HW #1

In [89]:

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

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

Q4

1. Data Information

In [14]:

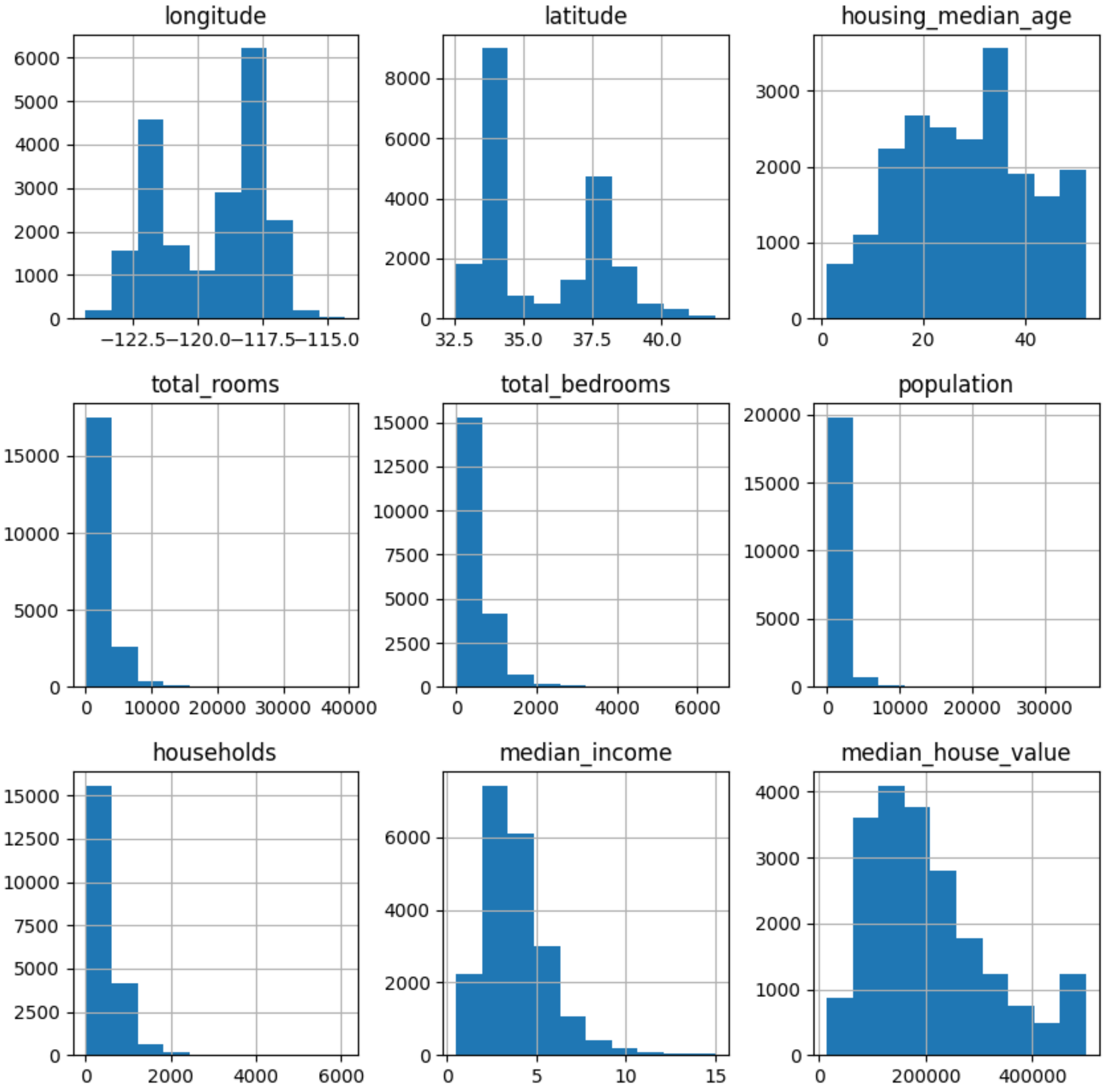
```
print('Data Head\n')
head = data.head()
print(head)
print('\nData Info\n')
info = data.info()
print('\nData Description\n')
describe = data.describe()
print(describe)
```

```
Data Head

   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   

   population  households  median_income  median_house_value ocean_proximi
ty
0       322.0       126.0         8.3252            452600.0        NEAR B
AY
1      2401.0      1138.0         8.3014            358500.0        NEAR B
AY
2       496.0       177.0         7.2574            352100.0        NEAR B
AY
3       558.0       219.0         5.6431            341300.0        NEAR B
AY
4       565.0       259.0         3.8462            342200.0        NEAR B
AY

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 Description

          longitude      latitude  housing_median_age   total_rooms  \
count  20640.000000  20640.000000        20640.000000  20640.000000   
mean    -119.569704     35.631861           28.639486   2635.763081   
std        2.003532      2.135952           12.585558   2181.615252   
min     -124.350000     32.540000            1.000000      2.000000   
25%     -121.800000     33.930000           18.000000   1447.750000   
50%     -118.490000     34.260000           29.000000   2127.000000   
75%     -118.010000     37.710000           37.000000   3148.000000   
max     -114.310000     41.950000           52.000000  39320.000000   

       total_bedrooms    population    households  median_income  \
count    20433.000000  20640.000000  20640.000000   20640.000000   
mean       537.870553   1425.476744    499.539680       3.870671   
std        421.385070   1132.462122    382.329753       1.899822   
min          1.000000      3.000000      1.000000       0.499900   
25%        296.000000    787.000000    280.000000       2.563400   
```

