STAT 4640/7640 **Homework 7**

Due: April 8, 2022

- **Instructions**: Make sure your name is on your paper and your answers are clearly written.

Early on in the pandemic, there were a lot of disparities between states both in terms of the number of positive cases and the number of tests being administered. The data“covid19.Rdata” conists of the number of positive tests of Covid-19 as well as the number of total tests administered for each of the 50 states between January 28, 2020 and March 17, 2020.

```
load("covid19.Rdata")
```

Let Y_i denote the number of positive tests for state i and N_i denote the number of total tests. We assume the following beloved model

$$Y_i \overset{indep}{\sim} Binomial(\theta_i, N_i)$$

where θ_i is the positive test rate for state i. Then, we specify the following prior and hyperprior distributions:

$$\theta_i \overset{indep}{\sim} Beta(\alpha, \beta)$$
$$\alpha \sim Gamma(10, 1)$$
$$\beta \sim Gamma(10, 1)$$

1. Write your own Metropolis-within-Gibbs sampling algorithm to fit the model. Note that each θ_i parameter can be updated using a Gibbs step (you know the closed-form full conditional for these and could easily rederive them using the process similar to that used in homework 6 for σ_i^2). Both of the hyperparameters, $\alpha > 0$ and $\beta > 0$, will require a Metropolis-Hastings algorithm. Use a truncated normal proposal distribution similar to that in homework 6 to ensure these parameters are positive. Make sure to tune your algorithm for good convergence. Obtain 10,000 posterior samples (post burn-in) from your model.

   (a) Make trace plots and marginal posterior densities of the parameters α, β, and $\theta_{Missouri}$.

   (b) Was there Bayesian learning for the parameters α and β? How do you know?

   (c) Report the posterior mean and 95% credible interval for α, β, and $\theta_{Missouri}$.

   (d) For Missouri and one other state of your choosing, create a boxplot of the posterior distributions $\theta_{Missouri}$ and $\theta_{OtherState}$. Comment on the comparison of the two positive test rates.

   (e) Conduct a hypothesis test where $H_0 : \theta_{Missouri} = \theta_{OtherState}$ vs $H_a : \theta_{Missouri} \neq \theta_{OtherState}$ and report your results.

2. Refit the model using JAGS. Run just 1 chain to start and obtain 10,000 samples from the posterior distriubion post burn-in. Make sure that your model estimates are the same as those you obtained from your sampling algorithm (but no need to report the new results).

3. Using the JAGS model output, we will assess convergence of the model.

   (a) To start, plot the autocorrelation of the chain for the parameters α, β, and $\theta_{Missouri}$ using the function

   ```
   autocorr.plot(samples[,j])
   ```

   where j corresponds to the parameter you want. Make sure to figure out the order of the parameters in the output of your JAGS model.

   (b) Compute the autocorrelation at lags 1, 5, 10 for $\theta_{Missouri}$ using the function

   ```
   autocorr(samples[,j],lag=L)
   ```

   where again, j is the parameter and L is the lag. You can give L as a vector if you want multiple lags at once.

   (c) Compute the effective sample size of the parameters α, β, and $\theta_{Missouri}$ using the function

   ```
   effectiveSize(samples[,j])
   ```

   and comment on what you find.

   (d) Compute the Geweke diagnostic for the parameters α, β, and $\theta_{Missouri}$ using the function

   ```
   geweke.diag(samples[,j],frac1=0.2,frac2=0.2)
   ```

   Here, frac1=0.2 and frac2=0.2 says to use the first 20% of the chains and the last 20% of the chains to do the two-sample test. Comment on what the results of this diagnostic means in terms of convernce of your MCMC.

   (e) Re-run the model with n.chains =3. Compute the Gelman-Rubin diagnostic using the function

   ```
   gelman.diag(samples)
   ```

   Report the Gelman-Rubin diagnostic for the parameters α, β, and $\theta_{Missouri}$, as well as the multivariate Gelman-Rubin statistic. Comment on what the results of this diagnostic means in terms of convernce of your MCMC.

4. You can use all of these diagnostics that are part of the coda package even when you don't use JAGS. All you have to do is turn your `samples` matrix that contains the chains of each of your parameters as columns into an `mcmc` object. That is,

   ```
   myMCMC=mcmc(samples,start=1001,end=11000)
   ```

   where here I assumed a burn-in of the first 1000 samples. Assess convergence of your Metropolis-within-Gibbs sampling algorithm that you did for question 1 and comment on any differences you detect between these results and those from the JAGS model fit (question 3a-e). You don't have to show your results, just write a few sentences of any differences/similarities you find.