Comparing the response time of younger and older age groups

With a Bayesian Hierarchical Model

Kimberly Schveder and Sonia Naeem

Introduction:

The COVID-19 pandemic is all the rage these days in the scientific and medical community. Taking a look at the COVID-19 data as of April 11, 2020, provided by the CDC about the population of the United States number of deaths of COVID, the flu, and all of the other causes by commonly known demographic information, such as age and sex, as well as by state.[2] Here, the analysis will focus on age groups, which are in 10-year intervals in the CDC data, and there are 11 age groups. The CDC warns older adult Americans that they need to take extra precautions because they are at higher risk. The CDC specifically warns the age group 65+ to take proper precautions and to have rigid social distancing with everyone as they are at the highest risk.[1]

Gibbs sampling simulation will be used to generate three posterior distribution of parameters from Bayesian hierarchical modeling based on death count from COVID-19 from younger and older people to predict the probability of older-aged people dying from COVID-19 more than younger people.

Explanations of the Dataset:

The decision of making the cut-off on making the age-group binary variable for the Monte Carlo Markov Chain (MCMC) algorithm is explained as follows. It could be fair to follow the CDC's older age group definition of 65+ year olds and the 0-64 year olds as the younger group. However, it didn't seem appropriate to have 44-64 year-olds in the same group as the younger and middle-aged groups, who tend to be healthier. Therefore, using the 44/45 age cut-off, a column of the binary age group of 0-44 year olds as "0", the younger/healthier group, vs. 45+ year olds as "1" older/unhealthier groups. The other variable that is used is COVID-19 death count.

Methods:

The MCMC algorithm uses the Bayesian theorem. Bayesian inference shows the relationship between two conditional probabilities that are the reverse of each other. This inference technique is based on Bayes theorem, which is based on the prior probability distribution of the unknown parameters. Such parameters are being estimated with the posterior probability. This probability takes into account the prior probability and the dataset. The parameter in Bayesian inference is treated like a random variable and uses the Bayesian theorem to estimate. If is the prior probability distribution for θ and $p(y|\theta)$ is the likelihood function combining the data and prior information, then Bayes' rule can be expressed as $p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$, where the posterior probability distribution is denoted as the $p(\theta|y)$. The $p(y)$ is a constant and is the data probability. The posterior probability will always be proportional to prior probability timing likelihood and thus it is under the same distribution family of the prior $p(\theta)$. The $p(\theta)$ is called conjugate prior.

The application of Markov Chain Monte Carlo (MCMC) has made the popularity of the Bayesian algorithm rise because the posterior probability density for the parameters being estimated can be calculated efficiently and accurately through the algorithm (Martin, A.D., Quinn, K.M. & Park, J.H., 2011). It is well known that a Markov chain is a sequence $x_1, x_2, \ldots$ of random elements of some set if the conditional distribution of x_{n+1} given $x_1, \ldots, x_n$ depends on x_n only (Geyer, C.J., 2011). Markov Chain Monte Carlo (MCMC) is a class of methods that simulates

draws samples from the data and uses previous sample values to randomly generate data to calculate the next sample value. This sampling method creates a Markov Chain and estimates the posterior distribution. Gibbs sampling a particular MCMC method and widely applicable in Bayesian analysis.

[1]https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/older-adults.html
[2]https://www.cdc.gov/nchs/nvss/vsrr/COVID19/

Gibbs sampling method starts out with deriving the posterior conditional for each of the random variables. Then, posterior samples from the target joint posterior are simulated by taking samples for a random variable from the corresponding posterior conditional probability. This occurs while the other variables are fixed. An example of Gibbs sampling is to take the posterior $p(\theta|y)$, where y is observed data and θ contains three parameters $\theta_1, \theta_2, \theta_3$ and following the 7 steps listed:

a. Define the set of full conditional distributions of $p(\theta_1|\theta_2, \theta_3, y)$, $p(\theta_2|\theta_1, \theta_3, y)$, and $p(\theta_3|\theta_1, \theta_2, y)$

b. Choose a vector for starting values $\theta^{(0)}$.

c. Start with any θ_1 (can also start with θ_2 or θ_3, order does not matter), then draw a value $\theta_1^{(1)}$ from the full conditional distribution$p(\theta_1|\theta_2^{(0)}, \theta_3^{(0)}, y)$

d. Use the value $\theta_1^{(1)}$, to draw a value $\theta_2^{(1)}$ (also can draw θ_3) from the full conditional distribution: $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, y)$

e. Using both updated value $\theta_1^{(1)}$ and $\theta_2^{(1)}$, draw a value $\theta_3^{(1)}$ from conditional distribution denoted as $p(\theta_3|\theta_1^{(1)}, \theta_2^{(1)}, y)$

f. Use$\theta^{(1)}$ to draw $\theta^{(2)}$to get the most updated values

g. Repeat until we get draws for vector θ with the expected number.

The Bayesian hierarchical model has multiple levels in it to eventually estimate all of the parameters and to describe any heterogeneity in the data, across and within several populations being compared. The structure of the hierarchy in the normal distribution model of the Bayesian hierarchical model is shown in Figure 1 below. In this graphical representation, observations {Y_1 ,…,Y_m } are independent and identical where Y_j~ N(θ_j, σ_j^2) , where{ (θ_1, σ_1^2), …, (θ_m, σ_m^2)} is within-group sampling distribution. {θ_1 , ..., θ_m} is a random variable with parameter (μ, τ^2) which is between-group heterogeneity in population means.

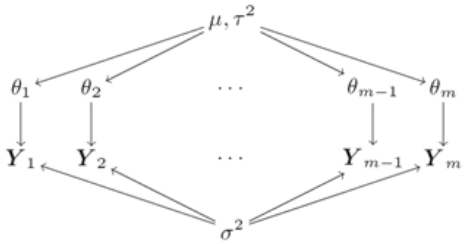

Figure 1: structure of the hierarchy in the normal distribution model of the Bayesian hierarchical model

There is a lack of information of the prior probability. Therefore, the MCMC samples were simulated possible and could not be the best approximation of the true distribution. The

convergence of the Markov chain is assessed with the use of some plots and statistics as a result.

Results and Analysis:

The boxplot of the number of COVID-19 deaths vs. age groups is shown in Figure 1 below. Notice how there is some overlap between the boxes. But, the medians and the means of the number of COVID-19 deaths in the two age groups are different. That is, as expected, the median line of the symmetric boxplot of the older age group is slightly higher than the median line of the symmetric boxplot of the younger age group. Both groups have a similar IQR and range, with the younger age group having a slightly higher IQR (700) and range (800) than the older age group (500 and 700, respectively), according to the boxplot in Figure 2.

Figure 2: boxplots of COVID-19 deaths by age group.

The younger and the older age groups are separately analyzed with bayesian hierarchical modeling. The posterior distribution of the mean will be represented for the death from COVID-19 for each of the groups. The number of deaths for each age group are both normally distributed according to the boxplots in Figure 2, with the same variance denoted as σ^2. In the model being computed, there are three parameters $\{\sigma^2, \mu, \tau^2\}$. The gamma family of distribution turns out to be a conjugate class of densities for the precision $1/\sigma^2$ (Hoff, P.D., 2009). Furthermore, the inverse gamma is a prior distribution for the variance parameter. This will lead to a proper posterior distribution calculation (Gelman, A., 2006).

Therefore, the Gibbs sampling procedure will in this context will be carried out in the following steps:

1. Specify three prior distributions:
   $p(\sigma^2) \sim Inverse\ Gamma(\nu_0/2, \nu_0\sigma_0^2/2)$
   $p(\mu) \sim Normal(\mu_0, \gamma_0^2)$
   $p(\tau^2) \sim Inverse\ Gamma\left(\frac{\eta_0}{2}, \frac{\eta_0\tau_0^2}{2}\right)$
2. Given a current state of the unknown $\{\theta_1^{(s)}, \dots, \theta_m^{(s)}, \mu^{(s)}, \tau^{2(s)}, \sigma^{2(s)}\}$, a new state is generated as:
   a. Sample $\mu^{(s+1)} \sim p\{\mu|\theta_1^{(s)}, \dots, \theta_m^{(s)}, \tau^{2(s)}\}$;
   b. Sample $\tau^{2(s+1)} \sim p\{\tau^2|\theta_1^{(s)}, \dots, \theta_m^{(s)}, \mu^{(s+1)}\}$;
   c. Sample $\sigma^{2(s+1)} \sim p\left\{\sigma^2\middle|\theta_1^{(s)}, \dots, \theta_m^{(s)}, y_1, \dots, y_m\right\}$;
   d. for each $j \in \{1, \dots, m\}$, sample $\theta_j^{(s+1)} \sim p\{\theta_j|\mu^{(s+1)}, \tau^{2(s+1)}, \sigma^{2(s+1)}, y_j\}$
3. After complicated calculation, each of the posterior distributions has full conditional distribution:
   $\{\theta_j|y_{1,j}, \dots, y_{n_j,j}, \sigma^2\} \sim Normal\ (\frac{n_j\bar{y}_j/\sigma^2 + 1/\tau^2}{n_j/\sigma^2 + 1/\tau^2}, [n_j/\sigma^2 + 1/\tau^2]^{-1})$

$$\{1/\sigma^2|\Theta, y_1, \ldots, y_m\} \sim Gamma\left(\frac{1}{2}\left[\upsilon_0 + \sum_{j=1}^{m} n_j\right], \frac{1}{2}\left[\upsilon_0\sigma_0^2 + \sum_{j=1}^{m}\sum_{i=1}^{n_j}(y_{i,j} - \theta_j)^2\right]\right)$$

$$\{\mu|\theta_1, \ldots, \theta_m, \tau^2\} \sim Normal\left(\frac{m\bar{\theta}/\tau^2 + \mu_0/\gamma_0^2}{m/\tau^2 + 1/\gamma_0^2}, [m/\tau^2 + 1/\gamma_0^2]^{-1}\right)$$

$$\{1/\tau^2|\theta_1, \ldots, \theta_m, \mu\} \sim Gamma\left(\frac{\eta_0 + m}{2}, \frac{\eta_0\tau_0^2 + \sum(\theta_j - \mu)^2}{2}\right)$$

4. Specify the starting values and prior parameter values:

1) For starting values of parameters in model, we set them based on the observed data. See the R-script.
   - σ^2 for calculation of full conditional distributions of θ_j is overall mean of variance for everyone.
   - μ for calculation of full conditional distributions of τ^2 is overall mean of all observed data.
   - τ^2 for calculation of full conditional distributions of θ_j and μ is variance for the mean value of all the data.
2) For the parameters of three priors, values are set as:
   - For prior of $p(\mu)$ of the younger age group, μ_0 is 4 since the overall mean number of COVID-19 deaths of this group is 4. γ_0^2 is 3 since for the normal distribution, most of the probability is within two standard deviation. Response time is higher than zero, which is $\mu_0 - 2 \times \gamma_0 > 0$, so we choose γ_0 is about 3. The same rule is for older age groups, μ_0 is 6 and γ_0 is 3.
   - For prior of $p(\sigma^2)$ (between-group) and $p(\tau^2)$ (within-group), ν_0 and η_0 are considered as prior sample size meanwhile σ_0^2 and τ_0^2 are considered as the sample variances, which implies when $\nu_0 \rightarrow 0$, the posterior distribution is more objective (Hoff, P.D., 2009). The range of sample data for the younger age group is around 8 and range for the older age group is around 7. So $\sigma_0^2 = \tau_0^2 = 8$ for the younger age group and $\sigma_0^2 = \tau_0^2 = 7$ for the older age group are set. Since different sample will generate different variance, we specify $\nu_0 = \eta_0 = 1$ as weakly informative prior parameter which is set purposely to include less information than whatever actual prior knowledge is available (Gelman, A., 2006) so that prior distribution can weakly centered around this value from other populations (Hoff, P.D., 2009).

Results from Gibbs sampling in R:

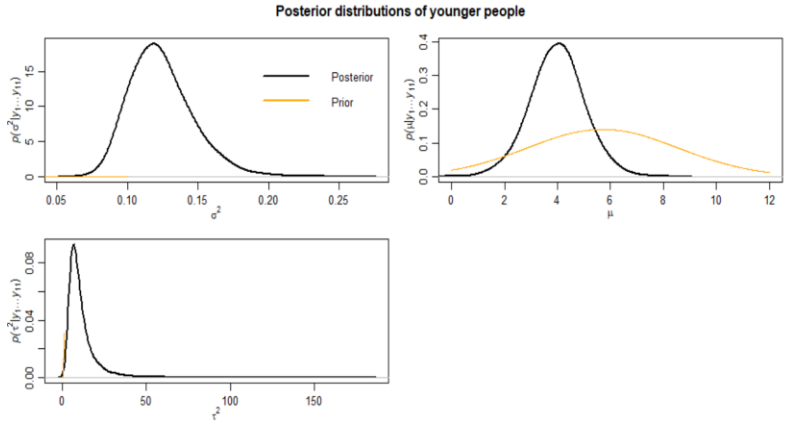

Figure 3: posterior distributions of the younger age group on the COVID-19 deaths

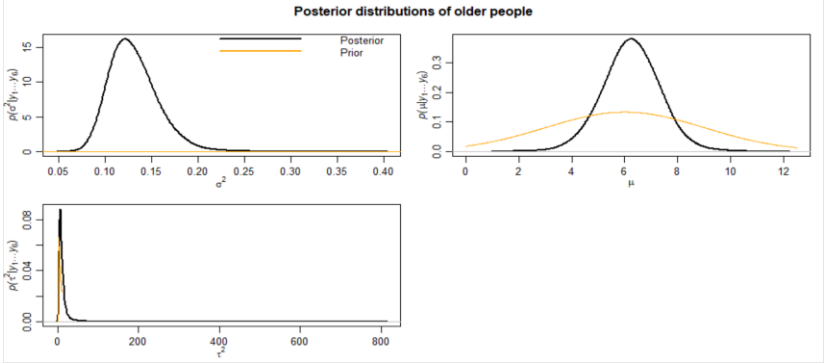

Figure 4: posterior distributions of the older age group on the COVID-19 deaths

The Gibbs sampling algorithm was completed with 10000 iterations based on hierarchical normal modeling with the prior information specified above. The results are shown in Figures 3 and 4 above. Such figures show the marginal posterior distributions for the younger age group and for the older age group of the parameters $\{\sigma^2, \mu, \tau^2\}$. Notice in Figures 3 and 4, that all posterior distributions are more concentrated than their corresponding prior distributions.

According to Figure 5, the posterior means of $\{\sigma^2, \mu, \tau^2\}$ are 0.1239976, 3.9931040 and 11.2238844 respectively to the younger age group. For the older age-group, only the μ differs greatly from that of the younger-age group of the three estimates. The posterior means of $\{\sigma^2, \mu, \tau^2\}$ are 0.1307413, 6.24 and 11.1982316 for the older age group. By comparing the left two panels in Figures 3 and 4 for each age group, it can be noticed that the range for most of posterior of σ^2 is lower than the range for most posterior of τ^2. This indicates that the between-group variability (τ^2) is higher than within-group variability (σ^2). This makes sense since generally variabilities from different individuals is higher than the variabilities from the same individuals on repeated measures.

```
> # mean for all posterior
> apply(MST1,2,mean)
    sigma2         mu       tau2
 0.1239976  3.9931040 11.2238844
> apply(MST2,2,mean)
    sigma2         mu       tau2
 0.1307413  6.2496733 11.1982316
> #probability for posterior mean of older group higher than younger group
> sum(MST1[,2]<MST2[,2])/10000
[1] 0.9377
```
Figure 5: the parameter estimates for the posterior distributions.

Next, in Figure 6, which shows the comparison of the posterior distributions of $\{\sigma^2, \mu, \tau^2\}$ of the younger and older age groups, it can be seen that the posterior distribution of μ (see top right) is higher for the older age group when compared to the younger age group. It is likely that the older age-group has more COVID-19 deaths than the younger age group, which is no surprise at all. The probability that the older age group has a larger average number of deaths than the younger age group is 0.9377, which is very high unfortunately, and consistent with what the CDC warns about older people with the virus. Notice that, however, from the right two plots, we conclude that within-group (σ^2) variabilities and between-group variabilities (τ^2) of the younger age group are higher in general than the older age group, which is consistent with the results above.

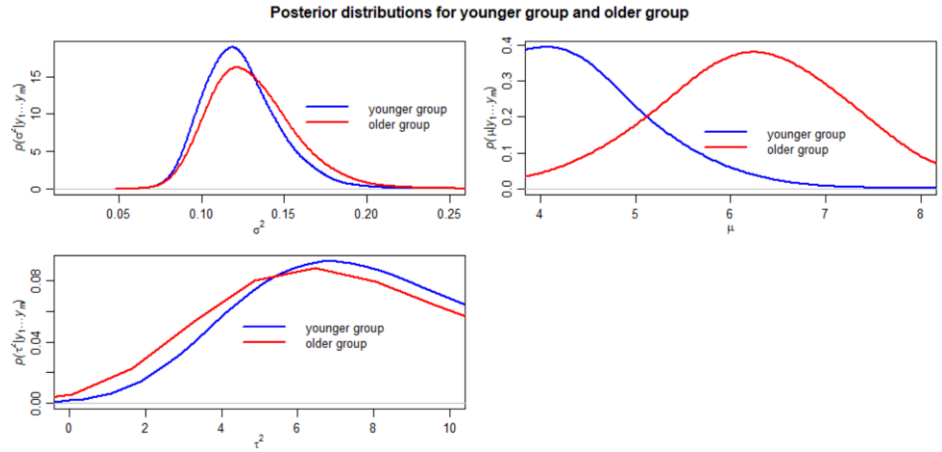

Figure 6: Comparing younger and older age groups posteriors

Convergence of MCMC:

Determining the convergence of the MCMC algorithm can be difficult. The convergence estimation is important because the MCMC simulation is often started at a random point in parameter space. The starting point is often far from the true high-density regions of the posterior distribution. This, the efficiency of the simulated samples need to be evaluated prior to making any inferences (Sahli, K., 2011).The convergence can be evaluated by analyzing trace plots, autocorrelation plots, and by running hypothesis tests.

Trace plots:

A trace plot is defined simply as a plot of the iteration number against the value of each sample of MCMC and is useful for assessing convergence and stationarity. A stationary distribution, or a distribution that is the true posterior distribution, an be inferred from a trace plot with relatively constant mean and variance. A chain in a trace plot that mixes well traverses its posterior space rapidly, and it can jump from one remote region of the posterior to another in relatively few steps ([1]). Figures 7 and 8 are trace plots for each Gibbs sample of posterior for the younger age group and the older age group. All three plots of the three parameters $\{\sigma^2, \mu, \tau^2\}$ for both the younger and older age groups show good mixing. Therefore, overall the MCMC algorithm converges. Notice that the iteration for τ^2 on the bottom left corner has one iteration that is not at all like the rest. There might be some risk that the τ^2 distribution does not converge.

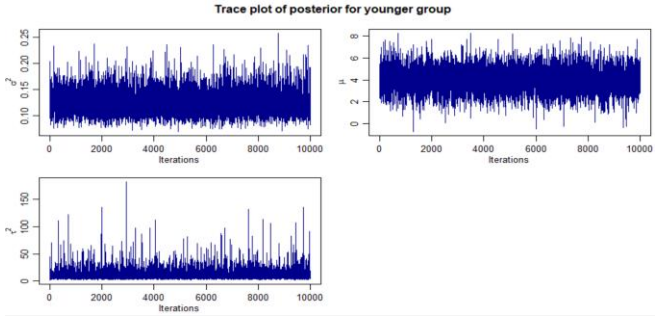

Figure 7: trace plots of posterior parameters $\{\sigma^2, \mu, \tau^2\}$ for the younger group

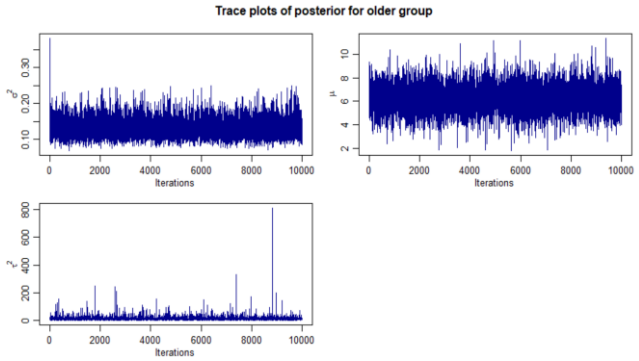

Figure 8: trace plots of posterior parameters $\{\sigma^2, \mu, \tau^2\}$ for the older group

Autocorrelation plots

Individuals in a sample will generally be correlated with each other. This will slow down the Markov chain on the attempt to take a sample from a stationary distribution (Cowels, M.K. and Carlin, B.P., 1996). Autocorrelation is a measure of how independent different posterior distribution samples are. A Markov chain with high autocorrelation moves around the parameter space very slow. This takes the algorithm a while to get enough samples to achieve the correct balance among the different regions of the parameter space and to obtain a level of precision (Hoff, P.D., 2009). Therefore, the smaller the autocorrelation, the faster and more efficient the algorithm.

In Figures 9 and 10 below, are plots of showing the autocorrelation of MCMC sample for each of the three parameter posteriors (σ^2 *is top left*, μ *is right*, τ^2 *is bottom left)*

The y-axis represents the autocorrelation coefficients. The blue dotted line in each of the plots is the 95% confidence interval. We can see all autocorrelations are small and closer to 0 from lag-1 or lag-2. This indicates our six Markov chain converge to a stationary distribution very quickly and take a short time to achieve the correct balance among the different regions of the parameter space.

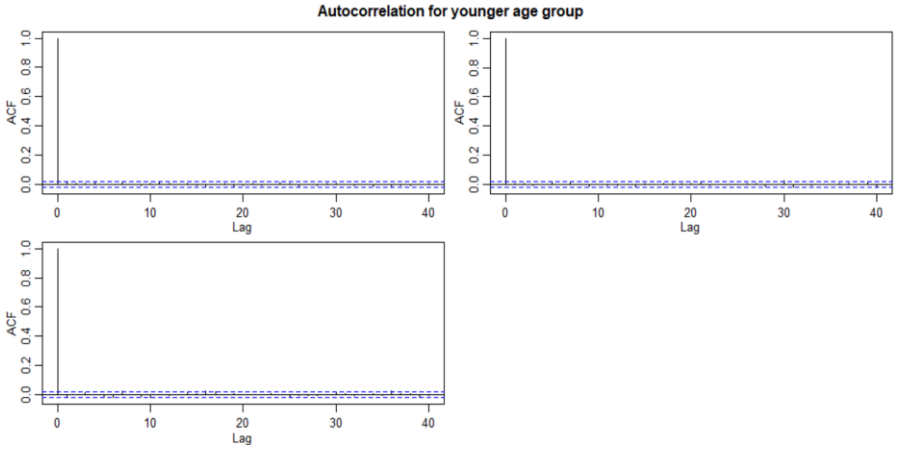

Figure 9: autocorrelation plots of posterior parameters $\{\sigma^2, \mu, \tau^2\}$ respectively for the younger group, (σ^2 is top left, μ is right, τ^2 is bottom left)

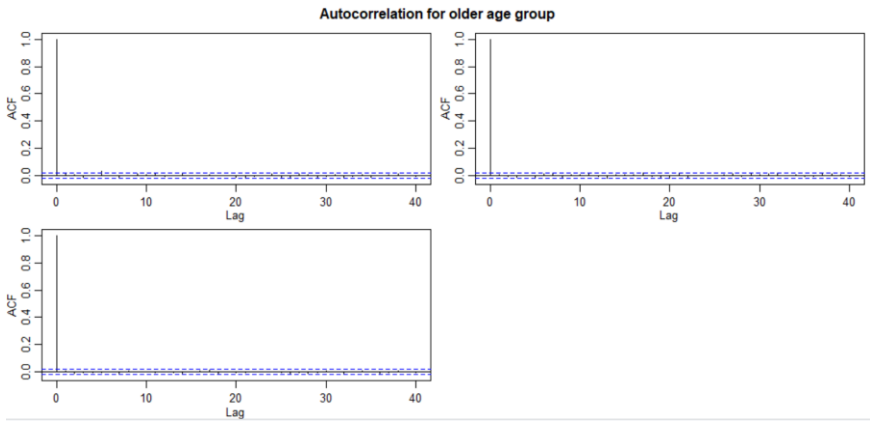

Figure 10: autocorrelation plots of posterior parameters $\{\sigma^2, \mu, \tau^2\}$ for the older group, (σ^2 is top left, μ is right, τ^2 is bottom left)

Tests for convergence:

The Geweke diagnostic compares the mean and variance of segments from the beginning and end of a single chain (usually the first 0.1 and last 0.5 proportions). This diagnosis procedure uses a hypothesis test to test if the difference of means test to determine if the both parts of the chain come from the same distribution (the null hypothesis) or not (the alternative hypothesis). The test statistic for the hypothesis test is a standard Z-score with the standard errors adjusted for autocorrelation. As the chain length $\rightarrow \infty$, then the sampling distribution of the Z-scores $\rightarrow$ N(0,1) if the chain has converged.

The in Figure 11, results from R for the result of MCMC for the younger age group and the older age group are shown. Because there are 10000 iterations for these six chains, the first 1000 samples and last 5000 samples for each chain are compared. The absolute value of the Z-scores for all but the μ for the younger age group are less than 1.96 (p-value>0.05). This indicates a rejection null hypothesis. Therefore, the two parts of each chain are not under same distribution.

```
> #Geweke diagnosis for convergency
> geweke.diag(MST1)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

  sigma2       mu     tau2
-1.00970  2.16941 -0.08823

> geweke.diag(MST2)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

sigma2     mu   tau2
1.0603 0.1346 0.5512
```

Figure 11: Geweke diagnosis for convergency results

In Figure 12 parts (a) and (b), horizontal dotted lines at Z=-1.96, 1.96 represent the 95% confidence interval for an N(0, 1) distribution. All Z-scores of each MCMC are distributed from -2 to 2 except 4 outliers of the chain for μ of the younger age group and 1 outlier from the chain of within-group variance (σ^2) of younger age group, and 1 outliers of the chain for μ of the older age group and 3 outlier from the chain of within-group variance (σ^2) of older age group. Since many of the Z-scores fall outside of the confidence interval, there is a possible convergence failure (Best, N., Cowles, M.K. and Vines, K., 1995). However, it is safe to conclude that the simulated

MCMC samples are converged. If there's lots of Z-scores far outside of the interval, run longer chain should be run.

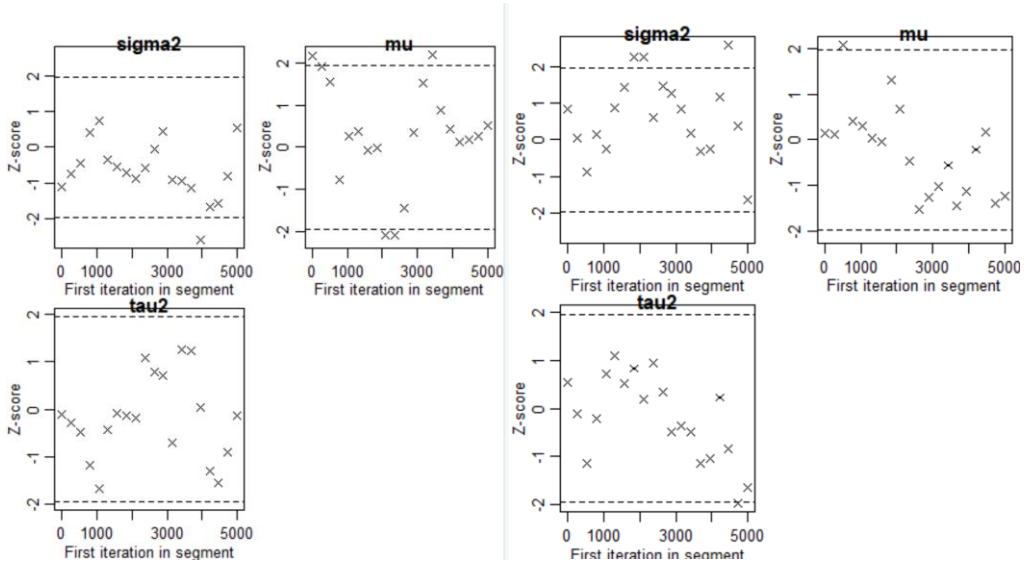

Figure 12: plots of z-score vs. first iteration of the posterior parameters $\{\sigma^2, \mu, \tau^2\}$ for (a) the younger age group (left), and (b) the older age group (right)

Discussion and conclusion:

A Markov chain method called Gibbs sampling was produced to approximate the posterior distribution of parameters from Bayesian hierarchical modeling based on the covid-19 death from the younger group and the older group and predict the probability of the younger group having a lower covid-19 death count than the older group. This probably was a little over 90%. The results appeared to converge, although there is a possible convergence failure. To deal with any possible convergence failure because the model needs to be confirmed to converge, in further research it is best to increase the number of iterations (i.e. from 10000 to 100000) and/or to change the initial values (i.e. changing $\nu_0 = \eta_0 = 1$ to $\nu_0 = \eta_0 = 7$ or 8).

References

Rathnayake, R.C. (2010) Bayesian Inference and Computation. Research Papers. Paper 29.

Parker, M. (2005) Foundations of Statistics – Frequentist and BayesianMartin, A.D., Quinn, K.M. & Park, J.H. (2011) MCMCpack: Markov Chain Monte Carlo in R: Journal of Statistical Software, volume 42, issue 9

Cowels, M.K. and Carlin, B.P. (1996) Markov Chain Monte Carlo Convergence Diagnostics: A comparative Review. Journal of the American Statistical Association

Belin, T.R. and Rubin, D.B. (1990). Analysis of a finite mixture model with variance components. Proceeding of the Social Social Statistics Section 211-215. ASA, Alexandria, Va.

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari A. and Rubin, D.B. (2013) Bayesian Data Analysis: Chapman & Hall/CRC Texts in Statistical Science, 3rd Edition

Stevens, J.W. (2009) What is Bayesian Statistics?

Sahli, K. (2011) Estimating convergence of Markov chain Monte Carlo simulations: Master thesis for Mathematical Statistics, Stockholm University

Geyer, C.J. (2011) Introduction to Markov Chain Monte Carlo: Handbook of Markov Chain Monte Carlo, by Chapman & Hall/CRC

Yildirim, I. (2012) Bayesian Inference: Metropolis-Hastings Sampling

Hoff. P.D. (2009) A First Course in Bayesian Statistical Methods: Springer

Gelman, A. (2006) Prior distribution for variance parameters in hierarchical models: 1, Number 3, pp. 515-533

Best, N., Cowles, M.K. and Vines, K. (1995) CODA* Convergence Diagnosis and Output Analysis Software for Gibbs sampling output: Version 0.30

[1] https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/older-adults.html
[2] https://www.cdc.gov/nchs/nvss/vsrr/COVID19/

R Code:

```
setwd("C:/Users/kasch/OneDrive/Desktop")

library(readr)

cdc_covid_age_group_data_April_11_2020 <- read.csv("cdc covid data age groups.csv")
View(cdc_covid_age_group_data_April_11_2020)

RES <- split(cdc_covid_age_group_data_April_11_2020$COVID.19.death,
             cdc_covid_age_group_data_April_11_2020$Age.group.binary)

#make histograms of COVID19 death by
RES$`0` <- as.numeric(RES$`0`)
RES$`1` <- as.numeric(RES$`1`)

y1=t(RES$`0`) #[1,]
y2=t(RES$`1`) #[1,]
n1 <- length(y1[1,])
n2 <- length(y2[1,])
m1=n1;m2=n2;n=n1+n2

boxplot(list(RES$`0`,RES$`1`),range=0,ylab="COVID deaths, in hundreds",
        names=c("Younger age group, 0-44 years",
                "Older age group, 45+ years"))
##Gibbs sampling

#Set starting values for younger individuals from y1
theta1=y1bar=apply(y1,2,mean)
sv1=apply(y1,1,var, na.rm = T)
sigma2_1=mean(sv1)
mu1=mean(y1)
tau2_1=var(y1bar)

#Set starting values for older individuals from y2
theta2=y2bar=apply(y2,2,mean)
sv2=apply(y2,1,var, na.rm = T)
sigma2_2=mean(sv2)
mu2=mean(y2)
tau2_2=var(y2bar)

#Given weakly informative prior parameters for covid19 deaths
#younger group:
nu0_1=1; s20_1=8           #within gropus (for prior of sigma2)
eta0_1=1; t20_1=8          #between groups  (for prior of tau2)
mu0_1=4; g20_1=3           #for prior of mu

#older group:
nu0_2=1; s20_2=7
eta0_2=1; t20_2=7
mu0_2=6; g20_2=3

#Setup MCMC
set.seed(234)
S=10000
THETA1=matrix(nrow=S,ncol=m1)
THETA2=matrix(nrow=S,ncol=m2)
MST1=MST2=matrix(nrow=S,ncol=3)
colnames(MST1)=colnames(MST2)=paste(c("sigma2","mu","tau2"))
```

```
#####MCMC algorithm#####
for(s in 1:S){
  for(j in 1:m1){
    ## sample new values of the thetas
    vtheta=1/(n/sigma2_1+1/tau2_1)
    etheta=vtheta*(y1bar[j]*n/sigma2_1+1/tau2_1)
    theta1[j]=rnorm(1,etheta,sqrt(vtheta))
  }
  for(j in 1:m2){
    vtheta=1/(n/sigma2_2+1/tau2_2)
    etheta=vtheta*(y2bar[j]*n/sigma2_2+1/tau2_2)
    theta2[j]=rnorm(1,etheta,sqrt(vtheta))
  }

  #sample new value of sigmas
  nun_1=nu0_1+n*m1
  ss_1=nu0_1*s20_1
  for(j in 1:m1){
    ss_1=ss_1+sum((y1[,j]-theta1[j])^2)
  }
  sigma2_1=1/rgamma(1,nun_1/2,ss_1/2)

  nun_2=nu0_2+n*m2
  ss_2=nu0_2*s20_2
  for(j in 1:m2){
    ss_2=ss_2+sum((y2[,j]-theta2[j])^2)
  }
  sigma2_2=1/rgamma(1,nun_2/2,ss_2/2)


  #sample new values of mu
  vmu_1=1/(m1/tau2_1+1/g20_1)
  emu_1=vmu_1*(m1*mean(theta1)/tau2_1+mu0_1/g20_1)
  mu_1=rnorm(1,emu_1,sqrt(vmu_1))

  vmu_2=1/(m2/tau2_2+1/g20_2)
  emu_2=vmu_2*(m2*mean(theta2)/tau2_2+mu0_2/g20_2)
  mu_2=rnorm(1,emu_2,sqrt(vmu_2))

  #sample new values of tau
  etam_1=eta0_1+m1
  ss_1=eta0_1*t20_1 + sum((theta1-mu1)^2)
  tau2_1=1/rgamma(1,etam_1/2,ss_1/2)

  etam_2=eta0_2+m2
  ss_2=eta0_2*t20_2 + sum((theta2-mu2)^2)
  tau2_2=1/rgamma(1,etam_2/2,ss_2/2)

  #store results
  THETA1[s,]=theta1
  THETA2[s,]=theta2
  MST1[s,]<-c(sigma2_1,mu_1,tau2_1)
  MST2[s,]<-c(sigma2_2,mu_2,tau2_2)
}

#install.packages("invgamma")
library(invgamma)

#####posterior for younger people#####
par(mfrow=c(2,2),mar=c(3,3,1,1),mgp=c(1.75,.75,0),oma=c(0,0,2,0))
plot(density(MST1[,1],adj=2),lwd=2,main="",xlab=expression(sigma^2),
     ylab=expression(paste(italic("p("),sigma^2,"|",italic(y[1]),"...",
                           italic(y[11]),")")))
```

```
x11=seq(0,0.1,length=1000)
lines(x11,dinvgamma(x11,nu0_1/2,nu0_1*s20_1/2),type="l",col="orange")
legend(0.18,18,legend=c("Posterior","Prior"),lwd=c(2,2),
       col=c("black","orange"),bty="n")
plot(density(MST1[,2],adj=2),xlab=expression(mu),main="",lwd=2,xlim=c(0,12),
     ylab=expression(paste(italic("p("),mu,"|",italic(y[1]),"...",
                           italic(y[11]),")")))
x21=seq(0,12,length=1000)
lines(x21,dnorm(x21,5.7,2.85),type="l",col="orange")
legend(0.18,18,legend=c("Posterior","Prior"),lwd=c(2,2),
       col=c("black","orange"),bty="n")
plot(density(MST1[,3],adj=2),xlab=expression(tau^2),main="",lwd=2,
     ylab=expression(paste(italic("p("),tau^2,"|",italic(y[1]),"...",
                           italic(y[11]),")")))
x31=seq(0,1.2,length=1000)
lines(x31,dinvgamma(x31,eta0_1/2,eta0_1*t20_1/2),type="l",col="orange")
legend(0.18,18,legend=c("Posterior","Prior"),lwd=c(2,1),
       col=c("black","orange"), bty="n")
title(main="Posterior distributions of younger people",outer=TRUE)

#####posterior for older people#####
par(mfrow=c(2,2),mar=c(3,3,1,1),mgp=c(1.75,.75,0),oma=c(0,0,2,0))
plot(density(MST2[,1],adj=2),lwd=2,main="",xlab=expression(sigma^2),
     ylab=expression(paste(italic("p("),sigma^2,"|",italic(y[1]),"...",
     italic(y[6]),")")))
x21=seq(0,1,length=1000)
lines(x21,dinvgamma(x21,nu0_2/2,nu0_2*s20_2/2),type="l",col="orange")
legend(0.18,18,legend=c("Posterior","Prior"),lwd=c(2,2),
       col=c("black","orange"),bty="n")
plot(density(MST2[,2],adj=2),xlab=expression(mu),main="",lwd=2,xlim=c(0,12.5),
     ylab=expression(paste(italic("p("),mu,"|",italic(y[1]),"...",
     italic(y[6]),")")))
x22=seq(0,12.5,length=1000)
lines(x22,dnorm(x22,6,3),type="l",col="orange")
legend(7,1.25,legend=c("Posterior","Prior"),lwd=c(2,2),
       col=c("black","orange"),bty="n")
plot(density(MST2[,3],adj=2),xlab=expression(tau^2),main="",lwd=2,
     ylab=expression(paste(italic("p("),tau^2,"|",italic(y[1]),"...",
     italic(y[6]),")")))
library(invgamma)
x23=seq(0,10,length=1000)
lines(x23,dinvgamma(x23,eta0_2/2,eta0_2*t20_2/2),type="l",col="orange")
legend(4,1.7,legend=c("Posterior","Prior"),lwd=c(2,2),
       col=c("black","orange"),bty="n")
title(main="Posterior distributions of older people",outer=TRUE)

#####Compare younger and older age groups posteriors#####
par(mfrow=c(2,2),mar=c(3,3,1,1),mgp=c(1.75,.75,0),oma=c(0,0,2,0))
plot(density(MST1[,1],adj=2),lwd=2,col="blue",main="",xlab=expression(sigma^2),

xlim=c(0.02,0.25),ylab=expression(paste(italic("p("),sigma^2,"|",italic(y[1]),"...",
                                        italic(y[m]),")")))
lines(density(MST2[,1],adj=2),lwd=2,col="red")
legend(0.15,13,legend=c("younger group","older group"),lwd=c(2,2),
       col=c("blue","red"),bty="n")
plot(density(MST1[,2],adj=2),lwd=2,col="blue",main="",xlab=expression(mu),
     xlim=c(4,8),ylab=expression(paste(italic("p("),mu,"|",italic(y[1]),"...",
                                       italic(y[m]),")")))
lines(density(MST2[,2],adj=2),lwd=2,col="red")
legend(5.5,0.2,legend=c("younger group","older group"),lwd=c(2,2),
       col=c("blue","red"),bty="n")
plot(density(MST1[,3],adj=2),lwd=2,col="blue",main="",xlab=expression(tau^2),
     xlim=c(0,10),ylab=expression(paste(italic("p("),tau^2,"|",italic(y[1]),"...",
```

```
                                        italic(y[m]),")")))
lines(density(MST2[,3],adj=2),lwd=2,col="red")
legend(4,0.06,legend=c("younger group","older group"),lwd=c(2,2),
       col=c("blue","red"),bty="n")
title(main="Posterior distributions for younger group and older group",
      outer=TRUE)

# mean for all posterior
apply(MST1,2,mean)
apply(MST2,2,mean)

#probability for posterior mean of older group higher than younger group
sum(MST1[,2]<MST2[,2])/10000

######Convergence of MCMC#####
#trace plots for MCMC
par(mfrow=c(2,2),oma=c(0,0,2,0))
plot(1:S,MST1[,1],type="l",col="dark blue",xlab="Iterations",ylab=expression(sigma^2))
plot(1:S,MST1[,2],type="l",col="dark blue",xlab="Iterations",ylab=expression(mu))
plot(1:S,MST1[,3],type="l",col="dark blue",xlab="Iterations",ylab=expression(tau^2))
title(main="Trace plot of posterior for younger group",outer=TRUE)
par(mfrow=c(2,2),oma=c(0,0,2,0))
plot(1:S,MST2[,1],type="l",col="dark blue",xlab="Iterations",ylab=expression(sigma^2))
plot(1:S,MST2[,2],type="l",col="dark blue",xlab="Iterations",ylab=expression(mu))
plot(1:S,MST2[,3],type="l",col="dark blue",xlab="Iterations",ylab=expression(tau^2))
title(main="Trace plots of posterior for older group",outer=TRUE)

##Autocorrelation plots
# #trace plots for MCMC
# par(mfrow=c(2,2),oma=c(0,0,2,0))
# plot(1:S,MST1[,1],type="l",col="dark
blue",xlab="Iterations",ylab=expression(sigma^2))
# plot(1:S,MST1[,2],type="l",col="dark blue",xlab="Iterations",ylab=expression(mu))
# plot(1:S,MST1[,3],type="l",col="dark blue",xlab="Iterations",ylab=expression(tau^2))
# title(main="Trace plot of posterior for younger age group",outer=TRUE)
# par(mfrow=c(2,2),oma=c(0,0,2,0))
# plot(1:S,MST2[,1],type="l",col="dark
blue",xlab="Iterations",ylab=expression(sigma^2))
# plot(1:S,MST2[,2],type="l",col="dark blue",xlab="Iterations",ylab=expression(mu))
# plot(1:S,MST2[,3],type="l",col="dark blue",xlab="Iterations",ylab=expression(tau^2))
# title(main="Trace plots of posterior for older age group",outer=TRUE)

# function of stationarity
stationarity.plot<-function(x,...){
  S<-length(x)
  scan<-1:S
  ng<-min( round(S/100),10)
  group<-S*ceiling( ng*scan/S) /ng
  boxplot(x~group,...)
}
# produce boxplots of sequential groups for s2, μ and t2
par(mfrow=c(1,3),mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0),oma=c(0,0,2,0))
stationarity.plot(MST1[,1],xlab="iteration",ylab=expression(sigma^2))
stationarity.plot(MST1[,2],xlab="iteration",ylab=expression(mu))
stationarity.plot(MST1[,3],xlab="iteration",ylab=expression(tau^2))
title(main="Group boxplots for posterior of younger age group",outer=TRUE)
par(mfrow=c(1,3),mar=c(2.75,2.75,.5,.5),mgp=c(1.7,.7,0),oma=c(0,0,2,0))
stationarity.plot(MST2[,1],xlab="iteration",ylab=expression(sigma^2))
stationarity.plot(MST2[,2],xlab="iteration",ylab=expression(mu))
stationarity.plot(MST2[,3],xlab="iteration",ylab=expression(tau^2))
title(main="Group boxplots for posterior of older age group",outer=TRUE)

# autocorrelation checking for each posterior of each group
```

```
par(mfrow=c(2,2),oma=c(0,0,2,0))
acf(MST1[,1],main=expression(sigma^2))
acf(MST1[,2],main=expression(mu))
acf(MST1[,3],main=expression(tau^2))
title(main="Autocorrelation for younger age group",outer=TRUE)
par(mfrow=c(2,2),oma=c(0,0,2,0))
acf(MST2[,1],main=expression(sigma^2))
acf(MST2[,2],main=expression(mu))
acf(MST2[,3],main=expression(tau^2))
title(main="Autocorrelation for older age group",outer=TRUE)
#acf(MST1[,1],40)$acf

#Check effetiveSize for sigma, mu and tau
#install.packages("lattice")
library(coda)
effectiveSize(MST1)
effectiveSize(MST2)

#Raftery-Lewis diagnosis for convegence
raftery.diag(mcmc(MST1))
raftery.diag(mcmc(MST2))

#Geweke diagnosis for convergency
geweke.diag(MST1)
geweke.diag(MST2)
geweke.plot(mcmc(MST1))
geweke.plot(mcmc(MST2))

#Heidelberger-Welch diagnosis for convergency
heidel.diag(mcmc(MST1))
heidel.diag(mcmc(MST2))
```