STAT461 Project Report

Aayushi Verma (84233071)

due Fri 15/10/21, Week 11

The project topic chosen is Bayesian Mixture Analysis.

1. Introduction

One problem of interest in statistics is that of clustering and mixture models. Given a dataset, to draw some inference, one may attempt to identify clusters, or sub-populations, of data points which share some common characteristics. The process of identifying clusters, or sub-populations within an overall population dataset is known as **mixture modelling**. Mixture models are useful for making statistical inferences about the characteristics of these sub-populations/clusters, given the population dataset.

Given a dataset with N number of data points, the aim of classical mixture modeling is to identify the closest cluster centre for each point, replace each cluster centre by the average of the closest data points, and repeat until the model converges. This is known as K-means clustering. This method of clustering often involves approximating the mixtures as Gaussian distributions. On the other hand, Bayesian mixture modelling uses Markov Chain Monte Carlo (MCMC) methods to find mixtures in the distribution with an unknown number of components.

Some real-world applications of mixture modelling from studies in various fields include identifying quantitative traits on chromosomes in biology, distinguishing between spam email and regular email in computer science, identification of fire ant infestations in habitats in Brisbane, identifying fuel spill areas in the soil in Antarctica, identifying galaxy populations in astronomy, resolution of images in computer science, and many more applications. Mixture modelling is a very versatile tool to analyse any kind of dataset which may have sub-populations within. This is also a good tool in machine learning, where the algorithms can be taught to cluster data points within certain mixture components, for example.

2. Theory and Methodology

As stated in the Introduction, the difference between the approach to classical and Bayesian cluster/mixture modeling and analysis is the usage of MCMC methods to determine the number of mixture components. For this project, we will consider the finite mixture problem, which deals with a finite number (may be known or unknown) of mixture components, say K mixture components, in the data.

Let's consider the general formulation of a mixture model. We start by assuming that in a sample, there are $i = 1, ..., n$ components, each of which belong to one of K sub-populations, or clusters, in the data. To simplify the mixture model, we introduce a latent variable z_i, which gives the index of the subcluster each observation belongs to. Then, we can write the distribution of an observation y_i for each component dependent on the latent variable z_i as:

$$y_i | z_i \sim f(\theta_{z_i}, \phi)$$

Here, $f(\theta, \phi)$ is a sampling distribution with parameters θ_k which corresponds to a subpopulation k for $k = 1, ..., K$, and ϕ which is a fixed, non-variable parameter. If we take the probability of a proportion of the population which belongs to a cluster k, i.e. the mixture weight, as:

$$Pr(z_i = k) = \pi_k$$

then the likelihood after marginalizing out the latent variable z_i is given by:

$$g(y|\pi, \theta, \phi) = \Sigma_{k=1}^{K} \pi_k f(y|\theta_k, \phi)$$

which is a finite mixture with K components. In the Bayesian framework of finite mixture models, the unknown parameters k, π and θ are each drawn from their own prior distributions. The prior for π is always taken to be a symmetric Dirichlet distribution, $\pi \sim D(\delta, ..., \delta)$ to allow uniform mixture weights per component. If we assume a normal distribution, then we have $f(y|\theta, \phi) = N(y|\theta, \phi)$, such that for each cluster k, the likelihood is given by: $(y_i|z_i = k) \sim N(\theta_k, \phi)$. However, this assumption of a normal distribution restricts the data to be assumed with a fixed shape. If we allow the mean to vary across clusters, then we have a more flexible mixture model:

$$g(y|\pi, \theta, \phi) = \Sigma_{k=1}^{K} \pi_k N(y|\theta_k, \phi)$$

A Bayesian finite mixture model as we just saw above, has multiple unknown parameters whose densities, priors and distributions are unknown and have to be estimated. One method of estimating the parameters, their priors and their distributions is using reverse-jump Markov Chain Monte Carlo simulation. Using the reverse jump MCMC methodology allows simulation of the posterior distribution of the parameter space(s), whether the components are known or unknown, using a *reversibility constraint* on moves, which is the main difference between ordinary MCMC and reverse jump MCMC. A brief overview of the Bayesian finite mixture modelling algorithm is presented below:

1. a jump from a model $\mathfrak{M}_1$ to a new model $\mathfrak{M}_2$ is proposed using reverse jump MCMC
2. the parameters k, π and θ are estimated by each being drawn from their prior distributions and the current values are updated
3. the posterior sample is generated from these parameters using reverse jump MCMC
4. a distribution is proposed for a new value y_{i+1} given the current value y_i
5. the distribution is calculated for the new value and either accepted or rejected with a certain probability
6. the steps are repeated until convergence is reached

To demonstrate the application of finite mixture modelling on a dataset, we will be using the "palmer-penguins" dataset, and analysing the penguins' bill depths and bill lengths. In this example of penguin bill depths vs bill lengths, we can represent our analysis model as:

$$y_i|k_i \sim N(\mu(k_i), \tau(k_i))$$

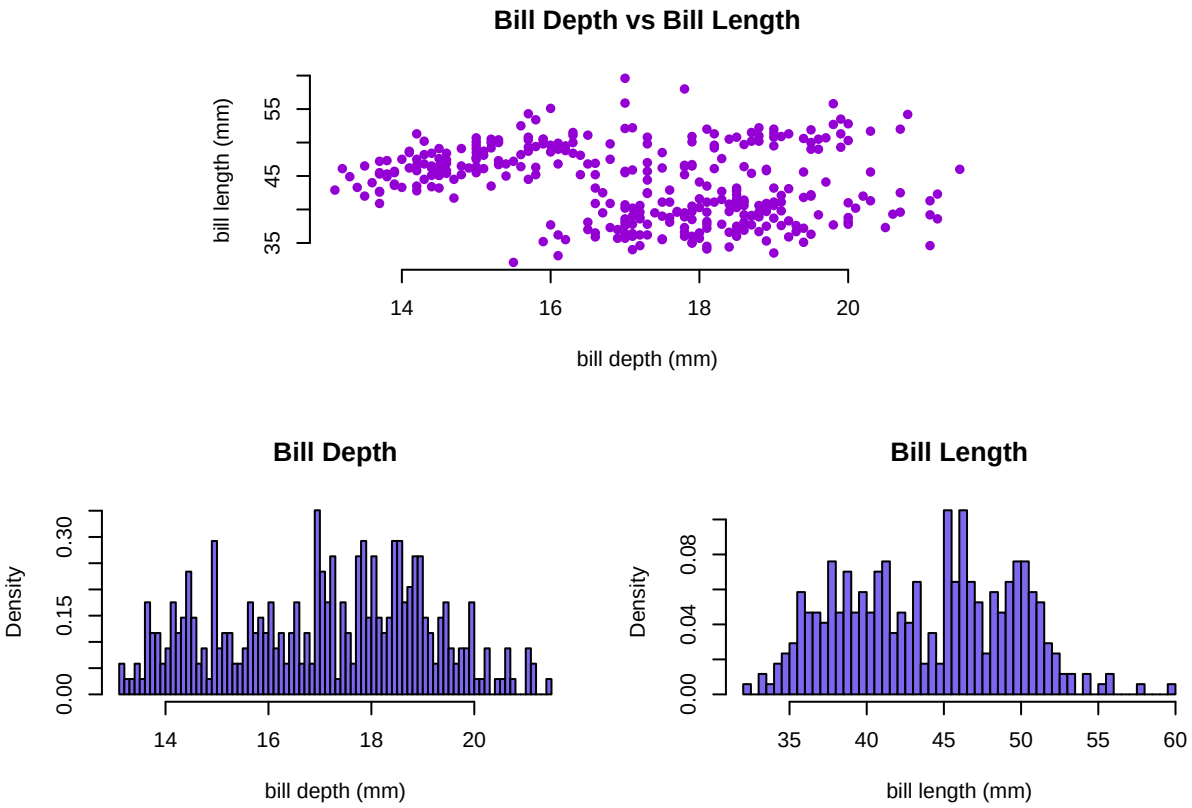
where y_i is a single observation of a penguin's bill depth (or length), k_i is the cluster to which the penguin for y_i belongs to, and μ and τ are the mean and variance of the data. We represent this distribution as a normal distribution. Its likelihood is given by:

$$g(y_i|\pi, k_i, \phi) = \Sigma_{i=1}^{K} \pi_i N(y_i|k_i, \phi)$$

The prior for π_i is the Dirichlet prior as given above and the prior for k_i is taken to be uniform. The prior of θ is unknown and has to be estimated using reverse jump MCMC. In the next section, we will find the posterior distribution of this model.

3. Example

To demonstrate how Bayesian cluster analysis/mixture modelling works, we will use the dataset "palmer-penguins" and use the R package "mixAK" to do the analysis. We will assume an unknown number of distributions, but let the maximum number of components be 10, and we will attempt to find clusters within the data for the penguins' bill depths and lengths. We start by initialising the data and doing some exploratory analysis. We then set parameters for running the MCMC simulation, and set a grid of values for evaluating predictive density for the models.



In the exploratory scatter plot, we observe that there are at least 2 clusters visible in the data, and the histograms for bill depth and bill length each appear to have at least 2 main peaks. We have kept the number of iterations for burn-in and keeping quite short to save computation time when compiling this report, however by increasing the number of iterations, a more accurate result may be found. We then set the priors. In this simulation, we are attempting to estimate the number of components using the reverse jump MCMC method. We will therefore now run mixAK's NMixMCMC function for finding the mixtures in the data. This function runs a MCMC model for an unknown density with a normal mixture for either known or unknown number of components. For a known number of components, the function will run Gibbs sampling MCMC, however since we are assuming an unknown number of components, the function uses reverse jump MCMC. Let's run 2 MCMC chains for comparison. The NMixMCMC function takes as input the dataset, a prior for which we have used a uniform prior with a maximum of 10 components and delta set to 1 for the mixture weights, and parameters for the reverse jump MCMC as well as the MCMC simulation parameters.

```
RJModel2 <- NMixMCMC(y0=my_dataset, prior=list(priorK="uniform", Kmax=10, delta=1),
                     RJMCMC=list(par.u1=c(2, 2),par.u2=c(2, 2),par.u3=c(1, 1)),
                     nMCMC=c(burn=5000, keep=20000, thin=10, info=10000),
                     scale=my_scale, PED=TRUE)
```

We now wish to find the number of mixtures, K, identified in the data in each chain. From the code below, we find the posterior probability for each chain for each of K components identified in the mixture. The K values will change every time this document is compiled. In most of the simulations run, the K-component with the highest proportion has been $K = 3$, which indicates with a strong probability that the number of mixtures identified in the sample is 3.

```
## [1] "Posterior probabilities for K for Chain 1:"
```

```
##
##       1       2       3       4       5       6       7       8       9
```

```
## 0.00155 0.04475 0.50905 0.31900 0.09910 0.02275 0.00340 0.00035 0.00005
```

```
## [1] "Posterior probabilities for K for Chain 2:"
```

```
##
##       1       2       3       4       5       6       7       8
## 0.01505 0.04420 0.47755 0.34670 0.09705 0.01645 0.00285 0.00015
```

From the reverse jump MCMC mixture analysis, we can derive the posterior distribution. Plots of the distribution and density are included in the Appendix, however we have summarised the results for Chain 1 for the bill depth and bill length posterior distributions here. A full output of statistics for both chains in included in the Appendix.

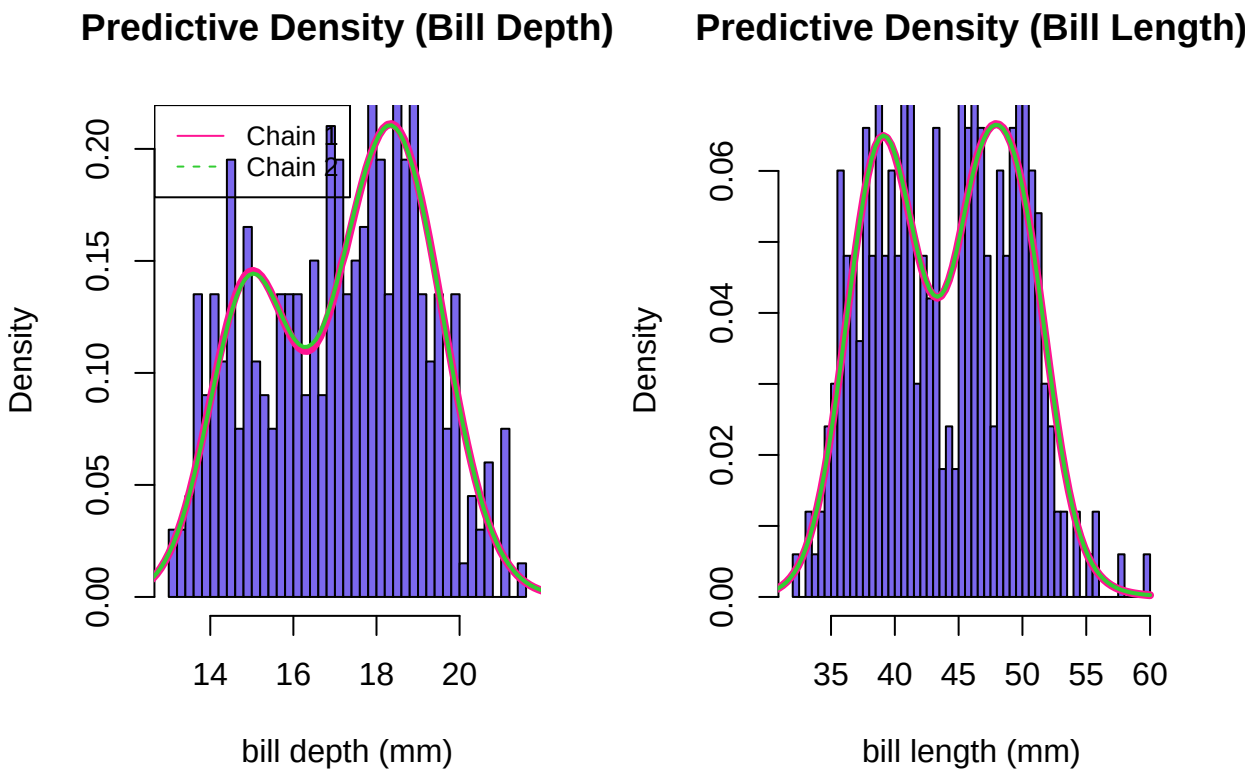
```
## [1] "Posterior distribution summary for bill depth:"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.48   15.83   17.22   17.20   18.54   23.89
```

```
## [1] "Posterior distribution summary for bill length:"
```

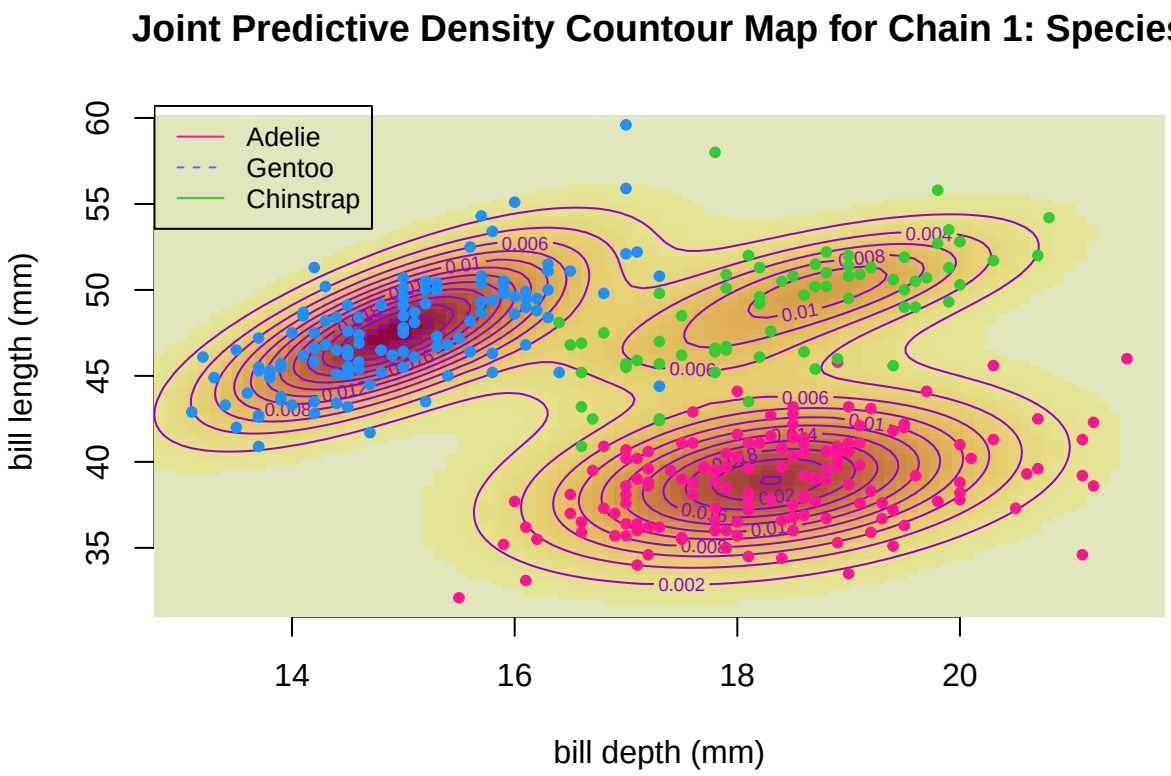
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.80   40.42   44.04   44.06   47.72   65.03
```

We now plot the 2 MCMC chains for predictive density distribution against histograms of the data to visualise the outcome of the mixture modelling. In most of the simulations run, the mixture analysis for both chains has identified 2 normal distributions per data axis (bill depth and bill length), however the fit of each chain may differ each time the simulation is run.

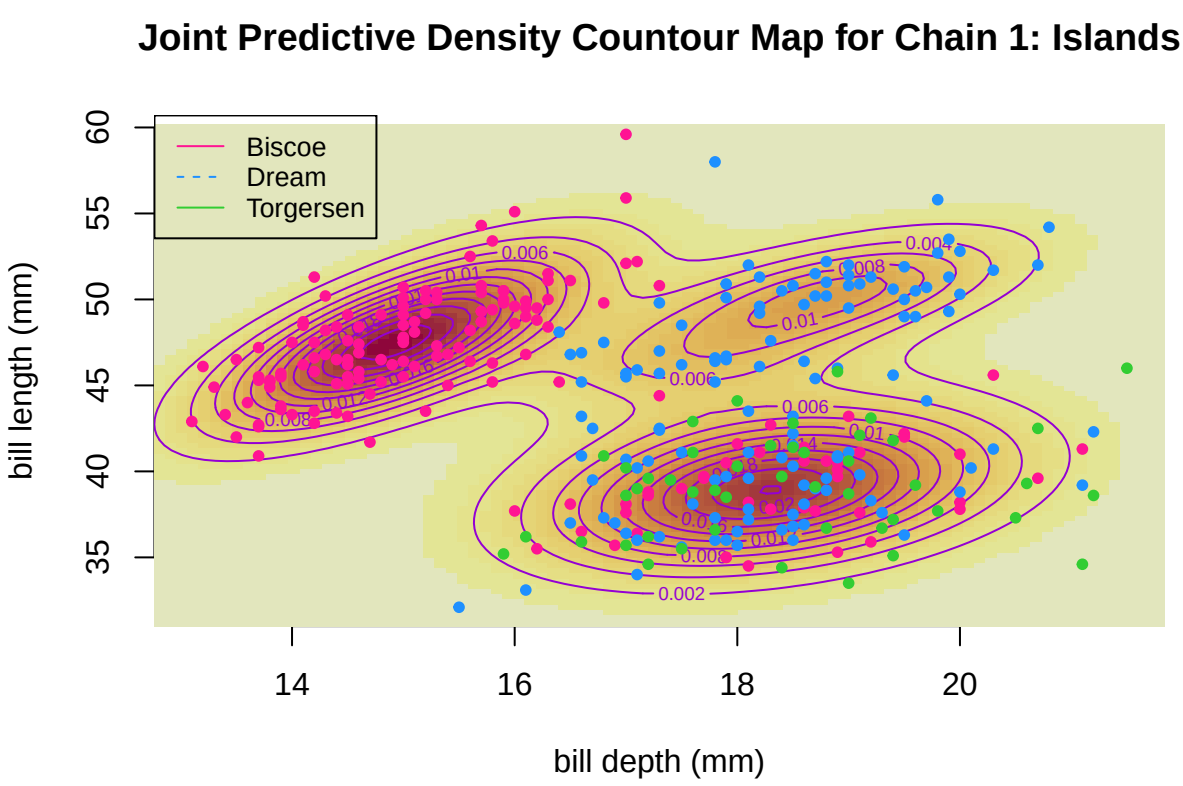


We observe the fitting of both chains to the data - for both plots, the mixture modelling has identified a mixture of 2 normal distributions. But how does this relate back to the original data? We want to observe how each data point is identified according to the cluster analysis and what that tells us about the original

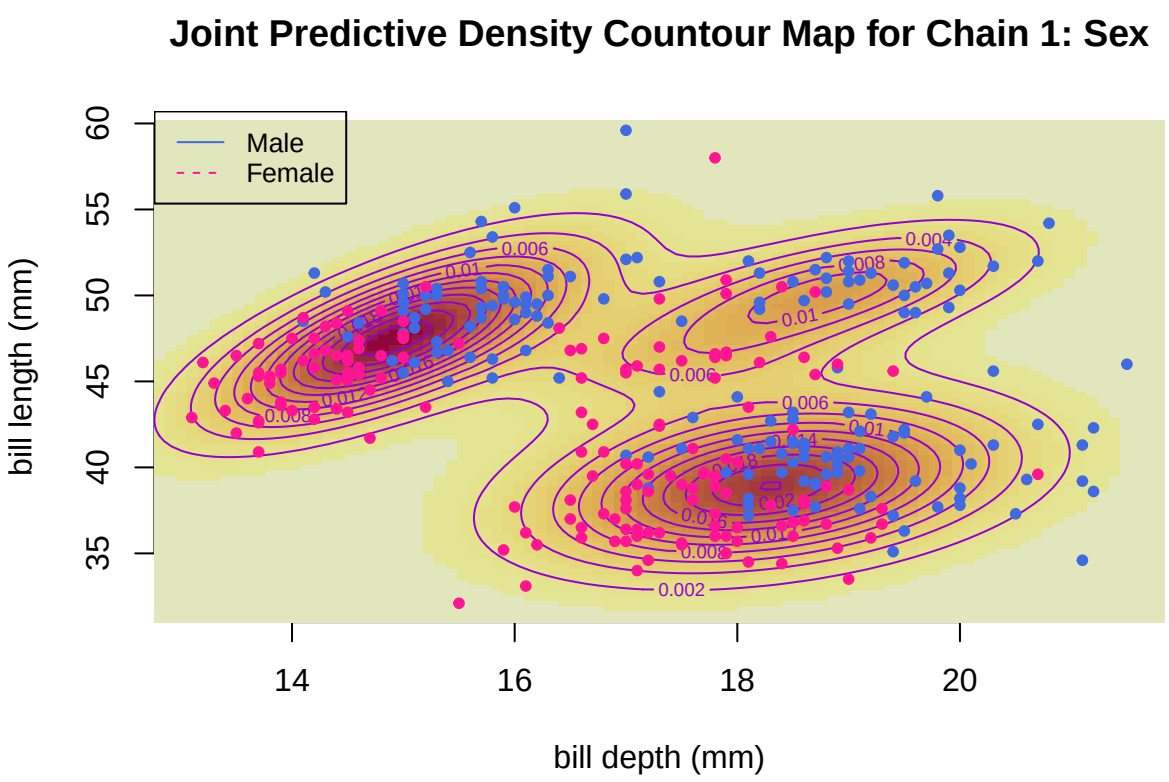
data. Below, we plot a busy figure, where several components are to be noted. First, the background of the plot is a heat map, showing the density concentrations of data points in the data set. This is overlaid by a contour plot. We see from the heat map that there are 3 darker regions, indicating a high concentration of data points in these regions. This observation is supplemented from the contour map, which shows dense contours on top of the dark regions. Now, we overlay the data points according to three differentiating features in the dataset: 1. penguin species, 2. islands on which penguins were observed, and 3. sex of the penguins. We now compare the combined heat map/contour map with the aforementioned three possible sub-populations which have been identified, and how well of a fit they provide.



For the above plot which maps the data points according to the species of the penguins, we see that the data points correspond well to the number of mixtures identified in the dataset.



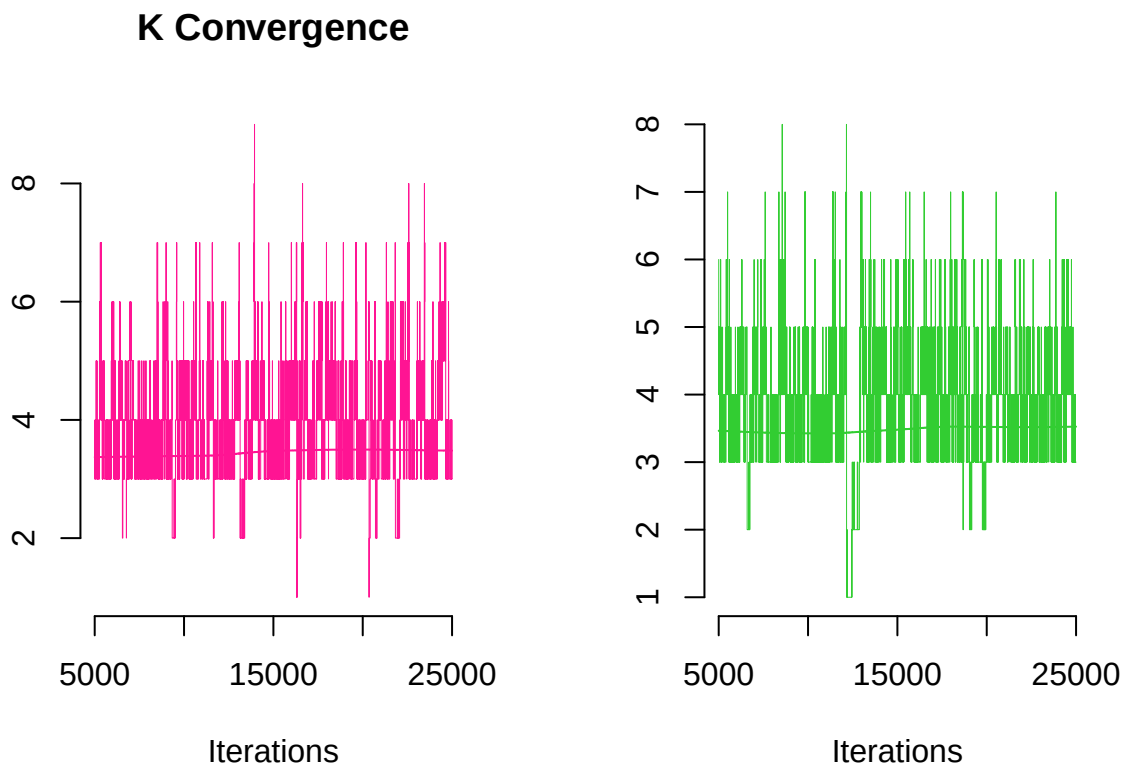
For the above plot where the data points have been plotted according to the island each penguin was observed on, we see that the data points do not correspond very well to the clusters identified within the dataset.



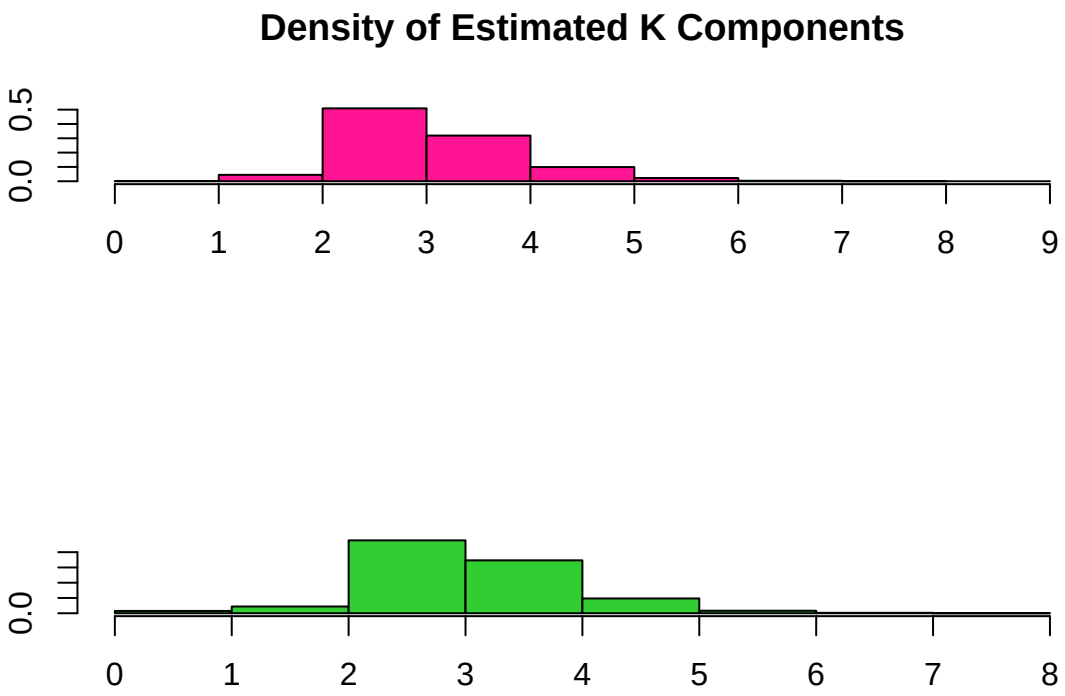
For the above plot, the data points were plotted according to the sec of each penguins, and again we see that the data points do not match the clusters at all.

Again, in most of the simulations run, 3 sub-populations within the bill depth/length data have been identified. We see that each of these clusters correspond to certain penguin characteristics. The cluster in the bottom-right shows that the associated penguin sub-population have short but deep bills, the cluster in the top-right also have deep bills like the bottom-right cluster, but longer bills instead of shorter bills. The top-left cluster have shallow and long bills. But the question is what sub-populations did the algorithm actually identify, and what do these correspond to? We find that out of the three possible sub-population types shown above, it is most likely that the data points are clustered according to their species, rather than island or sex. This output of the the algorithm can be verified by doing further analysis with the mixture weighting on each point, however this extra analysis has not been included in this report due to computation constraints.

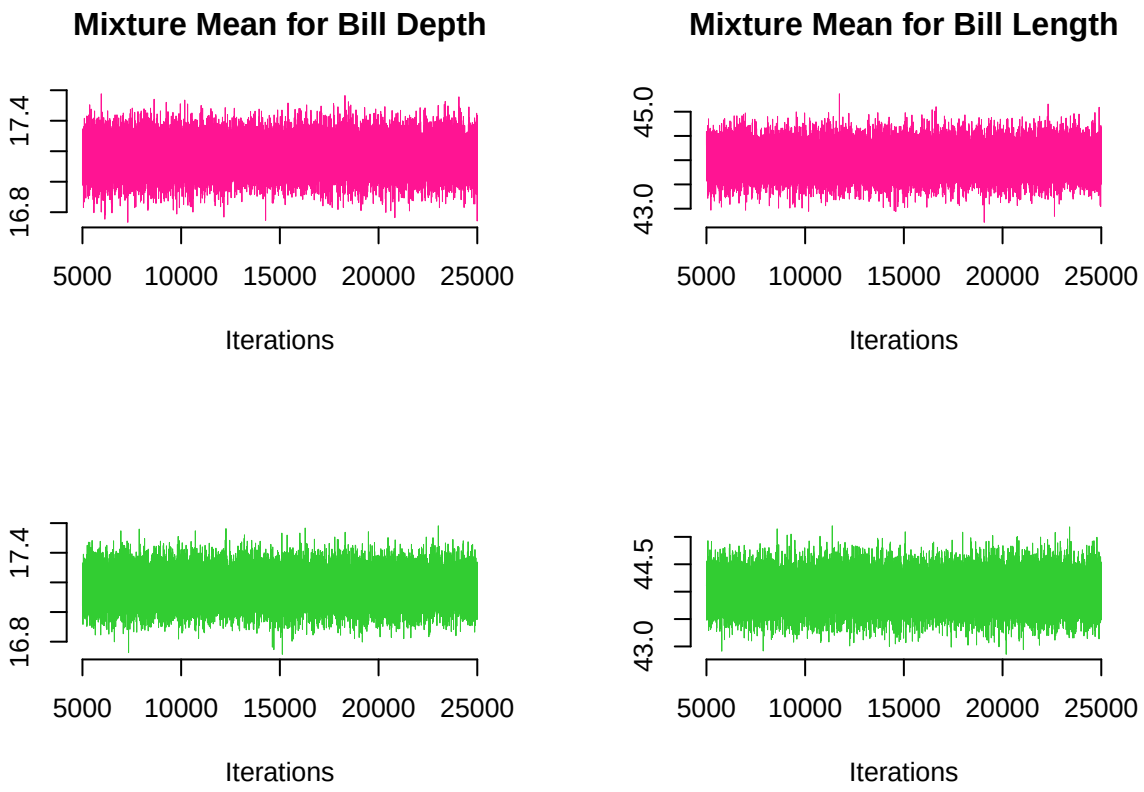
Now we wish to check convergence of the reverse jump MCMC output. For this, let's first check the convergence for the number of clusters K identified.



We see that the value of K modelled in each iteration of the reverse jump MCMC algorithm tends to be a discrete value, but converges to a certain value by the end of iterations for each chain. Let's compare this convergence to a density plot of the K value for each chain. Note that the pink colour refers to Chain 1, and the green colour to Chain 2.



We again find that the reverse jump MCMC identifies values of K with certain densities. In most simulations, the K value with highest density has been K=3. Now we will check the convergence for bill depth and bill length for each chain.



We see that the model converges smoothly to a certain value for the mean of each axis for each chain. These converged mean values correpond to the posterior mean identified earlier.

Hence, by analysing a simple dataset of penguin bill depths and lengths using mixture analysis, we have demonstrated the usage of mixture modelling for identifying sub-populations within a dataset with no supplemental information, and being able to make inferences on the sub-populations and overall data. More statistics and analysis are included in the Appendix.

4. Conclusion

In this report, we introduced the concept of mixture analysis, and how it differs from its Bayesian implementation. We considered the finite mixture problem with unknown mixture components, which uses the reverse jump MCMC methodology to simulate the posterior distribution of the parameter space using a reversibility constraint when jumping between models (hence the name). We then applied a finite mixture analysis with unknown components on the 'palmerpenguins' dataset using the R package 'mixAK', for the bill depths and bill lengths of the penguins. From this analysis, we found that the algorithm identified (with a higher probability) 3 clusters in the data, and it attributed each data point to a cluster according to penguin species. From this analysis, we learn that by considering the dataset of bill depths and lengths of the penguins, there are 3 sub-clusters, or sub-populations, which each correspond to one of 3 species, i.e. that short bill depths and long bill lengths correspond to Gentoo penguins, longer bill depths and short bill lengths correspond to Adelie penguins, and long bill depths and lengths correspond to Chinstrap penguins. This was a very simple toy example to demonstrate the capabilities of Bayesian mixture analysis, however its applications in the real world are very useful.

Not only can Bayesian mixture modelling be applied to a dataset with a finite number of mixtures, but also an infinite number of mixtures. In addition, this analysis can be applied to known or unknown number of mixtures, univariate or multivariate data, and it can use a weakly-informed or strongly-informed prior

(though a weak prior will not give accurate results), and can use different types of priors, e.g. 'mixAK' accepts fixed, uniform, or Poisson priors. An interesting problem with mixture models is the label-switching problem, which refers to the invariance of the likelihood under relabeling of the mixture components. This causes a very symmetric posterior distribution, which makes identification difficult. This effect becomes more prominent in multidimensional datasets, which means that estimating the parameters of the distribution becomes difficult.

Nevertheless, we have learned that the method of mixture modelling and cluster analysis is a very useful tool to analyse a variety of datasets and gain inference about sub-populations within the data, efficiently compute Bayes estimators for mixtures of these distributions, and estimate posterior distributions.

References

- Komarek, A., & Komarkova, L. (2014). Capabilities of R Package mixAK for Clustering Based on Multivariate Continuous and Discrete Longitudinal Data. Journal of Statistical Software, 59(12).
- Richardson, S., & Green, P. J. (1997). On Bayesian Analysis of Mixtures with an Unknown Number of Components. J. R. Statist. Soc. B, 59(4).
- Komárek, A. (2020). mixAK (5.3) [Multivariate Normal Mixture Models and Mixtures of Generalized Linear Mixed Models Including Model Based Clustering]. CRAN. https://cran.r-project.org/web/packages/mixAK/index.html
- Benaglia T, Chauveau D, Hunter DR, & Young D (2009). "mixtools: An R Package for Analyzing Finite Mixture Models." Journal of Statistical Software, 32(6), 1–29. http://www.jstatsoft.org/v32/i06/.
- Pelleg, D. & Moore, A. 1999. Accelerating exact k-means algorithms with geometric reasoning. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '99). Association for Computing Machinery, New York, NY, USA, 277–281. DOI:https://doi.org/10.1145/312129.312248
- Marin, J., Mengersen, K., & Robert, C. (2005). Bayesian Modelling and Inference on Mixtures of Distributions. Dey, D & Rao, C (Eds.) Handbook of Statistics. Elsevier, Netherlands, pp. 459-507.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., et al. (2021). Bayesian Data Analysis. Third edition (with errors fixed as of 6 April 2021).

Appendix

Below is the full output summary for RJModel2, the output of the mixture estimation NMixMCMC.

```
##
##      Normal mixture with at most 10 components estimated using RJ-MCMC
##      ==================================================================
## Posterior distribution of K:
## ----------------------------
##               1       2       3      4       5       6       7       8     9
## Chain 1 0.00155 0.04475 0.50905 0.3190 0.09910 0.02275 0.00340 0.00035 5e-05
## Chain 2 0.01505 0.04420 0.47755 0.3467 0.09705 0.01645 0.00285 0.00015 0e+00
##
## Posterior summary statistics for moments of mixture for original data:
## -----------------------------------------------------------------------
## Means (chain 1):
##            y.Mean.1   y.Mean.2
## Mean     17.1658769 44.0093718
## Std.Dev.  0.1089305  0.3015827
## Min.     16.7338174 42.7188863
## 2.5%     16.9518790 43.4223652
## 1st Qu.  17.0927529 43.8051288
## Median   17.1662878 44.0090951
## 3rd Qu.  17.2400405 44.2113653
```

```
## 97.5%    17.3792384 44.6028090
## Max.     17.5766132 45.3688942
##
## Means (chain 2):
##            y.Mean.1   y.Mean.2
## Mean     17.1667139 44.0097715
## Std.Dev.  0.1077331  0.3013937
## Min.     16.7147605 42.8567001
## 2.5%     16.9561153 43.4169041
## 1st Qu.  17.0940235 43.8062885
## Median   17.1666806 44.0110217
## 3rd Qu.  17.2399145 44.2127541
## 97.5%    17.3777850 44.5986473
## Max.     17.5818959 45.2018830
##
## Standard deviations and correlations (chain 1):
##            y.SD.1  y.Corr.2.1     y.SD.2
## Mean     1.977201 -0.22144433  5.4949340
## Std.Dev. 0.064429  0.05149429  0.2081760
## Min.     0.344340 -0.85541070  0.5845959
## 2.5%     1.861953 -0.31506180  5.1818592
## 1st Qu.  1.936194 -0.25649576  5.3769119
## Median   1.975613 -0.22264884  5.4834591
## 3rd Qu.  2.014895 -0.18896818  5.5969542
## 97.5%    2.099588 -0.11851619  5.8458359
## Max.     2.954479  0.20852836 15.3074835
##
## Standard deviations and correlations (chain 2):
##              y.SD.1  y.Corr.2.1     y.SD.2
## Mean     1.97637754 -0.22171122  5.4938285
## Std.Dev. 0.06519248  0.05069283  0.1845442
## Min.     0.97204466 -0.62506735  0.6664835
## 2.5%     1.86040813 -0.31659546  5.1821114
## 1st Qu.  1.93521290 -0.25545815  5.3799113
## Median   1.97410951 -0.22257496  5.4847109
## 3rd Qu.  2.01510128 -0.18912280  5.5957638
## 97.5%    2.10174932 -0.12135131  5.8378001
## Max.     3.64134701  0.32253691 10.0973627
```

Now we check the actual means of the bill depth and length data against the estimated values for each chain, along with the standard deviations.

```
mean(penguins.clean$bill_depth_mm) # actual mean of bill depths
```

```
## [1] 17.16486
```

```
mean(RJModel2[[1]]$mixture$y.Mean.1) # calculated bill depth mean for Chain 1
```

```
## [1] 17.16588
```

```
mean(RJModel2[[2]]$mixture$y.Mean.1) # calculated bill depth mean for Chain 2
```

```
## [1] 17.16671
```

```
mean(RJModel2[[1]]$mixture$y.SD.1) # calculated bill depth standard deviation for Chain 1
```

```
## [1] 1.977201
```

```
mean(RJModel2[[2]]$mixture$y.SD.1) # calculated bill depth standard deviation for Chain 2
```

```
## [1] 1.976378
```

```
mean(penguins.clean$bill_length_mm) # actual mean of bill lengths
```

```
## [1] 43.99279
```

```
mean(RJModel2[[1]]$mixture$y.Mean.2) # calculated bill length mean for Chain 1
```

```
## [1] 44.00937
```

```
mean(RJModel2[[2]]$mixture$y.Mean.2) # calculated bill length mean for Chain 2
```

```
## [1] 44.00977
```

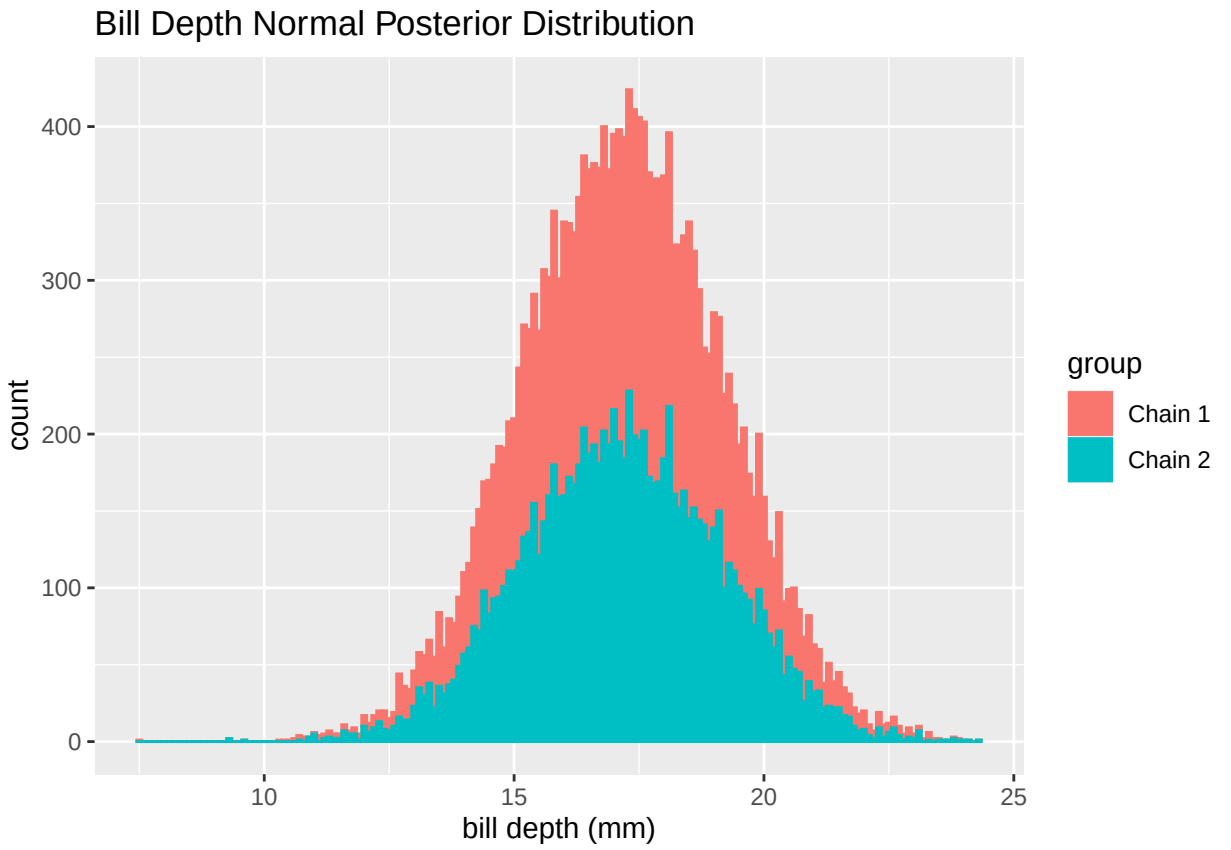
```
mean(RJModel2[[1]]$mixture$y.SD.2) # calculated bill length standard deviation for Chain 1
```

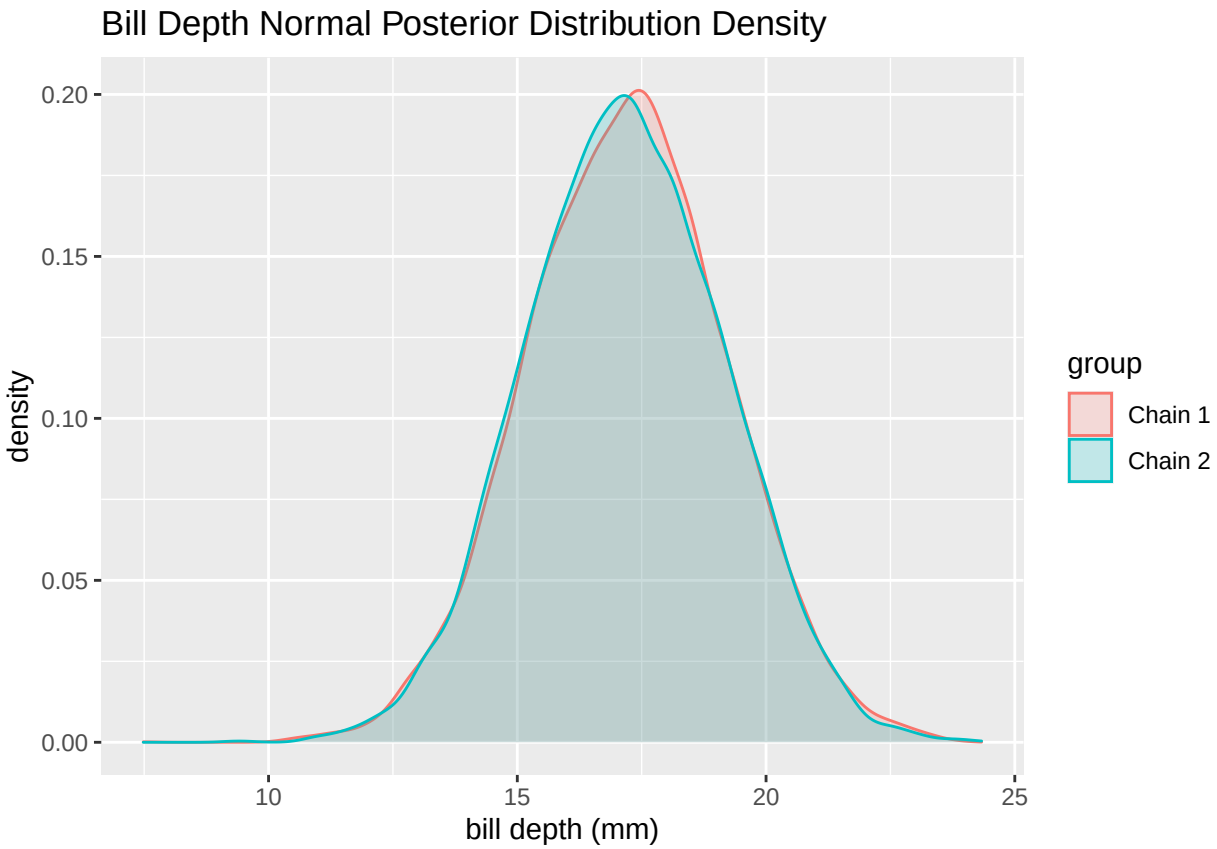
```
## [1] 5.494934
```

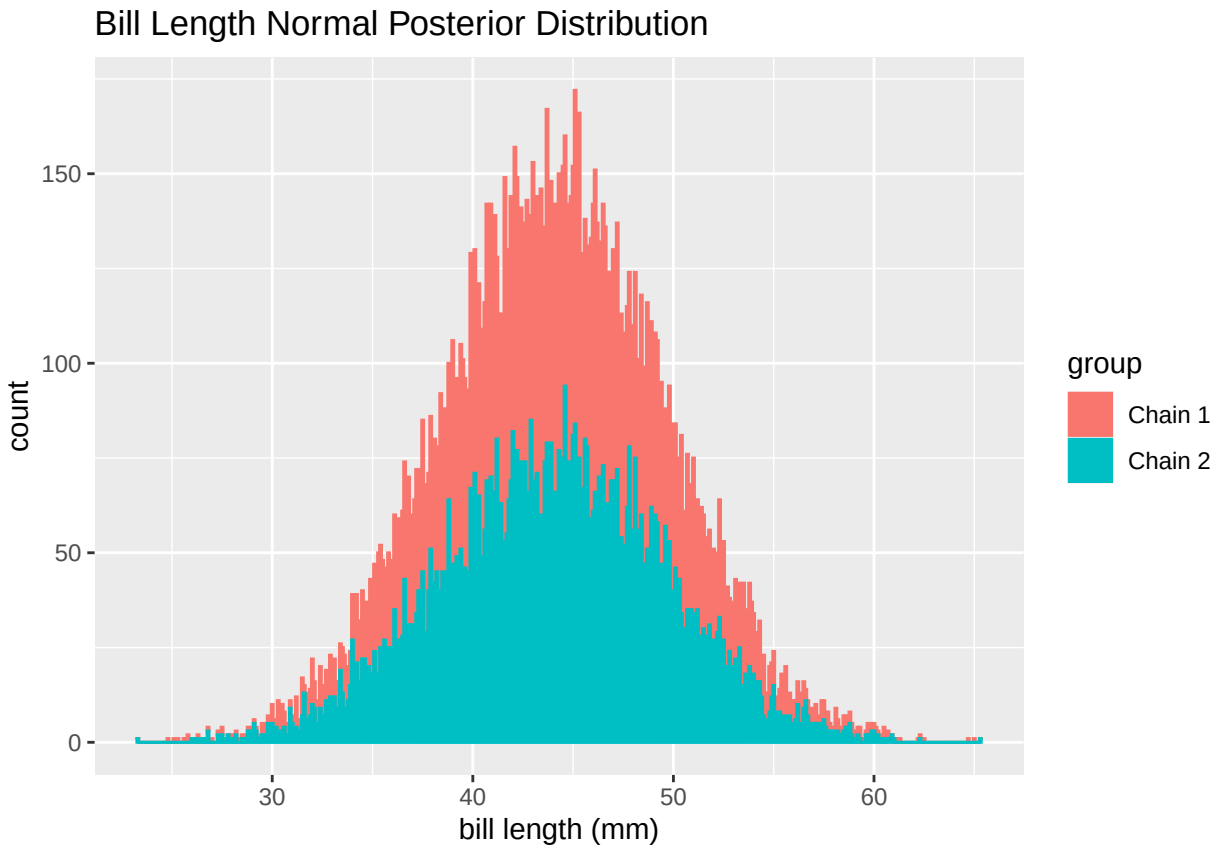
```
mean(RJModel2[[2]]$mixture$y.SD.2) # calculated bill length standard deviation for Chain 2
```

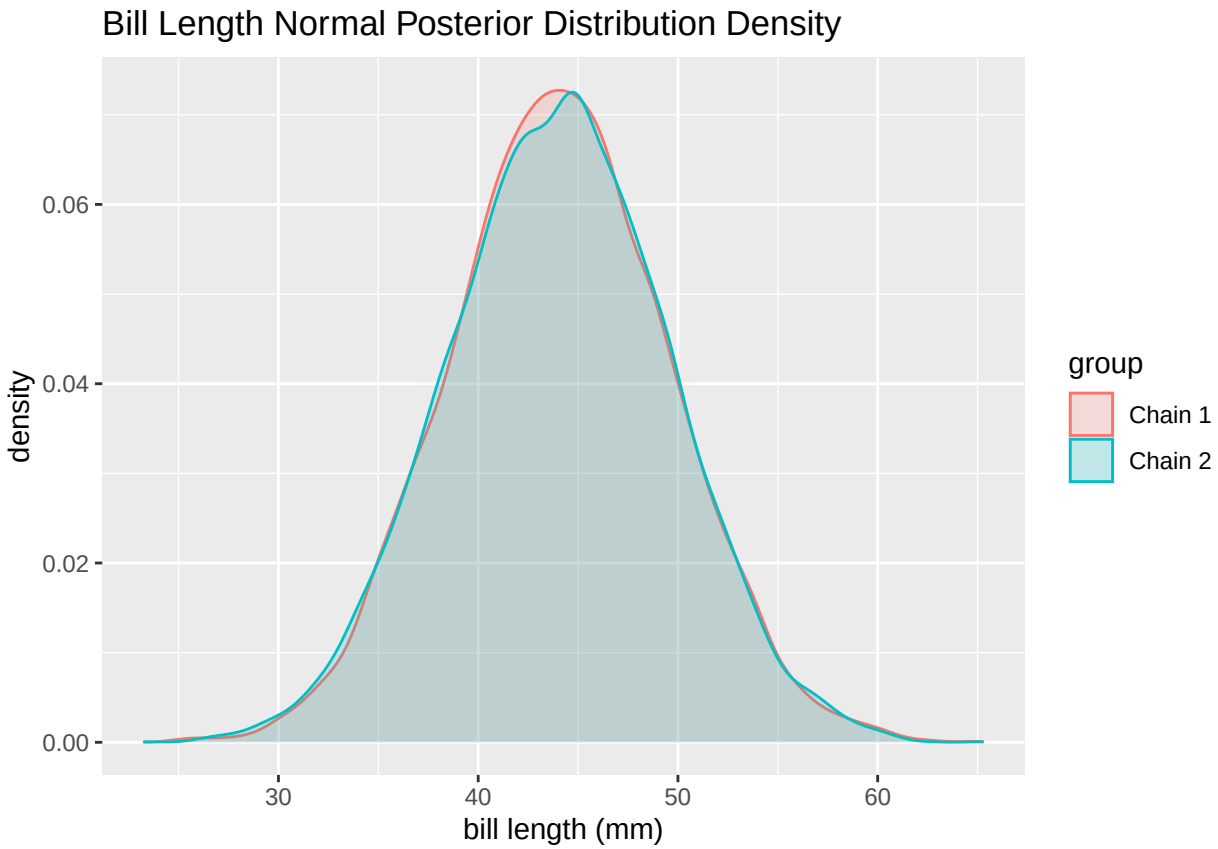
```
## [1] 5.493829
```

The posterior distributions and densities for each chain derived from the output of running the model NMixMCMC are below.




Bill Depth Normal Posterior Distribution Density
density
0.20
0.15
0.10
0.05
0.00
10
15
20
25
bill depth (mm)
group
Chain 1
Chain 2


Bill Length Normal Posterior Distribution
count
150
100
50
0
30
40
50
60
bill length (mm)
group
Chain 1
Chain 2


Bill Length Normal Posterior Distribution Density
density
0.06
0.04
0.02
0.00
30
40
50
60
bill length (mm)
group
Chain 1
Chain 2