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

**Ans:**

-Plot Multiple Line Plots with Different Scales, Plot Multiple Line Plots with Multiple Y-Axis, using hexcodes, opacity options, shading options.

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

**Ans:** compound interest basically calculates interest over interest amount, while higher rate of interest will calculate interest only on principle amount.

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

**Ans:** histogram is basically to draw the frequency of an item between a certain range given. Np.histogram is the method in numpy to create such histogram graphs.

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

**Ans:** Aspect ratio is the ratio of height to width of the image we want to display. Matplotlib provides us the feature of modifying the aspect ratio of our image by specifying the value for the optional aspect ratio attribute for our image plot.

We can change aspect ratio using below approach:

import matplotlib.pyplot as plt

plt.imshow(im, aspect='0.5') – where im is the image of graph

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.

**Ans:**

**Dot product -** This function returns the dot product of two arrays. For 2-D vectors, it is the equivalent to matrix multiplication. For 1-D arrays, it is the inner product of the vectors. For N-dimensional arrays, it is a sum product over the last axis of a and the second-last axis of b.

**Outer product –** if there are 2 arrays of size m and n then their outer product will be of size m\*n. Given two vectors of size m × 1 and n × 1 respectively

u = [ u 1 u 2 ⋮ u m ] , v = [ v 1 v 2 ⋮ v n ]

their outer product, denoted u ⊗ v,[1] is defined as the m × n matrix A obtained by multiplying each element of u by each element of v:

u ⊗ v = A = [ u 1 v 1 u 1 v 2 … u 1 v n u 2 v 1 u 2 v 2 … u 2 v n ⋮ ⋮ ⋱ ⋮ u m v 1 u m v 2 … u m v n ]

**Regular multiplication:** It returns the product of arr1 and arr2, element-wise.

import numpy as np

in\_arr1 = np.array([[2, -7, 5], [-6, 2, 0]])

in\_arr2 = np.array([[0, -7, 8], [5, -2, 9]])

out\_arr = np.multiply(in\_arr1, in\_arr2)

[[ 0 49 40]

[-30 -4 0]]

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

**Ans:** pmt() method of numpy.

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

**Ans:** to store string data in numpy array, length of the string should b fixed. If it will exceed the defined length, extra string will be cutoff.