**DATA SCIENCE TOOLBOX: PYTHON PROGRAMMING**

**PROJECT REPORT**

(Project Semester January-April 2025)

***Data Analysis Using EDA Process***

Submitted by

**JEET LOHAR**

Registration No:- 12321298

P132 – K23FK

Course Code:- INT375

Under the Guidance of

**Dr. Karan Bajaj (UID: 32130)**

**Discipline of CSE/IT**

**Lovely School of Computer Science and Engineering**

**Lovely Professional University, Phagwara**

**CERTIFICATE**

This is to certify that JEET LOHAR bearing Registration no. 12321298 has completed INT375 project titled, **“Data Analysis using EDA process”** under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

**Signature and Name of the Supervisor**

**Designation of the Supervisor**

**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab.

Date:

**DECLARATION**

I, **JEET LOHAR**, student of *P132: BTech (Computer Science and Engineering)* under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 12-04-2025 JEET LOHAR

Registration No:- 12321298 Name of the student

**ACKNOWLEDGEMENT**

I would like to express my heartfelt gratitude to everyone who has supported and guided me throughout the completion of my project on "***Data Analysis Using the Exploratory Data Analysis (EDA) Process***."

First and foremost, I extend my sincere thanks to my professor/mentor, Dr. Karan Bajaj, for their invaluable guidance, encouragement, and expertise, which were instrumental in shaping the direction and scope of this project. Their insights on data analysis techniques and interpretation have greatly enhanced my understanding of the subject.

I am also grateful to my institution Lovely Professional University for providing me with the resources and platform to work on this project. The tools and datasets made available to me were crucial in conducting meaningful analyses.

I would like to acknowledge the contributions of my peers and colleagues, whose constructive feedback and collaboration have helped me refine my work. Their perspectives and suggestions have added depth to my analysis.

Lastly, I owe my deepest gratitude to my family and friends for their unwavering support and motivation throughout the duration of this project. Their encouragement has been a constant source of inspiration.

This project has been a valuable learning experience, and I am deeply thankful to everyone who has contributed to its successful completion.

Sincerely,  
JEET LOHAR

**INTRODUCTION**

The automotive industry is a cornerstone of modern economies, continually evolving to meet consumer needs and technological advancements. This project, titled **"Data Analysis Using the Exploratory Data Analysis (EDA) Process,"** focuses on uncovering meaningful insights from a dataset containing detailed information about various cars.

The dataset includes key attributes such as vehicle types, engine specifications, driving modes, fuel efficiency, ground clearance, and customer preferences. Analysing such data provides valuable insights into market trends, consumer preferences, and the technical features that influence purchasing decisions.

Through the EDA process, this project aims to:

1. **Identify Patterns:** Highlight trends in vehicle features, such as the most preferred driving modes or the distribution of fuel efficiency across different vehicle types.
2. **Compare Attributes:** Examine how manufacturers differ in providing attributes like ground clearance or engine oil type compatibility.
3. **Understand Preferences:** Explore customer preferences, including preferred vehicle types, driving modes, and engine configurations.
4. **Draw Insights:** Derive actionable insights to assist manufacturers and stakeholders in making informed decisions.

By leveraging Python libraries like Pandas, NumPy, Matplotlib, and Seaborn, this project demonstrates the importance of EDA in breaking down complex datasets, identifying anomalies, and generating visual representations for effective communication.

This analysis not only provides a deeper understanding of the automotive dataset but also showcases the power of data in driving innovation and strategy in the automobile industry.

**SOURCE OF DATASET**

Dataset Link: <https://carapi.app/features/vehicle-csv-download>

GitHub Link: <https://github.com/Jeet-Lohar-itzJeeSKUULL/Data_Analysis_EDA_Process>

LinkedIn Link: <https://www.linkedin.com/posts/jeet-lohar_datascience-eda-python-activity-7316861809234194433-oC6j?utm_source=share&utm_medium=member_desktop&rcm=ACoAAEPSkZEBDzqouRNgxif965r15ikguhz7p6c>

**EDA PROCESS**

**What is EDA?**

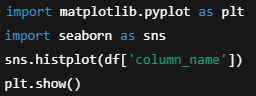
Exploratory Data Analysis (EDA) is a critical process in data analysis that involves summarizing, visualizing, and interpreting datasets to uncover underlying patterns, trends, and relationships. It serves as the foundation for developing a deeper understanding of the data and guiding subsequent steps in the data analysis or machine learning workflow.

1. **Key Goals of EDA**
2. **Understand the Dataset**
   * Identify the structure, dimensions, and types of data present.
   * Examine the quality of data by checking for missing, duplicate, or inconsistent values.
3. **Summarize Data**
   * Calculate statistical measures like mean, median, variance, and standard deviation to understand central tendencies and variability.
4. **Visualize Patterns**
   * Use plots and charts (e.g., histograms, scatter plots, box plots) to identify patterns, trends, or outliers.
5. **Detect Anomalies**
   * Spot unusual values or deviations that may need further investigation or correction.
6. **Prepare for Further Analysis**
   * Transform and clean data as needed to prepare it for modeling or hypothesis testing.
7. **Steps in the EDA Process**
8. **Data Collection and Loading**
   * Import datasets from various sources such as CSV files, databases, or APIs.

Example:

1. **Data Inspection**
   * Check data structure, types, and summary statistics.  
     Example:
2. **Data Cleaning**
   * Handle missing values, duplicates, and inconsistent data.  
     Example:



1. **Data Visualization**
   * Use visualization tools to explore distributions and relationships.  
     Example:
2. **Feature Analysis**
   * Analyse relationships between variables using correlation matrices or scatter plots.  
     Example:
3. **Interpretation and Insights**
   * Draw conclusions from the visualizations and summaries to inform decision-making or modelling.
4. **Techniques and Tools in EDA**

* **Descriptive Statistics:** Mean, median, mode, range, variance, and standard deviation.
* **Data Visualization:** Histograms, box plots, scatter plots, bar charts, and heatmaps.
* **Data Transformation:** Normalization, scaling, and encoding categorical variables.
* **Tools:** Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Plotly.

1. **Importance of EDA**

* Ensures data quality and integrity.
* Helps in understanding the dataset's structure and content.
* Guides feature selection and engineering for predictive modelling.
* Identifies patterns and trends that inform decision-making.

EDA is an iterative process that equips analysts with the tools to ask the right questions and develop data-driven solutions. It is a cornerstone of any data-driven project, laying the groundwork for meaningful analysis and actionable insights.

**ANALYSIS ON DATASET**

**# Objective 1: DATA CLEANING**

* **Introduction:**

Data cleaning is a critical process in data analysis that involves identifying and correcting errors, inconsistencies, and missing values in a dataset. This process ensures that the dataset is accurate, complete, and reliable for analysis. Poor data quality can lead to incorrect conclusions and flawed models, making data cleaning an essential step in any data-driven project.

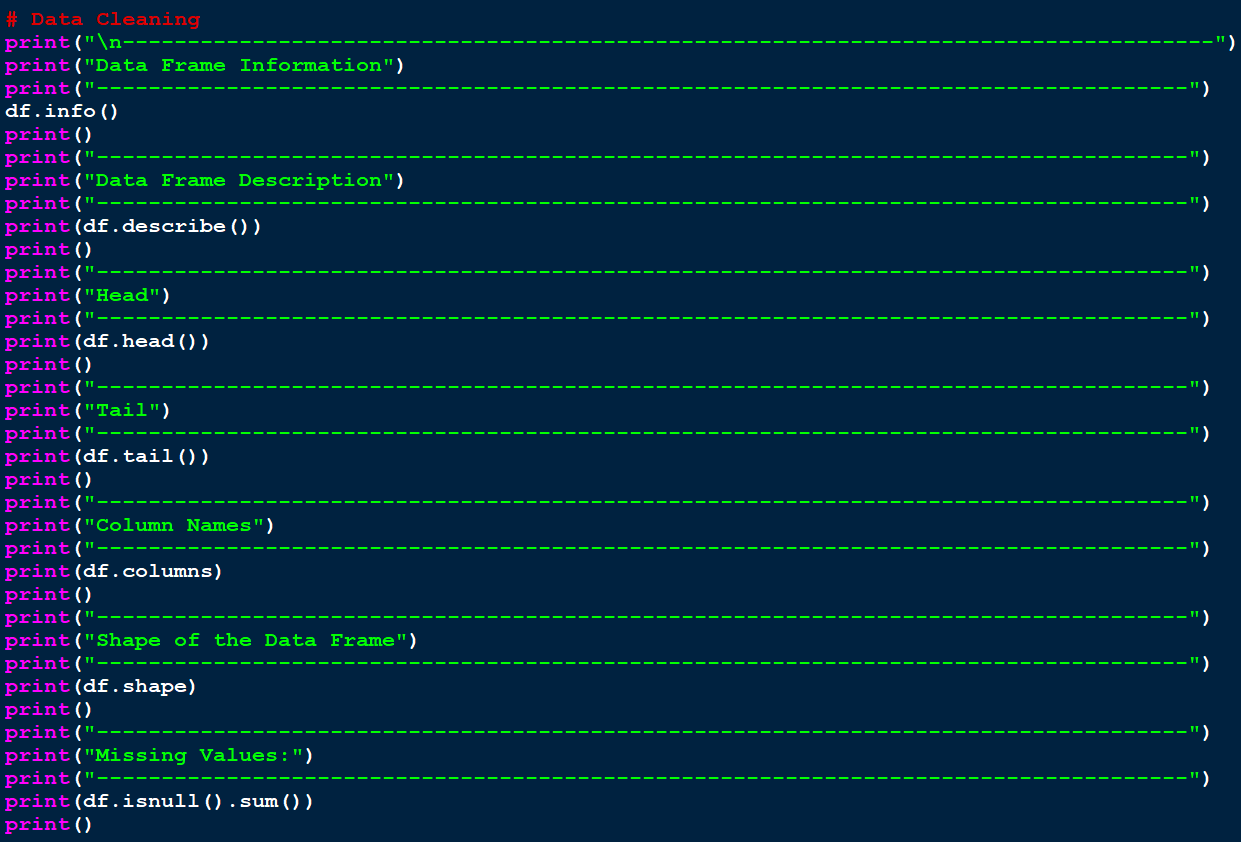
* **General Description:**

Data cleaning typically involves the following tasks:

