CaliforniaHousePrices

July 29, 2024

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

0.1 Importing data

[2]:
```
data = pd.read_csv("CaliforniaHousing.csv")
```

[3]:
```
data
```

[3]:
```
       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  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
…             …           …            …                   …
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
20637      1007.0       433.0         1.7000             92300.0
20638       741.0       349.0         1.8672             84700.0
20639      1387.0       530.0         2.3886             89400.0

      ocean_proximity
```

```
0           NEAR BAY
1           NEAR BAY
2           NEAR BAY
3           NEAR BAY
4           NEAR BAY
…              …
20635         INLAND
20636         INLAND
20637         INLAND
20638         INLAND
20639         INLAND

[20640 rows x 10 columns]
```

[4]:
```python
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
```

[5]:
```python
data.dropna(inplace=True)
```

[6]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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
```

[7]:
```
from sklearn.model_selection import train_test_split

x = data.drop(['median_house_value'], axis=1)
y = data['median_house_value']

y
```

[7]:
```
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
```

[8]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2)
```

[9]:
```
train_data = x_train.join(y_train)
```

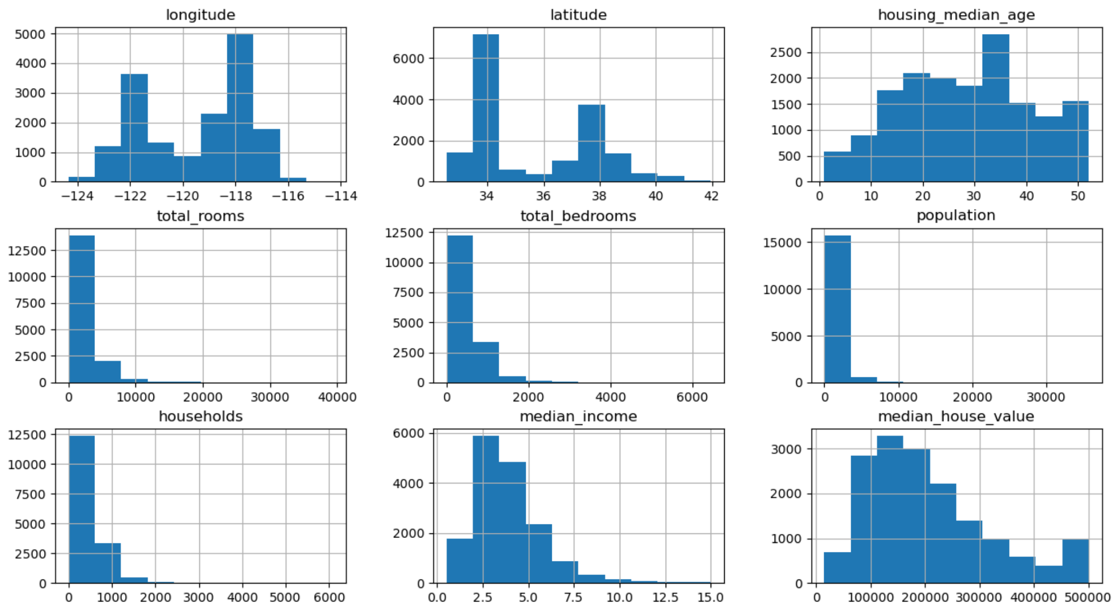
[10]:
```
train_data
```

[10]:
```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
11787    -121.24     38.79                15.0       2615.0           485.0
18773    -122.29     40.47                20.0       2858.0           612.0
3192     -119.72     36.34                33.0       1287.0           214.0
6669     -118.11     34.16                52.0       3158.0           459.0
4765     -118.35     34.04                38.0       1626.0           375.0
…          …       …                  …          …              …
3443     -118.41     34.25                19.0        280.0            84.0
9293     -122.53     38.01                27.0       3121.0           531.0
11095    -117.88     33.84                31.0       3301.0           712.0
20099    -120.24     37.96                34.0       1747.0           395.0
8924     -118.51     34.00                52.0       1241.0           502.0

       population  households  median_income ocean_proximity  \
```

```
11787      1063.0      428.0        3.7904        INLAND
18773      1422.0      589.0        1.9657        INLAND
3192        580.0      210.0        3.2019        INLAND
6669       1229.0      444.0        5.4223        INLAND
4765       1019.0      372.0        2.3687     <1H OCEAN
…             …          …             …             …
3443        483.0       87.0        1.9500     <1H OCEAN
9293       1318.0      489.0        5.4781      NEAR BAY
11095      1532.0      682.0        3.7303     <1H OCEAN
20099       935.0      362.0        1.6250        INLAND
8924        679.0      459.0        2.3098     <1H OCEAN

       median_house_value
11787            173200.0
18773             63000.0
3192             112500.0
6669             325600.0
4765             146800.0
…                     …
3443             137500.0
9293             310900.0
11095            223800.0
20099             79400.0
8924             500001.0

[16346 rows x 10 columns]
```

