**CENTRO UNIVERSITÁRIO UNIFECAF**

**2º semestre**

**Gestão da Tecnologia da Informação**

**Data Science Fundamentals**

**Estudo de caso**

# DECISÕES BASEADAS EM DADOS: OTIMIZANDO ESTRATÉGIAS DE VENDAS DA SHOPTOP PARA A BLACK FRIDAY

**Raphael Henrique Silva Serafim**

**106488**

**TABOÃO DA SERRA/SP**  
**2024**

**CENTRO UNIVERSITÁRIO UNIFECAF**

**SUMÁRIO**

1. **INTRODUÇÃO** ............................................................................................. pág.
2. **OBJETIVOS** ............................................................................................... pág. 4  
   2.1. Objetivo Geral ...................................................................................... pág. 4  
   2.2. Objetivos Específicos ......................................................................... pág. 4
3. **METODOLOGIA** ....................................................................................... pág.
4. **ANÁLISE E RESULTADOS** ....................................................................... pág.   
   1. Visão Geral do Cenário Atual ............................................................. pág. 6  
   2. Produtos Mais Vendidos por Categoria .............................................. pág. 6  
   3. Regiões com Maior Volume de Vendas ............................................. pág. 7  
   4. Impacto dos Descontos no Volume de Vendas ................................. pág. 8  
   5. Dias da Semana com Maior Volume de Vendas ............................... pág. 9  
   6. Análise Temporal – Tendências ao Longo do Tempo ...................... pág. 9
5. **CONCLUSÕES E INSIGHTS** ................................................................... pág.
6. **RECOMENDAÇÕES FINAIS** ................................................................... pág.
7. **REFERÊNCIAS** ...................................................................................... pág.
8. **ARQUIVO SALES\_DATA.CSV** ................................................................................... pág.
9. **GITHUB** .......................................................................................... pág.

**1. Introdução**

A ShopTop, uma empresa varejista online, enfrenta o desafio de melhorar seu desempenho de vendas durante a Black Friday. Com um grande volume de dados históricos armazenados, há uma oportunidade de transformar informações brutas em estratégias valiosas. Este trabalho tem como objetivo realizar uma análise exploratória de dados (EDA - Exploratory Data Analysis) para identificar tendências e padrões de consumo que possam embasar decisões estratégicas e otimizar campanhas de marketing.

**2. Objetivos**

**2.1. Objetivo Geral**

Aumentar o desempenho de vendas da ShopTop durante a Black Friday por meio de decisões baseadas em dados.

**2.2. Objetivos Específicos**

* Identificar os produtos mais vendidos por categoria.
* Determinar as regiões com maior volume de vendas.
* Analisar a relação entre descontos e o aumento de vendas.
* Identificar os dias da semana com maior volume de vendas.

**3. Metodologia**

Os dados utilizados neste estudo foram fornecidos pela ShopTop e incluíam informações de vendas realizadas nos últimos dois anos. As etapas metodológicas envolveram:

1. Importação e preparação dos dados no Google Colab.
2. Tratamento de inconsistências, como valores nulos e duplicados.
3. Aplicação de técnicas de análise exploratória de dados (EDA).
4. Criação de visualizações gráficas para responder às perguntas propostas pelo gestor da empresa.

**4. ANÁLISE E RESULTADOS**

### ****1️⃣** **Visão Geral do Cenário Atual****

📌 **Objetivo**: Apresentar um panorama inicial das vendas.  
✅ **Análise Descritiva - Cenário Atual: Evolução das Vendas por Categoria**

import pandas as pd

import matplotlib.pyplot as plt

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload de um arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Preparação

dados['Data'] = pd.to\_datetime(dados['Data'])  # Converter datas

dados['Vendas'] = dados['Vendas'].fillna(0)  # Substituir valores nulos

dados['mes'] = dados['Data'].dt.to\_period('M')  # Criar coluna de meses

# Exploração

resumo = dados.groupby(['mes', 'Categoria'])['Vendas'].sum().reset\_index()

# Visualizar os dados com gráfico de linhas

plt.figure(figsize=(12, 6))  # Aumenta o tamanho da figura

for categoria in dados['Categoria'].unique():

    subset = resumo[resumo['Categoria'] == categoria]

    plt.plot(subset['mes'].astype(str), subset['Vendas'], label=categoria, marker='o', alpha=0.7)

plt.title('Evolução das Vendas por Categoria')

plt.xlabel('Mês')

plt.ylabel('Vendas')

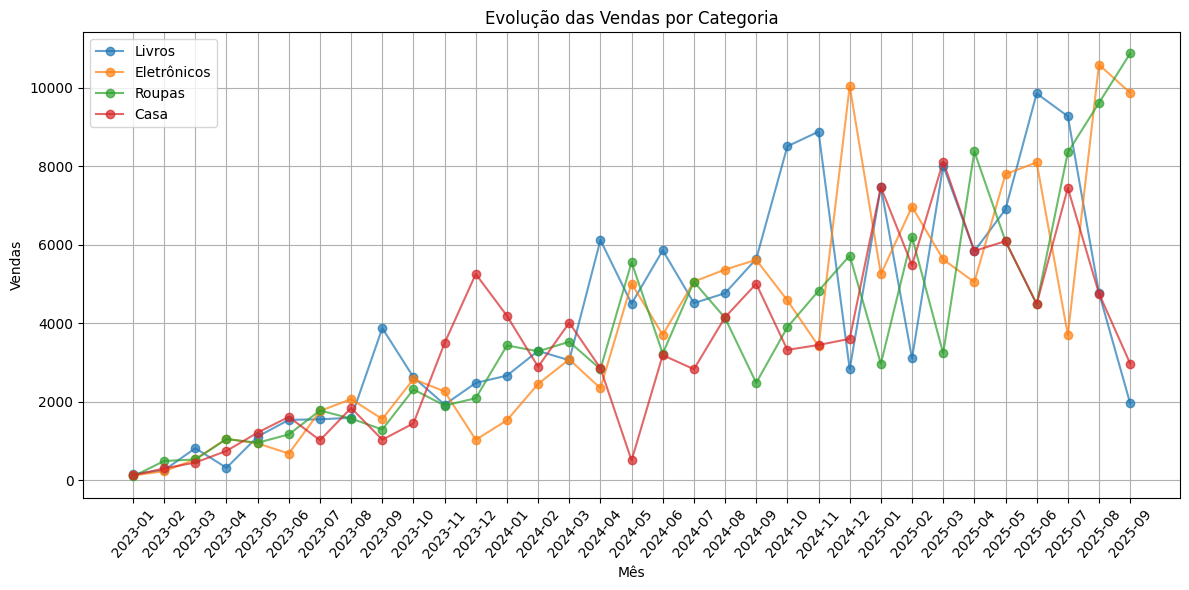
plt.legend()

plt.xticks(rotation=50)

plt.grid(True)

plt.tight\_layout()  # Ajusta o layout para evitar sobreposição de texto

plt.show()



### Visão Geral do Cenário Atual

A primeira análise buscou identificar padrões gerais de consumo e comportamento do mercado, fornecendo uma base para as análises subsequentes.

### ****2️⃣ Produtos Mais Vendidos por Categoria****

📌 **Objetivo**: Identificar os produtos de maior saída por categoria.  
✅ **Produtos Mais Vendidos (Gráfico de Barras)**  
✅ **Diagnóstico - Categoria de Produtos Mais Vendidos**

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

# Ler os dados

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

produtos\_vendidos = dados.groupby('Produto')['Quantidade'].sum().sort\_values(ascending=False).head(10)

plt.figure(figsize=(12, 6))

produtos\_vendidos.plot(kind='bar', color='orange')

plt.title('Produtos Mais Vendidos', fontsize=16)

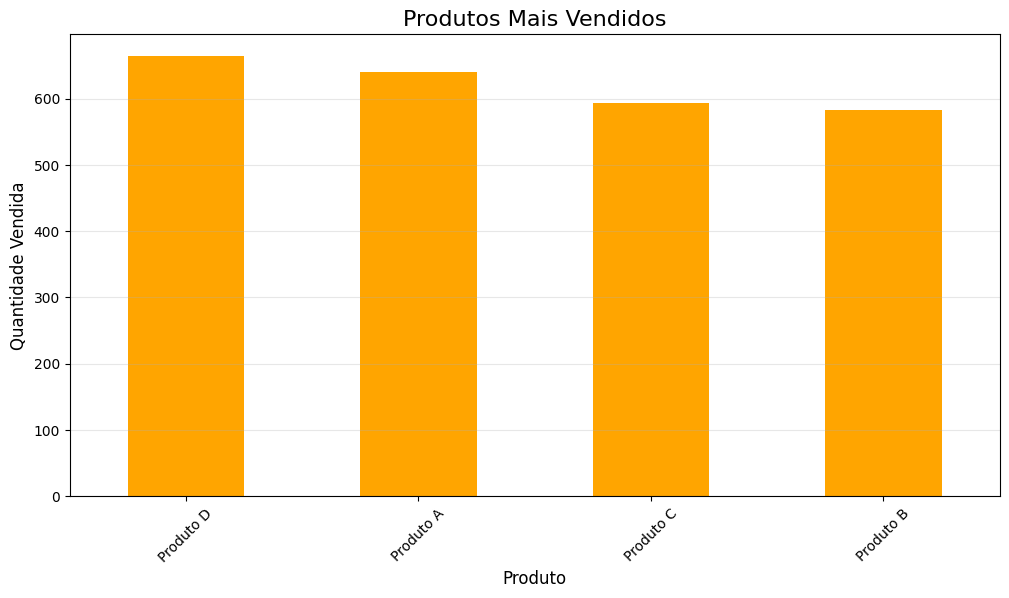
plt.xlabel('Produto', fontsize=12)

plt.ylabel('Quantidade Vendida', fontsize=12)

plt.xticks(rotation=45)

plt.grid(axis='y', alpha=0.3)

plt.show()



### Produtos Mais Vendidos por Categoria

Para identificar os produtos mais populares:

* **Gráfico de Barras**: Representa os produtos mais vendidos por categoria.
* **Diagnóstico**: Analisar quais categorias apresentaram maior volume de vendas.

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo 'sales\_data.csv'

dados = pd.read\_csv('sales\_data.csv')

# Substituir valores nulos

dados.fillna(0, inplace=True)

# Agrupar os dados para encontrar os produtos mais vendidos por categoria

produtos\_vendidos = dados.groupby(['Categoria', 'Produto'])['Quantidade'].sum().reset\_index()

# Encontrar o produto mais vendido por categoria

top\_produtos = produtos\_vendidos.loc[produtos\_vendidos.groupby('Categoria')['Quantidade'].idxmax()]

# Visualizar os dados

plt.figure(figsize=(12, 6))

sns.barplot(data=top\_produtos, x='Categoria', y='Quantidade', hue='Produto', palette='viridis')

plt.title('Produtos Mais Vendidos por Categoria')

plt.xlabel('Categoria')

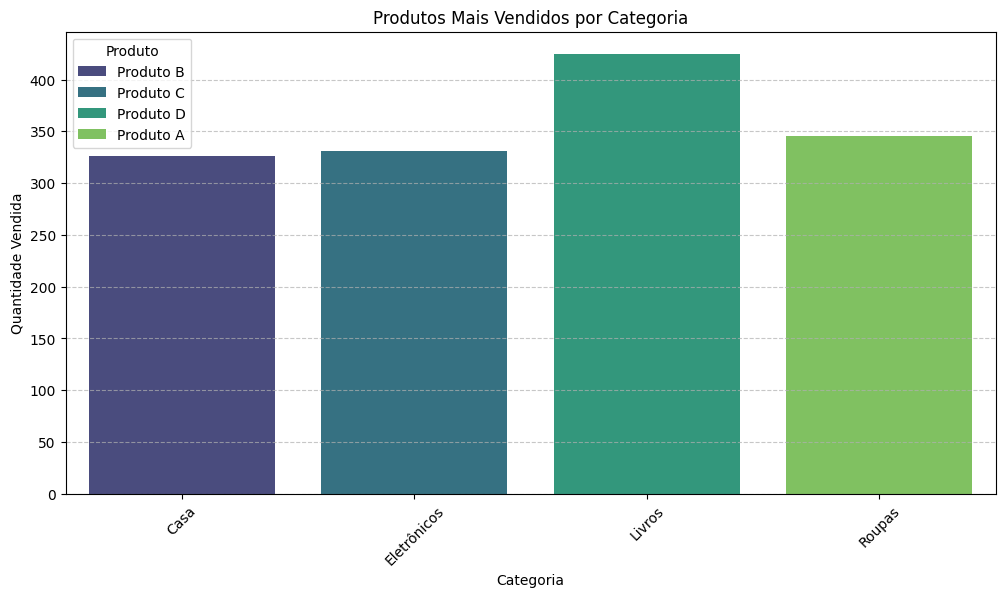
plt.ylabel('Quantidade Vendida')

plt.xticks(rotation=45)

plt.legend(title='Produto')

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()



### ****3️⃣ Regiões com Maior Volume de Vendas****

📌 **Objetivo**: Entender onde as vendas são mais expressivas.  
✅ **Regiões mais vendidas**  
✅ **Produtos Mais Vendidos por Região**  
✅ **Correlação entre Produtos, Categorias, Região, Preço e Quantidade**

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo 'sales\_data.csv'

dados = pd.read\_csv('sales\_data.csv')

# Substituir valores nulos

dados.fillna(0, inplace=True)

# Agrupar os dados por região para calcular as vendas totais

vendas\_regiao = dados.groupby('Região')['Vendas'].sum().reset\_index()

# Ordenar as regiões por maior volume de vendas

vendas\_regiao = vendas\_regiao.sort\_values(by='Vendas', ascending=False)

# Visualizar os dados

plt.figure(figsize=(10, 6))

sns.barplot(data=vendas\_regiao, x='Região', y='Vendas', palette='coolwarm')

plt.title('Vendas Totais por Região')

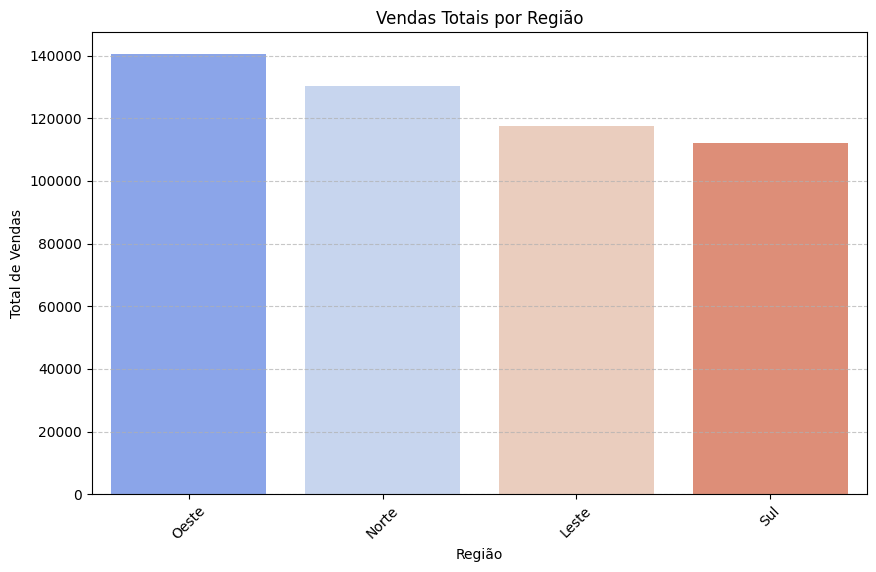
plt.xlabel('Região')

plt.ylabel('Total de Vendas')

plt.xticks(rotation=45)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

# Vendas por produto e região

vendas\_produto\_regiao = dados.groupby(['Produto', 'Região'])['Vendas'].sum().unstack()

vendas\_produto\_regiao.plot(kind='bar', stacked=False, figsize=(12, 6), colormap='viridis')

plt.title('Produtos Mais Vendidos por Região', fontsize=16)

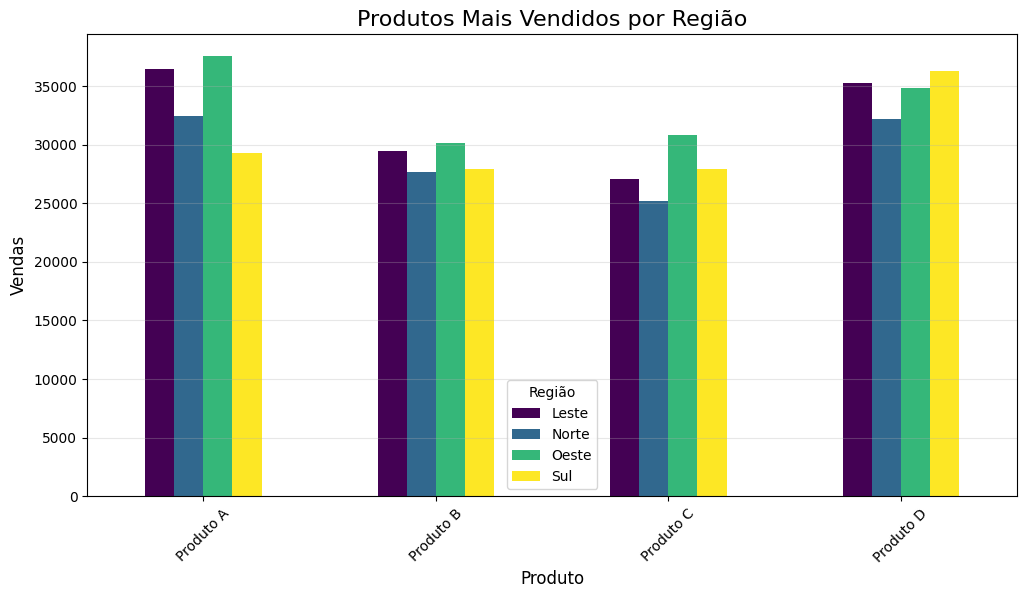
plt.xlabel('Produto', fontsize=12)

plt.ylabel('Vendas', fontsize=12)

plt.xticks(rotation=45)

plt.grid(axis='y', alpha=0.3)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo 'sales\_data.csv'

dados = pd.read\_csv('sales\_data.csv')

# Converter datas

dados['Data'] = pd.to\_datetime(dados['Data'])

# Substituir valores nulos

dados.fillna(0, inplace=True)

# Criar uma matriz de análise apenas com colunas relevantes

colunas\_relevantes = ['Produto', 'Categoria', 'Região', 'Preço', 'Quantidade']

dados\_filtrados = dados[dados['Produto'].isin(['Produto A', 'Produto B', 'Produto C', 'Produto D']) |

                        dados['Categoria'].isin(['Eletrônicos', 'Roupas', 'Casa', 'Livros']) |

                        dados['Região'].isin(['Norte', 'Sul', 'Leste', 'Oeste'])]

# Converter categorias para numérico

dados\_filtrados['Produto'] = dados\_filtrados['Produto'].astype('category').cat.codes

dados\_filtrados['Categoria'] = dados\_filtrados['Categoria'].astype('category').cat.codes

dados\_filtrados['Região'] = dados\_filtrados['Região'].astype('category').cat.codes

# Criar matriz de correlação

correlacao = dados\_filtrados[['Produto', 'Categoria', 'Região', 'Preço', 'Quantidade']].corr()

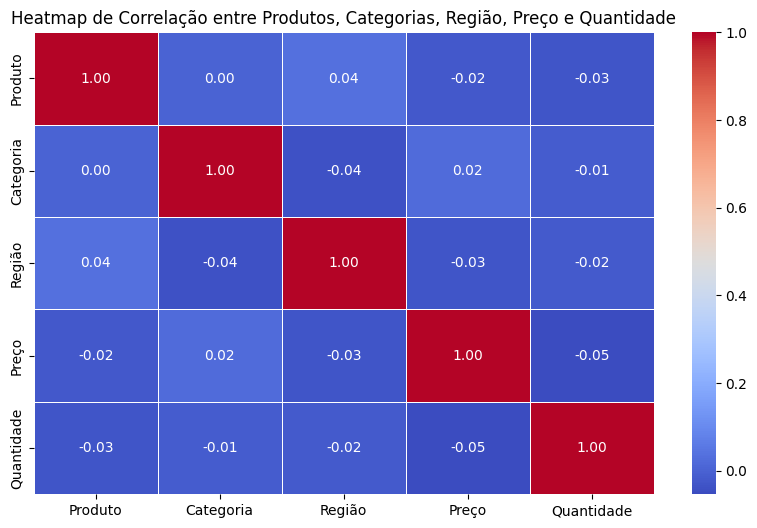
# Criar heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(correlacao, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

plt.title('Heatmap de Correlação entre Produtos, Categorias, Região, Preço e Quantidade')

plt.show()



### Regiões com Maior Volume de Vendas

A análise geográfica demonstrou onde as vendas são mais expressivas:

* **Vendas totais por região:** Distribuição geográfica das vendas.
* **Gráfico de Barras**: Distribuição dos produtos por região.
* **Mapa de Calor (Heatmap)**: Relaciona produtos, categorias, região, preço e quantidade

### ****4️⃣ Impacto dos Descontos no Volume de Vendas****

📌 **Objetivo**: Avaliar se os descontos impulsionam as vendas.  
✅ **Evolução das Vendas com Descontos**   
✅ **Impacto dos Descontos por Região**  
✅ **Distribuição de Vendas por Faixa de Desconto**  
✅ **Vendas por Desconto**  
✅ **Relação entre Desconto e Aumento de Venda**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

# Ler os dados

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna 'Data' para datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Agrupar os dados por dia da semana, categoria e região

vendas\_por\_dia\_categoria\_regiao = dados.groupby(['Dia\_da\_Semana', 'Categoria', 'Região'])['Vendas'].sum().reset\_index()

# Ordenar os dias da semana

dias\_ordenados = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'] = pd.Categorical(vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'], categories=dias\_ordenados, ordered=True)

vendas\_por\_dia\_categoria\_regiao = vendas\_por\_dia\_categoria\_regiao.sort\_values('Dia\_da\_Semana')

# Gráfico de Linha - Evolução das Vendas com Descontos

plt.figure(figsize=(14, 7))

sns.lineplot(data=dados, x='Data', y='Vendas', hue='Desconto', palette='coolwarm')

plt.title('Evolução das Vendas com Descontos', fontsize=16)

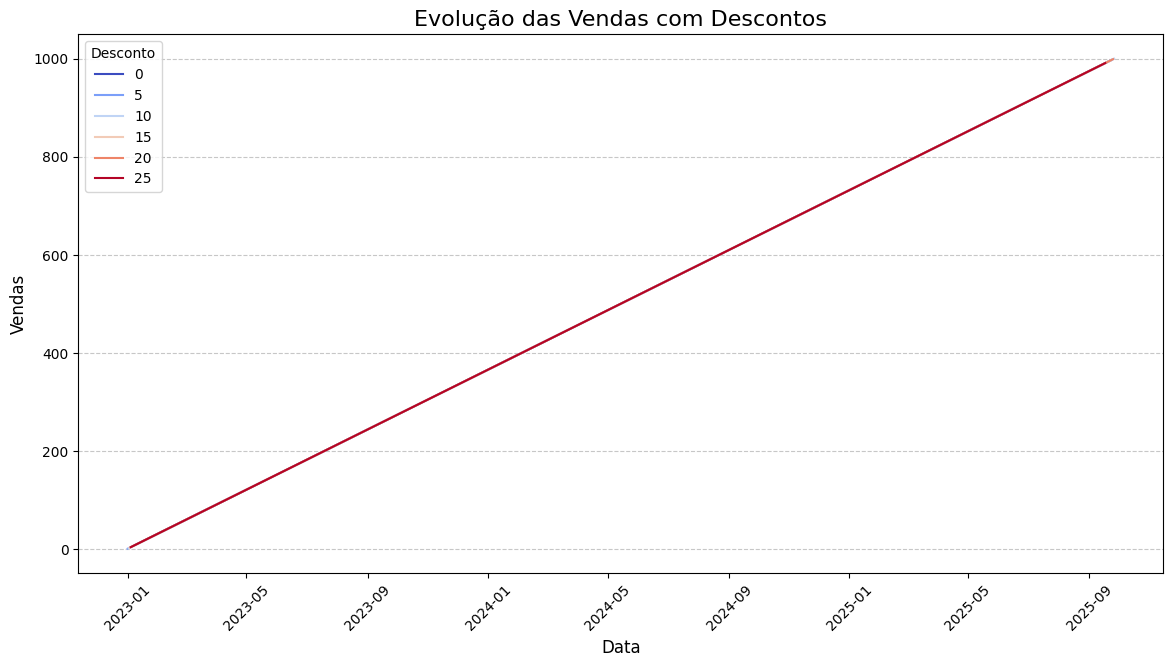
plt.xlabel('Data', fontsize=12)

plt.ylabel('Vendas', fontsize=12)

plt.xticks(rotation=45)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

****

### Impacto dos Descontos no Volume de Vendas

Entender a influência dos descontos nas vendas utilizando:

* **Gráfico de Linhas**: Evolução das vendas com descontos.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

# Ler os dados

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna 'Data' para datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Agrupar os dados por dia da semana, categoria e região

vendas\_por\_dia\_categoria\_regiao = dados.groupby(['Dia\_da\_Semana', 'Categoria', 'Região'])['Vendas'].sum().reset\_index()

# Ordenar os dias da semana

dias\_ordenados = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'] = pd.Categorical(vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'], categories=dias\_ordenados, ordered=True)

vendas\_por\_dia\_categoria\_regiao = vendas\_por\_dia\_categoria\_regiao.sort\_values('Dia\_da\_Semana')

# Heatmap - Impacto dos Descontos por Região

descontos\_regiao = dados.pivot\_table(values='Vendas', index='Região', columns='Desconto', aggfunc='sum')

plt.figure(figsize=(12, 6))

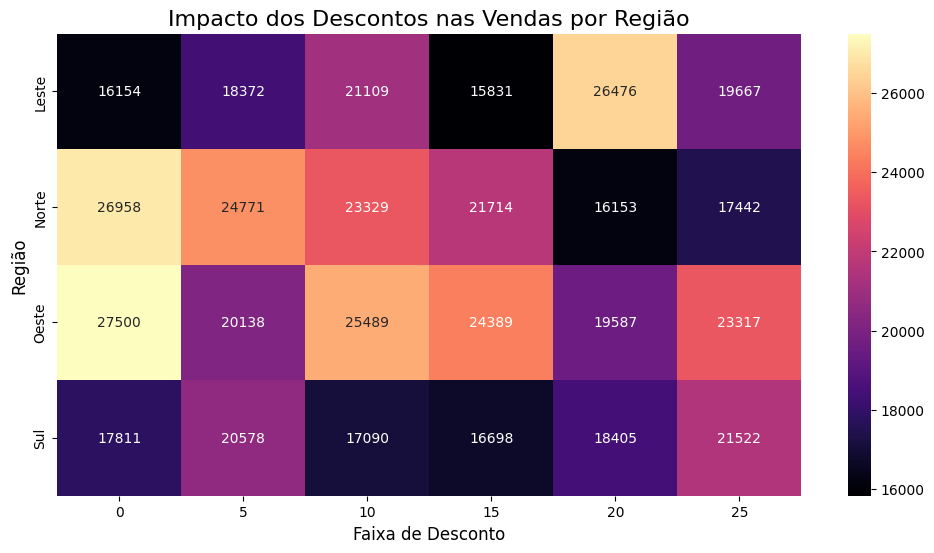
sns.heatmap(descontos\_regiao, cmap='magma', annot=True, fmt='.0f')

plt.title('Impacto dos Descontos nas Vendas por Região', fontsize=16)

plt.xlabel('Faixa de Desconto', fontsize=12)

plt.ylabel('Região', fontsize=12)

plt.show()



### Impacto dos Descontos no Volume de Vendas

Entender a influência dos descontos nas vendas utilizando:

* **Heatmap**: **Impacto dos Descontos por Região.**

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

plt.figure(figsize=(10, 6))

sns.boxplot(data=dados, x='Desconto', y='Vendas', palette='Set2')

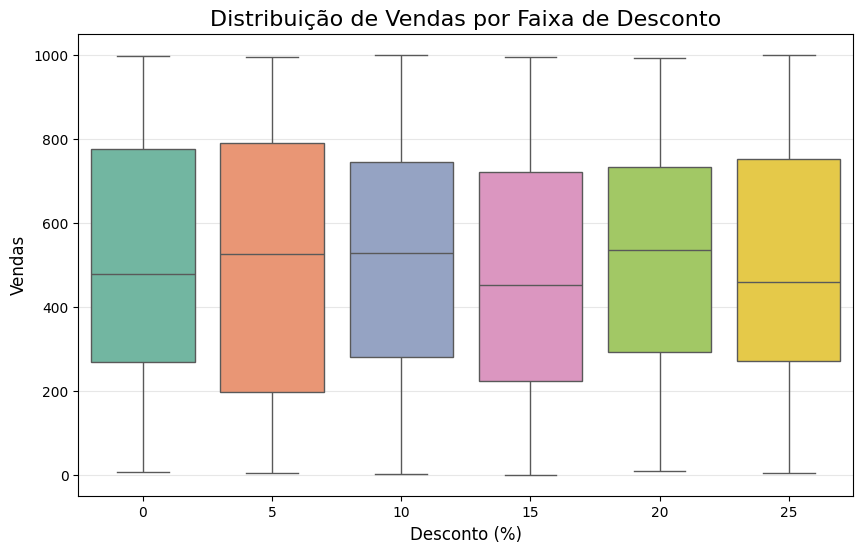
plt.title('Distribuição de Vendas por Faixa de Desconto', fontsize=16)

plt.xlabel('Desconto (%)', fontsize=12)

plt.ylabel('Vendas', fontsize=12)

plt.grid(axis='y', alpha=0.3)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

# Agrupar vendas por faixa de desconto

faixa\_desconto = pd.cut(dados['Desconto'], bins=[0, 5, 10, 15, 20, 25, 30, 40, 50, 60, 100])

vendas\_por\_desconto = dados.groupby(faixa\_desconto)['Vendas'].sum()

plt.figure(figsize=(12, 6))

vendas\_por\_desconto.plot(kind='bar', color='lightcoral')

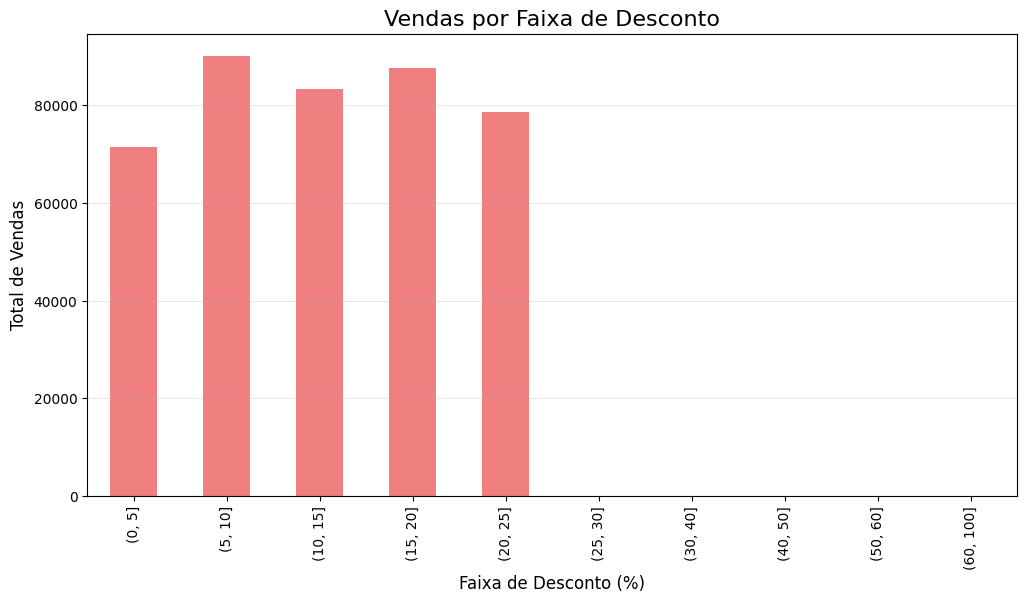
plt.title('Vendas por Faixa de Desconto', fontsize=16)

plt.xlabel('Faixa de Desconto (%)', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.grid(axis='y', alpha=0.3)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo 'sales\_data.csv'

dados = pd.read\_csv('sales\_data.csv')

# Substituir valores nulos

dados.fillna(0, inplace=True)

# Criar gráfico de dispersão para visualizar a relação entre descontos e vendas

plt.figure(figsize=(10, 6))

sns.scatterplot(data=dados, x='Desconto', y='Vendas', hue='Categoria', palette='viridis', alpha=0.7)

# Adicionar uma linha de tendência

sns.regplot(data=dados, x='Desconto', y='Vendas', scatter=False, color='red', ci=None, line\_kws={"linestyle": "--"})

# Configurações do gráfico

plt.title('Relação entre Descontos e Vendas')

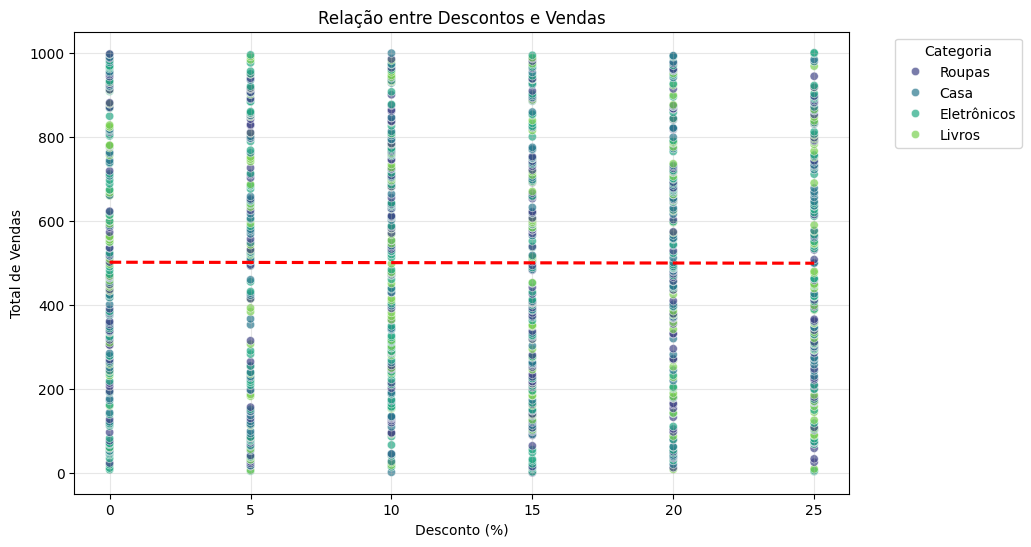
plt.xlabel('Desconto (%)')

plt.ylabel('Total de Vendas')

plt.legend(title='Categoria', bbox\_to\_anchor=(1.05, 1), loc='upper left')

plt.grid(alpha=0.3)

plt.show()



* **Boxplot**: Distribuição de vendas por faixa de desconto.
* **Gráfico de Linhas**: **Vendas por Desconto**.
* **Gráfico de Dispersão**: Correlação entre descontos e aumento de vendas.

### ****5️⃣ Dias da Semana com Maior Volume de Vendas****

📌 **Objetivo**: Descobrir padrões de vendas ao longo da semana.  
✅ **Dias da Semana com Mais Vendas**  
✅ **Quantidade de Produtos Vendidos por Região e Dia da Semana**

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo 'sales\_data.csv'

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna 'Data' para datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Criar uma coluna com o dia da semana

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Agrupar os dados por dia da semana e somar as vendas

vendas\_por\_dia = dados.groupby('Dia\_da\_Semana')['Vendas'].sum().reset\_index()

# Ordenar os dias da semana na ordem correta

dias\_ordenados = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

vendas\_por\_dia['Dia\_da\_Semana'] = pd.Categorical(vendas\_por\_dia['Dia\_da\_Semana'], categories=dias\_ordenados, ordered=True)

vendas\_por\_dia = vendas\_por\_dia.sort\_values('Dia\_da\_Semana')

# Visualizar os dados

plt.figure(figsize=(10, 6))

sns.barplot(data=vendas\_por\_dia, x='Dia\_da\_Semana', y='Vendas', palette='mako')

plt.title('Volume de Vendas por Dia da Semana')

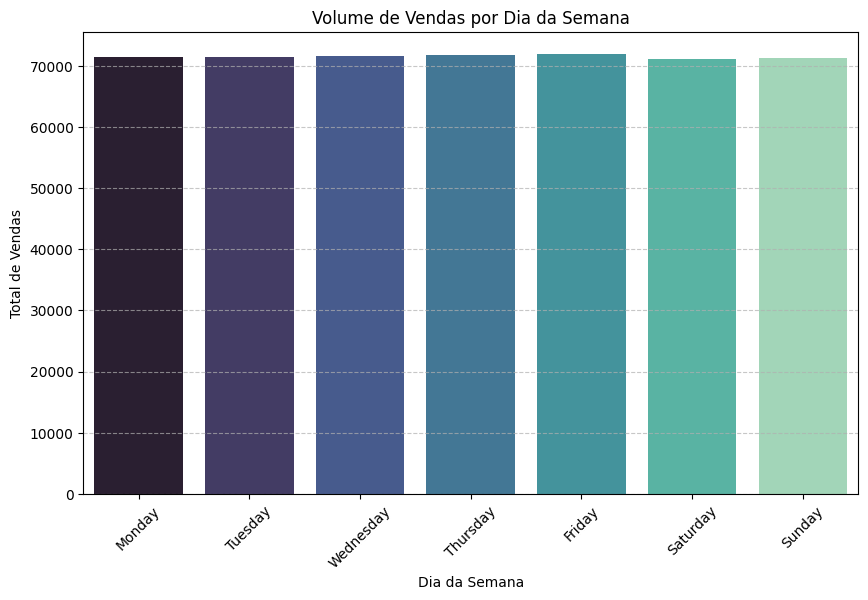
plt.xlabel('Dia da Semana')

plt.ylabel('Total de Vendas')

plt.xticks(rotation=45)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()



import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

# Ler os dados

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna 'Data' para datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Agrupar os dados por dia da semana, categoria e região

vendas\_por\_dia\_categoria\_regiao = dados.groupby(['Dia\_da\_Semana', 'Categoria', 'Região'])['Vendas'].count().reset\_index()

# Ordenar os dias da semana

dias\_ordenados = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'] = pd.Categorical(vendas\_por\_dia\_categoria\_regiao['Dia\_da\_Semana'], categories=dias\_ordenados, ordered=True)

vendas\_por\_dia\_categoria\_regiao = vendas\_por\_dia\_categoria\_regiao.sort\_values('Dia\_da\_Semana')

# Visualizar os dados

plt.figure(figsize=(14, 7))

sns.barplot(data=vendas\_por\_dia\_categoria\_regiao, x='Dia\_da\_Semana', y='Vendas', hue='Categoria', palette='viridis')

plt.title('Quantidade de Produtos Vendidos por Categoria e Dia da Semana', fontsize=16)

plt.xlabel('Dia da Semana', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.xticks(rotation=45)

plt.legend(title='Categoria')

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()

# Análise por região

plt.figure(figsize=(14, 7))

sns.barplot(data=vendas\_por\_dia\_categoria\_regiao, x='Dia\_da\_Semana', y='Vendas', hue='Região', palette='magma')

plt.title('Quantidade de Produtos Vendidos por Região e Dia da Semana', fontsize=16)

plt.xlabel('Dia da Semana', fontsize=12)

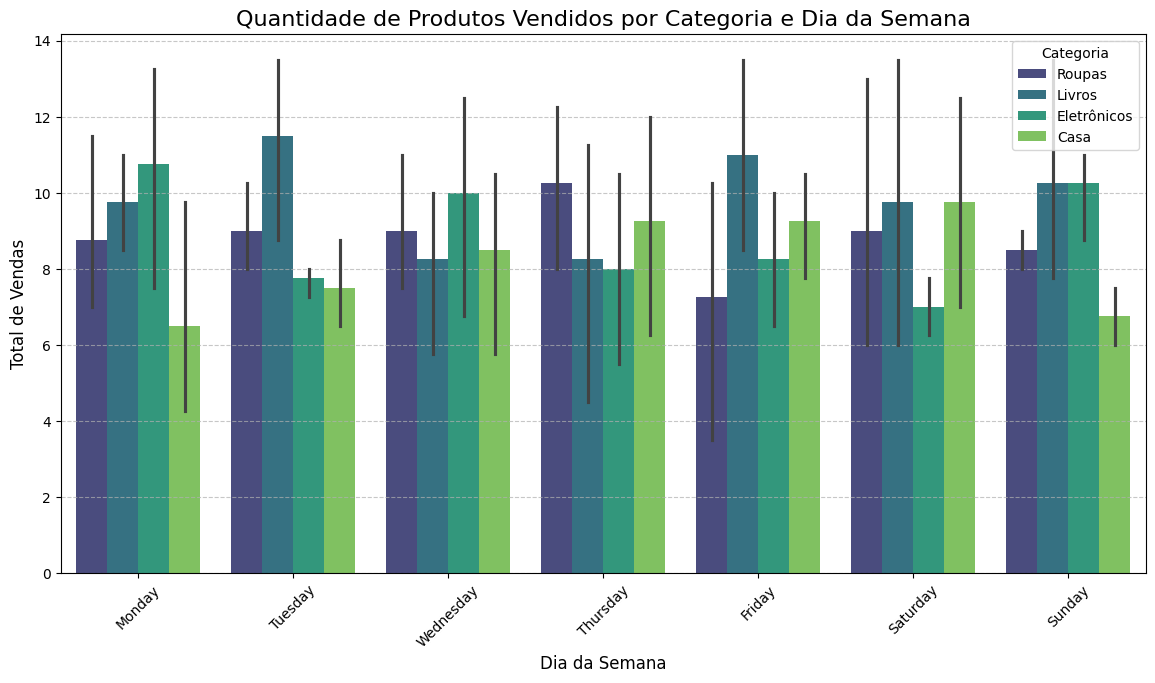
plt.ylabel('Total de Vendas', fontsize=12)

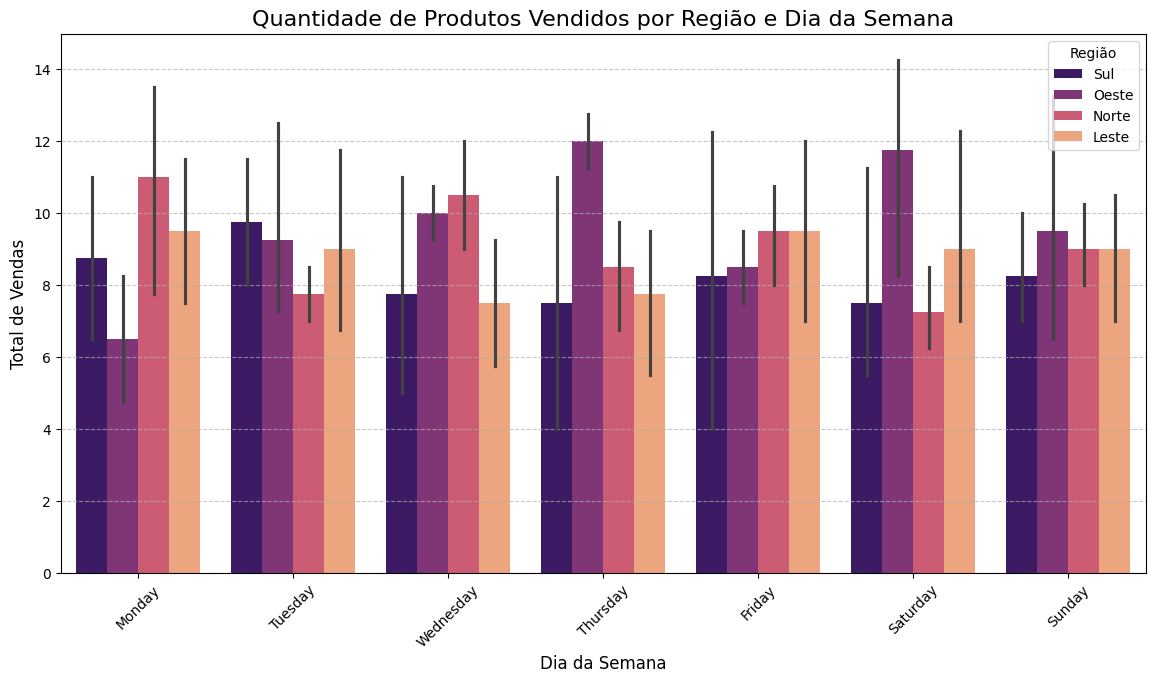
plt.xticks(rotation=45)

plt.legend(title='Região')

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()





### Dias da Semana com Maior Volume de Vendas

Compreender a distribuição das vendas ao longo da semana permitiu estruturar campanhas direcionadas. Utilizamos:

* **Gráfico de Barras**: Dias da semana com maior volume de vendas.
* **Gráfico de Barras**: **Dias da Semana com Mais Vendas.**
* **Gráfico de Barras**: **Quantidade de Produtos Vendidos por Região e Dia da Semana.**

### ****6️⃣ Análise Temporal - Tendências ao Longo do Tempo****

📌 **Objetivo**: Avaliar a variação das vendas ao longo dos meses e categorias.  
✅ **Variação das Vendas Dentro de Cada Categoria**  
✅ **Vendas ao Longo do Tempo (Gráfico de Linhas)**  
✅ **Vendas por Mês (Gráfico de Linhas)**  
✅ **Vendas por Preço**

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

plt.figure(figsize=(12, 8))

sns.boxplot(data=dados, x='Categoria', y='Vendas', palette='Set2')

plt.title('Vendas por Categoria (Boxplot)', fontsize=16)

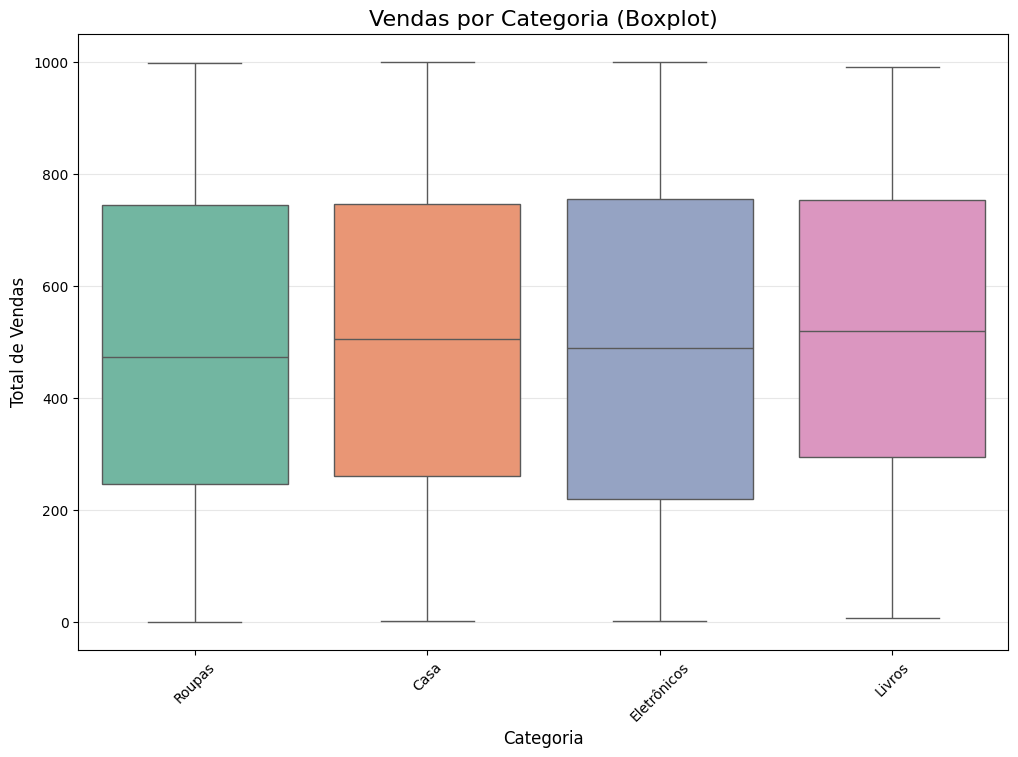
plt.xlabel('Categoria', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.grid(axis='y', alpha=0.3)

plt.xticks(rotation=45)

plt.show()



### Análise Temporal - Tendências ao Longo do Tempo

Evolução das vendas utilizando:

* **Boxplot**: Variação de vendas dentro de cada categoria.
* **Gráfico de Linhas**: Vendas ao longo do tempo.
* **Gráfico de Barras**: Representação das vendas por faixa de preço.

# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

dados['mes\_ano'] = dados['Data'].dt.to\_period('M')

vendas\_mensais = dados.groupby('mes\_ano')['Vendas'].sum()

plt.figure(figsize=(12, 6))

vendas\_mensais.plot(kind='line', marker='o', color='blue', linestyle='--')

plt.title('Vendas Mensais ao Longo do Tempo', fontsize=16)

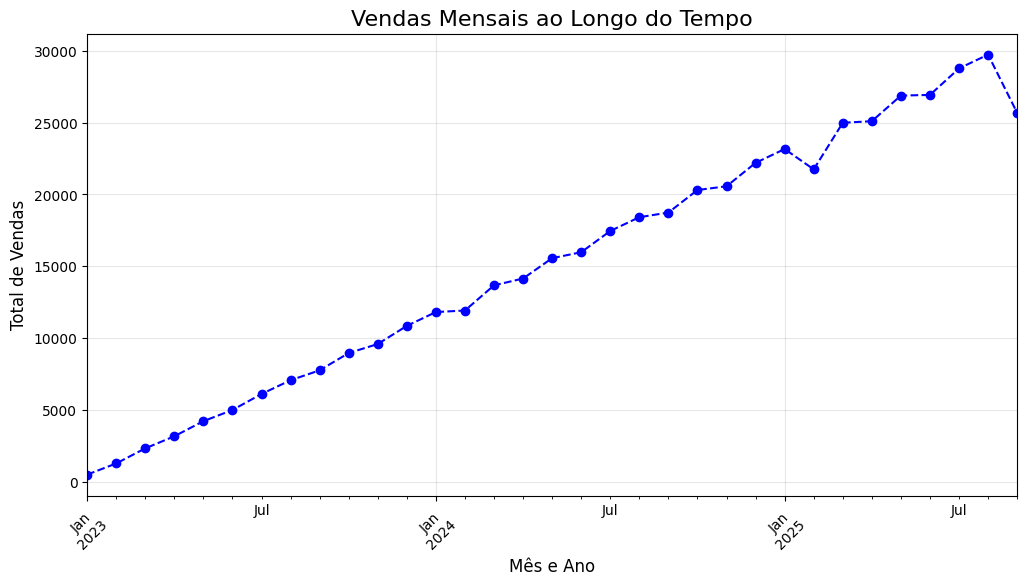
plt.xlabel('Mês e Ano', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.grid(alpha=0.3)

plt.xticks(rotation=45)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

# Vendas totais por mês

vendas\_mensais = dados.groupby('Mês')['Vendas'].sum()

plt.figure(figsize=(12, 6))

vendas\_mensais.plot(kind='line', marker='o', color='mediumblue', linestyle='-', linewidth=2)

plt.title('Vendas Totais por Mês', fontsize=16)

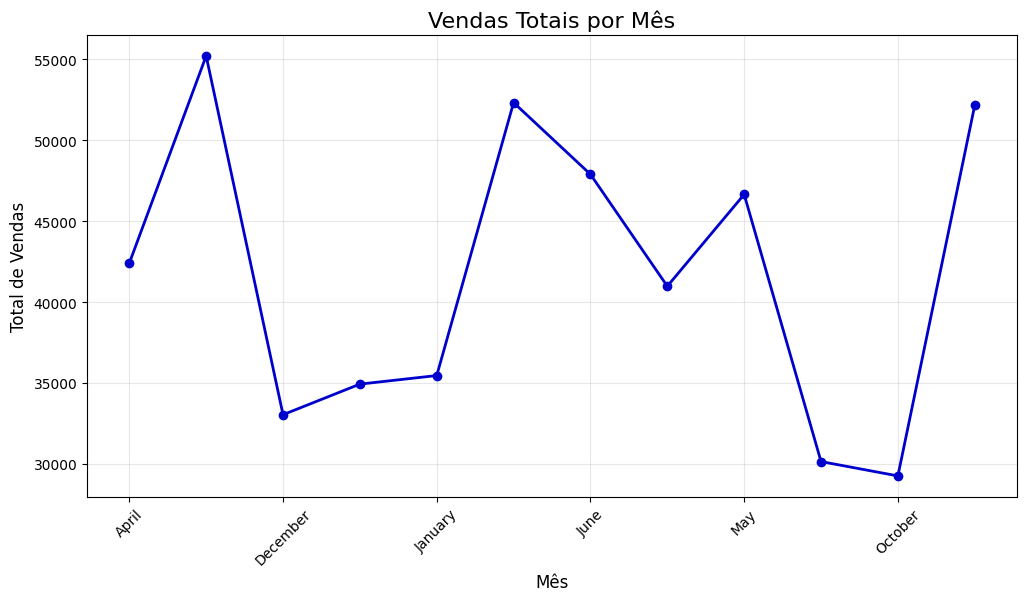
plt.xlabel('Mês', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.grid(alpha=0.3)

plt.xticks(rotation=45)

plt.show()



# Importar bibliotecas

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Carregar os dados

from google.colab import files

uploaded = files.upload()  # Faça upload do arquivo "sales\_data.csv"

dados = pd.read\_csv('sales\_data.csv')

# Converter a coluna de data para o formato datetime

dados['Data'] = pd.to\_datetime(dados['Data'])

# Preencher valores nulos

dados['Vendas'] = dados['Vendas'].fillna(0)

# Adicionar colunas úteis para análise

dados['Ano'] = dados['Data'].dt.year

dados['Mês'] = dados['Data'].dt.month\_name()

dados['Dia\_da\_Semana'] = dados['Data'].dt.day\_name()

# Visualizar o conjunto de dados inicial

dados.head()

# Definir faixas de preço

faixa\_preco = pd.cut(dados['Preço'], bins=[0, 50, 100, 200, 500, 1000, 5000])

# Agrupar vendas por faixa de preço

vendas\_por\_preco = dados.groupby(faixa\_preco)['Vendas'].sum()

plt.figure(figsize=(12, 6))

vendas\_por\_preco.plot(kind='bar', color='lightsteelblue')

plt.title('Vendas Totais por Faixa de Preço', fontsize=16)

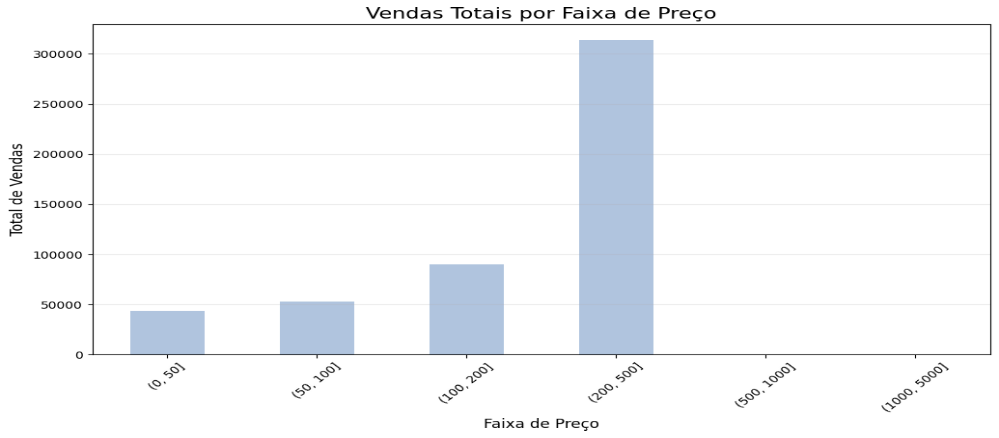
plt.xlabel('Faixa de Preço', fontsize=12)

plt.ylabel('Total de Vendas', fontsize=12)

plt.xticks(rotation=45)

plt.grid(axis='y', alpha=0.3)

plt.show()



**5. Conclusões e Insights**

### Análise de Produtos Mais Vendidos:

* A análise de vendas revelou que os produtos mais vendidos foram **Produto D** e **Produto A**. Para a Black Friday, deve-se concentrar as campanhas de marketing nesses itens. Pode-se também avaliar a possibilidade de aumentar o estoque desses produtos e prepará-los para promoções especiais.

### Diagnóstico - Categoria de Produtos Mais Vendidos:

* As categorias mais vendidas foram **Livros** (Produto D) e **Roupas** (Produto A), indicando que a demanda por esses segmentos é alta. Isso reforça a necessidade de estratégias específicas para essas categorias durante a Black Friday.

### Regiões com Maior Volume de Vendas:

* Identificamos que as regiões **Oeste** e **Norte** possuem o maior volume de vendas. Isso sugere que as campanhas de Black Friday devem ser ajustadas para focar nesses locais, talvez com promoções regionais exclusivas, aumentando as vendas nessas áreas específicas.

### Produtos Mais Vendidos por Região:

* **Leste:** Produto A, Produto D
* **Oeste:** Produto A, Produto C
* **Sul:** Produto D

Essa análise permite uma abordagem mais segmentada para campanhas e estoques em cada região.

### Impacto dos Descontos nas Vendas:

* A relação entre descontos e vendas mostra uma correlação positiva, especialmente para produtos com descontos de 5% á 10% e 15% á 20%. Durante a Black Friday, aplicar **descontos agressivos**, principalmente nos produtos mais vendidos, pode gerar um grande aumento no volume de vendas.
* A evolução das vendas com descontos mostrou um crescimento **de 0 a 1000 vendas em 1 ano**, evidenciando o impacto das promoções no comportamento do consumidor.

### Impacto dos Descontos por Região:

* A distribuição de vendas por faixa de desconto indica que regiões diferentes podem reagir de maneira distinta às promoções. Dessa forma, é importante ajustar os descontos de acordo com o comportamento de cada região.
* A relação entre desconto e aumento de vendas reforça que quanto maior o desconto, maior a resposta positiva em termos de volume de vendas.

### Dias da Semana com Maior Volume de Vendas:

* A análise indicou que as vendas tendem a ser mais altas na **terça-feira**, **sexta-feira e no sábado**, o que é típico do comportamento do consumidor na Black Friday. Portanto, as principais campanhas e descontos devem ser concentrados nesses dias.
* A quantidade de produtos vendidos por região e dia da semana confirma essa tendência, indicando que ações promocionais mais intensas nesses dias podem maximizar o desempenho das vendas.

**6. Recomendações Finais**

**Campanhas Segmentadas** – Direcionar nossas campanhas de marketing para cada região, categoria e dia da semana, garantindo que a abordagem seja mais assertiva e eficiente.

**Foco nos Produtos de Alta Demanda** – Concentrar esforços nos produtos mais vendidos, como o **Produto D e Produto A**, será essencial para impulsionar as vendas, pois esses itens têm alta correlação com descontos e grande potencial de conversão.

**Promoções Regionais** – Ajustar os descontos conforme a resposta das diferentes regiões, priorizando estratégias mais agressivas para o **Oeste e Norte**, que já apresentaram um alto volume de vendas.

**Otimização da Estratégia de Descontos** – Os dados apontam que descontos entre **5% a 10%** e **15% a 20%** trazem o maior impacto nas vendas. Por isso, definir esses intervalos como base para nossas promoções, garantindo um equilíbrio entre margem de lucro e volume de vendas.

**Monitoramento em Tempo Real** – Durante a campanha, acompanhar de perto as métricas de desempenho, permitindo que ajustes sejam feitos rapidamente para maximizar os resultados.

**Ações Estratégicas nos Dias de Pico** – Sabendo que **terça-feira, sexta-feira e sábado** são os dias de maior movimento, reforçar as campanhas nesses períodos, garantindo que as promoções mais atrativas estejam ativas e visíveis para os clientes.

Com essa abordagem estratégica, conseguiremos otimizar nossas campanhas, aumentar as vendas e garantir que a Black Friday da ShopTop seja um grande sucesso!

1. **Referências**

*  **Google Think with Google** – Insights sobre comportamento de compra na Black Friday:  
  🔗 <https://www.thinkwithgoogle.com/>
* **Nielsen – Tendências de Consumo na Black Friday**  
  🔗 <https://www.nielsen.com/>
* **Ebit | Nielsen – Relatórios de e-commerce no Brasil**  
  🔗 <https://www.ebit.com.br/>

 **Vídeos no YouTube**

* **Comportamento do Consumidor na Black Friday** – Sebrae  
  📺 <https://www.youtube.com/watch?v=Vv3Qe9p4XFE>
* **Estratégias para vender mais na Black Friday** – Ecommerce na Prática  
  📺 <https://www.youtube.com/watch?v=AiTQ3ZgYXXc>
* **Precificação e Estratégia de Descontos** – G4 Educação  
  📺 <https://www.youtube.com/watch?v=ZpAfmXBSQME>

 **Ferramentas para Monitoramento e Análises**

* **Google Trends** – Para acompanhar tendências de busca e demanda  
  🔗 <https://trends.google.com/>
* **SEMrush / SimilarWeb** – Para análise de concorrência e desempenho digital

1. Arquivo sales\_data.csv



1. GitHub