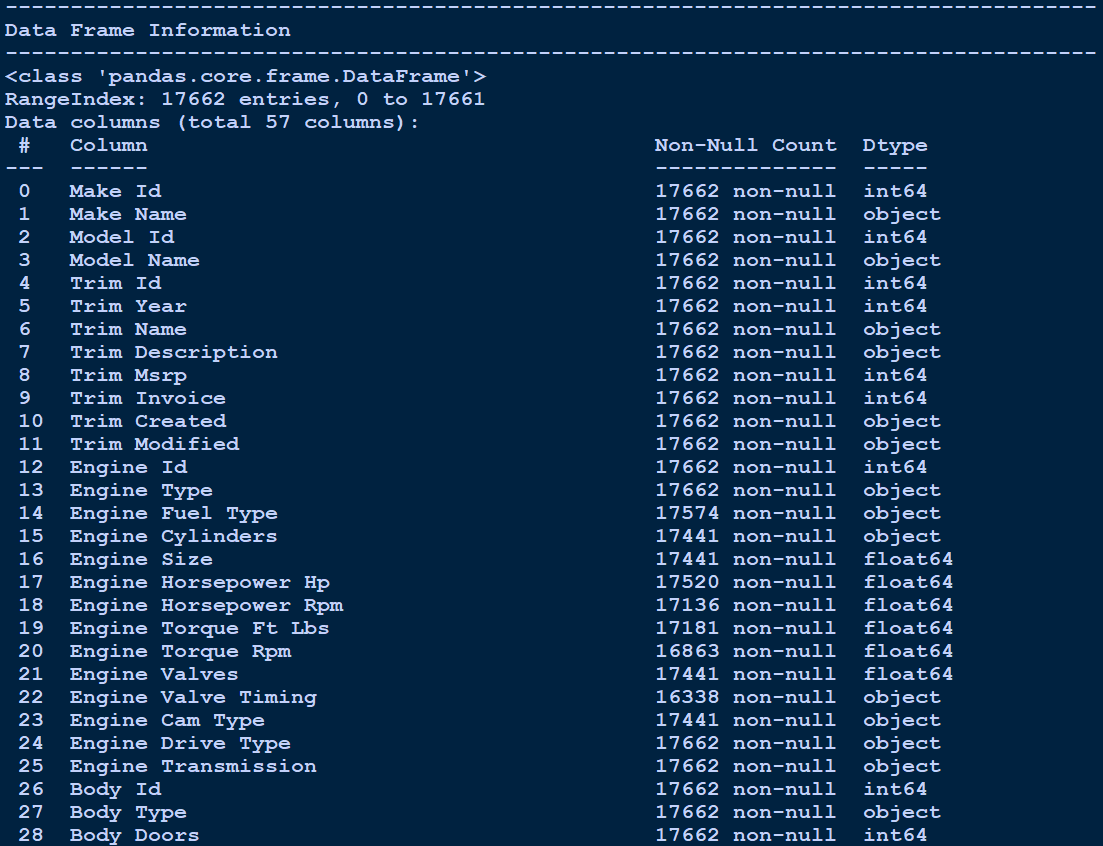
* + **Handling Missing Data:** Replacing missing values, deleting incomplete rows/columns, or imputing them using statistical methods.
  + **Removing Duplicates:** Identifying and removing duplicate entries to avoid bias in analysis.
  + **Correcting Errors:** Fixing incorrect data entries, such as typos or outliers.
  + **Standardizing Formats:** Ensuring consistency in data formats, such as date formats or categorical values.
  + **Renaming or Dropping Columns:** Renaming ambiguous column names and removing irrelevant features.
  + **Outlier Detection:** Identifying extreme values that may skew the results.
* **Specific Requirements for Data Cleaning**
  + **Dataset Accessibility:** A dataset loaded into a Python environment using Pandas.
  + **Understanding of Data:** Knowledge of the dataset's domain and context to make informed cleaning decisions.
  + **Libraries:** Python libraries like Pandas and NumPy.
  + **Tools for Visualization:** Libraries such as Matplotlib and Seaborn for detecting anomalies and trends.

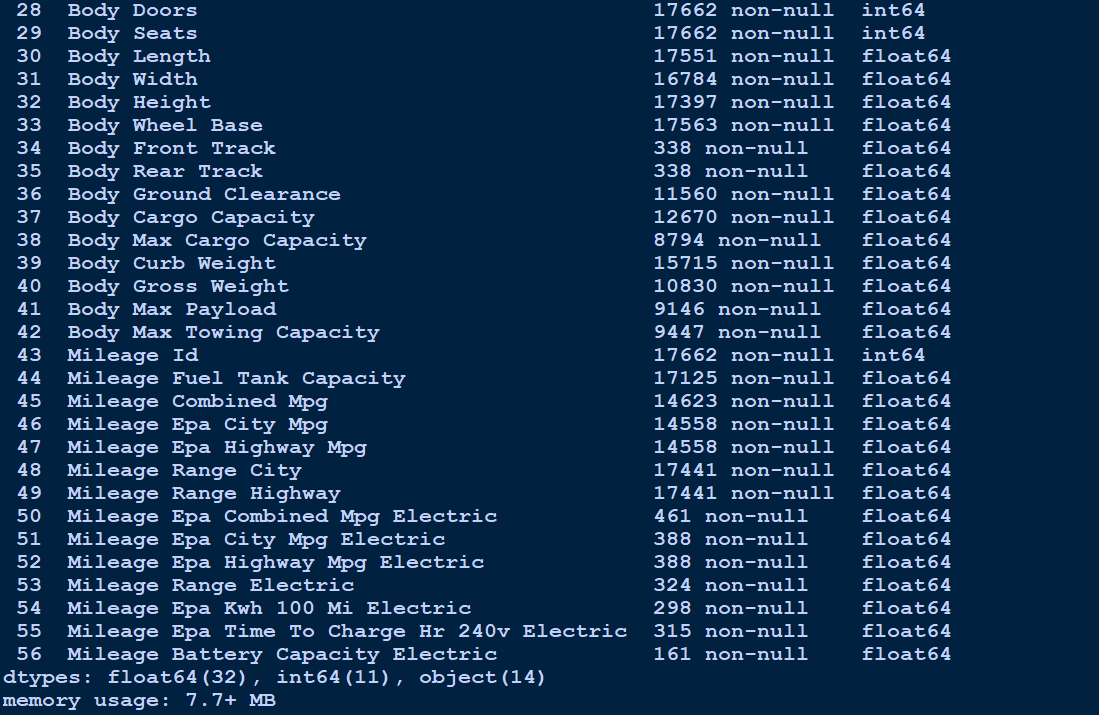
****

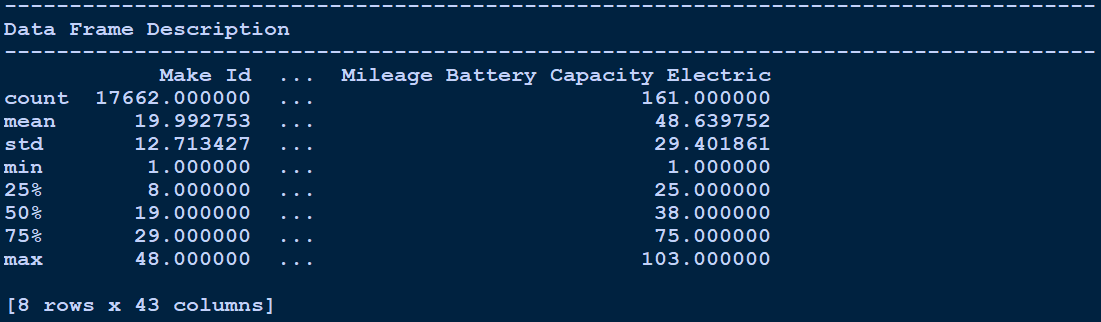
* **Functions and Formulas for Data Cleaning**
  + **Inspecting the Data**

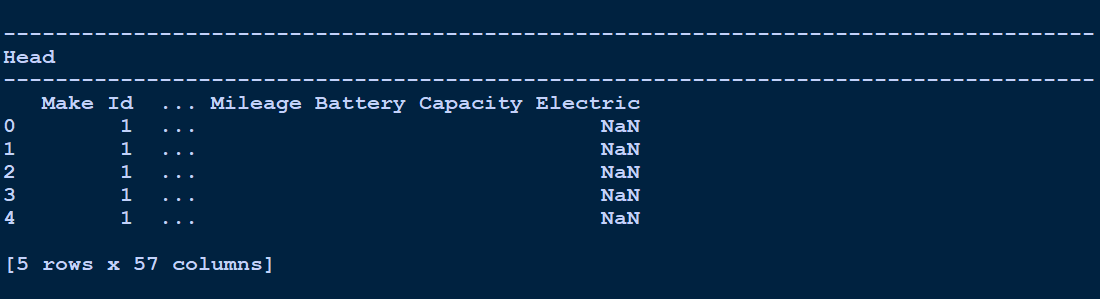
****

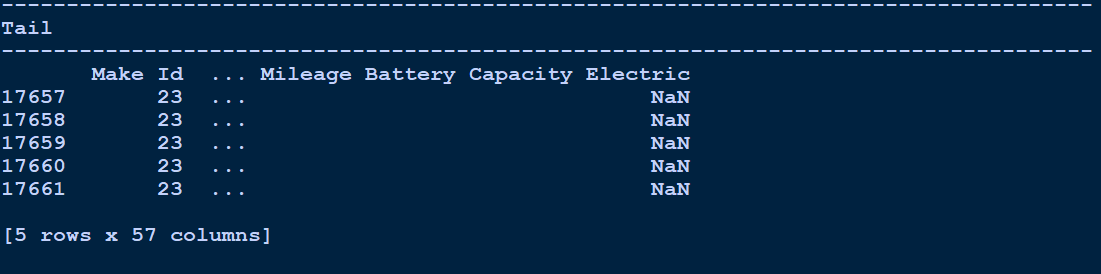
* **Benefits of Data Cleaning**
  + **Improves Data Quality: Ensures the dataset is accurate and consistent.**
  + **Enhances Analysis: Reduces noise and makes patterns and trends more apparent.**
  + **Increases Model Performance: Clean data leads to better machine learning model accuracy.**

