Data Cleaning

In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [6]:
```
df=pd.read_csv("global_cars_enhanced.csv")
df
```

Out[6]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic
...	...	...	...	...	...	...
295	CAR_0296	Audi	2015	Pickup	Hybrid	Automatic
296	CAR_0297	Ford	2023	Hatchback	Petrol	Manual
297	CAR_0298	Mercedes	2020	SUV	Electric	Automatic
298	CAR_0299	Ford	2023	Coupe	Diesel	Manual
299	CAR_0300	Kia	2023	SUV	Hybrid	Manual

300 rows × 16 columns

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Car_ID                 300 non-null    object 
 1   Brand                  300 non-null    object 
 2   Manufacture_Year       300 non-null    int64  
 3   Body_Type              300 non-null    object 
 4   Fuel_Type              300 non-null    object 
 5   Transmission           300 non-null    object 
 6   Engine_CC              300 non-null    int64  
 7   Horsepower             300 non-null    int64  
 8   Mileage_km_per_l       300 non-null    int64  
 9   Price_USD              300 non-null    int64  
 10  Manufacturing_Country  300 non-null    object 
 11  Car_Age                300 non-null    int64  
 12  Price_Category         300 non-null    object 
 13  HP_per_CC              300 non-null    float64
 14  Age_Category           300 non-null    object 
 15  Efficiency_Score       300 non-null    float64
dtypes: float64(2), int64(6), object(8)
memory usage: 37.6+ KB
```

In [10]:
```
# Cheking Duplicated Values
df["Car_ID"].duplicated()
```

Out[10]:
```
0      False
1      False
2      False
3      False
4      False
       ...
295    False
296    False
297    False
298    False
299    False
Name: Car_ID, Length: 300, dtype: bool
```

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

Out[12]:
```
Car_ID                  0
Brand                   0
Manufacture_Year        0
Body_Type               0
Fuel_Type               0
Transmission            0
Engine_CC               0
Horsepower              0
Mileage_km_per_l        0
Price_USD               0
Manufacturing_Country   0
Car_Age                 0
Price_Category          0
HP_per_CC               0
Age_Category            0
Efficiency_Score        0
dtype: int64
```

In [15]:
```
df.shape
```

Out[15]:
```
(300, 16)
```

In [13]:
```
df.head()
```

Out[13]:

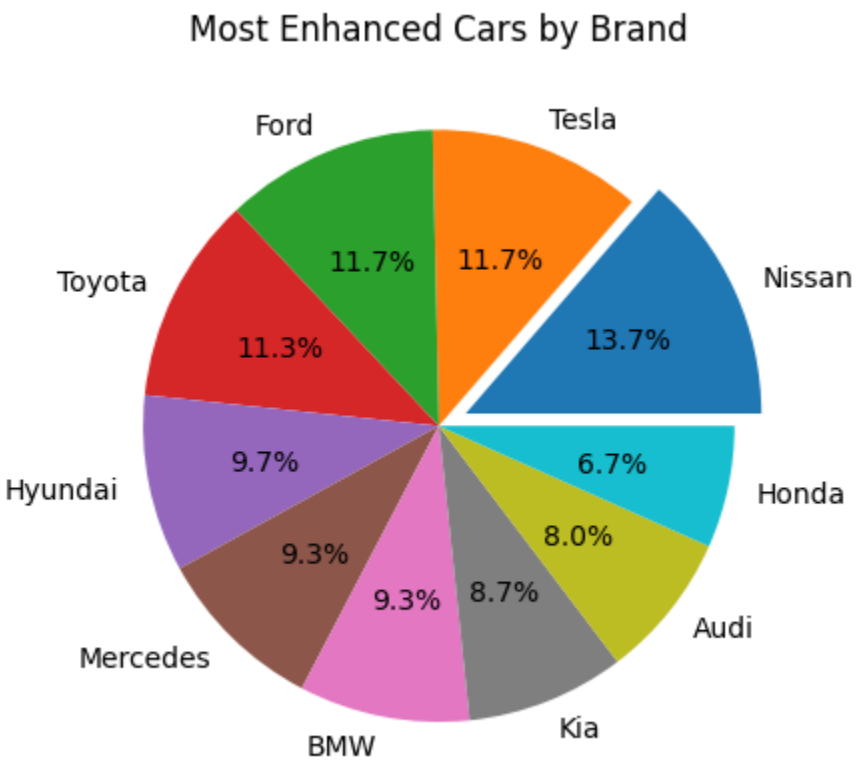
	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [18]:
```
count=df["Brand"].value_counts()
count
```

Out[18]:
```
Brand
Nissan      41
Tesla       35
Ford        35
Toyota      34
Hyundai     29
Mercedes    28
BMW         28
Kia         26
Audi        24
Honda       20
Name: count, dtype: int64
```

Data Visualization

In [23]:
```
plt.pie(count.values,labels=count.index,autopct="%1.1f%%",explode=(0.1,0,0,0,0
plt.title("Most Enhanced Cars by Brand")
plt.show()
```



In [21]:
```
# Most of the Nissan Cars are enhanced
```

In [24]:
```
df.head()
```

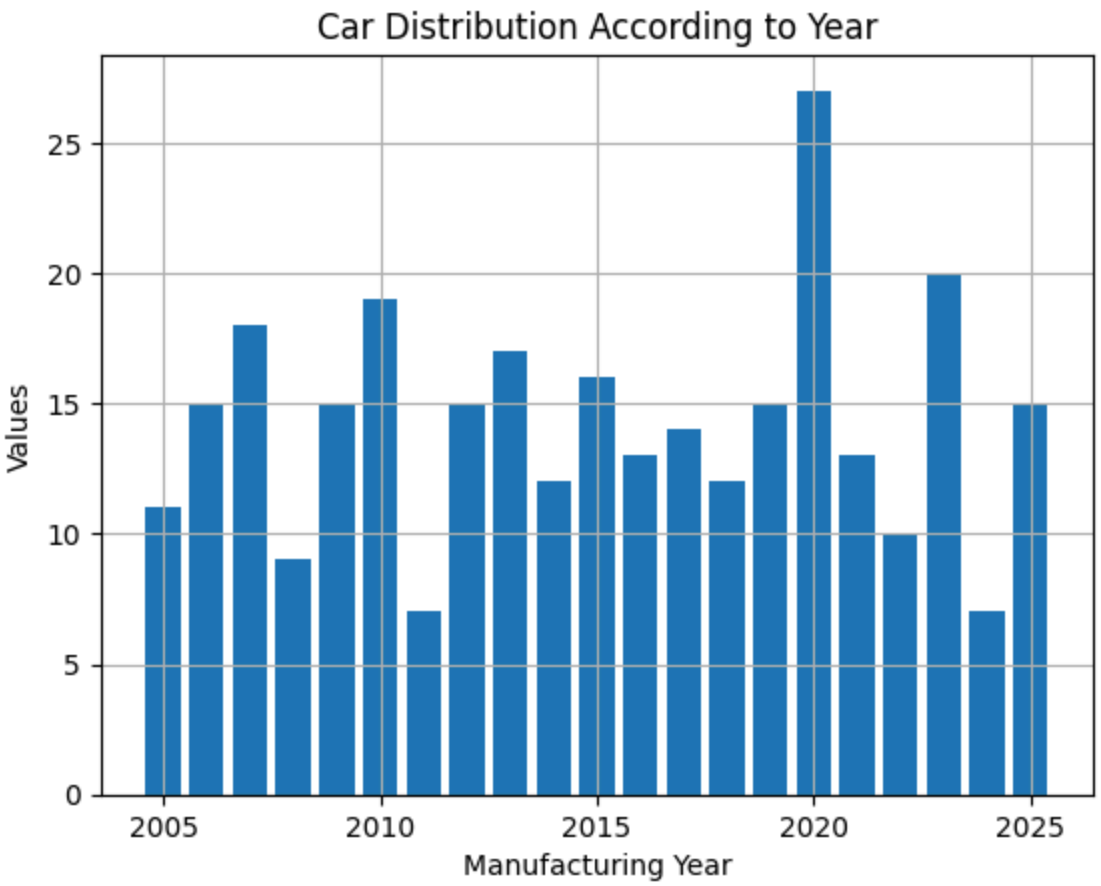
Out[24]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [105…
```
count=df["Manufacture_Year"].value_counts()
count
```

Out[105…
```
Manufacture_Year
2020    27
2023    20
2010    19
2007    18
2013    17
2015    16
2006    15
2009    15
2012    15
2025    15
2019    15
2017    14
2016    13
2021    13
2014    12
2018    12
2005    11
2022    10
2008     9
2011     7
2024     7
Name: count, dtype: int64
```

In [106…
```
plt.bar(count.index,count.values)
plt.xlabel("Manufacturing Year")
plt.ylabel("Values")
plt.title("Car Distribution According to Year")
plt.grid()
plt.show()
```



In [30]:
```
# 2020 has a most cars manufacture year.
```

In [31]:
```
df.head()
```

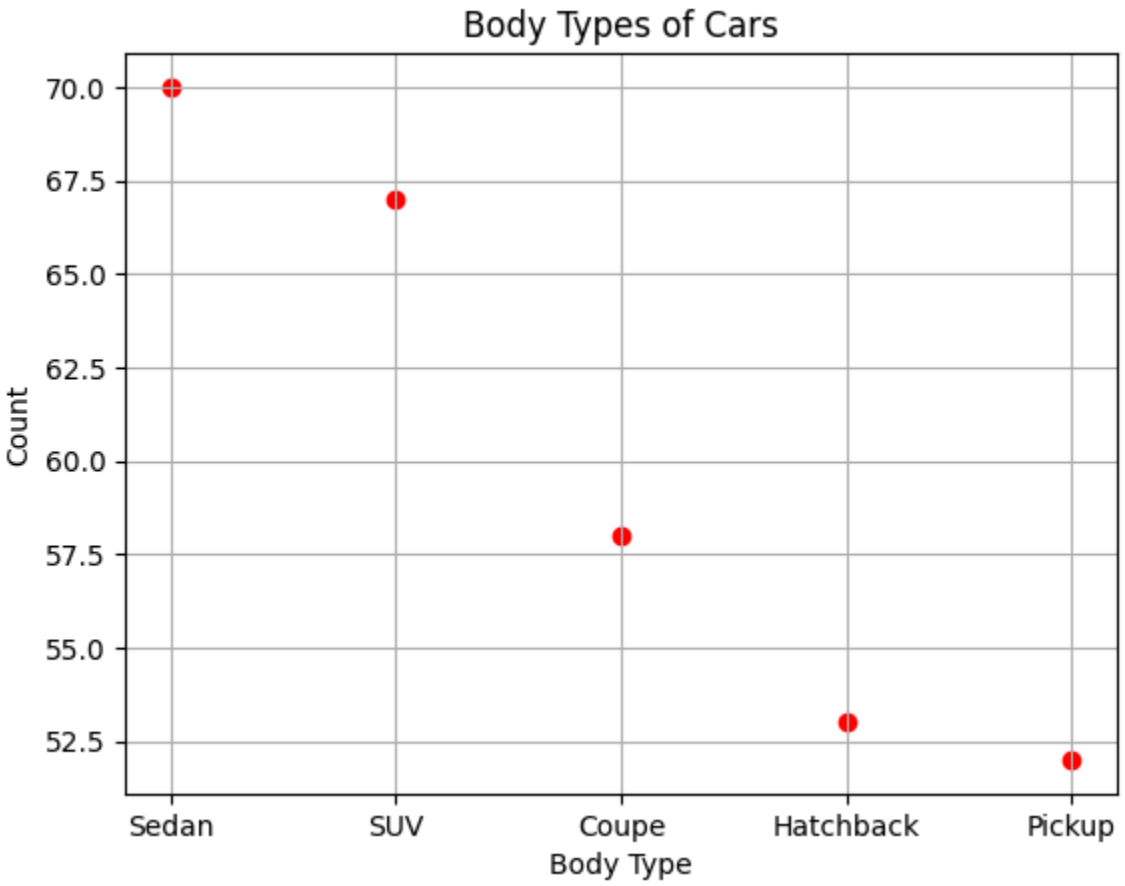
Out[31]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [33]:
```
count=df["Body_Type"].value_counts()
count
```

Out[33]:
```
Body_Type
Sedan        70
SUV          67
Coupe        58
Hatchback    53
Pickup       52
Name: count, dtype: int64
```

In [37]:
```
plt.scatter(count.index,count.values,alpha=1,color='red')
plt.xlabel("Body Type")
plt.ylabel("Count")
plt.title("Body Types of Cars")
plt.grid()
plt.show()
```



In [38]:
```
# Sedan have most body types than other
```

In [39]:
```
df.head()
```

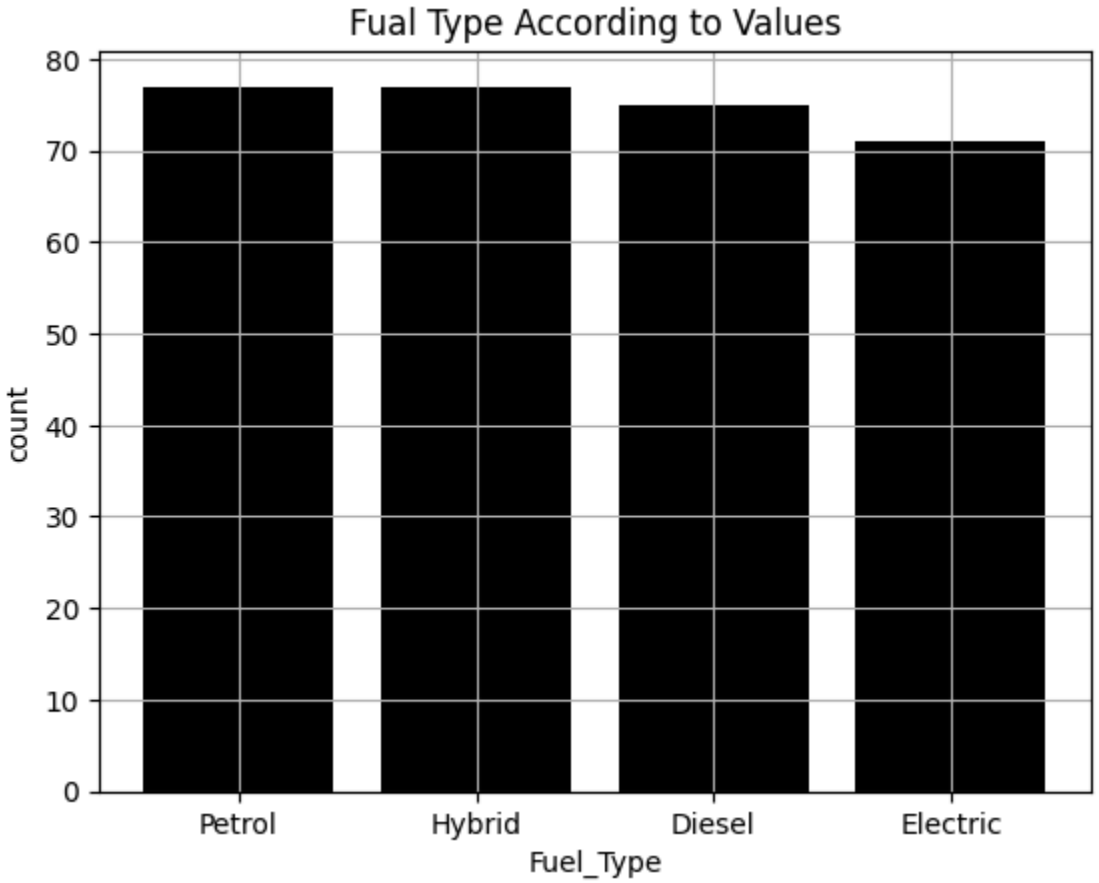
Out[39]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [42]:
```
count=df["Fuel_Type"].value_counts()
count
```

Out[42]:
```
Fuel_Type
Petrol      77
Hybrid      77
Diesel      75
Electric    71
Name: count, dtype: int64
```

In [44]:
```
plt.bar(count.index,count.values,color='black')
plt.xlabel("Fuel_Type")
plt.ylabel("count")
plt.title("Fual Type According to Values")
plt.grid()
plt.show()
```



In [45]:
```
# Petrol and Hybrid Fual Types car are equal with 1st rank
```

In [46]:
```
df.head()
```

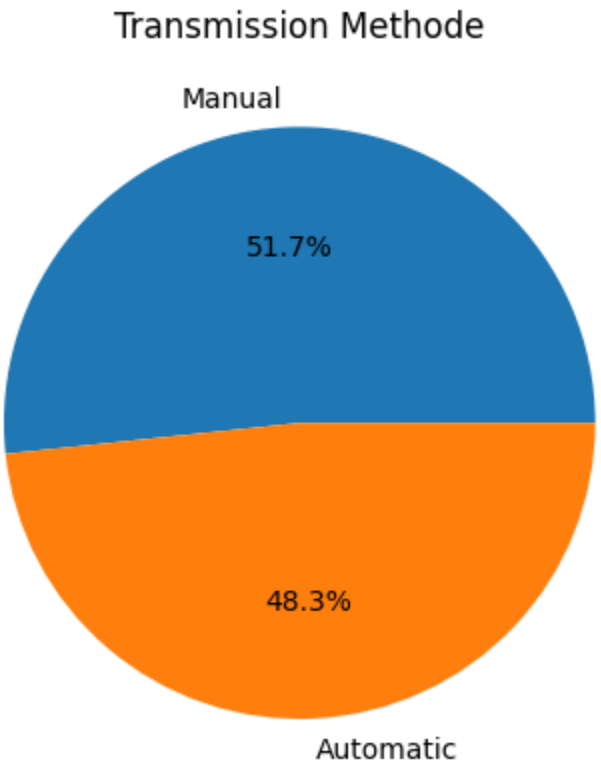
Out[46]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [47]:
```
count=df["Transmission"].value_counts()
count
```

Out[47]:
```
Transmission
Manual       155
Automatic    145
Name: count, dtype: int64
```

In [50]:
```
plt.pie(count.values,labels=count.index,explode=(0,0),autopct='%1.1f%%')
plt.title("Transmission Methode")
plt.show()
```



In [ ]:
```
# Manual Transmission is more than Automatic
```

In [51]:
```
df.head()
```

Out[51]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [53]:
```
df.describe()
```

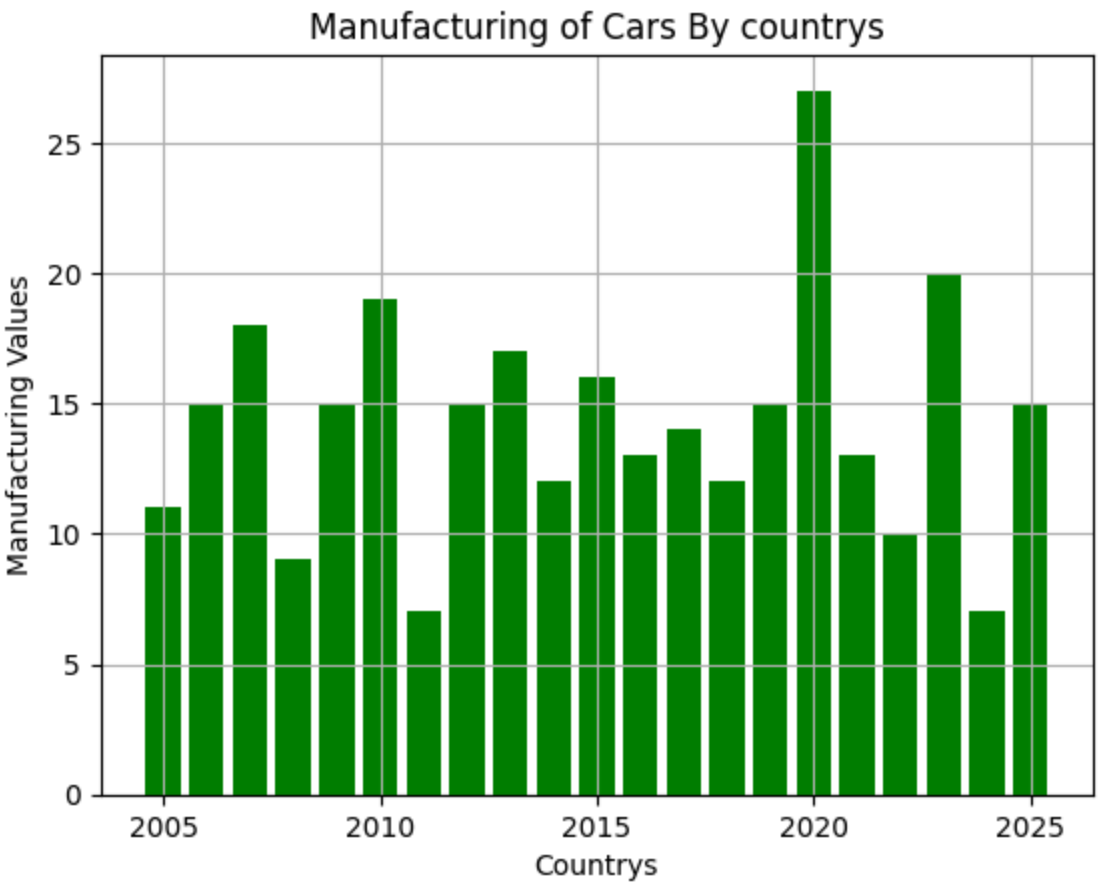
Out[53]:

	Manufacture_Year	Engine_CC	Horsepower	Mileage_km_per_l	Price
count	300.000000	300.000000	300.000000	300.000000	300.00
mean	2015.123333	3052.880000	328.346667	19.716667	60848.82
std	5.966023	1117.937497	153.202644	6.028061	34445.52
min	2005.000000	1001.000000	70.000000	10.000000	5221.00
