STAT40850 - Assignment 4 - Intro to Bayesian Analysis

Conor Ryan - 16340116

2023-03-28

Introduction

In this assignment we wish to develop beta-binomial models for a data set from 2015 which records the number of heart attacks and number of resulting deaths from 13 hospitals in Manhattan, New York. The data set is collated by the New York State Department of Health. The aim of this assignment is to investigate whether a hierarchical, separate or pooled model for the probability of death from heart attacks in each of the 13 hospitals, best describes the data.

```
# Reading in NYC heart attack data from excel file
data <- read_excel("NYC_heart_attack.xlsx") %>%
  arrange(desc(`death%`))

# Producing a table of the NYC heart attack data
kable(data,
      caption = "Heart Attacks and Resulting Deaths in Manhattan Hospitals - 2015",
      row.names = TRUE) %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Heart Attacks and Resulting Deaths in Manhattan Hospitals - 2015

	hospital	cases	deaths	death%
1	Mount Sinai Roosevelt	46	6	13.043478
2	NYP Hospital - Allen Hospital	105	13	12.380952
3	NYP/Lower Manhattan Hospital	41	4	9.756098
4	Metropolitan Hospital Center	84	7	8.333333
5	Mount Sinai Beth Israel	291	24	8.247423
6	Lenox Hill Hospital	228	18	7.894737
7	NYP Hospital - Columbia Presbyterian Center	353	25	7.082153
8	Mount Sinai St. Luke's	293	19	6.484642
9	NYU Hospitals Center	241	15	6.224066
10	Mount Sinai Hospital	270	16	5.925926
11	NYP Hospital - New York Weill Cornell Center	250	11	4.400000
12	Bellevue Hospital Center	129	4	3.100775
13	Harlem Hospital Center	35	1	2.857143

We will employ a binomial model treating the cases of heart attacks as 'trials' and the number of resulting deaths as 'successes'. Observing the outputted table of the data set above, we can see significant variation in the death rates (death%) in each of the 13 hospitals.

A *pooled model* will not allow for any variation in the death rate between the 13 hospitals and as a result, I don't expect it to fit the data well. On the other extreme, a *separate model* assumes a different prior

probability of death for each model. This essentially assumes that knowing the death rate in another Manhattan hospital does not inform the death rate in the current hospital. This does not seem like a reasonable assumption and as a result I would also expect this model to perform poorly out-of-sample.

Finally, a *hierarchical model* offers a compromise between the above two models, where the prior probability of death for each hospital are 'connected' via a distribution parameterised by hyperparameter(s) which means all θ_is are sampled from the same distribution but still allow variability for each hospital. I will outline below, the form of the hierarchical model that I will employ to fit this data.

Let y_i denote the observed number of deaths and n_i the observed number of heart attacks at hospital i. Let θ_i denote the probability of death at hospital i.

$$\begin{aligned} y_i &\sim Bin(n_i, \theta_i) \\ \theta_i &\sim Beta(\alpha, \beta) \end{aligned}$$

Where α and β are hyperparameters that are shared between hospitals.

The values for α and β have strong influence and to mitigate this we specify prior distributions for (α, β). We use the fact that we can interpret the parameters (α, β) of the beta distribution as follows;

1. α corresponds to the number of prior 'successes'.
2. β corresponds to the number of prior 'failures'.

We will place prior distributions on transformations of (α, β) where;

1. $\mu = \frac{\alpha}{(\alpha+\beta)}$, and can be interpreted as a prior mean for θ_i and takes values in $[0, 1]$.
2. $\eta = \alpha + \beta$, and can be interpreted as a prior sample size and takes the value of positive real numbers.

We now place prior distributions on μ and η as follows;

$$\begin{aligned} \mu &\sim Beta(a, b) \\ \eta &\sim Exp(c) \end{aligned}$$

Where values for a,b,c are chosen as 5, 45 and 0.01 and c represents the rate parameter of the $Exp()$ distribution. This results in a hierarchical prior for θ_i.

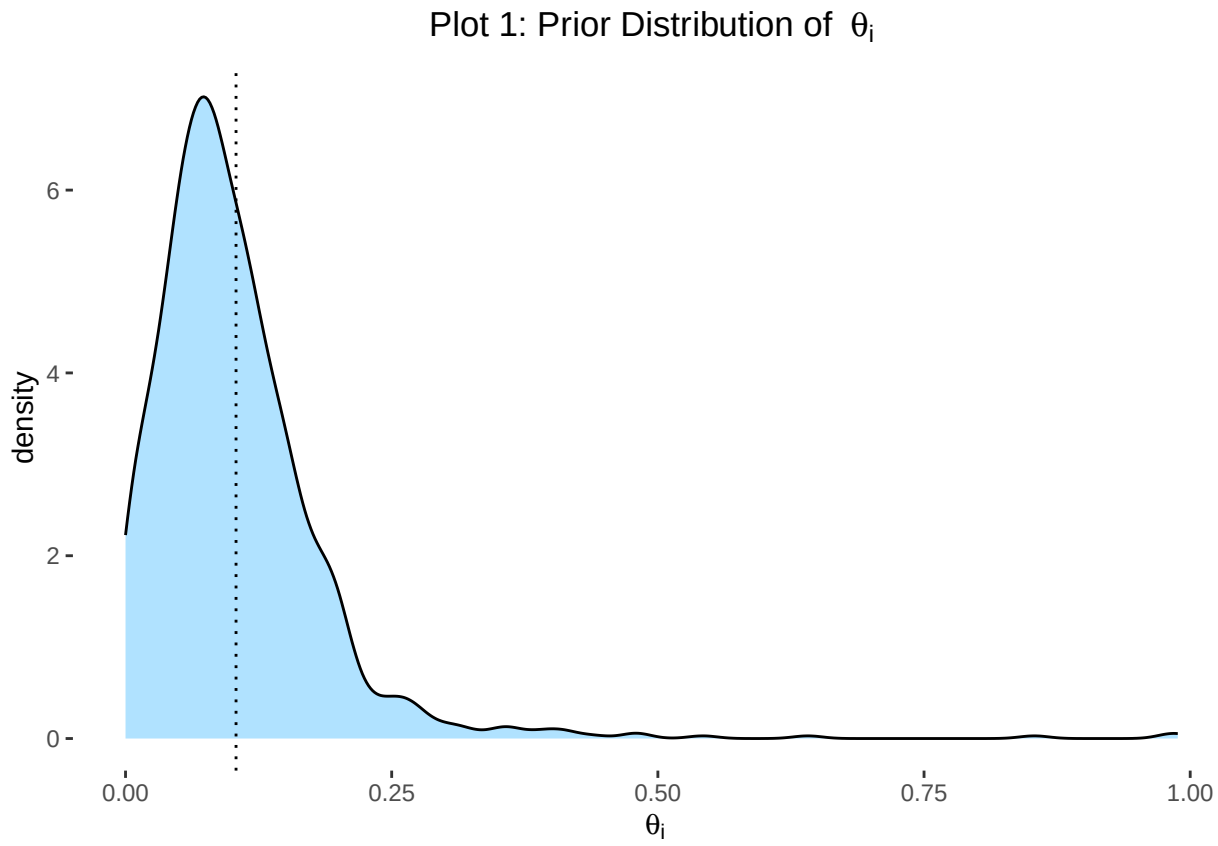
In summary, the hierarchical model can be represented mathematically as follows;

$$\begin{aligned} y_i &\sim Bin(n_i, \theta_i) \\ \theta_i &\sim Beta(\alpha, \beta) \\ \mu &\sim Beta(a, b) \\ \eta &\sim Exp(c) \\ \alpha &= \eta\mu \\ \beta &= (1-\mu)\eta \end{aligned}$$

Question 1

In question 1, I will analyse the prior distribution of θ_i by generating a sample of θ_i from the hierarchical prior $Beta(\alpha, \beta)$. I will first generate a sample of μ and η values from their respective prior distributions and use these to produce values of α and β to generate a sample of θ_i values.

