This is the Car sales data set which includes information about different cars. This data set is being taken from Analytixlabs for the purpose of prediction

My task is to analyze this dataset and provide insights to see which feature has more impact on car sales and carry out the result of this

Dataset Link: https://lnkd.in/d-pMEaMF

The visualization should answer these questions:

1- which is The top Manufacturer in sales

2- which Manufacturer has the most Year Resale Value

3- is the Vehicle Type affect sales?

4- Which feature has more impact on car sales and carry out the result of this

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Upload our Dataset

In [2]:

```
cars_data = pd.read_csv("Car_sales.csv")
cars_data.head()
```

Out[2]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_tho
0	Acura	Integra	16.919	16.360	Passenger	
1	Acura	TL	39.384	19.875	Passenger	
2	Acura	CL	14.114	18.225	Passenger	
3	Acura	RL	8.588	29.725	Passenger	
4	Audi	A4	20.397	22.255	Passenger	

In [3]:

```
cars_data.shape
```

Out[3]:

```
(157, 16)
```

Data Cleaning

First : we sum all the Null Values in our datasets

In [4]:
```
cars_data.isna().sum()
```

Out[4]:
```
Manufacturer          0
Model                 0
Sales_in_thousands    0
__year_resale_value   36
Vehicle_type          0
Price_in_thousands    2
Engine_size           1
Horsepower            1
Wheelbase             1
Width                 1
Length                1
Curb_weight           2
Fuel_capacity         1
Fuel_efficiency       3
Latest_Launch         0
Power_perf_factor     2
dtype: int64
```

Second Handling the Null values in the coulmn "__year_resale_value"

we found out that the coulmn (__year_resale_value) has a lot of missing value , we will solve this problem with calculate the mean and the median of that coulmn

and replacing all the Null value with the median value because it helps to ensure that missing values do not disproportionately influence the statistical analysis of the data and it is less sensitive to outliers.

In [5]:
```
print("mean : ",cars_data['__year_resale_value'].mean())
print("median : ",cars_data['__year_resale_value'].median())
```

```
mean :  18.07297520661157
median :  14.18
```

In [6]:
```
cars_data['__year_resale_value'].fillna(cars_data['__year_resale_value'].median(),i
cars_data.isna().sum()
```

```
Out[6]: Manufacturer           0
        Model                  0
        Sales_in_thousands     0
        __year_resale_value    0
        Vehicle_type           0
        Price_in_thousands     2
        Engine_size            1
        Horsepower             1
        Wheelbase              1
        Width                  1
        Length                 1
        Curb_weight            2
        Fuel_capacity          1
        Fuel_efficiency        3
        Latest_Launch          0
        Power_perf_factor      2
        dtype: int64
```

Third Hanlding the other Null value

1-threshold is calculated as 5% of the total number of rows and it's the maximum number of missing values that a column can have before it is dropped from the DataFrame

2-selects the columns where the number of missing values is less than or equal to the threshold value.

3-drops all rows that have missing values in the selected columns

```
In [7]: threhold = len(cars_data) * 0.05
        print("threhold : ",threhold)
        cols_to_drop = cars_data.columns[cars_data.isna().sum() <= threhold ]
        cars_data.dropna(subset=cols_to_drop,inplace=True)
        cars_data.isna().sum()
```

```
threhold :  7.8500000000000005
```

```
Out[7]: Manufacturer           0
        Model                  0
        Sales_in_thousands     0
        __year_resale_value    0
        Vehicle_type           0
        Price_in_thousands     0
        Engine_size            0
        Horsepower             0
        Wheelbase              0
        Width                  0
        Length                 0
        Curb_weight            0
        Fuel_capacity          0
        Fuel_efficiency        0
        Latest_Launch          0
        Power_perf_factor      0
        dtype: int64
```

generates a statistical summary of our Datasets

1-Provides a quick overview of the distribution of the data in each column

2-help you quickly understand the range and distribution of values in each column

In [8]:
```
cars_data.describe()
```

Out[8]:

	Sales_in_thousands	__year_resale_value	Price_in_thousands	Engine_size	Horsepowe
count	152.000000	152.000000	152.000000	152.000000	152.00000
mean	53.359072	17.144671	27.331822	3.049342	184.80921
std	68.938380	10.301344	14.418669	1.049818	56.82315
min	0.110000	5.160000	9.235000	1.000000	55.00000
25%	13.714000	12.527500	17.888750	2.300000	147.50000
50%	29.213000	14.180000	22.747000	3.000000	175.00000
75%	68.069750	17.806250	31.938750	3.575000	211.25000
max	540.561000	67.550000	85.500000	8.000000	450.00000

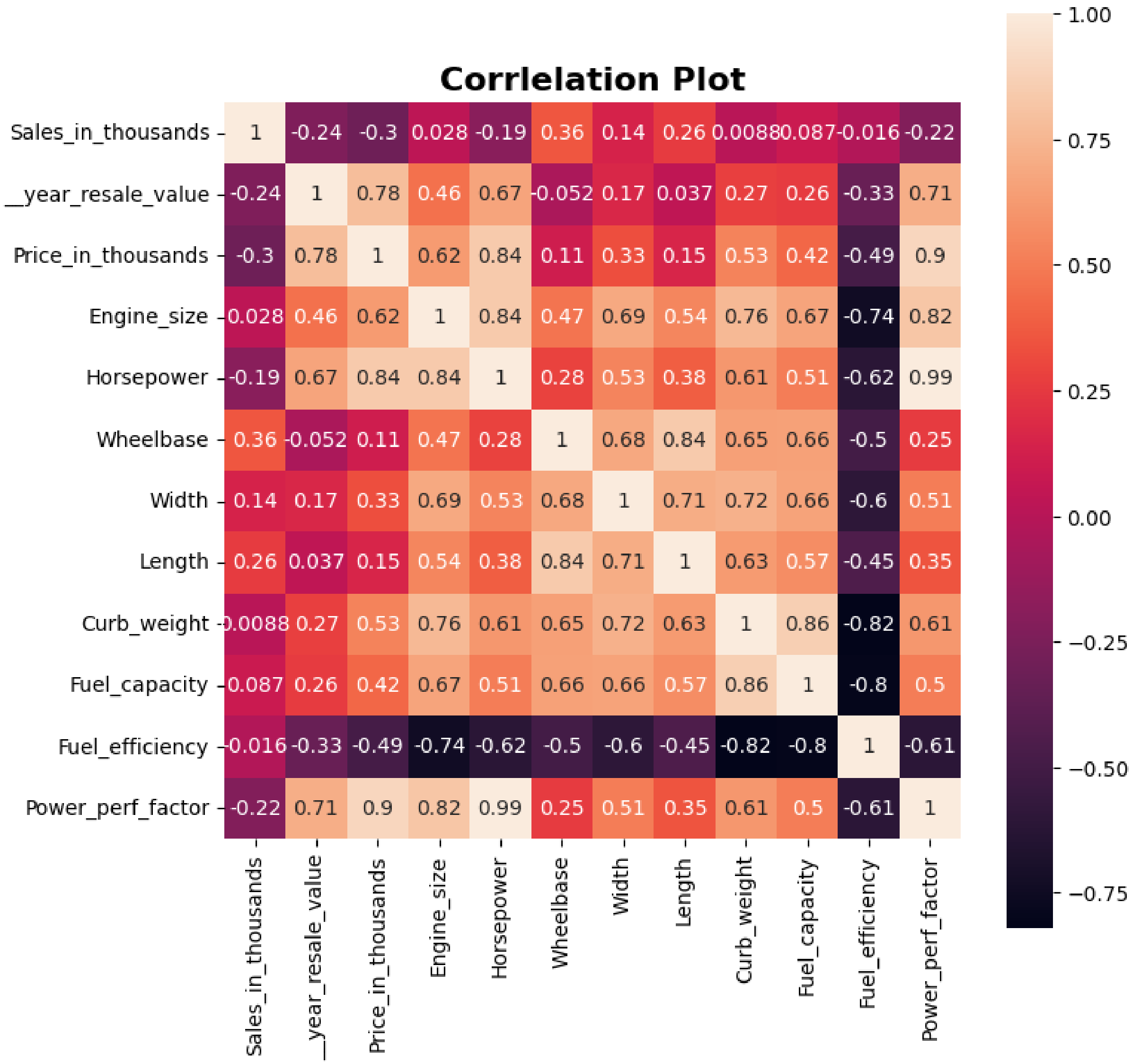
Finding Realtionship in data

calculates the correlation matrix between all pairs of numeric becuase it's provide a quick and easy way to explore the relationships between variables in the dataset

Dark colors in the heatmap indicate strong negative correlations

Light colors in the heatmap indicate strong Positive correlations

In [9]:
```
corr = cars_data.corr(numeric_only=True)
plt.figure(figsize=(8,8))
sns.heatmap(corr,annot=True,square=True)
plt.title("Corrlelation Plot",fontsize=16,fontweight="bold")
plt.show()
```

