ALLIANCE
UNIVERSITY
Private University established in Karnataka State by Act No.34 of year 2010
Recognized by the University Grants Commission (UGC), New Delhi

Alliance School of Advanced Computing

Department of Computer Science and Engineering

Class Assignment-1

Course Code: 5CS1025
Course Title: Artificial Intelligence
Semester: 04
Class: AIML-E
Name: V P JAIDITHYA
Reg no: 2023BCSE07AED412
GitHub:

2024-25

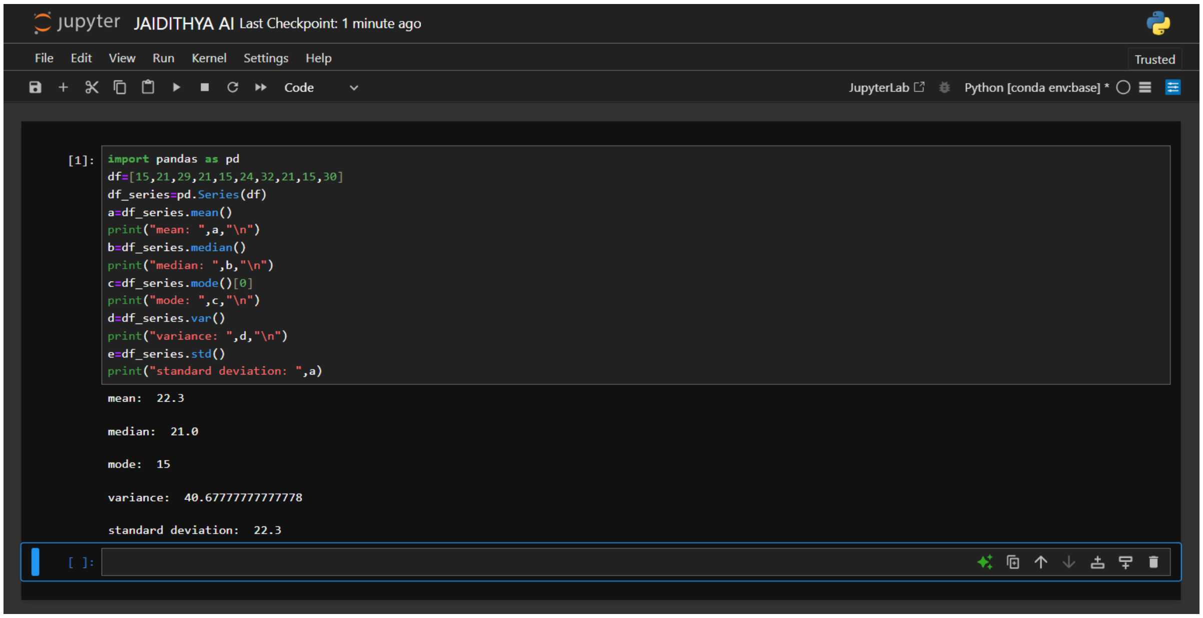
1. **Imagine you are tasked with designing a humanoid robot to assist in a home or office environment. The robot must be capable of interacting with people by talking and listening, walking to different locations, seeing and recognizing objects, and learning from its surroundings to adapt its behavior. What technologies, tools, and frameworks would you need to build such a robot? Give as flow chart**

Start

- Hardware Components
  - Sensors (Cameras, LiDAR, Microphones, IMUs, Touch Sensors)
  - Actuators (Motors, Servos, Hydraulics)
  - Processing Unit (CPU, GPU, Edge AI)
  - Power System (Battery, Power Management)
- Perception System
  - Computer Vision (OpenCV, Tensor Flow, YOLO, Depth Cameras)
  - Speech Recognition (Google ASR, Whisper, Vosk)
  - Object Detection (Deep Learning, CNNs, YOLO, Faster R-CNN)
  - Face & Emotion Recognition (Dlib, Deep Face, OpenFace)
- Movement & Navigation
  - Path Planning (A*, Dijkstra, RRT)
  - SLAM (Simultaneous Localization and Mapping) (Google Cartographer, ROS SLAM)
  - Motor Control (PID Controllers, Inverse Kinematics)
- Interaction & Communication
  - Natural Language Processing (GPT, BERT, Rasa, Dialog flow)
  - Text-to-Speech (Google TTS, Amazon Polly, Festival)
  - Conversational AI (ChatGPT, IBM Watson)
- Learning & Adaptation
  - Reinforcement Learning (OpenAI Gym, TensorFlow RL)
  - Federated Learning (Edge AI with TensorFlow Lite, PyTorch Mobile)
  - Continuous Learning (AutoML, Edge AI)
- Software Frameworks
  - Robot Operating System (ROS, ROS2)
  - AI Frameworks (TensorFlow, PyTorch, OpenAI)
  - Middleware (MQTT, ZeroMQ, Web Sockets)
  - Cloud Integration (AWS IoT, Azure Robotics, Google Cloud)
- User Interface & Control
  - Mobile App (Flutter, React Native)
  - Web Dashboard (React, Node.js, WebRTC)
  - Voice Commands (Alexa, Google Assistant)
  - Gesture Recognition (MediaPipe, OpenPose)

End

2. **Calculate and interpret mean, median, mode, variance and standard deviation for a given dataset. Data = [15,21,29,21,15,24,32,21,15,30]**



jupyter JAIDITHYA AI Last Checkpoint: 1 minute ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

```
[1]: import pandas as pd
df=[15,21,29,21,15,24,32,21,15,30]
df_series=pd.Series(df)
a=df_series.mean()
print("mean: ",a,"\n")
b=df_series.median()
print("median: ",b,"\n")
c=df_series.mode()[0]
print("mode: ",c,"\n")
d=df_series.var()
print("variance: ",d,"\n")
e=df_series.std()
print("standard deviation: ",a)
```

```
mean:  22.3

median:  21.0

mode:  15

variance:  40.67777777777778

standard deviation:  22.3
```

```
[ ]:
```

3. You are analyzing a dataset that captures the daily performance and activity of a humanoid robot in a simulated environment. The dataset link robot_dataset(robot_dataset)_1.csv includes the following attributes

Interaction Count: Number of conversations the robot had daily.

Steps Walked: Total steps taken each day.

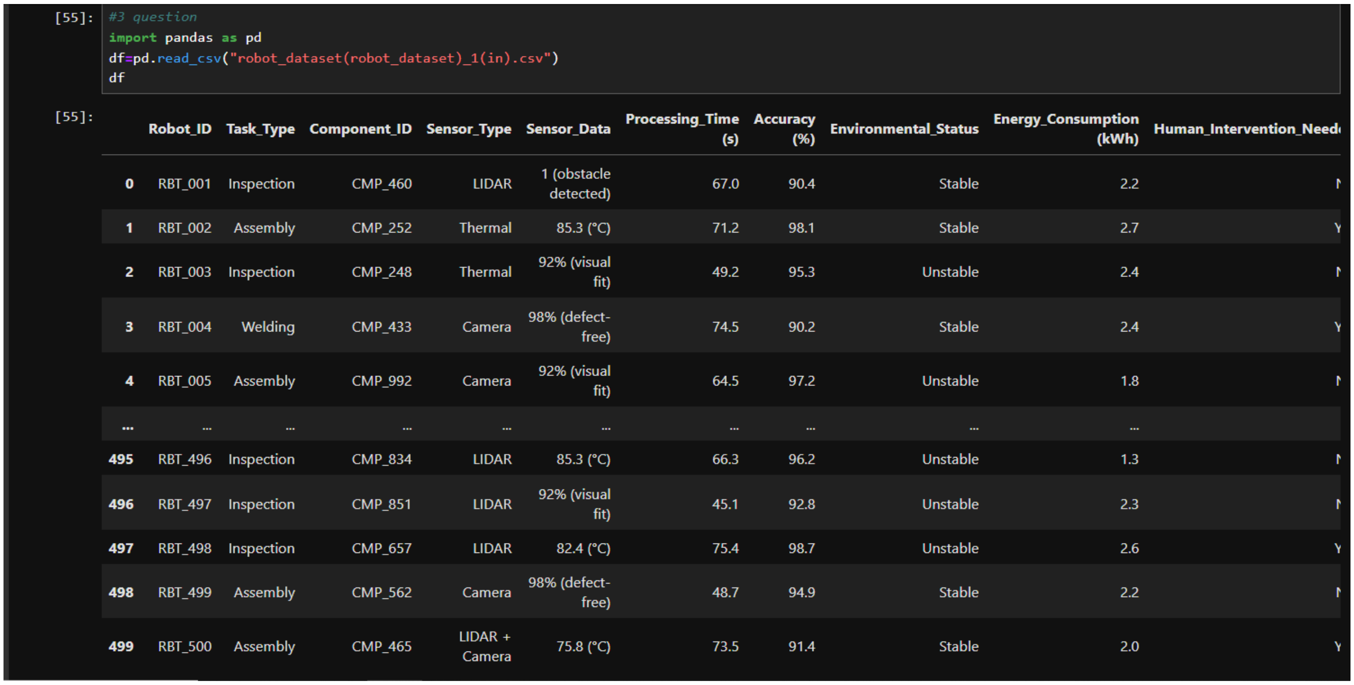
Objects Recognized: Number of objects successfully identified by the robot.

Learning Sessions: Number of learning tasks completed.

Energy Consumption (kWh): Daily energy usage of robots.

Perform Basic Statistical Operations:

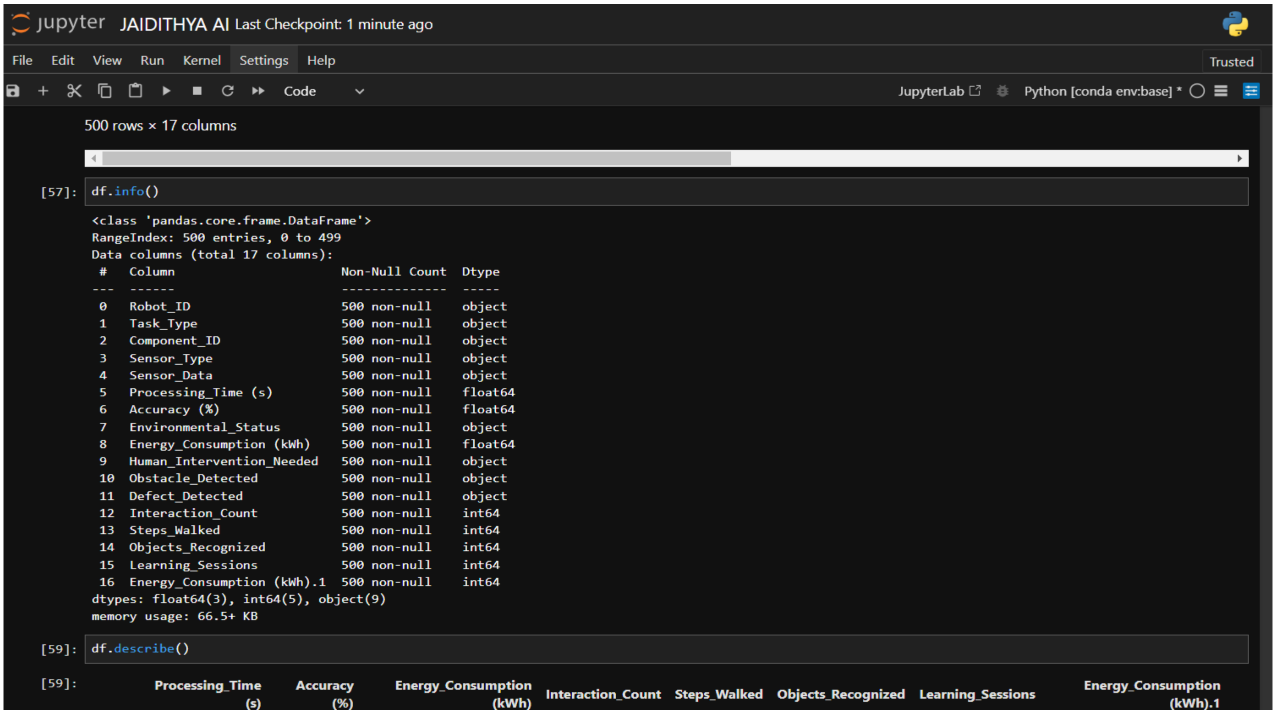
1) What is the **average (mean)** number of conversations the robot has daily?
2) Find the **total steps walked** by the robot over a given period.
3) Determine the **maximum and minimum energy consumption** in the dataset.
4) Calculate the **correlation** between the number of steps walked and energy consumption.
5) Analyze the **distribution** of objects recognized daily (e.g., histogram or box plot).
6) What is the **variance** in the number of learning sessions completed?



```
[55]: #3 question
      import pandas as pd
      df=pd.read_csv("robot_dataset(robot_dataset)_1(in).csv")
      df
```

[55]:

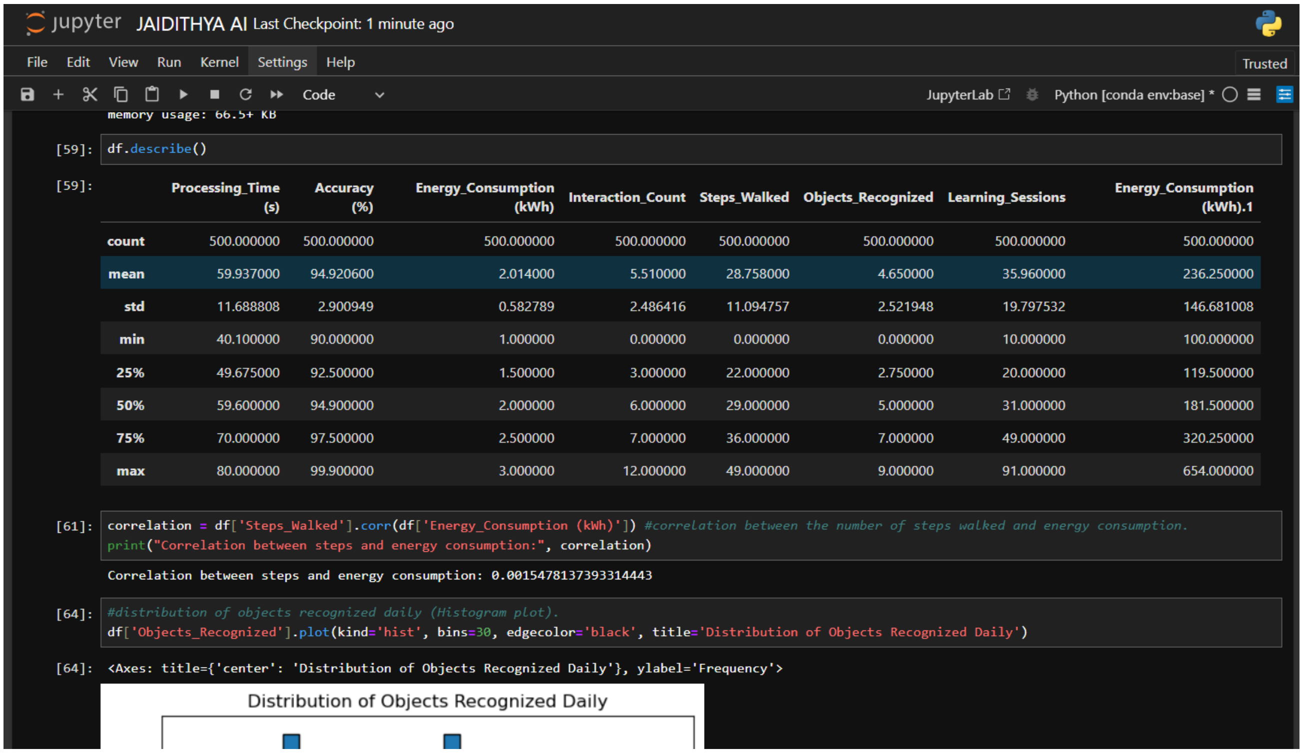
	Robot_ID	Task_Type	Component_ID	Sensor_Type	Sensor_Data	Processing_Time (s)	Accuracy (%)	Environmental_Status	Energy_Consumption (kWh)	Human_Intervention_Need
0	RBT_001	Inspection	CMP_460	LIDAR	1 (obstacle detected)	67.0	90.4	Stable	2.2	N
1	RBT_002	Assembly	CMP_252	Thermal	85.3 (°C)	71.2	98.1	Stable	2.7	Y
2	RBT_003	Inspection	CMP_248	Thermal	92% (visual fit)	49.2	95.3	Unstable	2.4	N
3	RBT_004	Welding	CMP_433	Camera	98% (defect-free)	74.5	90.2	Stable	2.4	Y
4	RBT_005	Assembly	CMP_992	Camera	92% (visual fit)	64.5	97.2	Unstable	1.8	N
...	...	...	...	...	...	...	...	...	...	
495	RBT_496	Inspection	CMP_834	LIDAR	85.3 (°C)	66.3	96.2	Unstable	1.3	N
496	RBT_497	Inspection	CMP_851	LIDAR	92% (visual fit)	45.1	92.8	Unstable	2.3	N
497	RBT_498	Inspection	CMP_657	LIDAR	82.4 (°C)	75.4	98.7	Unstable	2.6	Y
498	RBT_499	Assembly	CMP_562	Camera	98% (defect-free)	48.7	94.9	Stable	2.2	N
499	RBT_500	Assembly	CMP_465	LIDAR + Camera	75.8 (°C)	73.5	91.4	Stable	2.0	Y


jupyter JAIDITHYA AI Last Checkpoint: 1 minute ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

500 rows × 17 columns

```
[57]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Robot_ID                  500 non-null    object 
 1   Task_Type                 500 non-null    object 
 2   Component_ID              500 non-null    object 
 3   Sensor_Type               500 non-null    object 
 4   Sensor_Data               500 non-null    object 
 5   Processing_Time (s)       500 non-null    float64
 6   Accuracy (%)              500 non-null    float64
 7   Environmental_Status      500 non-null    object 
 8   Energy_Consumption (kWh)  500 non-null    float64
 9   Human_Intervention_Needed 500 non-null    object 
 10  Obstacle_Detected         500 non-null    object 
 11  Defect_Detected           500 non-null    object 
 12  Interaction_Count         500 non-null    int64  
 13  Steps_Walked              500 non-null    int64  
 14  Objects_Recognized        500 non-null    int64  
 15  Learning_Sessions         500 non-null    int64  
 16  Energy_Consumption (kWh).1  500 non-null    int64  
dtypes: float64(3), int64(5), object(9)
memory usage: 66.5+ KB

[59]: df.describe()

[59]:
```

Processing_Time (s) | Accuracy (%) | Energy_Consumption (kWh) | Interaction_Count | Steps_Walked | Objects_Recognized | Learning_Sessions | Energy_Consumption (kWh).1


jupyter JAIDITHYA AI Last Checkpoint: 1 minute ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

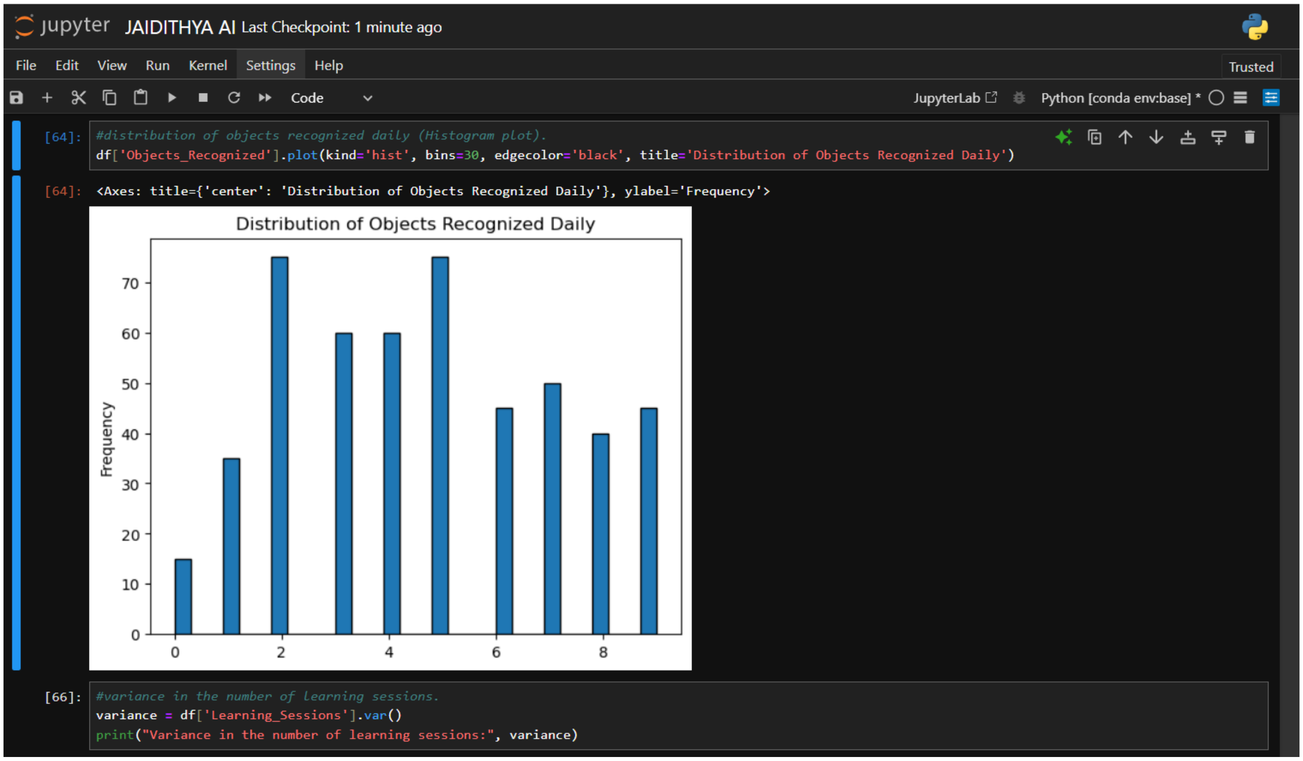
memory usage: 66.5+ KB

[59]: df.describe()

[59]:

	Processing_Time (s)	Accuracy (%)	Energy_Consumption (kWh)	Interaction_Count	Steps_Walked	Objects_Recognized	Learning_Sessions	Energy_Consumption (kWh).1
count	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000	500.000000
mean	59.937000	94.920600	2.014000	5.510000	28.758000	4.650000	35.960000	236.250000
std	11.688808	2.900949	0.582789	2.486416	11.094757	2.521948	19.797532	146.681008
min	40.100000	90.000000	1.000000	0.000000	0.000000	0.000000	10.000000	100.000000
25%	49.675000	92.500000	1.500000	3.000000	22.000000	2.750000	20.000000	119.500000
50%	59.600000	94.900000	2.000000	6.000000	29.000000	5.000000	31.000000	181.500000
75%	70.000000	97.500000	2.500000	7.000000	36.000000	7.000000	49.000000	320.250000
max	80.000000	99.900000	3.000000	12.000000	49.000000	9.000000	91.000000	654.000000

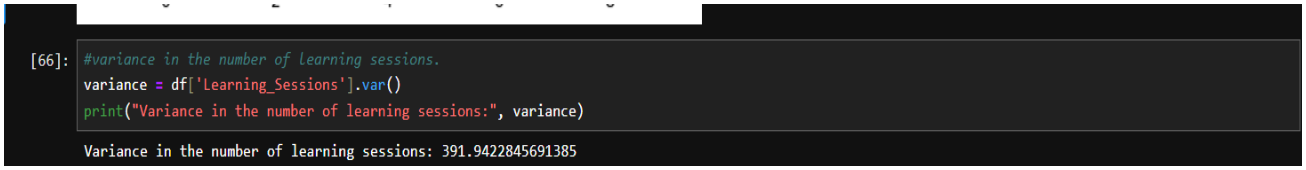
```
[61]: correlation = df['Steps_Walked'].corr(df['Energy_Consumption (kWh)']) #correlation between the number of steps walked and energy consumption.
      print("Correlation between steps and energy consumption:", correlation)

      Correlation between steps and energy consumption: 0.0015478137393314443

[64]: #distribution of objects recognized daily (Histogram plot).
      df['Objects_Recognized'].plot(kind='hist', bins=30, edgecolor='black', title='Distribution of Objects Recognized Daily')

[64]: <Axes: title={'center': 'Distribution of Objects Recognized Daily'}, ylabel='Frequency'>
```


jupyter JAIDITHYA AI Last Checkpoint: 1 minute ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *
Distribution of Objects Recognized Daily
Frequency
0
10
20
30
40
50
60
70
0
2
4
6
8

```
[64]: #distribution of objects recognized daily (Histogram plot).
      df['Objects_Recognized'].plot(kind='hist', bins=30, edgecolor='black', title='Distribution of Objects Recognized Daily')
```

```
[64]: <Axes: title={'center': 'Distribution of Objects Recognized Daily'}, ylabel='Frequency'>
```