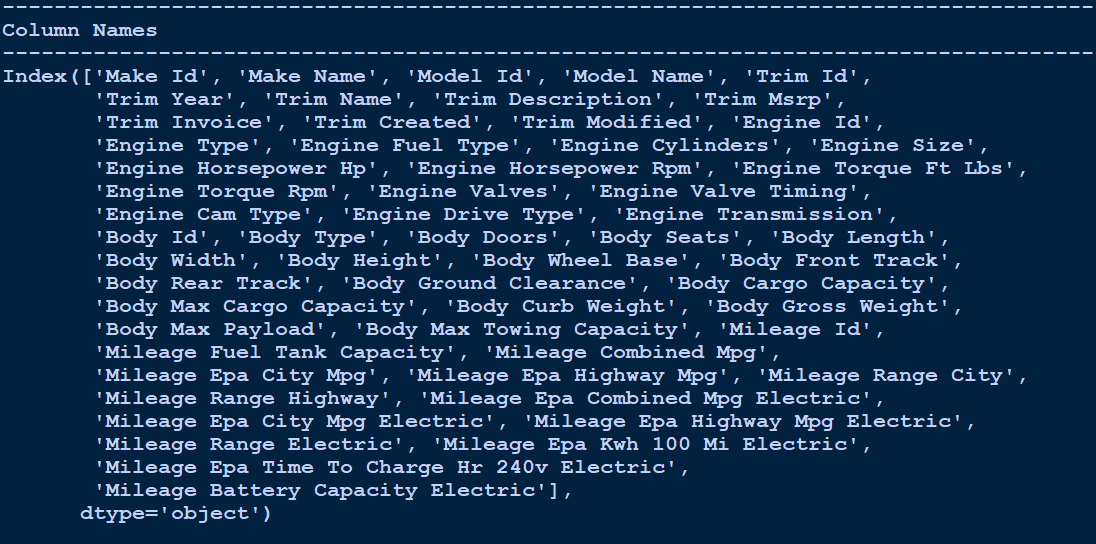
****

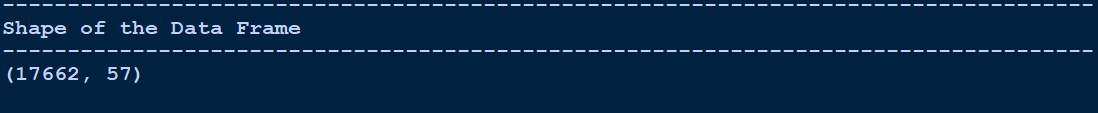
****

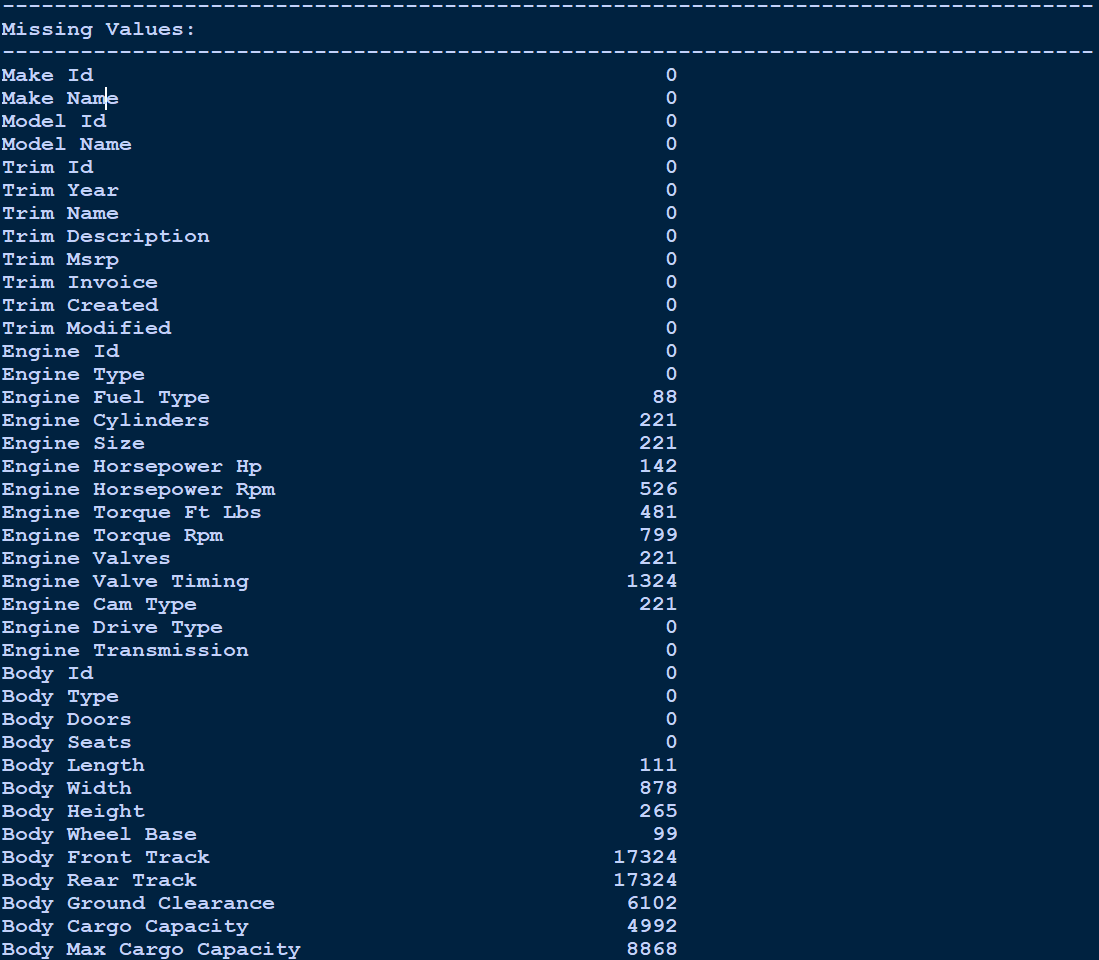
****

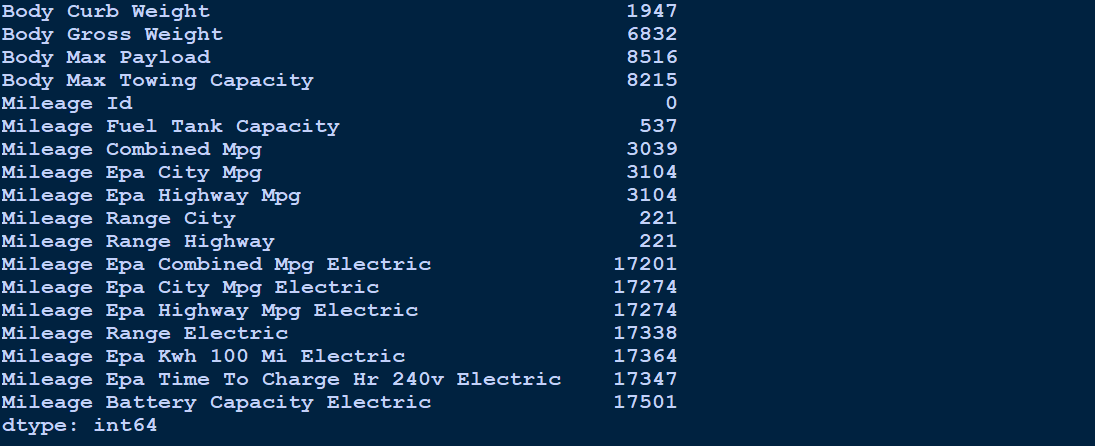
****

****

****

****

****

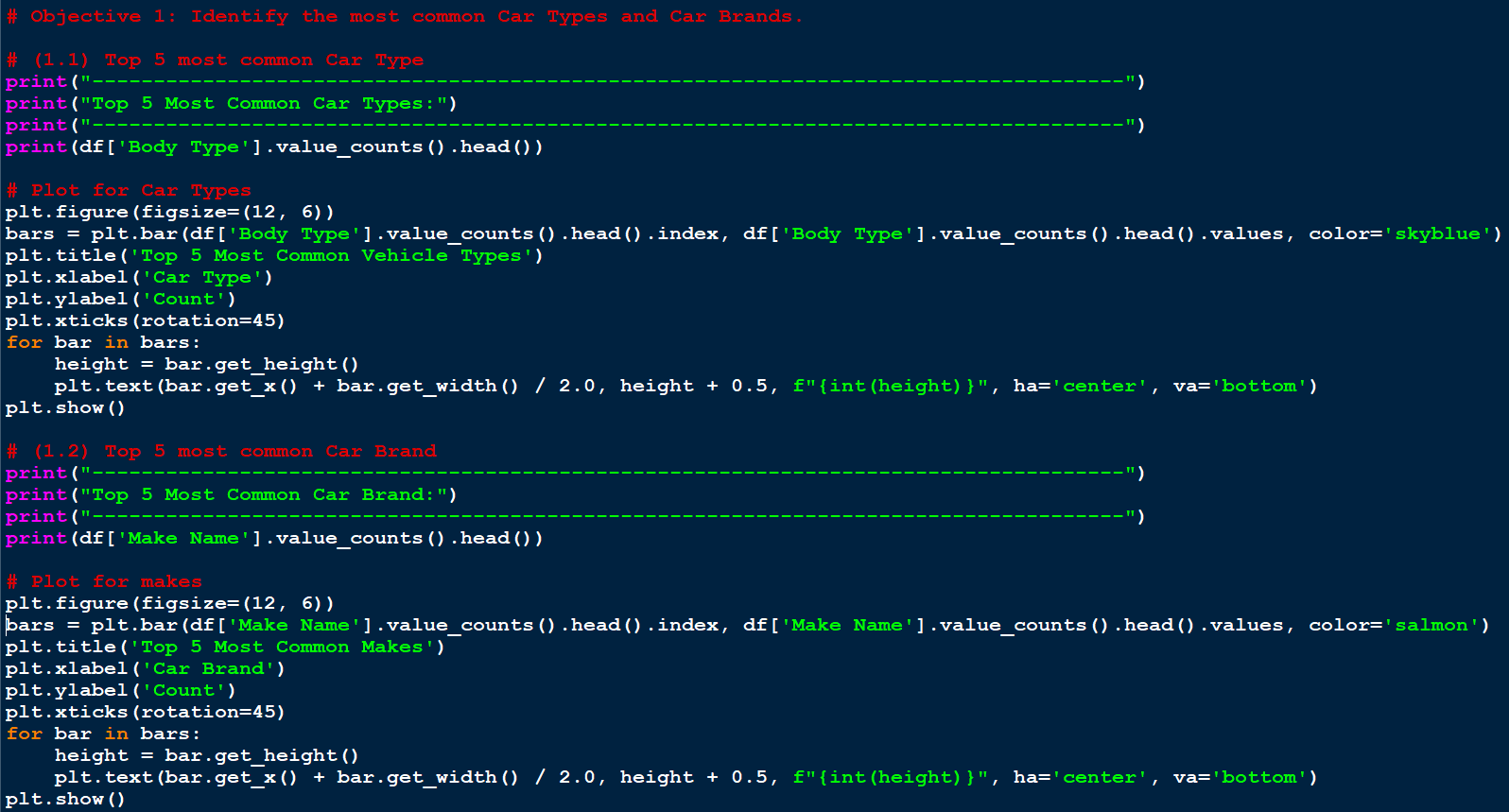
****

**# Objective 2: Identify the most common Car Types and Car Brands.**

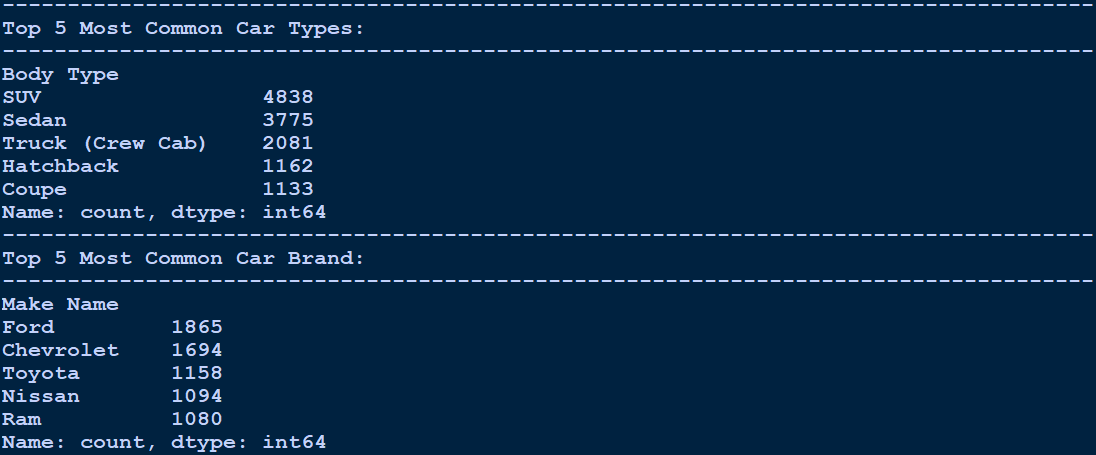
* **Introduction:**

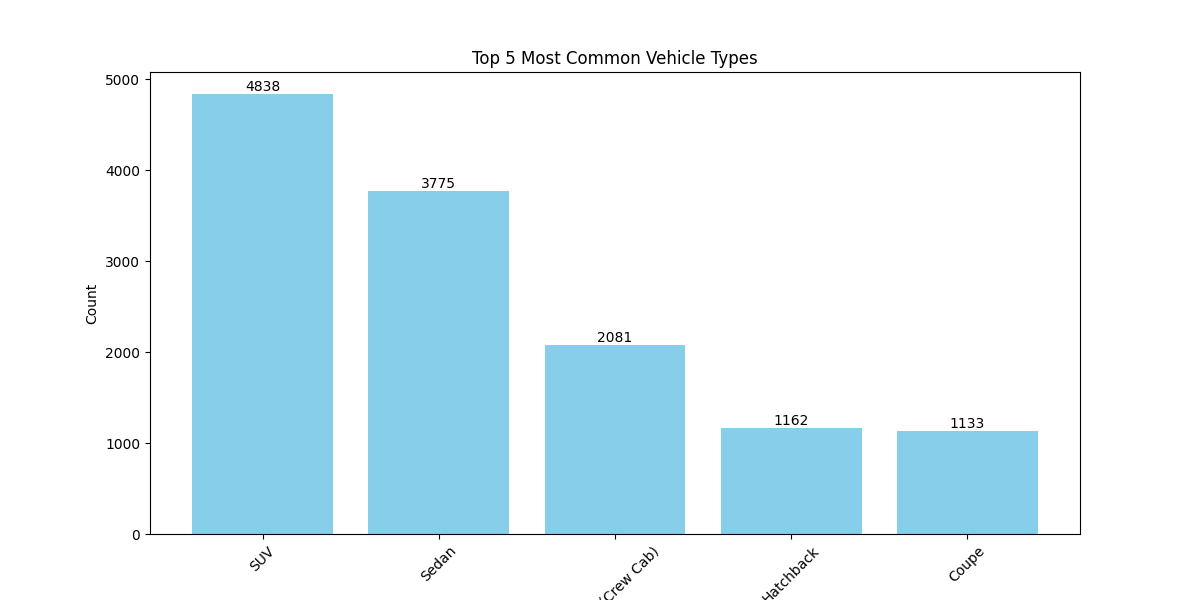
The objective is to identify and visualize the most common car types and car brands in the dataset. This information is crucial for understanding market trends and consumer preferences. By analysing the frequency of car types (e.g., SUV, Sedan) and car brands (e.g., Toyota, Ford), we can derive actionable insights into their popularity and potential market dominance.

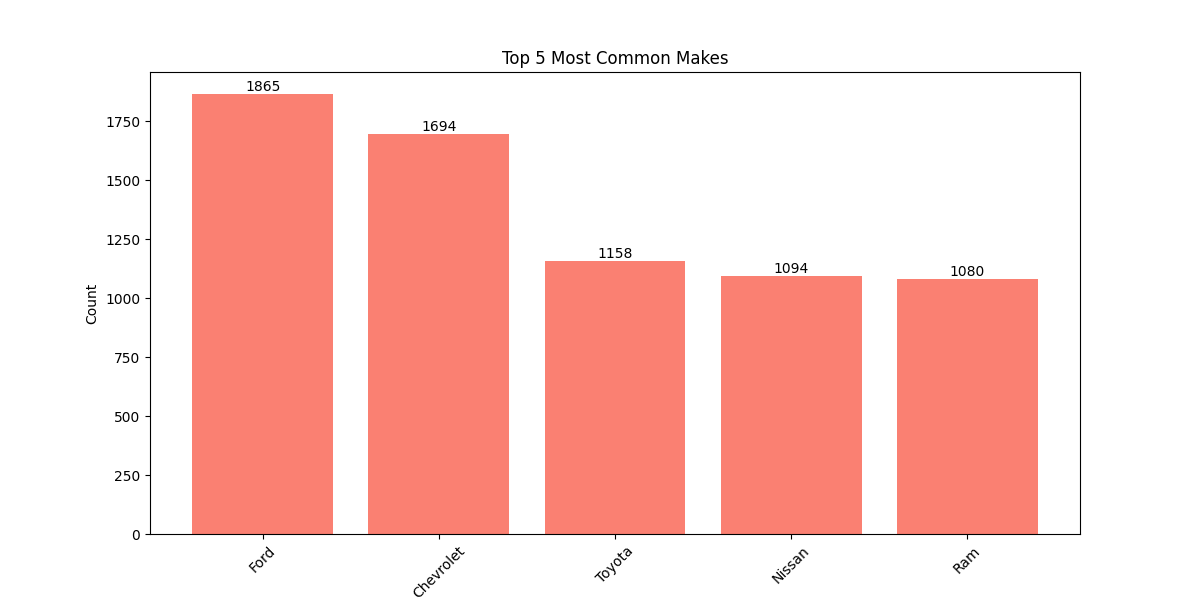
* **General Description:**
  + **Understanding Car Types and Brands:** Analyse the categorical columns, such as "Body Type" for vehicle types and "Make Name" for brands.
  + **Frequency Analysis:** Use the value\_counts() function to determine the counts for each unique value.
  + **Visualization:** Use bar plots to present the data visually, highlighting the top 5 car types and brands.
  + **Annotations:** Add data labels to the bar plots to enhance readability.
* **Specific Requirements**
  + **Dataset:** A Pandas Data Frame containing the columns:
  + **Body Type:** Represents the type of the vehicle (e.g., SUV, Sedan).
  + **Make Name:** Represents the brand of the vehicle (e.g., Toyota, Ford).
  + **Tools:**
    - Pandas for data manipulation.
    - Matplotlib for visualization.
  + **Code Output:**
    - Text summary of the top 5 car types and brands.
    - Bar plots showing the counts for these categories.
* **Functions and Formulas**
  + **Text Summary:**
    - df['column\_name'].value\_counts() provides the frequency of unique values.
  + **Bar Plot:**
    - Use plt.bar() to create bar plots for the top 5 categories.
    - Add labels using plt.text() for clarity.



* **Explanation of Code**
  + **Frequency Calculation:**
    - df['Body Type'].value\_counts().head() retrieves the top 5 most frequent car types.
    - Similarly, df['Make Name'].value\_counts().head() retrieves the top 5 car brands.
  + **Bar Plot Creation:**
    - plt.bar() creates a bar plot for each category.
    - bars.get\_height() and plt.text() annotate the bars with their counts.
  + **Customization:**
    - Added titles, labels, and gridlines for better readability.
    - Rotated x-axis labels for long category names.
* **Expected Outputs**
  + **Console Output:**
    - Text-based list of the top 5 car types and brands with their respective counts.



* + **Visualization:**
    - Two bar plots:
      * One for the top 5 car types.
      * Another for the top 5 car brands, with data labels showing counts.

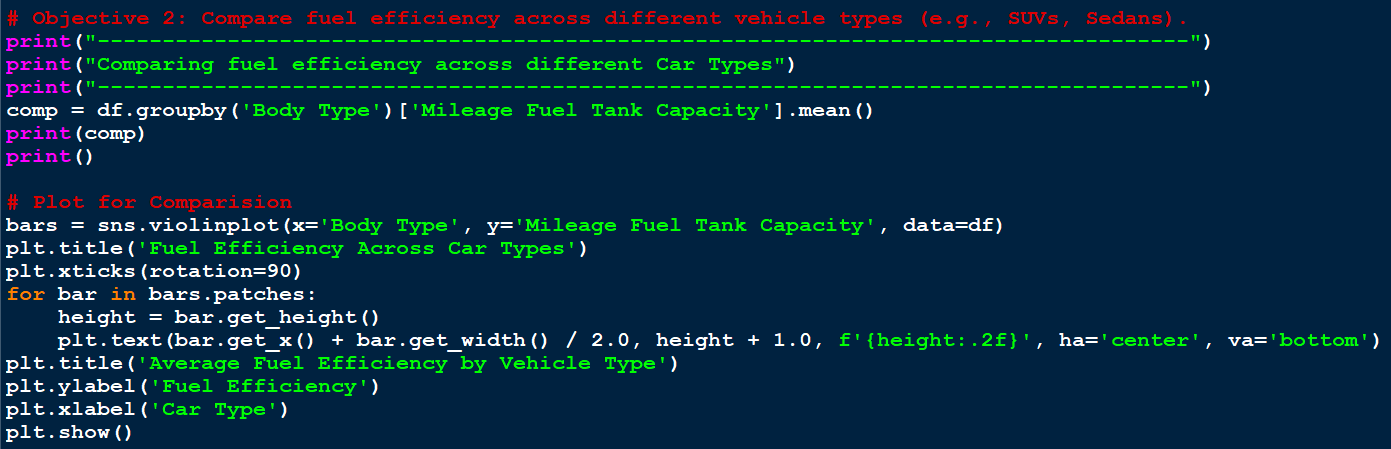


**# Objective 3: Compare fuel efficiency across different vehicle types (e.g., SUVs, Sedans).**

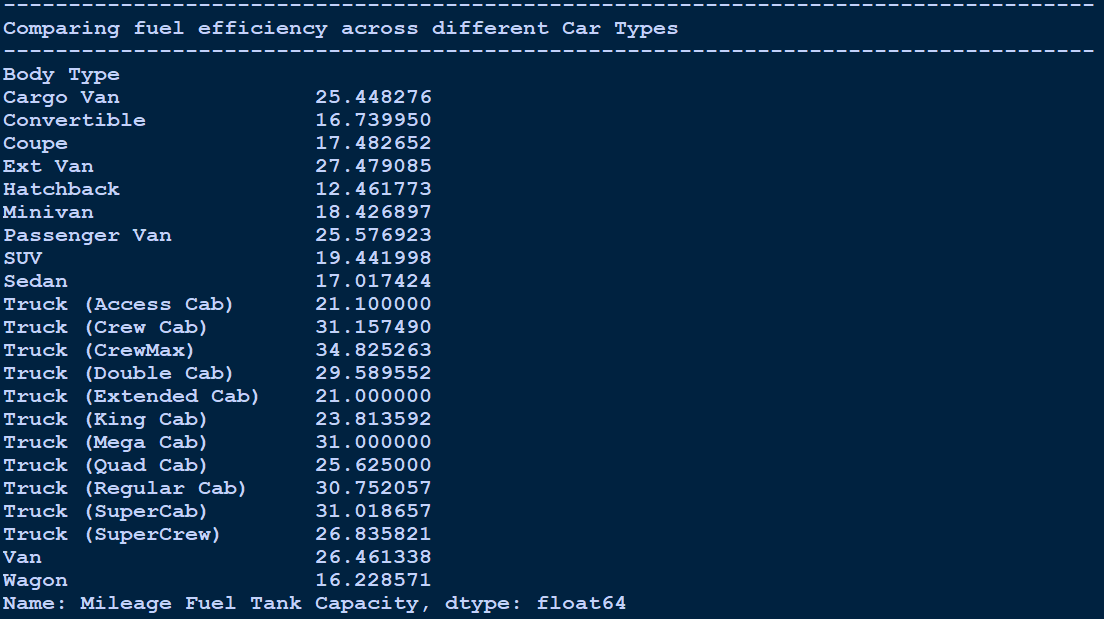
* **Introduction:**

Fuel efficiency is a critical factor in vehicle performance and customer decision-making. This objective aims to analyse and compare the fuel efficiency of various vehicle types, such as SUVs, sedans, and hatchbacks. By identifying trends in mileage across different categories, stakeholders can better understand which vehicle types offer the most efficient fuel consumption.

