

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
In [2]: car=pd.read_csv("car-sales-extended.csv")
car
```

```
Out[2]:
```

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431	4	15323
1	BMW	Blue	192714	5	19943
2	Honda	White	84714	4	28343
3	Toyota	White	154365	4	13434
4	Nissan	Blue	181577	3	14043
...
995	Toyota	Black	35820	4	32042
996	Nissan	White	155144	3	5716
997	Nissan	Blue	66604	4	31570
998	Honda	White	215883	4	4001
999	Toyota	Blue	248360	4	12732

1000 rows × 5 columns

```
In [3]: len(car)
```

```
Out[3]: 1000
```

```
In [7]: car.index
```

```
Out[7]: RangeIndex(start=0, stop=1000, step=1)
```

```
In [9]: car.columns
```

```
Out[9]: Index(['Make', 'Colour', 'Odometer (KM)', 'Doors', 'Price'], dtype='object')
```

Describe Data

```
In [12]: car.describe()
```

Out[12]:

	Odometer (KM)	Doors	Price
count	1000.000000	1000.000000	1000.000000
mean	131073.708000	4.014000	16045.665000
std	68859.723885	0.379405	8630.794219
min	10148.000000	3.000000	2796.000000
25%	71238.000000	4.000000	9481.500000
50%	131202.000000	4.000000	14264.000000
75%	192372.750000	4.000000	20738.750000
max	249860.000000	5.000000	52458.000000

In [14]: `car.head(5)`

Out[14]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431	4	15323
1	BMW	Blue	192714	5	19943
2	Honda	White	84714	4	28343
3	Toyota	White	154365	4	13434
4	Nissan	Blue	181577	3	14043

In [16]: `car.tail(7)`

Out[16]:

	Make	Colour	Odometer (KM)	Doors	Price
993	Nissan	Black	162523	4	4696
994	BMW	Blue	163322	3	31666
995	Toyota	Black	35820	4	32042
996	Nissan	White	155144	3	5716
997	Nissan	Blue	66604	4	31570
998	Honda	White	215883	4	4001
999	Toyota	Blue	248360	4	12732

In [18]: `car.dtypes`

```
Out[18]: Make          object
Colour          object
Odometer (KM)    int64
Doors            int64
Price            int64
dtype: object
```

In [20]: `car.info()`

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 5 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   Make                  1000 non-null   object  
1   Colour                1000 non-null   object  
2   Odometer (KM)         1000 non-null   int64  
3   Doors                 1000 non-null   int64  
4   Price                 1000 non-null   int64  
dtypes: int64(3), object(2)  
memory usage: 39.2+ KB
```

In [22]: `car.mode()`

Out[22]:

	Make	Colour	Odometer (KM)	Doors	Price
0	Toyota	White	73869.0	4.0	3974
1	NaN	NaN	129188.0	NaN	4566
2	NaN	NaN	NaN	NaN	7813
3	NaN	NaN	NaN	NaN	7994
4	NaN	NaN	NaN	NaN	8260
5	NaN	NaN	NaN	NaN	10429
6	NaN	NaN	NaN	NaN	11648
7	NaN	NaN	NaN	NaN	12392
8	NaN	NaN	NaN	NaN	12402
9	NaN	NaN	NaN	NaN	13560
10	NaN	NaN	NaN	NaN	15154
11	NaN	NaN	NaN	NaN	16519
12	NaN	NaN	NaN	NaN	17076
13	NaN	NaN	NaN	NaN	17520
14	NaN	NaN	NaN	NaN	18919
15	NaN	NaN	NaN	NaN	28414

In [24]: `car.std`

```
Out[24]: <bound method DataFrame.std of
0      Honda  White      35431      4  15323
1        BMW   Blue     192714      5  19943
2      Honda  White      84714      4  28343
3     Toyota  White     154365      4  13434
4     Nissan   Blue     181577      3  14043
..      ...    ...      ...    ...    ...
995  Toyota  Black      35820      4  32042
996  Nissan  White     155144      3   5716
997  Nissan   Blue      66604      4  31570
998  Honda   White     215883      4   4001
999  Toyota   Blue     248360      4  12732

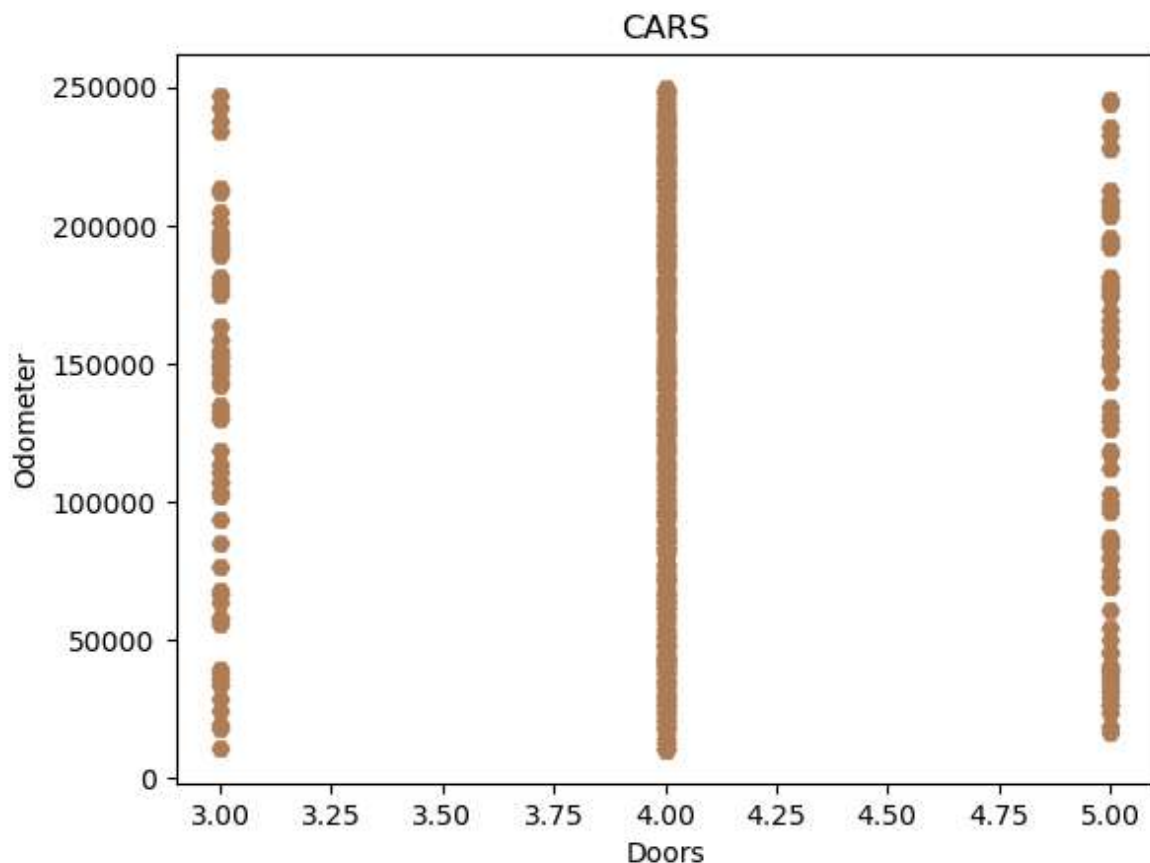
[1000 rows x 5 columns]>
```

```
In [26]: car.var
```

```
Out[26]: <bound method DataFrame.var of
0      Honda  White      35431      4  15323
1        BMW   Blue     192714      5  19943
2      Honda  White      84714      4  28343
3     Toyota  White     154365      4  13434
4     Nissan   Blue     181577      3  14043
..      ...    ...      ...    ...    ...
995  Toyota  Black      35820      4  32042
996  Nissan  White     155144      3   5716
997  Nissan   Blue      66604      4  31570
998  Honda   White     215883      4   4001
999  Toyota   Blue     248360      4  12732

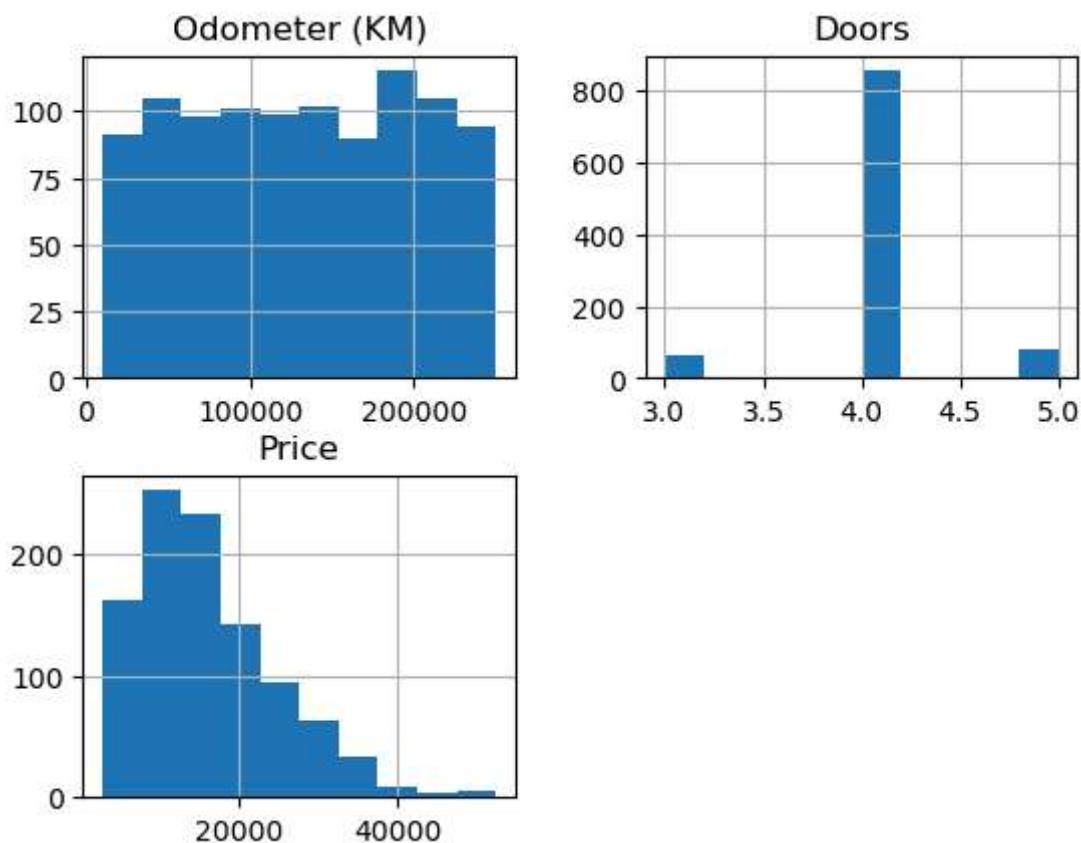
[1000 rows x 5 columns]>
```

```
In [37]: x=car['Doors']
y=car['Odometer (KM)']
plt.scatter(x,y,marker='H',c='#B17F59')
plt.title("CARS")
plt.xlabel("Doors")
plt.ylabel("Odometer")
plt.show()
```



```
In [39]: car.hist()
```

```
Out[39]: array([[<Axes: title={'center': 'Odometer (KM)'}>,
                  <Axes: title={'center': 'Doors'}>],
               [<Axes: title={'center': 'Price'}>, <Axes: >]], dtype=object)
```

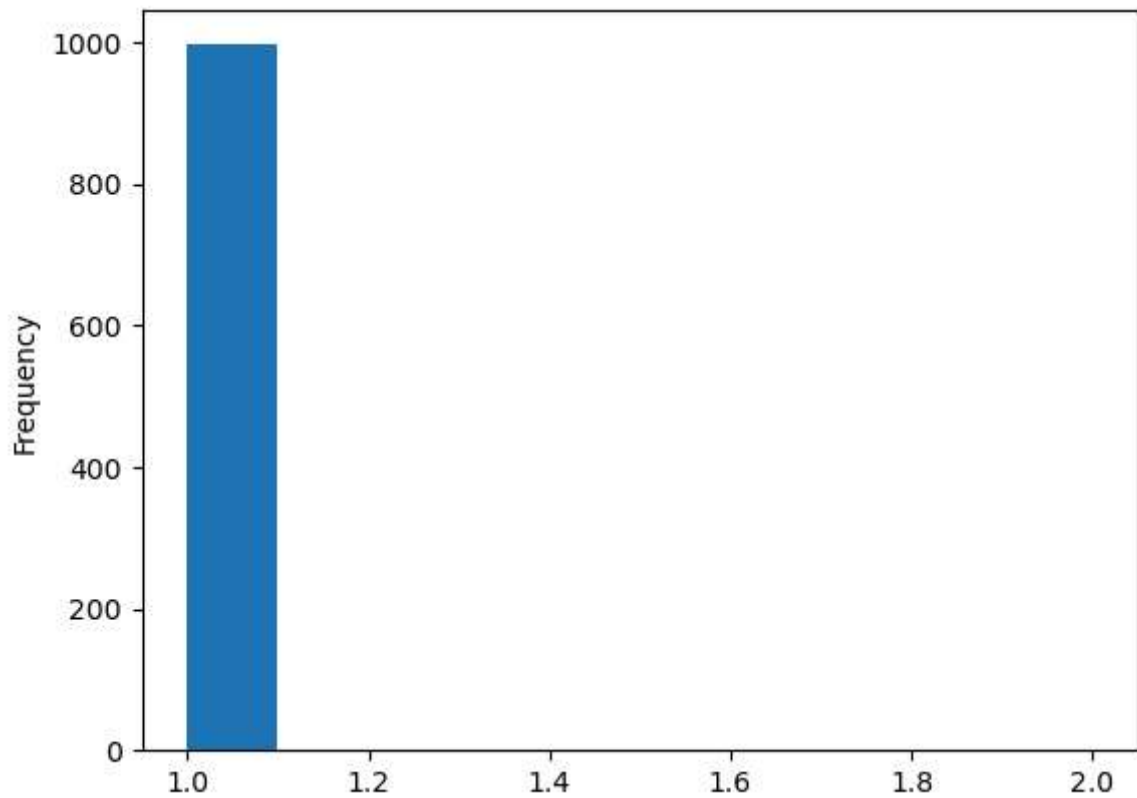


```
In [47]: car['Colour'].value_counts()
```

```
Out[47]: Colour
White    407
Blue     321
Black     99
Red       94
Green     79
Name: count, dtype: int64
```

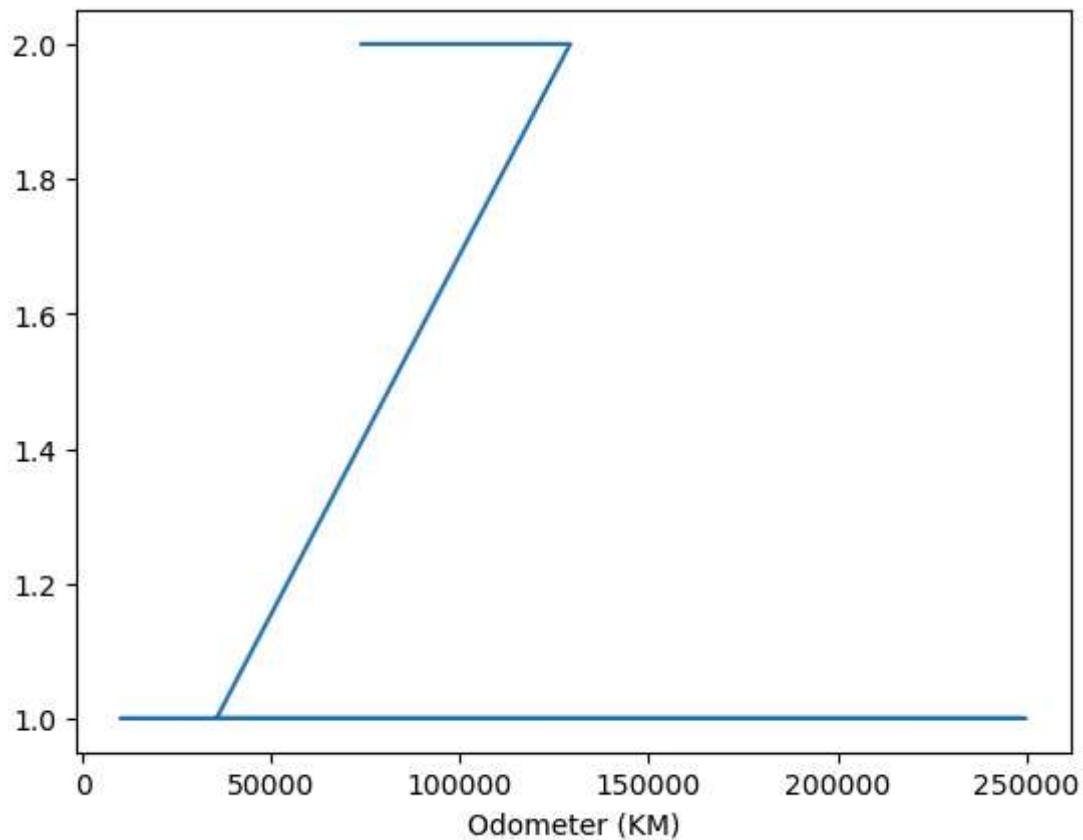
```
In [65]: car['Odometer (KM)'].value_counts().plot.hist()
```

```
Out[65]: <Axes: ylabel='Frequency'>
```



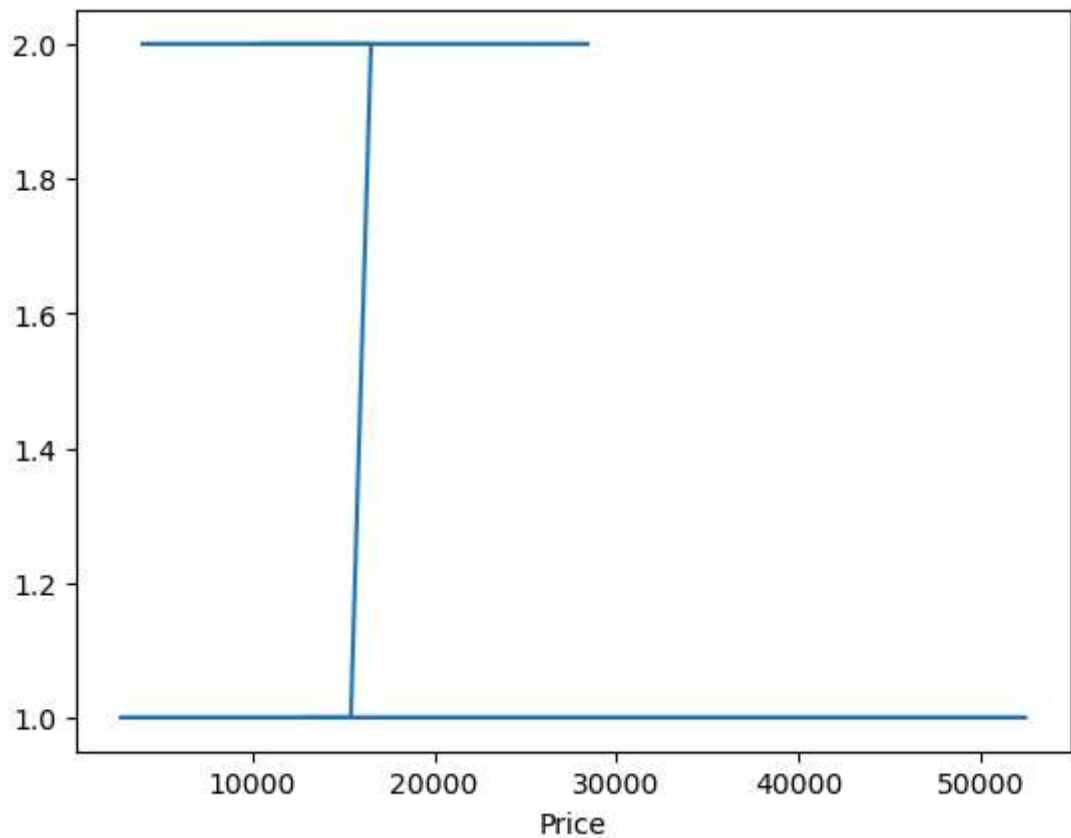
```
In [69]: car['Odometer (KM)'].value_counts().plot()
```

```
Out[69]: <Axes: xlabel='Odometer (KM)'>
```



```
In [85]: car['Price'].value_counts().plot()
```

```
Out[85]: <Axes: xlabel='Price'>
```

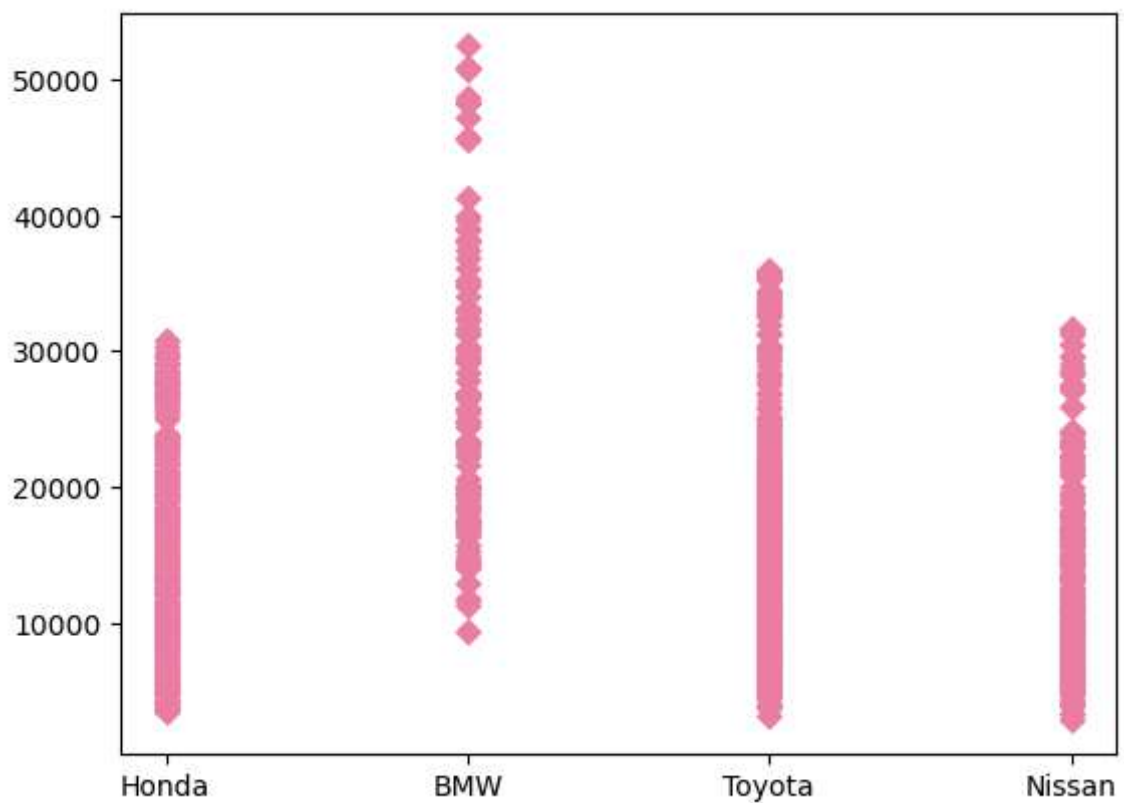


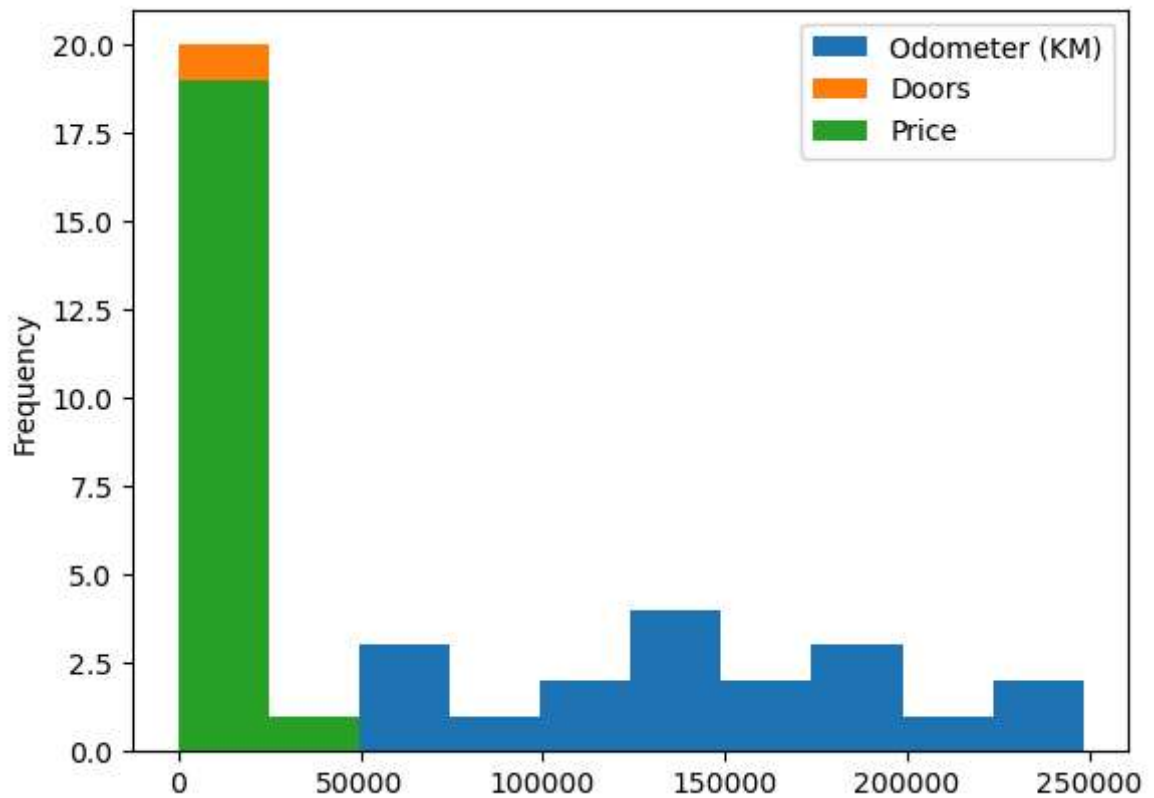
loc & iloc

```
In [92]: car.loc[72]
```

```
Out[92]: Make          Honda  
Colour         White  
Odometer (KM)   116986  
Doors           4  
Price           5648  
Name: 72, dtype: object
```

```
In [120... x=car['Make']  
y=car['Price']  
plt.scatter(x,y,marker='D',c='#EC7FA9')  
car.iloc[75:95].plot.hist()  
plt.show()
```





Conditional Filtering

`==, >, >=, <, <=, !=`

In [127... `car[car['Doors']!=4]`

Out[127...

	Make	Colour	Odometer (KM)	Doors	Price
1	BMW	Blue	192714	5	19943
4	Nissan	Blue	181577	3	14043
17	BMW	White	194189	5	17177
18	Nissan	White	67991	3	9109
33	Nissan	Green	153554	3	9780
...
969	BMW	Blue	45207	5	35254
971	BMW	Black	178164	3	24891
978	BMW	White	85739	5	48419
994	BMW	Blue	163322	3	31666
996	Nissan	White	155144	3	5716

144 rows × 5 columns

In [129... `car[car['Odometer (KM)']<=194189]`

Out[129...

	Make	Colour	Odometer (KM)	Doors	Price
0	Honda	White	35431	4	15323
1	BMW	Blue	192714	5	19943
2	Honda	White	84714	4	28343
3	Toyota	White	154365	4	13434
4	Nissan	Blue	181577	3	14043
...
993	Nissan	Black	162523	4	4696
994	BMW	Blue	163322	3	31666
995	Toyota	Black	35820	4	32042
996	Nissan	White	155144	3	5716
997	Nissan	Blue	66604	4	31570

767 rows × 5 columns

Divide new col with applying some maths

In [159...

```
car['New']=car['Price']/car['Doors']
car
```

Out[159...

	Make	Odometer (KM)	Doors	Price	New
0	Honda	35431	4	15323	3830.750000
1	BMW	192714	5	19943	3988.600000
2	Honda	84714	4	28343	7085.750000
3	Toyota	154365	4	13434	3358.500000
4	Nissan	181577	3	14043	4681.000000
...
995	Toyota	35820	4	32042	8010.500000
996	Nissan	155144	3	5716	1905.333333
997	Nissan	66604	4	31570	7892.500000
998	Honda	215883	4	4001	1000.250000
999	Toyota	248360	4	12732	3183.000000

1000 rows × 5 columns

cross tab

In [162...

```
car.nunique()
```

```
Out[162... Make          4
Odometer (KM)  998
Doors          3
Price          984
New            986
dtype: int64
```

```
In [168... pd.crosstab(car['Price'],car['Make'])
```

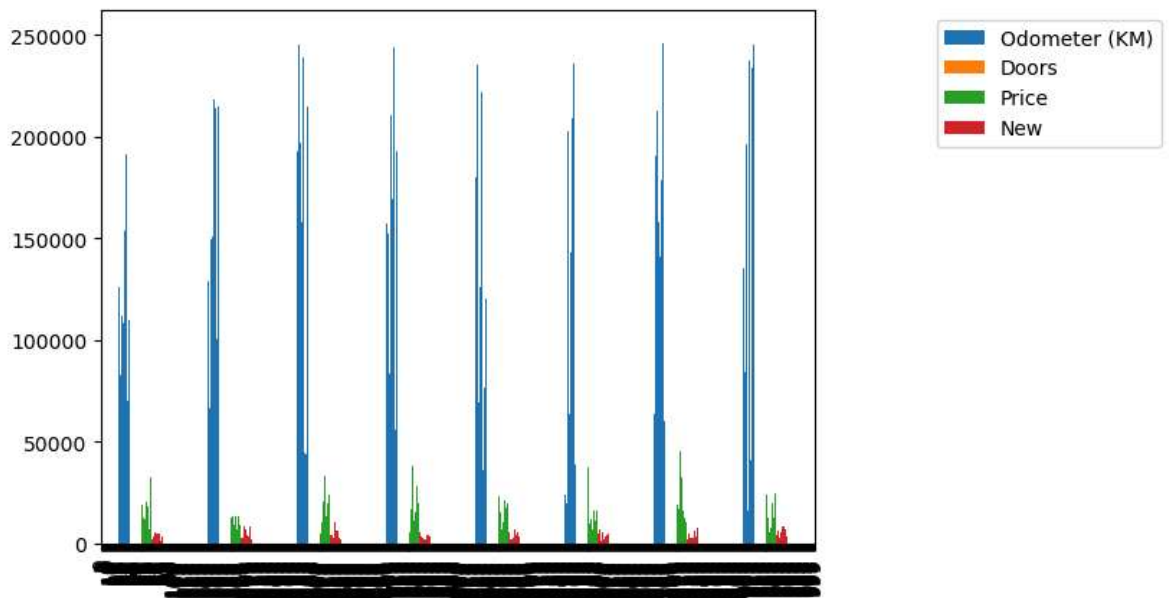
```
Out[168... Make  BMW  Honda  Nissan  Toyota

Price
2796    0    0    1    0
3125    0    0    0    1
3247    0    0    1    0
3300    0    0    1    0
3509    0    1    0    0
...    ...    ...    ...    ...
48419    1    0    0    0
48686    1    0    0    0
50822    1    0    0    0
50868    1    0    0    0
52458    1    0    0    0
```

984 rows × 4 columns

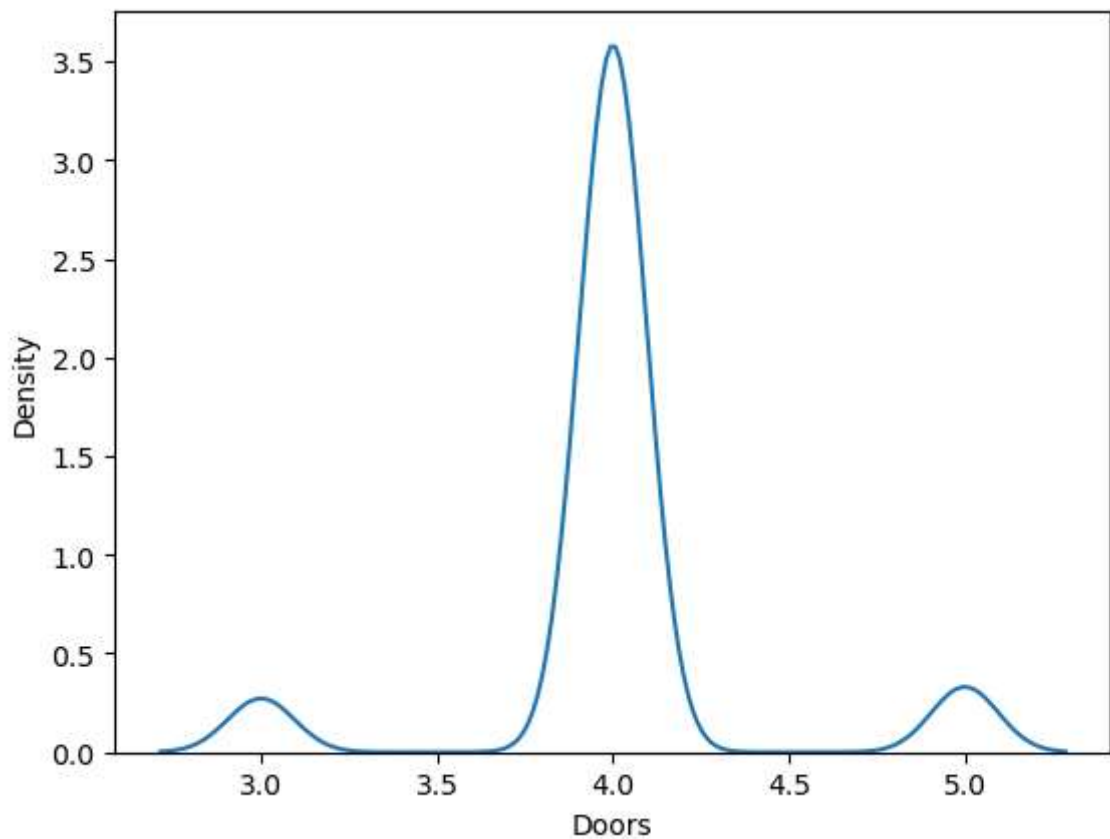
Sp. Graphs

```
In [183... car.plot(kind='bar')
plt.legend(bbox_to_anchor=(1.5,1))
plt.show()
```



In [195... `sns.kdeplot(car["Doors"])`

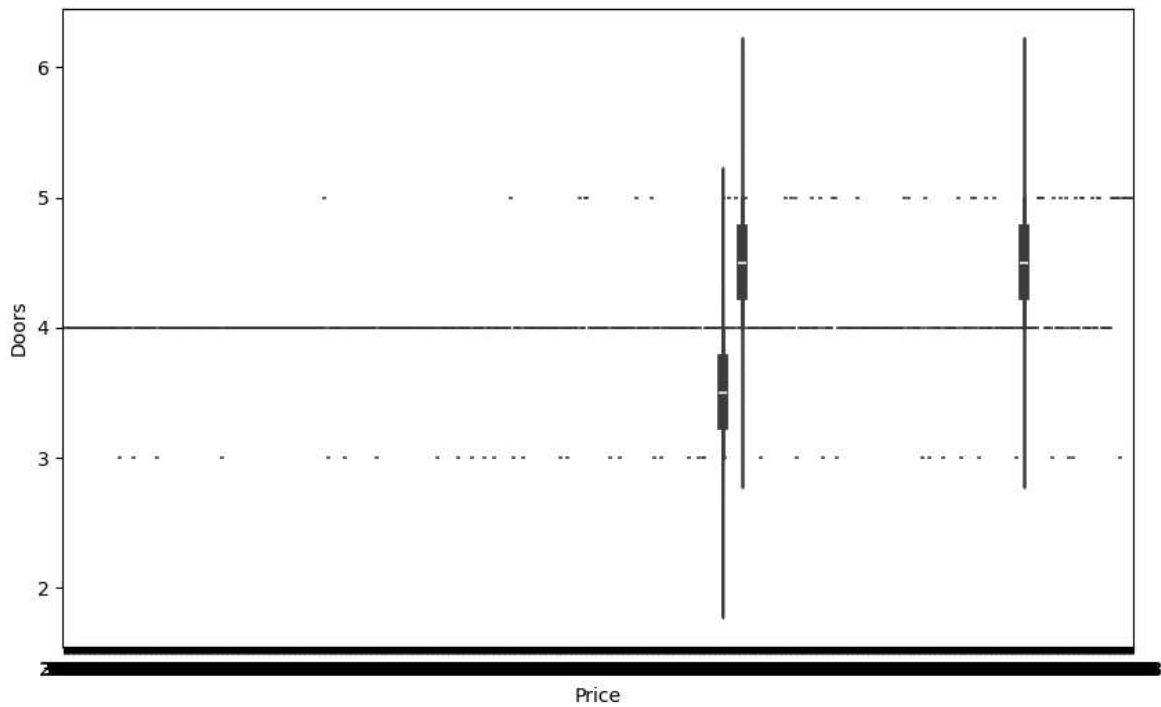
Out[195... `<Axes: xlabel='Doors', ylabel='Density'>`



Violin Chart

In [200... `plt.figure(figsize=(10,6))`
`sns.violinplot (x=car["Price"],y=car["Doors"],data=car)`

Out[200... `<Axes: xlabel='Price', ylabel='Doors'>`



```
In [201... from pandas.plotting import scatter_matrix
scatter_matrix(car, alpha=0.2, figsize=(10,10), diagonal='kde')
```

```
Out[201... array([[<Axes: xlabel='Odometer (KM)', ylabel='Odometer (KM)'>,
      <Axes: xlabel='Doors', ylabel='Odometer (KM)'>,
      <Axes: xlabel='Price', ylabel='Odometer (KM)'>,
      <Axes: xlabel='New', ylabel='Odometer (KM)'>],
      [<Axes: xlabel='Odometer (KM)', ylabel='Doors'>,
      <Axes: xlabel='Doors', ylabel='Doors'>,
      <Axes: xlabel='Price', ylabel='Doors'>,
      <Axes: xlabel='New', ylabel='Doors'>],
      [<Axes: xlabel='Odometer (KM)', ylabel='Price'>,
      <Axes: xlabel='Doors', ylabel='Price'>,
      <Axes: xlabel='Price', ylabel='Price'>,
      <Axes: xlabel='New', ylabel='Price'>],
      [<Axes: xlabel='Odometer (KM)', ylabel='New'>,
      <Axes: xlabel='Doors', ylabel='New'>,
      <Axes: xlabel='Price', ylabel='New'>,
      <Axes: xlabel='New', ylabel='New'>]], dtype=object)
```