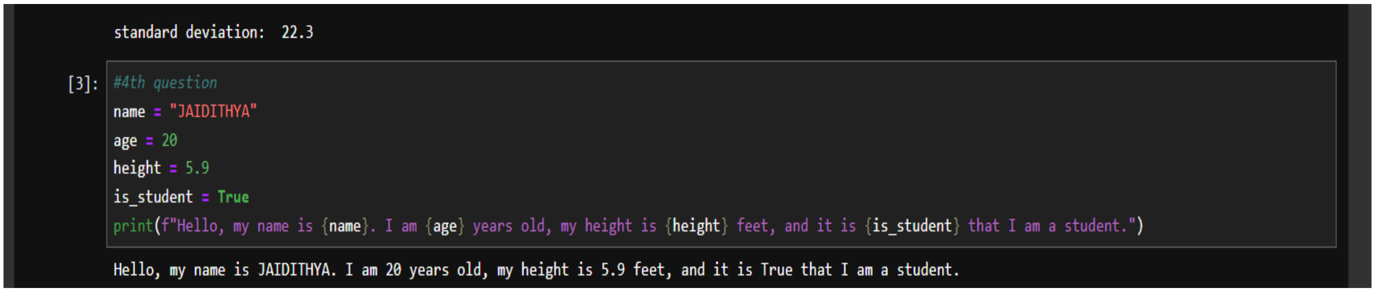
```
[66]: #variance in the number of learning sessions.
      variance = df['Learning_Sessions'].var()
      print("Variance in the number of learning sessions:", variance)
```



```
[66]: #variance in the number of learning sessions.
      variance = df['Learning_Sessions'].var()
      print("Variance in the number of learning sessions:", variance)
```

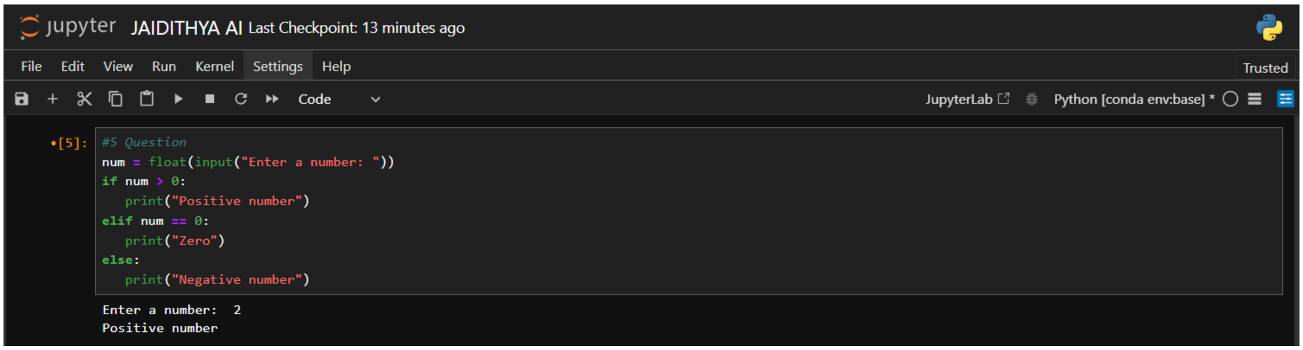
```
Variance in the number of learning sessions: 391.9422845691385
```

4. Write a Python program that declares variables of different data types (e.g., string, integer, float, and Boolean). Output the variables in a sentence format using print () and f-strings.



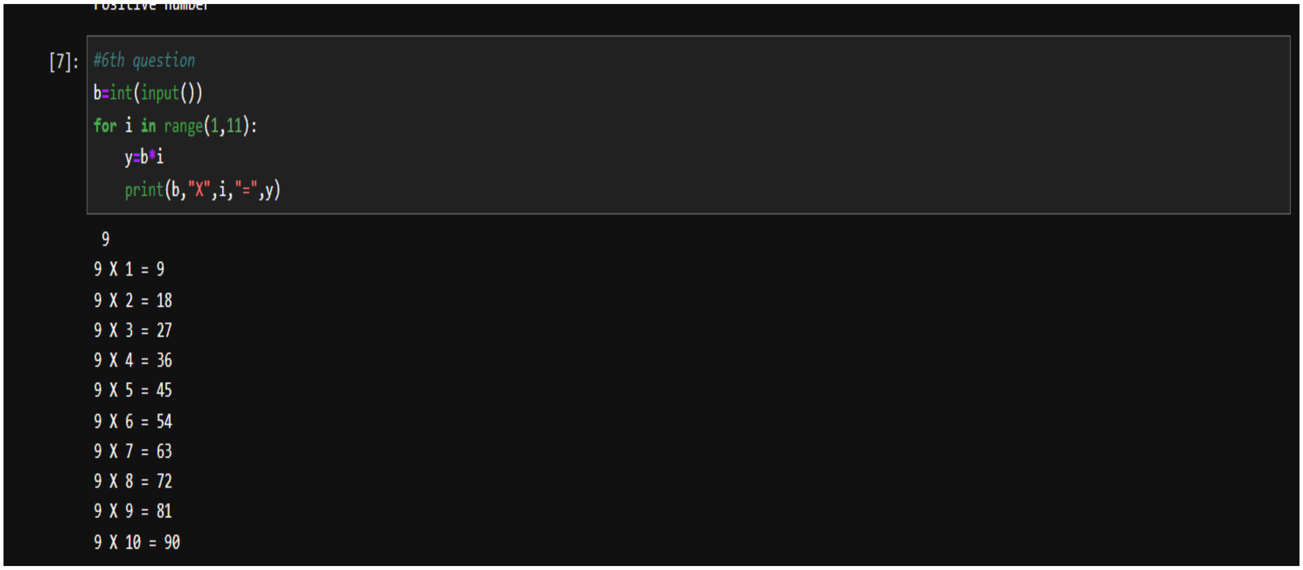
```
standard deviation:  22.3

[3]: #4th question
     name = "JAIDITHYA"
     age = 20
     height = 5.9
     is_student = True
     print(f"Hello, my name is {name}. I am {age} years old, my height is {height} feet, and it is {is_student} that I am a student.")

     Hello, my name is JAIDITHYA. I am 20 years old, my height is 5.9 feet, and it is True that I am a student.
```

5. Write a Python program that takes an integer input and checks whether the number is positive, negative, or zero using conditional statements (if-else).


jupyter
JAIDITHYA AI Last Checkpoint: 13 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

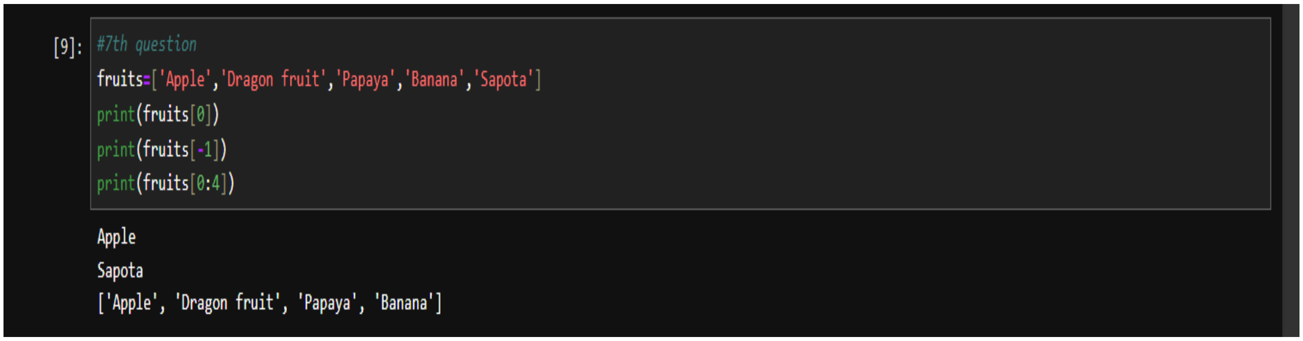
```
•[5]: #5 Question
      num = float(input("Enter a number: "))
      if num > 0:
         print("Positive number")
      elif num == 0:
         print("Zero")
      else:
         print("Negative number")

      Enter a number:  2
      Positive number
```

6. Write a Python program that takes a number as input and prints the multiplication table for that number (from 1 to 10).



