

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

Ans: - To increase the comparison between different figures on the same graph, you can use various types of comparison charts. Some of these include Comparison Bar Charts, Slope Charts, Progress Charts, Tornado Charts, Pie Charts, Double Bar Graphs, Donut Charts, Matrix Charts, Stacked Bar Charts, and Dual Axis Grouped Bar Charts. Each of these charts serves a specific purpose and can be used based on the nature of the data and the kind of comparison you want to make.

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

Ans: -Compound interest is beneficial over a higher rate of simple interest that does not compound because it allows your wealth to grow at a faster rate. This is due to the fact that compound interest is calculated not only on the initial principal but also on the accumulated interest from previous periods. This means that you earn interest on the money you invest, as well as on the returns at the end of every compounding period. Over time, this can significantly boost your investment returns.

Q3. What is a histogram, exactly? Name a numpy method for creating such a graph.

Ans: -. A histogram is a graphical representation of the distribution of data. It is represented by a set of rectangles, adjacent to each other, where each bar represents a kind of data. The numpy method for creating such a graph is `numpy.histogram`

Q4. If necessary, how do you change the aspect ratios between the X and Y axes?

Ans: -To change the aspect ratios between the X and Y axes, you can use the `set_aspect()` function in `matplotlib`. Another method is to manually calculate the aspect ratio and set it using the `ax.set_aspect()` function in `matplotlib`, where `ax` is the instance of the axes class.

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: - The three types of array multiplication between two numpy arrays are dot product, outer product, and regular multiplication.

- Dot product: If both arrays are 1-D, it is the inner product of vectors. If both arrays are 2-D, it is matrix multiplication.
- Outer product: It results in a matrix that is the product of the magnitude of the vectors. The number of rows of the matrix is the number of elements in the first vector, and the number of columns is the number of elements in the second vector.
- Regular multiplication: It is an element-wise multiplication. The shape of the two arrays needs to be the same or broadcastable.

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

Ans: - Before buying a home, you can use the `numpy.pmt()` function to calculate your monthly mortgage payment.

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

Ans: -Yes, string data can be stored in numpy arrays. However, there are some restrictions. 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 strings having a length not more than the maximum length at the time of the creation.