

## 1. Filtrando linhas com base em valores de colunas

python

```
import pandas as pd
```

```
data = {  
    'Date': ['April-10', 'April-11', 'April-12', 'April-13', 'April-14', 'April-16'],  
    'Sales': [200, 300, 400, 200, 300, 300],  
    'Price': [3, 1, 2, 4, 3, 2]  
}  
df = pd.DataFrame(data)
```

```
# Criando uma máscara booleana para Sales igual a 300
```

```
dfmask = df['Sales'] == 300
```

```
# Aplicando a máscara para filtrar o DataFrame
```

```
df_filtrado = df[dfmask]
```

```
print(df_filtrado)
```

## 2. Selecionando linhas e colunas específicas:

python

```
# Selecionar apenas as colunas 'Date' e 'Sales' onde Sales = 300
```

```
df_selecionado = df.loc[df['Sales'] == 300, ['Date', 'Sales']]
```

```
print(df_selecionado)
```

### 3. Seleção com múltiplas condições (exemplo com DataFrame de vendas):

python

```
# Selecionar produtos com 'preco' entre 50 e 100 e data maior que '2023-01-01'
```

```
df_vendas = pd.DataFrame({  
    'produto': ['Notebook', 'Celular', 'Camisa', 'Tenis'],  
    'preco': [2500, 800, 50, 120],  
    'data': ['2023-01-10', '2023-03-15', '2023-05-22', '2023-04-01']  
})
```

```
df_filtrado = df_vendas[(df_vendas['preco'] >= 50) & (df_vendas['preco'] <= 100) & (df_  
print(df_filtrado)
```

### 4. Selecionando linhas de acordo com múltiplos valores possíveis (exemplo cargos):

python

```
df_funcionarios = pd.DataFrame({  
    'nome': ['Carlos', 'Ana', 'Marcos', 'Fernanda'],  
    'cargo': ['Gerente', 'Atendente', 'Vendedor', 'Diretora'],  
    'salario': [5000, 2500, 3200, 7000]  
})
```

```
cargos_desejados = ['Gerente', 'Diretora']  
df_filtrado = df_funcionarios[df_funcionarios['cargo'].isin(cargos_desejados)]  
print(df_filtrado)
```

## 5. Combinação (merge) de DataFrames com chave comum

python

```
clientes = pd.DataFrame({
    'IDcliente': [1, 2, 3],
    'nome': ['Joao', 'Maria', 'Pedro'],
    'idade': [30, 25, 22]
})

pedidos = pd.DataFrame({
    'IDcliente': [1, 1, 2, 3],
    'produto': ['Camisa', 'Tenis', 'Celular', 'Notebook'],
    'valor': [50, 120, 800, 2500]
})

df_merge = clientes.merge(pedidos, on='IDcliente', how='inner')
print(df_merge)
```

## 6. Concatenação (concat) de DataFrames, horizontal e vertical

python

```
df1 = pd.DataFrame({'col1': [1, 2, 3], 'col2': ['A', 'B', 'C']})
df2 = pd.DataFrame({'col1': [3, 4, 5], 'col2': ['C', 'D', 'E']})

# Concatenação vertical (eixo 0) com reset de índice
df_concat_axis0 = pd.concat([df1, df2], ignore_index=True, axis=0)
print(df_concat_axis0)

# Concatenação horizontal (eixo 1)
df_concat_axis1 = pd.concat([df1, df2], axis=1)
print(df_concat_axis1)
```

## 7. Exemplo prático: juntando tabelas de vendas e análise

python

```
vendas = pd.DataFrame({
    'Data': pd.date_range(start='2024-05-01', periods=10, freq='D'),
    'QuantidadeVendida': [48, 38, 24, 17, 30, 48, 28, 32, 20, 20],
    'Sabor': ['Baunilha', 'Creme', 'Creme', 'Creme', 'Baunilha', 'Baunilha', 'Baunilha',
})

vendas_novas = pd.DataFrame({
    'Data': pd.date_range(start='2024-05-11', periods=10, freq='D'),
    'QuantidadeVendida': [33, 45, 49, 33, 12, 31, 11, 33, 39, 47],
    'Sabor': ['Baunilha', 'Morango', 'Baunilha', 'Creme', 'Baunilha', 'Creme', 'Creme',
})

# Concatenando as tabelas de vendas
vendas_completas = pd.concat([vendas, vendas_novas], ignore_index=True)

precos = pd.DataFrame({
    'Sabor': ['Baunilha', 'Chocolate', 'Morango', 'Creme'],
    'Preco': [5.0, 6.0, 5.5, 4.5]
})

# Merge entre vendas e preços
vendas_completas = vendas_completas.merge(precos, on='Sabor', how='left')

# Filtrando receitas acima de 200 (quantidade * preco)
vendas_completas['Receita'] = vendas_completas['QuantidadeVendida'] * vendas_completas['Preco']
filtro_receita = vendas_completas[vendas_completas['Receita'] > 200]
print(filtro_receita)
```