

**MODELLING FOR HOUSING PRICES:  
BAYESIAN APPROACH WITH MIXTURE MODELS**

PROJECT REPORT SUBMITTED

by

TAMOGHNA DEY

19MSM31

Under the guidance of

MADHUCHHANDA BHATTACHARJEE  
PROFESSOR

The project submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN STATISTICS  
(2021)

At



UNIVERSITY OF HYDERABAD

SCHOOL OF MATHEMATICS AND STATISTICS

UNIVERSITY OF HYDERABAD

PROF. CR RAO ROAD, GACHIBOWLI, HYDERABAD

500046

## Certificate

---

This is to certify that the project work entitled "**Modelling for Housing Prices: Bayesian Approach using Mixture Models**" submitted by Tamoghna Dey (**19MSMS31**) is a bonafide project work done under my supervision. It is being submitted in partial fulfillment of the requirements of Master of Science in Statistics degree of University of Hyderabad.



Dr. Madhuchhanda Bhattacharjee

Professor

School of Mathematics and Statistics

University of Hyderabad, Hyderabad

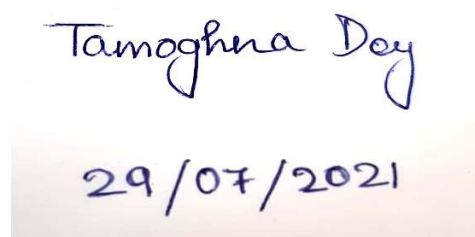
Date : 28/07/2021

Place : Hyderabad

## Declaration

---

I, Tamoghna Dey, hereby declare that the project work entitled “**Modelling for Housing Prices: Bayesian Approach with Mixture Models**” is an original record of studied and bonafide work carried out by me under the guidance of Dr Madhuchhanda Bhattacharjee, Professor, School of Mathematics and Statistics, University of Hyderabad, Hyderabad and has not been submitted by me elsewhere for the award of any degree, diploma, title or recognition before.

A photograph of a handwritten signature in blue ink that reads "Tamoghna Dey" and the date "29/07/2021" written below it.

Tamoghna Dey  
19MSMS31  
School of Mathematics and Statistics,  
University of Hyderabad,  
Hyderabad

# Acknowledgments

---

First and foremost, I would like to express my deep and sincere gratitude to my supervisor Dr. Madhuchhanda Bhattacharjee, Professor, School of Mathematics and Statistics, University of Hyderabad, Hyderabad for giving this opportunity and providing guidance and continuous support throughout this project work. Her vision and motivation have immensely inspired and encouraged me.

I would also like to thank my classmates, especially Mr. Suman Saha, Mr. Sagarnil Bose and Mr. Soham Mukherjee for their encouragement and support. Lastly I would like to thank my parents for their continuous support.

## Abstract

---

Housing Price data often contains values varying over a wide range. Depending on the neighborhood, number of bedrooms, etc. various attributes, housing prices tend to cluster together or vary among themselves. Questions that arises naturally is whether sub-populations may be hidden in housing price data, and if so, will a single distribution be enough to approximate it?

That is where Bayesian Mixture Models come in. Bayesian Mixture Models are a popular data analysis technique used to identify underlying classes in a data. It employs high-powered computational techniques such as Markov chain Monte Carlo (MCMC) to simulate posterior data points. In this project we use Finite Normal Mixture Models on the Boston housing dataset, a commonly used standard housing data to look for an appropriate model for median housing values.

# Contents

---

1. Introduction.....	1
2. About the Data.....	2
3. Models and Methods.....	4
3.1. Abstract.....	4
3.2. Choice of Prior Distributions .....	6
3.3. Hyperparameters for Prior Distributions.....	6
3.4. Parameter Estimation using Gibbs Sampler .....	7
3.5. Gibbs Sampling Algorithm.....	8
4. Data Analysis and Results.....	8
4.1. Method of Analysis.....	8
4.2. Two component Normal Mixture Model.....	9
4.3. Three component Normal Mixture Model.....	12
4.4. Four component Normal Mixture Model.....	16
4.5. Model Diagnostics, Results & Conclusion.....	20
5. Bibliography.....	24
6. Appendix (R Codes) .....	25

# MODELLING FOR HOUSING PRICES: BAYESIAN APPROACH WITH MIXTURE MODELS

---

## 1. INTRODUCTION

Mixture Modelling is a widely used modelling technique. It is primarily used when there are subpopulations present in the data and a single distribution is not enough to specify the parent population appropriately.

Karl Pearson (1894) was the first to explore this idea when he fit a mixture of two normal distributions to the forehead to body length ratio measurements of female crabs residing in the Bay of Naples. However, the method of moments that he proposed involved computing a 9<sup>th</sup> degree polynomial which at the time would've been very problematic. It was not until 1977 that mixture models came to be used frequently, when Dempster et al. published their seminal paper using the EM algorithm and illustrated its uses to simplify the application of maximum likelihood techniques to incomplete data settings. Bayesian mixture models began appearing in the literature in the early 1990s, with West (1992) and Diebolt and Robert (1994) being among the first published efforts which incorporated the missing data structure of mixture modelling with the use of Bayesian techniques. Bayesian mixture modelling is now routinely used in data analysis due to the natural setting for hierarchical models within the Bayesian framework, the availability of high-powered computing and developments in posterior simulation techniques, such as Markov chain Monte Carlo (MCMC).

## 2. ABOUT THE DATA

Here the Boston dataset from the MASS package in R has been used. This dataset contains information collected by the U.S Census Service concerning Housing in the area of Boston Mass. It is used by a lot of people for practical experience in getting exposure to real world data by building statistical models.

The Boston dataset contains 506 observations.

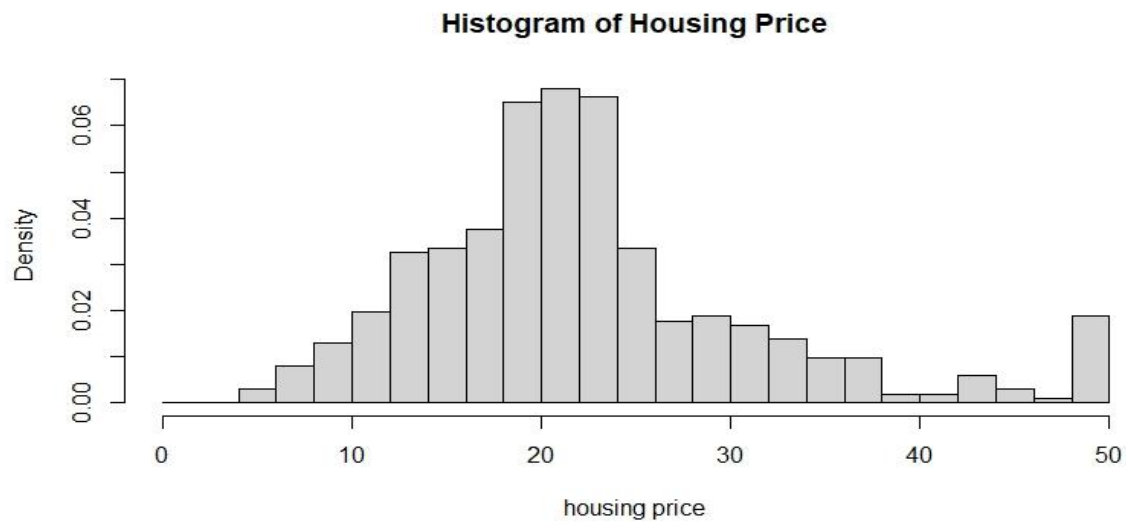
There are **14** variables in each observation of the dataset. They are:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 square ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per \$10,000
11. PTRATIO - pupil-teacher ratio by town
12. BLACK -  $1000(B_k - 0.63)^2$  where  $B_k$  is the proportion of blacks by town
13. LSTAT - % lower status of the population
14. MEDV - Median value of owner-occupied homes in \$1000's

The variable of interest here is median value of the houses (MEDV). The variable ranges from 5 to 50. It has a mean of 22.53 and a median of 21.2.

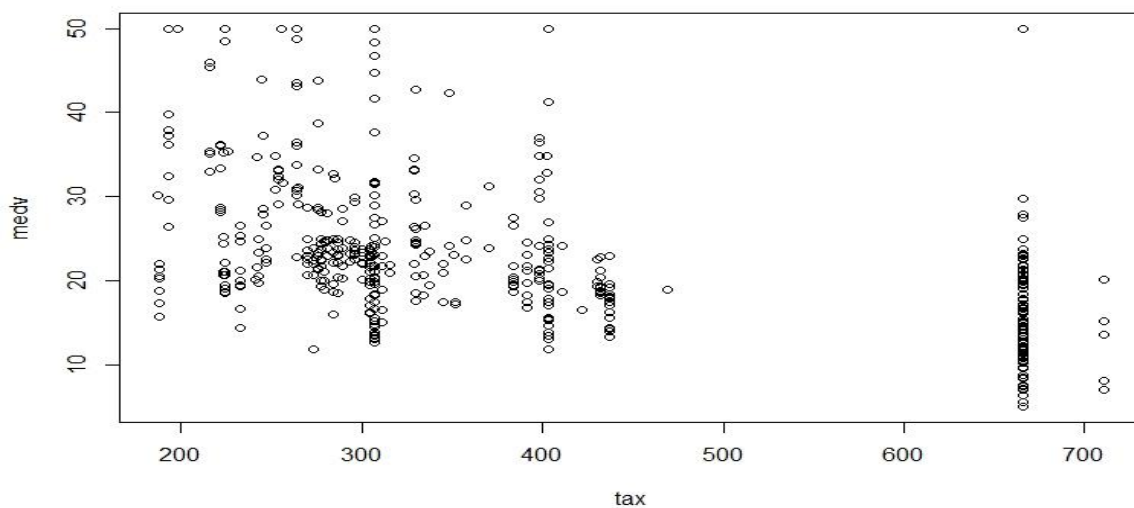
Our goal here is to fit an appropriate finite mixture of normal distributions to this variable. Looking at the histogram of medv and examining scatterplots of medv against the other variables for clusters may give us an idea about the number of components, say,  $k$  in the mixture. Let us denote medv by  $y$  for the rest.

