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

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	I
0	0	Tata	Nano Genx	Xt	Rs. 2,92,667	624 cc	2.0	2.0	
1	1	Tata	Nano Genx	Xe	Rs. 2,36,447	624 cc	2.0	2.0	
2	2	Tata	Nano Genx	Emax Xm	Rs. 2,96,661	624 cc	2.0	2.0	
3	3	Tata	Nano Genx	Xta	Rs. 3,34,768	624 cc	2.0	2.0	
4	4	Tata	Nano Genx	Xm	Rs. 2,72,223	624 cc	2.0	2.0	
5	5	Tata	Nano Genx	Xma	Rs. 3,14,815	624 cc	2.0	2.0	
6	6	Datsun	Redi-Go	D	Rs. 2,79,650	799 cc	3.0	4.0	
7	7	Datsun	Redi-Go	T	Rs. 3,51,832	799 cc	3.0	4.0	
8	8	Datsun	Redi-Go	A	Rs. 3,33,419	799 cc	3.0	4.0	

	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinder	I
9	9	Datsun	Redi-Go	S	Rs. 3,62,000	799 cc	3.0	4.0	

10 rows × 141 columns

In [13]:
```
type(google_data)
```

Out[13]:
```
pandas.core.frame.DataFrame
```

In [14]:
```
google_data.shape
```

Out[14]:
```
(1276, 141)
```

In [16]:
```
google_data.describe()
```

Out[16]:

	Unnamed: 0	Cylinders	Valves_Per_Cylinder	Doors	Seating_Capacity	Number_of_Airbags
count	1276.000000	1210.000000	1174.000000	1272.000000	1270.000000	1141.000000
mean	637.500000	4.380992	3.977853	4.550314	5.270079	3.787029
std	368.493781	1.660957	0.833763	0.747816	1.145231	2.522399
min	0.000000	2.000000	1.000000	2.000000	2.000000	1.000000
25%	318.750000	4.000000	4.000000	4.000000	5.000000	2.000000
50%	637.500000	4.000000	4.000000	5.000000	5.000000	2.000000
75%	956.250000	4.000000	4.000000	5.000000	5.000000	6.000000
max	1275.000000	16.000000	16.000000	5.000000	16.000000	14.000000

In [18]:
```
google_data.tail(10)
```

Out[18]:

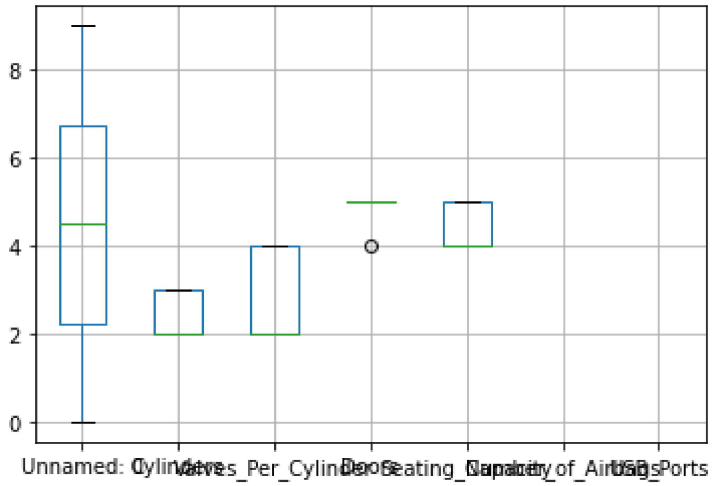
	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cy
1266	1266	Honda	City	V Mt Petrol	Rs. 10,65,900	1497 cc	4.0	

	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cy
1267	1267	Honda	City	Vx Mt Petrol	Rs. 11,82,000	1497 cc	4.0	
1268	1268	Honda	City	Vx Cvt Petrol	Rs. 13,12,000	1497 cc	4.0	
1269	1269	Honda	City	Sv Mt Diesel	Rs. 11,11,000	1498 cc	4.0	
1270	1270	Honda	City	V Mt Diesel	Rs. 11,91,000	1498 cc	4.0	
1271	1271	Honda	City	Vx Mt Diesel	Rs. 13,02,000	1498 cc	4.0	
1272	1272	Honda	City	Zx Mt Diesel	Rs. 14,21,000	1498 cc	4.0	
1273	1273	Honda	City	Zx Cvt Petrol	Rs. 14,31,000	1497 cc	4.0	
1274	1274	Honda	City	V Cvt Petrol	Rs. 12,01,000	1497 cc	4.0	
1275	1275	Mitsubishi	Montero	3.2 At	Rs. 68,62,560	3200 cc	4.0	

