

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 empenhado por razão de votos do candidato a prefeito do PP

0) Preparativos

```
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(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(estimatr) # para modelagem
library(modelsummary) # para visualização de resultados dos modelos
```

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_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 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[, 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_municpios[,  

  PIBpc := as.numeric(  

    gsub(", ", ".", gsub("[^0-9,]", "", PIBpc))
  )
]  

dt_municpios[,  

  taxa_escolarizacao := as.numeric(  

    gsub(", ", ".", gsub("[^0-9,]", "", taxa_escolarizacao)))
  )/100
]  
  

# Dar uma olhada na base  

str(dt_municpios)

```

```

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
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)]
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': 1197 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_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")

```

- 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_municipios[id_municipio_nome == "Restinga Sêca" & sigla_uf == "RS",
               id_municipio_nome := "Restinga Seca"]
dt_municipios[id_municipio_nome == "Vespasiano Corrêa" & sigla_uf == "RS",
               id_municipio_nome := "Vespasiano Correa"]
dt_municipios[id_municipio_nome == "Westfália" & sigla_uf == "RS",
               id_municipio_nome := "Westfalia"]
dt_municipios[id_municipio_nome == "Grão-Pará" & sigla_uf == "SC",
               id_municipio_nome := "Grão Pará"]
dt_municipios[id_municipio_nome == "Lauro Müller" & sigla_uf == "SC",
               id_municipio_nome := "Lauro Muller"]
dt_municipios[id_municipio_nome == "São Cristóvão do Sul" & sigla_uf == "SC",
               id_municipio_nome := "São Cristovão do Sul"]
dt_municipios[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_municipios <- unique(dt_municipios[, paste0(id_municipio_nome,
  ↴ sigla_uf)])
for (muni in municipios_dt_eleicoes) {
  if (!muni %in% municipios_dt_municipios) {
    print(muni)
  }
}
rm(municipios_dt_eleicoes, municipios_dt_municipios, muni)

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

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

```

```

Classes 'data.table' and 'data.frame': 1197 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 "52000050" "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
  ↵ = 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) Verificação de (des)continuidades

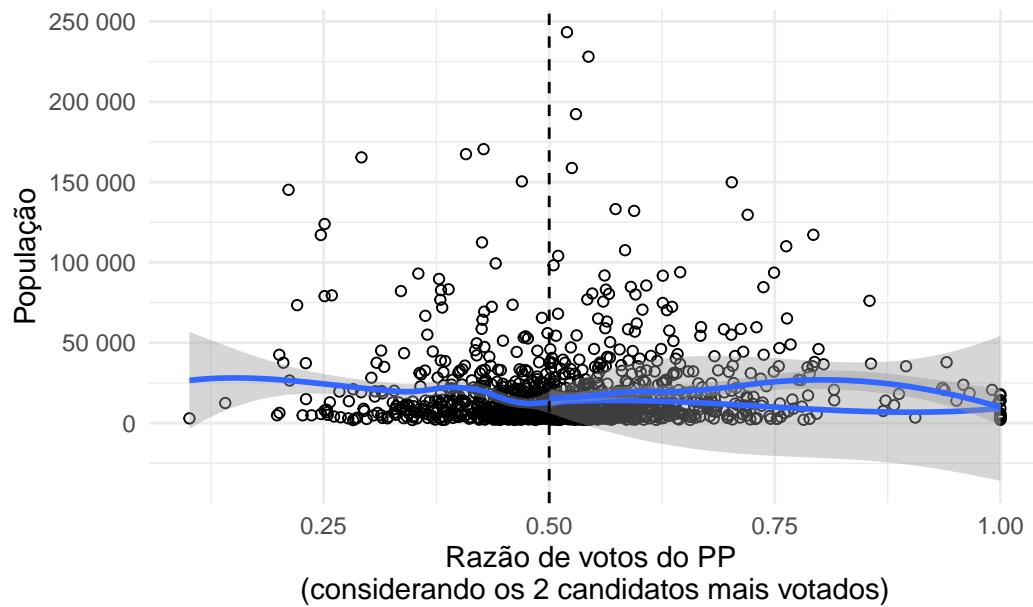
Note que filtramos os municípios para os percentis 2 a 99 de cada característica para evitar distorções causadas por outliers.

2.1.1) Características municipais

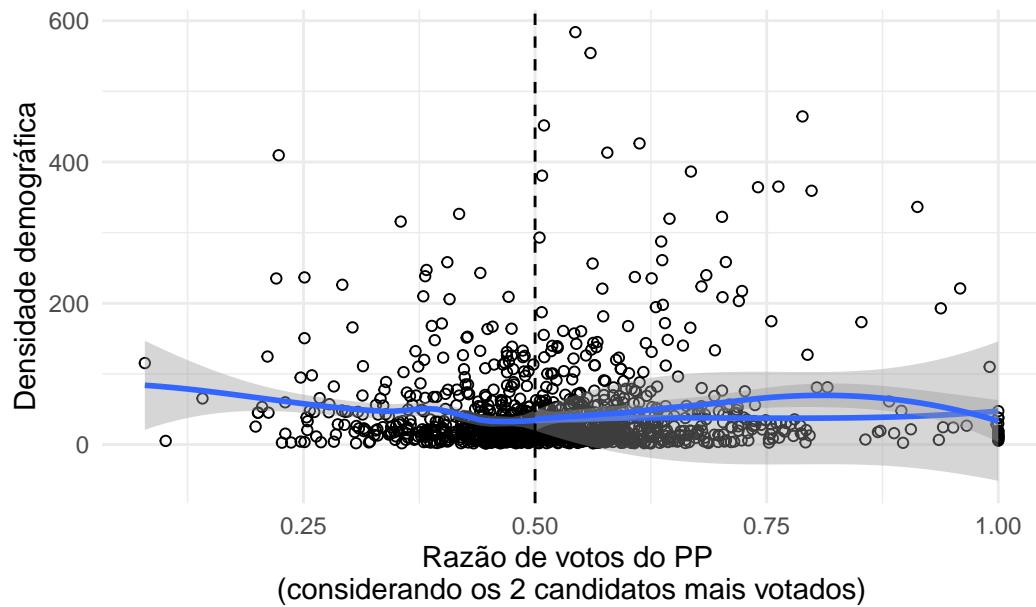
```

dt_todas[pop %between% quantile(pop, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = pop)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "População",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
         ↵ votados)")

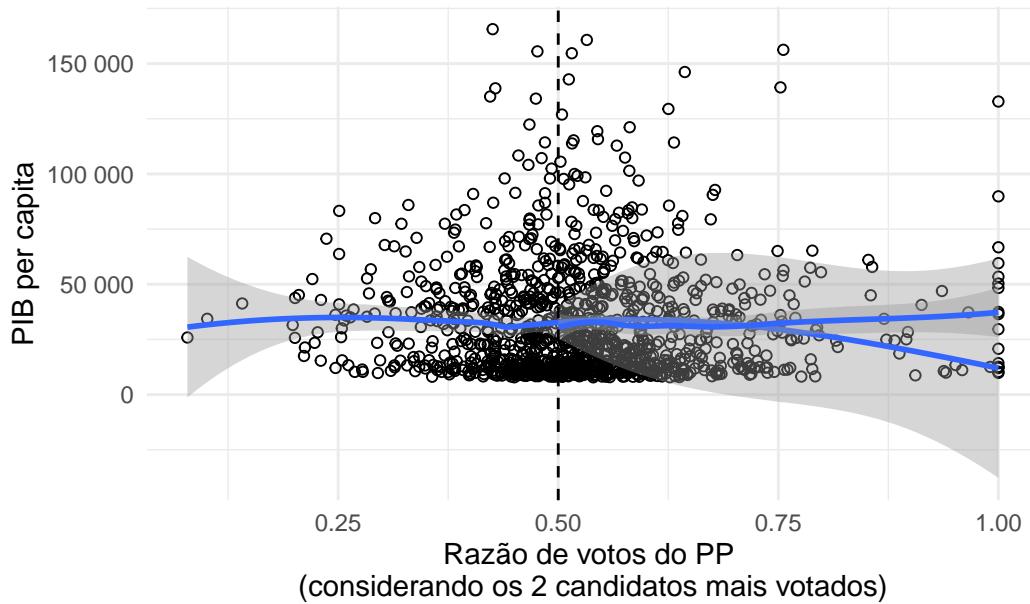
```



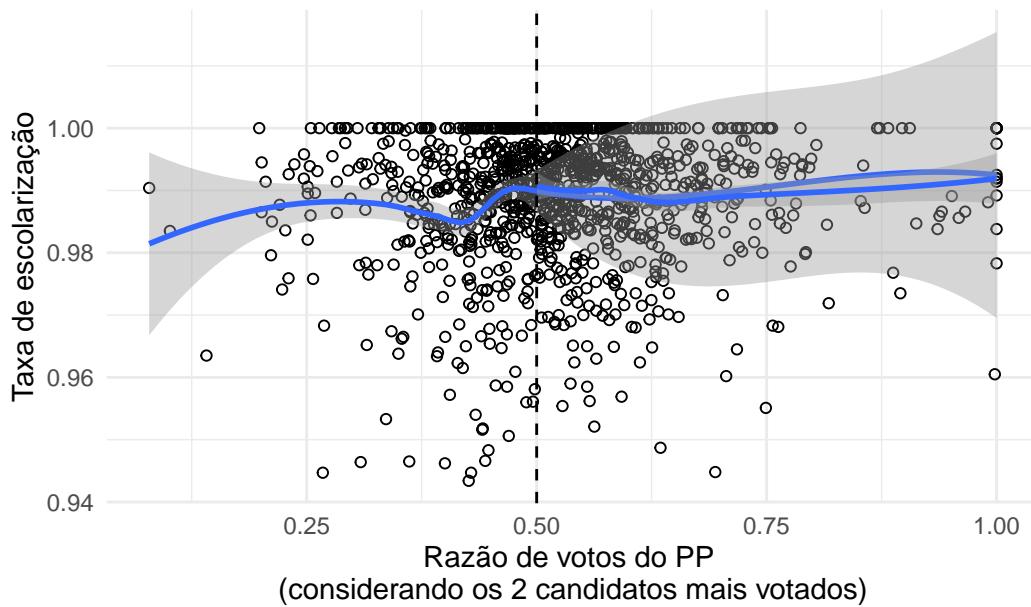
```
dt_todas[densidade_demografica %between% quantile(densidade_demografica,
  ↵ c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = densidade_demografica)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "Densidade demográfica",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
  ↵ votados)")
```



```
dt_todas[PIBpc %between% quantile(PIBpc, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = PIBpc)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "PIB per capita",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
       ↵ votados)")
```

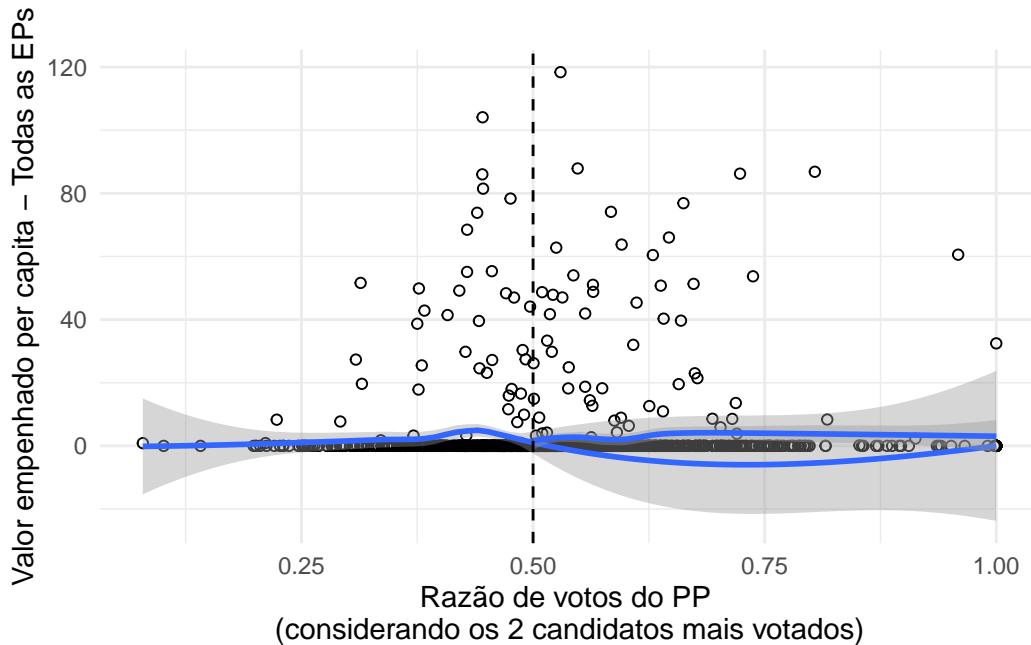


```
dt_todas[taxa_escolarizacao %between% quantile(taxa_escolarizacao, c(.02,
  .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = taxa_escolarizacao)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "Taxa de escolarização",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
  votados)")
```

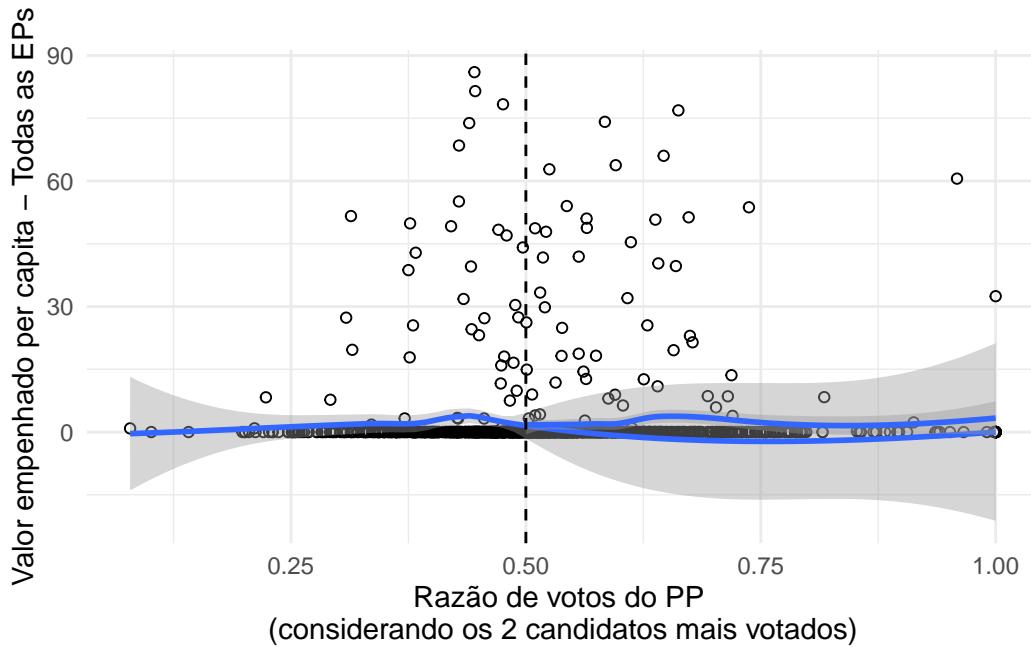


2.2.1) Valor empenhado *per capita* de emendas parlamentares

```
dt_todas[valor_empenhado_ipca_pc %between% quantile(valor_empenhado_ipca_pc,
  ↵ c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = valor_empenhado_ipca_pc)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "Valor empenhado per capita - Todas as EPs",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
  ↵ votados)")
```



```
dt_individuais[valor_empenhado_ipca_pc %between%
  ↪ quantile(valor_empenhado_ipca_pc, c(.02, .99), na.rm = TRUE)] %>%
  ggplot(aes(x = votos_razao_pp, y = valor_empenhado_ipca_pc)) +
  geom_point(shape = 21) +
  stat_smooth(aes(group = as.factor(vitoria_pp)), method = "loess") +
  geom_vline(xintercept = 0.5, linetype = "dashed") +
  scale_y_continuous(labels = label_number()) +
  theme_minimal() +
  labs(title = "",
       y = "Valor empenhado per capita – Todas as EPs",
       x = "Razão de votos do PP \n(considerando os 2 candidatos mais
  ↪ votados)")
```



2.2) Comparação entre tratados e controles

2.2.0) 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.

```
# Filtrar bases de dados para municípios em que a margem de vitória foi de 5%
# ou menos
dt_todas <- dt_todas[votos_razao_pp >= 0.45 & votos_razao_pp <= 0.55]
dt_individuais <- dt_individuais[votos_razao_pp >= 0.45 & votos_razao_pp <=
# 0.55]
dt_bancada <- dt_bancada[votos_razao_pp >= 0.45 & votos_razao_pp <= 0.55]
dt_relator <- dt_relator[votos_razao_pp >= 0.45 & votos_razao_pp <= 0.55]

# 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: 483
 Quantidade de municípios com eleições acirradas que receberam emendas: 39
 Quantidade de municípios com eleições acirradas que receberam emendas individuais: 36
 Quantidade de municípios com eleições acirradas que receberam emendas de bancada: 5
 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 5 receberam emendas de bancada, então vamos analisar somente emendas individuais.

2.2.1) Características municipais

2.2.1.1) Via médias

```

# Criar dt com médias
medias_municipios_eleicoes <- dt_municipios_eleicoes [
  ,
  .(
    media_pop = mean(pop),
    media_densidade_demografica = mean(densidade_demografica),
    media_PIBpc = mean(PIBpc),
    media_taxa_escolarizacao = mean(taxa_escolarizacao),
    media_mortalidade_infantil = mean(mortalidade_infantil, na.rm = T)
  ),
  by = vitoria_pp
]

# Plotar médias de quem venceu vs. não venceu - formatar

```

```

titulo <- ggplot() +
  labs(title = "Comparação das médias de características municipais",
       subtitle = "Fontes dos dados: TSE, IBGE (2021, 2022). Elaboração
       própria.") +
  theme_minimal()

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

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

g3 <- medias_municpios_eleicoes %>%
  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_municpios_eleicoes %>%
  ggplot(aes(y = media_taxa_escolarizacao, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "Taxa de escolarização",
       x = "Prefeito do PP?")

g5 <- medias_municpios_eleicoes %>%
  ggplot(aes(y = media_mortalidade_infantil, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +

```

```

  labs(title = "",
       y = "Mortalidade infantil",
       x = "Prefeito do PP?")

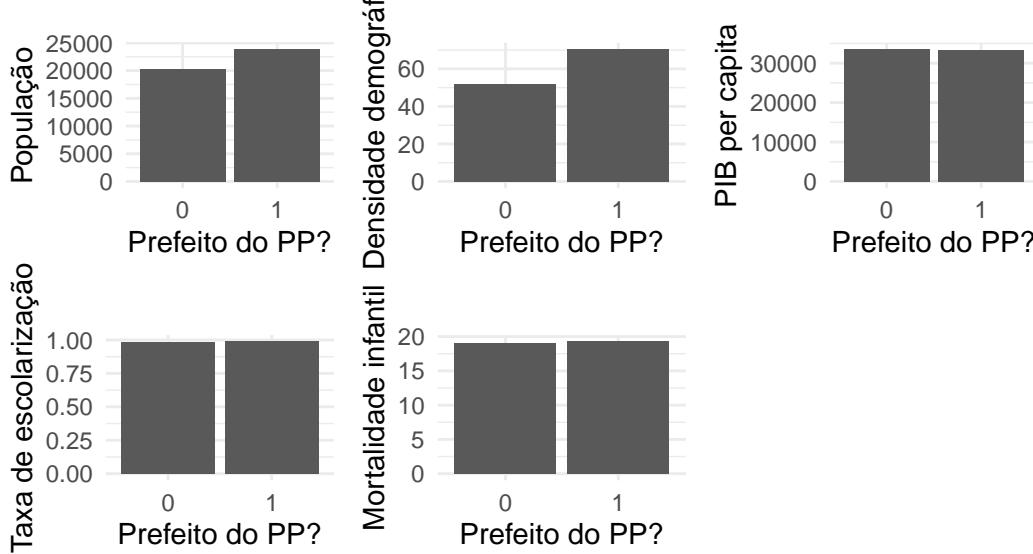
graficos <- plot_grid(g1, g2, g3, g4, g5)

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

```

Comparação das médias de características municipais

Fontes dos dados: TSE, IBGE (2021, 2022). Elaboração própria.



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

```
# Mostrar quais diferenças são estatisticamente significantes - fazer teste
# de comparação des
```

- As médias de população e densidade demográfica são bastante diferentes entre os dois grupos. Vamos olhar para a mediana, que reduz o efeito de outliers:

2.2.1.1) Via medianas

```

# Criar dt com medianas
medianas_municipios_eleicoes <- dt_municipios_eleicoes[
  ,
  ]

```

```

.(

  mediana_pop = median(pop),
  mediana_densidade_demografica = median(densidade_demografica),
  mediana_PIBpc = median(PIBpc),
  mediana_taxa_escolarizacao = median(taxa_escolarizacao),
  mediana_mortalidade_infantil = median(mortalidade_infantil, na.rm = T)
),

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

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

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

g4 <- medianas_municipios_eleicoes %>%

```

```

ggplot(aes(y = mediana_taxa_escolarizacao, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "Taxa de escolarização",
       x = "Prefeito do PP?")

g5 <- medianas_municipios_eleicoes %>%
  ggplot(aes(y = mediana_mortalidade_infantil, x = as.factor(vitoria_pp))) +
  geom_col() +
  theme_minimal() +
  labs(title = "",
       y = "Mortalidade infantil",
       x = "Prefeito do PP?")

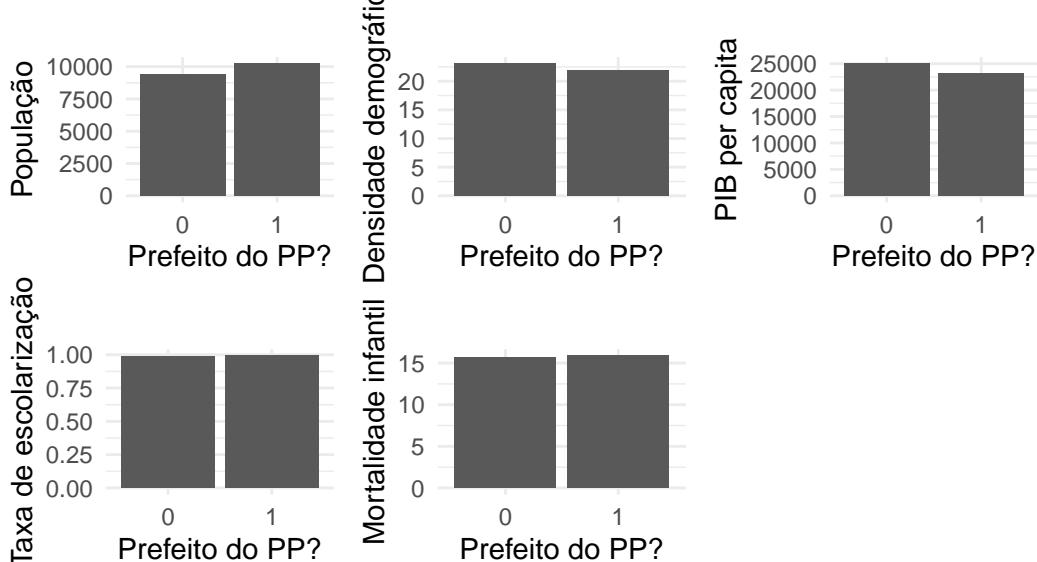
graficos <- plot_grid(g1, g2, g3, g4, g5)

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

```

Comparação das medianas de características municipais

Fontes dos dados: TSE (2021), IBGE (2021, 2022). Elaboração própria.



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

- Usando a mediana, as médias de população e densidade demográfica ficaram bem mais

parecidas.

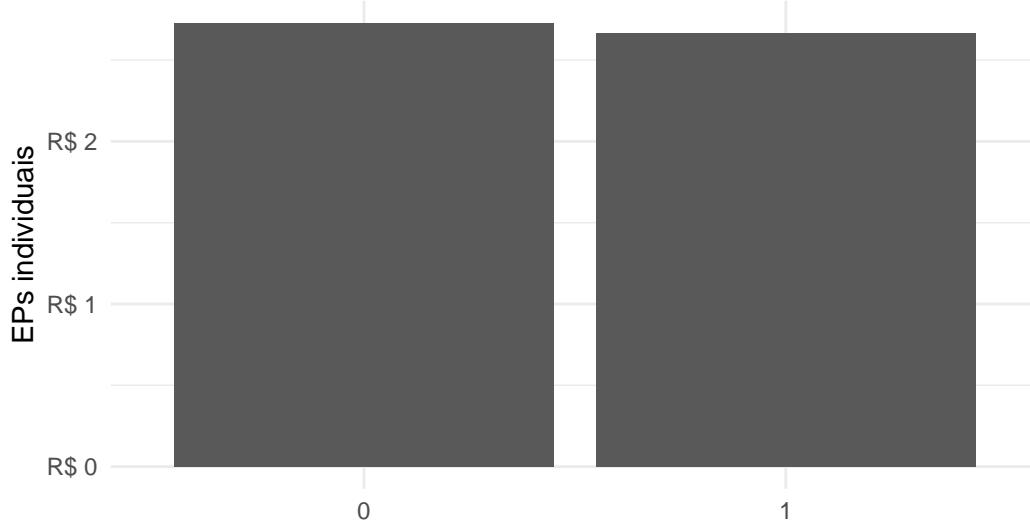
2.2.2) Valores empenhados de emendas parlamentares

Vamos plotar a média dos valores empenhados per capita dos municípios que tiveram eleições acirradas e elegeram ou não o candidato do PP, o que seria equivalente a visualizar o modelo mais simples de RDD.

```
# Criar dt com média por grupo
media_empenho_individuais <- dt_individuais[,
  .(media_empenho = mean(valor_empenhado_ipca_pc)),
  by = vitoria_pp]

# Plotar diferença em valores empenhados pelo resultado da eleição
media_empenho_individuais %>%
  ggplot(aes(y = media_empenho, x = as.factor(vitoria_pp))) +
  geom_col() +
  scale_y_continuous(labels = label_currency(prefix = "R$ ")) +
  theme_minimal() +
  labs(title = "Comparaçao de médias de valor empenhado per capita
  municipal",
       caption = "Fontes dos dados: TSE (2021), SIGA Brasil (2021)",
       y = "EPs individuais",
       x = "Prefeito do PP?")
```

Comparação de médias de valor empenhado per capita municipal



Fontes dos dados: TSE (2021), SIGA Brasil (2021)

```
rm(media_empenho_individuais)
```

- Note que aquelas poucas emendas de bancada distorcem bastante a média do valor empenhado por todas as emendas.
- Note que não faz sentido olhar para medianas porque seriam simplesmente 0 para os dois grupos.

3) 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 controle de razão de votos do PP
regs_lineares <- lm_robust(valor_empenhado_ipca_pc ~
  ↵ vitoria_pp*votos_razao_pp_centr, data = dt_individuais)
```

```

# Duas regressões com linearidade flexibilizada e controle de razão de votos
# do PP
regs_flexiveis <- lm_robust(valor_empenhado_ipca_pc ~
  ~ vitoria_pp*votos_razao_pp_centr + vitoria_pp*votos_razao_pp_centr_sq,
  ~ data = dt_individuais)

# Duas regressões lineares com controles
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)

# Duas regressões com linearidade flexibilizada e controles
regs_flexiveis_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)

# Resultados
modelsummary(list(
  "Duas regressões lineares" = regs_lineares,
  "Duas regressões não lineares" = regs_flexiveis,
  "Duas regressões lineares com controles" = regs_lineares_controles,
  "Duas regressões não lineares com controles" = regs_flexiveis_controles
),
  statistic = NULL, stars = T, gofomit = "R2|AIC|BIC|RMSE",
  output = "Resultados/ValorEmpenhadoPerCapitaIndividuais.png"
)

```

4) Teste de heterogeneidade

Vamos repetir a estimação por região, mas precisamos prestar atenção no número de observações e de tratados.

4.1) Norte

4.2) Nordeste

4.3) Sudeste

4.4) Sul

4.5) Centro-Oeste

5) Teste placebo

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