

# Análise descritiva dos dados: funções, tabelas e gráficos

Curso de Bioestatística com R

---

Ornella Scardua Ferreira

✉ [ornscar@gmail.com](mailto:ornscar@gmail.com)  [@ornscar](#)  [@ornscar](#)

## Sobre mim



*Amo gráficos como amo cavalos. Gosto de música ruim e de cinema (bom). Sou apaixonada pelo Botafogo e pelo Bayern de Munique. Prefiro Vila Velha a qualquer lugar no mundo. Não tenho sonhos, mas um dia espero ver a Palestina livre.*

# Cronograma

## 1. Dados

- Base 1: dados sobre gestantes diagnosticadas com diabetes gestacional.
- Base 2: dados do Programa das Nações Unidas para o Desenvolvimento (PNUD).

## 2. Funções bases do

- Frequência relativa e absoluta.
- Medidas-resumo: medidas de posição e de dispersão.

## 3. Tabelas descritivas

- O pacote `{gtsummary}`.
- Tabelas com frequências relativas e absolutas e medidas-resumos.

## 4. Gráficos

- O pacote `{ggplot2}`.
- Gráficos univariados: barras e histogramas.
- Gráficos bivariados:
  - Variáveis qualitativas x qualitativas: barras agrupadas e empilhadas.
  - Variáveis quantitativas x quantitativas: dispersão.
  - Variáveis qualitativas x quantitativas: linhas e *boxplot*.

# Os dados

# Base 1

- A base de dados é sobre **gestantes diagnosticadas com diabetes gestacional** que realizaram o pré-natal entre os anos de 2012 a 2015 no Hospital das Clínicas da Universidade de São Paulo.
- Contém **408 observações** e **8 variáveis**, a saber:

ano: ano em que a gestante realizou o pré-natal;

idade: idade da gestante;

cor: Tabagista da gestante;

n\_gestacoes: número de gestações anteriores;

imc\_classe: IMC categórico;

hb\_glicada: exame de hemoglobina glicada categórico;

glicemia\_jejum: valor do exame de glicemia de jejum (em mg/dL);

insulina: se a gestante precisou usar insulina antes do parto.

- No R:


```
# carregando os dados
dados1 <- readxl::read_xlsx("dados/diabetes.xlsx")
```

```
# panorama da base de dados
dplyr::glimpse(dados1)
```

```
## Rows: 408
## Columns: 10
## $ idade          <dbl> 24, 42, 35, 34, 42, 41, 41, 37, 31, 39, 28, 37, 42, ...
## $ imc_classe     <chr> "Até normal", "Obeso", "Sobrepeso", "Obeso", "Obeso"...
## $ n_gestacoes   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ hist_diab_fam  <chr> "Sim", "Não", "Sim", "Sim", "Sim", "Não", "Sim", "Si...
## $ hist_diab_gest <chr> "Não", "Não", "Não", "Não", "Não", "Não", "Não", "Nã...
## $ macrosomia_fetal <chr> "Não", "Não", "Não", "Não", "Não", "Não", "Não", "Nã...
## $ tabagista      <chr> "Não", "Não", "Não", "Não", "Não", "Sim", "Não", "Nã...
## $ hac            <chr> "Não", "Não", "Não", "Não", "Sim", "Não", "Não", "Nã...
## $ glicemia_jejum <dbl> 95, 110, 114, 92, 114, 102, 96, 92, 92, 102, 100, 10...
## $ insulina       <chr> "Não", "Não", "Sim", "Não", "Sim", "Sim", "Não", "Nã...
```

# Base 2

- Base de dados do **PNUD** cujas informações socioeconômicas são dos anos de 2012 a 2021 a nível Brasil, Unidade Federativa (UF) e região metropolitana.
- Contém **490 observações** e **5 variáveis**, a saber:
  - ano: ano de análise dos indicadores;
  - agregacao: nível nacional, estadual e região metropolitana;
  - nome: nome da UF e região metropolitana;
  - gini: Índice de Gini;
  - espvida: expectativa de vida, em anos.

- No :

```
# carregando os dados
dados2 <- readxl::read_xlsx("dados/pnud.xlsx")

# panorama da base de dados
dplyr::glimpse(dados2)
```

```
## Rows: 490
## Columns: 5
## $ ano      <dbl> 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, ...
## $ agregacao <chr> "BRASIL", "BRASIL", "BRASIL", "BRASIL", "BRASIL", "BRASIL", ...
## $ nome     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "Rondônia", "Acre", ...
## $ gini     <dbl> 0.540, 0.532, 0.526, 0.524, 0.537, 0.539, 0.545, 0.544, 0.52...
## $ espvida  <dbl> 74.48, 74.80, 75.11, 75.40, 75.68, 75.96, 76.22, 76.47, 76.2...
```



# Funções bases do

# Frequência absoluta e relativa

## Frequência absoluta

```
# frequencia absoluta da va 'raca/cor' sem considerar na  
freq_abs <- table(dados1$cor); freq_abs
```

```
## Warning: Unknown or uninitialised column: `cor`.
```

```
## < table of extent 0 >
```

```
# frequencia absoluta da va 'raca/cor' considerando na  
freq_abs_na <- table(dados1$cor, useNA = "always"); freq_abs_na
```

```
## Warning: Unknown or uninitialised column: `cor`.
```

```
##
```

```
## <NA>
```

```
## 0
```

## Frequência relativa

```
# frequencia relativa da va 'raca/cor' sem considerar na  
prop.table(freq_abs)
```

```
## numeric(0)
```

```
# frequencia relativa da va 'raca/cor' considerando na  
prop.table(freq_abs_na)
```

```
##  
## <NA>  
##
```

### Dica!

Use a função `round()` para arredondar os valores. Por exemplo, `round(prop.table(freq_abs), 2)`.

# Medidas de posição

## Valores mínimo e máximo

```
# valor minimo das vas 'idade' e 'expectativa de vida'  
min(dados1$idade); min(dados2$espvida)
```

```
## [1] 16
```

```
## [1] 66.34
```

```
# valor maximo da va 'expectativa de vida'  
max(dados1$idade); max(dados2$espvida)
```

```
## [1] 47
```

```
## [1] 81.22
```

## Observação!

Se a variável tem NA, é necessário incluir o argumento `na.rm = TRUE`. Por exemplo, `min(dados1$idade, na.rm = TRUE)`.

## Moda

```
# funcao para calcular moda
calc_moda <- function(x, na.rm = TRUE) {
  if (na.rm) {                                # se tiver na,
    x <- x[!is.na(x)]                        # filtra valores diferentes de na
  }
  freq <- table(x)                            # calcula as frequencias
  moda <- names(freq)[freq == max(freq)]      # encontrao valor mais frequente
  return(type.convert(modas, as.is = TRUE))  # mantem o tipo da variavel
}

# moda da va 'expectativa de vida'
calc_moda(dados2$espvida)
```

```
## [1] 72.03 74.16 74.20
```

### Observação!

Nesse caso, não é necessário usar o argumento `na.rm = TRUE` porque os NAs da variável já são desconsiderados.

## Mediana

```
# mediana das vas 'valor do exame de glicemia de jejum' e 'indice de gini'  
median(dados1$glicemia_jejum); median(dados2$gini)
```

```
## [1] 96
```

```
## [1] 0.525
```

## Média

```
# media das vas 'idade' e 'indice de gini'  
mean(dados1$idade); mean(dados2$gini)
```

```
## [1] 32.71814
```

```
## [1] 0.5204918
```

## Observação!

Se necessário, utilize o argumento `na.rm = TRUE`.

## Quartis

```
# quartis padrao da va 'valor do exame de glicemia de jejum'  
quantile(dados1$glicemia_jejum)
```

```
##    0%   25%   50%   75%  100%  
##    92    94    96   101   124
```

## Percentis

```
# percentis 10, 20 e 90 da va 'valor do exame de glicemia de jejum'  
quantile(dados1$glicemia_jejum, probs = c(0.1, 0.2, 0.9))
```

```
## 10% 20% 90%  
##  92  93 107
```

## Observação!

Se necessário, utilize o argumento `na.rm = TRUE`.

# Medidas de dispersão

## Amplitude

```
# amplitude da va 'idade'  
max(dados1$idade) - min(dados1$idade)
```

```
## [1] 31
```

## Intervalo interquartil

```
# intervalo interquartil da va 'idade'  
IQR(dados1$idade)
```

```
## [1] 8.25
```

## Observação!

Se necessário, utilize o argumento `na.rm = TRUE` na função `IQR()`.



## Variância

```
# variância da va 'indice de gini'  
var(dados2$gini)
```

```
## [1] 0.001634909
```

## Desvio-padrão

```
# desvio-padrao da va 'indice de gini'  
sd(dados2$gini)
```

```
## [1] 0.04043401
```

## Coeficiente de variação (CV)

```
(sd(dados2$gini) / mean(dados2$gini)) * 100
```

```
## [1] 7.768423
```

## Observação!

Nas funções de variância e desvio-padrão, pode-se utilizar o argumento `na.rm = TRUE`.

# Tabelas descritivas

# O pacote `{gtsummary}`

- Criado pelo [Daniel D. Sjoberg](#), bioestatístico do Memorial Sloan Kettering Cancer Center (EUA).
- Gera automaticamente tabelas com formatação elegante e profissional, prontas para relatórios, artigos científicos e apresentações.
- Repositório oficial: <https://github.com/ddsjoberg/gtsummary>.
- Infinitas possibilidades de customização!

## Outros pacotes

- `{modelsummary}`: <https://modelsummary.com>.
- `{summarytools}`: <https://cran.r-project.org/web/packages/summarytools/vignettes/introduction.html>.

## Tabela descritiva da Base 1

```
library(gtsummary)

# criando tabela gtsummary
dados1 |>
  tbl_summary()
```

Characteristic	N = 408 <sup>1</sup>
idade	34 (29, 37)
imc_classe	
Até normal	87 (21%)
Obeso	194 (48%)
Sobrepeso	127 (31%)
n_gestacoes	2.00 (1.00, 4.00)
hist_diab_fam	
Não	150 (37%)
Sim	257 (63%)
Unknown	1
hist_diab_gest	
Não	359 (88%)
Sim	49 (12%)
macrossomia_fetal	
Não	373 (91%)
Sim	35 (8.6%)
tabagista	

```
# incluindo tema a tabela
theme_gtsummary_journal(journal = "lancet")

dados1 |>
  tbl_summary(
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes)
  )
```

Characteristic	N = 408 <sup>1</sup>
idade	34 (29 – 37)
imc_classe	
Até normal	87 (21%)
Obeso	194 (48%)
Sobrepeso	127 (31%)
n_gestacoes	2.00 (1.00 – 4.00)
hist_diab_fam	
Não	150 (37%)
Sim	257 (63%)
Unknown	1
glicemia_jejum	96 (94 – 101)
insulina	
Não	273 (67%)
Sim	135 (33%)
<sup>1</sup> Median (IQR); n (%)	

```
dados1 |>
  tbl_summary(
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes),
    # rotulando as variaveis
    label = list(
      idade ~ "Idade (anos)",
      imc_classe ~ "IMC",
      n_gestacoes ~ "Nº de gestações anteriores",
      hist_diab_fam ~ "Histórico de diabetes na família",
      glicemia_jejum ~ "Valor do exame de glicemia em jejum",
      insulina ~ "Usou insulina antes do parto"
    )
  )
```

Characteristic	N = 408 <sup>1</sup>
Idade (anos)	34 (29 – 37)
IMC	
Até normal	87 (21%)
Obeso	194 (48%)
Sobrepeso	127 (31%)
Nº de gestações anteriores	2.00 (1.00 – 4.00)
Histórico de diabetes na família	
Não	150 (37%)
Sim	257 (63%)
Unknown	1
Valor do exame de glicemia (mg/dL)	96 (94 – 101)
Usou insulina antes do parto	
Não	273 (67%)
Sim	135 (33%)
<sup>1</sup> Median (IQR); n (%)	

```

dados1 |>
  tbl_summary(
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes),
    # rotulando as variaveis
    label = list(
      idade ~ "Idade (anos)",
      imc_classe ~ "IMC",
      n_gestacoes ~ "Nº de gestações anteriores",
      hist_diab_fam ~ "Histórico de diabetes na família",
      glicemia_jejum ~ "Valor do exame de glicemia em jejum (mg/dL)",
      insulina ~ "Usou insulina antes do parto"
    ),
    # calculando as medidas de interesse
    statistic = list(
      all_continuous() ~ "{mean} ± {sd}",
      all_categorical() ~ "{n} ({p}%)"
    )
  )

```

Characteristic	N = 408 <sup>1</sup>
Idade (anos)	33 ± 6
IMC	
Até normal	87 (21%)
Obeso	194 (48%)
Sobrepeso	127 (31%)
Nº de gestações anteriores	2.80 ± 1.69
Histórico de diabetes na família	
Não	150 (37%)
Sim	257 (63%)
Unknown	1
Valor do exame de glicemia (mg/dL)	98 ± 7
Usou insulina antes do parto	
Não	273 (67%)
Sim	135 (33%)
<sup>1</sup> Mean ± SD; n (%)	

```

dados1 |>
tbl_summary(
  # incluindo apenas as variaveis de interesse
  include = c(idade, imc_classe, n_gestacoes, hist_diab_fam, glicemia_jejum,
  # rotulando as variaveis
  label = list(
    idade ~ "Idade (anos)",
    imc_classe ~ "IMC",
    n_gestacoes ~ "Nº de gestações anteriores",
    hist_diab_fam ~ "Histórico de diabetes na família",
    glicemia_jejum ~ "Valor do exame de glicemia em jejum (mg/dL)",
    insulina ~ "Usou insulina antes do parto"
  ),
  # calculando as medidas de interesse
  statistic = list(
    all_continuous() ~ "{mean} ± {sd}",
    all_categorical() ~ "{n} ({p}%)"
  ),
  # formatando numero de digitos
  digits = list(
    all_continuous() ~ 2, # 2 digitos
    all_categorical() ~ c(0, 1) # 0 digitos
  )
)

```

Characteristic	N = 408 <sup>1</sup>
Idade (anos)	32.72 ± 6.06
IMC	
Até normal	87 (21.3%)
Obeso	194 (47.5%)
Sobrepeso	127 (31.1%)
Nº de gestações anteriores	2.80 ± 1.69
Histórico de diabetes na família	
Não	150 (36.9%)
Sim	257 (63.1%)
Unknown	1
Valor do exame de glicemia (mg/dL)	98.25 ± 6.53
Usou insulina antes do parto	
Não	273 (66.9%)
Sim	135 (33.1%)

<sup>1</sup> Mean ± SD; n (%)



```

dados1 |>
tbl_summary(
  # incluindo apenas as variaveis de interesse
  include = c(idade, imc_classe, n_gestacoes, hist_diab_fam, glicemia_jejum, usou_insulina),
  # rotulando as variaveis
  label = list(
    idade ~ "Idade (anos)",
    imc_classe ~ "IMC",
    n_gestacoes ~ "Nº de gestações anteriores",
    hist_diab_fam ~ "Histórico de diabetes na família",
    glicemia_jejum ~ "Valor do exame de glicemia em jejum (mg/dL)",
    usou_insulina ~ "Usou insulina antes do parto"
  ),
  # calculando as medidas de interesse
  statistic = list(
    all_continuous() ~ "{mean} ± {sd}",
    all_categorical() ~ "{n} ({p}%)"
  ),
  # formatando numero de digitos
  digits = list(
    all_continuous() ~ 2, # 2 digitos
    all_categorical() ~ c(0, 1) # 0 digitos
  ),
  # alterando rotulo dos valores faltantes
  missing_text = "NA"
)

```

Characteristic	N = 408 <sup>1</sup>
Idade (anos)	32.72 ± 6.06
IMC	
Até normal	87 (21.3%)
Obeso	194 (47.5%)
Sobrepeso	127 (31.1%)
Nº de gestações anteriores	2.80 ± 1.69
Histórico de diabetes na família	
Não	150 (36.9%)
Sim	257 (63.1%)
NA	1
Valor do exame de glicemia (mg/dL)	98.25 ± 6.53
Usou insulina antes do parto	
Não	273 (66.9%)
Sim	135 (33.1%)
<sup>1</sup> Mean ± SD; n (%)	

```

dados1 |>
  tbl_summary(
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes, hist_diab_fam, glicemia_jejum, usou_insulina),
    # rotulando as variaveis
    label = list(
      idade ~ "Idade (anos)",
      imc_classe ~ "IMC",
      n_gestacoes ~ "Nº de gestações anteriores",
      hist_diab_fam ~ "Histórico de diabetes na família",
      glicemia_jejum ~ "Valor do exame de glicemia (mg/dL)",
      usou_insulina ~ "Usou insulina antes do parto"
    ),
    # calculando as medidas de interesse
    statistic = list(
      all_continuous() ~ "{mean} ± {sd}",
      all_categorical() ~ "{n} ({p}%)"
    ),
    # formatando numero de digitos
    digits = list(
      all_continuous() ~ 2, # 2 digitos
      all_categorical() ~ c(0, 1) # 0 digitos
    ),
    # alterando rotulo dos valores faltantes
    missing_text = "NA"
  ) |>
  # modificando os nomes do cabecalho

```

Variável	N = 408 <sup>1</sup>
Idade (anos)	32.72 ± 6.06
IMC	
Até normal	87 (21.3%)
Obeso	194 (47.5%)
Sobrepeso	127 (31.1%)
Nº de gestações anteriores	2.80 ± 1.69
Histórico de diabetes na família	
Não	150 (36.9%)
Sim	257 (63.1%)
NA	1
Valor do exame de glicemia (mg/dL)	98.25 ± 6.53
Usou insulina antes do parto	
Não	273 (66.9%)
Sim	135 (33.1%)
<sup>1</sup> Mean ± SD; n (%)	

```

dados1 |>
tbl_summary(
  # incluindo apenas as variaveis de interesse
  include = c(idade, imc_classe, n_gestacoes, hist_diab_fam, glicemia_jejum, insulina),
  # rotulando as variaveis
  label = list(
    idade ~ "Idade (anos)",
    imc_classe ~ "IMC",
    n_gestacoes ~ "Nº de gestações anteriores",
    hist_diab_fam ~ "Histórico de diabetes na família",
    glicemia_jejum ~ "Valor do exame de glicemia em jejum (mg/dL)",
    insulina ~ "Usou insulina antes do parto"
  ),
  # calculando as medidas de interesse
  statistic = list(
    all_continuous() ~ "{mean} ± {sd}",
    all_categorical() ~ "{n} ({p}%)",
  ),
  # formatando numero de digitos
  digits = list(
    all_continuous() ~ 2, # 2 digitos
    all_categorical() ~ c(0, 1) # 0 digitos
  ),
  # alterando rotulo dos valores faltantes
  missing_text = "NA"
) |>
# modificando os nomes do cabecalho

```

Variável	N = 408 <sup>1</sup>
<b>Idade (anos)</b>	32.72 ± 6.06
<b>IMC</b>	
Até normal	87 (21.3%)
Obeso	194 (47.5%)
Sobrepeso	127 (31.1%)
<b>Nº de gestações anteriores</b>	2.80 ± 1.69
<b>Histórico de diabetes na família</b>	
Não	150 (36.9%)
Sim	257 (63.1%)
NA	1
<b>Valor do exame de glicemia (mg/dL)</b>	98.25 ± 6.53
<b>Usou insulina antes do parto</b>	
Não	273 (66.9%)
Sim	135 (33.1%)
<sup>1</sup> Mean ± SD; n (%)	

```

dados1 |>
  tbl_summary(
    # analisando por grupo de desfecho
    by = insulina,
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes),
    # rotulando as variaveis
    label = list(
      idade ~ "Idade (anos)",
      imc_classe ~ "IMC",
      n_gestacoes ~ "Nº de gestações anteriores",
      hist_diab_fam ~ "Histórico de diabetes na família",
      glicemia_jejum ~ "Valor do exame de glicemia (mg/dL)"
    ),
    # calculando as medidas de interesse
    statistic = list(
      all_continuous() ~ "{mean} ± {sd}",
      all_categorical() ~ "{n} ({p}%)"
    ),
    # formatando numero de digitos
    digits = list(
      all_continuous() ~ 2, # 2 digitos
      all_categorical() ~ c(0, 1) # 0 digitos
    ),
    # alterando rotulo dos valores faltantes
    missing_text = "NA"
  ) |>

```

Variável	Não N = 273 <sup>1</sup>	Sim N = 135 <sup>1</sup>
<b>Idade (anos)</b>	32.03 ± 6.12	34.12 ± 5.72
<b>IMC</b>		
Até normal	71 (26.0%)	16 (11.9%)
Obeso	114 (41.8%)	80 (59.3%)
Sobrepeso	88 (32.2%)	39 (28.9%)
<b>Nº de gestações anteriores</b>	2.61 ± 1.56	3.19 ± 1.87
<b>Histórico de diabetes na família</b>		
Não	113 (41.5%)	37 (27.4%)
Sim	159 (58.5%)	98 (72.6%)
NA	1	0
<b>Valor do exame de glicemia (mg/dL)</b>	96.81 ± 5.24	101.16 ± 7.81
<sup>1</sup> Mean ± SD; n (%)		

```

dados1 |>
tbl_summary(
  # analisando por grupo de desfecho
  by = insulina,
  # incluindo apenas as variaveis de interesse
  include = c(idade, imc_classe, n_gestacoes),
  # rotulando as variaveis
  label = list(
    idade ~ "Idade (anos)",
    imc_classe ~ "IMC",
    n_gestacoes ~ "Nº de gestações anteriores",
    hist_diab_fam ~ "Histórico de diabetes na família",
    glicemia_jejum ~ "Valor do exame de glicemia (mg/dL)"
  ),
  # calculando as medidas de interesse
  statistic = list(
    all_continuous() ~ "{mean} ± {sd}",
    all_categorical() ~ "{n} ({p}%"
  ),
  # formatando numero de digitos
  digits = list(
    all_continuous() ~ 2, # 2 digitos
    all_categorical() ~ c(0, 1) # 0 digitos
  ),
  # alterando rotulo dos valores faltantes
  missing_text = "NA"
) |>

```

Variável	Usou insulina antes do parto	
	Não N = 273 <sup>1</sup>	Sim N = 135 <sup>1</sup>
<b>Idade (anos)</b>	32.03 ± 6.12	34.12 ± 5.72
<b>IMC</b>		
Até normal	71 (26.0%)	16 (11.9%)
Obeso	114 (41.8%)	80 (59.3%)
Sobrepeso	88 (32.2%)	39 (28.9%)
<b>Nº de gestações anteriores</b>	2.61 ± 1.56	3.19 ± 1.87
<b>Histórico de diabetes na família</b>		
Não	113 (41.5%)	37 (27.4%)
Sim	159 (58.5%)	98 (72.6%)
NA	1	0
<b>Valor do exame de glicemia (mg/dL)</b>	96.81 ± 5.24	101.16 ± 7.81
<sup>1</sup> Mean ± SD; n (%)		

```
# traduzindo a tabela para pt-br
theme_gtsummary_language("pt", big.mark

tbl <- dados1 |>
  tbl_summary(
    # analisando por grupo de desfecho
    by = insulina,
    # incluindo apenas as variaveis de interesse
    include = c(idade, imc_classe, n_gestacoes),
    # rotulando as variaveis
    label = list(
      idade ~ "Idade (anos)",
      imc_classe ~ "IMC",
      n_gestacoes ~ "Nº de gestações anteriores",
      hist_diab_fam ~ "Histórico de diabetes na família",
      glicemia_jejum ~ "Valor do exame de glicemia (mg/dL)"
    ),
    # calculando as medidas de interesse
    statistic = list(
      all_continuous() ~ "{mean} ± {sd}",
      all_categorical() ~ "{n} ({p}%"
    ),
    # formatando numero de digitos
    digits = list(
      all_continuous() ~ 2, # 2 digitos
      all_categorical() ~ c(0, 1) # 0 digitos
    ),
  )
```

Variável	Usou insulina antes do parto	
	Não N = 273 <sup>1</sup>	Sim N = 135 <sup>1</sup>
<b>Idade (anos)</b>	32,03 ± 6,12	34,12 ± 5,72
<b>IMC</b>		
Até normal	71 (26,0%)	16 (11,9%)
Obeso	114 (41,8%)	80 (59,3%)
Sobrepeso	88 (32,2%)	39 (28,9%)
<b>Nº de gestações anteriores</b>	2,61 ± 1,56	3,19 ± 1,87
<b>Histórico de diabetes na família</b>		
Não	113 (41,5%)	37 (27,4%)
Sim	159 (58,5%)	98 (72,6%)
NA	1	0
<b>Valor do exame de glicemia (mg/dL)</b>	96,81 ± 5,24	101,16 ± 7,81

<sup>1</sup> Média ± Desvio Padrão; n (%)

# Salvando uma tabela gtsummary no

- A tabela precisa ser do tipo gt. Para transformá-la nesse formato, use a função `as_gt()`, do {gtsummary}.

```
# transformando a tabela de gtsummary para gt  
tbl <- tbl |> as_gt()
```

- Para salvar a tabela, execute a função `gtsave()`, do pacote {gt}.

```
# carregando o pacote {gt}  
library(gt)  
  
# salvando imagem no formato png  
gtsave(tbl, "tabelas/tbl_desc.png")  
  
# salvando imagem no formato docx  
gtsave(tbl, "tabelas/tbl_desc.docx")
```

# Gráficos



No  $\mathbb{R}$ , é possível fazer...

(outras representações gráficas)

... gráfico animado;

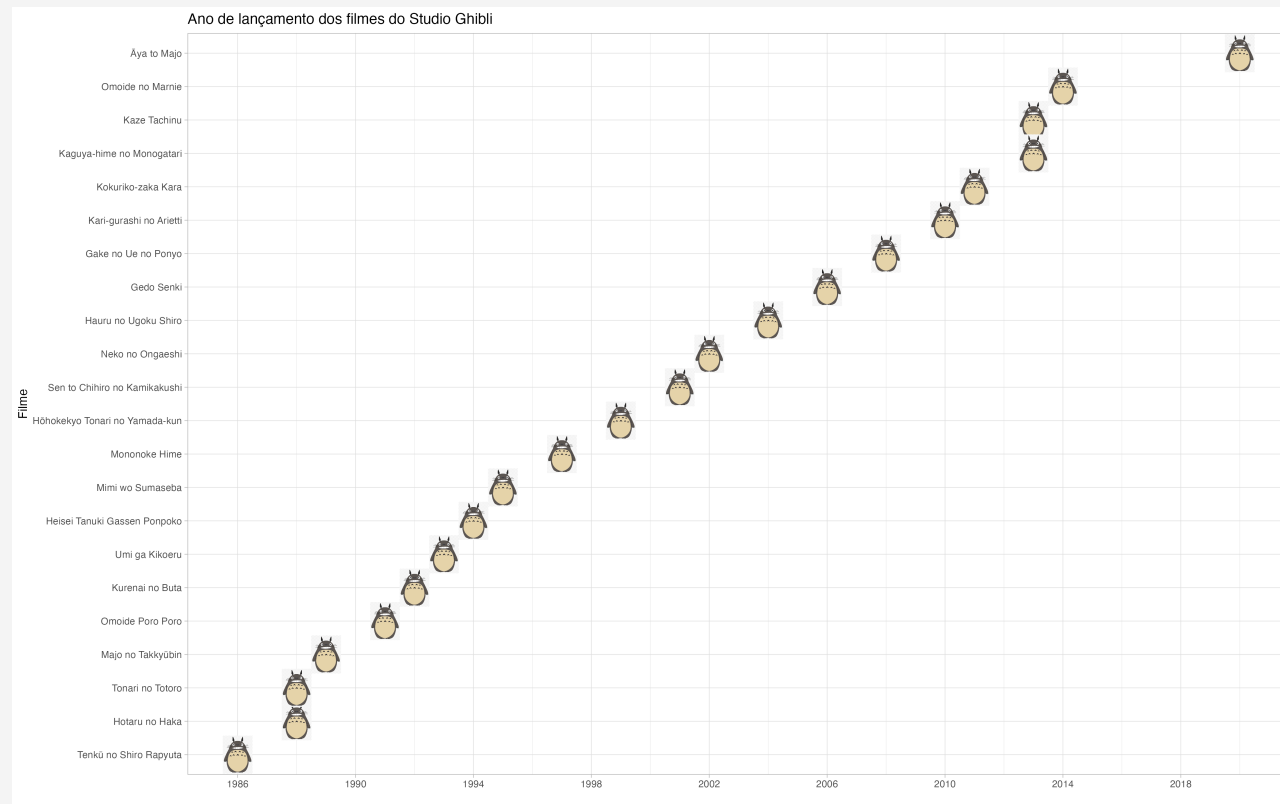


Figura 1: Gráfico animado do ano de lançamento dos filmes do Studio Ghibli.

... gráfico de densidades;

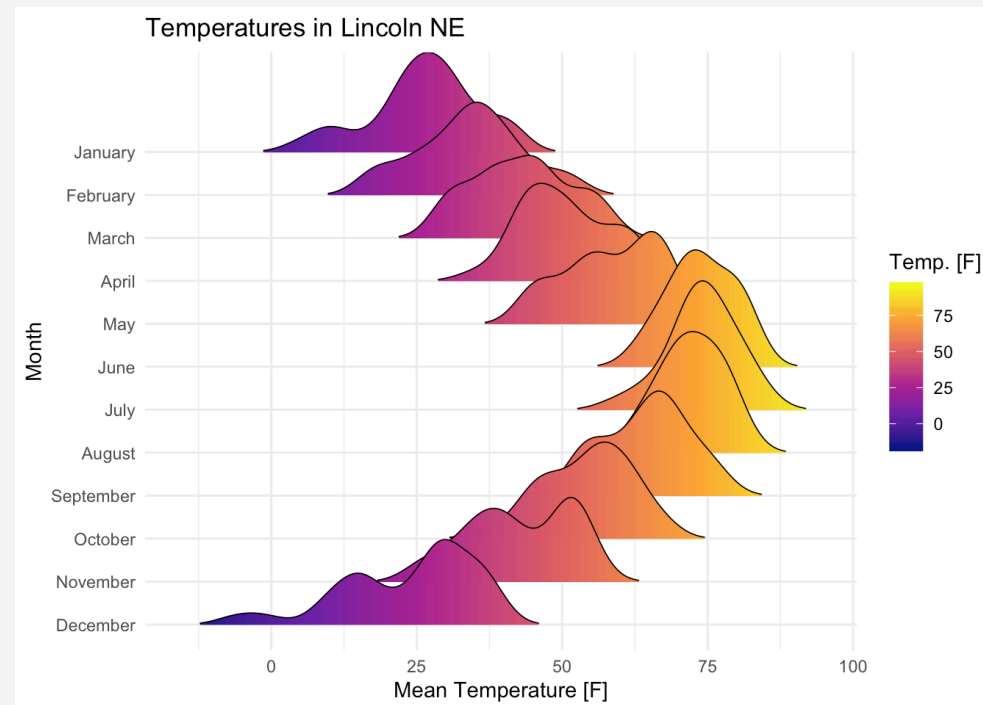


Figura 2: Gráfico das distribuições de densidade da temperatura, por mês do ano de 2016, na cidade de Lincoln, em Nebraska/EUA.

Fonte: [Datanovia](#) | [Elegant visualization of density distribution in R using ridgeline.](#)

... nuvem de palavras;

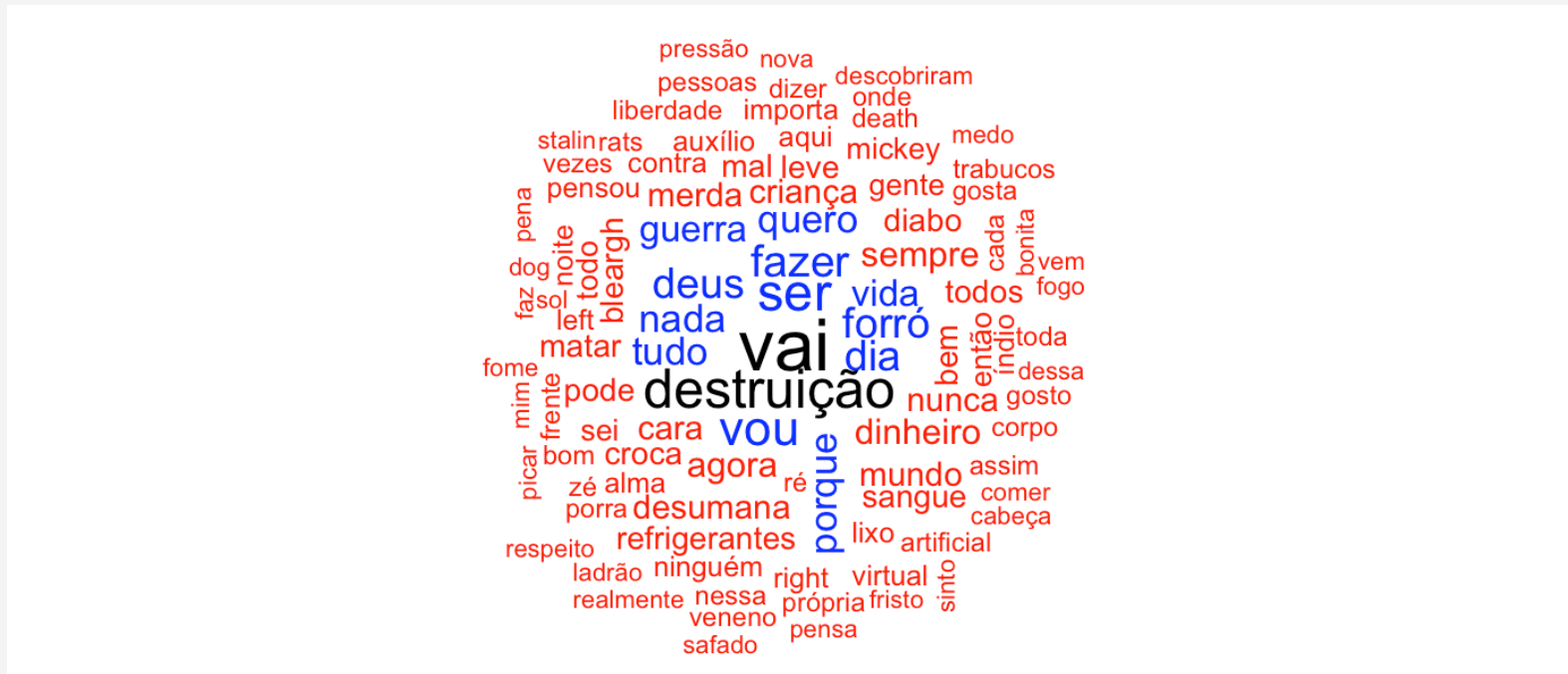


Figura 3: Nuvem de palavras das músicas da Mukeka di Rato.

... gráfico de florestas ( *forest plot*);

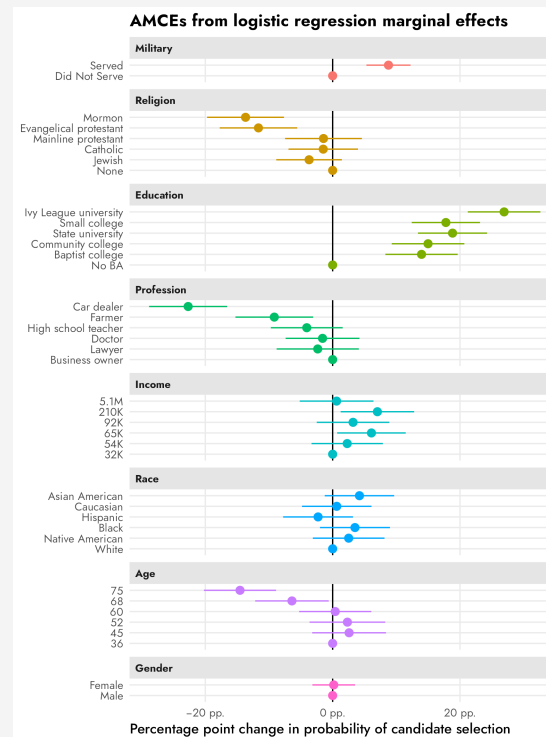


Figura 4: Gráfico de floresta dos efeitos marginais médios entre candidatos, segundo condições sociodemográficas.

Fonte: [Andrew Heiss | The ultimate practical guide to conjoint analysis with R.](#)

... gráfico de radar (ou de aranha);

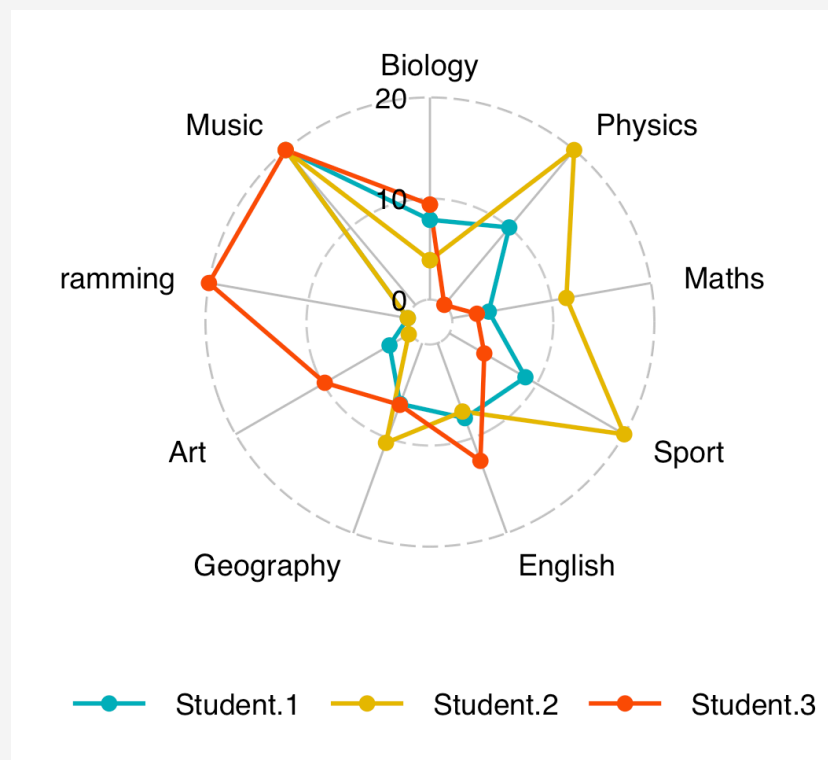


Figura 5: Gráfico de radar das notas de alunos de um determinado colégio.

Fonte: [Datanovia](#) | Beautiful radar chart in R using fmsb and ggplot packages.

... arte;

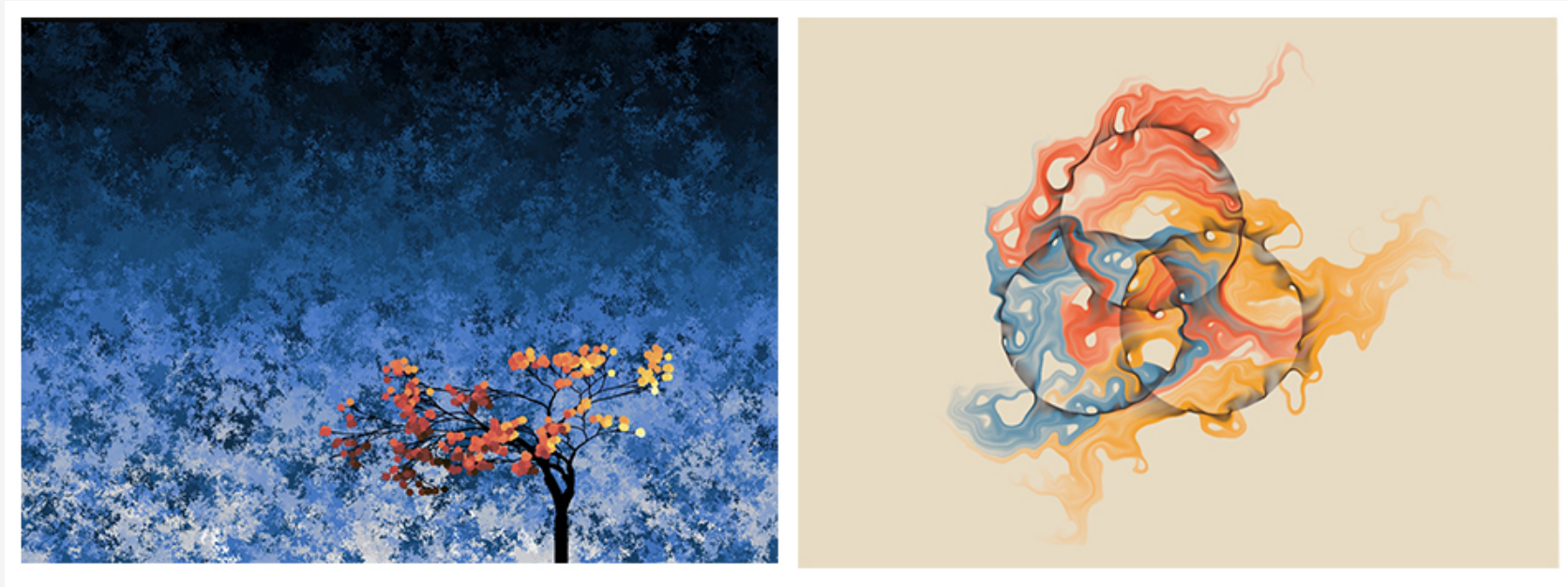


Figura 6: Arte com R por Danielle Navarro (à esquerda) e Thomas Lin Pedersen (à direita).

Fonte: [Art by Danielle Navarro](#) e [Data Imaginist - Visualization and beyond....](#)

... e muito mais!



# O pacote {ggplot2}

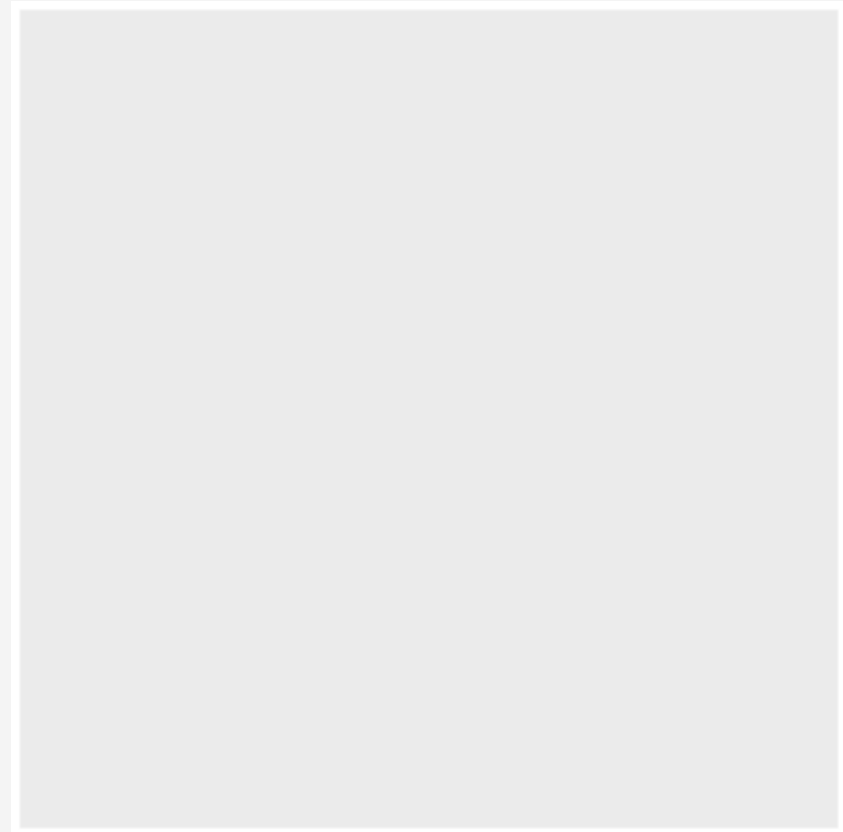
- Criado pelo estatístico neo-zelandês [Hadley Wickham](#).
- No livro *A layered grammar of graphics* (em português: "*Uma gramática em camadas dos gráficos*"), Hadley define que os elementos de um gráfico (dados, cores, formas geométricas, coordenadas, anotações etc) são camadas e que um gráfico é um conjunto de sobreposições de camadas.
- Vantagens:
  - MAIS bonitos;
  - MAIS intuitivos;
  - MAIS customizáveis;
  - sintaxe MAIS padronizada.
- Documentação: <https://ggplot2.tidyverse.org/>.
- Cheatsheet: <https://rstudio.github.io/cheatsheets/data-visualization.pdf>.

# Gráfico univariados

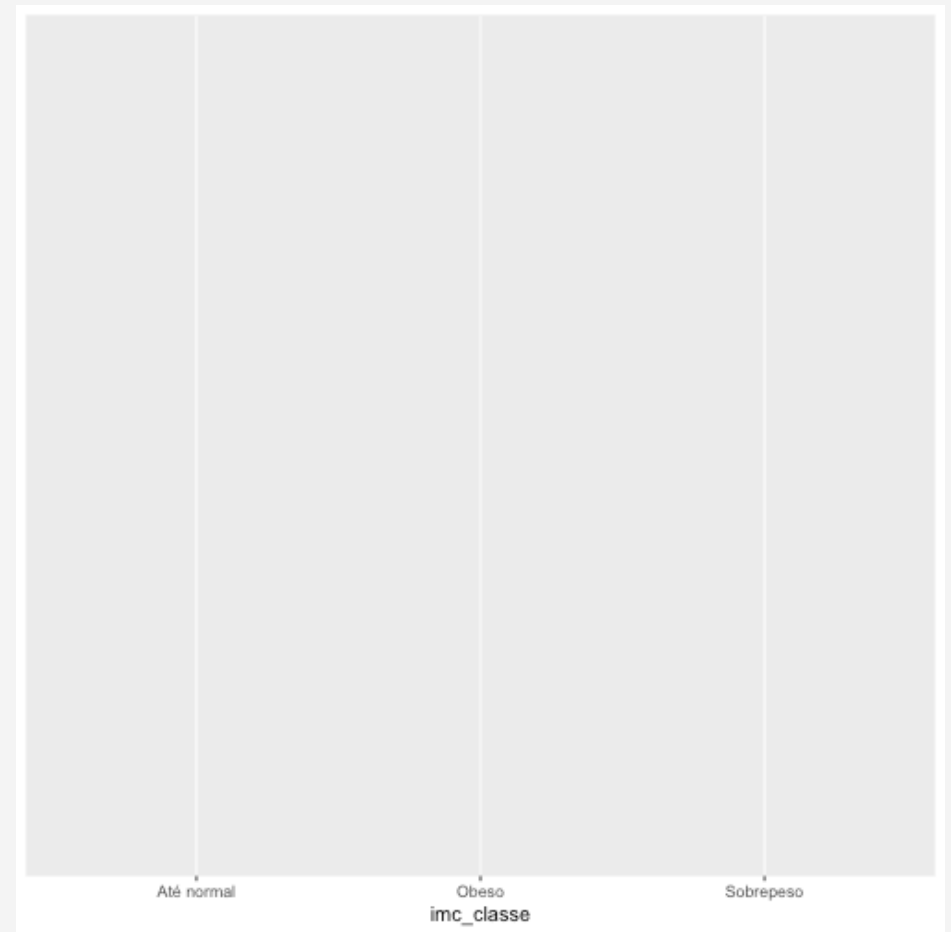
## Gráfico de barras

```
library(ggplot2)
```

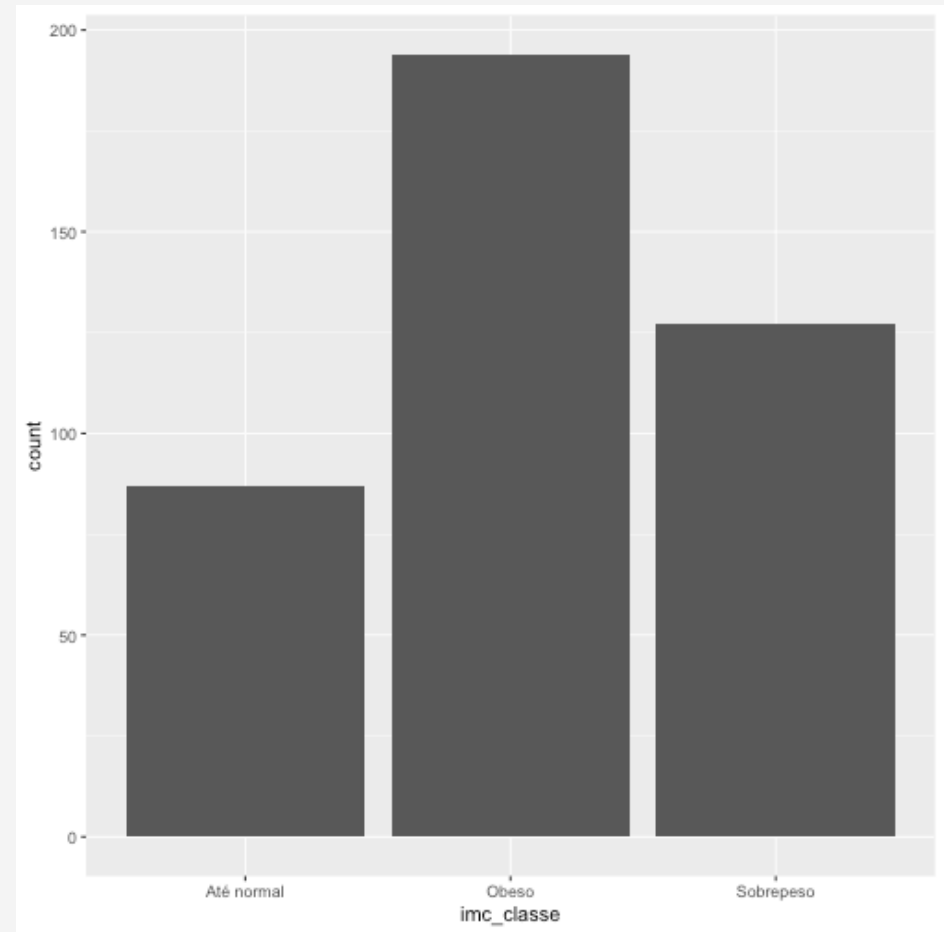
```
ggplot(dados1)
```



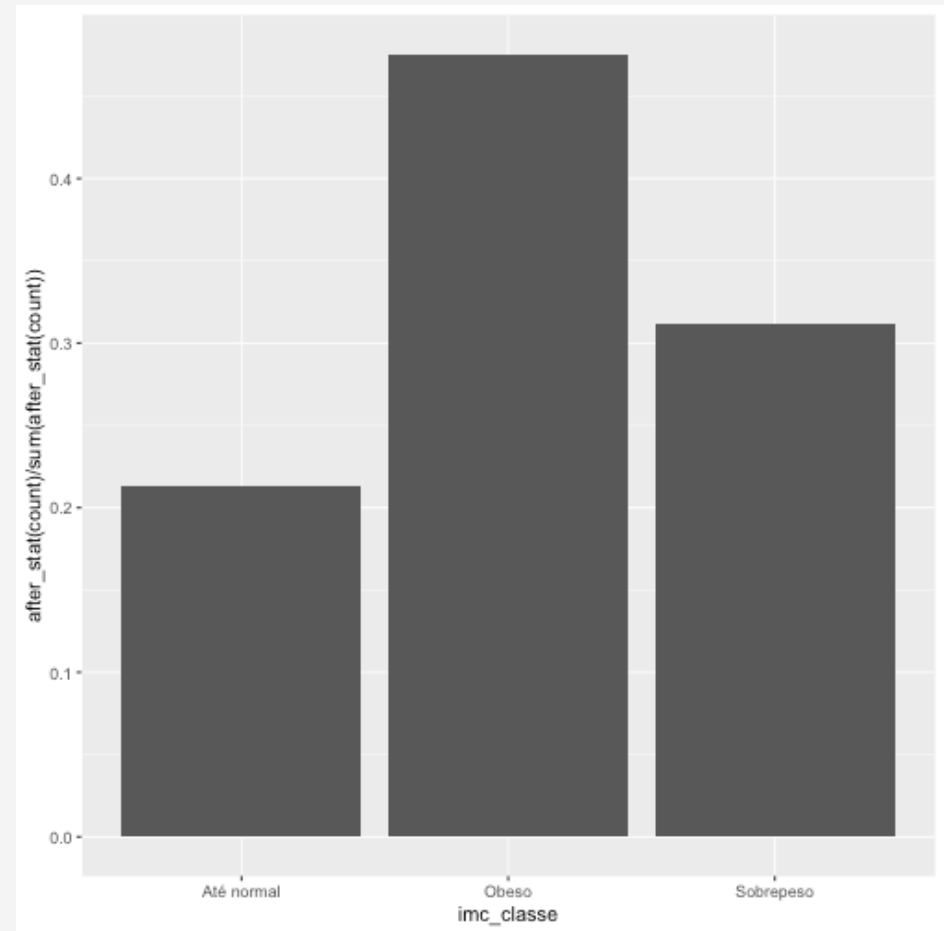
```
ggplot(dados1, aes(x = imc_classe))
```



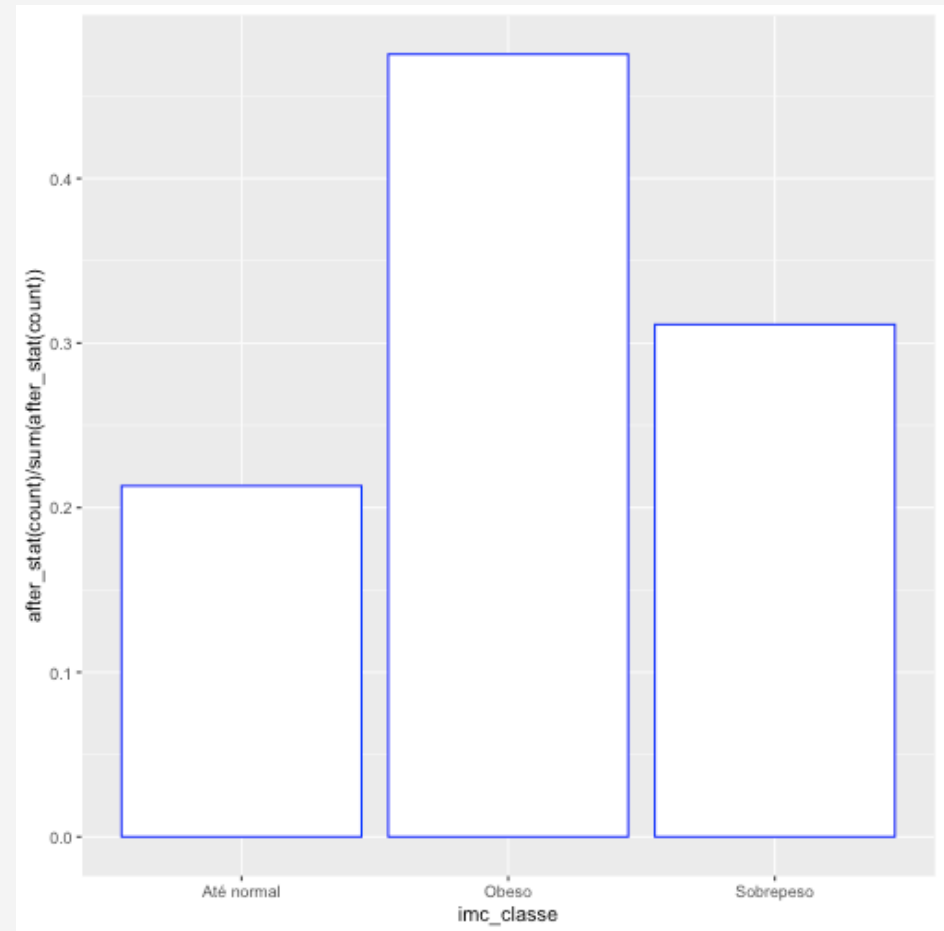
```
ggplot(dados1, aes(x = imc_classe)) +  
  geom_bar()
```



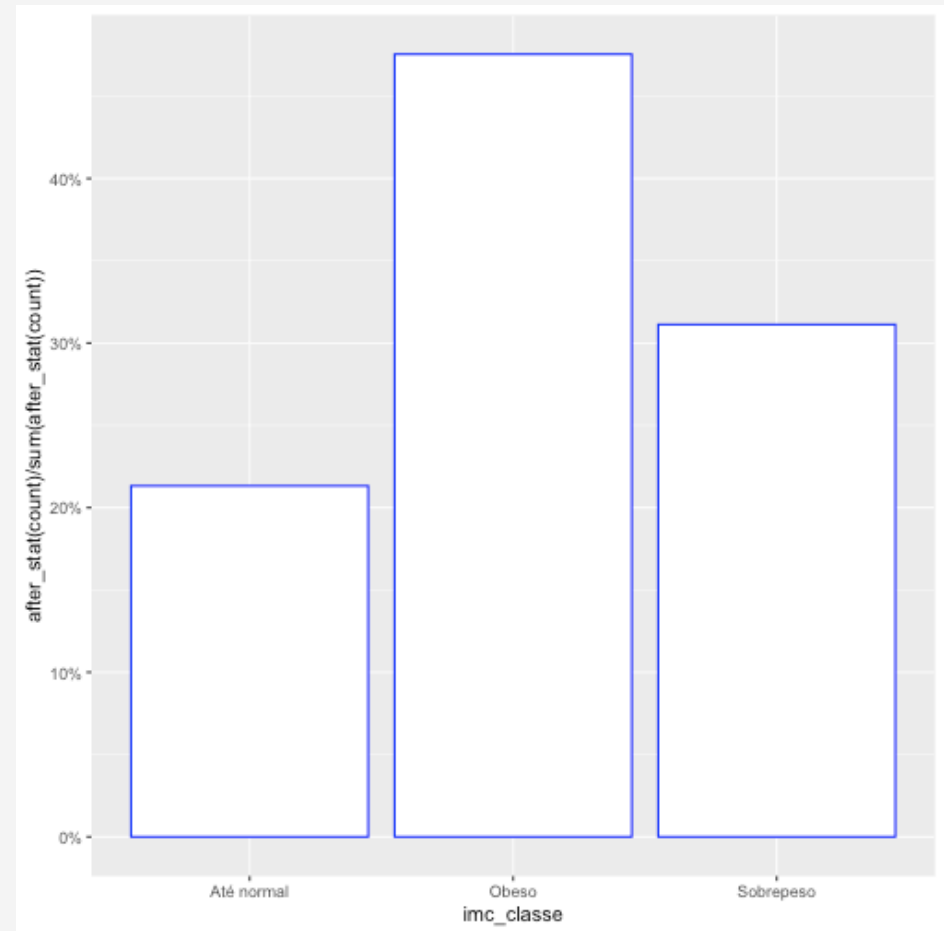
```
ggplot(dados1, aes(x = imc_classe, y = a  
geom_bar()
```



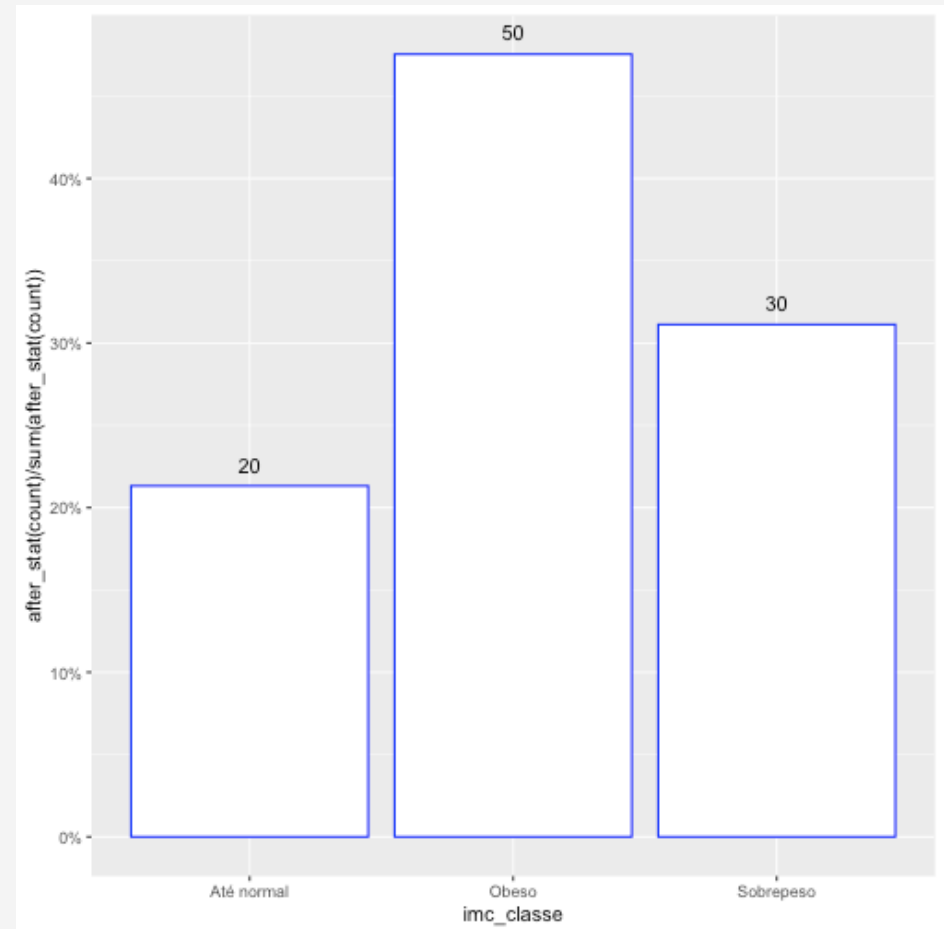
```
ggplot(dados1, aes(x = imc_classe, y = a  
geom_bar(color = "blue", fill = "white"
```



```
ggplot(dados1, aes(x = imc_classe, y = a  
  geom_bar(color = "blue", fill = "white"  
  scale_y_continuous(labels = scales::p
```



```
ggplot(dados1, aes(x = imc_classe, y = a
  geom_bar(color = "blue", fill = "white
  scale_y_continuous(labels = scales::p
  geom_text(
    stat = "count",
    aes(label = round(after_stat(count),
    vjust = -1)
  )
```

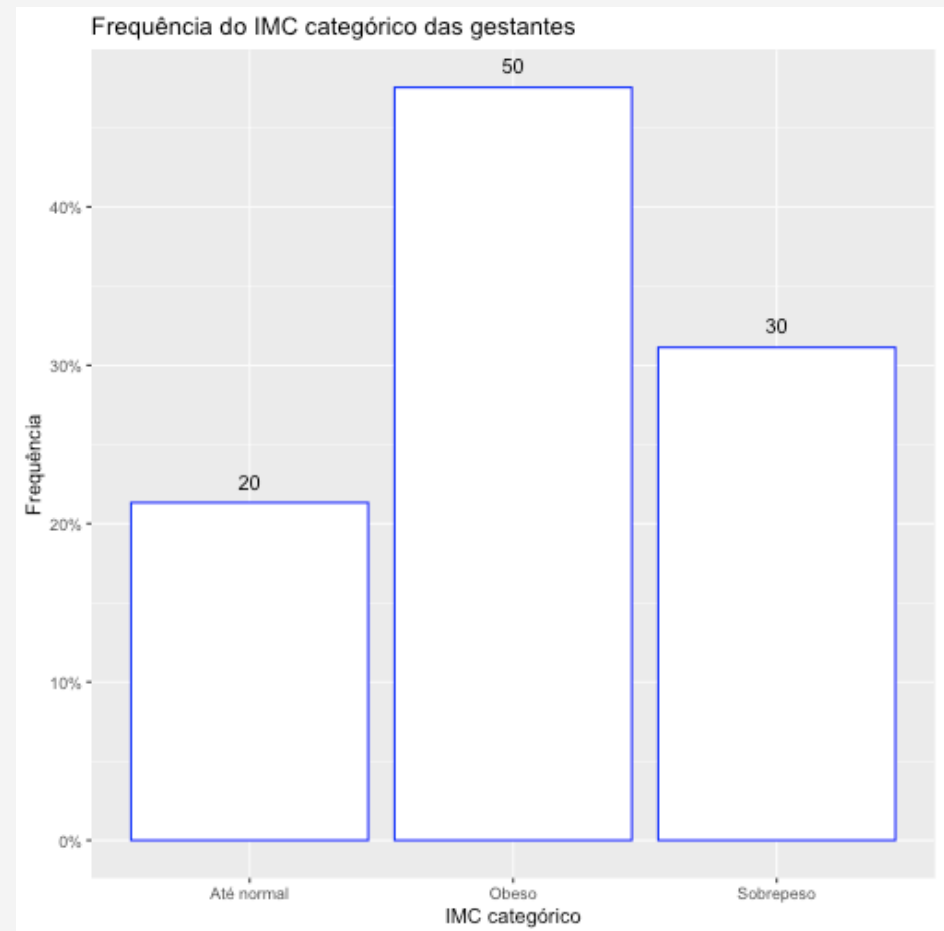




```

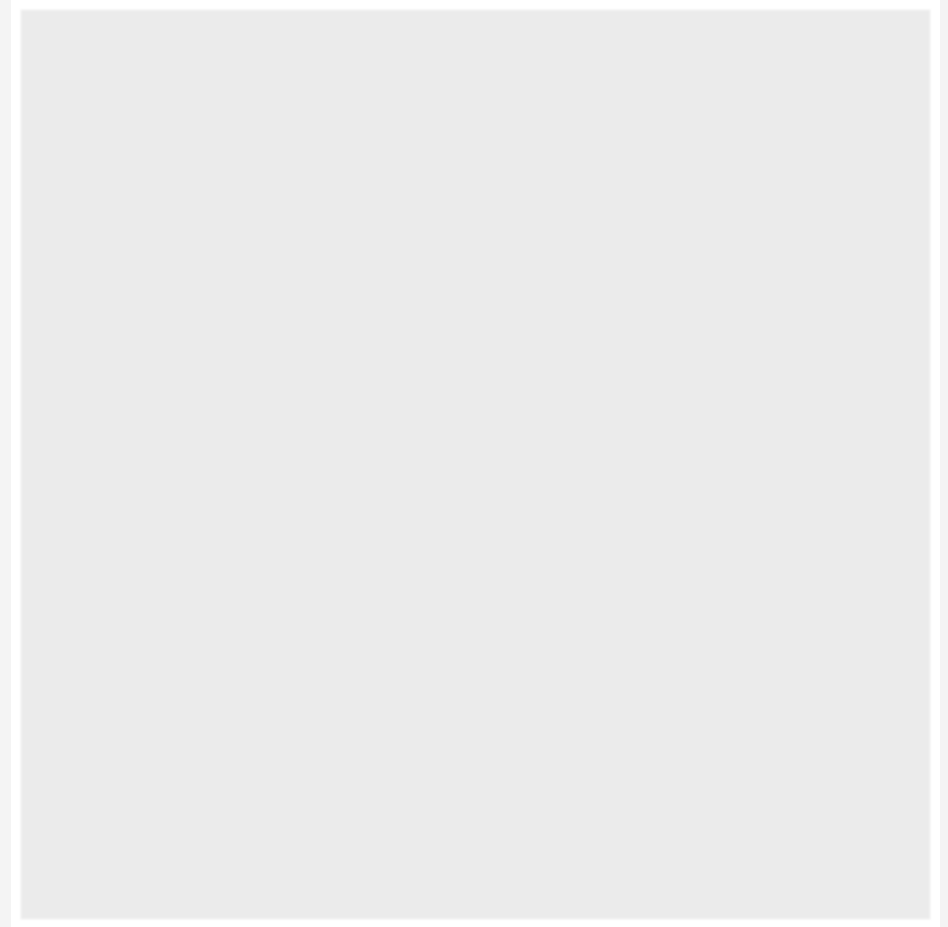
ggplot(dados1, aes(x = imc_classe, y = )) +
  geom_bar(color = "blue", fill = "white") +
  scale_y_continuous(labels = scales::percent) +
  geom_text(
    stat = "count",
    aes(label = round(after_stat(count), 0),
        vjust = -1)
  ) +
  labs(
    title = "Frequência do IMC categórico",
    x = "IMC categórico",
    y = "Frequência"
  )

```

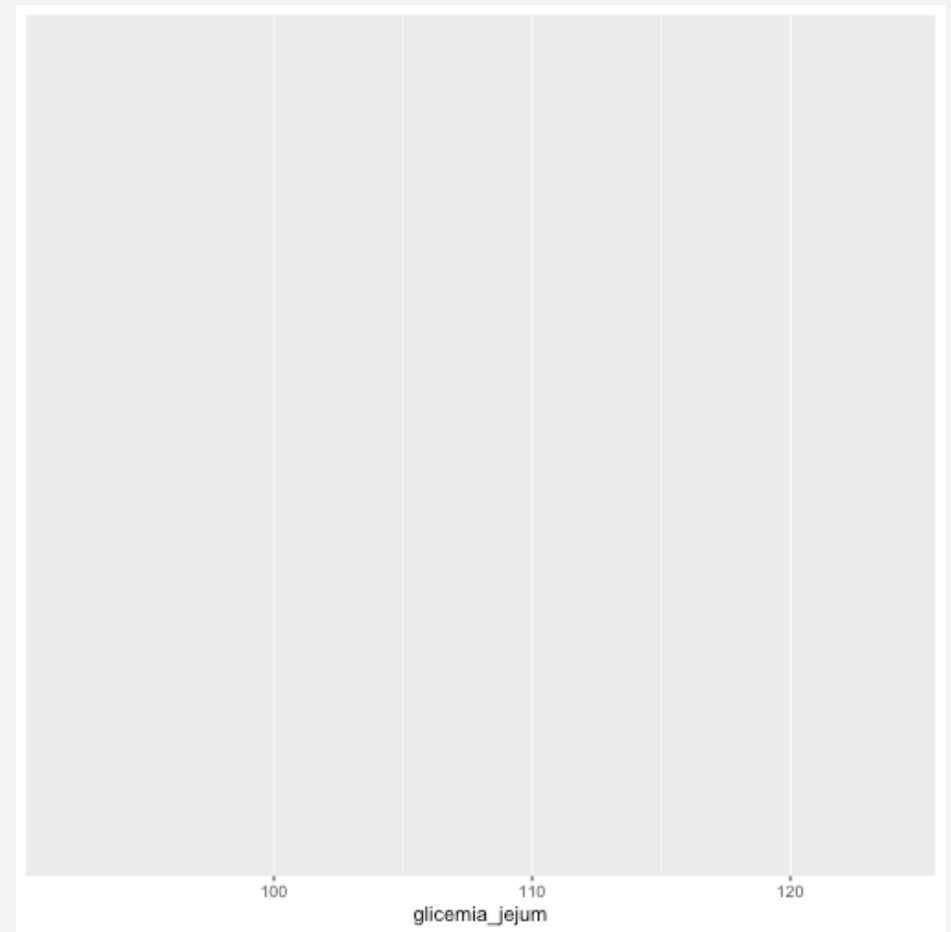


## Histograma

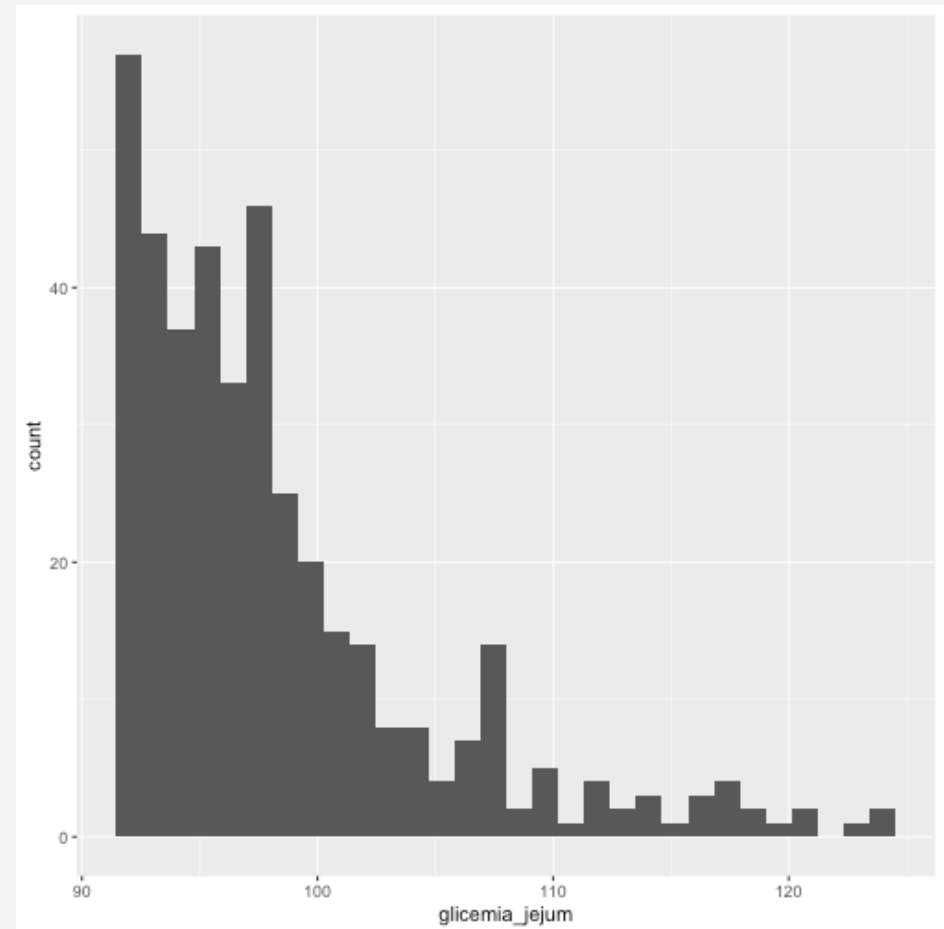
```
ggplot(dados1)
```



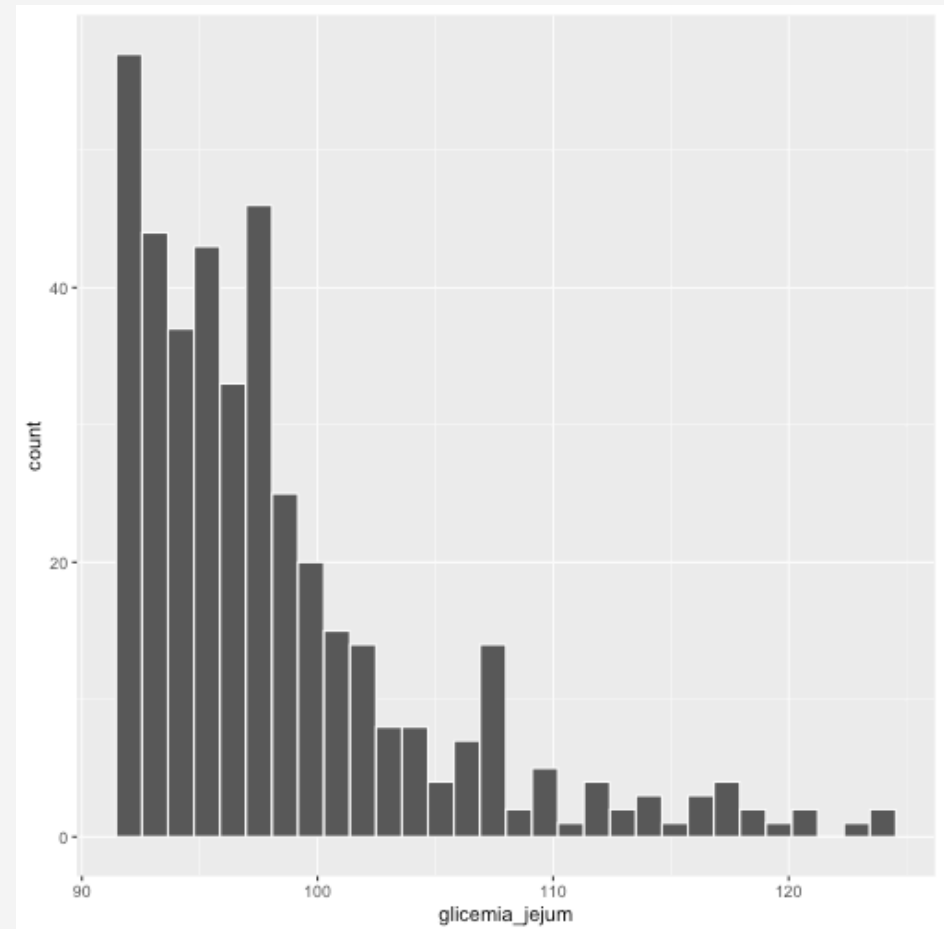
```
ggplot(dados1, aes(x = glicemia_jejum))
```



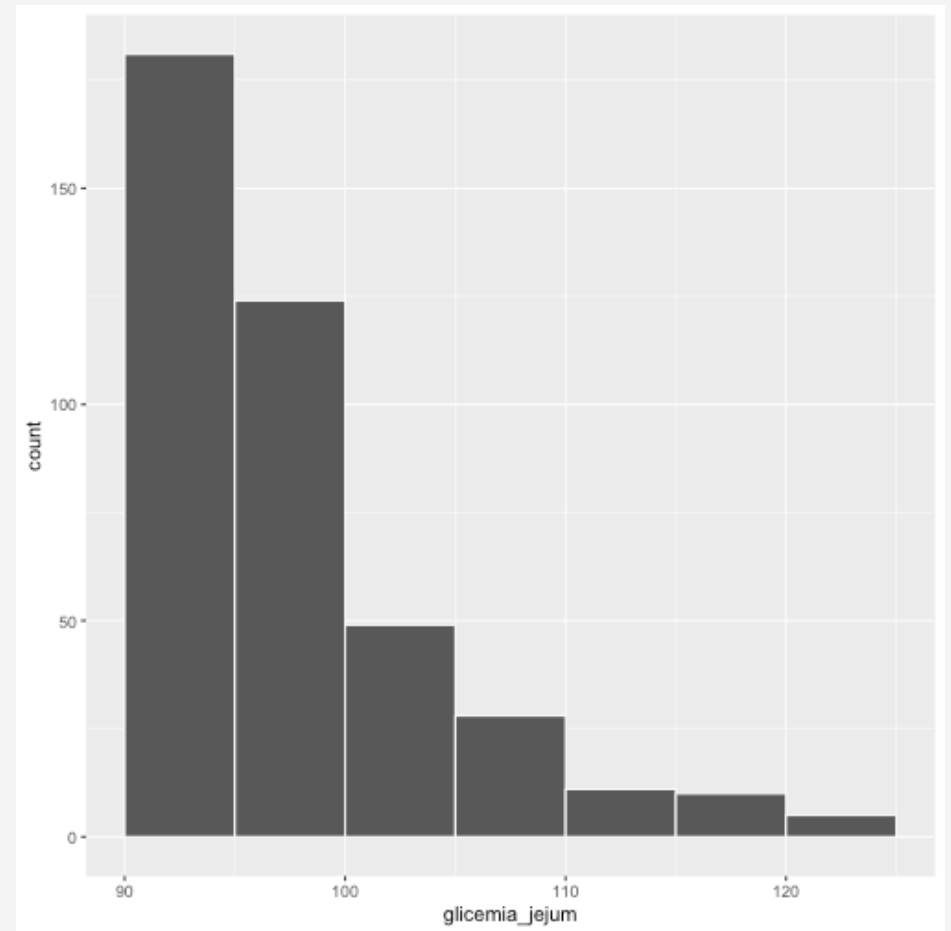
```
ggplot(dados1, aes(x = glicemia_jejum))  
  geom_histogram()
```



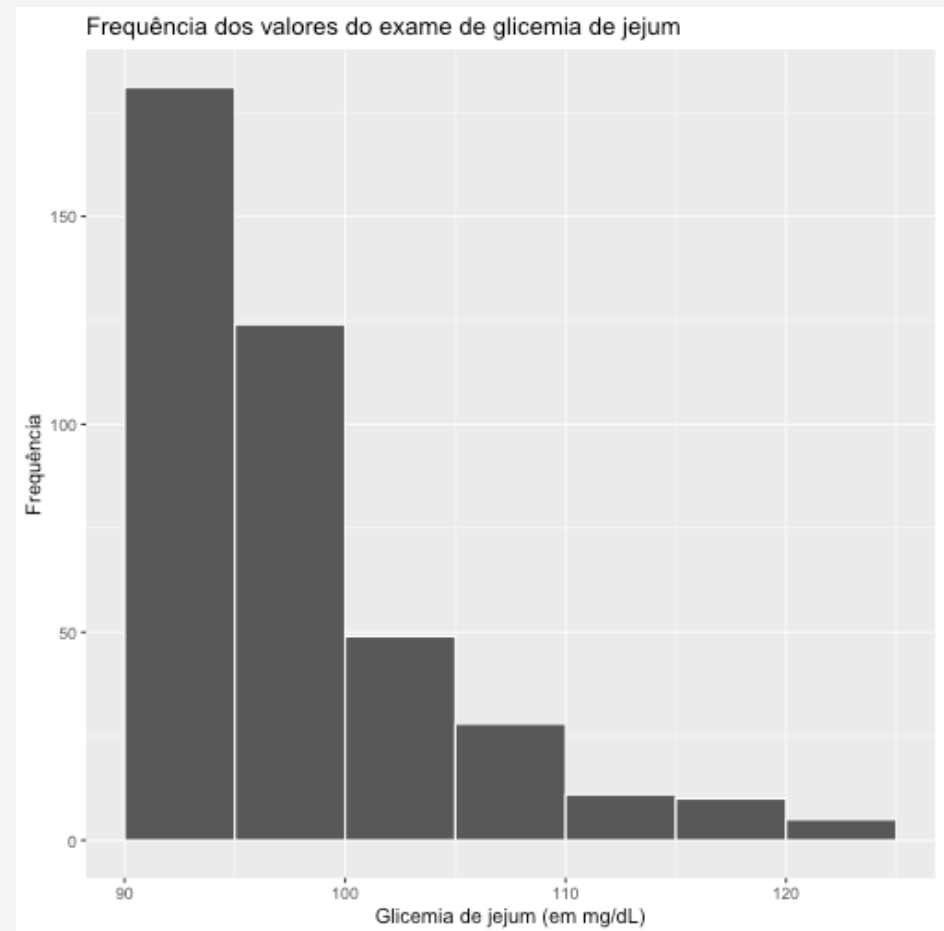
```
ggplot(dados1, aes(x = glicemia_jejum))  
  geom_histogram(color = "white")
```



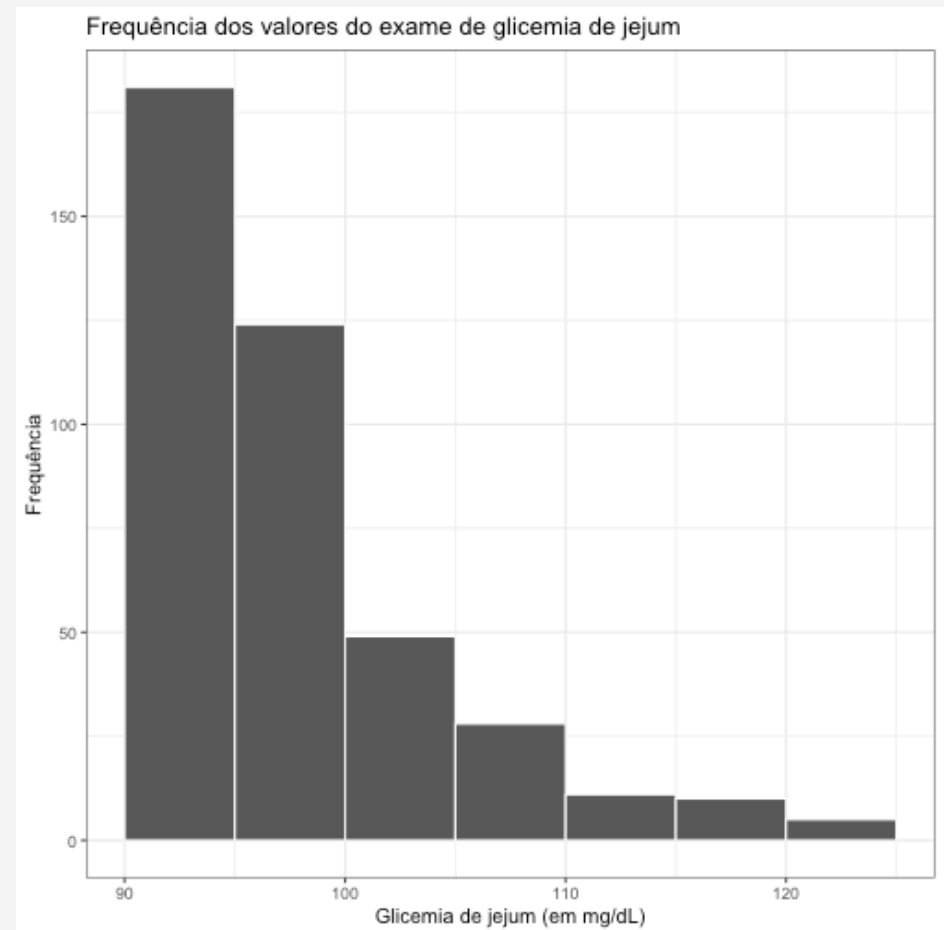
```
ggplot(dados1, aes(x = glicemia_jejum))  
  geom_histogram(color = "white", breaks
```



```
ggplot(dados1, aes(x = glicemia_jejum))  
  geom_histogram(color = "white", breaks  
  labs(  
    title = "Frequência dos valores do e  
    x = "Glicemia de jejum (em mg/dL)",  
    y = "Frequência"  
  )
```



```
ggplot(dados1, aes(x = glicemia_jejum))
  geom_histogram(color = "white", breaks
  labs(
    title = "Frequência dos valores do e
    x = "Glicemia de jejum (em mg/dL)",
    y = "Frequência"
  ) +
  theme_bw()
```

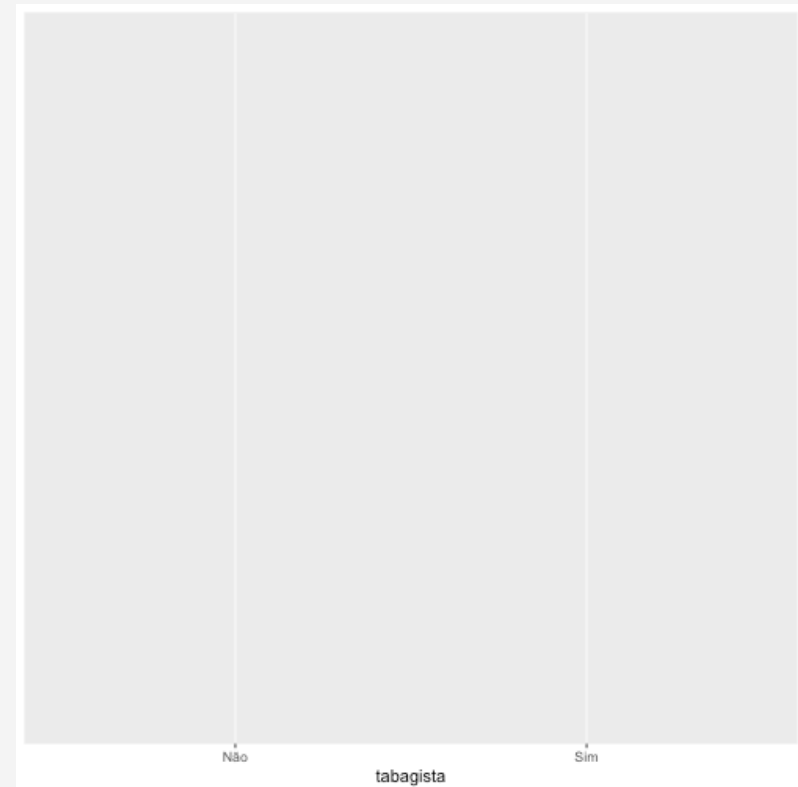




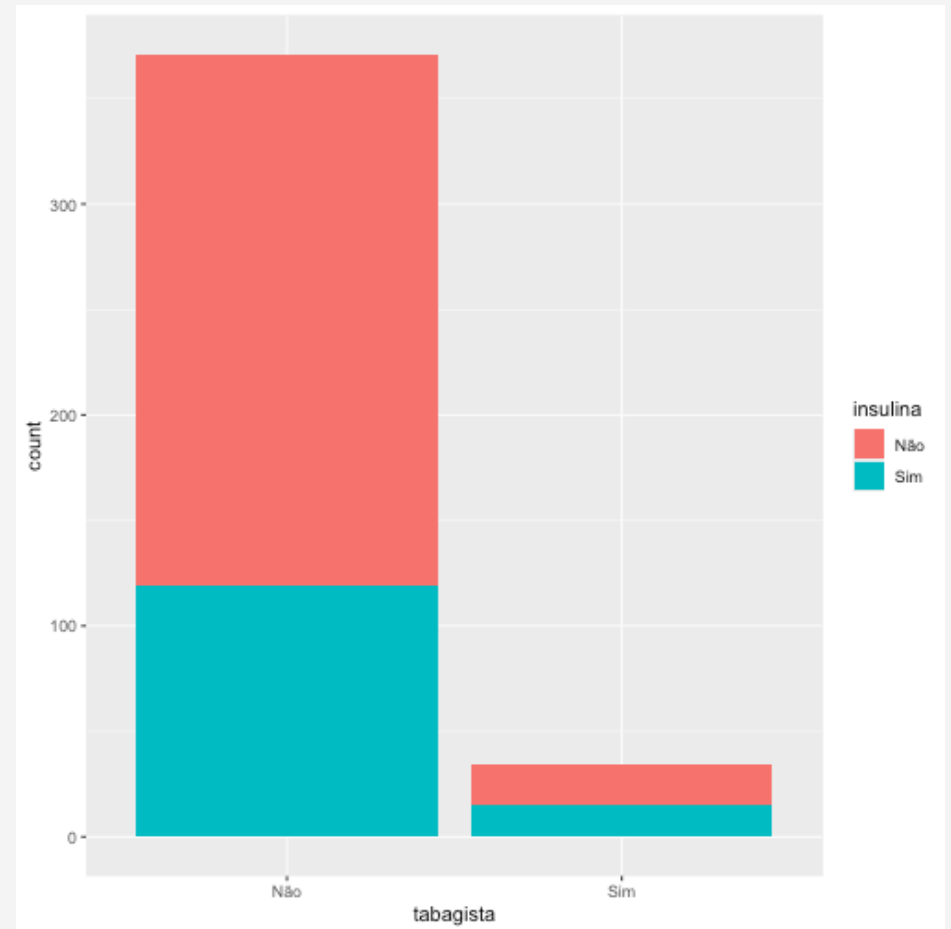
# Gráficos bivariados (qualitativas x qualitativas)

## Gráfico de barras agrupadas

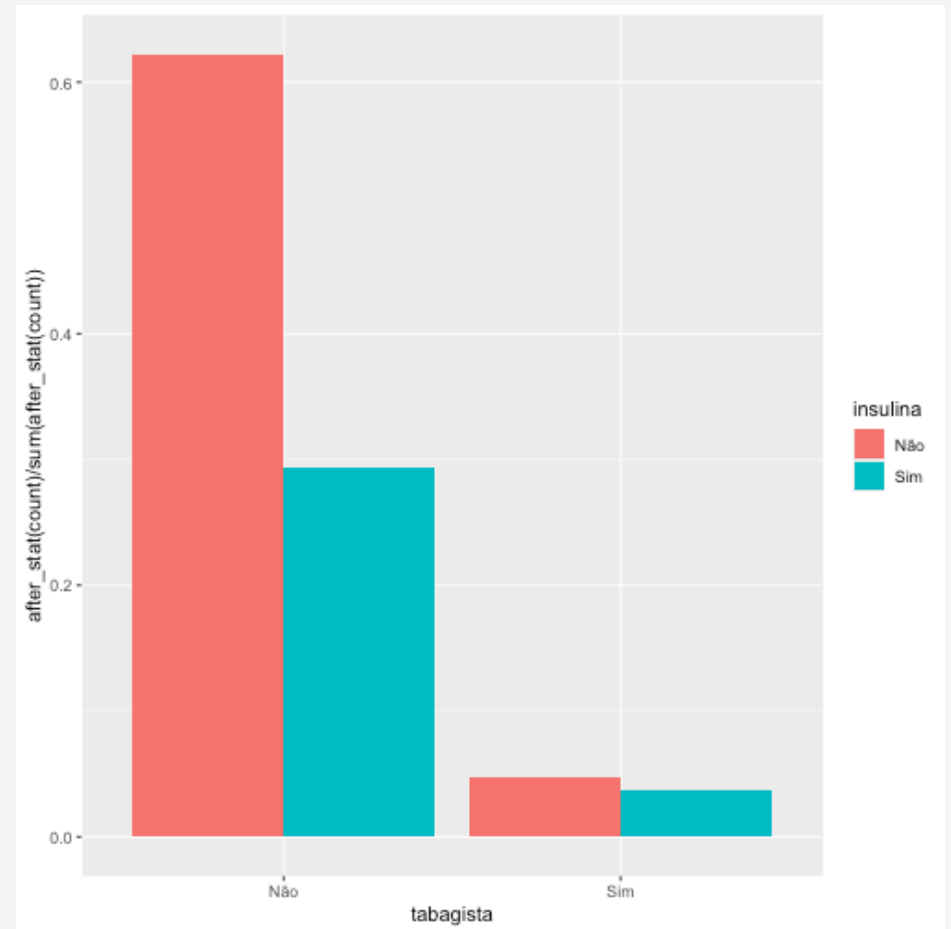
```
dados1 |>  
  dplyr::filter(!is.na(tabagista)) |>  
  ggplot(aes(x = tabagista, fill = insul
```



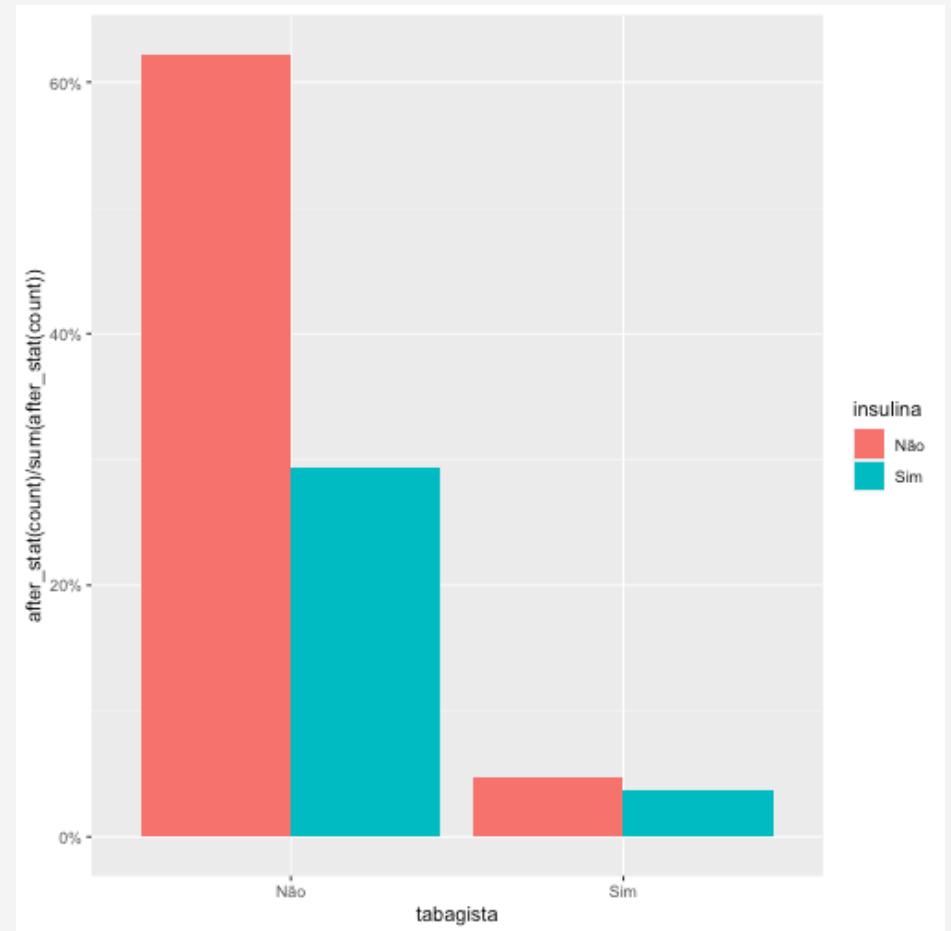
```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul
    geom_bar()
```



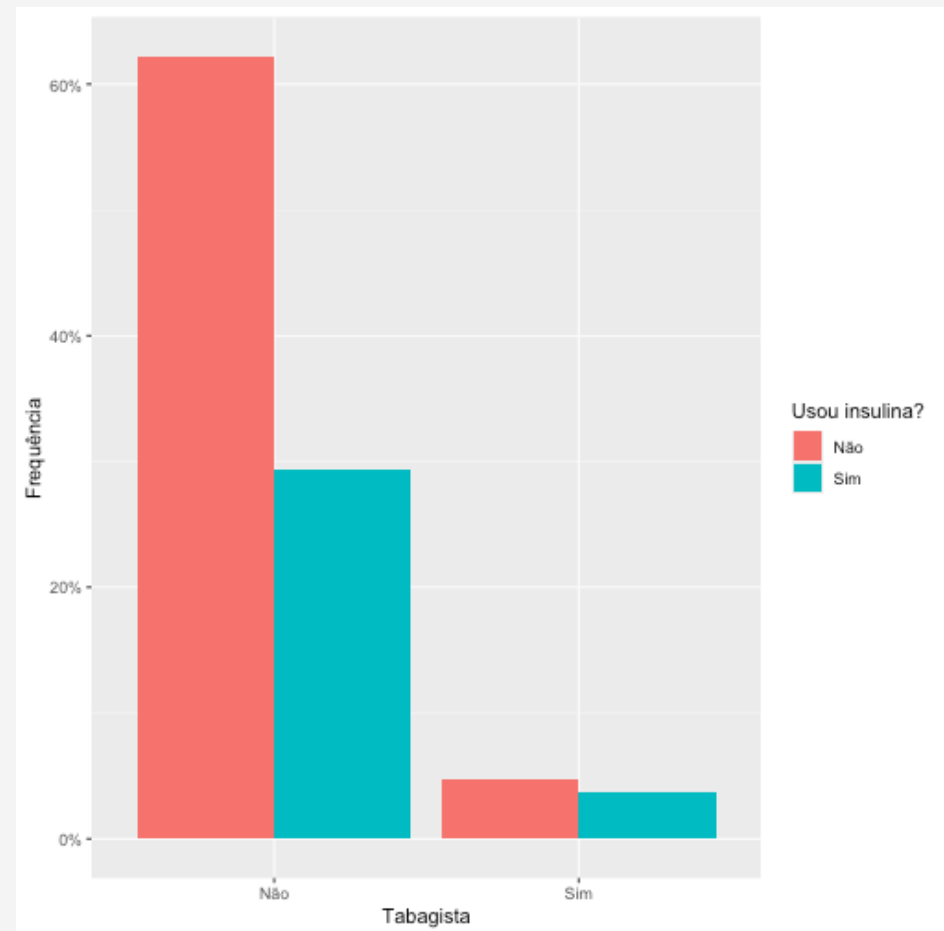
```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insulina))
  geom_bar(aes(y = after_stat(count)/sum(after_stat(count))))
```



```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insulina)) |>
  geom_bar(aes(y = after_stat(count)/sum(after_stat(count)))) |>
  scale_y_continuous(labels = scales::percent())
```



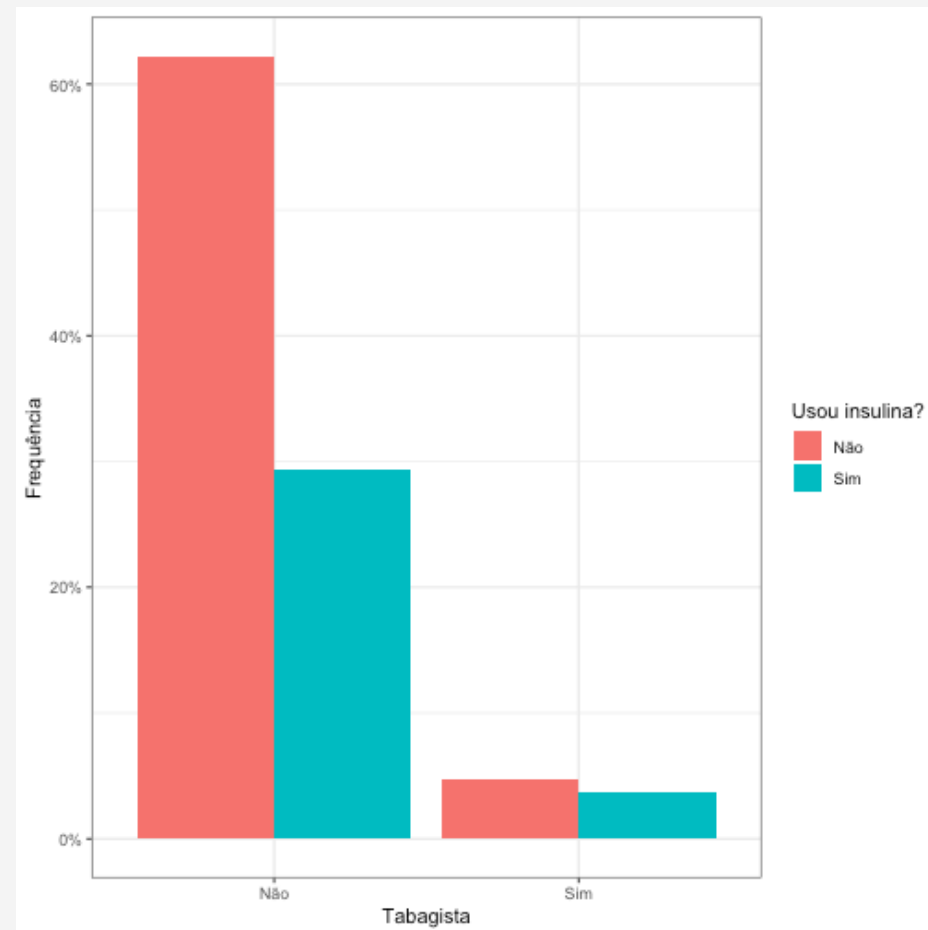
```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul)) +
    geom_bar(aes(y = after_stat(count)/sum(count))) +
    scale_y_continuous(labels = scales::percent()) +
    labs(
      x = "Tabagista",
      y = "Frequência",
      fill = "Usou insulina?"
    )
```



```

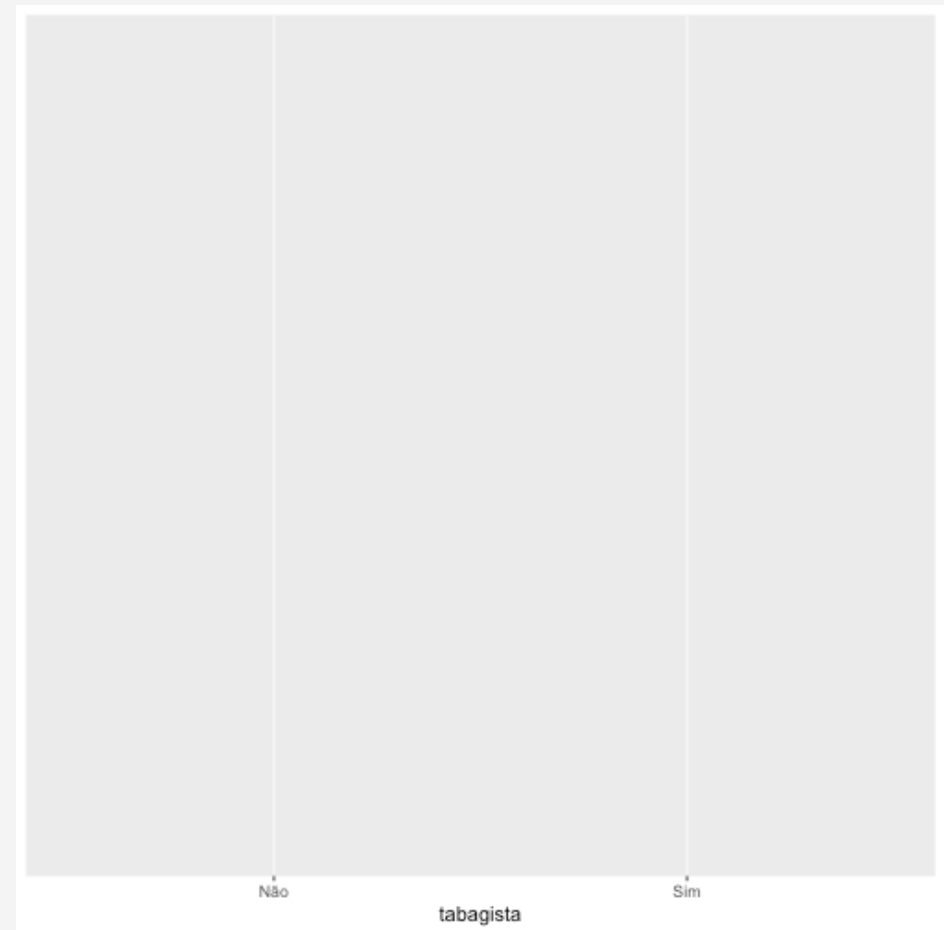
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul
    geom_bar(aes(y = after_stat(count)/s
    scale_y_continuous(labels = scales::
    labs(
      x = "Tabagista",
      y = "Frequência",
      fill = "Usou insulina?"
    ) +
    theme_bw()

```

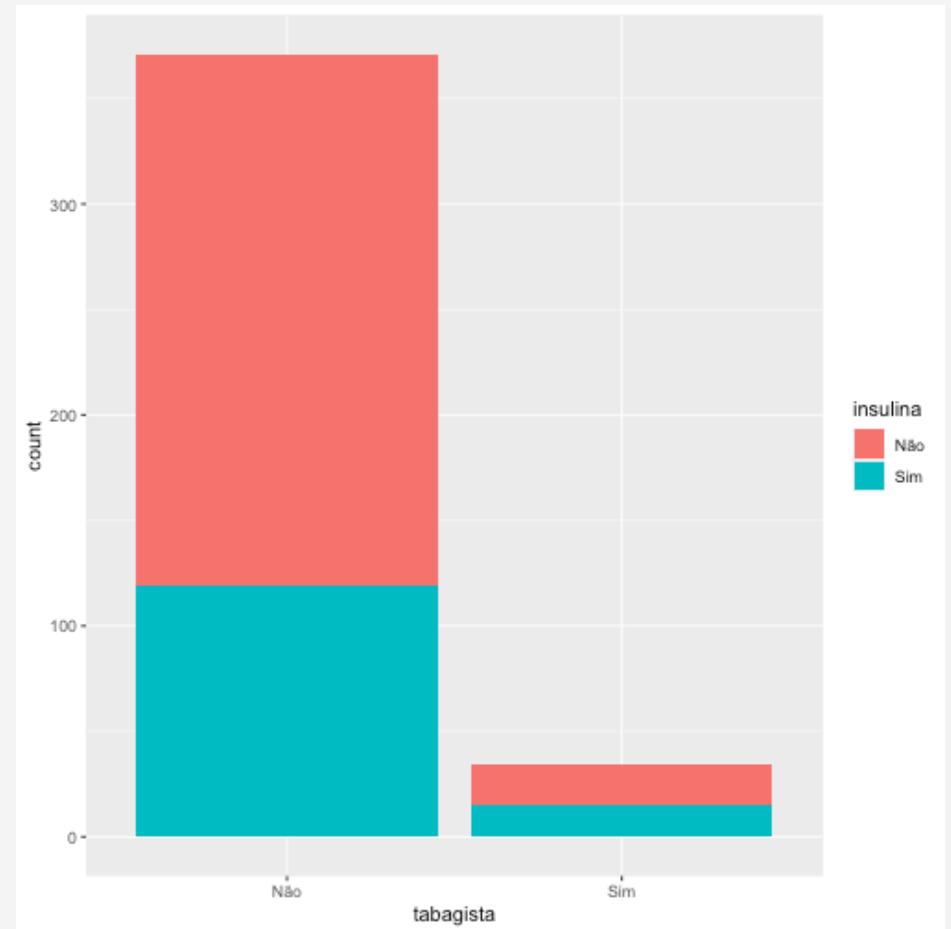


## Gráfico de barras empilhadas

```
dados1 |>  
  dplyr::filter(!is.na(tabagista)) |>  
  ggplot(aes(x = tabagista, fill = insul
```

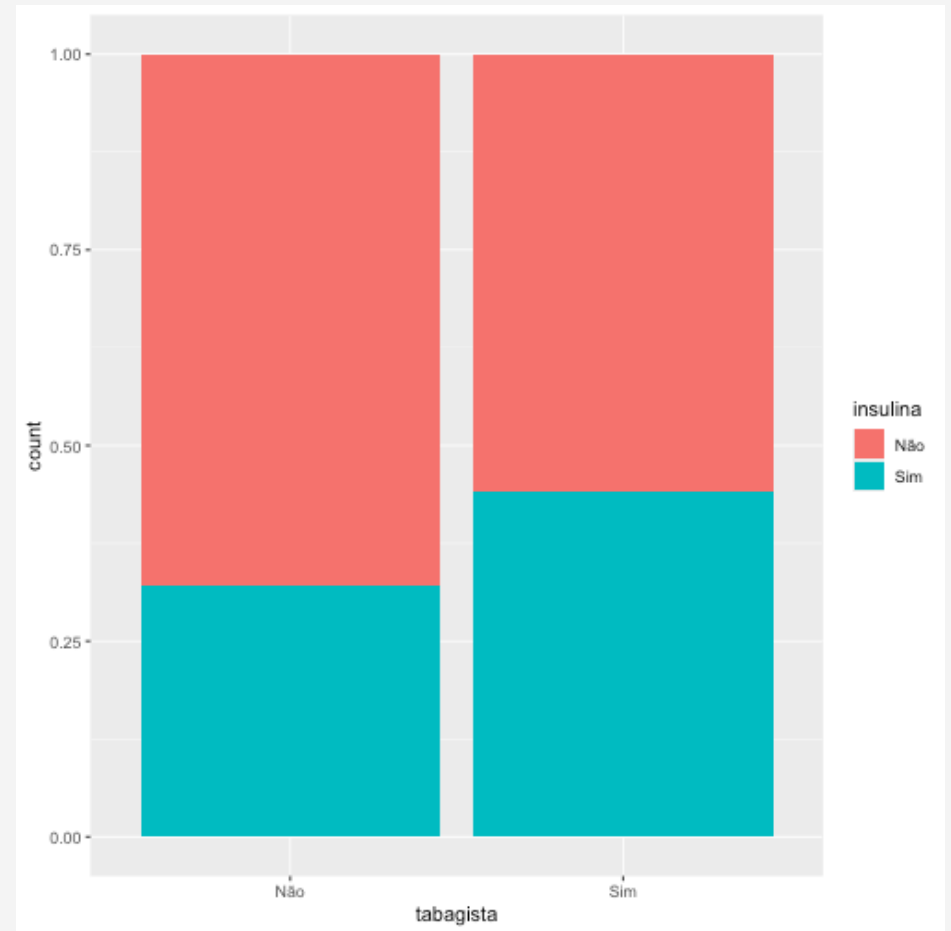


```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul
  geom_bar(position = "stack")
```

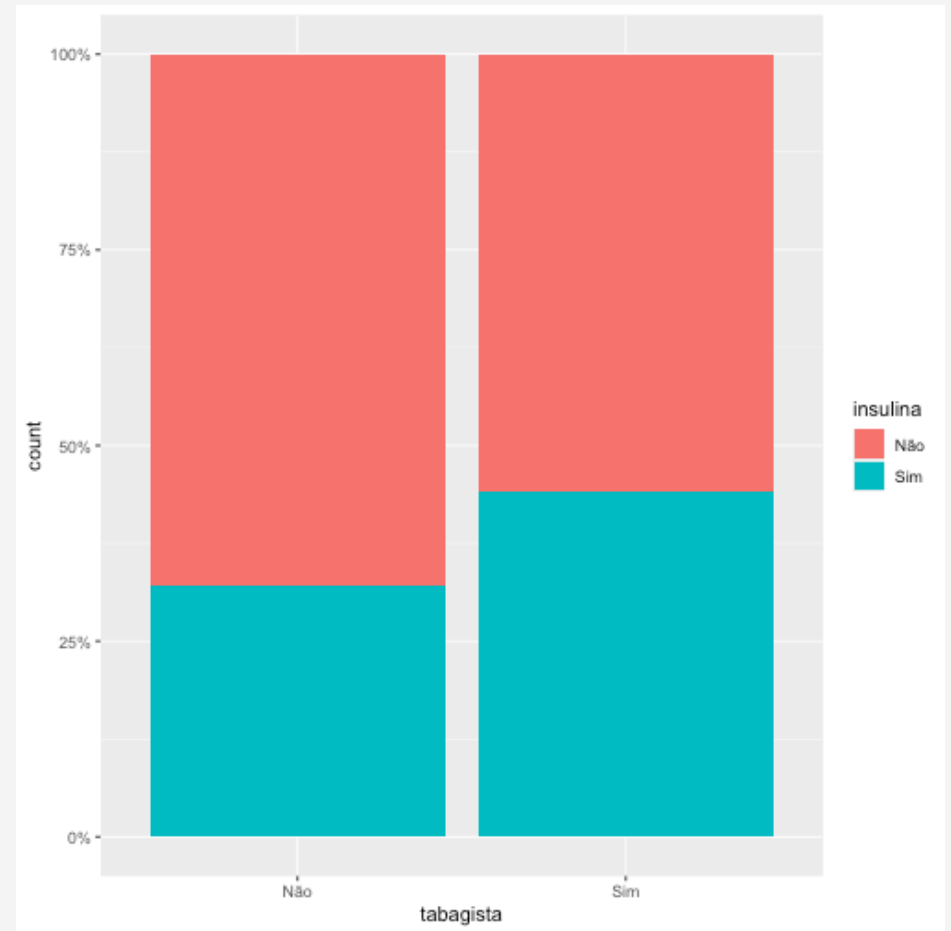




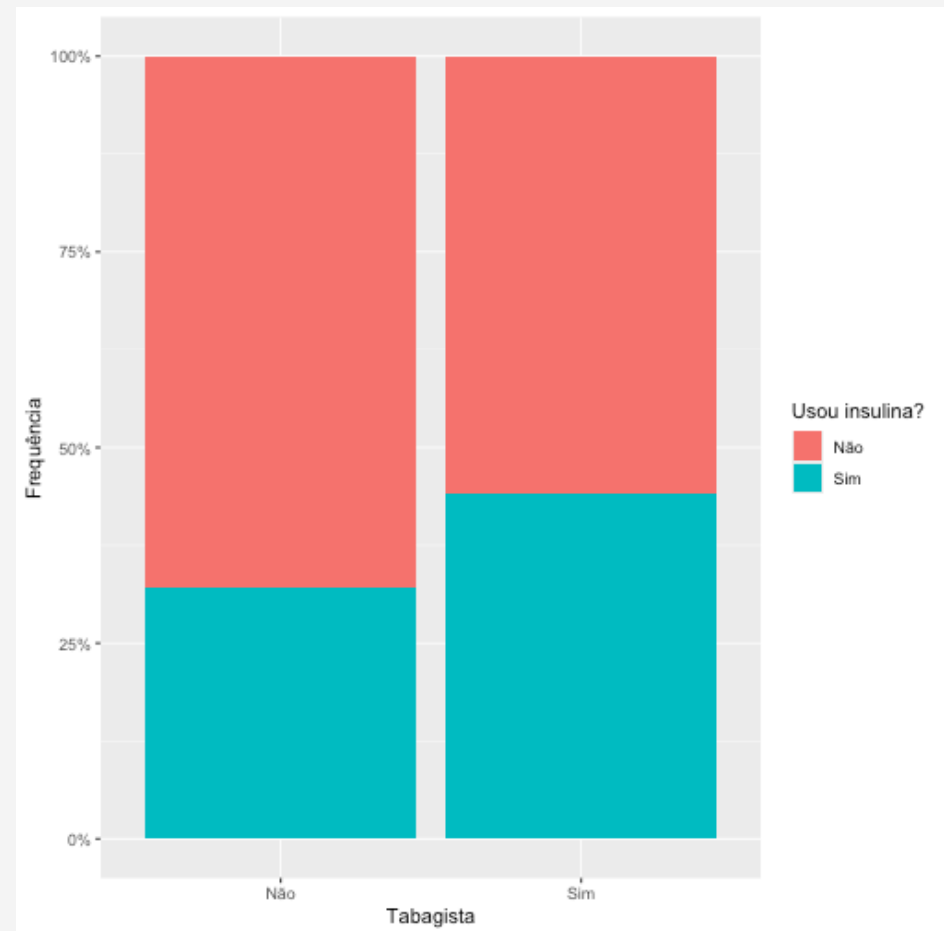
```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul
  geom_bar(position = "fill")
```



```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insulina)) +
    geom_bar(position = "fill") +
    scale_y_continuous(labels = scales::percent)
```



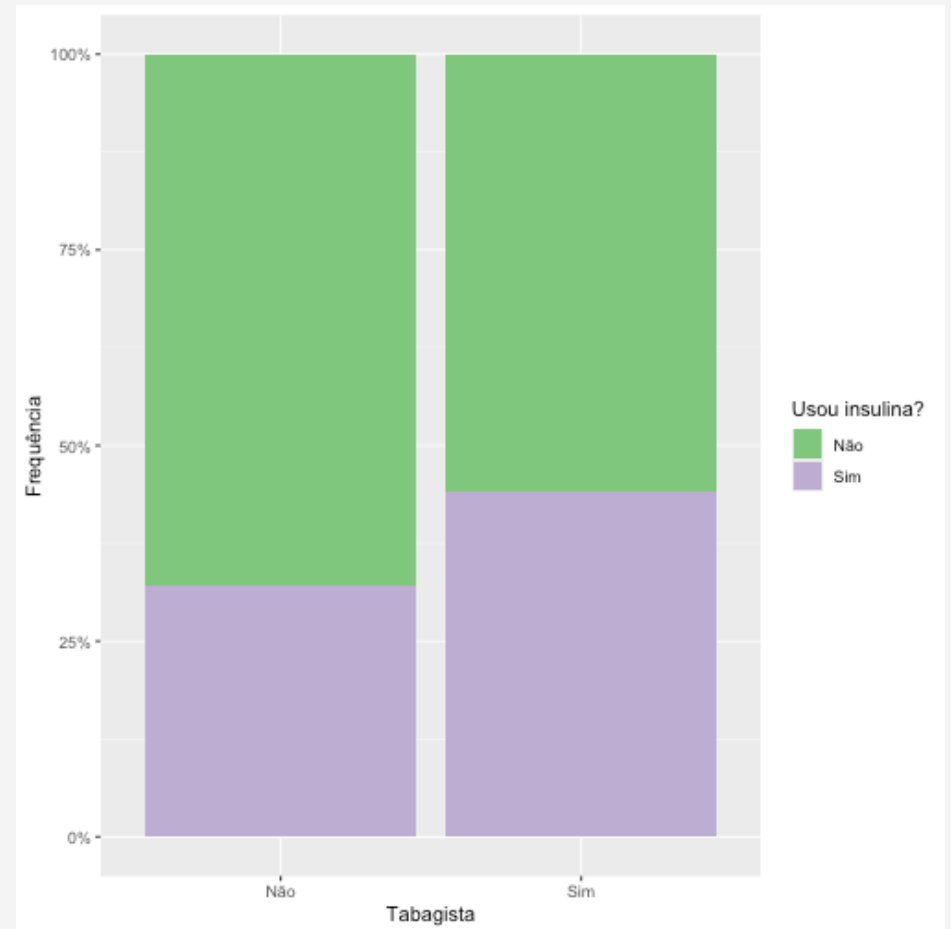
```
dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insul)) +
    geom_bar(position = "fill") +
    scale_y_continuous(labels = scales::percent) +
    labs(
      x = "Tabagista",
      y = "Frequência",
      fill = "Usou insulina?"
    )
```



```

dados1 |>
  dplyr::filter(!is.na(tabagista)) |>
  ggplot(aes(x = tabagista, fill = insulina)) +
    geom_bar(position = "fill") +
    scale_y_continuous(labels = scales::percent) +
    labs(
      x = "Tabagista",
      y = "Frequência",
      fill = "Usou insulina?"
    ) +
    scale_fill_brewer(palette = "Accent")

```



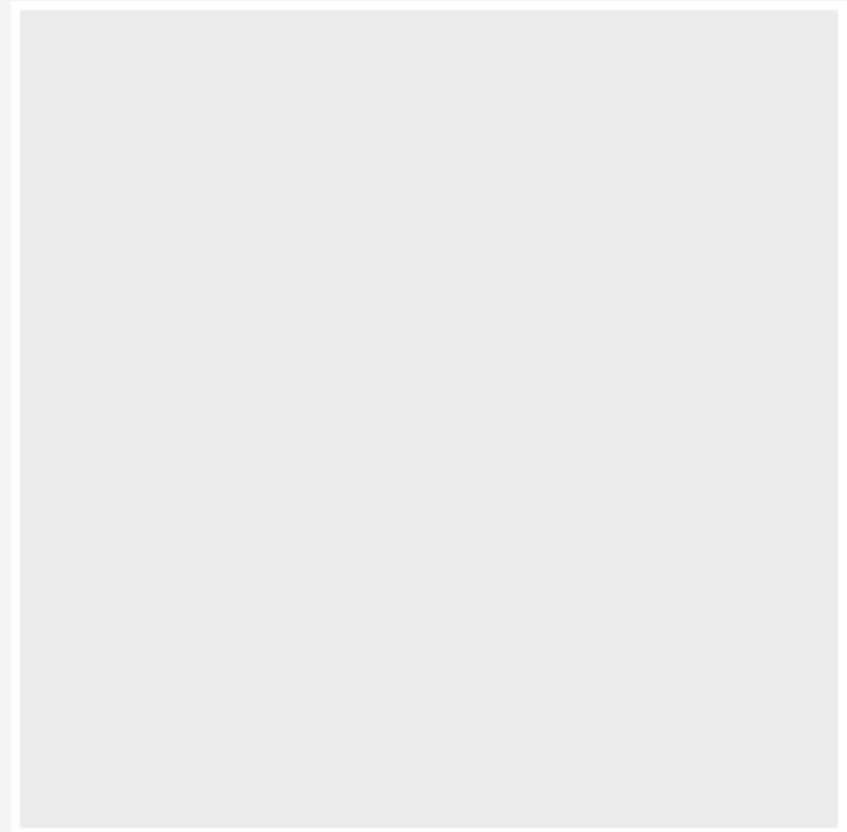
## Dica!

Para ver outras paletas R Color Brewer, [clique aqui!](#)

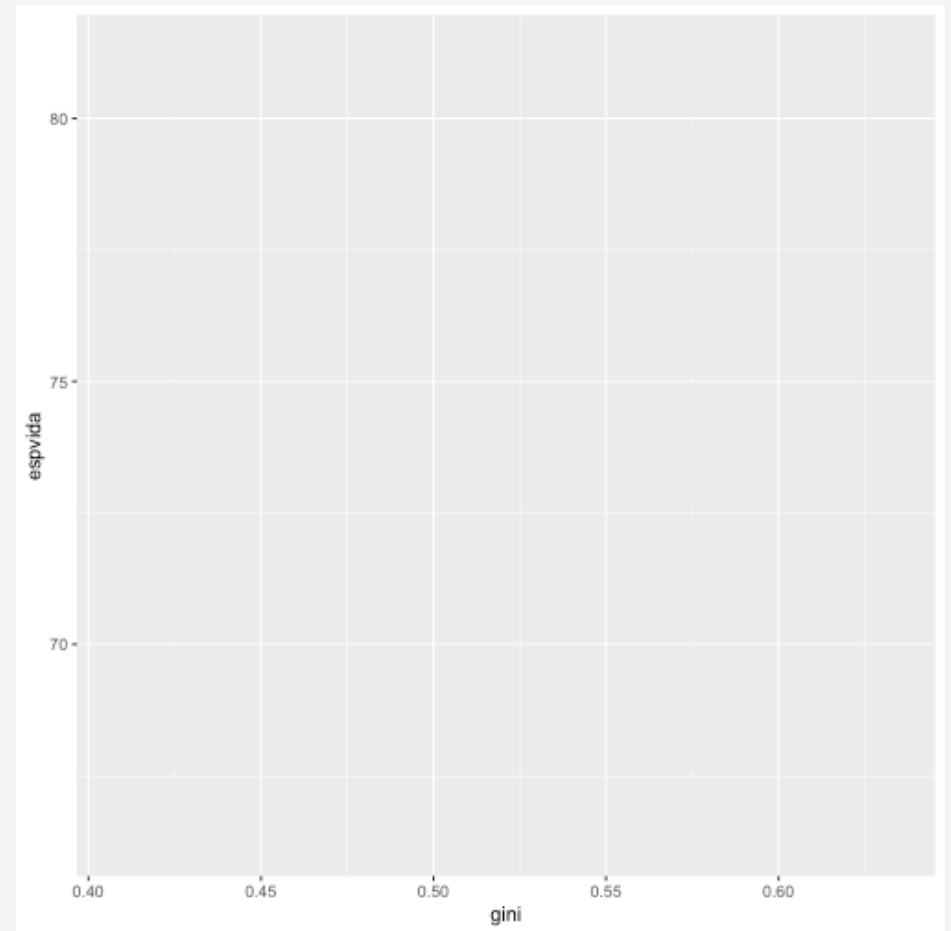
# Gráficos bivariados (quantitativas x quantitativas)

Gráfico de dispersão

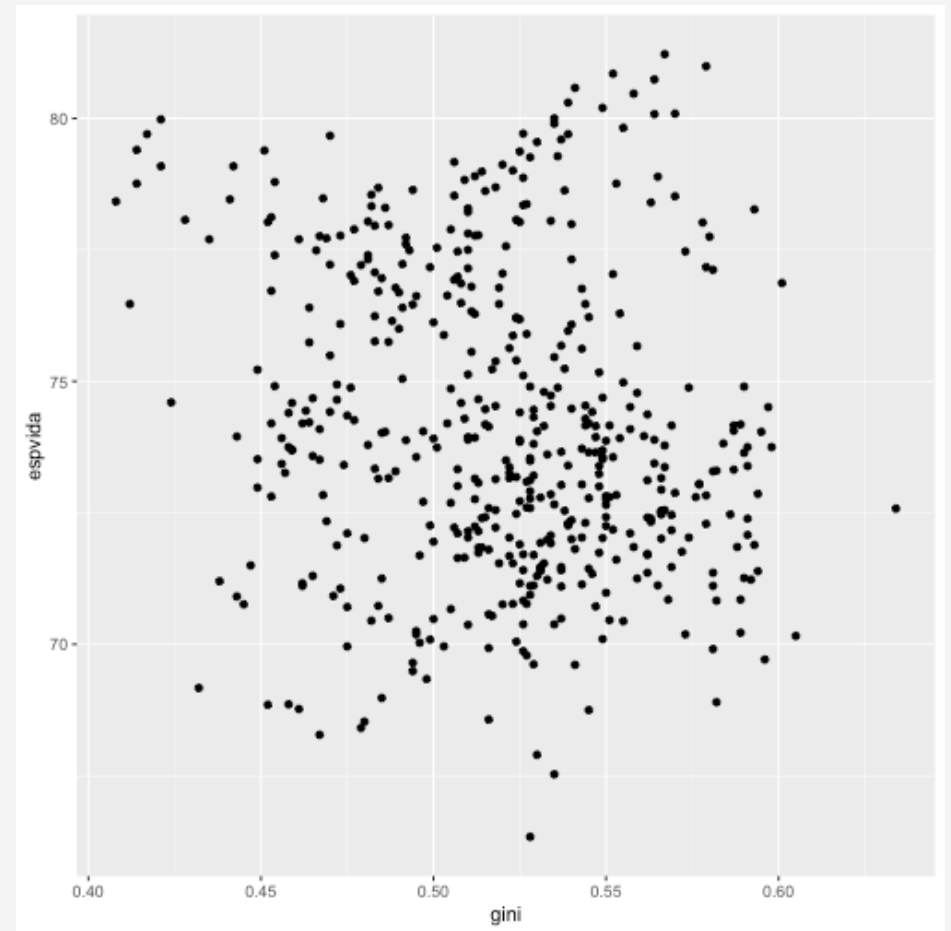
```
ggplot(dados2)
```



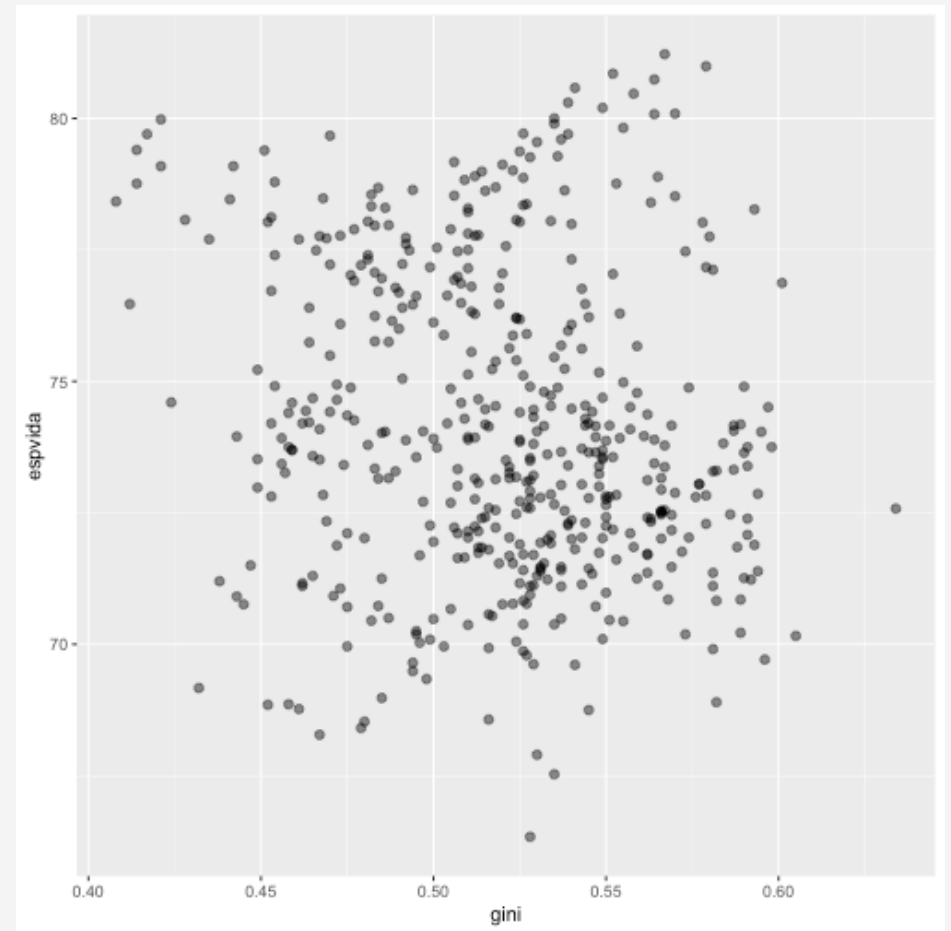
```
ggplot(dados2, aes(x = gini, y = espvida
```



```
ggplot(dados2, aes(x = gini, y = espvida))  
  geom_point()
```

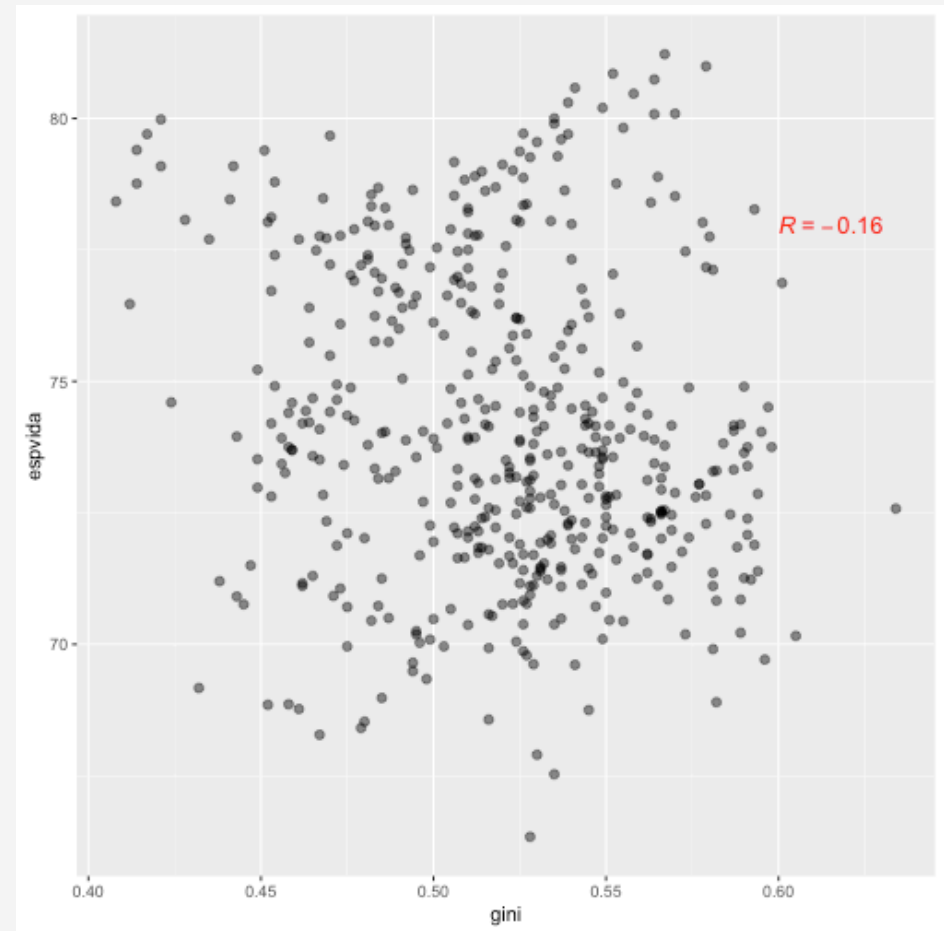


```
ggplot(dados2, aes(x = gini, y = espvida)) +  
  geom_point(size = 2, alpha = .5)
```

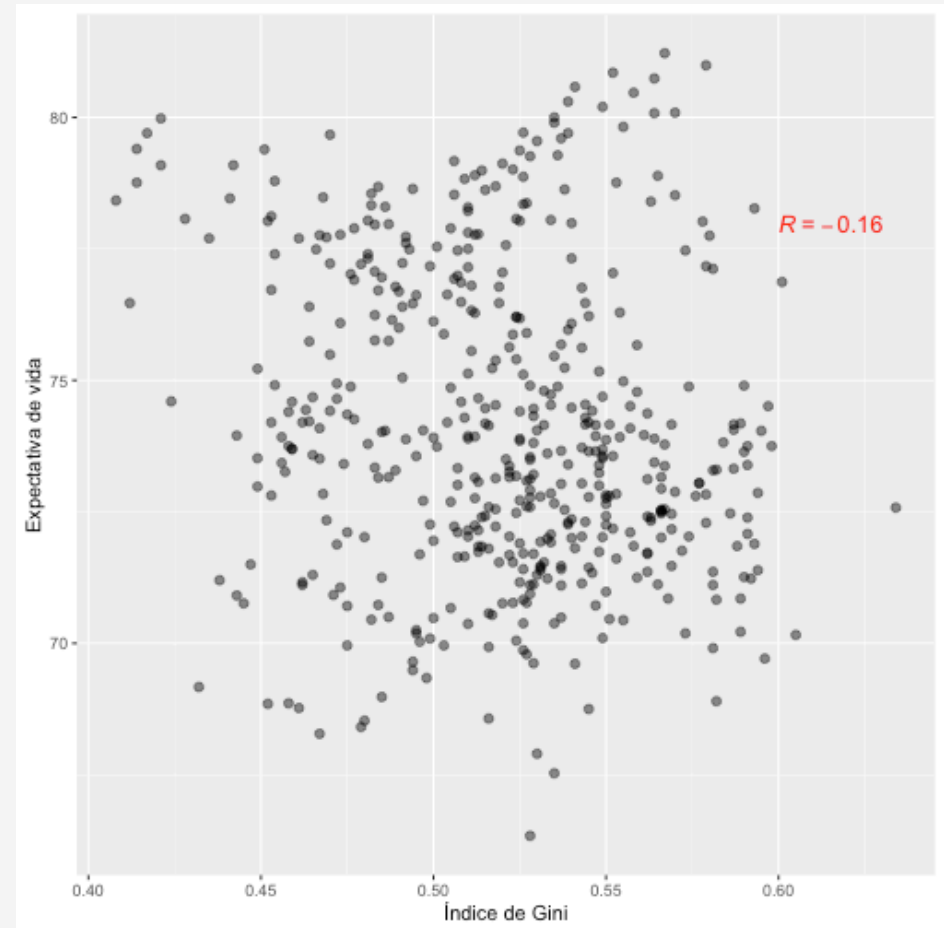




```
ggplot(dados2, aes(x = gini, y = espvida)) +
  geom_point(size = 2, alpha = .5) +
  ggpubr::stat_cor(
    aes(label = after_stat(r.label)), method = "s",
    label.x = 0.6, label.y = 78, size = 12
  )
```



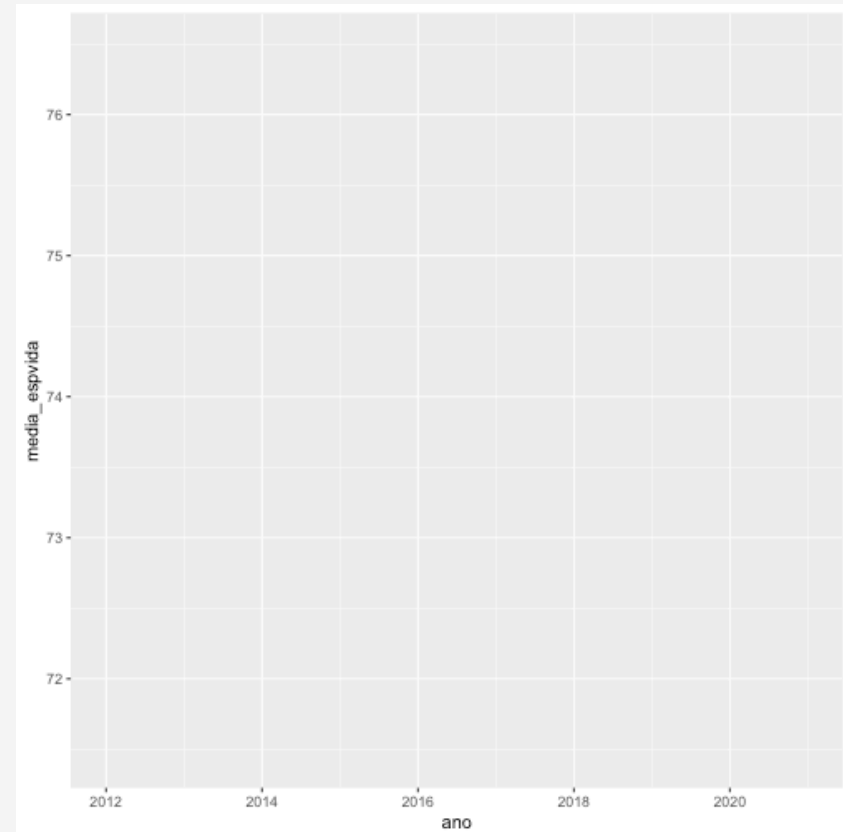
```
ggplot(dados2, aes(x = gini, y = espvida)) +
  geom_point(size = 2, alpha = .5) +
  ggpubr::stat_cor(
    aes(label = after_stat(r.label)), method = "s",
    label.x = 0.6, label.y = 78, size = 10
  ) +
  labs(
    x = "Índice de Gini",
    y = "Expectativa de vida"
  )
```



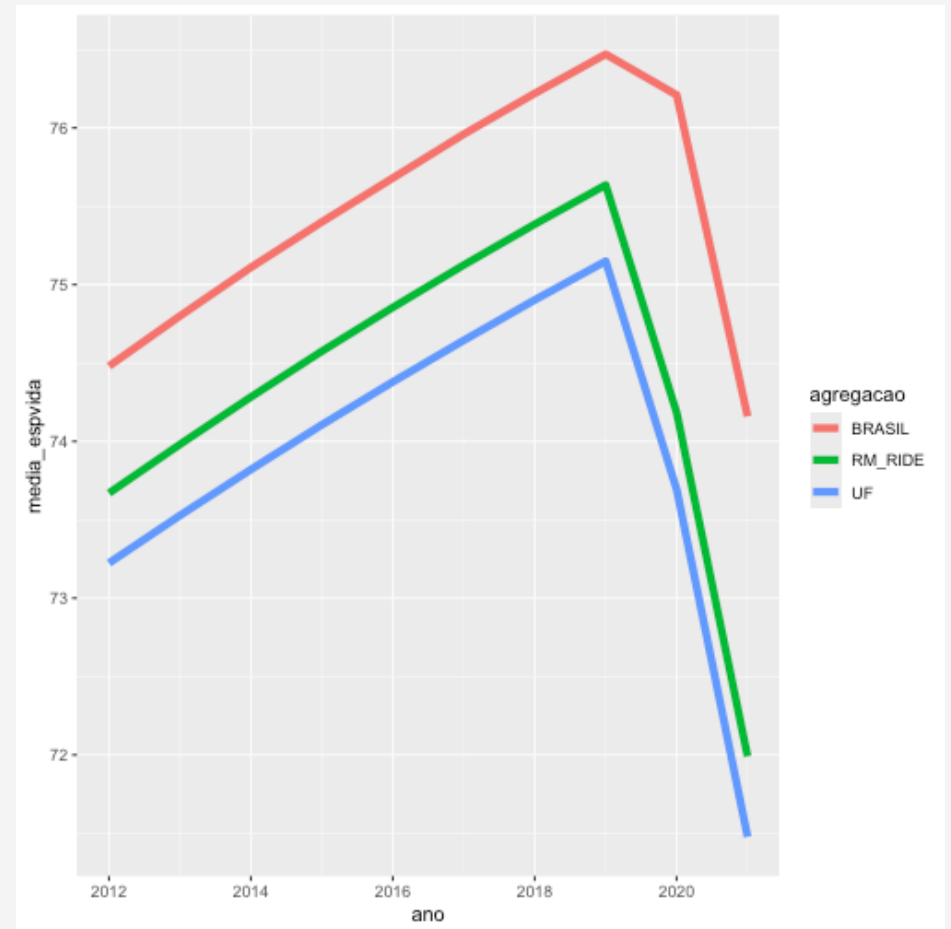
# Gráficos bivariados (qualitativas x quantitativas)

## Gráfico de linhas

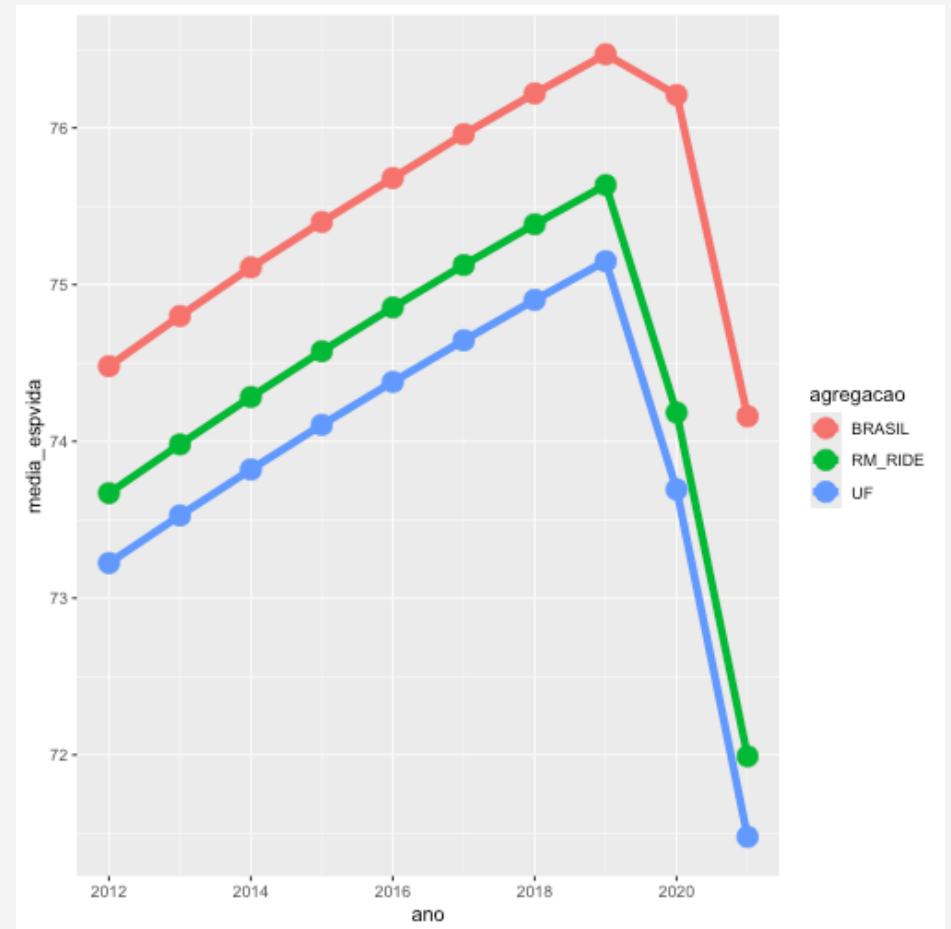
```
dados2 |>  
  dplyr::group_by(ano, agregacao) |>  
  dplyr::summarise(media_espvida = mean(  
    ggplot(aes(x = ano, y = media_espvida,
```



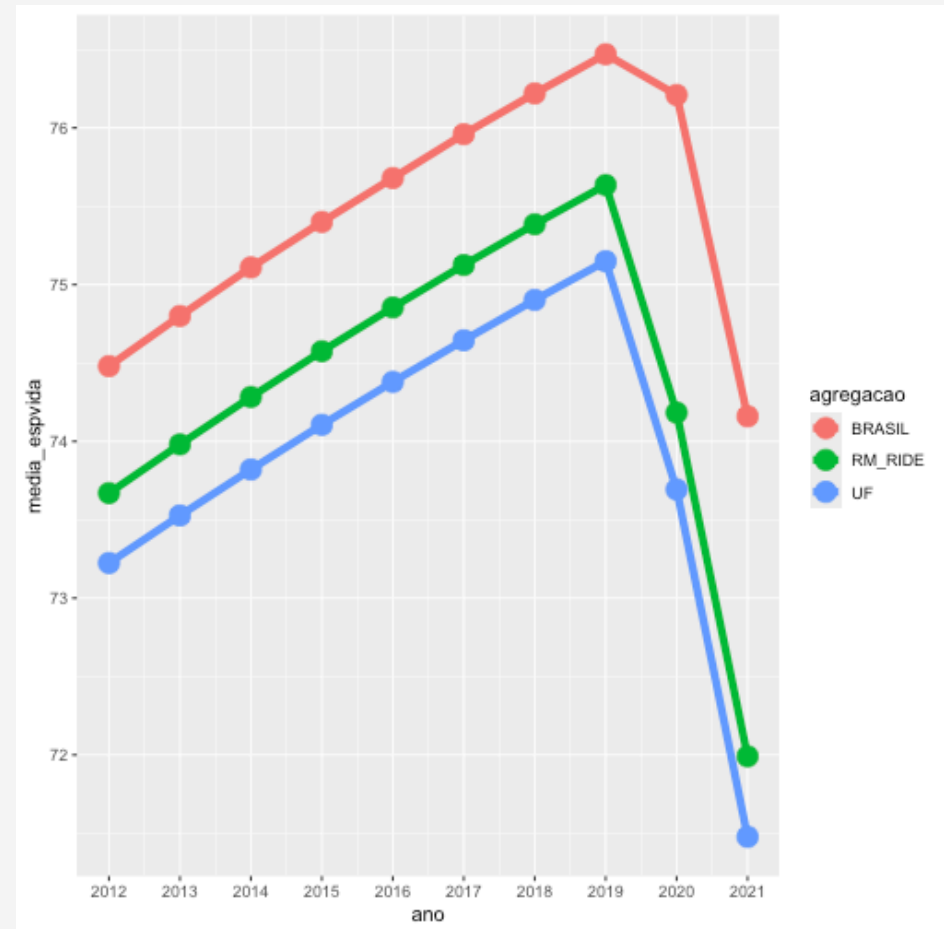
```
dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2)
```



```
dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2) +
      geom_point(size = 5)
```



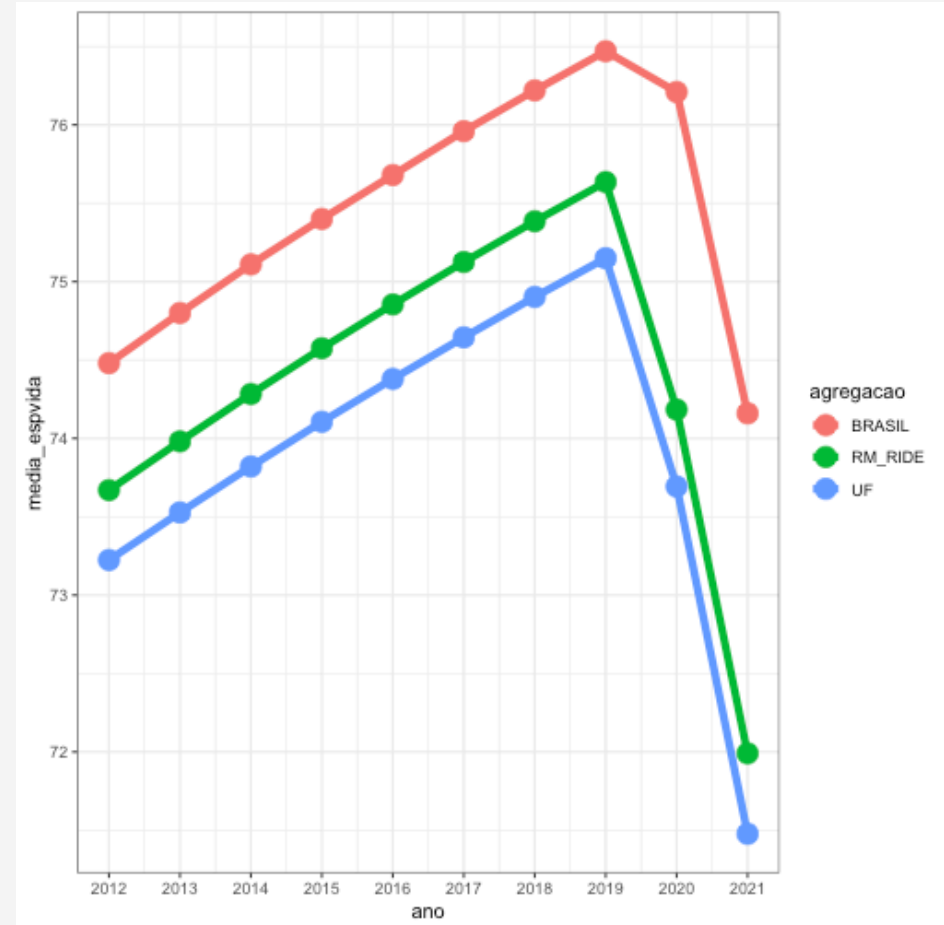
```
dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2) +
      geom_point(size = 5) +
      scale_x_continuous(breaks = seq(2012
```



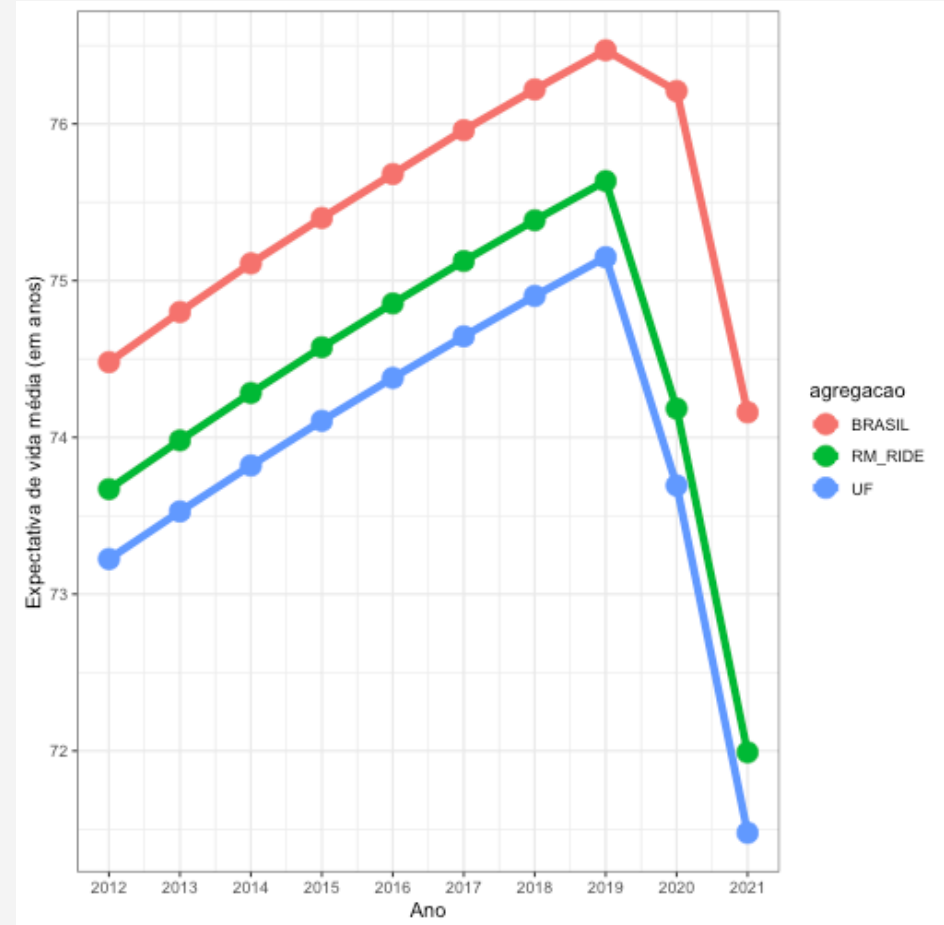
```

dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2) +
      geom_point(size = 5) +
      scale_x_continuous(breaks = seq(2012, 2021, by = 1))
      theme_bw()

```



```
dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2) +
      geom_point(size = 5) +
      scale_x_continuous(breaks = seq(2012, 2021, by = 1))
      theme_bw() +
      labs(
        x = "Ano",
        y = "Expectativa de vida média (em anos)",
        fill = "Agregação"
      )
    )
  )
```

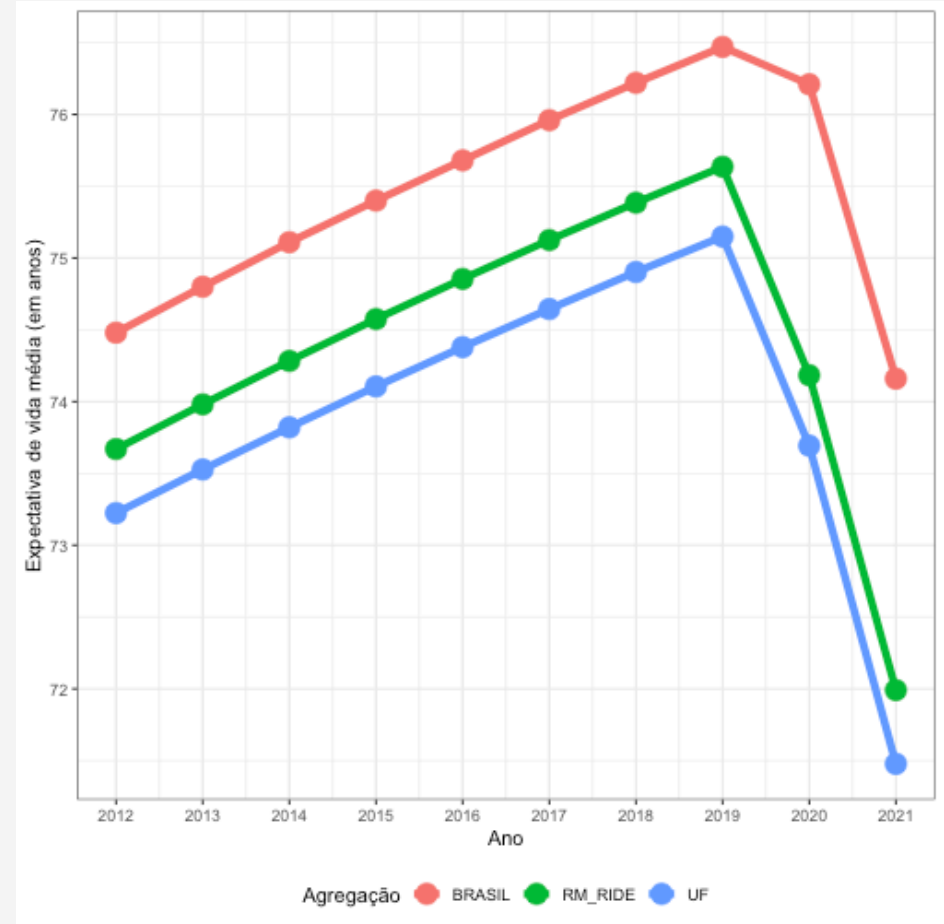




```

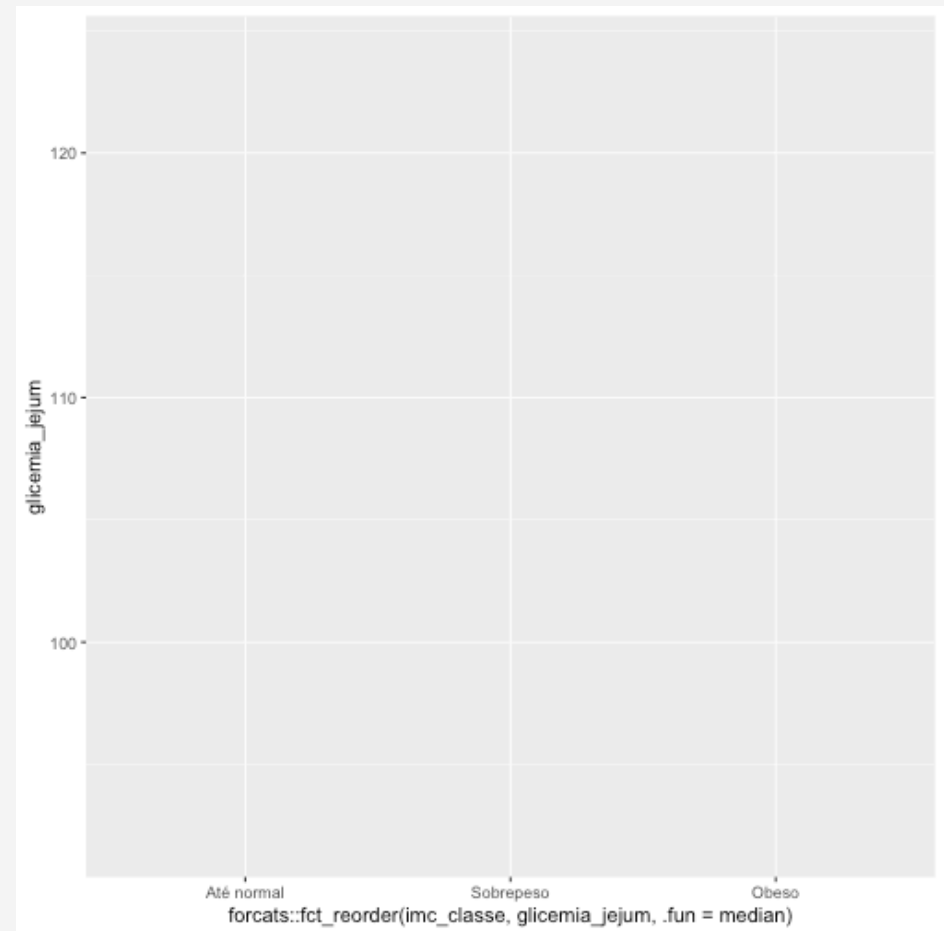
dados2 |>
  dplyr::group_by(ano, agregacao) |>
  dplyr::summarise(media_espvida = mean(
    ggplot(aes(x = ano, y = media_espvida,
      geom_line(linewidth = 2) +
      geom_point(size = 5) +
      scale_x_continuous(breaks = seq(2012,
      theme_bw() +
      labs(
        x = "Ano",
        y = "Expectativa de vida média (em
        color = "Agregação"
      ) +
      theme(legend.position = "bottom")

```

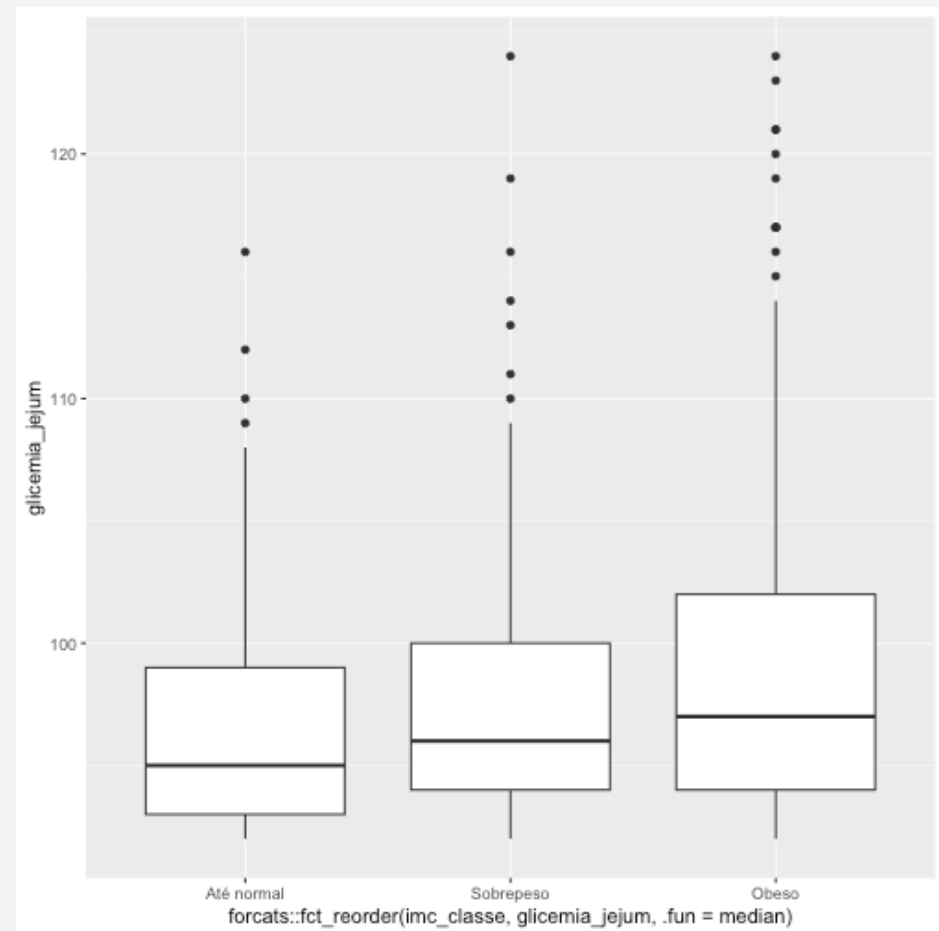


## Boxplot

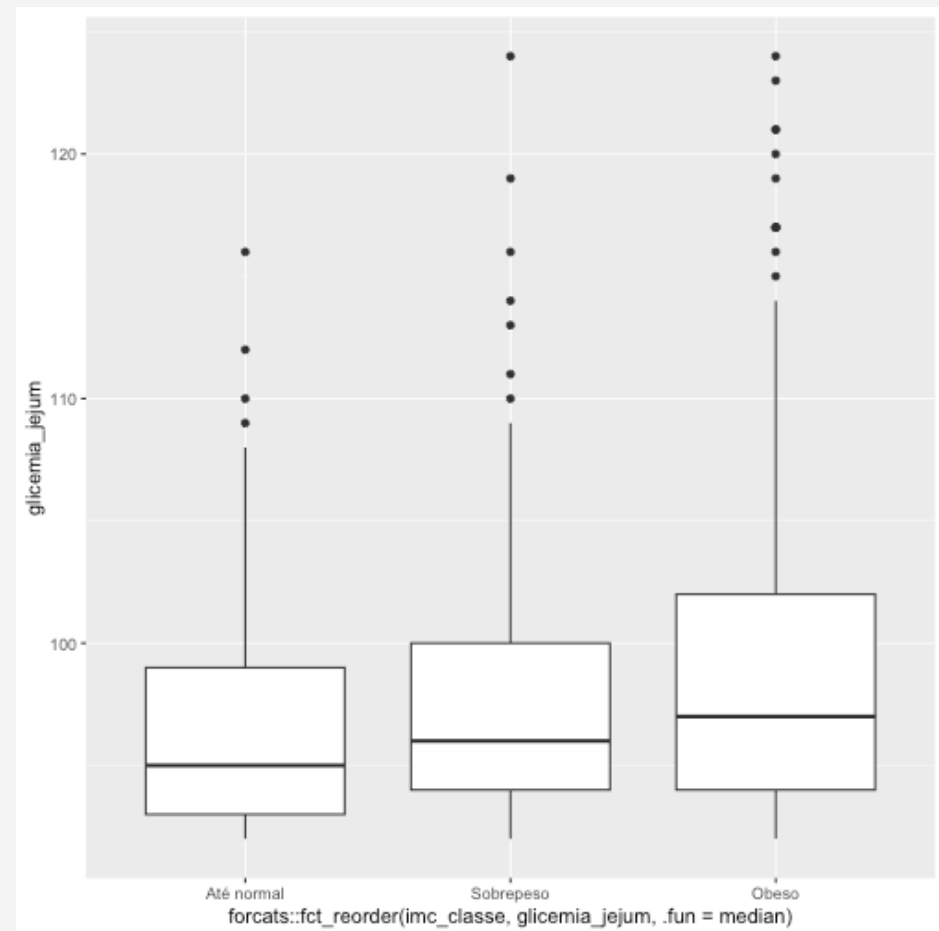
```
dados1 |>  
  ggplot(  
    aes(  
      x = forcats::fct_reorder(imc_classe,  
      y = glicemia_jejum  
    )  
  )  
)
```



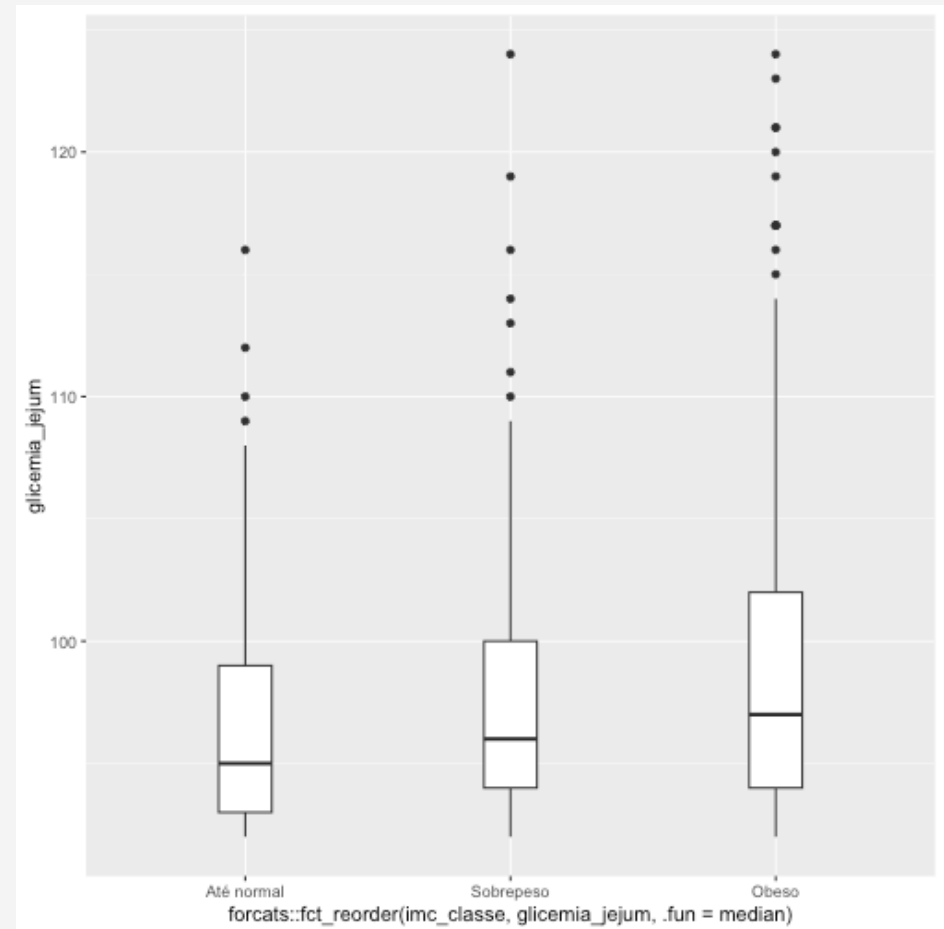
```
dados1 |>  
  ggplot(aes(x = forcats::fct_reorder(imc_classe, glicemia_jejum, .fun = median))  
  geom_boxplot())
```



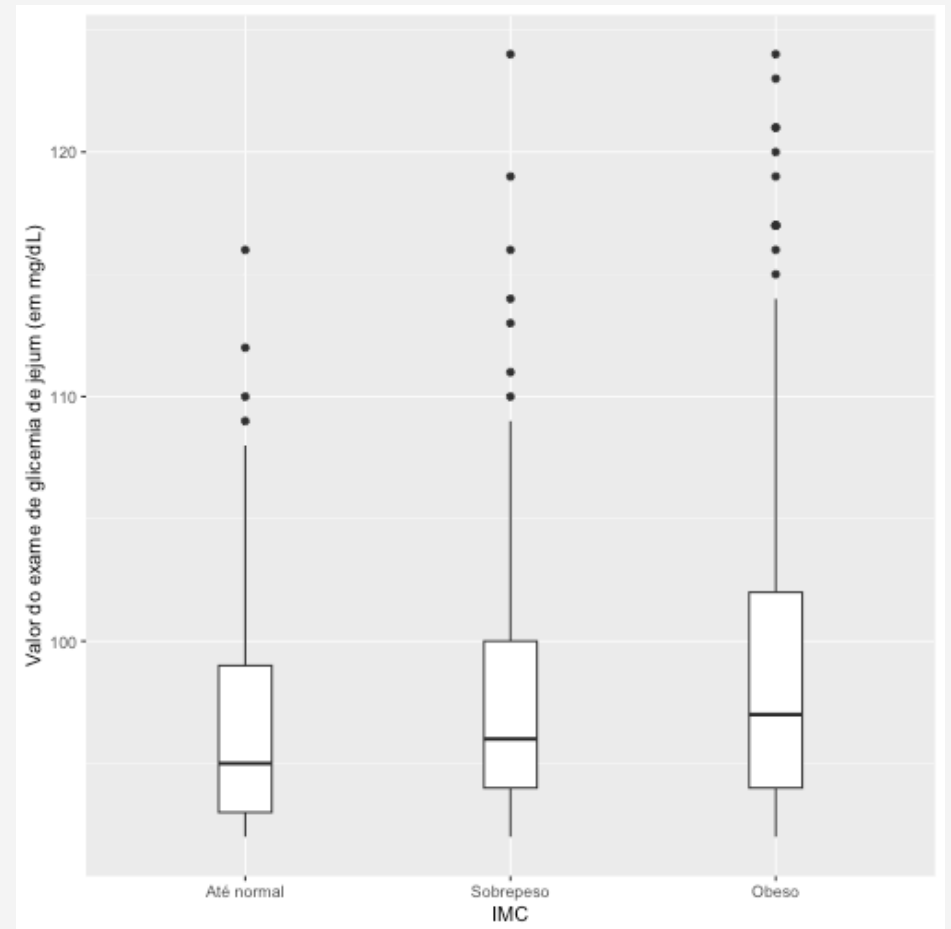
```
dados1 |>  
  ggplot(aes(x = forcats::fct_reorder(imc_classe, glicemia_jejum, .fun = median))  
  geom_boxplot())
```



```
dados1 |>  
  ggplot(aes(x = forcats::fct_reorder(imc_classe, glicemia_jejum, .fun = median)  
    geom_boxplot(width = .2)
```

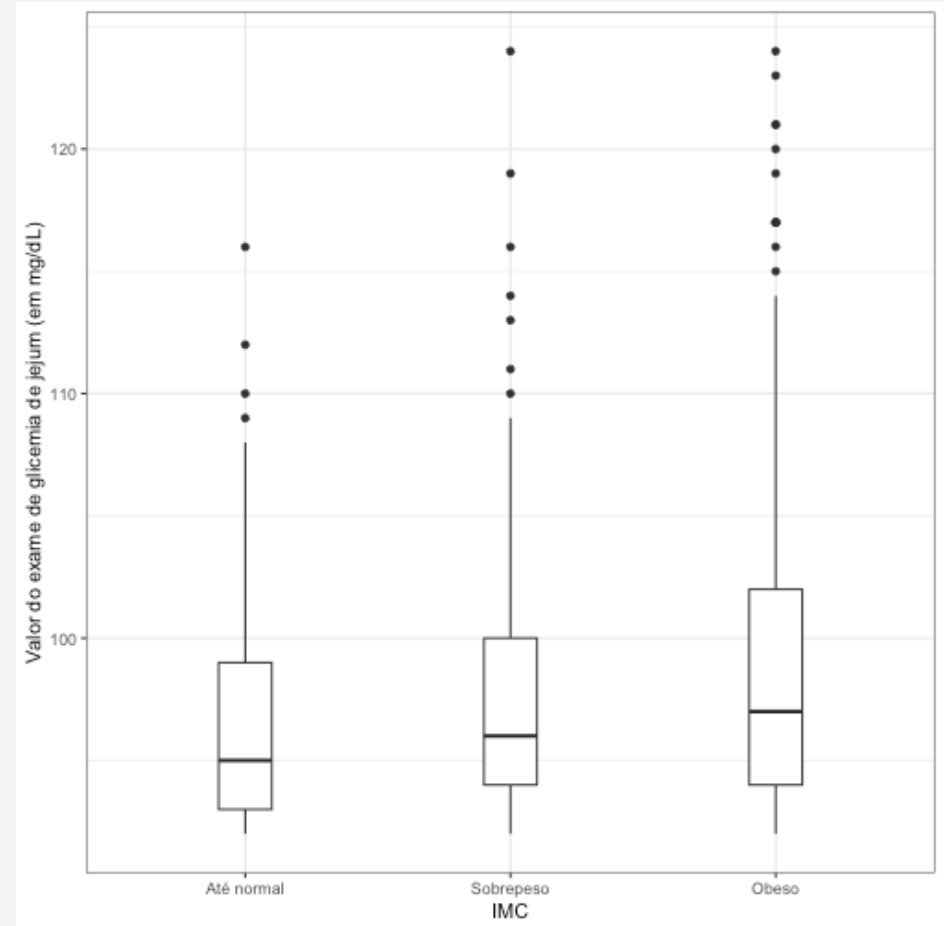


```
dados1 |>
  ggplot(aes(x = forcats::fct_reorder(imc,
    geom_boxplot(width = .2) +
    labs(
      x = "IMC",
      y = "Valor do exame de glicemia de jejum (em mg/dL)",
      fill = "Usou insulina?"
    )
  )
```



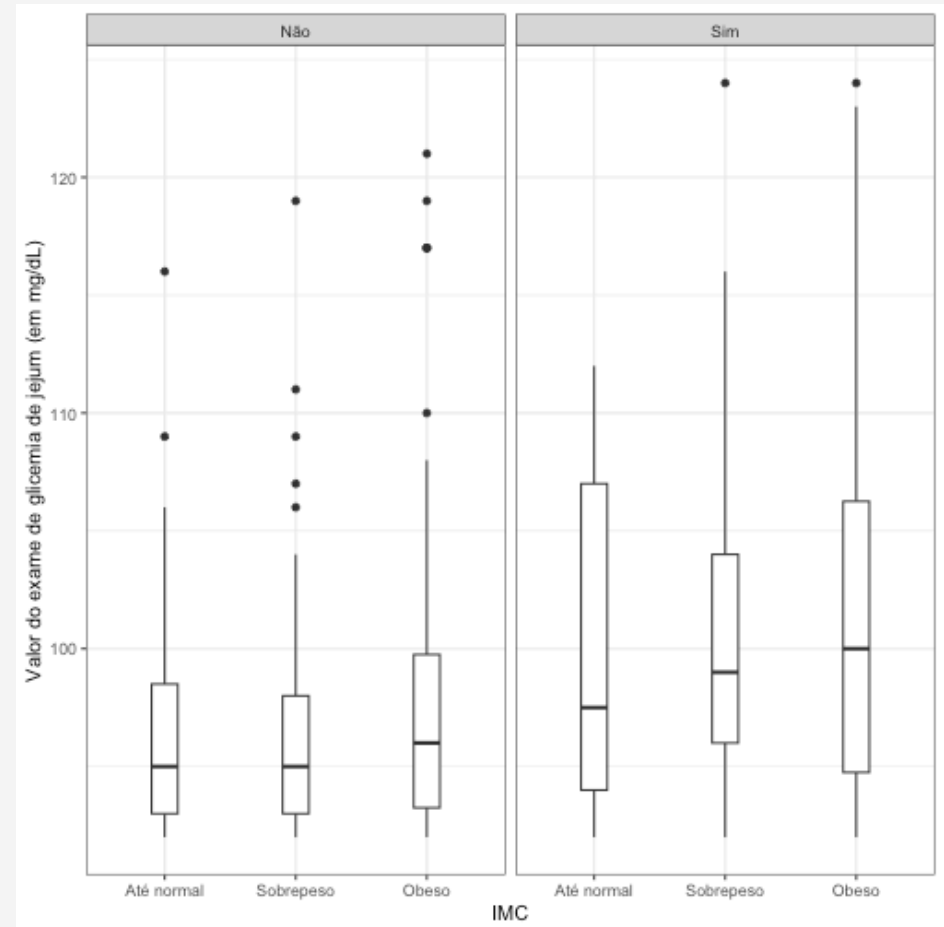
```

dados1 |>
  ggplot(aes(x = forcats::fct_reorder(imc,
    geom_boxplot(width = .2) +
    labs(
      x = "IMC",
      y = "Valor do exame de glicemia de jejum (em mg/dL)",
      fill = "Usou insulina?"
    ) +
    theme_bw()
  
```



```

dados1 |>
  ggplot(aes(x = forcats::fct_reorder(imc,
    geom_boxplot(width = .2) +
    labs(
      x = "IMC",
      y = "Valor do exame de glicemia de jejum (em mg/dL)",
      fill = "Usou insulina?"
    ) +
    theme_bw() +
    facet_wrap(. ~ insulina)
  
```





# Salvando um gráfico ggplot no

- Vamos atribuir o gráfico de barras e o histogramas a objetos:

```
g1 <- ggplot(dados1, aes(x = imc_classe, y = after_stat(count)/sum(after_stat(count))
  geom_bar(color = "blue", fill = "white") +
  scale_y_continuous(labels = scales::percent) +
  geom_text(stat = "count", aes(label = round(after_stat(count)/sum(after_stat(count)
  labs(title = "Frequência do IMC categórico das gestantes", x = "IMC categórico", y

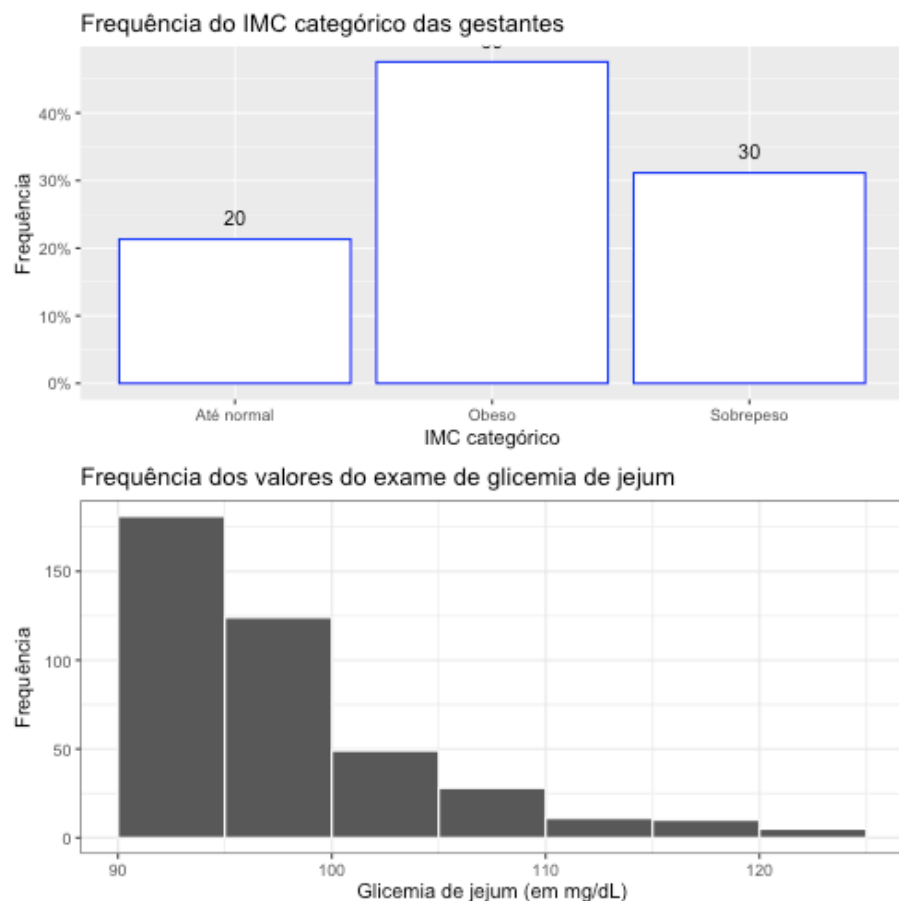
g2 <- ggplot(dados1, aes(x = glicemia_jejum)) +
  geom_histogram(color = "white", breaks = seq(90, 125, 5)) +
  labs(title = "Frequência dos valores do exame de glicemia de jejum", x = "Glicemia
  theme_bw()
```

- Podemos salvá-los em uma única imagem. Fazemos isso com o pacote {patchwork}.

```
library(patchwork)
```

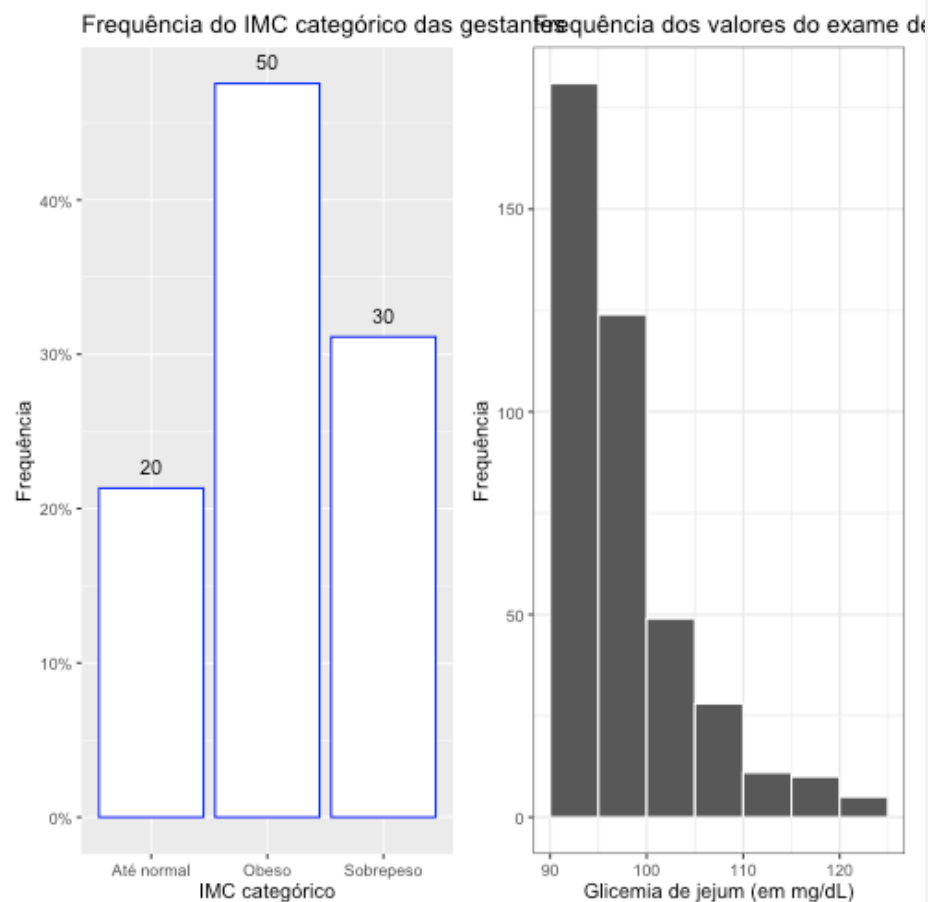
- Para o caso de dispor os gráficos um embaixo do outro, usa-se barra:

g1 / g2



- Já para o caso em que os gráficos fiquem lado a lado, usa-se sinal de adição ou barra vertical:

g1 + g2



- Agora sim, vamos salvar os gráficos! Fazemos isso com a função `ggsave()`:

```
ggsave("graficos/univariados.png", width = 16, height = 10)
```

- **Importante!** Por padrão, a função `ggsave()` salva o último gráfico que foi rodado em seu editor ou console.
- É possível salvar em vários formatos, como TEX, PDF, JPEG, TIFF, PNG e SVG.
- Por padrão, a imagem tem resolução 300dpi. Para alterá-la, use o argumento `dpi`.

# Meu obrigada!



 [ornscar@gmail.com](mailto:ornscar@gmail.com)

 [@ornscar](#)

 [@ornscar](#)