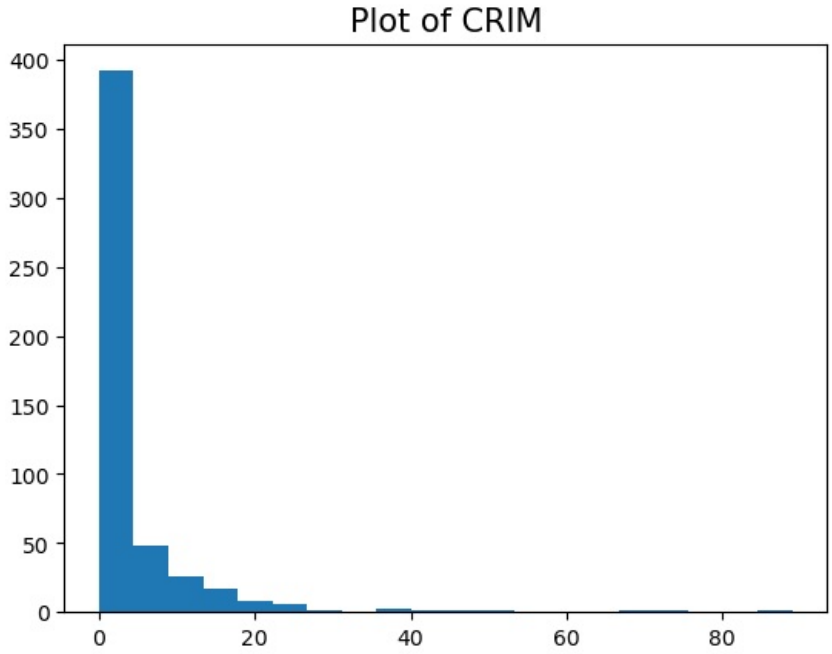
In [ ]:

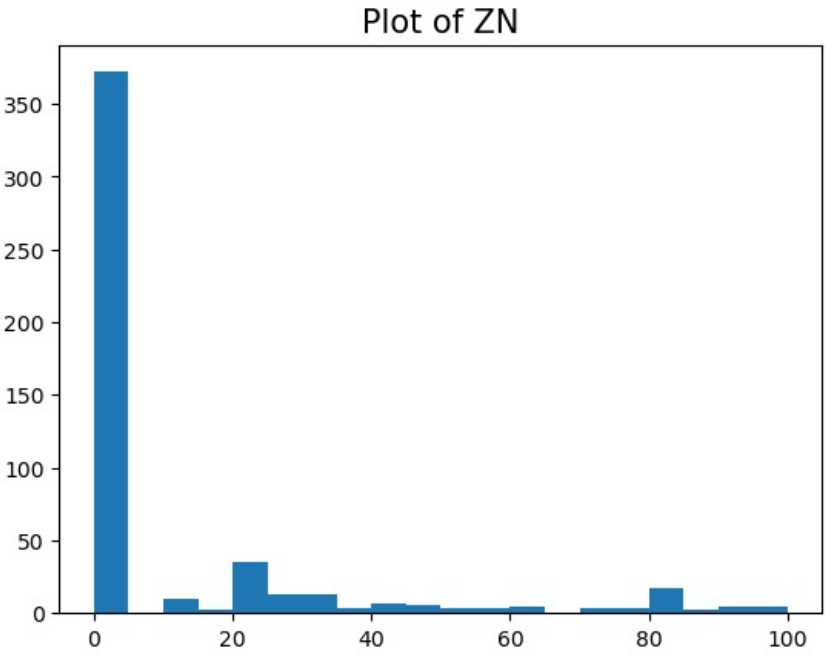
```
"""
Week 3 & 4 Exercises
"""

from __future__ import print_function
from itertools import zip_longest

import csv
import logging
import sys
import numpy as np
import pandas as pd
import random
import thinkplot
import thinkstats2
import datetime
import regression
import statsmodels.formula.api as smf
import statsmodels.api as sm
import matplotlib.pyplot as plt
import math


def ReadData(filename):
    ### Read in the Boston housing data set (given as a .csv file) from the local directory

    df = pd.read_csv(filename)
    return df

def PlotHistogram(df1):
    for c in df1.columns:
        plt.title("Plot of "+c,fontsize=15)
        plt.hist(df1[c],bins=20)
        plt.show()

def Createscatterplot(df1):
    plt.scatter(df1['CRIM'],df1['PRICE'])
    plt.show()

def CreatePlotUsingLog10(df1):
    plt.scatter(np.log10(df1['CRIM']),df1['PRICE'],c='red')
    plt.title("Crime rate (Log) vs. Price plot", fontsize=18)
    plt.xlabel("Log of Crime rate",fontsize=15)
    plt.ylabel("Price",fontsize=15)
    plt.grid(True)
    plt.show()


def CalculateMeanmedianAndpercentage(df1):
    ### Calculate mean rooms per dwelling
    print ('mean rooms per dwelling :', df1['RM'].mean());

    ### Calculate median age
    print('median Age :', df1['AGE'].median());

    ### Calculate average (mean) distances to five Boston employment centres
    print('mean distances to five Boston employment centres :',df1['DIS'].mean())

    ### calculate the percentage of houses with low price (< $20,000)
    # Create a Pandas series and directly compare it with 20 and we can do this because Pandas series is basica
    low_price=df1['PRICE']<20
    # This creates a Boolean array of True, False
    print(low_price)
    # True = 1, False = 0, so now if you take an average of this Numpy array, you will know how many 1's are th
    # That many houses are priced below 20,000. So that is the answer.
    # You can convert that into percentage by multiplying with 100
    pcnt=low_price.mean()*100
    print("\nPercentage of house with <20,000 price is: ",pcnt)


def strip_whitespace(s):
    return s.strip()

def Activity5_Excercise():

    ### Read in the Boston housing dataset (given as a .csv file) from the local direction:
    filename="Boston_housing.csv"
    df = ReadData(filename);

     ### Check first 10 records
    df.head(10)
    print (df)

    ### Find the total number of records:
    print(df.shape)

    ### Create a smaller DataFrame with columns which do not include 'CHAS', 'NOX', 'B', and 'LSTAT'
```

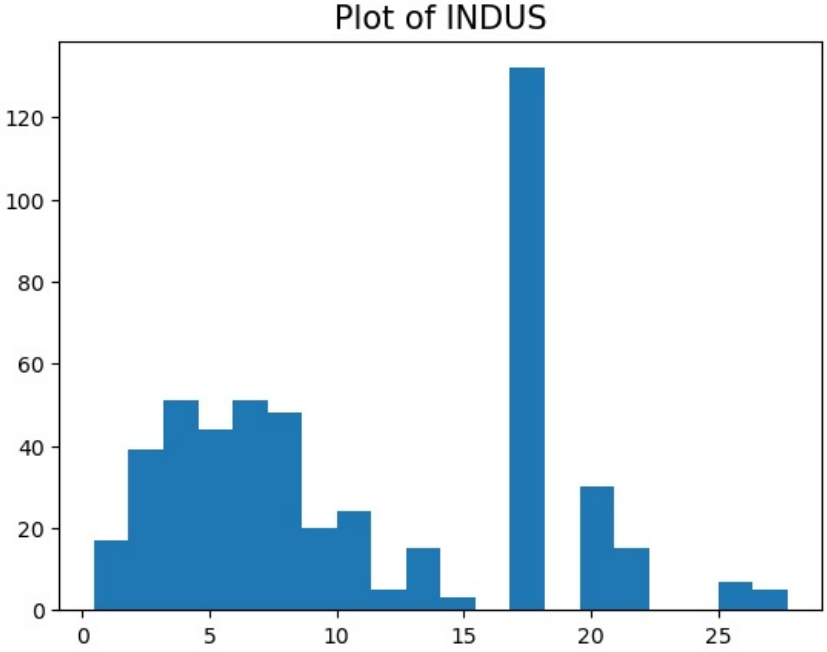
```
    df1=df[['CRIM','ZN','INDUS','RM','AGE','DIS','RAD','TAX','PTRATIO','PRICE']]

    ### Check the last 7 records of the new DataFrame you just created
    df1.tail(7)
    print (df1)

    ### Plot histograms of all the variables (columns) in the new DataFrame by using a for loop:
    ### Plot them all at once using a for loop. Try to add a unique title to a plot.
    PlotHistogram(df1)

    ### Crime rate could be an indicator of house price (people don't want to live in highcrime areas).
    ### Create a scatter plot of crime rate versus price:
    Createscatterplot(df1)

    ### Plot using log10(crime) versus price.
    CreatePlotUsingLog10(df1)

    ### Calculate some useful statistics, such as mean rooms per dwelling, median age, mean distances to five B
    ##  with a low price (< $20,000).
    CalculateMeanmedianAndpercentage(df1)


def Activity6_Excercise():

    ### Load the files
    filename="adult_income_data.csv"
    df = pd.read_csv(filename)
    df.head()
    print (df)

    ### Create a script that will read a text file line by line.
    names = []
    with open('adult_income_names.txt','r') as f:
        for line in f:
            f.readline()
            var=line.split(":")[0]
            names.append(var)

    print (names)

    ### Add a name of Income for the response variable to the dataset.
    names.append('Income')
    df = pd.read_csv("adult_income_data.csv",names=names)
    df.head()
    print (df)

    ### Find the missing values.
    print('Find the missing values :', df.describe())

    ### Create a DataFrame with only age, education, and occupation by using subsetting.
    vars_class = ['workclass','education','marital-status','occupation','relationship','race','sex','native-cou
    for v in vars_class:
        classes=df[v].unique()
        num_classes = df[v].nunique()
        print("There are {} classes in the \"{}\" column. They are: {}".format(num_classes,v,classes))
        print("-"*100)

    print ('Is there any missing (NULL) data in the dataset :', df.isnull().sum())

    df_subset = df[['age','education','occupation','race']]
    df_subset.head()
    print (df_subset)

    ### Plot a histogram of age with a bin size of 20.

    df_subset['age'].hist(bins=20)

    ### Create a function to strip the whitespace characters.
    #def strip_whitespace(s):
    #return s.strip()

    ### Use the apply method to apply this function to all the columns with string values, create a new column,

    # Education column
    df_subset['education_stripped']=df['education'].apply(strip_whitespace)
    df_subset['education']=df_subset['education_stripped']
    df_subset.drop(labels=['education_stripped'],axis=1,inplace=True)

    # Occupation column
    df_subset['occupation_stripped']=df['occupation'].apply(strip_whitespace)
    df_subset['occupation']=df_subset['occupation_stripped']
    df_subset.drop(labels=['occupation_stripped'],axis=1,inplace=True)

    # Race column
    df_subset['race_stripped']=df['race'].apply(strip_whitespace)
    df_subset['race']=df_subset['race_stripped']
    df_subset.drop(labels=['race_stripped'],axis=1,inplace=True)

    ### Find the number of people who are aged between 30 and 50.
```

