DATA SCIENCE LAB EXPERIMENT 1

Name: Sai Rane Div: D15C Roll No: 43

AIM: Introduction to Data science and Data preparation using Pandas steps.

THEORY:
Pandas: Pandas is an open-source Python library used for data manipulation and analysis. It provides high-performance data structures and functions for efficiently handling structured data.

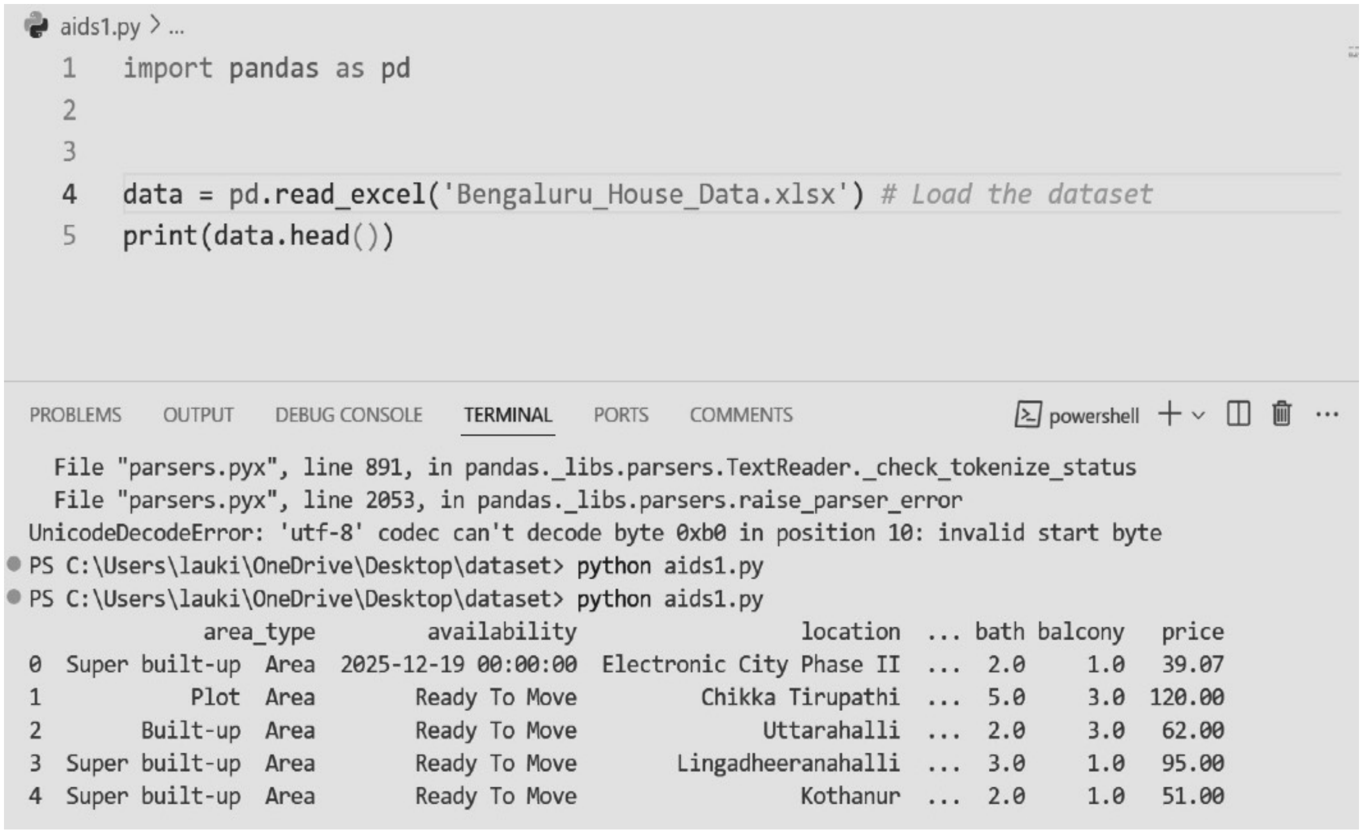
Key Pandas Functions for Data Cleaning

1. Handling Missing Data
   - df.isnull().sum() → Check the number of missing values in each column.
   - df.dropna() → Remove rows with missing values.
   - df.fillna(value, inplace=True) → Fill missing values with a specific value (e.g., mean or median).
2. Removing Duplicates
   - df.duplicated() → Identify duplicate rows.
   - df.drop_duplicates(inplace=True) → Remove duplicate rows.
3. Handling Incorrect Data Formats
   - df['column'] = pd.to_datetime(df['column']) → Convert a column to a datetime format.
   - df['column'] = df['column'].astype(int/float/str) → Change data types.

Topic: Bengaluru Housing Prices

DATA SCIENCE LAB EXPERIMENT 1

1. Loading Data in Pandas:



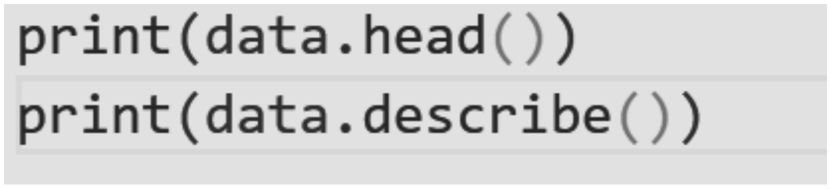
aids1.py > ...

```
import pandas as pd


data = pd.read_excel('Bengaluru_House_Data.xlsx') # Load the dataset
print(data.head())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS powershell

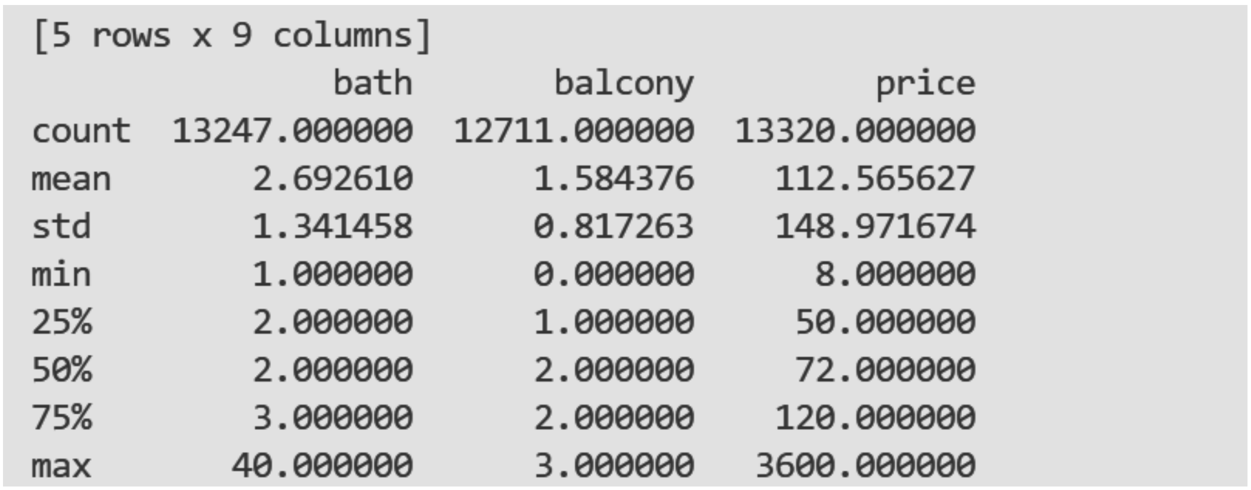
```
  File "parsers.pyx", line 891, in pandas._libs.parsers.TextReader._check_tokenize_status
  File "parsers.pyx", line 2053, in pandas._libs.parsers.raise_parser_error
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xb0 in position 10: invalid start byte
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
             area_type         availability               location  ... bath balcony   price
0  Super built-up  Area  2025-12-19 00:00:00  Electronic City Phase II  ...  2.0     1.0   39.07
1            Plot  Area        Ready To Move        Chikka Tirupathi  ...  5.0     3.0  120.00
2        Built-up  Area        Ready To Move             Uttarahalli  ...  2.0     3.0   62.00
3  Super built-up  Area        Ready To Move      Lingadheeranahalli  ...  3.0     1.0   95.00
4  Super built-up  Area        Ready To Move                Kothanur  ...  2.0     1.0   51.00
```

2. Description of the Dataset:



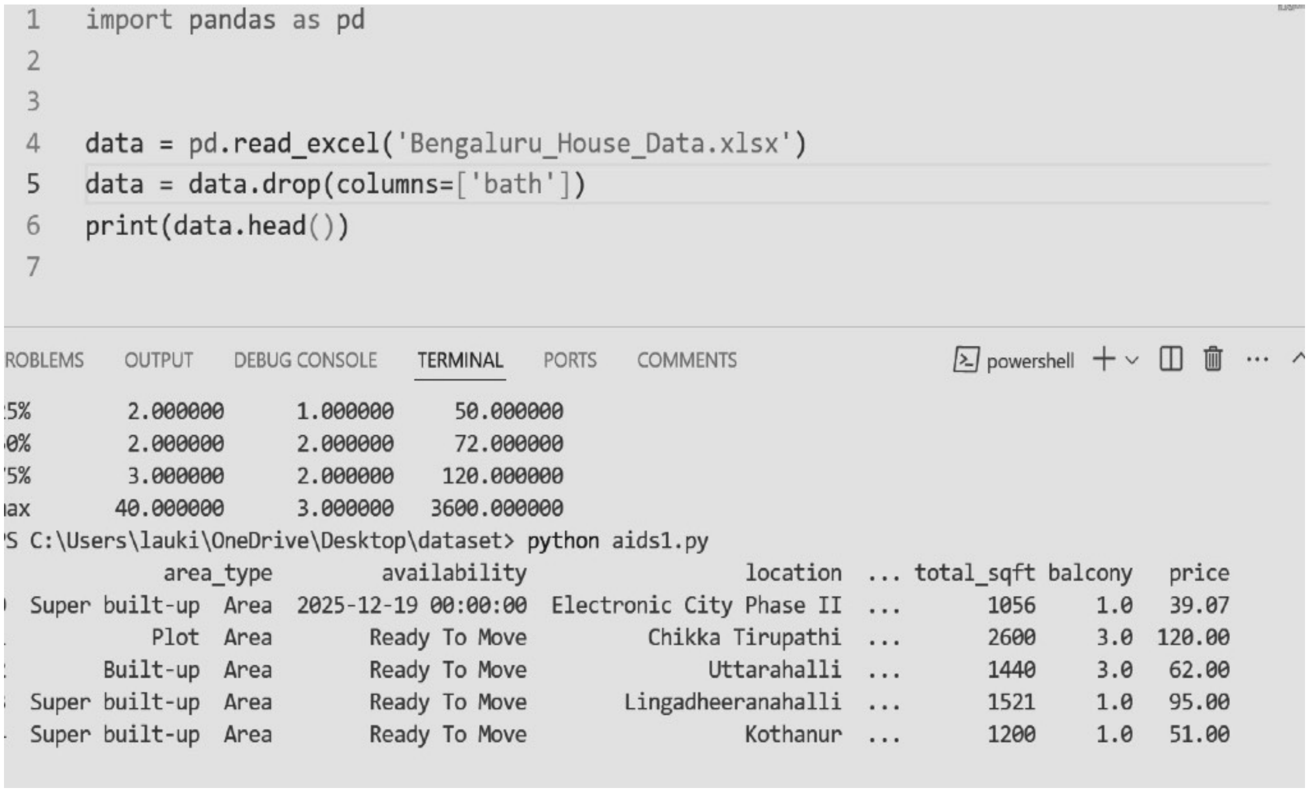
```
print(data.head())
print(data.describe())
```

DATA SCIENCE LAB EXPERIMENT 1



```
[5 rows x 9 columns]
               bath       balcony          price
count  13247.000000  12711.000000   13320.000000
mean       2.692610      1.584376     112.565627
std        1.341458      0.817263     148.971674
min        1.000000      0.000000       8.000000
25%        2.000000      1.000000      50.000000
50%        2.000000      2.000000      72.000000
75%        3.000000      2.000000     120.000000
max       40.000000      3.000000    3600.000000
```

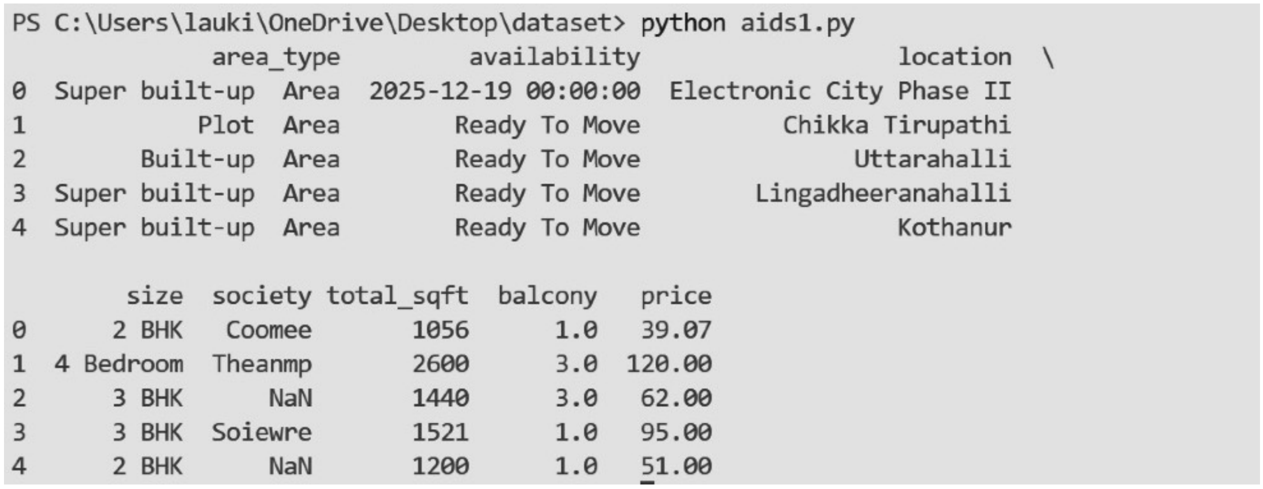
3. Drop columns that are not useful:



```python
import pandas as pd



data = pd.read_excel('Bengaluru_House_Data.xlsx')
data = data.drop(columns=['bath'])
print(data.head())

```

```
:5%       2.000000      1.000000      50.000000
0%        2.000000      2.000000      72.000000
'5%       3.000000      2.000000     120.000000
ax       40.000000      3.000000    3600.000000
'S C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
             area_type         availability              location ... total_sqft balcony   price
)  Super built-up  Area  2025-12-19 00:00:00  Electronic City Phase II ...       1056     1.0   39.07
.            Plot  Area         Ready To Move      Chikka Tirupathi ...       2600     3.0  120.00
:        Built-up  Area         Ready To Move           Uttarahalli ...       1440     3.0   62.00
;  Super built-up  Area         Ready To Move    Lingadheeranahalli ...       1521     1.0   95.00
.  Super built-up  Area         Ready To Move              Kothanur ...       1200     1.0   51.00
```

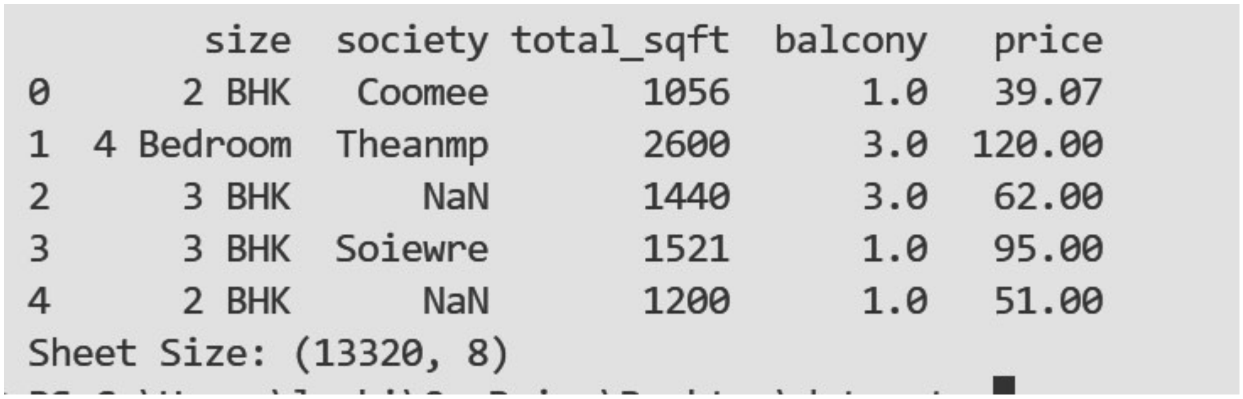
After dropping Number of bathrooms column:



