Analysis on the popular vote of the 2020 American federal election

Yena Joo, Woolim Kim, Guemin Kim

Nov 2, 2020

Title of your Report

Name(s) of Author(s)

Date

Model

Here we are interested in predicting the popular vote outcome of the 2020 American federal election (include citation). To do this we are employing a post-stratification technique. In the following sub-sections I will describe the model specifics and the post-stratification calculation.

Model Specifics

I will (incorrectly) be using a linear regression model to model the proportion of voters who will vote for Donald Trump. This is a naive model. I will only be using age, which is recorded as a numeric variable, to model the probability of voting for Donald Trump. The simple linear regression model I am using is:

$$y = \beta_0 + \beta_1 x_{age} + \epsilon$$

Where y represents the proportion of voters who will vote for Donald Trump. Similarly, β_0 represents the intercept of the model, and is the probability of voting for Donald Trump at age 0. Additionally, β_1 represents the slope of the model. So, for everyone one unit increase in age, we expect a β_1 increase in the probability of voting for Donald Trump.

Model specifics (woolim)

We will be using the logistic regression model and post-stratification to predict the proportion of voters who will vote for Donald Trump and Joe Biden. Using 6 different variables(age_group, gender, race, education, household_income , and state) to model the probability of voting for Trump and Biden. Since the vote intention variable is binary(either 'vote for' or 'not vote'), the logistic regression model is a suitable model to be used. The logistic regression model we are using is:

$$log(p_i/1 - p_i) = \beta_0 + \beta_1 x_{age\ group} + \beta_2 x_{gender} + \beta_3 x_{race} + \beta_4 x_{education} + \beta_5 x_{household\ income} + \beta_6 x_{state}$$

Explanation… where $(p_i/1 - p_i)$ represents the ratio of two odds, where p_i is the probability of voters who will vote for Donald trump or Joe Biden, $1 - p_i$ is the probability of not voting.Then we use the log function to find the proportion of voters who will vote for Trump or Biden. Similarly, β_0, β_1, ..., β_6 is our parameters of interest, the probability of voting for every one unit of age_group, gender, race, education, household income, and state.

```r
#Yena
# Creating the Model
#model <- glmer(vote_trump ~ age_group + sex + race + educ + household_income)

#vote for Trump
#model_t <- glmer(vote_trump ~ age_group + gender + race + education + (1|household_income),
            #data=survey_data, family = binomial)
model2_t <- glm(vote_trump ~ as.factor(age_group) + as.factor(gender) + as.factor(race) + as.factor(educ

#voting for Biden
#model_b <- glmer(vote_Biden ~ age_group + gender + race + education + household_income + (1|state),
            #data=survey_data, family = binomial)
model2_b <- glm(vote_Biden ~ as.factor(age_group) + as.factor(gender) + as.factor(race) + as.factor(educ
            data=survey_data, family="binomial")
# Model Results (to Report in Results section)
#summary(model_t)
#summary(model_b)

# OR
broom::tidy(model2_t)
```

```
## # A tibble: 70 x 5
##    term                                      estimate std.error statistic  p.value
##    <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                                -0.713     0.740     -0.964 3.35e- 1
##  2 as.factor(age_group)30-44 year olds         0.589     0.0926     6.37  1.94e-10
##  3 as.factor(age_group)45-64 year olds         0.732     0.0919     7.97  1.57e-15
##  4 as.factor(age_group)65 years and older      0.813     0.106      7.66  1.87e-14
##  5 as.factor(gender)Male                       0.400     0.0599     6.68  2.42e-11
##  6 as.factor(race)Black                       -1.39      0.202     -6.86  6.68e-12
##  7 as.factor(race)Native                       0.475     0.270      1.76  7.89e- 2
##  8 as.factor(race)Other                       -0.0760    0.191     -0.397 6.91e- 1
##  9 as.factor(race)White                        0.613     0.153      4.01  5.99e- 5
## 10 as.factor(education)Didn't graduate fr~     0.338     0.115      2.95  3.21e- 3
## # ... with 60 more rows
```

```r
broom::tidy(model2_b)
```

```
## # A tibble: 70 x 5
##    term                                      estimate std.error statistic  p.value
##    <chr>                                        <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                                -0.479     0.839     -0.571 5.68e- 1
##  2 as.factor(age_group)30-44 year olds        -0.213     0.0830    -2.56  1.03e- 2
##  3 as.factor(age_group)45-64 year olds        -0.274     0.0832    -3.29  1.01e- 3
##  4 as.factor(age_group)65 years and older     -0.107     0.0988    -1.08  2.78e- 1
##  5 as.factor(gender)Male                      -0.301     0.0578    -5.21  1.85e- 7
##  6 as.factor(race)Black                        1.00      0.157      6.39  1.68e-10
##  7 as.factor(race)Native                      -0.406     0.264     -1.54  1.24e- 1
##  8 as.factor(race)Other                        0.0558    0.165      0.338 7.35e- 1
```

```
##  9 as.factor(race)White                      -0.359     0.134     -2.67  7.63e- 3
## 10 as.factor(education)Didn't graduate fr~  -0.682     0.110     -6.20  5.60e-10
## # ... with 60 more rows
```

```r
###yps for glm trump
census_data$logodds_estimate_t <-
  model2_t %>%
  predict(newdata = census_data)

census_data$estimate2_t <-
  exp(census_data$logodds_estimate_t)/(1+exp(census_data$logodds_estimate_t))
#using group_by(income)
predict_t <-
census_data %>%
  filter(!is.na(estimate2_t))%>%
  mutate(vote_prop2 = estimate2_t*count) %>%
  group_by(household_income)%>%
  summarise(alp_predict = sum(vote_prop2)/sum(count))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
predict_t
```

```
## # A tibble: 9 x 2
##   household_income     alp_predict
##   <chr>                      <dbl>
## 1 $100,000 to $149,999       0.476
## 2 $15,000 to $24,999         0.380
## 3 $150,000 and over          0.495
## 4 $25,000 to $34,999         0.383
## 5 $35,000 to $44,999         0.390
## 6 $45,000 to $54,999         0.430
## 7 $55,000 to $74,999         0.418
## 8 $75,000 to $99,999         0.412
## 9 Less than $14,999          0.318
```

```r
#individual
predict_t_nogroup <-
census_data %>%
  filter(!is.na(estimate2_t))%>%
  mutate(vote_prop2 = estimate2_t*count) %>%
  summarise(alp_predict_2t = sum(vote_prop2)/sum(count))
predict_t_nogroup
```

```
## # A tibble: 1 x 1
##   alp_predict_2t
##            <dbl>
## 1          0.428
```

```r
###yps for glm Biden
census_data$logodds_b <-
  model2_b %>%
```

```r
  predict(newdata = census_data)

census_data$estimate2_b <-
  exp(census_data$logodds_b)/(1+exp(census_data$logodds_b))

#using group_by()
predict_b<-
census_data %>%
  filter(!is.na(estimate2_b))%>%
  mutate(vote_prop_b = estimate2_b*count) %>%
  group_by(household_income)%>%
  summarise(alp_predict_b = sum(vote_prop_b)/sum(count))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
predict_b
```

```
## # A tibble: 9 x 2
##   household_income     alp_predict_b
##   <chr>                        <dbl>
## 1 $100,000 to $149,999         0.362
## 2 $15,000 to $24,999           0.395
## 3 $150,000 and over            0.370
## 4 $25,000 to $34,999           0.385
## 5 $35,000 to $44,999           0.409
## 6 $45,000 to $54,999           0.382
## 7 $55,000 to $74,999           0.413
## 8 $75,000 to $99,999           0.420
## 9 Less than $14,999            0.385
```

```r
#without grouping
predict_b_nogroup <-
census_data %>%
  filter(!is.na(estimate2_b))%>%
  mutate(vote_prop_b = estimate2_b*count) %>%
  summarise(alp_predict_2b = sum(vote_prop_b)/sum(count))
predict_b_nogroup
```

```
## # A tibble: 1 x 1
##   alp_predict_2b
##            <dbl>
## 1          0.388
```

```r
summary(census_data)
```

```
##   age_group            gender              race               state
##  Length:55325       Length:55325       Length:55325       Length:55325
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##    education          household_income       count           cell_prop
##   Length:55325       Length:55325       Min.   :   1.00   Min.   :3.849e-07
##   Class :character   Class :character   1st Qu.:   2.00   1st Qu.:7.698e-07
##   Mode  :character   Mode  :character   Median :   8.00   Median :3.079e-06
##                                         Mean   :  46.96   Mean   :1.807e-05
##                                         3rd Qu.:  33.00   3rd Qu.:1.270e-05
##                                         Max.   :6305.00   Max.   :2.427e-03
##
##   logodds_estimate_t  estimate2_t      logodds_b        estimate2_b
##   Min.   :-5.2391    Min.   :0.0053   Min.   :-2.7893   Min.   :0.0579
##   1st Qu.:-1.5250    1st Qu.:0.1787   1st Qu.:-0.7782   1st Qu.:0.3147
##   Median :-0.7611    Median :0.3184   Median :-0.2836   Median :0.4296
##   Mean   :-0.8891    Mean   :0.3248   Mean   :-0.2315   Mean   :0.4475
##   3rd Qu.:-0.1863    3rd Qu.:0.4536   3rd Qu.: 0.2732   3rd Qu.:0.5679
##   Max.   : 1.7106    Max.   :0.8469   Max.   : 3.1671   Max.   :0.9596
##   NA's   :613        NA's   :613      NA's   :613       NA's   :613
```

Post-Stratification

In order to estimate the proportion of voters who will vote for Donald Trump I need to perform a post-stratification analysis. Here I create cells based off different ages. Using the model described in the previous sub-section I will estimate the proportion of voters in each age bin. I will then weight each proportion estimate (within each bin) by the respective population size of that bin and sum those values and divide that by the entire population size.

```
# exclude na observations for vote_trump and vote_Biden from survey_data
survey_data <-
  survey_data %>%
  filter(!is.na(vote_trump), !is.na(vote_Biden))

# Trump model
model_t <- glm(vote_trump ~ as.factor(age_group) + as.factor(gender) + as.factor(race) + as.factor(educa
                data=survey_data, family="binomial")

# Biden model
model_b <- glm(vote_Biden ~ as.factor(age_group) + as.factor(gender) + as.factor(race) + as.factor(educa
               data=survey_data, family="binomial")

#broom::tidy(model_t)
#broom::tidy(model_b)

#yps for glm trump
census_data$estimate_T <-
  model_t %>%
  predict(newdata = census_data)

ypsTrump<-
census_data %>%
  filter(!is.na(estimate_T))%>%
  mutate(vote_prop_t = estimate_T*cell_prop) %>%
  mutate(vote_prop_T = estimate_T*count) %>%
```

```r
  summarise(estimate_vote_t = sum(vote_prop_t), predict_vote_T = sum(vote_prop_T)/sum(count))

prob_Trump <- exp(ypsTrump$predict_vote_T)/(1+exp(ypsTrump$predict_vote_T))

#yps for glm Biden

census_data$estimate_B <-
  model_b %>%
  predict(newdata = census_data)

ypsBiden<-
census_data %>%
  filter(!is.na(estimate_B))%>%
  mutate(vote_prop_b = estimate_B*cell_prop) %>%
  mutate(vote_prop_B = estimate_B*count) %>%
  summarise(estimate_vote_b = sum(vote_prop_b), predict_vote_B = sum(vote_prop_B)/sum(count))

prob_Biden <- exp(ypsBiden$predict_vote_B)/(1+exp(ypsBiden$predict_vote_B))

prob_Trump
```

```
## [1] 0.4104034
```

```r
prob_Biden
```

```
## [1] 0.3805689
```

Results

Here you will include all results. This includes descriptive statistics, graphs, figures, tables, and model results. Please ensure that everything is well formatted and in a report style. You must also provide an explanation of the results in this section.

Please ensure that everything is well labelled. So if you have multiple histograms and plots, calling them Figure 1, 2, 3, etc. and referencing them as Figure 1, Figure 2, etc. in your report will be expected. The reader should not get lost in a sea of information. Make sure to have the results be clean, well formatted and digestible.

```r
predict_t_nogroup
```

```
## # A tibble: 1 x 1
##   alp_predict_2t
##            <dbl>
## 1          0.428
```

```r
predict_b_nogroup
```

```
## # A tibble: 1 x 1
##   alp_predict_2b
##            <dbl>
## 1          0.388
```

```
prob_Trump
```

```
## [1] 0.4104034
```

```
prob_Biden
```

```
## [1] 0.3805689
```

Discussion

Here you will summarize the previous sections and discuss conclusions drawn from the results. Make sure to elaborate and connect your analysis to the goal of the study.

Weaknesses

Here we discuss weaknesses of the study, data, analysis, etc. You can also discuss areas for improvement.

Next Steps

Here you discuss subsequent work to be done after this report. This can include next steps in terms of statistical analysis (perhaps there is a more efficient algorithm available, or perhaps there is a caveat in the data that would allow for some new technique). Future steps should also be specified in terms of the study setting (eg. including a follow-up survey on something, or a subsequent study that would complement the conclusions of your report).

References