



Visualização de Dados

Uma exploração abrangente das técnicas e ferramentas modernas para criar visualizações de dados impactantes e eficazes usando R e ggplot2.

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br
<https://eic.cefet-rj.br/~eogasawara>

Plotando Gráficos com R

Funções Básicas do R

O R vem com funções de plotagem básicas que são fáceis de operar para gráficos simples, mas apresentam limitações significativas quando se trata de visualizações mais complexas e personalizadas.

ggplot2: Uma Revolução em Gráficos

O ggplot2 é um sistema para criar gráficos de forma declarativa, baseado na Grammar of Graphics. Ele oferece flexibilidade excepcional e permite construir visualizações sofisticadas de maneira intuitiva.

Recursos de Aprendizado

O livro "R for Data Science" foi projetado para fornecer uma introdução abrangente ao ggplot2, com capítulos dedicados à visualização de dados e comunicação gráfica.

- ❑ A maneira mais fácil de obter o ggplot2 é instalar todo o tidyverse:
`install.packages("tidyverse")
install.packages("ggplot2")`

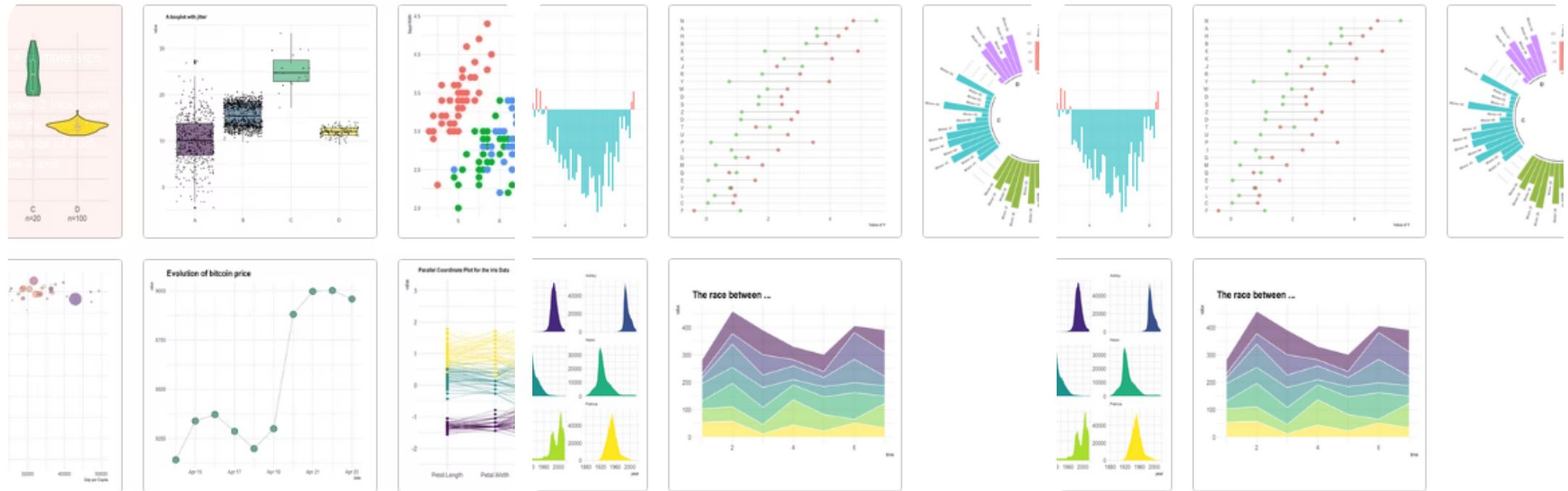
Referências:

Grammar of Graphics: [amazon.com/Grammar-Graphics](https://www.amazon.com/Grammar-Graphics)

R for Data Science: r4ds.had.co.nz

Exemplos de Gráficos com ggplot2

Esta galeria demonstra a ampla variedade de visualizações que você pode criar usando ggplot2. É um recurso excepcional para inspiração e aprendizado sobre como personalizar seus gráficos de acordo com suas necessidades específicas.



Explore a galeria completa: r-graph-gallery.com/ggplot2-package

Encapsulamento Fácil do ggplot2 com daltoolbox

O DAL Toolbox facilita a criação de gráficos encapsulando o ggplot2, permitindo um início mais simples enquanto você aprende a usar essa poderosa ferramenta de visualização.



Parâmetro X

O primeiro parâmetro é geralmente associado ao eixo x e representa a variável independente



Parâmetro Value

O segundo parâmetro está relacionado ao eixo y e representa os valores a serem plotados



Parâmetro Group

Às vezes, um terceiro parâmetro opcional define variáveis de agrupamento para categorização

- A maioria das funções requer um data.frame com esses parâmetros básicos. O DAL Toolbox é carregado usando a função library:
`library(daltoolbox)`
`library(ggplot2)`

