```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import mean_squared_error, r2_score,accuracy_score


df=pd.read_csv("/content/Housing.csv")


df
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hot
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	yes	no	yes	
541	1767150	2400	3	1	1	no	no	no	
542	1750000	3620	2	1	1	yes	no	no	
543	1750000	2910	3	1	1	no	no	no	
544	1750000	3850	3	1	2	yes	no	no	

545 rows × 13 columns

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

```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	

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

```
df.tail()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotw
540	1820000	3000	2	1	1	yes	no	yes	
541	1767150	2400	3	1	1	no	no	no	
542	1750000	3620	2	1	1	yes	no	no	
543	1750000	2910	3	1	1	no	no	no	
544	1750000	3850	3	1	2	yes	no	no	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   price             545 non-null    int64 
 1   area              545 non-null    int64 
 2   bedrooms          545 non-null    int64 
 3   bathrooms         545 non-null    int64 
 4   stories           545 non-null    int64 
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64 
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

```
df.describe()
```

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

```
df.shape
```

```
(545, 13)
```

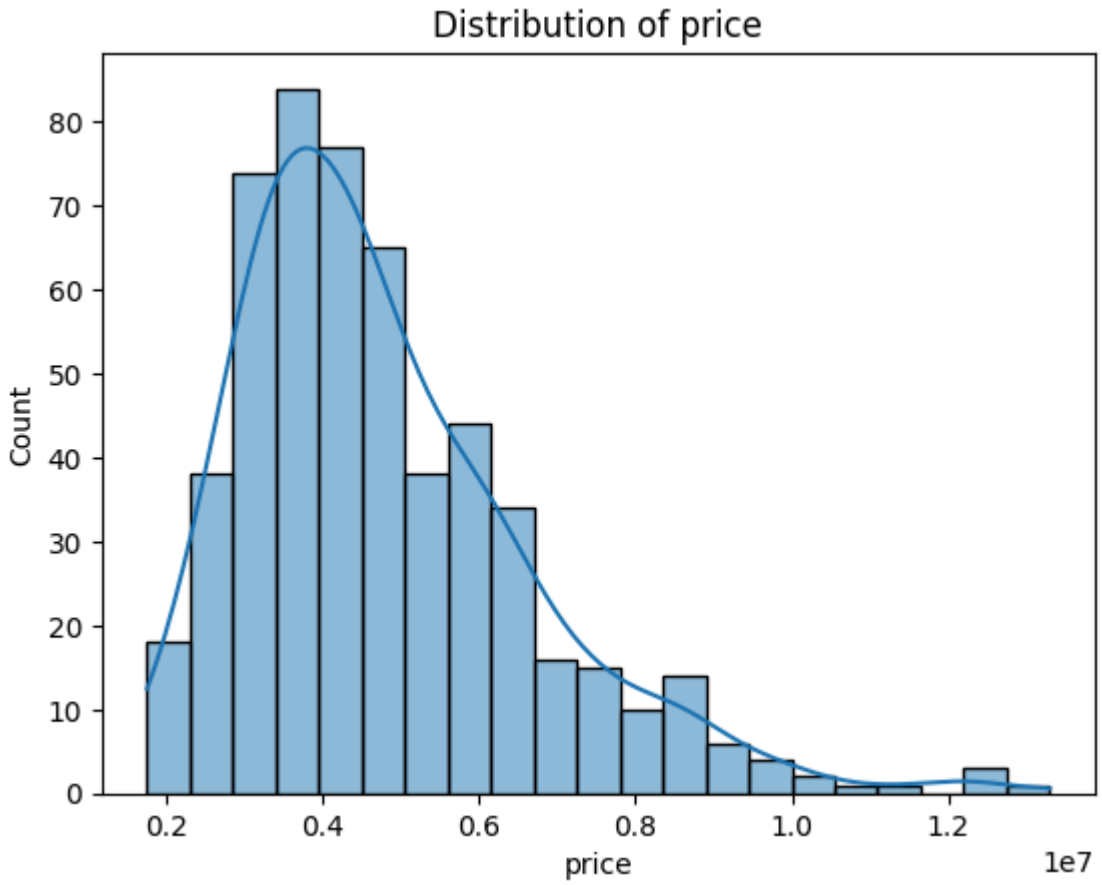
```
df.isnull().sum().sum()
```

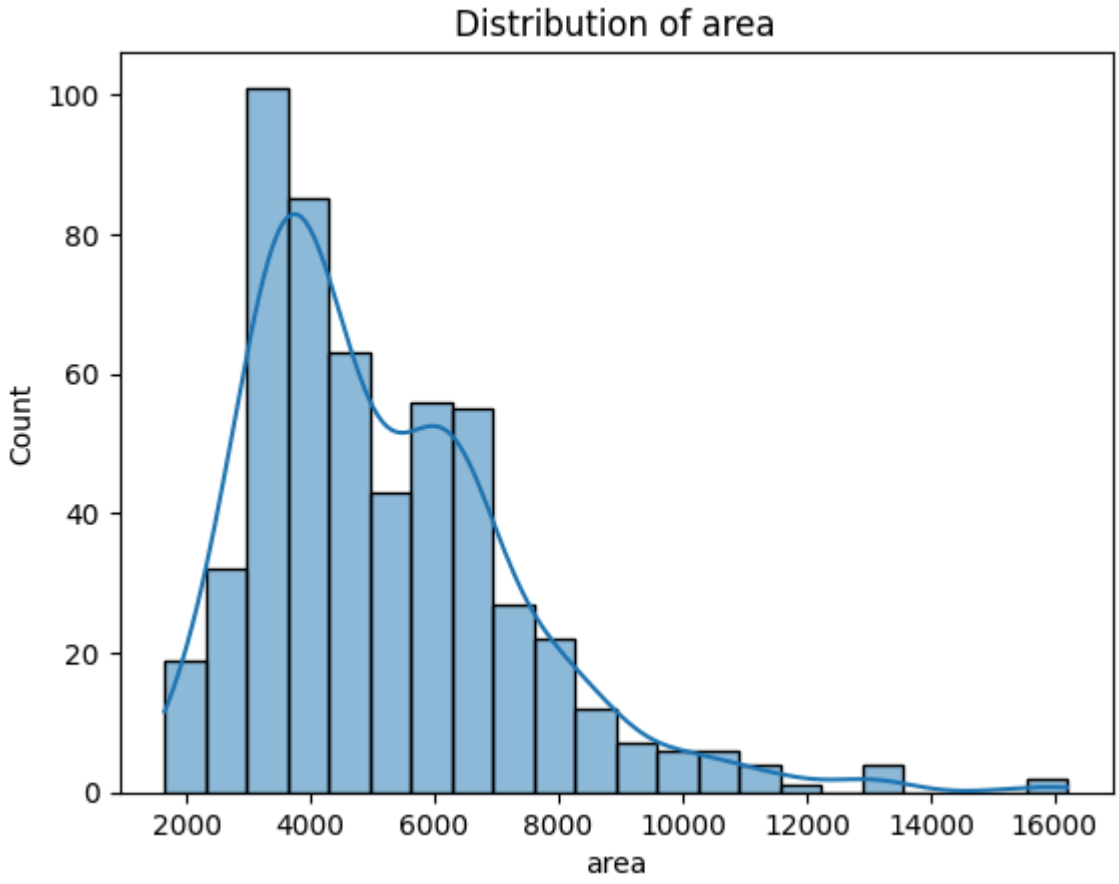
```
np.int64(0)
```

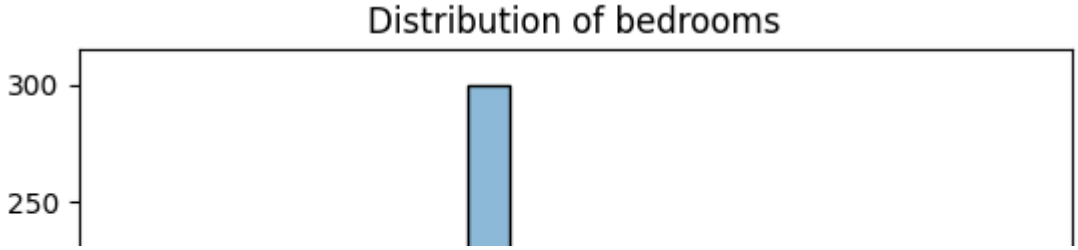
```
df.duplicated().sum()
```

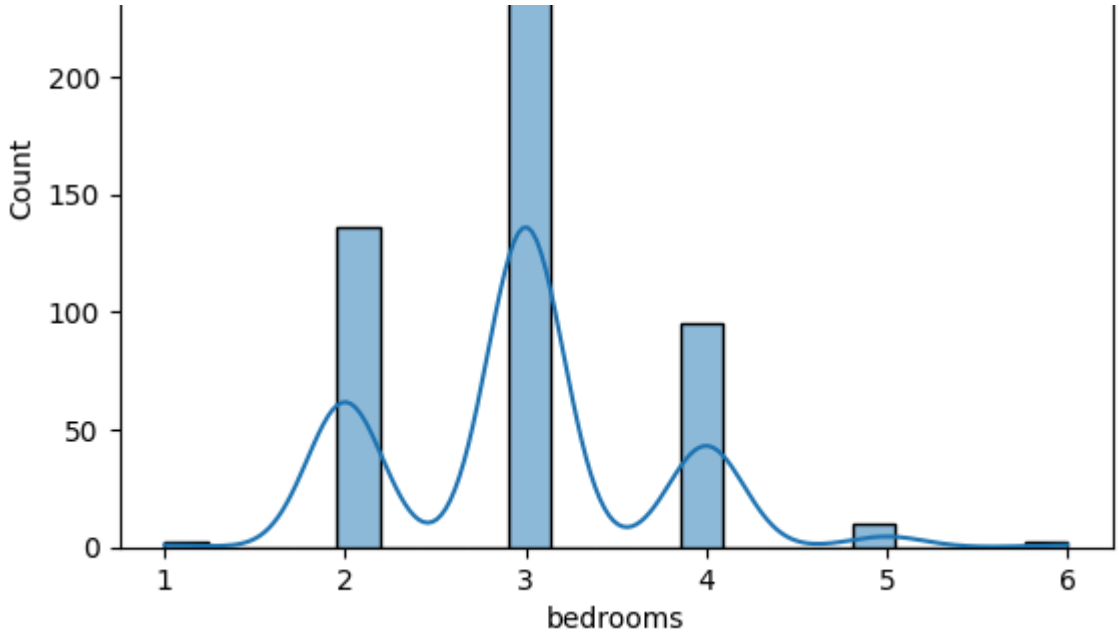
```
np.int64(0)
```

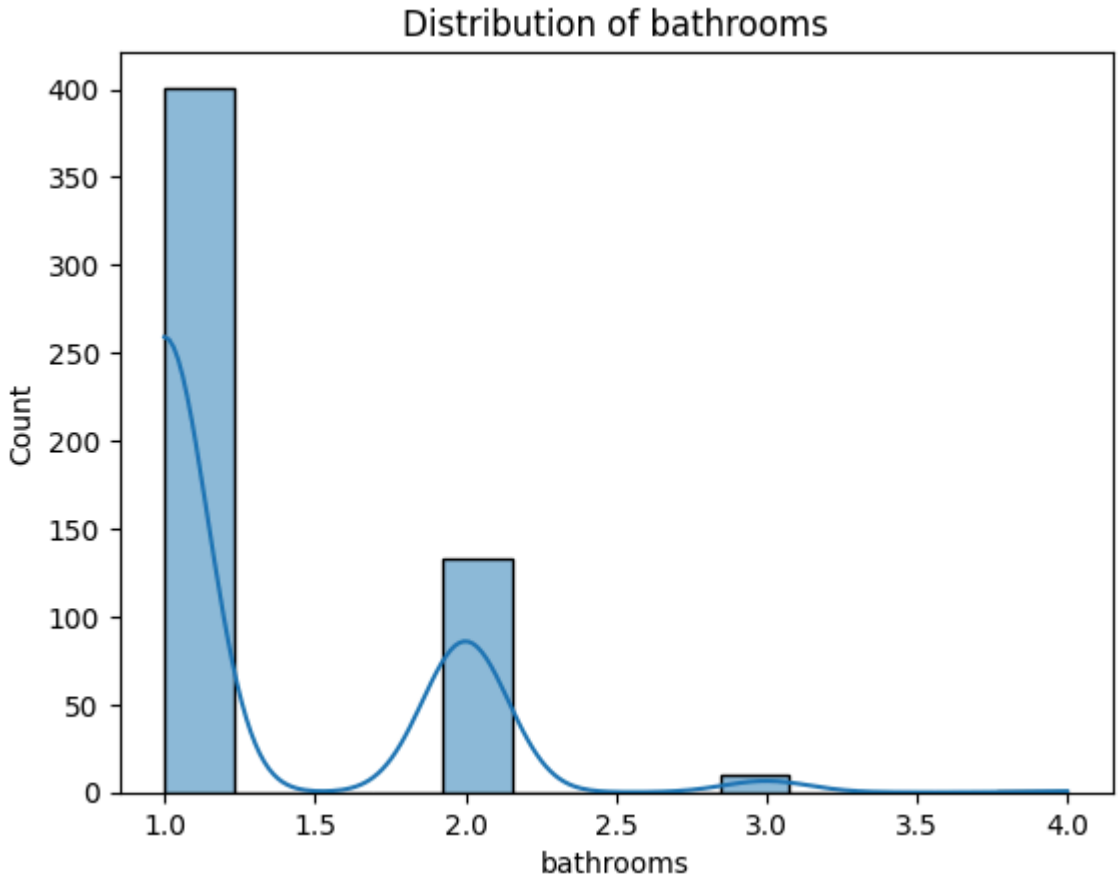
```
num_cols = ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'parking']
