a)Look for the missing values in all the columns and either impute them (replace with mean, median, or mode) or drop them. Justify your action for this task.

```
[5]  # Checking for missing values in the dataset
     missing_values = data.isnull().sum()
     print("Missing Values:")
     print(missing_values)
```

```
Missing Values:
Unnamed: 0             0
Name                   0
Location               0
Year                   0
Kilometers_Driven      0
Fuel_Type              0
Transmission           0
Owner_Type             0
Mileage                2
Engine                36
Power                 36
Seats                 38
New_Price           5032
Price                  0
dtype: int64
```

b) Remove the units from some of the attributes and only keep the numerical values (for example remove kmpl from “Mileage”, CC from “Engine”, bhp from “Power”, and lakh from “New_price”)

```
data['Mileage'] = data['Mileage'].apply(lambda x: re.findall(r'\d+\.?\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Mileage'].fillna(data['Mileage'].mean(), inplace=True)

data['Engine'] = data['Engine'].apply(lambda x: re.findall(r'\d+', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Engine'].fillna(data['Engine'].median(), inplace=True)

data['Power'] = data['Power'].apply(lambda x: re.findall(r'\d+\.?\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Power'].fillna(data['Power'].median(), inplace=True)

data['Seats'] = data['Seats'].apply(lambda x: re.findall(r'\d+\.?\d*', str(x))[0] if pd.notnull(x) else x).astype(float)
data['Seats'].fillna(data['Seats'].mean(), inplace=True)

#dropping new_price column because it contains high number of missing values
data.drop(['New_Price', 'Unnamed: 0'], axis=1, inplace=True)
```

```
7]  data
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50
1	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13.00	1199.0	88.70	5.0	4.50
2	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74
4	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08	1461.0	63.10	5.0	3.50
...	...	...	...	...	...	...	...	...	...	...	...	...
5842	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.40	1248.0	74.00	5.0	4.75
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.40	1120.0	71.00	5.0	4.00
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.00	2498.0	112.00	8.0	2.90

C)Change the categorical variables (“Fuel_Type” and “Transmission”) into numerical one hot encoded value

```
data = pd.get_dummies(data, columns=['Fuel_Type', 'Transmission'], prefix=['Fuel','Transmission'])
```

```
[10] data
```

	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automatic
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	1	0	0	0
1	Honda Jazz V	Chennai	2011	46000	First	13.00	1199.0	88.70	5.0	4.50	0	0	1	0
2	Maruti Ertiga VDI	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	1	0	0	0
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	1	0	0	1
4	Nissan Micra Diesel XV	Jaipur	2013	86999	First	23.08	1461.0	63.10	5.0	3.50	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5842	Maruti Swift VDI	Delhi	2014	27365	First	28.40	1248.0	74.00	5.0	4.75	1	0	0	0
5843	Hyundai Xcent 1.1	Jaipur	2015	100000	First	24.40	1120.0	71.00	5.0	4.00	1	0	0	0

d) Create one more feature and add this column to the dataset (you can use mutate function in
R for this). For example, you can calculate the current age of the car by subtracting “Year” value
from the current year

Task d) Create one more feature and add this column to the dataset (you can use mutate function in R for this). For example, you can calculate the current age of the car by subtracting “Year” value from the current year.

```
[12] # Import the datetime module
     from datetime import datetime

     # Calculating the current year
     current_year = datetime.now().year

     # Creating a new column 'Car_Age' by subtracting 'Year' from the current year
     data['Car_Age'] = current_year - data['Year']
```

```
[ ] data
```

	Name	Location	Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automatic
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	First	19.67	1582.0	126.20	5.0	12.50	1	0	0	0
1	Honda Jazz V	Chennai	2011	46000	First	13.00	1199.0	88.70	5.0	4.50	0	0	1	0
2	Maruti Ertiga VDI	Chennai	2012	87000	First	20.77	1248.0	88.76	7.0	6.00	1	0	0	0
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	1	0	0	1

✓ 0s completed at 6:08 PM

Year	Kilometers_Driven	Owner_Type	Mileage	Engine	Power	Seats	Price	Fuel_Diesel	Fuel_Electric	Fuel_Petrol	Transmission_Automatic	Transmission_Manual	Car_Age
2015	41000	First	19.67	1582.0	126.20	5.0	12.50	1	0	0	0	1	8
2011	46000	First	13.00	1199.0	88.70	5.0	4.50	0	0	1	0	1	12
2012	87000	First	20.77	1248.0	88.76	7.0	6.00	1	0	0	0	1	11
2013	40670	Second	15.20	1968.0	140.80	5.0	17.74	1	0	0	1	0	10
2013	86999	First	23.08	1461.0	63.10	5.0	3.50	1	0	0	0	1	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2014	27365	First	28.40	1248.0	74.00	5.0	4.75	1	0	0	0	1	9
2015	100000	First	24.40	1120.0	71.00	5.0	4.00	1	0	0	0	1	8
2012	55000	Second	14.00	2498.0	112.00	8.0	2.90	1	0	0	0	1	11

e) Perform select, filter, rename, mutate, arrange and summarize with group by operations (or their equivalent operations in python) on this dataset.

e) Perform select, filter, rename, mutate, arrange and summarize with group by operations (or their equivalent operations in python) on this dataset.

```
sorted_data = data.sort_values(by='Price', ascending=False)  # Sort the data by price in descending order


grouped_summary = data.groupby('Location').agg({'Price': 'mean', 'Kilometers_Driven': 'sum'}).reset_index()


# Printing the results
print("\Sorted Data:")
print(sorted_data)

print("\n")
print(grouped_summary)
```

```
\Sorted Data:
                                          Name    Location  Year  \
3952  Land Rover Range Rover 3.0 Diesel LWB Vogue   Hyderabad  2017
5620                  Lamborghini Gallardo Coupe       Delhi  2011
5752                      Jaguar F Type 5.0 V8 S   Hyderabad  2015
1457              Land Rover Range Rover Sport SE       Kochi  2019
1917                          BMW 7 Series 740Li  Coimbatore  2018
...                                          ...         ...   ...
3127                              Maruti 800 Std        Pune  2003
2758                           Hyundai Getz GLS        Pune  2005
3039                             Maruti Zen LXI      Jaipur  1998
1577                          Maruti 800 Std BSIII      Jaipur  2004
1660                                 Tata Nano Lx        Pune  2011

      Kilometers_Driven Owner_Type  Mileage  Engine  Power  Seats   Price  \
3952              25000      First    13.33  2993.0  255.0    5.0  160.00
```

```
3039                          Maruti Zen LXI      Jaipur  1998
1577                   Maruti 800 Std BSIII      Jaipur  2004
1660                           Tata Nano Lx        Pune  2011

      Kilometers_Driven Owner_Type  Mileage  Engine  Power  Seats   Price  \
3952              25000      First    13.33  2993.0  255.0    5.0  160.00
5620               6500      Third     6.40  5204.0  560.0    2.0  120.00
5752               8000      First    12.50  5000.0  488.1    2.0  100.00
1457              26013      First    12.65  2993.0  255.0    5.0   97.07
1917              28060      First    12.05  2979.0  320.0    5.0   93.67
...                 ...        ...      ...     ...    ...    ...     ...
3127              52000      First    16.10   796.0   37.0    4.0    0.50
2758              86000     Second    15.30  1341.0   83.0    5.0    0.45
3039              95150      Third    17.30   993.0   60.0    5.0    0.45
1577              12000     Second    16.10   796.0   37.0    4.0    0.45
1660              65000     Second    26.00   624.0   35.0    4.0    0.44

      Fuel_Diesel  Fuel_Electric  Fuel_Petrol  Transmission_Automatic  \
3952            1              0            0                       1
5620            0              0            1                       1
5752            0              0            1                       1
1457            1              0            0                       1
1917            0              0            1                       1
...           ...            ...          ...                     ...
3127            0              0            1                       0
2758            0              0            1                       0
3039            0              0            1                       0
1577            0              0            1                       0
1660            0              0            1                       0

      Transmission_Manual  Car_Age
3952                    0        7
5620                    0       13
5752                    0        9
1457                    0        5
1917                    0        6
```