Donald Trump Approval Ratings

Bella Shao
s2876426

June 2021

1 Introduction

The political election has always been a hot topic all around the world, especially during the election seasons. The one essential aspect for predicting which presidential candidates would win the election is to trace their satisfaction rates from populations. One popular method is to predict their approval rates based on all kinds of polls. In this report, we will apply a Bayesian MCMC simulation to predict Donald Trump's approval rates based on a data set from FiveThirtyEight.

The methodology used by FiveThirtyEight is a local polynomial regression. But we will try a different method which involves Bayesian MCMC simulations. Technically, we first decide the likelihood probability distribution base on the characteristics of the data. In our case, this is multinomial distribution. Secondly, we come up with a proper prior which,in the best case scenario,can be conjugate to the likelihood. Furthermore, a hyper prior distribution can be assigned to the parameter of prior distribution for the reason that we assume the data is randomly drawn from a population.

The data set we will use is extracted within the time period from 18/05/2020 to 24/05/2020, which consists of 37 rows. There are some characteristics about the data from FiveThirtyEight. They use almost all polls and wight them with their historical accuracy, which is the "grade". They also adjust for the house effects for each polls as well as accounting for all kinds of uncertainties with a wide range for predictions. Their sample sizes for each poll constitute votes from almost all kinds of voters, such as adults, likely-voters and registered-voters.

2 Data preparation

In this section, we make our data set ready for model building and analysis.

As to our small data set, the variables we will use are "startdate", "enddate", "samplesize", "approve" and "disapprove". Then we extract our sample data from the date of 18/05/2020 to 24/05/2020. In order to calculate the number of counts for "approve" and "disapprove" categories, we firstly change the percentages of both of the variables into decimal points and then multiply them with the "samplesize" of each pollster respectively. Because the percentages of "approve" and "disapprove" don't add up to 1, there are some voters who were neither approve nor disapprove with Donald Trump. Hence, we create a new category of "unsure" by subtracting the two categories from 1, which is 1 minus approve minus disapprove. Following the same method, we also calculate the counts from "unsure" category by multiplying the sample size. To this point, we have three new variables in our data set, which are "approve_counts", "disapprove_counts" and "unsure_counts".

3 Model building

Based on the previous section, we have three rating categories and their ratings sum up to 1 for each pollster. By plotting the number of counts for the three categories all together, Figure 1 shows that each category follows the similar trend and all the trend lines are distributed along with each other, which implies a potential multinomial distribution for the response variable.Thus we can propose a multinomial distribution as the probability distribution likelihood for our model. If we pool all the three

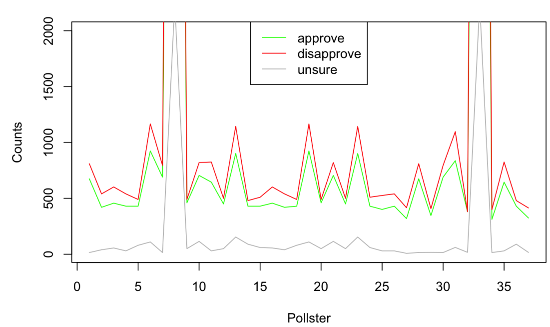


Figure 1: Counts distribution for the three categories

different categories together(Figure 2), we can still see the pattern of multinomial probability distribution for each category. Each category still maintains their own variance even we pooled them all together. The multinomial probability distribution

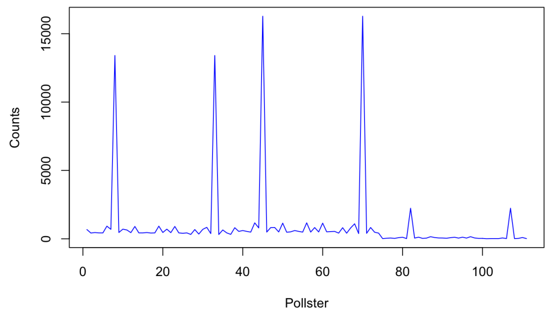


Figure 2: Pooled counts distribution

for each pollster is as follows:

$$y_n \sim multinomial(p_n),$$

where $n = 1, 2, \cdots, 37$ and $p_n = (p_{n1}, p_{n2}, p_{n3})$. Then the likelihood would be written as $L(y_n|p_n)$.

As we already decided the likelihood to be multinomial distribution, a Dirichlet probability distribution would be a good fit for the prior because they are conjugate. The Dirichlet distribution prior for the parameter $p_n$can be written as

$$p_n \sim dirichlet(\alpha),$$

where vector α is the parameter with the same length as the vector p_n.

After deciding the prior distribution, we focus on selecting the hyper prior for the Dirichlet distribution. Since we assume the pollsters in our data set are drawn from the population, the distribution for our hyper prior could be any distribution that are suitable. A gamma distribution could be a possible choice. The gamma distribution has two parameters and can be written as

$$\alpha \sim gamma(a, b),$$

where the values of the hyper parameters a and b are fixed and given by our choices and 3 values will be generated.

Our final model is written as follows

$$L(y_n|p_n)p(p_n|\alpha)p(\alpha|a,b),$$

where a and b are fixed values.

4 Model fitting by stan

In this part, we write our model into stan code and execute our model in R. The stan codes are as follows:

```
data {
  int<lower=0> N;      //the number of polls
  int<lower=0> K;      //the number of categories K = 3
  int<lower=0> m[N,K]; //N = 37 by K = 3 matrix


  real<lower=0> a; //hyperparameter for gamma
  real<lower=0> b; //hyperparameter for gamma
}


parameters {
  vector<lower=0>[K] alpha; //vector of length 3
  simplex[K] p[N]; //3 categories sum to 1
}


model {
  alpha ~ gamma(a,b);     //vector alpha consists of K = 3

  for (n in 1:N){
  p[n] ~ dirichlet(alpha);
  }  //prior

  for (l in 1:N){
    m[l,] ~ multinomial(p[l]); //likelihood
  }
```

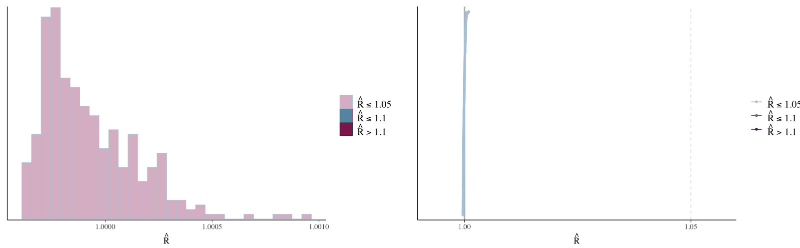
```
}

generated quantities{

  simplex[K] p_gnt[N];

  int <lower=0> y_gen[N,K];

  for (d in 1:N){
    p_gnt[d] = dirichlet_rng(alpha);}
    //generated ratings for each category


  for (s in 1:N){
    y_gen[s,] = multinomial_rng(p[s],100);}
    //generated new data from posterior p

}
```

In the block of generated quantities, since the prior is Dirichlet distribution and is conjugate to the Multinomial likelihood, thus the generated probabilities *p_gnt* is also from Dirichlet distribution with parameters of α. In addition, in order to check if our model converges and can predict well suited new data, we generate 100 data for each iteration by using the posterior probability of p, where we take the posterior uncertainties into account. Furthermore, we randomly choose the hyperparameters for *gamma* distribution of a and b to be 2 for both parameters.

5 Model diagnostics

After running the model from the previous section in R, in this part, we conduct model diagnostics based on the results.

Firstly we check the potential scale reduction factor Rhats from the results. Rhats measure the variances in each chain to the overall variances across all chains. In our model, both figures from Figure 3 show that all our rhats are around 1, thus all chains are at equilibrium.



(a) Histogram for Rhats (b) Rhats values

Figure 3: Rhats

Additionally, we check the effective sample size of our model. Effective sample size is the estimate independent draws from the estimated posterior distribution without autocorrelations. Figure 4 reveals that the majority of our effective sample size is bigger than 0.5, which indicates our posterior works well for the modeling.

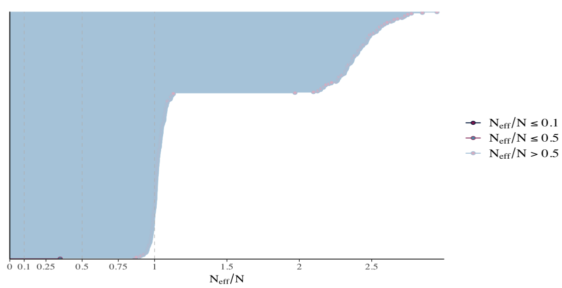


Figure 4: Effective sample size

We inspect some parameters randomly selected from p, p_gnt and α to check their autocorrelations. According to Figure 5, it is obvious that their autocorreal-tions immediately drop to zero or even below with increasing lags, which imply fast convergence of sample means to the true means. Moreover, we chose some random predicted posterior parameters and check their credible intervals.

Subsequently, we inspect the convergence of the No-U-Turn sampler for our model. NUTS-specific diagnostic values and log of the posterior density are extracted from the results. The NUTS(Figure 7) for each chains indicate the problem of divergence in the model is not a big concern. There are some diverged points at the tips of funnel-shaped graphs, but the majority of them are converged.

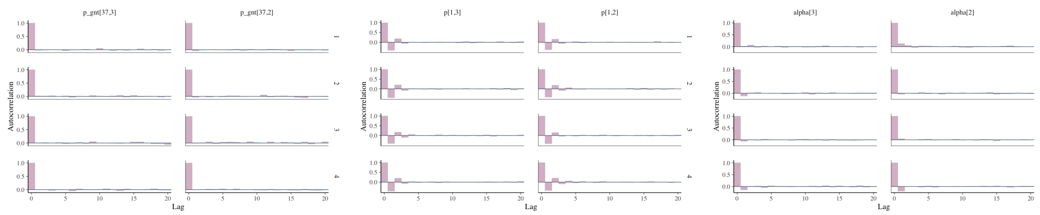


Figure 5: Autocorrelations

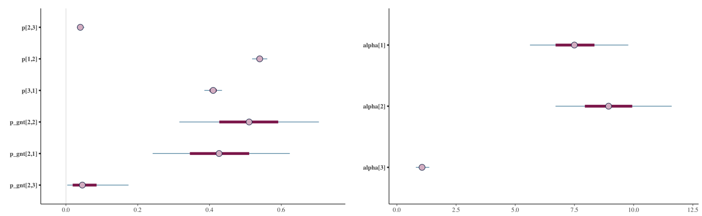


Figure 6: Credible intervals

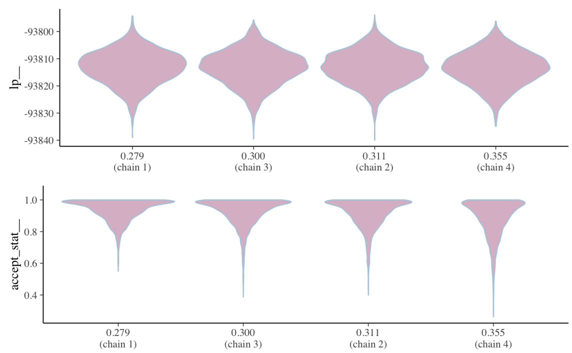


Figure 7: NUTS stepsize

To further checking the convergence of the model, we conduct pairwise comparisons from randomly selected parameters. We can check the "unsure" category under "p_gnt" interacting with other parameters. According to the paired plots(Figure 8), only the histogram of p_gnt under the "unsure" category skew to the right, all the other parameters distribute symmetrically around their means. In addition, the scatter plots of "p_gnt[37,3]" against other parameters indicate slow convergences in the funnel-like shaped scatter plots.

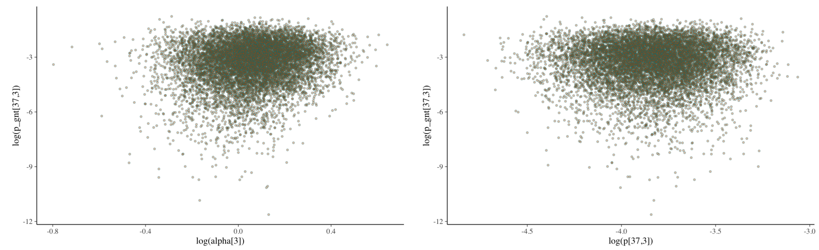


Figure 9: Log transformed parameters comparisons

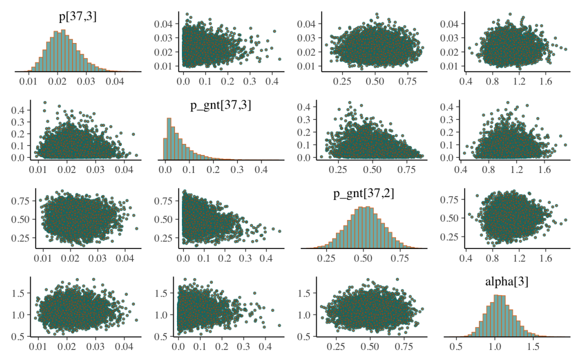


Figure 8: Pairwise comparisons

Thus we can investigate deeper into the parameter "p_gnt[37,3]" and the hierarchical parameter "alpha[3]"(we only investigate the "unsure" category).We transform both parameters into "unconstrained" just for allowing them to explore in the full space. As can be seen in this scatter plot(Figure 9), it shows a shape of Gaussian cloud, which indicates slow convergences with some over spread tails.This pattern can also be spotted between parameters p_gnt[37,3] and p[37,3].

Trace plots for parameters can also be treated as a good indicator for checking model convergence. For our model, the trace plots(Figure 10) from random selected parameters show good convergences except for "p_gnt[37,3]", which indicates lack of sampling. When we look deeper into "p_gnt[37,3]", we found out that this parameter is bounded by zero. However, since there is no divergence to plot, it might be because the random draws from Dirichlet posterior distribution are constrained by the multinomial likelihood which requires the probabilities of each category sum up to 1. Moreover, "unsure" is strongly correlated to the other two categories. "Approve" and "Disapprove" are independent random data from voters, however "unsure" is the sample size minus the other two categories. Thus this might also contribute to

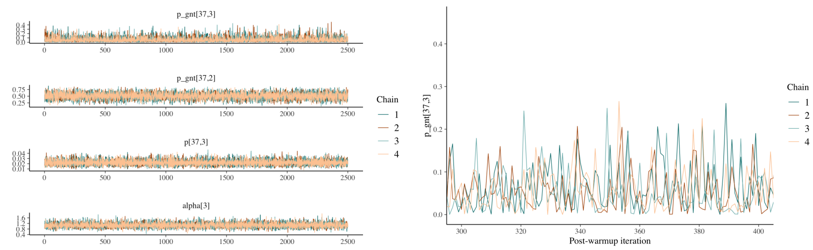


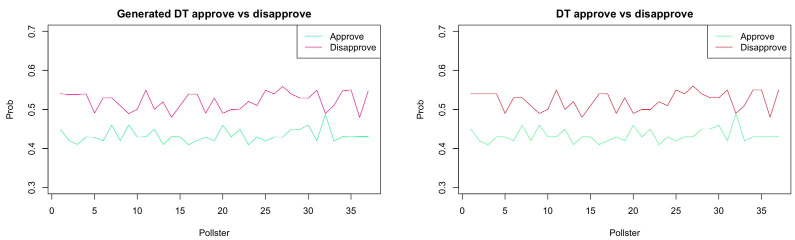
Figure 10: Trace plots

the undesirable predictions for "unsure" rates. Overall, our model converges well based on these diagnostics.

6 Model generations

Regarding to the previous model diagnostics, we know that divergence is not a big problem to concern for our model. To this end, we inspect the generated posterior quantities from our model.

As the prior is Dirichlet distribution with parameter α, the generated rates "*p_gnt*" for all the three categories are also follow the the same probability distribution with posterior α. Then we extract the mean rates for each pollster with three rating categories. Then the number of counts are calculated for each rating categories again by multiplying the generated rates "*p_gnt*" with the original sample size for each poll. The plot is shown in Figure 11. The trend line is similar as the counts from the original data set, which implies the chosen prior is conjugate well to the likelihood and the model converges.



(a) Generated ratings (b) Original ratings

Figure 12: Comparisons between generated ratings and original ratings

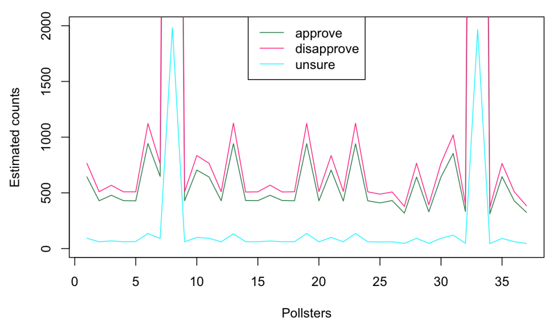


Figure 11: Estimated counts by p_gnt

Nevertheless, the conjugate prior is already expected at the beginning of our model constructions. To back check our model's generation quality, we generate new data by using the estimated posterior probabilities "p". The generated new data is "y_gnt". For each rating category, the generated data is summed up by polls. In other words, we generate votes for each category by each pollster. The sample size for each pollster is calculated by adding up all three rating categories' votes. After adding up all the data together, the total generated data size is 1000000. Hence, for each rating category, we calculate the estimated rates for each pollster by dividing the total data size. We make plots for both ratings in one graph and compare them with the rates in the original data set. The trend lines for both subplots in Figure 12 show similar patterns for each pollster.

Finally, we make the calibration plots for "Approve" and "Disapprove" category respectively to compare the predicted rates by generated data "y_gnt" with the rates in the original data. For the two categories, both calibration plots(Figure 13) show our predicted probabilities for each category are in line with the observed

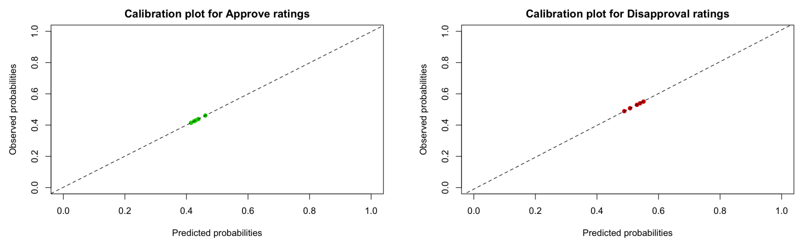


Figure 13: Calibration plots

probabilities.

7 Conclusion

In this report, we perform a MCMC simulation analysis for predicting the "Approve" and "Disapprove" rates for Donald Trump in 2020. In general, our model converges and predicts well. Since our model is a hierarchical model with a *gamma* hyper prior distributions, we also run the model with a *normal* hyperprior, this model converges as well but with much less effective sample size compared to *gamma* hyperprior. To this end, we could come up with a more calibrated hyperprior in the future analysis.

References

[1] Bob Carpenter. *Hierarchical Partial Pooling for Repeated Binary Trials.* https://mc-stan.org/users/documentation/case-studies/pool-binary-trials.html

[2] Bruno Nicenboim, Daniel Schad, and Shravan Vasishth. *An Introduction to Bayesian Data Analysis for Cognitive Analysis.* https://vasishth.github.io/bayescogsci/book/

[3] *Diagnostic plots for the No-U-Turn-Sampler (NUTS).* https://mc-stan.org/bayesplot/reference/MCMC-nuts.html

[4] *General MCMC diagnostics.* https://mc-stan.org/bayesplot/reference/MCMC-diagnostics.html

[5] Jonah Gabry and Martin Modrák. *Visual MCMC diagnostics using the bayesplot package* . https://mc-stan.org/bayesplot/articles/visual-mcmc-diagnostics.html

[6] Joe Blitzstein. *Statistics 110:Probability.* https://www.youtube.com/playlist?list=PL2SOU6wwxB0uwwH80KTQ6ht66KWxbzTIo

[7] Nate Silver. *How we're tracking Donald Trump's Approval ratings* https://fivethirtyeight.com/features/how-were-tracking-donald-trumps-approval-ratings/

[8] Shota Gugushvili. *Bayesian Statistics: Assignment.* https://brightspace.universiteitleiden.nl//content/enforced/16247-4433BAYSTY$_2021_W N/Assignment1_May2021.pdf?_{d2lSessionVal=fh5201K9qIOvw0eNoKp5bT7a4ou=16247}$

Appendices

We also run the model with normal hyper prior with mean equals to 0 and standard deviation equals to 1. We randomly selected some parameters and show their diagnostics plots below. The diagnostics plots all indicate that the model with normal hyper prior converges, but the effective sample sizes are less than the model with *gamma* hyper prior.

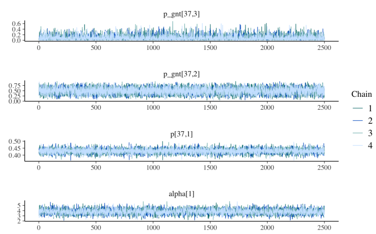


Figure 14: Trace plots

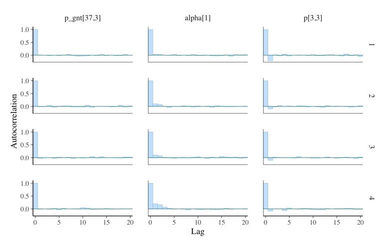


Figure 15: Autocorrelations

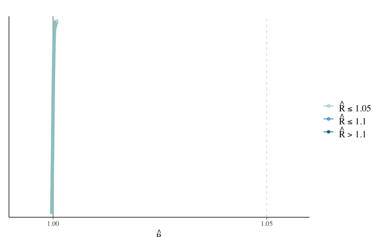


Figure 16: Rhats

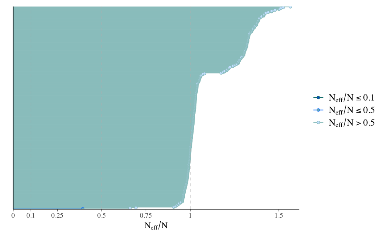


Figure 17: Effective sample sizes

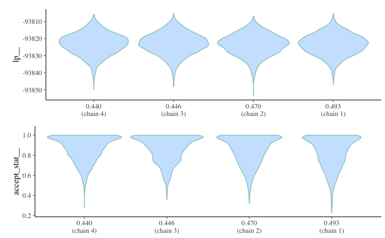


Figure 18: Divergence Check