* **General Description**
  + **Grouping by Vehicle Type:**
    - Analyse the Body Type column to categorize vehicles.
    - Calculate the average fuel efficiency (Mileage Fuel Tank Capacity) for each type.
  + **Visualization:**
    - Use a violin plot to visualize the distribution of fuel efficiency for each vehicle type.
  + **Annotations:**
    - Include data labels to highlight specific values for better interpretability.
* **Specific Requirements**
  + **Dataset:** A Pandas Data Frame containing the columns:
  + **Body Type:** Represents the vehicle type.
  + **Mileage Fuel Tank Capacity:** Represents the fuel efficiency.
  + **Tools:**
    - Pandas for data grouping and aggregation.
    - Seaborn and Matplotlib for visualization.
  + **Code Output:**
    - A textual summary of the average fuel efficiency for each car type.
    - A violin plot showing the distribution of fuel efficiency across car types.
* **Functions and Formulas**
  + **Grouping Data:**
    - df.groupby('Body Type')['Mileage Fuel Tank Capacity'].mean() calculates the average mileage for each vehicle type.
  + **Visualization:**
    - Use sns.violinplot() to create a violin plot, which shows the distribution and density of fuel efficiency data.



* **Explanation of Code**
  + **Grouping and Aggregation:**
    - The groupby() method groups the dataset by Body Type, and .mean() calculates the average mileage for each type.
  + **Visualization:**
    - The violin plot shows the distribution and range of fuel efficiency for each vehicle type.
    - sns.violinplot() is used to represent both the average and variability of the data.
  + **Customization:**
    - Added labels, a title, and gridlines for clarity.
    - Rotated x-axis labels to prevent overlap for longer vehicle type names.
* **Expected Outputs**
  + **Console Output:**
    - A table of average fuel efficiency for each vehicle type, e.g.:

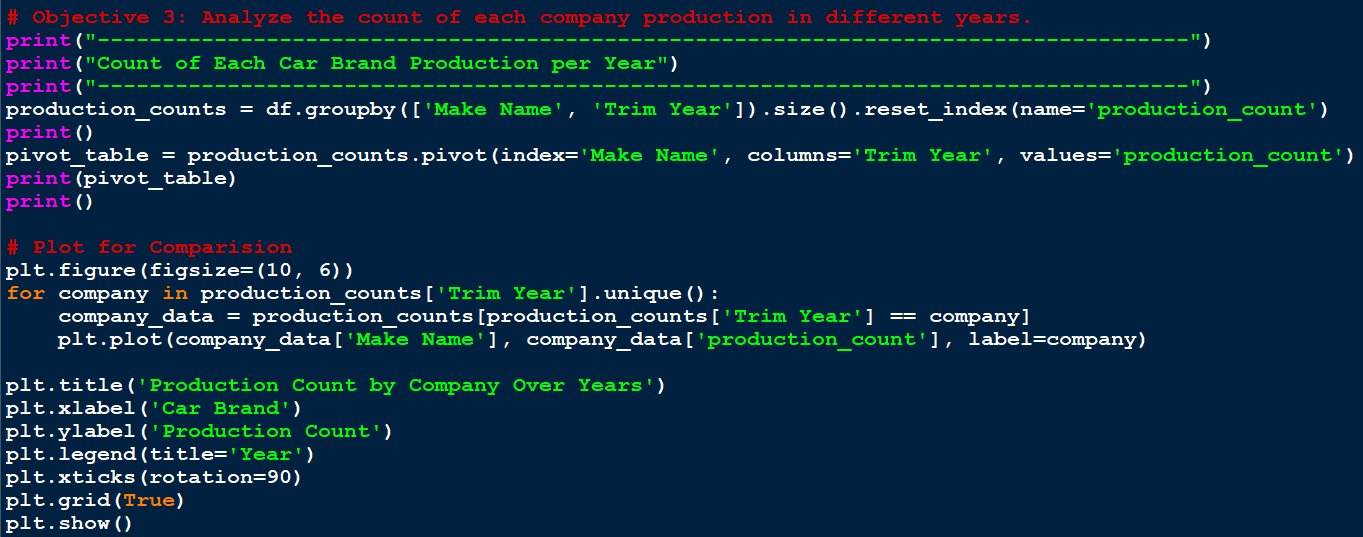
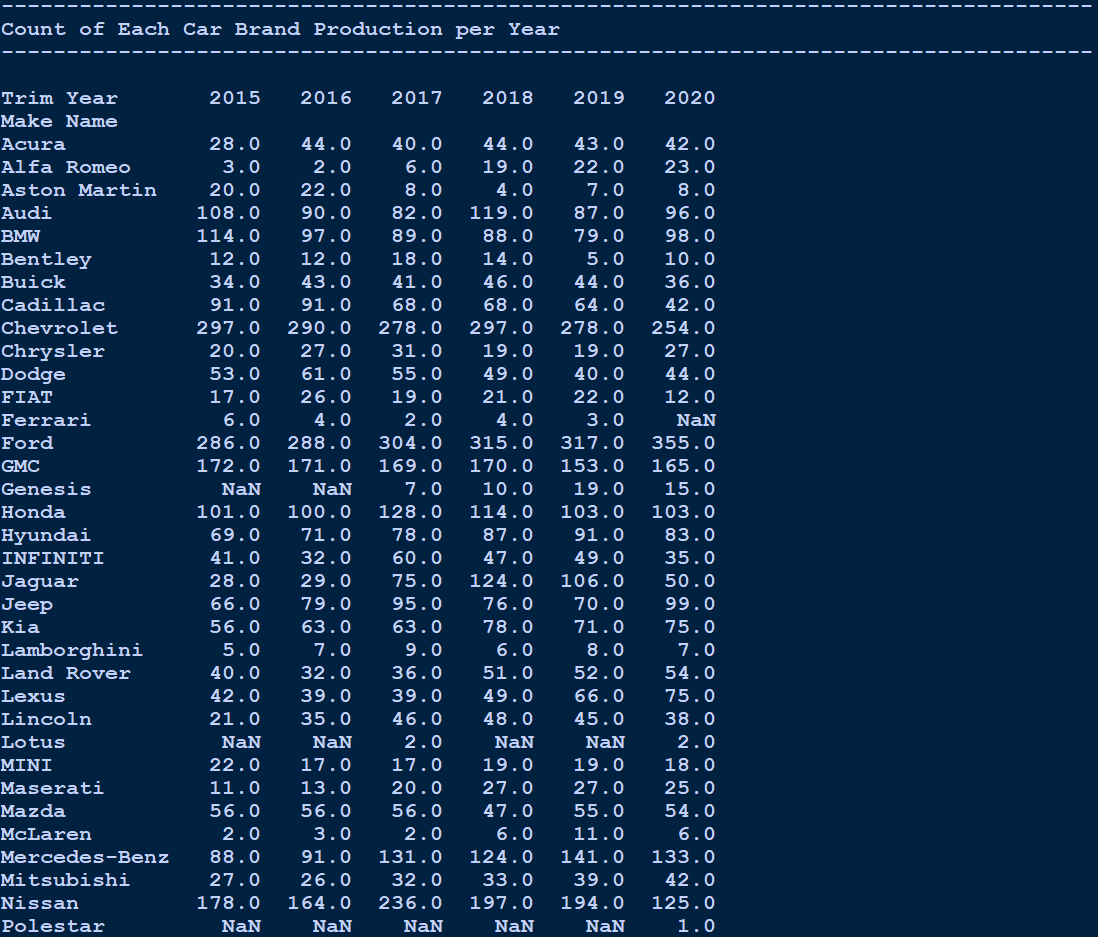


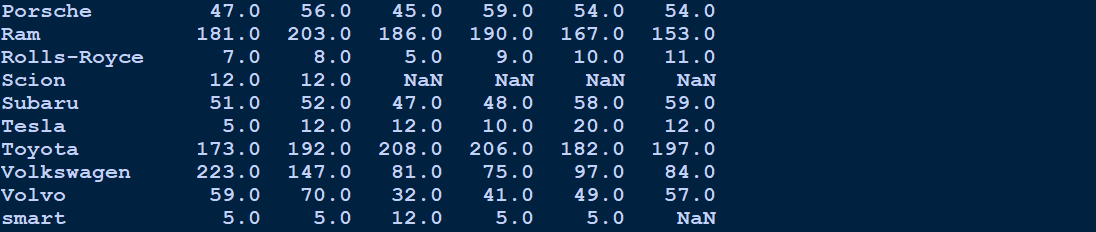
* **Visualization:**
  + - A violin plot illustrating the distribution of fuel efficiency across different car types. This includes the spread, density, and central tendency.
* **Benefits of Analysis**
  + Identifies which vehicle types offer higher fuel efficiency.
  + Highlights variability in fuel efficiency within each type.
  + Provides insights for manufacturers to target efficiency improvements in specific vehicle categories.

