Assignment – 4

STAT40850 – Bayesian Analysis

Meghashree Madhava Rao Ramachandrahosur (21200301)

Load libraries

```
# Library for stand code
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = 4)

# library for comparison
library(bridgesampling)

# Set working directory path
setwd("C:/Users/DELL/OneDrive/Documents/SEM-2/Bayesian/Assignment4")

# Read the data set
data <- read.csv("coffee.csv")

table(data$cafe)

##  1  2  3  4  5
## 18  9 14 16 10
```

We'll organize the data into a list

```
dat1 <- list(
  N = 67, # Number of observations
  G = 5, # Number of groups
  rating = data$ratings, # individual ratings
  g = data$cafe) # group assignments
```

Separate model:

We now write the separate model where we don't pool information across tanks. So each tank has its own posterior distribution.

$$\sigma_g = normal(20,50)$$

$$\boldsymbol{\mu}_g = normal(0,50)$$

$$rating_i = normal(\boldsymbol{\mu}_g, \sigma_g)$$

Where,

i varies from 1 to N(number of observations)

g is the number of groups(café)

The priors for this model were vaguely chosen, but keeping in mind that the ratings scale is between [0, 100].

Implement the model in stan

We now write a stan file sep_model.stan for our new model.

```
# Loading Stan file for the code
writeLines(readLines("sep_model.stan"))
```

```
## // Index values and observations.
## data {
##   int<lower = 1> N;               // Number of observations.
##   int<lower = 1> G;               // Number of groups.
##   vector[N] rating;                    // Vector of observations.
##   int<lower = 1, upper = G> g[N]; // Vector of group assignments.
## }
##
## // Parameters and hyperparameters.
## parameters {
##
##   vector[G] beta;                // Vector of group intercepts.
##   vector[G] sigma;                // Variance of the likelihood.(slope)
## }
##
## // Hierarchical regression.
## model {
##
##   // Population model and likelihood.
##   target += normal_lpdf(beta| 0,50);
##   target += normal_lpdf(sigma| 20, 50);
##
##   for (n in 1:N) {
##       target += normal_lpdf(rating[n]| beta[g[n]],sigma[g[n]]);
##   }
## }
## generated quantities{
##
##   vector[N] y_rep;
##   for (n in 1:N) {
##       y_rep[n] = normal_rng(beta[g[n]],sigma[g[n]]);
##   }
## }
```

We now fit the model.

```
fit_sm <- stan("sep_model.stan",  data=dat1, iter=5000)
```

Explore the output from the model

Print the output of the fit

```
round((summary(fit_sm)$summary[1:10,]),3)

##            mean se_mean    sd   2.5%    25%    50%    75%  97.5%    n_eff
Rhat
## beta[1]  33.441   0.042 4.140 25.267 30.739 33.463 36.132 41.501 9748.698
1.000
## beta[2]  45.745   0.082 6.385 32.825 41.970 45.823 49.654 57.993 6093.618
1.001
## beta[3]  74.538   0.033 2.975 68.485 72.650 74.586 76.462 80.372 7984.086
1.000
## beta[4]  62.081   0.025 2.212 57.740 60.642 62.097 63.526 66.411 7981.331
1.000
## beta[5]  51.725   0.060 4.000 43.958 49.386 51.787 54.200 59.311 4387.352
1.001
## sigma[1] 17.842   0.043 3.331 12.766 15.497 17.377 19.610 25.810 5885.014
1.000
## sigma[2] 18.313   0.132 6.205 10.827 14.291 17.100 20.692 32.842 2195.799
1.003
## sigma[3] 11.063   0.037 2.542  7.410  9.309 10.649 12.296 17.225 4712.933
1.000
## sigma[4]  8.652   0.023 1.749  6.023  7.407  8.382  9.601 12.754 5672.498
1.001
## sigma[5] 11.675   0.087 3.683  7.077  9.315 10.944 13.151 20.325 1807.810
1.002
```

Pooled model

We now write the pooled model where there is no distinction made between which coffee shop each rating was for.

$$\beta = normal(0,50)$$

$$\sigma = normal(20,50)$$

$$rating = normal(\beta, \sigma)$$

Implement the model in stan

We now write a stan file pooled_model.stan for our new model.

```
# Loading Stan file for the code
writeLines(readLines("pooled_model.stan"))

## // Index value and observations.
## data {
##   int<lower = 1> N;    // Number of observations.
##   vector[N] rating;         // Vector of observations.
## }
##
## // Parameters.
## parameters {
##   real beta;             // Mean of the regression.
##   real<lower = 0> sigma; // Variance of the regression.
## }
##
## // Regression.
## model {
##   // Priors.
##   target += normal_lpdf(beta| 0,50);
##   target += normal_lpdf(sigma| 20,50);
##
##   // Likelihood.
##   target += normal_lpdf(rating| beta,sigma);
## }
## generated quantities{
##
##   vector[N] y_rep;
##   for (n in 1:N) {
##       y_rep[n] = normal_rng(beta,sigma);
##   }
## }
```

We now fit the model.

```
dat2 <- list(N = 67, rating = data$ratings)
fit_pm <- stan("pooled_model.stan",  data=dat2, iter=5000)
```

Explore the output from the model

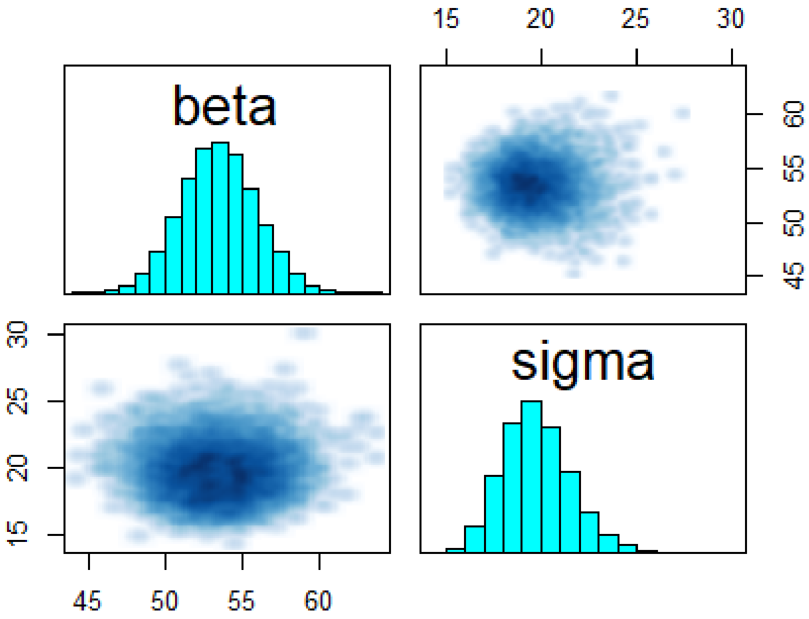
Print the output of the fit

```
round((summary(fit_pm)$summary[1:2,]),3)
```

```
##         mean se_mean    sd   2.5%    25%    50%    75%  97.5%    n_eff Rha
t
## beta  53.422   0.026 2.448 48.734 51.781 53.385 55.048 58.281 9136.611
1
## sigma 19.731   0.020 1.765 16.587 18.487 19.615 20.842 23.589 8060.108
1
```

Plot data for distribution visualization

```
pairs(fit_pm, pars = c("beta","sigma"))
```



Hierarchical model

We now write the pooled model where there is no distinction made between which coffee shop each rating was for.

$$T = normal(20,50)$$

$$\sigma = normal(20,50)$$

$$\mu_g = normal(0,50)$$

$$\beta = normal(\mu, T)$$

$$rating_i = normal(\beta_g, \sigma)$$

Where,

i varies from 1 to N(number of observations)

g is the number of groups(café)

Implement the model in stan

We now write a stan file hirar_model.stan for our new model.

```
# Loading Stan file for the code
writeLines(readLines("hirar_model.stan"))
```

```
## // Index values and observations.
## data {
##   int<lower = 1> N;               // Number of observations.
##   int<lower = 1> G;               // Number of groups.
##   vector[N] rating;                    // Vector of observations.
##   int<lower = 1, upper = G> g[N]; // Vector of group assignments.
## }
##
## // Parameters and hyperparameters.
## parameters {
##   real mu;                        // Mean of the population model.
##   real<lower = 0> tau;            // Variance of the population model.
##   vector[G] beta;                 // Vector of group intercepts.
##   real<lower = 0> sigma;          // Variance of the likelihood.
## }
##
## // Hierarchical regression.
## model {
##   // Priors.
##   target += normal_lpdf(mu| 0,50);
##   target += normal_lpdf(tau| 20,50);
##
##   // Prior.
##   target += normal_lpdf(sigma| 20,50);
##
##   // Population model and likelihood.
##   target += normal_lpdf(beta| mu,tau);
##
##   for (n in 1:N) {
##     target += normal_lpdf(rating[n]| beta[g[n]],sigma);
##
##   }
## }
## generated quantities{
##
##   vector[N] y_rep;
##   for (n in 1:N) {
##       y_rep[n] = normal_rng(beta[g[n]],sigma);
##   }
## }
```

We now fit the model.

```
fit_hm <- stan("hirar_model.stan",  data=dat1, iter=5000)
```

Explore the output from the model

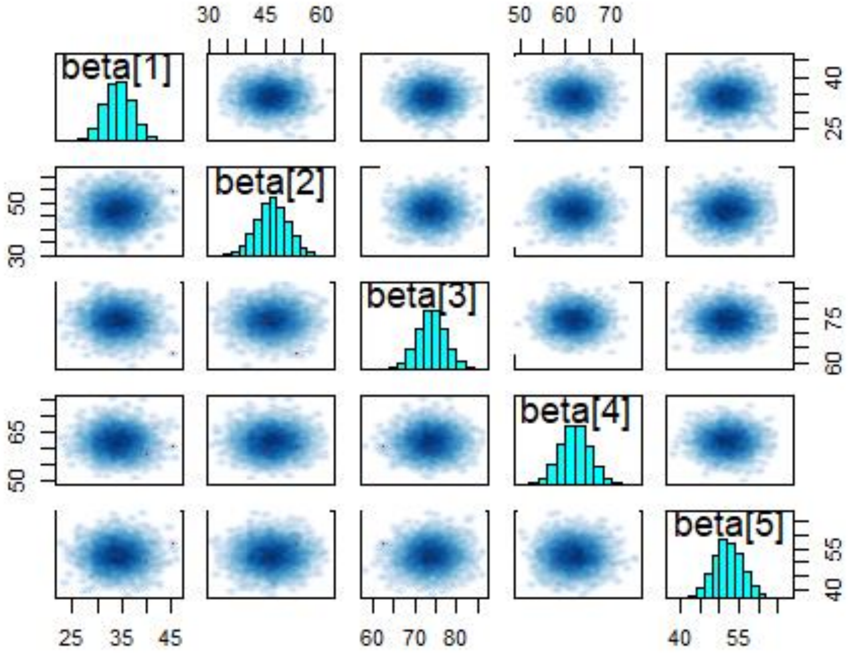
Print the output of the fit

```
round((summary(fit_hm)$summary[1:7,]),3)
```

```
##            mean se_mean     sd   2.5%    25%    50%    75%  97.5%     n_eff
## mu       51.380   0.187 11.765 25.755 45.985 52.076 57.842 72.738  3975.907
## tau      22.929   0.189 12.119  9.627 14.938 19.773 27.069 56.652  4100.529
## beta[1] 34.181   0.028  3.014 28.296 32.141 34.165 36.199 40.081 11947.569
## beta[2] 46.774   0.038  4.177 38.529 44.010 46.776 49.549 54.953 12391.002
## beta[3] 73.993   0.032  3.481 67.060 71.691 73.990 76.332 80.894 12116.241
## beta[4] 61.871   0.029  3.222 55.503 59.682 61.909 64.019 68.150 12764.584
## beta[5] 52.161   0.037  3.951 44.409 49.570 52.105 54.823 59.948 11209.028
##           Rhat
## mu       1.000
## tau      1.001
## beta[1] 1.000
## beta[2] 1.000
## beta[3] 1.000
## beta[4] 1.000
## beta[5] 1.000
```

Plot data for distribution visualization

```
pairs(fit_hm, pars = c("beta[1]","beta[2]","beta[3]","beta[4]","beta[5]"))
```



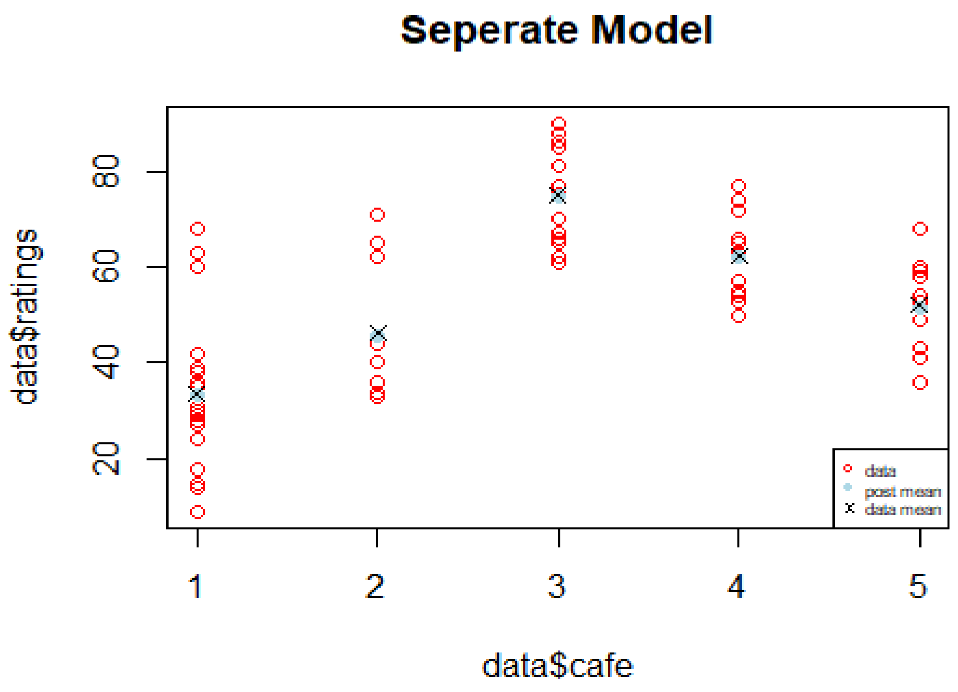
The distribution of the beta values signifies the normal spread of group data. From the summary of the model fit of all the 3 models, we can clearly notice that the standard deviation of café - 2 is higher when compared to others, which leads to poor prediction of the data.

This is due to the uneven spread of number of ratings on each café. This affects the prediction for café – 2 when compared to others.

Plot of data for Model Comparison

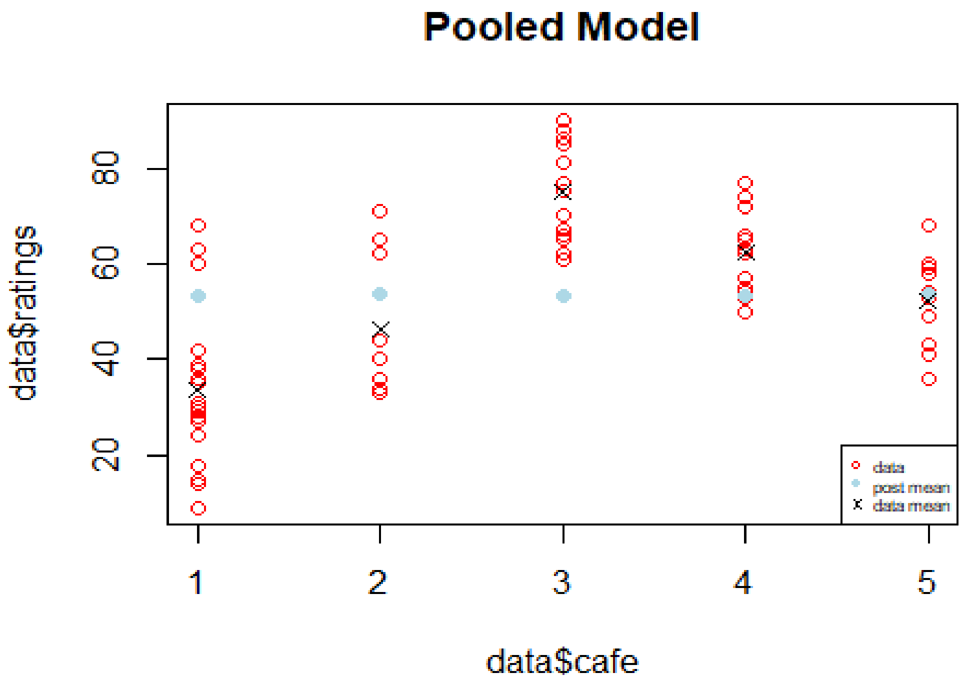
Separate Model

```
posterior1 <- as.data.frame(fit_sm, pars=c("y_rep"))
p1 <- colMeans(as.data.frame(posterior1))

plot(data$cafe, data$ratings , type="p", col="red", main = "Seperate Model")
points( tapply(p1, data$cafe, mean), col="lightblue", pch=19)
points( tapply(data$ratings, data$cafe, mean), col="black", pch=4)
legend("bottomright", legend=c("data","post mean", "data mean"), col=c("red",
"lightblue", "black"), pch=c(1,19,4), cex=0.5)
```



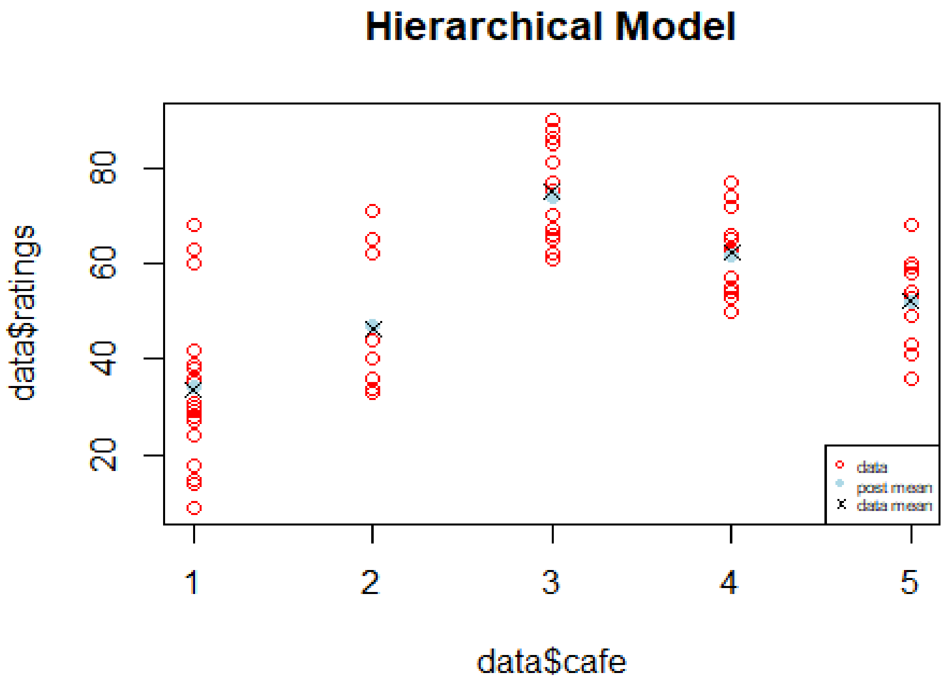
Pooled Model

```
posterior3 <- as.data.frame(fit_pm, pars=c("y_rep"))
p3 <- colMeans(as.data.frame(posterior3))

plot(data$cafe, data$ratings , type="p", col="red", main = "Pooled Model")
points( tapply(p3, data$cafe, mean), col="lightblue", pch=19)
points( tapply(data$ratings, data$cafe, mean), col="black", pch=4)
legend("bottomright", legend=c("data","post mean", "data mean"), col=c("red",
"lightblue", "black"), pch=c(1,19,4), cex=0.5)
```



Hierarchical Model

```
posterior2 <- as.data.frame(fit_hm, pars=c("y_rep"))
p2 <- colMeans(as.data.frame(posterior2))

plot(data$cafe, data$ratings , type="p", col="red",main = "Hierarchical Model
")
points( tapply(p2, data$cafe, mean), col="lightblue", pch=19)
points( tapply(data$ratings, data$cafe, mean), col="black", pch=4)
legend("bottomright", legend=c("data","post mean", "data mean"), col=c("red",
"lightblue", "black"), pch=c(1,19,4), cex=0.5)
```



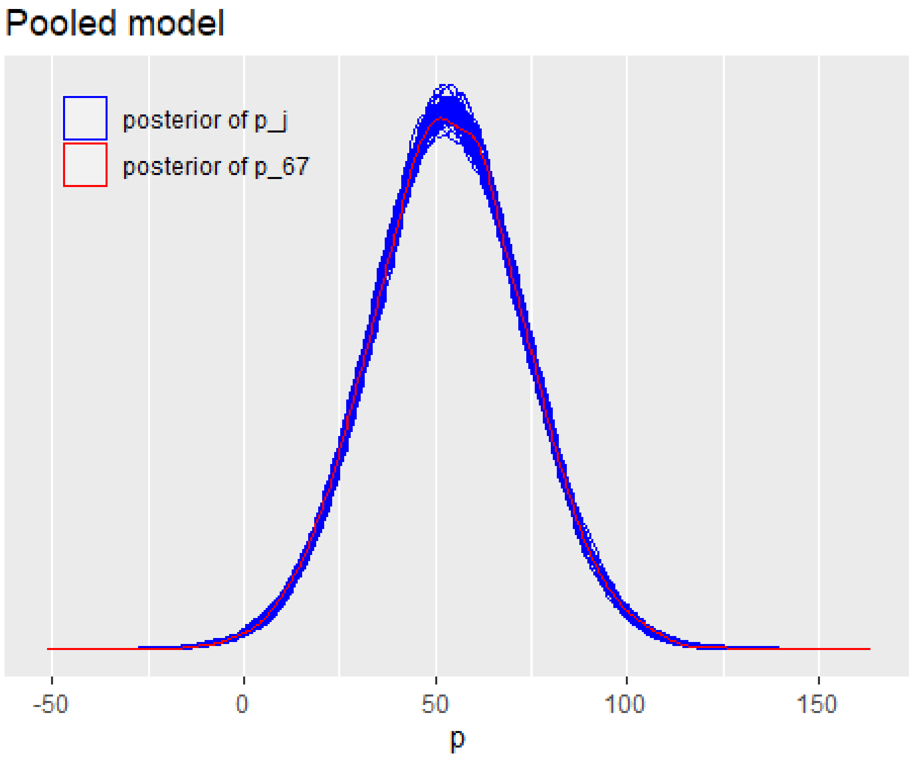
We can observe from the graphs above that Separate Model performs the best, even though there is not much difference observed between Separate and Hierarchical Model.

The Pooled Model does not make sense to this dataset because it does no have a prior information about the café for which the ratings are primarily given for. Hence, we rule out Pooled model for model comparison. This can also be visualized in the plot for the model and its variation from the data mean.