[11]:
```python
train_data.hist(figsize=(15,8))
```

[11]:
```
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)
```



[12]:
```python
new_data = train_data.drop(['ocean_proximity'], axis=1)
new_data.corr()
```

[12]:

	longitude	latitude	housing_median_age	total_rooms
longitude	1.000000	-0.924099	-0.110697	0.049000
latitude	-0.924099	1.000000	0.012777	-0.041604
housing_median_age	-0.110697	0.012777	1.000000	-0.358862
total_rooms	0.049000	-0.041604	-0.358862	1.000000
total_bedrooms	0.073939	-0.072184	-0.317790	0.929800
population	0.102691	-0.112643	-0.290141	0.855490
households	0.060572	-0.077259	-0.299187	0.919934
median_income	-0.017918	-0.079752	-0.121499	0.202397
median_house_value	-0.046663	-0.144848	0.103626	0.136636

	total_bedrooms	population	households	median_income
longitude	0.073939	0.102691	0.060572	-0.017918
latitude	-0.072184	-0.112643	-0.077259	-0.079752
housing_median_age	-0.317790	-0.290141	-0.299187	-0.121499
total_rooms	0.929800	0.855490	0.919934	0.202397
total_bedrooms	1.000000	0.875771	0.979891	-0.006319
population	0.875771	1.000000	0.905277	0.007335
households	0.979891	0.905277	1.000000	0.015310
median_income	-0.006319	0.007335	0.015310	1.000000
median_house_value	0.051181	-0.022465	0.067177	0.690243

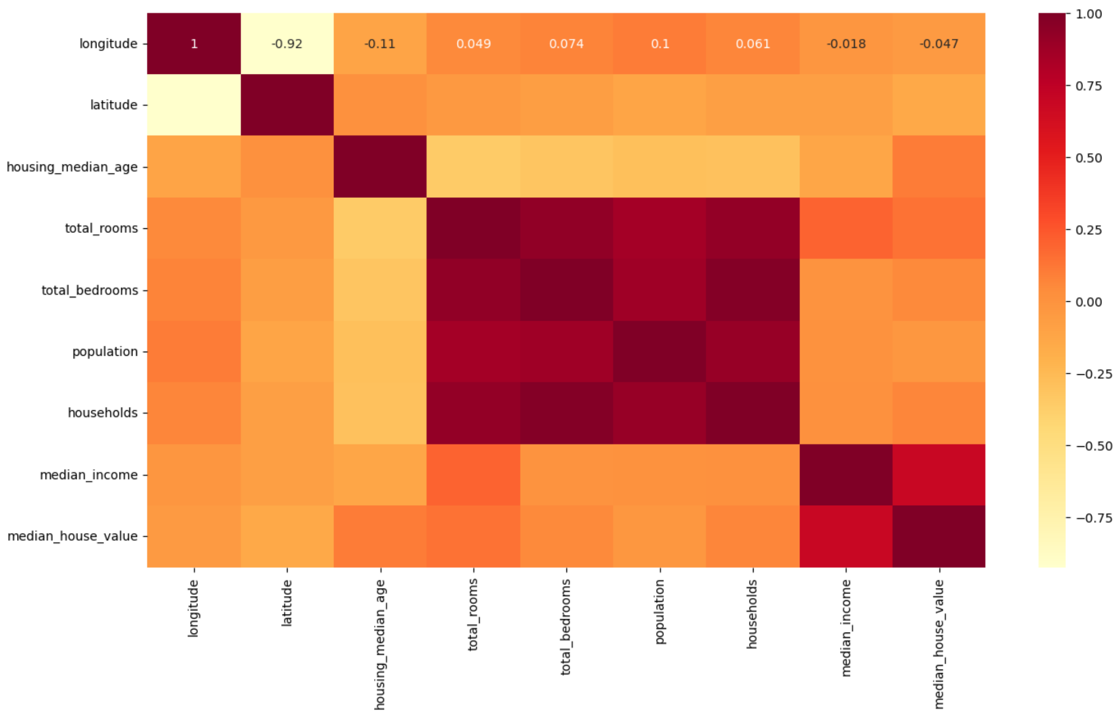
	median_house_value
longitude	-0.046663

```
latitude                   -0.144848
housing_median_age          0.103626
total_rooms                 0.136636
total_bedrooms              0.051181
population                 -0.022465
households                  0.067177
median_income               0.690243
median_house_value          1.000000
```

Checking correlation coefficients between parameters

[13]:
```python
plt.figure(figsize=(15,8))
sns.heatmap(new_data.corr(), annot=True, cmap="YlOrRd")
```

[13]:
```
<Axes: >
```

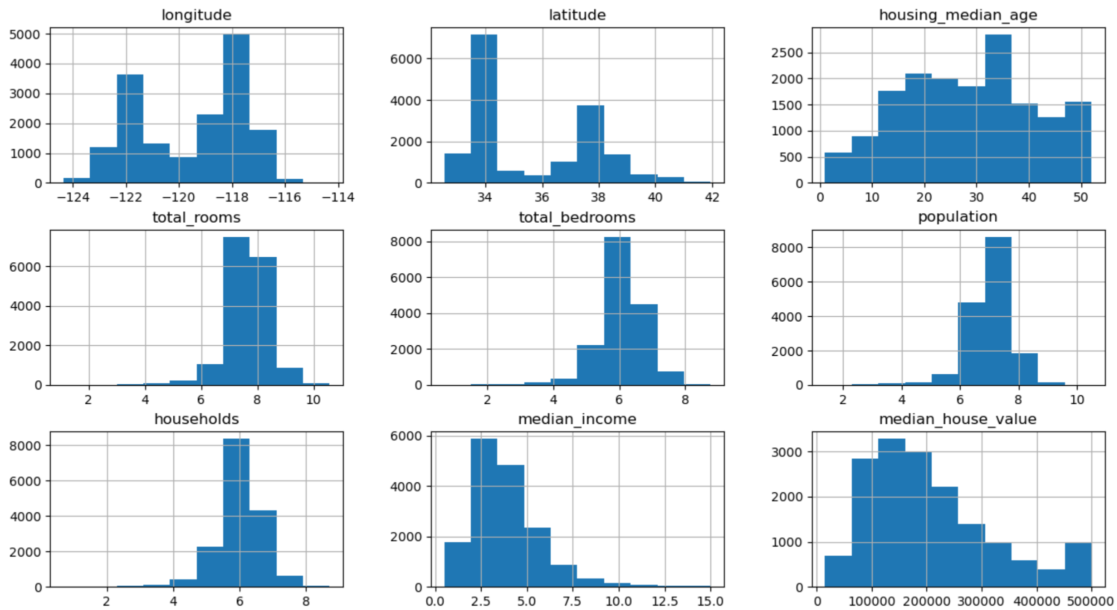


0.2 Data normalization

[14]:
```python
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)
```

[15]:
```python
train_data.hist(figsize=(15,8))
```

[15]:
```
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)
```



[16]:
```
train_data.ocean_proximity.value_counts()
```

[16]:
```
ocean_proximity
<1H OCEAN     7271
INLAND        5182
NEAR OCEAN    2083
NEAR BAY      1807
ISLAND           3
Name: count, dtype: int64
```

[17]:
```
#pro_train_data = train_data.join(pd.get_dummies(train_data.ocean_proximity))
```

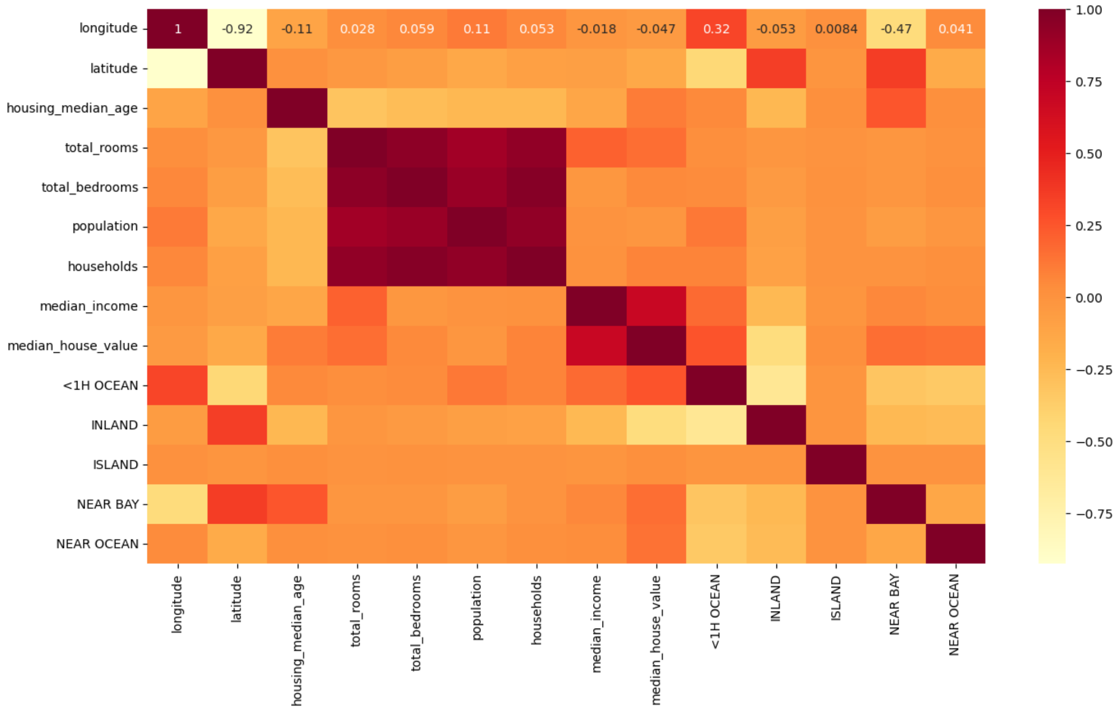
[18]:
```
train_data = train_data.join(pd.get_dummies(train_data.ocean_proximity)).
 ↪drop(['ocean_proximity'], axis=1)
```

[19]:
```
train_data
```

[19]:
```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
11787    -121.24     38.79                15.0     7.869402        6.186209
18773    -122.29     40.47                20.0     7.958227        6.418365
3192     -119.72     36.34                33.0     7.160846        5.370638
6669     -118.11     34.16                52.0     8.058011        6.131226
4765     -118.35     34.04                38.0     7.394493        5.929589
…            …         …                   …          …             …
3443     -118.41     34.25                19.0     5.638355        4.442651
9293     -122.53     38.01                27.0     8.046229        6.276643
11095    -117.88     33.84                31.0     8.102284        6.569481
20099    -120.24     37.96                34.0     7.466228        5.981414
8924     -118.51     34.00                52.0     7.124478        6.220590

       population  households  median_income  median_house_value  <1H OCEAN  \
11787    6.969791    6.061457         3.7904            173200.0      False
18773    7.260523    6.380123         1.9657             63000.0      False
3192     6.364751    5.351858         3.2019            112500.0      False
6669     7.114769    6.098074         5.4223            325600.0      False
4765     6.927558    5.921578         2.3687            146800.0       True
…             …           …              …                  …         …
3443     6.182085    4.477337         1.9500            137500.0       True
9293     7.184629    6.194405         5.4781            310900.0      False
11095    7.334982    6.526495         3.7303            223800.0       True
20099    6.841615    5.894403         1.6250             79400.0      False
8924     6.522093    6.131226         2.3098            500001.0       True

       INLAND  ISLAND  NEAR BAY  NEAR OCEAN
11787    True   False     False       False
18773    True   False     False       False
3192     True   False     False       False
6669     True   False     False       False
4765    False   False     False       False
…        …       …        …          …
3443    False   False     False       False
9293    False   False      True       False
11095   False   False     False       False
20099    True   False     False       False
8924    False   False     False       False

[16346 rows x 14 columns]
```

