Guia do Pandas

1. Leitura e Escrita de Dados

pd.read_csv()

Lê um arquivo CSV para um DataFrame

Parâmetros principais:

- `filepath_or_buffer`: Caminho do arquivo ou URL
- `sep`: Delimitador (padrão ',')
- `header`: Linha a ser usada como cabeçalho (padrão 0)
- `index_col`: Coluna para usar como índice
- `dtype`: Dicionário com tipos de dados para colunas

```
import pandas as pd
df = pd.read_csv('dados.csv', sep=';', encoding='utf-8')
```

pd.read_excel()

Lê uma planilha Excel para um DataFrame

Parâmetros principais:

- `io`: Caminho do arquivo ou objeto ExcelFile
- `sheet_name`: Nome ou índice da planilha
- `header`: Linha do cabeçalho
- `usecols`: Colunas para ler (range ou lista)

```
df = pd.read_excel('dados.xlsx', sheet_name='Vendas')
```

df.to_csv()

Salva o DataFrame em um arquivo CSV

Parâmetros principais:

- `path_or_buf`: Caminho de saída
- `sep`: Delimitador
- `index`: Se deve gravar o índice (padrão True)
- `encoding`: Codificação do arquivo

```
df.to_csv('saida.csv', index=False, encoding='utf-8')
```

pd.read_sql()

Lê dados de uma consulta SQL para um DataFrame

Parâmetros principais:

- `sql`: Consulta SQL ou nome da tabela
- `con`: Conexão com o banco de dados
- `params`: Parâmetros para a consulta

```
import sqlite3
conn = sqlite3.connect('database.db')
df = pd.read_sql('SELECT * FROM clientes', conn)
```

Guia do Pandas

2. Inspeção e Seleção de Dados

df.head() / df.tail()

Mostra as primeiras/últimas n linhas do DataFrame

Parâmetros principais:

```
  n: Número de linhas a mostrar (padrão 5)
```

```
df.head(10)  # Mostra as 10 primeiras linhas
```

df.info()

Mostra informações sobre o DataFrame (tipos de dados, valores não nulos, etc.)

```
df.info()
```

df.type()

Mostra informações mostra o tipo de objeto

```
type(df)
```

df.shape()

Mostra quantas linhas e colunas tem o DataFrame

```
df.shape()
```

df.columns

Mostra todas as colunas do DataFrame

```
df.columns
```

df[‘Coluna’].unique()

Mostra os itens únicos de uma coluna

```
df[‘nome da coluna’].unique()
```

df.describe()

Mostra estatísticas descritivas do DataFrame

Parâmetros principais:

```
  include: Tipos de dados a incluir (padrão apenas numéricos)
  percentiles: Percentis a incluir
```

```
df.describe(include='all')
```

df.loc[]

Acessa um grupo de linhas e colunas por rótulos

```
df.loc[10:20, ['nome', 'idade']]  # Linhas 10 a 20, colunas 'nome' e 'idade'
```

df.iloc[]

Acessa um grupo de linhas e colunas por índices inteiros

```
df.iloc[5:10, 2:4]  # Linhas 5 a 9, colunas 2 e 3
```

Guia do Pandas

3. Manipulação de Dados

df.drop()

Remove linhas ou colunas do DataFrame

Parâmetros principais:

```
  labels: Nomes das colunas ou índices das linhas
  axis: Eixo (0 para linhas, 1 para colunas)
  inplace: Se deve modificar o DataFrame original
```

```
df.drop(['coluna1', 'coluna2'], axis=1, inplace=True)
```

df.rename()

Renomeia colunas ou índices

Parâmetros principais:

```
  columns: Dicionário {antigo: novo} para colunas
  index: Dicionário para índices
  inplace: Modifica o DataFrame original
```

```
df.rename(columns={'old_name': 'new_name'}, inplace=True)
```

df.sort_values()

Ordena o DataFrame por valores

Parâmetros principais:

```
  by: Nome(s) da(s) coluna(s) para ordenação
  ascending: Ordem crescente ou decrescente
  na_position: Posição dos valores nulos ('first' ou 'last')
```

```
df.sort_values(['idade', 'nome'], ascending=[False, True])
```

df.drop_duplicates()

Remove linhas duplicadas

Parâmetros principais:

```
  subset: Colunas a considerar para duplicatas
  keep: Qual duplicata manter ('first', 'last' ou False)
```

```
df.drop_duplicates(subset=['email'], keep='first')
```

df.fillna()

Preenche valores nulos

Parâmetros principais:

```
  value: Valor para substituir nulos
  method: Método de preenchimento ('ffill', 'bfill')
  limit: Número máximo de preenchimentos consecutivos
```

```
df.fillna({'idade': df['idade'].mean()}, inplace=True)
```

4. Agregação e Transformação

df.groupby()

Agrupa dados por uma ou mais colunas

Parâmetros principais:

```
  by: Coluna(s) para agrupamento
  as_index: Se deve usar colunas de agrupamento como índice
```

```
df.groupby('categoria')['preco'].mean()
```

df.pivot_table()

Cria uma tabela dinâmica

Parâmetros principais:

```
  values: Coluna(s) para agregar
  index: Coluna(s) para linhas
  columns: Coluna(s) para colunas
  aggfunc: Função de agregação (padrão 'mean')
```

```
pd.pivot_table(df, values='vendas', index='regiao', columns='mes', aggfunc='sum')
```

df.apply()

Aplica uma função ao longo de um eixo

Parâmetros principais:

```
  func: Função a aplicar
  axis: Eixo (0 para colunas, 1 para linhas)
```

```
df['nome_maiusculo'] = df['nome'].apply(lambda x: x.upper())
```

pd.concat()

Concatena DataFrames ao longo de um eixo

Parâmetros principais:

```
  objs: Lista de DataFrames para concatenar
  axis: Eixo (0 para linhas, 1 para colunas)
  ignore_index: Se deve redefinir o índice
```

```
df_completo = pd.concat([df1, df2, df3], ignore_index=True)
```

pd.merge()

Faz junção de DataFrames (como JOIN em SQL)

Parâmetros principais:

```
  left: DataFrame à esquerda
  right: DataFrame à direita
  on: Coluna(s) para junção
  how: Tipo de junção ('left', 'right', 'inner', 'outer')
```

```
pd.merge(clientes, pedidos, on='cliente_id', how='left')
```

5. Séries Temporais

pd.to_datetime()

Converte para formato datetime

Parâmetros principais:

```
  arg: Dados a converter
  format: Formato da string de data
  errors: Comportamento para erros ('raise', 'coerce', 'ignore')
```

```
df['data'] = pd.to_datetime(df['data_string'], format='%d/%m/%Y')
```

df.resample()

Reamostra série temporal

Parâmetros principais:

```
  rule: Frequência ('D' para dia, 'M' para mês, etc.)
  on: Coluna datetime para usar (se não for o índice)
  aggfunc: Função de agregação
```

```
df.resample('M', on='data')['vendas'].sum()
```

df.rolling()

Calcula estatísticas em janelas móveis

Parâmetros principais:

```
  window: Tamanho da janela
  min_periods: Mínimo de valores necessários
```

```
df['media_movel'] = df['preco'].rolling(window=7).mean()
```

pd.date_range()

Cria um intervalo de datas

Parâmetros principais:

```
  start: Data inicial
  end: Data final
  periods: Número de períodos
  freq: Frequência ('D', 'M', 'H', etc.)
```

```
datas = pd.date_range('2023-01-01', periods=365, freq='D')
```

Guia do Pandas

6. Visualização de Dados

df.plot()

Cria gráficos a partir do DataFrame

Parâmetros principais:

```
kind: Tipo de gráfico ('line', 'bar', 'hist', 'scatter', etc.)
x/y: Colunas para eixos
title: Título do gráfico
figsize: Tamanho da figura
```

```
df.plot(kind='bar', x='categoria', y='vendas', title='Vendas por Categoria')
```

df.hist()

Cria histogramas para colunas numéricas

Parâmetros principais:

```
column: Coluna específica (todas numéricas por padrão)
bins: Número de bins
grid: Se mostra grid
```

```
df.hist(column='idade', bins=20, grid=False)
```

df.boxplot()

Cria boxplots para colunas numéricas

Parâmetros principais:

```
column: Coluna(s) para plotar
by: Coluna para agrupamento
vert: Orientação vertical/horizontal
```

```
df.boxplot(column='salario', by='departamento')
```

df.corr()

Mostra matriz de correlação entre colunas numéricas

```
df.corr().style.background_gradient(cmap='coolwarm')
```