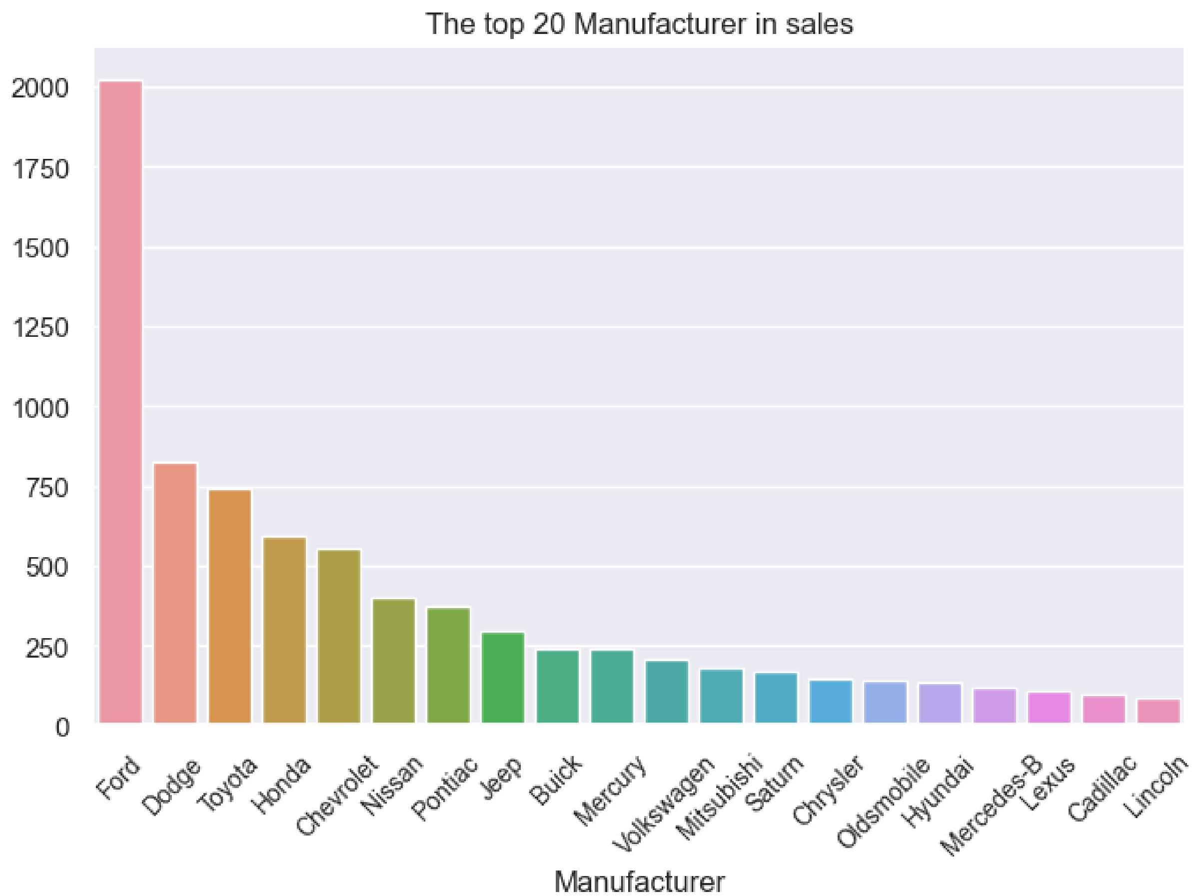


Analysing The Manufacturer's sales

This code generates a bar plot that shows the total sales (in thousands) for the top 20 car manufacturers in our Dataset

we understand from this Visualization that ford has the highest sales

In [58]:
```
order = cars_data.groupby("Manufacturer")["Sales_in_thousands"].sum()
order = order.nlargest(20)
plt.figure(figsize=(8, 5))
sns.set_theme(style="darkgrid")
sns.barplot(y=order.values, x=order.index)
plt.title('The top 20 Manufacturer in sales')
plt.xticks(fontsize=10, rotation=45)
plt.show()
```

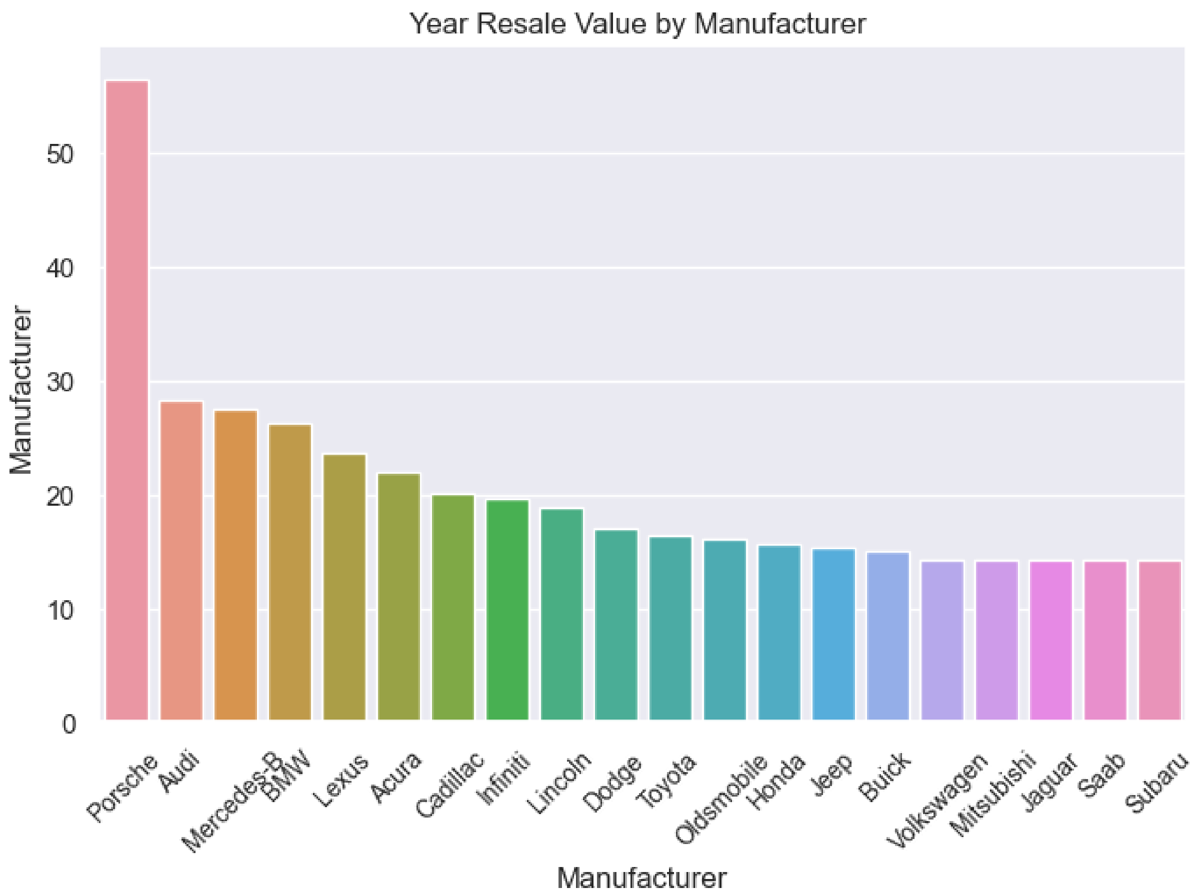


Year Resale Value by Manufacturer

code generates a bar plot that shows the average year resale value for the top 20 car manufacturers in our DataSet

we found that Porsche has highest average year resale values and produce cars with high resale values.

In [62]:
```
order1 = cars_data.groupby("Manufacturer")["__year_resale_value"].mean()
order1 = order1.nlargest(20)
plt.figure(figsize=(8, 5))
sns.barplot(y=order1.values, x=order1.index)
plt.title('Year Resale Value by Manufacturer ')
plt.xticks(fontsize=10, rotation=45)
plt.ylabel("Manufacturer")
plt.show()
```

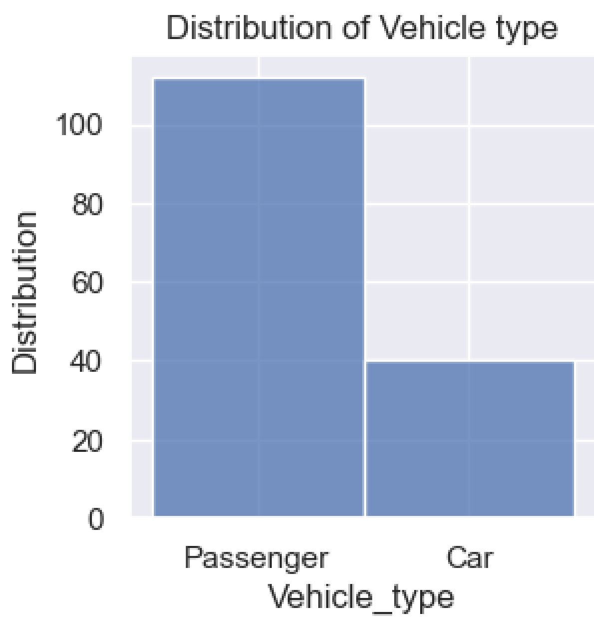


Analysing the Vehicle type

generates a histogram plot that shows the distribution of the "Vehicle_type" column in our DataSet

we found that passenger type is the most common and becoming more popular over time

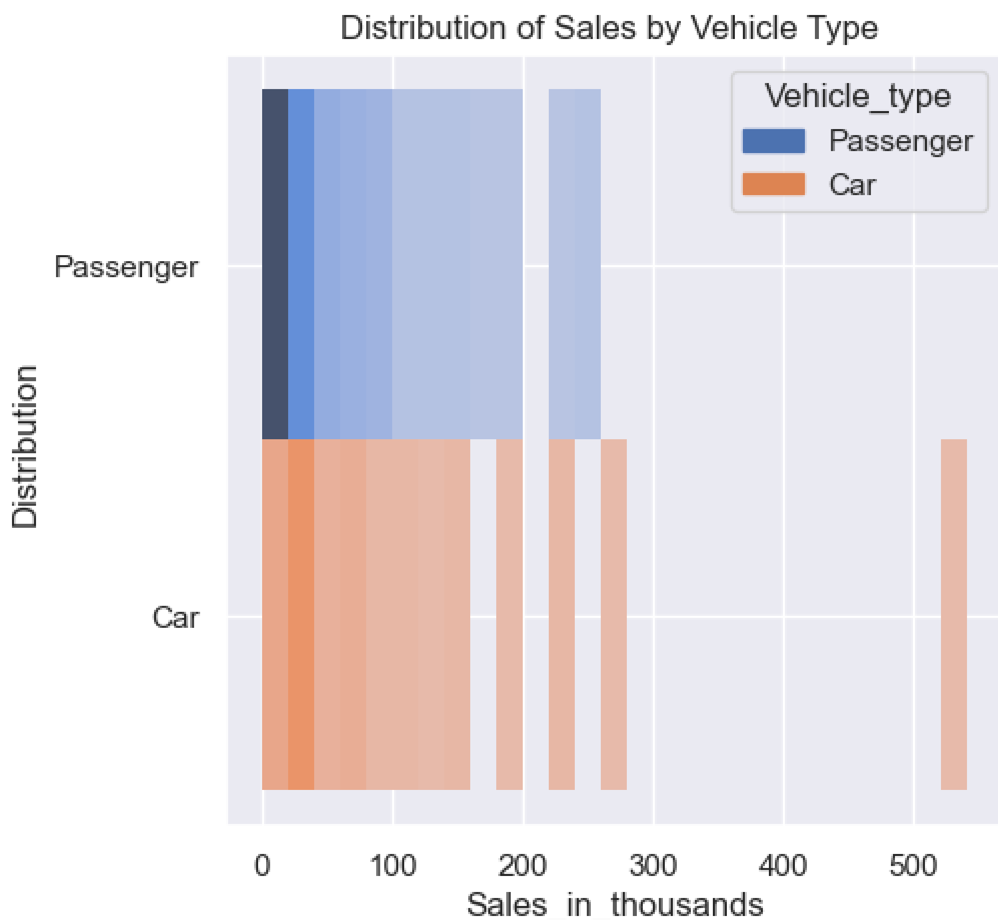
In [12]:
```
plt.figure(figsize=(3, 3))
sns.histplot(data=cars_data, x="Vehicle_type")
plt.title('Distribution of Vehicle type')
plt.ylabel("Distribution")
plt.show()
```



this code generates a histogram plot that shows the distribution of sales (in thousands) by vehicle type in our Dataset

we found that Car type has the highest sales

In [13]:
```
plt.figure(figsize=(5, 5))
sns.histplot(data=cars_data, x="Sales_in_thousands",y="Vehicle_type",hue="Vehicle_t
plt.title('Distribution of Sales by Vehicle Type ')
plt.ylabel("Distribution")
plt.show()
```



finding the feature That has more impact on car sales

Distribution Analysis

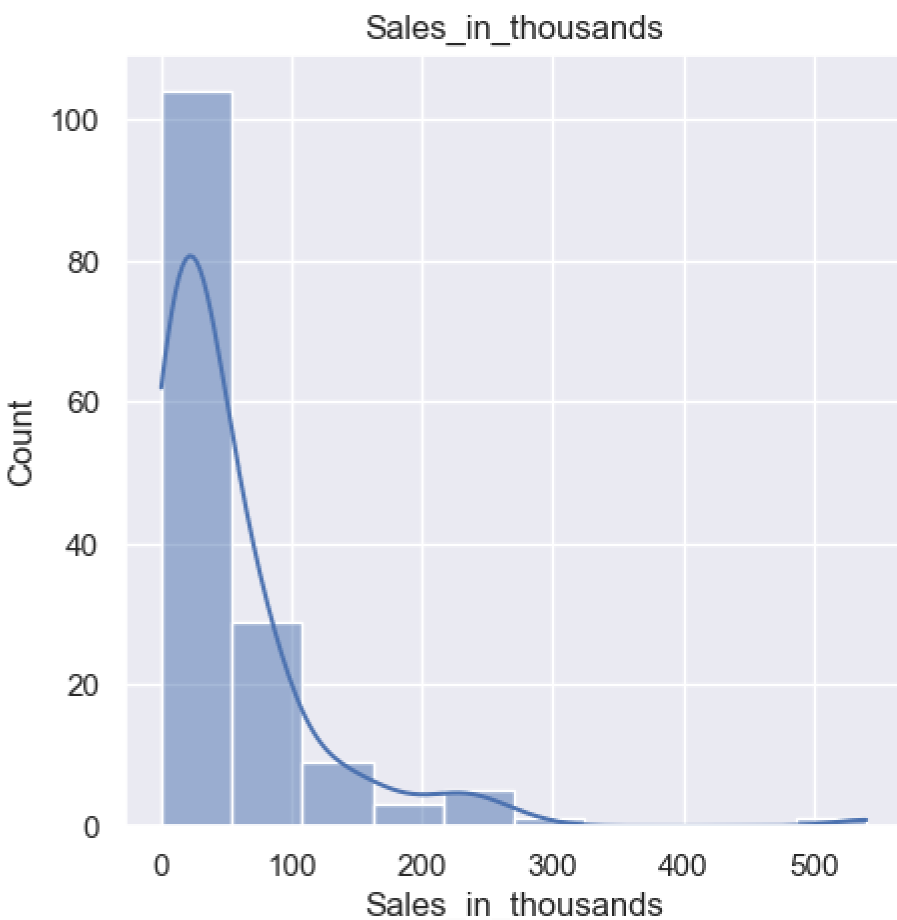
This code selects all columns in the cars_data DataFrame that have a data type of either float64 or int64, and then drops the "__year_resale_value" and "Price_in_thousands" columns from the resulting DataFrame.

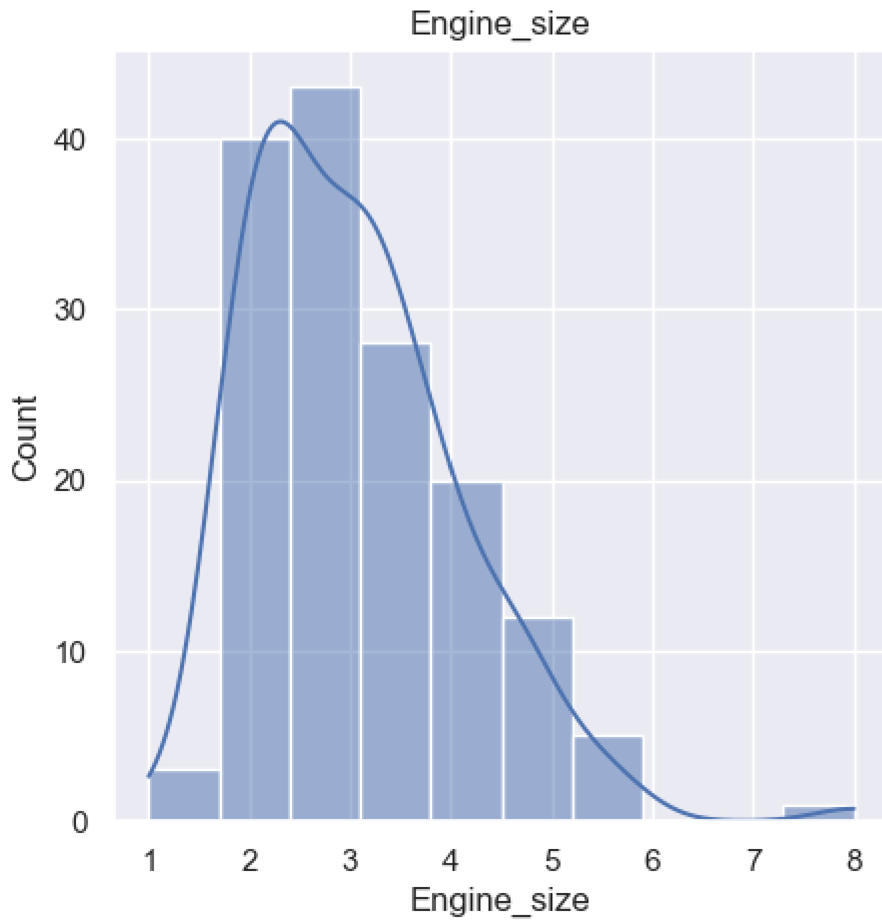
In [14]:
```
df_num = cars_data.select_dtypes(include = ['float64', 'int64'])
print(df_num.columns)
df_num = df_num.drop(columns=["__year_resale_value","Price_in_thousands"])
print(df_num.columns)
```

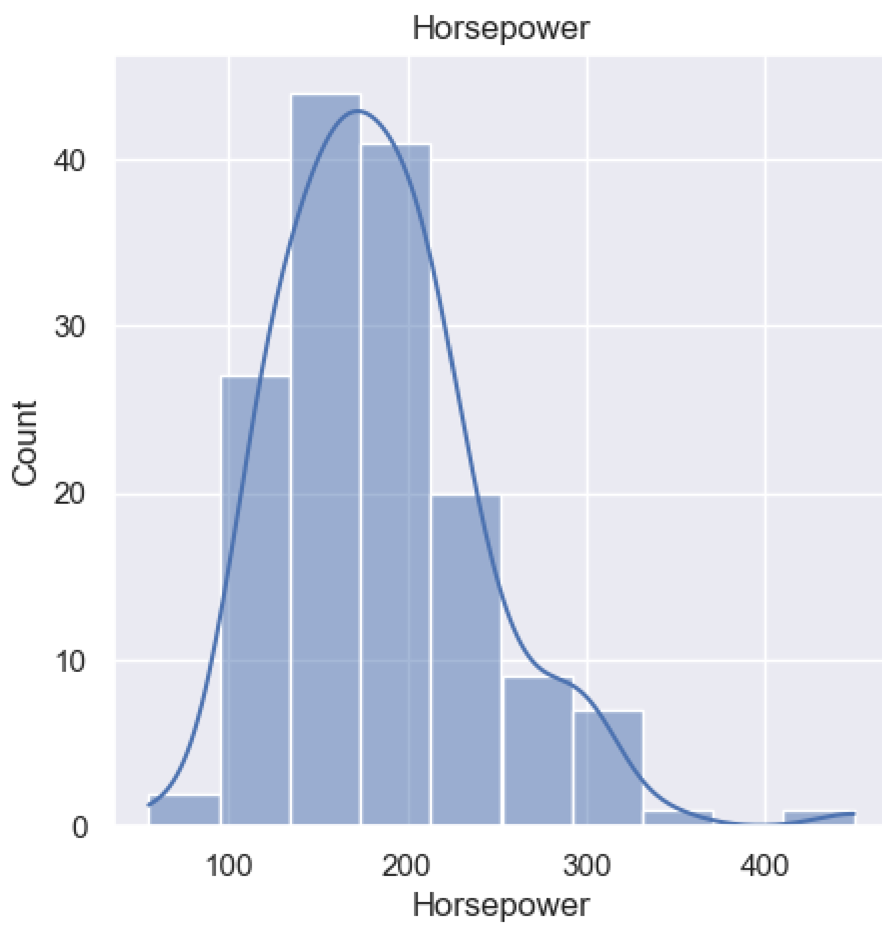
```
Index(['Sales_in_thousands', '__year_resale_value', 'Price_in_thousands',
       'Engine_size', 'Horsepower', 'Wheelbase', 'Width', 'Length',
       'Curb_weight', 'Fuel_capacity', 'Fuel_efficiency', 'Power_perf_factor'],
      dtype='object')
Index(['Sales_in_thousands', 'Engine_size', 'Horsepower', 'Wheelbase', 'Width',
       'Length', 'Curb_weight', 'Fuel_capacity', 'Fuel_efficiency',
       'Power_perf_factor'],
      dtype='object')
```

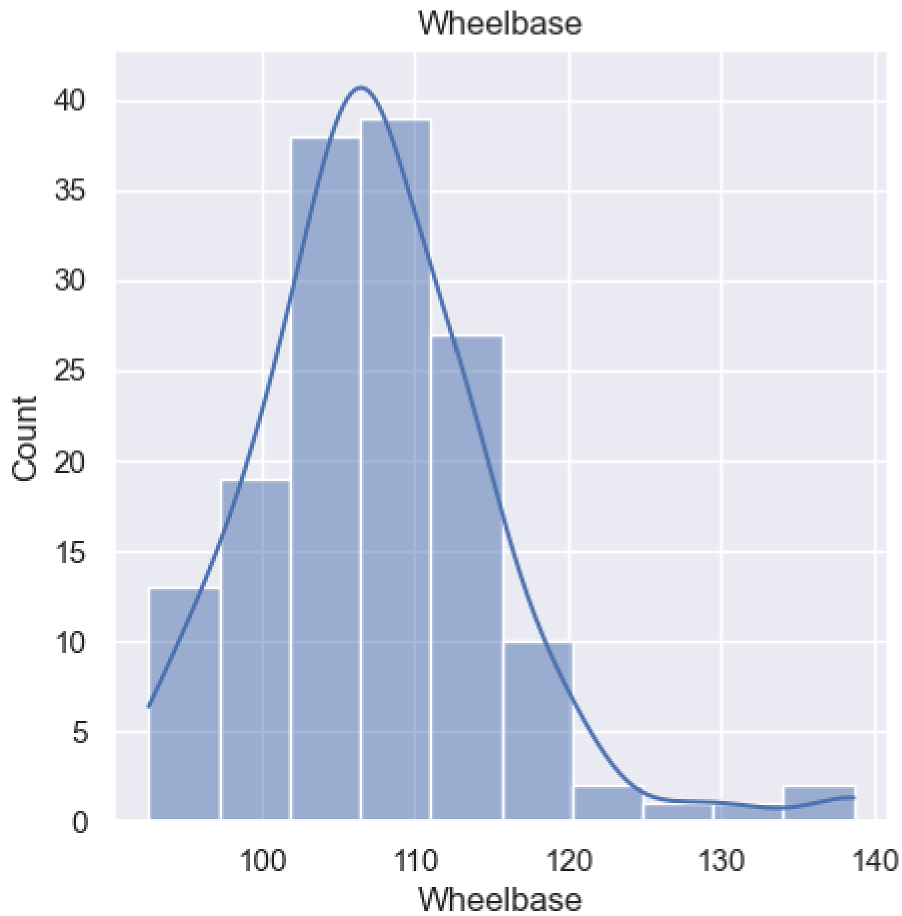
This code defines a function called plot() that creates a histogram plot for each column in the df_num DataFrame

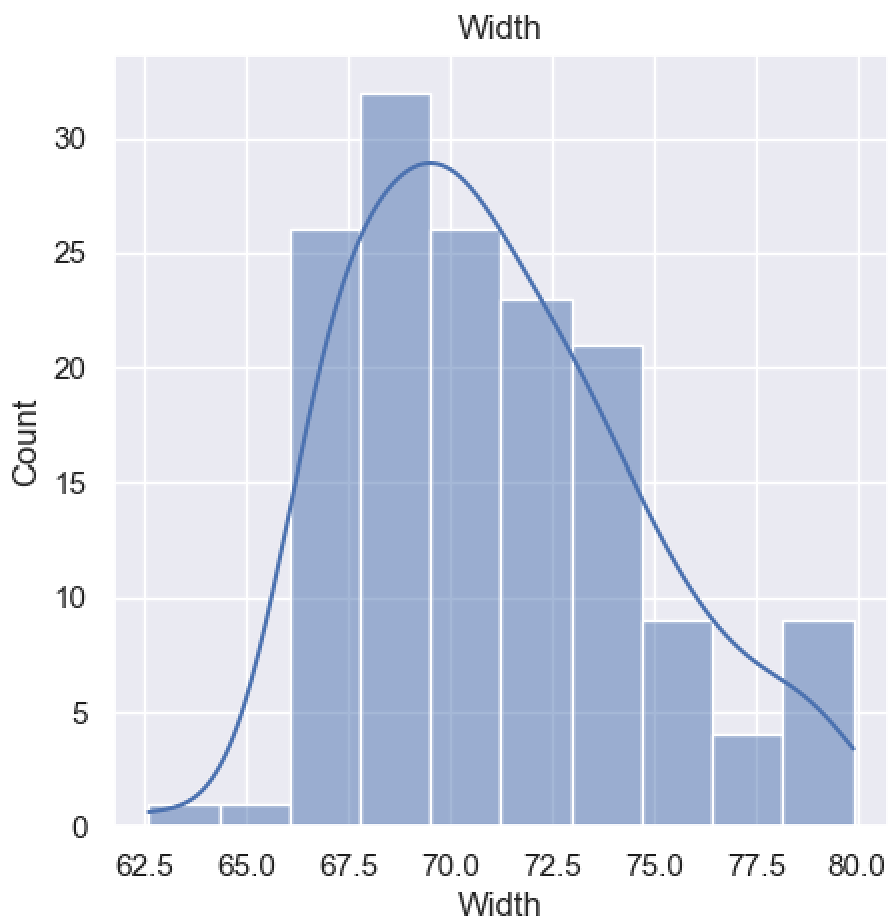
In [15]:
```
def plot():
    for p in df_num.columns:
        plt.figure(figsize=(5, 5))
        sns.histplot(df_num[p], bins=10,kde=True)
        plt.title(p)
        plt.show()
plot()
```

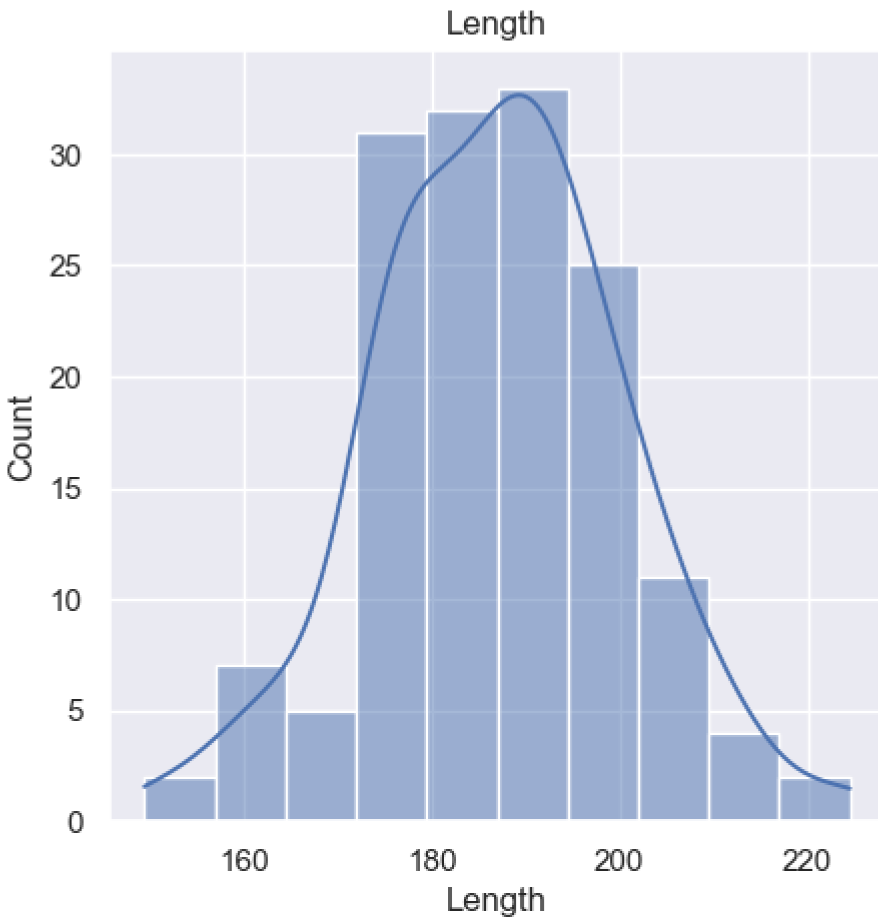


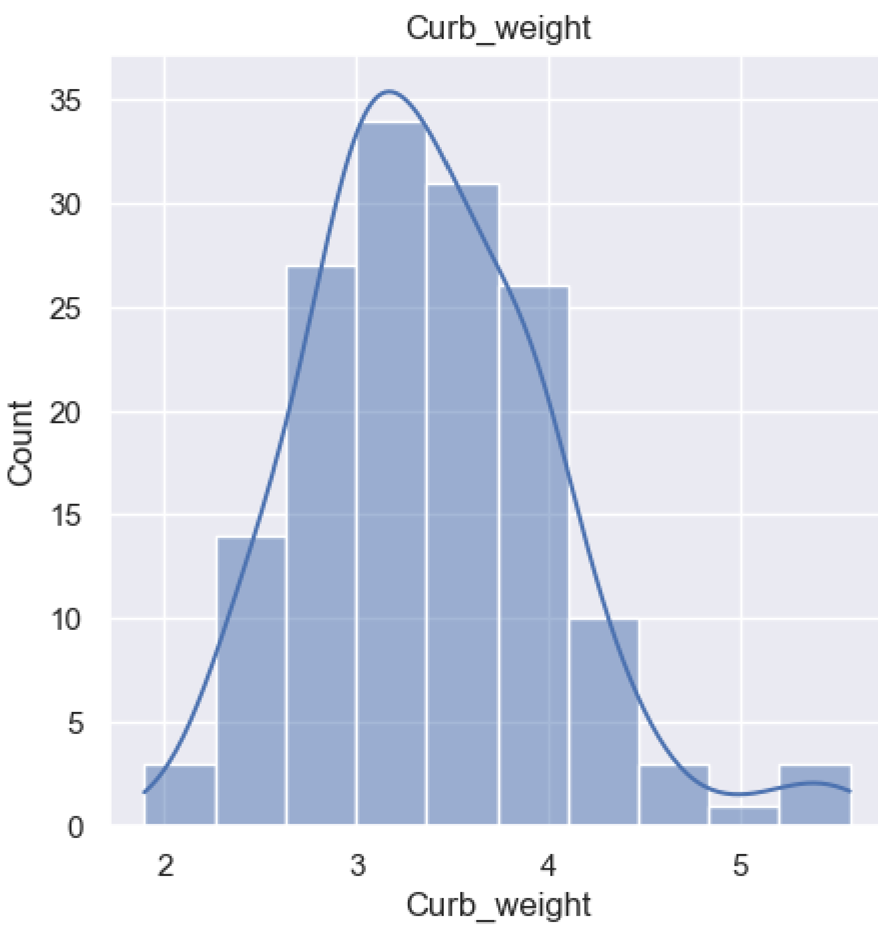

Engine_size
40
30
20
10
0
Count
1
2
3
4
5
6
7
8
Engine_size

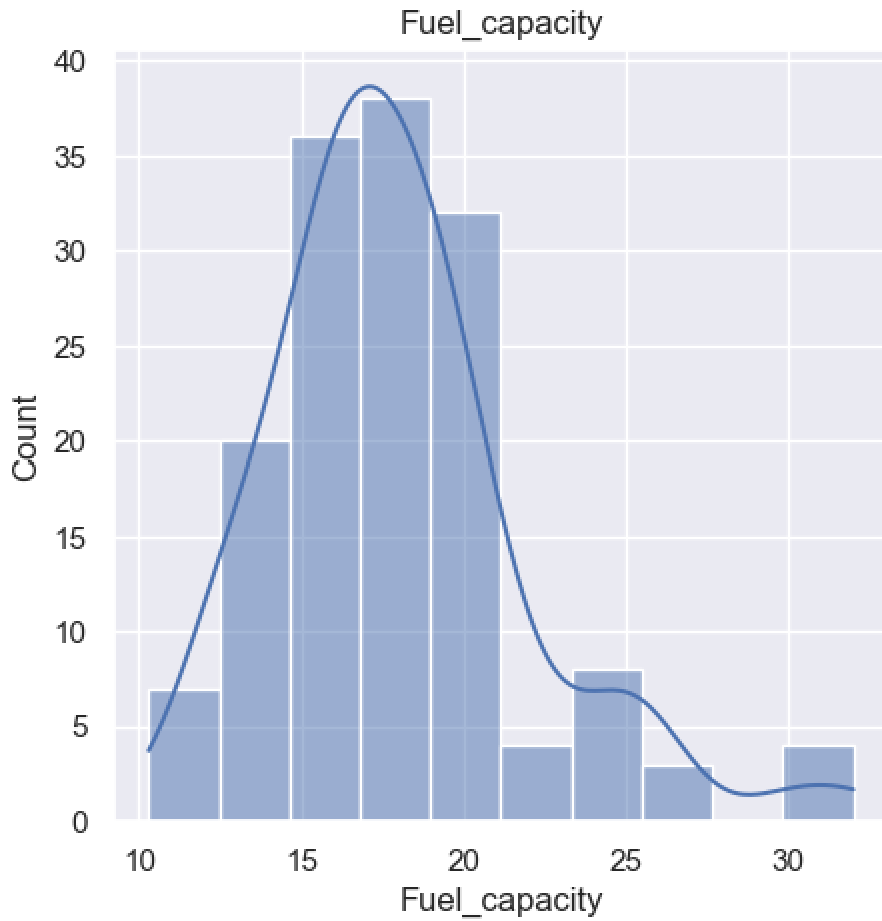

Horsepower
40
30
20
10
0
Count
100
200
300
400
Horsepower

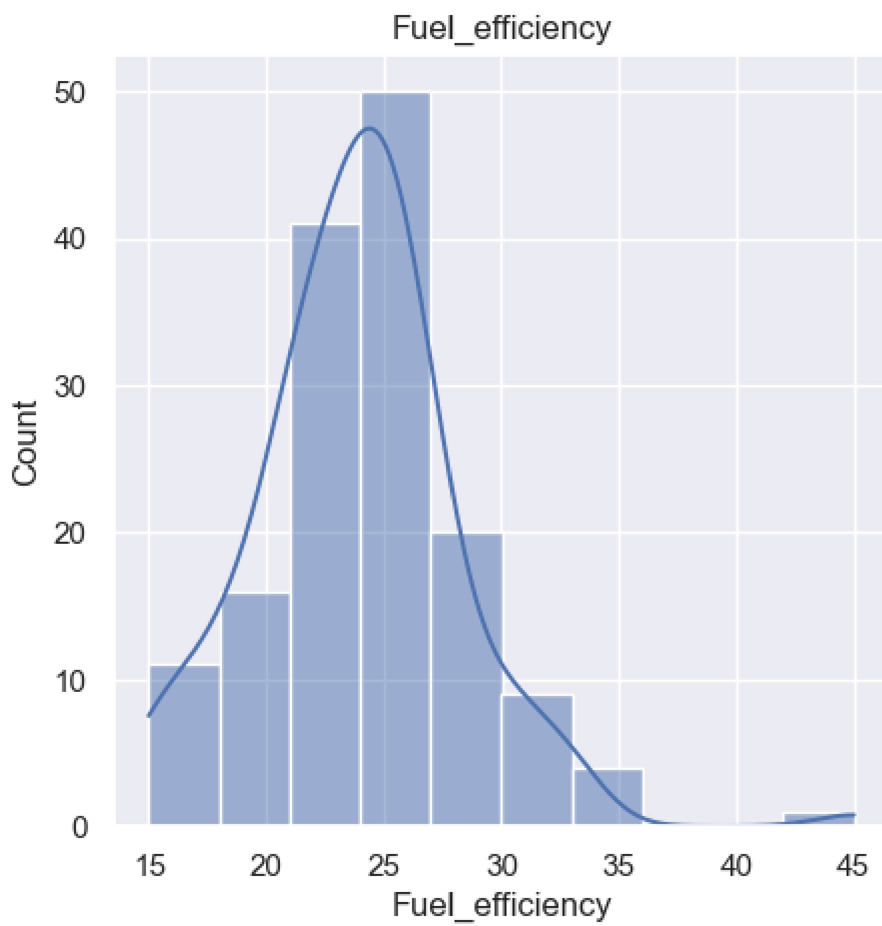

Wheelbase
Count
Wheelbase
0
5
10
15
20
25
30
35
40
100
110
120
130
140

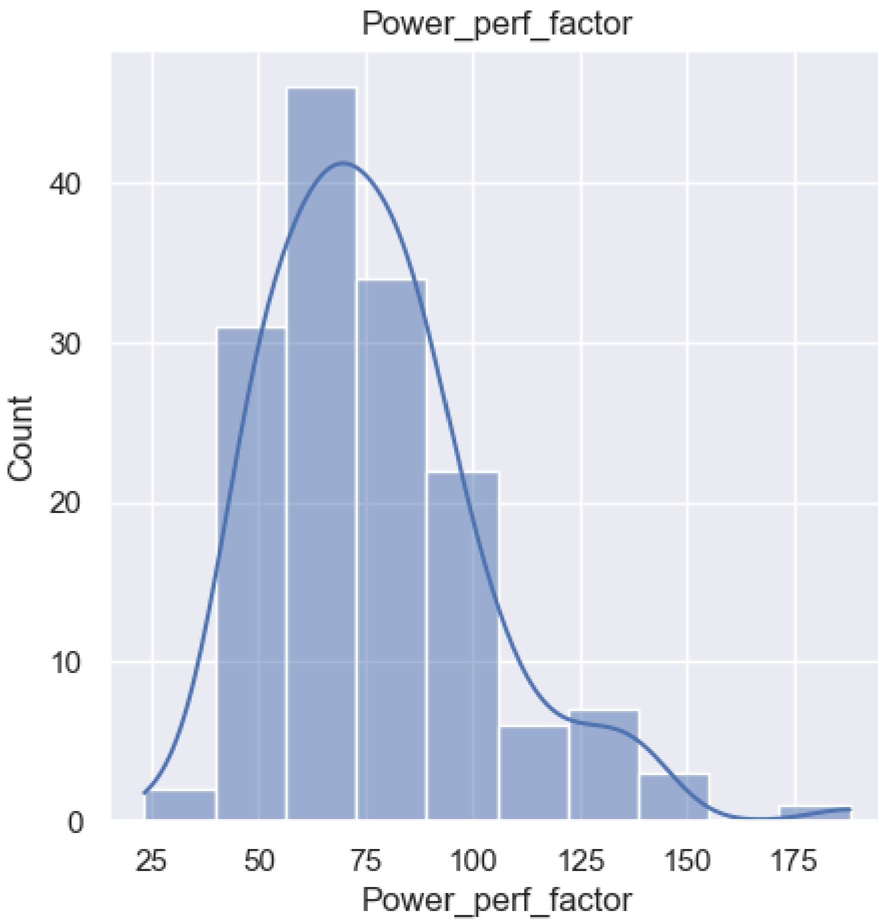

Width
Count
Width
0
5
10
15
20
25
30
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0


Length
Count
0
5
10
15
20
25
30
160
180
200
220
Length


Curb_weight
Count
0
5
10
15
20
25
30
35
2
3
4
5
Curb_weight


Fuel_capacity
Count
Fuel_capacity
0
5
10
15
20
25
30
35
40
10
15
20
25
30


Fuel_efficiency
Count
Fuel_efficiency
0
10
20
30
40
50
15
20
25
30
35
40
45



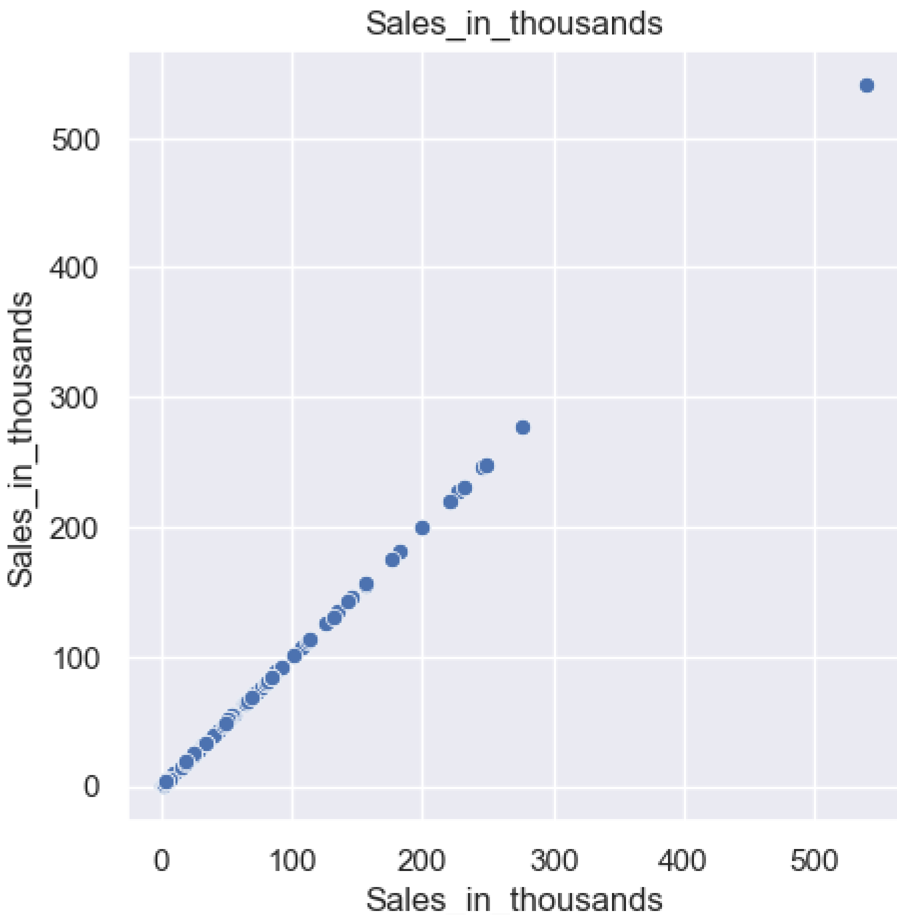
Bivariate Analysis

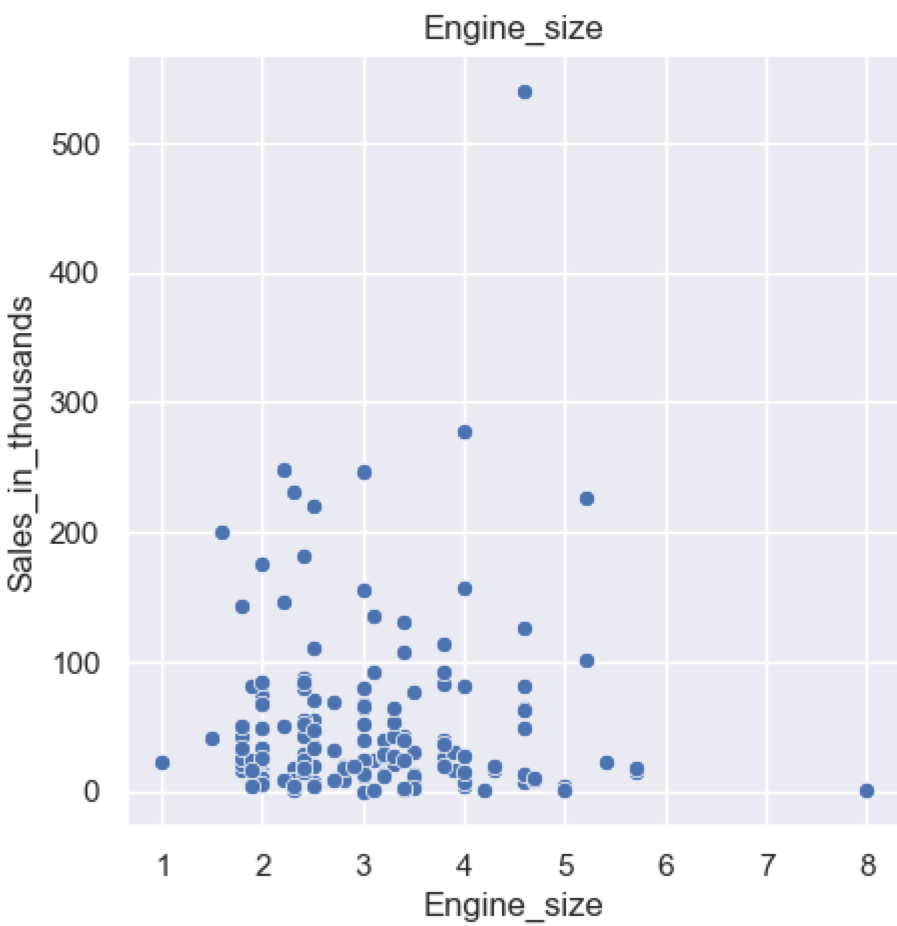
This code defines a function called Biplot() that creates a scatter plot for each column in the df_num DataFrame against the "Sales_in_thousands" column.

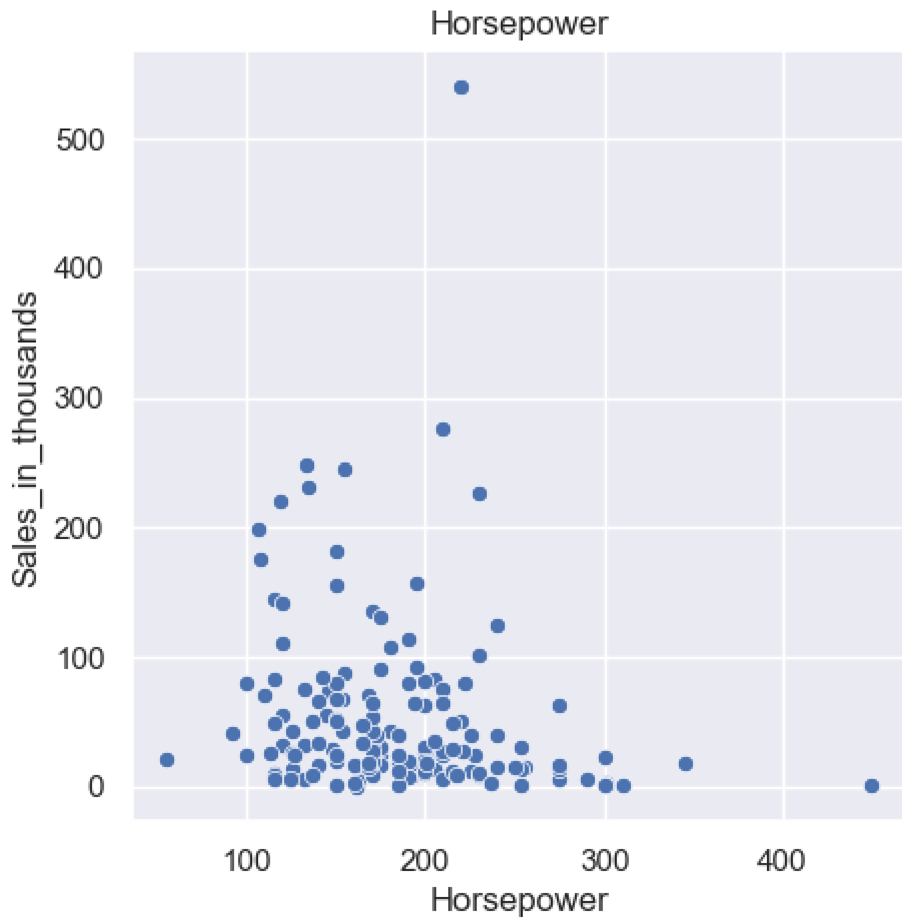
you can see the strength and direction of the relationship between each variable and sales

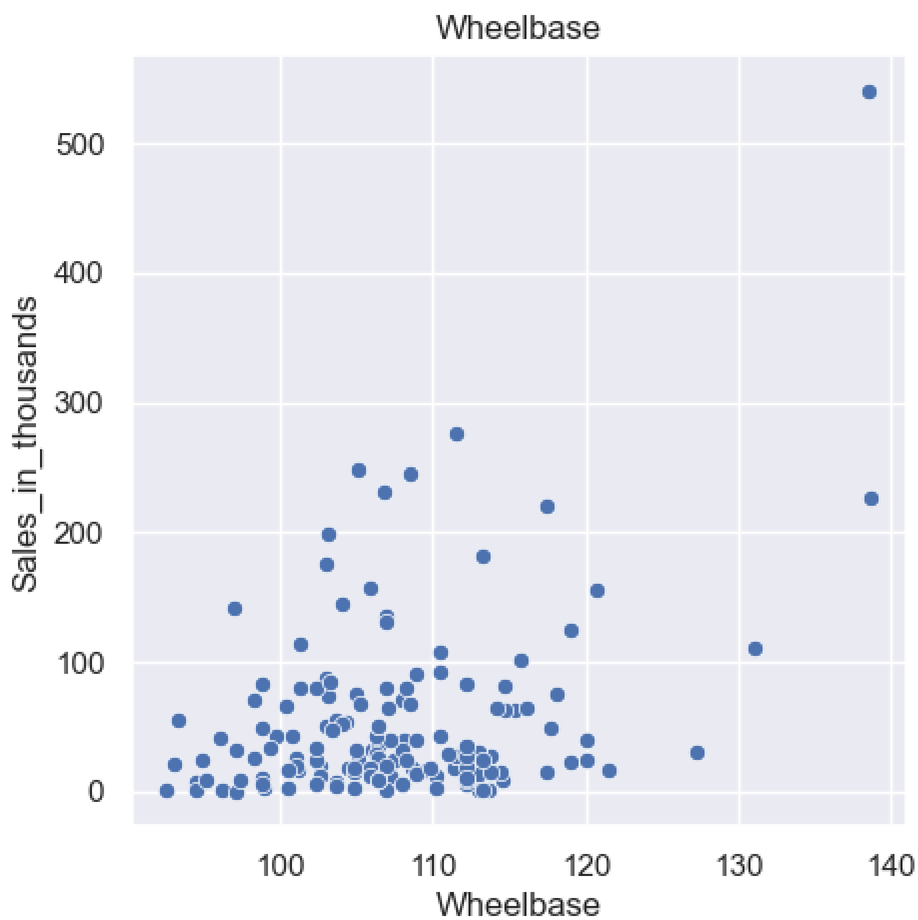
can be useful in understanding the factors that influence sales and identifying potential predictors for modeling.

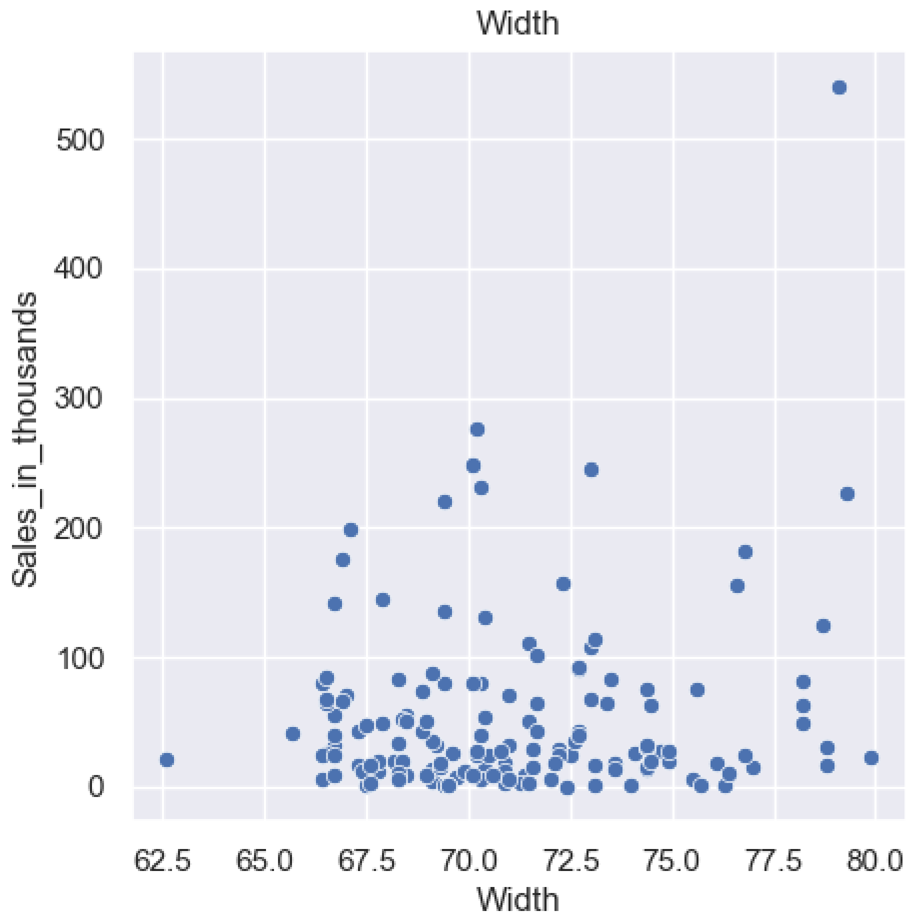
In [16]:
```
def Biplot():
    for u in df_num.columns:
        plt.figure(figsize=(5, 5))
        sns.scatterplot(x = u , y="Sales_in_thousands" , data = df_num )
        plt.title(u)
        plt.show()
Biplot()
```

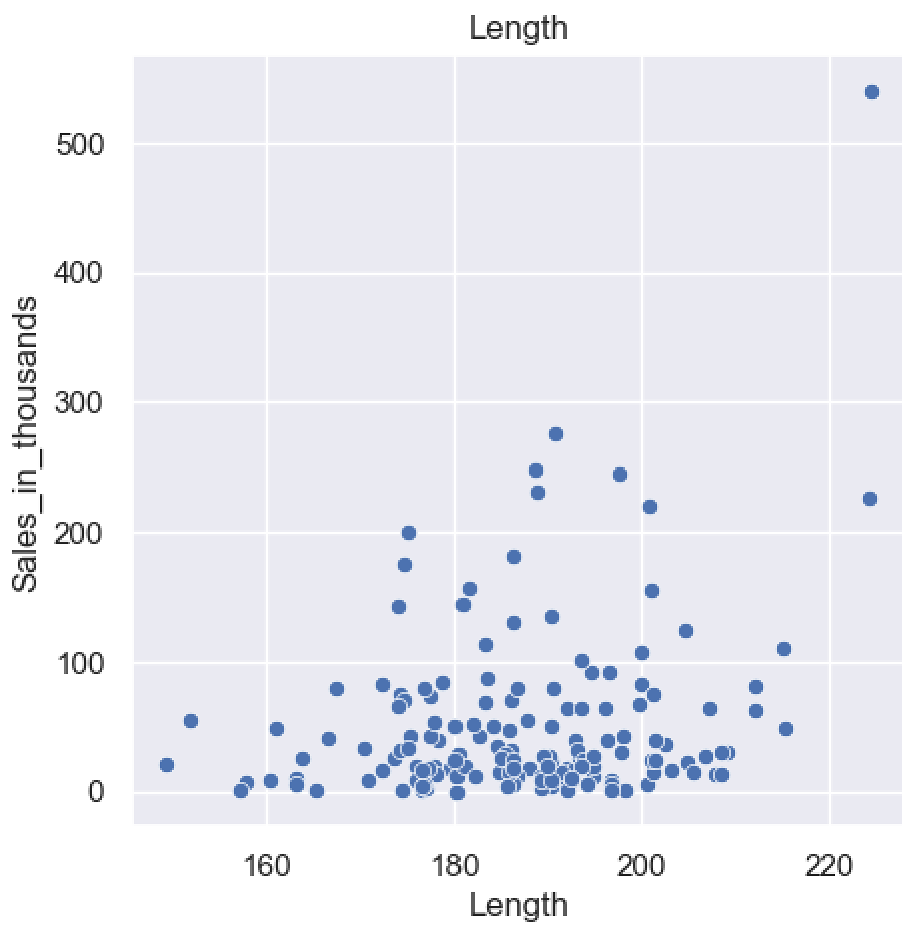

Sales_in_thousands
Sales_in_thousands
0
100
200
300
400
500
Sales_in_thousands
0
100
200
300
400
500

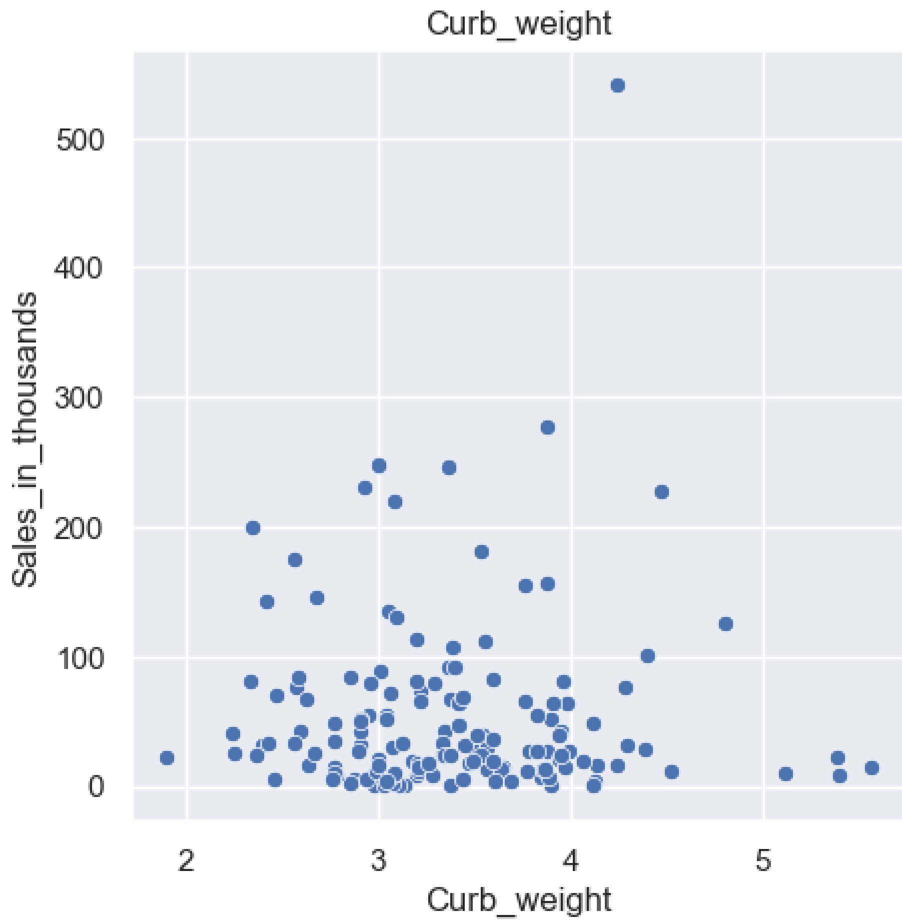

Engine_size
Sales_in_thousands
0
100
200
300
400
500
Engine_size
1
2
3
4
5
6
7
8

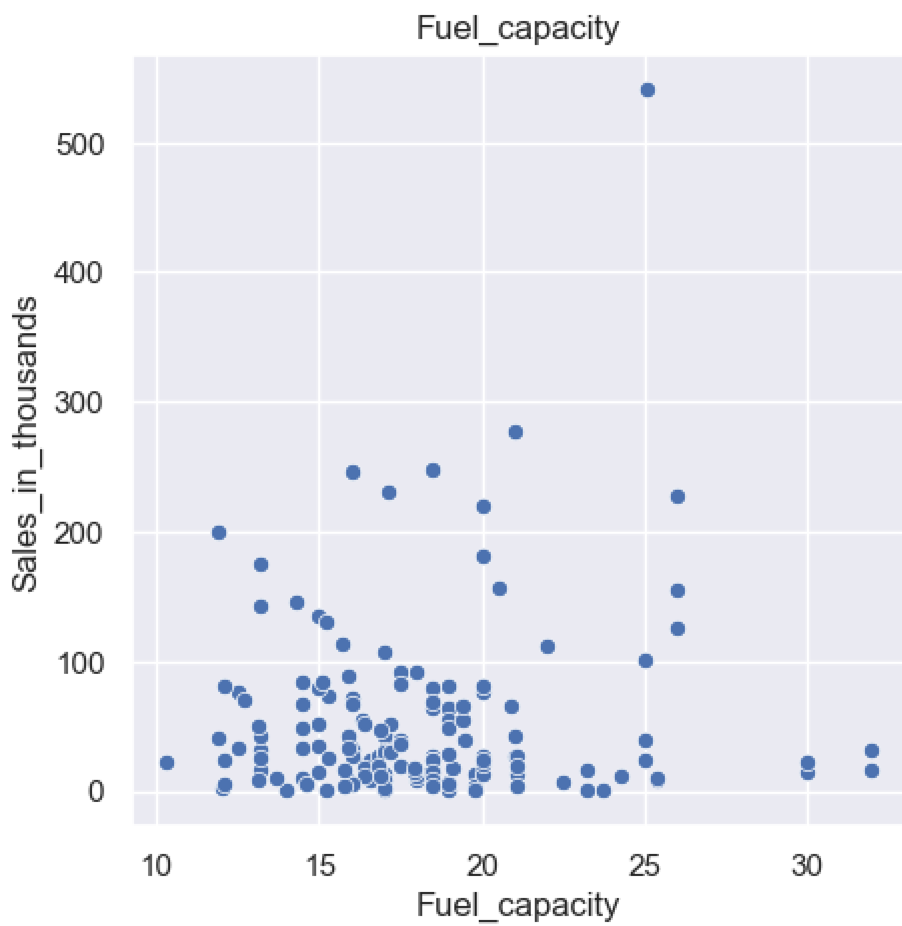

Horsepower
Sales_in_thousands
0
100
200
300
400
500
100
200
300
400
Horsepower

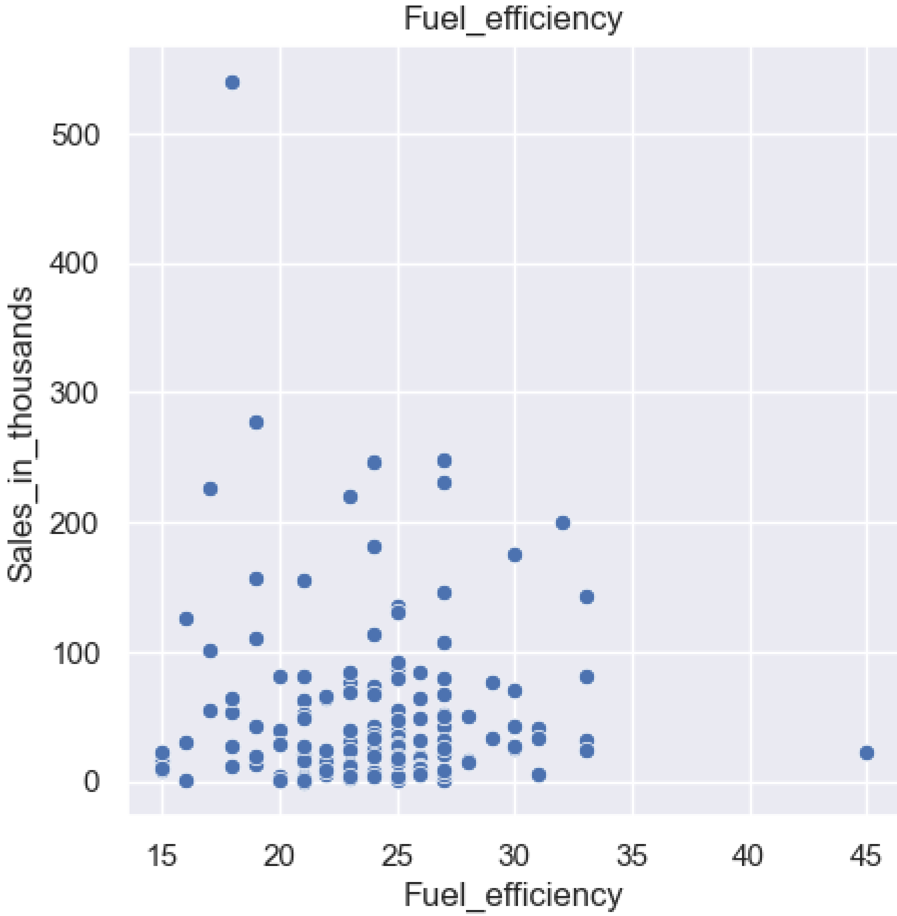

Wheelbase
Sales_in_thousands
0
100
200
300
400
500
100
110
120
130
140
Wheelbase

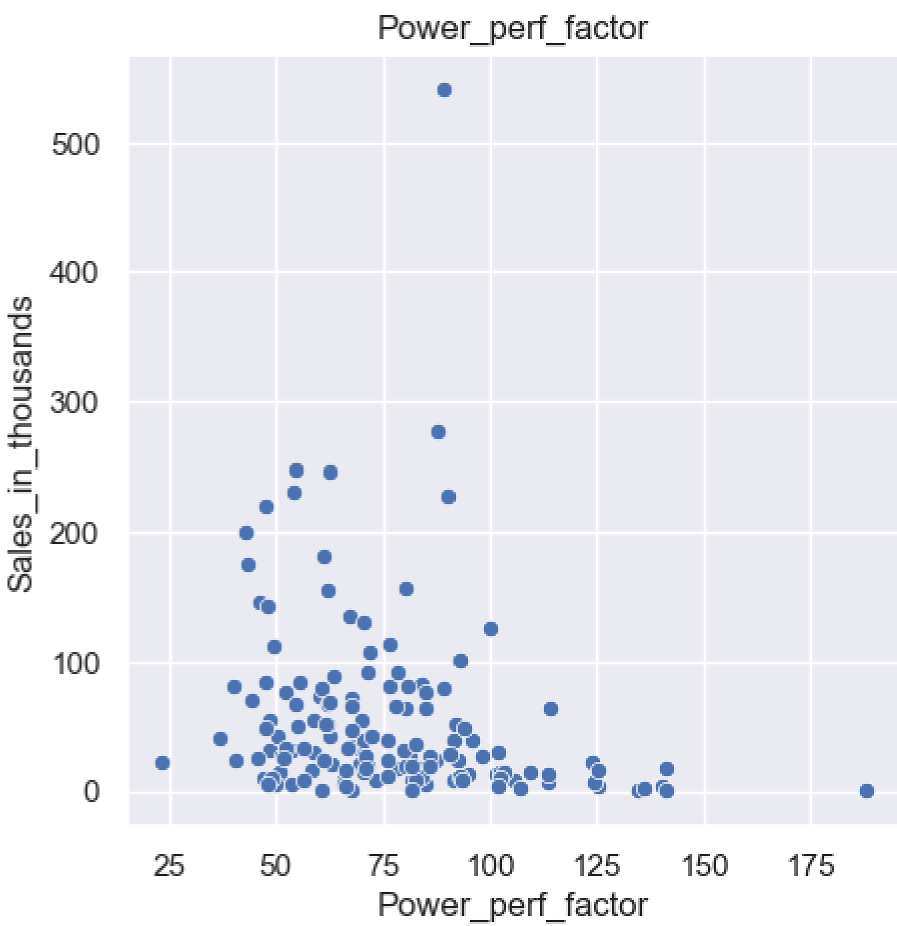

Width
Sales_in_thousands
0
100
200
300
400
500
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0
Width


Length
Sales_in_thousands
0
100
200
300
400
500
160
180
200
220
Length


Curb_weight
Sales_in_thousands
0
100
200
300
400
500
2
3
4
5
Curb_weight


Fuel_capacity
Sales_in_thousands
0
100
200
300
400
500
10
15
20
25
30
Fuel_capacity


Fuel_efficiency
Sales_in_thousands
0
100
200
300
400
500
15
20
25
30
35
40
45
Fuel_efficiency


Power_perf_factor
Sales_in_thousands
0
100
200
300
400
500
25
50
75
100
125
150
175
Power_perf_factor