CaseStudy_STA610

ElenaW.

11/18/2022

Introduction

Identifying and estimating how different groups will perform in big elections event is one of the most significant investigation in United States. In this study we could focus on political campaign analysis in North Carolina based on Bayesian hierarchical model evaluation. One original data about voter registration in 2016 and turnout data is from the North Carolina State Board of Elections (NCSBE), which the agency charged with the administration of the elections process and campaign finance disclosure and compliance. Another collects the Census data from Census Bureau in U.S. Federal Statistical System, and here we assume that these figures accurately represent populations in 2016. The goal will be designed by the interests in different registration performance in diverse demographic subgroups, various counties, separated genders and age groups for different party affiliations in 2016 selections.

Data Preprocess

1. First check all the notations for all columns in both datasets and convert the notations in Voter as same as in the Census dataset. Note here the dictionary for specific columns in Voter dataset:
`race`：
A: ASIAN, B: BLACK or AFRICAN AMERICAN, I: INDIAN AMERICAN or ALASKA NATIVE, M: TWO or MORE RACES, O: OTHER, U: UNDESIGNATED, W: WHITE (note here Native Hawaiian Or Other Pacific Islander Alone labeled as “O”)
`ethnic`：
HL: HISPANIC or LATINO, NL: NOT HISPANIC or NOT LATINO, UN: UNDESIGNATED
2. Checking the county identities in both dataset and indicating that both of them has the same `100 counties` and voter dataset is only for `voter` . Hence, dropping `stats_type`
3. we are not interested pin the information in `election_date` , `precinct_abbrv` , `vtd_abbrv` . Drop these 3 columns below.
4. Merge two datasets by the same geographical characteristic `county` , `race` , `ethnic` , `gender` , `age` and relative subgroups’ populations named `population_group` . Note here we assume that all of population is eligible to vote since all the records is only for 18+ age.
5. Some demographic subgroups are not identified in the registration part, which could be considered as Missing Completely at Random. However, missingness has large proportion of the dataset and should be imputed rather than dropped. Refer to the paper `Missing data and multiple imputation in clinical epidemiological research` , Multiple imputation is essentially an iterative form of stochastic imputation and provides unbiased and valid estimates of associations based on information from the available data. Here the variability is more accurate for each missing value since it considers variability due to sampling and due to imputation (standard error close to that of having full dataset with true values). Moreover, accounting for uncertainty allows us to calculate standard errors around estimations, which in turn leads to a better sense of uncertainty for the analysis. This method is also more flexible since it can be applied to any kind of data and any kind of analysis, and the researcher has flexibility in deciding how many imputations are necessary for the data at hand. Here I would use mice package to impute, and NULL variable is excluded for a priori except for `population_county` . But with “mincor = 0.1”, i decide to only use variables as predictor in the imputation model that are correlated with at least r=0.1 with the target-variable, and variables that are very weakly correlated are now left out. In addition, for `population_county` could be simply imputed by county based.
6. check the condition if `total_voters < population_group` , however, we get `3,466` observations violates this regulation. Good news is that they only take up small 0.6% proportion of the whole observations (`514,848`), which could be considered to drop.
7. Take a random sample of 30 counties out of all the counties in all datasets to satisfy the requirements:
“SCOTLAND”, “CHEROKEE”, “EDGECOMBE”, “ALAMANCE”, “BEAUFORT”, “ROCKINGHAM”, “GRAHAM”, “HARNETT”, “MCDOWELL”, “ORANGE”, “STOKES”, “GATES”, “WILKES”, “PERSON”, “SWAIN”, “FRANKLIN”, “CHOWAN” , “PITT”, “HAYWOOD”, “GRANVILLE”, “HALIFAX”, “LENOIR”, “STANLY”, “FORSYTH”, “DARE”, “PASQUOTANK”, “WAYNE”, “CALDWELL”, “SAMPSON”, “JOHNSTON”

Exploratory Data Analysis

Below demonstrates the demographic distributions for various characteristics such as `race` , `ethinic` , `age` , and `gender` . Figure1 obviously is showing that White people take the highest proportion in all counties and following is Black while other races is various in different counties. Figure2 displays that Hispanic group is much smaller than non-Hispanic in all counties. The gender groups in Figure4 look like around evenly separated, but should be tried in further analysis. Younger and older generations are minor groups. Therefore, these variables could be considered as effects in future models.

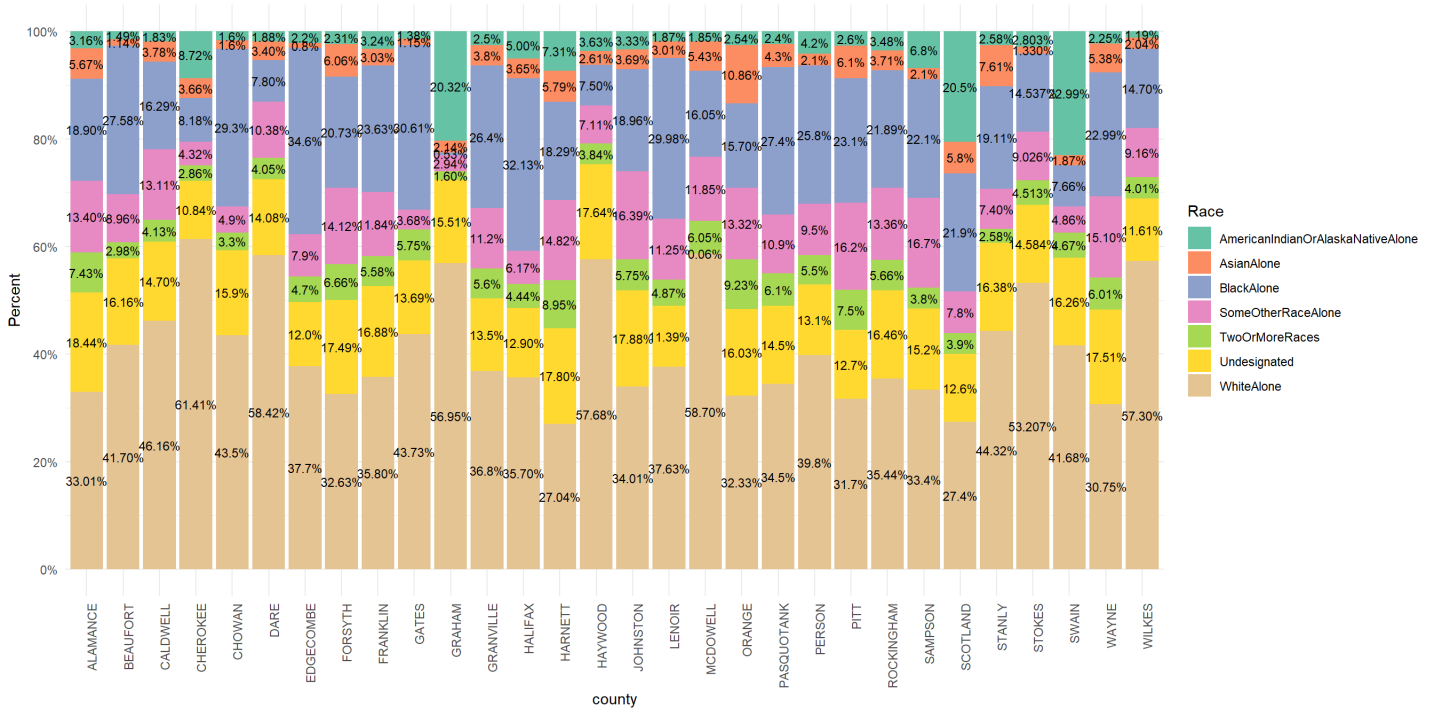

Figure1: Race Proportion in each County

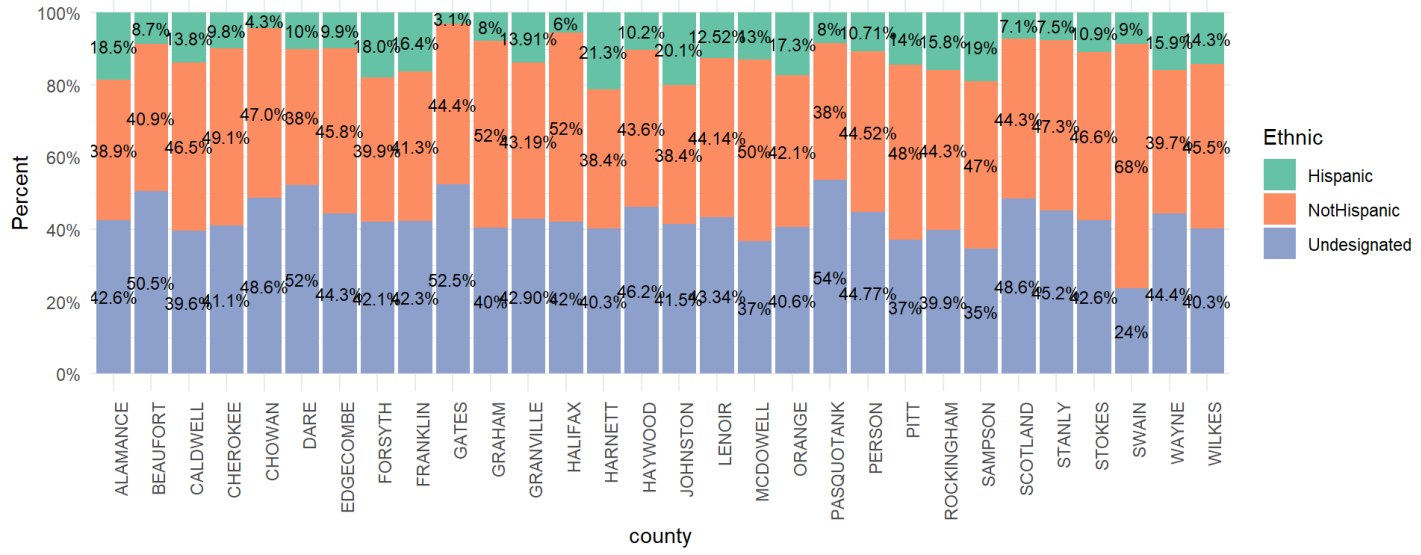

Figure2: Ethinity Proportion in each County

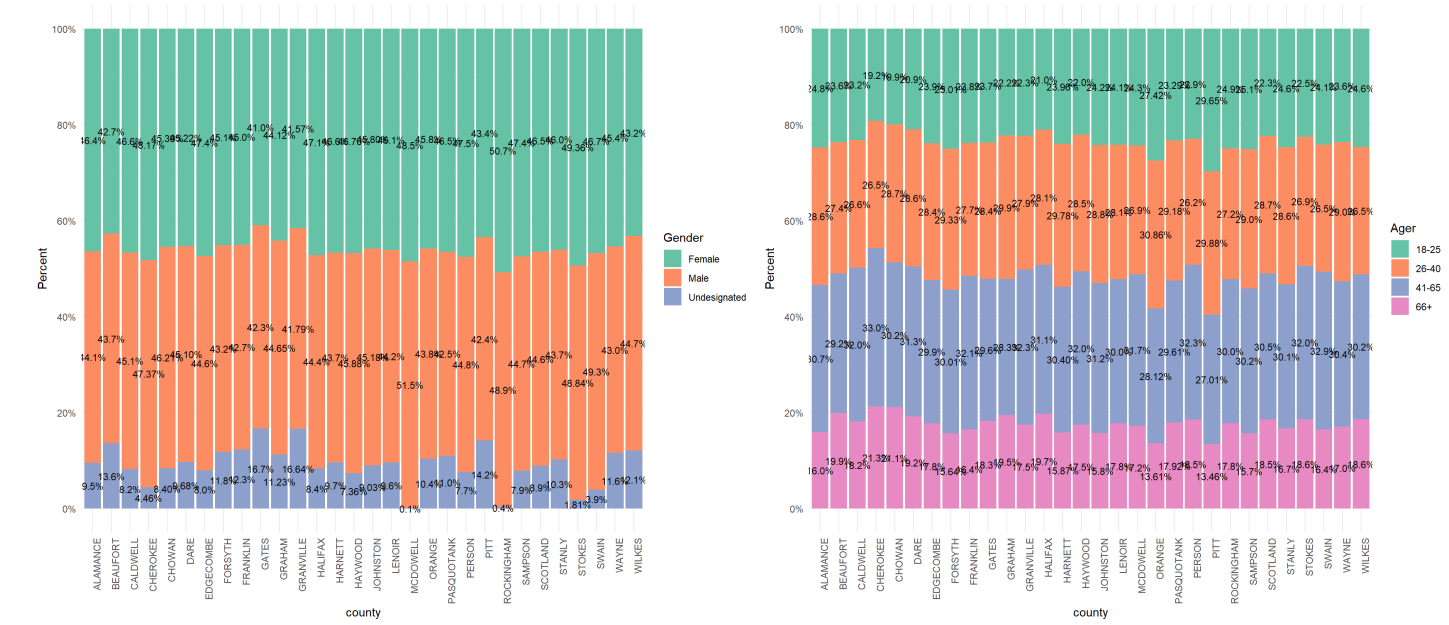

Figure3: Gender & Age Proportion in each County

Moreover, party affiliation is also our interests, however, we do not have party population in Census dataset. In this case, we could only simply have a brief view of party proportions differed by geographical characteristics. Note here to simplify the notations, I would categorize parties as “Democrat”, “Republican”, and “Others”. Figure4 shows some clear separation between different groups such as White Republican larger than Democrat. Except for Unknown characteristic, other groups are kind of have similar compositions to support diverse parties.

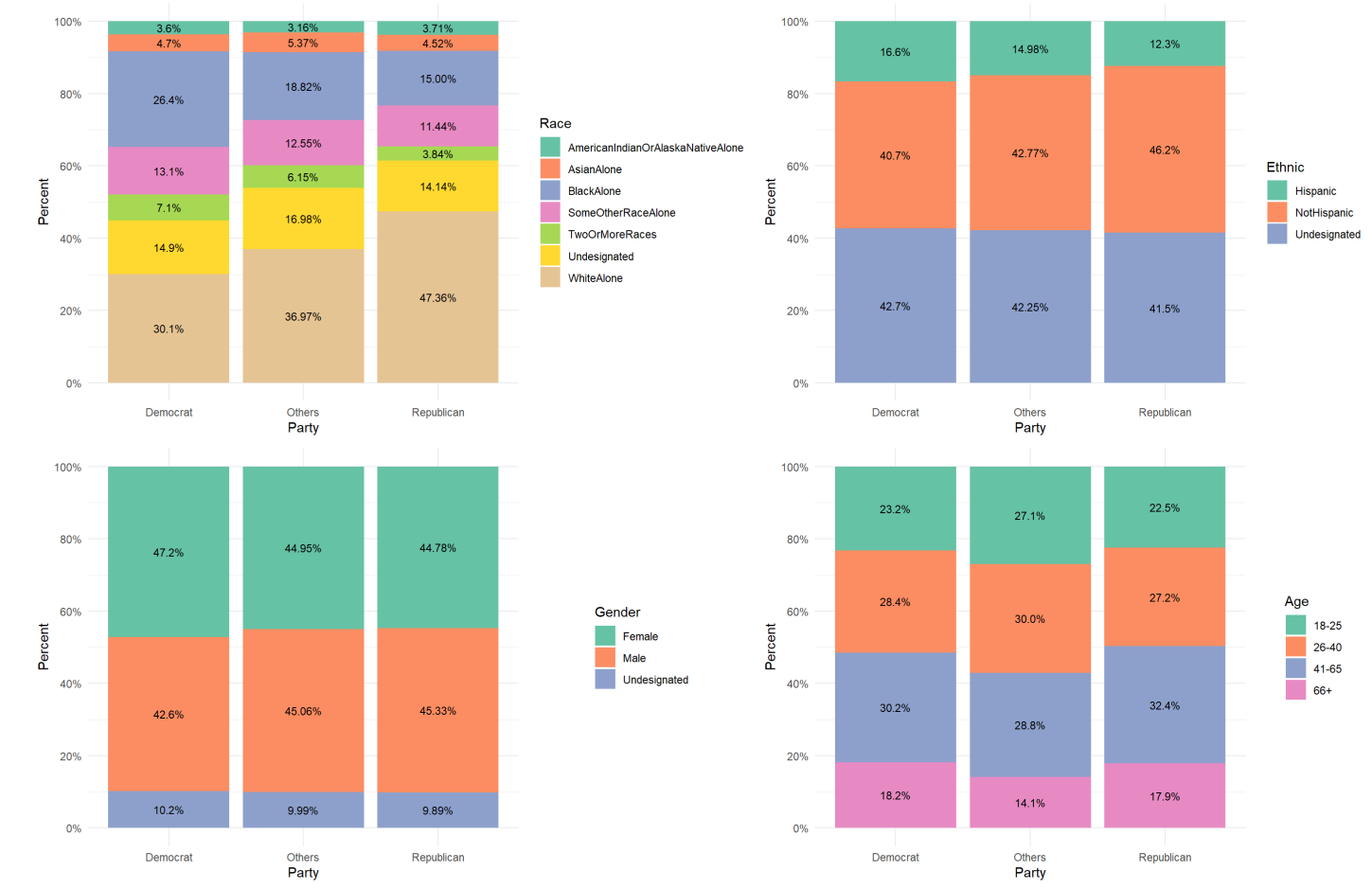

Figure4: Geographical Characteristics Proportion in each Party

`vote registration rate` could be calculated by the number of voters divided by eligible voting population. Taking consideration of future models, it is common to build log odds of the registration rate and check the distribution, which is taking the log of (number of voters/(eligible voting population-number of voters)). From Figure5, we could see that different groups are overlapped by each other, which means that they do

not have a obvious separations, but distributions preforms a quite differently in various groups. It is good to see that the distributions varied by counties are likely Gaussian, which satisfies the Bayesian Hierarchical Priors.

Figure6: Distributions for Log Odds of Registration Rate in Different Counties

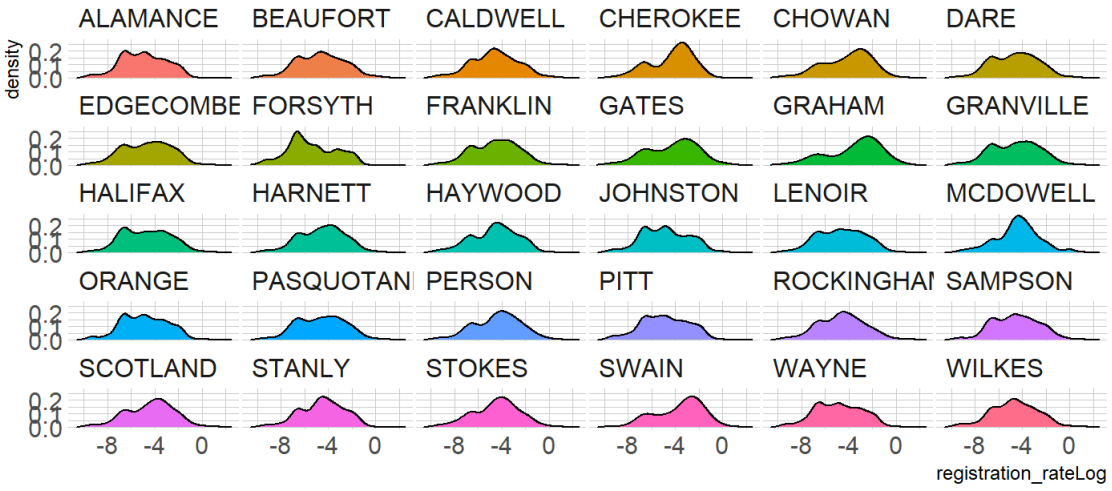

Figure7: Distributions for Log Odds of Registration Rate in Different Groups

Modelling

Demographic Estimation

The first model would evaluate the performance of the registration rate for males compare to the registration rate for females by controlling other cavariates and how differs between county groups. In this case, data is grouped by different counties, in which data is also grouped by various geographic characteristic. Hence, it is reasonable to build binomial logistic regression model to calculate the probability of registration of individuals in each demographic group within different counties. The covariates will include race, ethnic, gender, and age, and in this particular case, setting `gender` as random slope across different groups. For priors, intercept and other coefficient are poor N(0,1), standardization in county group are Half Cauchy common vague prior sto designate county level, and υ is good to set as lkj prior.
However, the whole 100k+ big dataset would take long time within gradient evaluation took 0.424 seconds, and only 1000 transition using 10 leapfrog steps per transition would take 4240 seconds. Due to model efficiency, I would only take 0.01% of original dataset.

$$y_{ij} \sim binomial(n_{ij}, \theta_{ij}), j = 1...30, i = 1...n_j$$

where y_{ij}: number of registrated voters in ith. geographic groups in jth. country
n_{ij}: population in ith. geographic groups in jth. country
θ_{ij}: the probability of registration of individuals in ith. geographic groups in jth. country

$$logit(\theta_{ij}) = \beta_{0,j} + \beta_{1,j}gender_i + \beta_{2,j}age_i + \beta_{3,j}ethnic_i + \beta_{3,j}race_i$$
$$\beta_{0,j} = \beta_0 + b_{0,j} \sim N(\beta_0, \sigma^2), \beta_{1,j} = \beta_1 + b_{1,j}$$
$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \end{bmatrix} \overset{i.i.d.}{\sim} N(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{pmatrix} v \begin{pmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{pmatrix})$$
$$j = 1...10, i = 1...n_j$$

The table below shows the intercept of each group and the model have random intercepts and slopes for `Gender`, which estimates that Male has $e^{-}0.0458 = 0.955$ times the odds of registration compared with females, and females will have higher odds of registration rate. Morevoer, people who above 66 years old tend to have higher registration rate, and with NotHispanic ethnic population would have more probability to vote. Figure8 displays the corresponding 95% credical intervals for Bayesian model. The plots below also show that these confidence intervals largely overlap each other and some counties have pretty various large range, especially for CHOWAN, hence the inference on these overall rankings has high variability.

```
##                          Estimate  Est.Error    l-95% CI      u-95% CI
## Intercept              -5.70332209 0.20612280 -6.08646169 -5.2676116258
## ethnicNotHispanic       1.14252695 0.08548513  0.98166402  1.3109038257
## ethnicUndesignated     -0.20228681 0.08454619 -0.36472734 -0.0342968605
## raceAsianAlone         -0.63990895 0.15299961 -0.94268862 -0.3489936869
## raceBlackAlone         -0.24925028 0.12873168 -0.50331852  0.0065539617
## raceSomeOtherRaceAlone  0.51249199 0.15486502  0.19473354  0.8145990661
## raceTwoOrMoreRaces      0.60228634 0.16080323  0.28087431  0.9012227592
## raceUndesignated       -0.91850809 0.14008559 -1.19907820 -0.6460021847
## raceWhiteAlone         -0.26250232 0.12836453 -0.51967093  0.0003669936
## genderMale             -0.04581746 0.28690579 -0.63855688  0.4991433972
## genderUndesignated     -0.16355405 0.24059882 -0.65573827  0.3011888712
## age26M40                0.10168112 0.03136278  0.04098756  0.1621253999
## age41M65                0.31751255 0.02877093  0.26024987  0.3727262529
## age66P                  0.58666637 0.03048515  0.52747740  0.6474405773
##                             Rhat Bulk_ESS Tail_ESS
## Intercept              1.002051 277.5476 404.3500
## ethnicNotHispanic      1.001092 901.7257 874.1911
## ethnicUndesignated     1.000858 872.8292 897.3654
## raceAsianAlone         1.004972 625.1122 678.2214
## raceBlackAlone         1.007647 554.5214 557.0437
## raceSomeOtherRaceAlone 1.006551 598.9874 549.4126
## raceTwoOrMoreRaces     1.008136 560.8733 784.5353
## raceUndesignated       1.007923 558.9997 720.1176
## raceWhiteAlone         1.008564 564.3297 678.9046
## genderMale             1.003372 211.3470 302.8417
## genderUndesignated     1.003962 533.9111 553.2445
## age26M40               1.002919 847.0667 714.3611
## age41M65               1.002927 905.9489 677.4664
## age66P                 1.002116 952.6733 853.2014
```

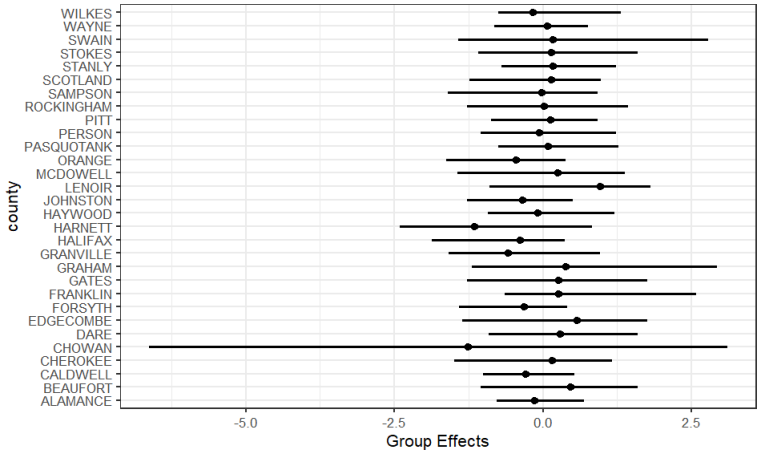

Figure8: Counties specific group effects on registration rate

Model Evaluation

For logistic regression, there are a few specific diagnostics one should examine. The following shows three binned residual plots, with each point showing y'-y, where y' is based on simulated data from the posterior predictive distributions, and y is the observed data. Note that we need the binned residuals or predictive errors, as the prediction error is either 0 or 1, as shown in the Figure9 below. The binned margins were based on the observed data, whereas the dots were predictive errors from replicated data. The model fits well as most of residual are around 0. The posterior distribution below (Figure10) is showing that the estimations center at the frequentist estimates, which proves that both of Bayesian and frequency models are good here, and we don't need adjust the informative priors and assumptions. The trace plot does not provide the evidence of non-convergence, which is also good for model results.

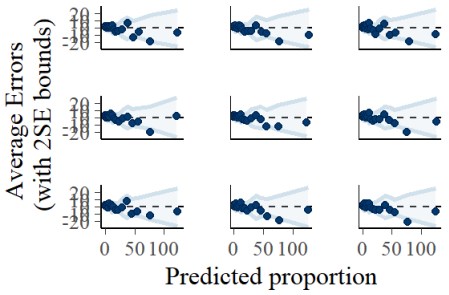

Figure9: three binned residual plots in model1

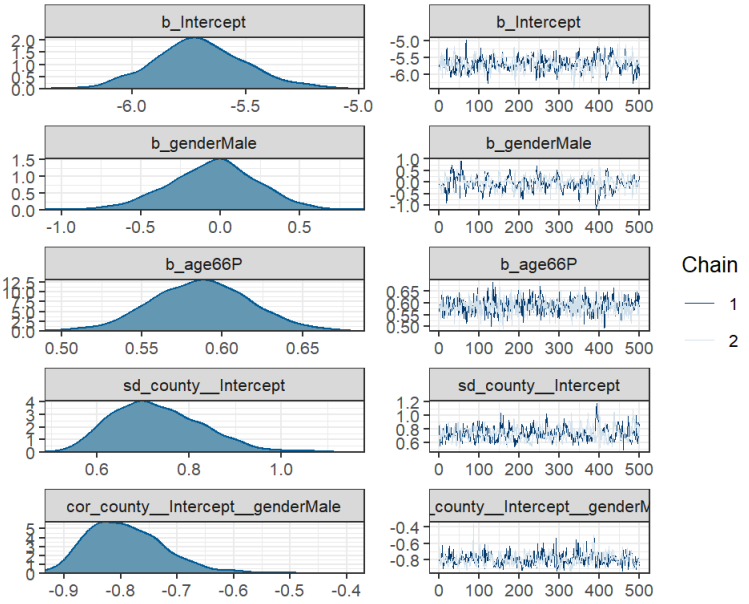

Figure10: Bayesian random effects model posterior distributions and traceplots in model1

Party Affiliation Analysis

This part would take consideration of `party` predictors such as gender and age differences in registration rates across parties. However, we do not have population in terms of parties group in Census. Thus the model are modified with conditional distributions firstly by `population_county` while filtering only `Democrat`, and then the second model will be built by the population except for Democrat groups while filtering `Republican`, and the final model could be evaluated from rest of them. The multinomial models with conditional distributions are developed below:

$$y_{ij;d} \sim binomial(n_{ij}, \theta_{ij}; d)$$
$$y_{ij;r}|y_{ij;d} \sim binomial(n_{ij} - y_{ij;d}, \theta_{ij;r|d})$$
$$y_{ij;o}|y_{ij;r}, y_{ij;d} \sim \sim binomial(n_{ij} - y_{ij;d} - y_{ij;r}, \theta_{ij;o|d,r})$$

where $y_{ij;d}, y_{ij;r}, y_{ij;o}$: respective number of registered voters in ith. geographic groups in jth. country for that party d,r,o
n_{ij}: total population in jth. country
$\theta_{ij;d}$: for Democrat voter, the probability of registration of individuals in ith. geographic groups in jth. country
$\theta_{ij;r|d}$: conditional probability that describe the probability of an individual in the same group and county to vote as a Republican, given by the individual not being a Democrat
$\theta_{ij;o|d,r}$: conditional probability that describe the probability of an individual in the same group and county to vote as other parties, given by individual not being both of Democrat and Republican
The fixed effects and random effects are built as same as the first model, and here I also add `gender` to vary across counties.

The tables below compare the gender effects among different parties and states that Democrat female have more probabilities to vote than male, which is the similar result we get in the first model. In addition, for Republican and Other parties, the odd ratios for male are really close to 1, which means that the registration rate do not separate gender, and they have similar probabilities to register, which satisfies the eda analysis from original data.

Furthermore, I also combine the results of different age groups for diverse parties. `age41M65` are the most active group to register for all parties, which is reasonable in reality and good to satisfies the finding we discovered in eda.

```
##                Estimate Est.Error   l-95% CI  u-95% CI     Rhat Bulk_ESS
## genderMale  -0.08665021 0.2538981 -0.5753599 0.4237551 1.003684 250.7107
## genderMale1  0.19253715 0.2315025 -0.2347200 0.6786595 1.006102 268.8737
## genderMale2  0.04162340 0.1851360 -0.3302108 0.4077157 1.009136 341.2208
##             Tail_ESS
## genderMale  508.3700
## genderMale1 302.6203
## genderMale2 542.1358
```

```
##            Estimate  Est.Error  l-95% CI  u-95% CI      Rhat  Bulk_ESS Tail_ESS
## age26M40  0.5852853 0.06288239 0.4675445 0.7109931 1.0001222  980.2619 845.2288
## age41M65  1.1322231 0.05966824 1.0134785 1.2501321 1.0021379  858.3501 738.5750
## age66P    1.1264504 0.06134613 1.0049599 1.2478176 1.0024059  952.4944 740.9975
## age26M401 0.6485730 0.07132427 0.5121424 0.7910791 1.0029870  934.9174 713.1447
## age41M651 1.4728353 0.06992188 1.3417348 1.6147965 0.9995427  869.5349 772.9718
## age66P1   1.0436982 0.07399328 0.9014560 1.1963592 1.0011239  866.5625 827.4207
## age26M402 0.3718354 0.02943352 0.3154402 0.4277962 1.0021204  976.4632 764.4824
## age41M652 0.9052634 0.02841133 0.8510775 0.9598928 1.0005900 1042.4785 699.6821
## age66P2   0.6361210 0.02936022 0.5761268 0.6939774 1.0014154 1081.7533 694.0117
```

Figure12 is the corresponding 95% credical intervals for Democrat model, which displays that these confidence intervals more largely overlap each other and all counties have larger range than the first model. This situation makes sense because the actual performance is different among diverse parties within the same group but they have more similar behavior in the same party across groups. Therefore, the inference on these overall rankings has higher variability, and we really distinguish which county has more obvious performance.

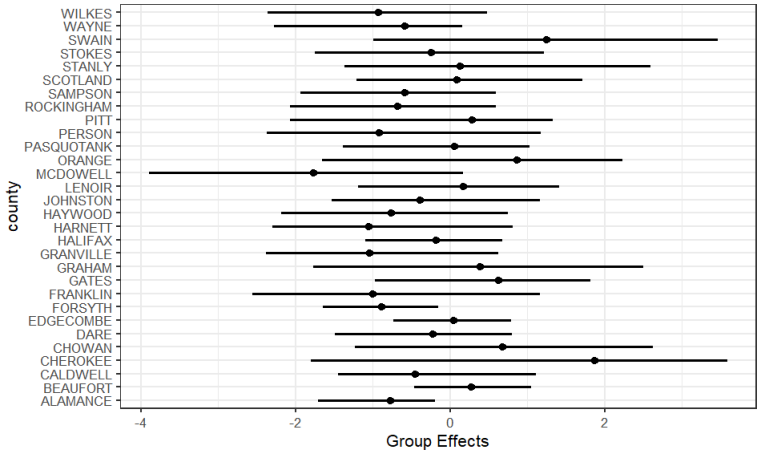

Figure12: Counties specific group effects on registration rate

Model Evaluation

Here I just randomly choose the first Democrat group to interpret. the three binned residual plots (Figure13)is showing that the model fits well as most of residual are around 0. The posterior distribution below (Figure14) is showing that the estimations center at the frequentist estimates, which proves that both of Bayesian and frequency models are good here, and we don't need adjust the informative priors and assumptions. The trace plot does not provide the evidence of non-convergence, which is also good for model results as well.

It's a good time to compare *AUC* for above models. Below is a table showing AUC comparison, and we could conculde that the first model, which does not have Part affiliation preforms better, which AUC is 0.6047611.

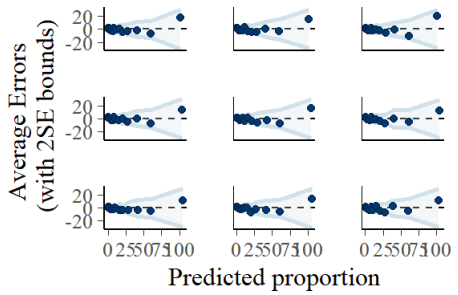

Figure13: three binned residual plots in model2

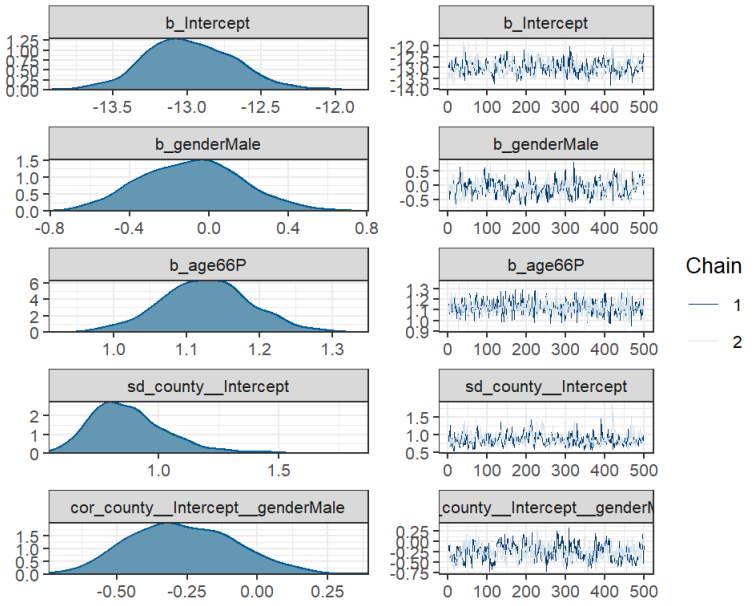

Figure14: Bayesian random effects model posterior distributions and traceplots in model2

```
##   models      AUCs
## 1 model1 0.6047611
## 2 model2 0.5365245
## 3 model3 0.6034004
```

Conclusion

So far, we have investigate the relationships between demographic subgroups especially for gender and discover the registration performance across counties. Meanwhile, we also evaluate these correlations within party affiliations by multinomial logistic regression from Bayesian perspective. Moreover, we have also discussed that all models are reasonable to develop and have a good fitness. Most of findings also satisfy the observation from eda part for original data.

To sum up, regardless party affiliation, females will have higher odds of registration rate then male. Moreover, people who above 66 years old tend to have higher registration rate, and with NotHispanic ethnic population would have more probability to vote. The behavious across counties largely overlap each other and some counties have pretty various large range, especially for CHOWAN, hence the inference on these overall rankings has high variability but we have somehow confident to conclude that registration event performs bit similar across counties due to large overlap. After counting parties, Democrat female have more probabilities to vote than male, and for Republican and Other parties, male and female have the similar probability to register. In addition, `age41M65` are the most active group to register for all parties, which is reasonable in reality and good to satisfies the finding we discovered in eda.

However, the above models have limitation should be considered in future analysis. First, since Bayesian way is extremely time consuming, I only sample around 1000 observations out of over 500k+ recordings to improve model efficiency. In this case, our sampling cannot represent population and worse to predict. The second limitations is that the original surveys has `undesignated` group, which reduce the accuracy for our models and should be noticed in future analysis. Moreover, we assume census in 2020 is as same as in 2016, which is also not appropriate for this study, and the data does not record population differed by parties, which arises a big problem in our study.

```
## ----setup, include=FALSE----------------------------------
knitr::opts_chunk$set(echo=F, eval=T, cache=F, warning=F, message=F,
fig.align="center", fig.pos="H")
library(tidyverse)
library(ggplot2)
library(lme4)
library(knitr)
library(sjPlot)
library(sjmisc)
library(glmmTMB)
library(knitr)
library(kableExtra)
# for grid
library(lubridate)
library(patchwork)
library(gridExtra)
library(brms)
library(lattice) # dotplot


## ---- include=FALSE----------------------------------------
voter = read.table("voter_stats_20161108.txt", sep="", header=TRUE)
census = read.table("Census2010_long.txt", sep="", header=TRUE)


## ---- include=FALSE----------------------------------------
# clean Census dataset by the same notation as voter dataset
unique(voter$sex_code)
unique(census$Gender)

unique(voter$age)
unique(census$Age)

unique(voter$ethnic_code)
unique(census$Hispanic)

unique(voter$race_code)
unique(census$Race)

# check county identity in both
length(unique(voter$county_desc))
length(unique(census$Geography))

# check states_type
unique(voter$stats_type)

# drop election_date, stats_type, precinct_abbrv, vtd_abbrv, party_cd
drop = c("election_date", "stats_type", "precinct_abbrv", "vtd_abbrv")
voter = voter[,!(names(voter) %in% drop)]


## ---- include=FALSE----------------------------------------
voter = voter %>% group_by(sex_code) %>%
  mutate(sex_code = case_when(
    sex_code == "F"~"Female",
    sex_code == "M"~"Male",
```

```
    sex_code == "U"~"Undesignated",
    TRUE ~ sex_code
  ))
voter = voter %>% group_by(age) %>%
  mutate(age = case_when(
    age == "Age 18 - 25"~"18-25" ,
    age == "Age 26 - 40"~"26-40" ,
    age == "Age 41 - 65"~"41-65" ,
    age == "Age Over 66"~"66+"   ,
    TRUE ~ age
  ))
voter = voter %>% group_by(ethnic_code) %>%
  mutate(ethnic_code = case_when(
    ethnic_code == "HL"~"Hispanic"    ,
    ethnic_code == "NL"~"NotHispanic" ,
    ethnic_code == "UN"~"Undesignated",
    TRUE ~ ethnic_code
  ))
voter = voter %>% group_by(race_code) %>%
  mutate(race_code = case_when(
    race_code == "W"~"WhiteAlone"                                ,
    race_code == "B"~"BlackAlone"                                ,
    race_code == "I"~"AmericanIndianOrAlaskaNativeAlone"         ,
    race_code == "A"~"AsianAlone"                                ,
    race_code == "U"~"Undesignated"                              ,
    race_code == "O"~"SomeOtherRaceAlone"                        ,
    race_code == "M"~"TwoOrMoreRaces"                            ,
    TRUE ~ race_code
  ))
voter = voter %>% group_by(race_code) %>%
  mutate(race_code = case_when(
    race_code == "W"~"WhiteAlone"                                ,
    race_code == "B"~"BlackAlone"                                ,
    race_code == "I"~"AmericanIndianOrAlaskaNativeAlone"         ,
    race_code == "A"~"AsianAlone"                                ,
    race_code == "U"~"Undesignated"                              ,
    race_code == "O"~"SomeOtherRaceAlone"                        ,
    race_code == "M"~"TwoOrMoreRaces"                            ,
    TRUE ~ race_code
  ))

#census = census %>% group_by(Race) %>%
#  mutate(Race = case_when(
#    Race == "NativeHawaiianOrOtherPacificIslanderAlone"~"SomeOtherRaceAlone",
#    TRUE ~ Race
#  ))
# change columns name
colnames(voter) = c("county", "party", "race", "ethnic", "gender",
"age","total_voters")
colnames(census) = c("county", "age", "gender", "ethnic", "race",
"population_group","population_county")


## ---- include=FALSE----------------------------------------
# merge two datasets
voter_nop = subset(voter, select=-c(party))
```

```
df = left_join(voter_nop, census, by=c("county", "age", "gender", "ethnic",
"race"))


## ---- include=FALSE----------------------------------------
# Multiple Imputation method for population_group
# https://statistics.ohlsen-web.de/multiple-imputation-with-mice/
library(mice)
set.seed(1234)
df_imp = subset(df, select = -c(population_county))
predictormatrix = quickpred(df_imp,
                            exclude=NULL,
                            mincor = 0.1)
imp_gen = mice(df_imp,
               predictorMatrix = predictormatrix,
               method = c('pmm'), # predictive mean matching as imputation
method
               diagnostics=TRUE,
               seed = 1234)
completedData = complete(imp_gen,1)


## ---- include=FALSE----------------------------------------
# change column name
colnames(df)[8] = "population_county"

# impute population_county by county groups
df = df %>%
  group_by(county) %>%
  fill(population_county) %>%
  fill(population_county, .direction = "up")
df = cbind(completedData, df$population_county)

# change column name
colnames(df)[8] = "population_county"

# check total_voters vs population_group
drop_list = which(df$total_voters>df$population_group)
df = df[-which(df$total_voters>df$population_group),]


## ---- include=FALSE----------------------------------------
# update voter for model Part2 (party)
voter = voter[-drop_list,]
df2 = cbind(df, voter$party)
# change column name
colnames(df2)[9] = "party"

df2$party[df2$party == "DEM"] = "Democrat"
df2$party[df2$party == "REP"] = "Republican"
df2$party[df2$party != "Democrat" & df2$party != "Republican"] = "Others"


## ---- include=FALSE----------------------------------------
# get 30 samples
set.seed(1)
```

```
county_list = df %>%
  group_by(county) %>%
  summarise(total_voters = sum(total_voters)) %>%
  sample_n(30)

df = df %>%
  filter(county %in% county_list$county)
voter = voter %>%
  filter(county %in% county_list$county)
census = census %>%
  filter(county %in% county_list$county)
df2 = df2 %>%
  filter(county %in% county_list$county)


## ----fig.height=7, fig.width=14, fig.cap="Figure1: Race Proportion in each
County"----
library(scales)
# create segmented bar chart
# adding labels to each segment
plotdata <- df %>%
  group_by(county, race) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

ggplot(plotdata,
       aes(x = county,
           y = pct,
           fill = factor(race))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Race",
       x = "county") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


## ----fig.height=4, fig.width=10, fig.cap="Figure2: Ethinity Proportion in
each County"----
plotdata <- df %>%
  group_by(county, ethnic) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

ggplot(plotdata,
       aes(x = county,
           y = pct,
```

```
          fill = factor(ethnic))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Ethnic",
       x = "county") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


## ----fig.height=8, fig.width=18, fig.cap="Figure3: Gender & Age Proportion
in each County"----
plotdata <- df %>%
  group_by(county, gender) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pt1 = ggplot(plotdata,
       aes(x = county,
           y = pct,
           fill = factor(gender))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Gender",
       x = "county") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

plotdata <- df %>%
  group_by(county, age) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pt2 = ggplot(plotdata,
       aes(x = county,
           y = pct,
           fill = factor(age))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
```

```
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Ager",
       x = "county") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

pt1+pt2+plot_layout(ncol=2)


## ---- include=FALSE--------------------------------------
voter$party[voter$party == "DEM"] = "Democrat"
voter$party[voter$party == "REP"] = "Republican"
voter$party[voter$party != "Democrat" & voter$party != "Republican"] =
"Others"


## ----fig.height=10, fig.width=15, fig.cap="Figure4: Geographical
Characteristics Proportion in each Party"----
# create segmented bar chart
# adding labels to each segment
plotdata <- voter %>%
  group_by(party, race) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pl1 = ggplot(plotdata,
       aes(x = party,
           y = pct,
           fill = factor(race))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Race",
       x = "Party") +
  theme_minimal()

plotdata <- voter %>%
  group_by(party, ethnic) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pl2 = ggplot(plotdata,
       aes(x = party,
           y = pct,
```

```
          fill = factor(ethnic))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Ethnic",
       x = "Party") +
  theme_minimal()

plotdata <- voter %>%
  group_by(party, gender) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pl3 = ggplot(plotdata,
       aes(x = party,
           y = pct,
           fill = factor(gender))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
       fill = "Gender",
       x = "Party") +
  theme_minimal()

plotdata <- voter %>%
  group_by(party, age) %>%
  summarize(n = n()) %>%
  mutate(pct = n/sum(n),
         lbl = scales::percent(pct))

pl4 = ggplot(plotdata,
       aes(x = party,
           y = pct,
           fill = factor(age))) +
  geom_bar(stat = "identity",
           position = "fill") +
  scale_y_continuous(breaks = seq(0, 1, .2),
                     label = percent) +
  geom_text(aes(label = lbl),
            size = 3,
            position = position_stack(vjust = 0.5)) +
  scale_fill_brewer(palette = "Set2") +
  labs(y = "Percent",
```

```
       fill = "Age",
       x = "Party") +
  theme_minimal()

pl1+pl2+pl3+pl4+plot_layout(ncol=2)


## ---- include=FALSE----------------------------------------
df$registration_rate = df$total_voters/df$population_group
df$registration_rateLog = log(df$total_voters/(df$population_group-
df$total_voters))


## ----fig.height=4, fig.width=8, fig.cap="Figure6: Distributions for Log Odds
of Registration Rate in Different Counties"----
pl1 <- ggplot(df,
              aes(x = registration_rateLog, fill = race, color = race)) +
  geom_density(alpha = 0.3) +
  labs(caption = "Race")

pl2 <- ggplot(df,
              aes(x = registration_rateLog, fill = ethnic, color = ethnic)) +
  geom_density(alpha = 0.3) +
  labs(caption = "Ethnic")

pl3 <- ggplot(df,
              aes(x = registration_rateLog, fill = gender, color = gender)) +
  geom_density(alpha = 0.3) +
  labs(caption = "Gender")

pl4 <- ggplot(df,
              aes(x = registration_rateLog, fill = age, color = age)) +
  geom_density(alpha = 0.3) +
  labs(caption = "Age")

pl1+pl2+pl3+pl4+plot_layout(ncol=2)


## ----fig.height=4, fig.width=8, fig.cap="Figure7: Distributions for Log Odds
of Registration Rate in Different Groups"----
library(hrbrthemes)
# Using Small multiple
ggplot(df, aes(registration_rateLog, group=county, fill=county)) +
    geom_density(adjust=1.5) +
    theme_ipsum() +
    facet_wrap(~county) +
    theme(
      legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank()
    )


## ---- include=FALSE----------------------------------------
df_samp = df[sample(nrow(df), size=0.01*nrow(df)), ]
model1 = brm(data = df_samp,
```

```
             family = binomial,
      total_voters|trials(population_group)~1+(1+gender|county)
+ethnic+race+gender+age,
      prior = c(prior(normal(0, 1), class=Intercept),
                prior(normal(0, 1), class=b),
                prior(cauchy(0, 1), class="sd",group="county"),
                prior(lkj(1), class = "cor")),
                iter = 1000, warmup = 500, chains = 2)


## ---- out.width="0.1%", out.height="0.1%", out.extra="0.1%"----
summary1 = summary(model1)
summary1$fix


## ---- fig.height=3, fig.width=5, fig.cap="Figure8: Counties specific group
effects on registration rate",echo=FALSE----
library(tidybayes)
model1 %>%
  spread_draws(r_county[county,]) %>%
  median_qi(`Group Effects` = r_county) %>%
  ggplot(aes(y=county, x=`Group Effects`, xmin=.lower, xmax=.upper)) +
  geom_pointinterval(orientation="horizontal", size=0.8) +
  theme_bw(base_size = 8)


## ---- fig.height=2, fig.width=3, fig.cap="Figure9: three binned residual
plots in model1"----
pp_check(model1, type = "error_binned", ndraws = 9)


## ---- fig.height=4, fig.width=5, fig.cap="Figure10: Bayesian random effects
model posterior distributions and traceplots in model1"----
plot(model1, variable=c("b_Intercept","b_genderMale", "b_age66P",
"sd_county__Intercept", "cor_county__Intercept__genderMale" ),
     theme=theme_bw(base_size = 10))


## ---- include=FALSE---------------------------------------
# Using the pROC package
AUC1 = pROC::roc(response = df_samp$total_voters,
          predictor = predict(model1, type = "response")[ , "Estimate"],
          plot = F, print.auc = TRUE)


## ---- include=FALSE---------------------------------------
df2_samp = df2[sample(nrow(df2), size=0.01*nrow(df2)), ]

model2 = brm(data = (df2_samp %>% filter(party == "Democrat")),
      family = binomial,
      total_voters|trials(population_county)~1+(1+gender|county)
+ethnic+race+gender+age,
      prior = c(prior(normal(0, 1), class=Intercept),
                prior(normal(0, 1), class=b),
                prior(cauchy(0, 1), class="sd",group="county"),
                prior(lkj(1), class = "cor")),
```

```
                iter = 1000, warmup = 500, chains = 2)


## ---- include=FALSE---------------------------------------
part1 = df2_samp %>%
  group_by(county, party) %>%
  summarise(sum(total_voters))
part1 = part1[which(part1$party == "Democrat"),]
part2 = df2_samp %>% group_by(county) %>%
summarise(population_county=first(population_county))
part3 = merge(part1, part2, by="county")
part3$population_noDem = part3$population_county - part3$`sum(total_voters)`
df2_samp3 = left_join(df2_samp, part3, by="county") %>% select(-
c("population_county.y", `sum(total_voters)`, "party.y"))
colnames(df2_samp3)[9] = "party"
colnames(df2_samp3)[8] = "population_county"


## ---- include=FALSE---------------------------------------
model3 = brm(data = (df2_samp3 %>% filter(party == "Republican")),
      family = binomial,
      total_voters|trials(population_noDem)~1+(1+gender|county)
+ethnic+race+gender+age,
      prior = c(prior(normal(0, 1), class=Intercept),
                prior(normal(0, 1), class=b),
                prior(cauchy(0, 1), class="sd",group="county"),
                prior(lkj(1), class = "cor")),
                iter = 1000, warmup = 500, chains = 2)


## ---- include=FALSE---------------------------------------
part1 = df2_samp3 %>%
  group_by(county, party) %>%
  summarise(sum(total_voters))
part1 = part1[which(part1$party == "Republican"),]
part2 = df2_samp3 %>% group_by(county) %>%
summarise(population_noDem=first(population_noDem))
part3 = merge(part1, part2, by="county")
part3$population_noRep = part3$population_noDem - part3$`sum(total_voters)`
df2_samp4 = left_join(df2_samp3, part3, by="county") %>% select(-
c("population_noDem.x", `sum(total_voters)`, "party.y"))
colnames(df2_samp4)[10] = "population_noDem"
colnames(df2_samp4)[9] = "party"


## ---- include=FALSE---------------------------------------
model4 = brm(data = df2_samp4,
      family = binomial,
      total_voters|trials(population_noRep)~1+(1+gender|county)
+ethnic+race+gender+age,
      prior = c(prior(normal(0, 1), class=Intercept),
                prior(normal(0, 1), class=b),
                prior(cauchy(0, 1), class="sd",group="county"),
                prior(lkj(1), class = "cor")),
                iter = 1000, warmup = 500, chains = 2)
```

```
## ---- out.width="0.1%", out.height="0.1%", out.extra="0.1%"----
summary2 = summary(model2)
gender2 = summary2$fix["genderMale",]
summary3 = summary(model3)
gender3 = summary3$fix["genderMale",]
summary4 = summary(model4)
gender4 = summary4$fix["genderMale",]
rbind(gender2, gender3, gender4)


## ---- out.width="0.1%", out.height="0.1%", out.extra="0.1%"----
age2 = summary2$fix[c("age26M40", "age41M65", "age66P"),]
age3 = summary3$fix[c("age26M40", "age41M65", "age66P"),]
age4 = summary4$fix[c("age26M40", "age41M65", "age66P"),]
rbind(age2, age3, age4)


## ---- fig.height=3, fig.width=5, fig.cap="Figure12: Counties specific group
effects on registration rate",include=TRUE,echo=FALSE----
library(tidybayes)
model2 %>%
  spread_draws(r_county[county,]) %>%
  median_qi(`Group Effects` = r_county) %>%
  ggplot(aes(y=county, x=`Group Effects`, xmin=.lower, xmax=.upper)) +
  geom_pointinterval(orientation="horizontal", size=0.8) +
  theme_bw(base_size = 8)


## ---- fig.height=2, fig.width=3, fig.cap="Figure13: three binned residual
plots in model2"----
pp_check(model2, type = "error_binned", ndraws = 9)


## ---- fig.height=4, fig.width=5, fig.cap="Figure14: Bayesian random effects
model posterior distributions and traceplots in model2"----
plot(model2, variable=c("b_Intercept","b_genderMale", "b_age66P",
"sd_county__Intercept", "cor_county__Intercept__genderMale" ),
     theme=theme_bw(base_size = 10))


## ------------------------------------------------------------
# Using the pROC package
response2 = df2_samp %>% filter(party == "Democrat")
response3 = df2_samp3 %>% filter(party == "Republican")

AUC2 = pROC::roc(response = response2$total_voters,
          predictor = predict(model2, type = "response")[ , "Estimate"],
          plot = FALSE, print.auc = TRUE)
AUC3 = pROC::roc(response = response3$total_voters,
          predictor = predict(model3, type = "response")[ , "Estimate"],
          plot = FALSE, print.auc = TRUE)
models = c("model1", "model2", "model3")
AUCs = c(AUC1$auc, AUC2$auc, AUC3$auc)
data.frame(models, AUCs)
```