Plotting with pandas ¶

In [1]:

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
data = pd.read_csv('./Car_sales.csv', parse_dates = ['Latest_Launch'])
data.head(10)
```

Out[2]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelba
0	Acura	Integra	16.919	16.360	Passenger	21.50	1.8	140.0	10
1	Acura	TL	39.384	19.875	Passenger	28.40	3.2	225.0	108
2	Acura	CL	14.114	18.225	Passenger	NaN	3.2	225.0	100
3	Acura	RL	8.588	29.725	Passenger	42.00	3.5	210.0	114
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150.0	102
5	Audi	A6	18.780	23.555	Passenger	33.95	2.8	200.0	108
6	Audi	A8	1.380	39.000	Passenger	62.00	4.2	310.0	113
7	BMW	323i	19.747	NaN	Passenger	26.99	2.5	170.0	107
8	BMW	328i	9.231	28.675	Passenger	33.40	2.8	193.0	107
9	BMW	528i	17.527	36.125	Passenger	38.90	2.8	193.0	11

In [3]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 157 entries, 0 to 156
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   Manufacturer         157 non-null    object        
 1   Model                157 non-null    object        
 2   Sales_in_thousands   157 non-null    float64       
 3   __year_resale_value  121 non-null    float64       
 4   Vehicle_type         157 non-null    object        
 5   Price_in_thousands   155 non-null    float64       
 6   Engine_size          156 non-null    float64       
 7   Horsepower           156 non-null    float64       
 8   Wheelbase            156 non-null    float64       
 9   Width                156 non-null    float64       
 10  Length               156 non-null    float64       
 11  Curb_weight          155 non-null    float64       
 12  Fuel_capacity        156 non-null    float64       
 13  Fuel_efficiency      154 non-null    float64       
 14  Latest_Launch        157 non-null    datetime64[ns]
 15  Power_perf_factor    155 non-null    float64       
dtypes: datetime64[ns](1), float64(12), object(3)
memory usage: 19.8+ KB
```

In [4]:

```
data.describe()
```

Out[4]:

	Sales_in_thousands	__year_resale_value	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length	Cu
count	157.000000	121.000000	155.000000	156.000000	156.000000	156.000000	156.000000	156.000000	1
mean	52.998076	18.072975	27.390755	3.060897	185.948718	107.487179	71.150000	187.343590	
min	0.110000	5.160000	9.235000	1.000000	55.000000	92.600000	62.600000	149.400000	
25%	14.114000	11.260000	18.017500	2.300000	149.500000	103.000000	68.400000	177.575000	
50%	29.450000	14.180000	22.799000	3.000000	177.500000	107.000000	70.550000	187.900000	
75%	67.956000	19.875000	31.947500	3.575000	215.000000	112.200000	73.425000	196.125000	
max	540.561000	67.550000	85.500000	8.000000	450.000000	138.700000	79.900000	224.500000	
std	68.029422	11.453384	14.351653	1.044653	56.700321	7.641303	3.451872	13.431754	

In [5]:

```
data.isnull().mean()*100
```

Out[5]:

```
Manufacturer            0.000000
Model                   0.000000
Sales_in_thousands      0.000000
__year_resale_value    22.929936
Vehicle_type            0.000000
Price_in_thousands      1.273885
Engine_size             0.636943
Horsepower              0.636943
Wheelbase               0.636943
Width                   0.636943
Length                  0.636943
Curb_weight             1.273885
Fuel_capacity           0.636943
Fuel_efficiency         1.910828
Latest_Launch           0.000000
Power_perf_factor       1.273885
dtype: float64
```

In [6]:

```
data.dropna(inplace=True)
data.drop_duplicates(inplace=True)
```

In [7]:

```
data.shape
```

Out[7]:

```
(117, 16)
```

In [8]:

```
data.sort_values(by='Latest_Launch', inplace=True)
```

In [9]:

```
data.reset_index(inplace=True, drop=True)
```

In [10]:

```
data.head(3)
```

Out[10]:

