

Aula 03 – Mapas Temáticos – Parte 2

Sumário

Aula 03 – Mapas Temáticos – Parte 2	1
1. Apresentação	2
2. Pacotes mais utilizados.....	3
3. Preparando dados de letalidade por SRAG por Covid-19	4
4. Baixando e importando arquivos <i>shapefile</i> do Brasil.....	4
5. Centroides por polígono geográfico.....	10
6. Mapas compostos: incluindo informações nos polígonos e centroides	12
7. Referências Bibliográficas	13
8. Exercícios	15

1. Apresentação

Olá! Seja bem-vindo ao terceiro capítulo do curso de “R aplicado a Análises Espaciais em Saúde”!

Nosso objetivo é promover o uso do R, uma linguagem de programação de alto nível, para análises espaciais em Saúde. Continuamos usando a interface de desenvolvimento do R, RStudio. Isso facilita a escrita dos nossos *scripts* e a visualização de *datasets* e mapas.

Em nosso segundo capítulo fomos iniciados aos mapas temáticos. Apresentamos pacotes importantes e conseguimos plotar nosso primeiro mapa temático. Hoje, avançaremos um pouco mais, plotando mapas temáticos e mapas de centroides proporcionais.

Antes de iniciar, você deve ter baixado o arquivo compactado “curso_r_geoanalise”. Após descompactar o arquivo, clique no projeto R com o nome “curso_r_geonanalise.Rproject”. Esse é o arquivo projeto do nosso curso, você deverá abri-lo e depois selecionar o *script* “03_mapas_tematicos_2.R”.

Abrindo pelo Rproject, você fica com todas as pastas alinhadas e terá acesso a todos os arquivos que usaremos nas aulas sem dificuldade. Embarque conosco mais uma vez e vamos continuar nessa jornada! Que seu aprendizado seja muito útil no SUS!

2. Pacotes mais utilizados

Na aula anterior, falamos de alguns pacotes que utilizamos para a criação de mapas, com ênfase nos pacotes que nos permitem criar mapas temáticos simples. Nesta aula, falaremos sobre mapas temáticos e mapas de centroides, que servem para comparar a magnitude de dados de um território com o outro.

Nós vamos usar os mesmos pacotes que utilizamos em aulas anteriores (1–18), mas vamos dar ênfase nos pacotes que são úteis para a criação dos mapas temáticos e sua combinação com mapas de centroide. Também incluímos o pacote *sf*, que nos ajudará a importar arquivos *shapefile* que baixamos da internet. Os mapas de centroide geram um círculo no centro do polígono que representa um determinado território e nos permite dimensionar seu tamanho conforme a proporcionalidade do indicador analisado.

Para a aula de hoje, vamos destacar três pacotes muito importantes:

- *ggplot2*: é o pacote mais popular do R. Esse pacote fornece funções que permitem programar o R como plataforma gráfica. É um pacote muito robusto e possui uma infinidade de gráficos que podem ser usados em diversas análises estatística. O pacote tem origem no livro “*The Grammar of Graphics*”, baseado na tese de doutorado do estatístico Leland Wilkinson (19).
- *ggspatial*: é o pacote do *framework* geoespacial para o *ggplot2*. Com esse pacote, podemos usar a base gráfica e a sintaxe do *ggplot2* para gerar e personalizar mapas.
- *sf*: é o pacote que permite a manipulação de vetores de dados espaciais padronizados. Vamos usar esse pacote para importar um arquivo *shapefile* que baixaremos do repositório *online* do Instituto Brasileiro de Geografia e Estatística.
- *maptools*: é um pacote de ferramentas de manipulação de mapas que permite criarmos visualizações geográficas no R e combinado com os outros pacotes, nos permite manipular os dados associados aos mapas para a customização necessária. Esse pacote será descontinuado no final de 2023 e suas funções poderão ser encontradas no pacote *sp*.

3. Preparando dados de letalidade por SRAG por Covid-19

Inicialmente, vamos importar os dados de casos por UF que preparamos na aula anterior. Vamos importar também o arquivo original do Sivep-Gripe de 2020 para realizarmos a cálculo no número de casos por UF.

```
#PREPARANDO DADOS DE LETALIDADE POR SRAG COVID-19
casos_uf = import("dados/sivep2020_casos.RDS")

sivep2020 = import("dados/sivep_gripe_2020.csv")
sivep2020_obitos = sivep2020 |> #criando um data.frame
  select(EVOLUCAO, SG_UF_NOT) |> #variáveis de interesse
  filter(EVOLUCAO== 2) |>      #apenas dados de Covid-19
  select(SG_UF_NOT) |>         #selecionando apenas dados de UF
  group_by(SG_UF_NOT) |>       #agrupando dados por UF
  summarise(OBITOS = n())      #contando número de óbitos por UF
```

Nós utilizamos algumas funções do pacote *dplyr* que faz parte do pacote *tidyverse* para contar o número de óbitos por UF, semelhante ao que fizemos na aula anterior para contar o número de casos. Nesse caso, com auxílio do dicionário de dados do Sivep-Gripe, nós selecionamos a variável “EVOLUCAO”, que se refere à evolução do caso, sendo que a classificação “2” representa o óbito. Usamos a função “*filter()*” para selecionar apenas as notificações de quem evoluiu a óbito por UF.

Após isso, vamos calcular as taxas de letalidade e incidência por UF. Vamos unir o *dataframe* de óbitos com o de casos, para que possamos calcular a taxa de letalidade, que é o percentual de óbitos em relação ao número de casos. Para calcular a taxa de incidência, importaremos novamente os dados de população por UF.

```
##CALCULANDO TAXA DE LETALIDADE
letal_uf = casos_uf |>
  left_join(sivep2020_obitos, by = c("SG_UF_NOT" = "SG_UF_NOT"))
letal_uf$TX_LETALIDADE = round(letal_uf$OBITOS/letal_uf$CASOS*100, 2)

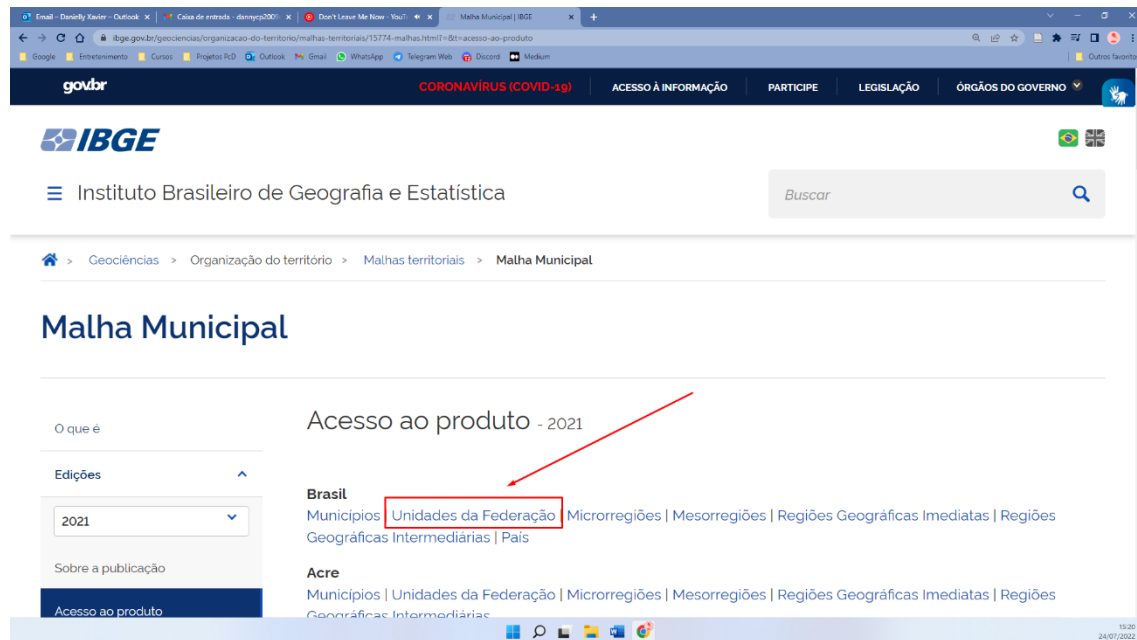
##CALCULANDO TAXA DE INCIDÊNCIA
populacao_uf = import("dados/populacao_uf_2020.csv")
letal_uf = letal_uf |>
  left_join(populacao_uf, by = c("SG_UF_NOT" = "sigla"))
letal_uf$TX_INCIDENCIA = round(letal_uf$CASOS/letal_uf$pop*100000, 2)
```

4. Baixando e importando arquivos *shapefile* do Brasil

No primeiro capítulo desse módulo, mostramos o caminho para baixar arquivos *shapefile* oficiais do Brasil, pelo repositório *online* do Instituto Brasileiro de Geografia e Bioestatística (IBGE). Esse repositório possui arquivos oficiais e configurados de forma correta para a importação em quaisquer *softwares* de

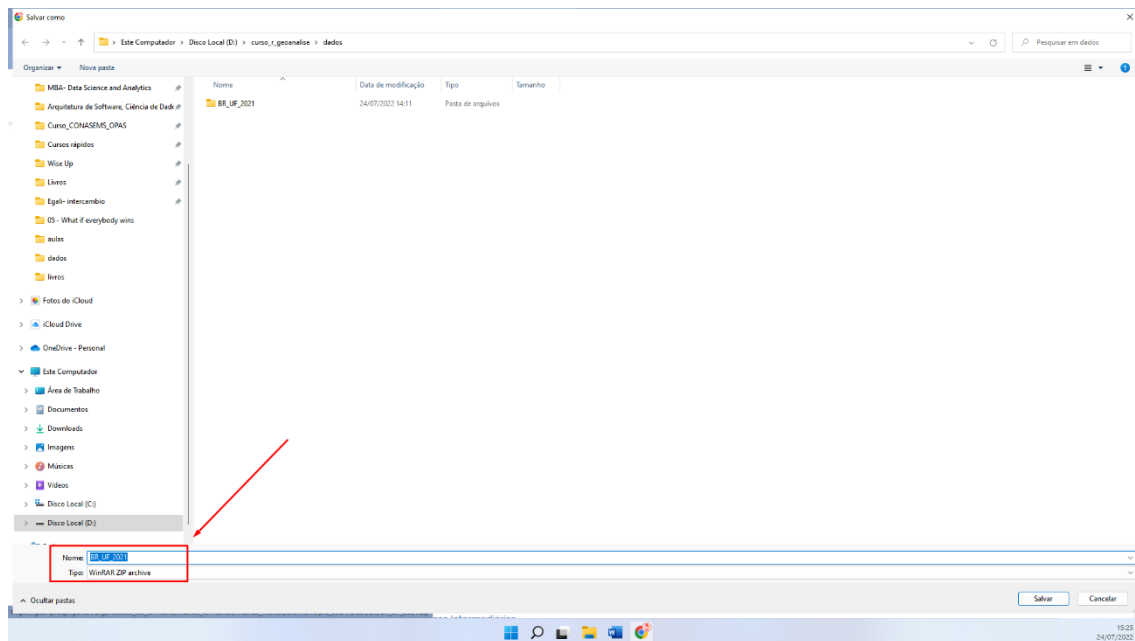
geoprocessamento, incluindo os pacotes do R. Lembre-se, sempre que for importar um mapa no R, ele precisará estar com os arquivos de atributos configurados da forma correta, pois não será possível visualizá-lo se houver alguma incorreção.

Vamos importar a malha territorial do Brasil por UF, por meio do repositório do IBGE, disponível [aqui](#). Ao clicar no link, você será direcionado para a página da imagem abaixo:

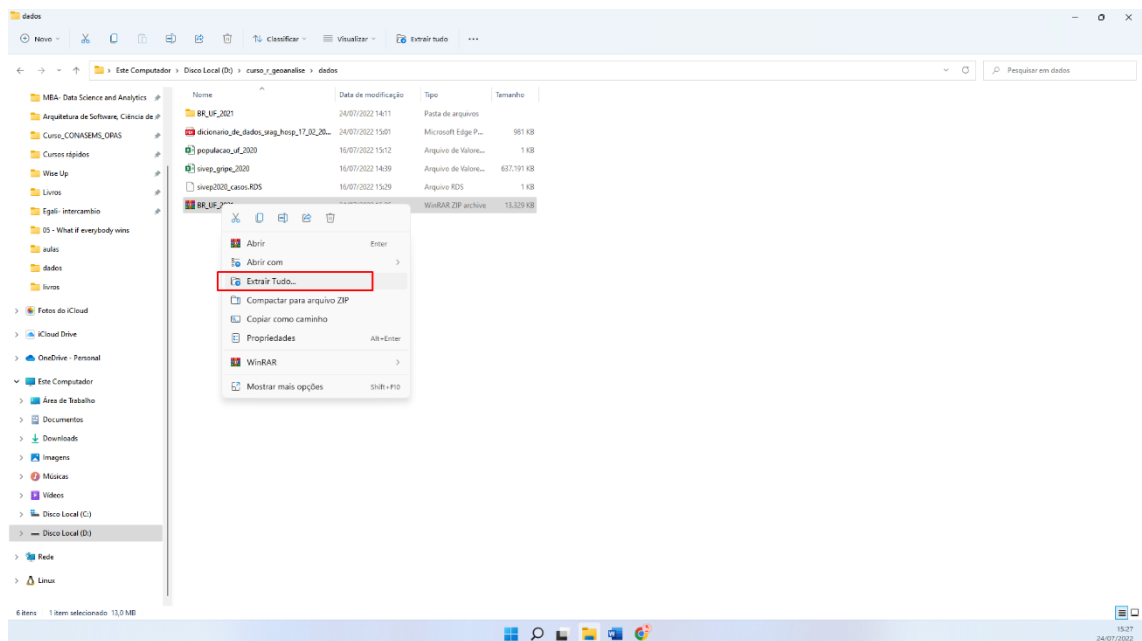


Em “Acesso ao Produto”, vamos clicar em “Unidades da Federação”, no tópico “Brasil”. Esse arquivo é o *shapefile* que possui as divisões geográficas das UF. Note que esse repositório possui arquivos de todas as UF e permite que você baixe qualquer um deles.

Ao clicar no arquivo, o navegador te direcionará para o *download*. Em geral, o arquivo “BR_UF_2021.zip” será salvo automaticamente em sua pasta “Downloads” (considerando usuários de Windows), ou você poderá escolher a pasta em que o arquivo ficará salvo, conforme a próxima imagem.

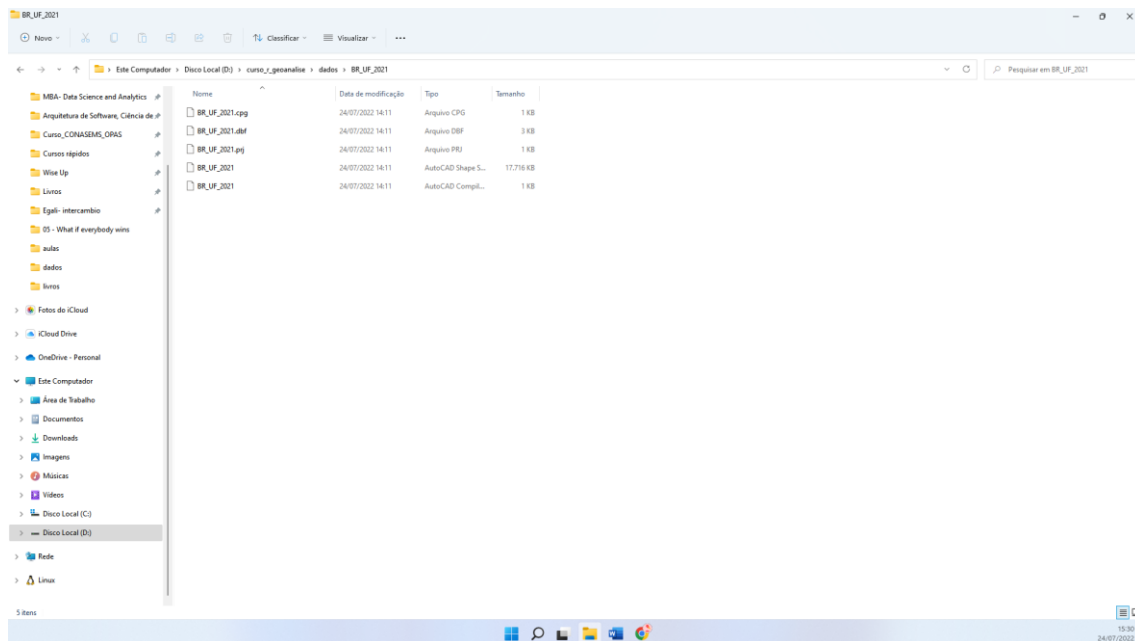


Após salvar o arquivo zip, você deverá descompactá-lo em uma pasta para onde possa direcionar o R posteriormente. Você pode descompactar o arquivo .zip clicando nele com o botão direito do seu *mouse* e clicando em “Extrair Tudo...”, conforme a imagem.



Após clicar em “Extrair Tudo...”, você poderá escolher o nome da pasta que receberá os arquivos descompactados. Por *default*, o *Windows* cria uma pasta com o mesmo nome do arquivo “.zip” para receber esses arquivos. Nós vamos usar esse nome para a pasta dos nossos arquivos geográficos. Após isso,

you will have the folder “BR_UF_2021” decompressed with all the files necessary for the importation in R.

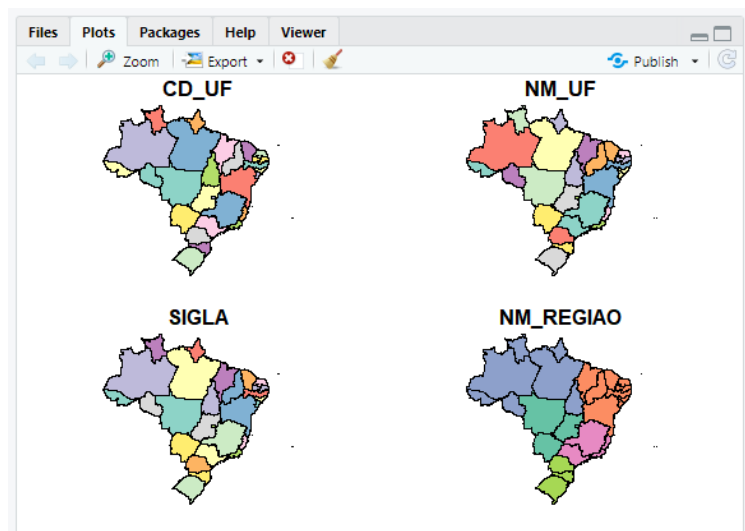


We saved the file in the “dados” folder of our R project and made it available for you in our Moodle environment, so it will not be necessary for you to download all the files again from IBGE. However, it is important to have this knowledge, as it can be very useful in the routines of the UF.

After importing the data and locating them in our “dados” folder, we will import the *shapefile* for our project using the function “*read_sf()*” from the package “*sf*”. We will also use the function “*plot()*” and “*glimpse()*” to understand what are the attributes of this *shapefile*. The function “*plot()*” will show us maps with the different attributes and the function “*glimpse()*” will show the table in *dataframe* format.

```
#IMPORTANDO SHAPEFILE DO BRASIL
br_shp = read_sf('dados/BR_UF_2021/BR_UF_2021.shp')
plot(br_shp) #plotando atributos básicos do shapefile
glimpse(br_shp) #visualizando a tabela de atributos
```

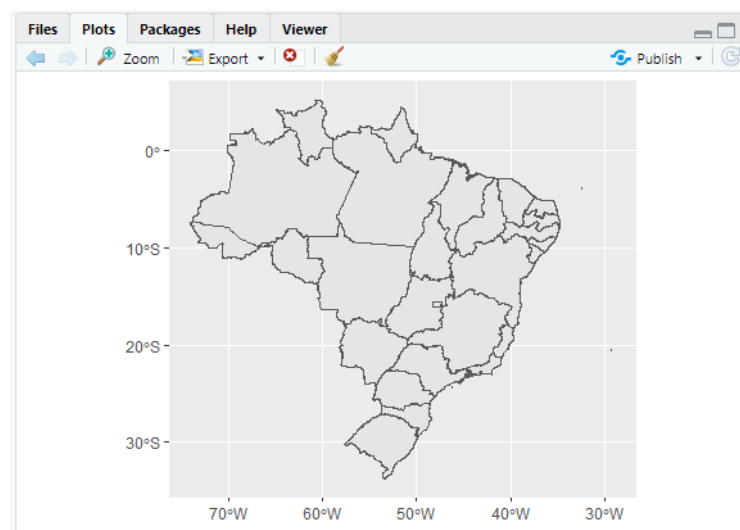
For the maps with the attributes, we have a visualization with the codes, names and abbreviations of the UF, which are unique, so each UF has a color in the thematic map, and the name of the region, which has only 5 names, so the map is colored in 5 colors that represent each one.



Nós vamos plotar o mapa do Brasil usando o *ggplot2*, sem destacar nenhum atributo, apenas para visualizar os polígonos, a latitude e a longitude do mapa.

```
ggplot() + #plotando o mapa com os dados do Brasil  
  geom_sf(data= br_shp)
```

Nós receberemos como *output* um mapa cinza, sem nenhuma adaptação de UF ou indicador de saúde.

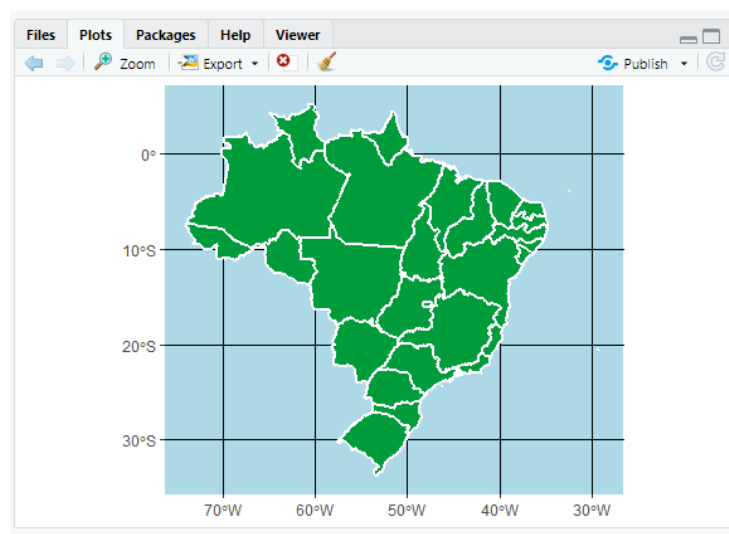


Nós vamos usar a os argumentos da função “*geom_sf()*” e a função “*theme()*”, ambas do pacote *ggplot2* para personalizar o nosso mapa. No capítulo anterior, usamos os argumentos “*fill*” e “*color*” para modificar a cor dos polígonos e das bordas do nosso mapa usando nome de cores que são reconhecidos pelo

R. Hoje, usaremos cores hexadecimais. Vamos preencher (*color* o mapa do Brasil com a cor verde de sua bandeira, que corresponde ao código hexadecimal "#009c3b", as bordas serão brancas de tamanho (*size*) 1. Para o fundo do mapa, usaremos um azul claro, com linhas pretas, usando a função *"theme"*.

```
ggplot() + #personalizando o mapa com os dados do Brasil
  geom_sf(data= br_shp,
    fill = "#009c3b", #verde da bandeira do Brasil em cor
    hexadecimal
    color = "white", size = 1)+
  theme(panel.background = element_rect(fill = "lightblue"),
    panel.grid.major = element_line(color = "black", size = 0.2))
```

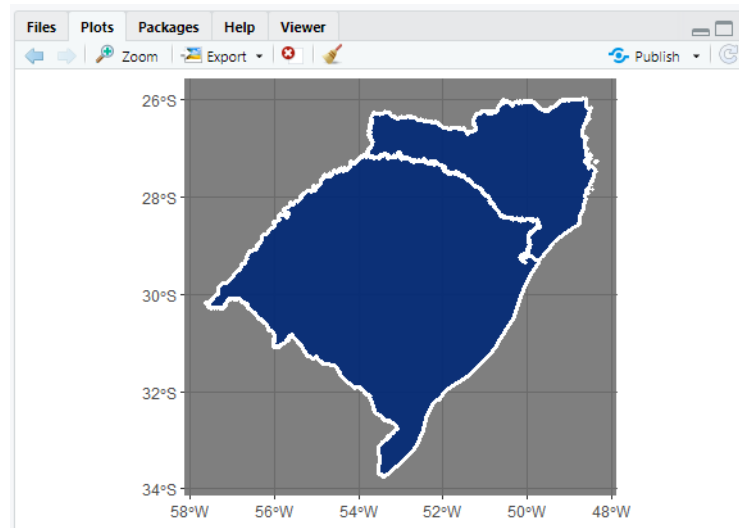
Como *output* teremos o seguinte mapa.



Com esse mapa, nós podemos ainda filtrar visualizações de estados, usando a função *"filter()"* e combinando com operadores lógicos. Nesse caso, vamos usar a função para filtrar os estados cujo nome é Santa Catarina ou cuja sigla é RS. Não vamos configurar o tema como no mapa anterior, mas vamos usar um tema pronto do *ggplot2* com o fundo preto. Vamos usar a cor azul da bandeira brasileira, com o código hexadecimal "#002776" para preencher os polígonos, que terão borda branca de tamanho 2 e uma transparência de 90%.

```
ggplot() + #realizando um filtro de unidades federadas
  geom_sf(data = br_shp %>%
    filter(NM_UF == "Santa Catarina" | SIGLA == "RS"),
    fill = "#002776", #azul da bandeira do Brasil em cor
    hexadecimal
    color = "white",
    alpha = 0.9,
    size = 1.2)+
  theme_dark()
```

Nós teremos um *output* com os estados de Santa Catarina e do Rio Grande do Sul em cor azul e borda branca, além do fundo escuro.



5. Centroides por polígono geográfico

Vamos elaborar um mapa de centroides com os dados de taxa de letalidade em cada UF. Mas, por que usamos as taxas de letalidade ou de incidência em vez do número de casos e número de óbitos? Quando estamos comparando diferentes localidades, como ocorre quando plotamos um mapa, devemos levar em consideração suas diferenças. Nesse caso, as populações de cada UF tem tamanhos diferentes, então os estados com o maior número de pessoas também terão o maior número de casos e, conseqüentemente, de óbitos. Quando plotamos os dados de casos ou óbitos, podemos dar a impressão de que um estado muito populoso possui a pior situação, quando na verdade ele só possui o maior número de pessoas. É importante prestar atenção nisso para não darmos a impressão errada aos consumidores das nossas análises.

Para criar o mapa de centroides, vamos criar dois *dataframes*. O primeiro receberá o código do IBGE, a taxa de letalidade e a taxa de incidência de casos por UF. O segundo será a junção do primeiro *dataframe* que criamos com a tabela de atributos do nosso *shapefile*.

```
#ELABORANDO MAPAS DE CENTROIDE POR UF

##UNINDO DATA.FRAME DE LETALIDADE E INCIDÊNCIA COM O SHAPEFILE DO
BRASIL

df_let = data.frame(
  CD_UF = as.character(letal_uf$ibge),
  TX_LETALIDADE = letal_uf$TX_LETALIDADE,
  TX_INCIDENCIA = letal_uf$TX_INCIDENCIA
```

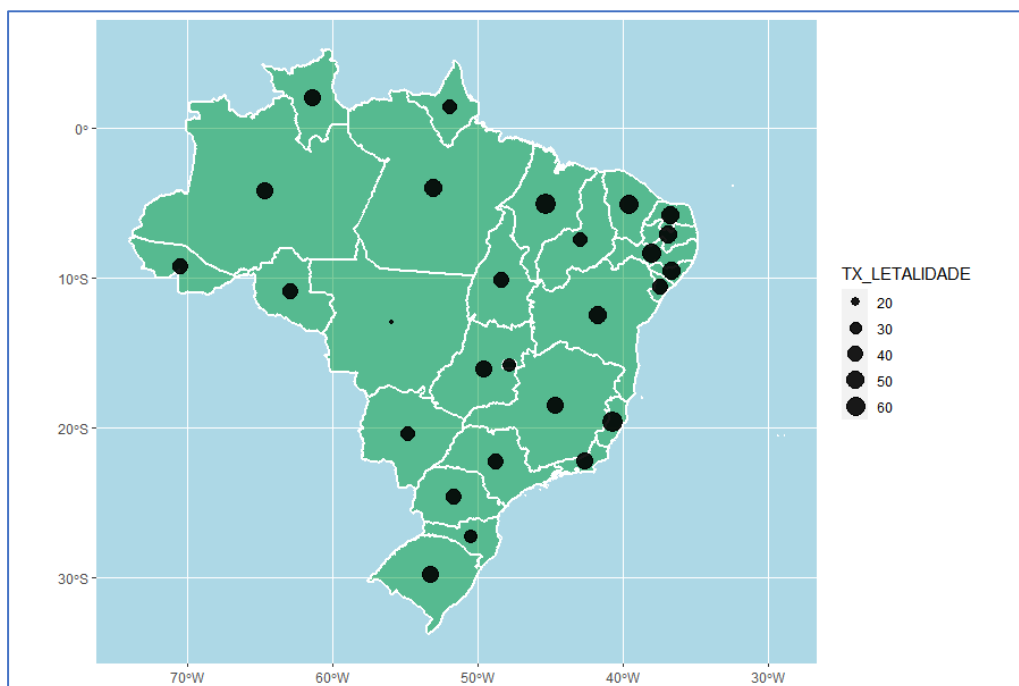
```
)
br_shp_let = br_shp |>
  left_join(df_let, by = "CD_UF")
```

Nós obteremos um objeto que possui, ao mesmo tempo, as informações geográficas de cada UF do Brasil e as informações epidemiológicas de letalidade e incidência por Covid-19. Nosso mapa vai mostrar a taxa de letalidade por UF, possuindo os atributos que mostramos anteriormente e os centroides que consideramos nessa análise.

```
coord_pontos <- br_shp_let %>% #gerando os centroides
  st_centroid()

ggplot() + #criando mapade centroides
  geom_sf(data = br_shp_let, fill = "#009c3b",
    color = "white", size = 1, alpha = 0.5) +
  geom_sf(data = coord_pontos, aes(size = TX_LETALIDADE), col =
    "black", alpha = .9,
    show.legend = "point")+
  theme(panel.background = element_rect(fill = "lightblue"),
    panel.grid.major = element_line(color = "white", size = 0.2))
```

Pronto! Temos nosso primeiro mapa de centroides. Perceba que usamos a função “*st_centroid()*” para gerar os centroides de cada uma das UF do nosso mapa. Para gerar usar o tamanho relacionado à taxa de letalidade, usamos a variável “TX_LETALIDADE” com o argumento *size* da função “*geom_sf()*” e usamos a configuração de cores do mapa e do tema de fundo que plotamos na seção anterior.



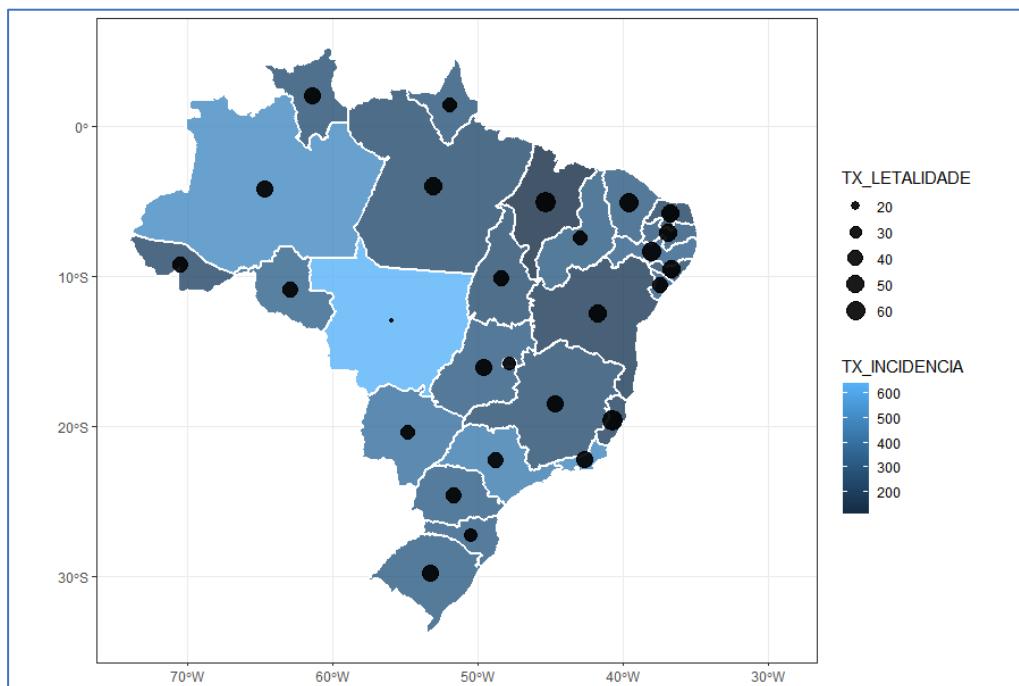
6. Mapas compostos: incluindo informações nos polígonos e centroides

Digamos que queremos comparar o desempenho das UF no enfrentamento à pandemia, e, para isso, vamos comparar os dados de incidência com letalidade, para identificar se as UF com as maiores taxas de incidência também tiveram maiores taxas de letalidade ou, se algumas UF com altas taxas de incidência conseguiram ter baixas taxas de letalidade.

Para isso, vamos criar um mapa com duas camadas de informações epidemiológicas. A primeira camada será um mapa temático, semelhante ao capítulo anterior, com os dados de taxa de incidência nos polígonos e com o dado de taxas de letalidade nos centroides. A diferença entre o anterior e esse é que estamos usando a variável “TX_INCIDENCIA” para graduar os polígonos usando o argumento “aes” e “fill”.

```
#ELABORANDO MAPAS COMPOSTOS: INFORMAÇÕES EM POLÍGONOS E CENTROIDES
ggplot() +
  geom_sf(data = br_shp_let, aes(fill = TX_INCIDENCIA), color =
"white", size = 0.8, alpha = 0.8) +
  geom_sf(data = coord_pontos, aes(size = TX_LETALIDADE), color =
"black", alpha = 0.9,
        show.legend = "point")+
  theme_bw()
```

Como *output* teremos um mapa com centroides e polígonos graduados.



7. Referências Bibliográficas

1. The Comprehensive R Archive Network [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://cran.r-project.org/>
2. devtools package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/devtools/versions/1.13.6>
3. Rinker T, Dason Kurkiewicz, Pastoor D. Pacman: Pacman Version 0.2.0 [Internet]. Zenodo; 2015 [citado 16 de julho de 2022]. Disponível em: <https://zenodo.org/record/15406>
4. ggsm package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/ggsm/versions/0.5.0>
5. sf package - RDocumentation [Internet]. [citado 24 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/sf/versions/1.0-8>
6. ggspatial package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/ggspatial/versions/1.1.6>
7. gpclib package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/gpclib/versions/1.5-6>
8. jsonlite package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/jsonlite/versions/1.8.0>
9. leaflet package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/leaflet/versions/2.1.1>
10. lubridate package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/lubridate/versions/1.8.0>
11. maptools package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/maptools/versions/1.1-4>
12. RColorBrewer package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/RColorBrewer/versions/1.1-3>
13. RCurl package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/RCurl/versions/1.98-1.7>

14. rgdal package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/rgdal/versions/1.5-32>
15. rgeos package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/rgeos/versions/0.5-9>
16. Ministério da Saúde B. Asis - Análise de Situação de Saúde. :282.
17. rjson package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/rjson/versions/0.2.21>
18. tidyverse package - RDocumentation [Internet]. [citado 16 de julho de 2022]. Disponível em: <https://www.rdocumentation.org/packages/tidyverse/versions/1.3.1>
19. The Grammar of Graphics [Internet]. [citado 24 de julho de 2022]. Disponível em: <https://link.springer.com/book/10.1007/0-387-28695-0>

8. Exercícios

Marque verdadeiro ou falso para a afirmativa	
Pata plotar mapas no R, é suficiente usar o pacote <i>ggplot2</i>	F
O pacote <i>ggspatial</i> é incompatível com o pacote <i>ggplot2</i>	F
Não conseguimos importar dados de arquivos <i>shapefile</i> diretamente no R, sendo necessário sempre baixar por um pacote como o <i>geobr</i>	F
Os arquivos <i>shapefile</i> do IBGE são incompatíveis com o R, sendo necessária a transformação manual dos dados antes da importação no R	F
Não é possível elaborar um mapa de centroides proporcionais diretamente no R, sendo necessário o uso de outras ferramentas de geoprocessamento ou edição de imagem	F
O IBGE disponibiliza malhas geográficas, que podem ser usadas no R, da União, estados, municípios e outras divisões no país em seu repositório <i>online</i>	V
É possível realizar o filtro de estados a partir de um <i>shapefile</i> do país, desde que esse atributo esteja disponível	V
Podemos combinar duas ou mais camadas de visualização em um único mapa R	V
Em um mapa com mais de uma camada de visualização, é possível estabelecer duas diferentes legendas usando o R	V
Para gerar os centroides de um mapa, podemos usar a função “ <i>st_centroid()</i> ” a partir de um mapa de polígonos.	V