

Problemas em equipe 01

Estudantes:

**Eduardo Eiji Goto,
Gustavo Hammerschmidt,
João Vitor Andrioli.**

- 1) A probabilidade de o Messi marcar o gol na cobrança de um pênalti é $2/3$, a probabilidade de o Cristiano Ronaldo marcar o gol na cobrança de um pênalti é $4/5$, e a probabilidade de o Neymar marcar o gol na cobrança de um pênalti é $7/10$. Se cada um vai cobrar um pênalti, calcule a probabilidade de apenas um marcar o gol.

$$P[\text{Messi}] = 2/3$$

$$P[\text{Neymar}] = 7/10$$

$$P[\text{CR7}] = 4/5$$

$$P[\text{Messi}'] = 1/3$$

$$P[\text{Neymar}'] = 3/10$$

$$P[\text{CR7}'] = 1/5$$

$$\begin{aligned} P[\text{Apenas um marcar o gol}] = & P[\text{Messi}] * P[\text{Neymar}'] * P[\text{CR7}'] + \\ & P[\text{Messi}'] * P[\text{Neymar}] * P[\text{CR7}'] + \\ & P[\text{Messi}'] * P[\text{Neymar}'] * P[\text{CR7}] \end{aligned}$$

$$\begin{aligned} P[\text{Apenas um marcar o gol}] = & (((((2/3) * 3) / 10) * 1) / 5) + \\ & (((((1/3) * 7) / 10) * 1) / 5)) + \\ & (((((1/3) * 3) / 10) * 4) / 5) \end{aligned}$$

$$P[\text{Apenas um marcar o gol}] = 0.04 + 0.0467 + 0.08$$

$$P[\text{Apenas um marcar o gol}] = 0.1667$$

$$P[\text{Apenas um marcar o gol}] = 16.67\%$$

2) Cálculo de probabilidade por simulação.

As probabilidades são calculadas a partir de um modelo de probabilidades (fórmula matemática). Uma outra maneira de resolver o problema é realizar o experimento aleatório muitas vezes, e observar a proporção de vezes que o evento ocorre. Dessa maneira estamos calculando a probabilidade por simulação.

Nesse exercício vamos trabalhar com um problema bem simples, que é calcular a probabilidade de se lançar dois dados e obter os números 3 e 6. Podemos calcular a probabilidade intuitivamente contando as possibilidades de ocorrer os números 3 e 6 e o total de possibilidades para dois dados.

Se representarmos o lançamento de dois dados por um par de números (d1, d2), onde d1 é o resultado do dado 1 e d2 o resultado do dado 2, temos a seguinte tabela de possibilidades, onde podemos observar que existem duas chances em 36, ou seja, a probabilidade é igual a $2/36 = 0,0556$.

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

Podemos usar a biblioteca numpy para simular o problema.

O comando `np.random.randint(1,7,(10,2))` sorteia dois valores aleatórios maiores ou iguais a 1 e menores do que 7 (dois primeiros parâmetros) e coloca em uma matriz com uma linha e duas colunas (terceiro parâmetro). Por exemplo, poderíamos simular 10 lançamentos de dados e salvar na matriz sorteio com os seguintes comandos:

```
import numpy as np
sorteio = np.random.randint(1,7,(10,2))
print(sorteio)
[[6 5]
 [3 6]
 [4 3]
 [3 2]
 [1 5]
 [4 1]
 [4 2]
 [5 1]
 [3 3]
 [6 4]]
```

Nesse exemplo, no 10 lançamento não observamos nenhuma vez uma dupla de resultados contendo os números 3 e 6.

Para simular o sorteio podemos programar o seguinte algoritmo:

```
def Dado(n):  
    sorteio = sortear n duplas de dados  
    deuCerto = 0  
    para i de 1 até n  
        sortear dado1  
        sortear dado2  
        se ((dado1 == 3) E (dado2 == 6)) OU ((d1 == 6) E (d2 == 3))  
            deuCerto = deuCerto + 1  
    imprimir ('Probabilidade simulada = deuCerto/n)
```

Utilizar a biblioteca numpy. Com a biblioteca random pode não funcionar.

Implementar a função Dado em Python.

Copiar o código aqui (não enviar o arquivo .py):

[Programa Dado.py - Start]

```
import numpy as np
```

```
throws = lambda tries: np.random.randint(1, 7, (tries, 2))
```

```
dados = lambda throws, diceA, diceB:  
    (sum([1 for throw in throws if (throw[0] == diceA and throw[1] ==  
    diceB) or (throw[1] == diceA and throw[0] == diceB)]/len(throws))
```

```
run = lambda: [print(_) for _ in
```

```
    ["\n\tDado.py>>>",  
    "\n\n\t\tProbabilidade simulada:\t\t{:.4f}".format(dados(throws(10000), 3, 6)),  
    "\n\t\tProbabilidade teórica:\t\t{:.4f}".format(2/36),  
    "\n\n"]
```

```
]
```

```
run()
```

[Programa Dado.py - End]

Obs.: Mantenha todas as funções lambda em uma só linha: estruturada dessa forma para melhor compreensão.