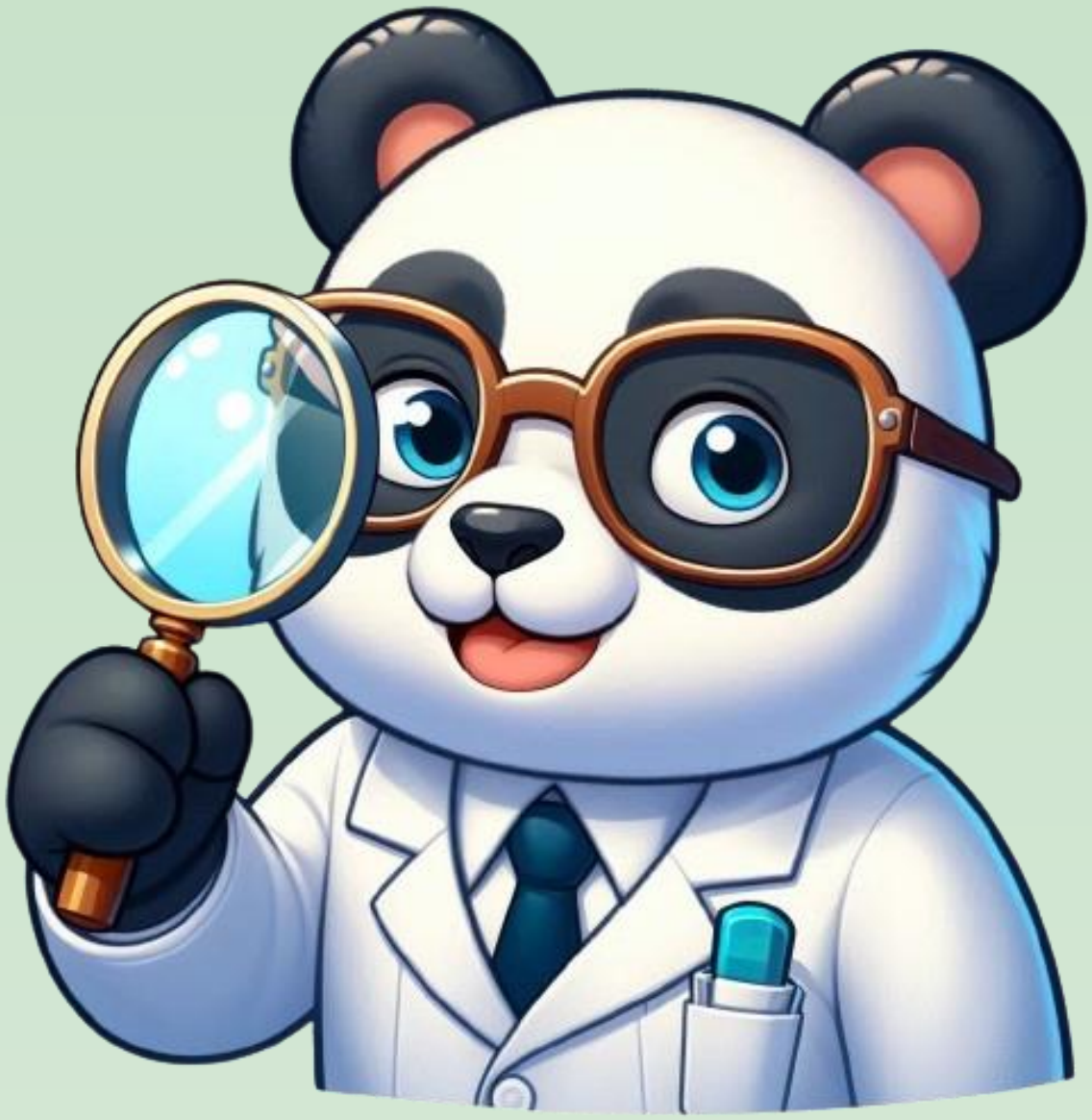
# EXPLORANDO O PANDAS

# GUIA PRÁTICO E SIMPLIFICADO



Aprenda os principais métodos para manipular dados usando Pandas

BRUNO OLIVEIRA

# PRINCIPAIS MÉTODOS DO PANDAS

## Manipulando dados facilmente

Pandas é uma biblioteca poderosa do Python utilizada para análise e manipulação de dados. Ela permite lidar com grandes conjuntos de dados de maneira eficiente e intuitiva.

Vamos explorar os principais métodos do Pandas com exemplos práticos para facilitar o entendimento.

# Carregando Dados com read_csv

Neste exemplo, carregamos um arquivo CSV chamado dados.csv e exibimos as primeiras cinco linhas com head(). Veja o exemplo:

```python
import pandas as pd
# Carregar dados de um arquivo CSV
dados = pd.read_csv('dados.csv')
print(dados.head())
```

# Selecionando Dados com loc e iloc

Selecionar dados específicos é uma tarefa essencial.
Com Pandas, você pode selecionar dados usando rótulos ou índices, usando loc ou iloc. Veja o exemplo:

## Selecionando por Rótulos com loc

```
# Selecionar linha com rótulo específico
linha_especifica = dados.loc[3]
print(linha_especifica)
```

# Selecionando Dados com loc e iloc

## Selecionando um Intervalo com loc

```
# Selecionar uma faixa de linhas e colunas
faixa_dados = dados.loc[1:3, ['coluna1', 'coluna2']]
print(faixa_dados)
```

# Selecionando Dados com loc e iloc

## Selecionando por Índice com iloc

Com iloc, é possível selecionar os dados usando diretamente os índices das linhas e colunas.
Lembre-se que a contagem do índice inicia com zero!

```
# Selecionar a terceira linha (índice 2)
linha_indice = dados.iloc[2]
print(linha_indice)
```

# Selecionando Dados com loc e iloc

## Selecionando um Intervalo com iloc

```python
# Selecionar um intervalo de linhas e colunas por índice
intervalo_dados = dados.iloc[1:3, 0:2]
print(intervalo_dados)
```

# Filtrando Dados com Condições

Filtrar dados com base em condições específicas é muito comum. No Pandas, você pode usar operadores lógicos para especificar as condições.
Veja o exemplo:

```
# Filtrar dados onde a coluna 'idade' é maior que 30
dados_filtrados = dados[dados['idade'] > 30]
print(dados_filtrados)
```

# Adicionando e Removendo Colunas

Para adicionar colunas, é possível declará-las com o novo nome, e criar uma condição matemática ou lógica para preencher os valores. Veja o exemplo

```
# Adicionar uma nova coluna 'nova_coluna' baseada em outra
dados['nova_coluna'] = dados['coluna_existente'] * 2
print(dados.head())
```

# Adicionando e Removendo Colunas

Você também pode remover uma ou mais colunas em uma única operação, usando uma lista de nomes. Veja o exemplo

```
# Remover a coluna 'coluna_existente'
dados = dados.drop(columns=['coluna_existente'])
print(dados.head())
```

# Lidando com Valores Nulos

Tratar valores nulos é crucial para manter a integridade dos dados. Veja o exemplo

```
# Remover linhas com valores nulos
dados_limpos = dados.dropna()
print(dados_limpos.head())
```

# Lidando com Valores Nulos

Tratar valores nulos é crucial para manter a integridade dos dados. Veja o exemplo

```
# Preencher valores nulos com a média da coluna
dados['coluna_com_nulos'].fillna(dados['coluna_com_nulos'].\
mean(), inplace=True)
print(dados.head())
```

# Agrupando Dados com groupby

Agrupar dados ajuda a realizar análises agregadas. Veja o exemplo

```
# Agrupar dados pela coluna 'categoria' e calcular a média
agrupados = dados.groupby('categoria').mean()
print(agrupados)
```

# Mesclando DataFrames com merge

Mesclar DataFrames permite combinar informações de diferentes fontes. Veja o exemplo

```
# Mesclar dois DataFrames pelo valor da coluna 'id'
dados1 = pd.DataFrame({'id': [1, 2, 3], 'valor1': [10, 20, 30]})
dados2 = pd.DataFrame({'id': [1, 2, 3], 'valor2': [40, 50, 60]})

dados_mesclados = pd.merge(dados1, dados2, on='id')
print(dados_mesclados)
```

# Conclusão

Pandas é uma ferramenta indispensável para qualquer analista de dados.

Este guia abordou apenas algumas das muitas funcionalidades que essa biblioteca oferece.

Experimente aplicar esses métodos aos seus próprios conjuntos de dados e veja como Pandas pode transformar a maneira como você analisa informações.