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

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
```

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
```

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
```

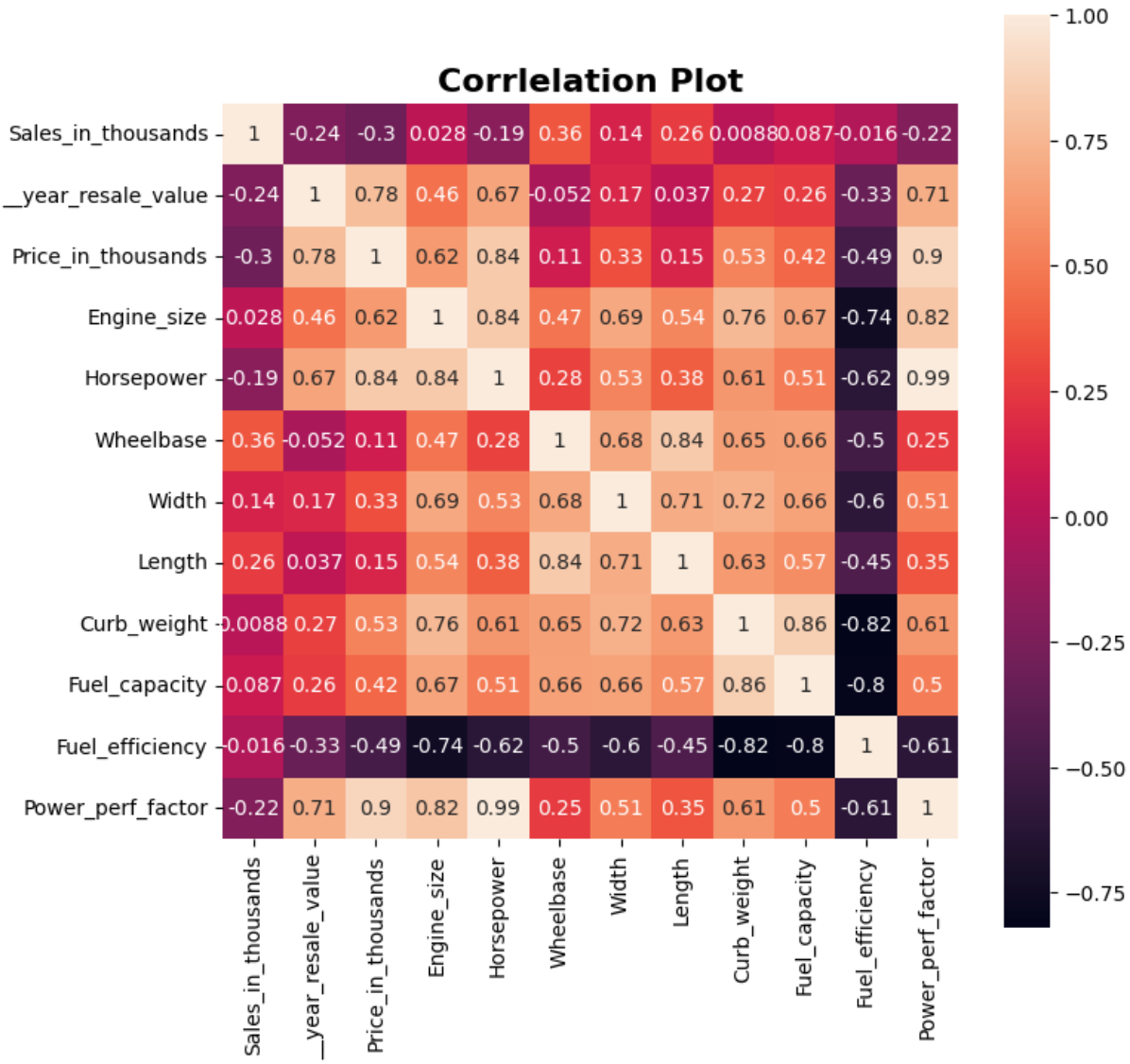
In [8]:
```
cars_data.describe()
```

Out[8]:

	Sales_in_thousands	__year_resale_value	Price_in_thousands	Engine_size	Horsepowe
count	152.000000	152.000000	152.000000	152.000000	152.000000
mean	53.359072	17.144671	27.331822	3.049342	184.809211
std	68.938380	10.301344	14.418669	1.049818	56.823152
min	0.110000	5.160000	9.235000	1.000000	55.000000
25%	13.714000	12.527500	17.888750	2.300000	147.500000
50%	29.213000	14.180000	22.747000	3.000000	175.000000
75%	68.069750	17.806250	31.938750	3.575000	211.250000
max	540.561000	67.550000	85.500000	8.000000	450.000000

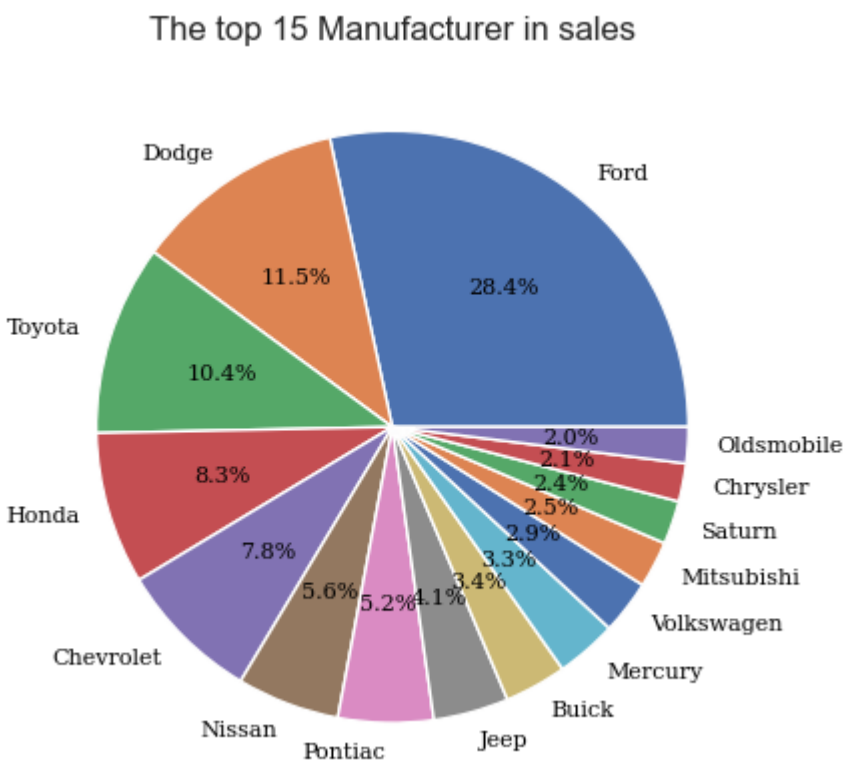
Finding Realtionship in data

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



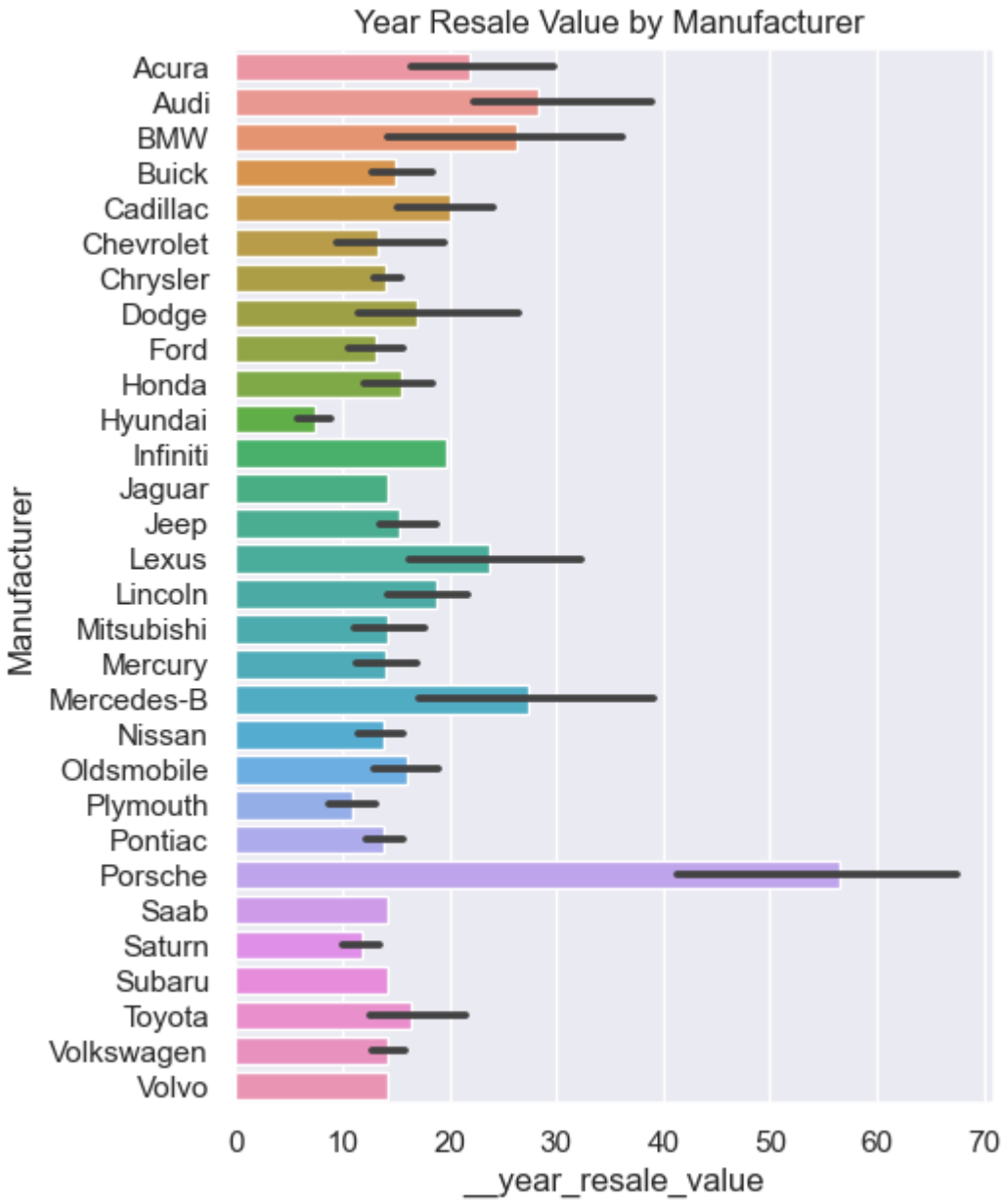
Analysing The Manufacturer's sales

In [10]:
```
man_sales = cars_data.groupby('Manufacturer')['Sales_in_thousands'].sum().nlargest(
font = {'family': 'serif',
        'color':  'black',
        'weight': 'normal',
        'size': 8,
        'rotation' : 0,
        }
sns.set_theme(style="darkgrid")
plt.pie(man_sales,labels=man_sales.index,autopct='%1.1f%%',textprops=font)
plt.title('The top 15 Manufacturer in sales')
plt.show()
```



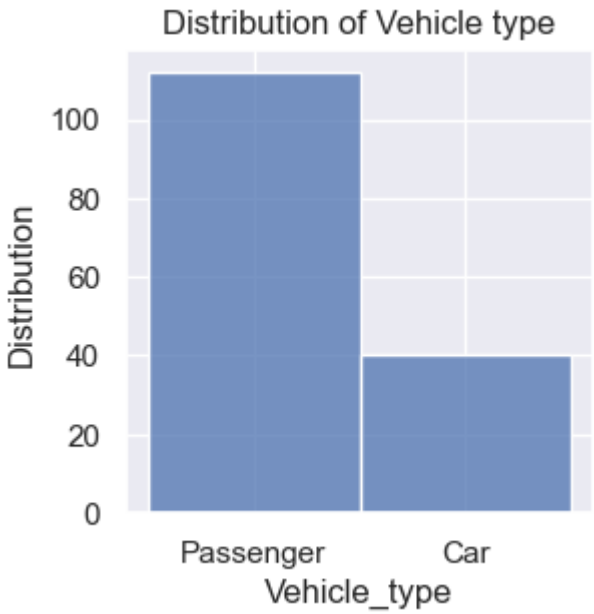
Year Resale Value by Manufacturer

In [11]:
```
plt.figure(figsize=(5, 7))
sns.barplot(data=cars_data, x="__year_resale_value",y="Manufacturer")
plt.title('Year Resale Value by Manufacturer ')
plt.ylabel("Manufacturer")
plt.show()
```

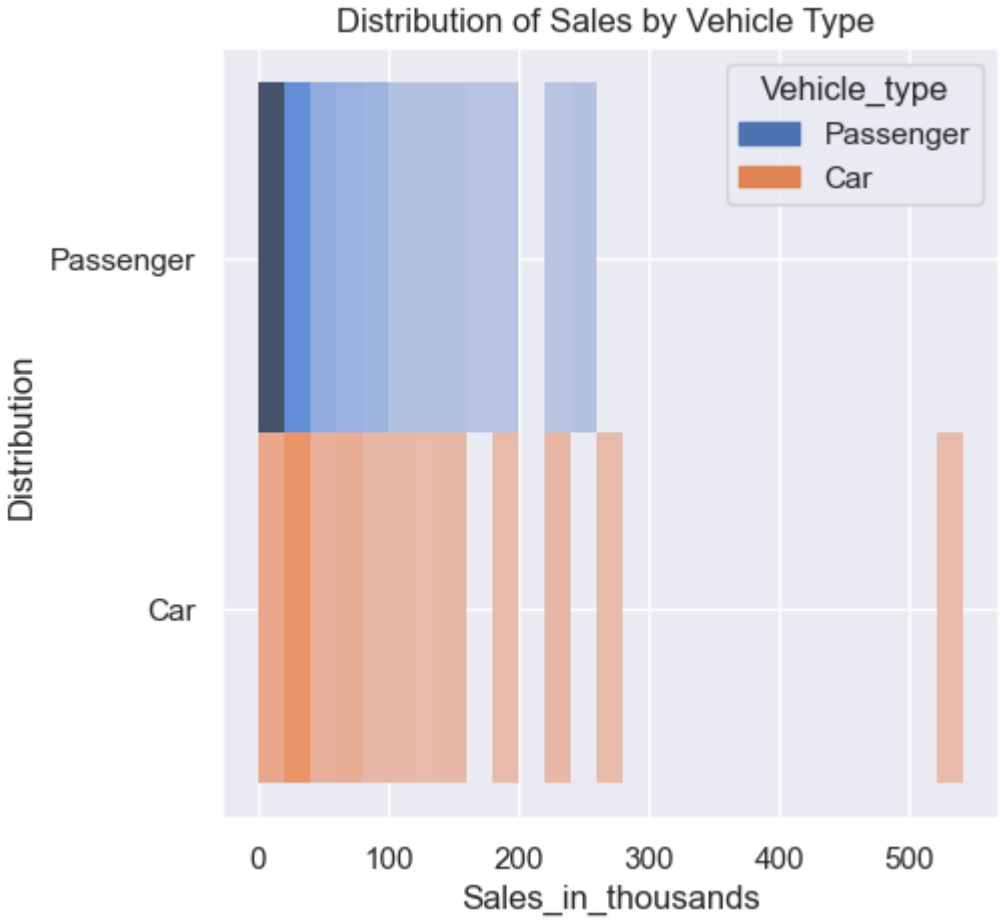


Analysing the Vehicle type

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()
```



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()
```



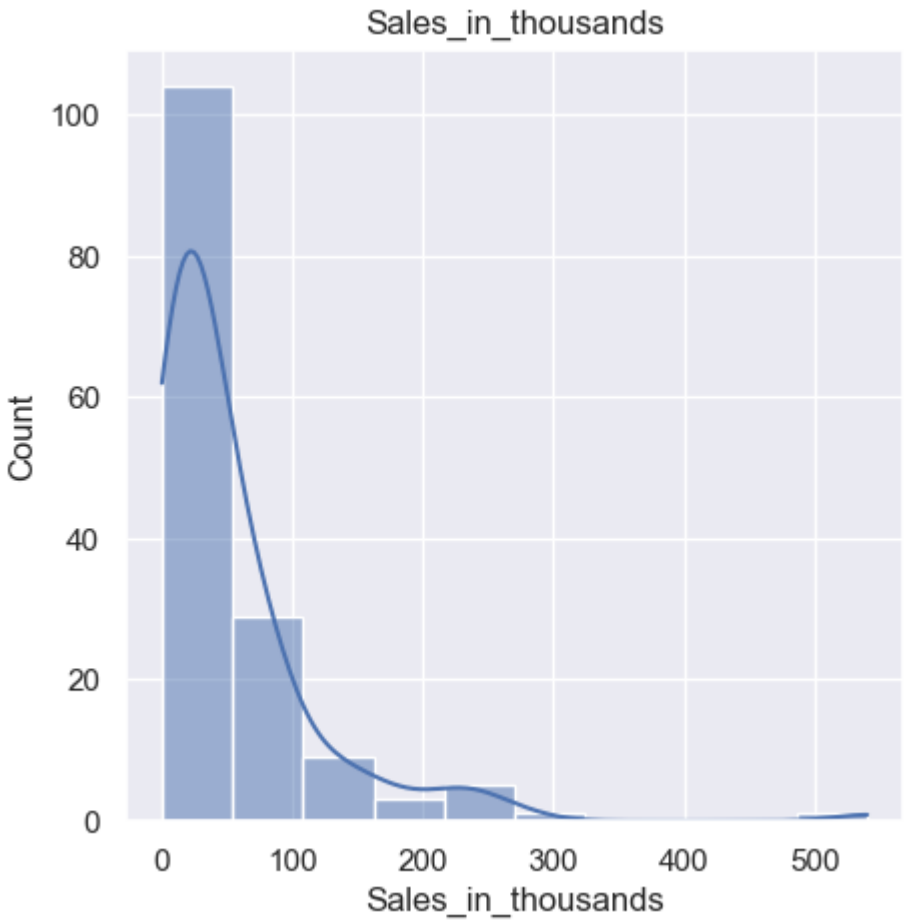
finding the feature That has more impact on car sales

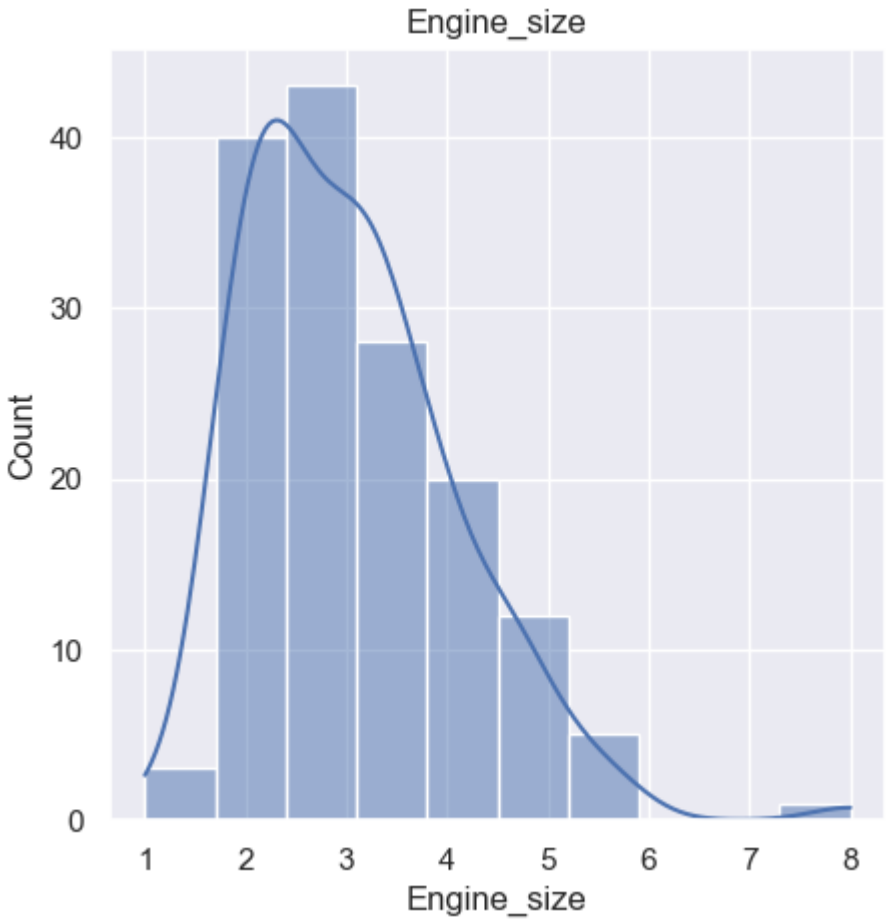
Distribution Analysis

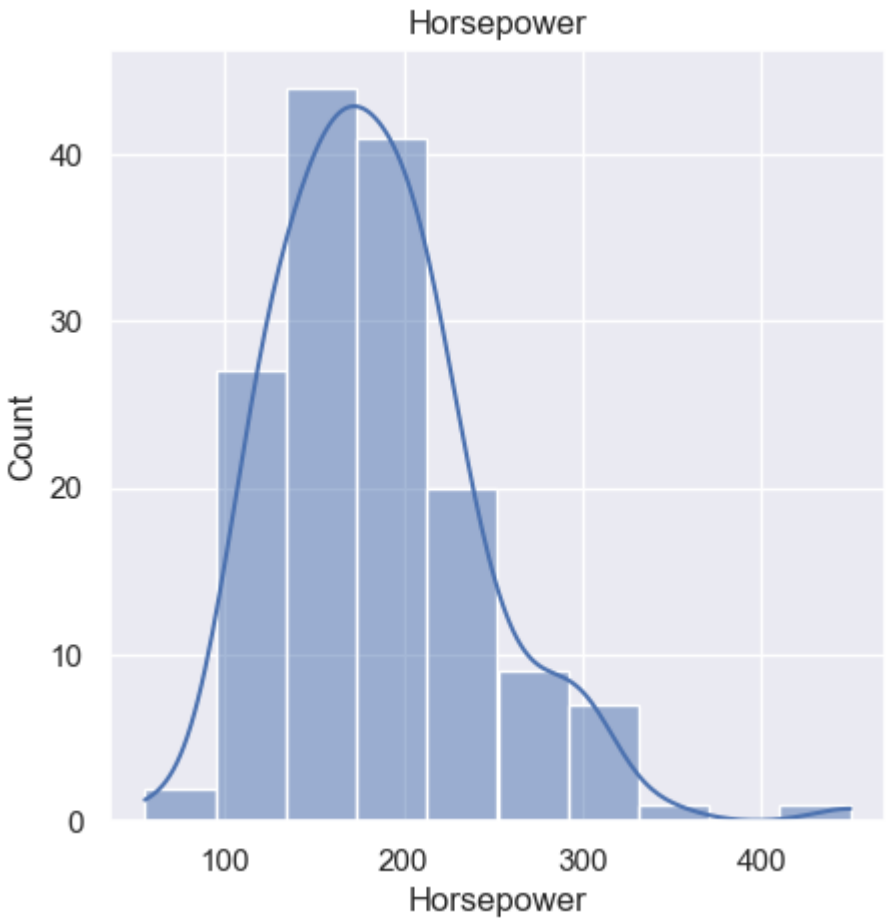
In [14]:
```
df_num = cars_data.select_dtypes(include = ['float64', 'int64'])
print(df_num.columns)
df_num = df_num.drop(columns=["__year_resale_value","Price_in_thousands"])
print(df_num.columns)
```

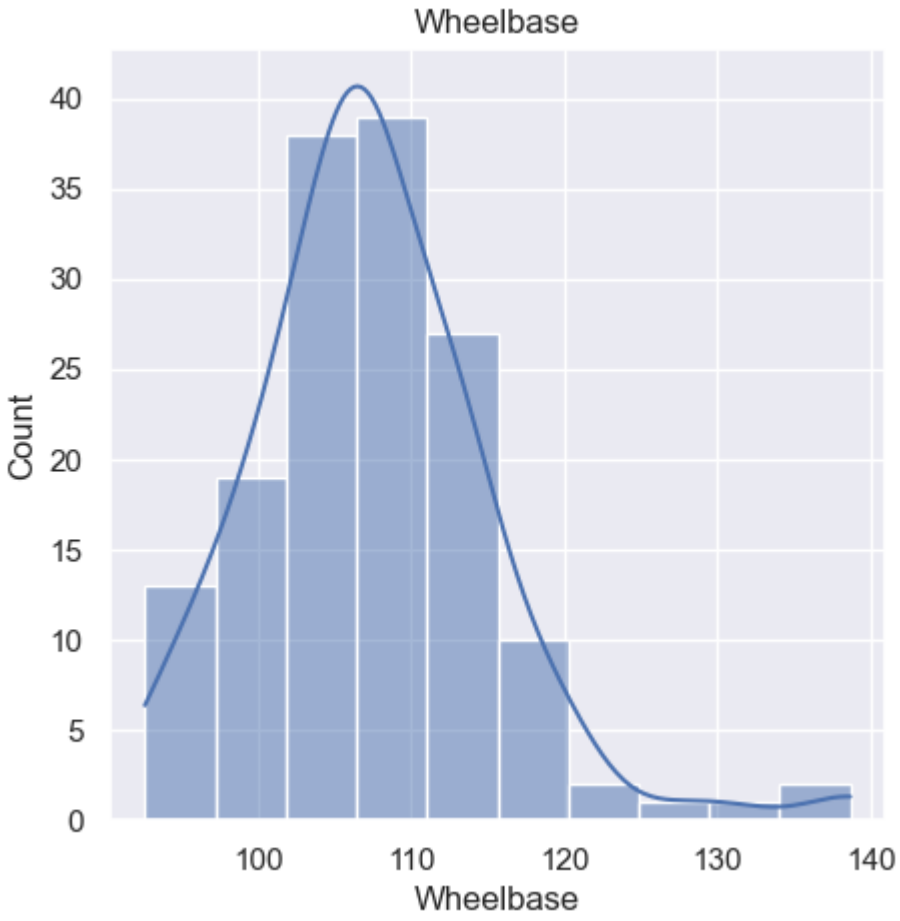
```
Index(['Sales_in_thousands', '__year_resale_value', 'Price_in_thousands',
       'Engine_size', 'Horsepower', 'Wheelbase', 'Width', 'Length',
       'Curb_weight', 'Fuel_capacity', 'Fuel_efficiency', 'Power_perf_factor'],
      dtype='object')
Index(['Sales_in_thousands', 'Engine_size', 'Horsepower',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

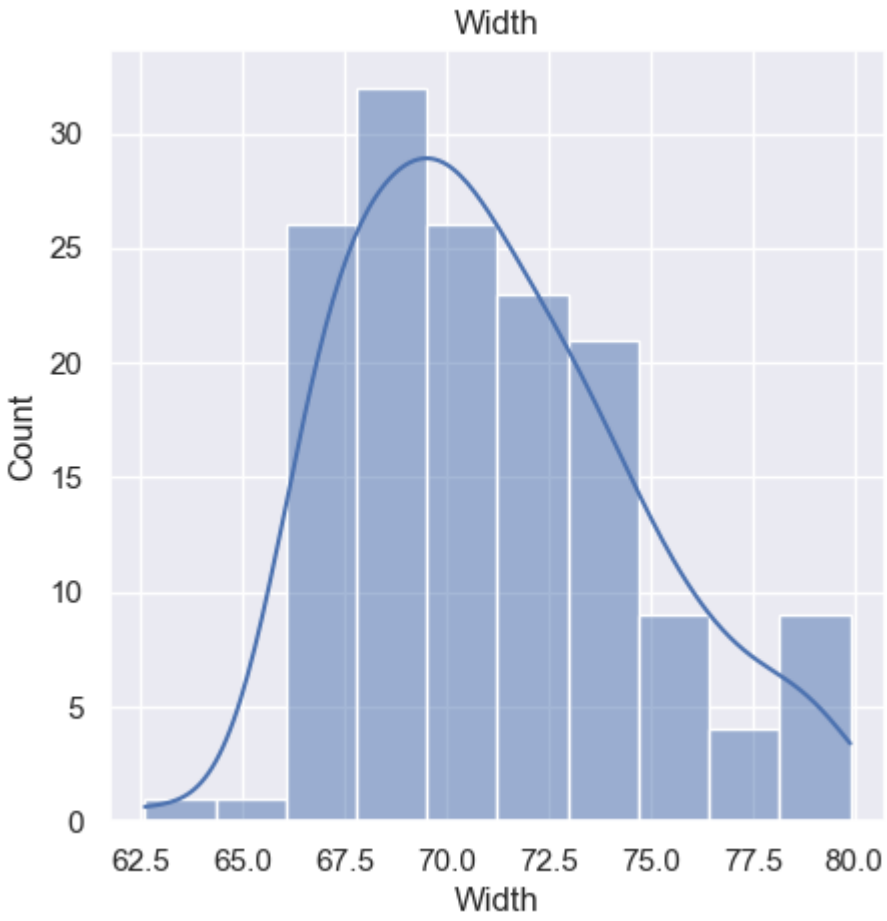
In [15]:
```
def plot():
    for p in df_num.columns:
        plt.figure(figsize=(5, 5))
        sns.histplot(df_num[p], bins=10,kde=True)
        plt.title(p)
        plt.show()
plot()
```

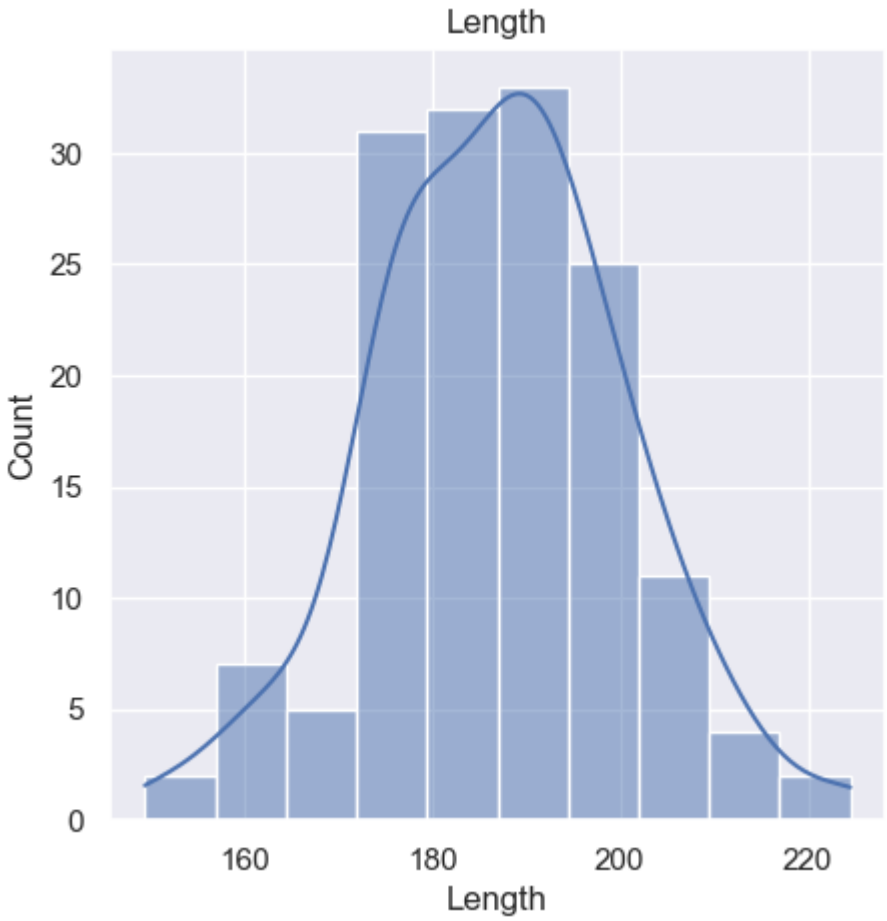

Sales_in_thousands
Count
100
80
60
40
20
0
0
100
200
300
400
500
Sales_in_thousands

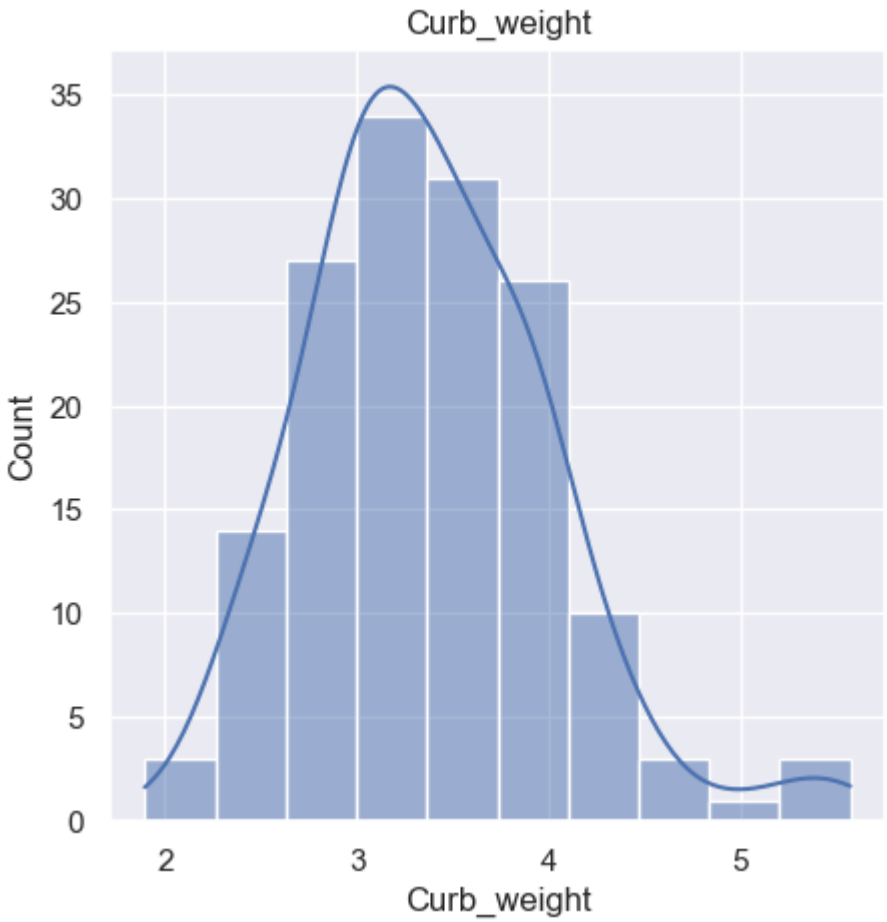

Engine_size
Count
40
30
20
10
0
1
2
3
4
5
6
7
8
Engine_size

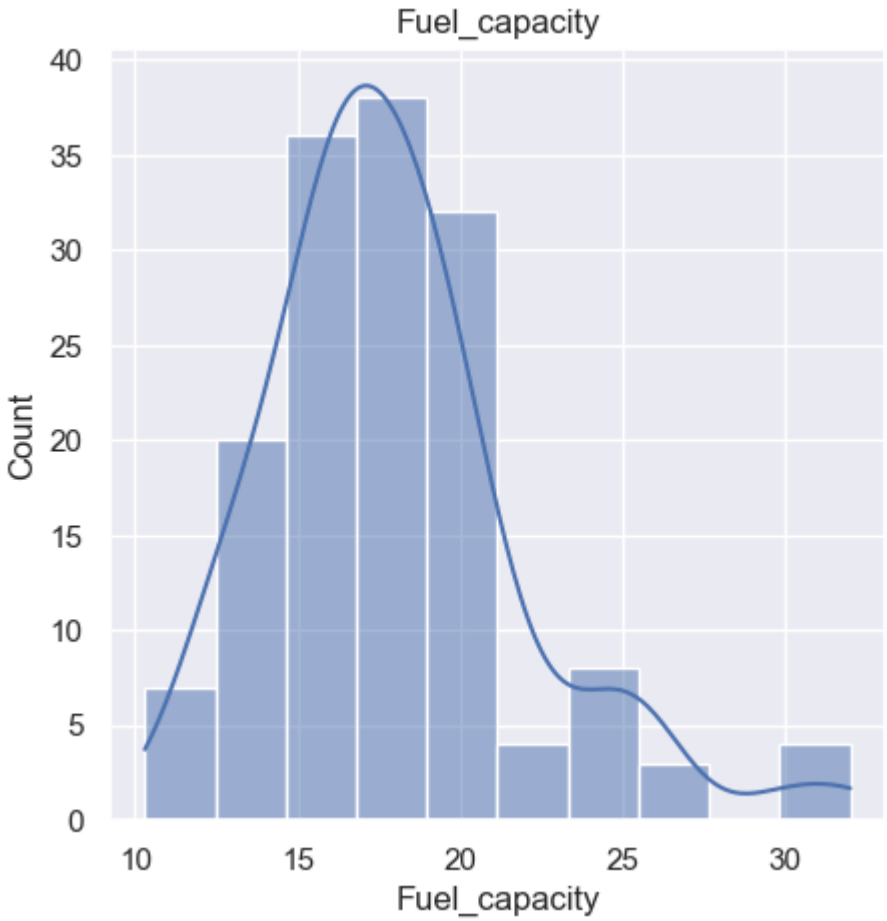

Horsepower
Count
0
10
20
30
40
100
200
300
400
Horsepower

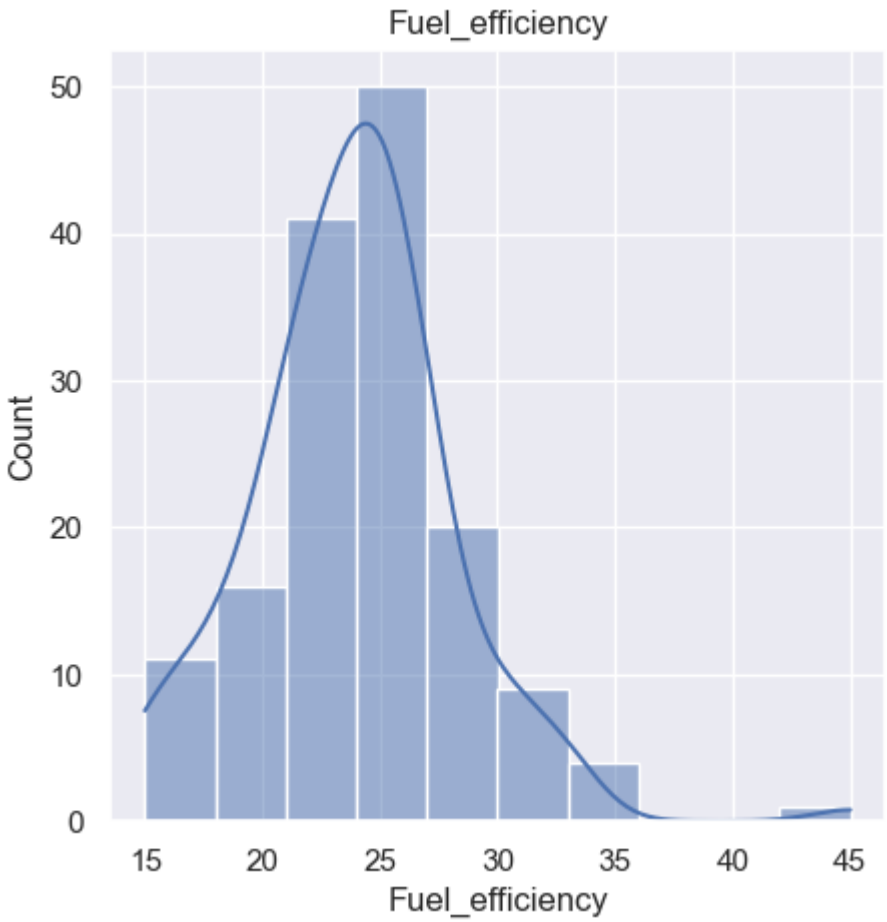

Wheelbase
Count
0
5
10
15
20
25
30
35
40
100
110
120
130
140
Wheelbase

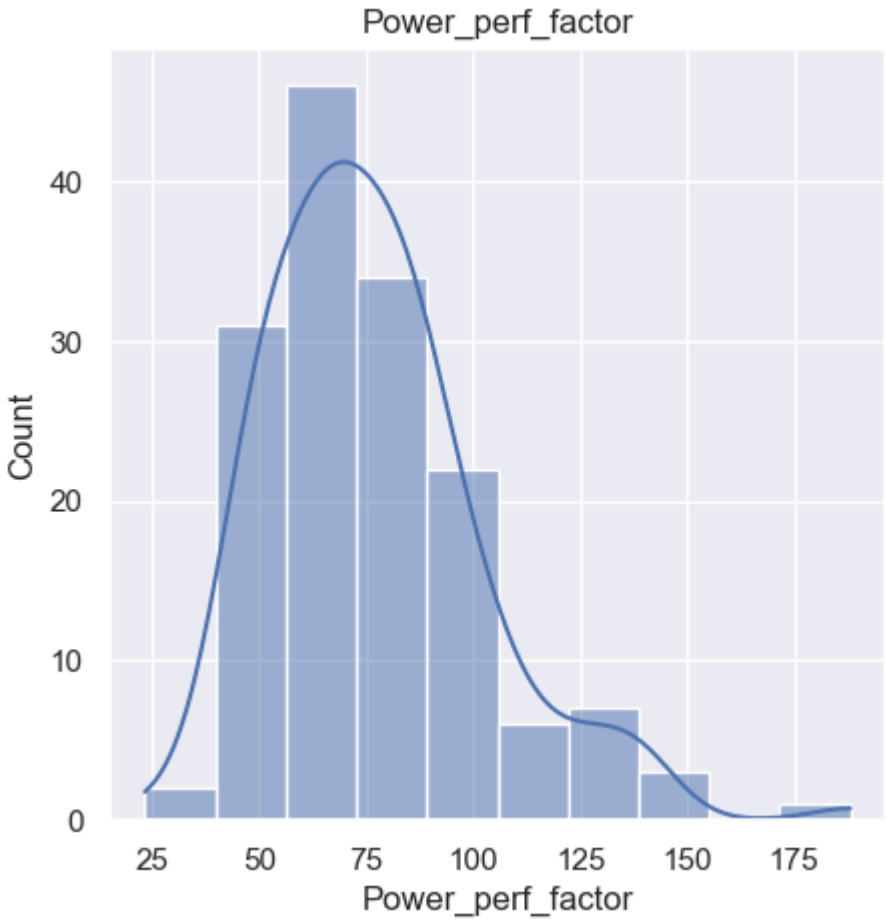

Width
Count
0
5
10
15
20
25
30
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0
Width


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

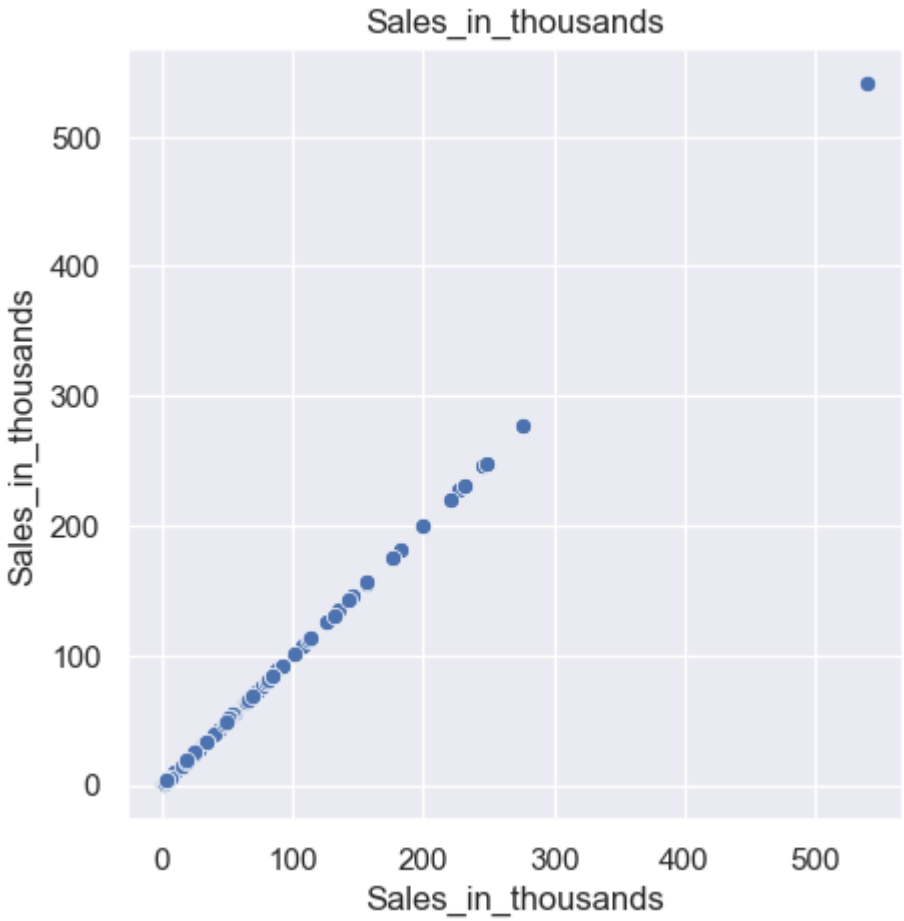

Fuel_capacity
Count
0
5
10
15
20
25
30
35
40
10
15
20
25
30
Fuel_capacity

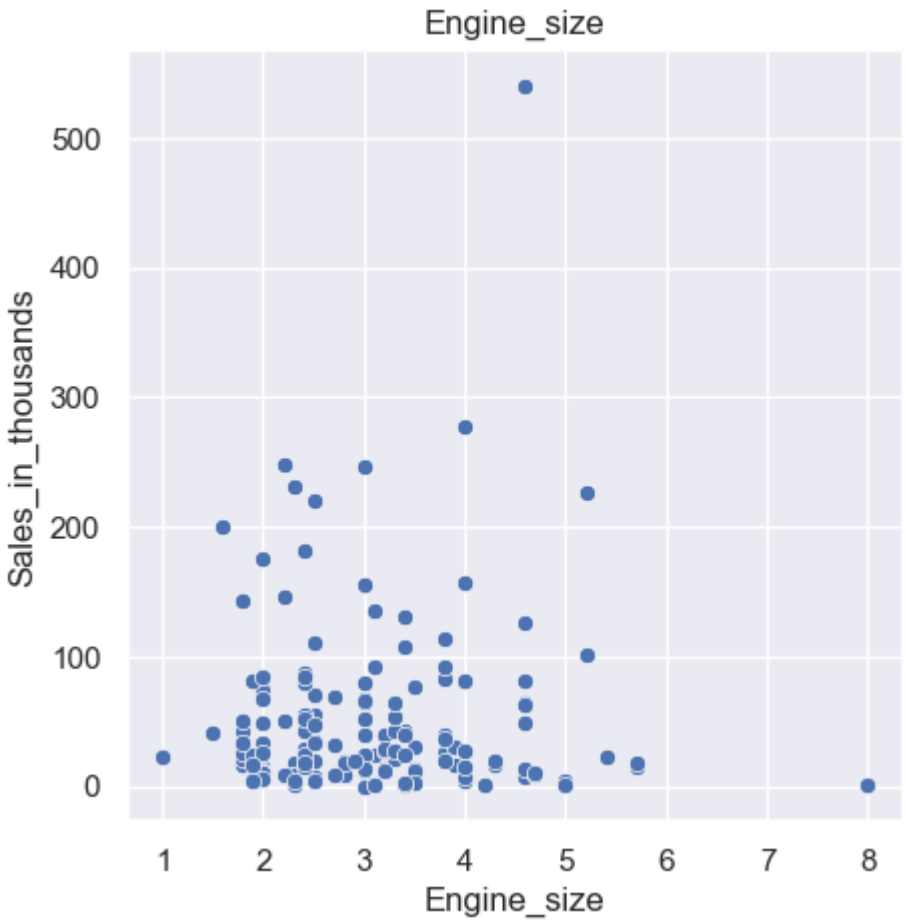

Fuel_efficiency
Count
0
10
20
30
40
50
15
20
25
30
35
40
45
Fuel_efficiency

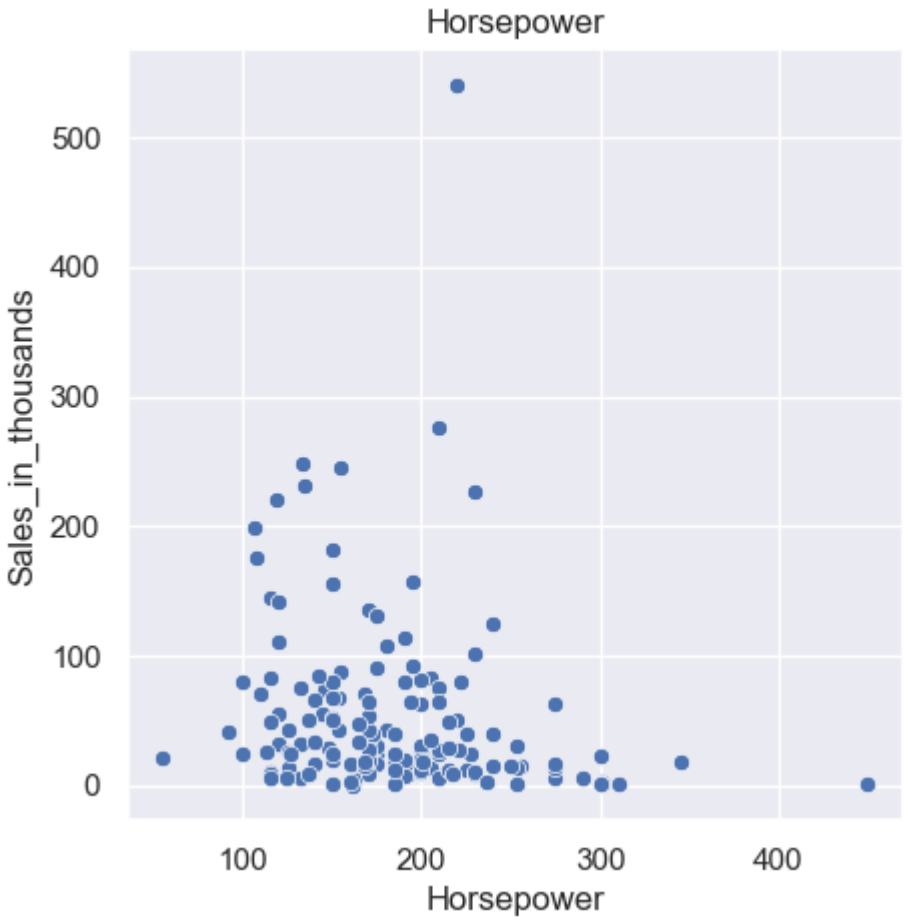

Power_perf_factor
Count
0
10
20
30
40
25
50
75
100
125
150
175
Power_perf_factor

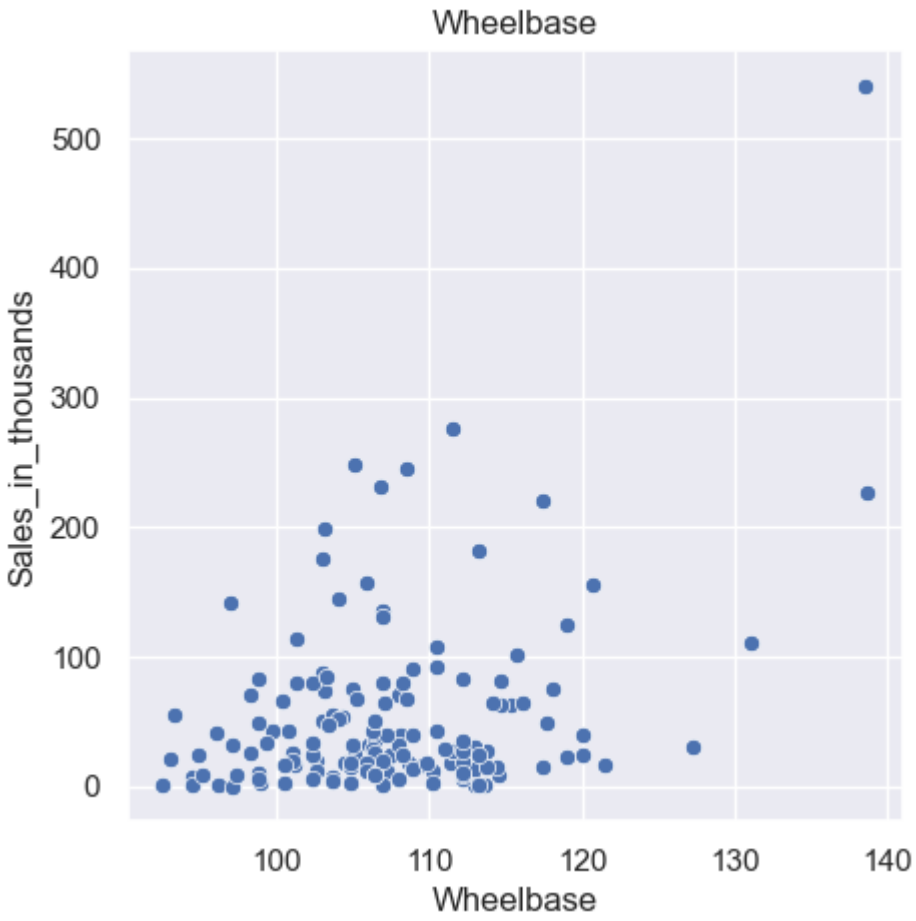
Bivariate Analysis

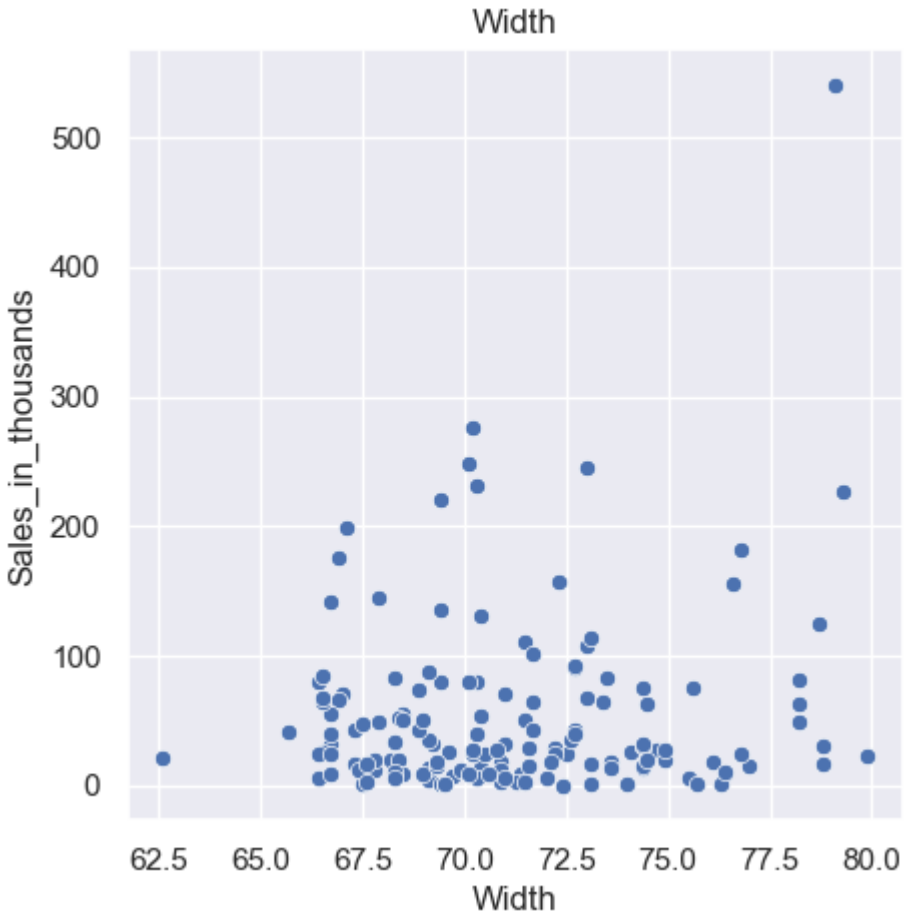
In [16]:
```
def Biplot():
    for u in df_num.columns:
        plt.figure(figsize=(5, 5))
        sns.scatterplot(x = u , y="Sales_in_thousands" , data = df_num )
        plt.title(u)
        plt.show()
Biplot()
```

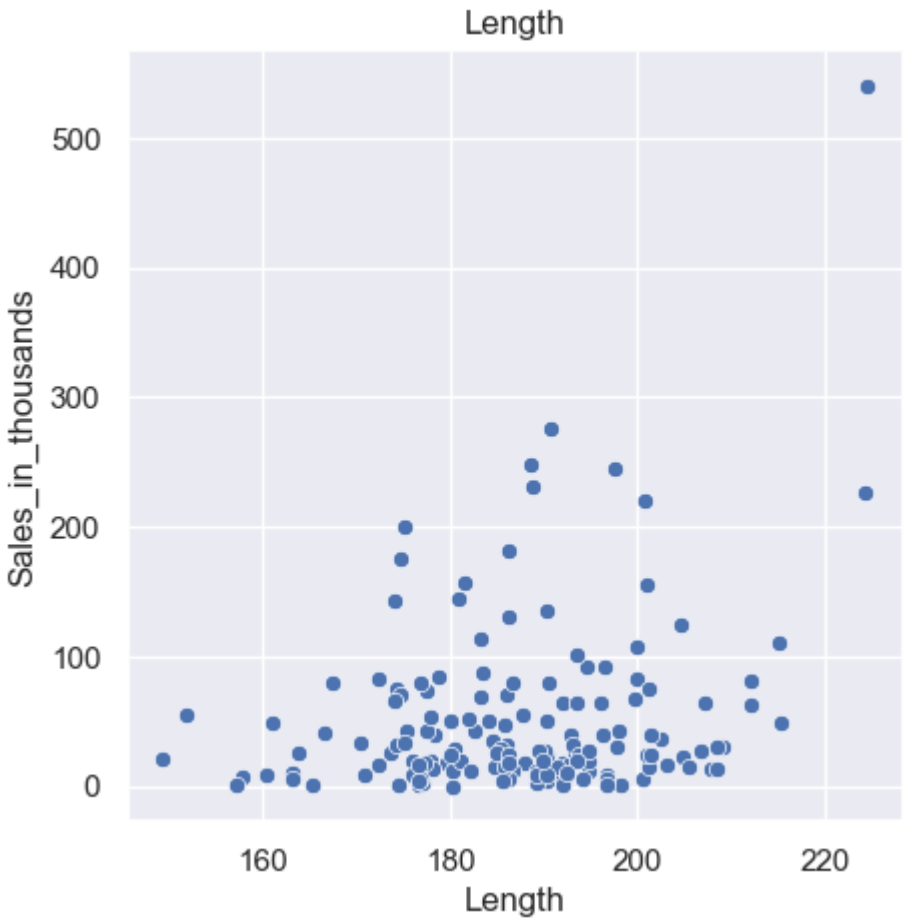


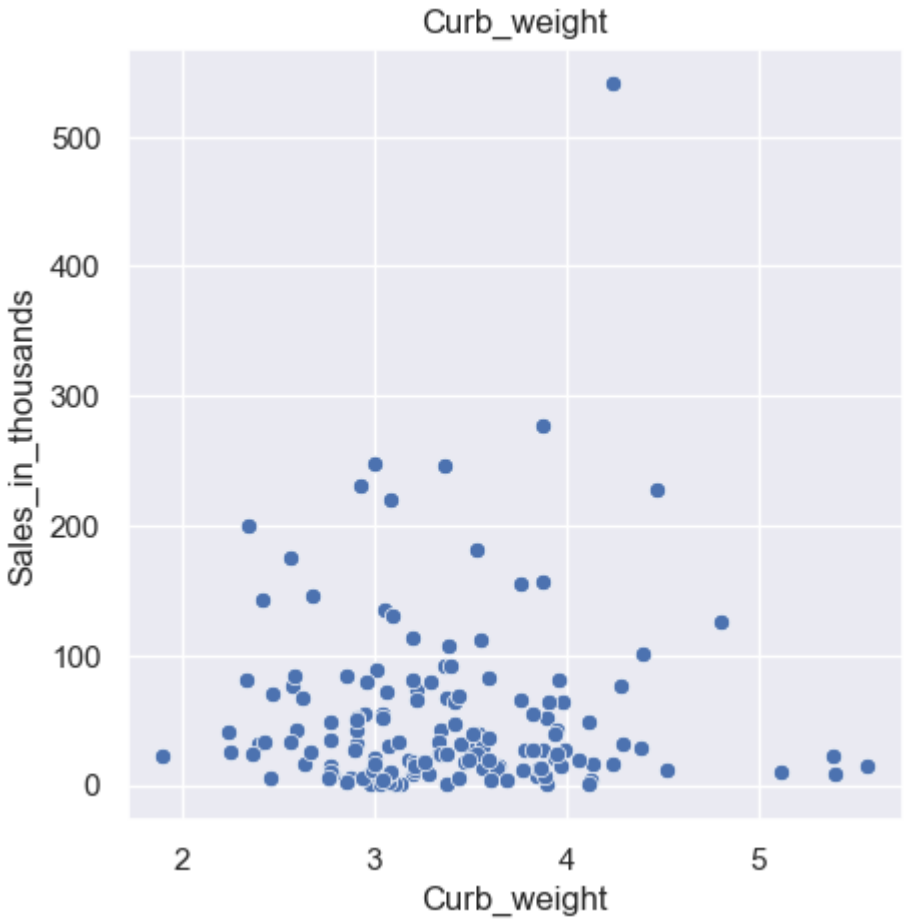

Engine_size
Sales_in_thousands
500
400
300
200
100
0
1
2
3
4
5
6
7
8
Engine_size

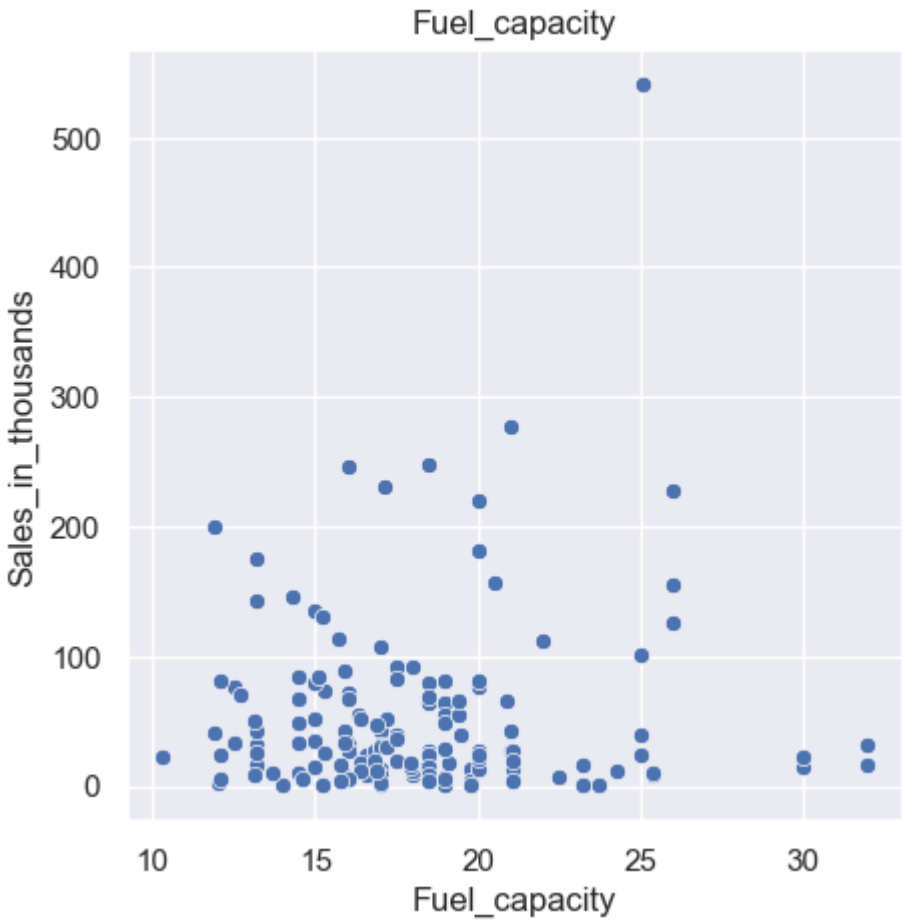

Horsepower
Sales_in_thousands
500
400
300
200
100
0
100
200
300
400
Horsepower

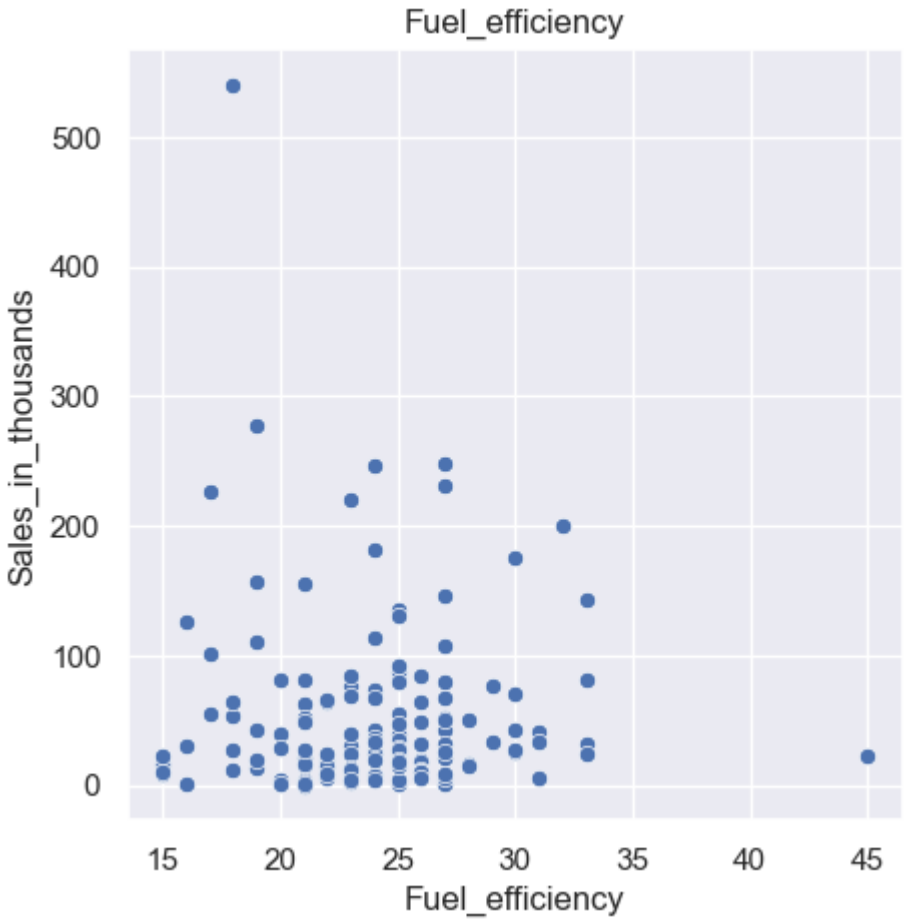

Wheelbase
Sales_in_thousands
500
400
300
200
100
0
100
110
120
130
140
Wheelbase

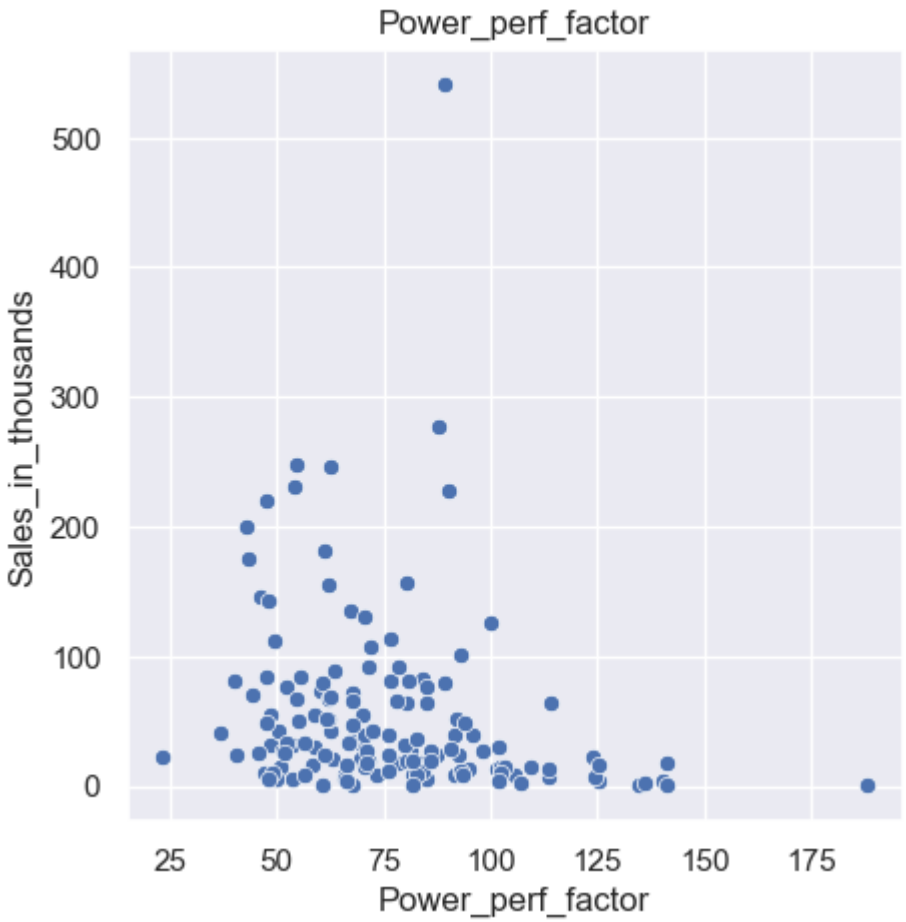

Width
Sales_in_thousands
500
400
300
200
100
0
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0
Width


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


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

In [ ]: