Advanced Bayesian Assignment 2

16343261

13/08/2023

Table of Contents

Before we begin this assignment, we will first load our necessary packages and our data.

```r
setwd("C:/Users/matth/Documents/Advanced Bayesian")

load("~/Advanced Bayesian/educ.RData")
library(bayesrules)
library(tidyverse)
library(rstanarm)
library(bayesplot)
library(tidybayes)
library(broom.mixed)
library(forcats)
library(caret)
library(brms)
```

Bayesian Workflow

The core question that will be answered in this report is the following: Model the Educ dataset to identify the significant predictors of repeat rates in Thai primary schools.

To begin answering this question, we must first define our Bayesian Workflow process. This can used throughout the document for the basis of our question answering process. The steps are as follows.

1. We must begin by specifying a prior distribution.
2. We then fit our Bayesian GLM model using the stan_glm function. We provide the formula specifying the outcome variable and predictor variables, the data, the prior

distribution, and specify the number of chains and iterations for the MCMC sampling.

3. Stating our priors, we then use the Update function to include our new priors into our model and simulate the posterior as necessary.
4. We then use density plots to either validate or invalidate our findings.

I will be loading in my model outputs from saved files in my directory to save computation time for RMarkdown, this entire analysis is reproducible by removing the eval = False notation in the .Rmd file.

```r
load("~/Advanced Bayesian/running_model_2.Rdata")
load("~/Advanced Bayesian/educ_model.Rdata")
load("~/Advanced Bayesian/educ_model_prior.Rdata")
load("~/Advanced Bayesian/educ_model_BLR.Rdata")
```

We will begin by first splitting our data set into a testing and training data sets.

```r
test_index <- sample.int(n = nrow(educ), size = 750)
educ_test <- educ[test_index,]
educ_train <- educ[-test_index,]
```

Complete Pooled Model

To begin, we will fit a Bayesian Binomial Glm with Repeat as our response variable and PPED, SEX and MSESC as explanatory variables. By specifying family = binomial in the stan_glm function, the model is fitted using the binomial likelihood function.

This is for all intents and purposes a complete pooled model. With complete pooled models we combine all data points together, not considering the presence of groups. We assume they are independent and that a universal model is appropriate for all groups. In this instance, we are ignoring the SchoolID variable.

```r
educ_model_prior <- stan_glm(
REPEAT ~ MSESC + PPED + SEX,
data = educ_train, family = binomial,
prior_intercept = normal(0, 2.5),
prior = normal(0, 2.5, autoscale = TRUE),
chains = 4, iter = 5000*2, seed = 3515)

educ_model_prior$prior.info$prior$adjusted_scale

## [1] 6.646194 4.999680 4.999914
```

So, stan_glm gives the following priors:

- β1 ~ Normal(0, (6.646194)^2)
- β2 ~ Normal(0, (4.999680)^2)
- β3 ~ Normal(0, (4.999680)^2)

Now lets fit our final model.

```
educ_model <- update(educ_model_prior, prior_PD = FALSE)
```

We will now use the tidy function to simulate our posterior. We will perform a hypothesis test with:

H0: βi /= 0 H1: βi = 0

for i = 1,2,3

```
tidy(educ_model, effects = "fixed", conf.int = TRUE, conf.level = 0.95)

## # A tibble: 4 x 5
##   term        estimate std.error conf.low conf.high
##   <chr>          <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)   -1.76     0.0614   -1.88    -1.64
## 2 MSESC         -0.259    0.0977   -0.456   -0.0698
## 3 PPED1         -0.545    0.0737   -0.694   -0.402
## 4 SEX1           0.374    0.0707    0.235    0.511
```

As we have 3 variables we will have to run a separate hypothesis test on each of our variables. From our table above, our model finds that there is a relationship between a student having to repeat and our explanatory variables.

As 0 is not contained in any of our confidence intervals, we find that for all our hypothesis tests, we fail to reject our null hypothesis. We find that given a student has attended pre-primary education, their chances of repeating reduces by 0.545 on average. Secondly, we find that if the mean pupil socio-economic status at the school level is 1, this reduces a students chances of repeating by 0.259 on average.

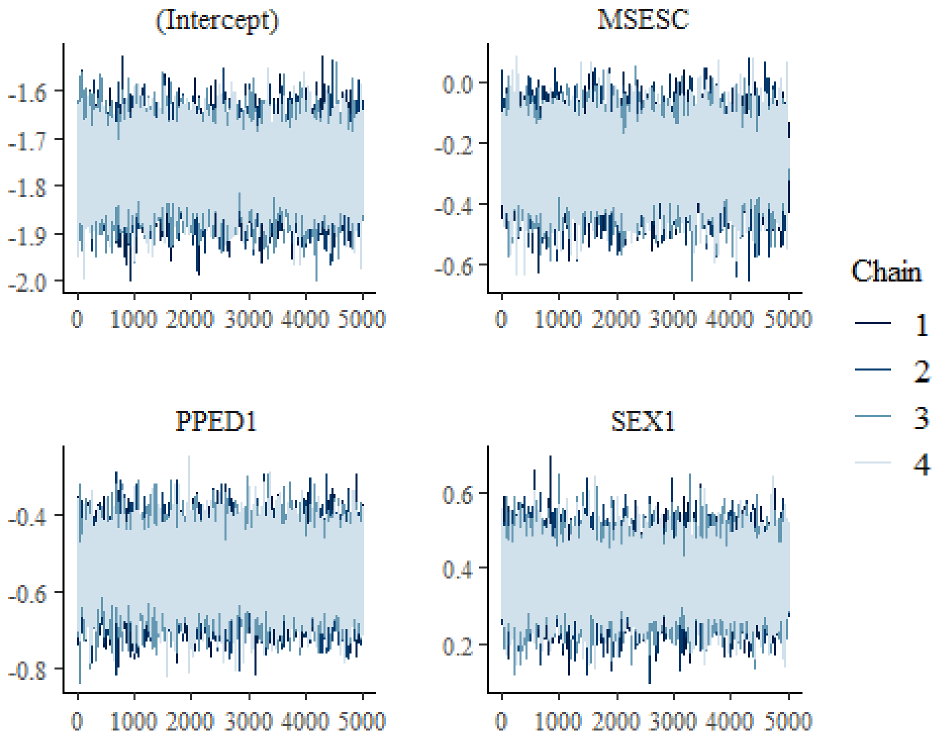
```
mean(educ$MSESC)

## [1] 0.009674029
```

Given our mean MSESC is 0.01, we can likely say that the most predominant factor in our data set driving repeat rates is the socio-economic status as they are significantly far from 1 on average. It can be extrapolated that this factor is significantly increasing the number of repeats.

The one positive factor in our model is our sex variable. Our findings indicate that if a student is a male, their chances of repeating increase by 0.374. This may be indicating that male students in general are just more likely to repeat than female students. However we should be cautious of this interpretation as there could be many cultural expectations and economic factors present in Thailand that may cause male students to perform worse in school than female students. It may be worthwhile to see if male students are more or less likely to receive pre-primary education as we know this has a major affect on repeat rates.

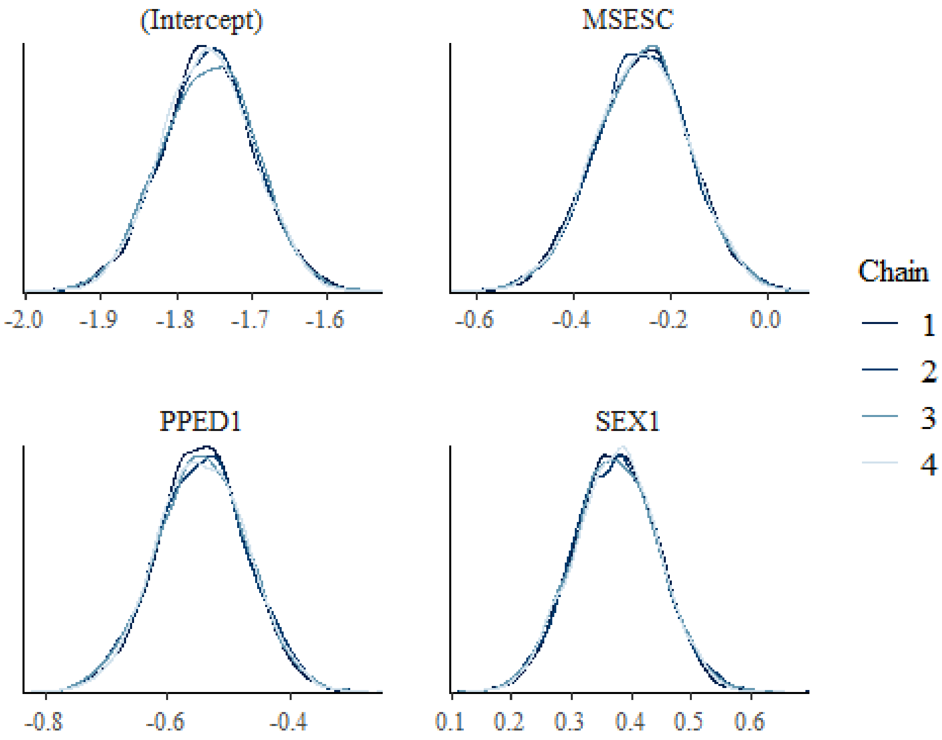
Before we validate our analysis, we must first check our MCMC trace graph.

```
mcmc_trace(educ_model)
```



