1. **Total de Vendas por Mês:  
Explicação:**

* **df.groupby('Mês')**: Esse comando agrupa o DataFrame df pela coluna **'Mês'**. Ou seja, ele organiza os dados em grupos com base no mês de cada venda.
* **['Receita Total'].sum()**: Após agrupar, pegamos a coluna **'Receita Total'** e calculamos a soma de todas as receitas para cada mês. Assim, a receita de cada mês é somada.
* **sort\_index()**: Organiza o resultado com base no índice (neste caso, no mês), garantindo que os meses apareçam na ordem correta de janeiro a dezembro.
* **sns.barplot()**: Esse comando cria um gráfico de barras, onde:
  + **x=vendas\_por\_mes.index**: No eixo X (horizontal), colocamos os meses, que estão no índice do vendas\_por\_mes.
  + **y=vendas\_por\_mes.values**: No eixo Y (vertical), colocamos as somas das receitas para cada mês, que são os valores de vendas\_por\_mes.
* **plt.title(), plt.xlabel(), plt.ylabel()**: Esses são comandos para adicionar o título do gráfico e as labels dos eixos X e Y.
* **plt.show()**: Exibe o gráfico gerado na tela.

2. **Produtos Mais Vendidos (em quantidade)**

**Explicação:**

* **df.groupby('Produto')**: Esse comando agrupa o DataFrame df pela coluna **'Produto'**. Ou seja, ele agrupa os dados de acordo com o nome de cada produto.
* **['Quantidade'].sum()**: Após agrupar, ele soma as quantidades de cada produto, ou seja, ele mostra quantas unidades de cada produto foram vendidas.
* **sort\_values(ascending=False)**: Ordena os produtos pela quantidade vendida, do maior para o menor (decrescente).
* **sns.barplot()**: Como antes, usamos o sns.barplot() para gerar o gráfico de barras. A diferença é que agora estamos plotando o **produto** no eixo X e a **quantidade vendida** no eixo Y.
* **plt.xticks(rotation=45)**: Esse comando rotaciona os nomes dos produtos no eixo X para que fiquem mais legíveis, pois, normalmente, os nomes dos produtos podem ser longos.

3. **Faturamento por Estado**

**Explicação:**

* **df.groupby('Estado')**: Agrupa os dados pela coluna **'Estado'**. Ou seja, ele organiza os dados com base no estado onde a venda foi realizada.
* **['Receita Total'].sum()**: Soma as receitas de cada estado, ou seja, calcula o total de faturamento gerado em cada estado.
* **sort\_values(ascending=False)**: Ordena os estados pelo faturamento de forma decrescente, para que os estados com maior faturamento apareçam primeiro.
* **sns.barplot()**: Utiliza o gráfico de barras novamente, agora com os **estados** no eixo X e o **faturamento total** no eixo Y.

4. **Distribuição das Formas de Pagamento**

**Explicação:**

* **df['Forma de Pagamento'].value\_counts()**: Esse comando conta a quantidade de ocorrências de cada valor na coluna **'Forma de Pagamento'**. Ou seja, ele mostra quantas vezes cada forma de pagamento foi utilizada nas vendas.
* **plt.pie()**: Cria um gráfico de pizza. O gráfico de pizza é útil para ver a distribuição proporcional dos dados.
  + **labels=formas\_pagamento.index**: Usamos os **nomes das formas de pagamento** como rótulos para cada fatia do gráfico.
  + **autopct='%1.1f%%'**: Exibe as porcentagens de cada fatia com uma casa decimal.
  + **startangle=90**: Gira o gráfico para que o primeiro segmento comece no topo.
* **plt.axis('equal')**: Isso garante que o gráfico de pizza tenha proporções iguais em todos os lados, para que ele seja circular.

5. **Vendas por Dia da Semana**

**Explicação:**

* **df.groupby('Dia da Semana')**: Agrupa os dados pela coluna **'Dia da Semana'**, ou seja, ele organiza as vendas por dias da semana (segunda-feira, terça-feira, etc.).
* **['Receita Total'].sum()**: Soma as receitas de cada dia da semana. Assim, conseguimos ver quanto de receita foi gerado em cada dia.
* **reindex(dias\_ordem)**: Esse comando reorganiza os dias da semana na ordem correta, começando pela **segunda-feira**. O dias\_ordem define essa sequência.
* **sns.barplot()**: Cria o gráfico de barras, colocando os **dias da semana** no eixo X e o **faturamento diário** no eixo Y.
* **plt.xticks(rotation=45)**: Esse comando rotaciona as labels no eixo X para melhorar a legibilidade dos nomes dos dias da semana.