Q1. What is a probability distribution, exactly? If the values are meant to be random, how can you predict them at all?

Solution - A probability distribution is a mathematical function that describes the likelihood of different outcomes or events in a random process or experiment. It provides a systematic way to represent the probabilities associated with various possible values of a random variable.

In a probability distribution, each possible value of the random variable is assigned a probability, which represents the likelihood of that value occurring. The probabilities assigned to different values of the random variable must satisfy certain conditions, such as summing up to 1.

While the values generated from a probability distribution are inherently random, the distribution itself characterizes the probabilities associated with each value. This allows us to make predictions or infer certain properties about the random process.

Q2. Is there a distinction between true random numbers and pseudo-random numbers, if there is one? Why are the latter considered "good enough"?

Solution - Yes, there is a distinction between true random numbers and pseudo-random numbers.

True random numbers are generated from physical processes that are inherently unpredictable, such as atmospheric noise, radioactive decay, or thermal noise. They are considered to be truly random because their values cannot be determined or predicted, even with complete knowledge of the system generating them. True random numbers are often used in cryptography, simulations, and other applications where high-quality randomness is essential.

On the other hand, pseudo-random numbers are generated by deterministic algorithms. These algorithms use a starting value called a seed and apply mathematical operations to generate a sequence of numbers that appear random. However, the generated sequence is actually deterministic and repeatable if the same seed is used. Pseudo-random numbers are generated using algorithms that have good statistical properties and exhibit randomness in their distribution and sequence. They are designed to mimic the properties of true random numbers, but they are not truly random.

Pseudo-random numbers are considered "good enough" for many applications because they exhibit statistical properties similar to true randomness. They pass various statistical tests for randomness and can provide a high level of unpredictability, especially when using high-quality algorithms and carefully chosen seed values.

Q3. What are the two main factors that influence the behaviour of a "normal" probability distribution?

Solution - The two main factors that influence the behavior of a "normal" probability distribution are the mean and the standard deviation.

Mean (μ): The mean represents the central tendency or average of the distribution. It determines the location of the peak of the distribution. In a normal distribution, the mean is also the median and the mode. Shifting the mean to the left or right will result in the distribution being centered around a different value.

Standard Deviation (σ): The standard deviation measures the spread or variability of the distribution. It determines the width of the distribution curve. A smaller standard deviation indicates that the data points are closely clustered around the mean, resulting in a narrower distribution. Conversely, a larger standard deviation indicates greater dispersion of data points, resulting in a wider distribution

Q4. Provide a real-life example of a normal distribution.

Solution - One real-life example of a normal distribution is the distribution of human heights. In many populations, heights tend to follow a normal distribution pattern. The majority of individuals have heights around the mean, with fewer individuals being either significantly taller or shorter.

Q5. In the short term, how can you expect a probability distribution to behave? What do you think will happen as the number of trials grows?

Solution - In the short term, the behavior of a probability distribution can be unpredictable and subject to fluctuations. The outcomes of individual trials may deviate from the expected probabilities due to the inherent randomness involved. However, as the number of trials increases, the behavior of the probability distribution tends to stabilize and converge towards the expected probabilities.

Q6. What kind of object can be shuffled by using random.shuffle?

Solution - The random.shuffle function in Python can be used to shuffle the elements of a sequence in-place. It can shuffle objects that are iterable and mutable, such as lists.

Some examples of objects that can be shuffled using random.shuffle include:
Lists: random.shuffle(my_list)
Arrays: random.shuffle(my_array)
Mutable sequences: random.shuffle(my_sequence)

Q7. Describe the math package's general categories of functions.

Solution - The math package in Python provides various mathematical functions for performing common mathematical operations. The functions in the math package can be grouped into several general categories:

Basic Arithmetic Functions: These functions perform fundamental arithmetic operations like addition, subtraction, multiplication, and division. Examples include math.add, math.subtract, math.multiply, and math.divide.

Trigonometric Functions: The math package provides functions for calculating trigonometric values such as sine, cosine, tangent, arc sine, arc cosine, and arc tangent. Examples include math.sin, math.cos, math.tan, math.asin, math.acos, and math.atan.

Exponential and Logarithmic Functions: These functions deal with exponential and logarithmic calculations. The math package includes functions like exponentiation (math.exp), natural logarithm (math.log), logarithm with base 10 (math.log10), and power (math.pow).

Common Mathematical Constants: The math package provides constants for important mathematical values such as π (pi) (math.pi), Euler's number (math.e), and the square root of 2 (math.sqrt(2)).

Numeric Functions: These functions handle numeric operations and transformations. Examples include rounding (math.round), absolute value (math.abs), maximum (math.max), minimum (math.min), and integer conversion (math.floor and math.ceil).

Special Functions: The math package also includes special mathematical functions such as factorial (math.factorial), combinatorial functions (math.comb and math.perm), and rounding towards the nearest integer (math.rint).

Q8. What is the relationship between exponentiation and logarithms?

Solution - Exponentiation and logarithms are inverse operations of each other and are closely related in mathematics.

Exponentiation involves raising a base number to a certain power. For example, in the expression "a raised to the power of b" or "a^b," a is the base and b is the exponent. Exponentiation represents repeated multiplication, where the base is multiplied by itself b times.

Logarithms, on the other hand, are used to find the exponent or power to which a given base must be raised to obtain a specific number. The logarithm of a number y to a base a, denoted as "log base a of y" or "log_a(y)," represents the exponent to which the base a must be raised to obtain the value y.

Q9. What are the three logarithmic functions that Python supports?

Solution - Python supports three logarithmic functions as part of the math module:

Natural Logarithm (ln): The natural logarithm function calculates the logarithm of a number to the base 'e' (Euler's number). It is denoted as math.log(x) and returns the natural logarithm of the argument 'x'.

Base-10 Logarithm (log10): The base-10 logarithm function calculates the logarithm of a number to the base 10. It is denoted as math.log10(x) and returns the base-10 logarithm of the argument 'x'.

Custom Base Logarithm (log): The logarithm function with a custom base calculates the logarithm of a number to a specified base. It is denoted as math.log(x, base) and returns the logarithm of the argument 'x' to the base specified by 'base'.