[20]:
```python
plt.figure(figsize=(15,8))
sns.heatmap(train_data.corr(), annot=True, cmap="YlOrRd")
```

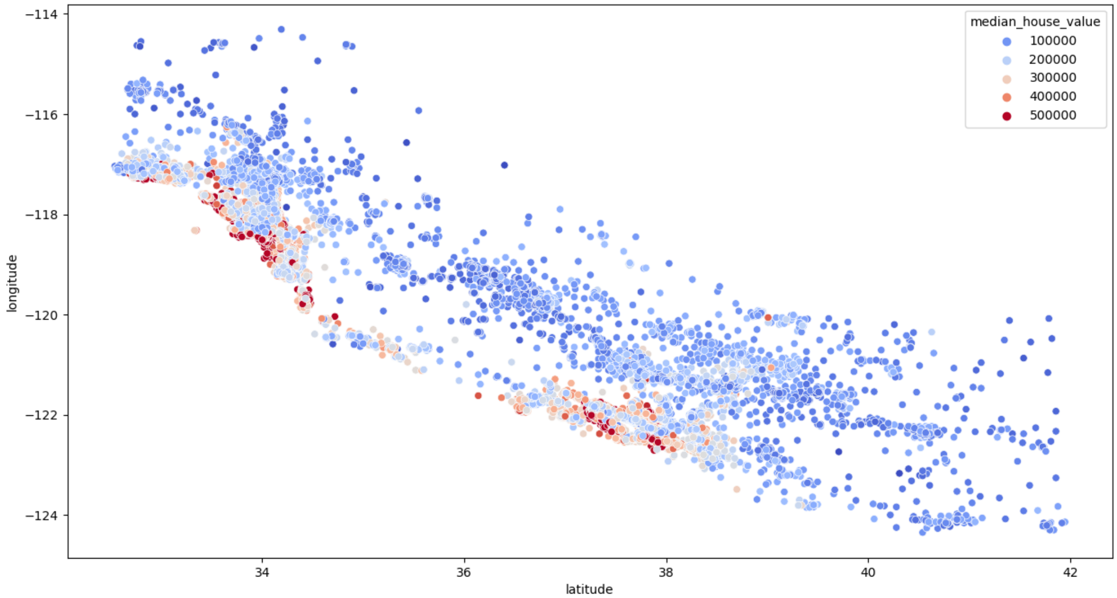
[20]: <Axes: >



0.3 Plotting the median house value

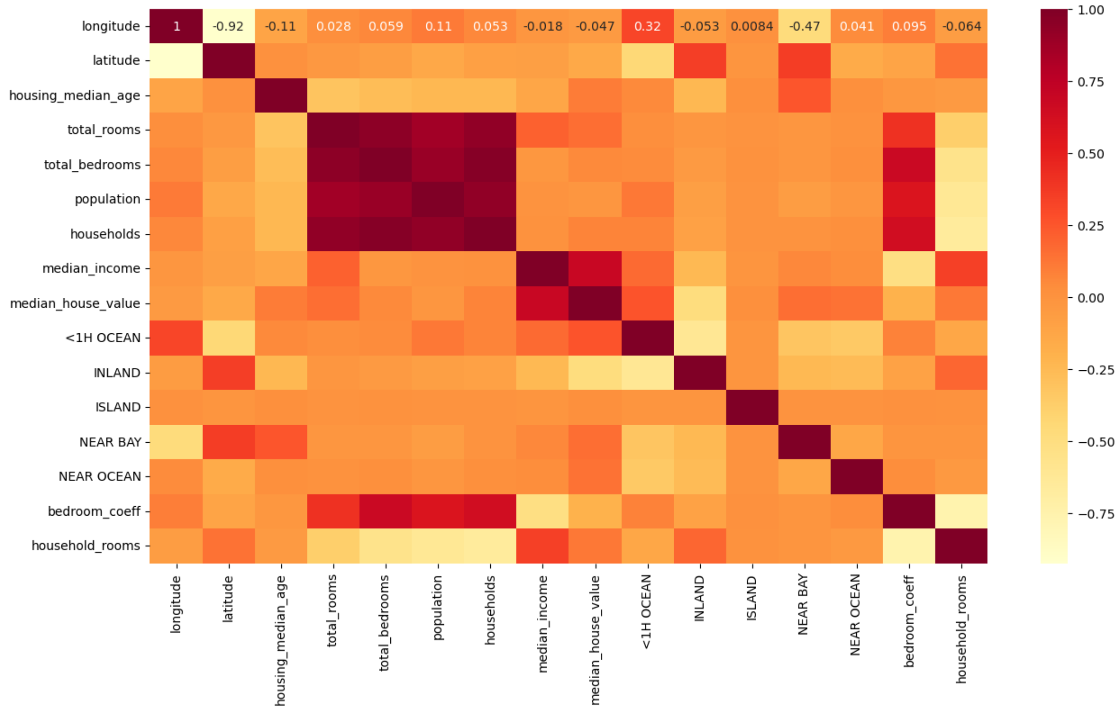
[21]:
```
plt.figure(figsize=(15,8))
sns.scatterplot(x='latitude', y='longitude', data=train_data,␣
 ↪hue="median_house_value", palette="coolwarm")
```

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




```
[22]: train_data['bedroom_coeff'] = 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="YlOrRd")
      #sns.heatmap(train_data[['median_house_value', 'bedroom_coeff',␣
       ↪'household_rooms']].corr(), annot=True, cmap="YlOrRd")
```

```
[22]: <Axes: >
```



0.4 Building a regression model

[23]:
```python
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
x_train, y_train = train_data.drop(['median_house_value'], axis=1),␣
↪train_data['median_house_value']
x_train_s = scaler.fit_transform(x_train)

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

[23]:
```
LinearRegression()
```

[24]:
```python
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_coeff'] = test_data['total_bedrooms']/
 ↪test_data['total_rooms']
test_data['household_rooms'] = test_data['total_rooms']/test_data['households']
```

[25]:
```
x_test, y_test = test_data.drop(['median_house_value'], axis=1),␣
 ↪test_data['median_house_value']
```

[26]:
```
x_test_s = scaler.transform(x_test)
```

[27]:
```
reg.score(x_test_s, y_test)
```

[27]: 0.6714006920334379

[28]:
```
x_train
```

[28]:
```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
11787    -121.24     38.79                15.0     7.869402        6.186209
18773    -122.29     40.47                20.0     7.958227        6.418365
3192     -119.72     36.34                33.0     7.160846        5.370638
