c(10, 11, 12, 13, 14,15)
predictor_coefs <- coefficients[predictor_indices1]
exp_coefs <- exp(predictor_coefs)

plot(exp_coefs, type = "p", pch = 19, col = "blue", xlab = "Predictors", ylab = "Exponentiated Coefficie
```

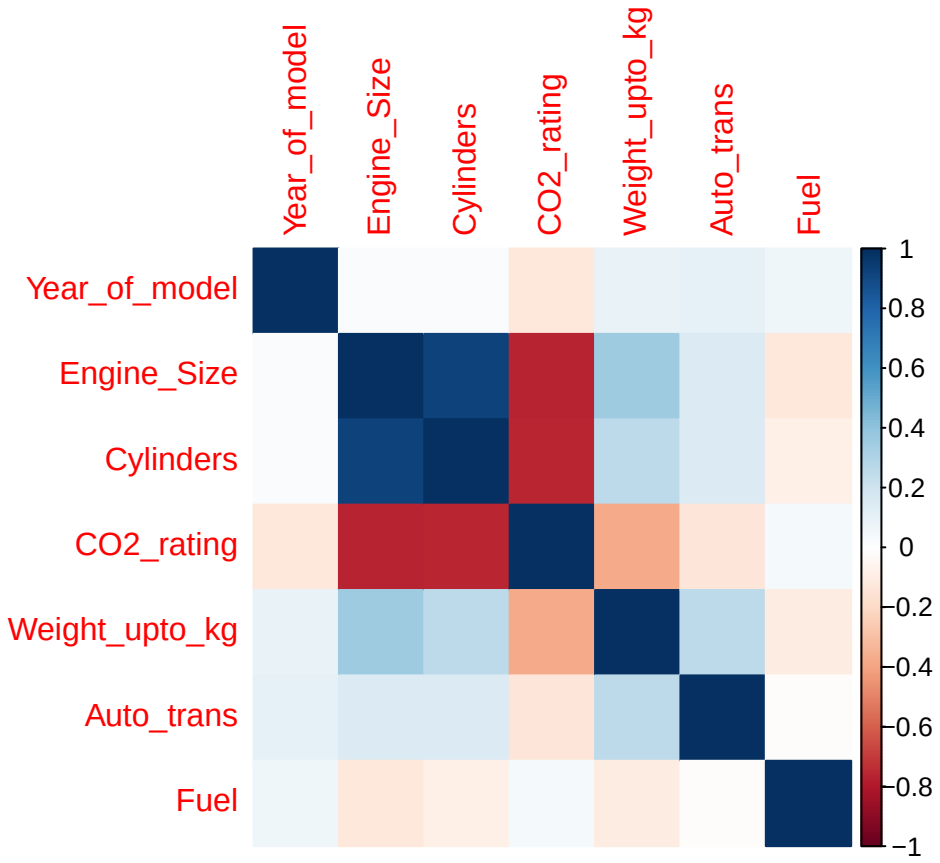


As expected, the exponentiated coefficients vs Predictors plot indicated ngine size(2) and Cylinder size(3) are the strongest contributors for the CO2 emission rating. A further Heatmap plot confirmed their strong correlation with the CO2-rating. Detail as follow:

```
mydata_subset <- mydata[, c("Year_of_model", "Engine_Size", "Cylinders", "CO2_rating","Weight_upto_kg")]

# Calculate the correlation matrix
cor_matrix <- cor(mynewdata)

# Plot a correlation matrix heatmap
#install.packages("corrplot", repo="https://github.com/taiyun/corrplot")
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor_matrix, method = "color")
```



Step 2. Check Good-of-fit of the multinomial cumulative regression model

Here the null hypothesis is defined as:

$$H_0 : \text{the cumulative logit model fits the observations}$$

