SRI GURU TEGH BAHADUR INSTITUTE OF MANAGEMENT AND INFORMATION TECHNOLOGY

**(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY)**



**BACHELOR OF COMPUTER APPLICATION**

# SESSION: 2021-2024

**DATA SCIENCE PROGRAMMING**

## SUBMITTED TO: SUBMITTED BY:

RITINDER KAUR GORANG AGGARWAL ASSISTANT PROFESSOR BCA 4A

02490202021

## 

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Title** | **Page**  **No.** | **Remarks** |
| 1. | Create 2 numpy arrays with 5 elements using arange () and linspace () and display the implement the concept of slicing on them. | 5 |  |
| 2. | Create two 2-d arrays and perform addition, subtraction and multiplication on these arrays. Print the value of bordered elements of both matrices. | 6 |  |
| 3. | Create two pandas series from a dictionary of values and an ndarray and display the values from 2nd to 5th index. Print the index, minimum and maximum values in the first series. | 8 |  |
| 4. | Create a Series and print all the elements that are above 75th percentile. Print minimum ,maximum and sum of series using aggregate() | 10 |  |
| 5. | Series objects Temp1, temp2, temp3, temp 4 stores the temperature of days of week 1, week 2, week 3, week 4. Write a script to:-  a. Print average temperature per week  b. Print average temperature of entire month | 11 |  |
| 6. | Create a series containing 2 NaN values. Perform check for null values and then replace the null values with 0.perform any 5 statistical functions on the series. | 12 |  |
| 7. | Two Series object, Population stores the details of four metro cities of India and another object AvgIncome store the total average income reported in four years in these cities. Calculate income per capita for each of these metro cities. | 13 |  |
| 8. | Given two series S1 and S2.  S1 S2  A 39 A 10  B 41 B 10  C 42 D 10  D 44 F 10  Find the output for following python pandas statements?  a. S1[ : 2]\*100 b. S1 \* S2 c. S2[ : : -1]\*10 | 14 |  |
| 9. | Write a program to create a Series having 10 random numbers in the range of 10 and 20 | 16 |  |
| 10. | Consider a series object s10 that stores the number of students in each section of class 12 as shown below. First two sections have been given task for selling tickets @ Rs.100/- per ticket as a part of social experiment. Write code to create the series and display how much section A and B have collected. A-39, B- 31, C- 32, D- 34, E- 35 | 17 |  |
| 11. | Write a program to create a DataFrame to store weight, age and name of three people. Print the DataFrame and its transpose. Add 5 rows in the dataframe through code. Rename the Weight column as Wgt. And then print the index names, column names and total amount of data. Sort the dataframe on the basis of age. | 18 |  |
| 12. | Create a DataFrame having age, name, weight of five students. Print the dataframe using head (). Modify the weight of student in first and 4th row. Display only the weight of first and fourth rows before and after modification. | 20 |  |
| 13. | Create a DataFrame based on E-Commerce data and generate mean, mode, and median. | 21 |  |
| 14. | Write a Program to create a CSV file with student data containing 10 rows. Create its DataFrame and use describe () to display its statistics. Write all the steps and definition of all the statistical functions displayed. | 22 |  |
| 15. | Consider the DataFrame QtrSales where each row contains the item category, item name and expenditure and group the rows by category, and print the average expenditure per category | 24 |  |
| 16. | Write a program to implement pivot() and pivot-table() on a DataFrame | 25 |  |
| 17. | Write a program to find mean absolute deviation on a DataFrame. | 26 |  |
| 18. | Create a DataFrame based on employee data and generate quartile and variance. | 27 |  |
| 19. | Program to implement Skewness on Random data. | 28 |  |
| 20. | Create a DataFrame on any Data and compute statistical function of Kurtosis. | 29 |  |
| 21. | CASE STUDY ON :CRIME\_BY\_STATE\_RT.CSV  [crime\_by\_state\_rt.zip](file:///C:\Users\hp\Downloads\crime_by_state_rt.zip)  21.1: Find the 75% percentile for data  21.2: Apply aggregate function to find count, min , max and sum  21.3: Perform group by on State and Year  21.4: Calculate the number of entries year wise  21.5: Create a pie chart for all states  21.6: Plot the no. of murders state-wise  21.7: Check the data type of all column  21.8: Check the robbery column for NULL values.  if exist replace with 0  21.9: Select the data set where robbery greater than 10 and year  = 2001  21.10: describe () the dataset  21.11: Perform sorting on Arson and Robbery  21.12: Perform renaming on two columns | 30 |  |
| 22. | CASE STUDY ON: investigating clinical data  [investigating clinical data.csv](investigating%20clinical%20data.csv)  22.1: describe () the dataset  22.2: Perform renaming on two columns  22.3: Is there any relation between the age of the patient and having a delay on the date of appointment.  22.4:  Is there any relation between gender and delaying appointments? (Show by plotting bar graph)  22.5: Do people delay their appointments when they have no scholarship? (Show by plotting)  22.6: Is there any relationship between gender having scholarships and delaying the appointment? (show by plotting)  22.7: Which gender has more appointments in which illness? (Show by plotting)  22.8: Does a person suffering from alcoholism tend to delay appointments? (Show by plotting)  22.9: Perform group by and count on neighborhood  22.10: Perform agg () function on dataset. |  |  |
| 23. | CASE STUDY ON: Credit card transaction  [Credit card transactions - India - Simple.csv](Credit%20card%20transactions%20-%20India%20-%20Simple.csv)  23.1: Show the head and visualize the basic state for all columns.  23.2: Display pivot table for create summary for total amount spread  a) month b) city  23.3: Analyze the impact of gender to study consumer behavior  23.4: Check for NULL values in columns .If exist replaces them with appropriate values.  23.5: Analyze the relationship type between expense type and amount.  23.6: Analyze the spending habits by city and gender.  23.7: implement Skewness and kurtosis on data  Draw plots along with appropriate question |  |  |
| 24. | CASE STUDY ON PROJECT TOPIC (Global superstore) |  |  |
| 25. | RESEACH PAPER |  |  |

**QUESTION 1:*–***

## Create 2 numpy arrays with 5 elements using arange () and linspace () and display the implement the concept of slicing on them.

**CODING:**

## import numpy as np

## array1 = np.arange (10, 25, 3)

## print ("array1:", array1)

## print (array1[1:4])

## print (array1[3 :])

## print (array1[:3])

## print (array1[-3:-1])

## print (array1[1:5:2])

## print ("")

## array2 = np.linspace (0, 10, 5)

## print ("array2:", array2)

## print (array2[1:4])

## print (array2[3 :])

## print (array2[:3])

## print (array2[-3:-1])

## print (array2[1:5:2])

**OUTPUT:**

## 

**QUESTION 2:*–***

**Create two 2-d arrays and perform addition, subtraction and multiplication on these arrays. Print the value of bordered elements of both matrices.**

**CODING:**

import numpy as np

array1 = np.array ([[11, 22, 33], [44, 55, 66], [77, 88, 99]])

array2 = np.array ([[1, 2, 2], [4, 5, 6], [7, 8, 9]])

addition = array1 + array2

subtraction = array1 - array2

multiplication = array1 \* array2

bordered\_elements\_array1 = array1[[0, -1],:]

bordered\_elements\_array2 = array2[:, [0, -1]]

print ("Array 1:", array1)

print ("\nBordered elements of Array 1:”, bordered\_elements\_array1)

print ("\nArray 2:”, array2)

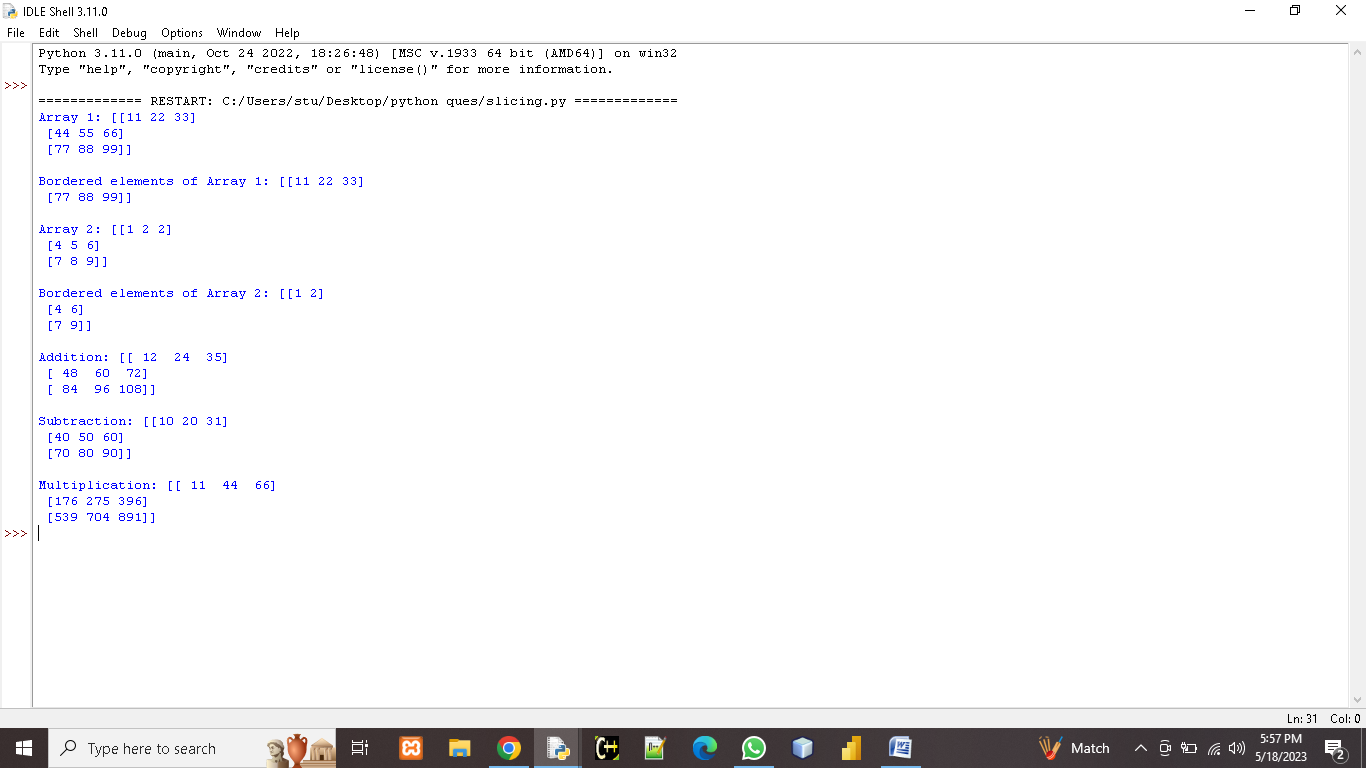
print ("\nBordered elements of Array 2:", bordered\_elements\_array2)

print ("\nAddition:", addition)

print ("\nSubtraction:", subtraction)

print ("\nMultiplication:", multiplication)

**OUTPUT:**



**QUESTION 3:*–***

**Create two pandas series from a dictionary of values and an ndarray and display the values from 2nd to 5th index. Print the index, minimum and maximum values in the first series.**

**CODING:**

import pandas as pd

import numpy as np

dict\_values = {'shobit': 101, 'parth': 102, 'prerna': 103, 'muskan': 104, 'sneha': 105}

series1 = pd.Series (dict\_values)

print ("Series1:", series1)

ndarray\_values = np.array ([11, 22, 33, 44, 55])

series2 = pd.Series (ndarray\_values)

print ("\nSeries 2:", series2)

print ("\nValues from the 2nd to 5th index in Series 1 :")

print (series1[1:5])

print ("\nValues from the 2nd to 5th index in Series 2 :")

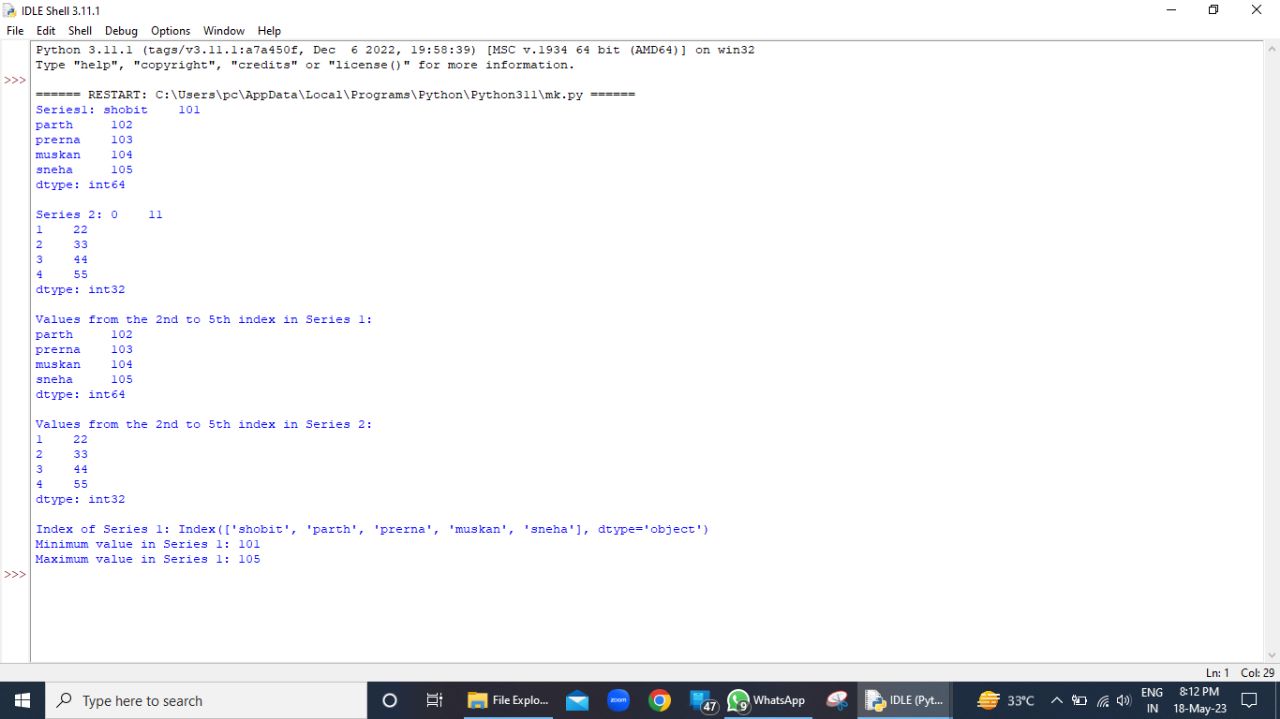
print (series2[1:5])

print ("\nIndex of Series 1:", series1.index)

print ("Minimum value in Series 1:", series1.min ())

print ("Maximum value in Series 1:", series1.max ())

**OUTPUT:**



**QUESTION 4:*–***

**Create a Series and print all the elements that are above 75th percentile. Print minimum, maximum and sum of series using aggregate ().**

**CODING:**

import pandas as pd

import numpy as np

series = pd.Series ([10, 20, 30, 40, 50, 60, 70, 80, 90, 100])

print ("Series:", series)

percentile\_75 = series.quantile (0.75)

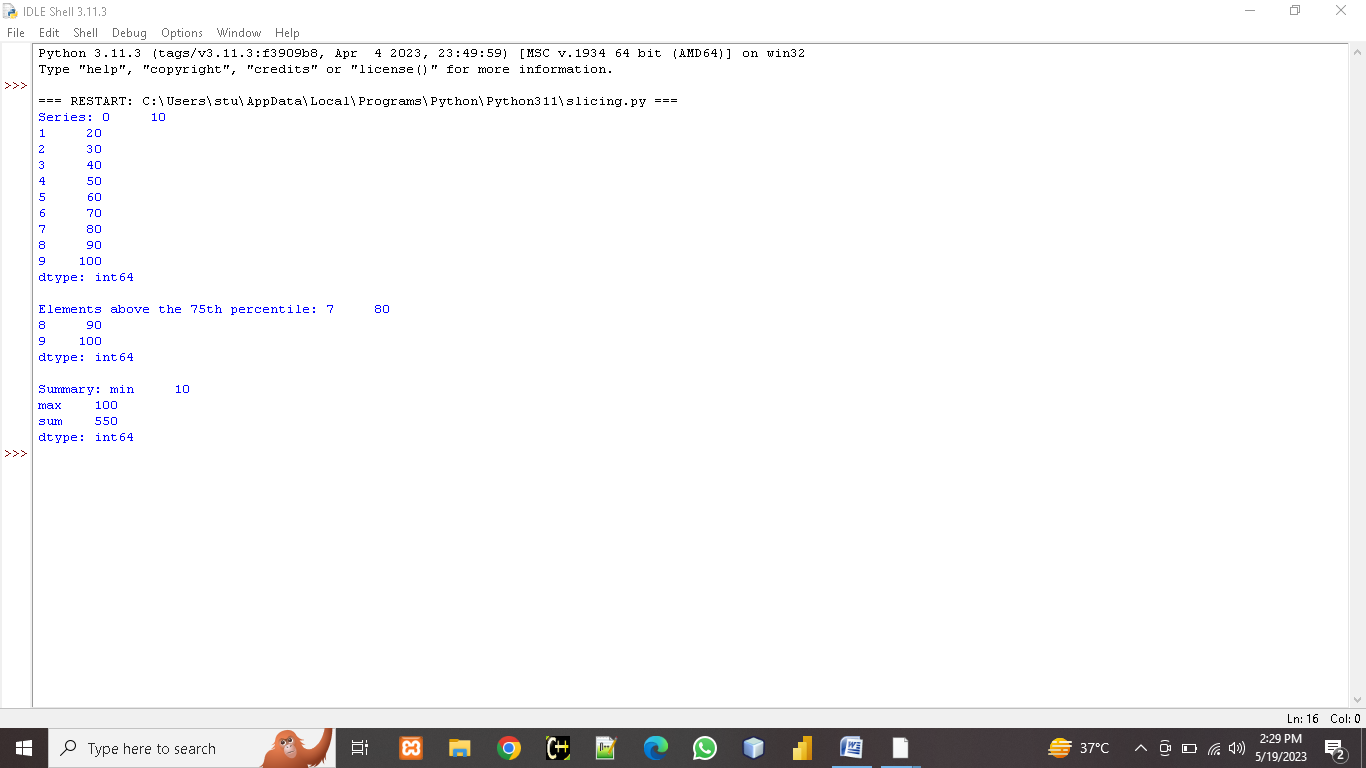
above\_percentile\_75 = series [series > percentile\_75]

print ("\n Elements above the 75th percentile:", above\_percentile\_75)

summary = series.aggregate ([min, max, sum])

print ("\nSummary:", summary)

**OUTPUT:**



**QUESTION 5:-**

**Series objects Temp1, temp2, temp3, temp 4 stores the temperature of days of week 1, week 2, week 3, week 4. Write a script to:-**

**a. Print average temperature per week**

**b. Print average temperature of entire month**

**CODING:**

import pandas as pd

week1 = [38, 38, 41, 40, 41, 42, 41]

week2 = [41, 41, 42, 41, 41, 43, 43]

week3 = [42, 42, 42, 44, 42, 41, 40]

week4 = [41, 42, 40, 41, 43, 44, 43]

temp1 = pd.Series (week1)

temp2 = pd.Series (week2)

temp3 = pd.Series (week3)

temp4 = pd.Series (week4)

weekly\_average = [temp1.mean (), temp2.mean (), temp3.mean (), temp4.mean ()]

monthly\_average = sum (weekly\_average) / len (weekly\_average)

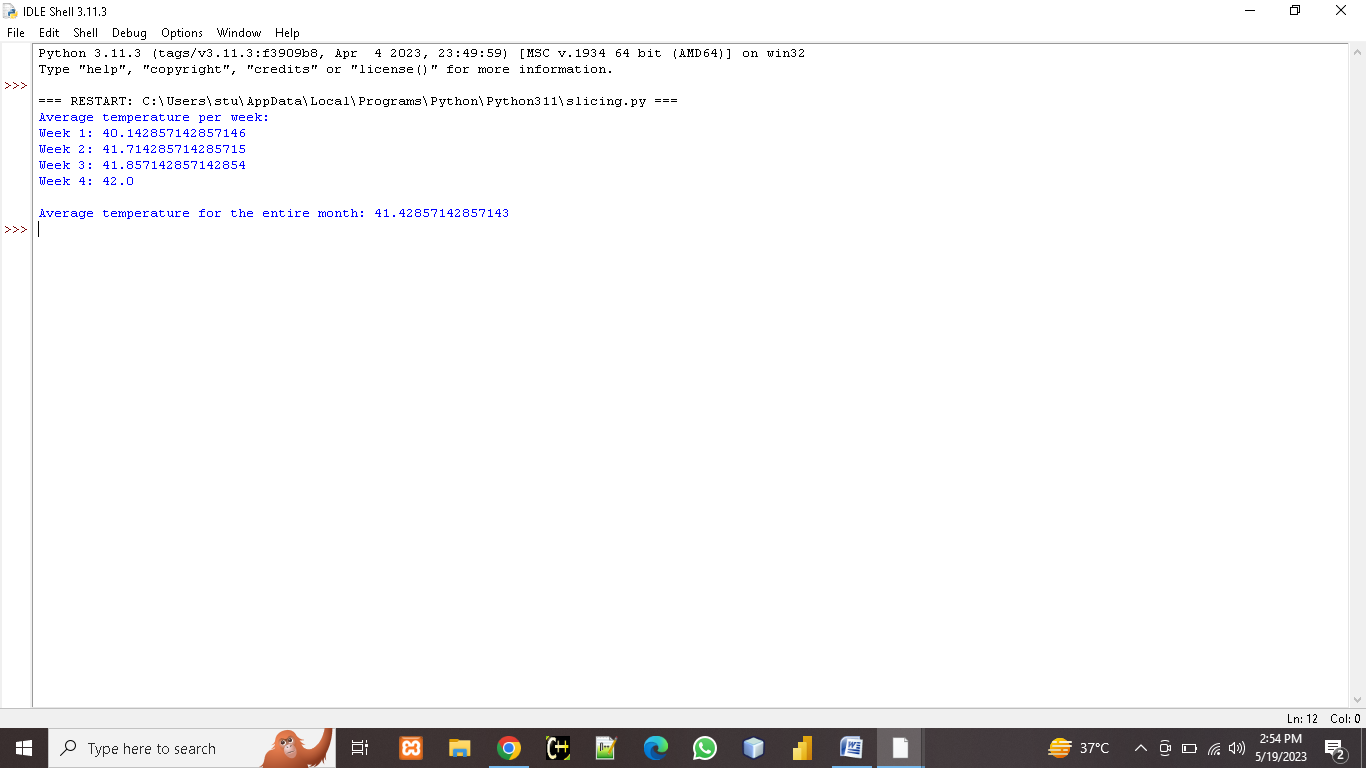
print ("Average temperature per week :")

for i, avg\_temp in enumerate (weekly\_average):

print (f" Week {i+1}:", avg\_temp)

print (f"\n Average temperature for the entire month:", monthly\_average)

**OUTPUT:**



**QUESTION 6:-**

Create a series containing 2 NaN values. Perform check for null values and then replace the null values with 0.perform any 5 statistical functions on the series.

