

## 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')
data.sample(10)
```

Out[2]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length
78	Lincoln	Navigator	22.925	NaN	Car	42.660	5.4	300.0	119.0	79.9	204.8
63	Hyundai	Elantra	66.692	7.825	Passenger	11.799	2.0	140.0	100.4	66.9	174.0
25	Chevrolet	Prizm	32.299	9.125	Passenger	13.960	1.8	120.0	97.1	66.7	174.3
12	Buick	Park Avenue	27.851	20.190	Passenger	31.965	3.8	205.0	113.8	74.7	206.8
1	Acura	TL	39.384	19.875	Passenger	28.400	3.2	225.0	108.1	70.3	192.9
15	Cadillac	Seville	15.943	27.100	Passenger	44.475	4.6	275.0	112.2	75.0	201.0
132	Saturn	LW	8.472	NaN	Passenger	18.835	2.2	137.0	106.5	69.0	190.4
24	Chevrolet	Corvette	17.947	36.225	Passenger	45.705	5.7	345.0	104.5	73.6	179.7
111	Oldsmobile	Aurora	14.690	19.890	Passenger	36.229	4.0	250.0	113.8	74.4	205.4
30	Chrysler	Concorde	31.148	13.725	Passenger	22.245	2.7	200.0	113.0	74.4	209.1

### Dropping na values

```
In [3]: data.dropna(inplace=True)
```

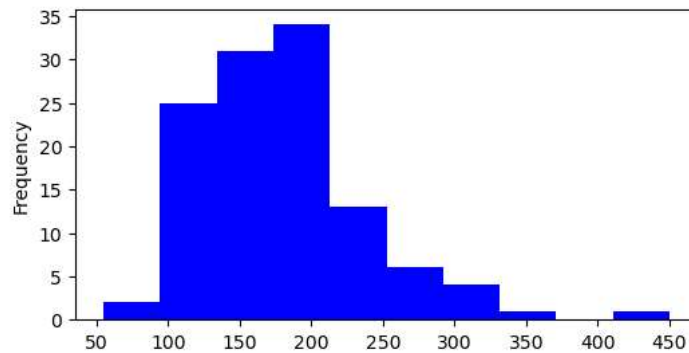
```
In [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 117 entries, 0 to 149
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Manufacturer          117 non-null    object
1   Model                 117 non-null    object
2   Sales_in_thousands    117 non-null    float64
3   __year_resale_value    117 non-null    float64
4   Vehicle_type          117 non-null    object
5   Price_in_thousands    117 non-null    float64
6   Engine_size           117 non-null    float64
7   Horsepower            117 non-null    float64
8   Wheelbase             117 non-null    float64
9   Width                 117 non-null    float64
10  Length                117 non-null    float64
11  Curb_weight           117 non-null    float64
12  Fuel_capacity          117 non-null    float64
13  Fuel_efficiency        117 non-null    float64
14  Latest_launch          117 non-null    object
15  Power_perf_factor      117 non-null    float64
dtypes: float64(12), object(4)
memory usage: 15.5+ KB
```

## Plots on numerical value columns

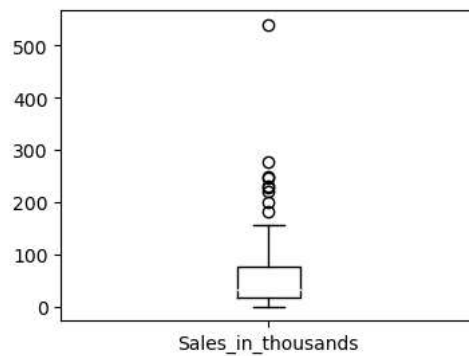
### Histogram

```
In [5]: data.Horsepower.plot.hist(figsize=(6,3), cmap='brg');
```

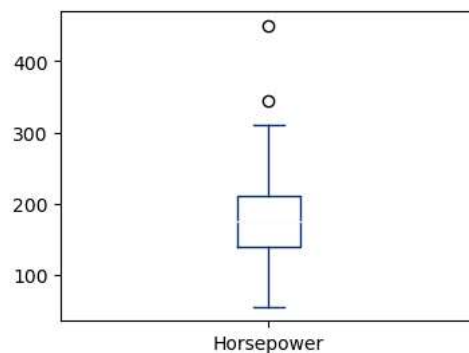


### Box plot

```
In [6]: data.Sales_in_thousands.plot.box(figsize=(4,3), cmap = 'gist_earth');
```

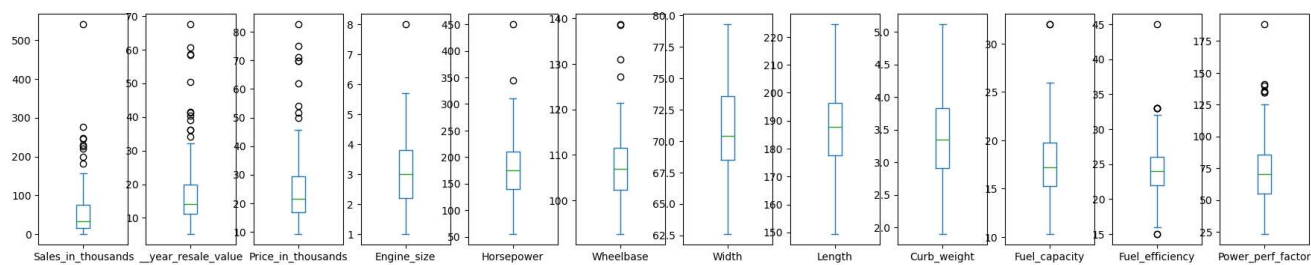


```
In [7]: data.Horsepower.plot.box(figsize=(4,3), cmap = 'Blues_r');
```



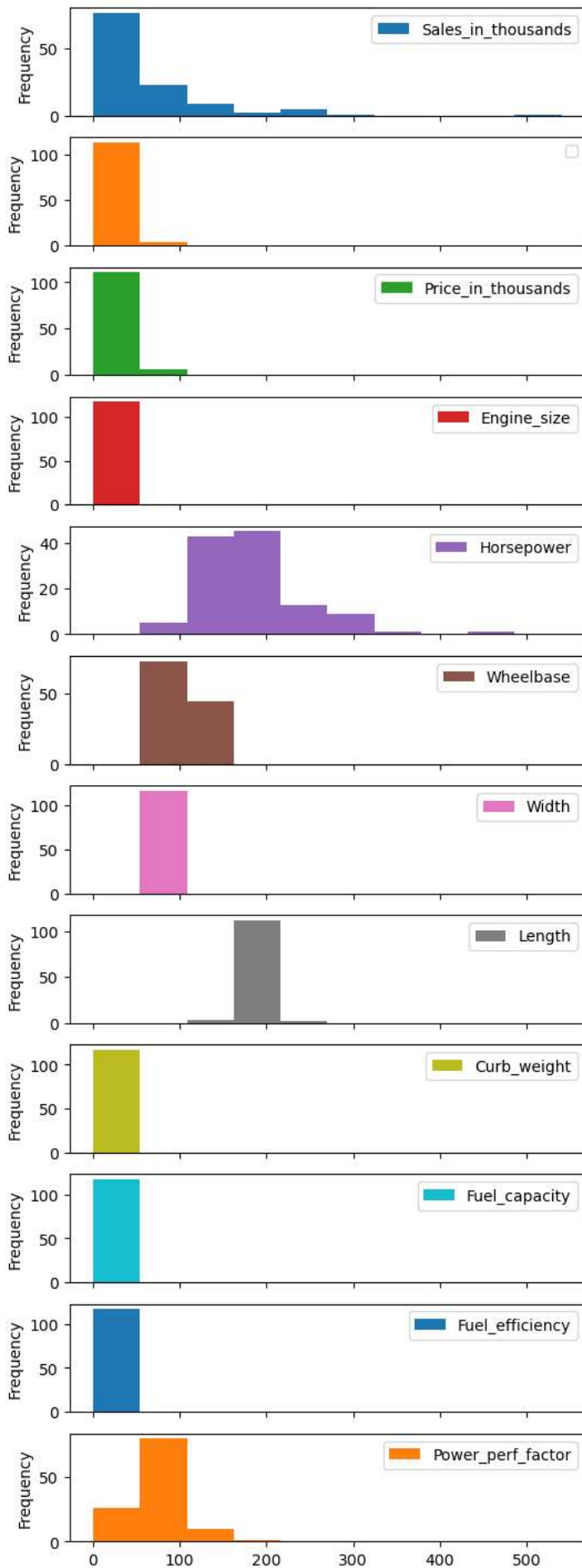
## Subplots

```
In [8]: data.plot.box(figsize=(22,4), subplots = True);
```



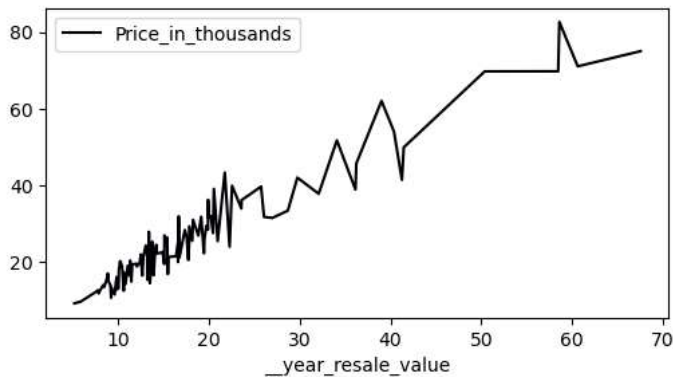
```
In [9]: data.plot.hist(figsize=(6,18),subplots = True);
```

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



## Line charts

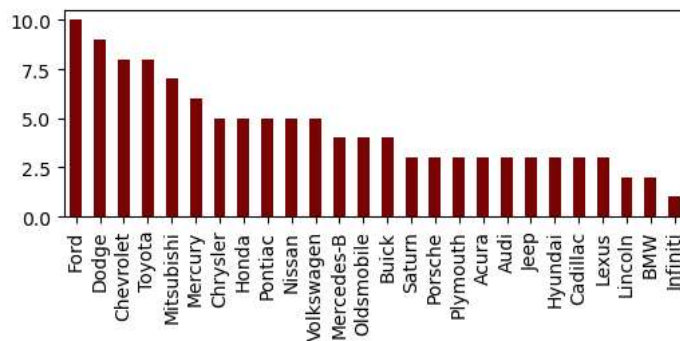
```
In [10]: data.sort_values(by='__year_resale_value').plot.line('__year_resale_value', 'Price_in_thousands', figsize=(6,3), cmap = 'inferno')
```



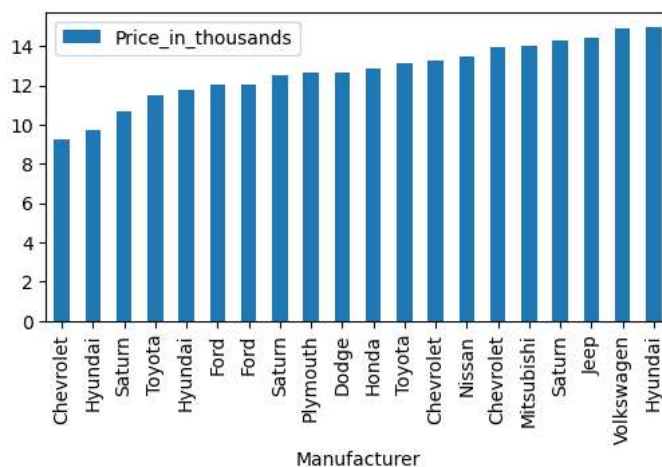
## Bar plot

To create the below shown bar chart we have to create the frequency table. Frequency table can be made using the `value_counts()` method.

```
In [11]: data.Manufacturer.value_counts().plot.bar(figsize=(6,2), cmap = 'turbo_r');
```

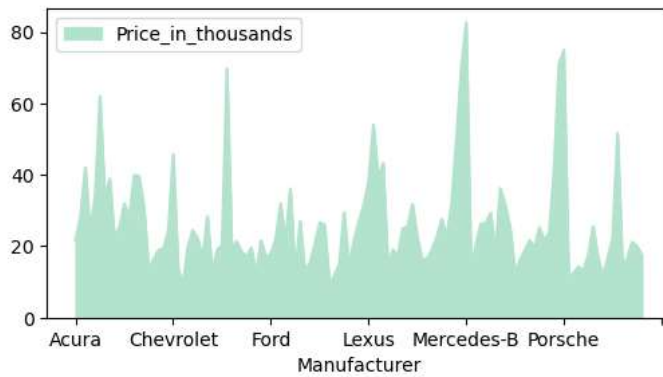


```
In [12]: data.sort_values(by='Price_in_thousands')[20].plot.bar('Manufacturer', 'Price_in_thousands', figsize=(6,3));
```

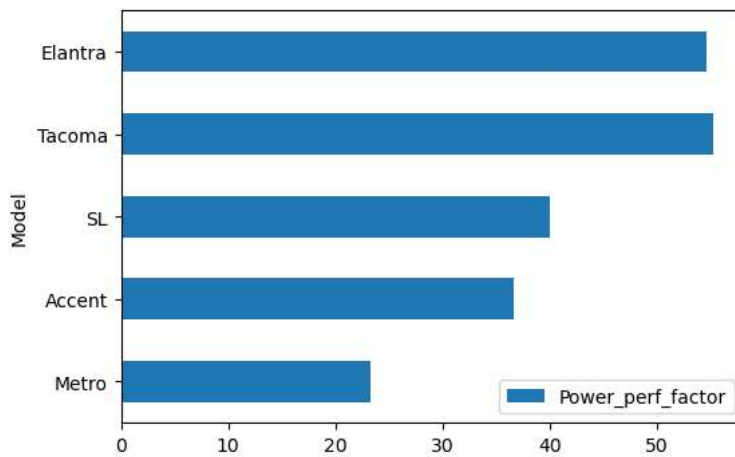


**Area chart**

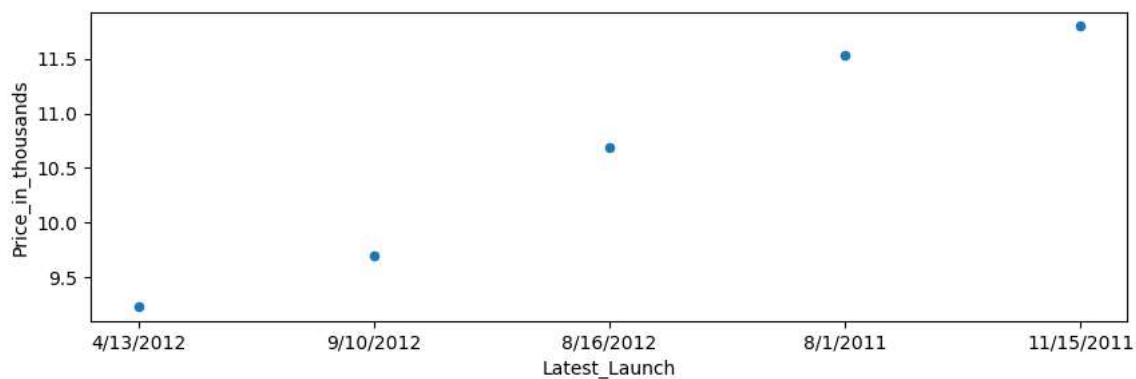
```
In [13]: data.plot.area('Manufacturer', 'Price_in_thousands', figsize=(6,3), cmap='Pastel2');
```

**Horizontal bar chart**

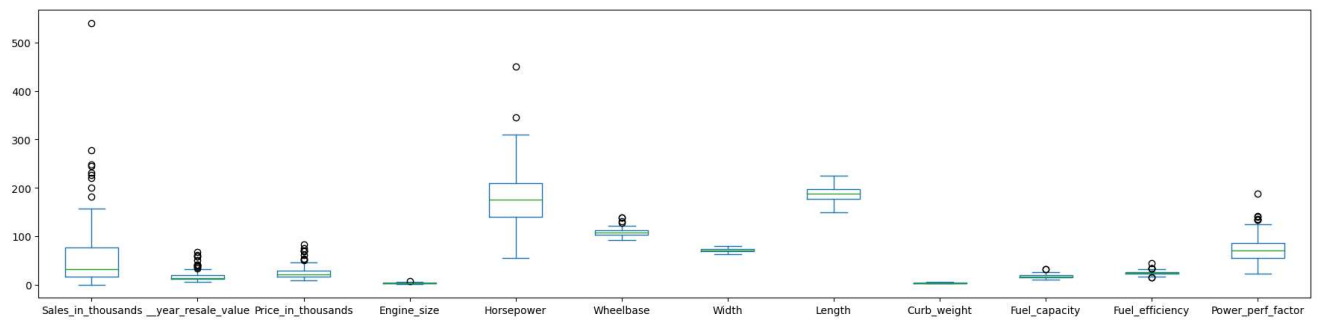
```
In [14]: data.sort_values(by='Price_in_thousands')[5].plot.barh('Model', 'Power_perf_factor', figsize=(6,4));
```

**Scatter plot**

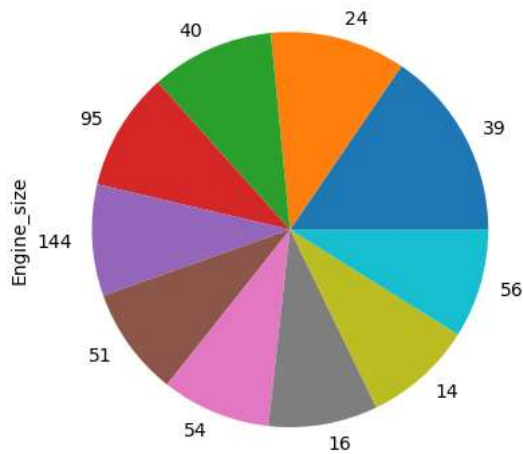
```
In [15]: data.sort_values(by='Price_in_thousands')[5].plot.scatter('Latest_Launch', 'Price_in_thousands', figsize=(10,3));
```



```
In [16]: data.plot.box(figsize=(22,5));
```

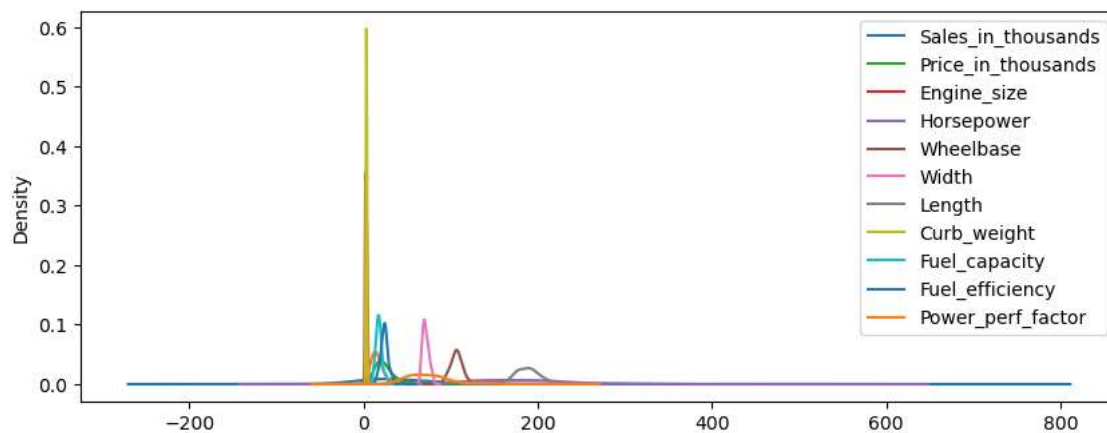


```
In [17]: data.sort_values(by='Engine_size', ascending=False)[:10].Engine_size.plot.pie();
```



### KDE Plot

```
In [18]: data.plot.kde(figsize=(10,3.8));
```

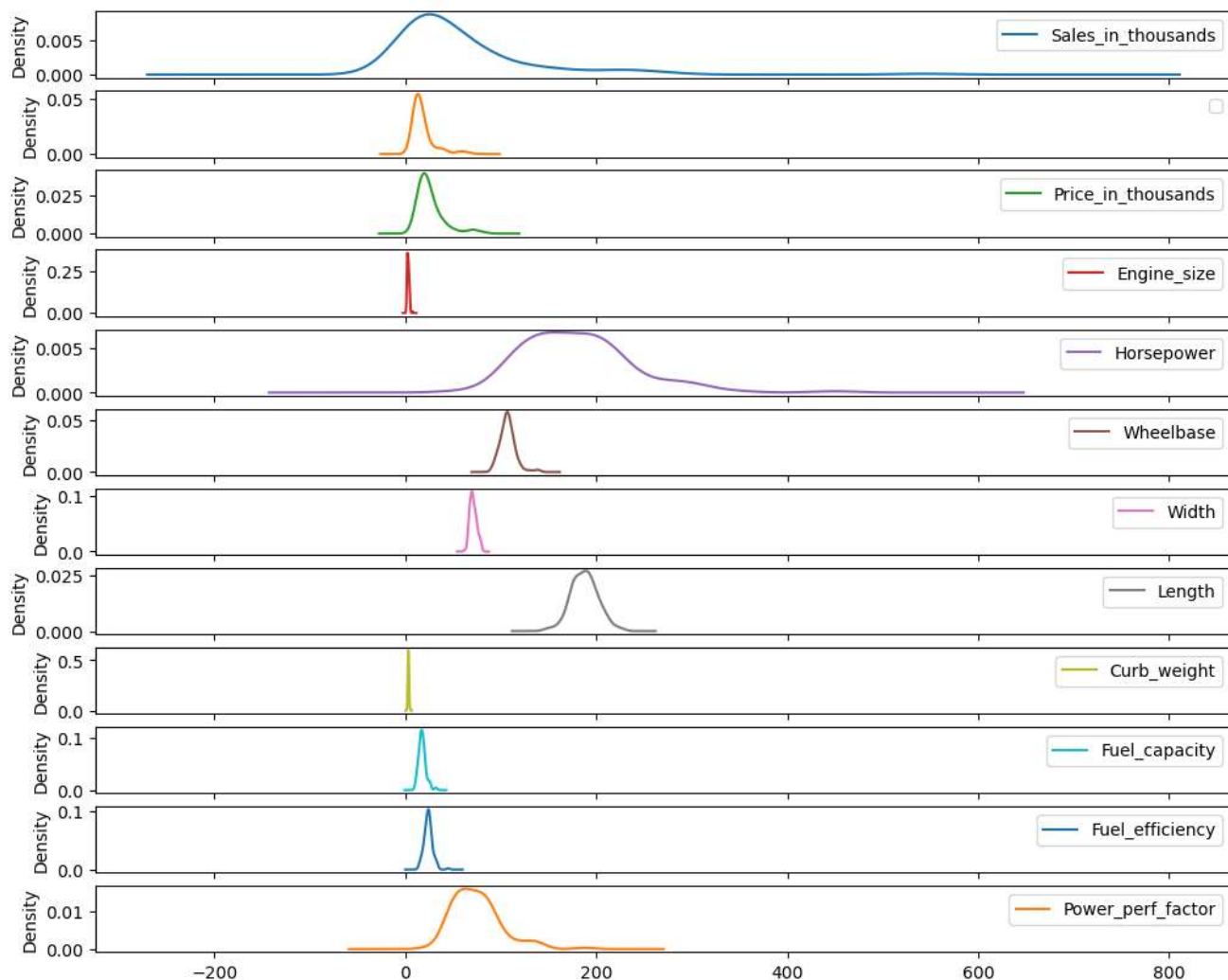


### Subplots



```
In [19]: data.plot.kde(figsize=(12,10), subplots = True);
```

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



## Hexbin plot

```
In [20]: data.plot.hexbin('Curb_weight', 'Fuel_capacity', gridsize=15, cmap='autumn');
```

