Análise Exploratória de Dados (AED)

Visualizar e Interpretar Dados para Identificar Padrões e Anomalias

Márcio Nicolau

2025-09-10

Table of contents

List of Figures

Introdução e Objetivos

Nas aulas anteriores, construímos uma base sólida em análise combinatória e probabilidade, e aprendemos a descrever dados através de medidas de tendência central e dispersão. Agora, vamos mergulhar na **Análise Exploratória de Dados (AED)**, uma filosofia e um conjunto de técnicas para investigar conjuntos de

dados, resumir suas características principais, e identificar padrões, anomalias e relações. A AED é um passo crítico no pipeline de ciência de dados, frequentemente o primeiro após a coleta e limpeza de dados.

A AED vai além dos números brutos, utilizando visualizações para contar a história dos dados. Ela nos permite formular hipóteses, validar suposições sobre os dados e preparar o terreno para modelagem estatística e aprendizado de máquina. Através de gráficos e tabelas, podemos descobrir insights que seriam difíceis de perceber apenas com medidas numéricas.

Objetivos de Aprendizagem

Ao final desta aula, você será capaz de:

- Compreender o propósito e a importância da Análise Exploratória de Dados.
- Utilizar diferentes tipos de gráficos para visualizar variáveis univariadas (quantitativas e qualitativas).
- Utilizar diferentes tipos de gráficos para explorar relações bivariadas (entre dois tipos de variáveis).
- Identificar visualmente padrões, tendências, distribuições e outliers nos dados.
- Interpretar as informações extraídas das visualizações para gerar insights.
- Aplicar ferramentas de visualização de dados em Python (`matplotlib`, `seaborn`) e R (`ggplot2`).

O que é Análise Exploratória de Dados (AED)?

A **Análise Exploratória de Dados (AED)**, popularizada por John Tukey, é uma abordagem para analisar conjuntos de dados para resumir suas características principais, muitas vezes com métodos visuais. Um modelo estatístico pode ser usado ou não, mas a AED é primariamente para ver o que os dados podem nos dizer além do formalismo. (Bussab; Morettin, 2017, p. 95–96)

A AED é um processo iterativo e investigativo que envolve:

- **Entendimento do Contexto:** Compreender o que os dados representam e de onde vieram.
- **Limpeza e Preparação:** Lidar com valores ausentes, formatar dados e corrigir erros.
- **Visualização:** Criar gráficos para revelar distribuições, relações e anomalias.
- **Resumo Numérico:** Calcular medidas descritivas (média, mediana, desvio padrão, etc.) para complementar as visualizações.
- **Formulação de Hipóteses:** Gerar perguntas sobre os dados que podem ser testadas posteriormente.

A AED é crucial porque:

- **Revela Insights:** Ajuda a descobrir padrões e relações que não seriam evidentes de outra forma.
- **Identifica Problemas:** Permite detectar erros nos dados, outliers, e problemas de qualidade.
- **Guia a Modelagem:** Informa a escolha de modelos estatísticos e técnicas de aprendizado de máquina.
- **Valida Suposições:** Permite verificar as suposições subjacentes aos métodos estatísticos.

Fluxo de Trabalho Típico da AED

Visualização de Dados para Variáveis Únicas (Análise Univariada)

A análise univariada foca na distribuição de cada variável individualmente.

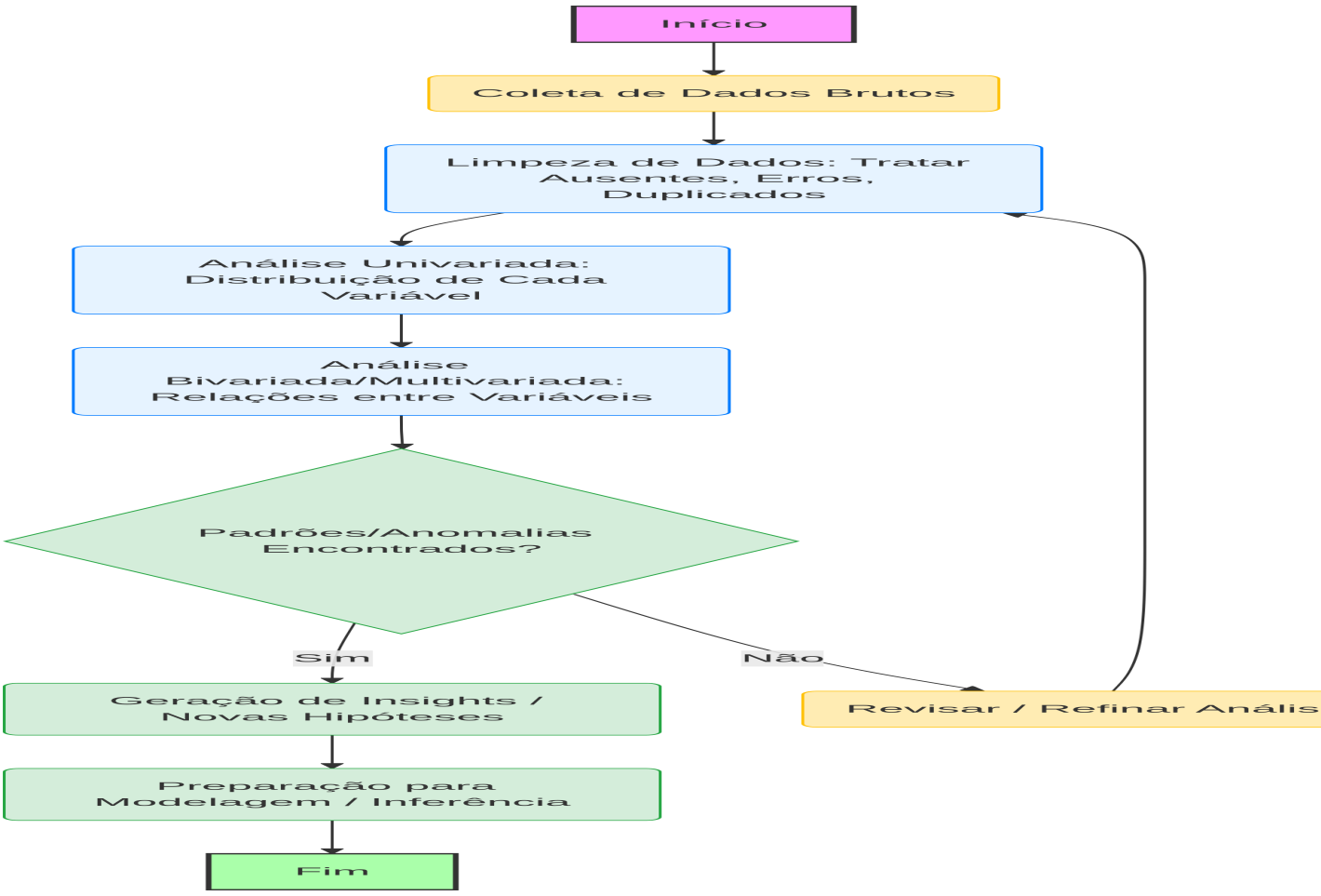

Figure 1: Fluxo de Trabalho Típico da Análise Exploratória de Dados (AED)

Variáveis Qualitativas (Nominais e Ordinais)

Para variáveis categóricas, os gráficos nos ajudam a visualizar as frequências de cada categoria.

Gráfico de Barras (Bar Plot)

Mostra a frequência ou contagem de cada categoria. É ideal para comparar a magnitude das categorias.

Exemplo: Contagem de diferentes cores de carros em um estacionamento.

Python

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Criando um dataset de exemplo
data_qual = {'Cor_Carro': ['Vermelho', 'Azul', 'Verde', 'Vermelho', 'Azul', 'Azul', 'Branco', 'Vermelho
df_qual = pd.DataFrame(data_qual)

plt.figure(figsize=(8, 5))
sns.countplot(x='Cor_Carro', data=df_qual, palette='viridis')
plt.title('Frequência de Cores de Carros')
plt.xlabel('Cor do Carro')
plt.ylabel('Contagem')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)
library(dplyr)

# Criando um dataset de exemplo
df_qual <- data.frame(
  Cor_Carro = c('Vermelho', 'Azul', 'Verde', 'Vermelho', 'Azul', 'Azul', 'Branco', 'Vermelho', 'Verde')
)

ggplot(df_qual, aes(x = Cor_Carro, fill = Cor_Carro)) +
  geom_bar() +
  labs(title = 'Frequência de Cores de Carros', x = 'Cor do Carro', y = 'Contagem') +
  theme_minimal() +
  theme(legend.position = "none") + # Oculta a legenda se a cor já estiver no eixo x
  scale_fill_viridis_d() # Usando uma paleta de cores discreta
```

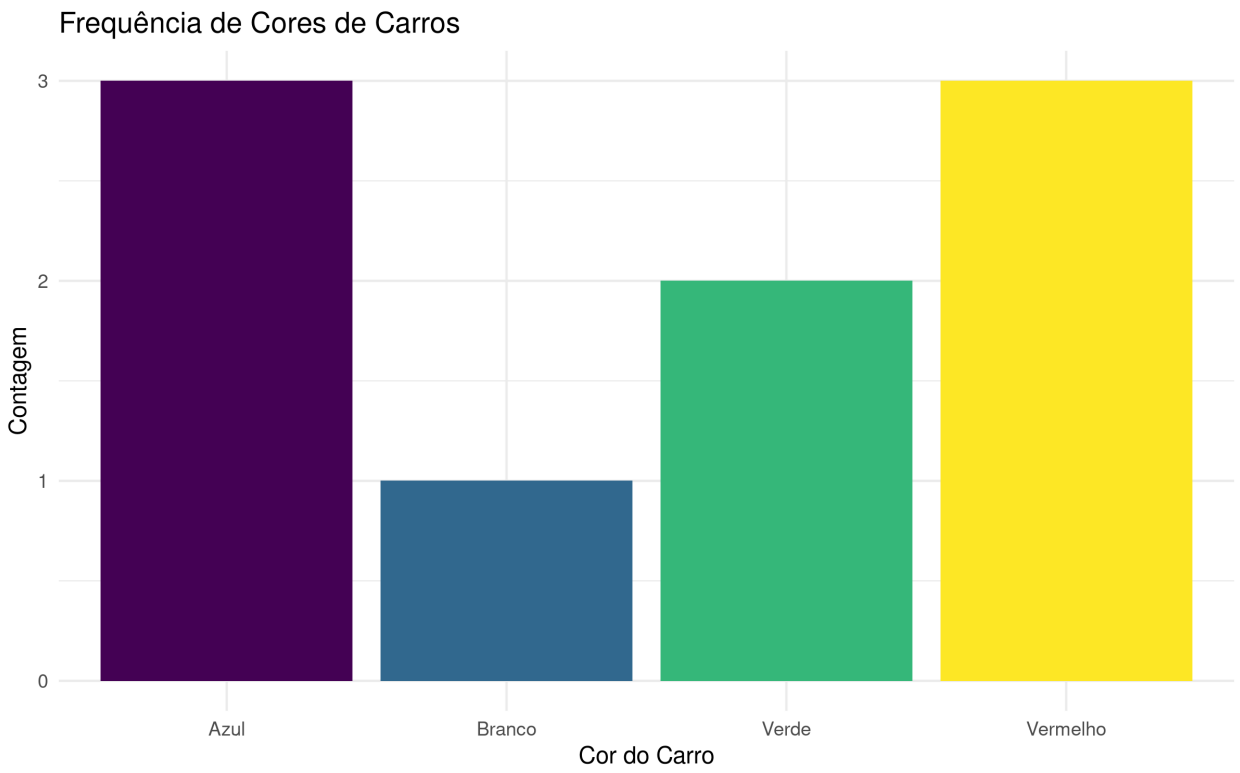

Figure 2: Gráfico de Barras para Variáveis Qualitativas

Gráfico de Setores (Pie Chart)

Representa a proporção de cada categoria em relação ao total. Útil para poucas categorias, mas pode ser enganoso com muitas.

Exemplo: Distribuição percentual do voto em uma eleição.

Python

```python
import pandas as pd
import matplotlib.pyplot as plt

# Contando as frequências do exemplo anterior
contagem_cores = df_qual['Cor_Carro'].value_counts()

plt.figure(figsize=(8, 8))
plt.pie(contagem_cores, labels=contagem_cores.index, autopct='%1.1f%%', startangle=140, colors=sns.colo
plt.title('Distribuição Percentual de Cores de Carros')
plt.axis('equal') # Garante que o círculo seja desenhado como um círculo
plt.show()
```

R

```r
# Contando as frequências do exemplo anterior
contagem_cores <- as.data.frame(table(df_qual$Cor_Carro))
colnames(contagem_cores) <- c("Cor", "Frequencia")

ggplot(contagem_cores, aes(x = "", y = Frequencia, fill = Cor)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = 'Distribuição Percentual de Cores de Carros') +
  theme_void() + # Tema vazio para gráficos de pizza
  geom_text(aes(label = paste0(round(Frequencia/sum(Frequencia)*100, 1), "%")),
            position = position_stack(vjust = 0.5), size = 4) +
  scale_fill_viridis_d()
```

Variáveis Quantitativas (Discretas e Contínuas)

Para variáveis numéricas, buscamos entender a forma da distribuição, a dispersão e a presença de valores atípicos.

Histograma

Mostra a distribuição de uma variável quantitativa, agrupando os valores em "bins" (intervalos) e exibindo a frequência de cada bin.

Exemplo: Distribuição de idades de clientes.

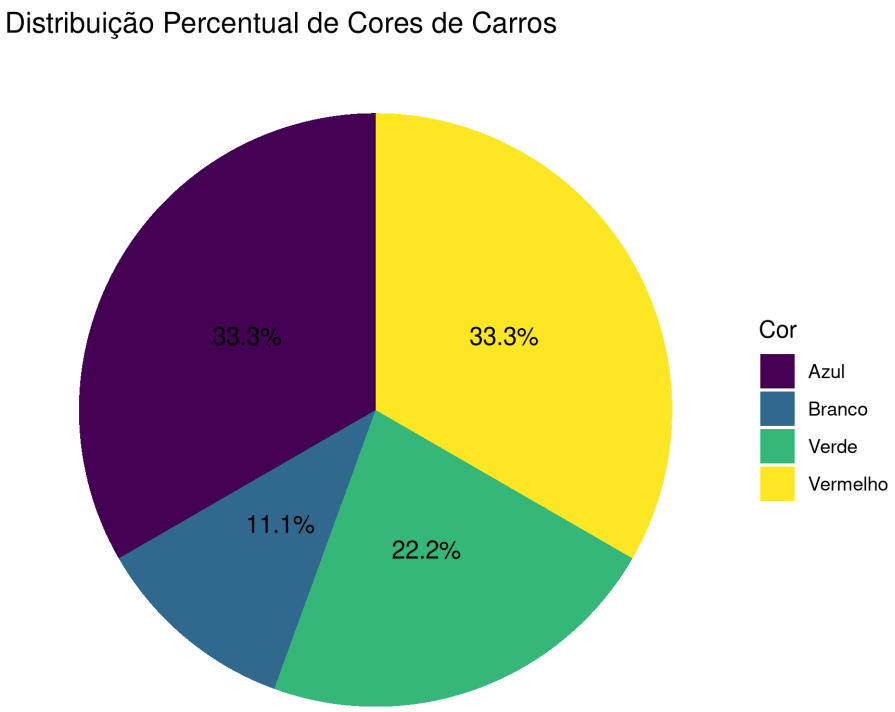

Figure 3: Gráfico de Setores para Variáveis Qualitativas

Python

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Criando um dataset de exemplo
np.random.seed(42)
idades = np.random.normal(loc=30, scale=8, size=100) # Média 30, desvio padrão 8
idades = idades[(idades > 18) & (idades < 60)] # Filtrando para idades mais realistas
idades = np.round(idades).astype(int) # Arredondando para inteiros

plt.figure(figsize=(10, 6))
sns.histplot(idades, bins=10, kde=True, color='skyblue', edgecolor='black')
plt.title('Distribuição de Idades de Clientes')
plt.xlabel('Idade')
plt.ylabel('Frequência')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Criando um dataset de exemplo
set.seed(42)
idades <- rnorm(100, mean = 30, sd = 8)
idades <- idades[idades > 18 & idades < 60]
idades <- round(idades)

df_idades <- data.frame(Idade = idades)

ggplot(df_idades, aes(x = Idade)) +
  geom_histogram(binwidth = 3, fill = "skyblue", color = "black", alpha = 0.7) +
  geom_density(aes(y = after_stat(count * 3)), color = "blue", size = 1) + # Multiplica por binwidth pa
  labs(title = 'Distribuição de Idades de Clientes', x = 'Idade', y = 'Frequência') +
  theme_minimal()
```

Box Plot (Diagrama de Caixa)

Fornece um resumo de cinco números: mínimo, primeiro quartil (Q1), mediana (Q2), terceiro quartil (Q3) e máximo. É excelente para identificar outliers e comparar distribuições.

Exemplo: Distribuição do tempo de resposta de servidores.

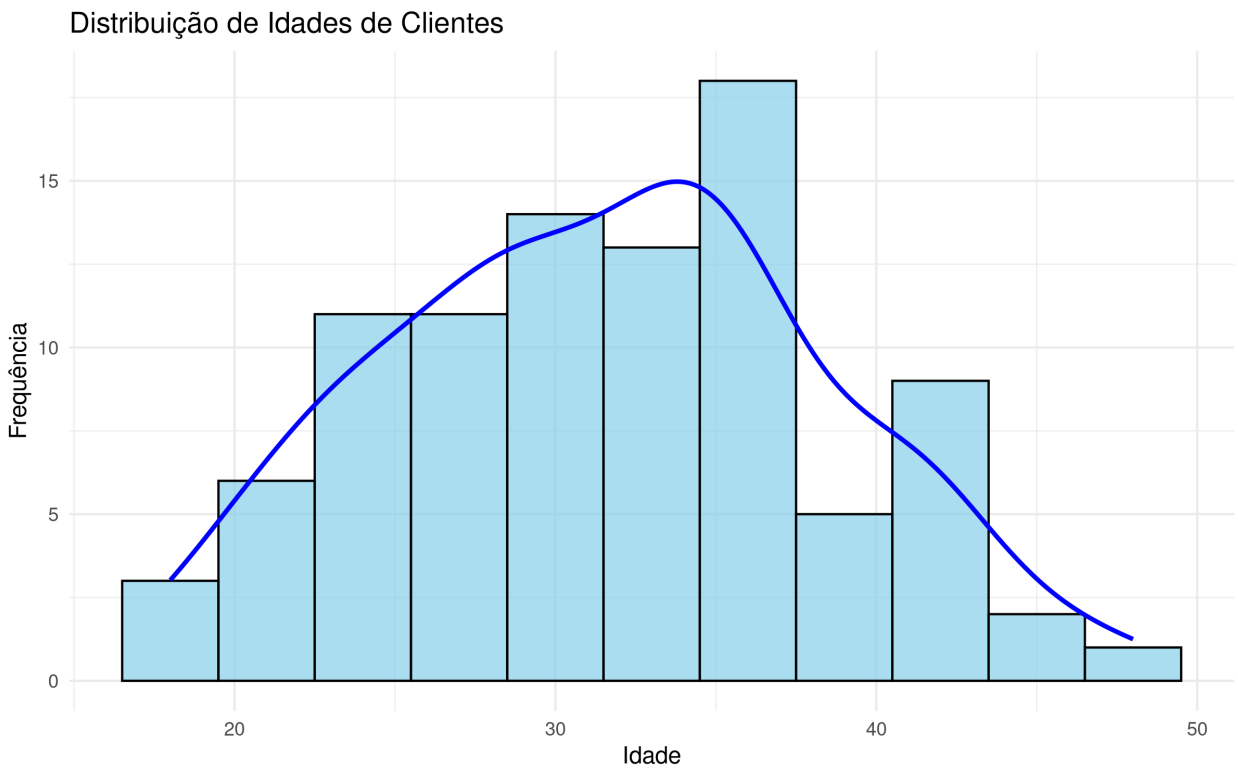

Figure 4: Gráfico de Histograma para Variáveis Quantitativas

Python

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Criando um dataset de exemplo com outliers
np.random.seed(42)
tempos_resposta = np.random.normal(loc=150, scale=20, size=50)
tempos_resposta = np.append(tempos_resposta, [30, 40, 500, 550]) # Adicionando outliers

plt.figure(figsize=(8, 6))
sns.boxplot(y=tempos_resposta, color='lightgreen')
plt.title('Distribuição do Tempo de Resposta de Servidores')
plt.ylabel('Tempo de Resposta (ms)')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Criando um dataset de exemplo com outliers
set.seed(42)
tempos_resposta <- rnorm(50, mean = 150, sd = 20)
tempos_resposta <- c(tempos_resposta, 30, 40, 500, 550)

df_tempos <- data.frame(Tempo = tempos_resposta)

ggplot(df_tempos, aes(y = Tempo)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = 'Distribuição do Tempo de Resposta de Servidores', y = 'Tempo de Resposta (ms)') +
  theme_minimal()
```

Gráfico de Densidade (KDE Plot)

Uma versão suavizada do histograma, que mostra a estimativa da função de densidade de probabilidade. É útil para visualizar a forma da distribuição sem a dependência da escolha dos bins.

Exemplo: Distribuição de pesos de amostras.

Python

```python
import numpy as np
import matplotlib.pyplot as plt
```

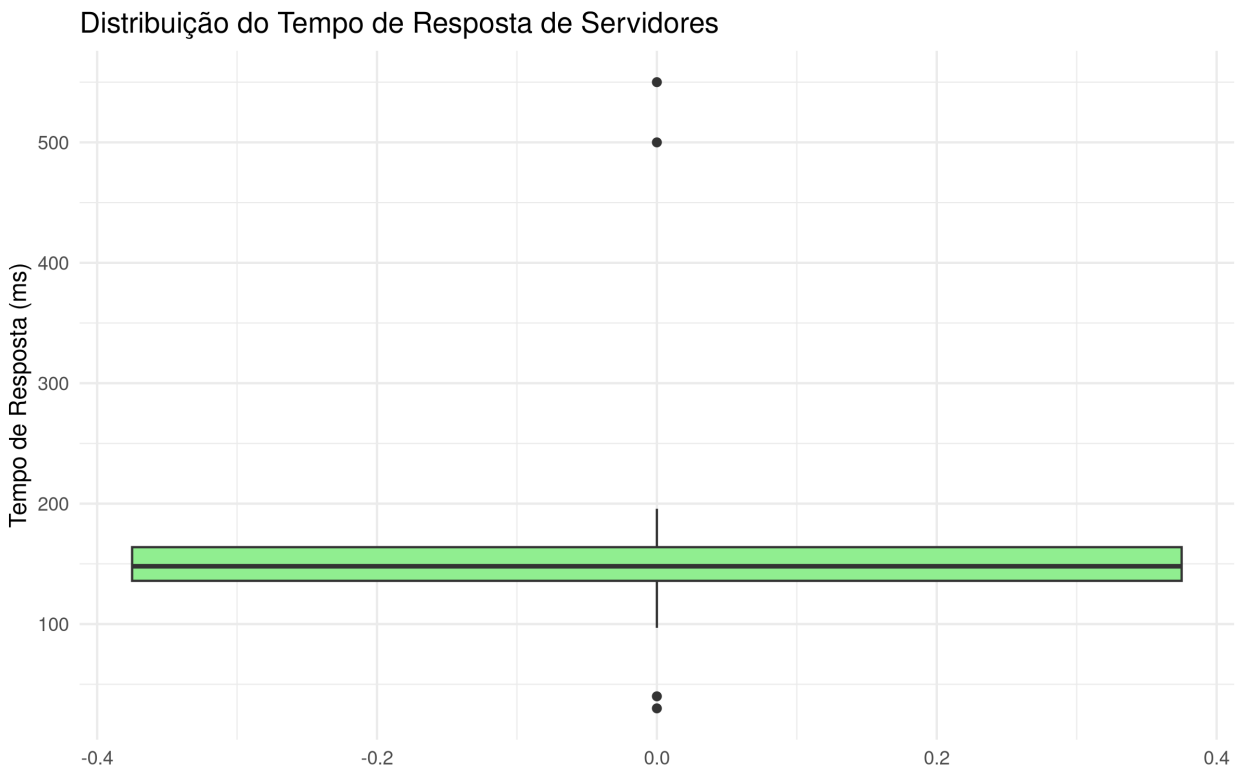

Figure 5: Gráfico de Caixa para Variáveis Quantitativas

```python
import seaborn as sns

# Reutilizando o dataset de idades
# idades = np.random.normal(loc=30, scale=8, size=100)
# idades = idades[(idades > 18) & (idades < 60)]

plt.figure(figsize=(10, 6))
sns.kdeplot(idades, fill=True, color='purple')
plt.title('Densidade da Distribuição de Idades de Clientes')
plt.xlabel('Idade')
plt.ylabel('Densidade')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Reutilizando o dataset de idades
# idades <- rnorm(100, mean = 30, sd = 8)
# idades <- idades[idades > 18 & idades < 60]
# df_idades <- data.frame(Idade = idades)

ggplot(df_idades, aes(x = Idade)) +
  geom_density(fill = "purple", alpha = 0.7) +
  labs(title = 'Densidade da Distribuição de Idades de Clientes', x = 'Idade', y = 'Densidade') +
  theme_minimal()
```

Visualização de Dados para Relações entre Variáveis (Análise Bivariada)

A análise bivariada explora a relação entre duas variáveis.

Quantitativa vs. Quantitativa

Gráfico de Dispersão (Scatter Plot)

Mostra a relação entre duas variáveis quantitativas. Cada ponto representa uma observação. É excelente para identificar padrões, correlações e clusters.

Exemplo: Relação entre horas de estudo e nota em um exame.

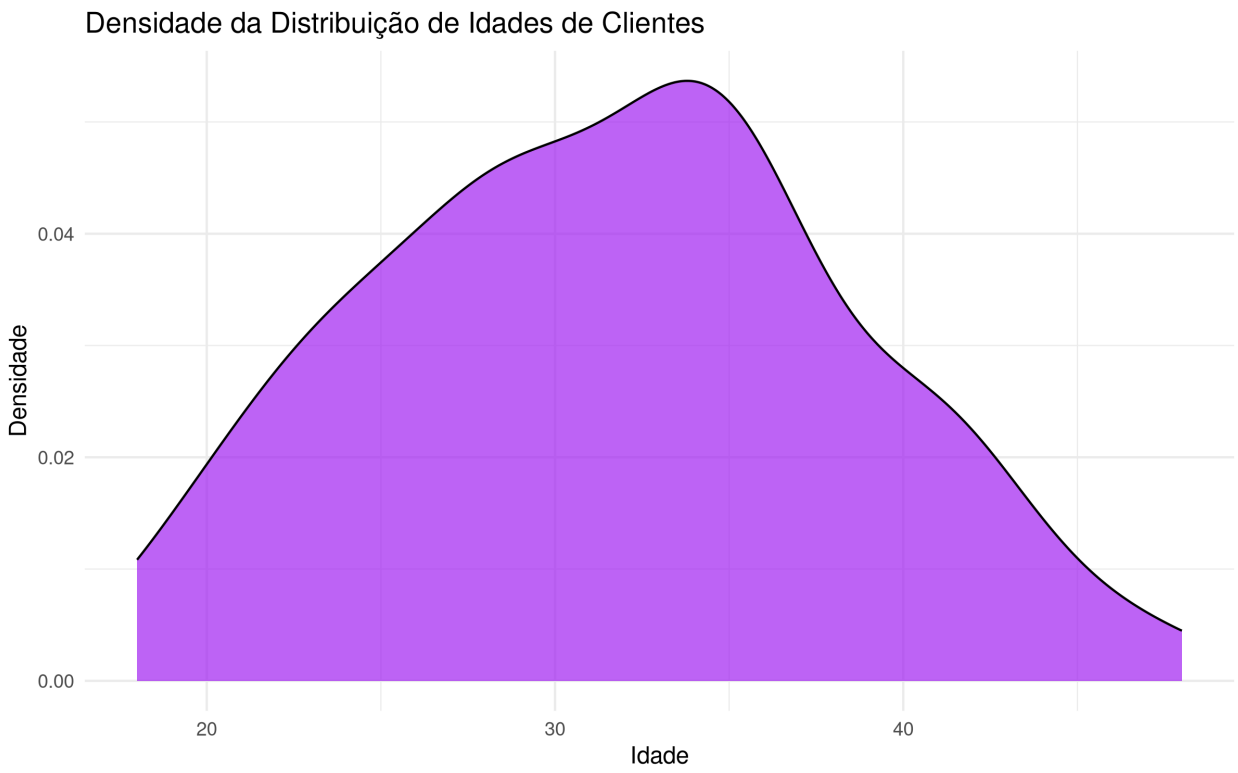

Figure 6: Gráfico de Densidade para Variáveis Quantitativas

Python

```python
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Criando um dataset de exemplo
np.random.seed(42)
horas_estudo = np.random.uniform(2, 10, 50)
notas = 50 + horas_estudo * 5 + np.random.normal(0, 7, 50)
notas = np.clip(notas, 0, 100) # Limita as notas entre 0 e 100

df_estudo = pd.DataFrame({'Horas_Estudo': horas_estudo, 'Notas': notas})

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Horas_Estudo', y='Notas', data=df_estudo, hue='Horas_Estudo', size='Notas', sizes=(20
plt.title('Relação entre Horas de Estudo e Notas em Exame')
plt.xlabel('Horas de Estudo')
plt.ylabel('Nota no Exame')
plt.grid(linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Criando um dataset de exemplo
set.seed(42)
horas_estudo <- runif(50, 2, 10)
notas <- 50 + horas_estudo * 5 + rnorm(50, 0, 7)
notas <- pmin(pmax(notas, 0), 100) # Limita as notas entre 0 e 100

df_estudo <- data.frame(Horas_Estudo = horas_estudo, Notas = notas)

ggplot(df_estudo, aes(x = Horas_Estudo, y = Notas, color = Horas_Estudo, size = Notas)) +
  geom_point(alpha = 0.7) +
  labs(title = 'Relação entre Horas de Estudo e Notas em Exame', x = 'Horas de Estudo', y = 'Nota no Exa
  theme_minimal() +
  scale_color_viridis_c()
```

Matriz de Correlação (Heatmap)

Quando há múltiplas variáveis quantitativas, um heatmap da matriz de correlação visualiza as forças e direções das relações lineares entre todos os pares de variáveis.

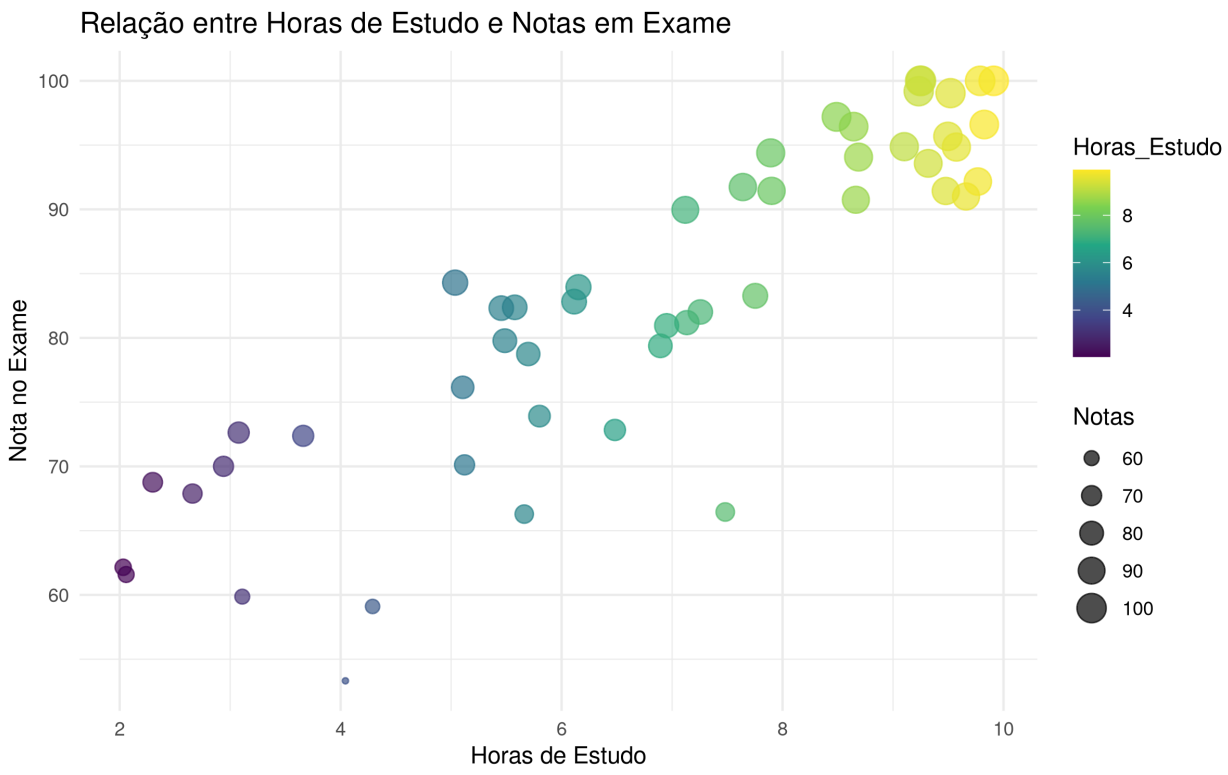

Figure 7: Gráfico de Dispersão para Variáveis Quantitativas

Exemplo: Correlação entre diferentes características de um dataset (e.g., Iris).

Python

```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Usando o dataset Iris para exemplo
iris = sns.load_dataset('iris')
numeric_iris = iris.select_dtypes(include=np.number) # Seleciona apenas colunas numéricas

plt.figure(figsize=(8, 6))
sns.heatmap(numeric_iris.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Matriz de Correlação do Dataset Iris')
plt.show()
```

R

```r
library(ggplot2)
library(reshape2) # Para a função melt

# Usando o dataset Iris para exemplo
data(iris)
numeric_iris <- iris %>% select_if(is.numeric)

# Calcular a matriz de correlação
cor_matrix <- cor(numeric_iris)

# Converter a matriz de correlação para um formato longo para ggplot2
melted_cor_matrix <- melt(cor_matrix)

ggplot(melted_cor_matrix, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlação") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  coord_fixed() +
  geom_text(aes(Var1, Var2, label = round(value, 2)), color = "black", size = 4) +
  labs(title = "Matriz de Correlação do Dataset Iris")
```

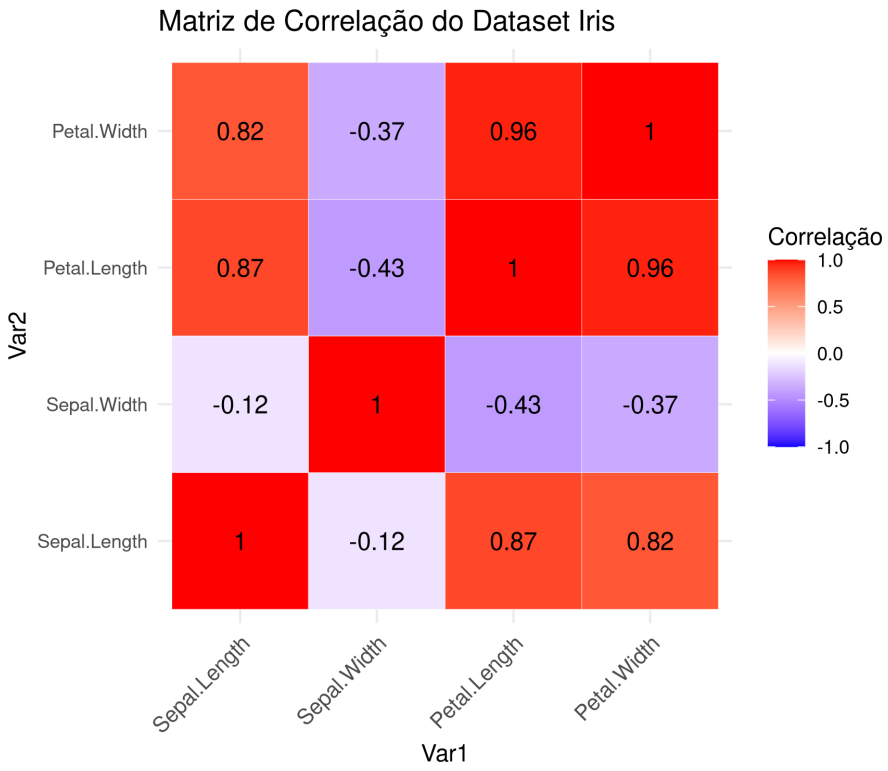

Figure 8: Matriz de Correlação do Dataset Iris

Qualitativa vs. Quantitativa

Box Plot (por Categoria)

Permite comparar a distribuição de uma variável quantitativa entre diferentes grupos (categorias de uma variável qualitativa).

Exemplo: Comparar salários por nível de escolaridade.

Python

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Criando um dataset de exemplo
data_mix = {
    'Escolaridade': ['Fundamental', 'Médio', 'Superior', 'Médio', 'Fundamental', 'Superior', 'Superior'
    'Salario': [2500, 3500, 6000, 3000, 2800, 7500, 5500, 3200]
}
df_mix = pd.DataFrame(data_mix)

plt.figure(figsize=(10, 6))
sns.boxplot(x='Escolaridade', y='Salario', data=df_mix, order=['Fundamental', 'Médio', 'Superior'], pal
plt.title('Salário por Nível de Escolaridade')
plt.xlabel('Nível de Escolaridade')
plt.ylabel('Salário (R$)')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Criando um dataset de exemplo
df_mix <- data.frame(
  Escolaridade = factor(c('Fundamental', 'Médio', 'Superior', 'Médio', 'Fundamental', 'Superior', 'Supe
                        levels = c('Fundamental', 'Médio', 'Superior')),
  Salario = c(2500, 3500, 6000, 3000, 2800, 7500, 5500, 3200)
)

ggplot(df_mix, aes(x = Escolaridade, y = Salario, fill = Escolaridade)) +
  geom_boxplot() +
  labs(title = 'Salário por Nível de Escolaridade', x = 'Nível de Escolaridade', y = 'Salário (R$)') +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2") # Usando uma paleta de cores
```

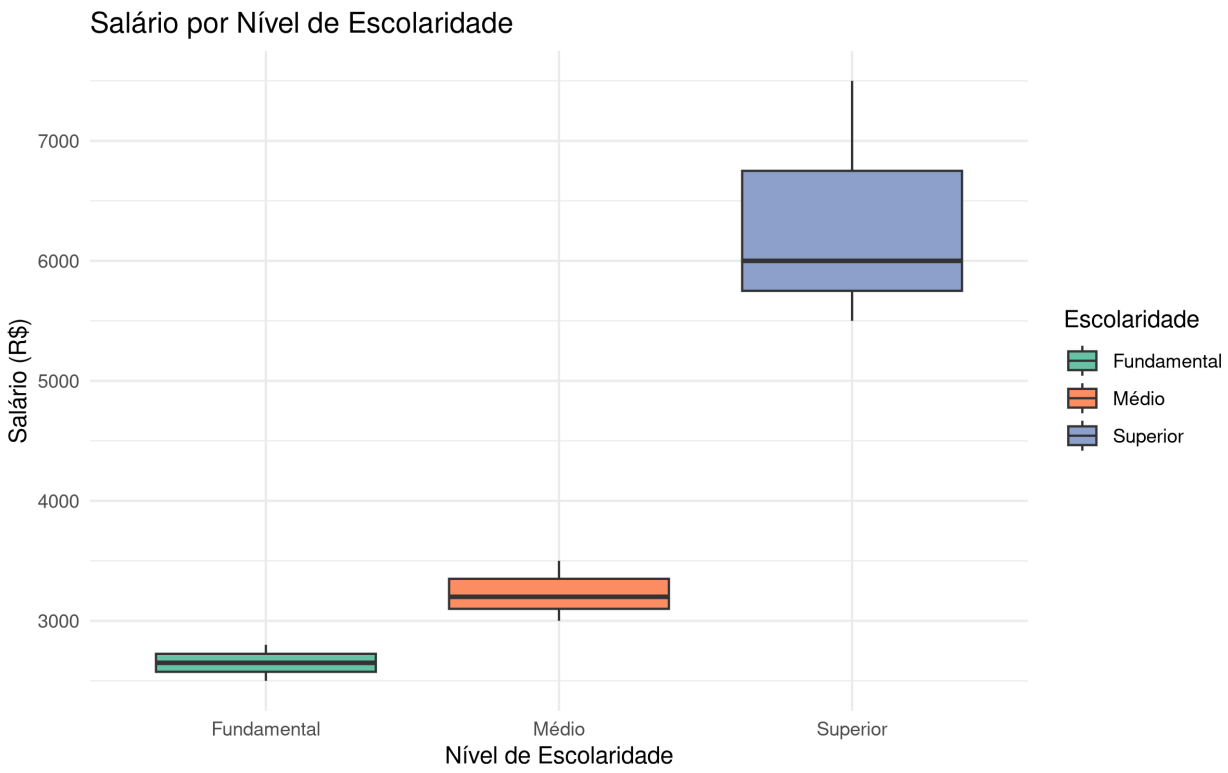

Figure 9: Box Plot para Variáveis Qualitativas

Violin Plot

Combina um box plot com um gráfico de densidade, mostrando a distribuição completa dos dados em cada categoria. É útil para ver a forma da distribuição, além dos quartis.

Exemplo: Comparar a distribuição do tempo de atendimento por tipo de serviço.

Python

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Criando um dataset de exemplo
np.random.seed(42)
data_violin = {
    'Tipo_Servico': np.random.choice(['Básico', 'Premium', 'Standard'], size=100),
    'Tempo_Atendimento': np.concatenate([
        np.random.normal(5, 1, 40), # Básico
        np.random.normal(8, 2, 30), # Premium
        np.random.normal(6.5, 1.5, 30) # Standard
    ])
}
df_violin = pd.DataFrame(data_violin)

plt.figure(figsize=(10, 6))
sns.violinplot(x='Tipo_Servico', y='Tempo_Atendimento', data=df_violin, palette='pastel')
plt.title('Distribuição do Tempo de Atendimento por Tipo de Serviço')
plt.xlabel('Tipo de Serviço')
plt.ylabel('Tempo de Atendimento (minutos)')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)

# Criando um dataset de exemplo
set.seed(42)
df_violin <- data.frame(
  Tipo_Servico = factor(sample(c('Básico', 'Premium', 'Standard'), size = 100, replace = TRUE,
                               prob = c(0.4, 0.3, 0.3))),
  Tempo_Atendimento = c(
    rnorm(40, mean = 5, sd = 1),
    rnorm(30, mean = 8, sd = 2),
```

```
    rnorm(30, mean = 6.5, sd = 1.5)
  )
)

ggplot(df_violin, aes(x = Tipo_Servico, y = Tempo_Atendimento, fill = Tipo_Servico)) +
  geom_violin(trim = FALSE) + # trim=FALSE mostra os "rabos" completos
  labs(title = 'Distribuição do Tempo de Atendimento por Tipo de Serviço', x = 'Tipo de Serviço', y = '
  theme_minimal() +
  scale_fill_brewer(palette = "Pastel1")
```

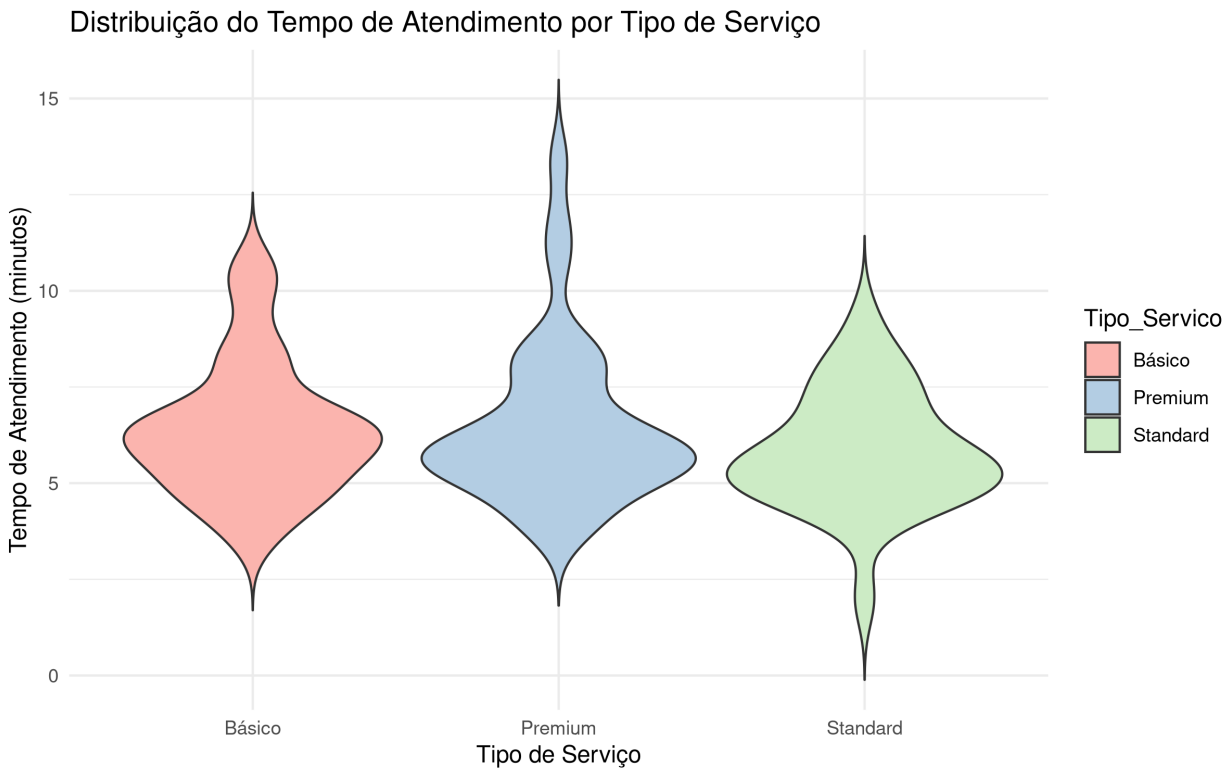

Figure 10: Violin Plot para Variáveis Quantitativas

Qualitativa vs. Qualitativa

Gráfico de Barras Agrupadas/Empilhadas

Compara as frequências ou proporções de uma categoria em relação a outra.

Exemplo: Gênero de clientes por tipo de produto adquirido.

Python

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Criando um dataset de exemplo
np.random.seed(42)
data_cat = {
    'Genero': np.random.choice(['Masculino', 'Feminino'], size=100),
    'Produto': np.random.choice(['Eletrônico', 'Vestuário', 'Alimentos'], size=100)
}
df_cat = pd.DataFrame(data_cat)

# Usando crosstab para criar a tabela de contingência
contingency_table = pd.crosstab(df_cat['Genero'], df_cat['Produto'])

# Gráfico de barras agrupadas
contingency_table.plot(kind='bar', figsize=(10, 6), colormap='tab10')
plt.title('Distribuição de Gênero por Tipo de Produto Adquirido')
plt.xlabel('Gênero')
plt.ylabel('Contagem')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()

# Gráfico de barras empilhadas (proporções)
contingency_table_percent = contingency_table.apply(lambda r: r/r.sum(), axis=1)
contingency_table_percent.plot(kind='bar', stacked=True, figsize=(10, 6), colormap='tab10')
plt.title('Proporção de Gênero por Tipo de Produto Adquirido')
plt.xlabel('Gênero')
plt.ylabel('Proporção')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```r
library(ggplot2)
library(dplyr)

# Criando um dataset de exemplo
set.seed(42)
df_cat <- data.frame(
```

```
  Genero = factor(sample(c('Masculino', 'Feminino'), size = 100, replace = TRUE)),
  Produto = factor(sample(c('Eletrônico', 'Vestuário', 'Alimentos'), size = 100, replace = TRUE))
)

# Gráfico de barras agrupadas
ggplot(df_cat, aes(x = Genero, fill = Produto)) +
  geom_bar(position = "dodge") +
  labs(title = 'Distribuição de Gênero por Tipo de Produto Adquirido', x = 'Gênero', y = 'Contagem') +
  theme_minimal() +
  scale_fill_brewer(palette = "Paired")

# Gráfico de barras empilhadas (proporções)
ggplot(df_cat, aes(x = Genero, fill = Produto)) +
  geom_bar(position = "fill") + # "fill" para proporções empilhadas
  labs(title = 'Proporção de Gênero por Tipo de Produto Adquirido', x = 'Gênero', y = 'Proporção') +
  theme_minimal() +
  scale_fill_brewer(palette = "Paired")
```

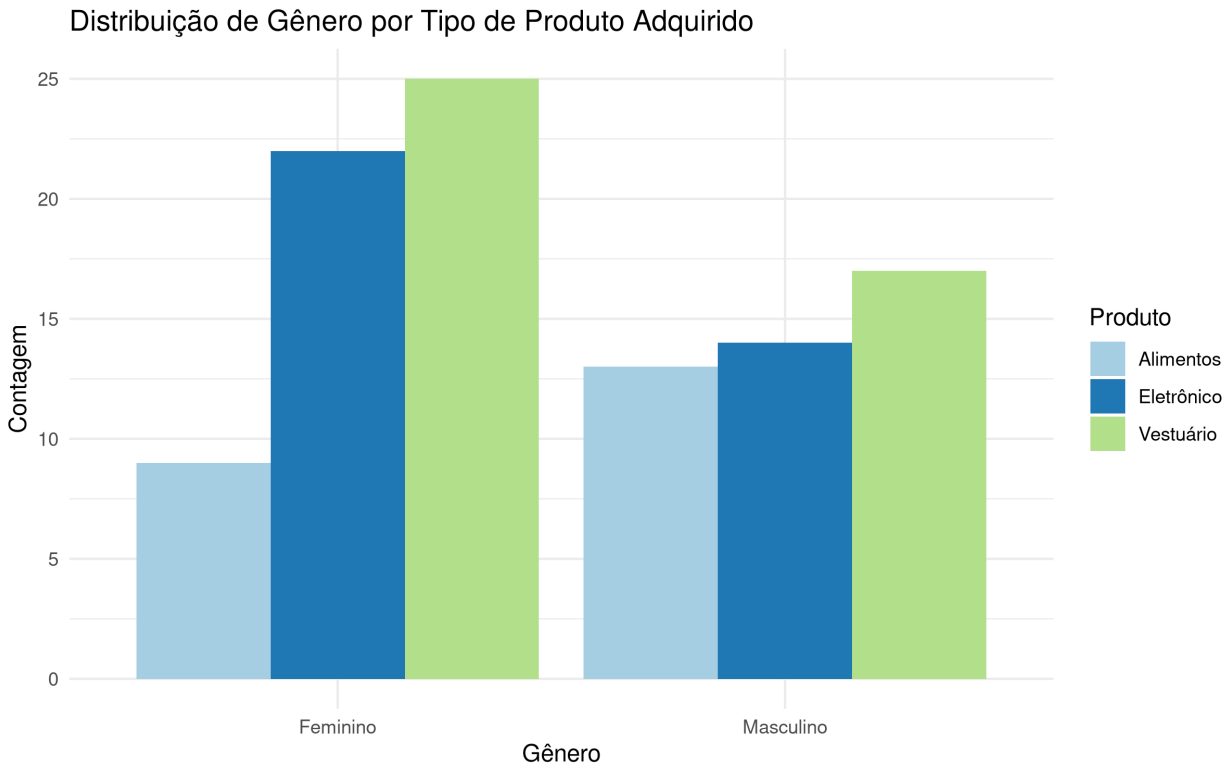

Figure 11: Gráfico de Barras Agrupadas e Empilhadas para Variáveis Qualitativas

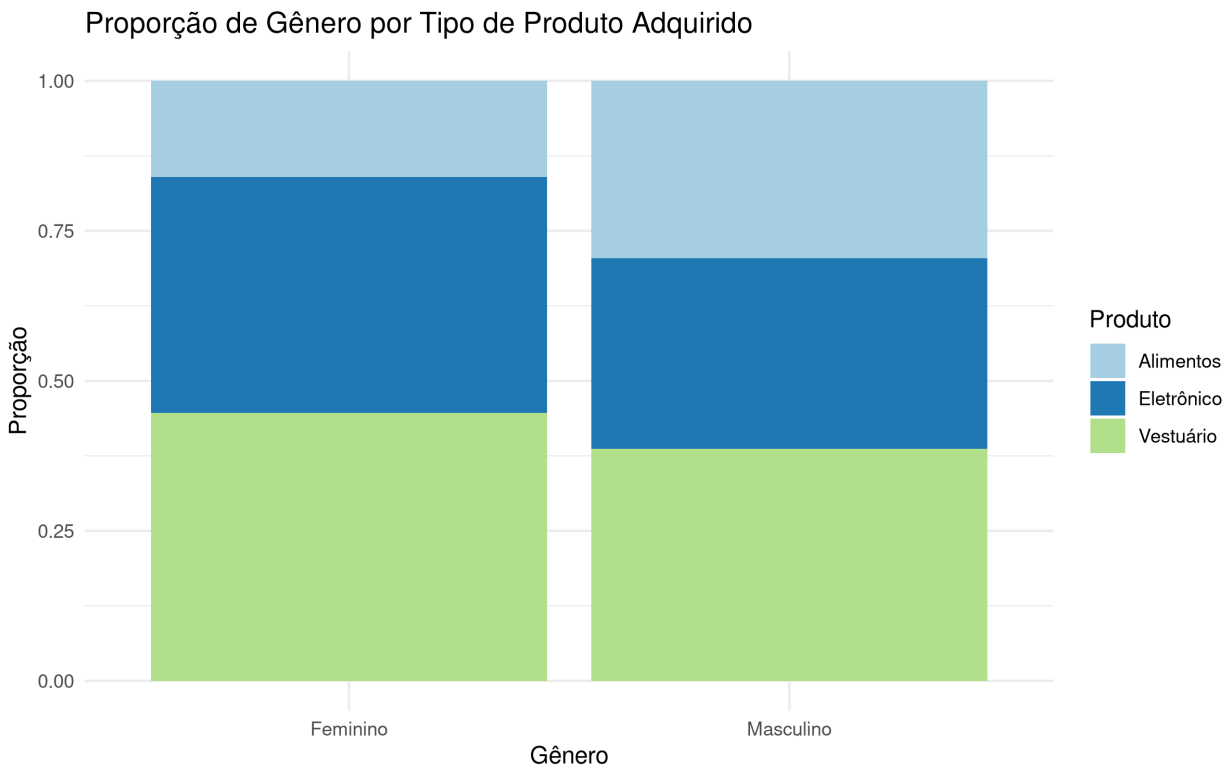

Figure 12: Gráfico de Barras Agrupadas e Empilhadas para Variáveis Qualitativas

Identificando Padrões e Anomalias

Durante a AED, a interpretação visual é fundamental.

- **Distribuições:**
  - **Simétricas (normal):** Histograma/densidade com forma de sino. Média $\approx$ Mediana $\approx$ Moda.
  - **Assimétricas à direita (positiva):** Cauda longa à direita. Moda $<$ Mediana $<$ Média.
  - **Assimétricas à esquerda (negativa):** Cauda longa à esquerda. Média $<$ Mediana $<$ Moda.
  - **Multimodais:** Múltiplos picos, indicando subgrupos.
- **Outliers (Valores Atípicos):** Pontos de dados que se desviam significativamente da maioria das observações. Visíveis como pontos isolados em Box Plots, Scatter Plots, ou em caudas muito longas em Histograms.
- **Tendências:** Direção geral em Scatter Plots (positiva, negativa, sem relação).
- **Agrupamentos (Clusters):** Aglomerados de pontos em Scatter Plots, sugerindo subgrupos naturais.
- **Variações e Dispersão:** A largura de Histogramas, Box Plots e Violin Plots indica o quão espalhados os dados estão.

Relação com Outros Conceitos

A AED atua como uma ponte entre a estatística descritiva (medidas de tendência central e dispersão) e a estatística inferencial. As medidas descritivas fornecem os números que a AED visualiza e interpreta. Os padrões e anomalias descobertos na AED frequentemente levam à formulação de hipóteses que serão testadas usando inferência estatística (próximas aulas).

Verificação de Aprendizagem

Para esta atividade, utilizaremos o famoso dataset `Iris`. Este dataset contém medidas de sépalas e pétalas (comprimento e largura) de três espécies de flores Iris: `setosa`, `versicolor` e `virginica`.

Carregue o dataset Iris em Python/R e execute as seguintes análises:

1. **Visão Geral do Dataset:**
   a) Carregue o dataset Iris.
   b) Exiba as primeiras 5 linhas.
   c) Verifique os tipos de dados e informações básicas (número de linhas, colunas, valores ausentes).
2. **Análise Univariada (Comprimento da Sépala):**
   a) Crie um **histograma** para a variável `sepal_length`.
   b) Crie um **box plot** para a variável `sepal_length`.
   c) Interprete a forma da distribuição (simétrica, assimétrica), o centro e a dispersão do `sepal_length` com base nos gráficos. Há outliers aparentes?
3. **Análise Bivariada (Comprimento da Sépala vs. Comprimento da Pétala por Espécie):**

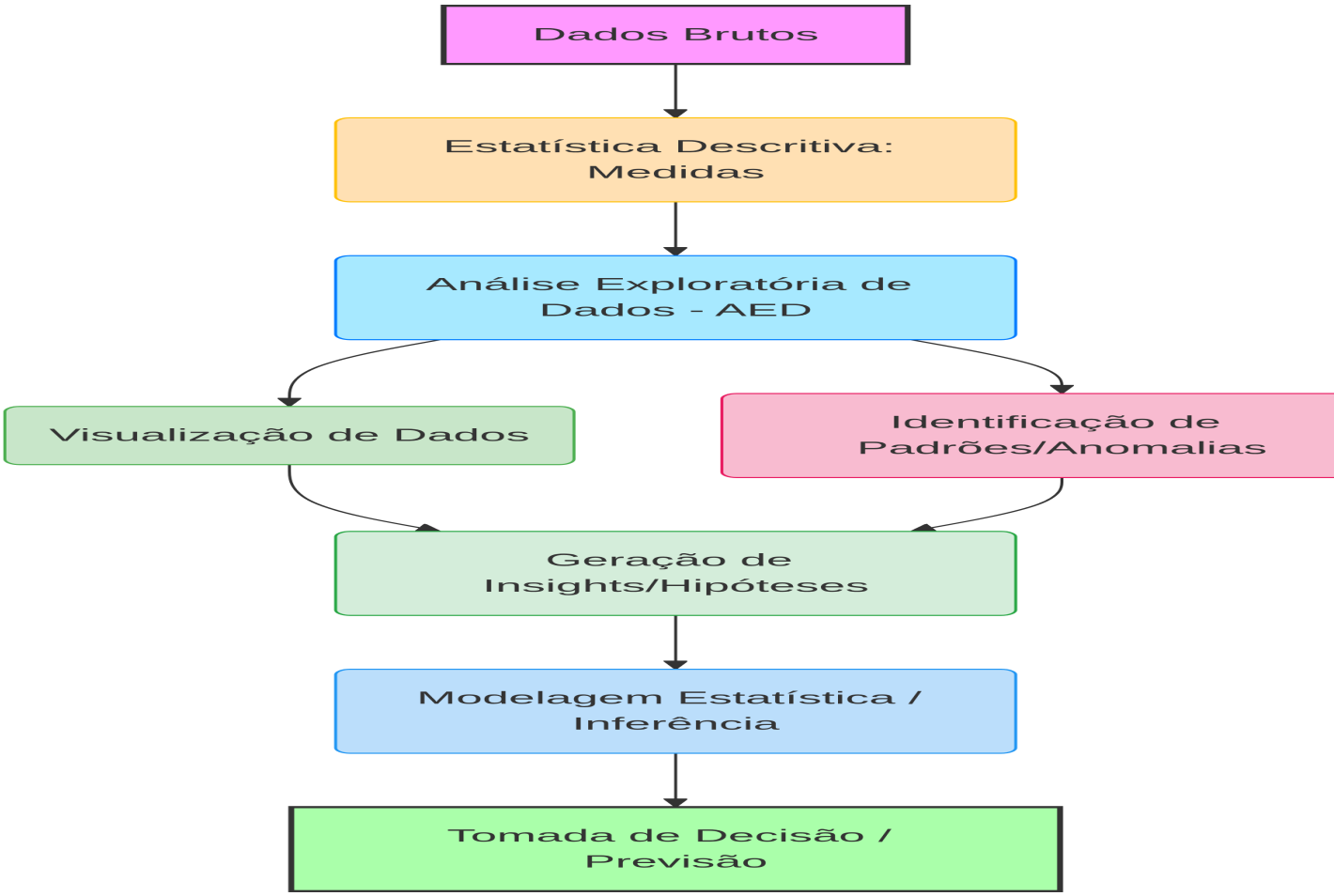

Figure 13: Diagrama de Relações entre AED e Outros Conceitos Estatísticos

a) Crie um **gráfico de dispersão** que mostre a relação entre `sepal_length` (eixo X) e `petal_length` (eixo Y).
b) Colora os pontos por `species` (espécie) para distinguir as três espécies.
c) Interprete o gráfico: Há uma correlação entre o comprimento da sépala e da pétala? Como as diferentes espécies se agrupam ou se separam neste gráfico?
d) Crie um **box plot** do `sepal_length` para cada `species`. Interprete as diferenças de comprimento da sépala entre as espécies.

4. **Matriz de Correlação:**

a) Calcule e visualize a **matriz de correlação** entre todas as variáveis numéricas do dataset Iris usando um heatmap.
b) Identifique os pares de variáveis com a correlação linear mais forte (positiva e negativa, se houver).

Referências Bibliográficas

BUSSAB, Luiz O. de M.; MORETTIN, Pedro A. **Estatística Básica**. 9. ed. São Paulo: Saraiva, 2017.