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.

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

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.

Q6. Before you buy a home, which numpy function will you use to measure 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.

A1. There are several ways to increase the comparison between different figures on the same graph. One way is to use consistent scales on both the x and y axes, which makes it easier to compare the relative sizes of different figures. Another way is to use different colors or patterns to differentiate between different figures. Additionally, using labels, titles, and legends can help to clarify the information presented on the graph.

A2. Compound interest refers to interest that is calculated on both the principal amount and any accumulated interest. The benefit of compound interest is that over time, the interest earned on the accumulated interest can become significant, leading to a higher overall return on investment. In contrast, a higher rate of interest that does not compound will only earn interest on the principal amount, resulting in a lower overall return.

A3. A histogram is a graphical representation of the distribution of a set of continuous data. It consists of a series of bars that represent the frequency of data within specific intervals or "bins." In NumPy, the `histogram` function can be used to create a histogram graph.

A4. To change the aspect ratios between the X and Y axes, you can use the `aspect` parameter in the `plt.subplots()` function. This parameter takes a float value representing the ratio of the height to width of the subplot. For example, to set the aspect ratio to 1:2, you would use `plt.subplots(aspect=0.5)`.

A5. The dot product of two NumPy arrays is a way of multiplying them element-wise and then summing the results. It produces a scalar value as output. The outer product of two

NumPy arrays creates a new array by multiplying each element of one array by each element of the other array, resulting in a new array with dimensions equal to the product of the dimensions of the input arrays. Regular multiplication of two NumPy arrays multiplies each element of one array by the corresponding element in the other array. The main difference between these three types of array multiplication is the shape of the resulting output.

A6. To measure your monthly mortgage payment, you can use the NumPy `pmt` function. This function takes as input the interest rate, the number of payments, and the principal amount and returns the monthly payment required to pay off the loan.

A7. Yes, string data can be stored in NumPy arrays. However, one restriction that applies to this data is that all of the strings in the array must have the same length. This is because NumPy arrays are designed to be homogeneous, meaning that all elements in the array should have the same data type and shape. If the strings in the array have different lengths, it can lead to issues with indexing and accessing the data.