PDS ASSIGNMENT – 2

Name: Vinisha Vimal Kumar Shukla **ID:** 16344280

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.

```
import pandas as pd

# Read the dataset
df_automobiles = pd.read_csv('/content/train.csv')
df_automobiles
```

	Unnamed: 0	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
1	2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	13 km/kg	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
2	3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
3	4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74
4	6	Nissan Micra Diesel XV	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmpl	1461 CC	63.1 bhp	5.0	NaN	3.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5842	6014	Maruti Swift VDI	Delhi	2014	27365	Diesel	Manual	First	28.4 kmpl	1248 CC	74 bhp	5.0	7.88 Lakh	4.75
5843	6015	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	Diesel	Manual	First	24.4 kmpl	1120 CC	71 bhp	5.0	NaN	4.00
5844	6016	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	Diesel	Manual	Second	14.0 kmpl	2498 CC	112 bhp	8.0	NaN	2.90
5845	6017	Maruti Wagon R VXI	Kolkata	2013	46000	Petrol	Manual	First	18.9 kmpl	998 CC	67.1 bhp	5.0	NaN	2.65
5846	6018	Chevrolet Beat Diesel	Hyderabad	2011	47000	Diesel	Manual	First	25.44 kmpl	936 CC	57.6 bhp	5.0	NaN	2.50

```
# Checking for missing values in the entire dataset
missing_vals = df_automobiles.isnull().sum()

# Display the number of missing values for columns with numeric datatype
print(missing_vals)
```

```
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
```

```
# Identify and count missing values in columns that have character datatype
missing_vals_char = df_automobiles.select_dtypes(include=['object']).isnull().sum()

# Display the number of missing values in columns that have character datatype
print(missing_vals_char)
```

```
Name               0
Location           0
Fuel_Type          0
Transmission       0
Owner_Type         0
Mileage            2
Engine            36
Power             36
New_Price       5032
dtype: int64
```

```
Name               0
Location           0
Year               0
Kilometers_Driven  0
Fuel_Type          0
Transmission       0
Owner_Type         0
Mileage            0
Engine             0
Power              0
Seats              0
Price              0
dtype: int64
<ipython-input-16-0ce23b57a77a>:8: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will def
  df_automobiles.fillna(df_automobiles.mean(), inplace=True)
```

	Name	Location	Year	Kilometers_Driven	\
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	
1	Honda Jazz V	Chennai	2011	46000	
2	Maruti Ertiga VDI	Chennai	2012	87000	
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	
4	Nissan Micra Diesel XV	Jaipur	2013	86999	

	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price
0	0	1	First	19.67	1582.0	126.20	5.0	12.50
1	1	1	First	13.00	1199.0	88.70	5.0	4.50
2	0	1	First	20.77	1248.0	88.76	7.0	6.00
3	0	0	Second	15.20	1968.0	140.80	5.0	17.74
4	0	1	First	23.08	1461.0	63.10	5.0	3.50

+ Code + Text

Justification:

- It ensures data integrity and completeness by retaining all available information in the dataset.
- By keeping observations with missing values, it maintains the sample size, preventing data loss and preserving statistical power.
- Imputation methods like mean and mode utilize existing data distribution, providing representative values for missing entries.
- This approach mitigates the risk of introducing bias into the analysis, as imputed values are derived from the observed data.
- Utilizing available information maximizes the dataset's potential for accurate analysis and modeling.
- It enhances the robustness of the analysis by leveraging the entire dataset, even in the presence of missing values.
- Imputing missing values facilitates downstream tasks like modeling and visualization, as the dataset remains intact and ready for analysis without additional preprocessing steps.

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”).

```
# Remove the units from some of the attributes and only keep the numerical values
df_automobiles['Mileage'] = df_automobiles['Mileage'].str.replace('[^0-9.]', '', regex=True).astype(float)
df_automobiles['Engine'] = df_automobiles['Engine'].str.replace('[^0-9.]', '', regex=True).astype(float)
df_automobiles['Power'] = df_automobiles['Power'].str.replace('[^0-9.]', '', regex=True).astype(float)
df_automobiles['New_Price'] = df_automobiles['New_Price'].str.replace('[^0-9.]', '', regex=True).astype(float)

# Print the summary of the data
print(df_automobiles.describe())
```

```
        Unnamed: 0         Year  Kilometers_Driven      Mileage       Engine  \
count  5847.000000  5847.000000       5.847000e+03  5845.000000  5811.000000
mean   3013.181461  2013.448435       5.841013e+04    18.158496  1631.552573
std    1736.398890     3.194949       9.237971e+04     4.358246   601.972587
min       1.000000  1998.000000       1.710000e+02     0.000000    72.000000
25%    1509.500000  2012.000000       3.346750e+04    15.260000  1198.000000
50%    3015.000000  2014.000000       5.257600e+04    18.190000  1497.000000
75%    4517.500000  2016.000000       7.249050e+04    21.100000  1991.000000
max    6018.000000  2019.000000       6.500000e+06    28.400000  5998.000000

             Power        Seats   New_Price         Price
count  5811.000000  5809.000000  815.000000  5847.000000
mean    113.803144     5.286452   20.484564     9.653742
std      53.896719     0.806668   20.248764    11.275966
min      34.200000     2.000000    1.000000     0.440000
25%      78.000000     5.000000    7.880000     3.550000
50%      98.600000     5.000000   11.480000     5.750000
75%     139.010000     5.000000   24.015000    10.250000
max     560.000000    10.000000   99.920000   160.000000
```

c). Change the categorical variables (“Fuel_Type” and “Transmission”) into numerical one hotencoded value.

```
#Load the encoded results csv
df_automobiles = pd.read_csv('/content/results_cars_encoded.csv')
df_automobiles['Current_Age'] = pd.to_datetime('today').year - df_automobiles['Year']
df_automobiles
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Current_Age
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	0	1	First	19.67	1582.0	126.20	5.0	12.50	9
1	Honda Jazz V	Chennai	2011	46000	1	1	First	13.00	1199.0	88.70	5.0	4.50	13
2	Maruti Ertiga VDI	Chennai	2012	87000	0	1	First	20.77	1248.0	88.76	7.0	6.00	12
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	0	0	Second	15.20	1968.0	140.80	5.0	17.74	11
4	Nissan Micra Diesel XV	Jaipur	2013	86999	0	1	First	23.08	1461.0	63.10	5.0	3.50	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5842	Maruti Swift VDI	Delhi	2014	27365	0	1	First	28.40	1248.0	74.00	5.0	4.75	10
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	0	1	First	24.40	1120.0	71.00	5.0	4.00	9
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	0	1	Second	14.00	2498.0	112.00	8.0	2.90	12
5845	Maruti Wagon R VXI	Kolkata	2013	46000	1	1	First	18.90	998.0	67.10	5.0	2.65	11
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	0	1	First	25.44	936.0	57.60	5.0	2.50	13

5847 rows × 13 columns

Next steps: Generate code with df_automobiles View recommended plots

Fuel_Type	Transmission
0	1
1	1
0	1
0	0
0	1
...	...
0	1
0	1
0	1
1	1
0	1

d). Create one more feature and add this column to the dataset. you can calculate the current age of the car by subtracting "Year" value from the current year.

```
# Write the data of cars age to a new file
df_automobiles.to_csv('/content/cars_current_age.csv', index=False)
df_automobiles
```

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Current_Age
0	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	0	1	First	19.67	1582.0	126.20	5.0	12.50	9
1	Honda Jazz V	Chennai	2011	46000	1	1	First	13.00	1199.0	88.70	5.0	4.50	13
2	Maruti Ertiga VDI	Chennai	2012	87000	0	1	First	20.77	1248.0	88.76	7.0	6.00	12
3	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	0	0	Second	15.20	1968.0	140.80	5.0	17.74	11
4	Nissan Micra Diesel XV	Jaipur	2013	86999	0	1	First	23.08	1461.0	63.10	5.0	3.50	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...
5842	Maruti Swift VDI	Delhi	2014	27365	0	1	First	28.40	1248.0	74.00	5.0	4.75	10
5843	Hyundai Xcent 1.1 CRDi S	Jaipur	2015	100000	0	1	First	24.40	1120.0	71.00	5.0	4.00	9
5844	Mahindra Xylo D4 BSIV	Jaipur	2012	55000	0	1	Second	14.00	2498.0	112.00	8.0	2.90	12
5845	Maruti Wagon R VXI	Kolkata	2013	46000	1	1	First	18.90	998.0	67.10	5.0	2.65	11
5846	Chevrolet Beat Diesel	Hyderabad	2011	47000	0	1	First	25.44	936.0	57.60	5.0	2.50	13

5847 rows × 13 columns

Price	Current_Age
12.50	9
4.50	13
6.00	12
17.74	11
3.50	11
...	...
4.75	10
4.00	9
2.90	12
2.65	11
2.50	13

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

Select:

```
# Display the results
print("Selected Columns:")
print(selected_columns.head())
```

```
Selected Columns:
                                 Name    Location  Year  Kilometers_Driven  \
0  Hyundai Creta 1.6 CRDi SX Option        Pune  2015              41000
1                      Honda Jazz V     Chennai  2011              46000
2                 Maruti Ertiga VDI     Chennai  2012              87000
3   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013              40670
4            Nissan Micra Diesel XV      Jaipur  2013              86999

   Fuel_Type  Transmission Owner_Type  Mileage  Engine   Power  Seats  Price
0          0             1      First    19.67  1582.0  126.20    5.0  12.50
1          1             1      First    13.00  1199.0   88.70    5.0   4.50
2          0             1      First    20.77  1248.0   88.76    7.0   6.00
3          0             0     Second    15.20  1968.0  140.80    5.0  17.74
4          0             1      First    23.08  1461.0   63.10    5.0   3.50
```

Filter:

```
[33] print("\nFiltered Data (Petrol only):")
     print(filtered_data.head())
```

```
Filtered Data (Petrol only):
Empty DataFrame
Columns: [Name, Location, Year, Kilometers_Driven, Fuel_Type, Transmission, Owner_Type, Mileage, Engine, Power, Seats, Price, Current_Age]
Index: []
```

Rename:

```
print("\nRenamed Columns:")
print(renamed_columns.head())
```

```
Renamed Columns:
                                 Name    Location  Year  Kilometers_Driven  \
0  Hyundai Creta 1.6 CRDi SX Option        Pune  2015              41000
1                      Honda Jazz V     Chennai  2011              46000
2                 Maruti Ertiga VDI     Chennai  2012              87000
3   Audi A4 New 2.0 TDI Multitronic  Coimbatore  2013              40670
4            Nissan Micra Diesel XV      Jaipur  2013              86999

   Fuel_Type  TransType OwnerType  Mileage  Engine   Power  Seats  Price  \
0          0          1     First    19.67  1582.0  126.20    5.0  12.50
1          1          1     First    13.00  1199.0   88.70    5.0   4.50
2          0          1     First    20.77  1248.0   88.76    7.0   6.00
3          0          0    Second    15.20  1968.0  140.80    5.0  17.74
4          0          1     First    23.08  1461.0   63.10    5.0   3.50

   Current_Age
0            9
1           13
2           12
3           11
4           11
```

Mutate:

```
[35] print("\nMutated Data (Added Power_per_Cylinder column):")
     print(df_cars[['Power', 'Seats', 'Power_per_Cylinder']].head())
```

```
Mutated Data (Added Power_per_Cylinder column):
     Power  Seats  Power_per_Cylinder
0  126.20    5.0               25.24
1   88.70    5.0               17.74
2   88.76    7.0               12.68
3  140.80    5.0               28.16
4   63.10    5.0               12.62
```

Sort/Arrange:

```
print("\nSorted Data (Descending order of Year):")
print(sorted_data.head())
```

```
Sorted Data (Descending order of Year):
                                Name    Location  Year  Kilometers_Driven  \
5405        Renault KWID RXT Optional       Kochi  2019               6568
942   Ford Endeavour 2.2 Trend AT 4X2       Kochi  2019              11209
5533        Maruti Omni 5 Seater BSIV  Coimbatore  2019               4721
770             Mahindra XUV500 W9 AT  Coimbatore  2019              19654
4267       Maruti Swift Dzire AMT ZDI     Chennai  2019              65000

      Fuel_Type  Transmission Owner_Type  Mileage  Engine  Power  Seats  \
5405          1             1      First    25.17   799.0   53.3    5.0
942           0             0      First    12.62  2198.0  158.0    7.0
5533          1             1      First    14.00   796.0   35.0    5.0
770           0             0      First    14.00  2179.0  155.0    7.0
4267          0             0      First    26.59  1248.0   74.0    5.0

      Price  Current_Age  Power_per_Cylinder
5405   5.09            5           10.660000
942   31.15            5           22.571429
5533   4.11            5            7.000000
770   17.63            5           22.142857
4267   6.75            5           14.800000
```

Group By:

```
print("\nGrouped Data (Mean Price and Total Kilometers_Driven per Location):")
print(grouped_data.head())
```

```
Grouped Data (Mean Price and Total Kilometers_Driven per Location):
                Price  Kilometers_Driven
Location
Ahmedabad    8.567248           11984485
Bangalore   13.482670           20496207
Chennai      7.958340           43075555
Coimbatore  15.160206           29625311
Delhi        9.881944           30880339
```

Conclusion:

The dataset consists of attributes of used cars, with the target variable being the price measured in lakhs. Missing values were addressed by imputing them or dropping them based on data integrity. Units were removed from specific attributes, and categorical variables were transformed into numerical one-hot encoded values. Additionally, a new feature indicating the current age of the car was created. Various data manipulation operations were performed for analysis and summary.