```
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
              area_type         availability                  location  \
0  Super built-up  Area  2025-12-19 00:00:00  Electronic City Phase II
1            Plot  Area         Ready To Move          Chikka Tirupathi
2        Built-up  Area         Ready To Move               Uttarahalli
3  Super built-up  Area         Ready To Move        Lingadheeranahalli
4  Super built-up  Area         Ready To Move                  Kothanur

        size  society total_sqft  balcony   price
0      2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
2      3 BHK      NaN       1440      3.0   62.00
3      3 BHK  Soiewre       1521      1.0   95.00
4      2 BHK      NaN       1200      1.0   51.00
```

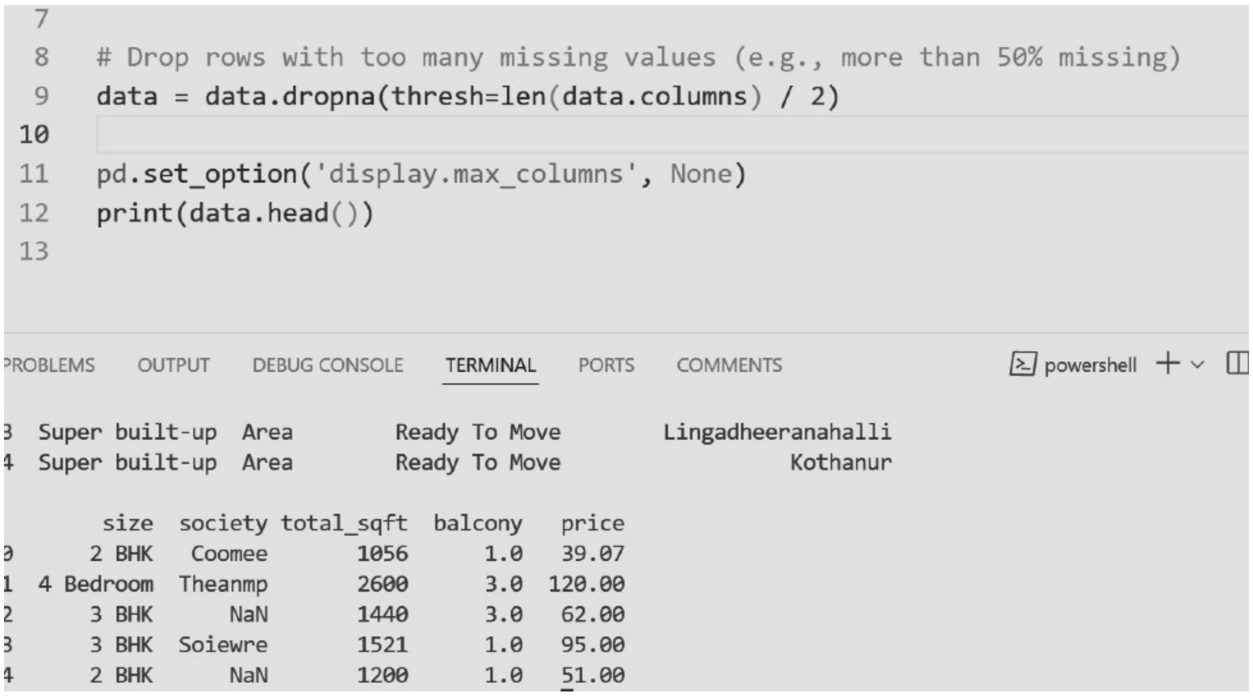
4. Drop rows with maximum missing values:

Before Dropping:



```
        size  society total_sqft  balcony   price
0      2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
2      3 BHK      NaN       1440      3.0   62.00
3      3 BHK  Soiewre       1521      1.0   95.00
4      2 BHK      NaN       1200      1.0   51.00
Sheet Size: (13320, 8)
```

DATA SCIENCE LAB EXPERIMENT 1

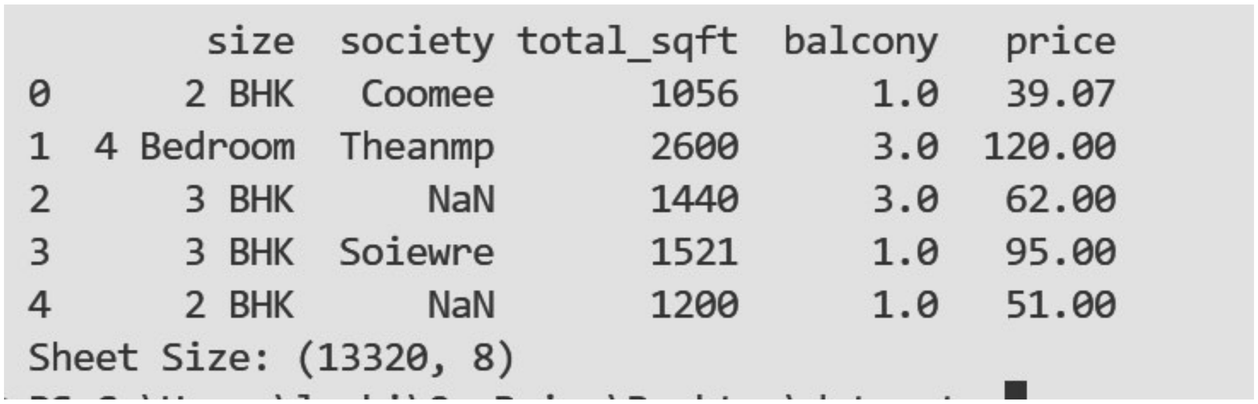


```python

# Drop rows with too many missing values (e.g., more than 50% missing)
data = data.dropna(thresh=len(data.columns) / 2)

pd.set_option('display.max_columns', None)
print(data.head())

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS powershell

```
3  Super built-up  Area        Ready To Move        Lingadheeranahalli
4  Super built-up  Area        Ready To Move                 Kothanur

        size  society total_sqft  balcony   price
0      2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
2      3 BHK      NaN       1440      3.0   62.00
3      3 BHK  Soiewre       1521      1.0   95.00
4      2 BHK      NaN       1200      1.0   51.00
```

After Dropping:

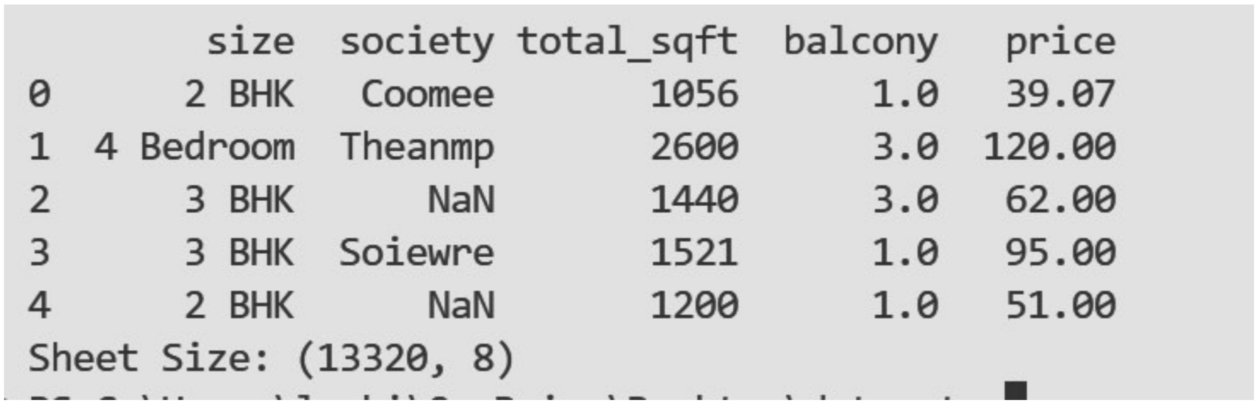


```
        size  society total_sqft  balcony   price
0      2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
2      3 BHK      NaN       1440      3.0   62.00
3      3 BHK  Soiewre       1521      1.0   95.00
4      2 BHK      NaN       1200      1.0   51.00
Sheet Size: (13320, 8)
```

Since there are no rows with maximum missing values (more than 50% of the cells being empty), no rows were dropped.

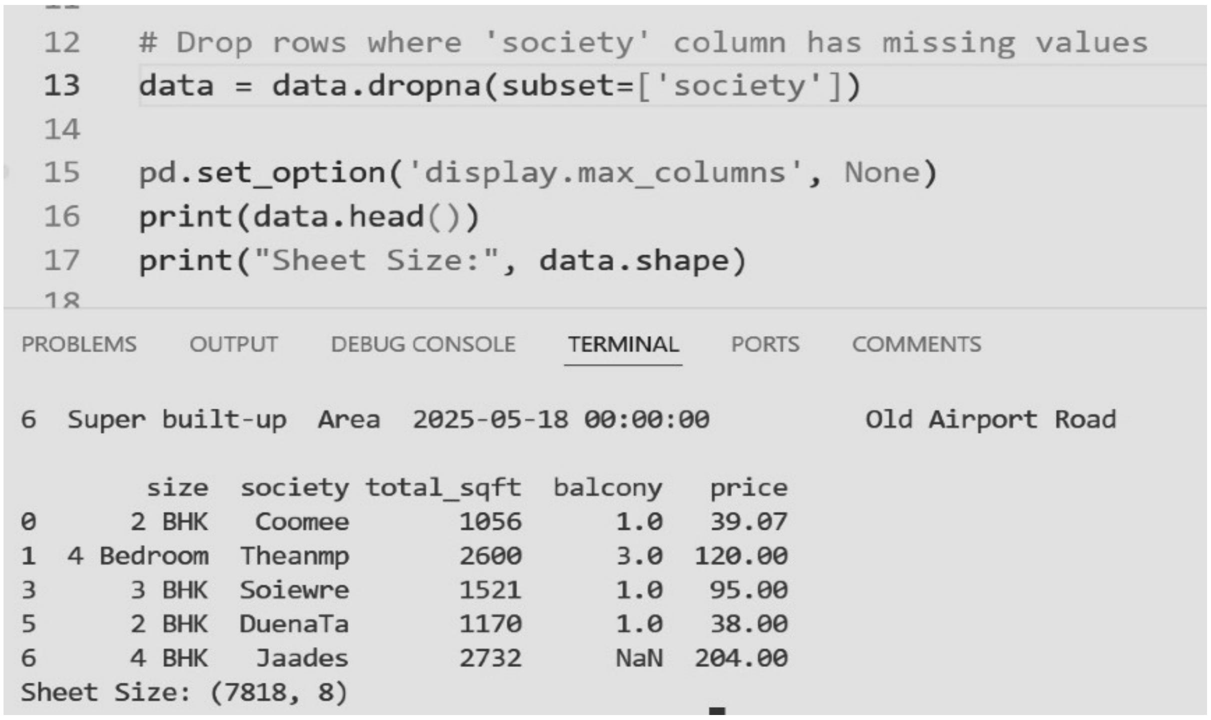
5. Take care of missing data:
   Dropping rows if society name is missing

   Before Dropping:



```
       size  society total_sqft  balcony   price
0     2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
2     3 BHK      NaN       1440      3.0   62.00
3     3 BHK  Soiewre       1521      1.0   95.00
4     2 BHK      NaN       1200      1.0   51.00
Sheet Size: (13320, 8)
```

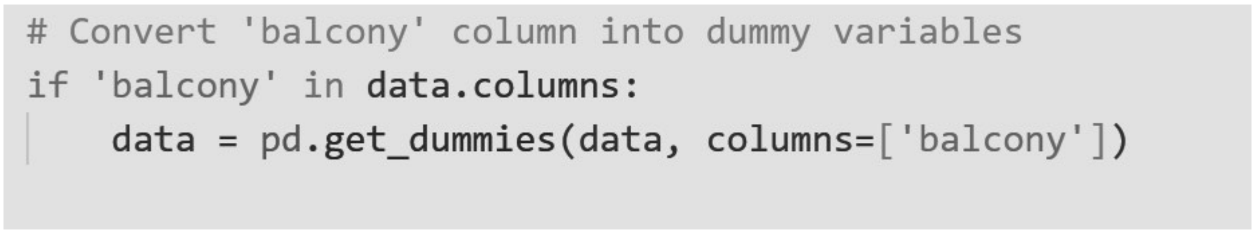
After Dropping:



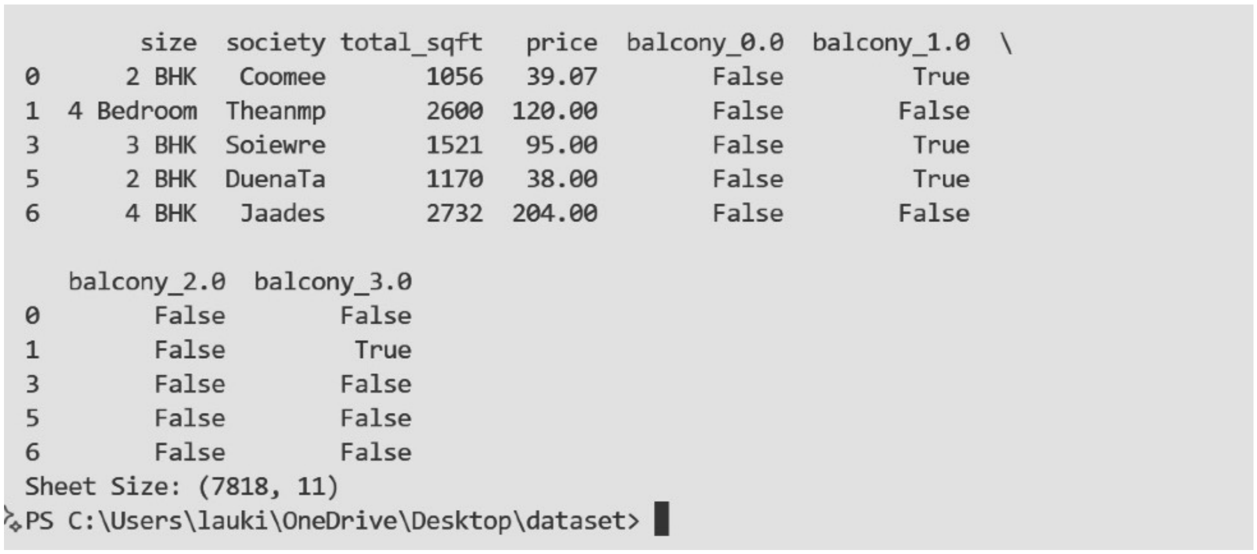
```
# Drop rows where 'society' column has missing values
data = data.dropna(subset=['society'])

pd.set_option('display.max_columns', None)
print(data.head())
print("Sheet Size:", data.shape)
```

```
6  Super built-up  Area  2025-05-18 00:00:00        Old Airport Road

       size  society total_sqft  balcony   price
0     2 BHK   Coomee       1056      1.0   39.07
1  4 Bedroom  Theanmp       2600      3.0  120.00
3     3 BHK  Soiewre       1521      1.0   95.00
5     2 BHK  DuenaTa       1170      1.0   38.00
6     4 BHK   Jaades       2732      NaN  204.00
Sheet Size: (7818, 8)
```

6. Creating Dummy variables for the balcony column:
   In data science, dummy values (or dummy variables) are used to represent categorical data in a numerical format so that machine learning models can process them effectively. Most machine learning models cannot handle categorical data directly. Converting categorical variables into dummy (binary) variables allows models to interpret them numerically.

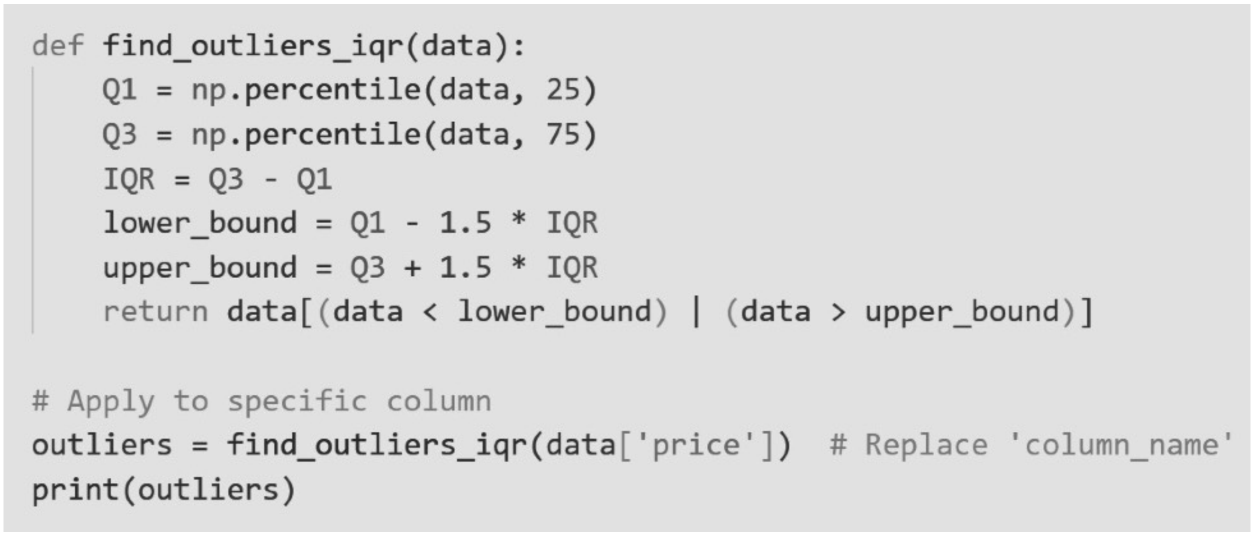


```python
# Convert 'balcony' column into dummy variables
if 'balcony' in data.columns:
    data = pd.get_dummies(data, columns=['balcony'])
```

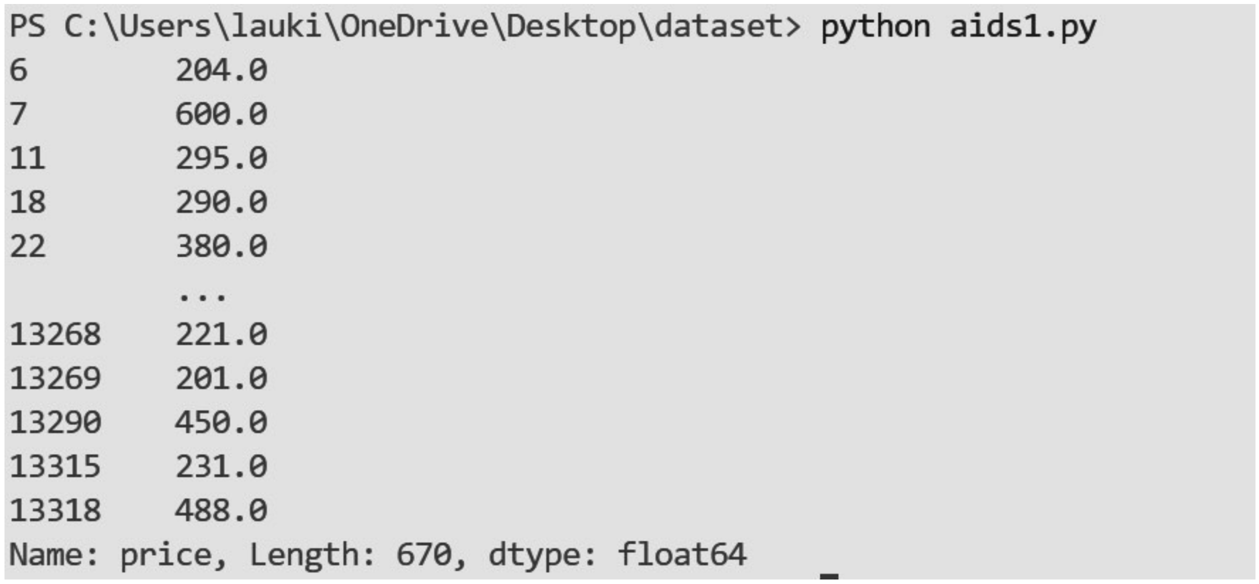


```
        size  society total_sqft   price  balcony_0.0  balcony_1.0  \
0      2 BHK   Coomee       1056   39.07        False         True
1  4 Bedroom  Theanmp       2600  120.00        False        False
3      3 BHK  Soiewre       1521   95.00        False         True
5      2 BHK  DuenaTa       1170   38.00        False         True
6      4 BHK   Jaades       2732  204.00        False        False

   balcony_2.0  balcony_3.0
0        False        False
1        False         True
3        False        False
5        False        False
6        False        False
Sheet Size: (7818, 11)
PS C:\Users\lauki\OneDrive\Desktop\dataset>
```

7. Finding Outliers:
   The IQR method is used to find the outliers manually. The IQR (Interquartile Range) method is a statistical technique used to detect and handle outliers in a dataset. It is based on the spread of the middle 50% of the data.



```python
def find_outliers_iqr(data):
    Q1 = np.percentile(data, 25)
    Q3 = np.percentile(data, 75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return data[(data < lower_bound) | (data > upper_bound)]

# Apply to specific column
outliers = find_outliers_iqr(data['price'])  # Replace 'column_name'
print(outliers)
```

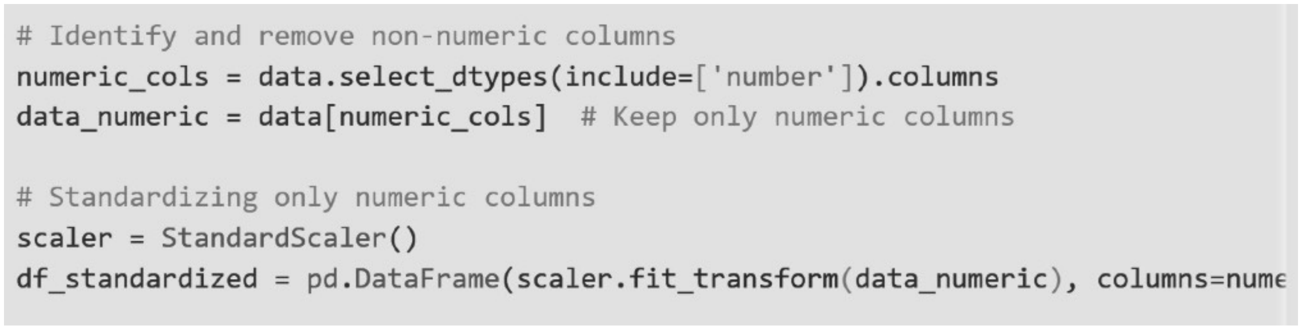


```
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
6        204.0
7        600.0
11       295.0
18       290.0
22       380.0
         ...
13268    221.0
13269    201.0
13290    450.0
13315    231.0
13318    488.0
Name: price, Length: 670, dtype: float64
```

8. Standardization and Normalization of columns

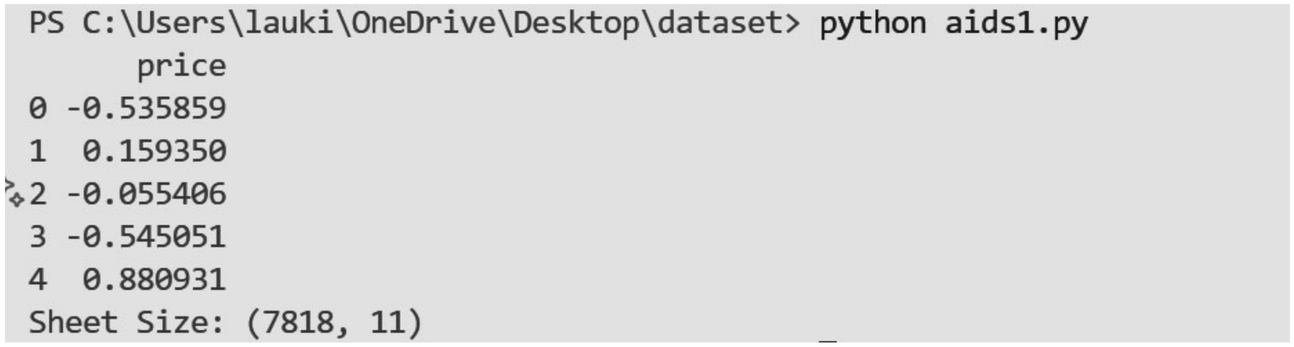
Standardization is to ensure that all the features are transformed such that the mean is 0 and standard deviation is 1.

