In [1]:

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

In [2]:

```
bd=pd.read_csv('E:/housing.csv')
```

In [3]:

```
bd.head(3)
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_valu
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.

In [4]:

```
bd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
longitude             20640 non-null float64
latitude              20640 non-null float64
housing_median_age    20640 non-null float64
total_rooms           20640 non-null float64
total_bedrooms        20433 non-null float64
population            20640 non-null float64
households            20640 non-null float64
median_income         20640 non-null float64
median_house_value    20640 non-null float64
ocean_proximity       20640 non-null object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

In [5]:

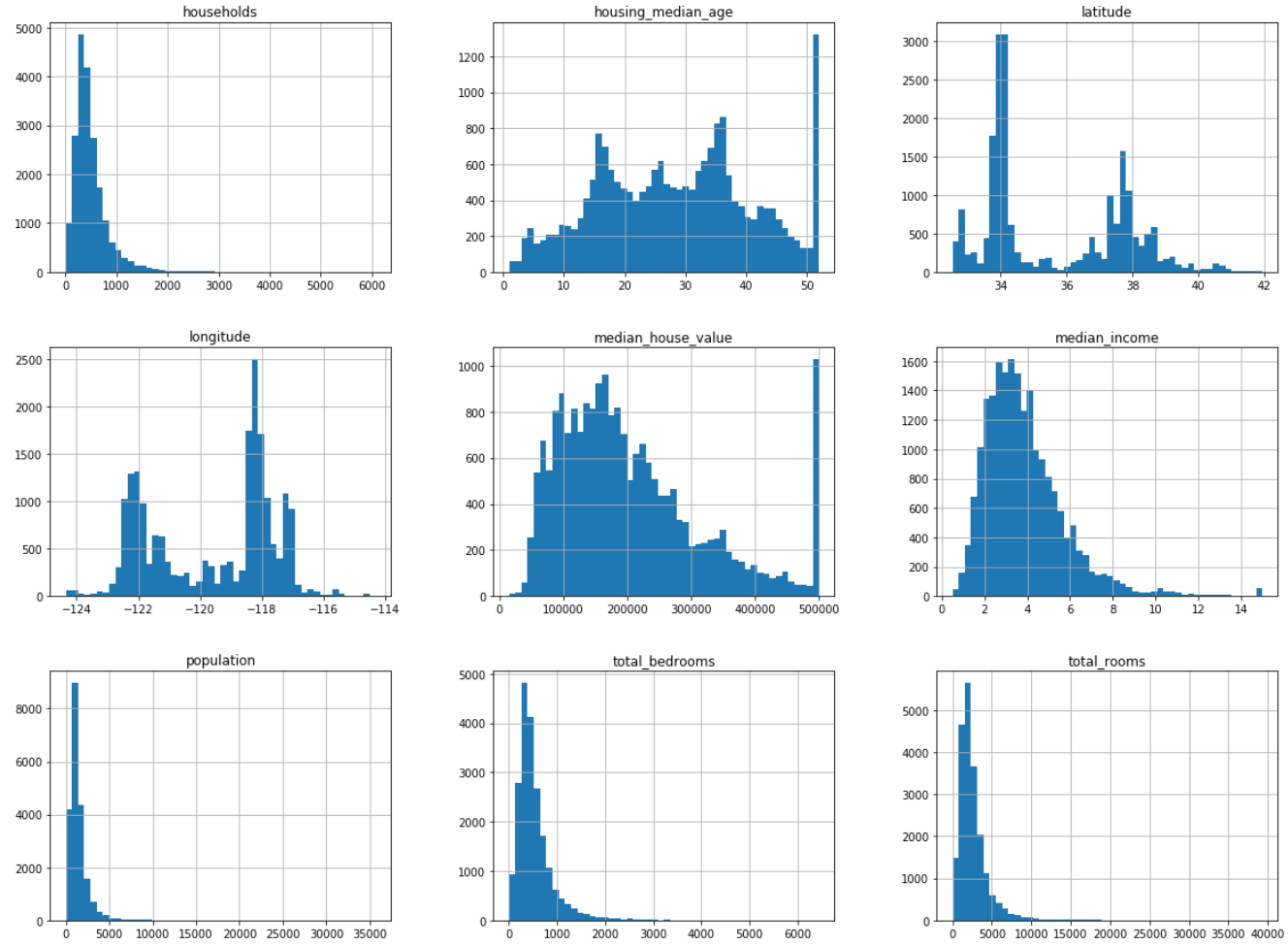
```
bd.describe()
```

Out[5]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100

In [6]:

```
bd.hist(bins=50,figsize=(20,15))
plt.show()
```

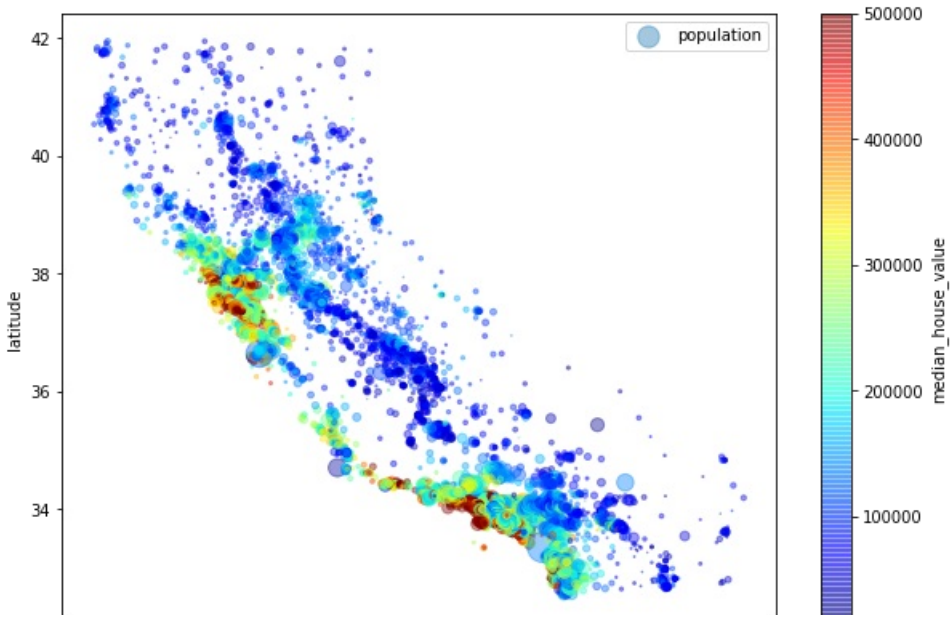


In [7]:

```
bd.plot(kind="scatter",x="longitude",y="latitude",alpha=0.4,
             s=bd["population"]/100,label="population",figsize=(10,7),
             c="median_house_value",cmap=plt.get_cmap("jet"),colorbar=True,
            )
plt.legend()
```

Out[7]:

```
<matplotlib.legend.Legend at 0x651823c508>
```



In [8]:

```
corr_matrix=bd.corr()
```

In [9]:

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

Out[9]:

```
median_house_value    1.000000
median_income         0.688075
total_rooms           0.134153
housing_median_age    0.105623
households            0.065843
total_bedrooms        0.049686
population           -0.024650
longitude            -0.045967
latitude             -0.144160
Name: median_house_value, dtype: float64
```

In [10]:

