# Experiment 2 : EDA on *Housing.csv*

In [34]:
```
import pandas as pd
df = pd.read_csv("Housing.csv")
df.head()
```

Out[34]:

| | id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot |
|---|---|---|---|---|---|---|---|
| **0** | 7229300521 | 20141013T000000 | 231300.0 | 2 | 1.00 | 1180 | 5650 |
| **1** | 6414100192 | 20141209T000000 | 538000.0 | 3 | 2.25 | 2570 | 7242 |
| **2** | 5631500400 | 20150225T000000 | 180000.0 | 2 | 1.00 | 770 | 10000 |
| **3** | 2487200875 | 20141209T000000 | 604000.0 | 4 | 3.00 | 1960 | 5000 |
| **4** | 1954400510 | 20150218T000000 | 510000.0 | 3 | 2.00 | 1680 | 8080 |

5 rows × 21 columns

In [35]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             21613 non-null  int64  
 1   date           21613 non-null  object 
 2   price          21613 non-null  float64
 3   bedrooms       21613 non-null  int64  
 4   bathrooms      21613 non-null  float64
 5   sqft_living    21613 non-null  int64  
 6   sqft_lot       21613 non-null  int64  
 7   floors         21613 non-null  float64
 8   waterfront     21613 non-null  int64  
 9   view           21613 non-null  int64  
 10  condition      21613 non-null  int64  
 11  grade          21613 non-null  int64  
 12  sqft_above     21613 non-null  int64  
 13  sqft_basement  21613 non-null  int64  
 14  yr_built       21613 non-null  int64  
 15  yr_renovated   21613 non-null  int64  
 16  zipcode        21613 non-null  int64  
 17  lat            21613 non-null  float64
 18  long           21613 non-null  float64
 19  sqft_living15  21613 non-null  int64  
 20  sqft_lot15     21613 non-null  int64  
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

In [36]:
```
df.describe()
```

Out[36]:

| | id | price | bedrooms | bathrooms | sqft_living | sqf |
|---|---|---|---|---|---|---|
| **count** | 2.161300e+04 | 2.161300e+04 | 21613.000000 | 21613.000000 | 21613.000000 | 2.161300 |
| **mean** | 4.580306e+09 | 5.400886e+05 | 3.370795 | 2.114757 | 2079.899736 | 1.510697 |
| **std** | 2.876570e+09 | 3.671268e+05 | 0.930105 | 0.770163 | 918.440897 | 4.142051 |
| **min** | 1.000102e+06 | 7.500000e+04 | 0.000000 | 0.000000 | 290.000000 | 5.200000 |
| **25%** | 2.123049e+09 | 3.219500e+05 | 3.000000 | 1.750000 | 1427.000000 | 5.040000 |
| **50%** | 3.904930e+09 | 4.500000e+05 | 3.000000 | 2.250000 | 1910.000000 | 7.618000 |
| **75%** | 7.308900e+09 | 6.450000e+05 | 4.000000 | 2.500000 | 2550.000000 | 1.068800 |
| **max** | 9.900000e+09 | 7.700000e+06 | 33.000000 | 8.000000 | 13540.000000 | 1.651359 |

In [37]:
```
df.shape
```

Out[37]:
```
(21613, 21)
```

In [38]:
```
df.columns
```

Out[38]:
```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

In [39]:
```
df = df.drop('date',axis=1)
```

In [40]:
```
df = df.drop('id',axis=1)
```

In [42]:
```
df['bedrooms'].unique()
```

Out[42]:
```
array([ 2,  3,  4,  5,  1,  6,  7,  0,  8,  9, 11, 10, 33])
```

In [8]:
```
df['view'].unique()
```

Out[8]:
```
array([0, 3, 4, 2, 1])
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
id               0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

In [10]:
```
df[df.duplicated()]
#no duplicate data
```

Out[10]:

| | id | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | wate |
|---|---|---|---|---|---|---|---|---|
| **3951** | 1825069031 | 550000.0 | 4 | 1.75 | 2410 | 8447 | 2.0 | |
| **14983** | 6308000010 | 585000.0 | 3 | 2.50 | 2290 | 5089 | 2.0 | |
| **20054** | 8648900110 | 555000.0 | 3 | 2.50 | 1940 | 3211 | 2.0 | |