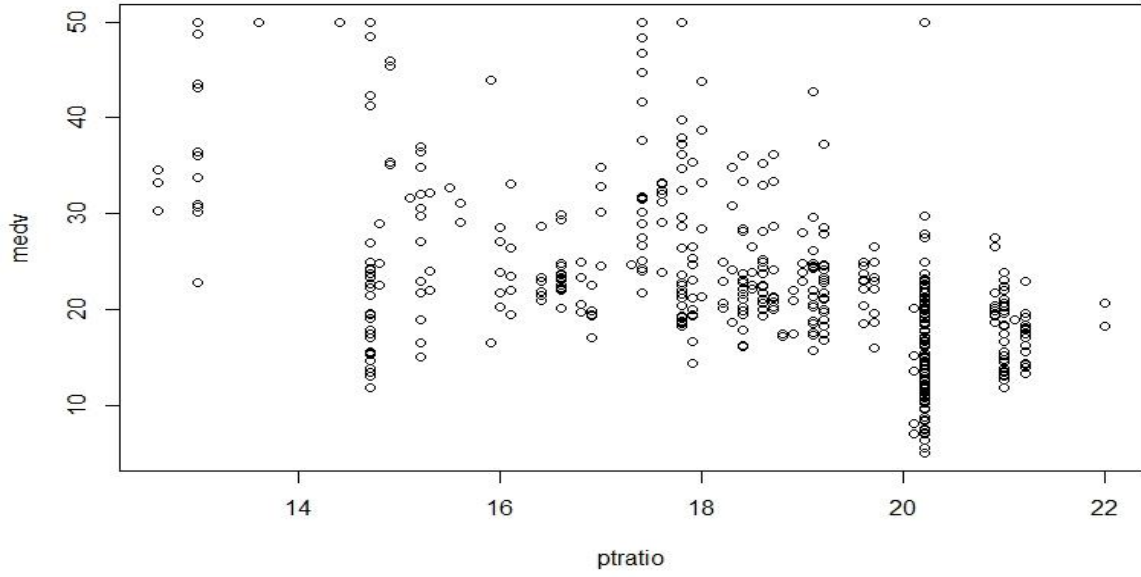




The histogram suggests that a single distribution will probably not be a good fit for the variable. Commonly used distributions that are frequently used to approximate data do not seem applicable here as sharp changes between density of adjacent intervals are visible throughout the histogram.

Scatterplots against CHAS, TAX and PTRATIO also reveal clustering which may have contributed to multiple distributions being in the distribution of MEDV.





Scatterplots of median housing values against tax and pupil-teacher ratio shows the presence of at least 4 and 9 clusters, respectively. As we are considering normal mixture models, based on these scatterplots we may hypothesize that some of these variables play an important part in determining the number of subpopulations hidden in the normal mixture model.

### 3. MODELS AND METHODS

#### 3.1. ABSTRACT

The normal mixture model can be represented by the likelihood

$$g(y) = \prod_{i=1}^N \sum_{j=1}^k \lambda_j \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left[ -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right]$$

where  $N$  is the sample size,  $k$  is the number of components in the mixture,  $\lambda_j$  represent the weight of component  $j$ ,  $\mu_j$  and  $\sigma_j^2$  are the mean and variance of the component  $j$ .

In Bayesian analysis, we estimate the posterior distribution of the unknown parameters  $(\mu_j, \sigma_j^2, \lambda_j)$  by allocating them a prior distribution and applying Bayes' rule.

$$p(\mu, \sigma^2, \lambda | y) \propto \prod_{i=1}^N \sum_{j=1}^k \lambda_j \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left[ -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right] p(\mu, \sigma^2, \lambda)$$

where  $p(\mu, \sigma^2, \lambda)$  represents the joint prior distribution.

In Bayesian setup, we usually create the mixture model within a hierarchical framework. This is done by introducing an  $1 \times k$  indicator variable  $z_i$  which represents the unobserved component membership of the  $i$ -th observation. It is of the form  $z_i = (z_{i1}, z_{i2}, \dots, z_{ik})$  where

$$\begin{aligned} z_{ij} &= 1, \text{ when } y_i \text{ belongs to the } j^{\text{th}} \text{ component of the mixture,} \\ &= 0, \text{ otherwise, } j = 1, 2, \dots, k \end{aligned}$$

Given the weights  $\lambda$ ,  $z_i$  follows a multinomial distribution.

The joint conditional distribution of the observations  $(y_i)$  and the unobserved component membership  $(z_i)$  is given by

$$p(y, z | \mu, \sigma^2, \lambda) \propto p(z | \lambda) \cdot p(y | \mu, \sigma^2, z) \propto \prod_{i=1}^N \prod_{j=1}^k \left\{ \lambda_j (2\pi\sigma_j^2)^{-\frac{1}{2}} \exp \left[ -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right] \right\}^{z_{ij}}$$

Our goal is to estimate the parameters  $\mu, \sigma^2, \lambda$  for each of the  $k$  components. For mixture distributions, we can assume independence between the weights  $(\lambda)$  and the component parameters  $(\mu, \sigma^2)$ . The joint posterior distribution of the unknown parameters is given by

$$\begin{aligned} p(\mu, \sigma^2, \lambda | y, z) &\propto p(z | \lambda) p(y | \mu, \sigma^2, z) p(\mu, \sigma^2, \lambda) \\ &\propto p(z | \lambda) p(\lambda) p(y | \mu, \sigma^2, z) p(\mu | \sigma^2) p(\sigma^2). \end{aligned}$$

Drawing samples from the posterior distribution through the Gibbs sampler is a popular way of estimating the parameters.

### 3.2. CHOICE OF PRIOR DISTRIBUTIONS

We need to choose suitable prior distributions in order to analyse the mixture model properly. In this analysis we use conjugate priors, i.e., priors which follow the same parametric form as their corresponding posterior distributions. These priors are both computationally convenient and suitable for the data:

$$\lambda \sim \text{Dirichlet}(\alpha_1, \alpha_2, \dots, \alpha_k)$$

$$\mu_j | \sigma_j^2 \sim N(\xi_j, \sigma_j^2 / m_j)$$

$$\sigma_j^2 \sim \text{Inverse Gamma}(v_j/2, s_j^2/2)$$

### 3.3. HYPERPARAMETERS FOR PRIOR DISTRIBUTION

To start the modelling process, values need to be assigned to the hyperparameters of the priors. As we have very little prior information on the parameters or their expected behaviour, the values for hyperparameters should be chosen to reflect this uncertainty. It is desirable to nominate priors that can encapsulate all likely values of the unknown parameters.

If we take unreasonably vague priors, they would not contribute much to the modelling rather make the technique perform well below its capabilities.

In this analysis all the  $\alpha$  values are given the value 1, which reflects our prior belief that the data could be equally divided into  $k$  components. For a Dirichlet distribution the mean value of each of the  $k$  proportions is  $1/k$ , and the distribution of each of them is contained in the interval  $[0,1]$ . By allocating 1 for each  $\alpha$ , we are ensuring that the estimates of  $\alpha$  is largely influenced by the data, as values of  $n_k$  will eventually dominate the simulations.

Here we will take the overall mean of  $y$  as  $\xi$  and take the  $m$  values as  $m = 2.6 / (y_{\max} - y_{\min})^2$  as hyperparameters of  $\mu$ . The choice of the mean is self-explanatory. We take  $m$  to be this small value, as  $m$  being small allows the variance of the mean to be large which increases its chances of encompassing all likely values.

We take  $v_j = 2.56$ ,  $s_j^2 = 0.72 s_y^2$ , as hyperparameters of  $\sigma^2$ . This choice indicates our limited knowledge about the component variances and our dependence on the data.

The priors that we are using were used by Raftery (1996).

An alternative choice of priors is one used by Bensmail et al. (1997) where  $\lambda = \text{mean}(y)$ ,  $m=1$ ,  $v_j = 5$ ,  $s_j^2 = s_y^2$ .

### 3.4. PARAMETER ESTIMATION USING GIBBS SAMPLER

Gibbs sampler is an algorithm that generates a sequence of sample from a joint distribution of two or more random variables. It is one of the most popular methods of summarizing complex posterior distributions.

Let us first consider a bivariate random variable  $(x,y)$  and we wish to compute one or both the marginals  $p(x)$  and  $p(y)$ . The idea behind the Gibbs sampler is that it is much easier to obtain a sequence of conditional distributions  $p(x_j|y)$  and  $p(y_j|x)$  than it is to obtain marginals by integrating the joint distribution.

In this algorithm we start with an initial value of  $y=y_0$ , and generate  $x_0$  from the conditional distribution  $p(x|y=y_0)$ . Then we generate  $y_1$  based on the value of  $x_0$  from the conditional distribution  $p(y|x=x_0)$ .

$$x_i \sim p(x|y=y_i)$$

$$y_i \sim p(y|x=x_{i-1})$$

Repeating this step  $k$  times, we obtain a Gibbs sequence of length  $k$ . A subset of this sequence  $(x,y)$  can be taken as the simulated draws from the joint distribution. After a sufficient burn in period to remove effects of initial sampling values, one can sample points from the chain for estimation and inference. The Gibbs sequence converges to a stationary distribution that is independent of starting values, which is the target distribution we are trying to produce.

In case more than two variables are involved, the process is modified. Suppose for the random variables  $\theta_1, \theta_2, \dots, \theta_p$  with initial values  $\theta_i = \theta_i^0$  the Gibbs sampler simulates updated values in this way:

$$\theta_1^t \sim p(\theta_1 | y, \theta_2^{t-1}, \dots, \theta_p^{t-1})$$

$$\theta_2^t \sim p(\theta_2 | y, \theta_1^t, \theta_3^{t-1}, \dots, \theta_p^{t-1})$$

.

.

$$\theta_p^t \sim p(\theta_p | y, \theta_1^t, \theta_2^t, \dots, \theta_{p-1}^t)$$

The process is iterated until satisfactory convergence is obtained.

### 3.5. GIBBS SAMPLING ALGORITHM

1. Update  $z_i \sim \text{Multinomial}(1; \omega_{i1}, \omega_{i2}, \dots, \omega_{ik})$ ,

$$\omega_{ij} = \frac{\lambda_j \left( \sqrt{2\pi\sigma_j^2} \right)^{-1} \exp \left[ -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right]}{\sum_{t=1}^k \lambda_t \left( \sqrt{2\pi\sigma_t^2} \right)^{-1} \exp \left[ -\frac{1}{2} \left( \frac{y_i - \mu_t}{\sigma_t} \right)^2 \right]}$$

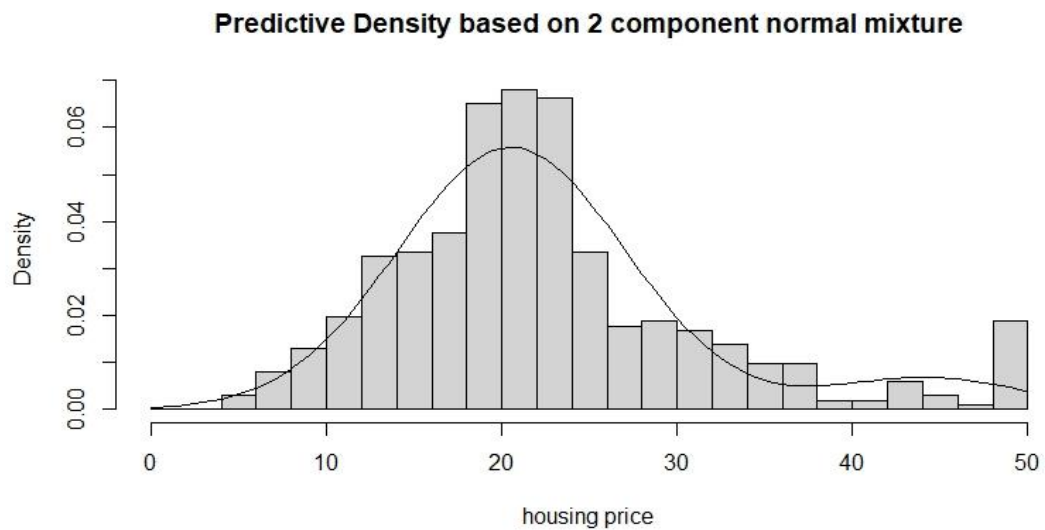
2. Calculate  $n_j = \sum_{i=1}^N z_{ij}$ ,  $\bar{y}_j = \left( \sum_{i=1}^N z_{ij} y_i \right) / n_j$ ,  $\hat{s}_j^2 = \sum_{i=1}^N z_{ij} (y_i - \bar{y}_j)^2$ .
3. Update  $\lambda \sim \text{Dirichlet}(n_1 + \alpha_1, n_2 + \alpha_2, \dots, n_k + \alpha_k)$ .
4. Update  $\mu_j \sim N \left( \frac{n_j \bar{y}_j + m_j \xi_j}{n_j + m_j}, \frac{\sigma_j^2}{n_j + m_j} \right)$ .
5. Update  $\sigma_j^2 \sim \text{InverseGamma} \left( \frac{n_j + m_j + 1}{2}, \frac{1}{2} \left[ s_j^2 + \hat{s}_j^2 + \frac{n_j m_j}{n_j + m_j} (\bar{y}_j - \xi_j)^2 \right] \right)$ .