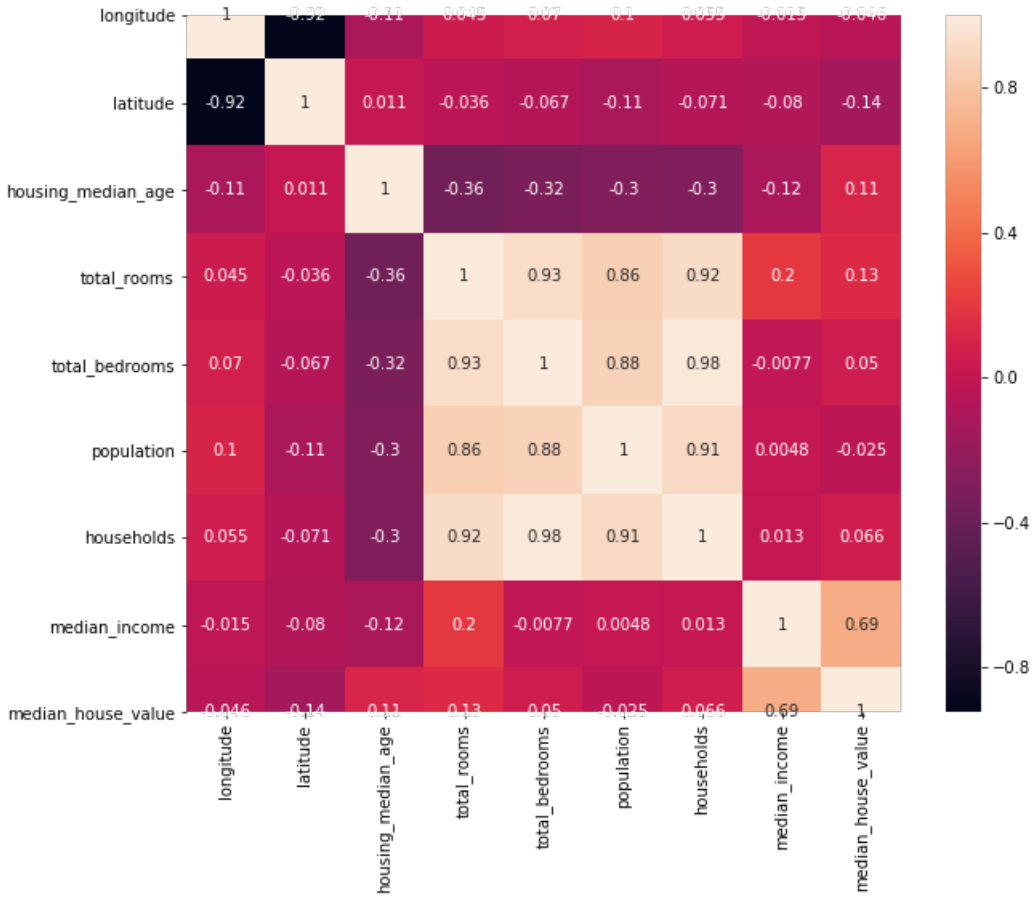
```
import seaborn as sns
```

In [11]:

```
plt.figure(figsize=(10,8))
sns.heatmap(bd.corr(), annot=True)
```

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x6517647788>
```

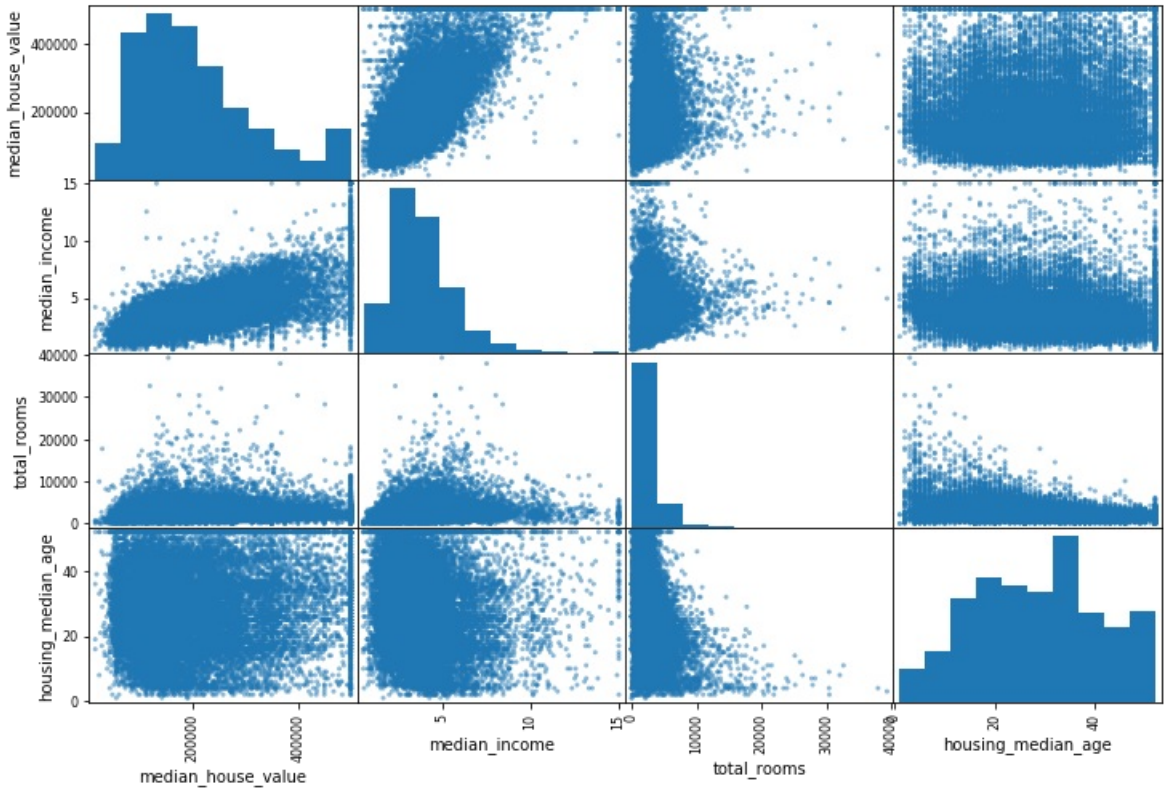


In [12]:

```
from pandas.plotting import scatter_matrix
attributes = ["median_house_value", "median_income","total_rooms","housing_median_age"]
scatter_matrix(bd[attributes],figsize=(12,8))
```

Out[12]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000065175D5848>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517692AC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517AD99C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517A921C8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x00000065176CEC08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000651789AD08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517B36E08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000065179A3F08>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000651798CB08>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000651B08DCC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517907288>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000006517550348>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000006517518488>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000065173CB588>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000065174A9E88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000065174E67C8>]],
      dtype=object)
```

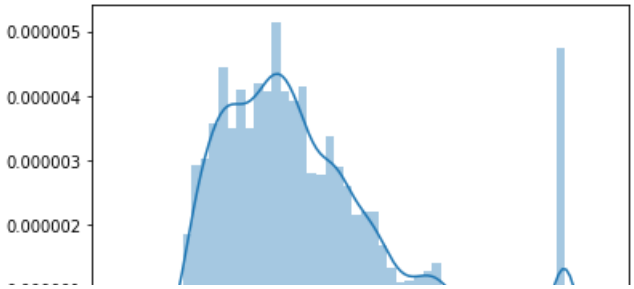


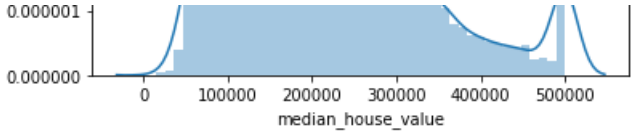
In [13]:

```
sns.distplot(bd.median_house_value)
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x65184b5a08>
```





In [14]:

```
bd.isnull().sum()
```

Out[14]:

```
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
```

In [15]:

```
df= bd.fillna(bd.mean())
```

In [16]:

```
df.isnull().sum()
```

Out[16]:

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

In [17]:

```
from sklearn.preprocessing import LabelEncoder
```

In [18]:

```
labelEncoder = LabelEncoder()
print(df["ocean_proximity"].value_counts())
df["ocean_proximity"] = labelEncoder.fit_transform(df["ocean_proximity"])
df["ocean_proximity"].value_counts()
df.describe()
```

```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

Out[18]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000

mean	-119.569704	35.631861	housing_median_age	total_rooms	total_bedrooms	population	households	median_income
std	2.003532	2.135952	12.585558	2181.615252	419.266592	1132.462122	382.329753	1.899822
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900
25%	-121.800000	33.930000	18.000000	1447.750000	297.000000	787.000000	280.000000	2.563400
50%	-118.490000	34.260000	29.000000	2127.000000	438.000000	1166.000000	409.000000	3.534800
75%	-118.010000	37.710000	37.000000	3148.000000	643.250000	1725.000000	605.000000	4.743250
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100

In [19]:

```
feature = df.drop(['median_house_value'], axis=1)
label = df.median_house_value
```

In [20]:

```
from sklearn.model_selection import train_test_split
feature_train, feature_test,label_train, label_test = train_test_split(feature,label, test_size=0.2
, random_state=19)
```

In [21]:

```
from sklearn import linear_model
from sklearn.metrics import r2_score, mean_squared_error
```

In [22]:

```
linear_reg = linear_model.LinearRegression()
linear_reg.fit(feature_train,label_train)
r2_score(linear_reg.predict(feature_train),label_train)
```

Out[22]:

```
0.433069087226281
```

In [23]:

```
linear_reg.predict(feature_test)
```

Out[23]:

```
array([266002.09754319,  72873.18180857, 208580.50204955, ...,
       192221.93798113, 174373.58529118, 267434.7920918 ])
```

In [ ]: