CENTRO UNIVERSITÁRIO IBMEC

CAMPUS BARRA

MBA EM IA, DATA SCIENCE E BIG DATA PARA NEGÓCIOS

ESTATÍSTICA COM R

AULA 3

Ibmec

Sumário

1. Tipos de Variáveis em Estatística

Em Estatística, as **variáveis** podem ser classificadas em dois grandes grupos: **quantitativas** e **qualitativas**. Essa categorização é essencial para entender como os dados podem ser medidos, organizados e analisados. Cada tipo de variável tem suas características específicas e influenciam diretamente os métodos estatísticos a serem aplicados. A Figura 1 representa esse esquema de classificação.

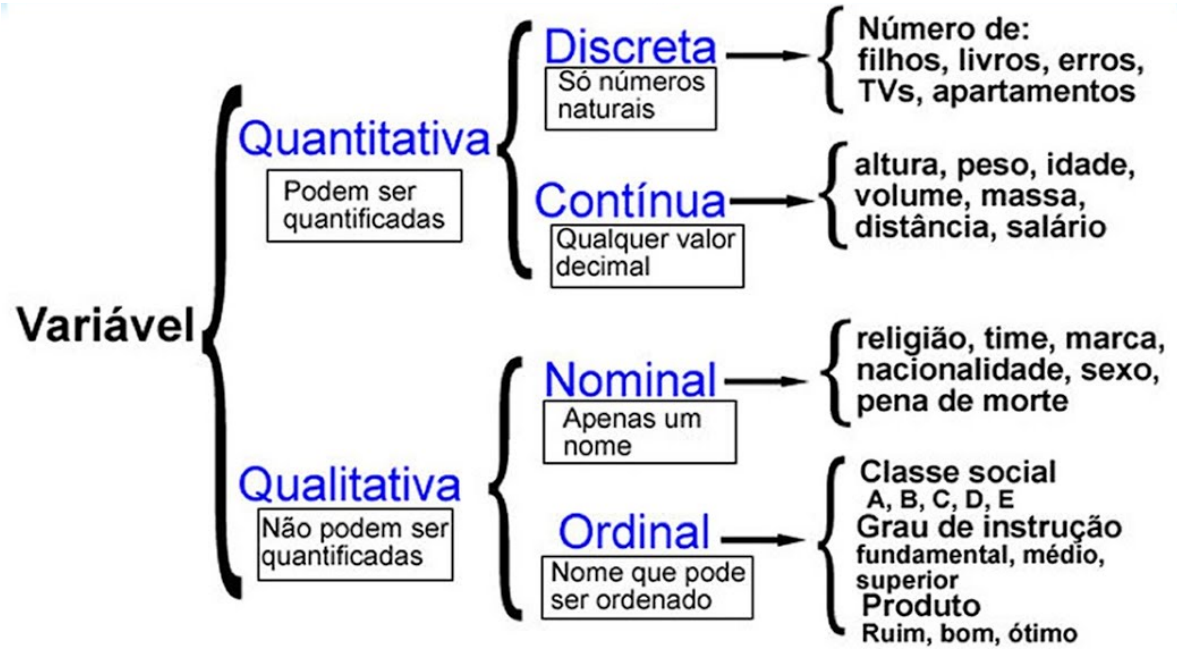


Figura 1: Tipos de variáveis (Prof. Grings – o matematico.com).

1.1. Variáveis Quantitativas

As variáveis **quantitativas** são aquelas que podem ser **quantificadas** — ou seja, expressas em números. Elas estão associadas a medidas e contagens. As variáveis quantitativas podem ser subdivididas em dois tipos: **discretas** e **contínuas**.

1.1.1. Variáveis Quantitativas Discretas

As variáveis **discretas** são aquelas que assumem **valores inteiros**. Ou seja, os valores dessas variáveis são números naturais (não fracionários). Elas surgem geralmente em **contagens**.

Exemplos de variáveis discretas:

- **Número de filhos**: 0, 1, 2, 3, etc.
- **Número de livros**: 5, 10, 15, etc.
- **Número de erros em uma prova**: 2, 3, 5, etc.
- **Número de aparelhos de TV em uma casa**: 1, 2, 3, etc.

Essas variáveis expressam contagens que não podem ser divididas em frações.

1.1.2. Variáveis Quantitativas Contínuas

As variáveis **contínuas** podem assumir **qualquer valor dentro de um intervalo**. Isso significa que podem incluir números decimais ou fracionários. Elas estão relacionadas a **medições**.

Exemplos de variáveis contínuas:

- **Altura**: 1,65 m, 1,75 m, etc.
- **Peso**: 65,8 kg, 72,3 kg, etc.
- **Idade**: 21,5 anos, 34,2 anos, etc.
- **Volume**: 200,5 ml, 1,25 litros, etc.
- **Distância**: 3,8 km, 12,5 km, etc.
- **Salário**: R$ 2.500,00, R$ 3.800,50, etc.

Essas variáveis podem assumir uma infinidade de valores dentro de um intervalo, permitindo que sejam fracionadas e medidas com precisão.

1.2. Variáveis Qualitativas

As variáveis **qualitativas**, também chamadas de **categóricas**, são aquelas que não podem ser quantificadas numericamente. Em vez disso, classificam-se as observações em diferentes **categorias** ou **grupos**. Elas são categorizadas em **nominais** e **ordinais**.

1.2.1 Variáveis Qualitativas Nominais

As variáveis **nominais** são categóricas e **não têm ordem** entre elas. São apenas nomes ou rótulos usados para identificar diferentes categorias, e essas categorias não podem ser organizadas hierarquicamente.

Exemplos de variáveis nominais:

- **Religião**: católico, evangélico, ateu, etc.
- **Time de futebol**: Flamengo, Corinthians, Palmeiras, etc.
- **Marca de carro**: Toyota, Ford, Honda, etc.
- **Nacionalidade**: brasileiro, argentino, francês, etc.
- **Sexo**: masculino, feminino.

Essas variáveis dividem os elementos em grupos que são mutuamente exclusivos, mas que **não têm uma ordem lógica**.

1.2.2 Variáveis Qualitativas Ordinais

As variáveis **ordinais** são categóricas, mas têm uma **ordem natural** ou **hierarquia**. Nesse tipo de variável, é possível organizar as categorias de maneira sequencial, embora não se saiba a **diferença exata** entre os níveis.

Exemplos de variáveis ordinais:

- **Classe social**: A, B, C, D, E.

- **Grau de instrução**: fundamental, médio, superior.
- **Avaliação de produtos**: ruim, bom, ótimo.

Embora essas variáveis representem categorias, há uma **ordem** entre elas. No entanto, as diferenças entre as categorias não são quantificadas de maneira precisa, apenas classificadas.

2. Tipos de Dados no R

O R possui uma variedade de tipos de dados que são fundamentais para manipular e armazenar informações. A escolha do tipo de dado apropriado depende do tipo de informação que você está lidando (como números, texto, fatores, etc.) e das operações que você pretende realizar sobre eles. A seguir estão os principais tipos de dados do R, com suas características e exemplos.

2.1. Tipos Básicos

Os **tipos básicos** são os tipos de dados mais simples no R. Cada objeto em R pertence a uma dessas classes básicas.

2.1.1 numeric (Numérico)

- Representa números de ponto flutuante, ou seja, números reais com casas decimais.
- O valor padrão de qualquer número em R é `numeric`, mesmo que não contenha casas decimais.

Exemplo:
```
num_exemplo <- 3.14
class(num_exemplo) #Retorna "numeric"
```

2.1.2. integer (Inteiro)

- Representa números inteiros, sem casas decimais.
- Números inteiros são definidos com um sufixo `L`.

Exemplo:
```
int_exemplo <- 5L
class(int_exemplo)  # Retorna "integer"
```

2.1.3. character (Caractere ou Texto)

- Representa texto ou cadeias de caracteres.
- Strings são delimitadas por aspas simples ou duplas.

Exemplo:
```
char_exemplo <- "Olá, mundo!"
class(char_exemplo)  # Retorna "character"
```

2.1.4. logical (Lógico)

- Representa valores booleanos: `TRUE` (verdadeiro) ou `FALSE` (falso).

Exemplo:
```
log_exemplo <- TRUE
class(log_exemplo)  # Retorna "logical"
```

2.1.5. complex (Complexo)

- Representa números complexos, com uma parte real e uma parte imaginária.

Exemplo:
```
comp_exemplo <- 1 + 2i
class(comp_exemplo)  # Retorna "complex"
```

2.2. Estruturas de Dados Compostas

Além dos tipos atômicos básicos, o R também possui estruturas de dados que podem armazenar diferentes tipos de informações, sejam elas homogêneas (todos os elementos do mesmo tipo) ou heterogêneas (elementos de tipos diferentes).

2.2.1. vector (Vetor)

- Armazena uma sequência de elementos que são todos do **mesmo tipo** (homogêneo).
- Pode ser de qualquer tipo atômico (numeric, integer, character, etc.).

Exemplo:
```
vetor_exemplo <- c(1, 2, 3, 4)
class(vetor_exemplo)  # Retorna "numeric"
```

2.2.2. factor (Fator)

- Armazena dados categóricos.
- Um fator é uma variável categórica com níveis.

Exemplo:
```
fator_exemplo <- factor(c("pequeno", "grande", "médio", "grande"))
class(fator_exemplo)  # Retorna "factor"
```

2.2.3. matrix (Matriz)

- Estrutura de dados bidimensional (linhas e colunas) que armazena elementos **homogêneos** (todos do mesmo tipo).

Exemplo:
```
matriz_exemplo <- matrix(1:9, nrow = 3, ncol = 3)
class(matriz_exemplo)  # Retorna "matrix"
```

2.2.4. data.frame (Data Frame)

- Estrutura de dados bidimensional que pode armazenar diferentes tipos de dados (heterogêneos) em suas colunas.

- É amplamente utilizado para armazenar e manipular conjuntos de dados.

Exemplo:
```
df_exemplo <- data.frame(
  Nome = c("Ana", "Bruno", "Carla"),
  Idade = c(22, 35, 28),
  Altura = c(1.65, 1.75, 1.60)
)
class(df_exemplo)  # Retorna "data.frame"
```

2.2.5. list (Lista)

- Estrutura de dados que pode armazenar elementos de **tipos diferentes**.
- Pode conter vetores, data frames, fatores, e até outras listas.

Exemplo:
```
lista_exemplo <- list(Nome = "João", Idade = 30, Notas = c(8.5, 9.0, 7.5))
class(lista_exemplo)  # Retorna "list"
```

2.3. Outros Tipos de Dados

2.3.1. NULL

- Representa a ausência de um valor ou objeto.

Exemplo:
```
objeto_nulo <- NULL
class(objeto_nulo)  # Retorna "NULL"
```

2.3.2. NA (Not Available)

- Representa um valor ausente ou faltante.

Exemplo:
```
vetor_na <- c(1, 2, NA, 4)
is.na(vetor_na)  # Verifica quais valores são NA
```

2.3.3. Inf e -Inf

- Representa valores infinitos positivos e negativos, respectivamente, geralmente resultantes de operações como divisão por zero.

Exemplo:
```
inf_exemplo <- 1 / 0
neg_inf_exemplo <- -1 / 0
```

2.3.4. NaN (Not a Number)

- Representa um resultado indefinido, como 0/0 ou a raiz quadrada de um número negativo.

Exemplo:
```
nan_exemplo <- 0 / 0
```

3. Visualização Gráfica dos Dados

A **visualização gráfica dos dados** é a representação visual de informações e dados por meio de gráficos, permitindo uma compreensão rápida e intuitiva das distribuições, tendências e padrões presentes em um conjunto de dados. Gráficos como histogramas, boxplots, gráficos de barras, gráficos de dispersão e gráficos de linhas são comumente utilizados para facilitar a interpretação dos dados. Essa abordagem transforma números e estatísticas em imagens compreensíveis, auxiliando na análise e tomada de decisões, ao mesmo tempo que destaca relações, variações e possíveis outliers de forma mais acessível do que apenas com números.

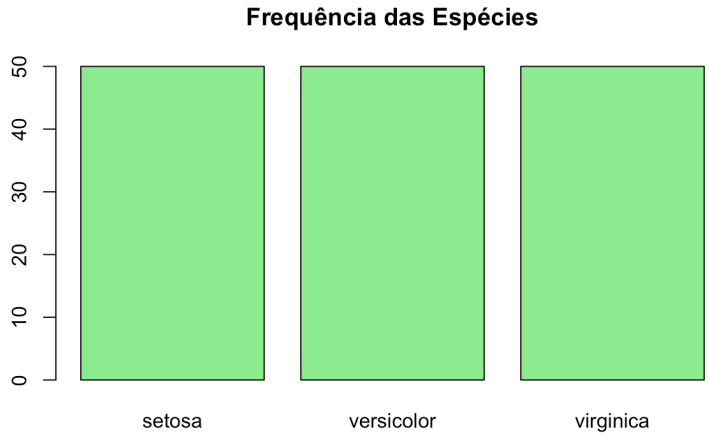
3.1. Principais Tipos de Gráficos

Em Estatística, os gráficos desempenham um papel essencial na visualização de dados e na comunicação de insights. Cada tipo de gráfico é adequado para diferentes tipos de dados e finalidades. Aqui estão os principais tipos de gráficos usados em Estatística:

3.1.1. Gráfico de Barras (Bar Plot)

- **Descrição:** Exibe categorias em barras verticais ou horizontais, com o comprimento das barras representando a frequência ou proporção de cada categoria.
- **Uso:** Ideal para comparar categorias ou grupos discretos.
- **Exemplo:** Frequência de espécies no dataset `iris`.

```
barplot(table(iris$Species), main = "Frequência das Espécies", col =
"lightgreen")
```

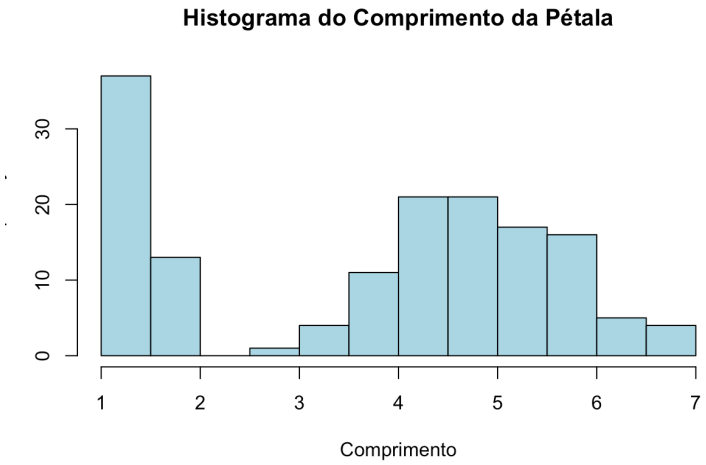


3.1.2. Histograma (Histogram)

- **Descrição:** Mostra a distribuição de uma variável numérica contínua, dividida em intervalos (*bins*). As barras são contíguas, representando a frequência de dados em cada intervalo.

- **Uso:** Utilizado para visualizar a distribuição de variáveis contínuas, detectando padrões como simetria, assimetria ou presença de *outliers*.
- **Exemplo:** Distribuição do comprimento das pétalas no dataset `iris`.

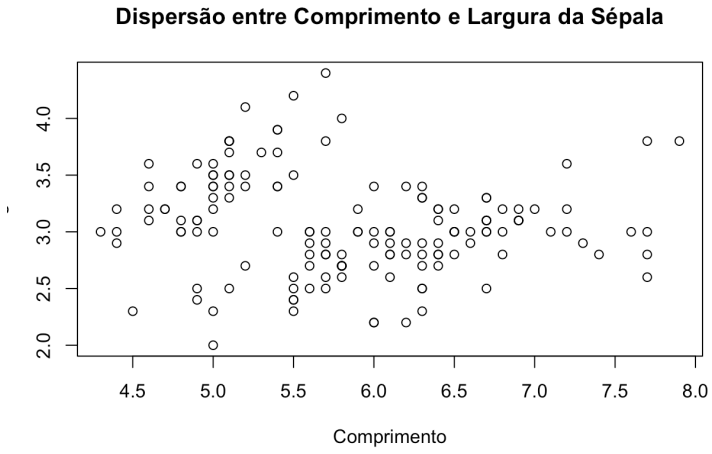
```
hist(iris$Petal.Length, main = "Histograma do Comprimento da Pétala", xlab =
"Comprimento", col = "lightblue")
```



3.1.3. Gráfico de Dispersão (Scatter Plot)

- **Descrição:** Representa a relação entre duas variáveis numéricas. Cada ponto no gráfico representa uma observação.
- **Uso:** Ideal para identificar correlação, tendências ou padrões entre duas variáveis contínuas.
- **Exemplo:** Relação entre o comprimento e a largura das sépalas.

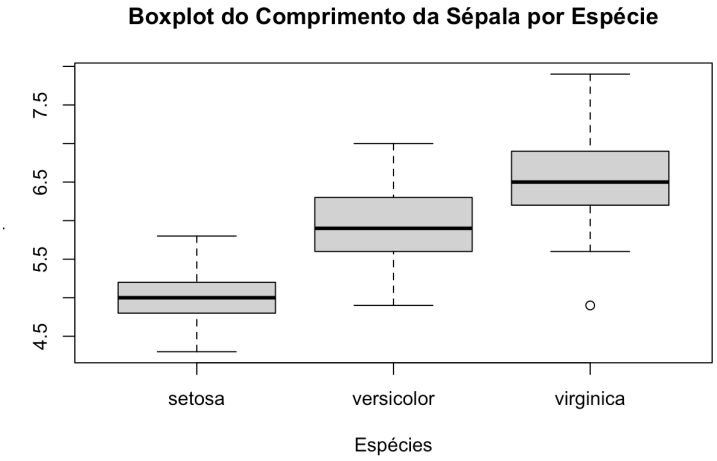
```
plot(iris$Sepal.Length, iris$Sepal.Width, main = "Dispersão entre Comprimento e
Largura da Sépala", xlab = "Comprimento", ylab = "Largura")
```



3.1.4. Boxplot (Gráfico de Caixa e Bigodes)

- **Descrição:** Mostra a distribuição de uma variável numérica através de seus quartis. Inclui a mediana, o primeiro e terceiro quartil (Q1 e Q3), os bigodes (extremos dos dados) e *outliers*.
- **Uso:** Utilizado para resumir a distribuição de dados e identificar outliers e assimetrias.
- **Exemplo:** Comparação do comprimento das sépalas entre espécies.

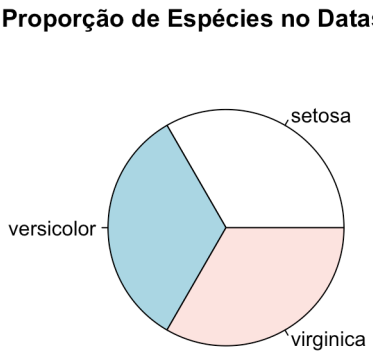
```
boxplot(Sepal.Length ~ Species, data = iris, main = "Boxplot do Comprimento da
Sépala por Espécie", xlab = "Espécies", ylab = "Comprimento")
```



3.1.5. Gráfico de Pizza (Pie Chart)

- **Descrição:** Divide um círculo em fatias que representam proporções de um todo, com cada fatia mostrando a proporção de uma categoria.
- **Uso:** Usado para visualizar proporções ou percentuais entre categorias.
- **Exemplo:** Proporção de cada espécie no dataset `iris`.

```
pie(table(iris$Species), main = "Proporção de Espécies no Dataset Iris")
```



Nota: Gráficos de pizza são frequentemente criticados por serem difíceis de interpretar quando há muitas categorias ou quando as proporções são muito próximas, e podem ser substituídos por gráficos de barras.

3.1.6. Gráfico de Linhas (Line Plot)

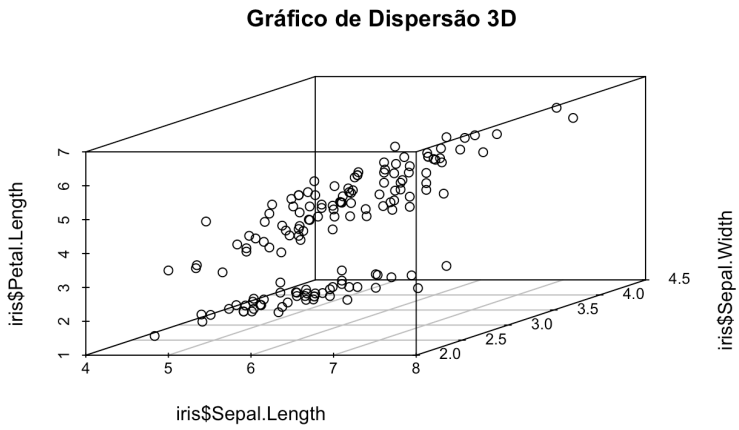
- **Descrição:** Mostra uma série de pontos conectados por uma linha, frequentemente usados para exibir mudanças ao longo do tempo ou a relação entre variáveis sequenciais.
- **Uso:** Comumente utilizado para séries temporais ou quando há uma sequência natural nas variáveis.
- **Exemplo:** Comparar a evolução de uma variável ao longo do tempo.

```
plot(1:10, rnorm(10), type = "l", main = "Gráfico de Linhas", xlab = "Tempo",
ylab = "Valor")
```

3.1.7. Gráfico de Dispersão 3D (3D Scatter Plot)

- **Descrição:** Semelhante ao gráfico de dispersão 2D, mas inclui uma terceira dimensão para visualizar a relação entre três variáveis numéricas.
- **Uso:** Ideal para identificar padrões em três variáveis contínuas, mas deve ser usado com cautela devido à dificuldade de interpretação em visualizações complexas.

```
library(scatterplot3d)
scatterplot3d(iris$Sepal.Length, iris$Sepal.Width, iris$Petal.Length, main =
"Gráfico de Dispersão 3D")
```



Resumo:

Tipo de Gráfico	Uso Principal	Tipos de Dados
Gráfico de barras	Comparar categorias	Categóricos
Histograma	Mostrar a distribuição de dados contínuos	Contínuos
Gráfico de dispersão	Visualizar a relação entre duas variáveis numéricas	Contínuos
Boxplot	Comparar distribuições e identificar *outliers*	Contínuos
Gráfico de pizza	Visualizar proporções de categorias	Categóricos
Gráfico de linhas	Mostrar tendência ao longo do tempo	Contínuos
Gráfico de dispersão 3D	Visualizar a relação entre três variáveis numéricas	Contínuos

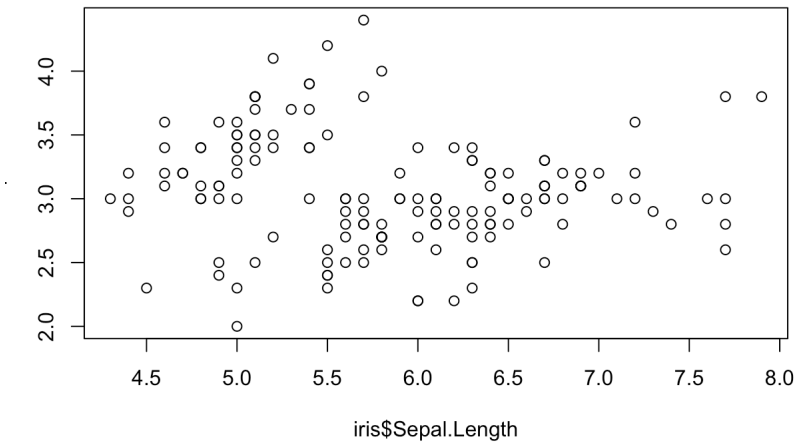
3.2. Práticas Ruins em Gráficos

Aqui estão alguns exemplos de gráficos que não seguem as boas práticas de visualização de dados em R.

3.2.1. Gráfico de Dispersão sem Rótulos e Título

Esse gráfico falha em fornecer contexto ao leitor, pois não tem títulos, rótulos de eixos ou legendas explicativas.

```
# Gráfico de dispersão ruim
plot(iris$Sepal.Length, iris$Sepal.Width)
```



Problemas:

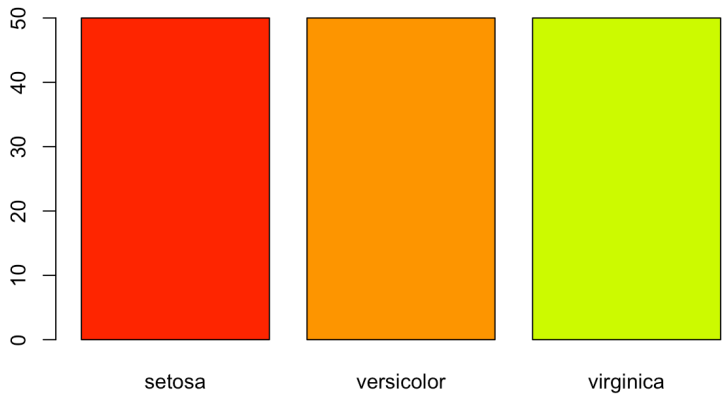
- **Sem título**: O leitor não sabe o que o gráfico representa.
- **Sem rótulos nos eixos**: Não há indicação clara do que está sendo medido.
- **Sem legenda**: Não há informações sobre como interpretar os dados.

Melhoria: Adicionar título, rótulos nos eixos e uma legenda explicativa.

3.2.2. Gráfico de Barras com Cores Excessivas

Usar muitas cores desnecessárias pode sobrecarregar a visualização e confundir o leitor.

```
# Gráfico de barras ruim com cores exageradas
barplot(table(iris$Species), col = rainbow(10))
```



Problemas:

- **Cores excessivas**: Embora as espécies sejam apenas 3, o gráfico usa 10 cores. Isso é visualmente poluído e desnecessário.
- **Dificuldade de leitura**: O uso de muitas cores não relacionadas confunde o usuário e tira o foco dos dados.

Melhoria: Utilizar uma paleta de cores simples e consistente que ajuda a distinguir as categorias sem sobrecarregar a visão.

3.2.3. Gráfico de Pizza com Muitos Segmentos

Gráficos de pizza não são adequados para comparar segmentos, especialmente quando há muitos valores ou eles são similares.

```
# Gráfico de pizza ruim
pie(table(iris$Species), main = "Distribuição das Espécies")
```

Problemas:

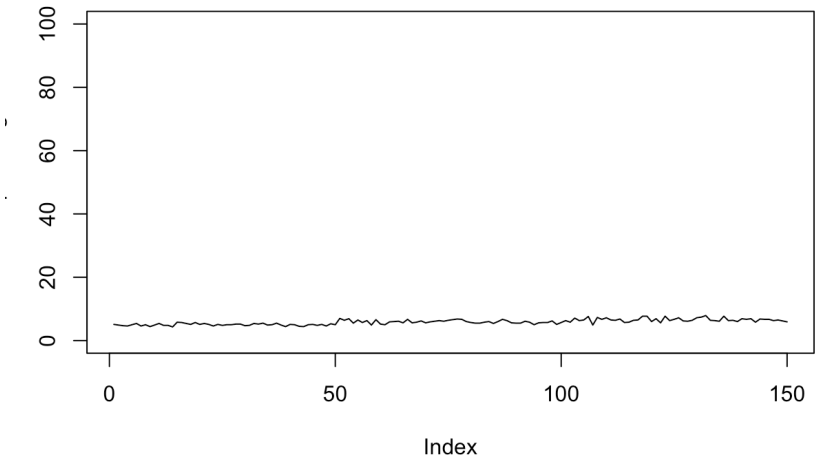
- **Dificuldade de comparação**: Gráficos de pizza são difíceis para comparar tamanhos de segmentos, principalmente quando os valores são próximos.
- **Número de segmentos**: Mesmo com apenas três categorias, esse gráfico já fica difícil de interpretar, pois os segmentos não possuem rótulos adicionais e estão muito próximos.

Melhoria: Substituir o gráfico de pizza por um gráfico de barras ou outro tipo mais adequado para comparação de categorias.

3.2.4. Gráfico com Escalas Inadequadas

Aqui temos um gráfico de linhas em que a escala não foi ajustada corretamente, criando uma impressão distorcida dos dados.

```
# Gráfico de linhas com escalas inadequadas
plot(iris$Sepal.Length, type = "l", ylim = c(0, 100))
```



Problemas:

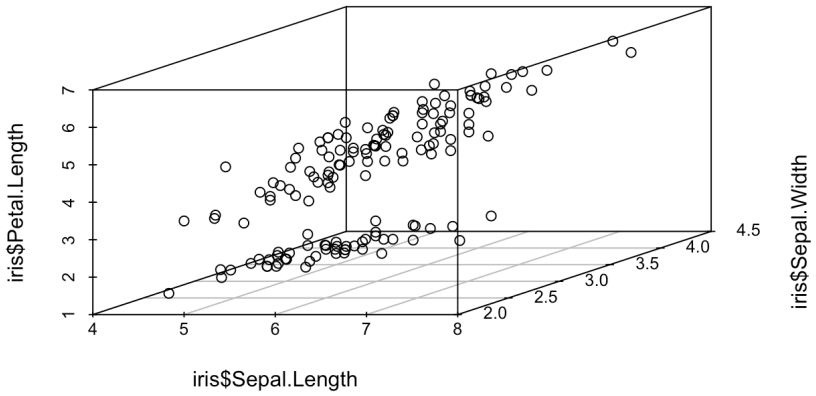
- **Escala vertical inadequada**: Definir manualmente o limite superior da escala em 100 faz com que os dados pareçam muito menores do que realmente são.
- **Linha sem contexto**: O gráfico utiliza `type = "l"`, mas os dados de sépala não são sequenciais, o que torna a linha irrelevante.

Melhoria: Permitir que o R defina automaticamente os limites da escala ou ajustar a escala de acordo com os dados.

3.2.5. Gráfico 3D Inútil

Gráficos tridimensionais podem parecer visualmente atraentes, mas muitas vezes são desnecessários e tornam a interpretação mais difícil.

```
# Exemplo de gráfico 3D ruim
library(scatterplot3d)
scatterplot3d(iris$Sepal.Length, iris$Sepal.Width, iris$Petal.Length)
```



Problemas:

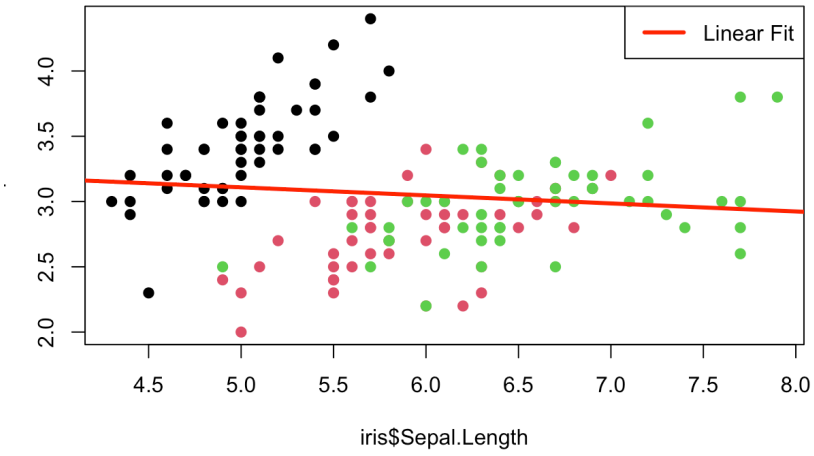
- **Dificuldade de leitura**: O gráfico 3D adiciona complexidade sem fornecer informações adicionais claras.
- **Sobrecarga visual**: Gráficos 3D tendem a distorcer proporções e dificultam a comparação dos pontos.
- **Excesso de informação**: A terceira dimensão não é útil ou necessária para este conjunto de dados.

Melhoria: Usar gráficos 2D claros e evitar gráficos 3D, a menos que sejam absolutamente necessários.

3.2.6. Gráfico com Sobrecarga de Informações

Este gráfico de dispersão adiciona muitos elementos visuais que tornam a leitura confusa.

```
# Gráfico de dispersão com muitos elementos
plot(iris$Sepal.Length, iris$Sepal.Width, col = iris$Species, pch = 19)
abline(lm(iris$Sepal.Width ~ iris$Sepal.Length), col = "red", lwd = 3)
legend("topright", legend = c("Linear Fit"), col = "red", lwd = 3)
```



Problemas:

- **Excesso de linhas**: A linha de regressão pode não ser necessária, adicionando ruído ao gráfico.
- **Elementos visuais sobrepostos**: O gráfico está poluído com diferentes cores e a linha grossa.

- **Legendas pouco claras**: A legenda apenas menciona o "Linear Fit", mas não explica o que os diferentes pontos coloridos representam.

Melhoria: Remover a linha de regressão e simplificar o gráfico para focar no objetivo da análise.

3.2.7. Gráfico de Linhas para Dados Categóricos

Este gráfico utiliza um gráfico de linhas para um conjunto de dados categóricos, o que é uma má prática.

```
# Gráfico de linhas aplicado incorretamente
plot(iris$Species, type = "l")
```

Problemas:

- **Tipo de gráfico inadequado**: Linhas sugerem uma continuidade nos dados, mas `Species` é uma variável categórica, e não faz sentido conectar os pontos.
- **Confusão no tipo de dado**: Utilizar linhas para dados categóricos cria uma interpretação incorreta do tipo de dado.

Melhoria: Utilizar um gráfico de barras ou boxplot, que são adequados para dados categóricos.