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

There are several ways to increase the comparison between different figures on the same graph:

- Using different colors for each figure.

- Using different line styles (dotted, dashed, solid, etc.) for each figure.

- Using markers to indicate data points for each figure.

- Using different shades of the same color for each figure.

- Using labels and a legend to clearly identify each figure.

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

Compound interest allows the interest earned on an investment to be added back to the principal, resulting in a larger balance that earns even more interest in the future. In contrast, simple interest only earns interest on the original principal amount, so the total interest earned over time is lower. Therefore, compound interest can lead to higher returns on an investment over time, even if the interest rate is lower than a simple interest rate that does not compound.

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

A histogram is a graphical representation of the distribution of a dataset. It consists of a set of rectangles (or bars) whose area is proportional to the frequency of a variable and whose width is equal to the range of the variable. It is commonly used to show the distribution of continuous numerical data.

In numpy, the method for creating a histogram is `numpy.histogram()`.

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

To change the aspect ratios between the X and Y axes, you can use the `aspect` parameter of the `matplotlib.pyplot.subplots()` function. The `aspect` parameter takes a string or a float value. If a string value is provided, it can be one of three values: `'equal'` (default), `'auto'`, or `'manual'`. If a float value is provided, it is the ratio of the Y-axis length to the X-axis length.

For example, to set the aspect ratio of the plot to 1:2, you can use the following code:

```

import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.plot(x, y)

# Set the aspect ratio to 1:2

ax.set\_aspect(0.5)

plt.show()

```

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.

- Regular multiplication: This is element-wise multiplication of two numpy arrays. The resulting array has the same shape as the input arrays, and each element is the product of the corresponding elements in the input arrays.

- Dot product: This is a form of matrix multiplication between two numpy arrays. The dot product of two arrays `a` and `b` is calculated as `np.dot(a, b)` or `a.dot(b)`. The resulting array has a shape that depends on the dimensions of the input arrays. If both arrays are 1D arrays, the dot product is the same as the inner product. If both arrays are 2D arrays, the dot product is the matrix multiplication of the two arrays.

- Outer product: This is also a form of matrix multiplication between two numpy arrays, but it results in a different shape than the dot product. The outer product of two arrays `a` and `b` is calculated as `np.outer(a, b)`. The resulting array has a shape that is the product of the shapes of the input arrays. Each element in the resulting array is the product of a corresponding element in `a` and all the elements in `b`.