baj2wz8tj

March 8, 2025

[1]:
```
# THIS IS AN INTELECTUAL PROPERTY OF ALEKSAS SLAVINSKAS

#importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np


#importing the dataset
df1 = pd.read_csv(r'/content/Updated_Vehicle_Dataset.csv')

#Reading the dataset
print(str(df1))
# and datatypes
print(df1.dtypes)
#dispalying all columns
pd.set_option('display.max_columns', None)
print(df1.head())


# Defininining the German tax calculation function

# Ensure column names are correctly formatted
df1.columns = df1.columns.str.strip()


# Define the tax calculation function
def calculate_total_tax(row):
    try:
        co2_emissions = float(row['CO2 Emissions(g/km)'])
        engine_size = float(row['Engine Size(L)'])
        fuel_type = str(row['Fuel Type']).lower()

        # CO -based tax
```

```
        co2_tax = max(0, co2_emissions - 95) * 2

        # Engine size-based tax
        if 'D' in fuel_type:
            engine_tax = (engine_size * 1000 / 100) * 9.50  # Diesel cars
        else:
            engine_tax = (engine_size * 1000 / 100) * 2  # other cars, all are␣
 ↪calculated on same formula as all use ignition-based fuel (petrol,␣
 ↪flex-fuel, ethanol, nat-gas)

        # Total tax
        return co2_tax + engine_tax
    except Exception as e:
        return None  # Return None for problematic rows

# Applying the function to calculate total tax
df1['Total Tax (€)'] = df1.apply(calculate_total_tax, axis=1)

df1.to_csv("Updated_Vehicle_Dataset2.csv", index=False)
print(df1.head())

df = pd.read_csv(r'/content/Updated_Vehicle_Dataset2.csv')

def visualizations1():
    # 1. Number of vehicles per segment
    segment_counts = df['Segment'].value_counts()
    plt.figure(figsize=(10, 5))
    sns.barplot(x=segment_counts.index, y=segment_counts.values)
    plt.title("Number of Vehicles per Segment")
    plt.xlabel("Segment")
    plt.ylabel("Count")
    plt.xticks(rotation=45)
    plt.show()

    # 2. Distribution of vehicle prices
    plt.figure(figsize=(10, 5))
    sns.histplot(df['Average Price'], bins=30, kde=True)
    plt.title("Distribution of Average Vehicle Prices")
    plt.xlabel("Average Price")
    plt.ylabel("Frequency")
    plt.show()

    # 3. Fuel consumption per fuel type
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Fuel Type'], y=df['Fuel Consumption Comb (L/100 km)'])
    plt.title("Fuel Consumption per Fuel Type")
    plt.xlabel("Fuel Type")
```

```
    plt.ylabel("Fuel Consumption (L/100km)")
    plt.show()

    # 4. CO2 Emissions per fuel type
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Fuel Type'], y=df['CO2 Emissions(g/km)'])
    plt.title("CO2 Emissions per Fuel Type")
    plt.xlabel("Fuel Type")
    plt.ylabel("CO2 Emissions (g/km)")
    plt.show()

    # 4.5 Total tax per fuel type
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Fuel Type'], y=df['Total Tax (€)'])
    plt.title("Total Tax (EUR) per Fuel Type")
    plt.xlabel("Fuel Type")
    plt.ylabel("Total Tax (EUR)")
    plt.show()

    # 5. Various pollution-related metrics per segment
    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Segment'], y=df['CO2 Emissions(g/km)'])
    plt.title("CO2 Emissions per Segment")
    plt.xlabel("Segment")
    plt.ylabel("CO2 Emissions (g/km)")
    plt.xticks(rotation=45)
    plt.show()

    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Segment'], y=df['Fuel Consumption Comb (L/100 km)'])
    plt.title("Fuel Consumption per Segment")
    plt.xlabel("Segment")
    plt.ylabel("Fuel Consumption (L/100km)")
    plt.xticks(rotation=45)
    plt.show()

    plt.figure(figsize=(10, 5))
    sns.boxplot(x=df['Segment'], y=df['Total Tax (€)'])
    plt.title("Total Pollution Tax per Segment")
    plt.xlabel("Segment")
    plt.ylabel("Total Tax (€)")
    plt.xticks(rotation=45)
    plt.show()

visualizations1() # we put visualizations as functions so that 8 visualizations␣
 ↪do not pop-up when I am running other bits of code, it just gets annoying to␣
 ↪shut them off every time
```

```
# Exploratory Data Analysis (EDA) with Additional Visualizations

def visualizations2():
    # 1. Correlation Heatmap
    plt.figure(figsize=(10, 6))
    sns.heatmap(df[['CO2 Emissions(g/km)', 'Fuel Consumption Comb (L/100 km)',␣
 ↪'Engine Size(L)', 'Average Price', 'Total Tax (€)']].corr(), annot=True,␣
 ↪cmap='coolwarm', linewidths=0.5)
    plt.title("Correlation Heatmap of Key Variables")
    plt.show()

    # 2. Scatter Plot: Price vs. CO Emissions
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=df['Average Price'], y=df['CO2 Emissions(g/km)'],␣
 ↪hue=df['Fuel Type'], alpha=0.7)
    plt.title("Vehicle Price vs CO2 Emissions")
    plt.xlabel("Price (€)")
    plt.ylabel("CO2 Emissions (g/km)")
    plt.legend(title="Fuel Type")
    plt.show()


    # 3. Scatter Plot: Engine Size vs. CO Emissions
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=df['Engine Size(L)'], y=df['CO2 Emissions(g/km)'],␣
 ↪hue=df['Fuel Type'], alpha=0.7)
    plt.title("Engine Size vs CO2 Emissions")
    plt.xlabel("Engine Size (L)")
    plt.ylabel("CO2 Emissions (g/km)")
    plt.legend(title="Fuel Type")
    plt.show()

    # 5. Average Pollution Intensity per Segment
    plt.figure(figsize=(10, 6))
    segment_pollution = df.groupby('Segment')['CO2 Emissions(g/km)'].mean().
 ↪sort_values()
    sns.barplot(x=segment_pollution.index, y=segment_pollution.values)
    plt.title("Average CO2 Emissions per Segment")
    plt.xlabel("Segment")
    plt.ylabel("Average CO2 Emissions (g/km)")
    plt.xticks(rotation=45)
    plt.show()

    # 6. Fuel Consumption vs. Total Tax
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=df['CO2 Emissions(g/km)'], y=df['Total Tax (€)'],␣
 ↪hue=df['Fuel Type'], alpha=0.7)
```

```
    plt.title("CO2 emissions vs Total Tax")
    plt.xlabel("CO2 Emissions(g/km)")
    plt.ylabel("Total Tax (€)")
    plt.legend(title="Fuel Type")
    plt.show()

visualizations2() # as before, I put this as a function so that I can shut it␣
 ↪off and run other bits of code, otherwise its annoying

# 7. Calculating Linear Regression: CO2 Emissions vs Total Tax
X = df[['CO2 Emissions(g/km)']].values  # Feature (CO2 Emissions)
y = df['Total Tax (€)'].values  # Target (Total Tax)

# Train the model
model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)

# Calculate metrics
r2 = r2_score(y, y_pred)
mse = mean_squared_error(y, y_pred)
print(f"R² Score: {r2:.4f}")
print(f"Mean Squared Error: {mse:.2f}")
print(f"Regression Coefficient (Slope): {model.coef_[0]:.2f}")
print(f"Intercept: {model.intercept_:.2f}")

# 8. Displaying the Scatter Plot with Regression Line on the Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['CO2 Emissions(g/km)'], y=df['Total Tax (€)'], alpha=0.6,␣
 ↪label="Actual Data")
plt.plot(df['CO2 Emissions(g/km)'], y_pred, color='red', label="Regression␣
 ↪Line")
plt.title("CO2 Emissions vs Total Tax (Linear Regression)")
plt.xlabel("CO2 Emissions (g/km)")
plt.ylabel("Total Tax (€)")
plt.legend()
plt.show()

# fitting the columns with the new coefficient, finding the new formula for a␣
 ↪more fair approach
# Recalculating tax using only CO2 emissions with the new coefficient (2.39)
df['Recalculated Tax (€)'] = df['CO2 Emissions(g/km)'] * 2.39

# Calculate total tax collected under both old and new formulas
total_old_tax = df['Total Tax (€)'].sum()
total_new_tax = df['Recalculated Tax (€)'].sum()
```

```
# Determinining adjustment factor so that we can match total tax collected to the original
adjustment_factor = total_old_tax / total_new_tax

df['Adjusted Tax (€)'] = df['Recalculated Tax (€)'] * adjustment_factor

# Print the new tax formula without engine size
print("\nNew Adjusted Tax Formula:")
print(f"Tax (€) = {2.39 * adjustment_factor:.2f} * CO2 Emissions (g/km)")

# Calculating Linear Regression, this time: CO2 Emissions vs Adjusted Tax
X_new = df[['CO2 Emissions(g/km)']].values  # Feature (CO2 Emissions)
y_new = df['Adjusted Tax (€)'].values  # Target (Adjusted Tax)

#logic of the code and calculation was done with AI, as in I could not figure out how find adjustemnt factor and that it will be needed (OPENAI, 2025)

# Train the model
model_new = LinearRegression()
model_new.fit(X_new, y_new)
y_new_pred = model_new.predict(X_new)

# Calculating new regression metrics
r2_new = r2_score(y_new, y_new_pred)
mse_new = mean_squared_error(y_new, y_new_pred)
print(f"\nNew Model Evaluation:")
print(f"R² Score (New Formula): {r2_new:.4f}")
print(f"Mean Squared Error (New Formula): {mse_new:.2f}")
print(f"Regression Coefficient (New Formula): {model_new.coef_[0]:.2f}")
print(f"Intercept (New Formula): {model_new.intercept_:.2f}")

# Plot regression results
plt.figure(figsize=(10, 6))
sns.scatterplot(x=df['CO2 Emissions(g/km)'], y=df['Adjusted Tax (€)'], alpha=0.6, label="Actual Data")
plt.plot(df['CO2 Emissions(g/km)'], y_new_pred, color='red', label="Regression Line")
plt.title("CO2 Emissions vs Adjusted Tax (Linear Regression)")
plt.xlabel("CO2 Emissions (g/km)")
plt.ylabel("Adjusted Tax (€)")
plt.legend()
plt.show()

# Calculate tax based on the new formula: Tax (€) = 1.49 * CO  Emissions (g/km)
df['New Formula Tax (€)'] = df['CO2 Emissions(g/km)'] * 1.49

# Compare total tax collected
```

