```
import pandas as pd
import numpy as np


data = pd.read_csv("Housing.csv")


data.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
```

```
col_list= list(data.columns)


col_list
```

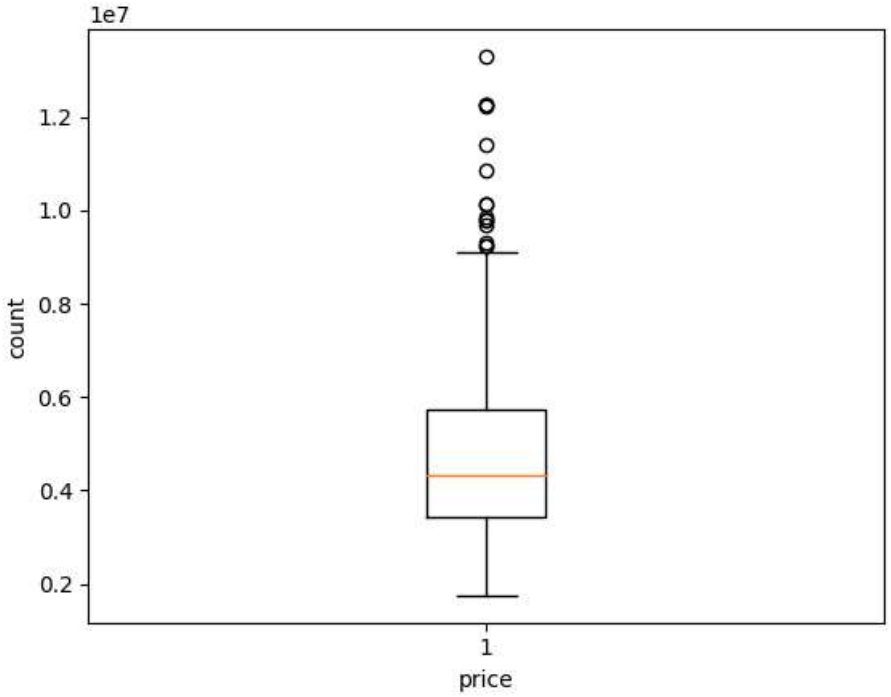
```
['price',
 'area',
 'bedrooms',
 'bathrooms',
 'stories',
 'mainroad',
 'guestroom',
 'basement',
 'hotwaterheating',
 'airconditioning',
 'parking',
 'prefarea',
 'furnishingstatus']
```

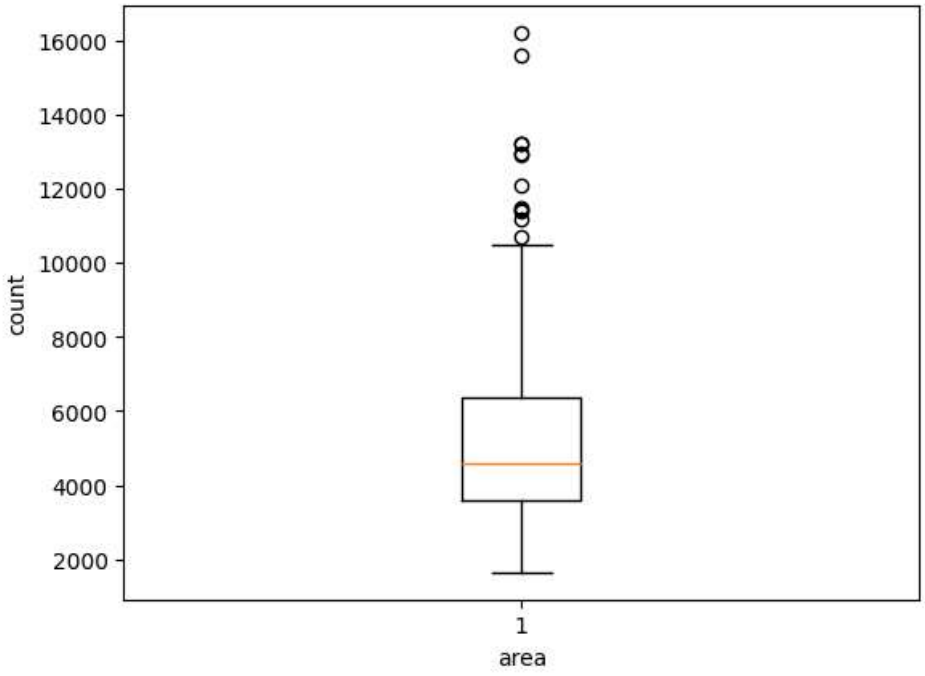
```
import matplotlib.pyplot as plt
%matplotlib inline


data.shape
```

