CENTRO UNIVERSITÁRIO ANHANGUERA PITÁGORAS UNOPAR DE CAMPO GRANDE

CURSO: ENGENHARIA DE SOFTWARE

DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO

ATIVIDADE PRÁTICA – UNIDADE 3, AULA 4

VISUALIZAÇÃO DE DADOS EM PYTHON

Aluno: Edmar Radanovis.

RA: 2025223493

Polo: Itapira / SP-UN944038

_____

Ano 2025 / 2º Semestre

ANHANGUERA EDUCACIONAL

ENGENHARIA DE SOFTWARE

DISCIPLINA: Linguagem de Programação

PROFESSOR: Anderson I. S. Abreu / Vanessa Matias Leite

ALUNO: Edmar Radanovis

RA: 2025223493

TÍTULO: Relatório da Aula Prática – Visualização de Dados em Python – Unidade 3 – Aula 4

POLO: Itapira / SP-UN944038

Monte Sião, 09 de Setembro de 2025.

RESULTADOS DA ATIVIDADE PRÁTICA

Proposta:

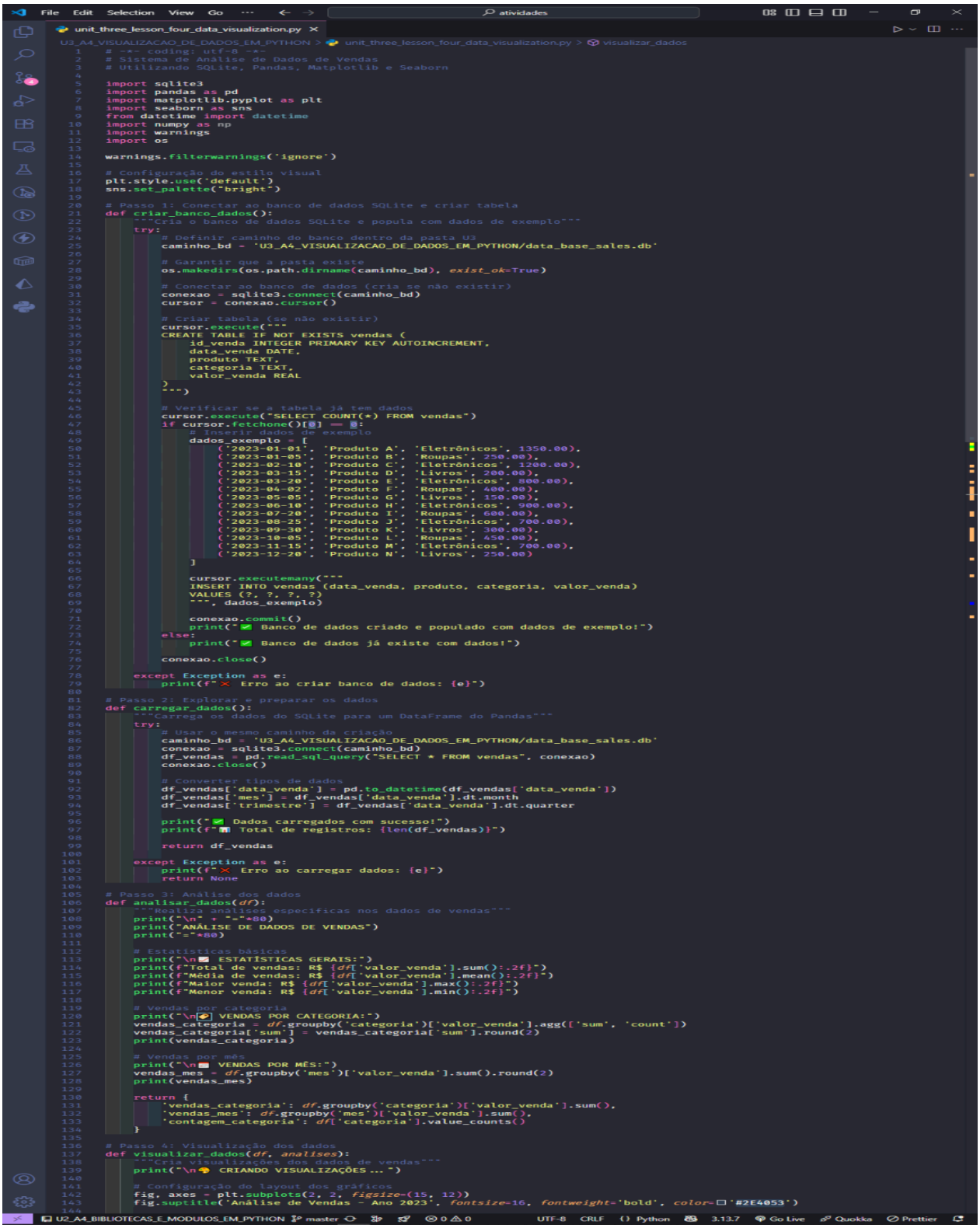
Introdução à Análise de Dados com Python - Visualização de Dados.

- Você trabalha em uma empresa de varejo e precisa analisar os dados de vendas do último ano para identificar padrões e insights para melhorar o desempenho. Os dados estão armazenados em um banco de dados SQLite, e você utilizará a biblioteca Pandas para manipular e analisar esses dados, além de gerar visualizações utilizando Matplotlib e Seaborn.

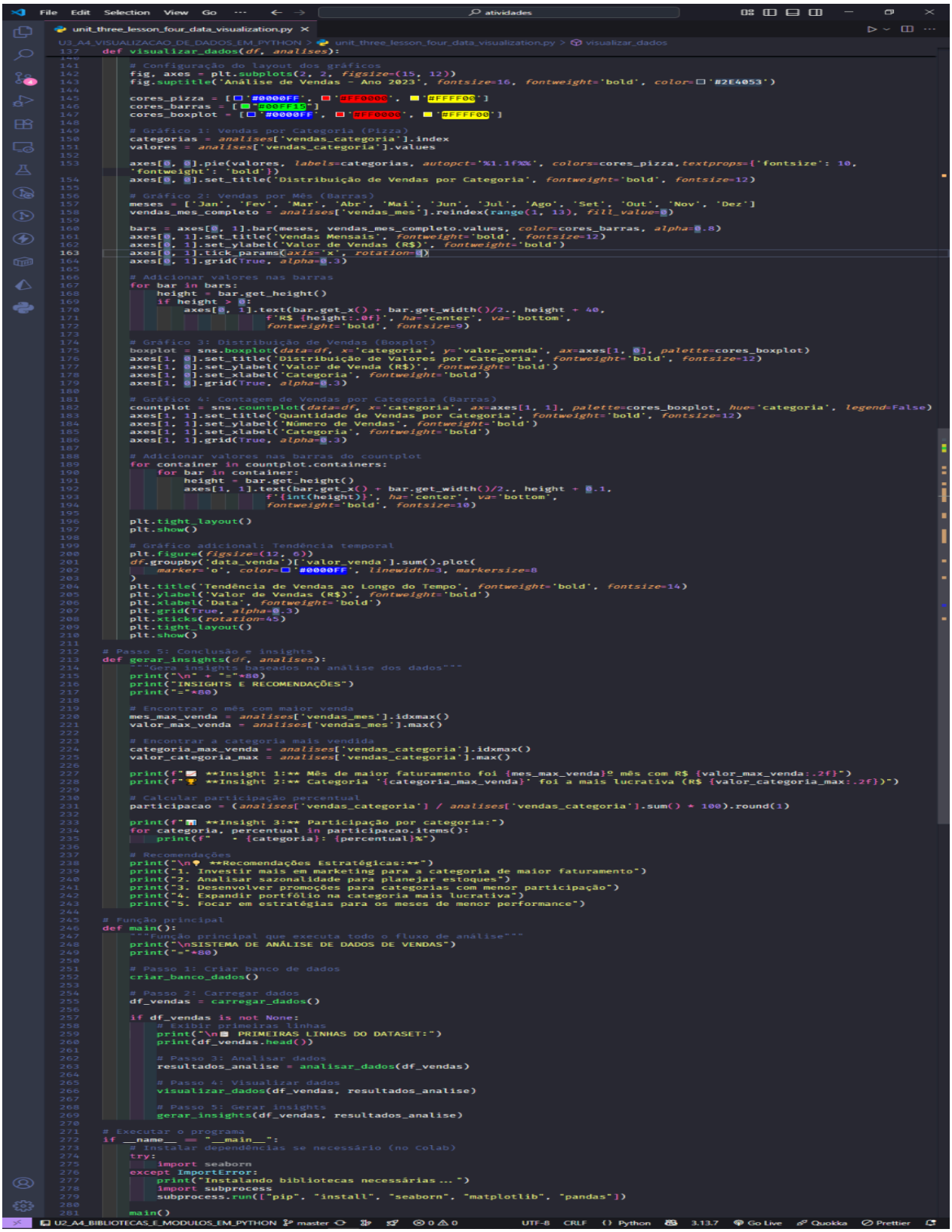
    - **Conectar ao banco de dados SQLite e criar uma tabela:**
        - Estabelecer uma conexão com o banco de dados SQLite e carregar os dados relevantes para análise.
        - Criar e conectar ao banco de dados.
        - Criar um cursor.
        - Criar uma tabela.
        - Inserir dados.
        - Confirmar as mudanças.
    - **Explorar e preparar os dados:**
        - Com os dados devidamente carregados em um DataFrame do Pandas, explorá-los e prepará-los para análise.

