INTRODUCTION OF THE PROJECT : In this California Dataset.They have provided with Longitude,latitude,total incomes and household as per the standards.This sample data is provided to get the idea about how people live thier lifes according to their median incomes.They also provided the overall population in data set.

This Dataset has around 20,640 observation in it with 10 columns and it's a mixed dataset containing categorical & numerical values.

In [1]:
```python
#importing necessary libraries..
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```python
df=pd.read_csv('housing.csv') #importing csv file
```

In [3]:
```python
df
```

Out[3]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	INLAND

20640 rows × 10 columns

In [4]:
```python
df.head(10)  #importing n number of starting data...
```

Out[4]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
5	-122.25	37.85	52	919	213.0	413	193	4.0368	269700	NEAR BAY
6	-122.25	37.84	52	2535	489.0	1094	514	3.6591	299200	NEAR BAY
7	-122.25	37.84	52	3104	687.0	1157	647	3.1200	241400	NEAR BAY
8	-122.26	37.84	42	2555	665.0	1206	595	2.0804	226700	NEAR BAY
9	-122.25	37.84	52	3549	707.0	1551	714	3.6912	261100	NEAR BAY

In [5]:
```python
df.shape #finding total number of rows and Columns...
```

Out[5]:
```
(20640, 10)
```

In [6]:
```python
df.info() #Complete Information about the data...
```

```
<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  int64  
 3   total_rooms         20640 non-null  int64  
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  int64  
 6   households          20640 non-null  int64  
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  int64  
 9   ocean_proximity     20640 non-null  object 
dtypes: float64(4), int64(5), object(1)
memory usage: 1.6+ MB
```

In [7]:
```python
df.describe()  #descriptive statistics...
```

Out[7]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

In [8]:
```python
df.isna().sum() #By this function. we can find a Null values in a dataset...
```

Out[8]:
```
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
```

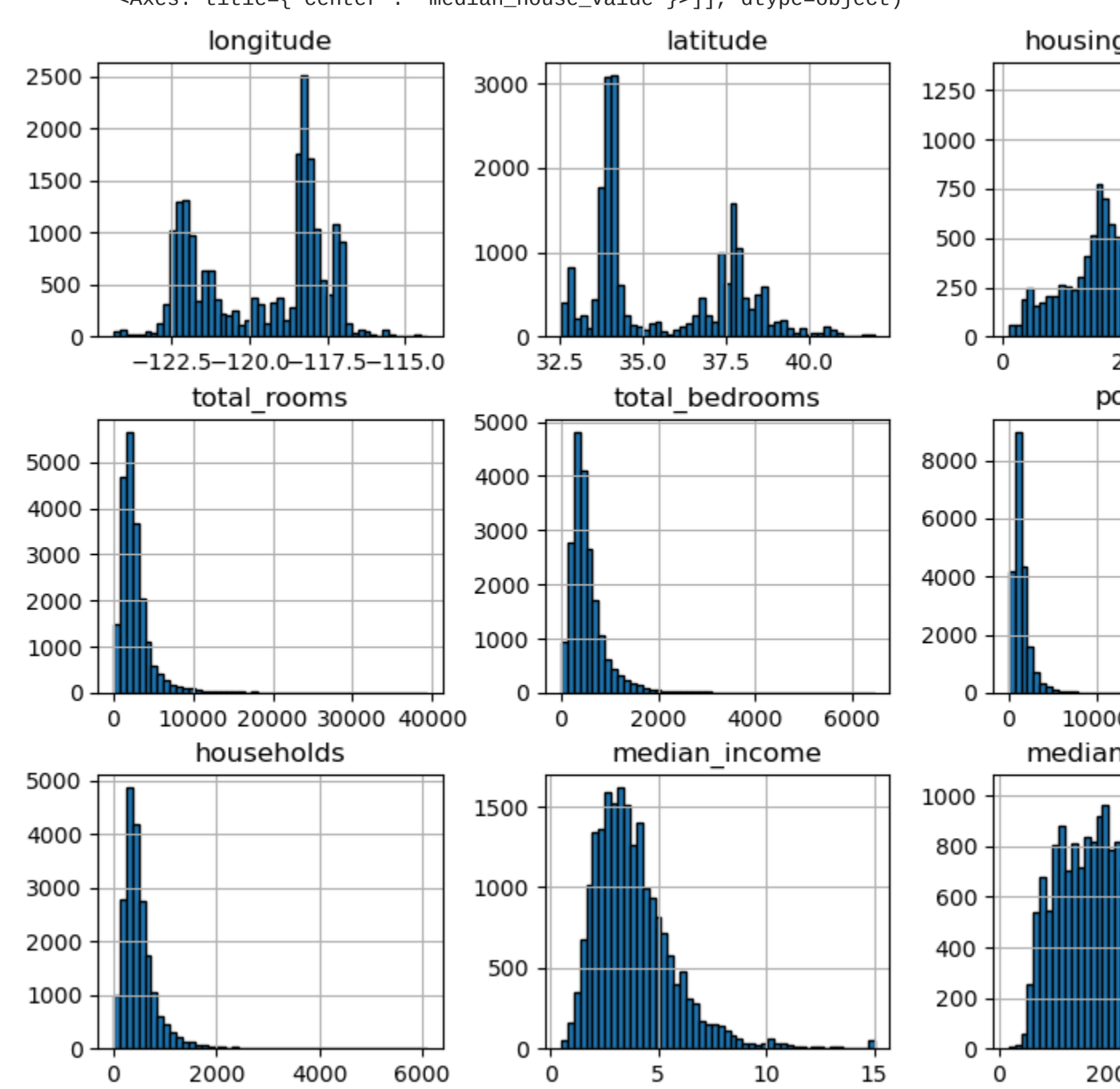
#1. What is the average median income of the data set and check the distribution of data using appropriate plots. Please explain the distribution of the plot