As can be seen above, we have the proper trace for all of our variables. Next we will check if all our chains are create the correct/ similar density plots.

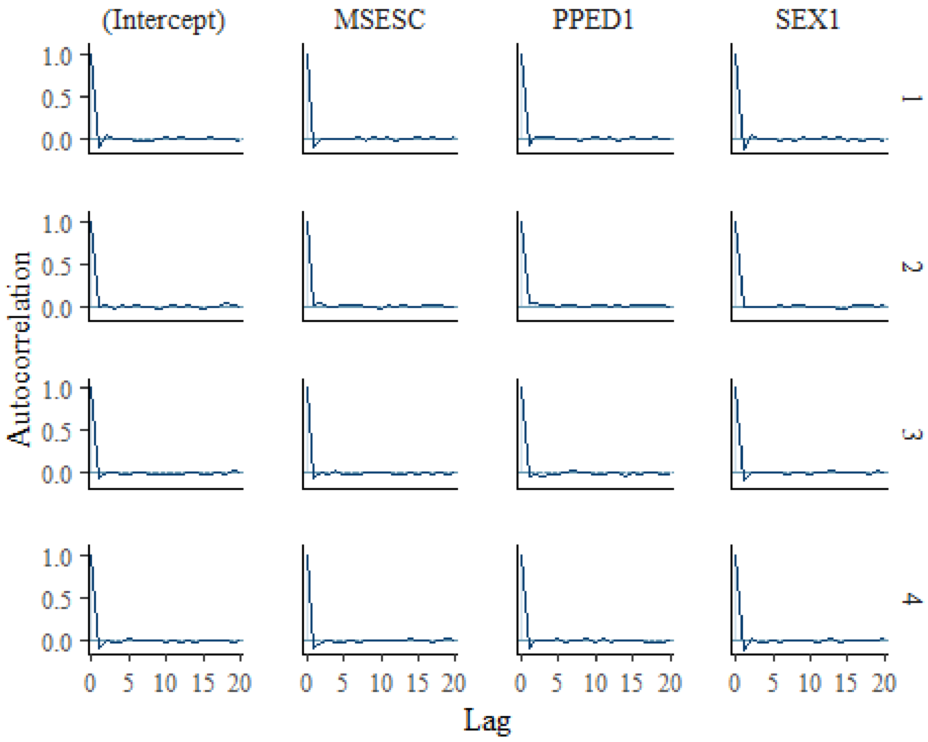
```
mcmc_dens_overlay(educ_model)
```



As we can see above, they are all quite similar, which means we can take interpret as non-suspect.

Lastly we will check our Autocorrelation plot.

```
mcmc_acf(educ_model)

## Warning: The `facets` argument of `facet_grid()` is deprecated as of
ggplot2 2.2.0.
## i Please use the `rows` argument instead.
## i The deprecated feature was likely used in the bayesplot package.
##   Please report the issue at <https://github.com/stan-
dev/bayesplot/issues/>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



As our autocorrelation decreases very quickly, we can accept our computation as valid.

Classification - Complete Pooled Model

To test the classification accuracy of our fitted complete pooled education model, we will fit a confusion model on our data. As our repeat variable is binary we will set 0.5 as our cut off value. We will then use our test data set and compare it with our predicted values.

```
set.seed(4561)
binary_prediction <- posterior_predict(
educ_model, newdata = educ_test)
Y_hat <- colMeans(binary_prediction)
Y_hat <- round(Y_hat, 2)
Y <- educ_test$REPEAT

confusion_50 <- table(Y, Y_hat = ifelse(Y_hat >= 0.5, "Yes", "No"))
confusion_50 <- matrix(c(646, 0, 104, 0), ncol = 2, byrow = TRUE)
rownames(confusion_50) <- c("0", "1")
colnames(confusion_50) <- c("No", "Yes")
sensitivity <- confusion_50[2, 2] / sum(confusion_50[2, ])
specificity <- confusion_50[1, 1] / sum(confusion_50[1, ])
accuracy <- sum(diag(confusion_50)) / sum(confusion_50)
result <- list(
  "confusion matrix" = confusion_50,
  "sensitivity" = sensitivity,
  "specificity" = specificity,
  "overall accuracy" = accuracy
)
result
```

```
## $`confusion matrix`
##     No Yes
## 0 646   0
## 1 104   0
## 
## $sensitivity
## [1] 0
## 
## $specificity
## [1] 1
## 
## $`overall accuracy`
## [1] 0.8613333
```

Here we find an overall accuracy of 86.1%, which can be considered a relatively high level of prediction. We will accept this as our baseline accuracy and compare it with our other models.

Hierarchical GLM Model

Now let's fit a hierarchical glm using SchoolID as our grouping variable and interpret our results.

```
running_model_2 <- stan_glmer(
REPEAT_NUM ~ PPED + SEX + MSESC + (1 | SCHOOLID),
data = educ, family = binomial(),
prior_intercept = normal(0, 2.5),
prior = normal(0, 2.5),
chains = 4, iter = 1000, seed = 84735, adapt_delta = 0.95
)

prior_summary(running_model_2)
```

```
## Priors for model 'running_model_2' 
## ------
## Intercept (after predictors centered)
##  ~ normal(location = 0, scale = 2.5)
## 
## Coefficients
##  ~ normal(location = [0,0,0], scale = [2.5,2.5,2.5])
## 
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
## See help('prior_summary.stanreg') for more details
```

As before, we will now use the tidy function to simulate our posterior. We will perform a hypothesis test with:

H0: βi /= 0 H1: βi = 0

for i = 1,2,3

```
tidy(running_model_2, effects = "fixed", conf.int = TRUE, conf.level = 0.95)

## # A tibble: 4 x 5
##   term        estimate std.error conf.low conf.high
##   <chr>          <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)   -2.24     0.112    -2.46     -2.05
## 2 PPED1         -0.624    0.0998   -0.830    -0.424
## 3 SEX1           0.533    0.0761    0.393     0.688
## 4 MSESC         -0.291    0.220    -0.698     0.138
```

Our findings are somewhat similar to our complete pooled model, however in our Hierarchical model we find that our MSESC variable fails to be statistically significant as 0 is included in it's 95% confidence interval. Therefore we must reject our null hypothesis and accept that MSESC's affect on Repeat rates could be zero according to our model.

We find that for all our other hypothesis tests, we fail to reject our null hypothesis. We find that given a student has attended pre-primary education, their chances of repeating reduces by 0.624 on average. Not too dissimilar from our Completed pooled model.

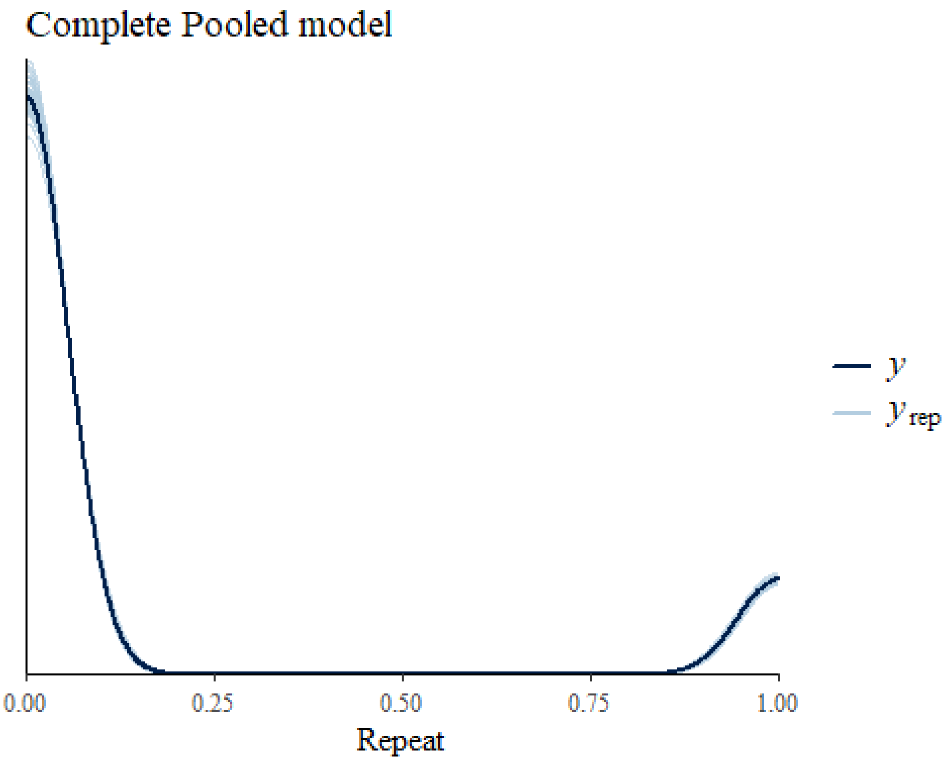
Once again, we find our one positive factor in our model is our sex variable. Our findings indicate that if a student is a male, there chances of repeating increase by 0.533. This is considerably higher than our calculated value in our completely pooled model. We will now perform a posterior analysis of within group variability.

```
tidy(running_model_2, effects = "ran_pars")

## # A tibble: 1 x 3
##   term                     group    estimate
##   <chr>                    <chr>       <dbl>
## 1 sd_(Intercept).SCHOOLID SCHOOLID     1.31
```

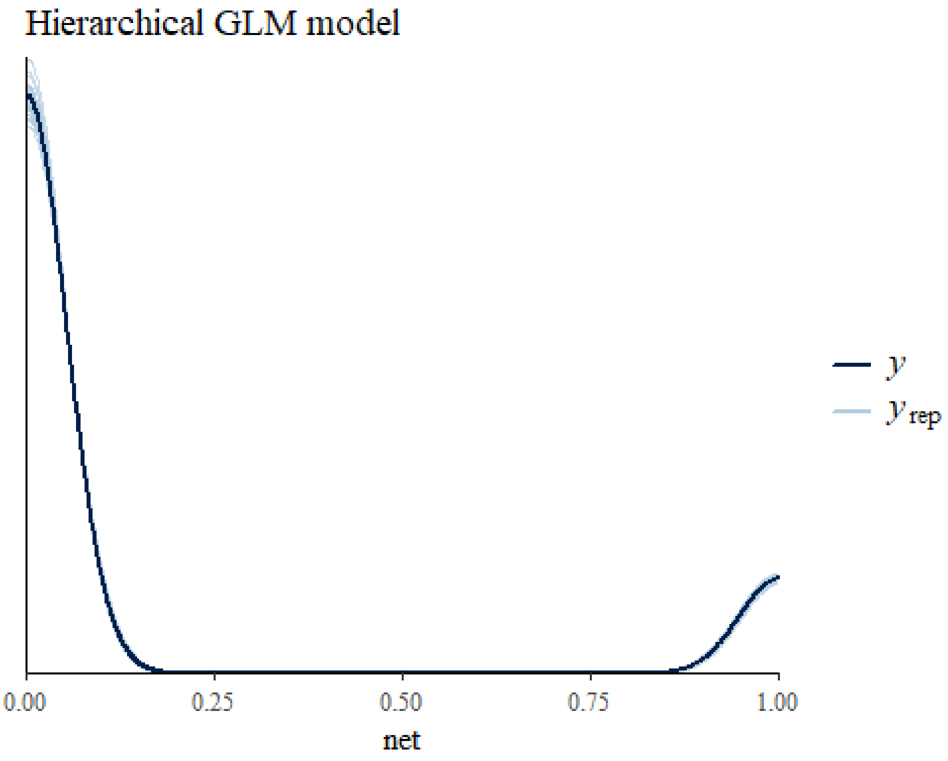
Here we find the standard deviation for our intercept within our SchoolID group is 1.31.

To directly compare our models, we will use the posterior predictive checks and compare.