```
# Check the good-of-fit of the model
1-pchisq(deviance(mn_model),df.residual(mn_model))
```

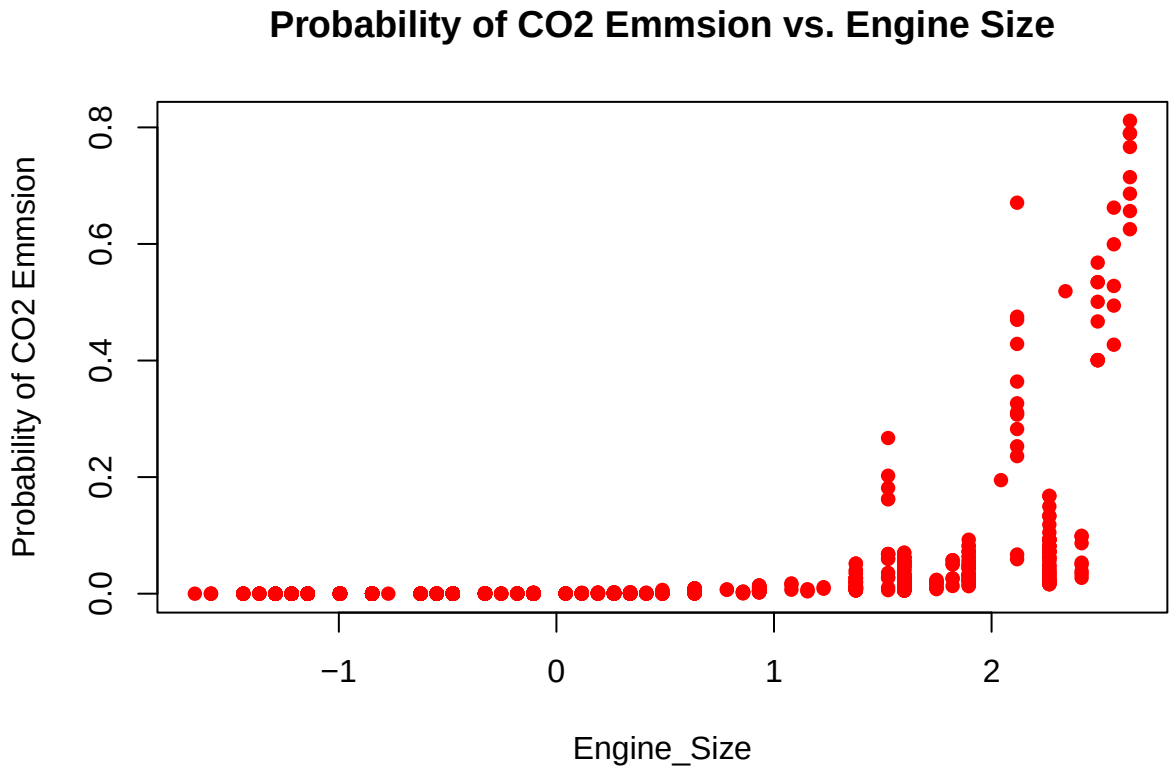
```
## [1] 1
```

The p-value here is so large, so we cannot reject H0. therefore, the cumulative model provided a perfect fit. However this didn't mean the model will provide high accuracy prediction.

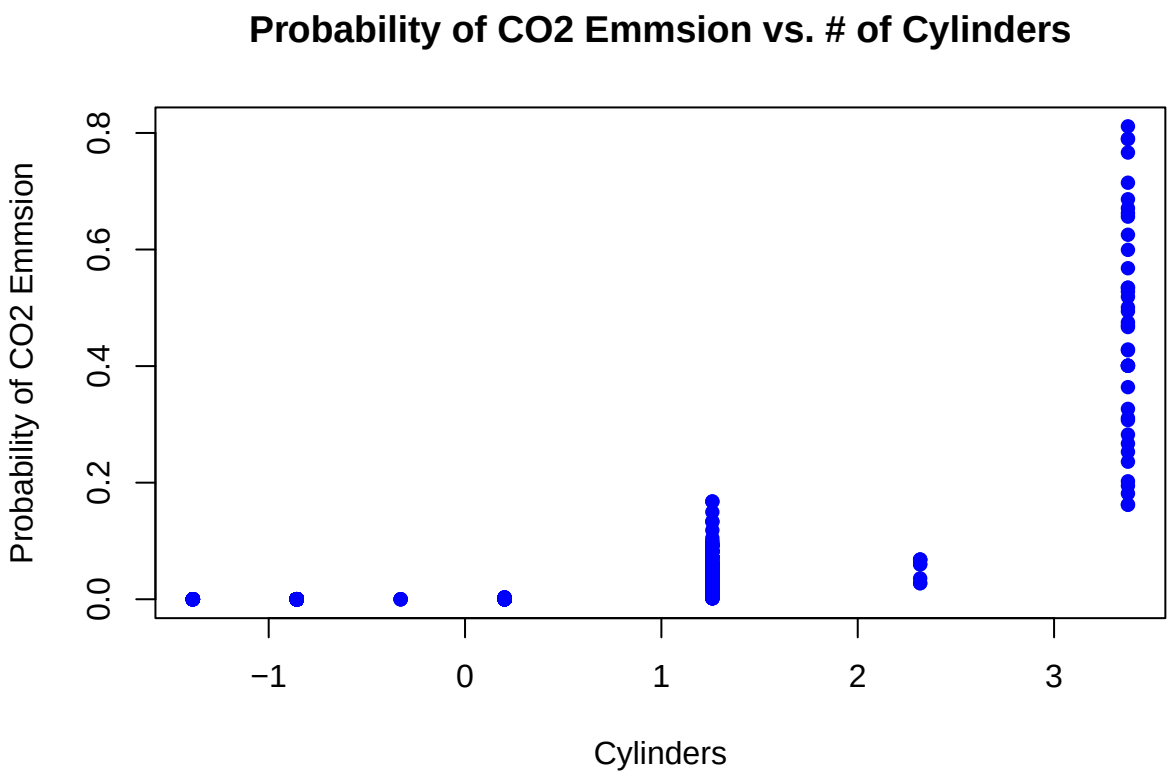
Step 3. Model prediction

```
# Generate prediction
mn_pred<-predict(mn_model,testData,type="response")

# Extract engine size and predicted probabilities
Engine_Size <- testData$Engine_Size
Cylinders <- testData$Cylinders
```

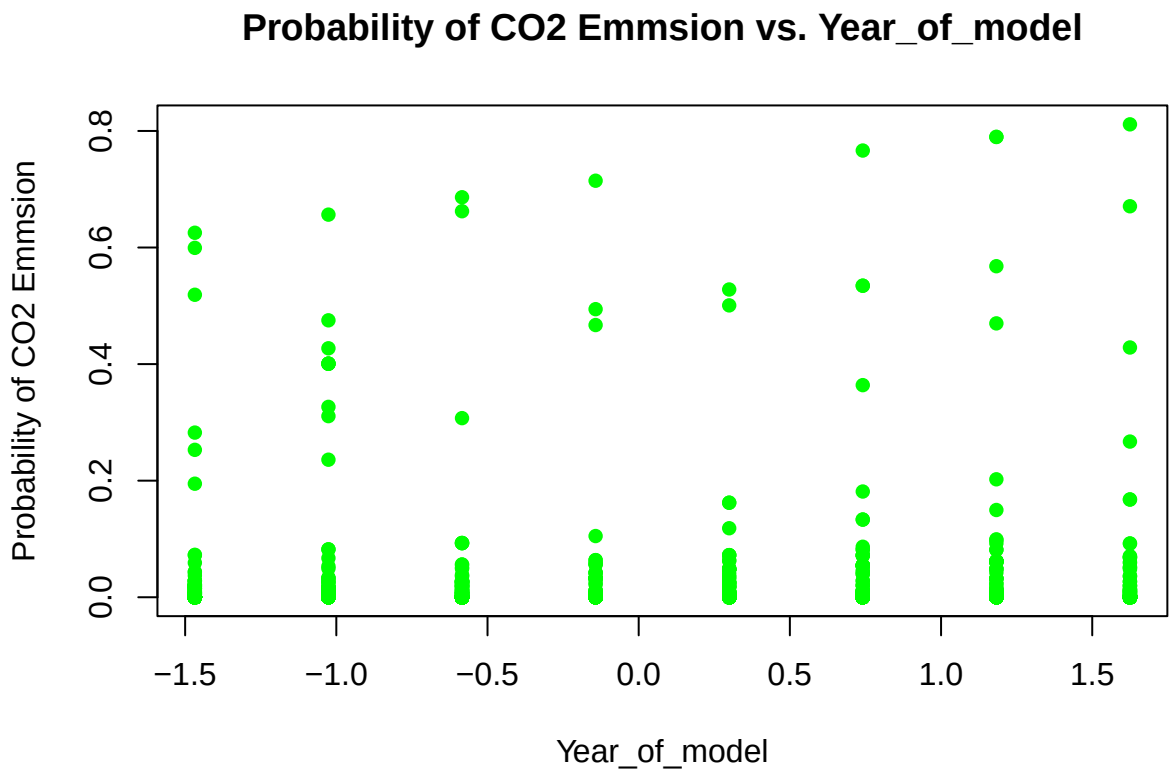
```r
Year_of_model <- testData$Year_of_model
Weight_upto_kg <- testData$Weight_upto_kg
prob_CO2 <- mn_pred[, 1]

#par(mfrow = c(2, 2))

# Create a scatter plot
plot(Engine_Size, prob_CO2, type = "p", pch = 16, col = "red",
     xlab = "Engine_Size", ylab = "Probability of CO2 Emmsion",
     main = "Probability of CO2 Emmsion vs. Engine Size")
```



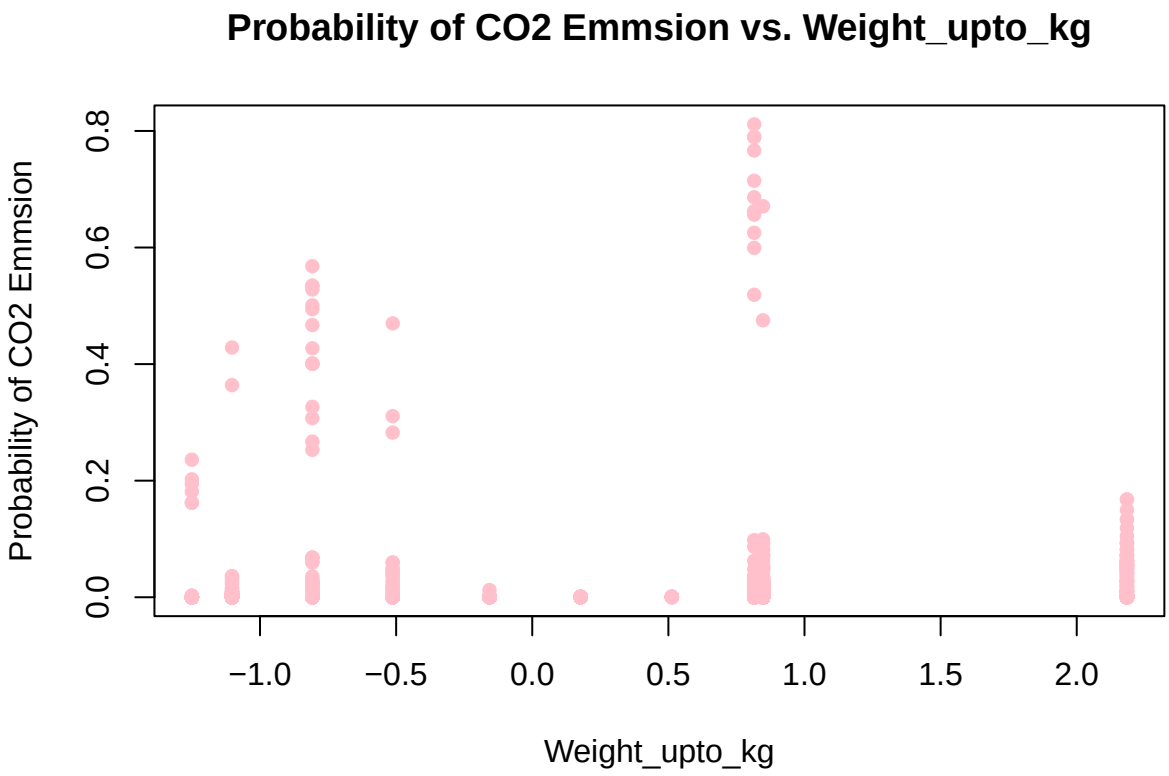

```r
# Create a scatter plot
plot(Cylinders, prob_CO2, type = "p", pch = 16, col = "blue",
     xlab = "Cylinders", ylab = "Probability of CO2 Emmsion",
     main = "Probability of CO2 Emmsion vs. # of Cylinders")
```




```r
# Create a scatter plot
plot(Year_of_model, prob_CO2, type = "p", pch = 16, col = "green",
     xlab = "Year_of_model", ylab = "Probability of CO2 Emmsion",
     main = "Probability of CO2 Emmsion vs. Year_of_model")
```




```
# Create a scatter plot
plot(Weight_upto_kg, prob_CO2, type = "p", pch = 16, col = "pink",
     xlab = "Weight_upto_kg", ylab = "Probability of CO2 Emmsion",
     main = "Probability of CO2 Emmsion vs. Weight_upto_kg")
```



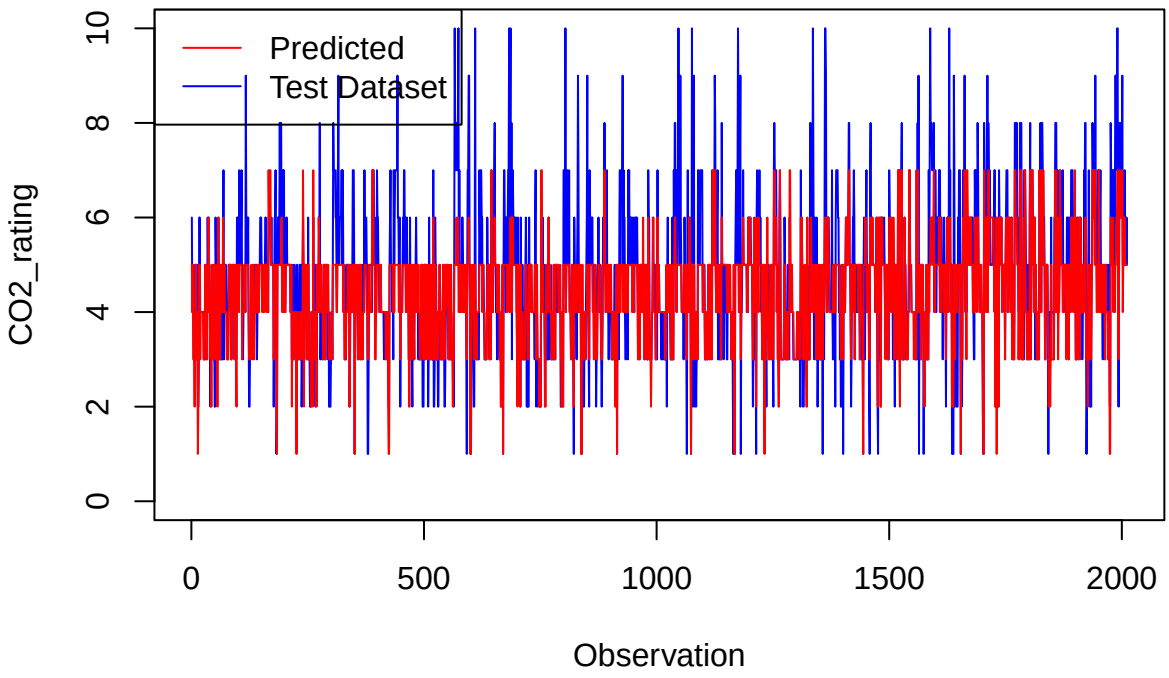
The CO2 emission probability plots clearly showed that, the probability of CO2 emission increase not only exponentially at a threshold number with Engine Size and Cylinder size, but also are notably effected by year of model and weight of the vehicle (weight_upto_kg) in a minor degree.

Step 4. Test the model's prediction

```
fitted.result<-colnames(mn_pred)[rowMaxs(mn_pred)]
misClasificError <- round(mean(fitted.result != testData$CO2_rating),3)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.482"
```

```
# Create an empty plot

x <- seq_along(testData$CO2_rating)
plot(x, testData$CO2_rating, type = "l", col = "blue", ylim = c(0, 10), ylab = "CO2_rating", xlab = "Ob
x1 <- seq_along(fitted.result)
# Add the predicted values
lines(x1, fitted.result, col = "red")

# Add a legend
legend("topleft", legend = c("Predicted", "Test Dataset"), col = c("red", "blue"), lty = 1)
```



The overlap plot of actual CO2 rating vs predicted CO2 rating indicated the multinomial regression model over-estimated the CO2 emission for High emission class (or 1-5 CO2 rating) and significantly udder-estimated the CO2 emmission for Low emission class (or 6-10 CO2 rating).

3 Conclusion

Based on the analysis of different models and their performance in predicting the CO2 emission class, the following conclusions can be drawn:

- Logistic regression: The logistic regression model achieved an accuracy of approximately 87.3% and a Kappa statistic of 0.745, indicating a good fit to the data. Significant predictors in this model include Year_of_model, Engine_Size, Weight_upto_kg, Auto_trans, and Fuel.
- Linear Discriminant Analysis (LDA): The LDA model achieved an accuracy of around 85.5% and a Kappa statistic of 0.709, suggesting a reasonable fit. The significant predictors in this model are Year_of_model, Engine_Size, Weight_upto_kg, Auto_trans, and Fuel.
- Quadratic Discriminant Analysis (QDA): The QDA model achieved an accuracy of approximately 83.3% and a Kappa statistic of 0.666, indicating a fair fit. The significant predictors in this model are Year_of_model, Engine_Size, Weight_upto_kg, Auto_trans, and Fuel.

Assumptions of multivariate normality and equal variance were not met for the LDA and QDA models. This indicates that the variables in the dataset do not follow a multivariate distribution within each class of CO2_class, and the variances of the predictor variables are not equal across different levels of CO2_class. These violations should be considered when interpreting the results and assessing the performance of the LDA and QDA models.

- Classification tree: The classification tree model demonstrated superior performance compared to the logistic regression, LDA, and QDA models, with an accuracy of 87.7%. The key predictor in this model is Engine_Size, which effectively partitions the predictor space.
- Multinomial regression: Unlike the other models, which used the CO2 emissions class (High/Low) as the response variable, the multinomial regression model employed the CO2 emissions rating as the response variable. The CO2 emissions rating is a scale ranging from 1 (worst) to 10 (best). This distinction in the response variable should be taken into account when comparing the results of the multinomial regression model with the other models. The multinomial regression model showed an accuracy of 48.2%. TSignificant predictors in this model include Year_of_model, Engine_Size, Cylinders, Weight_upto_kg, Auto_trans, and Fuel. However, the model exhibited the Hauck-Donner effect, which suggests potential issues with coefficient stability.

In summary, the classification tree model emerged as the best-performing model in terms of accuracy. The significant predictors across the different models generally include Year_of_model, Engine_Size, Weight_upto_kg, Auto_trans, and Fuel. These findings can provide insights into the relationship between vehicle characteristics and CO2 emissions, aiding in the development of more accurate prediction models and the promotion of sustainable practices in the transportation sector.

References

[1] Government of Canada. (2023). Fuel consumption ratings. Available at: https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64 [Accessed 29 May 2023]. Contains information licensed under the Open Government Licence – Canada.