Exploratory Data Analysis (EDA) - Bengaluru House Price Dataset

1. Import Libraries

```
!pip install seaborn

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

2. Load and Inspect Data

```
df = pd.read_csv('/content/Bengaluru_House_Data.csv')
print("Shape of dataset:", df.shape)
df.info()
df.head()
```

3. Check for Missing Values

```
df.isnull().sum()
```

4. Categorical Value Counts

```
df['area_type'].value_counts()
df['availability'].unique()
df['location'].nunique()
df['size'].value_counts()
df['bath'].value_counts()
```

5. Handle 'availability' Column

```
def handle_availability(value):
    try:
        return value.split('-')[1]
    except:
        return value

df['availability'] = df['availability'].apply(handle_availability)

def handle_availability2(value):
    if value in ['Jan','Feb','Mar']:
        return 'Q1'
    elif value in ['Apr','May','Jun']:
        return 'Q2'
    elif value in ['Jul','Aug','Sep']:
```

```
        return 'Q3'
    elif value in ['Oct','Nov','Dec']:
        return 'Q4'
    else:
        return value

df['availability'] = df['availability'].apply(handle_availability2)
```

6. Handle 'location' Column

```
location_counts = df['location'].value_counts()
location_under_10 = location_counts[location_counts <= 10]
df['location'] = df['location'].apply(lambda x: 'other' if x in location_under_10 else x)
```

7. Process 'size' Column

```
def extract_bhk(value):
    try:
        return int(value.split(' ')[0])
    except:
        return None

def extract_type(value):
    try:
        return value.split(' ')[1]
    except:
        return None

df['size_num'] = df['size'].apply(extract_bhk)
df['size_type'] = df['size'].apply(extract_type)
df.drop('size', axis=1, inplace=True)
```

8. Handle 'total_sqft' Column

```
import re

def convert_sqft(value):
    try:
        nums = re.findall(r"[-+]?(?:\d*\.*\d+)", str(value))
        if len(nums) == 2:
            return (float(nums[0]) + float(nums[1])) / 2
        return float(nums[0])
    except:
        return np.nan

df['total_sqft'] = df['total_sqft'].apply(convert_sqft)
```

Exploratory Data Analysis (EDA) - Bengaluru House Price Dataset

9. Drop Unnecessary Columns & Handle Nulls

```
df.drop('society', axis=1, inplace=True)
df.dropna(inplace=True)
```

10. Outlier Removal

```
Q1 = df['total_sqft'].quantile(0.25)
Q3 = df['total_sqft'].quantile(0.75)
IQR = Q3 - Q1
ll = Q1 - 1.5 * IQR
ul = Q3 + 1.5 * IQR
df = df[(df['total_sqft'] >= ll) & (df['total_sqft'] <= ul)]
df = df[df['bath'] <= 4]
```

11. Visualizations

```
sns.histplot(df['price'], kde=True)
sns.countplot(x='size_num', data=df)
sns.countplot(x='area_type', data=df)
sns.countplot(x='availability', data=df)
plt.scatter(df['total_sqft'], df['price'])
```

12. Final Dataset Info

```
print("Final Shape:", df.shape)
df.info()
```