```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

```
data = pd.read_csv('housing.csv')
```

```
data.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	med
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	

```
data.shape
```

```
(20640, 10)
```

```
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
```

```
data.isnull().sum()
```

```
longitude               0
latitude                0
housing_median_age      0
total_rooms             0
total_bedrooms        207
population              0
households              0
median_income           0
median_house_value      0
ocean_proximity         0
dtype: int64
```

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

```
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
```

```
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
```

```
data.isnull().sum()
```

```
data.shape
```

```
(20433, 10)
```

```
X = data.drop(['median_house_value'], axis=1)
Y = data['median_house_value']
```

```
print(X)
print(Y)
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0        -122.23     37.88                41.0        880.0           129.0   
1        -122.22     37.86                21.0       7099.0          1106.0   
2        -122.24     37.85                52.0       1467.0           190.0   
3        -122.25     37.85                52.0       1274.0           235.0   
4        -122.25     37.85                52.0       1627.0           280.0   
...          ...       ...                 ...          ...             ...   
20635    -121.09     39.48                25.0       1665.0           374.0   
20636    -121.21     39.49                18.0        697.0           150.0   
20637    -121.22     39.43                17.0       2254.0           485.0   
20638    -121.32     39.43                18.0       1860.0           409.0   
20639    -121.24     39.37                16.0       2785.0           616.0   
```

```
       population  households  median_income ocean_proximity
0           322.0       126.0         8.3252        NEAR BAY
1          2401.0      1138.0         8.3014        NEAR BAY
2           496.0       177.0         7.2574        NEAR BAY
3           558.0       219.0         5.6431        NEAR BAY
4           565.0       259.0         3.8462        NEAR BAY
...           ...         ...            ...             ...
20635       845.0       330.0         1.5603          INLAND
20636       356.0       114.0         2.5568          INLAND
20637      1007.0       433.0         1.7000          INLAND
20638       741.0       349.0         1.8672          INLAND
20639      1387.0       530.0         2.3886          INLAND

[20433 rows x 9 columns]
0        452600.0
1        358500.0
2        352100.0
3        341300.0
4        342200.0
           ...
20635     78100.0
20636     77100.0
20637     92300.0
20638     84700.0
20639     89400.0
Name: median_house_value, Length: 20433, dtype: float64
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2)
```

```
train_data = X_train.join(Y_train)
train_data
```

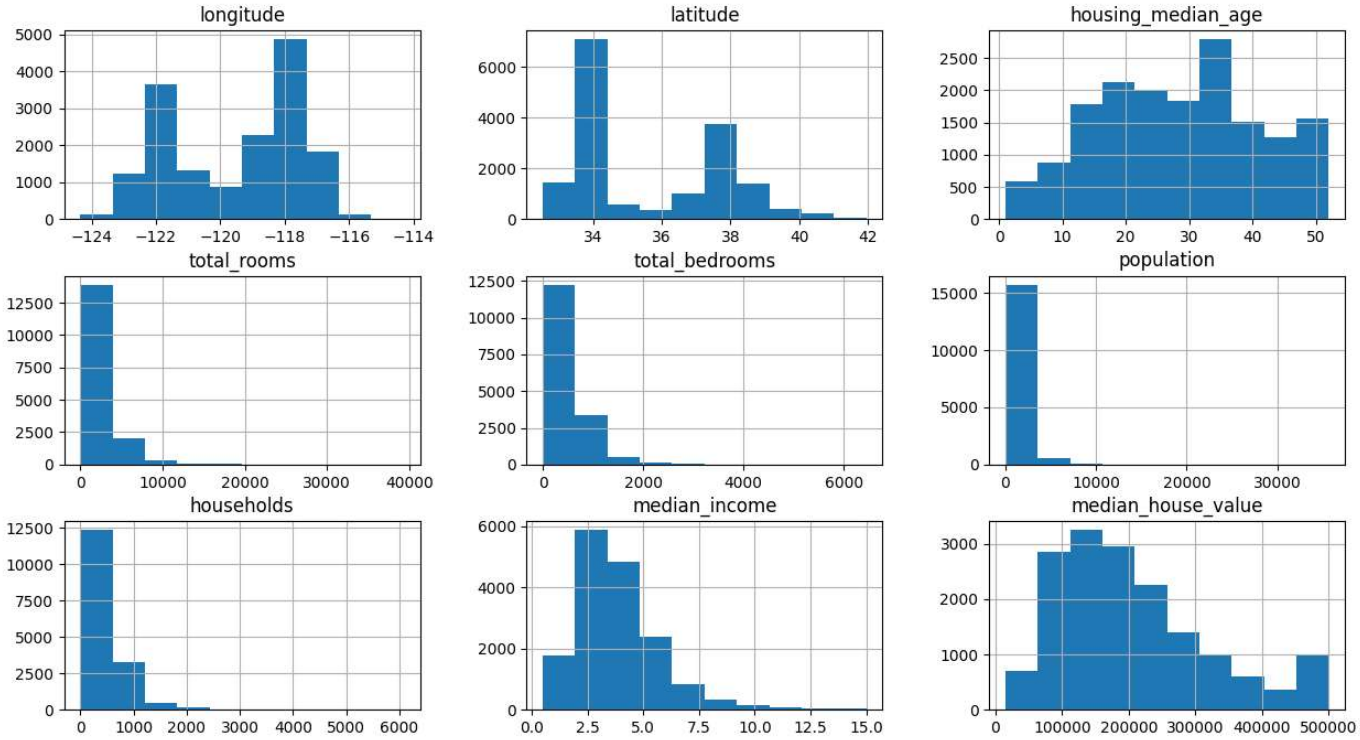
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
18334	-122.16	37.45	37.0	2926.0	874.0	1363.0	815.0	4.5987
3630	-118.42	34.22	29.0	1807.0	323.0	1234.0	310.0	5.3767
16629	-120.82	35.32	12.0	3522.0	683.0	1780.0	662.0	3.3958
10586	-117.77	33.70	15.0	1392.0	267.0	681.0	263.0	5.4248
8182	-118.10	33.80	37.0	1814.0	329.0	850.0	328.0	5.0574
...	...	...	...	...	...	...	...	...
19244	-122.76	38.52	6.0	2073.0	388.0	826.0	375.0	3.0550
19513	-121.03	37.62	43.0	1241.0	240.0	612.0	266.0	2.8194
15458	-117.26	33.18	9.0	4540.0	793.0	2235.0	746.0	4.5781
2262	-119.83	36.79	24.0	3505.0	819.0	2098.0	774.0	1.9575
7742	-118.16	33.94	25.0	3341.0	789.0	1685.0	751.0	3.6936

16346 rows × 10 columns

Next steps: Generate code with `train_data` View recommended plots

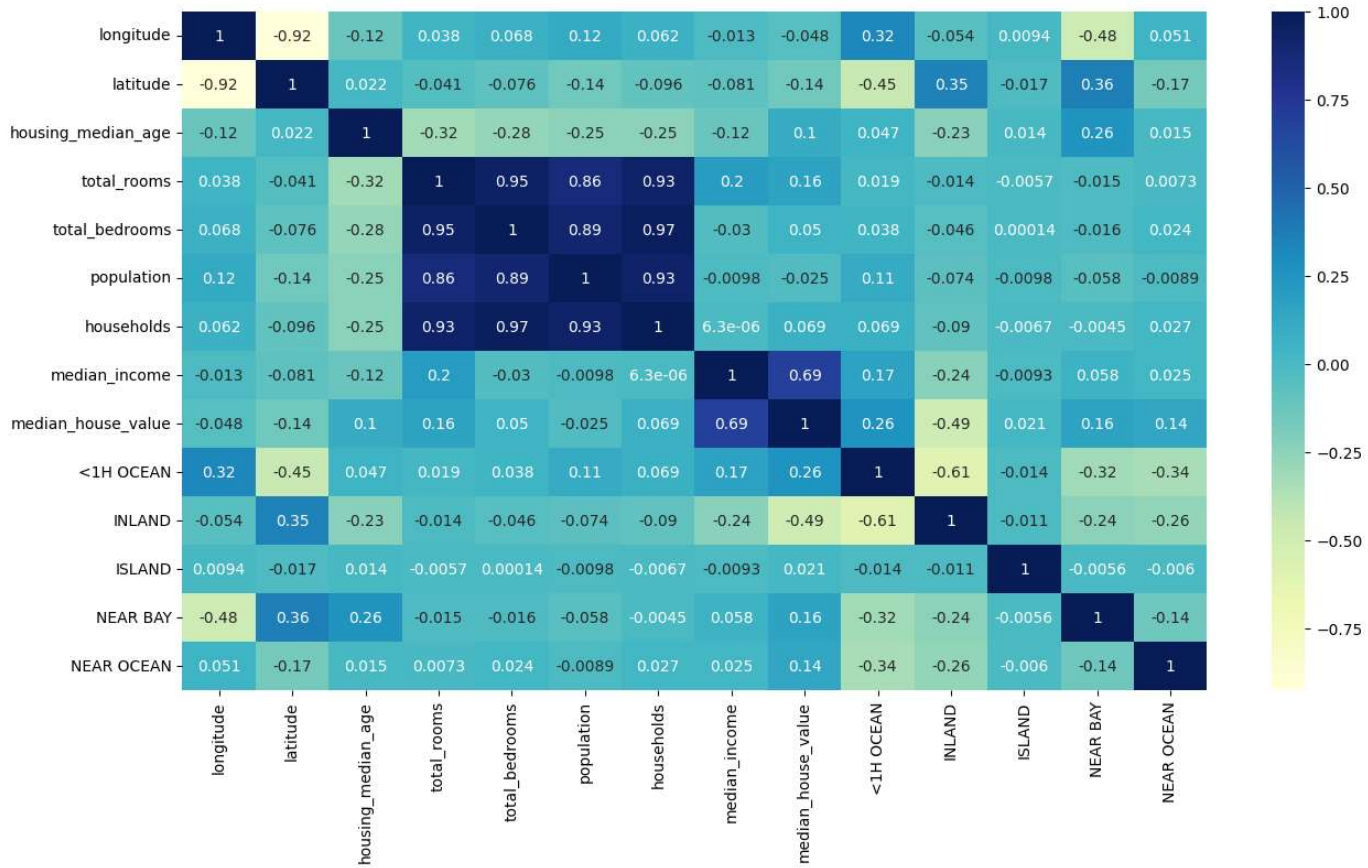
```
train_data.hist(figsize=(15,8))
```

```
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)
```



