

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)
- Diferença em relação à versão de 21/11/2025: Filtrei por eleições acirradas só mais tarde para poder fazer os gráficos de (des)continuidades de características e valor empregado por razão de votos do candidato a prefeito do PP; não analisei as emendas de bancada após constatar que elas são poucas
- Diferença em relação à versão de 22/11/2025: Escolhi as janelas ótimas segundo o método de Calonico *et al.* em vez de arbitrariamente e coloquei as análises quadráticas como teste de robustez e, como a janela ótima varia de acordo com controles e forma funcional, estimei um modelo linear sem controles, um modelo linear com controles e um modelo quadrático com controles separadamente (análogo ao que o Naércio fez no artigo dele)
- Diferença em relação à versão de 23/11/2025: Retirei as comparações de características municipais, pois entendi que não era para filtrar pela janela e ver se os municípios são parecidos (para isso, se confia no método rs). A ideia das comparações nos artigos é comparar os municípios tratados e controles ANTES de filtrar pela janela, para ver se eles são sistematicamente diferentes e, portanto, ter quase que uma indicação do grau de necessidade do RDD

0) Preparativos

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

```

library(data.table) # para transformação dos dados
library(tibble) # para transformação dos dados
library(janitor) # para transformação dos dados
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(showtext) # para usar outras fontes
library(estimatr) # para modelagem
library(rdrobust) # para escolher a janela ótima
library(modelsummary) # para visualização de resultados dos modelos

# Carregar fonte Computer Modern
font_add("CMU Serif", "Fontes/cmunrm.ttf")
showtext_auto()

```

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\ndemográfica (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_municpios <- 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_escolarizacao", "pop", "densidade_demografica")
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 espirito-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_municpios[, regiao := 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 ...
$ regiao              : 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 municípios que tiveram eleições suplementares (identificados pela
# → presença de mais de um candidato eleito na base) e Cachoeira dos Índios,
# → que parece ter passado por algo parecido
municipios_com_mais_de_um_eleito <- dt_eleicoes[resultado == "eleito", .N, by
# → = id_municipio][N>1, unique(id_municipio)]
dt_eleicoes <- dt_eleicoes[!(id_municipio %in%
# → municipios_com_mais_de_um_eleito) & id_municipio != 2503308]
rm(municipios_com_mais_de_um_eleito)

# 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)

# 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_centr := votos_razao_pp - 0.5]

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

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

```

```

Classes 'data.table' and 'data.frame': 1196 obs. of 7 variables:
$ id_municipio          : chr  "3170206" "3303500" "2507507" "3143302" ...
$ id_municipio_nome      : chr  "Uberlândia" "Nova Iguaçu" "João Pessoa" "Montes Claros" ...
$ sigla_uf               : chr  "MG" "RJ" "PB" "MG" ...
$ vitoria_pp             : num  1 1 1 0 1 0 0 1 1 1 ...
$ votos_razao_pp          : num  0.8523 0.8175 0.5316 0.0788 0.9127 ...
$ votos_razao_pp_centr   : num  0.3523 0.3175 0.0316 -0.4212 0.4127 ...
$ votos_razao_pp_centr_sq: num  0.124 0.101 0.001 0.177 0.17 ...
- attr(*, ".internal.selfref")=<externalptr>

```

Note que:

- 5567 municípios brasileiros tiveram eleições para prefeito em 2020;
- 1204 (22% dos 5567) tiveram um candidato do PP entre os dois candidatos mais votados

1.1.3) Emendas parlamentares

- A base principal virá do Siga Brasil. No entanto, essa base não contém o código IBGE do município favorecido pela emenda. Por isso, vamos começar carregando dados da Base dos Dados (originados nos dados da CGU) que associam o identificador da emenda parlamentar ao município de destinação.
- Origem dos dados: [Base dos Dados > Emendas Parlamentares](#)
- Variáveis selecionadas:
 - ano_emenda
 - * Filtro: “2021”
 - Apesar de essa variável representar o ano em que a emenda foi proposta, não o ano do empenho, o Siga Brasil mostra que todas as 1183 emendas destinadas a municípios com empenho em 2021 foram propostas em 2021.
 - id_emenda
 - numero_emenda
 - id_municipio_gasto

```
# # Fazer query
# query <- "
# SELECT
#     dados.id_emenda as id_emenda,
#     dados.ano_emenda as ano_emenda,
#     dados.id_autor_emenda as id_autor_emenda,
#     dados.numero_emenda as numero_emenda,
#     dados.id_municipio_gasto as id_municipio
# FROM `basedosdados.br_cgu_emendas_parlamentares.microdados` AS dados
# WHERE
#     dados.ano_emenda IN (2021)
# "
# de_para_emendas_municipios <- read_sql(query, billing_project_id =
#   "pub-450900")
# setDT(de_para_emendas_municipios)
# rm(query)
#
# # Criar emenda (número/ano) igual à base de emendas do Siga Brasil
# de_para_emendas_municipios[, emenda_numero_ano := paste0(id_autor_emenda,
#                                         numero_emenda,
#                                         "-",
#                                         ano_emenda)]
# "
```

```

# # Selecionar somente as colunas de de-para
# de_para_emendas_municipios <- de_para_emendas_municipios[,
#   .(emenda_numero_ano, id_autor_emenda, id_municipio)]
#
# # Salvar base de dados resultante
# saveRDS(de_para_emendas_municipios, "Bases de
#   dados/Brasil/BD_id_emenda_municipio.rds")

# Carregar base de dados que relaciona emendas e municípios (por código do
#   IBGE)
de_para_emendas_municipios <- readRDS("Bases de
#   dados/Brasil/BD_id_emenda_municipio.rds")
de_para_emendas_municipios[, id_autor_emenda := NULL]

```

- Origem dos dados: Senado Federal > Portal do Orçamento > Siga Brasil > Painel Emendas > Gráficos customizados
 - Variáveis selecionadas:
 - * Autor (Tipo)
 - * Emenda (Número-Ano)
 - * Empenho (Ano)
 - Filtro: “2021”
 - * Função (Desc)
 - * Funcional Localidade (Desc)
 - * Funcional Localidade (Tipo)
 - Filtro: “MUNICÍPIO”
 - * GND (Desc)
 - * Empenhado (IPCA)

```

# Carregar base de dados de emendas parlamentares
dt_emendas <- read_excel("Bases de dados/Brasil/SigaBrasil_emendas.xlsx")
setDT(dt_emendas)

# Renomear colunas
colnames(dt_emendas) <- make_clean_names(colnames(dt_emendas))

# Juntar com informação de códigos IBGE
dt_emendas <- merge(dt_emendas, de_para_emendas_municipios, all.x = T)
rm(de_para_emendas_municipios)

# Substituir id_municipio manualmente para as 4 emendas sem essa informação
#   na base

```

```
dt_emendas[emenda_numero_ano == "14510004-2021", id_municipio := "5002704"]
dt_emendas[emenda_numero_ano == "37390008-2021", id_municipio := "2504009"]
dt_emendas[emenda_numero_ano == "40680005-2021", id_municipio := "1301605"]
```

```
# Garantir que não há NAs
colSums(is.na(dt_emendas))
```

emenda_numero_ano	autor_tipo	ano_emissao_empenho
0	0	0
funcao	funcional_localidade	funcional_localidade_tipo
0	0	0
gnd_desc	empenhado_ipca	id_municipio
0	0	0

```
# Renomear coluna de tipo
setnames(dt_emendas, "autor_tipo", "tipo_emenda")
```

```
# Resumir dados por município, tipo de emenda, função e GND e armazenar dt
# para eventuais análises de heterogeneidade
dt_emendas_com_funcao_gnd <- dt_emendas[,  

  .(valor_empenhado_ipca = sum(empenhado_ipca)),  

  by = .(id_municipio, tipo_emenda, funcao, gnd_desc)  

]
```

```
# Resumir dados por município e tipo de emenda
dt_emendas <- dt_emendas[,  

  .(valor_empenhado_ipca = sum(empenhado_ipca)),  

  by = .(id_municipio, tipo_emenda)  

]
```

```
# Ver quais tipos de emendas há na base
dt_emendas[, unique(tipo_emenda)]
```

```
[1] "INDIVIDUAL"                  "BANCADA ESTADUAL (RP 7)"
[3] "RELATOR (RP 9)"
```

```
# Simplificar nome do tipo de emenda
dt_emendas[, tipo_emenda := fcase(
  tipo_emenda == "INDIVIDUAL", "individual",
  tipo_emenda == "BANCADA ESTADUAL (RP 7)", "bancada",
```

```
  tipo_emenda == "RELATOR (RP 9)", "relator"
)]
```

```
# Dar uma olhada na base
head(dt_emendas)
```

	id_municipio	tipo_emenda	valor_empenhado_ipca
	<char>	<char>	<num>
1:	2607109	individual	122383.8
2:	2612604	individual	131423.8
3:	2607000	individual	252882.8
4:	2603009	individual	316103.5
5:	2210201	individual	255310.4
6:	2206670	individual	379324.2

```
# Resumir valor empenhado de emendas por tipo
dt_emendas[, .(valor_empenhado_ipca = sum(valor_empenhado_ipca)), by =
  ↪ tipo_emenda]
```

	tipo_emenda	valor_empenhado_ipca
	<char>	<num>
1:	individual	943753489
2:	bancada	1482525667
3:	relator	15760159

```
# Resumir qtde de municípios que receberam cada tipo de emenda
writeLines(paste0(
  "Quantidade de municípios a receberem qualquer tipo de emenda: ",
  ↪ length(dt_emendas[, unique(id_municipio)]),
  "\nQuantidade de municípios a receberem emendas individuais: ",
  ↪ length(dt_emendas[tipo_emenda == "individual", unique(id_municipio)]),
  "\nQuantidade de municípios a receberem emendas de bancada: ",
  ↪ length(dt_emendas[tipo_emenda == "bancada", unique(id_municipio)]),
  "\nQuantidade de municípios a receberem emendas de relator: ",
  ↪ length(dt_emendas[tipo_emenda == "relator", unique(id_municipio)])
))
```

Quantidade de municípios a receberem qualquer tipo de emenda: 666
Quantidade de municípios a receberem emendas individuais: 633

Quantidade de municípios a receberem emendas de bancada: 66

Quantidade de municípios a receberem emendas de relator: 1

- Note que 666 municípios brasileiros foram favorecidos por algum empenho de emenda em 2021 e que essas emendas foram somente individuais, de bancada ou de relator.

1.2) Junção de dados

1.2.1) Juntar bases de municípios e eleições

```
# Ver se algum município de dt_eleicoes não está em dt_municípios
municipios_dt_eleicoes <- unique(dt_eleicoes[, paste0(id_municipio_nome,
  ↵ sigla_uf)])
municipios_dt_municipios <- unique(dt_municipios[, paste0(id_municipio_nome,
  ↵ sigla_uf)])
for (muni in municipios_dt_eleicoes) {
  if (!muni %in% municipios_dt_municipios) {
    print(muni)
  }
}
```

```
[1] "Restinga SecaRS"
[1] "São Vicente FerrerPE"
[1] "Lauro MullerSC"
[1] "Santa TeresinhaBA"
[1] "Grão ParáSC"
[1] "São Cristovão do SulSC"
[1] "São Thomé das LetrasMG"
[1] "Amparo de São FranciscoSE"
[1] "WestfaliaRS"
[1] "Vespasiano CorreaRS"
```

```
rm(municipios_dt_eleicoes, municipios_dt_municipios, muni)

# Renomear municípios de dt_municipios fora do padrão de dt_eleicoes
dt_municipios[id_municipio_nome == "Santa Terezinha" & sigla_uf == "BA",
  id_municipio_nome := "Santa Teresinha"]
dt_municipios[id_municipio_nome == "São Tomé das Letras" & sigla_uf == "MG",
  id_municipio_nome := "São Thomé das Letras"]
dt_municipios[id_municipio_nome == "São Vicente Férrer" & sigla_uf == "PE",
```

```

        id_municipio_nome := "São Vicente Ferrer"]
dt_municpios[id_municipio_nome == "Restinga Sêca" & sigla_uf == "RS",
             id_municipio_nome := "Restinga Seca"]
dt_municpios[id_municipio_nome == "Vespasiano Corrêa" & sigla_uf == "RS",
             id_municipio_nome := "Vespasiano Correa"]
dt_municpios[id_municipio_nome == "Westfália" & sigla_uf == "RS",
             id_municipio_nome := "Westfalia"]
dt_municpios[id_municipio_nome == "Grão-Pará" & sigla_uf == "SC",
             id_municipio_nome := "Grão Pará"]
dt_municpios[id_municipio_nome == "Lauro Müller" & sigla_uf == "SC",
             id_municipio_nome := "Lauro Muller"]
dt_municpios[id_municipio_nome == "São Cristóvão do Sul" & sigla_uf == "SC",
             id_municipio_nome := "São Cristovão do Sul"]
dt_municpios[id_municipio_nome == "Amparo do São Francisco" & sigla_uf ==
             "SE",
             id_municipio_nome := "Amparo de São Francisco"]

# Repetir procedimento para ver se deu certo
municipios_dt_eleicoes <- unique(dt_eleicoes[, paste0(id_municipio_nome,
             sigla_uf)])
municipios_dt_municpios <- unique(dt_municpios[, paste0(id_municipio_nome,
             sigla_uf)])
for (muni in municipios_dt_eleicoes) {
  if (!muni %in% municipios_dt_municpios) {
    print(muni)
  }
}
rm(municipios_dt_eleicoes, municipios_dt_municpios, muni)

# Adicionar dados sobre os municípios com eleições de interesse
dt_municpios_eleicoes <- merge(dt_eleicoes, dt_municpios, all.x = T)
rm(dt_eleicoes, dt_municpios)

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

```

```

Classes 'data.table' and 'data.frame': 1196 obs. of 13 variables:
 $ id_municipio_nome      : chr  "Abadia de Goiás" "Abadiânia" "Abel Figueiredo" "Acrelândia"
 $ sigla_uf                : chr  "GO" "GO" "PA" "AC" ...
 $ id_municipio            : chr  "5200050" "5200100" "1500131" "1200013" ...
 $ vitoria_pp              : num  1 1 0 0 1 0 1 1 0 1 ...

```

```

$ votos_razao_pp           : num  0.694 0.601 0.491 0.477 0.502 ...
$ votos_razao_pp_centr     : num  0.19419 0.10064 -0.00885 -0.0231 0.00169 ...
$ votos_razao_pp_centr_sq: num  3.77e-02 1.01e-02 7.83e-05 5.34e-04 2.85e-06 ...
$ mortalidade_infantil   : num  13.94 16.13 NA NA 7.52 ...
$ PIBpc                   : num  38622 21335 13999 25363 46316 ...
$ taxa_escolarizacao     : num  0.945 0.99 0.997 0.961 0.982 ...
$ pop                      : num  19128 17232 6136 14021 21568 ...
$ densidade_demografica  : num  133.43 16.5 9.99 7.74 13.77 ...
$ regiao                  : chr  "Centro-Oeste" "Centro-Oeste" "Norte" "Norte" ...
- attr(*, ".internal.selfref")=<externalptr>
- attr(*, "sorted")= chr [1:2] "id_municipio_nome" "sigla_uf"

```

```

# Verificar se houve algum NA
colSums(is.na(dt_municipios_eleicoes))

```

id_municipio_nome	sigla_uf	id_municipio
0	0	0
vitoria_pp	votos_razao_pp	votos_razao_pp_centr
0	0	0
votos_razao_pp_centr_sq	mortalidade_infantil	PIBpc
0	362	0
taxa_escolarizacao	pop	densidade_demografica
0	0	0
regiao		
0		

- Como mortalidade infantil tem vários valores NA, não vamos analisá-la nem usá-la como controle.

1.2.2) Juntar base de municípios e eleições com bases de emendas

1.2.2.1) Para emendas de todos os tipos

```

# Resumir emendas por id_municipio, independentemente de tipo
dt_emendas_todas <- dt_emendas[,
                           .(valor_empenhado_ipca =
                             sum(valor_empenhado_ipca)),
                           by = id_municipio]

# Adicionar dados sobre emendas ao dt_municipios_eleicoes
dt_todas <- merge(dt_municipios_eleicoes, dt_emendas_todas, all.x = T, by =
                   "id_municipio")

```

```

rm(dt_emendas_todas)

# Transformar NAs em 0
dt_todas[is.na(valor_empenhado_ipca), valor_empenhado_ipca := 0]

# Adicionar coluna de valor empenhado per capita (usando valor de 2021 e pop
# de 2022)
dt_todas[, valor_empenhado_ipca_pc := valor_empenhado_ipca/pop]

```

1.2.2.2) Para emendas individuais

```

# Filtrar por somente emendas individuais
dt_emendas_individuais <- dt_emendas[tipo_emenda == "individual"]
# Remover coluna de tipo
dt_emendas_individuais[, tipo_emenda := NULL]

# Adicionar dados sobre emendas ao dt_municipios_eleicoes
dt_individuais <- merge(dt_municipios_eleicoes, dt_emendas_individuais, all.x
# de T, by = "id_municipio")
rm(dt_emendas_individuais)

# Transformar NAs em 0
dt_individuais[is.na(valor_empenhado_ipca), valor_empenhado_ipca := 0]

# Adicionar coluna de valor empenhado per capita (usando valor de 2021 e pop
# de 2022)
dt_individuais[, valor_empenhado_ipca_pc := valor_empenhado_ipca/pop]

```

1.2.2.3) Para emendas de bancada

```

# Filtrar por somente emendas de bancada
dt_emendas_bancada <- dt_emendas[tipo_emenda == "bancada"]
# Remover coluna de tipo
dt_emendas_bancada[, tipo_emenda := NULL]

# Adicionar dados sobre emendas ao dt_municipios_eleicoes
dt_bancada <- merge(dt_municipios_eleicoes, dt_emendas_bancada, all.x = T, by
# = "id_municipio")
rm(dt_emendas_bancada)

```

```

# Transformar NAs em 0
dt_bancada[is.na(valor_empenhado_ipca), valor_empenhado_ipca := 0]

# Adicionar coluna de valor empenhado per capita (usando valor de 2021 e pop
# de 2022)
dt_bancada[, valor_empenhado_ipca_pc := valor_empenhado_ipca/pop]

```

1.2.2.4) Para emendas de relator

```

# Filtrar por somente emendas de relator
dt_emendas_relator <- dt_emendas[tipo_emenda == "relator"]
# Remover coluna de tipo
dt_emendas_relator[, tipo_emenda := NULL]

# Adicionar dados sobre emendas ao dt_municipios_eleicoes
dt_relator <- merge(dt_municipios_eleicoes, dt_emendas_relator, all.x = T, by
# = "id_municipio")
rm(dt_emendas_relator)

# Transformar NAs em 0
dt_relator[is.na(valor_empenhado_ipca), valor_empenhado_ipca := 0]

# Adicionar coluna de valor empenhado per capita (usando valor de 2021 e pop
# de 2022)
dt_relator[, valor_empenhado_ipca_pc := valor_empenhado_ipca/pop]

# rm(dt_emendas)

```

2) Estatísticas descritivas

2.1) Comparação das características municipais dos tratados e controles

```

# Criar dt com médias
medias <- dt_individuais[
  ,
  .(

```

```

    media_pop = mean(pop),
    media_densidade_demografica = mean(densidade_demografica),
    media_PIBpc = mean(PIBpc),
    media_taxa_escolarizacao = mean(taxa_escolarizacao),
    media_valor_empenhado = mean(valor_empenhado_ipca_pc)
),
by = vitoria_pp
]

# Plotar medianas de quem venceu vs. não venceu - formatar
titulo <- ggplot() +
  labs(title = "Comparação das medianas de características municipais",
       subtitle = "Fontes dos dados: TSE (2021), IBGE (2021, 2022). Elaboração
       própria.") +
  theme_minimal()

g1 <- medias %>%
  ggplot(aes(y = media_pop, x = as.factor(vitoria_pp), data = )) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "População",
       x = "Prefeito do PP?")

g2 <- medias %>%
  ggplot(aes(y = media_densidade_demografica, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "Densidade\ndemográfica",
       x = "Prefeito do PP?")

g3 <- medias %>%
  ggplot(aes(y = media_PIBpc, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "PIB per capita",
       x = "Prefeito do PP?")

g4 <- medias %>%
  ggplot(aes(y = media_taxa_escolarizacao, x = as.factor(vitoria_pp))) +

```

```

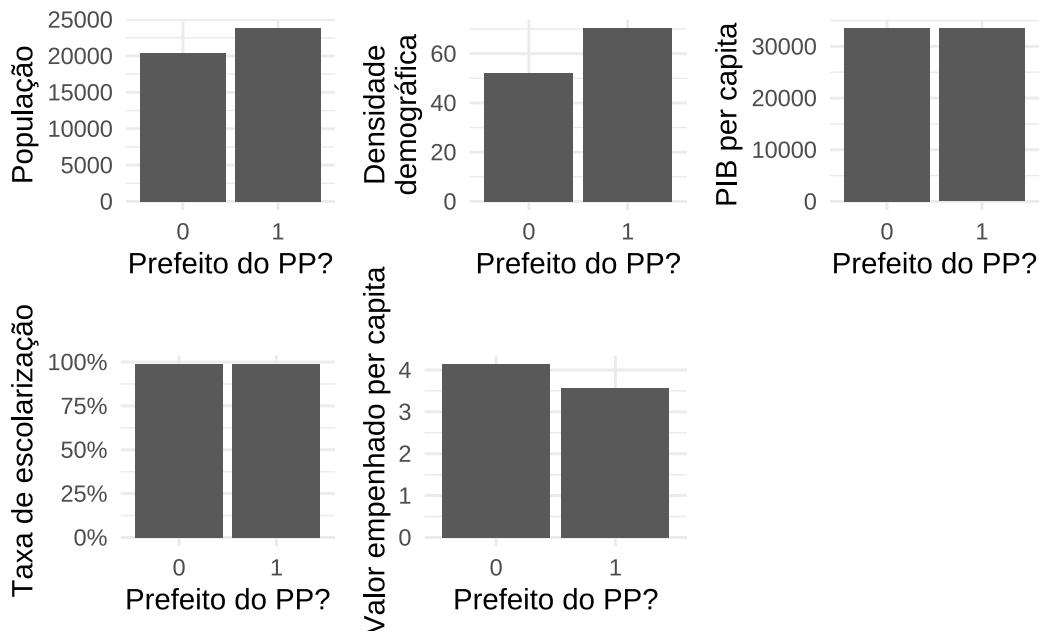
geom_col() +
scale_y_continuous(labels = label_percent()) +
theme_minimal() +
labs(title = "",
y = "Taxa de escolarização",
x = "Prefeito do PP?")

g5 <- medias %>%
ggplot(aes(y = media_valor_empenhado, x = as.factor(vitoria_pp))) +
geom_col() +
theme_minimal() +
labs(title = "",
y = "Valor empenhado per capita",
x = "Prefeito do PP?")

graficos <- plot_grid(g1, g2, g3, g4, g5)

# plot_grid(titulo, graficos, ncol = 1, rel_heights = c(0.15, 1))
graficos

```



```
rm(medias, titulo, graficos, g1, g2, g3, g4, g5)
```

2.2) Verificação de (des)continuidades

Note que filtramos os municípios para os percentis 2 a 99 de cada característica para evitar distorções causadas por outliers.

2.2.1) Características municipais

```
# Linear  
g_pop <- dt_todas[pop %between% quantile(pop, c(.02, .99), na.rm = TRUE)] %>%  
  ggplot(aes(x = votos_razao_pp_centr, y = pop, color =  
             ↵ as.factor(vitoria_pp))) +  
  geom_point(shape = 21, size = 0.5, color = "black") +  
  stat_smooth(method = "lm", se = F) +  
  scale_color_brewer(palette = "Dark2", direction = -1) +  
  geom_vline(xintercept = 0, linetype = "dashed") +  
  scale_y_continuous(labels = label_number()) +  
  theme_minimal() + guides(color = "none") +  
  labs(title = "",  
       y = "População",  
       x = "Margem de vitória do PP") + scale_x_continuous(labels =  
             ↵ label_percent()) + scale_x_continuous(labels = label_percent())  
  
g_densidade_demografica <- dt_todas[densidade_demografica %between%  
  ↵ quantile(densidade_demografica, c(.02, .99), na.rm = TRUE)] %>%  
  ggplot(aes(x = votos_razao_pp_centr, y = densidade_demografica, color =  
             ↵ as.factor(vitoria_pp))) +  
  geom_point(shape = 21, size = 0.5, color = "black") +  
  stat_smooth(method = "lm", se = F) +  
  scale_color_brewer(palette = "Dark2", direction = -1) +  
  geom_vline(xintercept = 0, linetype = "dashed") +  
  scale_y_continuous(labels = label_number()) +  
  theme_minimal() + guides(color = "none") +  
  labs(title = "",  
       y = "Densidade\ndemográfica",  
       x = "Margem de vitória do PP") + scale_x_continuous(labels =  
             ↵ label_percent())
```

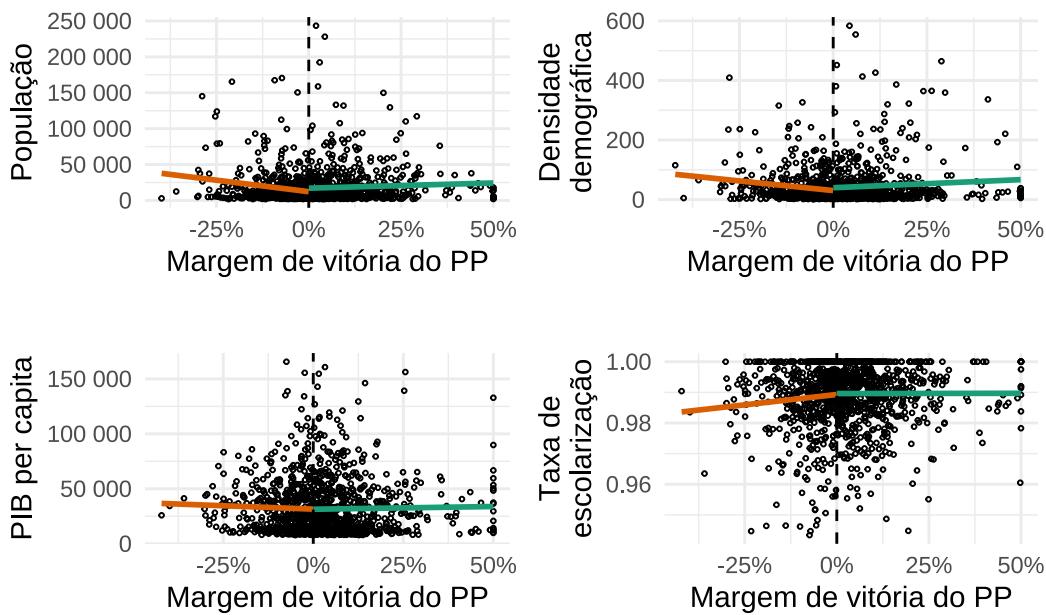
```

g_PIBpc <- dt_todas[PIBpc %between% quantile(PIBpc, c(.02, .99), na.rm =
  TRUE)] %>%
  ggplot(aes(x = votos_razao_pp_centr, y = PIBpc, color =
    as.factor(vitoria_pp))) +
  geom_point(shape = 21, size = 0.5, color = "black") +
  stat_smooth(method = "lm", se = F) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() + guides(color = "none") +
  labs(title = "",
       y = "PIB per capita",
       x = "Margem de vitória do PP") + scale_x_continuous(labels =
        label_percent())

g_taxa_escolarizacao <- dt_todas[taxa_escolarizacao %between%
  quantile(taxa_escolarizacao, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp_centr, y = taxa_escolarizacao, color =
    as.factor(vitoria_pp))) +
  geom_point(shape = 21, size = 0.5, color = "black") +
  stat_smooth(method = "lm", se = F) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() + guides(color = "none") +
  labs(title = "",
       y = "Taxa de\enescolarização",
       x = "Margem de vitória do PP") + scale_x_continuous(labels =
        label_percent())

plot_grid(g_pop, g_densidade_demografica, g_PIBpc, g_taxa_escolarizacao, ncol
  = 2)

```



3) Estimação

3.1) Obter janela ótima e filtrar base de dados

Agora, vamos filtrar as bases de dados por eleições acirradas.

```
# Guardar bases de dados completas
dt_todas_completa <- dt_todas
dt_individuais_completa <- dt_individuais
dt_bancada_completa <- dt_bancada
dt_relator_completa <- dt_relator

# Obter janela ótima para modelo linear com covariadas
covariadas <- dt_individuais[, .(pop, densidade_demografica, PIBpc,
  taxa_escolarizacao)]
janela <- rdbwselect(
  x = dt_individuais$votos_razao_pp_centr,
  y = dt_individuais$valor_empenhado_ipca_pc,
  p = 1,
  covs = covariadas
```

```

)
janela$bws

      h (left)  h (right)  b (left)  b (right)
mserd 0.08415324 0.08415324 0.1634445 0.1634445

# Filtrar base de dados para municípios cuja margem de vitória do PP está na
# janela
dt_todas <- dt_todas[votos_razao_pp_centr >= -janela$bws[1] &
                      votos_razao_pp_centr <= janela$bws[1]]
dt_individuais <- dt_individuais[votos_razao_pp_centr >= -janela$bws[1] &
                                      votos_razao_pp_centr <= janela$bws[1]]
dt_bancada <- dt_bancada[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]
dt_relator <- dt_relator[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]

# Ver quantos municípios com eleições acirradas receberam emendas
writeLines(paste0(
  "Quantidade de municípios com eleições acirradas: ",
  nrow(dt_todas),

  "\nQuantidade de municípios com eleições acirradas que receberam emendas:
  ",
  nrow(dt_todas[valor_empenhado_ipca != 0]),

  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  individuais: ",
  nrow(dt_individuais[valor_empenhado_ipca != 0]),

  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  bancada: ",
  nrow(dt_bancada[valor_empenhado_ipca != 0]),

  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  relator: ",
  nrow(dt_relator[valor_empenhado_ipca != 0])))

```

Quantidade de municípios com eleições acirradas: 708
 Quantidade de municípios com eleições acirradas que receberam emendas: 62

Quantidade de municípios com eleições acirradas que receberam emendas individuais: 59
Quantidade de municípios com eleições acirradas que receberam emendas de bancada: 7
Quantidade de municípios com eleições acirradas que receberam emendas de relator: 0

- Note que nenhum município com eleições acirradas recebeu emendas de relator e somente 7 receberam emendas de bancada, então vamos analisar somente emendas individuais.

3.2) Fazer estimação

Vamos estimar o valor adicional recebido via emendas parlamentares individuais pelos municípios em que o candidato a prefeito do PP ganhou por pouco, usando alguns modelos diferentes.

- Note que, quando o modelo inclui a running variable, é sua versão centralizada (votos_razao_pp_centr), porque isso evita que o intercepto tenha valores sem sentido (como valores empenhados negativos).

```
# Duas regressões lineares com controles de município
regs_lineares_controles <- lm_robust(valor_empenhado_ipca_pc ~
  ↵ vitoria_pp*votos_razao_pp_centr + regiao + pop + densidade_demografica +
  ↵ PIBpc + taxa_escolarizacao, data = dt_individuais)
```

3.2) Plotar estimativas

```
regs_lineares_controles_grafico <- lm(valor_empenhado_ipca_pc ~
  ↵ vitoria_pp*votos_razao_pp_centr + regiao + pop + densidade_demografica +
  ↵ PIBpc + taxa_escolarizacao, data = dt_individuais)

# 1) Filtrar como você vinha fazendo
dt_plot <- copy(dt_individuais)[
  valor_empenhado_ipca_pc %between% quantile(valor_empenhado_ipca_pc, c(.02,
  ↵ .99), na.rm = TRUE)
]

# 2) Escolher valores de referência para os controles:
# numéricos -> média; categórico (regiao) -> moda (categoria mais frequente)
num_vars <- c("pop", "densidade_demografica", "PIBpc", "taxa_escolarizacao")
cat_var <- "regiao"

refs <- list()
```

```

for (v in num_vars) refs[[v]] <- mean(dt_plot[[v]], na.rm = TRUE)
# modo para regiao
refs[[cat_var]] <- dt_plot[, .N, by = regiao][order(-N)][1, regiao]

# 3) Grid QUEBRADO por lado do cutoff

# lado perdedor (0): só valores < 0
x_left <- seq(min(dt_plot[vitoria_pp == 0]$votos_razao_pp_centr, na.rm =
  TRUE),
  0,
  length.out = 200)

grid_left <- data.table(
  vitoria_pp = 0,
  votos_razao_pp_centr = x_left
)

# lado vencedor (1): só valores > 0
x_right <- seq(0,
  max(dt_plot[vitoria_pp == 1]$votos_razao_pp_centr, na.rm =
  TRUE),
  length.out = 200)

grid_right <- data.table(
  vitoria_pp = 1,
  votos_razao_pp_centr = x_right
)

# adicionar controles aos dois grids
for(v in num_vars){
  grid_left[[v]] <- refs[[v]]
  grid_right[[v]] <- refs[[v]]
}

grid_left[[cat_var]] <- refs[[cat_var]]
grid_right[[cat_var]] <- refs[[cat_var]]

# juntar resultados
newdata_grid <- rbind(grid_left, grid_right)

# certifique-se que fatores no newdata têm os mesmos níveis que no
# modelo/data
# (se regiao for factor)

```

```

if (is.factor(dt_individuais$regiao)) {
  newdata_grid[, regiao := factor(regiao, levels =
  ↪ levels(dt_individuais$regiao))]
}

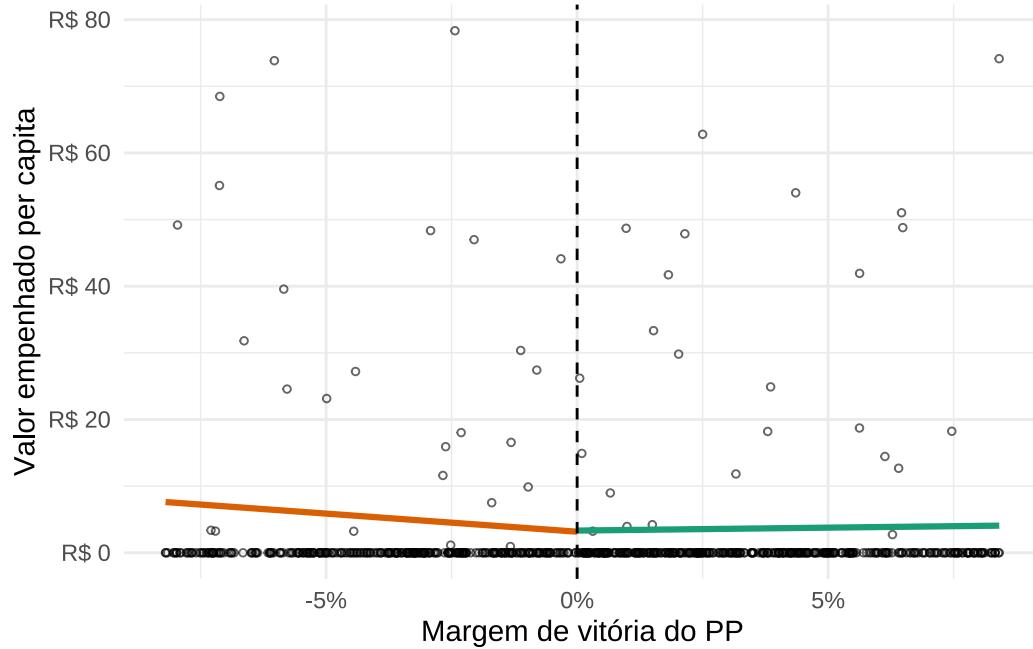
if (is.factor(dt_individuais$vitoria_pp)) {
  newdata_grid[, vitoria_pp := factor(vitoria_pp, levels =
  ↪ levels(dt_individuais$vitoria_pp))]
}

# 4) Prever usando o lm que você rodou (gera fit, lwr, upr)
preds <- predict(regs_lineares_controles_grafico,
  newdata = newdata_grid,
  interval = "confidence", level = 0.95)

newdata_grid[, `:=`(
  fit = preds[, "fit"],
  lwr = preds[, "lwr"],
  upr = preds[, "upr"]
)]

# 5) Plot: pontos reais + linhas preditas (uma por vitoria_pp) + ribbon
ggplot() +
  # pontos reais (opcional, com alpha pra não poluir)
  geom_point(data = dt_plot, aes(x = votos_razao_pp_centr, y =
  ↪ valor_empenhado_ipca_pc,
    color = as.factor(vitoria_pp)),
    shape = 21, color = "black", alpha = 0.6, size = 1) +
  # linha predita
  geom_line(data = newdata_grid,
    aes(x = votos_razao_pp_centr, y = fit, color =
    ↪ as.factor(vitoria_pp)),
    linewidth = 1.1) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  scale_fill_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_currency(prefix = "R$ ")) +
  scale_x_continuous(labels = label_percent()) +
  theme_minimal() + guides(color = "none", fill = "none") +
  labs(y = "Valor empenhado per capita", x = "Margem de vitória do PP")

```



4) Teste de heterogeneidade

Seria bom repetir a estimação por região, mas seria preciso prestar atenção no número de observações e de tratados.

4.1) Norte

```
# Filtrar base de dados pra região Norte
# Repetir estimação
```

4.2) Nordeste

```
# Filtrar base de dados pra região Nordeste
# Repetir estimação
```

4.3) Sudeste

```
# Filtrar base de dados pra região Sudeste  
# Repetir estimação
```

4.4) Sul

```
# Filtrar base de dados pra região Sul  
# Repetir estimação
```

4.5) Centro-Oeste

```
# Filtrar base de dados pra região Centro-Oeste  
# Repetir estimação
```

5) Teste placebo

Não vamos fazer, já que não encontramos efeito. (Se encontramos efeito para certa região, fazer para ela.)

6) Teste de robustez: Modelo linear sem controles

```
# Recuperação das versões completas das bases de dados  
dt_todas <- dt_todas_completa  
dt_individuais <- dt_individuais_completa  
dt_bancada <- dt_bancada_completa  
dt_relator <- dt_relator_completa
```

6.3) Estimação

6.3.1) Obter janela ótima e filtrar bases de dados

```
# Obter janela ótima para modelo linear sem covariadas
janela <- rdbwselect(
  x = dt_individuais$votos_razao_pp_centr,
  y = dt_individuais$valor_empenhado_ipca_pc,
  p = 1
)
janela$bws

      h (left) h (right)  b (left) b (right)
mserd 0.107061  0.107061 0.2173563 0.2173563

# Filtrar base de dados para municípios cuja margem de vitória do PP está na
# janela
dt_todas <- dt_todas[votos_razao_pp_centr >= -janela$bws[1] &
                      votos_razao_pp_centr <= janela$bws[1]]
dt_individuais <- dt_individuais[votos_razao_pp_centr >= -janela$bws[1] &
                                      votos_razao_pp_centr <= janela$bws[1]]
dt_bancada <- dt_bancada[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]
dt_relator <- dt_relator[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]

# Ver quantos municípios com eleições acirradas receberam emendas
writeLines(paste0(
  "Quantidade de municípios com eleições acirradas: ",
  nrow(dt_todas),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas:
  ",
  nrow(dt_todas[valor_empenhado_ipca != 0]),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas
  individuais: ",
  nrow(dt_individuais[valor_empenhado_ipca != 0]),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  bancada: ",
```

```

nrow(dt_bancada[valor_empenhado_ipca != 0]),

"\nQuantidade de municípios com eleições acirradas que receberam emendas de
← relator: ",
nrow(dt_relator[valor_empenhado_ipca != 0]))

```

Quantidade de municípios com eleições acirradas: 815
 Quantidade de municípios com eleições acirradas que receberam emendas: 70
 Quantidade de municípios com eleições acirradas que receberam emendas individuais: 65
 Quantidade de municípios com eleições acirradas que receberam emendas de bancada: 9
 Quantidade de municípios com eleições acirradas que receberam emendas de relator: 0

- Note que nenhum município com eleições acirradas recebeu emendas de relator e somente 9 receberam emendas de bancada, então vamos analisar somente emendas individuais.

6.3.1) Estimar

Vamos estimar o valor adicional recebido via emendas parlamentares individuais pelos municípios em que o candidato a prefeito do PP ganhou por pouco, usando alguns modelos diferentes.

- Note que, quando o modelo inclui a running variable, é sua versão centralizada (votos_razao_pp_centr), porque isso evita que o intercepto tenha valores sem sentido (como valores empenhados negativos).

```

# Duas regressões lineares sem controles de município
regs_lineares <- lm_robust(valor_empenhado_ipca_pc ~
← vitoria_pp*votos_razao_pp_centr, data = dt_individuais)

```

6.3.2) Plotar estimativas

```

# 1) Filtrar dados para o gráfico
dt_plot <- copy(dt_individuais)[
  valor_empenhado_ipca_pc %between% quantile(valor_empenhado_ipca_pc, c(.02,
← .99), na.rm = TRUE)
]

# 2) Rodar lm apenas para gerar previsões

```

```

regs_grafico <- lm(valor_empenhado_ipca_pc ~ vitoria_pp*votos_razao_pp_centr,
                     data = dt_plot)

# 3) Criar grids separados por lado do cutoff

# esquerda (vitoria_pp = 0)
x_left <- seq(min(dt_plot[vitoria_pp == 0]$votos_razao_pp_centr, na.rm =
    TRUE),
               0,
               length.out = 200)
grid_left <- data.table(
  vitoria_pp = 0,
  votos_razao_pp_centr = x_left
)

# direita (vitoria_pp = 1)
x_right <- seq(0,
                max(dt_plot[vitoria_pp == 1]$votos_razao_pp_centr, na.rm =
                    TRUE),
                length.out = 200)
grid_right <- data.table(
  vitoria_pp = 1,
  votos_razao_pp_centr = x_right
)

newdata_grid <- rbind(grid_left, grid_right)

# 4) Previsão das linhas
newdata_grid[, fit := predict(regs_grafico, newdata = newdata_grid)]

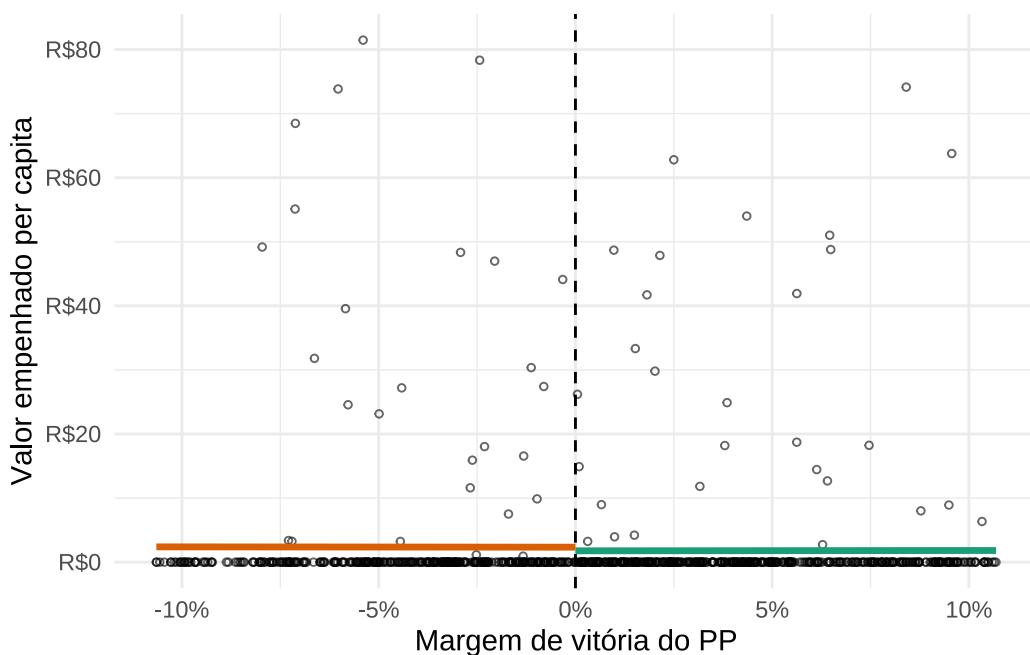
# 5) Plot
ggplot() +
  geom_point(data = dt_plot,
             aes(x = votos_razao_pp_centr, y = valor_empenhado_ipca_pc,
                  color = as.factor(vitoria_pp)),
             shape = 21, color = "black", alpha = 0.6, size = 1) +
  geom_line(data = newdata_grid,
            aes(x = votos_razao_pp_centr, y = fit, color =
                as.factor(vitoria_pp)),
            linewidth = 1.2) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_color_brewer(palette = "Dark2", direction = -1) +

```

```

scale_y_continuous(labels = scales::label_currency(prefix = "R$")) +
scale_x_continuous(labels = scales::label_percent()) +
theme_minimal() + guides(color = "none") +
labs(x = "Margem de vitória do PP", y = "Valor empenhado per capita")

```



7) Teste de robustez: Modelo quadrático

Vamos repetir as estatísticas e estimativa para modelos quadráticos.

```

# Recuperação das versões completas das bases de dados
dt_todas <- dt_todas_completa
dt_individuais <- dt_individuais_completa
dt_bancada <- dt_bancada_completa
dt_relator <- dt_relator_completa

```

7.1) Verificação de (des)continuidades

Note que filtramos os municípios para os percentis 2 a 99 de cada característica para evitar distorções causadas por outliers.

7.1.1) Características municipais

```
# Recuperação das versões completas das bases de dados
dt_todas <- dt_todas_completa
dt_individuais <- dt_individuais_completa
dt_bancada <- dt_bancada_completa
dt_relator <- dt_relator_completa

# Linear
g_pop <- dt_todas[pop %between% quantile(pop, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp_centr, y = pop, color =
  ↪ as.factor(vitoria_pp))) +
  geom_point(shape = 21, size = 0.5, color = "black") +
  stat_smooth(method = "lm", se = F, formula = y ~ poly(x, 2)) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() + guides(color = "none") +
  labs(title = "",
       y = "População",
       x = "Margem de vitória do PP") + scale_x_continuous(labels =
  ↪ label_percent()) + scale_x_continuous(labels = label_percent())

g_densidade_demografica <- dt_todas[densidade_demografica %between%
  ↪ quantile(densidade_demografica, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp_centr, y = densidade_demografica, color =
  ↪ as.factor(vitoria_pp))) +
  geom_point(shape = 21, size = 0.5, color = "black") +
  stat_smooth(method = "lm", se = F, formula = y ~ poly(x, 2)) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() + guides(color = "none") +
  labs(title = "",
       y = "Densidade\ndemográfica",
       x = "Margem de vitória do PP") + scale_x_continuous(labels =
  ↪ label_percent())

g_PIBpc <- dt_todas[PIBpc %between% quantile(PIBpc, c(.02, .99), na.rm =
  ↪ TRUE)] %>%
```

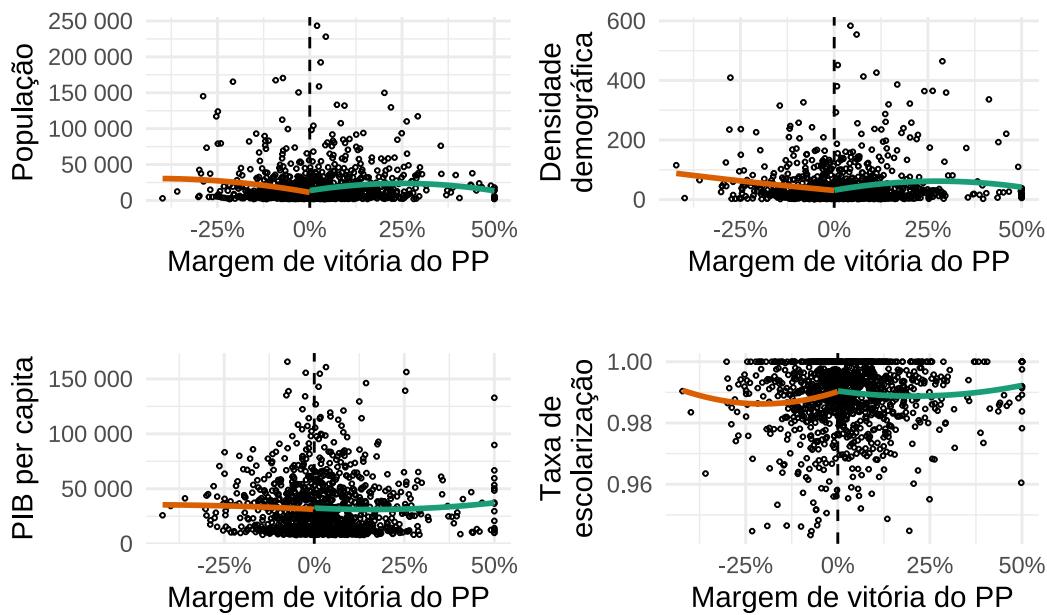
```

ggplot(aes(x = votos_razao_pp_centr, y = PIBpc, color =
  ↵  as.factor(vitoria_pp))) +
geom_point(shape = 21, size = 0.5, color = "black") +
stat_smooth(method = "lm", se = F, formula = y ~ poly(x, 2)) +
scale_color_brewer(palette = "Dark2", direction = -1) +
geom_vline(xintercept = 0, linetype = "dashed") +
scale_y_continuous(labels = label_number()) +
theme_minimal() + guides(color = "none") +
labs(title = "",
      y = "PIB per capita",
      x = "Margem de vitória do PP") + scale_x_continuous(labels =
  ↵  label_percent())

g_taxa_escolarizacao <- dt_todas[taxa_escolarizacao %between%
  ↵  quantile(taxa_escolarizacao, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp_centr, y = taxa_escolarizacao, color =
  ↵  as.factor(vitoria_pp))) +
  geom_point(shape = 21, size = 0.5, color = "black") +
  stat_smooth(method = "lm", se = F, formula = y ~ poly(x, 2)) +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() + guides(color = "none") +
  labs(title = "",
        y = "Taxa de\enescolarização",
        x = "Margem de vitória do PP") + scale_x_continuous(labels =
  ↵  label_percent())

plot_grid(g_pop, g_densidade_demografica, g_PIBpc, g_taxa_escolarizacao, ncol
  ↵  = 2)

```



7.2) Estimação

7.2.1) Obter janela ótima e filtrar bases de dados

Agora, vamos mostrar que os municípios em que o candidato do PP ganhou vs. perdeu por pouco de fato são semelhantes. Para isso, vamos filtrar as bases de dados por eleições acirradas.

```
# Obter janela ótima para modelo quadrático
### Sem covariadas
janela_outra <- rdbwselect(
  x = dt_individuais$votos_razao_pp_centr,
  y = dt_individuais$valor_empenhado_ipca_pc,
  p = 2
)
janela_outra$bws
```

```
h (left) h (right) b (left) b (right)
mserd  0.11978   0.11978  0.1789485 0.1789485
```

```

### Com covariadas
covariadas <- dt_individuais[, .(pop, densidade_demografica, PIBpc,
  ↵ taxa_escolarizacao)]
janela <- rdbwselect(
  x = dt_individuais$votos_razao_pp_centr,
  y = dt_individuais$valor_empenhado_ipca_pc,
  p = 2,
  covs = covariadas
)
janela$bws

```

```

      h (left) h (right)  b (left) b (right)
mserd 0.1450647 0.1450647 0.2261473 0.2261473

```

```

# Filtrar base de dados para municípios cuja margem de vitória do PP está na
  ↵ janela
dt_todas <- dt_todas[votos_razao_pp_centr >= -janela$bws[1] &
                      votos_razao_pp_centr <= janela$bws[1]]
dt_individuais <- dt_individuais[votos_razao_pp_centr >= -janela$bws[1] &
                                     votos_razao_pp_centr <= janela$bws[1]]
dt_bancada <- dt_bancada[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]
dt_relator <- dt_relator[votos_razao_pp_centr >= -janela$bws[1] &
                           votos_razao_pp_centr <= janela$bws[1]]

# Ver quantos municípios com eleições acirradas receberam emendas
writeLines(paste0(
  "Quantidade de municípios com eleições acirradas: ",
  nrow(dt_todas),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas:
  ↵ ",
  nrow(dt_todas[valor_empenhado_ipca != 0]),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas
  ↵ individuais: ",
  nrow(dt_individuais[valor_empenhado_ipca != 0]),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  ↵ bancada: ",
  nrow(dt_bancada[valor_empenhado_ipca != 0]),
  "\nQuantidade de municípios com eleições acirradas que receberam emendas de
  ↵ relator: ",
  nrow(dt_relator[valor_empenhado_ipca != 0])
))

```

```

"\nQuantidade de municípios com eleições acirradas que receberam emendas de
↳ relator: ",
nrow(dt_relator[valor_empenhado_ipca != 0]))

```

Quantidade de municípios com eleições acirradas: 951
 Quantidade de municípios com eleições acirradas que receberam emendas: 86
 Quantidade de municípios com eleições acirradas que receberam emendas individuais: 80
 Quantidade de municípios com eleições acirradas que receberam emendas de bancada: 12
 Quantidade de municípios com eleições acirradas que receberam emendas de relator: 0

- Note que nenhum município com eleições acirradas recebeu emendas de relator e somente 12 receberam emendas de bancada, então vamos analisar somente emendas individuais.

7.2.2) Estimar

Vamos estimar o valor adicional recebido via emendas parlamentares individuais pelos municípios em que o candidato a prefeito do PP ganhou por pouco, usando alguns modelos diferentes.

- Note que, quando o modelo inclui a running variable, é sua versão centralizada (votos_razao_pp_centr), porque isso evita que o intercepto tenha valores sem sentido (como valores empenhados negativos).

```

# Duas regressões quadráticas com controles de município
regs_quad_controles <- lm_robust(valor_empenhado_ipca_pc ~
  ↳ vitoria_pp*votos_razao_pp_centr + vitoria_pp*votos_razao_pp_centr_sq +
  ↳ regiao + pop + densidade_demografica + PIBpc + taxa_escolarizacao, data =
  ↳ dt_individuais)

```

7.2.3) Plotar estimativas

```

library(data.table)
library(ggplot2)
library(scales)

# 1) Filtrar dados para o gráfico
dt_plot <- copy(dt_individuais)[

```

```

  valor_empenhado_ipca_pc %between% quantile(valor_empenhado_ipca_pc, c(.02,
  ↵ .99), na.rm = TRUE)
]

# 2) Escolher valores de referência para os controles
num_vars <- c("pop", "densidade_demografica", "PIBpc", "taxa_escolarizacao")
cat_var  <- "regiao"

refs <- list()
for (v in num_vars) refs[[v]] <- mean(dt_plot[[v]], na.rm = TRUE)
refs[[cat_var]] <- dt_plot[, .N, by = regiao][order(-N)][1, regiao]

# 3) Criar grids separados por lado do cutoff

# esquerda (vitoria = 0)
x_left <- seq(min(dt_plot[vitoria_pp == 0]$votos_razao_pp_centr, na.rm =
  ↵ TRUE),
  0,
  length.out = 200)
grid_left <- data.table(
  vitoria_pp = 0,
  votos_razao_pp_centr = x_left
)

# direita (vitoria = 1)
x_right <- seq(0,
  max(dt_plot[vitoria_pp == 1]$votos_razao_pp_centr, na.rm =
  ↵ TRUE),
  length.out = 200)
grid_right <- data.table(
  vitoria_pp = 1,
  votos_razao_pp_centr = x_right
)

newdata_grid <- rbind(grid_left, grid_right)

# 4) Adicionar quadrático e controles
newdata_grid[, votos_razao_pp_centr_sq := votos_razao_pp_centr^2]

for(v in num_vars){
  newdata_grid[[v]] <- refs[[v]]
}

```

```

newdata_grid[[cat_var]] <- refs[[cat_var]]

# Certificar que fatores têm os mesmos níveis
if(is.factor(dt_individuais$regiao)) {
  newdata_grid[, regiao := factor(regiao, levels =
    ↪ levels(dt_individuais$regiao))]
}

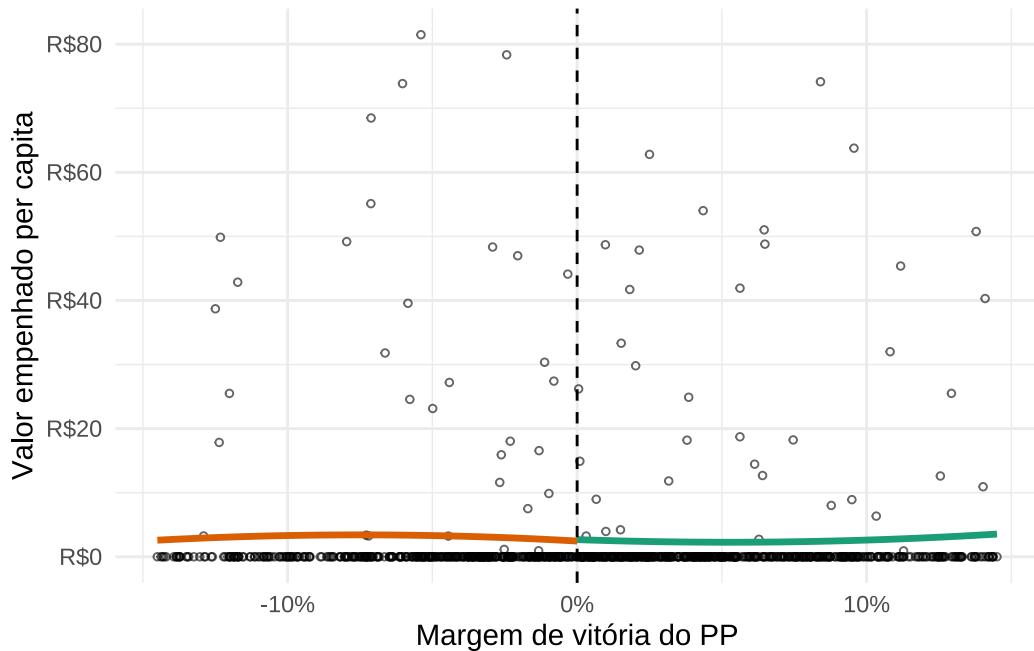
if(is.factor(dt_individuais$vitoria_pp)) {
  newdata_grid[, vitoria_pp := factor(vitoria_pp, levels =
    ↪ levels(dt_individuais$vitoria_pp))]
}

# 5) Criar LM apenas para o gráfico
regs_quad_grafico <- lm(
  valor_empenhado_ipca_pc ~ vitoria_pp*votos_razao_pp_centr +
  ↪ vitoria_pp*votos_razao_pp_centr_sq +
  regiao + pop + densidade_demografica + PIBpc + taxa_escolarizacao,
  data = dt_plot
)

# 6) Previsão
newdata_grid[, fit := predict(regs_quad_grafico, newdata = newdata_grid)]

# 7) Plot final
ggplot() +
  geom_point(data = dt_plot,
    aes(x = votos_razao_pp_centr, y = valor_empenhado_ipca_pc,
        color = as.factor(vitoria_pp)),
    shape = 21, color = "black", alpha = 0.6, size = 1) +
  geom_line(data = newdata_grid,
    aes(x = votos_razao_pp_centr, y = fit, color =
      ↪ as.factor(vitoria_pp)),
    linewidth = 1.2) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  scale_color_brewer(palette = "Dark2", direction = -1) +
  scale_y_continuous(labels = scales::label_currency(prefix = "R$")) +
  scale_x_continuous(labels = scales::label_percent()) +
  theme_minimal() + guides(color = "none") +
  labs(x = "Margem de vitória do PP", y = "Valor empenhado per capita")

```



***Tabela final de resultados**

```

# Resultados - Todos
nomes_variaveis <- c(
  "votos_razao_pp_entr_sq" = "Margem de vitória do PP ao quadrado",
  "votos_razao_pp_entr" = "Margem de vitória do PP",
  "vitoria_pp" = "Prefeito do PP"
)
gm <- tibble::tribble(
  ~raw, ~clean, ~fmt, ~omit,
  "nobs", "Número de observações", 0, FALSE
)
linhas_adicionais <- tribble(~variavel, ~linear_controles, ~linear,
  ~quad_controles,
    'Janela da margem de vitória do PP', '±8,4%', '±10,7%',
    ~ '±14,5%',
    'Forma funcional', "Linear", "Linear", "Quadrático",
    'Controles municipais', "Sim", "Não", "Sim")
modelsummary(list(
  "Linear com controles" = regs_lineares_controles,

```

```
"Linear sem controles" = regs_lineares,
"Quadrático com controles" = regs_quad_controles
),
statistic = NULL,
stars = T,
coef.omit = 4:11,
gof.map = gm,
gof.omit = "R2|AIC|BIC|RMSE",
coef.rename = nomes_variaveis,
add.rows = linhas_adicionais,
output = "Resultados/ValorEmpenhadoPerCapitaIndividuais.png"
)
```