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 "normal" 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's general categories of functions.

Q8. What is the relationship between exponentiation and logarithms?

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

Answers

## 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 possible outcomes for a random variable. It provides a way to assign probabilities to each possible value that the variable can take, ensuring that the total probability sums to 1. While the individual outcomes may be random and unpredictable, the distribution allows us to make predictions about the behavior of the variable over many trials. For instance, we can determine the expected value (mean) and variance of the outcomes, which helps in understanding the overall behavior of the random variable in a probabilistic sense.

## 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 inherently random physical processes, such as radioactive decay or thermal noise, making them unpredictable and not reproducible. In contrast, pseudo-random numbers are generated by algorithms that produce sequences of numbers that appear random but are actually deterministic and reproducible if the initial conditions (seed) are known. Pseudo-random numbers are considered "good enough" for most applications, including simulations and statistical sampling, because they can approximate the properties of true randomness closely enough for practical purposes, and they are much faster and easier to generate.

## 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 determines the center of the distribution, indicating where the peak of the bell curve is located. It represents the average value of the data.

2. \*\*Standard Deviation (σ)\*\*: The standard deviation measures the spread or dispersion of the distribution. A smaller standard deviation results in a steeper curve, indicating that the data points are closer to the mean, while a larger standard deviation results in a flatter curve, indicating that the data points are more spread out.

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

A real-life example of a normal distribution is the distribution of heights in a population. In many populations, heights tend to cluster around an average value (the mean), with fewer individuals being extremely short or extremely tall. This results in a bell-shaped curve when plotted, where most individuals' heights fall within one standard deviation of the mean, and progressively fewer individuals fall outside that range.

## 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 behavior of a probability distribution can be quite variable, with outcomes appearing random and potentially deviating significantly from expected values. However, as the number of trials grows (according to the Law of Large Numbers), the observed frequencies of outcomes will converge to the theoretical probabilities defined by the distribution. This means that the average of the outcomes will approach the expected value, and the distribution will become more stable and predictable.

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

The `random.shuffle` function can be used to shuffle \*\*mutable sequences\*\* in Python, such as lists. It randomly rearranges the elements of the list in place, altering the original order. For example:

```python

import random

my\_list = [1, 2, 3, 4, 5]

random.shuffle(my\_list)

print(my\_list) # The order of elements will be randomly changed

```

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

The `math` package in Python provides a variety of mathematical functions, which can be categorized into the following general categories:

1. \*\*Basic Arithmetic Functions\*\*: Functions for basic operations like addition, subtraction, multiplication, and division (e.g., `math.add`, `math.subtract`).

2. \*\*Trigonometric Functions\*\*: Functions that perform trigonometric calculations (e.g., `math.sin`, `math.cos`, `math.tan`).

3. \*\*Exponential and Logarithmic Functions\*\*: Functions for calculating exponentials and logarithms (e.g., `math.exp`, `math.log`, `math.log10`).

4. \*\*Power and Square Root Functions\*\*: Functions for raising numbers to a power and calculating square roots (e.g., `math.pow`, `math.sqrt`).

5. \*\*Constants\*\*: Mathematical constants such as π (pi) and e (Euler's number) are also provided (e.g., `math.pi`, `math.e`).

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

Exponentiation and logarithms are inverse operations. Exponentiation involves raising a base to a power, while logarithms determine the exponent needed to achieve a certain value from a given base. Mathematically, if $$ b^y = x $$, then $$ \log\_b(x) = y $$. This means that the logarithm of $$ x $$ to the base $$ b $$ gives the exponent $$ y $$ such that $$ b $$ raised to $$ y $$ equals $$ x $$.

## 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 no base is provided, 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 $$ x $$.

These functions allow for flexibility in calculating logarithms with different bases.