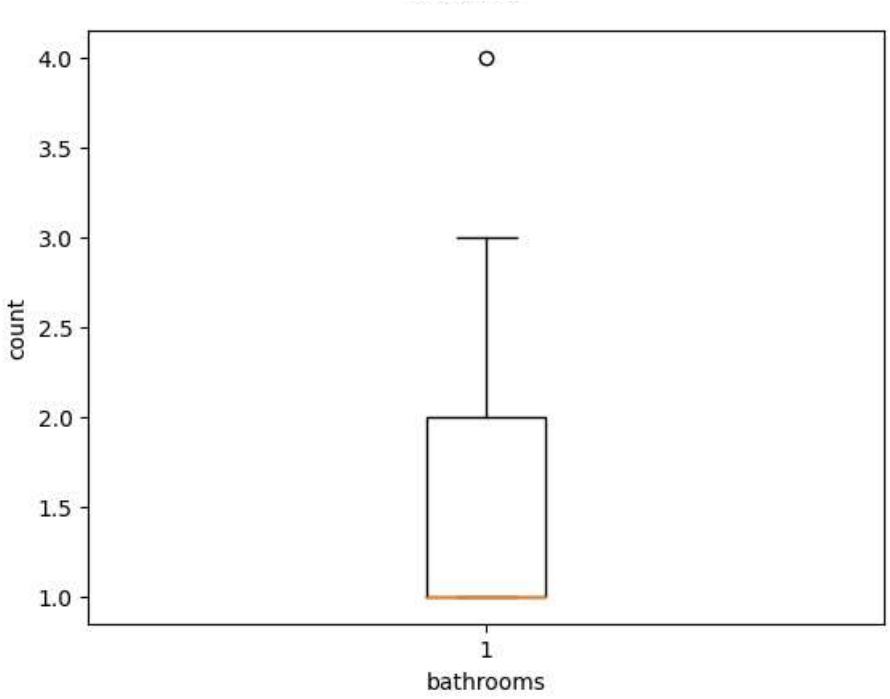
```
(545, 13)
```

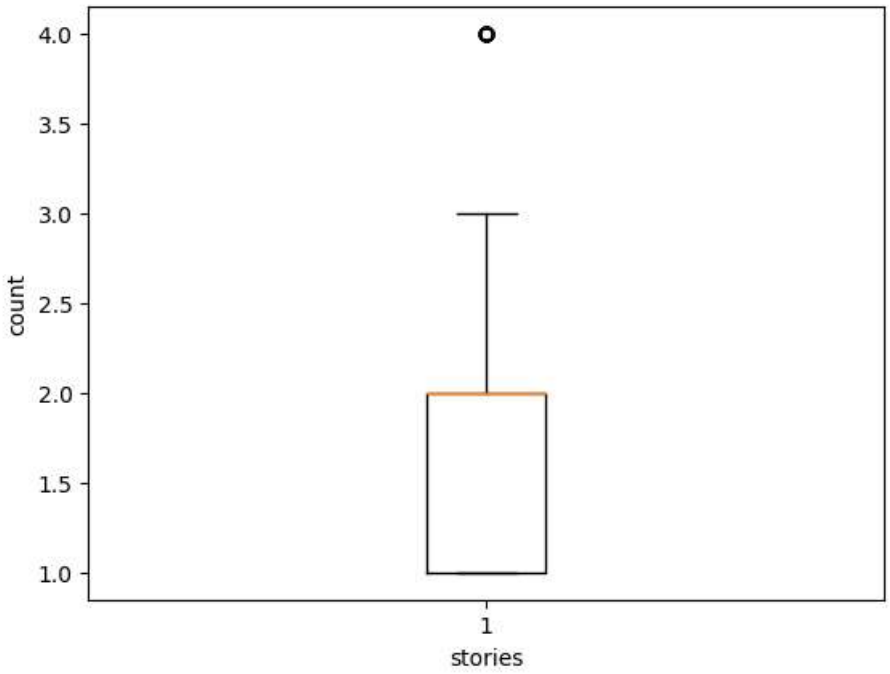
```
for col in col_list:
    if (data[col].dtype== 'int64'):
        plt.boxplot(data[col])
        plt.xlabel(col)
        plt.ylabel('count')
        plt.show()
```

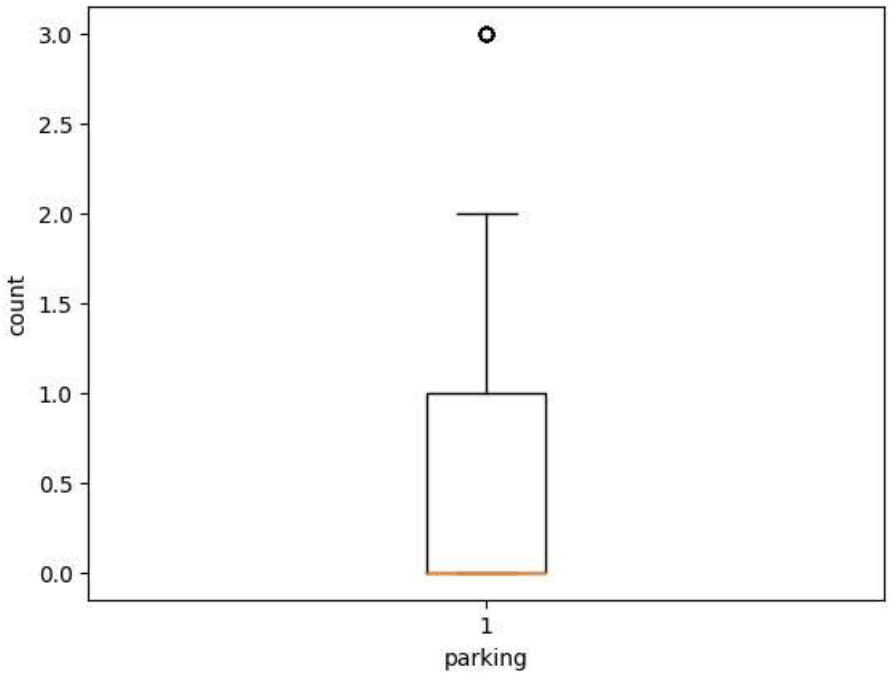

1e7
1.2
1.0
0.8
0.6
0.4
0.2
count
1
price


16000
14000
12000
10000
8000
6000
4000
2000
count
1
area


6
5
4
3
2
1
count
1
bedrooms


4.0
3.5
3.0
2.5
2.0
1.5
1.0
count
1
bathrooms


4.0
3.5
3.0
2.5
2.0
1.5
1.0
count
1
stories


3.0
2.5
2.0
1.5
1.0
0.5
0.0
count
1
parking

```
for col in col_list:
    if data[col].dtype == 'int64':
        Q1= data[col].quantile(0.25)
        Q3= data[col].quantile(0.75)
        IQR= Q3-Q1

        data= data[(data[col] >= Q1 - 1.5*IQR) & (data[col] <= Q3 + 1.5*IQR)]


data.shape
```

```
(365, 13)
```

```
for col in col_list:
    if data[col].dtype == 'int64':
        plt.boxplot(data[col])
        plt.xlabel(col)
        plt.ylabel('count')
        plt.show()
```

Show hidden output

```
for col in col_list:
    if data[col].dtype == 'int64':
        Q1= data[col].quantile(0.25)
        Q3= data[col].quantile(0.75)
        IQR= Q3-Q1

        data= data[(data[col] >= Q1 - 1.5*IQR) & (data[col] <= Q3 + 1.5*IQR)]


data.shape
```

```
(352, 13)
```

```
for col in col_list:
    if data[col].dtype == 'int64':
        plt.boxplot(data[col])
        plt.xlabel(col)
        plt.ylabel('count')
        plt.show()
```

Show hidden output

```
for col in col_list:
    if data[col].dtype == 'int64':
        Q1= data[col].quantile(0.25)
        Q3= data[col].quantile(0.75)
        IQR= Q3-Q1

        data= data[(data[col] >= Q1 - 1.5*IQR) & (data[col] <= Q3 + 1.5*IQR)]


data.shape
```

```
(349, 13)
```

```
for col in col_list:
    if data[col].dtype == 'int64':
        plt.boxplot(data[col])
        plt.xlabel(col)
        plt.ylabel('count')
        plt.show()
```

Show hidden output

I had applied Interquartile method thrice to remove outliers from the dataset. I also used boxplot to visualize the outliers in the dataset.

```
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
le= LabelEncoder()

for col in data.columns:
    if data[col].dtype == 'object':
        data[col]= le.fit_transform(data[col])
```

```
data.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa
74	6650000	4040	3	1	2	1	0	1	
78	6650000	5700	3	1	1	1	1	1	
80	6629000	6000	3	1	2	1	0	0	
84	6510000	3760	3	1	2	1	0	0	
86	6510000	6670	3	1	3	1	0	1	

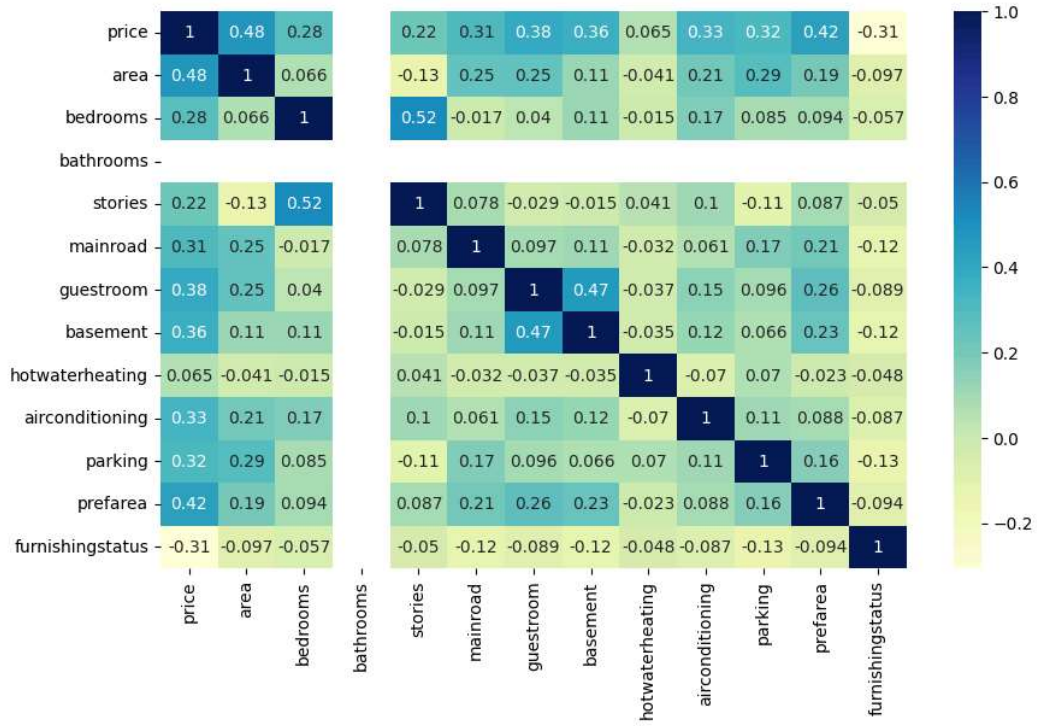
Next steps: Generate code with data | View recommended plots

```
data.corr()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroo
price	1.000000	0.477652	0.277841	NaN	0.224024	0.310214	0.38016
area	0.477652	1.000000	0.066180	NaN	-0.134290	0.251802	0.24579
bedrooms	0.277841	0.066180	1.000000	NaN	0.519217	-0.016870	0.03960
bathrooms	NaN	NaN	NaN	NaN	NaN	NaN	NaI
stories	0.224024	-0.134290	0.519217	NaN	1.000000	0.077832	-0.02937
mainroad	0.310214	0.251802	-0.016870	NaN	0.077832	1.000000	0.09716
guestroom	0.380166	0.245794	0.039608	NaN	-0.029370	0.097164	1.00000
basement	0.359980	0.114330	0.112594	NaN	-0.015250	0.106019	0.46561
hotwaterheating	0.065124	-0.041461	-0.014724	NaN	0.040611	-0.032389	-0.03724
airconditioning	0.332671	0.214785	0.170302	NaN	0.104024	0.061297	0.15013
parking	0.318497	0.291863	0.084851	NaN	-0.107325	0.171241	0.09606
prefarea	0.424797	0.193601	0.094154	NaN	0.086526	0.206838	0.26493
furnishingstatus	-0.306460	-0.097010	-0.056734	NaN	-0.050146	-0.124175	-0.08909

```
plt.figure(figsize=(10,6))
sns.heatmap(data.corr(), annot= True, cmap= 'YlGnBu')
```

<Axes: >




```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression


x= data.drop('price', axis= 1)
y= data['price']


from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()


x= scaler.fit_transform(x)


x_train, x_test, y_train, y_test= train_test_split(x, y, train_size= 0.75, random_state= 42)


l_model= LinearRegression()


l_model.fit(x_train, y_train)
```

▾ LinearRegression
LinearRegression()

```
preds= l_model.predict(x_test)


from sklearn.metrics import *


r2= r2_score(y_test, preds)
r2
```

```
0.5402589968769378
```

```
mape= mean_absolute_percentage_error(y_test, preds)
mape
```

```
0.1518900564840604
```