Calcular a possibilidade de obter o número 6 em uma amostra de 5000 lançamentos de um dado.

import numpy as np

lancamentos = np.random.randint(1, 7, size=5000)

prob\_estimada = np.mean(lancamentos == 6)

print(prob\_estimada)

import numpy as np

np.random.randint(1, 7, size=5000)

1. Função np.random.randint

np.random.randint(low, high, size) gera números inteiros aleatórios no intervalo [low, high).

low: valor mínimo inclusive.

high: valor máximo exclusive (ou seja, não é incluído).

size: quantidade de números que você quer gerar (pode ser um int ou uma tupla para arrays multidimensionais).

2. Aplicando aos seus valores

np.random.randint(1, 7, size=5000)

1 → valor mínimo incluído.

7 → valor máximo não incluído, então os números gerados vão de 1 a 6.

size=5000 → serão gerados 5.000 números inteiros aleatórios.

Em outras palavras, isso simula 5.000 lançamentos de um dado de 6 lados.

3. Resultado

Tipo de dado: numpy.ndarray

Formato: (5000,) → vetor 1D com 5.000 elementos

Valores possíveis: 1, 2, 3, 4, 5, 6

Exemplo de saída (parcial, aleatória):

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

lancamentos = np.random.randint(1, 7, size=5000)

np.mean(lancamentos == 6)

np.mean(lancamentos == 6)

1. lancamentos == 6

Supondo que lancamentos seja um array NumPy (como o que você gerou com np.random.randint(1, 7, size=5000)),

lancamentos == 6 cria um array booleano, onde cada posição será:

True se o valor for 6

False caso contrário

Exemplo:

lancamentos = np.array([3, 6, 1, 6])

lancamentos == 6

# array([False, True, False, True])

**2. np.mean(...) aplicado a booleanos**

* Em NumPy, True é tratado como 1 e False como 0.
* Portanto, np.mean(lancamentos == 6) calcula a **proporção de elementos iguais a 6** no array.

Exemplo:

np.mean(np.array([False, True, False, True]))

# (0 + 1 + 0 + 1) / 4 = 0.5

**3. Interpretação**

Se lancamentos tem 5000 elementos, np.mean(lancamentos == 6) retorna aproximadamente a **frequência relativa de 6** nos lançamentos, ou seja, a **probabilidade experimental de sair 6**:

* Para um dado justo, esperamos que seja próximo de 1/6 ≈ 0.1667.