Problemas em Equipe 03

Estudantes: Gustavo Hammerschmidt.

Parte 1 – Variáveis aleatórias contínuas

1) Um sistema gera mensagens com tamanhos uniformemente distribuídos entre 200 e 800 bytes. Qual é a probabilidade de uma mensagem ter tamanho entre 300 e 500 bytes?

```
==========PYTHON
import scipy.stats as st

print(st.uniform.cdf(500, 200, 600) - st.uniform.cdf(300, 200, 600))
==========OUTPUT
0.33333333333333337
[Finished in 594ms]
==========
```

Probabilidade de 33,33337%

2) O tempo de processamento de transações em um servidor web é distribuído exponencialmente com média igual a 10 milissegundos. Qual a probabilidade de uma transação levar entre 5 e 15 milissegundos?

```
==========PYTHON
import scipy.stats as st

print(st.expon.cdf(15,0,10) - st.expon.cdf(5,0,10))
==========OUTPUT
0.38340049956420363
[Finished in 609ms]
==========
```

Probabilidade de 38,34%

3) A duração de certos tipos de amortecedores, em km rodados é normalmente distribuída, possui duração média de 5000 km e desvio-padrão de 1000 km. Qual a probabilidade de um amortecedor escolhido ao acaso durar entre 4500 e 6350 km?

==========PYTHON

```
import scipy.stats as st

print(st.norm.cdf(6350,5000,1000)  - st.norm.cdf(4500,5000,1000))
```

==========OUTPUT

```
0.6029544698366112

[Finished in 566ms]
```

==========

Probabilidade de 60,29%

Parte 2 - Simulação do problema do aniversário

Deseja-se calcular a probabilidade de em um grupo com n pessoas, duas ou mais fazerem aniversário no mesmo dia e no mesmo mês, mas não necessariamente no mesmo ano.

Cálculo teórico

A probabilidade pode ser calculada pela seguinte expressão:

$$P[A] = 1 - \frac{P(365, n)}{365^n} = \frac{365}{365} \cdot \frac{364}{365} \cdot \frac{363}{365} \cdot \ldots \cdot \frac{365 - n + 1}{365}$$

A função aniverT calcula essa probabilidade.

```
import numpy as np
def aniverT(tGrupo):
    x = np.arange(365, 365 - tGrupo, -1, dtype = float)
    return(1 - np.prod(x)/(365**tGrupo))
```

Simulação interativa

A probabilidade pode ser calculada pelo seguinte algoritmo iterativo:
n = quantidade de pessoas no grupo
nSim = quantidade de simulações
deuCerto = 0
Realizar os seguintes passos nSim vezes
  Sortear n datas de aniversários (sortear n números inteiros aleatórios uniformemente distribuídos entre 1 e 365)
  Verificar se houve duas datas iguais
  Caso posititivo, somar 1 a deuCerto
Probabilidade simulada = deuCerto / nSim

A função aniver calcula essa simulação:

```
def aniver(tGrupo, nSim):
    deuCerto = 0
    for i in range(nSim):
        # sorteia grupo com tGrupo pessoas
        grupo = np.random.randint(1, 366, tGrupo)
        # se duas ou mais pessoa fazem aniver na mesma data
        if grupo.size > np.unique(grupo).size:
            deuCerto = deuCerto + 1
    return(deuCerto/nSim)
```

Simulação vetorial

A probabilidade pode ser calculada pelo seguinte algoritmo vetorial:

Sortear uma matriz com nSim linhas e tGrupo colunas (cada linha contém a simulação de um grupo).

Ordenar as linhas da matriz (as datas de aniversário de um grupo).

Calcular a diferença entre duas datas consecutivas (dica: usar função np.diff). Observação: se alguma diferença de datas no grupo for zero significa que houve aniversário no mesmo dia.

Calcular um vetor que contenha True ou False para cada linha da matriz (cada grupo sorteado). Será True, se alguma diferença na linha for 0, False caso contrário.

Retornar a quantidade de elementos no vetor com valor True dividido por nSim (dica: usar a função np.count_nonzero para contar a quantidade de elementos com valor True).

Programar a função que implementa a simulação vetorial.

Dica: usar o arquivo Aniver.ipynb

Copiar seu código aqui. Não enviar o arquivo Aniver.ipynb.

```
def aniverV(tGrupo, nSim):

    # sortear nSim grupos de tamanho tGrupo
    grupos = np.array([np.random.randint(1, 366, tGrupo) for i in r
ange(nSim)])

    # ordena
    ordenada = np.sort(grupos, 1)

    # calcula a diferenca entre as datas adjacentes
    diferenca = np.diff(grupos, 1)
    # se diferenca for zero tem aniver na mesma data

    # mesmaData eh um array unidimensional com valor verdadeiro
    # para cada grupo que tiver dois ou mais aniversarios no mesmo
dia
    mesmaData = [ not i for i in np.any(diferenca == 0, 1)]

    # conta quantos grupos tiveram aniversario no mesmo dia, divide
 por nsim e retorna
    return np.count_nonzero(mesmaData) / nSim
```

Obs.: Desconsidere a quebra das linhas, é a formatação do word.

Resultado:

```
probT = aniverT(40)
t1 = time.perf_counter()
probS = aniverV(40, 10000)
t2 = time.perf_counter()
print('Probabilidade simulada:  {:.4f}'.format(probS))
print('Probabilidade teórica: {:.4f}'.format(probT))
print('Tempo de simulação vetorial: {:.4f}'.format(t2-t1))
```

```
Probabilidade simulada:  0.8955
Probabilidade teórica: 0.8912
Tempo de simulação vetorial: 0.1434
```