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. To increase the comparison between different figures on the same graph, one can adjust the scale of the axis or use different colors, patterns, or symbols for each figure. One can also add labels or annotations to highlight important features of each figure.

A2. The benefit of compound interest over a higher rate of interest that does not compound is that compound interest generates more returns over time. With compound interest, the interest earned is added to the principal amount, and the interest for the next period is calculated based on the new total amount. This leads to exponential growth of the investment over time. In contrast, a higher rate of interest that does not compound only generates linear growth of the investment over time.

A3. A histogram is a graph that displays the frequency distribution of a continuous variable. It consists of a set of rectangles, where the area of each rectangle corresponds to the frequency of observations within a particular interval or bin. Numpy provides the method `numpy.histogram()` for creating a histogram.

A4. To change the aspect ratio between the X and Y axes in a matplotlib plot, one can use the `axis` method with the parameter `aspect`. For example, to set the aspect ratio to 2:1, one can use the following code: `plt.axis('equal')`.

A5. The dot product of two numpy arrays is the matrix multiplication of the arrays. It returns a single value or an array of values depending on the shape of the arrays. The outer product of two numpy arrays is the matrix of all possible combinations of the elements of the two arrays. It returns an array with a shape equal to the product of the shapes of the input arrays. Regular multiplication of two numpy arrays is element-wise multiplication. It returns an array with the same shape as the input arrays.

A6. To measure the monthly mortgage payment before buying a home, one can use the numpy function `numpy.pmt()`. This function calculates the payment for a loan with a fixed interest rate, fixed term, and fixed periodic payments.

A7. Yes, string data can be stored in numpy arrays. However, the main restriction is that numpy arrays are designed to store homogeneous data, which means that all elements in the array must have the same data type. Therefore, if a numpy array is used to store string data, all elements in the array must be strings of the same length. This can lead to wasted memory if some strings are shorter than others. Another restriction is that numpy arrays do not support variable-length strings, which means that the maximum length of the strings must be fixed and specified in advance.