Experiment 1

Aim: Introduction to Data science and Data preparation using Pandas steps.

Theory:

Data science is the study of data that helps us derive useful insight for business decision making. Data Science is all about using tools, techniques, and creativity to uncover insights hidden within data. It combines math, computer science, and domain expertise to tackle real-world challenges in a variety of fields.

Data science involves these key steps:

- **Data Collection:** Gathering raw data from various sources, such as databases, sensors, or user interactions.
- Data Cleaning: Ensuring the data is accurate, complete, and ready for analysis.
- Data Analysis: Applying statistical and computational methods to identify patterns, trends, or relationships.
- **Data Visualization:** Creating charts, graphs, and dashboards to present findings clearly.
- Decision-Making: Using insights to inform strategies, create solutions, or predict outcomes.

Dataset Overview:

The dataset consists of 21 columns, each providing valuable insights into vehicle details, registration data, and vehicle types across different locations. Below is a breakdown of the dataset's columns and their significance:

- **ID**: A unique identifier assigned to each vehicle, distinguishing them in the inventory.
- **Plate Type**: Represents the type of plate assigned to the vehicle, reflecting its registration or classification.
- **Primary Customer City**: The city where the primary customer resides, giving insights into regional distribution.
- **Primary Customer State**: The state where the primary customer resides, which helps in analyzing geographic patterns.
- **Registration Start Date**: The start date of the vehicle's registration, indicating the time frame for its use.
- Registration Expiration Date: The expiration date of the vehicle's registration, helping track the validity of the registration.

 Registration Usage: Represents the usage category of the vehicle's registration, such as regular or commercial use.

- **Vehicle Type**: The classification of the vehicle, such as Motorcycle or Passenger, giving insights into vehicle segmentation.
- Vehicle Weight: The weight of the vehicle, which could be relevant for logistics and fuel efficiency considerations.
- **Vehicle Year**: The year the vehicle was manufactured, which is crucial for understanding its age and market trends.
- **Vehicle Make**: The manufacturer or brand of the vehicle, which is vital for brand-specific market analysis.
- **Vehicle Model**: The specific model of the vehicle, helping identify product features and target consumer segments.
- **Vehicle Body**: Describes the type of body the vehicle has, such as Sedan (SD) or SUV, aiding in vehicle categorization.
- **Primary Color**: The primary color of the vehicle, which can influence customer preference and aesthetics.
- **Vehicle Declared Gross Weight**: The gross weight of the vehicle as declared by the manufacturer, important for regulatory and logistical purposes.
- **Fuel Code**: Represents the fuel type used by the vehicle, such as Electric (E00) or Hybrid (H04), helping to analyze energy consumption patterns.
- **Vehicle Recorded GVWR**: The Gross Vehicle Weight Rating (GVWR) recorded for the vehicle, a key measure for vehicle classification.
- **Vehicle Name**: The official name or model name of the vehicle, assisting in product-specific analysis.
- **Type**: The vehicle's classification type, such as BEV (Battery Electric Vehicle) or PHEV (Plug-in Hybrid Electric Vehicle), influencing environmental analysis.
- **Vehicle Category**: The classification of the vehicle based on its size and usage, such as Light-Duty (Class 1-2).

Problem Statement:

The dataset provides detailed information about various vehicles, including their registration, attributes, and classifications. The primary objectives of analyzing this dataset are:

- Vehicle Performance: Identifying which vehicle types, makes, and models perform better in terms of registration and customer preferences.
- **Customer Insights**: Understanding the impact of customer location (city and state) on vehicle preferences and registrations.

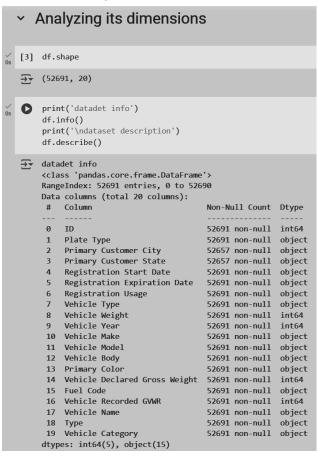
• **Fuel Type Analysis**: Investigating how different fuel types (BEV, PHEV, etc.) influence vehicle demand and registration trends.

