In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
data=pd.read_csv("housing.csv")
data.head()
```

Out[2]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	househ
0	-122.23	37.88	41.0	880.0	129.0	322.0	1
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	11
2	-122.24	37.85	52.0	1467.0	190.0	496.0	1
3	-122.25	37.85	52.0	1274.0	235.0	558.0	2
4	-122.25	37.85	52.0	1627.0	280.0	565.0	2

In [3]:

```
data.describe()
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	p
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	2064
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	142
std	2.003532	2.135952	12.585558	2181.615252	421.385070	113
min	-124.350000	32.540000	1.000000	2.000000	1.000000	
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	78
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	116
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	172
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	3568

In [4]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [5]:

```
data.dropna(inplace=True)
```

In [6]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20433 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   longitude           20433 non-null  float64
 1   latitude            20433 non-null  float64
 2   housing_median_age  20433 non-null  float64
 3   total_rooms         20433 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20433 non-null  float64
 6   households          20433 non-null  float64
 7   median_income       20433 non-null  float64
 8   median_house_value  20433 non-null  float64
 9   ocean_proximity     20433 non-null  object 
dtypes: float64(9), object(1)
memory usage: 1.7+ MB
```

In [7]:

```
from sklearn.model_selection import train_test_split
x=data.drop(['median_house_value'],axis=1)
y=data['median_house_value']
```

In [8]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [9]:

```
train_data=x_train.join(y_train)
```

In [10]:

```
train_data
```

Out[10]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	ho
2491	-120.25	36.65	31.0	1177.0	221.0	744.0	
19110	-122.64	38.23	52.0	1075.0	249.0	519.0	
14237	-117.03	32.69	8.0	2460.0	397.0	1784.0	
17859	-121.90	37.46	29.0	2385.0	513.0	1788.0	
12288	-116.98	33.94	27.0	3459.0	640.0	1760.0	
...	...	...	...	...	...	...	
12019	-117.51	33.95	12.0	9016.0	1486.0	4285.0	
184	-122.23	37.80	52.0	1252.0	299.0	844.0	
3764	-118.41	34.17	27.0	3277.0	648.0	1382.0	
3986	-118.64	34.18	33.0	3808.0	623.0	1784.0	
20549	-121.80	38.69	8.0	3544.0	691.0	2118.0	

16346 rows × 10 columns

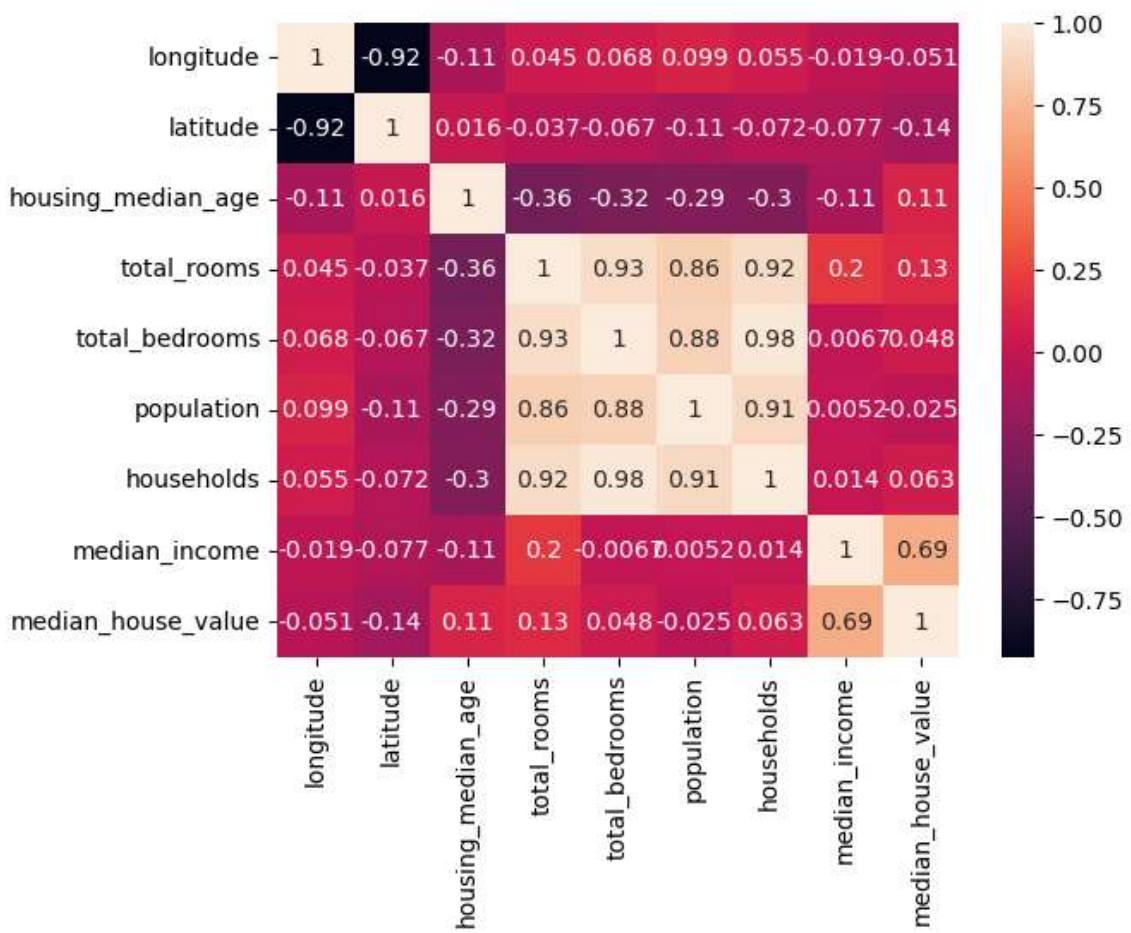
In [11]:

