In [2]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```python
df = pd.read_csv("Housing.csv")
```

In [4]:

```python
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
```

In [5]:

```python
df.describe()
```

Out[5]:

	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

In [6]:

```python
df.shape
```

Out[6]:

```
(545, 13)
```

In [7]:

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

Out[7]:

```
price               0
area                0
bedrooms            0
bathrooms           0
stories             0
mainroad            0
guestroom           0
basement            0
hotwaterheating     0
airconditioning     0
parking             0
prefarea            0
furnishingstatus    0
dtype: int64
```

In [8]:

```python
a = df.select_dtypes(include=["float64", "int64"])
a.corr()
```

Out[8]:

	price	area	bedrooms	bathrooms	stories	parking
price	1.000000	0.535997	0.366494	0.517545	0.420712	0.384394
area	0.535997	1.000000	0.151858	0.193820	0.083996	0.352980
bedrooms	0.366494	0.151858	1.000000	0.373930	0.408564	0.139270
bathrooms	0.517545	0.193820	0.373930	1.000000	0.326165	0.177496
stories	0.420712	0.083996	0.408564	0.326165	1.000000	0.045547
parking	0.384394	0.352980	0.139270	0.177496	0.045547	1.000000

In [9]:

```
c = df.select_dtypes(include=["object"])
c
```

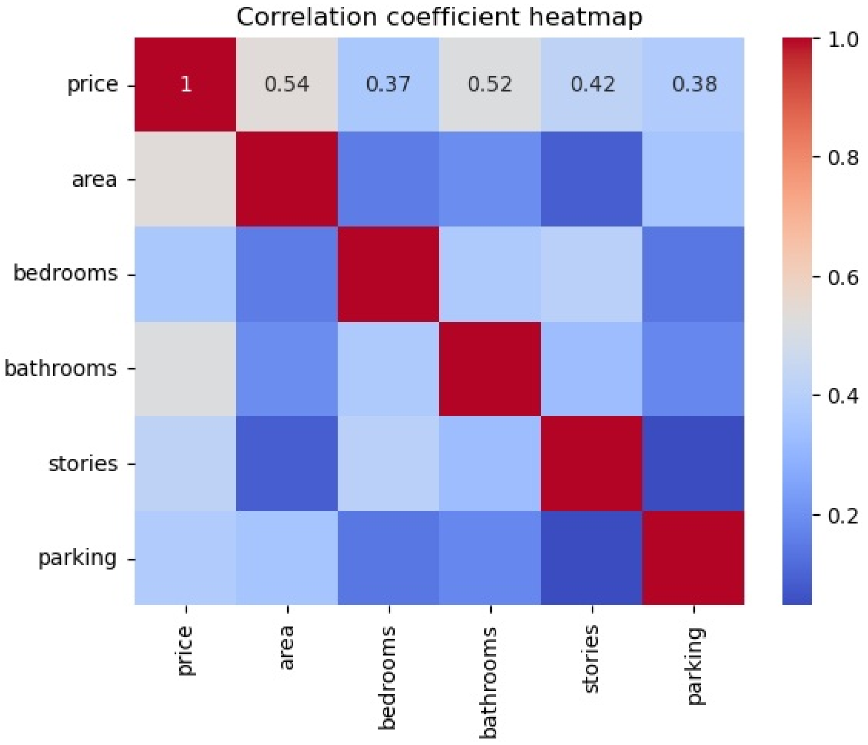
Out[9]:

	mainroad	guestroom	basement	hotwaterheating	airconditioning	prefarea	furnishingstatus
0	yes	no	no	no	yes	yes	furnished
1	yes	no	no	no	yes	no	furnished
2	yes	no	yes	no	no	yes	semi-furnished
3	yes	no	yes	no	yes	yes	furnished
4	yes	yes	yes	no	yes	no	furnished
...	...	...	...	...	...	...	...
540	yes	no	yes	no	no	no	unfurnished
541	no	no	no	no	no	no	semi-furnished
542	yes	no	no	no	no	no	unfurnished
543	no	no	no	no	no	no	furnished
544	yes	no	no	no	no	no	unfurnished

545 rows × 7 columns

In [10]:

```
sns.heatmap(a.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation coefficient heatmap")
plt.show()
```



Linear regression models

In [11]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from scipy import stats
lr = LinearRegression()
price = df[["price"]]
for i in range(5):
    b = df[a.columns[i + 1]]
    lr.fit(price, b)
    predict1 = lr.predict(price)
    r2 = r2_score(b, predict1)
    corr, p_value = stats.pearsonr(df["price"], b)
    print(f"Column: {a.columns[i + 1]}")
    print(f"Correlation Coefficient: {corr:.2f}")
    print(f"P-value: {p_value:.5f}")
    print(f"R-squared Score: {r2:.2f}")
    print("-------------------------")
```

```
Column: area
Correlation Coefficient: 0.54
P-value: 0.00000
R-squared Score: 0.29
-------------------------
Column: bedrooms
Correlation Coefficient: 0.37
P-value: 0.00000
R-squared Score: 0.13
-------------------------
Column: bathrooms
Correlation Coefficient: 0.52
P-value: 0.00000
R-squared Score: 0.27
-------------------------
Column: stories
Correlation Coefficient: 0.42
P-value: 0.00000
R-squared Score: 0.18
-------------------------
Column: parking
Correlation Coefficient: 0.38
P-value: 0.00000
R-squared Score: 0.15
-------------------------
```

In [12]:

```
a = pd.get_dummies(df, columns=["mainroad", "guestroom", "basement", "hotwaterheating", "airconditioning", "pre
b = ["mainroad_yes", "guestroom_yes", "basement_yes", "hotwaterheating_yes", "airconditioning_yes", "prefarea_ye
a
```

Out[12]:

	price	area	bedrooms	bathrooms	stories	parking	mainroad_no	mainroad_yes	guestroom_no	guestroom_yes	...	bas
0	13300000	7420	4	2	3	2	False	True	True	False	...	
1	12250000	8960	4	4	4	3	False	True	True	False	...	
2	12250000	9960	3	2	2	2	False	True	True	False	...	
3	12215000	7500	4	2	2	3	False	True	True	False	...	
4	11410000	7420	4	1	2	2	False	True	False	True	...	
...	...	...	...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	2	False	True	True	False	...	
541	1767150	2400	3	1	1	0	True	False	True	False	...	
542	1750000	3620	2	1	1	0	False	True	True	False	...	
543	1750000	2910	3	1	1	0	True	False	True	False	...	
544	1750000	3850	3	1	2	0	False	True	True	False	...	

545 rows × 21 columns

In [18]:

```
for i in b:
    lr.fit(price, a[i])
    predict1 = lr.predict(price)
    corr = df['price'].corr(a[i])
    r2 = r2_score(a[i], predict1)
    print(f"Column: {i}")
    print(f"Correlation Coefficient: {corr}")
    print(f"R-squared Score: {r2:.2f}")
    print("-------------------------")
```

```
Column: mainroad_yes
Correlation Coefficient: 0.2968984892639765
R-squared Score: 0.09
-------------------------
Column: guestroom_yes
Correlation Coefficient: 0.25551728993499967
R-squared Score: 0.07
-------------------------
Column: basement_yes
Correlation Coefficient: 0.18705659793805252
R-squared Score: 0.03
-------------------------
Column: hotwaterheating_yes
Correlation Coefficient: 0.09307284392139682
R-squared Score: 0.01
-------------------------
Column: airconditioning_yes
Correlation Coefficient: 0.4529540842560473
R-squared Score: 0.21
-------------------------
Column: prefarea_yes
Correlation Coefficient: 0.32977704986810746
R-squared Score: 0.11
-------------------------
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js