

Promoção de praticas em saude

Erick Nasareth

2025-04-27

```
if (!require("pacman")) install.packages("pacman")

## Carregando pacotes exigidos: pacman
pacman::p_load("tidyverse", "readxl")

#bolsa <- read_xlsx("Data/bolsa_familia_consolidado_geral_2023.xlsx")
populacao <- read_xls("Data/pop_cadastrada_2023.xls")
#atendimento <- read_xlsx("Data/atendimento_saude_2023.xlsx")
#palestras <- read_xlsx("Data/palestras_municipio_2023.xlsx") #temas
praticas <- read_xlsx("Data/praticas2023.xlsx")

populacao <- rename(populacao, Ibge = IBGE)
praticas$Ibge <- as.double(praticas$Ibge)

praticas$Antropometria <- gsub("\\.", "", praticas$Antropometria)
praticas$`Práticas corporais / atividade` <- gsub("\\.", "", praticas$`Práticas corporais / atividade`)
praticas$`Verificação da situação vacina` <- gsub("\\.", "", praticas$`Verificação da situação vacina`)

praticas$Antropometria <- as.numeric(praticas$Antropometria)
praticas$`Práticas corporais / atividade` <- as.numeric(praticas$`Práticas corporais / atividade`)
praticas$`Verificação da situação vacina` <- as.numeric(praticas$`Verificação da situação vacina`)

names(populacao)[7] <- "populacao_cadastrada"

populacao <- populacao %>%
  select(Ibge, populacao_cadastrada, Região)

names(populacao)[2] <- "População"

populacao <- populacao %>%
  filter(População > 0)

praticas_mun <- praticas %>%
  select(Ibge, Municipio, Antropometria, `Práticas corporais / atividade`, `Verificação da situação vaci

praticas_uf <- praticas %>%
  select(Ibge, Uf, Antropometria, `Práticas corporais / atividade`, `Verificação da situação vacina`)

praticas_regiao <- praticas %>%
  select(Ibge, Uf, Antropometria, `Práticas corporais / atividade`, `Verificação da situação vacina`)

praticas_mun <- merge(praticas_mun, populacao, by="Ibge")
praticas_uf <- merge(praticas_uf, populacao, by="Ibge")
praticas_regiao <- merge(praticas_regiao, populacao, by="Ibge")
```

Município

```
# Calculando as taxas por 1.000 habitantes
```

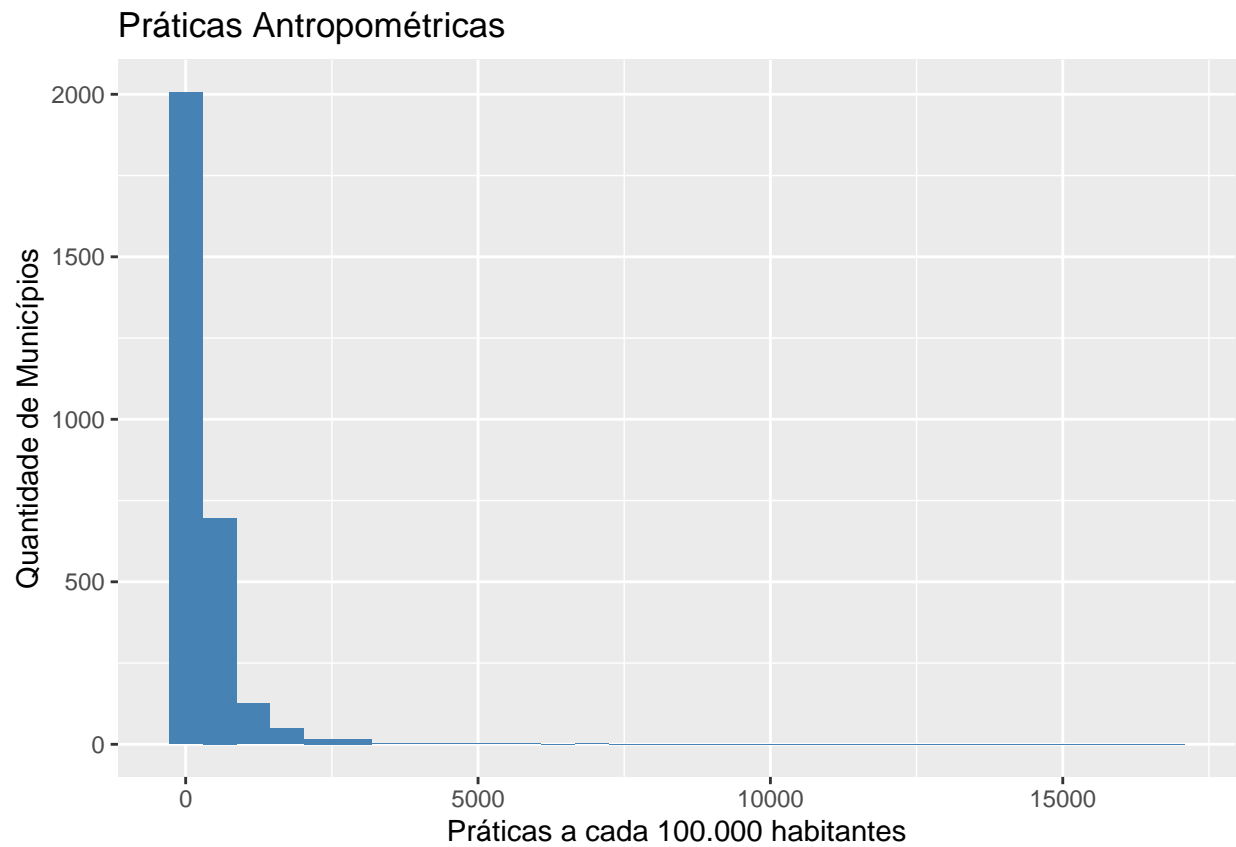
```
dados_taxas_mun <- praticas_mun %>%  
  summarise(  
    Taxa_Antropometria = (Antropometria / População) * 100000,  
    Taxa_Praticas = `Práticas corporais / atividade` / População * 100000,  
    Taxa_Vacinacao = `Verificação da situação vacina` / População * 100000  
  ) %>%  
  ungroup()
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in  
## dplyr 1.1.0.  
## i Please use `reframe()` instead.  
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`  
## always returns an ungrouped data frame and adjust accordingly.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.
```

```
dados_taxas_mun <- dados_taxas_mun %>%  
  filter(Taxa_Vacinacao != 0, Taxa_Praticas != 0, Taxa_Antropometria != 0)
```

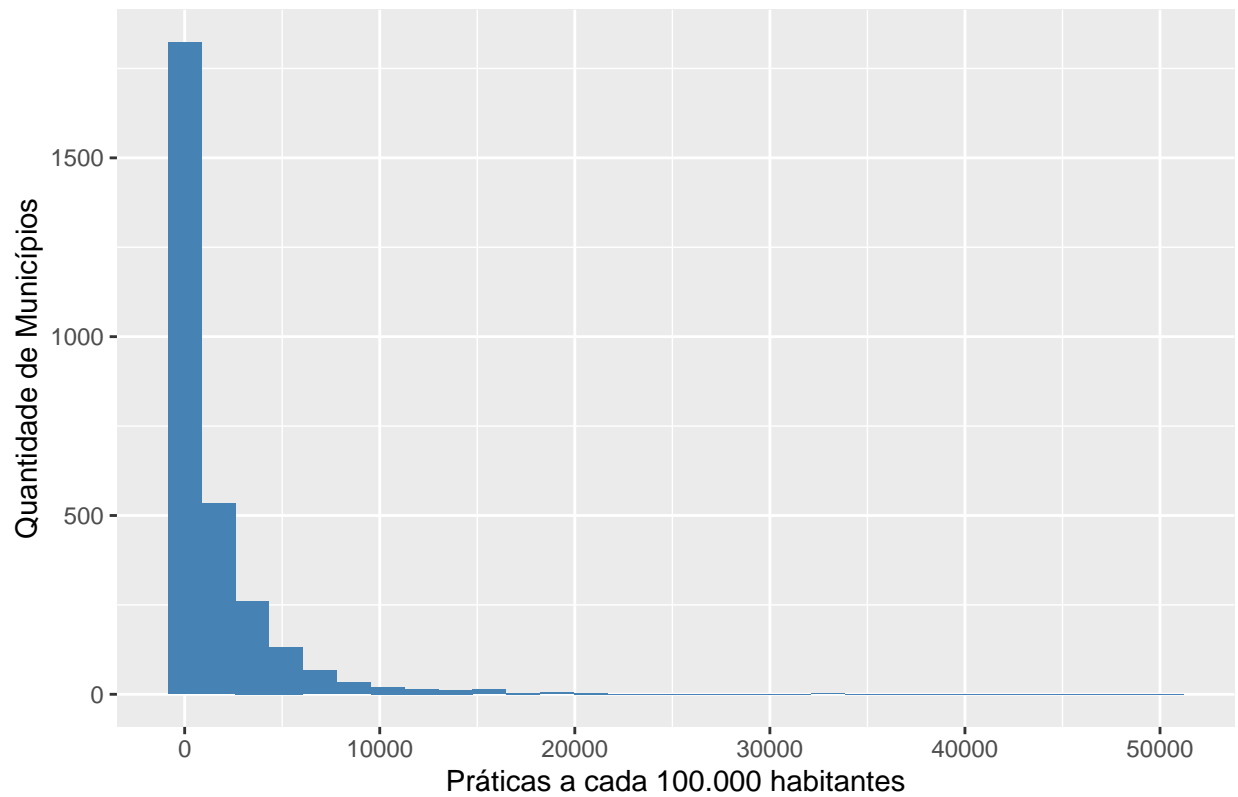
Promoção de Práticas a cada 100.000 habitantes em municípios.

```
dados_taxas_mun %>%  
  ggplot(aes(x = Taxa_Antropometria)) +  
  geom_histogram(fill = "steelblue") +  
  labs(title = "Práticas Antropométricas", x = "Práticas a cada 100.000 habitantes", y = "Quantidade de  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

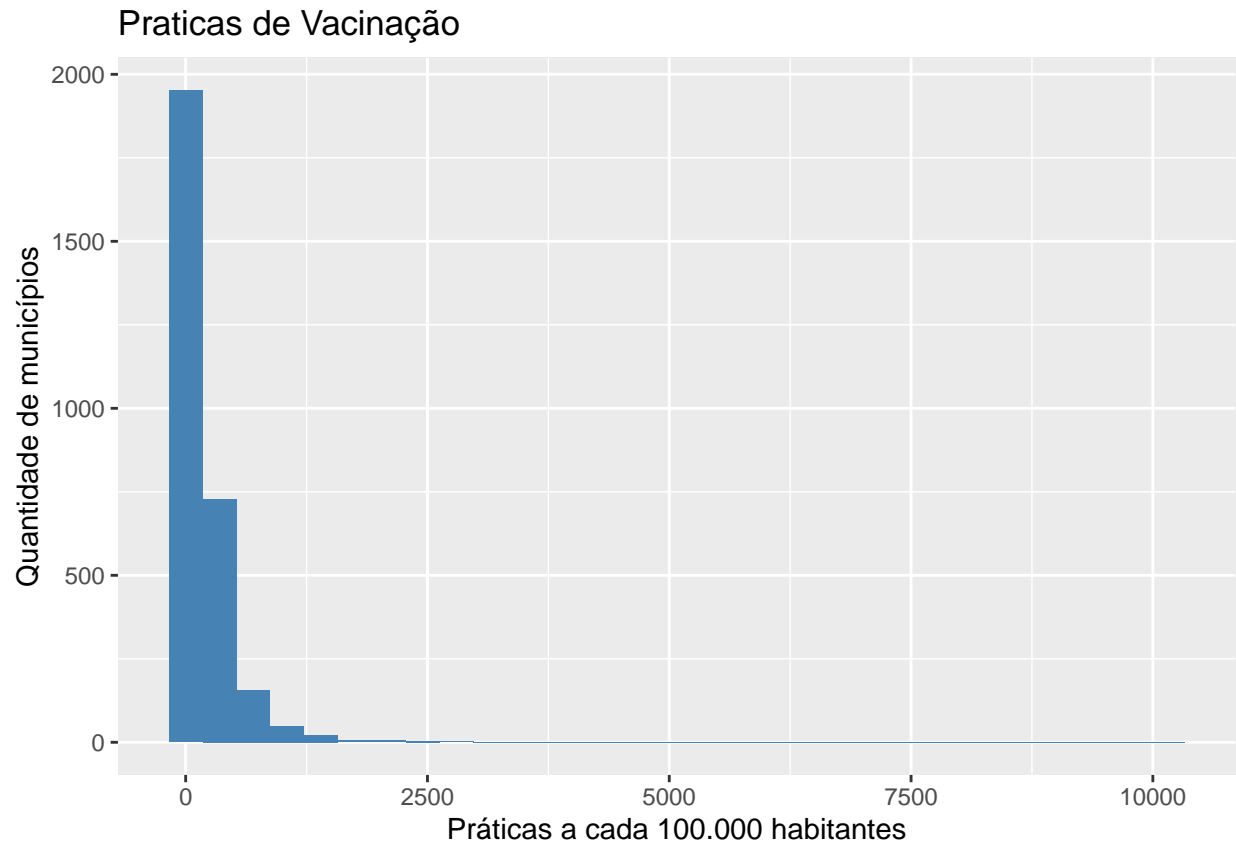


```
dados_taxas_mun %>%  
  ggplot(aes(x = Taxa_Praticas)) +  
  geom_histogram(fill = "steelblue") +  
  labs(x = "Práticas a cada 100.000 habitantes", y = "Quantidade de Municípios", title = "Práticas Corp  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Práticas Corporais



```
dados_taxas_mun %>%  
  ggplot(aes(x = Taxa_Vacinacao)) +  
  geom_histogram(fill = "steelblue") +  
  labs(x = "Práticas a cada 100.000 habitantes", y = "Quantidade de municípios", title = "Praticas de V  
  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



UF

```
# Calculando as taxas por 100.000 habitantes
dados_taxas <- praticas_uf %>%
  group_by(Uf, Região) %>%
  summarise(
    Taxa_Antropometria = sum(Antropometria) / sum(População) * 100000,
    Taxa_Praticas = sum(`Práticas corporais / atividade`) / sum(População) * 100000,
    Taxa_Vacinacao = sum(`Verificação da situação vacina`) / sum(População) * 100000
  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Uf'. You can override using the `.groups`
## argument.
```

```
media_global <- round(sum(dados_taxas$Taxa_Antropometria) / nrow(dados_taxas), 2)
```

```
ggplot(dados_taxas, aes(x = reorder(Uf, Taxa_Antropometria), y = Taxa_Antropometria, fill = Região)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f", round(Taxa_Antropometria, 2))),
    vjust = 0.4, size = 3, color = "black") +
  labs(title = "Promoção de Práticas Antropométricas por 100.000 habitantes por UF",
    x = "UF",
    y = "Taxa por 100.000 hab.") +
  geom_hline(
```

```

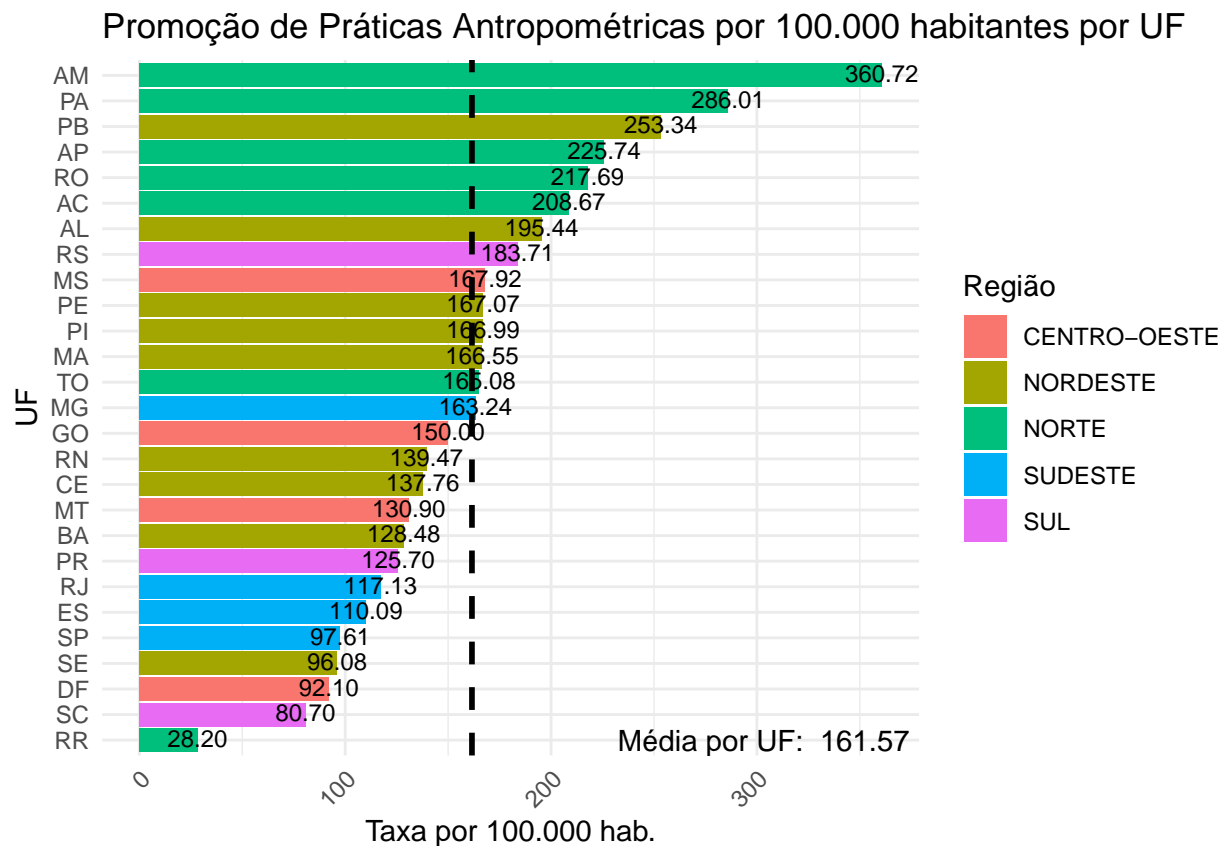
yintercept = media_global,
color       = "black",
linetype    = "dashed",
size        = 1
) +
annotate(
  "text",
  x         = 1,
  y         = media_global,
  label     = paste("Média por UF: ", media_global),
  hjust     = -0.5, vjust = 0.5,
  color     = "black"
) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
coord_flip()

```

```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```



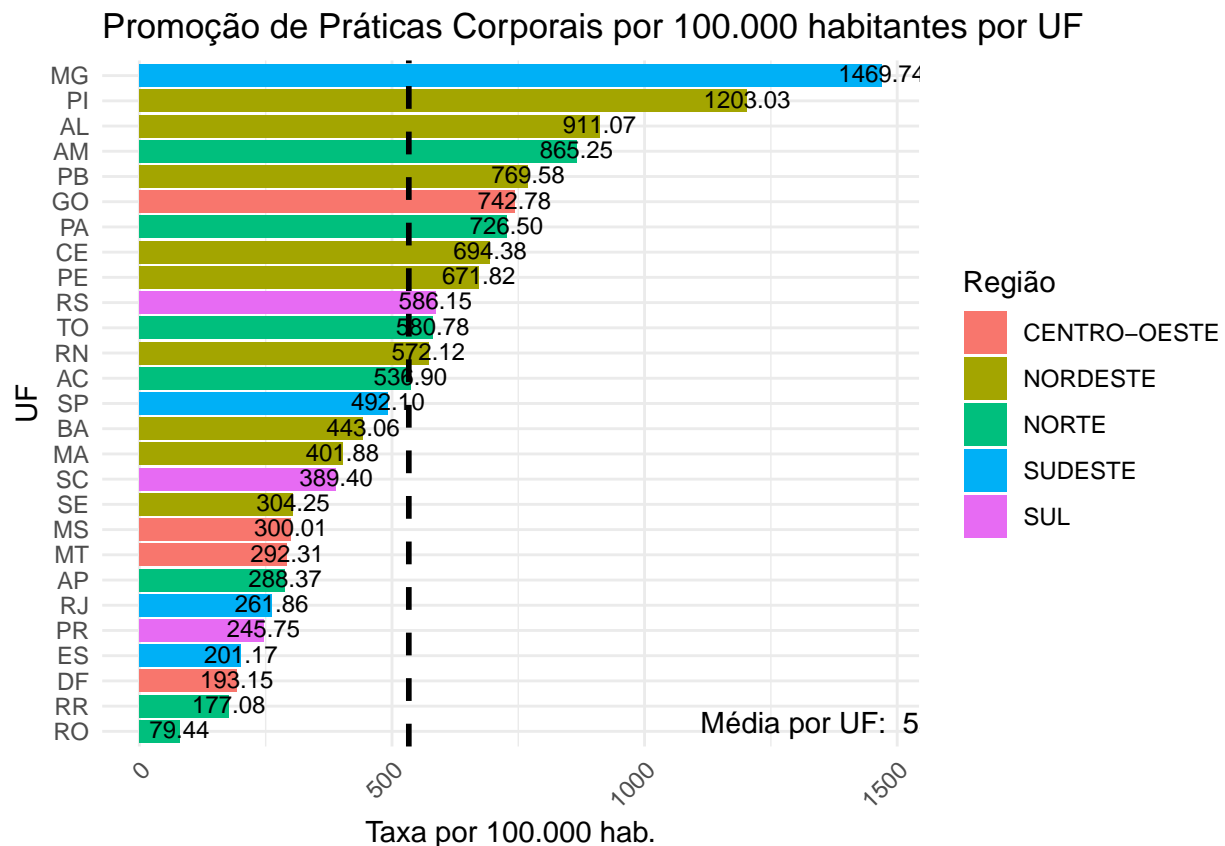
```

media_global <- round(sum(dados_taxas$Taxa_Praticas) / nrow(dados_taxas), 2)

ggplot(dados_taxas, aes(x = reorder(Uf, Taxa_Praticas), y = Taxa_Praticas, fill = Região)) +

```

```
geom_bar(stat = "identity") +
geom_text(aes(label = sprintf("%.2f", round(Taxa_Praticas, 2))),
          vjust = 0.4, size = 3, color = "black") +
labs(title = "Promoção de Práticas Corporais por 100.000 habitantes por UF",
      x = "UF",
      y = "Taxa por 100.000 hab.") +
geom_hline(
  yintercept = media_global,
  color       = "black",
  linetype    = "dashed",
  size        = 1
) +
annotate(
  "text",
  x          = 1,
  y          = media_global,
  label      = paste("Média por UF: ", media_global),
  hjust      = -1, vjust = 0.1,
  color      = "black"
) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
coord_flip()
```

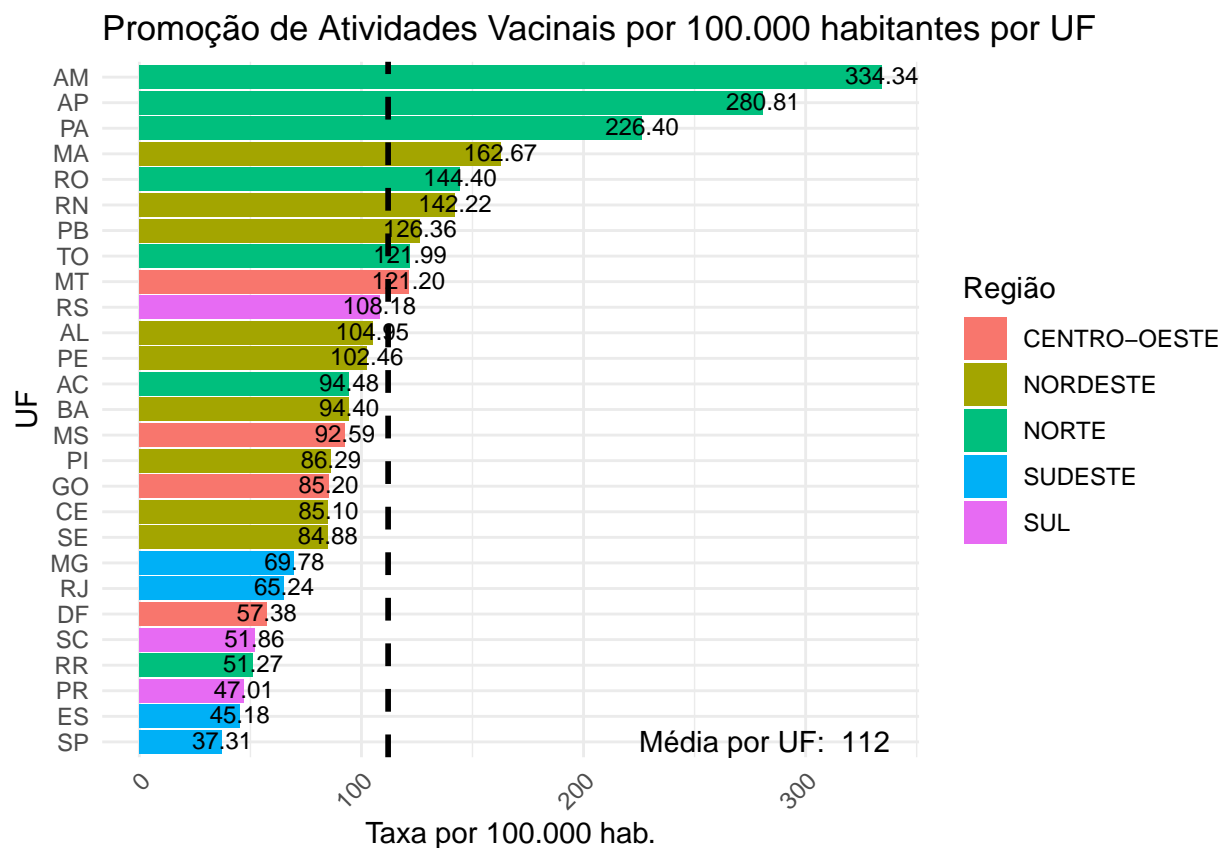


```
media_global <- round(sum(dados_taxas$Taxa_Vacinacao) / nrow(dados_taxas), 2)
```

```

ggplot(dados_taxas, aes(x = reorder(Uf, Taxa_Vacinacao), y = Taxa_Vacinacao, fill = Região)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f", round(Taxa_Vacinacao, 2))),
            vjust = 0.4, size = 3, color = "black") +
  labs(title = "Promoção de Atividades Vacinais por 100.000 habitantes por UF",
        x = "UF",
        y = "Taxa por 100.000 hab.") +
  geom_hline(
    yintercept = media_global,
    color       = "black",
    linetype    = "dashed",
    size        = 1
  ) +
  annotate(
    "text",
    x       = 1,
    y       = media_global,
    label   = paste("Média por UF: ", media_global),
    hjust   = -1, vjust = 0.5,
    color   = "black"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()

```

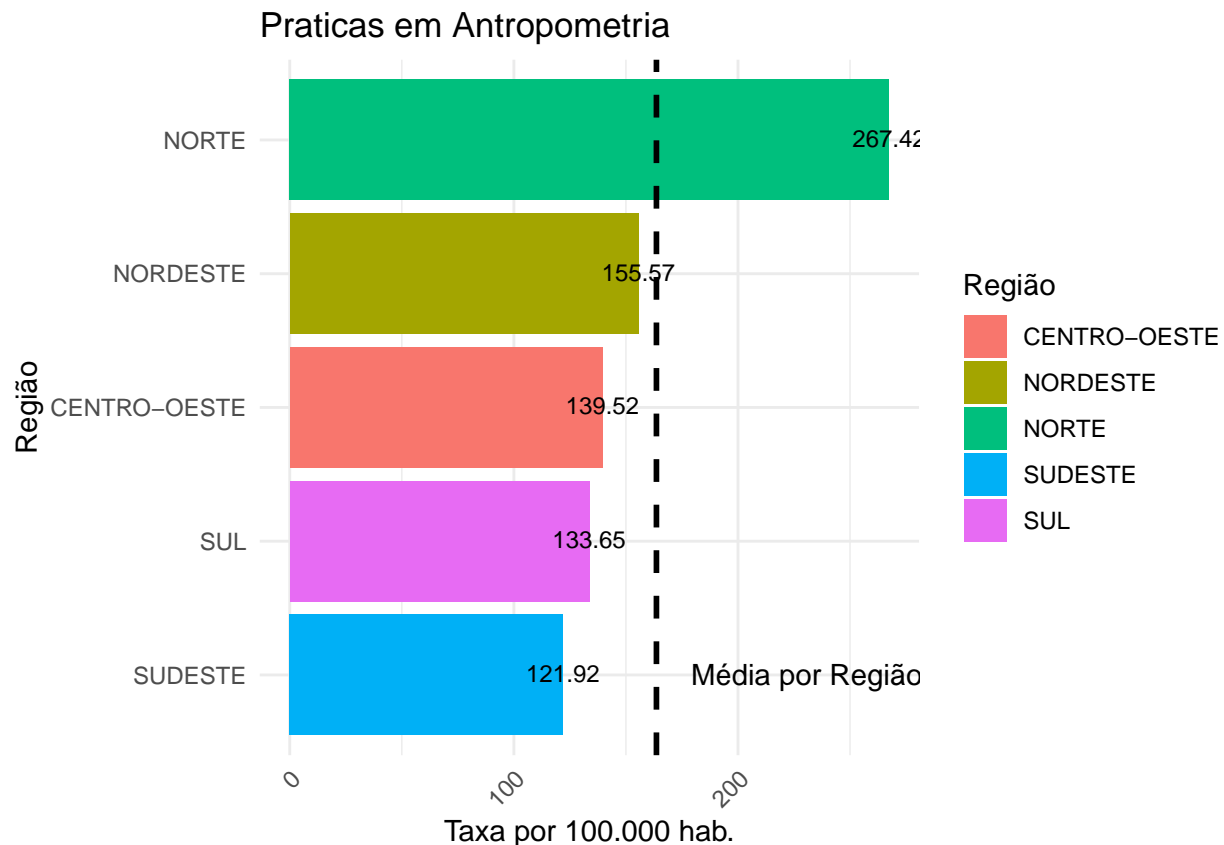


Promoção de Práticas ... a cada 100.000 habitantes por região

```
# Calculando as taxas por 100.000 habitantes
dados_taxas_regiao <- praticas_uf %>%
  group_by(Região) %>%
  summarise(
    Taxa_Antropometria = sum(Antropometria) / sum(População) * 100000,
    Taxa_Praticas = sum(`Práticas corporais / atividade`) / sum(População) * 100000,
    Taxa_Vacinacao = sum(`Verificação da situação vacina`) / sum(População) * 100000
  ) %>%
  ungroup()

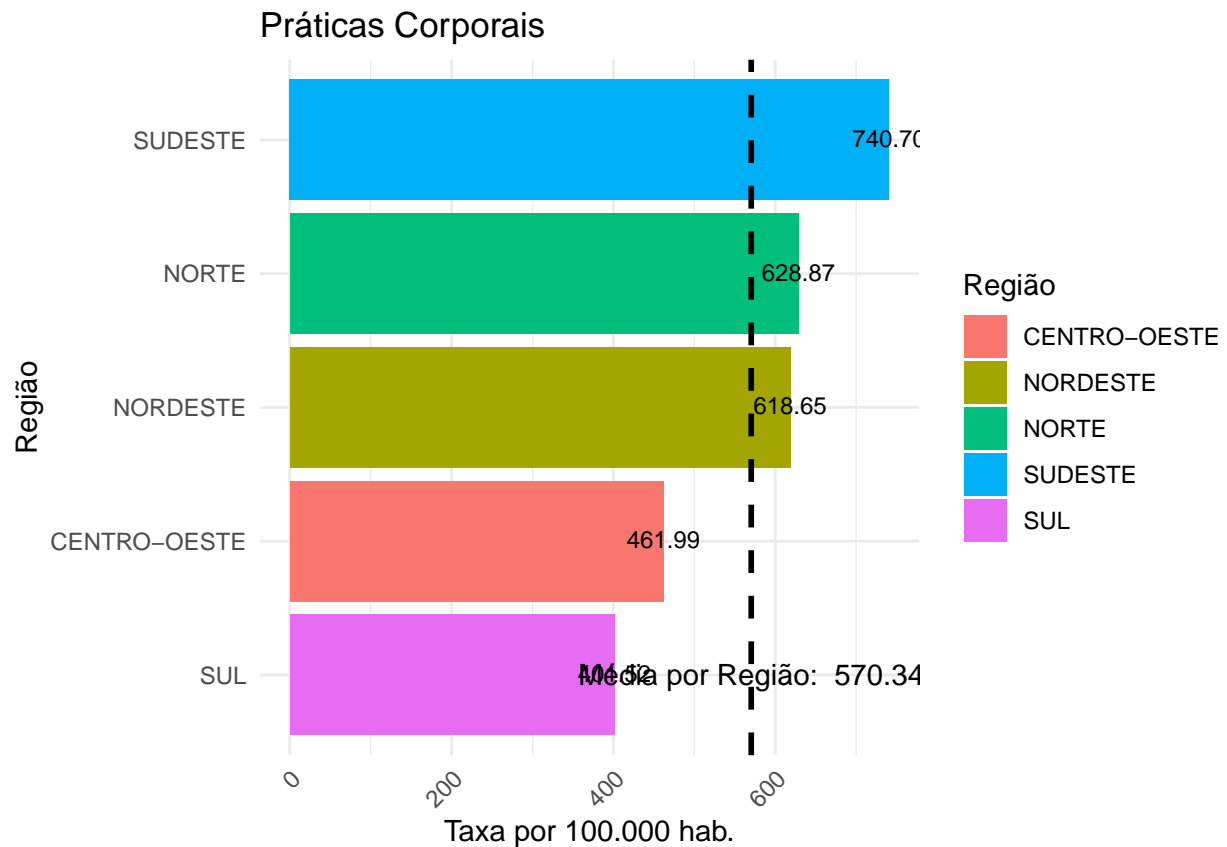
media_global <- round(sum(dados_taxas_regiao$Taxa_Antropometria) / nrow(dados_taxas_regiao), 2)

ggplot(dados_taxas_regiao, aes(x = reorder(Região, Taxa_Antropometria), y = Taxa_Antropometria, fill = Taxa_Antropometria)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f", round(Taxa_Antropometria, 2))),
    vjust = 0.4, size = 3, color = "black") +
  labs(title = "Praticas em Antropometria",
    x = "Região",
    y = "Taxa por 100.000 hab.") +
  geom_hline(
    yintercept = media_global,
    color = "black",
    linetype = "dashed",
    size = 1
  ) +
  annotate(
    "text",
    x = 1,
    y = media_global,
    label = paste("Média por Região: ", media_global),
    hjust = -0.1, vjust = 0.5,
    color = "black"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```



```
media_global <- round(sum(dados_taxas_regiao$Taxa_Praticas) / nrow(dados_taxas_regiao), 2)

ggplot(dados_taxas_regiao, aes(x = reorder(Região, Taxa_Praticas), y = Taxa_Praticas, fill = Região)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f", round(Taxa_Praticas, 2))),
            vjust = 0.4, size = 3, color = "black") +
  labs(title = "Práticas Corporais",
        x = "Região",
        y = "Taxa por 100.000 hab.") +
  geom_hline(
    yintercept = media_global,
    color       = "black",
    linetype    = "dashed",
    size        = 1
  ) +
  annotate(
    "text",
    x       = 1,
    y       = media_global,
    label   = paste("Média por Região: ", media_global),
    hjust   = 0.5, vjust = 0.5,
    color   = "black"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```



```
media_global <- round(sum(dados_taxas_regiao$Taxa_Vacinacao) / nrow(dados_taxas_regiao), 2)

ggplot(dados_taxas_regiao, aes(x = reorder(Região, Taxa_Vacinacao), y = Taxa_Vacinacao, fill = Região))
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f", round(Taxa_Vacinacao, 2))),
            vjust = 0.4, size = 3, color = "black") +
  labs(title = "Atividades vacinais",
        x = "Região",
        y = "Taxa por 100.000 hab.") +
  geom_hline(
    yintercept = media_global,
    color       = "black",
    linetype    = "dashed",
    size        = 1
  ) +
  annotate(
    "text",
    x       = 1,
    y       = media_global,
    label   = paste("Média por Região: ", media_global),
    hjust   = -0.5, vjust = 0.5,
    color   = "black"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_flip()
```