**CODING:**

import pandas as pd

import numpy as np

data = pd.Series ([1, 2, np.nan, 3, np.nan])

print (data)

print ("Null values before replacement:", data.isnull ())

data = data.fillna (0)

print (data)

print ("Statistical functions on the Series :")

print ("Mean:", data.mean ())

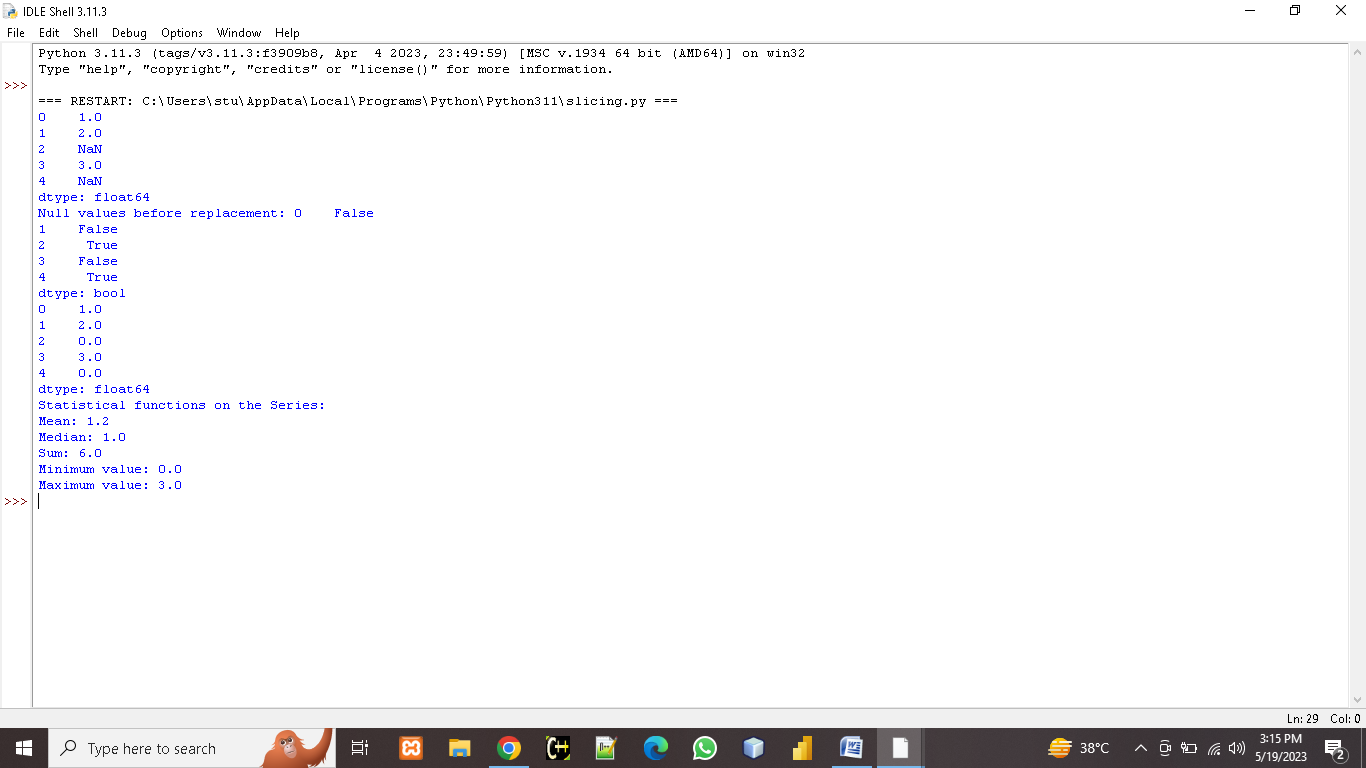
print ("Median:", data.median ())

print ("Sum:", data.sum ())

print ("Minimum value:", data.min ())

print ("Maximum value:", data.max ())

**OUTPUT:**



**QUESTION 7:-**

**Two Series object, Population stores the details of four metro cities of India and another object AvgIncome store the total average income reported in four years in these cities. Calculate income per capita for each of these metro cities.**

**CODING:**

import pandas as pd

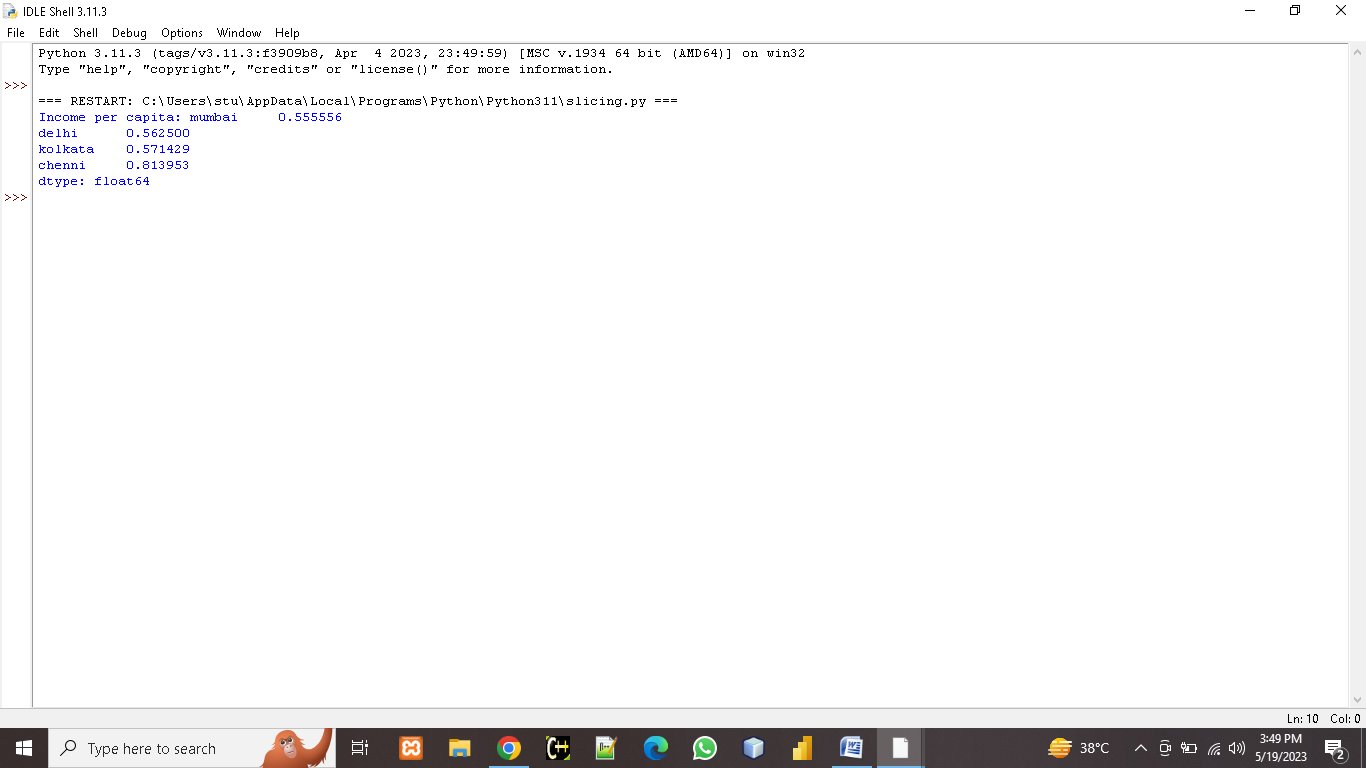
Population = pd.Series ([18000000, 16000000, 14000000, 8600000], index= ['mumbai', 'delhi', 'kolkata', 'chenni'])

AvgIncome = pd.Series ([10000000, 9000000, 8000000, 7000000], index= ['mumbai', 'delhi', 'kolkata', 'chenni'])

IncomePerCapita = AvgIncome / Population

print ("Income per capita:", IncomePerCapita)

**OUTPUT:**



**QUESTION 8:-**

Given two series S1 and S2.

S1 S2

A 39 A 10

B 41 B 10

C 42 D 10

D 44 F 10

Find the output for following python pandas statements?

a. S1 [: 2]\*100 b. S1 \* S2 c. S2 [: : -1]\*10

**CODING:**

import pandas as pd

S1 = pd.Series ([39, 41, 42, 44], index= ['A', 'B', 'C', 'D'])

S2 = pd.Series ([10, 10, 10, 10], index= ['A', 'B', 'D', 'F'])

print (S1)

print ("\n")

print (S2)

output\_a = S1 [:2] \* 100

output\_b = S1 \* S2

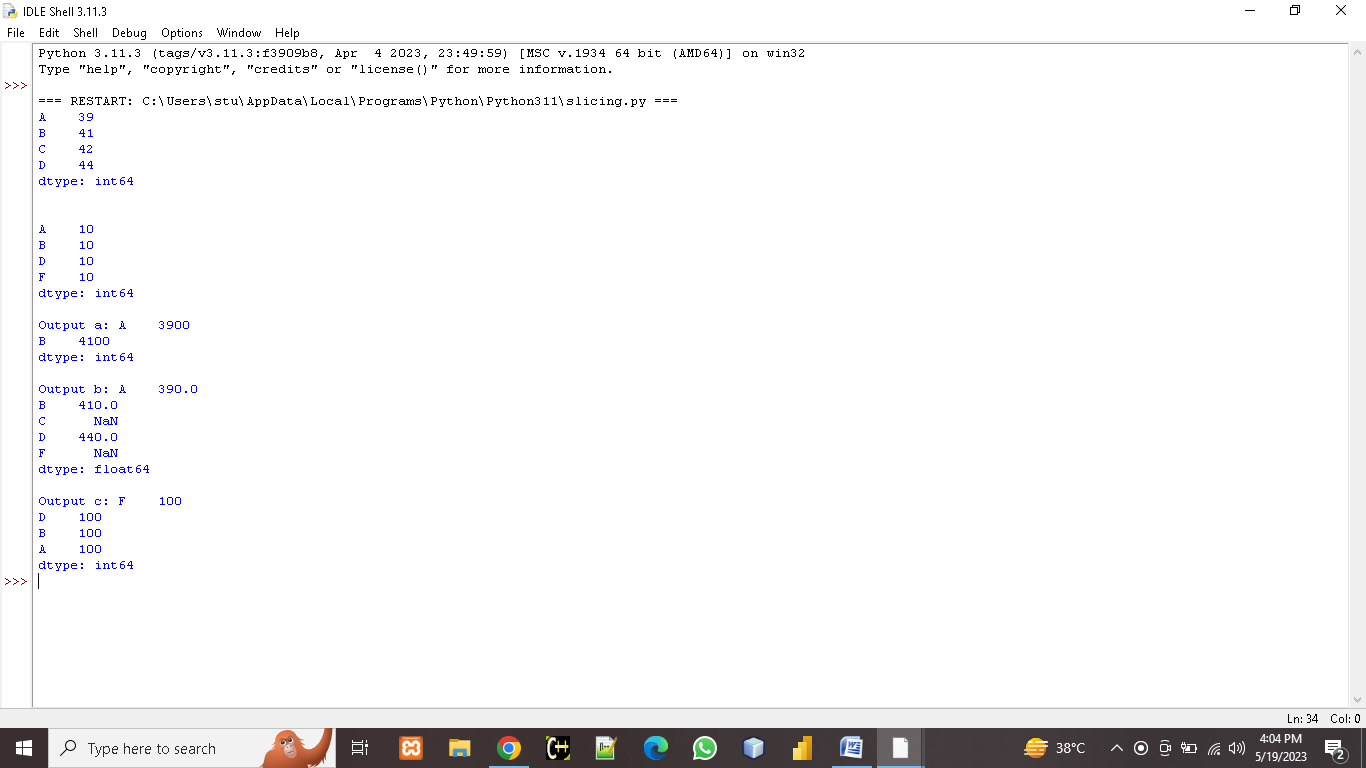
output\_c = S2 [::-1] \* 10

print ("\nOutput a:", output\_a)

print ("\nOutput b:", output\_b)

print ("\nOutput c:", output\_c)

**OUTPUT:**



**QUESTION 9:-**

Write a program to create a Series having 10 random numbers in the range of 10 and 20.

**CODING:**

import pandas as pd

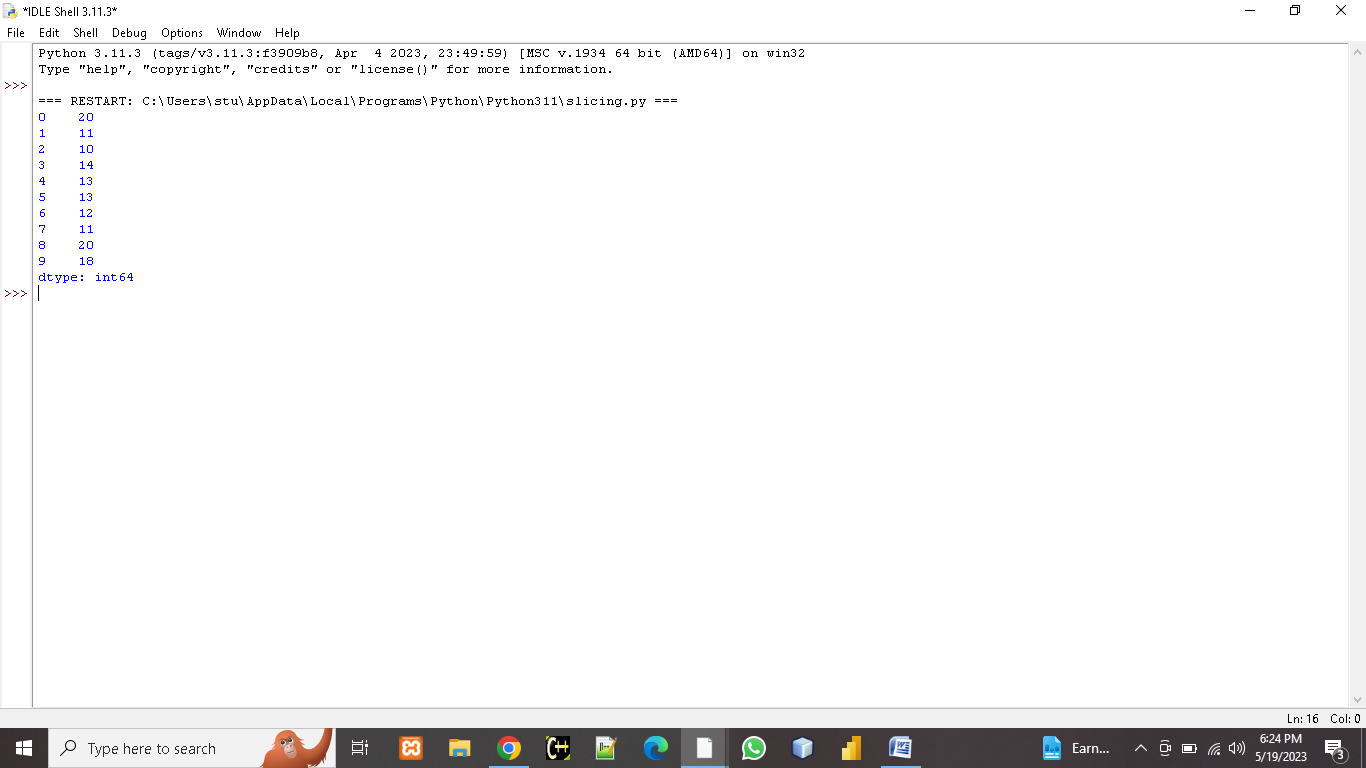
import random

random.seed (42)

series\_data = pd.Series ([random.randint (10, 20) for \_ in range (10)])

print (series\_data)

**OUTPUT:**



**QUESTION 10:-**

**Consider a series object s10 that stores the number of students in each section of class 12 as shown below. First two sections have been given task for selling tickets @ Rs.100/- per ticket as a part of social experiment. Write code to create the series and display how much section A and B have collected. A-39, B- 31, C- 32, D- 34, E- 35.**

**CODING:**

import pandas as pd

s10 = pd.Series ([39, 31, 32, 34, 35], index= ['A', 'B', 'C', 'D', 'E'])

ticket\_price = 100

amount\_collected\_A = s10['A'] \* ticket\_price

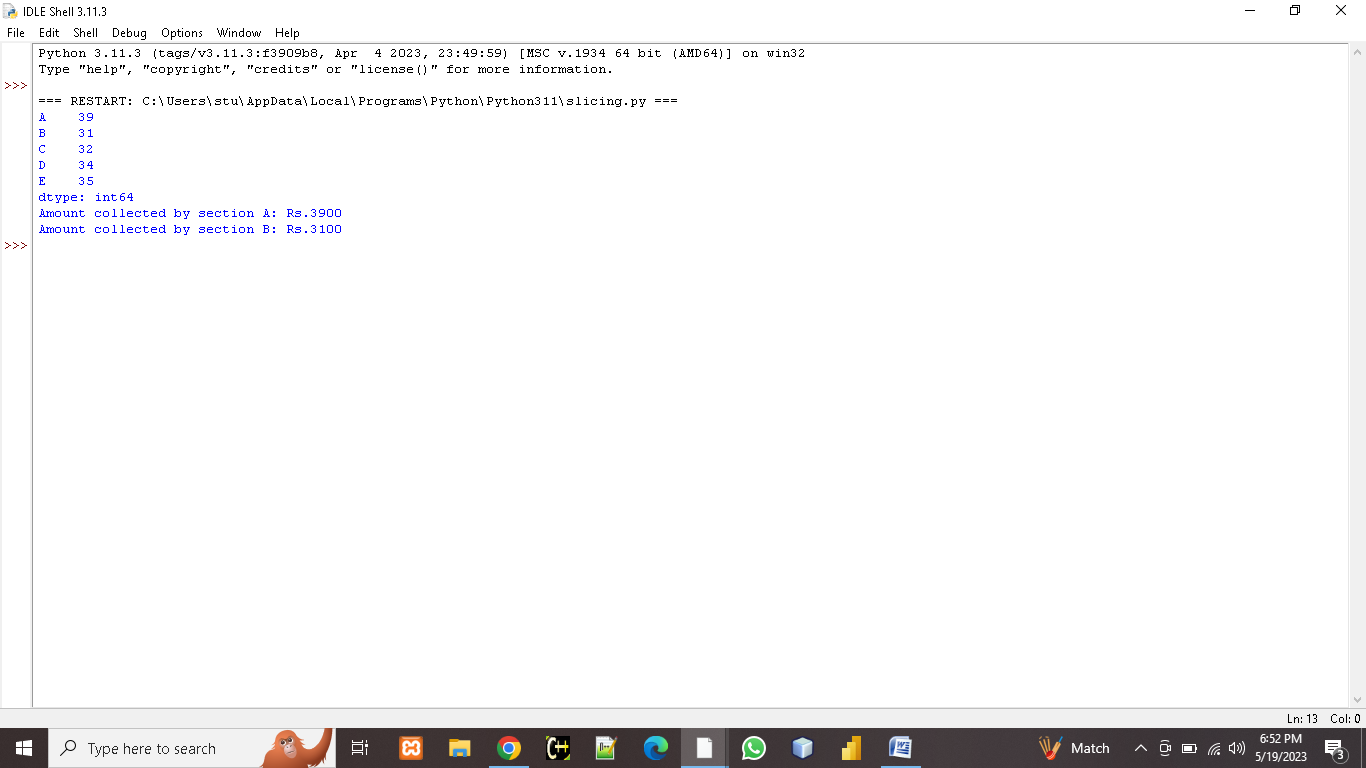
amount\_collected\_B = s10['B'] \* ticket\_price

print (s10)

print ("Amount collected by section A: Rs. {}".format (amount\_collected\_A))

print ("Amount collected by section B: Rs. {}".format (amount\_collected\_B))

**OUTPUT:**

****

**QUESTION 11:-**

**Write a program to create a DataFrame to store weight, age and name of three people. Print the DataFrame and its transpose. Add 5 rows in the dataframe through code. Rename the Weight column as Wgt. And then print the index names, column names and total amount of data. Sort the dataframe on the basis of age.**

**CODING:**

import pandas as pd

data = {'Weight': [70, 65, 75],

'Age': [25, 30, 35],

'Name': ['John', 'Jane', 'Mike']}

df = pd.DataFrame (data)

print ("Original DataFrame:", df)

print ("\n Transpose of the DataFrame:", df. transpose ())

new\_rows = [{'Weight': 80, 'Age': 40, 'Name': 'Sarah'},

{'Weight': 60, 'Age': 28, 'Name': 'Tom'},

{'Weight': 72, 'Age': 33, 'Name': 'Emily'},

{'Weight': 68, 'Age': 29, 'Name': 'Chris'},

{'Weight': 78, 'Age': 38, 'Name': 'Amy'}]

df1 = pd.DataFrame (new\_rows)

df = df.\_append (df1, ignore\_index=True)

df.rename (columns= {'Weight': 'Wgt'}, inplace=True)

print ("\n Index names:", df.index)

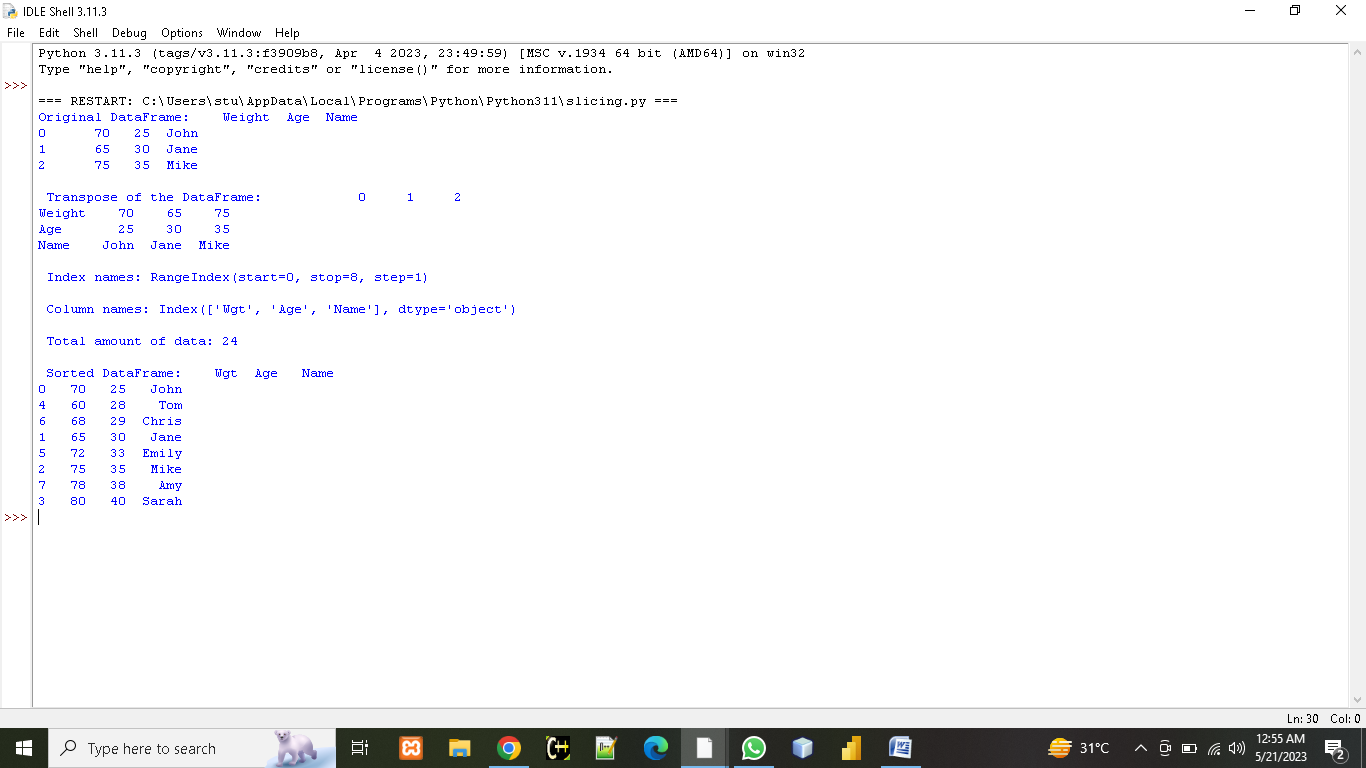
print ("\n Column names:", df.columns)

print ("\n Total amount of data:", df.size)

df\_sorted = df.sort\_values (by='Age')

print ("\n Sorted DataFrame:", df\_sorted)

**OUTPUT:**



**QUESTION 12:-**

Create a DataFrame having age, name, weight of five students. Print the dataframe using head (). Modify the weight of student in first and 4th row. Display only the weight of first and fourth rows before and after modification.

**CODING:**

import pandas as pd

data = {'Name': ['shobit', 'prerna', 'shikha', 'parth', 'sneha'],

'Age': [18, 19, 20, 21, 22],

'Weight': [65, 62, 68, 70, 72]}

df = pd.DataFrame (data)

print (df.head ())

print ("\n Weight before modification :")

print ("First row:", df.loc [0, 'Weight'])

print ("Fourth row:", df.loc [3, 'Weight'])

df.loc [0, 'Weight'] = 75

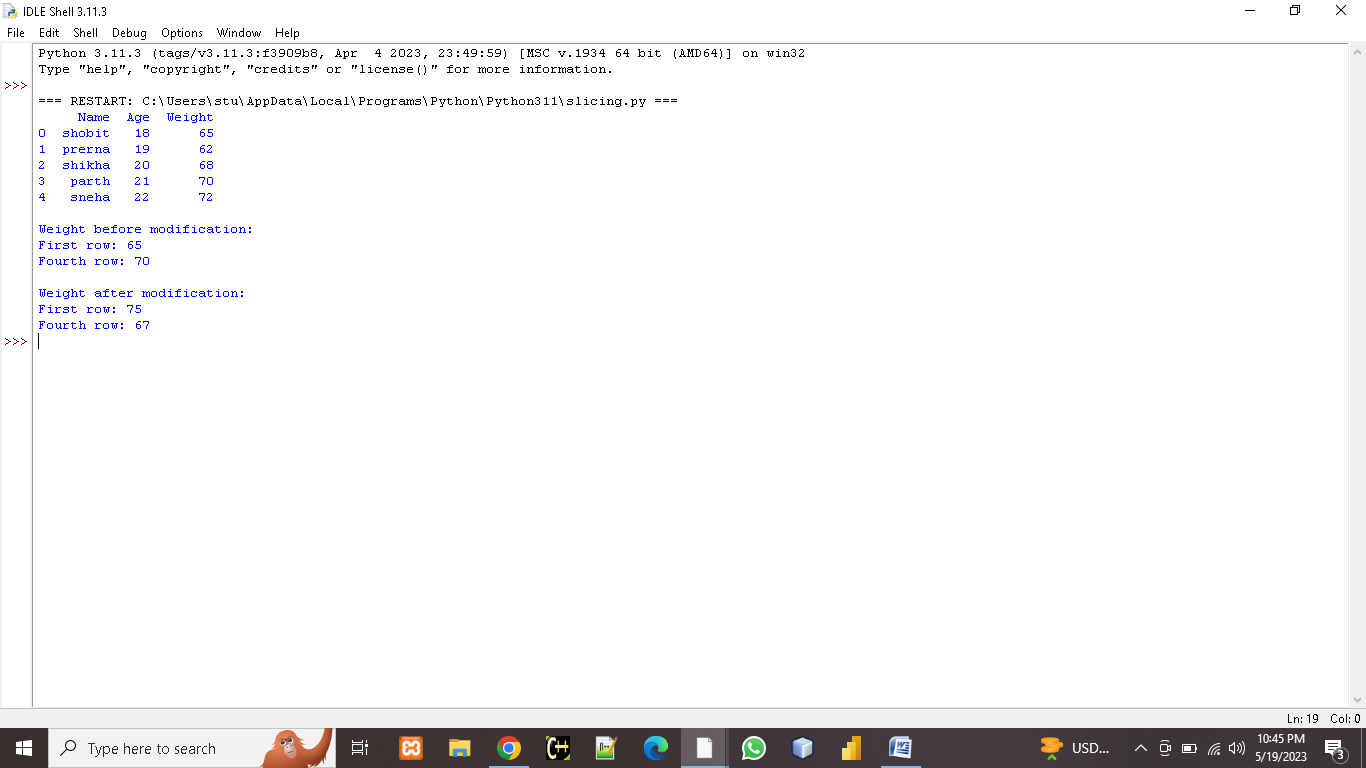
df.loc [3, 'Weight'] = 67

print ("\n Weight after modification :")

print ("First row:", df.loc [0, 'Weight'])

print ("Fourth row:", df.loc [3, 'Weight'])

**OUTPUT:**



**QUESTION 13:-**

**Create a DataFrame based on E-Commerce data and generate mean, mode, and median.**

**CODING:**

import pandas as pd

import numpy as np

data = {'Order\_ID': [101, 102, 103, 104, 105, 106],

'Customer\_Name': ['shobit', 'shikha', 'prerna', 'sneha', 'parth', 'tushar'],

'Product\_Name': ['Shirt', 'watch', 'jegging', 'top', 'Shoes', 'jeans'],

'Price': [25.99, 59.99, 15.99, 19.99, 29.99, 39.99]}

df = pd.DataFrame (data)

print (df)

mean\_price = df ['Price'].mean ()

mode\_price = df ['Price'].mode () [0]

