

Aula 04 – Personalizando mapas

Sumário

Aula 04 – Personalizando mapas	1
1. Apresentação	2
2. Importando e ajustando dados do nosso mapa	3
3. Quebras de dados e paleta de cores	5
4. Espessura das bordas.....	8
5. Destaque de uma área geográfica	9
6. Escala	12
7. Rosa dos ventos.....	14
8. Removendo latitude e longitude nas margens do mapa	16
9. Título	17
10. Exportar o mapa	18
11. Referências Bibliográficas	20
12. Exercícios	22

1. Apresentação

Olá! Seja bem-vindo ao último 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.

Nós já aprendemos a importar dados geográficos, uni-los aos dados epidemiológicos, calcular e plotar indicadores de saúde, construir mapas compostos e tivemos algumas noções de personalização dos dados. Nesse último módulo vamos aprender a customizar nossos mapas.

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* “04_customizando_mapas.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. Importando e ajustando dados do nosso mapa

Nós construímos alguns mapas nas aulas anteriores e vamos usá-los hoje para aprender como personalizá-los. Inicialmente, vamos carregar os pacotes da nossa aula (1–18). Vamos usar o pacote *pacman* que verificará se os pacotes estão instalados e carregará após a instalação, se for necessária.

```
##INSTALANDO E CARREGANDO PACOTES COM O PACMAN
library(pacman)
p_load(
  geobr,          #pacote com datasets geoespaciais do Brasil
  ggsn,           #símbolos de direção e barras de escala para
pacotes criados com "ggplot2" ou "ggmap"
  ggspatial,      #dados espaciais de suporte para o pacote "ggplot2"
  gpclib,         #tratamento de feições de polígonos/áreas para o R
  jsonlite,       #leitor e gerador de arquivos no formato JSON
  leaflet,        #versão do R para a biblioteca Leaflet do
Javascript para mapas interativos
  lubridate,      #pacote para tratamento de datas no R
  maptools,       #manipulação de dados geográficos
  RColorBrewer,   #paletas de cor para visualizações do R
  RCurl,          #interface de client para o R (usado para simular
visualizações web)
  rgdal,          #funções de integração para a biblioteca gdal
  rgeos,          #operações com topologia e geometrias
  rio,            #importação e exportação de arquivos variados
  rjson,          #conversão de arquivos JSON
  sf,             #pacote para a leitura de arquivos .shp
  tidyverse       #conjunto de pacotes de manipulação e tratamento de
dados e criação de gráficos
)
```

Nós importaremos os dados de casos e de população para construir um *data.frame* com a taxa de incidência por Covid-19 no Brasil em 2020. Após a importação dos dados que construímos na semana anterior, vamos calcular nossa taxa de incidência.

```
#CRIANDO O MAPA QUE SERÁ CUSTOMIZADO

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

##CALCULANDO TAXA DE INCIDÊNCIA
populacao_uf = import("dados/populacao_uf_2020.csv")

casos_uf = casos_uf |>
  left_join(populacao_uf, by = c("SG_UF_NOT" = "sigla"))

casos_uf$TX_INCIDENCIA = round(casos_uf$CASOS/casos_uf$pop*100000, 2)
```

Importaremos também o arquivo *shapefile* que baixamos na semana passada do repositório do Instituto Brasileiro de Geografia e Estatística (IBGE).

Após isso, vamos criar um novo *data.frame* com os dados de UF e taxa de incidência e vamos uni-lo aos dados do arquivo *shapefile* que importamos.

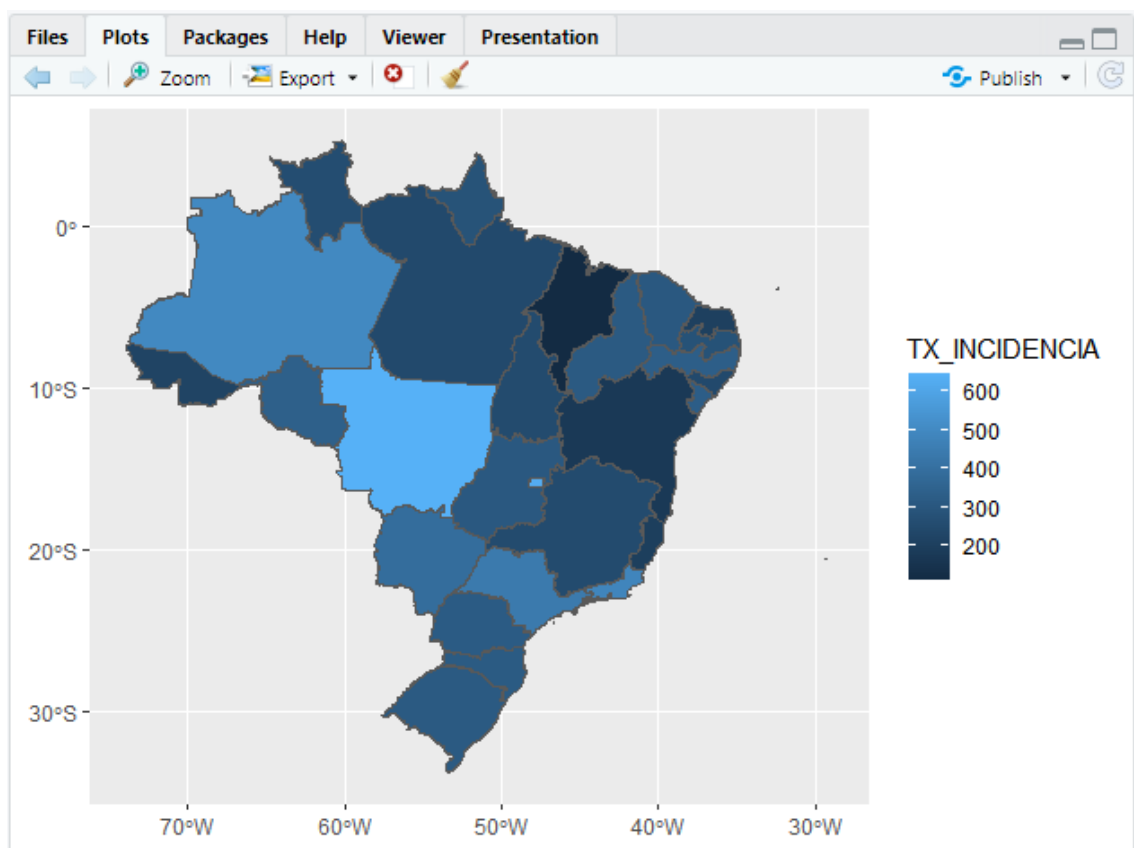
```
##IMPORTANDO SHAPEFILE DO BRASIL
br_shp = read_sf('dados/BR_UF_2021/BR_UF_2021.shp')

##UNINDO DADOS DO MAPA DO BRASIL COM OS DADOS DE INCIDÊNCIA
df_inc = data.frame(
  CD_UF = as.character(casos_uf$ibge),
  TX_INCIDENCIA = casos_uf$TX_INCIDENCIA
)

br_shp_inc = br_shp |>
  left_join(df_inc, by = "CD_UF")
```

Nós vamos plotar um mapa temático simples, com o padrão do *ggplot2* para que possamos customizá-lo nas próximas etapas.

```
##ELABORANDO MAPAS TEMÁTICO DE TAXA DE INCIDÊNCIA
ggplot() +
  geom_sf(data = br_shp_inc, aes(fill = TX_INCIDENCIA))
```



O *output* do nosso comando será um mapa do Brasil, na cor azul e com a graduação de incidência em centenas. Não temos um título, nem escala ou direção de norte no mapa. O título da legenda também é inadequado, visto que ele puxou automaticamente o título do nosso *data.frame*.

Nos próximos tópicos vamos customizar nosso mapa para deixá-lo de uma forma em que possamos usar em nossos relatórios ou artigos científicos.

3. Quebras de dados e paleta de cores

Por padrão, o *ggplot2* gradua a taxa de incidência do nosso mapa em centenas, mas a distribuição é um pouco confusa e não possui o mesmo número de estados. Nós vamos elaborar uma nova “quebra” nos dados, dividindo-os em quartis. Isso significa que nós vamos distribuir os dados em 4 partes iguais ordenando por sua taxa de incidência.

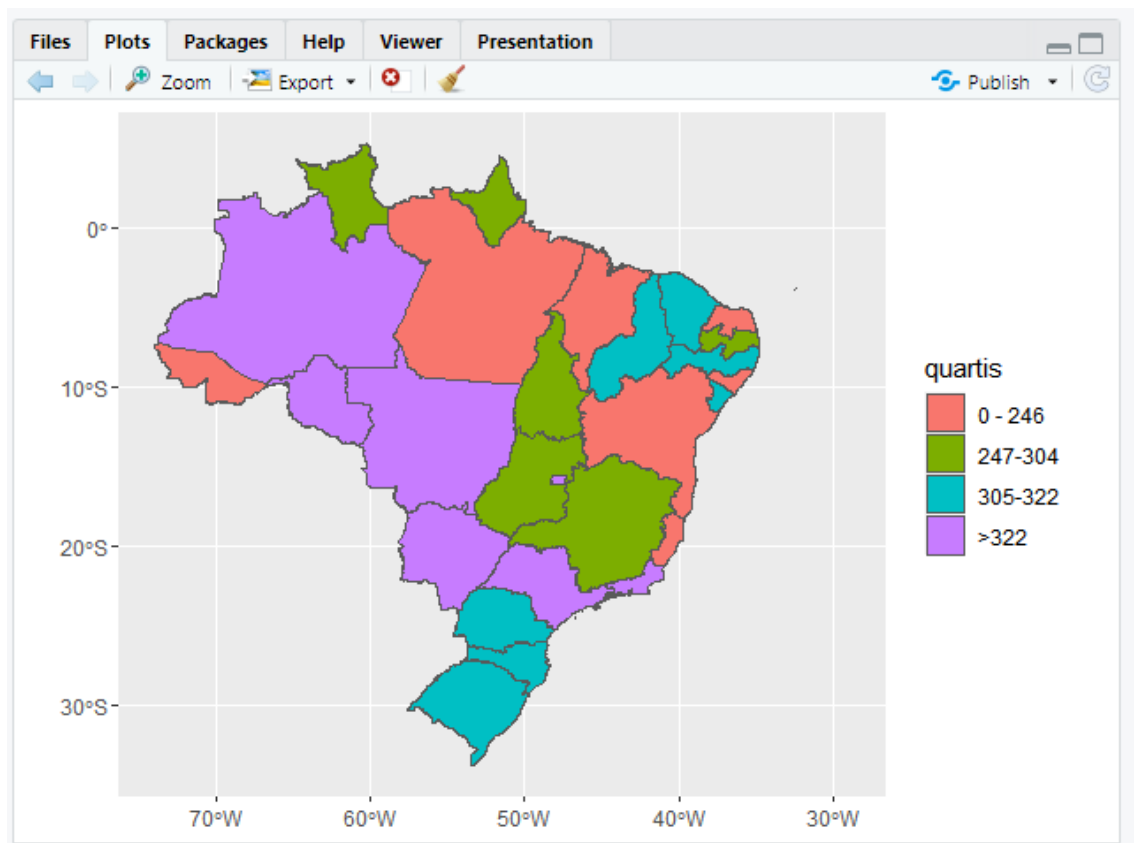
Para isso, vamos usar a função “*quantiles()*” para gerar essa quebra e depois criaremos uma nova variável com os valores determinados. Como esse valor será entendido como uma categoria no R, vamos usar a função “*fator()*” para gerar os intervalos e depois vamos gerar uma variável com esses dados como uma categoria ordinal usando a função “*levels()*”.

```
#CUSTOMIZANDO O MAPA

##GERANDO QUEBRAS DE ACORDO COM OS QUARTIS
quartis = quantile(br_shp_inc$TX_INCIDENCIA, prob=c(0.25, 0.5, 0.75))
br_shp_inc$quartis = factor(findInterval(br_shp_inc$TX_INCIDENCIA,
quartis))
levels(br_shp_inc$quartis) <- c("0 - 246", "247-304", "305-322",
">322") #usando os valores dos quartis para gerar a variável
```

Após isso, vamos plotar um novo mapa, agora usando a variável “quartis” em vez de “TX_INCIDENCIA”.

```
##PLOTANDO O MAPA
ggplot()+
  geom_sf(data = br_shp_inc,
          aes(fill = quartis))
```



O mapa que nós criamos possui a ordem dos quartis e os separou em cores. Porém, essas cores não são adequadas para representar a graduação da taxa de incidência que queremos mostrar no mapa porque o R entendeu que essas categorias não são numéricas.

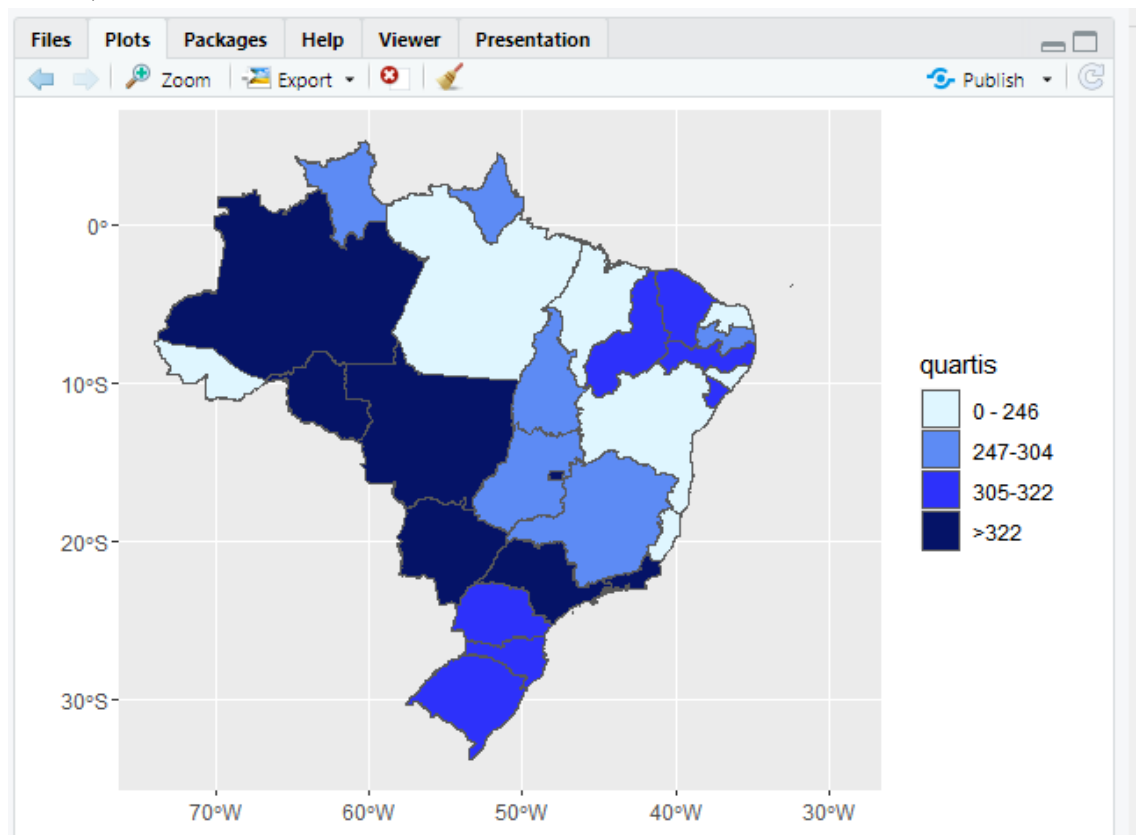
A taxa de incidência cresce em cada categoria que definimos com o R e nós queremos demonstrar isso no mapa. Para isso vamos usar paletas de cores que possuam a mesma cor, mas que vá escurecendo conforme nós aumentamos a taxa de incidência.

Existem duas principais formas de fazer isso. Na primeira forma, que é a escolha manual de cores, nós vamos escolher quatro tons do mesmo azul e vamos criar uma paleta com os 4 tons graduados, começando pelo mais claro e chegando no mais escuro e vamos salvá-los em um objeto chamado *pallette*. Depois nós vamos usar a função “*scale_fill_manual()*” para usar essa paleta de cores no mapa que nós criamos.

```
##MODIFICANDO PALETA DE CORES MANUALMENTE
pallette = c("#dff6ff", "#5d8bf4", "#2d31fa", "#051367") #tons
graduados de azul

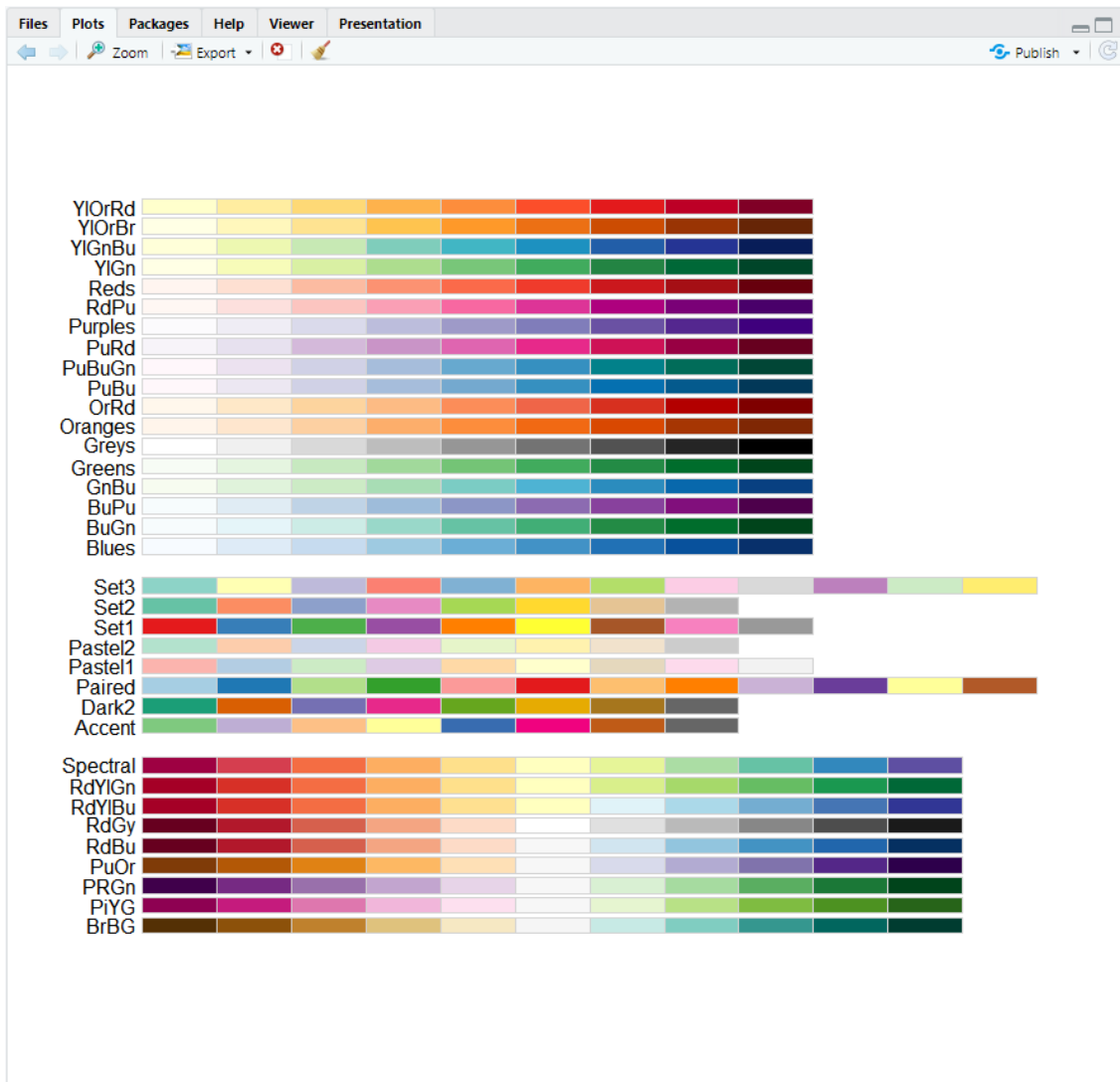
ggplot()+
  geom_sf(data = br_shp_inc,
          aes(fill = quartis))+
  scale_fill_manual(values = pallette)
```

Nós vamos ter um mapa com as cores graduadas. Pelo mapa podemos perceber agora que as piores taxas de incidência foram na região Norte e Centro-oeste, além de São Paulo e Rio de Janeiro.



Além dessa, existe uma forma automática de usar paletas já definidas no R, usando o pacote *RColorBrewer*, que é um pacote que possui diversas paletas de cores já montadas e que podem ser usadas com dados categóricos e dados numéricos. Nós podemos usar a função "*display.brewer.all()*" para identificar todas as paletas de cores que esse pacote possui.

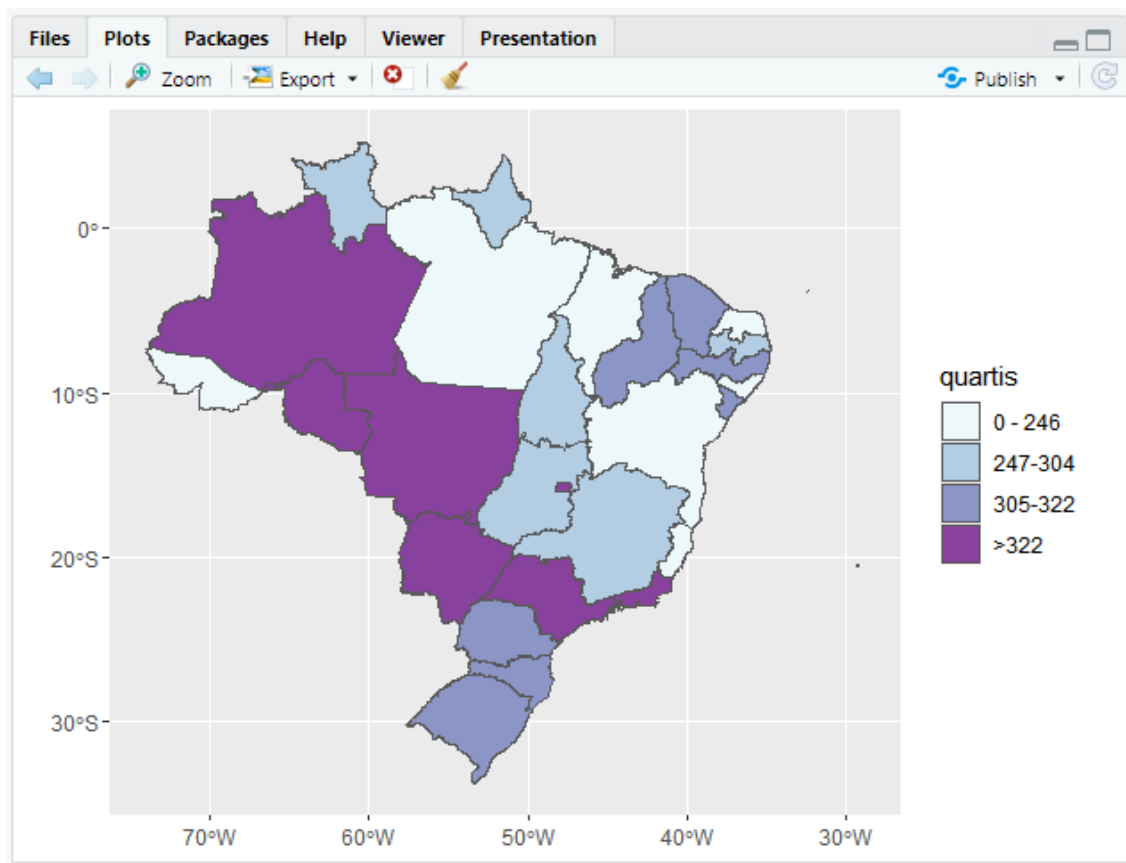
```
##MODIFICANDO PALETA COM O PACOTE "RColorBrewer"  
display.brewer.all() #plotando paletas de cores
```



O pacote possui diversas paletas de cores que podem ser utilizadas. Nós vamos usar a paleta “*BuPu*”, que começa com tons azuis e evolui para um roxo escuro.

```
##MODIFICANDO PALETA COM O PACOTE "RColorBrewer"
display.brewer.all() #plotando paletas de cores

ggplot()+
  geom_sf(data = br_shp_inc,
          aes(fill = quartis))+
  scale_fill_brewer(palette="BuPu")
```

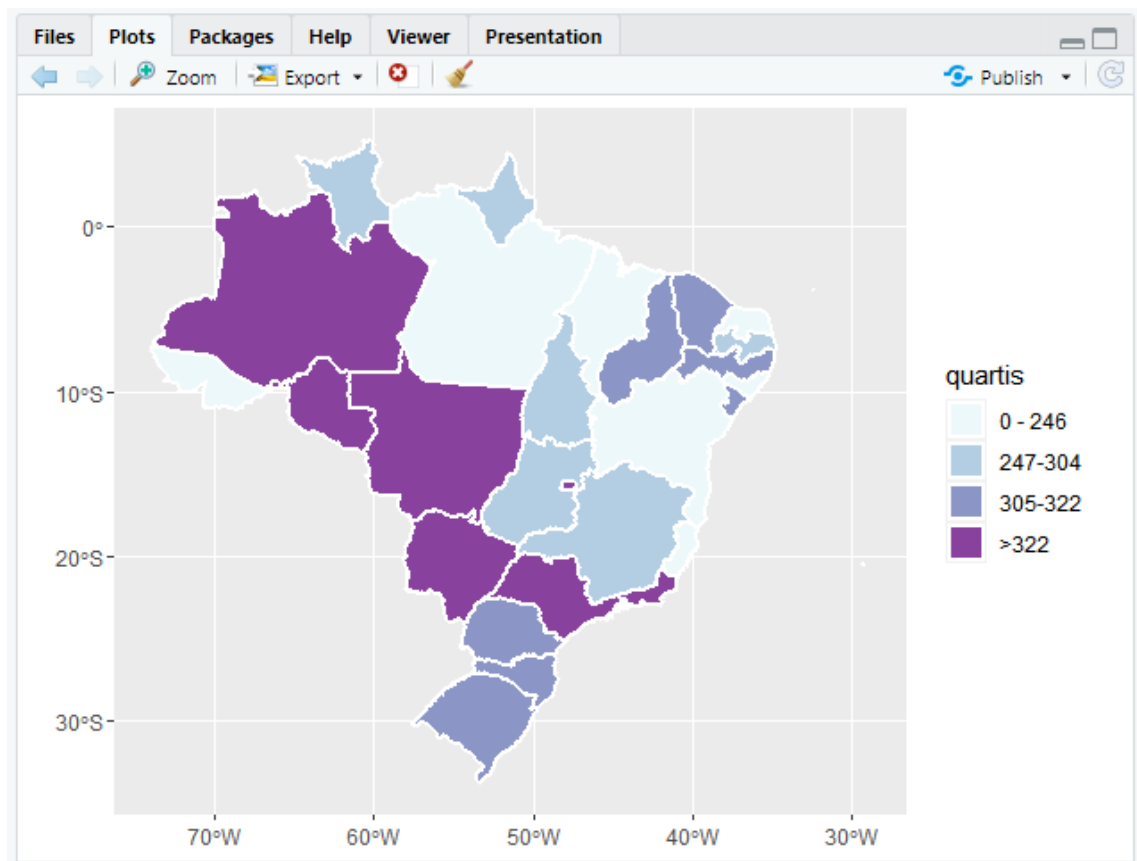



4. Espessura das bordas

Além de mudar a paleta de cores de preenchimento do mapa, nós podemos mudar suas bordas em cor e espessura, usando os argumentos “*color*” e “*size*”, respectivamente. Inicialmente, vamos usar a cor branca para as bordas, com um tamanho 1.

```
##MODIFICANDO ESPESSURA DAS BORDAS
ggplot()+
  geom_sf(data = br_shp_inc,
          aes(fill = quartis), color = "white", size = 1)+ #incluindo
cor branca na borda e tamanho igual a 1
  scale_fill_brewer(palette="BuPu")
```

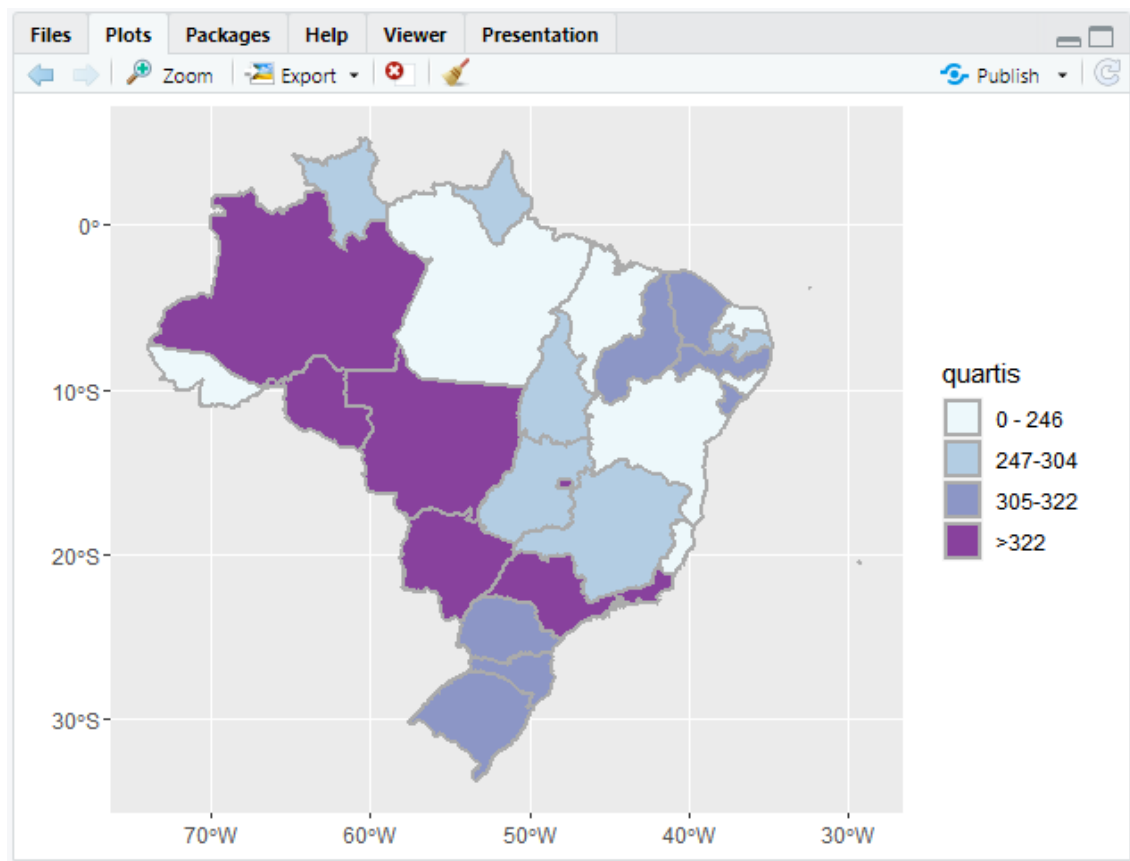
O nosso mapa ganhou cores brancas nas bordas, mas a visualização não ficou muito boa, principalmente nos tons mais claros do mapa.



Nós vamos modificar a cor para cinza, assim poderemos visualizar melhor nosso mapa.

```
ggplot()+  
  geom_sf(data = br_shp_inc,  
          aes(fill = quartis), color = "gray67", size = 1)+ #incluindo  
cor cinza na borda e tamanho igual a 1  
  scale_fill_brewer(palette="BuPu")
```

A visualização do mapa com a cor cinza com um contraste mais alto entre o primeiro tom da paleta e a borda deixa a visualização do mapa mais clara.



5. Destaque de uma área geográfica

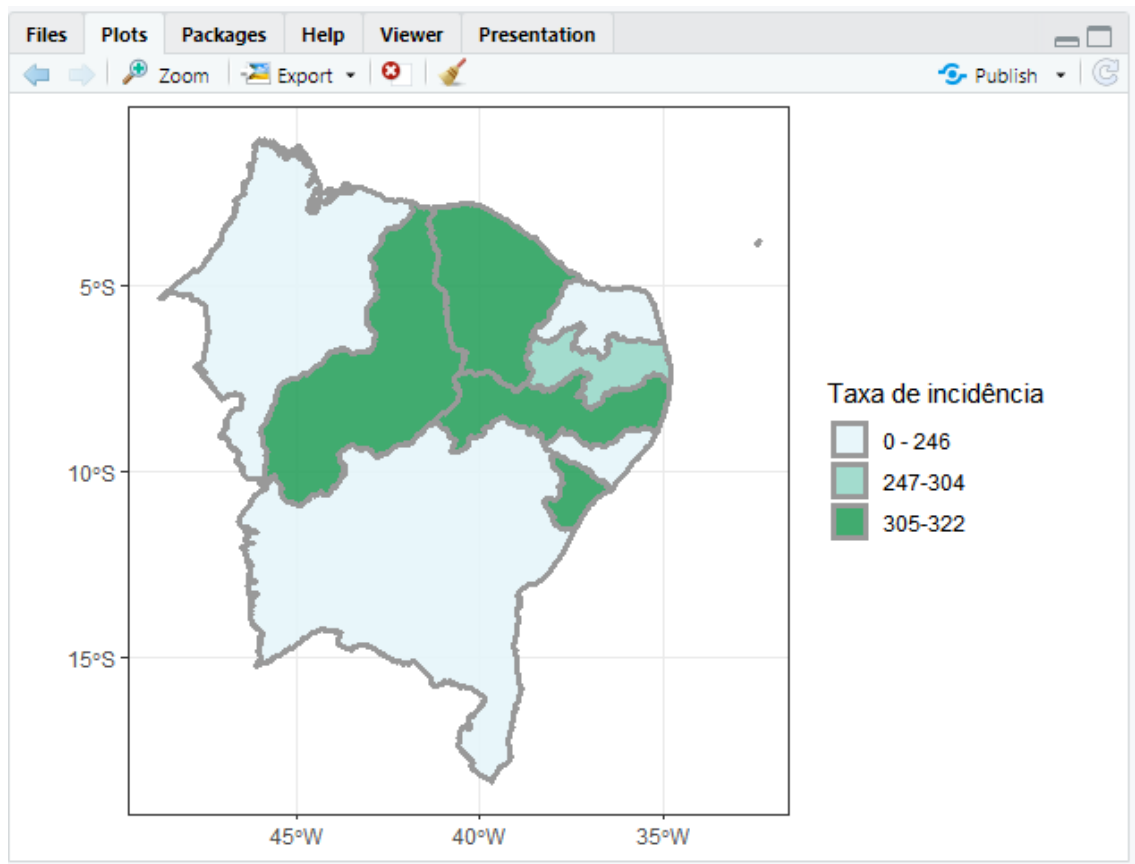
Nós podemos ainda destacar uma área geográfica usando a função “*filter()*”. Vamos usar a função para destacar os estados do Nordeste. Vamos ainda alterar o título da legenda usando a função “*guides()*”. Usaremos uma leve transparência no preenchimento e vamos usar uma borda no tamanho 1.2.

Nesse caso, se quiséssemos plotar um mapa da região com a distribuição de taxas de incidência para aquele território, deveríamos recalculer os quartis das taxas de incidência dessa região. Como vamos usar os quartis que foram definidos para o Brasil, os estados da região Nordeste ficarão com apenas 3 categorias, pois não há estados com taxas maiores de 322 casos por 100.000 habitantes.

```
##DESTACANDO UMA ÁREA GEOGRÁFICA

ggplot() + #realizando um filter de unidades federadas
  geom_sf(data = br_shp_inc %>%
    filter(NM_REGIAO == "Nordeste"),
    aes(fill = quartis), #graduando por quartis nacionais
    color = "gray60", #incluindo borda cinza
    alpha = 0.9, #gerando transparencia
    size = 1.2)+ #escolhendo tamanho da borda
  scale_fill_brewer(palette="BuGn")+
```

```
guides(fill=guide_legend(title="Taxa de incidência"))+ #mudando
título da legenda
theme_bw()
```



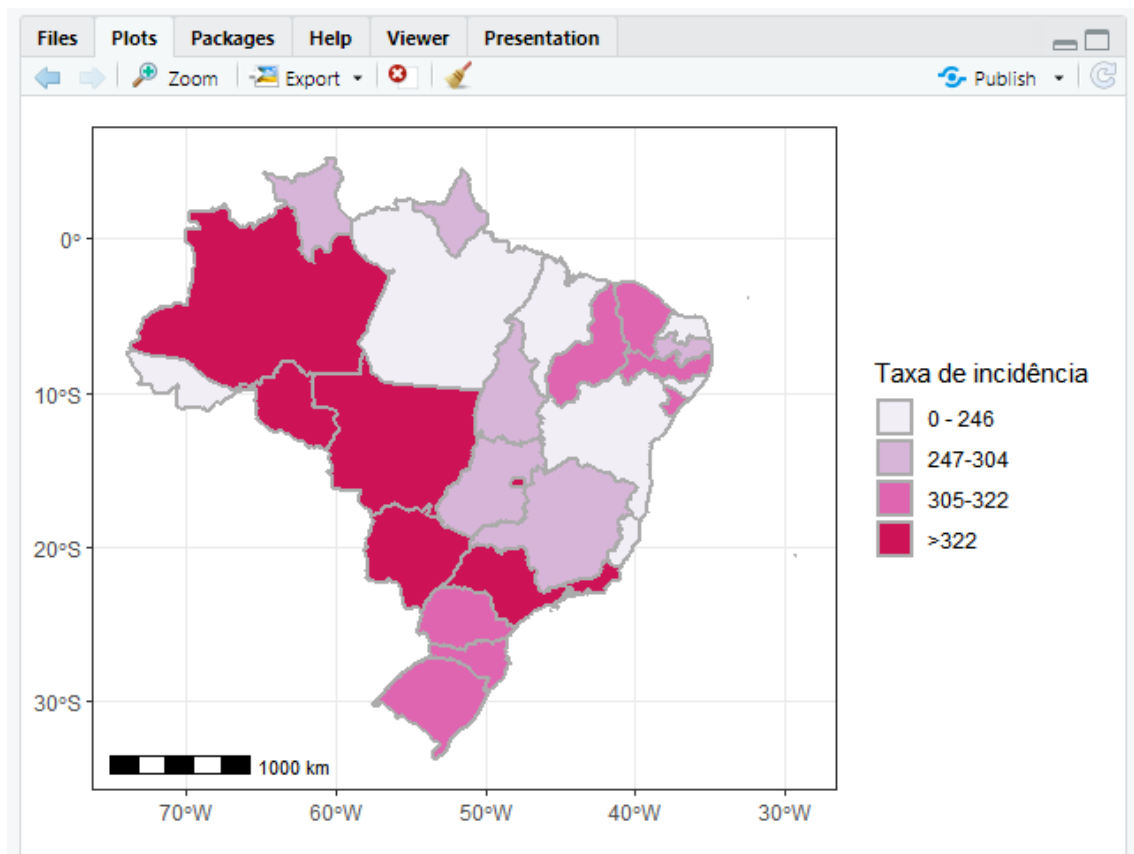
6. Escala

Um elemento muito importante na visualização de um mapa é a escala. Ela serve para mostrar a proporção entre os valores reais e o mapa plotado. No R, podemos usar a função “*annotation_scale()*” para incluir a barra de escala em um mapa.

```
##ESCALA

ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="PuRd")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme_bw()+
  annotation_scale() #adicionando escala
```

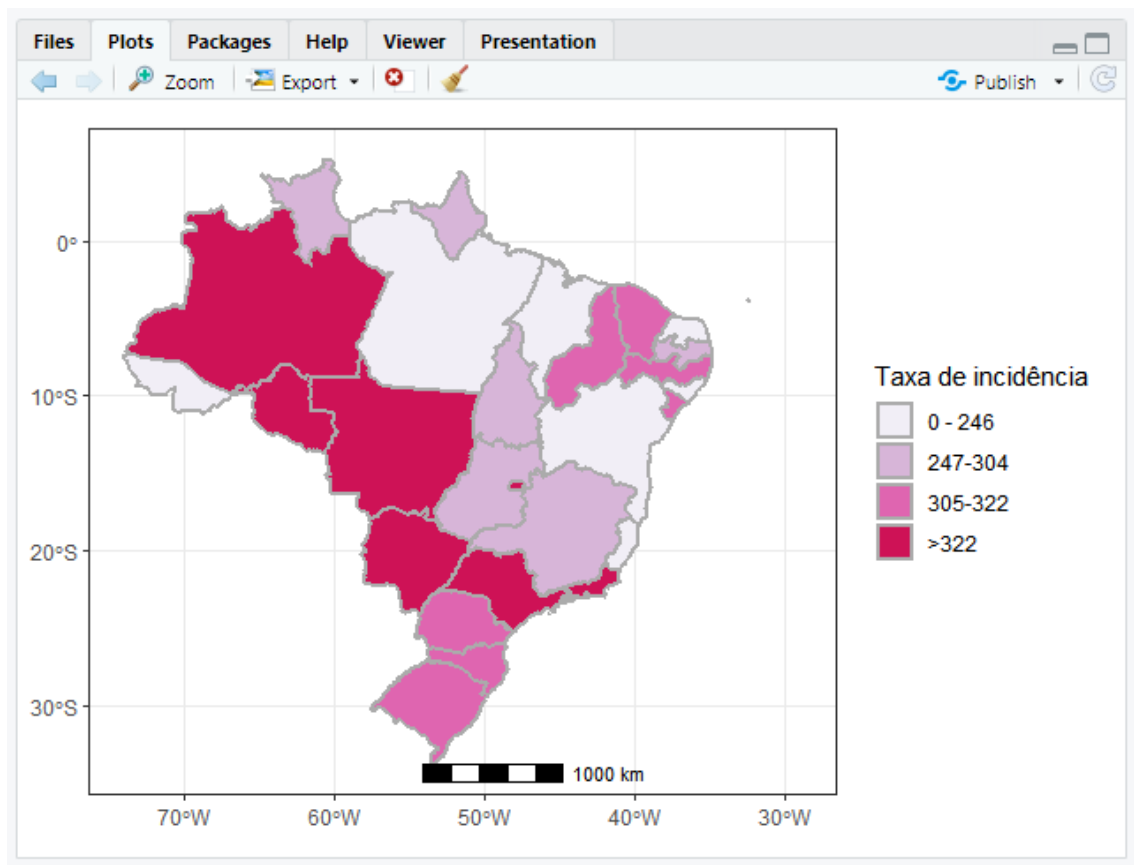
Por padrão, nós obteremos um mapa que possui a escala no canto inferior esquerdo do mapa.



É possível alterarmos o local em que a escala está disposta. Nós usaremos os argumentos “*pad_x*” e “*pad_y*” para alterar o local usando a posição nos eixos x e y.

```
ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="PuRd")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme_bw()+
  annotation_scale(pad_x = unit(7, "cm"), #mudando escala de lugar
    pad_y = unit(0.2, "cm"))
```

Com essas configurações, nós plotamos um mapa que possui a escala no centro inferior do mapa.



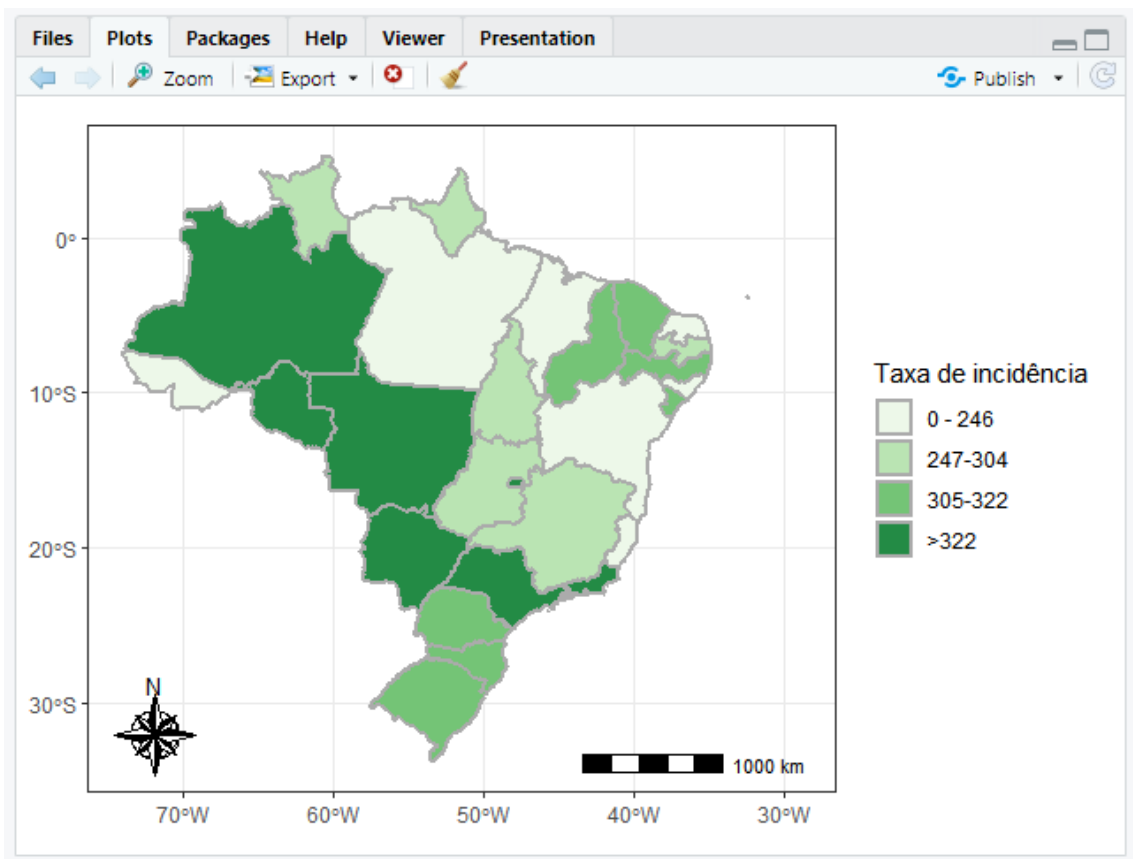
7. Rosa dos ventos

Além da escala, indicar a direção de norte é importante para plotarmos o nosso mapa de forma que outras pessoas possam compreendê-lo. A função “*annotation_north_arrow()*” possui várias opções de rosa dos ventos e setas de indicação de norte que podemos usar.

```
##ROSA DOS VENTOS

ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="Greens")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme_bw()+
  annotation_scale(pad_x = unit(7.4, "cm"),
    pad_y = unit(0.3, "cm"))+
  annotation_north_arrow(style=north_arrow_nautical()) #adicionando
rosa dos ventos
```

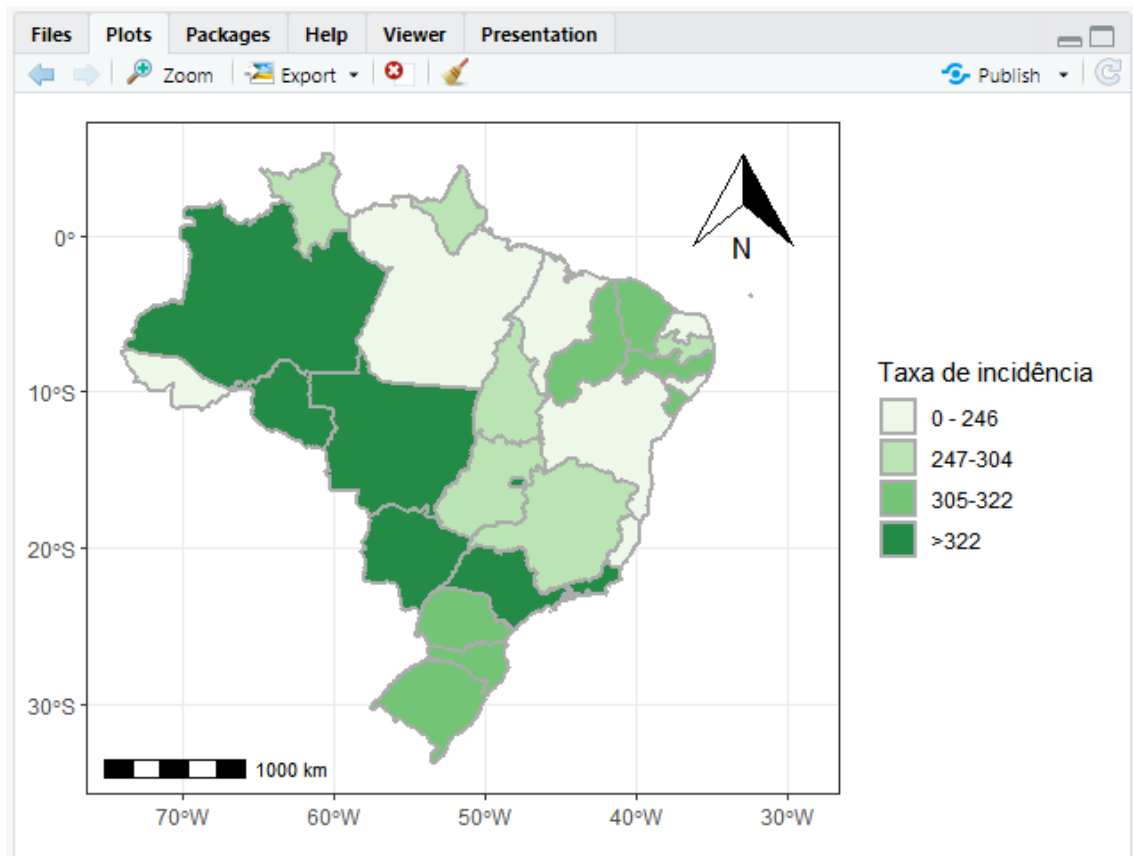
Por padrão, ao adicionarmos esse elemento, a disposição ocorre no canto inferior direito.



Nós conseguimos mudar o tipo de rosa dos ventos com o argumento “*style*” e conseguimos usar os argumentos “*pad_x*” e “*pad_y*” para alterar o local usando a posição nos eixos x e y.

```
ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="Greens")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme_bw()+
  annotation_scale()+
  annotation_north_arrow(style=north_arrow_orienteering(),
    #modificando rosa dos ventos para seta de norte
    pad_x = unit(14, "cm"), #mudando
    posição da seta
    pad_y = unit(13, "cm"))
```

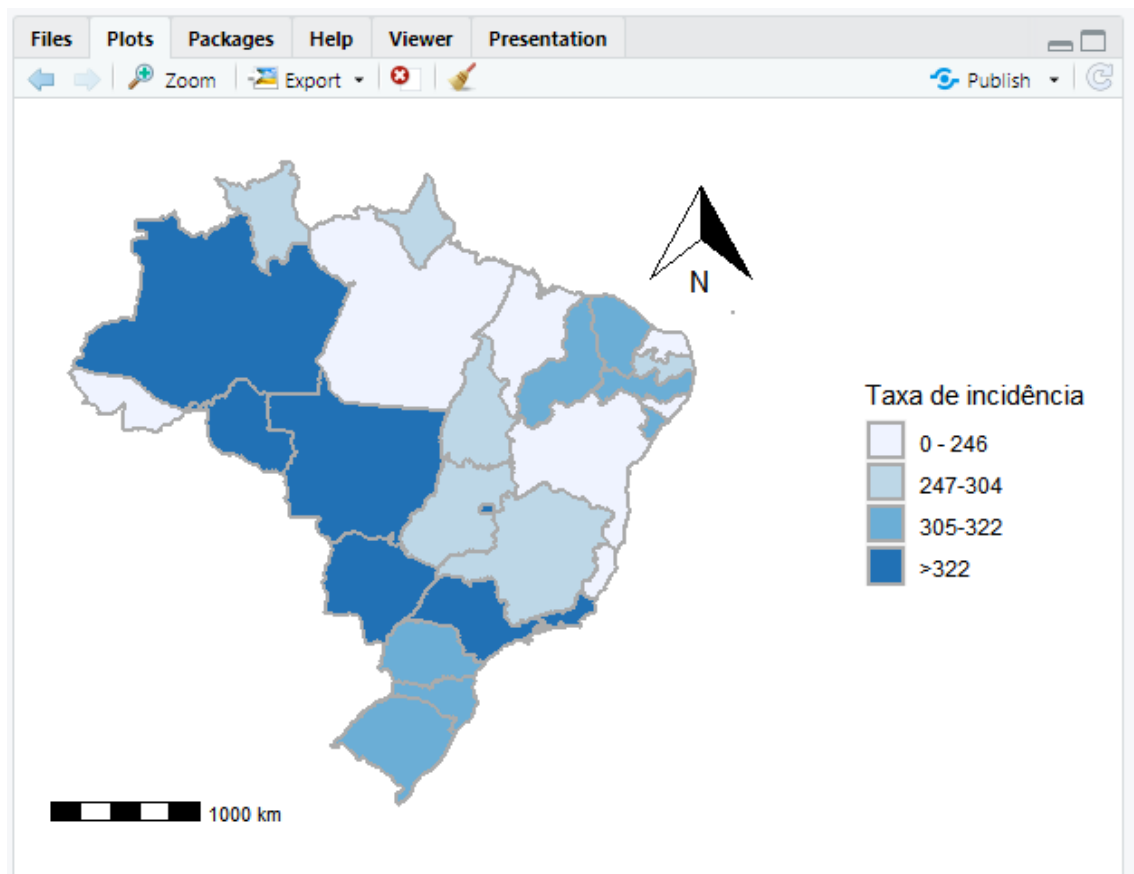
Nesse caso, incluímos uma seta de indicação de norte mais simples que a utilizada anteriormente e mudamos sua localização.



8. Removendo latitude e longitude nas margens do mapa

Por fim, para termos uma visualização mais “limpa” do mapa em si, vamos remover os eixos com a indicação de latitude e longitude e a grade cartesiana ao fundo do mapa.

```
Para isso, usaremos as configurações manuais da função "theme()" do
ggplot2.
ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="Blues")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme(panel.background = element_blank(), #mudando o tema para
excluir a grade de fundo
    axis.ticks = element_blank(),           #excluindo dados dos eixos
    axis.text = element_blank())+          #excluindo títulos dos
eixos
  annotation_scale()+
  annotation_north_arrow(style=north_arrow_orienteeing(),
    pad_x = unit(14, "cm"),
    pad_y = unit(13.5, "cm"))
```

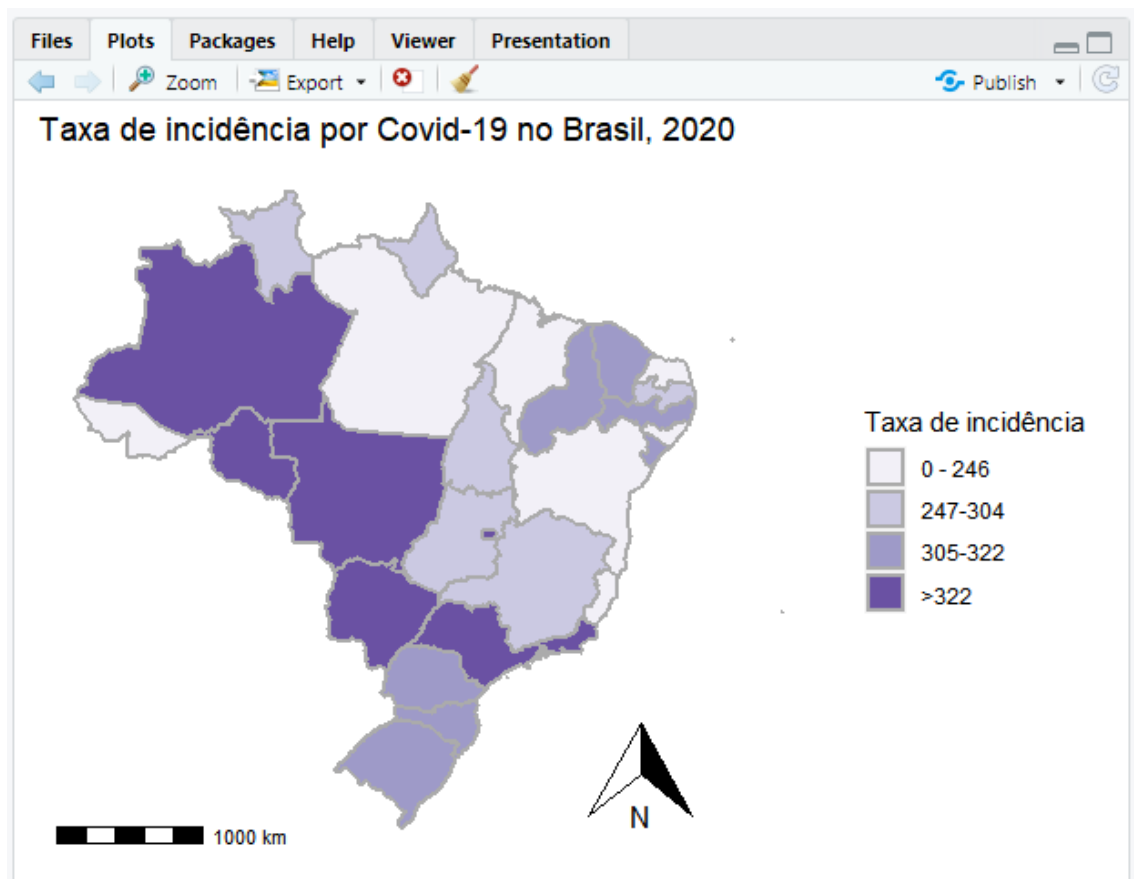



9. Título

Nós já temos um mapa bem definido e que pode ser utilizado em nossos documentos. Caso ele seja usado com outro tipo de legenda, não é necessário incluir o título na imagem, já que esse pode ser incluído diretamente no arquivo em que o mapa for colocado.

Entretanto, podemos incluir o título na imagem usando a função `"ggtitle()"`.

```
##TÍTULO
ggplot()+
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="Purples")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme(panel.background = element_blank(),
    axis.ticks = element_blank(),
    axis.text = element_blank())+
  annotation_scale()+
  annotation_north_arrow(style=north_arrow_orienteering(),
    pad_x = unit(9, "cm"),
    pad_y = unit(0.5, "cm"))+
  ggtitle("Taxa de incidência por Covid-19 no Brasil, 2020")
#Incluindo título
```



10. Exportar o mapa

É possível salvar os dados usando a função *print* do seu computador (se o sistema operacional for *Windows*) ou usando a função *Export* do *Rstudio*. Essas opções, no entanto, podem distorcer nosso mapa e proporcionar uma qualidade menor em relação à imagem.

Uma forma de exportar a imagem com uma qualidade melhor é usar a função “*ggsave()*”, que exporta como uma imagem em um formato específico o último mapa plotado em seu ambiente.

Nós vamos salvar o último mapa que fizemos em um objeto chamado “*mapa*”, após isso vamos plotar o mapa.

```
##EXPORTANDO O MAPA

mapa_final = ggplot()+ #salvando o mapa em um objeto
  geom_sf(data = br_shp_inc,
    aes(fill = quartis),
    color = "gray67",
    size = 1)+
  scale_fill_brewer(palette="Purples")+
  guides(fill=guide_legend(title="Taxa de incidência"))+
  theme(panel.background = element_blank(),
    axis.ticks = element_blank(),
```

```

    axis.text = element_blank(),
    plot.title = element_text(size = 20, face = "bold"))+
#aumentando o título e colocando em negrito
    annotation_scale()+
    annotation_north_arrow(style=north_arrow_orienteering(),
                           pad_x = unit(21, "cm"),
#alterando local para se ajustar ao tamanho do título
                           pad_y = unit(21, "cm"))+
    ggtitle("Taxa de incidência por Covid-19 no Brasil, 2020")

mapa_final

```

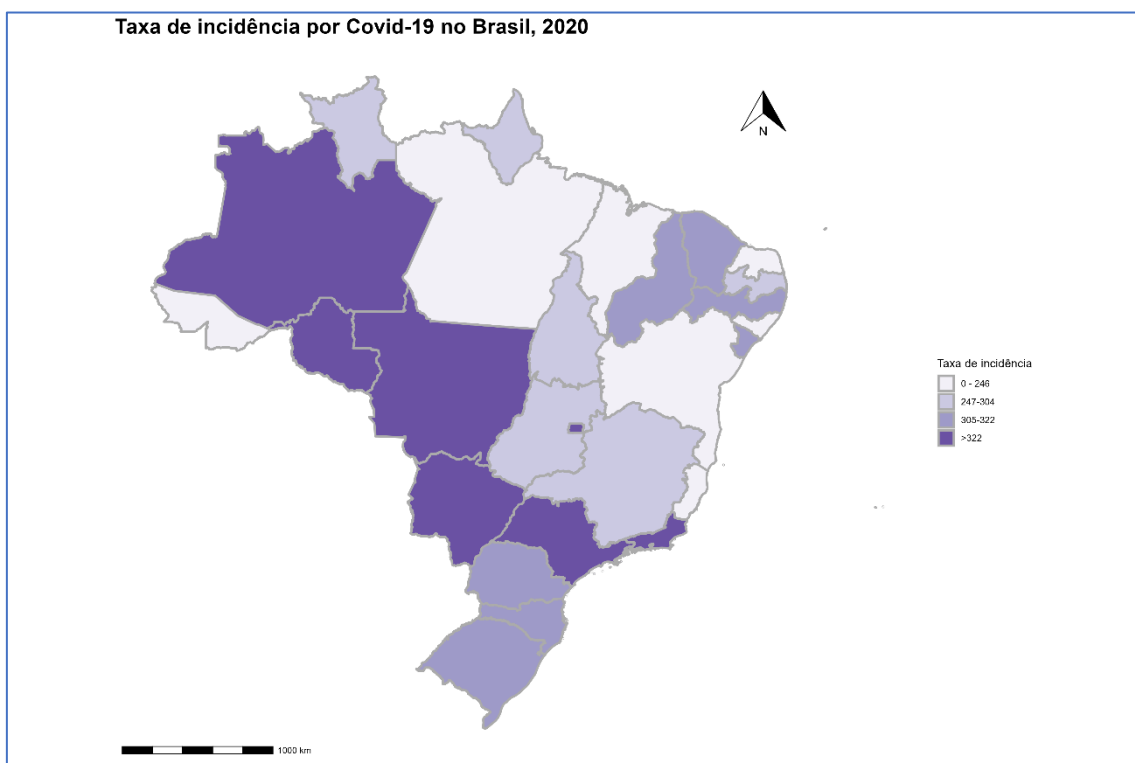
Após isso, vamos usar a função “*ggsave()*” para salvar um mapa chamado “*mapa_tx_inc_covid*” em formato “.png”. Vamos configurar a largura como 15 e a altura como 10 e a qualidade da imagem com 320 de dpi.

```

ggsave(filename = "mapa_tx_inc_covid.png", #exportando o último mapa
        plotado
        height = 10, width = 15, dpi = 320)

```

Por fim, obteremos um mapa com título e qualidade que poderá ser exportado para nossos documentos.



11. 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>

12. Exercícios

Marque verdadeiro ou falso para a afirmativa	
	F
	F
	F
	F
	F
	V
	V
	V
	V
	V