Documentação completa: github.com/cefet-rj-dal/daltoolbox/graphics

Usando Cores Efetivamente em Gráficos

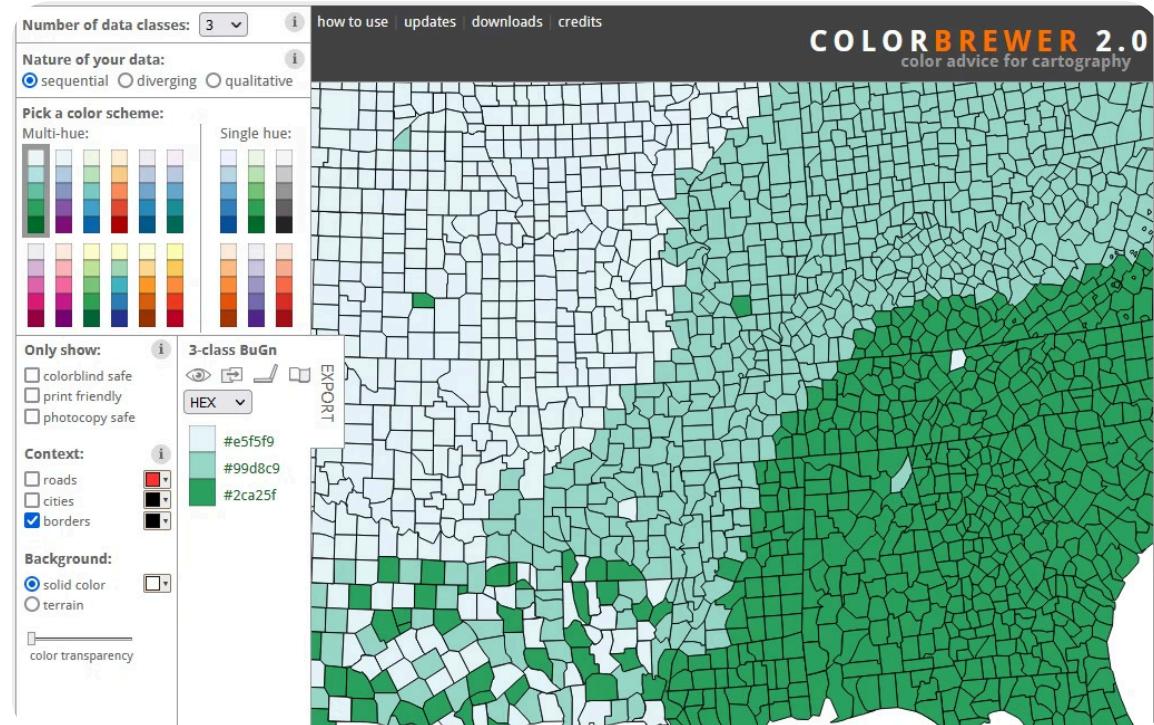
Princípios de Escolha de Cores

Escolhas claras de cores ajudam a transmitir significado e manter uma identidade visual consistente em suas visualizações.

Use ColorBrewer para selecionar paletas:

- **Sequential:** para dados ordenados que progredem de baixo para alto
- **Diverging:** para desvios de um ponto médio, destacando variações
- **Qualitative:** para dados categóricos sem ordem específica

Visite: colorbrewer2.org



Exemplo de Paletas usando ColorBrewer no R

O pacote RColorBrewer oferece acesso fácil às paletas ColorBrewer diretamente no R, permitindo que você crie visualizações profissionais com esquemas de cores cientificamente selecionados.

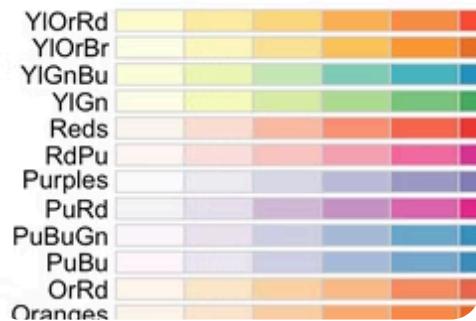
Instalação e uso básico:

```
#install.packages("RColorBrewer")
library(RColorBrewer)
colors <- brewer.pal(4, 'Set1')
```



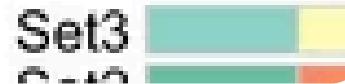
Sequential

Paletas sequenciais usam gradientes progressivos de cores para representar dados ordenados. Ideal para mostrar progressão de valores baixos para altos, como mapas de calor ou distribuições de densidade.



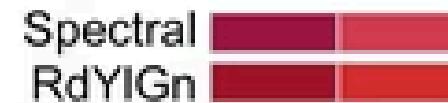
Qualitative

Paletas qualitativas usam cores distintas para diferenciar categorias sem implicar ordem. Perfeitas para gráficos de barras categóricos, gráficos de pizza ou qualquer visualização onde as categorias são nominais.



Diverging

Paletas divergentes enfatizam desvios de um ponto central, usando duas cores contrastantes. Úteis para mostrar dados que variam em torno de um valor médio ou crítico, como mudanças positivas e negativas.



Compreendendo o Formato de Entrada do DAL Toolbox

Dataset Iris: Um Exemplo Clássico

O conjunto de dados Iris é um dos conjuntos mais famosos em análise de dados e machine learning. Ele contém medições de flores de três espécies diferentes de íris.

```
head(iris, 3)
## Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1      5.1      3.5      1.4      0.2  setosa
## 2      4.9      3.0      1.4      0.2  setosa
## 3      4.7      3.2      1.3      0.2  setosa
```

Configurando Opções Gráficas

```
colors <- brewer.pal(4, 'Set1')
font <- theme(text = element_text(size=16))
```

Estrutura de Dados Simplificada

O DAL Toolbox simplifica o uso do ggplot2 para iniciantes através de um formato padronizado:

- **x:** variável independente (eixo horizontal)
- **value:** variável dependente (eixo vertical)
- **group:** opcional, para agrupar barras ou linhas

Carregado via: `library(daltoolbox)`

Essa abordagem incentiva visualizações limpas e rápidas, facilitando o aprendizado.

Gráfico de Dispersão (Scatter Plot)

Quando Usar

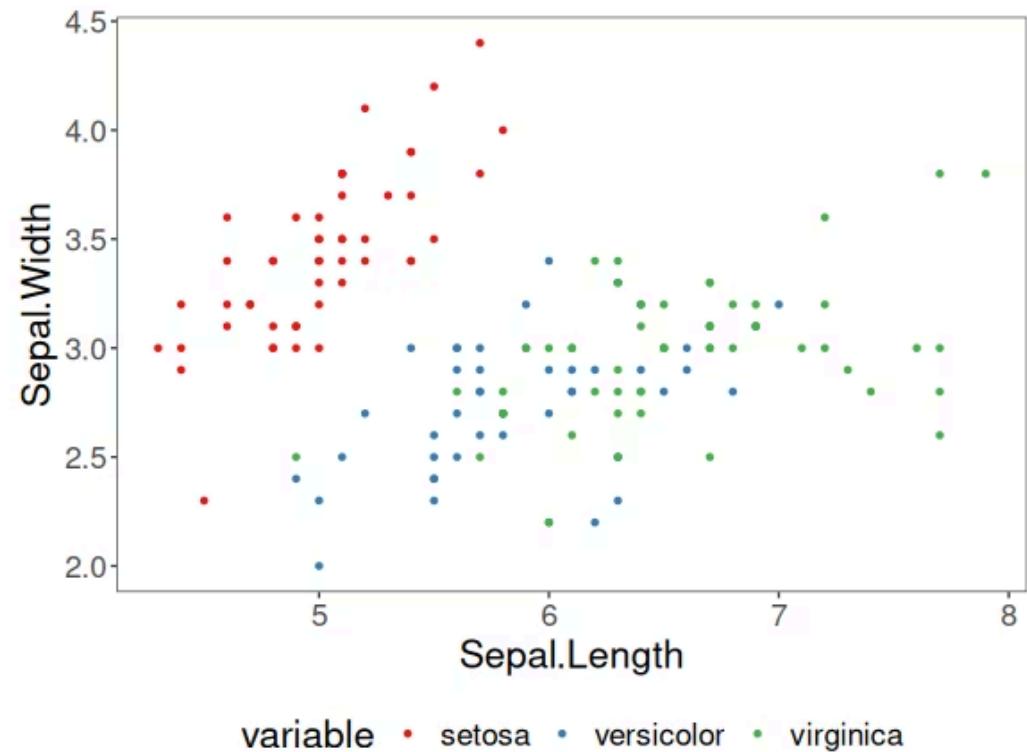
Gráficos de dispersão são usados para mostrar a relação entre duas variáveis numéricas, o que pode ajudar a identificar tendências, agrupamentos ou valores atípicos nos dados.

Código de Exemplo

```
library(dplyr)
data <- iris |>
  select(x = Sepal.Length,
         value = Sepal.Width,
         variable = Species)

grf <- plot_scatter(data,
                     label_x = "Sepal.Length",
                     label_y = "Sepal.Width",
                     colors=colors[1:3]) + font
plot(grf)
```

Documentação: [grf_scatter.md](#)



Este gráfico mostra a relação entre comprimento e largura das sépalas para três espécies diferentes de íris, revelando padrões distintos de agrupamento.

Gráfico de Barras (Bar Graph)

Características

Gráficos de barras são ideais para comparar quantidades entre categorias. Eles são simples, eficazes e amplamente utilizados em análise categórica.

Aplicações Comuns

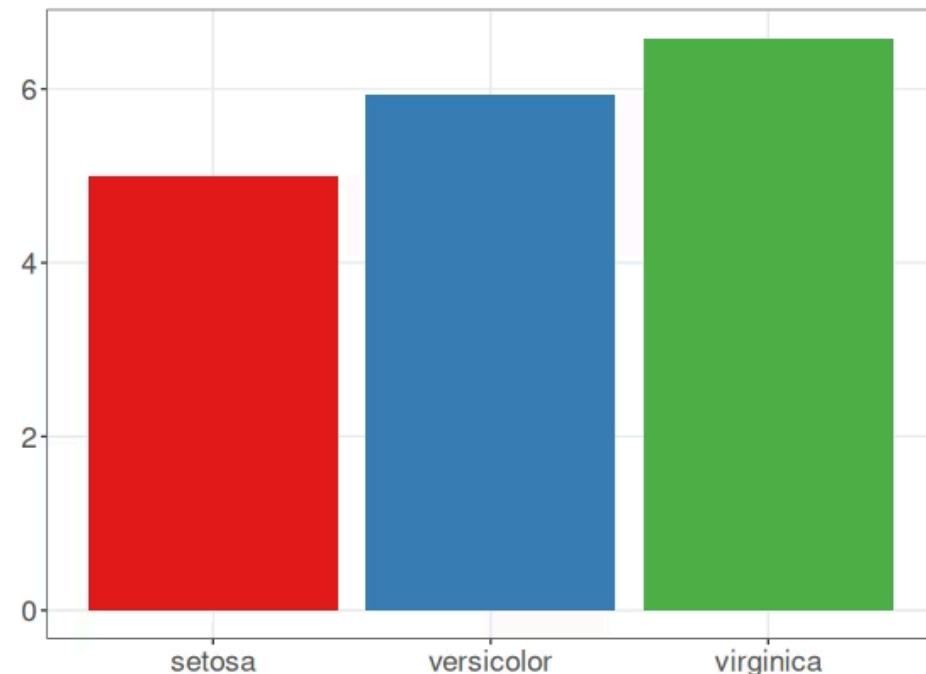
- Comparação de médias entre grupos
- Visualização de contagens por categoria
- Apresentação de resultados agregados

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(Sepal.Length=mean(Sepal.Length))

grf <- plot_bar(data, colors=colors[1:3]) + font
plot(grf)
```

Documentação: [grf_bar.md](#)



Este gráfico compara o comprimento médio das sépalas entre as três espécies de íris, facilitando a identificação de diferenças significativas.

Gráfico Lollipop

Vantagens Visuais

Um gráfico lollipop apresenta as mesmas informações que um gráfico de barras, mas com um estilo visual mais limpo e leve — especialmente eficaz quando há muitas categorias a serem exibidas.

Quando Preferir Lollipops

- Muitas categorias para visualizar
- Necessidade de um visual mais moderno
- Redução de "tinta" no gráfico
- Foco nos valores específicos

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(Sepal.Length=mean(Sepal.Length))

grf <- plot_lollipop(data,
  colors=colors[1],
  max_value_gap=0.2) +
  font + coord_flip()
plot(grf)
```



A orientação horizontal (`coord_flip`) facilita a leitura de rótulos de categoria longos e cria um visual elegante.

Gráfico de Barras com Barras de Erro

Representando Variabilidade

Usado para mostrar médias com variabilidade associada, como intervalos de confiança ou desvios padrão. As barras de erro comunicam a incerteza ou dispersão dos dados.

Interpretação

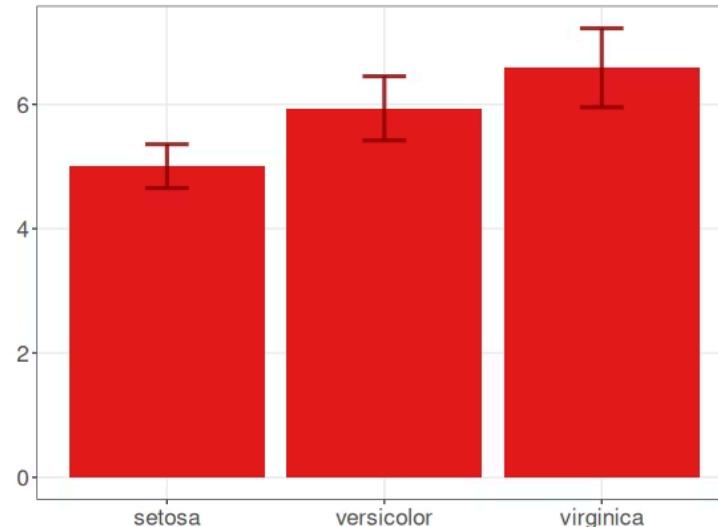
Barras de erro maiores indicam maior variabilidade nos dados, enquanto barras menores sugerem dados mais consistentes. Este tipo de visualização é essencial em pesquisas científicas.

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(mean=mean(Sepal.Length),
           sd=sd(Sepal.Length))

grf <- plot_bar(data, colors=colors[1], alpha=1) + font
grf <- grf +
  geom_errorbar(aes(x=Species,
                     ymin=mean-sd,
                     ymax=mean+sd),
                width=0.2, colour="darkred",
                alpha=0.8, size=1.1)
plot(grf)
```

Documentação: [grf_bar_error.md](#)



As barras vermelhas mostram um desvio padrão acima e abaixo da média, fornecendo contexto sobre a variabilidade dos dados.

Gráfico de Pizza (Pie Chart)

Representação de Proporções

Gráfico estatístico circular usado para ilustrar proporções numéricas.

Cada fatia representa uma categoria, e o tamanho da fatia é proporcional à quantidade que ela representa.

Considerações de Uso

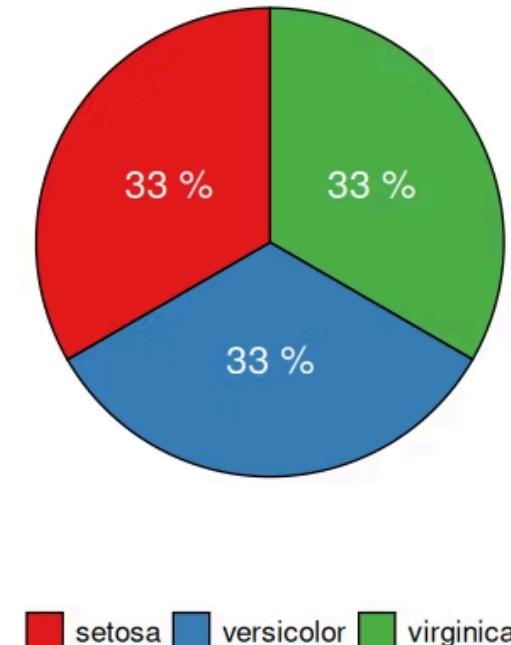
Embora populares, gráficos de pizza devem ser usados com cautela.

Eles funcionam melhor com poucas categorias (3-5) e quando as diferenças entre as proporções são substanciais.

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(n = n())

grf <- plot_pieplot(data, colors=colors[1:3]) + font
plot(grf)
```



Neste exemplo, as três espécies de íris estão igualmente representadas no dataset, cada uma com 33.3% do total.

Documentação: [grf_pie.md](#)

Gráfico de Barras Agrupadas

Comparações Multidimensionais

Gráficos de barras agrupadas permitem comparar subcategorias dentro de categorias principais. Cada grupo mostra barras lado a lado, uma por subcategoria.

Aplicações

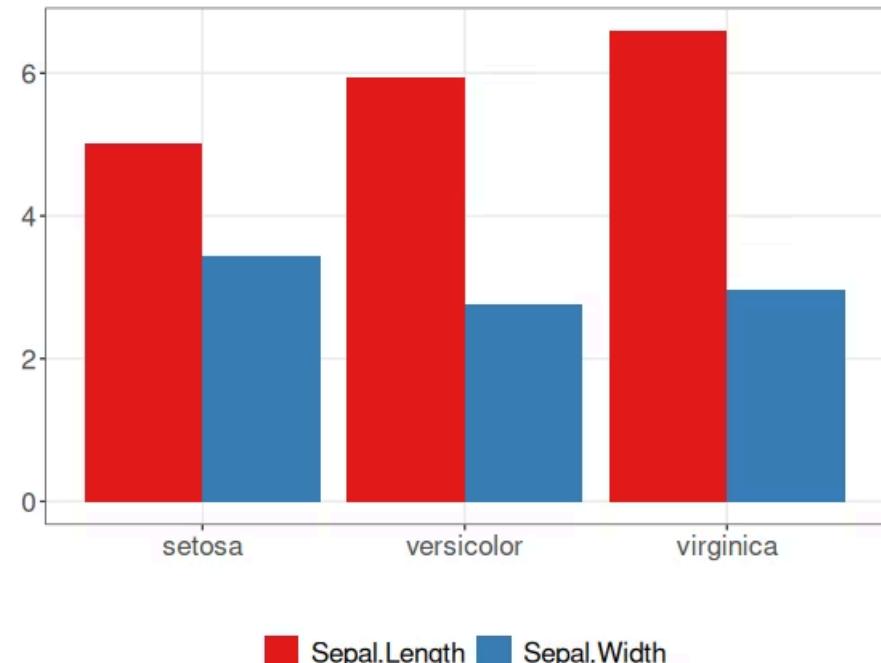
- Comparar múltiplas métricas por categoria
- Análise temporal com várias séries
- Estudos comparativos entre grupos

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(Sepal.Length=mean(Sepal.Length),
            Sepal.Width=mean(Sepal.Width))

grf <- plot_groupedbar(data, colors=colors[1:2]) + font
plot(grf)
```

Documentação: [grf_grouped_bar.md](#)



Este gráfico compara simultaneamente o comprimento e a largura médios das sépalas para cada espécie, facilitando análises comparativas multidimensionais.

Gráfico de Barras Empilhadas

Visualizando Totais Compostos

A altura da barra mostra o resultado combinado dos grupos. Este tipo de gráfico é útil para mostrar como diferentes componentes contribuem para um total.

Vantagens

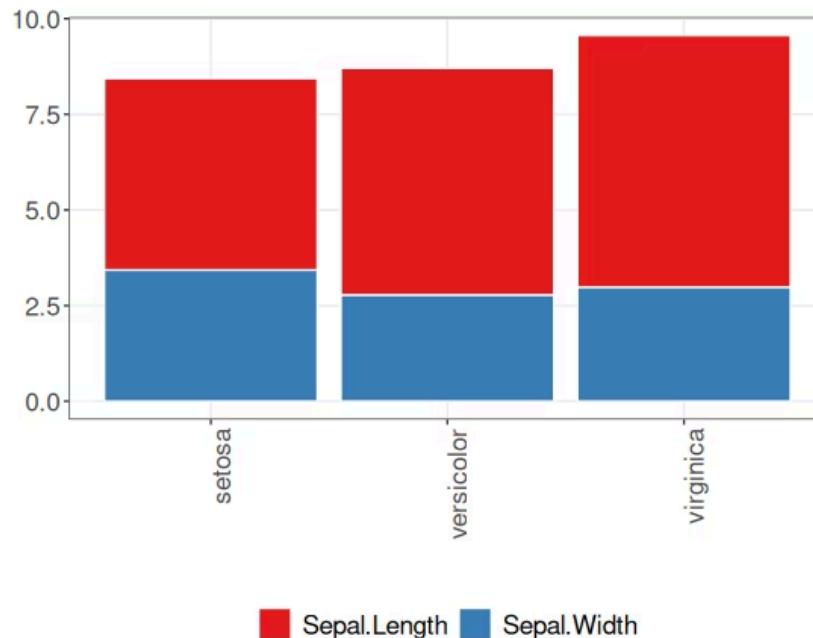
- Mostra tanto os totais quanto as partes
- Facilita comparação de composição
- Economiza espaço horizontal

Código de Exemplo

```
library(dplyr)
data <- iris |>
  group_by(Species) |>
  summarize(Sepal.Length=mean(Sepal.Length),
            Sepal.Width=mean(Sepal.Width))

grf <- plot_stackedbar(data, colors=colors[1:2]) + font
grf <- grf +
  theme(axis.text.x = element_text(angle=90, hjust=1))
plot(grf)
```

Documentação: [grf_stacked_bar.md](#)



Observe como os rótulos do eixo x foram rotacionados em 90 graus para melhorar a legibilidade quando os nomes das categorias são longos.

Gráfico de Linhas (Line Chart)

Visualizando Tendências Temporais

Gráficos de linhas são usados para exibir tendências ao longo do tempo ou qualquer sequência contínua. Eles são particularmente eficazes em análise de séries temporais, onde a ordem dos pontos de dados é importante.

Características

Cada ponto é conectado com linhas para destacar a evolução dos valores. Isso torna fácil identificar tendências, ciclos, picos e vales nos dados.

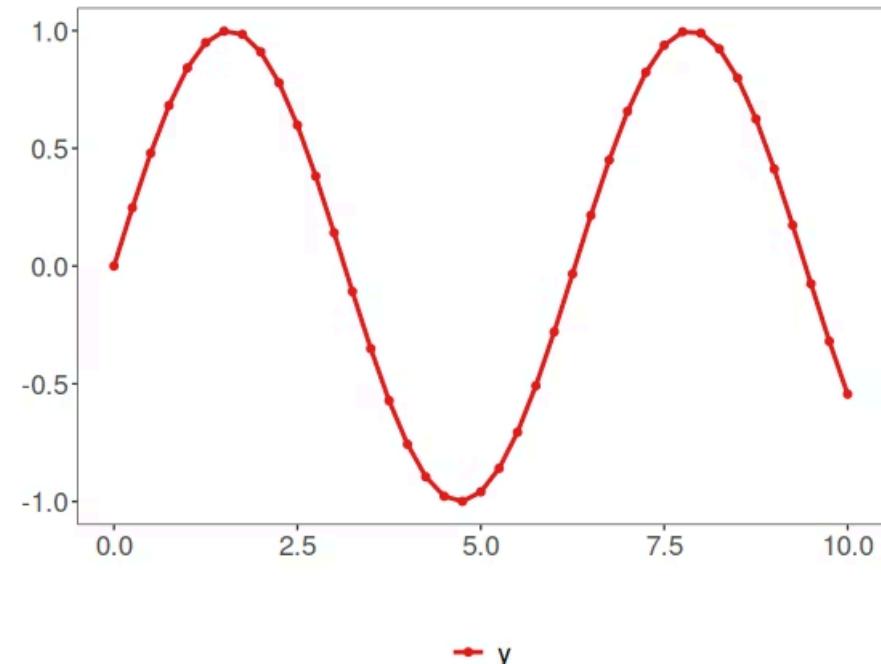
Aplicações Comuns

- Análise de séries temporais
- Monitoramento de KPIs ao longo do tempo
- Visualização de funções matemáticas
- Comparação de múltiplas tendências

Código de Exemplo

```
x <- seq(0, 10, 0.25)
serie <- data.frame(x, y=sin(x))

grf <- plot_series(serie, colors=colors[1]) + font
plot(grf)
```



Este exemplo mostra a função seno, demonstrando como gráficos de linhas podem visualizar efetivamente padrões cíclicos e comportamentos contínuos.



Visualização de Distribuições de Dados

Um guia completo para escolher e criar visualizações eficazes que revelam padrões, outliers e características fundamentais das suas distribuições de dados. Aprenda a comunicar insights estatísticos com clareza e precisão.

Preparando os Dados de Exemplo

Para demonstrar diferentes técnicas de visualização de distribuição, começamos criando um conjunto de dados sintético com três tipos distintos de distribuições estatísticas:

- **Exponencial:** 10.000 valores gerados com taxa = 1, representando eventos aleatórios ao longo do tempo
- **Uniforme:** 10.000 valores distribuídos igualmente entre 2,5 e 3,5
- **Normal:** 10.000 valores com média = 5, seguindo a curva em sino clássica

Essa abordagem permite comparar como diferentes visualizações revelam as características únicas de cada distribuição.

```
example <- data.frame(  
  exponential = rexp(10000, rate = 1),  
  uniform = runif(10000, min=2.5, max = 3.5),  
  normal = rnorm(10000, mean = 5)  
)
```

```
head(example)
```

- ❑ Usar variáveis aleatórias permite explorar visualmente como cada tipo de distribuição se manifesta em diferentes formatos gráficos.

Histogramas: Contando Frequências

O que é um histograma?

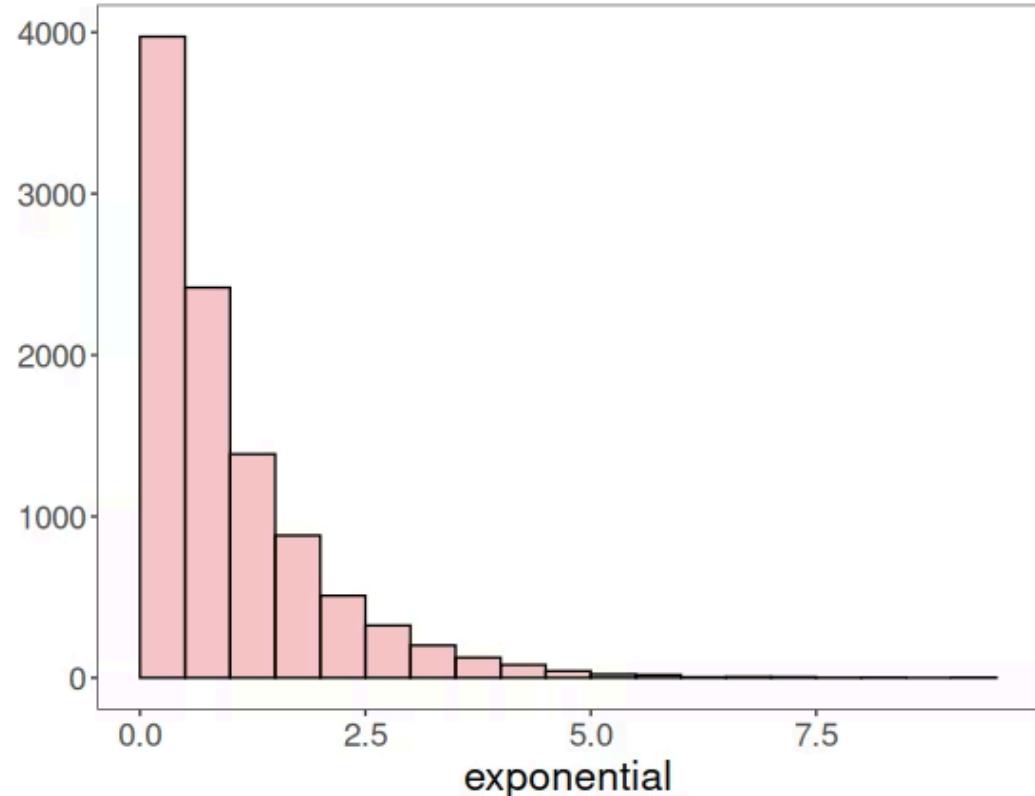
Um histograma organiza uma única variável contínua em **bins** (intervalos) e conta quantas observações caem em cada intervalo. É ideal para:

- Identificar a forma geral da distribuição
- Detectar assimetrias e outliers
- Visualizar concentração de valores
- Avaliar normalidade dos dados

```
library(dplyr)
data <- example |>
  select(exponential)

grf <- plot_hist(data,
  label_x = "exponential",
  color=colors[1]) + font

plot(grf)
```



Distribuição exponencial mostrando alta frequência próxima a zero com cauda longa à direita

Comparando Múltiplas Distribuições

Visualizar múltiplos histogramas lado a lado revela diferenças fundamentais entre distribuições. Observe como cada tipo de distribuição apresenta um padrão visual distinto e característico.

Exponencial

Concentração extrema em valores baixos com cauda longa. Típica de tempos de espera e processos de decaimento.

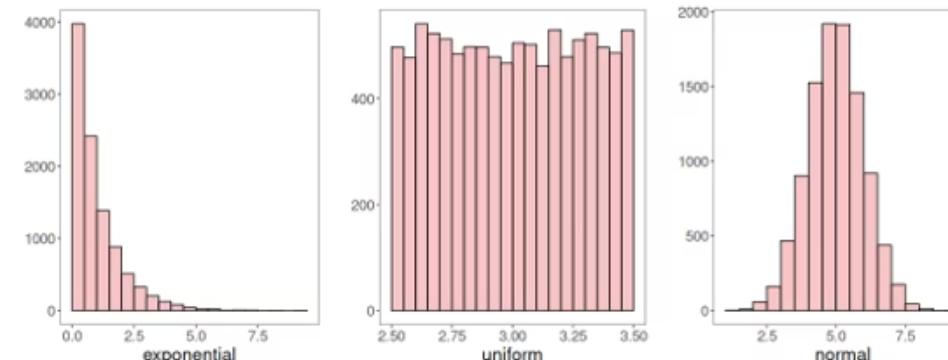
Uniforme

Distribuição plana e retangular onde todos os valores têm probabilidade igual. Ideal para modelar aleatoriedade pura.

Normal

Curva em sino simétrica, a distribuição mais comum na natureza. Fundamental para inferência estatística.

```
library(gridExtra)
grfe <- plot_hist(example |> select(exponential), label_x = "exponential", color=colors[1]) + font
grfu <- plot_hist(example |> select(uniform), label_x = "uniform", color=colors[1]) + font
grfn <- plot_hist(example |> select(normal), label_x = "normal", color=colors[1]) + font
grid.arrange(grfe, grfu, grfn, ncol=3)
```



Gráficos de Densidade: Suavizando a Distribuição

Estimativa Kernel de Densidade

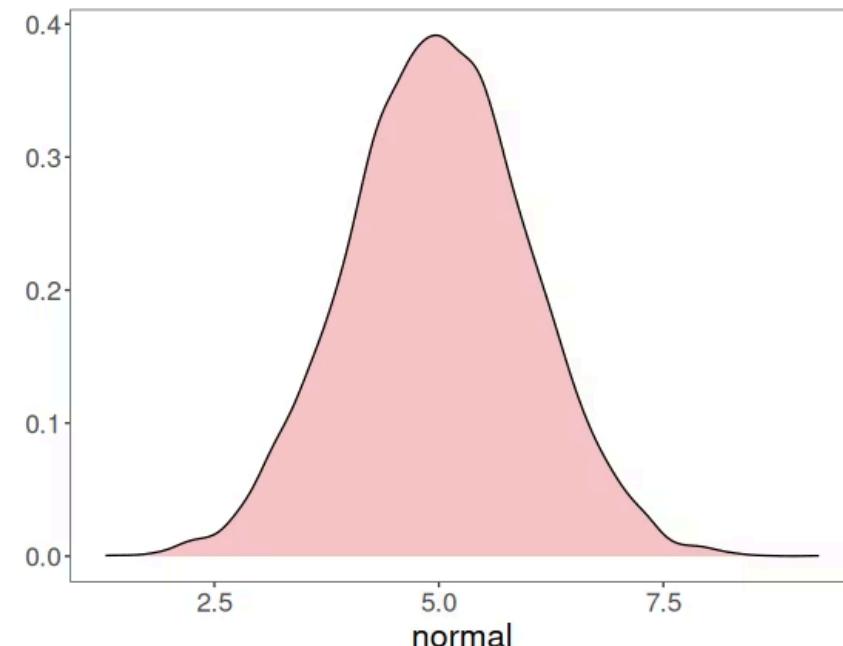
Enquanto histogramas usam bins discretos, gráficos de densidade criam uma **curva suave e contínua** que estima a função de densidade de probabilidade subjacente.

Vantagens sobre histogramas:

- Visualização mais suave e esteticamente clara
- Não depende da escolha arbitrária de bins
- Facilita comparação de múltiplas distribuições
- Revela multimodalidade com mais clareza

```
library(dplyr)
data <- example |>
  select(normal)

grf <- plot_density(data,
  label_x = "normal",
  color=colors[1]) + font
  plot(grf)
```



- ❑ A curva de densidade integra para 1, representando 100% da probabilidade total. A altura em qualquer ponto indica densidade relativa, não probabilidade absoluta.

Box Plots: Resumo Estatístico Visual

Anatomia do Box Plot

Box plots (diagramas de caixa) agrupam dados numéricos através de seus **quartis**, fornecendo um resumo visual poderoso de cinco estatísticas-chave:

01

Mínimo

Menor valor (excluindo outliers)

02

Q1(25%)

Primeiro quartil

03

Mediana (50%)

Valor central

04

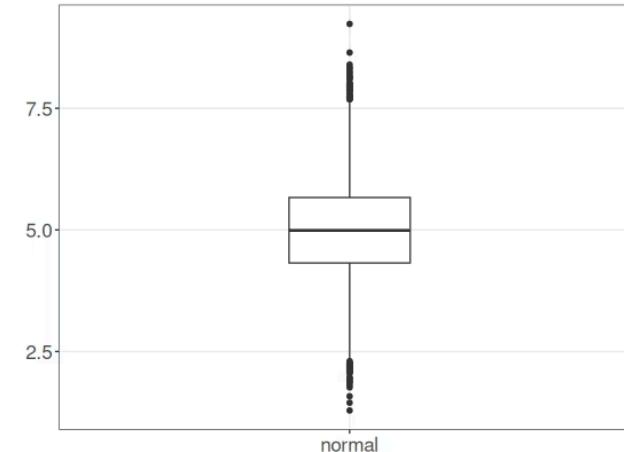
Q3(75%)

Terceiro quartil

05

Máximo

Maior valor (excluindo outliers)



```
library(dplyr)  
data <- example |>  
  select(normal)
```

```
grf <- plot_boxplot(data,  
  color="white") + font
```

```
plot(grf)
```

A caixa contém 50% dos dados centrais (IQR), com pontos individuais representando outliers potenciais.

 Documentação completa

Escolhendo a Visualização Certa

Selecionar o tipo de gráfico apropriado é fundamental para comunicar sua mensagem com eficácia. Esta tabela conecta objetivos analíticos com as visualizações mais adequadas.

| Objetivo da Análise | Gráfico Recomendado |
|-----------------------------------|--|
| Comparar valores entre categorias | Gráfico de barras, barras agrupadas/empilhadas |
| Mostrar parte de um todo | Gráfico de pizza, barras empilhadas |
| Mostrar distribuição | Histograma, boxplot, gráfico de densidade |
| Mostrar relacionamentos | Gráfico de dispersão, lollipop |
| Mostrar tendências temporais | Gráfico de linha (séries temporais) |
| Comparar grupos com variabilidade | Gráfico de barras com barras de erro |

-  **Dica profissional:** Considere sempre seu público-alvo e o nível de familiaridade deles com visualizações estatísticas ao escolher o formato.

Melhores Práticas em Visualização

Visualização de dados eficaz é sobre **comunicação, não decoração**. Seguir estes princípios garante que seus gráficos sejam claros, precisos e impactantes.



Rótulos Claros

Use títulos descritivos em eixos e legendas. Seu público não deveria adivinhar o que está visualizando.



Tipo Adequado

Combine o tipo de gráfico com sua estrutura de dados e pergunta de pesquisa. Não force dados em formatos inadequados.



Cores Acessíveis

Use paletas de cores como ColorBrewer que são distinguíveis para pessoas com daltonismo.



Evite Chartjunk

Remova decoração desnecessária como efeitos 3D, sombras excessivas ou elementos que não agregam informação.



Guie a Interpretação

Use tooltips ou anotações para destacar insights importantes e facilitar a compreensão.



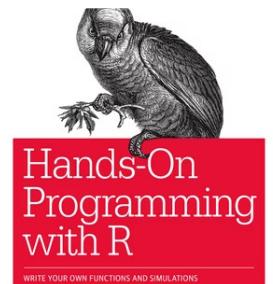
Escalas Consistentes

Mantenha unidades e escalas consistentes ao comparar múltiplos gráficos ou visualizações.

"O objetivo da visualização é comunicação - não decoração."

— Edward Tufte, pioneiro em visualização de dados

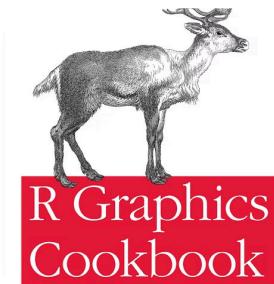
Referências



Hands-on Programming with R

Aprenda a escrever suas próprias funções e simulações

[rstudio-
education.github.io/hopr](https://rstudio-education.github.io/hopr)



R Graphics Cookbook

Guia completo para criar visualizações impressionantes

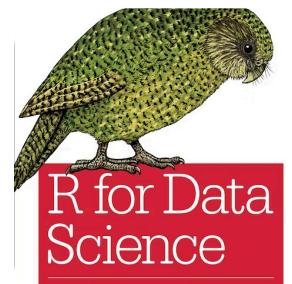
r-graphics.org



R Packages

Desenvolva e compartilhe seus próprios pacotes R

r-pkgs.org



R for Data Science

Referência definitiva para análise de dados moderna

r4ds.had.co.nz

Leitura adicional sobre fundamentos: <https://rstudio-education.github.io/hopr/basics.html>