

# Aula 4

Renato Rodrigues Silva

Universidade Federal de Goiás

(2020-12-15)

# Introdução

- O objetivo da aula é fazer a importação, limpeza e formatação dos dados, visualização e alguns testes de hipóteses a respeito dos dados de Covid-19 que estão armazenados no banco de dados SIVEP.
- Para fins didáticos, o objetivo do nosso estudo é verificar a mortalidade de pacientes diagnosticados com Covid 19 no Estado de Goiás. Adicionalmente, vamos estudar de forma simples uma associação entre o desfecho negativo e morbidades.

## Conjunto de dados - SIVEP

- O conjunto de dados analisado na aula de hoje é oriundo do Sistema de Informação da Vigilância Epidemiológica da Gripe (SIVEP-Gripe).
- Temos registros desde o dia 20 de março de 2020 até 19 de outubro de 2020.

# Leitura dos dados

- Vamos começar carregando a biblioteca tidyverse, lendo os dados e dando uma visualizada rápida com o comando

```
library(tidyverse)  
  
dat = read.csv("sivep_atualizado.csv", header= TRUE)  
  
glimpse(dat)
```

# Alguns comentários

- Como podemos perceber, o conjunto de dados é grande. Temos 8340 linhas e 165 colunas.
- Vale lembrar que no banco SIVEP não temos apenas dados de Covid-19, temos outras síndromes e doenças.
- Sendo assim, precisamos saber como filtrar esses dados.

# Filtragem dos dados - Tabela de Frequencia Absoluta

- Inicialmente, vamos conferir a frequência absoluta dos pacientes de acordo com a classificação final
- O campo CLASSI\_FIN tem as seguintes classificações: 1. 1-SRAG por influenza; 2. 2-SRAG por outro vírus respiratório; 3-SRAG por outro agente etiológico; 4-SRAG não especificado e 5-SRAG por COVID-19.
- O comando `table` calcula a frequência absoluta.

```
table( dat$CLASSI_FIN,exclude = NULL)
```

```
##  
##      1      2      3      4      5 <NA>  
##    19     85     32 2336 4988   880
```

- Observe que temos 4988 registros de pacientes classificados como SRAG-Covid19.

# Filtragem dos dados - Tabela de Frequencia Absoluta

- Podemos investigar um pouco mais sobre esses 4988 registros.
- Como exemplo, pode-se investigar qual foi o procedimento de diagnostico da doença
- No campo CRITERIO temos as seguintes opções: 1-Laboratorial; 2-Clinico-Epidemiológico; 3-Clínico e 4-Clínico-Imagem.
- Vamos contar a frequência deles

```
table(dat$CRITERIO, exclude = NULL)
```

```
##  
##      1      2      3      4 <NA>  
## 6768  116   90  209 1157
```

- O interessante é ter 1157 registros em que não se sabe como foi o critério de encerramento do caso.

# Filtragem dos dados - Tabela de Contigência

- Agora vamos cruzar as informações entre os campos CRITERIO e CLASSI\_FIN para verificar inconsistência.
- Fazendo a tabela de contigência entre os campos CRITERIO e CLASSI\_FIN, temos:

```
table(dat$CRITERIO, dat$CLASSI_FIN, exclude = NULL) %>% addmargins()
```

```
##
##      1      2      3      4      5 <NA> Sum
##  1     19     84     23 2063 4579      0 6768
##  2      0      0      3     64     49      0  116
##  3      0      0      3     63     24      0   90
##  4      0      0      2     23    184      0  209
## <NA>      0      1      1    123    152    880 1157
## Sum     19     85     32 2336 4988    880 8340
```

# Filtragem dos dados - Tabela de Contigência

## Interpretações

- Dentre 4988 pacientes classificados como SRAG por Covid19, temos:
- 4579 por critério Laboratorial;
- 49 por critério Clínico-Epidemiológico;
- 24 por critério Clínico;
- 184 por critério Clínico-Imagem;
- **152 não se sabe o critério.**



# Filtragem dos dados - Tabela de Contigência

## Comando xtabs: Sintaxe (frequencia ~ linhas + colunas)

```
xtabs(~CRITERIO + CLASSI_FIN, data=dat, addNA = TRUE) %>% addmargins()
```

##		CLASSI_FIN						
##	CRITERIO	1	2	3	4	5	<NA>	Sum
##	1	19	84	23	2063	4579	0	6768
##	2	0	0	3	64	49	0	116
##	3	0	0	3	63	24	0	90
##	4	0	0	2	23	184	0	209
##	<NA>	0	1	1	123	152	880	1157
##	Sum	19	85	32	2336	4988	880	8340

# Minerando os dados

**Selecionar os pacientes que testaram positivo para presença do vírus SARS-Cov2 via teste RT-PCR e classificados como SRAG-Covid19.**

- O campo PCR\_SARS2 (nome do campo na base de dados) tem a seguinte descrição: Resultado diagnóstico do RT- PCR para (SARS-CoV- 2). Com a seguinte observação: Habilitado se campo 66- Agente etiológico – RT-PCR/outro método por Biologia Molecular: Positivo para outros vírus? = 1.
- O contexto aqui é que a inserção de dados de Covid19 começou a partir desse ano, mas o SIVEP já registrava outros agentes que causavam síndromes gripais e síndrome respiratórias agudas graves há algum tempo.
- O campo que a descrição anterior se refere é o POS\_PCROUT, cuja a descrição é dada por: Resultado da RT- PCR foi positivo para outros vírus que atacam o sistema respiratório.

## Contando registros de testes RT- PCR

```
xtabs( ~POS_PCROUT + PCR_SARS2, data = dat, subset = CLASSI_FIN == 5
```

```
##              PCR_SARS2
## POS_PCROUT    1 <NA>  Sum
##      1      3591   53 3644
##      2         0   20   20
##      9         0    6    6
##     <NA>        0 1941 1941
##      Sum   3591 2020 5611
```

- Logo, temos 3591 registros de pacientes que testaram positivo para SARS-Cov2 e que foram classificados como SRAG por Covid19 com critério laboratorial.

# Contando registros de testes antigênicos

```
xtabs( ~AN_OUTRO + AN_SARS2, data = dat, subset = CLASSI_FIN == 5 &  
      addNA = TRUE) %>% addmargins()
```

```
##           AN_SARS2  
## AN_OUTRO      1 <NA>  Sum  
##      1         8   49   57  
##      <NA>  119 5435 5554  
##      Sum   127 5484 5611
```

- Dentre dos 4579 pacientes classificados como SRAG por Covid19 por critério laboratorial, temos:
- 8 testaram positivo para outro vírus que atacam o sistema respiratório e para SARS-Cov2 (o que pode ser interpretado ter testado positivo somente para SARS Cov2 )
- 119 testaram positivo para SARS Cov2 e provavelmente para outros vírus que atacam o sistema respiratório que já eram monitorados no SIVEP.

Testes Antigênicos:

## Contando registros de testes sorológicos

```
xtabs(~ TP_SOR, data = dat, subset = CLASSI_FIN == 5 & CRITERIO ==
```

```
## TP_SOR
##      0      1      2      3      4
## 4189  337   10   24   19
```

- Dentre dos 4579 registros classificados como SRAG por Covid19 usando critérios laboratorial: 337 testaram positivos para Covid 19 com Teste rápido, 10 com teste Elisa, 24 com Quimiluminescência e 19 com outros testes.

# Filtragem dos dados

**Obs: Baqui et. al. (2020) usaram apenas pacientes diagnosticados com Covid19 e confirmados por meio do teste RT-PCR. Vamos utilizar um critério igual para a primeira filtragem.**

```
datRT_PCR = filter(dat, CLASSI_FIN == 5 & POS_PCROUT == 1 & PCR_SARS  
dim(datRT_PCR)
```

```
## [1] 3706 165
```

- Temos 3706 registros de pacientes classificados como SRAG por Covid19 e confirmados por meio do teste RT-PCR, veja que não necessariamente o critério de encerramento foi laboratorial.

# Filtragem dos dados

## Local de Notificação versus Internação versus Residência do Paciente

- Uma vez que desejamos fazer uma análise do surto de Covid19 no Estado de Goiás, temos um possível conflito entre município da unidade de sentinela onde foi feita a notificação, município de residência do paciente e município da internação.
- Para fins didáticos, eu vou fazer a seguinte filtragem. Serão considerados apenas os registros dos municípios das unidades de sentinela onde foram feitas as notificações dentro do Estado de Goiás.
- Ou seja, registros de cidadãos goianos que foram notificados em outros Estados e/ou no Distrito Federal serão desconsiderados.

# Filtragem dos dados referentes a municípios

- Para fazer essa segunda filtragem, precisamos inicialmente descobrir quais são todos os municípios das unidades de sentinelas onde foram feitas as notificações.
- Uma forma de fazer isso é determinar quais são os distintos elementos do campo ID\_MUNICIP condicionado a Unidade de Federação ser o Estado de Goiás.
- Para saber a unidade de federação devemos consultar o campo SG\_UF\_NOT.



# Filtragem dos dados referentes a municípios

- Para começar a fazer a nossa tarefa, podemos utilizar os seguintes comandos:

```
mn = filter(datRT_PCR, SG_UF_NOT == "GO" ) %>% select(ID_MUNICIP) %>%
```

- Essa linha de comando cria um data.frame com todos os municípios de unidades de sentinela onde foram feitas as notificações.

# Filtragem dos dados referentes a municípios

- Agora podemos fazer a filtragem dos registros que atendem as nossas necessidades.

```
dat_mn = filter(datRT_PCR, ID_MUNICIP %in% mn$ID_MUNICIP)
```

- Para verificar se está tudo ok, basta eu fazer uma tabela de frequência do campo SG\_UF\_NOT. Todos os registros devem pertencer ao estado de Goiás.

```
xtabs(~SG_UF_NOT, data = dat_mn, addNA = TRUE)
```

```
## SG_UF_NOT  
##      GO  
## 3396
```

# Filtragem dos dados - Evolução Caso

- O campo EVOLUCAO tem as seguintes opções: 1-Cura; 2-Óbito 3- Óbito por outras causas 9-Ignorado.
- Inicialmente, vamos calcular a frequência absoluta

```
xtabs(~EVOLUCAO, data = dat_mn, addNA = TRUE)
```

```
## EVOLUCAO
##      1      2      9 <NA>
## 1817 1322    31   226
```

- Observa-se que 226 registros não tem qualquer classificação.
- Uma possível interpretação: tanto o código 9 como não preenchimento (NA) seria que o paciente ainda está em tratamento ou foi transferido para outra unidade de saúde.

Prevalência IPTSP

# Estudando Morbidades

- Agora vamos estudar as morbidades no conjunto de dados.
- Para isso, inicialmente vamos analisar o campo 'FATOR\_RISC'
- Esse campo é descrito como sendo: "Paciente apresenta algum fator de risco", com as seguinte opções: 1-Sim; 2-Não e 9-Ignorado.
- Segundo o nosso entendimento, as morbidades são preenchidas da seguinte forma:
- Caso houver um fator de risco o campo 'FATOR\_RISC' assume 1 e os campos referentes as morbidades são preenchidos.
- Caso o paciente não tiver qualquer fator de risco, o campo 'FATOR\_RISC' assume 2 e os campos referentes as morbidades não são preenchidas.
- Vamos verificar essas suposições: Os nomes dos campos referentes as morbidades são: CARDIOPATI, HEMATOLOGI, HEPATICA, ASMA, DIABETES, NEUROLOGIC, PNEUMOPATI, IMUNODEPRE, RENAL e OBESIDADE.
- Para verificar essa hipótese vamos inicialmente fazer uma tabela de contingência entre os campos FATOR\_RISC e CARDIOPATI.

# Estudando Morbidades

```
xtabs(~ CARDIOPATI, addNA = TRUE, data = dat_mn)
```

```
## CARDIOPATI
##      1      2      9 <NA>
## 1028   689   30 1649
```

```
xtabs(~ FATOR_RISC + CARDIOPATI, addNA = TRUE, data = dat_mn)
```

```
##           CARDIOPATI
## FATOR_RISC      1      2      9 <NA>
##           1 1028   689   30   335
##           2      0      0      0 1314
```

- Podemos interpretar que 1314 registros na verdade não tem qualquer fator de risco e portanto não sofrem de cardiopatia. Cabe ao analista corrigir essa inconsistência.
- Por outro lado, para 365 registros (335 +30=365), não se sabe o diagnostico do paciente quanto à cardiopatias.

# Estudando Morbidades - Corrigindo inconsistências

```
morbs = c("CARDIOPATI", "HEMATOLOGI", "HEPATICA", "ASMA", "DIABETES");  
corr.morbs = function(x){  ifelse(dat_mn$FATOR_RISC ==2, 2, x)    }  
dat_mnF = mutate_at(dat_mn, morbs, corr.morbs)  
xtabs(~ FATOR_RISC + CARDIOPATI, addNA = TRUE, data = dat_mnF)
```

```
##           CARDIOPATI  
## FATOR_RISC      1      2      9 <NA>  
##           1 1028   689   30   335  
##           2      0 1314      0      0
```

```
xtabs(~ CARDIOPATI, addNA = TRUE, data = dat_mnF)
```

```
## CARDIOPATI  
##      1      2      9 <NA>  
## 1028 2003   30   335
```

# Estudando Morbidades - Corrigindo inconsistências

- Deste ponto em diante, não há mais distinção entre os códigos 9 e NA, portanto podemos substituir 9 por NA

```
variaveis = c("EVOLUCAO", "CARDIOPATI", "HEMATOLOGI", "HEPATICA", "ASM")
corr.na = function(x){ ifelse(x == 9, NA, x) }
dat_mnF = mutate_at(dat_mnF, variaveis, corr.na )
xtabs(~ FATOR_RISC + CARDIOPATI, addNA = TRUE, data = dat_mnF)
```

```
##           CARDIOPATI
## FATOR_RISC      1      2 <NA>
##           1 1028   689   365
##           2      0 1314      0
```

```
xtabs(~ CARDIOPATI, addNA = TRUE, data = dat_mnF)
```

```
## CARDIOPATI
##      1      2 <NA>
## 1028 2003   365
```

# Tabelas de Distribuição de Frequencias

- Para fazer calculo de frequencia relativa, podemos usar a função `CrossTable()` do pacote `gmodels`.
- Essa função não é eficiente para fazer tabelas de contingência com dados perdidos, mas por outro lado tem inúmeras vantagens
- Vamos fazer aqui a tabela de frequencia da evolução dos casos



## Tabelas de Frequencia Unidimensional

```
#install.packages("gmodels")  
  
library(gmodels)  
  
CrossTable(dat_mnF$EVOLUCAO,
```

## Tabelas Bidimensionais com Colunas Fixadas

```
CrossTable(dat_mnF$EVOLUCAO, dat  
          digits=3,  
          expected = FALSE, pro  
          prop.r=FALSE,prop.c=T  
          format = "SPSS")
```

## Tabelas Bidimensionais com Linhas Fixadas

```
CrossTable(dat_mnF$EVOLUCAO, dat  
          expected = FALSE, pro  
          digits=3,  
          prop.r=TRUE,prop.c=FA  
          format = "SPSS")
```

## Teste de Independencia

```
CrossTable(dat_mnF$EVOLUCAO, dat  
          expected = TRUE, prop  
          chisq = TRUE,digits=3  
          prop.r=FALSE,prop.c=F  
          format = "SPSS")
```

## Tabelas de Distribuição de Frequências

```
##
##      Cell Contents
## |-----|
## |                      Count |
## |                      Row Percent |
## |-----|
##
## Total Observations in Table:  3139
##
##           |           1 |           2 |
## |-----|-----|
## |       1817 |       1322 |
## |   57.885% |   42.115% |
## |-----|-----|
##
## Number of Missing Observations: 257 (7.567727%)
##
##
```

# Tabelas de Distribuição de Frequencias Bidimensionais - Linhas Fixadas

```
##
##      Cell Contents
## |-----|
## |                      Count |
## |-----|
##
## Total Observations in Table:  2803
##
##      | dat_mnF$CARDIOPATI
## dat_mnF$EVOLUCAO |          1 |          2 | Row Total |
## -----|-----|-----|-----|
##              1 |        484 |        1153 |        1637 |
## -----|-----|-----|-----|
##              2 |        472 |         694 |        1166 |
## -----|-----|-----|-----|
##      Column Total |        956 |        1847 |        2803 |
## -----|-----|-----|-----|
##
##
```

# Tabelas de Distribuição de Frequências Bidimensionais - Colunas Fixadas

```
##
##      Cell Contents
## |-----|
## |                      Count |
## |          Column Percent |
## |-----|
##
## Total Observations in Table:  2803
##
##
##      | dat_mnF$CARDIOPATI
## dat_mnF$EVOLUCAO |          1 |          2 | Row Total |
## -----|-----|-----|-----|
##           1 |          484 |          1153 |          1637 |
##           |          50.628% |          62.426% |
## -----|-----|-----|-----|
##           2 |          472 |          694 |          1166 |
##           |          49.372% |          37.574% |
## -----|-----|-----|-----|
##      Column Total |          956 |          1847 |          2803 |
##           |          34.106% |          65.894% |
## -----|-----|-----|-----|
##
```

	dat_mnF\$CARDIOPATI	Row Total
dat_mnF\$EVOLUCAO	1	2
1	484	1153
	50.628%	62.426%
2	472	694
	49.372%	37.574%
Column Total	956	1847
	34.106%	65.894%

```
##
##      Cell Contents
## |-----|
## |                      Count |
## |-----|
##
## Total Observations in Table:  2803
##
##
##      | dat_mnF$CARDIOPATI
## dat_mnF$EVOLUCAO |          1 |          2 | Row Total |
## -----|-----|-----|-----|
##              1 |        484 |        1153 |        1637 |
## -----|-----|-----|-----|
##              2 |        472 |         694 |        1166 |
## -----|-----|-----|-----|
##      Column Total |        956 |        1847 |        2803 |
## -----|-----|-----|-----|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## -----
## Chi^2 =   36.09217      d.f. =   1      p =   1.882014e-09
##
## Pearson's Chi-squared test with Yates' continuity correction
```

# Estudos de Taxas de Mortalidade

- Apenas por questões didáticas a respeito do uso do software R, vamos estimar taxa de mortalidade devido a SRAG por Covid 19 por 10000 habitantes em cada regional de saúde.
- Os dados da população de cada regional de saúde foram obtidos a partir desse [endereço](#):. Esse [endereço](#)" tem uma dissertação relacionada ao tema.
- Para fazer o estudo de prevalência, o primeiro passo é calcular o número de casos de SRAG por Covid19 por regional de saúde.

```
pp = c(1781530, 849421, 248821, 812707, 286433, 145865, 140321, 11684199333, 167159, 167391, 125947, 420427, 215282, 243991)

dat_tx = group_by(dat_mnF, ID_REGIONA, EVOLUCAO) %>% summarise(obitos)

dat_tx
```

## Estudos de Taxas de Mortalidade

```
## # A tibble: 17 x 4
```

##	ID_REGIONA	obitos	pop	taxa
##	<chr>	<int>	<dbl>	<dbl>
##	1 CENTRAL	650	1781530	3.65
##	2 CENTRO SUL	236	849421	2.78
##	3 ENTORNO NORTE	5	248821	0.201
##	4 ENTORNO SUL	41	812707	0.504
##	5 ESTRADA DE FERRO	49	286433	1.71
##	6 NORDESTE	1	145865	0.069
##	7 NORTE	7	140321	0.499
##	8 OESTE I	2	116841	0.171
##	9 OESTE II	9	112423	0.801
##	10 PIRINEUS	113	488380	2.31
##	11 RIO VERMELHO	12	199333	0.602
##	12 SAO PATRICIO	6	167159	0.359
##	13 SAO PATRICIO II	14	167391	0.836
##	14 SERRA DA MESA	6	125947	0.476
##	15 SUDOESTE I	92	420427	2.19
##	16 SUDOESTE II	27	215282	1.25
##	17 SUL	52	243991	2.13

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- Agora vamos tentar fazer um mapa coroplético que represente a taxa de mortalidade devido a Covid 19 por municípios do estado de Goiás.
- Os dados foram obtidos no seguinte endereço:



# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- Primeiro passo ler os dados referentes a população de Goiás e filtrar apenas os nomes referentes ao estado de Goiás.
- Para facilitar a união dos dois conjuntos de dados vamos instalar a biblioteca `textclean`. Essa biblioteca tem uma função que se chama `replace_non_ascii` que vai nos ajudar a remover todos os caracteres especiais dos nomes. A função `toupper` converte letras maiúsculas em minúsculas.

```
##install.packages("textclean")
```

```
library(textclean)
```

```
dat_IBGE = read.csv("Municipios.csv",header=TRUE,skip=1) %>% filter((  
head(dat_IBGE)
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

##	ID_MUNICIP	POPULACAO
## 1	ABADIA DE GOIAS	8583
## 2	ABADIANIA	19614
## 3	ACREUNA	22182
## 4	ADELANDIA	2517
## 5	AGUA FRIA DE GOIAS	5676
## 6	AGUA LIMPA	1872

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- O próximo passo é calcular a o número de óbitos por municípios.

```
dat_tx2 = group_by(dat_mnF, ID_MUNICIP, EVOLUCAO) %>% summarise(OBITO = sum(OBITO))  
head(dat_tx2)
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- O próximo passo é calcular a o número de óbitos por municípios.

```
## # A tibble: 6 x 2
##   ID_MUNICIP      OBITOS
##   <chr>          <int>
## 1 AGUAS LINDAS DE GOIAS      16
## 2 ALOANDIA                  1
## 3 ANAPOLIS                 111
## 4 APARECIDA DE GOIANIA     221
## 5 APARECIDA DO RIO DOCE      1
## 6 ARAGARCAS                 1
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- Agora precisamos unir os dois conjuntos de dados e para municípios que não se tem registros de óbitos que ele fique nulo. Podemos fazer isso com a função `left_join`.
- Por fim, vamos calcular a taxa de mortalidade por 10000 habitantes.

```
dat_tx2 = left_join(dat_IBGE, dat_tx2) %>%  
  mutate(TX_MORTALIDADE = (OBITOS/POPULACAO)*10^4 ) %>% mutate(TX_MOF
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

##	ID_MUNICIP	TX_MORTALIDADE
## 1	ABADIA DE GOIAS	0
## 2	ABADIANIA	0
## 3	ACREUNA	0
## 4	ADELANDIA	0
## 5	AGUA FRIA DE GOIAS	0
## 6	AGUA LIMPA	0

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- Enfim, passamos para etapa da construção do mapa de fato.
- Para fazer, o mapa vamos carregar algumas dados de coordenadas que já estão prontos no R na biblioteca `geobr`. Além disso, vamos instalar a biblioteca `sf` para manipular dados espaciais.
- Para ler os dados referentes aos municípios, pode-se usar o comando `read_municipality`

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

```
##install.packages("sf")
```

```
library(geobr)
```

```
library(sf)
```

```
#Carregando todos os municipios de Goias
```

```
muniGo = read_municipality(code_muni= "GO", year=2019)
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- Agora, precisamos unir o conjunto de dados com as coordenadas espaciais com os dados de mortalidade, vamos usar o comando `inner_join`.
- Mas antes vamos retirar os caracteres especiais e transformar os demais caracteres em letras maiúsculas. Vamos mudar o nome da coluna `ID_MUNICIP` para `name_muni` no conjunto `dat_tx2`.

```
muniGo = mutate(muniGo, name_muni = replace_non_ascii(name_muni) %>%  
dat_tx2 = rename(dat_tx2, name_muni = ID_MUNICIP)  
muniGo = inner_join(muniGo, dat_tx2)
```



# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

- A seguir mostramos os codigos para fazer o mapa

```
no_axis <- theme(axis.title=element_blank(),
                  axis.text=element_blank(),
                  axis.ticks=element_blank())

ggplot() +
  geom_sf(data=muniGo, aes(fill=TX_MORTALIDADE), color= NA, size=.1) +
  labs(subtitle="Taxa de Mortalidade de Covid 19 no Estado de Go-") +
  scale_fill_distiller(palette = "Spectral", name="Taxa de Morta") +
  theme_minimal() +
  no_axis
```

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

# Mapas coropléticos - Estudos de taxa de mortalidade de Covid 19

## Considerações Finais

- Obviamente, o software R possui mais ferramentas para elaboração de mapas e de análise estatística espacial.
- Maiores detalhes sobre confecção de mapas podem ser visto neste [endereço](#)..

# Referências

- Baqui P, Bica I, Marra V, Ercole A, van der Schaar M. Ethnic and regional variations in hospital mortality from COVID-19 in Brazil: a cross-sectional observational study. Lancet Glob Health 2020; 8:e1018-e1026.
- Faculdade de Medicina. Universidade do Porto. Primeira aula prática curso de Medicina, terceiro ano. Disponível em: [http://cc04-10.med.up.pt/Epidemiologiapraticas/Aula1\\_Nova.pdf](http://cc04-10.med.up.pt/Epidemiologiapraticas/Aula1_Nova.pdf). Acesso em 9 de dezembro de 2020.
- Quijano. F. A. D. Medidas de Frequência em Epidemiologia. Disponível em: [https://edisciplinas.usp.br/pluginfile.php/4660029/mod\\_resource/content/2/Slides-MedidasFrequencia2019.pdf](https://edisciplinas.usp.br/pluginfile.php/4660029/mod_resource/content/2/Slides-MedidasFrequencia2019.pdf). Acesso em 9 de dezembro de 2020.
- Mayer, F. Análise Exploratória de Dados. Disponível em: [http://leg.ufpr.br/~fernandomayer/aulas/ce001e-2016-2/02\\_Analise\\_Exploratoria\\_de\\_Dados.html](http://leg.ufpr.br/~fernandomayer/aulas/ce001e-2016-2/02_Analise_Exploratoria_de_Dados.html). Acesso em 9 de dezembro de 2020.

# Referências

- Pereira, R.H.M.; Gonçalves, C.N.; et. all (2019). geobr: Loads Shapefiles of Official Spatial Data Sets of Brazil. R package. R Found. Stat. Comput., Vienna. <https://CRAN.R-project.org/package=geobr>.
- Pereira, R.H.M.; Gonçalves, C.N.; et. all (2019). An introduction to geobr. Disponível em: [https://cran.r-project.org/web/packages/geobr/vignettes/intro\\_to\\_geobr.html](https://cran.r-project.org/web/packages/geobr/vignettes/intro_to_geobr.html). Acesso em 11 de dezembro de 2020.