In [1]:

```
# 機械学習　p.58～64

# 2.4 データを研究、可視化して理解を深める
# 研究セットの抽出（今回は省略…データ数少ない）
```

In [2]:

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

from sklearn.model_selection import StratifiedShuffleSplit

housing = pd.read_csv("datasets/housing/housing.csv")

housing["income_cat"] = pd.cut(housing["median_income"],
                               bins=[0.0, 1.5, 3.0, 4.5, 6.0,np.inf], labels=[1, 2, 3, 4, 5])
split = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)

for train_index, test_index in split.split(housing, housing["income_cat"]):
    strat_train_set = housing.loc[train_index]
    strat_test_set = housing.loc[test_index]

for set_ in (strat_train_set, strat_test_set):
    set_.drop("income_cat", axis=1, inplace=True)

housing = strat_train_set.drop("median_house_value", axis=1)
housing_labels = strat_train_set["median_house_value"].copy()
```

In [3]:

```
# longitude latitude housing_median_age total_rooms total_bedrooms households median_income median_house_value ocean_proximity
# 経度      緯度     築年数の中央値      部屋数       寝室数         世帯数     収入の中央値  住宅価格の中央値    海との位置関係
```

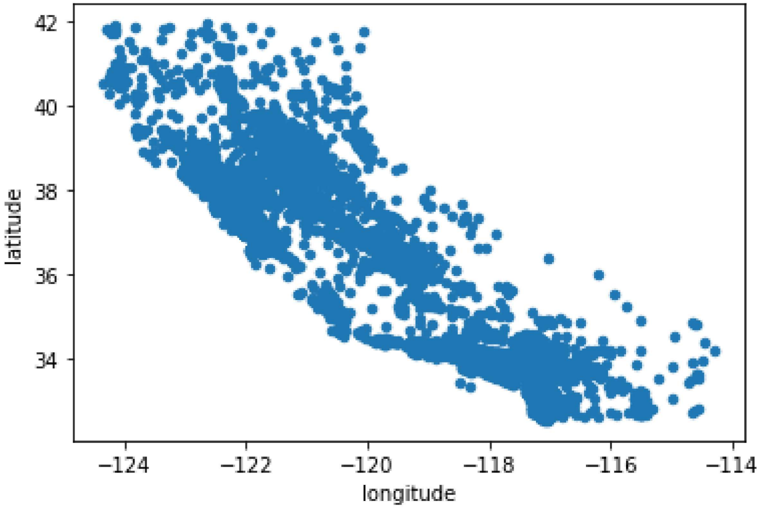
In [4]:

```
# 今回のプログラム～

# 地理情報（軽度、緯度）による
housing = strat_train_set.copy()
housing.plot(kind="scatter", x="longitude", y="latitude")
```

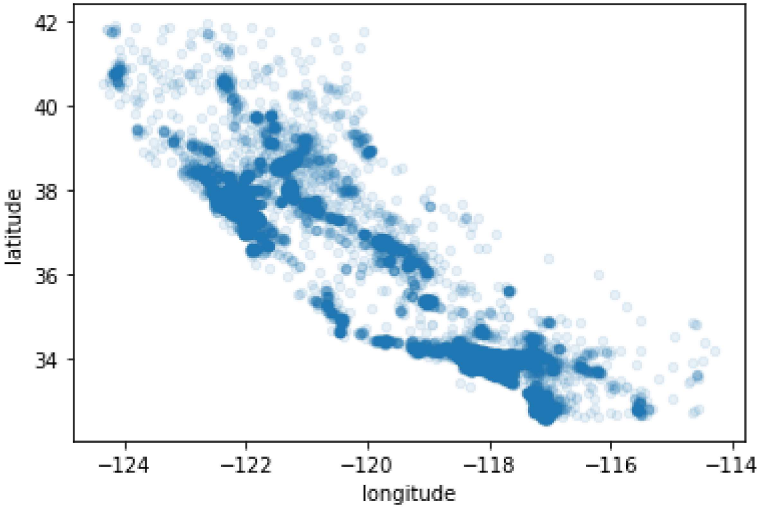
Out[4]:

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



In [5]:
```
# 密度の濃淡表示
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
```

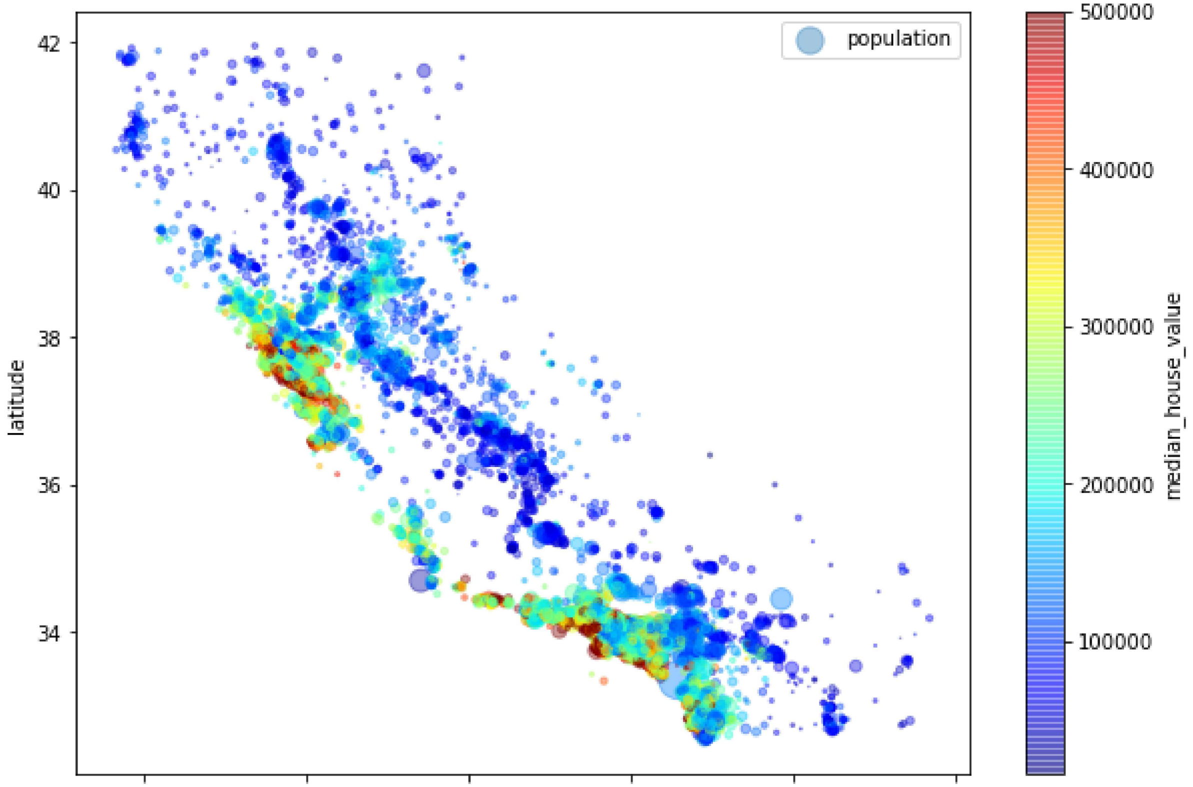
Out[5]:
```
<AxesSubplot:xlabel='longitude', ylabel='latitude'>
```



In [6]:
```
# 住宅価格：色、人口：円の面積表示
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4,
             s=housing["population"]/100, label="population", figsize=(10,7),
             c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True
)
plt.legend()
```

Out[6]:

```
<matplotlib.legend.Legend at 0x2186cba7040>
```



In [7]:

```
# 標準相関係数・ピアソンの r
corr_matrix = housing.corr()
corr_matrix["median_house_value"].sort_values(ascending=False)
```

Out[7]:

```
median_house_value    1.000000
median_income         0.687160
total_rooms           0.135097
housing_median_age    0.114110
households            0.064506
total_bedrooms        0.047689
population           -0.026920
longitude            -0.047432
latitude             -0.142724
Name: median_house_value, dtype: float64
```

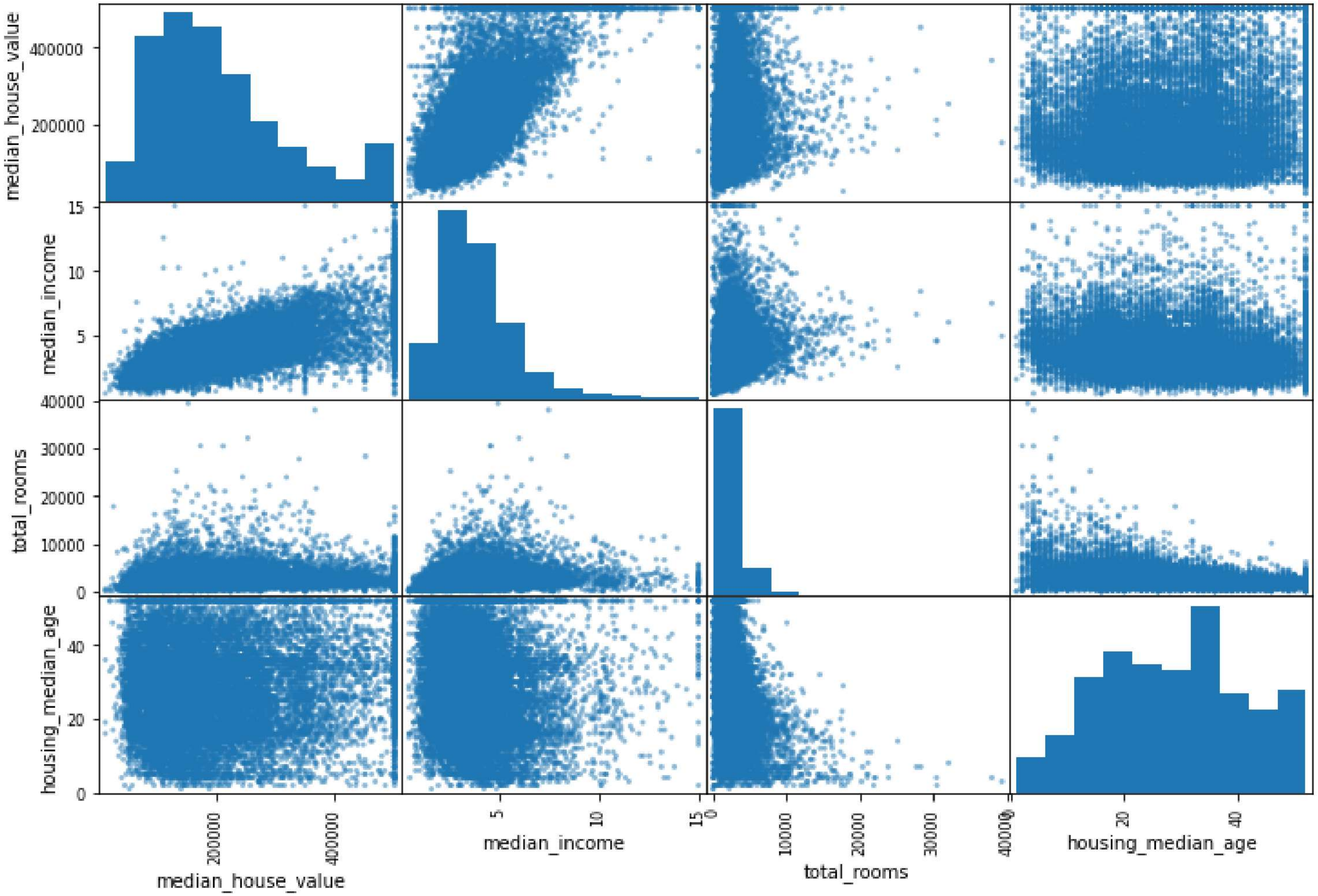
In [8]:

```
# Pandas による要素間の相互相関
from pandas.plotting import scatter_matrix

attributes =["median_house_value", "median_income", "total_rooms", "housing_median_age"]
scatter_matrix(housing[attributes], figsize=(12,8))
```

Out[8]:

```
array([[<AxesSubplot:xlabel='median_house_value', ylabel='median_house_value'>,
        <AxesSubplot:xlabel='median_income', ylabel='median_house_value'>,
        <AxesSubplot:xlabel='total_rooms', ylabel='median_house_value'>,
        <AxesSubplot:xlabel='housing_median_age', ylabel='median_house_value'>],
       [<AxesSubplot:xlabel='median_house_value', ylabel='median_income'>,
        <AxesSubplot:xlabel='median_income', ylabel='median_income'>,
        <AxesSubplot:xlabel='total_rooms', ylabel='median_income'>,
        <AxesSubplot:xlabel='housing_median_age', ylabel='median_income'>],
       [<AxesSubplot:xlabel='median_house_value', ylabel='total_rooms'>,
        <AxesSubplot:xlabel='median_income', ylabel='total_rooms'>,
        <AxesSubplot:xlabel='total_rooms', ylabel='total_rooms'>,
        <AxesSubplot:xlabel='housing_median_age', ylabel='total_rooms'>],
       [<AxesSubplot:xlabel='median_house_value', ylabel='housing_median_age'>,
        <AxesSubplot:xlabel='median_income', ylabel='housing_median_age'>,
        <AxesSubplot:xlabel='total_rooms', ylabel='housing_median_age'>,
        <AxesSubplot:xlabel='housing_median_age', ylabel='housing_median_age'>]],
      dtype=object)
```

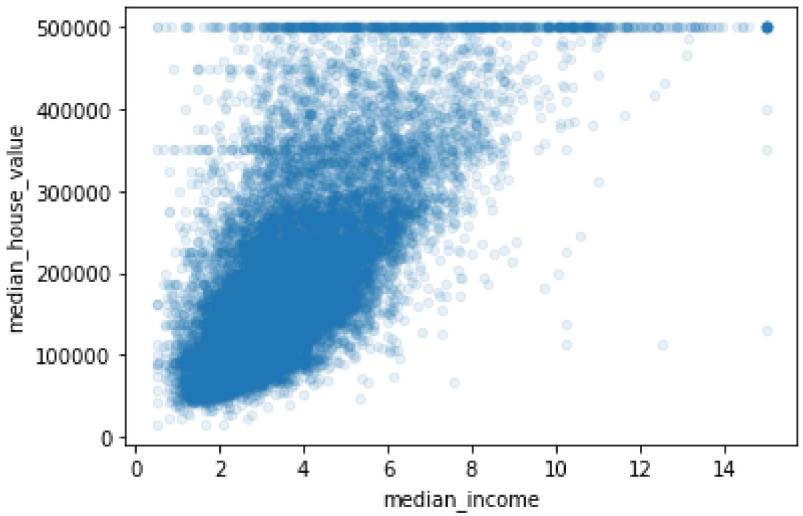


In [9]:

```
housing.plot(kind="scatter", x="median_income", y="median_house_value", alpha=0.1)
```

Out[9]:

```
<AxesSubplot:xlabel='median_income', ylabel='median_house_value'>
```



In [10]:

```
# 属性の組み合わせによる新項目の作成と相関
housing["rooms_per_household"] = housing["total_rooms"] / housing["households"]
housing["bedrooms_per_room"] = housing["total_bedrooms"] / housing["total_rooms"]
housing["population_per_household"] = housing["population"] / housing["households"]

corr_matrix = housing.corr()
corr_matrix["median_house_value"].sort_values(ascending=False)
```

Out[10]:

```
median_house_value           1.000000
median_income                0.687160
rooms_per_household          0.146285
total_rooms                  0.135097
housing_median_age           0.114110
households                   0.064506
total_bedrooms               0.047689
population_per_household    -0.021985
population                  -0.026920
longitude                   -0.047432
latitude                    -0.142724
bedrooms_per_room           -0.259984
Name: median_house_value, dtype: float64
```