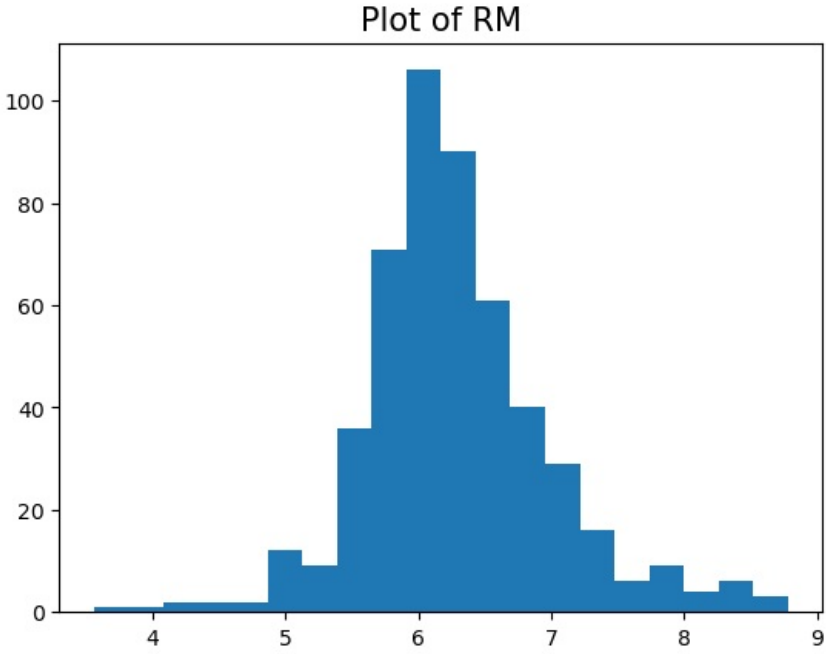
```
    df_filtered=df_subset[(df_subset['race']=='Black') & (df_subset['age']>=30) & (df_subset['age']<=50)]
    df_filtered.head()
    answer_1=df_filtered.shape[0]
    print("There are {} black people of age between 30 and 50 in this dataset.".format(answer_1))

    ### Group the records based on age and education to find how the mean age is distributed.

    print("Group the records based on age and education : ", df_subset.groupby(['race','education']).mean())

    ### Group by occupation and show the summary statistics of age. Find which profession has the oldest worker

    print('Group by occupation and show the summary statistics of age :' , df_subset.groupby('occupation').desc

    ### share of the workforce above the 75th percentile.
    print ('share of the workforce above the 75th percentile :', df_subset.groupby('occupation').describe()['ag

    ### Use subset and groupby to find outliers.
    ### Plot the values on a bar chart.
    ### Merge the data using common keys.

    occupation_stats= df_subset.groupby('occupation').describe()['age']
    plt.figure(figsize=(15,8))
    plt.barh(y=occupation_stats.index,width=occupation_stats['count'])
    plt.yticks(fontsize=13)
    plt.show()

    df_1 = df[['age','workclass','occupation']].sample(5,random_state=101)
    df_1.head()
    df_2 = df[['education','race','occupation']].sample(5,random_state=101)
    df_2.head()
    df_merged = pd.merge(df_1,df_2,on='occupation',how='inner').drop_duplicates()
    print('df_merged : ', df_merged)

def SeriesandPracticeArithmeticSteps():
    series1 = pd.Series([7.3, -2.5, 3.4, 1.5],  index = ['a', 'c', 'd', 'e'])
    series2 = pd.Series([-2.1, 3.6, -1.5, 4, 3.1],  index = ['a', 'c', 'e', 'f', 'g'])

    series_sum = series1 + series2
    print('series_sum : ',series_sum)

    series_diff = series1 - series2
    print('series_diff : ',series_diff)

def main():
    print('Inside Main function')

    ### Activity 5: Generating Statistics from a CSV File
    Activity5_Excercise()

    ### Activity 6: Working with the Adult Income Dataset (UCI)
    Activity6_Excercise()

    ### Create a series and practice basic arithmetic steps
    # a. Series 1 = 7.3, -2.5, 3.4, 1.5
    # i. Index = ‘a’, ‘c’, ‘d’, ‘e’
    # b. Series 2 = -2.1, 3.6, -1.5, 4, 3.1
    # i. Index = ‘a’, ‘c’, ‘e’, ‘f’, ‘g’
    # c. Add Series 1 and Series 2 together and print the results
    # d. Subtract Series 1 from Series 2 and print the results
    SeriesandPracticeArithmeticSteps()

if __name__ == "__main__":
    main()
```

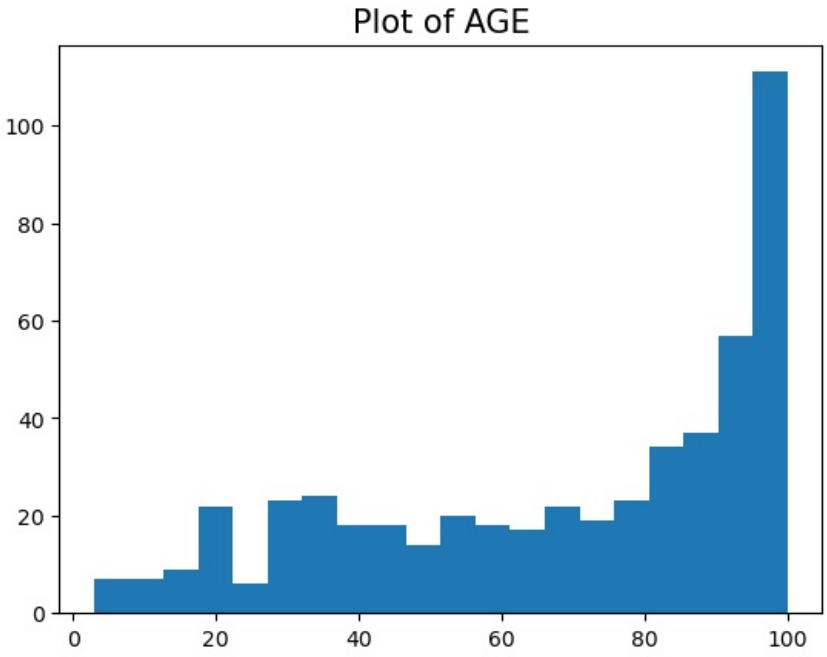
```
Inside Main function
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
0    0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296
1    0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242
2    0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242
3    0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222
4    0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222
..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...
501  0.06263   0.0  11.93     0  0.573  6.593  69.1  2.4786    1  273
502  0.04527   0.0  11.93     0  0.573  6.120  76.7  2.2875    1  273
503  0.06076   0.0  11.93     0  0.573  6.976  91.0  2.1675    1  273
504  0.10959   0.0  11.93     0  0.573  6.794  89.3  2.3889    1  273
505  0.04741   0.0  11.93     0  0.573  6.030  80.8  2.5050    1  273

     PTRATIO       B  LSTAT  PRICE
0       15.3  396.90   4.98   24.0
1       17.8  396.90   9.14   21.6
2       17.8  392.83   4.03   34.7
3       18.7  394.63   2.94   33.4
4       18.7  396.90   5.33   36.2
..       ...     ...    ...    ...
501     21.0  391.99   9.67   22.4
502     21.0  396.90   9.08   20.6
503     21.0  396.90   5.64   23.9
504     21.0  393.45   6.48   22.0
505     21.0  396.90   7.88   11.9

[506 rows x 14 columns]
(506, 14)
        CRIM    ZN  INDUS     RM   AGE     DIS  RAD  TAX  PTRATIO  PRICE
0    0.00632  18.0   2.31  6.575  65.2  4.0900    1  296     15.3   24.0
1    0.02731   0.0   7.07  6.421  78.9  4.9671    2  242     17.8   21.6
2    0.02729   0.0   7.07  7.185  61.1  4.9671    2  242     17.8   34.7
3    0.03237   0.0   2.18  6.998  45.8  6.0622    3  222     18.7   33.4
4    0.06905   0.0   2.18  7.147  54.2  6.0622    3  222     18.7   36.2
..       ...   ...    ...    ...   ...     ...  ...  ...      ...    ...
501  0.06263   0.0  11.93  6.593  69.1  2.4786    1  273     21.0   22.4
502  0.04527   0.0  11.93  6.120  76.7  2.2875    1  273     21.0   20.6
503  0.06076   0.0  11.93  6.976  91.0  2.1675    1  273     21.0   23.9
504  0.10959   0.0  11.93  6.794  89.3  2.3889    1  273     21.0   22.0
505  0.04741   0.0  11.93  6.030  80.8  2.5050    1  273     21.0   11.9

[506 rows x 10 columns]
```

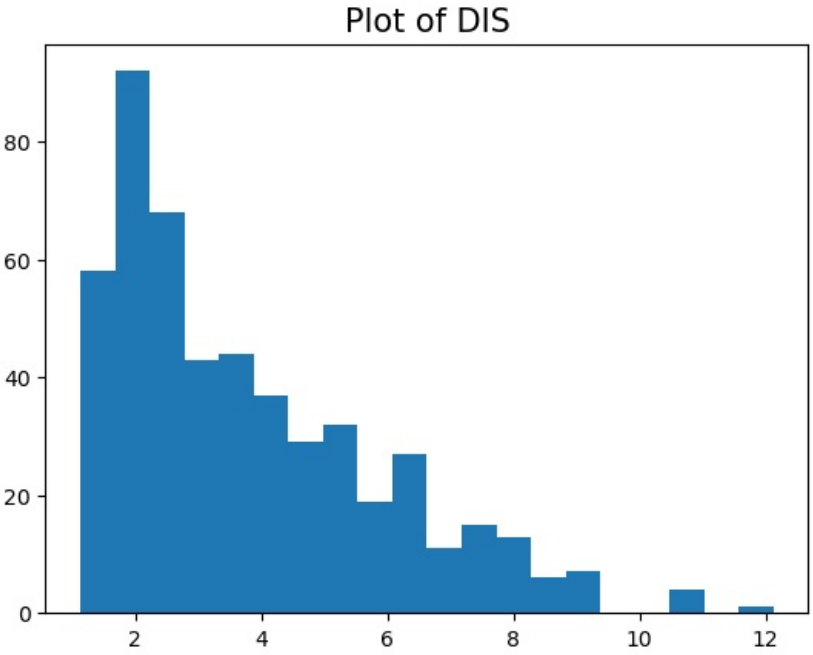


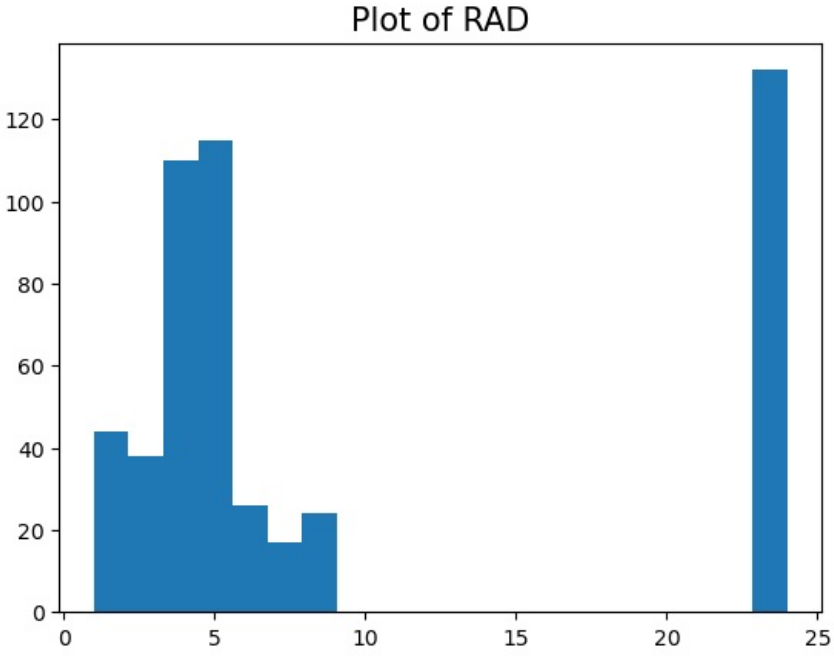

Plot of ZN
350
300
250
200
150
100
50
0
0
20
40
60
80
100

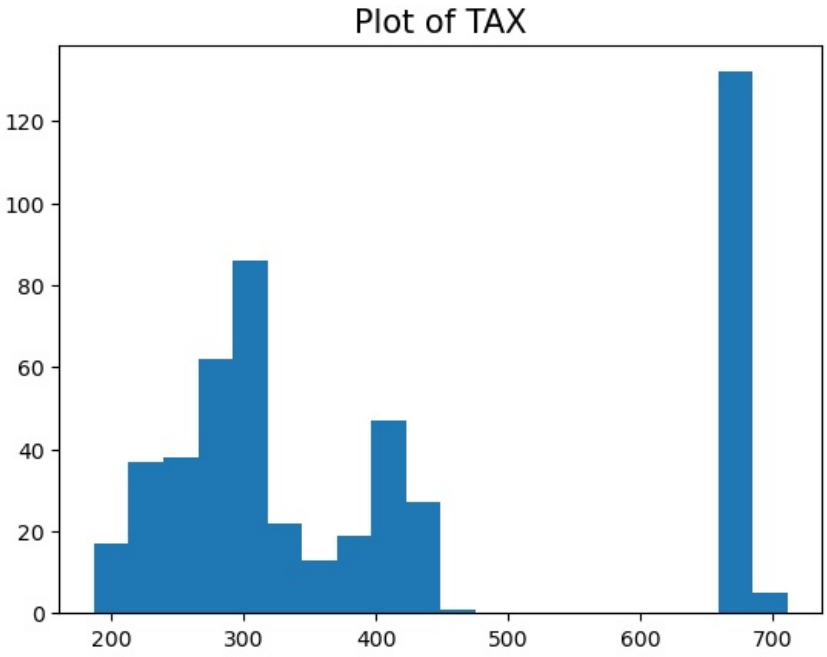

Plot of INDUS
120
100
80
60
40
20
0
0
5
10
15
20
25

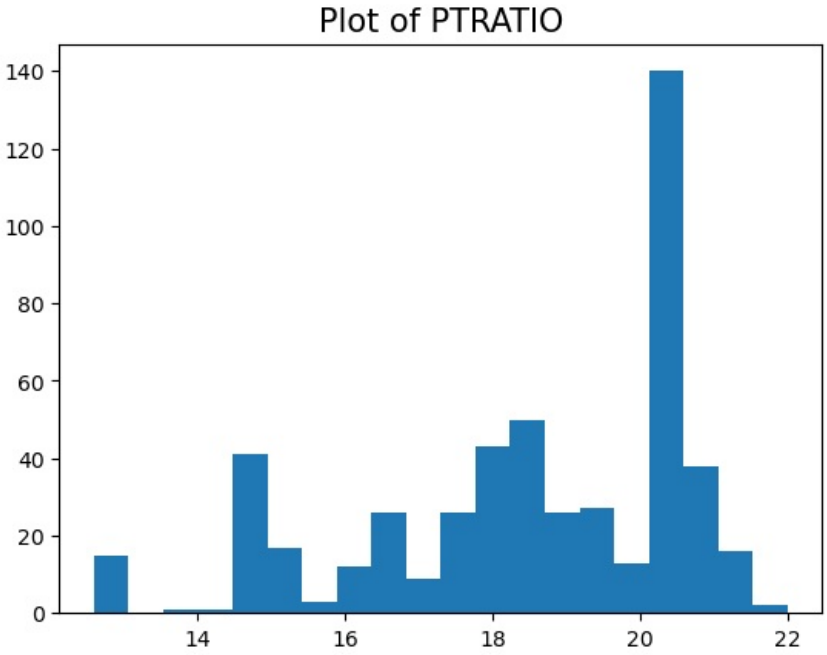

Plot of RM
100
80
60
40
20
0
4
5
6
7
8
9

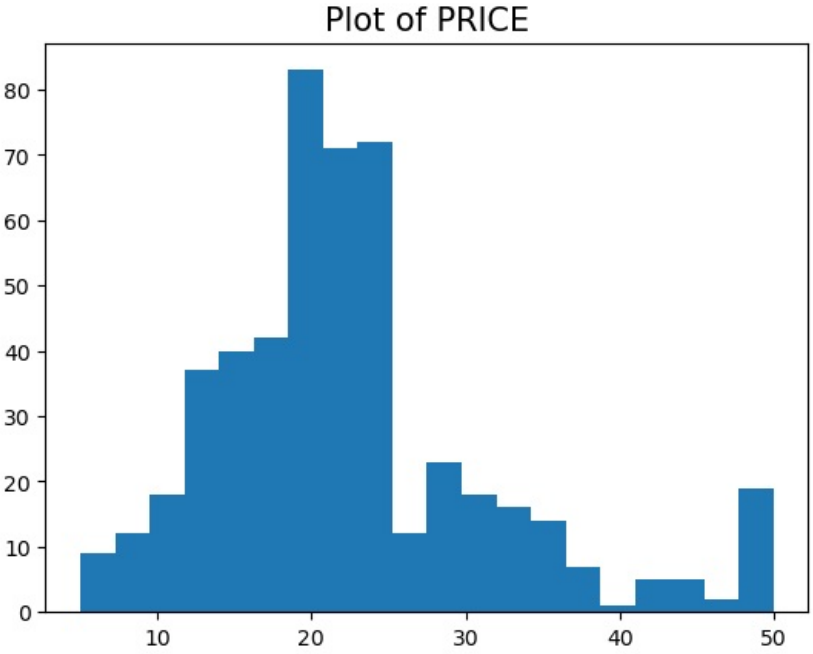

Plot of AGE
100
80
60
40
20
0
0
20
40
60
80
100

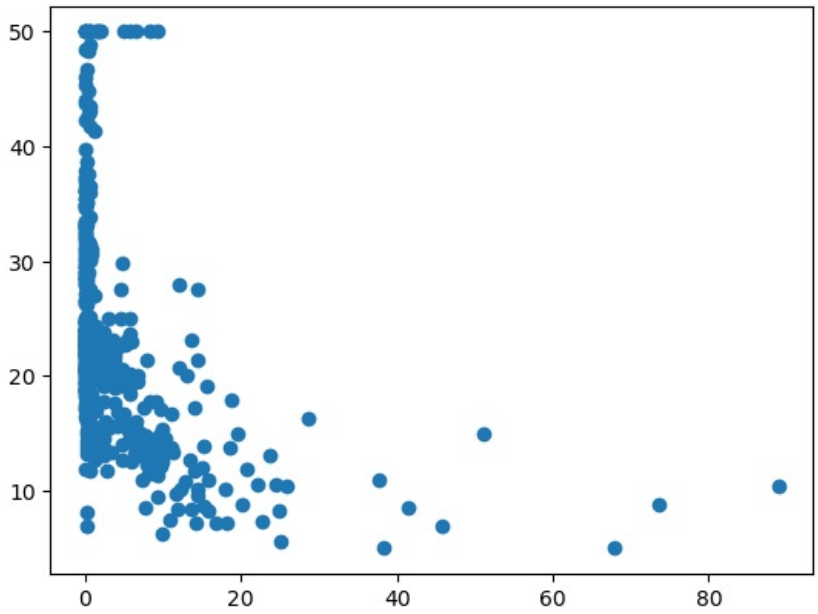

Plot of DIS
80
60
40
20
0
2
4
6
8
10
12

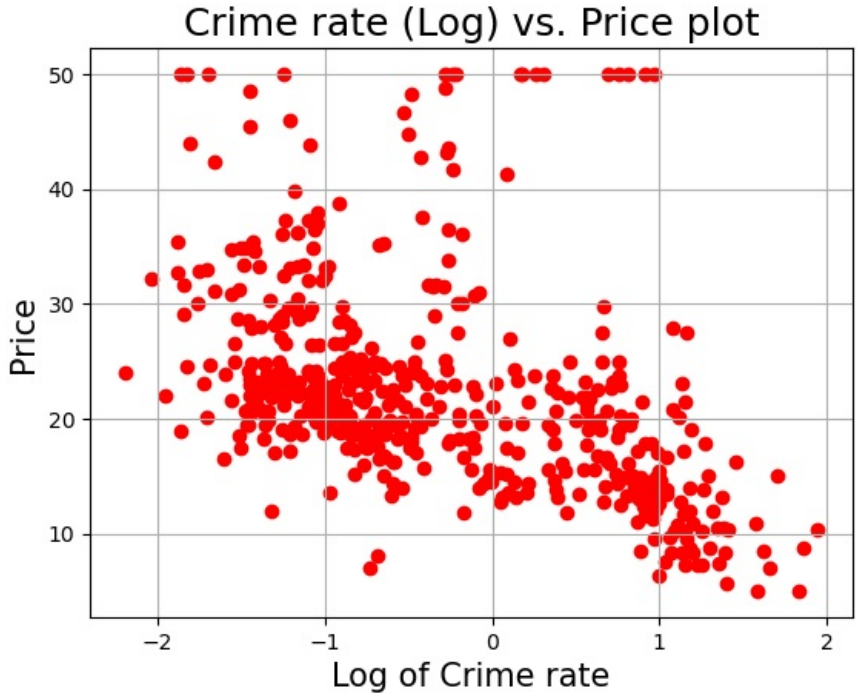

Plot of RAD
120
100
80
60
40
20
0
0
5
10
15
20
25


Plot of TAX
120
100
80
60
40
20
0
200
300
400
500
600
700


Plot of PTRATIO
140
120
100
80
60
40
20
0
14
16
18
20
22


Plot of PRICE
80
70
60
50
40
30
20
10
0
10
20
30
40
50





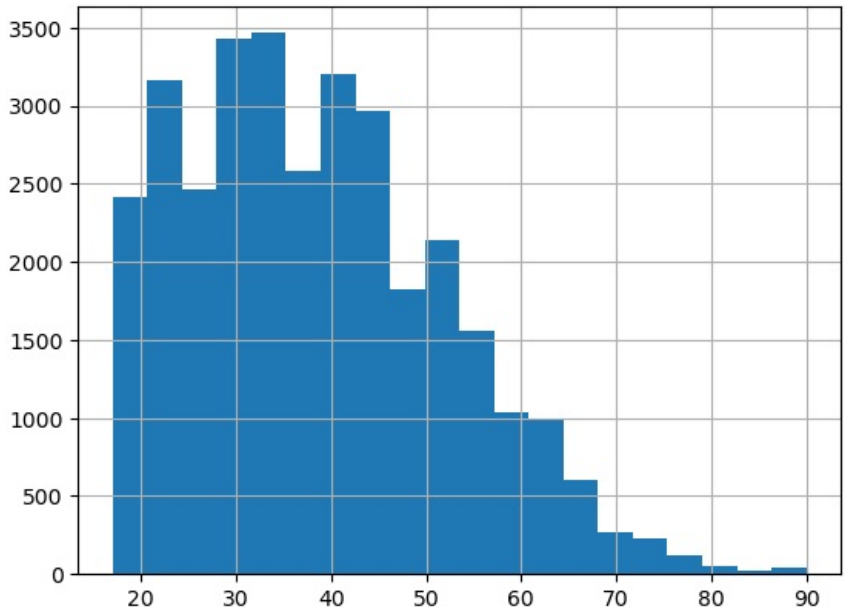

```
mean rooms per dwelling : 6.284634387351779
median Age : 77.5
mean distances to five Boston employment centres : 3.795042687747036
0      False
1      False
2      False
3      False
4      False
       ...
501    False
502    False
503    False
504    False
505     True
Name: PRICE, Length: 506, dtype: bool

Percentage of house with <20,000 price is:  41.50197628458498
       39          State-gov   77516    Bachelors   13        Never-married  \
0      50   Self-emp-not-inc   83311    Bachelors   13   Married-civ-spouse
1      38            Private  215646      HS-grad    9             Divorced
2      53            Private  234721         11th    7   Married-civ-spouse
3      28            Private  338409    Bachelors   13   Married-civ-spouse
4      37            Private  284582      Masters   14   Married-civ-spouse
...    ..                ...     ...          ...  ...                  ...
32555  27            Private  257302   Assoc-acdm   12   Married-civ-spouse
32556  40            Private  154374      HS-grad    9   Married-civ-spouse
32557  58            Private  151910      HS-grad    9              Widowed
32558  22            Private  201490      HS-grad    9        Never-married
32559  52       Self-emp-inc  287927      HS-grad    9   Married-civ-spouse

             Adm-clerical   Not-in-family   White     Male   2174   0   40  \
0         Exec-managerial         Husband   White     Male      0   0   13
1       Handlers-cleaners   Not-in-family   White     Male      0   0   40
```

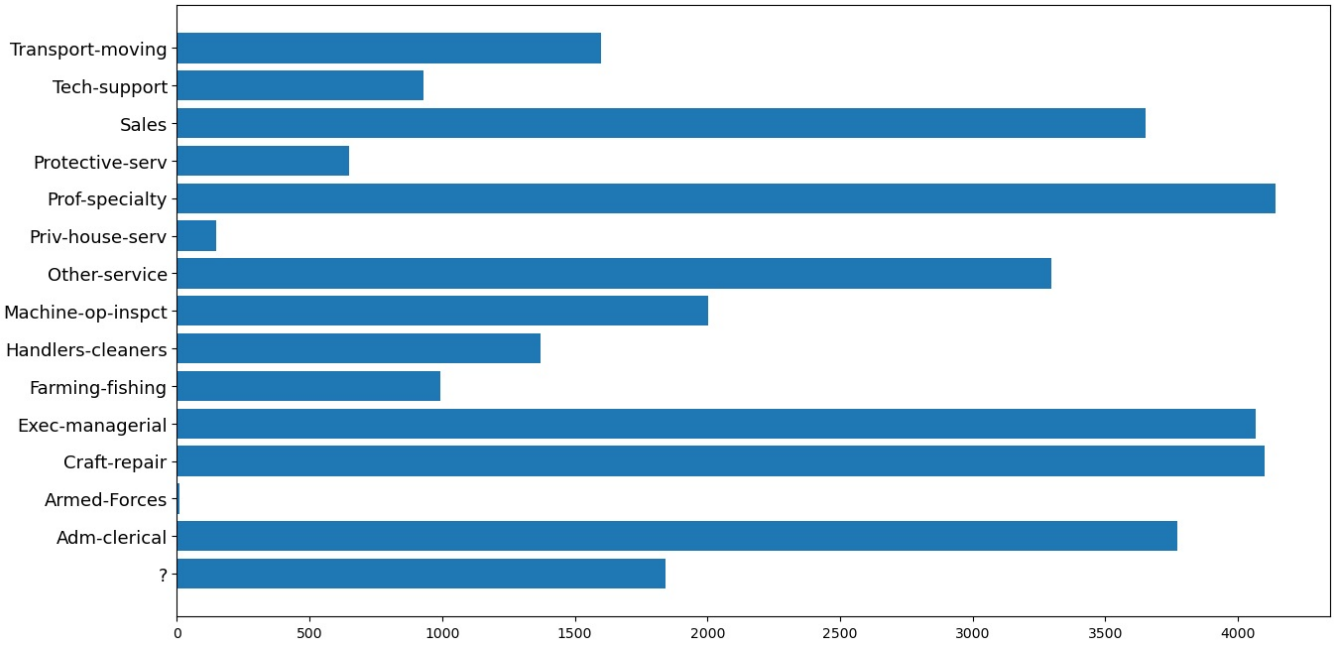
```
2       Handlers-cleaners         Husband   Black     Male      0   0   40
3          Prof-specialty            Wife   Black   Female      0   0   40
4        Exec-managerial             Wife   White   Female      0   0   40
...                  ...             ...     ...      ...    ...  ..  ...
32555        Tech-support            Wife   White   Female      0   0   38
32556   Machine-op-inspct         Husband   White     Male      0   0   40
32557        Adm-clerical       Unmarried   White   Female      0   0   40
32558        Adm-clerical       Own-child   White     Male      0   0   20
32559     Exec-managerial            Wife   White   Female  15024   0   40

        United-States   <=50K
0       United-States   <=50K
1       United-States   <=50K
2       United-States   <=50K
3                Cuba   <=50K
4       United-States   <=50K
...               ...     ...
32555   United-States   <=50K
32556   United-States    >50K
32557   United-States   <=50K
32558   United-States   <=50K
32559   United-States    >50K

[32560 rows x 15 columns]
['age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'r
ace', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country']
       age          workclass  fnlwgt    education  education-num  \
0       39          State-gov   77516    Bachelors             13
1       50   Self-emp-not-inc   83311    Bachelors             13
2       38            Private  215646      HS-grad              9
3       53            Private  234721         11th              7
4       28            Private  338409    Bachelors             13
...    ...                ...     ...          ...            ...
32556   27            Private  257302   Assoc-acdm             12
32557   40            Private  154374      HS-grad              9
32558   58            Private  151910      HS-grad              9
32559   22            Private  201490      HS-grad              9
32560   52       Self-emp-inc  287927      HS-grad              9

            marital-status          occupation    relationship    race  \
0            Never-married        Adm-clerical   Not-in-family   White
1       Married-civ-spouse     Exec-managerial         Husband   White
2                 Divorced   Handlers-cleaners   Not-in-family   White
3       Married-civ-spouse   Handlers-cleaners         Husband   Black
4       Married-civ-spouse      Prof-specialty            Wife   Black
...                    ...                 ...             ...     ...
32556   Married-civ-spouse        Tech-support            Wife   White
32557   Married-civ-spouse   Machine-op-inspct         Husband   White
32558              Widowed        Adm-clerical       Unmarried   White
32559        Never-married        Adm-clerical       Own-child   White
32560   Married-civ-spouse     Exec-managerial            Wife   White

           sex  capital-gain  capital-loss  hours-per-week  native-country  \
0         Male          2174             0              40   United-States
1         Male             0             0              13   United-States
2         Male             0             0              40   United-States
3         Male             0             0              40   United-States
4       Female             0             0              40            Cuba
...        ...           ...           ...             ...             ...
32556   Female             0             0              38   United-States
32557     Male             0             0              40   United-States
32558   Female             0             0              40   United-States
32559     Male             0             0              20   United-States
32560   Female         15024             0              40   United-States

       Income
0       <=50K
1       <=50K
2       <=50K
3       <=50K
4       <=50K
...       ...
32556   <=50K
32557    >50K
32558   <=50K
32559   <=50K
32560    >50K

[32561 rows x 15 columns]
Find the missing values :                 age        fnlwgt  education-num  capital-gain  capital-loss  \
count  32561.000000  3.256100e+04   32561.000000  32561.000000  32561.000000
mean      38.581647  1.897784e+05      10.080679   1077.648844     87.303830
std       13.640433  1.055500e+05       2.572720   7385.292085    402.960219
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000
25%       28.000000  1.178270e+05       9.000000      0.000000      0.000000
50%       37.000000  1.783560e+05      10.000000      0.000000      0.000000
75%       48.000000  2.370510e+05      12.000000      0.000000      0.000000
max       90.000000  1.484705e+06      16.000000  99999.000000   4356.000000
```

```
       hours-per-week
count    32561.000000
mean        40.437456
std         12.347429
min          1.000000
25%         40.000000
50%         40.000000
75%         45.000000
max         99.000000
There are 9 classes in the "workclass" column. They are: [' State-gov' ' Self-emp-not-inc' ' Private' ' Federal
-gov' ' Local-gov'
 ' ?' ' Self-emp-inc' ' Without-pay' ' Never-worked']
--------------------------------------------------------------------------------------------------------
There are 16 classes in the "education" column. They are: [' Bachelors' ' HS-grad' ' 11th' ' Masters' ' 9th' '
Some-college'
 ' Assoc-acdm' ' Assoc-voc' ' 7th-8th' ' Doctorate' ' Prof-school'
 ' 5th-6th' ' 10th' ' 1st-4th' ' Preschool' ' 12th']
--------------------------------------------------------------------------------------------------------
There are 7 classes in the "marital-status" column. They are: [' Never-married' ' Married-civ-spouse' ' Divorce
d'
 ' Married-spouse-absent' ' Separated' ' Married-AF-spouse' ' Widowed']
--------------------------------------------------------------------------------------------------------
There are 15 classes in the "occupation" column. They are: [' Adm-clerical' ' Exec-managerial' ' Handlers-clean
ers' ' Prof-specialty'
 ' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
 ' Farming-fishing' ' Machine-op-inspct' ' Tech-support' ' ?'
 ' Protective-serv' ' Armed-Forces' ' Priv-house-serv']
--------------------------------------------------------------------------------------------------------
There are 6 classes in the "relationship" column. They are: [' Not-in-family' ' Husband' ' Wife' ' Own-child' '
Unmarried'
 ' Other-relative']
--------------------------------------------------------------------------------------------------------
There are 5 classes in the "race" column. They are: [' White' ' Black' ' Asian-Pac-Islander' ' Amer-Indian-Eski
mo' ' Other']
--------------------------------------------------------------------------------------------------------
There are 2 classes in the "sex" column. They are: [' Male' ' Female']
--------------------------------------------------------------------------------------------------------
There are 42 classes in the "native-country" column. They are: [' United-States' ' Cuba' ' Jamaica' ' India' '
?' ' Mexico' ' South'
 ' Puerto-Rico' ' Honduras' ' England' ' Canada' ' Germany' ' Iran'
 ' Philippines' ' Italy' ' Poland' ' Columbia' ' Cambodia' ' Thailand'
 ' Ecuador' ' Laos' ' Taiwan' ' Haiti' ' Portugal' ' Dominican-Republic'
 ' El-Salvador' ' France' ' Guatemala' ' China' ' Japan' ' Yugoslavia'
 ' Peru' ' Outlying-US(Guam-USVI-etc)' ' Scotland' ' Trinadad&Tobago'
 ' Greece' ' Nicaragua' ' Vietnam' ' Hong' ' Ireland' ' Hungary'
 ' Holand-Netherlands']
--------------------------------------------------------------------------------------------------------
Is there any missing (NULL) data in the dataset : age              0
workclass         0
fnlwgt            0
education         0
education-num     0
marital-status    0
occupation        0
relationship      0
race              0
sex               0
capital-gain      0
capital-loss      0
hours-per-week    0
native-country    0
Income            0
dtype: int64
       age    education          occupation    race
0       39    Bachelors        Adm-clerical   White
1       50    Bachelors     Exec-managerial   White
2       38      HS-grad   Handlers-cleaners   White
3       53         11th   Handlers-cleaners   Black
4       28    Bachelors      Prof-specialty   Black
...    ...          ...                 ...     ...
32556   27   Assoc-acdm        Tech-support   White
32557   40      HS-grad   Machine-op-inspct   White
32558   58      HS-grad        Adm-clerical   White
32559   22      HS-grad        Adm-clerical   White
32560   52      HS-grad     Exec-managerial   White

[32561 rows x 4 columns]
There are 1630 black people of age between 30 and 50 in this dataset.
Group the records based on age and education :                                          age
race              education
Amer-Indian-Eskimo 10th          37.250000
                   11th          31.642857
                   12th          25.400000
                   1st-4th       45.750000
                   5th-6th       39.000000
...                                    ...
White              HS-grad       39.270777
                   Masters       44.338972
                   Preschool     40.289474
```

```
                  Prof-school   44.996109
                  Some-college  35.980989

[79 rows x 1 columns]
Group by occupation and show the summary statistics of age :                    count       mean        std
min   25%   50%   75%   max
occupation
?                 1843.0  40.882800  20.336350  17.0  21.0  35.0  61.0  90.0
Adm-clerical      3770.0  36.964456  13.362998  17.0  26.0  35.0  46.0  90.0
Armed-Forces         9.0  30.222222   8.089774  23.0  24.0  29.0  34.0  46.0
Craft-repair      4099.0  39.031471  11.606436  17.0  30.0  38.0  47.0  90.0
Exec-managerial   4066.0  42.169208  11.974548  17.0  33.0  41.0  50.0  90.0
Farming-fishing    994.0  41.211268  15.070283  17.0  29.0  39.0  52.0  90.0
Handlers-cleaners 1370.0  32.165693  12.372635  17.0  23.0  29.0  39.0  90.0
Machine-op-inspct 2002.0  37.715285  12.068266  17.0  28.0  36.0  46.0  90.0
Other-service     3295.0  34.949621  14.521508  17.0  22.0  32.0  45.0  90.0
Priv-house-serv    149.0  41.724832  18.633688  17.0  24.0  40.0  57.0  81.0
Prof-specialty    4140.0  40.517633  12.016676  17.0  31.0  40.0  48.0  90.0
Protective-serv    649.0  38.953775  12.822062  17.0  29.0  36.0  47.0  90.0
Sales             3650.0  37.353973  14.186352  17.0  25.0  35.0  47.0  90.0
Tech-support       928.0  37.022629  11.316594  17.0  28.0  36.0  44.0  73.0
Transport-moving  1597.0  40.197871  12.450792  17.0  30.0  39.0  49.0  90.0
share of the workforce above the 75th percentile :                    count       mean        std   min   25%
50%   75%   max
occupation
?                 1843.0  40.882800  20.336350  17.0  21.0  35.0  61.0  90.0
Adm-clerical      3770.0  36.964456  13.362998  17.0  26.0  35.0  46.0  90.0
Armed-Forces         9.0  30.222222   8.089774  23.0  24.0  29.0  34.0  46.0
Craft-repair      4099.0  39.031471  11.606436  17.0  30.0  38.0  47.0  90.0
Exec-managerial   4066.0  42.169208  11.974548  17.0  33.0  41.0  50.0  90.0
Farming-fishing    994.0  41.211268  15.070283  17.0  29.0  39.0  52.0  90.0
Handlers-cleaners 1370.0  32.165693  12.372635  17.0  23.0  29.0  39.0  90.0
Machine-op-inspct 2002.0  37.715285  12.068266  17.0  28.0  36.0  46.0  90.0
Other-service     3295.0  34.949621  14.521508  17.0  22.0  32.0  45.0  90.0
Priv-house-serv    149.0  41.724832  18.633688  17.0  24.0  40.0  57.0  81.0
Prof-specialty    4140.0  40.517633  12.016676  17.0  31.0  40.0  48.0  90.0
Protective-serv    649.0  38.953775  12.822062  17.0  29.0  36.0  47.0  90.0
Sales             3650.0  37.353973  14.186352  17.0  25.0  35.0  47.0  90.0
Tech-support       928.0  37.022629  11.316594  17.0  28.0  36.0  44.0  73.0
Transport-moving  1597.0  40.197871  12.450792  17.0  30.0  39.0  49.0  90.0
```

```
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:162: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['education_stripped']=df['education'].apply(strip_whitespace)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:163: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['education']=df_subset['education_stripped']
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:164: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset.drop(labels=['education_stripped'],axis=1,inplace=True)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:167: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['occupation_stripped']=df['occupation'].apply(strip_whitespace)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:168: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['occupation']=df_subset['occupation_stripped']
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:169: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset.drop(labels=['occupation_stripped'],axis=1,inplace=True)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:172: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['race_stripped']=df['race'].apply(strip_whitespace)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:173: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset['race']=df_subset['race_stripped']
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:174: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  df_subset.drop(labels=['race_stripped'],axis=1,inplace=True)
C:\Users\sasinha\AppData\Local\Temp\ipykernel_57252\756109474.py:184: FutureWarning: The default value of numer
ic_only in DataFrameGroupBy.mean is deprecated. In a future version, numeric_only will default to False. Either
specify numeric_only or select only columns which should be valid for the function.
  print("Group the records based on age and education : ", df_subset.groupby(['race','education']).mean())
```






```
df_merged :     age workclass          occupation education    race
0   51   Private   Machine-op-inspct   HS-grad   White
1   19   Private               Sales      11th   White
2   40   Private     Exec-managerial   HS-grad   White
3   17   Private   Handlers-cleaners      10th   White
4   61   Private        Craft-repair   7th-8th   White
series_sum :  a    5.2
c    1.1
d    NaN
e    0.0
f    NaN
g    NaN
dtype: float64
series_diff :  a    9.4
c   -6.1
d    NaN
e    3.0
f    NaN
g    NaN
dtype: float64
```