Assignment – 3 ADS

Name – Manjunaatt G K

Reg no – 20bci7298

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

In [3]:
```
df = pd.read_csv("Housing.csv")
df
```

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

545 rows × 12 columns

In [4]:
```
df.shape
```

(545, 12)

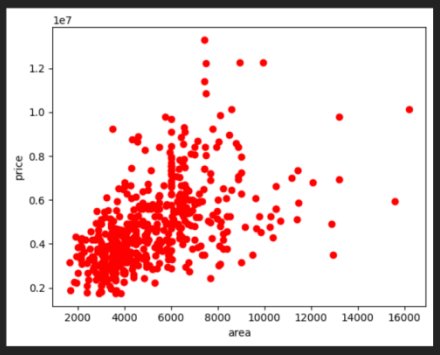
In [5]:
```
df.isna().sum()
```

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

Univariate Analysis

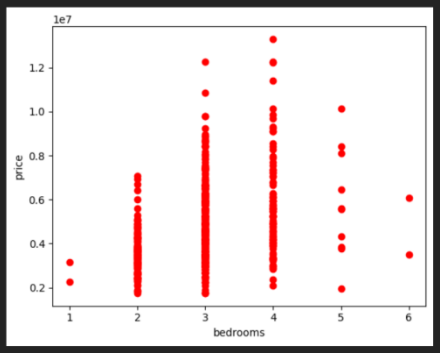
In [6]:
```
plt.scatter(df.area, df.price, color='red')
plt.xlabel('area')
plt.ylabel('price')
plt.show()
```

```
In [6]:
plt.scatter(df.area, df.price, color='red')
plt.xlabel('area')
plt.ylabel('price')
plt.show()
```

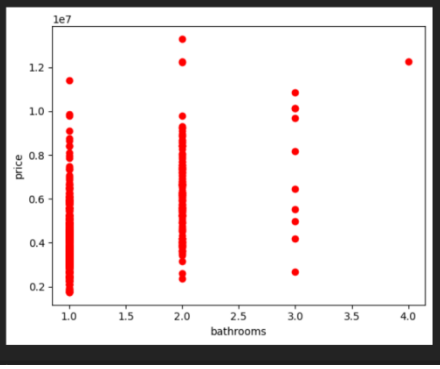



```
In [7]:
plt.scatter(df.bedrooms, df.price, color='red')
plt.xlabel('bedrooms')
plt.ylabel('price')
```

```
plt.scatter(df.bedrooms, df.price, color='red')
plt.xlabel('bedrooms')
plt.ylabel('price')
plt.show()
```

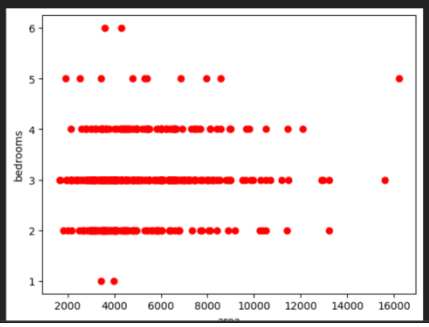



```
plt.xlabel('bathrooms')
plt.ylabel('price')
plt.show()
```

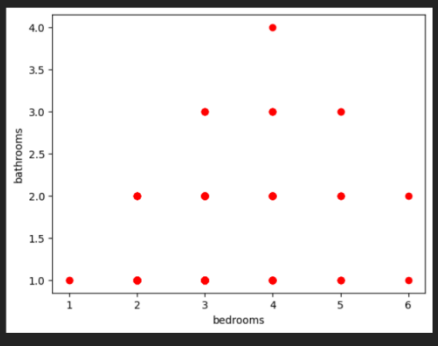


Bi - variate Analysis

In [9]:
```
plt.scatter(df.area, df.bedrooms, color='red')
plt.xlabel('area')
plt.ylabel('bedrooms')
plt.show()
```



In [10]:
```
plt.scatter(df.bedrooms, df.bathrooms, color='red')
plt.xlabel('bedrooms')
plt.ylabel('bathrooms')
plt.show()
```



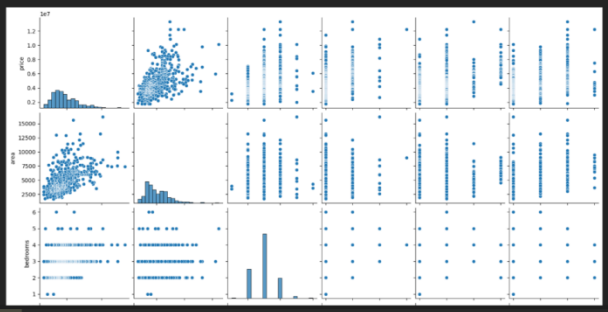
Multi - variate analysis

In [11]:
```
df.corr()
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\1134722465.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence th is warning.
  df.corr()
```

	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 [12]:
```
sns.pairplot(df)
plt.show()
```



In [13]:
```
df.mean()
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\3698961737.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.mean()
```

```
price       4.766729e+06
area        5.150541e+03
bedrooms    2.965138e+00
```

In [14]:
```
df.median()
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.media
n is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only val
id columns or specify the value of numeric_only to silence this warning.
  df.median()


price        4340000.0
area            4600.0
bedrooms           3.0
bathrooms          1.0
stories            2.0
parking            0.0
dtype: float64
```

In [15]:
```
df.mode()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking
0	3500000	6000.0	3.0	1.0	2.0	yes	no	no	no	no	0.0
1	4200000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

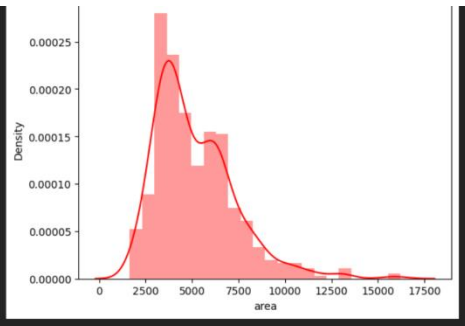
In [16]:
```
sns.distplot(df.area,color='red')

```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\1825346803.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
```



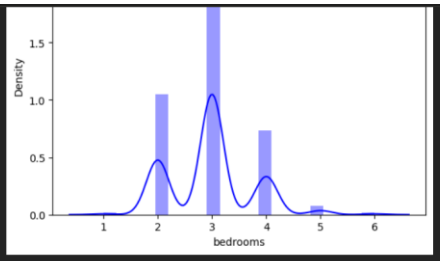
In [17]:
```
sns.distplot(df.bedrooms,color='blue')
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\4088092191.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



In [18]:
```
sns.distplot(df.bathrooms, color='orange')
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\403326755.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.bathrooms, color='orange')


<AxesSubplot: xlabel='bathrooms', ylabel='Density'>
```

In [20]:
```
df.max, df.min
```

```
(<bound method NDFrame._add_numeric_operations.<locals>.max of        price  area  bedrooms  bathrooms  stories mainroad guestroom baseme
nt  \
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

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

[545 rows x 12 columns]>,
<bound method NDFrame._add_numeric_operations.<locals>.min of        price  area  bedrooms  bathrooms  stories mainroad guestroom baseme
nt  \
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
```


```
    hotwaterheating airconditioning  parking furnishingstatus
0                no             yes        2        furnished
1                no             yes        3        furnished
2                no              no        2   semi-furnished
3                no             yes        3        furnished
4                no             yes        2        furnished
..              ...             ...      ...              ...
540              no              no        2      unfurnished
541              no              no        0   semi-furnished
542              no              no        0      unfurnished
543              no              no        0        furnished
544              no              no        0      unfurnished

[545 rows x 12 columns]>,
<bound method NDFrame._add_numeric_operations.<locals>.min of        price  area  bedrooms  bathrooms  stories mainroad guestroom baseme
nt  \
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

    hotwaterheating airconditioning  parking furnishingstatus
0                no             yes        2        furnished
1                no             yes        3        furnished
2                no              no        2   semi-furnished
3                no             yes        3        furnished
4                no             yes        2        furnished
..              ...             ...      ...              ...
540              no              no        2      unfurnished
541              no              no        0   semi-furnished
542              no              no        0      unfurnished
543              no              no        0        furnished
544              no              no        0      unfurnished
```


```
df.skew()
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\1665899112.py:1: FutureWarning: The default value of numeric_only in DataFrame.skew
is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid
columns or specify the value of numeric_only to silence this warning.
  df.skew()


price        1.212239
area         1.321188
bedrooms     0.495684
bathrooms    1.589264
stories      1.082088
parking      0.842062
dtype: float64
```

In [22]:
```
df.kurt()
```

```
C:\Users\manjunaatt\AppData\Local\Temp\ipykernel_19792\1257127604.py:1: FutureWarning: The default value of numeric_only in DataFrame.kurt
is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid
columns or specify the value of numeric_only to silence this warning.
  df.kurt()


price        1.960130
area         2.751480
bedrooms     0.728323
bathrooms    2.164856
stories      0.679404
parking     -0.573063
dtype: float64
```

In [23]:
```
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

In [24]:
```
df.head()
```

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

In [25]:
```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

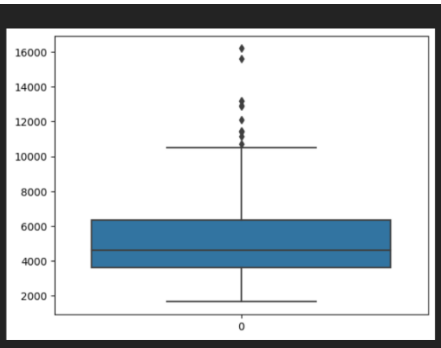
In [26]:
```
cat_cols = ['mainroad','guestroom','basement','hotwaterheating','airconditioning','furnishingstatus']
```

In [27]:
```
for cols in cat_cols:
    le.fit(df[cols])
    df[cols] = le.transform(df[cols])
```

In [28]:
```
df
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking
0	13300000	7420	4	2	3	1	0	0	0	1	2
1	12250000	8960	4	4	4	1	0	0	0	1	3
2	12250000	9960	3	2	2	1	0	1	0	0	2
3	12215000	7500	4	2	2	1	0	1	0	1	3
4	11410000	7420	4	1	2	1	1	1	0	1	2
...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	1	0	1	0	0	2
541	1767150	2400	3	1	1	0	0	0	0	0	0
542	1750000	3620	2	1	1	1	0	0	0	0	0
543	1750000	2910	3	1	1	0	0	0	0	0	0
544	1750000	3850	3	1	2	1	0	0	0	0	0

545 rows × 12 columns



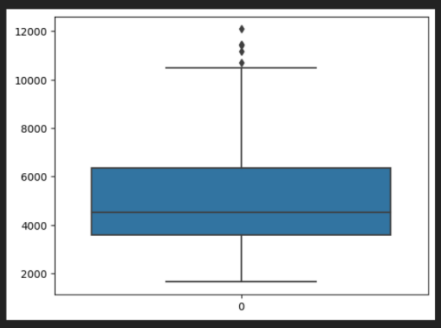
In [31]:
```
perc1 = df.area.quantile(0.99)
perc1
```

12543.599999999955

In [32]:
```
df = df[df.area<=perc1]
sns.boxplot(df.area)
```

<AxesSubplot: >

<AxesSubplot: >

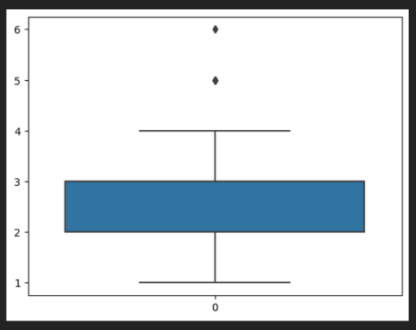


In [33]:
```
sns.boxplot(df.bedrooms)
```

<AxesSubplot: >

In [33]:
```
sns.boxplot(df.bedrooms)
```

<AxesSubplot: >

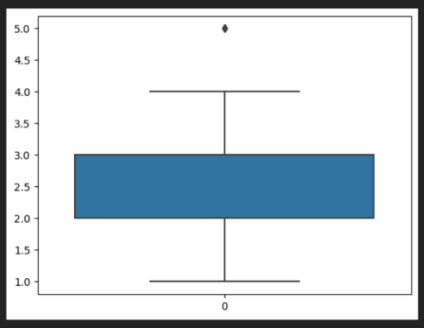


In [34]:
```
perc2 = df.bedrooms.quantile(0.99)
perc2
```

5.0

```
df = df[df.bedrooms<=perc2]
sns.boxplot(df.bedrooms)
```

<AxesSubplot: >



In [36]:
```
X = df.drop('price',axis=1)
y = df.price
```

In [37]:
```
X
```

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	furnishi
0	7420	4	2	3	1	0	0	0	1	2	0
1	8960	4	4	4	1	0	0	0	1	3	0
2	9960	3	2	2	1	0	1	0	0	2	1
3	7500	4	2	2	1	0	1	0	1	3	0
4	7420	4	1	2	1	1	1	0	1	2	0
...	...	...	...	...	...	...	...	...	...	...	...
540	3000	2	1	1	1	0	1	0	0	2	2
541	2400	3	1	1	0	0	0	0	0	0	1
542	3620	2	1	1	1	0	0	0	0	0	2
543	2910	3	1	1	0	0	0	0	0	0	0
544	3850	3	1	2	1	0	0	0	0	0	2

537 rows × 11 columns

In [38]:
```
y
```

```
0      13300000
1      12250000
2      12250000
3      12215000
4      11410000
         ...
540     1820000
541     1767150
542     1750000
543     1750000
544     1750000
Name: price, Length: 537, dtype: int64
```

In [39]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [40]:
```
scaler = MinMaxScaler()
```

In [41]:
```
scaler.fit(X)
```

MinMaxScaler

In [42]:
```
X = scaler.transform(X)
```

In [43]:
```
y = np.array(y).reshape(-1,1)
```

In [44]:
```
scaler.fit(y)
```

MinMaxScaler

In [45]:
```
y = scaler.transform(y)
```

In [46]:
```
y.shape
```

```
(537, 1)
```

In [45]:
```
y = scaler.transform(y)
```

In [46]:
```
y.shape
```

```
(537, 1)
```

In [47]:
```
X.shape
```

```
(537, 11)
```

In [48]:
```
from sklearn.model_selection import train_test_split
```

In [49]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2)
```

In [50]:
```
X_train
```

```
array([[0.23266284, 0.5       , 0.        , ..., 0.        , 0.        ,
        0.5       ],
       [0.02442529, 1.        , 0.        , ..., 0.        , 0.        ,
        0.5       ],
       [0.27298851, 0.25      , 0.        , ..., 0.        , 0.        ,
        0.5       ],
       ...,
       [0.17720307, 0.75      , 0.33333333, ..., 0.        , 0.66666667,
        0.        ],
       [0.83429119, 0.25      , 0.        , ..., 0.        , 0.33333333,
        0.5       ],
       [0.47413793, 0.5       , 0.        , ..., 1.        , 1.        ,
```

```
In [51]: y_train

array([[0.21212121],
       [0.22424242],
       [0.13030303],
       [0.30242424],
       [0.23030303],
       [0.12727273],
       [0.63636364],
       [0.36969697],
       [0.12727273],
       [0.42424242],
       [0.12121212],
       [0.22969697],
       [0.16363636],
       [0.33333333],
       [0.17515152],
       [0.21212121],
       [0.15757576],
       [0.17575758],
       [0.0569697 ],
       [0.0969697 ],
       [0.7030303 ],
       [0.16969697],
       [0.51515152],

In [52]: from sklearn.linear_model import LinearRegression

In [53]: model = LinearRegression()

In [54]: model.fit(X_train, y_train)
```

```
In [52]: from sklearn.linear_model import LinearRegression

In [53]: model = LinearRegression()

In [54]: model.fit(X_train, y_train)

         LinearRegression

In [55]: model.intercept_

array([0.00423543])

In [56]: model.coef_

array([[ 0.26370325,  0.05847392,  0.24485328,  0.12323452,  0.04649091,
         0.02700413,  0.04478288,  0.00873801,  0.07569185,  0.07841004,
        -0.03454041]])

In [57]: model.score(X_test, y_test)

0.531101037030258 9
```

```
In [56]: model.coef_

array([[ 0.26370325,  0.05847392,  0.24485328,  0.12323452,  0.04649091,
         0.02700413,  0.04478288,  0.00873801,  0.07569185,  0.07841004,
        -0.03454041]])

In [57]: model.score(X_test, y_test)

0.531101037030258 9

In [58]: y_pred = model.predict(X_test)

In [59]: from sklearn.metrics import mean_squared_error
         from sklearn.metrics import mean_absolute_error

In [60]: mean_squared_error(y_test, y_pred)

0.007446552626579722

In [61]: mean_absolute_error(y_test, y_pred)

0.06723314050366874
```