Name : Cheruvathoor Abin Anto

Roll No : TIT2425008

**Practical 4**

Perform the following to apply Descriptive Statistics on Data Set **Adult**: Data Preparation, Exploraory Data Analysis - Summarizing the Data(Mean, Variance, Median), Data Distributions, Outlier Treatment, Measuring Asymmetry: Skewness and Pearson’s Median Skewness Coefficient (Discussions to be included).

**Data Preparation**

**Creating a dictionary**

file = open ('D:/adult', 'r')

def chr\_int(a):

if a.isdigit(): return int(a)

else : return 0

data = []

for line in file :

data1 = line.split(', ')

# why 15 here because 15 details of a person is in one line of data in the notepad file

## 39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K

# to return the integer as integer , we define the function chr\_int() manually else numbers will be taken as text.

if len(data1) == 15:

data.append([ chr\_int(data1 [0]), data1[1],

chr\_int(data1 [2]), data1[3],

chr\_int(data1 [4]), data1[5],

data1[6], data1[7], data1[8],

data1[9], chr\_int (data1[10]),

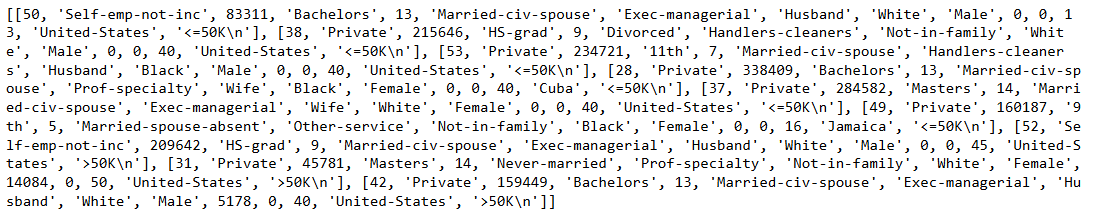
chr\_int(data1[11]),

chr\_int(data1[12]),

data1 [13], data1[14]

])

print(data[1:10])



**Using the DataFrame structure**

import pandas as pd

df = pd.DataFrame( data)

df.columns = [

'age', 'type\_employer', 'fnlwgt',

'education', 'education\_num', 'marital',

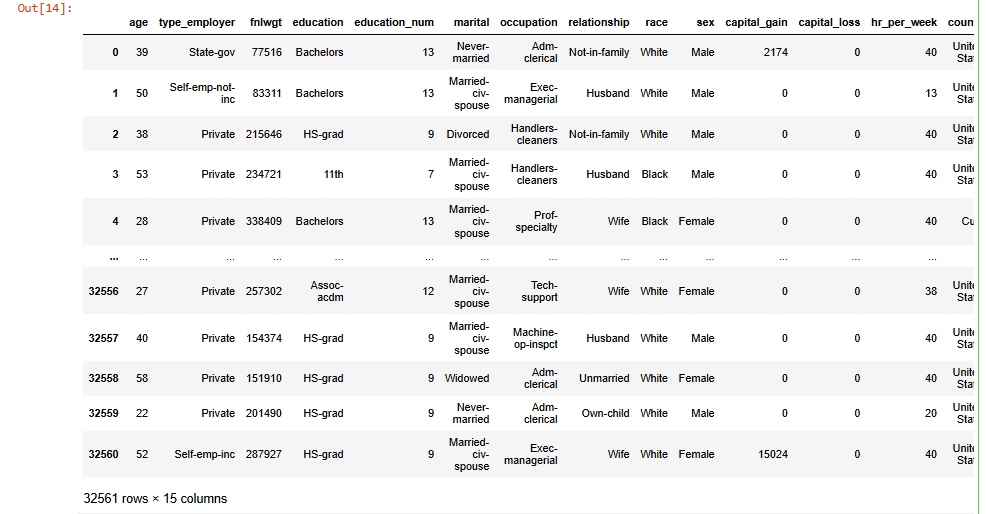
'occupation','relationship', 'race',

'sex', 'capital\_gain', 'capital\_loss',

'hr\_per\_week', 'country', 'income'

]

df



**The command shape gives exactly the number of data samples (in rows, in this**

**case) and features (in columns) :**

df.shape



**Count the number of items per country & size() : Counts the number of rows in each group, returning a pandas Series where :**

counts = df.groupby('country').size()

print (counts.head())

