

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly as plt

In [11]:
google_data=pd.read_csv("C:/Users/HP/Downloads/cars_engage_2022.csv")

In [17]:
google_data.head(10)

Out[17]:
   Unnamed: 0  Make  Model  Variant  Ex-Showroom_Price  Displacement  Cylinders  Valves_Per_Cylinder  Drivetrain  Cylinder_Configuration  ...  Leather_Wrapped  Steering  Automatic_Headlamps  Engine_Type  ASR_/Traction_Control
0            0      Tata  Nano  GenX  Xi  Rs. 2,90,867  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
1            1      Tata  Nano  GenX  Xe  Rs. 2,36,447  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
2            2      Tata  Nano  GenX  Emix  GenX  Rs. 2,96,661  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
3            3      Tata  Nano  GenX  Xta  Rs. 3,34,766  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
4            4      Tata  Nano  GenX  Xmi  Rs. 2,72,223  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
5            5      Tata  Nano  GenX  Xma  Rs. 3,34,815  624 cc  2.0  2.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
6            6  Datsun  Redi-Go  D  Rs. 2,79,850  799 cc  3.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
7            7  Datsun  Redi-Go  T  Rs. 3,51,832  799 cc  3.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
8            8  Datsun  Redi-Go  A  Rs. 3,33,419  799 cc  3.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
9            9  Datsun  Redi-Go  S  Rs. 3,02,000  799 cc  3.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN

10 rows x 141 columns

In [13]:
type(google_data)
pandas.core.frame.DataFrame

In [14]:
google_data.shape
(1276, 141)

In [16]:
google_data.describe()

Out[16]:
Unnamed: 0      0  Cylinders  Valves_Per_Cylinder  Doors  Seating_Capacity  Number_of_Airbags  USB_Ports
count  1276.000000  1210.000000  1174.000000  1272.000000  1270.000000  1141.000000  29.000000
mean      637.500000  4.369992  3.977853  4.550314  5.270079  3.787009  1.793103
std       588.897861  1.323050  0.832763  0.777768  1.140241  0.522399  0.778642
min         0.000000  2.000000  1.000000  1.000000  2.000000  2.000000  1.000000
25%      318.750000  4.000000  4.000000  4.000000  5.000000  3.000000  1.000000
50%      637.500000  4.000000  4.000000  5.000000  5.000000  2.000000  2.000000
75%      956.250000  4.000000  4.000000  5.000000  5.000000  6.000000  2.000000
max     1275.000000  16.000000  16.000000  5.000000  16.000000  14.000000  3.000000

In [18]:
google_data.tail(10)

Out[18]:
   Unnamed: 0  Make  Model  Variant  Ex-Showroom_Price  Displacement  Cylinders  Valves_Per_Cylinder  Drivetrain  Cylinder_Configuration  ...  Leather_Wrapped  Steering  Automatic_Headlamps  Engine_Type  ASR_/Traction_Control
1266      1266  Honda  City  Vt  Petrol  Rs. 12,65,500  1497 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
1267      1267  Honda  City  Vt  Petrol  Rs. 13,82,000  1497 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  Yes  NaN  NaN  NaN  NaN
1268      1268  Honda  City  Vt  Petrol  Rs. 13,12,000  1497 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  Yes  NaN  NaN  NaN  NaN
1269      1269  Honda  City  Vt  Diesel  Rs. 11,11,000  1498 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
1270      1270  Honda  City  Vt  Diesel  Rs. 11,91,000  1498 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
1271      1271  Honda  City  Vt  Diesel  Rs. 13,02,000  1498 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  Yes  NaN  NaN  NaN  NaN
1272      1272  Honda  City  Zx  Diesel  Rs. 14,21,000  1498 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  Yes  Yes  NaN  NaN  NaN
1273      1273  Honda  City  Zx  Petrol  Rs. 14,31,000  1497 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  Yes  Yes  NaN  NaN  NaN
1274      1274  Honda  City  Vt  Petrol  Rs. 12,01,000  1497 cc  4.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
1275      1275  Mitsubishi  Montero  3.2 4i  Rs. 68,62,560  3200 cc  4.0  4.0  4WD (4x4 Drive)  In-line ...  Yes  NaN  NaN  NaN  NaN

10 rows x 141 columns

In [19]:
google_data.head(10).boxplot()

Out[19]:
<AxesSubplot>

In [20]:
google_data.boxplot()

Out[20]:
<AxesSubplot>

In [21]:
google_data.head(10).hist()

Out[21]:
array([[<AxesSubplot:~>: Unnamed: 0'],
       [<AxesSubplot:~>: Cylinders'],
       [<AxesSubplot:~>: Valves_Per_Cylinder'],
       [<AxesSubplot:~>: Doors'],
       [<AxesSubplot:~>: Seating_Capacity'],
       [<AxesSubplot:~>: Number_of_Airbags'],
       [<AxesSubplot:~>: USB_Ports'],
       [<AxesSubplot:~>: dtype=object]])

In [22]:
google_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1276 entries, 0 to 1275
Columns: 141 entries, Unnamed: 0 to Electric_Range
dtypes: float64(8), int64(1), object(134)
memory usage: 1.4+ MB

In [23]:
#data cleaning
google_data.isnull()

Out[23]:
   Unnamed: 0  Make  Model  Variant  Ex-Showroom_Price  Displacement  Cylinders  Valves_Per_Cylinder  Drivetrain  Cylinder_Configuration  ...  Leather_Wrapped  Steering  Automatic_Headlamps  Engine_Type  ASR_/Traction_Control
0      False  False  False  False  False  False  False  False  False  False  ...  True  True  True  True  True
1      False  False  False  False  False  False  False  False  False  False  ...  True  True  True  True  True
2      False  False  False  False  False  False  False  False  False  False  ...  True  True  True  True  True
3      False  False  False  False  False  False  False  False  False  False  ...  True  True  True  True  True
4      False  False  False  False  False  False  False  False  False  False  ...  True  True  True  True  True
...
1271     True  False  False  False  ...
1272     True  False  False  False  ...
1273     True  False  False  False  ...
1274     True  False  False  False  ...
1275     True  False  False  False  ...
1276     True  True  True  True  True

1276 rows x 141 columns

In [24]:
#counting number of missing values in each column
google_data.isnull().sum()

Out[24]:
Unnamed: 0      0
Make            75
Model           0
Variant         0
Ex-Showroom_Price  ...
USB_Ports      127
Heads-Up_Display 125
Welcome_Lights 120
Battery         129
Electric_Range 126
Length: 141, dtype: int64

In [25]:
#cylinders greater than 10
google_data[google_data['Cylinders'] > 12]

Out[25]:
   Unnamed: 0  Make  Model  Variant  Ex-Showroom_Price  Displacement  Cylinders  Valves_Per_Cylinder  Drivetrain  Cylinder_Configuration  ...  Leather_Wrapped  Steering  Automatic_Headlamps  Engine_Type  ASR_/Traction_Control
355      355  Bugati  Chron  W16  Rs. 19,21,42,937  7993 cc  16.0  4.0  RWD (Rear Wheel Drive)  W ...  Yes  Yes  Yes  NaN  NaN
356      356  Bugatti  Chron  Sport  Rs. 21,21,55,397  7993 cc  16.0  4.0  RWD (Rear Wheel Drive)  W ...  Yes  Yes  Yes  NaN  NaN

2 rows x 141 columns

In [ ]:
google_data.drop([355],inplace=True)

In [45]:
google_data.drop([356],inplace=True)

In [47]:
google_data[353:357]

Out[47]:
   Unnamed: 0  Make  Model  Variant  Ex-Showroom_Price  Displacement  Cylinders  Valves_Per_Cylinder  Drivetrain  Cylinder_Configuration  ...  Leather_Wrapped  Steering  Automatic_Headlamps  Engine_Type  ASR_/Traction_Control
354      354  NaN  NaN  Phe-Phe  6.8 L  Rs. 7,73,12,661  6749 cc  12.0  4.0  RWD (Rear Wheel Drive)  V ...  Yes  Yes  Yes  NaN  NaN
357      357  Bajaj  Qute  Qute  Rs. 2,03,000  216 cc  NaN  1.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
358      358  Bajaj  Qute  Qute  Rs. 2,03,000  216 cc  NaN  1.0  RWD (Rear Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN
359      359  Maruti  Suzuki  Alto  Std  Rs. 2,94,000  796 cc  3.0  4.0  FWD (Front Wheel Drive)  In-line ...  NaN  NaN  NaN  NaN  NaN

4 rows x 141 columns

In [48]:
google_data.head(10).boxplot()

Out[48]:
<AxesSubplot>

In [49]:
google_data.boxplot()

Out[49]:
<AxesSubplot>

In [50]:
google_data.head(20).hist()

Out[50]:
array([[<AxesSubplot:~>: Unnamed: 0'],
       [<AxesSubplot:~>: Cylinders'],
       [<AxesSubplot:~>: Valves_Per_Cylinder'],
       [<AxesSubplot:~>: Doors'],
       [<AxesSubplot:~>: Seating_Capacity'],
       [<AxesSubplot:~>: Number_of_Airbags'],
       [<AxesSubplot:~>: USB_Ports'],
       [<AxesSubplot:~>: dtype=object]])

In [51]:
google_data.hist()

Out[51]:
array([[<AxesSubplot:~>: Unnamed: 0'],
       [<AxesSubplot:~>: Cylinders'],
       [<AxesSubplot:~>: Valves_Per_Cylinder'],
       [<AxesSubplot:~>: Doors'],
       [<AxesSubplot:~>: Seating_Capacity'],
       [<AxesSubplot:~>: Number_of_Airbags'],
       [<AxesSubplot:~>: USB_Ports'],
       [<AxesSubplot:~>: dtype=object]])

In [52]:
#remove columns that are 98% empty
threshold=(google_data["0"].sum()/len(google_data))
threshold

Out[52]:
127.38800000000001

In [56]:
google_data.dropna(thresh = threshold,axis=1,inplace=True)

In [57]:
print(google_data.isnull().sum())

Unnamed: 0      0
Make            75
Model           0
Variant         0
Ex-Showroom_Price  ...
Paddle_Shifters      0
Leather_Wrapped_Steering 689
Automatic_Headlamps  774
ASR_/Traction_Control  834
Cruise_Control      727
Length: 126, dtype: int64

In [59]:
google_data.shape
(1273, 126)

In [59]:
#data imputation and manipulation
#filling null values with mean , median and mode
def impute_median(series):
    return series.fillna(series.median())

In [60]:
google_data.Cylinders=google_data['Cylinders'].transform(impute_median)

In [60]:
print(google_data.isnull().sum())

Unnamed: 0      0
Make            75
Model           0
Variant         0
Ex-Showroom_Price  ...
Paddle_Shifters      0
Leather_Wrapped_Steering 689
Automatic_Headlamps  774
ASR_/Traction_Control  834
Cruise_Control      727
Length: 126, dtype: int64

In [68]:
#filling categorical values with mode
print(google_data["Make"].mode())

0      Maruti Suzuki
dtype: object

In [70]:
print(google_data["Paddle_Shifters"].mode())

0      Yes
dtype: object

In [71]:
print(google_data["Cruise_Control"].mode())

0      Yes
dtype: object

In [72]:
google_data["Make"].fillna(str(google_data["Make"].mode().values[0]),inplace=True)
google_data["Paddle_Shifters"].fillna(str(google_data["Paddle_Shifters"].mode().values[0]),inplace=True)
google_data["Cruise_Control"].fillna(str(google_data["Cruise_Control"].mode().values[0]),inplace=True)

In [73]:
print(google_data.isnull().sum())

Unnamed: 0      0
Make            0
Model           0
Variant         0
Ex-Showroom_Price  ...
Paddle_Shifters      0
Leather_Wrapped_Steering 689
Automatic_Headlamps  774
ASR_/Traction_Control  834
Cruise_Control      0
Length: 126, dtype: int64

In [79]:
google_data.describe()

Out[79]:
   Unnamed: 0  Cylinders  Valves_Per_Cylinder  Doors  Seating_Capacity  Number_of_Airbags
count  1273.000000  1273.000000  1171.000000  1266.000000  1267.000000  1138.000000
mean      638.182459  4.369999  3.977787  4.554768  5.272643  3.778438
std       588.604447  1.508426  0.834831  0.741543  1.139021  0.520013
min         0.000000  2.000000  1.000000  2.000000  2.000000  1.000000
25%      318.000000  4.000000  4.000000  4.000000  5.000000  2.000000
50%      638.000000  4.000000  4.000000  5.000000  5.000000  2.000000
75%      957.000000  4.000000  4.000000  5.000000  5.000000  6.000000
max     1275.000000  12.000000  16.000000  5.000000  16.000000  14.000000

In [81]:
#data visualization
fig=plt.figure(figsize=(16,5))
fig=plt.plot(x=google_data["Cylinders"],y=google_data["Make"],color="g")
fig=plt.xticks(rotation=90)
fig=plt.title("Make wise Cylinders")
fig=plt.xlabel("Make")
fig=plt.ylabel("Cylinders")
fig=plt.show()

Make
Make wise Cylinders

In [97]:
fig=plt.figure(figsize=(16,5))
fig=plt.plot(x=google_data["Seating_Capacity"],y=google_data["Make"],color="g")
fig=plt.xticks(rotation=90)
fig=plt.title("Make wise Seating_Capacity")
fig=plt.xlabel("Make")
fig=plt.ylabel("Seating_Capacity")
fig=plt.show()

C:\Users\HP\AppData\Local\Temp\ipykernel_18848\85614818.py:2: UserWarning: color is redundantly defined by the 'color' keyword argument and the fast string "g" (-> color='r'). The keyword argument will take precedence.
fig=plt.plot(x=google_data["Seating_Capacity"],y=google_data["Make"],color="g")

Make wise Seating_Capacity

In [100]:
fig=plt.figure(figsize=(16,5))
fig=plt.plot(x=google_data["Doors"],y=google_data["Make"],color="g")
fig=plt.xticks(rotation=90)
fig=plt.title("Make wise Doors")
fig=plt.xlabel("Make")
fig=plt.ylabel("Doors")
fig=plt.show()

C:\Users\HP\AppData\Local\Temp\ipykernel_18848\85614818.py:2: UserWarning: color is redundantly defined by the 'color' keyword argument and the fast string "g" (-> color='g'). The keyword argument will take precedence.
fig=plt.plot(x=google_data["Doors"],y=google_data["Make"],color="g")

Make wise Doors
```