```
sns.heatmap(train_data.corr(),annot=True)
```

```
C:\Users\adity\AppData\Local\Temp\ipykernel_22300\3904379400.py:1: FutureW
arning: The default value of numeric_only in DataFrame.corr is deprecated.
In a future version, it will default to False. Select only valid columns o
r specify the value of numeric_only to silence this warning.
  sns.heatmap(train_data.corr(),annot=True)
```

Out[11]:

```
<Axes: >
```

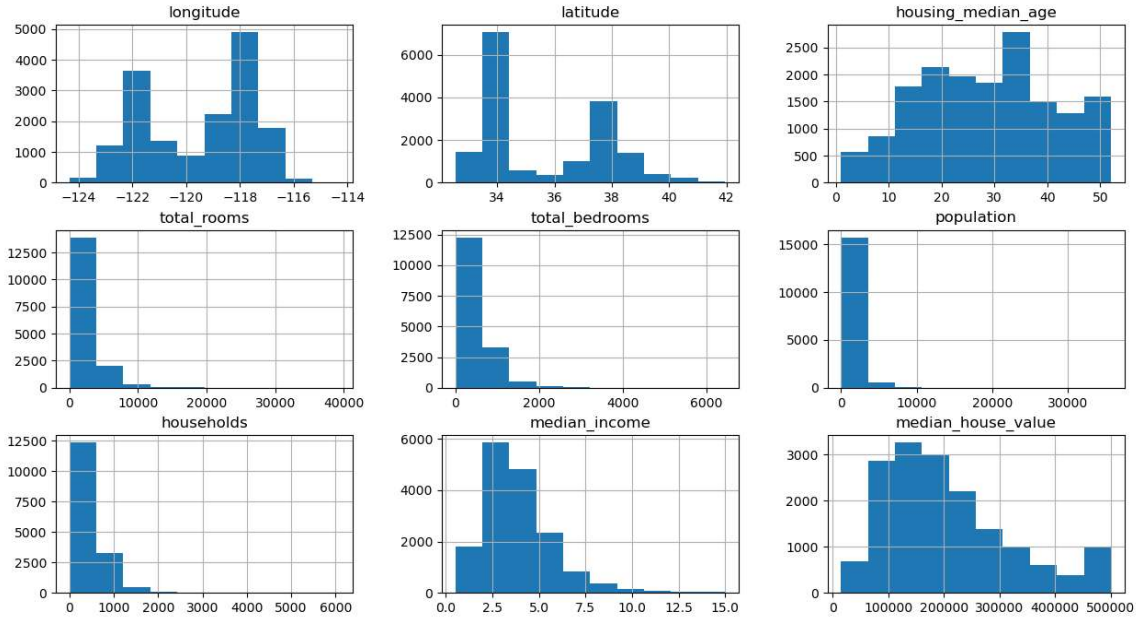


In [12]:

```
train_data.hist(figsize=(15,8))
```

Out[12]:

```
array([[<Axes: title={'center': 'longitude'}>,
        <Axes: title={'center': 'latitude'}>,
        <Axes: title={'center': 'housing_median_age'}>],
       [<Axes: title={'center': 'total_rooms'}>,
        <Axes: title={'center': 'total_bedrooms'}>,
        <Axes: title={'center': 'population'}>],
       [<Axes: title={'center': 'households'}>,
        <Axes: title={'center': 'median_income'}>,
        <Axes: title={'center': 'median_house_value'}>]], dtype=object)
```



In [13]:

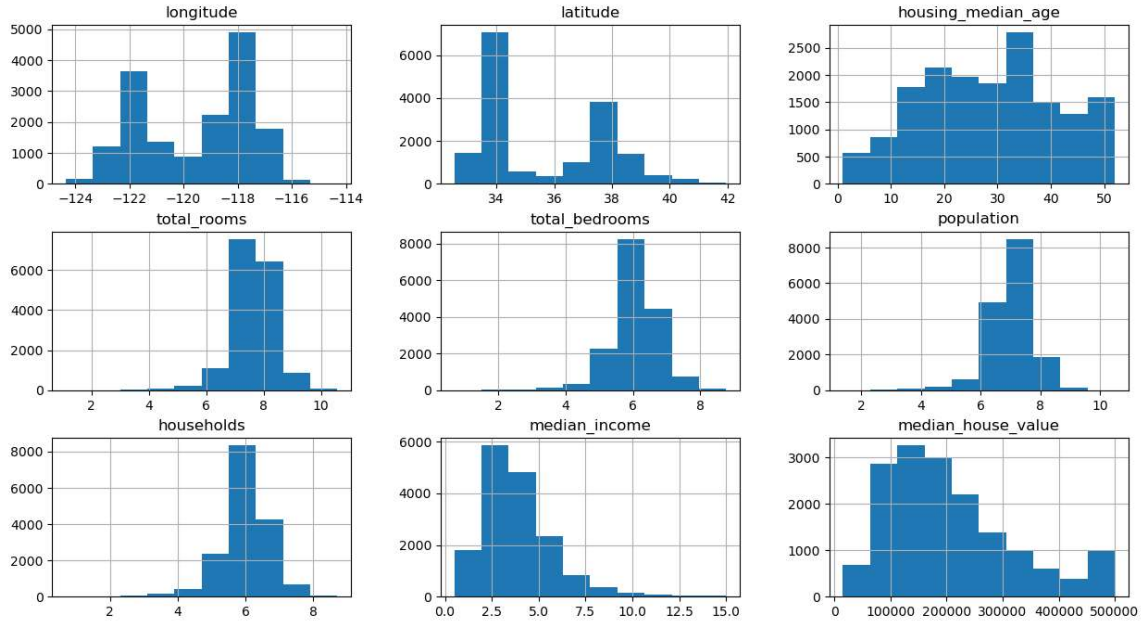
```
train_data['total_rooms']=np.log(train_data['total_rooms']+1)
train_data['total_bedrooms']=np.log(train_data['total_bedrooms']+1)
train_data['population']=np.log(train_data['population']+1)
train_data['households']=np.log(train_data['households']+1)
```

In [14]:

```
train_data.hist(figsize=(15,8))
```

Out[14]:

```
array([[<Axes: title={'center': 'longitude'}>,
        <Axes: title={'center': 'latitude'}>,
        <Axes: title={'center': 'housing_median_age'}>],
       [<Axes: title={'center': 'total_rooms'}>,
        <Axes: title={'center': 'total_bedrooms'}>,
        <Axes: title={'center': 'population'}>],
       [<Axes: title={'center': 'households'}>,
        <Axes: title={'center': 'median_income'}>,
        <Axes: title={'center': 'median_house_value'}>]], dtype=object)
```



In [18]:

```
train_data=train_data.join(pd.get_dummies(train_data.ocean_proximity)).drop(['ocean_prox
```

In [19]:

```
train_data
```

Out[19]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	ho
2491	-120.25	36.65	31.0	7.071573	5.402677	6.613384	
19110	-122.64	38.23	52.0	6.981006	5.521461	6.253829	
14237	-117.03	32.69	8.0	7.808323	5.986452	7.487174	
17859	-121.90	37.46	29.0	7.777374	6.242223	7.489412	
12288	-116.98	33.94	27.0	8.149024	6.463029	7.473637	
...	...	...	...	...	...	...	
12019	-117.51	33.95	12.0	9.106867	7.304516	8.363109	
184	-122.23	37.80	52.0	7.133296	5.703782	6.739337	
3764	-118.41	34.17	27.0	8.094989	6.475433	7.232010	
3986	-118.64	34.18	33.0	8.245122	6.436150	7.487174	
20549	-121.80	38.69	8.0	8.173293	6.539586	7.658700	

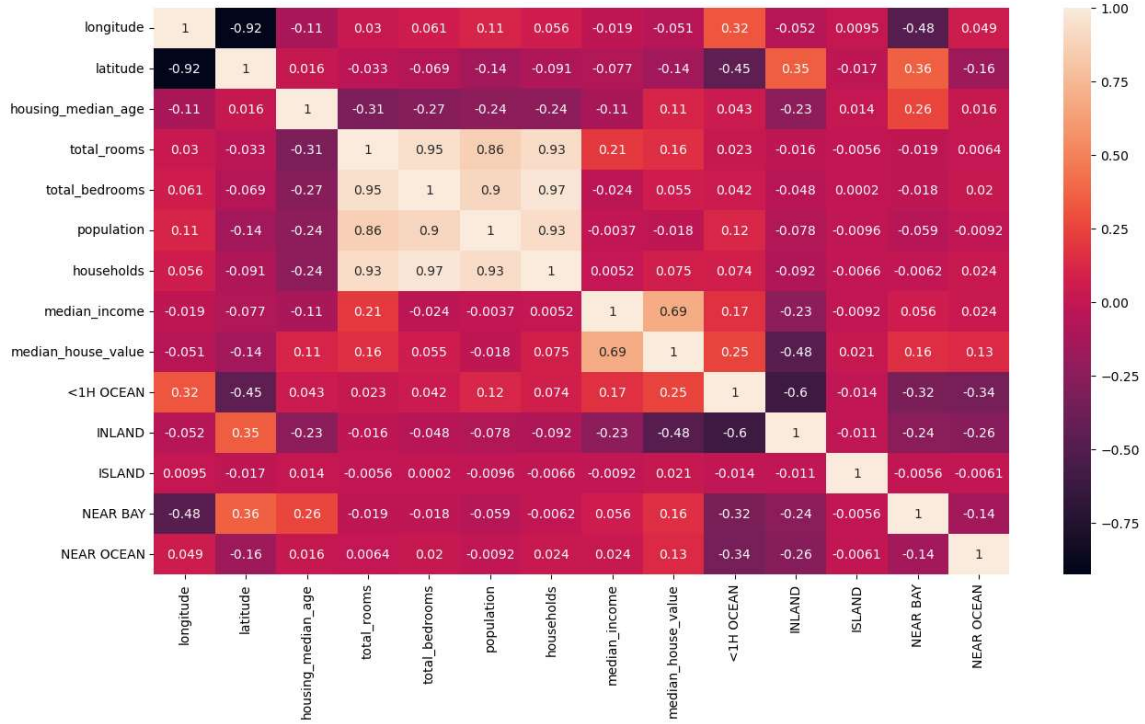
16346 rows × 14 columns

In [20]:

```
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(),annot=True)
```

Out[20]:

```
<Axes: >
```



In [21]:

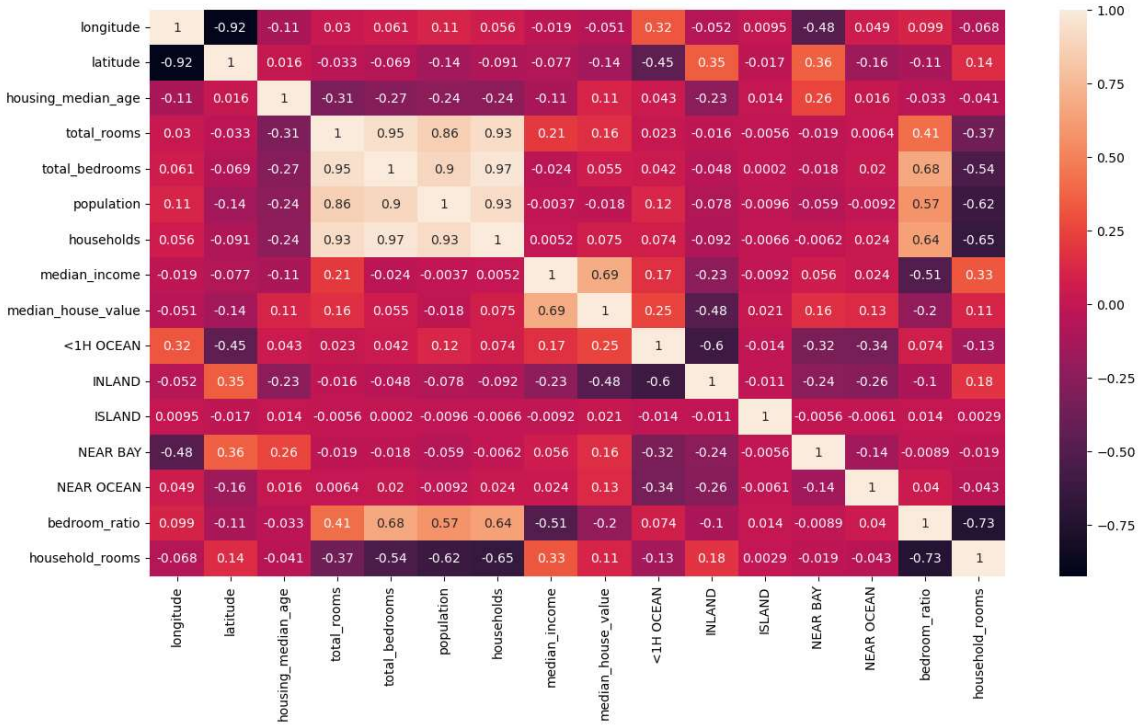
```
train_data['bedroom_ratio']=train_data['total_bedrooms']/train_data['total_rooms']
train_data['household_rooms']=train_data['total_rooms']/train_data['households']
```

In [22]:

```
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(),annot=True)
```

Out[22]:

```
<Axes: >
```



In [23]:

```
from sklearn.linear_model import LinearRegression
x_train,y_train=train_data.drop(['median_house_value'],axis=1),train_data['median_house_

reg=LinearRegression()
reg.fit(x_train,y_train)
```

Out[23]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [24]:

```
test_data = x_test.join(y_test)

test_data['total_rooms']=np.log(test_data['total_rooms']+1)
test_data['total_bedrooms']=np.log(test_data['total_bedrooms']+1)
test_data['population']=np.log(test_data['population']+1)
test_data['households']=np.log(test_data['households']+1)

test_data=test_data.join(pd.get_dummies(test_data.ocean_proximity)).drop(['ocean_proximi

test_data['bedroom_ratio']=test_data['total_bedrooms']/test_data['total_rooms']
test_data['household_rooms']=test_data['total_rooms']/test_data['households']
```

In [25]:

```
x_test,y_test=test_data.drop(['median_house_value'],axis=1),test_data['median_house_valu
```

In [26]:

```
reg.score(x_test,y_test)
```

Out[26]:

```
0.6731988363117716
```

In [ ]:

```

```