ISSN 2959-6149

Car Features and Prices Dataset

Mintao Hu

Abstract:

The goal of study is to investigate the relationship between these automobilecharacteristics and the Manufacturer Suggested Retail Price (MSRP) of automobiles, with a particular focus on identifying the most vital and largest-influencing factors. And the analysis using Python, the Pandas library, Seaborn, and additional data analysis tools into this project

Keywords: Car, price, data analysis

Introduction

The Car Features and Prices Dataset was selected for final project. The dataset contains 11,914 car records spanning the years 1990 to 2017. It includes information on associated vehicle features, including manufacturer (Make), model (Model), and other relevant information, in addition to a variety of car brands and models.

Access to this dataset is certainly possible for the instructor via the subsequent URL: kaggle-link

Additionally, it is available for direct distribution at the following URL: direct-download

The goal of study is to investigate the relationship between these automobilecharacteristics and the Manufacturer Suggested Retail Price (MSRP) of automobiles, with a particular focus on identifying the most vital and largest-influencing factors. By means of this analysis, our intention is to furnish car manufacturers, market analysts, and consumers with beneficial recommendations that will assist them in comprehending market dynamics more effectively and facilitating more informed vehicle purchases.

In order to accomplish this objective,initially performed data cleansing, addressed missing values and outliers, and verified the data’s quality and accuracy are very important. Subsequently, descriptive statistical analysis was employed to determine the fundamental attributes of the dataset. An in-depth examination of the influence of diverse automobile features on MSRP was conducted utilizing a range of data visualization methodologies, including box plots, violin plots, and scatter plots .Analyzing the impact of engine horsepower, car brand, and model on the suggested retail price of automobiles, gived particular consideration to these variables individually or in combination.

In conclusion, in light of the mentioned evaluation,I conducted a thorough assessment of the influence that each factor had on price and assigned them corresponding scores.

By incorporating Python, the Pandas library, Seaborn, and additional data analysis tools into this final project, not only enhanced my comprehension of data analysis methodologies but also implemented the theoretical concepts acquired in the classroom to address real-world challenges.

Dataset Overview

The dataset includes 11,914 car records from 1990 to 2017, with each record covering the following fields:

- Make: The car manufacturer
- Model: The car model
- Year: The production year
- Engine Fuel Type: The type of fuel used by the engine
- Engine HP: The horsepower of the engine
- Engine Cylinders: The number of cylinders in the engine
- Transmission Type: The type of transmission
- Driven_Wheels: The configuration of the driven wheels
- Number of Doors: The number of doors on the car
- Market Category: The market category of the vehicle
- Vehicle Size: The size of the vehicle
- Vehicle Style: The style of the vehicle
- Highway MPG: The fuel economy on the highway (miles per gallon)
- City MPG: The fuel economy in the city (miles per gallon)
- Popularity: The popularity of the brand
- MSRP: The Manufacturer Suggested Retail Price

##Analysis Process

###Data Cleaning It is essential to prepare our dataset in the first phase of our Exploratory Data research (EDA) to make sure it is readable, tidy, and manageable for additional research. We make use of the pandas package to do this. To deal with the data in a structured manner, specifically a DataFrame, the first step is to read the dataset using pandas.

```
import pandas as pd
from google.colab import files
import requests
import matplotlib.pyplot as plt
import seaborn as sns
file_id = '1ODgDhp6AL7KIZlAhWuUnG2fnvTRxisYN'
download_url = f'https://drive.google.com/uc?export=download&id={file_id}'
response = requests.get(download_url, stream=True)
if response.status_code == 200:
filename = 'data.csv'
with open(filename, 'wb') as file:
file.write(response.content)
else:
print('Failed to download file.')
df = pd.read_csv("data.csv")
print(df.head())#this line can help us to understand the sturcture of the data
print(df.info())#this line is use to view the type of the data, and check wrong or blank data
```

```
Make       Model  Year            Engine Fuel Type  Engine HP  \
0 BMW  1 Series M  2011  premium unleaded (required)  335.0
1 BMW    1 Series  2011  premium unleaded (required)  300.0
2 BMW    1 Series  2011  premium unleaded (required)  300.0
3 BMW    1 Series  2011  premium unleaded (required)  230.0
4 BMW    1 Series  2011  premium unleaded (required)  230.0
Engine Cylinders Transmission Type      Driven_Wheels  Number of Doors  \
0 6.0          MANUAL  rear wheel drive           2.0
1 6.0          MANUAL  rear wheel drive           2.0
2 6.0          MANUAL  rear wheel drive           2.0
3 6.0          MANUAL  rear wheel drive           2.0
4 6.0          MANUAL  rear wheel drive           2.0
Market Category Vehicle Size Vehicle Style  \
0 Factory Tuner,Luxury,High-Performance       Compact  Coupe
1 Luxury,Performance      Compact   Convertible
2 Luxury,High-Performance      Compact        Coupe
3 Luxury,Performance      Compact        Coupe
4 Luxury      Compact   Convertible
highway MPG  city mpg  Popularity   MSRP
0 26        19        3916  46135
1 28        19        3916  40650
2 28        20        3916  36350
3 28        18        3916  29450
4 28        18        3916  34500
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11914 entries, 0 to 11913
Data columns (total 16 columns):
#   Column             Non-Null Count  Dtype
---  ------            --------------  -----
0 Make              11914 non-null  object
1 Model             11914 non-null  object
2 Year              11914 non-null  int64
3 Engine Fuel Type   11911 non-null  object
4 Engine HP          11845 non-null  float64
5 Engine Cylinders   11884 non-null  float64
6 Transmission Type  11914 non-null  object
7 Driven_Wheels      11914 non-null  object
8 Number of Doors    11908 non-null  float64
9 Market Category    8172 non-null   object
10  Vehicle Size      11914 non-null  object
11  Vehicle Style     11914 non-null  object
12  highway MPG       11914 non-null  int64
13  city mpg          11914 non-null  int64
14  Popularity        11914 non-null  int64
15  MSRP              11914 non-null  int64
dtypes: float64(3), int64(5), object(8)
memory usage: 1.5+ MB
None
```

Initially, the dataset comprises multiple columns, including the car's Make, Model, Year, Engine Fuel Type, Engine HP, Engine Cylinders, Transmission Type, Driven Wheels, Number of Doors, Market Category, Vehicle Size, Vehicle Style, highway MPG, city mpg, Popularity, and MSRP.

Upon a preliminary inspection of the data, we found that there are missing values, particularly in the Engine HP, Engine Cylinders, Number of Doors, and Market Category columns. Handling missing data is a critical step in the data cleaning process, as it can significantly affect subsequent analyses and model development. For addressing these missing values, we adopted the following strategies:

For Number of Doors and Engine Cylinders, as these attributes are typically fixed based on the car's model, we decided to fill in the missing values using the mode within each model. For Engine HP, we considered a more nuanced approach, seeking the most similar models to those with missing values and then using the mode of these similar models for imputation. This method assumes that similar models will also be similar in terms of engine horsepower. The Market Category has a large number of missing values. Given the unstructured nature of this column and its high rate of missing data, we decided it might be ignored in subsequent analyses or handled according to specific analytical needs.

```
# Handling missing values for ‘Number of Doors’
# Filling missing values with the median of ‘Number of Doors’
df[‘Number of Doors’].fillna(df[‘Number of Doors’].median(), inplace=True)
# For ‘Engine HP’ and ‘Engine Cylinders’, fill missing values based on the mode of the respective ‘Model’
for column in [‘Engine HP’, ‘Engine Cylinders’]:
df[column] = df.groupby(‘Model’)[column].transform(lambda x: x.fillna(x.mode()[0] if not x.mode().empty else x.median()))
# Save the cleaned dataset
df.to_csv(“cleaned_data.csv”, index=False)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
Year     Engine HP  Engine Cylinders  Number of Doors  \
count  11914.000000  11876.000000      11885.000000     11914.000000
mean    2010.384338    249.262883          5.628355         3.436377
std        7.579740    109.101188          1.781233         0.881184
min     1990.000000     55.000000          0.000000         2.000000
25%     2007.000000    170.000000          4.000000         2.000000
50%     2015.000000    227.000000          6.000000         4.000000
75%     2016.000000    300.000000          6.000000         4.000000
max     2017.000000   1001.000000         16.000000         4.000000
highway MPG      city mpg    Popularity          MSRP
count  11914.000000  11914.000000  11914.000000  1.191400e+04
mean      26.637485     19.733255   1554.911197  4.059474e+04
std        8.863001      8.987798   1441.855347  6.010910e+04
min       12.000000      7.000000      2.000000  2.000000e+03
25%       22.000000     16.000000    549.000000  2.100000e+04
50%       26.000000     18.000000   1385.000000  2.999500e+04
75%       30.000000     22.000000   2009.000000  4.223125e+04
```

```
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
/usr/local/lib/python3.10/dist-packages/numpy/lib/nanfunctions.py:1215: RuntimeWarning: Mean of empty slice
return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

###Descriptive Analysis: After preparing the data initially, we conducted a descriptive statistical analysis as part of our study to fully comprehend the basic statistical properties of the variables in the dataset. This stage is critical for exposing the fundamental distribution of the data and offering necessary background information for more in-depth investigation.

For numerical variables, we started by completing a statistical summary that included mean, median, standard deviation, minimum, and maximum value computations. The pandas library’s describe() function made this process easier. The distribution of categorical variables was evaluated by computing the frequency of occurrences for every category.

```
c_df = pd.read_csv(“cleaned_data.csv”)
# Statistical summary for numerical variables
numerical_summary = c_df.describe()
# Display the statistical summary
print(numerical_summary)
# Frequency of occurrences for each category in categorical variables
categorical_summary = c_df.select_dtypes(include=[‘object’]).apply(pd.Series.value_counts)
# Display the categorical summary
print(categorical_summary)
```

```
max      354.000000    137.000000    5657.000000  2.065902e+06
Make  Model  Engine Fuel Type  Transmission Type  \
1 Series           NaN   16.0              NaN               NaN
1 Series M         NaN    1.0              NaN               NaN
100              NaN   15.0              NaN               NaN
124 Spider         NaN    3.0              NaN               NaN
190-Class          NaN    6.0              NaN               NaN
..               ...   ...              ...               ...
regular unleaded   NaN    NaN            7172.0               NaN
tC               NaN   14.0              NaN               NaN
xA               NaN    6.0              NaN               NaN
xB               NaN   10.0              NaN               NaN
xD               NaN   10.0              NaN               NaN
Driven_Wheels  Market Category  Vehicle Size  Vehicle Style
1 Series                NaN            NaN          NaN           NaN
1 Series M                NaN            NaN          NaN           NaN
100                     NaN            NaN          NaN           NaN
124 Spider                NaN            NaN          NaN           NaN
190-Class                 NaN            NaN          NaN           NaN
..                        ...            ...          ...           ...
regular unleaded            NaN            NaN          NaN           NaN
tC                        NaN            NaN          NaN           NaN
xA                        NaN            NaN          NaN           NaN
xB                        NaN            NaN          NaN           NaN
xD                        NaN            NaN          NaN           NaN
[1069 rows x 8 columns]
```

This analysis reveals that the dataset includes various vehicles ranging from the years 1990 to 2017.

• Year: The manufacturing dates of the vehicles range from 1990 to 2017, with an average year around 2010. This indicates that our dataset leans towards newer vehicle models, providing a strong basis for analysis.

• Engine HP: The engine horsepower varies significantly, ranging from 55 to 1001, with an average horsepower of approximately 249. This variation indicates that the dataset includes a wide range of vehicles from high-performance sports cars to standard passenger cars.

• Engine Cylinders: The number of engine cylinders varies from 0 to 16, with an average close to 6 cylinders. Vehicles with 0 cylinders are electric vehicles.

• Number of Doors, MPG, and Popularity: The dataset contains vehicles with 2 to 4 doors, an average highway MPG of 26.63, an average city MPG of 19.73, and a wide range of popularity scores. These factors can provide more directions for our further research.

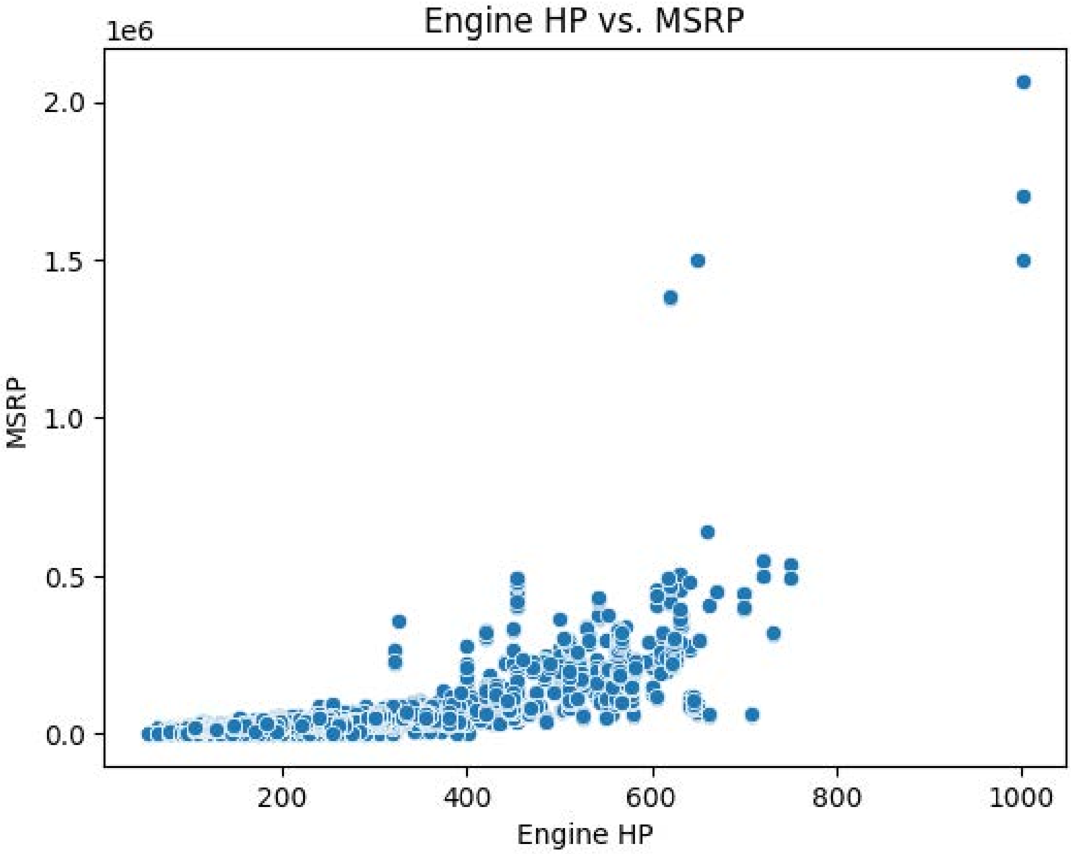
• In addition to this, the dataset also includes categorical variables such as manufacturer, model, engine fuel type, transmission type, driven wheels, market category, vehicle size, and vehicle style.

###Data Visualization: A variety of visualization approaches were used to obtain deeper insights. To depict the distribution of MSRP across several parameters, including Engine HP, Engine Cylinders, and Vehicle Style, box plots, violin plots, and scatter plots were employed. This made it easier to spot trends, anomalies, and the variation in costs across several categories.

Engine HP vs. MSRP

```
# Engine HP vs. MSRP
sns.scatterplot(x='Engine HP', y='MSRP', data=df)
```

```
plt.xlabel(‘Engine HP’)
plt.ylabel(‘MSRP’)
plt.title(‘Engine HP vs. MSRP’)
plt.show()
```



It can be performed a thorough examination of the link between an automobile’s engine horsepower (Engine HP) and the manufacturer’s suggested retail price (MSRP) using the scatter plot as a basis:

1. Positive Correlation: A car’s suggested retail price (MSRP) tends to climb in tandem with an increase in engine horsepower. This illustrates how higher vehicle configurations, improved performance, and higher manufacturing costs are typically linked to more powerful engines.

2. Variability: The MSRP is more variable at the lower end of engine horsepower and becomes less variable as engine horsepower increases. The bulk of cars with minimal horsepower are priced under $50,000.

3. Outliers: High horsepower and high MSRP outliers could be indicative of luxury or high-end models available in the market. These cars could cost a lot more than other comparable cars because they offer extra amenities like cutting-edge technology, opulent interiors, or limited edition designs.

4. Data Density: The bulk of cars sold in the market have comparatively lower horsepower, as indicated by the data points being densest in the lower horsepower range. This could be connected to the general customer demand for affordable, mid-performance cars.

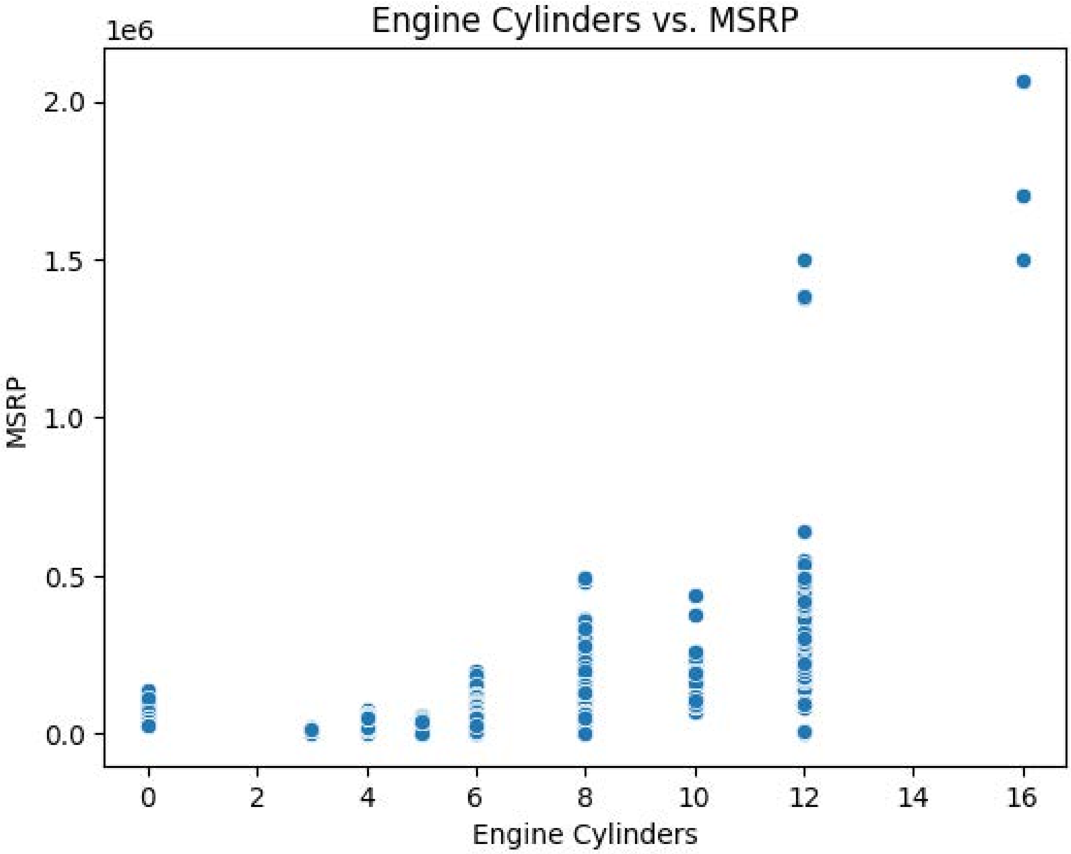
5. High-End Market: The lower market share of high-performance, high-priced automobiles may be the reason for the lack of data points in the area of high horsepower and high MSRP. Usually, these cars are made for particular customer segments, such the wealthy or those who love sports cars.

6. Non-linear Pattern: The MSRP distribution is more diversified in the mid-horsepower range, despite the fact that the overall trend is favorably connected. This implies that the price of the car is also influenced by other elements, such as brand, model, and technical advancement. Therefore, in order to more precisely grasp the relationship between automobile costs and automotive attributes, we need to take into account more variables while studying prices.

These findings lead us to the conclusion that there is more to the link between car engine horsepower and MSRP than just a straightforward linear one. In order to more precisely examine the relationship between automobile costs and automotive features, additional variables, such as vehicle brand and model, must be taken into account when performing more thorough data analysis. Furthermore, because the costs of these vehicles may be impacted by a range of non-performance aspects, enough consideration should be given to the unique characteristics of the high-end market.

####Engine Cylinders vs. MSRP

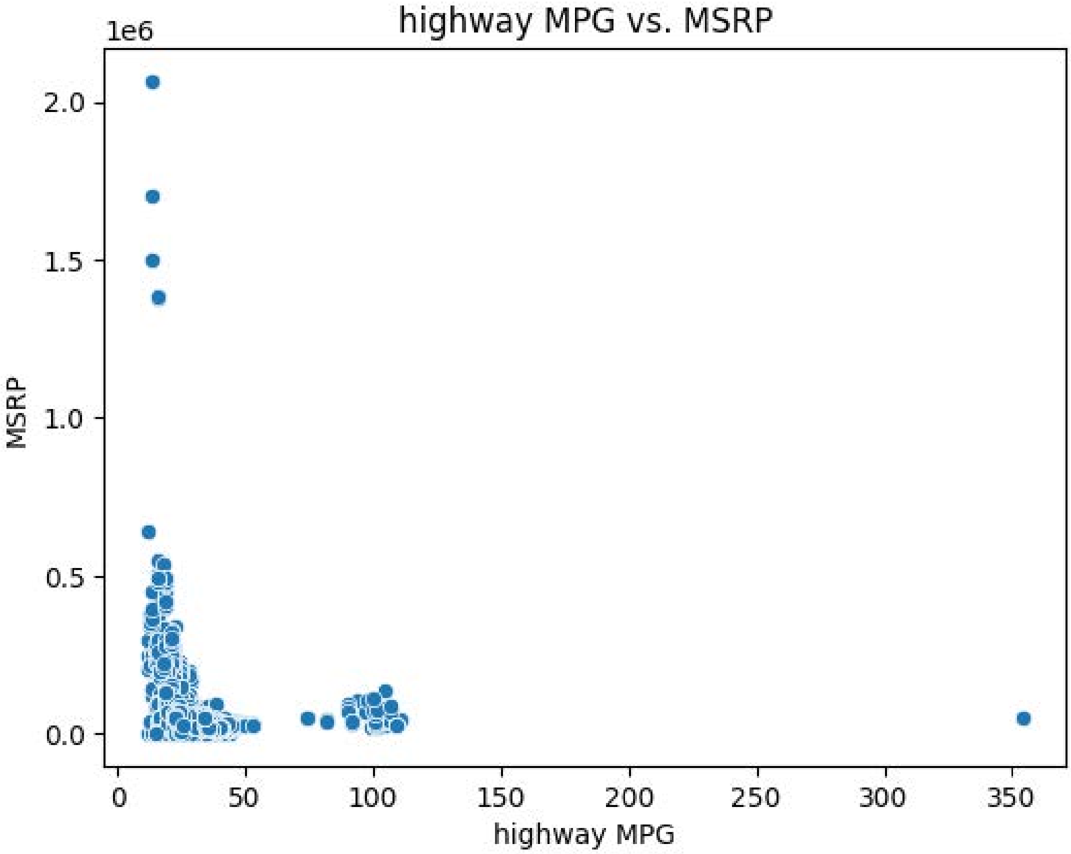
```
sns.scatterplot(x='Engine Cylinders', y='MSRP', data=df)
plt.xlabel('Engine Cylinders')
plt.ylabel('MSRP')
plt.title('Engine Cylinders vs. MSRP')
plt.show()
```



The scatter plot shows that the manufacturer's suggested retail price (MSRP) and the number of engine cylinders (Engine Cylinders) have a positive link. The MSRP of a vehicle usually increases with the number of cylinders, especially for vehicles with 10, 12, and 16 cylinders. As a prevalent configuration in the market, the bulk of cars are centered in the 4 to 8 cylinder range. In general, vehicles with fewer cylinders—such as those with two or three—are less expensive.

####highway MPG vs. MSRP

```
sns.scatterplot(x='highway MPG', y='MSRP', data=df)
plt.xlabel('highway MPG')
plt.ylabel('MSRP')
plt.title('highway MPG vs. MSRP')
plt.show()
```

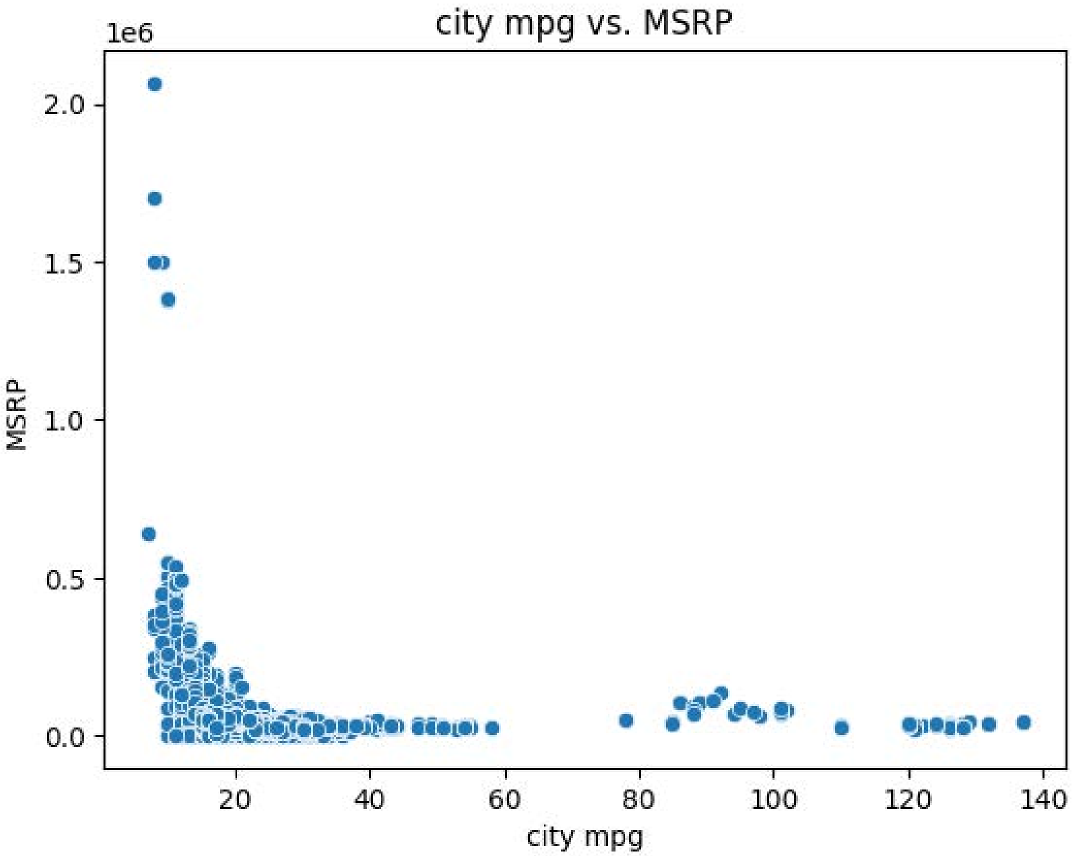


The scatter plot shows that the association between the manufacturer's suggested retail price (MSRP) and highway miles per gallon (highway MPG) is not what we had anticipated. In general, it makes sense to think that cars with greater fuel efficiency would be more cost-effective. Nevertheless, the plot indicates that a number of very fuel-efficient cars also have very expensive MSRPs, which I assume is because these are hybrid or electric vehicle types.

The range of most conventional fuel vehicles is reflected by the concentration of vehicles in the lower MPG area. There are a few prominent price peaks in the low MPG range, which are brought about by high-performance or luxury brands that are substantially more costly than other car kinds even if they have worse fuel efficiency.

In conclusion, there is no discernible pattern in the scatter plot, indicating that fuel efficiency has little bearing on MSRP.

```
####city mpg vs. MSRP
sns.scatterplot(x='city mpg', y='MSRP', data=df)
plt.xlabel('city mpg')
plt.ylabel('MSRP')
plt.title('city mpg vs. MSRP')
plt.show()
```

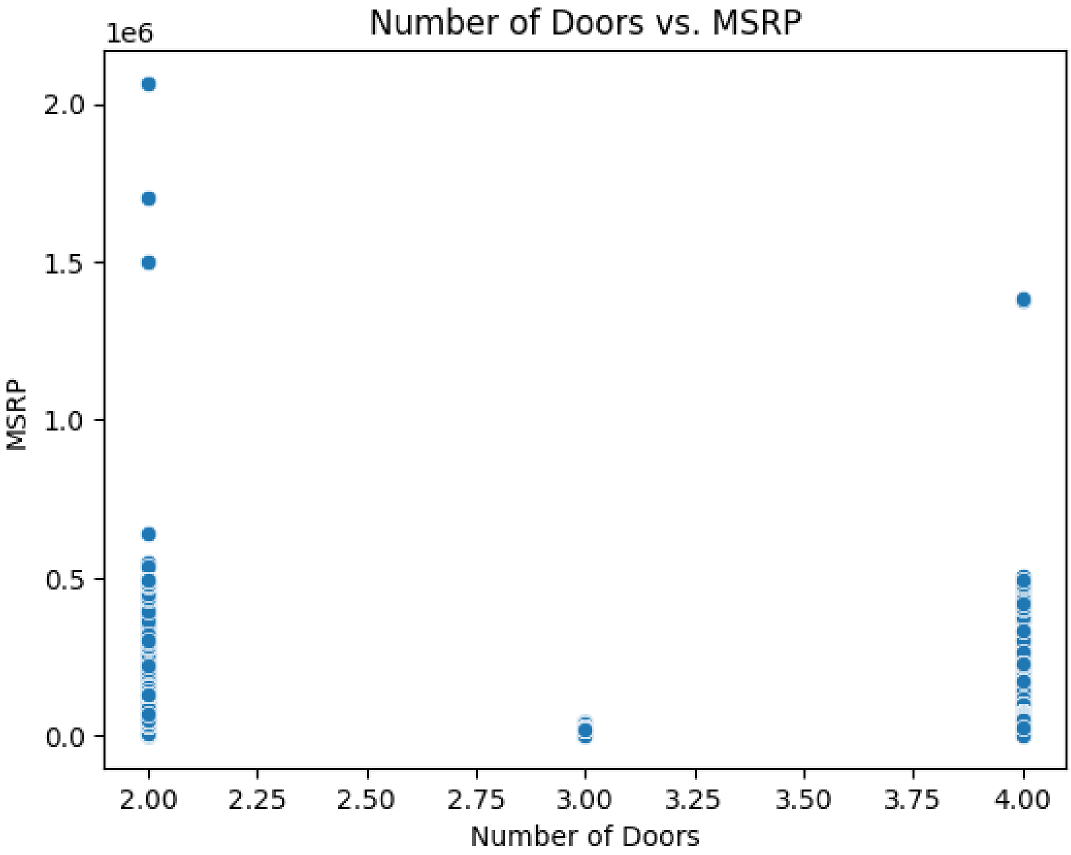


This scatter plot's data is primarily concentrated in the lower range of city miles per gallon (MPG), namely in the 20–30 mpg band. While the number of automobiles with higher city MPG is quite low—especially those that achieve above 60 mpg—their MSRP distribution is quite large. Some sparser data points, likely representing high-end cars on the market, may be seen in the higher MSRP regions of the scatter figure.

In conclusion, even while increased fuel economy typically translates into lower operating costs, there is no linear relationship between fuel economy and the cost of purchasing a car. Therefore, it can be concluded that this is not the main factor affecting the MSRP.

####Number of Doors vs. MSRP

```
sns.scatterplot(x='Number of Doors', y='MSRP', data=df)
plt.xlabel('Number of Doors')
plt.ylabel('MSRP')
plt.title('Number of Doors vs. MSRP')
plt.show()
```



We can infer from this scatter plot that there isn't a definite correlation between a car's MSRP (manufacturer's suggested retail price) and how many doors it has. The majority of cars have four doors, yet their MSRPs differ greatly, from very cheap to quite costly.

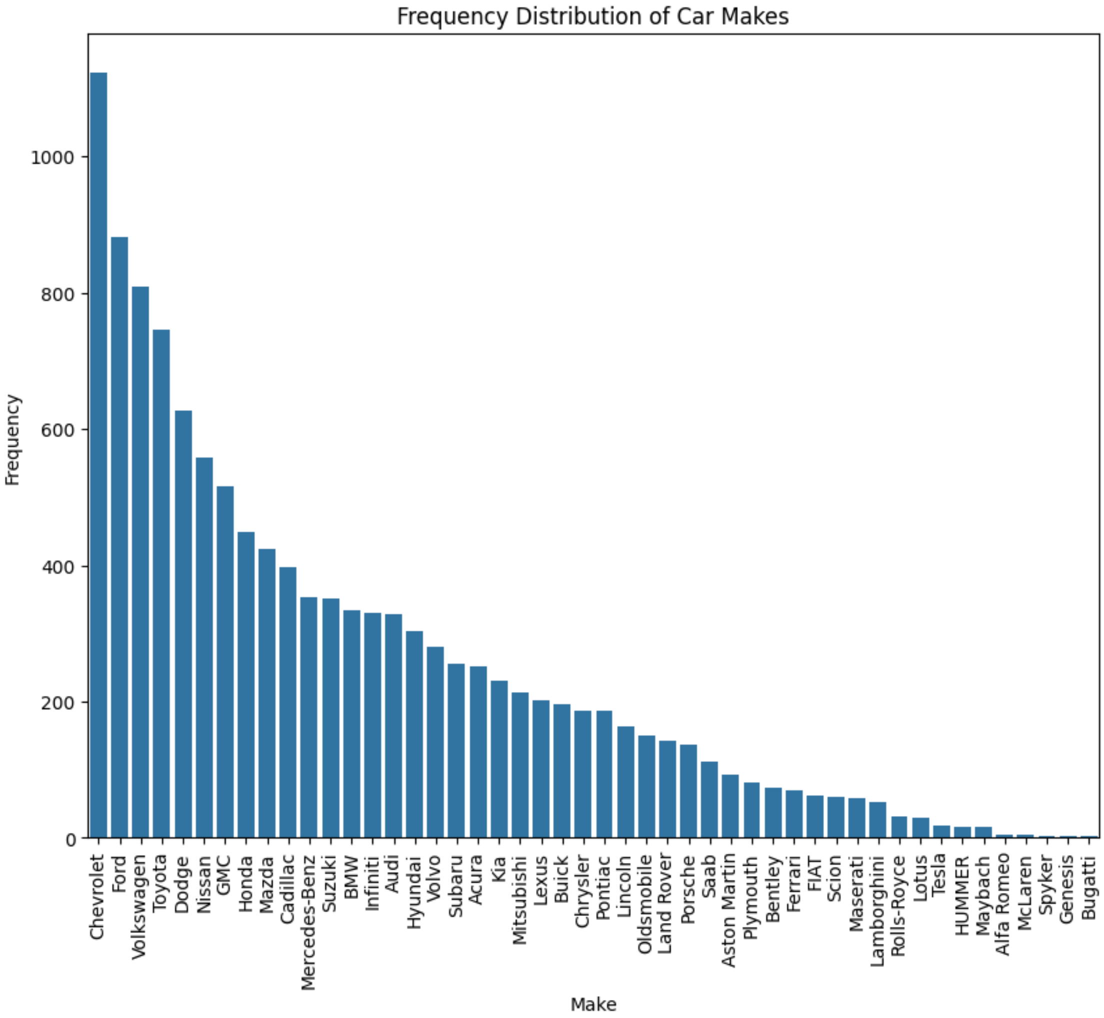
Since only a small percentage of cars have two doors—usually sports or high-performance models—their MSRP is greater than that of cars with multiple doors. We can thus presume that the exceedingly costly cars are either custom models or high-end luxury cars, which frequently include special features or performance characteristics that push the price to extraordinarily high levels.

In conclusion, the type of vehicle, brand positioning, and other amenities may have a greater influence on a car's pricing than its door count. When a car costs over a million dollars, its high cost is probably a result of its performance, design, and brand value.

####Frequency Distribution of Car Makes

The next step is studying categorical data, having finished the descriptive analysis of numerical data. One approach that is frequently used to determine how different categories are distributed within a dataset is frequency distribution analysis. We can determine the relationship between features and price by knowing which categories are more prevalent and which are more rare. To have a more intuitive view of the dataset's composition, we will specifically utilize the value_counts() method from the pandas package to calculate the occurrence frequency of each distinct category and visualize this data.

```
make_counts = c_df['Make'].value_counts()
plt.figure(figsize=(10,8))
sns.barplot(x=make_counts.index, y=make_counts.values)
plt.xticks(rotation=90)
plt.xlabel('Make')
plt.ylabel('Frequency')
plt.title('Frequency Distribution of Car Makes')
plt.show()
```



This frequency distribution chart shows the market share or popularity of several auto brands (Make). Top-ranking brands on the bar chart, like Toyota, Ford, Volkswagen, and Chevrolet, show how well-liked they are in the marketplace. As a result, these brands probably provide a wider range and quantity of automobile models that are accessible for purchase. Conversely, high-end brands such as Lamborghini, Ferrari, and McLaren may be seen at the bottom of the chart, indicating their comparatively little market share, which could be attributed to their high cost and small production volume.

This distribution might have something to do with the target markets and pricing policies of various companies. In order to cater to a wider range of consumers, mass-market companies usually provide models at moderate pricing, whereas luxury brands tend to concentrate more on the high-end market and offer vehicles that are highly priced, highly performing, and of superior quality.

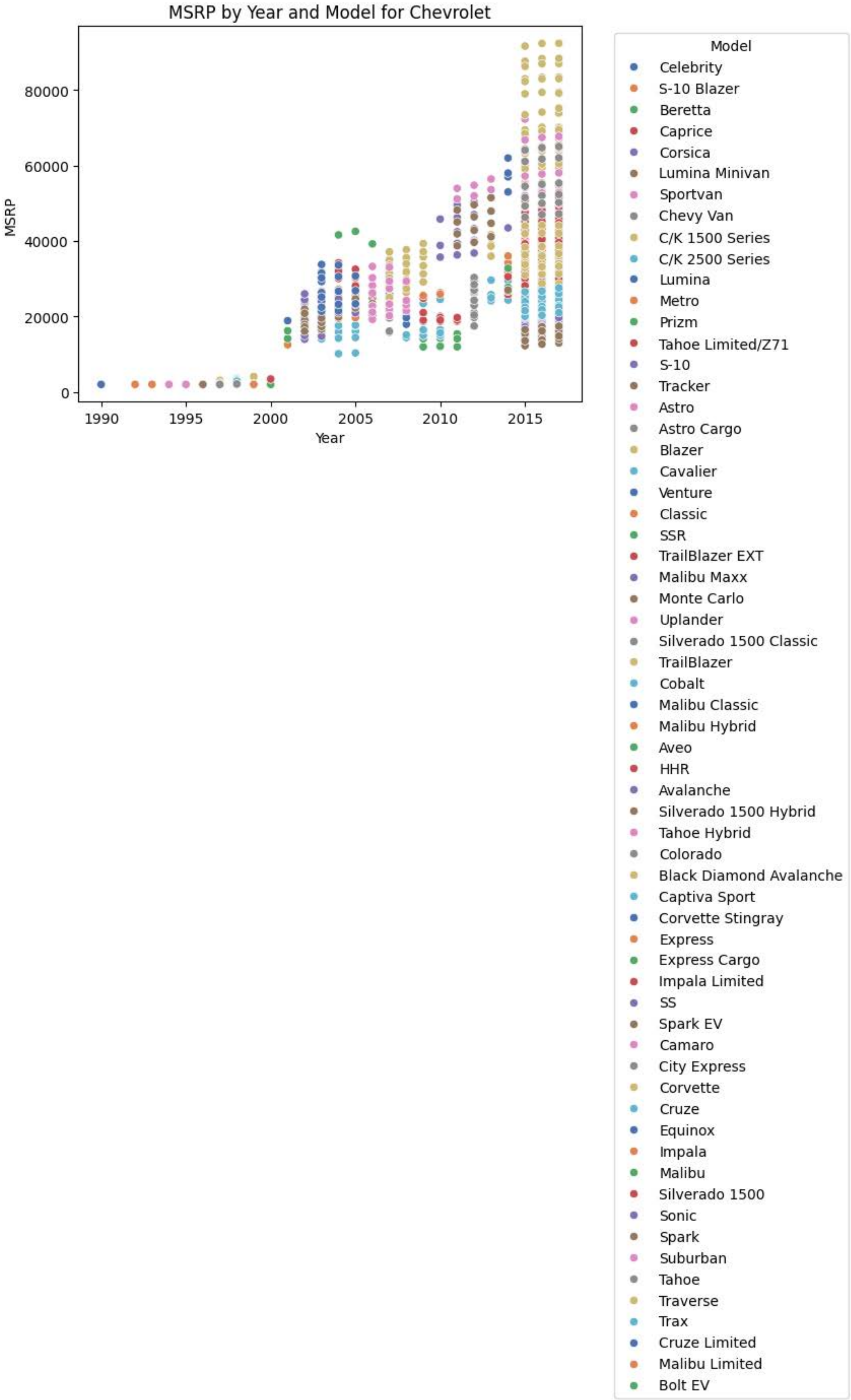
####MSRP by Year and Model

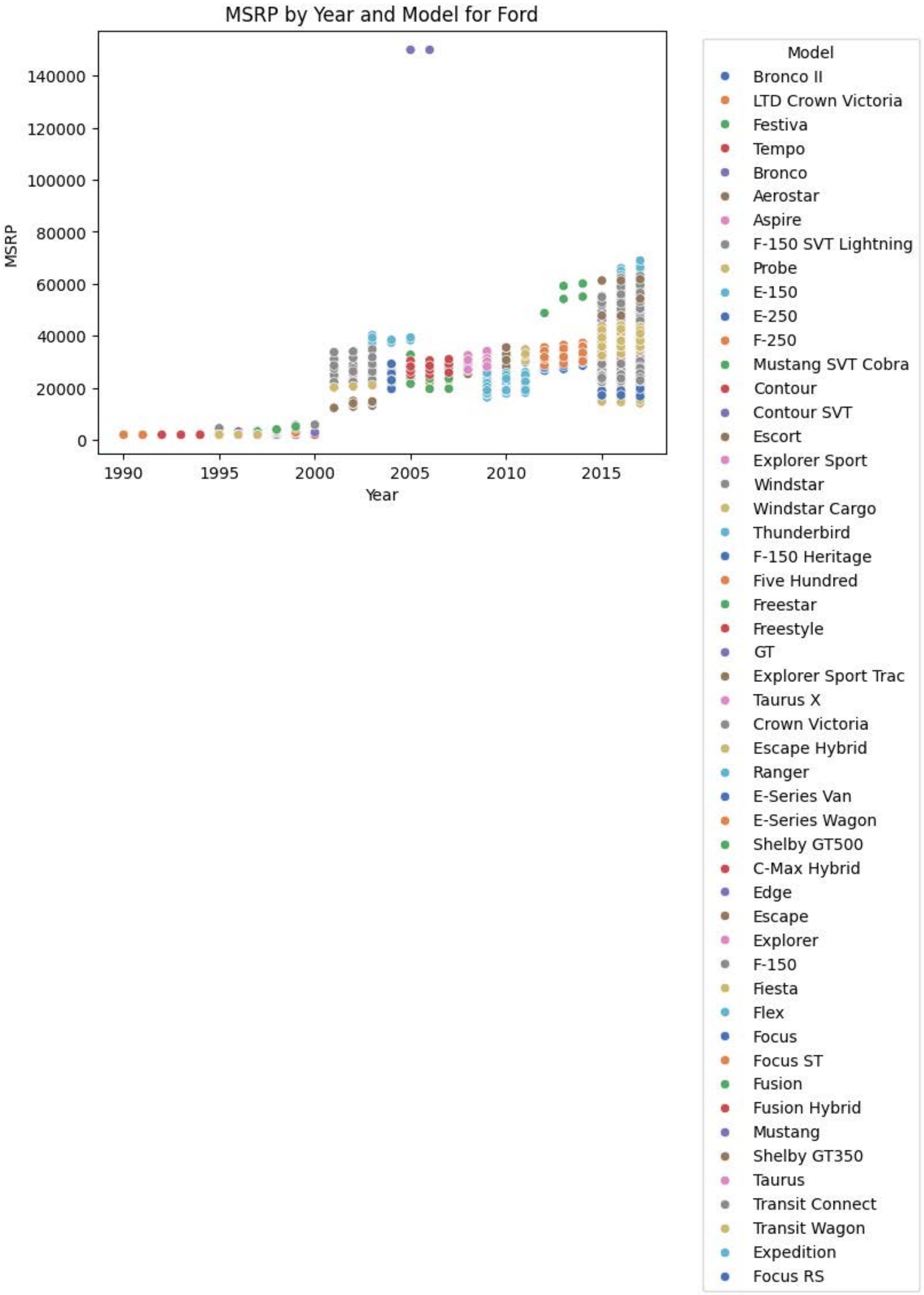
In order to delve deeper into the relationship between automakers and their Manufacturer Suggested Retail Price (MSRP), we have taken a more in-depth approach by performing targeted analyses on the different models manufactured by each brand. With the help of this approach, we can comprehend the variations in price tactics used by various brands or models as well as the possible causes of these variations. For example, some brands might concentrate mostly on producing expensive, high-performance or luxury vehicles, while others might concentrate on making affordable, low-cost vehicles.

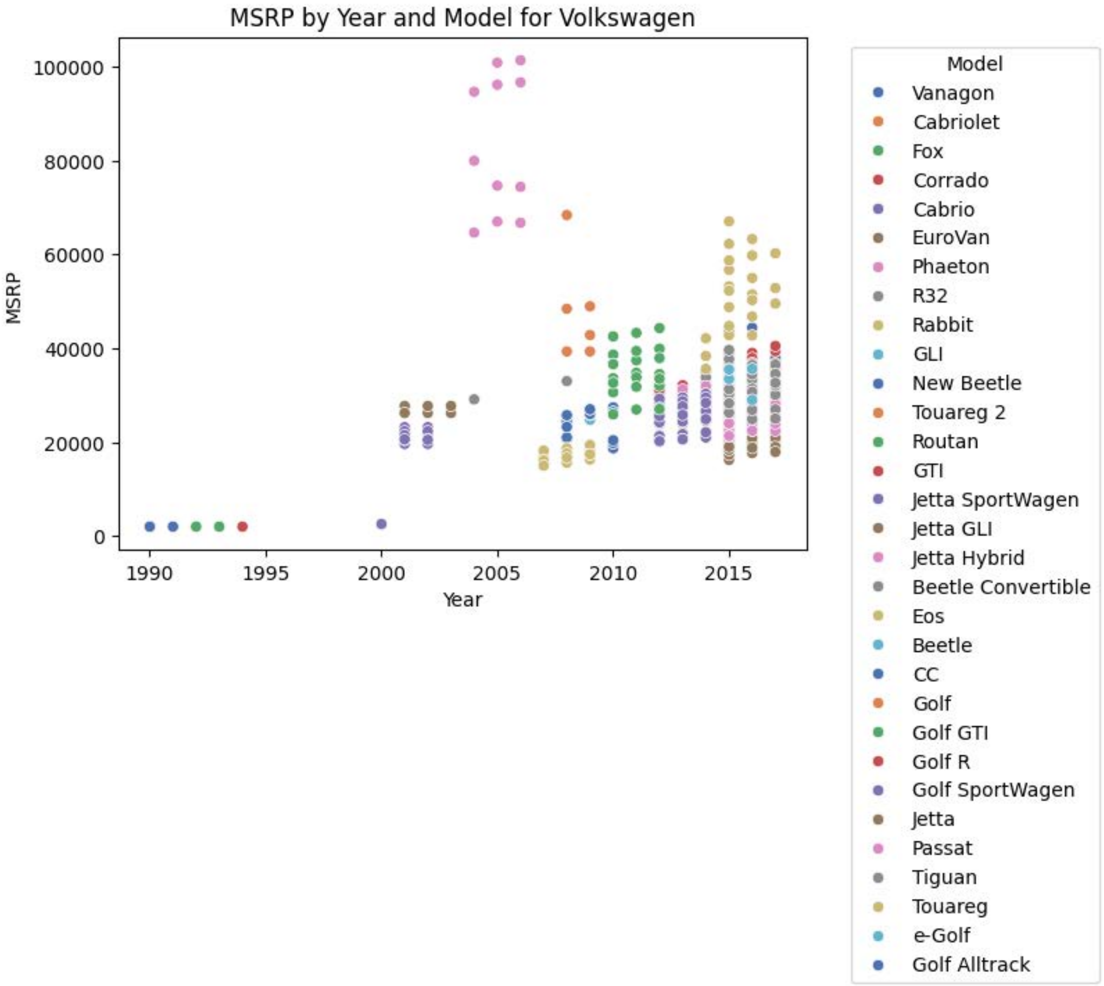
By using this study,we can further investigate each brand’s pricing tactics and target consumer groups, as well as more correctly portray each brand’s market positioning and competitive strength. To be more precise, we will first use the dataset to determine which five brands are the most and least popular. Next, we will do scatter plot studies based on the year and MSRP for the models of these brands. Using this method will enable us to investigate the pricing distribution of various models and track changes in their costs over time.

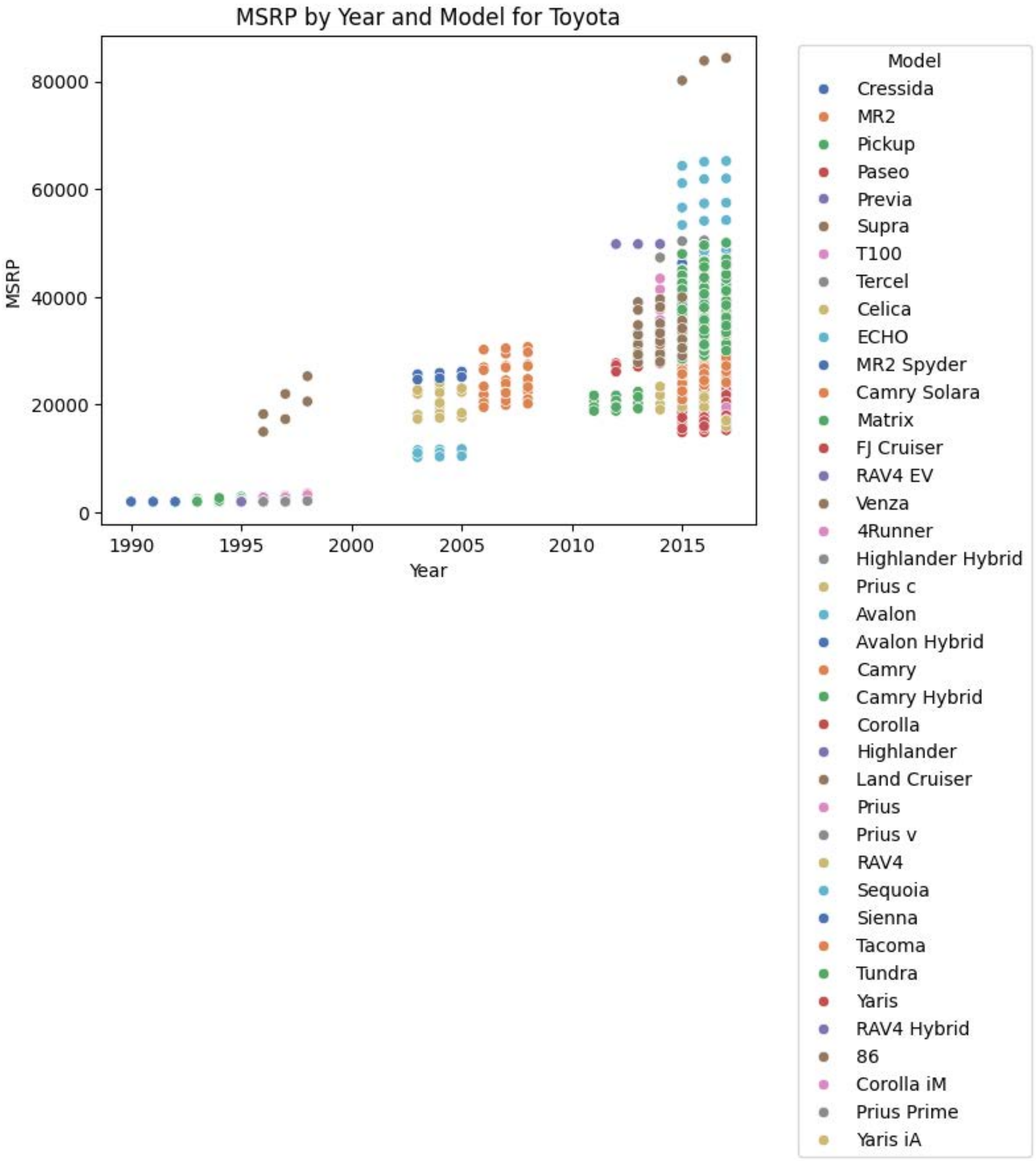
```
top_makes = make_counts.head(5).index.tolist()
bottom_makes = make_counts.tail(5).index.tolist()
selected_makes = top_makes + bottom_makes
filtered_df = df[df[‘Make’].isin(selected_makes)]
filtered_df.sort_values(by=[‘Make’, ‘Year’, ‘Model’], inplace=True)
for make in selected_makes:
make_df = filtered_df[filtered_df[‘Make’] == make]
sns.scatterplot(data=make_df, x=’Year’, y=’MSRP’, hue=’Model’, palette=’deep’, legend=’full’)
plt.title(f’MSRP by Year and Model for {make}’)
plt.xlabel(‘Year’)
plt.ylabel(‘MSRP’)
plt.legend(title=’Model’, bbox_to_anchor=(1.05, 1), loc=’upper left’)
plt.show()
```

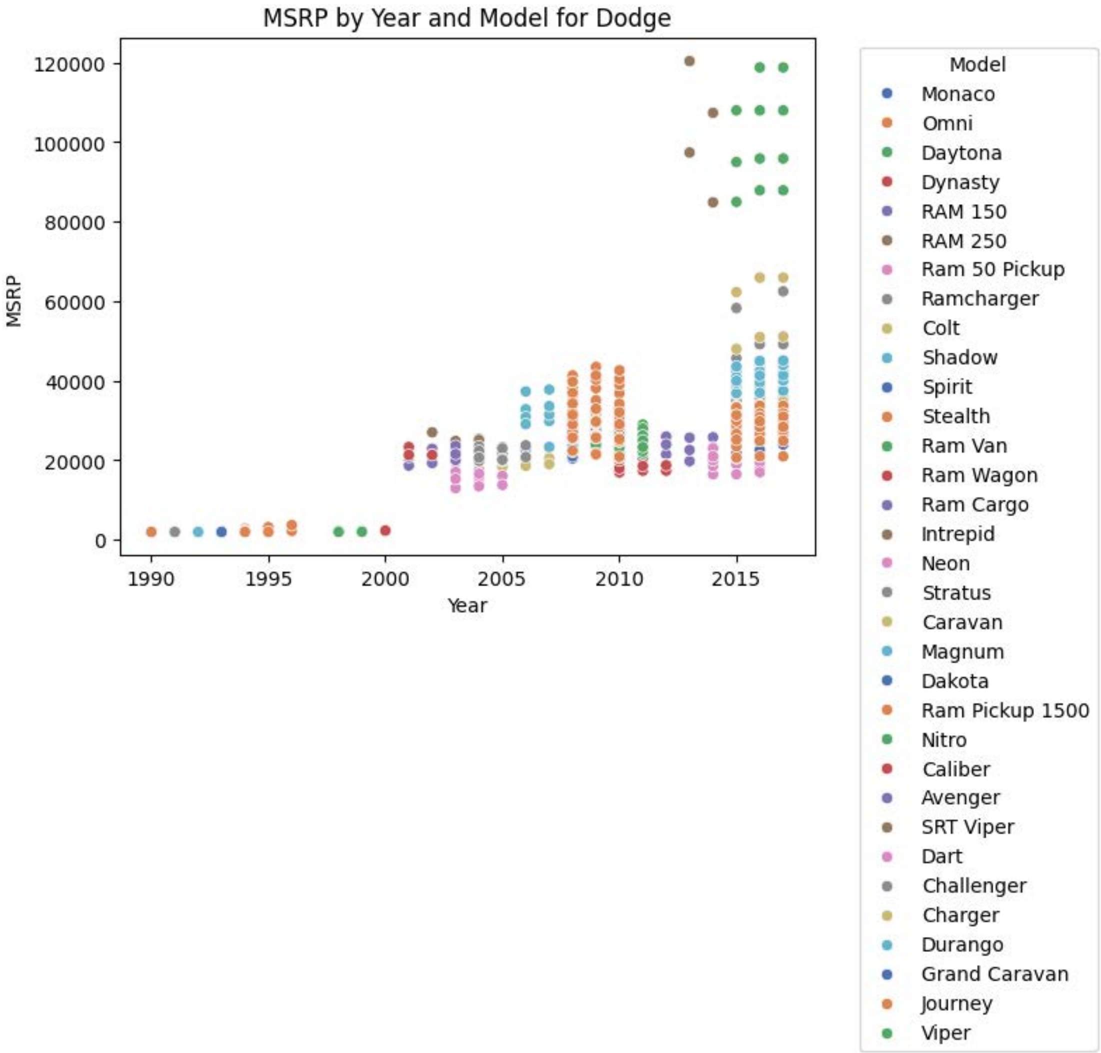
```
<ipython-input-11-08be73958b0b>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
filtered_df.sort_values(by=[‘Make’, ‘Year’, ‘Model’], inplace=True)
```

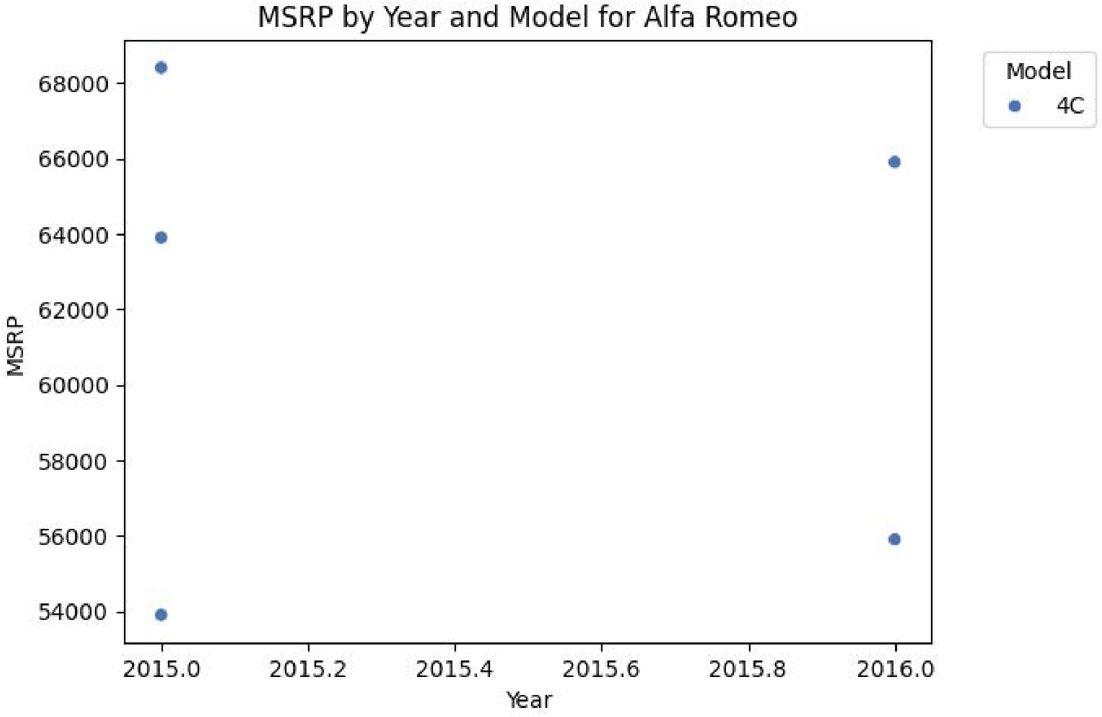

MSRP by Year and Model for Chevrolet
MSRP
80000
60000
40000
20000
0
1990
1995
2000
2005
2010
2015
Year
Model
Celebrity
S-10 Blazer
Beretta
Caprice
Corsica
Lumina Minivan
Sportvan
Chevy Van
C/K 1500 Series
C/K 2500 Series
Lumina
Metro
Prizm
Tahoe Limited/Z71
S-10
Tracker
Astro
Astro Cargo
Blazer
Cavalier
Venture
Classic
SSR
TrailBlazer EXT
Malibu Maxx
Monte Carlo
Uplander
Silverado 1500 Classic
TrailBlazer
Cobalt
Malibu Classic
Malibu Hybrid
Aveo
HHR
Avalanche
Silverado 1500 Hybrid
Tahoe Hybrid
Colorado
Black Diamond Avalanche
Captiva Sport
Corvette Stingray
Express
Express Cargo
Impala Limited
SS
Spark EV
Camaro
City Express
Corvette
Cruze
Equinox
Impala
Malibu
Silverado 1500
Sonic
Spark
Suburban
Tahoe
Traverse
Trax
Cruze Limited
Malibu Limited
Bolt EV

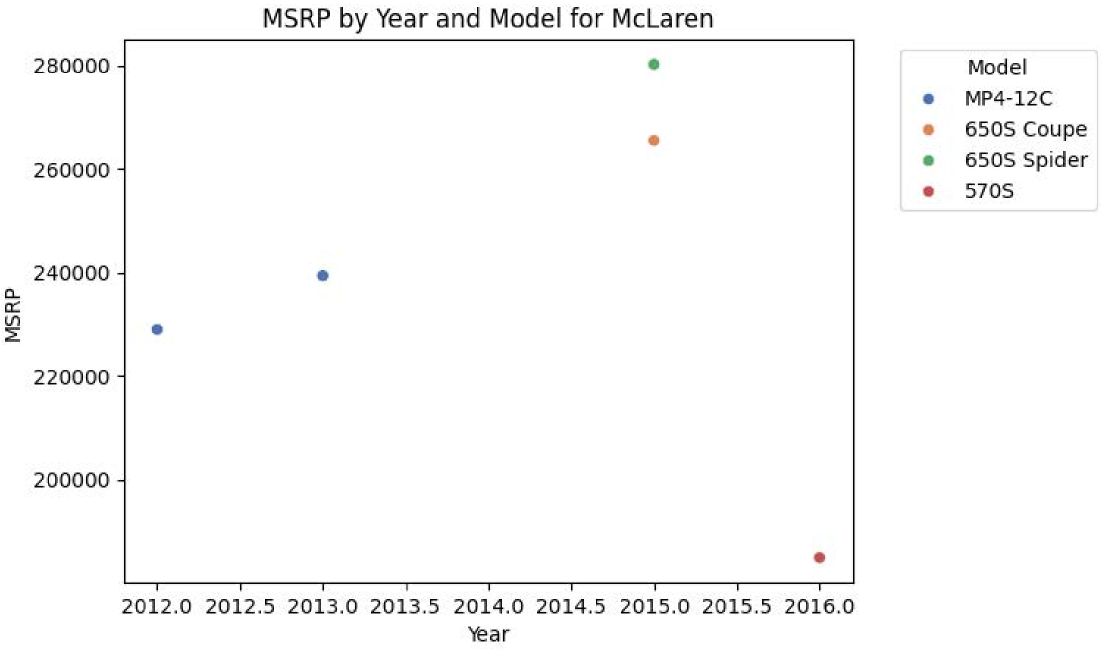

MSRP by Year and Model for Ford
MSRP
0
20000
40000
60000
80000
100000
120000
140000
Year
1990
1995
2000
2005
2010
2015
Model
Bronco II
LTD Crown Victoria
Festiva
Tempo
Bronco
Aerostar
Aspire
F-150 SVT Lightning
Probe
E-150
E-250
F-250
Mustang SVT Cobra
Contour
Contour SVT
Escort
Explorer Sport
Windstar
Windstar Cargo
Thunderbird
F-150 Heritage
Five Hundred
Freestar
Freestyle
GT
Explorer Sport Trac
Taurus X
Crown Victoria
Escape Hybrid
Ranger
E-Series Van
E-Series Wagon
Shelby GT500
C-Max Hybrid
Edge
Escape
Explorer
F-150
Fiesta
Flex
Focus
Focus ST
Fusion
Fusion Hybrid
Mustang
Shelby GT350
Taurus
Transit Connect
Transit Wagon
Expedition
Focus RS

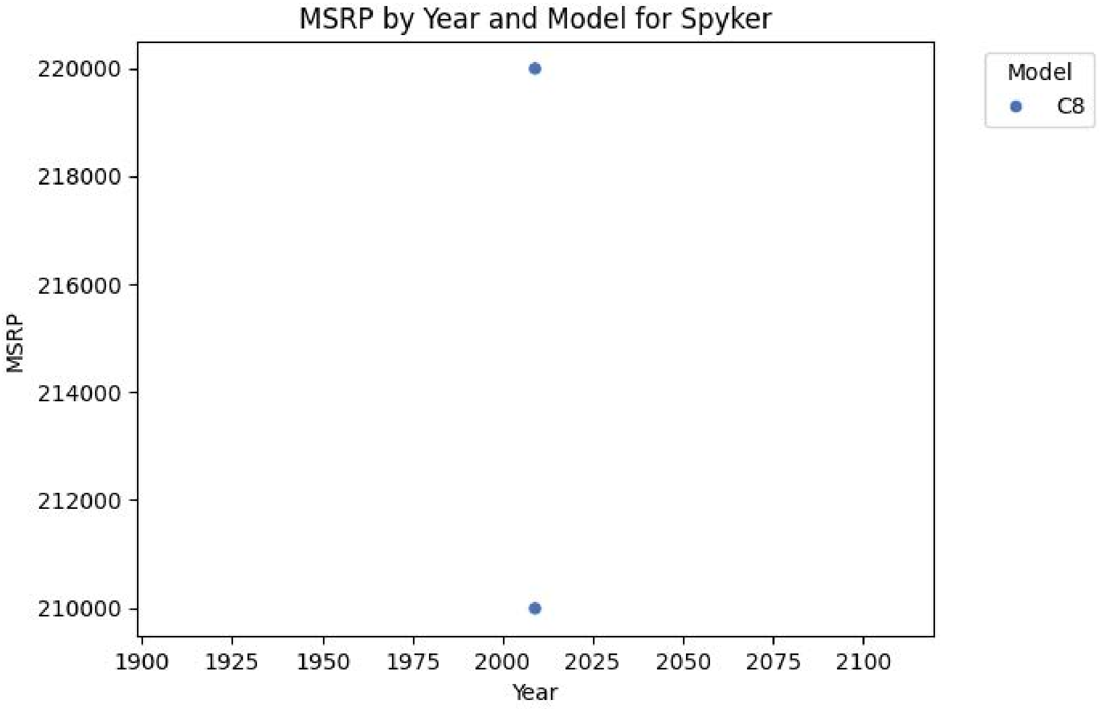

MSRP by Year and Model for Volkswagen
MSRP
0
20000
40000
60000
80000
100000
Year
1990
1995
2000
2005
2010
2015
Model
Vanagon
Cabriolet
Fox
Corrado
Cabrio
EuroVan
Phaeton
R32
Rabbit
GLI
New Beetle
Touareg 2
Routan
GTI
Jetta SportWagen
Jetta GLI
Jetta Hybrid
Beetle Convertible
Eos
Beetle
CC
Golf
Golf GTI
Golf R
Golf SportWagen
Jetta
Passat
Tiguan
Touareg
e-Golf
Golf Alltrack

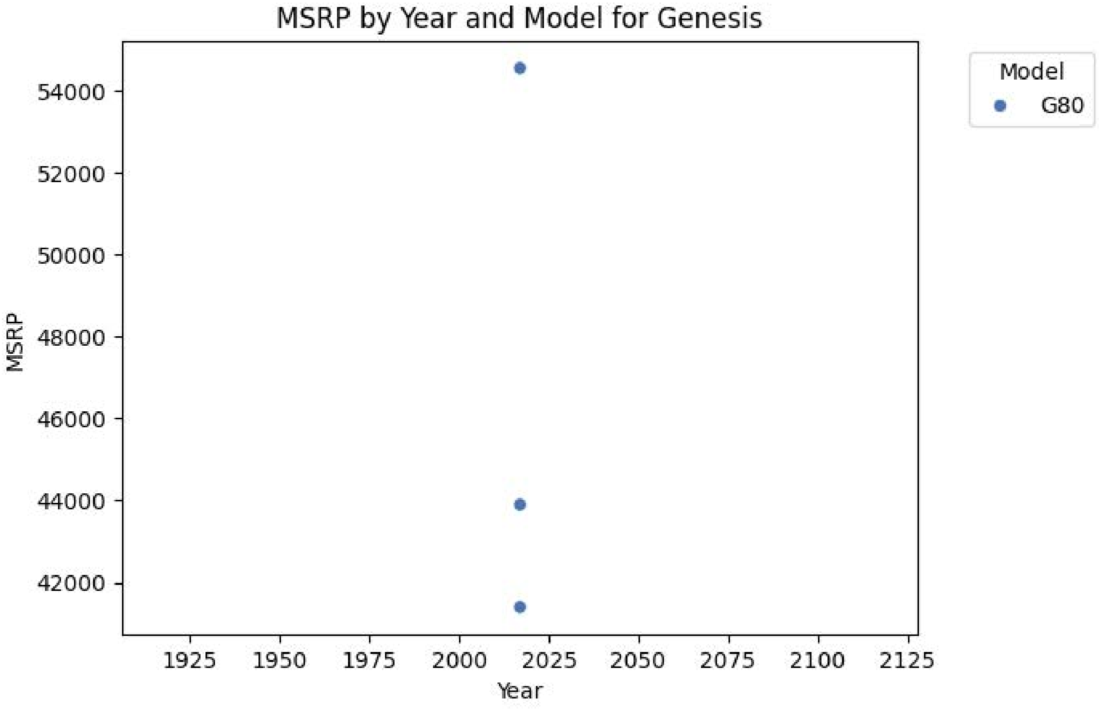

MSRP by Year and Model for Toyota
MSRP
0
20000
40000
60000
80000
1990
1995
2000
2005
2010
2015
Year
Model
Cressida
MR2
Pickup
Paseo
Previa
Supra
T100
Tercel
Celica
ECHO
MR2 Spyder
Camry Solara
Matrix
FJ Cruiser
RAV4 EV
Venza
4Runner
Highlander Hybrid
Prius c
Avalon
Avalon Hybrid
Camry
Camry Hybrid
Corolla
Highlander
Land Cruiser
Prius
Prius v
RAV4
Sequoia
Sienna
Tacoma
Tundra
Yaris
RAV4 Hybrid
86
Corolla iM
Prius Prime
Yaris iA

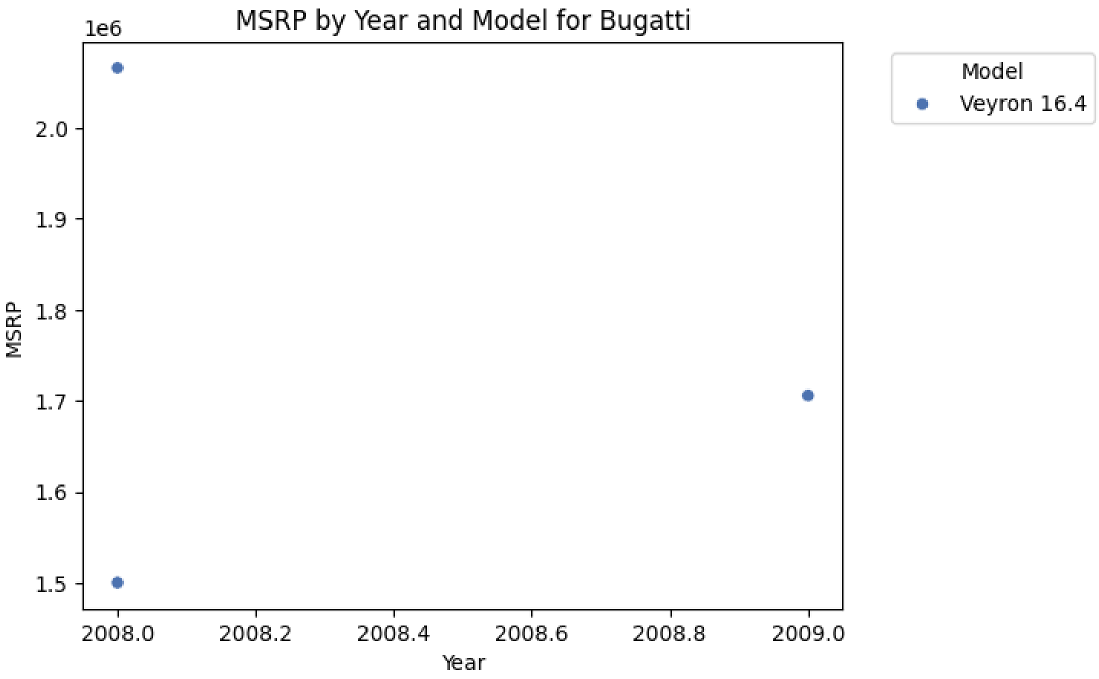

MSRP by Year and Model for Dodge
MSRP
Year
0
20000
40000
60000
80000
100000
120000
1990
1995
2000
2005
2010
2015
Model
Monaco
Omni
Daytona
Dynasty
RAM 150
RAM 250
Ram 50 Pickup
Ramcharger
Colt
Shadow
Spirit
Stealth
Ram Van
Ram Wagon
Ram Cargo
Intrepid
Neon
Stratus
Caravan
Magnum
Dakota
Ram Pickup 1500
Nitro
Caliber
Avenger
SRT Viper
Dart
Challenger
Charger
Durango
Grand Caravan
Journey
Viper


MSRP by Year and Model for Alfa Romeo
Model
● 4C
MSRP
68000
66000
64000
62000
60000
58000
56000
54000
Year
2015.0
2015.2
2015.4
2015.6
2015.8
2016.0


MSRP by Year and Model for McLaren
Model
● MP4-12C
● 650S Coupe
● 650S Spider
● 570S
MSRP
280000
260000
240000
220000
200000
Year
2012.0
2012.5
2013.0
2013.5
2014.0
2014.5
2015.0
2015.5
2016.0


MSRP by Year and Model for Spyker
MSRP
220000
218000
216000
214000
212000
210000
Year
1900
1925
1950
1975
2000
2025
2050
2075
2100
Model
● C8


MSRP by Year and Model for Genesis
MSRP
54000
52000
50000
48000
46000
44000
42000
Year
1925
1950
1975
2000
2025
2050
2075
2100
2125
Model
● G80



These charts reveal a number of important insights:

1. Because of their smaller market share, smaller automakers might concentrate on premium or niche market models. This explains why their prices are higher for the limited models they offer.

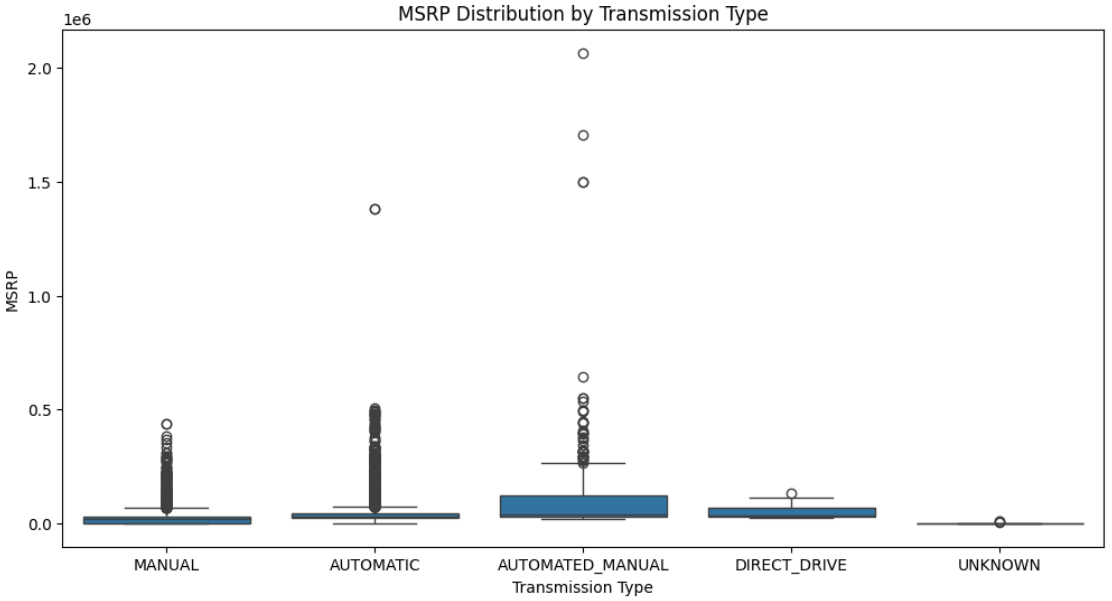
2. Well-known automakers have a wide variety of models to choose from, suggesting that they serve a range of market niches from affordable to high-end cars. This implies that they can provide models in a variety of pricing points to cater to the demands of various customer segments.

3. Since 2000, big automakers haven't stopped coming up with new ideas, perhaps in reaction to the industry's expanding demand and consumers' evolving tastes. The additional value of these models is reflected in the introduction of new designs and technology as well as the rise in MSRP.

4. The trend toward price concentration in cars suggests that even major automakers have a core range of prices. The preferences and purchasing power of the majority of consumers may be represented by this range.

5. Smaller automakers may use a higher price point to offset lower sales volumes, or it can be a result of their models' distinct benefits in terms of performance, brand value, or distinctiveness.

```
####Transmission Type vs. MSRP
plt.figure(figsize=(12, 6))
sns.boxplot(x='Transmission Type', y='MSRP', data=df)
plt.title('Transmission Type vs. MSRP')
plt.xlabel('Transmission Type')
plt.ylabel('MSRP')
plt.show()
```



The following boxplot observations can be made when examining the link between transmission types and the suggested retail price (MSRP):

Transmission Type: Manual (MANUAL):

- The lowest median price suggests that cars with manual gearboxes are often less expensive.
- A narrow interquartile range indicates that most car costs vary slightly from the median.
- The existence of a few expensive outliers suggests that a small number of cars are priced far higher than average.

Transmission that is automatic (AUTOMATIC):

- The median price of vehicles with automatic transmissions is higher than that of vehicles with manual gearboxes, suggesting that automatic vehicles are generally more expensive.
- A broad price distribution with notable variations in car pricing is indicated by a wide interquartile range.
- Numerous expensive outliers point to a greater frequency of luxury automatic cars.

Transmission that is Automated Manual (AUTOMATED_MANUAL):

- The median MSRP is less variable and has a larger price concentration than automatic transmissions.

Drive Direct (DIRECT_DRIVE):

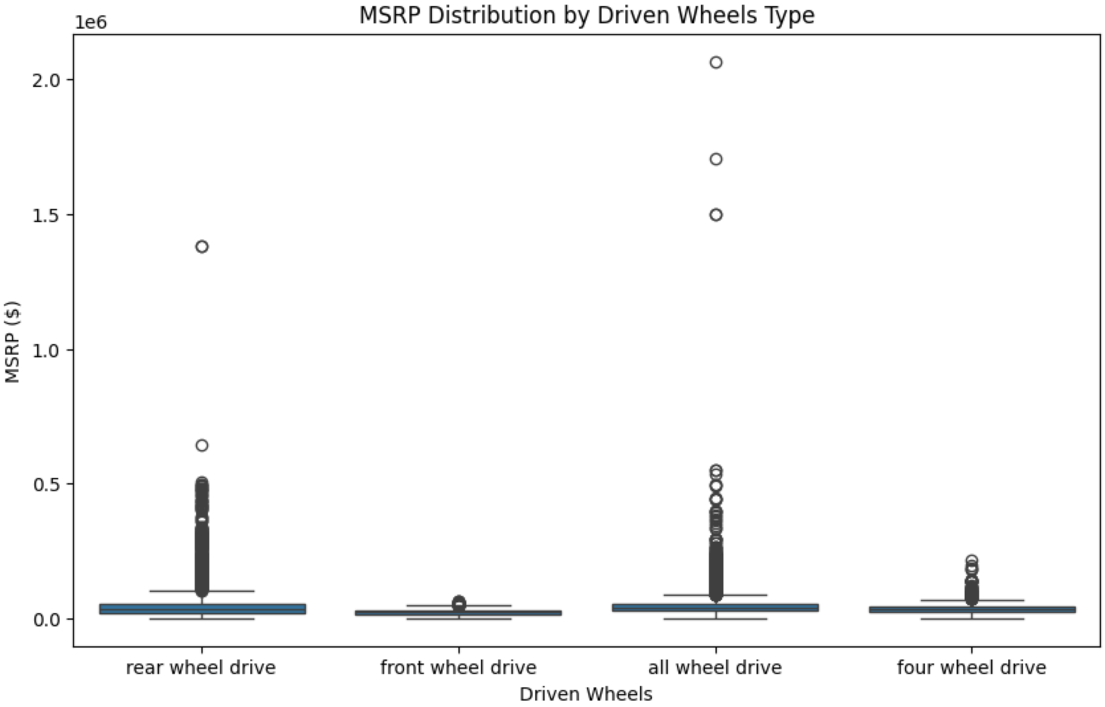
- This category, which has a lower MSRP median, is primarily found in electric automobiles.
- There is almost no interquartile range, which suggests that the costs of these kinds of cars are very fixed.
- There are very few outliers, indicating price stability.

Unknown (UNKNOWN):

- A low median price with a higher number of outliers, which could be the result of missing data or unique conditions. can be ignored.

####Driven Wheels Type vs. MSRP

