Visualização de dados com R

Aula 1 - Introdução a visualização de dados

Marcus Ramalho

2024-02-03

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

Marcus Ramalho

- Pesquiso FII's/PKM/Web3.0 com ciência de dados
- Matlab/Fortran/Pascal/VBA/AutoLISP(LISP)/PHP/HTML/ M(power query)/TypeScript/Python/R e Rust(loading)
- Administração UFF
- Mestrando PPGAd-UFF

Objetivos

Aula 1

- Tipos de Gráficos por Tipo de Variável
- Gráficos no R base, plot, hist, boxplot
- Exercícios de fixação 1

- gramática dos gráficos e ggplot2
- Exercícios de fixação 2
- Escalas
- Exploração de dados com o pacote esquisse
- Exercícios de fixação
- Tarefa com nota

Tipos de variáveis

- Variáveis categóricas - Representam categorias ou grupos
  - Nominais - Não possuem ordem Exemplo: sexo, cor dos olhos
  - Ordinais - Possuem ordem Exemplo: escolaridade, estado civil
- Variáveis contínuas - Representam valores numéricos
  - Discretas - Valores inteiros Exemplo: número de filhos, número de carros, número de acessos
  - Contínuas - Valores reais Exemplo: peso, altura, salário, etc.

Tipo de Gráficos por Tipo de Variável

- Variáveis categóricas
  - barras
  - setor
  - etc...
- Variáveis contínuas
  - Histograma
  - boxplot
  - Gráfico de dispersão
  - etc...

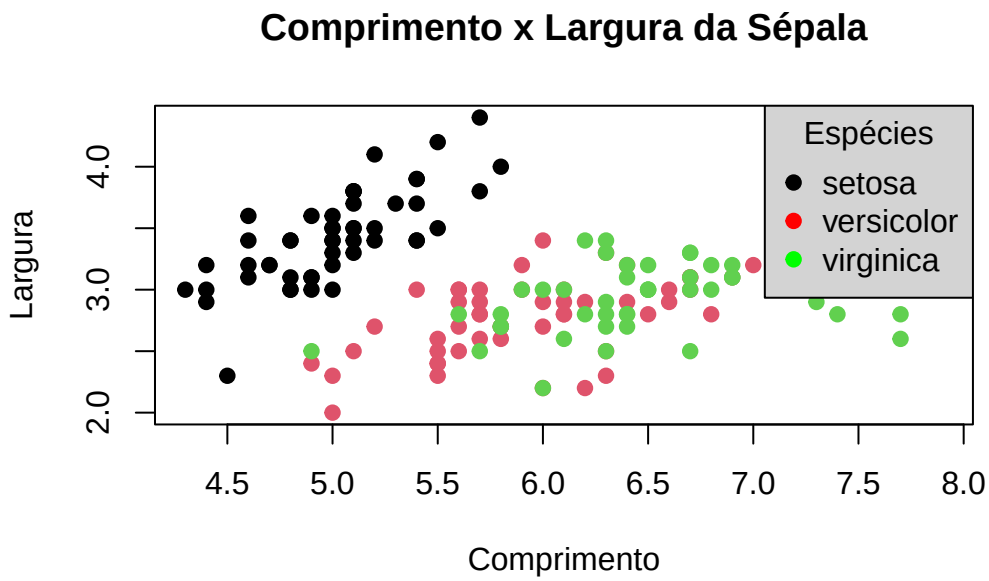
Um gráfico para cada tipo de variável

Ferramentas úteis:

Gráficos no R base - plot

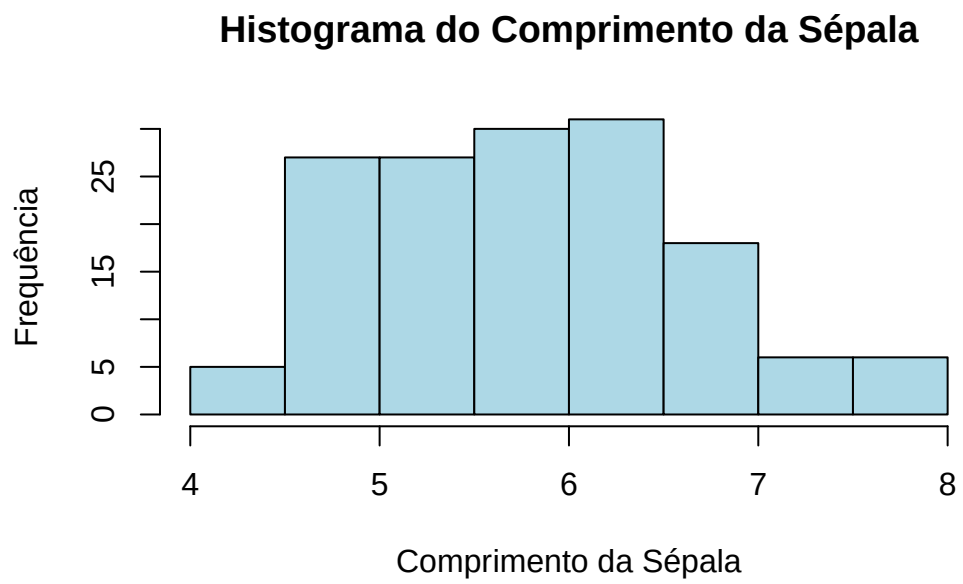
```r
plot(iris$Sepal.Length, iris$Sepal.Width,
    col = iris$Species, pch = 19,
    xlab = "Comprimento", ylab = "Largura", main = "Comprimento x Largura da Sépala"
)

# Legenda
legend(
    "topright",
    legend = levels(iris$Species),
    col = c("black", "red", "green"),
    pch = 19,
    title = "Espécies",
    bg = "lightgray"
)
```

Gráficos no R base - plot



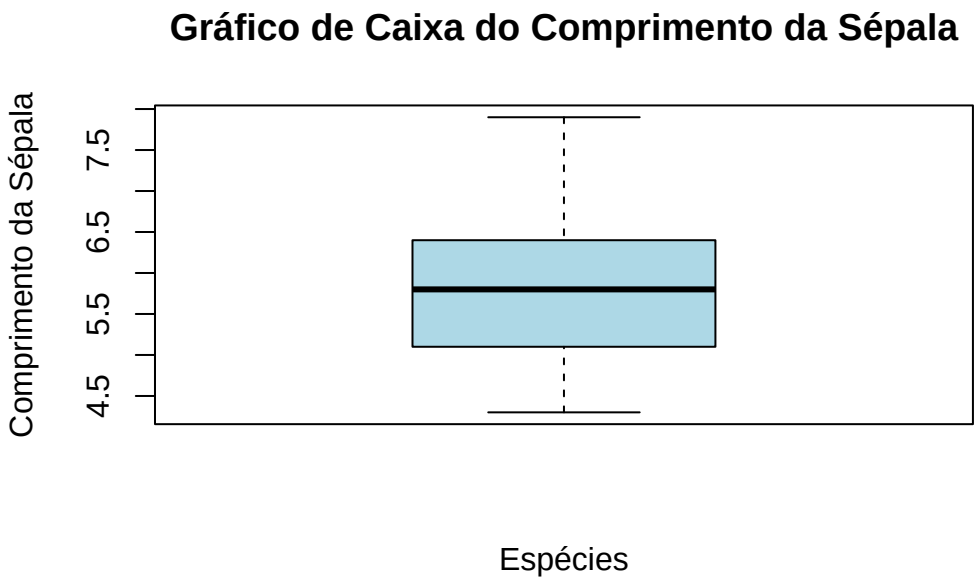
Gráficos no R base - hist

```
hist(iris$Sepal.Length,
    xlab = "Comprimento da Sépala",
    ylab = "Frequência",
    main = "Histograma do Comprimento da Sépala",
    col = "lightblue",
    border = "black"
)
```



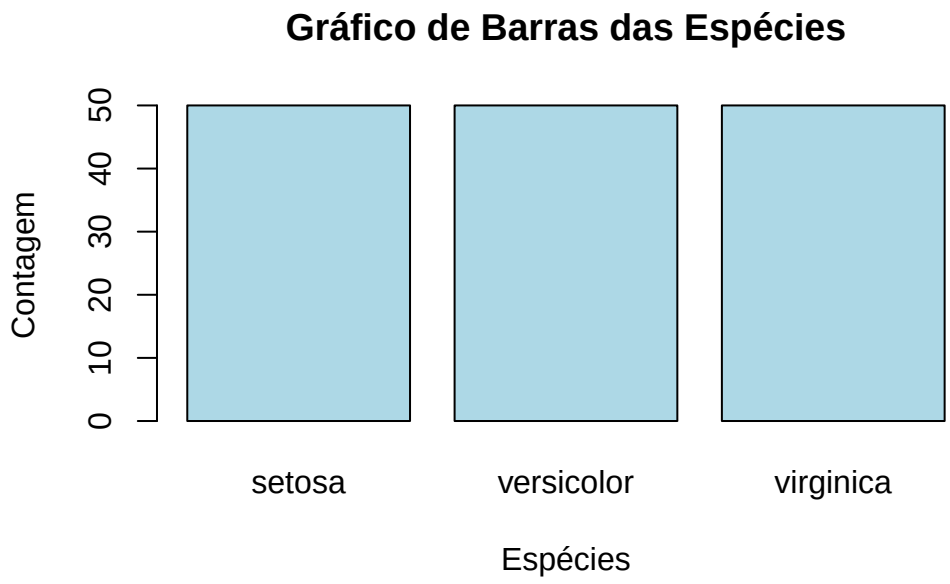
Gráficos no R base - boxplot

```
boxplot(iris$Sepal.Length,
    xlab = "Espécies",
    ylab = "Comprimento da Sépala",
    main = "Gráfico de Caixa do Comprimento da Sépala",
    col = "lightblue",
    border = "black"
)
```



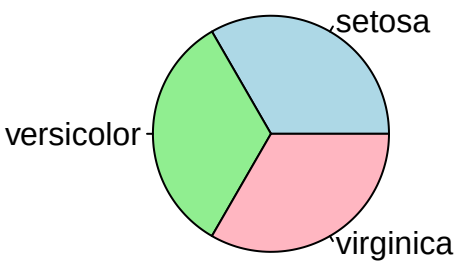
Gráficos no R base - barplot

```
barplot(table(iris$Species),
    xlab = "Espécies",
    ylab = "Contagem",
    main = "Gráfico de Barras das Espécies",
    col = "lightblue",
    border = "black"
)
```



Gráficos no R base - pie

```
pie(table(iris$Species),
    main = "Gráfico de Pizza das Espécies",
    col = c("lightblue", "lightgreen", "lightpink"),
    border = "black"
)
```



Exercícios de fixação 1

1 - Use o conjunto de dados 'casas' do pacote 'dados' para responder as questões a seguir.

```
install.packages("dados")
library(dados)
```

a. Identifique duas variáveis categóricas. Liste cinco exemplos de valores únicos para uma das variáveis categóricas.

b. Utilizando o pacote de gráficos base do R, escolha uma variável contínua e gere um histograma para visualizar a distribuição. Explique o que você observa no gráfico.

2 - Use o conjunto de dados 'dados_atmosfera' do pacote 'dados' para responder as questões a seguir.

a. Utilizando o pacote de gráficos base do R, crie um gráfico de barras para representar a distribuição da variável categórica "nuvem_baixa". Adicione rótulos aos eixos.

b. Crie um gráfico de dispersão para visualizar a relação entre as variáveis contínuas "temp_superficie" e "pressao". Adicione rótulos aos eixos x e y, e destaque cores diferentes para cada ano (utilizando a variável categórica "ano").

Exercícios de fixação 1 a - Resposta

Identificando todas as variáveis não numéricas do conjunto de dados

```
# Carregar o pacote
library(dados)

casas %>%
    select_if(~ is.character(.)) %>%
    names(.)
```

```
 [1] "pid"                          "moradia_classe"
 [3] "moradia_zoneamento"           "rua_tipo"
 [5] "beco_tipo"                    "lote_formato"
 [7] "terreno_contorno"             "utilidades"
 [9] "lote_config"                  "terreno_declive"
[11] "vizinhanca"                   "condicao_1"
[13] "condicao_2"                   "moradia_tipo"
[15] "moradia_estilo"               "geral_qualidade"
[17] "geral_condicao"               "telhado_estilo"
[19] "telhado_material"             "exterior_cobertura_1"
[21] "exterior_cobertura_2"         "alvenaria_tipo"
[23] "exterior_qualidade"           "exterior_condicao"
[25] "fundacao_tipo"                "porao_qualidade"
[27] "porao_condicao"               "porao_exposicao"
[29] "porao_acabamento_1"           "porao_acabamento_2"
[31] "aquecimento_tipo"             "aquecimento_qualidade_condicao"
[33] "ar_condicionado_central"      "sistema_eletrico_tipo"
[35] "cozinha_qualidade"            "funcional"
[37] "lareira_qualidade"            "garagem_tipo"
[39] "garagem_acabamento"           "garagem_qualidade"
[41] "garagem_condicao"             "entrada_veiculo_pavimentada"
[43] "piscina_qualidade"            "cerca_qualidade"
[45] "funcionalidades_diversas"     "venda_tipo"
[47] "venda_condicao"
```

Exercícios de fixação 1 a - Resposta

Listando cinco exemplos de valores únicos para a variável categórica “vizinhanca”

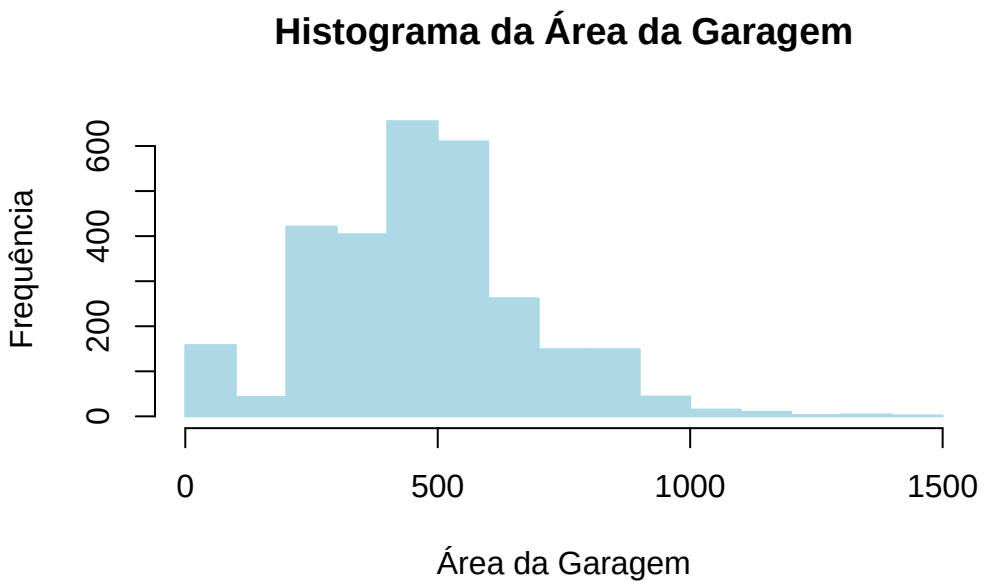
```r
casas %>%
    count(vizinhanca) %>%
    head(5)
```

```
# A tibble: 5 x 2
  vizinhanca              n
  <chr>               <int>
1 Bloomington Heights    28
2 Bluestem               10
3 Briardale              30
4 Brookside             108
5 Clear Creek            44
```