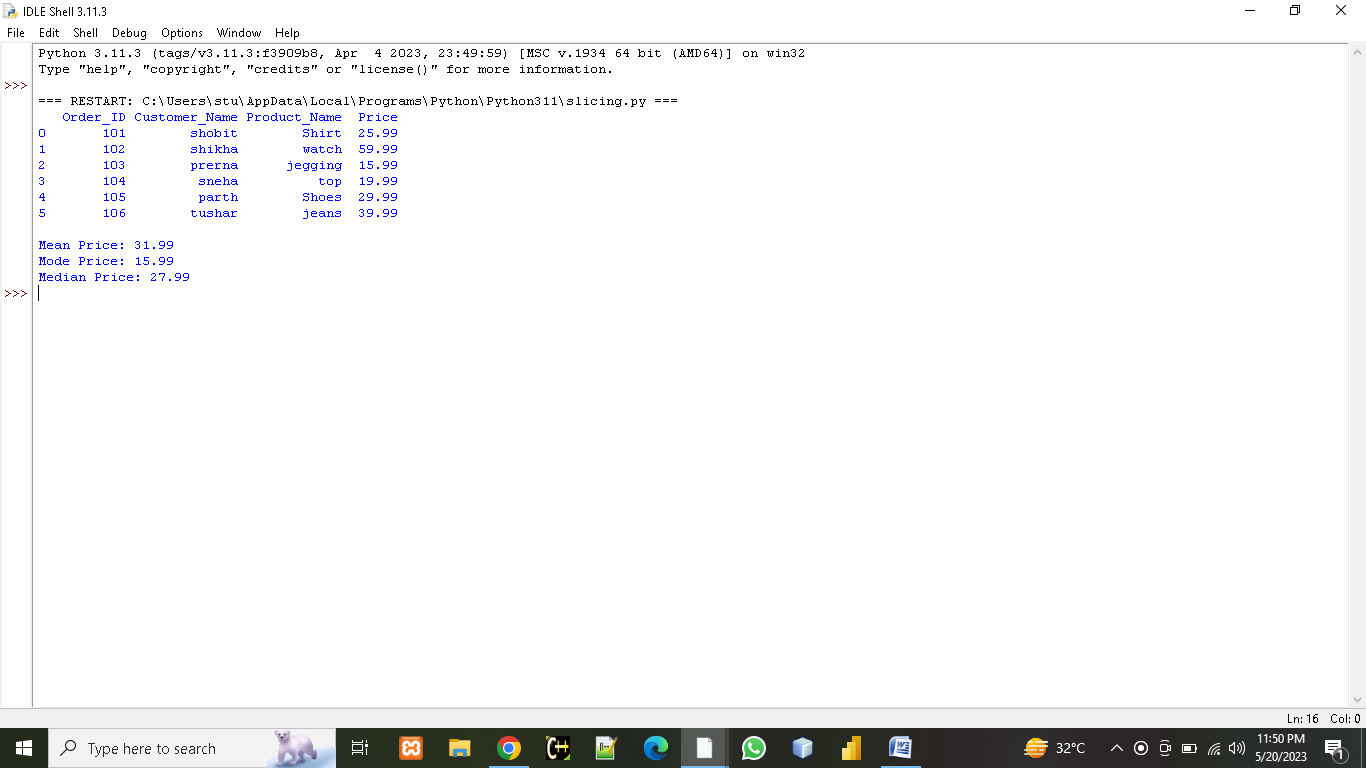
median\_price = df ['Price'].median ()

print ("\n Mean Price:", mean\_price)

print ("Mode Price:", mode\_price)

print ("Median Price:", median\_price)

**OUTPUT:**

****

**QUESTION 14:-**

**Write a Program to create a CSV file with student data containing 10 rows. Create its DataFrame and use describe () to display its statistics. Write all the steps and definition of all the statistical functions displayed.**

**CODING:**

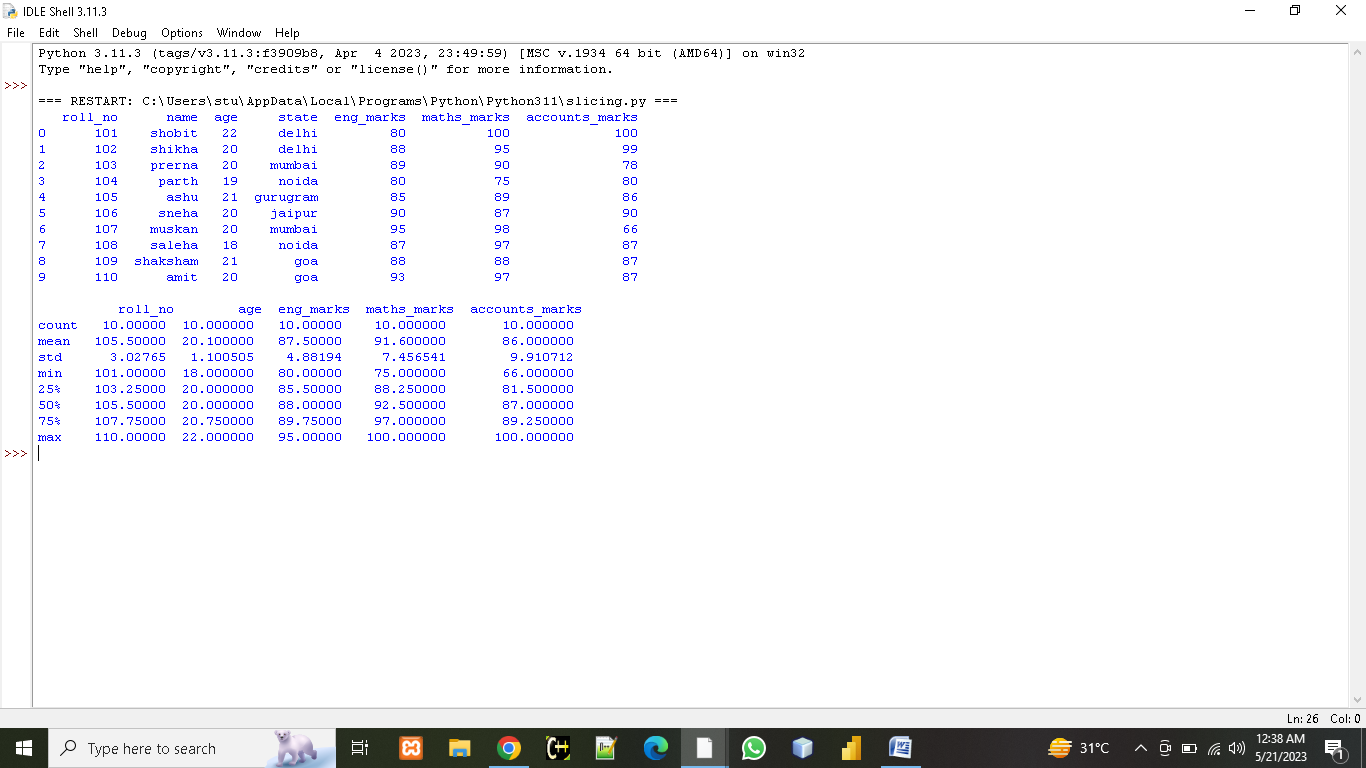
import pandas as pd

df= pd.read\_csv ("C:\\Users\\stu\\Documents\\student.csv")

print (df)

print ("\n", df.describe ())

**OUTPUT:**

****

**Steps:**

1. We create student.csv file with 7 columns roll\_no., names, age, state, eng\_marks, math\_marks and accounts\_marks.
2. We read the csv file into a DataFrame using the read\_csv () function and assign it to the variable df.
3. We use the describe () function on the DataFrame df to generate descriptive statistics. The describe () function calculates and displays statistics like count, mean, standard deviation, minimum, 25th percentile, median, 75th persentile and maximum for each numerical column in the DataFrame.
4. Then we print it using print (df).

Statistical functions displayed here are as follows:

* **Count:** Count the number of (not NULL) values in each row.
* **Mean:** Return the average (mean) value for each column.
* **Std:** Calculates the standard deviation of a dataframe or series.
* **Min:** Returns minimum value.
* **Max:** Returns maximum value.
* **25%, 50%, 75%:** These are the percentiles for the values.

**QUESTION 15:-**

**Consider the DataFrame QtrSales where each row contains the item category, item name and expenditure and group the rows by category, and print the average expenditure per category**

**CODING:**

import pandas as pd

dict= {'Item Name': ['cold coffee', 'sprite', 'chips', 'mango Juice', 'Ice Cream', 'Maggi', 'pastry', 'macroni'],

'Item Category': ['Drink', 'Drink', 'Food', 'Drink', 'Sweet', 'Food', 'Sweet', 'Food'],

'Expenditure': [200, 100, 150, 200, 120, 100, 300, 250]}

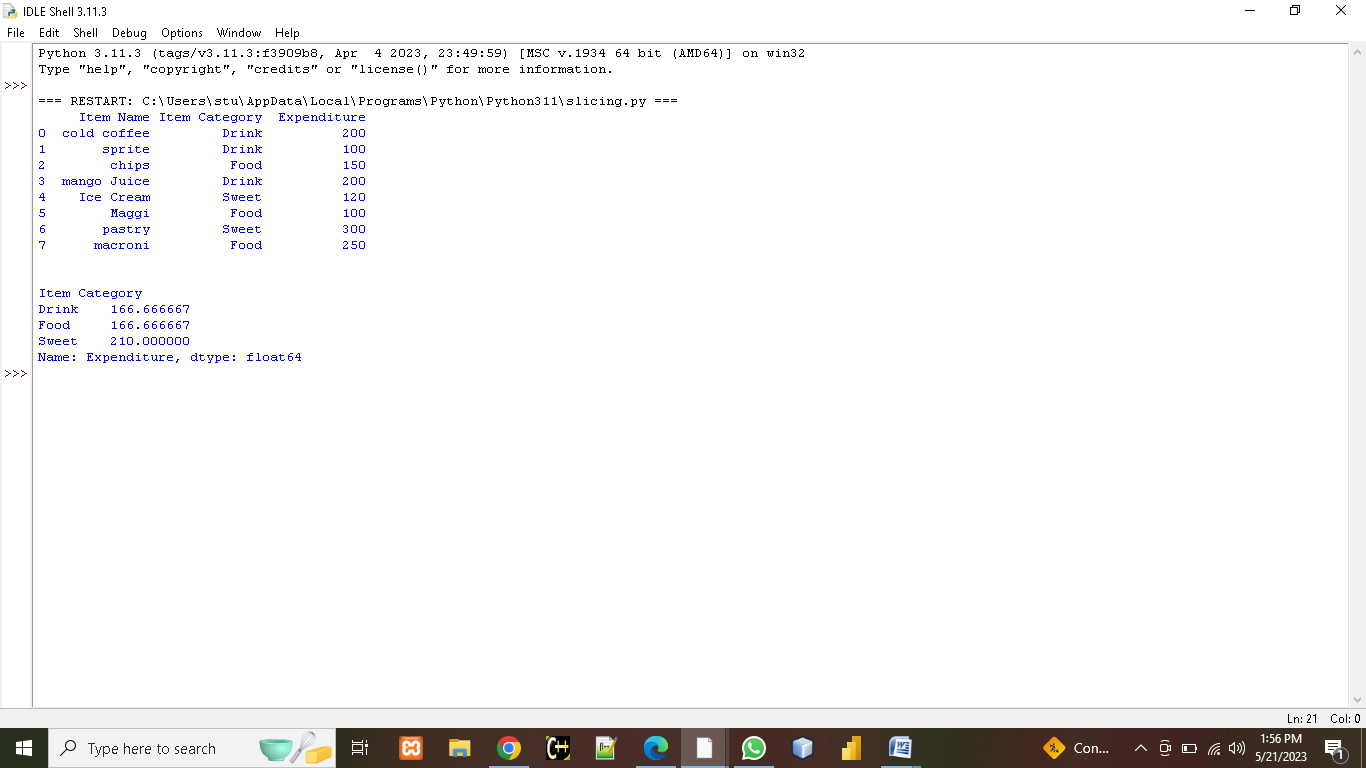
qtrsales = pd.DataFrame (dict)

print (qtrsales)

print ("\n")

print (qtrsales.groupby ('Item Category') ['Expenditure'].mean ())

**OUTPUT:**



**QUESTION 16:-**

**Write a program to implement pivot () and pivot-table () on a DataFrame.**

**CODING:**

import pandas as pd

data = {

'Name': ['chips', 'namkeen', 'cake', 'chocolate', 'cold dring'],

'City': ['New York', 'London', 'New York', 'Paris', 'London'],

'Sales': [100, 200, 150, 250, 300]

