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

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

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

* A histogram is a graph that shows the frequency or number of values that fall within specified ranges, known as bins. This type of graph is often used to represent the distribution of a dataset. To create a histogram in Numpy, you can use the numpy.histogram() method. This method takes in a dataset and the desired number of bins, and returns the binned data and the edges of the bins.

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

To change the aspect ratio between the X and Y axes in a plot, you can use the plt.axis() method from Matplotlib. This method takes in several parameters that control the range of values shown on each axis, as well as the aspect ratio.

For example, to set the X axis to range from 0 to 10 and the Y axis from 0 to 5, with a 1:2 aspect ratio (i.e. the Y axis is twice as long as the X axis), you could use the following code:

*import matplotlib.pyplot as plt*

*plt.axis([0, 10, 0, 5], aspect=1/2)*

The aspect parameter specifies the ratio of the length of the Y axis to the length of the X axis. In this case, we set it to 1/2 to make the Y axis twice as long as the X axis.

**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.**

* The dot product, also known as the scalar product or inner product, is a binary operation that takes two arrays and returns a scalar value. It is computed as the sum of the element-wise product of the two arrays. The arrays must have the same shape and size, and the resulting scalar value is equivalent to the magnitude of the two arrays multiplied together.
* The outer product, on the other hand, is a binary operation that takes two arrays and returns a new array with dimensions equal to the sum of the dimensions of the two input arrays. It is computed by taking the element-wise product of the two arrays, then arranging the resulting products in a new array according to the dimensions of the input arrays.
* Regular multiplication of two numpy arrays, on the other hand, is a binary operation that takes two arrays and returns a new array with the same dimensions as the input arrays. It is computed by multiplying each element in one array by the corresponding element in the other array, resulting in a new array with the same shape and size as the input arrays.

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

In order to calculate the monthly mortgage payment, we’ll 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.

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

Yes, string data can be stored in numpy arrays. However, there are a few restrictions that apply to storing string data in numpy arrays. One restriction is that all of the strings in a numpy array must be the same length. This means that if you want to store a mixture of strings of different lengths in a numpy array, you will need to pad the shorter strings with whitespace or some other character so that they are all the same length. Additionally, numpy arrays can only store strings of a fixed length, so if you have strings that are very long, they may not be able to fit in a numpy array.