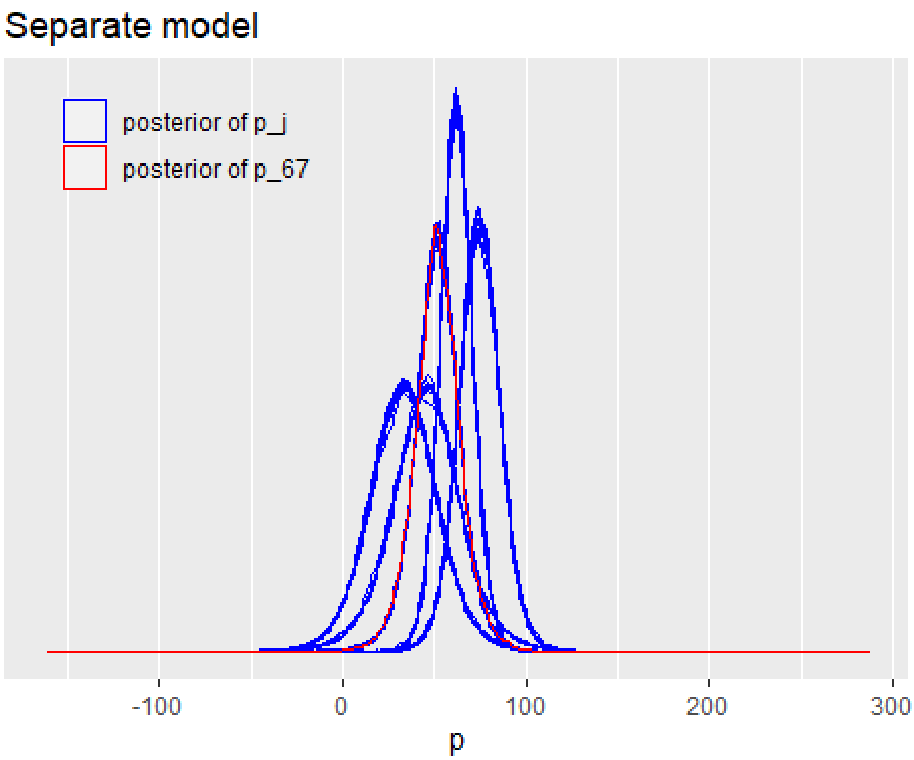
Keeping the standard deviation same for all the café does not make complete sense due to the lack of parameters considered for this analysis. Treating different café like in the Separate Model highlights this issue and provides a better prediction. Parameters like, same customers ratings at different café, items tasted/served etc. amounts to a significant difference.

