

Uma rápida introdução ao **geobr** (R)*

Gabrielito Menezes[†]

2023

1 Introdução ao **geobr**

O pacote **geobr** configura-se como uma solução acessível e prática para o acesso a conjuntos de dados espaciais oficiais do território brasileiro. Sua sintaxe simplificada possibilita aos usuários a rápida e fácil obtenção de uma ampla variedade de conjuntos de dados, com geometrias atualizadas e atributos harmonizados, bem como projeções geográficas em diferentes regiões e anos. Tal solução traz benefícios notáveis para a análise de informações espaciais, ao permitir a obtenção de dados confiáveis e atualizados para a realização de pesquisas e trabalhos em áreas como economia, geografia, dentre outras. Neste sentido, a presente material apresenta uma breve e concisa introdução ao pacote **geobr**, fornecendo ao leitor as informações necessárias para explorar suas funcionalidades e recursos com facilidade, eficiência e precisão¹.

1.1 Instalação do pacote **geobr**

Para dar início à utilização do pacote **geobr**, é necessário realizar sua instalação. Essa operação pode ser realizada de forma simples e direta, a partir da utilização da função `install.packages()`, como exemplificado no **chunck** apresentado abaixo. Esse procedimento é fundamental para garantir a disponibilidade das funcionalidades do pacote e, conseqüentemente, para a realização de análises e trabalhos que envolvam dados espaciais no território brasileiro.

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

Com a execução do código acima, o pacote **geobr** será instalado e estará pronto para ser utilizado em seus projetos e análises. É importante ressaltar que essa etapa deve ser realizada previamente à utilização de qualquer funcionalidade do pacote.

2 Próximos passos

Vamos precisar carregar outros pacotes do R, para utilizarmos junto com o **geobr**.

```
library(geobr)
library(sf)
library(magrittr)
library(tidyverse)
```

*Versão preliminar, sugestões serão muito bem-vindas.

[†]Professor Adjunto da Universidade Federal de Pelotas (UFPel). E-mail: gabrielitorm@gmail.com

¹Essa introdução ao **geobr** foi realizada com a versão do R 4.0.5 e a versão do RStudio foi 2022.07.2+576 “Spotted Wakerobin” for Windows Mozilla/5.0 (Windows NT 10.0; Win64; x64).

3 Mapas com o geobr

Para prosseguir em sua utilização do pacote `geobr`, é possível acessar as informações relacionadas a suas funcionalidades e recursos disponíveis. Além disso, é possível acessar as *vignettes* do pacote, que apresentam exemplos práticos de utilização de suas funcionalidades e recursos.

```
conj_dados <- list_geobr()
```

```
conj_dados
```

```
## # A tibble: 27 x 4
##   `function`      geography      years source
##   <chr>          <chr>      <chr> <chr>
## 1 `read_country`  Country    1872~ IBGE
## 2 `read_region`  Region     2000~ IBGE
## 3 `read_state`   States     1872~ IBGE
## 4 `read_meso_region` Meso region 2000~ IBGE
## 5 `read_micro_region` Micro region 2000~ IBGE
## 6 `read_intermediate_region` Intermediate region 2017~ IBGE
## 7 `read_immediate_region` Immediate region 2017~ IBGE
## 8 `read_municipality` Municipality 1872~ IBGE
## 9 `read_municipal_seat` Municipality seats (sedes municipais) 1872~ IBGE
## 10 `read_weighting_area` Census weighting area (área de ponde~ 2010 IBGE
## # ... with 17 more rows
```

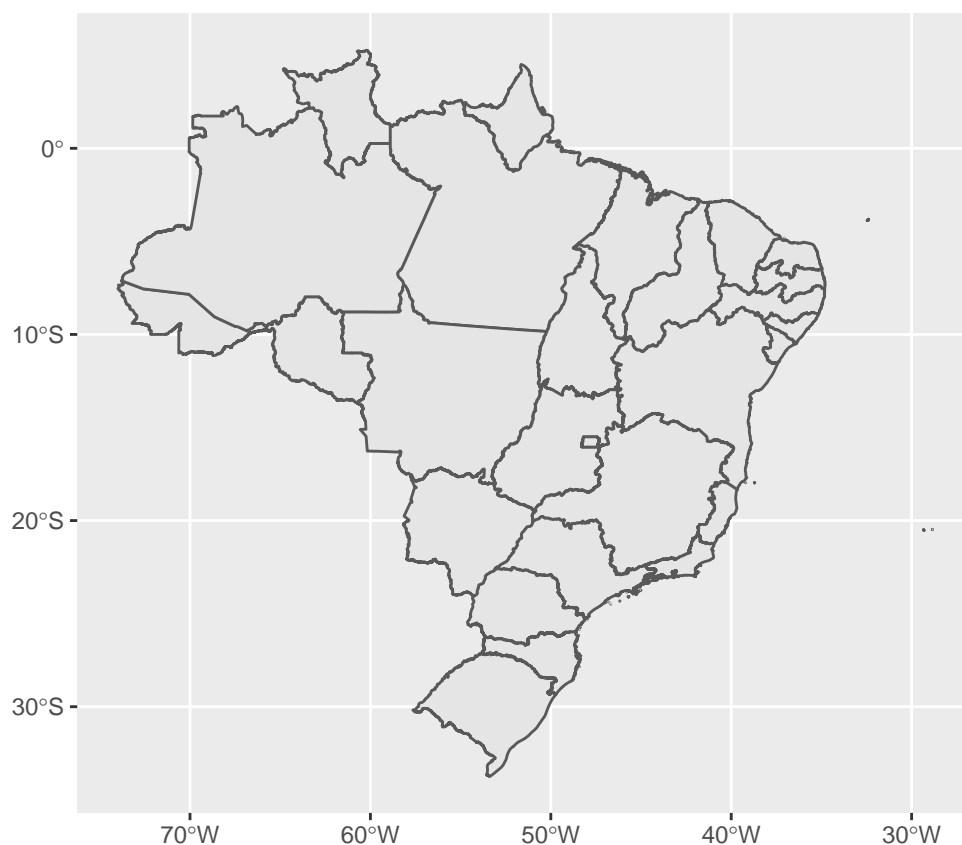
O pacote `geobr` tem as rotinas para o *download* dos mapas com divisões territoriais variadas do Brasil como um todo. Vamos fazer um exemplo para o Brasil e outro para o Estado do Rio Grande do Sul (RS).

3.1 Mapa para os Estados do Brasil

Para realizar o *download* do mapa do Brasil por estados, é necessário utilizar uma função específica que permita acessar a estrutura estadual do país. Nesse sentido, a função `read_state()` é uma alternativa que possibilita a obtenção dos dados referentes aos estados brasileiros para o ano de 2018.

```
brasil <- read_state(code_state = "all", year = 2018)
```

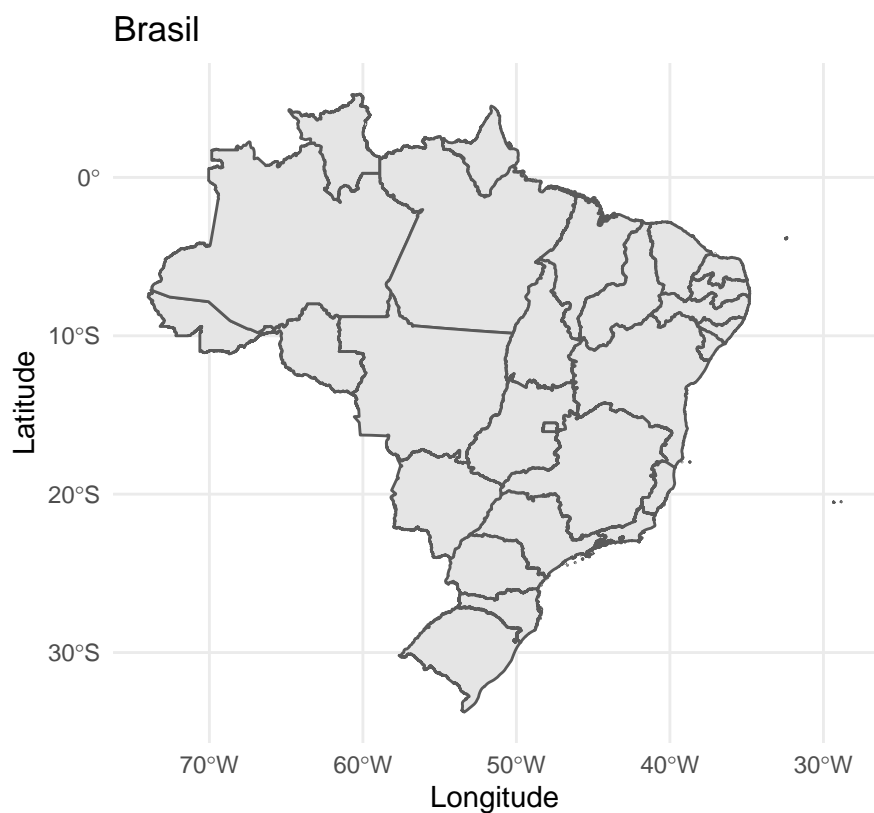
```
ggplot() + geom_sf(data = brasil)
```



Podemos aprimorar a aparência do nosso mapa ao utilizar o pacote `ggplot2` juntamente com alguns temas personalizados. Esse pacote é amplamente utilizada na criação de gráficos e visualizações de dados em R, permitindo a customização de diversos aspectos visuais, como cores, fontes e estilos.

Ao adicionar temas personalizados ao nosso mapa, podemos torná-lo mais elegante e profissional, transmitindo informações de forma clara e objetiva. Além disso, essa personalização pode ajudar a destacar aspectos importantes do mapa, como regiões com maior densidade populacional, indicadores socioeconômicos ou outras informações relevantes.

```
ggplot() + geom_sf(data = brasil) + theme_minimal() +  
  ggtitle("Brasil") + labs(caption = "Fonte: IBGE (2018)") +  
  labs(x="Longitude", y="Latitude")
```



Fonte: IBGE (2018)

Então, o que você achou?

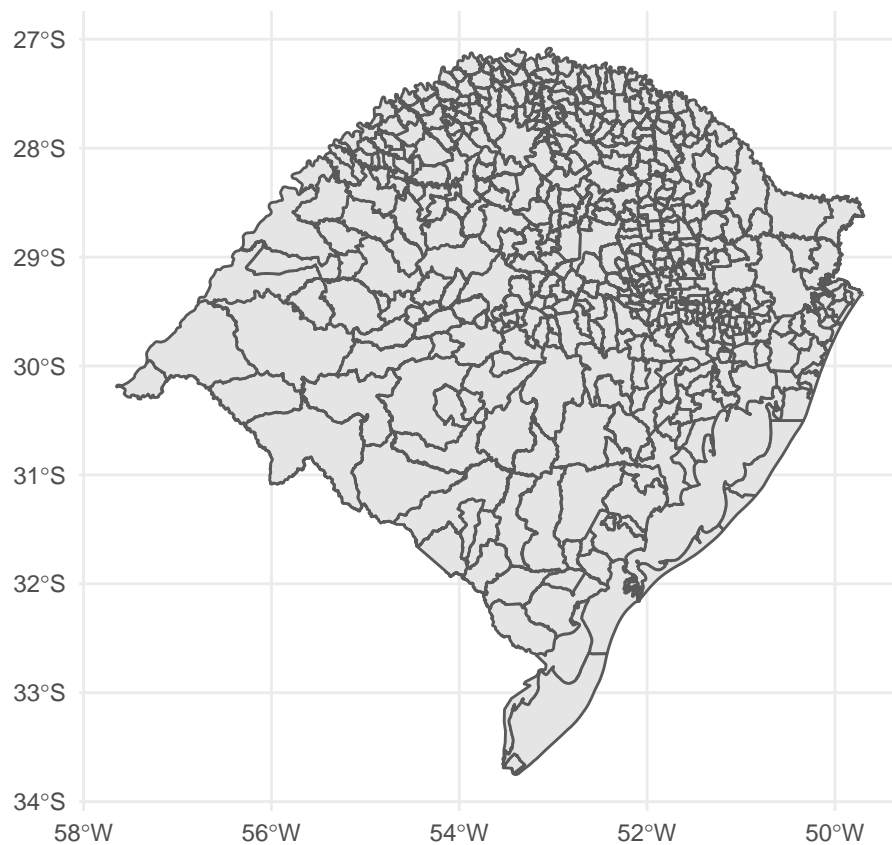
3.2 Mapa para os municípios do Estado do RS

Para fazer o *download* do mapa do Rio Grande do Sul, é possível utilizar a função `rea_municipality()`, que permite o acesso aos dados referentes aos municípios do estado, como está no *chunk* abaixo. Com a obtenção dos dados referentes aos municípios do Rio Grande do Sul, é possível criar visualizações e gráficos que permitem a compreensão das características e peculiaridades de cada região, bem como a identificação da distribuição espacial da variável que pode ser projetada no mapa.

```
municipios <- read_municipality(code_muni = 43, year = 2018)
```

De posse dos nossos dados, vamos fazer o plot do mapa do RS! O que você está observando?

```
ggplot() + geom_sf(data=municipios) + theme_minimal()
```



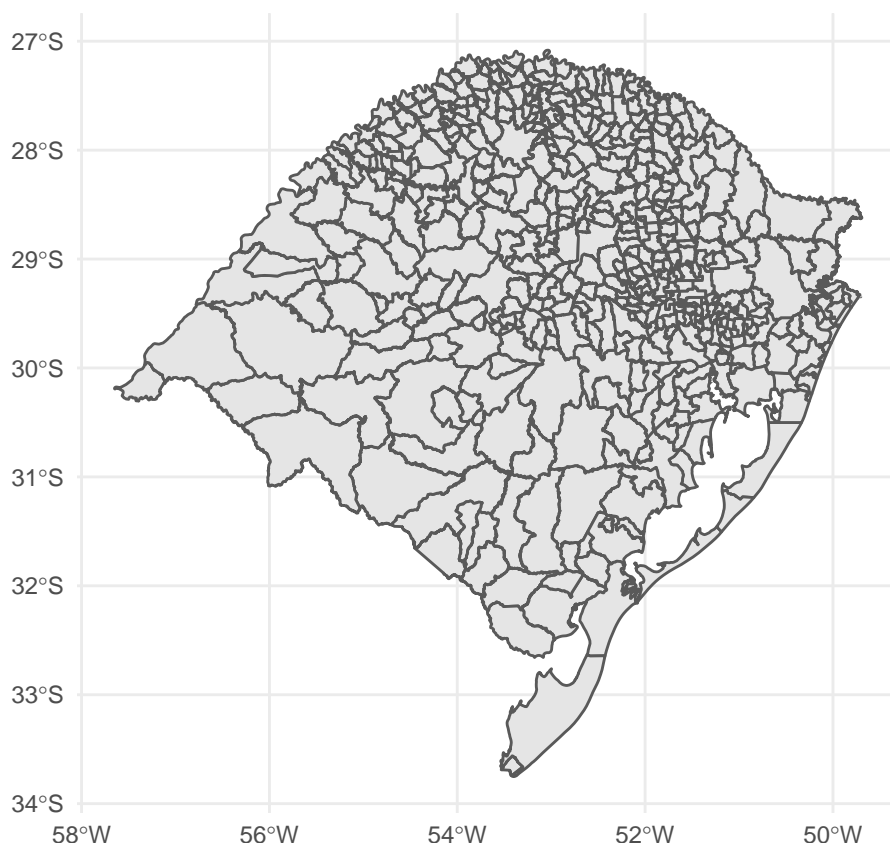
Vamos ter que retirar a Lagoa Mirim e a Lado dos Patos do nosso shapefile²! Temos duas opções! Uma é fazer um `subset` e a outra retirar os dados da nossa amostra.

```
rs_01 <- municipios[3:499,]
```

Agora sim:

```
ggplot() + geom_sf(data = rs_01) + theme_minimal()
```

²O formato consiste em pelo menos quatro arquivos: `.shp` (feição), `.dbf` (tabela de atributos), `.shx` (ligação entre `.shp` e `.dbf`) e `.prj` (projeção).



4 Unindo os datasets

Para realizar a integração da base de dados com o shapefile do Rio Grande do Sul, é necessário utilizar o pacote `readxl`, que permite a leitura de arquivos em formato Excel. Ao integrar a base de dados com o shapefile, é possível criar visualizações e mapas que relacionem as informações contidas na base de dados com a distribuição geográfica das diferentes regiões do Rio Grande do Sul.

```
library(readxl)
```

```
dados.rs <- read_excel("BaseDadosRS2.xlsx")
```

A função `head()` é uma ferramenta útil para verificar se a importação dos dados foi realizada com sucesso. Essa função permite visualizar os primeiros registros da base de dados, permitindo a identificação de possíveis erros ou inconsistências na importação. Vou deixar para você fazer isso quando estiver replicando esse *script*.

```
head(dados.rs)
```

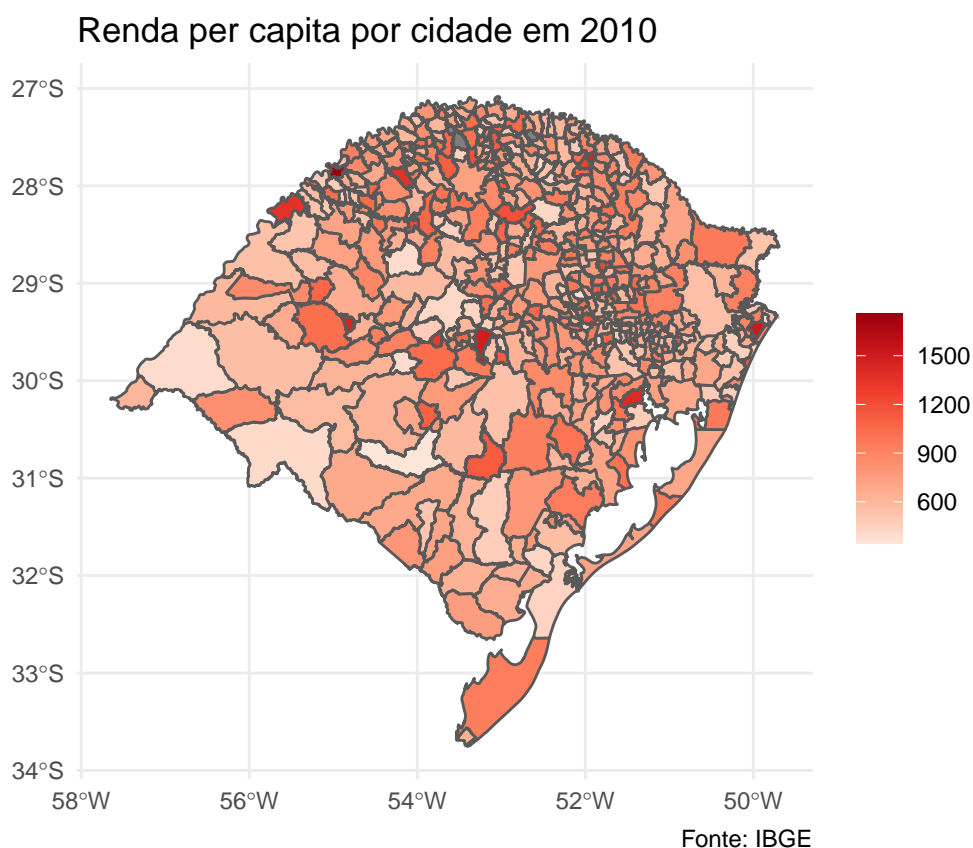
Importante destacar que para fazer o *join*, precisamos ter a planilha com uma coluna com os códigos dos municípios igual a que está na coluna do shapefile `code_muni`. O bom de fazer o *join* usando o código dos municípios é que ocorre menos erros de digitação, quando comparado pelos nomes dos municípios. Então, vamos lá:

```
rs.dados <- rs_01 %>% left_join(dados.rs, by = "code_muni")
```

5 Fazendo o plot da renda *per capita* do RS

Após realizar a união dos dados necessários para a análise, é possível prosseguir com a criação de um mapa que represente a distribuição da *renda per capita* no Rio Grande do Sul no ano de 2010. Essa etapa é fundamental para a visualização dos resultados da análise e para a identificação de possíveis padrões espaciais presentes nos dados. Para a criação do mapa, pode-se utilizar o pacote `ggplot2`, que disponibiliza uma série de ferramentas para a elaboração de gráficos e mapas em R.

```
ggplot(rs.dados) +  
  geom_sf(aes(fill = `Renda per capita 2010`))) +  
    scale_fill_distiller(type = "seq",  
                        palette = "Reds",  
                        direction = 1) +  
  
  theme_minimal() +  
  labs(title = "Renda per capita por cidade em 2010",  
       fill = NULL, caption = "Fonte: IBGE")
```



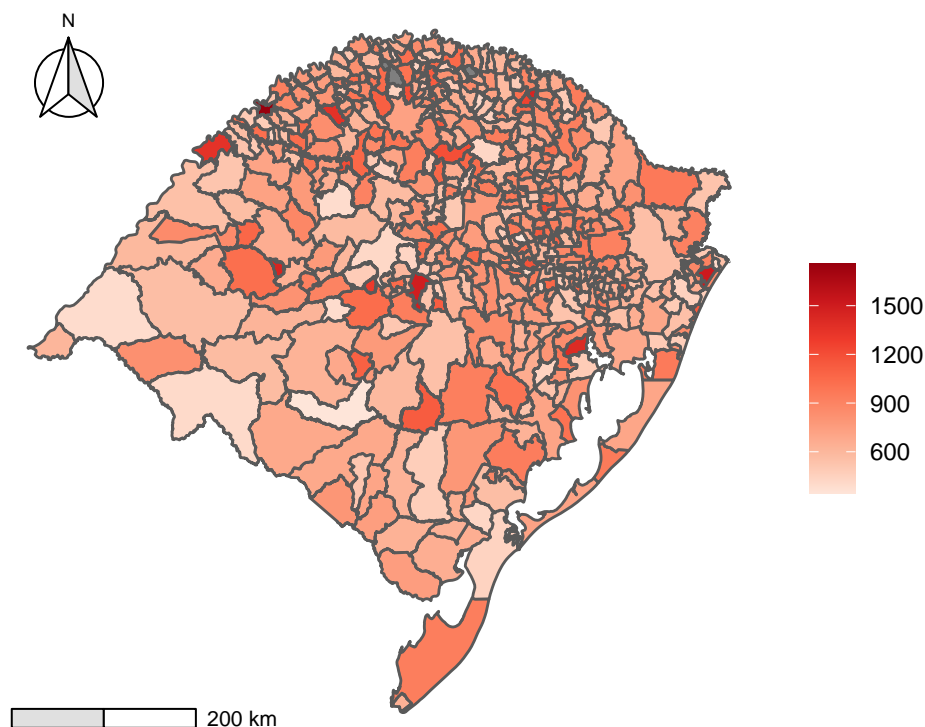
Com a execução do `chunk` acima, foi criado um mapa que representa a distribuição da renda *per capita* no Rio Grande do Sul no ano de 2010. Esse mapa é criado a partir da utilização da função `geom_sf()`, que permite a criação de mapas a partir de objetos `sf`. Além disso, foram utilizados outros parâmetros para personalização do mapa, como a escala de cores e o tema utilizado na visualização.

Dando continuidade à elaboração de mapas, é possível explorar o potencial do pacote `ggspatial` para incluir elementos de orientação e escala nos mapas produzidos. A adoção deste pacote possibilita uma maior precisão e clareza visual em relação à apresentação de informações geográficas, além de possibilitar uma melhor compreensão do território e dos fenômenos espaciais em estudo.

```
library(ggspatial)

ggplot(rs.dados) +
  geom_sf(aes(fill = (`Renda per capita 2010`))) +
  scale_fill_distiller(type = "seq",
    palette = "Reds",
    direction = 1) +
  theme_minimal() +
  labs(title = "Renda per capita por cidade em 2010",
    fill = NULL, caption = "Fonte: IBGE") +
  annotation_scale(bar_cols = c("gray88", "white")) +
  coord_sf(datum = NA) +
  annotation_north_arrow(location = 'tl',
    style = north_arrow_fancy_orienteering(text_size = 7,
    fill = c("white", "gray88")))
```

Renda per capita por cidade em 2010



Fonte: IBGE

A habilidade de elaborar mapas eficientes e precisos, juntamente com a capacidade de manipular bases de dados em formato shapefile no ambiente do software R, representa um avanço significativo na análise de informações espaciais. Para ampliar ainda mais as possibilidades de exploração do potencial deste ambiente computacional, é recomendável a exploração do material disponível nas funções *help* e **vignettes** do pacote **geobr** e dos demais pacotes utilizados neste material. Ao se aprofundar nestes recursos, é possível acessar informações valiosas sobre o manejo dos pacotes citados, ampliando a compreensão das suas funcionalidades e recursos. Deste modo, torna-se possível obter uma performance ainda mais eficaz na elaboração de mapas e na análise de informações geográficas, conferindo uma maior precisão e clareza nas informações apresentadas.

6 Referências

- Dewey Dunnington (2022). ggspatial: Spatial Data Framework for ggplot2. R package version 1.1.7. <https://CRAN.R-project.org/package=ggspatial>
- Rafael H. M. Pereira and Caio Nogueira Goncalves (2021). geobr: Download Official Spatial Data Sets of Brazil. R package version 1.6.1. <https://CRAN.R-project.org/package=geobr>
- Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal 10 (1), 439-446, <https://doi.org/10.32614/RJ-2018-009>
- Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, <https://doi.org/10.21105/joss.01686>
- Opções de Chunk. <https://cdr.ibpad.com.br/rmarkdown.html#op%C3%A7%C3%B5es-de-chunk>