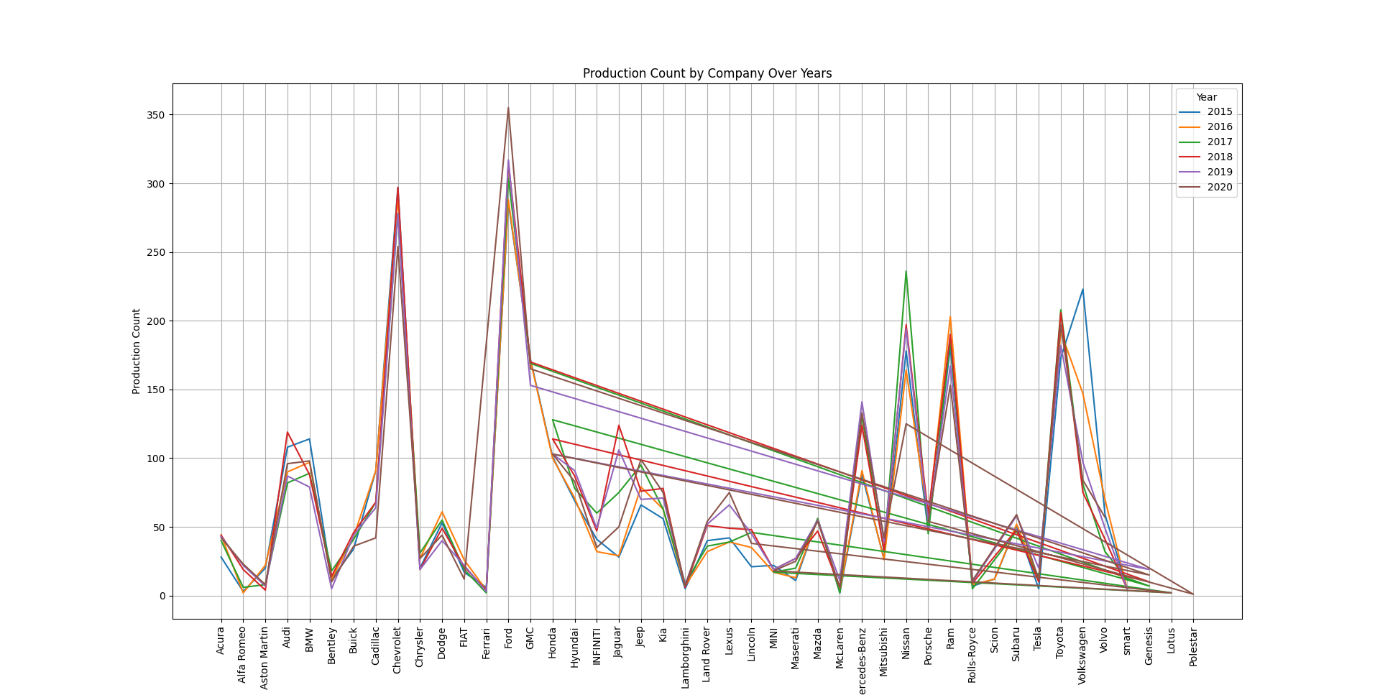
**# Objective 4: Analyze the count of each company production in different years.**

* **Introduction:**

Analyzing the production count of car brands across different years provides insights into manufacturing trends, market dominance, and year-over-year growth. This analysis can reveal production patterns for companies and help identify years of peak or reduced production.

* **General Description**
  + **Grouping and Aggregation:**
    - Group the dataset by Make Name (car brands) and Trim Year (production year).
    - Count the number of vehicles produced by each brand per year.
  + **Pivot Table:**
    - Use a pivot table for a structured view of production counts across years.
  + **Visualization:**
    - Create a line plot to visualize production trends for car brands over different years.
* **Specific Requirements**
  + **Dataset:**
    - **Columns required:**
      * **Make Name:** Represents car brands.
      * **Trim Year:** Represents production years.
  + **Tools:**
    - Pandas for grouping and pivoting.
    - Matplotlib for line plot visualization.
  + **Code Output:**
    - A pivot table showing production counts per year for each brand.
    - A line plot to visualize production trends.
* **Functions and Formulas**
  + **Grouping Data:**
    - df.groupby(['Make Name', 'Trim Year']).size() calculates the production count.
  + **Pivot Table:**
    - pivot() rearranges the data for easier readability.
  + **Visualization:**
    - Use plt.plot() to plot production trends for each company.
* **Explanation of Code:**
  + **Grouping and Counting:**
    - df.groupby(['Make Name', 'Trim Year']).size() groups the data by Make Name and Trim Year and counts the occurrences.
    - reset\_index() converts the grouped data into a DataFrame with a new column for production counts.
  + **Pivot Table:**
    - pivot() organizes the data into a tabular format where rows are Make Name, columns are Trim Year, and values are production counts.
  + **Visualization:**
    - A line plot is used to visualize the trends for each brand over the years.
    - Each line represents production counts for a specific year.
  + **Customization:**
    - Added a legend for years, rotated x-axis labels for readability, and included gridlines for clarity.
* **Expected Outputs**
  + **Console Output:**
    - A grouped table showing production counts for each brand in each year.
    - A pivot table, e.g.:



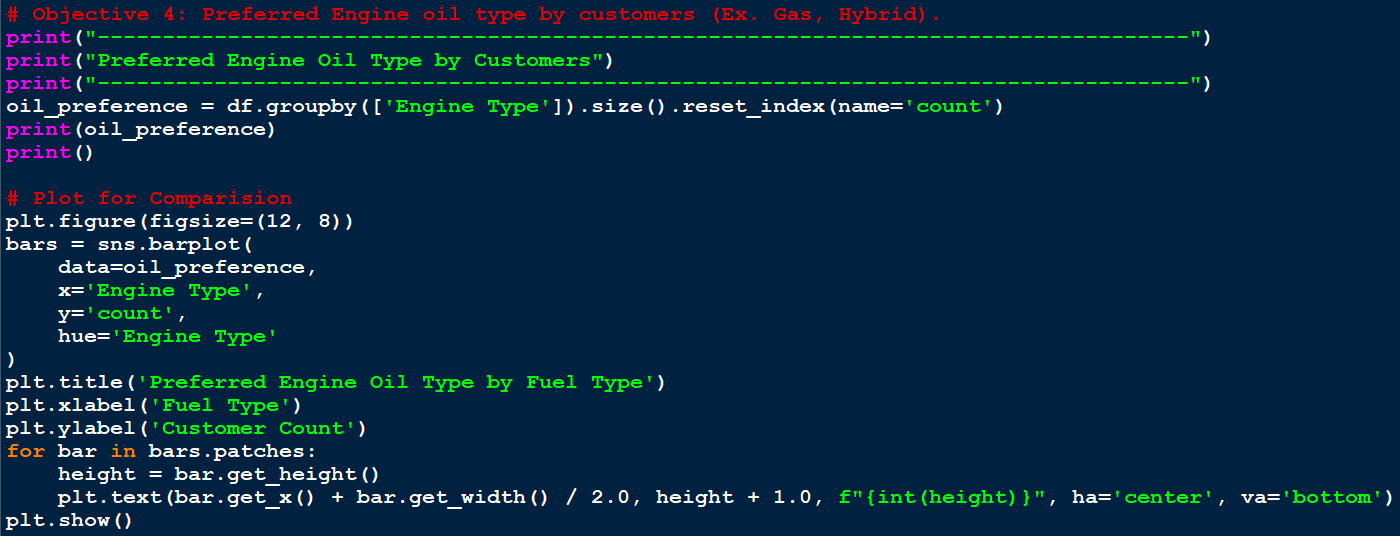
* **Visualization:**
  + A line plot where:
  + Each line represents a production year.
  + The x-axis lists car brands, and the y-axis shows production counts.
* **Benefits of Analysis**
  + Identifies production trends for each company.
  + Highlights peak production years.
  + Helps detect inconsistencies or anomalies in production over time.

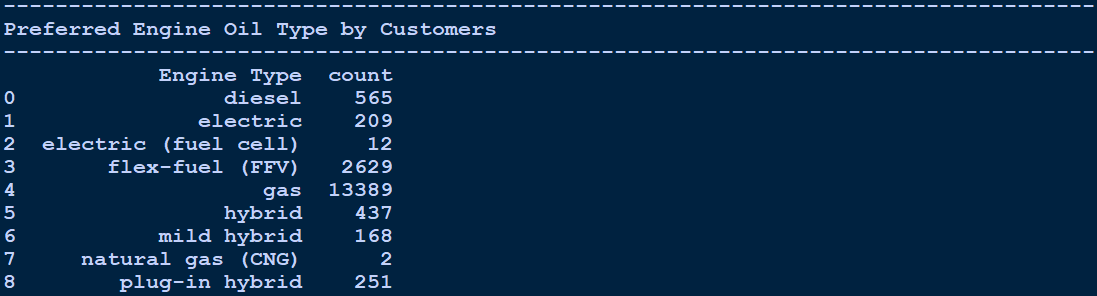
**# Objective 5: Preferred Engine oil type by customers (Ex. Gas, Hybrid).**

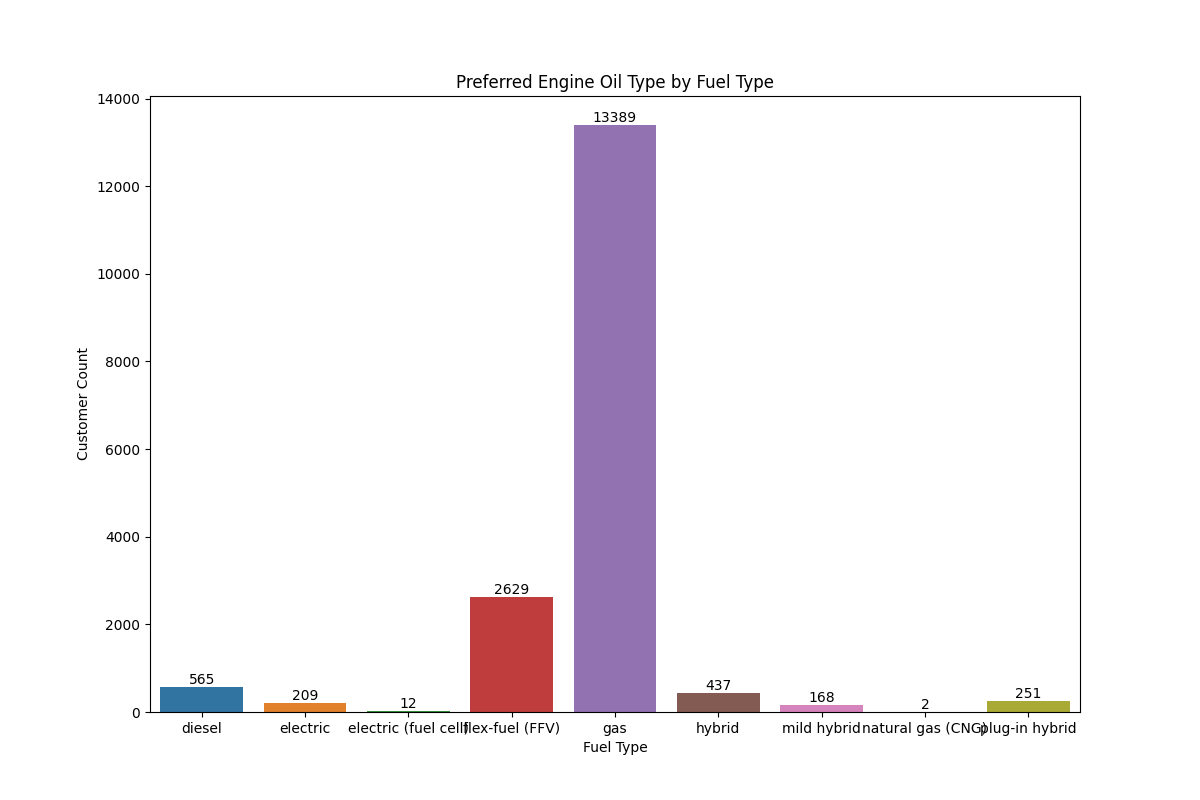
* **Introduction:**

Understanding the preferred engine oil types (e.g., gas, hybrid, diesel) helps manufacturers and distributors optimize their offerings based on customer preferences. This analysis identifies the popularity of each engine type among customers using the dataset.