In [11]:
```
df.drop_duplicates(inplace=True)
```

In [12]:
```
df.shape
```

Out[12]:
```
(21610, 20)
```

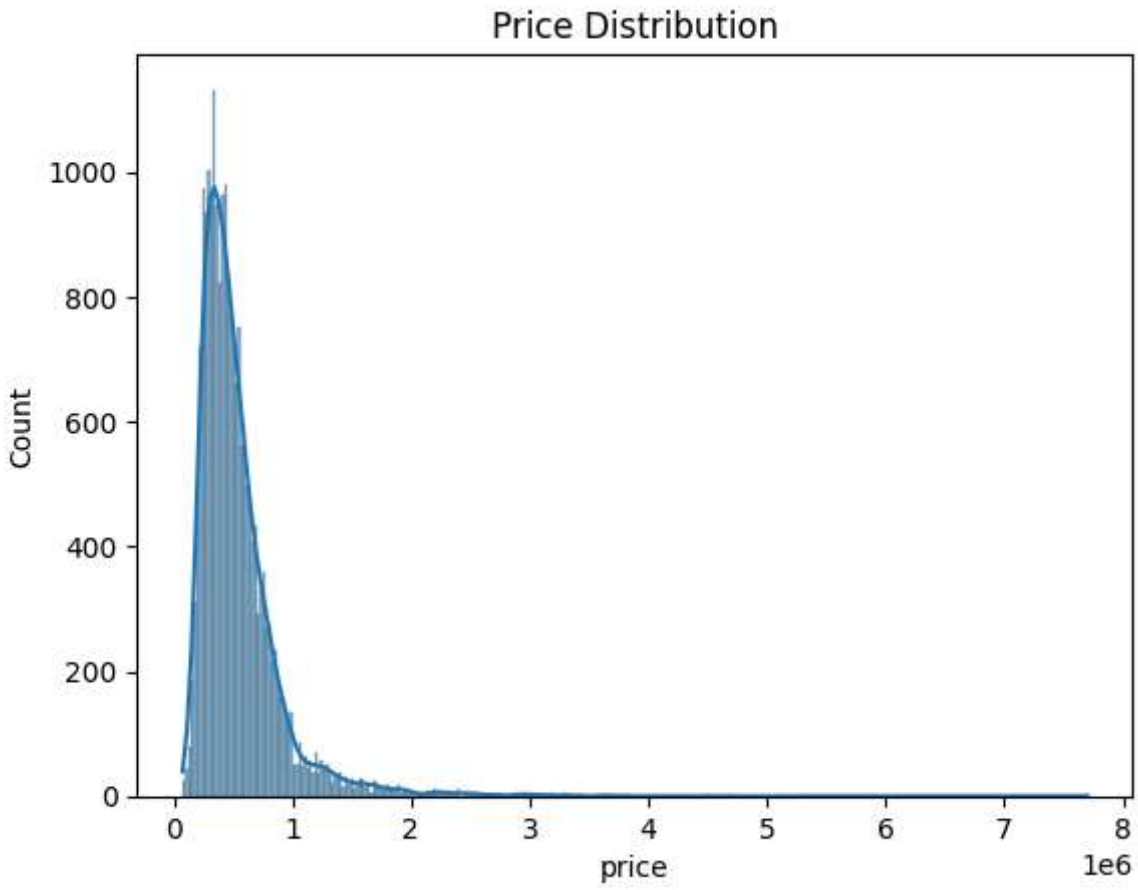
In [13]:
```
df.corr()
```

Out[13]:

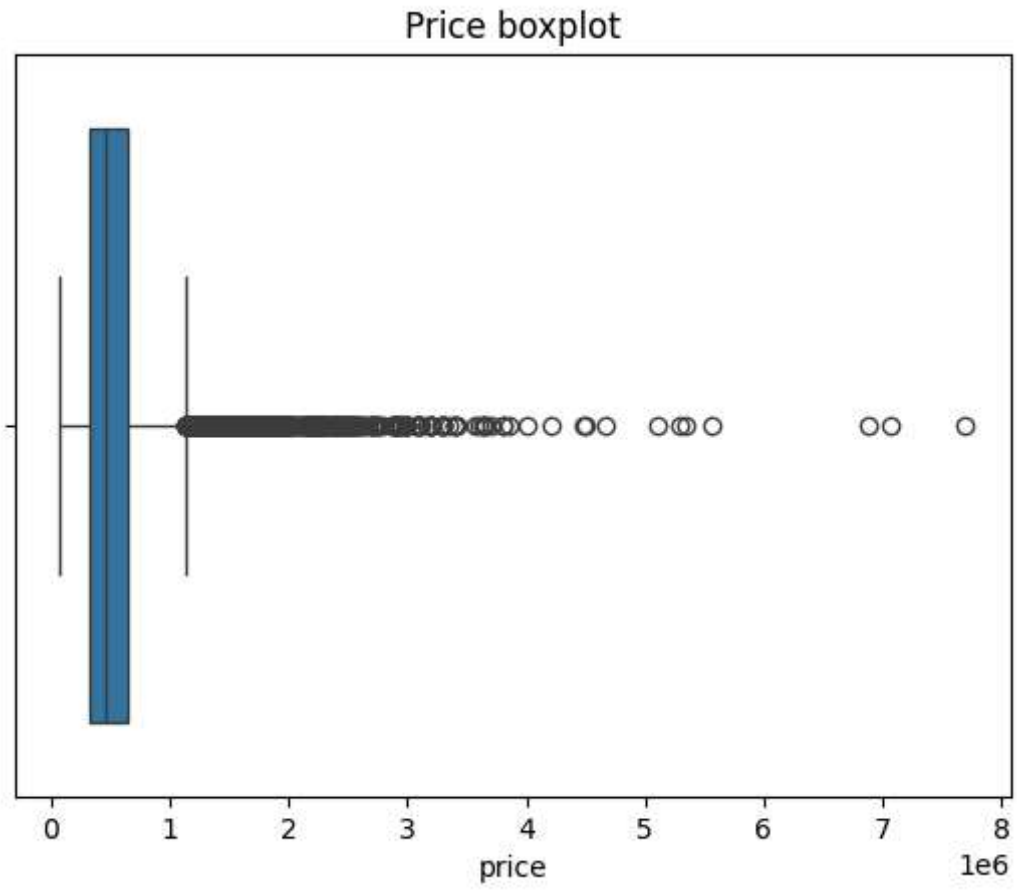
| | id | price | bedrooms | bathrooms | sqft_living | sqft_lot | f |
|---|---|---|---|---|---|---|---|
| **id** | 1.000000 | -0.016769 | 0.001307 | 0.005091 | -0.012241 | -0.132101 | 0.01 |
| **price** | -0.016769 | 1.000000 | 0.308385 | 0.525142 | 0.702037 | 0.089663 | 0.25 |
| **bedrooms** | 0.001307 | 0.308385 | 1.000000 | 0.515983 | 0.576696 | 0.031709 | 0.17 |
| **bathrooms** | 0.005091 | 0.525142 | 0.515983 | 1.000000 | 0.754688 | 0.087750 | 0.50 |
| **sqft_living** | -0.012241 | 0.702037 | 0.576696 | 0.754688 | 1.000000 | 0.172830 | 0.35 |
| **sqft_lot** | -0.132101 | 0.089663 | 0.031709 | 0.087750 | 0.172830 | 1.000000 | -0.00 |
| **floors** | 0.018480 | 0.256801 | 0.175485 | 0.500669 | 0.353954 | -0.005171 | 1.00 |
| **waterfront** | -0.002717 | 0.266371 | -0.006578 | 0.063747 | 0.103820 | 0.021601 | 0.02 |
| **view** | 0.011785 | 0.397413 | 0.079445 | 0.187891 | 0.284641 | 0.074753 | 0.02 |
| **condition** | -0.023687 | 0.036366 | 0.028452 | -0.124940 | -0.058768 | -0.008967 | -0.26 |
| **grade** | 0.008093 | 0.667450 | 0.357014 | 0.664990 | 0.762719 | 0.113644 | 0.45 |
| **sqft_above** | -0.010859 | 0.605570 | 0.477633 | 0.685351 | 0.876600 | 0.183527 | 0.52 |
| **sqft_basement** | -0.005085 | 0.323827 | 0.303095 | 0.283814 | 0.435054 | 0.015271 | -0.24 |
| **yr_built** | 0.021217 | 0.054009 | 0.154296 | 0.505997 | 0.318099 | 0.053105 | 0.48 |
| **yr_renovated** | -0.016688 | 0.126495 | 0.018705 | 0.050879 | 0.055314 | 0.007677 | 0.00 |
| **zipcode** | -0.008126 | -0.053195 | -0.152807 | -0.203831 | -0.199435 | -0.129611 | -0.05 |
| **lat** | -0.001862 | 0.307006 | -0.008935 | 0.024590 | 0.052521 | -0.085680 | 0.04 |
| **long** | 0.020776 | 0.021622 | 0.129483 | 0.223047 | 0.240221 | 0.229552 | 0.12 |
| **sqft_living15** | -0.002870 | 0.585387 | 0.391662 | 0.568665 | 0.756420 | 0.144620 | 0.27 |
| **sqft_lot15** | -0.138778 | 0.082449 | 0.029244 | 0.087191 | 0.183285 | 0.718556 | -0.01 |

In [45]:

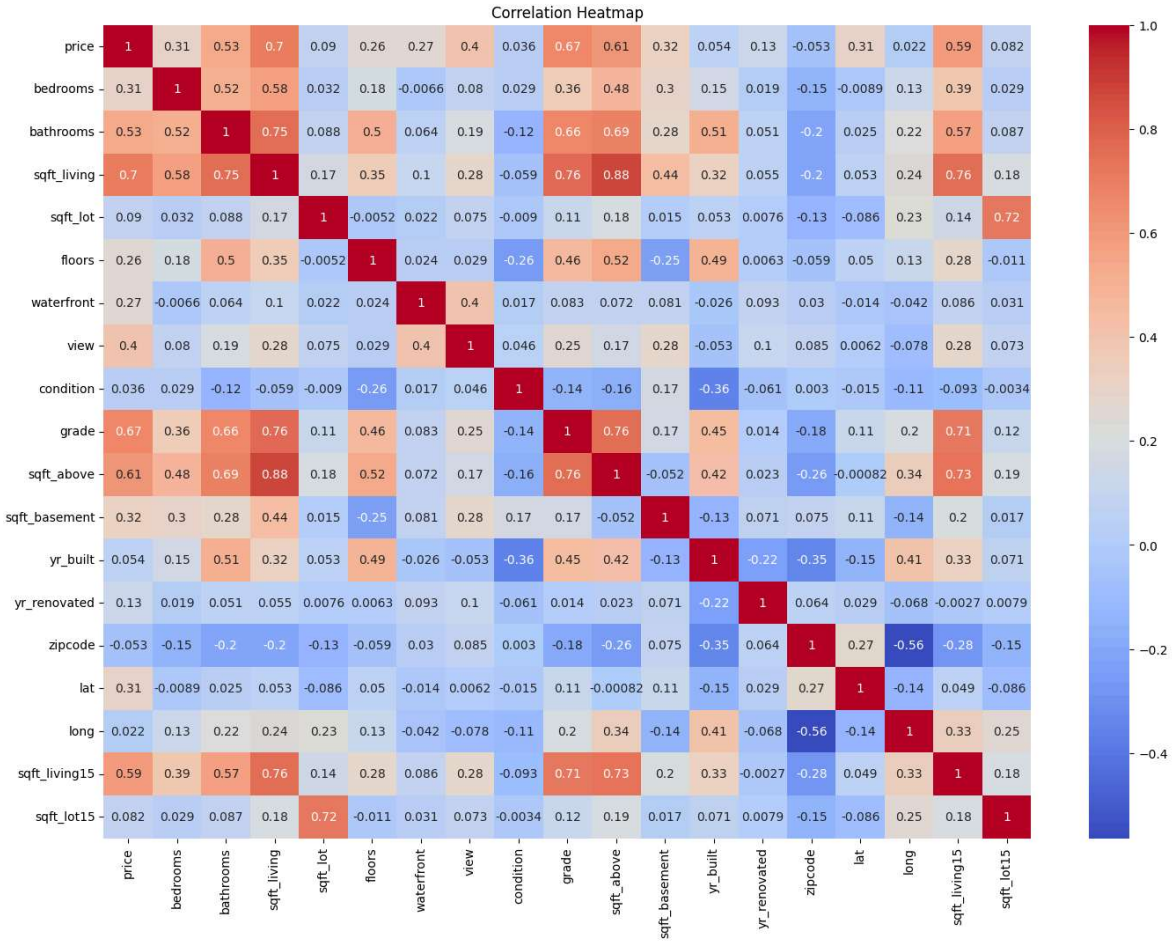
```
# Univariate Analysis : analyze distribution of target variable(price)
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure()
sns.histplot(df['price'],kde=True)
plt.title("Price Distribution")
plt.show()
```



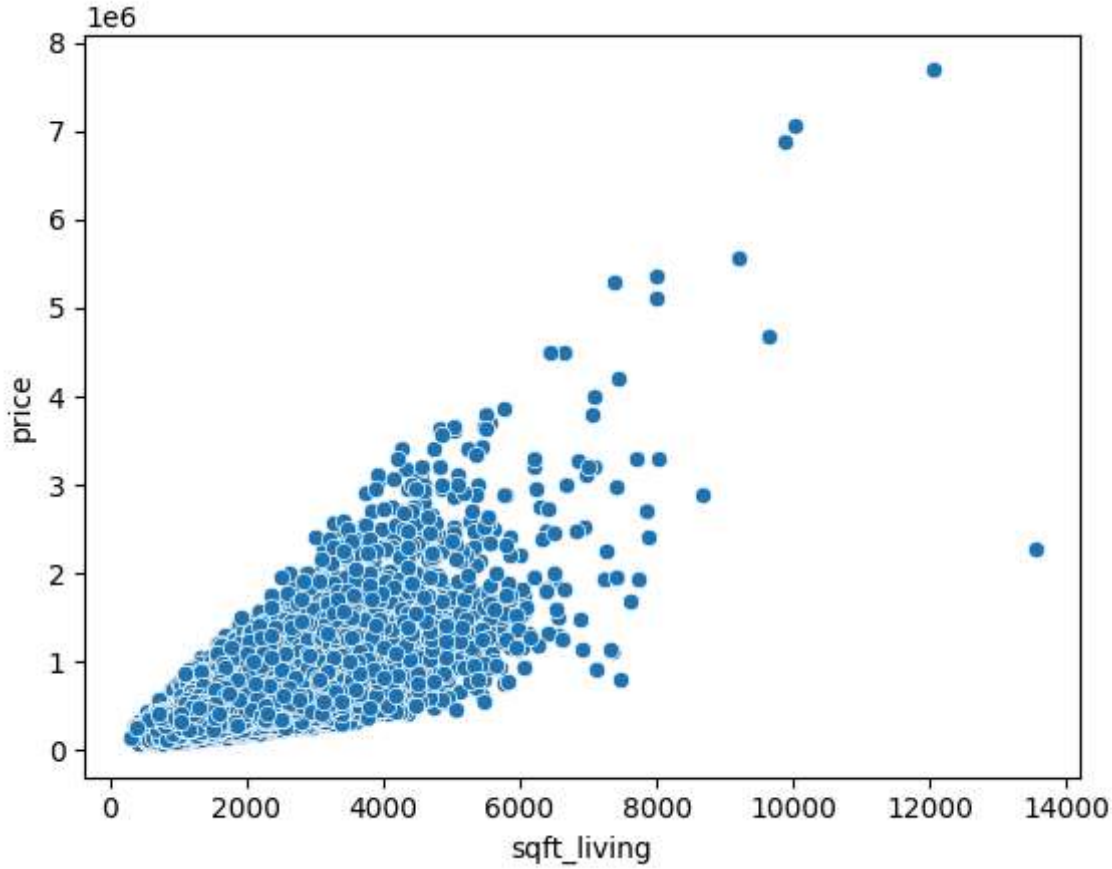

In [67]:
```
# outlier detection : Checking extreme values in price
plt.figure()
sns.boxplot(x=df['price'])
plt.title("Price boxplot")
plt.show()
```



