

Projeto final da disciplina FLS6397

[Code ▼](#)

Introdução à Análise de Dados, Programação e Visualização para as Ciências Sociais

Beatriz Milz - Nº 7974879

Julho/2020

HTML compilado em: 22 de julho de 2020

Introdução

Este arquivo corresponde ao projeto final da disciplina “FLS6397 - Introdução à Análise de Dados, Programação e Visualização para as Ciências Sociais” (https://jonnyphillips.github.io/Ciencia_de_Dados/). As instruções para a realização do projeto estão disponíveis na página da disciplina (https://jonnyphillips.github.io/Ciencia_de_Dados/projeto.html).

Esse projeto final foi realizado com R (R Core Team 2019) e o pacote `{rmarkdown}` (Xie, Allaire, and Golemund 2018; Allaire et al. 2020).

Objetivos: Explorar os dados de coleta e tratamento de esgoto para os municípios das seguintes Unidades de Gerenciamento de Recursos Hídricos (**UGRHI**): Paraíba do Sul, Litoral Norte, Piracicaba/Capivari/Jundiaí, Alto Tietê, Baixada Santista e Tietê/Sorocaba. Essas UGRHIs foram escolhidas por estarem totalmente ou majoritariamente no território da Macrometrópole Paulista (**MMP**) (considerando a delimitação do DAEE¹).

Dados utilizados

- **Dados de Saneamento:** A Companhia Ambiental do Estado de São Paulo (CETESB)² publica todos os anos o “Relatório de Qualidade das Águas Interiores do Estado de São Paulo” (<https://cetesb.sp.gov.br/aguas-interiores/publicacoes-e-relatorios/>). Desde o relatório referente ao ano de 2016, é publicado o **Apêndice C**, que contém **dados de saneamento por município** do Estado de São Paulo. Os arquivos são disponibilizados em arquivo PDF, e especificamente para o ano de 2016 é um arquivo PDF que não possibilita copiar as informações (https://cetesb.sp.gov.br/aguas-interiores/wp-content/uploads/sites/12/2018/03/Ap%C3%AAndice-C-Dados-do-saneamento-b%C3%A1sico...-munic%C3%ADpios-paulistas-2016_17-04.pdf) (como uma imagem, por exemplo).
- **Dados shapefile dos municípios:** Foi utilizado o pacote `{geobr}` (Pereira and Goncalves 2020), que possibilita acessar dados espaciais oficiais do Brasil.

Pacotes necessários

[Hide](#)

```
library(ggplot2)
library(dplyr)
library(magrittr)
library(geobr)
# install.packages("pdftables")
library(pdftables)
library(readr)
library(stringr)
library(abjutils)
library(ggspatial)
library(tibble)
library(knitr)
library(sf)
```

Referências dos pacotes utilizados: {abjutils} (ABJ 2019), {dplyr} (Wickham, François, et al. 2020), {geobr} (Pereira and Goncalves 2020), {ggplot2} (Wickham, Chang, et al. 2020; Wickham 2016), {ggspatial} (Dunnington 2020), {knitr} (Xie 2020, 2015), {magrittr} (Bache and Wickham 2014), {pdftables} (Persson 2016), {readr} (Wickham, Hester, and Francois 2018), {rmarkdown} (Allaire et al. 2020; Xie, Allaire, and Golemund 2018), {sf} (Pebesma 2020, 2018), {stringr} (Wickham 2019), {tibble} (Müller and Wickham 2020).

Abrir e arrumar as bases brutas

Dados de Saneamento

- O primeiro passo é fazer o download do arquivo PDF referente ao Apêndice C e converter em CSV. Para isso, utilizei o pacote {pdftables} (Persson 2016), que possibilita converter o PDF para CSV diretamente do R, através da API (e omiti a minha chave da API no código):

Hide

```
url <-
  "https://cetesb.sp.gov.br/aguas-interiores/wp-content/uploads/sites/12/2019/10/Ap%C3%A
  Andice-C_Dados-de-Saneamento-por-Munic%C3%ADpio.pdf" # link do apêndice C para o ano de
  2018

# Comentei os passos de download e conversão do arquivo PDF, pois após a primeira vez, n
  ão é preciso repetir essa operação para compilar o arquivo, e deixa o código mais lento.

# Fazer o download do arquivo PDF
# download.file(url,
#               destfile = "dados/apendice_c_2018.pdf",
#               method = "curl")

# Converter o arquivo PDF em CSV. Utilizei a API que obtive no site, mas para compilar,
  omiti a API key.
# pdftables::convert_pdf("dados/apendice_c_2018.pdf",
#                         output_file = "dados/apendice_c_2018.csv",
#                         api_key = "...")
```

- A tabela convertida em `.csv` pode ser acessada neste link (https://beatrizmilz.github.io/2020-FLS6397/projeto_final/dados/apendice_c_2018.csv). O próximo passo é abrir a base:

Hide

```
apendice_c_2018 <-  
  readr::read_csv(  
    "dados/apendice_c_2018.csv",  
    col_names = c(  
      # define o nome das colunas, pois ao converter para pdf fica desconfigurado (em 2  
      linhas)  
      "ugrhi",  
      "municipio",  
      "consessao",  
      "pop_urbana",  
      "atendimento_coleta_porc",  
      "atendimento_tratamento_porc",  
      "eficiencia",  
      "carga_poluidora_potencial",  
      "carga_poluidora_remanescente",  
      "icitem",  
      "corpo_receptor"  
    ),  
    locale = readr::locale(encoding = "ISO-8859-1"),  
    # encoding dos dados  
    skip = 5 # Quantas linhas para pular no CSV antes de começar a ler os dados.  
  )
```

- A base deve conter 645 linhas, referente ao número de municípios no estado de São Paulo³:

Hide

```
nrow(apendice_c_2018) # consulta quantas linhas tem na base
```

```
## [1] 701
```

- A base contém mais linhas do que municípios. O código abaixo retira linhas que só contém NA, e linhas que não tem dados dos municípios:

Hide

```
apendice_c_filtrado <- apendice_c_2018 %>%  
  dplyr::filter_all(dplyr::any_vars(!is.na(.))) %>% # Retira as linhas que apenas contém NA  
  dplyr::filter(!municipio %in% c("Estado de São Paulo", "Município", NA, "MUNICÍPIO"))  
# Filtrar linhas que não contém municípios
```

- Agora a base tem 645 linhas, o que corresponde aos 645 municípios do estado de SP. É importante também verificar o tipo de dados nas colunas:

Hide

```
tibble::glimpse(apendice_c_filtrado)
```

```
## Rows: 645
## Columns: 11
## $ ugrhi <chr> "1", "1", "1", "2", "2", "2", "2", "2...
## $ municipio <chr> "Campos do Jordão", "Santo Antônio do...
## $ consessao <chr> "SABESP", "SABESP", "SABESP", "SAAE",...
## $ pop_urbana <chr> "51440", "4033", "5224", "35604", "18...
## $ atendimento_coleta_porc <chr> "52", "47", "92", "70", "91", "100", ...
## $ atendimento_tratamento_porc <chr> "100", "100", "75", "28", "100", "0",...
## $ eficiencia <chr> "97", "80", "76", "70", "82", NA, "62...
## $ carga_poluidora_potencial <chr> "2.778", "218", "282", "1.923", "101"...
## $ carga_poluidora_remanescente <chr> "1.377", "136", "133", "1.659", "26",...
## $ ictem <chr> "6,06", "4,65", "6,14", "2,56", "8,22...
## $ corpo_receptor <chr> "Rio Capivari, Rio Sapucaí-Guaçu e af..."
```

- Algumas colunas são de dados numéricos mas que foram carregadas como texto, portanto devem ser convertidas:

Hide

```
apendice_c <- apendice_c_filtrado %>%
  dplyr::mutate(
    pop_urbana = as.double(pop_urbana) ,
    atendimento_coleta_porc = as.double(atendimento_coleta_porc),
    atendimento_tratamento_porc = as.double(atendimento_tratamento_porc),
    eficiencia = as.double(eficiencia),

    # As conversões abaixo tem uma etapa a mais, devido à padronização diferente utiliza
    da com o ponto e a vírgula para representar as casas decimais:
    carga_poluidora_potencial = stringr::str_replace_all(carga_poluidora_potencial,
    "\\.", "") %>% as.double(),

    carga_poluidora_remancescente = stringr::str_replace_all(carga_poluidora_remancesce
nte, "\\.", "") %>% as.double(),
    ictem = stringr::str_replace_all(ictem, ",", "\\.") %>% as.double()

  )
```

- Agora podemos observar novamente o tipo de dados nas colunas, e verificar se todos os dados estão no formato ideal para a análise:

Hide

```
tibble::glimpse(apendice_c)
```

```
## Rows: 645
## Columns: 11
## $ ugrhi <chr> "1", "1", "1", "2", "2", "2", "2", "2...
## $ municipio <chr> "Campos do Jordão", "Santo Antônio do...
## $ consessao <chr> "SABESP", "SABESP", "SABESP", "SAAE",...
## $ pop_urbana <dbl> 51440, 4033, 5224, 35604, 1863, 2598,...
## $ atendimento_coleta_porc <dbl> 52, 47, 92, 70, 91, 100, 89, 100, 100...
## $ atendimento_tratamento_porc <dbl> 100, 100, 75, 28, 100, 0, 100, 99, 10...
## $ eficiencia <dbl> 97, 80, 76, 70, 82, NA, 62, 75, 86, 8...
## $ carga_poluidora_potencial <dbl> 2778, 218, 282, 1923, 101, 140, 470, ...
## $ carga_poluidora_remanescente <dbl> 1377, 136, 133, 1659, 26, 140, 210, 1...
## $ ictem <dbl> 6.06, 4.65, 6.14, 2.56, 8.22, 1.50, 6...
## $ corpo_receptor <chr> "Rio Capivari, Rio Sapucaí-Guaçu e af...
```

- Agora a base está pronta para uso!

Dados shapefile dos município

Os dados foram obtidos com o pacote `geobr`:

Hide

```
municipios_sp <- geobr::read_municipality("SP", 2018)
```

Lista de UGRHIS

- Eu criei manualmente uma tibble com o número e nome das UGRHIs que farão parte da análise:

Hide

```
ugrhis <- tibble::tibble(
  ugrhi = c("2",
            "3",
            "5",
            "6",
            "7",
            "10"),
  nome_ugrhi = c(
    "Paraíba do Sul",
    "Litoral Norte",
    "Piracicaba/Capivari/Jundiaí",
    "Alto Tietê",
    "Baixada Santista",
    "Tietê/Sorocaba"
  )
)
```

Unir as bases !

- A base da CETESB não possui o código de município do IBGE (o ideal para fazer o Join). Neste caso, podemos usar o nome do município, porém é preciso padronizar os nomes em relação à maiúsculas/minúsculas, acentos, presença de traços, entre outros. A maior diferença encontrada foi na

grafia do nome do município “São Luiz do Paraitinga”: segundo o site da Assembléia Legislativa do Estado de São Paulo, e o site do município, Luiz é escrito com Z, porém a base da CETESB utiliza a forma incorreta: “São Luís do Paraitinga”. Essas inconsistências foram corrigidas com código abaixo:

Hide

```
municipios_sp_limpo <-
  municipios_sp %>% dplyr::mutate(
    nome_muni = stringr::str_to_lower(name_muni),
    nome_muni = stringr::str_replace_all(nome_muni, "-", " "),
    nome_muni = abjutils::rm_accent(nome_muni)
  )

apendice_c_limpo <- apendice_c %>% dplyr::mutate(
  nome_muni = dplyr::case_when(
    municipio == "São Luís do Paraitinga" ~
      "São Luiz do Paraitinga",
    TRUE ~ municipio
  ),
  nome_muni = stringr::str_to_lower(nome_muni),
  nome_muni = stringr::str_replace_all(nome_muni, "-", " "),
  nome_muni = abjutils::rm_accent(nome_muni))
```

- Após arrumar a base, podemos unir com o Join:

Hide

```
apendice_c_geo <-
  dplyr::full_join(municipios_sp_limpo, apendice_c_limpo) %>%
  dplyr::left_join(ugrhis)

apendice_c_geo %>% nrow() # Confirmando se a nova base tem o número de municípios do estado.
```

```
## [1] 645
```

- Ao unir as bases, temos colunas duplicadas ou desnecessárias, então é interessante removê-las. Após este procedimento, a base será filtrada para que apenas municípios que fazem parte das UGRHIs analisadas estejam na tibble gerada.

Além disso, o valor de porcentagem de atendimento de tratamento de esgoto é um valor de porcentagem em relação ao volume de esgoto **coletado**. Por exemplo, o município de Bertioga, segundo os dados da CETESB para 2018, apresenta uma porcentagem de coleta de apenas 34 % do esgoto gerado, e uma porcentagem de 100 % do esgoto tratado. Isso significa que 100 % do esgoto coletado é tratado, e não mostra a porcentagem de todo esgoto gerado que foi tratado. Para isso, criei também uma coluna (chamada `porc_real_trat`) onde é feito esse cálculo (utilizando a função `mutate`).

Hide

```
saneamento <- apendice_c_geo %>%
  dplyr::select(-nome_muni, -municipio, -code_state) %>%
  dplyr::filter(ugrhi %in% ugrhis$ugrhi) %>%
  dplyr::mutate(porc_real_trat = atendimento_tratamento_porc * atendimento_coleta_porc /
100) # calcula o número real de porcentagem de tratamento de esgoto.
```

A base final que usaremos na análise contém dados de 171 municípios, que fazem parte de 6UGRHs diferentes. A soma da população urbana destes municípios é de 32.79 milhões de habitantes, o que corresponde à 75.1 % da população urbana do Estado de São Paulo (segundo os dados da base completa utilizada nessa análise).

Explorando os dados

Agora temos uma base “limpa”, e podemos explorá-la.

- Dentre os municípios que fazem parte da análise, quais são os municípios com menor porcentagem de atendimento de coleta de esgoto? E de que UGRHI fazem parte?

Hide

```
saneamento %>%
  arrange(atendimento_coleta_porc) %>%
  st_drop_geometry() %>%
  slice(1:10) %>%
  select(name_muni,
         nome_ugrhi,
         consessao,
         atendimento_coleta_porc,
         porc_real_trat) %>%
  DT::datatable(
    colnames = c(
      "Município",
      "UGRHI",
      "Consessão",
      "Atendimento de coleta de esgoto (%)",
      "Tratamento do total de esgoto gerado (%)"
    ),
    caption = "Tabela: 10 municípios com menor porcentagem de atendimento de coleta de e
sgoto, no ano de 2018, segundo a CETESB (2019)."
```

Show **10** entries

Search:

Tabela: 10 municípios com menor porcentagem de atendimento de coleta de esgoto, no ano de 2018, segundo a CETESB (2019).

	Município	UGRHI	Consessão	Atendimento de coleta de esgoto (%)	Tratamento do total de esgoto gerado (%)
1	Nazaré Paulista	Piracicaba/Capivari/Jundiaí	SABESP	14	14

	Município	UGRHI	Consessão	Atendimento de coleta de esgoto (%)	Tratamento do total de esgoto gerado (%)
2	Jarinu	Piracicaba/Capivari/Jundiaí	SABESP	19	19
3	Mairiporã	Alto Tietê	SABESP	25	17.75
4	Itapecerica Da Serra	Alto Tietê	SABESP	29	28.42
5	Igaratá	Paraíba do Sul	SABESP	31	31
6	Vargem Grande Paulista	Tietê/Sorocaba	SABESP	32	8.96
7	Bertioga	Baixada Santista	SABESP	34	34
8	Ilhabela	Litoral Norte	SABESP	35	1.4
9	Embu-Guaçu	Alto Tietê	SABESP	37	37
10	Santana De Parnaíba	Alto Tietê	SABESP	38	8.36

Showing 1 to 10 of 10 entries

Previous 1 Next

Dentre os municípios avaliados, os 10 municípios com menor percentual de coleta de esgoto tem esse serviço concessionado pela SABESP (Companhia de Saneamento Básico do Estado de São Paulo).

Visualizando os dados

O código abaixo é referente ao estilo do mapa, que aplicarei em todos os mapas seguintes.

Hide

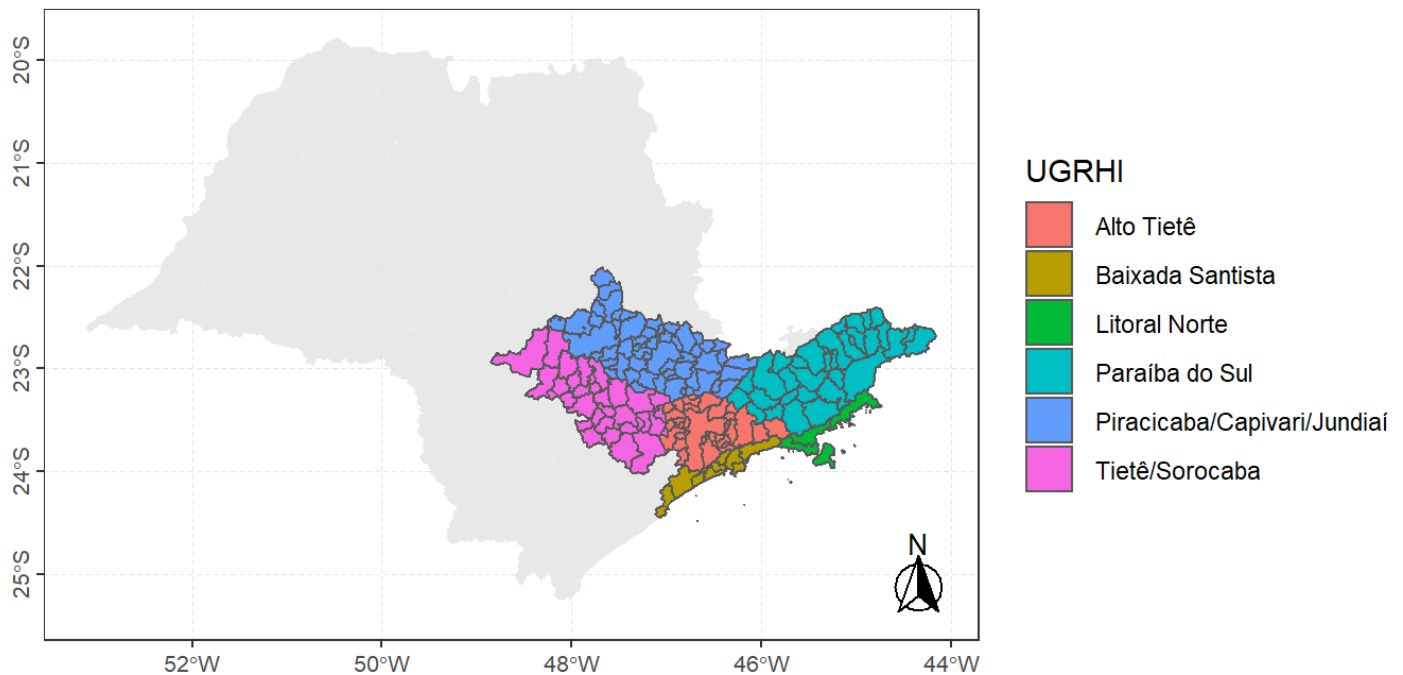

```
tema_mapa <- theme_bw() +
  theme(
    axis.text.y = element_text(
      angle = 90,
      hjust = 0.5,
      size = 8
    ),
    axis.text.x = element_text(size = 8),
    axis.title.y = element_text(size = rel(0.8)),
    axis.title.x = element_text(size = rel(0.8))
  ) +
  theme(
    panel.grid.major = element_line(
      color = gray(0.9),
      linetype = "dashed",
      size = 0.1
    ),
    panel.background = element_rect(fill = "white") +
      annotation_scale(location = "br", width_hint = 0.30)
  )
```

O mapa abaixo apresenta os municípios que fazem parte da análise, segundo a UGRHI, e a localização destes municípios no Estado de São Paulo:

Hide

```
saneamento %>%
  ggplot() +
  geom_sf(data = apendice_c_geo,
    alpha = .9,
    color = NA) +
  geom_sf(aes(fill = nome_ugrhi)) +
  labs(fill = "UGRHI",
    title = "Municípios que fazem parte da análise, segundo a UGRHI") +
  annotation_north_arrow(
    location = "br",
    which_north = "true",
    height = unit(1, "cm"),
    width = unit(1, "cm"),
    pad_x = unit(0.1, "in"),
    pad_y = unit(0.1, "in"),
    style = north_arrow_fancy_orienteering
  ) +
  tema_mapa
```

Municípios que fazem parte da análise, segundo a UGRHI



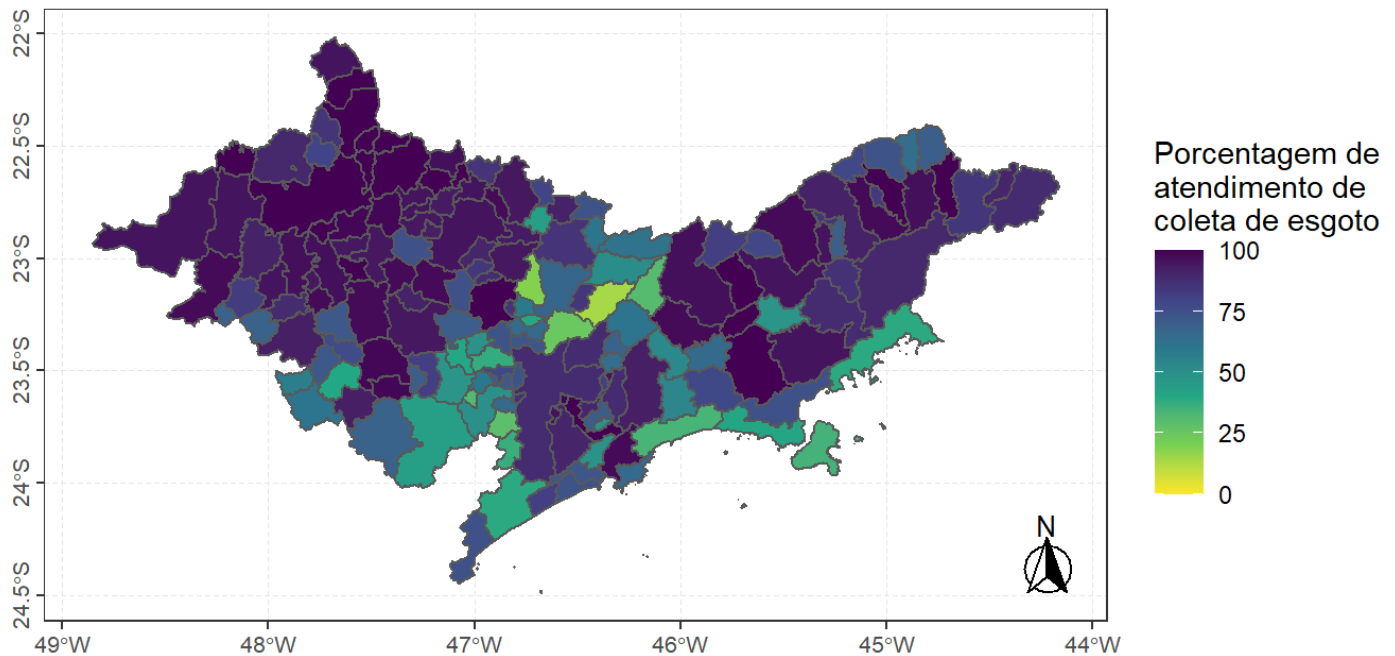
O mapa abaixo apresenta a porcentagem de atendimento de coleta de esgoto, por município:

Hide

```
saneamento %>%
  ggplot() +
  geom_sf(aes(fill = atendimento_coleta_porc)) +
  scale_fill_viridis_c(direction = -1, limits = c(0, 100)) +
  theme_bw() +
  labs(fill = "Porcentagem de \natendimento de \ncoleta de esgoto",
       title = "Porcentagem de atendimento de coleta de esgoto, por município",
       subtitle = "Dados da CETESB, para o ano de 2018.") +
  tema_mapa +
  annotation_north_arrow(
    location = "br",
    which_north = "true",
    height = unit(1, "cm"),
    width = unit(1, "cm"),
    pad_x = unit(0.1, "in"),
    pad_y = unit(0.1, "in"),
    style = north_arrow_fancy_orienteering
  )
```

Porcentagem de atendimento de coleta de esgoto, por município

Dados da CETESB, para o ano de 2018.



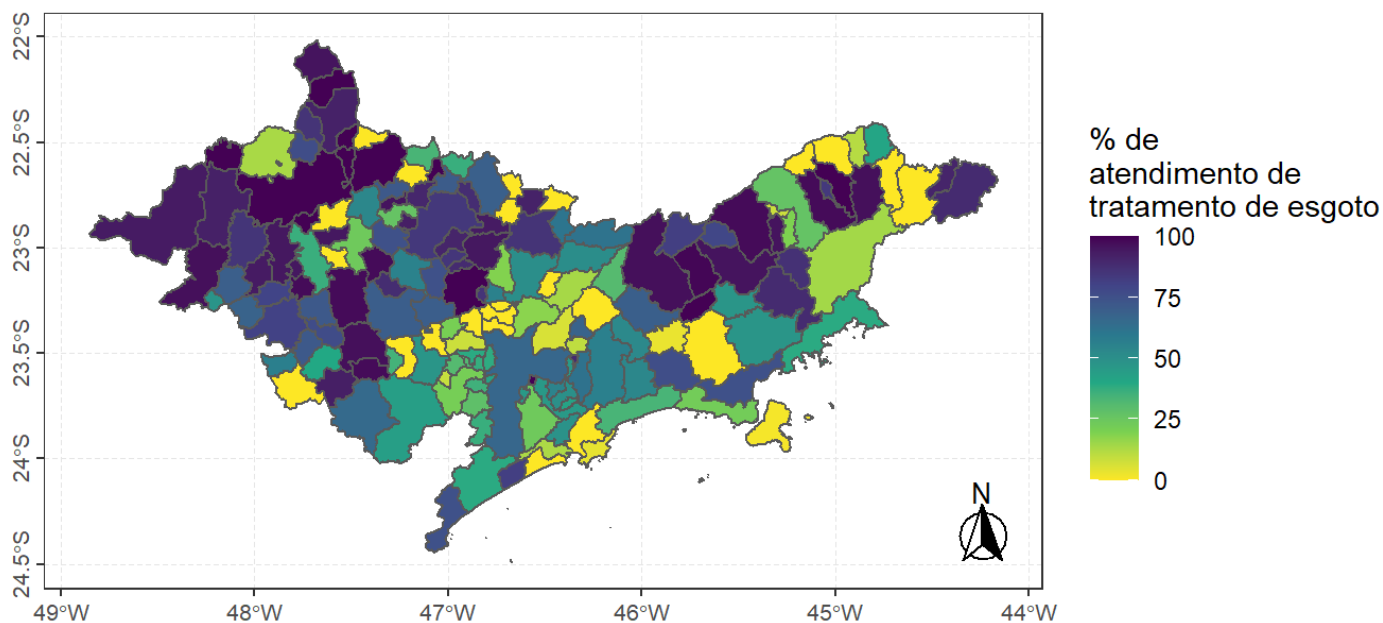
O mapa abaixo apresenta a porcentagem de atendimento de tratamento de esgoto, considerando o total de esgoto coletado, por município:

Hide

```
saneamento %>%
  ggplot() +
  geom_sf(aes(fill = porc_real_trat)) +
  scale_fill_viridis_c(direction = -1, limits = c(0, 100)) +
  theme_bw() +
  labs(fill = "% de \natendimento de \ntratamento de esgoto ",
       title = "Porcentagem de atendimento de tratamento de esgoto, por município",
       subtitle = "Calculado a partir de dados da CETESB, para o ano de 2018.") +
  tema_mapa +
  annotation_north_arrow(
    location = "br",
    which_north = "true",
    height = unit(1, "cm"),
    width = unit(1, "cm"),
    pad_x = unit(0.1, "in"),
    pad_y = unit(0.1, "in"),
    style = north_arrow_fancy_orienteering
  )
```

Porcentagem de atendimento de tratamento de esgoto, por município

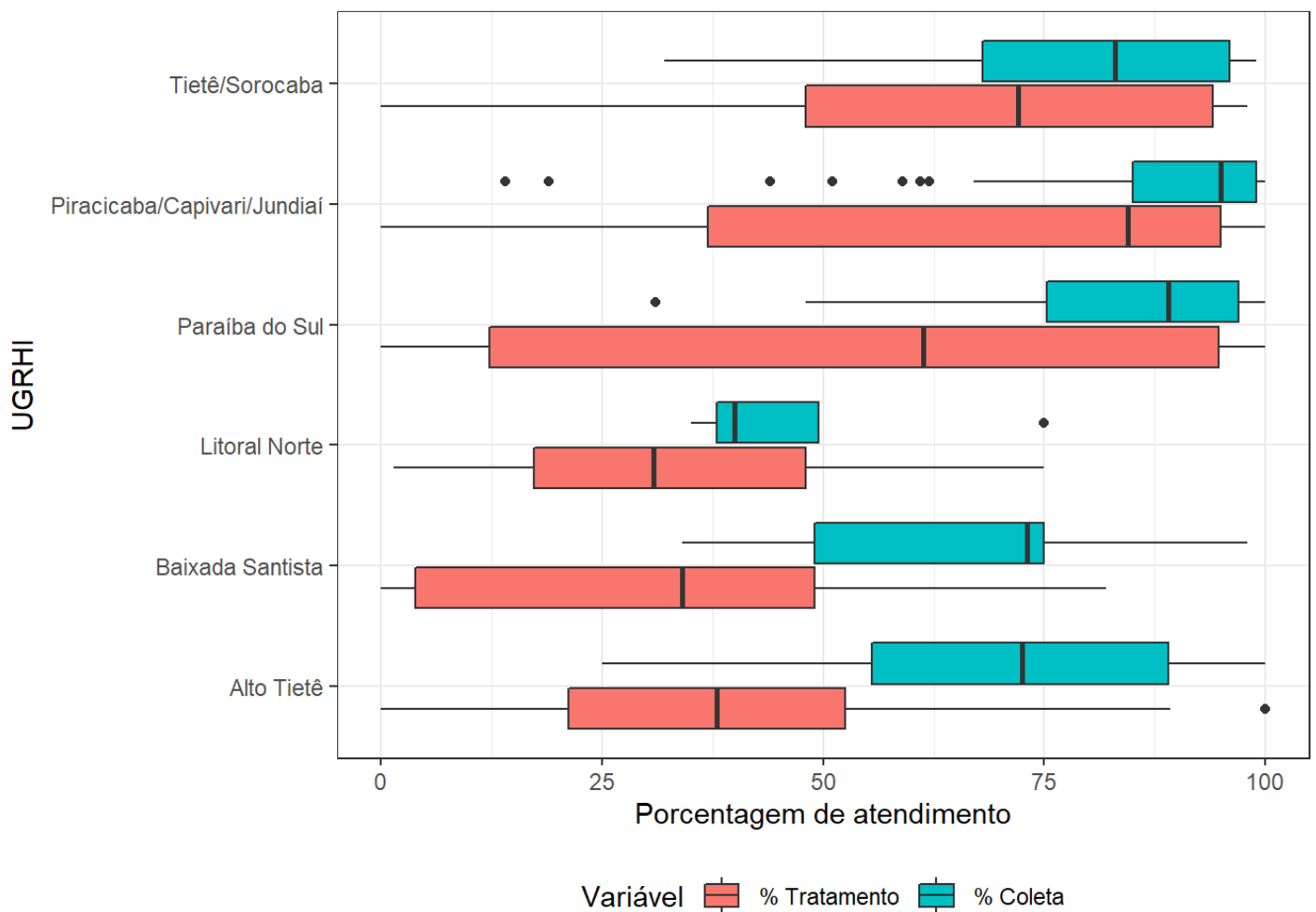
Calculado a partir de dados da CETESB, para o ano de 2018.



Os gráficos bloxplot abaixo apresentam os valores de coleta e de tratamento de esgoto, agrupados pela UGRHI. Com o boxplot, fica mais fácil de visualizar com os dados se distribuem por UGRHI. É possível verificar que a UGRHI Litoral Norte necessita de uma melhoria no sistema de coleta (e consequentemente de tratamento) de esgotos, pois é a UGRHI com os menores percentuais de coleta de esgoto.

Hide

```
saneamento %>%
  tidyr::pivot_longer(cols = c(atendimento_coleta_por, porc_real_trat),
    names_to = "tipo") %>%
  mutate(tipo = forcats::fct_reorder(tipo, value)) %>%
  ggplot() +
  geom_boxplot(aes(x = nome_ugrhi, y = value, fill = tipo)) +
  scale_fill_discrete(labels = c("% Tratamento", "% Coleta")) +
  coord_flip() +
  theme_bw() +
  scale_y_continuous(limits = c(0, 100)) +
  labs(x = "UGRHI",
    y = "Porcentagem de atendimento",
    fill = "Variável") +
  theme(legend.position = "bottom")
```



Recomendações para a base da CETESB

- Adicionar uma explicação do que significa os dados de cada coluna.
- Seria melhor disponibilizar também o arquivo .csv . Não disponibilizar PDFs digitalizados (como o para o ano de 2016).
- Seria útil adicionar, para os próximos relatórios, a coluna de código IBGE do município.
- Verificar a grafia do nome dos municípios (está inconsistente com a lista de municípios da Assembleia Legislativa do Estado de São Paulo⁴): São Luiz do Paraitinga, Biritiba Mirim, Itaoca.

Informações sobre a sessão do R e RStudio

- As informações abaixo são interessantes para registrar a versão do R utilizada, versões de pacotes, entre outros.

Hide

```
sessioninfo::session_info()
```

```
## - Session info -----
## setting value
## version R version 3.6.2 (2019-12-12)
## os Windows 10 x64
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate Portuguese_Brazil.1252
## ctype Portuguese_Brazil.1252
## tz America/Sao_Paulo
## date 2020-07-22
##
## - Packages -----
## package * version date lib source
## abjutils * 0.2.4 2020-06-26 [1] Github (abjur/abjutils@cc3ec0c)
## assertthat 0.2.1 2019-03-21 [1] CRAN (R 3.6.2)
## class 7.3-15 2019-01-01 [2] CRAN (R 3.6.2)
## classInt 0.4-3 2020-04-07 [1] CRAN (R 3.6.2)
## cli 2.0.2 2020-02-28 [1] CRAN (R 3.6.3)
## colorspace 1.4-1 2019-03-18 [1] CRAN (R 3.6.1)
## crayon 1.3.4 2017-09-16 [1] CRAN (R 3.6.2)
## crosstalk 1.1.0.1 2020-03-13 [1] CRAN (R 3.6.3)
## curl 4.3 2019-12-02 [1] CRAN (R 3.6.2)
## DBI 1.1.0 2019-12-15 [1] CRAN (R 3.6.2)
## digest 0.6.25 2020-02-23 [1] CRAN (R 3.6.2)
## dplyr * 1.0.0 2020-05-29 [1] CRAN (R 3.6.3)
## DT 0.14 2020-06-24 [1] CRAN (R 3.6.3)
## e1071 1.7-3 2019-11-26 [1] CRAN (R 3.6.2)
## ellipsis 0.3.1 2020-05-15 [1] CRAN (R 3.6.3)
## evaluate 0.14 2019-05-28 [1] CRAN (R 3.6.2)
## fansi 0.4.1 2020-01-08 [1] CRAN (R 3.6.2)
## farver 2.0.3 2020-01-16 [1] CRAN (R 3.6.2)
## forcats 0.5.0 2020-03-01 [1] CRAN (R 3.6.3)
## generics 0.0.2 2018-11-29 [1] CRAN (R 3.6.2)
## geobr * 1.3 2020-03-29 [1] CRAN (R 3.6.3)
## ggplot2 * 3.3.2 2020-06-19 [1] CRAN (R 3.6.2)
## ggspatial * 1.1.3 2020-06-09 [1] CRAN (R 3.6.3)
## glue 1.4.1 2020-05-13 [1] CRAN (R 3.6.3)
## gtable 0.3.0 2019-03-25 [1] CRAN (R 3.6.2)
## hms 0.5.3 2020-01-08 [1] CRAN (R 3.6.2)
## htmltools 0.5.0 2020-06-16 [1] CRAN (R 3.6.3)
## htmlwidgets 1.5.1 2019-10-08 [1] CRAN (R 3.6.2)
## http 1.4.1 2019-08-05 [1] CRAN (R 3.6.2)
## jsonlite 1.7.0 2020-06-25 [1] CRAN (R 3.6.3)
## KernSmooth 2.23-16 2019-10-15 [2] CRAN (R 3.6.2)
## knitr * 1.29 2020-06-23 [1] CRAN (R 3.6.3)
## labeling 0.3 2014-08-23 [1] CRAN (R 3.6.0)
## lifecycle 0.2.0 2020-03-06 [1] CRAN (R 3.6.3)
## magrittr * 1.5 2014-11-22 [1] CRAN (R 3.6.2)
## munsell 0.5.0 2018-06-12 [1] CRAN (R 3.6.2)
## pdftables * 0.1 2016-02-15 [1] CRAN (R 3.6.3)
## pillar 1.4.4 2020-05-05 [1] CRAN (R 3.6.3)
## pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 3.6.2)
```

```
## purrr      0.3.4    2020-04-17 [1] CRAN (R 3.6.3)
## R6         2.4.1    2019-11-12 [1] CRAN (R 3.6.2)
## Rcpp       1.0.4.6  2020-04-09 [1] CRAN (R 3.6.3)
## readr      * 1.3.1    2018-12-21 [1] CRAN (R 3.6.2)
## rlang      0.4.6    2020-05-02 [1] CRAN (R 3.6.3)
## rmarkdown  2.3      2020-06-18 [1] CRAN (R 3.6.3)
## scales     1.1.1    2020-05-11 [1] CRAN (R 3.6.3)
## sessioninfo 1.1.1    2018-11-05 [1] CRAN (R 3.6.2)
## sf         * 0.9-4    2020-06-13 [1] CRAN (R 3.6.3)
## stringi    1.4.6    2020-02-17 [1] CRAN (R 3.6.2)
## stringr    * 1.4.0    2019-02-10 [1] CRAN (R 3.6.2)
## tibble     * 3.0.1    2020-04-20 [1] CRAN (R 3.6.2)
## tidyr      1.1.0    2020-05-20 [1] CRAN (R 3.6.3)
## tidyselect 1.1.0    2020-05-11 [1] CRAN (R 3.6.3)
## units      0.6-7    2020-06-13 [1] CRAN (R 3.6.3)
## utf8       1.1.4    2018-05-24 [1] CRAN (R 3.6.2)
## vctrs      0.3.1    2020-06-05 [1] CRAN (R 3.6.3)
## viridisLite 0.3.0    2018-02-01 [1] CRAN (R 3.6.2)
## withr      2.2.0    2020-04-20 [1] CRAN (R 3.6.3)
## xfun       0.15     2020-06-21 [1] CRAN (R 3.6.3)
## yaml       2.2.1    2020-02-01 [1] CRAN (R 3.6.2)
##
## [1] C:/Users/beatr/Documents/R/win-library/3.6
## [2] C:/Program Files/R/R-3.6.2/library
```

Referências

ABJ. 2019. *Abjutils: Useful Tools for Jurimetrical Analysis Used by the Brazilian Jurimetrics Association*. <https://github.com/abjur/abjutils> (<https://github.com/abjur/abjutils>).

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020. *Rmarkdown: Dynamic Documents for R*. <https://CRAN.R-project.org/package=rmarkdown> (<https://CRAN.R-project.org/package=rmarkdown>).

Bache, Stefan Milton, and Hadley Wickham. 2014. *Magrittr: A Forward-Pipe Operator for R*. <https://CRAN.R-project.org/package=magrittr> (<https://CRAN.R-project.org/package=magrittr>).

Dunnington, Dewey. 2020. *Ggspatial: Spatial Data Framework for Ggplot2*. <https://CRAN.R-project.org/package=ggspatial> (<https://CRAN.R-project.org/package=ggspatial>).

Müller, Kirill, and Hadley Wickham. 2020. *Tibble: Simple Data Frames*. <https://CRAN.R-project.org/package=tibble> (<https://CRAN.R-project.org/package=tibble>).

Pebesma, Edzer. 2018. "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal* 10 (1): 439–46. <https://doi.org/10.32614/RJ-2018-009> (<https://doi.org/10.32614/RJ-2018-009>).

———. 2020. *Sf: Simple Features for R*. <https://CRAN.R-project.org/package=sf> (<https://CRAN.R-project.org/package=sf>).

Pereira, Rafael H. M., and Caio Nogueira Goncalves. 2020. *Geobr: Loads Shapefiles of Official Spatial Data Sets of Brazil*. <https://CRAN.R-project.org/package=geobr> (<https://CRAN.R-project.org/package=geobr>).

Persson, Eric. 2016. *Pdftables: Programmatic Conversion of Pdf Tables*. <https://CRAN.R-project.org/package=pdftables> (<https://CRAN.R-project.org/package=pdftables>).

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/> (<https://www.R-project.org/>).

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org> (<https://ggplot2.tidyverse.org>).

———. 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. <https://CRAN.R-project.org/package=stringr> (<https://CRAN.R-project.org/package=stringr>).

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2> (<https://CRAN.R-project.org/package=ggplot2>).

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2020. *Dplyr: A Grammar of Data Manipulation*. <https://CRAN.R-project.org/package=dplyr> (<https://CRAN.R-project.org/package=dplyr>).

Wickham, Hadley, Jim Hester, and Romain Francois. 2018. *Readr: Read Rectangular Text Data*. <https://CRAN.R-project.org/package=readr> (<https://CRAN.R-project.org/package=readr>).

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <https://yihui.org/knitr/> (<https://yihui.org/knitr/>).

———. 2020. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. <https://CRAN.R-project.org/package=knitr> (<https://CRAN.R-project.org/package=knitr>).

Xie, Yihui, J. J. Allaire, and Garrett Grolemund. 2018. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman; Hall/CRC. <https://bookdown.org/yihui/rmarkdown> (<https://bookdown.org/yihui/rmarkdown>).

-
1. Plano Diretor de Aproveitamento dos Recursos Hídricos para a Macrometrópole Paulista - <http://www.daee.sp.gov.br/> (<http://www.daee.sp.gov.br/>)↵
 2. <https://cetesb.sp.gov.br/> (<https://cetesb.sp.gov.br/>)↵
 3. <https://www.al.sp.gov.br/documentacao/municipios-paulistas/> (<https://www.al.sp.gov.br/documentacao/municipios-paulistas/>)↵
 4. <https://www.al.sp.gov.br/documentacao/municipios-paulistas/> (<https://www.al.sp.gov.br/documentacao/municipios-paulistas/>)↵