- **Analisar os dados :**
    - Analisar específicamente para extrair insights.
- **Visualizar os dados :**
    - Utilizar Matplotlib e Seaborn para criar visualizações que ajudem a interpretar os resultados.
- **Concluir e analisar os insights dos dados :**
    - Finalizar o exercício com uma breve análise dos insights obtidos e sugestões para a empresa com base nos dados analisados.

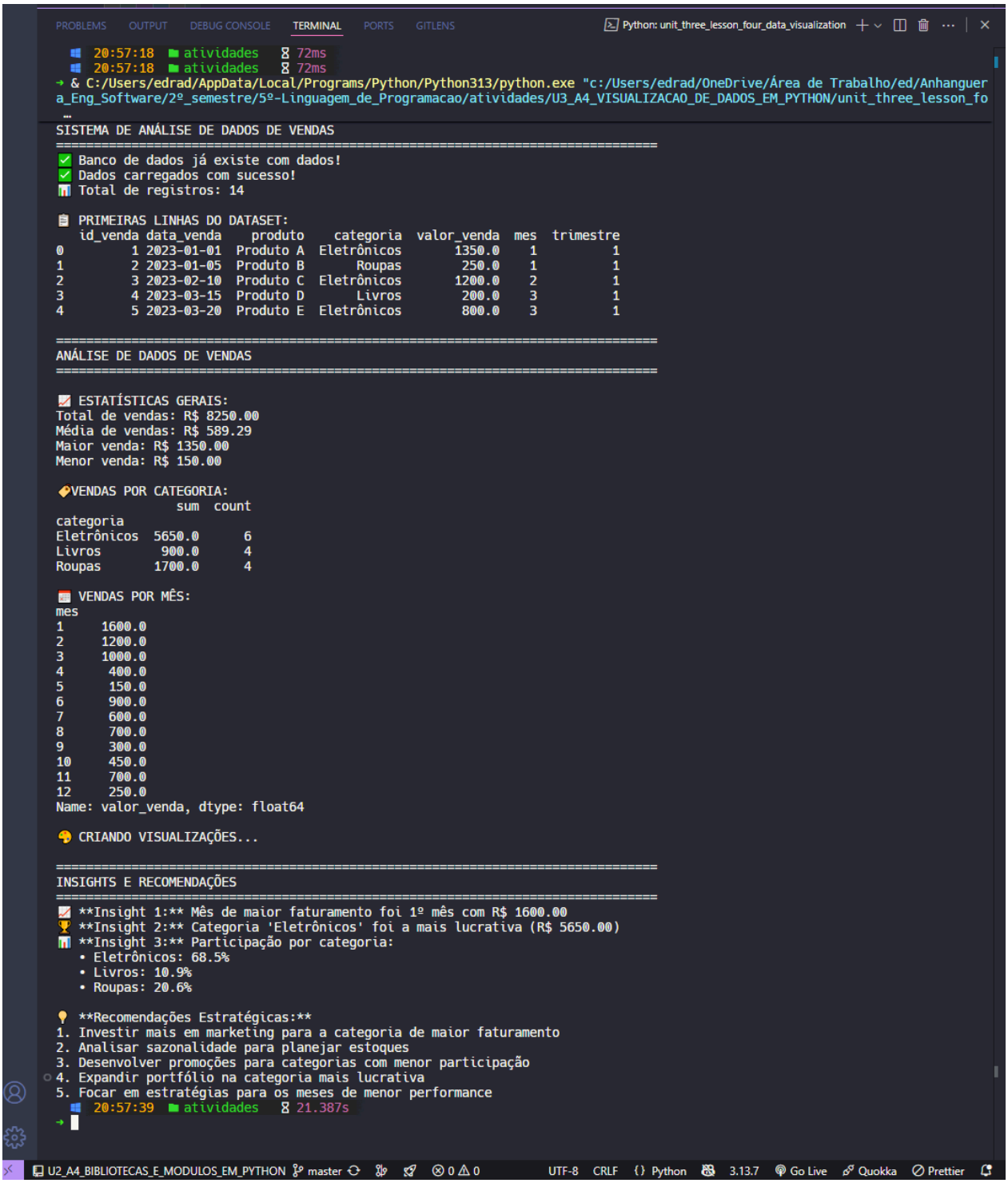
link do repositório no GitHub:
https://github.com/ed-radanovis/Eng_Software_L-P_U3-A4_09-2025-.git



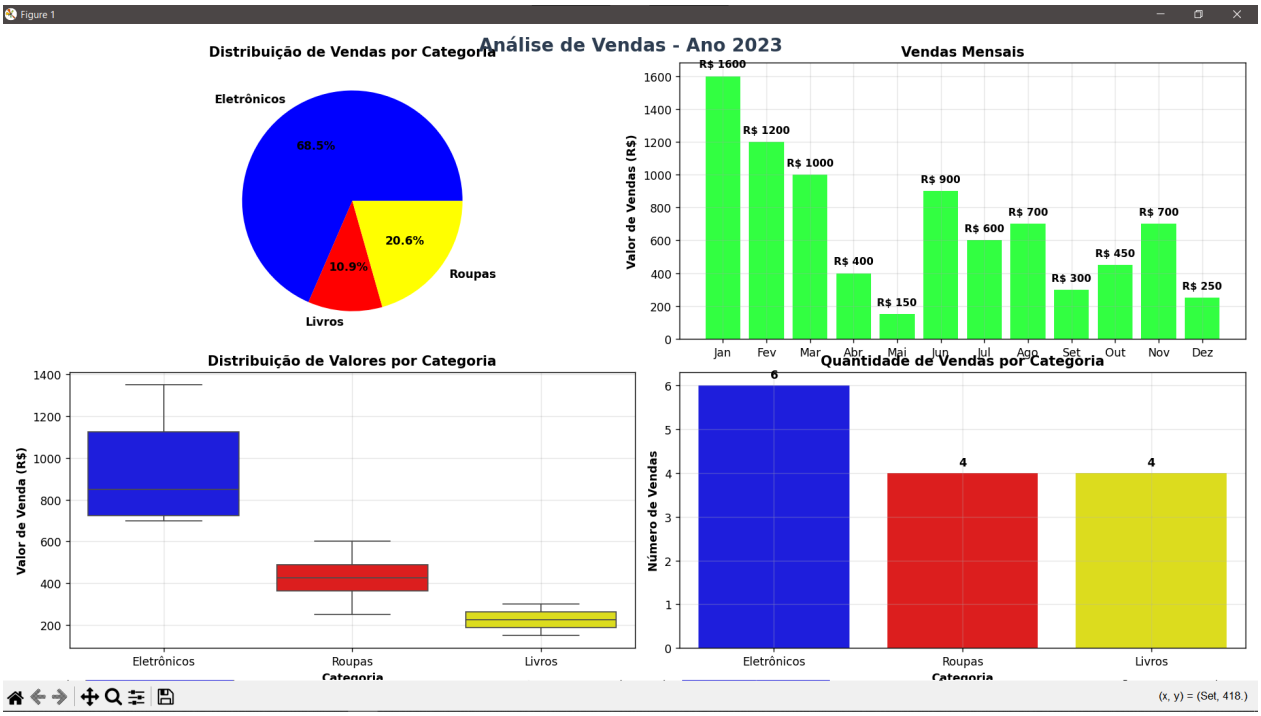
=> Figura 1: Print da tela com o código 1º trecho.



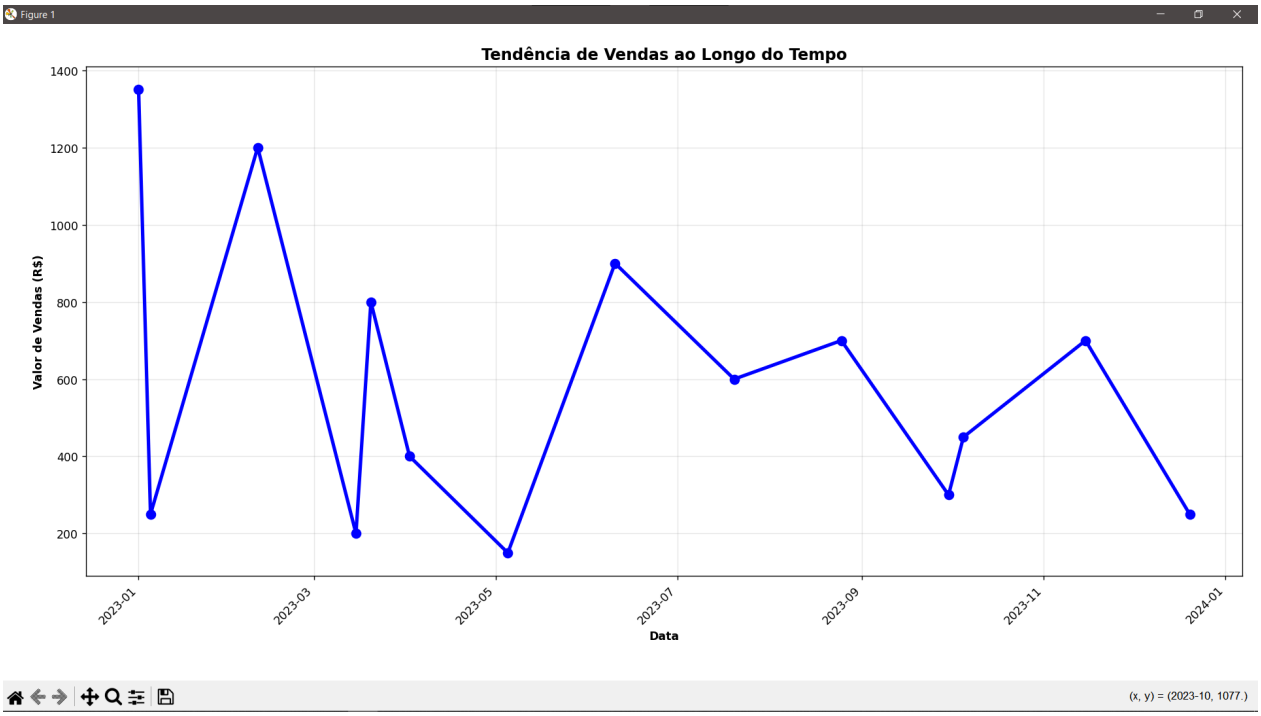
=> Figura 2: Print da tela com o código 2º trecho.



=> Figura 3: Print da tela do terminal executando as funções propostas.



=> Figura 4: Resultado do gráfico de análise das vendas 2023.



=> Figura 5: Resultado do gráfico de tendências de vendas.

📊 Lógica Utilizada para a Atividade de Análise de Dados

O sistema foi desenvolvido seguindo uma abordagem modular e orientada a objetos, implementando os seguintes componentes principais:

🏗️ Arquitetura do Sistema:

1. Estrutura em Camadas:

- **Banco de Dados SQLite:** Armazenamento persistente dos dados de vendas.
- **Camada de ETL:** Extração, Transformação e Carregamento com *Pandas*.
- **Camada de Análise:** Processamento e cálculo de métricas.
- **Camada de Visualização:** Geração de gráficos com *Matplotlib/Seaborn*.
- **Camada de Apresentação:** Relatório final com insights.

2. Fluxo de Dados:

```
SQLite → Pandas (DataFrame) → Análise Estatística → Visualização → Insights
```

🔗 Lógica de Implementação:

1. Gestão do Banco de Dados:

- Criação automática da tabela vendas se não existir.
- Inserção de dados de exemplo com transação controlada.
- Conexão segura com tratamento de exceções.
- Caminho relativo para portabilidade do projeto.

2. Processamento de Dados com Pandas:

```
# Transformações aplicadas:
df['data_venda'] = pd.to_datetime(df['data_venda'])  # Conversão para datetime
df['mes'] = df['data_venda'].dt.month                # Extração do mês
df['trimestre'] = df['data_venda'].dt.quarter        # Extração do trimestre
```

3. Análise Estatística:

- ❖ **Agregações por categoria:** Soma e contagem de vendas.
- ❖ **Análise temporal:** Vendas mensais e sazonais.
- ❖ **Estatísticas descritivas:** Média, máximo, mínimo, soma total.
- ❖ **Cálculo de participação:** percentual por categoria.

4. Visualização de Dados:

- **Gráficos implementados:**
- ❖ **Pizza:** Distribuição percentual por categoria.
- ❖ **Barras:** Vendas mensais com valores anotados.
- ❖ **Boxplot:** Distribuição de valores por categoria.
- ❖ **Countplot:** Quantidade de vendas por categoria.
- ❖ **Linha:** Tendência temporal das vendas.

5. Palette de Cores:

- **Cores vibrantes para melhor visualização:**
- ❖ #0000FF.
- ❖ #FF0000.
- ❖ #FFFF00.
- ❖ #00FF15.

6. Geração Automática de Insights:

- ❖ Identificação do mês de pico de vendas.
- ❖ Categoria mais lucrativa.
- ❖ Participação percentual por segmento.
- ❖ Recomendações estratégicas baseadas em dados.

🎯 Pontos Técnicos Destacados:

1. Tratamento de Erros:

```
try:
    # Operações críticas
except Exception as e:
    print(f"❌ Erro: {e}")  # Feedback amigável ao usuário
```

2. Otimizações:

- ❖ Re-indexação para meses sem vendas (evita gaps nos gráficos).
- ❖ Prevenção de duplicação de dados no banco .
- ❖ Layout responsivo com plt.tight_layout().
- ❖ Labels claros e informações anotadas.

3. Padrões de Projeto:

- ❖ Separação de concerns em funções especializadas.
- ❖ Data-driven programming com *Pandas*.
- ❖ Visualization pipeline com *Matplotlib*.
- ❖ Relatório automático com insights acionáveis.

📈 Valor Gerado:

O sistema transforma dados brutos em:

✅ Informações estruturadas. ✅ Visualizações compreensíveis.

✅ Insights acionáveis. ✅ Recomendações estratégicas.

Resultado: Uma solução completa de business intelligence em Python que permite à empresa tomar decisões baseadas em dados concretos sobre performance de vendas, sazonalidade e mix de produtos. 🚀