Model Comparison

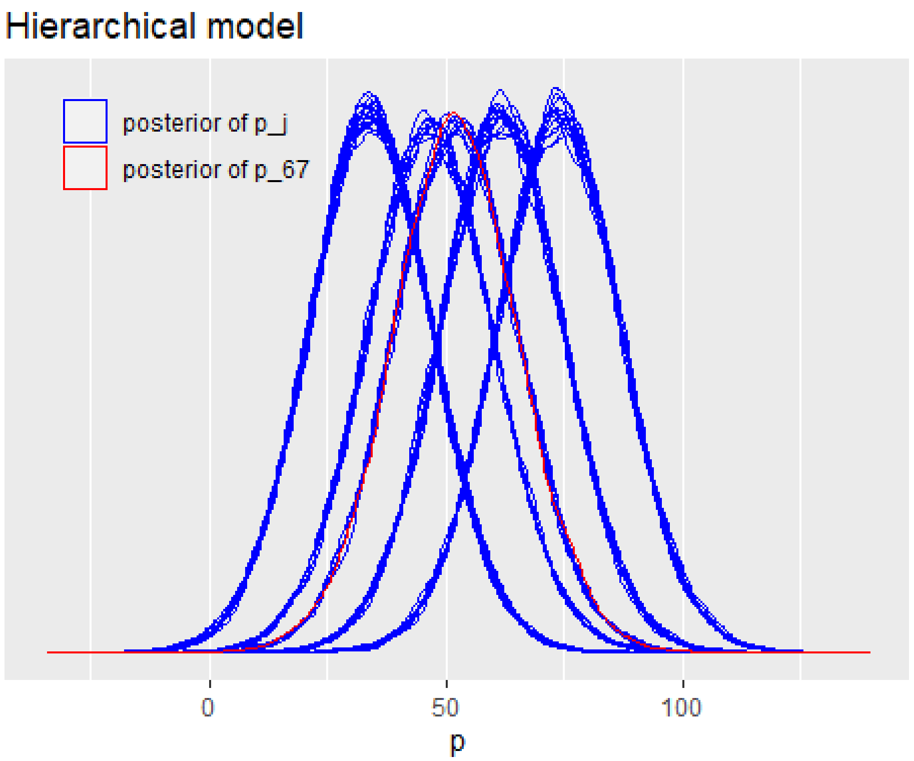
```
posterior <- as.data.frame(fit_sm, pars=c("y_rep"))
p1 <- stack(posterior)

posterior <- as.data.frame(fit_hm, pars=c("y_rep"))
p2 <- stack(posterior)

posterior <- as.data.frame(fit_pm, pars=c("y_rep"))
p3 <- stack(posterior)

labs1 <- paste('posterior of', c('p_j', 'p_67'))
plot_sep1 <- ggplot(data = p1) +
  geom_density(aes(values, color = (ind=='y_rep[67]'), group = ind)) +
  labs(x = 'p', y = '', title = 'Separate model', color = '') +
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('blue','red'), labels = labs1) +
  theme(legend.background = element_blank(), legend.position = c(0.2,0.9))

labs1 <- paste('posterior of', c('p_j', 'p_67'))
plot_sep2 <- ggplot(data = p2) +
  geom_density(aes(values, color = (ind=='y_rep[67]'), group = ind)) +
  labs(x = 'p', y = '', title = 'Hierarchical model', color = '') +
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('blue','red'), labels = labs1) +
  theme(legend.background = element_blank(), legend.position = c(0.2,0.9))

labs1 <- paste('posterior of', c('p_j', 'p_67'))
plot_sep3 <- ggplot(data = p3) +
  geom_density(aes(values, color = (ind=='y_rep[67]'), group = ind)) +
  labs(x = 'p', y = '', title = 'Pooled model', color = '') +
  scale_y_continuous(breaks = NULL) +
  scale_color_manual(values = c('blue','red'), labels = labs1) +
  theme(legend.background = element_blank(), legend.position = c(0.2,0.9))

plot_sep3
```



```
plot_sep1
```


Separate model
posterior of p_j
posterior of p_67
-100
0
100
200
300
p

```
plot_sep2
```


Hierarchical model
posterior of p_j
posterior of p_67
0
50
100
p

Computing the (Log) Marginal Likelihoods

```
# compute log marginal likelihood via bridge sampling for Separate Model
sm.bridge <- bridge_sampler(fit_sm, silent = TRUE)

# compute log marginal likelihood via bridge sampling for Hierarchical Model
hm.bridge <- bridge_sampler(fit_hm, silent = TRUE)

# Comparison using Bayes factor
bayes_factor(hm.bridge, sm.bridge)

## Estimated Bayes factor in favor of x1 over x2: 3536.18346
```

High difference in bayes factor obtained (>100) implies that the model H1 = Separate Model provides the best fit for the above considered dataset.

Although pooled is a more straightforward model, it may not be suited for the data. It does not capture the significance of the different cafés. Which makes it a very poor predictive model for ratings.

A separate model implies that each café has its own distribution, which divides the data and prevents it from being used by other cafés.

A hierarchical model is halfway between the two approaches, with each café having its own set of parameters but sharing the same hyper-parameters. The ratings given by other customers on different cafes surely seems to not effect the rating of a particular café. This might be due to the fact that prior knowing of the rating of a previous café does not necessarily affect the rating of the next café. Each café has it’s own set of menu, customer base and personal taste acts as a significant variable when considering such data set.

Hence, we choose **Separate Model to be the best Model** for this particular dataset which has a lack of much more information.