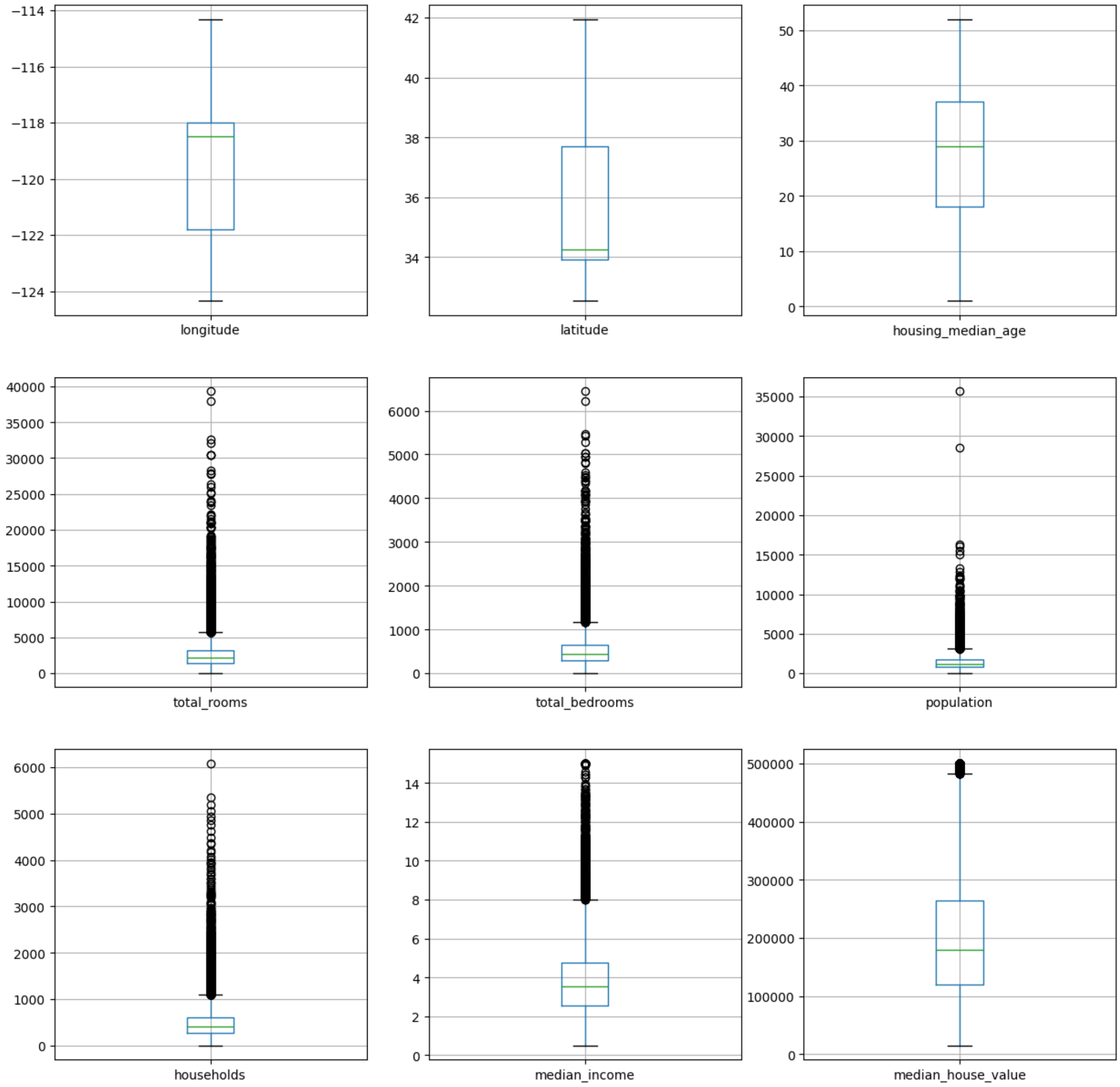
```
50%        435.000000   1166.000000    409.000000        3.534800
75%        647.000000   1725.000000    605.000000        4.743250
max       6445.000000  35682.000000   6082.000000       15.000100

       median_house_value
count        20640.000000
mean        206855.816909
std         115395.615874
min          14999.000000
25%         119600.000000
50%         179700.000000
75%         264725.000000
max         500001.000000
```

2. Visualize the Data

In [147…
```
data.hist(figsize=(10, 10))
plt.show()
```



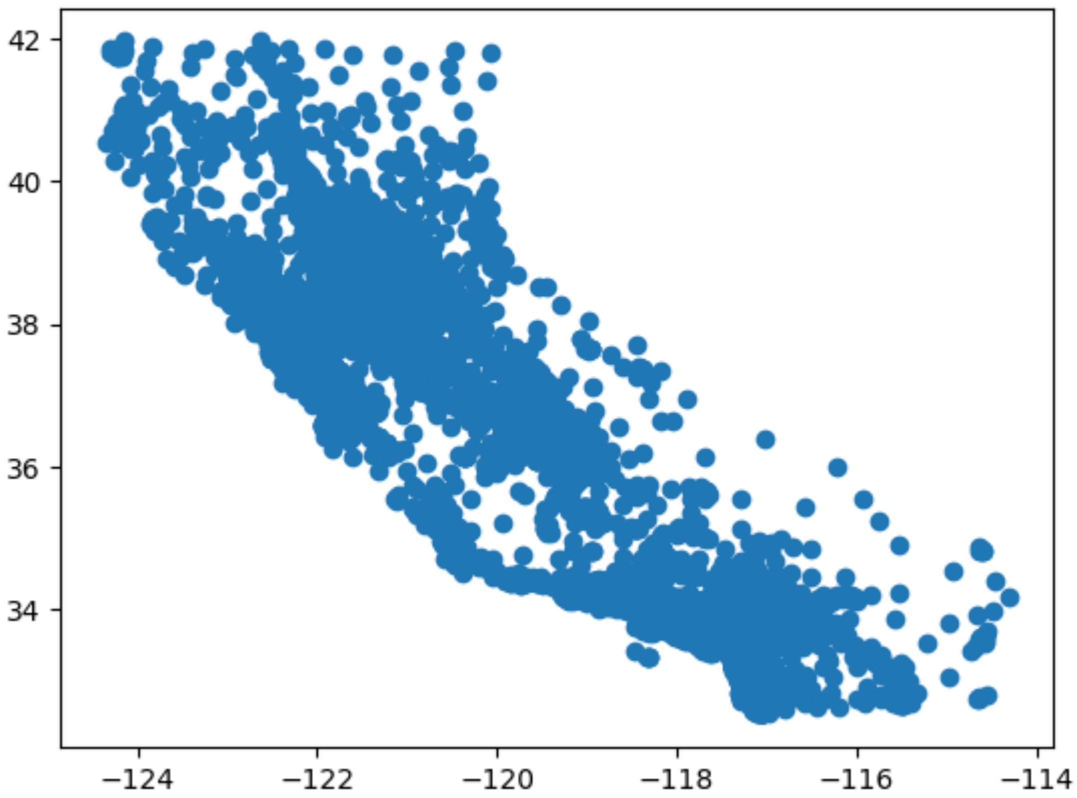
In [81]:
```
# Remove ocean_proxmitity
numerical_cols = data.columns.delete(9)
fig, axes = plt.subplots(3,3, figsize=(15, 15))
for index, col_name in enumerate(numerical_cols):
    row = index % 3
    col = int(index / 3)
    data.boxplot(col_name, ax=axes[col, row])
```



