Denis Ostroushko - PUBH 7440 - HW4 - Part 1

Problem 1

Prerequisites

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

We have 67 counties, $i = 1, 2, ..., 67$, 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 $log\lambda_{ia} = \beta_{0a} + z_{ia}$.

So, the death rate for each county and age group is some function of an average effect for a given age group and an age-group-and-county specific random effect.

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

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

As mentioned previously, β_{0a} and z_{ia} are random variable {because Bayesian Analysis framework}, and therefore they have prior distributions:

$\beta_{0a}|\mu = 0, \tau_a^2 \sim N(0, \tau_a^2)$, where $\tau_a^2 = 10,000$. This equation represents three prior distributions for each age group subject to analysis. They all have identical prior distributions.

$z_{ia}|\mu = 0, \sigma_a^2 \sim N(0, \sigma_a^2)$, where σ_a^2 is also a random variable that has it's own prior distribution. Note that each county and age group (201 total data points) each have their own random effect. But, within an age group a, all random variables $z_{i,a=a}$ have the the same prior distribution with variance $\sigma_{a=a}^2$

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

Suppressed values of deaths with county and age-groups levels

- Note: I am reusing the description of the imputation procedure given in HW3, only changing max value from 10 to 9

In order to impute missing/suppressed values of $Y_{i\alpha}$ we need to use a truncated left tail of a poisson distribution with corresponding parameter $n_{i\alpha}\lambda_{i\alpha}$. We will set a maximum value at the tail equal to 9, meaning that for our imputations we will be sampling integers from 0 to 9 from poisson distributions. In order to do that, we follow these steps:

1. For each county for each group age, determine a parameter for the poisson distribution, refer to it as $\Lambda_{i\alpha}$.
2. For each county for each age group, determine quantile corresponding to value of 10 under $\Lambda_{i\alpha}$, call this quantile q

- use `ppois()` to get this quantile

3. Sample a number from a uniform distribution between 0 and q. This will be between 0 and some number less than or equal to 1 always.

- use `runif(n=1, min = 0, max = .)`

4. Using inverse CDF of a poisson distribution with parameter $\Lambda_{i\alpha}$, obtain a value corresponding to a randomly sampled quantile

- use `qpois()` for this step

5. Impute missing value with sampled values between 0 and 9

Full hirerachical model

$$
\begin{aligned}
p(\beta_{0a}, z_{ia}, \sigma_{0a}^2|\mathbf{Y}) \propto & \Pi_{i,a}\left[Pois(Y_{ia}|n_{ia} * exp(\beta_{0a} + z_{ia}))\right] && \textit{full data likelihood} \\
& Norm(\beta_{0a}|0, \tau_a^2) && \textit{prior for } \beta_{0a} \\
& Norm(z_{ia}|0, \sigma_a^2) && \textit{prior for } z_{ia} \\
& IG(\sigma_a^2|0.001, 0.001) && \textit{prior for } \sigma_a^2
\end{aligned}
$$

Problem 2

Full conditional for β_{0a}

Full conditional for z_{ia}

Full conditional for σ_a^2

Problem 3

Code to fit the model and obtain posterior distributions for parameters of interest is attached in the appendix after comparison with the HW3 results.

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

- Assume a symmetric candidate density $\beta_0 \sim Norm(\beta_0^{`}, q)$ where $q = 0.075$
- Assume a symmetric candidate density $z_{ia} \sim Norm(z_{ia}^{`}, q_{ia})$ where q_{ia} is proportional to the data-driven point estimate for the county-specific effect on the observed log-rate of stroke related mortality rate
- Assume an asymmetric candidate density $\sigma_a^2 \sim IG(q, q * \sigma_a^{`2})$, where q = 3 and $\sigma_a^{`2}$ is the most recent updated value of σ_a^2 from the Metropolis-Hastings iteration

Results for β_{0a}

Figure 1 shows posterior distributions for the age-group overall effect on deaths associated with stroke. Since $\log \lambda_{ia} = \beta_{0a} + z_{ia}$, presented values are on the logarithmic scale. We can make an observation that as overall age increases, the age-group overall 'average' death rate increases, which is something that we would expect to observe.

All posterior distributions have a nice symmetric shape, fitting a normal candidate distribution.

Figure 2 presents county-specific age-adjusted rates. Overall, the map looks similar to what we observed under the Poisson-Gamma model (HW3).

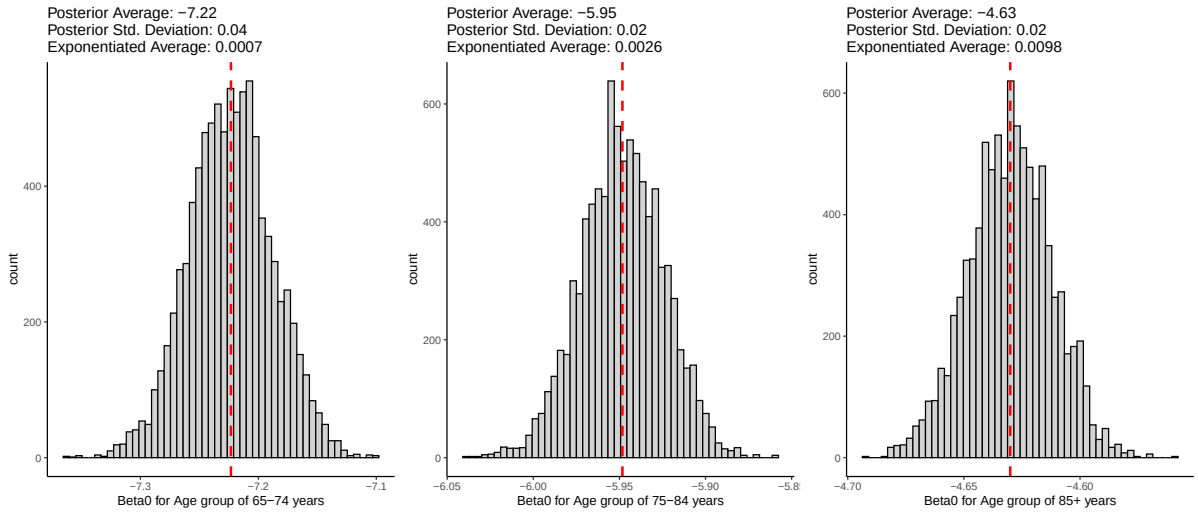

Figure 1: Posterior Distribtuions of Beta0 for the three age groups

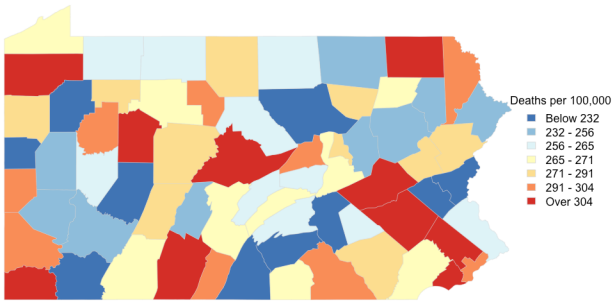

Figure 2: Final Map of Rates

Appendix

Comparison with Poisson-Gamma model

Figure 3 shows the differences for the county specific estimates between the two approaches. It it evident that under my analysis, mixed-effects regression model tends to estimate much higher stroke related mortality rates for counties with smaller population size.

In some cases the differences are as high as 20%. I suspect that the primary difference between the results are due to the use of random effects.

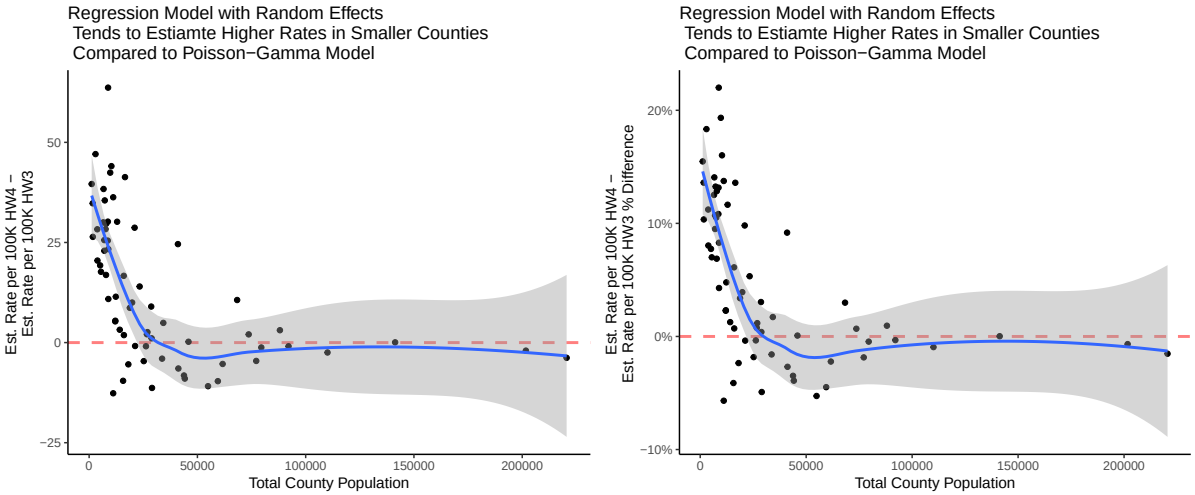

Figure 3: Comparion of Age Adjusted Rates Uisng Mixed-Effects Regression Model (HW4) and Poisson-Gamma Model (HW3)

Metropolis-Hastings Sampling Algorithm R-code

```
set.seed(182)

reps = 10000

lambda_ia <-  ## this is a 201 x 10,000 matrix that will
              ## track updates for each county-age group Lambda
              ## each row has 10,000 entries -
              ## to collect 10,000 metropolis updates
              ## I initiated algorithm with lambda =
              ## 75, 250, 1000 death per 100,000 people
  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_oa as the group average from data
# first, if there are missing values, impute with prior guess for lambda0

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 ## my guess for Beta 0 is data driven

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 z_ia as the difference between observed Y minus age_group avera
# first, if there are missing values, impute with prior guess for lambda0

stroke_clean %>%
  mutate(final_y = ifelse(is.na(deaths), lambda_0 * population, deaths),
         log_rate = log(final_y/population)
         ) %>%
  group_by(age.group) %>%
  mutate(b0a = mean(log_rate)) %>%
  ungroup() %>%
  mutate(z_ia = log_rate -b0a) %>%
  select(z_ia) %>%
  unlist() -> zi_guess ## initial values for random effects z_ia are also data driven
```

```r
z_ia <-
  cbind(
    zi_guess,
    matrix(data = NA,
           nrow =   length(zi_guess),
           ncol = (reps-1)
             )
  )

sigma_0a <-
  cbind(
    c(10,10,10),
    matrix(data = NA,
           nrow =   3,
           ncol = (reps-1)
             )
  )## iniitiate sigma 2 for three age groups at 1,1,1. Not sure if these values even matte

tau2 = 10000
a = 0.001
b = 0.001

q_norm_b  = 0.05
# q_norm_zi = 0.005
q_norm_zi = abs(zi_guess)*5 # make it such that the step size for each z_ia is poportional
#                             ratio is 1/1
q_ig = 1

n = 67

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
```

```r
# 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
sig_proposed = 1/rgamma(n = 3, q_ig, q_ig * sigma_0a[,(i-1)]) # values 1,2,3 correspond
        ## to young, mid, old age groups

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
  z_rows <- seq(from = SIGMA,
                to = length(final_ys_iter) - (3- SIGMA),
                length.out = 67)

  # data for ratio
  z_ia[z_rows, (i-1)] -> random_effs

  (s_prop/s_curr)^(2*q_ig - a - n/2) *

    exp(-1/2 * sum(random_effs^2) * (1/s_prop - 1/s_curr)) *

    exp(-b * ((1/s_prop - 1/s_curr))) *

    exp(q_ig * (s_prop/s_curr - s_curr/s_prop)) -> ratio

  sigma_0a[,(i)][SIGMA] <- ifelse(ratio > runif(1), s_prop, s_curr)

}

############
```

```
# UPDATE Z_ia

b_0a = beta_0a[,(i-1)]
b_0a_calc = rep(b_0a, n)

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

z_ia_curr = z_ia[,(i-1)]
z_ia_prop = rnorm(n = n*3, mean = z_ia_curr, sd = q_norm_zi)

(-stroke_clean$population *
      (exp(b_0a_calc + z_ia_prop) + exp(b_0a_calc +z_ia_curr ))
    ) +

  (final_ys_iter*(z_ia_prop - z_ia_curr)) +

  (-1/(2 * sig2_calc) * (z_ia_prop^2 - z_ia_curr^2)) -> ratio

z_ia[,i] <- ifelse(exp(ratio) > runif(n = length(ratio)), z_ia_prop, z_ia_curr)
############
# 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)

  Y_ipop = final_ys_iter[POP_rows]
  n_ipo = stroke_clean$population[POP_rows]
  z_ia_calc = z_ia[,i][POP_rows]

  ratio  <-
    (sum(Y_ipop) * (sampled_beta0a - most_recent_beta0a)) +

    (sum(n_ipo * exp(z_ia_calc)) * (exp(most_recent_beta0a) - exp(sampled_beta0a)) ) +
```

```
      (-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 lambda based on beta and zeta
  beta_0a_for_lambda = rep(beta_0a[,i], n)

  lambda_ia[,(i)] = exp(beta_0a_for_lambda + z_ia[,i])

}

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

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

write_rds(x = res,
          file = "./MHresults/MH results7.rds"
          )
```