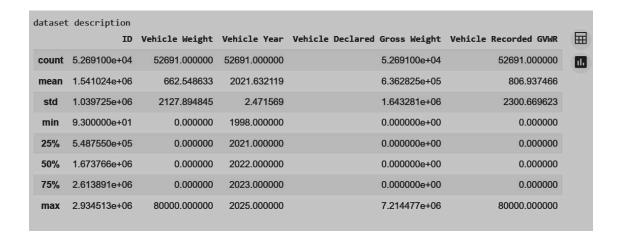
- Pricing and Brand Impact: Analyzing how vehicles make, model, and type correlate with pricing strategies and customer behavior.
- Sales and Registration Trends: Exploring how vehicle year and type affect registration volume over time, helping in forecasting and inventory management.

By processing and analyzing this dataset, the goal is to uncover trends that can assist in strategic decisions related to vehicle marketing, inventory, and customer targeting.

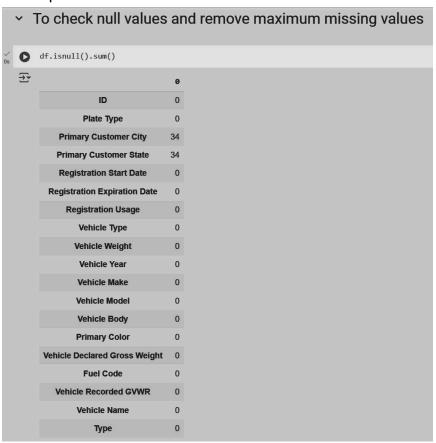
Code:

- 1] This returns a tuple indicating the number of rows and columns in the DataFrame. This code prints "dataset info" and displays the DataFrame's structure including data types and non-null counts using df.info().
- 2] It then prints "dataset description" and shows summary statistics like mean, standard deviation, and percentiles for numerical columns with df.describe().





3] This code checks each column in the DataFrame for missing values and sums them up.



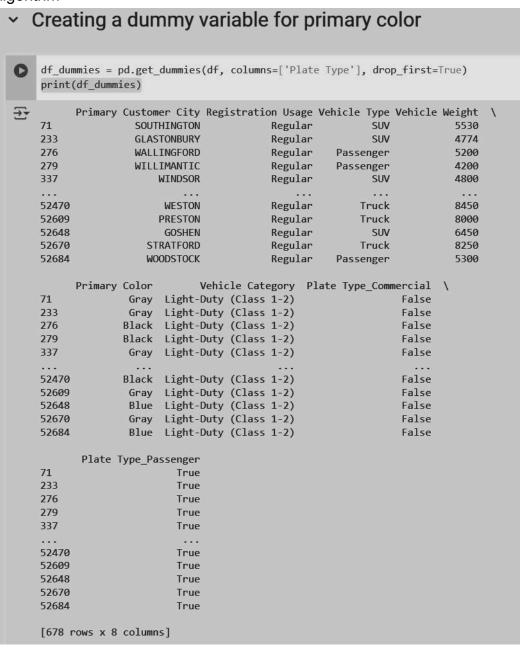
4] This code replaces zeros in the DataFrame with NA values, then removes all rows containing any missing data to create a cleaned DataFrame. It prints the cleaned DataFrame and confirms no missing values remain by summing NA entries for each column.

```
df.replace(0, pd.NA, inplace=True)
    df cleaned = df.dropna()
    print(df cleaned.isna().sum())
→ Plate Type
                                0
    Primary Customer City
                                0
    Registration Usage
                                0
    Vehicle Type
                                0
    Vehicle Weight
    Primary Color
                                0
    Vehicle Category
    Z Score
                                0
    Vehicle Weight Normalized
    dtype: int64
```

This code removes duplicate rows from the cleaned DataFrame.



5] This code creates dummy data out of plate type as commercial and passenger. This helps to convert categorical data to numerical data and helps in analysis in the algorithm



6] The head () method to display the first ten rows of the DataFrame, to identify outliers manually we use the standardization approach (z score method). We find mean and standard deviation of the vehicle weight and calculate its z score; if its less than -3 or greater than 3 means its an outlier.