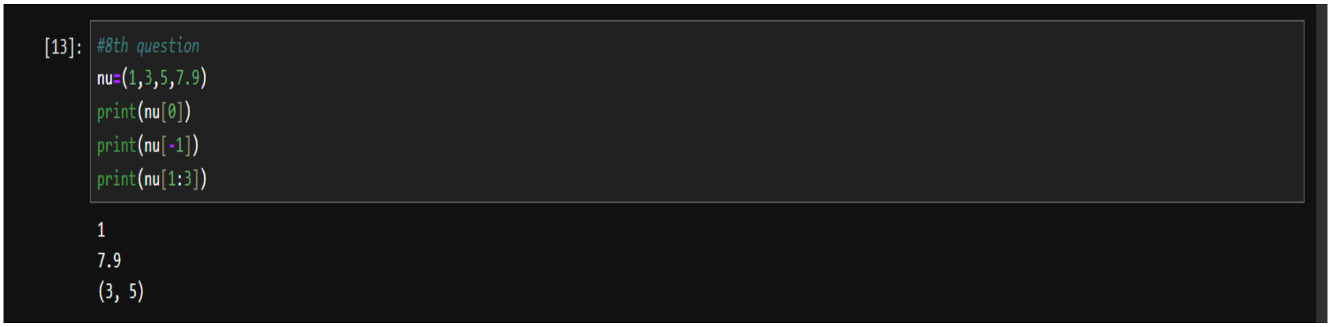
```
     Positive number

[7]: #6th question
     b=int(input())
     for i in range(1,11):
         y=b*i
         print(b,"X",i,"=",y)

      9
     9 X 1 = 9
     9 X 2 = 18
     9 X 3 = 27
     9 X 4 = 36
     9 X 5 = 45
     9 X 6 = 54
     9 X 7 = 63
     9 X 8 = 72
     9 X 9 = 81
     9 X 10 = 90
```

7. Create a Python list that contains the names of 5 different fruits. Perform the given operations on the list.



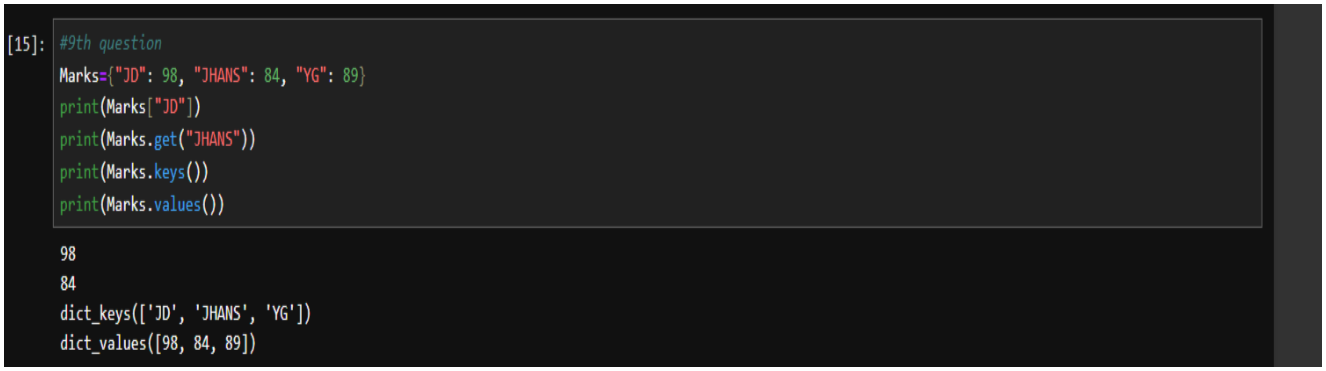
```
[9]: #7th question
     fruits=['Apple','Dragon fruit','Papaya','Banana','Sapota']
     print(fruits[0])
     print(fruits[-1])
     print(fruits[0:4])

     Apple
     Sapota
     ['Apple', 'Dragon fruit', 'Papaya', 'Banana']
```

8. Write a Python program that creates a tuple containing 5 numbers. Perform the given operations on the tuple.



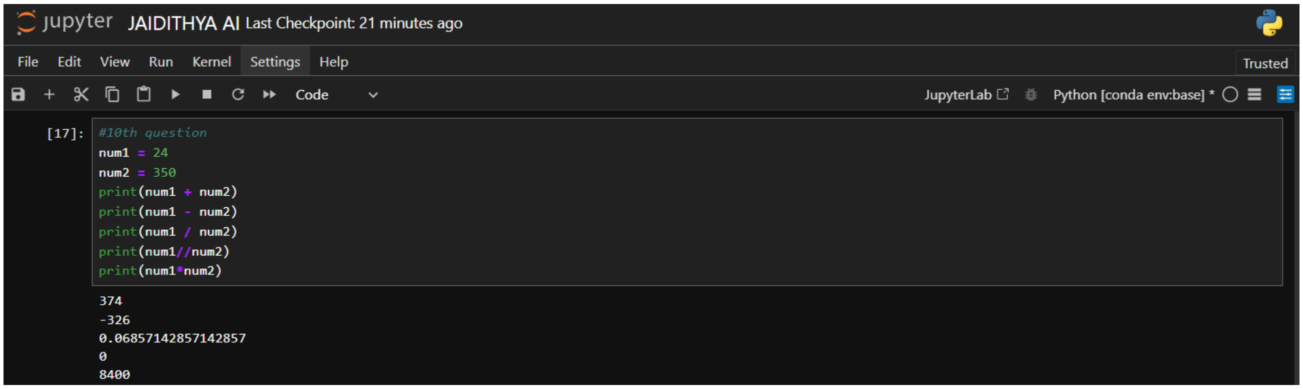
```
[13]: #8th question
      nu=(1,3,5,7.9)
      print(nu[0])
      print(nu[-1])
      print(nu[1:3])

      1
      7.9
      (3, 5)
```

9. Create a dictionary that stores the names of 3 students as keys and their marks in mathematics as values. Perform the given operations.



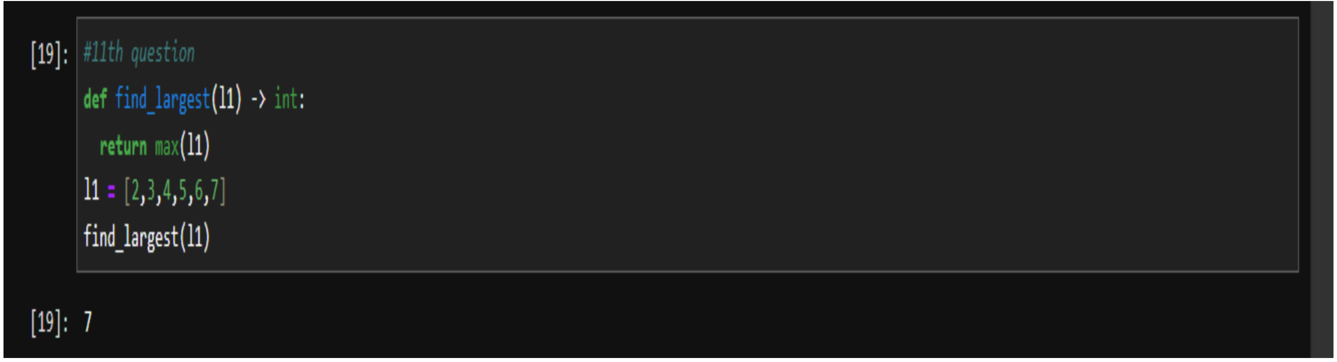
```
[15]: #9th question
      Marks={"JD": 98, "JHANS": 84, "YG": 89}
      print(Marks["JD"])
      print(Marks.get("JHANS"))
      print(Marks.keys())
      print(Marks.values())

      98
      84
      dict_keys(['JD', 'JHANS', 'YG'])
      dict_values([98, 84, 89])
```

10. Create two sets of integers. Perform the given set operations.


jupyter
JAIDITHYA AI Last Checkpoint: 21 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python [conda env:base] *

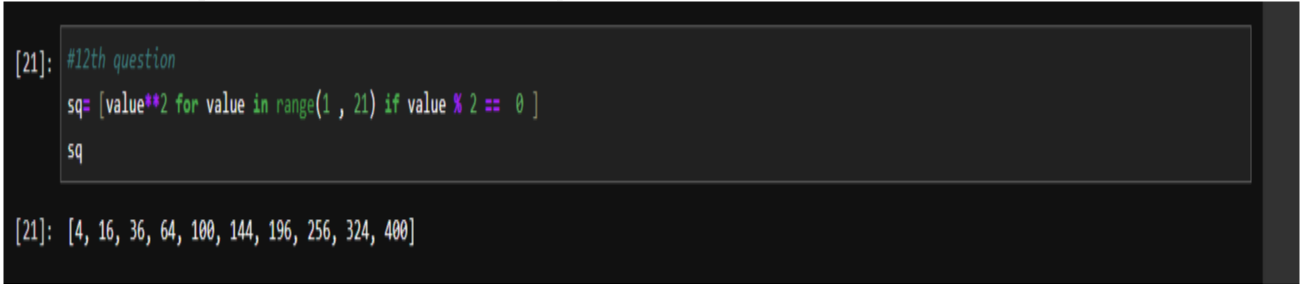
```
[17]: #10th question
      num1 = 24
      num2 = 350
      print(num1 + num2)
      print(num1 - num2)
      print(num1 / num2)
      print(num1//num2)
      print(num1*num2)

      374
      -326
      0.06857142857142857
      0
      8400
```

11. Write a Python function called find largest () that takes a list of numbers as input and returns the largest number from the list. Test the function with a sample list.



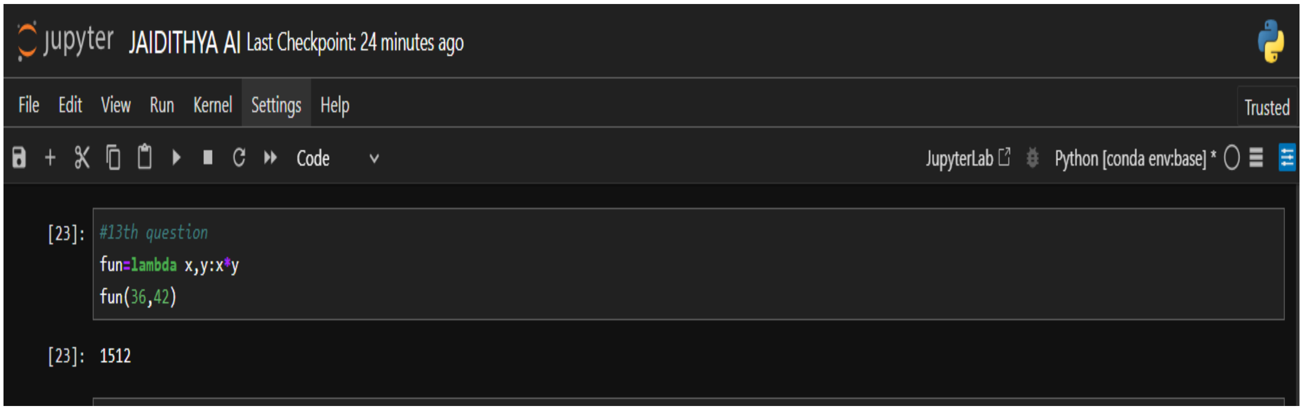
```
[19]: #11th question
      def find_largest(l1) -> int:
        return max(l1)
      l1 = [2,3,4,5,6,7]
      find_largest(l1)

[19]: 7
```

12. Use list comprehension to create a list of squares of all even numbers between 1 and 20.



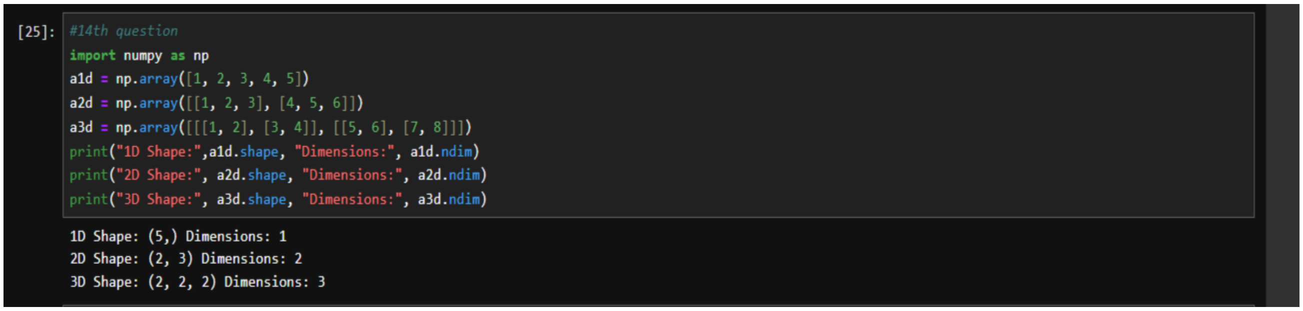
```
[21]: #12th question
      sq= [value**2 for value in range(1 , 21) if value % 2 ==  0 ]
      sq

[21]: [4, 16, 36, 64, 100, 144, 196, 256, 324, 400]
```

13. Write a Python script that uses a lambda function to calculate the product of two numbers provided by the user.


jupyter JAIDITHYA AI Last Checkpoint: 24 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

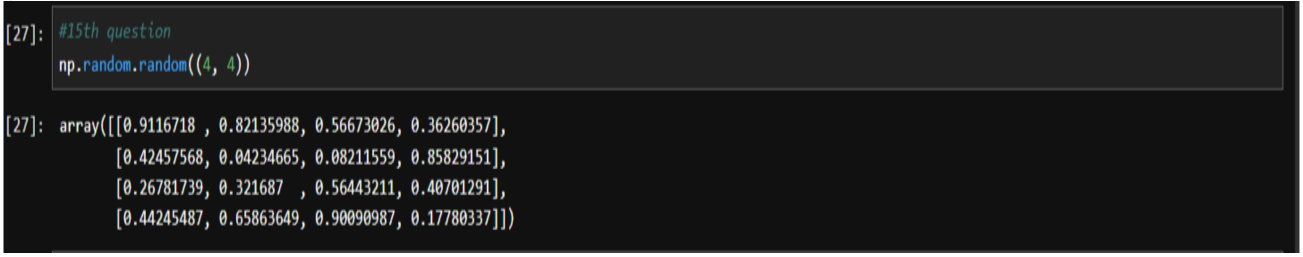
```
[23]: #13th question
      fun=lambda x,y:x*y
      fun(36,42)

[23]: 1512
```

14. Write a Python program to create a one-dimensional, two-dimensional, and three-dimensional NumPy array. Print the shape and dimensions of each array



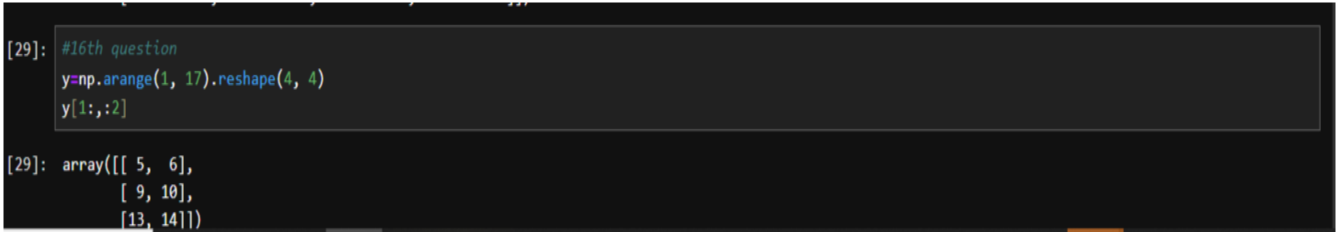
```
[25]: #14th question
      import numpy as np
      a1d = np.array([1, 2, 3, 4, 5])
      a2d = np.array([[1, 2, 3], [4, 5, 6]])
      a3d = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
      print("1D Shape:",a1d.shape, "Dimensions:", a1d.ndim)
      print("2D Shape:", a2d.shape, "Dimensions:", a2d.ndim)
      print("3D Shape:", a3d.shape, "Dimensions:", a3d.ndim)

      1D Shape: (5,) Dimensions: 1
      2D Shape: (2, 3) Dimensions: 2
      3D Shape: (2, 2, 2) Dimensions: 3
```

15. Write a Python program to create a 5x5 NumPy array of random integers and Perform array indexing as given



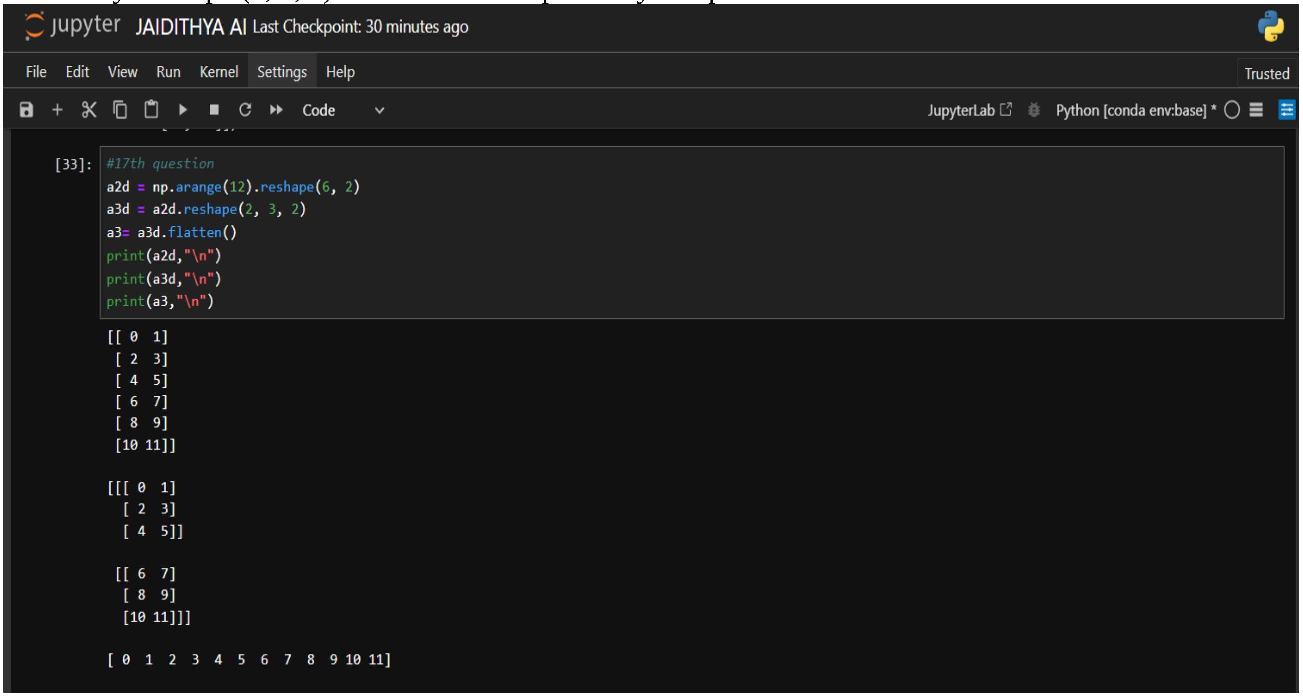
```
[27]: #15th question
      np.random.random((4, 4))

[27]: array([[0.9116718 , 0.82135988, 0.56673026, 0.36260357],
             [0.42457568, 0.04234665, 0.08211559, 0.85829151],
             [0.26781739, 0.321687  , 0.56443211, 0.40701291],
             [0.44245487, 0.65863649, 0.90090987, 0.17780337]])
```

16. Create a NumPy array of shape (4, 4) containing numbers from 1 to 16. Use slicing to extract for the given conditions



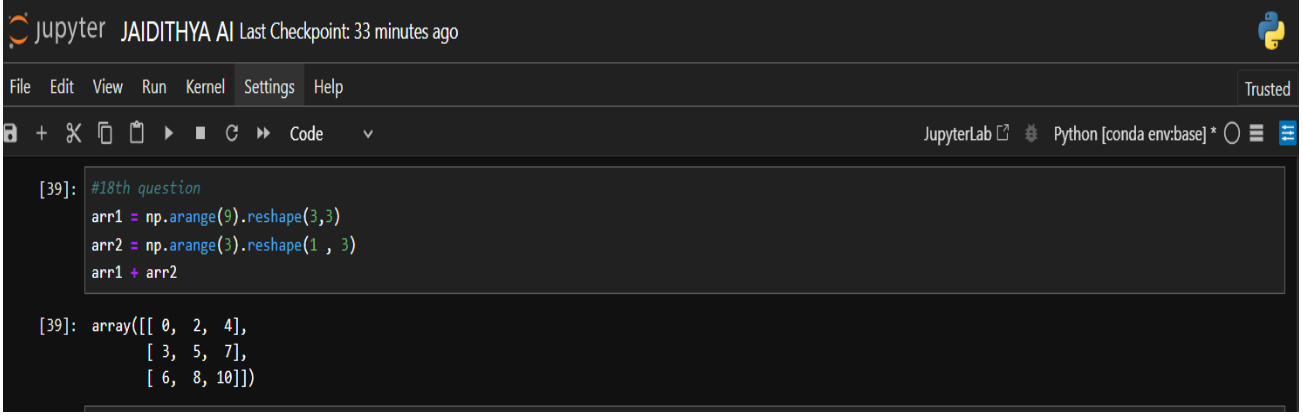
```
[29]: #16th question
      y=np.arange(1, 17).reshape(4, 4)
      y[1:,:2]

[29]: array([[ 5,  6],
             [ 9, 10],
             [13, 14]])
```

17. Write a Python program that creates a 2D array of shape (6, 2) using np.arange () and then reshapes it into a 3D array of shape (2, 3, 2). Flatten the reshaped array and print the result


jupyter
JAIDITHYA AI Last Checkpoint: 30 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python [conda env:base] *

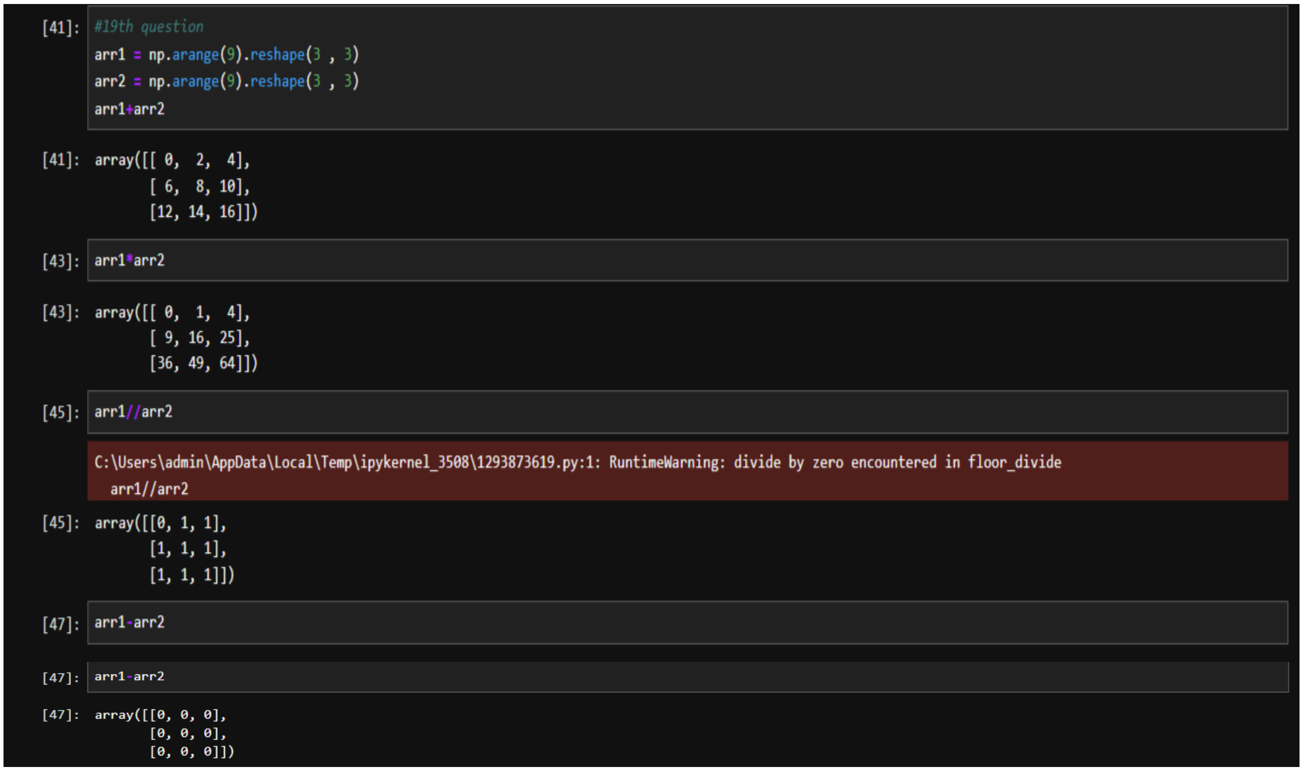
```
[33]: #17th question
      a2d = np.arange(12).reshape(6, 2)
      a3d = a2d.reshape(2, 3, 2)
      a3= a3d.flatten()
      print(a2d,"\n")
      print(a3d,"\n")
      print(a3,"\n")

      [[ 0  1]
       [ 2  3]
       [ 4  5]
       [ 6  7]
       [ 8  9]
       [10 11]]

      [[[ 0  1]
        [ 2  3]
        [ 4  5]]

       [[ 6  7]
        [ 8  9]
        [10 11]]]

      [ 0  1  2  3  4  5  6  7  8  9 10 11]
```

18. Write a Python program to demonstrate broadcasting. Create an array of shape (3, 3) and add a one-dimensional array of shape (1, 3) to it using broadcasting


jupyter JAIDITHYA AI Last Checkpoint: 33 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

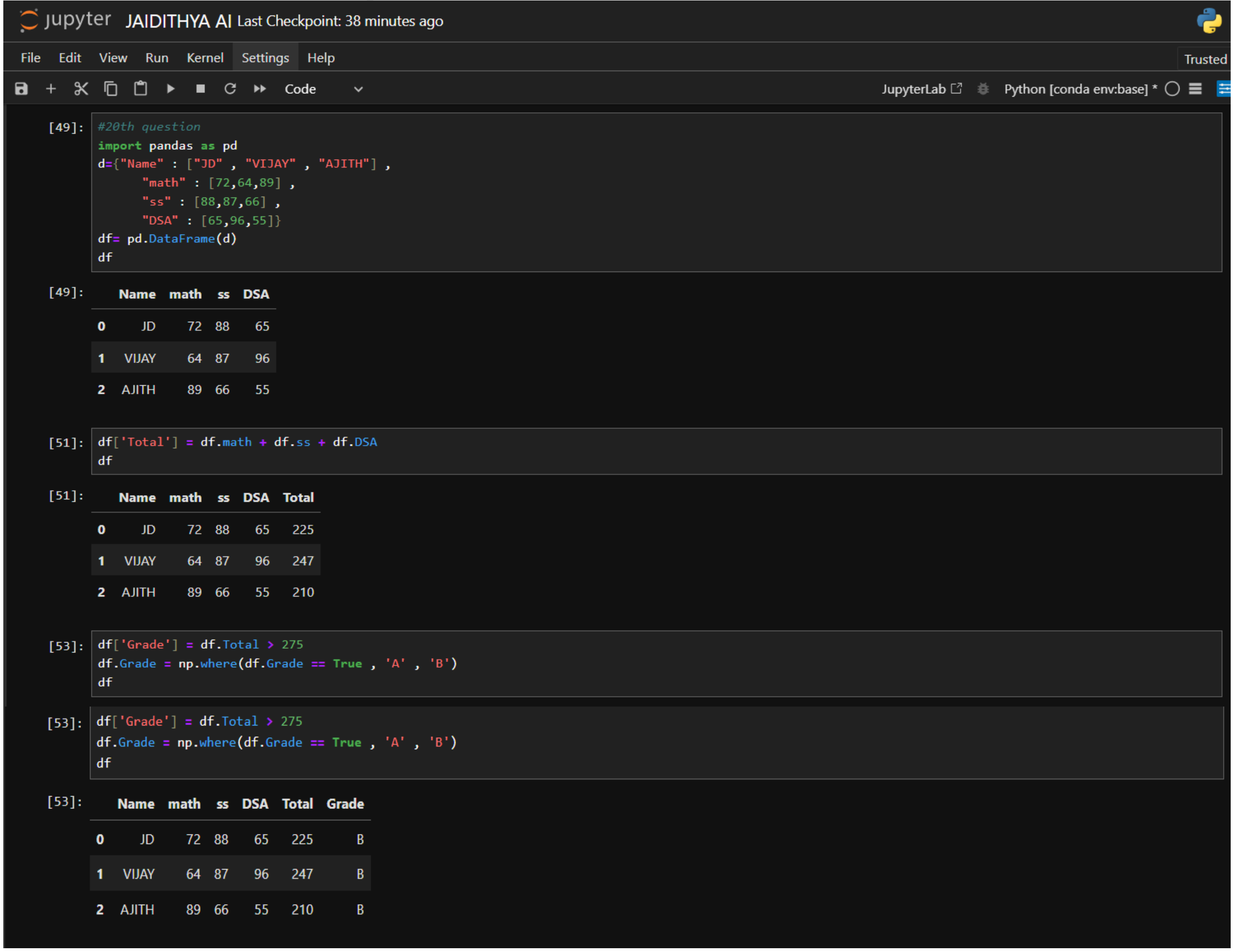
```
[39]: #18th question
      arr1 = np.arange(9).reshape(3,3)
      arr2 = np.arange(3).reshape(1 , 3)
      arr1 + arr2

[39]: array([[ 0,  2,  4],
             [ 3,  5,  7],
             [ 6,  8, 10]])
```

19. Create two NumPy arrays of the same shape, A and B. Perform the following arithmetic operations:
Element-wise addition.
Element-wise subtraction.
Element-wise multiplication.
Element-wise division



```
[41]: #19th question
      arr1 = np.arange(9).reshape(3 , 3)
      arr2 = np.arange(9).reshape(3 , 3)
      arr1+arr2

[41]: array([[ 0,  2,  4],
             [ 6,  8, 10],
             [12, 14, 16]])

[43]: arr1*arr2

[43]: array([[ 0,  1,  4],
             [ 9, 16, 25],
             [36, 49, 64]])

[45]: arr1//arr2

      C:\Users\admin\AppData\Local\Temp\ipykernel_3508\1293873619.py:1: RuntimeWarning: divide by zero encountered in floor_divide
        arr1//arr2

[45]: array([[0, 1, 1],
             [1, 1, 1],
             [1, 1, 1]])

[47]: arr1-arr2

[47]: arr1-arr2

[47]: array([[0, 0, 0],
             [0, 0, 0],
             [0, 0, 0]])
```

20. Create a Pandas Data Frame with the given Name and marks of 3 courses:
Add a new column named 'Total' that represents the sum of all the courses. Add 'Grade' based on the values of the 'Total'. Print the updated Data Frame with the new 'Total' and 'Grade' column



jupyter JAIDITHYA AI Last Checkpoint: 38 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python [conda env:base] *

```
[49]: #20th question
      import pandas as pd
      d={"Name" : ["JD" , "VIJAY" , "AJITH"] ,
            "math" : [72,64,89] ,
            "ss" : [88,87,66] ,
            "DSA" : [65,96,55]}
      df= pd.DataFrame(d)
      df
```

```
[49]:    Name  math  ss  DSA
      0    JD    72  88   65
      1 VIJAY    64  87   96
      2 AJITH    89  66   55
```

```
[51]: df['Total'] = df.math + df.ss + df.DSA
      df
```

```
[51]:    Name  math  ss  DSA  Total
      0    JD    72  88   65    225
      1 VIJAY    64  87   96    247
      2 AJITH    89  66   55    210
```

```
[53]: df['Grade'] = df.Total > 275
      df.Grade = np.where(df.Grade == True , 'A' , 'B')
      df
```

```
[53]: df['Grade'] = df.Total > 275
      df.Grade = np.where(df.Grade == True , 'A' , 'B')
      df
```

```
[53]:    Name  math  ss  DSA  Total  Grade
      0    JD    72  88   65    225      B
      1 VIJAY    64  87   96    247      B
      2 AJITH    89  66   55    210      B
```