

# Trabalho de Avaliação de Políticas Sociais

Ana Di Nur, Gabriel Figueiredo, Tiago Brancher

- Diferença em relação à versão de 14/11/2025: Vou juntar dt\_municipios\_eleicoes com dt\_emendas antes de cada exercício em vez de tentar juntar tudo no começo e ir alterando depois.
- Diferença em relação à versão de 15/11/2025: Mudei os dados selecionados de características municipais (para serem mais recentes) e a fonte de dados de emendas parlamentares (para que o ano da emenda refletisse seu ano de empenho, não proposição)

## 0) Preparativos

```
library(readxl) # para obtenção dos dados
library(basedosdados) # para obtenção dos dados
library(dplyr) # para transformação dos dados
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
library(data.table) # para transformação dos dados
```

Attaching package: 'data.table'

The following objects are masked from 'package:dplyr':

```
between, first, last
```

```
library(janitor) # para transformação dos dados
```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

```
chisq.test, fisher.test
```

```
library(ggplot2) # para visualização dos dados
library(cowplot) # para visualização dos dados (plotá-los lado a lado)
library(scales) # para visualização dos dados (fazer eixos bonito)
library(estimatr) # para modelagem
library(modelsummary) # para visualização de resultados dos modelos
```

`modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing backend. Learn more at: <https://vincentarelbundock.github.io/tinytable/>

Revert to `kableExtra` for one session:

```
options(modelsummary_factory_default = 'kableExtra')
options(modelsummary_factory_latex = 'kableExtra')
options(modelsummary_factory_html = 'kableExtra')
```

Silence this message forever:

```
config_modelsummary(startup_message = FALSE)
```

## 1) Coleta e junção de dados

### 1.1) Coleta de dados

#### 1.1.1) Características municipais

- Origem dos dados: Acessar [IBGE Cidades](#) > Selecionar UFs > Selecionar variáveis de interesse

- Hipótese: Perfil municipal não mudou significativamente entre os anos de coleta dos dados (2021, 2022) e o ano da eleição (2020) e do empenho das emendas (2021)
- Variáveis selecionadas:
  - \* População no último censo (2022)
  - \* Densidade demográfica (2022)
  - \* PIB *per capita* municipal (2021)
  - \* Taxa de escolarização de 6 a 14 anos de idade (2022)
  - \* Mortalidade infantil (2023)

```
# Carregar planilha montada a partir dos relatórios do IBGE Cidades
dt_municipios <- read_excel("Bases de dados/Brasil/IBGE_municipios.xlsx")
setDT(dt_municipios)

# Renomear colunas
nomes <- c("id_municipio_nome", "sigla_uf", "mortalidade_infantil", "PIBpc", "taxa_escolariza
colnames(dt_municipios) <- nomes
rm(nomes)

# Transformar gentílico na sigla do UF
dt_municipios[, sigla_uf := fcase(
  sigla_uf == "acriano", "AC",
  sigla_uf == "alagoano", "AL",
  sigla_uf == "amapaense", "AP",
  sigla_uf == "amazonense", "AM",
  sigla_uf == "baiano", "BA",
  sigla_uf == "cearense", "CE",
  sigla_uf == "brasiliense", "DF",
  sigla_uf == "capixaba ou espírito-santense", "ES",
  sigla_uf == "goiano", "GO",
  sigla_uf == "maranhense", "MA",
  sigla_uf == "mato-grossense", "MT",
  sigla_uf == "sul-mato-grossense ou mato-grossense-do-sul", "MS",
  sigla_uf == "mineiro", "MG",
  sigla_uf == "paranaense", "PR",
  sigla_uf == "paraibano", "PB",
  sigla_uf == "paraense", "PA",
  sigla_uf == "pernambucano", "PE",
  sigla_uf == "piauiense", "PI",
  sigla_uf == "potiguar, norte-rio-grandense, rio-grandense-do-norte", "RN",
```

```

sigla_uf == "gaúcho ou sul-rio-grandense", "RS",
sigla_uf == "fluminense", "RJ",
sigla_uf == "rondoniense ou rondoniano", "RO",
sigla_uf == "roraimense", "RR",
sigla_uf == "catarinense ou barriga-verde", "SC",
sigla_uf == "sergipano ou sergipense", "SE",
sigla_uf == "paulista", "SP",
sigla_uf == "tocantinense", "TO"
)]

# Criar coluna de região
dt_municipios[, regioao := fcase(
  sigla_uf %in% c("AC", "AM", "AP", "PA", "RO", "RR", "TO"), "Norte",
  sigla_uf %in% c("AL", "BA", "CE", "MA", "PE", "PB", "PI", "RN", "SE"), "Nordeste",
  sigla_uf %in% c("DF", "GO", "MS", "MT"), "Centro-Oeste",
  sigla_uf %in% c("ES", "MG", "RJ", "SP"), "Sudeste",
  sigla_uf %in% c("PR", "RS", "SC"), "Sul")]

# Transformar colunas em variáveis numéricas
## Inteiros
dt_municipios[, pop := gsub("[^0-9.-]", "", pop) %>% as.numeric()]
## Decimais
dt_municipios[,
  densidade_demografica := as.numeric(
    gsub(",", ".", gsub("[^0-9,]", "", densidade_demografica))
  )
]
dt_municipios[,
  mortalidade_infantil := as.numeric(
    gsub(",", ".", gsub("[^0-9,]", "", mortalidade_infantil))
  )
]
dt_municipios[,
  PIBpc := as.numeric(
    gsub(",", ".", gsub("[^0-9,]", "", PIBpc))
  )
]
dt_municipios[,
  taxa_escolarizacao := as.numeric(
    gsub(",", ".", gsub("[^0-9,]", "", taxa_escolarizacao))
  )/100
]

```

```
# Dar uma olhada na base
str(dt_municipios)
```

Classes 'data.table' and 'data.frame': 5571 obs. of 8 variables:

```
$ id_municipio_nome      : chr  "Acrelândia" "Assis Brasil" "Brasiléia" "Bujari" ...
$ sigla_uf               : chr  "AC" "AC" "AC" "AC" ...
$ mortalidade_infantil   : num   NA 7.78 5.78 19.53 11.11 ...
$ PIBpc                  : num   25363 17508 25279 28455 41723 ...
$ taxa_escolarizacao     : num   0.961 0.957 0.959 0.957 0.96 ...
$ pop                    : num   14021 8100 26000 12917 10392 ...
$ densidade_demografica  : num    7.74 1.63 6.62 4.26 6.09 ...
$ regio                  : chr   "Norte" "Norte" "Norte" "Norte" ...
- attr(*, ".internal.selfref")=<externalptr>
```

### 1.1.2) Eleições

- Origem dos dados: Acessar [Base dos Dados > Eleições Brasileiras](#) > Resultados por Candidato e Município

```
# # Fazer query
# query <- "SELECT
#     dados.ano AS ano,
#     dados.turno AS turno,
#     dados.sigla_uf AS sigla_uf,
#     diretorio_sigla_uf.nome AS sigla_uf_nome,
#     dados.id_municipio AS id_municipio,
#     diretorio_id_municipio.nome AS id_municipio_nome,
#     dados.cargo AS cargo,
#     dados.numero_partido AS numero_partido,
#     dados.sigla_partido AS sigla_partido,
#     dados.resultado AS resultado,
#     dados.votos AS votos
# FROM `basedosdados.br_tse_eleicoes.resultados_candidato_municipio` AS dados
# LEFT JOIN (
#     SELECT DISTINCT sigla, nome
#     FROM `basedosdados.br_bd_diretorios_brasil.uf`
# ) AS diretorio_sigla_uf
#     ON dados.sigla_uf = diretorio_sigla_uf.sigla
# LEFT JOIN (
#     SELECT DISTINCT id_municipio, nome
```

```

# FROM `basedosdados.br_bd_diretorios_brasil.municipio`
# ) AS diretorio_id_municipio
# ON dados.id_municipio = diretorio_id_municipio.id_municipio
# WHERE
#     dados.ano IN (2020)
#     AND dados.cargo = 'prefeito'
# "
# dt_eleicoes <- read_sql(query, billing_project_id = "pub-450900")
# setDT(dt_eleicoes)
# rm(query)
#
# # Salvar base de dados resultante
# saveRDS(dt_eleicoes, "Bases de dados/Brasil/TSE_eleicoes.rds")

# Carregar base de dados de eleições
dt_eleicoes <- readRDS("Bases de dados/Brasil/TSE_eleicoes.rds")
setDT(dt_eleicoes)

# Retirar primeiro turno dos municípios que tiveram segundo turno
municipios_com_segundo_turno <- dt_eleicoes[turno == 2, unique(id_municipio)]
dt_eleicoes <- dt_eleicoes[!(turno == 1 &
                           id_municipio %in% municipios_com_segundo_turno)]
rm(municipios_com_segundo_turno)

# Selecionar os dois candidatos mais votados
setorder(dt_eleicoes, -votos)
dt_eleicoes <- dt_eleicoes[, head(.SD, 2), by = id_municipio]

# Definir razão de votos dos top 2 (contando somente os votos desses 2)
dt_eleicoes[, votos_total := sum(votos), by = id_municipio]
dt_eleicoes[, votos_razao := votos/votos_total]

# Retirar municípios em que nenhum dos dois primeiros colocados são do PP
municipios_com_candidato_pp <- dt_eleicoes[sigla_partido == "PP", unique(id_municipio)]
dt_eleicoes <- dt_eleicoes[id_municipio %in% municipios_com_candidato_pp]
rm(municipios_com_candidato_pp)

# Selecionar municípios em que a margem de vitória foi de 5% ou menos
dt_eleicoes <- dt_eleicoes[votos_razao >= 0.45 & votos_razao <= 0.55]

# Criar dummy que indica se o candidato do PP ganhou

```

```

municipios_com_vitoria_pp <- dt_eleicoes[sigla_partido == "PP" & resultado == "eleito",
                                         unique(id_municipio)]
dt_eleicoes[, vitoria_pp := ifelse(id_municipio %in% municipios_com_vitoria_pp, 1, 0)]
rm(municipios_com_vitoria_pp)

# Selecionar só colunas e linhas relevantes
dt_eleicoes <- dt_eleicoes[
  sigla_partido == "PP",
  .(
    id_municipio,
    id_municipio_nome,
    sigla_uf,
    vitoria_pp,
    votos_razao
  )
]
setnames(dt_eleicoes, "votos_razao", "votos_razao_pp")

# Criar coluna com razão de votos centralizada
dt_eleicoes[, votos_razao_pp_centra := votos_razao_pp - 0.5]

# Criar coluna com razão de votos centralizada ao quadrado
dt_eleicoes[, votos_razao_pp_centra_sq := votos_razao_pp_centra^2]

# Dar uma olhada na base de dados resultante
str(dt_eleicoes)

```

```

Classes 'data.table' and 'data.frame':  487 obs. of  7 variables:
 $ id_municipio      : chr  "2507507" "1100205" "2408003" "3302502" ...
 $ id_municipio_nome : chr  "João Pessoa" "Porto Velho" "Mossoró" "Magé" ...
 $ sigla_uf          : chr  "PB" "RO" "RN" "RJ" ...
 $ vitoria_pp        : num  1 0 0 1 1 1 1 0 1 1 ...
 $ votos_razao_pp     : num  0.532 0.456 0.475 0.544 0.52 ...
 $ votos_razao_pp_centra : num  0.0316 -0.0445 -0.0252 0.0435 0.0197 ...
 $ votos_razao_pp_centra_sq: num  0.001001 0.00198 0.000634 0.001896 0.000387 ...
- attr(*, ".internal.selfref")=<externalptr>

```

Note que:

5567 municípios brasileiros tiveram eleições para prefeito em 2020;

1203 (22% dos 5567) tiveram um candidato do PP entre os dois candidatos mais votados;