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

When comparing different figures on the same graph, there are several choices to increase the comparison:

1. Use consistent scales: Ensure that the scales of the axes are consistent across different figures. This allows for a fair visual comparison between the data points. Inconsistent scales can distort the perception of the data and lead to incorrect interpretations.
2. Adjust plot limits: If the range of values in the data varies significantly between figures, you can adjust the plot limits to focus on the relevant range. By zooming in or out on specific sections of the plot, you can enhance the comparison between the figures.

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 refers to the concept of earning interest on both the initial principal amount and the accumulated interest from previous periods. In contrast, a higher rate of interest that does not compound would only earn interest on the initial principal.

The benefit of compound interest is that it allows for exponential growth of savings or investments over time. As the interest is added to the principal and then earns further interest, the total amount grows at an increasing rate.

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 displays the frequencies or counts of observations within specified intervals, called bins, along the horizontal axis. The height or vertical axis represents the frequency or count of observations falling within each bin.

Numpy provides a method called histogram() to create a histogram. This method takes an array of data values and returns the frequencies and bin edges. The bin edges define the intervals for the histogram bins. Here's an example:

import numpy as np

import matplotlib.pyplot as plt

# Generate some random data

data = np.random.randn(1000)

# Create a histogram

hist, bins = np.histogram(data, bins=10)

# Plot the histogram

plt.hist(data, bins=10)

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.title('Histogram')

plt.show()

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.axes().set\_aspect() method from the matplotlib.pyplot module. This method allows you to set the aspect ratio by specifying a ratio value.

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.

dot product yields a scalar value, the outer product yields a matrix, and regular multiplication performs element-wise multiplication. The requirements for shape compatibility differ between these operations, and the resulting shapes and values also differ.

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

To calculate the monthly mortgage payment before buying a home, you can use the NumPy function np.pmt(). The np.pmt() function is used to calculate the periodic payment for a loan or mortgage based on a fixed interest rate, loan amount, and loan duration.

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 certain restrictions and considerations when working with string data in numpy arrays:

1. Fixed-length strings: Numpy arrays require fixed-length strings. This means that all strings within a numpy array must have the same length. If a string exceeds the specified length, it will be truncated. This restriction ensures that the array has a uniform shape and size.
2. Performance impact: Working with large arrays of strings can have a performance impact compared to numerical data. String operations often involve more memory and computational overhead, so operations on string arrays may be slower compared to numerical arrays.