6669     -118.11     34.16                52.0     8.058011        6.131226
4765     -118.35     34.04                38.0     7.394493        5.929589
…            …         …                   …          …               …
3443     -118.41     34.25                19.0     5.638355        4.442651
9293     -122.53     38.01                27.0     8.046229        6.276643
11095    -117.88     33.84                31.0     8.102284        6.569481
20099    -120.24     37.96                34.0     7.466228        5.981414
8924     -118.51     34.00                52.0     7.124478        6.220590

       population  households  median_income  <1H OCEAN  INLAND  ISLAND  \
11787    6.969791    6.061457         3.7904      False    True   False
18773    7.260523    6.380123         1.9657      False    True   False
3192     6.364751    5.351858         3.2019      False    True   False
6669     7.114769    6.098074         5.4223      False    True   False
4765     6.927558    5.921578         2.3687       True   False   False
…             …           …              …          …       …
3443     6.182085    4.477337         1.9500       True   False   False
9293     7.184629    6.194405         5.4781      False   False   False
11095    7.334982    6.526495         3.7303       True   False   False
20099    6.841615    5.894403         1.6250      False    True   False
8924     6.522093    6.131226         2.3098       True   False   False

       NEAR BAY  NEAR OCEAN  bedroom_coeff  household_rooms
11787     False       False       0.786109         1.298269
18773     False       False       0.806507         1.247347
3192      False       False       0.750001         1.338011
6669      False       False       0.760886         1.321403
```

```
4765      False      False      0.801893        1.248737
…         …       …           …              …
3443      False      False      0.787934        1.259310
9293       True      False      0.780073        1.298951
11095     False      False      0.810818        1.241445
20099     False      False      0.801129        1.266664
8924      False      False      0.873129        1.161999

[16346 rows x 15 columns]
```

[29]:
```python
from sklearn.ensemble import RandomForestRegressor

forest = RandomForestRegressor()

forest.fit(x_train_s,y_train)
```

[29]:
```
RandomForestRegressor()
```

[30]:
```python
forest.score(x_test_s, y_test)
```

[30]:
```
0.8177708117987288
```

[31]:
```python
from sklearn.model_selection import GridSearchCV

forest = RandomForestRegressor()
param_grid ={
    "n_estimators": [3,10,30],
    "max_features": [2,4,6,8],
}

grid_search = GridSearchCV(forest, param_grid, cv=5,
                          scoring="neg_mean_squared_error",
                          return_train_score=True)

grid_search.fit(x_train_s, y_train)
```

[31]:
```
GridSearchCV(cv=5, estimator=RandomForestRegressor(),
             param_grid={'max_features': [2, 4, 6, 8],
                         'n_estimators': [3, 10, 30]},
             return_train_score=True, scoring='neg_mean_squared_error')
```

[32]:
```python
best_forest = grid_search.best_estimator_
best_forest
```

[32]:
```
RandomForestRegressor(max_features=8, n_estimators=30)
```

[33]:
```python
best_forest.score(x_test_s, y_test)
```

[33]: 0.8155366846753866

[34]:
```
import statsmodels.api as sm
import seaborn as sns
sns.set()

x_test_s
```

[34]:
```
array([[-1.4729364 ,  1.0647998 ,  0.10908972, …, -0.38215469,
          0.01059181, -0.13032222],
       [ 0.75408919, -0.81637233, -0.20845646, …, -0.38215469,
         -0.58954348, -0.01692021],
       [-1.35808978,  1.01800448,  1.37927447, …, -0.38215469,
         -4.49708037, -0.61863355],
       …,
       [-0.96860997,  1.34557176,  0.34724936, …, -0.38215469,
         -0.79844496,  0.46825826],
       [ 0.63424925, -0.79297467,  0.585409  , …, -0.38215469,
          0.85926029, -0.52184048],
       [-1.20828986,  1.092877  ,  0.10908972, …, -0.38215469,
         -0.14162882,  0.08168569]])
```

[35]:
```
def process_and_predict(input_data):


    input_data['bedroom_coeff'] = input_data['total_bedrooms'] /␣
 ↪input_data['total_rooms']
    input_data['household_rooms'] = input_data['total_rooms'] /␣
 ↪input_data['households']

    input_data_s = scaler.transform(input_data)

    #linear reg
    predict_lr = reg.predict(input_data_s)

    #Random forest
    predict_rf = reg.predict(input_data_s)

    return predict_lr, predict_rf
```

0.5 Price prediction

[36]:
```
new_input_data = pd.DataFrame({
    'longitude': [-122.23],
    'latitude': [37.88],
    'housing_median_age': [25],
    'total_rooms': [7000],
```

```
    'total_bedrooms': [1100],
    'population': [2400],
    'households': [1130],
    'median_income': [8.0000],
    '<1H OCEAN': False,
    'INLAND': False,
    'ISLAND': False,
    'NEAR BAY': True,
    'NEAR OCEAN': False

})

predict_lr, predict_rf = process_and_predict(new_input_data)
predict_lr
```

[36]:
```
array([-6.43324079e+08])
```

0.6 Result

[37]:
```
predict_rf
```

[37]:
```
array([-6.43324079e+08])
```

[38]:
```
data
```

[38]:
```
       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  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
…             …           …            …                   …
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
```

```
20637      1007.0       433.0         1.7000              92300.0
20638       741.0       349.0         1.8672              84700.0
20639      1387.0       530.0         2.3886              89400.0

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
…                 …
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20433 rows x 10 columns]
```

```
[ ]:
```