}

df = pd.DataFrame (data)

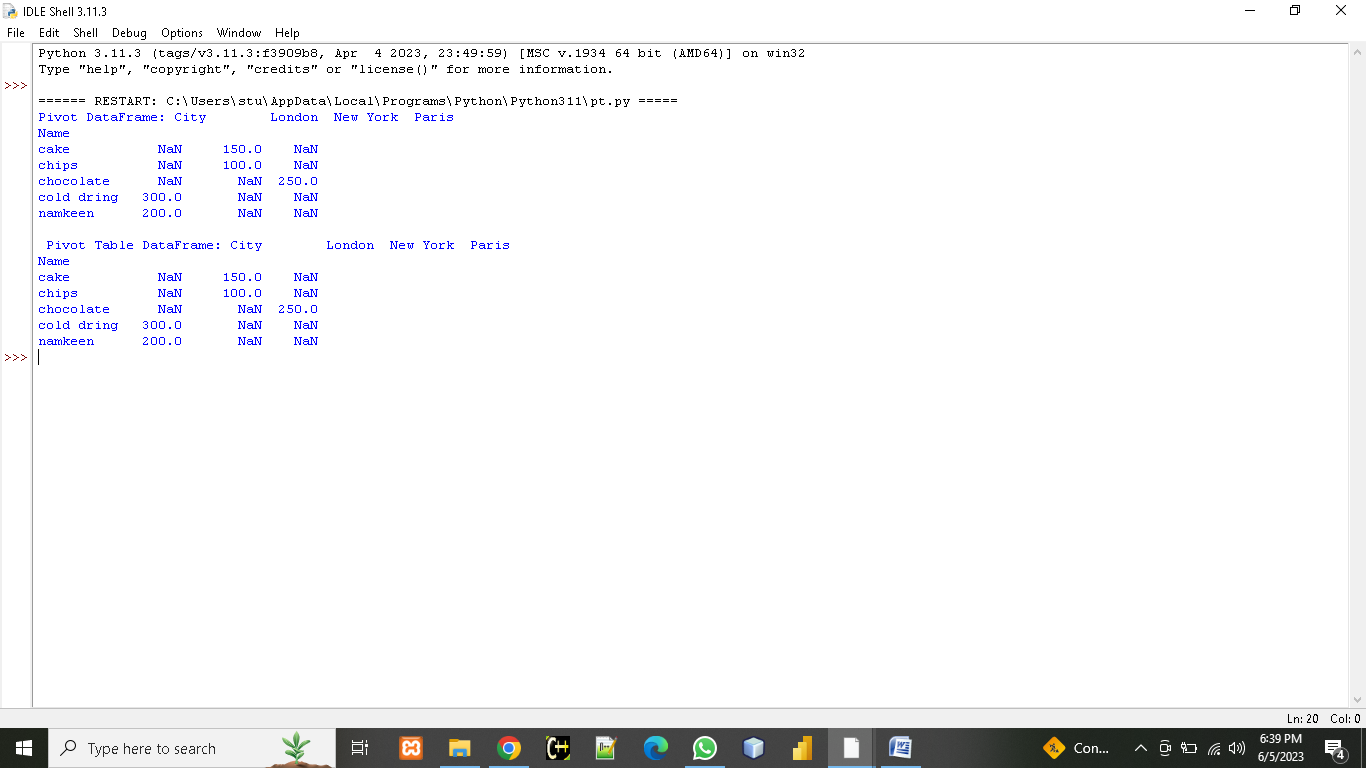
pivot\_df = df. pivot (index='Name', columns='City', values='Sales')

print ("Pivot DataFrame:", pivot\_df)

pivot\_table\_df = pd.pivot\_table (df, values='Sales', index='Name', columns='City', aggfunc='sum')

print ("\n Pivot Table DataFrame:", pivot\_table\_df)

**OUTPUT:**

****

**QUESTION 17:-**

**Write a program to find mean absolute deviation on a DataFrame.**

**CODING:**

import pandas as pd

info = {

'A': [10, 20, 30, 40, 50],

'B': [99, 88, 77, 66, 55],

'C': [100, 150, 200, 250, 300]

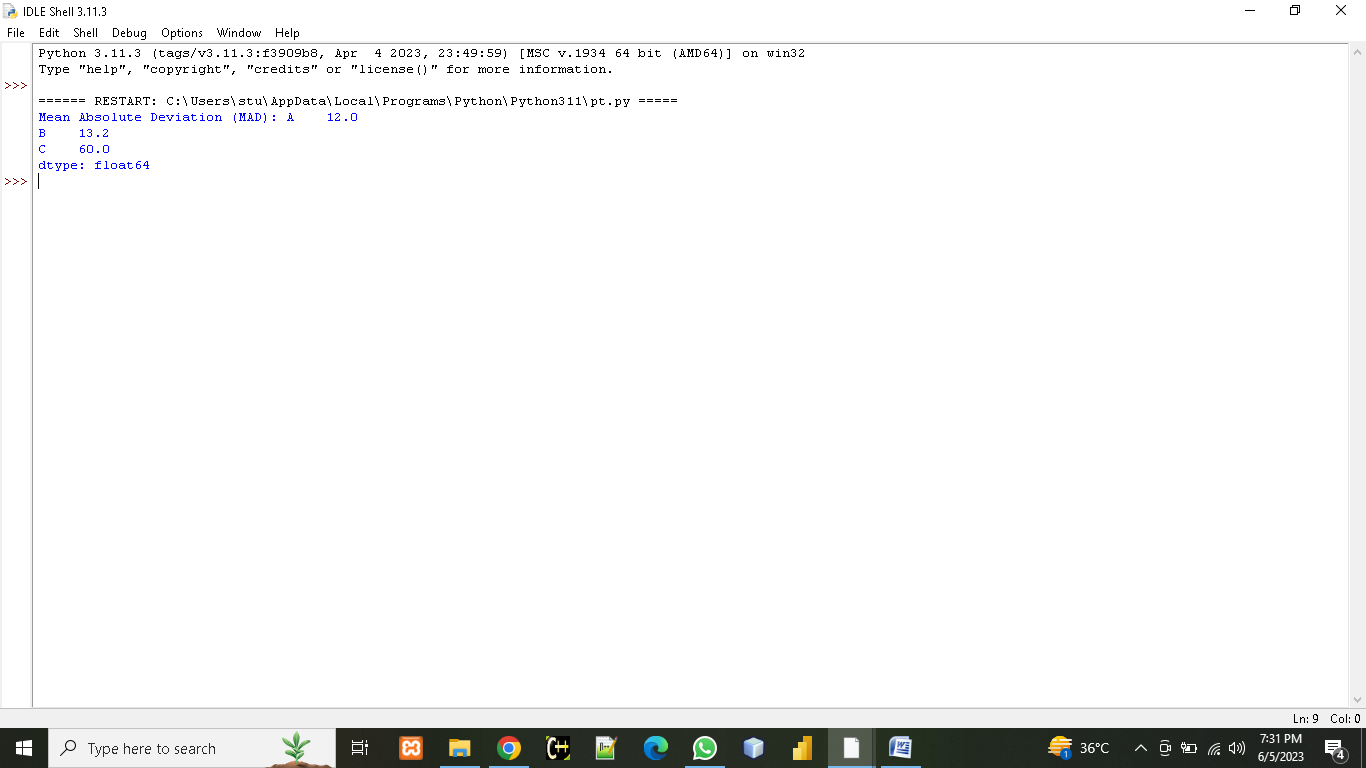
}

df = pd.DataFrame (info)

ma = (df - df.mean ()).abs ().mean ()

print ("Mean Absolute Deviation (MAD):", ma)

**OUTPUT:**



**QUESTION 18:-**

**Create a DataFrame based on employee data and generate quartile and variance.**

**CODING:**

import pandas as pd

data = {

'EmployeeID': [1, 2, 3, 4, 5],

'Age': [25, 32, 28, 45, 36],

'Salary': [50000, 60000, 55000, 70000, 65000],

'Experience': [3, 8, 5, 12, 9]

}

df = pd.DataFrame (data)

print (df)

quartiles = df. quantile ([0.25, 0.5, 0.75])

print ("\n Quartiles :")

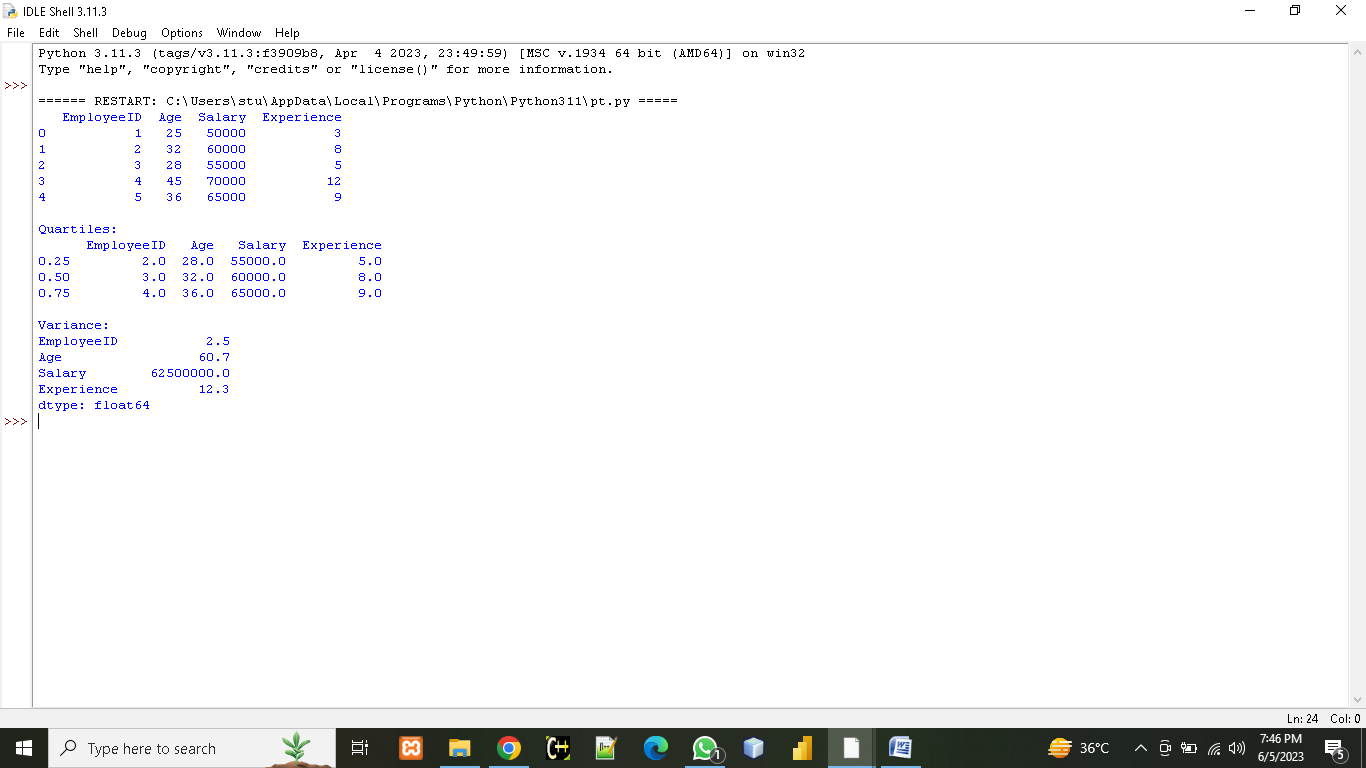
print (quartiles)

variance = df.var ()

print ("\n Variance :")

print (variance)

**OUTPUT:**

****

**QUESTION 19:-**

**Program to implement Skewness on Random data.**

**CODING:**

import pandas as pd

import numpy as np

np.random.seed ()

data = np.random.normal (loc=10, scale=5, size=50)

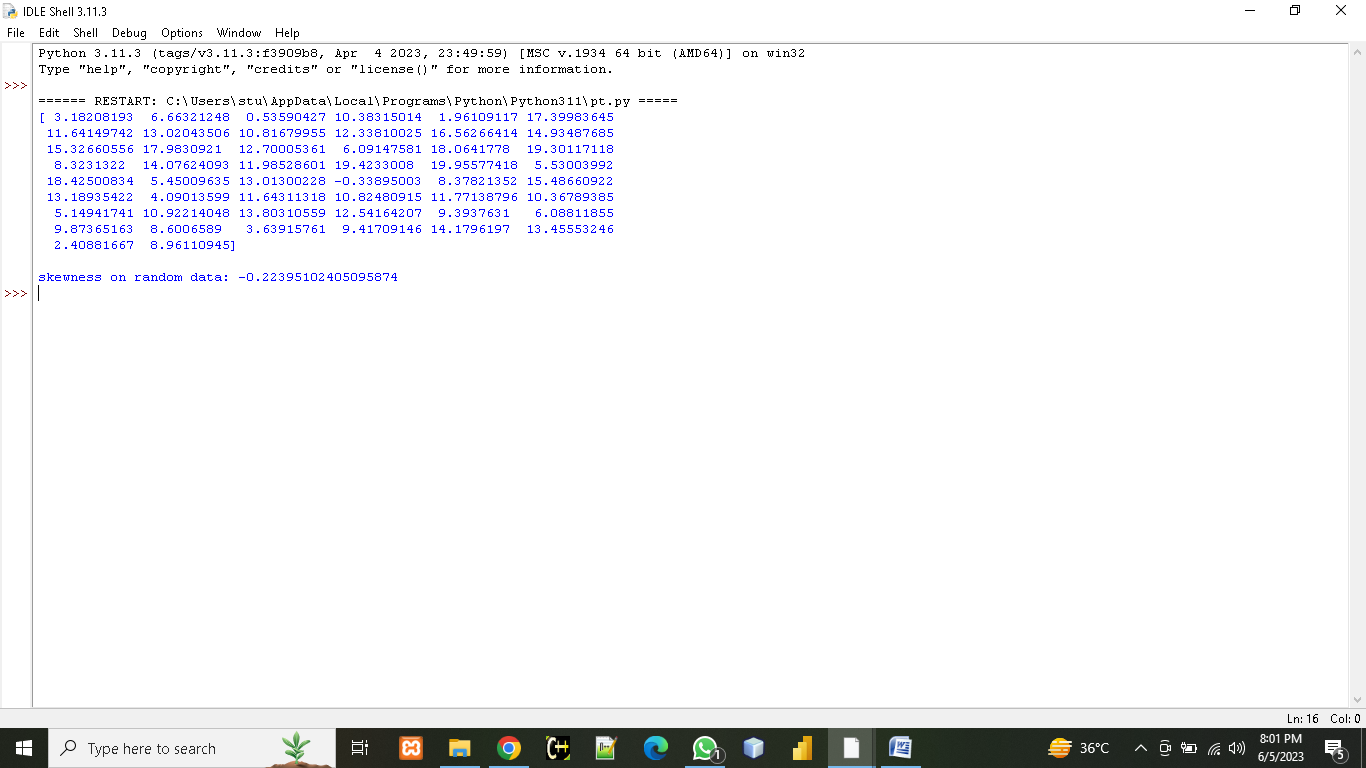
print (data)

df = pd.DataFrame (data, columns= ['Random Data'])

Skewness = df['Random Data'].skew ()

print ("\n Skewness on random data:", skewness)

**OUTPUT:**

****

**QUESTION 20:-**

**Create a DateFrame on any Data and compute statistical function of Kurtosis.**

**CODING:**

import pandas as pd

data = {

'Category': ['A', 'B', 'C', 'D', 'E'],

'Value': [10, 20, 30, 40, 50]

}

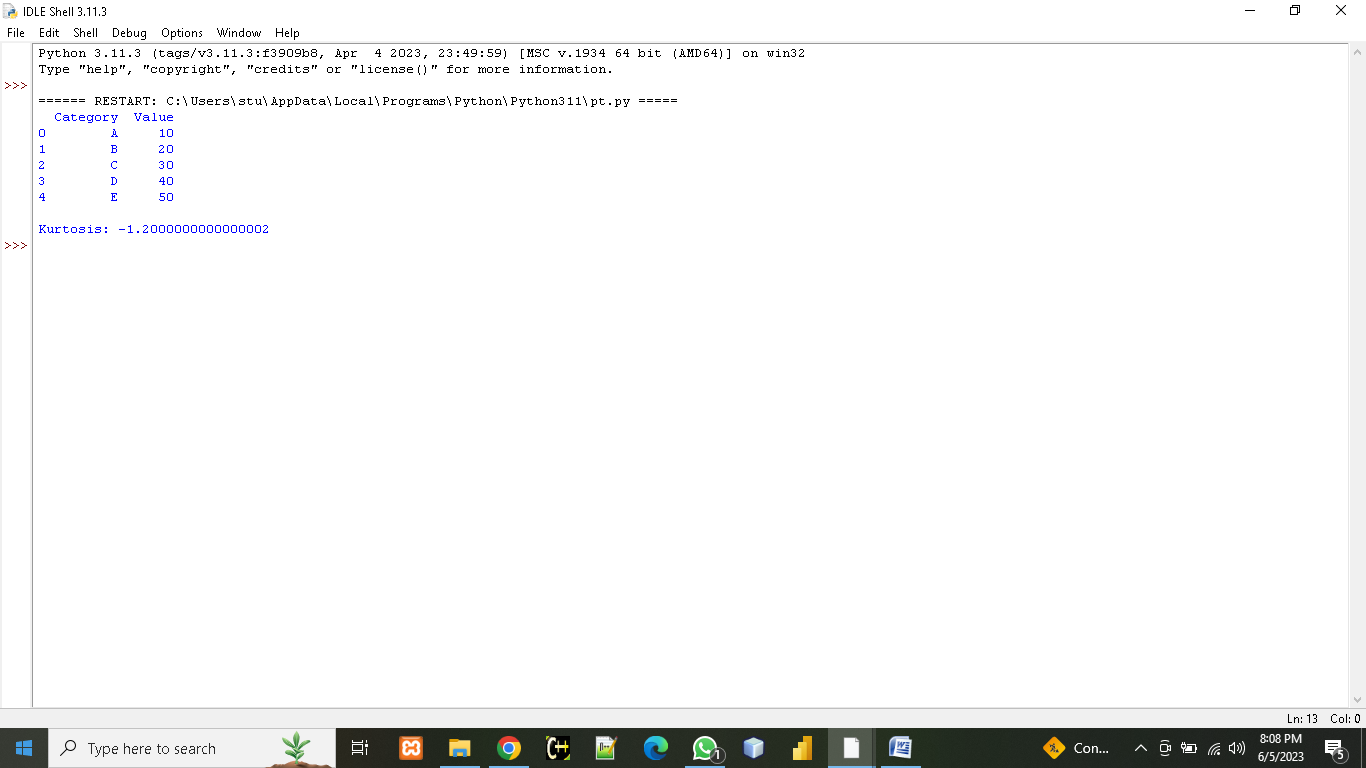
df = pd.DataFrame (data)

print (df)

Kurtosis = df ['Value'].kurtosis ()

print ("\n Kurtosis:", Kurtosis)

**OUTPUT:**

****

**QUESTION 21:-**

**CASE STUDY ON: CRIME\_BY\_STATE\_RT.CSV**

[**crime\_by\_state\_rt.zip**](file:///C:\Users\hp\Downloads\crime_by_state_rt.zip)

21.1: Find the 75% percentile for data

21.2: Apply aggregate function to find count, min, max and sum

21.3: Perform group by on State and Year

21.4: Calculate the number of entries year wise

21.5: Create a pie chart for all states

21.6: Plot the no. of murders state-wise

21.7: Check the data type of all column

21.8: Check the robbery column for NULL values.

If exist replace with 0

21.9: Select the data set where robbery greater than 10 and year= 2021

21.10: describe () the dataset

21.11: Perform sorting on Arson and Robbery

21.12: Perform renaming on two columns

**CODING:**

**OUTPUT:**

**QUESTION 22:-**

**CASE STUDY ON: investigating clinical data**

[**investigating clinical data.csv**](investigating%20clinical%20data.csv)

22.1: describe () the dataset

22.2: Perform renaming on two columns

22.3: Is there any relation between the age of the patient and having a delay on the date of appointment.

22.4:  Is there any relation between gender and delaying appointments? (Show by plotting bar graph)

22.5: Do people delay their appointments when they have no scholarship? (Show by plotting)

22.6: Is there any relationship between gender having scholarships and delaying the appointment? (Show by plotting)

22.7: Which gender has more appointments in which illness? (Show by plotting)

22.8: Does a person suffering from alcoholism tend to delay appointments? (Show by plotting)

22.9: Perform group by and count on neighborhood

22.10: Perform agg () function on dataset.

**CODING:**

**OUTPUT:**

**QUESTION 23:-**

**CASE STUDY ON: Credit card transaction**

[**Credit card transactions - India - Simple.csv**](Credit%20card%20transactions%20-%20India%20-%20Simple.csv)

23.1: Show the head and visualize the basic state for all columns.

23.2: Display pivot table for create summary for total amount spread-

a) Month b) city

23.3: Analyze the impact of gender to study consumer behavior

23.4: Check for NULL values in columns .If exist replaces them with appropriate values.

23.5: Analyze the relationship type between expense type and amount.

23.6: Analyze the spending habits by city and gender.

23.7: implement Skewness and kurtosis on data

Draw plots along with appropriate question

**CODING:**

**OUTPUT:**

**QUESTION 24:-**

**CASE STUDY ON PROJECT TOPIC (Global superstore)**

**CODING:**

**OUTPUT:**

**QUESTION 25:-**

**RESEACH PAPER**

**CODING:**

**OUTPUT:**