Senate Election Forecast 2022 Using Metropolis-Hastings and Markov Chain Monte Carlo Sampling

Yaswanth Pothuru

Abstract

Aside from the obvious interest in predicting the likely outcome of the 2022 U.S. Senate election, it is also useful to anticipate the outcome of that race in each state so that last-minute campaign resources can be spent properly. Polls assessing voter intentions and historical data on the outcomes of prior presidential elections in each state are potential sources of evidence to support such projections. I provide a model based on these two sources in this work. The model considers the election in each state I to be the result of a Bernoulli p_i trial. A series of (approximately) weekly polls in each state are also treated as the results of a p_i experiment. To account for changes in p_i during the campaign, these polls are preceded by a power prior with exponent a_0. Each state's electoral history is used to create an informative prior on p_i. The values for p_i and a_0 are then approximated using Metropolis-Hastings-within-Gibbs Markov-chain Monte Carlo sampling. The p_i values are utilized to generate an election result. Finally, we predicted that Democrats would take hold of the US Senate in 2022.

1. Introduction

The 2022 senate elections in the United States proved to be more interesting as the winning party will take the power of passing the bills that are taken by the president. There are numerous potential predictors that might be used to forecast the outcome of this election. This election is widely seen as distinctive in a variety of ways, and this perceived uniqueness can make it tempting to use a wide range of such predictions. Given that the two major-party candidates look to have powerful and fascinating demographic splits1, one can reasonably seek to anticipate the rates of voter turnout among various demographic groupings, with the hope of better predicting voter behaviour overall. We are trying the analyse the given data and visualize it. We are also trying to predict the outcome of the senate Election 2022 in each state by using Metropolis-Hastings-within-Gibbs and Markov-chain Monte Carlo sampling.

Data:

The data on electoral outcomes by state over the last ten election cycles was obtained from the Office of the Federal Register website. For two primary reasons, this research exclusively looks at the last ten elections. One reason is that the two major parties have grown greatly since the Civil Rights era, and hence going back further than the mid-1970s may be deceptive. Second, and more importantly, the current primary system did not emerge fully formed until the early 1970s. This fact makes our current election considerably more similar, at least procedurally, to ones held after 1976 than to those held before the 1970s. The same page tells you how many electoral votes you have.

Google Consumer surveys made public by Google were utilised in this project. These polls were chosen primarily because they are conducted at regular intervals (approximately once per week) in every state at the same time, and they also include information useful to determining the respondent's position as a probable voter. This research made no use of national polls. This paper may be improved by adding a broader range of polls, particularly national polls.

The dimensionality of the dataset is 381900 rows with 8 columns. The following is the example of the columns that we have and the respective information.

'data.frame': 381900 obs. of 8 variables:
$ Date : chr "08-10-2022" "08-10-2022" "08-10-2022" "08-10-2022" ...
$ Geography : chr "US-FL" "US-VA" "US-TX" "US-NC" ...
$ Initial.Weight : num 0.801 0.711 1.855 0.743 2.014 ...
$ Weight : num 0.823 0.73 1.905 0.763 2.069 ...
$ Question_1 : chr "100% likely" "100% likely" "100% likely" "100% likely" ...
$ Question_2 : chr "Undecided" "Democratic" "Democratic" "Undecided" ...
$ Question_3 : chr "Male" "Male" "Male" "Male" ...
$ Question_4 : chr "55-64" "55-64" "25-34" "65+" ...

2. Data Analysis

Gender Based Analysis:

We are trying to analyse some key factors that can be of great importance in the elections. Once such feature is the Question_3 that is asked in the survey. We will try to analyse how Male population and the female population reacted in the survey. We will try to plot some histograms and analyse how male and female population reacted.

Of the total male voters 58099 voters chose Democratic Party and 69058 voters chose Republican Party, 70379 voters remained undecided and around 3600 voters chose other candidates.

Figure1. depicts the histogram for the choice of the male voters.

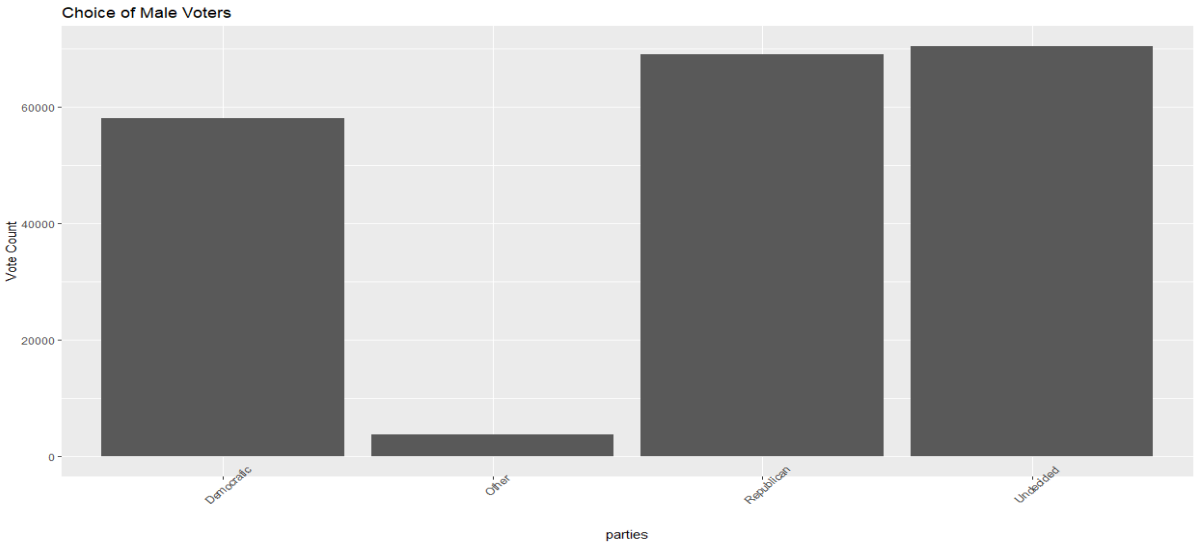


Fig1. Choice of Male Voters

In contrast to the male voters, female voters thought that democrats are more likely to win which is shown in Fig2.

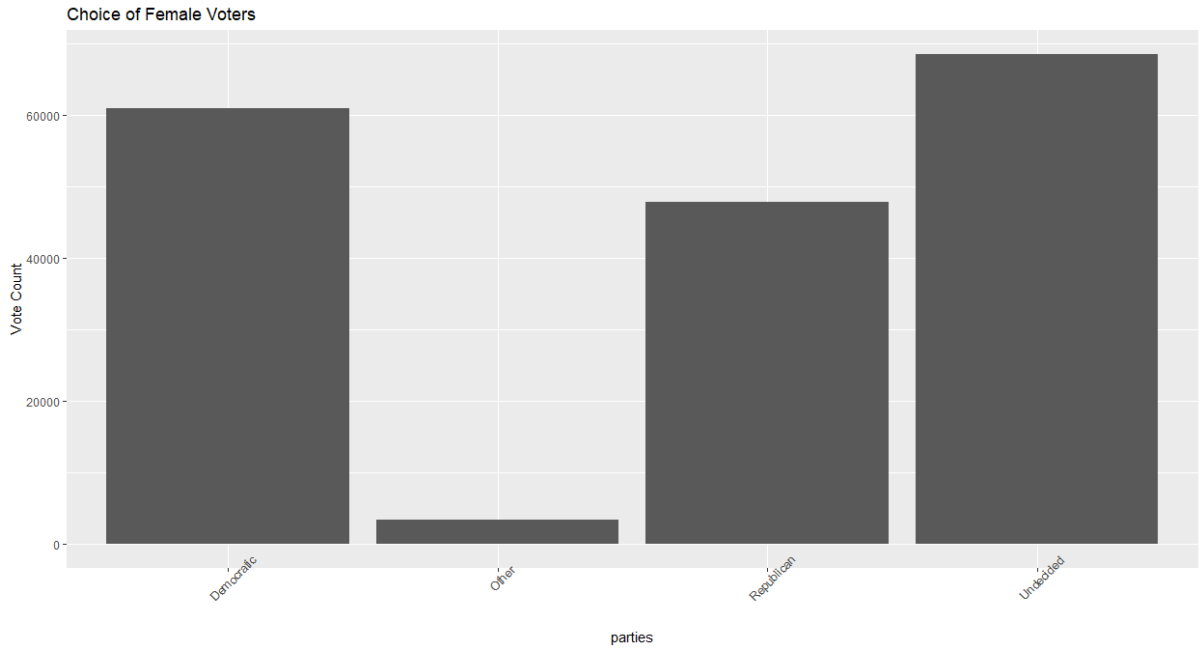
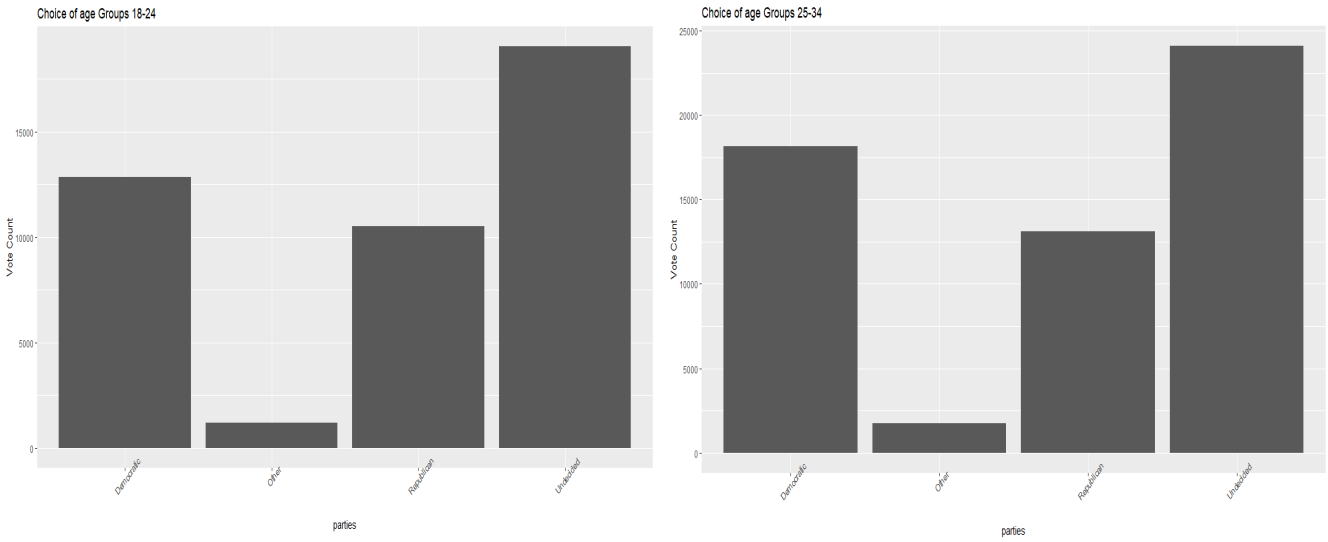


Fig 2. Choice of Female Voters

We can see that in both the cases of male and female voters, there are very high numbers of undecided voters. We will see if this becomes in interesting factor going forward.

Age Based Analysis:

There are total of 6 age groups those who participated in the survey. They are ages 18-24, 24-34, 35-44, 45-54, 55-64, 65+. We will now try to analyse which age group inclined to which party. We will try analysing this by using histograms.



