ESCOLA E FACULDADE DE TECNOLOGIA SENAI GASPAR RICARDO JÚNIOR

DESENVOLVIMENTO DE SISTEMAS

FELIPE MARQUES

GABRIEL RIBEIRO DE CAMARGO MIRANDA

JOÃO VICTOR DE SOUZA SANTOS

LUCAS MIGUEL LEITE

PROFESSOR ANDRÉ CASSULINO ARAÚJO SOUZA

CIÊNCIA DE DADOS

ANÁLISE ESTATÍSTICA COM PYTHON

SOROCABA

20/06/2025

SUMÁRIO

1 INTRODUÇÃO

A análise de dados tem se mostrado uma ferramenta fundamental para a tomada de decisões estratégicas, especialmente no contexto empresarial. Entre suas diversas aplicações, destaca-se a análise de vendas e do comportamento de consumo, que possibilita compreender padrões, antecipar tendências e identificar oportunidades de crescimento no mercado.

Este projeto tem como objetivo aplicar os conhecimentos adquiridos nas aulas de Ciência de Dados, ministradas pelo professor André, por meio da análise de dados relacionados a vendas e hábitos de consumo. A intenção é extrair informações relevantes, como sazonalidades e preferências dos consumidores, evidenciando, na prática, o potencial da análise de dados como apoio à tomada de decisões comerciais.

A escolha do tema se justifica por sua ampla aplicabilidade e relevância no cenário corporativo. Compreender o comportamento dos consumidores permite às empresas desenvolver estratégias mais eficazes e competitivas. Os códigos utilizados na análise estão disponíveis no **Apêndice A**, no Google Colab e no repositório do GitHub, ambos citados na seção de Referências.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os principais conceitos estatísticos e computacionais aplicados na análise dos dados, como técnicas de amostragem, medidas descritivas, testes de normalidade e correlação.

2.1 Conceitos Estatísticos e Computacionais

2.1.1 Amostragem

Foram aplicadas três técnicas de amostragem:

2.1.1.1 Amostragem Aleatória Simples:

- Seleção aleatória de registros com df.sample();
- Garante que cada elemento tenha igual probabilidade de seleção;
- Utilizada para análises gerais.

2.1.1.2 Amostragem Sistemática:

- Seleção periódica (ex.: a cada 10º registro) com df.iloc[::10];
- Útil quando os dados estão ordenados de forma não enviesada.

2.1.1.3 Amostragem Estratificada:

- Divisão proporcional por categoria (stratify=df['Product Category']);
- Garante representatividade em subgrupos.

2.1.2 Escalas de Medição

- **Nominal**:
  - Dados categóricos sem ordem (ex.: Product Category, Payment Method).
- **Razão**:
  - Dados numéricos com zero absoluto (ex.: Units Sold, Total Revenue).

2.1.3 Medidas Descritivas

- **Tendência Central**:
    - **Média**: Valor médio (df.mean());
    - **Mediana**: Valor central (df.median());
    - **Moda**: Valor mais frequente (df.mode()).
- **Dispersão**:
    - **Amplitude**: Diferença entre máximo e mínimo (df.max() - df.min());
    - **Variância**: Medida de dispersão (df.var());
    - **Desvio Padrão**: Dispersão em relação à média (df.std()).

2.1.4 Testes de Normalidade

- **Shapiro-Wilk**:
    - Testa se os dados seguem distribuição normal;
    - p-valor < 0.05 indica não normalidade.
- **QQ Plot**:
    - Gráfico para comparar distribuição dos dados com a normal.

2.1.5 Correlação

- **Pearson**: Mede correlação linear (adequado para dados normais).
- **Spearman**: Mede correlação monotônica (não exige normalidade).

2.2 Bibliotecas Python Utilizadas

Biblioteca	Função
Pandas	Manipulação de DataFrames (df.head(), df.dropna()).
NumPy	Cálculos numéricos (np.mean(), np.std()).
Matplotlib	Visualização básica (plt.hist(), plt.scatter()).
Seaborn	Visualização avançada (sns.boxplot(), sns.heatmap()).
SciPy	Testes estatísticos (stats.shapiro(), stats.pearsonr()).

Scikit-learn	Divisão de dados (train_test_split() para amostragem estratificada).

