## 1. IMPORTED THE DATASET AND THE NECESSARY LIBRARIES, CHECKED THE DATA TYPES/DIMENSIONS THE STATISTICAL SUMMARY AND SHAPE, EXCLUDED THE NULL VALUES

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv('/kaggle/input/car-price-prediction/CarPrice_Assignment.csv')
```

In [3]:
```
df.head(5)
```

Out[3]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 26 | 19 | 3916 | 46135 |
| 1 | BMW | 1 Series | 2011 | premium unleaded (required) | 300.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Convertible | 28 | 19 | 3916 | 40650 |
| 2 | BMW | 1 Series | 2011 | premium unleaded (required) | 300.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,High-Performance | Compact | Coupe | 28 | 20 | 3916 | 36350 |
| 3 | BMW | 1 Series | 2011 | premium unleaded (required) | 230.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Coupe | 28 | 18 | 3916 | 29450 |
| 4 | BMW | 1 Series | 2011 | premium unleaded (required) | 230.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Convertible | 28 | 18 | 3916 | 34500 |

In [4]:
```
df.tail(5)
```

Out[4]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11913 | Lincoln | Zephyr | 2016 | regular unleaded | 221.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury | Midsize | Sedan | 23 | 16 | 561 | 51795 |
| 11914 | Lexus | ES 350 | 2015 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11915 | Lexus | ES 350 | 2016 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11916 | Lexus | ES 350 | 2017 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11917 | Lincoln | Zephyr | 2016 | regular unleaded | 221.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury | Midsize | Sedan | 23 | 16 | 561 | 51795 |

In [5]:
```
df.info()
```

Out[5]:
```
<class'pandas.core.frame.DataFrame'>
RangeIndex: 11914 entries, 0 to 11913
Data columns (total 16 columns):
 Make               11914 non-null  object
 Model              11914 non-null  object
 Year               11914 non-null  int64
 Engine Fuel Type   11911 non-null  object
 Engine HP          11845 non-null  float64
 Engine Cylinders   11884 non-null  float64
 Transmission Type  11914 non-null  object
 Driven_Wheels      11914 non-null  object
 Number of Doors    11908 non-null  float64
 Market Category    8172 non-null   object
 Vehicle Size       11914 non-null  object
 Vehicle Style      11914 non-null  object
 Highway MPG        11914 non-null  int64
 City mpg           11914 non-null  int64
 Popularity         11914 non-null  int64
 MSRP               11914 non-null  int64
dtype: float64(4), int64(6), object(6)
memory usage: 1.5 MB
```

In [6]:
```
df.describe()
```

Out[6]:

| | 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 | 11914.000000 |
| mean | 2010.384338 | 249.386719 | 5.628829 | 3.436093 | 26.637485 | 19.733255 | 1554.911197 | 40594.737032 |
| std | 7.579740 | 109.191870 | 1.756203 | 0.881315 | 8.863001 | 8.987798 | 1441.855347 | 60732.54193 |
| min | 2004.000000 | 55.000000 | 0.000000 | 2.000000 | 12.000000 | 7.500000 | 2.000000 | 2000.000000 |
| 25% | 2007.000000 | 170.000000 | 4.000000 | 2.000000 | 22.000000 | 16.000000 | 549.000000 | 21877.000000 |
| 50% | 2010.000000 | 227.000000 | 6.000000 | 4.000000 | 26.000000 | 18.000000 | 1385.000000 | 29994.000000 |
| 75% | 2012.000000 | 304.000000 | 6.000000 | 4.000000 | 29.000000 | 22.000000 | 2009.000000 | 39690.000000 |
| max | 2017.000000 | 1020.000000 | 16.000000 | 4.000000 | 354.000000 | 137.000000 | 5657.000000 | 2065902.000000 |

In [7]:
```
df.shape
```

Out[7]:
```
(11914, 16)
```

In [8]:
```
df.columns
```

Out[8]:
```
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 [9]:
```
df.shape[0]
```

Out[9]:
```
11914
```

In [10]:
```
df.shape[1]
```

Out[10]:
```
16
```

In [11]:
```
df.isnull()
```

Out[11]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | 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 | False | False |
| 2 | False | False | 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 | False | False |
| 4 | False | False | 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 | False | False |
| 11914 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11915 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11916 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11917 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |

11914 rows × 16 columns

In [12]:
```
df.isnull().shape
```

Out[12]:
```
(11914, 16)
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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    4
Vehicle Size       0
Vehicle Style      0
Highway MPG        0
City mpg           0
Popularity         0
MSRP               0
dtype: int64
```

In [14]:
```
df.drop(['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'], axis=1)
```

Out[14]:

| | Make | Model | Year |
|---|---|---|---|
| 0 | BMW | 1 Series M | 2011 |
| 1 | BMW | 1 Series | 2011 |
| 2 | BMW | 1 Series | 2011 |
| 3 | BMW | 1 Series | 2011 |
| 4 | BMW | 1 Series | 2011 |
| ... | ... | ... | ... |
| 11913 | Lincoln | Zephyr | 2016 |
| 11914 | Lexus | ES 350 | 2015 |
| 11915 | Lexus | ES 350 | 2016 |
| 11916 | Lexus | ES 350 | 2017 |
| 11917 | Lincoln | Zephyr | 2016 |

11914 rows × 3 columns

## 2. WE HAVE SELECTED THE FOLLOWING COLUMNS AS THEY ARE INDEPENDANT OF THE MSRP (IN THE DEPENDANT)

In [15]:
```
df[['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']].corrwith(df['MSRP'])
```

Out[15]:
```
Engine HP          0.660823
Engine Cylinders   0.526988
Number of Doors    -0.139349
Highway MPG        -0.227390
City mpg           -0.224851
Popularity         0.304708
dtype: float64
```

In [16]:
```
df.corr()
```

Out[16]:

| | Year | Engine HP | Engine Cylinders | Number of Doors | Highway MPG | City mpg | Popularity | MSRP |
|---|---|---|---|---|---|---|---|---|
| Year | 1.000000 | 0.338351 | -0.142542 | -0.040755 | 0.258740 | 0.198171 | 0.226805 | 0.167241 |
| Engine HP | 0.338351 | 1.000000 | 0.774851 | -0.153190 | -0.415826 | -0.313346 | 0.061724 | 0.660823 |
| Engine Cylinders | -0.142542 | 0.774851 | 1.000000 | 0.042922 | -0.614213 | -0.486342 | 0.040908 | 0.526988 |
| Number of Doors | -0.040755 | -0.153190 | 0.042922 | 1.000000 | 0.582884 | 0.557192 | -0.085557 | -0.139349 |
| Highway MPG | 0.258740 | -0.415826 | -0.614213 | 0.582884 | 1.000000 | 0.885657 | -0.027659 | -0.227390 |
| City mpg | 0.198171 | -0.313346 | -0.486342 | 0.557192 | 0.885657 | 1.000000 | -0.004021 | -0.224851 |
| Popularity | 0.226805 | 0.061724 | 0.040908 | -0.085557 | -0.027659 | -0.004021 | 1.000000 | 0.304708 |
| MSRP | 0.167241 | 0.660823 | 0.526988 | -0.139349 | -0.227390 | -0.224851 | 0.304708 | 1.000000 |

In [17]:
```
df.shape
```

Out[17]:
```
(11914, 16)
```

In [18]:
```
df.shape[0]
```

Out[18]:
```
11914
```

In [19]:
```
df.shape[1]
```

Out[19]:
```
16
```

## 3. WE HAVE REMARKED THE COLUMNS AS BELOW

In [20]:
```
df[['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']]
```

Out[20]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 26 | 19 | 3916 | 46135 |
| 1 | BMW | 1 Series | 2011 | premium unleaded (required) | 300.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Convertible | 28 | 19 | 3916 | 40650 |
| 2 | BMW | 1 Series | 2011 | premium unleaded (required) | 300.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,High-Performance | Compact | Coupe | 28 | 20 | 3916 | 36350 |
| 3 | BMW | 1 Series | 2011 | premium unleaded (required) | 230.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Coupe | 28 | 18 | 3916 | 29450 |
| 4 | BMW | 1 Series | 2011 | premium unleaded (required) | 230.0 | 6.0 | MANUAL | rear/wheel drive | 2.0 | Luxury,Performance | Compact | Convertible | 28 | 18 | 3916 | 34500 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 11913 | Lincoln | Zephyr | 2016 | regular unleaded | 221.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury | Midsize | Sedan | 23 | 16 | 561 | 51795 |
| 11914 | Lexus | ES 350 | 2015 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11915 | Lexus | ES 350 | 2016 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11916 | Lexus | ES 350 | 2017 | premium unleaded (required) | 302.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury,Performance,Hybrid | Midsize | Sedan | 34 | 27 | 2009 | 54990 |
| 11917 | Lincoln | Zephyr | 2016 | regular unleaded | 221.0 | 6.0 | AUTOMATIC | all-wheel drive | 4.0 | Luxury | Midsize | Sedan | 23 | 16 | 561 | 51795 |

11914 rows × 16 columns

## 4. WE HAVE CHECKED FOR DUPLICATES, NULL VALUES AS WELL AS MISSING DATA AND EXCLUDED THEM FROM THE GIVEN DATASET

In [21]:
```
df.duplicated
```

Out[21]:
```
<bound method DataFrame.duplicated of       Make       Model  Year  ...  Popularity  MSRP
0      BMW  1 Series M  2011  ...        3916 ...
1      BMW   1 Series  2011  ...        3916 ...
2      BMW   1 Series  2011  ...        3916 ...
3      BMW   1 Series  2011  ...        3916 ...
4      BMW   1 Series  2011  ...        3916 ...
...    ...        ...   ...  ...         ...  ...
11913  ...        ...   ...  ...         ...  ...
11914  ...        ...   ...  ...         ...  ...
11915  ...        ...   ...  ...         ...  ...
11916  ...        ...   ...  ...         ...  ...
11917  ...        ...   ...  ...         ...  ...

[11914 rows x 16 columns]>
```

In [22]:
```
df.duplicated.shape
```

Out[22]:
```
(11914, 16)
```

### WE HAVE DEFINED ON DUPLICATE VALUES WHICH ARE UNIQUE AND DROPPED FURTHER

In [23]:
```
df.drop_duplicates
```

Out[23]:
```
<bound method DataFrame.drop_duplicates of       Make       Model  Year  ...  Popularity  MSRP
0      BMW  1 Series M  2011  ...        3916 ...
1      BMW   1 Series  2011  ...        3916 ...
2      BMW   1 Series  2011  ...        3916 ...
3      BMW   1 Series  2011  ...        3916 ...
4      BMW   1 Series  2011  ...        3916 ...
...    ...        ...   ...  ...         ...  ...
11913  ...        ...   ...  ...         ...  ...
11914  ...        ...   ...  ...         ...  ...
11915  ...        ...   ...  ...         ...  ...
11916  ...        ...   ...  ...         ...  ...
11917  ...        ...   ...  ...         ...  ...

[11914 rows x 16 columns]>
```

In [24]:
```
df.drop_duplicates.shape
```

Out[24]:
```
(11914, 16)
```

### WE HAVE REMOVED NULL VALUES AND AS A COLUMNS WHICH ARE DROPPED FURTHER

In [25]:
```
df.isnull()
```

Out[25]:

| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | 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 | False | False |
| 2 | False | False | 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 | False | False |
| 4 | False | False | 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 | False | False |
| 11914 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11915 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11916 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |
| 11917 | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False | False |

11914 rows × 16 columns

In [26]:
```
df.isnull.shape
```

Out[26]:
```
(11914, 16)
```

In [27]:
```
df.isnull.sum()
```

Out[27]:
```
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    4
Vehicle Size       0
Vehicle Style      0
Highway MPG        0
City mpg           0
Popularity         0
MSRP               0
dtype: int64
```

## 5. WE DECLARE THE MAIN GRAPH FOR THE TOP 10 REPRESENTED CARS IN THE DATA SET AND BE ABLE TO DISPLAY THE MAXIMUM REPRESENTED CAR IN THE BAR GRAPH, ALSO THE BAR GRAPH IN COLUMNS TO THE MAXIMUM REPRESENTED DATA POINTS FOR THE GIVEN DATASET

In [28]:
```
df.Make.value_counts().nlargest(10).plot(kind='bar', rot=45);
plt.ylabel('Number of Cars');
plt.title('TOP 10 CAR BRANDS');
```

Out[28]:

### HERE THE MOST REPRESENTED CAR BRAND IS CHEVROLET WITH ALMOST 15% OF THE CARS FROM THE COMPANY IN THE DATASET

In [29]:
```
df.Make.value_counts().nlargest(10).plot(kind='pie', autopct='%1.1f%%');
plt.ylabel('Percentage');
plt.title('TOP 10 CAR BRANDS');
```

Out[29]:

## 6. WE DECLARE WE HAVE PLOTTED THE CORRELATION MATRIX AND GIVEN OUR INSIGHTS FROM THE OBTAINED RESULTS

In [30]:
```
df[['Year', 'Engine HP', 'Engine Cylinders', 'Number of Doors', 'Highway MPG', 'City mpg', 'Popularity', 'MSRP']].corr()
```

Out[30]:

| | Year | Engine HP | Engine Cylinders | Number of Doors | Highway MPG | City mpg | Popularity | MSRP |
|---|---|---|---|---|---|---|---|---|
| Year | 1.000000 | 0.338351 | -0.142542 | -0.040755 | 0.258740 | 0.198171 | 0.226805 | 0.167241 |
| Engine HP | 0.338351 | 1.000000 | 0.774851 | -0.153190 | -0.415826 | -0.313346 | 0.061724 | 0.660823 |
| Engine Cylinders | -0.142542 | 0.774851 | 1.000000 | 0.042922 | -0.614213 | -0.486342 | 0.040908 | 0.526988 |
| Number of Doors | -0.040755 | -0.153190 | 0.042922 | 1.000000 | 0.582884 | 0.557192 | -0.085557 | -0.139349 |
| Highway MPG | 0.258740 | -0.415826 | -0.614213 | 0.582884 | 1.000000 | 0.885657 | -0.027659 | -0.227390 |
| City mpg | 0.198171 | -0.313346 | -0.486342 | 0.557192 | 0.885657 | 1.000000 | -0.004021 | -0.224851 |
| Popularity | 0.226805 | 0.061724 | 0.040908 | -0.085557 | -0.027659 | -0.004021 | 1.000000 | 0.304708 |
| MSRP | 0.167241 | 0.660823 | 0.526988 | -0.139349 | -0.227390 | -0.224851 | 0.304708 | 1.000000 |

## 7. WE HAVE PERFORMED EDA AND PLOTTED DIFFERENT GRAPHS AND DOCUMENTED OUT INSIGHTS

In [31]:
```
df.groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('Average price per year');
```

Out[31]:

In [32]:
```
df[df.Year>2000].groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('BoxPlot Of Price Per Year For Cars Sold Before Year 2000');
```

Out[32]:

In [33]:
```
df[df.Year<2000].groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('BoxPlot Of Price per Year Filter');
```

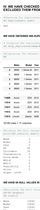
Out[33]:

In [34]:
```
df[df.Year>2000].groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('BoxPlot Of Price per Year');
```

Out[34]:

In [35]:
```
df[df.Year<2000].groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('BoxPlot Of Price per Year Filtered');
```

Out[35]:




In [36]:
```
df.Year.hist(bins=20);
plt.ylabel('Count Year');
plt.title('CAR DISTRIBUTION OVER YEARS');
```

Out[36]:

### HERE THE GRAPH WE CAN CONCLUDE THAT THE NUMBER OF CARS SOLD AFTER 2010 SHOWS TREMENDOUS INCREASE

In [37]:
```
df[df.Year.isin(df[df.Make.isin(['Chevrolet', 'Ford', 'Toyota', 'Volkswagen', 'Nissan', 'Dodge', 'GMC', 'Honda', 'Mazda', 'Cadillac'])].Year.value_counts().nlargest(10).index)].groupby('Year').mean()['MSRP'].plot(linestyle='-');
plt.ylabel('Average Price');
plt.title('Price over 10 most represented car brands');
```

Out[37]:

In [38]:
```
sns.heatmap(df[['Year', 'Engine HP', 'Engine Cylinders', 'Number of Doors', 'Highway MPG', 'City mpg', 'Popularity', 'MSRP']].corr(), annot=True, fmt='.1f', cmap='RdBu');
plt.title('CORRELATION MATRIX');
```

Out[38]:

CORRELATION MATRIX
Year
Engine HP
Engine Cylinders
Number of Doors
Highway MPG
City mpg
Popularity
MSRP
0.3
0.7
0.5
-0.1
-0.4
-0.3
0.2
0.2
0.1
0.7
0.5
-0.1
-0.4
-0.3
0.0
0.6
-0.1
-0.6
-0.5
0.6
-0.1
-0.4
-0.3
0.0
0.3
-0.2
-0.2
-0.1
0.0
0.0
0.3
0.2
0.2
0.1
0.6

```
In [18]: %cd ..
/content/drive/MyDrive/Colab Notebooks/Regression Calculator/..
/content/drive/MyDrive
```

Engine Cylinders

**OF THIS CAR THE NUMBER OF CARS CONTAINING 4 AND 6 CYLINDERS ARE HIGH COMPARED TO OTHERS FOLLOWED BY 8 CYLINDER CARS**

```
In [19]: plt.figure(figsize=(8,8))
sns.pie(df['Transmission'], labels=df.index,
        colors=sns.color_palette('Set2'),
        autopct='%1.1f%%',
        shadow=True,
        startangle=90,
        explode=[0.0,0.0,0.0,
```