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

from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>

Saving housing.xlsx to housing.xlsx
```

```
data=pd.read_excel("housing.xlsx")
print(data)
```

```
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
0        -122.23     37.88                  41          880
129.0
1        -122.22     37.86                  21         7099
1106.0
2        -122.24     37.85                  52         1467
190.0
3        -122.25     37.85                  52         1274
235.0
4        -122.25     37.85                  52         1627
280.0
...          ...       ...                 ...          ...
...
20635    -121.09     39.48                  25         1665
374.0
20636    -121.21     39.49                  18          697
150.0
20637    -121.22     39.43                  17         2254
485.0
20638    -121.32     39.43                  18         1860
409.0
20639    -121.24     39.37                  16         2785
616.0

       population  households  median_income  median_house_value  \
0             322         126         8.3252              452600
1            2401        1138         8.3014              358500
2             496         177         7.2574              352100
3             558         219         5.6431              341300
4             565         259         3.8462              342200
...           ...         ...            ...                 ...
20635         845         330         1.5603               78100
20636         356         114         2.5568               77100
20637        1007         433         1.7000               92300
20638         741         349         1.8672               84700
```

```
20639        1387         530         2.3886                89400

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20640 rows x 10 columns]
```

#In this project, we have provided you with a California dataset to answer the following questions. Before starting the project, please explore the dataset online also. All these questions are mandatory. It is a 'must' to give an introduction of the project. You must put comments after codes to explain the step. Each result must be explained after getting the result and figures. If you miss any of these, marks will be deducted. These steps are a must to write a good project. You must explain your project and features in the introduction section. Please explain which feature is nominal, ordinal, discrete or continuous.

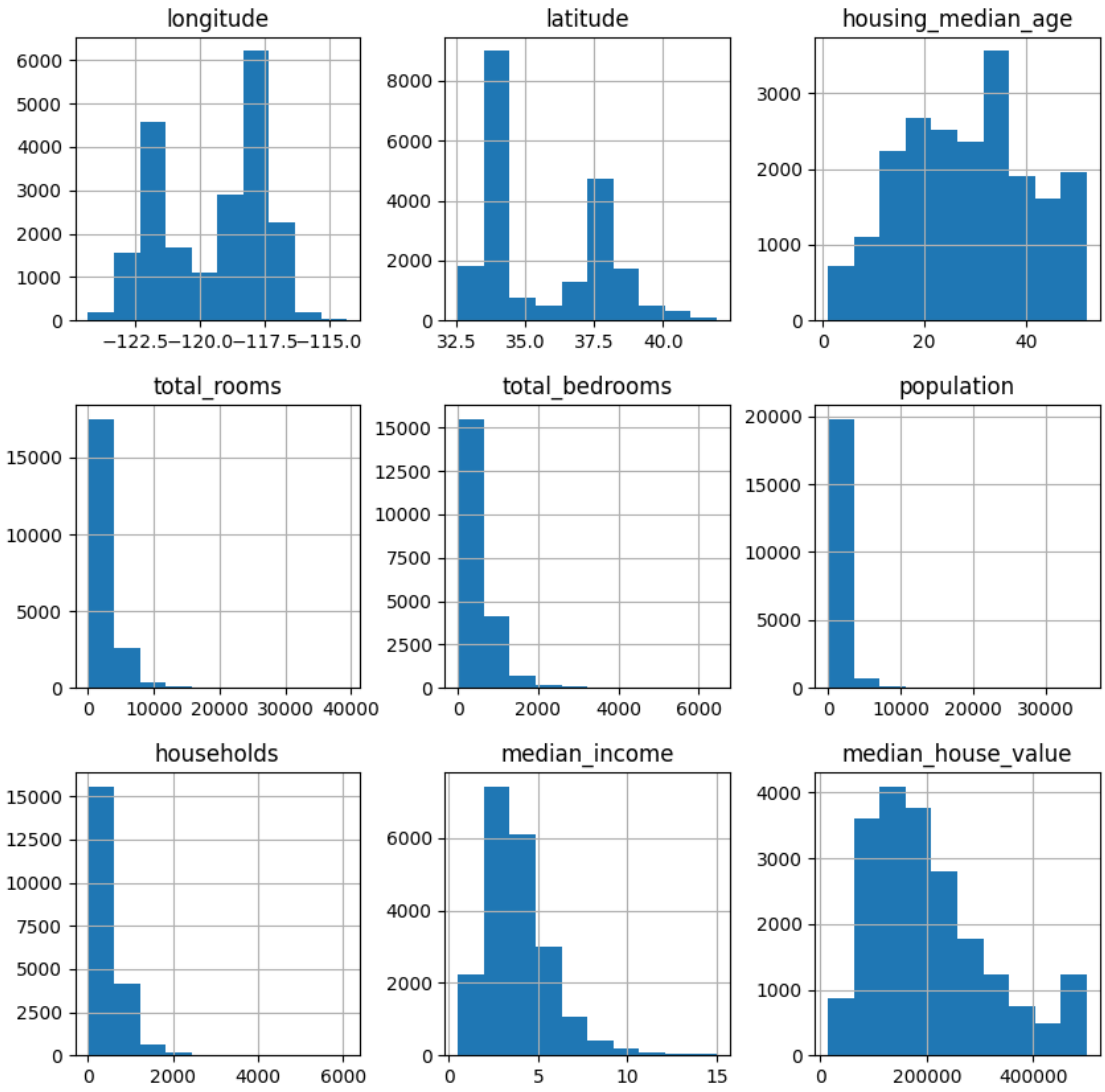
Please address the following questions:

#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.

