Tarefa III de survey

Felipe Lamarca

2025-09-06

Nesta tarefa, utilizo dados do Estudo Eleitoral Brasileiro (ESEB) de 2022 para implementar duas abordagens de pós-ajuste simples: pós-estratificação e rake. O objetivo é avaliar em que medida essas estratégias de ponderação, aplicadas a uma amostra coletada através de um desenho amostral complexo, melhoram a qualidade das estimativas dos parâmetros populacionais. Observamos que ambas as abordagens corrigem parte das discrepâncias entre amostra e população. Embora os ajustes sejam geralmente sutis, em alguns casos as diferenças são substantivamente relevantes, especialmente em contextos de disputa acirrada, como o da eleição de 2022. Por fim, discute-se a possibilidadede incluir variáveis adicionais, como religião ou perfil político, para refinar futuras estratégias de ponderação.

Introdução

Em desenhos amostrais simples, como o desenho AAS, a estimação da média e da variância é *straightforward* – afinal, conhecemos as probabilidades de inclusão de cada indivíduo da população na amostra. Por outro lado, desenhos amostrais simples como o AAS são raros por uma série de questões, incluindo logísticas, de custo, etc. Na prática, utilizamos desenhos bem mais complexos e, frequentemente, não inteiramente probabilísticos.

Com efeito, temos probabilidades de inclusão desiguais e a amostra coletada acaba por não espelhar perfeitamente as características da população. Para lidar com isso, utilizamos estratégias de ponderação: a partir de variáveis auxiliares, estimamos pesos que nos permitem aproximar os totais amostrais dos totais populacionais (Lumley 2011; Wolf et al. 2016). Nesta tarefa, trabalho com a amostra do Estudo Eleitoral Brasileiro (ESEB) de 2022, realizada pelo Cesop e pela Quaest. Ela foi coletada com um desenho de *area-sampling* com quotas em três estágios: sorteio de municípios e setores, por PPT; e seleção de pessoas entrevistadas por quotas. Dado esse desenho completo, implemento duas abordagens de pós-ajuste simples e independentes, que comparo na seção de resultados: pós-estratificação e rake.

Esta tarefa está dividida em algumas seções, além desta introdução. A seguir, na seção de *setup*, faço uma série de manipulações nas bases utilizadas na tarefa; depois, apresento a metodologia em duas partes: na primeira, comparo as distribuições de variáveis demográficas na amostra e na população para escolher quais delas serão utilizadas no pós-ajuste; na segunda, implemento as duas abordagens de pós-ajuste. Por fim, nos resultados, comparo as estimativas obtidas a partir de cada estratégia de ponderação com o parâmetro populacional verdadeiro (quando conhecido, é claro) e com a estimativa obtida a partir do uso da amostra sem ponderação. Os resultados mostram que ambas as abordagens melhoram a qualidade das estimativas, ainda que com ganhos sutis em algumas variáveis e ganhos mais relevantes em outras.

Setup da tarefa e manipulação dos dados

No *chunk* de código abaixo, importo todas as bibliotecas necessárias para a realização da tarefa, defino uma `seed` arbitrariamente escolhida e faço a leitura dos dados. No caso do banco de dados amostral, importamos uma base fornecida na descrição da tarefa; no caso do censo de 2022, importamos os dados por intermédio da biblioteca `sidrar`.

```
# pacotes utilizados
library(modelsummary)
library(tidyverse)
library(gt)
library(sidrar)
library(stringr)
library(stringi)
library(survey)
library(patchwork)

# seed
set.seed(42)

# dados do eseb (amostra)
eseb22 <- read.csv("data/eseb22.csv")

# dados do censo (via sidrar)
censo22 <- get_sidra(
  api = paste0(
    "/t/10061/n3/all/v/allxp/p/all/",
    "c1568/allxt/",
    "c58/1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,1155,2503,100052/",
    "c2/allxt/",
    "c86/allxt"
  )
)
```

Note que os formatos das tabelas são distintos. Na base do ESEB, cada linha representa um respondente; na base do censo, por outro lado, as informações são agregadas, o que significa que cada linha representa uma combinação de uma série de variáveis sociodemográficas, e a coluna `Valor` indica o número de indivíduos da população brasileira que se enquadram naquela particular combinação.

Uma etapa importante e que deve ser realizada antes da etapa de ponderação é o tratamento das bases de dados. Em particular, para realizar os pós-ajustes de maneira adequada, é importante que as categorias das variáveis auxiliares sejam compatíveis entre as duas bases. Por conta disso, empreendemos uma série de tratamentos, incluindo desde a filtragem de colunas até a recodificação de variáveis sociodemográficas. No caso do censo de 2022 foi necessário, por exemplo, recodificar as categorias das variáveis de idade, cor/raça e sexo. Já no caso do ESEB, recodificamos algumas variáveis para facilitar a interpretação dos resultados. Removemos também a parte da amostra composta por menores de 18 anos, já que a base do censo extraída do censo não dispõe dessa informação

```r
# manutencao de apenas algumas colunas
censo22 <- censo22 %>%
  select(
    "Unidade da Federação",
    "Ano",
    "Nível de instrução",
    "Grupo de idade",
    "Sexo",
    "Cor ou raça",
    "Valor"
    )

# renomeacao das variaveis para manter consistencia
censo22 <- censo22 %>%
  rename(
    "uf" = "Unidade da Federação",
    "escolaridade" = "Nível de instrução",
    "sexo" = "Sexo",
    "cor_raca" = "Cor ou raça",
    "idade" = "Grupo de idade"
  )

# recodificacao das variaveis de idade, cor_raca e sexo
censo22 <- censo22 %>%
  mutate(
    idade = case_when(
      idade == "18 a 24 anos" ~ "18 a 24 anos",
      idade %in% c("25 a 29 anos", "30 a 34 anos") ~ "25 a 34 anos",
      idade %in% c("35 a 39 anos", "40 a 44 anos") ~ "35 a 44 anos",
      idade %in% c("45 a 49 anos", "50 a 54 anos",
                   "55 a 59 anos") ~ "45 a 59 anos",
      TRUE ~ "60 anos ou mais"
    ),
    cor_raca = case_when(
      cor_raca %in% c("Amarela", "Indígena") ~ "Outras",
      TRUE ~ cor_raca
    ),
    sexo = case_when(
      sexo == "Homens" ~ "Masculino",
      sexo == "Mulheres" ~ "Feminino"
    )
  )

# recodificacao das variaveis para facilitar interpretacao
eseb22 <- eseb22 %>%
  mutate(
    confianca_governo = as.character(confianca_governo),
```

```
    confianca_judiciario = as.character(confianca_judiciario),
    voto_2t = as.character(voto_2t)
  ) %>%
  mutate(
    confianca_governo = case_when(
      confianca_governo == "1" ~ "Muita confiança",
      confianca_governo == "2" ~ "Alguma confiança",
      confianca_governo == "3" ~ "Pouca confiança",
      confianca_governo == "4" ~ "Nenhuma confiança",
      TRUE ~ confianca_governo
    ),
    confianca_judiciario = case_when(
      confianca_judiciario == "1" ~ "Muita confiança",
      confianca_judiciario == "2" ~ "Alguma confiança",
      confianca_judiciario == "3" ~ "Pouca confiança",
      confianca_judiciario == "4" ~ "Nenhuma confiança",
      TRUE ~ confianca_judiciario
    ),
    voto_2t = case_when(
      voto_2t == "1" ~ "Bolsonaro",
      voto_2t == "2" ~ "Lula",
      TRUE ~ voto_2t
    )
  )

# removemos menores de 18 anos, porque nao temos informacao na base do SIDRA
eseb22 <- eseb22 %>%
  filter(idade != "16 e 17 anos")
```

Vale destacar aqui que as categorias da variável de escolaridade do ESEB eram incompatíveis com as categorias do censo. Consequentemente, não incluímos essa variável entre as variáveis de ajuste, o que tornou desnecessária qualquer manipulação dessa variável. Poderíamos, é claro, ter extraído a distribuição marginal dos dados educacionais de alguma outra fonte de dados, como a PNAD, e incorporado essa distribuição usando `rake` como procedimento de pós-ajuste. Mas, para manter a simplicidade, optou-se por ignorar essa variável na ponderação.

No mais, vale esclarecer também quais variáveis as bases nos permitem comparar diretamente. São elas: `idade`, `sexo`, `cor_raca`, `uf`, e suas combinações. Embora o ESEB colete dados de religião e os dados do censo de 2022 a esse respeito já estejam disponíveis, também seria necessário utilizar a distribuição marginal via `rake`.

Uma última manipulação foi o ajuste dos nomes dos estados, que foram normalizados para permitir uma comparação mais direta entre as duas bases.

```
# converte o texto para minusculo e remove acentos
normalizar_uf <- function(x) {
  x %>%
    str_to_lower() %>%                # converte para minúsculas
```

```
    stri_trans_general("Latin-ASCII")  # remove acentos
}

# aplica em uf
eseb22 <- eseb22 %>%
  mutate(uf = normalizar_uf(uf))

# aplica em uf
censo22 <- censo22 %>%
  mutate(uf = normalizar_uf(uf))
```

Metodologia

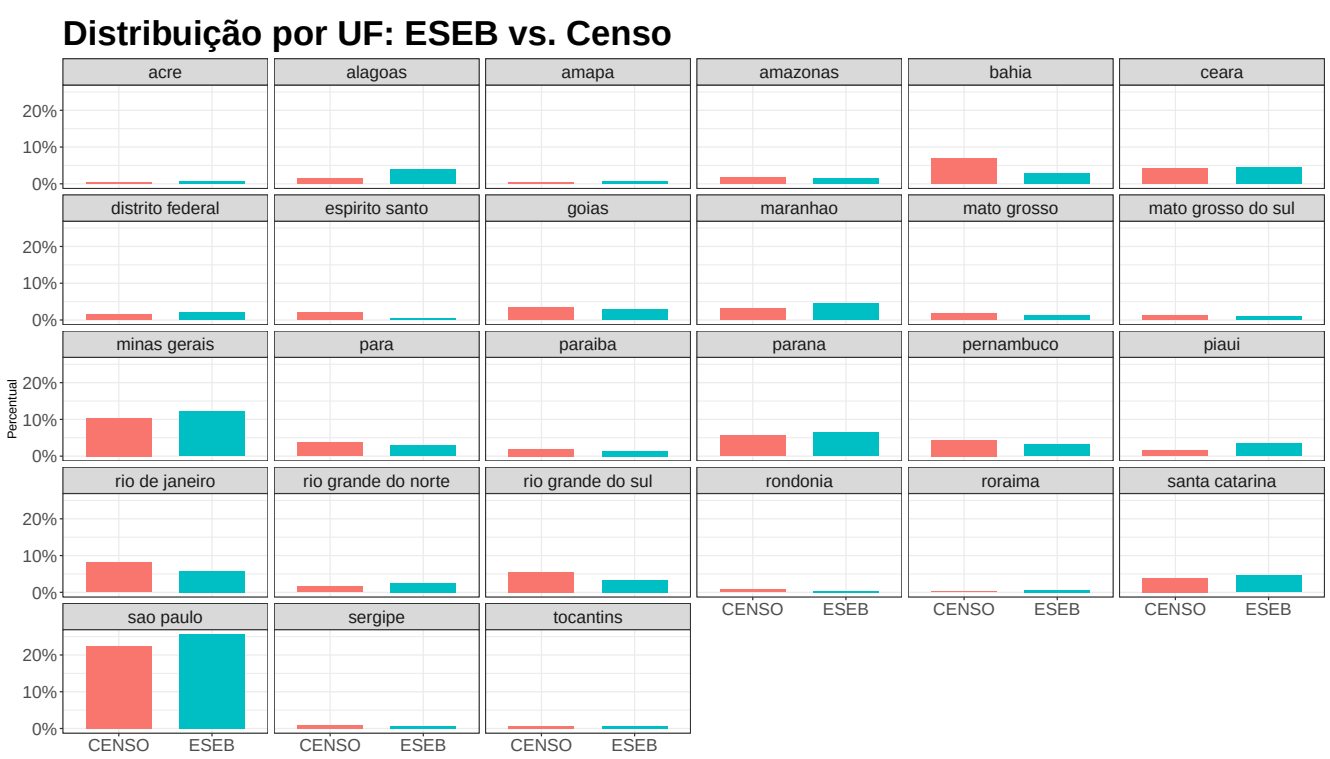
Quais variáveis incluir no pós-ajuste?

Antes de partir para o pós-ajuste e o cálculo dos pesos, é necessário, é claro, definir quais variáveis serão utilizadas para ponderar a amostra. Não se trata de uma escolha arbitrária: devemos, de saída, comparar na amostra e na população as distribuições marginais e conjuntas de variáveis candidatas à ponderação. Caso as distribuições sejam significativamente distintas para uma dada variável, temos aí uma boa candidata a ser incluída no procedimento de ponderação.

Vamos começar pelos estados. A ideia é simples: calculamos a proporção da amostra e da população em cada estado, juntamos as duas bases agrupadas e comparamos as proporções usando um único gráfico.

```
# ----- CENSO: total por UF -----
pop_censo <- censo22 %>%
  group_by(uf) %>%
  summarise(n = sum(Valor, na.rm = TRUE), .groups = "drop") %>%
  mutate(Freq = n / sum(n), origem = "CENSO")

# ----- ESEB: total por UF -----
pop_eseb <- eseb22 %>%
  group_by(uf) %>%
  summarise(n = n(), .groups = "drop") %>%
  mutate(Freq = n / sum(n), origem = "ESEB")

# ----- Junta -----
pop_comparado <- bind_rows(pop_censo, pop_eseb)

# ----- Gráfico facetado -----
ggplot(pop_comparado, aes(x = origem, y = Freq, fill = origem)) +
  geom_col(position = "dodge", width = 0.7) +
  facet_wrap(~ uf, ncol = 6) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(
    title = "Distribuição por UF: ESEB vs. Censo",
```

```
    x = NULL,
    y = "Percentual"
  ) +
  theme_bw() +
  theme(
    legend.position = "none",
    axis.text.x = element_text(size = 15),
    axis.text.y = element_text(size = 15),
    strip.text = element_text(size = 15),
    legend.title = element_text(size = 15),
    legend.text = element_text(size = 15),
    plot.title = element_text(size = 30, face = "bold"),
  )
```



Há algumas informações importantes a serem extraídas aqui. Em primeiro lugar, chama atenção o fato de que o estado de São Paulo parece estar sobre-representado na amostra da pesquisa do ESEB quando comparamos com o censo. O mesmo acontece com Minas Gerais e Paraná, por exemplo. Depois, observamos que outros estados parecem estar sub-representados, como o Rio de Janeiro, a Bahia e o Rio Grande do Sul. De fato, isso é uma evidência de que a variável `uf` deveria ser levada em conta no processo de ponderação.

Para facilitar o processo de comparação das distribuições das demais variáveis, criei a função descrita abaixo. Ela recebe uma variável sociodemográfica qualquer e retorna dois gráficos: o primeiro com a distribuição marginal dessa variável no censo e na amostra do ESEB; e o segundo com a distribuição dessa variável separada entre os estados, também no censo e na amostra do ESEB. Espero, com isso, agregar mais informação à escolha das variáveis de ponderação.

```r
comparar_variavel <- function(var) {

  var_str <- deparse(substitute(var))

  # ------------ ESEB - Brasil ------------
  eseb_br <- eseb22 %>%
    group_by(.data[[var_str]]) %>%
    summarise(n = n(), .groups = "drop") %>%
    mutate(Freq = n / sum(n),
           origem = "ESEB")

  # ------------ CENSO - Brasil ------------
  censo_br <- censo22 %>%
    group_by(.data[[var_str]]) %>%
    summarise(n = sum(Valor, na.rm = TRUE), .groups = "drop") %>%
    mutate(Freq = n / sum(n),
           origem = "CENSO")

  brasil <- bind_rows(eseb_br, censo_br)

  # ------------ ESEB - UF ------------
  eseb_uf <- eseb22 %>%
    group_by(uf, .data[[var_str]]) %>%
    summarise(n = n(), .groups = "drop") %>%
    group_by(uf) %>%
    mutate(Freq = n / sum(n),
           origem = "ESEB")

  # ------------ CENSO - UF ------------
  censo_uf <- censo22 %>%
    select(uf, all_of(var_str), Valor) %>%
    drop_na() %>%
    group_by(uf, .data[[var_str]]) %>%
    summarise(n = sum(Valor), .groups = "drop") %>%
    group_by(uf) %>%
    mutate(Freq = n / sum(n),
           origem = "CENSO")

  uf_all <- bind_rows(eseb_uf, censo_uf)

  # ------------ Gráfico Brasil ------------
  p_brasil <- ggplot(brasil, aes(x = .data[[var_str]], y = Freq, fill = origem)) +
    geom_col(position = "dodge") +
    scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
    labs(
      title = paste("Brasil -", var_str),
      x = NULL, y = "Percentual", fill = "Fonte"
```

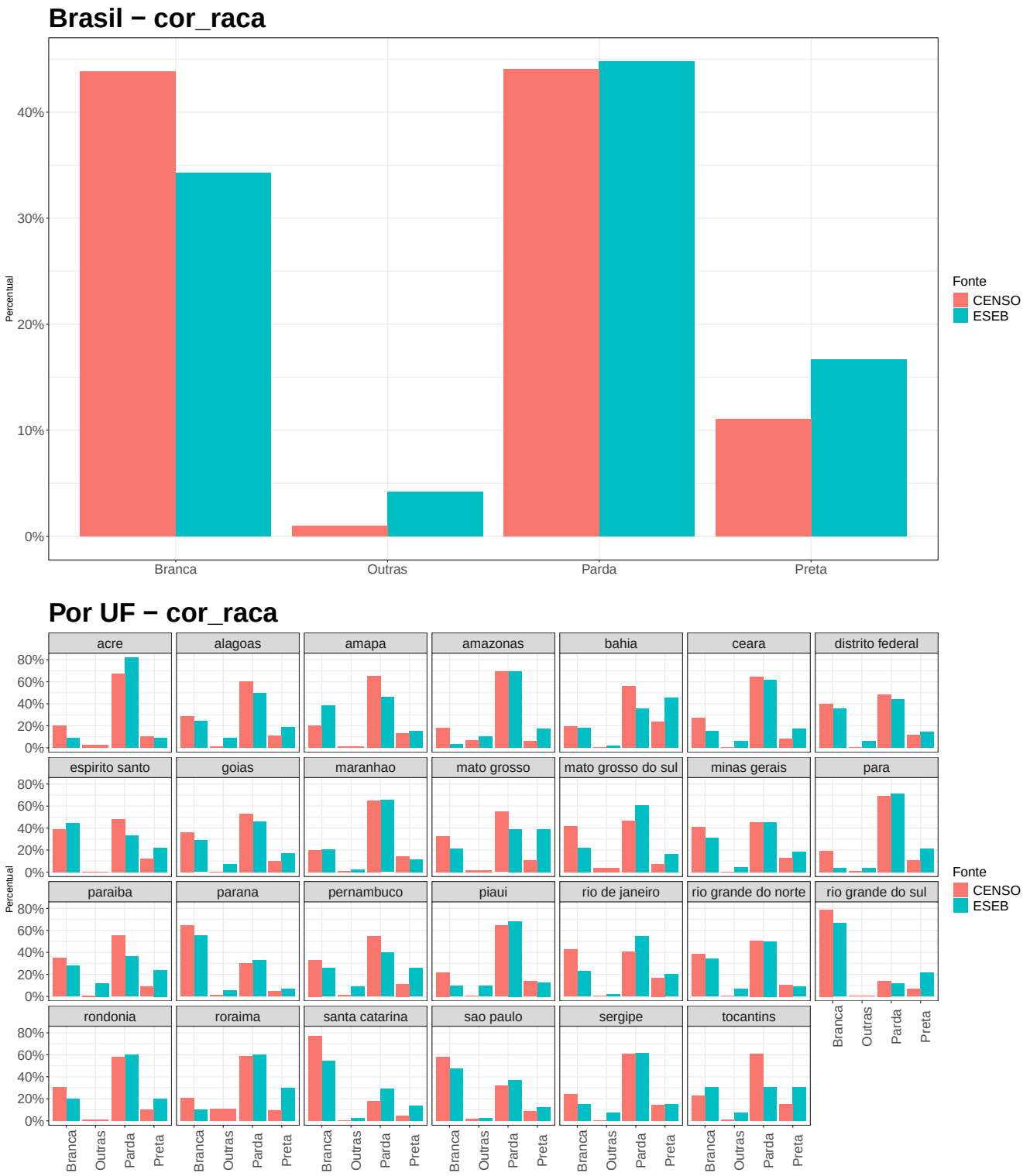
```
    ) +
    theme_bw() +
    theme(
      axis.text.x = element_text(size = 15),
      axis.text.y = element_text(size = 15),
      strip.text = element_text(size = 15),
      legend.title = element_text(size = 15),
      legend.text = element_text(size = 15),
      plot.title = element_text(size = 30, face = "bold")
    )

  # ------------ Gráfico por UF ------------
  p_uf <- ggplot(uf_all, aes(x = .data[[var_str]], y = Freq, fill = origem)) +
    geom_col(position = "dodge") +
    facet_wrap(~ uf, ncol = 7, nrow = 5) +
    scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
    labs(
      title = paste("Por UF -", var_str),
      x = NULL, y = "Percentual", fill = "Fonte"
    ) +
    theme_bw() +
    theme(
      axis.text.x = element_text(size = 15, angle = 90),
      axis.text.y = element_text(size = 15),
      strip.text = element_text(size = 15),
      legend.title = element_text(size = 15),
      legend.text = element_text(size = 15),
      plot.title = element_text(size = 30, face = "bold")
    )

  # ----------------------------------------
  print(p_brasil)
  print(p_uf)

}
```

```
comparar_variavel(idade)
```

A distribuição marginal já mostra algumas diferenças importantes e sistemáticas. Em especial, a amostra do ESEB subestima a população de 60 anos ou mais e superestima a população de 25 a 34 anos. As diferenças são substantivas. Quando observamos os dados no nível dos estados, identificamos que as maiores discrepâncias ocorrem especialmente em estados onde a amostra coletada é menor, como Rondônia, Tocantins e Amapá. A despeito das diferenças ainda ocorrerem, elas são menos aparentes em

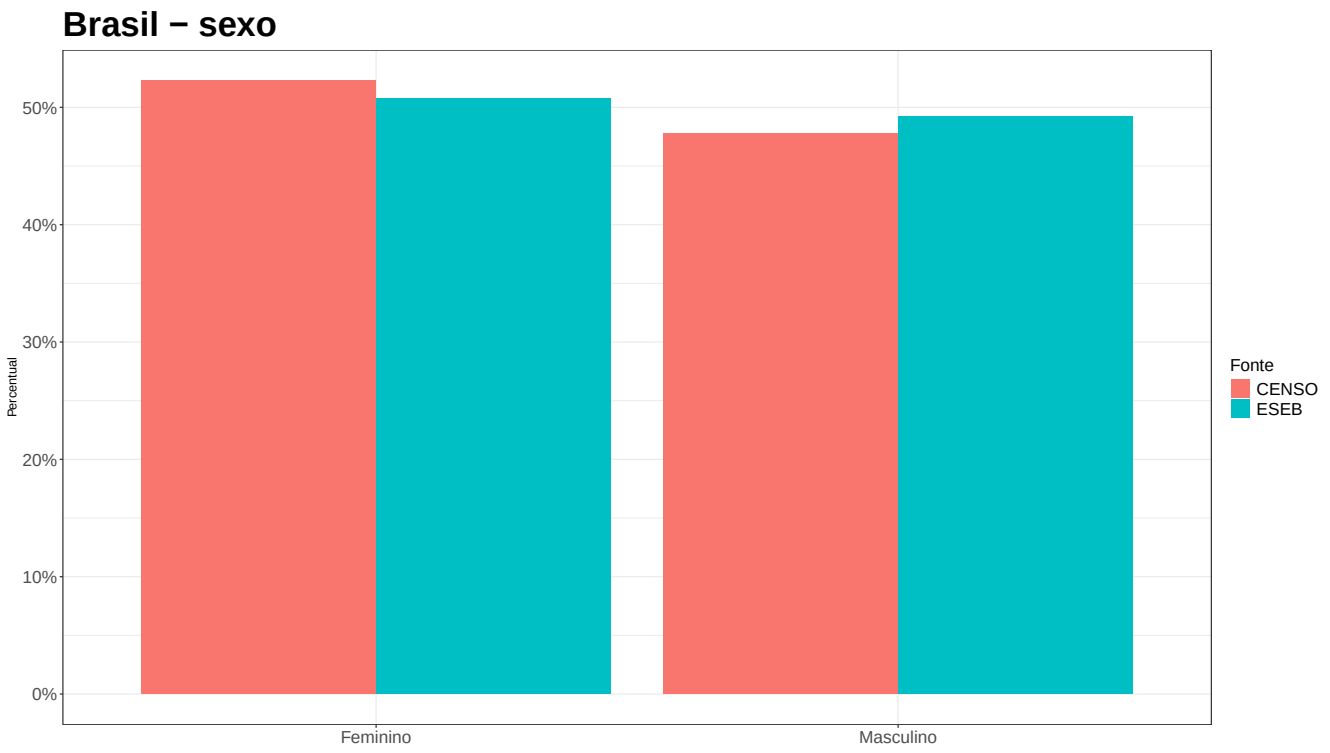
estados maiores, como o Rio de Janeiro. De todo modo, trata-se também de uma variável importante a ser considerada na etapa de pós-ajuste.

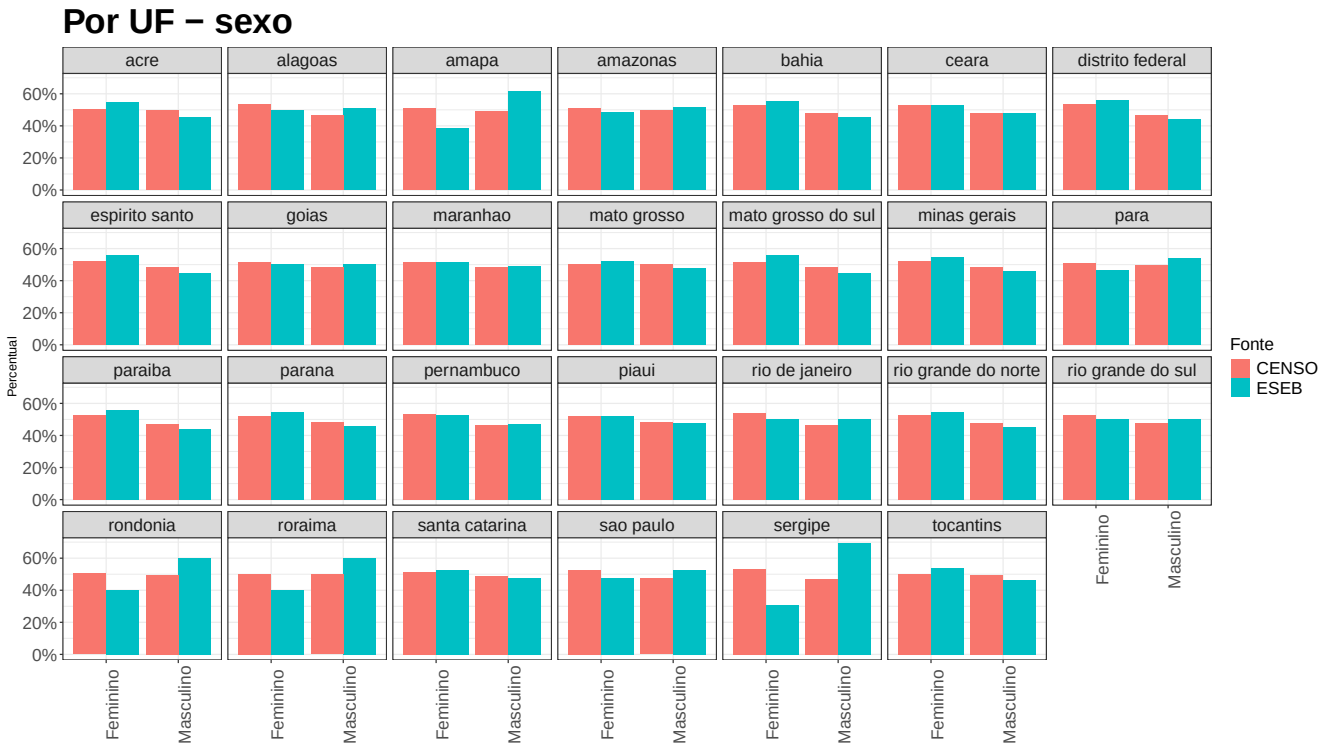
```
comparar_variavel(cor_raca)
```



Algo semelhante ocorre na variável `cor_raca`. No gráfico da distribuição marginal, para todo o Brasil, observamos que a há uma sistemática subestimação da população branca em algo em torno de 9 pontos percentuais, enquanto a população preta é superestimada em aproximadamente 5 pontos percentuais. Outras religiões também são superestimadas. Pela magnitude dessas diferenças, certamente será importante incluir `cor_raca` no pós-ajuste.

```
comparar_variavel(sexo)
```





No caso do sexo, as diferenças são bem menos aparentes – o sexo feminino é levemente subestimado na amostra do ESEB e o sexo masculino é levemente superestimado. As maiores diferenças ocorrem em estados onde a amostra coletada é pequena. Para evitar um pós-ajuste muito complexo (isto é, com muitas variáveis), considerando que já incluiremos `uf`, `cor_raca` e `idade`, deixaremos `sexo` "de lado". Ao corrigir para as demais variáveis, inclusive, não seria uma surpresa que a própria variável de `sexo` seja parcialmente corrigida de maneira natural.

Pós-ajuste

Aqui, por curiosidade científica, extrapolarei levemente o que foi solicitado no enunciado da tarefa e implementarei duas estratégias de ponderação distintas, ao invés de uma só: tanto a pós-estratificação quanto o rake. Nos dois casos, explicarei exatamente as estratégias utilizadas. De antemão, vale reforçar o que já foi indicado na seção anterior: optei por incluir na ponderação as variáveis `idade`, `cor_raca` e `uf`. Na seção de resultados, comparo os resultados de cada abordagem.

No caso da pós-estratificação, a ideia é construir uma base de estratos em que cada linha é uma combinação das variáveis utilizadas na ponderação, com a indicação do total de indivíduos que respeitam cada combinação em uma variável chamada `Freq`. O código abaixo cria essa base e faz a pós-estratificação:

```
# constroi a base de estratos
estratos <- censo22 %>%
  group_by(uf, cor_raca, idade) %>%
  summarise(Freq = sum(Valor, na.rm = TRUE))

# faz a pos-estratificacao
poststrat_design <- postStratify(
```

```
  design = svydesign(ids = ~id_entrevista, data=eseb22),
  strata = ~uf + cor_raca + idade,
  population = estratos,
  partial = TRUE
)
```

Observe que, no código acima, utilizamos o argumento `partial` com o valor `TRUE`. Isso é necessário nos casos em que nem todas as combinações possíveis de estado, cor/raça e idade na população estão presentes na amostra. Há, naturalmente, consequências. Se alguns estratos não existem na amostra, não somos capazes de ponderar essa parte (inexistente, com o perdão da repetição) da amostra para aproximá-la da população. De fato, o que estamos fazendo na prática é pós-estratificando apenas um subconjunto de estratos possíveis.

Conforme descrito pelo próprio Thomas Lumley, criador do pacote `survey` neste post do Cross Validated, há maneiras de contornar isso – por exemplo, colapsando categorias de uma ou várias variáveis. No nosso caso, a variável mais provável de resolver o problema ao ser colapsada seria o estado; seria razoável, em particular, utilizar a região ao invés da unidade federativa. Não fiz testes nesse sentido, e assumo que se trata de uma limitação do processo de ponderação que implementei.

Agora, vamos implementar o rake. Diferente da abordagem de pós-estratificação, o rake é um processo iterativo que procura aproximar as distribuições marginais do survey das distribuições marginais da população. Trata-se de uma abordagem especialmente útil em situações em que não podemos lançar mão das distribuições conjuntas das variáveis de ponderação. Poderíamos, com isso, ter incluído as distribuições marginais da variável de escolaridade e religião, que escolhemos ignorar para simplificar a execução da tarefa. O código abaixo implementa o rake:

```
# distribuicao marginal de uf
pop_uf <- censo22 %>%
  group_by(uf) %>%
  summarise(Freq = sum(Valor, na.rm = TRUE))

# distribuicao marginal de cor_raca
pop_cor <- censo22 %>%
  group_by(cor_raca) %>%
  summarise(Freq = sum(Valor, na.rm = TRUE))

# distribuicao marginal de idade
pop_idade <- censo22 %>%
  group_by(idade) %>%
  summarise(Freq = sum(Valor, na.rm = TRUE))

# rake
rake_design <- rake(
  design = svydesign(ids = ~id_entrevista, data=eseb22),
  sample.margins = list(~uf, ~cor_raca, ~idade),
  population.margins = list(pop_uf, pop_cor, pop_idade)
)
```

Resultados

Implementadas as duas estratégias de ponderação, resta-nos comparar os resultados. Essa etapa será facilitada pela função abaixo, `compare_estimates`. Ela recebe os *designs* de pós-estratificação e rake, os dados amostrais e os dados populacionais. No caso das variáveis demográficas, para as quais conhecemos o parâmetro populacional, a função nos permite incluir a coluna referente à população, apenas para permitir mais um nível de comparação.

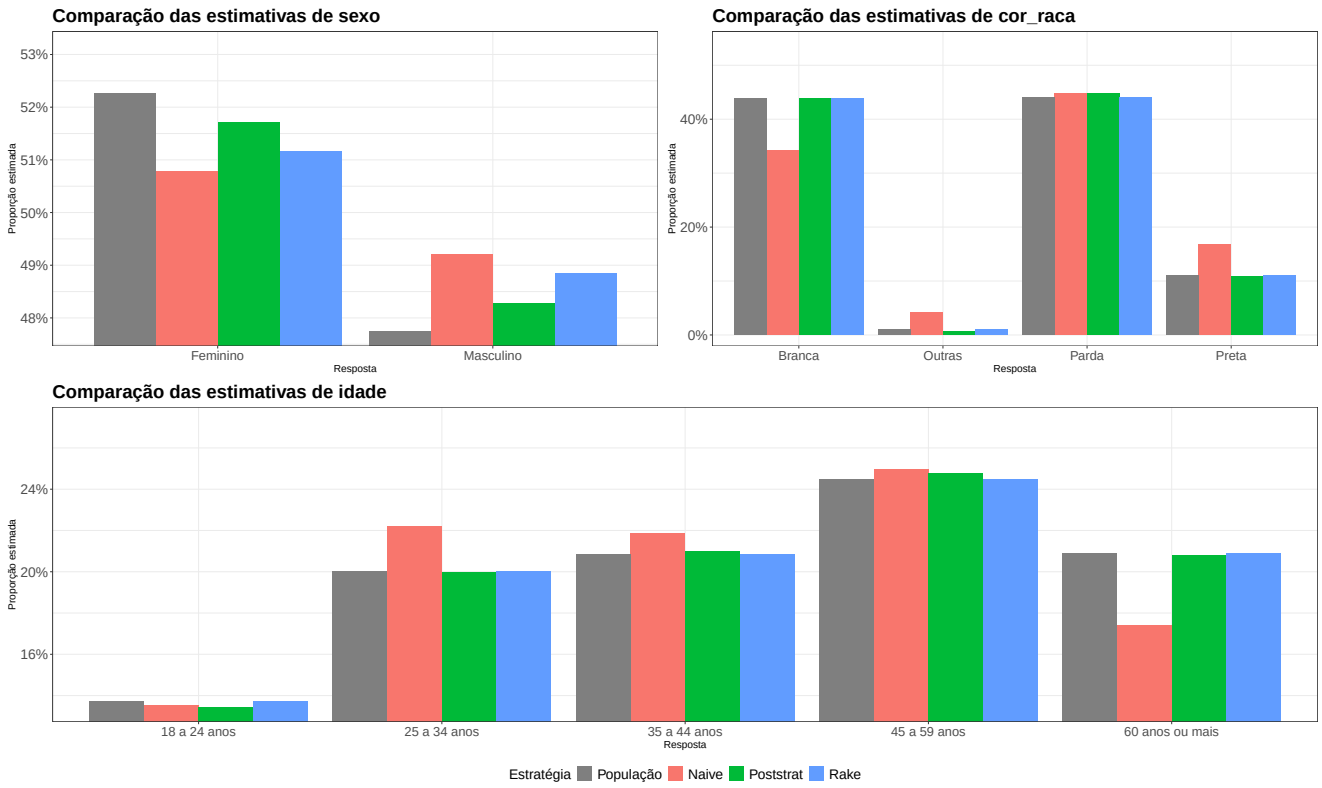
```
compare_estimates <- function(varname,
                              raw_data,
                              poststrat_design,
                              rake_design,
                              ordered_levels,
                              population_data = NULL,
                              show_population = FALSE) {

  var_sym <- sym(varname)

  # Proporções cruas (naive)
  naive <- raw_data %>%
    count(!!var_sym) %>%
    mutate(mean = n / sum(n)) %>%
    select(!!var_sym, mean) %>%
    rename(category = !!var_sym) %>%
    mutate(strategy = "Naive")

  # Função auxiliar para limpar nome da variável dos nomes das categorias
  clean_categories <- function(names_vec, varname) {
    str_replace(names_vec, paste0("^", varname), "")
  }

  # Poststratificado
  post <- svymean(as.formula(paste0("~", varname)), design = poststrat_design)
  post <- tibble(
    category = clean_categories(names(post), varname),
    mean = as.numeric(post),
    strategy = "Poststrat"
  )

  # Rake
  rake <- svymean(as.formula(paste0("~", varname)), design = rake_design)
  rake <- tibble(
    category = clean_categories(names(rake), varname),
    mean = as.numeric(rake),
    strategy = "Rake"
  )
```

```r
# Combina resultados
result <- bind_rows(naive, post, rake)

# Adiciona dados populacionais, se solicitado
if (show_population && !is.null(population_data)) {
  pop <- population_data %>%
    group_by(!!var_sym) %>%
    summarise(Freq = sum(Valor, na.rm = TRUE)) %>%
    mutate(mean = Freq / sum(Freq)) %>%
    select(!!var_sym, mean) %>%
    rename(category = !!var_sym) %>%
    mutate(strategy = "População")

  result <- bind_rows(result, pop)
}

# Ordena categorias e estratégias
result$category <- factor(result$category,
                          levels = ordered_levels, ordered = TRUE)

result$strategy <- factor(result$strategy,
                          levels = c("População", "Naive", "Poststrat", "Rake"))

y_min <- min(result$mean)
y_max <- max(result$mean)
y_padding <- (y_max - y_min) * 0.2

plot <- ggplot(result, aes(x = category, y = mean, fill = strategy)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  scale_fill_manual(
    values = c(
      "População" = "grey50",
      "Naive" = "#F8766D",
      "Poststrat" = "#00BA38",
      "Rake" = "#619CFF"
    )
  ) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  coord_cartesian(ylim = c(y_min, y_max + y_padding)) +
  labs(
    title = paste("Comparação das estimativas de", varname),
    x = "Resposta",
    y = "Proporção estimada",
    fill = "Estratégia"
  ) +
  theme_bw() +
  theme(
```

```
      axis.text.x = element_text(size = 14),
      axis.text.y = element_text(size = 15),
      strip.text = element_text(size = 15),
      legend.title = element_text(size = 15),
      legend.text = element_text(size = 15),
      plot.title = element_text(size = 20, face = "bold")
    )

  return(plot)
}
```

Abaixo, apresentamos os resultados ponderados para as variáveis demográficas. Note de antemão que os eixos y não iniciam necessariamente no 0, uma escolha arbitrária para garantir uma visualização mais eficiente das diferenças entre as abordagens. A desvantagem é que a magnitude da diferença deve ser analisada com maior cautela.

```
sexo <- compare_estimates(
  varname = "sexo",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  population_data = censo22,
  show_population = TRUE
)

cor_raca <- compare_estimates(
  varname = "cor_raca",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  population_data = censo22,
  show_population = TRUE
)

idade <- compare_estimates(
  varname = "idade",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  population_data = censo22,
  show_population = TRUE
)

(sexo | cor_raca) /
(idade) +
plot_layout(guides = "collect") & theme(legend.position = "bottom")
```



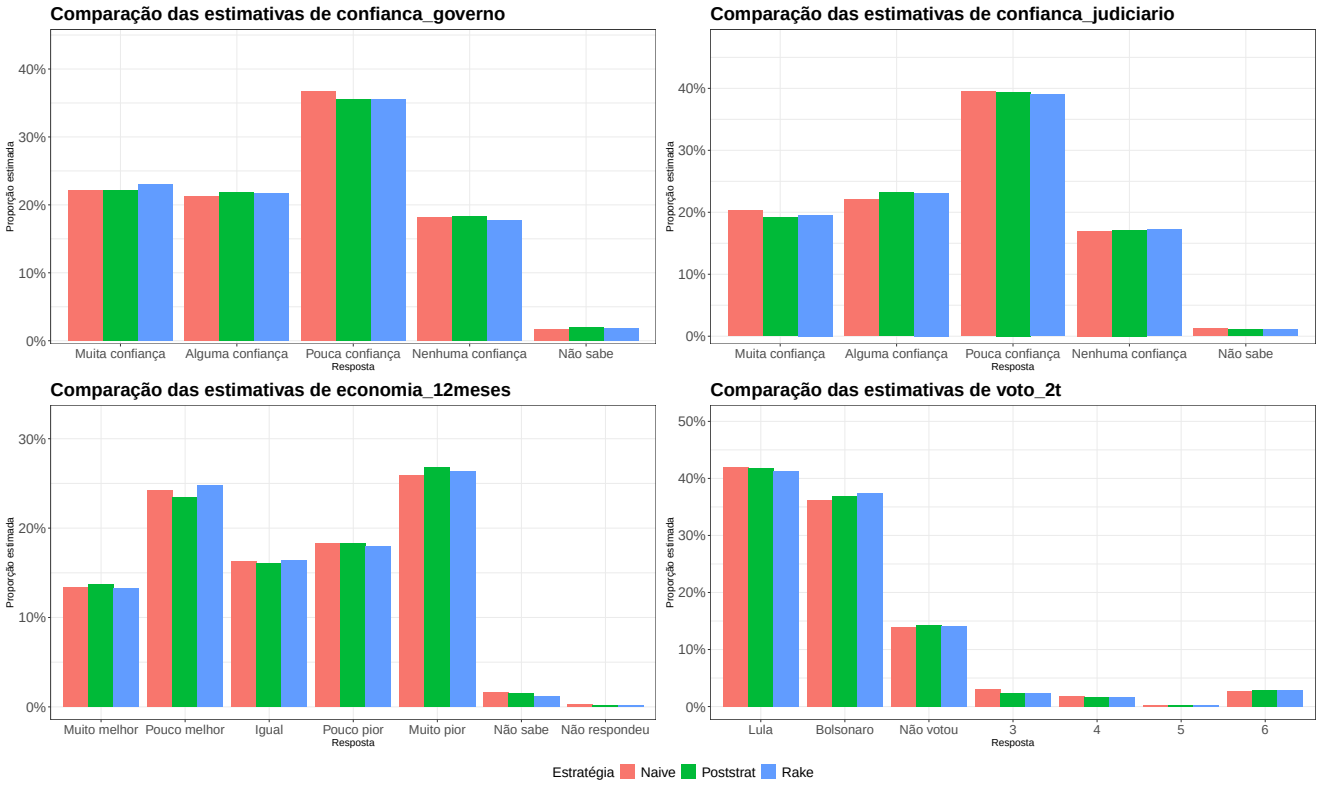
As duas abordagens foram capazes de aproximar muito bem o parâmetro populacional em praticamente todos os casos. No caso das disparidades de idade, por exemplo, conseguimos superar o problema da subestimação da população idosa e a superestimação da população entre 25 e 34 anos de idade. No caso da cor/raça, também corrigimos o problema da amostra.

Observe, agora, o caso do sexo: apesar de não termos incluído explicitamente essa variável na ponderação, ela também foi razoavelmente corrigida. Nesse caso, a pós-estratificação se saiu melhor que o rake. De fato, é um resultado razoável; afinal, a pós-estratificação utiliza a distribuição conjunta das variáveis de ponderação – isto é, suas interações – e, portanto, se existe alguma mínima correlação dessa distribuição conjunta com a variável de sexo, ela tende a ser indiretamente ajustada, pelo menos parcialmente.

Por fim, vejamos os resultados para as variáveis para as quais desconhecemos o parâmetro populacional:

```
confianca_governo <- compare_estimates(
  varname = "confianca_governo",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  ordered_levels = c("Muita confiança", "Alguma confiança", "Pouca confiança",
                     "Nenhuma confiança", "Não sabe")
)

confianca_judiciario <- compare_estimates(
  varname = "confianca_judiciario",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
```

```
  rake_design = rake_design,
  ordered_levels = c("Muita confiança", "Alguma confiança", "Pouca confiança",
                     "Nenhuma confiança", "Não sabe")
)

economia_12meses <- compare_estimates(
  varname = "economia_12meses",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  ordered_levels = c("Muito melhor", "Pouco melhor", "Igual",
                     "Pouco pior", "Muito pior",
                     "Não sabe", "Não respondeu")
)

voto_2t <- compare_estimates(
  varname = "voto_2t",
  raw_data = eseb22,
  poststrat_design = poststrat_design,
  rake_design = rake_design,
  ordered_levels = c("Lula", "Bolsonaro", "Não votou",
                     "3", "4", "5", "6")
)

(confianca_governo | confianca_judiciario) /
(economia_12meses | voto_2t) +
plot_layout(guides = "collect") & theme(legend.position = "bottom")
```



Aqui, os resultados são mais sutis, com ajustes na faixa de um ou alguns poucos pontos percentuais. No caso da confiança no governo, por exemplo, a estimativa *naive* superestima levemente os indivíduos com pouca confiança no governo, enquanto parece subestimar em pequeno grau os indivíduos com alguma ou muita confiança. Já no caso da avaliação da economia nos últimos 12 meses, por exemplo, houve uma subestimação daqueles que acreditam que ela estava "Muito pior".

Por fim, vale a pena destacar as estimativas de voto no segundo turno. A abordagem *naive* superestima ligeiramente a votação de Lula e subestima a de Bolsonaro em comparação com as abordagens ajustadas. É uma diferença pequena, mas, considerando o contexto altamente acirrado das eleições de 2022, bastante importante.

Seria interessante acrescentar a religião às variáveis de ajuste, considerando a literatura recente que associa o "voto evangélico" – termo ao qual faço referência sem a necessária cautela teórico-metodológica neste caso – a Bolsonaro. No que diz respeito às variáveis `confianca_governo`, `confianca_judiciario` e `economia_12meses`, seria razoável esperar que a inclusão de variáveis políticas à ponderação (por exemplo, a proporção de votos em Bolsonaro e Lula via rake) surtiria efeitos mais significativos[1].

Além disso, um ponto importante é que análises futuras poderiam testar utilizar a variável de região ao invés de unidade federativa, ou mesmo colapsar outras variáveis, com o objetivo de garantir que todos os estratos obtidos pelo censo apareçam na amostra. Essas abordagens potencialmente melhorariam a qualidade do ajuste por pós-estratificação.

[1] Isso só faria sentido, é claro, em um cenário onde as eleições já ocorreram e estamos explorando a opinião da população a respeito dessas outras variáveis.

Referências

Lumley, Thomas. 2011. *Complex Surveys: A Guide to Analysis Using R*. John Wiley & Sons.

Wolf, Christof, Dominique Joye, Tom W. Smith, e Yang-chih Fu. 2016. «Analysis of Data from Stratified and Clustered Surveys». Em *The SAGE Handbook of Survey Methodology*. Sage.