Durante a missão prática, realizei uma série de etapas essenciais de limpeza e manipulação de dados utilizando a biblioteca Pandas no Python. O objetivo principal foi preparar o conjunto de dados fornecido para análise e mineração, aplicando métodos de tratamento de valores nulos, transformação de tipos de dados e manipulação de colunas.

Iniciei com as microatividades, onde explorei operações básicas com Pandas:

1.Leitura e Visualização dos Dados: Na *Microatividade 1*, fiz a leitura do arquivo CSV e verifiquei a estrutura dos dados, imprimindo informações gerais e exibindo as primeiras e últimas linhas do conjunto. Por exemplo:

```
df = pd.read_csv('Online_Retail.csv', sep=';', encoding='utf-8', engine='python')
print(df.info())
```

2. Criação de Subconjuntos: Na *Microatividade 2*, aprendi a criar subconjuntos de dados. Atribuí parte das colunas a uma nova variável e exibi essas colunas específicas:

```
subconjunto = df[['CustomerID', 'InvoiceDate', 'Country']]
print(subconjunto)
```

3.Configuração de Exibição: Na *Microatividade 3*, ajustei a configuração de Pandas para aumentar o número máximo de linhas exibidas, utilizando o parâmetro `max_rows`:

```
pd.set_option('display.max_rows', 9999)
```

```
print(df.to_string())
```

4.Exibição de Linhas Específicas: Na *Microatividade 4*, imprimi as primeiras e últimas 10 linhas do conjunto de dados:

```
print(df.head(10))

print(df.tail(10))
```

5.Informações Gerais: Na *Microatividade 5*, utilizei o método `info()` para exibir as características das colunas, a quantidade de dados nulos, tipos de dados e o uso de memória:

```
print(df.info())
```

Após essa base, parti para a missão prática completa, onde:

• **Tratei valores nulos**: Substituí os valores nulos da coluna Country por 0 e os da coluna InvoiceDate por `'1900/01/01'`.

```
df_copy['Country'].fillna(0, inplace=True)

df_copy['InvoiceDate'].fillna('1900/01/01', inplace=True)
```

• **Transformei datas**: Corrigi formatos de data incorretos e converti a coluna InvoiceDate para o tipo `datetime`:

```
df_copy['InvoiceDate'] = pd.to_datetime(df_copy['InvoiceDate'], errors='coerce')
```

• **Removi registros inválidos**: Excluí os registros que ainda continham valores nulos após o tratamento:

```
df_cleaned = df_copy.dropna(subset=['InvoiceDate'])
```

Essas operações permitiram que eu preparasse os dados para análise, garantindo que o conjunto estivesse limpo e pronto para uso em tarefas de mineração de dados.