Task 1:

Ø Read the file as DataFrame and create a deep copy of it

In [2]:
```
# import all the necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
# read file

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

Ø Find the basic information of the dataset.

In [4]:
```
toyota.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
 15  Quarterly_Tax     1436 non-null   int64
 16  Weight            1436 non-null   int64
 17  Mfr_Guarantee     1436 non-null   int64
 18  BOVAG_Guarantee   1436 non-null   int64
 19  Guarantee_Period  1436 non-null   int64
 20  ABS               1436 non-null   int64
 21  Airbag_1          1436 non-null   int64
 22  Airbag_2          1436 non-null   int64
 23  Airco             1436 non-null   int64
 24  Automatic_airco   1436 non-null   int64
 25  Boardcomputer     1436 non-null   int64
 26  CD_Player         1436 non-null   int64
 27  Central_Lock      1436 non-null   int64
 28  Powered_Windows   1436 non-null   int64
 29  Power_Steering    1436 non-null   int64
 30  Radio             1436 non-null   int64
 31  Mistlamps         1436 non-null   int64
 32  Sport_Model       1436 non-null   int64
 33  Backseat_Divider  1436 non-null   int64
 34  Metallic_Rim      1436 non-null   int64
 35  Radio_cassette    1436 non-null   int64
 36  Tow_Bar           1436 non-null   int64
dtypes: int64(35), object(2)
memory usage: 415.2+ KB
```

Ø Find the dimensions of the data frame.

In [5]:
```
toyota.ndim
```

Out[5]: 2

Ø Determine the number of features available.

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

Out[6]: (1436, 37)

Total number of features = 37

Ø Perform 5 number summary (min, lower quartile, median, upper quartile, max.

In [7]:
```
toyota.describe()
```

Out[7]:

	Id	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	
count	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000
mean	721.555014	10730.824513	55.947075	5.548747	1999.625348	68533.259749	101.502
std	416.476890	3626.964585	18.599988	3.354085	1.540722	37506.448872	14.981
min	1.000000	4350.000000	1.000000	1.000000	1998.000000	1.000000	69.000
25%	361.750000	8450.000000	44.000000	3.000000	1998.000000	43000.000000	90.000
50%	721.500000	9900.000000	61.000000	5.000000	1999.000000	63389.500000	110.000
75%	1081.250000	11950.000000	70.000000	8.000000	2001.000000	87020.750000	110.000
max	1442.000000	32500.000000	80.000000	12.000000	2004.000000	243000.000000	192.000

8 rows × 35 columns

Ø Access the top 10 rows from the dataset.

In [8]:
```
toyota.head(10)
```

Out[8]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	1	...
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	1	...
2	3	? TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	9	2002	41711	Diesel	90	1	...
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	7	2002	48000	Diesel	90	0	...
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	3	2002	38500	Diesel	90	0	...
5	6	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	12950	32	1	2002	61000	Diesel	90	0	...
6	7	? TOYOTA Corolla 2.0 D4D 90 3DR TERRA 2/3-Doors	16900	27	6	2002	94612	Diesel	90	1	...
7	8	TOYOTA Corolla 2.0 D4D 90 3DR TERRA 2/3-Doors	18600	30	3	2002	75889	Diesel	90	1	...

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Color	...
8	9	? TOYOTA Corolla 1800 T SPORT VVT I 2/3-Doors	21500	27	6	2002	19700	Petrol	192	0	...
9	10	? TOYOTA Corolla 1.9 D HATCHB TERRA 2/3-Doors	12950	23	10	2002	71138	Diesel	69	0	...

10 rows × 37 columns

Ø Access last 2 rows from the dataset

In [9]:
```
toyota.tail(2)
```

Out[9]:

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

2 rows × 37 columns

Task 2:

Ø Access a group of rows and columns by label(s).

['Price', 'Age', 'KM', 'FuelType']

In [10]:
```
toyota.loc[:,['Price', 'Age_08_04', 'KM', 'Fuel_Type']]
```

Out[10]:

	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 × 4 columns

Ø Find the missing or null values for each column.

In [11]:
```
toyota.isnull()
```

Out[11]:

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

1436 rows × 37 columns

Ø Display total number of missing values for each column.

In [12]:
```
toyota.isnull().sum()
```

Out[12]:
```
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
Guarantee_Period    0
ABS                 0
Airbag_1            0
Airbag_2            0
Airco               0
Automatic_airco     0
Boardcomputer       0
CD_Player           0
Central_Lock        0
Powered_Windows     0
Power_Steering      0
Radio               0
Mistlamps           0
Sport_Model         0
Backseat_Divider    0
Metallic_Rim        0
Radio_cassette      0
Tow_Bar             0
dtype: int64
```

Ø Replace missing values with mean for continuous variable and mode for categorical variables.

Also display the result for total missing values after replacing the missing values.

-->toyota dataframe has 0 missing/NaN values so theres no need to fill missing values

Ø Remove the following features from the dataset [‘CC’, ’Doors’, ’Weight’]

In [13]:
```
toyota.drop(['cc','Doors','Weight'],axis=1)
# toyota.drop(['cc',Doors','Weight'],axis=1,inplace=True)
```

Out[13]:

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Col
0	1	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13500	23	10	2002	46986	Diesel	90	
1	2	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13750	23	10	2002	72937	Diesel	90	
2	3	? TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	13950	24	9	2002	41711	Diesel	90	
3	4	TOYOTA Corolla 2.0 D4D HATCHB TERRA 2/3-Doors	14950	26	7	2002	48000	Diesel	90	
4	5	TOYOTA Corolla 2.0 D4D HATCHB SOL 2/3-Doors	13750	30	3	2002	38500	Diesel	90	
...	...	...	...	...	...	...	...	...	...	...
1431	1438	TOYOTA Corolla 1.3 16V HATCHB G6 2/3-Doors	7500	69	12	1998	20544	Petrol	86	
1432	1439	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	10845	72	9	1998	19000	Petrol	86	

	Id	Model	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	Fuel_Type	HP	Met_Col
1433	1440	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	8500	71	10	1998	17016	Petrol	86	
1434	1441	TOYOTA Corolla 1.3 16V HATCHB LINEA TERRA 2/3-...	7250	70	11	1998	16916	Petrol	86	
1435	1442	TOYOTA Corolla 1.6 LB LINEA TERRA 4/5-Doors	6950	76	5	1998	1	Petrol	110	

1436 rows × 34 columns

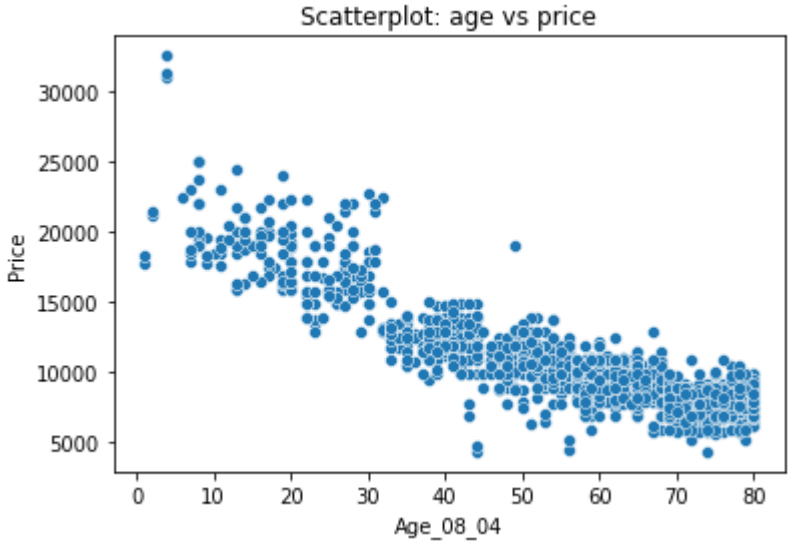
Task 3:

Ø Visualize the data using scatter plot for two features (x='Age', y='Price').Also interpret the result.

· Provide title, and labels for both axis.

In [14]:
```
sns.scatterplot(x = toyota['Age_08_04'], y = toyota['Price'])
plt.title('Scatterplot: age vs price')
```

Out[14]:
```
Text(0.5, 1.0, 'Scatterplot: age vs price')
```



As the age of car increases the price decreases

also whose age is greater than 40 are more and their price is less than 15000

· Apply some marker and set different colors for bar and marker.