DATA SCIENCE LAB EXPERIMENT 1



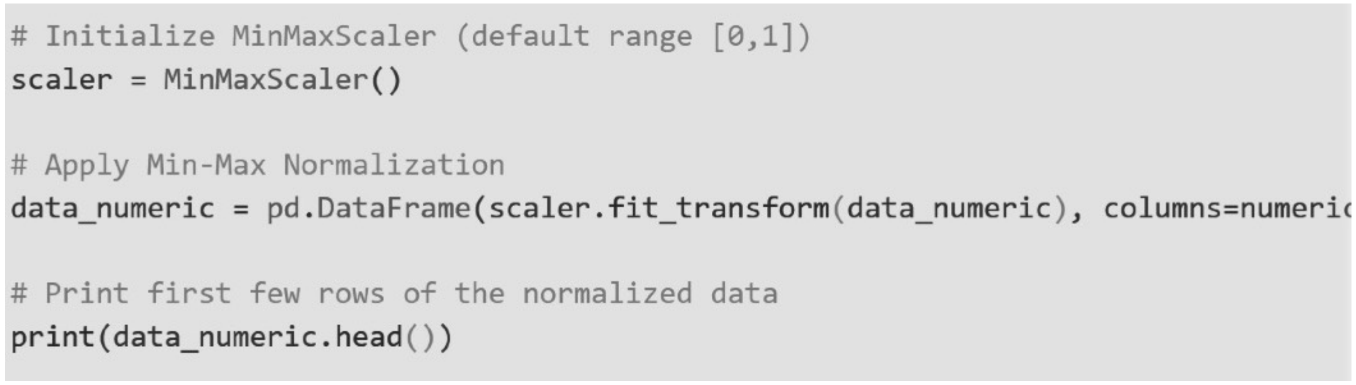
```
# Identify and remove non-numeric columns
numeric_cols = data.select_dtypes(include=['number']).columns
data_numeric = data[numeric_cols]  # Keep only numeric columns

# Standardizing only numeric columns
scaler = StandardScaler()
df_standardized = pd.DataFrame(scaler.fit_transform(data_numeric), columns=nume
```

This code will help to print the standardized values:
X(standardized)= (X−μ)/σ



```
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
      price
0 -0.535859
1  0.159350
2 -0.055406
3 -0.545051
4  0.880931
Sheet Size: (7818, 11)
```

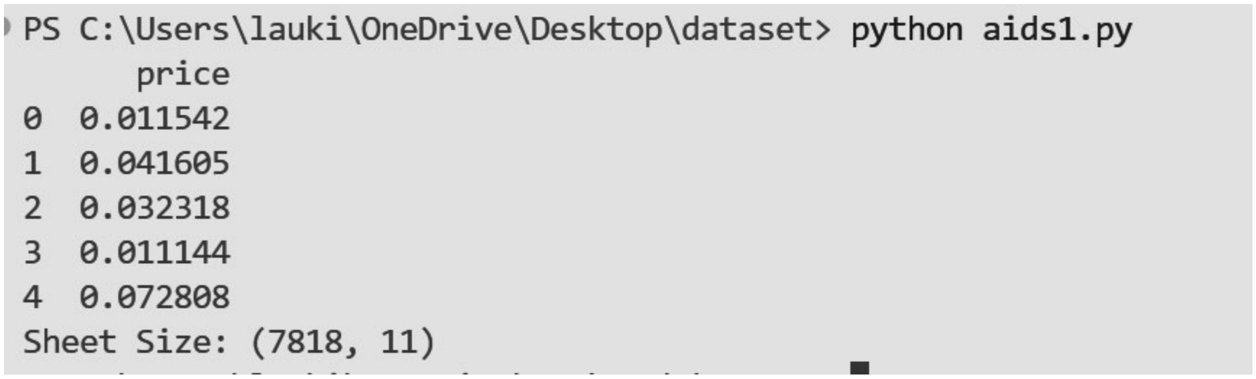
Normalization is the process of scaling all the features to a range [0, 1]. It is also called min-max scaling.



```
# Initialize MinMaxScaler (default range [0,1])
scaler = MinMaxScaler()

# Apply Min-Max Normalization
data_numeric = pd.DataFrame(scaler.fit_transform(data_numeric), columns=numeric

# Print first few rows of the normalized data
print(data_numeric.head())
```

This code will help us to perform min max normalization and scale the features to range between [0, 1].
X(normalized) = (X - X(min))/(X(max) - X(min))

DATA SCIENCE LAB EXPERIMENT 1



```
PS C:\Users\lauki\OneDrive\Desktop\dataset> python aids1.py
      price
0  0.011542
1  0.041605
2  0.032318
3  0.011144
4  0.072808
Sheet Size: (7818, 11)
```

Conclusion: Thus we have successfully prepared the data from an unclean dataset using Pandas. It helps us in loading data from various file formats (e.g., CSV, Excel, SQL) into a structured DataFrame for easier manipulation, cleaning, and analysis. Removing irrelevant or redundant columns helps to reduce dimensionality and focus on important features, improving model performance. Remove irrelevant or redundant columns to reduce dimensionality and focus on important features, improving model performance. Remove irrelevant or redundant columns to reduce dimensionality and focus on important features, improving model performance. This experiment has helped us to understand these concepts efficiently.