25%	2010.000000	2074.000000	188.250000	15.000000	29418.50
50%	2015.000000	3117.500000	329.500000	19.500000	59179.50
75%	2020.000000	3964.000000	454.750000	24.000000	89692.25
max	2025.000000	4994.000000	599.000000	30.000000	119587.00

In [55]:
```
count=df["Manufacturing_Country"].value_counts()
count
```

Out[55]:
```
Manufacturing_Country
Germany        55
USA            54
China          53
Japan          49
UK             46
South Korea    43
Name: count, dtype: int64
```

In [107…
```
plt.bar(count.index,count.values,color="green")
plt.xlabel("Countrys")
plt.ylabel("Manufacturing Values")
plt.title("Manufacturing of Cars By countrys")
plt.grid()
plt.show()
```



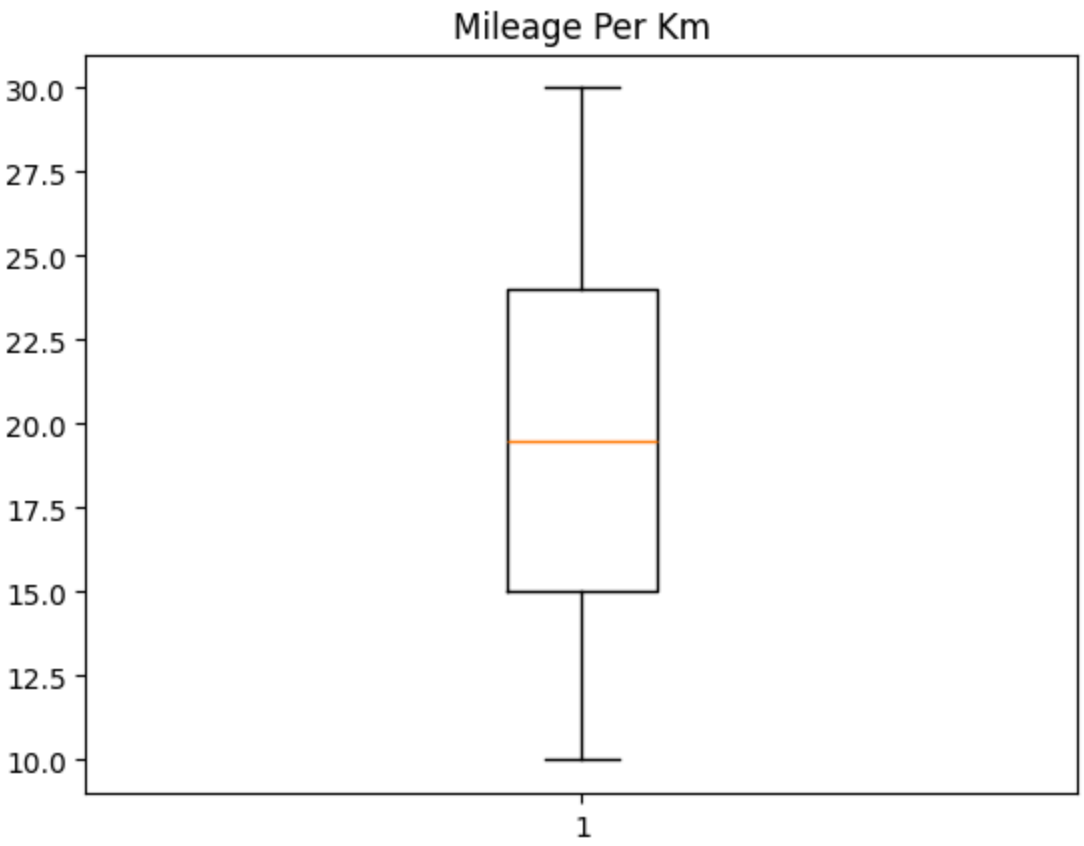
In [59]:
```
# Germany Manufacture Most Cars
# Usa is Closer to Germany
# Germany Manufacture only one more car compaire to Usa
```

In [60]:
```
df.head()
```

Out[60]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [62]:
```
plt.boxplot(df["Mileage_km_per_l"])
plt.title("Mileage Per Km")
plt.show()
```



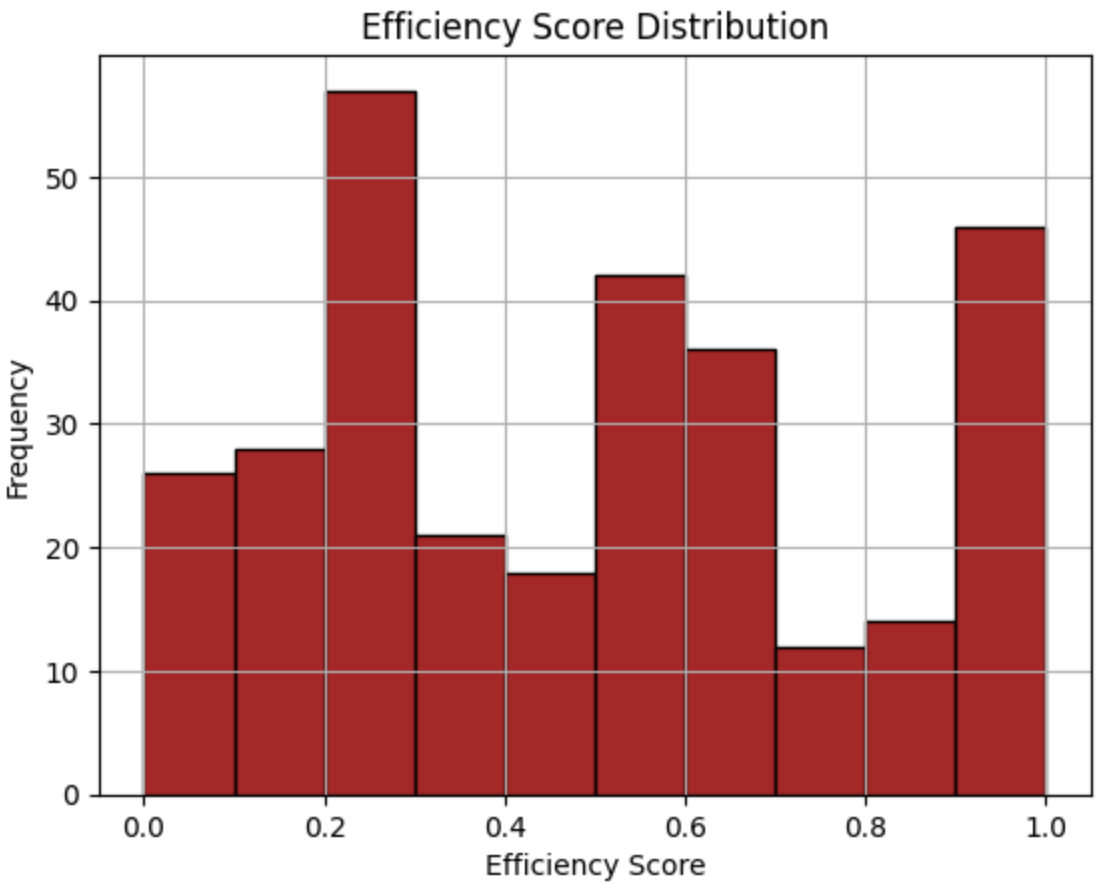
In [63]:
```
# here minimum mileage have 20 per km
```

In [64]:
```
df.head()
```

Out[64]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [66]:
```
plt.hist(df["Efficiency_Score"],color='brown',edgecolor='black')
plt.xlabel("Efficiency Score")
plt.ylabel("Frequency")
plt.title("Efficiency Score Distribution")
plt.grid()
plt.show()
```



In [67]:
```
df.head()
```

Out[67]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

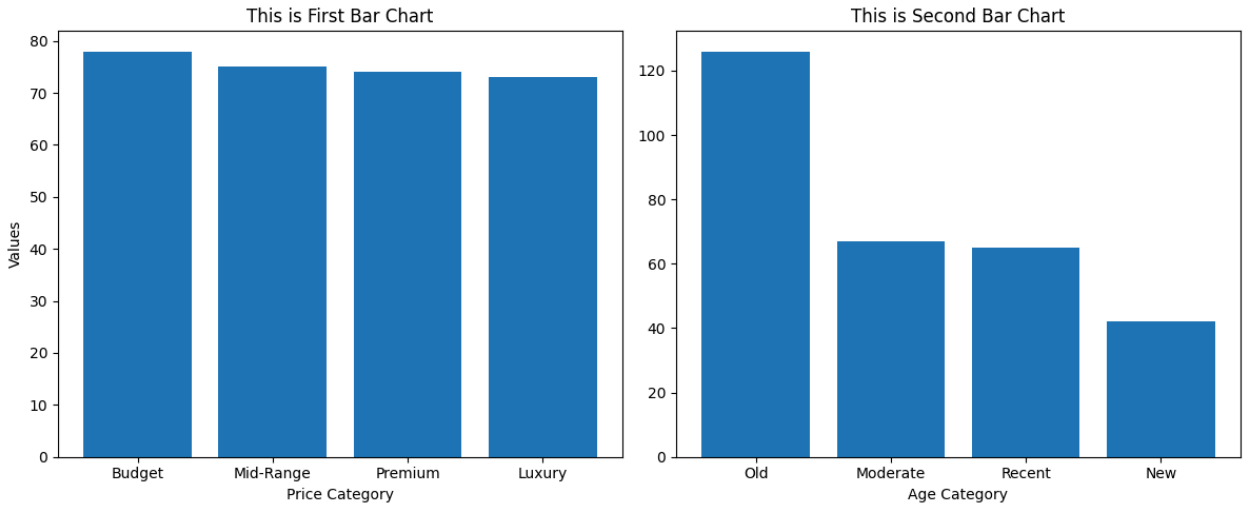
In [68]:
```
count=df["Price_Category"].value_counts()
count
```

Out[68]:
```
Price_Category
Budget       78
Mid-Range    75
Premium      74
Luxury       73
Name: count, dtype: int64
```

In [69]:
```
counts=df["Age_Category"].value_counts()
counts
```

Out[69]:
```
Age_Category
Old         126
Moderate     67
Recent       65
New          42
Name: count, dtype: int64
```

In [75]:
```
fig,(ax1,ax2)=plt.subplots(1,2,figsize=(12,5))
ax1.bar(count.index,count.values)
ax1.set_xlabel("Price Category")
ax1.set_ylabel("Values")
ax1.set_title("This is First Bar Chart")
ax2.bar(counts.index,counts.values)
ax2.set_xlabel("Age Category")
ax2.set_title("This is Second Bar Chart")
plt.tight_layout()
plt.show()
```



In [76]:
```
df.head()
```

Out[76]:

	Car_ID	Brand	Manufacture_Year	Body_Type	Fuel_Type	Transmission	I
0	CAR_0001	Mercedes	2006	SUV	Petrol	Manual	
1	CAR_0002	Nissan	2023	Coupe	Petrol	Automatic	
2	CAR_0003	Nissan	2007	Hatchback	Diesel	Manual	
3	CAR_0004	Nissan	2013	Coupe	Petrol	Manual	
4	CAR_0005	Hyundai	2009	Hatchback	Hybrid	Automatic	

In [83]:
```
df.groupby(['Brand','Car_Age']).size()
```

Out[83]:
```
Brand   Car_Age
Audi    1          2
        3          1
        6          1
        7          3
        9          5
                  ..
Toyota  16         3
        17         4
        18         2
        19         2
        21         1
Length: 154, dtype: int64
```

In [86]:
```
df.groupby(['Body_Type','Price_USD']).size()
```

Out[86]:
```
Body_Type  Price_USD
Coupe      5407         1
           7376         1
           10700        1
           13058        1
           14652        1
                       ..
Sedan      114015       1
           114361       1
           114695       1
           116822       1
           116860       1
Length: 300, dtype: int64
```

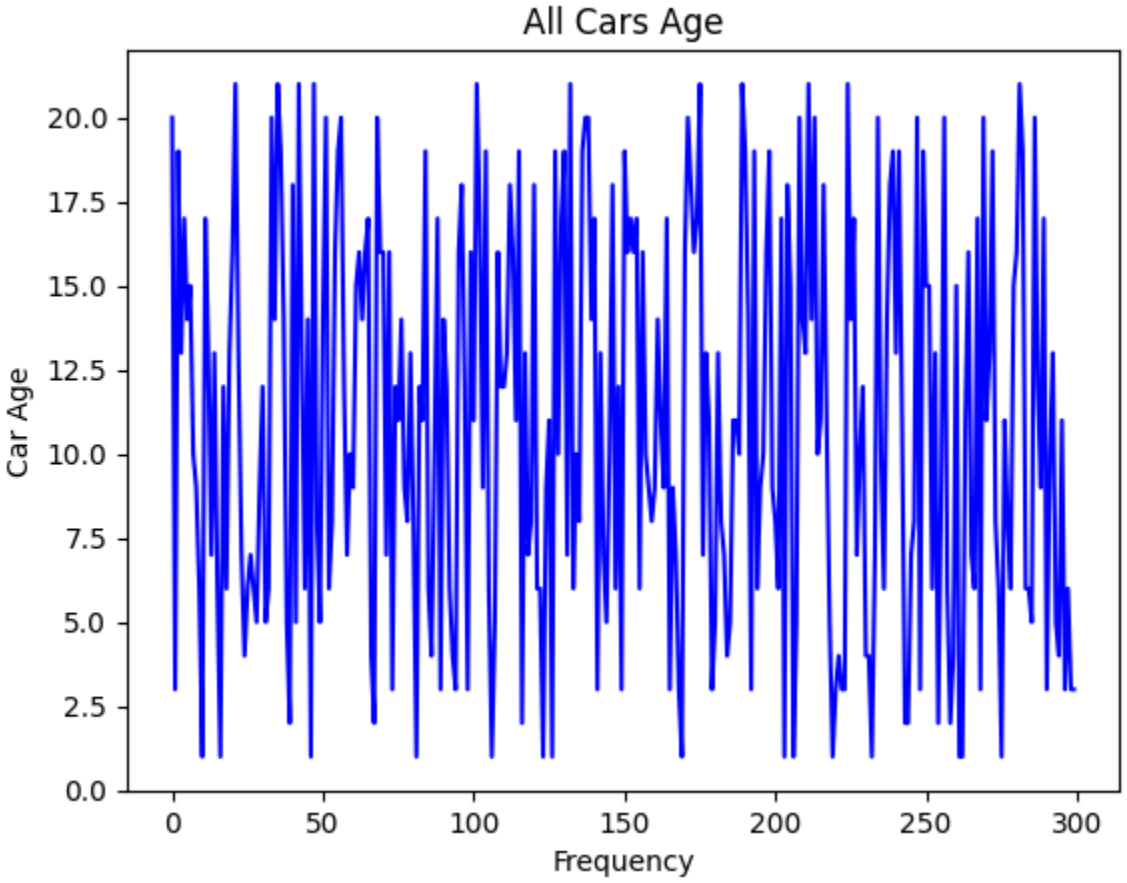
In [92]:
```
df.loc[df["Price_USD"].idxmax()]
```

Out[92]:
```
Car_ID                    CAR_0076
Brand                          Kia
Manufacture_Year              2015
Body_Type                   Pickup
Fuel_Type                   Diesel
Transmission             Automatic
Engine_CC                     2443
Horsepower                     243
Mileage_km_per_l                16
Price_USD                   119587
Manufacturing_Country    South Korea
Car_Age                          11
Price_Category               Luxury
HP_per_CC                    0.0995
Age_Category                Moderate
Efficiency_Score                0.3
Name: 75, dtype: object
```

In [93]:
```
# kia Car is Most Expensive than other Cars
```

In [100…
```
plt.plot(df["Car_Age"],color='blue')
```

```
plt.xlabel("Frequency")
plt.ylabel("Car Age")
plt.title("All Cars Age")
plt.show()
```



In [101…
```
df.loc[df["HP_per_CC"].idxmax()]
```

Out[101…
```
Car_ID                  CAR_0081
Brand                    Toyota
Manufacture_Year           2018
Body_Type                Pickup
Fuel_Type                Hybrid
Transmission          Automatic
Engine_CC                  1020
Horsepower                  536
Mileage_km_per_l             22
Price_USD                 50814
Manufacturing_Country   Germany
Car_Age                       8
Price_Category        Mid-Range
HP_per_CC                0.5255
Age_Category           Moderate
Efficiency_Score            0.6
Name: 80, dtype: object
```

In [ ]:
```
# Toyota have more hp per cc than other cars
```

```
# Made in Germany
```