```
plt.figure(figsize=(10, 6))
sns.boxplot(x=’Driven_Wheels’, y=’MSRP’, data=df)
plt.title(‘Driven Wheels Type vs. MSRP’)
plt.ylabel(‘MSRP ($)’)
plt.xlabel(‘Driven Wheels’)
plt.show()
```



#####The Manufacturer Suggested Retail Price (MSRP) distribution for various drivetrain types is shown in this boxplot.

• Rear-wheel Drive (RWD):

– There is a wide variety in the MSRP values.

– A number of anomalies suggest that the cost of certain RWD cars is noticeably more than that of others.

– RWD is frequently associated with luxury and sports vehicles, which could account for the high outliers.

• Front-Wheel Drive (FWD):

– The majority of FWD cars fall into the lower MSRP bracket, indicating that they are typically more affordable or standard.

– The distribution is less erratic and more compact, suggesting no change in price.

• All-Wheel Drive (AWD):

– Indicates a median price that is higher than that of FWD.

– A larger range of prices indicates more pricing fluctuation.

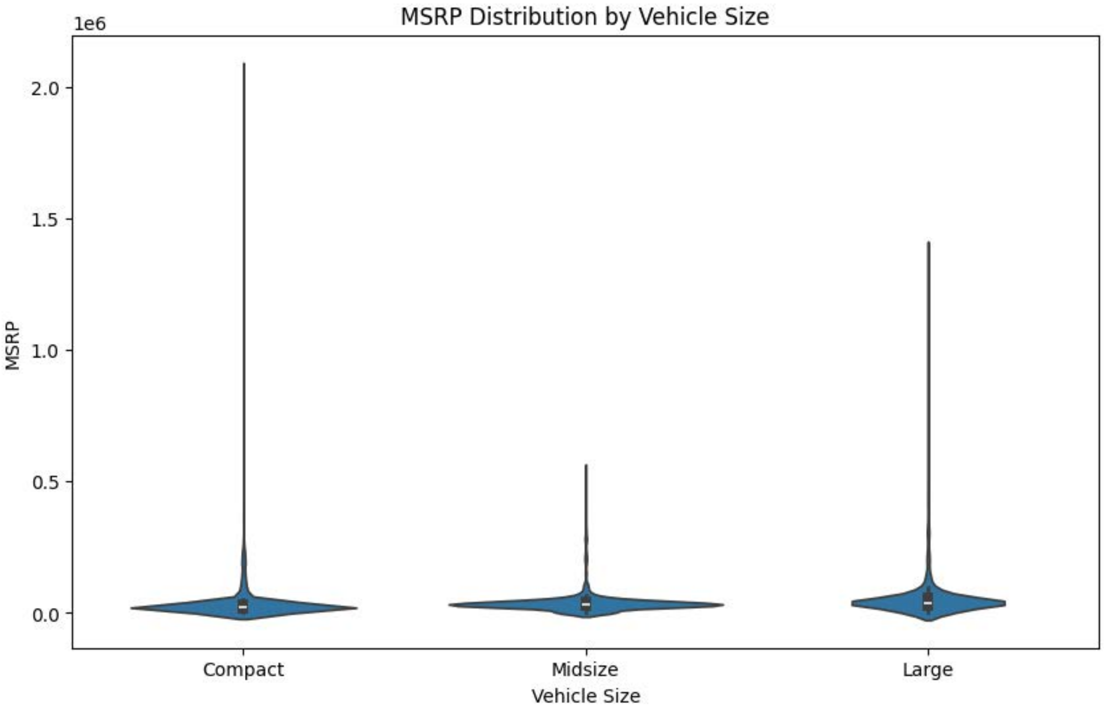
• Four-Wheel Drive (4WD):

– 4WD has a lower price range than AWD but a median price that is comparable to FWD. -This points to a more targeted pricing range for 4WD cars.

In conclusion, the MSRPs of RWD and AWD/4WD vehicles are generally greater than those of FWD vehicles, which are generally less expensive. This indicates that a vehicle's pricing element is its drivetrain type.

####Vehicle Size vs. MSRP

```
plt.figure(figsize=(10, 6))
sns.violinplot(x='Vehicle Size', y='MSRP', data=df)
plt.title('Vehicle Size vs. MSRP')
plt.show()
size_stats = df.groupby('Vehicle Size')['MSRP'].describe()
print(size_stats)
```



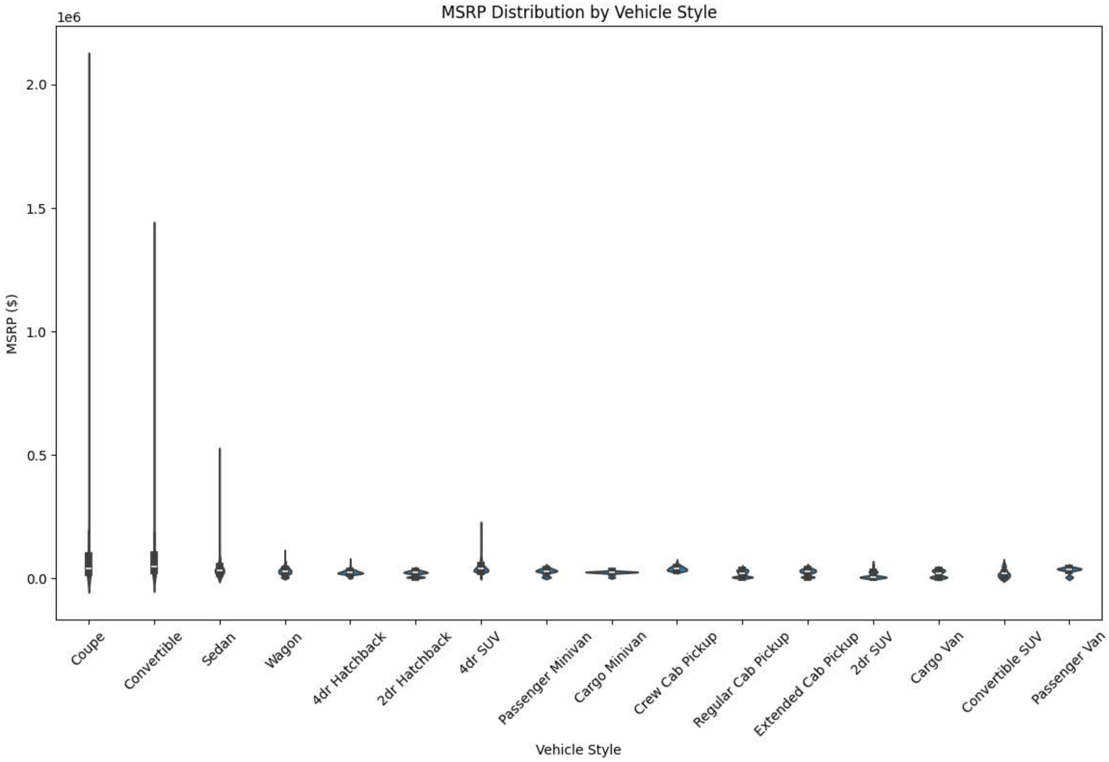
```
count         mean           std     min       25%      50%  \
Vehicle Size
Compact       4764.0  34275.336482  66110.113087  2000.0  16378.75  23580.0
Large        2777.0  53890.500540  70145.662905  2000.0  29730.00  39380.0
Midsize       4373.0  39035.919049  42441.340136  2000.0  24800.00  32785.0
75%        max
Vehicle Size
Compact       31292.5  2065902.0
Large        56235.0  1382750.0
Midsize       42195.0   548800.0
```

We may see the following from the data distribution displayed in this violin plot:

- Compact: The price distribution of compact automobiles is concentrated, with most models priced below the average. However, a few models have extremely high prices, which probably correspond to high-performance or luxury models.
- Midsize: With a wider price range, the price distribution of midsize automobiles is slightly more spread than that of compact cars. This suggests that there are a range of models of midsize cars, from affordable to luxurious.
- Large: The price distribution of large cars exhibits a more noticeable dispersion, with a few high-priced outliers. This implies that a wide range of car models, from high-end luxury cars to reasonably priced versions, are available on the market.
- Price Range: Midsize cars sit in between large and small cars, with the former having the narrowest price range.
- Outliers: The most expensive outliers across all car sizes are huge cars; this is probably because the majority of luxury car brands and models fall into this category, with costs that are much higher than the average level of the market.

Overall, there is a relationship between a vehicle's price and size, but other elements including performance, brand, and model configuration also have an impact on the final price. Because they cover a wide range of models, large and midsize automobiles have a somewhat broad price distribution, but compact cars have a more narrow price distribution because of their more targeted market positioning.

```
####Vehicle Style vs. MSRP
plt.figure(figsize=(14, 8))
sns.violinplot(x='Vehicle Style', y='MSRP', data=df)
plt.xticks(rotation=45)
plt.title('Vehicle Style vs. MSRP')
plt.xlabel('Vehicle Style')
plt.ylabel('MSRP ($)')
Text(0, 0.5, 'MSRP ($)')
```

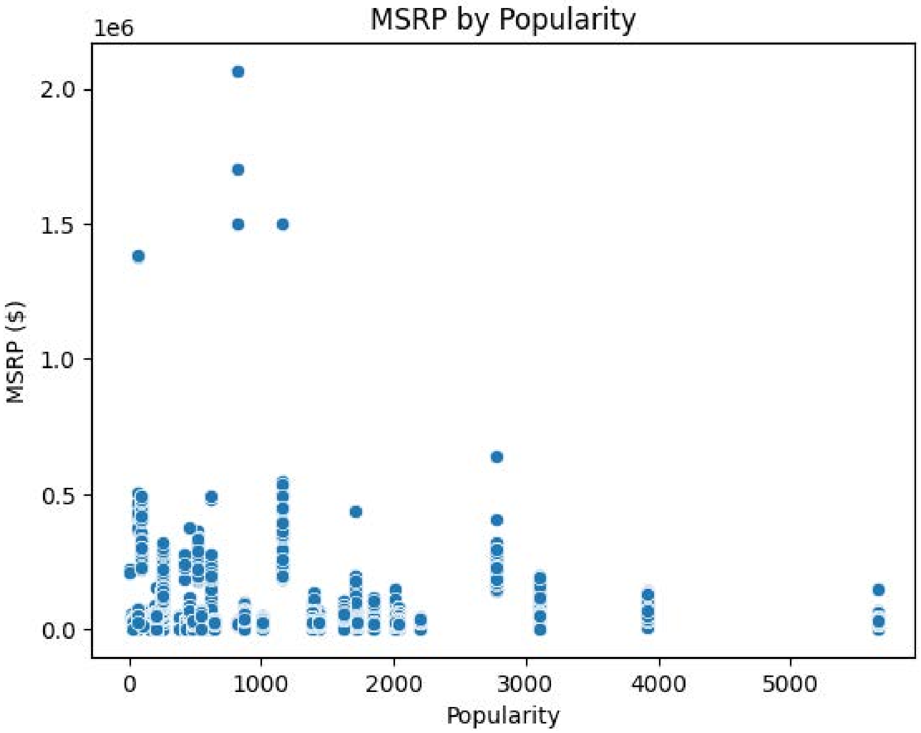


When examine the association between car style and MSRP, it could be found that, in comparison to other car styles, coupes and convertibles have a higher median and MSRP range. This illustrates the fact that these cars are frequently more costly, maybe as a result of their brand positioning, performance, or degree of luxury. Wagons and sedans represent a wide range of market sectors, from low-cost to high-end, which may account for their more concentrated MSRP distribution. Although they are centered in a lower price bracket, hatchbacks and SUVs also have a wide MSRP distribution. This is probably because these vehicles are more practical, offer more space, and cater to a wider client base. Minivans are positioned as family vehicles, emphasizing comfort and room above performance, which may account for their comparatively cheaper MSRPs. The lower MSRPs of passenger and cargo vans are indicative of their practical and business-oriented nature.

In general, a car’s type greatly affects its MSRP; family and utility vehicles are typically more reasonably priced, whereas coupes and convertibles are typically more costly.

####Popularity vs. MSRP

```
sns.scatterplot(x=’Popularity’, y=’MSRP’, data=df)
plt.title(‘Popularity vs. MSRP’)
plt.xlabel(‘Popularity’)
plt.ylabel(‘MSRP ($)’)
plt.show()
```



##Conclusions It can rank the ways in which each component influences the Manufacturer Suggested Retail Price (MSRP) when all of the analysis is complete. According to the research, brand-specific models, horsepower, and engine type have the most effects on price, while popularity and number of doors have less of an impact. Though these factors may be paired with other factors, the vehicle's style and type of gearbox also have an impact on price.

Horsepower (Engine HP): □□□□□
Cars with higher horsepower typically offer greater performance and come with higher prices.

Engine Type (Engine Cylinders): □□□□
Vehicles with more cylinders are often associated with sports cars.

Brand and Model (Make and Model): □□□□□
Some brands and models are priced higher due to their technology and brand value.

Popularity (Popularity): □□
Popularity may reflect market acceptance but is not always correlated with a higher MSRP.

Number of Doors (Number of Doors): □
The number of doors has a relatively minor impact on price.

Driven Wheels (Driven Wheels): □□□
The drivetrain can have a certain influence on the vehicle's performance and price.

Vehicle Style (Vehicle Style): □□□
Different vehicle styles cater to various market demands. SUVs, for example, are generally more expensive than compact cars, but the specific impact is also determined by other factors and performance.

Fuel Efficiency (MPG): □□
High fuel efficiency may reduce operating costs but has a minimal effect on price.

Market Category (Market Category): □□□□
Market categories, such as luxury and high-performance, significantly affect the price.

Vehicle Size (Vehicle Size): □□□
Larger vehicles tend to have higher prices.

Transmission Type (Transmission Type): □□□
Advanced transmission systems, like automatics, may increase the MSRP.

Year (Year): □□□□
Newer models often command higher prices due to the incorporation of the latest technology.

References

Rana, R. S. (2024, February 4). *Car features and prices dataset*. Kaggle. https://www.kaggle.com/datasets/rupindersinghrana/car-features-and-prices-dataset