Prediction of 2020 USA ELECTION

Yehao Zheng, Kaicheng Huang, Yujing Hua, Tingyi Li

Nov 2rd

Model

Prepration to create model

Create the model

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## vote_2020 ~ (1 + race | cell) + agegroup + sex + state + household_income
##    Data: survey_data
##
##      AIC      BIC   logLik deviance df.resid
##   5629.2   6305.7  -2708.6   5417.2     4263
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5882 -0.8692  0.2642  0.9248  2.2995
##
```

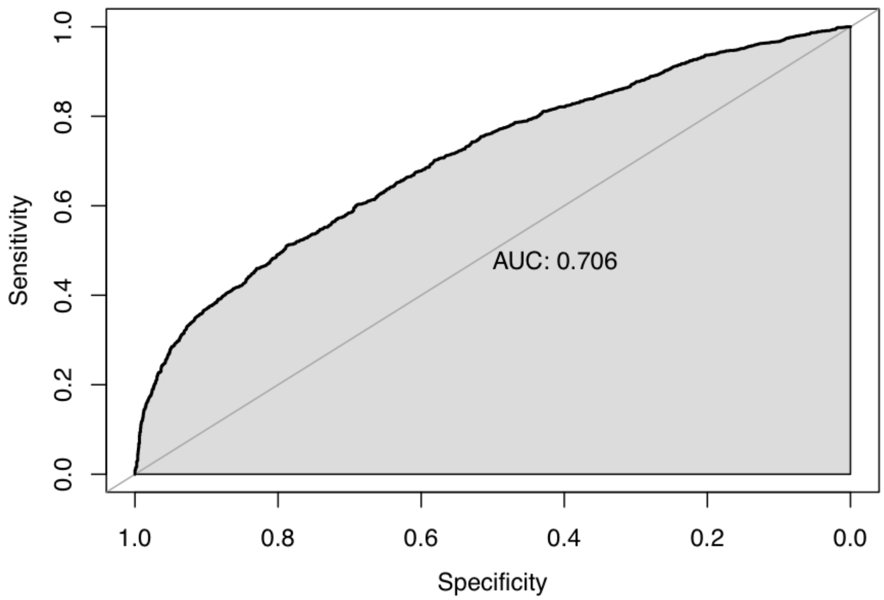
```
## Random effects:
##  Groups Name                               Variance Std.Dev. Corr
##  cell   (Intercept)                        0.5856   0.7652
##         raceBlack, or African American     1.6179   1.2720    0.09
##         raceChinese                        0.2159   0.4647   -0.83  0.48
##         raceJapanese                       4.5094   2.1235    0.06  0.31
##         raceOther Asians and Pacific Islander 0.5855   0.7652   -1.00 -0.09
##         raceOther race                     0.5856   0.7652   -1.00 -0.09
##         raceWhite                          0.3346   0.5785   -0.50 -0.26
##
##
##
##
##   0.13
##   0.83 -0.06
##   0.83 -0.06  1.00
##   0.30 -0.36  0.50  0.50
## Number of obs: 4369, groups:  cell, 14
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -0.82573    1.14302  -0.722 0.470043
## agegroup40 to 60                 -0.34626    0.07833  -4.421 9.84e-06 ***
## agegroup60 to 80                 -0.22109    0.08456  -2.615 0.008934 **
## agegroupabove 80                 -0.50063    0.36101  -1.387 0.165523
## sexMale                          -0.74001    0.21331  -3.469 0.000522 ***
## stateAL                           1.14944    1.16538   0.986 0.323976
## stateAR                           0.84710    1.21335   0.698 0.485084
## stateAZ                           1.34101    1.15082   1.165 0.243912
## stateCA                           2.02343    1.13831   1.778 0.075473 .
## stateCO                           1.78438    1.16358   1.534 0.125145
## stateCT                           2.48935    1.17794   2.113 0.034574 *
## stateDC                           2.01044    1.25000   1.608 0.107758
## stateDE                           2.41194    1.24543   1.937 0.052791 .
## stateFL                           1.52545    1.13937   1.339 0.180618
## stateGA                           1.17197    1.15096   1.018 0.308555
## stateHI                           1.78063    1.24081   1.435 0.151272
## stateIA                           1.77554    1.18512   1.498 0.134082
## stateID                           0.87982    1.23026   0.715 0.474516
## stateIL                           1.84403    1.14382   1.612 0.106926
## stateIN                           1.60526    1.15962   1.384 0.166267
## stateKS                           1.07583    1.19470   0.900 0.367856
## stateKY                           1.73606    1.16239   1.494 0.135300
## stateLA                           1.57892    1.16895   1.351 0.176788
## stateMA                           2.41752    1.16189   2.081 0.037464 *
## stateMD                           1.82410    1.16422   1.567 0.117160
## stateME                           2.06227    1.24333   1.659 0.097184 .
## stateMI                           1.87577    1.14802   1.634 0.102277
## stateMN                           1.75808    1.16837   1.505 0.132392
## stateMO                           1.63417    1.15682   1.413 0.157763
## stateMS                           0.94744    1.20540   0.786 0.431868
## stateMT                           1.58919    1.26708   1.254 0.209762
## stateNC                           1.49585    1.14753   1.304 0.192393
## stateND                         -12.15040  678.60863  -0.018 0.985715
```

```
## stateNE                              2.07330    1.25816   1.648 0.099376 .
## stateNH                              1.82551    1.24898   1.462 0.143851
## stateNJ                              1.69101    1.14789   1.473 0.140711
## stateNM                              2.49729    1.26240   1.978 0.047905 *
## stateNV                              1.24506    1.17711   1.058 0.290180
## stateNY                              1.81730    1.13959   1.595 0.110780
## stateOH                              1.67989    1.14499   1.467 0.142332
## stateOK                              1.27143    1.18363   1.074 0.282743
## stateOR                              1.96135    1.16285   1.687 0.091666 .
## statePA                              1.45990    1.14417   1.276 0.201975
## stateRI                              2.47270    1.43812   1.719 0.085542 .
## stateSC                              0.89457    1.16763   0.766 0.443590
## stateSD                              1.31513    1.27362   1.033 0.301794
## stateTN                              0.87621    1.16581   0.752 0.452298
## stateTX                              1.16255    1.14080   1.019 0.308172
## stateUT                              1.22689    1.20661   1.017 0.309244
## stateVA                              1.85586    1.14892   1.615 0.106243
## stateVT                              3.79235    1.55535   2.438 0.014758 *
## stateWA                              2.04585    1.15514   1.771 0.076546 .
## stateWI                              2.09722    1.15906   1.809 0.070387 .
## stateWV                              1.05745    1.20683   0.876 0.380907
## stateWY                              1.98284    1.82558   1.086 0.277415
## household_income$125,000 to $149,999  0.10638    0.17142   0.621 0.534870
## household_income$15,000 to $19,999    0.54206    0.19086   2.840 0.004511 **
## household_income$150,000 to $174,999  0.24562    0.20222   1.215 0.224496
## household_income$175,000 to $199,999 -0.26663    0.24337  -1.096 0.273276
## household_income$20,000 to $24,999    0.29310    0.18440   1.590 0.111940
## household_income$200,000 to $249,999 -0.64041    0.22782  -2.811 0.004938 **
## household_income$25,000 to $29,999    0.24909    0.18253   1.365 0.172353
## household_income$250,000 and above   -0.15075    0.23270  -0.648 0.517105
## household_income$30,000 to $34,999    0.29630    0.18169   1.631 0.102938
## household_income$35,000 to $39,999    0.31522    0.18862   1.671 0.094683 .
## household_income$40,000 to $44,999    0.46467    0.19538   2.378 0.017390 *
## household_income$45,000 to $49,999    0.17960    0.18676   0.962 0.336217
## household_income$50,000 to $54,999    0.22436    0.17666   1.270 0.204078
## household_income$55,000 to $59,999    0.13065    0.22489   0.581 0.561281
## household_income$60,000 to $64,999    0.30123    0.21711   1.387 0.165296
## household_income$65,000 to $69,999    0.25142    0.23499   1.070 0.284663
## household_income$70,000 to $74,999    0.47029    0.20750   2.266 0.023425 *
## household_income$75,000 to $79,999    0.09047    0.21042   0.430 0.667235
## household_income$80,000 to $84,999    0.47206    0.25180   1.875 0.060831 .
## household_income$85,000 to $89,999    0.47700    0.26268   1.816 0.069379 .
## household_income$90,000 to $94,999    0.31102    0.29192   1.065 0.286681
## household_income$95,000 to $99,999    0.60594    0.22325   2.714 0.006645 **
## household_incomeLess than $14,999     0.39800    0.15393   2.586 0.009722 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 78 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## convergence code: 0
## unable to evaluate scaled gradient
```

```
## Model failed to converge: degenerate  Hessian with 4 negative eigenvalues
## failure to converge in 10000 evaluations

## $table
##               Reference
## Prediction     Donald Trump Joe Biden
##   Donald Trump         1399       888
##   Joe Biden             684      1398

##  Accuracy
## 0.6401923

## Setting levels: control = Donald Trump, case = Joe Biden

## Setting direction: controls < cases

## Area under the curve: 0.7063
```



In this project, we choose to build a generalized linear Mixed-Effects Models with one random effect factor 'Vote_2020' indicating who will get more vote in the election by our prediction and four fixed effect factors 'agegroup','sex','state','household_income', which we will consider and use in our model to predict who will win. Our mathematical formula is vote_2020~ agegroup+sex+state+household_income+(1+race|cell) . From the output above, we see the coefficient of intercept beta_0 is -0.82573 which is the log odd of vote_2020 when we set agegroup, sex, state,household_income to be their reference value.

We choose dunction glmer() based on several reasons. Firstly, as mentioned before we have both random and fixed effect factors in the model and the data we choose is clustered by support Trump or Biden. Second, we do not expect a linear relationship between response and explanatory variables since we want to have a binomial outcome. Also, from the output we got a value of auc=0.7063 which means our model has a good performance since auc is larger than 0.5, this is also showed by the figure of AUC (figure 1) such that our predict model is close to true model with more than 50% of accuracy

Model Selection

```
## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues

## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from :
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix co
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## vote_2020 ~ (1 + agegroup | cell) + race + sex + state + household_income
##    Data: survey_data
##
##      AIC      BIC   logLik deviance df.resid
##   5577.1   6157.9  -2697.5   5395.1     4278
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7985 -0.8686  0.2782  0.9255  2.3259
##
## Random effects:
##  Groups Name             Variance Std.Dev. Corr
##  cell   (Intercept)      0.03517  0.1875
##         agegroup40 to 60 0.13106  0.3620   -1.00
##         agegroup60 to 80 0.10354  0.3218   -0.74  0.78
##         agegroupabove 80 0.30282  0.5503   -0.65  0.60 -0.03
## Number of obs: 4369, groups:  cell, 14
##
## Fixed effects:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -2.00010    1.18310  -1.691  0.09092 .
## raceBlack, or African American         2.37367    0.35142   6.755 1.43e-11 ***
## raceChinese                            1.46934    0.50017   2.938  0.00331 **
## raceJapanese                           1.28790    0.70503   1.827  0.06774 .
## raceOther Asians and Pacific Islander  0.79610    0.37466   2.125  0.03360 *
## raceOther race                         0.87722    0.34863   2.516  0.01186 *
## raceWhite                              0.12215    0.31999   0.382  0.70265
## sexMale                               -0.54174    0.07660  -7.072 1.52e-12 ***
## stateAL                                1.21877    1.16711   1.044  0.29636
## stateAR                                0.94296    1.21576   0.776  0.43798
## stateAZ                                1.47531    1.15355   1.279  0.20092
## stateCA                                2.11288    1.14065   1.852  0.06398 .
## stateCO                                1.88036    1.16597   1.613  0.10681
## stateCT                                2.58460    1.18004   2.190  0.02851 *
## stateDC                                2.16294    1.25311   1.726  0.08434 .
## stateDE                                2.49690    1.24800   2.001  0.04542 *
```

```
## stateFL                                  1.62925    1.14172   1.427  0.15357
## stateGA                                  1.28899    1.15354   1.117  0.26382
## stateHI                                  1.88337    1.24563   1.512  0.13054
## stateIA                                  1.88999    1.18772   1.591  0.11155
## stateID                                  0.98011    1.23222   0.795  0.42638
## stateIL                                  1.95490    1.14633   1.705  0.08813 .
## stateIN                                  1.70510    1.16213   1.467  0.14232
## stateKS                                  1.17055    1.19834   0.977  0.32866
## stateKY                                  1.84165    1.16501   1.581  0.11393
## stateLA                                  1.69748    1.17155   1.449  0.14736
## stateMA                                  2.53409    1.16470   2.176  0.02957 *
## stateMD                                  1.93276    1.16689   1.656  0.09765 .
## stateME                                  2.20355    1.24634   1.768  0.07706 .
## stateMI                                  1.95900    1.15036   1.703  0.08858 .
## stateMN                                  1.86905    1.17094   1.596  0.11045
## stateMO                                  1.73476    1.15953   1.496  0.13463
## stateMS                                  1.05497    1.20817   0.873  0.38256
## stateMT                                  1.70796    1.26925   1.346  0.17842
## stateNC                                  1.59960    1.14996   1.391  0.16422
## stateND                                -11.15089  440.91995  -0.025  0.97982
## stateNE                                  2.16485    1.26107   1.717  0.08604 .
## stateNH                                  1.92060    1.25179   1.534  0.12496
## stateNJ                                  1.79005    1.15048   1.556  0.11973
## stateNM                                  2.58535    1.26503   2.044  0.04098 *
## stateNV                                  1.34528    1.17969   1.140  0.25413
## stateNY                                  1.93401    1.14213   1.693  0.09039 .
## stateOH                                  1.77556    1.14742   1.547  0.12176
## stateOK                                  1.35904    1.18625   1.146  0.25193
## stateOR                                  2.06816    1.16521   1.775  0.07591 .
## statePA                                  1.56282    1.14665   1.363  0.17290
## stateRI                                  2.60114    1.44200   1.804  0.07126 .
## stateSC                                  1.01489    1.16977   0.868  0.38561
## stateSD                                  1.41200    1.27631   1.106  0.26859
## stateTN                                  0.99999    1.16819   0.856  0.39199
## stateTX                                  1.27766    1.14317   1.118  0.26372
## stateUT                                  1.36726    1.20917   1.131  0.25817
## stateVA                                  1.96529    1.15134   1.707  0.08783 .
## stateVT                                  3.99097    1.55394   2.568  0.01022 *
## stateWA                                  2.15777    1.15765   1.864  0.06233 .
## stateWI                                  2.19583    1.16136   1.891  0.05866 .
## stateWV                                  1.14505    1.20927   0.947  0.34369
## stateWY                                  2.17820    1.82750   1.192  0.23330
## household_income$125,000 to $149,999    0.09656    0.17204   0.561  0.57463
## household_income$15,000 to $19,999      0.53380    0.19114   2.793  0.00523 **
## household_income$150,000 to $174,999    0.23541    0.20300   1.160  0.24620
## household_income$175,000 to $199,999   -0.26199    0.24340  -1.076  0.28176
## household_income$20,000 to $24,999      0.29168    0.18490   1.578  0.11468
## household_income$200,000 to $249,999   -0.62698    0.22867  -2.742  0.00611 **
## household_income$25,000 to $29,999      0.25026    0.18250   1.371  0.17027
## household_income$250,000 and above     -0.14381    0.23294  -0.617  0.53699
## household_income$30,000 to $34,999      0.28048    0.18190   1.542  0.12309
## household_income$35,000 to $39,999      0.29964    0.18886   1.587  0.11261
## household_income$40,000 to $44,999      0.45528    0.19577   2.326  0.02004 *
## household_income$45,000 to $49,999      0.16300    0.18736   0.870  0.38430
```

```
## household_income$50,000 to $54,999      0.19550    0.17711   1.104  0.26965
## household_income$55,000 to $59,999      0.11920    0.22516   0.529  0.59655
## household_income$60,000 to $64,999      0.29562    0.21760   1.359  0.17430
## household_income$65,000 to $69,999      0.23544    0.23524   1.001  0.31690
## household_income$70,000 to $74,999      0.46765    0.20773   2.251  0.02437 *
## household_income$75,000 to $79,999      0.08967    0.21038   0.426  0.66995
## household_income$80,000 to $84,999      0.47477    0.25213   1.883  0.05970 .
## household_income$85,000 to $89,999      0.45799    0.26292   1.742  0.08152 .
## household_income$90,000 to $94,999      0.30524    0.29236   1.044  0.29647
## household_income$95,000 to $99,999      0.58803    0.22372   2.628  0.00858 **
## household_incomeLess than $14,999       0.39236    0.15421   2.544  0.01095 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 81 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
## failure to converge in 10000 evaluations
```

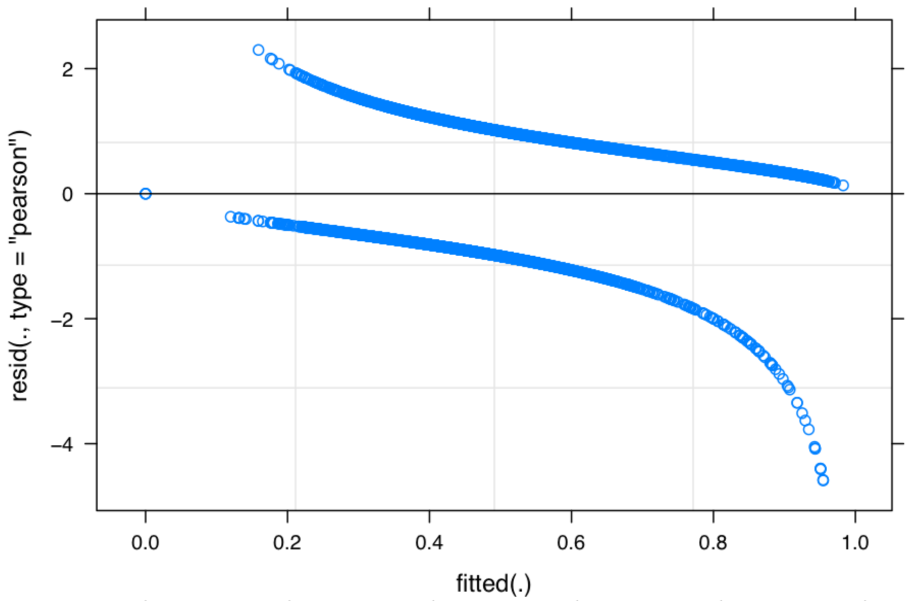
Model Checking

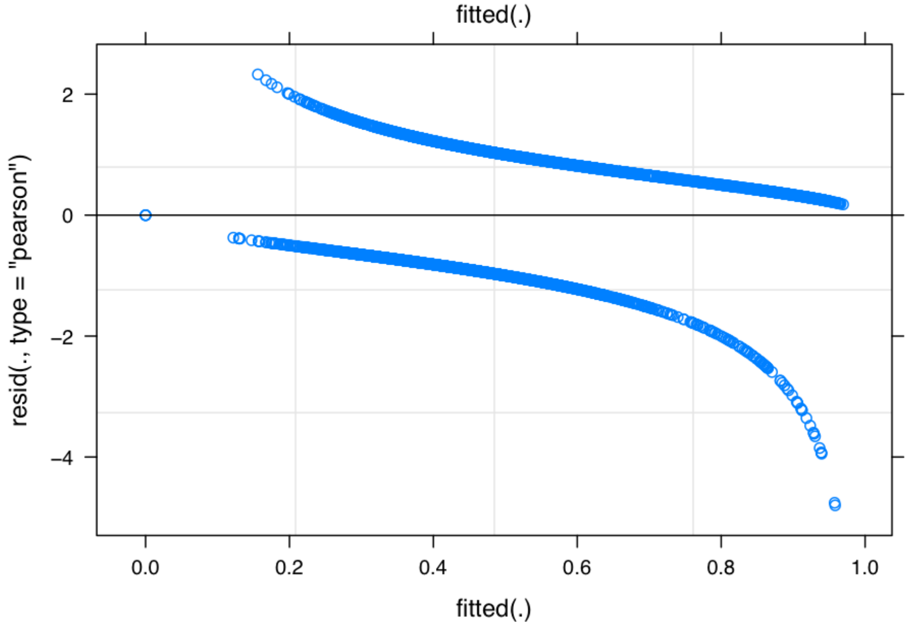
```
## [1] 5629.2
```

```
## [1] 5577.099
```

```
## [1] 6305.722
```

```
## [1] 6157.887
```





Post-Stratification

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

```
## # A tibble: 2 x 2
##   winner       total_votes
##   <chr>              <dbl>
## 1 Donald Trump         216
## 2 Joe Biden            322
```

Post-stratification is a sampling tool based on that similar units in a population should be treated equally. It is a statistical fact that grouping similar units in sampling usually reduces the variance of the resulting survey estimate. Stratification can be done during or after the study. The latter application often termed as post-stratification. We use glmer to estimate y for each cell. Then we can use demographics to "extrapolate" how the entire population will vote. It is useful because when the sample has been selected by the simple random sampling, post-stratification is often appropriate when a simple random is not represented by a balance. It can be performed with any useful model. In this case, we can use it to predict how people will vote. We create a cell in the survey data. The variable we used to create the cell is race. In addition, we use constant 1 as an interpreted variable combined with variable race to create this cell. The reason why we choose this variable is we think this variable is the most important effect related to the result of the vote. We didn't choose other variables because they have less influences related to the result of the vote.

For the addition analysis of Psot-Stratification, firstly, we use the predict function to calculate the probability of voting in census data and denote as result_2. Then we combine result_2 and census data denote and result of census. In the second step, we get the actual votes based on person weight and calculate the votes in each state by ifelse and summarise functions. At the end, we use mutate function to get the electoral votes in each state.

Results

```
## # A tibble: 51 x 3
## # Groups:   state [51]
##    state total_vote_Trump total_vote_Biden
##    <chr>            <dbl>            <dbl>
##  1 WI                  32               45
##  2 VA                  61               81
##  3 TX                 161              131
##  4 WA                  36               53
##  5 MA                  25               52
##  6 CA                 185              283
##  7 NC                  68               75
##  8 MD                  27               44
##  9 FL                 176              177
## 10 WV                  18                9
## # ... with 41 more rows
```

```
## # A tibble: 2 x 3
## # Groups:   sex [2]
##   sex    total_vote_Trump total_vote_Biden
##   <chr>             <dbl>            <dbl>
## 1 Female              848             1254
## 2 Male               1235             1032
```

```
## # A tibble: 4 x 3
## # Groups:   agegroup [4]
##   agegroup   total_vote_Trump total_vote_Biden
##   <chr>                 <dbl>            <dbl>
## 1 40 to 60                744              683
## 2 60 to 80                585              568
## 3 40 or less              733             1021
```

```
## 4 above 80                21                14
```

```
## # A tibble: 7 x 3
## # Groups:   race [7]
##   race                                total_vote_Trump total_vote_Biden
##   <chr>                                          <dbl>            <dbl>
## 1 White                                           1865             1536
## 2 Black, or African American                        55              431
## 3 Other Asians and Pacific Islander                 46               84
## 4 Chinese                                            9               35
## 5 Japanese                                           4               11
## 6 Other race                                        80              169
## 7 American Indian or Alaska Native                  24               20
```

We estimate that the proportion of voters in favour of voting for 'Joe Biden' to be 0.599. This is based on our post-stratification analysis of the proportion of voters in favour of 'Joe Biden' modelled by a glmer model, which accounted for 'agegroup','sex','state','household_income' and 'race'. Base on our further analysis above, in terms of sex, we can see male are more inventive to vote for Trump and Biden, and differences between male and female are not huge in both cases. Also form next calculation, we have almost same number people aged from 18-80 vote for Trump while Biden gains more vote from age group of 40 or less. Another result we have is that there are three states CA, TX, and FL,which both of Trump and Biden get most of their votes from. Based on races, we noticed that most of people who vote are white apparently, and people from other races are not prefer to vote for Trump but Biden, since Biden wins more vote from Black and other races.

Discussion

Firstly, we cleaned raw data with five variables we used and made some assumptions to the final data. We assumed people who intended to vote in the election must be older than 18 years old and we ignored observations with NA value. Also we tried to match each variable in survey data and census data have the same contents such that for the 'state' variable they have the same names of states. Secondly, we build a glmer model to predict the result of votes by using selected variables. For our model, we summarize it and display the model on survey data. Then we use model selection to create a new logit model with different cells and compare its AIC, BIC and diagnostic plot with our original model. Also, we found that some of our calculations contradict our knowledge in the real life. For example, we think Trump are more popular in the group of females rather than male but out statistics shows female vote more for Biden. However, there still are some data that follow what we heard and read from public medias. Our result shows Trump gains more vote from elder people while Biden is more popular in young ages and people who support both of them are white. Therefore, we think our model is quite close to the real life situation because most our results match the facts that are happening. Lastly, we use post-stratification on each state to make further analysis.

In conclusion, we tried predict who will win the election 2020 by looking at the age, income, state and sex of these who have invention to vote. Even our model stated the winner will be Biden but this is not what we thought before and what will happen in the future since the result of the election is based on the electoral college not the actual vote. We do not know who will attend the electoral college from each state and what their political preference is, so it is hard to say who will win. However, we believe Trump has a higher probability to gain more votes since we read some articles which showed a higher degree of popularity in U.S.

Weaknesses

Firstly, according to our model selection, the logit model with cell agegroup has lower AIC than our original model with cell race. Although the difference of AIC for these two model is relative small (52.101), this may be a drawback since the model with cell agegroup has more accuracy, The second point is our sample size is not large enough. Furthermore, the limitation of computing power causes that we cannot add more variables in our cell to predict the vote result. Therefore, it may cause our model less accuracy. Lastly, our assumption is too idealized and simplified and the survey data was selected a few months ago, thus the data does not have time sensitivity.

Next Steps

A lot of tuning needs to be done to improve it. Firstly, we can improve the data set, enlarge our sample size for the model. Secondly, better equipment also can improve computing power which could add more variables in the cell in order to get more accurate predictions. This includes improved feature selection and perhaps principal component analysis for numerical variables.

References

1. Census dara

Team, MPC UX/UI. "U.S. CENSUS DATA FOR SOCIAL, ECONOMIC, AND HEALTH RESEARCH." IPUMS USA, 2020, usa.ipums.org/usa/index.shtml.

2. Survey data

Press, Courier &, et al. "New: Second Nationscape Data Set Release." Democracy Fund Voter Study Group, 30 Oct. 2020, www.voterstudygroup.org/publication/nationscape-data-set.

3. Post-stratification and further topic on stratification.

"6.3 - Poststratification and Further Topics on Stratification: STAT 506." PennState: Statistics Online Courses, online.stat.psu.edu/stat506/lesson/6/6.3.

4. Purpose of glmer model

Love, Kim, and Eliza. "What Is the Purpose of a Generalized Linear Mixed Model?" The Analysis Factor, 30 Oct. 2020, www.theanalysisfactor.com/what-is-the-purpose-of-a-generalized-linear-mixed-model/.

5. Template code from Prof Rohan Alexander and Sam Caetano, rohan.alexander@utoronto.ca

6.


```r
citation("lme4")
```

```
##
## To cite lme4 in publications use:
##
##   Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015).
##   Fitting Linear Mixed-Effects Models Using lme4. Journal of
##   Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Fitting Linear Mixed-Effects Models Using {lme4}},
##     author = {Douglas Bates and Martin M{\"a}chler and Ben Bolker and Steve Walker},
##     journal = {Journal of Statistical Software},
##     year = {2015},
##     volume = {67},
```

```
##     number = {1},
##     pages = {1--48},
##     doi = {10.18637/jss.v067.i01},
##   }
```

```r
citation("tidybayes")
```

```
##
## Kay M (2020). _tidybayes: Tidy Data and Geoms for Bayesian Models_.
## doi: 10.5281/zenodo.1308151 (URL:
## http://doi.org/10.5281/zenodo.1308151), R package version 2.1.1, <URL:
## http://mjskay.github.io/tidybayes/>.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {{tidybayes}: Tidy Data and Geoms for {Bayesian} Models},
##     author = {Matthew Kay},
##     year = {2020},
##     note = {R package version 2.1.1},
##     url = {http://mjskay.github.io/tidybayes/},
##     doi = {10.5281/zenodo.1308151},
##   }
```

```r
citation("caret")
```

```
##
## To cite package 'caret' in publications use:
##
##   Max Kuhn (2020). caret: Classification and Regression Training. R
##   package version 6.0-86. https://CRAN.R-project.org/package=caret
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {caret: Classification and Regression Training},
##     author = {Max Kuhn},
##     year = {2020},
##     note = {R package version 6.0-86},
##     url = {https://CRAN.R-project.org/package=caret},
##   }
```

```r
citation("pROC")
```

```
##
## If you use pROC in published research, please cite the following paper:
##
##   Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti,
##   Frédérique Lisacek, Jean-Charles Sanchez and Markus Müller (2011).
##   pROC: an open-source package for R and S+ to analyze and compare ROC
##   curves. BMC Bioinformatics, 12, p. 77.  DOI: 10.1186/1471-2105-12-77
##   <http://www.biomedcentral.com/1471-2105/12/77/>
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
```

```
##     title = {pROC: an open-source package for R and S+ to analyze and compare ROC curves},
##     author = {Xavier Robin and Natacha Turck and Alexandre Hainard and Natalia Tiberti and Frédériqu
##     year = {2011},
##     journal = {BMC Bioinformatics},
##     volume = {12},
##     pages = {77},
##   }
```

```r
citation("haven")
```

```
##
## To cite package 'haven' in publications use:
##
##   Hadley Wickham and Evan Miller (2020). haven: Import and Export
##   'SPSS', 'Stata' and 'SAS' Files. R package version 2.3.1.
##   https://CRAN.R-project.org/package=haven
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files},
##     author = {Hadley Wickham and Evan Miller},
##     year = {2020},
##     note = {R package version 2.3.1},
##     url = {https://CRAN.R-project.org/package=haven},
##   }
```

```r
citation("tidyverse")
```

```
##
##   Wickham et al., (2019). Welcome to the tidyverse. Journal of Open
##   Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {Welcome to the {tidyverse}},
##     author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agosti
##     year = {2019},
##     journal = {Journal of Open Source Software},
##     volume = {4},
##     number = {43},
##     pages = {1686},
##     doi = {10.21105/joss.01686},
##   }
```