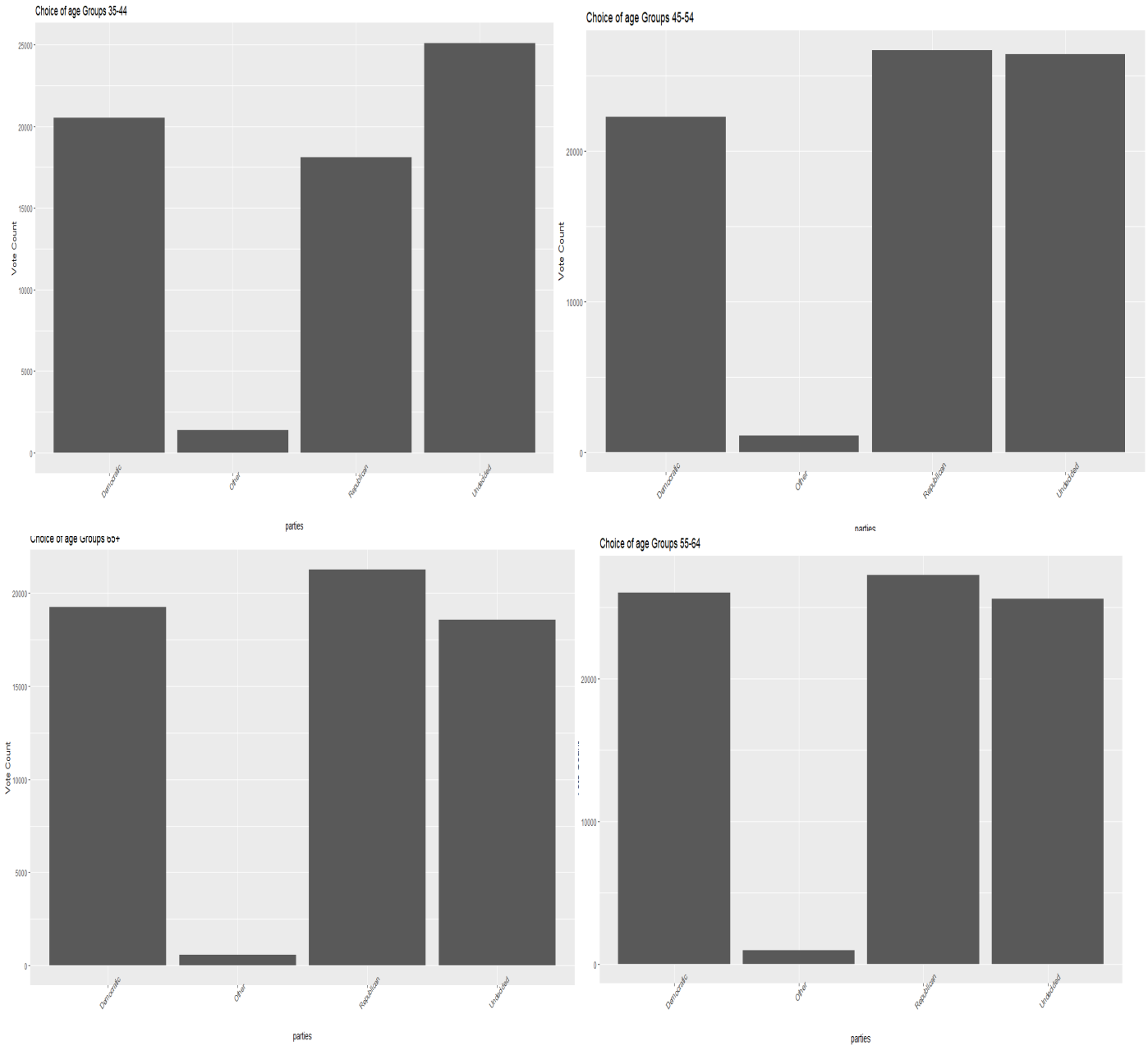


Fig3. Graphical Representation of Age wise choices

From the above histograms, we can interpret that the younger age groups inclined to the Democrats while the older population were inclined to the Republicans. Does this show the thought process difference between the age groups? This might be interesting topic to work on.

3. Methodology

The election history of each state I (including the District of Columbia) is utilized to produce an informative beta prior for p_i, the Bernoulli success probability of Clinton victory in that state. Each such p_i 's initial previous is set to beta (2,2). The hyperparameters (shape, rate) = (2,2) are chosen to show that p_i is unlikely to be very close to 0 or 1. Each prior election in state I (during the last 40 years) is viewed as a Bernoulli(p_i) trial (where Democratic victory is coded as success). As a result, the total number of Democratic victories in presidential elections in state I is a Binomial (10, p_i). Where E_i is the

number of Democratic victories in state i, then, we have that the informative prior for this election takes pi to be distributed as a beta($E_i + 2,\ 10 - E_i + 2$); i.e., beta($E_i + 2,\ 12 - E_i$).

The j^{th} poll in state i from this election is treated as a realization from a binomial(n_{ij} , p_i), where n_{ij} is the number of polled likely voters in the j^{th} poll of state i. To accommodate both shifting voter preferences during the election cycle and possible differences between polling responses and election behaviour, a power prior is used. Specifically: the election itself is treated as the final "poll". For $j = 1$, the posterior distribution of pi given the j^{th} poll relies on a power prior based on the $j - 1^{st}$ poll. Since the posterior of pi given the $j - 1^{st}$ poll itself relies on a power prior based on the $j - s^{nd}$ poll (for $j > 2$), the result is that the posterior distribution of pi given the final poll (just prior to the election) implicitly is influenced by all previous polls by way of decreasing power prior exponents. Thus, polls closer to the election have a proportionally greater impact on the distribution of pi than do earlier polls. The power prior exponent a_0 is treated as unknown with a uniform(0,1) prior.

Definitions:	
E_i	Total number of Democratic victories in last 40 years in state i
m	Total number of polls
p_i	Bernoulli probability of Democratic victory in state i
p	Vector of p_i for $i = 1, \cdots, 51$
D_{ij}	Ordered pair (n_{ij}, Y_{ij})
n_{ij}	Total poll responses from poll j in state i
Y_{ij}	Total pro-Clinton responses from poll j in state i
D_i	Matrix containing D_{ij} for $j = 1, \cdots, 10$
D	Matrix containing D_i for $i = 1, \cdots, 51$
a_0	Power prior exponent for effect of polls on p_i

Informative prior:

$$\pi(p_i|E_i) \sim \text{beta}(2 + E_i, 12 - E_i)$$

Posterior:

$$\pi(p_i|D_{im}) \sim \text{beta}\left(2 + E_i + \sum_{k=1}^{m} Y_{ik} \cdot a_0^{m-k},\ 12 - E_i + \sum_{k=1}^{m} (n_{ik} - Y_{ik}) a_0^{m-k}\right)$$

The reason that Metropolis-Hastings (rather than Metropolis) is required here is that the proposal distribution f is not symmetric. Although the normal distribution is symmetric, we are dealing with a normal distribution among logit transforms of a_0, so symmetry is lost. The ratio of the two proposal pdfs equals the ratio of the logit transform's derivative; that is:

$$\begin{aligned} \frac{f(a_0|a_0^*)}{f(a_0^*|a_0)} &= \frac{1/[a_0(1-a_0)]}{1/[a_0^*(1-a_0^*)]} \\ &= \frac{a_0^*(1-a_0^*)}{a_0(1-a_0)} \end{aligned}$$

Thus, we have:

$$a = \min\left\{1, \frac{\prod_{i=1}^{51} \pi(p_i|D_i, a_0^*) \cdot a_0^*(1-a_0^*)}{\prod_{i=1}^{51} \pi(p_i|D_i, a_0) \cdot a_0(1-a_0)}\right\}$$

The resulting MCMC algorithm for drawing M samples is as follows:

For $k = 1, \cdots, M$:

1. Use metropolis-Hastings.

2. Use posterior distribution.
3. Draw Bernoulli to the election results for each state.
4. Use information on electoral votes for each state to determine national election result.

4. Predictions

Only likely voters were examined in this project. Likely voters were defined as those who said they were "100% likely" or "Extremely Likely" they would vote in the election. Third-party and undecided voters were not considered, so this study only includes decided, two-party probable voters.

An updated version of this research would seek to model undecided voters' behaviour as well as (perhaps less importantly) third-party voters' behaviour. It would also be possible to create a more refined approach of identifying likely voters than the one employed here.

This analysis' model and data predict that Democrats will win the election. This is based on M = 100,000 samples, with 5000 samples removed as burn-in.

However, the model does not heavily favour Democrats. The fraction of the samples drawn in which they win is approximately 0.51149. The results for each state are shown below. The winner is determined by calculating the mean of the p_i samples. This is shown beside $\alpha = 0.1$ HPD interval for each p_i. (Recall that p_i is the likelihood of Democrats winning state i.) A swing state is one in which 0.5 falls inside the HPD interval for a given state.

To validate this methodology, look at the autocorrelations and effective sample sizes of its national election projection, estimate of p_i for each I and estimate of a_0. To begin with, the binary predictor of national election outcome has almost little autocorrelation: see Fig4. As a result, it has a full effective sample size of 95,000 out of a total sample size of 100,000, with 5,000 removed as burn-in.

The autocorrelation of the p_i is similar; there is very little autocorrelation for any of these, and the effective sample size is near to or equal to 95,000 in all cases. The lowest effective sample size is for Alaska, at 72679.54, which is still quite high enough for our purposes.

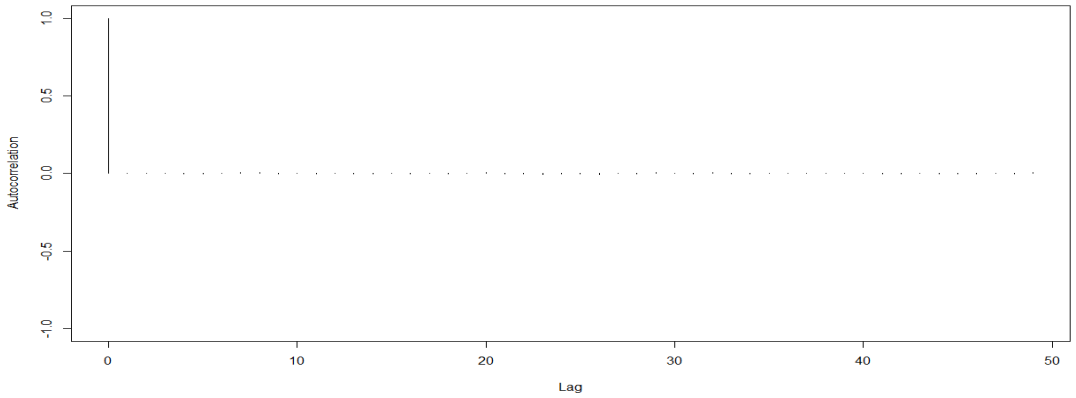


Fig4. Autocorrelation of binary national election outcome

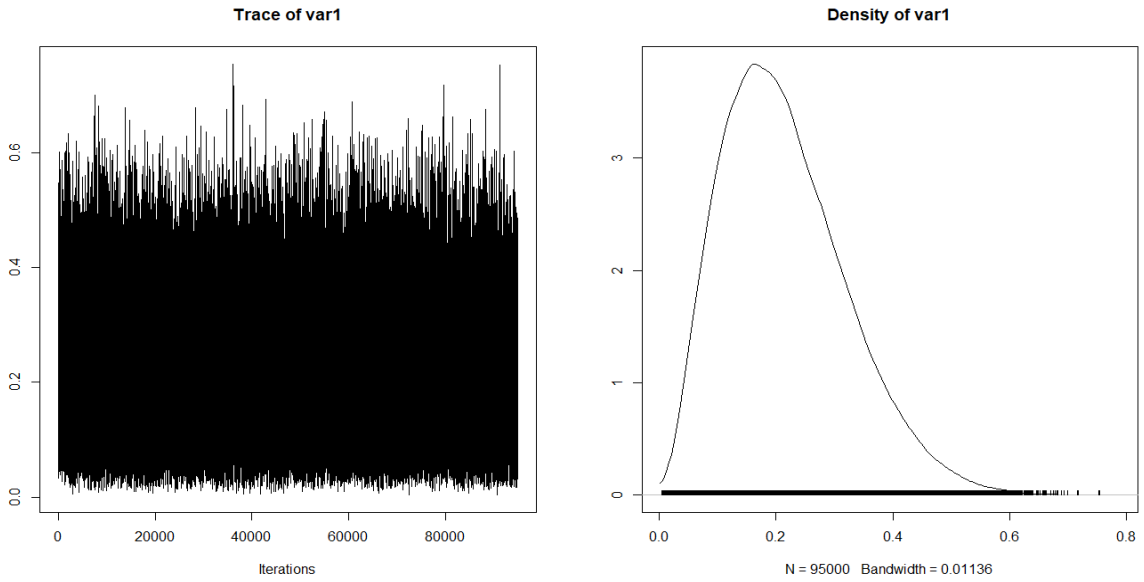


Fig5: Plot and variance of the samples of p_i where I is Texas

We can examine the graphs and plots of each of the pi and their autocorrelation. For our reference we have depicted few states' graphs in the Fig 5 and Fig 6 respectively.

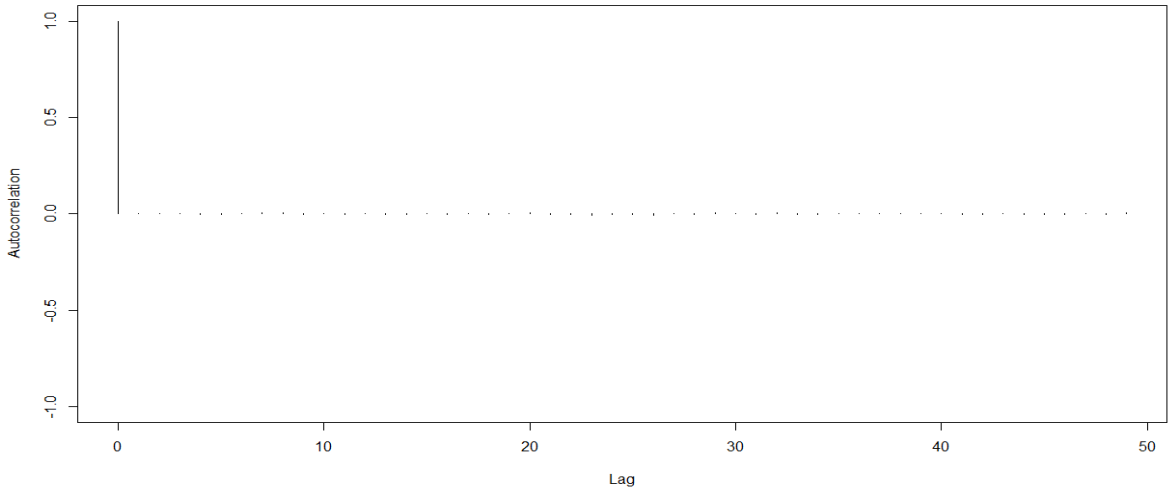


Fig6: Autocorrelation of p_i, where i is California.

5. Results:

As told before, we have predicted the states in which the parties are going to wing. We are also able to get the info about the swing states. Our model predicted that the democrats will win the senate elections with a probability of around 0.516. This is really a close call.

The following is the information about the predictions that were made in each state along with the information about the swing states.

STATE	VICTORY	p_i	SWING	HPD	INTERVAL
[1] US-AK	Republican	0.143		0.009	0.272
[1] US-AL	Republican	0.214		0.045	0.375
[1] US-AR	Republican	0.358	Swing	0.153	0.557
[1] US-AZ	Democratic	0.214		0.044	0.374
[1] US-CA	Democratic	0.571		0.362	0.382
[1] US-CO	Democratic	0.357	Swing	0.149	0.552
[1] US-CT	Democratic	0.571		0.361	0.282
[1] US-DC	Democratic	0.857		0.726	0.490
[1] US-DE	Democratic	0.643		0.443	0.344
[1] US-FL	Republican	0.428		0.213	0.234
[1] US-GA	Republican	0.357		0.153	0.157
[1] US-HI	Democratic	0.786		0.628	0.159
[1] US-IA	Democratic	0.571	Swing	0.367	0.787
[1] US-ID	Republican	0.143		0.008	0.272
[1] US-IL	Democratic	0.571	Swing	0.359	0.777
[1] US-IN	Republican	0.214		0.047	0.377
[1] US-KS	Democratic	0.143		0.010	0.272
[1] US-KY	Republican	0.358		0.150	0.354
[1] US-LA	Republican	0.357		0.151	0.254
[1] US-MA	Democratic	0.714		0.527	0.401
[1] US-MD	Democratic	0.715		0.531	0.406
[1] US-ME	Democratic	0.572	Swing	0.368	0.787
[1] US-MI	Democratic	0.571	Swing	0.359	0.779
[1] US-MN	Democratic	0.857		0.726	0.290
[1] US-MO	Republican	0.357	Swing	0.155	0.560
[1] US-MS	Republican	0.214		0.048	0.378
[1] US-MT	Republican	0.215		0.047	0.378
[1] US-NC	Republican	0.286		0.096	0.471
[1] US-ND	Republican	0.143		0.007	0.272
[1] US-NE	Republican	0.143		0.009	0.273
[1] US-NH	Republican	0.500		0.287	0.413
[1] US-NJ	Democratic	0.572		0.367	0.389
[1] US-NM	Democratic	0.500		0.285	0.412
[1] US-NV	Democratic	0.429		0.216	0.237
[1] US-NY	Democratic	0.714		0.529	0.105
[1] US-OH	Republican	0.499	Swing	0.283	0.710
[1] US-OK	Republican	0.143		0.009	0.272
[1] US-OR	Democratic	0.643	Swing	0.444	0.849
[1] US-PA	Republican	0.642	Swing	0.446	0.848
[1] US-RI	Democratic	0.786		0.623	0.354
[1] US-SC	Republican	0.215		0.048	0.378
[1] US-SD	Republican	0.143		0.010	0.272
[1] US-TN	Republican	0.356	Swing	0.152	0.555
[1] US-TX	Republican	0.215		0.045	0.376
[1] US-UT	Republican	0.143		0.008	0.272
[1] US-VA	Democratic	0.286		0.095	0.470
[1] US-VT	Democratic	0.572	Swing	0.365	0.785
[1] US-WA	Democratic	0.643		0.444	0.246

[1] US-WI	Democratic	0.714		0.530	0.202
[1] US-WV	Republican	0.500	Swing	0.285	0.710
[1] US-WY	Republican	0.143		0.010	0.273

We have also calculated auto correlation of binary national election outcome and the Bernoulli probability shown in the graphs before.

Finally, when we look at a_0, we discover that it has far more autocorrelation than any p_i, but not by a significant amount. Our M = 100,000 draws yielded an effective sample size of 9072.23. Fig 7 depicts the autocorrelation.

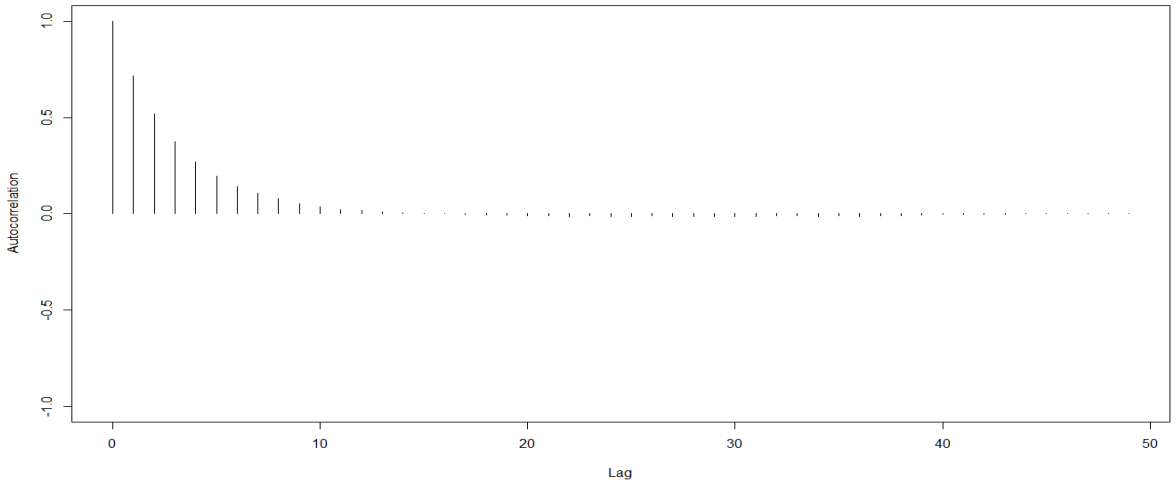


Fig7. Autocorrelation of a_0

The average value of a_0 is around 0.4998348. α = 0.05 HPD interval is provided below.

HPDinterval(a0.mcmc,prob=0.95)

	lower	upper
var1	0.4038384	0.5949509

attr(,"Probability")

[1] 0.95

The MCMC samples appear to be sufficiently uncorrelated to provide acceptable confidence in the analysis's conclusions.

6. Conclusions:

As per the data analysis that was done earlier, we can conclude that the greater number of male reporters were inclined to Republicans where the female voters were inclined to Democrats. Also, the age based data analysis shows us that the younger group of people chose democrats whereas the aged people showed interest towards Republicans. The results that we interpreted in the project were very close to the actual results. As of now Democrats hold the lead in the Senate with 51-49 with Georgia going for a run-off election. We were able to predict the winners of the senate very accurately with the probability of around 0.5 with the help of Metropolis-Hastings-within-Gibbs Markov-chain Monte Carlo sampling.

7. References:

1. 2022 Senate Election Survey from https://www.bloomberg.com/graphics/2022-us-election-results/senate/

2. 2022 Senate Election Forcast from https://projects.fivethirtyeight.com/2022-election-forecast/senate/ yEight

3. Monte Carlo Markov Chain (MCMC) from https://towardsdatascience.com/monte-carlo-markov-chain-mcmc-explained-94e3a6c8de11

4. Metropolis and Gibbs Sampling from https://people.duke.edu/~ccc14/sta-663-2018/notebooks/S10D_MCMC.html

5. Election Night 2022 | US Senate Elections (Prediction) from Election Night 2022 | US Senate Elections (Prediction) – U.S. Voters

Appendix:

R Code Implemented for the project

```
#####Senate Election Forecast 2022 Using Markov Chain Monte Carlo Sampling and Metropolis-Hastings#######
getwd()
setwd('/Users/91799/Downloads/Senate_election_forecast_2022-master/senate_election_forecast_2022-master')
rm(list=ls())

# Get data from csv
polls <- read.csv("all_polls.csv")
dim(polls)
str(polls)
#Get the summary of the data
summary(polls)

lvpolls <- polls[polls[,5]=='100% likely' | polls[,5]=='Extremely likely',]
dlvpolls <- lvpolls[lvpolls[,6]=='Democratic'|lvpolls[,6]=='Republican',]

library(ggplot2)
library(ggmap)
# Get unique dates

#FIlter by male voters

male_count <- filter(polls, Question_3 == 'Male')
male_count
library(dplyr)

df_male <- data.frame(table(male_count$Question_2))
df_male <- df_male %>% slice(-c(1))
df_male
plot(df_male$Freq, main = "Choice of MLmale Voters", xlab = "parties", ylab = "votes", type = "o")

ggplot(aes(x = Var1, y = Freq), data = df_male) + geom_bar(stat = 'identity') +
```

```
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of Male Voters')


#FIlter by male voters

female_count <- filter(polls, Question_3 == 'Female')
female_count

df_female <- data.frame(table(female_count$Question_2))
df_female <- df_female %>% slice(-c(1))
df_female
ggplot(aes(x = Var1, y = Freq), data = df_female) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of Female Voters')

#Filter by Age Groups 18-24
age_1 <- filter(polls, Question_4 == '18-24')
age_1 <- data.frame(table(age_1$Question_2))
age_1 <- age_1%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_1) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 18-24')

#Filter by Age Groups 25-34
age_2 <- filter(polls, Question_4 == '25-34')
age_2 <- data.frame(table(age_2$Question_2))
age_2 <- age_2%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_2) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 25-34')

#Filter by Age Groups 35-44
age_3 <- filter(polls, Question_4 == '35-44')
age_3 <- data.frame(table(age_3$Question_2))
age_3 <- age_3%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_3) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
```

```
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 35-44')

#Filter by Age Groups 65+
age_6 <- filter(polls, Question_4 == '65+')
age_6 <- data.frame(table(age_6$Question_2))
age_6 <- age_6%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_6) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 65+')


#Filter by Age Groups 45-54
age_4 <- filter(polls, Question_4 == '45-54')
age_4 <- data.frame(table(age_4$Question_2))
age_4
age_4 <- age_4%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_4) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 45-54')

#Filter by Age Groups 55-64
age_5 <- filter(polls, Question_4 == '55-64')
age_5 <- data.frame(table(age_5$Question_2))
age_5
age_5 <- age_5%>% slice(-c(1))

ggplot(aes(x = Var1, y = Freq), data = age_5) + geom_bar(stat = 'identity') +
  theme(axis.text.x = element_text(angle = 45)) +
  xlab('parties') +
  ylab('Vote Count') +
  ggtitle('Choice of age Groups 55-64')


dates=unique(polls[,1])
npolls=length(dates) # number of distinct polls

#Get the Info about Unique Values
gender <- unique(polls[["Question_3"]])
table(polls$Question_3)

#Get the info about the age groups
age <- unique(polls["Question_4"])
table(polls$Question_4)
```

```
#In total there are 6 age groups starting from 18 to 65+

#Likeliness
likes <- unique(polls["Question_1"])
table(polls$Question_1)
#Based on the likeliness the weightage is assigned by the survey

#parties
party <- unique(polls["Question_2"])
table(polls$Question_2)
#Accoriding to the above data people have been classified into 4 groups Democratic, Republican,
Undecided, Other


# Get unique states, alphabetical
states = sort(unique(polls[,2]))
nstates = length(states) # number of states + DC

# Make vector with number of electoral votes in each state
evotes = c(3,9,6,11,55,9,7,3,3,29,16,4,6,4,20,11,6,8,8,
          11,10,4,16,10,10,6,3,15,3,5,4,14,5,6,29,18,7,
          7,20,4,9,3,11,38,6,13,3,12,10,5,3)

# Get outcome of polls
nn = matrix(-99,nstates,npolls) # Record total decided likely voters here
outcomes = matrix(-99,nstates,npolls) # Record successes here
for ( ii in 1:length(states) ) {
  for ( jj in 1:length(dates) ) {
    n = sum(dlvpolls$Date == dates[jj] & dlvpolls$Geography == states[ii])
    y = sum(dlvpolls$Date == dates[jj] & dlvpolls$Geography == states[ii] &
              dlvpolls$Question..2.Answer == 'Democratic')
    nn[ii,jj] = n
    outcomes[ii,jj] = y
  }
}

# Define prior on p_i
p.hypers <- matrix(2,nstates,2) # column 1 is shape, col 2 is rate for beta dist
prev_elections_table <-
  read.csv("modern_results_by_state.csv")
npe <- dim(prev_elections_table)[2] -1 # Number of previous elections
prev_elections <- prev_elections_table[,2:(npe+1)]
# dem_rep_vics will hold number of dem and rep victories in previous elections
dem_rep_vics <- matrix(-99,nstates,2)
for ( i in 1:nstates ) {
  dem_rep_vics[i,] = c(sum(prev_elections[i,]), npe-sum(prev_elections[i,]) )
}
#repvics = rep(10,nstates) - demvics
p.hypers = p.hypers + dem_rep_vics
```

```
# MCMC settings
M = 1e5 # Total number of samples (including burn-in)
burn_in = 5e3 # floor(M/3)

# MCMC loop function
forecast <- function(a0,
                     M,
                     burnIn = 0,
                     p=rep(0.5,51),
                     ns = nn,
                     poll.outcomes = outcomes,
                     e.votes = evotes,
                     p.hyper = p.hypers,
                     sigma = 1) {

  # Get some initial settings
  alpha0  = p.hyper[,1]
  beta0   = p.hyper[,2]
  nstates = dim(poll.outcomes)[1]
  a0.vec  = rep(a0,npolls)^seq(npolls-1,0) # Get vector of decreasing powers of a0
  g0      = log(a0/(1-a0)) # logit transform to eliminate boundary constraints
  # Get parameters for initial beta draw
  alpha   = alpha0 + poll.outcomes %*% a0.vec
  beta    = beta0 + (ns - poll.outcomes) %*% a0.vec

  # Set up vehicles for records:
  a0.rec = rep(-99,M)
  accept.rec = rep(-99,M)
  r.rec = rep(-99,M)
  p.rec = matrix(-99,M,nstates)
  state.results.rec = matrix(-99,M,nstates)
  forecast.rec = rep(-99,M)

  for (ii in 1:M) {

    ### MH step to get a0
    g0.s = rnorm(1,g0,sigma) # Draw new g0 candidate
    a0.s = exp(g0.s)/(1+exp(g0.s)) # Reverse logit trans
    a0.s.vec = rep(a0.s,npolls)^seq(npolls-1,0)
    alpha.s = alpha0 + poll.outcomes %*% a0.s.vec
    beta.s  = beta0 + (ns - poll.outcomes) %*% a0.s.vec

    mh.lnum <- sum( dbeta(p, alpha.s, beta.s, log = T) + log(a0.s*(1-a0.s)) ) # log of numerator of MH
acceptance ratio
    mh.lden <- sum( dbeta(p, alpha, beta, log = T) + log(a0*(1-a0)) ) # log of denominator of MH
acceptance ratio
    r = exp(mh.lnum - mh.lden) # acceptance ratio

    accept = 0 # logical to tell us whether new candidate accepted
    if (runif(1) < r) {
```

```
      a0 = a0.s
      alpha = alpha.s
      beta = beta.s
      g0 = g0.s
      a0.vec = a0.s.vec
      accept = 1
    }

    ### Draw p_i from conditional dist for each state
    p = rbeta(nstates, alpha, beta)

    ### Generate state outcomes
    state.results = rbinom(nstates,1,p)

    ### Generate election outcome
    forecast=0
    hrc.evotes = state.results %*% e.votes
    if (hrc.evotes >= 270) {forecast = 1} # Democratic victory

    ### Record things
    a0.rec[ii] = a0
    accept.rec[ii] = accept
    r.rec[ii] = r
    p.rec[ii,] = p
    state.results.rec[ii,] = state.results
    forecast.rec[ii] = forecast

    ### Get periodic update
    if (ii %% 1000 == 0) {
      par(mfrow=c(2,2))
      plot(a0.rec[1:ii])
      plot(p.rec[1:ii,10])
      plot(state.results.rec[(ii-100):ii,40])
      plot(forecast.rec[(ii-100):ii])
    }
  }

  nonBurnIn = M - burnIn
  return(list("forecasts"       = tail(forecast.rec,nonBurnIn),
              "state.forecasts" = tail(p.rec,nonBurnIn),
              "a0"              = tail(a0.rec,nonBurnIn),
              "accept"          = tail(accept.rec,nonBurnIn),
              "state.results"   = tail(state.results.rec,nonBurnIn)))

}

# Run MCMC and get prediction
res <- forecast(.9,M,burn_in)
prediction <- sum(res$forecasts)/length(res$forecasts)
prediction # This is the estimated probability of Democratic victory
```

```
library(coda)

# Get results for each state
state.predictions <- apply(res$state.forecasts,2,mean)
pred.mcmc = as.mcmc(res$state.forecasts) # Coerce the vector into a MCMC object
# pred.mcmc.hpd gives a HPD interval for the probability of each state
# going for Democratic.
pred.mcmc.hpd = round(HPDinterval(pred.mcmc, prob = 0.9),3)                # Find 95% HPD interval
for theta using the CODA function
# The following loop makes a table with one line per state, along with
# expected victor in that state, estimated prob. of Democratic victory,
# and .9 HPD interval of Democratic victory. In addition, if a state's
# .9 HPD interval inclues .5, then that state is labelled as a
# swing state.
for (ii in 1:nstates) {
  safety = '  Swing  '
  if (0.5 > pred.mcmc.hpd[ii,2]) {safety = '         '}
  if (0.5 < pred.mcmc.hpd[ii,1]) {safety = '         '}
  winner = '  Democratic  '
  if (state.predictions[ii]<0.5) {winner = '  Republican    '}
  print( paste( states[ii],
                winner,
                formatC(state.predictions[ii],format='f',digits=3),
                safety,
                formatC(pred.mcmc.hpd[ii,1],format='f',digits=3),
                formatC(pred.mcmc.hpd[ii,2],format='f',digits=3)),
         quote=F)
}

par(mfrow=c(1,1))

## Examine senate prediction
pres.mcmc = as.mcmc(res$forecasts)
#Check autocorrelation of senate prediction
autocorr.plot(pres.mcmc)
# Check effective sample size of senate prediction
effectiveSize(pres.mcmc)

## Examine state predictions
plot(pred.mcmc, ask = FALSE)
# Check autocorrelation of state p_i
autocorr.plot(pred.mcmc, ask = FALSE)
# Check effective sample size of state p_i
effectiveSize(pred.mcmc)
# Now let's do that for some particular states:
plot(pred.mcmc[,41])
# California Data
autocorr.plot(pred.mcmc[,5])
```

```
# Florida Data
effectiveSize(pred.mcmc[,10])

## Examine a0
a0.mcmc = as.mcmc(res$a0)
# Check autocorrelation of a0
autocorr.plot(a0.mcmc)
# Check effective sample size of a0
effectiveSize(a0.mcmc)
mean(a0.mcmc)
HPDinterval(a0.mcmc,prob=0.95)

plot(res$a0, typ = 'l')
plot(res$state.forecasts[,10], typ = 'l')
```