Estatística descritiva com aplicação em Saúde usando **R**

Exploração e visualização de Dados

Profa Carolina e Prof Gilberto

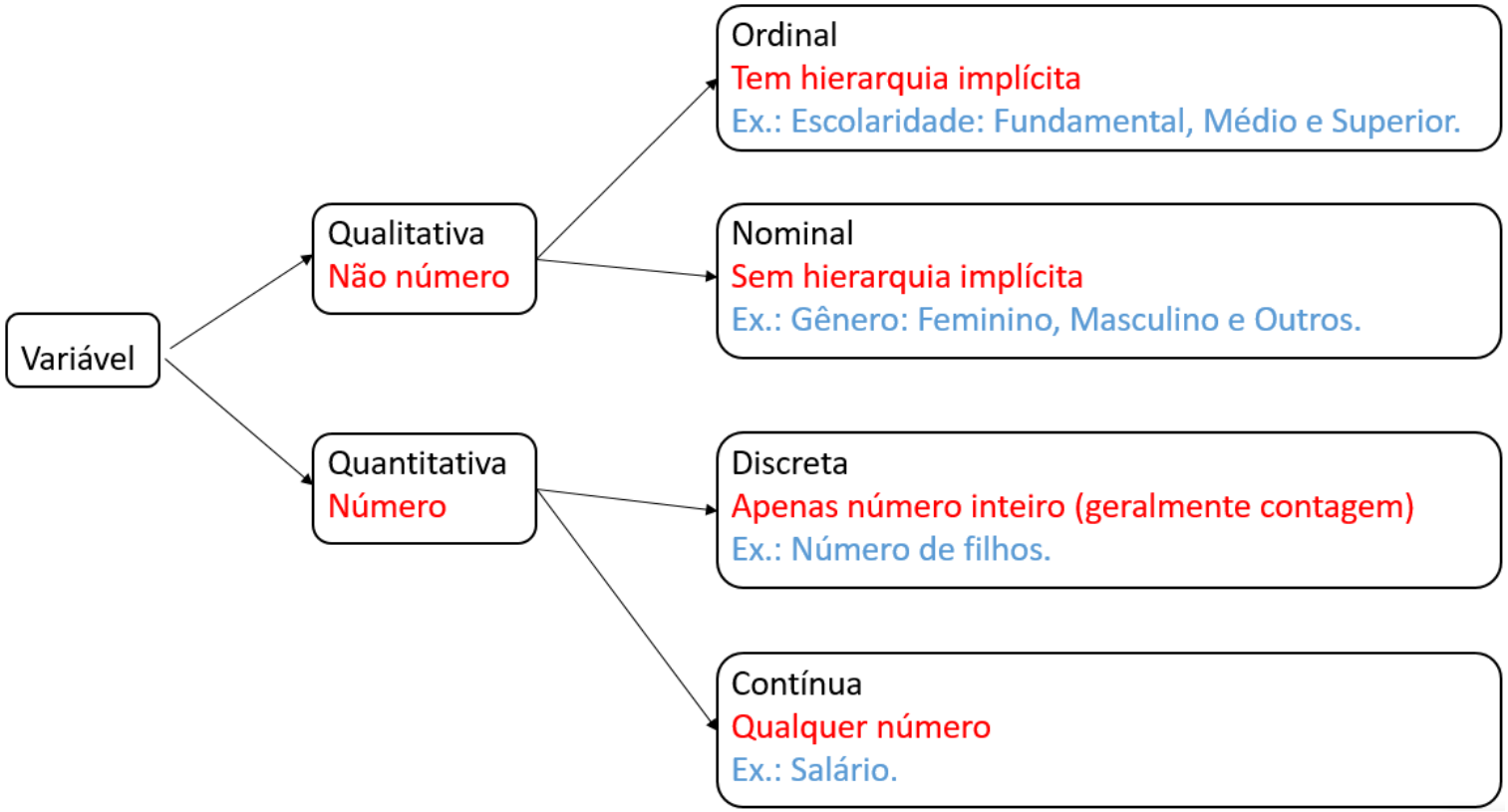
Parte 2

Conceitos básicos

Começamos com alguns conceitos básicos, que usaremos durante todo esse curso.

- **População**: Todos os elementos ou indivíduos alvo do estudo;
- **Amostra**: Parte da população;
- **Parâmetro:** característica da população (grandeza);
- **Estimativa:** característica da amostra. Usamos a estimativa para aproximar o parâmetro;
- **Variável:** *característica de um elemento da população (idade ou peso).* Geralmente usamos uma letra maiúscula do alfabeto latino para representar uma variável (idade ou peso), e uma letra minúscula do alfabeto latino para representar o valor de uma variável para um elemento da população. Por exemplo, podemos representar *a variável "idade" por* X e um *x=30 é idade de um elemento da amostra*.

Classificação de variáveis



Classificação de variáveis.

Tabela de distribuição de frequência variável qualitativa

A primeira coisa que fazemos é contar!

X	frequência	frequência relativa	porcentagem
B_1	n_1	f_1	$100 \cdot f_1\%$
B_2	n_2	f_2	$100 \cdot f_2\%$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_k	n_k	f_k	$100 \cdot f_k\%$
Total	n	1	100%

Em que n é o tamanho da amostra.

Geralmente não incluímos a coluna de *frequência relativa*.

(Re)codificação de variáveis

Precisamos usar o dicionário de variáveis para (re)codificar as variáveis.

```
df_base_raw <- read_xlsx("data/raw/base_has.xlsx", na = "NA")
df_base_processed <- df_base_raw |>
  mutate(sex = recode(sex, `1` = "M", `2` = "F"),
      racacor = recode(racacor, `1` = "Branca", `2` = "Preta",
                       `3` = "Indígena", `4` = "Parda"),
      esc = recode(esc, `1` = "Fundamental incompleto", `2` = "Médio completo",
                   `3` = "Superior incompleto", `4` = "Superior completo"))
write_xlsx(df_base_processed, "data/processed/base_has_processed.xlsx")
df_base_processed |> select(racacor, sex, esc) |> head(n = 4)
```

```
## # A tibble: 4 × 3
##   racacor sex   esc
##   <chr>   <chr> <chr>
## 1 Parda   M     Superior completo
## 2 Preta   M     Superior completo
## 3 Branca  M     Superior incompleto
## 4 Branca  F     <NA>
```

Tabela de distribuição de frequência variável qualitativa

Pacotes: `janitor`

```
library(janitor)
df_base_processed |>
    tabyl(sex) |>
    arrange(desc(n)) |>
    adorn_totals() |>
    adorn_pct_formatting(digits = 2) |>
    rename(
        "Gênero" = sex,
        "Frequência" = n,
        "Porcentagem" = percent,
        "Porcentagem sem missing" = valid_percent
    )
```

```
##  Gênero Frequência Porcentagem Porcentagem sem missing
##       M        144      64.00%                  67.29%
##       F         70      31.11%                  32.71%
##    <NA>         11       4.89%                      -
##   Total        225     100.00%                 100.00%
```

Tabela de distribuição de frequência variável quantitativa discreta

Pacotes: `janitor`

```
library(janitor)
df_mtcarros <- read_csv2("data/raw/mtcarros.csv")
df_mtcarros |>
    tabyl(marchas) |>
    arrange(desc(n)) |>
    adorn_totals() |>
    adorn_pct_formatting(digits = 2) |>
    rename(
        "Marchas" = marchas,
        "Frequência" = n,
        "Porcentagem" = percent
    )
```

```
##  Marchas Frequência Porcentagem
##        3         15      46.88%
##        4         12      37.50%
##        5          5      15.62%
##    Total         32     100.00%
```

Tabela de distribuição de frequência variável quantitativa contínua

Primeiro agregamos os valores em intervalos.

1. Usamos intervalos usados na área de saúde (outro artigo, por exemplo)
2. Regra de Sturge: $\lceil 1 + \log_2(n) \rceil$ (n é o tamanho da amostra)

```
k <- round(1 + log2(nrow(df_base_processed)))
faixas <- seq(
    from = min(df_base_processed$ida, na.rm = TRUE),
    to = max(df_base_processed$ida, na.rm = TRUE),
    length.out = k
)
df_base_processed <- df_base_processed |>
    mutate(faixa_idade = cut(
        ida,
        breaks = faixas,
        include.lowest = T,
        right = F
    ))
```

Tabela de distribuição de frequência variável quantitativa contínua

```
df_base_processed |>
    tabyl(faixa_idade) |>
    adorn_totals() |>
    adorn_pct_formatting(digits = 2) |>
    rename(
        "Idade por intervalos" = faixa_idade,
        "Frequência Absoluta" = n,
        "Porcentagem" = percent,
        "Porcentagem sem missing" = valid_percent
    )
```

Idade por intervalos	Frequência Absoluta	Porcentagem	Porcentagem sem missing
[18,25.4)	43	19.11%	19.72%
[25.4,32.8)	56	24.89%	25.69%
[32.8,40.1)	51	22.67%	23.39%
[40.1,47.5)	33	14.67%	15.14%
[47.5,54.9)	15	6.67%	6.88%
[54.9,62.2)	15	6.67%	6.88%
[62.2,69.6)	2	0.89%	0.92%
[69.6,77]	3	1.33%	1.38%
<NA>	7	3.11%	-
Total	225	100.00%	100.00%

Medidas de Resumo

Medidas de posição e dispersão

A ideia é encontrar um ou alguns valores que sintetizem todos os valores.

Medidas de posição (tendência central)

A ideia é encontrar um valor que representa *bem* todos os valores.

- **Média:** $\overline{x} = \dfrac{x_1 + \cdots + x_n}{n}$
- **Mediana:** valor que divide a sequência ordenada de valores em duas partes iguais.

Medidas de dispersão

A ideia é medir a homogeneidade dos valores.

- **Variância:** $s^2 = \dfrac{(x_1 - \overline{X})^2 + \cdots + (x_n - \overline{X})^2}{n-1}$;
- **Desvio padrão:** $s = \sqrt{s^2}$ (mesma unidade dos dados);
- **coeficiente de variação** $cv = \dfrac{s}{\overline{x}} \cdot 100\%$ (adimensional, ou seja, “sem unidade”)

Medidas de resumo

Pacote: `dplyr`

```
df_base_processed |>
    group_by(racacor) |>
    summarise(
        media = mean(imc),
        mediana = median(imc),
        variancia = var(imc),
        dp = sd(imc),
        cv = dp / media,
        frequencia = n()
    )
```

```
## # A tibble: 5 × 7
##   racacor  media mediana variancia    dp     cv frequencia
##   <chr>    <dbl>   <dbl>     <dbl> <dbl>  <dbl>      <int>
## 1 Branca    27.1    27.1      30.4  5.51  0.204        109
## 2 Indígena  24.2    24.2      NA   NA    NA             1
## 3 Parda     27.4    28.8      21.8  4.67  0.171         60
## 4 Preta     27.4    28.5      25.5  5.05  0.184         52
## 5 <NA>      25.4    24.8      19.4  4.40  0.173          3
```

Quantis

Ideia

$q(p)$ é um valor que satisfaz:

- $100 \cdot p\%$ das observações é no máximo $q(p)$
- $100 \cdot (1-p)\%$ das observações é no mínimo $q(p)$

Alguns quantis especiais

- **Primeiro quartil:** $q_1 = q\left(\frac{1}{4}\right)$
- **Segundo quartil:** $q_2 = q\left(\frac{2}{4}\right)$
- **Terceiro quartil:** $q_3 = q\left(\frac{3}{4}\right)$

Quantis

```
df_base_processed |>
    group_by(sex) |>
    summarise(
        q1 = quantile(imc, 0.25),
        q2 = quantile(imc, 0.5),
        q3 = quantile(imc, 0.75),
        frequencia = n()
    )
```

```
## # A tibble: 3 × 5
##   sex      q1    q2    q3 frequencia
##   <chr> <dbl> <dbl> <dbl>      <int>
## 1 F      21.9  29.1  31.1         70
## 2 M      21.6  27.2  31.1        144
## 3 <NA>   24.8  29.1  31.5         11
```

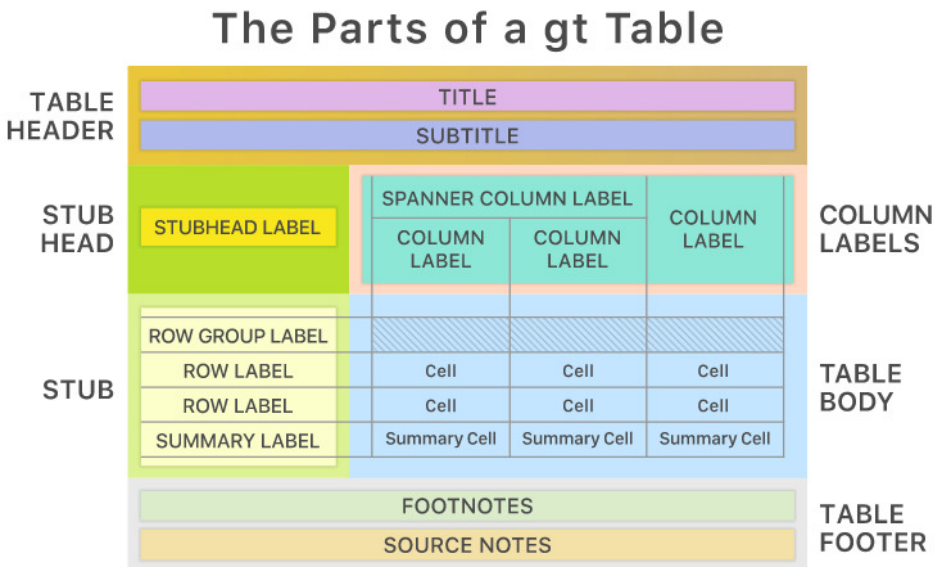
Exportando tabelas

pacote **gt**

Pacote `gt`

Vamos usar o pacote `gt` para customizar a apresentação de uma tabela.

A ideia do pacote `gt` é melhorar apresentação por camadas.



Para mais detalhes, visite `pacote gt`.

Exemplo

Vamos customizar e salvar a tabela com as medidas de resumo para a variável `imc` do conjunto de dados *base_.xlsx*.

```
tab <- df_base_processed |>
    filter(!is.na(racacor) & !is.na(imc) & racacor != "Indígena") |>
    group_by(racacor) |>
    summarise(
        media = mean(imc),
        mediana = median(imc),
        dp = sd(imc),
        cv = dp * 100 / media,
        q1 = quantile(imc, 1 / 4),
        q3 = quantile(imc, 3 / 4)
    )
```

Cabeçalho e subcabeçalho

- `tab_header`: permite incluir cabeçalho (`title`) e subcabeçalho (`subtitle`)
- `gtsave`: permite salvar tabela em formato `html` (página web), `tex` ($\LaTeX$) e `rtf` (word)

```
library(gt)
gt_tab <- gt(tab) |>
  tab_header(
    title = md("**IMC**"),
    subtitle = md("**por raça autodeclarada**")
  )
gtsave(gt_tab, filename = "output/gt_tab.html")
gtsave(gt_tab, filename = "output/gt_tab.tex")
gtsave(gt_tab, filename = "output/gt_tab.rtf")
```

Incluindo fonte dos dados

- `tab_source_note`: inclusão de *fonte de dados*
- `md`: formatação de texto usando a sintaxe `markdown`
- `html`: formatação de texto usando sintaxe `html`

```
gt_tab <- gt_tab |>
  tab_source_note(
    source_note = md("**Fonte:** Elaboração própria.")
  )
gt_tab
```

IMC

por raça autodeclarada

racacor	media	mediana	dp	cv	q1	q3
Branca	27.05990	27.14681	5.509258	20.35949	21.38976	31.02041
Parda	27.36131	28.76630	4.669262	17.06520	22.01759	31.27128
Preta	27.43333	28.49819	5.051074	18.41218	23.40553	31.39357

Fonte: Elaboração própria.

Rótulo para grupo de linhas

`tab_row_group`: permite colocar *rótulo* para um grupo de linhas

```
gt_tab <- gt_tab |>
  tab_row_group(
    label = md("**Pessoas negras**"),
    rows = c(2, 3)
  )
gt_tab
```

IMC

por raça autodeclarada

racacor	media	mediana	dp	cv	q1	q3
Pessoas negras						
Parda	27.36131	28.76630	4.669262	17.06520	22.01759	31.27128
Preta	27.43333	28.49819	5.051074	18.41218	23.40553	31.39357
Branca	27.05990	27.14681	5.509258	20.35949	21.38976	31.02041

Fonte: Elaboração própria.

Rótulo para grupo de colunas

`tab_spanner`: permite colocar *rótulo* para grupo de colunas

```
gt_tab <- gt_tab |>
  tab_spanner(
    label = html("<strong>Quantis</strong>"),
    columns = c(q1, mediana, q3)
  )
gt_tab
```

IMC

por raça autodeclarada

				Quantis		
racacor	media	dp	cv	q1	mediana	q3
Pessoas negras						
Parda	27.36131	4.669262	17.06520	22.01759	28.76630	31.27128
Preta	27.43333	5.051074	18.41218	23.40553	28.49819	31.39357
Branca	27.05990	5.509258	20.35949	21.38976	27.14681	31.02041

Fonte: Elaboração própria.

Movendo colunas

- `col_move_to_start`: move uma ou mais colunas para o início da tabela
- `col_move_to_end`: move uma ou mais colunas para o fim da tabela
- `col_move`: move uma coluna ou mais colunas depois uma determinada coluna

```
gt_tab <- gt_tab |>
    cols_move_to_start(
        columns = racacor
    ) |>
    cols_move_to_end(
        columns = c(q1, mediana, q3)
    ) |>
    cols_move(
        columns = cv,
        after = media
    )
gt_tab
```

IMC

por raça autodeclarada

				Quantis		
racacor	media	cv	dp	q1	mediana	q3
Pessoas negras						
Parda	27.36131	17.06520	4.669262	22.01759	28.76630	31.27128
Preta	27.43333	18.41218	5.051074	23.40553	28.49819	31.39357
Branca	27.05990	20.35949	5.509258	21.38976	27.14681	31.02041

Fonte: Elaboração própria.

Atualização dos rótulos das colunas

`cols_label`: permite atualizar os *rótulos* de colunas

```
gt_tab <- gt_tab |>
    cols_label(
        racacor = md("_Raça_"),
        media = html("<em>Média</em>"),
        cv = md("_Coeficiente de variação_"),
        dp = html("<em>Desvio padrão</em>"),
        q1 = md("_Primeiro quartil_"),
        mediana = html("<em>Segundo quartil</em>"),
        q3 = md("_Terceiro quartil_")
    )
gt_tab
```

IMC

por raça autodeclarada

				Quantis		
Raça	*Média*	*Coeficiente de variação*	*Desvio padrão*	*Primeiro quartil*	*Segundo quartil*	*Terceiro quartil*
Pessoas negras						
Parda	27.36131	17.06520	4.669262	22.01759	28.76630	31.27128
Preta	27.43333	18.41218	5.051074	23.40553	28.49819	31.39357
Branca	27.05990	20.35949	5.509258	21.38976	27.14681	31.02041

Fonte: Elaboração própria.

Formatação de valores nas colunas

`fmt_number:` formatação de valores numéricos em uma tabela

```
gt_tab <- gt_tab |>
    fmt_number(
        columns = c(media, mediana, dp, q1, q3),
        decimals = 2,
        dec_mark = ",",
        sep_mark = "."
    ) |>
    fmt_number(
        columns = cv,
        decimals = 2,
        dec_mark = ",",
        sep_mark = ".",
        pattern = "{x}%"
    )
gt_tab
```

IMC por raça autodeclarada						
				Quantis		
Raça	*Média*	*Coeficiente de variação*	*Desvio padrão*	*Primeiro quartil*	*Segundo quartil*	*Terceiro quartil*
Pessoas negras						
Parda	27,36	17,07%	4,67	22,02	28,77	31,27
Preta	27,43	18,41%	5,05	23,41	28,50	31,39
Branca	27,06	20,36%	5,51	21,39	27,15	31,02

Fonte: Elaboração própria.

Gráficos

`ggplot2`

Gráficos no **R**

- **Pacote:** `ggplot2`
- Permite gráficos personalizados com uma sintaxe simples e rápida, e iterativa *por camadas*
- Começamos com um camada com os dados `ggplot(dados)`, e vamos adicionando as camadas de anotações, e sumários estatísticos
- Usa a *gramática de gráficos* proposta por Leland Wilkinson: Grammar of Graphics
- Ideia desta gramática: delinear os atributos estéticos das figuras geométricas (incluindo transformações nos dados e mudança no sistema de coordenadas)
- Para mais detalhes, você pode consultar `ggplot2`: elegant graphics for data analysis e documentação do `ggplot2`

Estrutura básica `ggplot2`

```
#| eval: false
ggplot(data = <data possible tibble>) +
  <Geom functions>(mapping = aes(<MAPPINGS>)) +
  <outras camadas>
```

Você pode usar diversos temas e extensões que a comunidade cria e criou para melhorar a aparência e facilitar a construção de `ggplot2`.

- Lista com extensões do `ggplot`: extensões do `ggplots`

Indicação de extensões:

- Temas adicionais para o pacote `ggplot2`: `ggthemes`
- Gráfico de matriz de correlação: `ggcorrplot`
- Gráfico quantil-quantil: `qqplotr`

Gráfico de barras

Gráfico de Barras no `ggplot2`

- **função:** `geom_bar()`. Para porcentagem: `geom_bar(x = <variável no eixo x>, y = ..prop.. * 100)`.
- Argumentos adicionais:
  - **`fill`**: mudar a cor do preenchimento das figuras geométricas
  - **`color`**: mudar a cor da figura geométrica

Rótulos dos eixos

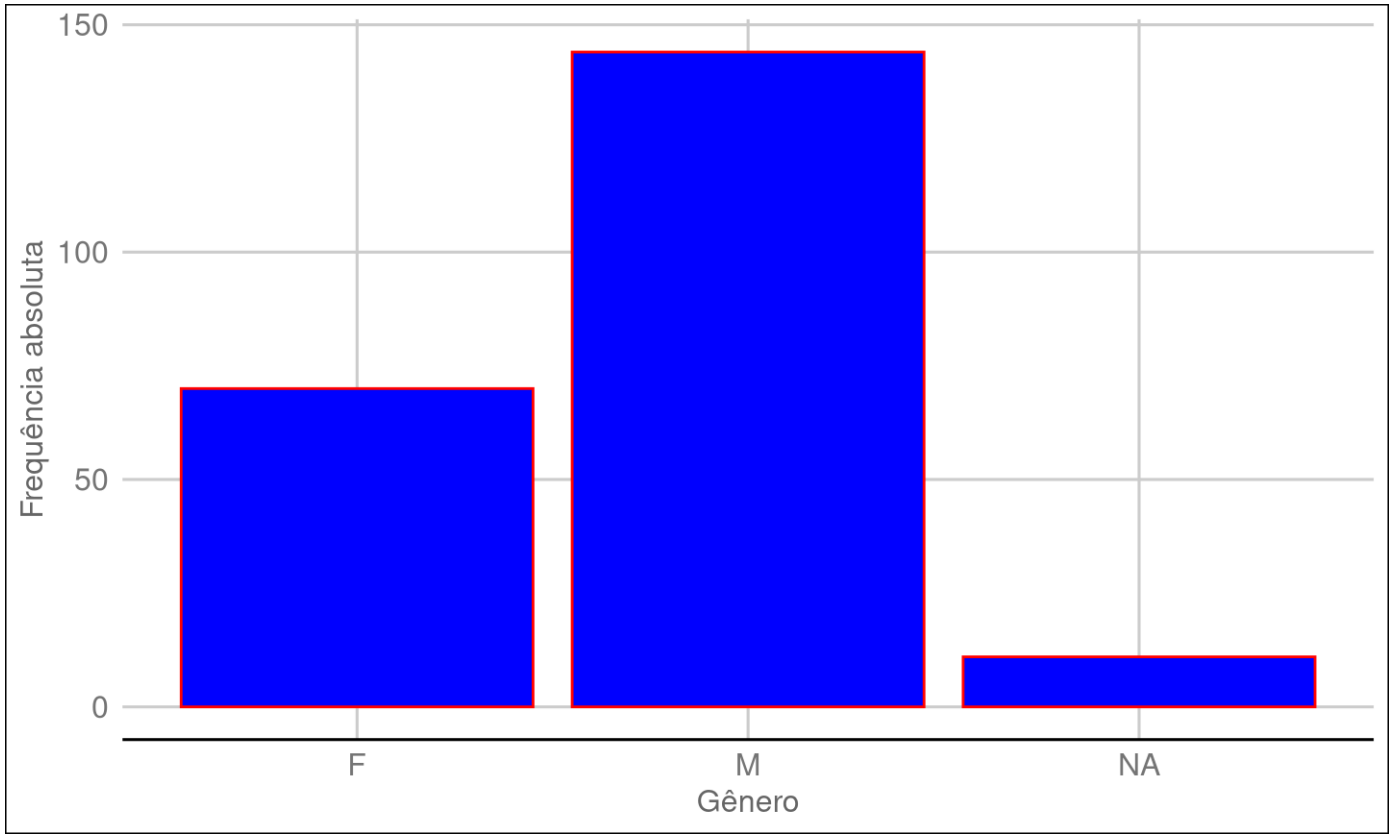
- **Mudar os rótulos:** `labs(x = <rótulo do eixo x>, y = <rótulo do eixo y>, title = <legenda do gráfico>)`
- **Trocar o eixo-x pelo eixo-y:** `coord_flip()`

Salvar gráficos

- `ggsave()`: salvar gráficos nos formatos `pdf`, `png` e `jpeg`

Gráfico de barras

```
library(ggplot2)
library(ggthemes)
ggplot(df_base_processed) +
    geom_bar(aes(x = sex), fill = "blue", color = "red") +
    labs(x = "Gênero", y = "Frequência absoluta") +
    theme_gdocs()
ggsave("figures/barras.jpeg")
ggsave("figures/barras.png")
ggsave("figures/barras.pdf")
```

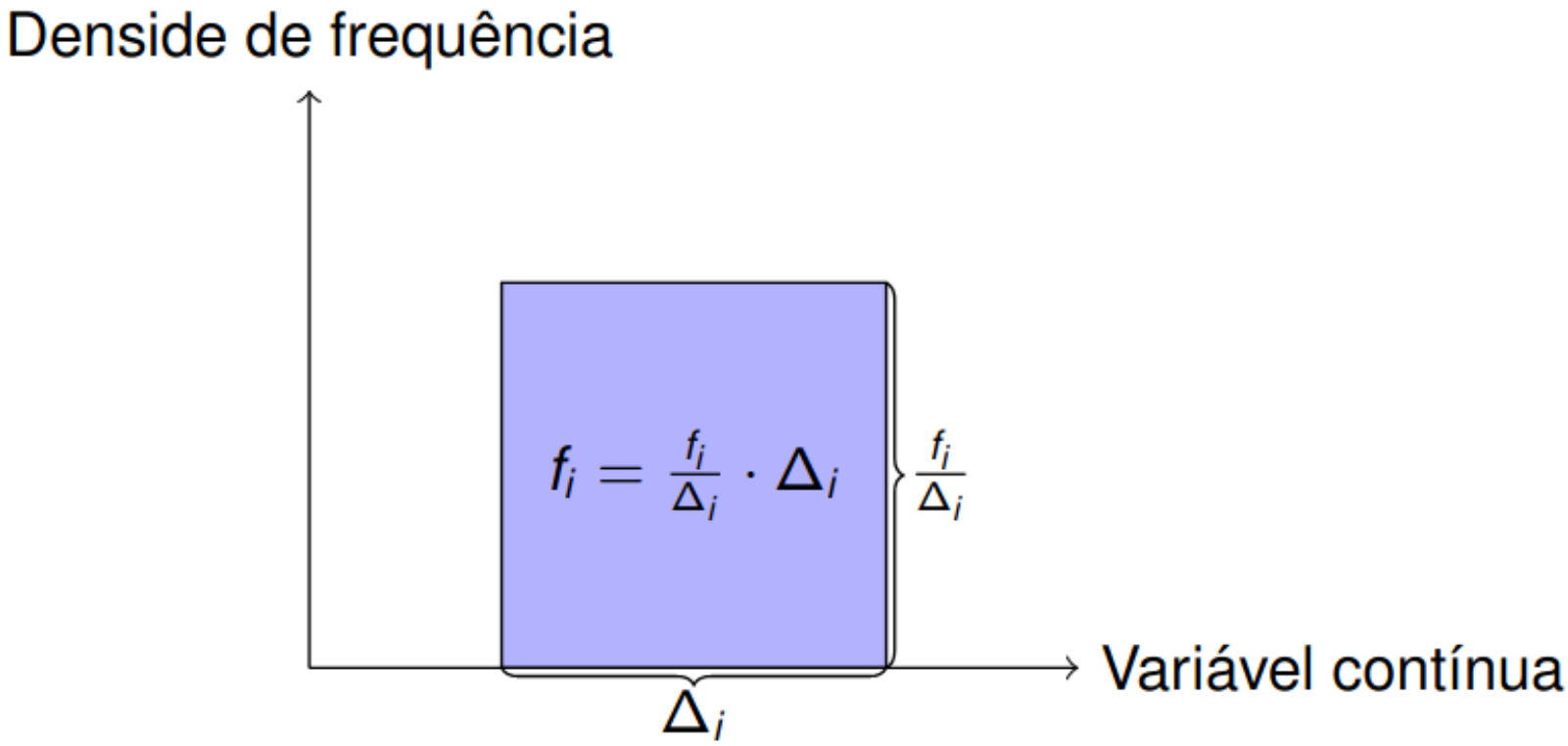

150
100
50
0
Frequência absoluta
F
M
NA
Gênero

Histograma

Para variávieis quantitativas contínuas usamos *histograma*.

- O histograma é um gráfico de barras contíguas em que a área de cada barra é igual à frequência relativa.
- Cada faixa de valor $[l_{i-1}, l_i), i = 1, \ldots, n,$ será representada por um barra com área $f_i, i = 1, \ldots, n$.
- Como cada barra terá área igual a f_i e base $\Delta_i = l_i - l_{i-1}$, e a altura de cada barra será $\frac{f_i}{l_i - l_{i-1}}$.
- $\frac{f_i}{l_i - l_{i-1}}$ é denominada de densidade de frequência.
- Podemos fornecer:
  - `bins`: número de intervalos
  - `binwidth`: tamanho dos intervalos
  - `breaks`: limites dos intervalos

Histograma


Denside de frequência
$f_i = \frac{f_i}{\Delta_i} \cdot \Delta_i$
$\frac{f_i}{\Delta_i}$
Variável contínua
Δ_i

Histograma

```
k <- round(1 + log2(nrow(df_base_processed)))

ggplot(df_base_processed) +
  geom_histogram(aes(x = imc, y = ..density..),
                 bins = k, fill = "blue", closed = "left") +
  theme_gdocs() +
  labs(
    x = "IMC",
    y = "Densidade de frequência",
    title = "Histograma"
  )
```

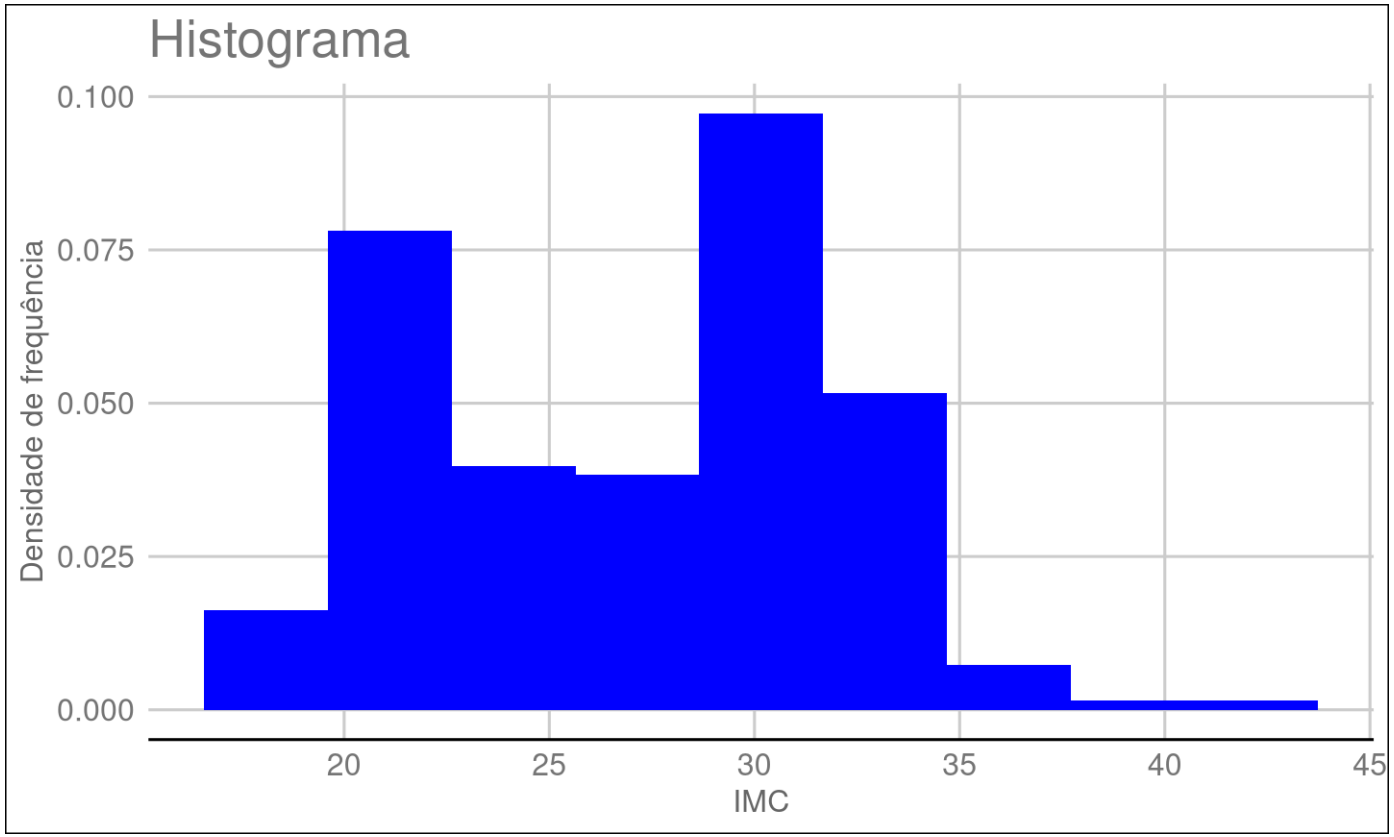

Histograma
0.100
0.075
0.050
0.025
0.000
Densidade de frequência
20
25
30
35
40
45
IMC

Diagrama de caixa

medida de dispersão: distância entre q_1 e q_3 pequena indica homogeneidade

Diferença de quartis: $dq = q_3 - q_1$

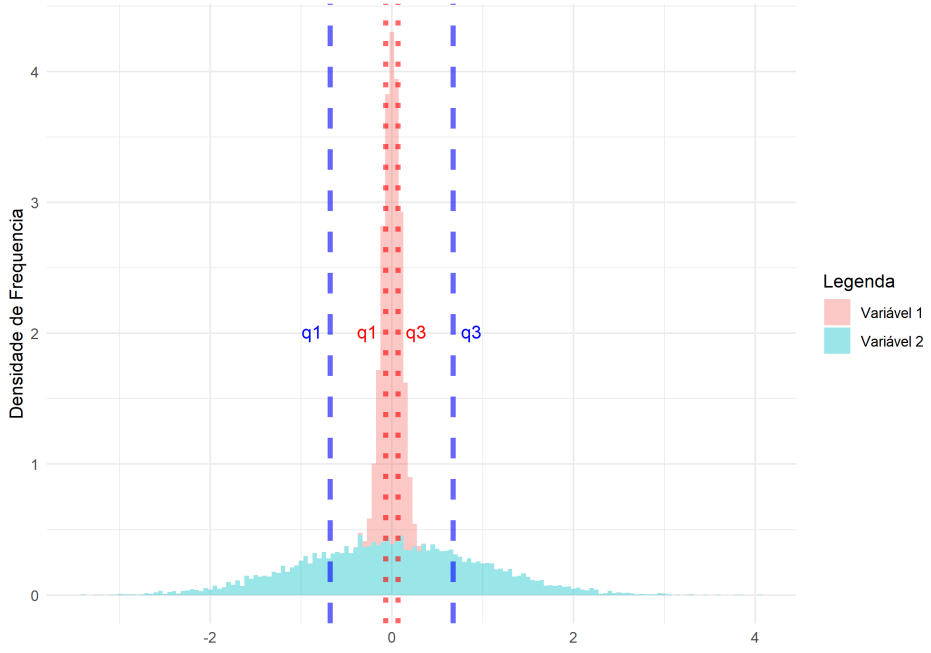


Diagrama de caixa

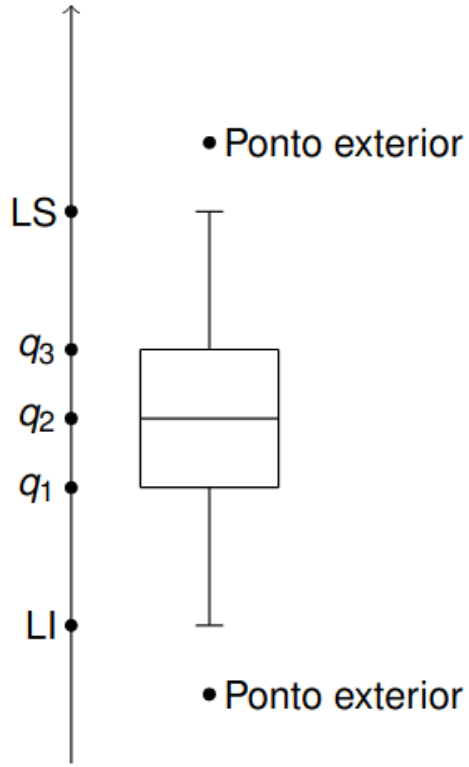


Diagrama de caixa

Assimetria

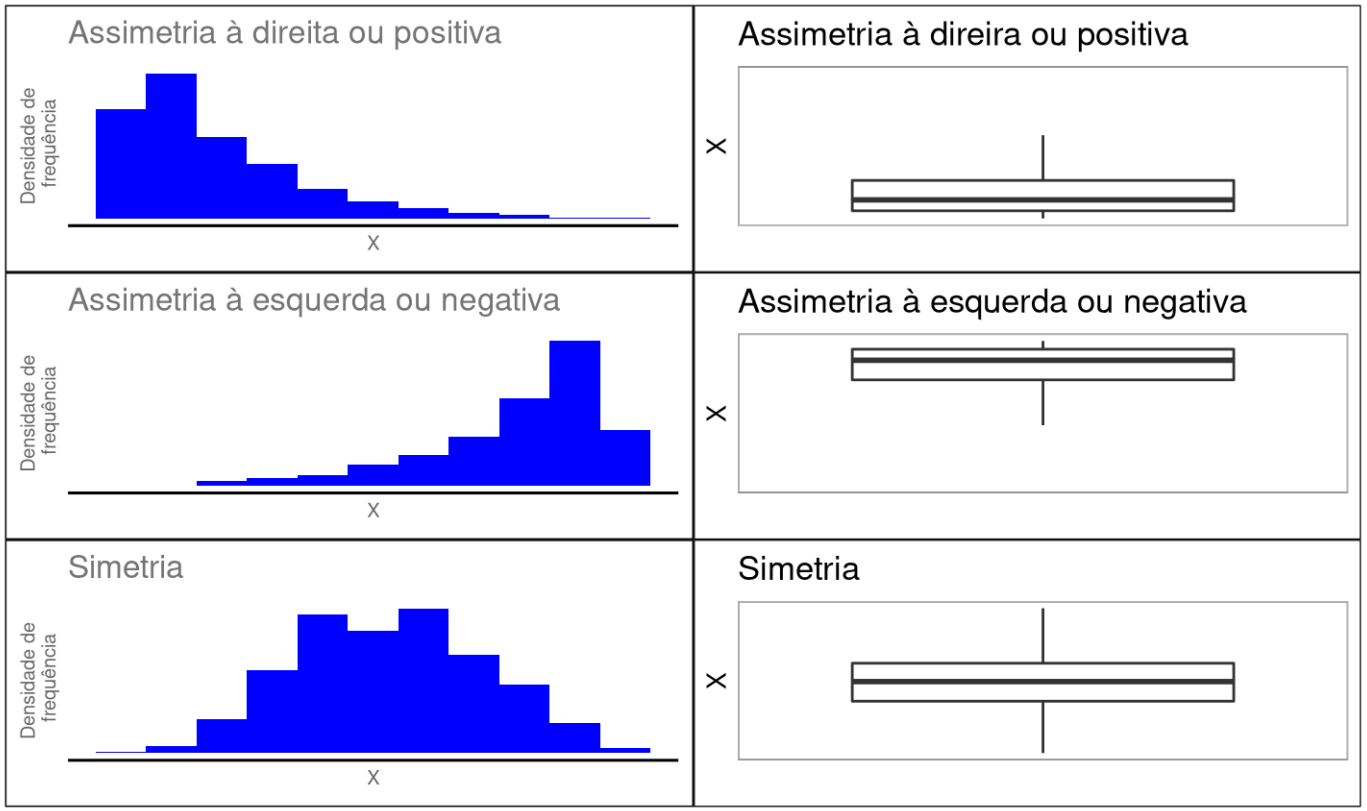
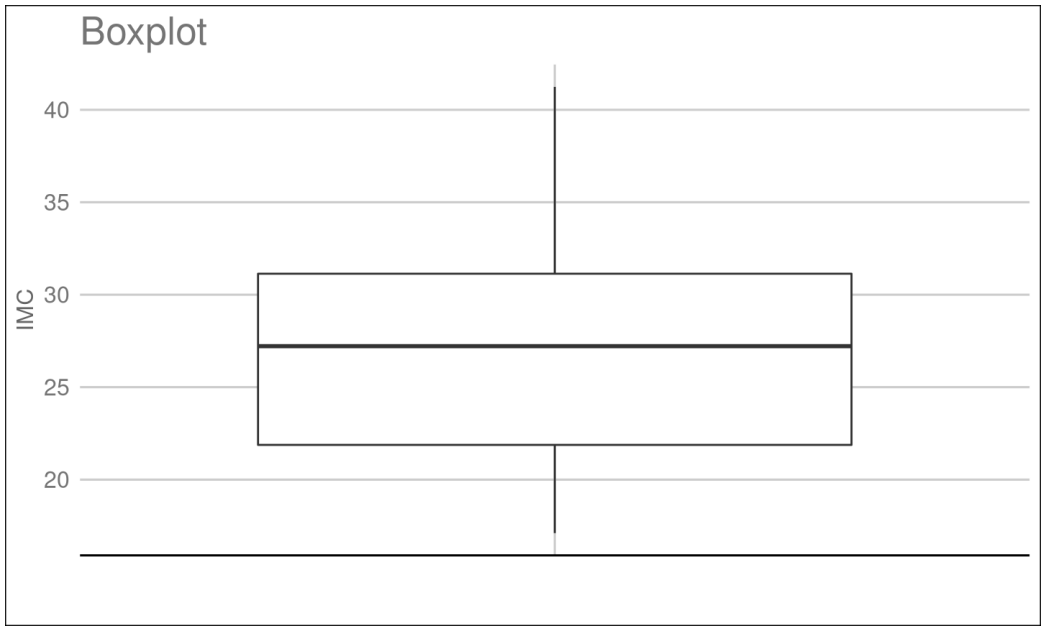


Diagrama de caixa

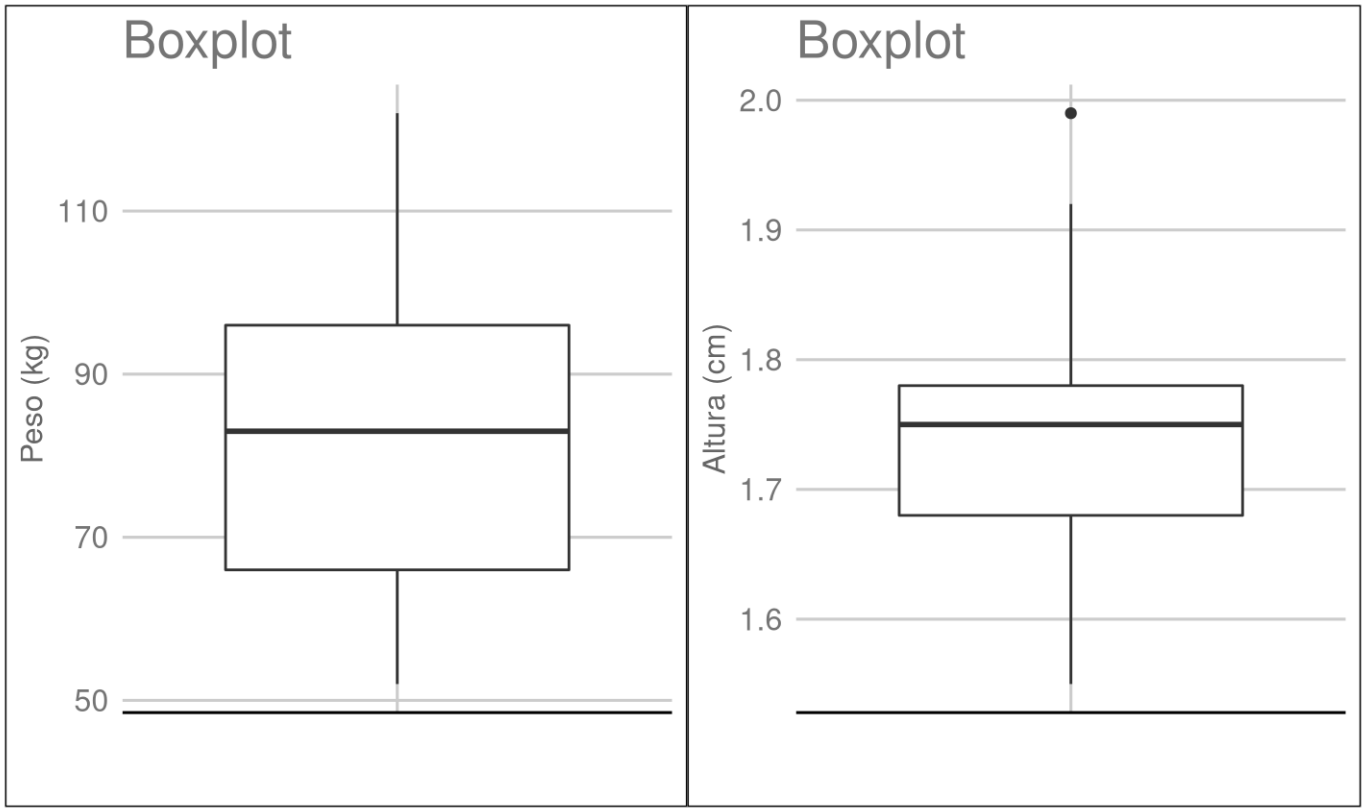
```
ggplot(df_base_processed) +
  geom_boxplot(aes(x = "", y = imc)) +
  labs(x = "", y = "IMC", title = "Boxplot") +
  theme_gdocs()
```



Gráficos lado a lado com `patchwork`

- `patchwork` permite colocar gráficos lado a lado com os operadores binários `+` (ao lado) e `\` (embaixo)
- Mais detalhes em documentação `patchwork`

```
g1 <- ggplot(df_base_processed) +
  geom_boxplot(aes(x = "", y = pes)) +
  labs(x = "", y = "Peso (kg)", title = "Boxplot") +
  theme_gdocs()
g2 <- ggplot(df_base_processed) +
  geom_boxplot(aes(x = "", y = alt)) +
  labs(x = "", y = "Altura (cm)", title = "Boxplot") +
  theme_gdocs()
g1 + g2
```


Boxplot
Peso (kg)
110
90
70
50
Boxplot
Altura (cm)
2.0
1.9
1.8
1.7
1.6

Gráficos

Duas variáveis

Gráfico de dispersão

Ideia: estudar a associação entre duas variáveis quantitativas

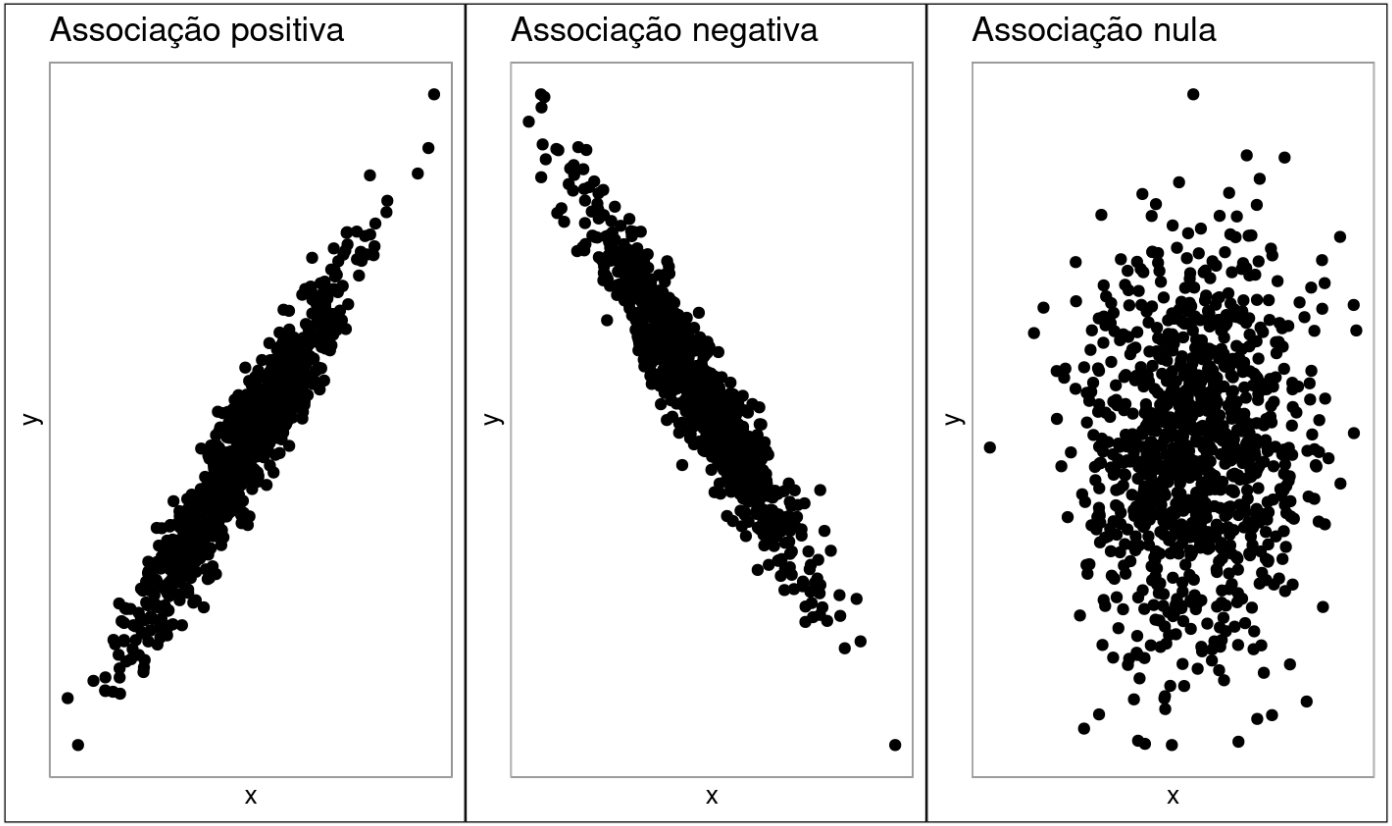


Gráfico de dispersão

```
ggplot(df_base_processed) +
  geom_point(aes(pes, imc)) +
  labs(
    x = "Peso (kg)",
    y = "IMC",
    title = "Gráfico de dispersão"
  ) +
  theme_gdocs()
```

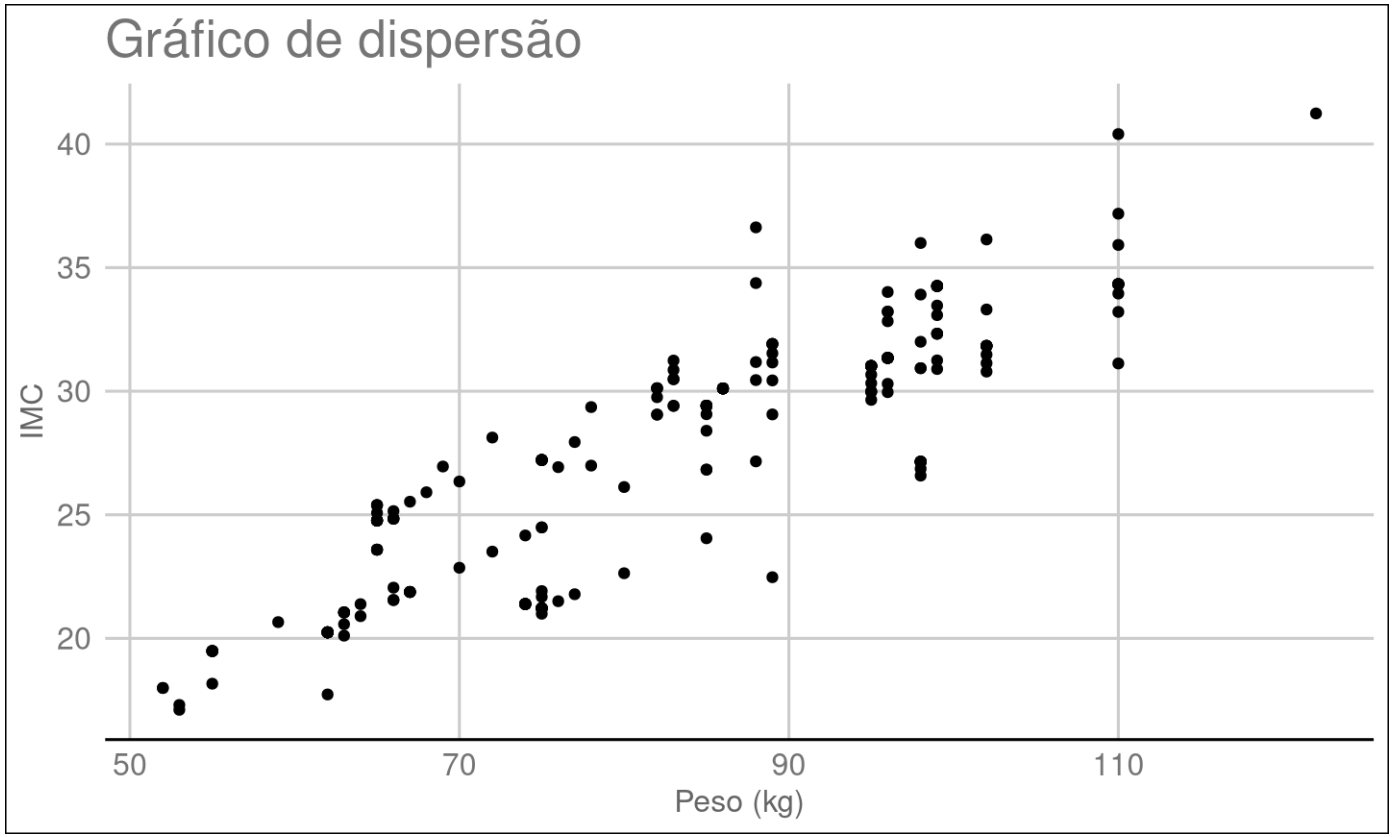


Gráfico de dispersão
IMC
40
35
30
25
20
50
70
90
110
Peso (kg)

Gráfico de barras

Ideia

Sejam X e Y duas variáveis qualitativas com seguintes valores possíveis

- X: $A_1, \ldots, A_r$
- Y. $B_1, \ldots, B_s$

Desejamos estudar a associação entre X e Y.

Associação entre X e Y

Suponha que A_i tenha porcentagem $f_i \cdot 100\%$. Então, X e Y são:

- **não associados** se ao conhecermos o valor de Y para um elemento da população, *continuamos* com a porcentagem $100 \cdot f_i\%$ deste elemento ter valor de X igual a A_i
- **associados** se ao conhecermos o valor de Y para um elemento da população, *alteramos* a porcentagem $100 \cdot f_i\%$ deste elemento ter valor de X igual a A_i

Gráfico de barras

```
df_filtrada <- df_base_processed |>
  filter(racacor != "Indígena" & !is.na(racacor) & !is.na(esc))
ggplot(df_filtrada) +
  geom_bar(aes(x = racacor, fill = esc), position = "fill") +
  labs(x = "Raça", y = "Porcentagem") +
  scale_y_continuous(labels = scales::percent)+
  scale_fill_manual(name = "Escolaridade", values = c("blue", "orange", "magenta", "brown")) +
  theme_gdocs()
```

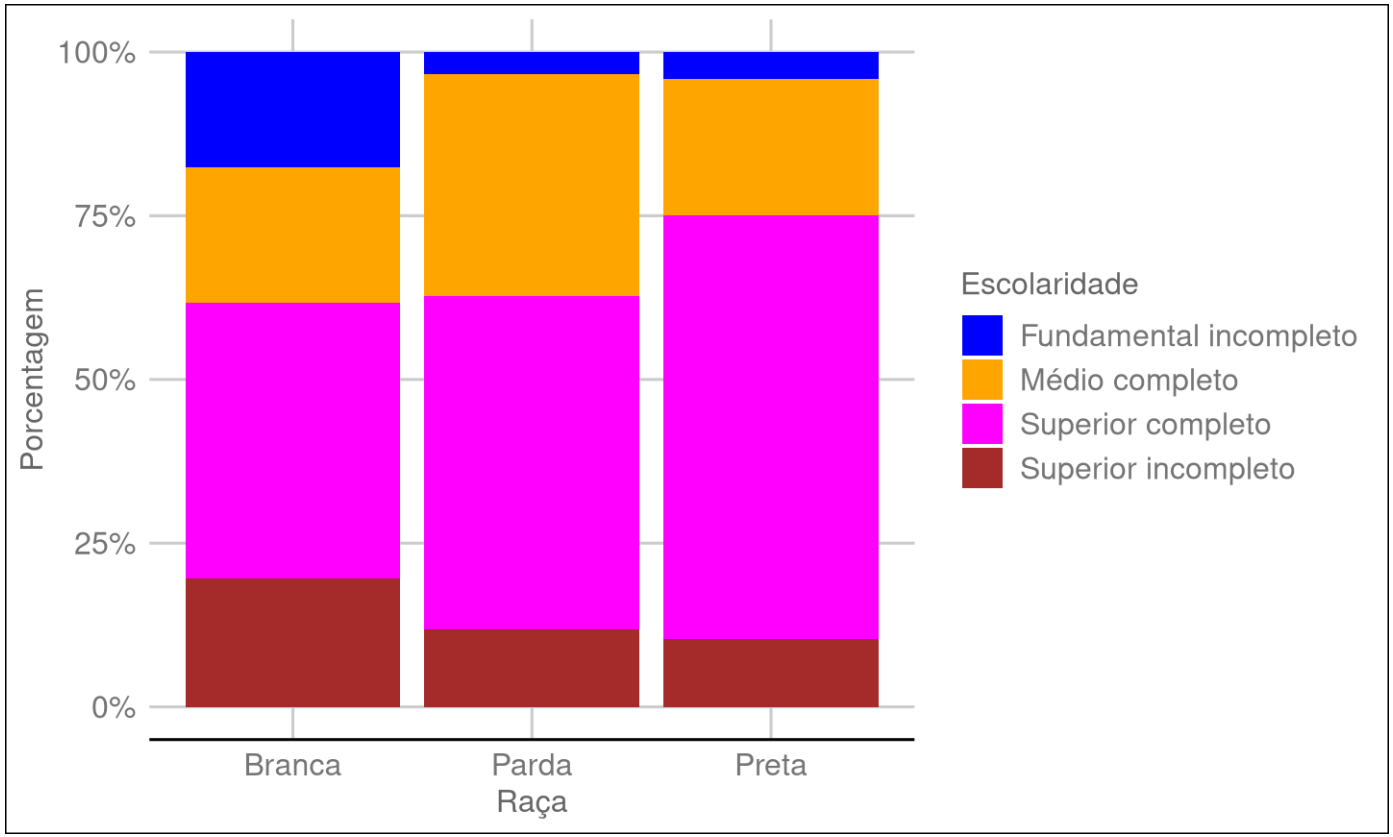

100%
75%
50%
25%
0%
Porcentagem
Branca
Parda
Preta
Raça
Escolaridade
Fundamental incompleto
Médio completo
Superior completo
Superior incompleto

Gráfico de barras

Podemos agrupar as barras por grupos para analisar a associação entre duas variáveis qualitativas.

```
df_filtrada <- df_base_processed |>
  filter(racacor != "Indígena" & !is.na(racacor) & !is.na(esc))
ggplot(df_filtrada) +
  geom_bar(aes(x = racacor, fill = esc), position = "dodge") +
  labs(x = "Raça", y = "Porcentagem") +
  scale_y_continuous(labels = scales::percent)+
  scale_fill_manual(name = "Escolaridade", values = c("blue", "orange", "magenta", "brown")) +
  theme_gdocs()
```

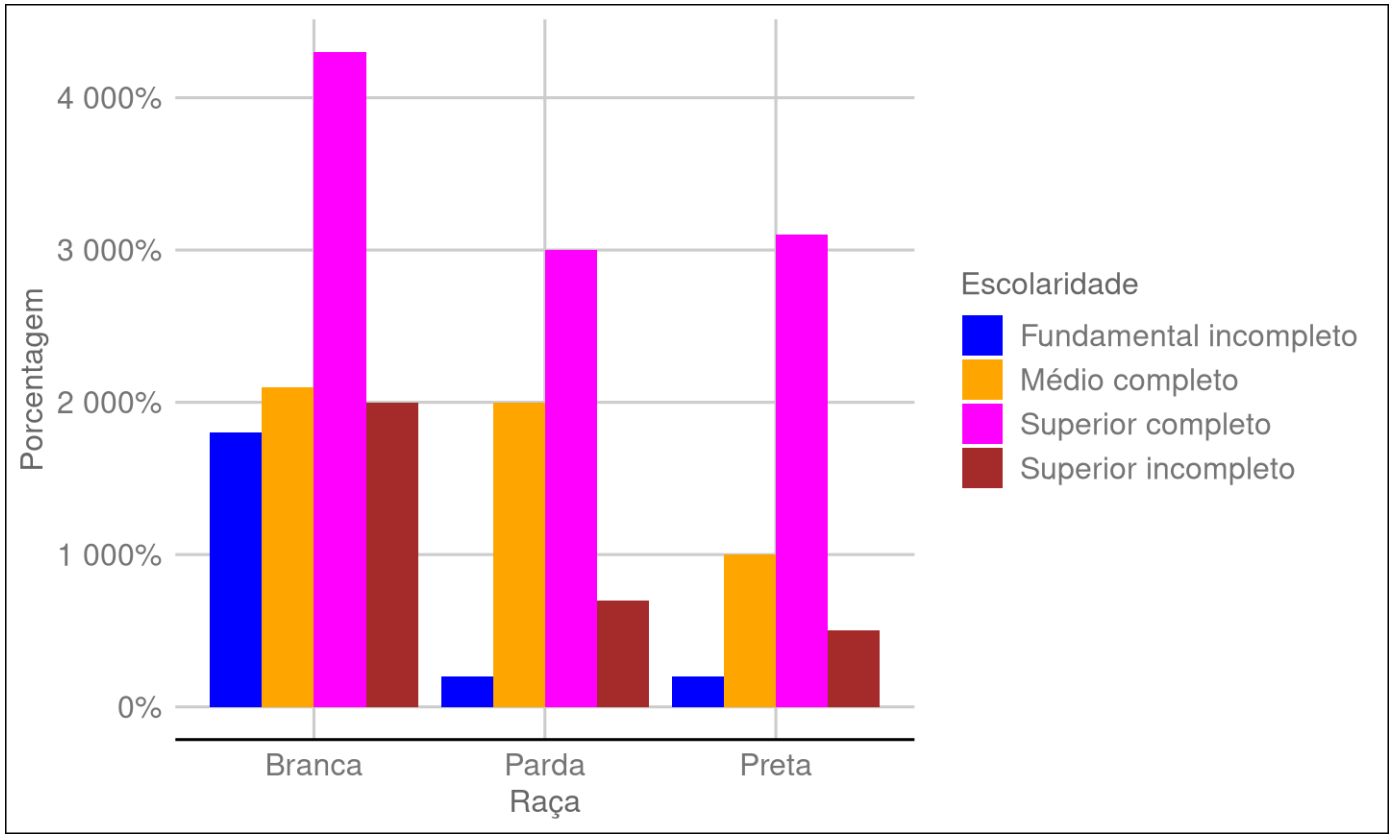
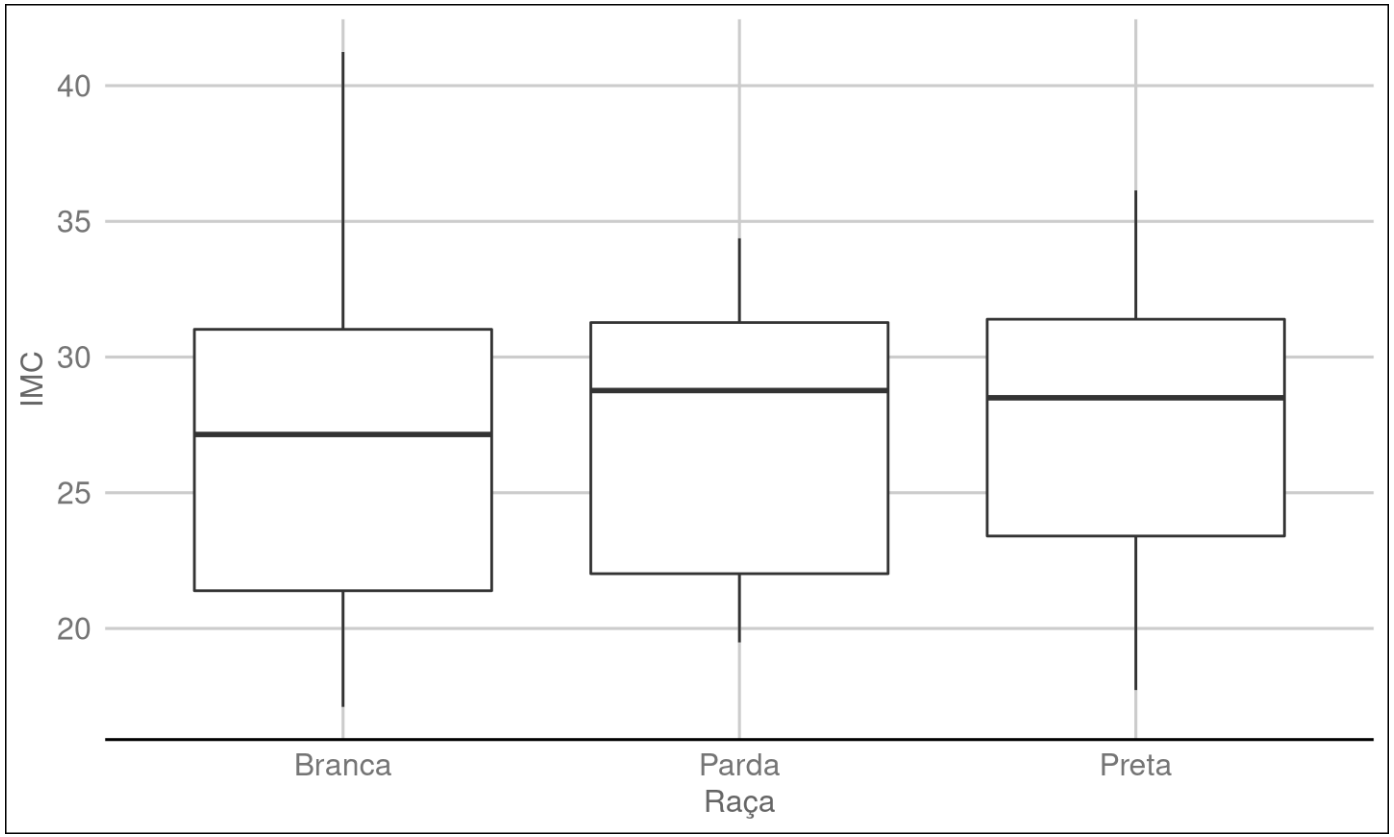

4 000%
3 000%
2 000%
1 000%
0%
Porcentagem
Branca
Parda
Preta
Raça
Escolaridade
Fundamental incompleto
Médio completo
Superior completo
Superior incompleto

Diagrama de caixa

Podemos comparar medianas de diferentes grupos usando o diagrama de caixa.

```
df_filtrada <- df_base_processed |>
  filter(racacor != "Indígena" & !is.na(racacor) & !is.na(imc))
ggplot(df_filtrada) +
  geom_boxplot(aes(x = racacor, y = imc)) +
  labs(x = "Raça", y = "IMC") +
  theme_gdocs()
```


40
35
30
25
20
IMC
Branca
Parda
Preta
Raça