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

predict them at all?

Q2. Is there a distinction between true random numbers and pseudo-random numbers, if there is

one? Why are the latter considered “good enough”?

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

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

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?

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

Q7. Describe the math package&#39;s general categories of functions.

Q8. What is the relationship between exponentiation and logarithms?

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

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

A probability distribution is a mathematical function that describes the likelihood of different outcomes in a random experiment. It provides a framework to determine the probability that a random variable will take on a particular value or fall within a certain range. While the individual values are random and unpredictable, the distribution allows you to make predictions about the overall behavior of the random variable over many trials. This means that while you can't predict a single outcome, you can predict the relative frequencies of different outcomes over time.

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

Yes, there is a distinction between true random numbers and pseudo-random numbers. **True random numbers** are generated from a physical process, such as radioactive decay or thermal noise, and are inherently unpredictable. **Pseudo-random numbers**, on the other hand, are generated by deterministic algorithms that produce sequences of numbers that only appear random. Despite being deterministic, pseudo-random numbers are considered "good enough" for most applications because they pass statistical tests for randomness and can be reproduced, which is useful for debugging and consistency in simulations.

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

The two main factors that influence the behavior of a normal probability distribution are:

1. **Mean (μ):** The mean is the central value around which the distribution is centered. It determines the location of the peak of the distribution on the horizontal axis.
2. **Standard Deviation (σ):** The standard deviation measures the spread or dispersion of the distribution. A smaller standard deviation results in a steeper, narrower curve, while a larger standard deviation produces a flatter, wider curve.

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

A real-life example of a normal distribution is human height. In a large population, the distribution of adult heights tends to follow a bell curve, with most people having a height close to the average and fewer people being significantly taller or shorter than the average.

### **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?**

In the short term, the outcomes of a probability distribution may vary widely, and the distribution might not appear to match the expected theoretical pattern. However, as the number of trials grows (according to the Law of Large Numbers), the observed distribution will converge more closely to the theoretical probability distribution, and the relative frequencies of different outcomes will stabilize.

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

The random.shuffle function can shuffle any mutable sequence, such as a list. It randomly reorders the elements of the sequence in place.

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

The math package in Python includes functions that fall into several general categories:

* **Arithmetic functions:** Basic operations like abs, ceil, floor, and modf.
* **Trigonometric functions:** Functions like sin, cos, tan, and their inverses, as well as conversions between degrees and radians.
* **Logarithmic and exponential functions:** Functions like log, log10, exp, and pow.
* **Special functions:** Advanced functions like factorial, gamma, and hypot for various mathematical computations.
* **Constants:** Mathematical constants like pi and e.

### **Q8. What is the relationship between exponentiation and logarithms?**

Exponentiation and logarithms are inverse operations. If ab=ca^b = cab=c, then the logarithm base aaa of ccc is bbb (i.e., log⁡a(c)=b\log\_a(c) = bloga​(c)=b). Exponentiation raises a number to a power, while logarithms determine the power to which the base must be raised to produce a given number.

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

Python supports the following three logarithmic functions:

1. **math.log(x, base):** Returns the logarithm of x to the specified base. If the base is not specified, it defaults to the natural logarithm (base e).
2. **math.log10(x):** Returns the base-10 logarithm of x.
3. **math.log2(x):** Returns the base-2 logarithm of x.