

Introdução à análise de Dados com Python

Atividade Prática – Visualização de Dados em Python

1. Conexão com o Banco de Dados:

- A gente cria um banco de dados usando SQLite, que é um banco de dados simples.
- Dentro desse banco, criamos uma tabela chamada "vendas1" para guardar as vendas. Cada venda tem uma data, produto, categoria e o valor da venda.
- Depois disso, colocamos alguns dados de vendas fictícios (vendas de diferentes produtos ao longo do ano). Carregando e Organizando os Dados.
- Com a biblioteca Pandas, carregamos esses dados do banco para um formato chamado DataFrame, que é tipo uma tabela organizada.
- Convertemos a coluna da data para um formato que o Python entende como data de verdade.
- Exploramos os dados, vendo estatísticas como a soma dos valores e organizando-os por categorias e meses.

2. Analisando os Dados:

- Agrupamos as vendas por categoria (por exemplo, quantas vendas tiveram na categoria "Eletrônicos", "Roupas", etc.).
- Calculamos as vendas de cada mês, para ver como as vendas se comportaram ao longo do ano.
- Descobrimos quais foram os 3 produtos que mais venderam no ano.

3. Criando Gráficos:

- Com o Matplotlib e o Seaborn, criamos gráficos para visualizar melhor as vendas:
 - Um gráfico de barras que mostra quanto foi vendido em cada categoria.
 - Um gráfico de linha para ver as vendas mês a mês.
 - Um gráfico de barras mostrando os produtos mais vendidos.
 - E um gráfico de distribuição (boxplot) para entender como as vendas estão distribuídas por categoria.

```

import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Passo 1.1: Conectar ao banco de dados (ou criar, se não existir)
conexao = sqlite3.connect('dados_vendas.db')

# Passo 1.2: Criar um cursor
cursor = conexao.cursor()

# Passo 1.3: Criar uma tabela (se não existir)
cursor.execute('''
CREATE TABLE IF NOT EXISTS vendas1 (
    id_venda INTEGER PRIMARY KEY AUTOINCREMENT,
    data_venda DATE,
    produto TEXT,
    categoria TEXT,
    valor_venda REAL
) ''')

# Passo 1.4: Inserir alguns dados (apenas insira uma vez, ou descomente se rodar novamente)
cursor.execute('''
INSERT INTO vendas1 (data_venda, produto, categoria, valor_venda)
VALUES
('2023-01-01', 'Produto A', 'Eletrônicos', 1500.00),
('2023-01-05', 'Produto B', 'Roupas', 350.00),
('2023-02-10', 'Produto C', 'Eletrônicos', 1200.00),
('2023-03-15', 'Produto D', 'Livros', 200.00),
('2023-03-20', 'Produto E', 'Eletrônicos', 800.00),
('2023-04-02', 'Produto F', 'Roupas', 400.00),
('2023-05-05', 'Produto G', 'Livros', 150.00),
('2023-06-10', 'Produto H', 'Eletrônicos', 1000.00),
('2023-07-20', 'Produto I', 'Roupas', 600.00),
('2023-08-25', 'Produto J', 'Eletrônicos', 700.00),
('2023-09-30', 'Produto K', 'Livros', 300.00),
('2023-10-05', 'Produto L', 'Roupas', 450.00),
('2023-11-15', 'Produto M', 'Eletrônicos', 900.00),
('2023-12-20', 'Produto N', 'Livros', 250.00);
''')

# Passo 1.5: Confirmar as mudanças
conexao.commit()

# Carregar os dados do SQLite para um DataFrame
df_vendas = pd.read_sql_query("SELECT * FROM vendas1", conexao)

# Exibir os primeiros registros para verificar a estrutura dos dados
print(df_vendas.head())

# Converter a coluna de data para o tipo datetime
df_vendas['data_venda'] = pd.to_datetime(df_vendas['data_venda'])

# Exibir resumo estatístico dos dados
print(df_vendas.describe())

```

```

# Agrupar por categoria e somar as vendas
vendas_por_categoria = df_vendas.groupby('categoria')['valor_venda'].sum()
print("Vendas por categoria:")
print(vendas_por_categoria)

# Vendas totais por mês
df_vendas['mes'] = df_vendas['data_venda'].dt.month
vendas_por_mes = df_vendas.groupby('mes')['valor_venda'].sum()
print("\nVendas por mês:")
print(vendas_por_mes)

# Top 3 produtos mais vendidos
top_produtos = df_vendas.groupby('produto')['valor_venda'].sum().sort_values(ascending=False).head(3)
print("\nTop 3 produtos mais vendidos:")
print(top_produtos)

# Gráfico de barras - Vendas por Categoria
plt.figure(figsize=(8,6))
vendas_por_categoria.plot(kind='bar', color='skyblue')
plt.title('Vendas por Categoria')
plt.xlabel('Categoria')
plt.ylabel('Valor das Vendas')
plt.show()

# Gráfico de linha - Vendas por Mês
plt.figure(figsize=(8,6))
vendas_por_mes.plot(kind='line', marker='o', color='orange')
plt.title('Vendas por Mês')
plt.xlabel('Mês')
plt.ylabel('Valor das Vendas')
plt.show()

# Gráfico de barras - Top 3 Produtos mais Vendidos
plt.figure(figsize=(8,6))
top_produtos.plot(kind='bar', color='green')
plt.title('Top 3 Produtos mais Vendidos')
plt.xlabel('Produto')
plt.ylabel('Valor das Vendas')
plt.show()

# Gráfico Seaborn - Distribuição de Vendas por Categoria
plt.figure(figsize=(8,6))
sns.boxplot(x='categoria', y='valor_venda', data=df_vendas)
plt.title('Distribuição de Vendas por Categoria')
plt.show()

```

```

id_venda  data_venda  produto  categoria  valor_venda
0         1  2023-01-01  Produto A  Eletrônicos    1500.0
1         2  2023-01-05  Produto B   Roupas         350.0
2         3  2023-02-10  Produto C  Eletrônicos    1200.0
3         4  2023-03-15  Produto D   Livros         200.0
4         5  2023-03-20  Produto E  Eletrônicos     800.0

count  id_venda  data_venda  valor_venda
mean    7.5000  2023-06-09 00:00:00    628.571429
min     1.0000  2023-01-01 00:00:00    150.000000
25%     4.2500  2023-03-16 06:00:00    312.500000
50%     7.5000  2023-05-23 00:00:00    525.000000
75%    10.7500  2023-09-21 00:00:00    875.000000
max    14.0000  2023-12-20 00:00:00   1500.000000
std     4.1833                NaN    407.484915

Vendas por categoria:
categoria
Eletrônicos    6100.0
Livros         900.0
Roupas         1800.0
Name: valor_venda, dtype: float64

Vendas por mês:
mes
1     1850.0
2     1200.0
3     1000.0
4      400.0
5      150.0
6     1000.0
7      600.0
8      700.0
9      300.0
10     450.0
11     900.0
12     250.0
Name: valor_venda, dtype: float64

Top 3 produtos mais vendidos:
produto
Produto A    1500.0
Produto C    1200.0
Produto H    1000.0
Name: valor_venda, dtype: float64

```

Unidade 2 – Gustavo Nascimento Cazzine

