In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression, Ridge
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
```

In [2]:
```python
#### DATA CLEANING
```

In [3]:
```python
df = pd.read_csv("pakwheels_used_cars.csv")
df.head()
```

Out[3]:

	ad_ref	assembly	body	ad_city	color	engine_cc	fuel_type	make	mileage	model	registered	transmission	year	price
0	7927285	Imported	Van	Lahore	Pearl White	2000.0	Hybrid	Nissan	124000	Serena	Un-Registered	Automatic	1905.0	8990000.0
1	7679303	Imported	Hatchback	Lahore	Grey	996.0	Petrol	Toyota	30738	Vitz	Punjab	Automatic	1905.0	4190000.0
2	7915479	NaN	Sedan	Lahore	Super white	1798.0	Petrol	Toyota	183000	Corolla	Punjab	Automatic	1905.0	3990000.0
3	7918380	NaN	Sedan	Lahore	Crystal Black Pearl	1500.0	Petrol	Honda	41000	Civic	Punjab	Automatic	1905.0	6490000.0
4	7676167	Imported	MPV	Lahore	Silver	3000.0	Petrol	Toyota	126000	Alphard	Punjab	Automatic	1905.0	4750000.0

In [4]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77237 entries, 0 to 77236
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   ad_ref        77237 non-null  int64
 1   assembly      23976 non-null  object
 2   body          68372 non-null  object
 3   ad_city       77237 non-null  object
 4   color         75727 non-null  object
 5   engine_cc     77235 non-null  float64
 6   fuel_type     76303 non-null  object
 7   make          77237 non-null  object
 8   mileage       77237 non-null  int64
 9   model         77237 non-null  object
 10  registered    77237 non-null  object
 11  transmission  77237 non-null  object
 12  year          72516 non-null  float64
 13  price         76588 non-null  float64
dtypes: float64(3), int64(2), object(9)
memory usage: 8.2+ MB
```

In [6]:
```python
df.isna().sum().sort_values(ascending=False)
```

Out[6]:
```
assembly        53261
body             8865
year             4721
color            1510
fuel_type         934
price             649
engine_cc           2
ad_ref              0
ad_city             0
make                0
mileage             0
model               0
registered          0
transmission        0
dtype: int64
```

In [7]:
```python
df.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```python
##### DATA EXPLORATION AND VISUALIZATION
```

In [9]:
```python
df.columns
```

Out[9]:
```
Index(['ad_ref', 'assembly', 'body', 'ad_city', 'color', 'engine_cc',
       'fuel_type', 'make', 'mileage', 'model', 'registered', 'transmission',
       'year', 'price'],
      dtype='object')
```

In [10]:
```python
new_df = df.dropna()
print(new_df.to_string())
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [11]:
```python
new_df
```

Out[11]:

	ad_ref	assembly	body	ad_city	color	engine_cc	fuel_type	make	mileage	model	registered	transmission	year	price
0	7927285	Imported	Van	Lahore	Pearl White	2000.0	Hybrid	Nissan	124000	Serena	Un-Registered	Automatic	1905.0	8990000.0
1	7679303	Imported	Hatchback	Lahore	Grey	996.0	Petrol	Toyota	30738	Vitz	Punjab	Automatic	1905.0	4190000.0
4	7676167	Imported	MPV	Lahore	Silver	3000.0	Petrol	Toyota	126000	Alphard	Punjab	Automatic	1905.0	4750000.0
5	7908190	Imported	SUV	Lahore	White	2700.0	Petrol	Toyota	34000	Prado	Un-Registered	Automatic	1905.0	28900000.0
7	7900746	Imported	Van	Lahore	Black	2000.0	Hybrid	Nissan	97000	Serena	Un-Registered	Automatic	1905.0	7700000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
77202	7499054	Imported	Hatchback	Islamabad	Unlisted	1000.0	Petrol	Toyota	118000	Passo	Islamabad	Automatic	1905.0	2095000.0
77205	7756215	Imported	Crossover	Islamabad	White	1500.0	Petrol	Honda	54000	Vezel	Un-Registered	Automatic	1905.0	7600000.0
77208	7757605	Imported	Hatchback	Sialkot	Silver	660.0	Petrol	Daihatsu	109902	Mira	Punjab	Automatic	1905.0	2150000.0
77211	7757642	Imported	Hatchback	Karachi	Grey	1000.0	Petrol	Suzuki	70000	Cultus	Karachi	Manual	1905.0	1170000.0
77219	7847675	Imported	Hatchback	Karachi	Silver	1000.0	Petrol	Toyota	94890	Passo	Karachi	Automatic	1905.0	2300000.0

16168 rows × 14 columns

In [12]:
```python