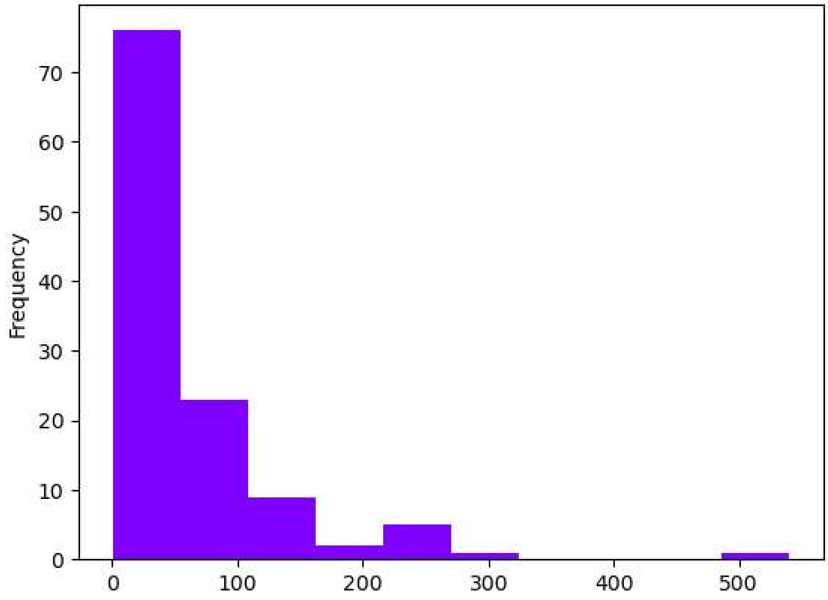
	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wh
0	Mercury	Mountaineer	27.609	20.430	Car	27.56	4.0	210.0	
1	Mercury	Villager	20.380	14.795	Car	22.51	3.3	170.0	
2	Saturn	SW	5.223	10.790	Passenger	14.29	1.9	124.0	

Univariate Analysis

Histogram

In [11]:

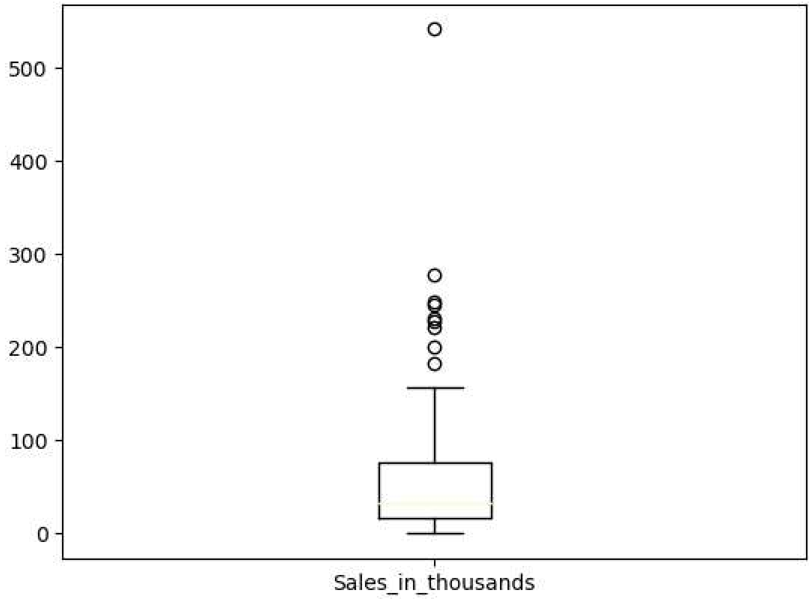
```
data.Sales_in_thousands.plot.hist(cmap='rainbow');
```



Box plot

In [12]:

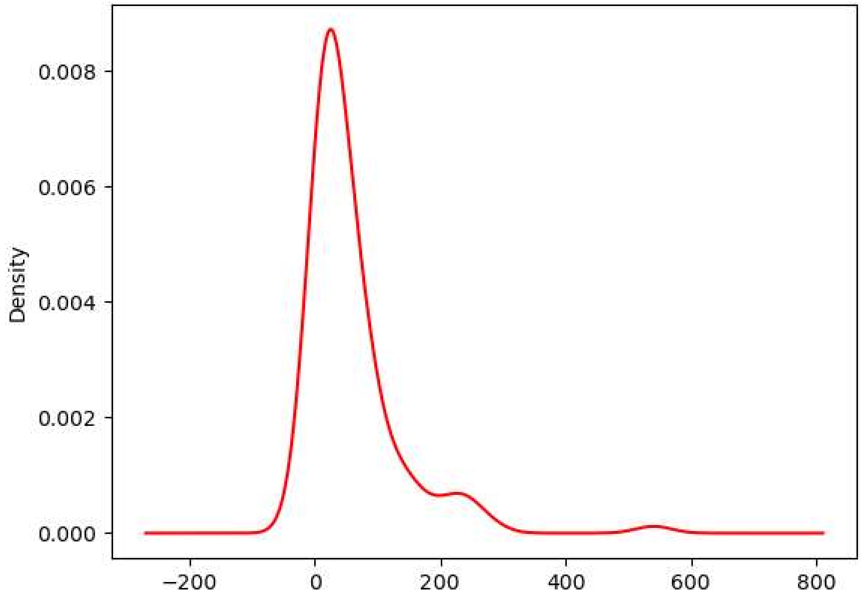
```
data.Sales_in_thousands.plot.box(cmap ='magma');
```



Kernel Density Estimation Plot (KDE PLOT)