In [9]:
```python
df['median_income'].mean() #Finding average median income from the given Dataset..
```

Out[9]:
```
3.8706710029069766
```

In [10]:
```python
df.hist(edgecolor='black',bins=50,figsize=(10,8)) #creating the histogram plots for the dataset
```

Out[10]:
```
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)
```



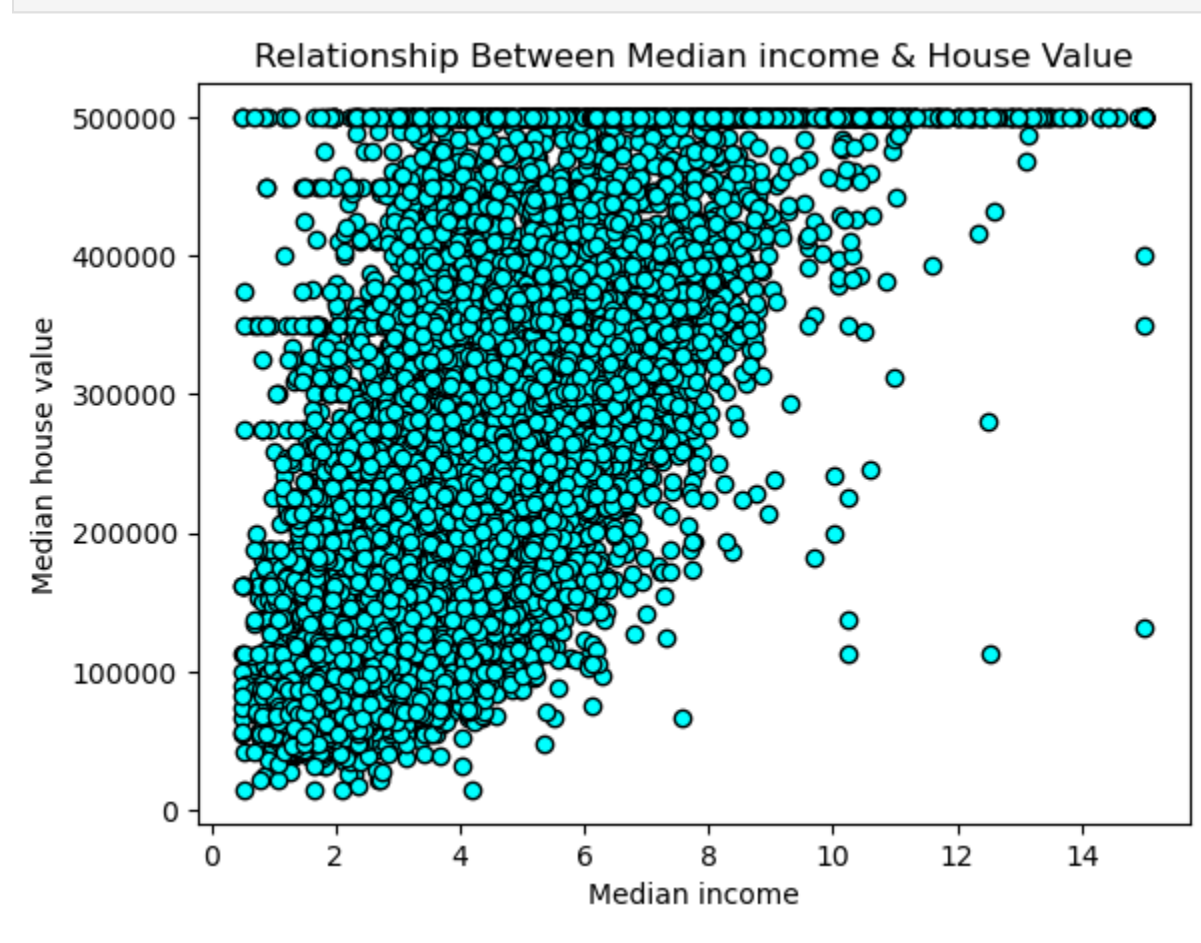
#2. Draw an appropriate plot to see the distribution of housing_median_age and explain your observations

In [11]:
```python
plt.hist(df.housing_median_age,facecolor='Yellow',edgecolor='black',bins=20) ##plotting histogram..
plt.xlabel('Median age') #labeling X-axis..
plt.ylabel('Frequency') #labeling Y-axis...
plt.title('Distribution of housing Median age')  #Giving title to the graph...
plt.show() #displaying the graph
```

Observation : in the above graph of housing median age.We can see that the people's who are in the age between 30-40 they have more income

#3. Show with the help of visualization, how median_income and median_house_values are related

In [12]:
```python
plt.scatter(df.median_income,df.median_house_value,edgecolor='black',facecolor='cyan') #Plotting  Scatter plot
plt.xlabel('Median income') #Labeling X-axis..
plt.ylabel('Median house value') #Labeling Y-axis..
plt.title('Relationship Between Median income & House Value') #Giving title to graph
plt.show() #Displaying graph
```



Observation : If the points generally move upwards from left to right, it suggests a positive correlation: areas with higher median incomes tend to have higher median house values.in our scenario as we can see that the graph is moving towards the left to right that means the people having higher incomes are moving towards to higher pricing houses.somewhat its showing a slight differnce inbetween median incomes and hosue value may be due to their own choices some people tend to locate at the same place.overall there is a positive correlation we can say

#4. Create a data set by deleting the corresponding examples from the data set for which total_bedrooms are not available

In [13]:
```python
df.isna().sum() #To count the number of null values..
```

Out[13]:
```
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 [14]:
```python
df1=df.dropna() #it will delete all the null rows...
df1
```

Out[14]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	INLAND

20433 rows × 10 columns

In [15]:
```python
df.dropna(subset=['total_bedrooms']) #Creation of New Data after removing Null Values from the data..
df1 # New data without null values..
```

Out[15]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	INLAND

20433 rows × 10 columns

#5. Create a data set by filling the missing data with the mean value of the total_bedrooms in the original data set

In [16]:
```python
df
```

Out[16]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	INLAND

20640 rows × 10 columns

In [17]:
```python
df2=df.fillna(value=df.mean()) #fillna function is used to fill the empty values of null..
df2
```

```
C:\Users\santo\AppData\Local\Temp\ipykernel_324\3772388735.py:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df2=df.fillna(value=df.mean())
```

Out[17]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	-121.09	39.48	25	1665	374.0	845	330	1.5603	78100	INLAND
20636	-121.21	39.49	18	697	150.0	356	114	2.5568	77100	INLAND
20637	-121.22	39.43	17	2254	485.0	1007	433	1.7000	92300	INLAND
20638	-121.32	39.43	18	1860	409.0	741	349	1.8672	84700	INLAND
20639	-121.24	39.37	16	2785	616.0	1387	530	2.3886	89400	INLAND

20640 rows × 10 columns

In [18]:
```python
df2.isna().sum() #isna function is used to find out null values..
```

Out[18]:
```
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
```

#6. Write a programming construct (create a user defined function) to calculate the median value of the data set wherever required.

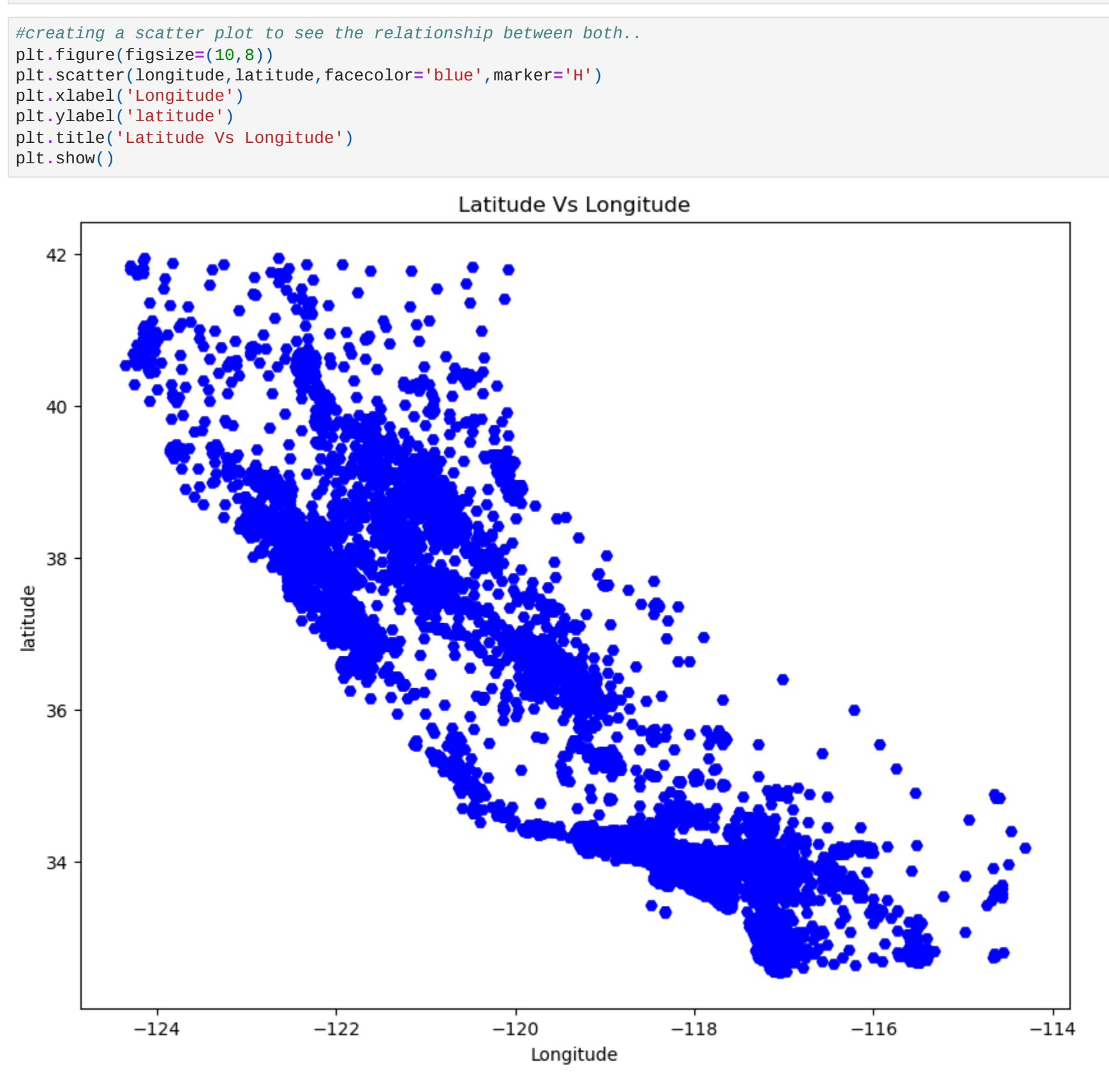
In [22]:
```python
def median_value(df2):
  New_sorted_data=sorted(df2)
  n=len(New_sorted_data)


  if n % 2 == 1:                             #as u can see that it will print the odd numbers
        median = New_sorted_data[n // 2]
  else:                                      # as u can see that it will print th Even number of elements
        middle1 = New_sorted_data[(n - 1) // 2]
        middle2 = New_sorted_data[n // 2]
        median = (middle1 + middle2) / 2

  return median


#calculating median of total bedrooms

median_total_bedrooms = median_value(df2['total_bedrooms'])
print("Total_Median_Bedrooms:", median_total_bedrooms)

#calculating median of population
median_total_population =median_value(df2["population"])
print ("Total_median_population:",median_total_population)

#calculating median of median_house_value
median_house_value =median_value(df2["median_house_value"])
print("total median of house value:",median_house_value)
```

```
Total_Median_Bedrooms: 438.0
Total_median_population: 1166.0
total median of house value: 179700.0
```

#7. Plot latitude versus longitude and explain your observations

In [23]:
```python
#Extracting latitude and longitude...
latitude=df2["latitude"]
longitude=df2["longitude"]
```

In [27]:
```python
#creating a scatter plot to see the relationship between both..
plt.figure(figsize=(10,8))
plt.scatter(longitude,latitude,facecolor='blue',marker='H')
plt.xlabel('Longitude')
plt.ylabel('latitude')
plt.title('Latitude Vs Longitude')
plt.show()
```



Observation : As We can see the point is generally moving upwards from right to left.so it will be considered as negavite correlation.

#8. Create a data set for which the ocean_proximity is 'Near ocean'.

In [32]:
```python
filtered_data=df2[df2['ocean_proximity']=="NEAR OCEAN"] #By using filter function we can find the exact values of necessary data..
filtered_data
```

Out[32]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
1850	-124.17	41.80	16	2739	480.0	1259	436	3.7557	109400	NEAR OCEAN
1851	-124.30	41.80	19	2672	552.0	1298	478	1.9797	85800	NEAR OCEAN
1852	-124.23	41.75	11	3159	616.0	1343	479	2.4805	73200	NEAR OCEAN
1853	-124.21	41.77	17	3461	722.0	1947	647	2.5795	68400	NEAR OCEAN
1854	-124.19	41.78	15	3140	714.0	1645	640	1.6654	74600	NEAR OCEAN
...	...	...	...	...	...	...	...	...	...	...
20380	-118.83	34.14	16	1316	194.0	450	173	10.1597	500001	NEAR OCEAN
20381	-118.83	34.14	16	1956	312.0	671	319	6.4001	321800	NEAR OCEAN
20423	-119.00	34.08	17	1822	438.0	578	291	5.4346	428600	NEAR OCEAN
20424	-118.75	34.18	4	16704	2704.0	6187	2207	6.6122	357600	NEAR OCEAN
20425	-118.75	34.17	18	6217	858.0	2703	834	6.8075	325900	NEAR OCEAN

2658 rows × 10 columns

In [34]:
```python
df2["ocean_proximity"].value_counts() #counting the number of rows in data set....
```

Out[34]:
```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

#9. Find the mean and median of the median income for the data set created in question 8

In [38]:
```python
filtered_data['median_income'].mean()  #Finding Mean From the Data set created in Question 8...
```

Out[38]:
```
4.0057848006019565
```

In [40]:
```python
filtered_data['median_income'].median()  #Finding Median From the Data set created in Question 8...
```

Out[40]:
```
3.64705
```

#10. Please create a new column named total_bedroom_size. If the total bedrooms is 10 or less, it should be quoted as small. If the total bedrooms is 11 or more but less than 1000, it should be medium, otherwise it should be considered large.

In [41]:
```python
df['total_bedroom_size']=["small" if x<=10 else "large" for x in df.total_bedrooms]
df.head(20)
```

Out[41]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity	total_bedroom_size
0	-122.23	37.88	41	880	129.0	322	126	8.3252	452600	NEAR BAY	large
1	-122.22	37.86	21	7099	1106.0	2401	1138	8.3014	358500	NEAR BAY	large
2	-122.24	37.85	52	1467	190.0	496	177	7.2574	352100	NEAR BAY	large
3	-122.25	37.85	52	1274	235.0	558	219	5.6431	341300	NEAR BAY	large
4	-122.25	37.85	52	1627	280.0	565	259	3.8462	342200	NEAR BAY	large
5	-122.25	37.85	52	919	213.0	413	193	4.0368	269700	NEAR BAY	large
6	-122.25	37.84	52	2535	489.0	1094	514	3.6591	299200	NEAR BAY	large
7	-122.25	37.84	52	3104	687.0	1157	647	3.1200	241400	NEAR BAY	large
8	-122.26	37.84	42	2555	665.0	1206	595	2.0804	226700	NEAR BAY	large
9	-122.25	37.84	52	3549	707.0	1551	714	3.6912	261100	NEAR BAY	large
10	-122.26	37.85	52	2202	434.0	910	402	3.2031	281500	NEAR BAY	large
11	-122.26	37.85	52	3503	752.0	1504	734	3.2705	241800	NEAR BAY	large
12	-122.26	37.85	52	2491	474.0	1098	468	3.0750	213500	NEAR BAY	large
13	-122.26	37.84	52	696	191.0	345	174	2.6736	191300	NEAR BAY	large
14	-122.26	37.85	52	2643	626.0	1212	620	1.9167	159200	NEAR BAY	large
15	-122.26	37.85	50	1120	283.0	697	264	2.1250	140000	NEAR BAY	large
16	-122.27	37.85	52	1966	347.0	793	331	2.7750	152500	NEAR BAY	large
17	-122.27	37.85	52	1228	293.0	648	303	2.1202	155500	NEAR BAY	large
18	-122.26	37.84	50	2239	455.0	990	419	1.9911	158700	NEAR BAY	large
19	-122.27	37.84	52	1503	298.0	690	275	2.6033	162900	NEAR BAY	large