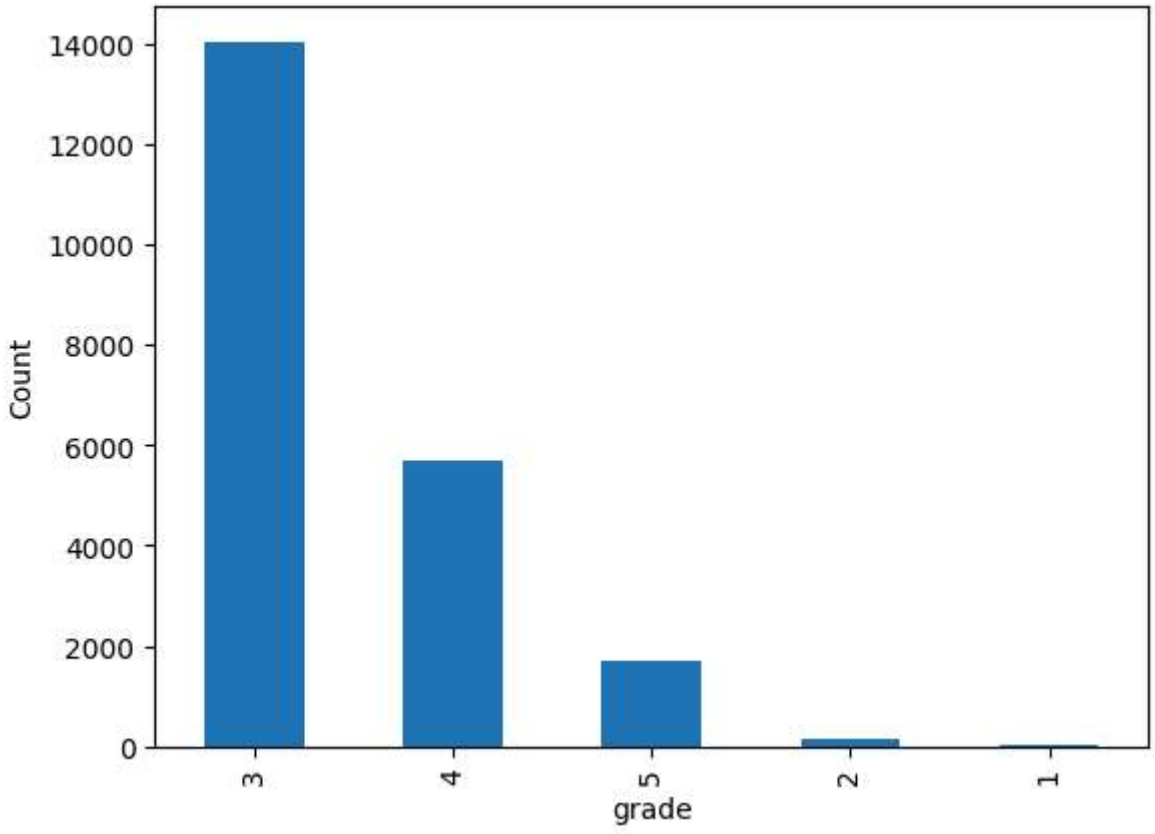

In [53]:
```
# Correlation matrix : checking relationship between features.

import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(17,12))
sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```




In [54]:
```
# Feature vs price Analysis
plt.figure()
sns.scatterplot(x='sqft_living',y='price',data=df)
plt.show()
```




In [15]:
```
#conclusion- It is an imbalanced dataset
df.condition.value_counts().plot(kind='bar')
plt.xlabel("grade")## Visualization
plt.ylabel("Count")
plt.show()
```

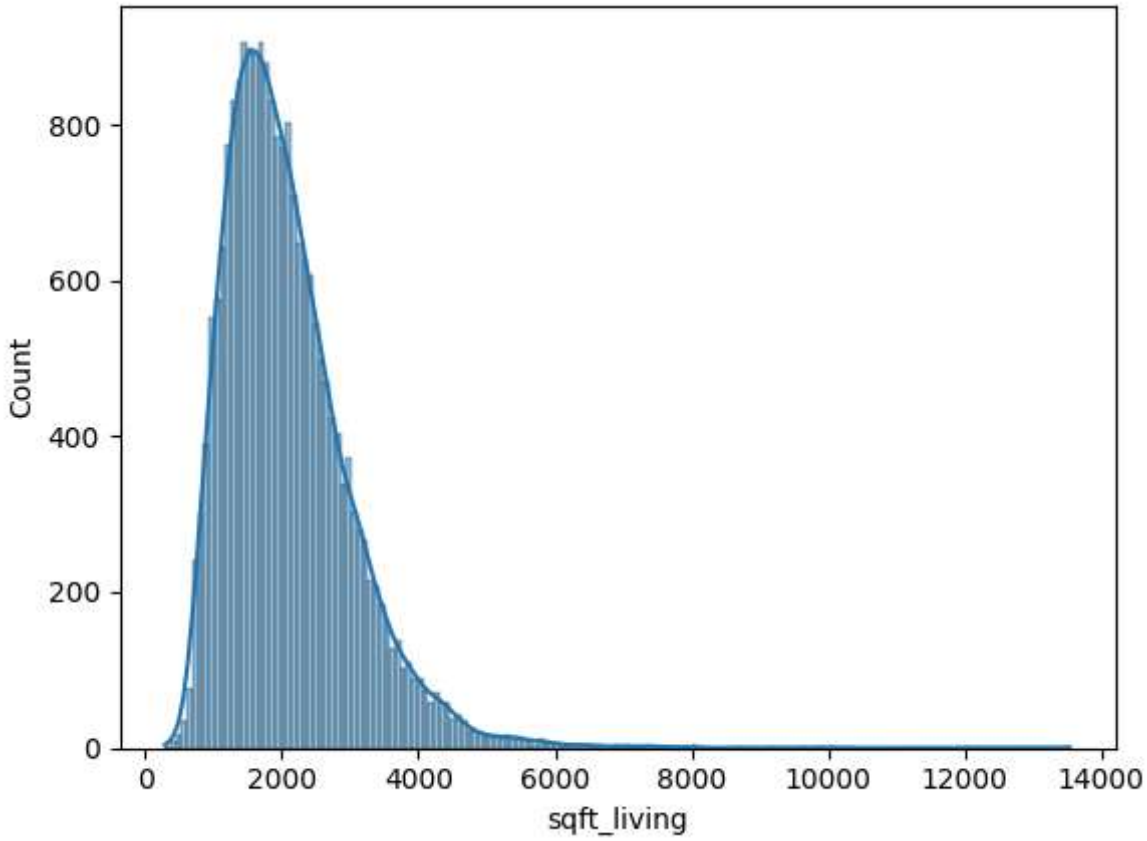


In [56]:
```
df.columns
```

Out[56]:
```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'waterfront', 'view', 'condition', 'grade', 'sqft_above',
       'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long',
       'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

