**FSDS MAY BATCH 2022(Python Assignment -23)**

**Submitted by: Shubham Tiwari**

Q1: If you have any, what are your choices for increasing the comparison between different figures on the same graph?

Ans: **1) Line graph.**

Line graphs are used to track changes over short and long periods of time. When smaller changes exist, line graphs are better to use than bar graphs. Line graphs can also be used to compare changes over the same period of time for more than one group.

**2)Pie Chart.**

Pie charts are best to use when you are trying to compare parts of a whole. They do not show changes over time.

**3)Bar Graph.**

Bar graphs are used to compare things between different groups or to track changes over time. However, when trying to measure change over time, bar graphs are best when the changes are larger.

**4) Area Graph.**

Area graphs are very similar to line graphs. They can be used to track changes over time for one or more groups. Area graphs are good to use when you are tracking the changes in two or more related groups that make up one whole category (for example public and private groups).

**5)an X-Y Plot.**

X-Y plots are used to determine relationships between the two different things. The x-axis is used to measure one event (or variable) and the y-axis is used to measure the other. If both variables increase at the same time, they have a positive relationship. If one variable decreases while the other increases, they have a negative relationship. Sometimes the variables don't follow any pattern and have no relationship.

Q2: Can you explain the benefit of compound interest over a higher rate of interest that does not compound after reading this chapter?

Ans: Following are some important benefits such as :

1)**Time Complexity**: since no loop is used the algorithm takes up constant time to perform the operations

2)**Auxiliary** **Space**: since no extra array is used so the space taken by the algorithm is constant.

Q3: If necessary, how do you change the aspect ratios between the X and Y axes?

Ans: We can change the aspect ratio using the **pbaspect function**. Set the ratio as a three-element vector of positive values that represent the relative axis lengths.

Q4: What is a histogram, exactly? Name a numpy method for creating such a graph.

Ans: Histogram is the best way to visualize the frequency distribution of a dataset by splitting it into small equal-sized intervals called bins. The Numpy histogram function is similar to the hist() function of matplotlib library, the only difference is that the Numpy histogram gives the numerical representation of the dataset while the hist() gives graphical representation of the dataset.

Numpy has a built-in numpy.histogram() function which represents the frequency of data distribution in the graphical form. The rectangles having equal horizontal size corresponds to class interval called bin and variable height corresponding to the frequency.

**Syntax:**

*numpy.histogram(data, bins=10, range=None, normed=None, weights=None, density=None)*

Q5: Compare and contrast the three types of array multiplication between two numpy arrays: dot product, outer product, and regular multiplication of two numpy arrays.

Ans: There are primarily three different types of matrix multiplication :

|  |  |
| --- | --- |
| **Function** | **Description** |
| np.matmul(array a, array b) | Returns matrix product of two given arrays |
| np.multiply(array a, array b) | Returns element-wise multiplication of two given arrays |
| np.dot(array a, array b) | Returns scalar or dot product of two given arrays |

**1. Numpy Matrix multiplication element wise.**

If you want element-wise matrix multiplication, you can use multiply() function.

**import numpy as np**

**arr1 = np.array([[1, 2],**

**[3, 4]])**

**arr2 = np.array([[5, 6],**

**[7, 8]])**

**arr\_result = np.multiply(arr1, arr2)**

**print(arr\_result)**

**Output:**

**[[ 5 12]**

**[21 32]]**

**2)Matrix Product of two numpy arrays.**

If you want the matrix product of two arrays, use matmul() function.

**import numpy as np**

**arr1 = np.array([[1, 2],**

**[3, 4]])**

**arr2 = np.array([[5, 6],**

**[7, 8]])**

**arr\_result = np.matmul(arr1, arr2)**

**print(f'Matrix Product of arr1 and arr2 is:\n{arr\_result}')**

**arr\_result = np.matmul(arr2, arr1)**

**print(f'Matrix Product of arr2 and arr1 is:\n{arr\_result}')**

**Output:**

**Matrix Product of arr1 and arr2 is:**

**[[19 22]**

**[43 50]]**

**Matrix Product of arr2 and arr1 is:**

**[[23 34]**

**[31 46]]**

**3)Dot product of two numpy arrays.**

The numpy dot() function returns the dot product of two arrays. The result is the same as the matmul() function for one-dimensional and two-dimensional arrays.

**import numpy as np**

**arr1 = np.array([[1, 2],**

**[3, 4]])**

**arr2 = np.array([[5, 6],**

**[7, 8]])**

**arr\_result = np.dot(arr1, arr2)**

**print(f'Dot Product of arr1 and arr2 is:\n{arr\_result}')**

**arr\_result = np.dot(arr2, arr1)**

**print(f'Dot Product of arr2 and arr1 is:\n{arr\_result}')**

**arr\_result = np.dot([1, 2], [5, 6])**

**print(f'Dot Product of two 1-D arrays is:\n{arr\_result}')**

**Output:**

**Dot Product of arr1 and arr2 is:**

**[[19 22]**

**[43 50]]**

**Dot Product of arr2 and arr1 is:**

**[[23 34]**

**[31 46]]**

**Dot Product of two 1-D arrays is:**

**17**

Q6: Before you buy a home, which numpy function will you use to measure your monthly mortgage payment?

Ans: In order to calculate the monthly mortgage payment, you will use the **numpy function .pmt**(rate, nper, pv) where:

* rate = The periodic (monthly) interest rate.
* nper = The number of payment periods (months) in the lifespan of the mortgage loan.
* pv = The total value of the mortgage loan.

Q7: Can string data be stored in numpy arrays? If so, list at least one restriction that applies to this data.

Ans: The dtype of any numpy array containing string values is the maximum length of any string present in the array. Once set, it will only be able to store new string having length not more than the maximum length at the time of the creation. If we try to reassign some another string value having length greater than the maximum length of the existing elements, it simply discards all the values beyond the maximum length.