



escola
britânica de
artes criativas
& tecnologia

Módulo | Análise de Dados: Fundamentos de Matemática

Caderno de **Exercícios**

Professor [André Perez](#)

Tópicos

1. Vetorização;
2. Arrays Numpy;
3. Operações.

Exercícios

1. Tráfego de São Paulo

Neste exercício, vamos analisar dados de mobilidade urbana da cidade de São Paulo. A base de dados está neste [link](#) e é uma copia do dado original, presente neste [link](#). A base de dados contem a quantidade de acidentes ocorridos na cidade entre 14/12/09 e 18/12/09, das 07:00h às 20:00h, agregados em intervalos de 30 minutos.

```
In [ ]: !wget -q "https://raw.githubusercontent.com/andre-marcos-perez/ebac-course-
```

```
In [ ]: import pandas as pd
```

```
In [ ]: df = pd.read_csv('traffic.csv', sep=';')
```

```
In [ ]: df.head()
```

- **Exemplo:**

O código abaixo extrai a 13ª linha do arquivo que representa a 14ª meia hora do dia 14/12/09, contadas a partir das 07:00h, ou seja, todos os incidentes ocorridos na cidade no dia em questão entre as 13:30h e 14:00h. Vemos, por exemplo, que dois ônibus e dois caminhões ficaram parados nas ruas da cidade, atrapalhando o tráfego.

In []:

```
df.iloc[[13]]
```

1.1. Nativo

Nesta primeira atividade, você deve ler o código abaixo implementado em Python nativo e responder a seguinte pergunta:

O que o código abaixo computa?

Resposta: substitua esse texto pela sua resposta

In []:

```
# -- read

data = None

with open(file='traffic.csv', mode='r', encoding='utf8') as fp:

    fp.readline()
    data = fp.read()

# -- analytics

day = 14
incidents = 0
incident_by_day = dict()

for timebox in data.split(sep='\n'):

    timebox_data = timebox.split(sep=';')

    # --
    # -- inicio da computação escalar
    # --

    for incident in timebox_data[1: len(timebox_data)-1]:
        incidents = incidents + int(incident)

    # --
    # -- fim da computação escalar
    # --

    try:

        half_hour = int(timebox_data[0])

        if half_hour == 27:
            incident_by_day[day] = incidents
            day = day + 1
            incidents = 0

    except ValueError:
        continue

# -- results

for day in incident_by_day:

    print(f'{day}: {incident_by_day[day]}')
```

1.2. NumPy

Substitua o trecho do código do algoritmo que utiliza da **computação escalar** por um que utiliza da **computação vetorial**. Use o pacote NumPy.

In []:

```
# -- read

data = None

with open(file='traffic.csv', mode='r', encoding='utf8') as fp:

    fp.readline()
    data = fp.read()

# -- analytics

import ...

day = 14
incidents = 0
incident_by_day = dict()

for timebox in data.split(sep='\n'):

    timebox_data = timebox.split(sep=';')

    # --
    # -- inicio da computação vetorial
    # --

    ...

    # --
    # -- fim da computação vetorial
    # --

    try:

        half_hour = int(timebox_data[0])

        if half_hour == 27:
            incident_by_day[day] = incidents
            day = day + 1
            incidents = 0

    except ValueError:
        continue

# -- results

for day in incident_by_day:

    print(f'{day}: {incident_by_day[day]}')
```