new_df[new_df["make"]=="Toyota"]
```

Out[12]:

	ad_ref	assembly	body	ad_city	color	engine_cc	fuel_type	make	mileage	model	registered	transmission	year	price
1	7679303	Imported	Hatchback	Lahore	Grey	996.0	Petrol	Toyota	30738	Vitz	Punjab	Automatic	1905.0	4190000.0
4	7676167	Imported	MPV	Lahore	Silver	3000.0	Petrol	Toyota	126000	Alphard	Punjab	Automatic	1905.0	4750000.0
5	7908190	Imported	SUV	Lahore	White	2700.0	Petrol	Toyota	34000	Prado	Un-Registered	Automatic	1905.0	28900000.0
8	7917469	Imported	SUV	Lahore	White Pearl Crystal Shine	4600.0	Petrol	Toyota	9500	Land	Un-Registered	Automatic	1905.0	79800000.0
15	7816517	Imported	SUV	Lahore	Beige	4200.0	Diesel	Toyota	250000	Land	Punjab	Automatic	1905.0	8850000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
77153	7573090	Imported	Hatchback	Lahore	Red	1000.0	Petrol	Toyota	67000	Vitz	Punjab	Automatic	1905.0	3750000.0
77164	6312615	Imported	Sedan	Abbottabad	Black	2400.0	Petrol	Toyota	240000	Camry	Karachi	Automatic	1905.0	3800000.0
77187	7757734	Imported	Hatchback	Peshawar	White	996.0	Petrol	Toyota	7500	Vitz	Un-Registered	Automatic	1905.0	5500000.0
77202	7499054	Imported	Hatchback	Islamabad	Unlisted	1000.0	Petrol	Toyota	118000	Passo	Islamabad	Automatic	1905.0	2095000.0
77219	7847675	Imported	Hatchback	Karachi	Silver	1000.0	Petrol	Toyota	94890	Passo	Karachi	Automatic	1905.0	2300000.0

7589 rows × 14 columns

In [15]:
```python
new_df.describe()
```

Out[15]:

	ad_ref	engine_cc	mileage	year	price
count	1.616800e+04	16168.000000	16168.000000	16168.0	1.616800e+04
mean	7.779527e+06	1551.346982	102151.778389	1905.0	5.444228e+06
std	3.033256e+05	964.852260	78178.429289	0.0	8.561760e+06
min	1.117870e+06	100.000000	1.000000	1905.0	1.650000e+05
25%	7.785870e+06	996.000000	55412.250000	1905.0	2.175000e+06
50%	7.851165e+06	1400.000000	96000.000000	1905.0	3.325000e+06
75%	7.895679e+06	1800.000000	135000.000000	1905.0	5.200000e+06
max	7.931905e+06	6800.000000	1000000.000000	1905.0	1.800000e+08

In [16]:
```python
cities = df.ad_city.unique()
len(cities)
```

Out[16]:
```
297
```

In [17]:
```python
cities_by_price = df.ad_city.value_counts()
```

In [18]:
```python
cities_by_price
```

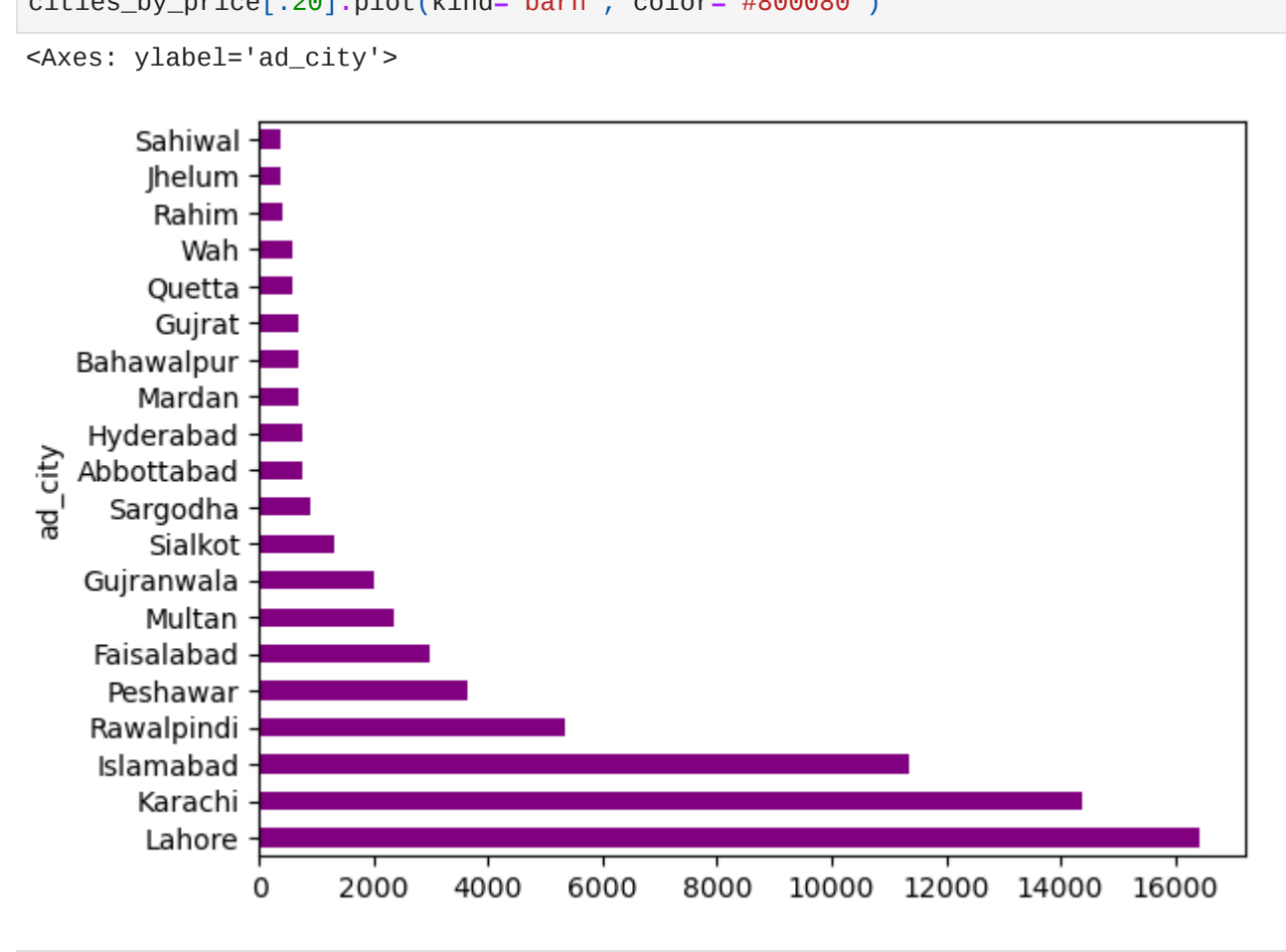
Out[18]:
```
ad_city
Lahore        16418
Karachi       14364
Islamabad     11344
Rawalpindi     5352
Peshawar       3634
              ...
Gakhar            1
Bhan              1
Lasbela           1
Jatlaan           1
Makli             1
Name: count, Length: 297, dtype: int64
```

In [19]:
```python
cities_by_price[:10]
```

Out[19]:
```
ad_city
Lahore        16418
Karachi       14364
Islamabad     11344
Rawalpindi     5352
Peshawar       3634
Faisalabad     2979
Multan         2341
Gujranwala     1994
Sialkot        1323
Sargodha        889
Name: count, dtype: int64
```

In [20]:
```python
cities_by_price[:20].plot(kind="barh", color='#800080')
```

Out[20]:
```
<Axes: ylabel='ad_city'>
```



In [21]:
```python
df['fuel_type'].value_counts()
```

Out[21]:
```
fuel_type
Petrol    70028
Diesel     3444
Hybrid     2831
Name: count, dtype: int64
```

In [22]:
```python
df['body'].value_counts()
```

Out[22]:
```
body
Sedan                29951
Hatchback            24868
SUV                   5001
Crossover             2135
Mini Van              1324
Compact sedan          771
MPV                    764
Double Cabin           751
Van                    735
Pick Up                520
Micro Van              516
Compact SUV            462
Station Wagon          231
Coupe                   89
Truck                   85
High Roof               75
Convertible             45
Single Cabin            25
Off-Road Vehicles       12
Mini Vehicles            8
Compact hatchback        4
Name: count, dtype: int64
```

In [23]:
```python
df['color'].value_counts()
```

Out[23]:
```
color
White                          21324
Silver                          8116
Black                           6962
Grey                            4164
Solid White                     4070
                                ...
