



Curso de programação em R - LABIMEC (2ª parte - 1º dia)

Pedro Milreu Cunha

Universidade Federal da Paraíba (UFPB) - PPGE/UFPB & LABIMEC

05-12-2022

Revisão

1. R, RStudio e instalação de pacotes;

`install.packages()`, `library()`, ...

2. Importação de dados;

`read.csv()`, `read.delim()`, `read.table()`, ...

3. Limpeza e manipulação de dados;

`%>%`, `filter()`, `select()`, `mutate()`, ...

4. Análise de dados;

`group_by()`, `summarise()`, `ggplot()`, ...

Configurações necessárias para essa aula

Configurações necessárias para essa aula

```
# Bibliotecas ----
library(dplyr) # Manipulação de dados
library(readxl) # Importação de arquivos .xlsx

library(sf) # Importação e manipulação de dados geográficos (.shp)
library(tmap) # Criação de choropleth maps
library(RColorBrewer) # Paleta de cores

# Comando especial ----
`%notin%` <- Negate(`%in%`)

# Função para renderizar figuras em .svg corretamente no LaTeX
show_fig <- function(f) # Fonte: https://stackoverflow.com/a/56044642
{if (knitr::is_latex_output())
  {
    output = xfun::with_ext(f, 'pdf')
    rsvg::rsvg_pdf(xfun::with_ext(f, 'svg'), file = output)
  } else {
    output = xfun::with_ext(f, 'svg')
  }
  knitr::include_graphics(output)
}
```

Criação de mapas

- ▶ Os mapas são elaborados a partir de polígonos com coordenadas geográficas, permitindo uma visualização espacial dos dados;
- ▶ Em particular, têm-se os chamados *choropleth maps*, que utilizam escalas de cores para demonstrar a distribuição espacial de alguma variável de interesse.
- ▶ Há várias bibliotecas disponíveis para esse fim: `'spplot'`, `'tmap'` e `'ggplot2'` são algumas opções. **Faremos uso do pacote `tmap` nesse curso.**

Exemplo de mapa elaborado utilizando a biblioteca tmap

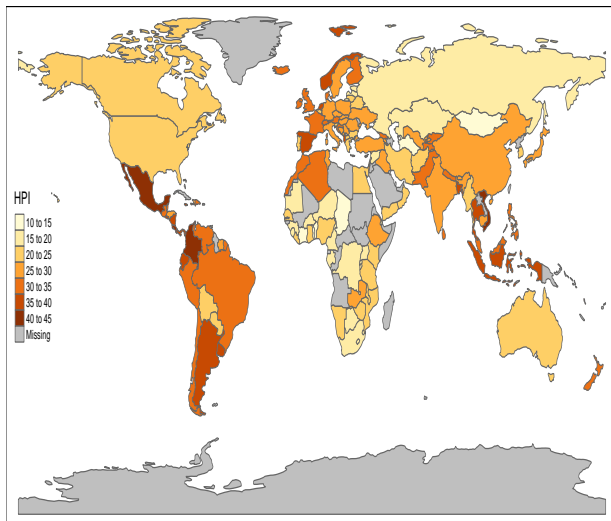


Figura 1: Exemplo de mapa criado pela biblioteca 'tmap'.

O que é necessário para gerar um mapa utilizando R?

1. Shapefiles:

as informações absolutamente *necessárias* para criação de mapas são aquelas referentes à geografia e polígonos, ou seja, os valores que mostram como os dados se distribuem espacialmente. São obtidos com os *shapefiles*¹, conjunto de arquivos com as extensões *.cpq*, *.dbf*, *.prj*, *.shp*, *.shx*;

2. Variáveis distribuídas de acordo com o nível geográfico utilizado para criação do mapa:

dados referentes aos fenômenos cuja distribuição espacial queremos analisar, como, por exemplo: i) a população ou PIB per capita para cada UF brasileira; ii) o nível de abertura comercial para cada país do Mercosul, etc.

¹Esse não é o único formato possível de representação de dados geográficos.

Onde obter os *shapefiles* para o Brasil?

- ▶ Os *shapefiles* para o Brasil são disponibilizados pelo *Instituto Brasileiro de Geografia e Estatística (IBGE)* em seu site: [IBGE](#);
- ▶ É possível obter as malhas territoriais de municípios, UFs, micro e mesorregiões, regiões geográficas imediatas e intermediárias e do país como um todo;
- ▶ Para que os arquivos sejam importados corretamente é necessário que todos os arquivos extraídos após download estejam em uma mesma pasta. No caso da malha territorial por estado, tem-se o arquivo comprimido `BR_UF_2021.zip` com os seguintes arquivos dentro: `BR_UF_2021.cpg`, `BR_UF_2021.dbf`, `BR_UF_2021.prj`, `BR_UF_2021.shp`, `BR_UF_2021.shx`.

Manipulação dos dados

1. O primeiro passo é importar a geometria base do mapa. Para tanto, utilizamos o arquivo BR_UF_2021.shp e a função `st_read()` da biblioteca `sf`. Além disso, também criamos uma coluna com as siglas das UFs brasileira no arquivo importado.

```
dados_geo <- st_read("data/shapefiles/BR_UF_2021.shp")
dados_geo$Sigla_UF <- c("RO", "AC", "AM", "RR",
                        "PA", "AP", "TO", "MA",
                        "PI", "CE", "RN", "PB",
                        "PE", "AL", "SE", "BA",
                        "MG", "ES", "RJ", "SP",
                        "PR", "SC", "RS", "MS",
                        "MT", "GO", "DF")
```

Passo necessário para a posterior união dos dados

2. Em seguida, lemos o arquivo com os dados cuja distribuição espacial queremos analisar e realizamos as transformações necessárias (nesse caso adicionar as siglas dos estados ausentes, dando o valor NaN para as observações em questão). Nesse exemplo vamos utilizar os dados referentes à subnotificação de estupros disponíveis no arquivo `data/sub_estimadas_estupros.xlsx`.

```
dados_sub <- read_excel("data/sub_estimadas_estupros.xlsx") %>%  
  select(Sigla_UF, sub_lag_end) %>%  
  group_by(Sigla_UF) %>%  
  mutate(sub_lag_end = mean(sub_lag_end, na.rm = TRUE)) %>%  
  unique()  
  
for (uf in dados_geo$Sigla_UF) { # Adicionar as siglas ausentes  
  if (uf %notin% dados_sub$Sigla_UF) {  
    dados_sub[nrow(dados_sub) + 1, ] <- list(uf, NaN)  
  }  
}
```

3. Para finalizar a manipulação dos dados, unimos as duas bases com o comando `left_join`.

```
dados <- dados_geo %>%  
  left_join(dados_sub, by = "Sigla_UF")  
str(dados)
```

```
## Classes 'sf' and 'data.frame':  27 obs. of  7 variables:  
## $ CD_UF      : chr  "11" "12" "13" "14" ...  
## $ NM_UF      : chr  "Rondônia" "Acre" "Amazonas" "Roraima" ...  
## $ SIGLA      : chr  "RO" "AC" "AM" "RR" ...  
## $ NM_REGIAO  : chr  "Norte" "Norte" "Norte" "Norte" ...  
## $ Sigla_UF   : chr  "RO" "AC" "AM" "RR" ...  
## $ sub_lag_end: num  0.0175 0.4131 0.3582 0.4839 0.0383 ...  
## $ geometry   :sfc_MULTIPOLYGON of length 27; first list element: List of 1  
## ..$ :List of 1  
## .. ..$ : num [1:8812, 1:2] -62.9 -62.9 -62.9 -62.8 -62.8 ...  
## ..- attr(*, "class")= chr [1:3] "XY" "MULTIPOLYGON" "sfg"  
## - attr(*, "sf_column")= chr "geometry"  
## - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",...: NA NA NA NA NA NA  
## ..- attr(*, "names")= chr [1:6] "CD_UF" "NM_UF" "SIGLA" "NM_REGIAO" ...
```

Visualização dos resultados

- ▶ Utilizamos três comandos básicos para criar um *choropleth map*: `tm_shape()`, `tm_borders()` e `tm_fill()`. O primeiro deles cria a base geográfica do mapa, o segundo gera as bordas entre os entes representados e o terceiro preenche o mapa de acordo com o a distribuição de valores de alguma variável. Para mais informações sobre cada um deles consulte `?tm_shape`, `?tm_borders`, `?tm_fill`.

```
mapa_inicial <- tm_shape(dados) +  
  tm_borders() +  
  tm_fill("sub_lag_end")
```

O resultado obtido com os parâmetros padrões das funções pode ser visto no próximo slide.

- ▶ Para além dos funções já apresentadas e seus parâmetros, destacam-se `?tm_credits`, `?tm_logo`, `?tm_text`, `?tm_layout`, com as três primeiras servindo para adicionar créditos (ou fonte), logos e textos ao mapa e a última servindo para customizar o «layout da imagem».²

²Por «layout da imagem» entenda-se desde a posição da legenda ou título até o tamanho e família das fontes utilizadas.

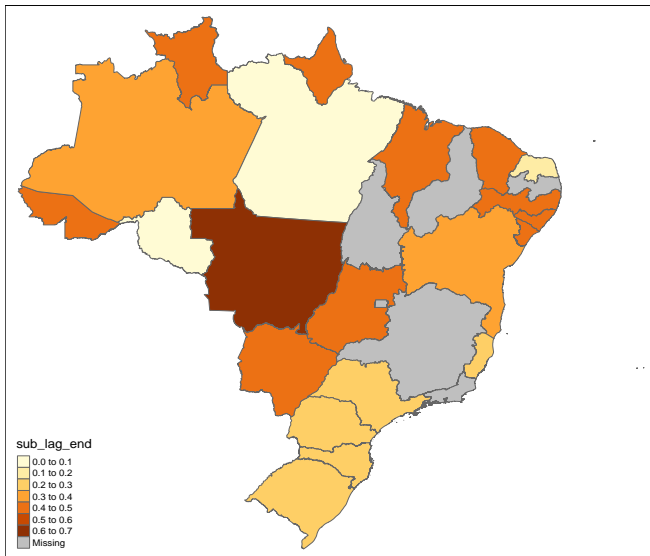


Figura 2: Mapa inicial.

► Visando melhorar o mapa inicial que geramos anteriormente, iremos:

1. Afinar as bordas entre os polígonos:

```
tm_borders(lwd = 0.55)
```

2. Remover o título da legenda, mudar a paleta de cores e associar uma cor e rótulo específicos para os valores ausentes:

```
tm_fill("sub_lag_end", title = "", textNA = "Dados ausentes",  
        colorNA = "lightgrey", palette = "GnBu",  
        labels = c("< 10%", "10% - 20%",  
                    "20% - 30%", "30% - 40%",  
                    "40% - 50%", "50% - 60%",  
                    "60% - 70%"))
```


3. Identificar cada UF no mapa com a sua respectiva sigla:

```
tm_text("Sigla_UF", size = 0.65, fontfamily = "serif",  
        fontface = "bold", col = "black")
```

4. Adicionar uma nota e a fonte dos dados ao mapa:

```
tm_credits("Nota: Média do período jan/2012 - dez/2020.",  
          fontface = "italic", size = 0.7,  
          align = "right")
```

5. Adicionar um título principal ao gráfico, remover o “quadro” envolta do mapa e formatar a fonte da legenda e título:

```
tm_layout(main.title = paste("Distribuição espacial da  
subnotificação", "de estupros de mulheres no Brasil",  
                             sep = "\n"),  
          title.size = 1, main.title.position = "center",  
          fontfamily = "serif", main.title.fontface = "italic",  
          scale = 1, frame = FALSE,  
          legend.title.size = 1.5, legend.text.size = 0.8,  
          legend.outside.position = c("left", "bottom"))
```

- Juntando todos os passos e salvando o mapa ao fim do processo com a função `tmap_save()`, temos:

```
mapa <- tm_shape(dados) +
  tm_borders(lwd = 0.55) +
  tm_fill("sub_lag_end",
    title = "",
    textNA = "Dados ausentes",
    colorNA = "lightgrey", palette = "GnBu",
    labels = c("< 10%", "10% - 20%",
      "20% - 30%", "30% - 40%",
      "40% - 50%", "50% - 60%",
      "60% - 70%")) +
  tm_text("Sigla_UF", size = 0.65, fontfamily = "serif",
    fontface = "bold", col = "black") +
  tm_credits("Nota: Média do período jan/2012 - dez/2020.",
    fontface = "italic", size = 0.7,
    align = "right") +
  tm_credits("Fonte: Elaboração própria a partir dos dados das SSPs.",
    size = 0.8, align = "right") +
  tm_layout(main.title = paste("Distribuição espacial da subnotificação",
    "de estupros de mulheres no Brasil",
    sep = "\n"),
    title.size = 1,
    main.title.position = "center", fontfamily = "serif",
    main.title.fontface = "italic", scale = 1, frame = FALSE,
    legend.title.size = 1.5, legend.text.size = 0.8,
    legend.outside.position = c("left", "bottom"))

#tmap_save(mapa, "figures/mapa_sub_estupros.svg",
#  width = 12, height = 12, units = "cm")
```

Distribuição espacial da subnotificação de estupros de mulheres no Brasil

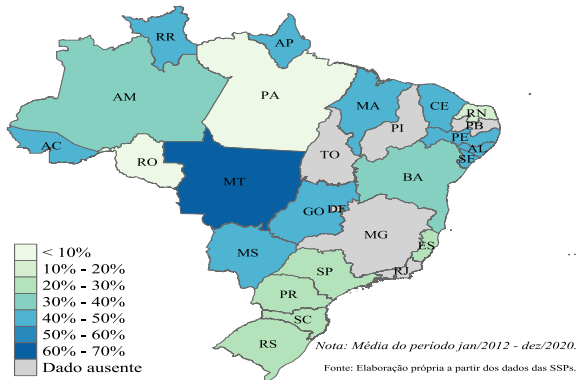


Figura 3: Mapa customizado.

Pacote stargazer

Exemplos em ASCII (.txt)

[cont.] Exemplos em ASCII (.txt)

[cont.] Exemplos em ASCII (.txt)

Exemplos em \LaTeX (.pdf)

[cont.] Exemplos em \LaTeX (.pdf)

[cont.] Exemplos em \LaTeX (.pdf)

Funções

O que são funções na linguagem R?

Como criar funções?

Temas gráficos customizados

O que é um tema gráfico?

Temas básicos disponíveis no ggplot2

Criação de temas customizados: parâmetros importantes

Dúvidas?

Muito obrigado pela atenção e presença. Boa noite!



*Um evento organizado pelo **Laboratório de Inteligência Artificial e Macroeconomia Computacional (LABIMEC)** da UFPB.*