```
pp_check(educ_model) +
labs(x = "Repeat",title = "Complete Pooled model")
```


Complete Pooled model

```r
pp_check(running_model_2) +
labs(x = "net",title = "Hierarchical GLM model")
```


Hierarchical GLM model

It does appear from our plot that for both models our simulated datasets are very close to the ones which we are observing. This is indicative of both of our models being comparable.

Classification - Hierarchical GLM

We will now compare the classification accuracy of our Hierarchical GLM model with our baseline accuracy.

```r
binary_prediction <- posterior_predict(
  running_model_2, newdata = educ_test)
  Y_hat <- colMeans(binary_prediction)
Y_hat <- round(Y_hat, 2)
Y <- educ_test$REPEAT

confusion_50 <- matrix(c(640, 6, 83, 21), ncol = 2, byrow = TRUE)
rownames(confusion_50) <- c("0", "1")
colnames(confusion_50) <- c("No", "Yes")
sensitivity <- confusion_50[2, 2] / sum(confusion_50[2, ])
specificity <- confusion_50[1, 1] / sum(confusion_50[1, ])
accuracy <- sum(diag(confusion_50)) / sum(confusion_50)
result <- list(
  "confusion matrix" = confusion_50,
  "sensitivity" = sensitivity,
  "specificity" = specificity,
  "overall accuracy" = accuracy
)
result
```

```
## $`confusion matrix`
##    No Yes
## 0 640   6
## 1  83  21
## 
## $sensitivity
## [1] 0.2019231
## 
## $specificity
## [1] 0.9907121
## 
## $`overall accuracy`
## [1] 0.8813333
```

Here we find a classification accuracy of 88.1% which is 2% higher than our complete pooled. This indicates that our Hierarchical GLM model is better at distinguishing between classes. This is potentially because of our inclusion of the SCHOOLID variable.

Bayesian Logistic Regression using Variational Inference

We will now fit Bayesian logistic regression using variational inference with the Jaakola and Jordan bound and compare our results to both our Complete Pooled Model and our Hierarchical GLM model.

Let's begin by preparing our data:

```
# Prepare the data
X <- educ_train %>% select(MSESC, PPED, SEX) %>%
  add_column(intercept = 1) # column for intercept
X <- as.matrix(X) # ensure X is in matrix form
X <- matrix(as.numeric(X),ncol = ncol(X))
Y <- educ_train$REPEAT
Y <- as.numeric(Y)
# Priors for beta_1, beta_2 and beta_3 are from from
educ_model_prior$prior.info
m_0 <- c(-2, -0.2, -0.5, 0.4)
# prior means
S_0 <- diag(c(0.248, 0.315, 0.271, 0.266), 4)
ELBO <- rep(0,6766)
```

Next lets fit our Bayesian Logistic Regression function.

```
BayesianLogisticRegression_VI<- function(Y, X, m_0, S_0, nstart){
  inv_S_0 <- solve(S_0)
  a <- nrow(X)  # Number of observations
  set.seed(3451)
  E <- rnorm(a)
  expression_result <- 0
  expression_result2 <- 0
  expression_result3 <- 0
  sigma_E <- c(0)
  lambda_E <- c(0)
  Q_E_E_old <- 0
  for (n in 1:nstart) {
    sigma_E[n] <- 1/(1 + exp(-E[n]))
    lambda_E[n] <- (1/2*E[n])*(sigma_E[n]-1/2)
    expression_result <- expression_result + sum((Y[n] - 0.5) * X[1:n, ])
    expression_result2 <- expression_result2 + sum(lambda_E[n]*(X[1:n,
])%*%(t(X[1:n, ])))
    expression_result3 <- expression_result3 + log(sigma_E[n]) + E[n]/2 +
lambda_E[n]*(E[n]^2)
  }
# E - Step
  inv_S_N <- solve(S_0) + 2*expression_result2
  S_N <- solve(inv_S_N)
  m_N <- S_N%*%(solve(S_0)%*%m_0 + expression_result)
  beta <- solve(inv_S_N) %*% m_N
  e_beta <- beta%*%t(beta)
  expected_beta<- mean(e_beta)
# M - Step
  for (n in 1:nstart) {
    Q_E_E_old <- Q_E_E_old + sum(log(sigma_E[n]) - E[n]/2 -
                               lambda_E[n]*(t(X[n, ])*expected_beta*X[n,
] - (E[n])^2))
  }
  E_new_squared <- t(X[1:n, ])%*%X[1:n, ]*expected_beta
# Lower bound of the log - likelihood
  ll <- 0
  ll <- (1/2)*log(det(S_N)/det(S_0)) + (1/2)*t(m_N)%*%inv_S_N%*%m_N -
    (1/2)*t(m_0)%*%inv_S_0%*%m_0 + expression_result3
  ELBO[n] <- ll
  # Calculate posterior summary
  post.mean <- m_N
  post.sd <- sqrt(diag(S_N))
  conf.low.95 <- post.mean - 1.96 * post.sd
  conf.high.95 <- post.mean + 1.96 * post.sd
  res <- c(0)
  # Format the summary
  summary <- data.frame(
    post.mean = post.mean,
    post.sd = post.sd,
    conf.low.95 = conf.low.95,
    conf.high.95 = conf.high.95,
    res <- ELBO[n])
  return(summary)
}
```

To repeat until convergence, I will use the below for loop:

```
tolerance <- 1e-6
res_v <- numeric(6766)
# Preallocate a numeric vector
for (i in 1:6766) {
  res_v[i] <- BayesianLogisticRegression_VI(Y, X, m_0, S_0, nstart = i)[1,5]
# Store the ELBO directly
  if (i > 1 && abs(res_v[i] - res_v[i - 1]) < tolerance) {
    break
  }
  print(i)
}
```

The methodology I have used here is considerably computationally intensive, resulting in an incredibly slow assessment. This may be because my threshold was too low to select an adequate value. After roughly 25 minutes, my model has only gone through 800 values. I have decided to take the maximum from this subset as it is clear some aspect of my model is inefficient.

```
BayesianLogisticRegression_VI(Y, X, m_0, S_0, nstart = 695)
```

```
##     post.mean   post.sd conf.low.95 conf.high.95 res....ELBO.n.
## 1 -1.47939582 0.4382776 -2.33841996   -0.6203717       1374.346
## 2  0.46125128 0.4741260 -0.46803567    1.3905382       1374.346
## 3  0.06888602 0.4519242 -0.81688535    0.9546574       1374.346
## 4  0.95838997 0.4490839  0.07818549    1.8385944       1374.346
```

From these results, it is clear that our model has not been fit with the right En value as both the MSESC and PPED1 variables are statistically insignificant. Leaving only our sex variable and our intercept as significant, with sex playing a massive role in explaining repeat rates. It is highly unlikely that this is the case.

The last model I will fit will be Bayesian Logistic Regression with variational inferance. using stan code. I will achieve this using the the mean field algorithm. Mean field approximation is a technique commonly used in Bayesian inference, particularly in Variational Inference (VI), to approximate complex posterior distributions with simpler distributions. It is used to make Bayesian inference more computationally tractable when dealing with high-dimensional or intricate models where exact inference is analytically or computationally challenging. This can be seen below:

```
educ_model_BLR <- stan_glm(
  REPEAT ~ MSESC + PPED + SEX,
  data = educ_train, family = binomial,
  prior_intercept = normal(0, 2.5),
  prior = normal(0, 2.5,  autoscale = TRUE), algorithm = "meanfield")

tidy(educ_model_BLR, effects = "fixed", conf.int = TRUE, conf.level = 0.95)

## # A tibble: 4 x 5
##   term        estimate std.error conf.low conf.high
##   <chr>          <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)   -1.76     0.0599   -1.90    -1.63
## 2 MSESC         -0.260    0.0984   -0.462   -0.0729
## 3 PPED1         -0.546    0.0745   -0.675   -0.402
## 4 SEX1           0.377    0.0685    0.235    0.513
```

The results from our Mean Field stan model are almost identical to that of our complete pool model. Our interpretation of these results will be identical, for this reason, we will go no further with our checks.

Conclusion

In conclusion, we find our fitted Hierarchical Model to be our model with the highest classification rate. It identified the following significant predictors of repeat rates in Thai primary schools.

- We find that our MSESC variable fails to be statistically significant as 0 is included in it's 95% confidence interval. Therefore we must reject our null hypothesis and accept that MSESC's affect on Repeat rates could be zero according to our model.

- We find that given a student has attended pre-primary education, their chances of repeating reduces by 0.624 on average.
- We find our one positive factor in our model is our sex variable. Our findings indicate that if a student is a male, their chances of repeating increase by 0.533.
- Grouping our dataset by SchoolID appears to improve our models classification accuracy.