Crimson Red                        1
Sun Gold Black                     1
Premium Silver Metallic            1
Florett Silver Metallic            1
Atlantis Turquoise Pearl           1
Name: count, Length: 386, dtype: int64
```

In [24]:
```python
df['make'].value_counts()
```

Out[24]:
```
make
Toyota      24678
Suzuki      22322
Honda       16270
Daihatsu     3143
KIA          1687
            ...
GUGU            1
Roma            1
Classic         1
Opel            1
Rinco           1
Name: count, Length: 69, dtype: int64
```

In [25]:
```python
imported_cars=df[df.price>7000000]
imported_cars
```

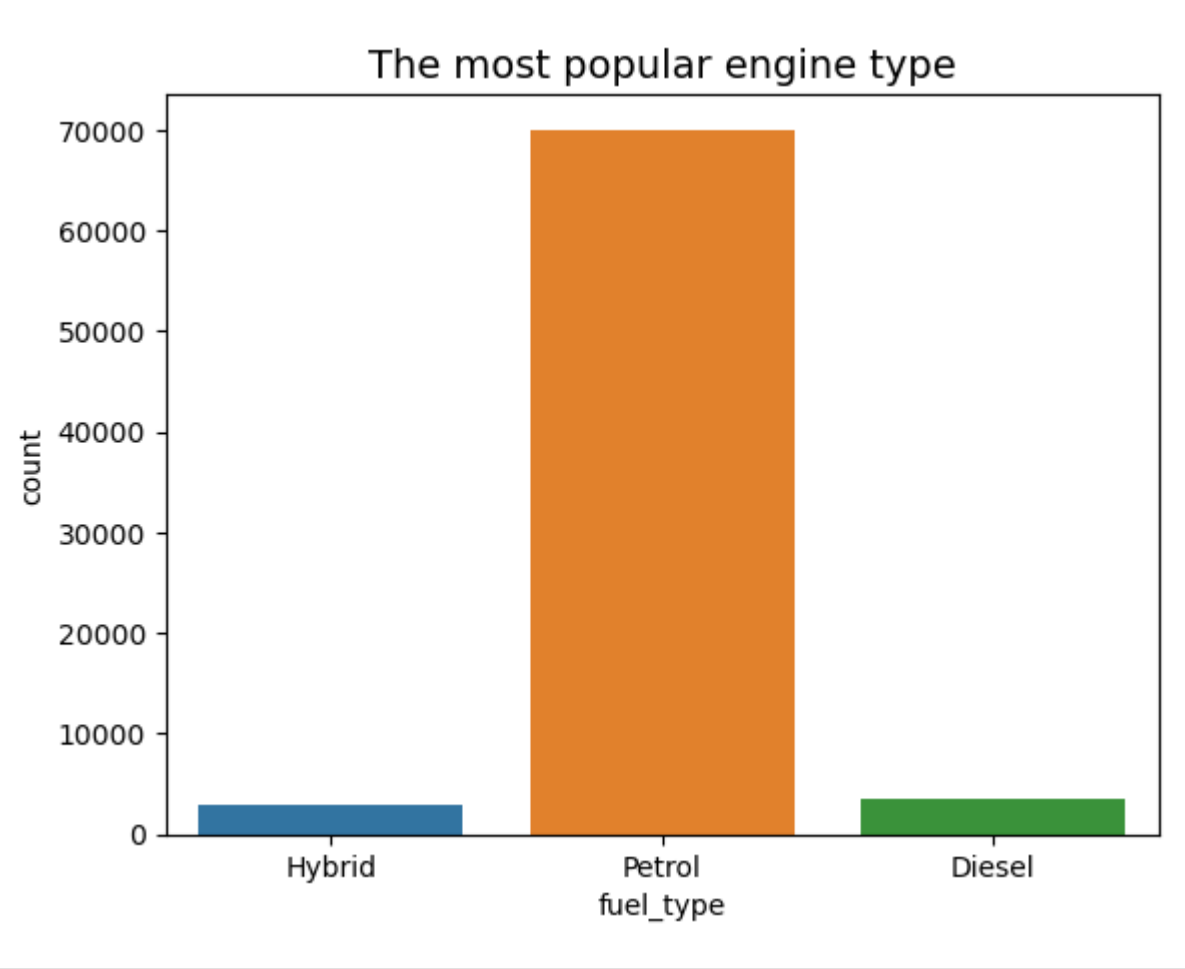
Out[25]:

	ad_ref	assembly	body	ad_city	color	engine_cc	fuel_type	make	mileage	model	registered	transmission	year	price
0	7927285	Imported	Van	Lahore	Pearl White	2000.0	Hybrid	Nissan	124000	Serena	Un-Registered	Automatic	1905.0	8990000.0
5	7908190	Imported	SUV	Lahore	White	2700.0	Petrol	Toyota	34000	Prado	Un-Registered	Automatic	1905.0	28900000.0
7	7900746	Imported	Van	Lahore	Black	2000.0	Hybrid	Nissan	97000	Serena	Un-Registered	Automatic	1905.0	7700000.0
8	7917469	Imported	SUV	Lahore	White Pearl Crystal Shine	4600.0	Petrol	Toyota	9500	Land	Un-Registered	Automatic	1905.0	79800000.0
10	7931905	Imported	Sedan	Sialkot	Black	1800.0	Petrol	Audi	61000	A4	Islamabad	Automatic	1905.0	7300000.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
77183	7757733	NaN	SUV	Sialkot	Polar White	1999.0	Petrol	Hyundai	20000	Tucson	Islamabad	Automatic	1905.0	8000000.0
77184	7757731	NaN	Sedan	Karachi	Diamond Metallic Black	2497.0	Petrol	Hyundai	60	Sonata	Karachi	Automatic	NaN	12500000.0
77196	7743710	NaN	Compact SUV	Islamabad	Black	1500.0	Petrol	Haval	20	Jolion	Un-Registered	Automatic	NaN	9100000.0
77205	7756215	Imported	Crossover	Islamabad	White	1500.0	Petrol	Honda	54000	Vezel	Un-Registered	Automatic	1905.0	7600000.0
77216	7757677	NaN	Sedan	Bahawalpur	White	1800.0	Petrol	Toyota	10800	Corolla	Islamabad	Automatic	1905.0	7600000.0

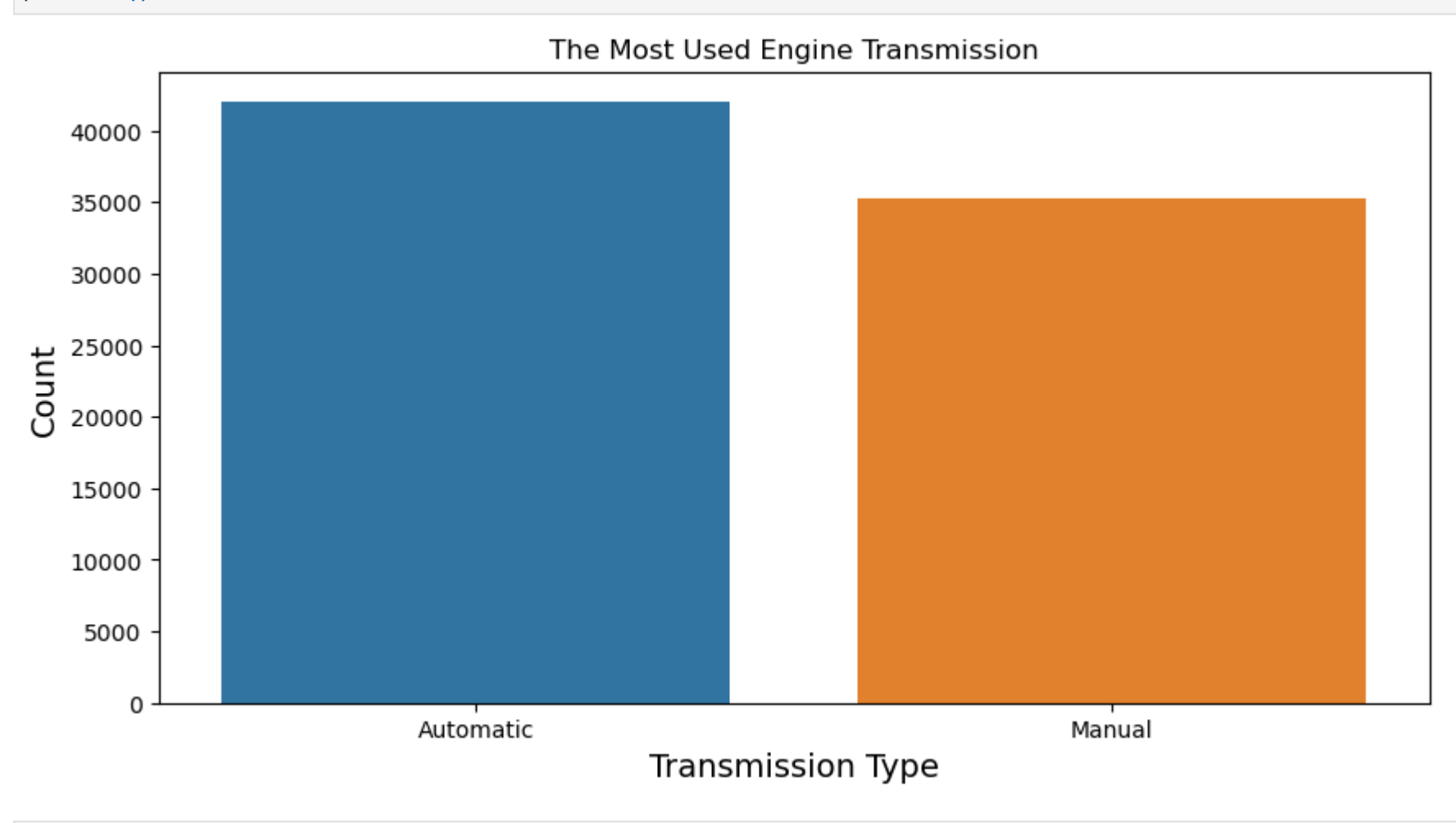
8093 rows × 14 columns

In [26]:
```python
sns.countplot(data=df, x='fuel_type')
plt.title('The most popular engine type',size=14)
```

Out[26]:
```
Text(0.5, 1.0, 'The most popular engine type')
```



In [27]:
```python
plt.figure(figsize=(10,5))
sns.countplot(data=df ,x="transmission" )
plt.xlabel('Transmission Type',size=14)
plt.ylabel('Count',size=14)
plt.title('The Most Used Engine Transmission')
plt.show()
```



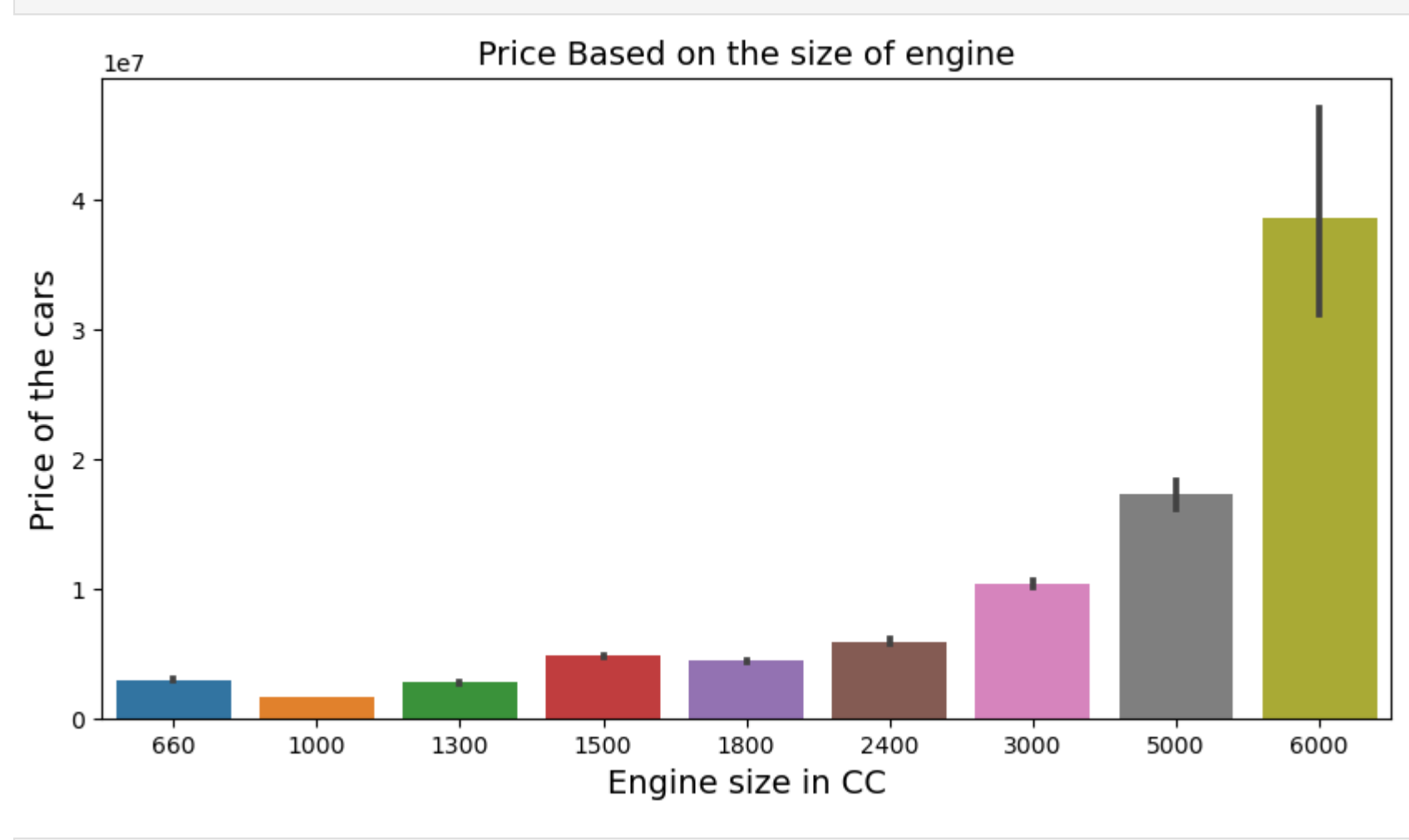
In [28]:
```python
# Here we are binning the engine capacity into different bins.
group_names=['660','1000', '1300', '1500', '1800', '2400','3000', '5000', '6000']
cut_bins = [0, 660, 1000, 1300, 1500, 1800, 2400, 3000,5000, 6000]
df['Engine Size'] = pd.cut(df['engine_cc'], bins=cut_bins, labels=group_names)
```

In [29]:
```python
df.head()
```

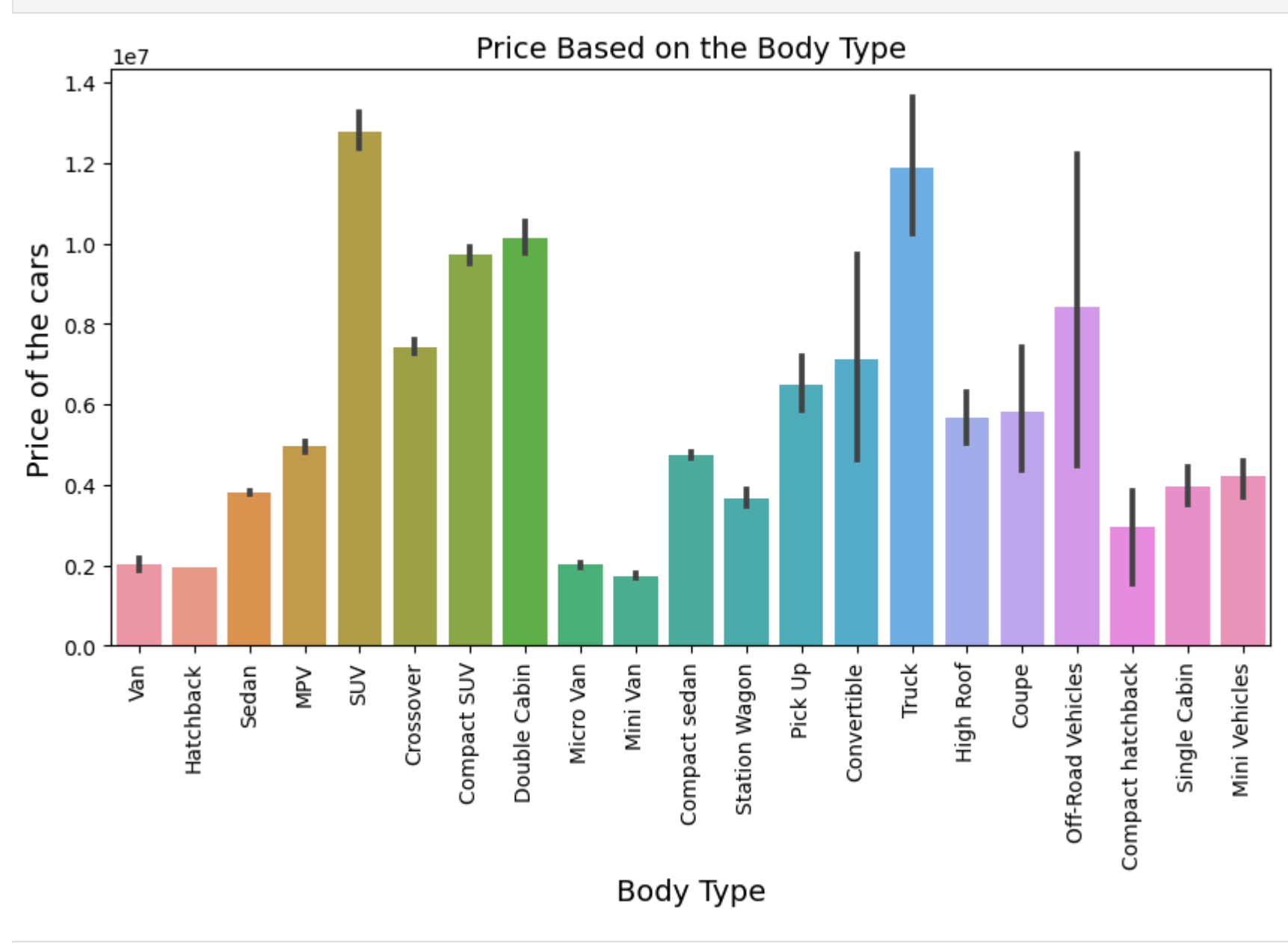
Out[29]:

	ad_ref	assembly	body	ad_city	color	engine_cc	fuel_type	make	mileage	model	registered	transmission	year	price	Engine Size
0	7927285	Imported	Van	Lahore	Pearl White	2000.0	Hybrid	Nissan	124000	Serena	Un-Registered	Automatic	1905.0	8990000.0	2400
1	7679303	Imported	Hatchback	Lahore	Grey	996.0	Petrol	Toyota	30738	Vitz	Punjab	Automatic	1905.0	4190000.0	1000
2	7915479	NaN	Sedan	Lahore	Super white	1798.0	Petrol	Toyota	183000	Corolla	Punjab	Automatic	1905.0	3990000.0	1800
3	7918380	NaN	Sedan	Lahore	Crystal Black Pearl	1500.0	Petrol	Honda	41000	Civic	Punjab	Automatic	1905.0	6490000.0	1500
4	7676167	Imported	MPV	Lahore	Silver	3000.0	Petrol	Toyota	126000	Alphard	Punjab	Automatic	1905.0	4750000.0	3000

In [30]:
```python
plt.figure(figsize=(10,5))
sns.barplot(data=df,x='Engine Size',y='price')
plt.title("Price Based on the size of engine", size=14)
plt.xlabel('Engine size in CC',size=14)
plt.ylabel('Price of the cars',size=14)
plt.xticks(rotation=0)
plt.show()
```



In [31]:
```python
plt.figure(figsize=(10,5))
sns.barplot(data=df,x='body',y='price')
plt.title("Price Based on the Body Type", size=14)
plt.xlabel('Body Type',size=14)
plt.ylabel('Price of the cars',size=14)
plt.xticks(rotation=90)
plt.show()
```



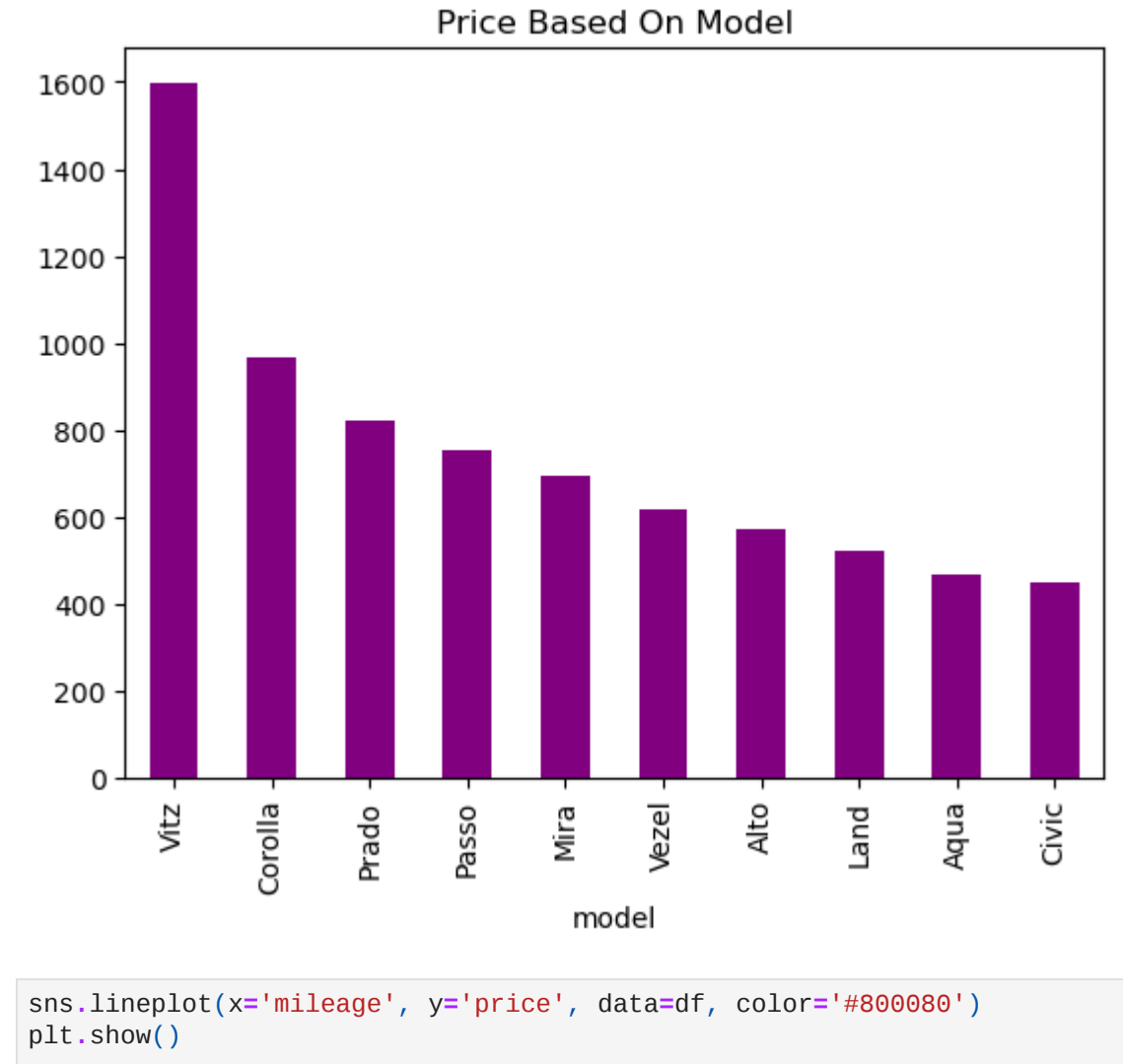
In [32]:
```python
df_group=df[['transmission','body', 'price']]
grouped_df=df_group.groupby(['transmission','body']).mean().sort_values(by='price', ascending=False)
grouped_df
```

Out[32]:

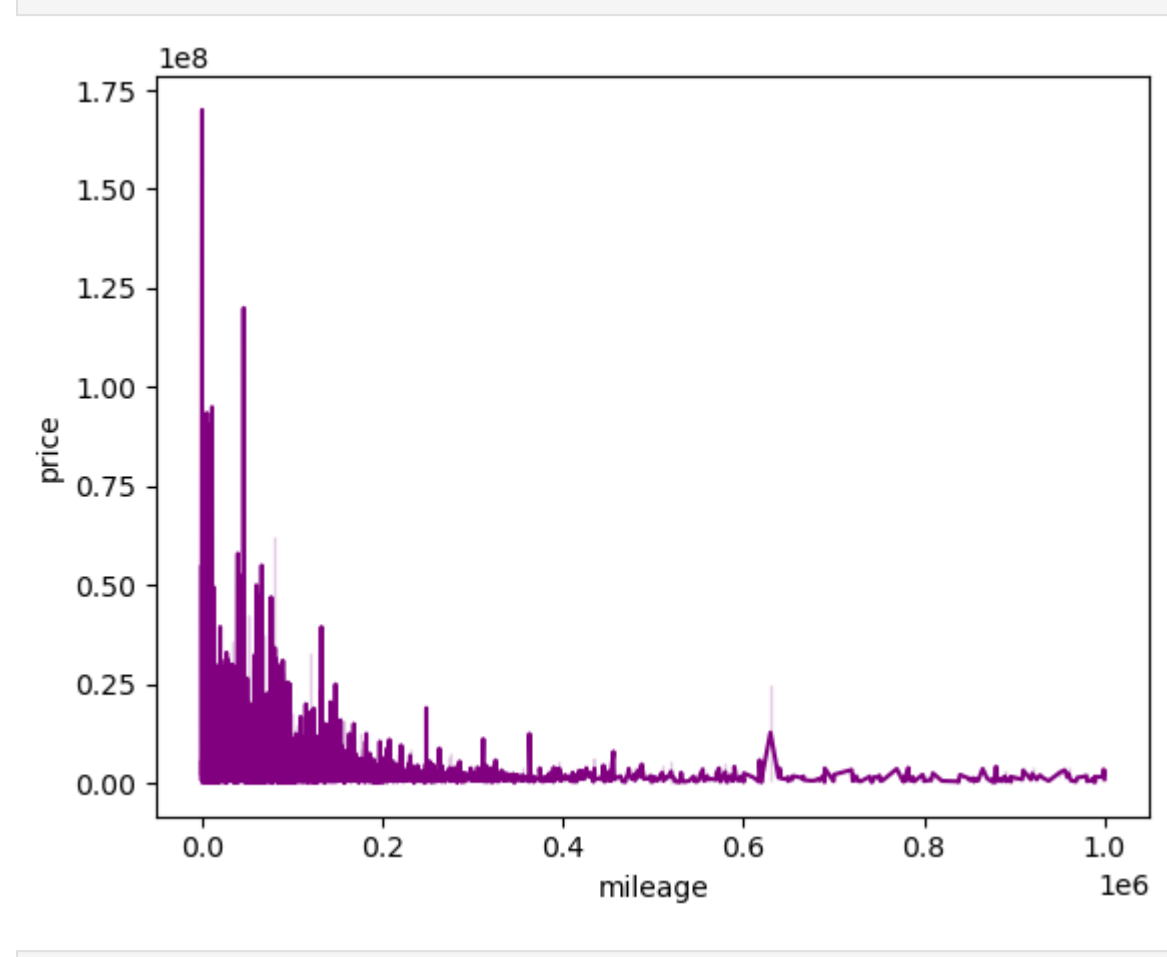
		price
transmission	body	
Automatic	Truck	1.650465e+07
	Pick Up	1.587910e+07
	Off-Road Vehicles	1.388333e+07
	SUV	1.362961e+07
Manual	Compact SUV	1.100000e+07
Automatic	Double Cabin	1.053871e+07
	Compact SUV	9.706819e+06
	Convertible	8.191515e+06
	Coupe	7.659907e+06
	Crossover	7.441756e+06
Manual	High Roof	6.245000e+06
Automatic	Single Cabin	5.750000e+06
	High Roof	5.648984e+06
Manual	Double Cabin	5.423158e+06
Automatic	Van	5.235519e+06
	Compact sedan	5.172347e+06
	MPV	5.142566e+06
	Sedan	4.878547e+06
	Mini Vehicles	4.228750e+06
Manual	Truck	3.923800e+06
	Single Cabin	3.782500e+06
	Crossover	3.743500e+06
Automatic	Station Wagon	3.693237e+06
Manual	Compact sedan	3.523510e+06
	Convertible	3.512000e+06
	MPV	3.022000e+06
Automatic	Hatchback	2.964102e+06
	Compact hatchback	2.952500e+06
Manual	Coupe	2.737000e+06
Automatic	Mini Van	2.446187e+06
Manual	SUV	2.388323e+06
	Sedan	2.307182e+06
Automatic	Micro Van	2.051941e+06
