Analyzing the Impact of Car Features

A DATA ANALYTICS PROJECT

Project By:

Yaadhav R

Project Description

The objective of this project is to comprehensively analyze car features, market categories, and pricing dynamics to gain insights into consumer demand. By leveraging regression analysis and market segmentation techniques, the aim is to develop pricing strategies that strike a balance between maximizing profitability and meeting consumer demand. Additionally, the project seeks to provide guidance for product development efforts, directing focus towards features and categories that promise high profitability. Through data-driven decisions, the goal is to enhance the manufacturer's competitiveness in the automotive market, ensuring alignment with consumer preferences while maximizing profitability over time.

Tech-Stack Used

Jupyter Notebook - Version 7.0.7 is used to implement Python with various other libraries :

- Pandas - Data Manipulation and Analysis.
- Numpy - Mathematical Computations.
- Matplotlib - Data Visualization.
- Seaborn - Advanced Data Visualization.

Jupyter Notebook Github Link : 🔗

Handling Missing Data

Approach

- First, the missing values in each column are identified.
- The missing values in numerical columns are filled with their respective median of the grouped data of the selected feature.
- Similar, the missing values in categorical columns are with their respective column mode.

```
gd = data.groupby('city mpg')['Engine Cylinders'].transform('median')
data['Engine Cylinders'].fillna( value=gd, inplace=True )

data['Engine Cylinders'].isnull().sum()
```

```
gd = data.groupby('Vehicle Size')['Number of Doors'].transform(lambda x: x.mode().iloc[0])
data['Number of Doors'].fillna( value=gd, inplace=True )

data['Number of Doors'].isnull().sum()
```

Insights

- Initially, there were only 3850 null values in the given dataset.
- The column ‘Market Category’ had the 3742 null values, which is more than 97% of the total null values.
- After replacing the missing values with th respective column median & mode values, there were no duplicate rows in the data.
- This was noticed when the shape of the data frame remained the same (11914, 16) before and after executing the drop_duplicates function.
- Also no rows were dropped from the dataset, so the original shape of the dataframe was retained.

Outliers Detection

Approach

- First the feature columns are selected. The numerical columns are chosen as the features out of which 2 unnecessary columns were removed.
- The outliers in these feature columns are detected with the help of box plot. Then inter-quartile range and z-score methods are employed to remove them.
- In inter-quartile range method, the inter-quartile range(iqr) is calculated from the difference of Q3(75%) and Q1(25%), with this iqr the upper bound and lower bound are calculated.

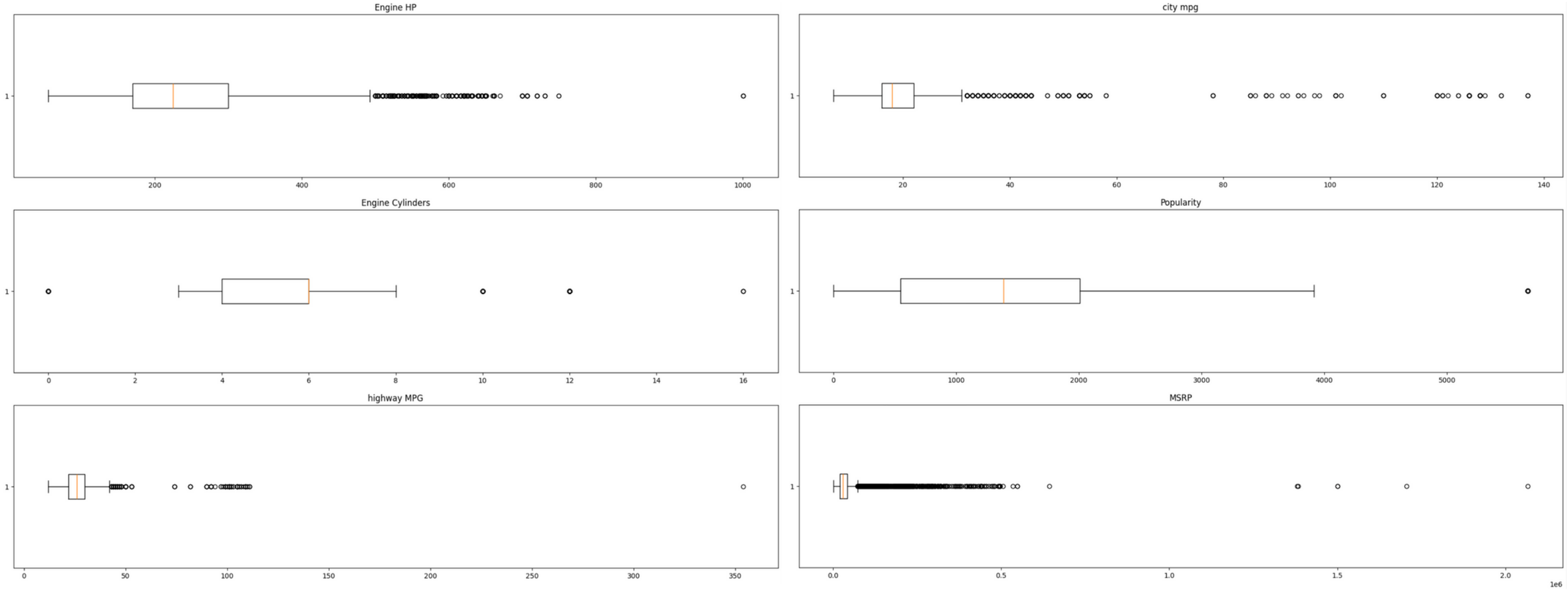
- In each column, the values more than the upper bound and less than the lower bound are considered as outliers and the respective rows are filtered out from the data.

```
for col in features:
    q1, q3 = np.percentile( data[col], [25,75])
    iqr = q3-q1

    lower = q1-(iqr*1.5)
    upper = q3+(iqr*1.5)

    data = data[(data[col]<upper) & (data[col]>lower)]
data.shape
```

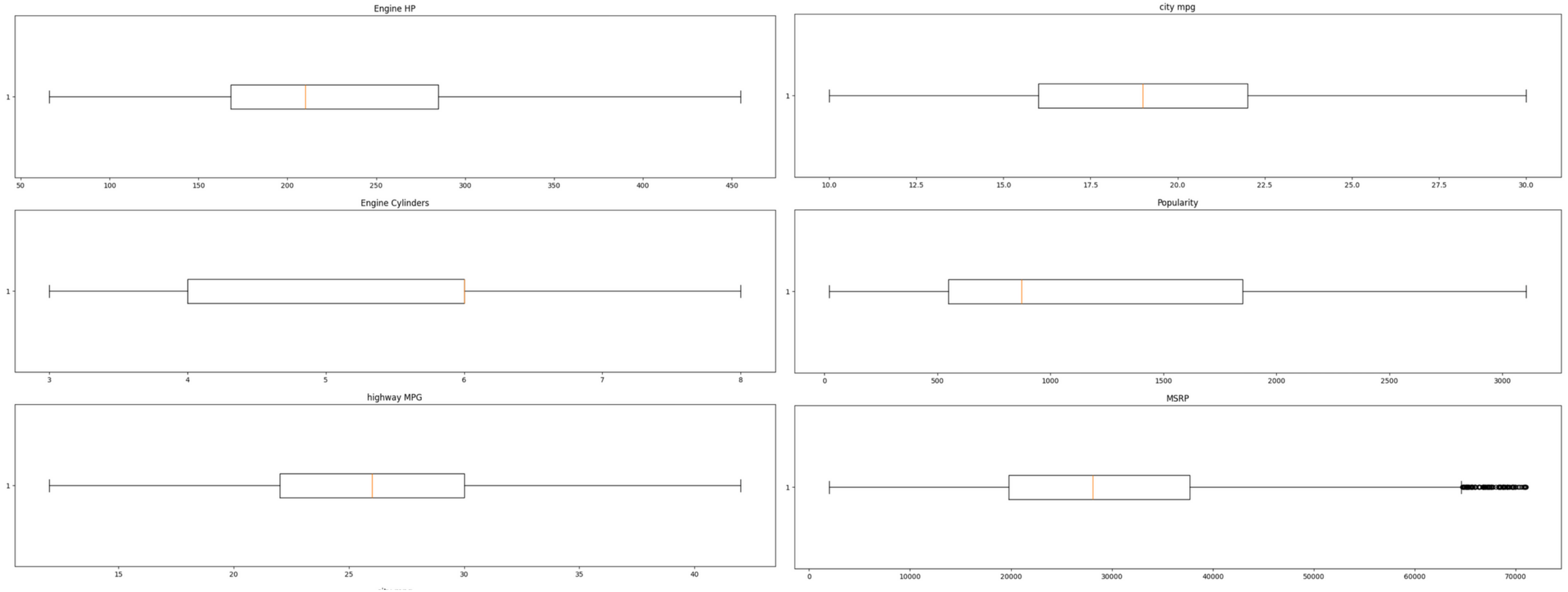
- In Z-Score method, the z-score of each column is calculated and the entire row of the values with score > 3 are removed.

```
for col in features:
    z_scores = (data[col]-data[col].mean())/data[col].std()
    data = data[ abs(z_scores)<3 ]
data.shape
```

With the Presence of Outliers



After the Removal of Outliers



Insights

- The shape of the data frame was (11914, 16) before the removal of outliers.
- 2343 rows were dropped from the data after applying the inter-quartile range method and the new shape was (9571, 16).
- Further 265 rows were dropped from the data after applying z-score method and the final shape was (9306, 16).
- Hence a total of 2608 rows were dropped from the data.
- Even the though two methods were employed to remove the outliers, still there were outliers in the columns ‘MSRP’ but the outliers were significantly less so they were neglected.

Analysis on Features

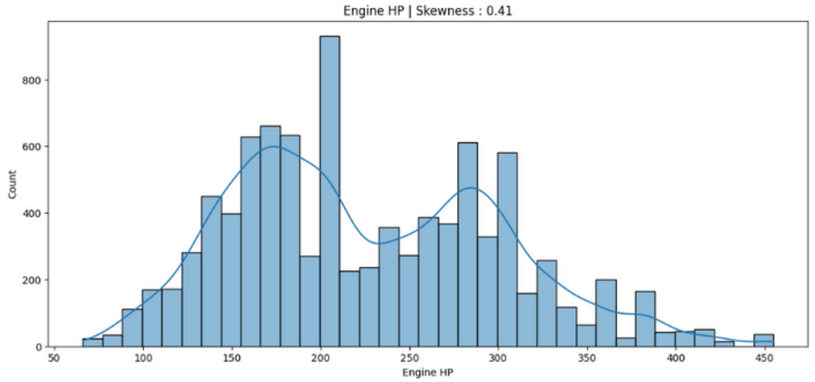
Approach

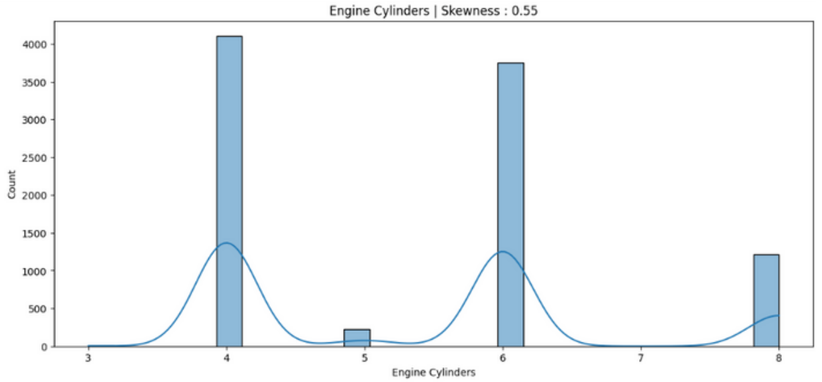
- For every graphs, first a figure size is defined then the type of graph is plotted on this figure with the required parameters and finally the plot is displayed.
- A sample of one graph is shown below.

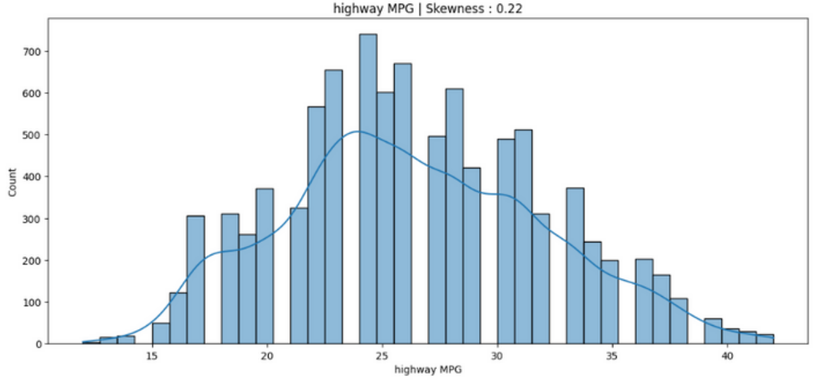
```
plt.figure(figsize=(20,20))

i=1
for col in features:
    plt.subplot( 3, 2, i)
    sns.histplot( data[col], kde=True)
    plt.title(f'{col} | Skewness : {round( data[col].skew(), 2)}')
    i+=1

plt.show()
```

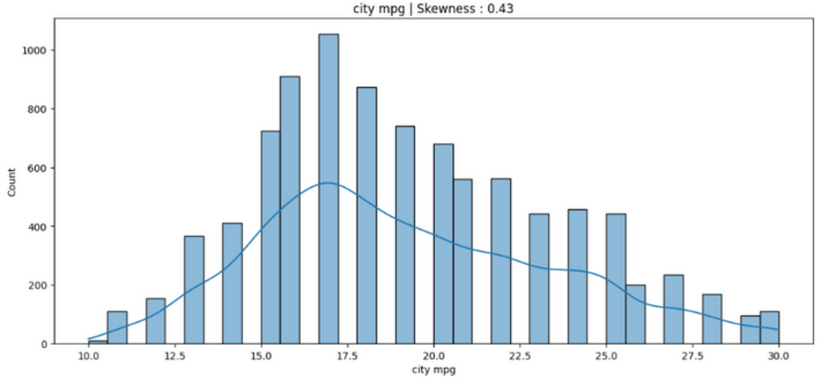
```
plt.figure(figsize=(20,20))

plt.subplot( 2, 1, 1)
plt.scatter( data[features[0]], data[features[5]])
plt.title(f'{features[5]} vs {features[0]}')
plt.xlabel(features[0])
plt.ylabel(features[5])

plt.subplot( 2, 1, 2)
plt.scatter( data[features[0]], data[features[4]])
plt.title(f'{features[4]} vs {features[0]}')
plt.xlabel(features[4])
plt.ylabel(features[0])

plt.show()
```

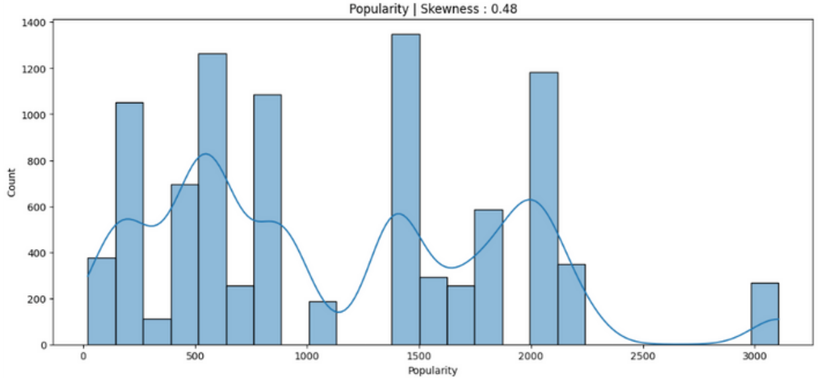
```
plt.figure(figsize=(20,30))

for i in range(1,4):
    plt.subplot( 4, 2, i*2-1)
    sns.barplot( x=features[i], y=features[5], data=data[features])
    plt.title(f'{features[i]} vs {features[5]}')

    plt.subplot( 4, 2, i*2)
    sns.barplot( x=features[i], y=features[4], data=data[features])
    plt.title(f'{features[i]} vs {features[4]}')
    i+=1

plt.show()
```

