

In [1]:

```
import pandas as pd
```

In [2]:

```
df1=pd.read_csv('ToyotaCorolla.csv')
```

In [3]:

```
df1
```

Out[3]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	3	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	1	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165
4	13750	30	38500	Diesel	90	0	0	2000	3	1170
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015
1435	6950	76	1	Petrol	110	0	0	1600	5	1114

1436 rows × 10 columns

In [7]:

```
df1.copy(deep=True)
```

Out[7]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	3	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	1	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165
4	13750	30	38500	Diesel	90	0	0	2000	3	1170
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015
1435	6950	76	1	Petrol	110	0	0	1600	5	1114

1436 rows × 10 columns

In [11]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Price       1436 non-null  int64
1   Age         1436 non-null  int64
2   KM          1436 non-null  int64
3   FuelType    1436 non-null  object
4   HP          1436 non-null  int64
5   MetColor    1436 non-null  int64
6   Automatic   1436 non-null  int64
7   CC          1436 non-null  int64
8   Doors       1436 non-null  int64
9   Weight      1436 non-null  int64
dtypes: int64(9), object(1)
memory usage: 112.3+ KB
```

In [15]:

```
df1.shape
```

Out[15]:

(1436, 10)

In [16]:

```
df1.value_counts()
```

Out[16]:

Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
24950	8	13253	Diesel	116	1	0	2000	5	1320
4350	44	158320	Diesel	69	0	0	1800	5	1110
10950	54	71725	Petrol	110	1	0	1600	4	1035
	57	80470	Petrol	110	0	0	1600	5	1085
		61682	Petrol	110	0	0	1600	3	1050
..									
8950	54	149329	Diesel	72	1	0	2000	5	1135
		64000	Petrol	97	1	0	1400	3	1025
		61000	Diesel	69	0	0	2000	5	1140
	53	115113	Petrol	97	1	0	1400	3	1025
32500	4	1	Diesel	116	0	0	2000	5	1480
Length: 1435, dtype: int64									

In [17]:

```
df1.describe()
```

Out[17]:

	Price	Age	KM	HP	MetColor	Automatic	
count	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	14
mean	10730.824513	55.947075	68533.259749	101.502089	0.674791	0.055710	15
std	3626.964585	18.599988	37506.448872	14.981080	0.468616	0.229441	1
min	4350.000000	1.000000	1.000000	69.000000	0.000000	0.000000	13
25%	8450.000000	44.000000	43000.000000	90.000000	0.000000	0.000000	14
50%	9900.000000	61.000000	63389.500000	110.000000	1.000000	0.000000	16
75%	11950.000000	70.000000	87020.750000	110.000000	1.000000	0.000000	16
max	32500.000000	80.000000	243000.000000	192.000000	1.000000	1.000000	20



In [20]:

```
df1.head(10)
```

Out[20]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	3	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	1	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165
4	13750	30	38500	Diesel	90	0	0	2000	3	1170
5	12950	32	61000	Diesel	90	0	0	2000	3	1170
6	16900	27	94612	Diesel	90	1	0	2000	3	1245
7	18600	30	75889	Diesel	90	1	0	2000	3	1245
8	21500	27	19700	Petrol	192	0	0	1800	3	1185
9	12950	23	71138	Diesel	69	0	0	1900	3	1105

In [21]:

```
df1.tail(2)
```

Out[21]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015
1435	6950	76	1	Petrol	110	0	0	1600	5	1114

In [31]:

```
df1.loc[:, "Price"]
```

Out[31]:

```
0      13500
1      13750
2      13950
3      14950
4      13750
...
1431     7500
1432    10845
1433     8500
1434     7250
1435     6950
Name: Price, Length: 1436, dtype: int64
```

In [32]:

```
df1.loc[:, "Age"]
```

Out[32]:

```
0      23
1      23
2      24
3      26
4      30
...
1431   69
1432   72
1433   71
1434   70
1435   76
Name: Age, Length: 1436, dtype: int64
```

In [33]:

```
df1.loc[:, "KM"]
```

Out[33]:

```
0      46986
1      72937
2      41711
3      48000
4      38500
...
1431   20544
1432   19000
1433   17016
1434   16916
1435         1
Name: KM, Length: 1436, dtype: int64
```

In [34]:

```
df1.loc[:, "FuelType"]
```

Out[34]:

```
0      Diesel
1      Diesel
2      Diesel
3      Diesel
4      Diesel
...
1431   Petrol
1432   Petrol
1433   Petrol
1434   Petrol
1435   Petrol
Name: FuelType, Length: 1436, dtype: object
```

In [41]:

```
df1.isnull().sum()
```

Out[41]:

```
Price      0
Age        0
KM         0
FuelType   0
HP         0
MetColor   0
Automatic  0
CC         0
Doors      0
Weight     0
dtype: int64
```

In [38]:

```
df1
```

Out[38]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	2000	3	1165
1	13750	23	72937	Diesel	90	1	0	2000	3	1165
2	13950	24	41711	Diesel	90	1	0	2000	3	1165
3	14950	26	48000	Diesel	90	0	0	2000	3	1165
4	13750	30	38500	Diesel	90	0	0	2000	3	1170
...
1431	7500	69	20544	Petrol	86	1	0	1300	3	1025
1432	10845	72	19000	Petrol	86	0	0	1300	3	1015
1433	8500	71	17016	Petrol	86	0	0	1300	3	1015
1434	7250	70	16916	Petrol	86	1	0	1300	3	1015
1435	6950	76	1	Petrol	110	0	0	1600	5	1114

1436 rows × 10 columns

In [42]:

```
df1.drop('CC',axis=1,inplace=True)
```

In [43]:

```
df1
```

Out[43]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic	Doors	Weight
0	13500	23	46986	Diesel	90	1	0	3	1165
1	13750	23	72937	Diesel	90	1	0	3	1165
2	13950	24	41711	Diesel	90	1	0	3	1165
3	14950	26	48000	Diesel	90	0	0	3	1165
4	13750	30	38500	Diesel	90	0	0	3	1170
...
1431	7500	69	20544	Petrol	86	1	0	3	1025
1432	10845	72	19000	Petrol	86	0	0	3	1015
1433	8500	71	17016	Petrol	86	0	0	3	1015
1434	7250	70	16916	Petrol	86	1	0	3	1015
1435	6950	76	1	Petrol	110	0	0	5	1114

1436 rows × 9 columns

In [44]:

```
df1.drop('Doors',axis=1,inplace=True)
```

In [88]:

```
x1=df1.drop('Weight',axis=1)
x1
```

-
KeyError Traceback (most recent call last)

Input In [88], in <cell line: 1>()
----> 1 x1=df1.drop('Weight',axis=1)
2 x1

File ~\anaconda3\lib\site-packages\pandas\util_decorators.py:311, in deprecate_nonkeyword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
rgs)

```
305 if len(args) > num_allow_args:
306     warnings.warn(
307         msg.format(arguments=arguments),
308         FutureWarning,
309         stacklevel=stacklevel,
310     )
--> 311 return func(*args, **kwargs)
```

File ~\anaconda3\lib\site-packages\pandas\core\frame.py:4954, in DataFrame.drop(self, labels, axis, index, columns, level, inplace, errors)

```
4806 @deprecate_nonkeyword_arguments(version=None, allowed_args=["self", "labels"])
4807 def drop(
4808     self,
4809     (...)
4815     errors: str = "raise",
4816 ):
4817     """
4818     Drop specified labels from rows or columns.
4819
4820     (...)
4952         weight  1.0      0.8
4953     """
-> 4954     return super().drop(
4955         labels=labels,
4956         axis=axis,
4957         index=index,
4958         columns=columns,
4959         level=level,
4960         inplace=inplace,
4961         errors=errors,
4962     )
```

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:4267, in NDFrame.drop(self, labels, axis, index, columns, level, inplace, errors)

```
4265 for axis, labels in axes.items():
4266     if labels is not None:
-> 4267         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
4269 if inplace:
4270     self._update_inplace(obj)
```

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:4311, in NDFrame._drop_axis(self, labels, axis, level, errors, consolidate, only_slice)

```
4309     new_axis = axis.drop(labels, level=level, errors=errors)
4310     else:
-> 4311     new_axis = axis.drop(labels, errors=errors)
4312     indexer = axis.get_indexer(new_axis)
4314     # Case for non-unique axis
4315     else:
```

```

File ~\anaconda3\lib\site-packages\pandas\core\indexes\base.py:6644, in In
dex.drop(self, labels, errors)
    6642 if mask.any():
    6643     if errors != "ignore":
-> 6644         raise KeyError(f"{list(labels[mask])} not found in axis")
    6645     indexer = indexer[~mask]
    6646 return self.delete(indexer)

```

KeyError: "['Weight'] not found in axis"

In [46]:

```
df1
```

Out[46]:

	Price	Age	KM	FuelType	HP	MetColor	Automatic
0	13500	23	46986	Diesel	90	1	0
1	13750	23	72937	Diesel	90	1	0
2	13950	24	41711	Diesel	90	1	0
3	14950	26	48000	Diesel	90	0	0
4	13750	30	38500	Diesel	90	0	0
...
1431	7500	69	20544	Petrol	86	1	0
1432	10845	72	19000	Petrol	86	0	0
1433	8500	71	17016	Petrol	86	0	0
1434	7250	70	16916	Petrol	86	1	0
1435	6950	76	1	Petrol	110	0	0

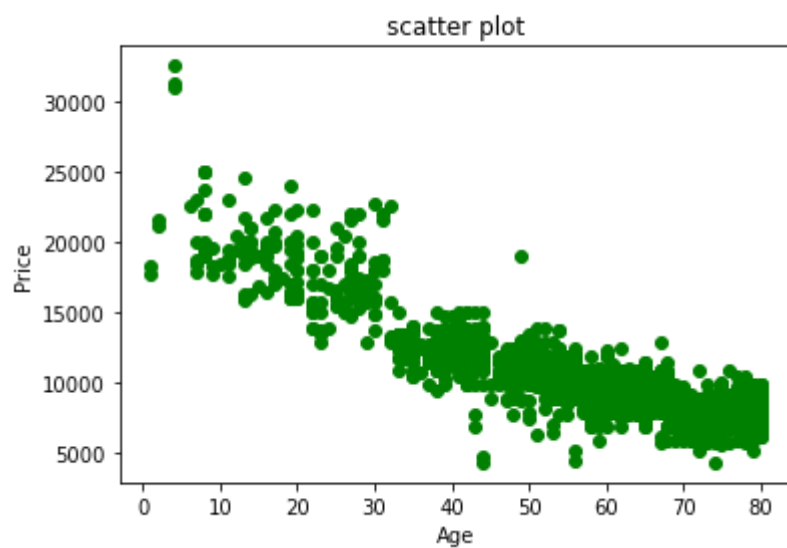
1436 rows × 7 columns

In [49]:

```
import matplotlib.pyplot as plt
```

In [72]:

```
plt.scatter(data=df1,x='Age',y='Price',color='green')
plt.title('scatter plot')
plt.xlabel('Age')
plt.ylabel('Price')
plt.show()
```

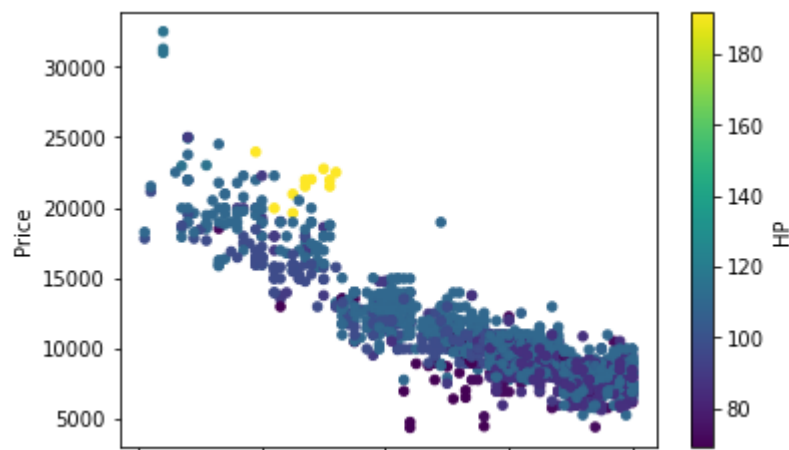


In [70]:

```
df1.plot.scatter('Age','Price',c='HP',colormap='viridis')
```

Out[70]:

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



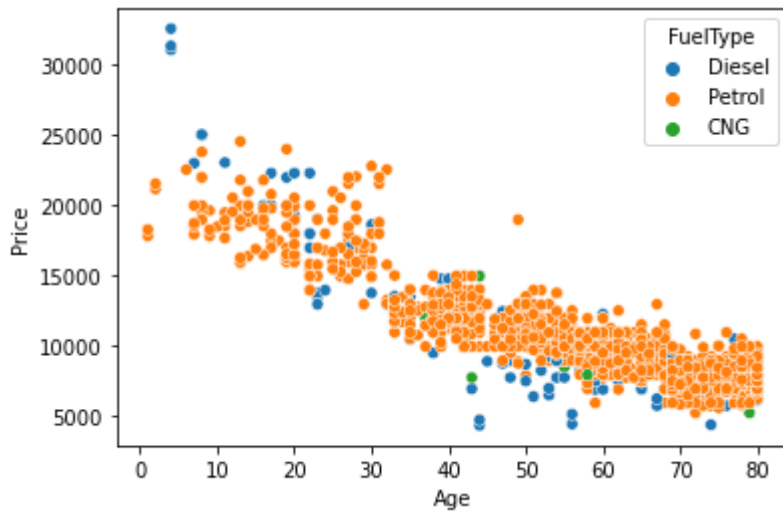
In [80]:

```
import seaborn as sns
sns.scatterplot(data=df1,x='Age',y='Price',hue='FuelType')

plt.plot()
```

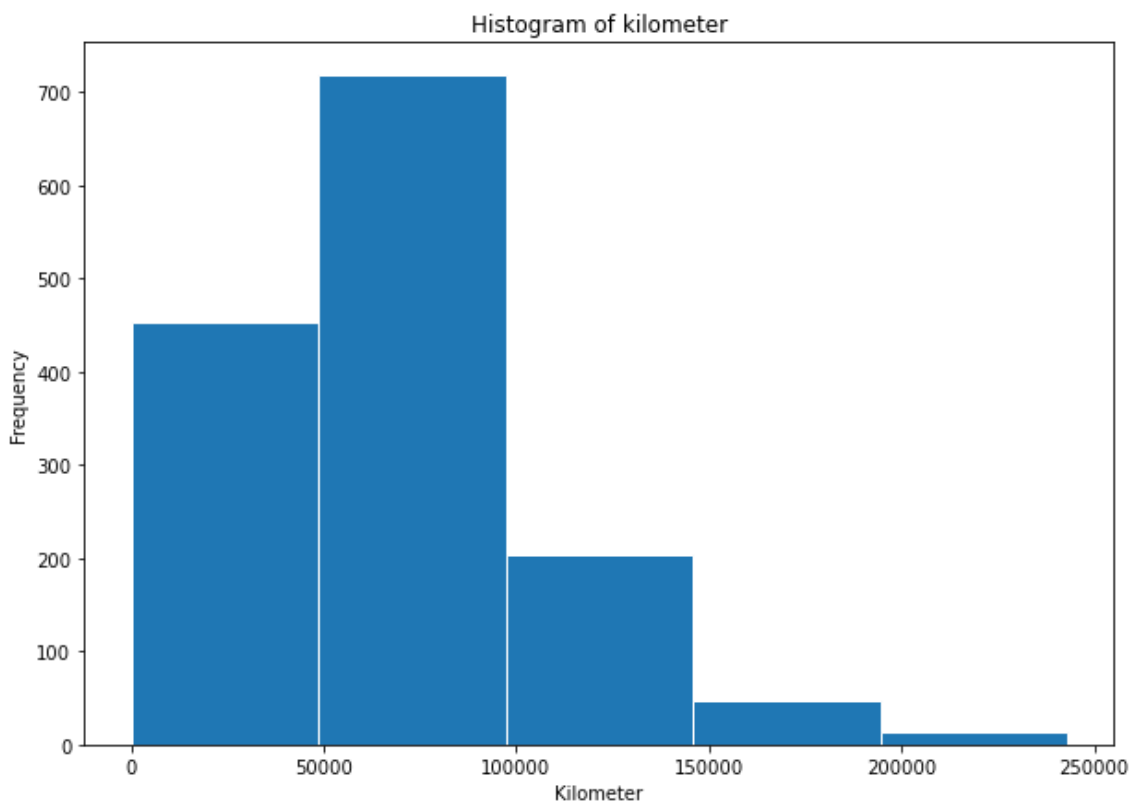
Out[80]:

[]



In [76]:

```
plt.figure(figsize=(10,7))
plt.hist(df1['KM'],bins=5,edgecolor='white')
plt.title('Histogram of kilometer')
plt.xlabel('Kilometer')
plt.ylabel('Frequency')
plt.show()
```

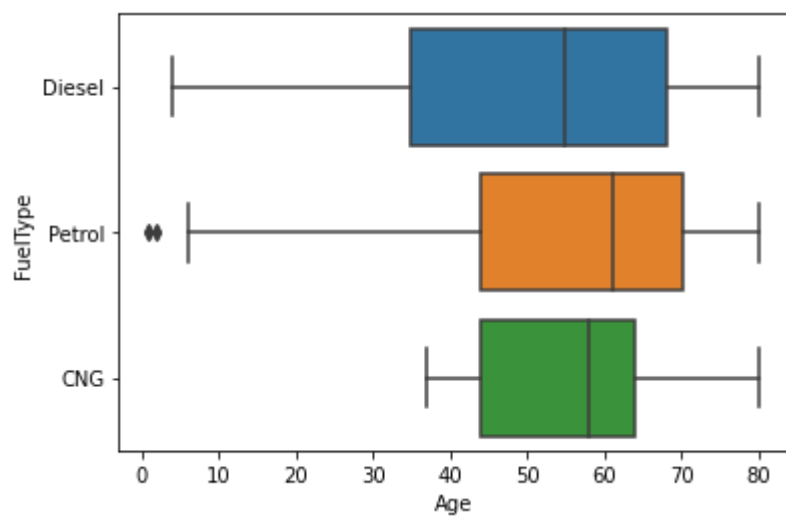


In [78]:

```
sns.boxplot(data=df1,x='Age',y='FuelType')
```

Out[78]:

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



In []: