**Name: Deepak Kumar**

**Roll No: 04211102021**

**Q1**. **Create a pandas series from a dictionary of values and an ndarray.**

**Code**:

**## Series using dictionary.**

**print("Series using Dictionary")**

**fruits\_price={'Apple':150,'Mango':100,'Banana':80,'Orange':120,'Pineapple':300,'Watermelon':35}**

**series\_fruits = pd.Series(data=fruits\_price)**

**print(series\_fruits)**

**print()**

**# Series using ndarray.**

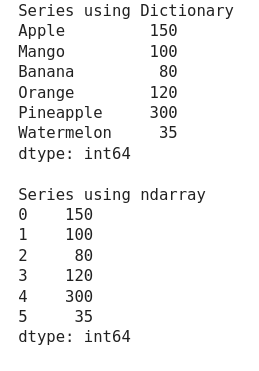
**print("Series using ndarray")**

**arr = np.array([150,100,80,120,300,35])**

**series\_arr = pd.Series(arr)**

**print(series\_arr)**

**Output:**

****

**Q2. Create a Series and print all the elements that are above 75th percentile.**

**Code:**

**#printing all elments above 75th percentile**

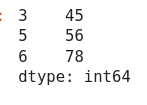
**num = np.array([10,21,13,45,21,56,78,100,12])**

**num\_series = pd.Series(num)**

**filt = (num\_series>num\_series.quantile(0.5)) & (num\_series<num\_series.max())**

**num\_series[filt]**

**Output:**

 **Q3.** **Perform sorting on Series data and DataFrames.**

**Code:**

**#Sorting values from the series**

**print("Sorting values from the series")**

**data\_num = np.array([10,13,21,34,6,24,15,78])**

**series = pd.Series(data\_num)**

**sorted\_series = series.sort\_values()**

**print(sorted\_series)**

**print()**

**print("sorting the values from dataframe")**

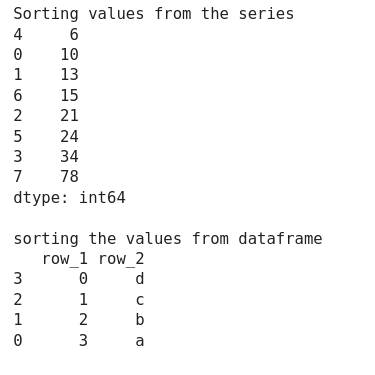
**data = {'row\_1': [3, 2, 1, 0], 'row\_2': ['a', 'b', 'c', 'd']}**

**df = pd.DataFrame(data=data)**

**sorted\_df = df.sort\_values(by="row\_1")**

**print(sorted\_df)**

**Output:**



**Q4.** **Write a program to implement pivot() and pivot-table() on a DataFrame.**

**Code:**

**#implementing pivot**

**data\_col = pd.DataFrame(data={"name":["rohit","deepak","vaibhav","aryan"],"age":[22,20,19,20],"rollno":[39,42,22,41]})**

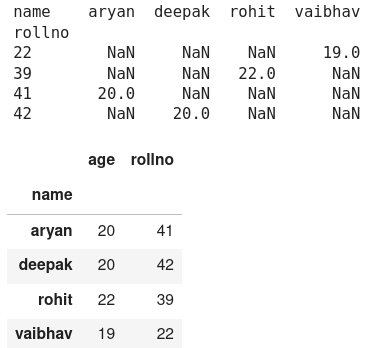
**print(data\_col.pivot(index="rollno",columns="name",values="age"))**

**print()**

**#implementing pandas.pivot\_table()**

**pd.pivot\_table(data\_col,index="name")**

**Output:**



**Q5.** **Write a program to find mean absolute deviation on a DataFrame.**

**Code:**

**student\_data = pd.DataFrame(data={'ID': {0: 23, 1: 43, 2: 12,**

**3: 13, 4: 67, 5: 89,**

**6: 90, 7: 56, 8: 34},**

**'Name': {0: 'Ram', 1: 'Deep',**

**2: 'Yash', 3: 'Aman',**

**4: 'Arjun', 5: 'Aditya',**

**6: 'Akash',7: 'Chalsea',**

**8: 'Divya'},**

**'Marks': {0: 89, 1: 97, 2: 45,**

**3: 78, 4: 56, 5: 76,**

**6: 81, 7: 87, 8: 100},**

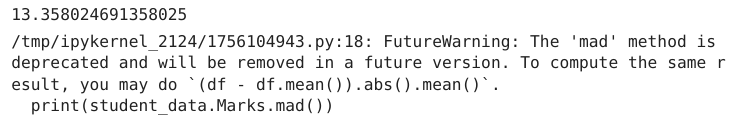
**'Grade': {0: 'B', 1: 'A', 2: 'F', 3: 'C',**

