**ADITYA AMIN**

**ASSIGN : 23**

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, you can employ various techniques to enhance the visual comparison. Here are some choices you can consider:

Adjusting the Scale: If the figures have different scales or ranges, you can adjust the axis limits or scale to ensure a fair visual comparison.

Using Different Colors or Markers: Assigning distinct colors or markers to different figures can aid in visually differentiating them

Adding Legends or Labels: Including legends or labels on the graph can provide clear identification for each figure, making it easier to associate specific lines, bars, or data points with their corresponding data series.

Employing Annotations or Text: Adding annotations or text directly onto the graph can highlight important features or provide additional context.

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

Growth of Investments: Compound interest allows investments to grow exponentially over time. As interest is continuously added to the principal amount and then reinvested, the investment has the potential to grow at an increasing rate.

Time Value of Money: Compound interest takes into account the time value of money. By reinvesting the interest earned, you are effectively utilizing the power of time to generate more wealth.

Enhanced Wealth Accumulation: Compound interest facilitates wealth accumulation over the long term. Even with a lower interest rate but continuous compounding, the accumulated interest can compound significantly over time and surpass the returns of a higher 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 provides a visual representation of the frequency or count of observations falling into different intervals or bins.

In a histogram, the x-axis represents the range of values in the dataset, divided into equal intervals or bins. The y-axis represents the frequency or count of observations that fall within each bin. The height of each bar in the histogram represents the number of observations in that particular bin.

import numpy as np

import matplotlib.pyplot as plt

# Generate a random dataset

data = np.random.normal(0, 1, 1000)

# Create a histogram

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

# Plot the histogram

plt.hist(data, bins=10, edgecolor='black')

# Add labels and title

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.title('Histogram of Random Data')

# Show the plot

plt.show()

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

import matplotlib.pyplot as plt

# Create a plot

plt.plot([1, 2, 3], [4, 5, 6])

# Adjust the aspect ratio

plt.axis('equal') # equal aspect ratio

# or

plt.axis('scaled') # scaled aspect ratio

# Show the plot

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.

Dot Product:

The dot product is a mathematical operation that calculates the sum of the element-wise products of two arrays.

It is performed using the numpy.dot() function or the dot method of an array object.

The dot product requires that the two arrays have compatible dimensions, typically one-dimensional or two-dimensional arrays.

Outer Product:

The outer product calculates the tensor product or Kronecker product between two arrays.

It is performed using the numpy.outer() function.

The outer product requires that the two arrays have compatible shapes, typically one-dimensional arrays.

Regular Multiplication:

Regular multiplication, also known as element-wise multiplication or Hadamard product, multiplies corresponding elements of two arrays.

It is performed using the \* operator or the numpy.multiply() function.

Regular multiplication requires that the two arrays have the same shape or compatible broadcasting rules.

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

NumPy does not have a specific function to calculate monthly mortgage payments directly. However, NumPy can be used in conjunction with other mathematical operations and formulas to calculate mortgage payments.

To calculate your monthly mortgage payment, you can use the following formula:

M = P \* (r \* (1 + r)\*\*n) / ((1 + r)\*\*n - 1)

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

Fixed-Length Strings: NumPy arrays require fixed-length strings, which means all the strings in the array must have the same length. This length is determined by the maximum length of any string in the array. If a string exceeds this maximum length, it will be truncated.

Performance Impact: Storing and manipulating string data in NumPy arrays can have performance implications. NumPy arrays are designed to work efficiently with homogeneous numerical data, and handling variable-length strings or large amounts of string data may result in decreased performance compared to numerical computations.