In [1]:
```
import pandas as pd
import numpy as np
```

loading the data set

In [12]:
```
file_path = r"C:\Users\Dell\Documents\Sujal\DSPS\car_price_dataset.csv"
```

Read the CSV file into a DataFrame

In [14]:
```
df = pd.read_csv(file_path)
```

In [15]:
```
total_records = df.shape[0]
print(f"Total number of records in the dataset: {total_records}")
```

```
Total number of records in the dataset: 10000
```

View the first n rows of the DataFrame

In [16]:
```
print(df.head(n=5))
```

```
        Brand   Model  Year  Engine_Size Fuel_Type    Transmission  Mileage  
\
0         Kia     Rio  2020          4.2    Diesel          Manual   289944   
1   Chevrolet  Malibu  2012          2.0    Hybrid       Automatic     5356   
2    Mercedes     GLA  2020          4.2    Diesel       Automatic   231440   
3        Audi      Q5  2023          2.0  Electric          Manual   160971   
4  Volkswagen    Golf  2003          2.6    Hybrid  Semi-Automatic   286618   

   Doors  Owner_Count  Price  
0      3            5   8501  
1      2            3  12092  
2      4            2  11171  
3      2            1  11780  
4      3            3   2867  
```

View the last n rows of the DataFrame

In [17]:
```
print(df.tail(n=5))
```

```
           Brand     Model  Year  Engine_Size Fuel_Type    Transmission  \
9995         Kia    Optima  2004          3.7    Diesel  Semi-Automatic   
9996   Chevrolet    Impala  2002          1.4  Electric       Automatic   
9997         BMW  3 Series  2010          3.0    Petrol       Automatic   
9998        Ford  Explorer  2002          1.4    Hybrid       Automatic   
9999  Volkswagen    Tiguan  2001          2.1    Diesel          Manual   

      Mileage  Doors  Owner_Count  Price  
9995     5794      2            4   8884  
9996   168000      2            1   6240  
9997    86664      5            1   9866  
9998   225772      4            1   4084  
9999   157882      3            3   3342  
```

View the DataFrame info (datatypes, memory usage, etc.)

In [18]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Brand         10000 non-null  object 
 1   Model         10000 non-null  object 
 2   Year          10000 non-null  int64  
 3   Engine_Size   10000 non-null  float64
 4   Fuel_Type     10000 non-null  object 
 5   Transmission  10000 non-null  object 
 6   Mileage       10000 non-null  int64  
 7   Doors         10000 non-null  int64  
 8   Owner_Count   10000 non-null  int64  
 9   Price         10000 non-null  int64  
dtypes: float64(1), int64(5), object(4)
memory usage: 781.4+ KB
None
```

View summary statistics of numerical columns

In [20]:

```
print(df.describe())
```

```
               Year   Engine_Size        Mileage         Doors   Owner_Count  \
count  10000.000000  10000.000000   10000.000000  10000.000000  10000.000000   
mean    2011.543700      3.000560  149239.111800      3.497100      2.991100   
std        6.897699      1.149324   86322.348957      1.110097      1.422682   
min     2000.000000      1.000000      25.000000      2.000000      1.000000   
25%     2006.000000      2.000000   74649.250000      3.000000      2.000000   
50%     2012.000000      3.000000  149587.000000      3.000000      3.000000   
75%     2017.000000      4.000000  223577.500000      4.000000      4.000000   
max     2023.000000      5.000000  299947.000000      5.000000      5.000000   

              Price  
count  10000.00000  
mean    8852.96440  
std     3112.59681  
min     2000.00000  
25%     6646.00000  
50%     8858.50000  
75%    11086.50000  
max    18301.00000  
```

View the shape of the DataFrame (rows, columns)

In [21]:

```
print(df.shape)
```

```
(10000, 10)
```

View the column names

In [22]:

```
print(df.columns)
```

```
Index(['Brand', 'Model', 'Year', 'Engine_Size', 'Fuel_Type', 'Transmission',
       'Mileage', 'Doors', 'Owner_Count', 'Price'],
      dtype='object')
```

view the index of the dataframe

In [24]:

```
print(df.index)
```

```
RangeIndex(start=0, stop=10000, step=1)
```

Select a single column

In [25]:

```
df["Brand"]
```

Out[25]:

```
0              Kia
1        Chevrolet
2         Mercedes
3             Audi
4       Volkswagen
           ...    
9995           Kia
9996     Chevrolet
9997           BMW
9998          Ford
9999    Volkswagen
Name: Brand, Length: 10000, dtype: object
```

select multiple rows and columns

In [30]:

```
print(df[['Model', 'Year']])
```

```
         Model  Year
0          Rio  2020
1       Malibu  2012
2          GLA  2020
3           Q5  2023
4         Golf  2003
...        ...   ...
9995    Optima  2004
9996    Impala  2002
9997  3 Series  2010
9998  Explorer  2002
9999    Tiguan  2001

[10000 rows x 2 columns]
```

Select specific rows and columns

In [31]:

```
df.loc[2, 'Brand']
```

Out[31]:

```
'Mercedes'
```

Select by condition (Boolean indexing)

In [35]:

```
df['Year'] > 5
```

Out[35]:

```
0       True
1       True
2       True
3       True
4       True
        ... 
9995    True
9996    True
9997    True
9998    True
9999    True
Name: Year, Length: 10000, dtype: bool
```

Adding index filed it will create column of row numbers

In [37]:

```
df["IndexFilld"] = np.arange(len(df))
print(df)
```

```
           Brand     Model  Year  Engine_Size Fuel_Type    Transmission  \
0            Kia       Rio  2020          4.2    Diesel          Manual   
1      Chevrolet    Malibu  2012          2.0    Hybrid       Automatic   
2       Mercedes       GLA  2020          4.2    Diesel       Automatic   
3           Audi        Q5  2023          2.0  Electric          Manual   
4     Volkswagen      Golf  2003          2.6    Hybrid  Semi-Automatic   
...          ...       ...   ...          ...       ...             ...   
9995         Kia    Optima  2004          3.7    Diesel  Semi-Automatic   
9996   Chevrolet    Impala  2002          1.4  Electric       Automatic   
9997         BMW  3 Series  2010          3.0    Petrol       Automatic   
9998        Ford  Explorer  2002          1.4    Hybrid       Automatic   
9999  Volkswagen    Tiguan  2001          2.1    Diesel          Manual   

      Mileage  Doors  Owner_Count  Price  IndexFilld  
0      289944      3            5   8501           0  
1        5356      2            3  12092           1  
2      231440      4            2  11171           2  
3      160971      2            1  11780           3  
4      286618      3            3   2867           4  
...       ...    ...          ...    ...         ...  
9995     5794      2            4   8884        9995  
9996   168000      2            1   6240        9996  
9997    86664      5            1   9866        9997  
9998   225772      4            1   4084        9998  
9999   157882      3            3   3342        9999  

[10000 rows x 11 columns]
```

In [ ]: