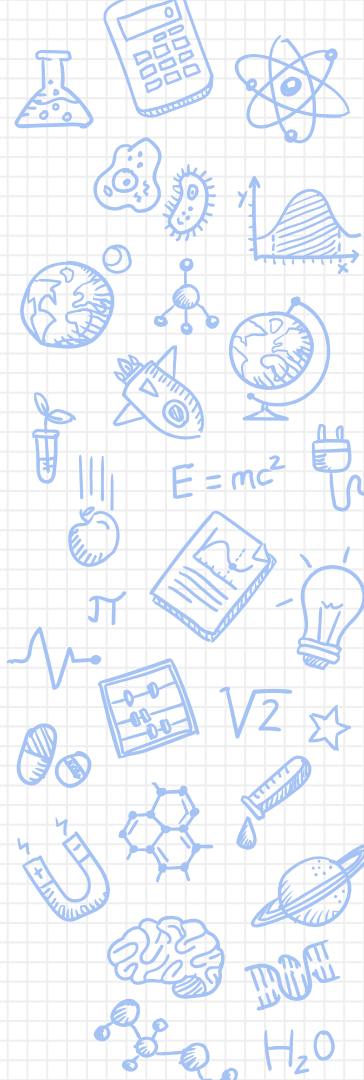


-

Como serão as aulas?

Aulas

- Apresentar, de forma detalhada, as técnicas de estatística utilizando Python e ciência de dados:
 - Teoria;
 - Trabalhos práticos;
 - Frameworks;
 - Estatística.
- **Unidade I** – Apresentação do que é ciência de dados.
- **Unidade II** – Frameworks.
- **Unidade III** – Modelos estatísticos.
- **Unidade IV** – Solução de modelos estatísticos.

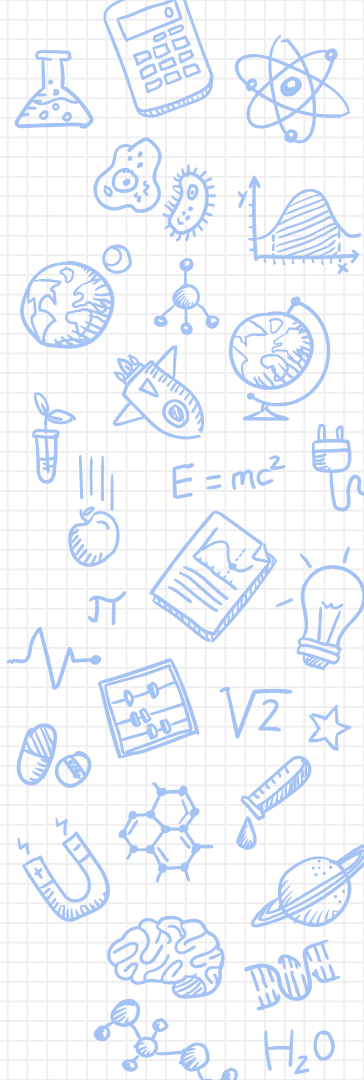


Link: https://github.com/paulovpcotta/aula_l_python_estatis

Removendo info. do DataFrame

Vamos começar falando sobre a remoção de informações de um DataFrame. Informações que não sejam úteis em um dado momento sempre podem ser removidas para facilitar seu trabalho.

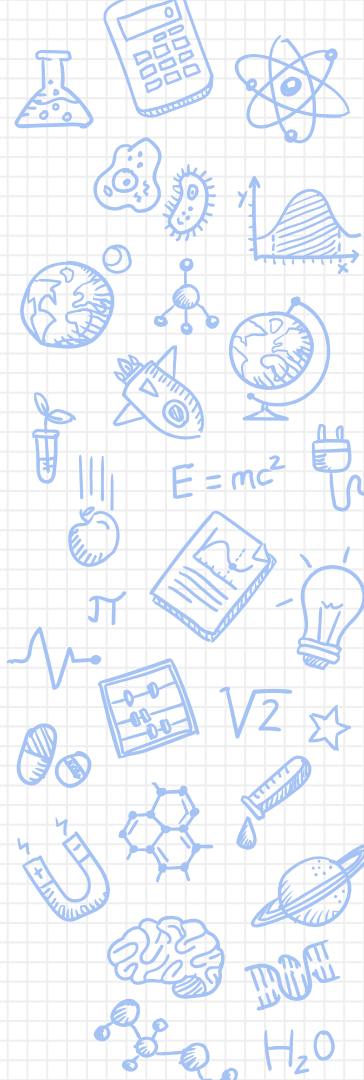
Para remover informações de um DataFrame, use o `drop`. Para o `drop`, você deve incluir como argumento o índice a ser retirado, seja ele nome ou posição. Se você não definir o argumento `axis`, ele remove a linha. Se você definir `axis=1` como argumento, ele remove a coluna. Você também pode passar uma lista como argumento, e todas as linhas ou colunas dentro da lista serão retiradas. É importante notar que ele não remove a informação diretamente do DataFrame em que for usado. Caso você queira manter o `drop`, deve criar um novo objeto junto com o comando.



NaN

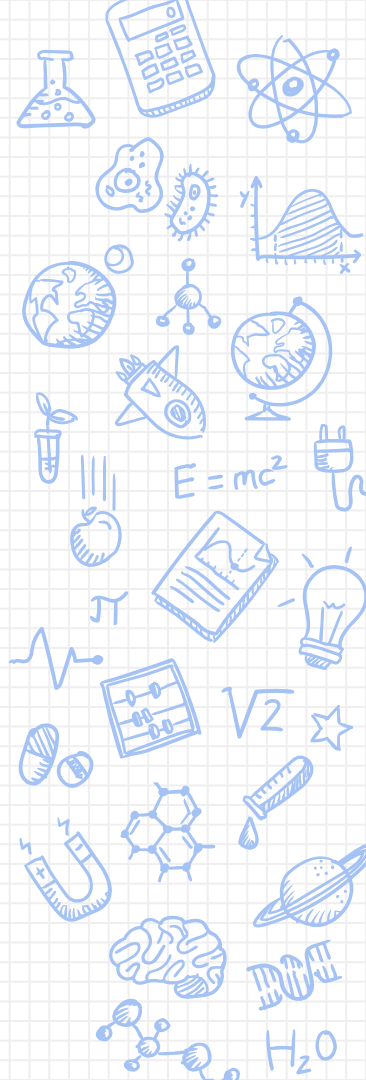
Caso o Dataset em que você esteja trabalhando esteja com valores inexistentes demais, o Pandas possui funções para preencher estes valores inexistentes da melhor forma possível.

Primeiramente, você pode verificar onde no seu DataFrame existem valores não existentes, com a função `isnull()`. No caso do Dataset do Titanic especificamente, a coluna "Cabin" possui muitos valores não existentes, então você pode conferir o comando filtrando esta coluna.



Algumas Estatíst. em DataFrames

O Pandas também possui uma série de comandos para cálculo de estatísticas descritivas ao longo de um DataFrame, podendo calcular ao longo de linhas ou colunas. Entre estas estatísticas estão média, moda, mediana, contagem, soma, entre muitos outros. Por padrão, o cálculo é feito por coluna, mas caso você passe o argumento `axis=1`, o cálculo é feito ao longo das linhas.



Para leitura dos dados, também pode ser importante ordenar um DataFrame, ou seja, fazer um Sort. Para ordenar pelos índices das linhas, vamos primeiro reindexar o DataFrame para que os índices fiquem revertidos. Depois vamos usar a função `sort_index()` para ordená-lo. Você também pode passar o argumento `ascending=False` para ordená-lo de forma decrescente.



Obrigado!