**4: 'E', 5: 'C', 6: 'B', 7: 'B',**

**8: 'A'}})**

**print(student\_data.Marks.mad())**

**Output:**

 **Q6.** **Two Series object, Population stores the details of four metro cities of India and another object AvgIncome stores the total average income reported in four years in these cities. Calculate income per capita for each of these metro cities**

**Code:**

**population = pd.Series(data={"NCR":46069000,"Mumbai":20961000,"Kolkata":15134000,"Banglore":13193000})**

**print("population of Cities")**

**print(population)**

**print()**

**AvgIncome = pd.Series(data={"NCR":12000000,"Mumbai":160000000,"Kolkata":650000,"Banglore":8900000})**

**print("Average income of cities")**

**print(AvgIncome)**

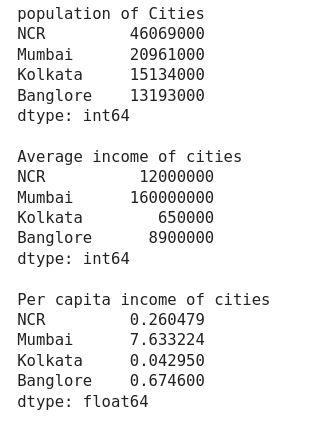
**print()**

**percapita = AvgIncome/population**

**print("Per capita income of cities")**

**print(percapita)**

**Output:**



**Q7.** **Create a DataFrame based on E-Commerce data and generate mean, mode, median.**

**Code:**

**ecom={**

**"products":["headphone","mobile","airpodes","speaker","laptop","grinder"],**

**"quantity":[3000,2000,8000,3432,1200,870],**

**"price":[1200,25000,3000,5000,70000,1200],**

**"discount":[0.12,0.15,0.07,0.05,0.30,0.10]**

**}**

**ecom\_data = pd.DataFrame(ecom)**

**print("data")**

**print(ecom\_data)**

**print()**

**print("mean")**

**print(ecom\_data["price"].mean())**

**print()**

**print("mode")**

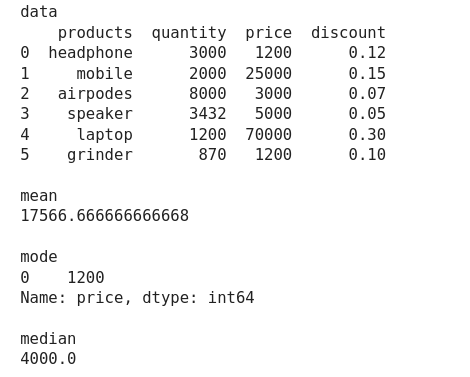
**print(ecom\_data["price"].mode())**

**print()**

**print("median")**

**print(ecom\_data["price"].median())**

**Output:**



**Q8. Create a DataFrame based on employee data and generate quartile and variance.**

**Code:**

**empData = {"Name": ["Deepak Kumar", "Rohit Kumar Kain", "Aryan Mahana", "Adarsh Mishra", "Mannat Mishra"],**

**"Age": [20,21,21,19,18],**

**"Salary":[100000, 120000, 123000, 123400, 123450],**

**"Country":["America", "Canada", "Finland", "Switzerland", "Australia"]}**

**df\_empData = pd.DataFrame(empData)**

**#quartile is:**

**print("Quartile is: ")**

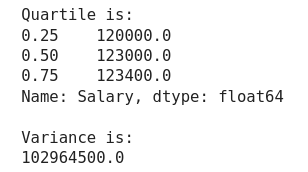
**print(df\_empData['Salary'].quantile([0.25, 0.50, 0.75]))**

**#variance is:**

**print("\nVariance is: ")**

**print(df\_empData["Salary"].var())**

**Output:**



**Q9. Program to implement Skewness on Random data.**

**Code:**

**# Random Data**

**data = np.random.randint(1,10000,50)**

**# skewness on random data**

**df\_data = pd.DataFrame(data)**

**print("Skewness is: ", df\_data.skew().to\_list()[0])**

**Output:**



**Q10. Create a DateFrame on any Data and compute statistical function of Kurtosis.**

**Code:**

**data = np.random.randint(1,10000,50)**

**df\_data = pd.DataFrame(data)**

**print("Kurtosis is: ", df\_data.kurt().to\_list()[0])**

**Output:**



**Q11. 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**

**Code:**

**Temp1 = pd.Series(np.random.randint(20,42,7))**

**Temp2 = pd.Series(np.random.randint(20,42,7))**

**Temp3 = pd.Series(np.random.randint(20,42,7))**

**Temp4 = pd.Series(np.random.randint(20,42,7))**

**print("Average Temperature per Week:")**

**print("Week 1 average Temperature: ", Temp1.mean())**

**print("Week 2 average Temperature: ", Temp2.mean())**

**print("Week 3 average Temperature: ", Temp3.mean())**

**print("Week 4 average Temperature: ", Temp4.mean())**

**print("Average Temperature of whole month: ", pd.concat([Temp1, Temp2, Temp3, Temp4]).mean())**

**Output:**

