In [1]:
```
pwd
```

Out[1]:
```
'/home/ajith'
```

In [23]:
```
import os
import tarfile
from six.moves import urllib
DOWNLOAD_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml/master/"
HOUSING_PATH = os.path.join("datasets", "housing")
HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"
def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PATH):
 if not os.path.isdir(housing_path):
    os.makedirs(housing_path)
    tgz_path = os.path.join(housing_path, "housing.tgz")
    urllib.request.urlretrieve(housing_url, tgz_path)
    housing_tgz = tarfile.open(tgz_path)
    housing_tgz.extractall(path=housing_path)
    housing_tgz.close()

fetch_housing_data()

import pandas as pd
def load_housing_data(housing_path=HOUSING_PATH):
     csv_path = os.path.join(housing_path, "housing.csv")
     return pd.read_csv(csv_path)
```

In [27]:
```
housing = load_housing_data()
housing.head()
```

Out[27]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	ho
0	-122.23	37.88	41.0	880.0	129.0	322.0	12(
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	113

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	ho
2	-122.24	37.85	52.0	1467.0	190.0	496.0	17
3	-122.25	37.85	52.0	1274.0	235.0	558.0	21
4	-122.25	37.85	52.0	1627.0	280.0	565.0	25

In [28]:
```
housing.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
```

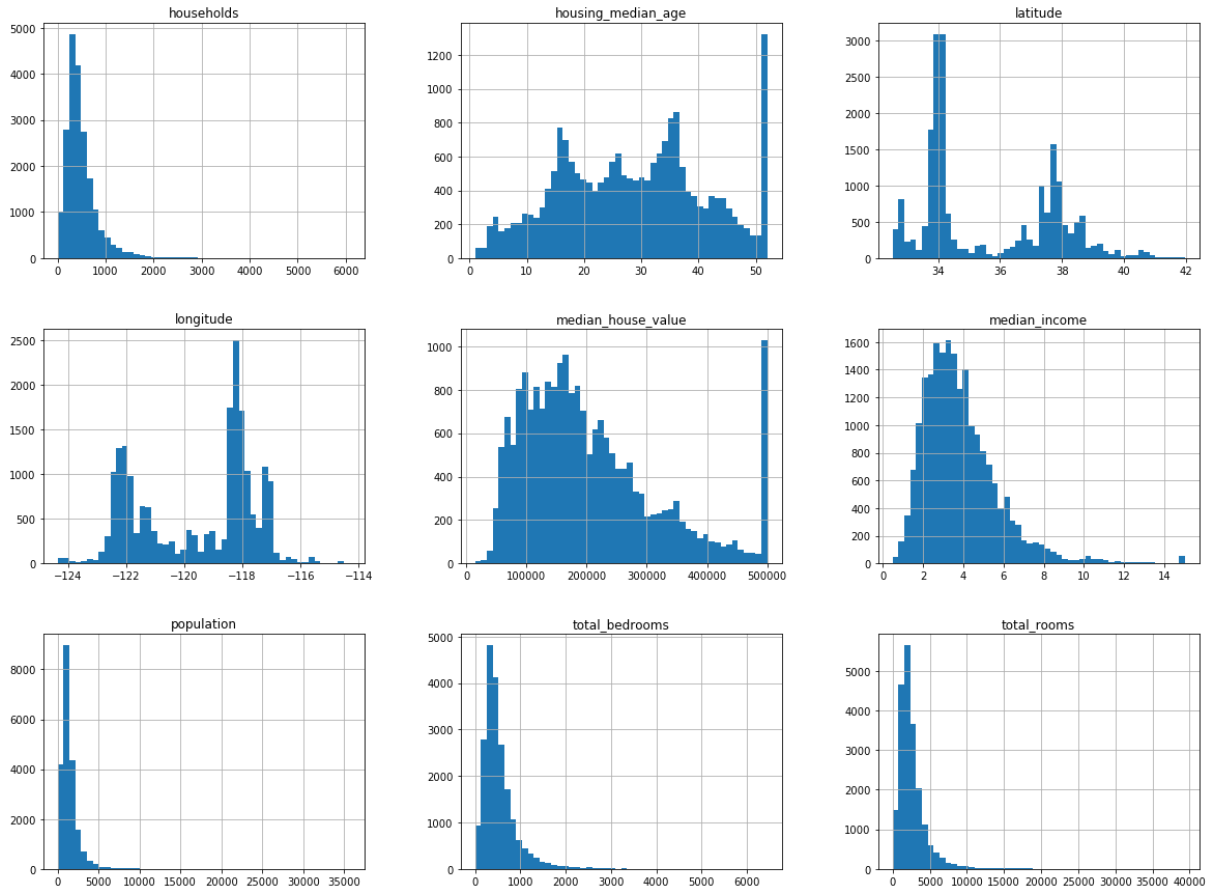
In [30]:
```
housing.describe()
```

Out[30]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553
std	2.003532	2.135952	12.585558	2181.615252	421.385070
min	-124.350000	32.540000	1.000000	2.000000	1.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000

In [31]:
```
%matplotlib inline
# only in a Jupyter notebook
import matplotlib.pyplot as plt
housing.hist(bins=50, figsize=(20,15))
plt.show()
```



In [32]:
```
import os
os.environ['PATH']
```

Out[32]:
```
'/home/ajith/anaconda3/bin:/home/ajith/anaconda3/condabin:/home/ajith/anaconda3/bin:/home/ajith/.rvm/gems/ruby-2.5.1/bin:/home/ajith/.rvm/gems/ruby-2.5.1@global/bin:/home/ajith/.rvm/rubies/ruby-2.5.1/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/ajith/.rvm/bin:/home/ajith/.rvm/bin'
```

In [5]:
```
import numpy as np
def split_train_test(data, test_ratio):
```

```
    shuffled_indices = np.random.permutation(len(data))
    test_set_size = int(len(data) * test_ratio)
    test_indices = shuffled_indices[:test_set_size]
    train_indices = shuffled_indices[test_set_size:]
    return data.iloc[train_indices], data.iloc[test_indices]
```

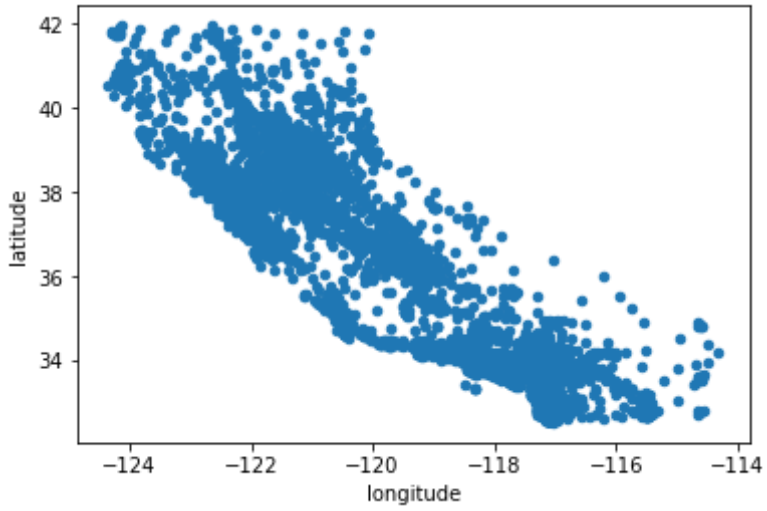
In [17]:
```
import hashlib
def test_set_check(identifier, test_ratio, hash):
     return hash(np.int64(identifier)).digest()[-1] < 256 * test_ratio
def split_train_test_by_id(data, test_ratio, id_column, hash=hashlib.md
5):
    ids = data[id_column]
    in_test_set = ids.apply(lambda id_: test_set_check(id_, test_ratio,
 hash))
    return data.loc[~in_test_set], data.loc[in_test_set]
    housing = strat_train_set.copy()
    housing.plot(kind="scatter", x="longitude", y="latitude")
```

In [19]:
```
import hashlib
def test_set_check(identifier, test_ratio, hash):
     return hash(np.int64(identifier)).digest()[-1] < 256 * test_ratio
def split_train_test_by_id(data, test_ratio, id_column, hash=hashlib.md
5):
    ids = data[id_column]
    in_test_set = ids.apply(lambda id_: test_set_check(id_, test_ratio,
 hash))
    return data.loc[~in_test_set], data.loc[in_test_set]
    housing = strat_train_set.copy()
    housing.plot(kind="scatter", x="longitude", y="latitude")
```

In [29]:
```
import os
import tarfile
from six.moves import urllib
DOWNLOAD_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml/ma
ster/"
HOUSING_PATH = os.path.join("datasets", "housing")
HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"
def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PA
```

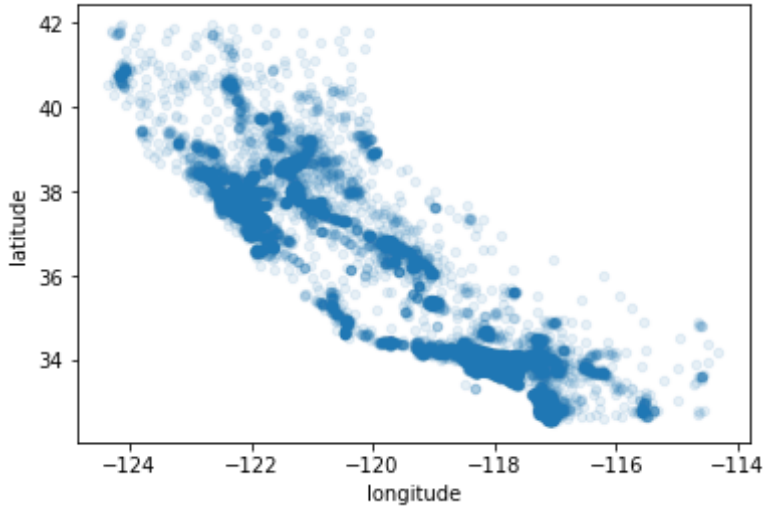
```
TH):
 if not os.path.isdir(housing_path):
    os.makedirs(housing_path)
    tgz_path = os.path.join(housing_path, "housing.tgz")
    urllib.request.urlretrieve(housing_url, tgz_path)
    housing_tgz = tarfile.open(tgz_path)
    housing_tgz.extractall(path=housing_path)
    housing_tgz.close()

fetch_housing_data()
import pandas as pd
def load_housing_data(housing_path=HOUSING_PATH):
     csv_path = os.path.join(housing_path, "housing.csv")
     return pd.read_csv(csv_path)
%matplotlib inline
# only in a Jupyter notebook

housing = load_housing_data()
housing.plot(kind="scatter", x="longitude", y="latitude")
```

Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7feb1838d588>



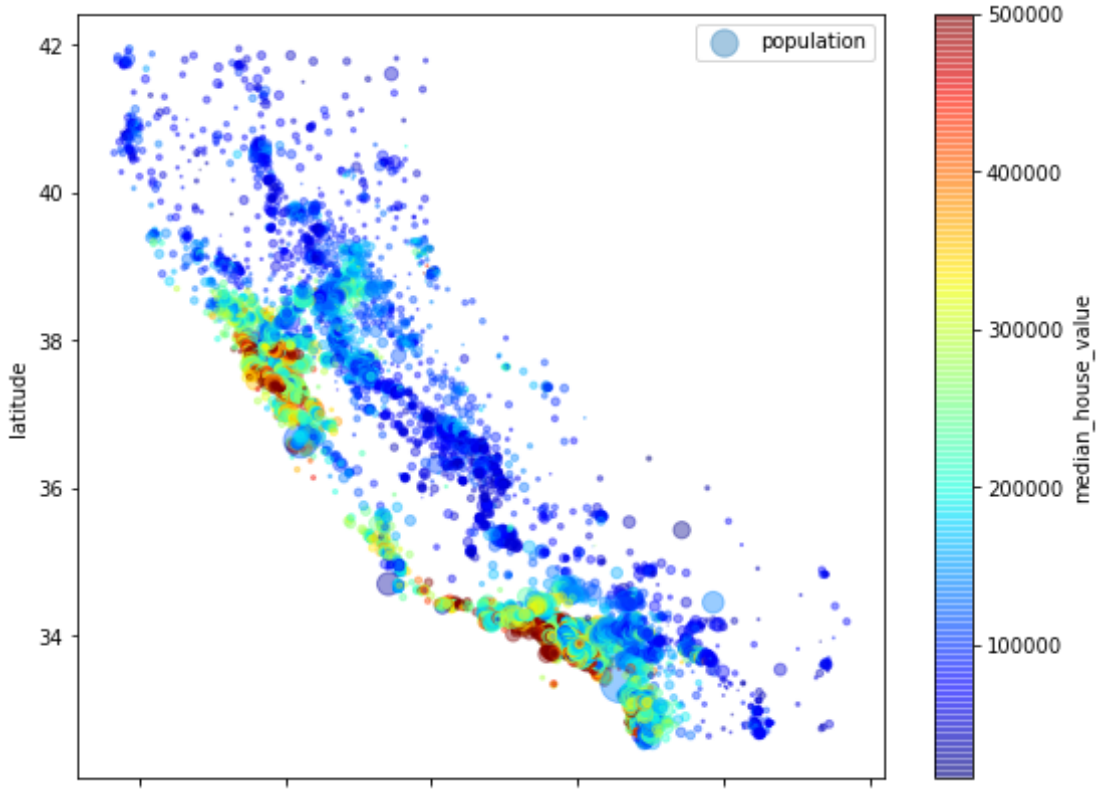
In [30]:
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.1)
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7feb1a6e6ac8>



In [32]:
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0.4,
s=housing["population"]/100, label="population", figsize=(9,7),
c="median_house_value", cmap=plt.get_cmap("jet"), colorbar=True,
)
plt.legend()
```

Out[32]: <matplotlib.legend.Legend at 0x7feb1ad58f98>

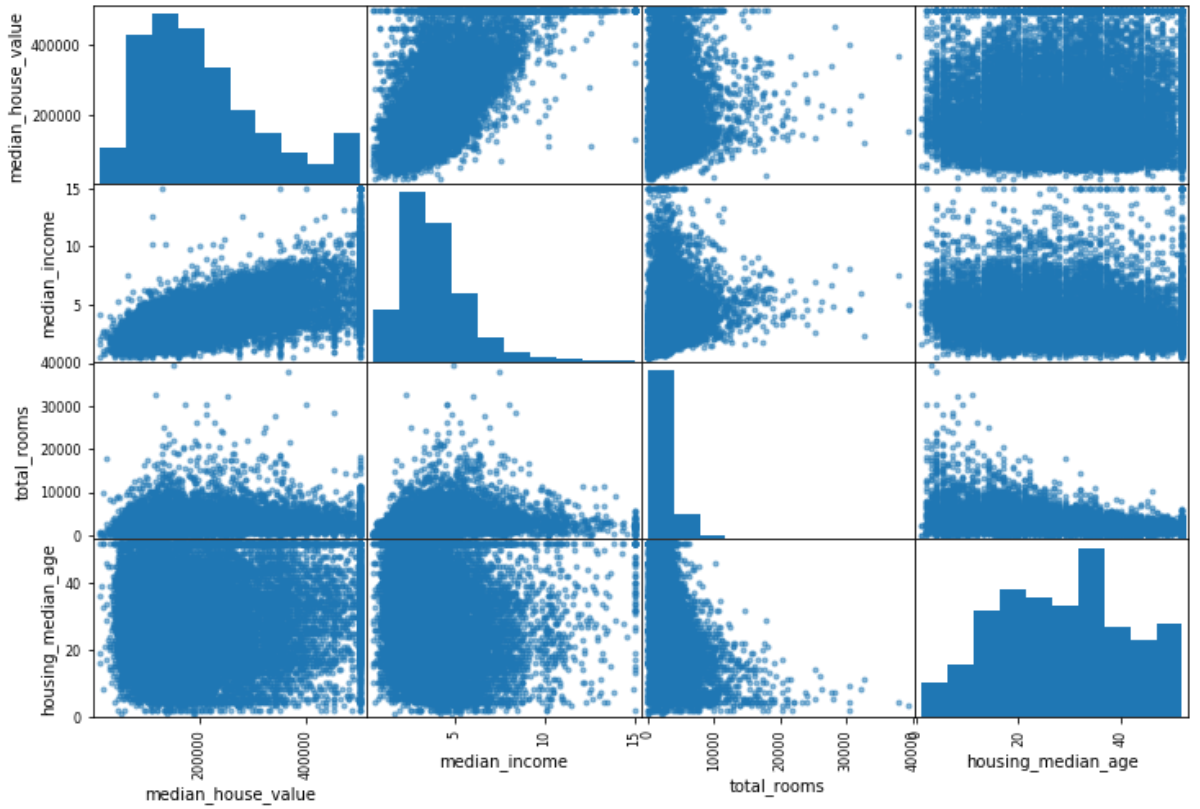


In [36]:
```
corr_matrix = housing.corr()
corr_matrix["total_rooms"].sort_values(ascending=False)
```

Out[36]:
```
total_rooms           1.000000
total_bedrooms        0.930380
households            0.918484
population            0.857126
median_income         0.198050
median_house_value    0.134153
longitude             0.044568
latitude             -0.036100
housing_median_age   -0.361262
Name: total_rooms, dtype: float64
```

In [37]:
```
from pandas.tools.plotting import scatter_matrix
attributes = ["median_house_value", "median_income", "total_rooms",
"housing_median_age"]
scatter_matrix(housing[attributes], figsize=(12, 8))
```

Out[37]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7feb172ec6a0
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb181cceb8
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb181aca90
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17b091d0
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7feb17c44668
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17c44208
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17d5cfd0
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17cc5a20
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7feb183e8ac8
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17de97f0
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17d37908
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17ee62b0
>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7feb17e5d828
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17fda470
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17f48b70
>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7feb17f670f0
>]],
      dtype=object)
```



In [39]:
```
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"]
```

In [ ]:
```

```