Final Project

Hunter Schuller

4/26/2022

The hierarchical model I am proposing for the status of K9C9 is as follows. Since each observation is conditionally independent from each other, we can consider the probability of all observations to be the product of the probability of each observation from every country. We have that each ith observation Y from the jth country follows a Bernoulli distribution with the probability of θ_{ij} of the ith observation from the jth country. The index i varies from 1 to the number of observations from each country, which is 100. The index j varies from 1 to the number of countries, which is 10. The logit of theta is equal to $\alpha_j + \beta_j * x_{ij}$. There are j elements of alpha and beta, for each country. The variable x is a binary variable for the EZK test status; it will be 1 if it is a positive result, and 0 if it is a negative result. For alpha and beta, we can assume conditional independence and say the joint probability given each of its means and variances is equal to the product of the probability of each α_j given its mean and variance times the probability of each β_j given its mean and variance. Each alpha and beta will follow a normal distribution, with parameters $\mu_\alpha, \sigma^2_\alpha, \mu_\beta, \sigma^2_\beta$, respectively. We can assume conditional independence and say that the joint probabilities of the means and variances of alpha and beta is equal to the products of all the probabilities of μ's and σ^2's. The means, μ_α and μ_β, will both follow normal distributions, with parameters μ is equal to 0 and σ is equal to 1. The variances,σ^2_α and σ^2_β, will both follow a uniform distribution with parameters minimum of 1 and maximum of 2.

These parameters are all used to predict the outcome of `Infected`. The theta parameter equates to the probability that person Y_{ij} is infected with K9C9. The α_j parameter is the jth country's log odds of an infected observation in the reference group, which would be someone who tested negative on the EZK test. The β_j parameter is the increase in the jth country's log odds for an infected observation that tested positive on the EZK test versus tested negative on the EZK test. The μ_α parameter is the mean of the log odds of any observation testing positive on the EZK. The μ_β parameter is the mean increase in log odds for an infected observation that tested positive on the EZK test versus tested negative. The σ^2_α parameter is the deviation across countries in the country-specific log odds among the reference group. The σ^2_β parameter

is the deviation across countries in the country-specific log odds increase for an infected observation that tested positive on the EZK test versus tested negative.

The final model can be used to assess the quality of an EZK test's results. The final model takes into account the multiple levels of variability. Each country has its own tolerance to the virus, which is accounted for in the variance of each alpha and beta parameter.The variance of the alpha and beta parameters follows a uniform distribution between 1 and 2. This is enough to account for the variation between each country's population and their tolerance to the K9C9 virus. Each alpha and beta parameter's mean has its own mean and variance. This next level of the hierarchical model accounts for other levels of variation in the problem, such as accuracy of each country's tests or how easily the virus can be spread in each country.

The final model was run for 50,000 iterations total. There were 10,000 iterations set aside for both burn in period and adaptive steps. The starting values are as follows: $\mu_\alpha = 0$, $\mu_\beta = 0$, $\sigma_\alpha = 0$, $\sigma_\beta = 0$, and all of the $\alpha_j's$ and $\beta_j's$ started at 0. The model did converge with these starting values and after all the iterations. The trace plots for the thetas, alphas, and betas passed the "Fat Marker Test". The Effective Sample Size for each parameter was at least 7,000 and at most 16,000. All of the values are at least 15% of the number of iterations of 50,000, which means they are all effective. The ACF plots for the parameters all show there is little lag after a few iterations. With all of these diagnostics in mind, we can say the model has converged and is able to be used.

The model can be used to test the efficiency of the EZK test, and how the virus is impacting the population of each country. The alpha values will show the probability of a person in country j to be infected with a negative EZK test. Using the mean values from the rjags output, we can see that the EZK test is more efficient in different countries. For example in country A, the log odds of an infected person testing negative on the EZK test is .554. In country B, the log odds of an infected person testing negative on the EZK test is 0.24. We can see that the EZK test is more effective at detecting K9C9 in country B over country A. This difference is still prevalent when comparing the outputs of infected people. For a positive EZK test, the log odds of being infected in country A increases by 0.77. For the same situation in country B, the log odds of being infected increases by 0.64. For each country, these values ultimately impact the probability that a given person from country j is infected or not. For example, in country A, the probability that someone is infected is 0.71. In country B, this probability is 0.3. Each country has a difference in the efficiency of the EZK test and how much of the population is infected.

The final model and decision theory can be used to help the insurance provider with the treatment of Neonicon. There are two approaches that can be taken: giving Neonicon to any person with a positive EZK test, or paying for the hospital treatment for every infected person. Using decision theory, we can calculate the amount of money the insurance provider would have to spend in each scenario, and ultimately choose

the one with the least amount of money spent.

Looking at the scenario where the insurance company gives Neonicon to every EZK positive test, we can compute how much money it would cost for each person in their population. In the data, 52% of the people in country D tested positive for the EZK test. For each of those people, the insurance company would have to pay for their Neonicon, which would be 457 dollars. Taking the probability and multiplying the cost of Neonicon, we get that the cost per person in their population would be 237.64 dollars.

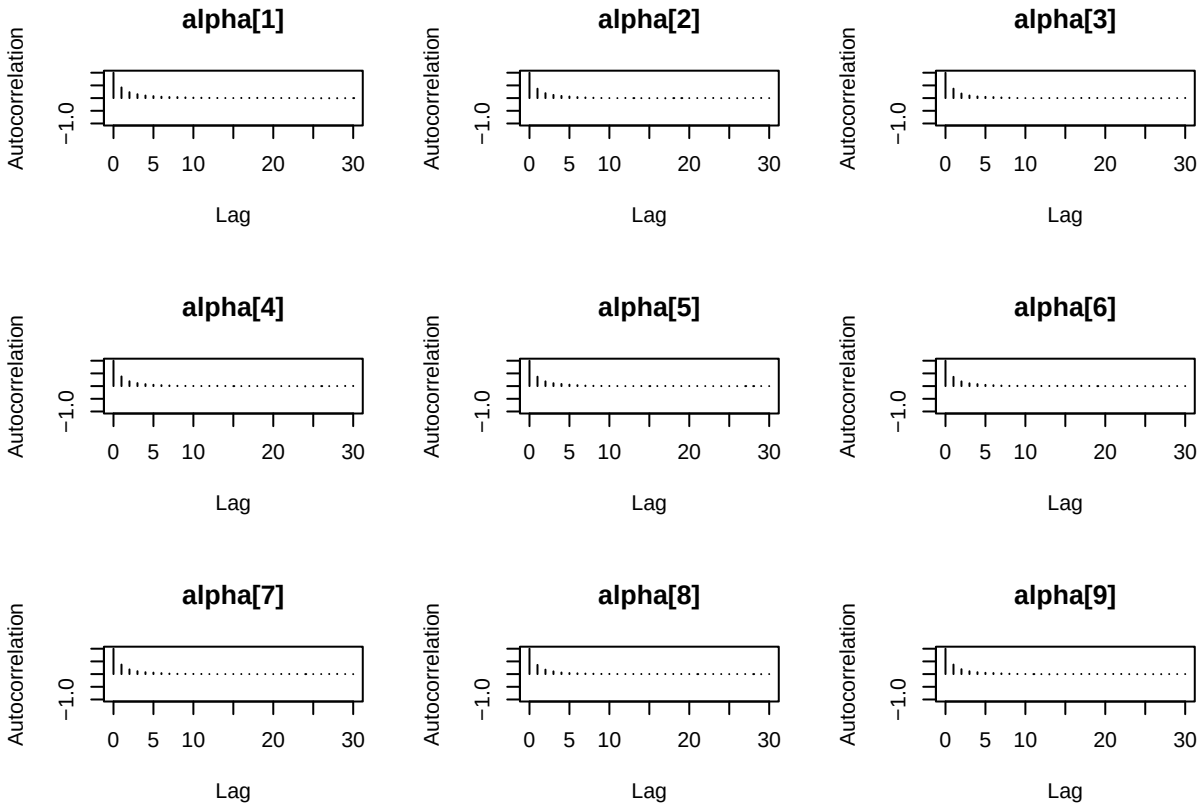
Looking at the scenario where the insurance company pays for hospital bills of each infected, we can compute how costly it would be for each infected person in their population. According to the data, 58% of the population was infected with K9C9. For each of those people, the insurance company would have to pay 1490 dollars in hospital fees. Taking these values and multiplying them together would get a cost per person of 864.20 dollars. This is assuming that the hospitalization rate for the virus is 100%. If we take in to account the chance it is not, we can say the cost per person is $864.20L, where L is the hospitalization percentage of the country D.

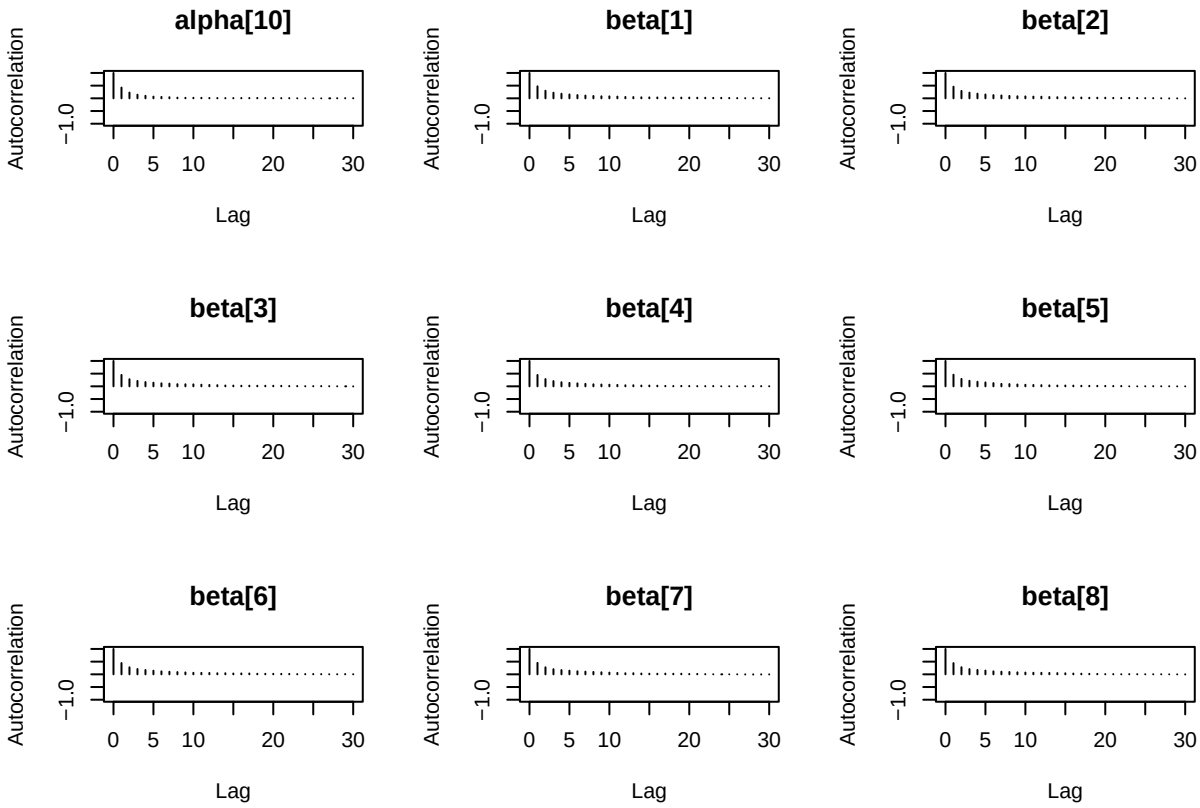
For a value of L greater than 0.27, which equates to a 27% hospitalization rate in country D, it is more cost efficient to pay for Neonicon for every positive EZK test. For a value of L less than 0.27, or a hospitalization rate of less than 27%, it is more cost efficient to pay for the hospital bills of the infected people. With this info, the insurance company would be able to find the path that minimizes costs.

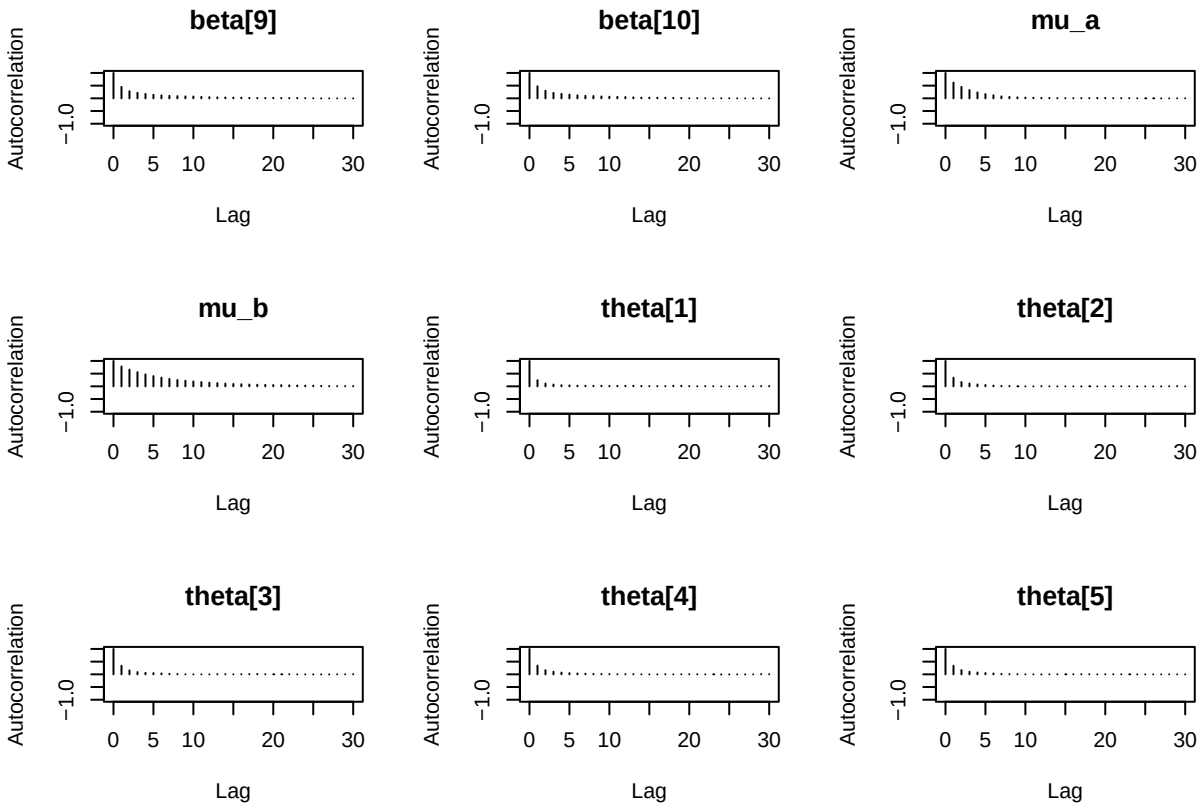
```
par(mfrow=c(1,1))
effectiveSize(window(fit.samples,start=nburns+nadapt)[[1]])
```

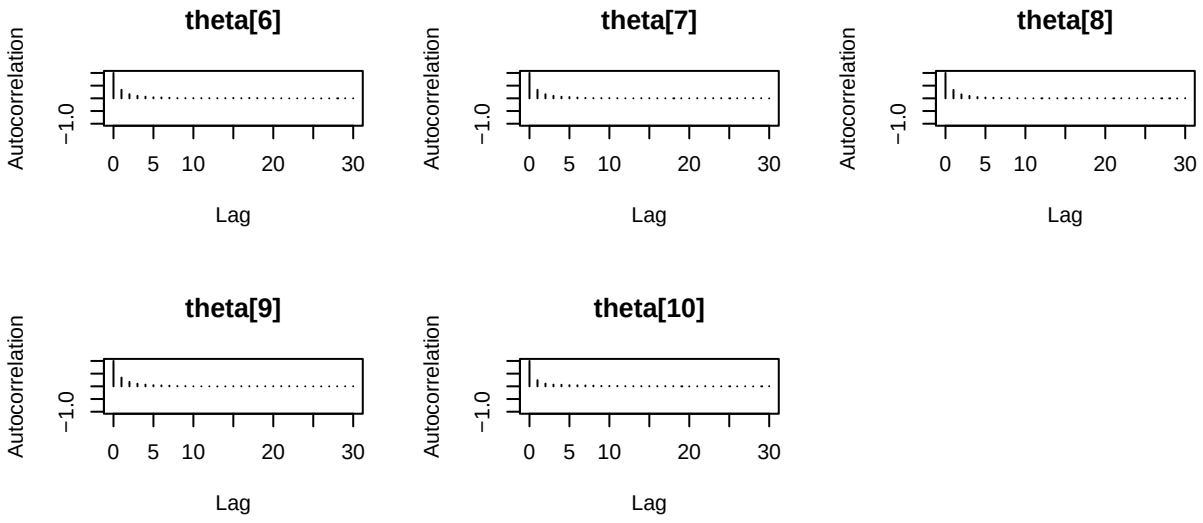
```
##  alpha[1]  alpha[2]  alpha[3]  alpha[4]  alpha[5]  alpha[6]  alpha[7]  alpha[8]
## 11629.159 13864.705 14515.232 14305.061 14224.392 14825.181 14246.278 15061.265
##  alpha[9] alpha[10]   beta[1]   beta[2]   beta[3]   beta[4]   beta[5]   beta[6]
## 14045.419 12017.415  7240.500  7202.944  7729.400  7902.021  8110.568  7471.192
##   beta[7]   beta[8]   beta[9]  beta[10]      mu_a      mu_b  theta[1]  theta[2]
##  7993.821  7422.511  7859.066  7685.936  7439.031  3394.098 15178.553 15025.932
##  theta[3]  theta[4]  theta[5]  theta[6]  theta[7]  theta[8]  theta[9] theta[10]
## 15574.140 15157.880 15289.151 15435.653 15595.862 17022.093 14978.129 16914.276
```

```
autocorr.plot(window(fit.samples,start=nburns+nadapt)[[1]])
```




alpha[10]
beta[1]
beta[2]
beta[3]
beta[4]
beta[5]
beta[6]
beta[7]
beta[8]
Autocorrelation
−1.0
0
5
10
20
30
Lag


beta[9]
beta[10]
mu_a
mu_b
theta[1]
theta[2]
theta[3]
theta[4]
theta[5]
Autocorrelation
−1.0
0
5
10
20
30
Lag



#Appendix ##Code

```
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)

library(ggplot2)

library(rjags)

library(coda)

library(readr)

setwd("~/STAT 3303")

flu <- read.csv("data 75")

changeToNumeric <- function(x){

  y <- 0

  if(x == 'A'){

    y = 1

  }else if(x == 'B'){

    y = 2
```

```
  }else if(x == 'C'){
    y = 3
  }else if(x == 'D'){
    y = 4
  }else if(x == 'E'){
    y = 5
  }else if(x == 'F'){
    y = 6
  }else if(x == 'G'){
    y = 7
  }else if(x == 'H'){
    y = 8
  }else if(x == 'I'){
    y = 9
  }else if(x == 'J'){
    y = 10
  }
  return(y)
}
for(i in 1:nrow(flu)){
  flu$Country[i] <- changeToNumeric(flu$Country[i])
}
expit <- function(x){
  return(exp(x)/(1+exp(x)))
}
niters=50000
nburns=10000
nadapt=10000
nchains=2

country <- c(1:10)
y <- flu$Infected
```

```
x <- flu$EZK
n <- length(country)
theta <- rep(0.5, n)
numObs <- rep(100, 10)
numINf <- pull(flu %>% group_by(Country) %>%
  summarise(inf = sum(Infected), EZK = sum(EZK)) %>%
  select(inf))
numEZK <- pull(flu %>% group_by(Country) %>%
  summarise(inf = sum(Infected), EZK = sum(EZK)) %>%
  select(EZK))

mydata = list(n=n,y=y,x=x,country=country)

myinit = list(mu_a = 0, mu_b = 0, sigma_a = 1.5, sigma_b = 1.5, beta=rep(0,n), alpha=rep(0,n))

mod = "model {
  # likelihood
  for (i in 1:n) {
    y[i] ~ dbern(theta[i])
    logit(theta[i]) <- alpha[country[i]] + beta[country[i]]*x[i]
  }

  # priors
  for (j in 1:n) {
    alpha[j] ~ dnorm(mu_a,tau_sig_a)
    beta[j] ~ dnorm(mu_b,tau_sig_b)
  }
  mu_a ~ dnorm(0,tau_mu_a)
  mu_b ~ dnorm(0,tau_mu_b)
  tau_mu_a <- pow(1, -2)
  tau_mu_b <- pow(1, -2)
  sigma_a ~ dunif(1,2)
```

```
  sigma_b ~ dunif(1,2)
  tau_sig_a <- pow(sigma_a, -2)
  tau_sig_b <- pow(sigma_b, -2)

}"
fit=jags.model(textConnection(mod),
               data=mydata, inits=myinit, n.chains=nchains,n.adapt=nadapt)
fit.samples=coda.samples(fit,c("theta", "alpha", "beta", "mu_a", "mu_b"),n.iter=niters)
par(mar=c(1, 1, 1, 1))
summary(window(fit.samples,start=nburns+nadapt))
countryD <- flu %>%
  filter(Country == 4) %>%
  summarize(posRate = sum(Infected == 1)/100,
            negRate = sum(Infected == 0)/100,
            posBoth = sum(Infected == 1 & EZK == 1)/100,
            negBoth = sum(Infected == 0 & EZK == 0)/100,
            falseNeg = sum(Infected == 1 & EZK == 0)/100,
            falsePos = sum(Infected == 0 & EZK == 1)/100,
            EZKPos = sum(EZK == 1)/100)
medicine <- 457
hospTreat <- 1490
lossMedicine <- medicine*countryD$posBoth + medicine*countryD$falsePos
lossHospTreat <- hospTreat*countryD$posRate
```