df.head(10)							
	Plate Type	Primary Customer City	Registration Usage	Vehicle Type	Vehicle Weight	Primary Color	Vehicle Category
71	Passenger	SOUTHINGTON	Regular	SUV	5530	Gray	Light-Duty (Class 1-2
233	Passenger	GLASTONBURY	Regular	SUV	4774	Gray	Light-Duty (Class 1-2
276	Passenger	WALLINGFORD	Regular	Passenger	5200	Black	Light-Duty (Class 1-2
279	Passenger	WILLIMANTIC	Regular	Passenger	4200	Black	Light-Duty (Class 1-2
337	Passenger	WINDSOR	Regular	SUV	4800	Gray	Light-Duty (Class 1-2
416	Passenger	BRISTOL	Regular	SUV	5530	Silver	Light-Duty (Class 1-2
420	Passenger	WEST SIMSBURY	Regular	Passenger	5600	Blue	Light-Duty (Class 1-2
616	Passenger	NEW HAVEN	Regular	Passenger	3840	Black	Light-Duty (Class 1-2
782	Passenger	GLASTONBURY	Regular	SUV	4800	Silver	Light-Duty (Class 1-2
857	Passenger	GUILFORD	Regular	SUV	4709	Red	Light-Duty (Class 1-2

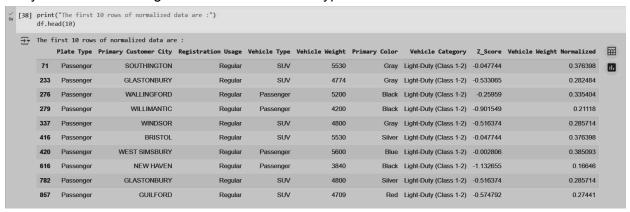
```
#By Z-score method
 mean_vehicle_weight = df['Vehicle Weight'].mean()
 std_vehicle_weight = df['Vehicle Weight'].std()
 print(f"Mean of Vehicle Weight: {mean_vehicle_weight}")
 print(f"Standard Deviation of Vehicle Weight: {std_vehicle_weight}")
 # Calculate the Z-score for each vehicle weight
 df['Z_Score'] = (df['Vehicle Weight'] - mean_vehicle_weight) / std_vehicle_weight
 print(df[['Vehicle Weight', 'Z_Score']])
 # Identify outliers based on the Z-score
 outliers = df[df['Z_Score'].abs() > 3]
 print(outliers)
 Mean of Vehicle Weight: 5604.371681415929
 Standard Deviation of Vehicle Weight: 1557.7315042063165
       Vehicle Weight Z_Score
                 5530 -0.047744
 71
                4774 -0.533065
 233
               5200 -0.25959
 276
               4200 -0.901549
 279
                4800 -0.516374
 337
 52470
              8450 1.826777
52609 8000 1.537896
52648 6450 0.542859
52670 8250 1.698385
52684 5300 -0.195394
 52684
               5300 -0.195394
 [678 rows x 2 columns]
        Plate Type Primary Customer City Registration Usage Vehicle Type \
 15639 Combination THOMASTON Combination
 23230 Combination
                                                 Combination
                               FAIRFIELD
                                                                     Truck
 32996 Combination
                             WILLIMANTIC
                                                 Combination
                                                                      Van
                                              Combination
 50047 Combination
                                                                   Truck
                               WINDSOR
                             y Color Vehicle Category Z_Score
White Light-Duty (Class 1-2) 3.174891
White Light-Duty (Class 1-2) 3.174891
       Vehicle Weight Primary Color
               10550
 15639
                10550
 23230
 32996
                10360
                             Orange Medium-Duty (Class 3-6) 3.052919
                             White Medium-Duty (Class 3-6) 3.142793
 50047
                10500
```

7] We normalize the data across the vehicle weights on a scale of 0 to 1.

```
    Normalization

   # Min-Max Normalization
    min_vehicle_weight = df['Vehicle Weight'].min()
    max_vehicle_weight = df['Vehicle Weight'].max()
     # Apply normalization
    df['Vehicle Weight Normalized'] = (df['Vehicle Weight'] - min_vehicle_weight) / (max_vehicle_weight - min_vehicle_weight)
    print(df[['Vehicle Weight', 'Vehicle Weight Normalized']])
₹
          Vehicle Weight Vehicle Weight Normalized
             5530 0.376398
    233
                                      0.282484
                                     0.335404
    276
                  5200
                 4200
4800
    279
                                       0.21118
                                     0.285714
    337
                                     0.73913
    52470
                8450
                 8000
    52609
                                      0.68323
    52648
                   6450
                                      0.490683
    52670
                  8250
                                      0.714286
                                      0.347826
    [678 rows x 2 columns]
```

8] This is our normalized data with respect to vehicle weight and can be used to analyse the vehicle weight distribution across its type and color.



Conclusion:

Thus we have pre-processed our dataset by various techniques mentioned and can be used for analysis and trained under algorithms for predictions.