10 rows × 141 columns

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

Out[19]:
```
<AxesSubplot:>
```

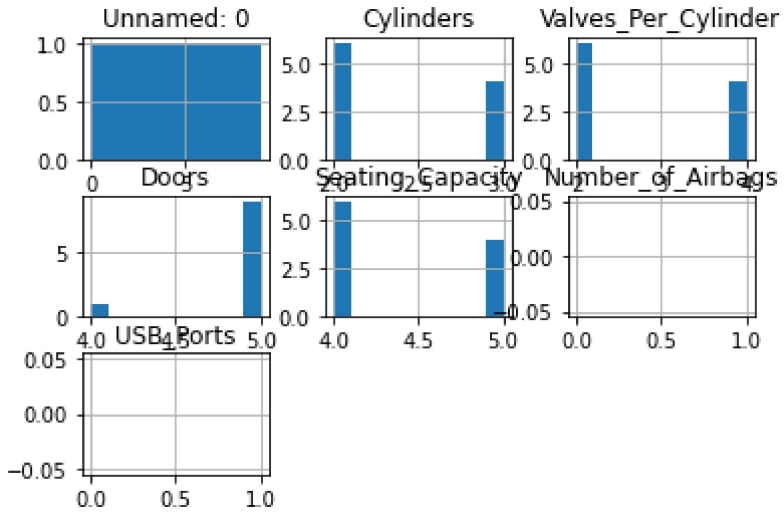


In [20]:
```
google_data.boxplot()
```

Out[20]:
```
<AxesSubplot:>
```

In [33]:
```
google_data.head(10).hist()
```

Out[33]:
```
array([[<AxesSubplot:title={'center':'Unnamed: 0'}>,
        <AxesSubplot:title={'center':'Cylinders'}>,
        <AxesSubplot:title={'center':'Valves_Per_Cylinder'}>],
       [<AxesSubplot:title={'center':'Doors'}>,
        <AxesSubplot:title={'center':'Seating_Capacity'}>,
        <AxesSubplot:title={'center':'Number_of_Airbags'}>],
       [<AxesSubplot:title={'center':'USB_Ports'}>, <AxesSubplot:>,
        <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(6), 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
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
1271	False	False	False	False	False	False	False	False
1272	False	False	False	False	False	False	False	False
1273	False	False	False	False	False	False	False	False
1274	False	False	False	False	False	False	False	False
1275	False	False	False	False	False	False	False	False

1276 rows × 141 columns

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

Out[24]:
```
Unnamed: 0              0
Make                   75
Model                   0
Variant                 0
Ex-Showroom_Price       0
                      ...
USB_Ports            1247
Heads-Up_Display     1225
Welcome_Lights       1207
Battery              1263
Electric_Range       1259
Length: 141, dtype: int64
```

In [39]:
```
#cylinders greather than 10
google_data[google_data.Cylinders > 12]
```

Out[39]:

	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinder
355	355	Bugatti	Chiron	W16	Rs. 19,21,42,937	7993 cc	16.0	4.0