3 METODOLOGIA

A metodologia adotada envolveu a utilização de uma base de dados em formato CSV, contendo informações sobre vendas, categorias de produtos, regiões e métodos de pagamento. Foram realizadas etapas de tratamento e limpeza dos dados, com a seleção de colunas relevantes, remoção de valores nulos e aplicação de técnicas de amostragem aleatória, sistemática e estratificada.

3.1 Descrição da Base de Dados

- **Fonte**: Arquivo CSV (online_sales_data.csv);
- **Variáveis Analisadas**:
    - **Qualitativas**:
        - Product Category (Nominal;
        - Region (Nominal);
        - Payment Method (Nominal).
    - **Quantitativas**:
        - Units Sold (Razão);
        - Total Revenue (Razão).

3.2 Tratamento e Limpeza de Dados

3.2.1 Seleção de Colunas Relevantes:

- `df = df[['Product Category', 'Units Sold', 'Total Revenue', 'Region', 'Payment Method']].dropna()`

3.2.2 Remoção de Valores Nulos:

- dropna() para garantir consistência.

3.2.3 Amostragem:

- Aleatória, sistemática e estratificada para diferentes análises.

3.3 Ferramentas e Processos

- **Ambiente**: Google Colab;
- **Processos Principais**:
    - Análise exploratória (df.describe());
    - Visualização (sns.histplot(), plt.boxplot());
    - Testes estatísticos (shapiro(), pearsonr()).

4 ANÁLISE DE DADOS

A análise dos dados permitiu explorar padrões relevantes no comportamento de vendas e consumo. Foram aplicadas técnicas estatísticas e visuais, como histogramas, boxplots, scatterplots e testes de normalidade, para examinar distribuições, identificar outliers e avaliar relações entre variáveis.

4.1 Aplicação dos Tópicos Selecionados

4.1.1 Distribuição de Frequência:

- Histograma da Total Revenue mostrou assimetria à direita.

4.1.2 Comparação entre Categorias:

- Boxplot revelou outliers em categorias específicas.

4.1.3 Relação entre Variáveis:

- Scatterplot mostrou correlação negativa entre Units Sold e Total Revenue.

4.1.4 Teste de Normalidade:

- Shapiro-Wilk confirmou não normalidade (p-valor < 0.05).

4.1.5 Análise de Correlação:

- Pearson: -0.171 (fraca correlação negativa);
- Spearman: -0.140 (relação monotônica fraca).

4.1.6 Medidas de Dispersão:

- Total Revenue apresentou alto desvio padrão (485.80).

4.2 Visualizações e Interpretações

- **Gráfico 1**: Histograma de Total Revenue com distribuição assimétrica;
- **Gráfico 2**: Boxplot mostrando variação de receita por categoria;
- **Gráfico 3**: Mapa de calor indicando baixa correlação entre variáveis.

4.3 Discussão dos Resultados

- **Baixa Correlação**: Sugere que produtos com mais unidades vendidas não necessariamente geram maior receita;
- **Outliers**: Indicam vendas excepcionais em categorias específicas;
- **Não Normalidade**: Implica que testes não paramétricos (ex.: Spearman) são mais adequados.

5 CONCLUSÃO

A etapa final deste trabalho reúne reflexões a partir dos resultados obtidos ao longo da análise de dados.

5.1 Principais Achados

- **Heterogeneidade**: Grande variação em Total Revenue;
- **Categorias Dominantes**: Algumas se destacam em vendas;
- **Dados Não Normais**: Impacta escolha de métodos estatísticos.

5.2 Limitações

- **Tamanho da Amostra**: Pode afetar testes de normalidade;
- **Desequilíbrio**: Categorias com poucos registros.

5.3 Sugestões para Análises Futuras

- **Transformação de Dados**: Aplicar log para normalizar;
- **Análise Temporal**: Verificar sazonalidade;
- **Segmentação Avançada**: Machine Learning para identificar padrões.

REFERÊNCIAS

Normas da ABNT: regras de formatação para trabalhos acadêmicos. Disponível em: <https://www.todamateria.com.br/normas-abnt-trabalhos/>. Acesso em: 19 jun. 2025.

Ciencia_dados: Repositorio para salvar trabalho referente a disciplina de ciencia de dados. Disponível em: <https://github.com/Trabalho-BD/ciencia_dados>. Acesso em: 20 jun. 2025.

SOUZA, A. **Ciência de dados at main · profAndreSouza/Material**. Disponível em: <https://github.com/profAndreSouza/Material/tree/main/Ci%C3%AAncia%20de%20Dados>. Acesso em: 20 jun. 2025.

Google colab. Disponível em: <https://colab.research.google.com/drive/1CxTB0AITDrERKeyCHMWFOhPFtBvKpdTs>. Acesso em: 20 jun. 2025.

APÊNDICE A - CÓDIGOS DO GOOGLE COLAB

```
# 1. Bibliotecas

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from sklearn.model_selection import train_test_split

import pylab

from google.colab import files


sns.set(style="whitegrid")

plt.rcParams["figure.figsize"] = (10, 6)


# Upload do CSV

uploaded = files.upload()


# Leitura do CSV

df = pd.read_csv('online_sales_data.csv')
```

```
# Visualizar colunas e primeiras linhas

print(df.columns)

df.head()



# Renomear e limpar colunas

df = df[['Product Category', 'Units Sold', 'Total Revenue', 'Region', 'Payment Method']].dropna()


df.head()


# Aleatória simples

amostra_aleatoria = df.sample( random_state=42)


# Sistemática (a cada 10 linhas)

amostra_sistematica = df.iloc[::10]


# Estratificada por categoria

amostra_estratificada, _ = train_test_split(df, test_size=0.8, stratify=df['Product Category'], random_state=42)
```

```
# Exemplo de classificação
escalas = {
    'Product Category': 'Nominal',
    'Region': 'Nominal',
    'Payment Method': 'Nominal',
    'Units Sold': 'Razão',
    'Total Revenue': 'Razão'
}
pd.DataFrame(list(escalas.items()), columns=['Variável', 'Escala de Medição'])

# Tendência Central
print('Médias:\n', df[['Units Sold', 'Total Revenue']].mean())
print('\nMedianas:\n', df[['Units Sold', 'Total Revenue']].median())
print('\nModas:\n', df[['Units Sold', 'Total Revenue']].mode().iloc[0])

# Dispersão
print('\nAmplitude:\n', df[['Units Sold', 'Total Revenue']].max() - df[['Units Sold', 'Total Revenue']].min())
```

```
print('\nVariância:\n', df[['Units Sold', 'Total Revenue']].var())

print('\nDesvio Padrão:\n', df[['Units Sold', 'Total Revenue']].std())


sns.histplot(df['Total Revenue'], bins=30, kde=True)

plt.title("Distribuição de Receita Total")

plt.xlabel("Total Revenue")

plt.ylabel("Frequência")

plt.show()


# Boxplot por Categoria

sns.boxplot(x='Product Category', y='Total Revenue', data=df)

plt.title("Receita Total por Categoria de Produto")

plt.xticks(rotation=45)

plt.show()


# Scatterplot: Units Sold vs Total Revenue

sns.scatterplot(x='Units Sold', y='Total Revenue', hue='Product Category', data=df)

plt.title("Relação entre Unidades Vendidas e Receita")

plt.show()
```

```
# Teste Shapiro-Wilk

for col in ['Units Sold', 'Total Revenue']:

    stat, p = stats.shapiro(df[col].dropna().sample(240, random_state=42))

    print(f"{col}: W={stat:.3f}, p-valor={p:.4f}")


# QQ Plot: Total Revenue

stats.probplot(df['Total Revenue'].sample(240, random_state=42), dist="norm", plot=pylab)

plt.title("QQ Plot - Receita Total")

plt.show()


# Pearson e Spearman

corr_pearson = df['Units Sold'].corr(df['Total Revenue'], method='pearson')

corr_spearman = df['Units Sold'].corr(df['Total Revenue'], method='spearman')


print(f"Correlação de Pearson: {corr_pearson:.3f}")

print(f"Correlação de Spearman: {corr_spearman:.3f}")


# Mapa de calor

sns.heatmap(df[['Units Sold', 'Total Revenue']].corr(), annot=True, cmap='viridis')
```

```
plt.title("Mapa de Correlação")

plt.show()
```