

# Denis Ostroushko - PUBH 7440 - HW4 - Part 2

## Problem 1

### Prerequisites

In this assignment we analyze stroke-related mortality rates at the county-age-group levels in PA.

We have 67 counties,  $i = 1, 2, \dots, 67$ , and three age groups  $a = 1, 2, 3$  within each county

We assume that the number of observed death in county  $i$  and age group  $a$  is distributed by a Poisson distribution with parameter  $n_{ia}\lambda_{ia}$ , where  $\log\lambda_{ia} = \theta_{0a}$ .

So, the death rate for each county is a function of a random variable with some underlying distribution.

Given that Poisson distribution parameter is a function of two random variables, we can write pmf of  $Y_{ia}$  as:

$$Y_{ia} = \frac{e^{-(n_{ia}e^{\theta_{0a}})} \times (n_{ia}e^{\theta_{0a}})^{Y_{ia}}}{Y_{ia}!}$$

As mentioned previously,  $\theta_{0a}$  is a random variable {because Bayesian Analysis framework}, and therefore it has a prior distributions:

$\theta_{ia}|\beta_{0a}, \sigma_a^2 \sim N(\beta_{0a}, \sigma_a^2)$ . This equation represents 201 distributions for each county. Within an age group, log death rates for the county are iid normal random variables.

$\beta_{0a}|\mu = 0, \tau_a^2 \sim N(0, \tau_a^2)$ , where  $\tau^2 = 10,000$ . So, the prior distribution of the average log death rate of a given age group is non-informative.

$\sigma_a^2 \sim IG(0.001, 0.001)$ , so variance comes from a non-informative Inverse Gamma (IG) distribution.

**For suppressed values of deaths with county and age-groups levels:** refer to HW4-Part 1 for the imputation scheme.

### Full hirerachical model

$$\begin{aligned}
 p(\beta_{0a}, \theta_{ia}, \sigma_{0a}^2 | \mathbf{Y}) &\propto \prod_{i,a} [Pois(Y_{ia} | n_{ia} * \exp(\theta_{0a}))] \times && \text{full data likelihood} \\
 &I(Y_{ia} > 10) \times && \text{indicator of observed } Y_{ia} \\
 &Norm(\theta_{0a} | \beta_{0a}, \sigma_a^2) \times && \text{prior for } \theta_{0a} \\
 &Norm(\beta_{0a} | 0, \tau_a^2) \times && \text{prior for } \beta_{0a} \\
 &IG(\sigma_a^2 | 0.001, 0.001) \times && \text{prior for } \sigma_a^2
 \end{aligned}$$

## Problem 2

### Full conditional for $\beta_{0a}$

- Term  $\beta_{0a}$  appears in the distribution of  $\theta_{ia}$  as the mean, and in its own distribution. Note,  $\beta_{0a}$  is the mean for the age group, and is the average log death rate for a given age group. Therefore, we need to consider a joint likelihood function for 67 iid  $\theta_{ia}$ .

Full conditional for  $\beta_{0a}$ , holding  $a$  fixed. This full conditional generalizes to three terms for each age group:

$$\begin{aligned}
 p(\beta_{0a} | \cdot) &\propto \prod_{i=1}^{N=67} (2\pi\sigma_a^2)^{-1} \exp\left(-\frac{1}{2\sigma_a^2}(\theta_{ia} - \beta_{0a})^2\right) \times && \text{joint distribution of } \theta_{ia} \\
 &(2\pi\tau_a^2)^{-1} \exp\left(-\frac{1}{2\tau_a^2}\beta_{0a}^2\right) && \text{prior distribution of } \beta_{0a} \\
 &\propto \exp\left(-\frac{1}{2\sigma_a^2} \sum_i (\theta_{ia} - \beta_{0a})^2\right) \times \exp\left(-\frac{1}{2\tau_a^2}\beta_{0a}^2\right)
 \end{aligned}$$

I do not recognize this kernel as a known distribution, so we are not able to use Gibbs sampling technique to obtain posterior distribution of  $\beta_{0a}$ . We will use Metropolis algorithm to obtain samples for the posterior distribution. More on this later, but we will use Metropolis update because we assume a symmetric candidate density for  $\beta_{ia}$

### Full conditional for $\theta_{ia}$

- Term  $\theta_{ia}$  appears in the likelihood of the data and its own prior distribution. Other parameters of the hierarchical model are treated as constants with respect to  $\theta_{ia}$ .

This derivation gives a general version of a full conditional distribution for county  $i$  and age group  $a$ , we obtain 201 posterior distributions in our analysis.

$$\begin{aligned} p(\theta_{ia}|\cdot) &\propto \exp(-n_{ia} - \exp(\theta_{ia})) \times (n_{ia} \exp(\theta_{ia}))^{Y_{ia}} \times (Y_{ia}!)^{-1} \times \\ &\quad (2\pi\sigma_a^2)^{-1} \exp(-\frac{1}{2\sigma_a^2}(\theta_{ia} - \beta_{0a})^2) \\ &\propto \exp(-n_{ia} \exp(\theta_{ia}) + \theta_{ia}(Y_{ia} - \frac{1}{2\sigma_a^2}\theta_{ia} + \frac{\beta_{0a}}{\sigma_a^2})) \end{aligned}$$

I do not recognize this kernel as a known distribution, so we are not able to use Gibbs sampling technique to obtain posterior distribution of  $\theta_{ia}$ . We will use Metropolis algorithm to obtain samples for the posterior distribution. More on this later, but we will use Metropolis update because we assume a symmetric candidate density for  $\theta_{ia}$

### Full conditional for $\sigma_a^2$

- Term  $\sigma_a^2$  appears in the prior distribution of random effects  $\theta_{ia}$ . All other models terms are treated as constants with respect to  $\sigma_a^2$ , and will make up a normalizing constant for the proper density of the full conditional distribution.
- Because  $\sigma_a^2$  appears in each of 67  $\theta_{ia}$  for a given age group  $a$ , we need to use a joint likelihood of random effects. Assuming independence of  $z_{ia}$ , we can obtain joint prior distribution as a product of 67 prior densities.
- Similar to  $\beta_{0a}$ , we derive a general form of the full conditional distribution for this parameter. In our analysis, we will obtain three posterior distributions for  $\sigma_a^2$  for each age group.
- Inverse Gamma is a conjugate prior to a normal distribution, so we should expect to obtain an Inverse Gamma posterior distribution. We will therefore use Gibbs sampling to get samples of  $\sigma_a^2$

Full conditional:

$$\begin{aligned}
p(\sigma_a^2 | \cdot) &\propto \prod_{i=1}^{N=67} \frac{1}{\sqrt{2\pi\sigma_a^2}} \exp\left(-\frac{1}{2\sigma_a^2} \sum_i (\theta_{ia} - \beta_{0a})^2\right) \times (\sigma_a^2)^{-\alpha-1} \exp(-\beta/\sigma_a^2) \\
&\propto (\sigma_a^2)^{-n/2-a-1} \times \exp\left(-\left(b + \frac{1}{2} \sum_i (\theta_{ia} - \beta_{0a})^2\right) * 1/\sigma_a^2\right)
\end{aligned}$$

Full conditional for  $\sigma_a^2$  is proportional to the kernel of the Inverse Gamma with parameters  $\alpha = n/2 + a$  and  $\beta = b + \frac{1}{2} \sum_i (\theta_{ia} - \beta_{0a})^2$

### Problem 3

Code to fit the model and obtain posterior distributions for parameters of interest is attached in the appendix.

To fit the model, I used the following parameters and candidate densities:

- Assume a symmetric candidate density  $\beta_0 \sim \text{Norm}(\beta_0^{\backslash}, q)$  where  $q = 1$
- Assume a symmetric candidate density  $\theta_{ia} \sim \text{Norm}(\theta_{ia}^{\backslash}, q_{ia})$  where  $q_{ia} = 1$

#### Ratio for Updates $\beta_{0a}$

Using full conditional for  $\beta_{0a}$ , and omitting candidate density due to symmetrical property, we obtain the following equation to test each proposed sample for acceptance/rejection:

$$r(\beta_{0a}^*) = \exp\left(-\frac{1}{2\sigma_a^2} \left[\sum_i (\theta_{ia} - \beta_{0a}^*)^2 - \sum_i (\theta_{ia} - \beta_{0a}^{\backslash})^2\right]\right) \times \exp\left(-\frac{1}{2\tau_a^2} ((\beta_{0a}^*)^2 - (\beta_{0a}^{\backslash})^2)\right)$$

#### Ratio for Updates $\theta_{ia}$

Using full conditional for  $\theta_{ia}$ , and omitting candidate density due to symmetrical property, we obtain the following equation to test each proposed sample for acceptance/rejection. Recall that each  $\theta_{ia}$  is specific to the county-age-group level, so we do not aggregate the data, like with the  $\beta_{0a}$  example.

$$\begin{aligned}
r(\theta_{ia}^*) &= \exp(-n_{ia}[\exp(\theta_{ia}^*) - \exp(\theta_{ia}^{\backslash})]) \times \exp(Y_{ia}(\theta_{ia}^* - \theta_{ia}^{\backslash})) \times \\
&\quad \exp\left(-\frac{1}{2\sigma_a^2} [(\theta_{ia}^* - \beta_{0a})^2 - (\theta_{ia}^{\backslash} - \beta_{0a})^2]\right)
\end{aligned}$$

**Ratios for Update  $\sigma_a^2$**  is not needed because we use Gibbs sampling

**Notes for the derivation** are attached at the end of the document after the code section

## Model Comparison

Figure 1 presents county-specific age-adjusted rates using regression with age as a predictor. Overall, the map looks similar to what we observed under the Poisson-Gamma model (HW3). Figure 2 shows results obtained in part 1, using mixed effects regression adjustment for age. The two maps appear very similar.

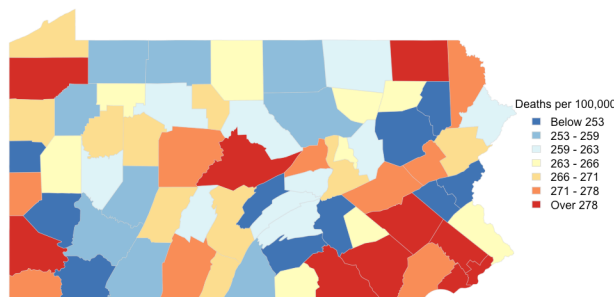


Figure 1: Final Map of Rates using Age-Adjusted Regression (Part 2)

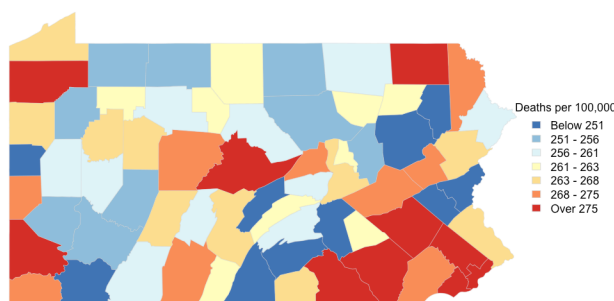


Figure 2: Final Map of Rates Using Mixed Effects Regression (Part 1)

Figure 4 presents posterior distribution for the average log-death rate for each group. Top row represents estimates obtained from part 1, and bottom row are part 2 estimates (this HW). Recall: part 2 is regression for age adjustment, and part 1 is regression for age adjustment with county specific random effects.

Overall, the two models produce similar posterior distributions for all  $\beta_{0a}$ . It is quite notable that posterior distributions from part 1 are less smooth. I am not sure why this happens, it could be due to the fact that the model is more complex. It also could be due to the fact that my algorithm are not tuned for optimal convergence. Perhaps, allowing for 100,000 replications of the loop will smooth out the distributions more. However, it seems that for

practical interpretation of the average death rate for the age group, both models produce identical results.

It seems that there might be minor skewness at the tails of varying degree for all of these posterior distributions. Perhaps, tuning sampling procedure to allow distributions to converge at a better and faster rate will result in better posterior distributions.

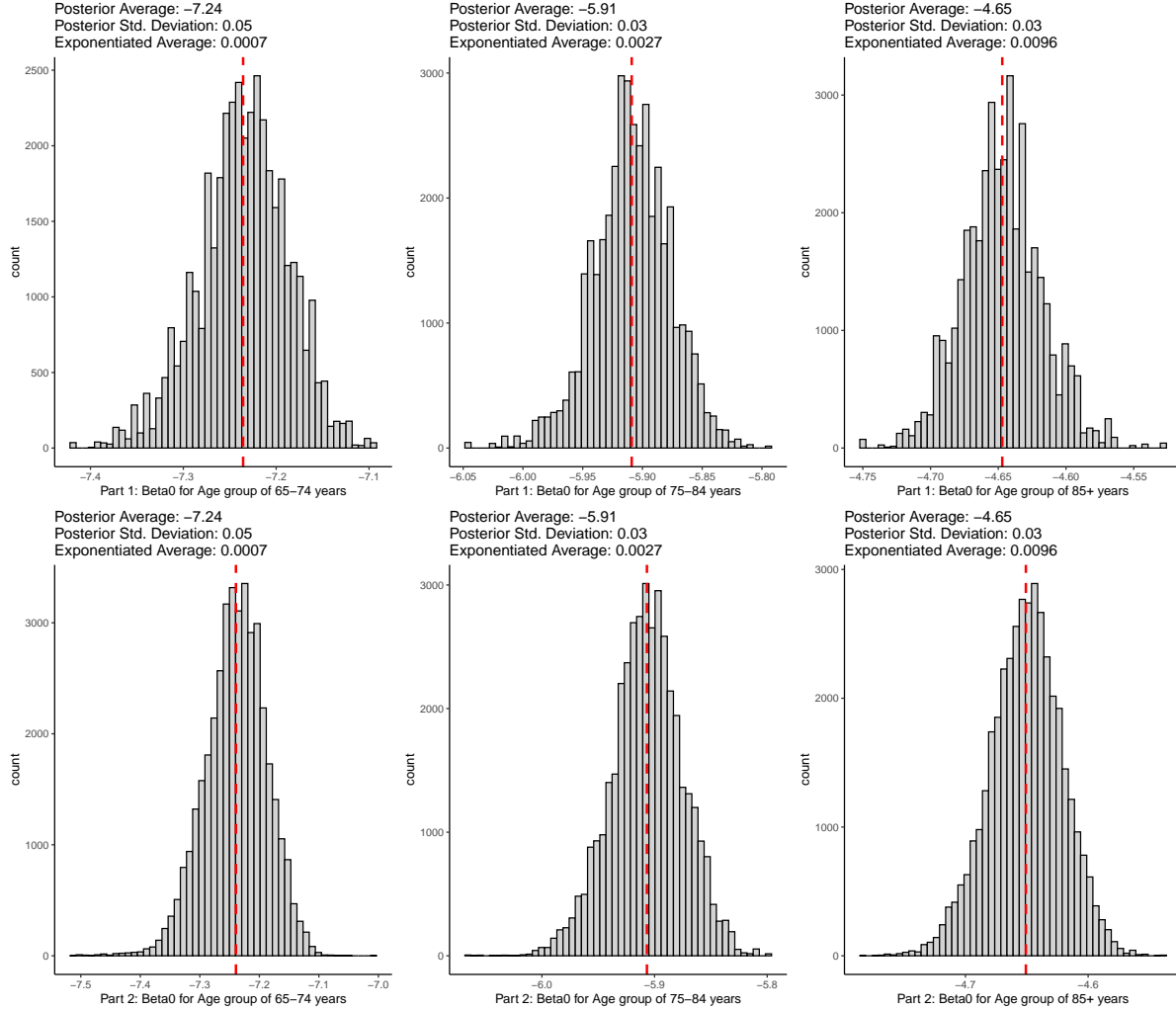


Figure 3: Posterior Distributions of Beta0 for the three age groups

Previously, we saw that the average rates of mortality at the county level between method 1 and 2 were quite similar. Figure 4 visualizes correlation between method 1 and method 2 estimates for each county.

The two methods produce very similar estimates. County level estimates are highly correlated, and have a 1-1 correspondence.

It appears that the choice of the model does not matter as much, as they both result in identical estimates of stroke related mortality at the county level

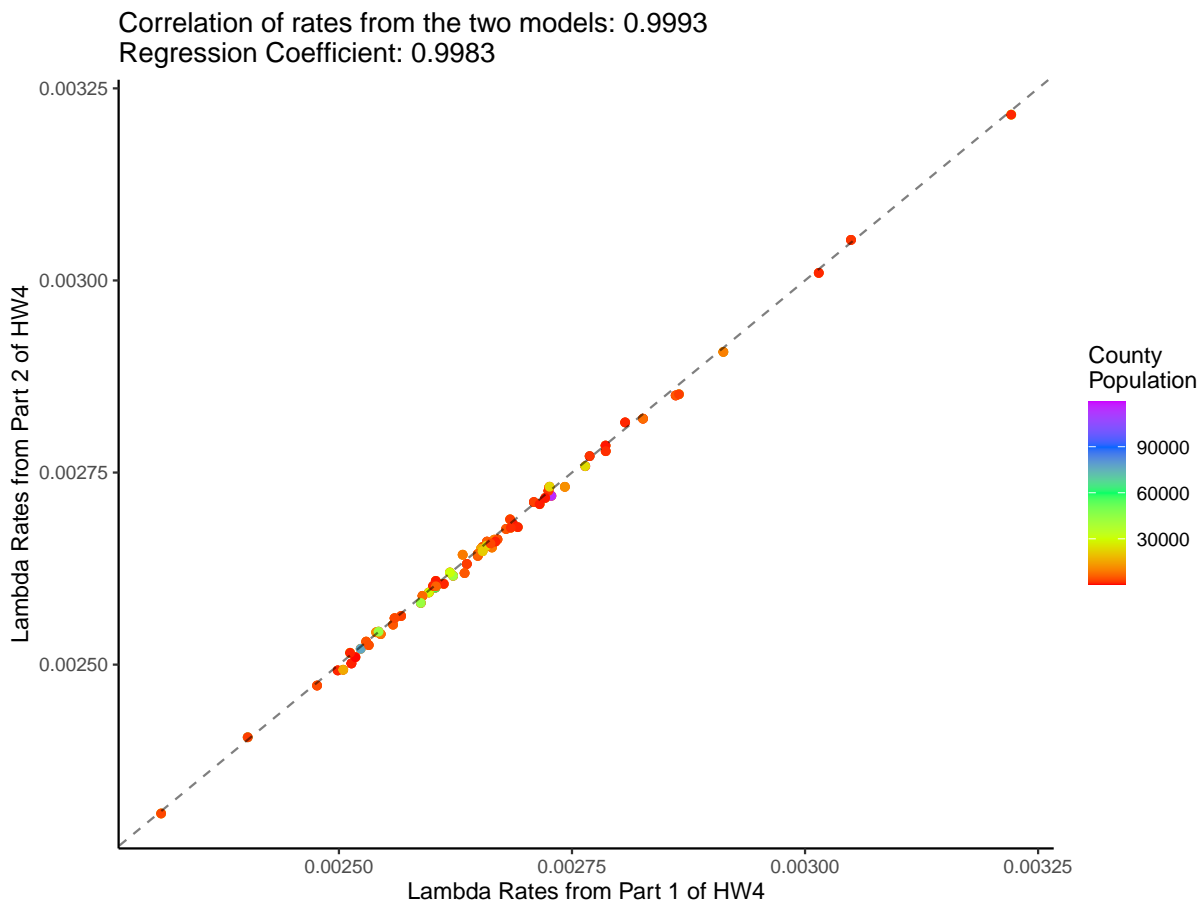


Figure 4: Lambda compariosns

## Appendix

### Metropolis Hastings Code

```
set.seed(1234)

reps = 50000

lambda_ia <-
  cbind(
    stroke_clean %>% select(lambda_0) %>% unlist(),
    matrix(data = NA,
           nrow = stroke_clean %>% select(lambda_0) %>% unlist() %>% length(),
           ncol = (reps-1)
          )
  )

## get guesses for beta_0a as the group average from data
# first, if there are missing values, impute with prior guess for lambda_0

stroke_clean %>%
  mutate(final_y = ifelse(is.na(deaths), lambda_0 * population, deaths),
         log_rate = log(final_y/population)
        ) %>%
  group_by(age.group) %>%
  summarize(b0a = mean(log_rate)) %>%
  ungroup() %>%
  select(b0a) %>%
  unlist() -> boa_guess

beta_0a <-
  cbind(
    boa_guess,
    matrix(data = NA,
           nrow = 3,
           ncol = (reps-1)
          )
  ) ## these are some pretty bad guesses for the betas, but it will work for now

## get initial guesses for theta_ia as the log of the prior guess of mortality rate for
## a corresponding age group
theta_ai_guess <- log(stroke_clean$lambda_0)
```



```

theta_ia <-
  cbind(
    theta_ai_guess,
    matrix(data = NA,
           nrow = length(theta_ai_guess),
           ncol = (reps-1)
          )
  )

sigma_0a <-
  cbind(
    c(1,1,1),
    matrix(data = NA,
           nrow = 3,
           ncol = (reps-1)
          )
  )## initial guesses for the variance of theta_ia distribution

####
# parameters for the prior distributions
tau2 = 10000
a = 0.001
b = 0.001

q_norm_b = 0.075 # variance of normal candidate densities
q_norm_theta_ai = 1

n = 67 # number of counties in the analysis

for(i in 2:reps){

  if(i %% 1000 == 0){print(i)}
  #####
  # DATA IMPUTATION STEP
  #####

  lambda_ia[, (i-1)] * stroke_clean$population -> poisson_lambdas_iter

  ppois(9.5, poisson_lambdas_iter) -> limits_detection_iter

  # using these numbers between 0 and somewhere less than 1, sample from uniform distribut

```

```

runif(n = length(limits_detection_iter), min = 0, max = limits_detection_iter) -> sample

# get imputed values by putting unifrom random samples into 'inverse' CDF
qpois(sampled_u, lambda = poisson_lambdas_iter) -> imp

# get final imputed vector of the observed data
stroke_clean$deaths -> final_ys_iter
final_ys_iter[which(is.na(final_ys_iter))] <- imp[which(is.na(final_ys_iter))]

#####
# UPDATE sigma_a

# sample new sigma from the candidate density
## sample new values of sigma from posterior
for(SIGMA in 1:3){

  # s_prop = sig_proposed[SIGMA]
  # s_curr = sigma_0a[, (i-1)] [SIGMA]
  # identify what rows of random effects to grab
  theta_ia_rows <- seq(from = SIGMA,
                      to = length(final_ys_iter) - (3- SIGMA),
                      length.out = 67)

  # data for ratio
  theta_ia[theta_ia_rows, (i-1)] -> thetas
  beta <- beta_0a[SIGMA, (i-1)]
  sq_sums <- sum((thetas - beta)^2)

  sigma_0a[, (i)] [SIGMA] = 1 / rgamma(n = 1, n/2 + a, sq_sums/2+b )

}

#####
# UPDATE B_0a
for(POP in 1:3){

  most_recent_beta0a <- beta_0a[POP, (i-1)]
  sampled_beta0a <- rnorm(n = 1, mean = most_recent_beta0a, sd = q_norm_b)

  POP_rows <- seq(from = POP,
                  to = length(final_ys_iter) - (3- POP),

```

```

length.out = 67)

thetas_ipop = theta_ia[POP_rows, (i-1)]

sum_sq_w_prop = sum((thetas_ipop-sampled_beta0a)^2)
sum_sq_w_curr = sum((thetas_ipop-most_recent_beta0a)^2)

sigma_calc = sigma_0a[POP, i]

ratio <-
  -1/(2 * sigma_calc) * (sum_sq_w_prop - sum_sq_w_curr) -
  1/(2*tau2) * (sampled_beta0a^2 - most_recent_beta0a^2)

beta_0a[POP, (i)] <- ifelse(exp(ratio) > runif(n=1), sampled_beta0a, most_recent_beta0a)
}

#####
# UPDATE THETA_ia

b_0a = beta_0a[, (i)]
b_0a_calc = rep(b_0a, n)

sig2 = sigma_0a[, (i)]
sig2_calc = rep(sig2, n)

theta_ia_curr = theta_ia[, (i-1)]

theta_ia_prop = rnorm(n = n*3, mean = theta_ia_curr, sd = q_norm_theta_ai)

ns = stroke_clean$population

ratio <-
  -ns*(exp(theta_ia_prop) - exp(theta_ia_curr)) +
  final_ys_iter * (theta_ia_prop - theta_ia_curr) -
  1/(2 * sig2_calc) * (
    (theta_ia_prop - b_0a_calc)^2 - (theta_ia_curr - b_0a_calc)^2
  )

```

```

theta_ia[,i] <- ifelse(exp(ratio) > runif(n = length(ratio)), theta_ia_prop, theta_ia_cu

## update lambda based on theta

lambda_ia[, (i)] = exp(theta_ia[,i])
}

res <- list(lambda_ia,
            sigma_0a,
            theta_ia,
            beta_0a
            )

names(res) <- c("lambdas", "sigmas", "thetas", "betas")

write_rds(x = res,
          file = "./MHresults/MH results.rds"
          )

```