```r
# generating samples of mu and eta values mu ~ Beta(5,45), eta ~ Exp(0.01)
sample_mu <- rbeta(1000,5,45)
sample_eta <- rexp(1000,0.01)

# calculating sample alpha and beta values from generated mu and eta samples
sample_alpha <- sample_eta * sample_mu
sample_beta <- (1-sample_mu) * sample_eta

# generating sample of theta_is based on generated samples above
prior_theta <- rbeta(1000, sample_alpha, sample_beta)

# estimating probability that theta_i > 0.2
prob_theta <- sum(prior_theta > 0.2) / length(prior_theta)

# plotting prior density of theta_i
ggplot() +
  geom_density(aes(prior_theta), fill = 'lightskyblue1') +
  # Add line for mean prior mean of theta
  geom_vline(xintercept = mean(prior_theta), linetype='dotted') +
  labs(x = expression(~theta[i]),
       title = expression("Plot 1: Prior Distribution of "~theta[i])) +
  theme(plot.title = element_text(hjust = 0.5))
```



We observe the prior density for θ_i above and can see it is centered on 0.1 with the majority (c.90%) of theta values being between 0 and 0.2.

We can use the sample values of θ_i to estimate prior probability that $\theta_i > 0.2$. We can observe the number of times that the sampled prior value of $\theta_i > 0.2$ and divide this by the total number of samples of θ_i and use this to estimate the probability $\theta_i > 0.2$.

Per the sampled values of θ_i, I estimate that the probability $\theta_i > 0.2$ is **0.073**.

Question 2

In question 2, I fit the hierarchical model described previously in Stan and provide a summary of the model output along with commentary of the results. See 'hier.stan' file for reference. I fit the model using 5,000 iterations and the default 4 chains and set a seed in order to ensure reproducible results.

```
# organising data into a list to be read into Stan model
dat <- list(H = nrow(data),
            N = data$cases,
            S = data$deaths)

# fitting the model using stan
fit_hier <- stan(file = 'hier.stan',data = dat,seed = 759,iter = 5000)
```

```
# print output from model
print(fit_hier, probs = c(0.05, 0.5, 0.95), pars = c("theta","alpha","beta","log_lik"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
##               mean se_mean     sd    5%    50%    95% n_eff Rhat
## theta[1]      0.09    0.00   0.02  0.06   0.08   0.13  7554    1
## theta[2]      0.09    0.00   0.02  0.06   0.09   0.13  7365    1
## theta[3]      0.08    0.00   0.02  0.05   0.08   0.12  8644    1
## theta[4]      0.08    0.00   0.02  0.05   0.07   0.11  9293    1
## theta[5]      0.08    0.00   0.01  0.06   0.08   0.10 10084    1
## theta[6]      0.08    0.00   0.01  0.06   0.08   0.10 10502    1
## theta[7]      0.07    0.00   0.01  0.05   0.07   0.09 10014    1
## theta[8]      0.07    0.00   0.01  0.05   0.07   0.09 10425    1
## theta[9]      0.07    0.00   0.01  0.05   0.07   0.09 10404    1
## theta[10]     0.06    0.00   0.01  0.05   0.06   0.09 11112    1
## theta[11]     0.06    0.00   0.01  0.04   0.06   0.08  8592    1
## theta[12]     0.05    0.00   0.01  0.03   0.05   0.08  8246    1
## theta[13]     0.06    0.00   0.02  0.04   0.06   0.10 10075    1
## alpha        14.48    0.13   8.51  4.71  12.67  30.87  4334    1
## beta        186.11    1.70 111.77 57.52 161.09 404.12  4317    1
## log_lik[1]   -2.46    0.01   0.59 -3.60  -2.34  -1.78  7946    1
## log_lik[2]   -3.01    0.01   0.82 -4.61  -2.80  -2.15  7079    1
## log_lik[3]   -1.83    0.00   0.31 -2.43  -1.72  -1.58  6213    1
## log_lik[4]   -2.09    0.00   0.33 -2.72  -1.96  -1.86  5658    1
## log_lik[5]   -2.84    0.01   0.53 -3.88  -2.64  -2.47  5210    1
## log_lik[6]   -2.66    0.01   0.48 -3.61  -2.48  -2.33  5127    1
## log_lik[7]   -2.85    0.01   0.52 -3.85  -2.66  -2.50  4745    1
## log_lik[8]   -2.71    0.01   0.48 -3.69  -2.52  -2.36  4810    1
## log_lik[9]   -2.59    0.01   0.47 -3.52  -2.41  -2.25  5702    1
## log_lik[10]  -2.65    0.01   0.52 -3.70  -2.46  -2.28  6558    1
```

```
## log_lik[11]  -2.73    0.01   0.73 -4.23  -2.48  -2.11  7930    1
## log_lik[12]  -2.59    0.01   0.84 -4.21  -2.40  -1.64  9264    1
## log_lik[13]  -1.51    0.00   0.40 -2.26  -1.44  -1.02 10700    1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 28 12:56:17 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

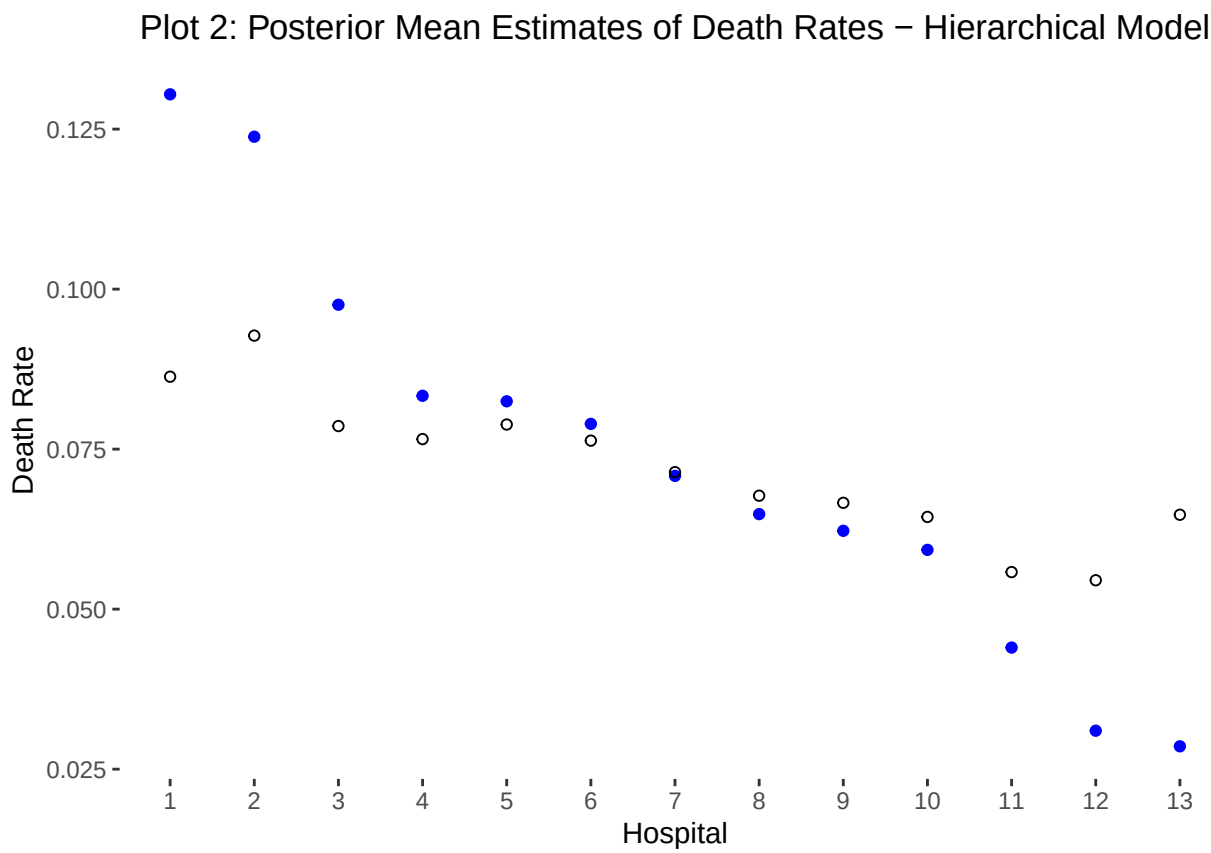
```
# extracting sample parameters of interest
posterior_hier <- as.data.frame(fit_hier,pars = c("theta","alpha","beta","log_lik"))
p_hier <- stack(posterior_hier[,1:13])
theta_hier <- as.data.frame(extract(fit_hier,pars = "theta"))
log_lik_hier <- extract(fit_hier)$log_lik

# calculating posterior mean estimates for death rate for each hospital
data$p_mean_hier <- apply(theta_hier,2,mean)
```

I have provided the output of the hierarchical model from stan above. We can see the mean, standard deviation and 90% credible intervals for each of the θ_i values for each hospital. We see the θ_1 for Mount Sinai Roosevelt has a mean of 0.09 with a 90% credible interval in [0.06, 0.13]. Mount Sinai Roosevelt has the highest death rate per the data (c.13%) but also a relatively low case size so we would expect the hierarchical model decrease it's posterior mean estimate for θ down significantly, which it has.

