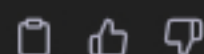




Claro, aqui estão os dez principais métodos para manipulação de DataFrames usando a biblioteca Pandas em Python:



1. `head()` e `tail()`: Esses métodos retornam as primeiras ou últimas linhas do DataFrame, respectivamente. Isso ajuda a verificar os dados rapidamente.

python

Copy code

```
df.head() # Retorna as primeiras linhas do DataFrame (por padrão, as cinco)
df.tail() # Retorna as últimas linhas do DataFrame (por padrão, as cinco úl
```

1. `info()`: Esse método fornece informações sobre o DataFrame, incluindo o tipo de dados das colunas e a contagem de valores não nulos. É útil para entender a estrutura geral dos dados.

python

Copy code

```
df.info() # Retorna informações sobre o DataFrame
```

1. `describe()`: Esse método calcula estatísticas descritivas para as colunas numéricas do DataFrame, como média, mediana, mínimo, máximo, etc.

python

Copy code

```
df.describe() # Retorna estatísticas descritivas para colunas numéricas
```

1. `shape`: Este atributo retorna uma tupla que representa o número de linhas e colunas no DataFrame.

python

Copy code

```
df.shape # Retorna a dimensão do DataFrame (número de linhas, número de col
```

1. `columns`: Este atributo retorna uma lista com os nomes das colunas do DataFrame.

python

Copy code

```
df.columns # Retorna uma lista de nomes de colunas
```

1. `iloc[]` e `loc[]`: Esses métodos permitem acessar dados em um DataFrame por meio de indexação baseada em posição (`iloc[]`) ou rótulos de índice (`loc[]`).

python

Copy code

```
df.iloc[0] # Acessa a primeira linha do DataFrame
df.loc[0] # Acessa a linha com rótulo de índice 0 do DataFrame
df.iloc[:, 1] # Acessa a segunda coluna do DataFrame
df.loc[:, 'coluna'] # Acessa a coluna com rótulo 'coluna' do DataFrame
```

1. `groupby()`: Esse método permite agrupar o DataFrame com base em uma ou mais colunas e, em seguida, aplicar funções de agregação, como soma, média, etc.

python

Copy code

```
grouped = df.groupby('coluna') # Agrupa o DataFrame pela coluna 'coluna'
grouped['outra_coluna'].mean() # Calcula a média da coluna 'outra_coluna' p
```

1. `sort_values()`: Este método ordena o DataFrame com base em uma ou mais colunas.

python

Copy code

```
df.sort_values(by='coluna') # Ordena o DataFrame com base na coluna 'coluna
```

1. `drop()`: Esse método permite remover linhas ou colunas do DataFrame.

python

Copy code

```
df.drop(columns=['coluna']) # Remove a coluna 'coluna' do DataFrame
df.drop(index=[0, 1]) # Remove as linhas com índices 0 e 1 do DataFra
```

1. `fillna()`: Esse método preenche os valores ausentes (NaN) no DataFrame com valores especificados.

python

Copy code

```
df.fillna(0) # Preenche valores ausentes com 0
```

Esses são apenas alguns dos muitos métodos disponíveis no Pandas para manipulação de DataFrames. A escolha do método depende das tarefas específicas que você deseja realizar com seus dados.