Representação Gráfica de Dados de Pesquisa

Criar Gráficos Eficazes para Comunicar Insights de Dados

Márcio Nicolau

2025-11-12

Table of contents

Introdução e Objetivos **2**
Objetivos de Aprendizagem . . . 2

A Importância da Representação Gráfica na Comunicação **3**
O Processo de Visualização de Dados . . . 3

Princípios de Design de Gráficos Eficazes **3**

Tipos de Gráficos e Suas Aplicações **5**
1. Distribuição de uma Única Variável . . . 5
Gráfico de Barras (para Variáveis Categóricas/Qualitativas) . . . 5
Python . . . 5
R . . . 5
Histograma (para Variáveis Quantitativas) . . . 6
Python . . . 6
R . . . 6
Box Plot (para Variáveis Quantitativas, Comparação de Grupos) . . . 7
Python . . . 7
R . . . 8
2. Relação entre Duas Variáveis . . . 8
Gráfico de Dispersão (Scatter Plot - Quantitativa vs. Quantitativa) . . . 8
Python . . . 8
R . . . 9
Gráfico de Linha (para Séries Temporais - Quantitativa vs. Temporal) . . . 9
Python . . . 10
R . . . 10
Gráfico de Barras Agrupadas/Empilhadas (Qualitativa vs. Qualitativa) . . . 11
Python . . . 11
R . . . 12
3. Gráficos para Múltiplas Variáveis . . . 13

Heatmap de Correlação (para Múltiplas Variáveis Quantitativas) . . . 13
Python . . . 13
R . . . 13

Boas Práticas e Erros Comuns na Visualização **14**
Boas Práticas . . . 14
Erros Comuns a Evitar . . . 15

Relação com Outros Conceitos **15**

Verificação de Aprendizagem **15**

Referências Bibliográficas **17**

List of Figures

1 Fluxo do Processo de Criação de Visualizações Eficazes . . . 4
2 Relação da Representação Gráfica com Outros Conceitos Estatísticos . . . 16

Introdução e Objetivos

Nas aulas anteriores, mergulhamos em todo o ciclo de análise de dados: desde a coleta e a garantia da qualidade, passando pela estatística descritiva, inferência (estimação e testes de hipóteses) e a interpretação. Agora, chegamos à fase de apresentar os resultados de forma clara, concisa e impactante. A **representação gráfica de dados de pesquisa** é a ponte entre a análise complexa e a compreensão intuitiva.

Um gráfico bem elaborado pode revelar padrões, anomalias e tendências em segundos, enquanto uma tabela de números brutos pode exigir minutos de estudo e ainda assim obscurecer insights importantes. A visualização de dados não é apenas uma habilidade técnica; é uma forma de arte e ciência que nos permite contar a história dos nossos dados de maneira convincente.

Nesta aula, exploraremos os princípios de um bom design gráfico, os tipos de gráficos mais eficazes para diferentes cenários de dados e como utilizá-los para comunicar seus achados de forma poderosa e ética.

Objetivos de Aprendizagem

Ao final desta aula, você será capaz de:

- Compreender a importância e o impacto da visualização de dados na comunicação de resultados de pesquisa.
- Aplicar princípios de design gráfico para criar visualizações claras, precisas e eficazes.
- Selecionar o tipo de gráfico mais apropriado para diferentes tipos de dados e relações.
- Criar e personalizar gráficos univariados e bivariados usando `matplotlib`/`seaborn` em Python e `ggplot2` em R.
- Identificar erros comuns e armadilhas na visualização de dados.
- Interpretar e comunicar insights de dados através de visualizações.

A Importância da Representação Gráfica na Comunicação

No mundo da análise de dados, os resultados só têm valor se puderem ser compreendidos e utilizados. Gráficos são ferramentas poderosas para:

- **Clarificar Informações Complexas:** Transformar tabelas extensas de números em representações visuais fáceis de assimilar.
- **Revelar Padrões e Tendências:** O olho humano é excelente em detectar padrões visuais que podem ser invisíveis em dados brutos ou resumos numéricos.
- **Identificar Anomalias e Outliers:** Pontos de dados incomuns se destacam facilmente em muitos tipos de gráficos.
- **Suportar Argumentos e Decisões:** Um bom gráfico fornece evidências visuais convincentes para as conclusões da sua pesquisa.
- **Engajar a Audiência:** Visualizações atraentes e bem projetadas mantêm o interesse e facilitam o aprendizado.

O Processo de Visualização de Dados

Princípios de Design de Gráficos Eficazes

Criar um bom gráfico vai além de escolher o tipo certo; é sobre design para clareza e impacto.

1. **Clareza e Simplicidade:**
    - **Evite o "Chartjunk":** Informações desnecessárias (cores vibrantes demais, texturas, sombras 3D) que distraem da mensagem.
    - **Priorize a Mensagem:** O objetivo do gráfico deve ser evidente.
    - **Menos é Mais:** Reduza a tinta não-dado (Edward Tufte).
2. **Precisão e Integridade:**
    - **Não Distorça os Dados:** Eixos devem começar em zero quando apropriado (especialmente para comparações de magnitude). Evite escalas enganosas.
    - **Seja Honesto:** Não manipule as visualizações para apoiar uma agenda.
3. **Rótulos e Títulos:**
    - **Título Claro:** Descreve o conteúdo do gráfico de forma concisa.
    - **Rótulos dos Eixos:** Identificam claramente o que cada eixo representa, incluindo unidades de medida.
    - **Legenda:** Explica o significado de cores, formas ou tamanhos usados no gráfico.
    - **Fonte dos Dados:** Sempre inclua a fonte, se possível.
4. **Uso de Cores:**
    - **Significado:** Use cores de forma consistente e com propósito (ex: cores diferentes para categorias distintas, gradientes para valores quantitativos).
    - **Acessibilidade:** Considere daltônicos (use paletas amigáveis para daltonismo).
    - **Contraste:** Garanta que o texto e os elementos sejam legíveis.

Figure 1: Fluxo do Processo de Criação de Visualizações Eficazes

5. **Escolha do Tipo de Gráfico:** O tipo de gráfico deve ser adequado ao tipo de dados e à relação que você deseja mostrar.

Tipos de Gráficos e Suas Aplicações

Revistaremos alguns gráficos da AED, agora com foco na eficácia da comunicação.

1. Distribuição de uma Única Variável

Gráfico de Barras (para Variáveis Categóricas/Qualitativas)

Mostra a frequência ou proporção de cada categoria. Ideal para comparar a magnitude entre categorias discretas.

- **Objetivo:** Comparar volumes ou contagens entre categorias.
- **Exemplo:** Distribuição de sistemas operacionais em smartphones.

Python

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

data = {'SO': ['Android', 'iOS', 'Android', 'iOS', 'Android', 'Outro', 'iOS', 'Android', 'Android', 'iOS
df = pd.DataFrame(data)

plt.figure(figsize=(8, 5))
sns.countplot(x='SO', data=df, palette='pastel', order=df['SO'].value_counts().index)
plt.title('Distribuição de Sistemas Operacionais em Smartphones', fontsize=14)
plt.xlabel('Sistema Operacional', fontsize=12)
plt.ylabel('Número de Usuários', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

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

df <- data.frame(
  SO = c('Android', 'iOS', 'Android', 'iOS', 'Android', 'Outro', 'iOS', 'Android', 'Android', 'iOS')
)

ggplot(df, aes(x = SO, fill = SO)) +
  geom_bar(stat = "count") +
```

```
  labs(title = 'Distribuição de Sistemas Operacionais em Smartphones',
       x = 'Sistema Operacional',
       y = 'Número de Usuários') +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold")) +
  scale_fill_brewer(palette = "Set3") # Melhor para categorias
```

Histograma (para Variáveis Quantitativas)

Exibe a distribuição de uma variável numérica, mostrando a frequência de valores dentro de intervalos (bins). Ótimo para entender a forma da distribuição, centralidade e dispersão.

- **Objetivo:** Mostrar a forma, picos e espalhamento de dados numéricos.
- **Exemplo:** Distribuição de idades de uma população.

Python

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

np.random.seed(42)
idades = np.random.normal(loc=35, scale=10, size=500)
idades = idades[(idades >= 18) & (idades <= 80)]

plt.figure(figsize=(10, 6))
sns.histplot(idades, bins=15, kde=True, color='skyblue', edgecolor='black')
plt.title('Distribuição de Idades em uma População Amostral', fontsize=14)
plt.xlabel('Idade (anos)', fontsize=12)
plt.ylabel('Frequência', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

```
library(ggplot2)

set.seed(42)
idades <- rnorm(500, mean = 35, sd = 10)
idades <- idades[idades >= 18 & idades <= 80]
df_idades <- data.frame(Idade = idades)

ggplot(df_idades, aes(x = Idade)) +
```

```r
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black", alpha = 0.8) +
  geom_density(aes(y = after_stat(density * 5 * nrow(df_idades))), color = "darkblue", size = 1) + # Es
  labs(title = 'Distribuição de Idades em uma População Amostral',
       x = 'Idade (anos)',
       y = 'Frequência') +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"))
```

Box Plot (para Variáveis Quantitativas, Comparação de Grupos)

Fornece um resumo de 5 números (mínimo, Q1, mediana, Q3, máximo) e identifica outliers. Excelente para comparar a distribuição de uma variável numérica entre diferentes categorias.

- **Objetivo:** Comparar centralidade, dispersão e identificar outliers entre grupos.
- **Exemplo:** Salários por nível de escolaridade.

Python

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

data_salarios = {
    'Escolaridade': np.random.choice(['Ensino Médio', 'Graduação', 'Pós-Graduação'], size=100),
    'Salario': np.random.normal(loc=4000, scale=800, size=100)
}
df_salarios = pd.DataFrame(data_salarios)
df_salarios.loc[df_salarios['Escolaridade'] == 'Graduação', 'Salario'] = np.random.normal(loc=6000, sca
df_salarios.loc[df_salarios['Escolaridade'] == 'Pós-Graduação', 'Salario'] = np.random.normal(loc=9000,
df_salarios['Salario'] = np.maximum(df_salarios['Salario'], 1500).astype(int) # Salario mínimo
# Adicionar um outlier manual para exemplo
df_salarios.loc[0, 'Salario'] = 25000

plt.figure(figsize=(10, 6))
sns.boxplot(x='Escolaridade', y='Salario', data=df_salarios,
            order=['Ensino Médio', 'Graduação', 'Pós-Graduação'], palette='viridis')
plt.title('Distribuição de Salários por Nível de Escolaridade', fontsize=14)
plt.xlabel('Nível de Escolaridade', fontsize=12)
plt.ylabel('Salário (R$)', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

R

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

set.seed(42)
df_salarios <- data.frame(
  Escolaridade = factor(sample(c('Ensino Médio', 'Graduação', 'Pós-Graduação'), size = 100, replace = TR
                        levels = c('Ensino Médio', 'Graduação', 'Pós-Graduação')),
  Salario = rnorm(100, mean = 4000, sd = 800)
)
df_salarios <- df_salarios %>%
  mutate(Salario = case_when(
    Escolaridade == 'Graduação' ~ rnorm(n(), mean = 6000, sd = 1200),
    Escolaridade == 'Pós-Graduação' ~ rnorm(n(), mean = 9000, sd = 1800),
    TRUE ~ Salario
  ))
df_salarios$Salario <- pmax(df_salarios$Salario, 1500) %>% round()
# Adicionar um outlier manual para exemplo
df_salarios$Salario <- 25000

ggplot(df_salarios, aes(x = Escolaridade, y = Salario, fill = Escolaridade)) +
  geom_boxplot() +
  labs(title = 'Distribuição de Salários por Nível de Escolaridade',
       x = 'Nível de Escolaridade',
       y = 'Salário (R$)') +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.position = "none") +
  scale_fill_viridis_d()
```

2. Relação entre Duas Variáveis

Gráfico de Dispersão (Scatter Plot - Quantitativa vs. Quantitativa)

Mostra a relação entre duas variáveis numéricas. Essencial para identificar padrões, tendências (correlação) e agrupamentos.

- **Objetivo:** Visualizar a correlação entre duas variáveis.
- **Exemplo:** Relação entre horas de exercício e pressão arterial.

Python

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

```python
import seaborn as sns
import pandas as pd

np.random.seed(42)
horas_exercicio = np.random.uniform(1, 10, 100)
pressao_arterial = 140 - horas_exercicio * 3 + np.random.normal(0, 5, 100)
df_saude = pd.DataFrame({'Horas_Exercicio': horas_exercicio, 'Pressao_Arterial': pressao_arterial})

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Horas_Exercicio', y='Pressao_Arterial', data=df_saude, hue='Pressao_Arterial', size=
plt.title('Relação entre Horas de Exercício Semanal e Pressão Arterial', fontsize=14)
plt.xlabel('Horas de Exercício por Semana', fontsize=12)
plt.ylabel('Pressão Arterial (mmHg)', fontsize=12)
plt.grid(linestyle='--', alpha=0.7)
plt.legend(title='Pressão Arterial', bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.tight_layout()
plt.show()
```

R

```r
library(ggplot2)

set.seed(42)
horas_exercicio <- runif(100, 1, 10)
pressao_arterial <- 140 - horas_exercicio * 3 + rnorm(100, 0, 5)
df_saude <- data.frame(Horas_Exercicio = horas_exercicio, Pressao_Arterial = pressao_arterial)

ggplot(df_saude, aes(x = Horas_Exercicio, y = Pressao_Arterial, color = Pressao_Arterial, size = Horas_
  geom_point(alpha = 0.7) +
  labs(title = 'Relação entre Horas de Exercício Semanal e Pressão Arterial',
       x = 'Horas de Exercício por Semana',
       y = 'Pressão Arterial (mmHg)') +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold")) +
  scale_color_viridis_c() +
  guides(size = "none") # Esconde a legenda de tamanho se não for essencial
```

Gráfico de Linha (para Séries Temporais - Quantitativa vs. Temporal)

Mostra a evolução de uma variável quantitativa ao longo do tempo. Essencial para identificar tendências, sazonalidades e ciclos.

- **Objetivo:** Visualizar mudanças ao longo do tempo.
- **Exemplo:** Vendas mensais ao longo de um ano.

Python

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

data_vendas = {
    'Mes': pd.to_datetime(['2024-01-01', '2024-02-01', '2024-03-01', '2024-04-01', '2024-05-01', '2024-
                           '2024-07-01', '2024-08-01', '2024-09-01', '2024-10-01', '2024-11-01', '2024-
    'Vendas':
}
df_vendas_mensais = pd.DataFrame(data_vendas)

plt.figure(figsize=(12, 6))
sns.lineplot(x='Mes', y='Vendas', data=df_vendas_mensais, marker='o', color='darkorange', linewidth=2)
plt.title('Vendas Mensais ao Longo de 2024', fontsize=14)
plt.xlabel('Mês', fontsize=12)
plt.ylabel('Vendas (em milhares de R$)', fontsize=12)
plt.grid(linestyle='--', alpha=0.7)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

R

```r
library(ggplot2)
library(lubridate) # Para trabalhar com datas

df_vendas_mensais <- data.frame(
  Mes = ymd(c('2024-01-01', '2024-02-01', '2024-03-01', '2024-04-01', '2024-05-01', '2024-06-01',
              '2024-07-01', '2024-08-01', '2024-09-01', '2024-10-01', '2024-11-01', '2024-12-01')),
  Vendas = c(150, 160, 170, 180, 190, 200, 220, 210, 200, 190, 180, 170)
)

ggplot(df_vendas_mensais, aes(x = Mes, y = Vendas)) +
  geom_line(color = "darkorange", size = 1.2) +
  geom_point(color = "darkorange", size = 3) +
  labs(title = 'Vendas Mensais ao Longo de 2024',
       x = 'Mês',
       y = 'Vendas (em milhares de R$)') +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"),
        axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y")
```

Gráfico de Barras Agrupadas/Empilhadas (Qualitativa vs. Qualitativa)

Compara as frequências ou proporções de uma categoria em relação a outra. Útil para mostrar a composição ou a distribuição relativa de categorias.

- **Objetivo:** Comparar a distribuição de uma variável categórica entre diferentes grupos de outra variável categórica.
- **Exemplo:** Preferência por tipo de café por gênero.

Python

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

np.random.seed(42)
data_cafe = {
    'Genero': np.random.choice(['Masculino', 'Feminino'], size=200, p=[0.5, 0.5]),
    'Tipo_Cafe': np.random.choice(['Expresso', 'Cappuccino', 'Latte', 'Preto'], size=200)
}
df_cafe = pd.DataFrame(data_cafe)

# Ajuste para mostrar algumas diferenças
df_cafe.loc[df_cafe['Genero'] == 'Feminino', 'Tipo_Cafe'] = np.random.choice(['Cappuccino', 'Latte', 'E
df_cafe.loc[df_cafe['Genero'] == 'Masculino', 'Tipo_Cafe'] = np.random.choice(['Expresso', 'Preto', 'Ca


# Gráfico de barras agrupadas
plt.figure(figsize=(10, 6))
sns.countplot(x='Genero', hue='Tipo_Cafe', data=df_cafe, palette='Set2')
plt.title('Preferência por Tipo de Café por Gênero', fontsize=14)
plt.xlabel('Gênero', fontsize=12)
plt.ylabel('Contagem', fontsize=12)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(title='Tipo de Café', bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.tight_layout()
plt.show()

# Gráfico de barras empilhadas (proporções)
plt.figure(figsize=(10, 6))
df_cafe_prop = df_cafe.groupby('Genero')['Tipo_Cafe'].value_counts(normalize=True).unstack().loc[['Femi
df_cafe_prop.plot(kind='bar', stacked=True, colormap='Set2')
plt.title('Proporção de Preferência por Tipo de Café por Gênero', fontsize=14)
plt.xlabel('Gênero', fontsize=12)
plt.ylabel('Proporção', fontsize=12)
```

```
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(title='Tipo de Café', bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.tight_layout()
plt.show()
```

R

```
library(ggplot2)
library(dplyr)

set.seed(42)
df_cafe <- data.frame(
  Genero = factor(sample(c('Masculino', 'Feminino'), size = 200, replace = TRUE)),
  Tipo_Cafe = factor(sample(c('Expresso', 'Cappuccino', 'Latte', 'Preto'), size = 200, replace = TRUE))
)

# Ajuste para mostrar algumas diferenças
df_cafe <- df_cafe %>%
  mutate(Tipo_Cafe = case_when(
    Genero == 'Feminino' ~ sample(c('Cappuccino', 'Latte', 'Expresso'), n(), replace = TRUE, prob = c(0
    Genero == 'Masculino' ~ sample(c('Expresso', 'Preto', 'Cappuccino'), n(), replace = TRUE, prob = c(
    TRUE ~ Tipo_Cafe
  ))

# Gráfico de barras agrupadas
ggplot(df_cafe, aes(x = Genero, fill = Tipo_Cafe)) +
  geom_bar(position = "dodge") +
  labs(title = 'Preferência por Tipo de Café por Gênero',
       x = 'Gênero',
       y = 'Contagem') +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.position = "right") +
  scale_fill_brewer(palette = "Set2")

# Gráfico de barras empilhadas (proporções)
ggplot(df_cafe, aes(x = Genero, fill = Tipo_Cafe)) +
  geom_bar(position = "fill") + # "fill" para proporções empilhadas
  labs(title = 'Proporção de Preferência por Tipo de Café por Gênero',
       x = 'Gênero',
       y = 'Proporção') +
  theme_minimal(base_size = 14) +
```

```
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.position = "right") +
  scale_fill_brewer(palette = "Set2")
```

3. Gráficos para Múltiplas Variáveis

Heatmap de Correlação (para Múltiplas Variáveis Quantitativas)

Visualiza a força e a direção das relações lineares entre todos os pares de variáveis numéricas.

- **Objetivo:** Entender rapidamente o padrão de correlações em um dataset.
- **Exemplo:** Correlação entre características de imóveis (preço, tamanho, número de quartos).

Python

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

# Criando um dataset de exemplo com correlações variadas
np.random.seed(42)
data_multi = pd.DataFrame({
    'Var1': np.random.rand(50),
    'Var2': np.random.rand(50) * 0.7 + np.random.rand(50) * 0.3, # Correlação fraca
    'Var3': np.random.rand(50) * 0.8 + np.random.rand(50) * 0.2 + np.random.rand(50), # Correlação mode
    'Var4': np.random.rand(50) * -0.9 + np.random.rand(50) * 0.1 # Correlação forte negativa
})
data_multi['Var2'] = data_multi['Var1'] * 0.6 + np.random.normal(0, 0.1, 50) # Ajustar para correlação
data_multi['Var3'] = data_multi['Var1'] * 0.05 + data_multi['Var2'] * 0.8 + np.random.normal(0, 0.05, 5
data_multi['Var4'] = data_multi['Var1'] * -0.7 + np.random.normal(0, 0.1, 50)

plt.figure(figsize=(10, 8))
sns.heatmap(data_multi.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.title('Matriz de Correlação de Variáveis', fontsize=14)
plt.xticks(rotation=45, ha='right')
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

R

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

```r
library(dplyr)

set.seed(42)
df_multi <- data.frame(
  Var1 = runif(50),
  Var2 = runif(50) * 0.6 + runif(50) * 0.4, # Correlação fraca
  Var3 = runif(50) * 0.8 + runif(50) * 0.2 + runif(50), # Correlação moderada
  Var4 = runif(50) * -0.9 + runif(50) * 0.1 # Correlação forte negativa
)
# Ajustar para correlação mais controlada
df_multi <- df_multi %>%
  mutate(
    Var2 = Var1 * 0.6 + rnorm(n(), 0, 0.1),
    Var3 = Var1 * 0.05 + Var2 * 0.8 + rnorm(n(), 0, 0.05),
    Var4 = Var1 * -0.7 + rnorm(n(), 0, 0.1)
  )

cor_matrix <- cor(df_multi)
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(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1, face = "bold"),
        axis.text.y = element_text(face = "bold"),
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  coord_fixed() +
  geom_text(aes(Var1, Var2, label = round(value, 2)), color = "black", size = 4) +
  labs(title = "Matriz de Correlação de Variáveis")
```

Boas Práticas e Erros Comuns na Visualização

Boas Práticas

- **Simplicidade:** Mantenha o gráfico o mais simples possível, removendo elementos desnecessários.
- **Consistência:** Use cores, fontes e estilos de forma consistente em todos os gráficos.
- **Foco na Mensagem:** Cada gráfico deve ter uma história clara para contar.
- **Acessibilidade:** Garanta que os gráficos sejam compreensíveis para pessoas com deficiência visual (ex: daltonismo). Use texto alternativo.
- **Interatividade (se aplicável):** Para dashboards e aplicações web, a interatividade pode enriquecer a exploração de dados.

Erros Comuns a Evitar

- **Gráficos 3D Desnecessários:** Raramente adicionam valor, frequentemente obscurecem os dados (ex: gráficos de pizza 3D).
- **Eixos Truncados:** Começar um eixo Y em um valor diferente de zero pode exagerar diferenças.
- **Escalas Inapropriadas:** Escolher uma escala que comprime ou expande demais os dados, levando a interpretações errôneas.
- **Sobrecarga de Informação:** Muitos elementos, cores ou categorias em um único gráfico tornam-no ilegível.
- **Escolha Errada do Gráfico:** Usar um gráfico de linhas para dados categóricos não sequenciais, por exemplo.
- **Cores Mal Utilizadas:** Exagerar nas cores, usar cores conflitantes ou não acessíveis.

Relação com Outros Conceitos

A representação gráfica é o produto final de um processo analítico bem-sucedido e serve como uma ferramenta de diagnóstico e comunicação para todas as etapas:

- **Estatística Descritiva e AED:** Os gráficos são a manifestação visual dessas etapas, permitindo a exploração e o resumo dos dados.
- **Coleta e Qualidade dos Dados:** Gráficos podem expor problemas de qualidade (outliers, valores ausentes).
- **Estimação e Testes de Hipóteses:** Intervalos de confiança podem ser visualizados (ex: barras de erro), e a significância pode ser destacada.
- **Análise e Interpretação de Dados de Pesquisa:** A visualização eficaz é a chave para comunicar as conclusões e recomendações de forma impactante.

Verificação de Aprendizagem

Para esta atividade, utilizaremos o dataset `tips` (gorjetas), disponível no pacote `seaborn` (Python) e frequentemente usado em exemplos `ggplot2` (R). Este dataset contém informações sobre o valor da conta, gorjeta, sexo do pagador, fumante (sim/não), dia da semana, hora do dia e tamanho da mesa.

Carregue o dataset `tips` em Python/R e execute as seguintes visualizações, interpretando-as:

1. **Distribuição Univariada:**
   a) Crie um **histograma** para a variável `total_bill` (valor total da conta).
   b) Crie um **gráfico de barras** para a variável `day` (dia da semana).
2. **Relações Bivariadas:**
   a) Crie um **gráfico de dispersão** mostrando `total_bill` (eixo X) vs. `tip` (gorjeta, eixo Y). Adicione `sex` (sexo) como cor dos pontos e `smoker` (fumante) como estilo dos marcadores.
   b) Crie um **box plot** para `tip` (gorjeta) por `day` (dia da semana).
3. **Gráfico de Proporções:**

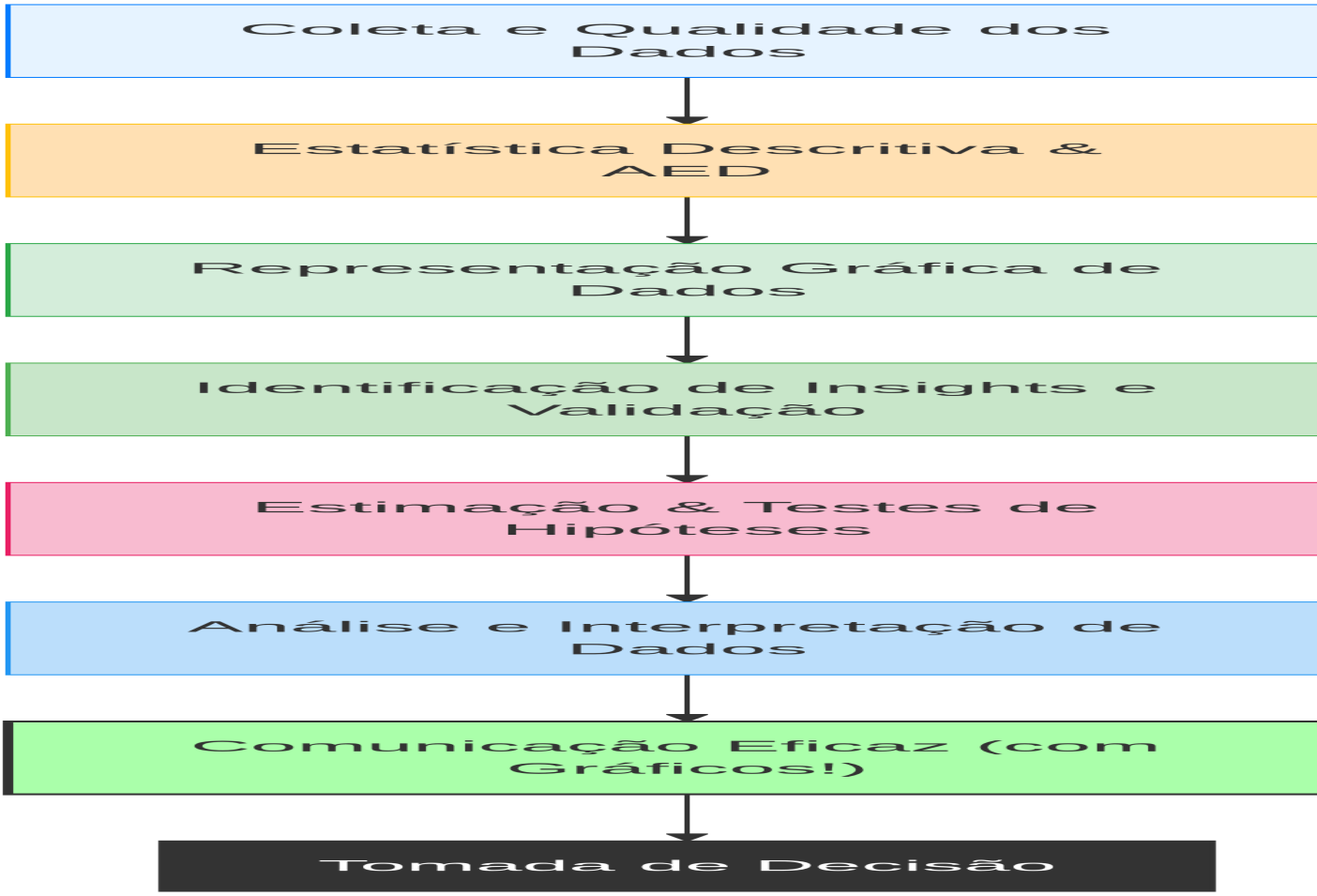

Figure 2: Relação da Representação Gráfica com Outros Conceitos Estatísticos

a) Crie um **gráfico de barras empilhadas** mostrando a proporção de `smoker` (fumante) por `day` (dia da semana).

Interprete cada gráfico para identificar padrões, tendências ou anomalias.

Referências Bibliográficas