* **General Description**
  + **Grouping and Counting:**
    - Analyze the Engine Type column to count the occurrences of each engine oil type.
  + **Visualization:**
    - Use a bar plot to display the counts for each engine type.
  + **Annotations:**
    - Add data labels to the bar plot for better readability.
* **Specific Requirements**
  + **Dataset:**
    - **Column required:** Engine Type, representing the type of engine oil (e.g., Gas, Diesel, Hybrid).
  + **Tools:**
    - Pandas for data aggregation.
    - Seaborn and Matplotlib for visualization.
  + **Code Output:**
    - A table showing counts for each engine type.
    - A bar plot to visualize the preferences.
* **Functions and Formulas**
  + **Grouping Data:**
    - df.groupby('Engine Type').size() calculates the count of each engine type.
  + **Visualization:**
    - Use sns.barplot() to create a bar plot for the engine types.

****

* **Explanation of Code**
  + **Grouping and Aggregation:**
    - df.groupby(['Engine Type']).size() groups the data by Engine Type and counts the number of occurrences.
  + **Visualization:**
    - A bar plot created with sns.barplot() displays the preferences for each engine type.
    - Data labels on each bar show the count values.
  + **Customization:**
    - Rotated x-axis labels for readability.
    - Added gridlines for clarity.
* **Expected Outputs**
  + **Console Output:**
    - A table showing the count of each engine type, e.g.:
* **Visualization:**
  + A bar plot where:
  + The x-axis represents engine types.
  + The y-axis shows the count of customers preferring each type.
  + Bars are annotated with their respective counts.



**Benefits of Analysis**

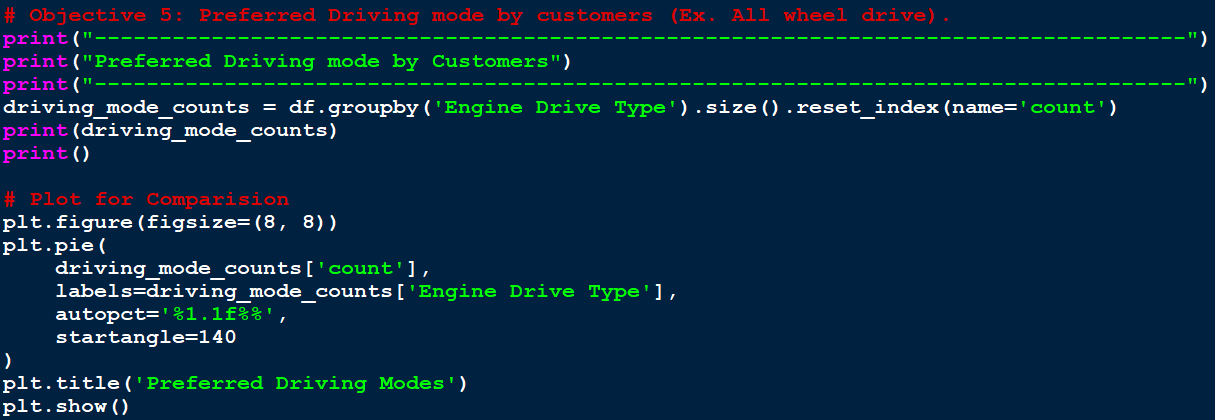
* Identifies the most preferred engine oil types.
* Highlights less popular types, offering scope for market exploration.
* Helps manufacturers align production and marketing strategies with customer preferences.

**# Objective 6: Preferred Driving mode by customers (Ex. All wheel drive).**

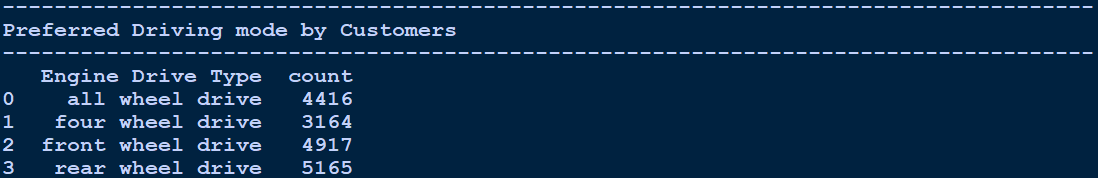
* **Introduction:**

Driving modes like All-Wheel Drive (AWD), Front-Wheel Drive (FWD), and Rear-Wheel Drive (RWD) are significant preferences that reflect customer priorities such as performance, terrain adaptability, and fuel efficiency. This analysis identifies the popularity of driving modes among customers.

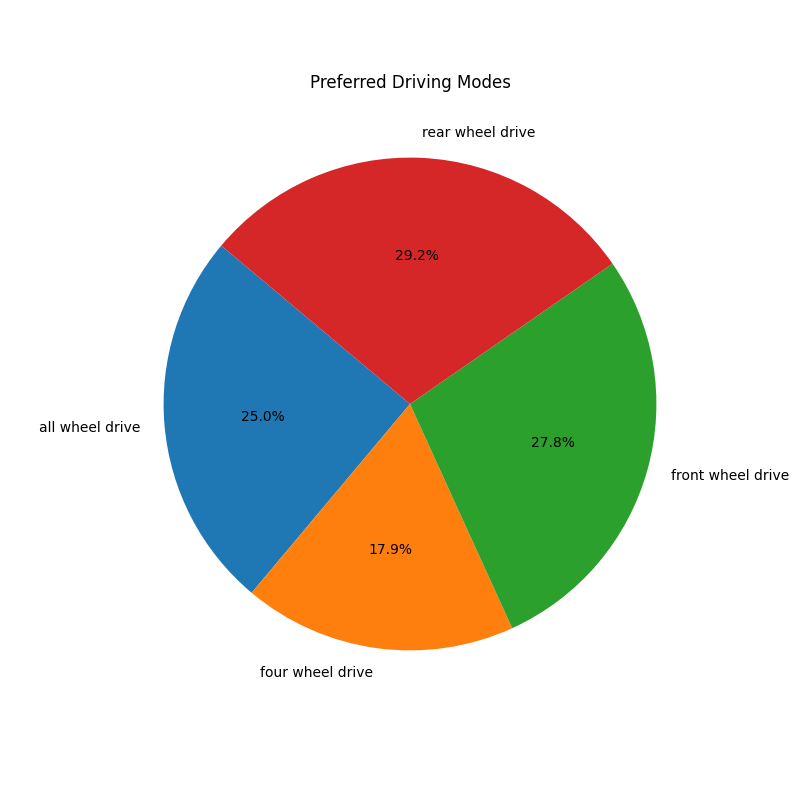
* **General Description**
  + **Grouping and Counting:**
    - Analyze the Engine Drive Type column to count occurrences of each driving mode.
  + **Visualization:**
    - Use a pie chart to display the percentage distribution of driving modes.
  + **Annotations:**
    - Display percentage labels on the pie chart for better interpretability.
* **Specific Requirements**
  + **Dataset:**
    - Column required: Engine Drive Type, representing driving modes (e.g., AWD, FWD, RWD).
  + **Tools:**
    - Pandas for data aggregation.
    - Matplotlib for visualization.
  + **Code Output:**
    - A table showing counts for each driving mode.
    - A pie chart to visualize the distribution.
* **Functions and Formulas**
  + **Grouping Data:**
    - df.groupby('Engine Drive Type').size() calculates the count of each driving mode.
  + **Visualization:**
    - Use plt.pie() to create a pie chart.

****

* **Explanation of Code**
  + **Grouping and Aggregation:**
    - df.groupby('Engine Drive Type').size() groups the data by Engine Drive Type and counts the number of occurrences.
  + **Visualization:**
    - A pie chart created with plt.pie() displays the percentage distribution of each driving mode.
  + **Customization:**
    - Added percentage labels to the pie chart for clarity.
    - Used startangle=140 to adjust the starting angle for better alignment.
* **Expected Outputs**
  + **Console Output:**
    - A table showing the count of each driving mode, e.g.:



* + **Visualization:**
    - A pie chart where:
    - Each slice represents a driving mode.
    - The size of the slice corresponds to the count.
    - Percentage values are displayed on each slice.



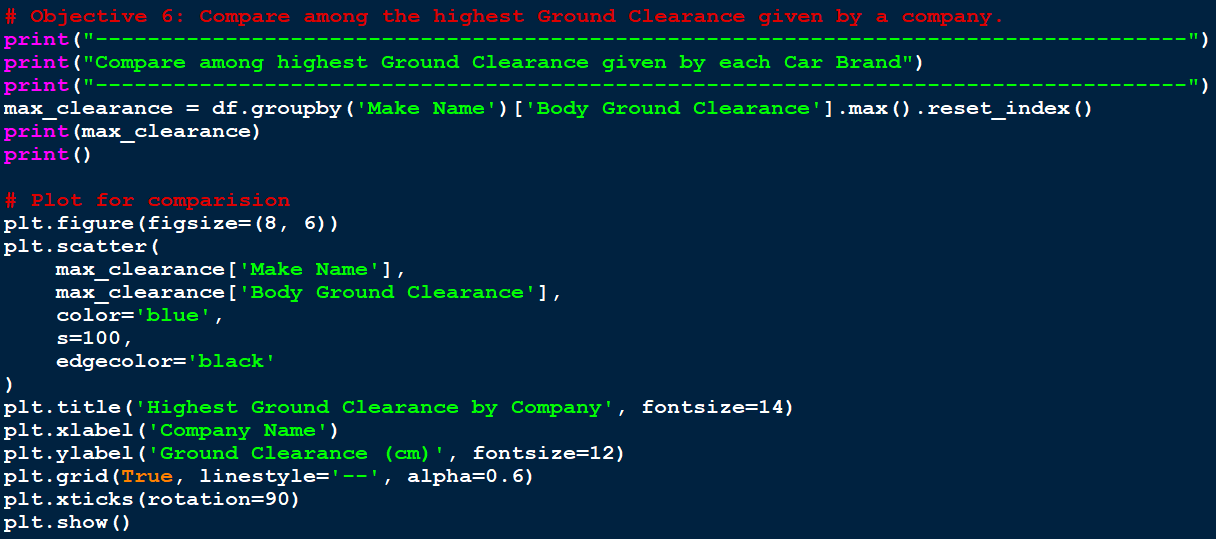
* **Benefits of Analysis**
  + Identifies the most and least preferred driving modes.
  + Provides insights into market demand for different drivetrain configurations.
  + Helps manufacturers tailor product offerings to customer preferences.

**# Objective 7: Compare among the highest Ground Clearance given by a company.**

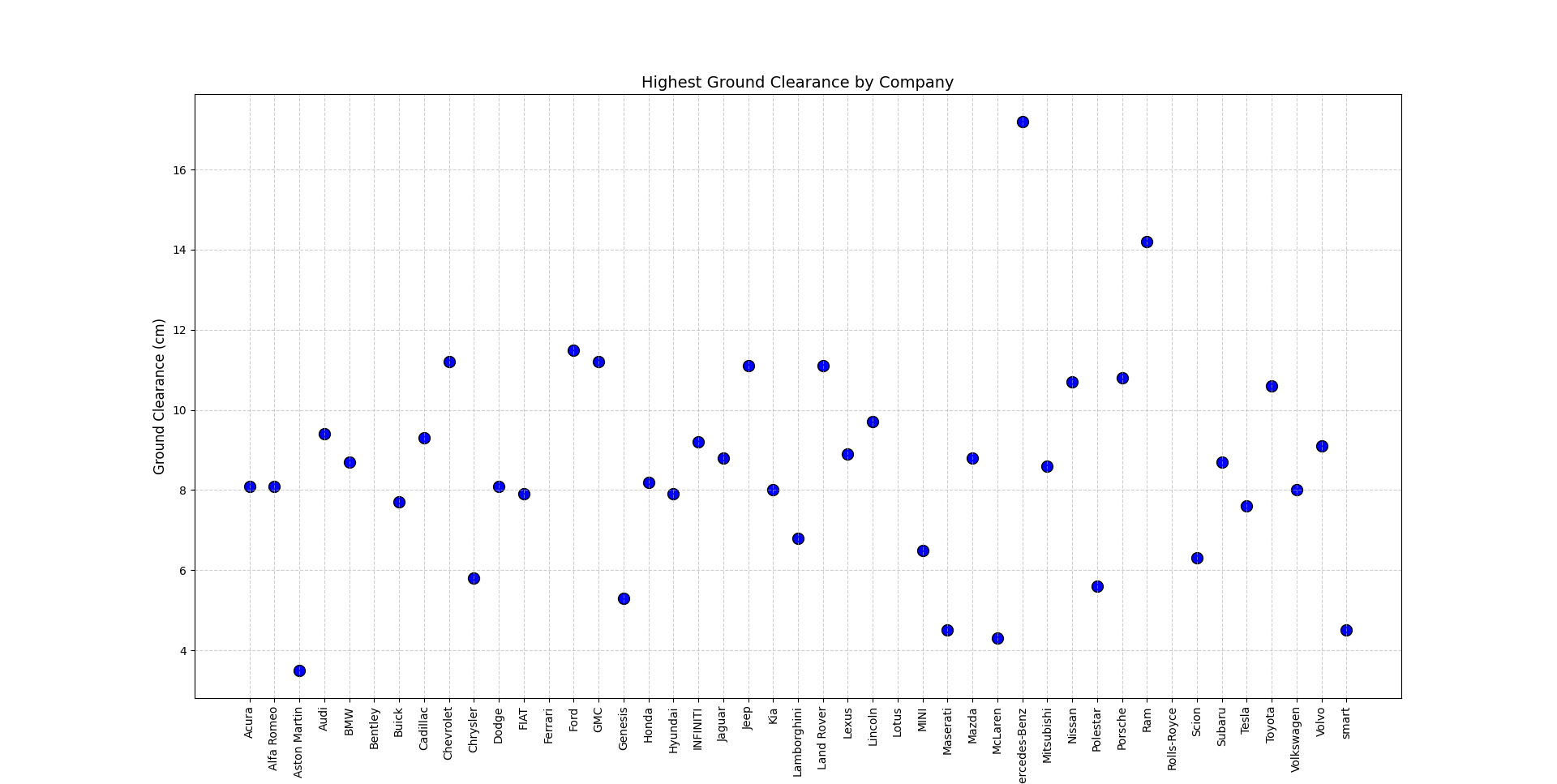
* **Introduction:**

Ground clearance is an essential specification for vehicles, particularly for those meant for off-road and rugged terrain. Comparing the highest ground clearance provided by each car manufacturer offers insights into their market strategies and product diversity.

* **General Description**
  + **Data Handling:**
    - Handle missing values in the Body Ground Clearance column.
    - Calculate the maximum ground clearance for each company using group aggregation.
  + **Visualization:**
    - Use a scatter plot to compare maximum ground clearance values for different companies.
  + **Annotations:**
    - Ensure axis labels and gridlines improve the chart's readability.
* **Specific Requirements**
  + **Dataset:**
    - **Column required:** Make Name for company names and Body Ground Clearance for clearance values.
  + **Tools:**
    - Pandas for data handling and aggregation.
    - Matplotlib for visualization.
  + **Code Output:**
    - A table showing the highest ground clearance values for each company.
    - A scatter plot to visualize the comparison.
* **Functions and Formulas**
  + **Handling Missing Values:**
    - Replace NaN in Body Ground Clearance with 0 using fillna(0).
  + **Grouping Data:**
    - Use df.groupby('Make Name')['Body Ground Clearance'].max() to find the maximum value for each company.
  + **Visualization:**
    - Use plt.scatter() to create a scatter plot.

****

* **Explanation of Code**
  + **Data Cleaning:**
    - Missing values in Body Ground Clearance are replaced with 0 to avoid computation errors.
  + **Grouping and Aggregation:**
    - Maximum ground clearance for each company is calculated using groupby and max().
  + **Visualization:**
    - A scatter plot is created to visualize the highest ground clearance provided by each company.
    - Customizations include gridlines, rotated labels for readability, and larger point sizes for clarity.
* **Expected Outputs**
  + **Console Output:**
    - A table showing the highest ground clearance values for each car brand, e.g.:
  + **Visualisation:**
    - A scatter plot where:
    - The x-axis represents car brands.
    - The y-axis represents ground clearance values.
    - Each dot corresponds to a brand’s maximum ground clearance.



* **Benefits of Analysis**
  + Highlights which companies prioritize high ground clearance.
  + Helps identify manufacturers that cater to off-road vehicle markets.
  + Provides comparative insights into the design priorities of different brands.

**CONCLUSION**

The data analysis project using Exploratory Data Analysis (EDA) techniques has provided valuable insights into vehicle specifications, customer preferences, and company strategies based on the dataset of car details. This project utilized a structured approach to analyze key features such as fuel efficiency, car types, ground clearance, engine oil preferences, and driving modes, offering a comprehensive understanding of the automotive dataset.

**Brief Description of the Project**

This project aimed to explore and analyze car details using the EDA process. The main objectives were:

1. **Identify the most common car types and brands:** Understanding market dominance.
2. **Compare fuel efficiency across vehicle types:** Evaluating performance and economy.
3. **Analyze production trends by year and company:** Studying manufacturing patterns.
4. **Identify preferred engine oil types and driving modes:** Highlighting customer preferences.
5. **Compare ground clearance among brands:** Assessing terrain adaptability and design.

The dataset underwent **data cleaning** to handle missing values and inconsistencies. **Visualization techniques** such as bar plots, scatter plots, pie charts, and violin plots were employed to present findings in an intuitive manner. **Python libraries** like Pandas, Matplotlib, and Seaborn were instrumental in performing these analyses.

This project demonstrated the power of EDA in uncovering actionable insights from raw data. The findings can guide stakeholders, including manufacturers, distributors, and customers, in making informed decisions. The combination of data cleaning, statistical analysis, and visualization ensures that the results are robust and reliable, offering a solid foundation for further exploration in the automotive domain.

**FUTURE SCOPES**

1. **Enhanced Predictive Analytics**
   * Incorporating advanced machine learning models with the cleaned and analyzed data can predict trends such as vehicle sales, customer preferences, and market demands.
   * Predicting fuel efficiency or maintenance costs based on specific vehicle attributes.
2. **Customer Behavior Insights**
   * Detailed segmentation of customer preferences using clustering algorithms.
   * Analyzing regional and demographic preferences for car types, engine oil, and driving modes.
3. **Optimized Vehicle Design**
   * Using data to recommend optimal specifications for future vehicle designs, including ground clearance, fuel efficiency, and drivetrain configurations.
   * Tailoring designs for niche markets like off-road vehicles or urban compact cars.
4. **Integration of Environmental Factors**
   * Analyzing the environmental impact of various vehicle types (e.g., emissions and fuel consumption) to guide eco-friendly automotive strategies.
   * Identifying trends in electric and hybrid vehicle adoption.
5. **Real-Time Data Analysis**
   * Expanding the scope to include real-time data from IoT-connected vehicles for dynamic analysis of performance metrics, usage patterns, and predictive maintenance.
6. **Market Expansion Analysis**
   * Utilizing data to explore untapped markets by comparing production trends and customer preferences globally.
   * Identifying opportunities for growth in emerging markets or developing countries.
7. **Competitive Analysis**
   * Comparing automotive companies' strategies using multi-company datasets to identify market leaders and potential gaps.
   * Analyzing competitor benchmarks for pricing, features, and performance.
8. **Integration with Emerging Technologies**
   * Combining data with autonomous vehicle technology insights to optimize design and performance for self-driving cars.
   * Analyzing the impact of connected and smart vehicle features on customer satisfaction.
9. **Policy Development Support**
   * Assisting policymakers by analyzing the impact of regulations on vehicle specifications, production trends, and customer behavior.
   * Providing insights into the adoption of sustainable technologies, such as EVs or biofuels.
10. **Extending Beyond Automotive**
    * Applying the EDA framework to related industries, such as aviation, railways, and logistics, to analyze operational efficiency and customer preferences.

**REFERENCES**

**Technical Resources**

1. **Python Libraries:**
   * **Pandas Documentation:** <https://pandas.pydata.org/docs/>
   * **Matplotlib Documentation:** <https://matplotlib.org/stable/contents.html>
   * **Seaborn Documentation:** <https://seaborn.pydata.org/>
2. **Concepts of EDA:**
   * Tutorials and guidelines were referred to from [Towards Data Science](https://towardsdatascience.com/) and [DataCamp](https://www.datacamp.com/).
   * Exploratory Data Analysis theories: "An Introduction to EDA" by John Tukey.
3. **Visualization Techniques:**
   * Resources for creating advanced plots were referred to from [Python Graph Gallery](https://www.python-graph-gallery.com/).
   * Best practices for data visualization: "Storytelling with Data" by Cole Nussbaumer Knaflic.

**Acknowledged Contributions**

1. **Peer Discussions:**
   * Insights and suggestions from peers and mentors.
   * Collaborative brainstorming on objectives and methodologies.
2. **Online Tutorials:**
   * FreeCodeCamp: <https://www.freecodecamp.org/>
   * GeeksforGeeks: <https://www.geeksforgeeks.org/>