Manual	Station Wagon	2.050000e+06
	Off-Road Vehicles	1.890000e+06
	Micro Van	1.645521e+06
	Mini Van	1.488299e+06
	Van	1.460105e+06
	Hatchback	1.410876e+06
	Pick Up	1.353674e+06

In [33]:
```python
new_df.groupby('model')['price'].count().nlargest(10).plot(kind='bar', color='#800080')
plt.title("Price Based On Model")
```

Out[33]:
```
Text(0.5, 1.0, 'Price Based On Model')
```

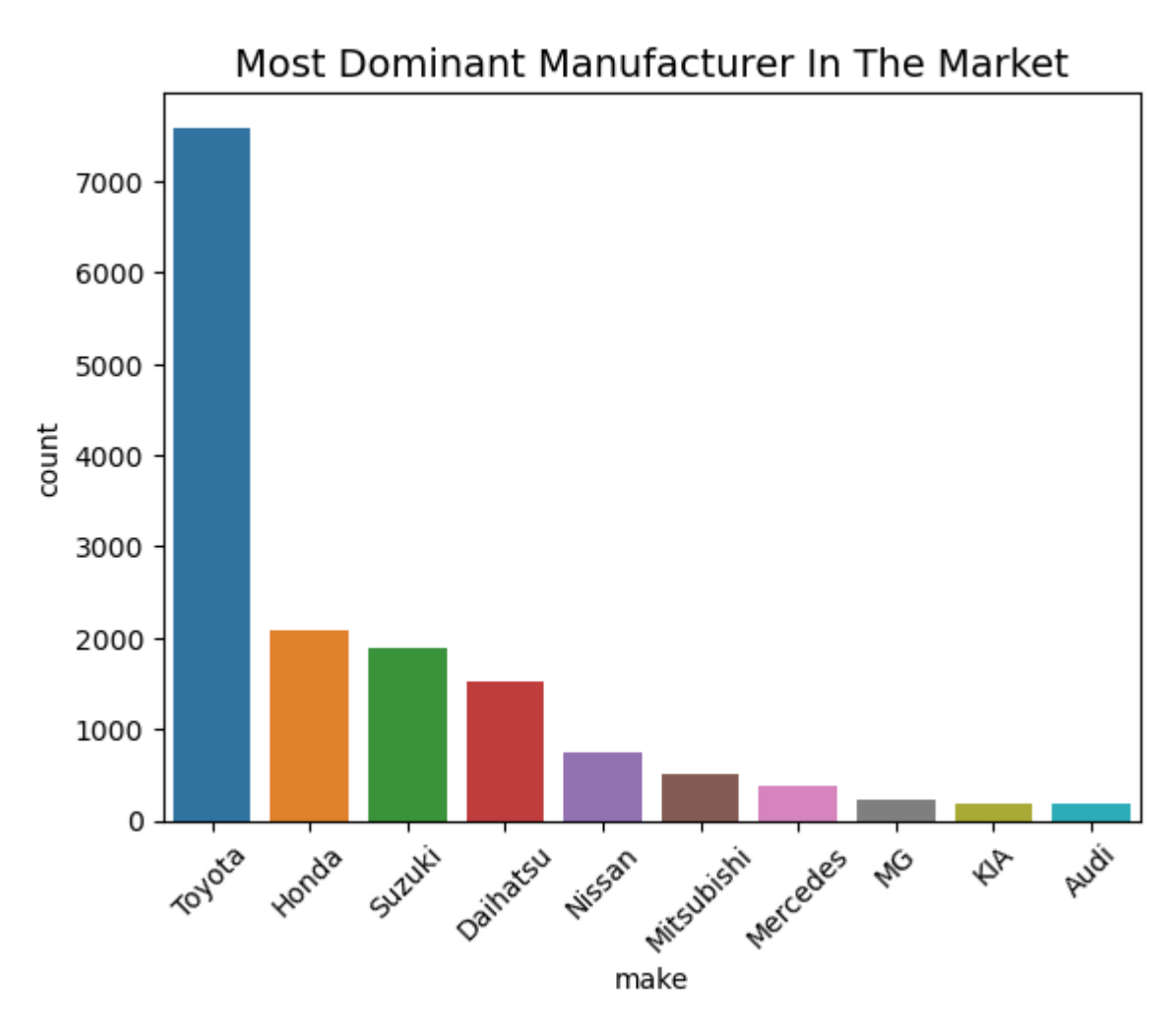


In [34]:
```python
sns.lineplot(x='mileage', y='price', data=df, color='#800080')
plt.show()
```



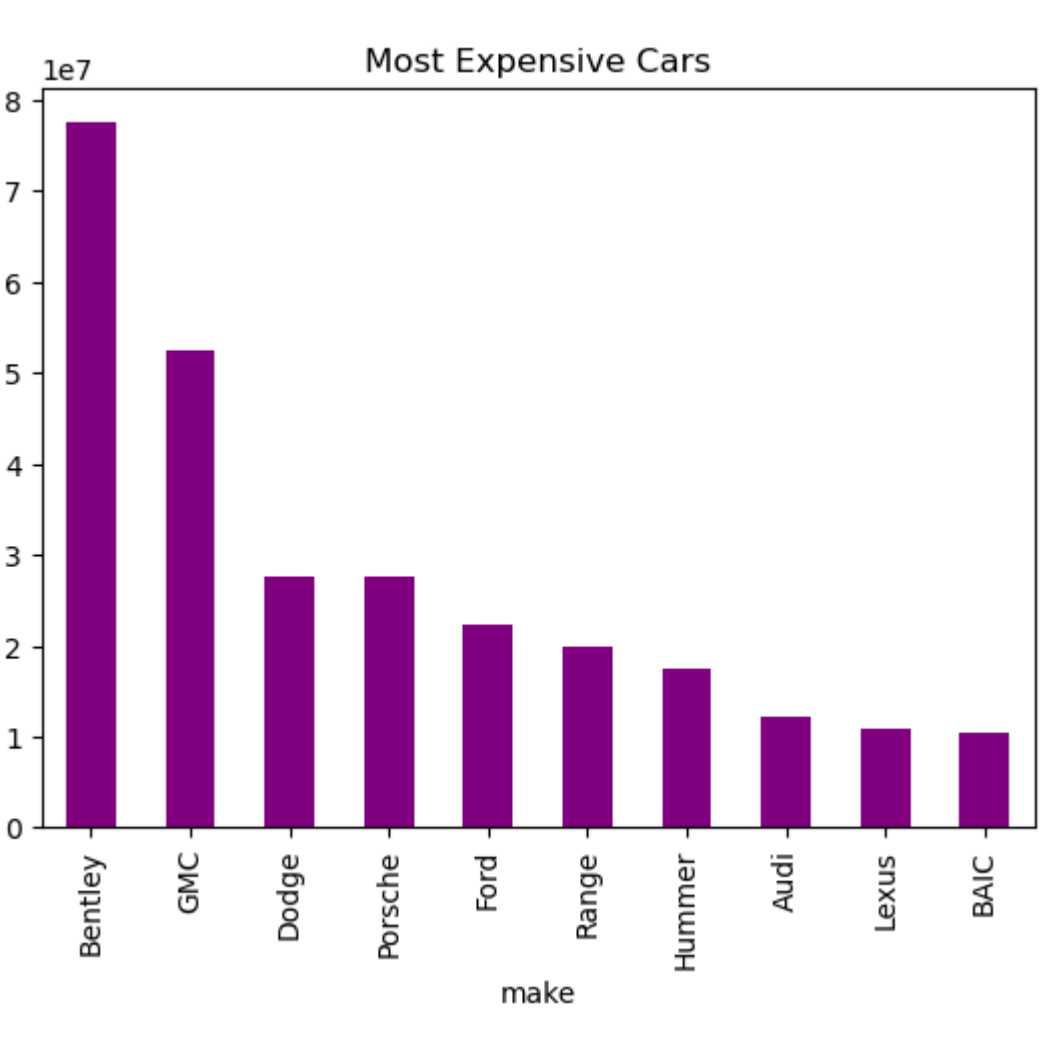
In [35]:
```python
plt.xticks(rotation=45)
sns.countplot(data=new_df,x='make',order=new_df['make'].value_counts().iloc[:10].index)
plt.title("Most Dominant Manufacturer In The Market", size=14)
```

Out[35]:
```
Text(0.5, 1.0, 'Most Dominant Manufacturer In The Market')
```



In [36]:
```python
new_df.groupby('make')['price'].median().nlargest(10).plot(kind='bar', color='#800080')
plt.title('Most Expensive Cars')
```

Out[36]:
```
Text(0.5, 1.0, 'Most Expensive Cars')
```



In [ ]: