BayesAnalysisAssignment4

Gavin Connolly

25/04/2022

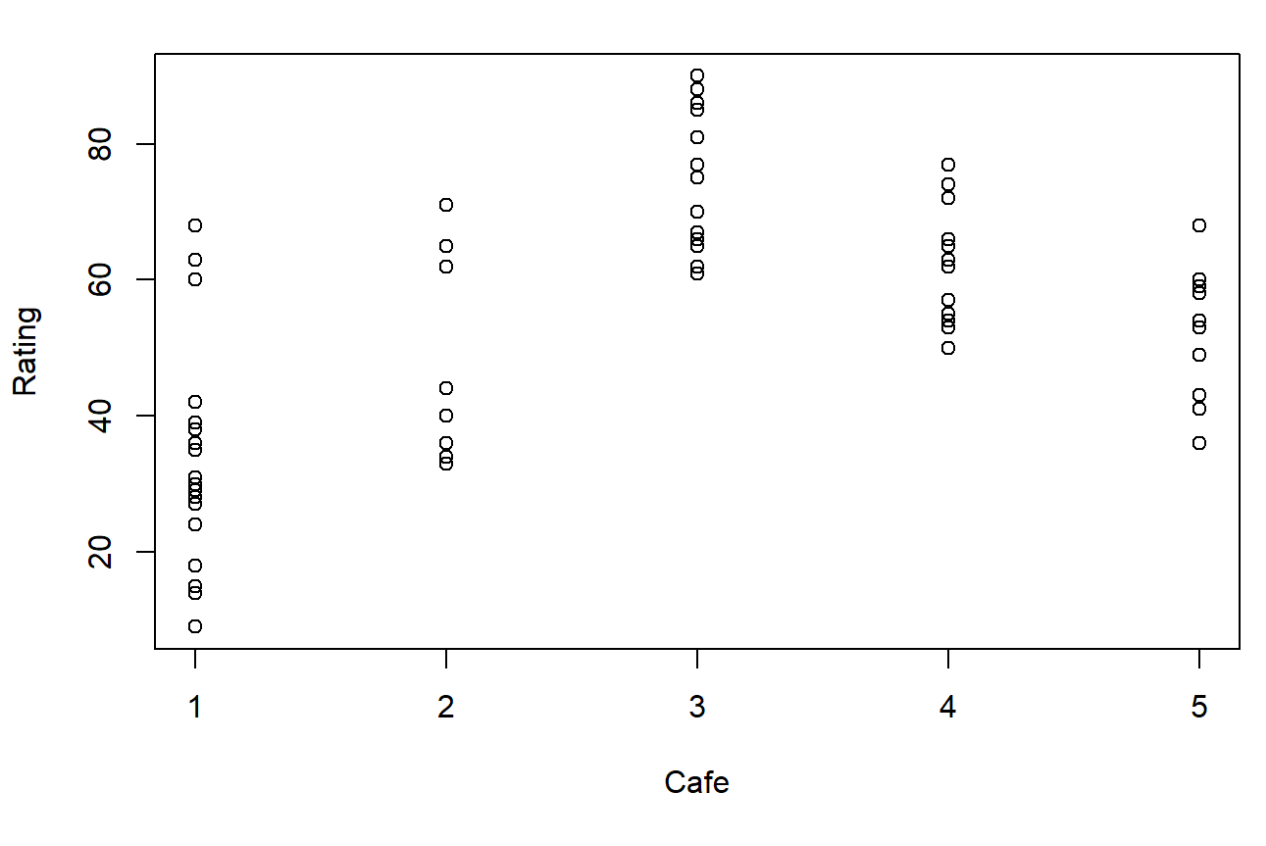
Importation of Data

Import data relating to the ratings of various cafés. Create a plot of the data to get basic idea as to their distribution.

```
SEED <- 151515 # set random seed for reproducability

data <- read.csv("coffee.csv", header = TRUE)

# Initial analysis of data
str(data)
```

```
## 'data.frame':    67 obs. of  2 variables:
##  $ ratings: int  18 31 42 60 9 29 15 35 28 27 ...
##  $ cafe   : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
plot(data[,2], data[,1], xlab = "Cafe", ylab = 'Rating')
```



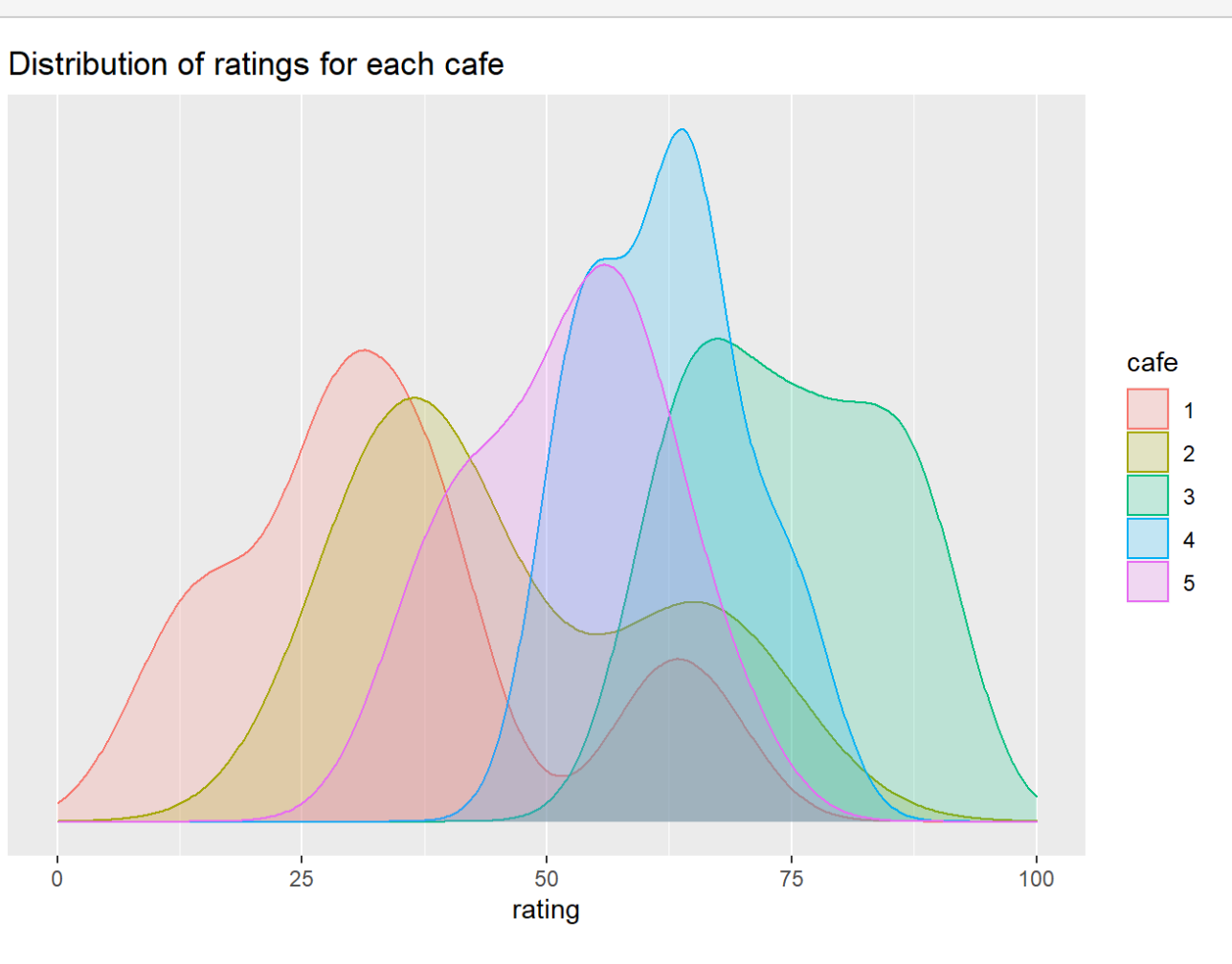
Plot Density of Distributions

From the above plot of the distribution of ratings amongst the 5 different cafés, we see that the ratings appear to be different for each of the 5 cafés, with Café 3 in particular standing out as having much more of its density concentrated within the upper end of the scale. The variance of observed ratings also seems to vary between cafés, with cafés 1, 2 & 3 having flatter distributions than either café 4 or 5. In addition to this, there seems to be an element of bimodality in the distributions of the first 2 cafés which is not seen within the others. It is of note however that the number of observations for each café is relatively low, ranging from 9 observations for café 2 to 18 observations for café 1, so we should not infer too much from these distributions.

```
# Plot the density of ratings for each of the cafes
plot <- matrix(data = NA, nrow = nrow(data), ncol = 2)
plot$ratings <- data$ratings
```

```
## Warning in plot$ratings <- data$ratings: Coercing LHS to a list
```

```
plot$cafe <- as.character(data$cafe)

ggplot(as.data.frame(plot), aes(ratings, fill = cafe, colour = cafe)) +
  geom_density(alpha = 0.2) + theme_grey()+
  labs(x = 'rating', y = '', title = 'Distribution of ratings for each cafe') +
  scale_y_continuous(breaks = NULL) + xlim(0, 100)
```


Distribution of ratings for each cafe

Store the ratings for each café individually so we can create individual models for each café.

```
# store the ratings associated with each cafe individually
ratings1 = data$ratings[which(data[,2]==1)]
ratings2 = data$ratings[which(data[,2]==2)]
ratings3 = data$ratings[which(data[,2]==3)]
ratings4 = data$ratings[which(data[,2]==4)]
ratings5 = data$ratings[which(data[,2]==5)]
```

Seperate Model

The first model we assessed was the model which treats each of the ratings distributions as being entirely unrelated to one another. This model assumes that the distribution of ratings for one café is completely uninformative as to the distributions of the other cafés.

Stan code for seperate models

```
## data{
##  int<lower=1> N;
##  array[N] real<lower=0,upper=100> rating;
##  array[N] int<lower=1, upper=5> cafe;
## }
## parameters{
##  real a;
##  real<lower=0> sigma; // assume uniform prior on sigma for all cafes
## }
## transformed parameters{
##  real mu;
##  mu = inv_logit(a)*100;
## }
## model{
##  a ~ normal(0,3);
##  rating ~ normal(mu, sigma);
## }
## generated quantities{
##  array[N] real y_rep;
##  for ( i in 1:N ) {
##          y_rep[i] = normal_rng( mu, sigma );
##  }
## }
```

```
cafe1 <- list(rating = ratings1, N = length(which(data[,2]==1)), cafe = data$cafe[which(data[,2]==1)])
fit_cafe1 <- stan("coffee.stan",  data=cafe1, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 1.0 seconds.
```

```
#print(fit_cafe1, probs = c(0.10, 0.5, 0.9))

cafe_sep_1 <- as.data.frame(fit_cafe1, pars=c("mu"))
p1 <- stack(as.data.frame(cafe_sep_1))
p1$ind = "cafe1"

cafe2 <- list(rating = ratings2, N = length(which(data[,2]==2)), cafe = data$cafe[which(data[,2]==2)])
fit_cafe2 <- stan("coffee.stan",  data=cafe2, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 0.9 seconds.
```

```
#print(fit_cafe2, probs = c(0.10, 0.5, 0.9))

cafe_sep_2 <- as.data.frame(fit_cafe2, pars=c("mu"))
p2 <- stack(as.data.frame(cafe_sep_2))
p2$ind = "cafe2"

cafe3 <- list(rating = ratings3, N = length(which(data[,2]==3)), cafe = data$cafe[which(data[,2]==3)])
fit_cafe3 <- stan("coffee.stan",  data=cafe3, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 1.0 seconds.
```

```
#print(fit_cafe3, probs = c(0.10, 0.5, 0.9))

cafe_sep_3 <- as.data.frame(fit_cafe3, pars=c("mu"))
p3 <- stack(as.data.frame(cafe_sep_3))
p3$ind = "cafe3"

cafe4 <- list(rating = ratings4, N = length(which(data[,2]==4)), cafe = data$cafe[which(data[,2]==4)])
fit_cafe4 <- stan("coffee.stan",  data=cafe4, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 1.0 seconds.
```

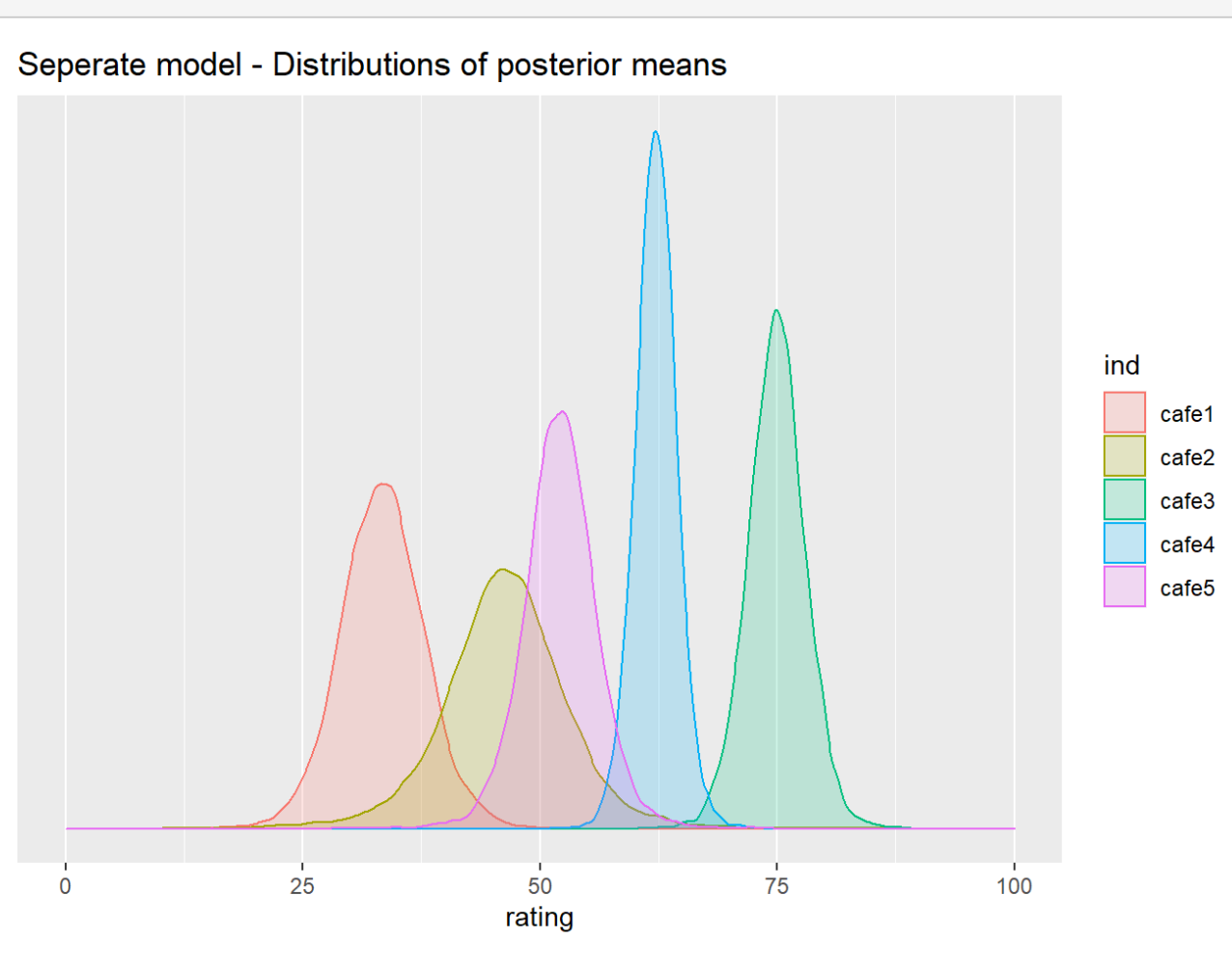
```
#print(fit_cafe4, probs = c(0.10, 0.5, 0.9))

cafe_sep_4 <- as.data.frame(fit_cafe4, pars=c("mu"))
p4 <- stack(as.data.frame(cafe_sep_4))
p4$ind = "cafe4"

cafe5 <- list(rating = ratings5, N = length(which(data[,2]==5)), cafe = data$cafe[which(data[,2]==5)])
fit_cafe5 <- stan("coffee.stan",  data=cafe5, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 0.8 seconds.
```

```
#print(fit_cafe5, probs = c(0.10, 0.5, 0.9))

cafe_sep_5 <- as.data.frame(fit_cafe5, pars=c("mu"))
p5 <- stack(as.data.frame(cafe_sep_5))
p5$ind = "cafe5"
```

The distributions for each café seem to be quite distinct from one another, with both the centres & variances of each of the posterior mean distributions clearly differing between distributions. This would indicate that the ratings for each café do not come from the same underlying distribution.

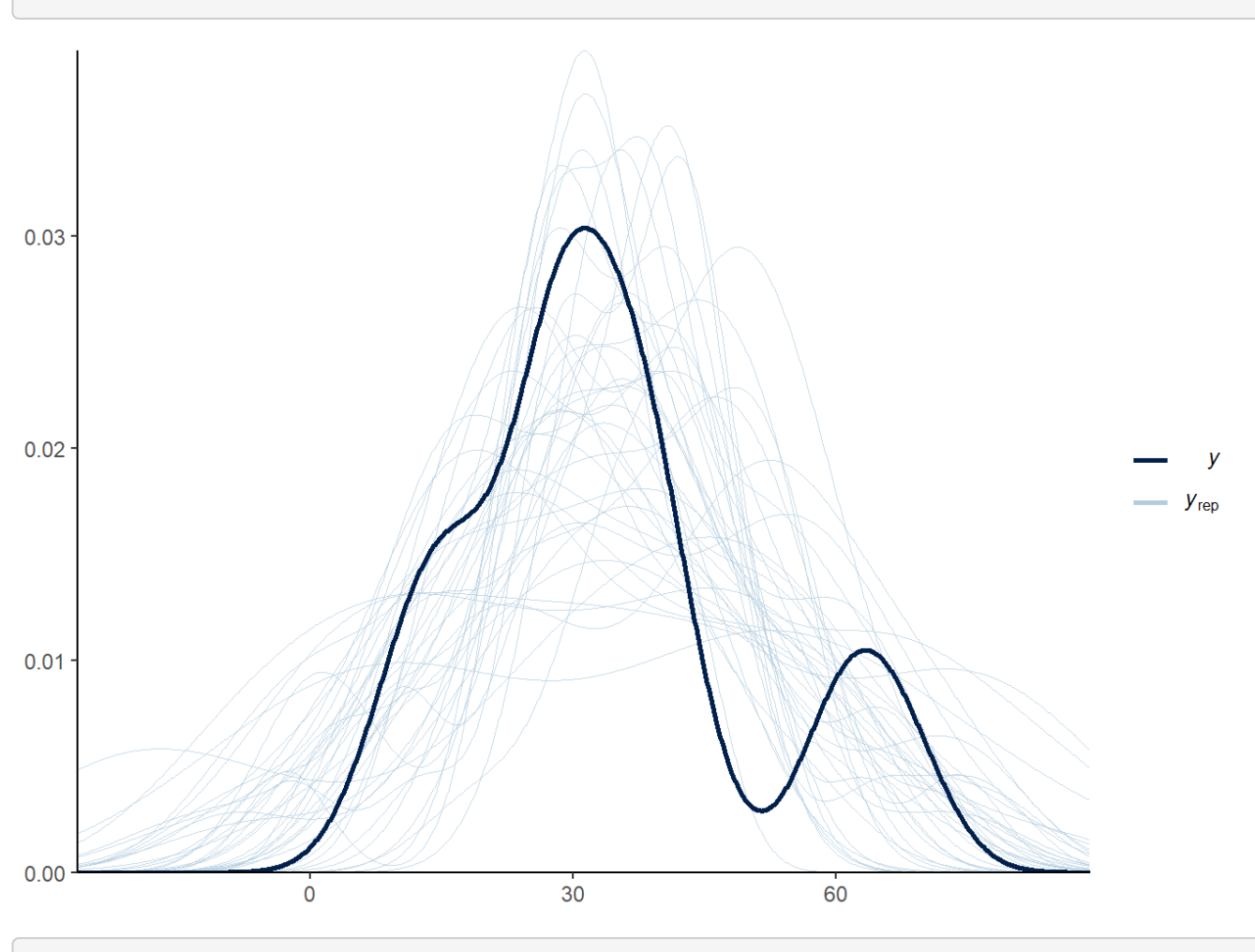
```
posterior_sep <- rbind(p1,p2, p3, p4, p5)
ggplot(posterior_sep, aes(values, fill = ind, colour = ind)) +
  geom_density(alpha = 0.2) + theme_grey()+
  labs(x = 'rating', y = '', title = 'Seperate model - Distributions of posterior means') +
  scale_y_continuous(breaks = NULL) + xlim(0, 100)
```


Seperate model - Distributions of posterior means

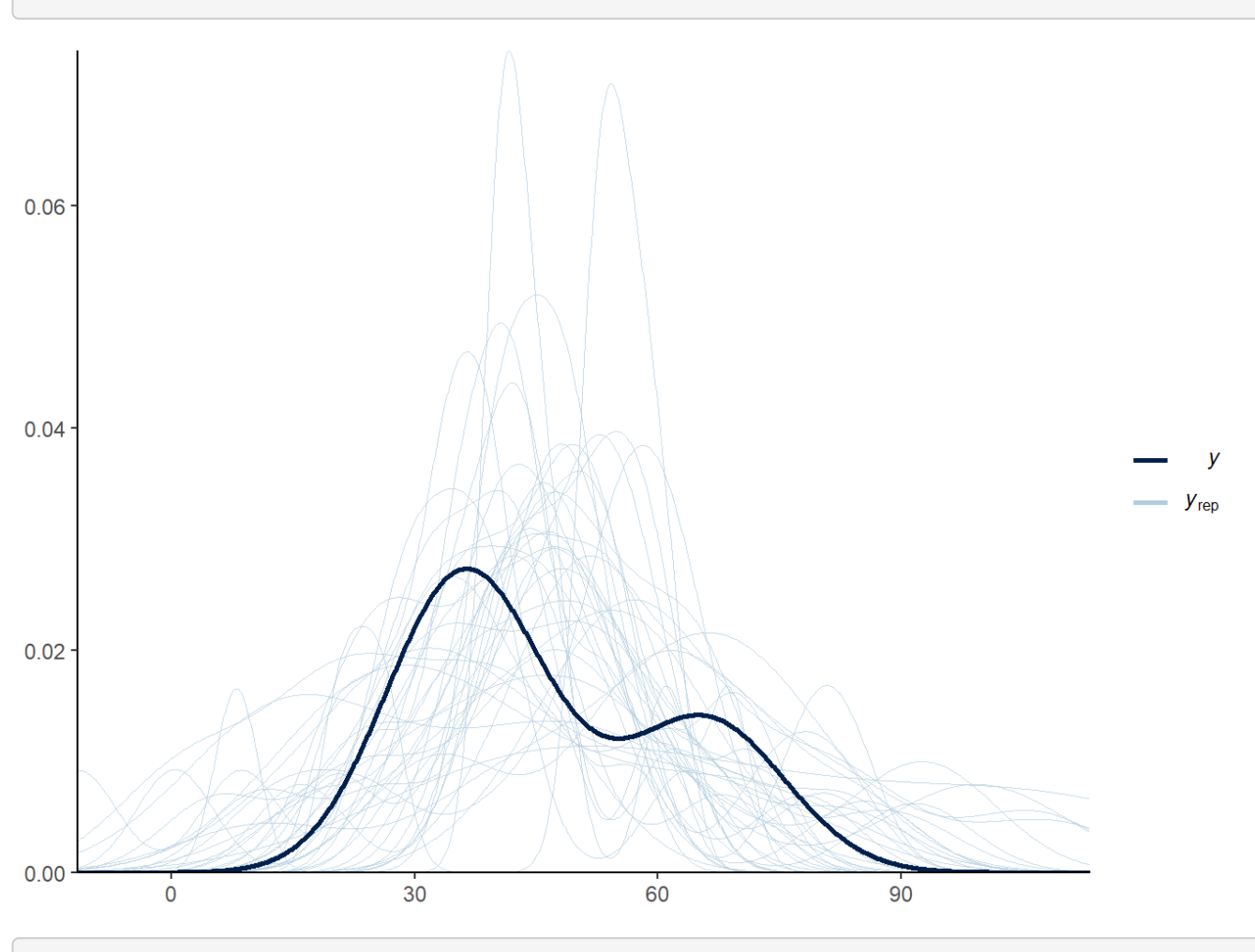
We see from these plots that the replicated values tend to have their density mostly concentrated around the same areas as the observed values, but they do not line up very closely in terms of the shape of the distribution. This is likely due to the relatively low sample sizes involved in the dataset, leading to replicated distributions which vary a lot from sample to sample.

Overall, I feel that the separate model is quite a good fit for the data, as each café seems to have a relatively distinct distribution, and the replicated samples line up quite well with the observed values when considering how small the sample sizes involved are.

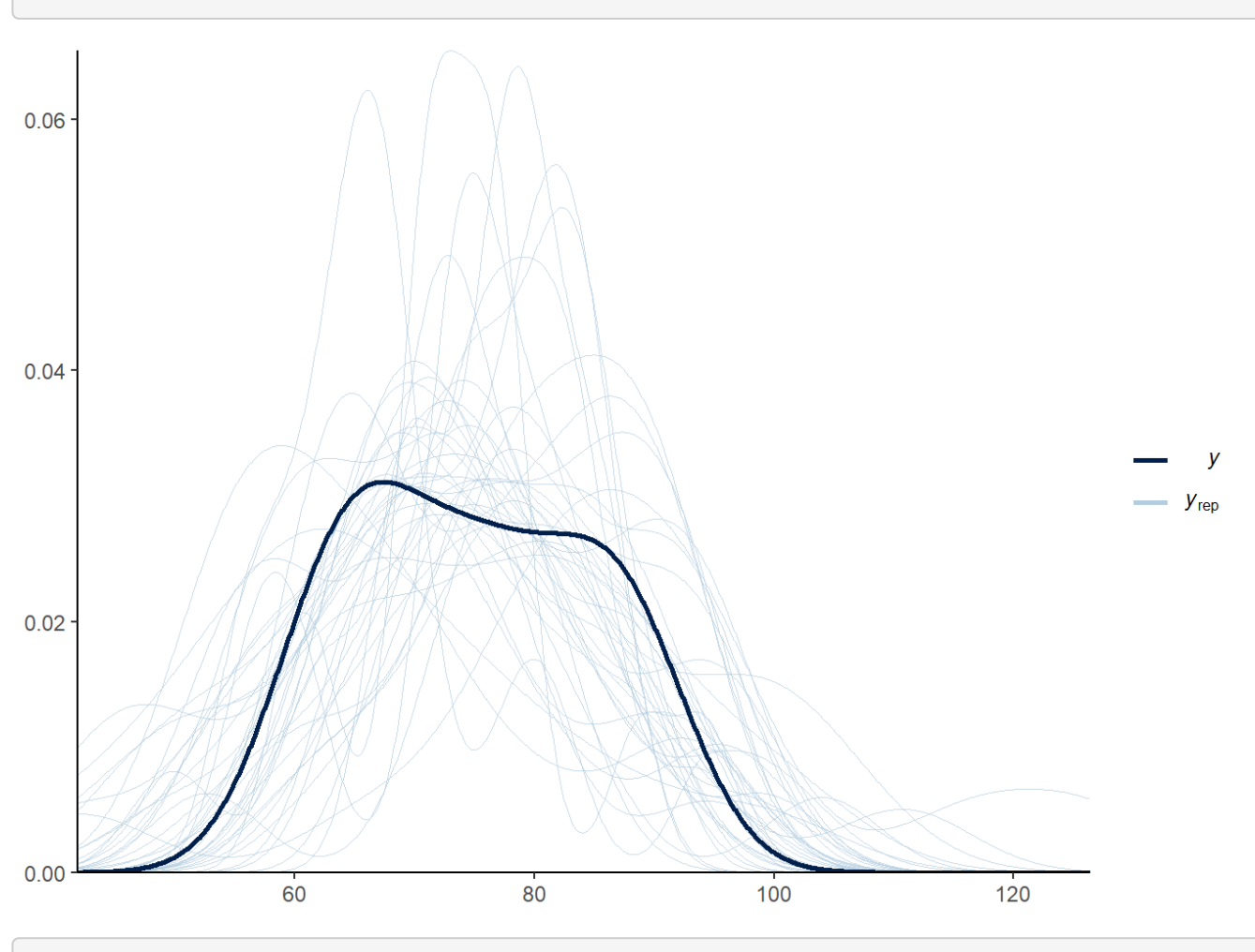
```
y_rep1 <- as.matrix(fit_cafe1, pars=c("y_rep"))
ppc_dens_overlay(data[which(data[,2]==1),1], y_rep1[1:40,])+theme_classic()
```



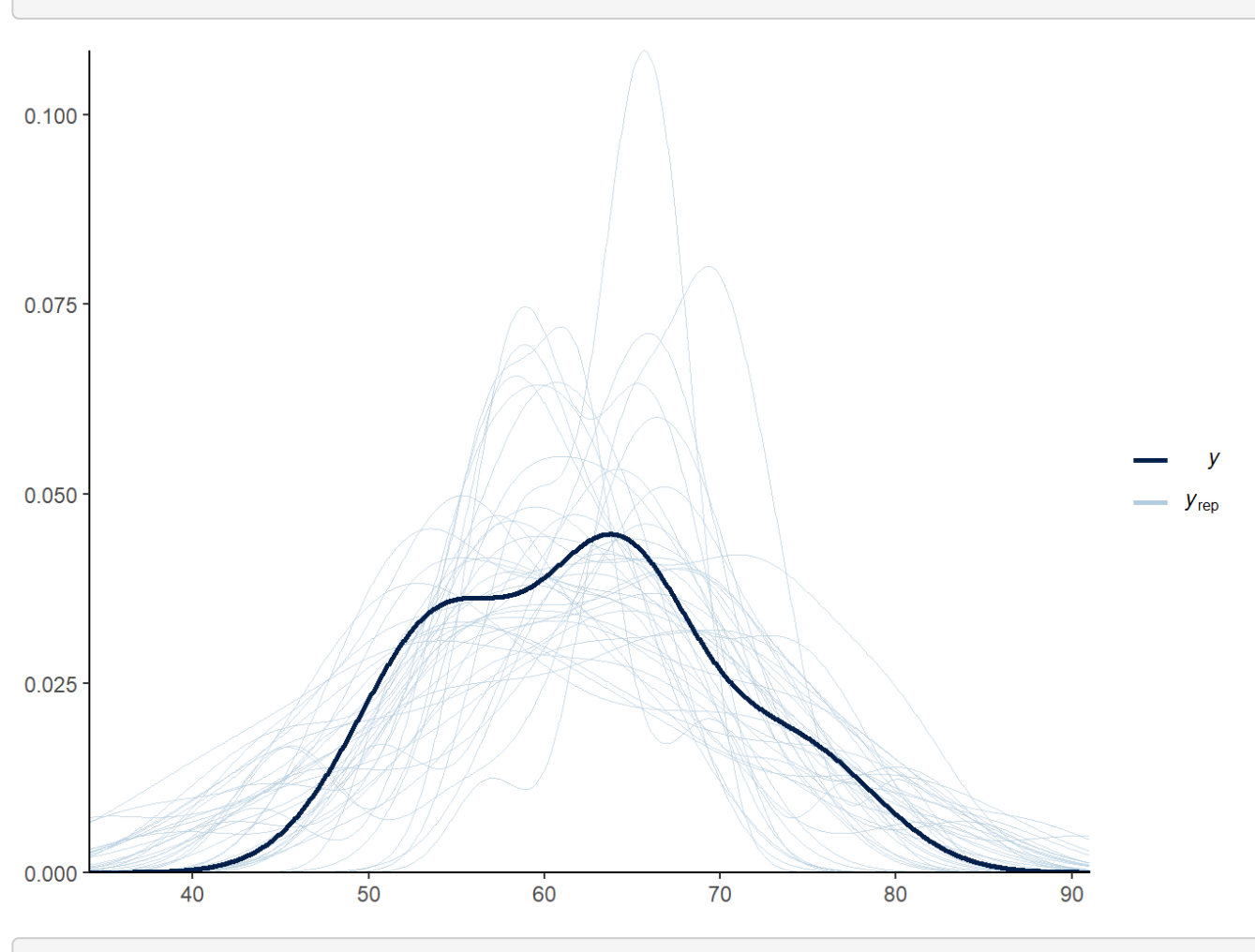
```
y_rep2 <- as.matrix(fit_cafe2, pars=c("y_rep"))
ppc_dens_overlay(data[which(data[,2]==2),1], y_rep2[1:40,])+theme_classic()
```



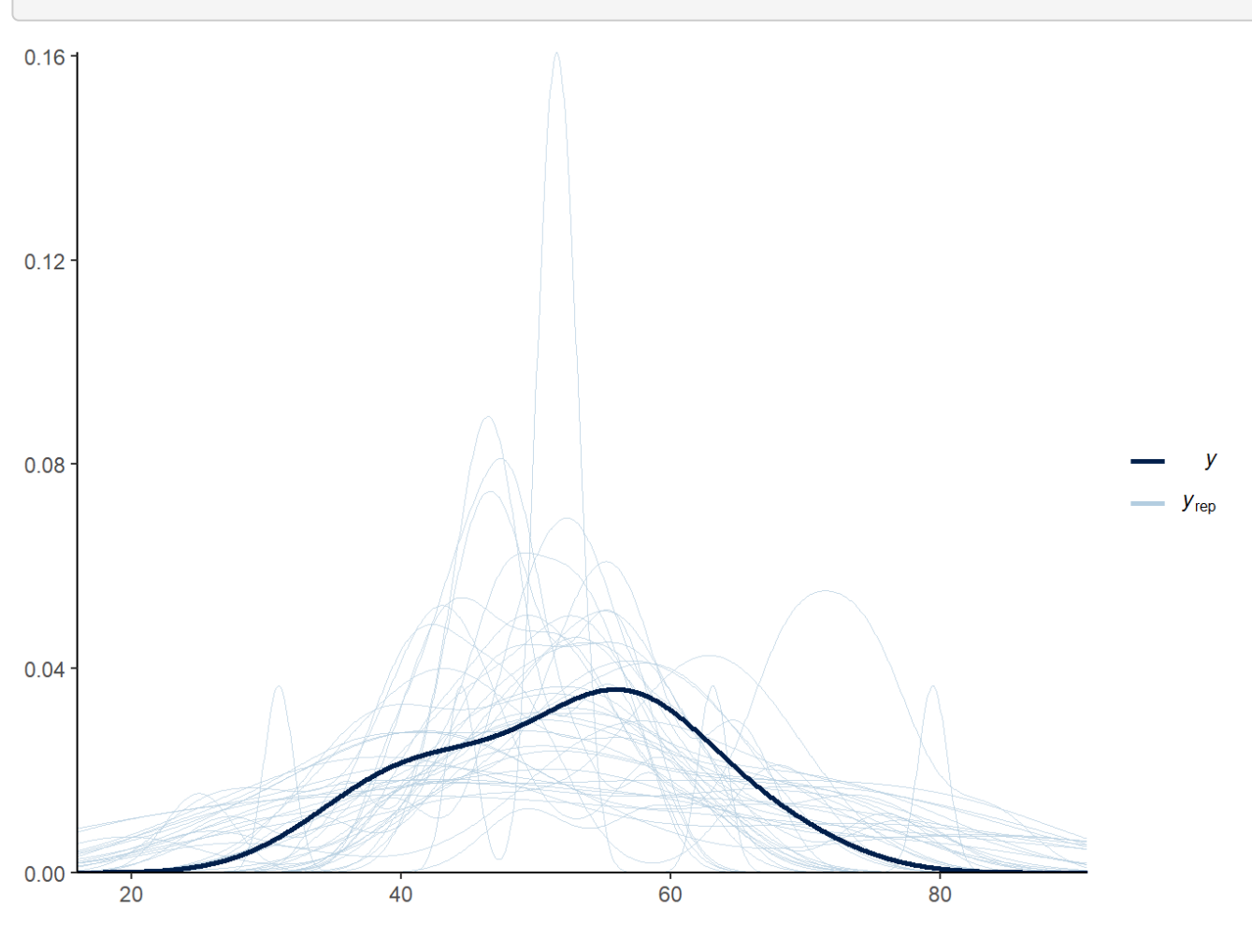
```
y_rep3 <- as.matrix(fit_cafe3, pars=c("y_rep"))
ppc_dens_overlay(data[which(data[,2]==3),1], y_rep3[1:40,])+theme_classic()
```



```
y_rep4 <- as.matrix(fit_cafe4, pars=c("y_rep"))
ppc_dens_overlay(data[which(data[,2]==4),1], y_rep4[1:40,])+theme_classic()
```



```
y_rep5 <- as.matrix(fit_cafe5, pars=c("y_rep"))
ppc_dens_overlay(data[which(data[,2]==5),1], y_rep5[1:40,])+theme_classic()
```



Pooled Model

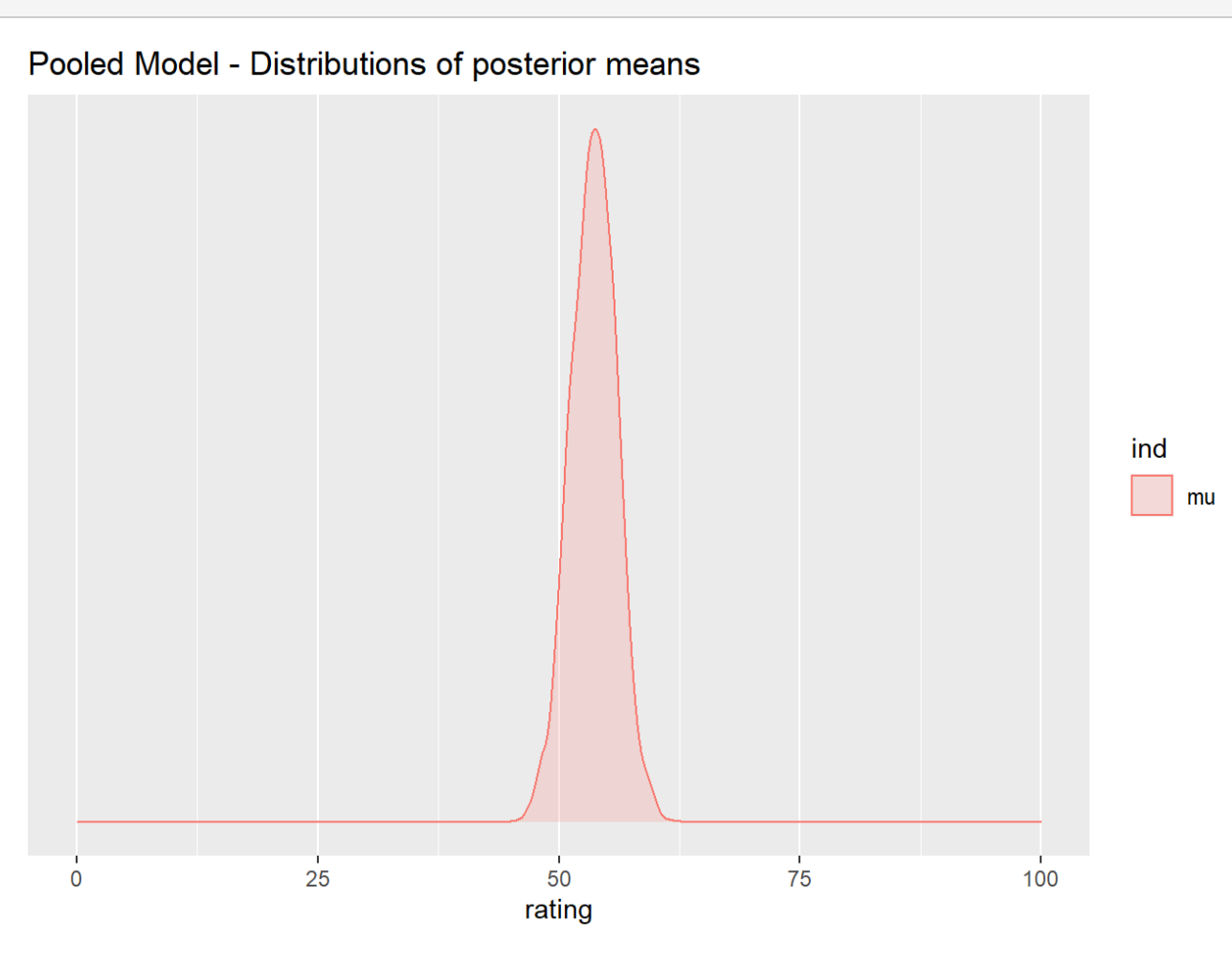
The second model under consideration assumes that each of the ratings is derived from the same underlying distribution & that there is no difference in the distributions between each of the cafés.

As this model assumes a single distribution for each café, we see a much more concentrated distribution for our estimates in this case. The distribution is roughly centred on the overall average rating of 53.55, with the estimated standard deviation being roughly equal to that of the data (19.37).

```
cafe_pooled <- list(rating = data$ratings, N = nrow(data), cafe = data$cafe)
fit_pooled <-stan("coffee.stan",  data=cafe_pooled, iter=5000, refresh = 0)
```

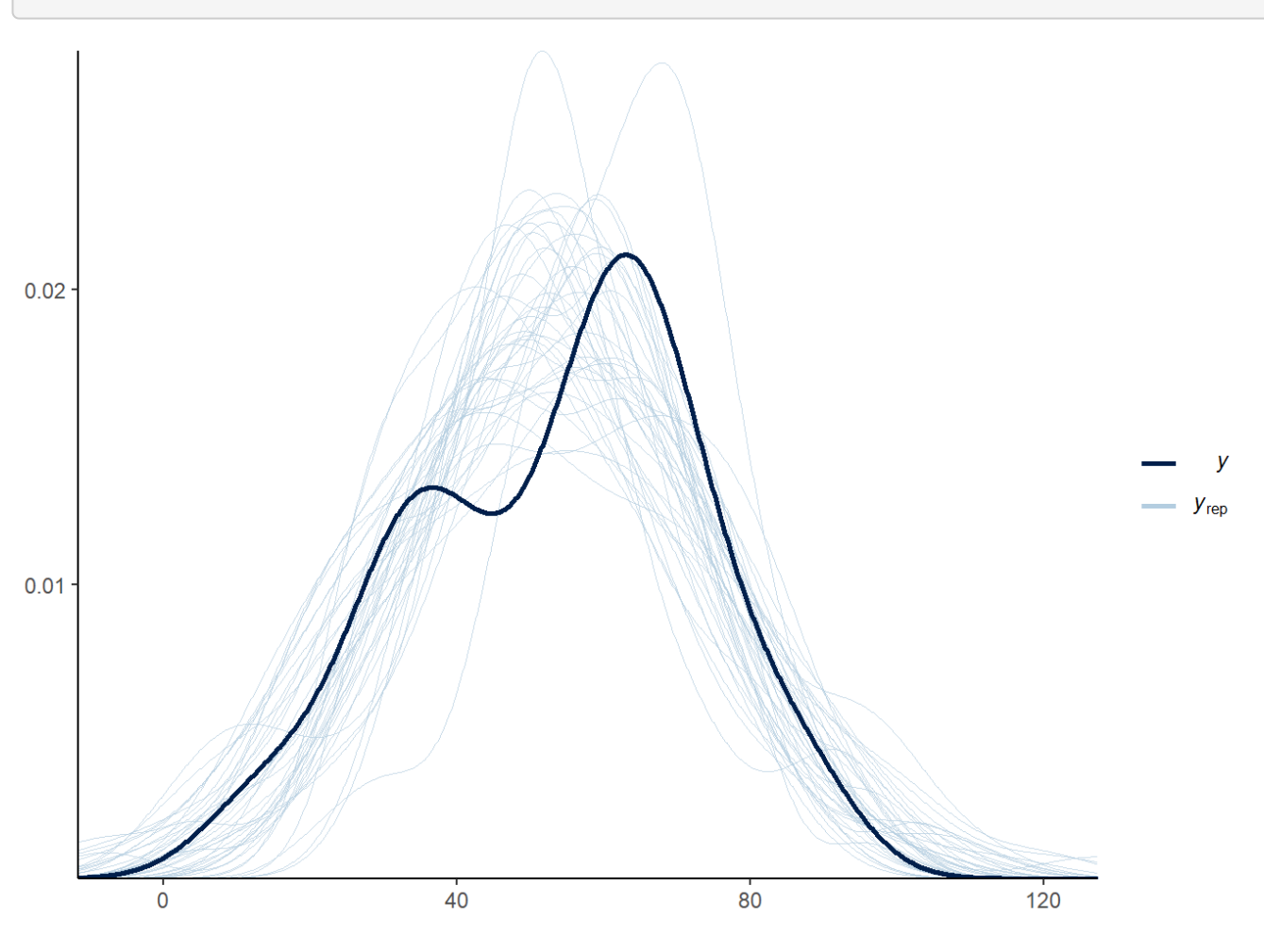
```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 0.6 seconds.
```

```
#print(fit_pooled, probs = c(0.10, 0.5, 0.9))

pooled <- as.data.frame(fit_pooled, pars=c("mu"))
p6 <- stack(as.data.frame(pooled))

ggplot(p6, aes(values, fill = ind, colour = ind)) +
  geom_density(alpha = 0.2) + theme_grey()+
  labs(x = 'rating', y = '', title = 'Pooled Model - Distributions of posterior means') +
  scale_y_continuous(breaks = NULL) + xlim(0, 100)
```


Pooled Model - Distributions of posterior means

A plot of the density of our replicated ratings values once again shows that the simulated values share their density within the same area as the observed data with the distribution generally being closer in shape to the observed values than was seen in the separate models, likely due to the larger sample size of the replications. We do note however that the simulated values do not seem to share the bimodality seen in the observed values. The pooled model does not seem a bad fit for the data overall, though as it assumes the same distribution for each of the cafés, something which would probably go against our intuition that different cafés would likely have a varying standards of service/quality, I would have concerns at how well this model would perform when predicting the ratings for individual cafés.

```
y_rep_pooled <- as.matrix(fit_pooled, pars=c("y_rep"))
ppc_dens_overlay(data[,1], y_rep_pooled[1:40,])+theme_classic()
```



The final model under consideration is a hierarchical model, which allows the mean rating to vary between the cafés, while also assuming that the means are drawn from a population distribution with the same hyperparameters for each café. This means that the distribution of ratings for each café will inform us to some extent about the distribution of ratings of the other cafés, while also allowing the strength of this relation to be updated based on the results of our MCMC sampling. In our case, the magnitude of the τ parameter will inform us as to the strength of this relation, with high values indicating that there is a high level of commonality in the distributions of the ratings of each café, and low values indicating that the distributions for each café are quite distinct from one another.

Stan code for hierarchical model

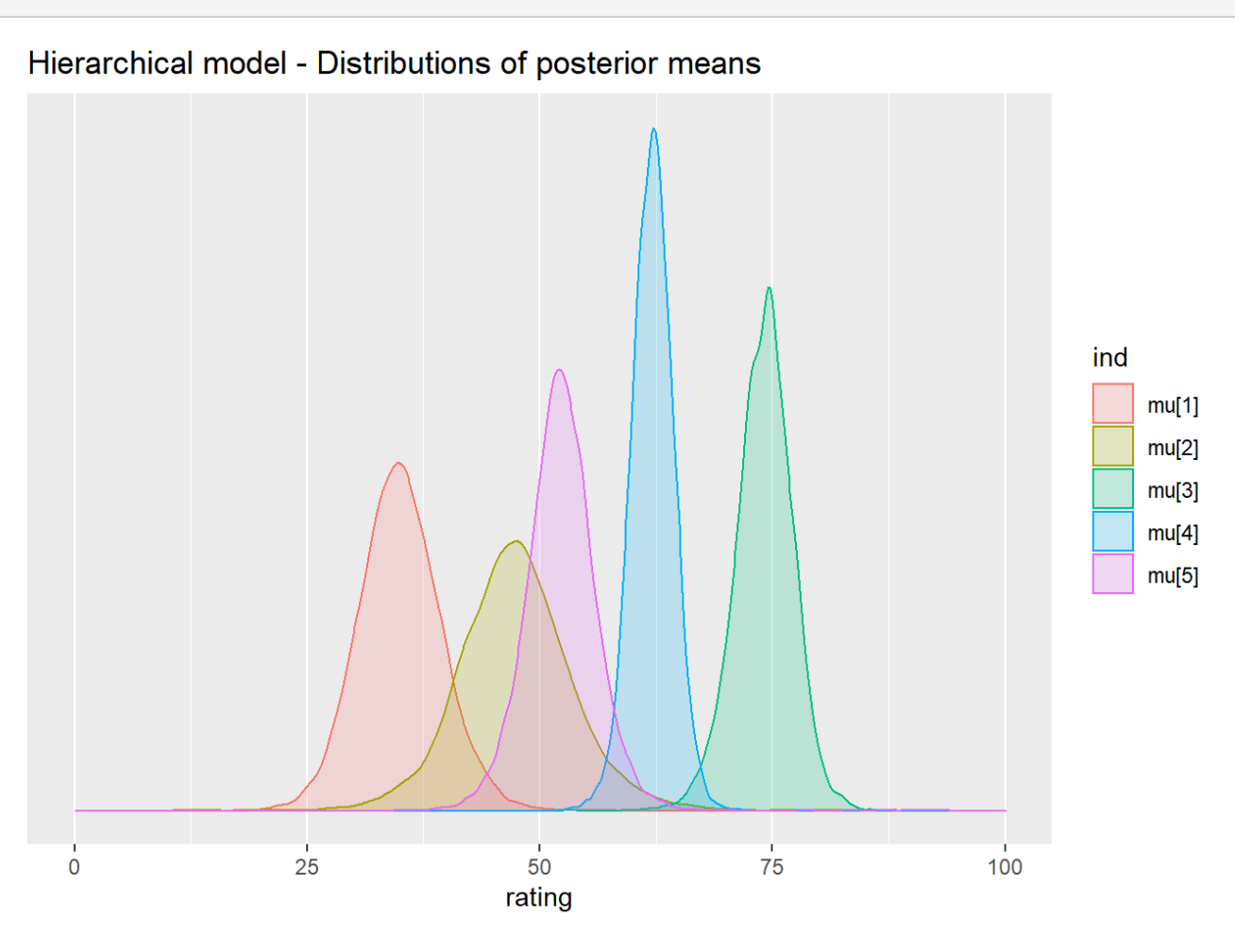
```
## data{
##  int<lower=1> N1;
##  int<lower=1> N2;
##  int<lower=1> N3;
##  int<lower=1> N4;
##  int<lower=1> N5;
##  vector[N1] ratings1;
##  vector[N2] ratings2;
##  vector[N3] ratings3;
##  vector[N4] ratings4;
##  vector[N5] ratings5;
## }
##
## parameters{
##  array[5] real<lower=0> sigma;
##  vector[5] a;
##  real a_bar;
##  real<lower=0> tau;
## }
##
## transformed parameters{
##  real<lower=0, upper=100> mu_bar;
##  mu_bar = inv_logit(a_bar)*100;
##  vector[5] mu;
##  mu = inv_logit(a)*100;
## }
##
## model{
##
##  tau ~ exponential( 1 );
##  a_bar ~ normal( 0 , 1 );
##  a ~ normal( a_bar , tau );
##  sigma ~ uniform(0, 100);
##
##  ratings1 ~ normal(mu[1], sigma[1]);
##  ratings2 ~ normal(mu[2], sigma[2]);
##  ratings3 ~ normal(mu[3], sigma[3]);
##  ratings4 ~ normal(mu[4], sigma[4]);
##  ratings5 ~ normal(mu[5], sigma[5]);
## }
##
## generated quantities{
##  vector[N1] y_rep1;
##  vector[N2] y_rep2;
##  vector[N3] y_rep3;
##  vector[N4] y_rep4;
##  vector[N5] y_rep5;
## for (i in 1:N1){
##  y_rep1[i] = normal_rng(mu[1], sigma[1]);
##  }
## for (i in 1:N2){
##  y_rep2[i] = normal_rng(mu[2], sigma[2]);
##  }
## for (i in 1:N3){
##  y_rep3[i] = normal_rng(mu[3], sigma[3]);
##  }
## for (i in 1:N4){
##  y_rep4[i] = normal_rng(mu[4], sigma[4]);
##  }
## for (i in 1:N5){
##  y_rep5[i] = normal_rng(mu[5], sigma[5]);
##  }
## }
```

It is apparent from our graph that the distributions of our posterior estimates for each café under the hierarchical model are very similar to those observed under our separate models. Indeed, observing the estimate of our τ hyperparameter (0.80) we see that it is quite close to 0, which would indicate that the distributions for each café are completely independent from one another.

```
cafe_hier <- list(ratings1 = ratings1, ratings2 = ratings2, ratings3 = ratings3, ratings4 = ratings4, ratings5 =
ratings5,
                  N1 = length(ratings1), N2 = length(ratings2), N3 = length(ratings3), N4 = length(ratings4), N5
= length(ratings5), cafe = data$cafe)

fit_hier <-stan("cafe_hierarchical.stan",  data=cafe_hier, iter=20000, refresh = 0)
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 6.6 seconds.
```

```
#print(fit_hier, probs = c(0.10, 0.5, 0.9))

post_hier <- as.data.frame(fit_hier, pars=c("mu"))
posterior_hier <- stack(as.data.frame(post_hier))

ggplot(posterior_hier, aes(values, fill = ind, colour = ind)) +
  geom_density(alpha = 0.2) + theme_grey()+
  labs(x = 'rating', y = '', title = 'Hierarchical model - Distributions of posterior means') +
  scale_y_continuous(breaks = NULL) + xlim(0, 100)
```


Hierarchical model - Distributions of posterior means

Similarly to the separate model, we once again see that the replicated values do not line up particularly well with the observed values in terms of the shape of their distribution, likely for the same reasons as before, namely the low sample size for each café. Overall though, the replicated values seem to have density in the same areas as the observed values.

We see from a comparison between the posterior mean estimates that the hierarchical model has the effect of bringing the posterior mean estimates for each café closer to the overall mean of 53.55. This should regularise for differences in the distributions of ratings due to error due to the particular sample of customers involved in rating each particular café & error in our estimates due the low sample size.

```
mean(data$ratings)
```

```
## [1] 53.55224
```

```
var(data$ratings)/nrow(data)
```

```
## [1] 5.601214
```

```
c(mean(ratings1), mean(ratings2), mean(ratings3), mean(ratings4), mean(ratings5))
```

```
## [1] 33.66667 46.44444 74.85714 62.18750 52.10000
```

```
c(var(ratings1)/length(ratings1), var(ratings2)/length(ratings2), var(ratings3)/length(ratings3), var(ratings4)/l
ength(ratings4), var(ratings5)/length(ratings5))
```

```
## [1] 15.222222 25.864198  7.086342  3.918490  9.743333
```

```
post_sep <- cbind(cafe_sep_1, cafe_sep_2, cafe_sep_3, cafe_sep_4, cafe_sep_5)
colnames(post_sep) <- c("mu1", "mu2", "mu3", "mu4", "mu5")
# examine posterior means for the average rating for each cafe under each model
colMeans(post_sep)
```

```
##      mu1      mu2      mu3      mu4      mu5
## 33.36547 46.32243 74.94352 62.23394 52.13834
```

```
mean(pooled$mu)
```

```
## [1] 53.56266
```

```
colMeans(post_hier)
```

```
##    mu[1]    mu[2]    mu[3]    mu[4]    mu[5]
## 34.95473 47.38663 74.09356 62.08450 52.29061
```

```
# compare these figures to the overall mean
abs(colMeans(post_sep)-mean(data$ratings))
```

```
##       mu1       mu2       mu3       mu4       mu5
## 20.186767  7.229809 21.391280  8.681702  1.413895
```

```
abs(mean(pooled$mu)-mean(data$ratings))
```

```
## [1] 0.01042067
```

```
abs(colMeans(post_hier)-mean(data$ratings))
```

```
##     mu[1]     mu[2]     mu[3]     mu[4]     mu[5]
## 18.597507  6.165611 20.541317  8.532261  1.261632
```

```
# examine the variance of estimates under each model
apply(post_sep, 2, var)
```

```
##       mu1       mu2       mu3       mu4       mu5
## 19.205595 43.623937  9.231688  4.887535 15.376698
```

```
var(pooled$mu)
```

```
## [1] 5.513733
```

```
apply(post_hier, 2, var)
```

```
##     mu[1]     mu[2]     mu[3]     mu[4]     mu[5]
## 19.196753 36.208612  9.478116  4.993576 14.330473
```

Conclusion

To conclude, I feel that the hierarchical model is the best fit for modelling the ratings of the cafés as the model seems to fit relatively well with the data, even despite the low number of observations, and because intuitively, it makes sense to assume that the rating habits of customers will share some level of commonality across the various cafés, even if the strength of this commonality has not proved to be particularly great in our case. I feel like the hierarchical model should lend itself better than either the separate or pooled models in terms of its potential application in analysing the ratings distributions for cafés other than the 5 considered in the creation of our models, as it accounts for these common rating behaviours, while also acknowledging that these behaviours are unlikely to be exactly the same for every café. It is also less likely to suffer from the overfitting of a separate model, or the underfitting of a complete pooling model, instead striking a balance between the two approaches which should generally yield better results. It should be acknowledged however, that it is impossible to properly discern to which degree the observed differences in the distribution of ratings for the various cafés are due to random chance versus tangible differences in the underlying distribution. For this reason, we should remain open to reconsidering our conclusion if future data provides evidence in support of either potentiality.