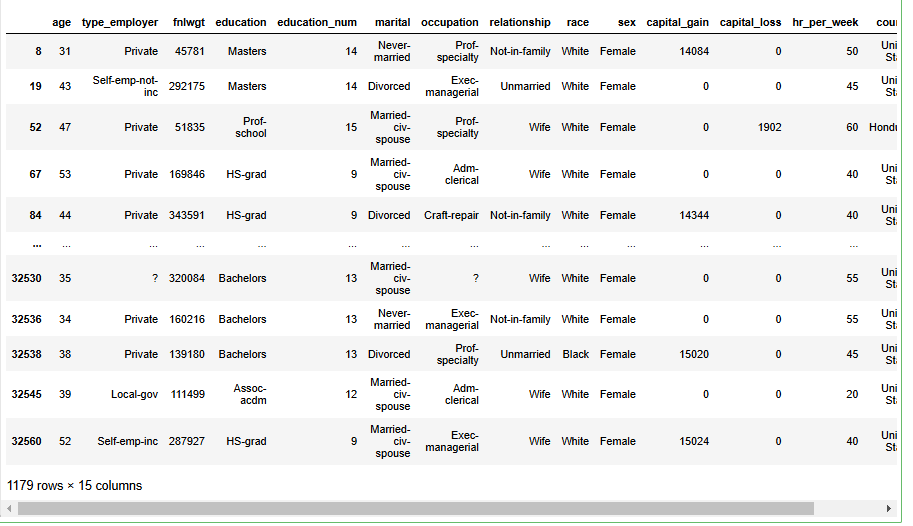


**Split people according to their gender into two groups : men and women**

ml = df[(df.sex == 'Male') ]

ml1 = df[(df.sex == 'Male') & (df.income== '>50K\n')]

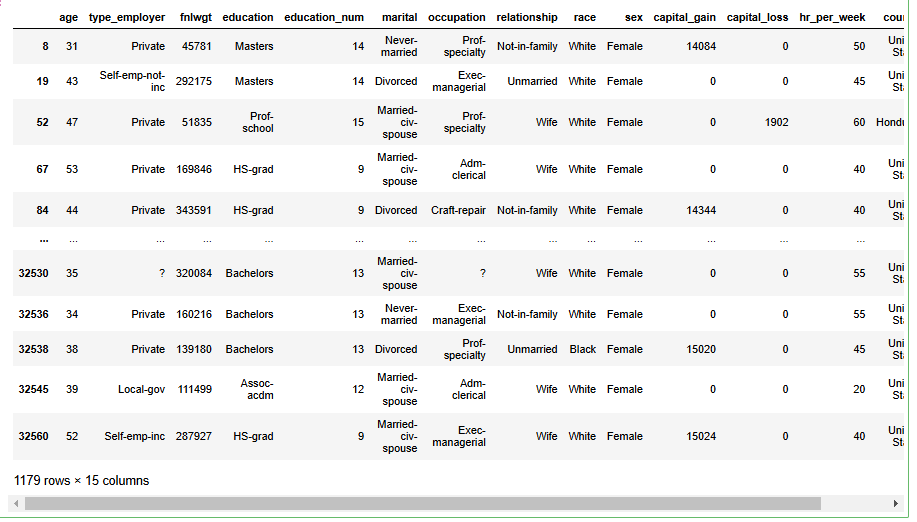
ml1



fm = df[(df.sex =='Female') ]

fm1 = df[(df.sex == 'Female') & (df.income == '>50K\n')]

fm1



**Exploratory Data Analysis**

**Summarizing the Data**

df1 = df[(df.income== '>50K\n') ]

print ('The rate of people with high income is: ',int( len(df1)/ float ( len(df))\*100), '%.')

print( 'The rate of men with high income is: ',int( len(ml1)/ float ( len(ml))\*100), '%.')

print ('The rate of women with high income is: ',int( len(fm1)/ float ( len(fm))\*100), '%.')



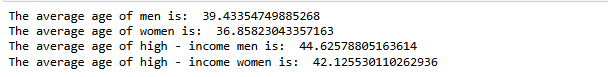
**Mean**

print('The average age of men is: ',ml['age'].mean())

print ('The average age of women is: ',fm['age'].mean())

print ('The average age of high - income men is: ',ml1['age'].mean())

print( 'The average age of high - income women is: ',fm1['age'].mean())



**Sample Variance**

ml\_mu = ml['hr\_per\_week'].mean()

fm\_mu = fm['hr\_per\_week'].mean()

ml\_var = ml['hr\_per\_week'].var()

fm\_var = fm['hr\_per\_week'].var()

ml\_std = ml['hr\_per\_week'].std()

fm\_std = fm['hr\_per\_week'].std()

print( 'Statistics of age for men: mu:',ml\_mu , 'var:', ml\_var , 'std:', ml\_std)

print ('Statistics of age for women: mu:',fm\_mu , 'var:', fm\_var , 'std:', fm\_std)



**Sample Median**

ml\_median = ml['age'].median()

fm\_median = fm['age'].median()

print( "Median age per men and women: ",ml\_median , fm\_median)

ml\_median\_age = ml1['age'].median()

fm\_median\_age = fm1['age'].median()

print( "Median age per men and women with high - income: ", ml\_median\_age , fm\_median\_age)



**Data Distribution**

**Frequency Distribution**

ml\_age = ml['age']

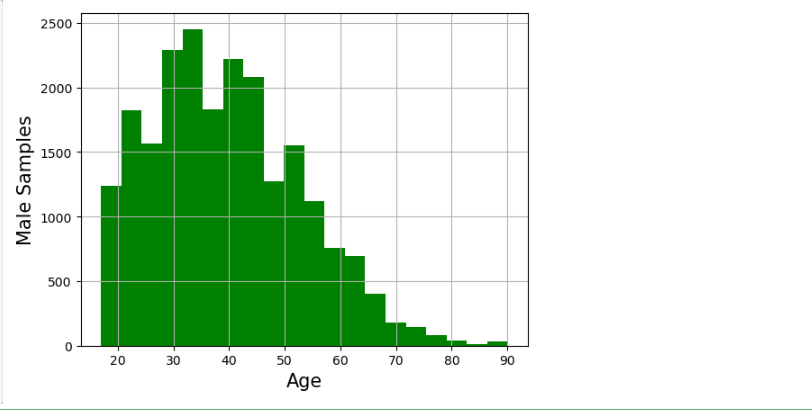
ml\_age.hist(histtype = 'stepfilled',bins = 20, color = "green")

from matplotlib import pyplot as plt

plt.xlabel('Age',fontsize=15)

plt.ylabel('Male Samples',fontsize=15)

plt.show();



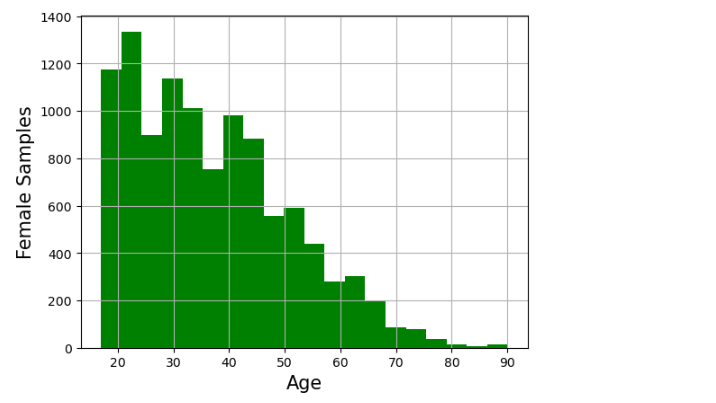
fm\_age = fm['age']

fm\_age.hist( histtype = 'stepfilled',bins = 20, color = "green")

plt.xlabel('Age',fontsize=15)

plt.ylabel('Female Samples',fontsize=15)

plt.show();



import seaborn as sns

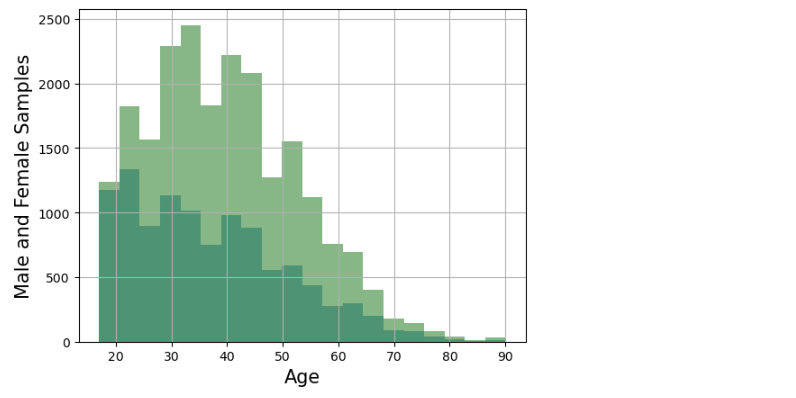
fm\_age.hist( histtype = 'stepfilled',alpha = .5, bins = 20)

ml\_age.hist( histtype = 'stepfilled',alpha = .5, color = sns.desaturate("green",.75),bins = 20)

plt.xlabel('Age',fontsize=15)

plt.ylabel('Male and Female Samples',fontsize=15)

plt.show();



**OR THIS CODE**

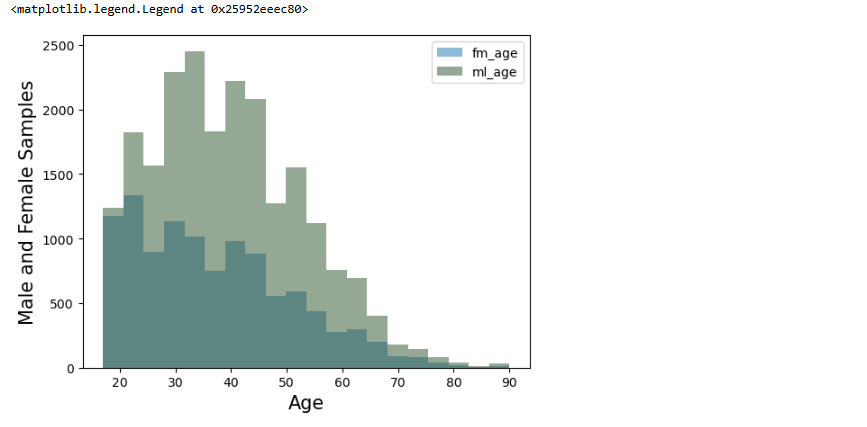
plt.hist(fm\_age, histtype='stepfilled', alpha=0.5, bins=20, label='fm\_age')

plt.hist(ml\_age, histtype='stepfilled', alpha=0.5, color=sns.desaturate("green", 0.3), bins=20, label='ml\_age')

plt.xlabel('Age',fontsize=15)

plt.ylabel('Male and Female Samples',fontsize=15)

plt.legend()



import matplotlib.pyplot as plt

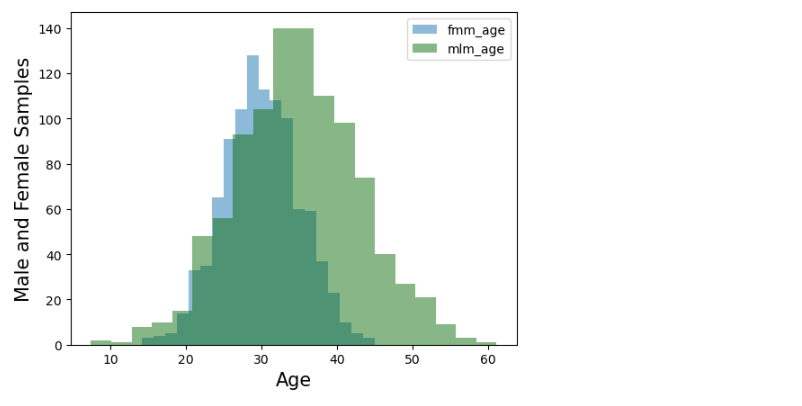
import seaborn as sns

import numpy as np

# Generate some example data

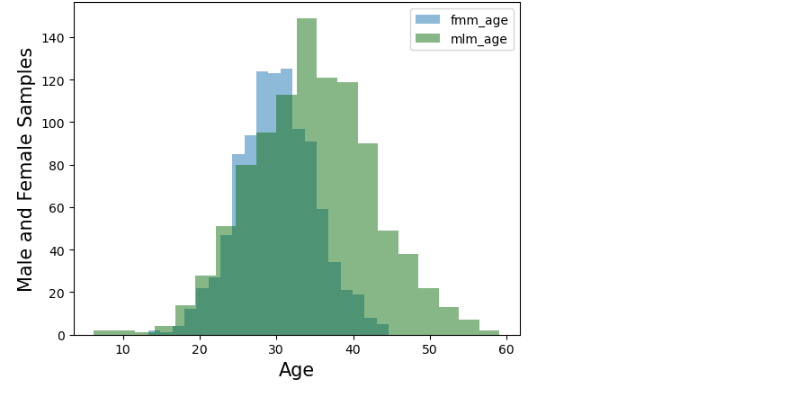
fmm\_age = np.random.normal(loc=30, scale=5, size=1000)

mlm\_age = np.random.normal(loc=35, scale=8, size=1000)



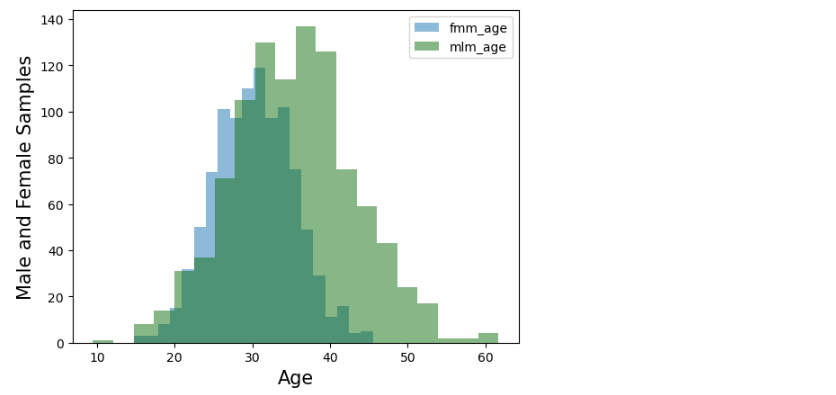
# Create a histogram for fm\_age

plt.hist(fmm\_age, histtype='stepfilled', alpha=0.5, bins=20, label='fmm\_age')



# Create a histogram for ml\_age with a desaturated green color

plt.hist(mlm\_age, histtype='stepfilled', alpha=0.5, color=sns.desaturate("green", 0.75), bins=20, label='mlm\_age')

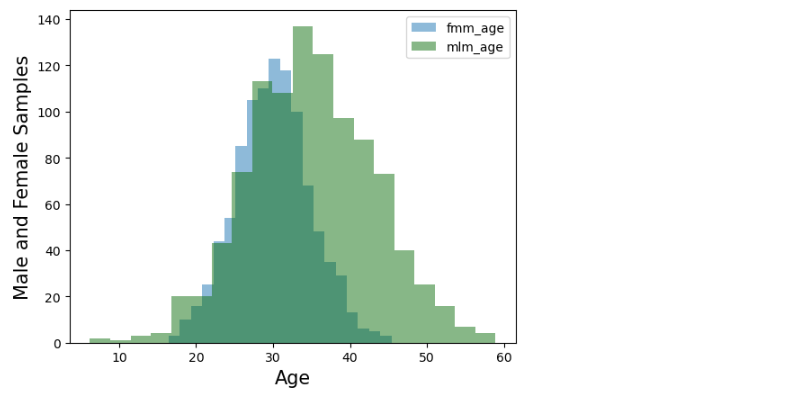


# Set plot properties

plt.xlabel('Age', fontsize=15)

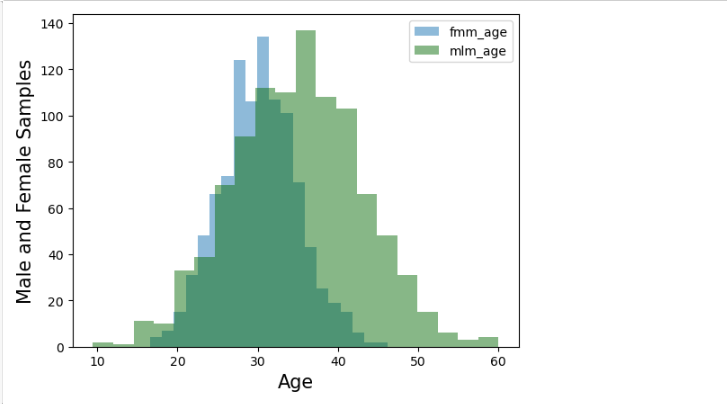
plt.ylabel('Male and Female Samples', fontsize=15)

plt.legend() # Add legend to distinguish between datasets



# Show the plot

plt.show()



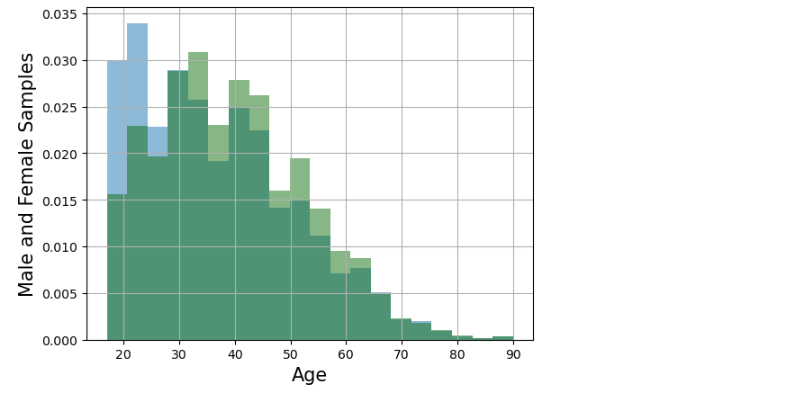
fm\_age.hist( density = True,histtype = 'stepfilled',alpha = .5, bins = 20)

ml\_age.hist(density = True,histtype = 'stepfilled',alpha = .5, bins = 20,color = sns.desaturate("green",.75))

plt.xlabel('Age',fontsize=15)

plt.ylabel('Male and Female Samples',fontsize=15)

plt.show()



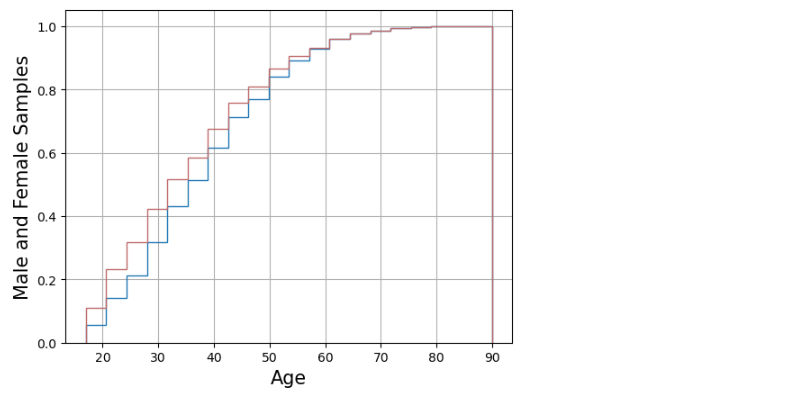
ml\_age.hist(density = True,histtype = 'step',cumulative = True ,bins = 20)

fm\_age.hist(density = True,histtype='step',cumulative = True ,bins = 20,color = sns.desaturate("indianred",.75))

plt.xlabel('Age',fontsize=15)

plt.ylabel('Male and Female Samples',fontsize=15)

plt.show();



**Outlier Treatment**

df2 = df.drop(df.index[

(df.income == '>50K\n') &

(df['age'] > df['age'].median() + 35) |

(df['age'] < df['age'].median() -15)

])

ml1\_age = ml1['age']

fm1\_age = fm1['age']

ml2\_age = ml1\_age. drop( ml1\_age.index[

(ml1\_age > df['age'].median() + 35) &

(ml1\_age > df['age'].median() - 15)

])

fm2\_age = fm1\_age. drop( fm1\_age.index[

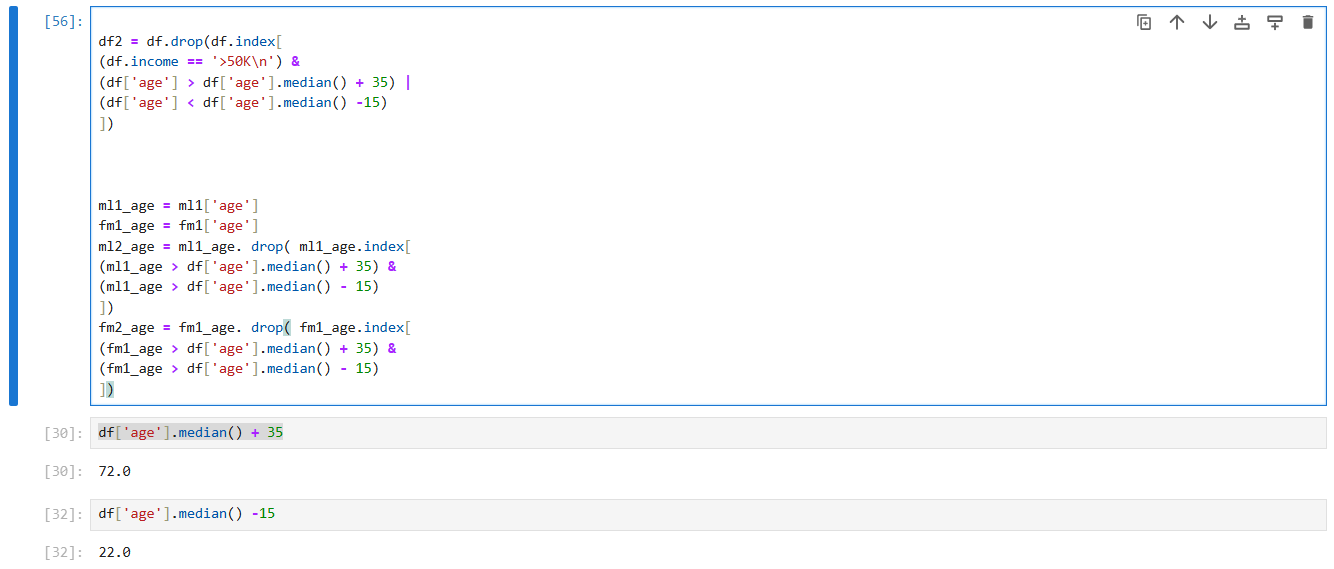
(fm1\_age > df['age'].median() + 35) &

(fm1\_age > df['age'].median() - 15)

])

df['age'].median() + 35

df['age'].median() -15



mu2ml = ml2\_age .mean()

std2ml = ml2\_age.std()

md2ml = ml2\_age.median()

mu2fm = fm2\_age .mean()

std2fm = fm2\_age.std()

md2fm = fm2\_age.median()

print ("Men statistics:")

print( "Mean:", mu2ml , "Std:", std2ml)

print( "Median:", md2ml)

print ("Min:", ml2\_age. min (), "Max:", ml2\_age.max ())

print ("Women statistics:")

print ("Mean:", mu2fm , "Std:", std2fm)

print ("Median:", md2fm)

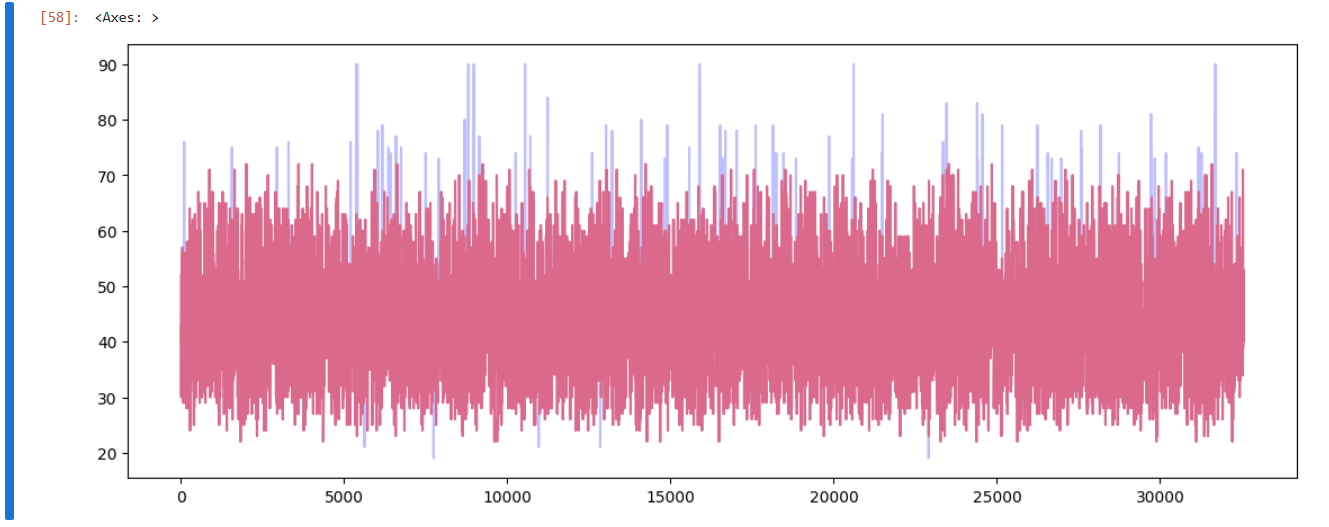
print( "Min:", fm2\_age. min (), "Max:", fm2\_age.max ())



plt.figure(figsize = (13.4, 5))

df.age[(df.income == '>50K\n')].plot(alpha = .25, color = 'blue')

df2.age[(df2.income == '>50K\n')].plot(alpha = .45, color = 'red')



print ('The mean difference with outliers is: %4.2f'% (ml\_age.mean() - fm\_age.mean()))

print ('The mean difference without outliers is:%4.2f.'% (ml2\_age. mean() - fm2\_age. mean()))



import numpy as np

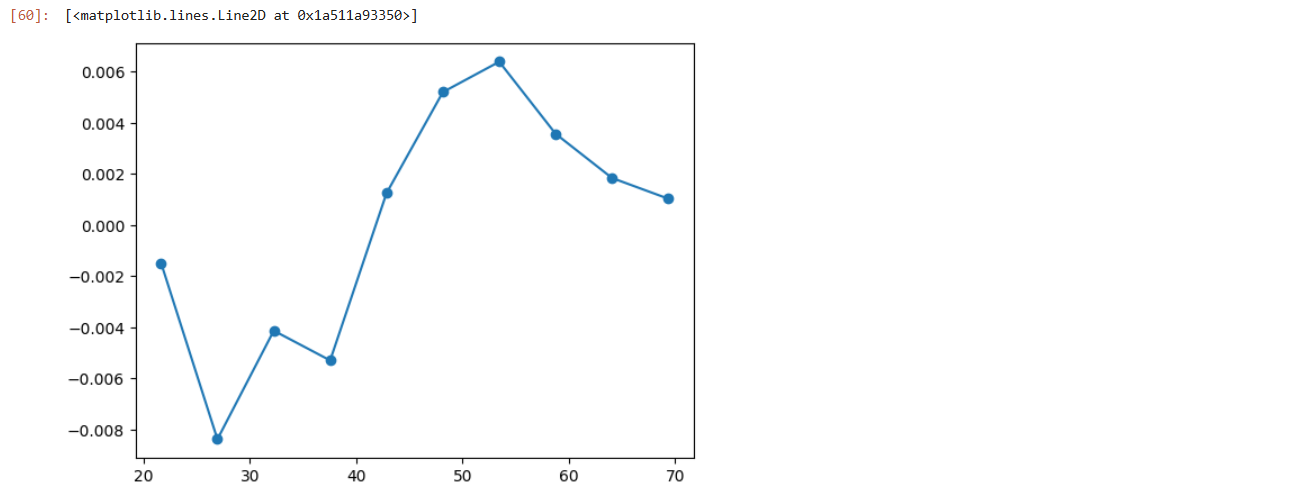
countx , divisionx = np.histogram( ml2\_age , density =True)

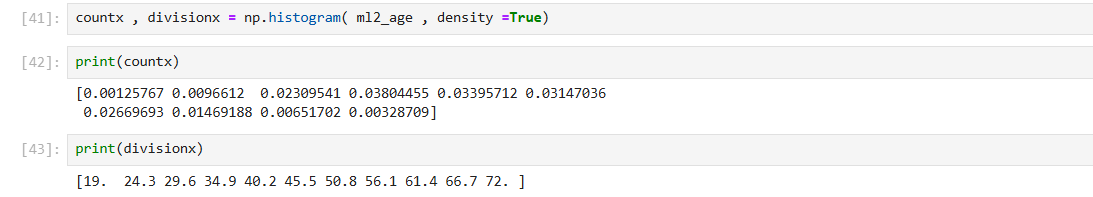
county , divisiony = np.histogram( fm2\_age , density =True)

val = [(divisionx[i] + divisionx[i +1])/2

for i in range ( len(divisionx) - 1)]

plt.plot(val , countx - county , 'o-')





**Measuring Asymmetry : Skewness and Pearson’s Median Skewness Coefficient**

def skewness(x):

res = 0

m = x.mean()

s = x.std()

for i in x:

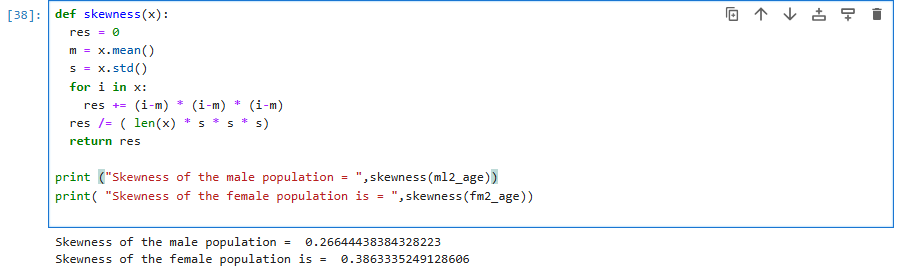
res += (i-m) \* (i-m) \* (i-m)

res /= ( len(x) \* s \* s \* s)

return res

print ("Skewness of the male population = ",skewness(ml2\_age))

print( "Skewness of the female population is = ",skewness(fm2\_age))



def pearson(x):

return 3\*(x.mean() - x.median())\*x.std()

print ("Pearson’s coefficient of the male population= ",pearson(ml2\_age))

print ("Pearson’s coefficient of the female population = ",pearson(fm2\_age))