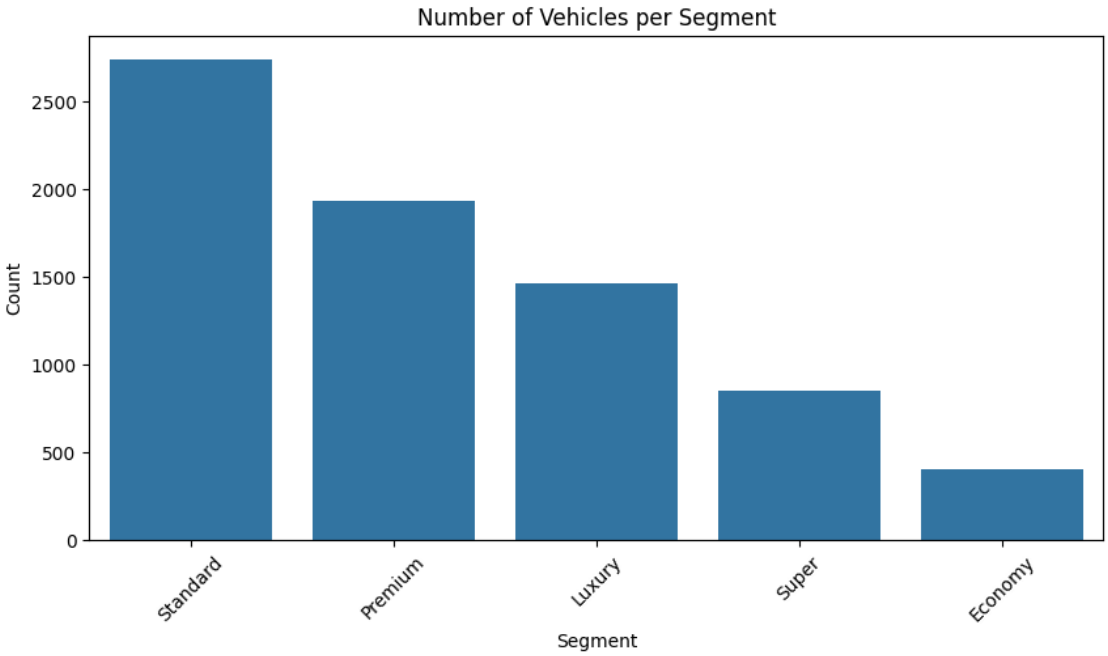
```
total_new_formula_tax = df['New Formula Tax (€)'].sum()

print("old tax", total_old_tax )
print("new(ish) tax", total_new_tax)
print("new formula tax", total_new_formula_tax)
```

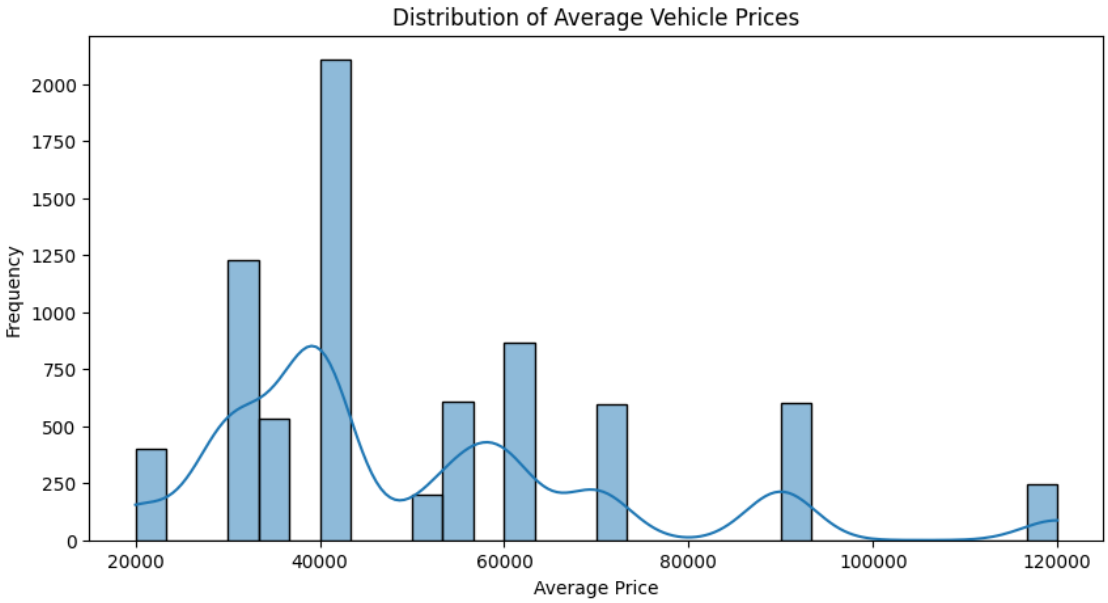
```
       Make         Model   Vehicle Class  Engine Size(L)  Cylinders  \
0     ACURA           ILX         COMPACT             2.0          4
1     ACURA           ILX         COMPACT             2.4          4
2     ACURA   ILX HYBRID         COMPACT             1.5          4
3     ACURA      MDX 4WD     SUV - SMALL             3.5          6
4     ACURA      RDX AWD     SUV - SMALL             3.5          6
…     …           …               …               …         …
7380  VOLVO  XC40 T5 AWD     SUV - SMALL             2.0          4
7381  VOLVO  XC60 T5 AWD     SUV - SMALL             2.0          4
7382  VOLVO  XC60 T6 AWD     SUV - SMALL             2.0          4
7383  VOLVO  XC90 T5 AWD  SUV - STANDARD             2.0          4
7384  VOLVO  XC90 T6 AWD  SUV - STANDARD             2.0          4

     Transmission Fuel Type  Fuel Consumption City (L/100 km)  \
0             AS5         Z                              9.9
1              M6         Z                             11.2
2             AV7         Z                              6.0
3             AS6         Z                             12.7
4             AS6         Z                             12.1
…           …       …                                 …
7380          AS8         Z                             10.7
7381          AS8         Z                             11.2
7382          AS8         Z                             11.7
7383          AS8         Z                             11.2
7384          AS8         Z                             12.2

      Fuel Consumption Hwy (L/100 km)  Fuel Consumption Comb (L/100 km)  \
0                                 6.7                               8.5
1                                 7.7                               9.6
2                                 5.8                               5.9
3                                 9.1                              11.1
4                                 8.7                              10.6
…                                 …                                 …
7380                              7.7                               9.4
7381                              8.3                               9.9
7382                              8.6                              10.3
7383                              8.3                               9.9
7384                              8.7                              10.7

      Fuel Consumption Comb (mpg)  CO2 Emissions(g/km)  Average Price  \
0                              33                  196          30000
```

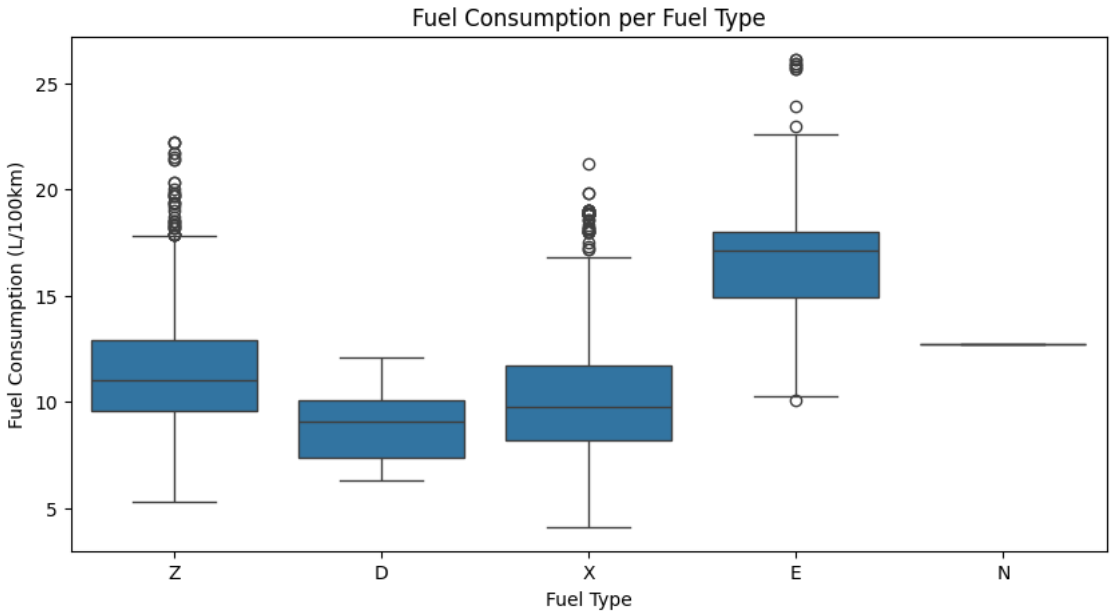
```
1                                29                  221          30000
2                                48                  136          20000
3                                25                  255          60000
4                                27                  244          60000
…                               …                    …            …
7380                             30                  219          40000
7381                             29                  232          40000
7382                             27                  240          40000
7383                             29                  232          40000
7384                             26                  248          40000

       Segment
0     Standard
1     Standard
2      Economy
3       Luxury
4       Luxury
…         …
7380  Standard
7381  Standard
7382  Standard
7383  Standard
7384  Standard

[7385 rows x 14 columns]
Make                               object
Model                              object
Vehicle Class                      object
Engine Size(L)                    float64
Cylinders                           int64
Transmission                       object
Fuel Type                          object
Fuel Consumption City (L/100 km)  float64
Fuel Consumption Hwy (L/100 km)   float64
Fuel Consumption Comb (L/100 km)  float64
Fuel Consumption Comb (mpg)         int64
CO2 Emissions(g/km)                 int64
Average Price                       int64
Segment                            object
dtype: object
    Make       Model Vehicle Class  Engine Size(L)  Cylinders Transmission  \
0  ACURA         ILX       COMPACT             2.0          4          AS5
1  ACURA         ILX       COMPACT             2.4          4           M6
2  ACURA  ILX HYBRID       COMPACT             1.5          4          AV7
3  ACURA     MDX 4WD   SUV - SMALL             3.5          6          AS6
4  ACURA     RDX AWD   SUV - SMALL             3.5          6          AS6

  Fuel Type  Fuel Consumption City (L/100 km)  \
```

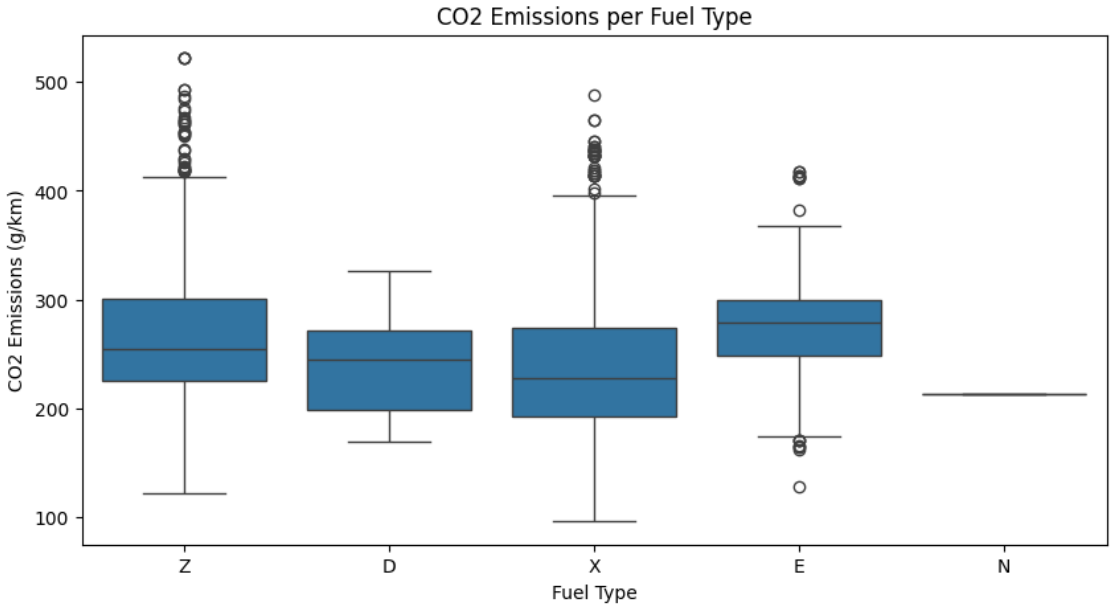
```
0         Z                              9.9
1         Z                             11.2
2         Z                              6.0
3         Z                             12.7
4         Z                             12.1

   Fuel Consumption Hwy (L/100 km)  Fuel Consumption Comb (L/100 km)  \
0                              6.7                               8.5
1                              7.7                               9.6
2                              5.8                               5.9
3                              9.1                              11.1
4                              8.7                              10.6

   Fuel Consumption Comb (mpg)  CO2 Emissions(g/km)  Average Price   Segment
0                           33                  196          30000  Standard
1                           29                  221          30000  Standard
2                           48                  136          20000   Economy
3                           25                  255          60000    Luxury
4                           27                  244          60000    Luxury
    Make       Model Vehicle Class  Engine Size(L)  Cylinders Transmission  \
0  ACURA         ILX       COMPACT             2.0          4           AS5
1  ACURA         ILX       COMPACT             2.4          4            M6
2  ACURA  ILX HYBRID       COMPACT             1.5          4           AV7
3  ACURA     MDX 4WD   SUV - SMALL             3.5          6           AS6
4  ACURA     RDX AWD   SUV - SMALL             3.5          6           AS6

  Fuel Type  Fuel Consumption City (L/100 km)  \
0         Z                              9.9
1         Z                             11.2
2         Z                              6.0
3         Z                             12.7
4         Z                             12.1

   Fuel Consumption Hwy (L/100 km)  Fuel Consumption Comb (L/100 km)  \
0                              6.7                               8.5
1                              7.7                               9.6
2                              5.8                               5.9
3                              9.1                              11.1
4                              8.7                              10.6

   Fuel Consumption Comb (mpg)  CO2 Emissions(g/km)  Average Price   Segment  \
0                           33                  196          30000  Standard
1                           29                  221          30000  Standard
2                           48                  136          20000   Economy
3                           25                  255          60000    Luxury
4                           27                  244          60000    Luxury

   Total Tax (€)
```

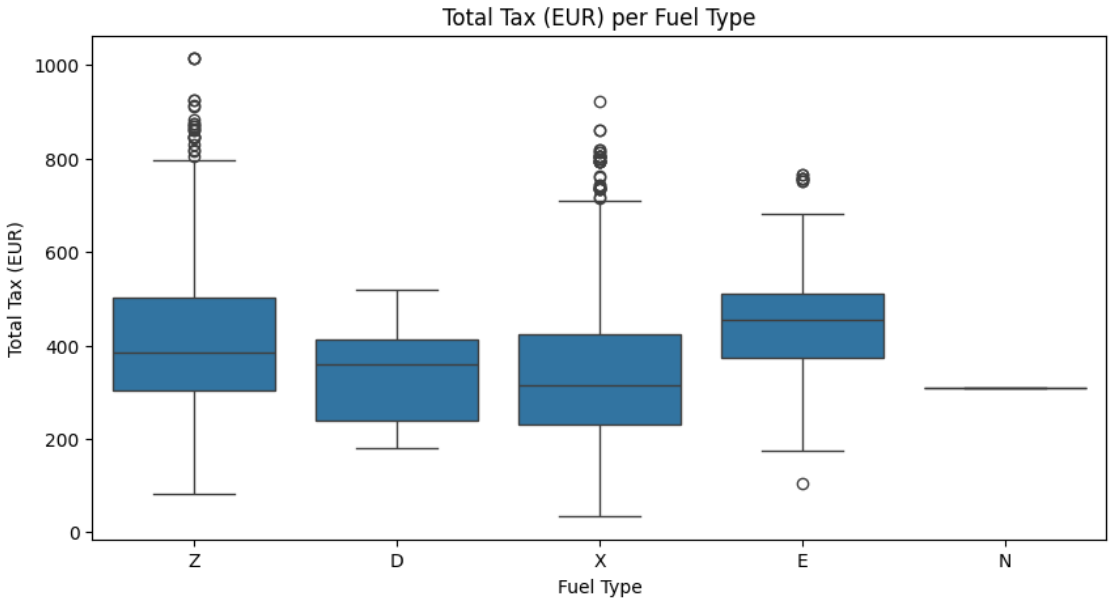
```
0          242.0
1          300.0
2          112.0
3          390.0
4          368.0
```

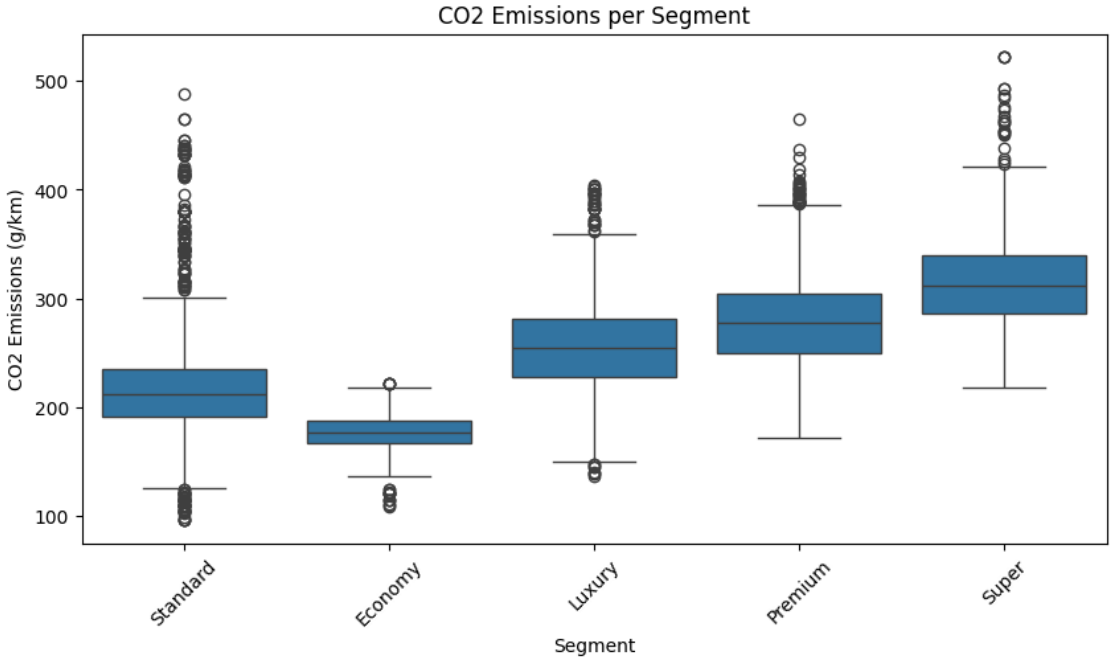


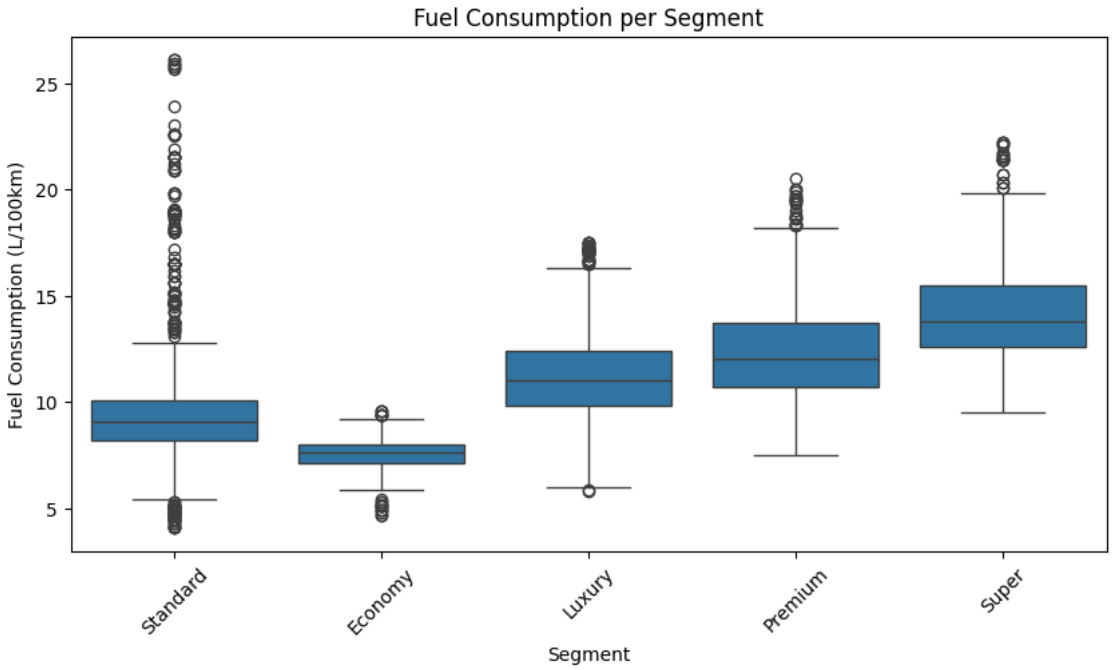


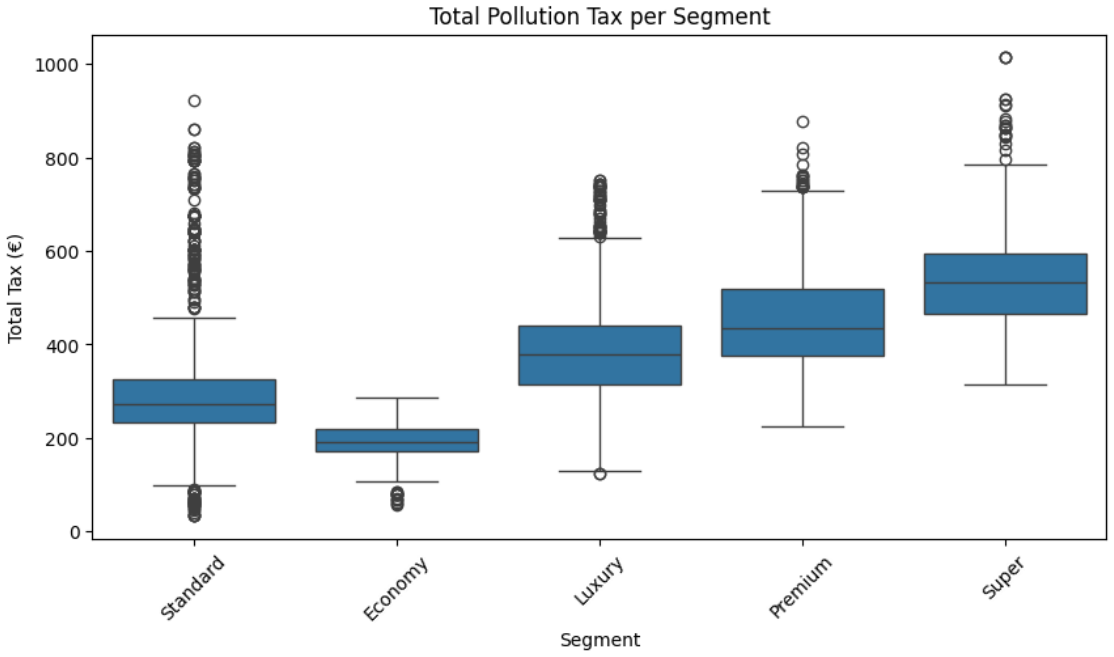

Fuel Consumption per Fuel Type
Fuel Consumption (L/100km)
25
20
15
10
5
Z
D
X
E
N
Fuel Type

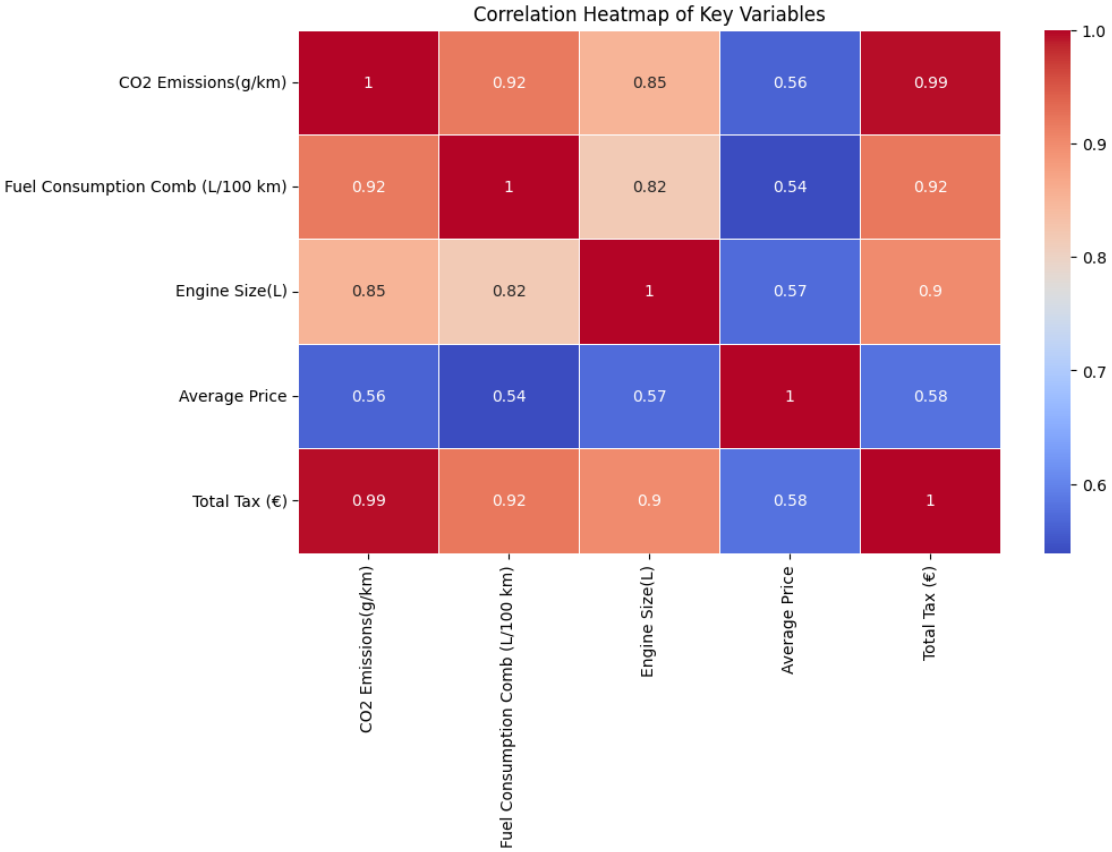

CO2 Emissions per Fuel Type
CO2 Emissions (g/km)
500
400
300
200
100
Z
D
X
E
N
Fuel Type

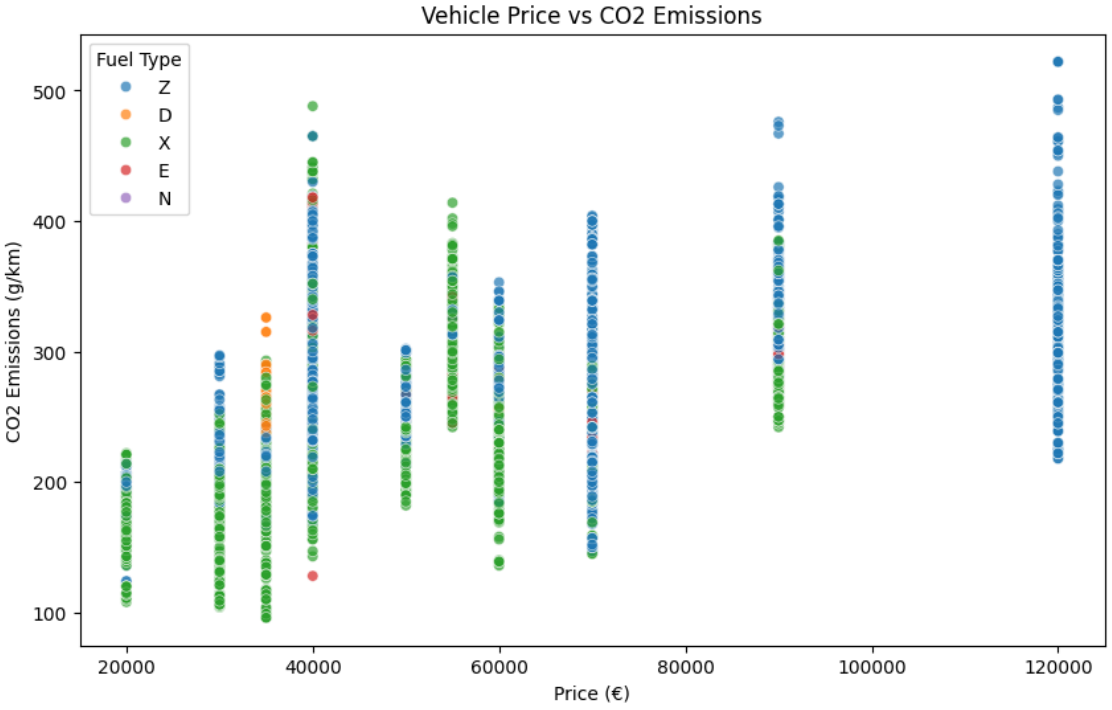

Total Tax (EUR) per Fuel Type
Total Tax (EUR)
0
200
400
600
800
1000
Z
D
X
E
N
Fuel Type

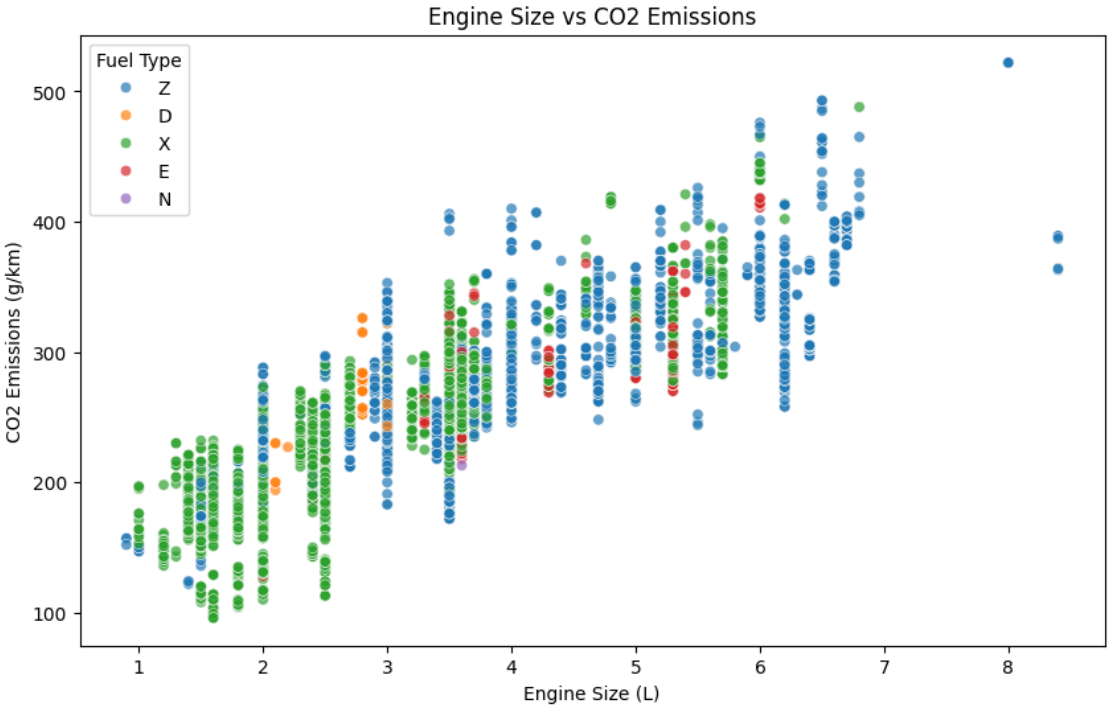

CO2 Emissions per Segment
CO2 Emissions (g/km)
100
200
300
400
500
Standard
Economy
Luxury
Premium
Super
Segment

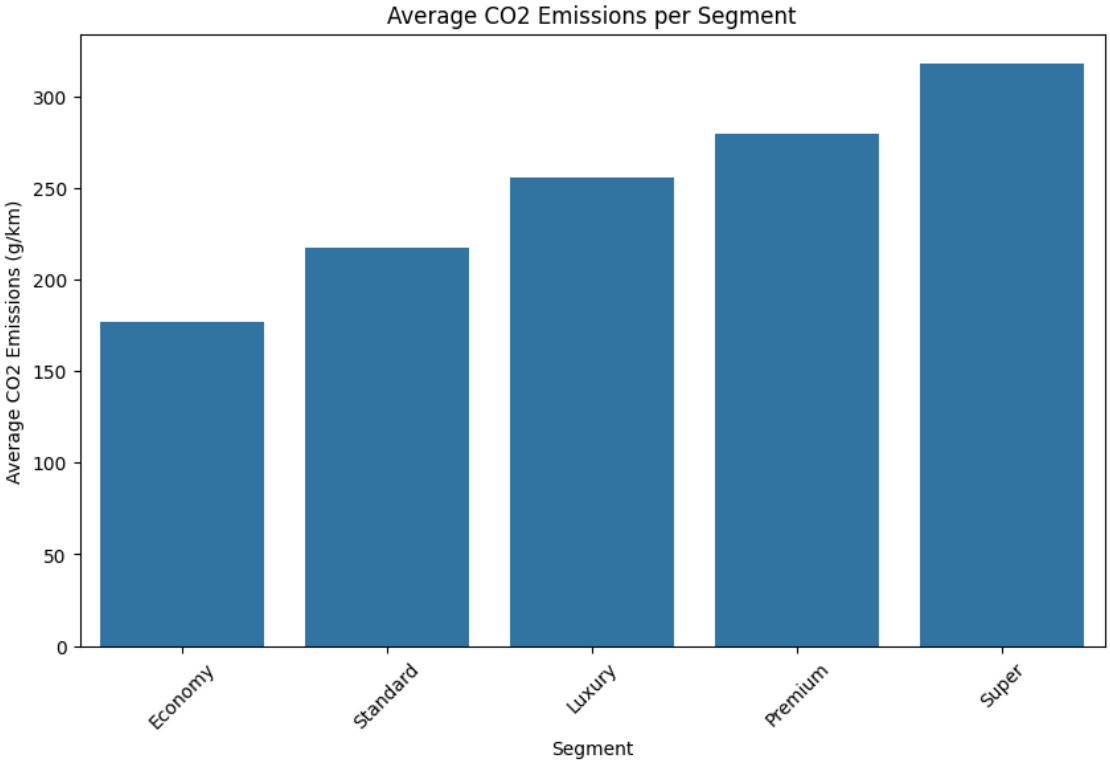

Fuel Consumption per Segment
Fuel Consumption (L/100km)
25
20
15
10
5
Standard
Economy
Luxury
Premium
Super
Segment

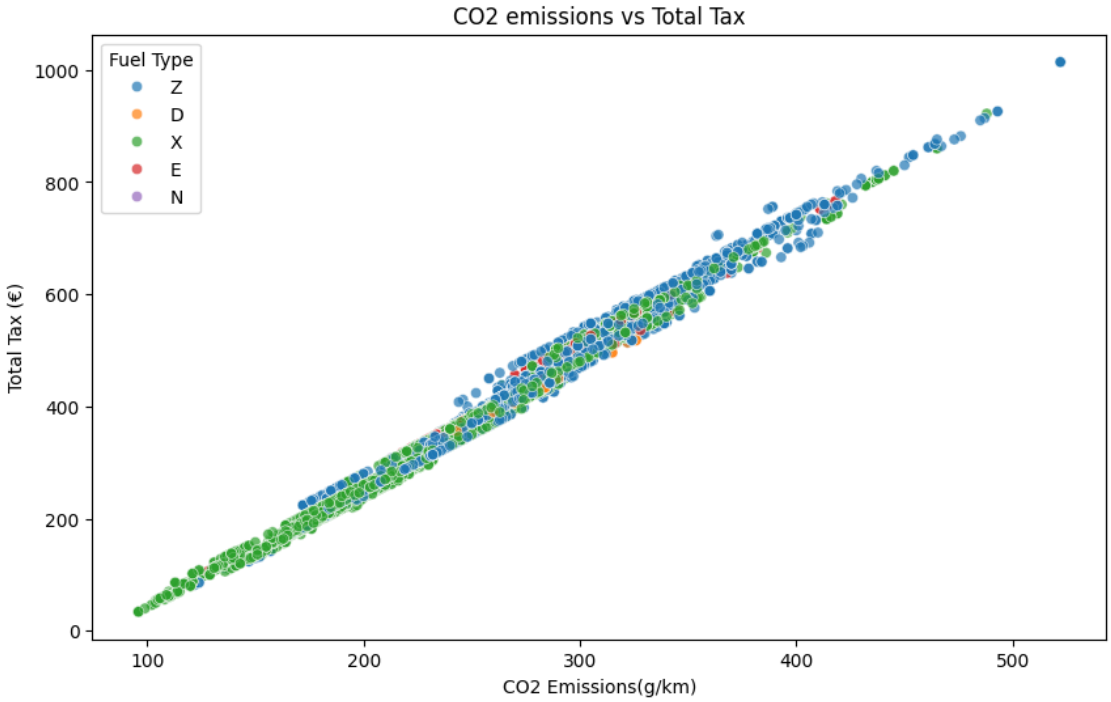

Total Pollution Tax per Segment
Total Tax (€)
1000
800
600
400
200
0
Standard
Economy
Luxury
Premium
Super
Segment


Correlation Heatmap of Key Variables
CO2 Emissions(g/km)
Fuel Consumption Comb (L/100 km)
Engine Size(L)
Average Price
Total Tax (€)
1
0.92
0.85
0.56
0.99
0.92
1
0.82
0.54
0.92
0.85
0.82
1
0.57
0.9
0.56
0.54
0.57
1
0.58
0.99
0.92
0.9
0.58
1
1.0
0.9
0.8
0.7
0.6

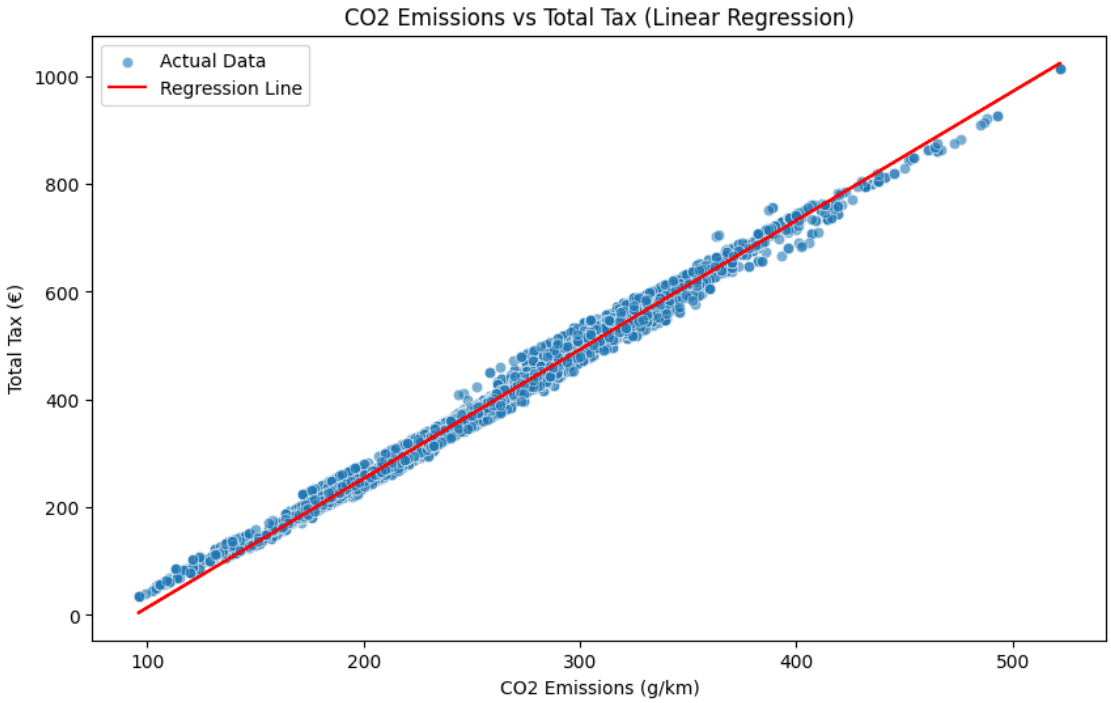

Vehicle Price vs CO2 Emissions
Fuel Type
Z
D
X
E
N
CO2 Emissions (g/km)
100
200
300
400
500
20000
40000
60000
80000
100000
120000
Price (€)


Engine Size vs CO2 Emissions
Fuel Type
Z
D
X
E
N
CO2 Emissions (g/km)
Engine Size (L)
100
200
300
400
500
1
2
3
4
5
6
7
8

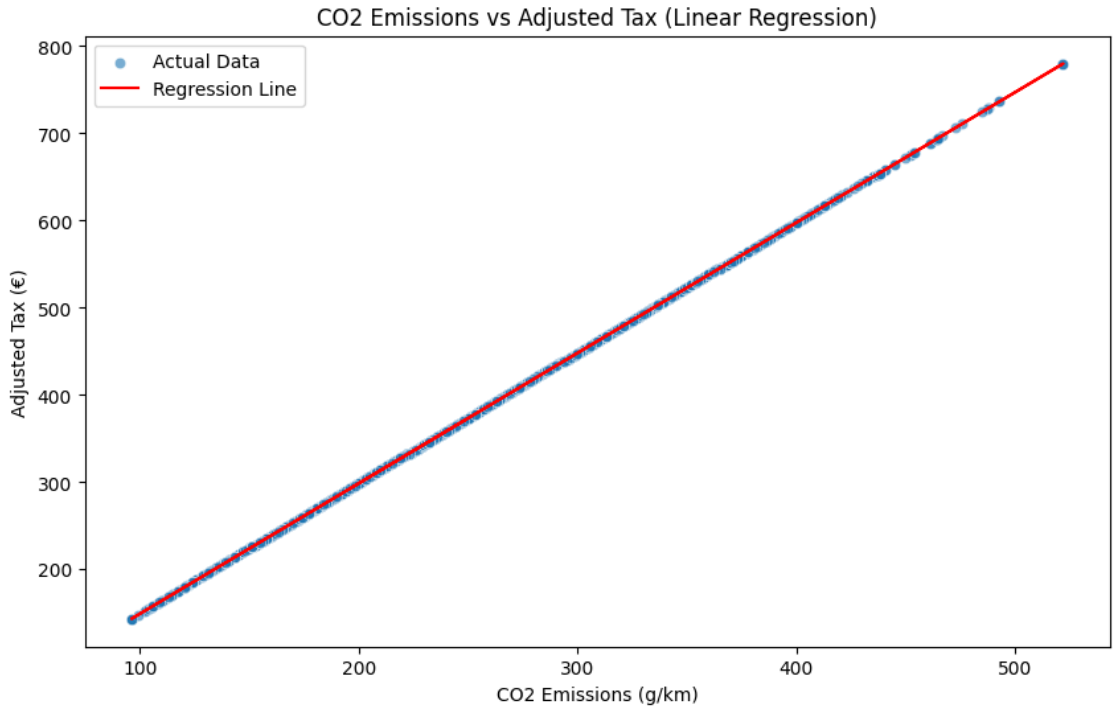

Average CO2 Emissions per Segment
Average CO2 Emissions (g/km)
Segment
0
50
100
150
200
250
300
Economy
Standard
Luxury
Premium
Super



R^2 Score: 0.9898
Mean Squared Error: 202.09
Regression Coefficient (Slope): 2.39
Intercept: -225.52




```
New Adjusted Tax Formula:
Tax (€) = 1.49 * CO2 Emissions (g/km)

New Model Evaluation:
R² Score (New Formula): 1.0000
Mean Squared Error (New Formula): 0.00
Regression Coefficient (New Formula): 1.49
Intercept (New Formula): 0.00
```




```
old tax 2764728.0
new(ish) tax 4422857.52
new formula tax 2757346.3200000003
```