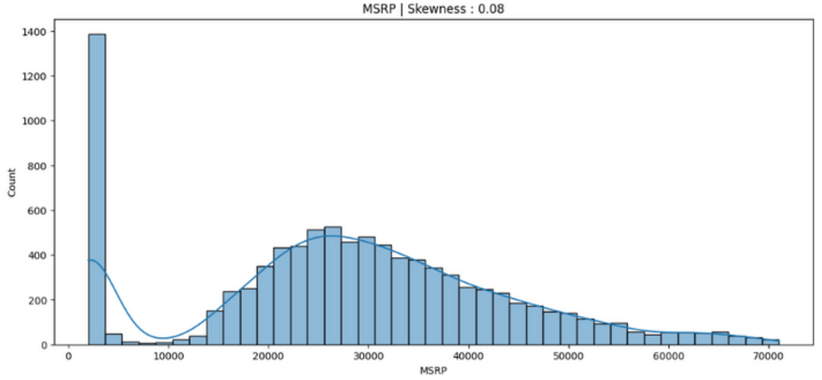

Engine HP | Skewness : 0.41
Count
Engine HP

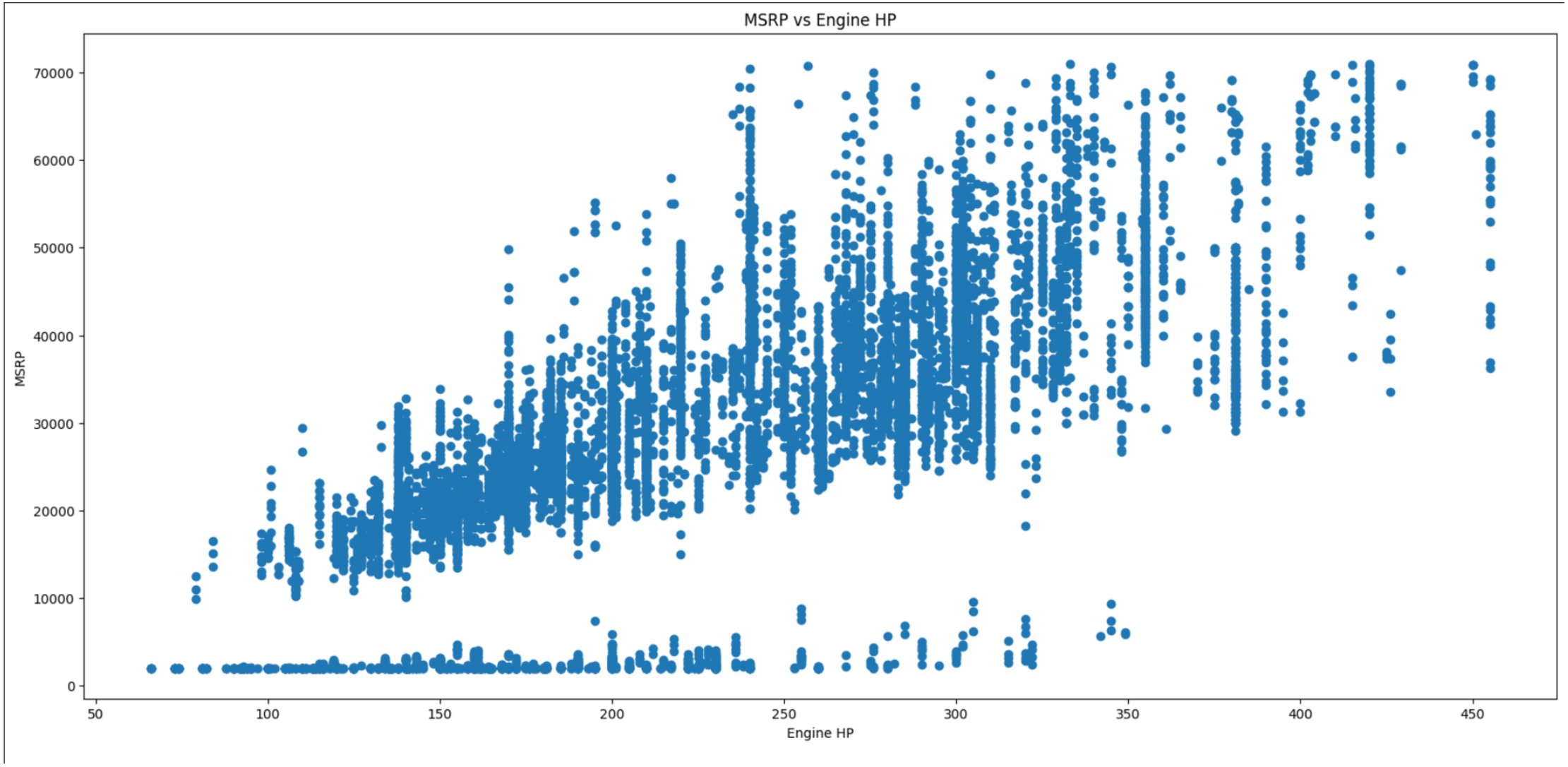

Engine Cylinders | Skewness : 0.55
Count
Engine Cylinders

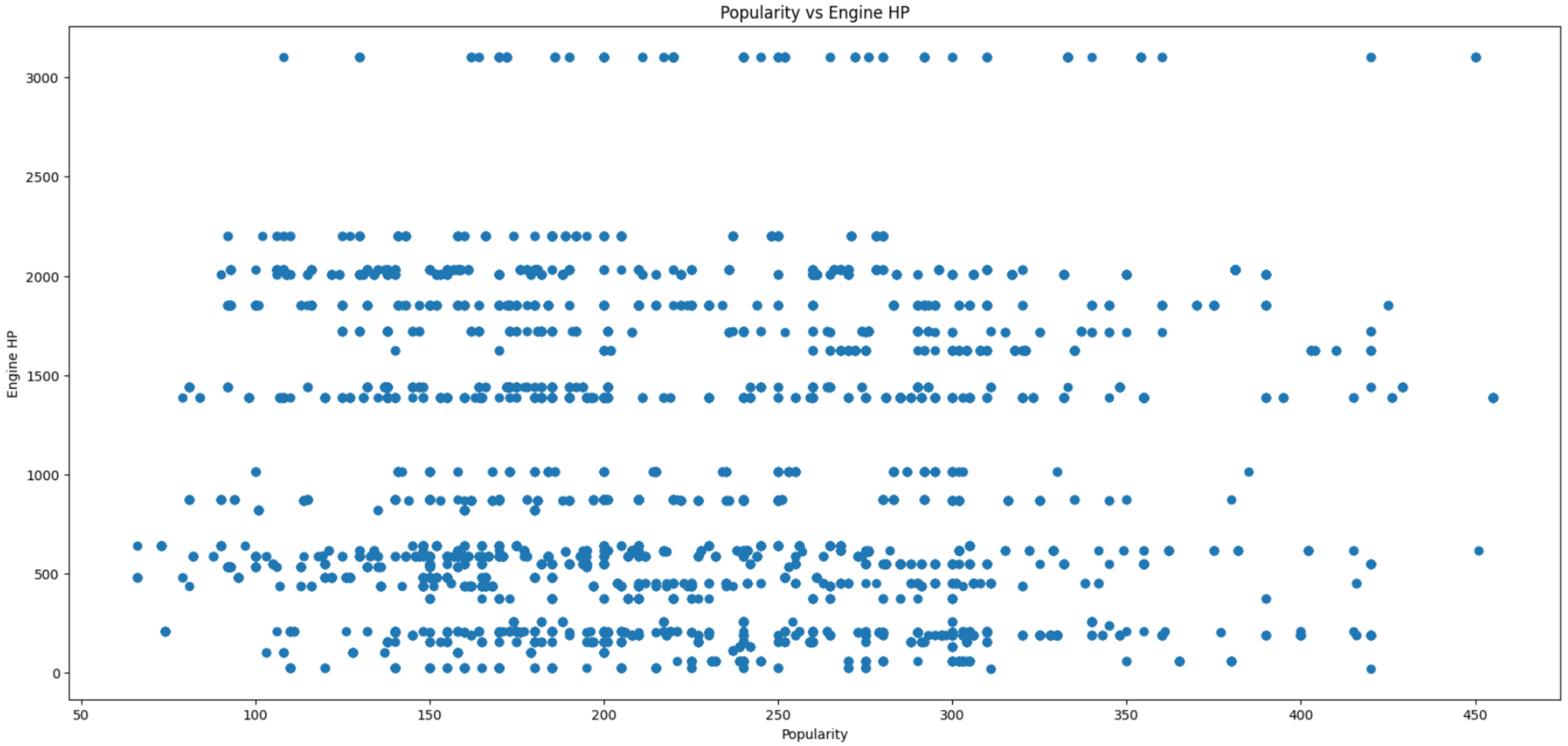

highway MPG | Skewness : 0.22
Count
highway MPG

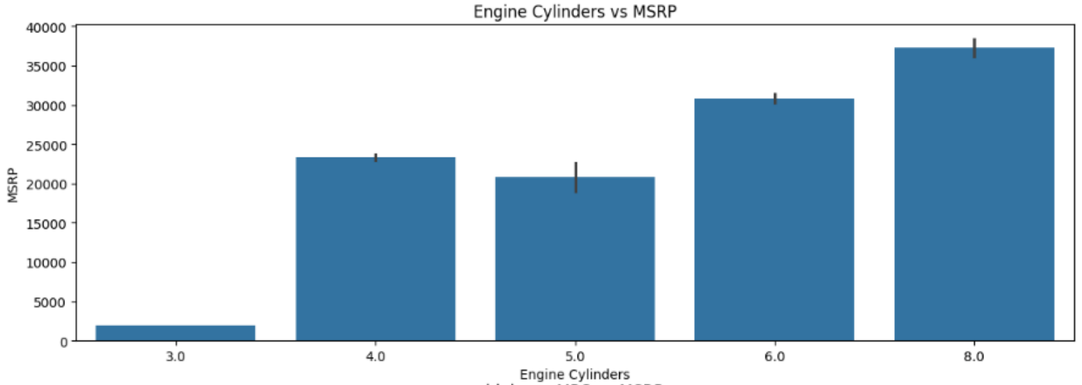

city mpg | Skewness : 0.43
Count
city mpg

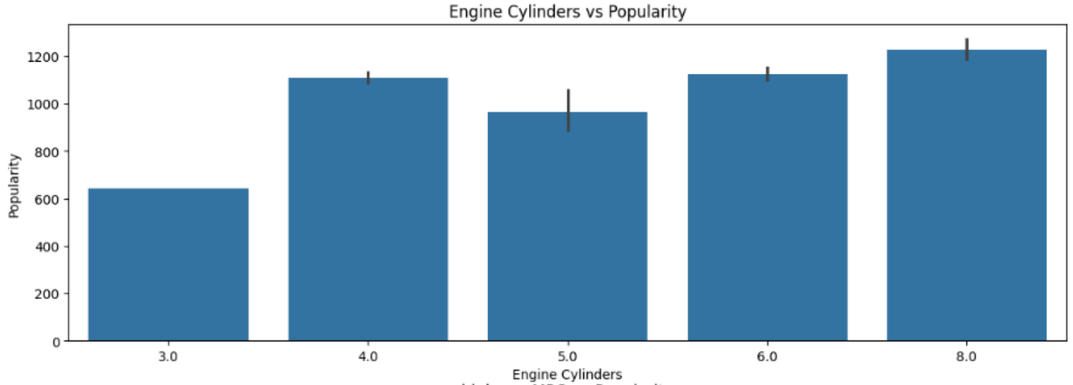

Popularity | Skewness : 0.48
Count
Popularity

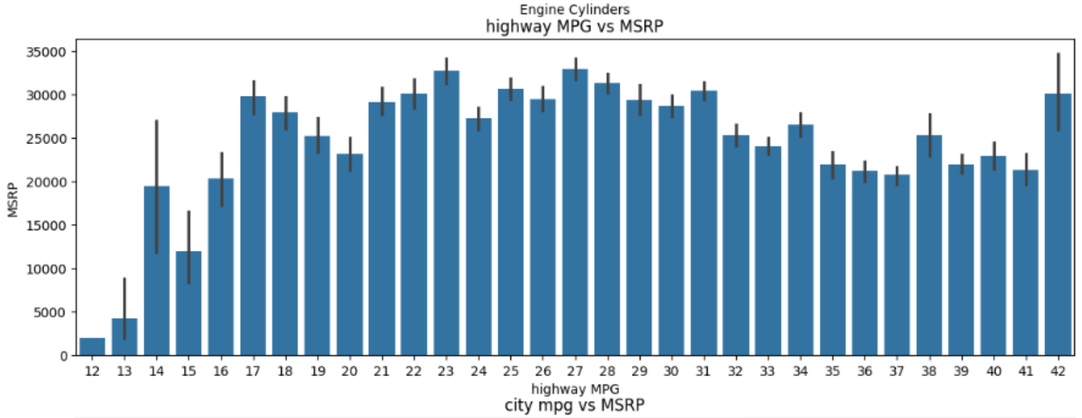