In [13]:

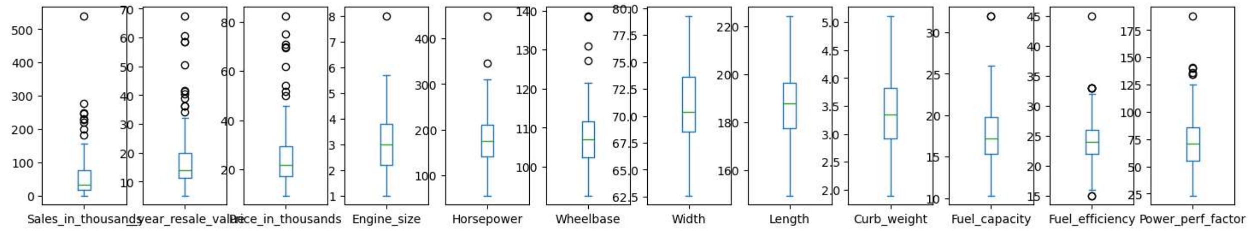
```
data.Sales_in_thousands.plot.kde(cmap = 'flag');
```



Sub plots

In [14]:

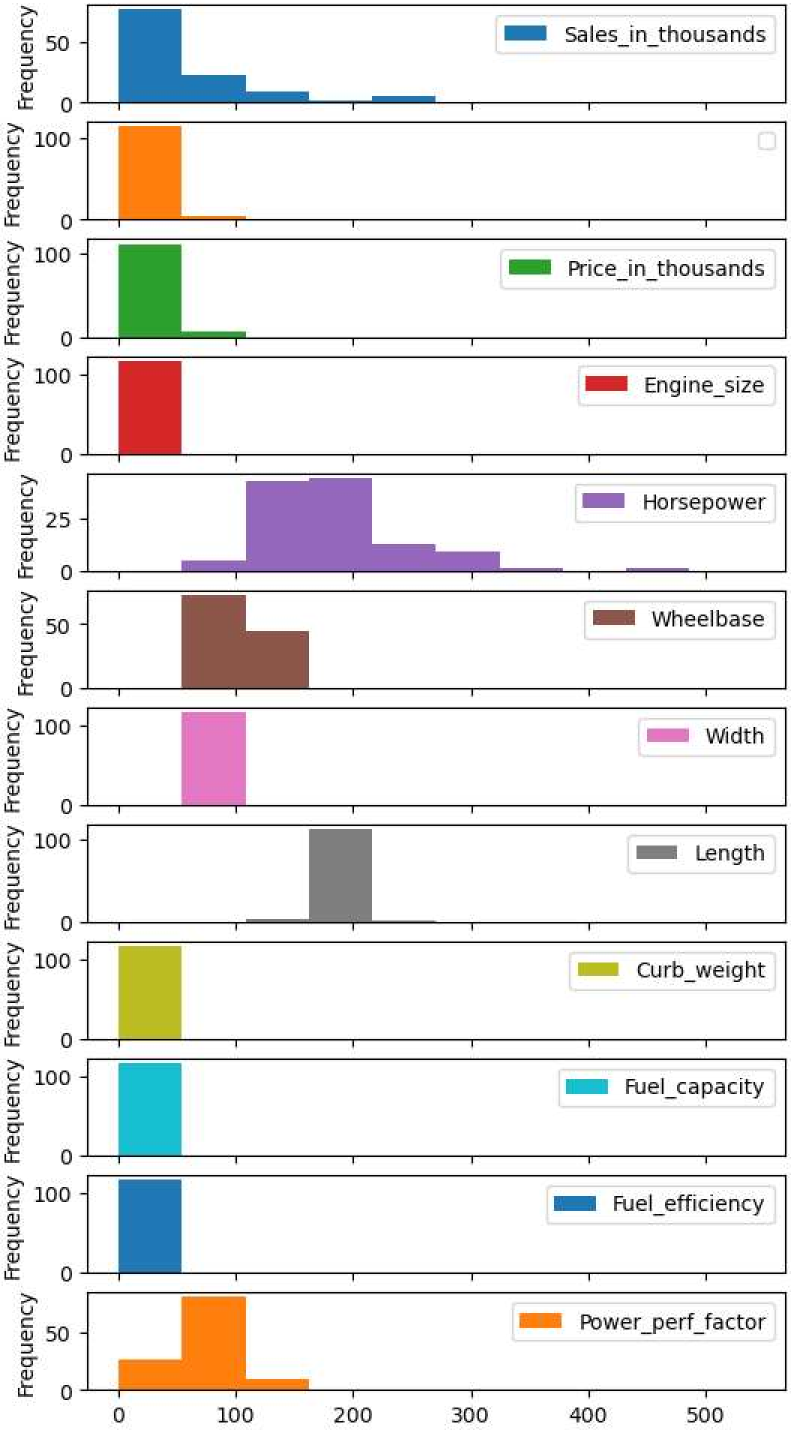
```
data.plot.box( figsize=(18,3), subplots = True);
```



In [15]:

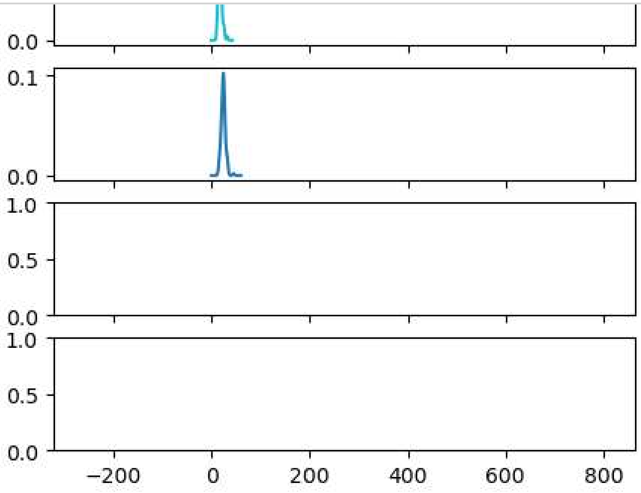
```
data.plot.hist( figsize=(6,12), subplots = True);
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are i
gnored when legend() is called with no argument.
```



In [16]:

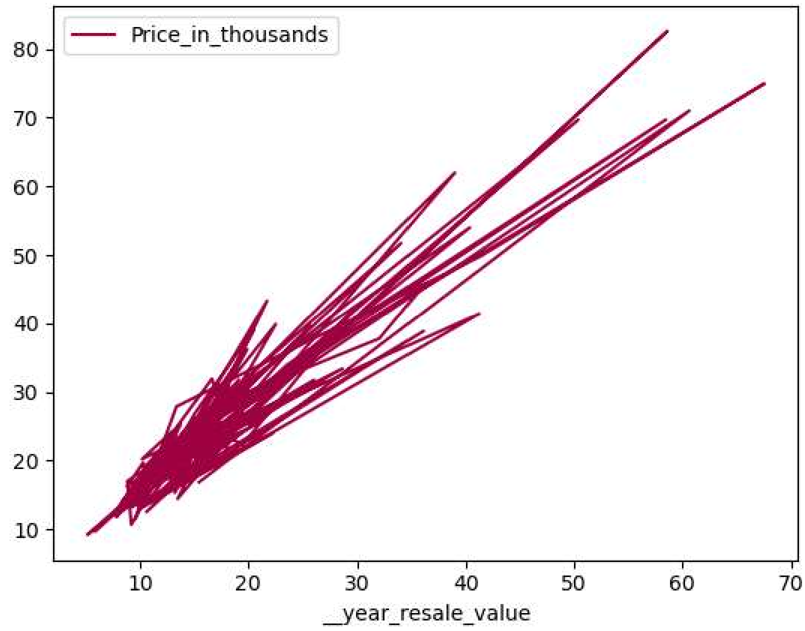
```
data.plot.kde(figsize = (5,15),subplots = True);
```



Line charts

In [17]:

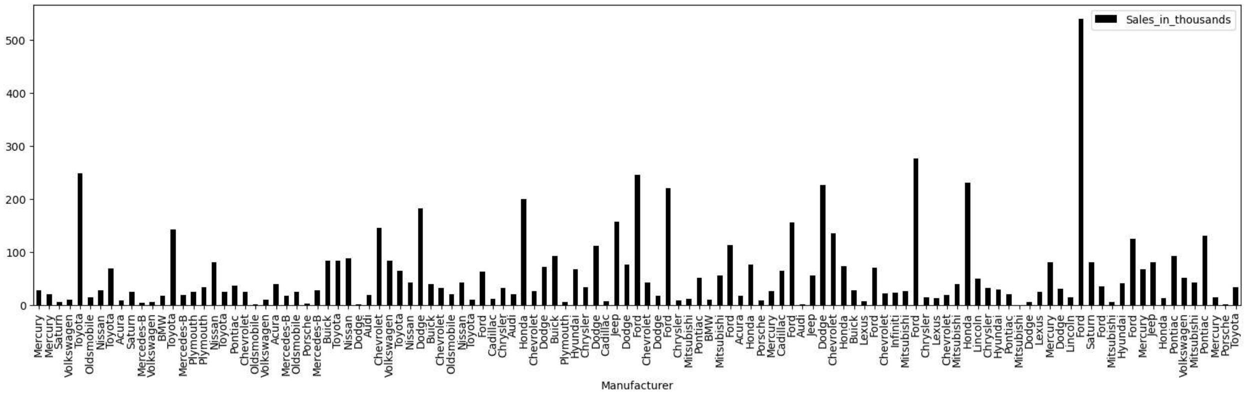
```
data.plot.line(x='__year_resale_value', y='Price_in_thousands', cmap = 'Spectral');
```



Bar chart

In [18]:

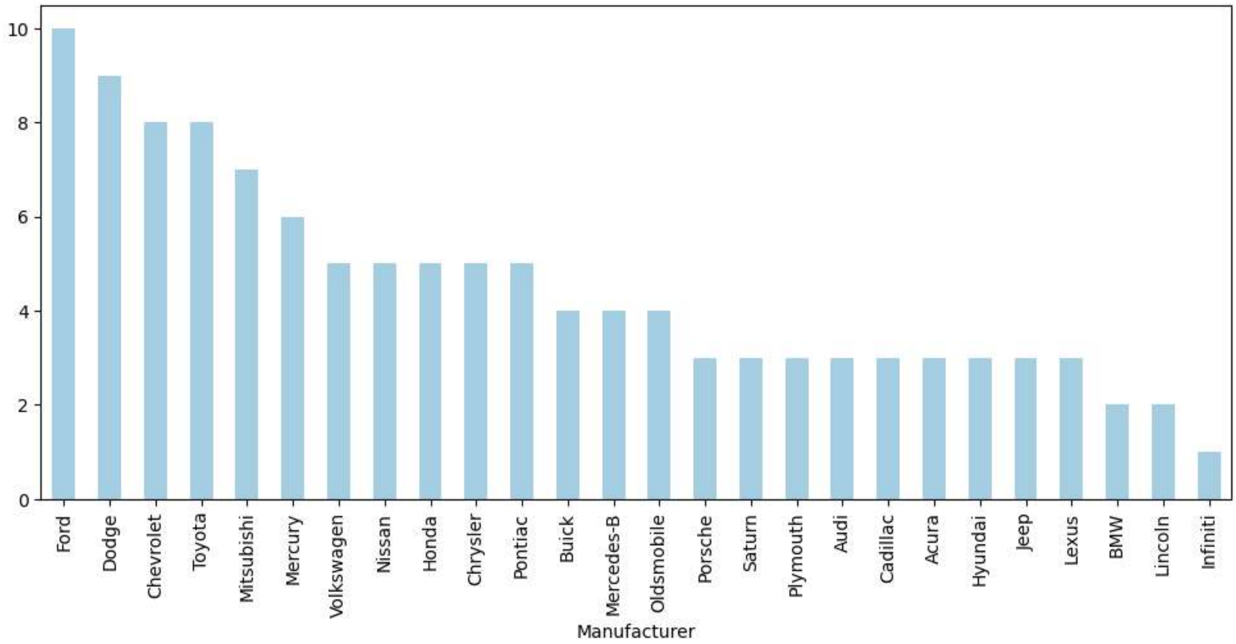
```
data.plot.bar(x='Manufacturer', y='Sales_in_thousands', figsize = (20, 5),cmap = 'gist_earth');
```



Count plot

In [19]:

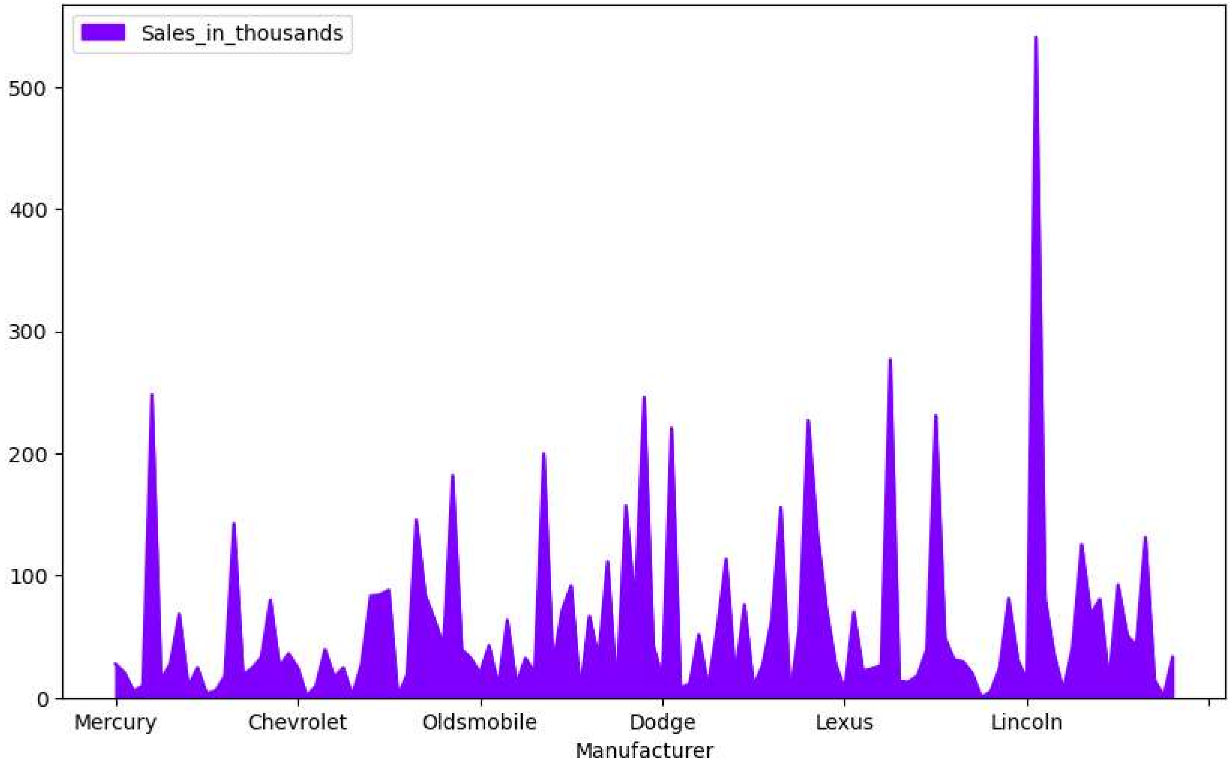
```
data.Manufacturer.value_counts().plot.bar(figsize=(12,5), cmap = 'Paired');
```



Area chart

In [20]:

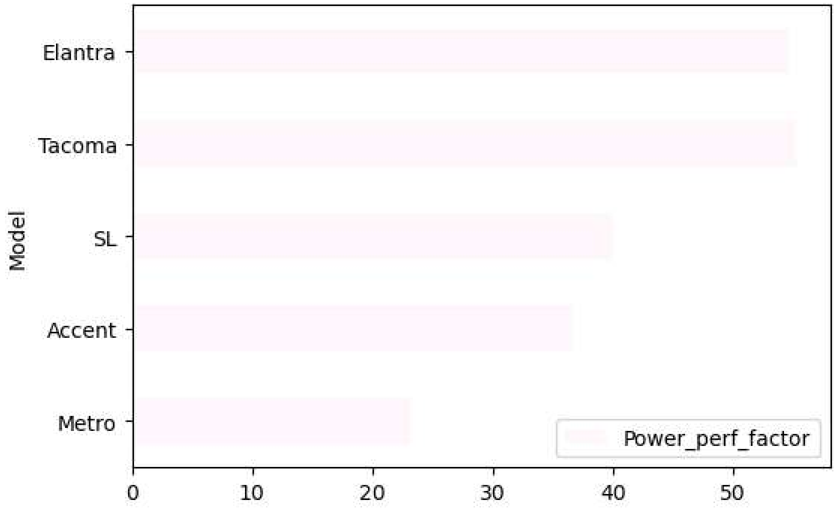
```
data.plot.area(x='Manufacturer', y='Sales_in_thousands', figsize = (10,6), cmap = 'rainbow');
```



Horizontal bar chart

In [21]:

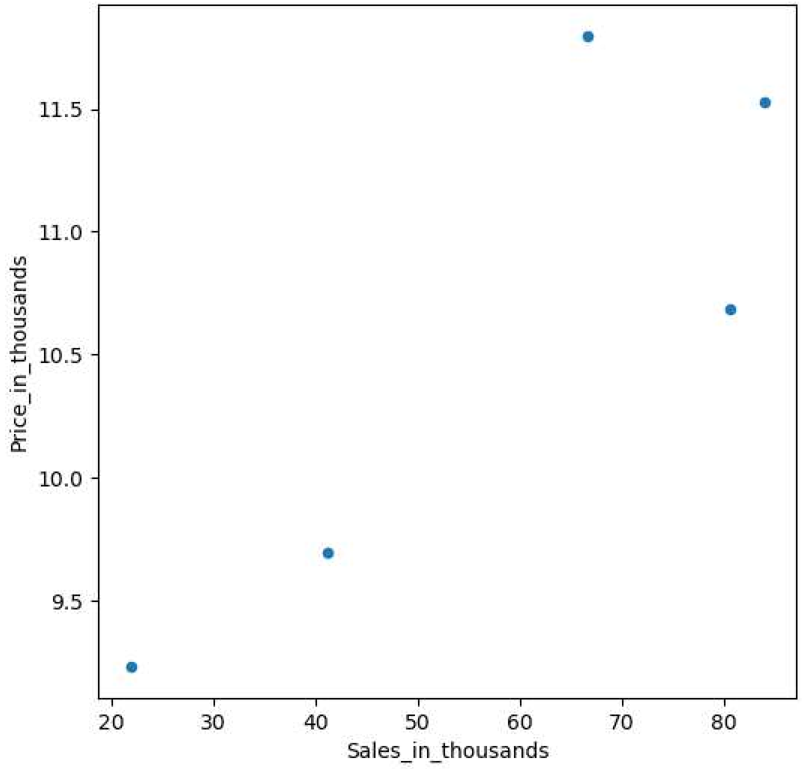
```
data.sort_values(by='Price_in_thousands')[:5].plot.barh(x='Model', y= 'Power_perf_factor', figsize = (6,4), cmap = 'PuBu')
```



Scatter plot

In [22]:

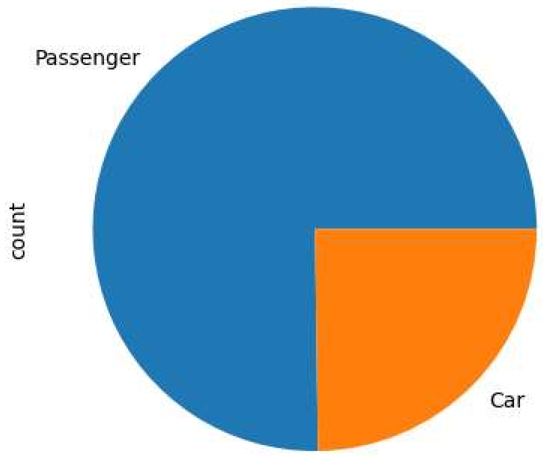
```
data.sort_values(by='Price_in_thousands')[:5].plot.scatter(x='Sales_in_thousands', y='Price_in_thousands',
                                                            figsize = (6,6), cmap='PiYG');
```



Pie chart

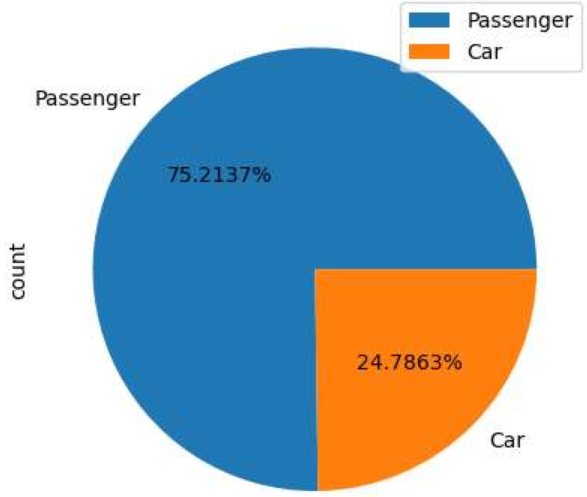
In [23]:

```
data.Vehicle_type.value_counts().plot.pie();
```



In [24]:

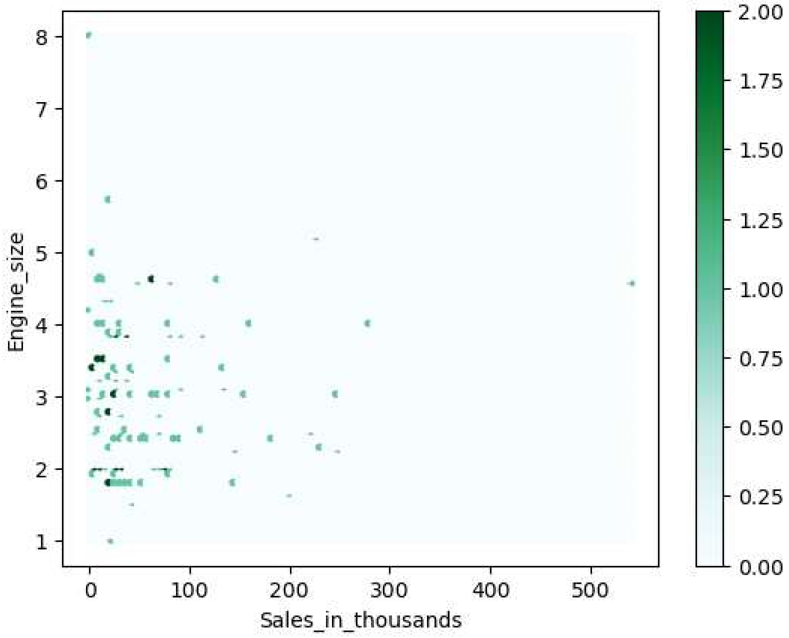
```
data.Vehicle_type.value_counts().plot.pie(autopct ='%1.4f%%', legend = True);
```



Hexbin plot

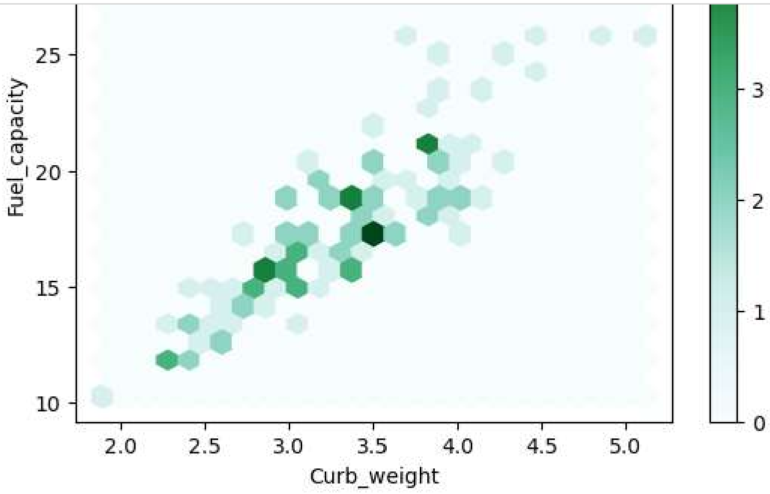
In [25]:

```
data.plot.hexbin('Sales_in_thousands', 'Engine_size');
```



In [26]:

```
data.plot.hexbin('Curb_weight', 'Fuel_capacity', gridsize=25);
```

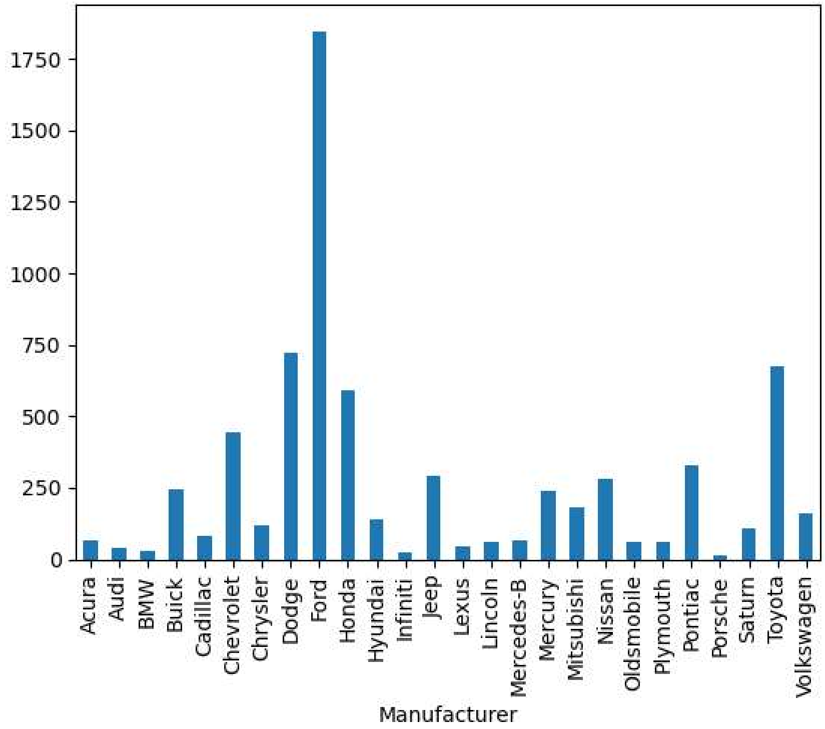


In [27]:

```
Manufacturer_Sales=data.groupby(['Manufacturer'])['Sales_in_thousands'].sum()
```

In [28]:

```
Manufacturer_Sales.plot.bar();
```



In [29]:

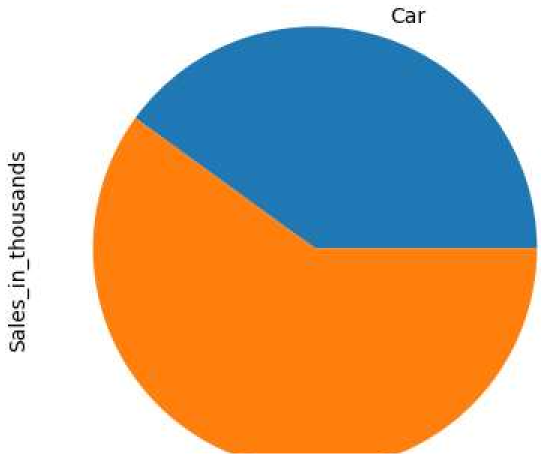
```
Vehicle_type_Sales=data.groupby(['Vehicle_type'])['Sales_in_thousands'].sum()
Vehicle_type_Sales
```

Out[29]:

```
Vehicle_type
Car          2766.779
Passenger    4149.362
Name: Sales_in_thousands, dtype: float64
```

In [30]:

```
Vehicle_type_Sales.plot.pie();
```


Car
Sales_in_thousands