In [15]:
```
sns.scatterplot(x = toyota['Age_08_04'], y = toyota['Price'], marker = '*', color
plt.title('Scatterplot: age vs price')
```

Out[15]:
```
Text(0.5, 1.0, 'Scatterplot: age vs price')
```

Ø Create a histogram for the feature ‘KM’.

· Also set the following properties:

o No of bins=5

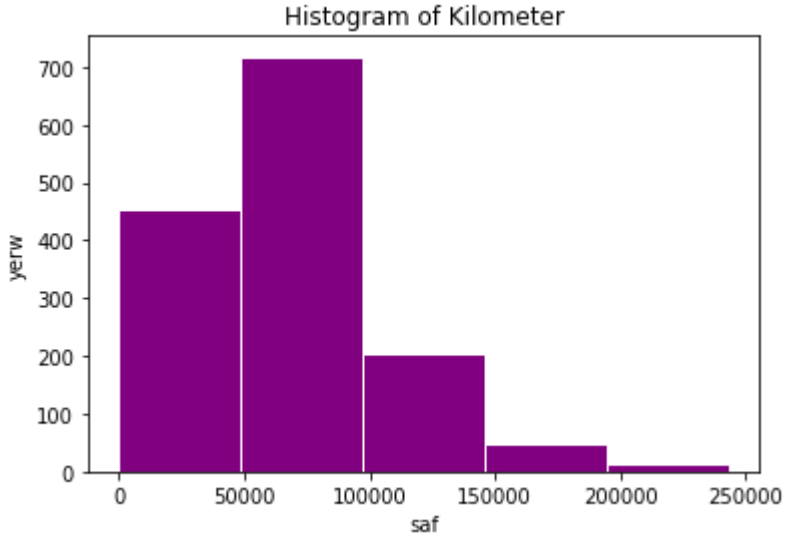
o Edge color=White

o X label= Kilometer

o Y label= Frequency

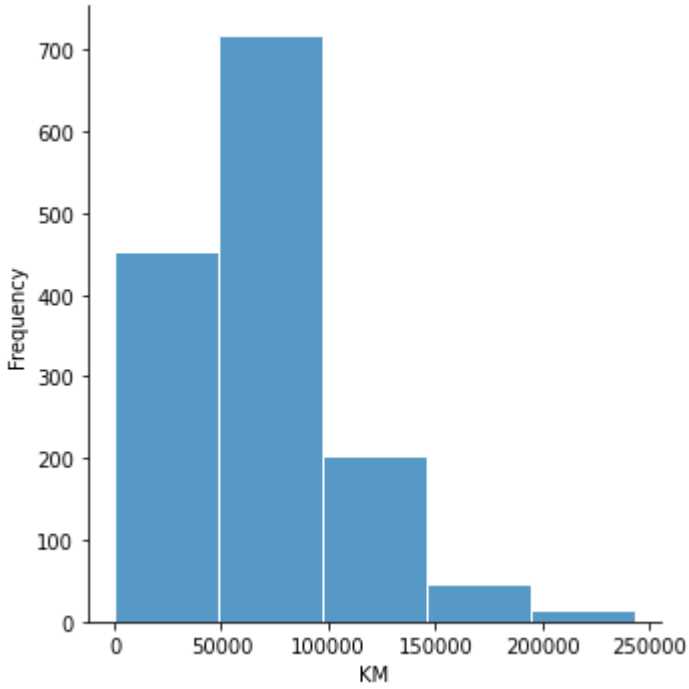
o Title= Histogram of Kilometer

In [16]:
```
plt.hist(x=toyota['KM'],color='purple',bins=5,edgecolor='white')
plt.title("Histogram of Kilometer")
plt.xlabel("saf")
plt.ylabel("yerw")
plt.show()
```



In [17]:
```
histogram = sns.displot(toyota['KM'],bins=5,edgecolor='white')
histogram.set_axis_labels('KM', 'Frequency')
```

Out[17]: <seaborn.axisgrid.FacetGrid at 0x24ddc474d00>



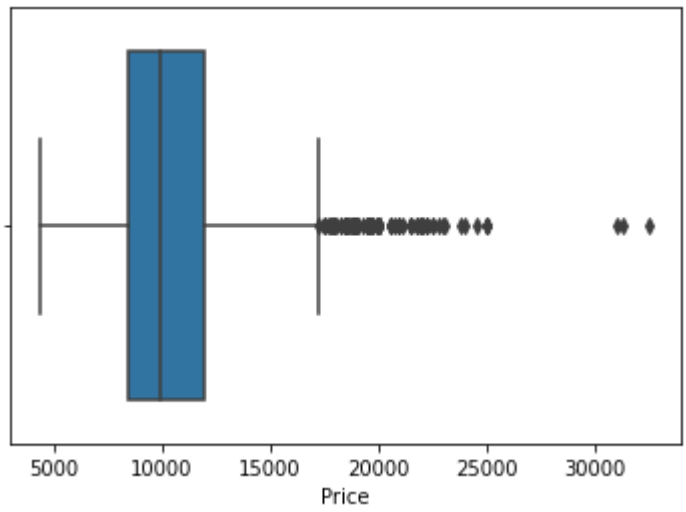
Ø Detect Outliers

· Apply box and whisker plot to find the outliers in the dataset.

· Also interpret the result.

In [18]:
```
sns.boxplot(toyota['Price'])
```

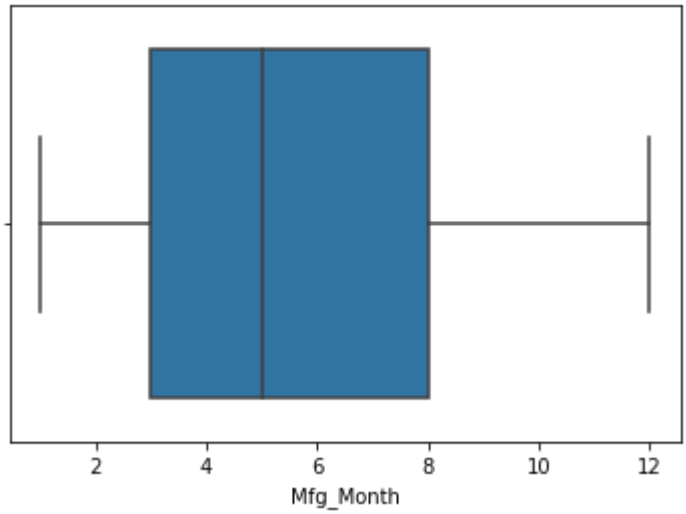
Out[18]: <AxesSubplot:xlabel='Price'>



In price column there are outlier presents

In [20]:
```
sns.boxplot(toyota['Mfg_Month'])
```

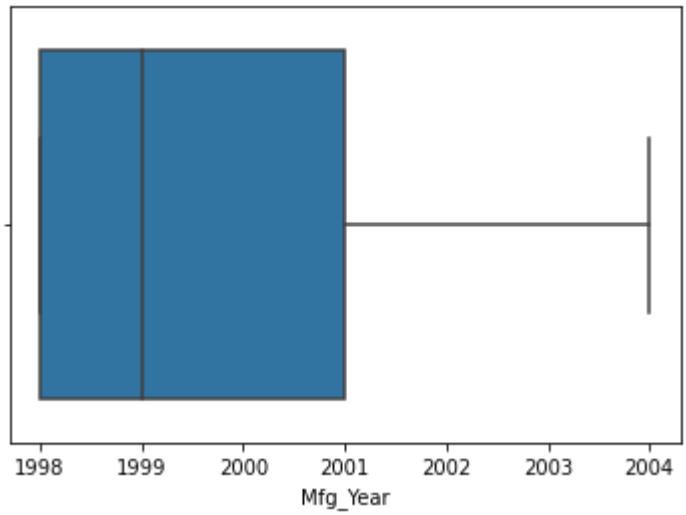
Out[20]: <AxesSubplot:xlabel='Mfg_Month'>



No outlier present in mfg_month

In [21]:
```
sns.boxplot(toyota['Mfg_Year'])
```

Out[21]: <AxesSubplot:xlabel='Mfg_Year'>



In [ ]:
```
No outlier present
```