In [45]:

```
# 1. data loading
# 2. data cleaning
# 3. data manipulation
# 4. outlier handling & removal
# 5. data visualization
```

In [46]:

```
# Modules for data preprocessing & statistical analysis
import numpy as np
import pandas as pd

# Modules for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

In [47]:

```
# Reading the data from .csv file using pandas.read_csv() function.
car_data = pd.read_csv("/kaggle/input/cardataset/data.csv")

# Getting the top 5 rows from dataset
car_data.head()
```

Out[47]:

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	hig
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	

In [48]:

```
# Getting the bottom 5 rows from dataset
car_data.tail()
```

Out[48]:

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size
11909	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize
11910	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize
11911	Acura	ZDX	2012	premium unleaded (required)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize
11912	Acura	ZDX	2013	premium unleaded (recommended)	300.0	6.0	AUTOMATIC	all wheel drive	4.0	Crossover,Hatchback,Luxury	Midsize
11913	Lincoln	Zephyr	2006	regular unleaded	221.0	6.0	AUTOMATIC	front wheel drive	4.0	Luxury	Midsize

In [49]:

```
car_data.describe()
```

Out[49]:

	Year	Engine HP	Engine Cylinders	Number of Doors	highway MPG	city mpg	Popularity	MSRP
count	11914.000000	11845.00000	11884.000000	11908.000000	11914.000000	11914.000000	11914.000000	1.191400e+04
mean	2010.384338	249.38607	5.628829	3.436093	26.637485	19.733255	1554.911197	4.059474e+04
std	7.579740	109.19187	1.780559	0.881315	8.863001	8.987798	1441.855347	6.010910e+04
min	1990.000000	55.00000	0.000000	2.000000	12.000000	7.000000	2.000000	2.000000e+03
25%	2007.000000	170.00000	4.000000	2.000000	22.000000	16.000000	549.000000	2.100000e+04
50%	2015.000000	227.00000	6.000000	4.000000	26.000000	18.000000	1385.000000	2.999500e+04
75%	2016.000000	300.00000	6.000000	4.000000	30.000000	22.000000	2009.000000	4.223125e+04
max	2017.000000	1001.00000	16.000000	4.000000	354.000000	137.000000	5657.000000	2.065902e+06

In [50]:

```
car_data.isnull()
```

Out[50]:

	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
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
11909	False	False	False	False	False	False	False	False	False	False	False	False	False	False
11910	False	False	False	False	False	False	False	False	False	False	False	False	False	False
11911	False	False	False	False	False	False	False	False	False	False	False	False	False	False
11912	False	False	False	False	False	False	False	False	False	False	False	False	False	False
11913	False	False	False	False	False	False	False	False	False	False	False	False	False	False

11914 rows × 16 columns

In [51]:

```
car_data.isnull().sum()
```

```
Out[51]: Make                    0
         Model                   0
         Year                    0
         Engine Fuel Type        3
         Engine HP              69
         Engine Cylinders       30
         Transmission Type       0
         Driven_Wheels           0
         Number of Doors         6
         Market Category      3742
         Vehicle Size            0
         Vehicle Style           0
         highway MPG             0
         city mpg                0
         Popularity              0
         MSRP                    0
         dtype: int64
```

```
In [52]: car_data.info()
```

```
<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
```

In [54]:
```
car_data.shape #(No. of rows, No. of columns)
```

Out[54]:
```
(11914, 16)
```

In [55]:
```
car_data = car_data.dropna()
```

In [56]:
```
car_data.shape
```

Out[56]:
```
(8084, 16)
```

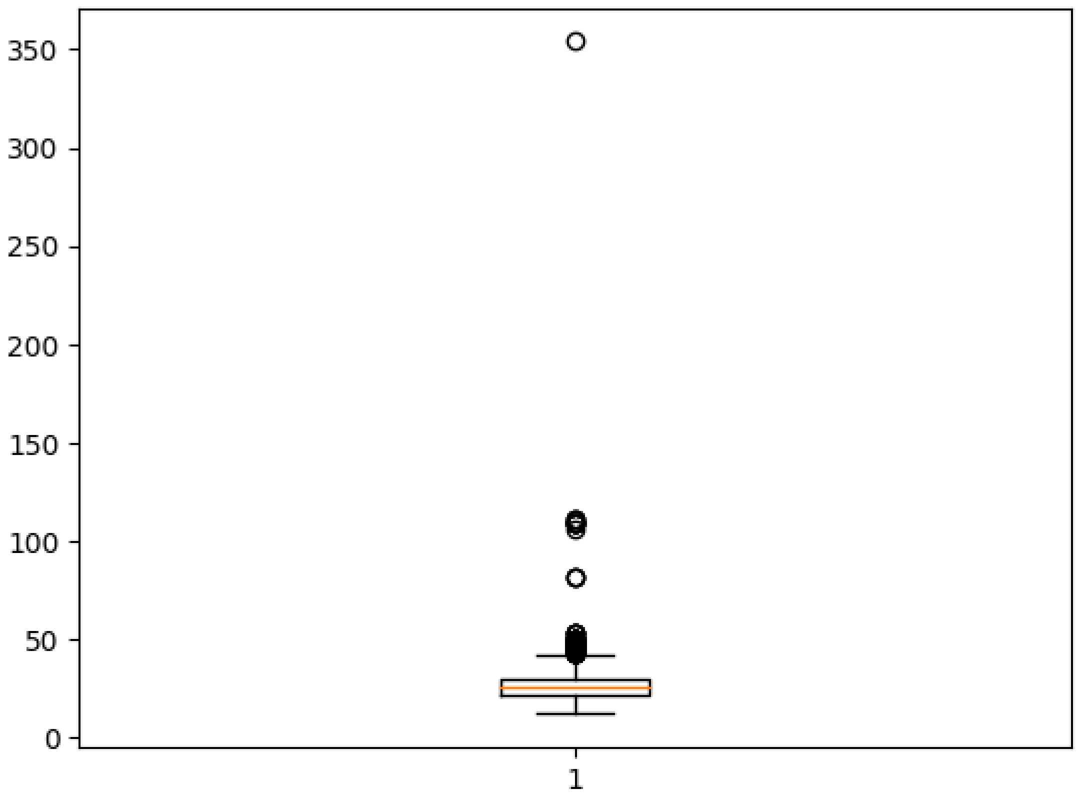
In [57]:
```
car_data.head()
```

Out[57]:

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	hig
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	

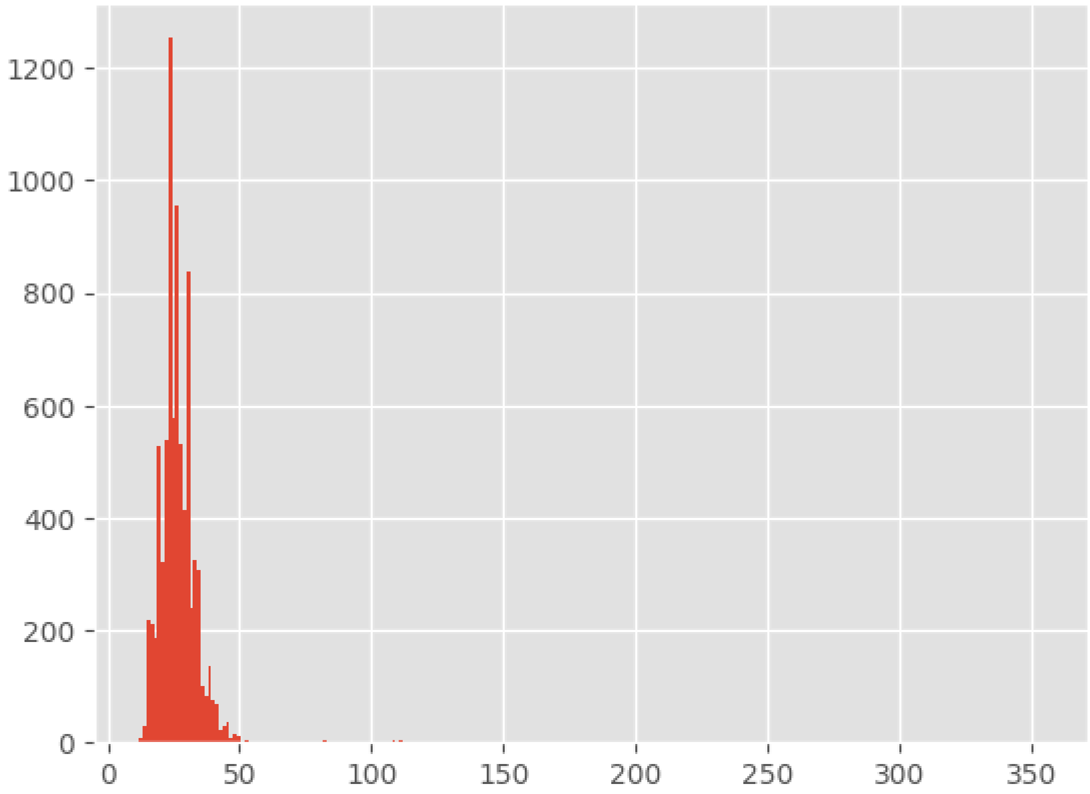
In [9]:

```
plt.boxplot(car_data["highway MPG"]);
```



In [69]:

```
plt.hist(highway_mpg_data, bins = 250);
```



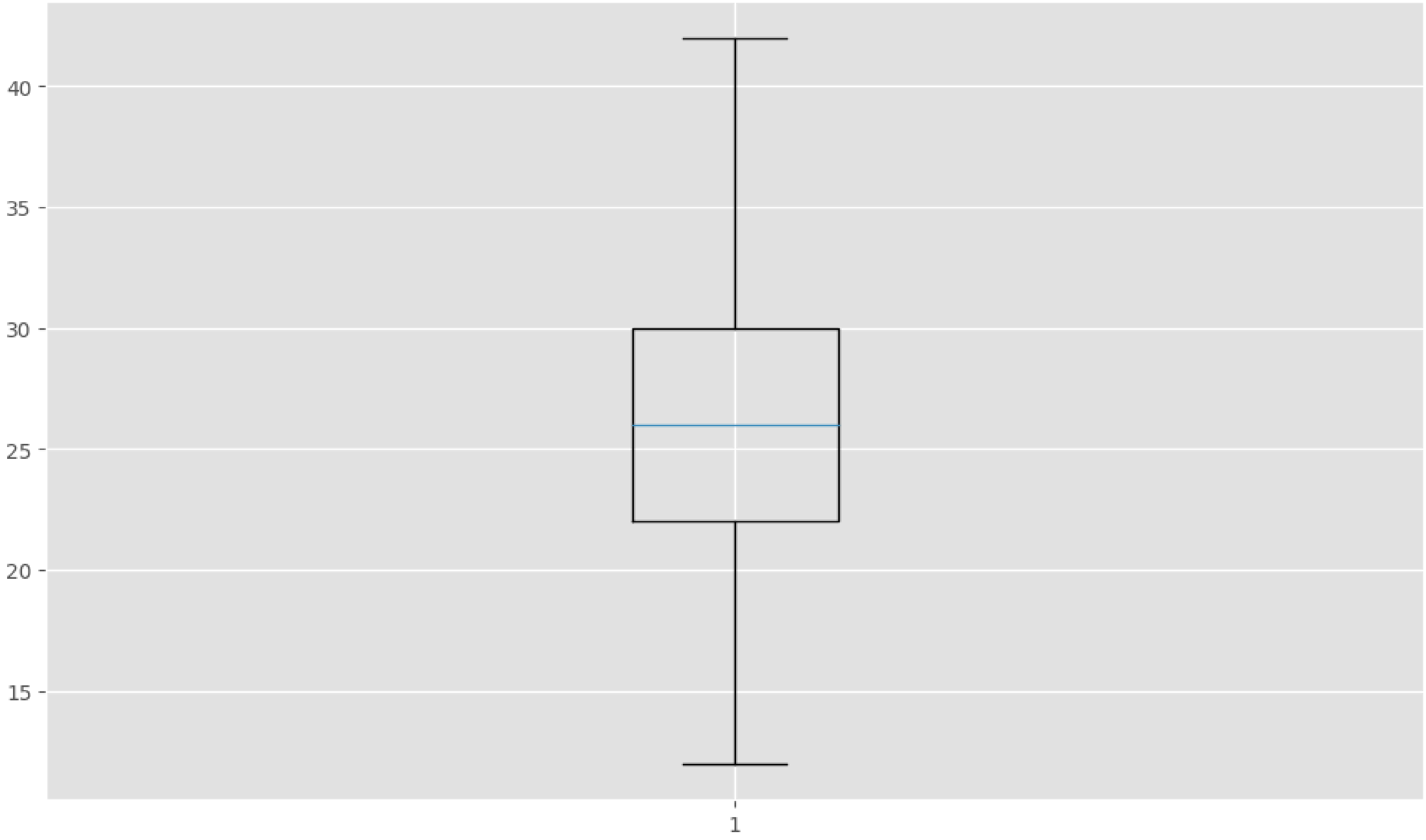
In [70]:

```
# finding the 1st quartile
q1 = np.quantile(car_data["highway MPG"], 0.25)

# finding the 3rd quartile
q3 = np.quantile(car_data["highway MPG"], 0.75)
med = np.median(car_data["highway MPG"])

# finding the iqr region
iqr = q3-q1

# finding upper and lower whiskers
upper_bound = q3+(1.5*iqr)
lower_bound = q1-(1.5*iqr)
print(iqr, upper_bound, lower_bound)
```

```
8.0 42.0 10.0
```

In [71]:

```
arr2 = car_data["highway MPG"][(car_data["highway MPG"] >= lower_bound) & (car_data["highway MPG"] <= upper_bound)]
plt.figure(figsize=(12, 7))
plt.boxplot(arr2)
plt.show()
```



In [73]:

```python
inlier_range = int(np.percentile(car_data["highway MPG"], 99.99))
inlier_range
```

Out[73]:

```
157
```

In [13]:

```python
car_data[car_data["highway MPG"] > inlier_range]
```

Out[13]:

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style
1119	Audi	A6	2017	premium unleaded (recommended)	252.0	4.0	AUTOMATED_MANUAL	front wheel drive	4.0	Luxury	Midsize	Sedan

In [14]:

```python
car_data.columns
```

Out[14]:

```
Index(['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', 'MSRP'],
      dtype='object')
```

In [15]:

```python
car_data = car_data.drop(car_data[car_data["highway MPG"] > inlier_range].index)
```

In [16]:

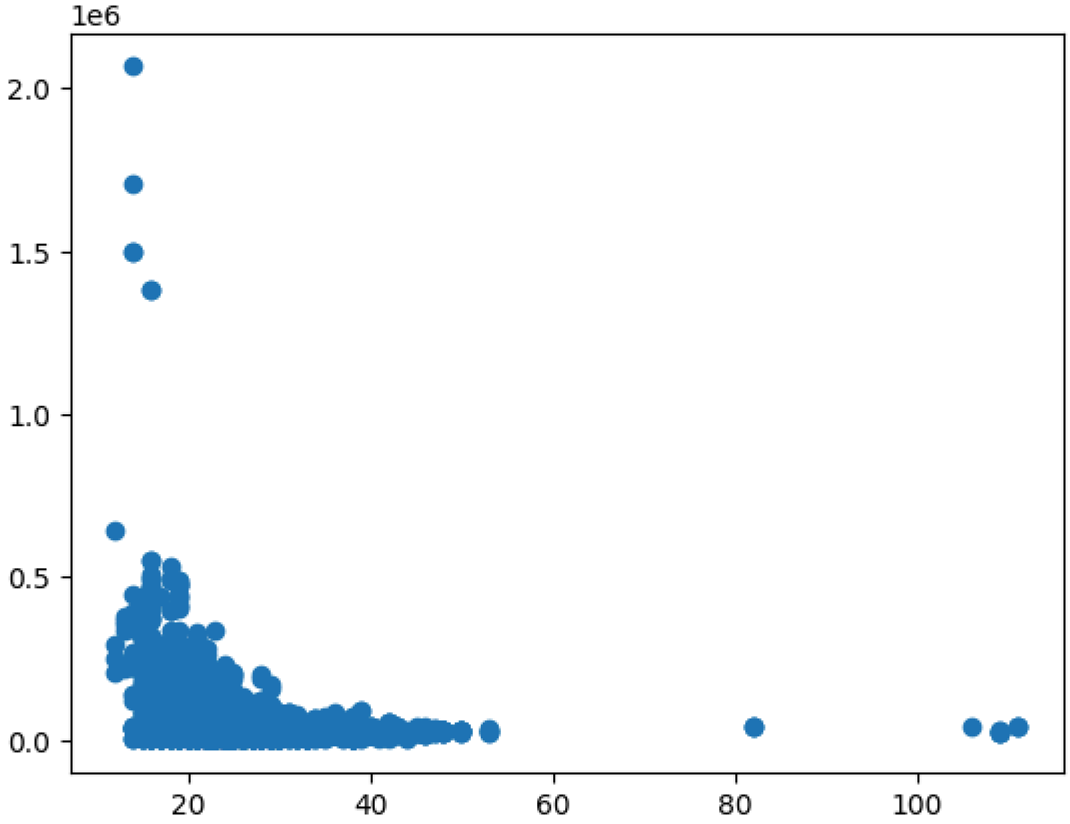
```python
inlier_range = int(np.percentile(car_data["highway MPG"], 99))
inlier_range
```

Out[16]:

```
45
```

In [17]:

```python
plt.scatter(car_data["highway MPG"], car_data["MSRP"]);
```



Hypothesis - 1

In here, we will try to identify relation between two variables Engine HP and MSRP

```
In [74]: car_data.head()
```

Out[74]:

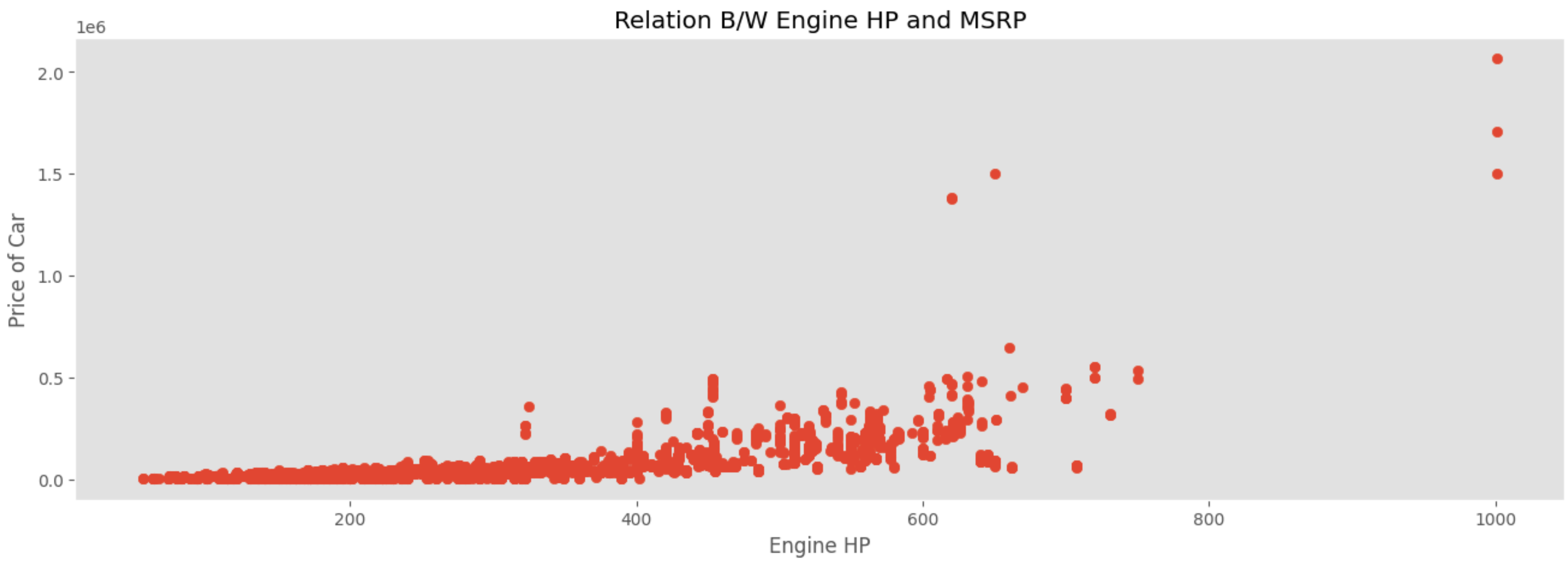
	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	hig
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	

In [88]:

```
plt.style.use("ggplot") # Styles of matplotlib
```

In [94]:

```
plt.figure(figsize = (16, 5)) # (horizontal size, vertical size)
plt.title("Relation B/W Engine HP and MSRP")
plt.grid(False)
plt.xlabel("Engine HP")
plt.ylabel("Price of Car")
plt.scatter(car_data["Engine HP"], car_data["MSRP"]);
```



Here, By looking at the graph we can conclude that the cars with high horsepower engines tends to have higher Price.

2. Hypothesis - 2

Lets analyze Car Make and popularity

In [100…

```
car_data.head()
```

Out[100…

	Make	Model	Year	Engine Fuel Type	Engine HP	Engine Cylinders	Transmission Type	Driven_Wheels	Number of Doors	Market Category	Vehicle Size	Vehicle Style	hig
0	BMW	1 Series M	2011	premium unleaded (required)	335.0	6.0	MANUAL	rear wheel drive	2.0	Factory Tuner,Luxury,High-Performance	Compact	Coupe	
1	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Convertible	
2	BMW	1 Series	2011	premium unleaded (required)	300.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,High-Performance	Compact	Coupe	
3	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury,Performance	Compact	Coupe	
4	BMW	1 Series	2011	premium unleaded (required)	230.0	6.0	MANUAL	rear wheel drive	2.0	Luxury	Compact	Convertible	

In [111…

```
popularity_sum = car_data.groupby(["Make"]).sum()["Popularity"]
model_count = car_data.groupby(["Make"]).count()["Model"]
```

In [114…

```
average_popularity = popularity_sum / model_count
average_popularity.head()
```

Out[114…

```
Make
Acura             204.0
Alfa Romeo        113.0
Aston Martin      259.0
Audi             3105.0
BMW              3916.0
dtype: float64
```

In [116…

```
sorted_popularity_data = average_popularity.sort_values(ascending = False)
sorted_popularity_data.head(3)
```

Out[116…

```
Make
Ford     5657.0
BMW      3916.0
Audi     3105.0
dtype: float64
```

In [121…

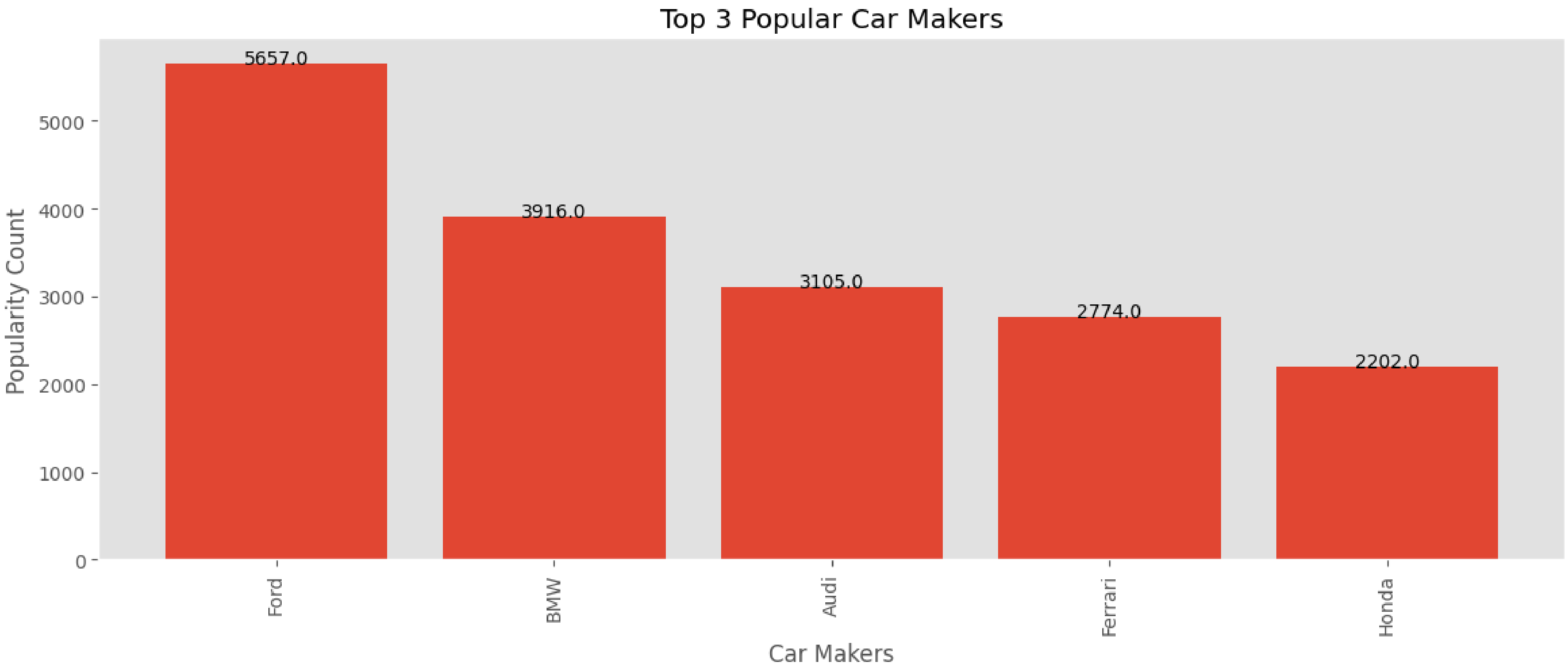
```
x_data = sorted_popularity_data.index
y_data = sorted_popularity_data.values
```

In [132…

```
top_x = 5
```

In [133…

```
plt.figure(figsize = (14, 5))
plt.style.use("ggplot")
plt.grid(False)

plt.title("Top 3 Popular Car Makers")
plt.xlabel("Car Makers")
plt.ylabel("Popularity Count")

plt.xticks(rotation=90)

bar_chart = plt.bar(x_data[:top_x], y_data[:top_x]);

for bar in bar_chart:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2.0, yval + 0.005, yval, ha="center")
```



Here, By looking at the graph we can identify the top X Most Populer Car Makers