We also observe the α and β hyperparameter values. α has a mean of 14.48 and 90% support between [4.71, 30.87] and β has a mean of 186.11 and 90% support between [57.52, 404.12]. The log-likelihood values for each hospital are also included.

I will now plot the observed death rate for each hospital per the data and overlay the posterior mean estimates from the hierarchical model for the death rate of each model.

```
# plot of observed death rates with overlay of posterior mean
# death rates from the hierarchical model

# adding a row index for the x-axis values (names too large)
data <- data %>%
  mutate(Hospital_No = row_number())

ggplot(data = data) +
  geom_point(aes(factor(Hospital_No),`death%`/100), col = 'blue') +
  geom_point(aes(factor(Hospital_No), p_mean_hier), shape = 1, col = 'black') +
  labs(y = "Death Rate",
       x = "Hospital",
       title = "Plot 2: Posterior Mean Estimates of Death Rates - Hierarchical Model") +
  theme(plot.title = element_text(hjust = 0.5))
```

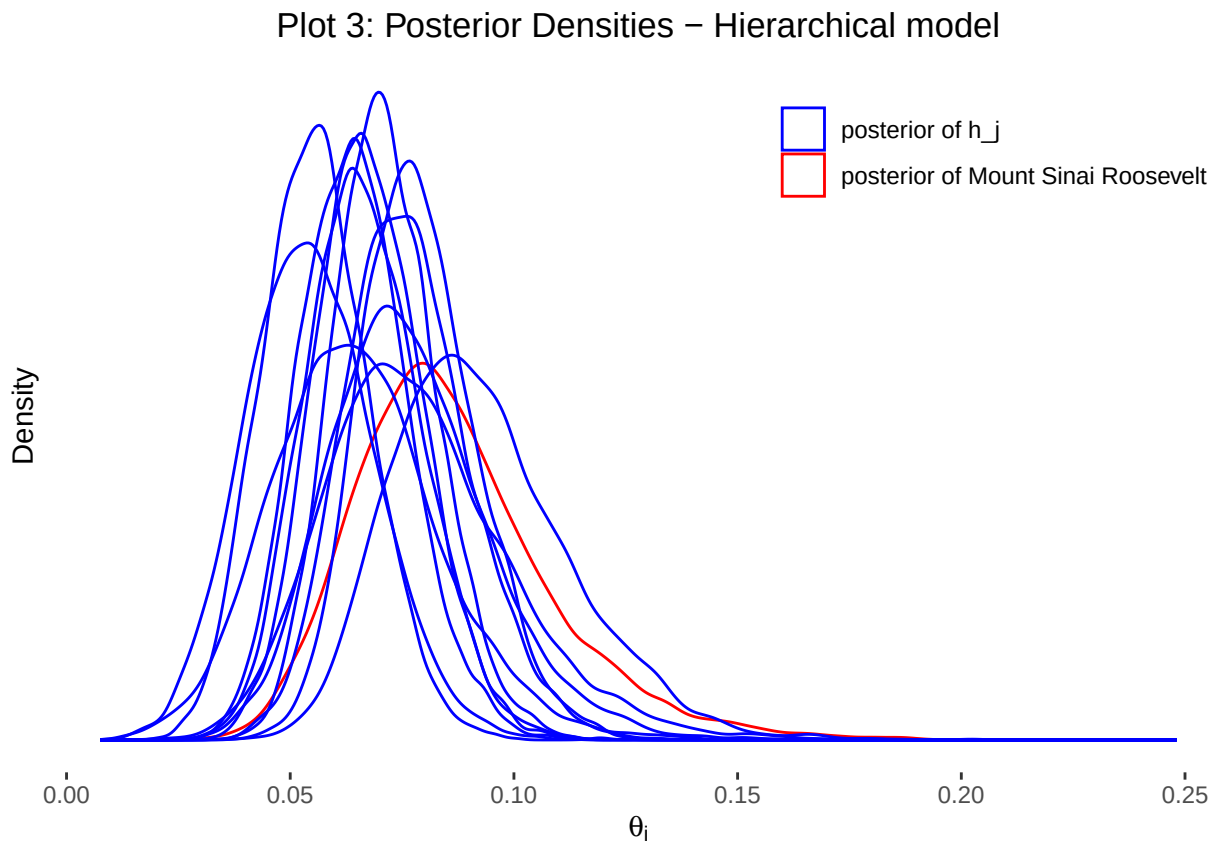


We observe the death rates per the data (blue dots) and the posterior mean estimates of the death rate per the model (black circles) in the above plot. We can see hospitals with larger observed death rates have their posterior mean estimates reduced by the hierarchical model. We also note that this 'pulling' effect is larger for hospitals with lower case numbers e.g. Mount Sinai Roosevelt (Hospital 1).

Similarly, hospitals with lower observed death rates have their posterior mean estimates increased by the hierarchical model due to the 'connected' nature of the theta values for each hospital specified in the model.

I will now plot the posterior densities of each hospital generated from the hierarchical model. I have highlighted Mount Sinai Roosevelt in red for reference.

```
# Plot Posterior Densities for each hospital
labs1 <- paste('posterior of', c('h_j', 'Mount Sinai Roosevelt'))
plot_hier <- ggplot(data = p_hier) +
  geom_density(aes(values, color = (ind=='theta[1]'), group = ind)) +
  labs(x = expression(theta[i]), y = 'Density', title = 'Plot 3: Posterior Densities - Hierarchical mod
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('blue','red'), labels = labs1) +
  theme(legend.background = element_blank(), legend.position = c(0.8,0.9),
        plot.title = element_text(hjust = 0.5))

plot_hier
```



We can see from the above posterior density plots that while each θ_i has it's own distribution, each distribution of θ is clumped together in the hierarchical model due to the shared hyperparameters (α, β) used to sample each θ_i value. Overall, we can see that most of the densities are centered between 0.05 to 0.09 which aligns with the model summary output from Stan.

Question 3

In question 3, I will fit a separate model where each hospital has it's own θ_i prior distribution. This means that we do not use information from other hospitals to inform the death rate of another hospital.

For the separate model, I have used the same prior distributions for μ and η as in the hierarchical model only. The difference with the separate model is that these hyperparameters are not common to each hospital i.e. each hospital has it's own μ and η for each sample thus creating separate α and β values for each hospital. The separate model can be represented mathematically as follows;

$$\begin{aligned} y_i &\sim Bin(n_i, \theta_i) \\ \theta_i &\sim Beta(\alpha_i, \beta_i) \\ \mu_i &\sim Beta(a, b) \\ \eta_i &\sim Exp(c) \\ \alpha_i &= \eta_i \mu_i \\ \beta_i &= (1 - \mu_i)\eta_i \end{aligned}$$

Where a = 5, b = 45 and c = 0.01.

I fit the separate model using 5,000 iterations and the default 4 chains and set a seed in order to ensure reproducible results, see separate.stan file for reference.

```
# fitting the model using stan
fit_sep <- stan(file = 'separate.stan',data = dat,seed = 759,iter = 5000)
```

```
# print output from model
print(fit_sep, probs = c(0.05, 0.5, 0.95), pars = c("theta","log_lik"))
```

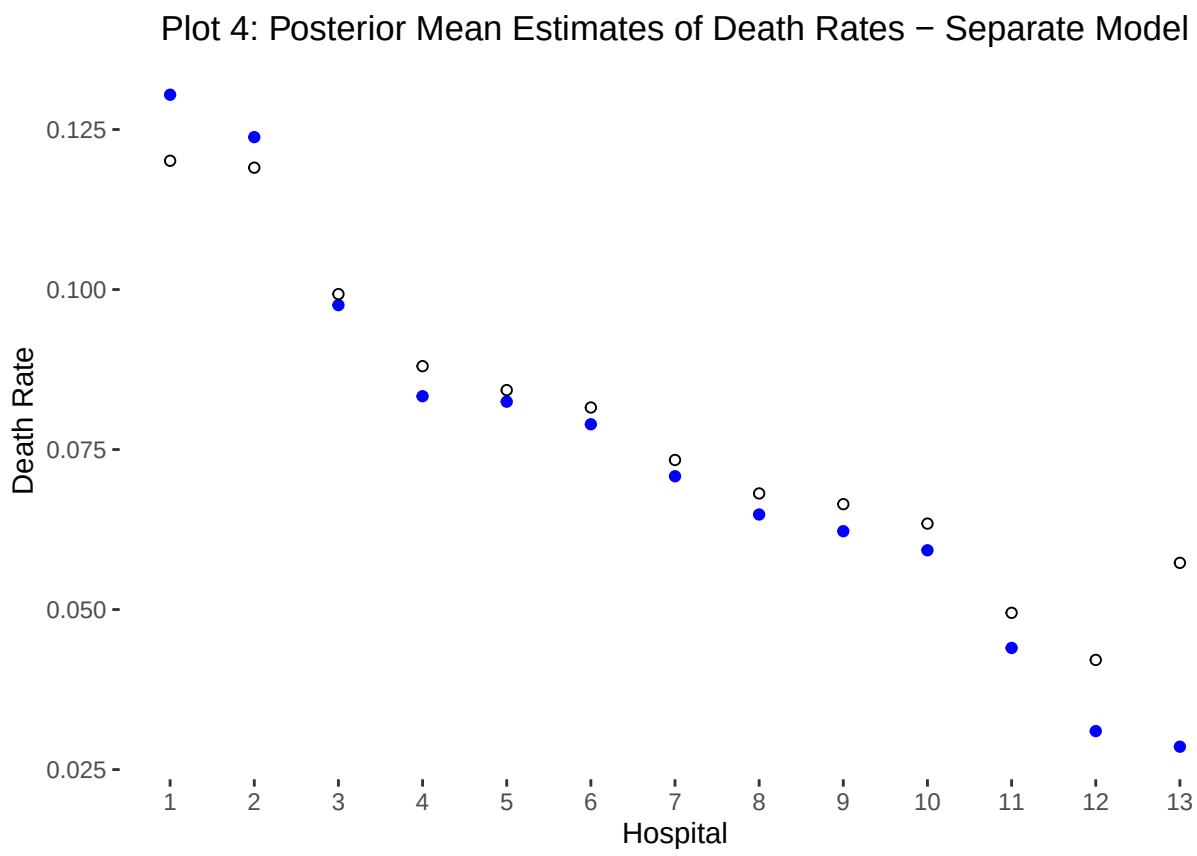
```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
##              mean se_mean   sd    5%   50%   95% n_eff Rhat
## theta[1]     0.12    0.00 0.04  0.06  0.12  0.19 11790    1
## theta[2]     0.12    0.00 0.03  0.08  0.12  0.17 12072    1
## theta[3]     0.10    0.00 0.04  0.05  0.10  0.16 11820    1
## theta[4]     0.09    0.00 0.03  0.05  0.09  0.14 11475    1
## theta[5]     0.08    0.00 0.02  0.06  0.08  0.11 14839    1
## theta[6]     0.08    0.00 0.02  0.06  0.08  0.11 15383    1
## theta[7]     0.07    0.00 0.01  0.05  0.07  0.10 16509    1
## theta[8]     0.07    0.00 0.01  0.05  0.07  0.09 15618    1
## theta[9]     0.07    0.00 0.02  0.04  0.07  0.09 14403    1
## theta[10]    0.06    0.00 0.01  0.04  0.06  0.09 13847    1
## theta[11]    0.05    0.00 0.01  0.03  0.05  0.07 14724    1
## theta[12]    0.04    0.00 0.02  0.02  0.04  0.07 13359    1
## theta[13]    0.06    0.00 0.03  0.01  0.05  0.11  9482    1
## log_lik[1]  -2.12    0.01 0.52 -3.15 -1.92 -1.76  5868    1
## log_lik[2]  -2.54    0.01 0.58 -3.68 -2.33 -2.14  5607    1
## log_lik[3]  -1.89    0.01 0.45 -2.78 -1.72 -1.58  5347    1
## log_lik[4]  -2.24    0.01 0.53 -3.32 -2.04 -1.86  5814    1
## log_lik[5]  -2.93    0.01 0.64 -4.24 -2.69 -2.47  4987    1
## log_lik[6]  -2.76    0.01 0.61 -4.00 -2.52 -2.33  4971    1
## log_lik[7]  -2.97    0.01 0.65 -4.31 -2.71 -2.50  5026    1
## log_lik[8]  -2.83    0.01 0.66 -4.17 -2.58 -2.36  5404    1
## log_lik[9]  -2.72    0.01 0.68 -4.06 -2.46 -2.25  5118    1
## log_lik[10] -2.75    0.01 0.66 -4.03 -2.50 -2.28  5676    1
## log_lik[11] -2.62    0.01 0.73 -4.08 -2.34 -2.11  5277    1
## log_lik[12] -2.19    0.01 0.77 -3.77 -1.89 -1.62  8266    1
## log_lik[13] -1.51    0.01 0.62 -2.75 -1.29 -0.99  9313    1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 28 12:56:42 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
# extracting sample parameters of interest
posterior_sep <- as.data.frame(fit_sep,pars = c("theta","log_lik"))
p_sep <- stack(posterior_sep[,1:13])
theta_sep <- as.data.frame(extract(fit_sep,pars = "theta"))
log_lik_sep <- extract(fit_sep)$log_lik

# calculating posterior mean estimates for death rate for each hospital
data$p_mean_sep <- apply(theta_sep,2,mean)
```

I have provided the output of the separate model from stan above. We can see the mean, standard deviation and 90% credible intervals for each of the θ_i values for each hospital. Comparing the above output to the output of the hierarchical model, we observe larger mean values for the θ_is in the separate model as there is no pooling effect between hospitals. This results in mean posterior estimates for the θ_is to be much closer to the observed death rates. The separate model is over-fitting the values for θ_i for each hospital and will likely result in a lower out-of-sample performance.

I will now plot the observed death rate for each hospital per the data and overlay the posterior mean estimates from the separate model for the death rate of each model. The difference between the mean posterior values for the separate model compared to the hierarchical model is easily observed when comparing plots 2 and 4.

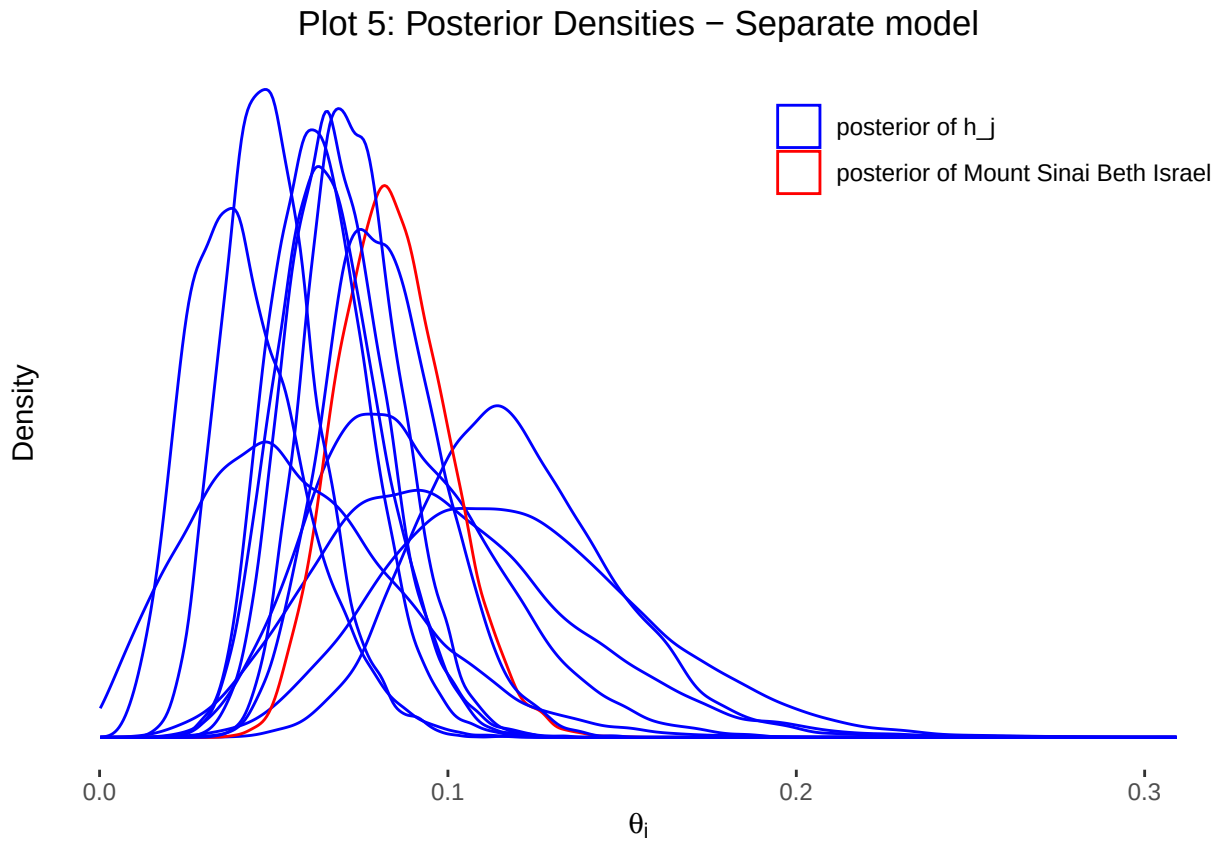
```
# plot of observed death rates with overlay of posterior mean
# death rates from the hierarchical model
ggplot(data = data) +
  geom_point(aes(factor(Hospital_No),`death%`/100), col = 'blue') +
  geom_point(aes(factor(Hospital_No), p_mean_sep), shape = 1, col = 'black') +
  labs(y = "Death Rate",
       x = "Hospital",
       title = "Plot 4: Posterior Mean Estimates of Death Rates - Separate Model") +
  theme(plot.title = element_text(hjust = 0.5))
```



As mentioned above, we can see that the posterior mean estimates of death rates for the separate model are much closer to the observed death rates due to the absence of a pooling effect in this model.

I will now plot the posterior densities of each hospital's death rate from the separate model.

```r
# Plot Posterior Densities for each hospital
labs1 <- paste('posterior of', c('h_j', 'Mount Sinai Beth Israel'))
plot_sep <- ggplot(data = p_sep) +
  geom_density(aes(values, color = (ind=='theta[5]'), group = ind)) +
  labs(x = expression(theta[i]), y = 'Density', title = 'Plot 5: Posterior Densities - Separate model',
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('blue','red'), labels = labs1) +
  theme(legend.background = element_blank(), legend.position = c(0.8,0.9),
        plot.title = element_text(hjust = 0.5))

plot_sep
```



Observing the above density plots of θ_i from the separate model, we can see much more variance between distributions than we did in the same plot produced for the hierarchical model. As mentioned before, this is due to the lack of pooling effect in the separate model. We can see that the densities are centered between values of around [0.05, 0.12].

Question 4

In question 4, I will fit a pooled model where each hospital has the same prior distribution for θ. In this model we do not allow any variation in the death rate between the 13 hospitals. This doesn't seem to be a sensible assumption as we observe fairly significant variation in the death rates between the 13 hospitals.

For the pooled model, I have chosen that the alpha and beta parameters be 5 and 45 respectively, reflecting the prior belief that the mean death rate is 0.1 (5/50). The pooled model can be represented mathematically as follows;

$$
\begin{aligned}
y_i &\sim Bin(n_i, \theta) \\
\theta &\sim Beta(5, 45)
\end{aligned}
$$

I fit the separate model using 5,000 iterations and the default 4 chains and set a seed in order to ensure reproducible results, see pooled.stan file for reference.

```
# fitting the model using stan
fit_pool <- stan(file = 'pooled.stan',data = dat,seed = 759,iter = 5000)
```

```
# print output from model
print(fit_pool, probs = c(0.05, 0.5, 0.95), pars = c("theta","log_lik"))
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=5000; warmup=2500; thin=1;
## post-warmup draws per chain=2500, total post-warmup draws=10000.
##
##              mean se_mean   sd    5%   50%   95% n_eff Rhat
## theta        0.07    0.00 0.01  0.06  0.07  0.08  3437    1
## log_lik[1]  -2.85    0.00 0.22 -3.24 -2.84 -2.50  3409    1
## log_lik[2]  -4.16    0.01 0.46 -4.96 -4.13 -3.45  3408    1
## log_lik[3]  -1.82    0.00 0.09 -1.99 -1.81 -1.68  3406    1
## log_lik[4]  -2.00    0.00 0.10 -2.19 -1.98 -1.88  3431    1
## log_lik[5]  -2.90    0.01 0.31 -3.51 -2.84 -2.51  3437    1
## log_lik[6]  -2.53    0.00 0.19 -2.91 -2.49 -2.33  3484    1
## log_lik[7]  -2.57    0.00 0.11 -2.80 -2.53 -2.50  4840    1
## log_lik[8]  -2.47    0.00 0.14 -2.74 -2.42 -2.36  4158    1
## log_lik[9]  -2.39    0.00 0.15 -2.70 -2.34 -2.25  3823    1
## log_lik[10] -2.55    0.00 0.23 -3.00 -2.50 -2.29  3662    1
## log_lik[11] -3.56    0.01 0.51 -4.48 -3.52 -2.81  3485    1
## log_lik[12] -3.46    0.01 0.40 -4.15 -3.44 -2.85  3458    1
## log_lik[13] -1.56    0.00 0.11 -1.76 -1.56 -1.38  3455    1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 28 12:57:05 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

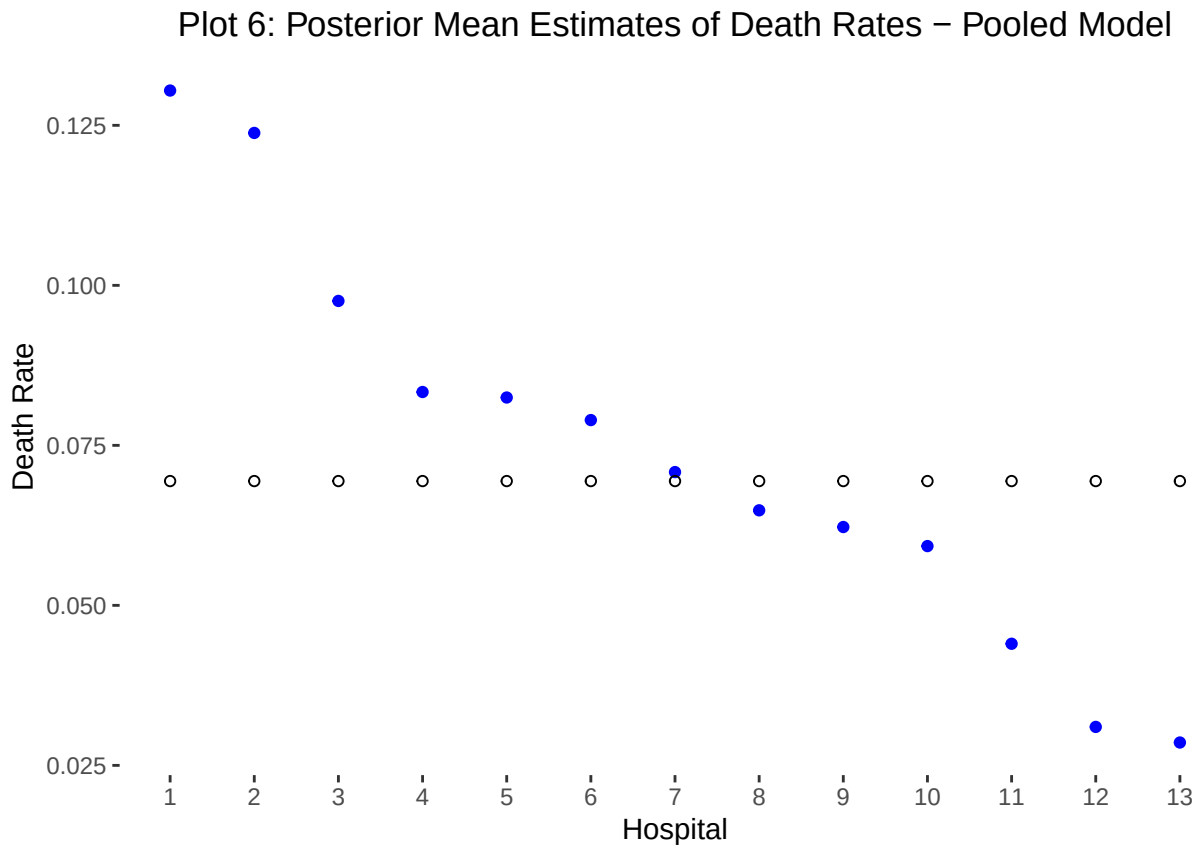
```
# extracting sample parameters of interest
posterior_pool <- as.data.frame(fit_pool,pars = c("theta","log_lik"))
#p_pool <- stack(posterior_pool[,1:13])
theta_pool <- as.data.frame(extract(fit_pool,pars = "theta"))
log_lik_pool <- extract(fit_pool)$log_lik

# calculating posterior mean estimates for death rate for each hospital
data$p_mean_pool <- apply(theta_pool,2,mean)
```

I have provided an output of the pooled model from stan above. We can see the mean, standard deviation and 90% credible intervals for the single θ value used for each hospital. We can see θ has a mean of 0.07 (observed death rate for entire data set is 0.069) and 90% credible interval in [0.06, 0.08].

In contrast to the previous hierarchical and separate models, the pooled model assumes no variation in the death rate between each of the 13 hospitals. As we observe fairly significant variance in the death rates between the hospitals this model does not fit the data very well.

I will now plot the observed death rate for each hospital per the data and overlay the posterior mean estimates from the pooled model for the death rate of each model.

```
# plot of observed death rates with overlay of posterior mean
# death rates from the hierarchical model
ggplot(data = data) +
  geom_point(aes(factor(Hospital_No),`death%`/100), col = 'blue') +
  geom_point(aes(factor(Hospital_No), p_mean_pool), shape = 1, col = 'black') +
  labs(y = "Death Rate",
       x = "Hospital",
       title = "Plot 6: Posterior Mean Estimates of Death Rates - Pooled Model") +
  theme(plot.title = element_text(hjust = 0.5))
```

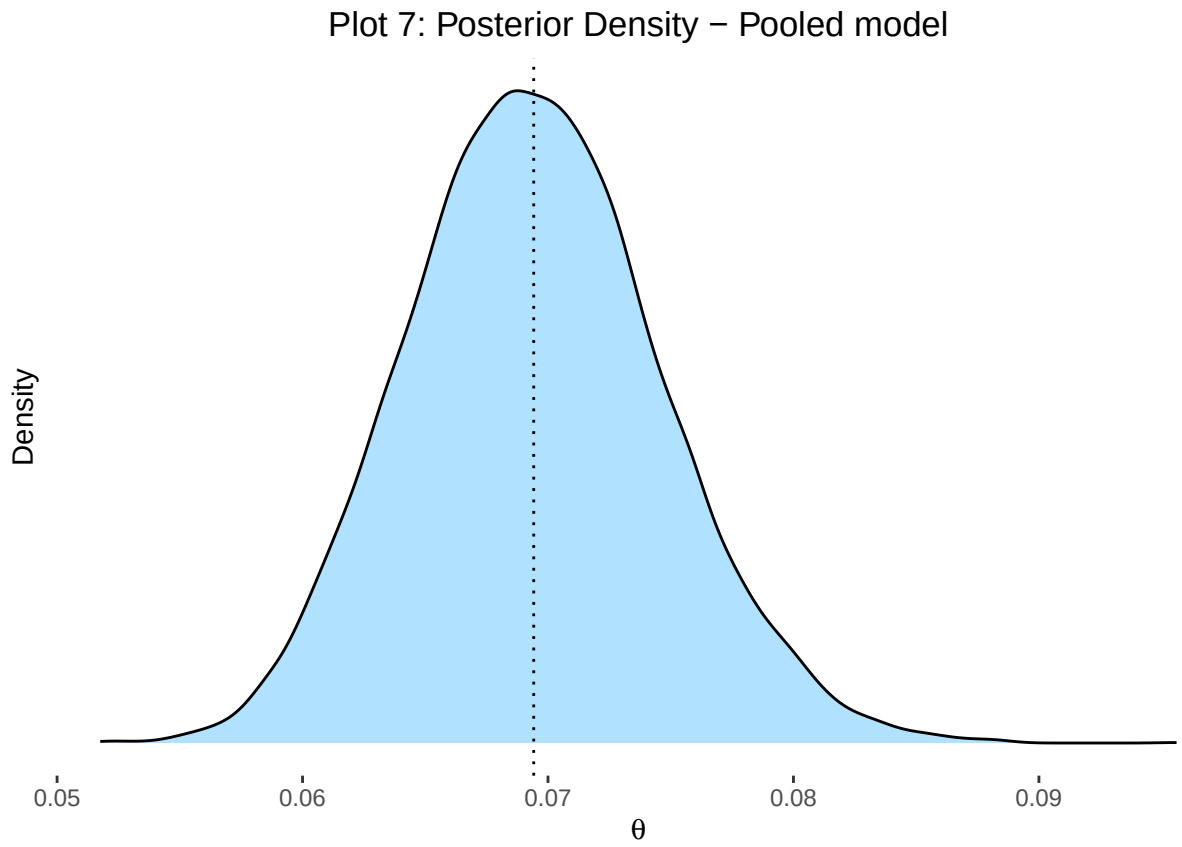


Observing the plot above of the posterior mean estimate of death rate against each observed death rate for each hospital, we can see the complete "pooling" effect. We have only one estimate for theta that is shared with all the hospitals and so the variation of death rate within each hospital is ignored. We can see that this model does not fit the observed data very well as a result.

I will now plot the posterior densities of each hospital's death rate from the pooled model.

```
# Plot Posterior Density for each hospital
ggplot(posterior_pool) +
  geom_density(aes(theta),fill = 'lightskyblue1') +
```

```
  geom_vline(xintercept = mean(theta_pool$theta), linetype='dotted') +
  labs(x = expression(theta), y = 'Density', title = 'Plot 7: Posterior Density - Pooled model') +
  scale_y_continuous(breaks = NULL) +
  theme(plot.title = element_text(hjust = 0.5))
```



I have plotted the posterior density of θ from the pooled model above. We can see that it is centered around 0.07 which aligns with the overall death rate within the data, ignoring the split between hospitals (163 deaths / 2366 cases).

Question 5

In question 5, we are asked which of the 3 models we prefer. To effectively compare the 3 models and assess which model fits the data best (out-of-sample) I will use leave-one-out cross validation available from the 'loo' package. The two functions I will employ to compare the models are 'loo_compare' and 'WAIC' which perform Leave-one-out cross-validation and widely applicable information criterion methods respectively.

```
#Comparing 3 models using loo package
loo_hier <- loo(fit_hier)
loo_sep <- loo(fit_sep)
loo_pool <- loo(fit_pool)
loo_compare(loo_hier,loo_sep,loo_pool)
```

```
##        elpd_diff se_diff
## model3  0.0       0.0
```

```
## model1 -0.8       1.4
## model2 -2.8       2.5
```

We observe the output from the leave-one-out cross-validation above and we can see that the best performing model out-of-sample is actually the pooled model. This is surprising given the variation in death rates between hospitals. The hierarchical model is the second-best performing with an elpd_diff of -0.8. We also not that the standard error is 1.4 which covers the elpd_diff between the models and could suggest that if these models were run again that the hierarchical model would be the best performing. Finally, the worst performing model out of sample is the separate model and could indicate that the model was over-fitted to the data which results in a poor out of sample performance.

```
waic_hier <- waic(log_lik_hier)
waic_sep <- waic(log_lik_sep)
waic_pool <- waic(log_lik_pool)
loo_compare(waic_hier, waic_sep, waic_pool)
```

```
##        elpd_diff se_diff
## model1  0.0       0.0
## model3 -0.2       1.5
## model2 -0.4       1.2
```

I also produce the output of the widely applicable information criterion (WAIC) analysis above. Interestingly, the results from Loo and WAIC contradict each other here. Using the WAIC, the best performing model is the hierarchical model followed by the pooled model and then the separate model. We also note that the standard error for each of the pooled and separate models in the WAIC covers the elpd_diff suggesting there is not much difference in out-of-sample performance between the models.

Question 6

In question 6, we are asked to use the hierarchical model to assess if the posterior probability of death rate in hospital 5 (Mount Sinai Beth Israel) is worse than that of hospital 2 (NYP Hospital - Allen Hospital).

I have sorted the data by descending death rate and so NYP Hospital - Allen Hospital has moved from hospital number 10 to hospital number 2.

I will use the generated posterior probability (θ_is) from each model and compare these in order to answer this question. In this context, hospital 5 will be worse than hospital 10 if it's posterior probability of death is higher more often.

```
# Extracting theta samples from the two hospitals from hierarchical model
H5_theta_samples <- p_hier$values[p_hier$ind == "theta[5]"]
H2_theta_samples <- p_hier$values[p_hier$ind == "theta[2]"]

# sample frequency of H5 death rate > H2 death rate
print(sum(H5_theta_samples > H2_theta_samples)/length(H5_theta_samples))
```

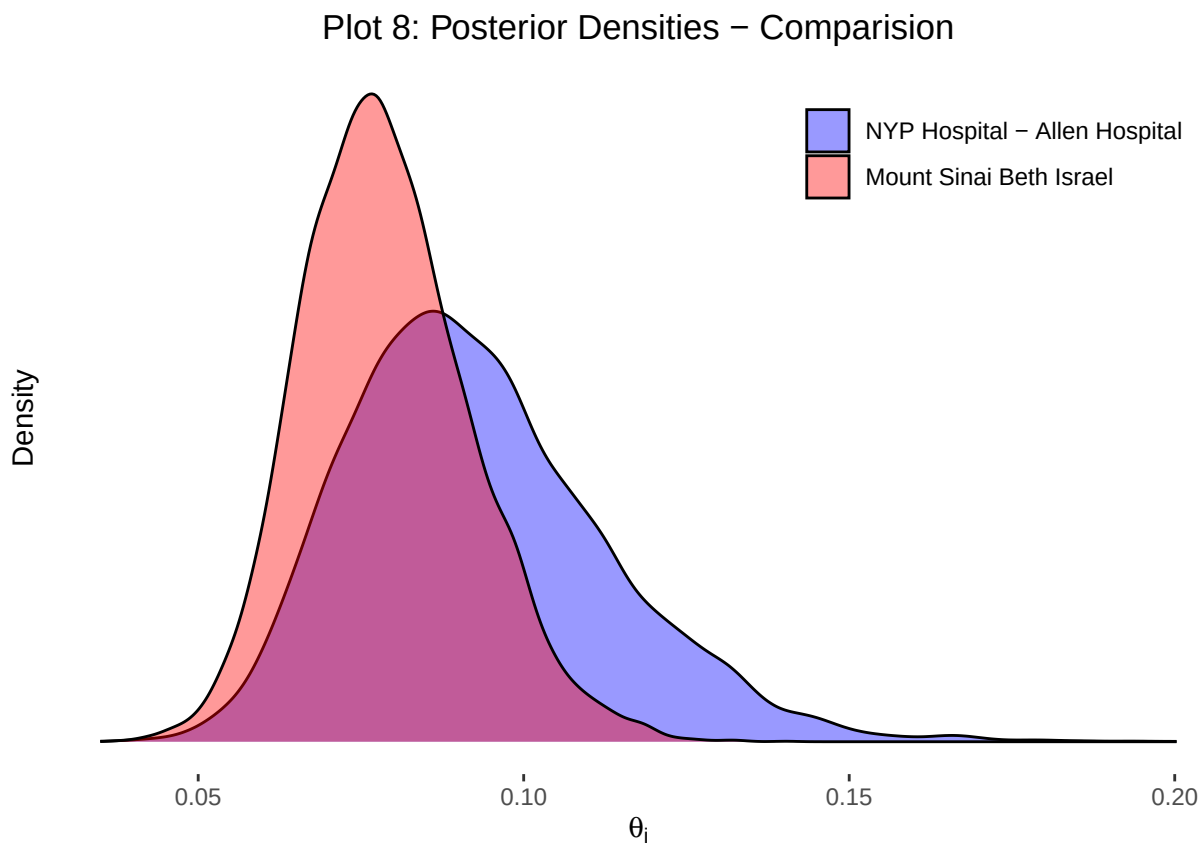
```
## [1] 0.2689
```

As we can see above, out of the 10,000 samples from the posterior distribution, the posterior probability of death for hospital 5 (Mount Sinai Beth Israel) is greater than the posterior probability of death for hospital 2 (NYP Hospital - Allen Hospital) in c.27% of the samples. This is in line with the Stan output for the hierarchical model where we observed the mean posterior estimate for the death rate for hospital 5 was 0.08

with a standard deviation of 0.01 whereas for hospital 2 the mean posterior estimate for the death rate was 0.09 with a standard deviation of 0.02.

Therefore, we can conclude that the posterior probability of death in hospital 5 is not worse than that of hospital 2. I plot the posterior distributions for each hospital below. We can see that NYP Hospital - Allen has much more denisty in higher θ values than Mount Sinai and outlines graphically what we have concluded above.

```
# plotting posterior distributions of theta for hospital 5 & 2
ggplot(p_hier[p_hier$ind %in% c("theta[5]","theta[2]"),]) +
  geom_density(aes(x = values, fill = ind),alpha = 0.4 ) +
  scale_y_continuous(breaks = NULL) +
  scale_fill_manual(values = c('blue','red'), labels = c("NYP Hospital - Allen Hospital",
                                                         "Mount Sinai Beth Israel"), name = '') +
  labs(x = expression(theta[i]), y = "Density", title = "Plot 8: Posterior Densities - Comparision") +
  theme(legend.background = element_blank(), legend.position = c(0.8,0.9),
        plot.title = element_text(hjust = 0.5))
```



Question 7

In question 7, we are asked to assess the posterior probability that hospital 12 (Bellevue Hospital Center) is ranked 1st in terms of lowest death rate using the hierarchical model. Similar to question 6 above we can use the 10,000 generated samples of the posterior death rate for hospital 12 and then compare these values to each of the other hospital's posterior death rates. In this way, we can estimate the posterior probability that hospital 12 is ranked 1st - 13th in terms of lowest death rate.

```r
# calculating rank of each hospitals posterior death rate for each sample (row)

# I use the apply function along with the rank function in order to produce
# the rank for each hospital's posterior death rate.
# t() is used to transpose the result of the apply function
rank_theta_hier <- t(apply(theta_hier, 1, rank))
colnames(rank_theta_hier) <- data$hospital

# probability hospital 12 is ranked lowest (i.e. has a rank of 1)
print(sum(rank_theta_hier[,12] == 1) / nrow(rank_theta_hier))
```

```
## [1] 0.318
```

```r
# repeating analysis for each of ranks 1-13 (store results in new data frame)
H5_rank_df <- data.frame(rank = 1:13)
for (i in 1:nrow(H5_rank_df)) {
  H5_rank_df$rank_prob[i] = (sum(rank_theta_hier[,12] == i) / nrow(rank_theta_hier))
}
```

As we can see from the result above, the posterior probability of death of hospital 12 (Bellevue Hospital Center) is ranked first in terms of lowest death rate in 31.8% of the 10,000 generated samples from the hierarchical model.

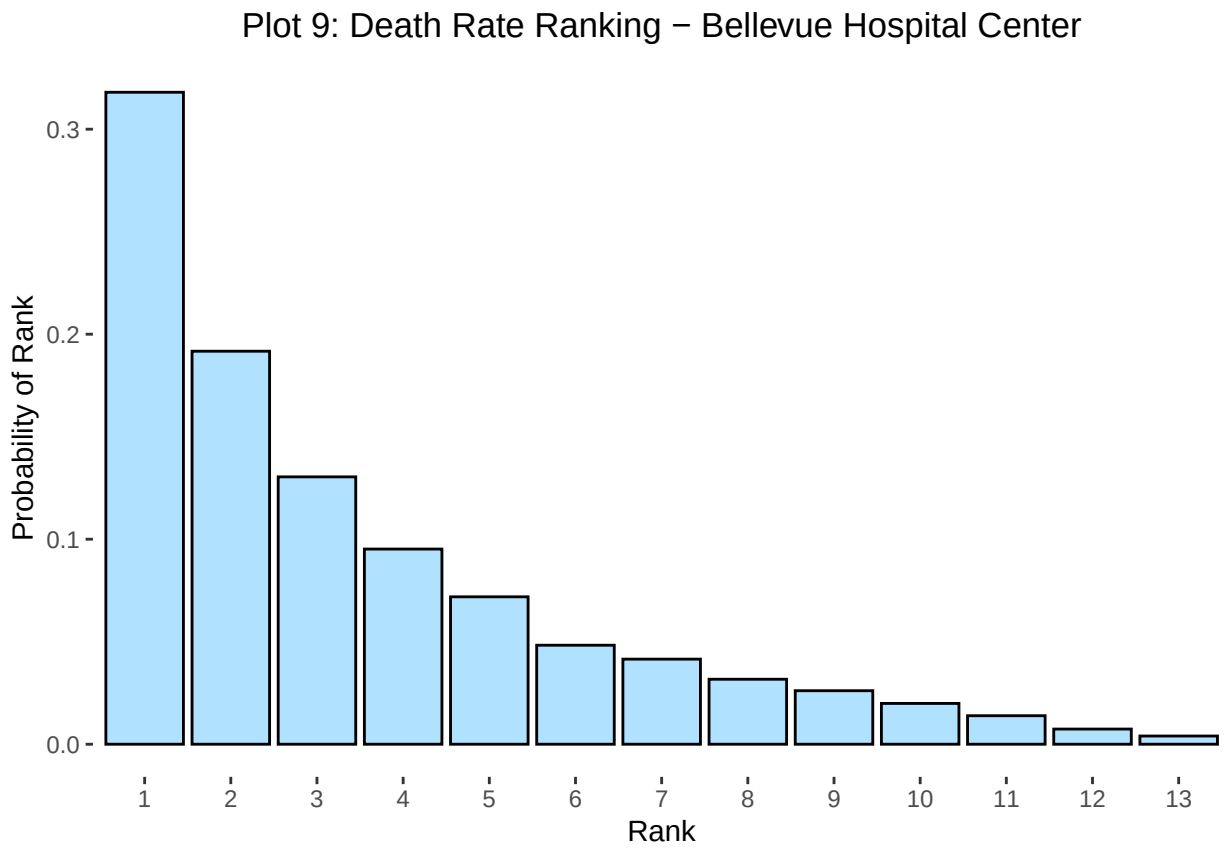
I have repeated this analysis for the remaining 12 ranks (2-13) and provide a summary table as well as a plot below to show the posterior probability that hospital 12's death rate achieves each of the rankings (1-13).

```r
# summary table showing hospital 12s ranking
kable(H5_rank_df,
      caption = "Bellevue Hospital Centre - Death Rate Ranking",
      align = "l") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 2: Bellevue Hospital Centre - Death Rate Ranking

rank	rank_prob
1	0.3180
2	0.1917
3	0.1304
4	0.0952
5	0.0719
6	0.0483
7	0.0415
8	0.0317
9	0.0261
10	0.0199
11	0.0139
12	0.0074
13	0.0040

```r
# histogram of rankings of posterior death rate for hospital 12
ggplot(data = H5_rank_df) +
  geom_col(aes(x = factor(rank), y = rank_prob), fill = "lightskyblue1", col = "black") +
  labs(x = "Rank", y = "Probability of Rank", title = "Plot 9: Death Rate Ranking - Bellevue Hospital C
  theme(plot.title = element_text(hjust = 0.5))
```



We can see from the summary table and bar chart above that hospital 12 (Bellevue Hospital Center) is ranked first (lowest) in terms of death rate in 31.8% of the 10,000 generated samples. Observing the chart above we can see that hospital 12 performs very well in terms of it's ranking, being in the top 5 ranks c.80% of the time.

Question 8

In question 8, we are asked to assess the posterior probability that each hospital is ranked 1st in terms of lowest death rate using the hierarchical model. I will use the same method as in questions 6 and 7 where the generated samples of theta values from the stan output are used to estimate the posterior probability of each hospital being ranked first (lowest) in terms of death rate.

```r
# Using the previously calculated rank matrix from question 7
# to estimate the posterior probability of each hospital being ranked
# first in terms of lowest death rate

# posterior probability that each hospital is ranked first
all_hospitals_rank_prob <- data.frame(hospital_no = 1:13)
all_hospitals_rank_prob$hospital_name <- data$hospital
```

```r
for (i in 1:13) {
  all_hospitals_rank_prob$rank_1_prob[i] = (sum(rank_theta_hier[,i] == 1) / nrow(rank_theta_hier))
}

# sort the ranked data frame from highest probability of lowest death rate to
# lowest probability
all_hospitals_rank_prob <- all_hospitals_rank_prob %>%
  arrange(desc(rank_1_prob))

# summary table showing hospital 12s ranking
kable(all_hospitals_rank_prob,
      caption = "All Hospitals - Posterior Probability of Ranked 1st in Death Rate",
      align = "l") %>%
  kable_styling(latex_options = "HOLD_position")
```

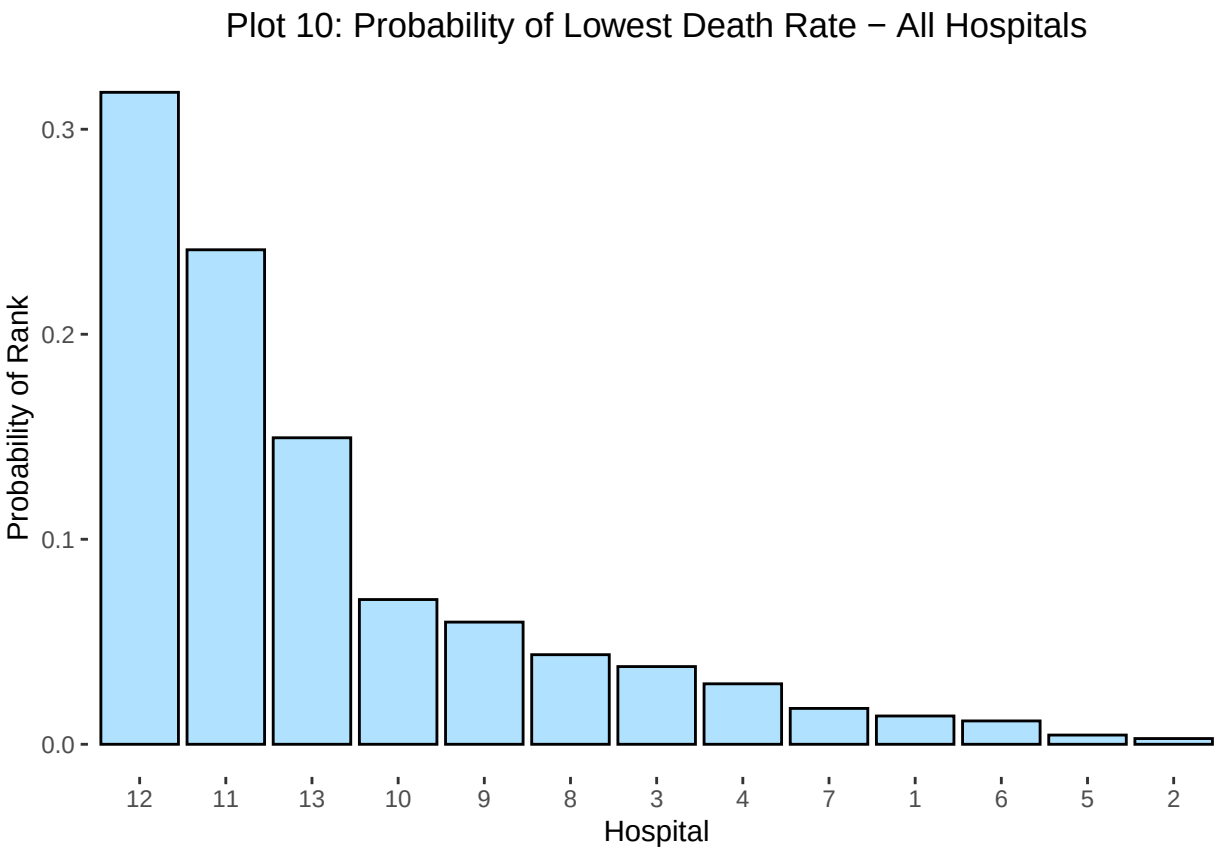
Table 3: All Hospitals - Posterior Probability of Ranked 1st in Death Rate

hospital_no	hospital_name	rank_1_prob
12	Bellevue Hospital Center	0.3180
11	NYP Hospital - New York Weill Cornell Center	0.2412
13	Harlem Hospital Center	0.1495
10	Mount Sinai Hospital	0.0706
9	NYU Hospitals Center	0.0596
8	Mount Sinai St. Luke's	0.0437
3	NYP/Lower Manhattan Hospital	0.0379
4	Metropolitan Hospital Center	0.0295
7	NYP Hospital - Columbia Presbyterian Center	0.0175
1	Mount Sinai Roosevelt	0.0138
6	Lenox Hill Hospital	0.0114
5	Mount Sinai Beth Israel	0.0045
2	NYP Hospital - Allen Hospital	0.0028

I have produced a summary table above showing the posterior probability of each hospital being ranked first (lowest) in terms of it's death rate. We can see that Bellevue hospital center is most likely to be ranked first, with a probability of 31.8% followed by NYP Hospital - New York Weill Cornell Center with a probability of 24.1% and Harlem Hospital Center with a probability of 15%. NYP Hospital - Allen has the lowest probability of being ranked first in terms of lowest death rate with a posterior probability of 0.28%.

I have also produced a bar chart below ordered from most likely to least likely of every hospital being ranked first.

```r
# histogram of posterior probability that
# each hospital is ranked lowest in death rate
ggplot(data = all_hospitals_rank_prob) +
  geom_col(aes(x = factor(hospital_no, levels = `hospital_no`), y = rank_1_prob), fill = "lightskyblue1
  labs(x = "Hospital", y = "Probability of Rank", title = "Plot 10: Probability of Lowest Death Rate - A
  theme(plot.title = element_text(hjust = 0.5))
```


Plot 10: Probability of Lowest Death Rate − All Hospitals
Probability of Rank
0.0
0.1
0.2
0.3
12
11
13
10
9
8
3
4
7
1
6
5
2
Hospital