

# Projeto Prático 1:

## Análise Climatológica:

Descrições dos desafios e dicas abaixo:

Baixe o arquivo **dados\_climatologicos.csv** para começar!

### 1 - Cálculo da Média e Desvio Padrão:

Calcule e imprima na tela a temperatura e precipitação médias ao longo do ano. Além disso, determine o desvio padrão para obter uma ideia da variação dos dados. Ou seja, imprima na tela a média e desvio padrão das temperaturas e a média e desvio padrão das precipitações.

### 2 - Dias Chuvosos:

Calcule e imprima na tela quantos dias tiveram uma precipitação acima de 10mm. Estes são considerados "dias chuvosos".

### 3 - (Bônus) - Variação de Temperatura:

Calcule e imprima na tela a diferença entre a temperatura mais alta e a mais baixa de cada mês. Isso ajudará a entender a amplitude térmica mensal.

### 4 - (Bônus) - Precipitação Mensal:

Calcule e imprima na tela o total de precipitação para cada mês do ano.

---

## Dicas:

### 1 - Cálculo da Média e Desvio Padrão:

Calcule e imprima na tela a temperatura e precipitação médias ao longo do ano. Além disso, determine o desvio padrão para obter uma ideia da variação dos dados. Ou seja, imprima na tela a média e desvio padrão das temperaturas e a média e desvio padrão das precipitações.

#### Dicas

- Calcule a temperatura e precipitação médias ao longo do ano.
- Determine o desvio padrão para obter uma ideia da variação dos dados.
- **Dica:** Use `np.mean()` para calcular a média e `np.std()` para o desvio padrão.
- **Dica:** Lembre-se de aplicar o `np.mean` em todas as linhas, porém somente na coluna do dado que precisa, ex:
  - Caso precise acessar todas os valores da segunda linha(temperatura), faça `dados[:, 0]`
  - Caso precise acessar todas os valores da segunda linha(precipitação), faça `dados[:, 1]`
  - Se precisar saber da média da primeira coluna, faça `np.mean(dados[:, 0])`
  - Se precisa da segunda coluna `np.mean(dados[:, 1])` e assim sucessivamente

### 2 - Dias Chuvosos:

Calcule e imprimir na tela quantos dias tiveram uma precipitação acima de 10mm. Estes são considerados "dias chuvosos".

#### Dicas

- Identifique quantos dias tiveram uma precipitação acima de 10mm. Estes são considerados "dias chuvosos".
  - Caso precise acessar todas os valores da segunda linha(precipitação), faça `dados[:, 1])`
- **Dica:** Utilize a indexação booleana para filtrar os dias com precipitação acima de 10mm e, em seguida, utilize `np.sum()` para contar esses dias.

### 1. Temperaturas Extremas:

Encontre o dia mais quente e o dia mais frio do ano, juntamente com suas respectivas temperaturas.

Dicas

- Encontre o dia mais quente e o dia mais frio do ano, juntamente com suas respectivas temperaturas.
- **Dica:** Use `np.argmax()` para encontrar o índice do dia mais quente e `np.argmin()` para o dia mais frio.

## 3 - (Bônus) - Variação de Temperatura:

Calcule e imprima na tela a diferença entre a temperatura mais alta e a mais baixa de cada mês. Isso ajudará a entender a amplitude térmica mensal.

**Dicas:**

1. Para calcular a amplitude térmica de um mês, você precisa da temperatura máxima e mínima desse mês.
2. Lembre-se de que o conjunto de dados representa temperaturas diárias ao longo de um ano. Divida o ano em meses. Para simplificar, considere cada mês tendo 30 dias.
3. Utilize a indexação para acessar as temperaturas de cada mês. Por exemplo, os primeiros 30 dias representam o mês 1, os próximos 30 representam o mês 2 e assim por diante.
4. Use a função `np.max()` para encontrar a temperatura máxima de um mês e `np.min()` para encontrar a temperatura mínima.
5. A amplitude térmica é a diferença entre a máxima e a mínima.
6. **List Comprehension:** No Python, você pode usar a compreensão de listas para gerar listas de maneira concisa. Neste caso, você deseja criar uma lista onde

cada item é a amplitude térmica de um mês. Imagine que você está fazendo um loop `for`, mas de uma forma mais compacta. A estrutura é: `[expressão for item in iterável]`. Aqui, a "expressão" é o cálculo da amplitude térmica, e "item in iterável" refere-se a cada mês (conjunto de 30 dias) no conjunto de dados.

#### a. Passo 1: Acessar a Coluna de Temperaturas

Para trabalhar com as temperaturas, selecionem a primeira coluna do array `dados`:

```
temperaturas = dados[:, 0]
```

`dados[:, 0]` acessa todos os elementos da primeira coluna.

#### Passo 2: Selecionar uma Janela de Dados

Vocês precisarão selecionar uma janela de 30 dias dos dados para análise:

```
temperaturas[i:i+30]
```

Este código pega 30 dias de dados de temperatura, começando do índice `i`.

#### Passo 3: Calcular a Amplitude Térmica

Calcule a amplitude térmica (diferença entre a máxima e a mínima temperatura) para cada janela de 30 dias:

```
np.max(janela) - np.min(janela)
```

#### Passo 6: Compreensão de Listas para Calcular Amplitudes

Agora, em vez de usar um loop `for` para calcular a amplitude para cada mês, tentem usar a compreensão de listas. Aqui está a lógica:

- Usem um loop `for` dentro de uma compreensão de lista para iterar sobre o array `temperaturas`.
- Em cada iteração, calculem a amplitude térmica para uma janela de 30 dias usando as funções `np.max()` - `np.min()` em cima de cada conjunto

de 30 dias

- Repita isso por todo o conjunto de dados, usando a função `range()`
- Lembrem-se de que a compreensão de lista deve resultar em uma nova lista contendo todas as amplitudes calculadas.

## 4 - (Bônus) - Precipitação Mensal:

Calcule e imprima na tela o total de precipitação para cada mês do ano.

### Dicas:

1. Assim como na tarefa anterior, considere cada mês tendo 30 dias.
2. Use a função `np.sum()` para somar todos os valores de precipitação de um mês.
3. Acesse os valores de precipitação para cada mês usando indexação. Por exemplo, para o primeiro mês, você acessaria os primeiros 30 valores de precipitação.
4. **List Comprehension:** Assim como na tarefa anterior, você pode usar a compreensão de listas para calcular a soma da precipitação de cada mês. A expressão será a soma da precipitação, e você iterará sobre cada conjunto de 30 dias no conjunto de dados.

#### a. 1. Acessar a Coluna de Precipitação

Depois de carregar os dados do arquivo `dados_climaticos.csv` com o NumPy, concentrem-se na coluna que representa a precipitação:

```
precipitacao = dados[:, 1]
```

Esta linha seleciona todos os elementos da segunda coluna de `dados`, que contém as informações de precipitação.

### 2. Definir Janelas de Tempo para Análise

Para analisar a precipitação mensal, vão considerar períodos de 30 dias. Estas serão as suas janelas de tempo para o cálculo.

### 3. Usar Compreensão de Listas para Calcular a Precipitação Mensal

Agora, vocês vão utilizar a compreensão de listas para calcular a soma total da precipitação em cada janela de 30 dias. Aqui está o que vocês precisam considerar:

- Criem uma list comprehension que percorra o array `precipitacao` em janelas de 30 dias.
- Para cada janela de 30 dias, calculem a soma total da precipitação.
- A expressão na list comprehension deve ser algo como `np.sum(precipitacao[i:i+30])`, onde `i` varia de 0 a 365 em passos de 30 dias.
- O resultado será uma lista com a precipitação total para cada mês do ano.