df[num_cols].describe()
for col in num_cols:
    sns.histplot(df[col], kde=True)
    plt.title(f"Distribution of {col}")
    plt.show()
```

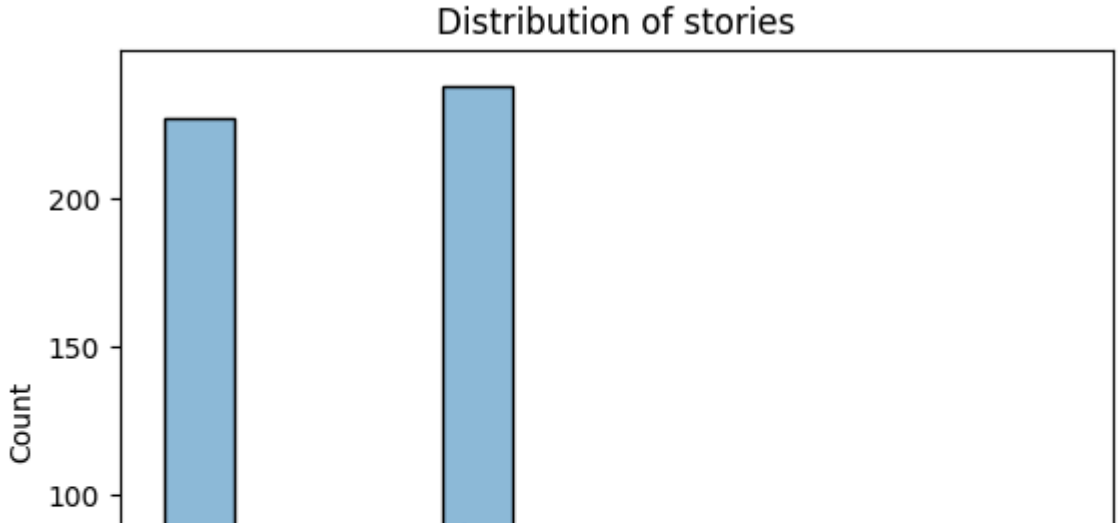

Distribution of price
Count
price
1e7
0
10
20
30
40
50
60
70
80
0.2
0.4
0.6
0.8
1.0
1.2

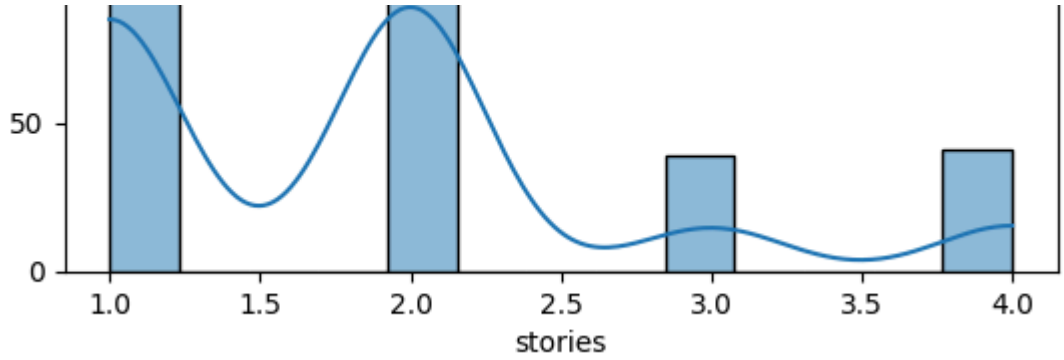

Distribution of area
Count
area
0
20
40
60
80
100
2000
4000
6000
8000
10000
12000
14000
16000

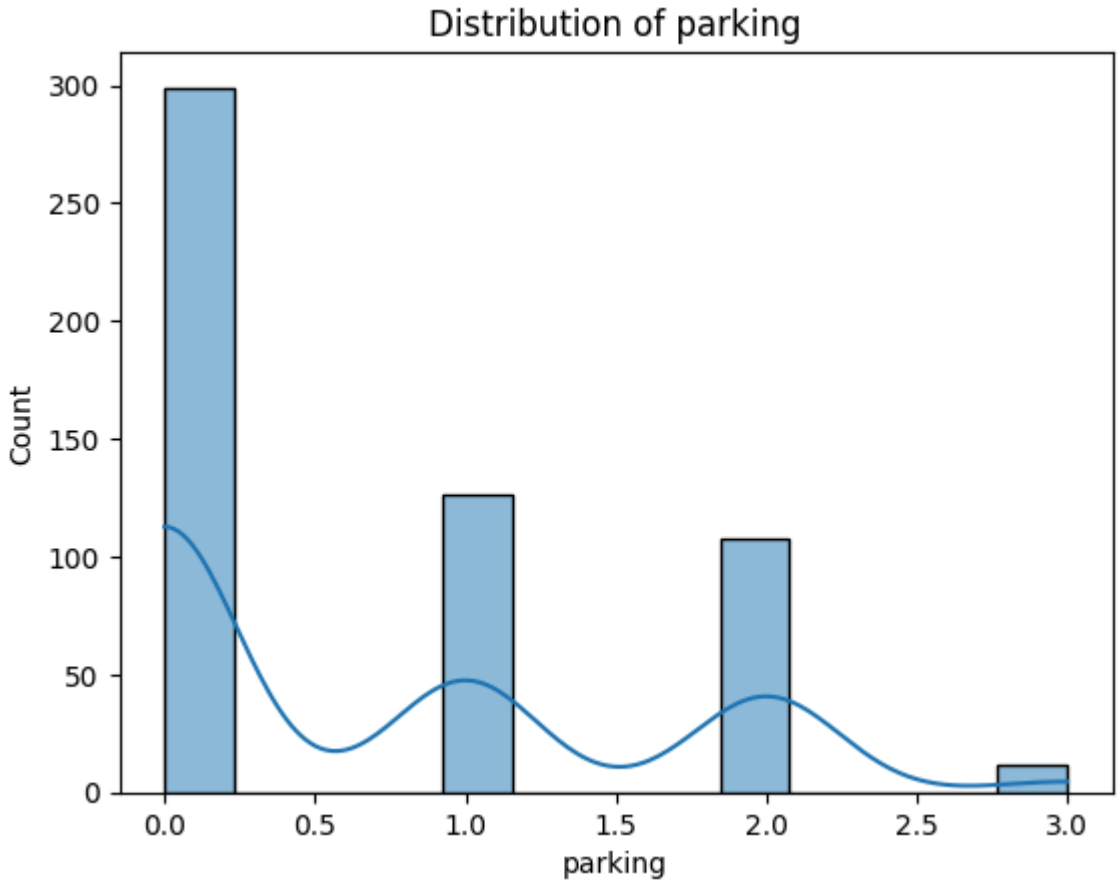

Distribution of bedrooms
250
300


Count
200
150
100
50
0
1
2
3
4
5
6
bedrooms

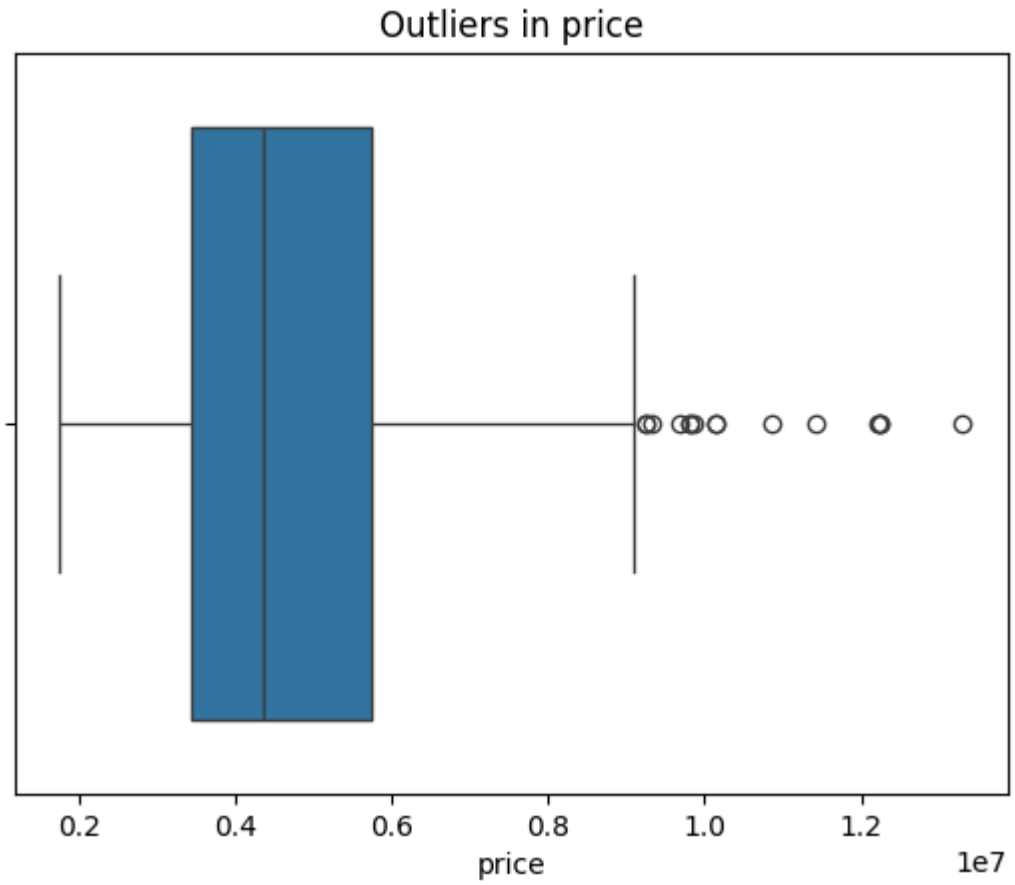

Distribution of bathrooms
Count
400
350
300
250
200
150
100
50
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
bathrooms

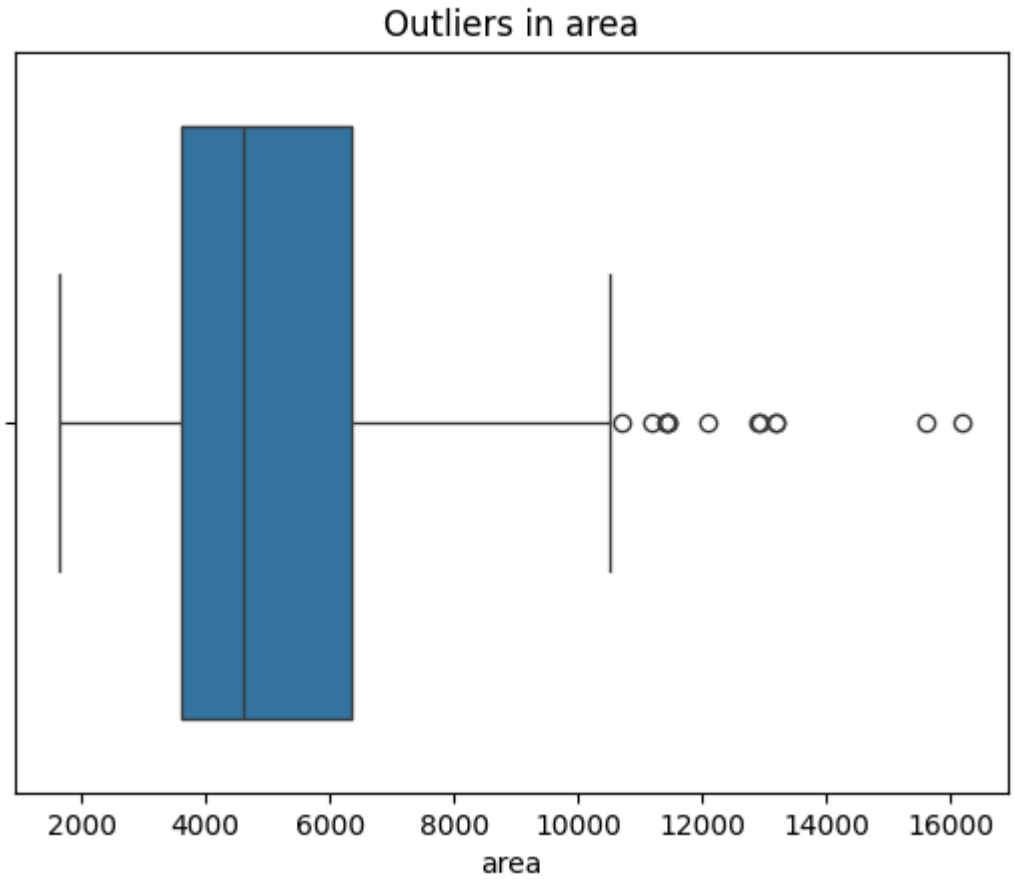

Distribution of stories
Count
200
150
100

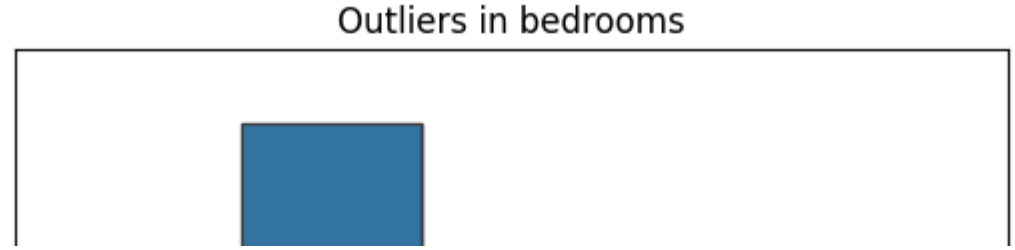

50
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
stories


Distribution of parking
300
250
200
150
100
50
0
Count
0.0
0.5
1.0
1.5
2.0
2.5
3.0
parking

```
    for col in num_cols:
        sns.boxplot(x=df[col])
        plt.title(f"Outliers in {col}")
        plt.show()
```


Outliers in price
0.2
0.4
0.6
0.8
1.0
1.2
price
1e7


Outliers in area
2000
4000
6000
8000
10000
12000
14000
16000
area


Outliers in bedrooms