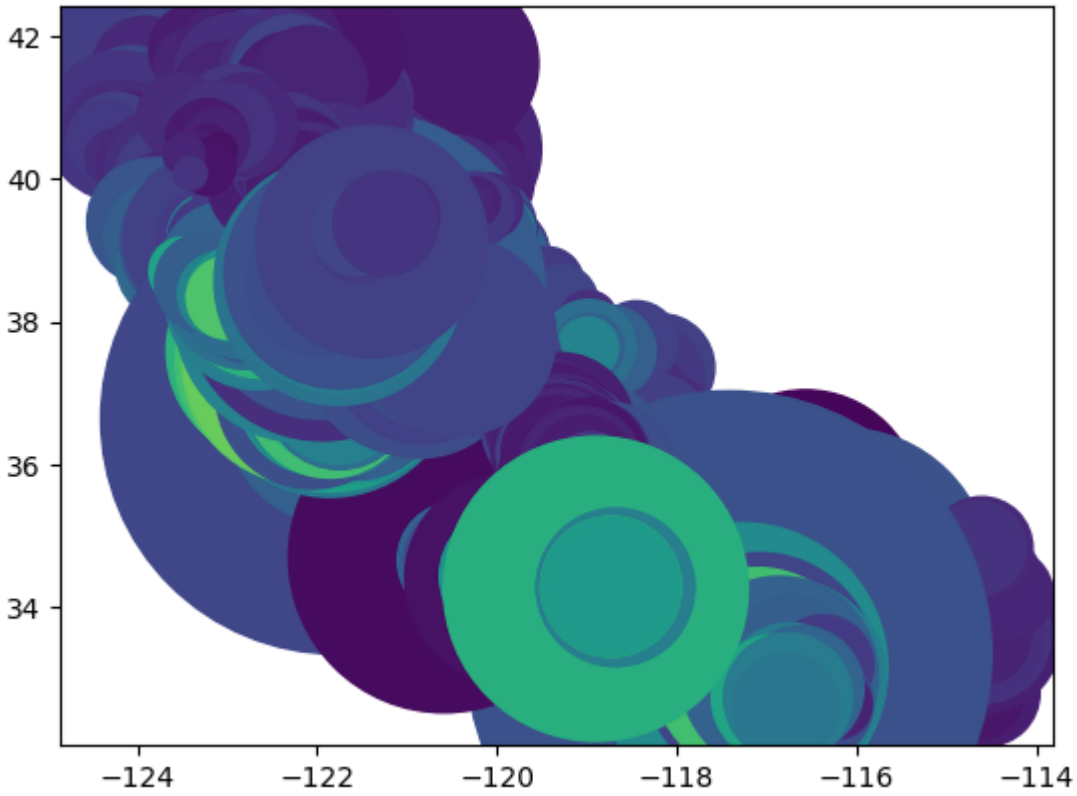
3. Longitude and Latitude Scatterplots

In [86]:

```
plt.scatter(data['longitude'], data['latitude'])
plt.show()
plt.scatter(data['longitude'], data['latitude'],data['population'], data[
```



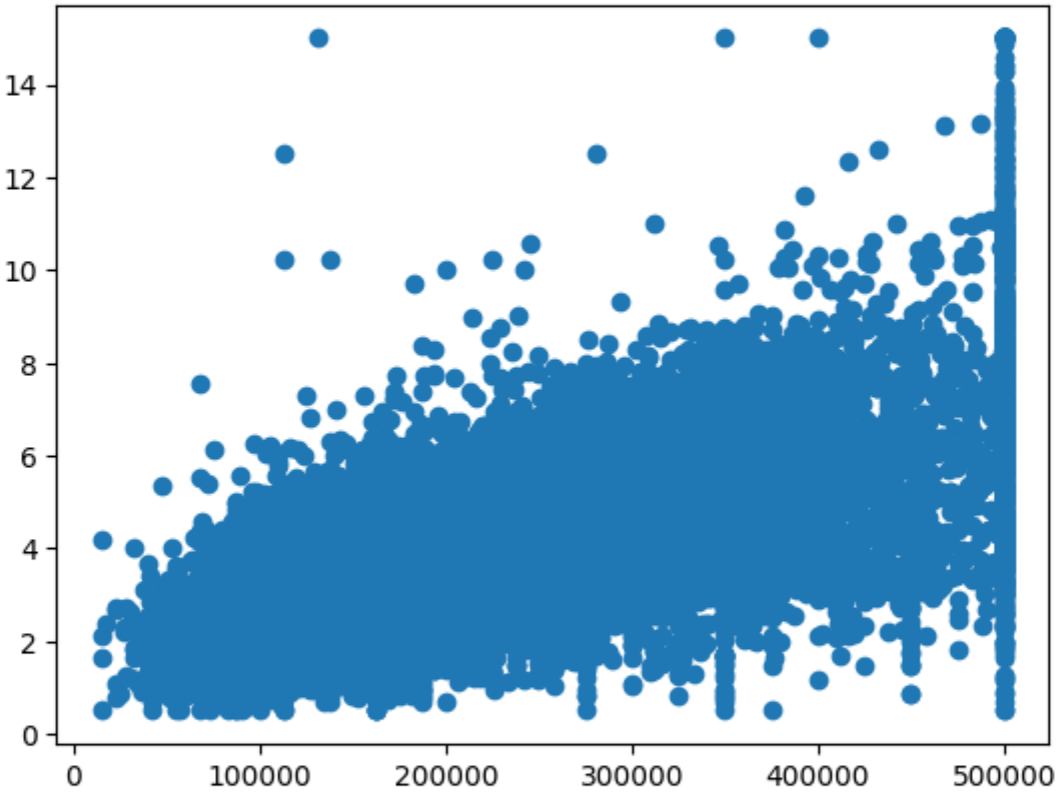
```
Out[86]: <matplotlib.collections.PathCollection at 0x78288d194f20>
```



4. Median House Price and Median Income Correlation

```
In [88]: plt.scatter(data['median_house_value'],data['median_income'])

         print(f'Correlation is {data['median_house_value'].corr(data['median_inco
```

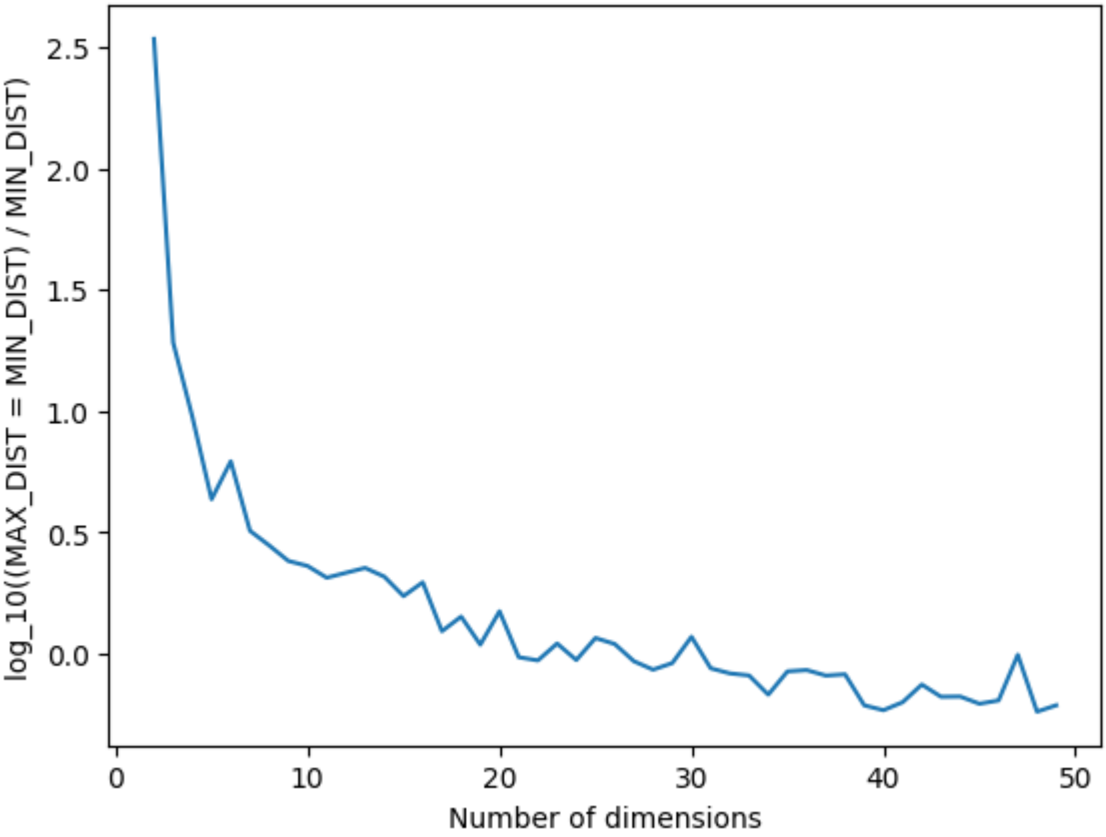
```
Correlation is 0.688075207958548
```



Q5

In [146…

```
import math
differences = []
MIN_DIM = 2
MAX_DIM = 50
SIZE = 500
DIM_RANGE = range(MIN_DIM, MAX_DIM)
for dim in DIM_RANGE:
    matrix_500_dim = np.random.rand(SIZE, dim)
    matrix_dim = np.random.rand(dim)
    diff = [np.linalg.norm(matrix_500_sliced - matrix_dim) for matrix_500

    max_diff = max(diff)
    min_diff = min(diff)
    log_diff = math.log10((max_diff-min_diff)/min_diff)
    differences.append(log_diff)
plt.plot(DIM_RANGE, differences)
plt.xlabel('Number of dimensions')
plt.ylabel('log_10((MAX_DIST = MIN_DIST) / MIN_DIST)')
plt.show()
```


log_10((MAX_DIST = MIN_DIST) / MIN_DIST)
2.5
2.0
1.5
1.0
0.5
0.0
0
10
20
30
40
50
Number of dimensions