Exercícios de fixação 1 b - Resposta

Criando um histograma para a variável contínua "area_construida"

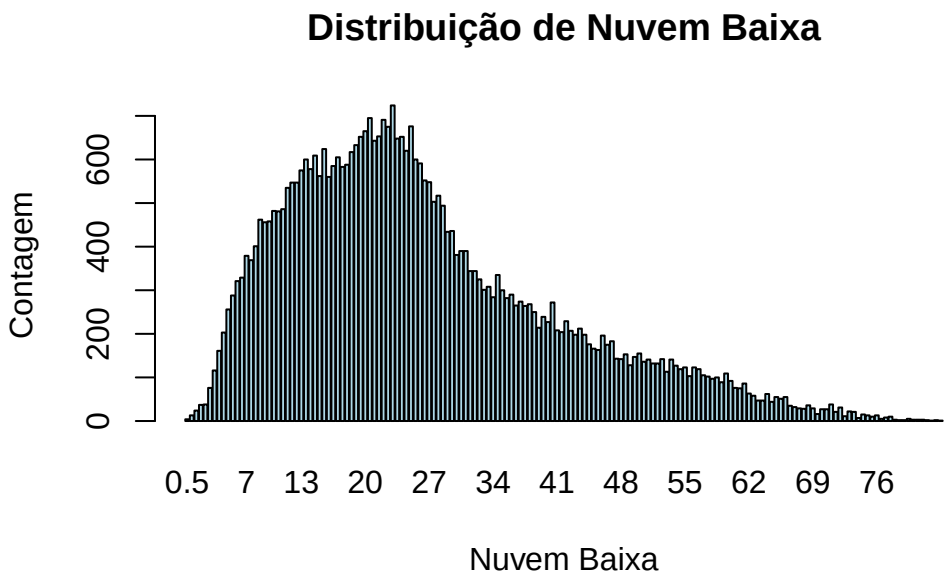
```r
# Criar o gráfico
hist(casas$garagem_area,
    xlab = "Área da Garagem",
    ylab = "Frequência",
    main = "Histograma da Área da Garagem",
    col = "lightblue",
    border = "lightblue"
)
```



Exercícios de fixação 2 a - Resposta

Criando um gráfico de barras para a variável categórica “nuvem_baixa”

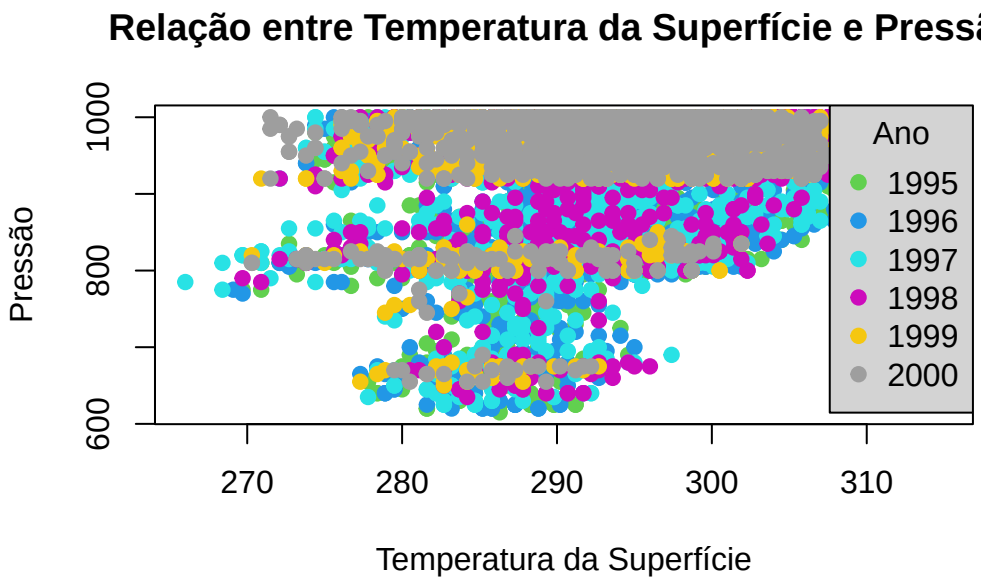
```
barplot(table(dados_atmosfera$nuvem_baixa),
    xlab = "Nuvem Baixa",
    ylab = "Contagem",
    main = "Distribuição de Nuvem Baixa",
    col = "lightblue",
    border = "black"
)
```



Exercícios de fixação 2 b - Resposta

Criando um gráfico de dispersão para as variáveis contínuas “temp_superficie” e “pressao”

```
# Criar o gráfico
plot(dados_atmosfera$temp_superficie, dados_atmosfera$pressao,
     col = as.numeric(dados_atmosfera$ano), pch = 19,
     xlab = "Temperatura da Superfície",
     ylab = "Pressão",
     main = "Relação entre Temperatura da Superfície e Pressão"
)

# Adicionar a legenda
legend(
     "topright",
     legend = unique(dados_atmosfera$ano),
     col = unique(as.numeric(dados_atmosfera$ano)),
     pch = 19,
     title = "Ano",
     bg = "lightgray"
)
```



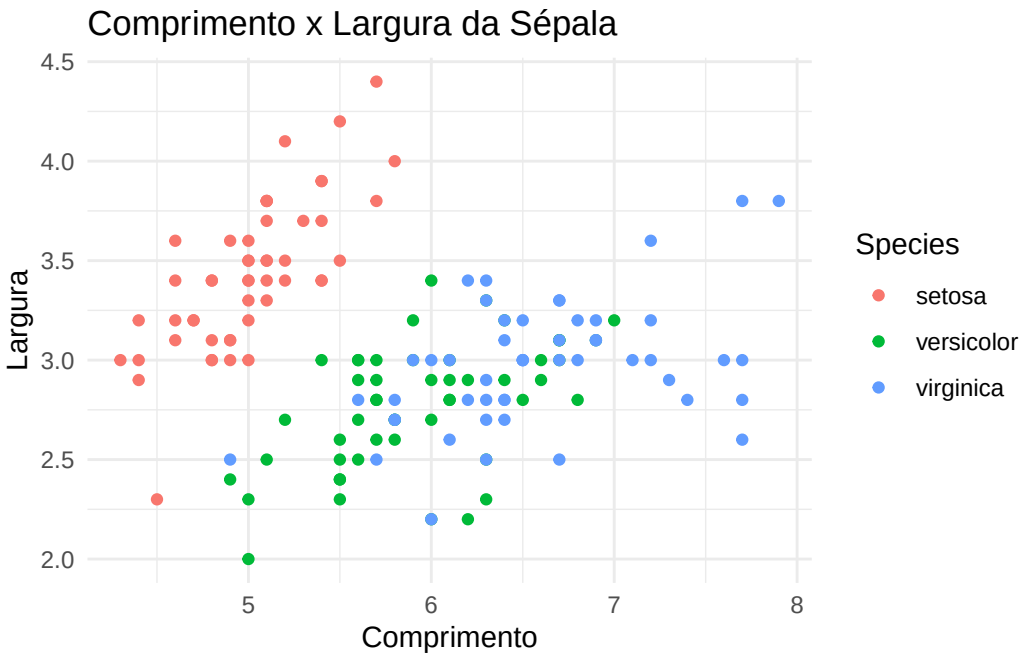
Gramática dos gráficos e ggplot2

O que é a gramática dos gráficos?

- É um conjunto de regras que descrevem a estrutura de um gráfico e como ele é construído a partir dos dados.
- O ggplot2 é um pacote do R que implementa a gramática dos gráficos.

ggplot2 - Exemplo

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
    geom_point() +
    labs(title = "Comprimento x Largura da Sépala", x = "Comprimento", y = "Largura") +
    theme_minimal()
```



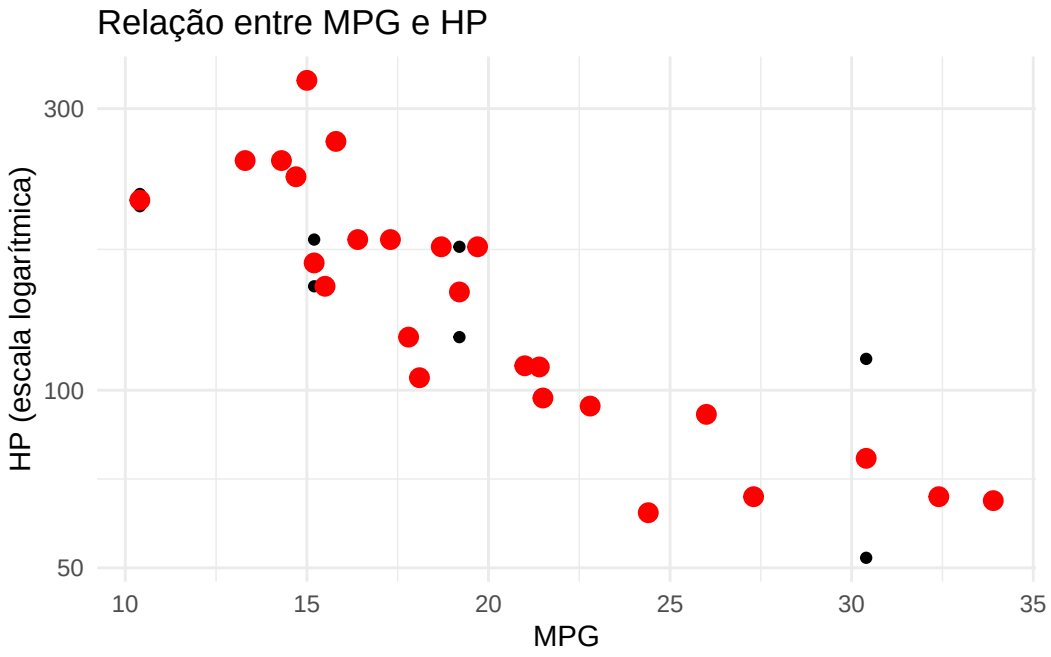
ggplot2 - Elementos da gramática

```
ggplot()
```

ggplot2 - Elementos da gramática

```
# Dados: Este é o conjunto de dados que queremos visualizar.
data <- mtcars

# Aesthetics: Mapeia variáveis nos dados para aspectos visuais do gráfico.
aes <- aes(x = mpg, y = hp)

# Geometrias: Descreve o tipo de gráfico que queremos criar.
geom <- geom_point()

# Escalas: Controla como os dados são mapeados para os aspectos visuais do gráfico.
scale <- scale_y_log10()

# Estatísticas: Realiza cálculos nos dados.
stat <- stat_summary(fun = mean, geom = "point", color = "red", size = 3)

# Tema: Controla a aparência não-dados do gráfico, como a cor de fundo e a fonte do texto.
theme <- theme_minimal()

# Títulos e legendas
```

```r
labels <- labs(
    title = "Relação entre MPG e HP",
    x = "MPG",
    y = "HP (escala logarítmica)",
    color = "Número de cilindros"
)

# Criar o gráfico
ggplot(data, aes) + geom + scale + stat + theme + labels
```



ggplot2 - Elementos da gramática

```r
# Dados: Este é o conjunto de dados que queremos visualizar.
data <- mtcars

# Aesthetics: Mapeia variáveis nos dados para aspectos visuais do gráfico.
aes <- aes(x = mpg, y = hp)

# Geometrias: Descreve o tipo de gráfico que queremos criar.
geom <- geom_point()
```

```r
# Escalas: Controla como os dados são mapeados para os aspectos visuais do gráfico.
scale <- scale_y_log10()

# Estatísticas: Realiza cálculos nos dados.
stat <- stat_summary(fun = mean, geom = "point", color = "red", size = 3)

# Coordenadas: Define o sistema de coordenadas do gráfico.

# Tema: Controla a aparência não-dados do gráfico, como a cor de fundo e a fonte do texto.
theme <- theme_minimal()

# Títulos e legendas
labels <- labs(
    title = "Relação entre MPG e HP",
    x = "MPG",
    y = "HP (escala logarítmica)",
    color = "Número de cilindros"
)

# Criar o gráfico
ggplot(data, aes) + geom + scale + stat + theme + labels
```

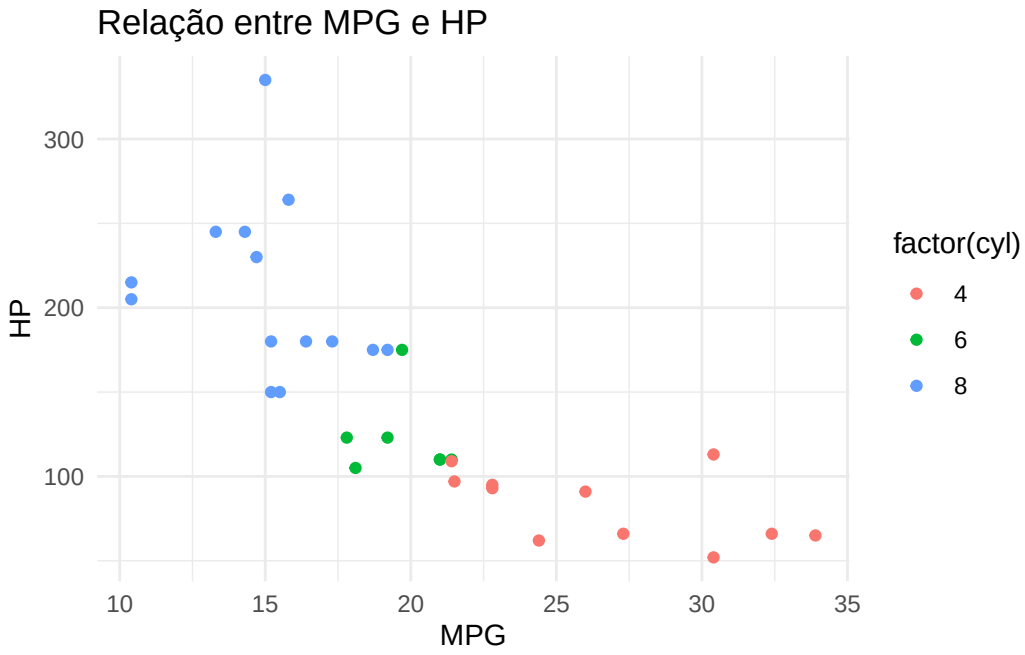
ggplot2 - Elementos da gramática

```r
# Criar o gráfico
ggplot(mtcars, aes(x = mpg, y = hp)) +
    geom_point() +
    scale_y_log10() +
    facet_wrap(~am) + # cria uma faceta para cada valor único da variável am (transmissão ma
    stat_summary(fun = mean, geom = "point", color = "red", size = 3) +
    theme_minimal() +
    labs(
        title = "Relação entre MPG e HP",
        x = "MPG",
        y = "HP (escala logarítmica)",
        color = "Número de cilindros"
    )
```

Gramática dos gráficos e ggplot2 - cores

O argumento color mapeia uma variável categórica para as cores dos pontos

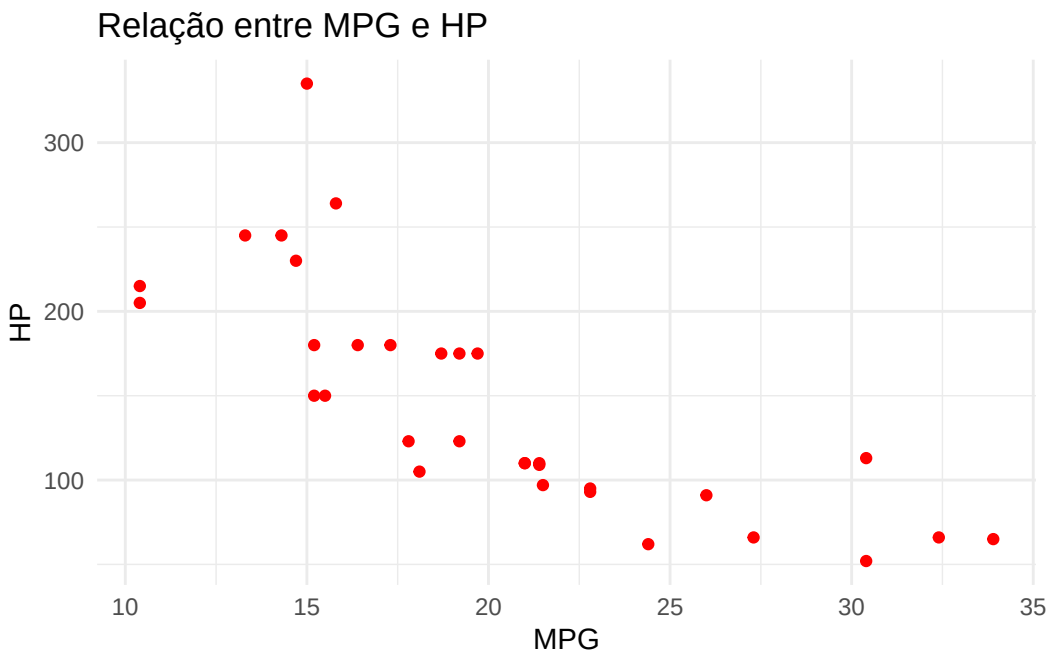
```
ggplot(data, aes(x = mpg, y = hp, color = factor(cyl))) +
    geom_point() +
    labs(title = "Relação entre MPG e HP", x = "MPG", y = "HP") +
    theme_minimal()
```



Gramática dos gráficos e ggplot2 - cores

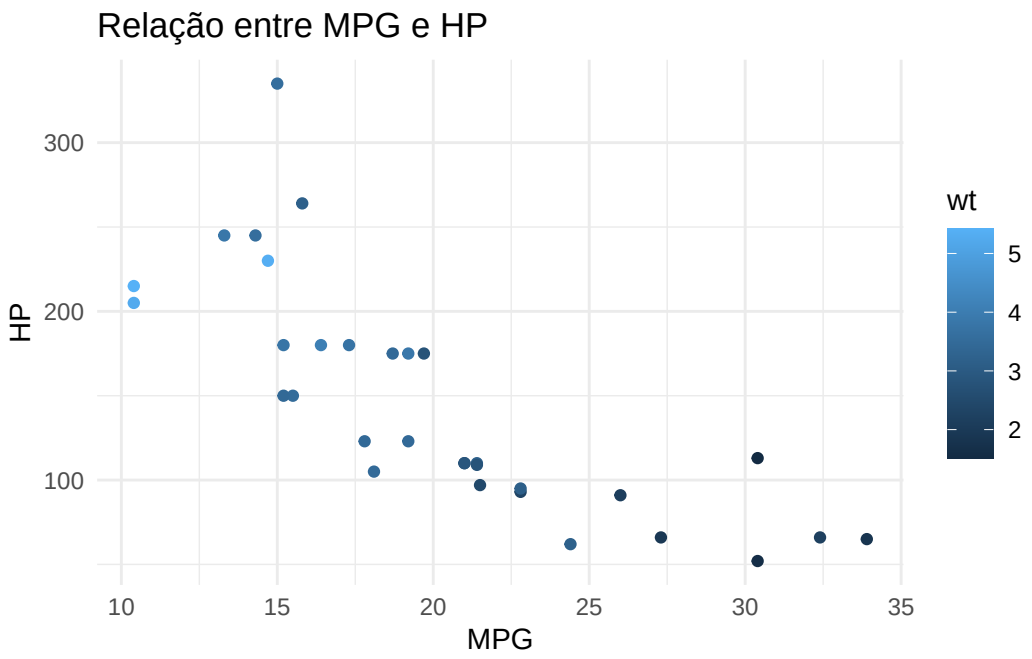
Se quisermos definir uma cor especifica temos que tomar cuidado com a localização do argumento color, por exemplo, se quisermos que todos os pontos sejam vermelhos, devemos colocar o argumento color dentro da função geom_point mas fora da função aes. Mas se quisermos mapear uma variável contínua para as cores dos pontos, devemos colocar o argumento color dentro da função aes.

```
ggplot(data, aes(x = mpg, y = hp)) +
    geom_point(color = "red") +
    labs(title = "Relação entre MPG e HP", x = "MPG", y = "HP") +
    theme_minimal()
```



Gramática dos gráficos e ggplot2 - cores

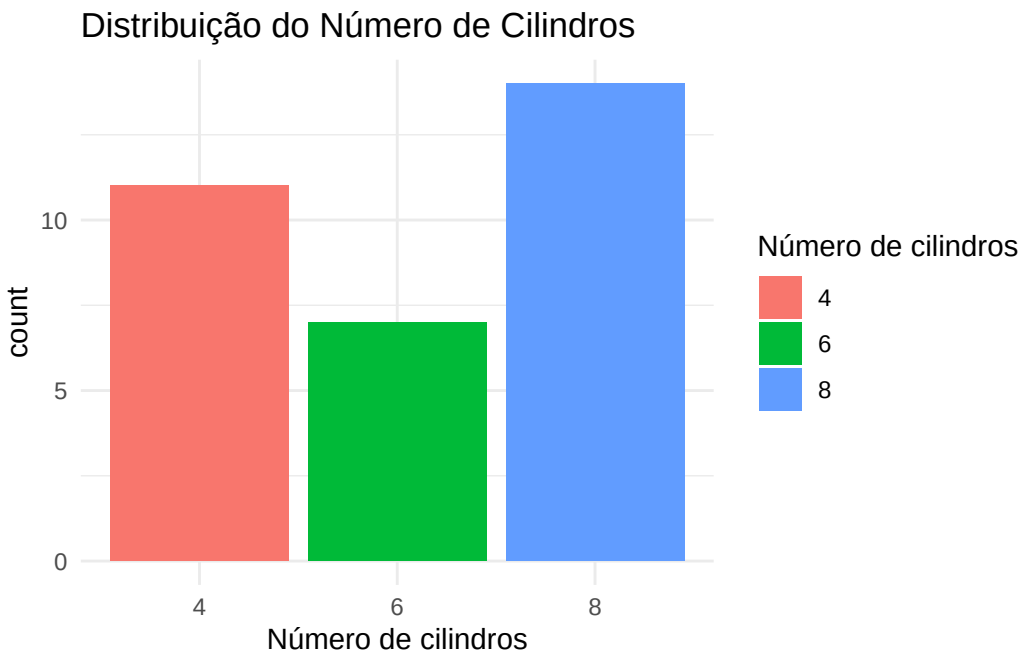
```
ggplot(data, aes(x = mpg, y = hp, color = wt)) +
    geom_point() +
    labs(title = "Relação entre MPG e HP", x = "MPG", y = "HP") +
    theme_minimal()
```



Gramática dos gráficos e ggplot2 - fill

O argumento fill mapeia uma variável categórica para o preenchimento das barras, a diferença entre color e fill é que color mapeia para a cor da borda e fill mapeia para o preenchimento.

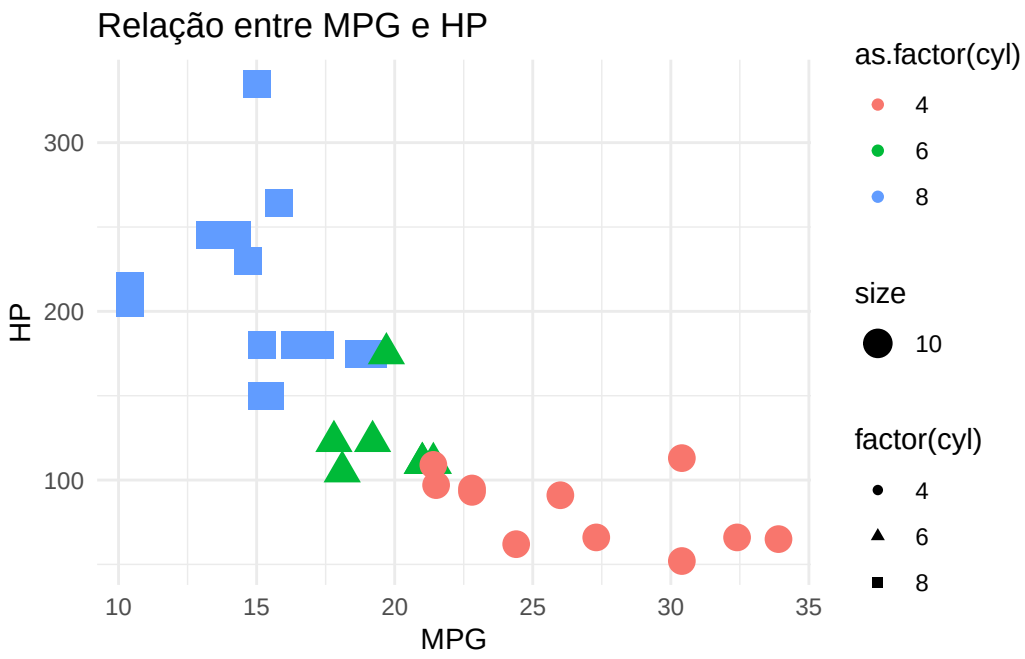
```
ggplot(mtcars, aes(x = as.factor(cyl), fill = as.factor(cyl))) +
    geom_bar() +
    labs(
        title = "Distribuição do Número de Cilindros",
        x = "Número de cilindros",
        fill = "Número de cilindros"
    ) +
    theme_minimal()
```



Gramática dos gráficos e ggplot2 - formas

Exemplo

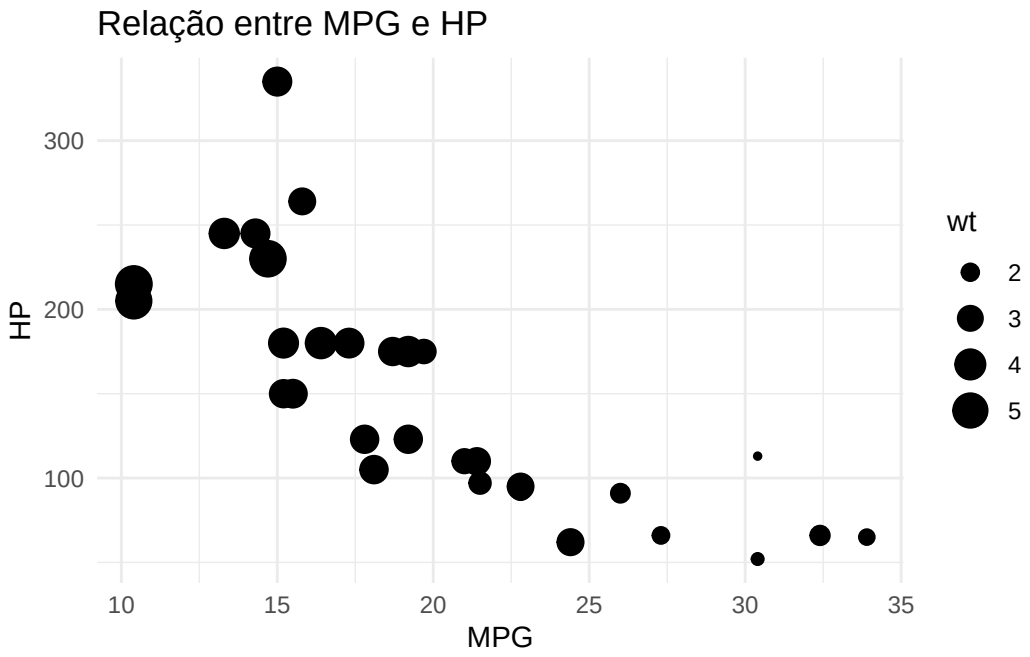
```
ggplot(data, aes(x = mpg, y = hp, shape = factor(cyl), size = 10, color = as.factor(cyl))) +
    geom_point() +
    labs(title = "Relação entre MPG e HP", x = "MPG", y = "HP") +
    theme_minimal()
```



Gramática dos gráficos e ggplot2 - tamanhos

Exemplo

```
ggplot(data, aes(x = mpg, y = hp, size = wt)) +
    geom_point() +
    labs(title = "Relação entre MPG e HP", x = "MPG", y = "HP") +
    theme_minimal()
```



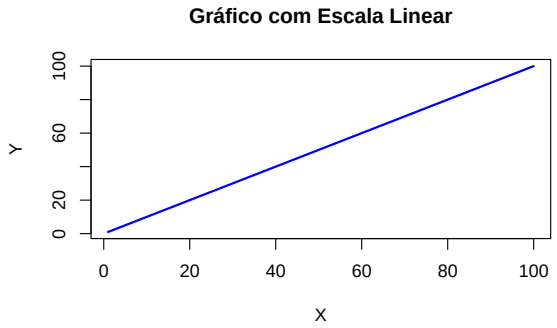
Exercícios de fixação 3

1. Refaça os gráficos dos exercícios de fixação 1 e 2 utilizando o pacote ggplot2 e a gramática dos gráficos. Comente sobre as diferenças entre os gráficos gerados com o pacote base do R e com o pacote ggplot2.
2. Explore os geoms disponíveis no ggplot2 e crie um gráfico usando o pacote dados e o geom de sua escolha.

Escalas Lineares e Logarítmicas

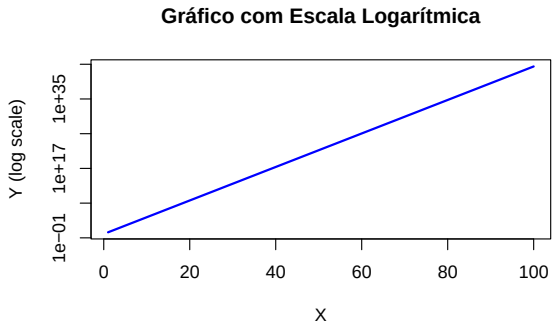
```r
x <- seq(1, 100, 1)
y <- x

plot(x, y, type = "l", col = "blue", lwd = 2, xlab = "X", ylab = "Y", main = "Gráfico com Es
```

Escalas Lineares e Logarítmicas



- Escalas lineares são as mais comuns
- Representam a relação entre duas variáveis de forma proporcional
- Dados variam de forma linear

```r
# Dados
x <- seq(1, 100, 1)
y <- exp(x)

# Gráfico com escala logarítmica no eixo y
plot(x, y,
    type = "l", col = "blue", lwd = 2,
    xlab = "X", ylab = "Y (log scale)",
    main = "Gráfico com Escala Logarítmica",
    log = "y"
)
```



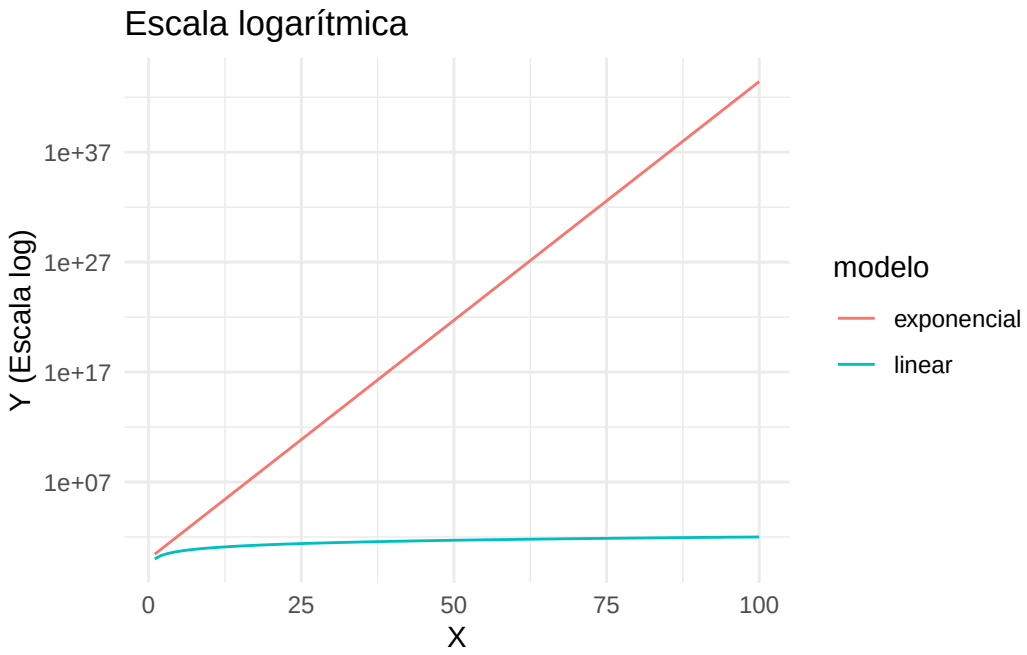
Escalas logarítmicas são usadas quando os dados variam de forma exponencial

Escalas Lineares e Logarítmicas

Quando precisamos representar variáveis em escalas diferentes, podemos usar a função scale_x_log10() ou scale_y_log10() para poder visualizar os dados de forma mais clara. No

exemplo abaixo queremos comparar duas regressões que tem ordem de grandeza diferente, para isso usamos a escala logarítmica no eixo y.

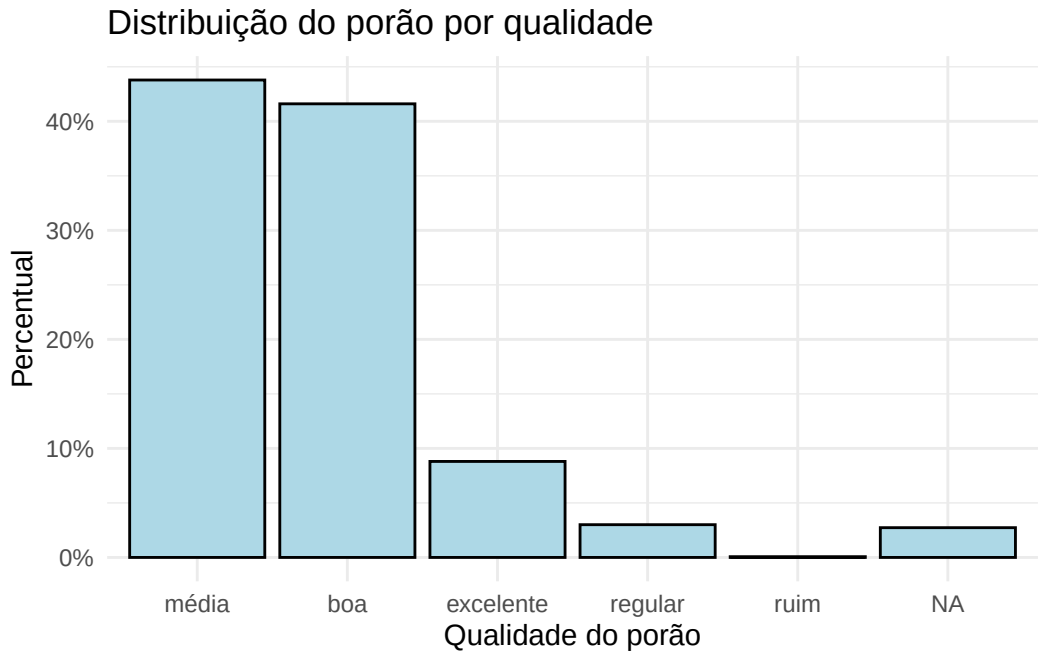
```
x <- seq(1, 100, 1)

y1 <- x

y2 <- exp(x)

# Gráfico com escala logarítmica no eixo y

df <- data.frame(x = rep(x, 2), y = c(y1, y2), modelo = rep(c("linear", "exponencial"), each

ggplot(df, aes(x = x, y = y, color = modelo)) +
    geom_line() +
    scale_y_log10() +
    labs(title = "Escala logarítmica", x = "X", y = "Y (Escala log)") +
    theme_minimal()
```



Exemplo de gráfico de barras com escala percentual para variável categórica

Usando o conjunto de dados casas, vamos criar um gráfico de barras para visualizar a distribuição da variável categórica "porao_qualidade" e adicionar rótulos aos eixos de forma que a escala seja percentual.

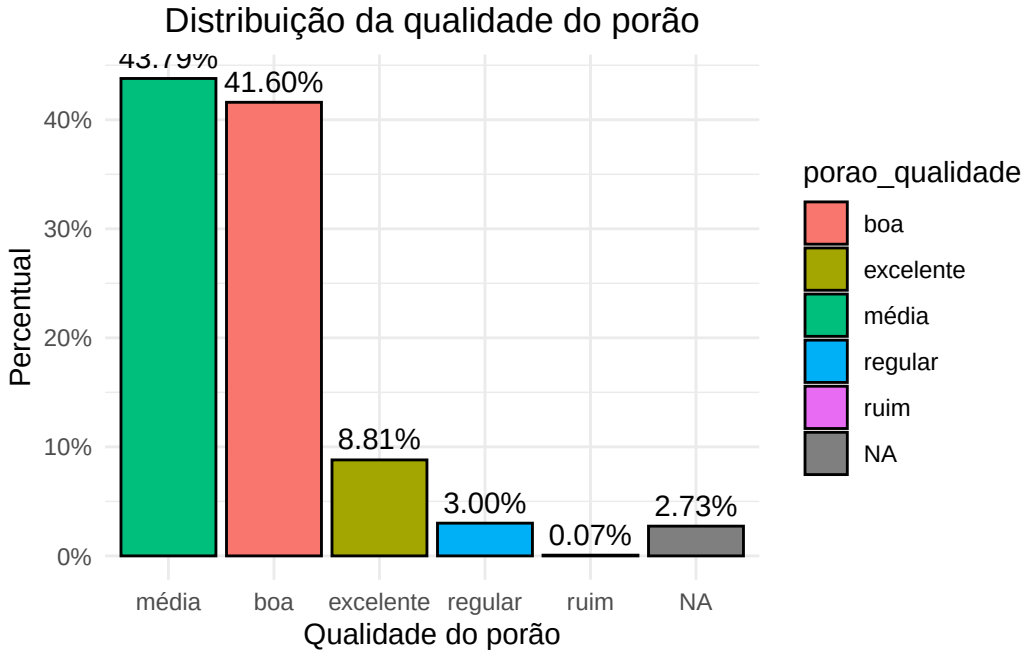
```
#| echo: true

casas %>%
    count(porao_qualidade) %>%
    mutate(percentual = n / sum(n)) %>% # Calcula a frequência relativa
    ggplot(aes(x = reorder(porao_qualidade, -percentual), y = percentual)) + # Reordena as ba
    geom_bar(stat = "identity", fill = "lightblue", color = "black") +
    labs(title = "Distribuição do porão por qualidade", x = "Qualidade do porão", y = "Percen
    scale_y_continuous(labels = scales::percent) + # Define a escala percentual no eixo y
    theme_minimal()
```



Exemplo de gráfico de barras com escala percentual para variável categórica

Também podemos adicionar rótulos para cada barra com a função geom_text

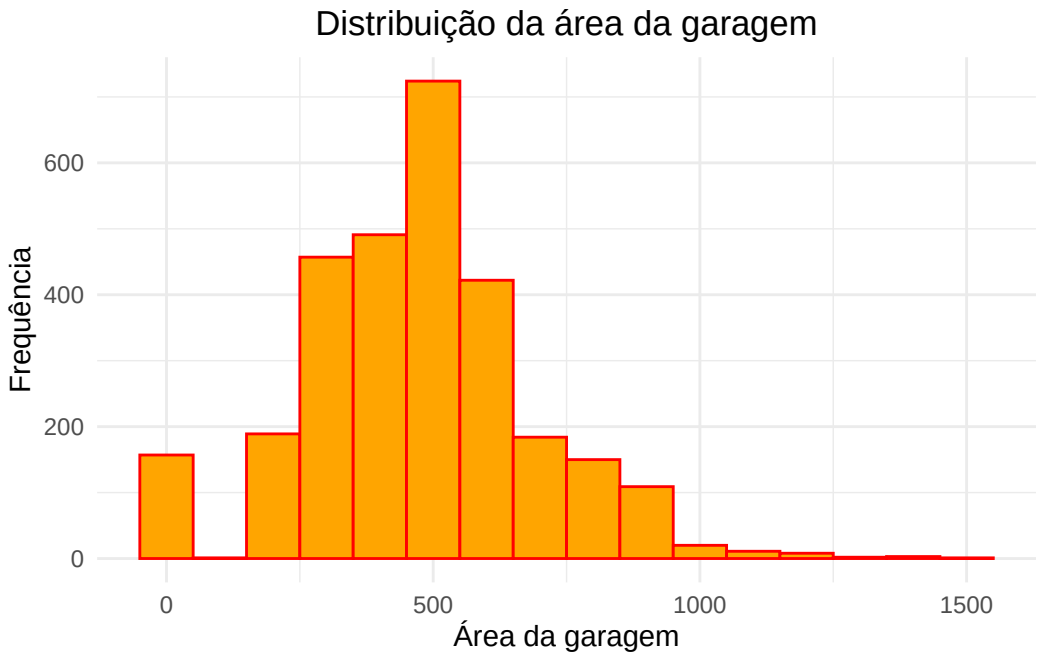
```r
casas %>%
    count(porao_qualidade) %>%
    mutate(percentual = n / sum(n)) %>%
    ggplot(aes(x = reorder(porao_qualidade, -percentual), y = percentual, fill = porao_qualid
    geom_bar(stat = "identity", color = "black") +
    labs(title = "Distribuição da qualidade do porão", x = "Qualidade do porão", y = "Percent
    scale_y_continuous(labels = scales::percent) +
    geom_text(aes(label = scales::percent(percentual)), vjust = -0.5) + # Adiciona rótulos e
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5)) # Centraliza o título
```



Exemplo de gráfico de histograma no ggplot2

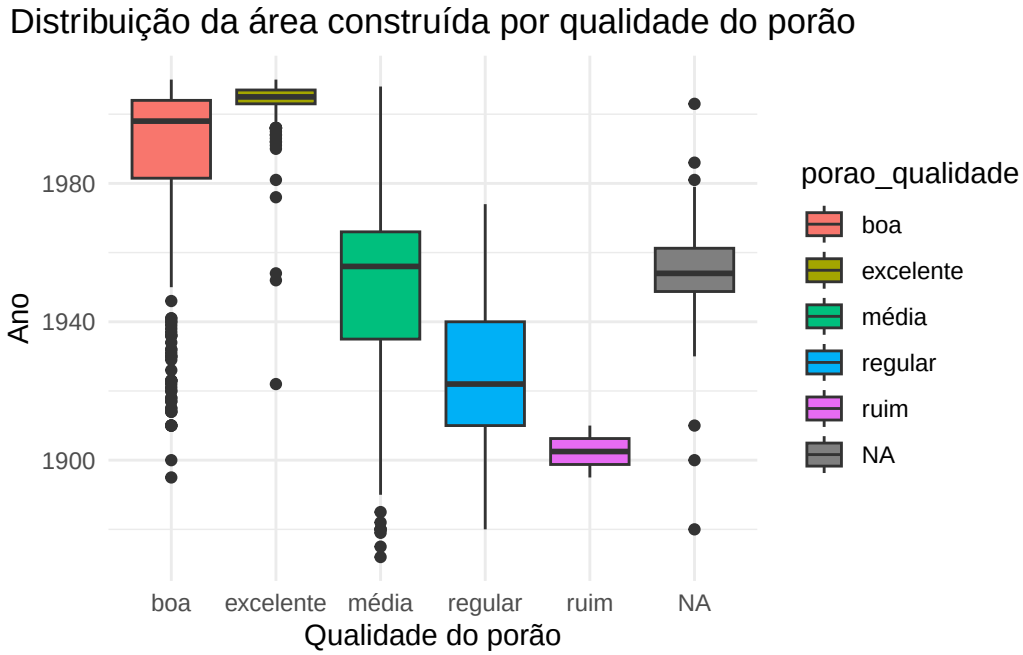
```r
casas %>%
    ggplot(aes(x = garagem_area)) +
    geom_histogram(binwidth = 100, fill = "orange", color = "red") + # binwidth define a larg
    labs(title = "Distribuição da área da garagem", x = "Área da garagem", y = "Frequência")
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5)) # Centraliza o título
```

```
Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```



Exemplo de boxplot no ggplot2 com o dataset casas

```
casas %>%
    ggplot(aes(x = porao_qualidade, y = construcao_ano, fill = porao_qualidade)) +
    geom_boxplot() +
    labs(title = "Distribuição da área construída por qualidade do porão", x = "Qualidade do
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5)) # Centraliza o título
```



Explorando dados com o pacote esquisse

O pacote esquisse é uma ferramenta interativa para explorar e visualizar dados. podemos usá-lo para criar gráficos rapidamente e explorar a relação entre variáveis.

```
# Instalar o pacote

install.packages("esquisse")

library(esquisse)

esquisse::esquisser() # use :: para chamar a função esquisser do pacote esquisse sem precisar
```

Exercícios de fixação 4

Utilize o pacote esquisse para explorar o conjunto de dados casas. Crie dois gráficos, um explorando variáveis categóricas e outro explorando variáveis contínuas. Apresente os gráficos e comente sobre as relações observadas.

Tarefa final

Procure um conjunto de dados do seu interesse e crie um relatório exploratório dos dados utilizando o pacote esquisse. O relatório deve conter: um gráfico de barras, um histograma, um boxplot e um gráfico de dispersão. Comente sobre as relações observadas nos gráficos.

Sugestões :

Caso não tenha um conjunto de dados específico em mente, você pode utilizar um dos conjuntos de dados disponíveis no pacote 'dados' ou no kaggle.

O relatório deve ser entregue em um arquivo qmd ou rmd.