In [1]:
```
import pandas as pd
```

In [2]:
```
#Task 1:
#Read the file as DataFrame and create a deep copy of it
df=pd.read_csv('toyota.csv')
df
```

Out[2]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Ra
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	1	...	1	1	1	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	1	...	1	0	1	
		? TOYOTA Corolla													

In [3]:
```
res=df.copy(deep=True)
print(res)
```

```
        Id                                    Model  Price  \
0        1    TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13500
1        2    TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13750
2        3   ?TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13950
3        4    TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  14950
4        5      TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors  13750
...    ...                                              ...    ...
1431  1438         TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors   7500
1432  1439  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...  10845
1433  1440  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   8500
1434  1441  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   7250
1435  1442      TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors   6950

      Age_08_04  Mfg_Month  Mfg_Year     KM Fuel_Type  HP  Met_Color  ...  \
0            23         10      2002  46986    Diesel  90          1  ...
1            23         10      2002  72937    Diesel  90          1  ...
2            24          9      2002  41711    Diesel  90          1  ...
3            26          7      2002  48000    Diesel  90          0  ...
4            30          3      2002  38500    Diesel  90          0  ...
```

In [4]:
```
#Find the basic information of the dataset.
df.head()
```

Out[4]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Radio	M
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	1	...	1	1	1	0	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	1	...	1	0	1	0	
		? TOYOTA Corolla														

In [5]:
```
df.tail()
```

Out[5]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Ra
1431	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	12	1998	20544	Petrol	86	1	...	1	1	1	
1432	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	9	1998	19000	Petrol	86	0	...	0	0	1	
1433	1440	TOYOTA Corolla 1.3 16V HATCHB	8500	71	10	1998	17016	Petrol	86	0	...	0	0	1	

In [6]:
```
df.shape
```

Out[6]:
```
(1436, 37)
```

In [7]:
```
df.size
```

Out[7]:
```
53132
```

In [8]:
```
#Find the dimensions of the data frame.
df.ndim
```

Out[8]:
```
2
```

In [9]:
```
#Determine the number of features available.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 37 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Id                1436 non-null   int64 
 1   Model             1436 non-null   object
 2   Price             1436 non-null   int64 
 3   Age_08_04         1436 non-null   int64 
 4   Mfg_Month         1436 non-null   int64 
 5   Mfg_Year          1436 non-null   int64 
 6   KM                1436 non-null   int64 
 7   Fuel_Type         1436 non-null   object
 8   HP                1436 non-null   int64 
 9   Met_Color         1436 non-null   int64 
 10  Automatic         1436 non-null   int64 
 11  cc                1436 non-null   int64 
 12  Doors             1436 non-null   int64 
 13  Cylinders         1436 non-null   int64 
 14  Gears             1436 non-null   int64 
```

In [10]:
```
#Perform 5 number summary (min, lower quartile, median, upper quartile, max.
df.min(0,skipna=True)
```

Out[10]:
```
Id                                                            1
Model               ?TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors
Price                                                      4350
Age_08_04                                                     1
Mfg_Month                                                     1
Mfg_Year                                                   1998
KM                                                            1
Fuel_Type                                                   CNG
HP                                                           69
Met_Color                                                     0
Automatic                                                     0
cc                                                         1300
Doors                                                         2
Cylinders                                                     4
Gears                                                         3
Quarterly_Tax                                                19
Weight                                                     1000
Mfr_Guarantee                                                 0
BOVAG_Guarantee                                               0
Guarantee_Period                                              3
```

In [11]:

```
df.quantile([.2,.1],interpolation='lower')
```

Out[11]:

	Id	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	HP	Met_Color	Automatic	cc	...	Central_Lock	Powered_Windows	Power_Steering	Radio	Mistla
0.2	289	7950	40	2	1998	37320	86	0	0	1400	...	0	0	1	0	
0.1	145	7450	27	1	1998	26221	86	0	0	1300	...	0	0	1	0	

2 rows × 35 columns

In [12]:

```
df.quantile([.6,.8],interpolation='higher')
```

Out[12]:

	Id	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	HP	Met_Color	Automatic	cc	...	Central_Lock	Powered_Windows	Power_Steering	Radio	Mis
0.6	865	10500	65	6	2000	72090	110	1	0	1600	...	1	1	1	0	
0.8	1154	12500	73	9	2001	94606	110	1	0	1600	...	1	1	1	0	

2 rows × 35 columns

In [13]:

```
df['Price'].median()
```

Out[13]:

```
9900.0
```

In [14]:

```
df.median(0)
```

```
C:\Users\admin\AppData\Local\Temp\ipykernel_12020\475342755.py:1: FutureWarning: Dropping of nuisance columns in DataFrame re
ductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  df.median(0)
```

Out[14]:

```
Id                   721.5
Price               9900.0
Age_08_04             61.0
Mfg_Month              5.0
Mfg_Year            1999.0
KM                 63389.5
HP                   110.0
Met_Color              1.0
Automatic              0.0
cc                  1600.0
Doors                  4.0
Cylinders              4.0
Gears                  5.0
Quarterly_Tax         85.0
Weight              1070.0
```

In [15]:

```
df.max()
```

Out[15]:

```
Id                                                1442
Model               TOYOTA Corolla VERSO 2.0 D4D SOL (7) MPV
Price                                            32500
Age_08_04                                           80
Mfg_Month                                           12
Mfg_Year                                          2004
KM                                              243000
Fuel_Type                                       Petrol
HP                                                 192
Met_Color                                            1
Automatic                                            1
cc                                               16000
Doors                                                5
Cylinders                                            4
Gears                                                6
Quarterly_Tax                                      283
Weight                                            1615
Mfr_Guarantee                                        1
BOVAG_Guarantee                                      1
Guarantee_Period                                    36
```

In [16]:
```
#Access the top 10 rows from the dataset.
df.head(10)
```

Out[16]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Radio	I
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	1	...	1	1	1	0	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	1	...	1	0	1	0	
		? TOYOTA Corolla 2.0 D4D														

In [17]:
```
#Access last 2 rows from the dataset
df.tail(2)
```

Out[17]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Radio
1434	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	11	1998	16916	Petrol	86	1	...	0	0	0	0
1435	1442	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	5	1998	1	Petrol	110	0	...	0	0	1	0

2 rows × 37 columns

In [18]:
```
#Task 2:
#Access a group of rows and columns by label(s).
#['Price, Age, KM, FuelType]
data=df[['Price','Age_08_04','KM','Fuel_Type']]
print(data)
```

```
      Price  Age_08_04     KM Fuel_Type
0     13500         23  46986    Diesel
1     13750         23  72937    Diesel
2     13950         24  41711    Diesel
3     14950         26  48000    Diesel
4     13750         30  38500    Diesel
...     ...        ...    ...       ...
1431   7500         69  20544    Petrol
1432  10845         72  19000    Petrol
1433   8500         71  17016    Petrol
1434   7250         70  16916    Petrol
1435   6950         76      1    Petrol

[1436 rows x 4 columns]
```

In [19]:
```
#Find the missing or null values for each column.
df.isnull()
```

Out[19]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Radio
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1431	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1432	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1433	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1434	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False
1435	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False

1436 rows × 37 columns

In [20]:
```
#Display total number of missing values for each column.
df.isnull().sum()
```

Out[20]:
```
Id                  0
Model               0
Price               0
Age_08_04           0
Mfg_Month           0
Mfg_Year            0
KM                  0
Fuel_Type           0
HP                  0
Met_Color           0
Automatic           0
cc                  0
Doors               0
Cylinders           0
Gears               0
Quarterly_Tax       0
Weight              0
Mfr_Guarantee       0
BOVAG_Guarantee     0
```

In [21]:
```
#Replace missing values with mean for continuous variable and mod for categorical variables.
#Also display the result for total missing values after replacing the missing values.
df.fillna(df.mean(),inplace=True)
print(df)
```

```
        Id                                          Model  Price  \
0        1      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13500
1        2      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13750
2        3     ?TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13950
3        4      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  14950
4        5        TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors  13750
...    ...                                            ...    ...
1431  1438         TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors   7500
1432  1439  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...  10845
1433  1440  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   8500
1434  1441  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   7250
1435  1442        TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors   6950

      Age_08_04  Mfg_Month  Mfg_Year     KM Fuel_Type   HP  Met_Color  ...  \
0            23         10      2002  46986    Diesel   90          1  ...
1            23         10      2002  72937    Diesel   90          1  ...
2            24          9      2002  41711    Diesel   90          1  ...
3            26          7      2002  48000    Diesel   90          0  ...
4            30          3      2002  38500    Diesel   90          0  ...
```

In [22]:
```
df.fillna(df.mode(),inplace=True)
print(df)
```

```
        Id                                          Model  Price  \
0        1      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13500
1        2      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13750
2        3     ?TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  13950
3        4      TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors  14950
4        5        TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors  13750
...    ...                                            ...    ...
1431  1438         TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors   7500
1432  1439  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...  10845
1433  1440  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   8500
1434  1441  TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...   7250
1435  1442        TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors   6950

      Age_08_04  Mfg_Month  Mfg_Year     KM Fuel_Type   HP  Met_Color  ...  \
0            23         10      2002  46986    Diesel   90          1  ...
1            23         10      2002  72937    Diesel   90          1  ...
2            24          9      2002  41711    Diesel   90          1  ...
3            26          7      2002  48000    Diesel   90          0  ...
4            30          3      2002  38500    Diesel   90          0  ...
```

In [23]:
```
#Remove the following features from the dataset
#[CC, Doors, Weight]
remove=df.drop(['cc','Doors','Weight'],axis=1)
remove
```

Out[23]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...	Central_Lock	Powered_Windows	Power_Steering	Ra
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	1	...	1	1	1	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	1	...	1	0	1	
		? TOYOTA Corolla 2.0 D4D													

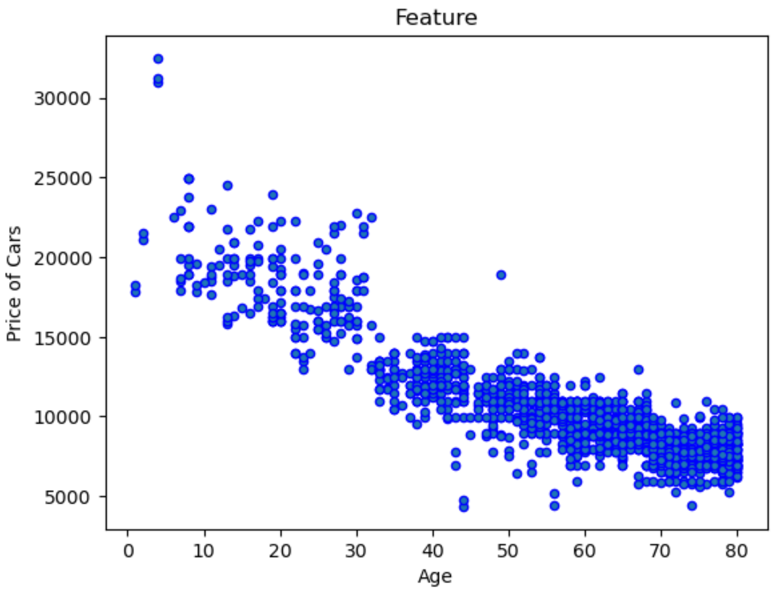
In [24]:
```
#Task 3:
```

In [25]:
```
from matplotlib import pyplot as plt
```

In [26]:
```
import seaborn as sns
```

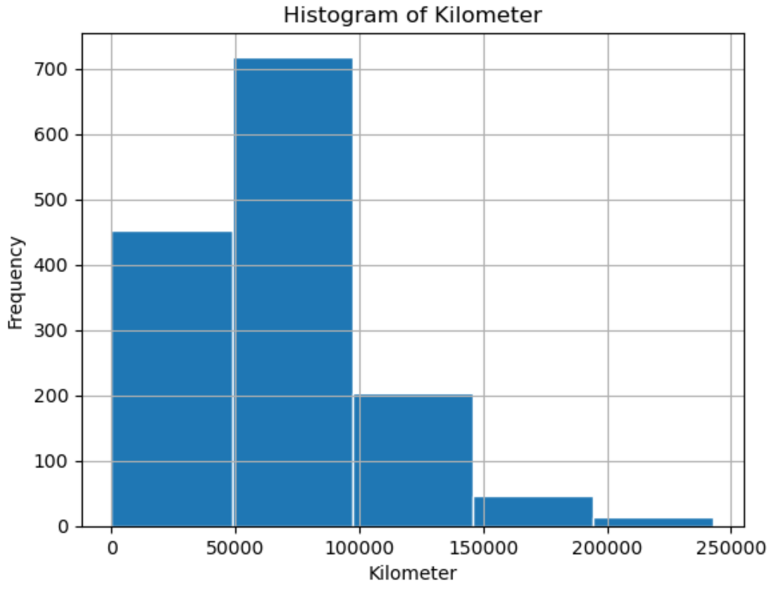
In [27]:

```
#Visualize the data using scatter plot for two features (x=Age, y=Price).Also interpret the result.
#Provide title, and labels for both axis.
#Apply some marker and set different colors for bar and marker.

df=pd.DataFrame(df)
df.plot.scatter(x='Age_08_04',y='Price',title='Feature',marker='o',edgecolor='blue')
plt.xlabel('Age')
plt.ylabel('Price of Cars')
plt.show()
```



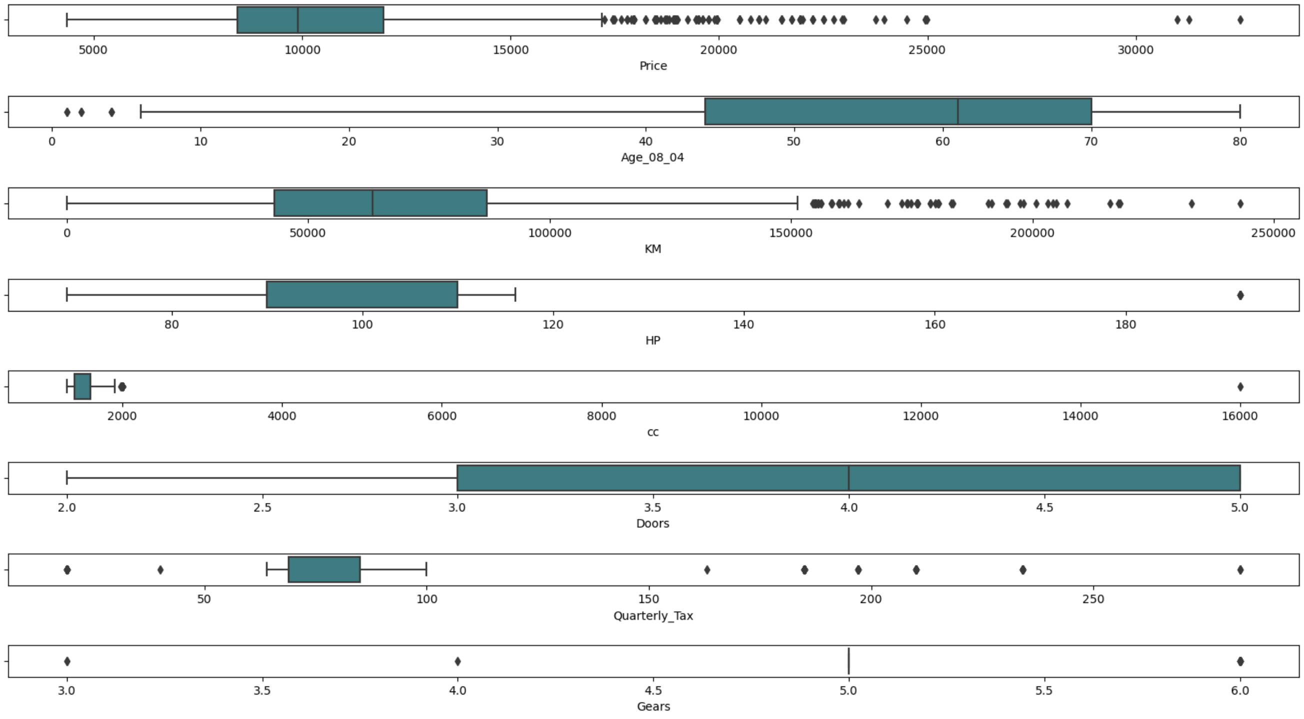
In [28]:

```
'''Create a histogram for the feature KM.
Also set the following properties:
o   No of bins=5
o   Edge color=White
o   X label= Kilometer
o   Y label= Frequency
o   Title= Histogram of Kilometer'''


df.hist('KM',bins=5,edgecolor='white')
plt.xlabel('Kilometer')
plt.ylabel('Frequency')
plt.title('Histogram of Kilometer')
plt.show()
```



In [30]:

```
'''Detect Outliers
Apply box and whisker plot to find the outliers in the dataset.
Also interpret the result.'''

fig, axes=plt.subplots(8,1,figsize=(16,9),sharex=False,sharey=False)
sns.boxplot(x='Price',data=df,palette='crest',ax=axes[0])
sns.boxplot(x='Age_08_04',data=df,palette='crest',ax=axes[1])
sns.boxplot(x='KM',data=df,palette='crest',ax=axes[2])
sns.boxplot(x='HP',data=df,palette='crest',ax=axes[3])
sns.boxplot(x='cc',data=df,palette='crest',ax=axes[4])
sns.boxplot(x='Doors',data=df,palette='crest',ax=axes[5])
sns.boxplot(x='Quarterly_Tax',data=df,palette='crest',ax=axes[6])
sns.boxplot(x='Gears',data=df,palette='crest',ax=axes[7])
plt.tight_layout(pad=2.0)
```



In [ ]: