

Acessando dados criminais com BrazilCrime

Giovanni Vargette, Igor Laltuf, Marcelo Justus

Introdução

O pacote BrazilCrime permite acessar dados públicos de segurança do sistema SINESP/VDE com filtros por localidade, tipologia criminal, categoria e período.

A função central, `get_sinesp_vde_data()`, permite obter dados por UF, município, categoria (arma de fogo, bombeiros, desaparecidos/localizados, drogas, mandado de prisão cumprido, ocorrências, profissionais de segurança e vítimas), para 28 tipologias de evento (Apreensão de Cocaína, Apreensão de Maconha, Arma de Fogo Apreendida, Atendimento pré-hospitalar, Busca e salvamento, Combate a incêndios, Emissão de Alvarás de licença, Estupro, Feminicídio, Homicídio doloso, Lesão corporal seguida de morte, Mandado de prisão cumprido, Morte de Agente do Estado, Morte no trânsito ou em decorrência dele (exceto homicídio doloso), Morte por intervenção de Agente do Estado, Mortes a esclarecer (sem indício de crime), Pessoa Desaparecida, Pessoa Localizada, Realização de vistorias, Roubo a instituição financeira, Roubo de carga, Roubo de veículo, Roubo seguido de morte (latrocínio), Suicídio, Suicídio de Agente do Estado, Tentativa de homicídio, Tráfico de drogas) com granularidade mensal ou anual.

Exemplos de Uso

Exemplo 1 - Visualização simples

```
dados <- get_sinesp_vde_data()
#> Query completed.

kable(head(dados), format = "latex", booktabs = TRUE, caption = "Visualização dos primeiros
← registros") |>
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 1: Visualização dos primeiros registros

uf	municipio	ano	mes	categoria	evento	agente	arma	faixa_etaria	feminino	masculino	nao_informado	total	total_peso	total_vitima	data	
AC	ACRELANDIA	2015	10	Mandado de prisao cumprido	Mandado de prisão cumprido	NA	NA	NA	NA	NA	NA	NA	NA	NA	2015-01-01	
AC	ACRELANDIA	2015	01	vítimas	Homicídio	NA	NA	NA	0	0	0	NA	NA	NA	0	2015-01-01
AC	ACRELANDIA	2015	01	vítimas	Homicídio doloso	NA	NA	NA	0	0	0	NA	NA	NA	0	2015-01-01
AC	ACRELANDIA	2015	01	vítimas	Lesão corporal seguida de morte	NA	NA	NA	0	0	0	NA	NA	NA	0	2015-01-01
AC	ACRELANDIA	2015	01	vítimas	Morte no trânsito ou em decorrência dele (exceto homicídio doloso)	NA	NA	NA	0	0	0	NA	NA	NA	0	2015-01-01
AC	ACRELANDIA	2015	01	vítimas	Mortes a esclarecer (sem indício de crime)	NA	NA	NA	0	0	0	NA	NA	NA	0	2015-01-01

Exemplo 2 - Filtrar por Estado, cidade e tipologia

```

recife <- get_sinesp_vde_data(state = "PE", city = "Recife", typology = "Homicídio doloso", year =
  ↵ 2020:2022)
#> Query completed.

kable(head(recife), format = "latex", booktabs = TRUE, caption = "Visualização dos primeiros
  ↵ registros") |>
  kable_styling(latex_options = c("scale_down", "hold_position"))

```

Table 2: Visualização dos primeiros registros

uf	municipio	ano	mes	categoria	evento	agente	arma	faixa_etaria	feminino	masculino	nao_informado	total	total_peso	total_vitima	data
PE	RECIFE	2020	01	vítimas	Homicídio doloso	NA	NA	NA	1	42	0	NA	NA	43	2020-01-01
PE	RECIFE	2020	02	vítimas	Homicídio doloso	NA	NA	NA	2	42	0	NA	NA	44	2020-02-01
PE	RECIFE	2020	03	vítimas	Homicídio doloso	NA	NA	NA	2	37	0	NA	NA	39	2020-03-01
PE	RECIFE	2020	04	vítimas	Homicídio doloso	NA	NA	NA	2	45	0	NA	NA	47	2020-04-01
PE	RECIFE	2020	05	vítimas	Homicídio doloso	NA	NA	NA	2	46	0	NA	NA	48	2020-05-01
PE	RECIFE	2020	06	vítimas	Homicídio doloso	NA	NA	NA	0	37	0	NA	NA	37	2020-06-01

Exemplo 3 - Dados agregados por ano, para Estado e categoria específica

```

sp_anuais <- get_sinesp_vde_data(state = "SP", category = "ocorrencias", granularity = "year")
#> Query completed.

kable(head(sp_anuais), format = "latex", booktabs = TRUE, caption = "Visualização dos primeiros
  ↵ registros") |>
  kable_styling(latex_options = c("scale_down", "hold_position"))

```

Table 3: Visualização dos primeiros registros

uf	municipio	ano	mes	categoria	evento	agente	arma	faixa_etaria	feminino	masculino	nao_informado	total	total_peso	total_vitima	data
SP	NÃO INFORMADO	2015	01	ocorrencias	Furto de veículo	NA	NA	NA	NA	NA	NA	9563	NA	NA	2015-01-01
SP	NÃO INFORMADO	2015	01	ocorrencias	Roubo a instituição financeira	NA	NA	NA	NA	NA	NA	17	NA	NA	2015-01-01
SP	NÃO INFORMADO	2015	01	ocorrencias	Roubo de carga	NA	NA	NA	NA	NA	NA	782	NA	NA	2015-01-01
SP	NÃO INFORMADO	2015	01	ocorrencias	Roubo de veículo	NA	NA	NA	NA	NA	NA	7490	NA	NA	2015-01-01
SP	NÃO INFORMADO	2015	02	ocorrencias	Furto de veículo	NA	NA	NA	NA	NA	NA	8966	NA	NA	2015-02-01
SP	NÃO INFORMADO	2015	02	ocorrencias	Roubo a instituição financeira	NA	NA	NA	NA	NA	NA	17	NA	NA	2015-02-01

Exemplo 4 - Visualização rápida

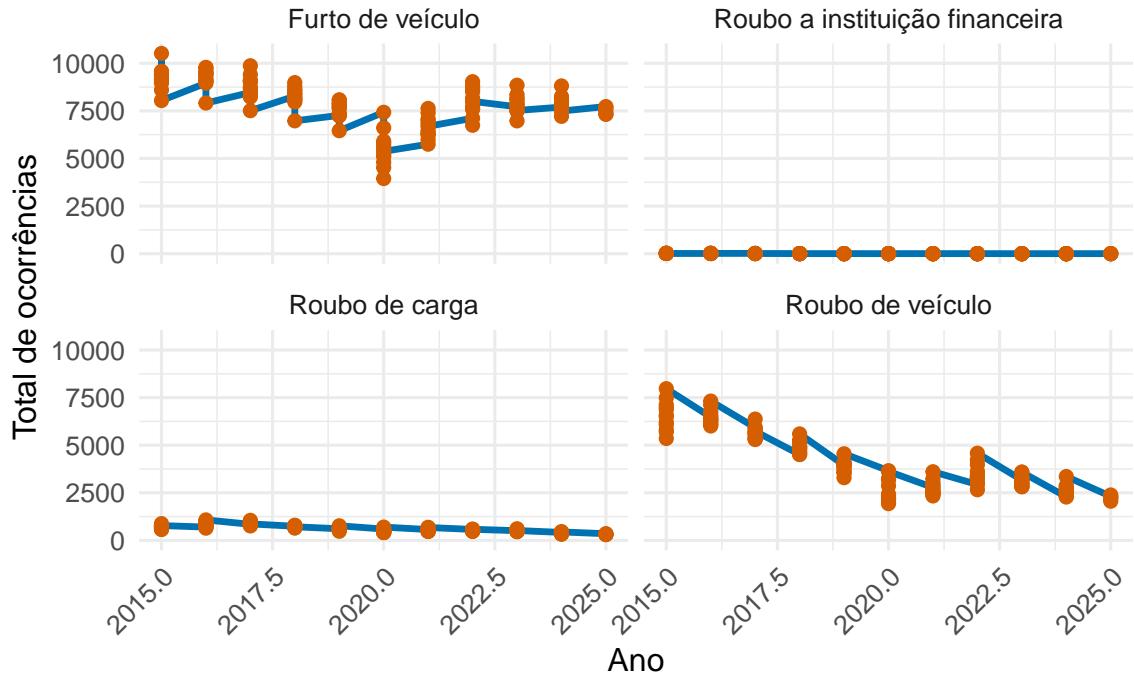
```

library(ggplot2)

ggplot(sp_anuais, aes(x = ano, y = total)) + geom_line(color = "#0072B2", size = 1.2) +
  ↵ geom_point(color = "#D55E00",
    size = 2) + facet_wrap(~evento) + labs(title = "Evolução anual de ocorrências no Estado de SP",
      ↵ y = "Total de ocorrências",
      x = "Ano") + theme_minimal(base_size = 12) + theme(plot.title = element_text(face = "bold",
        ↵ size = 14),
        axis.text.x = element_text(angle = 45, hjust = 1))
#> Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
#> i Please use `linewidth` instead.
#> This warning is displayed once every 8 hours.
#> Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

```

Evolução anual de ocorrências no Estado de SP



Exemplo 5 - Visualização de dados com vítimas

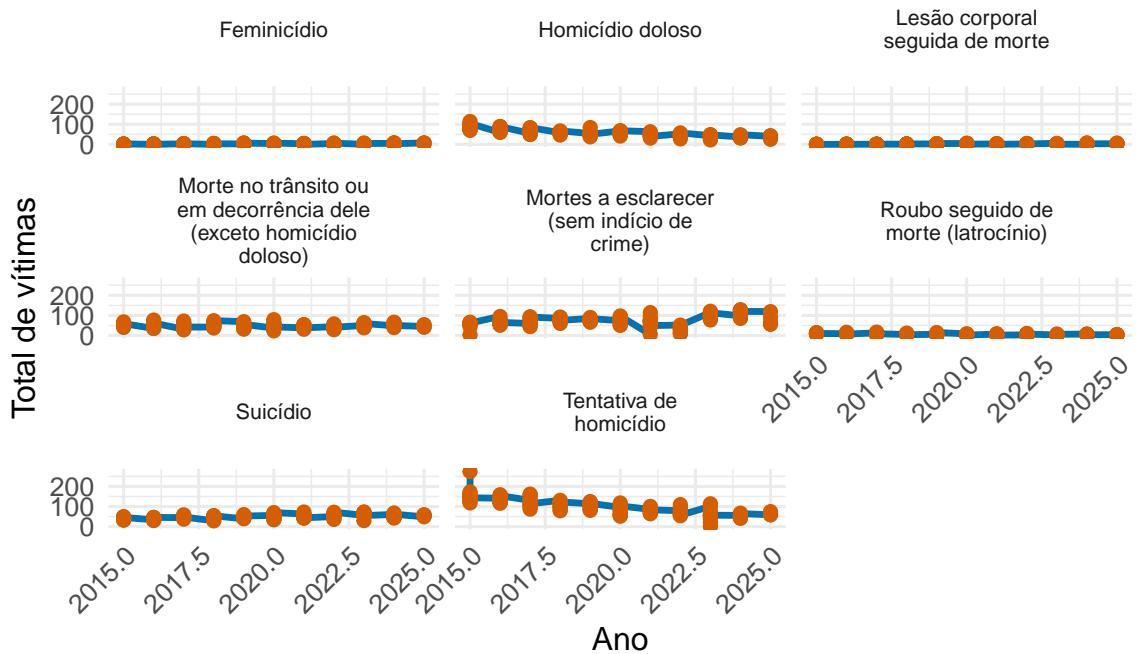
```
library(ggplot2)

sp_anuais_vit <- get_sinesp_vde_data(state = "SP", city = "SÃO PAULO", category = "vítimas",
  ↵ granularity = "year")
#> Query completed.

# Quebrar nomes longos nos títulos dos facetos
sp_anuais_vit$evento <- stringr::str_wrap(sp_anuais_vit$evento, width = 20)

ggplot(sp_anuais_vit, aes(x = ano, y = total_vitima)) + geom_line(color = "#0072B2", size = 1.2) +
  ↵ geom_point(color = "#D55E00",
  size = 2) + facet_wrap(~evento) + labs(title = "Evolução anual de vítimas na cidade de SP", y =
  ↵ "Total de vítimas",
  x = "Ano") + theme_minimal(base_size = 12) + theme(plot.title = element_text(face = "bold",
  ↵ size = 14),
  axis.text.x = element_text(angle = 45, hjust = 1), strip.text = element_text(size = 8))
```

Evolução anual de vítimas na cidade de SP



Exemplo 6 - Buscando tendências com o argumento geom_smooth em ggplot

```
# Carregar dados
dados_var <- get_sinesp_vde_data(state = "BA", typology = "Homicídio doloso", city = "Salvador",
  category = "vítimas",
  granularity = "month", year = 2015:2023)
#> Query completed.

dados_var <- dados_var |>
  dplyr::mutate(data = as.Date(paste0(ano, "-", mes, "-01")))

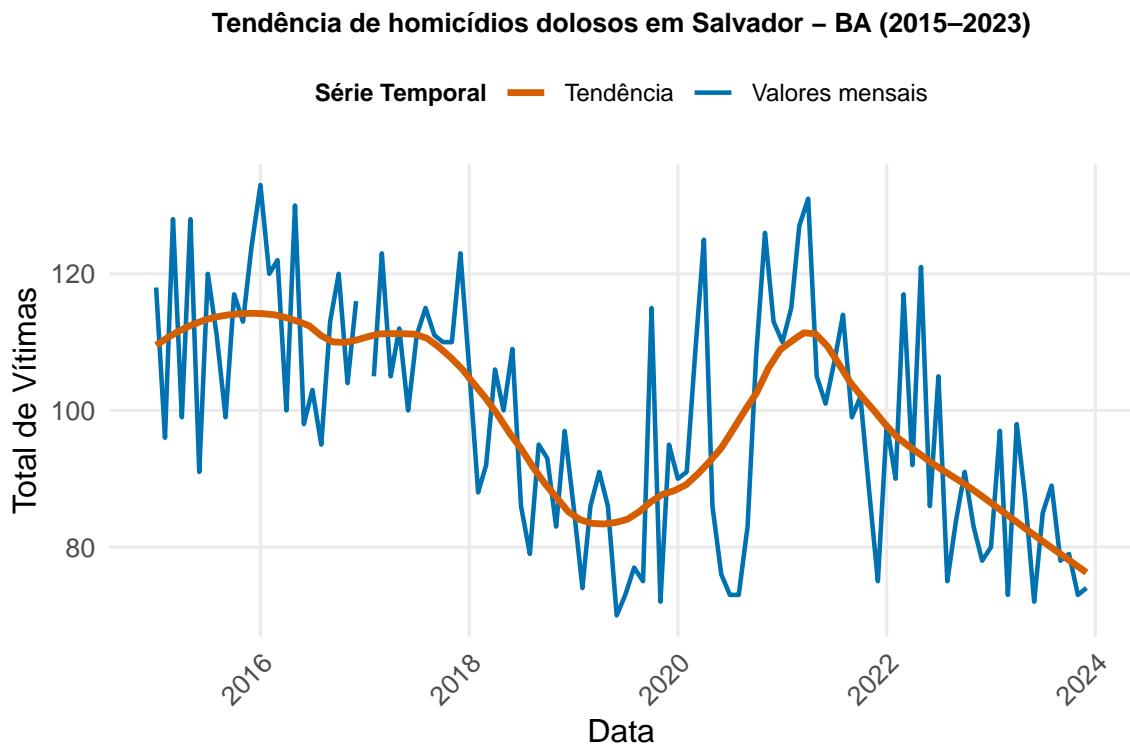
# Carregar pacotes
library(ggplot2)

# Criar gráfico com smoothing
ggplot(dados_var, aes(x = data)) + geom_line(aes(y = total_vitima, color = "Valores mensais"), size =
  0.8) +
  geom_smooth(aes(y = total_vitima, color = "Tendência"), method = "loess", se = FALSE, size =
  1.2, span = 0.3) +
  scale_color_manual(name = "Série Temporal", values = c(`Valores mensais` = "#0072B2", Tendência =
  "#D55E00")) +
  labs(title = "Tendência de homicídios dolosos em Salvador - BA (2015-2023)", x = "Data", y =
  "Total de Vítimas") +
  theme_minimal(base_size = 12) + theme(plot.title = element_text(face = "bold", size = 10, hjust
  = 0.5),
  axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.minor = element_blank(),
  legend.position = "top",
  legend.direction = "horizontal", legend.box = "horizontal", legend.title = element_text(size =
  9, face = "bold")),
```

```

    legend.text = element_text(size = 9))
#> `geom_smooth()` using formula = 'y ~ x'
#> Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).

```



Exemplo 7 - Análise de correlação

```

# Pacotes necessários
library(BrazilCrime)
library(tidyverse)
library(ggcorrplot)

# Buscar dados para SP em 2022 (cross-section municipal)
dados_cross <- get_sinesp_vde_data(state = "SP", granularity = "year", year = 2022, category =
  "vítimas")
#> Query completed.

# Selecionar algumas tipologias comuns
tipos_desejados <- c("Feminicídio", "Homicídio doloso", "Suicídio", "Lesão corporal seguida de
  morte")

# Agrupar e pivotar os dados corretamente
dados_filtrados <- dados_cross |>
  filter(evento %in% tipos_desejados) |>
  group_by(municipio, evento) |>
  summarise(total_vitima = sum(total_vitima, na.rm = TRUE), .groups = "drop") |>
  pivot_wider(names_from = evento, values_from = total_vitima) |>
  drop_na() |>
  mutate(across(-municipio, as.numeric))

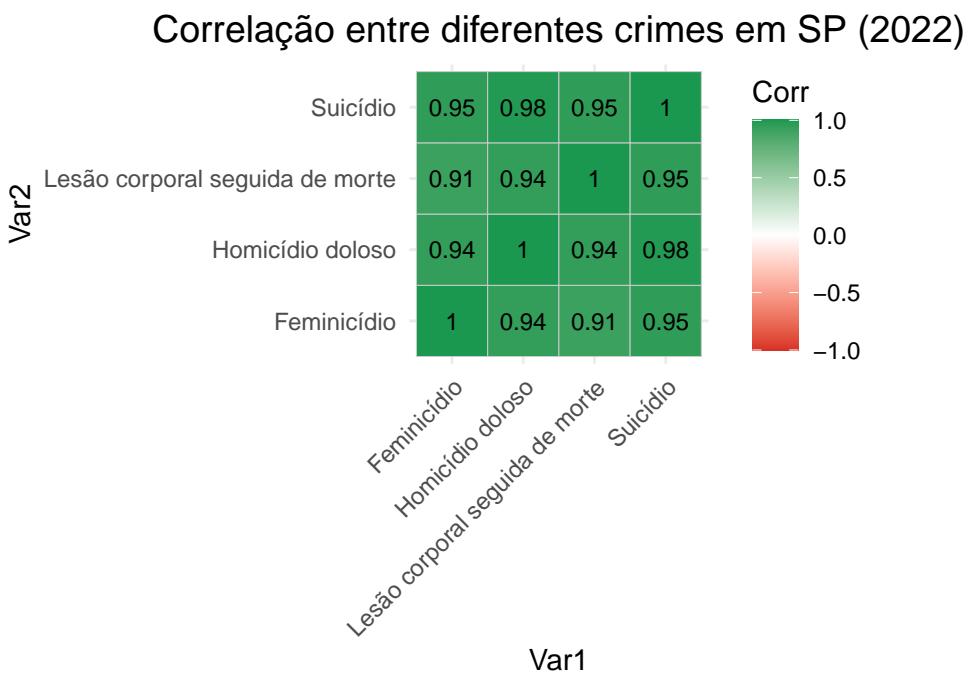
```

```

# Calcular matriz de correlação
matriz_cor <- cor(dados_filtrados[-1], method = "pearson")

# Gerar gráfico de correlação
ggcorrplot(matriz_cor, lab = TRUE, lab_size = 3, colors = c("#D73027", "white", "#1A9850"),
  ↪ outline.color = "gray80") +
  ggtitle("Correlação entre diferentes crimes em SP (2022)") + theme_minimal() + theme(plot.title
  ↪ = element_text(size = 14,
  hjust = 0.5), plot.margin = margin(t = 20, r = 10, b = 20, l = 10), axis.text.x =
  ↪ element_text(angle = 45,
  vjust = 1, hjust = 1))

```



References

- [1] Ministério da Justiça e Segurança Pública (MJSP). *SINESP - Sistema Nacional de Informações de Segurança Pública*. Disponível em: <https://www.gov.br/mj/pt-br/assuntos/sua-seguranca/seguranca-publica/estatistica/dados-nacionais-1/base-de-dados-e-notas-metodologicas-dos-gestores-estaduais-sinesp-vde-2022-e-2023>. Acesso em: julho de 2025.
- [2] Vargette, G., Laltuf, I., Justus, M. (2024). *BrazilCrime: Interface to Brazilian Crime Data*. CRAN - Comprehensive R Archive Network. Disponível em: <https://CRAN.R-project.org/package=BrazilCrime>
- [3] Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>
- [4] Gu, Z., Eils, R., Schlesner, M. (2022). *ggcorrplot: Visualization of a Correlation Matrix using ggplot2*. Disponível em: <https://cran.r-project.org/package=ggcorrplot>

- [5] R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- [6] Zhu, H. (2021). *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*. CRAN. <https://cran.r-project.org/package=kableExtra>
- [7] Wickham, H., François, R., Henry, L., Müller, K. (2019). *dplyr: A Grammar of Data Manipulation*. <https://dplyr.tidyverse.org>
- [8] Wickham, H. (2023). *tidyverse: Tidy Messy Data*. <https://tidyverse.org>
- [9] Chang, W. (2023). *stringr: Simple, Consistent Wrappers for Common String Operations*. <https://stringr.tidyverse.org>