Exercícios - Probabilidade

Questão 1

Um teste para COVID-19 possui uma taxa de 5% de falso positivo e 1% de falso negativo, ou seja, em 5% dos não-infectados acusa como infectados, e em 1% dos infectados acusa como não infectados. Sabendo que atualmente 1% da população está infectada, calcule a probabilidade de um paciente cujo teste apontou positivo estar realmente infectado.

Resposta:

	Pessoa Doente	Pessoa Não Doente
Teste positivo	95 %	5 %
Teste negativo	1 %	99 %

P(D) = 0,01 (1% da população está Doente)
P(¬D) = 0,99 (99% da população Não está Doente)

	Pessoa Doente	Pessoa Não Doente
Prob. Teste positivo	P(D) . 95%	P(¬D) . 5%
Prob. Teste negativo	P(D) . 1%	P(¬D) . 99%

	Pessoa Doente	Pessoa Não Doente
Prob. Teste positivo	0,0095	0,0495
Prob. Teste negativo	0,0001	0,9801

Total Positivo (novo espaço amostral) = 0,0095 + 0,0495 = 0,059 = 5,9%
Doentes de verdade: 95%

P = 0,95 / 0,059 = 0,161 = **16,1%**

Questão 2

A probabilidade de dado honesto é conhecido e igual a $1/6$, mas quando vamos testar isso com um dado físico, não necessariamente vai cair uma vez cada face do dado (se eu lançar ele 6 vezes). A ideia é que conforme fazermos vários testes a tendência é que a probabilidade de cada face irá convergir pro valor esperado.

Vamos testar este conceito utilizando da programação seguintes os itens abaixo:

A) Desenvolva uma função que receba um valor n, onde n é a quantidade de elementos e o objetivo da função é retorna uma *Series* com n elementos sendo esses elementos números variando de 1 e 6;

B) Teste a função para n igual a 10, 100, 1.000, 10.000 e 100.000 casos e para cada um deles calcule a proporção de vezes que aparece cada número;

DIca.: utilize a função do *Pandas* `.value_counts`

C) O que podermos dizer sobre a afirmação do enunciado, a tendência é que as probabilidades irão convergir para $1/6$?

In [1]:

```
import numpy as np
import pandas as pd

def gera_serie(n):
    return pd.Series(np.random.randint(1, 6, n))

print("Probabilidade 1/6:",1/6)
print("=" * 38)

for n in range(1,7):
    qtd = 10**(n)
    print(qtd,":")
    print(gera_serie(qtd).value_counts(normalize=True))
```

```
Probabilidade 1/6: 0.16666666666666666
======================================
10 :
3    0.3
4    0.3
1    0.2
2    0.1
5    0.1
dtype: float64
100 :
4    0.25
5    0.21
3    0.20
1    0.18
2    0.16
dtype: float64
1000 :
1    0.207
3    0.206
2    0.200
4    0.199
5    0.188
dtype: float64
10000 :
3    0.2060
5    0.2060
4    0.2006
2    0.1940
1    0.1934
dtype: float64
100000 :
1    0.20215
5    0.20080
2    0.20007
4    0.19929
3    0.19769
dtype: float64
1000000 :
2    0.200860
3    0.200076
5    0.200072
1    0.200003
4    0.198989
dtype: float64
```

Resposta: Sim, a probabilidade converte para 1/6

Questão 3

Na casa de Luiza e Rafael tem um pote de balas de diferentes cores. Certo dia a mamãe deles contou quantas balas tinha de cada uma das cores:

- 14 balas azuis;
- 22 balas vermelhas;
- 17 balas amarelas.

A mamãe deles deixou que cada um pegasse 3 balas começando pela Luiza e depois o Rafael. Dado isso resolva os item abaixo:

A) Qual a probabilidade da Luiza pegar uma bala de cada na ordem da contagem e o Rafael pegar 2 azuis e uma vermelha, dado que é um evento sem reposição?

B) Luiza decidiu ficar só com a bala amarela e devolveu as demais para o pote antes que o Rafael pegasse suas balas. Como que ficaria a probabilidade para o Rafael agora?

Respostas:
A)
P(Az) = 14/53 = 0,2642%
P(Ve) = 22/52 = 0,4231%
P(Am) = 17/51 = 0,3333%
P(Az E Ve E Am) = P(Az) *P(Ve)* P(Am) = 5236 / 14556 = 0,03725 = **3,72%**

Restariam, nesse caso, 13 balas azuis, 21 balas vermelhas e 16 balas amarelas (50 balas)

P(Az) = 13/50 = 0,2600
P(Az) = 12/49 = 0,2449
P(Ve) = 21/48 = 0,4375
P((Az E Az E Ve) OU P(Az E Ve E Az) OU (Ve E Az e Az)) = 3 * 3276 / 117600 = 0,08357 = **8,36%**

B)
Se Luiza ficou apenas com a bala amarela, antes de Rafael pegar suas balas, teremos:
14 balas azuis;
22 balas vermelhas;
16 balas amarelas.

P(Az) = 14/52 = 0,2500
P(Az) = 13/51 = 0,2352
P(Ve) = 22/50 = 0,4200
P((Az E Az E Ve) OU P(Az E Ve E Az) OU (Ve E Az e Az)) = 3 * 4004 / 132600 = 0,0906 = **9,06%**

Questão 4

Um programa de computador gera uma pontuação aleatória para cada usuário, sendo que:

- Em 1/2 dos casos, gera 0.5;
- Em 1/4 dos casos, gera 0.25;

- Em 1/8 dos casos, gera 0.125.

E essa lógica segue assim por diante. Resolva os itens a seguir:

A) Calcule o **valor esperado** utilizando apenas os 3 primeiros casos;

B) Faça um *loop* para calcular o **valor esperado** para 100 casos e avalie se só com os 3 primeiros casos temos uma boa aproximação ou não.

A)Valor esperado = (1/2 *0.5) + (1/4* 0.25) + (1/8 * 0.125) = **0.328125**

In [3]:
```
(1/2 * 0.5) + (1/4 * 0.25) + (1/8 * 0.125)
```

Out[3]:
```
0.328125
```

B) Sim, os 3 primeiros representam uma ótima aproximação.

In [14]:
```
ve = 0
for i in range(1,100):
    valor = (1/(2**i))**2
    ve += valor
print('ve =',ve)
```

```
ve = 0.3333333333333333
```

In [16]:
```
print('0,328125 / 0.333333 = ', round(0.328125 / 0.333333 * 100, 2),"%")
```

```
0,328125 / 0.333333 =  98.44 %
```

Questão 5

Em um programa de auditório, havia 3 portas (A, B, C), dentro da qual uma continha um prêmio. O participante escolheu a porta A e antes de abrir a porta o apresentador abriu a porta C, na qual não havia nada. O participante aumentará a probabilidade de acertar a porta caso mude a escolha dele da porta A para a B?

Resposta: SIM.

P(A) = 1/3 = 0,333
P(Ac) = 2/3 = 0,675

A primeira escolha representa uma possiblidade de 1/3 de acerto.
Quando o apresentador abre uma das portas sem prêmio, que não é a porta escolhida, ele reduz o universo de possibilidades.
A porta escolhida ainda possui 1/3 de probabilidade de acerto. Já a outra porta, agora, representa 1/2 de chance de acerto.

In [ ]: