BISG Testing

Chris Iyer

2023-03-29

BISG Sum Methods Compare

Here, I wanted to test two different ways of using our BISG data. 1) "Categorical" (our initial default): Assign each voter entry a race based on the most probably category assigned by BISG. For example, compute Asian turnout by dividing the number of votes from people who are most likely Asian by the number of registered voters who are most likely Asian. 2) "Probabilistic" (potential best practice in the field): Keep each voter's *probabilities* for all races, and compute turnout and other stats using weighted sums of everyone. For example, compute Asian turnout by dividing the sum of the P(Asian) values for *all* collected votes and the sum of the P(Asian) values for all registered voters.

Our hypothesis here is that in a dataset so large, these two methods will not differ much. Effectively, the categorical method is rounding some voters up (e.g., from P(Asian)=0.7 to 1) and some voters down (e.g., from P(Asian = 0.1) to 0). I'd estimate that these effects cancel out in the aggregate, and we will get very similar metrics of turnout. However, since I've seen the probabilistic method used in reports by experts in the field, so I wanted to test this question.

Here, I take our 2020 TX voter history + voter registration files and calculate turnout in the 2012, 2016, and 2020 general elections, grouping by race with these two different BISG sum methods.

Whole-file race totals

First, take a look here at the totals of each race in the registered voter file, using the two different methods. If they really balance out, the totals should be roughly equal.

```
sums <- c(
  'aapi' = sum(df_voters$pred.asi),
  'black' = sum(df_voters$pred.bla),
  'latinx' = sum(df_voters$pred.his),
  'other' = sum(df_voters$pred.oth),
  'white' = sum(df_voters$pred.whi)
)

# COMPARE THESE
print('Sum of probabilities:')
```

```
## [1] "Sum of probabilities:"
```

```
sums
```

```
##       aapi      black     latinx      other      white
##   726822.3  2494623.4  5135368.3   828394.6 10155440.1
```

```
print('Sum of categorical race counts:')
```

```
## [1] "Sum of categorical race counts:"
```

```
t <- table(df_voters$race_pred)
t
```

```
##
##     aapi    black   latinx    other    white
##   712376  2387724  5133116   282136 10875756
```

```
print('Differences:')
```

```
## [1] "Differences:"
```

```
sums - t
```

```
##
##       aapi      black      latinx       other       white
##   14446.322  106899.441    2252.348  546258.643 -720315.896
```

We see that for some categories more than others, this is more or less true. For white voters, using categorical race coding results in a *higher* count of white voters, whereas using categorical race coding results in a *lower* count for all other groups. We're not sure which is closer to the ground truth, but this is important to know.

Turnout comparison

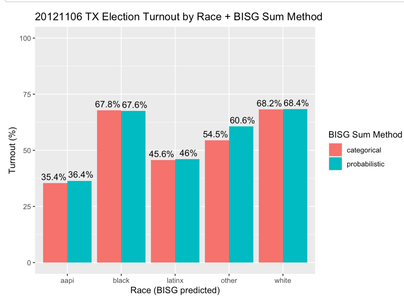
But, these differences are probably only very important if they yield differences in our summary statistics, e.g., turnout. So, here is a function that calculates voter turnout using the two different BISG sum methods, and then produces a comparison plot below.

```
compare_turnout <- function(date) {

  # METHOD 1: CATEGORICAL
  votes <- df_history %>%
    filter(election_date == date, election_voting_method %in% c('EV', 'ED', 'AB', 'PB')) %>%
    group_by(race_pred) %>% summarise(n=n())

  voters <- df_voters %>% filter(edr < date) %>% group_by(race_pred) %>% summarise(n=n())

  turnout <- votes %>%
    rename(votes = n) %>%
    mutate(voters = voters$n,
           turnout_cat = votes/voters*100,
           turnout_cat_pretty = paste0(round(turnout_cat,1), '%')) %>% select(race_pred, turnout_cat, turnout_cat
_pretty)

  # METHOD 2: PROBABILISTIC
  votes <- df_history %>% filter(election_date == date, election_voting_method %in% c('EV', 'ED', 'AB', 'PB'))
  voters <- df_voters %>% filter(edr < date)

  race_cols <- c(aapi='pred.asi', black='pred.bla', latinx='pred.his', other='pred.oth', white='pred.whi')
  turnout$turnout_prob <- rep(0, nrow(turnout))
  turnout$turnout_prob_pretty <- rep(0, nrow(turnout))
  for (i in 1:nrow(turnout)) {
    race <- turnout$race_pred[i]
    curr <- votes %>% select(race_cols[race]) %>% sum() / voters %>% select(race_cols[race]) %>% sum()*100
    turnout$turnout_prob[i] <- curr
    turnout$turnout_prob_pretty[i] <- paste0(round(curr,1), '%')
  }

  # COMPARE PLOT
  plot <- data.frame(race = rep(turnout$race_pred, 2),
                     turnout = c(turnout$turnout_cat, turnout$turnout_prob),
                     turnout_pretty = c(turnout$turnout_cat_pretty, turnout$turnout_prob_pretty),
                     sum_method = c(rep('categorical',5), rep('probabilistic',5))) %>%
      ggplot(aes(x = race,y=turnout, fill=sum_method)) +
      geom_bar(position="dodge", stat="identity")+
      geom_text(aes(label = turnout_pretty), vjust = -0.5, position = position_dodge(width = 1)) +
      ylim(0,100)+
      labs(x = 'Race (BISG predicted)', y = 'Turnout (%)', fill='BISG Sum Method',
           title= paste0(date, ' TX Election Turnout by Race + BISG Sum Method'))
  return(plot)
}
```

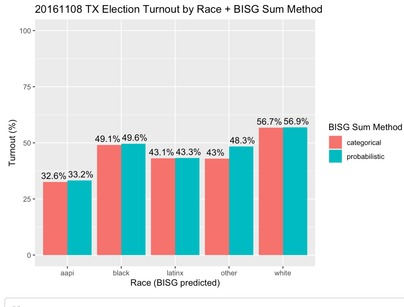
I'll execute this comparison for 3 general elections (2012, 2016, 2020).

```
lapply(c(20121106, 20161108, 20201103), compare_turnout)
```

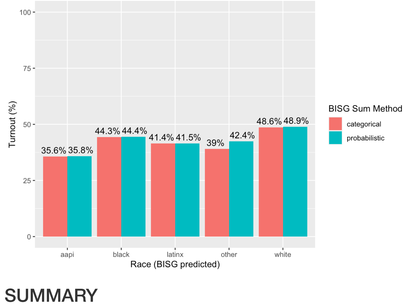
```
## [[1]]
```




```
##
## [[2]]
```




```
##
## [[3]]
```



SUMMARY

We can see in the plots above that our estimates of turnout vary only *very* slightly. If we're interested in qualitative patterns of which groups have low or high turnout, this method is not going to make a difference. However, because I have seen Fraga, the Brennan Center, and others use the probabilistic summing method, I will go ahead and use that for the future. But, this analysis has shown us that it won't make much of a difference which method we use.