```
from google.colab import files
uploaded = files.upload()
```

Choose files Real_Estate…Afii.csv.xlsx

- **Real_Estate_Dataset_Afii.csv.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 9209 bytes, last modified: 24/06/2025 - 100% done

Saving Real_Estate_Dataset_Afii.csv.xlsx to Real_Estate_Dataset_Afii.csv.xlsx

```
import pandas as pd

# Load Excel file
df = pd.read_excel("Real_Estate_Dataset_Afii.csv.xlsx")
df.head()
```

	Area (sqft)	Bedrooms	Bathrooms	Location	Price (Lakhs)
0	1200	2	2	Chennai	55
1	1500	3	2	Bangalore	75
2	1800	3	3	Hyderabad	80
3	1000	2	1	Delhi	50
4	2200	4	3	Mumbai	120

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Check data types and missing values
df.info()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Area (sqft)    10 non-null     int64
 1   Bedrooms       10 non-null     int64
 2   Bathrooms      10 non-null     int64
 3   Location       10 non-null     object
 4   Price (Lakhs)  10 non-null     int64
dtypes: int64(4), object(1)
memory usage: 532.0+ bytes
```

	0
Area (sqft)	0
Bedrooms	0
Bathrooms	0
Location	0
Price (Lakhs)	0

dtype: int64

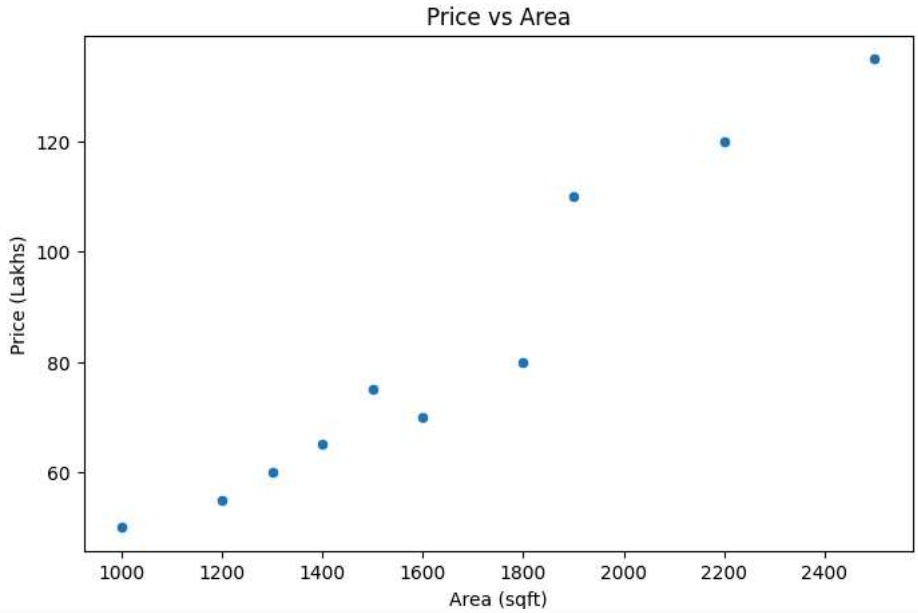
```
df.describe()
```

	Area (sqft)	Bedrooms	Bathrooms	Price (Lakhs)
count	10.000000	10.000000	10.000000	10.000000
mean	1640.000000	2.900000	2.100000	82.000000
std	464.758002	0.737865	0.737865	29.363621
min	1000.000000	2.000000	1.000000	50.000000
25%	1325.000000	2.250000	2.000000	61.250000
50%	1550.000000	3.000000	2.000000	72.500000
75%	1875.000000	3.000000	2.750000	102.500000
max	2500.000000	4.000000	3.000000	135.000000

```
import matplotlib.pyplot as plt
import seaborn as sns
```

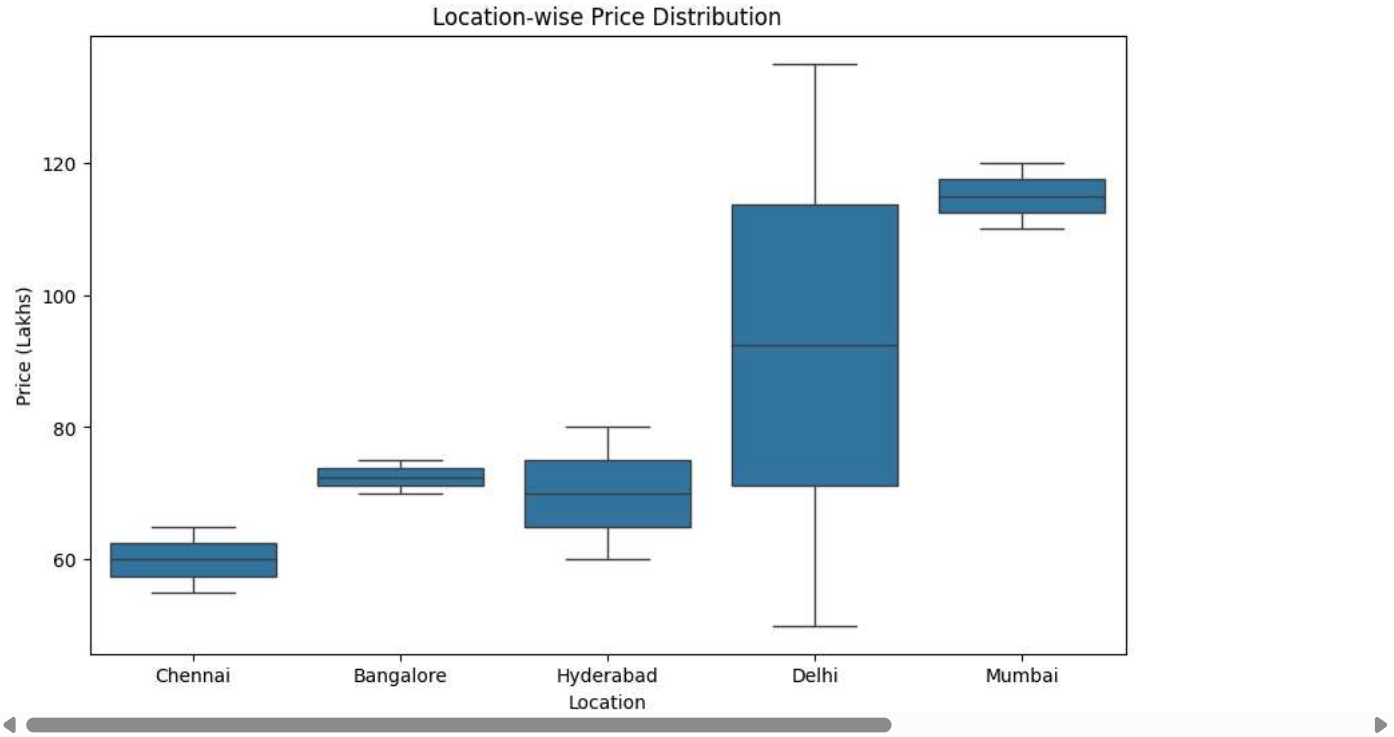
```
plt.figure(figsize=(8,5))
sns.scatterplot(x='Area (sqft)', y='Price (Lakhs)', data=df)
plt.title("Price vs Area")
```

```
Text(0.5, 1.0, 'Price vs Area')
```




```
plt.figure(figsize=(10,6))
sns.boxplot(x='Location', y='Price (Lakhs)', data=df)
plt.title("Location-wise Price Distribution")
```

```
Text(0.5, 1.0, 'Location-wise Price Distribution')
```




```
# Convert categorical 'Location' to numeric codes
df['Location_encoded'] = df['Location'].astype('category').cat.codes

# Check to confirm
df[['Location', 'Location_encoded']].head()
```

```
# Input features
X = df[['Area (sqft)', 'Bedrooms', 'Bathrooms', 'Location_encoded']]

# Target variable
y = df['Price (Lakhs)']
```

```
from sklearn.model_selection import train_test_split

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression

# Create the model
model = LinearRegression()

# Train the model with training data
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Predict on the test set
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

```
print("R² Score:", round(r2, 2))
print("RMSE:", round(rmse, 2))
```

```
R² Score: 0.93
RMSE: 8.2
```