```
data['median_income'].mean()        # mean()function gives the mean of
the values
                                    # mean gives the average values of
a dataset.

# Here average median_income is 3.870671 that is 3.9
```

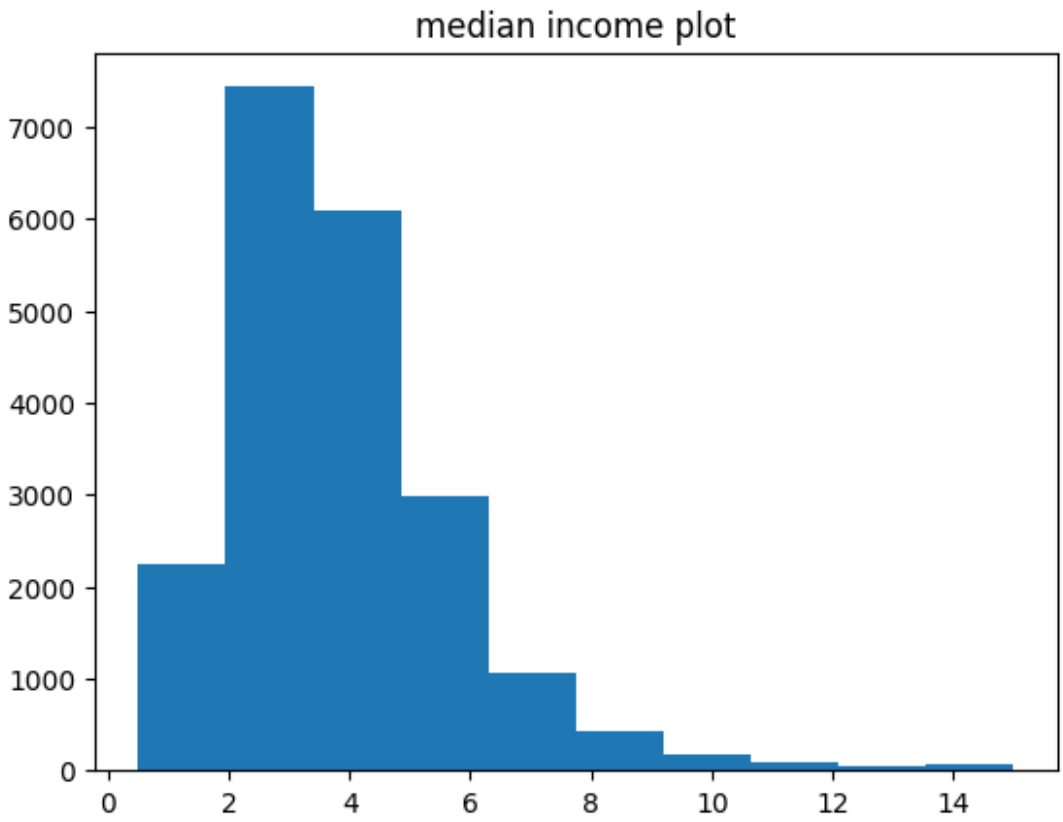
```
3.8706710029069766
```

```
data.hist(bins=10,figsize=(10,10))  # Histogram is used to see the
distribution of a numerical value
plt.show()

#From the above plot,it is to be noted that the outliers are present
for housing_median_age and for median_house_value
#while total_rooms,total_bedrooms,population,households,median_income
are of Right skewed.
#While latitude and logitude are of asymmetric,i.e.,highly skewed.
```



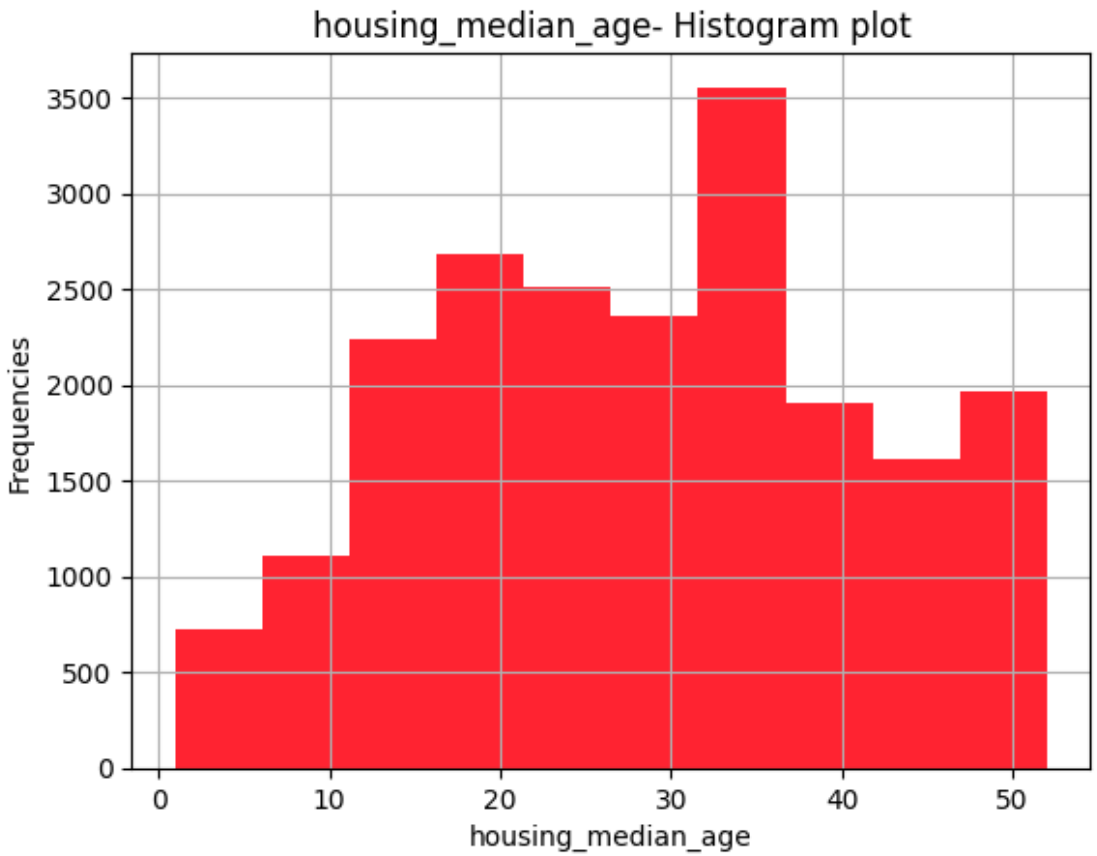

```
median_income = data.loc[:,"median_income"]
print("median income is :",median_income.mean())
plt.hist(median_income)
plt.title("median income plot")
plt.show()
```

```
median income is : 3.8706710029069766
```

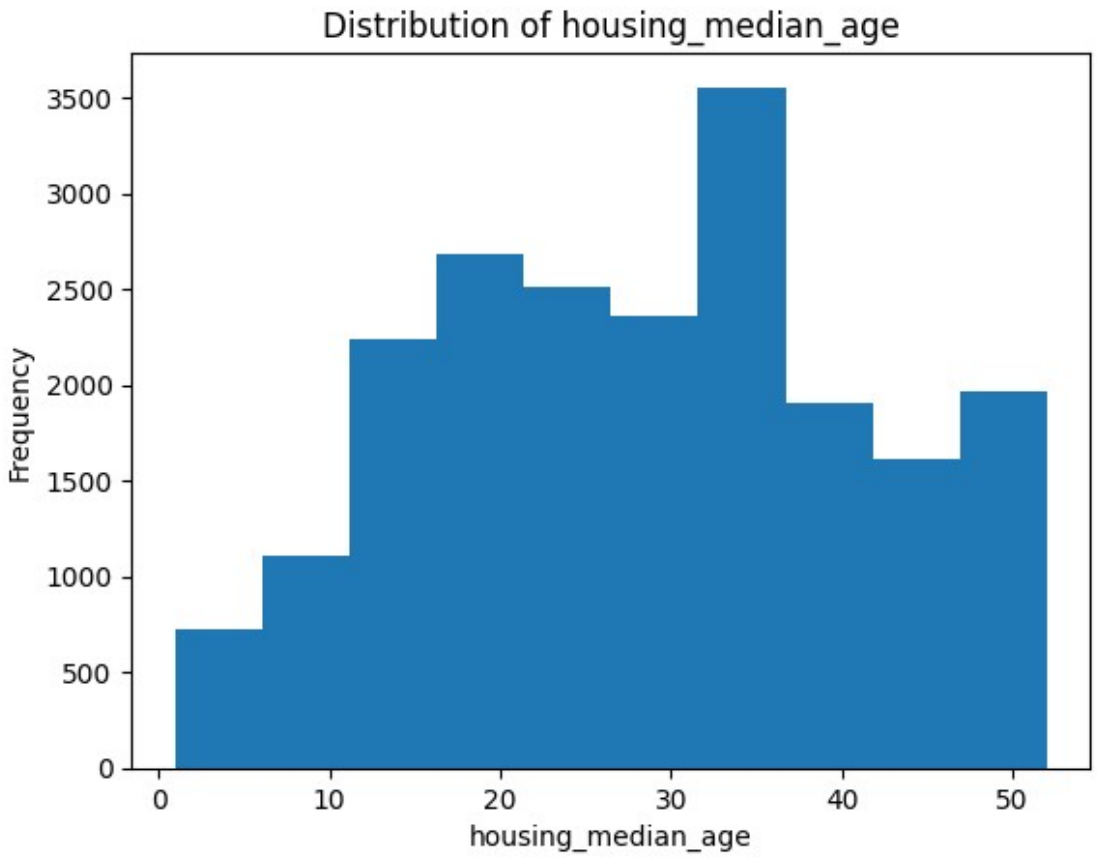


#2. Draw an appropriate plot to see the distribution of housing_median_age and explain your observations.

```
plt.hist(data["housing_median_age"],color='#FF2331')  #Histogram is
used to see the distribution of a numerical value.
plt.title("housing_median_age- Histogram plot")    # x-
axis=housing_median_age,y-axis=Frequencies
plt.xlabel("housing_median_age")
plt.ylabel("Frequencies")
plt.grid(True)
plt.show()
```




```
plt.hist(data.housing_median_age)  # Histogram is used to see the
distribution of a numerical value.
plt.xlabel("housing_median_age")                                    #
x-axis=housing_median_age,y-axis=Frequencies
plt.ylabel("Frequency")
plt.title("Distribution of housing_median_age")
plt.show()

# From the above hist plot we can come to the analysis that it is
distributed symmetrically.
# we can know the skewness of the above plot by using :Skewed
=3*(mean-median)/std()
```




```
data['housing_median_age'].mean()
```

```
28.639486434108527
```

```
data['housing_median_age'].median()
```

```
29.0
```

```
data['housing_median_age'].std()
```

```
12.58555761211165
```

```
Skewed=3*(28.63-29.0)/12.58
Skewed
```

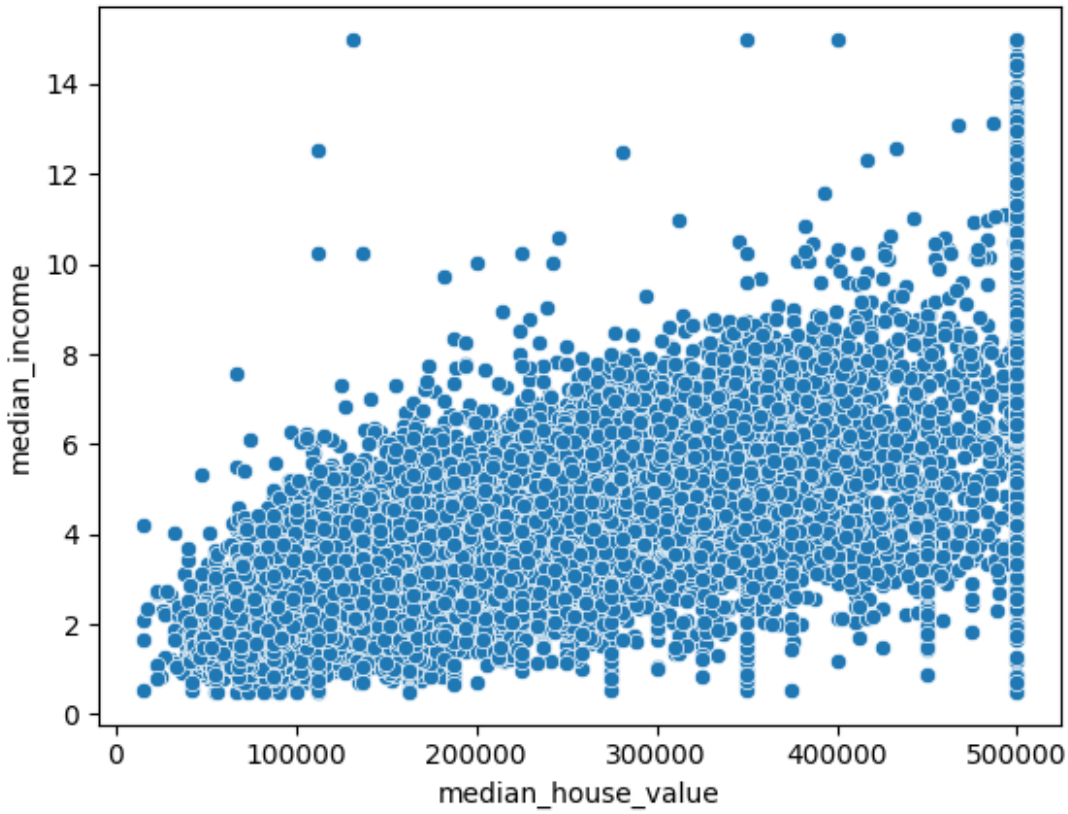
```
-0.0882352941176473
```

```
# The Skewness of the above plot is -0.08 which is -0.5 to -0.1.From
this it is to be concluded that it is of perfectly symmetrical.
# Finally,from the visualisation plot and from the Skewness it is to
be noted that the housing_median_age is perfectly symmetrical.
```

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

```
sns.scatterplot(x="median_house_value",y="median_income",data=data)  #
scatter plot gives a relation between two numerical values.
                                                                     #
x-axis median_house_value  #Y axis median_income
```

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




```
#From the above visualisation it is to be analysed that with an
increase in the median_house_value
#there is also an increase in the median income.While,
#an outlier is present in median_house_value which is shown in the
fig.Therefore,median_house_value is directly proportional to median
income
```

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

```
data[data.isnull().any(axis=1)]#The isnull() method returns a
DataFrame object where all the values are
                            #replaced with a Boolean value True for
NULL values, and otherwise False.
                            #Here missing values are denoted by NaN

#In the above code,missing values are identified by using
```

isnull()method.This missing values are identified in the column 'total_bedrooms'.

```
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
290       -122.16     37.77                  47         1256
NaN
341       -122.17     37.75                  38          992
NaN
538       -122.28     37.78                  29         5154
NaN
563       -122.24     37.75                  45          891
NaN
696       -122.10     37.69                  41          746
NaN
...           ...       ...                 ...          ...
...
20267     -119.19     34.20                  18         3620
NaN
20268     -119.18     34.19                  19         2393
NaN
20372     -118.88     34.17                  15         4260
NaN
20460     -118.75     34.29                  17         5512
NaN
20484     -118.72     34.28                  17         3051
NaN

       population  households  median_income  median_house_value  \
290           570         218         4.3750              161900
341           732         259         1.6196               85100
538          3741        1273         2.5762              173400
563           384         146         4.9489              247100
696           387         161         3.9063              178400
...           ...         ...            ...                 ...
20267        3171         779         3.3409              220500
20268        1938         762         1.6953              167400
20372        1701         669         5.1033              410700
20460        2734         814         6.6073              258100
20484        1705         495         5.7376              218600

      ocean_proximity
290          NEAR BAY
341          NEAR BAY
538          NEAR BAY
563          NEAR BAY
696          NEAR BAY
...               ...
20267      NEAR OCEAN
20268      NEAR OCEAN
```

```
20372       <1H OCEAN
20460       <1H OCEAN
20484       <1H OCEAN

[207 rows x 10 columns]
```

```
new_data=data.dropna(subset=["total_bedrooms"])
new_data.head()
```

```
   longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
0    -122.23     37.88                  41          880
129.0
1    -122.22     37.86                  21         7099
1106.0
2    -122.24     37.85                  52         1467
190.0
3    -122.25     37.85                  52         1274
235.0
4    -122.25     37.85                  52         1627
280.0

   population  households  median_income  median_house_value
ocean_proximity
0         322         126         8.3252              452600
NEAR BAY
1        2401        1138         8.3014              358500
NEAR BAY
2         496         177         7.2574              352100
NEAR BAY
3         558         219         5.6431              341300
NEAR BAY
4         565         259         3.8462              342200
NEAR BAY
```

```
new_data=data.dropna(subset=["total_bedrooms"])#dropna() method allows
the user to analyze and drop Rows/Columns with Null values in
different ways.
new_data
```

```
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
0        -122.23     37.88                  41          880
129.0
1        -122.22     37.86                  21         7099
1106.0
2        -122.24     37.85                  52         1467
190.0
3        -122.25     37.85                  52         1274
235.0
4        -122.25     37.85                  52         1627
```

```
280.0
...          ...       ...                  ...          ...
...
20635    -121.09     39.48                  25         1665
374.0
20636    -121.21     39.49                  18          697
150.0
20637    -121.22     39.43                  17         2254
485.0
20638    -121.32     39.43                  18         1860
409.0
20639    -121.24     39.37                  16         2785
616.0

       population  households  median_income  median_house_value  \
0             322         126         8.3252              452600
1            2401        1138         8.3014              358500
2             496         177         7.2574              352100
3             558         219         5.6431              341300
4             565         259         3.8462              342200
...           ...         ...            ...                 ...
20635         845         330         1.5603               78100
20636         356         114         2.5568               77100
20637        1007         433         1.7000               92300
20638         741         349         1.8672               84700
20639        1387         530         2.3886               89400

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20433 rows x 10 columns]
```

```
# While in the above code the missing values are dropped from the
column named'total_bedrooms'by using dropna() method.
```

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

```
data["total_bedrooms"]=data["total_bedrooms"].fillna(data['total_bedro
oms'].mean())
data
```

```
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
0        -122.23     37.88                  41          880
129.0
1        -122.22     37.86                  21         7099
1106.0
2        -122.24     37.85                  52         1467
190.0
3        -122.25     37.85                  52         1274
235.0
4        -122.25     37.85                  52         1627
280.0
...          ...       ...                 ...          ...
...
20635    -121.09     39.48                  25         1665
374.0
20636    -121.21     39.49                  18          697
150.0
20637    -121.22     39.43                  17         2254
485.0
20638    -121.32     39.43                  18         1860
409.0
20639    -121.24     39.37                  16         2785
616.0

       population  households  median_income  median_house_value  \
0             322         126         8.3252              452600
1            2401        1138         8.3014              358500
2             496         177         7.2574              352100
3             558         219         5.6431              341300
4             565         259         3.8462              342200
...           ...         ...            ...                 ...
20635         845         330         1.5603               78100
20636         356         114         2.5568               77100
20637        1007         433         1.7000               92300
20638         741         349         1.8672               84700
20639        1387         530         2.3886               89400

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
```

```
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20640 rows x 10 columns]
```

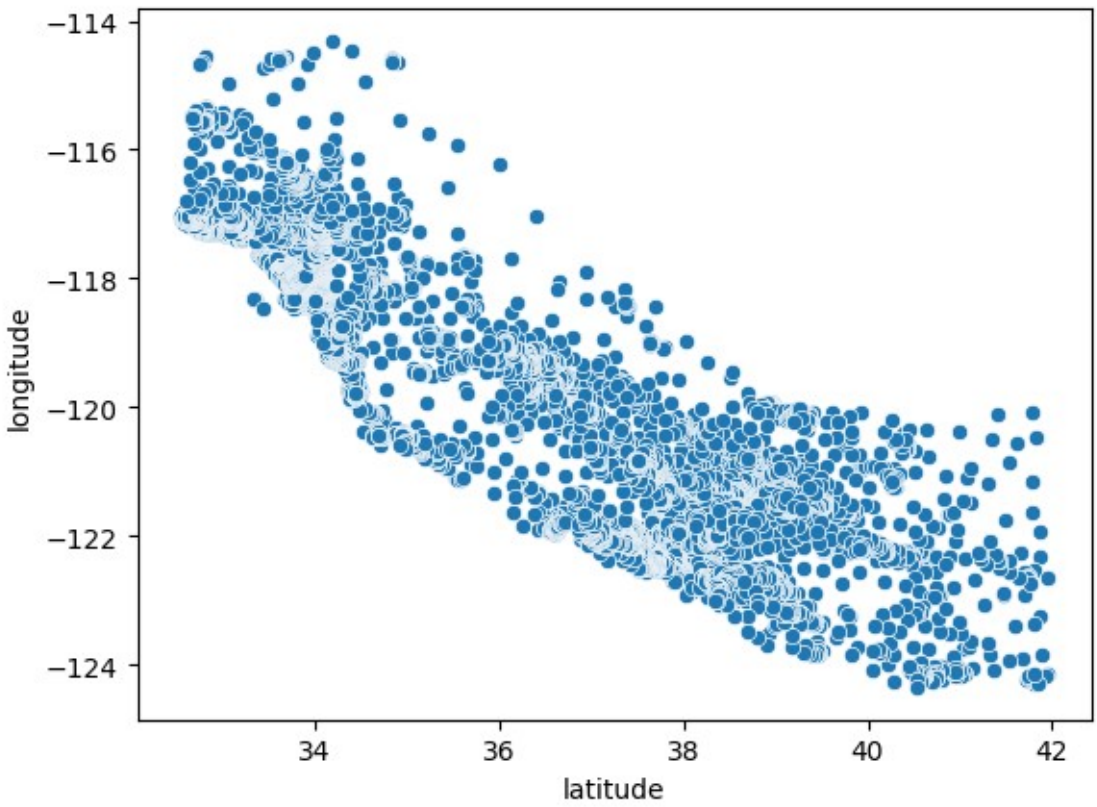
```
#In the above code,a new dataset had been created where the missing
values in the
#'total_bedrooms'which are denoted by NaN are replaced with the mean
value of the 'total_bedrooms'.

#For eg:row no.290,341,538 and other rows with the missing values are
replaced with the mean value of 'total_bedrooms'
i.e.,537.8705525375618,respectively.
```

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

#7. Plot latitude versus longitude and explain your observations.

```
sns.scatterplot(x='latitude',y='longitude',data=data)  #Scatter plot
gives a relationship between two numerical values.
                                                       #x-axis =
latitude #y-axis = longitude
```

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




```
#From the above plot,it is to be noted that with an decrease in
longitude,latitude is increased.
#From this point of view,it is to be known that latitude and longitude
are not dependent on each other.
#From this we can say that longitude is inversely proportional to
latitude.

#From the above plot it is to be noted that latitude vs longitude has
negative correlation
#as here y-axis is increasing while x-axis is decreasing,i.e., both
are moving in an opposite direction.
```

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

```
ocean_data=data.loc[data["ocean_proximity"]=="NEAR OCEAN"] # loc is an
label based method which is used to select rows and columns by
Names/Labels.
ocean_data                                                  #loc is an
essential pandas methods used for filtering ,selecting and
manipulating the data.
```

```
       longitude  latitude  housing_median_age  total_rooms
total_bedrooms  \
1850     -124.17     41.80                  16         2739
480.0
1851     -124.30     41.80                  19         2672
```

```
552.0
1852     -124.23     41.75                 11         3159
616.0
1853     -124.21     41.77                 17         3461
722.0
1854     -124.19     41.78                 15         3140
714.0
...          ...       ...                ...          ...
...
20380    -118.83     34.14                 16         1316
194.0
20381    -118.83     34.14                 16         1956
312.0
20423    -119.00     34.08                 17         1822
438.0
20424    -118.75     34.18                  4        16704
2704.0
20425    -118.75     34.17                 18         6217
858.0

       population  households  median_income  median_house_value  \
1850         1259         436         3.7557              109400
1851         1298         478         1.9797               85800
1852         1343         479         2.4805               73200
1853         1947         647         2.5795               68400
1854         1645         640         1.6654               74600
...           ...         ...            ...                 ...
20380         450         173        10.1597              500001
20381         671         319         6.4001              321800
20423         578         291         5.4346              428600
20424        6187        2207         6.6122              357600
20425        2703         834         6.8075              325900

      ocean_proximity
1850       NEAR OCEAN
1851       NEAR OCEAN
1852       NEAR OCEAN
1853       NEAR OCEAN
1854       NEAR OCEAN
...               ...
20380      NEAR OCEAN
20381      NEAR OCEAN
20423      NEAR OCEAN
20424      NEAR OCEAN
20425      NEAR OCEAN

[2658 rows x 10 columns]
```

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

```
ocean_data['median_income'].mean() #mean() gives the averages of the
data
                                   #mean()is applicable for discrete and
continuous data but not for categorical data.
```

```
4.0057848006019565
```

```
ocean_data['median_income'].median() #median()value gives the 50th
percentile of the set of all observations.
```

```
3.64705
```

```
#The mean and median value of the 'median_income'in the created new
dataset are:4.005784 and 3.64705 respectively.
```

#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.

```
conditions = [
              (data['total_bedrooms'] <=10),
              (data['total_bedrooms'] >=11)&(data['total_bedrooms']
<=1000),
              (data['total_bedrooms'] > 1000)
             ]
values = ['small','medium','large']

data['total_bedroom_size']=np.select(conditions,values)
data
```

https://www.kaggle.com/code/keerthijyoshna/housing-data-ipynb