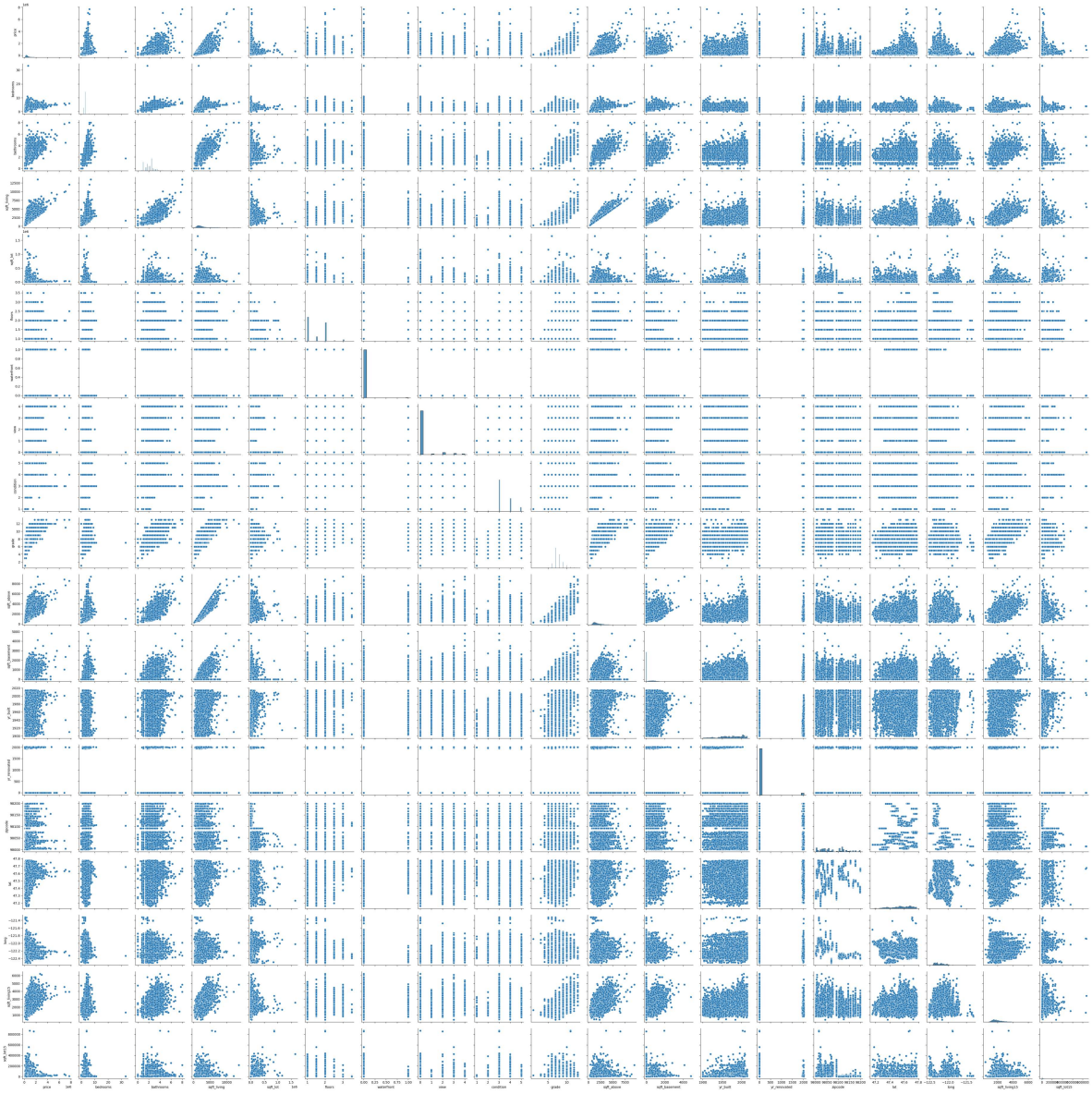
In [58]:
```
sns.histplot(df['sqft_living'],kde=True)
```

Out[58]:
```
<Axes: xlabel='sqft_living', ylabel='Count'>
```



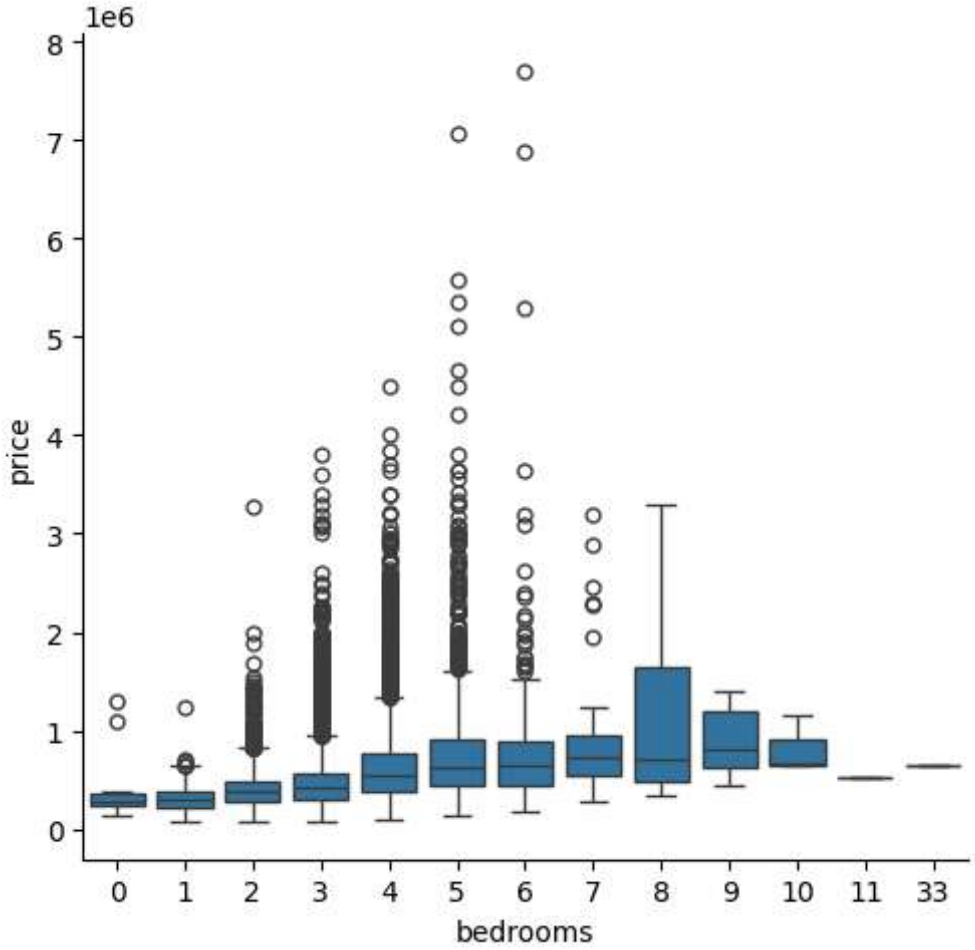

In [59]:
```
# univariate, bivariate, multivariate analysis
sns.pairplot(df)
```

Out[59]:
```
<seaborn.axisgrid.PairGrid at 0x1c2b4b06270>
```



In [65]:
```
sns.catplot(x='bedrooms',y='price',data=df,kind="box")
```

Out[65]:
```
<seaborn.axisgrid.FacetGrid at 0x1c2fb14a710>
```


1e6
8
7
6
5
4
3
2
1
0
price
0
1
2
3
4
5
6
7
8
9
10
11
33
bedrooms