

Metodologias Informacionais com R

Modulo V: Análise espacial aplicada de dados

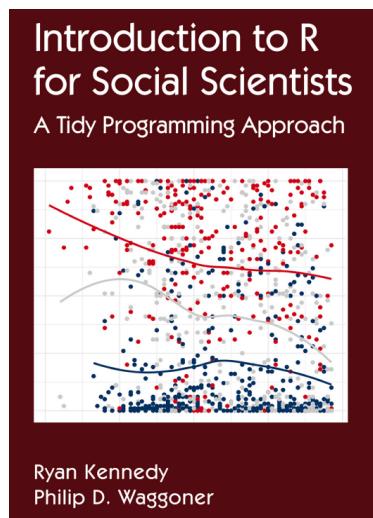
Telmo dos Santos Klipp (telmo.klipp@inpe.br)

Informações Gerais sobre o Curso

- Materiais disponibilizados via **Classroom**;
- O aprendizado requer a prática que será constante nas aulas;

Bibliografia Básica:

- Kennedy, R., & Waggoner, P. D. (2021). Introduction to r for social scientists: a tidy programming approach. CRC Press.



Bibliografia Complementar:

- Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). R for data science (2e): import, tidy, transform, visualize, and model data. "O'Reilly Media, Inc.". Disponível em: <https://r4ds.hadley.nz/>. Acesso em: 14 de junho, 2023. (Online)
- Damiani, A. et. al., (2022). Ciência de Dados em R. Curso-R. Disponível em: <https://livro.curso-r.com>. Acesso em: 12 de maio, 2023. (Online)
- de Aquino, J. A. (2014). R para cientistas sociais. Editora da UESC (editus). Disponível em: <http://www.uesc.br/editora/>. Acesso em: 12 de maio, 2023.
- de Oliveira, P. F., Guerra, S., McDonnell, R. (2018). Ciência de Dados com R: Introdução. Editora IBPAD. Disponível em: <https://cdr.ibpad.com.br/index.html>. Acesso em: 12 de maio, 2023. (Online)

Nas últimas aulas vimos:

- Aquisição de dados da PNADC por meio do pacote **PNADcIBGE** e construção do plano amostral.
- Geração de estatísticas com **survey/srvyr**.
- Manipulação de dados da PNADC usando **dplyr** e geração de estatísticas considerando ponderação por pesos dos domicílios e das pessoas.
- Manipulação de dados da PNADC usando **dplyr** em combinação com **survey/srvyr**, ou seja, considerando o plano amostral.
- Geração de gráficos com **survey** e **ggplot2**.

Agora trataremos da visualização espacial de dados

📊 Visualização de 📈 dados – Relembrando

A visualização de gráficos pode ser crucial para a fase de análise exploratória de dados – obter informações, entender os dados, relações entre variáveis, desenvolver ou validar hipóteses. Não obstante, também é importante em etapas posteriores de comunicação e divulgação – em relatórios, documentos científicos, páginas na internet, entre outros.

O R possui como pacote gráfico básico o **graphics** (vem pré-instalado). Outros pacotes gráficos são:

- Gráficos estáticos: **ggplot2** (gráficos de tipos diversos), **lattice** (gráficos essencialmente do tipo *trellis*);
- Gráficos interativos: **plotly**, **dygraphs**, **highcharter**, **ggvis**, **ganimate** (gera animações);
- Mapas estáticos: **graphics**, **ggplot2**, **sp**, **sf**, **tmap**, **ggmap**;
- Mapas interativos: **ganimate**, **leaflet**, **mapview**, **tmap**, **plotly**;
- Gráficos 3D: **graphics**, **plotly**, **plot3d**, **scatterplot3d**, **lattice**, **RGL**, **ggrgl**, **rayshader**, **rayrender** (cenas 3D).

A visualização geoespacial de dados agrega interpretabilidade às informações, permitindo a combinação de dados e metadados com subdivisões geográficas. Para isso, trabalharemos principalmente com **ggplot2**.

📊 Visualização de 📈 dados - **ggplot2** – Relembrando

Os cinco componentes de um gráfico segundo a interpretação de "Layered Grammar" no **ggplot2** são:

1. Camada(s) – representação visual das propriedades físicas dos dados:
 - Dados (***data***);
 - Mapeamento de objetos (***aesthetic***) – define as variáveis que compõem o gráfico; divisões categóricas dos dados que são traduzidas em cores, formatos e tamanhos; elementos da legenda e agrupamentos;
 - Geometria dos objetos (***geom***) – basicamente o tipo do gráfico;
 - Transformações estatísticas (***stat***) – basicamente summarização de dados;
 - Posicionamento (***position***) – permite o deslocamento dos elementos gráficos, se houver sobreposição.
2. Escalas (***scales***) – permite o controle sobre atributos estéticos como cores, formas e tamanhos dos elementos geométricos, espaçamento e disposição dos *labels* nos eixos e legenda, entre outros.
3. Sistema de coordenadas (***coord***) – permite definir posição/aparência dos objetos mapeados no plano.
4. Facetamento/tabulamento (***faceting***) – gera painéis contendo subdivisões dos dados baseadas em informações categóricas. Os subgráficos compartilham os mesmos atributos estéticos.
5. Tema (***theme***) – Define elementos gerais como cor do plano de fundo, tamanho de texto e tipo de fonte.

Visualização espacial de dados - ggplot2

O **ggplot2** permite a produção de mapas estáticos segundo algumas **possibilidades**:

- Representação de mapas e dados por meio de polígonos:
 - Funções comumente usadas: `geom_polygon()`, `geom_map()`, `coord_map()` e `coord_quickmap()`.
 - Requer variáveis que indiquem as coordenadas dos vértices dos polígonos.
 - Talvez o uso seja mais simples, porém, é menos flexível nas representações espaciais.
 - Possui limitações entre os tipos de projeções existentes (Observe em [link 1](#) e [link 2](#)).
- Representação de mapas e dados por meio de objetos espaciais conhecidos como "*simple features*":
 - Funções comumente usadas: `geom_sf()` e `coord_sf()`.
 - Flexibilidade em representações espaciais (ex: pontos, linhas e polígonos).
 - Padronização seguida pela *Open Geospatial Consortium*.
 - Diversidade/facilidade entre os tipos de projeções existentes (Observe em [link 1](#) e [link 2](#)).

Observação: Em uma projeção, queremos mapear o globo terrestre para uma superfície plana, o que envolve transformações de coordenadas (latitudes e longitudes) da superfície esférica para a plana. Existem diversas formas de projeções possíveis (ex: cilíndrica, cônica, plana, Robinson e Mollweide).

Visualização espacial de dados - ggplot2

A visualização espacial pode ser dividida em duas etapas:

1. Obter um conjunto de dados de um mapa base.
2. Adicionar/mapear os dados e metadados que nos interessam ao mapa base.

Existem diversos pacotes no r que fornecem mapas:

- Representação por meio de polígonos: **maps**.
- Representação por meio de objetos **sf** (17 tipos de "simple features"): **rnatgearth**, **geobr**.
- Representação por meio de objetos **sp** (pontos, grades, linhas, anéis e polígonos): **rworldmap**, **rworldxtra**, **spData**.

Existem diversos pacotes no r que permitem manipulação e análise de objetos e dados espaciais (ex: **sf**, **sp**, **raster**, **terra**). Não obstante, começamos por carregar o **ggplot2** e outros pacotes do **tidyverse**.

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

Visualização espacial de dados - ggplot2

Vejamos exemplos de mapas construídos com polígonos. Carregamos um mapa base do pacote **maps**, usando a função `ggplot2::map_data()`:

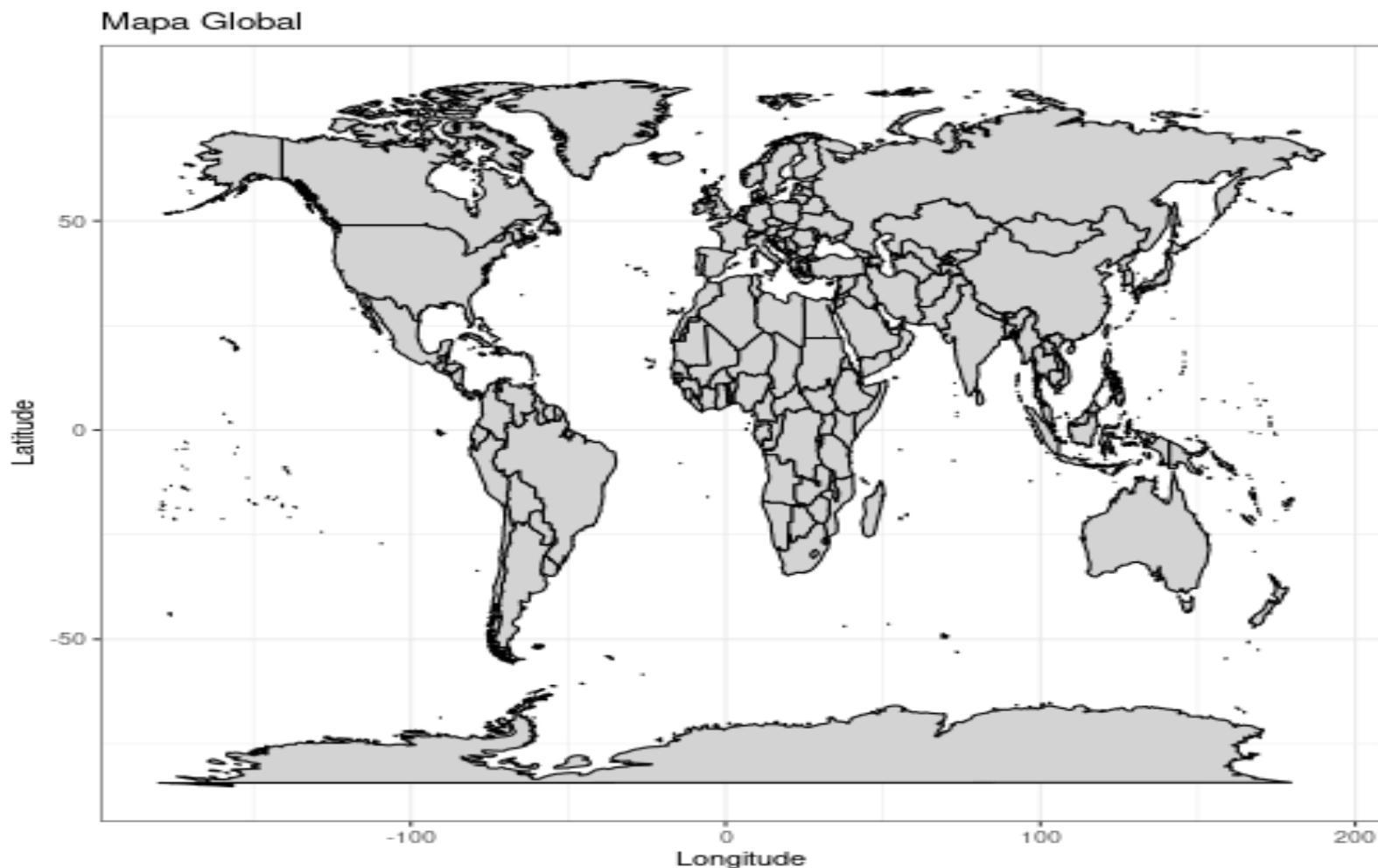
```
# install.packages("maps")
library(maps)
world_map <- map_data("world")
```

Podemos gerar esse mapa simples, conforme o código abaixo. A variável **group** possui os códigos individuais dos países, que indicamos no atributo estético, de mesmo nome, para unir os vértices dos polígonos (coordenadas das colunas **long** e **lat**), ou seja, para gerar os contornos territoriais de cada país. Fixamos o preenchimento dos polígonos na cor cinza e as fronteiras como linhas pretas.

```
ggplot(data = world_map, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill = "lightgray", color = "black") +
  labs(title = "Mapa Global", x = "Longitude", y = "Latitude") +
  theme_bw()
```



Visualização espacial de dados - ggplot2



Visualização espacial de dados - ggplot2

Vejamos como filtrar e produzir um mapa dos países da América do Sul.

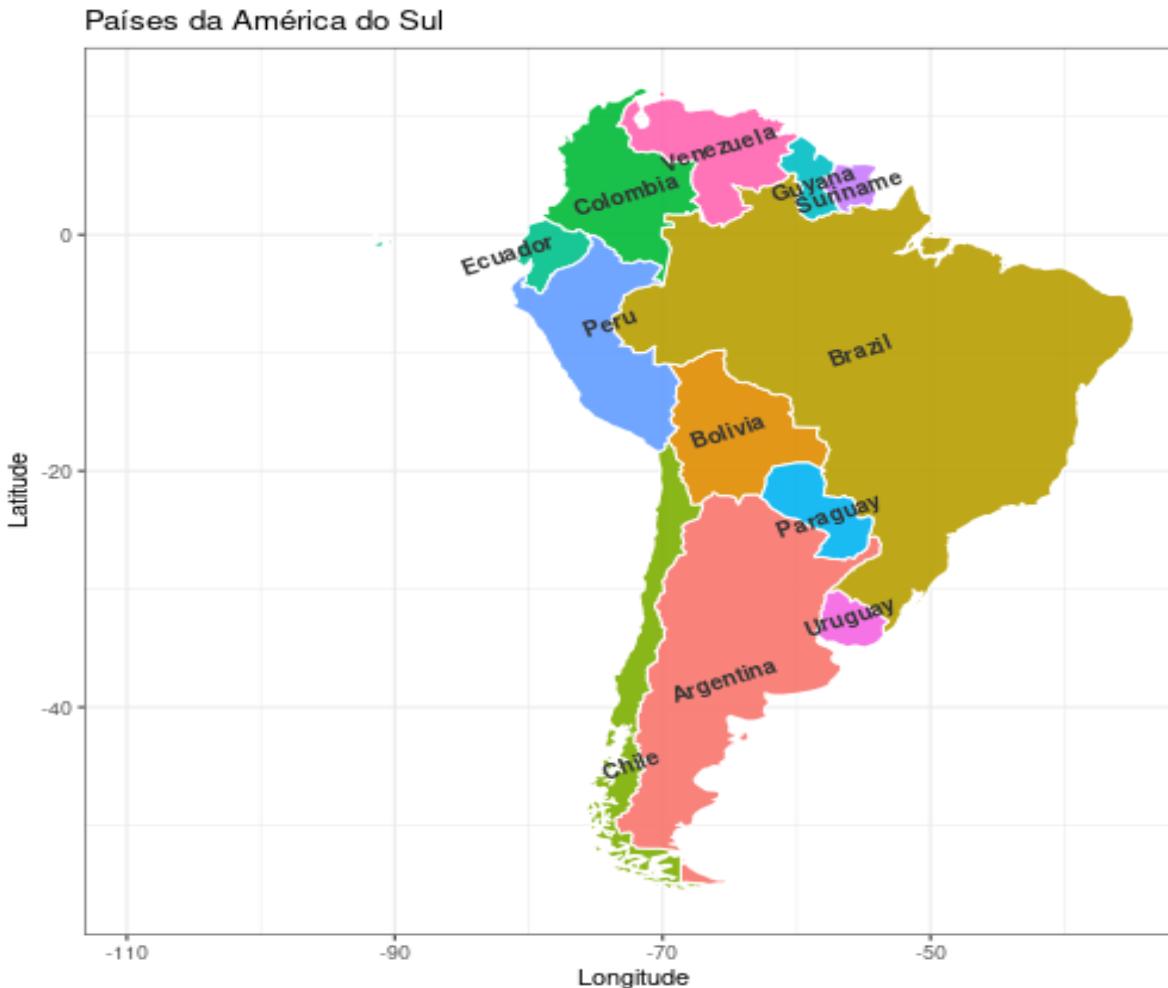
```
south_america <- c("Brazil", "Argentina", "Chile", "Uruguay", "Paraguay", "Ecuador",
                    "Peru", "Venezuela", "Colombia", "Bolivia", "Suriname", "Guyana")
south_america_countries <- filter(world_map, region %in% south_america)
south_america_labels <- summarise(south_america_countries,
                                    long = mean(long), lat = mean(lat), .by = region)
```

Agora, o preenchimento dos polígonos de cada país ocorrerá com as cores básicas do **ggplot2**, ao indicar a variável **region**. Os contornos serão em linhas brancas. Com **geom_text()**, posicionaremos os nomes dos países (coluna **region**) como *labels*, cujas coordenadas foram calculadas como a média das longitudes e latitudes (vértices) dos polígonos.

```
ggplot(data = south_america_countries, aes(x = long, y = lat)) +
  geom_polygon(aes(group = group, fill = region), color = "white", alpha = .9) +
  geom_text(data = south_america_labels, aes(label = region),
            size = 3.5, fontface = 2, color = "gray20", angle = 20, check_overlap = TRUE) +
  labs(title = "Países da América do Sul", x = "Longitude", y = "Latitude") +
  theme_bw() + theme(legend.position = "none")
```



Visualização espacial de dados - ggplot2



Visualização espacial de dados - ggplot2

Como já temos o mapa base, podemos adicionar/mapear dados ao mesmo. Como exemplo, vamos adquirir dados de expectativa de vida da *World Health Organization* usando o pacote **WHO**. Este pacote não está disponível no **CRAN**, mas podemos instalá-lo direto do seu repositório no GitHub através de **devtools**.

```
# From Github
library(devtools)
install_github("expresso/WHO")
```

Agora, obtemos os dados de expectativa de vida do ano de 2019, fazendo algumas alterações, conforme:

```
library(WHO)
expectativa_de_vida <- get_data("WHOSIS_000001") |>
  filter(year == 2019) |>
  rename(region_at = region, region = country, life_exp = value) |>
  mutate(region = ifelse(region == "United States of America", "USA", region))
```

A coluna **region** contém os nomes dos países e estará presente tanto no conjunto de dados do mapa quanto no de expectativa de vida. Assim, podemos estabelecer uma relação entre as tabelas.

Visualização espacial de dados - ggplot2

Podemos gerar um mapa com `geom_map()`, mapeando os dados de expectativa de vida, na forma:

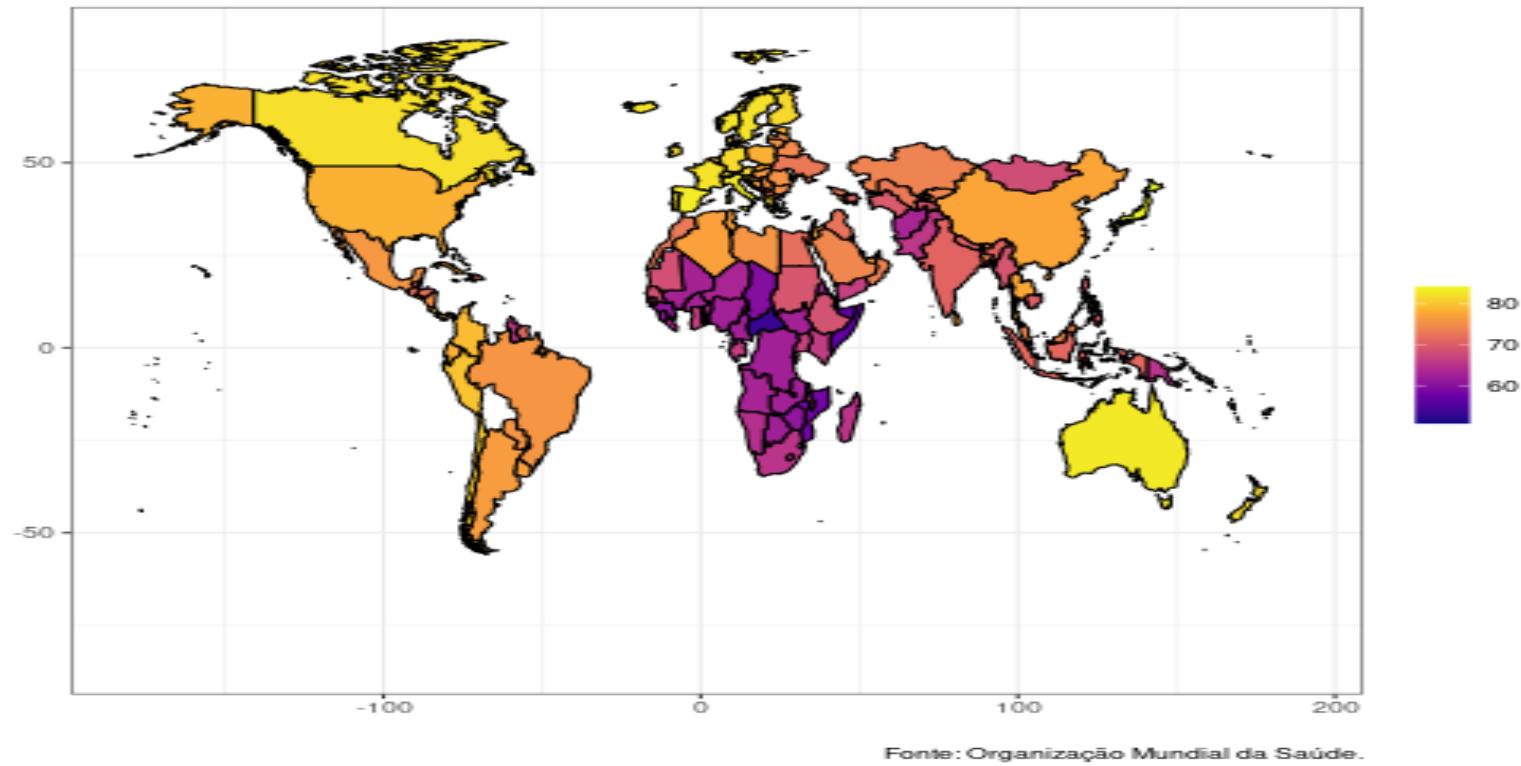
```
expectativa_de_vida |>
  filter(sex == "Both sexes") |>
  ggplot() +
  geom_map(aes(fill = life_exp, map_id = region), map = world_map, color = "black") +
  expand_limits(x = world_map$long, y = world_map$lat) +
  scale_fill_viridis_c(option = "C") +
  labs(title = "Expectativa de vida em diversos países do mundo em 2019.",
       x = "", y = "", caption = "Fonte: Organização Mundial da Saúde.") +
  theme_bw() + theme(legend.title = element_blank())
```

A função `expand_limits()` é necessaria para indicar as coordenadas nos eixos do mapa a ser gerado, já que as latitudes e longitudes não estão presentes nos dados de expectativa de vida.



Visualização espacial de dados - ggplot2

Expectativa de vida em diversos países do mundo em 2019.



Visualização espacial de dados - ggplot2

Podemos unir os dados com o mapa base e gerar a visualização com `geom_polygon()` ou `geom_map()`:

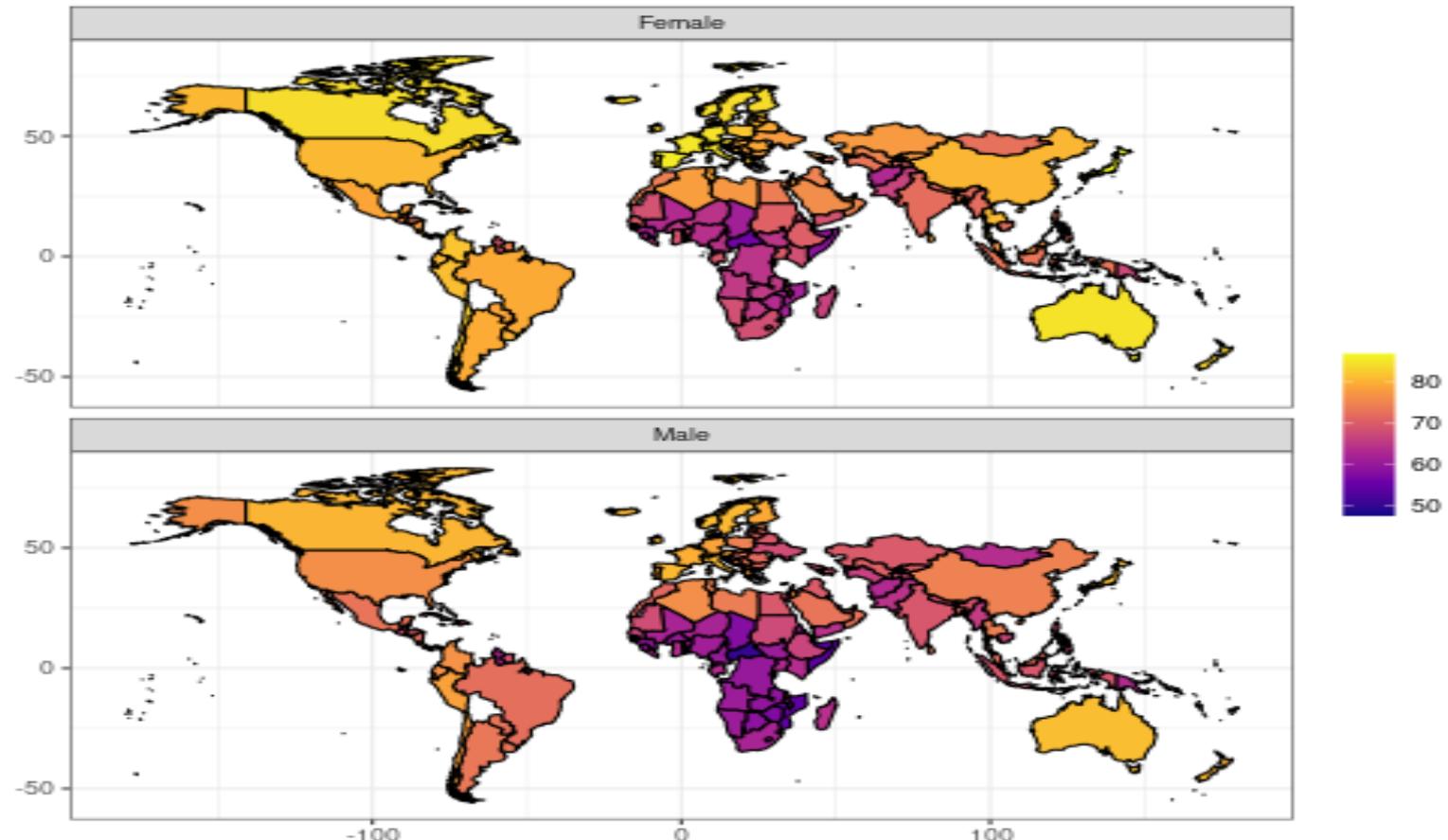
```
my_map <- left_join(expectativa_de_vida, world_map, by = "region") # unindo os conjuntos
my_title <- "Expectativa de vida (homens e mulheres) em diversos países, no ano de 2019."
my_cap <- "Fonte: Organização Mundial da Saúde."
my_map |>
  filter(sex != "Both sexes") |>
  ggplot(aes(long, lat, group = group)) +
  geom_polygon(aes(fill = life_exp ), color = "black") + facet_wrap(~sex, nrow = 2) +
  scale_fill_viridis_c(option = "C") +
  labs(title = my_title, x = "", y = "", caption = my_cap) +
  theme_bw() +
  theme(legend.title = element_blank()) # ou usando geom_map()
```

```
my_map |> filter(sex != "Both sexes") |>
  ggplot(aes(long, lat, fill = life_exp, map_id = region)) +
  geom_map(map = my_map, color = "black") + facet_wrap(~sex, nrow = 2) +
  scale_fill_viridis_c(option = "C") +
  labs(title = my_title, x = "", y = "", caption = my_cap) +
  theme_bw() + theme(legend.title = element_blank())
```



Visualização espacial de dados - ggplot2

Expectativa de vida (homens e mulheres) em diversos países, no ano de 2019.



Fonte: Organização Mundial da Saúde.

Visualização espacial de dados - ggplot2

Agora, veremos exemplos de mapas baseados em "*simple features*". O pacote **sf**, em específico, possui uma diversidade de representações espaciais, operações geométricas, transformações de projeção e suporte para outras bibliotecas geoespaciais. Para os exemplos, obteremos um mapa através dos pacotes **rnatu**re**earth** e **rnatu**re**earth****d**a**t**a:

```
install.packages(c("rnaturalearth", "rnaturalearthdata"))
# Pacotes auxiliares para gráficos do ggplot2
install.packages(c("ggrepel", "ggspatial"))
```

```
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
library(ggrepel) # permite evitar a sobreposição de textos usados como labels
library(ggspatial) # fornece anotações (ex: escala) e outras operações em mapas
```

Carregamos o mapa base usando a função **rnaturalearth::ne_countries()**:

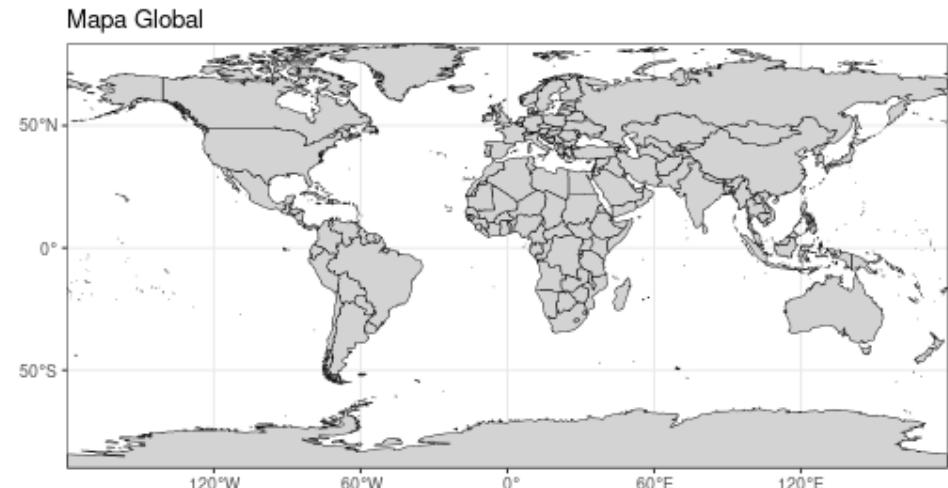
```
world_map <- ne_countries(scale = "medium", returnclass = "sf")
```

Visualização espacial de dados - ggplot2

Podemos gerar o mapa, na forma:

```
ggplot(data = world_map) +  
  geom_sf(fill = "lightgray",  
          color = "black") +  
  coord_sf(expand = FALSE) +  
  labs(title = "Mapa Global") +  
  theme_bw()
```

A função `coord_sf(expand = FALSE)` é necessária – nesse caso específico – para não expandir os limites do mapa, tornando possível a visualização dos *labels* das coordenadas.



Visualização espacial de dados - ggplot2

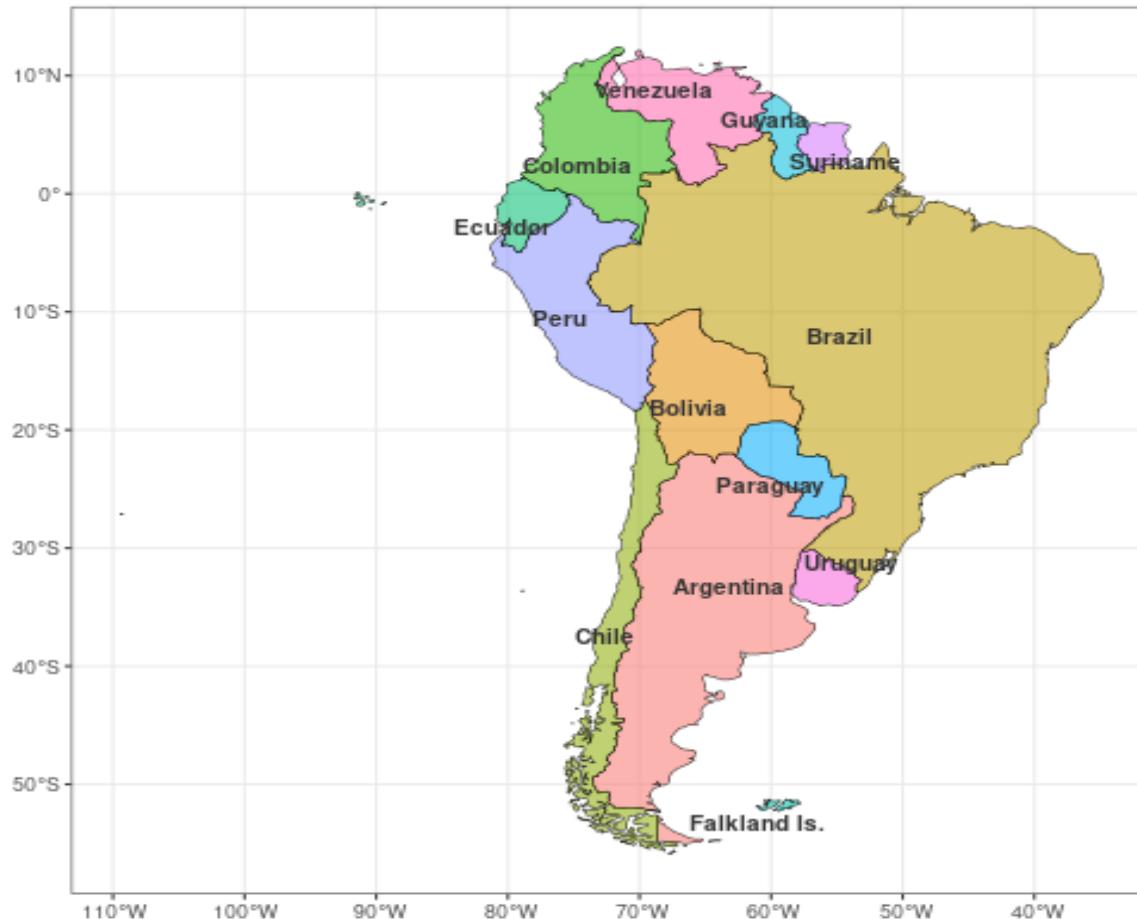
Para produzir um mapa da América do Sul, filtramos os dados usando a variável `continent`. Como o pacote `sf` possui operações sobre objetos geométricos, podemos calcular os centros dos multi-polígonos que representam os países (coluna `geometry`) e converter para coordenadas (latitudes e longitudes). Estas serão usadas para posicionar os nomes dos países como *labels* no mapa com a função `geom_text_repel()`, que corrigirá, caso ocorra, sobreposições entre os textos.

```
world_map |>
  filter(continent == "South America") |>
  mutate(lon = sf::st_coordinates(sf::st_centroid(geometry))[,1],
         lat = sf::st_coordinates(sf::st_centroid(geometry))[,2]) |>
  ggplot() +
  geom_sf(aes(fill = name, alpha = .9), color = "black") +
  geom_text_repel(aes(lon, lat, label = name), size = 3.5, fontface = 2, color = "gray20") +
  labs(title = "América do Sul", x = "", y = "") +
  theme_bw() + theme(legend.position = "none")
```



Visualização espacial de dados - ggplot2

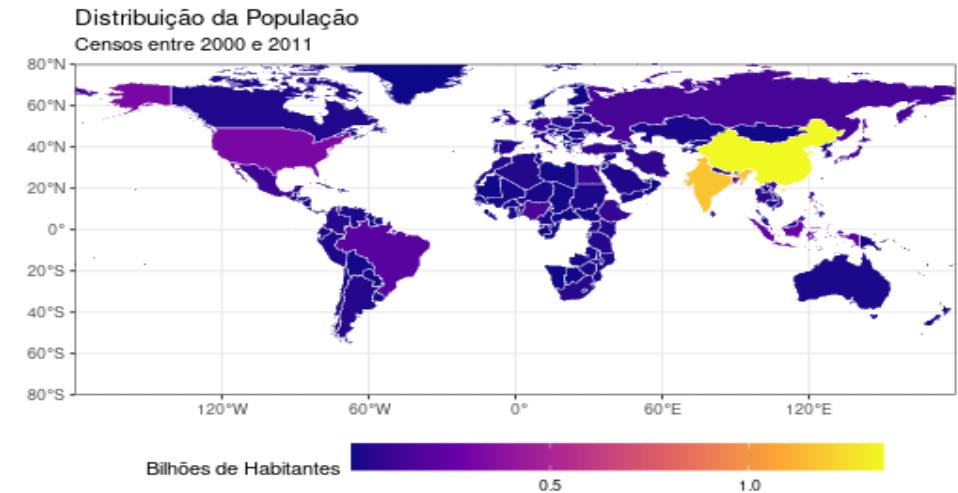
América do Sul



Visualização espacial de dados - ggplot2

O mapa base que baixamos já possui informações anexas. Observe:

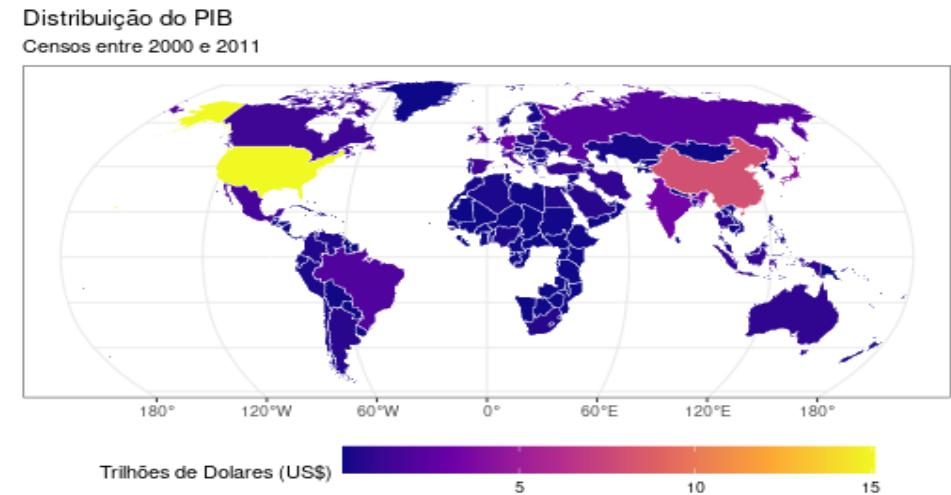
```
t1 = "Distribuição da População"
sub1 = "Censos entre 2000 e 2011"
world_map |> filter(lastcensus >= 2000) |>
  ggplot() +
    geom_sf(aes(fill = pop_est * 10^-9),
            color = "white") +
    coord_sf(expand = FALSE) +
    scale_y_continuous(limits = c(-80, 80)) +
    scale_fill_viridis_c(
      option = "C",
      name = "Bilhões de Habitantes") +
    labs(title = t1, subtitle = sub1) +
    theme_bw() +
    theme(legend.direction = "horizontal",
          legend.position = "bottom",
          legend.key.width = unit(2, "cm"))
```



Visualização espacial de dados - ggplot2

Podemos facilmente mudar a projeção com `coord_sf()`. Observe:

```
t2 = "Distribuição do PIB"
world_map |> subset(lastcensus >= 2000) |>
ggplot() +
  geom_sf(aes(fill = gdp_md_est * 10^-6),
          color = "white") +
  coord_sf(crs = "+proj=robin") +
  scale_fill_viridis_c(
    option = "C",
    name = "Trilhões de Dolares (US$)") +
  labs(title = t2, subtitle = sub1) +
  theme_bw() +
  theme(legend.direction = "horizontal",
        legend.position = "bottom",
        legend.key.width = unit(2, "cm"))
```



Visualização espacial de dados - ggplot2

Vimos algumas formas de produzir mapas. Sem embargo, o uso de objetos do tipo "*simple features*" junto ao **ggplot2** tem se tornado predominante pela flexibilidade e disponibilidade de operações sobre mapas e dados. Seguiremos essa linha. Também precisamos de mapas do Brasil, portanto, usaremos o pacote **geobr**:

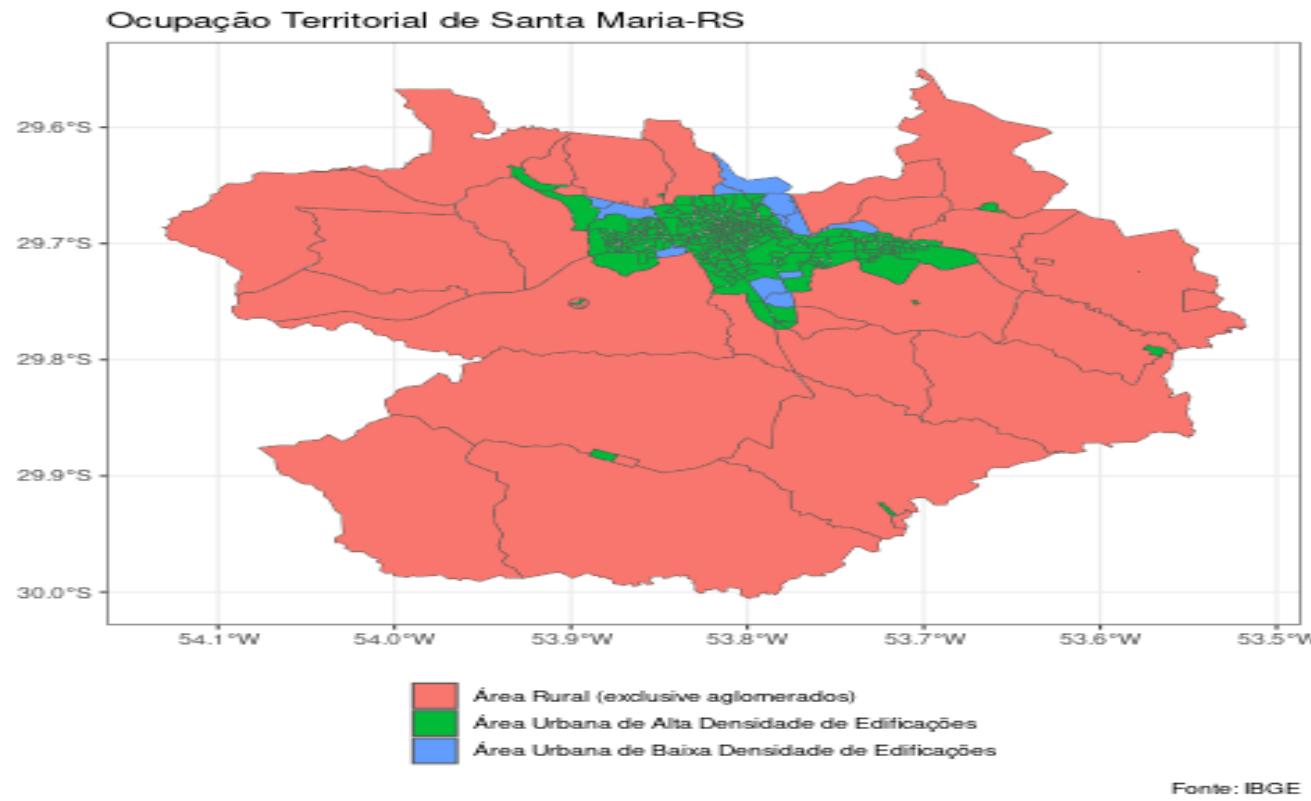
```
install.packages("geobr")
library(geobr)
```

O pacote **geobr** fornece mapas de divisões territoriais do Brasil e outros conjuntos de dados, conforme às funções que disponibiliza - ex: `read_state()`, `read_municipality()` e `read_health_facilities()`. Veja, por exemplo, a obtenção do mapa da cidade de Santa Maria segundo a ocupação territorial:

```
censo_sm <- geobr::read_census_tract(code_tract = 4316907, year = 2020, showProgress = FALSE)
ggplot(data = censo_sm, aes(fill=zone)) +
  geom_sf() +
  labs(title = "Ocupação Territorial de Santa Maria-RS", caption = "Fonte: IBGE") +
  theme_bw() + theme(legend.title = element_blank(),
                     legend.direction = "vertical", legend.position = "bottom")
```



Visualização espacial de dados - ggplot2



A partir de agora, trabalharemos com dados da PNADC.

Microdados da PNAD Contínua - Relembrando

Para trabalhar com os microdados da PNAD Contínua, vamos fazer uso dos pacotes:

- **PNADcIBGE** – possibilita baixar dados da PNAD Contínua.
- **survey** – possibilita produzir diversas operações estatísticas em dados com amostragem complexa.
- **srvyr** – uma extensão do pacote **survey** que permite o uso de funções do **dplyr** e sua sintaxe.

Vamos instalar e carregar os pacotes:

```
install.packages(c("PNADcIBGE", "survey", "srvyr"))
```

```
library(PNADcIBGE)
library(survey)
library(srvyr)
```

Microdados da PNAD Contínua - Relembrando

Continuaremos trabalhando com microdados do primeiro trimestre de 2023, que podem ser baixados/carregados usando funções do pacote **PNADcIBGE**.

- Baixar os dados:

```
get_pnadc(year = 2023, quarter = 1, savedir = "PNADC20231")
```

Microdados da PNAD Contínua - Relembrando

- Carregar os dados dos arquivos:

```
# Lendo os dados, excluindo os pesos replicados e mantendo alguns códigos numéricos
pnadc_2023q1 <- read_pnadc(
  microdata = "PNADC20231/PNADC_012023.txt",
  input_txt = "PNADC20231/input_PNADC_trimestral.txt"
) |>
  select(-(V1028001:V1028200)) |>
  mutate(code_state = as.numeric(UF),
         code_capital = as.numeric(Capital),
         anos_de_estudo = as.numeric(VD3005)) |>
  pnadc_labeller(dictionary.file = "PNADC20231/dicionario_PNADC_microdados_trimestral.xls") |>
  pnadc_deflator(deflator.file = "PNADC20231/deflator_PNADC_2023_trimestral_040506.xls") |>
  mutate(VD4020_real = VD4020 * Efetivo,
        VD4019_habitual = VD4019 * Habitual)
```

Microdados da PNAD Contínua – visualização espacial

Obtemos o mapa do Brasil e a localização das capitais usando funções do pacote **geobr**:

```
estados_br <- read_state(year = 2020, showProgress = FALSE)
capitais_br <- read_capitals(showProgress = FALSE) |>
  mutate(code_state = as.numeric(code_state))
```

```
glimpse(estados_br)
```

```
## Rows: 27
## Columns: 6
## $ code_state    <dbl> 11, 12, 13, 14, 15, 16, 17, 21, 22, 23, 24, 25, 26, 27, 2...
## $ abbrev_state  <chr> "RO", "AC", "AM", "RR", "PA", "AP", "TO", "MA", "PI", "CE...
## $ name_state    <chr> "Rondônia", "Acre", "Amazônas", "Roraima", "Pará", "Amapá...
## $ code_region   <dbl> 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, ...
## $ name_region   <chr> "Norte", "Norte", "Norte", "Norte", "Norte", "Norte", "No...
## $ geom          <MULTIPOLYGON [°]> MULTIPOLYGON (((-65,38 -10, ..., MULTIPOLYGON...
```

Perceba que o conjunto de dados do mapa possui o código dos estados (coluna `code_state`).

Microdados da PNAD Contínua – visualização espacial

Agora, como exemplo, podemos gerar as estimativas do rendimento médio mensal real efetivo das pessoas com 14 anos ou mais de idade, segundo o nível de instrução (**VD3004**) e unidade da federação (**code_state**). Não aplicaremos o plano amostral, mas sim, a média ponderada pelo peso dos domicílios e das pessoas (**V1028**):

```
rendimento_mensal <- pnadc_2023q1 |>
  summarise(rendimento_medio = weighted.mean(VD4020_real, w = V1028, na.rm = TRUE),
            .by = c(code_state, VD3004)) |>
  drop_na()
```

```
glimpse(rendimento_mensal)
```

```
## Rows: 189
## Columns: 3
## $ code_state      <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 12, 12, 12, 12, 12, 12, 1...
## $ VD3004          <fct> Médio completo ou equivalente, Superior completo, Fun...
## $ rendimento_medio <dbl> 2411,0, 4724,8, 2185,1, 2590,1, 2000,3, 1947,2, 1977,...
```

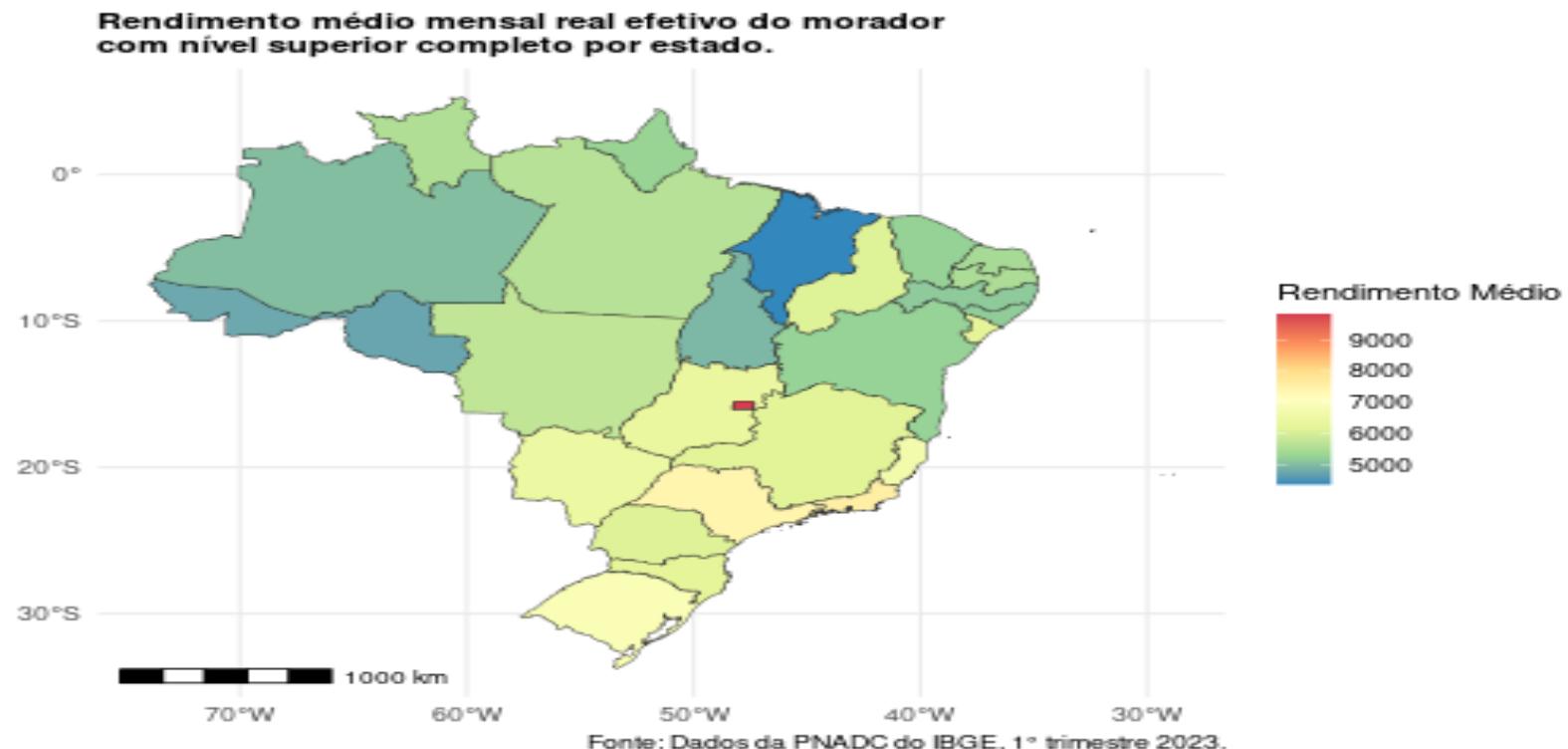
O conjunto de dados gerado também possui o código dos estados.

Microdados da PNAD Contínua – visualização espacial

Propositadamente, mantivemos no conjunto de dados a variável `code_state`, usada como agrupamento na geração da estatística. Dessa forma, podemos unir o mapa base com os dados de rendimento médio mensal real efetivo e gerar o seguinte gráfico:

```
my_cap = "Fonte: Dados da PNADC do IBGE, 1º trimestre 2023."  
left_join(estados_br, rendimento_mensal, by = "code_state") |>  
  filter(VD3004 == "Superior completo") |>  
ggplot() +  
  geom_sf(aes(fill = rendimento_medio), color = "gray20") +  
  labs(title = paste0("Rendimento médio mensal real efetivo do morador\ncom nível ",  
                     "superior completo por estado."), caption = my_cap) +  
  scale_fill_distiller(palette = "Spectral", name = "Rendimento Médio") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 11, face = 'bold')) +  
  ggspatial::annotation_scale()
```

Microdados da PNAD Contínua – visualização espacial



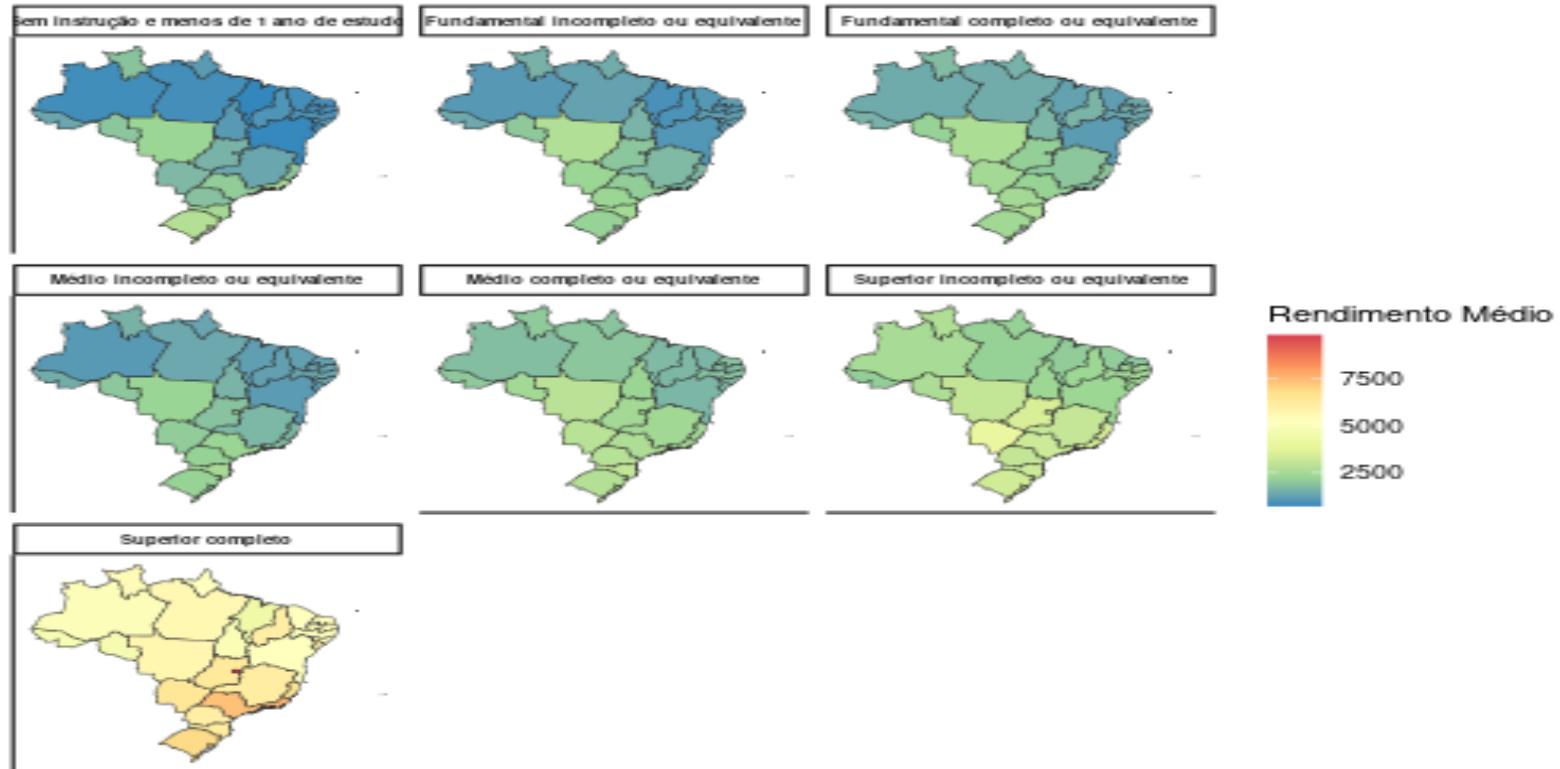
Microdados da PNAD Contínua – visualização espacial

Também podemos gerar mapas de distribuição do rendimento, conforme o nível de instrução:

```
left_join(estados_br, rendimento_mensal, by = "code_state") |>
  ggplot() +
  geom_sf(aes(fill = rendimento_medio), color = "gray20") +
  facet_wrap(~VD3004) +
  labs(title = paste0("Rendimento médio mensal real efetivo do morador com 14 anos\n",
                      "ou mais de idade por nível de instrução e unidade da federação."),
       caption = my_cap) +
  scale_fill_distiller(palette = "Spectral", name = "Rendimento Médio") +
  theme_classic() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        axis.ticks = element_blank(),
        strip.text = element_text(size = 6, face = "bold"),
        plot.title = element_text(size = 11, face = "bold"))
```

Microdados da PNAD Contínua – visualização espacial

Rendimento médio mensal real efetivo do morador com 14 anos ou mais de idade por nível de instrução e unidade da federação.



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Vamos gerar também estimativas do rendimento médio mensal real efetivo e anos de estudo por estado e para as capitais. Usamos a função `st_sf()`, após as demais operações, para converter o objeto gerado na junção dos conjuntos de dados ao tipo `sf`.

```
estados <- pnadc_2023q1 |>
  summarise(rendimento_medio = weighted.mean(VD4020_real, w = V1028, na.rm = TRUE),
            anos_de_estudo = weighted.mean(anos_de_estudo, w = V1028, na.rm = TRUE),
            .by = "code_state") |>
  drop_na() |>
  right_join(estados_br, by = "code_state") |> st_sf()

capitais <- pnadc_2023q1 |>
  summarise(rendimento_medio = weighted.mean(VD4020_real, w = V1028, na.rm = TRUE),
            anos_de_estudo = weighted.mean(anos_de_estudo, w = V1028, na.rm = TRUE),
            .by = "code_capital") |>
  drop_na() |>
  right_join(capitais_br, by = c("code_capital" = "code_state")) |> st_sf()
```

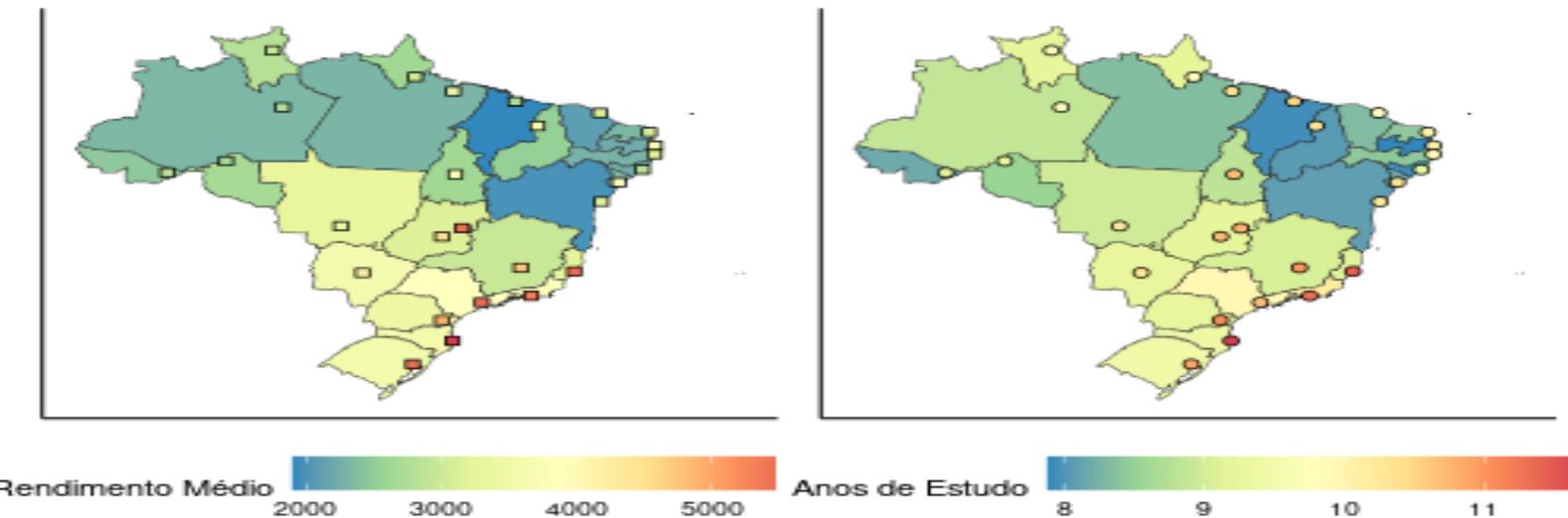
Microdados da PNAD Contínua – visualização espacial

Produziremos os seguintes gráficos:

```
my_title <- paste0("Rendimento médio mensal real efetivo e anos de estudo do morador com\n",
                   "14 anos ou mais de idade, nos estados e capitais.")
my_theme = theme_classic() + theme(axis.text = element_blank(), axis.title = element_blank(),
                                    axis.ticks = element_blank(), legend.key.width = unit(1.2, "cm"),
                                    legend.position = "bottom", legend.direction = "horizontal")
p1 <- ggplot(data = estados, aes(fill = rendimento_medio)) +
  geom_sf(color = "gray20") +
  geom_sf(data = capitais, shape = 22, size = 2) +
  scale_fill_distiller(palette = "Spectral", name = "Rendimento Médio") + my_theme
p2 <- ggplot(data = estados, aes(fill = anos_de_estudo)) +
  geom_sf(color = "gray20") +
  geom_sf(data = capitais, shape = 21, size = 2) +
  scale_fill_distiller(palette = "Spectral", name = "Anos de Estudo") + my_theme
# install.packages("patchwork")
library(patchwork)
p1 + p2 + plot_annotation(title = my_title, caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Rendimento médio mensal real efetivo e anos de estudo do morador com 14 anos ou mais de idade, nos estados e capitais.



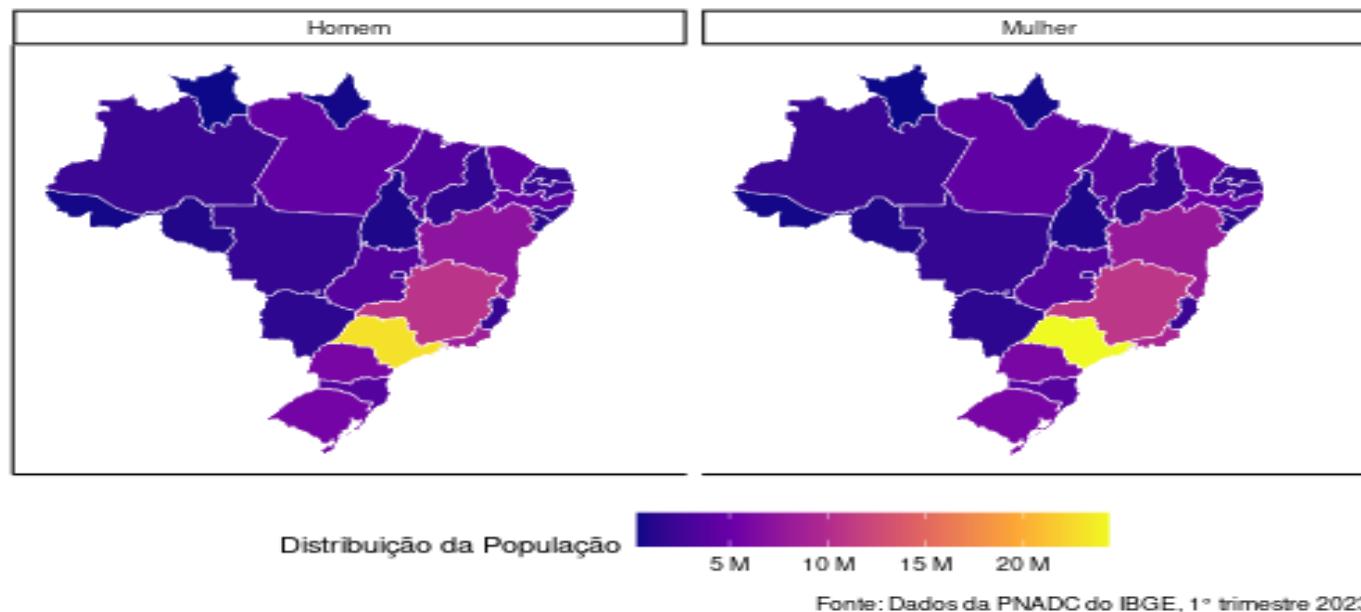
Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Distribuição da população por sexo e unidade da federação? Ao somar os valores da coluna **V1028** (peso dos domicílios e das pessoas), estimamos o total de habitantes da população brasileira – formataremos em milhões de habitantes com `label_number()`. Como a estimativa será agregada por sexo e unidade da federação, os totais irão considerar essas divisões categóricas.

```
pnadc_2023q1 |>
  summarise(total = sum(V1028), .by = c(V2007, code_state)) |>
  right_join(estados_br, by = "code_state") |>
  st_sf() |>
  ggplot() +
  geom_sf(aes(fill = total), color = "white") +
  facet_wrap(~V2007) +
  labs(caption = my_cap) +
  scale_fill_viridis_c(option = "C", name = "Distribuição da População",
                       labels = scales::label_number(suffix = " M", scale = 1e-6)) +
  theme_classic() + my_theme
```

Microdados da PNAD Contínua – visualização espacial



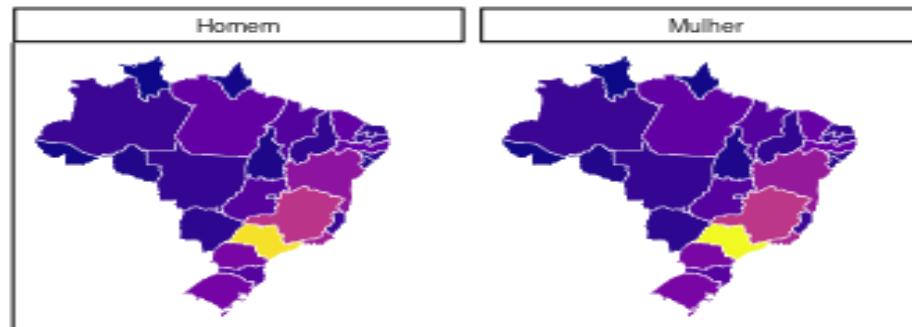
Microdados da PNAD Contínua – visualização espacial

Número total e proporção da população por sexo e unidade da federação? Podemos agrregar somente por estado, para estimar com a função `count()` a distribuição da população segundo o sexo, considerando pesos (`wt = V1028`). Os valores serão armazenados na variável `n`, e a proporção pode ser obtida ao dividir o total por sexo (`n`) em cada estado pela soma desses totais (homens + mulheres = `sum(n)`), sendo que o agrupamento permite isso:

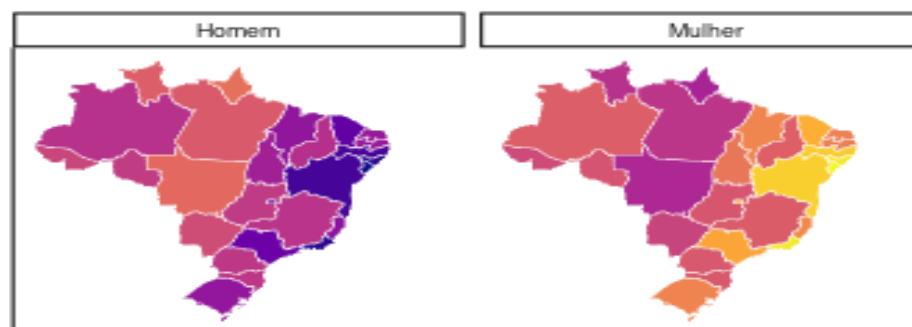
```
escala1 <- scale_fill_viridis_c(option = "C", name = "Total",
                                  labels = scales::label_number(suffix = " M", scale = 1e-6))
escala2 <- scale_fill_viridis_c(option = "C", name = "Proporção", labels = scales::percent)
populacao <- pnadc_2023q1 |>
  group_by(code_state) |>
  count(V2007, wt = V1028) |>
  mutate(proporcao = n/sum(n)) |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(data = populacao) + geom_sf(aes(fill = n), color = "white") +
  facet_wrap(~V2007) + escala1 + my_theme
p2 <- ggplot(data = populacao) + geom_sf(aes(fill = proporcao), color = "white") +
  facet_wrap(~V2007) + escala2 + my_theme
p1 / p2 + plot_annotation(title = "Distribuição da população.", caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Distribuição da população.



Total
5 M 10 M 15 M 20 M



Proporção
48.0% 49.0% 50.0% 51.0% 52.0%

Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

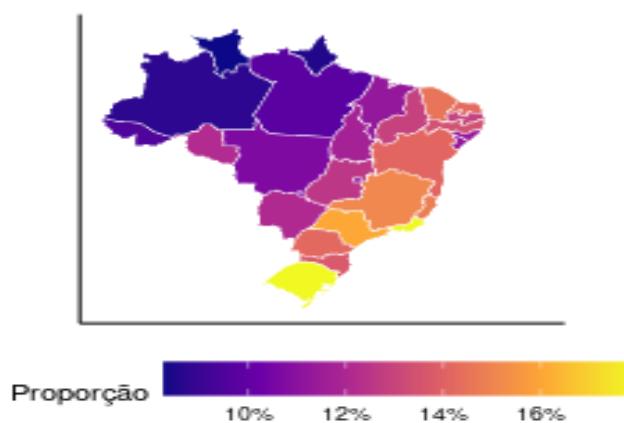
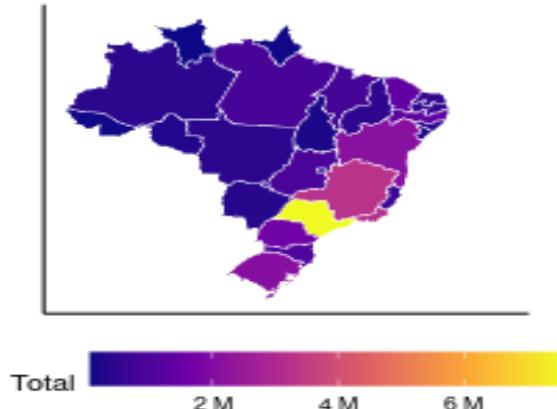
Microdados da PNAD Contínua – visualização espacial

Número total e proporção das pessoas com 60 anos ou mais de idade por unidade da federação?

```
populacao <- pnadc_2023q1 |>
  group_by(code_state) |>
  count(V2009 > 60, wt = V1028) |>
  mutate(proporcao = n/sum(n)) |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(data = filter(populacao, `V2009 > 60` == TRUE)) +
  geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p2 <- ggplot(data = filter(populacao, `V2009 > 60` == TRUE)) +
  geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p1 / p2 + plot_annotation(
  title = "Distribuição da população com idade\ningual ou superior a 60 anos.",
  caption = my_cap
)
```

Microdados da PNAD Contínua – visualização espacial

Distribuição da população com idade igual ou superior a 60 anos.



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

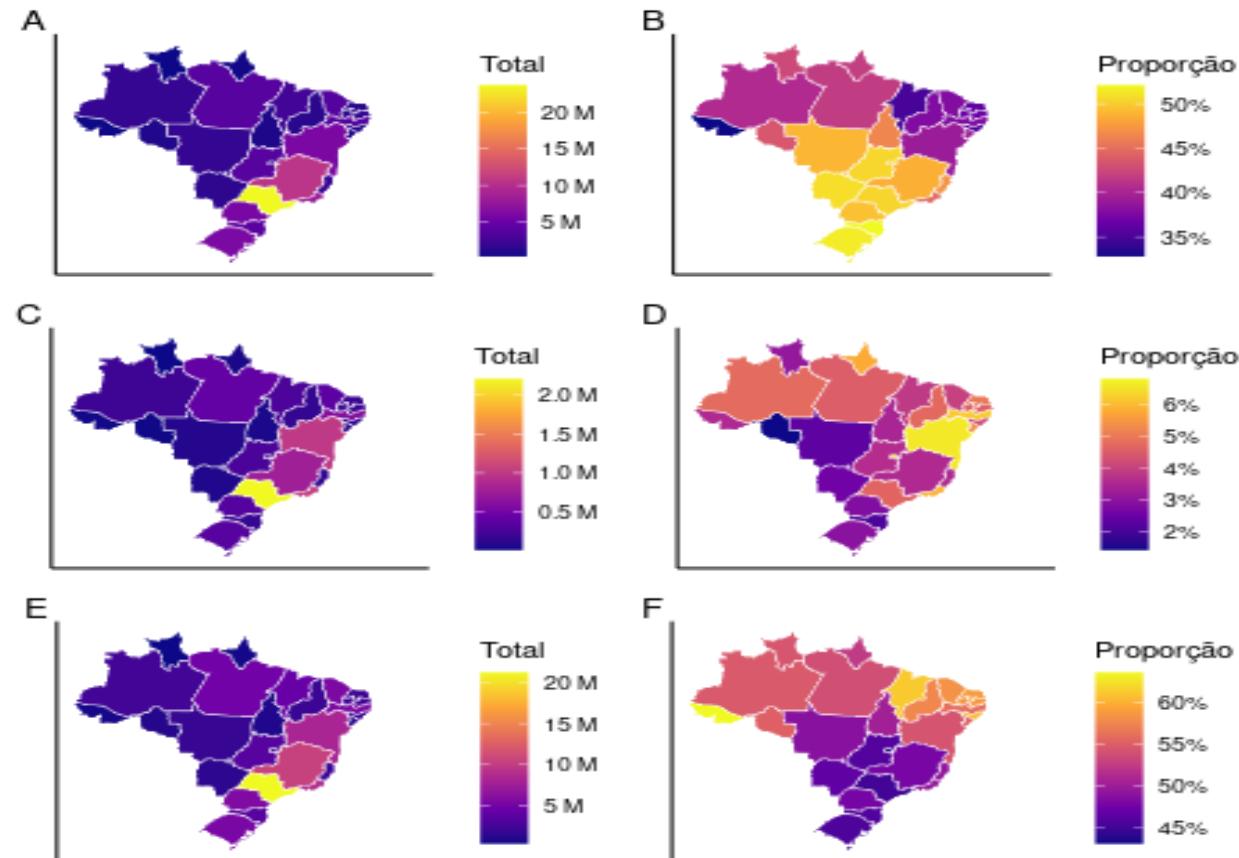
Microdados da PNAD Contínua – visualização espacial

Número total e proporção dos trabalhadores, segundo a condição de ocupação?

```
my_title = "Condição da Força de Trabalho."
my_subtitle = "(A,B) Ocupados, (C,D) Desocupados, (E, F) Não aplicável"
my_theme = theme_classic() + theme(axis.text = element_blank(), axis.title = element_blank(),
                                    axis.ticks = element_blank())
forca_trabalho <- pnadc_2023q1 |> group_by(code_state) |> count(VD4002, wt = V1028) |>
  mutate(proporcao = n/sum(n) ) |> right_join(estados_br, by = "code_state") |> st_sf()
p <- ggplot(data = filter(forca_trabalho, VD4002 == "Pessoas ocupadas"))
p1 <- p + geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p2 <- p + geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p <- ggplot(data = filter(forca_trabalho, VD4002 == "Pessoas desocupadas"))
p3 <- p + geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p4 <- p + geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p <- ggplot(data = filter(forca_trabalho, is.na(VD4002)))
p5 <- p + geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p6 <- p + geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
(p1 + p2) / (p3 + p4) / (p5 + p6) + plot_annotation(tag_levels = 'A', title = my_title,
                                                    subtitle = my_subtitle, caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Condição da Força de Trabalho.
(A,B) Ocupados, (C,D) Desocupados, (E, F) Não aplicável



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

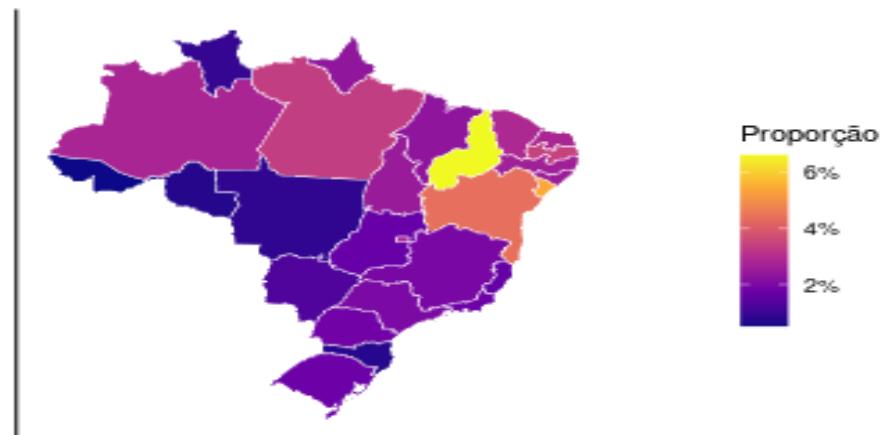
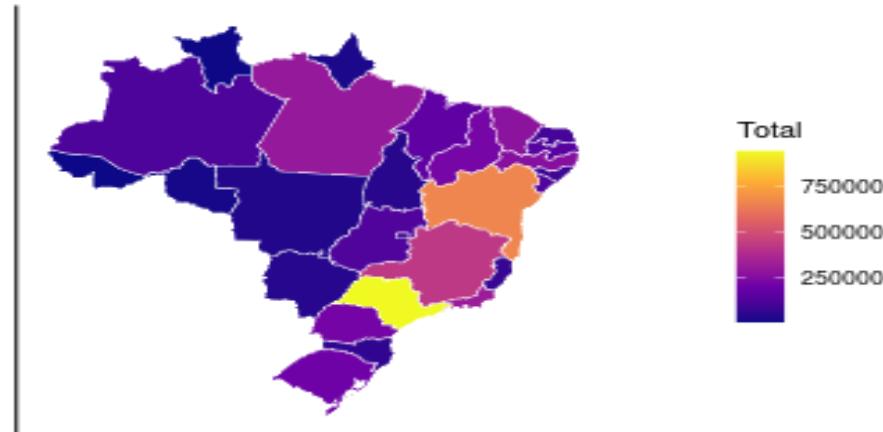
Microdados da PNAD Contínua – visualização espacial

Número total e proporção dos subocupados (insuficiência de horas habitualmente trabalhadas) por unidade da federação?

```
populacao <- pnadc_2023q1 |>
  group_by(code_state) |>
  count(condicao = VD4004A, wt = V1028) |>
  mutate(proporcao = n/sum(n)) |>
  filter(condicao == "Pessoas subocupadas") |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(data = populacao) +
  geom_sf(aes(fill = n), color = "white") +
  scale_fill_viridis_c(option = "C", name = "Total") + my_theme
p2 <- ggplot(data = populacao) +
  geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p1 / p2 + plot_annotation(title = "Subocupados no Brasil.",
                           caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Subocupados no Brasil.



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Número total e proporção dos desalentados por unidade da federação?

```
populacao <- pnadc_2023q1 |>
  group_by(code_state) |>
  count(condicao = VD4005, wt = V1028) |>
  mutate(proporcao = n/sum(n)) |>
  filter(condicao == "Pessoas desalentadas") |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(data = populacao) +
  geom_sf(aes(fill = n), color = "white") +
  scale_fill_viridis_c(option = "C", name = "Total") + my_theme
p2 <- ggplot(data = populacao) +
  geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p1 / p2 + plot_annotation(title = "Desalentados no Brasil.",
                           caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Desalentados no Brasil.



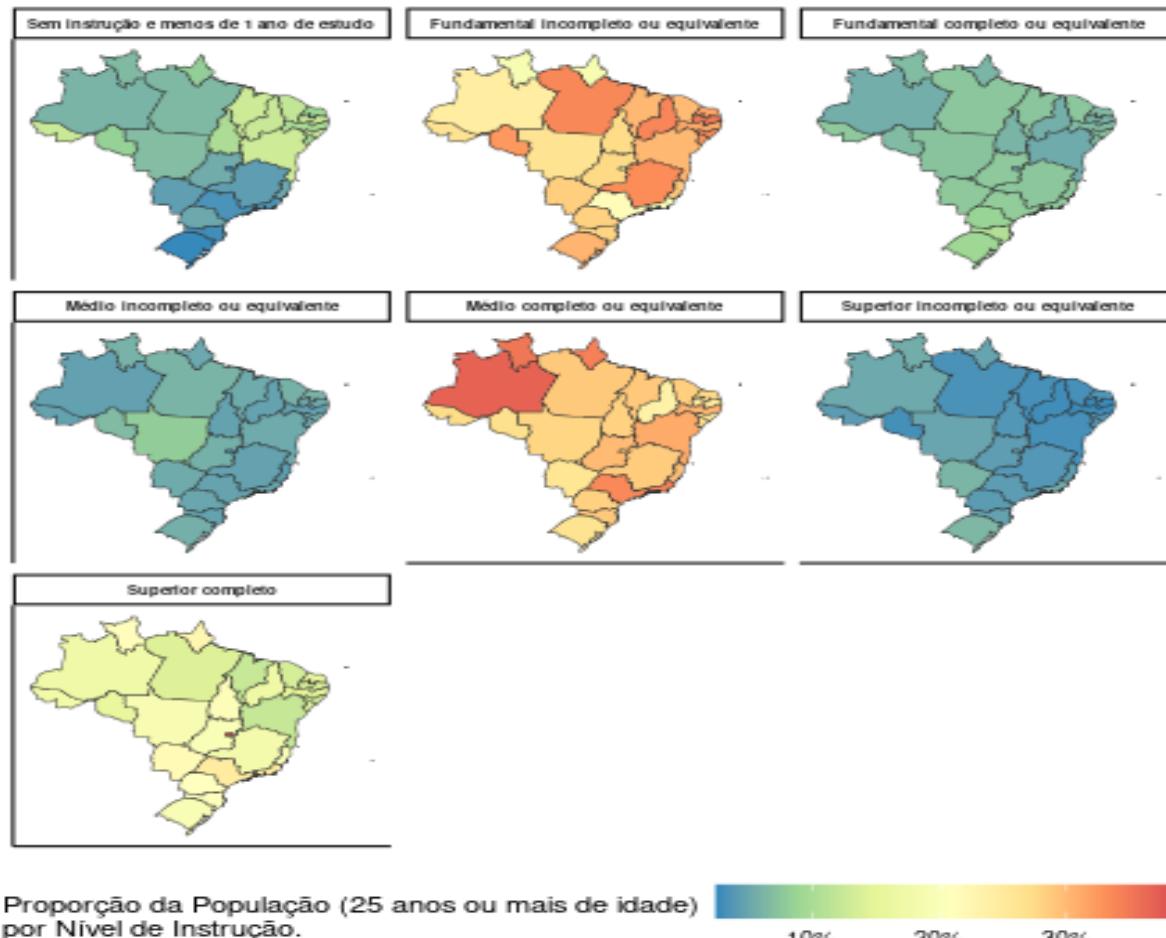
Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Proporção da população, com idade igual ou superior a 25 anos, por estado e segundo o nível de instrução?

```
my_theme2 <- theme_classic() +  
  theme(axis.text = element_blank(), axis.title = element_blank(),  
        axis.ticks = element_blank(), strip.text = element_text(size = 6, face = "bold"),  
        legend.direction = "horizontal", legend.position = "bottom",  
        legend.key.width = unit(1.2, "cm"))  
pnadc_2023q1 |>  
  filter(V2009 >= 25) |>  
  group_by(code_state) |>  
  count(VD3004, wt = V1028) |>  
  mutate(proporcao = n/sum(n)) |>  
  right_join(estados_br, by = "code_state") |> drop_na() |> st_sf() |>  
  ggplot() +  
    geom_sf(aes(fill = proporcao), color = "gray20") + facet_wrap(~VD3004) +  
    scale_fill_distiller(palette = "Spectral",  
                         name = paste0("Proporção da População (25 anos ou mais de idade)",  
                                      "\npor Nível de Instrução."), labels = scales::percent) +  
    labs(caption = my_cap) + my_theme2
```

Microdados da PNAD Contínua – visualização espacial



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Parcela da população nos níveis de instrução: "Sem instrução e menos de 1 ano de estudo", "Superior completo"?

```
my_title = "Parcela da população (25 anos ou mais de idade):"
my_subtitle = "(A,B) Sem instrução e menos de 1 ano de estudo, (C,D) Superior completo"
nivel_instrucao <- pnadc_2023q1 |>
  filter(V2009 >= 25) |>
  group_by(code_state) |>
  count(VD3004, wt = V1028) |>
  mutate(proporcao = n/sum(n) ) |>
  right_join(estados_br, by = "code_state") |> st_sf()
p <- ggplot(data = filter(nivel_instrucao,
                           VD3004 == "Sem instrução e menos de 1 ano de estudo"))
p1 <- p + geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p2 <- p + geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
p <- ggplot(data = filter(nivel_instrucao, VD3004 == "Superior completo"))
p3 <- p + geom_sf(aes(fill = n), color = "white") + escala1 + my_theme
p4 <- p + geom_sf(aes(fill = proporcao), color = "white") + escala2 + my_theme
(p1|p2)/(p3+p4) + plot_annotation(tag_levels = 'A', title = my_title, subtitle = my_subtitle,
                                    caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Parcela da população (25 anos ou mais de idade):
(A,B) Sem instrução e menos de 1 ano de estudo, (C,D) Superior completo



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Situação do analfabetismo das pessoas com 15 anos ou mais de idade por unidade da federação?

```
populacao <- pnadc_2023q1 |>
  filter(V2009 >= 15) |>
  group_by(code_state) |>
  count(sabe_ler = V3001, wt = V1028) |>
  mutate(proporcao = n/sum(n)) |>
  filter(sabe_ler == "Não") |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(populacao) + geom_sf(aes(fill = n), color = "white") +
  scale_fill_viridis_c(option = "C", name = "Total") + my_theme
p2 <- ggplot(populacao) + geom_sf(aes(fill = proporcao), color = "white") +
  escala2 + my_theme
p1 / p2 + plot_annotation(title = paste0("Analfabetismo no Brasil ",
                                         "(pessoas com 15 anos ou mais de idade)."),
                           caption = my_cap)
```

Microdados da PNAD Contínua – visualização espacial

Analfabetismo no Brasil (pessoas com 15 anos ou mais de idade).



Fonte: Dados da PNADC do IBGE, 1º trimestre 2023.

Microdados da PNAD Contínua – visualização espacial

Vamos ver exemplos mais elaborados para produzir gráficos painelizados. Antes, produziremos um conjunto de dados com o total e a proporção das pessoas segundo o nível de curso que frequentam.

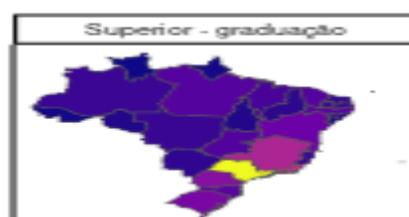
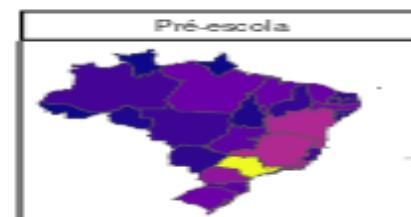
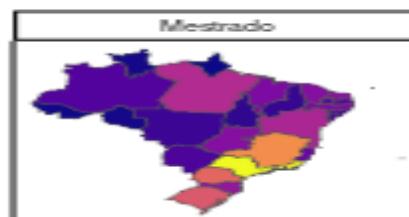
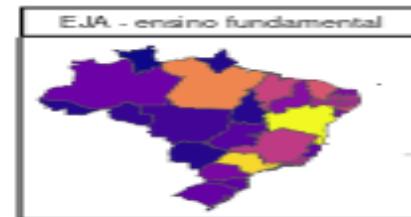
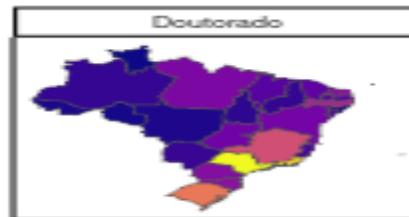
```
my_theme = theme_classic() +  
  theme(axis.text = element_blank(), axis.title = element_blank(),  
        axis.ticks = element_blank(), legend.position = "bottom",  
        legend.direction = "horizontal", legend.key.width = unit(.6,"cm"),  
        legend.text = element_text(size = 5), legend.title = element_text(size=7),  
        strip.text = element_text(size = 7))  
  
populacao <- pnadc_2023q1 |>  
  group_by(code_state) |>  
  count(curso = as.character(V3003A), wt = V1028) |>  
  mutate(proporcao = n/sum(n),  
        curso = case_when(  
          curso == "Educação de jovens e adultos (EJA) do ensino fundamental" ~  
            "EJA - ensino fundamental",  
          curso == "Educação de jovens e adultos (EJA) do ensino médio" ~  
            "EJA - ensino médio",  
          .default = curso)  
  ) |> right_join(estados_br, by = "code_state") |> st_sf()
```

Microdados da PNAD Contínua – visualização espacial

Qual o total populacional cursando cada nível de ensino por unidade da federação? Para isso, dividiremos nossos dados com a função `base::split()`, segundo o nível de ensino. Depois, usaremos a função `purrr::map()` para mapear cada divisão categórica (nível do curso) do conjunto de dados no respectivo painel gráfico, gerado pelo `ggplot()`. Dessa forma, as escalas serão geradas individualmente para cada painel – comportamento aqui almejado. Ainda precisamos da função `cowplot::plot_grid()` para recompor os painéis individuais em um único gráfico.

```
# install.packages("cowplot")
populacao %>%
  split(. $curso) %>%
  purrr::map(
    ~ ggplot(., aes(fill = n)) +
      geom_sf() +
      facet_wrap(~curso) +
      escala1 + my_theme
  ) %>%
  cowplot::plot_grid(plotlist = .)
```

Microdados da PNAD Contínua – visualização espacial

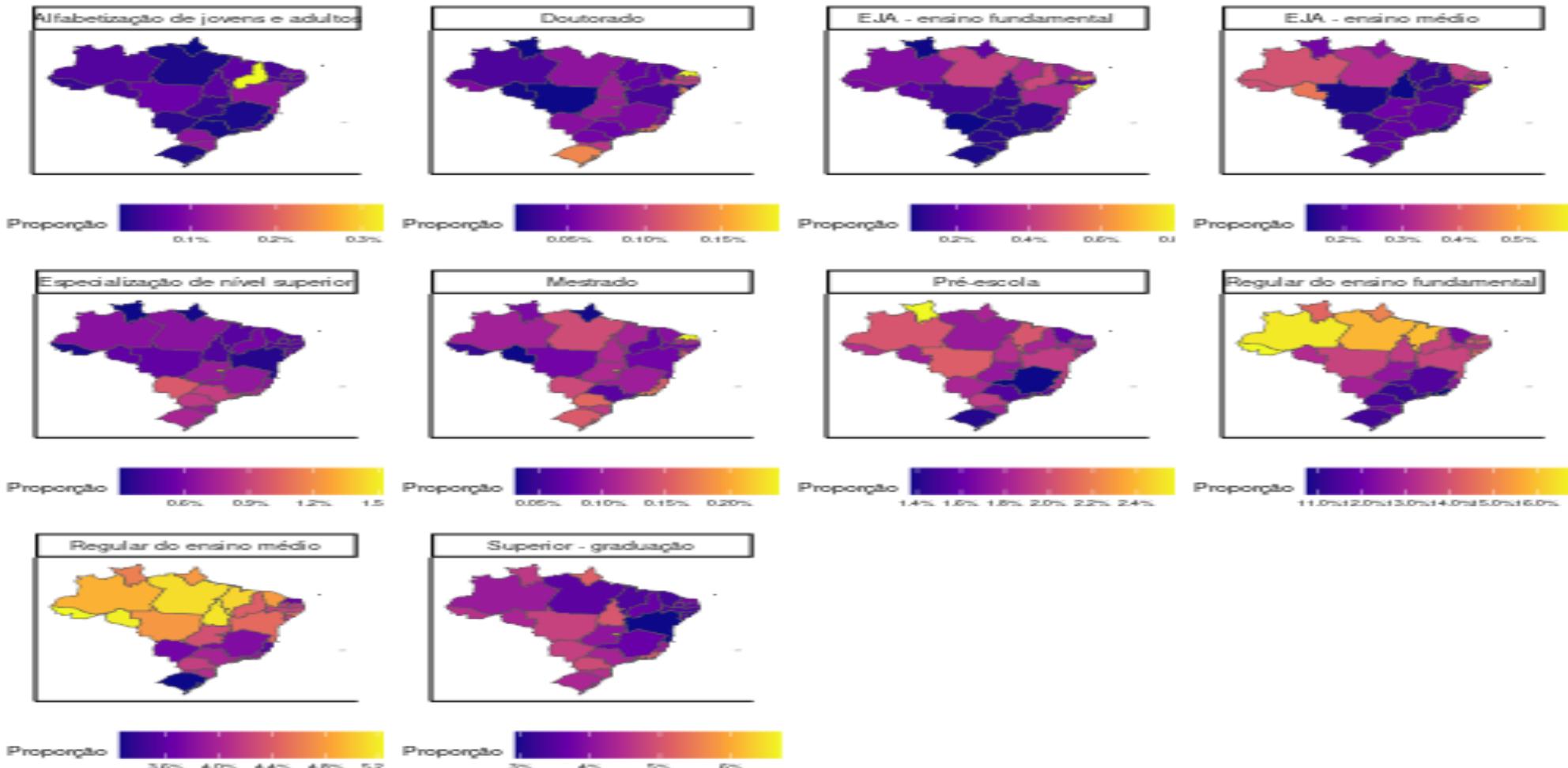


Microdados da PNAD Contínua – visualização espacial

Qual a proporção da população cursando cada nível de ensino por unidade da federação?

```
populacao %>%
  split(. $curso) %>%
  purrr::map(
    ~ ggplot(., aes(fill = proporcao)) +
      geom_sf() +
      facet_wrap(~curso) +
      escala2 + my_theme
  ) %>%
  cowplot::plot_grid(plotlist = .)
```

Microdados da PNAD Contínua – visualização espacial



Microdados da PNAD Contínua – visualização espacial

Agora, vamos aplicar o plano amostral com **PNADCIBGE**, mas antes selecionamos algumas variáveis:

```
pnadc_2023q1 <- pnadc_2023q1 |>  
  select(Ano, Trimestre, UF, code_state, Capital, Estrato, UPA, posest, V1027, V1028, V1029,  
         V2009, VD3004, anos_de_estudo, VD4019_habitual, VD4020_real, ID_DOMICILIO)  
  
pnadc_2023q1 <- PNADCIBGE::pnadc_design(data_pnadc = pnadc_2023q1)
```

O pacote **convey** (documentação [online](#)) permite estimar diversas medidas de concentração de renda e desigualdade em dados com amostragem complexa como o índice de Gini. Vamos carregar esse pacote e preparar os dados, sendo necessário converter o tipo do objeto com a função **convey::convey_prep()**

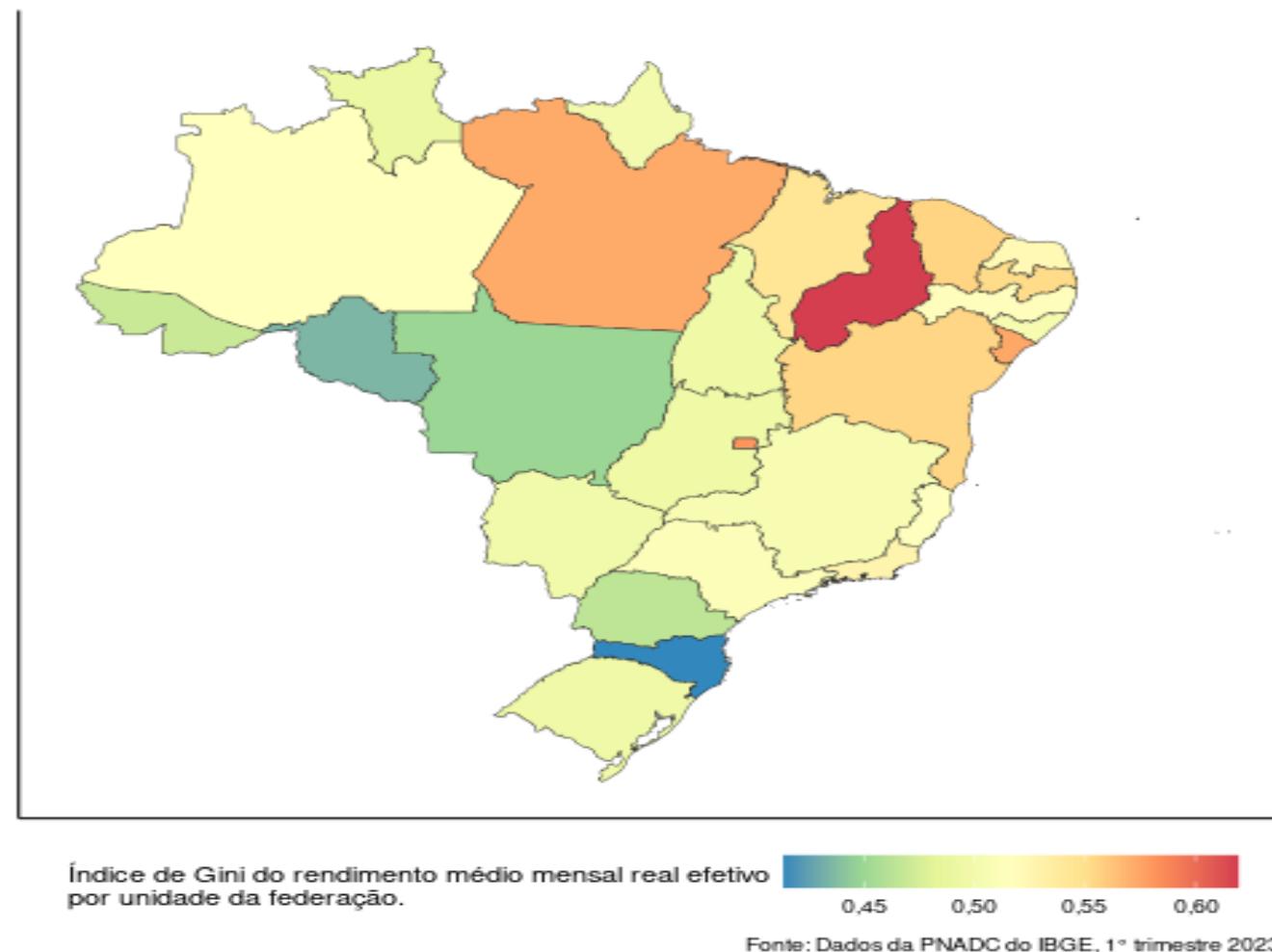
```
install.packages("convey")  
library(convey)  
  
pnadc_2023q1 <- convey_prep(design = pnadc_2023q1)
```

Microdados da PNAD Contínua – visualização espacial

Assim, podemos calcular o índice de Gini do rendimento médio mensal real por unidade da federação e gerar um mapa. Para isso, usaremos a função `svyby()` com o parâmetro `FUN = convey::svygini()`. O índice de Gini irá variar entre 0 e 1 quanto a concentração de renda: 0 = máxima igualdade e 1 = máxima desigualdade.

```
svyby(formula = ~VD4020_real, by = ~code_state, design = pnadc_2023q1,
       FUN = svygini, na.rm = TRUE) |>
  right_join(estados_br, by = "code_state") |> st_sf() |>
  ggplot() +
  geom_sf(aes(fill = VD4020_real), color = "gray20") +
  scale_fill_distiller(
    palette = "Spectral",
    name = "Índice de Gini do rendimento médio mensal real efetivo\npor unidade da federação."
  ) +
  labs(caption = my_cap) + my_theme2
```

Microdados da PNAD Contínua – visualização espacial

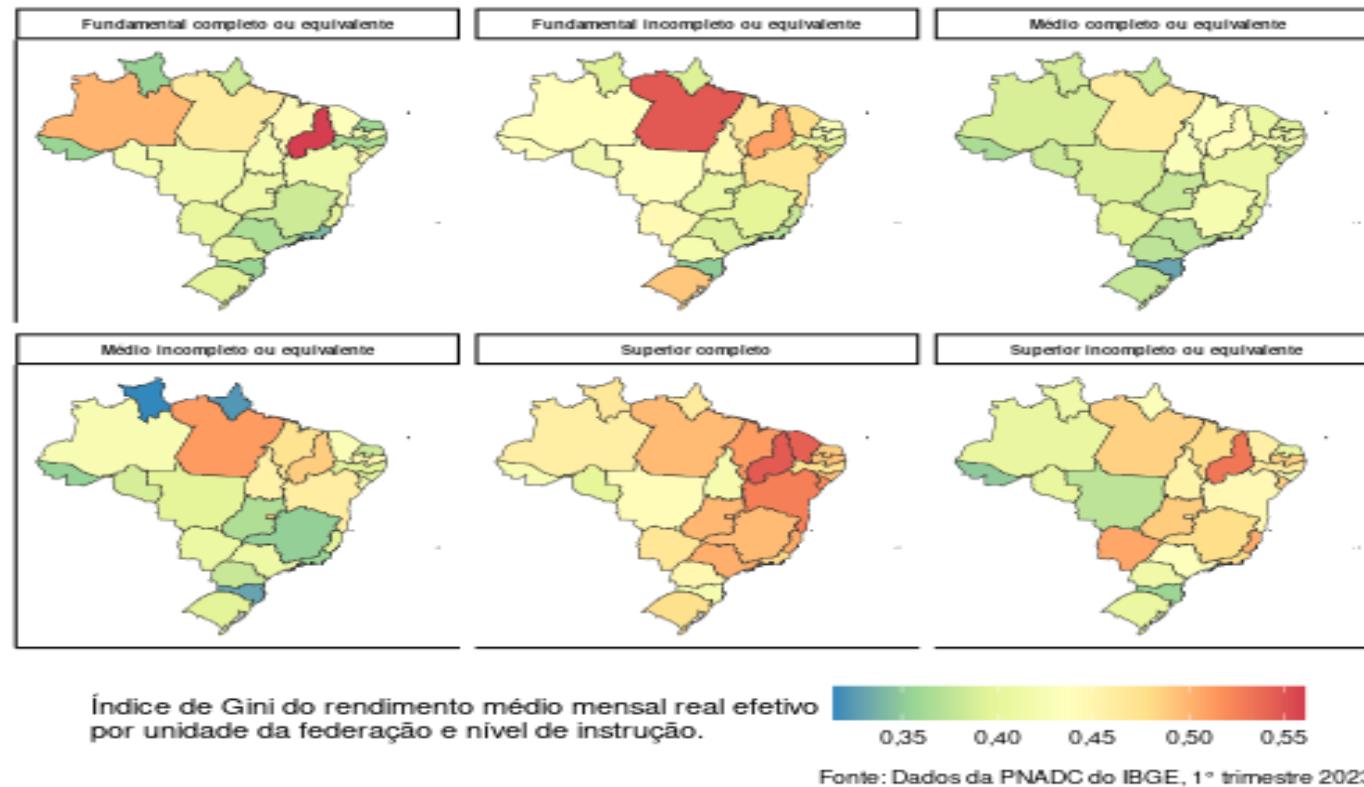


Microdados da PNAD Contínua – visualização espacial

Vamos calcular o índice de Gini por unidade da federação, considerando cada nível de instrução. Combinamos as funções `svyby()` e `interaction()` para obter os agrupamentos desejados. Esta última permite o cruzamento de duas ou mais variáveis categóricas e produzirá a coluna `interaction(code_state, VD3004)`, contendo as combinações possíveis. Após a geração da estatística, extrairemos com `gsub()` os níveis categóricos originais, armazenando os valores nas colunas `code_state` e `instrucao`, conforme.

```
svyby(formula = ~VD4020_real, by = ~interaction(code_state, VD3004), design = pnadc_2023q1,
       FUN = svygini, na.rm = TRUE) |>
  mutate(code_state = as.numeric(gsub("\\"D", "", `interaction(code_state, VD3004)`)),
         instrucao = gsub(".*\\".", "", `interaction(code_state, VD3004)`)) |>
  right_join(estados_br, by = "code_state") |> st_sf() |>
  ggplot() +
  geom_sf(aes(fill = VD4020_real), color = "gray20") + facet_wrap(~instrucao) +
  scale_fill_distiller(
    palette = "Spectral",
    name = paste0("Índice de Gini do rendimento médio mensal real efetivo\n",
                 "por unidade da federação e nível de instrução."))
) +
  labs(caption = my_cap) + my_theme2
```

Microdados da PNAD Contínua – visualização espacial

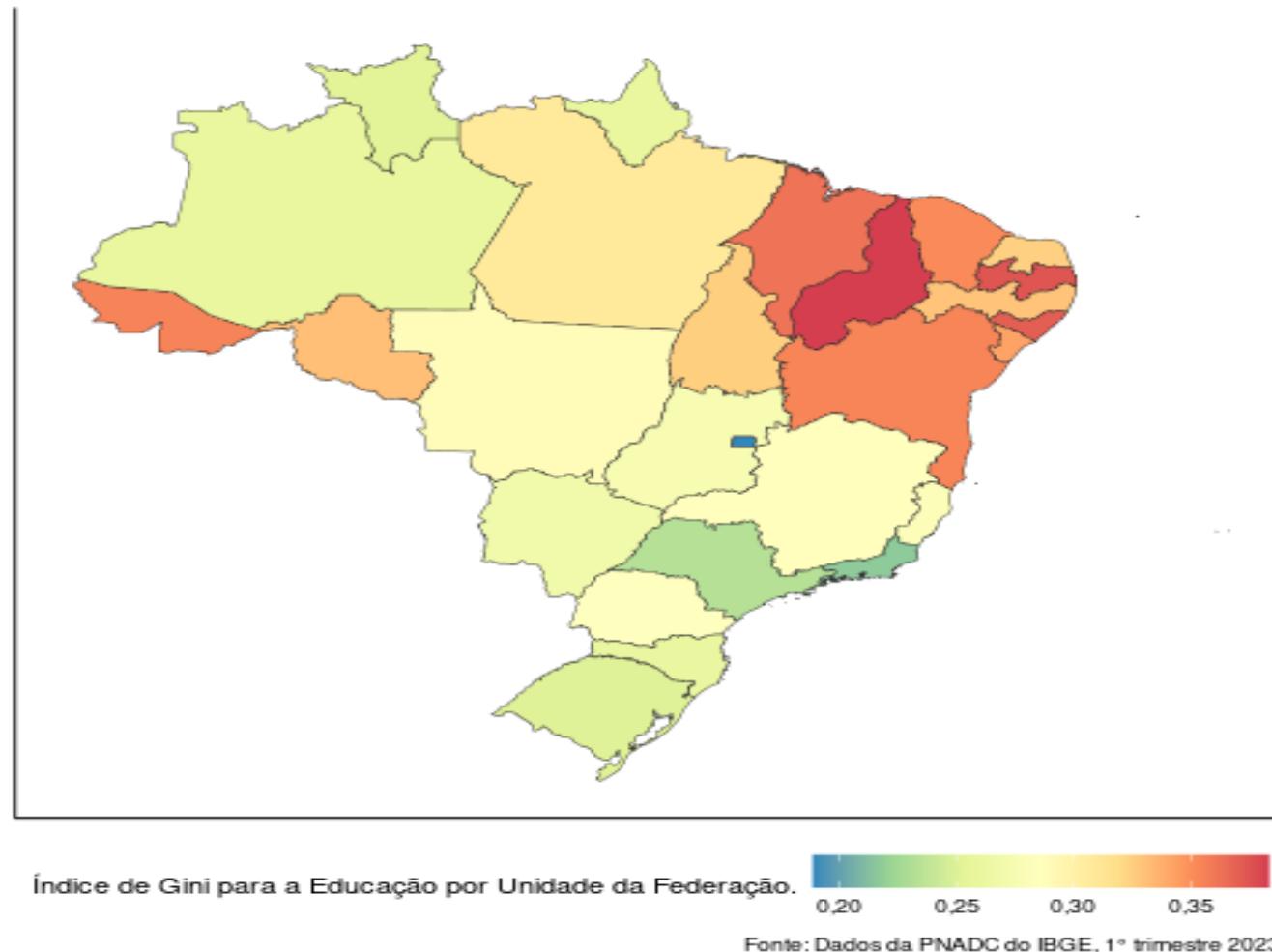


Microdados da PNAD Contínua – visualização espacial

Também podemos calcular o índice de Gini referente à educação (anos de estudo) por unidade da federação. Consideraremos a média de anos estudo das pessoas com 25 anos ou mais de idade:

```
svyby(formula = ~anos_de_estudo, by = ~code_state, design = subset(pnadc_2023q1, V2009 >= 25),  
       FUN = svygini, na.rm = TRUE) |>  
  right_join(estados_br, by = "code_state") |> st_sf() |>  
  ggplot() +  
    geom_sf(aes(fill = anos_de_estudo), color = "gray20") +  
    scale_fill_distiller(  
      palette = "Spectral",  
      name = "Índice de Gini para a Educação por Unidade da Federação."  
    ) +  
    labs(caption = my_cap) + my_theme2
```

Microdados da PNAD Contínua – visualização espacial



Microdados da PNAD Contínua – visualização espacial

Agora, vamos obter microdados anuais acumulados na 5^a entrevista do ano de 2022, conforme:

- Baixar os dados:

```
get_pnadc(year = 2022, interview = 5, savedir = "PNADC2022IT5")
```

Microdados da PNAD Contínua – visualização espacial

- Carregar os dados dos arquivos:

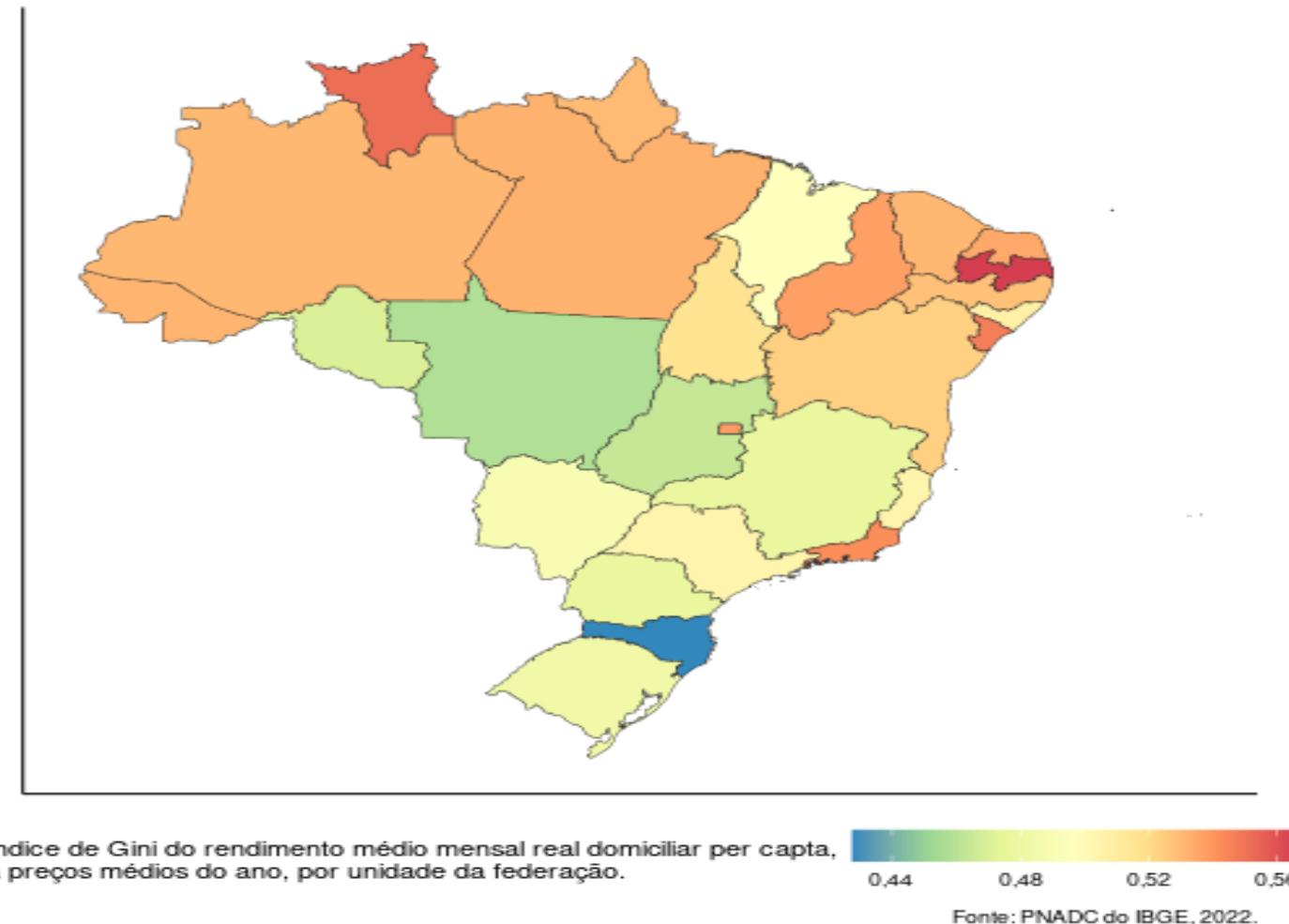
```
# Ler variáveis específicas, excluir os pesos replicados e manter alguns códigos numéricos
pnadc_2022_it5 <- read_pnadc(
  microdata = "PNADC2022IT5/PNADC_2022_visita5.txt",
  input_txt = "PNADC2022IT5/input_PNADC_2022_visita5_20230811.txt",
  vars = c("Ano", "Trimestre", "UF", "Capital", "UPA", "Estrato",
          "V1008", "V1014", "V1030", "V1031", "V1032", "V1034",
          "posest", "posest_sxi", "V2003", "V2009",
          "VD3004", "VD3005", "VD5005")) |>
  select(-(V1032001:V1032200)) |>
  mutate(code_state = as.numeric(UF),
         code_capital = as.numeric(Capital),
         anos_de_estudo = as.numeric(VD3005)) |>
  pnadc_labeller(
    dictionary.file = "PNADC2022IT5/dicionario_PNADC_microdados_2022_visita5_20230811.xls"
  ) |>
  pnadc_deflator(deflator.file = "PNADC2022IT5/deflator_PNADC_2022.xls") |>
  mutate(VD5005_real = VD5005 * C01e) |> # obter o rendimento efetivo corrigido
  PNADcIBGE::pnadc_design() |> convey::convey_prep()
```

Microdados da PNAD Contínua – visualização espacial

Com o novo conjunto de dados, podemos calcular o índice de Gini do rendimento médio mensal real domiciliar per capita (**VD5005_real**) por unidade da federação e gerar uma visualização espacial, conforme:

```
svyby(formula = ~VD5005_real, by = ~code_state, design = pnadc_2022_it5,
       FUN = svygini, na.rm = TRUE) |>
  right_join(estados_br, by = "code_state") |> st_sf() |>
  ggplot() +
  geom_sf(aes(fill = VD5005_real), color = "gray20") +
  scale_fill_distiller(
    palette = "Spectral",
    name = paste0("Índice de Gini do rendimento médio mensal real domiciliar per capita",
                 "\na preços médios do ano, por unidade da federação."))
) +
  labs(caption = "Fonte: PNADC do IBGE, 2022.") + my_theme2
```

Microdados da PNAD Contínua – visualização espacial

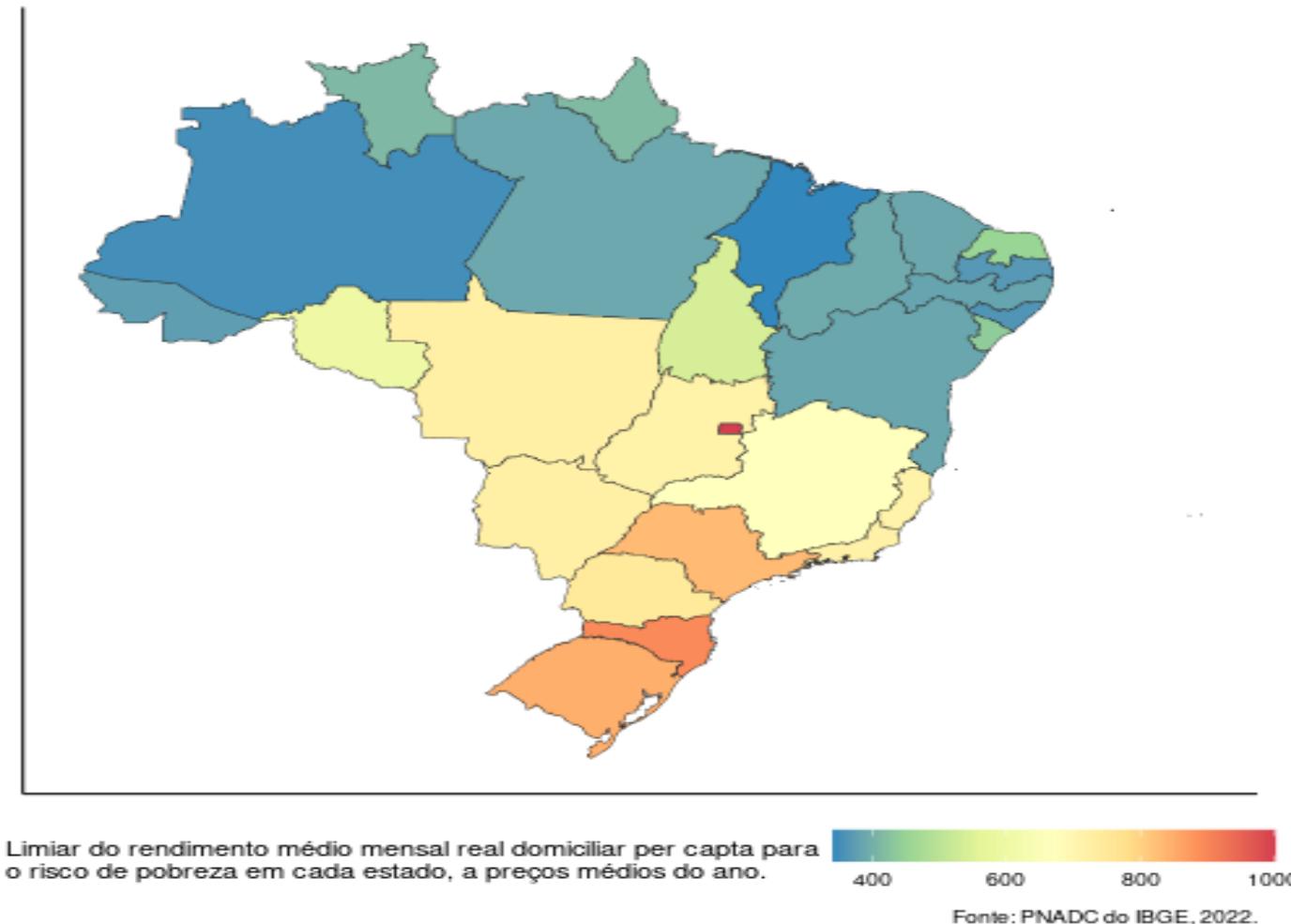


Microdados da PNAD Contínua – visualização espacial

Com a função `convey::svyby()`, podemos calcular os valores limiares de rendimento para o risco de pobreza (60% da mediana do rendimento) em cada estado. Rendimentos abaixo desses limiares podem indicar pessoas em risco de pobreza. Visualizaremos esses valores limiares em um mapa, conforme:

```
svyby(formula = ~VD5005_real, by = ~code_state, design = pnadc_2022_it5,
       FUN = svyby, na.rm = TRUE) |>
  right_join(estados_br, by = "code_state") |> st_sf() |>
  ggplot() + geom_sf(aes(fill = VD5005_real), color = "gray20") +
  scale_fill_distiller(
    palette = "Spectral",
    name = paste0("Limiar do rendimento médio mensal real domiciliar per capita para",
                 "\no risco de pobreza em cada estado, a preços médios do ano.")
  ) +
  labs(caption = "Fonte: PNADC do IBGE, 2022.") + my_theme2
```

Microdados da PNAD Contínua – visualização espacial

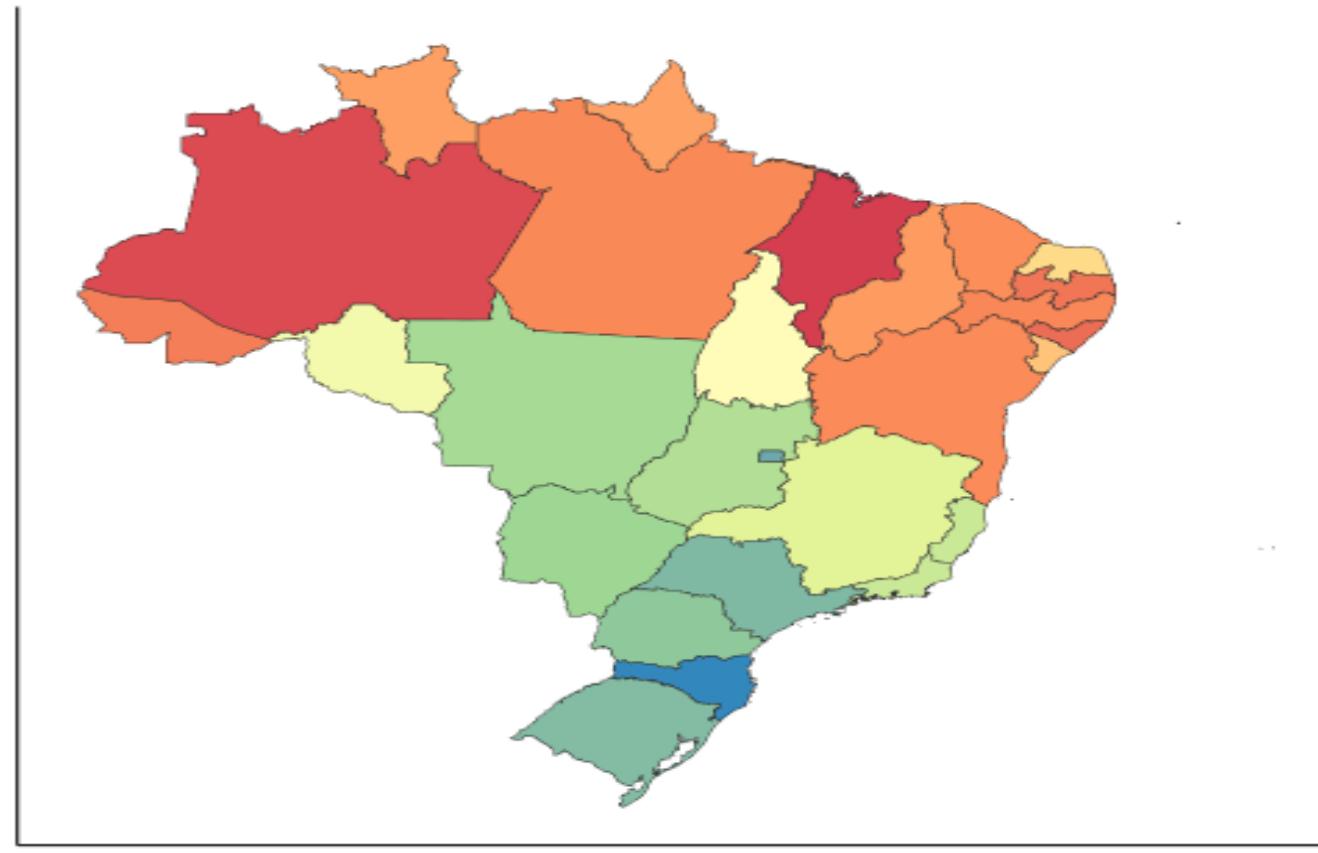


Microdados da PNAD Contínua – visualização espacial

Com a função `convey::svyby()`, podemos calcular a proporção da população sob risco de pobreza em cada unidade da federação. Serão considerados os rendimentos limiares de cada estado para estimar essa parcela da população.

```
svyby(formula = ~VD5005_real, by = ~code_state, design = pnadc_2022_it5,
       FUN = svybypr, na.rm = TRUE) |>
  right_join(estados_br, by = "code_state") |> st_sf() |>
  ggplot() +
  geom_sf(aes(fill = VD5005_real), color = "gray20") +
  scale_fill_distiller(
    palette = "Spectral",
    name = paste0("Proporção da população sob risco\nde pobreza em cada estado."),
    labels = scales::percent
  ) +
  labs(caption = "Fonte: PNADC do IBGE, 2022.") + my_theme2
```

Microdados da PNAD Contínua – visualização espacial



Proporção da população sob risco
de pobreza em cada estado.

20% 30% 40% 50%

Fonte: PNADC do IBGE, 2022.

Microdados da PNAD Contínua – visualização espacial

Podemos estimar a proporção da população em risco de pobreza por unidade da federação. No entanto, o limiar de rendimento será um valor único, calculado em nível nacional, conforme:

```
escala3 <- scale_fill_distiller(palette = "Spectral", name = "Total",
                                 labels = scales::label_number(suffix = " M", scale = 1e-6))
escala4 <- scale_fill_distiller(palette = "Spectral", name = "Proporção",
                                 labels = scales::percent)
risco_pobreza_br <- svyarp(~VD5005_real, pnadc_2022_it5, na.rm = TRUE)
populacao <- pnadc_2022_it5$variables |> group_by(code_state) |>
  count(sob_pobreza = VD5005_real <= risco_pobreza_br, wt = V1032) |>
  mutate(proporcao = n/sum(n)) |> filter(sob_pobreza == TRUE) |>
  right_join(estados_br, by = "code_state") |> st_sf()
p1 <- ggplot(data = populacao) + geom_sf(aes(fill = n), color = "black")
p2 <- ggplot(data = populacao) + geom_sf(aes(fill = proporcao), color = "black")
(p1 + escala3 + my_theme2) + (p2 + escala4 + my_theme2) +
  plot_annotation(
    title = paste0("População em risco de pobreza, cujo rendimento médio mensal real ",
                  "domiciliar\\nper capita esta abaixo de ", round(risco_pobreza_br, digits = 2),
                  " reais."), caption = "Fonte: PNADC do IBGE, 2022.")
```

Microdados da PNAD Contínua – visualização espacial

População em risco de pobreza, cujo rendimento médio mensal real domiciliar per capita está abaixo de 624,62 reais.



Total

2 M 4 M 6 M



Proporção

20% 30% 40% 50%

Fonte: PNADC do IBGE, 2022.

Fontes Adicionais de Dados de Interesse Social

Existem outros pacotes no R que fornecem acesso simplificado a dados como:

- Pacote **censoBR** – dados de censos demográficos.
- Pacote **electionsBR** – dados do TSE sobre eleições no Brasil.
- Pacote **microdadosSUS** – dados do Sistema Único de Saúde do Brasil (SUS), fornecidos pelo departamento de informática do SUS – o DATASUS. Obs: Este pacote não está disponível no CRAM, sendo necessário instalá-lo através de repositório no GitHub.

Dados Domésticos do Censo IBGE – visualização espacial

O pacote **censobr** (documentação [online](#)) permite baixar dados de censos demográficos realizados no Brasil. Vamos ver alguns exemplos de obtenção e visualização desses dados. Primeiro, instalamos e carregamos o pacote:

```
install.packages("censobr"); library(censobr);
```

Focaremos em dados domésticos de residências, que são baixados com **censobr::read_households()**:

```
domestico_br <- read_households(year = 2010, add_labels = 'pt', showProgress = FALSE)
```

O objeto **domestico_br** ainda não possui os dados de domicílios. Como o pacote **censobr** foi implementado considerando o pacote **arrow** (provém armazenamento, manipulação e acesso remoto de dados), o objeto **domestico_br** permite baixar dados conforme à demanda do usuário – questão importante devido às possíveis limitações computacionais (principalmente memória RAM) e a grande quantidade de dados censitários.

```
class(domestico_br)  
## [1] "arrow_dplyr_query"
```

Dados Domésticos do Censo IBGE – visualização espacial

Já que podemos baixar variáveis específicas, devemos conhecer os dados. Para fazer consultas, basta usar a função `censobr::data_dictionary()`.

```
data_dictionary(2010, 'households')
```

Vamos obter dados sobre o tipo de esgotamento sanitário (variável **V0207**) disponível nos domicílios, calculando o total e a proporção de cada tipo nos municípios. É necessário considerar o peso amostral (variável **V0010**). Com a função `dplyr::collect()`, coletamos nossos dados, que são fornecidos pelo objeto `domestico_br`, observe:

```
esgotamento_sanitario <- domestico_br |>  
  collect() |>  
  group_by(code_muni) |>  
  count(tipo = V0207, wt = V0010) |>  
  mutate(proporcao = n / sum(n)) |>  
  collect()
```

Dados Domésticos do Censo IBGE – visualização espacial

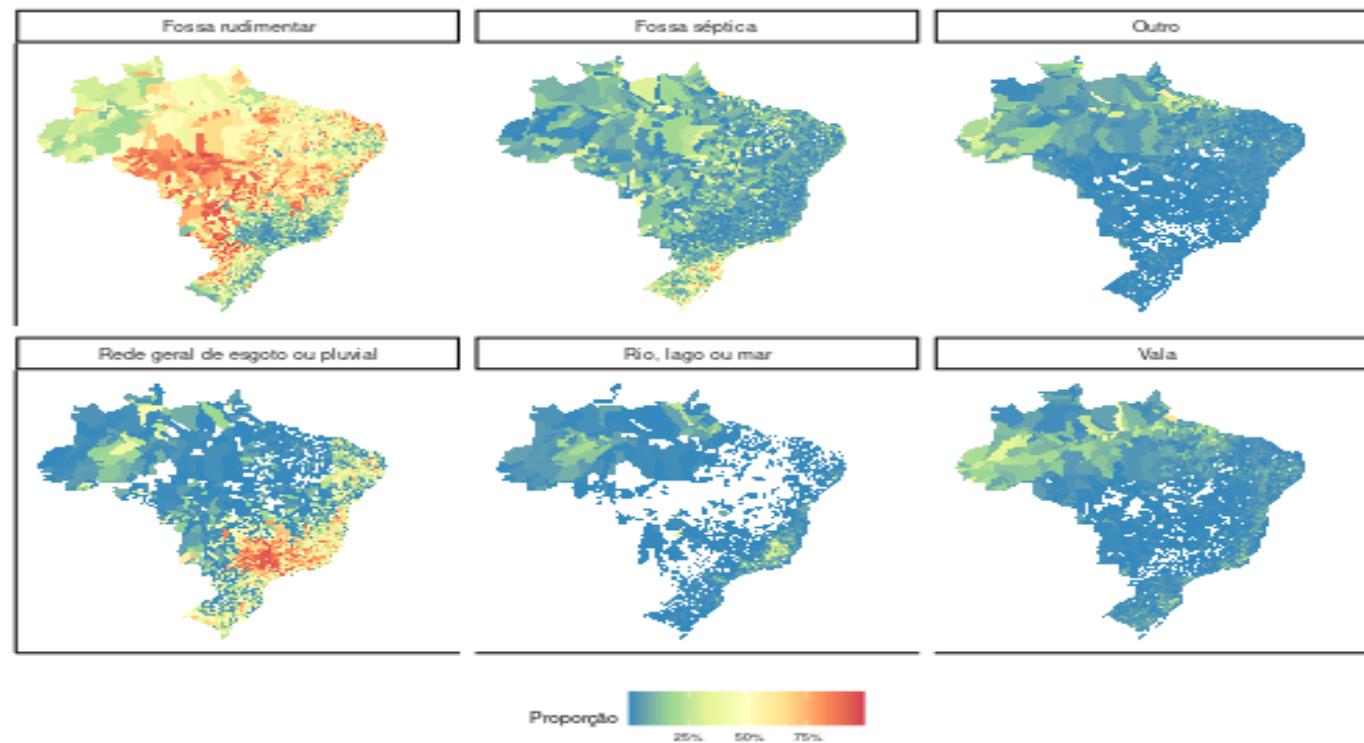
Também vamos obter o mapa do Brasil com a divisão territorial dos municípios:

```
municipios_br <- geobr::read_municipality(year = 2010, showProgress = FALSE) |>  
  mutate(code_muni = as.character(code_muni))
```

Agora, produziremos mapas dos dados, conforme:

```
left_join(esgotamento_sanitario, municipios_br, by = "code_muni") |>  
  st_sf() |>  
  drop_na() |>  
  ggplot(aes(fill = proporcao)) +  
    geom_sf(linewidth = 0) +  
    facet_wrap(~tipo) +  
    escala4 + my_theme
```

Dados Domésticos do Censo IBGE – visualização espacial



Dados Domésticos do Censo IBGE – visualização espacial

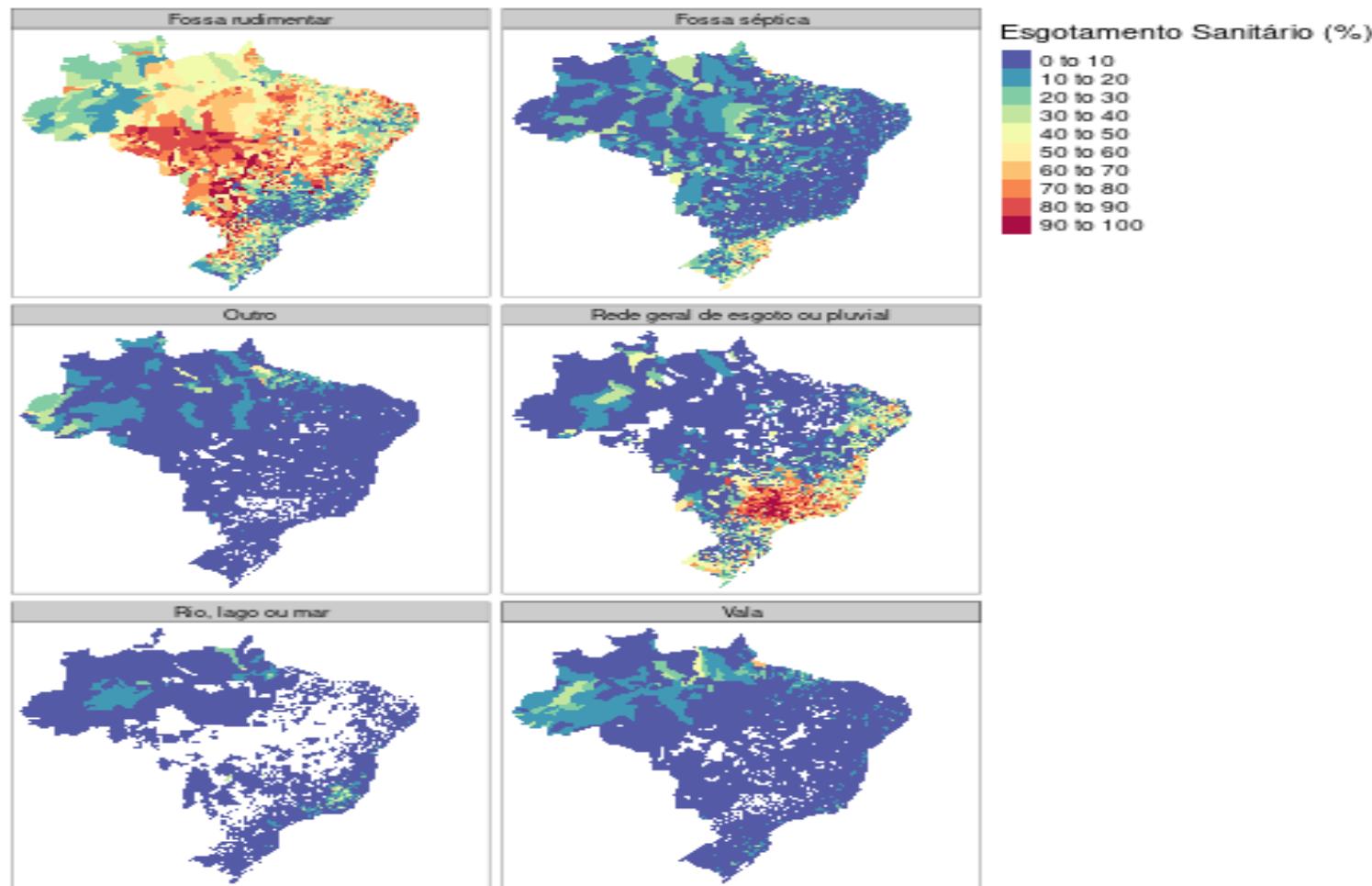
Podemos gerar mapas com outros pacotes gráficos, como o **tmap**.

```
install.packages(c("lwgeom", "tmap"))
library(tmap)
```

Vejamos um exemplo de mapa com os mesmos dados de esgotamento sanitário:

```
esgotamento_sanitario |>
  mutate(proporcao = proporcao * 100) |>
  right_join(municipios_br, by = "code_muni") |>
  st_sf() |> drop_na() |>
  tm_shape() +
  tmap_options(check.and.fix = TRUE) +
  tm_fill("proporcao",
          n = 10,
          palette = '-Spectral',
          title = "Esgotamento Sanitário (%)") +
  tm_facets(by = "tipo")
```

Dados Domésticos do Censo IBGE – visualização espacial

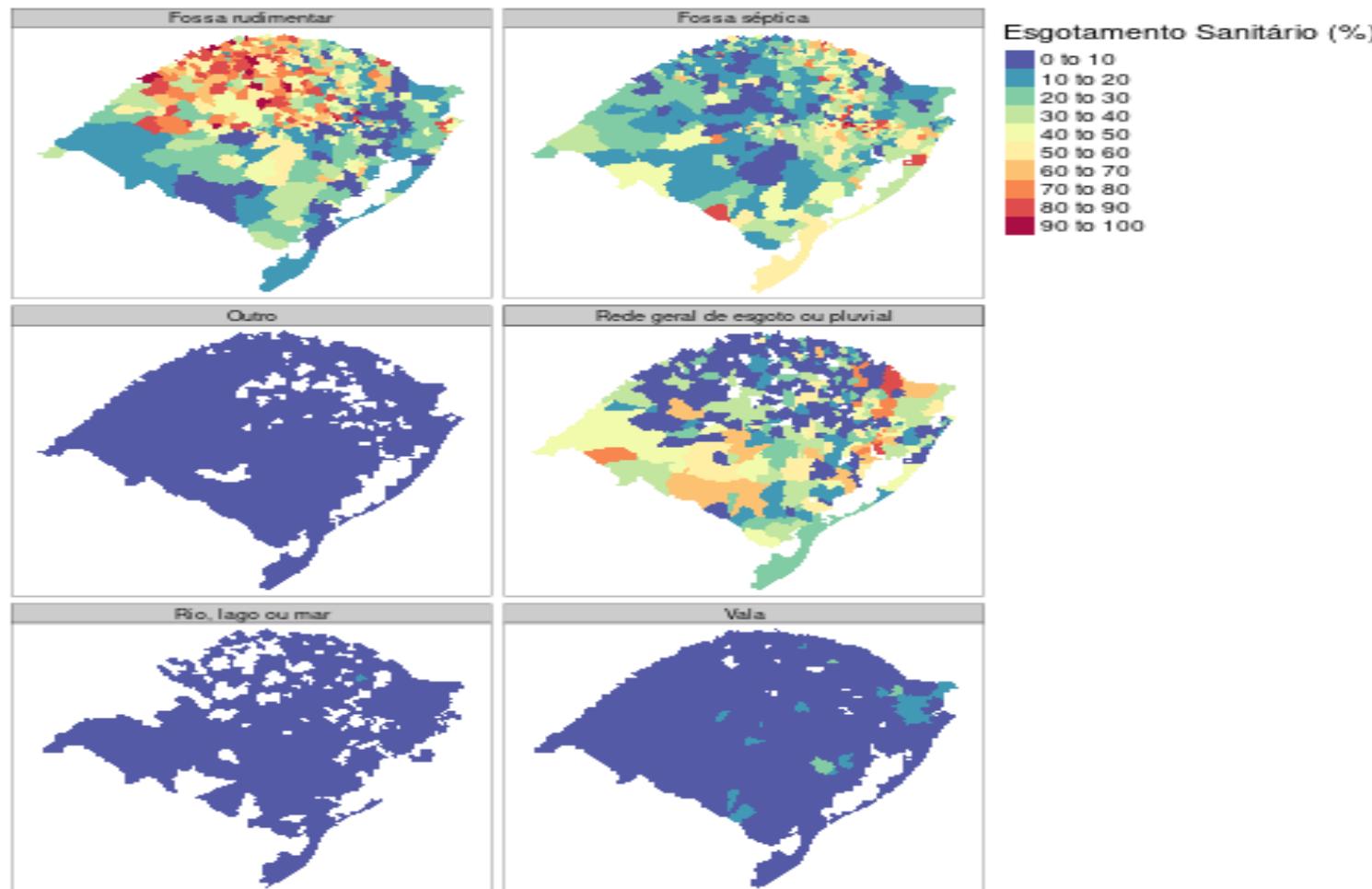


Dados Domésticos do Censo IBGE – visualização espacial

Somente para o estado do Rio grande do sul?

```
esgotamento_sanitario |>
  mutate(proporcao = proporcao * 100) |>
  right_join(y = filter(municipios_br, code_state == 43), by = "code_muni") |>
  st_sf() |>
  drop_na() |>
  tm_shape() +
  tmap_options(check.and.fix = TRUE) +
  tm_fill("proporcao",
          n = 10,
          palette = '-Spectral',
          title = "Esgotamento Sanitário (%)") +
  tm_facets(by = "tipo")
```

Dados Domésticos do Censo IBGE – visualização espacial

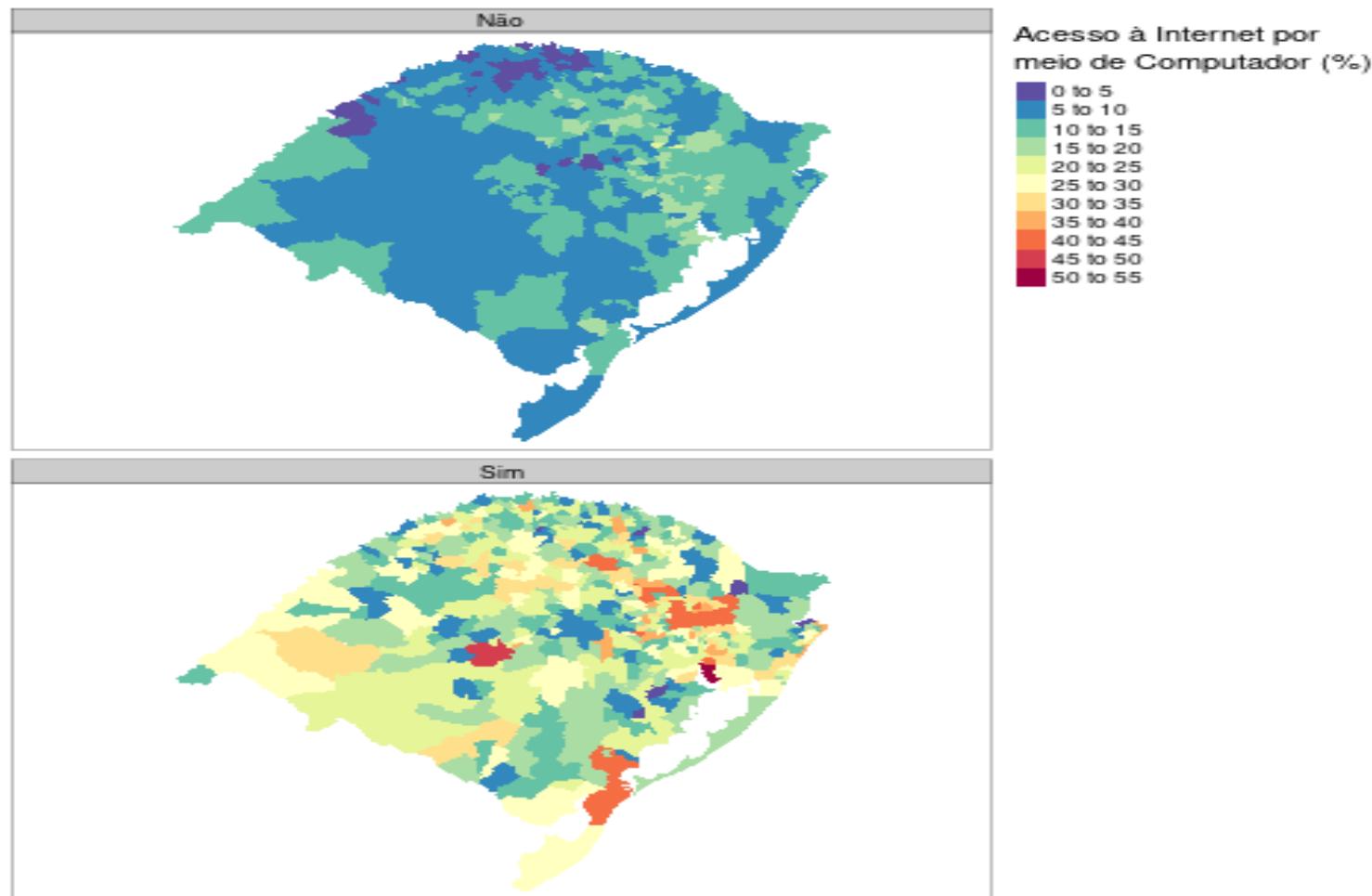


Dados Domésticos do Censo IBGE – visualização espacial

Computador com acesso à internet (variável V0220), no estado do Rio grande do sul?

```
domestico_br |>
  collect() |>
  group_by(code_muni) |>
  count(acesso_internet = V0220, wt = V0010) |>
  mutate(proporcao = n / sum(n) * 100) |>
  collect() |>
  right_join(y = filter(municipios_br, code_state == 43), by = "code_muni") |>
  st_sf() |>
  drop_na() |>
  tm_shape() +
  tmap_options(check.and.fix = TRUE) +
  tm_fill("proporcao",
          n = 10,
          palette = '-Spectral',
          title = "Acesso à Internet por\nmeio de Computador (%)") +
  tm_facets(by = "acesso_internet")
```

Dados Domésticos do Censo IBGE – visualização espacial



Visualização Espacial Interativa

Existem alternativas no R para visualizar mapas interativos e dinâmicos. Para simplificar, podemos continuar usando o pacote **tmap**, que fornece visualizações interativas. Veremos um breve exemplo por meio deste [tutorial](#).

Bibliografia Complementar

Há uma diversidade de bons materiais que orientam o uso dos microdados da PNAD Contínua através do R:

- Documentação **online** desenvolvido por um dos criadores do pacote **PNADclIBGE**.
- Livro "**POR DENTRO DA PNAD CONTÍNUA: Uma introdução ao tratamento de dados usando o R**".
- Curso de R disponível no repositório do Git Hub: **Introdução à Programação em R**.
- Materiais online sobre o pacote **srvyr**: [Link 1](#) e [Link 2](#).

Documentações para uso dos pacotes: **convey**, **geobr**, **censobr**, **electionsBR**, **microdatasus** e **tmap**.

Livro sobre o pacote **tmap** e **tutorial** mais curto.

Livro com conteúdo mais avançado sobre manipulação e visualização de geodados: **Geocomputation with R**.

Metodologias Informacionais com 

Muito Obrigado pela Atenção!