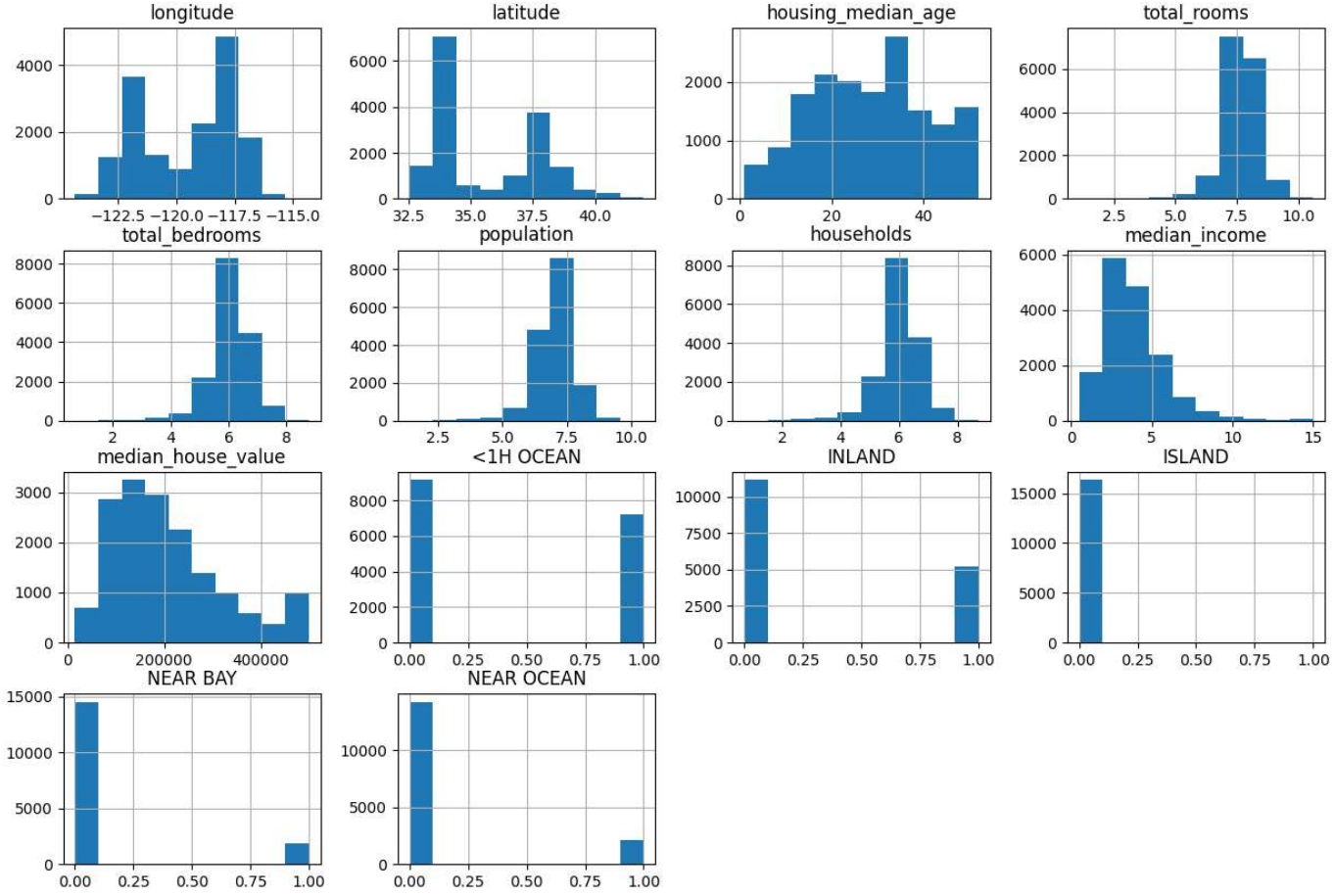

```
train_data = train_data.join(pd.get_dummies(train_data.ocean_proximity)).drop(["ocean_proximity"],axis=1)


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)


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

```
<Axes: >
```



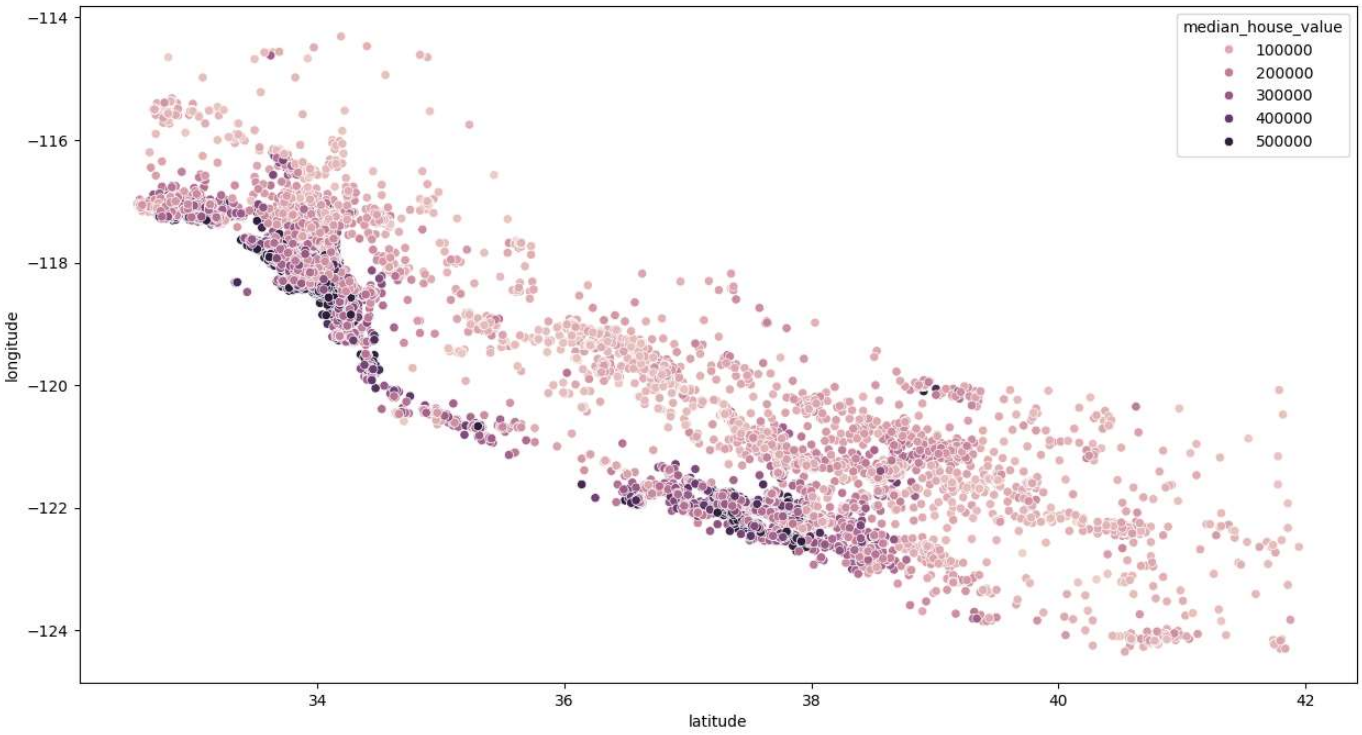

```
train_data.hist(figsize=(15,10))
# They look noramal distribution (Gaussian bell)
```

```
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'}>,
        <Axes: title={'center': '<1H OCEAN'}>,
        <Axes: title={'center': 'INLAND'}>,
        <Axes: title={'center': 'ISLAND'}>],
       [<Axes: title={'center': 'NEAR BAY'}>,
        <Axes: title={'center': 'NEAR OCEAN'}>, <Axes: >, <Axes: >]],
      dtype=object)
```




```
plt.figure(figsize=(15,8))
sns.scatterplot(x='latitude', y='longitude', data=train_data, hue='median_house_value')
```

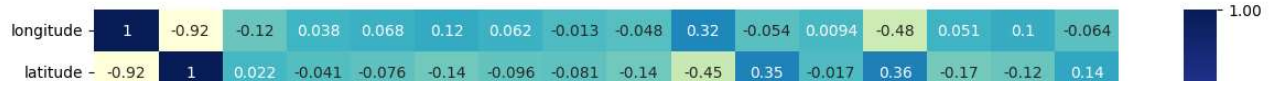
```
<Axes: xlabel='latitude', ylabel='longitude'>
```




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

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

```
<Axes: >
```




```
X_train , Y_train = train_data.drop(['median_house_value'], axis=1) , train_data['median_house_value']
```


```
reg = LinearRegression()
reg.fit(X_train,Y_train)
```

```
▾ LinearRegression
LinearRegression()
```


```
reg.score(X_train , Y_train)
```

```
0.6694780165489402
```


```
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_proximity"],axis=1)
test_data['bedroom_ratio'] = test_data['total_bedrooms'] / test_data['total_rooms']
test_data['household_rooms'] = test_data['total_rooms'] / test_data['households']


test_data
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
19804	-123.48	40.34	19.0	6.251904	4.691348	5.379897	4.394449	2.7083
4280	-118.32	34.09	32.0	6.335054	5.257495	6.440947	5.225747	2.0341
7186	-118.18	34.03	39.0	6.413459	4.983607	6.538140	4.905275	2.9167
1720	-122.33	37.99	4.0	8.294050	6.984716	7.372746	6.788972	3.9110