MSRP | Skewness : 0.08
Count
MSRP

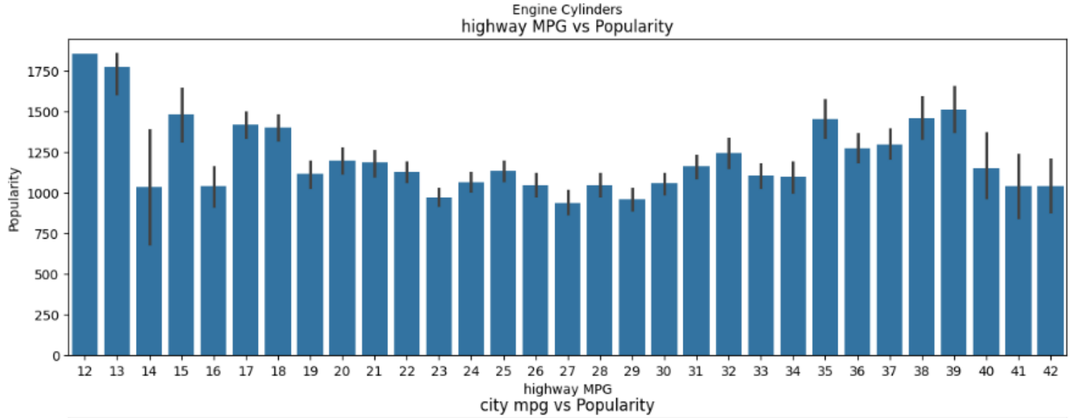

MSRP vs Engine HP
MSRP
Engine HP
0
10000
20000
30000
40000
50000
60000
70000
50
100
150
200
250
300
350
400
450

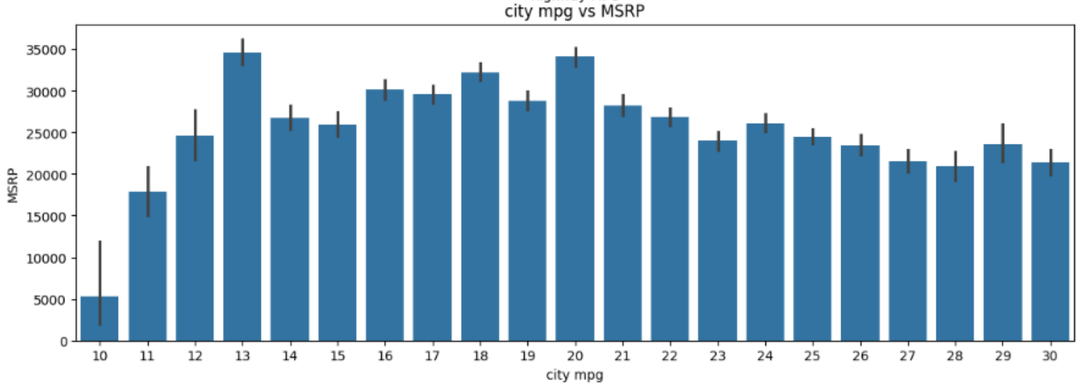

Popularity vs Engine HP
Engine HP
Popularity
3000
2500
2000
1500
1000
500
0
50
100
150
200
250
300
350
400
450

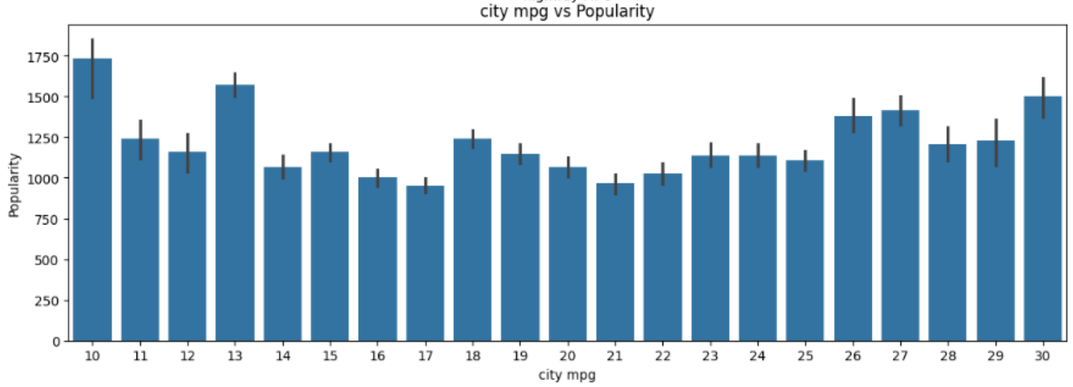

Engine Cylinders vs MSRP
MSRP
Engine Cylinders
0
5000
10000
15000
20000
25000
30000
35000
40000
3.0
4.0
5.0
6.0
8.0

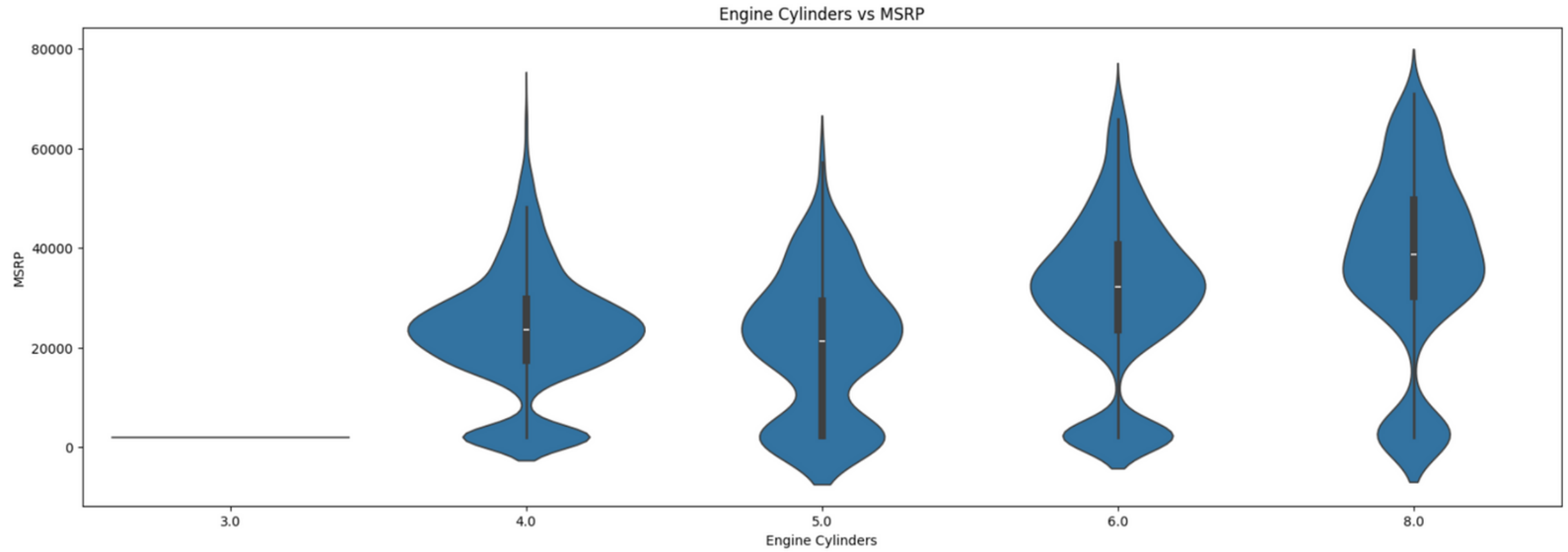

Engine Cylinders vs Popularity
Popularity
Engine Cylinders
0
200
400
600
800
1000
1200
3.0
4.0
5.0
6.0
8.0

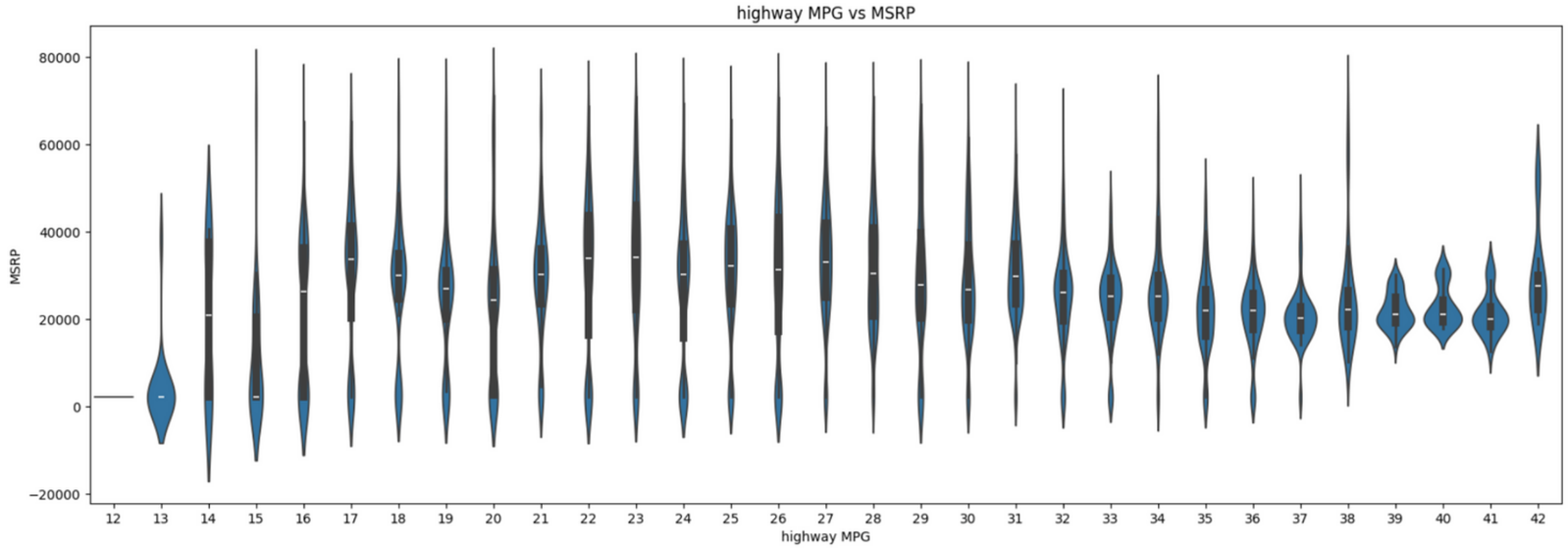

highway MPG vs MSRP
MSRP
highway MPG
0
5000
10000
15000
20000
25000
30000
35000
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42


highway MPG vs Popularity
Popularity
highway MPG
0
250
500
750
1000
1250
1500
1750
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42


city mpg vs MSRP
MSRP
city mpg
0
5000
10000
15000
20000
25000
30000
35000
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30


city mpg vs Popularity
Popularity
city mpg
0
250
500
750
1000
1250
1500
1750
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30


Engine Cylinders vs MSRP
MSRP
80000
60000
40000
20000
0
3.0
4.0
5.0
6.0
8.0
Engine Cylinders


highway MPG vs MSRP
MSRP
80000
60000
40000
20000
0
−20000
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
highway MPG

Insights

- From the univariate analysis of features, the graphs of ‘Engine HP’, ‘highway mpg’, ‘city mpg’ were similar and the skewness of features was in range.
- From the bivariate analysis it is observed that the ‘Engine HP’ had an impact on ‘MSRP’ but the ‘Engine HP’ did not have any impact on ‘Popularity’.
- The feature ‘Engine Cylinders’ had little impact on ‘MSRP’ and ‘Engine HP’ but it was not significant.
- The remaining features ‘highway mpg’ and ‘city mpg’ didn’t have any impact on ‘MSRP’ as well as ‘Engine HP’.
- So it is safe to say that the features ‘highway mpg’ and ‘city mpg’ does not affect the pricing and popularity of a car.

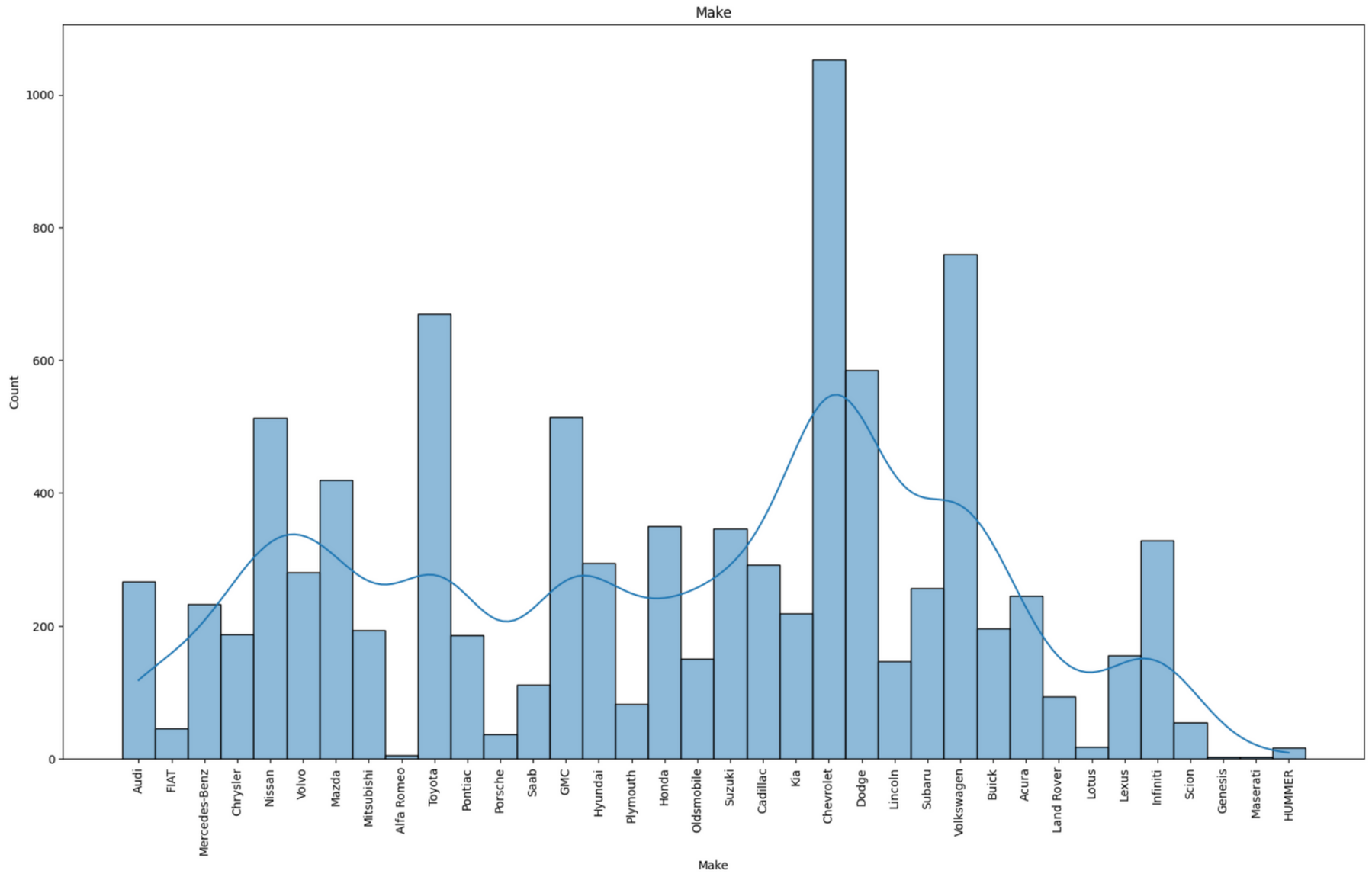
Analysis on Categories

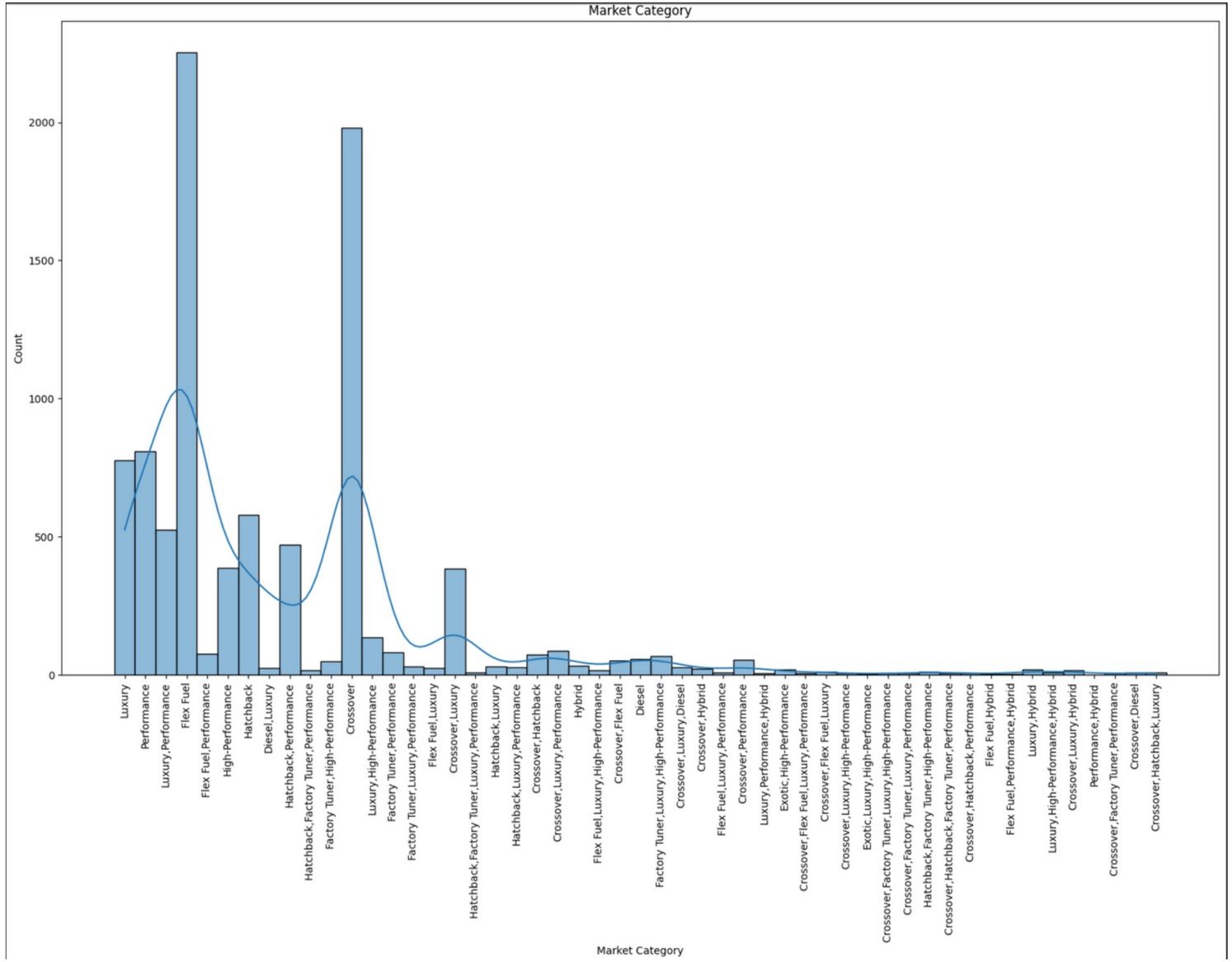
Approach

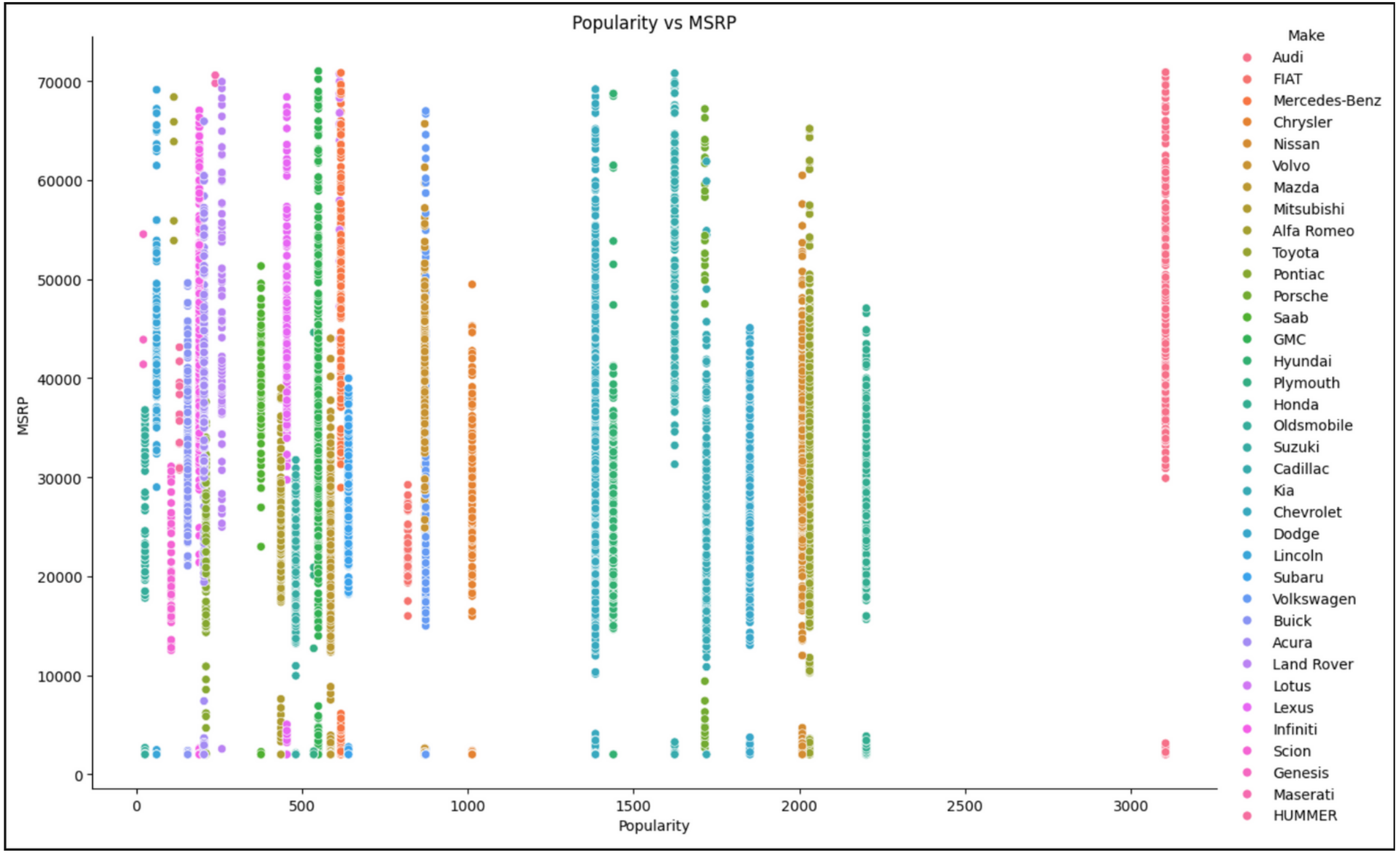
- For every graphs, first a figure size is defined then the type of graph is plotted on this figure with the required parameters and finally the plot is displayed.
- A sample of one graph is shown below.

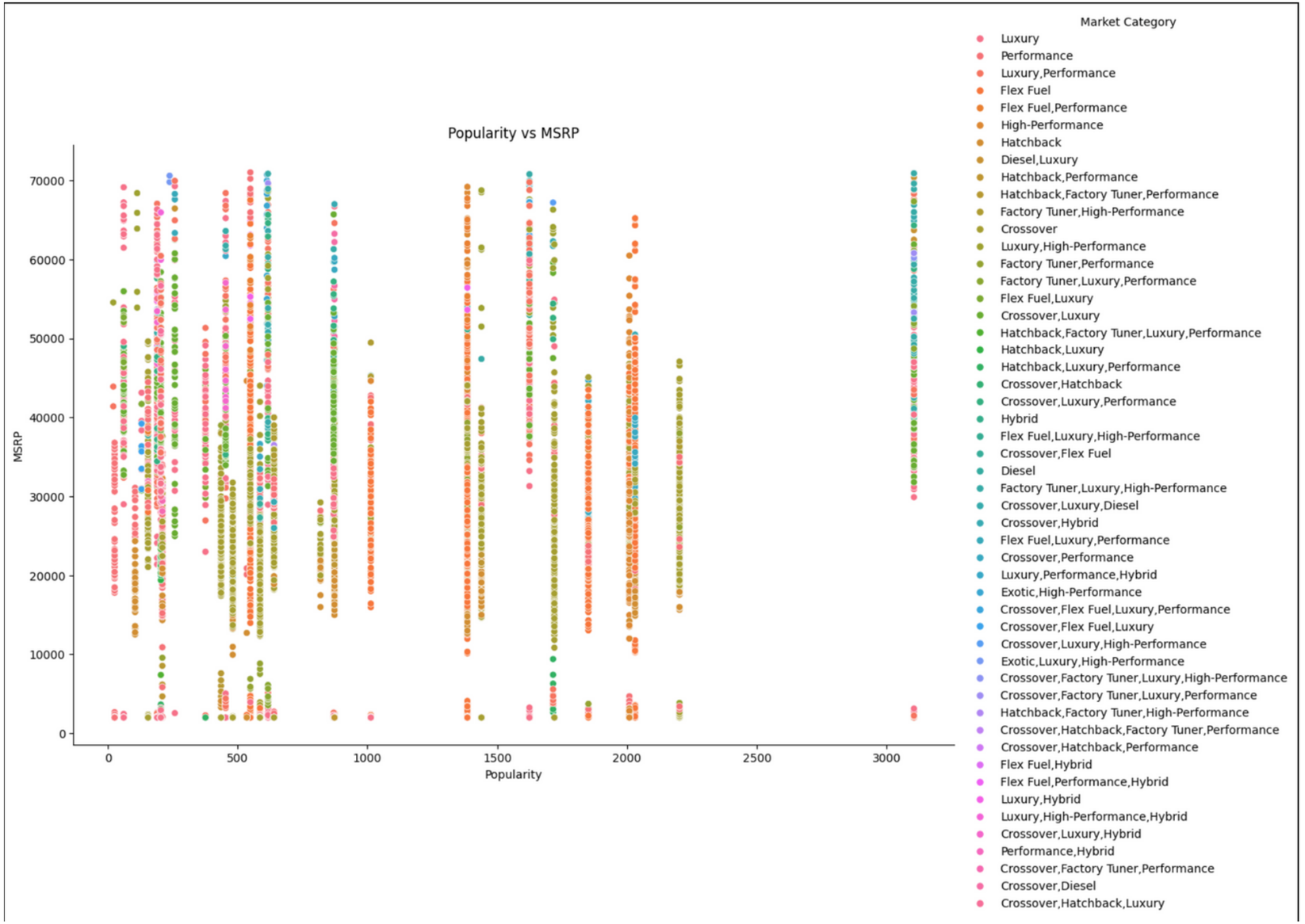
```
sns.relplot( x=categories[3], y=categories[4], data=data, hue=categories[0], height=8, aspect=1.5)
plt.title( f'{categories[3]} vs {categories[4]}' )

plt.show()
```

```
sns.relplot( x=categories[3], y=categories[4], data=data, hue=categories[2], height=8, aspect=1.5)
plt.title( f'{categories[3]} vs {categories[4]}' )

plt.show()
```


Make
Count
1000
800
600
400
200
0
Audi
FIAT
Mercedes-Benz
Chrysler
Nissan
Volvo
Mazda
Mitsubishi
Alfa Romeo
Toyota
Pontiac
Porsche
Saab
GMC
Hyundai
Plymouth
Honda
Oldsmobile
Suzuki
Cadillac
Kia
Chevrolet
Dodge
Lincoln
Subaru
Volkswagen
Buick
Acura
Land Rover
Lotus
Lexus
Infiniti
Scion
Genesis
Maserati
HUMMER
Make


Market Category
Count
2000
1500
1000
500
0
Luxury
Performance
Luxury,Performance
Flex Fuel
Flex Fuel,Performance
High-Performance
Hatchback
Diesel,Luxury
Hatchback,Performance
Hatchback,Factory Tuner,Performance
Factory Tuner,High-Performance
Crossover
Luxury,High-Performance
Factory Tuner,Performance
Factory Tuner,Luxury,Performance
Flex Fuel,Luxury
Crossover,Luxury
Hatchback,Factory Tuner,Luxury,Performance
Hatchback,Luxury
Hatchback,Luxury,Performance
Crossover,Hatchback
Crossover,Luxury,Performance
Hybrid
Flex Fuel,Luxury,High-Performance
Crossover,Flex Fuel
Diesel
Factory Tuner,Luxury,High-Performance
Crossover,Luxury,Diesel
Crossover,Hybrid
Flex Fuel,Luxury,Performance
Crossover,Performance
Luxury,Performance,Hybrid
Exotic,High-Performance
Crossover,Flex Fuel,Luxury,Performance
Crossover,Flex Fuel,Luxury
Crossover,Luxury,High-Performance
Exotic,Luxury,High-Performance
Crossover,Factory Tuner,Luxury,High-Performance
Crossover,Factory Tuner,Luxury,Performance
Hatchback,Factory Tuner,High-Performance
Crossover,Hatchback,Factory Tuner,Performance
Crossover,Hatchback,Performance
Flex Fuel,Hybrid
Flex Fuel,Performance,Hybrid
Luxury,Hybrid
Luxury,High-Performance,Hybrid
Crossover,Luxury,Hybrid
Performance,Hybrid
Crossover,Factory Tuner,Performance
Crossover,Diesel
Crossover,Hatchback,Luxury
Market Category


Popularity vs MSRP
MSRP
Popularity
0
10000
20000
30000
40000
50000
60000
70000
0
500
1000
1500
2000
2500
3000
Make
Audi
FIAT
Mercedes-Benz
Chrysler
Nissan
Volvo
Mazda
Mitsubishi
Alfa Romeo
Toyota
Pontiac
Porsche
Saab
GMC
Hyundai
Plymouth
Honda
Oldsmobile
Suzuki
Cadillac
Kia
Chevrolet
Dodge
Lincoln
Subaru
Volkswagen
Buick
Acura
Land Rover
Lotus
Lexus
Infiniti
Scion
Genesis
Maserati
HUMMER


Popularity vs MSRP
MSRP
Popularity
0
10000
20000
30000
40000
50000
60000
70000
0
500
1000
1500
2000
2500
3000
Market Category
Luxury
Performance
Luxury,Performance
Flex Fuel
Flex Fuel,Performance
High-Performance
Hatchback
Diesel,Luxury
Hatchback,Performance
Hatchback,Factory Tuner,Performance
Factory Tuner,High-Performance
Crossover
Luxury,High-Performance
Factory Tuner,Performance
Factory Tuner,Luxury,Performance
Flex Fuel,Luxury
Crossover,Luxury
Hatchback,Factory Tuner,Luxury,Performance
Hatchback,Luxury
Hatchback,Luxury,Performance
Crossover,Hatchback
Crossover,Luxury,Performance
Hybrid
Flex Fuel,Luxury,High-Performance
Crossover,Flex Fuel
Diesel
Factory Tuner,Luxury,High-Performance
Crossover,Luxury,Diesel
Crossover,Hybrid
Flex Fuel,Luxury,Performance
Crossover,Performance
Luxury,Performance,Hybrid
Exotic,High-Performance
Crossover,Flex Fuel,Luxury,Performance
Crossover,Flex Fuel,Luxury
Crossover,Luxury,High-Performance
Exotic,Luxury,High-Performance
Crossover,Factory Tuner,Luxury,High-Performance
Crossover,Factory Tuner,Luxury,Performance
Hatchback,Factory Tuner,High-Performance
Crossover,Hatchback,Factory Tuner,Performance
Crossover,Hatchback,Performance
Flex Fuel,Hybrid
Flex Fuel,Performance,Hybrid
Luxury,Hybrid
Luxury,High-Performance,Hybrid
Crossover,Luxury,Hybrid
Performance,Hybrid
Crossover,Factory Tuner,Performance
Crossover,Diesel
Crossover,Hatchback,Luxury

Insights

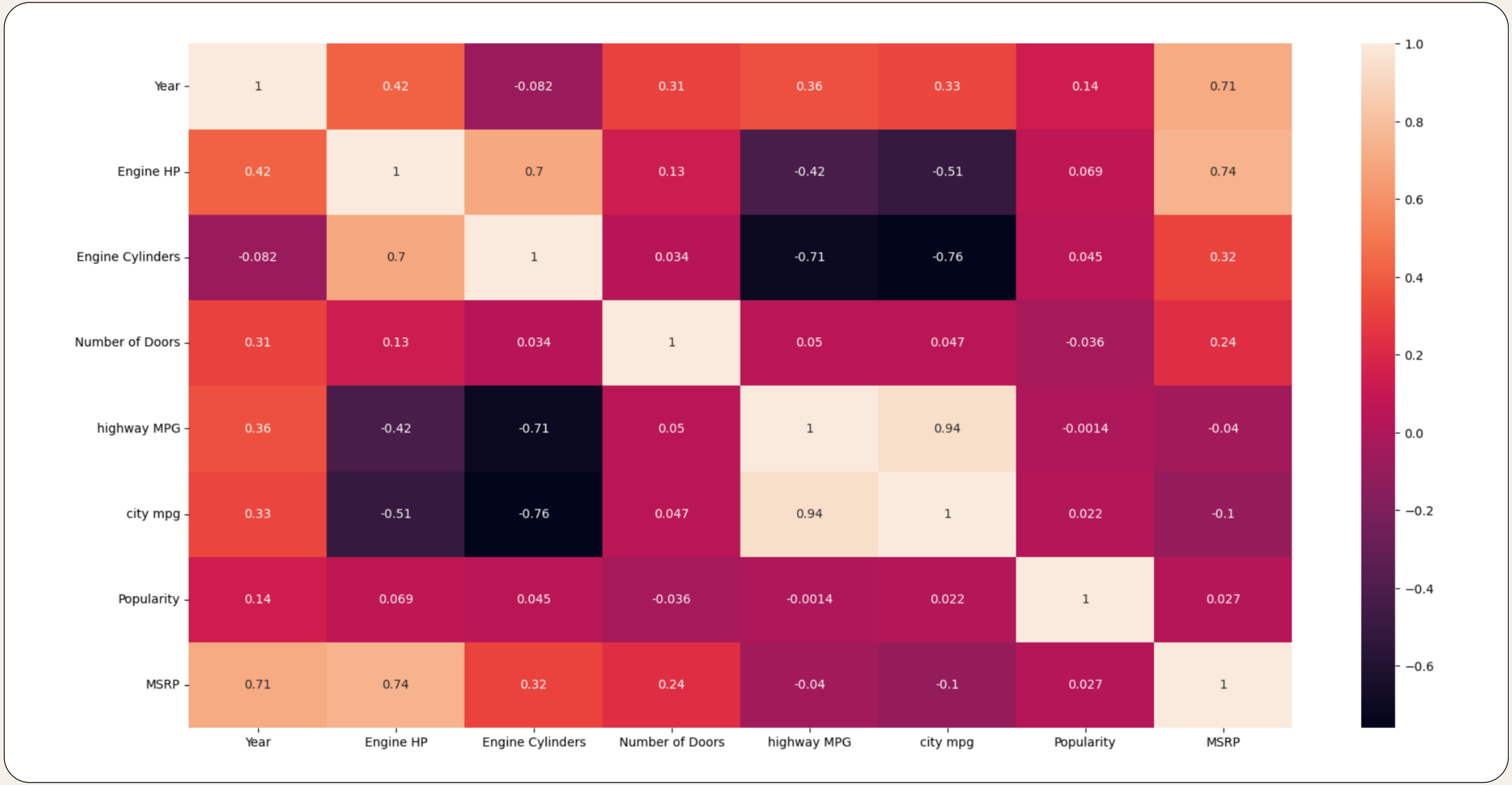
- Through univariate analysis of the 'market category', it became evident that the top 10 categories out of 51 collectively account for nearly 90% of the total.
- From the bivariate analysis of ‘MRSP’ and ‘Popularity’ based on ‘Make’, it is observed that the brand of the has the largest impact on popularity and price of a car as there was a clear difference in the scatter plot.
- But in the case of market category, there was no significant observation from scatter plot as the categories were well distributed in the graph.

Correlation

Approach

- The Correlation of a data frame can be found directly by using the corr() function.
- corr() function can only be applied on numerical columns, select_dtypes() is used to filter out the numerical columns.
- This only returns a dataframe with correlation coefficients.
- So heatmap() is applied upon this for visualization.

```
plt.figure( figsize=(20,10))
sns.heatmap( data.select_dtypes(include=['int64','float64']).corr(), annot=True )
plt.show()
```


Year
Engine HP
Engine Cylinders
Number of Doors
highway MPG
city mpg
Popularity
MSRP
1 0.42 -0.082 0.31 0.36 0.33 0.14 0.71
0.42 1 0.7 0.13 -0.42 -0.51 0.069 0.74
-0.082 0.7 1 0.034 -0.71 -0.76 0.045 0.32
0.31 0.13 0.034 1 0.05 0.047 -0.036 0.24
0.36 -0.42 -0.71 0.05 1 0.94 -0.0014 -0.04
0.33 -0.51 -0.76 0.047 0.94 1 0.022 -0.1
0.14 0.069 0.045 -0.036 -0.0014 0.022 1 0.027
0.71 0.74 0.32 0.24 -0.04 -0.1 0.027 1
Year
Engine HP
Engine Cylinders
Number of Doors
highway MPG
city mpg
Popularity
MSRP
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Insights

- As mentioned before, the same observations was derived from the correlation matrix as well.
- None of the features had a significant impact on the ‘Popularity’ as the corr coefficient was close to zero which indicates a low level dependency of these features on popularity.
- The correlation was same for 'MSRP' as well, except for 'Engine HP', which demonstrated a significant impact with a correlation coefficient of 0.74, signifying a high level of dependency.
- Additionally, 'Year' played a role in influencing 'MSRP', and although it had a correlation coefficient of 0.14 with 'Popularity', it was the highest among the other features.