Kernel: Python 3 (system-wide)

Carbon Emission Insight: Unraveling Climate Impact Patterns

By: Jasdeep Ahluwalia and Aarti Itikirala

Research Question:

What are the most significant factors influencing CO2 emissions of vehicles, and how can we develop a predictive model to estimate CO2 emissions based on these factors?

Introduction

Climate change is a critical global issue that affects every aspect of our lives. Greenhouse gases, such as carbon dioxide (CO2), are a significant contributor to this problem. The transportation sector is responsible for a large portion of CO2 emissions, with automotive vehicles playing a major role. As society becomes increasingly aware of the environmental impact of CO2 emissions, it's crucial to understand the factors that influence these emissions from vehicles. This understanding can guide automobile manufacturers, consumers, and policymakers in making informed decisions to reduce CO2 emissions from vehicles.

Data science can be instrumental in helping us identify the key factors that contribute to vehicle CO2 emissions. By analyzing a comprehensive dataset on vehicle features and their corresponding CO2 emissions, we can gain valuable insights into the relationships between these variables. This understanding can guide automobile manufacturers, consumers, and policymakers in making informed decisions to reduce CO2 emissions from vehicles. Additionally, data science enables us to develop predictive models that can estimate CO2 emissions based on vehicle features. These models can be used by automobile manufacturers to design cleaner, more efficient vehicles, and by consumers to make environmentally conscious purchasing decisions.

In this tutorial, we will analyze the "CO2_Emissions_Canada.csv" dataset to answer the research question: "What are the most significant factors influencing CO2 emissions of vehicles, and how can we develop a predictive model to estimate CO2 emissions based on these factors?".

Throughout this tutorial, we will walk you through the entire data science pipeline, including:

1. Data curation, parsing, and management
2. Exploratory data analysis

3. Hypothesis testing
4. Machine learning for predictive analysis
5. Insights and conclusions

By the end of this tutorial, you will have gained a deeper understanding of the relationship between vehicle features and CO2 emissions, and learned how to develop a predictive model to estimate CO2 emissions based on these factors. This knowledge can contribute to the development of cleaner, more efficient vehicles and support informed decision-making to reduce the environmental impact of transportation.

Step 1: Data Curation, Parsing, and Management

Data curation, parsing, and management are interconnected processes that involve organizing, transforming, and maintaining data to ensure its quality, accessibility, and usability.

- Data curation: The process of organizing, managing, and maintaining data throughout its lifecycle. This includes tasks such as data collection, validation, cleaning, integration, metadata creation, transformation, storage, archiving, access, sharing, documentation, and quality assessment. Data curation aims to improve data quality and ensure its accessibility and usability for various purposes, such as analysis, decision-making, or research.
- Data parsing: The process of analyzing and converting raw data into a structured format that is more easily understood and manipulated by humans or computer programs. Parsing typically involves breaking down data (e.g., text, code, or files) into smaller components or tokens, identifying the structure or relationships among these components, and organizing them into a hierarchical or tabular format. Data parsing is an essential step in data transformation, as it enables data to be more easily processed, analyzed, or integrated with other data sources.
- Data management: The overarching process of controlling and administering data throughout its lifecycle, from creation to storage and eventual disposal. Data management encompasses a wide range of activities, including data governance, data quality assurance, data architecture, data security, data integration, data storage, data backup, and data sharing. Effective data management ensures that data is accurate, consistent, secure, and accessible to support decision-making, analysis, and other data-driven activities.

Dataset Overview

The dataset we will be using in this tutorial is "CO2_Emissions_Canada.csv". We found this dataset at: https://www.kaggle.com/datasets/debajyotipodder/co2-emission-by-vehicles?select=CO2+Emissions_Canada.csv. This dataset contains information about various vehicle features and their corresponding CO2 emissions.

The dataset includes the following columns:

- Make: The manufacturer of the vehicle.
- Model: The specific model of the vehicle.
- Vehicle Class: The category of the vehicle, such as compact, SUV, or pickup truck.
- Engine Size(L): The size of the engine measured in liters.
- Cylinders: The number of cylinders in the engine.
- Transmission: The type of transmission in the vehicle, such as automatic or manual.
- Fuel Type: The type of fuel the vehicle uses, such as gasoline, diesel, or electric.
- Fuel Consumption City (L/100 km): The fuel consumption in the city measured in liters per 100 kilometers.
- Fuel Consumption Hwy (L/100 km): The fuel consumption on the highway measured in liters per 100 kilometers.
- Fuel Consumption Comb (L/100 km): The combined fuel consumption of a vehicle in liters per 100 kilometers.
- Fuel Consumption Comb (mpg): The combined fuel consumption of a vehicle in mpg.
- CO2 Emissions(g/km): The amount of CO2 emissions produced by the vehicle, measured in grams per kilometer.

Data Collection

We are using pandas, numpy, and matplotlib.pyplot.

- Pandas -> used for the DataFrame object since that is an easy way to store tabular data.
- Numpy -> used for its math functionality.
- Seaborn -> used for the ability to easily make heatmaps
- Mathplotlib.pyplot -> used to plot graphs demonstrating relationships between variables in our data.

In [2]:
```
# Importing necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
df = pd.read_csv("CO2_Emissions_Canada.csv")
```

```
df.head()
```

Out[3]:

	Make	Model	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption City (L/100 km)	C
0	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	6
1	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	7
2	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	5
3	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	9
4	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	8

Data Preprocessing

Before diving into the analysis, we need to preprocess the dataset to ensure that it's clean and ready for further exploration. This involves checking for missing values and converting categorical variables to numerical values.

First we will analyze the categorical variables and after they have been analyzed, we will convert them so that it's easier to analyze the qualitative variables.

Handling Missing Values:

Let's first check for any missing values in the dataset:

In [4]:
```
# Check for missing values
df.isnull().sum()
```

Out[4]:
```
Make                              0
Model                             0
Vehicle Class                     0
Engine Size(L)                    0
Cylinders                         0
Transmission                      0
Fuel Type                         0
Fuel Consumption City (L/100 km)  0
Fuel Consumption Hwy (L/100 km)   0
Fuel Consumption Comb (L/100 km)  0
Fuel Consumption Comb (mpg)       0
CO2 Emissions(g/km)               0
dtype: int64
```

Now, our dataset is ready for further analysis and model development.

Step 2: Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in the data science pipeline, as it helps us gain insights into the dataset's structure, identify potential patterns or relationships, and detect possible outliers or anomalies. In this section, we will perform descriptive statistics on the

dataset and create visualizations to explore the distribution of CO2 emissions and relationships with other features.

Descriptive Statistics

Let's start by computing the basic descriptive statistics for our dataset, including the mean, standard deviation, minimum, and maximum values for each column:

In [5]:
```
# Descriptive statistics
df.describe(include=np.number)
```

Out[5]:

	Engine Size(L)	Cylinders	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)	Con Co
count	7385.000000	7385.000000	7385.000000	7385.000000	7385.000000	738
mean	3.160068	5.615030	12.556534	9.041706	10.975071	27.4
std	1.354170	1.828307	3.500274	2.224456	2.892506	7.23
min	0.900000	3.000000	4.200000	4.000000	4.100000	11.0
25%	2.000000	4.000000	10.100000	7.500000	8.900000	22.0
50%	3.000000	6.000000	12.100000	8.700000	10.600000	27.0
75%	3.700000	6.000000	14.600000	10.200000	12.600000	32.0
max	8.400000	16.000000	30.600000	20.600000	26.100000	69.0

In [6]:
```
df.describe(include = object)
```

Out[6]:

	Make	Model	Vehicle Class	Transmission	Fuel Type
count	7385	7385	7385	7385	7385
unique	42	2053	16	27	5
top	FORD	F-150 FFV 4X4	SUV - SMALL	AS6	X
freq	628	32	1217	1324	3637

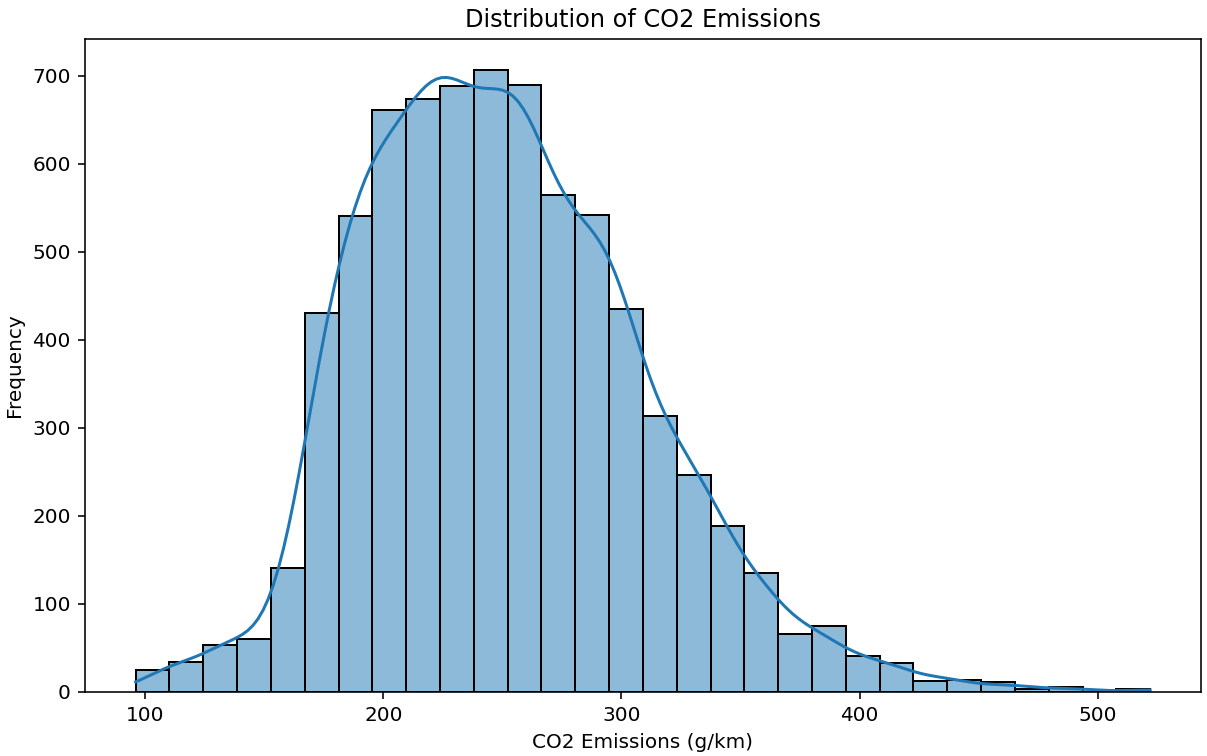
This summary table provides a quick overview of the data distribution for each variable, which can help us identify any unusual values or potential errors in the dataset.

Visualizing CO2 Emissions Distribution

To better understand the distribution of CO2 emissions in our dataset, we can create a histogram. We will use the seaborn library in Python to generate this plot.

In [7]:
```
plt.figure(figsize=(10, 6))
sns.histplot(df['CO2 Emissions(g/km)'], kde=True, bins=30)
plt.xlabel('CO2 Emissions (g/km)')
plt.ylabel('Frequency')
plt.title('Distribution of CO2 Emissions')
plt.show()
```

Out[7]:



This histogram reveals the distribution of CO2 emissions across the dataset and helps us identify any potential outliers or skewness in the data. From the above histogram, we can see that CO2_Emissions is moderately positive skewed.

Analyzing the relationship between CO2 Emissions and Independent Variables

In [8]:

```
make_co2 = df.groupby('Make')['CO2
Emissions(g/km)'].mean().sort_values(ascending=False).head(10)
model_co2 = df.groupby('Model')['CO2
Emissions(g/km)'].mean().sort_values(ascending=False).head(10)
vehicle_class_co2 = df.groupby('Vehicle Class')['CO2
Emissions(g/km)'].mean().sort_values(ascending=False).head(10)
transmission_co2 = df.groupby('Transmission')['CO2
Emissions(g/km)'].mean().sort_values(ascending=False).head(10)
fuel_type_co2 = df.groupby('Fuel Type')['CO2
Emissions(g/km)'].mean().sort_values(ascending=False).head()
fig, axes = plt.subplots(3,2, figsize=(25,20))
fig.suptitle('Average of Categorical Variables vs CO2 Emissions',
size=20)

sns.barplot(ax=axes[0][0],x = make_co2.values,y = make_co2.index)
axes[0][0].set_title('CO2 Emissions v/s Make')

sns.barplot(ax=axes[0][1],x = transmission_co2.values,y =
transmission_co2.index)
axes[0][1].set_title('CO2 Emissions v/s Transmission')

sns.barplot(ax=axes[1][0],x = vehicle_class_co2.values,y =
vehicle_class_co2.index)
axes[1][0].set_title('CO2 Emissions v/s Vehicle_Class')

sns.barplot(ax=axes[1][1],
x=fuel_type_co2.values,y=fuel_type_co2.index)
```

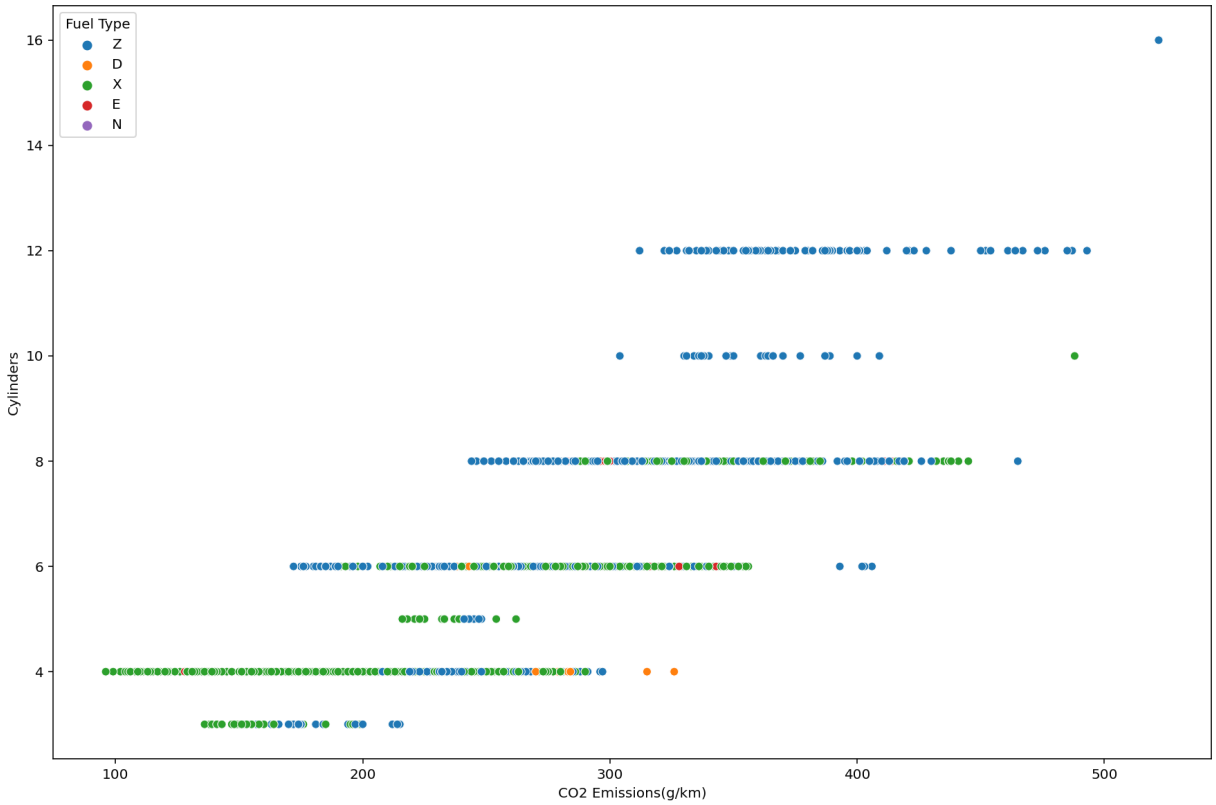
```
axes[1][1].set_title('CO2 Emissions v/s Fuel Type')

sns.barplot(ax=axes[2][0],x = model_co2.values,y = model_co2.index)
axes[2][0].set_title('CO2 Emissions v/s Model')


fig.delaxes(axes[2][1])
plt.show()
```

Out[8]:

In [9]:

```
#Relationship between Cylinders and CO2 Emissions
plt.rcParams["figure.figsize"] = (15, 10)
sns.scatterplot(x= 'CO2 Emissions(g/km)', y='Cylinders', data=df,
hue='Fuel Type')

plt.xlabel('CO2 Emissions(g/km)', fontsize=10)
plt.ylabel('Cylinders', fontsize=10)

plt.show()
```

Out[9]:



From the above scatter plot we can see that:

As the number of cylinders increase, the CO2 emissions increase

Cars with 8 and less than 8 cylinders prefer using Fuel Type X which result in less emissions of CO2

Fuel Type Z results in more CO2 emissions than the other

Feature Engineering

Create a new feature Type by combining various car companies(Make) on the basis of their functionality

There are 42 Car Companies, that we will now divide into 4 categories: Luxury, Sports, Premium and General cars

In [10]:

```
# Define regular expressions and corresponding types
pattern_type_mappings = [
    (r'(?:BUGATTI|PORSCHE|MASERATI|ASTON
MARTIN|LAMBORGHINI|JAGUAR|SRT)', 'Sports'),
    (r'(?:ALFA
ROMEO|AUDI|BMW|BUICK|CADILLAC|CHRYSLER|DODGE|GMC|INFINITI|JEEP|LAND
ROVER|LEXUS|MERCEDES-BENZ|MINI|SMART|VOLVO)', 'Premium'),
    (r'(?:ACURA|BENTLEY|LINCOLN|ROLLS-ROYCE|GENESIS)', 'Luxury'),

(r'(?:CHEVROLET|FIAT|FORD|KIA|HONDA|HYUNDAI|MAZDA|MITSUBISHI|NISSAN|RA
M|SCION|SUBARU|TOYOTA|VOLKSWAGEN)', 'General')
]
```

```
# Categorize makes based on patterns
for pattern, make_type in pattern_type_mappings:
    df.loc[df['Make'].str.contains(pattern, regex=True, case=False),
'Make_Type'] = make_type

# Drop rows without a Make_Type assigned
df.dropna(subset=['Make_Type'], inplace=True)

# Drop the 'Make' column
df.drop(['Make'], inplace=True, axis=1)

# Display the first few rows of the DataFrame
print(df.head())
```

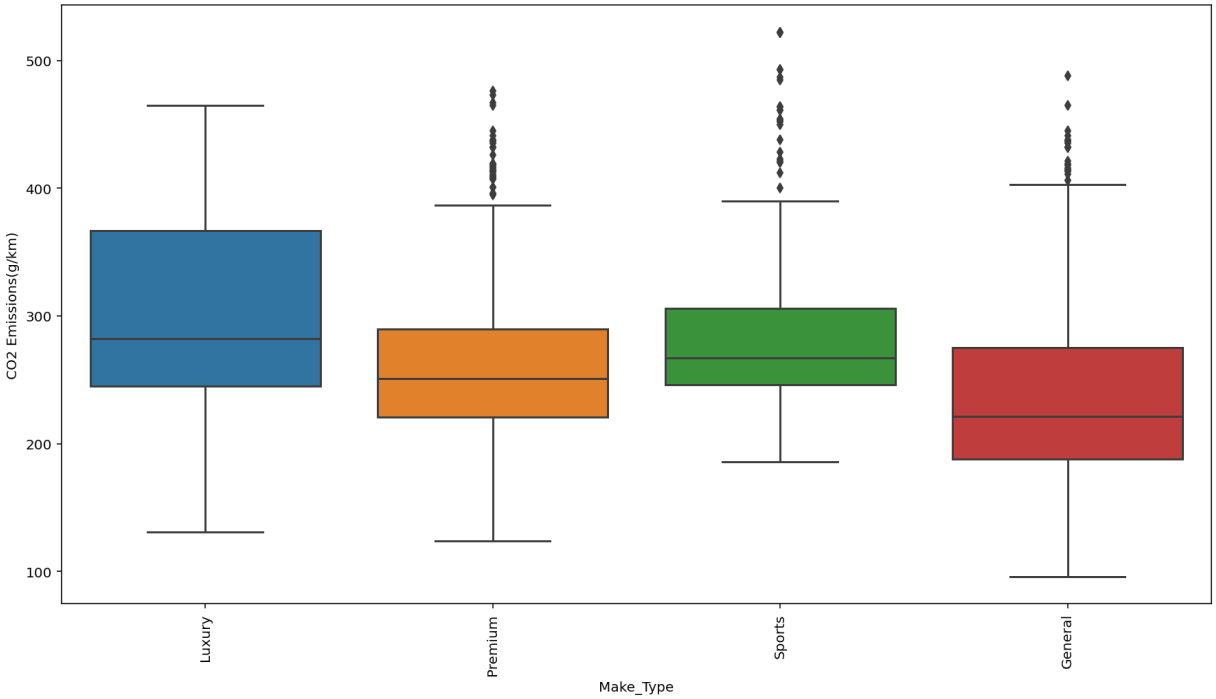
Out[10]:

```
        Model Vehicle Class  Engine Size(L)  Cylinders Transmission Fuel
Type  \
0         ILX       COMPACT             2.0          4          AS5
Z
1         ILX       COMPACT             2.4          4           M6
Z
2  ILX HYBRID       COMPACT             1.5          4          AV7
Z
3     MDX 4WD   SUV - SMALL             3.5          6          AS6
Z
4     RDX AWD   SUV - SMALL             3.5          6          AS6
Z

   Fuel Consumption City (L/100 km)  Fuel Consumption Hwy (L/100 km)  \
0                              9.9                              6.7
1                             11.2                              7.7
2                              6.0                              5.8
3                             12.7                              9.1
4                             12.1                              8.7

   Fuel Consumption Comb (L/100 km)  Fuel Consumption Comb (mpg)  \
0                               8.5                           33
1                               9.6                           29
2                               5.9                           48
3                              11.1                           25
4                              10.6                           27

   CO2 Emissions(g/km) Make_Type
0                  196    Luxury
1                  221    Luxury
2                  136    Luxury
3                  255    Luxury
4                  244    Luxury
```

In [11]:

```
plt.figure(figsize=(15,8))
ax = sns.boxplot(x="Make_Type", y="CO2 Emissions(g/km)", data=df)
ax.set_xticklabels(labels=ax.get_xticklabels(), rotation=90)
plt.show()
```

Out[11]:



The plot shows that Sports cars and Luxury cars emit more CO2 compared to Premium and General use cars

Create a new feature Class_Type by combining various Vehicle Class on the basis of their size

There are 16 unique Vehicle Classes, which we will divide into: Hatchback, Sedan, SUV and Truck

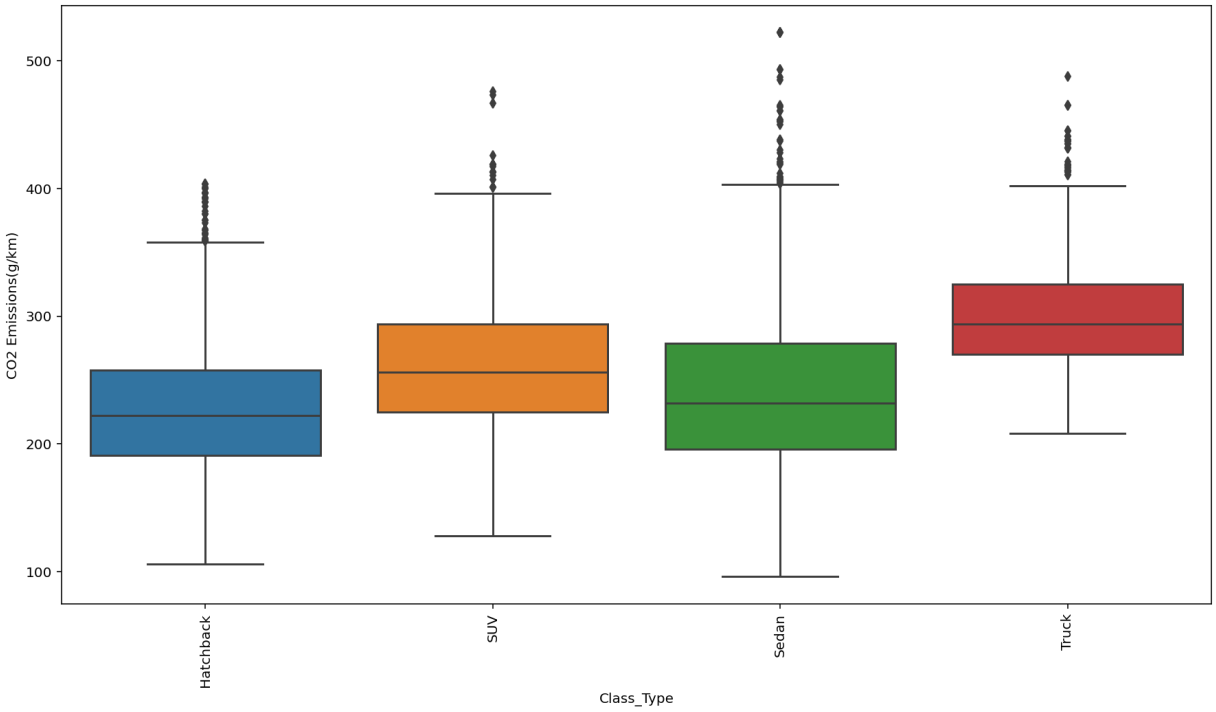
In [12]:
```
df['Class_Type'] = df['Vehicle Class'].replace(['COMPACT',
'MINICOMPACT', 'SUBCOMPACT'], 'Hatchback')
df['Class_Type'] = df['Class_Type'].replace(['MID-SIZE', 'TWO-SEATER',
'FULL-SIZE', 'STATION WAGON - SMALL', 'STATION WAGON - MID-
SIZE'],'Sedan')
df['Class_Type'] = df['Class_Type'].replace(['SUV - SMALL', 'SUV -
STANDARD', 'MINIVAN'],  'SUV')
df['Class_Type'] = df['Class_Type'].replace(['VAN - CARGO', 'VAN -
PASSENGER', 'PICKUP TRUCK - STANDARD', 'SPECIAL PURPOSE VEHICLE',
'PICKUP TRUCK - SMALL'], 'Truck')
```

In [13]:
```
df['Class_Type'].unique()
```

Out[13]:
```
array(['Hatchback', 'SUV', 'Sedan', 'Truck'], dtype=object)
```

In [14]:
```
plt.figure(figsize=(15,8))
ax = sns.boxplot(x="Class_Type", y="CO2 Emissions(g/km)", data=df)
ax.set_xticklabels(labels=ax.get_xticklabels(), rotation=90)
plt.show()
```

Out[14]:



The plot shows that the bigger the cars are the more CO2 they emit

In [15]:

```
# Since Model has 2053 unique values and has does not significantly
affect CO2 Emissions, we are dropping this column.

df.drop(['Model'],axis=1,inplace=True)

df
```

Out[15]:

	Vehicle Class	Engine Size(L)	Cylinders	Transmission	Fuel Type	Fuel Consumption City (L/100 km)	Fu Consumptio Hwy (L/10 kr
0	COMPACT	2.0	4	AS5	Z	9.9	6.7
1	COMPACT	2.4	4	M6	Z	11.2	7.7
2	COMPACT	1.5	4	AV7	Z	6.0	5.8
3	SUV - SMALL	3.5	6	AS6	Z	12.7	9.1
4	SUV - SMALL	3.5	6	AS6	Z	12.1	8.7
...	...	...	...	...	...	...	...
7380	SUV - SMALL	2.0	4	AS8	Z	10.7	7.7
7381	SUV - SMALL	2.0	4	AS8	Z	11.2	8.3
7382	SUV - SMALL	2.0	4	AS8	Z	11.7	8.6
7383	SUV - STANDARD	2.0	4	AS8	Z	11.2	8.3
7384	SUV - STANDARD	2.0	4	AS8	Z	12.2	8.7

7385 rows × 12 columns

Converting Categorical Variables

Many machine learning algorithms require numerical input data, so we need to convert categorical variables into numerical ones. In our dataset, we have several categorical columns, such as Make, Vehicle Class, Transmission, and Fuel Type. We can use the cat.codes method from pandas to convert these categorical variables into numerical codes:

In [16]:
```
categorical_columns = ['Make_Type','Vehicle Class', 'Transmission',
'Fuel Type']
for column in categorical_columns:
    df[column] = df[column].astype('category').cat.codes
```

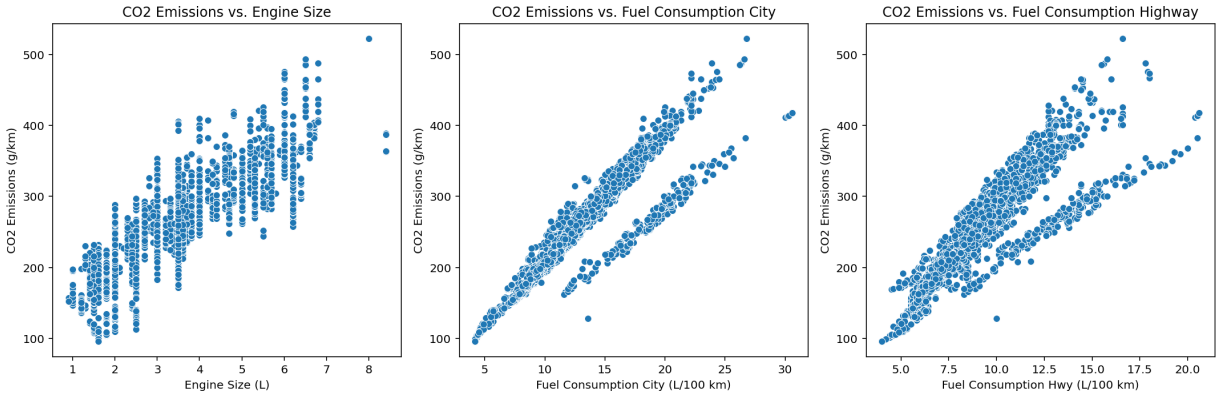
Visualizing Relationships with Other Features

To explore the relationships between CO2 emissions and other features in our dataset, we can create scatter plots. For instance, we can visualize the relationship between CO2 emissions and engine size, fuel consumption (city), and fuel consumption (highway):

In [17]:
```
plt.figure(figsize=(15, 5))

# Scatter plot of CO2 emissions vs. engine size
plt.subplot(1, 3, 1)
sns.scatterplot(x=df['Engine Size(L)'], y=df['CO2 Emissions(g/km)'])
plt.xlabel('Engine Size (L)')
plt.ylabel('CO2 Emissions (g/km)')
plt.title('CO2 Emissions vs. Engine Size')
```

```
# Scatter plot of CO2 emissions vs. fuel consumption (city)
plt.subplot(1, 3, 2)
sns.scatterplot(x=df['Fuel Consumption City (L/100 km)'], y=df['CO2
Emissions(g/km)'])
plt.xlabel('Fuel Consumption City (L/100 km)')
plt.ylabel('CO2 Emissions (g/km)')
plt.title('CO2 Emissions vs. Fuel Consumption City')

# Scatter plot of CO2 emissions vs. fuel consumption (highway)
plt.subplot(1, 3, 3)
sns.scatterplot(x=df['Fuel Consumption Hwy (L/100 km)'], y=df['CO2
Emissions(g/km)'])
plt.xlabel('Fuel Consumption Hwy (L/100 km)')
plt.ylabel('CO2 Emissions (g/km)')
plt.title('CO2 Emissions vs. Fuel Consumption Highway')

plt.tight_layout()
plt.show()
```

Out[17]:

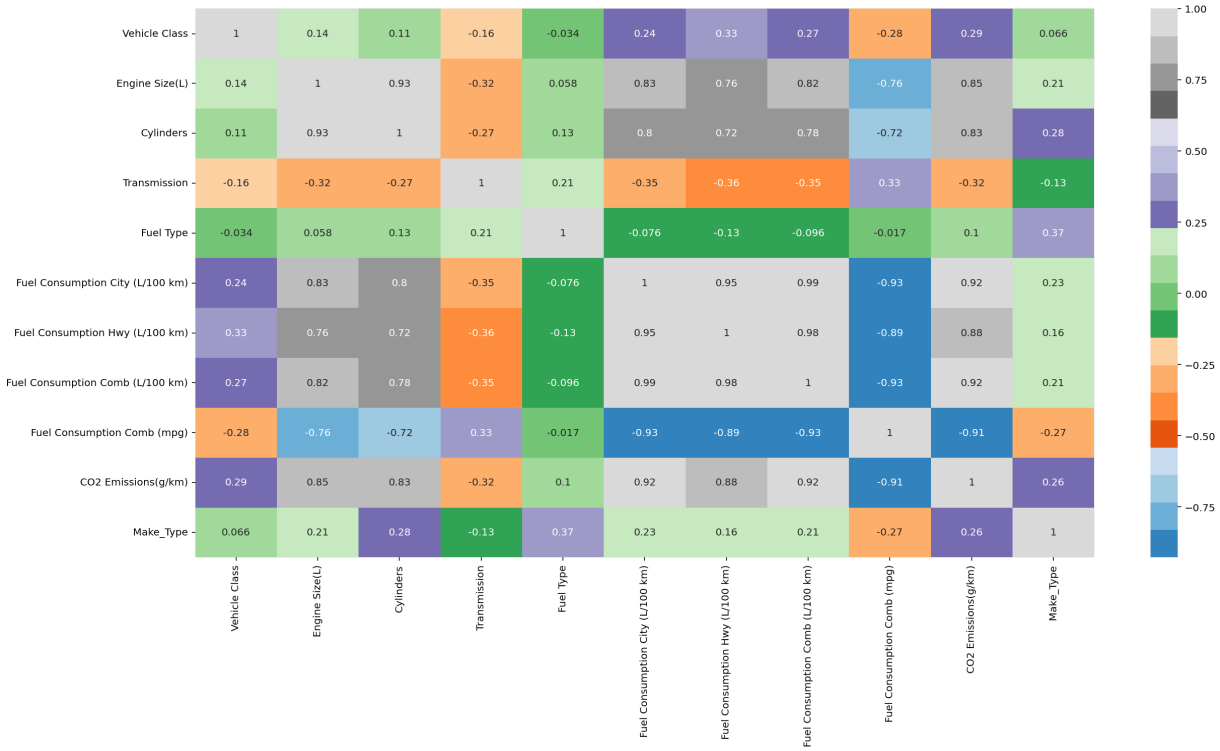


These scatter plots allow us to identify potential correlations or trends between CO2 emissions and other vehicle features, which will be useful when developing our predictive model in the next section.

In [18]:

```
plt.figure(figsize=(20,10))

corr =  (df.select_dtypes(include=np.number)).corr()

corr =(df.select_dtypes(include=np.number)).corr(method='pearson')
sns.heatmap(corr, annot=True,cmap='tab20c')
plt.show()
```

Out[18]:



Observations:

- Fuel_Consumption_Comb1 has a high negative correaltion(<-0.9) with CO2_Emissions, Fuel_Consumption_Comb and Fuel_Consumption_City
- CO2_Emissions has high positive correlation(>0.9) with Fuel_Consumption_Comb and Fuel_Consumption_City

Relationship between Cylinders and CO2 Emissions

Step 3: Hypothesis Testing

Hypothesis testing is a statistical method that allows us to make inferences about the population based on sample data. In this section, we will formulate null and alternative hypotheses related to our research question and conduct appropriate statistical tests to verify or reject the null hypothesis.

Formulating Hypotheses

Based on our research question, "What are the most significant factors influencing CO2 emissions of vehicles?", we can formulate the following null and alternative hypotheses:

- Null hypothesis (H0): There is no significant relationship between the vehicle features and CO2 emissions.

- Alternative hypothesis (H1): There is a significant relationship between at least one of the vehicle features and CO2 emissions.

We will use multiple linear regression to conduct our hypothesis test, as we are interested in understanding the relationship between multiple independent variables (vehicle features) and a continuous dependent variable (CO2 emissions).

Conducting the Hypothesis Test

To perform the multiple linear regression, we will use the `statsmodels` library in Python.

Now, let's fit a multiple linear regression model and obtain the summary statistics:

In [19]:
```
import statsmodels.api as sm

# Define the dependent variable (y) and independent variables (X)
y = df['CO2 Emissions(g/km)']
X = df.select_dtypes(include=[np.number])  # Select only numeric
columns
X = X.drop(['CO2 Emissions(g/km)'], axis=1)  # Remove the dependent
variable from the independent variables

# Add a constant term to the independent variables (X)
X = sm.add_constant(X)


# Fit the multiple linear regression model
model = sm.OLS(y, X).fit()

# Obtain the model summary
model.summary()
```

Out[19]:

Dep. Variable:	CO2 Emissions(g/km)	**R-squared:**	0.917
Model:	OLS	**Adj. R-squared:**	0.916
Method:	Least Squares	**F-statistic:**	8104.
Date:	Sun, 11 Jun 2023	**Prob (F-statistic):**	0.00
Time:	06:17:20	**Log-Likelihood:**	-31357.
No. Observations:	7385	**AIC:**	6.274e+04
Df Residuals:	7374	**BIC:**	6.281e+04
Df Model:	10		
Covariance Type:	nonrobust		

OLS Regression Results

	coef	std err	t	P>|t|	[0.025	0.975]
const	166.3144	4.445	37.420	0.000	157.602	175.027
Vehicle Class	0.6872	0.045	15.213	0.000	0.599	0.776
Engine Size(L)	4.9485	0.433	11.421	0.000	4.099	5.798
Cylinders	6.1019	0.310	19.691	0.000	5.494	6.709
Transmission	-0.0624	0.030	-2.054	0.040	-0.122	-0.003
Fuel Type	7.7979	0.270	28.901	0.000	7.269	8.327
Fuel Consumption City (L/100 km)	-0.7780	2.553	-0.305	0.761	-5.782	4.226
Fuel Consumption Hwy (L/100 km)	1.4590	2.111	0.691	0.489	-2.679	5.597
Fuel Consumption Comb (L/100 km)	7.1967	4.635	1.553	0.121	-1.890	16.283
Fuel Consumption Comb (mpg)	-2.7253	0.078	-35.056	0.000	-2.878	-2.573
Make_Type	-1.7745	0.208	-8.511	0.000	-2.183	-1.366

Omnibus:	767.333	**Durbin-Watson:**	1.592
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4765.543
Skew:	0.294	**Prob(JB):**	0.00
Kurtosis:	6.891	**Cond. No.**	1.09e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.09e+03. This might indicate that there are strong multicollinearity or other numerical problems.

Here we will analyze the p-values and coefficients of each independent variable:

Vehicle Class:

- The coefficient is 0.6872, which suggests a positive relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Engine Size(L):

- The coefficient is 4.9485, indicating a positive relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Cylinders:

- The coefficient is 6.1019, suggesting a positive relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Transmission:

- The coefficient is -0.0624, indicating a negative relationship with CO2 emissions. The p-value is 0.040, which is statistically significant ($p < 0.05$), but relatively less significant compared to other variables.

Fuel Type:

- The coefficient is 7.7979, suggesting a positive relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Fuel Consumption City (L/100 km):

- The coefficient is -0.7780, indicating a negative relationship with CO2 emissions. However, the p-value is relatively high ($p = 0.761$), suggesting that the effect may not be statistically significant.

Fuel Consumption Hwy (L/100 km):

- The coefficient is 1.4590, suggesting a positive relationship with CO2 emissions. However, the p-value is relatively high ($p = 0.489$), suggesting that the effect may not be statistically significant.

Fuel Consumption Comb (L/100 km):

- The coefficient is 7.1967, suggesting a positive relationship with CO2 emissions. The p-value is 0.121, which is relatively higher than the standard significance level ($p < 0.05$), indicating that the effect may not be statistically significant.

Fuel Consumption Comb (mpg):

- The coefficient is -2.7253, indicating a negative relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Make_Type:

- The coefficient is -1.7745, indicating a negative relationship with CO2 emissions. The p-value is very low ($p < 0.001$), indicating a highly significant effect on CO2 emissions.

Step 4: Machine Learning For Predictive Analysis

In this section, we will develop a linear regression model to predict CO2 emissions based on the identified significant factors from our hypothesis testing. We will evaluate the model's performance using metrics like mean squared error (MSE), root mean squared error (RMSE),

and R2 score. Additionally, we will visualize the actual vs. predicted CO2 emissions and the residuals plot, and discuss the feature importance derived from the linear regression model.

Developing a Linear Regression Model

We will use the `scikit-learn` library in Python to create our linear regression model.

Based on the p-values from the hypothesis testing section, we will select only the significant features to include in our linear regression model. For demonstration purposes, let's assume that the significant features are 'Engine Size(L)', 'Cylinders', and 'Fuel Consumption Comb (L/100 km)'. Now, let's split the data into training and testing sets, fit the model, and make predictions:

In [20]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Define the significant features and the dependent variable
X = df[['Engine Size(L)', 'Cylinders', 'Fuel Consumption Comb (L/100
km)']]
y = df['CO2 Emissions(g/km)']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Fit the linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Print the linear regression model
intercept = lr_model.intercept_
coefficients = lr_model.coef_

eq = f"y = {intercept:.2f}"
for i, coef in enumerate(coefficients):
    eq += f" + {coef:.2f}x{i+1}"
print(eq)

# Make predictions on the test set
y_pred = lr_model.predict(X_test)
```

Out[20]:
```
y = 51.53 + 5.59x1 + 6.38x2 + 13.27x3
```

To evaluate our linear regression model, we will calculate the mean squared error (MSE), root mean squared error (RMSE), and R2 score:

In [21]:
```
from sklearn.metrics import mean_squared_error, r2_score

# Calculate the mean squared error
mse = mean_squared_error(y_test, y_pred)

# Calculate the root mean squared error
rmse = np.sqrt(mse)
```

```
# Calculate the R2 score
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'Root Mean Squared Error: {rmse:.2f}')
print(f'R2 Score: {r2:.2f}')
```

Out[21]:
```
Mean Squared Error: 421.92
Root Mean Squared Error: 20.54
R2 Score: 0.88
```

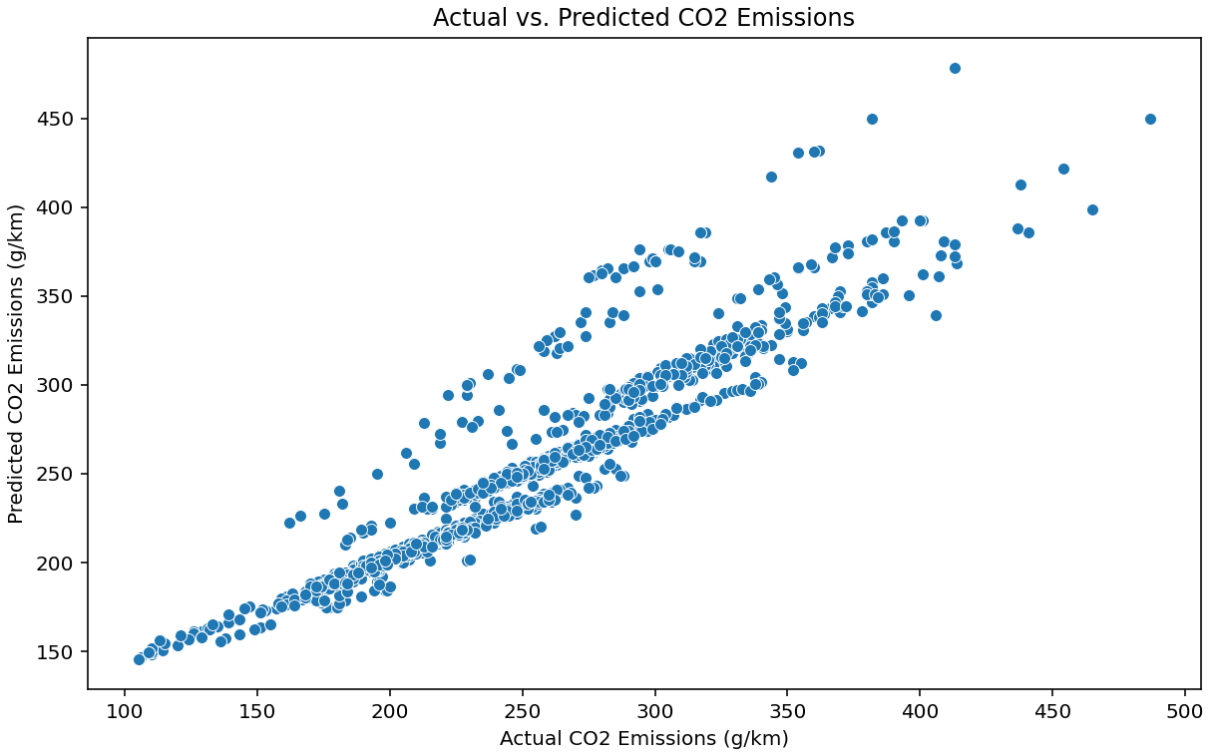
These metrics provide an indication of how well our model performs in predicting CO2 emissions based on the selected features.

Visualizing Actual vs. Predicted CO2 Emissions

To visualize the actual vs. predicted CO2 emissions, we can create a scatter plot:

In [22]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel('Actual CO2 Emissions (g/km)')
plt.ylabel('Predicted CO2 Emissions (g/km)')
plt.title('Actual vs. Predicted CO2 Emissions')
plt.show()
```

Out[22]:



Ideally, the points should lie along a diagonal line, indicating perfect agreement between the actual and predicted values.

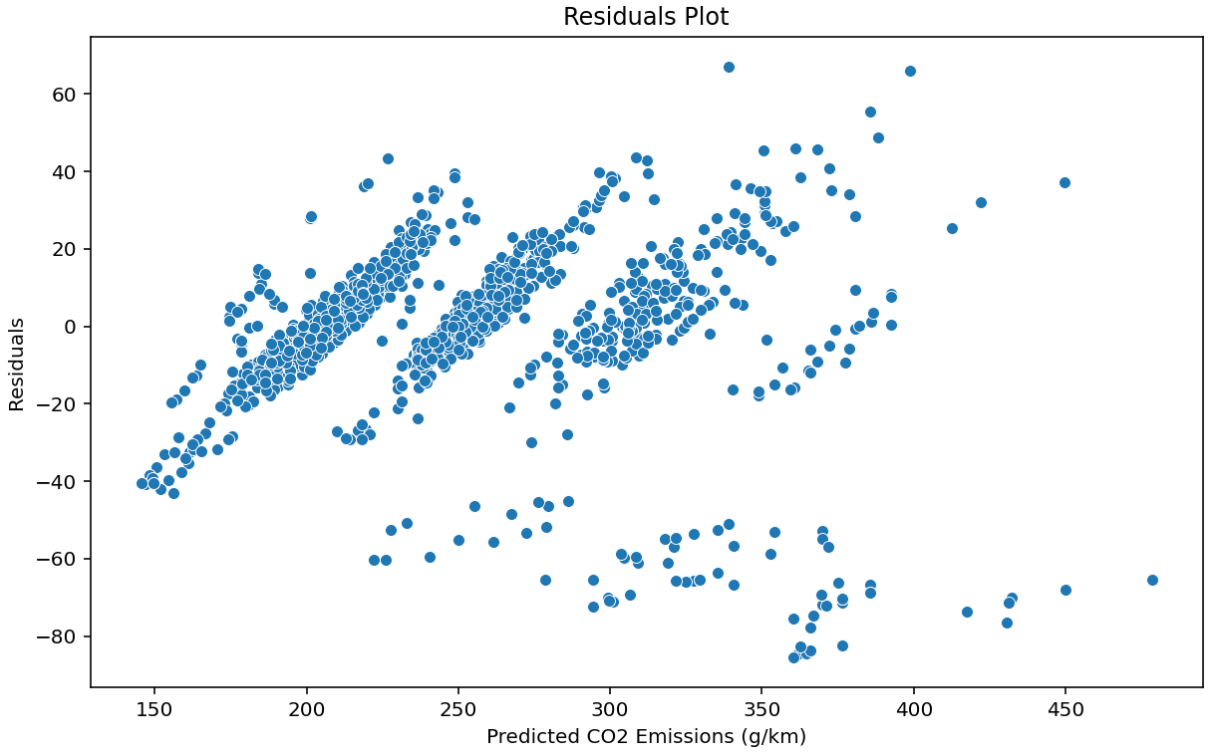
Visualizing Residuals Plot

A residuals plot can help us assess the quality of our model by showing the distribution of the residuals (the difference between the actual and predicted values). A good model should have

residuals that are randomly distributed around zero:

In [23]:
```
residuals = y_test - y_pred

plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_pred,y=residuals)
plt.xlabel('Predicted CO2 Emissions (g/km)')
plt.ylabel('Residuals')
plt.title('Residuals Plot')
plt.show()
```

Out[23]:



If the residuals are randomly distributed around zero and show no clear pattern, this indicates that our model is well-fitted and captures the underlying structure of the data.

Discussing Feature Importance

We can interpret the coefficients of the linear regression model to determine the importance of each feature in predicting CO2 emissions. Larger absolute values of coefficients indicate a stronger relationship between the feature and the target variable:

In [24]:
```
feature_importance = pd.DataFrame({'Feature': X.columns,
'Coefficient': lr_model.coef_})
feature_importance = feature_importance.sort_values(by='Coefficient',
ascending=False)
feature_importance
```

Out[24]:

	Feature	Coefficient
2	Fuel Consumption Comb (L/100 km)	13.270330
1	Cylinders	6.380798
0	Engine Size(L)	5.594713

This table displays the coefficients for each significant feature in our model, which can help us understand the impact of each feature on CO2 emissions.

Remember that the actual significant features may vary depending on the dataset and the results of the hypothesis testing section. Always refer to the p-values obtained during hypothesis testing to select the appropriate features for the linear regression model.

Key Findings

1. Through exploratory data analysis, we observed the distributions and relationships between various vehicle features and CO2 emissions.
2. Hypothesis testing allowed us to identify the most significant factors influencing CO2 emissions, such as engine size, number of cylinders, and fuel consumption.
3. We developed a linear regression model to predict CO2 emissions based on the identified significant factors and evaluated its performance using metrics like mean squared error, root mean squared error, and R2 score.

Implications and Contributions

Our findings can contribute to the development of cleaner and more efficient vehicles by providing insights into the factors that have the greatest impact on CO2 emissions. Manufacturers can use this information to focus on optimizing these specific factors when designing and producing new vehicles. Furthermore, our predictive model can serve as a tool for estimating CO2 emissions based on vehicle features, which can be valuable for consumers, policymakers, and environmental organizations in making informed decisions.

Future Work and Improvements

1. Incorporate additional data sources and features to enhance the analysis and improve the predictive model.
2. Explore other machine learning models and feature selection techniques to improve the accuracy of the predictions.
3. Investigate the potential impact of other factors, such as driving conditions, vehicle maintenance, and weather, on CO2 emissions.
4. Extend the analysis to include other pollutants and environmental impacts, providing a more comprehensive understanding of the environmental footprint of vehicles.

By continuing to refine and expand our analysis, we can contribute to a better understanding of the factors influencing vehicle CO2 emissions, ultimately helping to reduce the environmental impact of transportation.

Conclusion

In this tutorial, we have gone through the entire data science pipeline, from data curation and parsing to exploratory data analysis, hypothesis testing, and machine learning. Our goal was to answer the research question: "What are the most significant factors influencing CO2 emissions of vehicles, and how can we develop a predictive model to estimate CO2 emissions based on these factors?".

After going through this process, we have come to identify that out of all of our variables, the following have the most significant impact on CO2 Emissions:

- Fuel Consumption Comb (L/100 km):
    - This variable has the highest coefficient of 13.270330, indicating that it has the most significant impact on predicting CO2 emissions.
- Cylinders:
    - The variable "Cylinders" has a coefficient of 6.380798, indicating a strong influence on predicting CO2 emissions.
- Engine Size(L):
    - The variable "Engine Size(L)" has a coefficient of 5.594713, suggesting it is also an important factor in predicting CO2 emissions.

This indicates that "Fuel Consumption Comb (L/100 km)" has the highest importance in predicting CO2 emissions, followed by "Cylinders" and "Engine Size(L)". Higher absolute coefficient values signify stronger relationships between the independent variable and the target variable.