Housing

March 30, 2025

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
df= pd.read_csv(r"C:\Users\udayy\Downloads\Applied Stats\Housing.csv")
```

[3]:
```
df.head()
```

[3]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	\
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	

	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	no	yes	2	yes	furnished
1	no	yes	3	no	furnished
2	no	no	2	yes	semi-furnished
3	no	yes	3	yes	furnished
4	no	yes	2	no	furnished

[4]:
```
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
```

[5]:
```
df.columns
```

[5]:
```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

[6]:
```
df.head()
```

[6]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

[7]:
```
df.value_counts('furnishingstatus')
```

[7]:
```
furnishingstatus
semi-furnished    227
unfurnished       178
furnished         140
Name: count, dtype: int64
```

[8]:
```
df.head()
```

[8]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

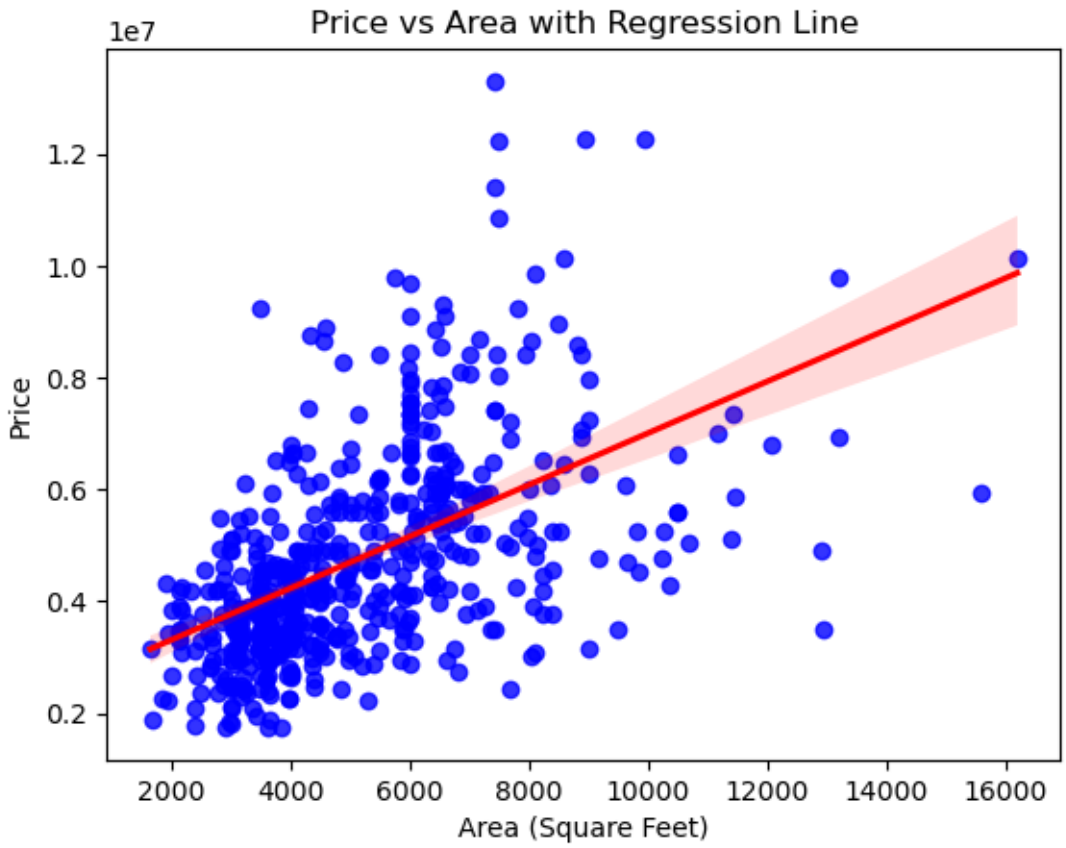
```
  hotwaterheating airconditioning  parking prefarea furnishingstatus
0              no              yes        2      yes        furnished
1              no              yes        3       no        furnished
2              no               no        2      yes   semi-furnished
3              no              yes        3      yes        furnished
4              no              yes        2       no        furnished
```