## 4. DATA ANALYSIS AND RESULTS

### 4.1. METHOD OF ANALYSIS

Using the R software, the Gibbs sampler was run for 50000 iterations (10000 burn-in) starting with k=2 components and adding components until a satisfactory density plot and comparatively lower value of DIC is obtained.

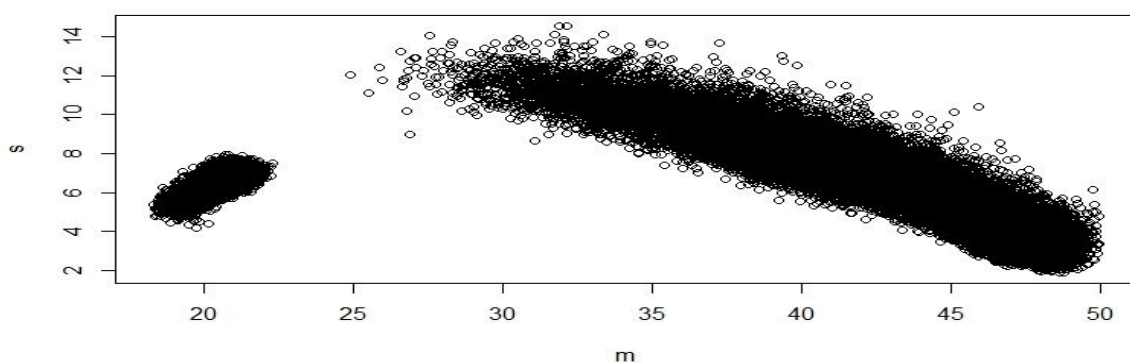
## 4.2. 2 Component Mixture Model



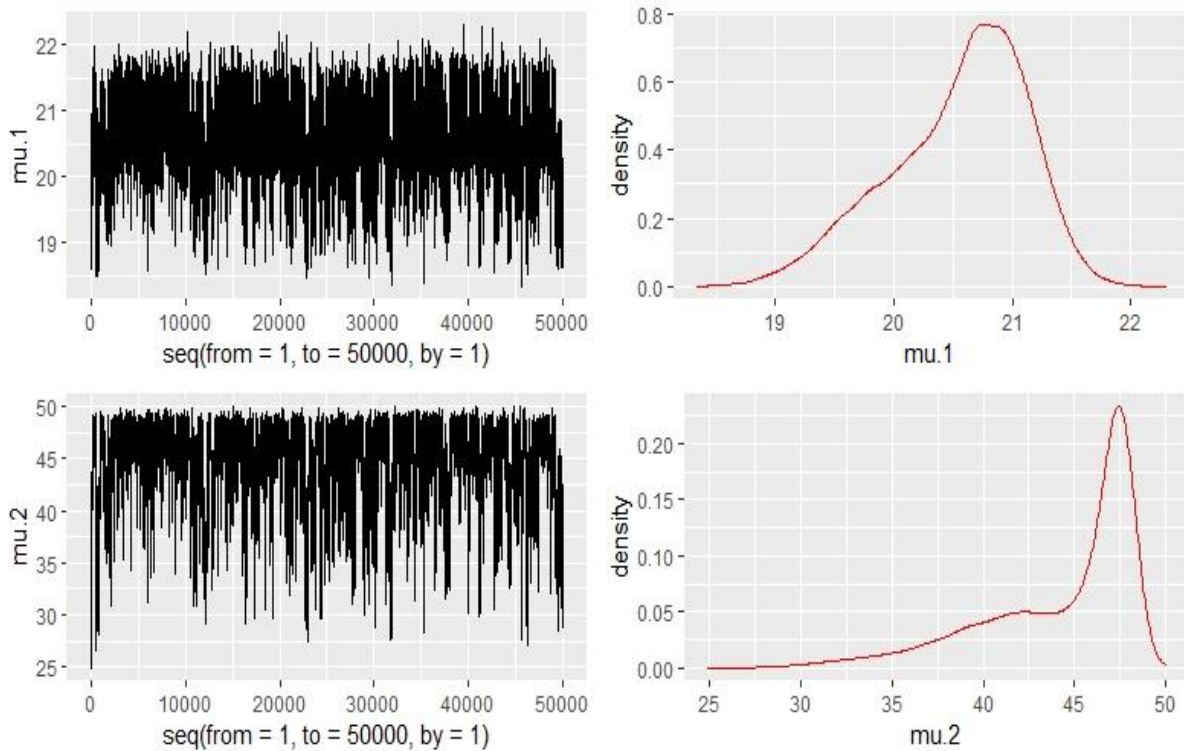
The above figure shows that the predictive density based on the 2 component Normal mixture model is not very accurate for the observed data. The parameter estimation for this model is given in the table below:

Component	Posterior mean	Posterior SD	Posterior proportion
1	20.541	6.49	0.9076562
2	44.28	5.452	0.09234384

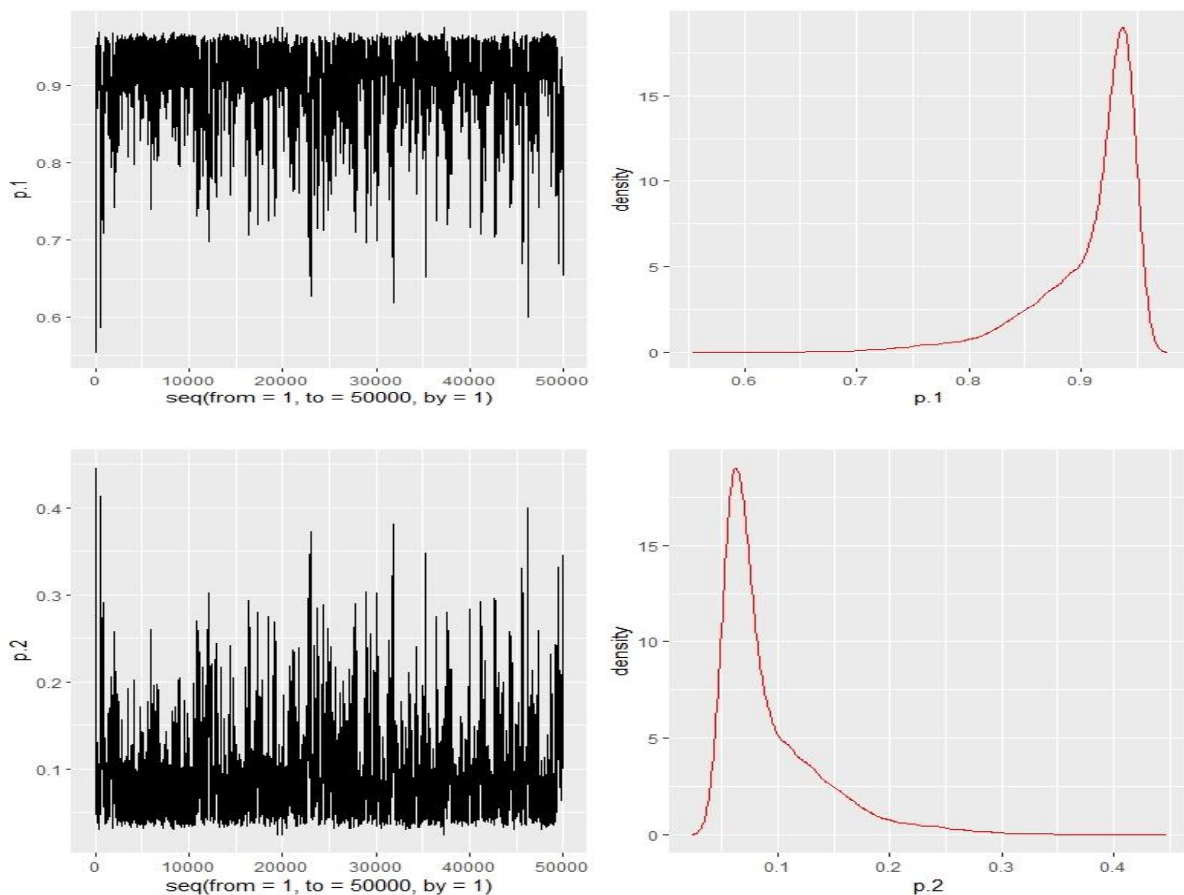
Here we have the mean vs sigma graph for the 2 component Normal mixture model. The data can be split into two categories without overlap. There is a considerable difference between the variance of the 2 groups. The mixture model assigns a very small weight to the component with the mean at 44.28.



### Trace plots and Density plots for $\mu$ parameters:

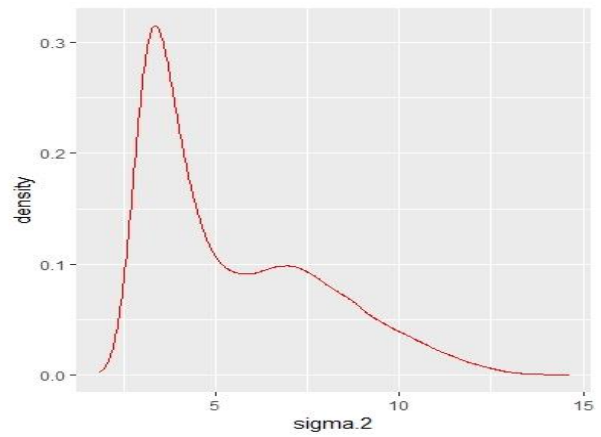
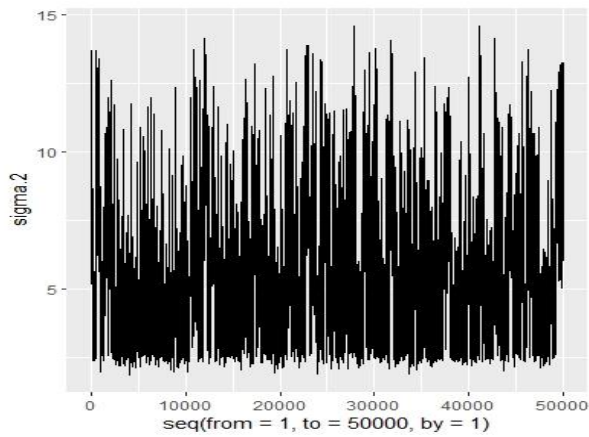
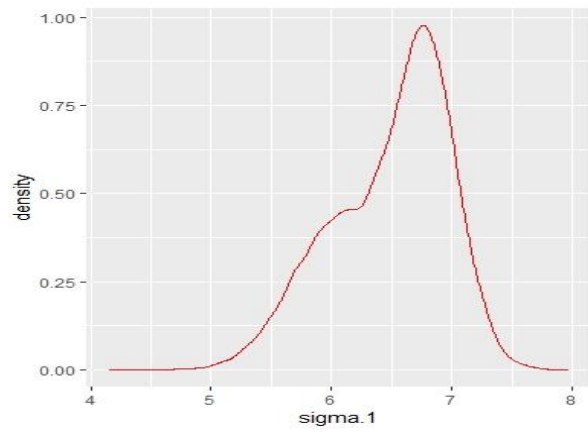
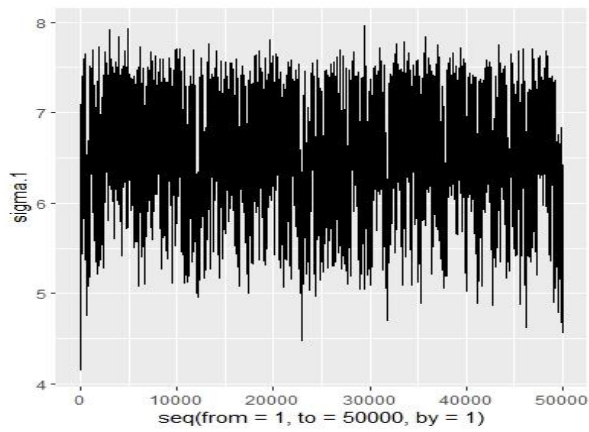


### Trace plots and Density plots for the $\lambda$ parameters:

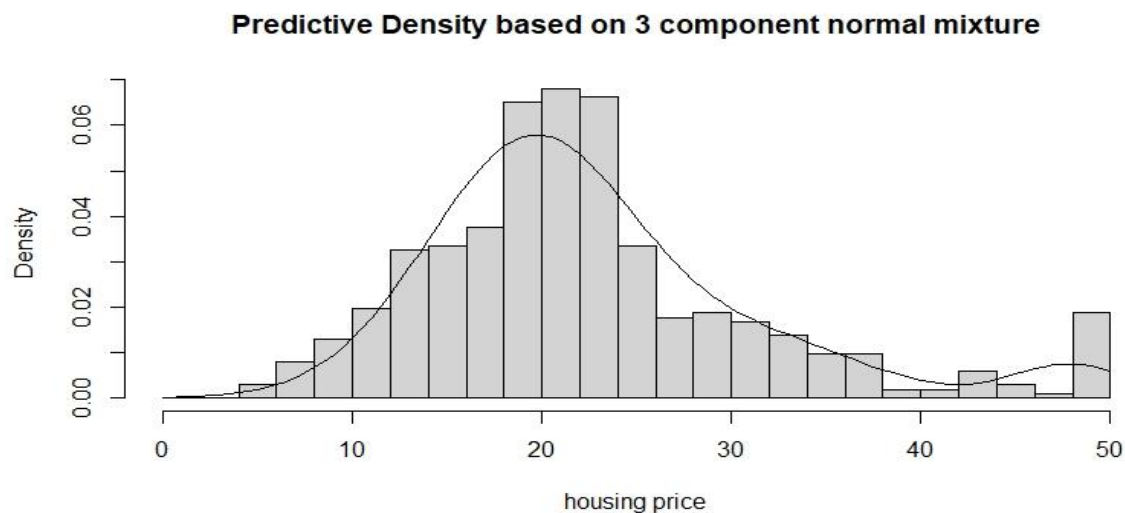




## Trace plots and Density plots for the $\sigma$ parameters:



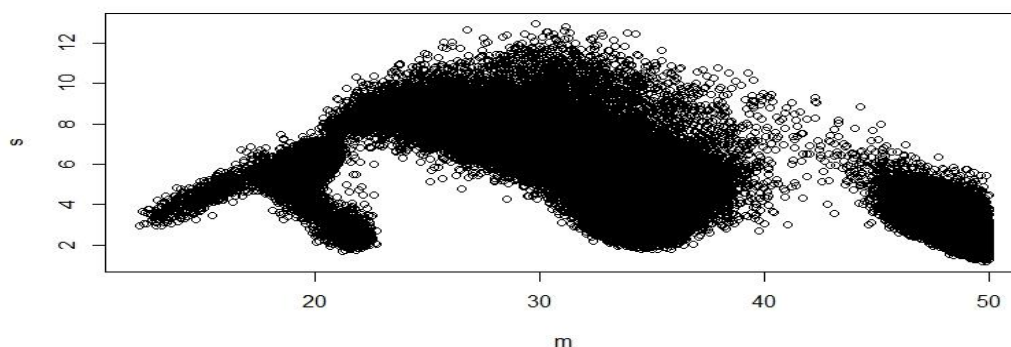
### 4.3. Three Component Mixture Model



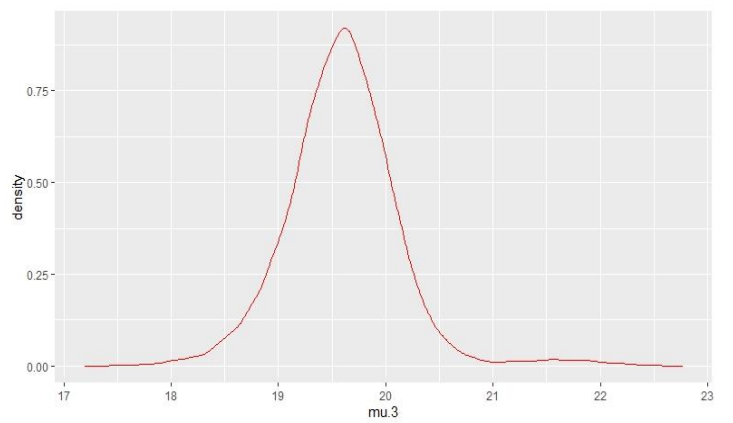
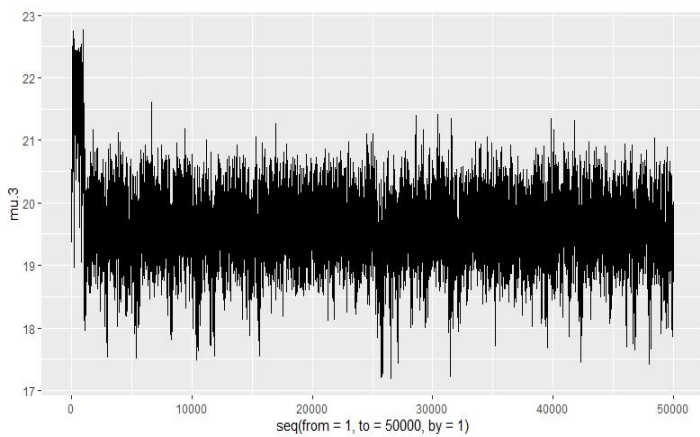
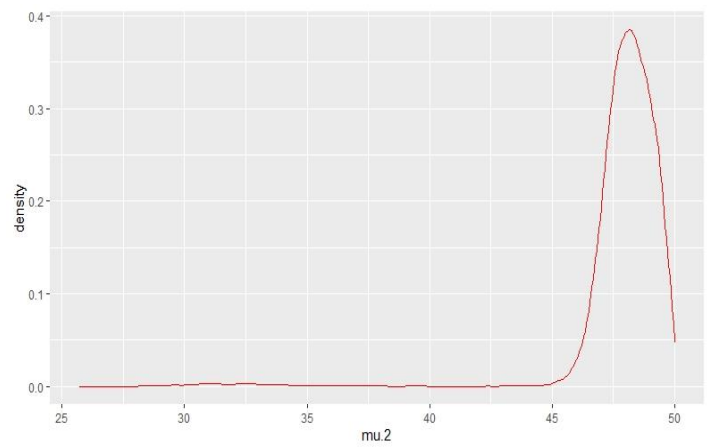
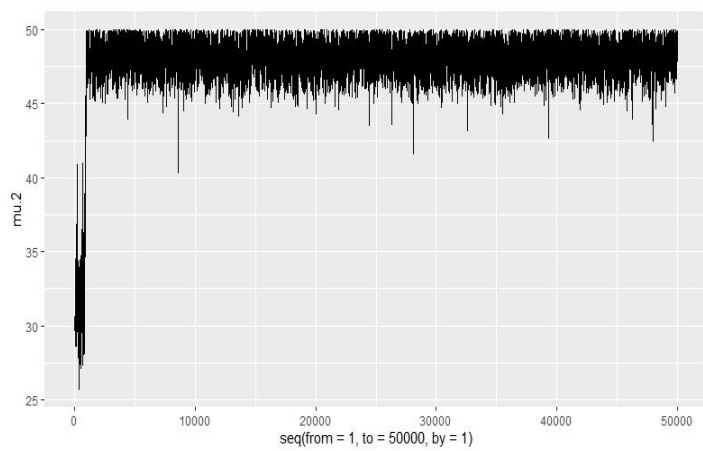
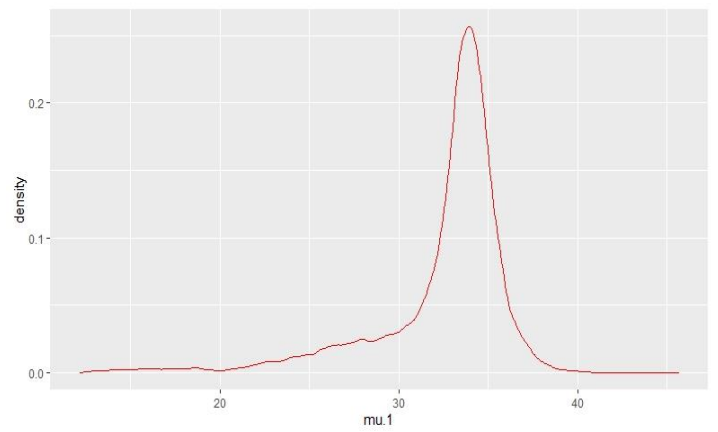
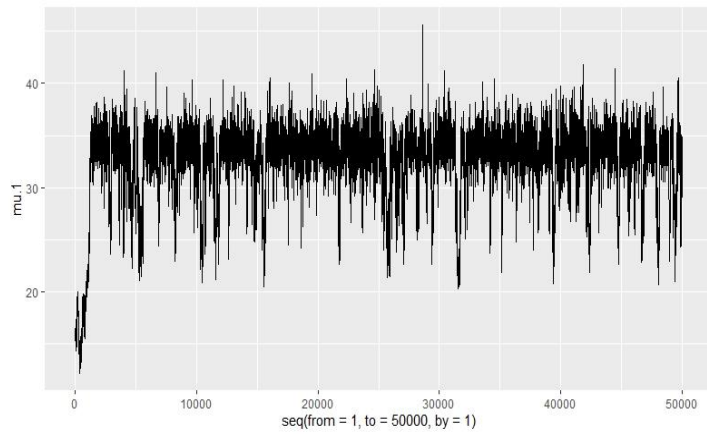
Simply by looking at the predictive density we can say that the 3-component normal mixture does not offer much improvement over the 2-component mixture model.

Component	Posterior mean	Posterior sigma	Posterior proportion
1	30.157	4.983	0.1638694
2	45.957	3.965	0.0859609
3	19.837	5.235	0.7501697

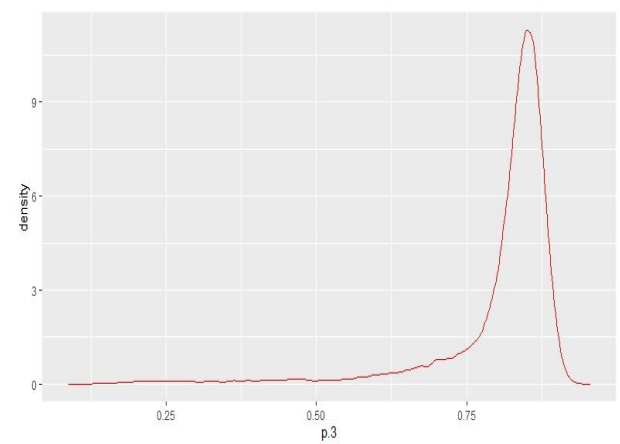
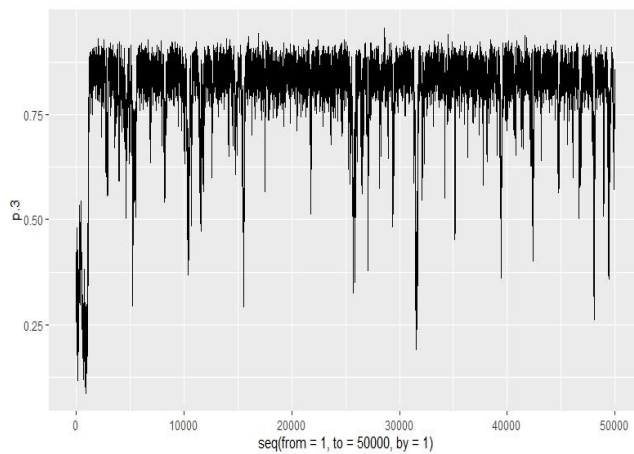
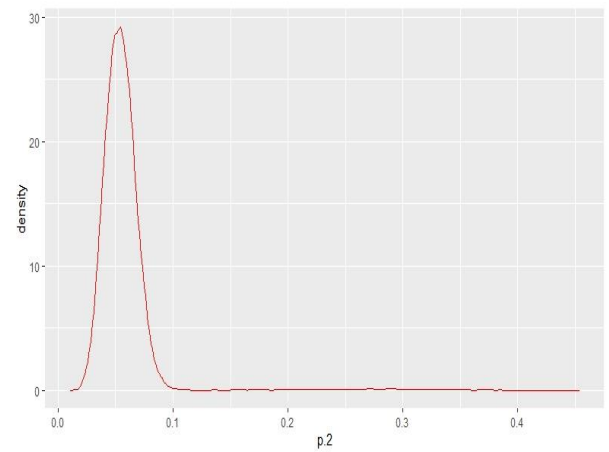
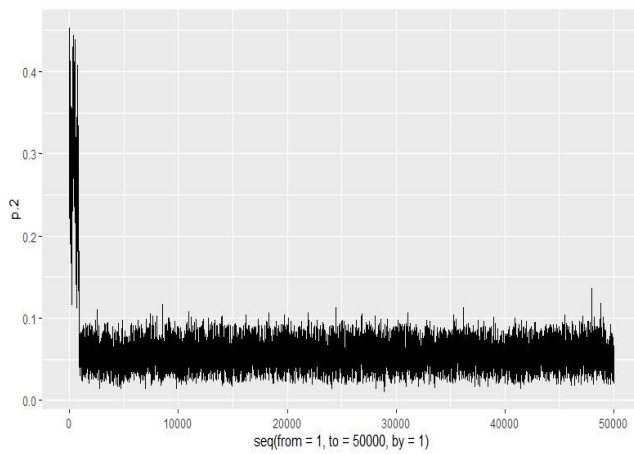
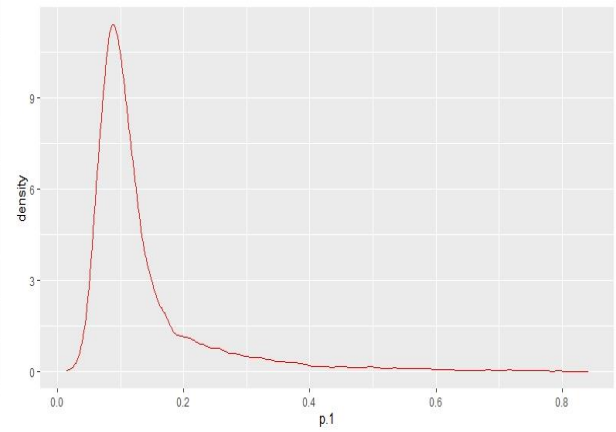
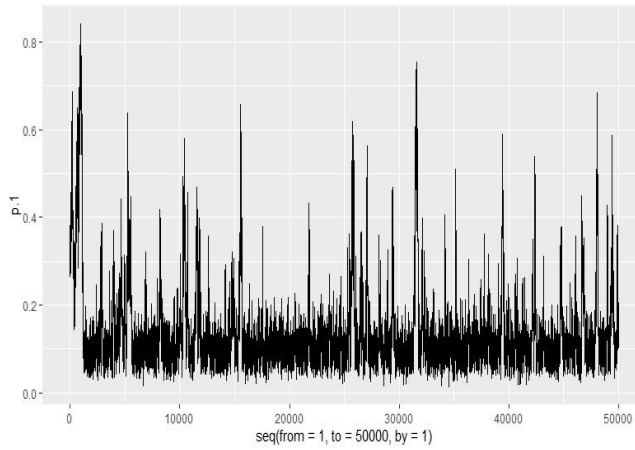
From the mean vs sigma graph we can see that the clusters are not well defined and there is overlap between them. The bigger cluster from before gets dissociated in 2 parts. The smaller cluster gets split in multiple directions one of which overlaps with one the other clusters.



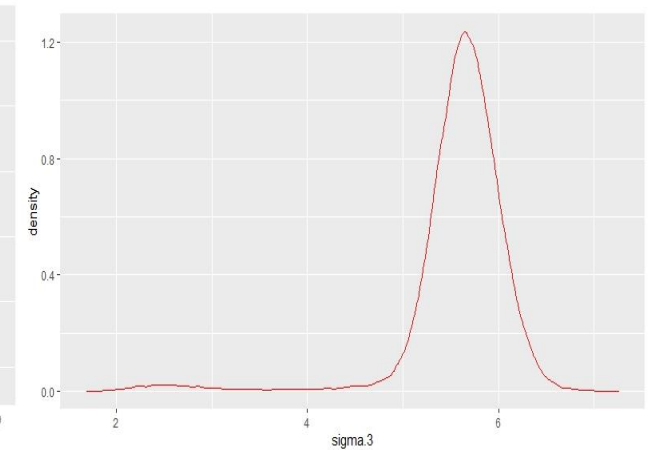
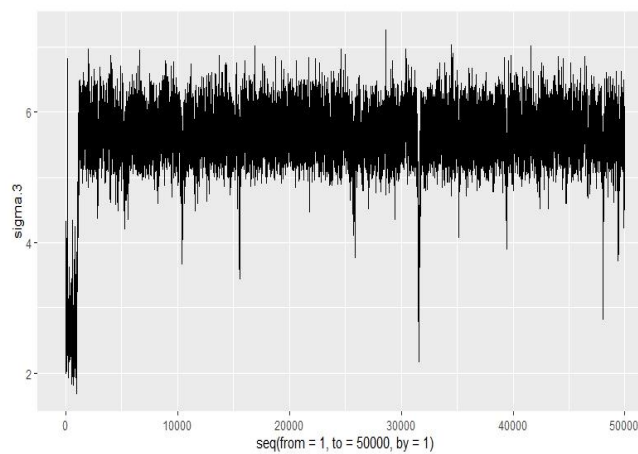
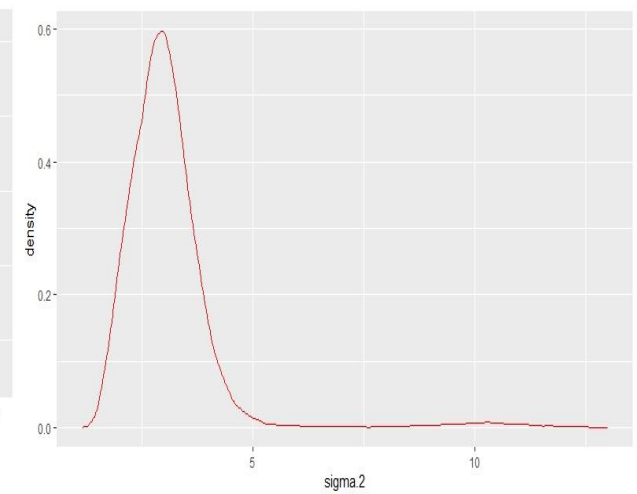
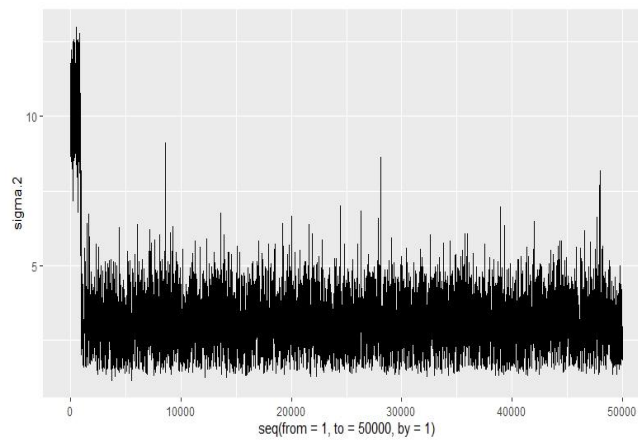
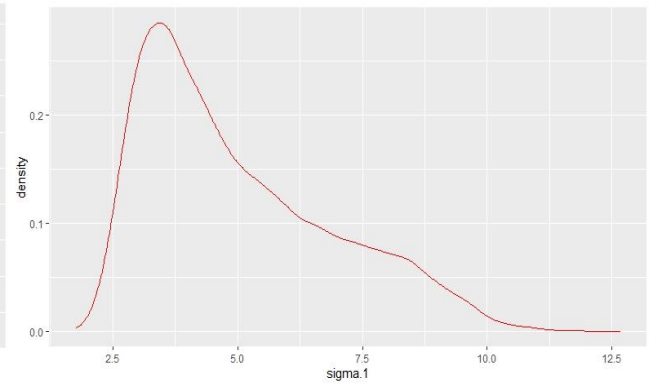
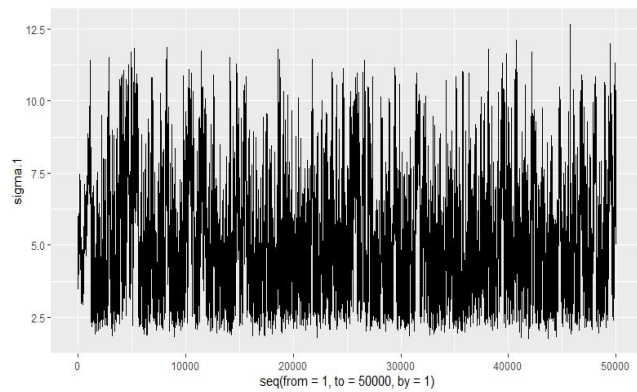
## Trace plots and Density plots for $\mu$ parameters:



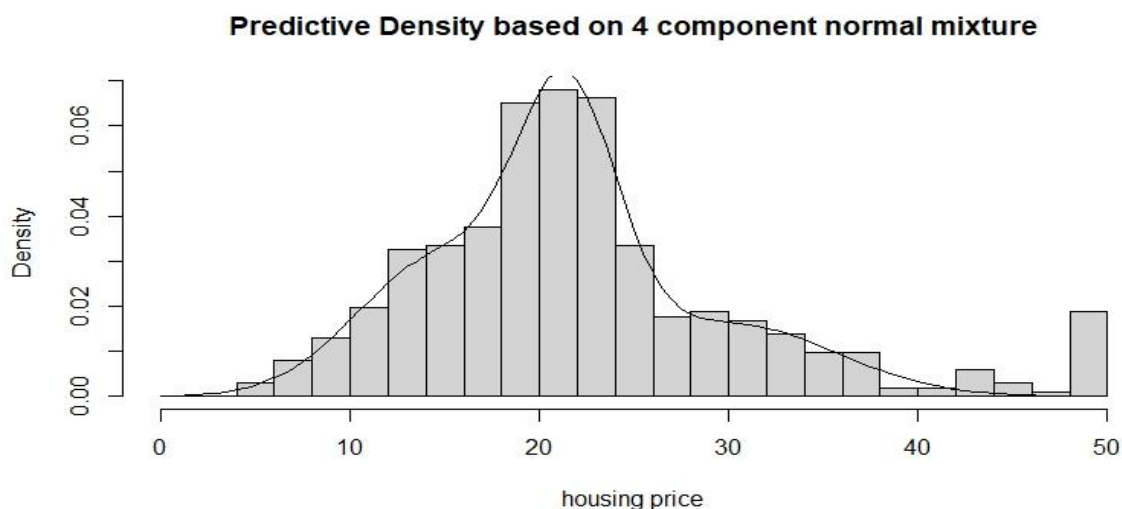
## Trace plots and Density plots for the $\lambda$ parameters:



## Trace plots and Density plots for the $\sigma$ parameters:



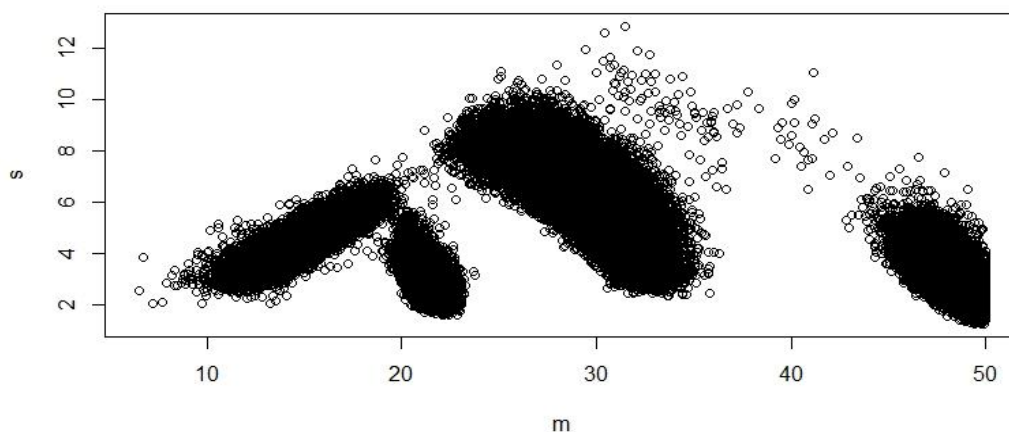
#### 4.4. Four Component Mixture Model



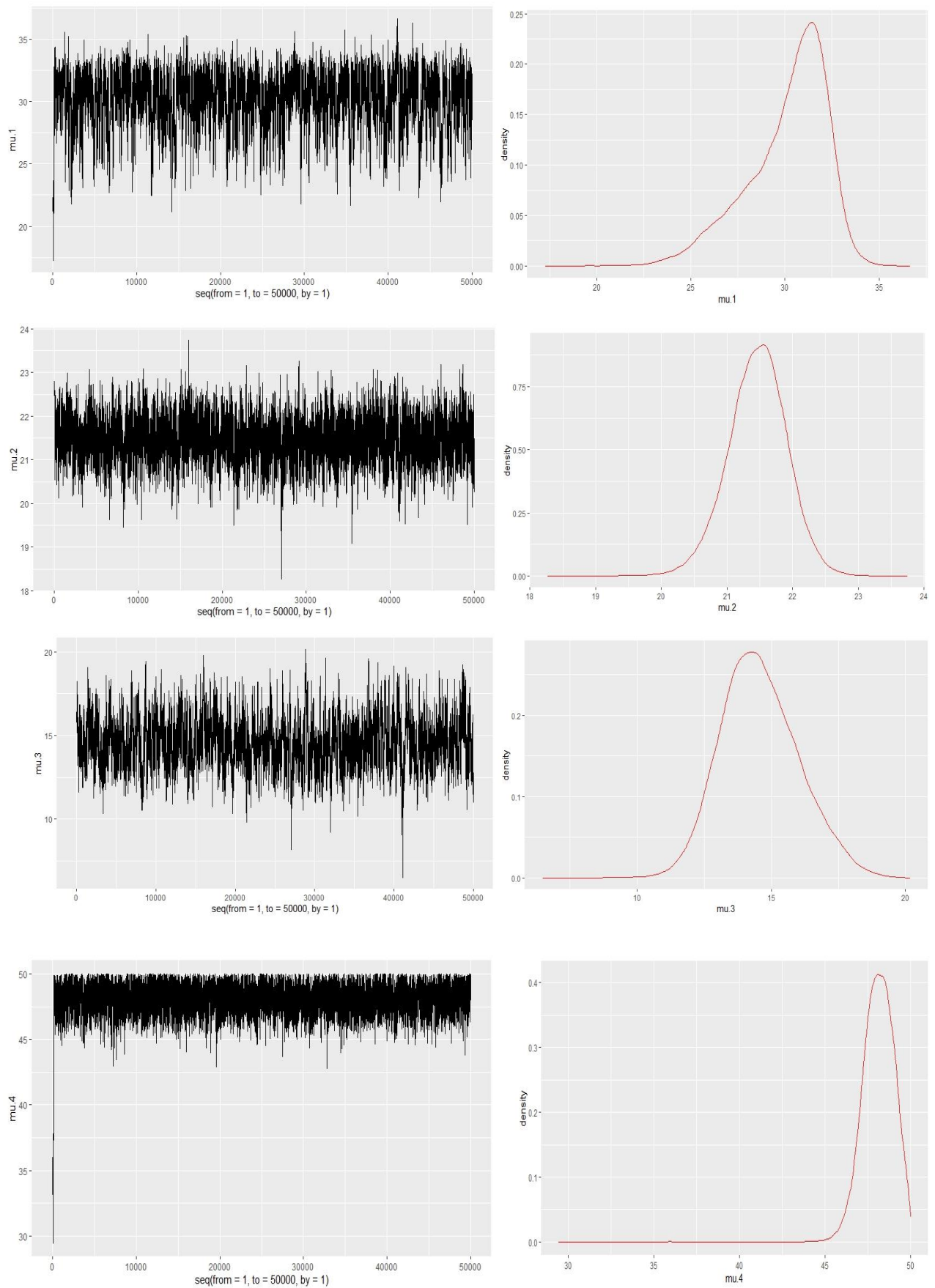
This seems to be a pretty good fit. Almost all the peaks in the data are accounted for by this predictive density. This is a significant improvement over the previous two fits.

Component	Posterior mean	Posterior SD	Posterior proportion
1	30.124	5.532	0.2220729
2	21.463	2.702	0.4000296
3	14.581	4.285	0.3230326
4	48.095	3.01	0.05486128

The mean vs sigma plot shows 4 clearly visible clusters. Apart from a slight overlap between the 2<sup>nd</sup> and 3<sup>rd</sup> components it does not show any irregularities.

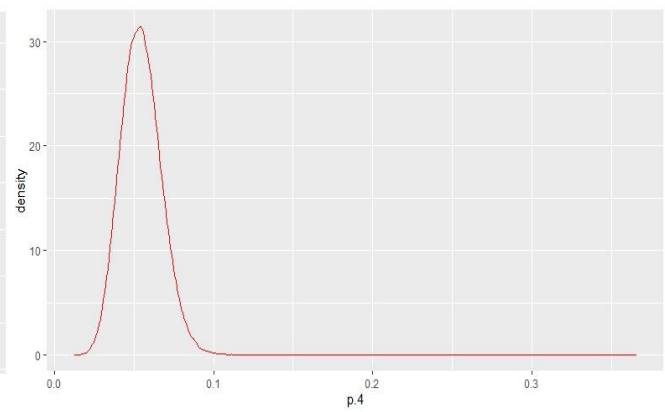
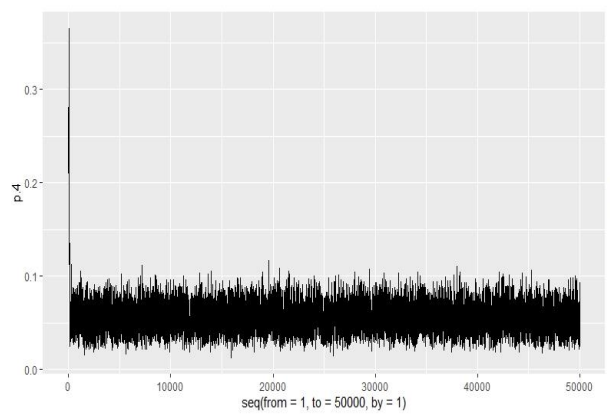
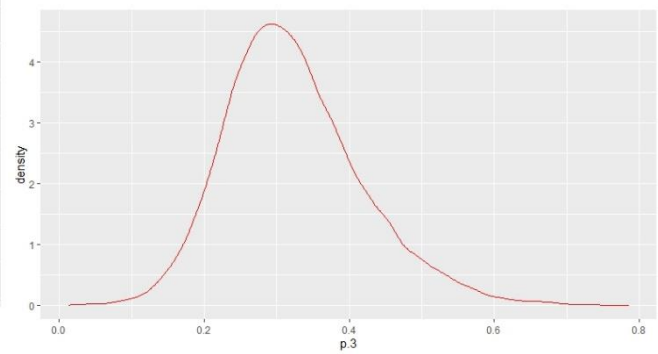
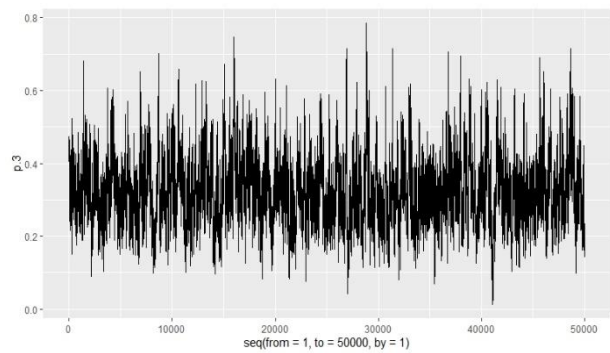
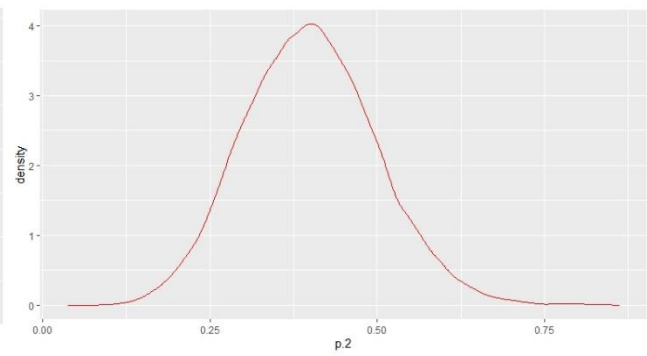
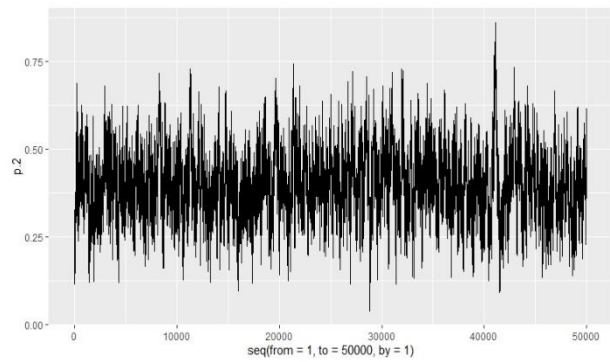
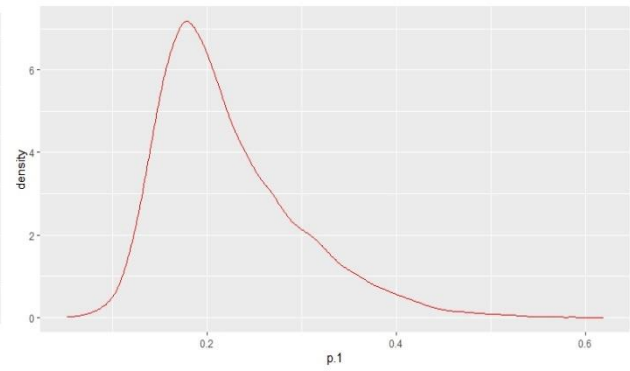
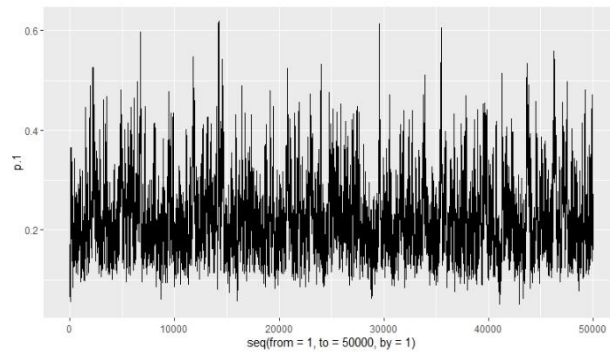


## Trace plots and Density plots for $\mu$ parameters:



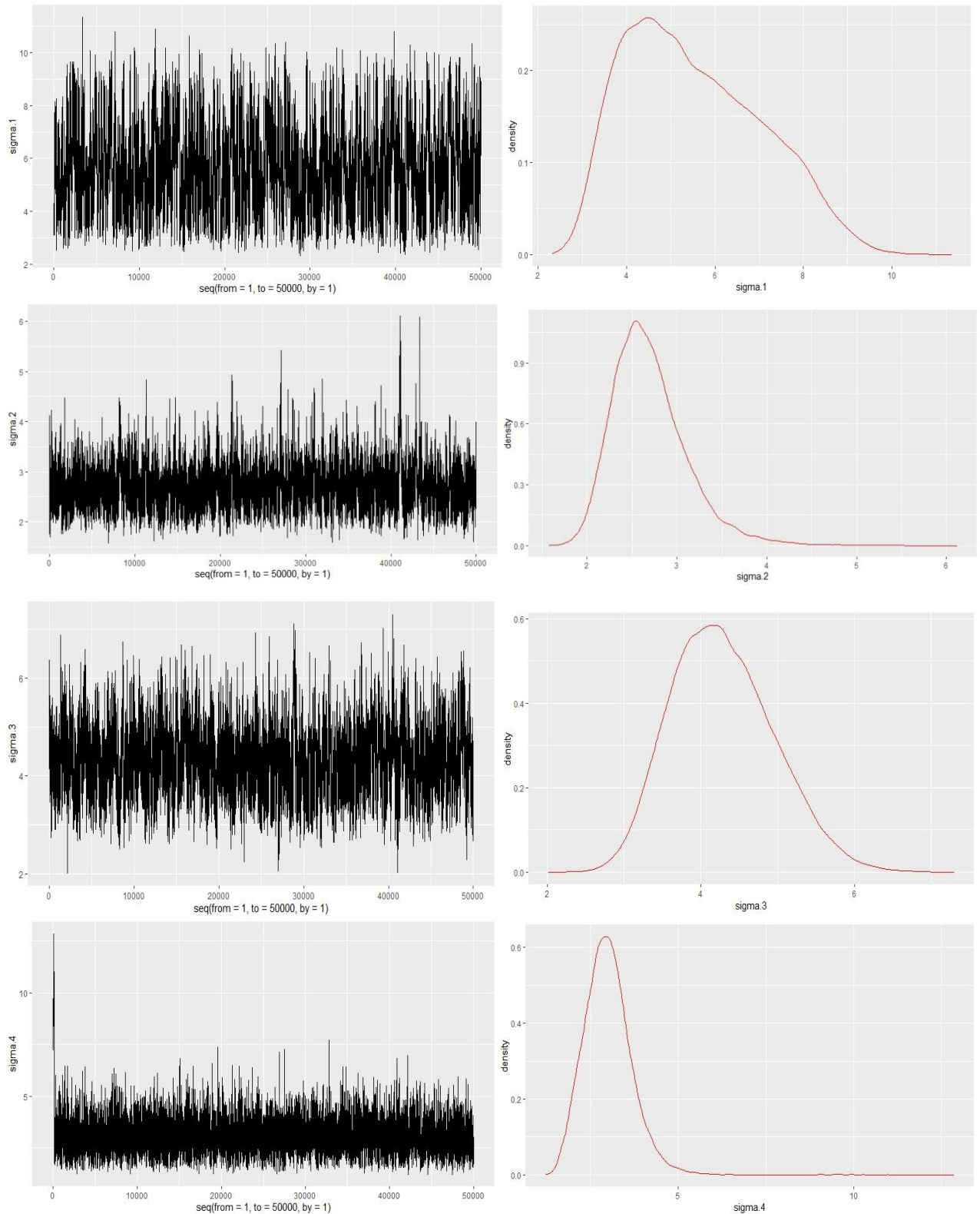


## Trace plots and Density plots for the $\lambda$ parameters:





## Trace plots and Density plots for the $\sigma$ parameters:



## 4.5. MODEL DIAGNOSTICS, RESULTS AND CONCLUSION

Based on the models fitted in the previous sections we now have to draw inference on the various aspects of a model. On observing the models, it is noticed that adding components to the mixture model gives no guarantee that the model will explain the data better. That being said we can say that the 4-component normal mixture model provides the best fit among these models.

- **Convergence:** We will use the trace plots from before to check whether the parameter estimates converge or not. A scattered trace plot, that is, one where the parameter estimates vary over a large range throughout the iterations indicates the possibility of not converging.

As a large number of iterations (50000) are taken, we do not consider multiple chains. This deals with problematic cases where long burn-in periods are required or there is high autocorrelation between the chains. Looking at the trace plots from the 3 models, we observe that almost all the parameter estimates show no signs of not converging.

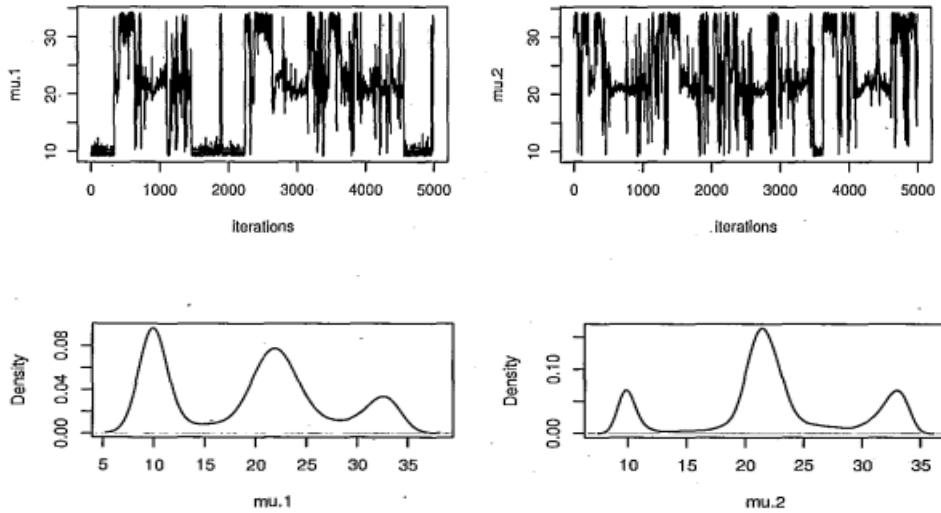
In particular we see in the 2-component model, the trace plots for  $\mu_2$  and  $\sigma_2$  seem to be quite scattered. This is also evidence that the 2-component model is inadequate for the data, since after parameter estimates of one component have been ascertained the parameter estimates of the other component varies over a wide range of values.

- **The Problem of Label Switching:** A problem frequently encountered while dealing with mixture models is that it is invariant under permutation of the indices of its components. Hence the components are not identifiable marginally. This is popularly known as “Label Switching”. It makes the exploration and maximization of the posterior surface more difficult. In some cases, it makes the posterior expectations of multiple parameters identical.

Label Switching in mixture models is characterized by multimodal posterior density of the parameters and highly fluctuating trace plots with considerable number of deviations from the central values.

Examples of trace plots and density plots where label switching occurred are

given below:



The trace plots and density plots for the parameters of our fitted models do not exhibit such patterns. To correct label switching, specific choice of priors is often required. As it is not present in our model, it can be suspected that our choice of prior may have been appropriate.

- **Comparison of Model Fit through DIC:** Deviance Information Criterion (DIC) is a hierarchical modelling generalization of the Akaike Information Criterion (AIC). It is very useful in Bayesian Model selection problems where posterior distributions have been obtained by MCMC simulation. Specifically in this case, we have used the formula provided below:

$$\begin{aligned}
 \text{DIC} &= \overline{D(\theta)} + p_D \\
 &= D(\tilde{\theta}) + 2p_D \\
 &= 2\overline{D(\theta)} - D(\tilde{\theta}) \\
 &= -4\mathbb{E}_{\theta}[\log f(\mathbf{y}|\theta)|\mathbf{y}] + 2\log f(\mathbf{y}|\tilde{\theta}).
 \end{aligned}$$

$$\text{DIC} = -4\mathbb{E}_{\theta}[\log(f(\mathbf{y}|\theta)|\mathbf{y})] + 2\log \hat{f}(\mathbf{y}), \text{ where } \hat{f}(\mathbf{y}) = \prod_{i=1}^n f(y_i).$$

Here our model is a normal mixture model with density

$$f(y|\theta) = \sum_{i=1}^K p_i \phi(y|\mu_i, \sigma_i^2),$$

we have

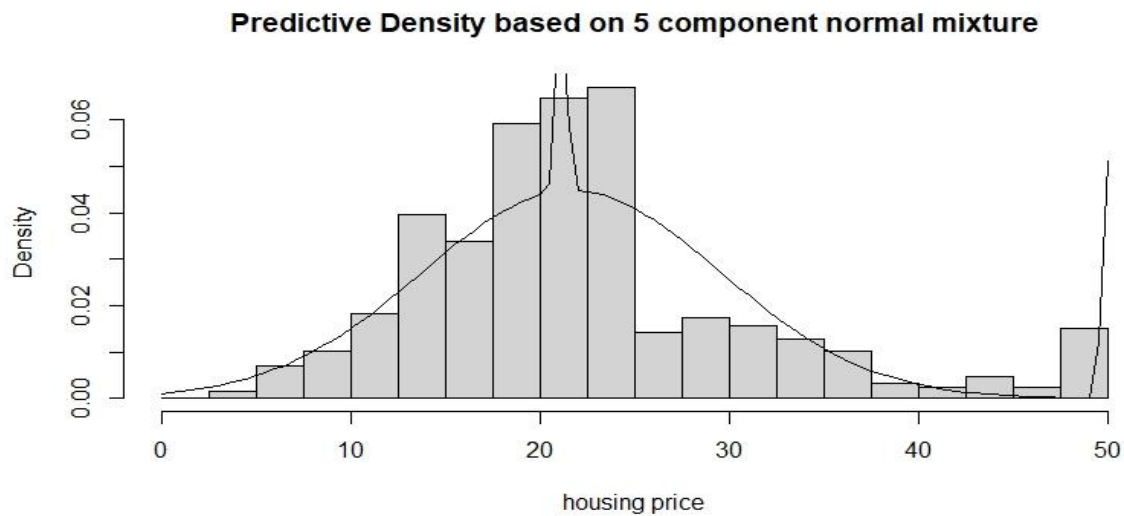
$$\hat{f}(y) = \frac{1}{m} \sum_{l=1}^m \sum_{i=1}^K p_i^{(l)} \phi(y|\mu_i^{(l)}, \sigma_i^{2(l)}) \approx \mathbb{E}_\theta[f(y|\theta)|\mathbf{y}],$$

The lower the DIC, the more relevant is the model.

The table given below shows that the DIC value decreases gradually as number of components are increased from k=2, reaches a minima at k=4 and then increases again.

Number of Components	Deviance Information Criterion
2	-14465.88
3	-14478.02
4	-14581.118
5	-14523.98

From these results we gain more confidence in the decision that the 4-component model is the best fit among these. On further increasing the number of components, the DIC is not only increased but the predictive density plot also shows a worse fit than before.



- **CONCLUSION:** This analysis yielded the conclusion that a 4-component normal mixture is a good fit for the housing values. The component normal mixtures are centered around 15, 20, 30 and 47. The components centered around 20 and 47 are assigned the highest and lowest proportions respectively. It is also observed from all the fitted models that the component centered around 47 consistently comprises of the least proportion in the model, the highest being 0.09234 in the 2-component mixture model.

Although the 4-component normal mixture model fits most of the housing values well, its shortcoming lies in not being able to explain the highest housing values of the data. As the proportion of the component with highest mean is very low, the predictive density is probably unable to fit that part of the data properly.

In the “About the Data” section earlier, we came across some variables which produced clustering when plotted against `medv`; and we suspected how they might play a role in determining the number of components in the normal mixture model. The analysis that has been performed on median housing values has been done without taking into account any of the other variables in the Boston dataset.

The scatterplot of median housing values vs property tax rates showed around 4 clusters. As a 4-component normal mixture model seems appropriate for the data, it supports our hypothesis that the property tax rate has a direct relation with the number sub-populations hidden in the distribution of housing prices.

## 5. BIBLIOGRAPHY

1. Jean-Michel Marin, Kerrie Mengerson and Christian P. Robert. “Bayesian Inference and Modelling on Mixtures of Distribution.”
2. Zhihui Liu (under supervision of Prof. Peter D.M MacDonald). “Bayesian Mixture Models”.
3. G. Celeux, F. Forbes, C.P Roberts and D.M Titterington. “Deviance Information Criteria for Missing Data Models”.
4. Jean-Michel Marin, Christian Robert. “Approximating the marginal likelihood in mixture models”.
5. Tatiana Tatarinova, Alan Schumitzky. “Non-Linear Mixture Models: A Bayesian Approach”.
6. Iljoon Chang, Seong W. Kim. “Modelling for Identifying Accident Prone Spots: Bayesian Approach with a Poisson Mixture Model”
7. Clair L. Alston, Kerrie L. Mengerson, Anthony N. Pettitt. “Case Studies in Bayesian Statistical Modelling and Analysis”.
8. Peter Congdon. “Bayesian Statistical Modelling”.

## 6. APPENDIX

### R Codes (For 4 component Normal Mixture Model)

#### Creating Gibbs Sampler

```
gibbsnorm <- function (dat, k, niter, alpha = 1.28, beta = 0.36 * var(dat),
                        lam = mean(dat), tau = 2.6/(diff(range(dat)))^2, g = 1)
{
  rigamma <- function(n, a, b) { return(1/rgamma(n, shape = a, rate = b))
                                }

  rdirichlet <- function(n, par) {
    k = length(par)
    z = array (0,dim=c (n, k))
    s = array (0, dim = c (n, 1) )
    for (i in 1 : k) {
      z [,i]= rgamma (n,shape= par[i] )
      s = s + z [ , i]
    }
    for (i in 1:k) {
      z[,i] = z[, i]/s
    }
    return(z)
  }

  n <- length(dat)
  mu <- rnorm(k, mean = mean(dat), sd = sd(dat))
  sig <- sd(dat)/k
  p <- rep(1/k, k)
  mixparam <- list(p = p, mu = mu, sig = sig)
  z <- rep(0, k)
```

```

nj <- z
sj <- z
sj2 <- z
gibbsmu <- matrix(0, nrow = niter, ncol = k)
gibbssig <- gibbsmu
gibbsp <- gibbsmu
for (i in 1:niter)
{
  for (t in 1:n)
  {
    prob <- mixparam$p * dnorm(dat[t], mean = mixparam$mu,
                                sd = mixparam$sig)
    z[t] <- sample(x = 1:k, size = 1, prob = prob)
  }
  for (j in 1 : k) {
    nj[j] <- sum(z == j)
    sj[j] <- sum(as.numeric(z == j) * dat)
  }
  repeat {
    gibbsmu[i, ] <- rnorm(k, mean = (lam * tau + sj)/(nj + tau),
                          sd = sqrt(mixparam$sig^2/(tau + nj)))
    if (max(gibbsmu[i, ]) < max(dat) & min(gibbsmu[i, ]) > min(dat))
      break
  }
  mixparam$mu <- gibbsmu[i, ]
  for(j in 1 : k) {
    sj2[j] = sum(as.numeric(z==j) * (dat - mixparam$mu[j])^2)
  }
  gibbssig[i,] <- sqrt(rgamma(k, alpha + 0.5 *(nj + 1),
                             beta + 0.5* tau *(mixparam$mu-lam)^2+ 0.5*sj2))
  mixparam$sig <- gibbssig[i, ]
}

```



```

gibbsp[i, ] <- rdirichlet(1, par = nj + g)

mixparam$p <- gibbsp[i, ]
}
data.frame(p = gibbsp, mu = gibbsmu, sigma = gibbsig)
}

```

## Running the Code for Required number of Components and Iterations

```
m4<-gibbsnorm(Boston$medv,4,50000)
```

## Code for Mixture Density Using Created Dataframe

```

mixdensity <- function(x,mu1,mu2,mu3,mu4,s1,s2,s3,s4,p1,p2,p3,p4) {
  den<-
  ((p1)*dnorm(x,mean=mu1,sd=s1))+((p2)*dnorm(x,mean=mu2,sd=s2))+((p3)*d
  norm(x,mean=mu3,sd=s3))
  +((p4)*dnorm(x,mean=mu4,sd=s4))
  return(den)
}

```

## Plotting the Predictive Density over the Histogram of the Data

```

p1<-mean(m4$p.1)
p2<-mean(m4$p.2)
p3<-mean(m4$p.3)
p4<-mean(m4$p.4)

```

```

mu1<-mean(m4$mu.1)
mu2<-mean(m4$mu.2)

```

```

mu3<-mean(m4$mu.3)
mu4<-mean(m4$mu.4)

s1<-mean(m4$sigma.1)
s2<-mean(m4$sigma.2)
s3<-mean(m4$sigma.3)
s4<-mean(m4$sigma.4)

par(mfrow=c(1,1))

hist(Boston$medv,xlab="housing price",main="Predictive Density based on 4
component normal mixture",
      probability = TRUE,breaks=seq(0,50,by=2))

curve(mixeddensity(x,mu1,mu2,mu3,mu4,s1,s2,s3,s4,p1,p2,p3,p4),range(0,50),ad
d=TRUE)

```

## Plotting Traceplots

```

library(ggplot2)

p <- ggplot(m4, aes(x=seq(from=1,to=50000,by=1), y=mu.2)) +
  geom_line()

p

```

## Plotting Mean vs Sigma Graphs

```

m<-c(m42$mu.1,m42$mu.2,m42$mu.3,m42$mu.4)

s<-c(m42$sigma.1,m42$sigma.2,m42$sigma.3,m42$sigma.4)

par(mfrow=c(1,1))

plot(m,s)

```

## Finding BIC value for the fitted model

```

zm4=matrix(NA,nrow=50000,ncol=506)

for(i in 1:50000){
  for(j in 1:506)

```

```

{ zm4[i,j]<-
mixdensity4(x[j],m4$mu.1[i],m4$mu.2[i],m4$mu.3[i],m4$mu.4[i],
            m4$sigma.1[i],m4$sigma.2[i],m4$sigma.3[i],m4$sigma.4[i],
            m4$p.1[i],m4$p.2[i],m4$p.3[i],m4$p.4[i])
}
}}
logzm4<-log(zm4)
lik4<-c()
for(i in 1:50000){
  for(j in 1:506){
    lik4[i]=sum(logzm4[i,j])
  }
}
mean(lik4)
l4<-c()
for(i in 1:50000){
  for(j in 1:506){
    l4[j]=sum(zm4[i,j])/50000
  }
}
ll4<-log(l4)
mean(ll4)

(-4*mean(lik4))+(2*sum(ll4))

```