356	356	Bugatti	Chiron	Sport	Rs. 21,21,55,397	7993 cc	16.0	4.0

2 rows × 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]:

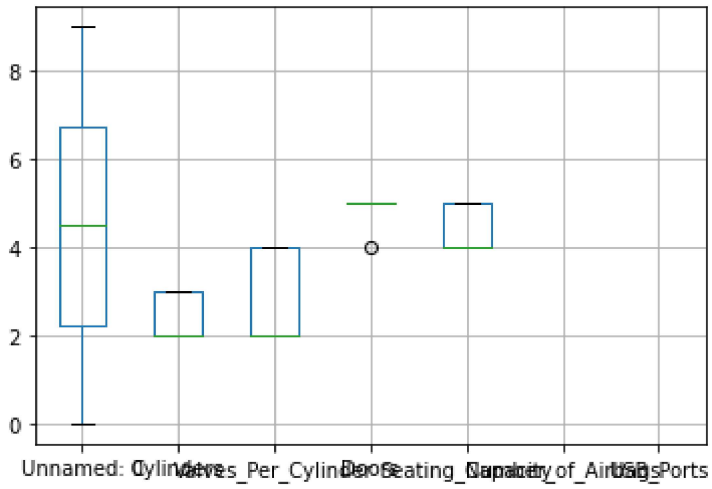
	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinde
354	354	NaN	Rolls-Royce Phantom Coupe	6.8 L	Rs. 7,73,12,661	6749 cc	12.0	4.
357	357	Bajaj	Qute (Re60)	Qcar	Rs. 2,63,000	216 cc	NaN	1.
358	358	Bajaj	Qute (Re60)	Qcar Cng	Rs. 2,83,000	216 cc	NaN	1.

	Unnamed: 0	Make	Model	Variant	Ex-Showroom_Price	Displacement	Cylinders	Valves_Per_Cylinde
359	359	Maruti Suzuki	Alto	Std	Rs. 2,94,800	796 cc	3.0	4.

4 rows × 141 columns

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

Out[48]: <AxesSubplot:>



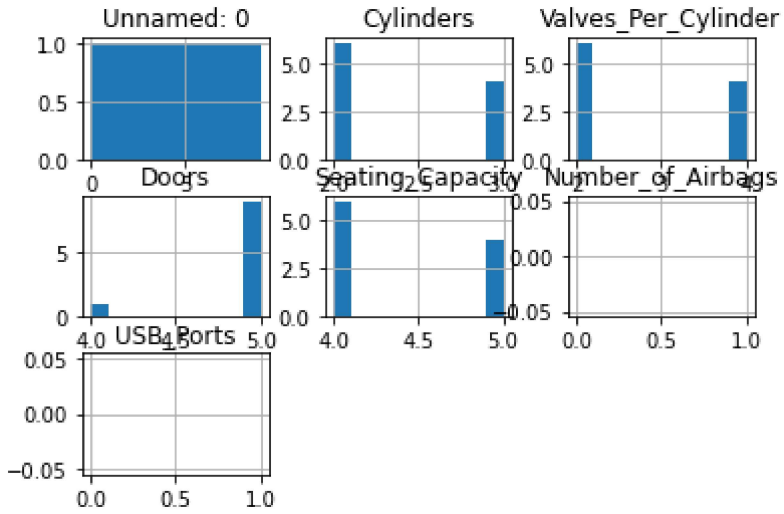
In [49]:
```
google_data.boxplot()
```

Out[49]: <AxesSubplot:>

In [50]:
```
google_data.head(10).hist()
```

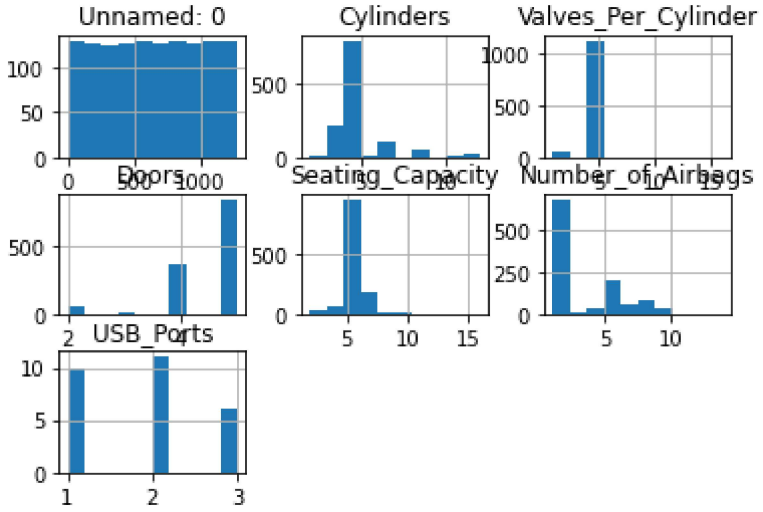
Out[50]:
```
array([[<AxesSubplot:title={'center':'Unnamed: 0'}>,
        <AxesSubplot:title={'center':'Cylinders'}>,
```

```
 <AxesSubplot:title={'center':'Valves_Per_Cylinder'}>],
[<AxesSubplot:title={'center':'Doors'}>,
 <AxesSubplot:title={'center':'Seating_Capacity'}>,
 <AxesSubplot:title={'center':'Number_of_Airbags'}>],
[<AxesSubplot:title={'center':'USB_Ports'}>, <AxesSubplot:>,
 <AxesSubplot:>]], dtype=object)
```



In [51]:
```
google_data.hist()
```

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



In [52]:
```
#remove columns that are 90% emmpty
threshold=len(google_data)*0.1
threshold
```

Out[52]:
```
127.30000000000001
```

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            0
                          ...
Paddle_Shifters            972
Leather_Wrapped_Steering   689
Automatic_Headlamps        774
ASR_/_Traction_Control     834
Cruise_Control             727
Length: 126, dtype: int64
```

In [59]:
```
google_data.shape
```

Out[59]:
```
(1273, 126)
```

In [60]:
```
#Data imputation and manipulation
#filling null values with mean , median and mode\
def impute_median(series):
    return series.fillna(series.median())
```

In [65]:
```
google_data.Cylinders=google_data['Cylinders'].transform(impute_median)
```

In [66]:
```
print(google_data.isnull().sum())
```

```
Unnamed: 0                   0
Make                        75
Model                        0
Variant                      0
Ex-Showroom_Price            0
                          ...
Paddle_Shifters            972
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[
google_data['Cruise_Control'].fillna(str(google_data['Cruise_Control'].mode().values[0]
```

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

```
Unnamed: 0                   0
Make                         0
Model                        0
Variant                      0
Ex-Showroom_Price            0
                          ...
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	1269.000000	1267.000000	1138.000000
mean	638.192459	4.336999	3.977797	4.554768	5.276243	3.779438
std	368.650447	1.539426	0.834831	0.741643	1.138621	2.520913
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%	639.000000	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

grp=google_data.groupby('Make')
x=grp['Cylinders'].agg(np.mean)
y=grp['Seating_Capacity'].agg(np.sum)
z=grp['Doors'].agg(np.mean)
print(x)
```

```
print(y)
print(z)
```

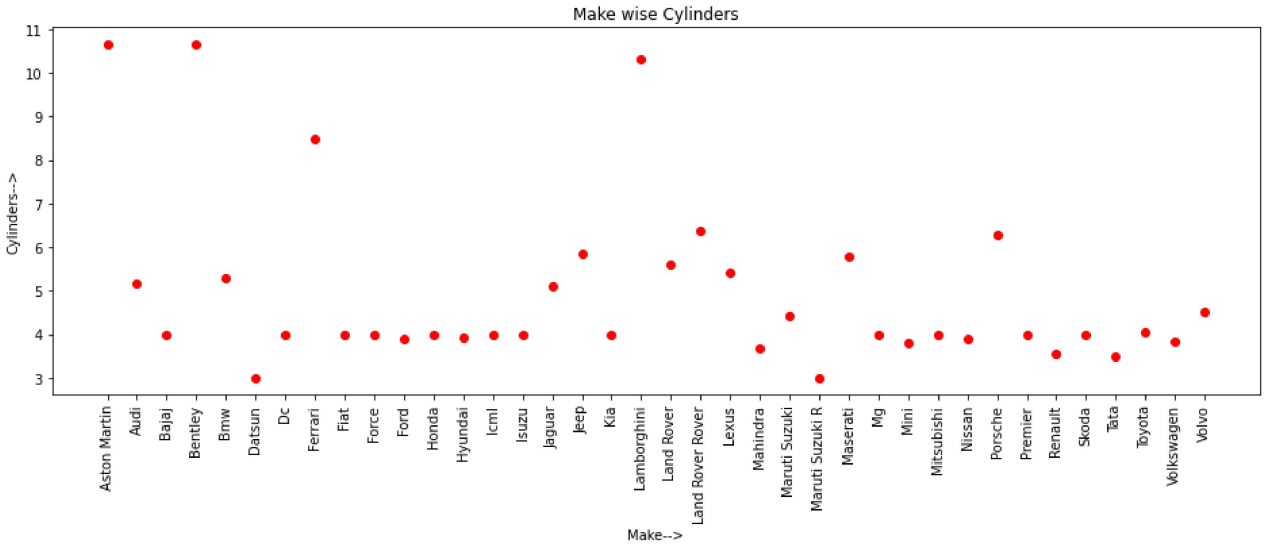
```
Make
Aston Martin        10.666667
Audi                 5.161290
Bajaj                4.000000
Bentley             10.666667
Bmw                  5.277778
Datsun               3.000000
Dc                   4.000000
Ferrari              8.500000
Fiat                 4.000000
Force                4.000000
Ford                 3.883721
Honda                4.000000
Hyundai              3.938462
Icml                 4.000000
Isuzu                4.000000
Jaguar               5.090909
Jeep                 5.857143
Kia                  4.000000
Lamborghini         10.307692
Land Rover           5.600000
Land Rover Rover     6.370370
Lexus                5.400000
Mahindra             3.680672
Maruti Suzuki        4.433036
Maruti Suzuki R      3.000000
Maserati             5.777778
Mg                   4.000000
Mini                 3.800000
Mitsubishi           4.000000
Nissan               3.896552
Porsche              6.285714
Premier              4.000000
Renault              3.555556
Skoda                4.000000
Tata                 3.480000
Toyota               4.048780
Volkswagen           3.823529
Volvo                4.500000
Name: Cylinders, dtype: float64
Make
Aston Martin          8.0
Audi                161.0
Bajaj                 8.0
Bentley              26.0
Bmw                 166.0
Datsun               75.0
Dc                    2.0
Ferrari              20.0
Fiat                115.0
Force                34.0
Ford                220.0
Honda               340.0
Hyundai             649.0
Icml                 93.0
Isuzu                29.0
Jaguar               82.0
```

```
Jeep                140.0
Kia                 115.0
Lamborghini          29.0
Land Rover           70.0
Land Rover Rover    157.0
Lexus                53.0
Mahindra            770.0
Maruti Suzuki      1166.0
Maruti Suzuki R      70.0
Maserati             39.0
Mg                   65.0
Mini                 45.0
Mitsubishi           49.0
Nissan              144.0
Porsche              49.0
Premier              30.0
Renault             199.0
Skoda               217.0
Tata                524.0
Toyota              465.0
Volkswagen          165.0
Volvo                96.0
Name: Seating_Capacity, dtype: float64
Make
Aston Martin        2.666667
Audi                4.225806
Bajaj               4.000000
Bentley             4.000000
Bmw                 4.250000
Datsun              5.000000
Dc                  2.000000
Ferrari             2.000000
Fiat                4.695652
Force               4.000000
Ford                4.651163
Honda               4.437500
Hyundai             4.692308
Icml                5.000000
Isuzu               4.400000
Jaguar              3.454545
Jeep                5.000000
Kia                 5.000000
Lamborghini         2.230769
Land Rover          5.000000
Land Rover Rover    4.962963
Lexus               3.800000
Mahindra            4.881356
Maruti Suzuki       4.553571
Maruti Suzuki R     5.000000
Maserati            3.777778
Mg                  5.000000
Mini                4.000000
Mitsubishi          5.000000
Nissan              4.655172
Porsche             3.750000
Premier             5.000000
Renault             4.805556
Skoda               4.023256
Tata                4.840000
Toyota              4.609756
```

```
Volkswagen          4.363636
Volvo               4.777778
Name: Doors, dtype: float64
```

In [91]:

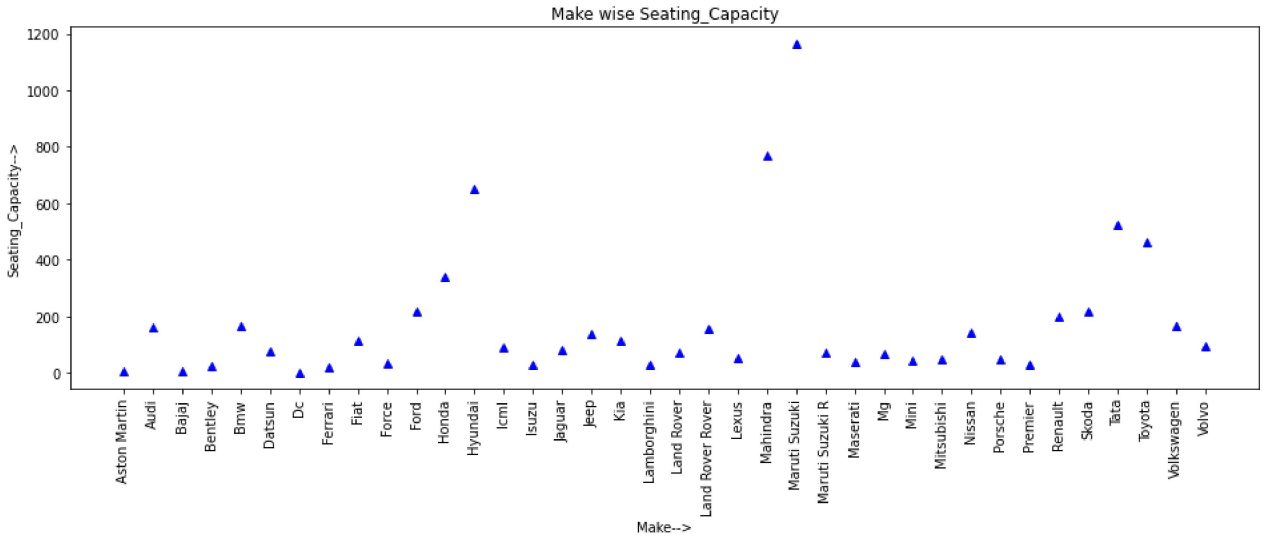
```
plt.figure(figsize=(16,5))
plt.plot(x,'ro')
plt.xticks(rotation=90)
plt.title('Make wise Cylinders')
plt.xlabel('Make-->')
plt.ylabel('Cylinders-->')
plt.show()
```



In [97]:

```
plt.figure(figsize=(16,5))
plt.plot(y,'r^',color='b')
plt.xticks(rotation=90)
plt.title('Make wise Seating_Capacity')
plt.xlabel('Make-->')
plt.ylabel('Seating_Capacity-->')
plt.show()
```

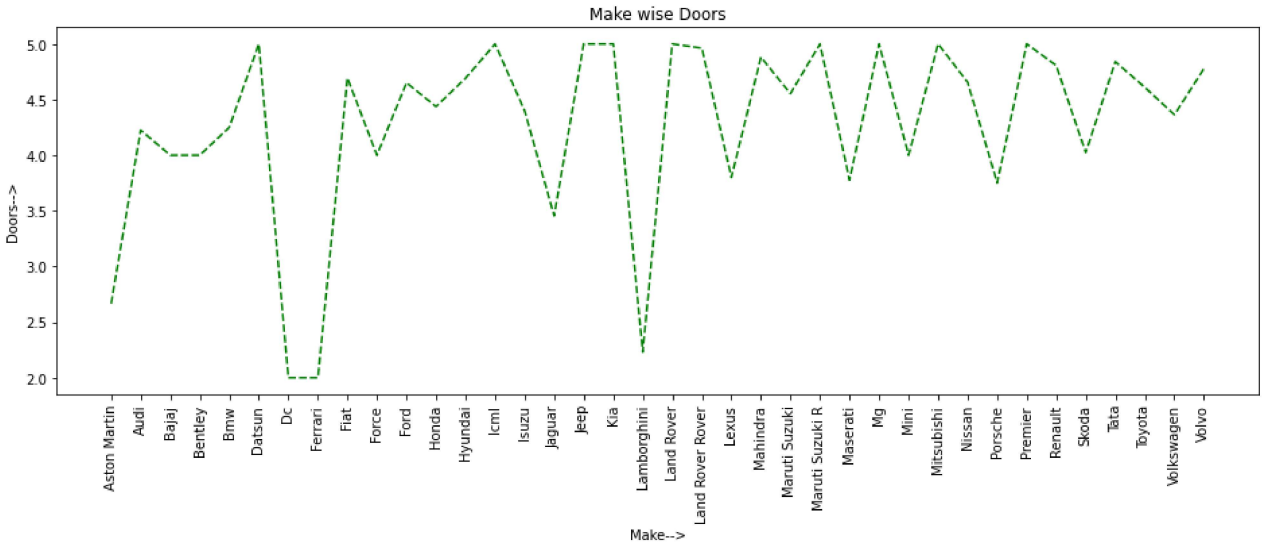
```
C:\Users\HP\AppData\Local\Temp/ipykernel_18048/403418410.py:2: UserWarning: color is red
undantly defined by the 'color' keyword argument and the fmt string "r^" (-> color='r').
The keyword argument will take precedence.
  plt.plot(y,'r^',color='b')
```



In [102…

```
plt.figure(figsize=(16,5))
plt.plot(z,'g--', color='g')
plt.xticks(rotation=90)
plt.title('Make wise Doors')
plt.xlabel('Make-->')
plt.ylabel('Doors-->')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp/ipykernel_18048/856140347.py:2: UserWarning: color is red
undantly defined by the 'color' keyword argument and the fmt string "g--" (-> color
='g'). The keyword argument will take precedence.
  plt.plot(z,'g--', color='g')
```



In [ ]: