

In [1]:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import seaborn as sns
5 import os
6

```

In [5]:

```

1 os.chdir('C:/Users/MSCT/Desktop/FDS Niraj/Datasets-20220805T064950Z-001 (1)/Data
2 cars_data=pd.read_csv('Toyota.csv',index_col=0)
3 cars_data.head()
4
5

```

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23.0	46986	Diesel	90	1.0	0	2000	three	1165
1	13750	23.0	72937	Diesel	90	1.0	0	2000	3	1165
2	13950	24.0	41711	Diesel	90	NaN	0	2000	3	1165
3	14950	26.0	48000	Diesel	90	0.0	0	2000	3	1165
4	13750	30.0	38500	Diesel	90	0.0	0	2000	3	1170

In [6]:

```

1 cars_data.dropna(axis=0,inplace=True)
2 cars_data.size

```

11110

In [7]:

```

1 os.chdir('C:/Users/MSCT/Desktop/FDS Niraj/Datasets-20220805T064950Z-001 (1)/Data
2 cars_data=pd.read_csv('Toyota.csv',index_col=0, na_values=['??','????'])
3 cars_data.size

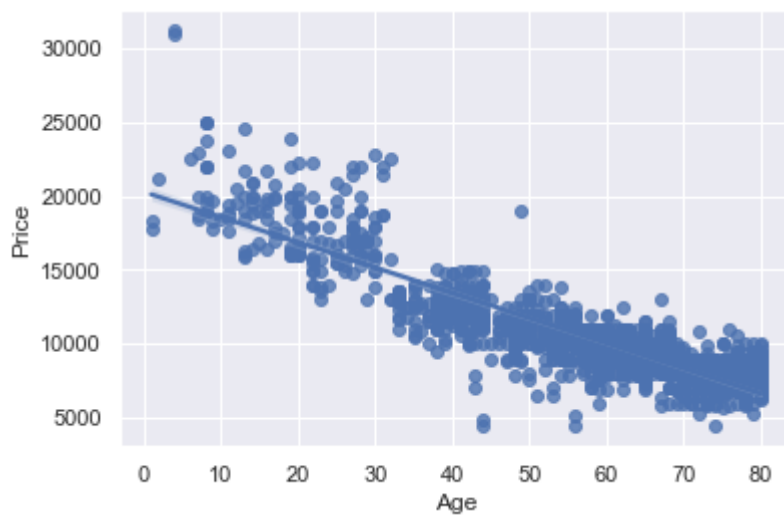
```

14360

In [9]:

```
1 import seaborn as sns
2 sns.set(style='darkgrid')
3 sns.regplot(x=cars_data['Age'],y=cars_data['Price'])
4 #It estimate
```

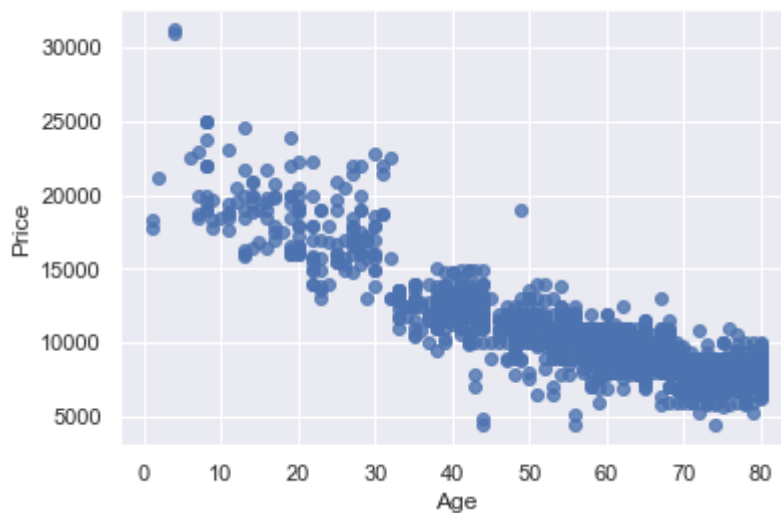
<AxesSubplot:xlabel='Age', ylabel='Price'>



In [11]:

```
1 #scatter plot of price vs Age without regression
2 sns.regplot(x=cars_data['Age'],y=cars_data['Price'],fit_reg=False)
```

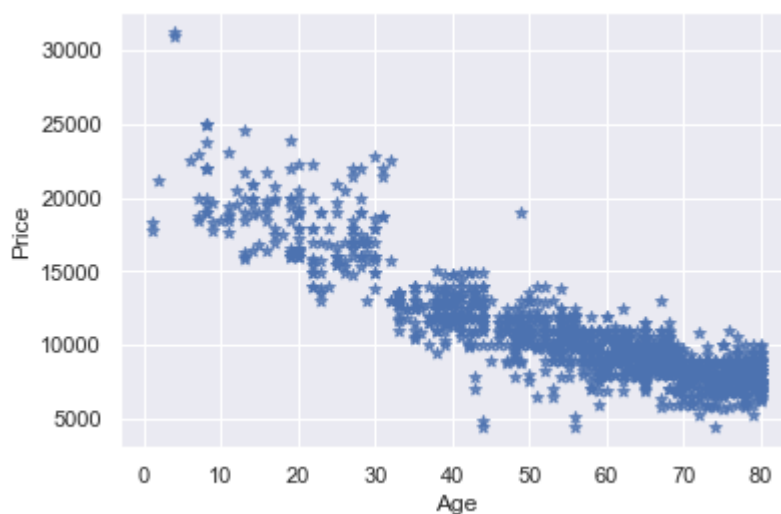
<AxesSubplot:xlabel='Age', ylabel='Price'>



In [12]:

```
1 sns.regplot(x=cars_data['Age'],y=cars_data['Price'],marker='*',fit_reg=False)
2 #scatter plot of price vs age by customizing the apperance of markers
```

<AxesSubplot:xlabel='Age', ylabel='Price'>



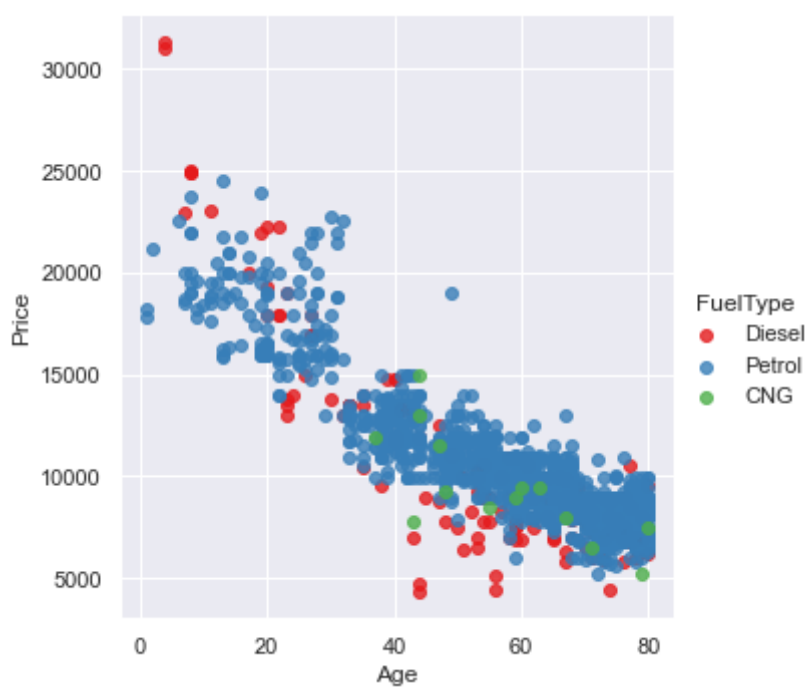
```
In [16]:
```

```
1 #using hue parameter
```

```
2
```

```
3 sns.lmplot(x='Age',y='Price',data=cars_data,fit_reg=False,hue="FuelType",legend=Tr
```

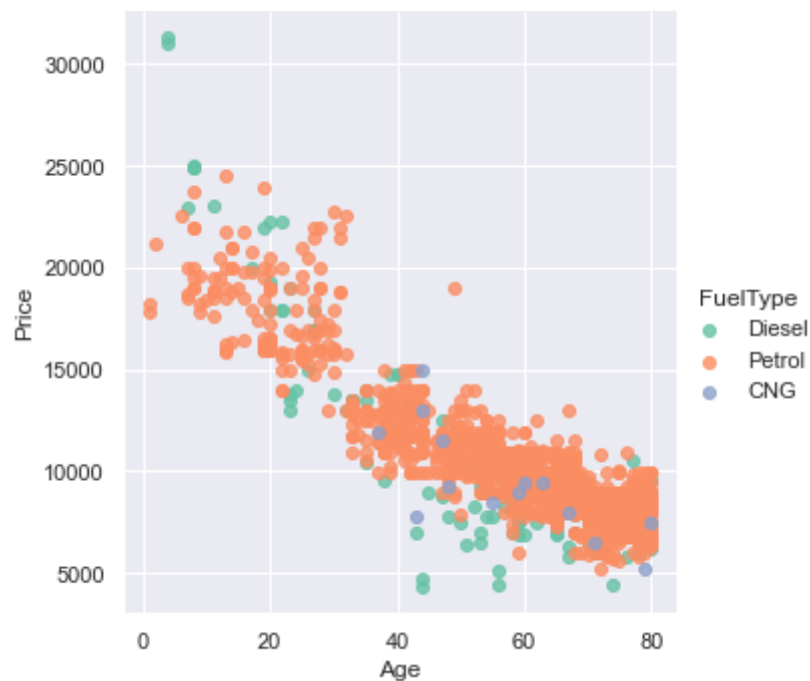
<seaborn.axisgrid.FacetGrid at 0x27303afdc40>



```
In [17]:
```

```
1 sns.lmplot(x='Age',y='Price',data=cars_data,fit_reg=False,hue="FuelType",legend=Tr
```

<seaborn.axisgrid.FacetGrid at 0x27303adefd0>

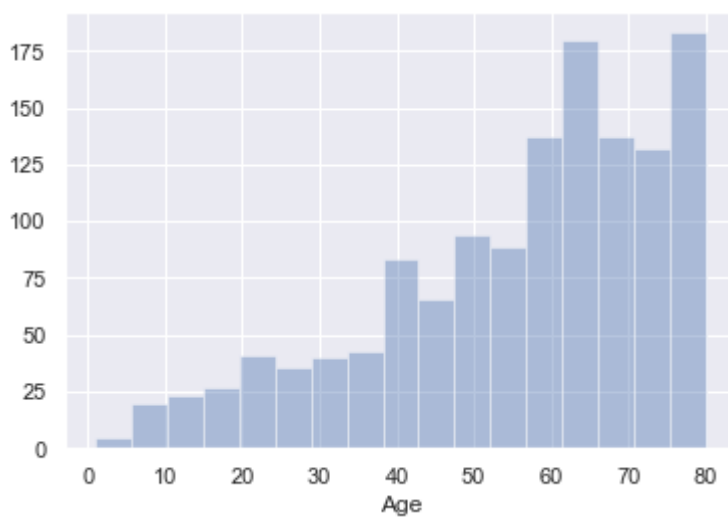


In [23]:

```
1 sns.distplot(cars_data['Age'],kde=False)
```

C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function. Please adapt your code to use either `displot` (a figure-level function with similar semantics to `distplot`) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Age'>

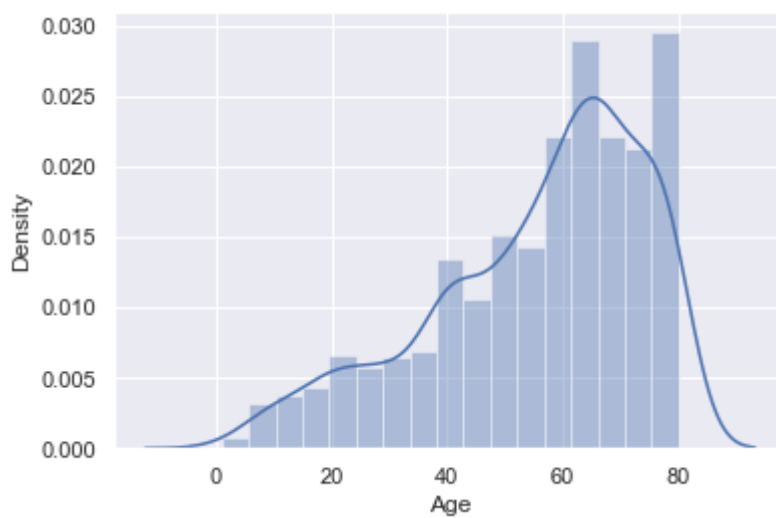


In [22]:

```
1 sns.distplot(cars_data['Age'])
```

C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function. Please adapt your code to use either `displot` (a figure-level function with similar semantics to `distplot`) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Age', ylabel='Density'>

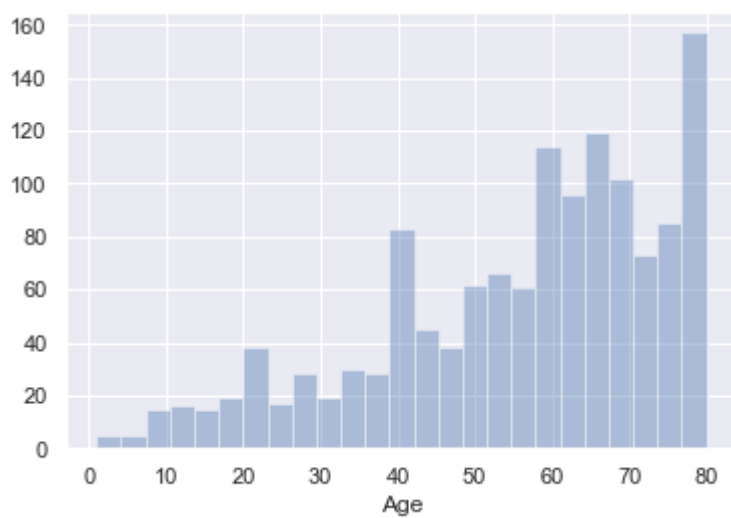


In [24]:

```
1 sns.distplot(cars_data['Age'],kde=False, bins=25)
```

C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function. Please adapt your code to use either `displot` (a figure-level function with similar semantics to `distplot`) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

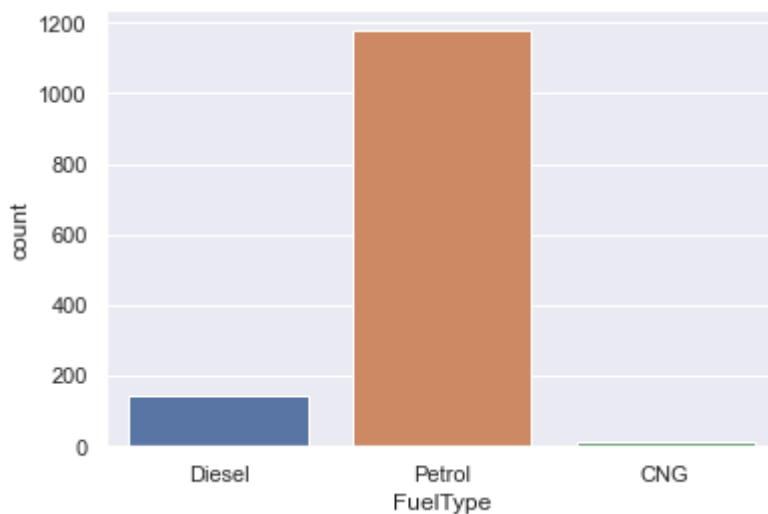
<AxesSubplot:xlabel='Age'>



In [25]:

```
1 sns.countplot(x='FuelType', data=cars_data)
```

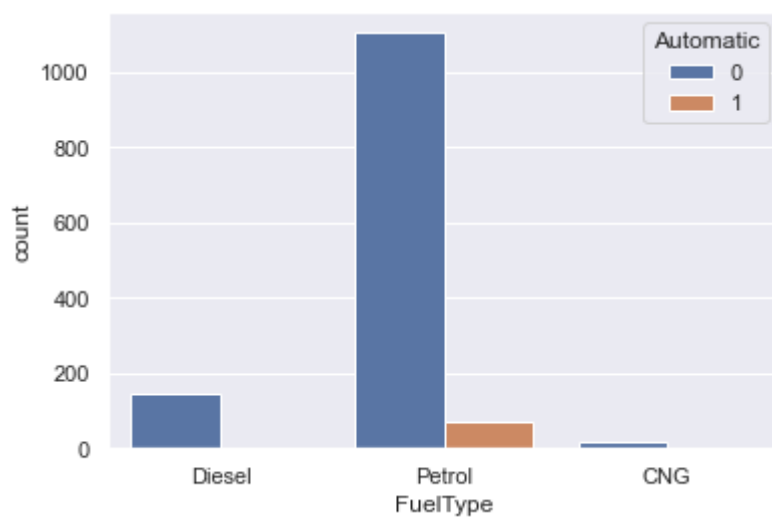
<AxesSubplot:xlabel='FuelType', ylabel='count'>



In [26]:

```
1 sns.countplot(x='FuelType', data=cars_data,hue='Automatic')
2 #using hue function
```

<AxesSubplot:xlabel='FuelType', ylabel='count'>



In [27]:

```
1 #crosstab=new table with my given value
2 #making frequency table of car fuel type
3 pd.crosstab(index=cars_data['Automatic'],columns=cars_data['FuelType'],dropna=True)
```

FuelType	CNG	Diesel	Petrol
----------	-----	--------	--------

Automatic			
-----------	--	--	--

0	15	144	1104
---	----	-----	------

1	0	0	73
---	---	---	----

In [28]:

```
1 pd.crosstab(index=cars_data['Automatic'],columns=cars_data['FuelType'])
```

FuelType	CNG	Diesel	Petrol
----------	-----	--------	--------

Automatic			
-----------	--	--	--

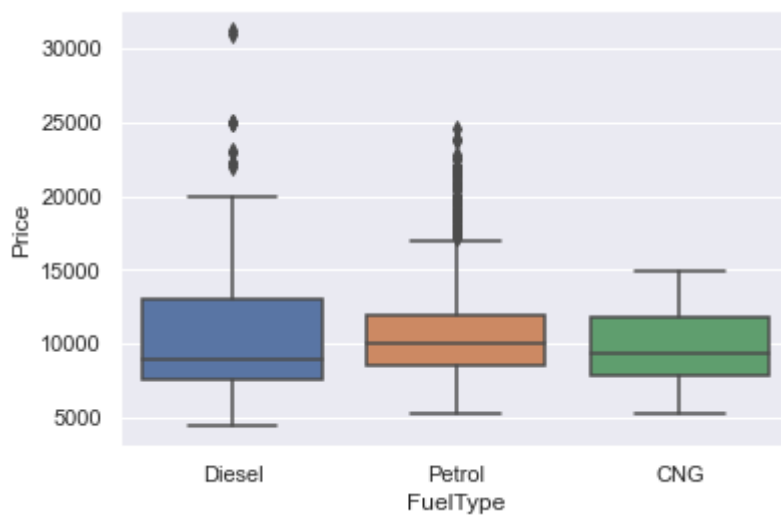
0	15	144	1104
---	----	-----	------

1	0	0	73
---	---	---	----

In [29]:

```
1 sns.boxplot(x=cars_data['FuelType'],y=cars_data['Price'])
```

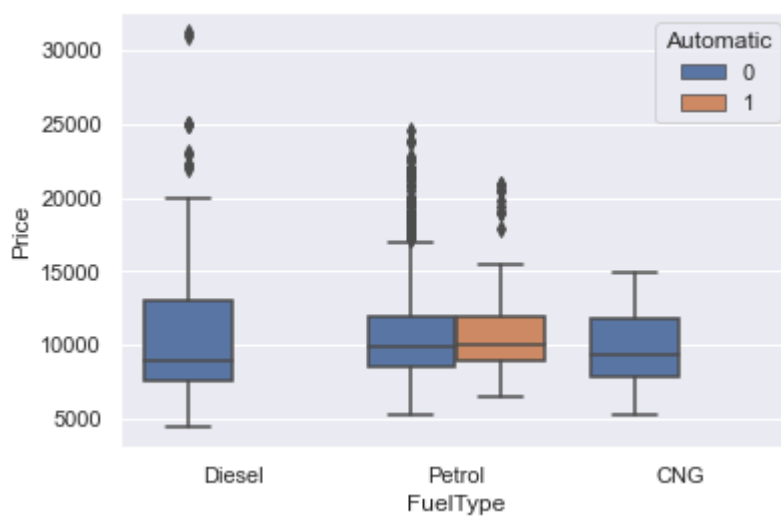
<AxesSubplot:xlabel='FuelType', ylabel='Price'>



In [30]:

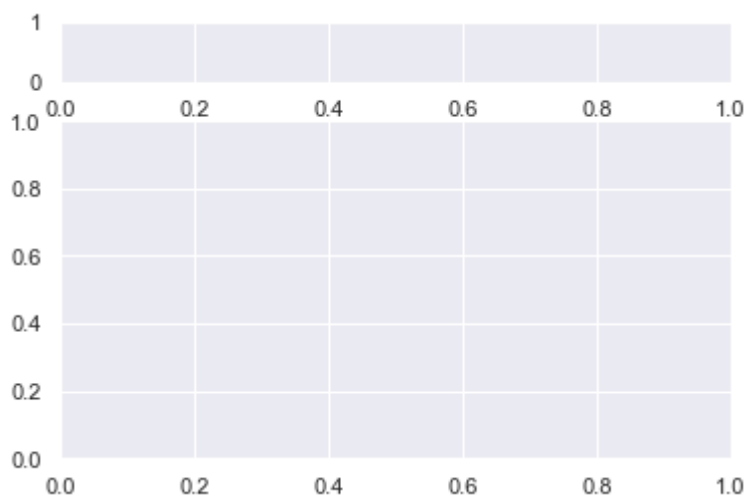
```
1 sns.boxplot(x=cars_data['FuelType'],y=cars_data['Price'],hue='Automatic',data=cars)
```

<AxesSubplot:xlabel='FuelType', ylabel='Price'>



In [33]:

```
1 f,(ax_box,ax_hist)=plt.subplots(2,gridspec_kw={'height_ratios':(.15, .85)})  
2 #2 graph
```



In [35]:

```

1 f,(ax_box,ax_hist)=plt.subplots(2,gridspec_kw={'height_ratios':(.15, .85)})
2 sns.boxplot(cars_data['Price'],ax=ax_box)
3 sns.displot(cars_data['Price'],ax=ax_hist,kde=False)

```

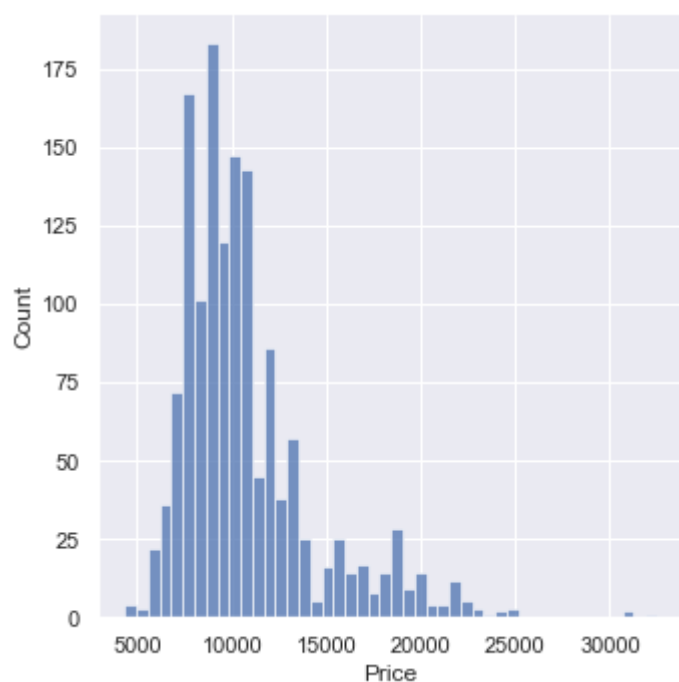
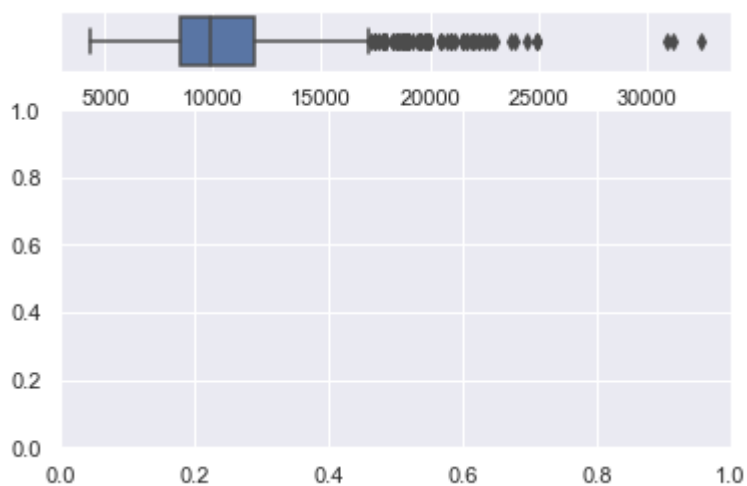
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword argument: 'data'. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit `data` argument will result in an error or misinterpretation.

warnings.warn(

C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:2211: UserWarning: `displot` is a figure-level function. It does not accept the `ax` parameter. You may wish to try `histplot`.

warnings.warn(msg, UserWarning)

<seaborn.axisgrid.FacetGrid at 0x27301363100>



In [37]:

```
1 sns.pairplot(cars_data,kind='scatter',hue='FuelType',diag_kws={'bw': 0.1})
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
C:\Users\MSCIT\anaconda3\lib\site-packages\seaborn\distributions.py:1699: FutureWarning: The `bw` parameter is method` and `bw_adjust`. Using 0.1 for `bw_method`, but please see the docs for the new parameters and update ; warnings.warn(msg, FutureWarning)
```

```
<seaborn.axisgrid.PairGrid at 0x27304810610>
```



In [38]:

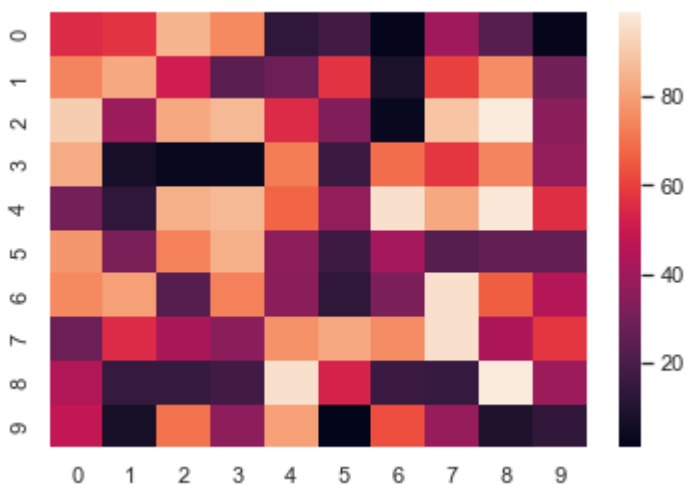
```
1 data=np.random.randint(1,100,size=(10,10))
2 print('The data to be plotted: \n')
3 print(data)
```

The data to be plotted:

```
[[55 57 85 75 13 18  2 40 23  2]
 [74 82 51 24 28 57  8 60 76 29]
 [91 39 82 86 55 33  3 89 99 35]
 [83  7  3  3 72 16 69 58 74 37]
 [30 13 84 86 67 37 96 82 98 56]
 [78 32 73 84 36 17 41 23 26 26]
 [75 80 23 73 35 13 32 96 66 45]
 [28 55 42 35 77 82 76 96 43 58]
 [44 15 15 18 96 53 16 15 99 39]
 [48  7 70 36 80  1 63 38  9 14]]
```

In [39]:

```
1 #plotting Heatmap
2 hm=sns.heatmap(data=data)
3 plt.show()
```



In [40]:

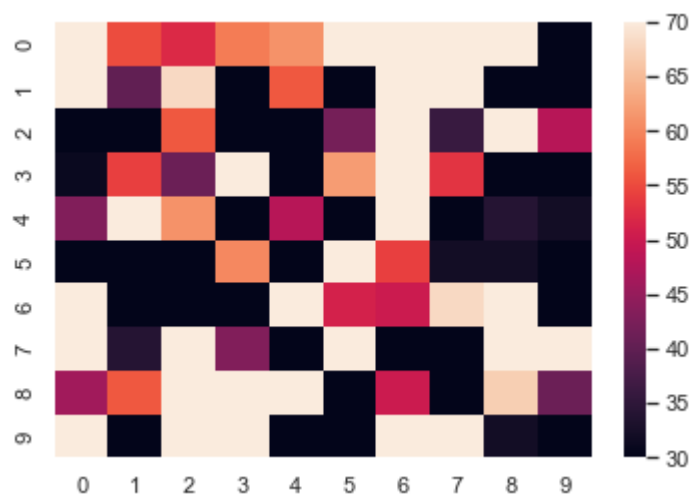
```
1 data=np.random.randint(1,100,size=(10,10))
2 print('The data to be plotted:')
3 print(data)
```

The data to be plotted:

```
[[79 55 52 59 61 75 98 97 81  7]
 [72 40 68  9 56 18 77 99  6 26]
 [ 9 18 56 11  2 42 92 36 74 48]
 [31 54 41 73 20 62 95 53  4 22]
 [43 74 61 17 48 14 90 21 34 32]
 [ 7 22 18 60 30 77 54 32 32 24]
 [99 27 27 18 77 51 50 68 90 16]
 [77 34 85 43 18 71 15 10 93 72]
 [46 56 79 89 89 12 50  1 67 41]
 [96 26 88 79 24 23 93 83 32 17]]
```

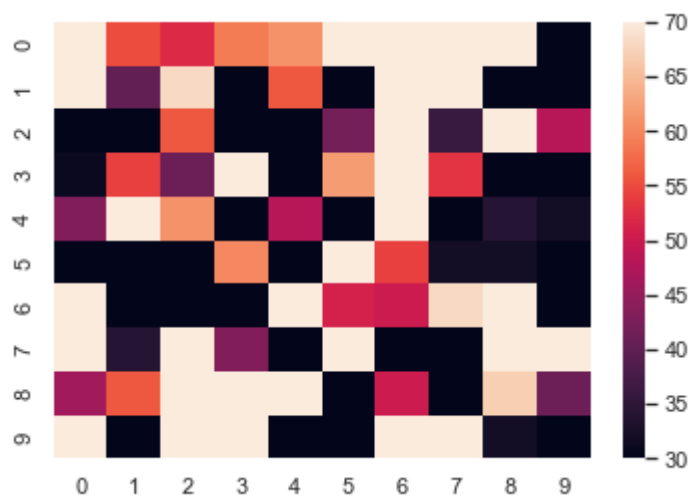
In [42]:

```
1 hm=sns.heatmap(data=data,
2                 vmin='30',vmax='70')
3 plt.show()
```




```
In [43]:
```

```
1 hm=sns.heatmap(data=data,  
2                 vmin=30,vmax=70)  
3 plt.show()
```



In [46]:

```

1  #setting the parameter values
2  #To change the color of heatmap
3
4  cmap='tab20'
5  center=0
6
7  #setting the parametr value
8  annot=True
9
10 #plotting the heatmap
11 hm=sns.heatmap(data=data,cmap=cmap,annot=annot)
12 plt.show()
13
14

```



In []:

1