

Python Básico

Andre Nepomuceno

June 15, 2021

Exercícios - Semana 03

3.1 A lei de Benford (ou lei do primeiro dígito) é uma distribuição de frequência dos primeiros dígitos de números em diversos conjuntos de dados. A distribuição é dada por:

$$P(d) = \log_{10} \left(\frac{d+1}{d} \right)$$

onde d é o primeiro dígito do número. Assim, num grande conjunto de números, aqueles que começam com 1 são mais frequentes em relação aos que começam com 2, que não mais frequentes que os iniciados por 3, e assim por diante, sendo os números iniciados por 9 os mais raros. As probabilidades de ocorrência são:

1	0.301
2	0.176
3	0.125
4	0.097
5	0.079
6	0.067
7	0.058
8	0.051
9	0.046

O arquivo "proteinas.py", disponível no *classroom*, contém o comprimento de sequências de aminoácidos de 500 proteínas escolhidas aleatoriamente. Os dados estão numa lista "naa" que pode ser importada para um programa da seguinte forma: **from** proteínas **import** naa. Escreva um programa para verificar se o comprimento de proteínas segue a lei de Benford.

Sugestão: uma das possíveis soluções é usar o método de lista `count(element)`.

Curiosidade: para uma aplicação da lei de Benford as eleições, veja RBE.

3.2 Os valores de resistência e tolerância de um resistor são indicados por um código de cores: as duas primeiras cores indicam os primeiros dois dígitos, a terceira um decimal multiplicador, e a quarta, se houver, indica a incerteza no valor. O mapa (cor,valor) é mostrado na figura abaixo.

Cor:	1ª Faixa:	2ª Faixa:	Multiplicador:	Tolerância:
Preto	0	0	1	$\pm 20\%$
Marrom	1	1	10	$\pm 1\%$
Vermelho	2	2	100	$\pm 2\%$
Laranja	3	3	1000	$\pm 3\%$
Amarelo	4	4	10000	$\pm 4\%$
Verde	5	5	100000	-
Azul	6	6	-	-
Violeta	7	7	-	-
Cinza	8	8	-	-
Branco	9	9	-	-
Dourado	-	-	0,1	$\pm 5\%$
Prateado	-	-	0,01	$\pm 10\%$

Por exemplo, um resistor com bandas violeta, amarelo, vermelho e dourado, terá resistência $74 \times 10^2 \Omega$ com tolerância de 5%, ou seja, $R = (7400 \pm 370) \Omega$.

Escreva um programa que traduza a lista de cores de um resistor para os valores de resistência e tolerância (para cores que não tem tolerância, use 20%). Por exemplo, seu programa pode retornar uma tupla com o par (resistência, tolerância):

```
>>> valor_resistencia(['violeta','amarelo','vermelho','dourado'])  
(7400,370)
```

Compare o resultado do seu programa com: Calculadora

3.3 Dado os arrays:

```
a = np.array ( [1, 2, 3, 5] )  
b = np.array( [2, 4, 6, 8] )
```

Escreva o resultado das operações abaixo antes de executá-las no computador.

```
b/a +1  
b/(a+1)  
1/a  
np.dot(a, b)
```

3.4 A posição e velocidade de um objeto que cai sobre a influência da gravidade e resistência do ar são dadas, respectivamente, por:

$$y(t) = -s_t [t + \tau(e^{-t/\tau} - 1)]$$

$$v(t) = -s_t \left[1 - \exp\left(-\frac{t}{\tau}\right) \right]$$

onde $s_t = 20$ m/s é a velocidade terminal e $\tau = 5$ s é uma constante de tempo. Utilizando arrays, escreva um programa que calcule a posição e velocidade do objeto a cada 0.5 segundos, durante 20 segundos. Assuma $y(0) = 0$.

3.5 Dado o array $A = \text{np.arange}(20)$, deduza o resultado dos comandos abaixo antes de executá-los.

```
A[:]  
A[::]  
A[5:15]  
A[5:15:3]  
A[5::]  
A[:5:]  
A[::5]
```