[9]:
```
df.describe()
```

[9]:
```
              price          area    bedrooms   bathrooms     stories  \
count  5.450000e+02    545.000000  545.000000  545.000000  545.000000
mean   4.766729e+06   5150.541284    2.965138    1.286239    1.805505
std    1.870440e+06   2170.141023    0.738064    0.502470    0.867492
min    1.750000e+06   1650.000000    1.000000    1.000000    1.000000
25%    3.430000e+06   3600.000000    2.000000    1.000000    1.000000
50%    4.340000e+06   4600.000000    3.000000    1.000000    2.000000
75%    5.740000e+06   6360.000000    3.000000    2.000000    2.000000
max    1.330000e+07  16200.000000    6.000000    4.000000    4.000000

          parking
count  545.000000
mean     0.693578
std      0.861586
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      3.000000
```

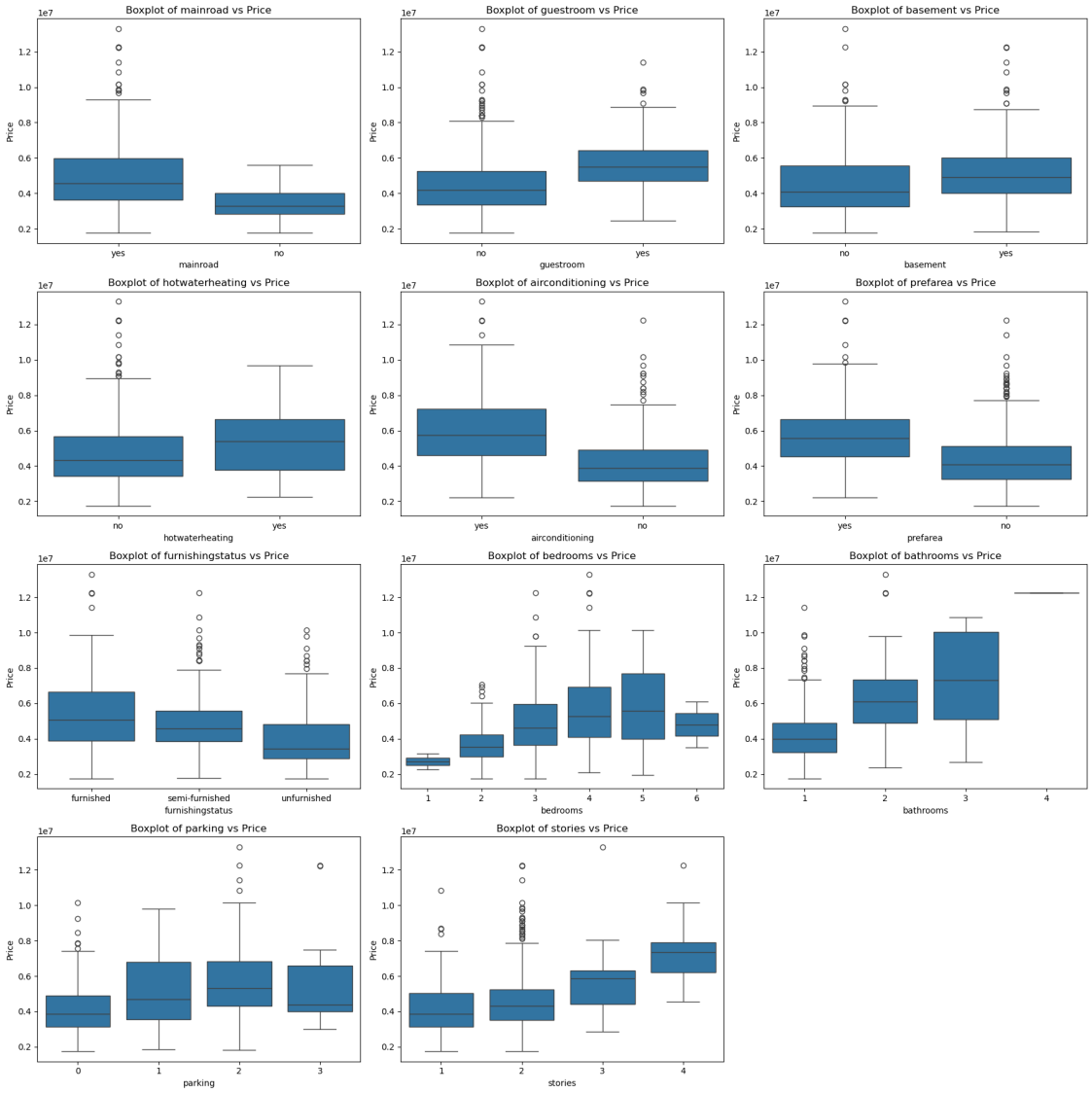
[10]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['area'], y=df['price'])
plt.xlabel("Area")
plt.ylabel("Price")
plt.title("Price vs Area")

plt.show()
```

[11]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.regplot(x='area', y='price', data=df, scatter_kws={'color': 'blue'},␣
 ↪line_kws={'color': 'red'})

# Adding labels and title
plt.xlabel('Area (Square Feet)')
plt.ylabel('Price')
plt.title('Price vs Area with Regression Line')

plt.show()
```



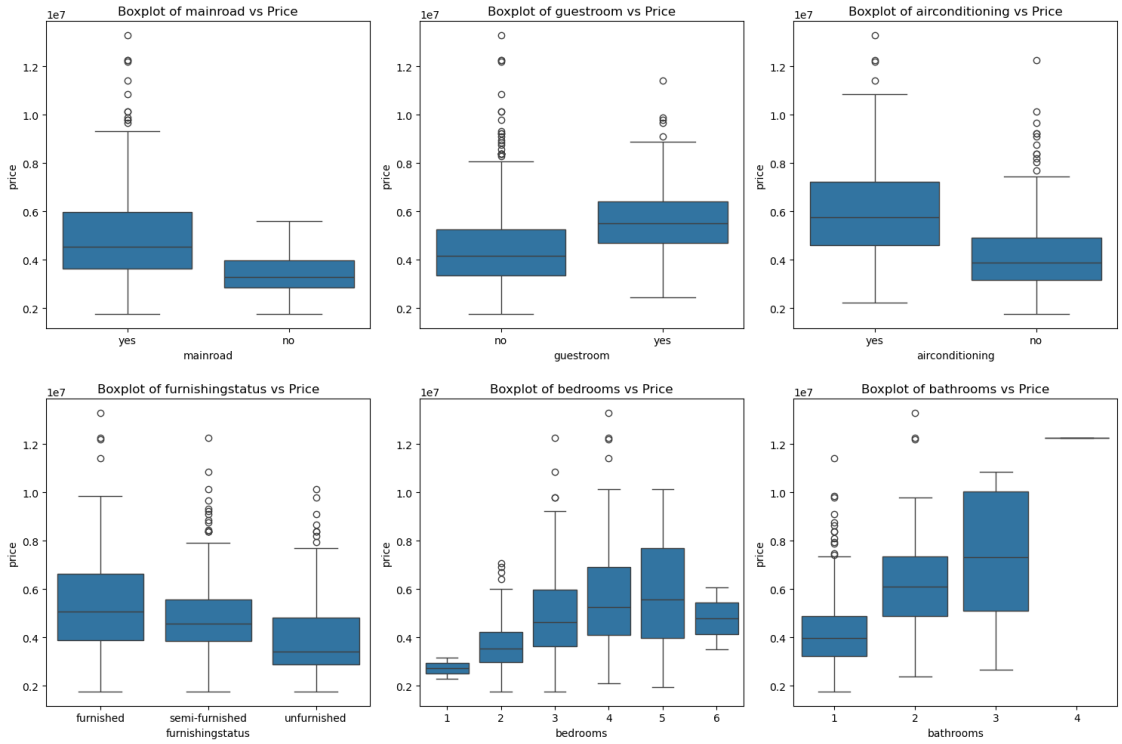
[12]:
```
import seaborn as sns
import matplotlib.pyplot as plt

categorical_vars = ['mainroad', 'guestroom', 'basement', 'hotwaterheating',
                    'airconditioning', 'prefarea', 'furnishingstatus']

numerical_vars = ['bedrooms', 'bathrooms', 'parking', 'stories']

plt.figure(figsize=(18, 18))

for i, var in enumerate(categorical_vars, 1):
    plt.subplot(4, 3, i)  # Adjust grid size to 4x3
    sns.boxplot(x=df[var], y=df['price'])
    plt.title(f'Boxplot of {var} vs Price')
    plt.xlabel(var)
    plt.ylabel('Price')

for i, var in enumerate(numerical_vars, len(categorical_vars) + 1):
    plt.subplot(4, 3, i)  # Adjust grid size to 4x3
    sns.boxplot(x=df[var], y=df['price'])
```

```
    plt.title(f'Boxplot of {var} vs Price')
    plt.xlabel(var)
    plt.ylabel('Price')

# Show all plots
plt.tight_layout()
plt.show()
```



[13]:
```
import matplotlib.pyplot as plt
import seaborn as sns

selected_vars = ['mainroad', 'guestroom', 'airconditioning',␣
 ↪'furnishingstatus', 'bedrooms', 'bathrooms']
```

```
plt.figure(figsize=(15, 10))

for i, var in enumerate(selected_vars, 1):
    plt.subplot(2, 3, i)  # 2 rows, 3 columns grid
    sns.boxplot(x=df[var], y=df['price'])
    plt.title(f'Boxplot of {var} vs Price')

plt.tight_layout()
plt.show()
```



[14]:
```
df['mainroad'] = df['mainroad'].astype('category').cat.codes
df['guestroom'] = df['guestroom'].astype('category').cat.codes

df['basement'] = df['basement'].astype('category').cat.codes
df['hotwaterheating'] = df['hotwaterheating'].astype('category').cat.codes
df['airconditioning'] = df['airconditioning'].astype('category').cat.codes
df['furnishingstatus'] = df['furnishingstatus'].astype('category').cat.codes

df['prefarea'] = df['prefarea'].astype('category').cat.codes
```

[15]:
```
df.head()
```

[15]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom
0	13300000	7420	4	2	3	1	0
1	12250000	8960	4	4	4	1	0
2	12250000	9960	3	2	2	1	0
3	12215000	7500	4	2	2	1	0
4	11410000	7420	4	1	2	1	1

	basement	hotwaterheating	airconditioning	parking	prefarea
0	0	0	1	2	1
1	0	0	1	3	0
2	1	0	0	2	1
3	1	0	1	3	1
4	1	0	1	2	0

	furnishingstatus
0	0
1	0
2	1
3	0
4	0

[16]:
```
df.describe().round(2)
```

[16]:

	price	area	bedrooms	bathrooms	stories	mainroad
count	545.00	545.00	545.00	545.00	545.00	545.00
mean	4766729.25	5150.54	2.97	1.29	1.81	0.86
std	1870439.62	2170.14	0.74	0.50	0.87	0.35
min	1750000.00	1650.00	1.00	1.00	1.00	0.00
25%	3430000.00	3600.00	2.00	1.00	1.00	1.00
50%	4340000.00	4600.00	3.00	1.00	2.00	1.00
75%	5740000.00	6360.00	3.00	2.00	2.00	1.00
max	13300000.00	16200.00	6.00	4.00	4.00	1.00

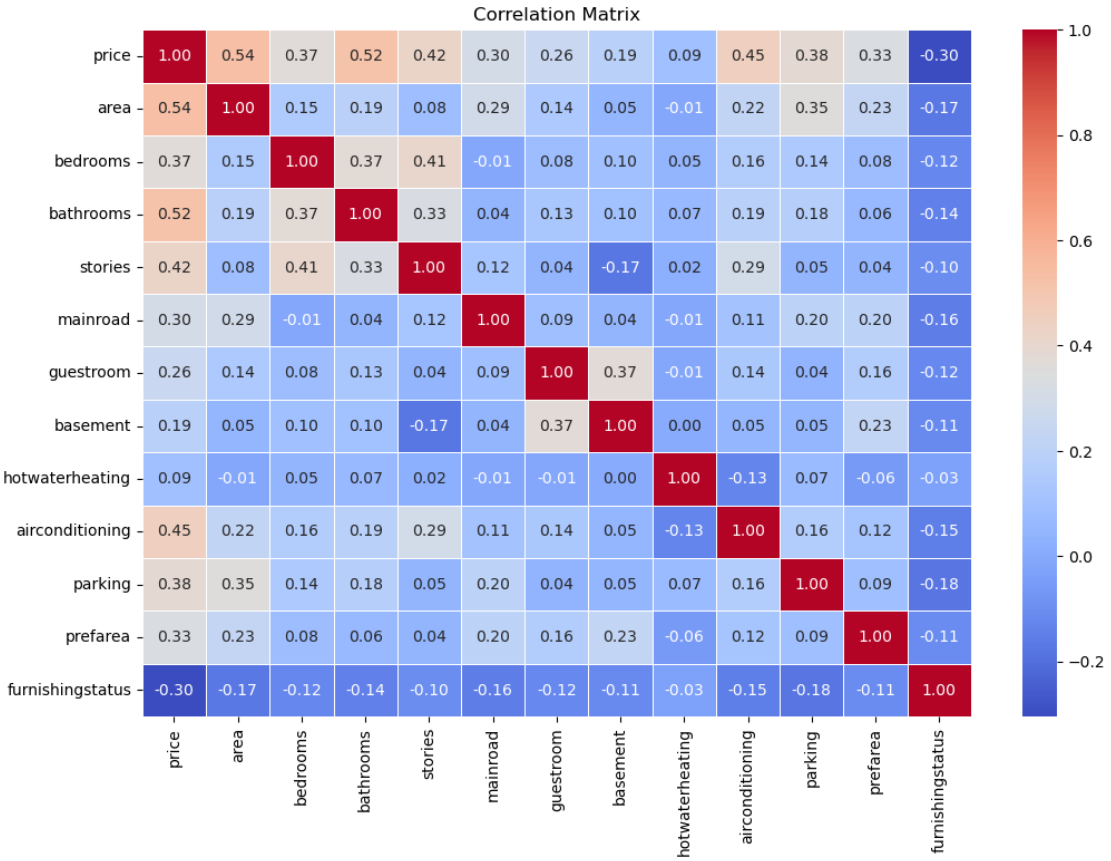
	guestroom	basement	hotwaterheating	airconditioning	parking
count	545.00	545.00	545.00	545.00	545.00
mean	0.18	0.35	0.05	0.32	0.69
std	0.38	0.48	0.21	0.47	0.86
min	0.00	0.00	0.00	0.00	0.00
25%	0.00	0.00	0.00	0.00	0.00
50%	0.00	0.00	0.00	0.00	0.00
75%	0.00	1.00	0.00	1.00	1.00
max	1.00	1.00	1.00	1.00	3.00

	prefarea	furnishingstatus
count	545.00	545.00
mean	0.23	1.07
std	0.42	0.76

```
min         0.00                0.00
25%         0.00                0.00
50%         0.00                1.00
75%         0.00                2.00
max         1.00                2.00
```

[ ]:

[17]:
```
correlation_matrix = df.corr()
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f',␣
 ↪linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```



[18]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor

X = df.drop('price', axis=1)  # Exclude the dependent variable

vif_data = pd.DataFrame()
vif_data["Variable"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.
 ↪shape[1])]

print(vif_data)
```

```
            Variable        VIF
0               area   8.270130
1           bedrooms  16.368165
2          bathrooms   9.408363
3            stories   7.880723
4           mainroad   6.852485
5          guestroom   1.472838
6           basement   2.013876
7    hotwaterheating   1.089167
8    airconditioning   1.759717
9            parking   1.985880
10          prefarea   1.492621
11  furnishingstatus   2.648467
```

[19]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor

X = df.drop(columns=[ 'price','bedrooms'])

vif_data = pd.DataFrame()
vif_data["Variable"] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.
 ↪shape[1])]

print(vif_data)
```

```
            Variable       VIF
0               area  7.690092
1          bathrooms  7.945867
2            stories  6.309101
3           mainroad  6.714722
4          guestroom  1.468507
5           basement  1.905308
6    hotwaterheating  1.088204
7    airconditioning  1.755826
8            parking  1.983821
9           prefarea  1.491638
10  furnishingstatus  2.457347
```

[20]:
```
import statsmodels.api as sm

x = df[['area', 'bathrooms','stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus']]  # Independent variables
y = df['price']  # Dependent variable
x=sm.add_constant(x)
model = sm.OLS(y, x).fit()
pridictions =model.predict(x)
# Get the regression results
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.678
Model:                            OLS   Adj. R-squared:                  0.672
Method:                 Least Squares   F-statistic:                     102.2
Date:                Sun, 30 Mar 2025   Prob (F-statistic):          1.40e-123
Time:                        23:18:30   Log-Likelihood:                -8334.4
No. Observations:                 545   AIC:                         1.669e+04
Df Residuals:                     533   BIC:                         1.674e+04
Df Model:                          11
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------------------
const              3.393e+05    2.2e+05      1.546      0.123   -9.19e+04
7.71e+05
area                247.0602     24.295     10.169      0.000     199.335
294.785
bathrooms          1.027e+06   1.01e+05     10.158      0.000    8.28e+05
1.23e+06
stories            4.875e+05   6.03e+04      8.088      0.000    3.69e+05
6.06e+05
mainroad           3.945e+05   1.42e+05      2.785      0.006    1.16e+05
6.73e+05
guestroom          2.931e+05   1.32e+05      2.218      0.027    3.35e+04
5.53e+05
basement           3.832e+05   1.09e+05      3.500      0.001    1.68e+05
5.98e+05
hotwaterheating    8.802e+05   2.24e+05      3.936      0.000    4.41e+05
1.32e+06
airconditioning    8.515e+05   1.09e+05      7.847      0.000    6.38e+05
1.06e+06
parking            2.866e+05   5.86e+04      4.895      0.000    1.72e+05
```

```
4.02e+05
prefarea          6.509e+05   1.16e+05      5.610      0.000    4.23e+05
8.79e+05
furnishingstatus -2.17e+05   6.31e+04     -3.438      0.001   -3.41e+05
-9.3e+04
==============================================================================
Omnibus:                       99.207   Durbin-Watson:                   1.207
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              265.358
Skew:                           0.901   Prob(JB):                     2.39e-58
Kurtosis:                       5.904   Cond. No.                     2.90e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.9e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[21]:
```
df.columns
```

[21]:
```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

[22]:
```
import statsmodels.api as sm

x = df[['area', 'bedrooms','bathrooms','stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus']]  # Independent variables
y = df['price']  # Dependent variable
x=sm.add_constant(x)
model = sm.OLS(y, x).fit()
pridictions =model.predict(x)
# Get the regression results
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.680
Model:                            OLS   Adj. R-squared:                  0.673
Method:                 Least Squares   F-statistic:                     94.24
Date:                Sun, 30 Mar 2025   Prob (F-statistic):          3.81e-123
Time:                        23:18:30   Log-Likelihood:                -8333.0
No. Observations:                 545   AIC:                         1.669e+04
Df Residuals:                     532   BIC:                         1.675e+04
Df Model:                          12
Covariance Type:            nonrobust
```

```
==============================================================================
====
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----
const             1.019e+05   2.62e+05      0.388      0.698   -4.14e+05
6.17e+05
area               243.9069     24.332     10.024      0.000     196.109
291.705
bedrooms          1.195e+05   7.27e+04      1.644      0.101   -2.33e+04
2.62e+05
bathrooms         9.889e+05   1.04e+05      9.551      0.000    7.85e+05
1.19e+06
stories           4.504e+05   6.43e+04      7.006      0.000    3.24e+05
5.77e+05
mainroad          4.231e+05   1.42e+05      2.970      0.003    1.43e+05
7.03e+05
guestroom          2.98e+05   1.32e+05      2.259      0.024    3.89e+04
5.57e+05
basement          3.579e+05    1.1e+05      3.243      0.001    1.41e+05
5.75e+05
hotwaterheating   8.729e+05   2.23e+05      3.909      0.000    4.34e+05
1.31e+06
airconditioning   8.536e+05   1.08e+05      7.879      0.000    6.41e+05
1.07e+06
parking           2.798e+05   5.86e+04      4.774      0.000    1.65e+05
3.95e+05
prefarea          6.471e+05   1.16e+05      5.585      0.000    4.19e+05
8.75e+05
furnishingstatus -2.132e+05   6.31e+04     -3.381      0.001   -3.37e+05
-8.93e+04
==============================================================================
Omnibus:                       94.906   Durbin-Watson:                   1.209
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              247.728
Skew:                           0.872   Prob(JB):                     1.61e-54
Kurtosis:                       5.805   Cond. No.                     3.37e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 3.37e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[ ]: