

Assignment-4-Mayank Raunak

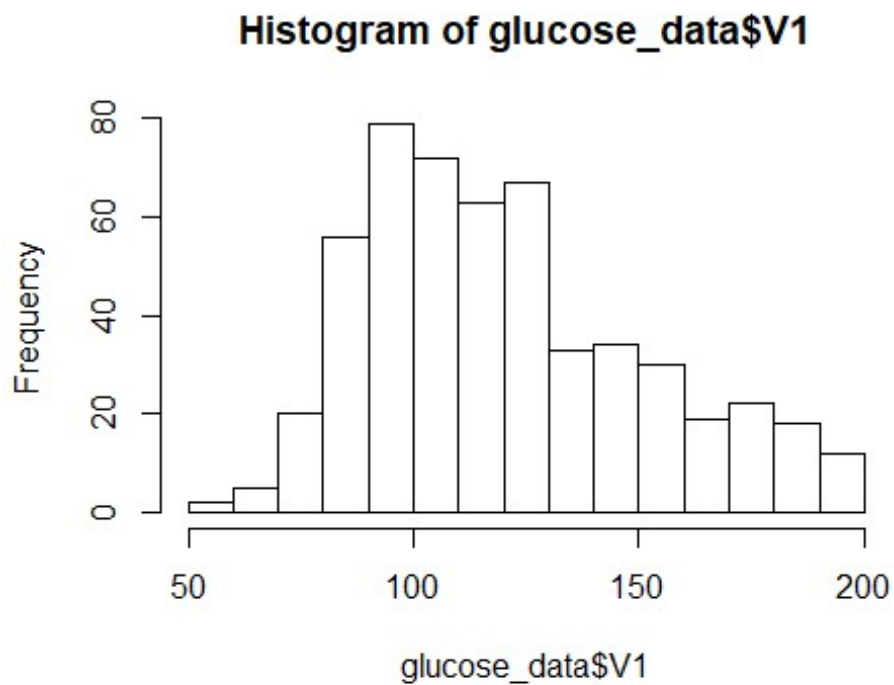
November 11, 2019

6.2

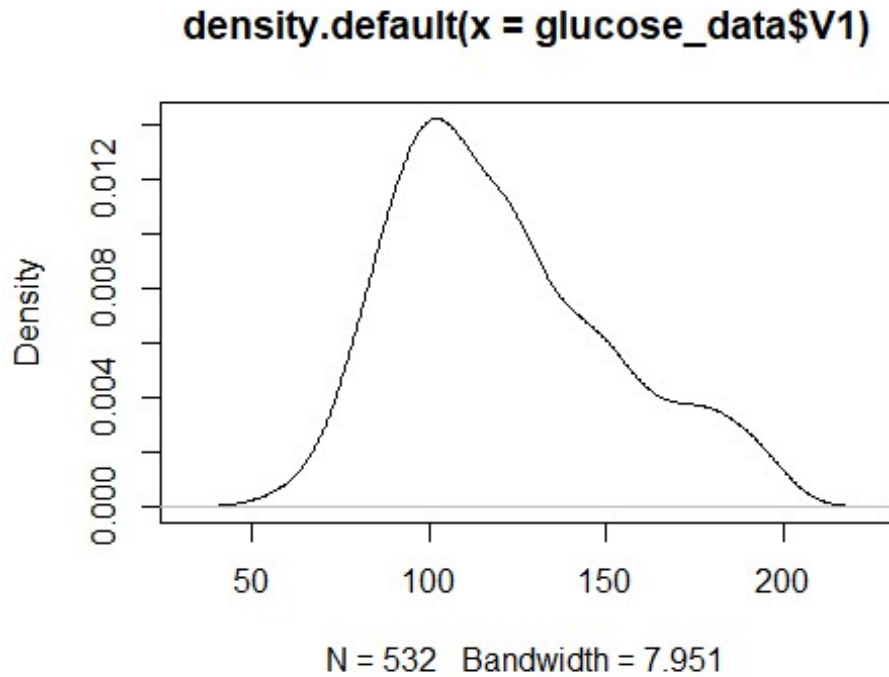
```
glucose_data <-  
read.table('http://www.stat.washington.edu/~pdhoff/Book/Data/hwdata/glucose.dat')
```

Part: a Make a histogram or kernel density estimate of the data. Describe how this empirical distribution deviates from the shape of a normal distribution

```
library(ggplot2)  
h<-hist(glucose_data$V1)
```



```
d<-density(glucose_data$V1)  
plot(d)
```



The normal distribution is symmetric and have same mean and median value. But in this case the empirical distribution appears to be skewed right significantly. A right-skewed distribution has a long right tail. Right-skewed distributions are also called positive-skew distributions. That's because there is a long tail in the positive direction on the number line. The mean is also to the right of the peak.

#Part B: Obtain the full conditional distributions of (X_1, \dots, X_n) , p , θ_1 , θ_2 , σ_1^2 , σ_2^2 ,

we know,

$$x_i \sim \text{Bernoulli} \left(\frac{p \times \text{dnorm}(y_i, \theta_1, \sigma_1^2)}{p \times \text{dnorm}(y_i, \theta_1, \sigma_1^2) + (1 - p) \times \text{dnorm}(y_i, \theta_2, \sigma_2^2)} \right)$$

$$X_i = 1$$

$$\begin{aligned} P(X_i = 1 \mid y_i, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) &= \frac{P(X_i = 1 \mid p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \times p(y_i \mid X_i = 1, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2)}{P(y_i \mid p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2)} \\ &= \frac{P(X_i = 1 \mid p) \times p(y_i \mid X_i = 1, \theta_1, \sigma_1^2)}{P(X_i = 1 \mid p) \times p(y_i \mid X_i = 1, \theta_1, \sigma_1^2) + P(X_i = 2 \mid p) \times p(y_i \mid X_i = 2, \theta_2, \sigma_2^2)} \\ &= \frac{p \times \text{dnorm}(y_i, \theta_1, \sigma_1^2)}{p \times \text{dnorm}(y_i, \theta_1, \sigma_1^2) + (1 - p) \times \text{dnorm}(y_i, \theta_2, \sigma_2^2)} \end{aligned}$$

$$\begin{aligned}
P(X_i = 2 \mid y_i, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) &= \frac{P(X_i = 2 \mid p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \times p(y_i \mid X_i = 2, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2)}{P(y_i \mid p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2)} \\
&= \frac{P(X_i = 2 \mid p) \times p(y_i \mid X_i = 2, \theta_2, \sigma_2^2)}{P(X_i = 2 \mid p) \times p(y_i \mid X_i = 2, \theta_2, \sigma_2^2) + P(X_i = 1 \mid p) \times p(y_i \mid X_i = 1, \theta_1, \sigma_1^2)} \\
&= \frac{p \times \text{dnorm}(y_i, \theta_2, \sigma_2^2)}{p \times \text{dnorm}(y_i, \theta_2, \sigma_2^2) + (1 - p) \times \text{dnorm}(y_i, \theta_1, \sigma_1^2)}
\end{aligned}$$

p

For this and later calculations, let $n_1 = \sum x_i$ (i.e. number of 1s in \mathbf{x}) and $n_2 = n - n_1$ (i.e. number of 2s).

$$\begin{aligned}
p(p \mid \mathbf{x}, \mathbf{y}, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) &\propto p(p) \times p(\mathbf{x}, \mathbf{y}, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2 \mid p) \\
&\propto p(p) \times p(\mathbf{x} \mid p) p(\mathbf{y} \mid \mathbf{x}, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) p(\theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \\
&\propto p(p) \times p(\mathbf{x} \mid p) \\
&\propto \text{dbeta}(p, a, b) \times \text{dbinom}(n_1, n, p) \\
&\propto p^{a-1} (1-p)^{b-1} \times p^{n_1} (1-p)^{n_2} \\
&= p^{a+n_1-1} (1-p)^{b+n_2-1} \\
&= \text{dbeta}(p, a + n_1, b + n_2)
\end{aligned}$$

θ_1

Let $\mathbf{y}_1 = \{y_i \in \mathbf{y} : x_i = 1\}$ and $\mathbf{y}_2 = \{y_i \in \mathbf{y} : x_i = 2\}$

$$\begin{aligned}
p(\theta_1 \mid \mathbf{x}, \mathbf{y}, p, \theta_2, \sigma_1^2, \sigma_2^2) &\propto p(\theta_1 \mid \mathbf{x}, p, \theta_2, \sigma_1^2, \sigma_2^2) \times p(\mathbf{y} \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \\
&\propto p(\theta_1) \times \prod_{i=1}^n (\text{dnorm}(y_i, \theta_1, \sigma_1^2)^{x_i} \text{dnorm}(y_i, \theta_2, \sigma_2^2)^{1-x_i}) \\
&\propto \text{dnorm}(\theta_1, \mu_0, \tau_0^2) \times \prod_{i=1}^n \text{dnorm}(y_i, \theta_1, \sigma_1^2)^{x_i} \\
&\propto \text{dnorm}(\theta_1, \mu_0, \tau_0^2) \times \prod_{y \in \mathbf{y}_1} \text{dnorm}(y, \theta_1, \sigma_1^2) \\
&\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta_1 - \mu_0)^2\right) \times \prod_{y \in \mathbf{y}_1} \exp\left(-\frac{1}{2\sigma_1^2}(y - \theta_1)^2\right) \\
&\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta_1 - \mu_0)^2\right) \times \exp\left(-\frac{1}{2\sigma_1^2} \sum_{y \in \mathbf{y}_1} (y - \theta_1)^2\right) \\
&\propto \mathcal{N}(\tau_{n,1}^2, \mu_{n,1})
\end{aligned}$$

where

$$\tau_{n,1}^2 = \frac{1}{\frac{1}{\tau_0^2} + \frac{n_1}{\sigma_1^2}}$$

$$\mu_{n,1} = \frac{\frac{1}{\tau_0^2} \mu_0 + \frac{n_1}{\sigma_1^2} \bar{y}_{\cdot,1}}{\frac{1}{\tau_0^2} + \frac{n_1}{\sigma_1^2}}$$

θ_2

$$\begin{aligned} p(\theta_2 \mid \mathbf{x}, \mathbf{y}, p, \theta_2, \sigma_1^2, \sigma_2^2) &\propto p(\theta_2 \mid \mathbf{x}, p, \theta_1, \sigma_1^2, \sigma_2^2) \times p(\mathbf{y} \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \\ &\propto p(\theta_2) \times \prod_{i=1}^n (\text{dnorm}(y_i, \theta_1, \sigma_1^2)^{x_i} \text{dnorm}(y_i, \theta_2, \sigma_2^2)^{1-x_i}) \\ &\propto \text{dnorm}(\theta_2, \mu_0, \tau_0^2) \times \prod_{i=1}^n \text{dnorm}(y_i, \theta_2, \sigma_2^2)^{1-x_i} \\ &\propto \text{dnorm}(\theta_2, \mu_0, \tau_0^2) \times \prod_{y \in \mathbf{y}_2} \text{dnorm}(y, \theta_2, \sigma_2^2) \\ &\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta_2 - \mu_0)^2\right) \times \prod_{y \in \mathbf{y}_2} \exp\left(-\frac{1}{2\sigma_2^2}(y - \theta_2)^2\right) \\ &\propto \exp\left(-\frac{1}{2\tau_0^2}(\theta_2 - \mu_0)^2\right) \times \exp\left(-\frac{1}{2\sigma_2^2} \sum_{y \in \mathbf{y}_2} (y - \theta_2)^2\right) \\ &\propto \mathcal{N}(\mu_{n,2}, \tau_{n,2}^2) \end{aligned}$$

$$\tau_{n,2}^2 = \frac{1}{\frac{1}{\tau_0^2} + \frac{n_2}{\sigma_2^2}}$$

$$\mu_{n,2} = \frac{\frac{1}{\tau_0^2} \mu_0 + \frac{n_2}{\sigma_2^2} \bar{y}_2}{\frac{1}{\tau_0^2} + \frac{n_2}{\sigma_2^2}}$$

σ_1^2

As probably expected, this will look like inference for a standard normal model, except using only the data in group 1.

$$\begin{aligned}
p(\sigma_1^2 \mid \mathbf{x}, \mathbf{y}, p, \theta_1, \theta_2, \sigma_2^2) &\propto p(\sigma_1^2 \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_2^2) \times p(\mathbf{y} \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \\
&\propto p(\sigma_1^2) \times \prod_{i=1}^n (\text{dnorm}(y_i, \theta_1, \sigma_1^2)^{x_i} \text{dnorm}(y_i, \theta_2, \sigma_2^2)^{1-x_i}) \\
&\propto \text{inverse} - \text{gamma}(\sigma_1^2, \nu_0, \sigma_0^2 \nu_0 / 2) \times \prod_{y \in \mathbf{y}_1} \text{dnorm}(y, \theta_1, \sigma_1^2) \\
&\propto \exp\left((\sigma_1^2)^{-(\nu_0/2)-1} \exp\left(-\frac{1}{\sigma_1^2} \sigma_0^2 \nu_0 / 2\right)\right) \times (\sigma_1^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_1^2} \sum_{y \in \mathbf{y}_1} (y - \theta_1)^2\right) \\
&\propto \text{inverse} - \text{gamma}(\nu_{n,1}/2, \sigma_{n,1}^2(\theta_1) \nu_{n,1}/2)
\end{aligned}$$

where

$$\begin{aligned}
\nu_{n,1} &= \nu_0 + n_1 \\
\sigma_{n,1}^2(\theta_1) &= \frac{1}{\nu_{n,1}} [\nu_0 \sigma_0^2 + n_1 s_{n,1}^2(\theta_1)]
\end{aligned}$$

σ_2^2

$$\begin{aligned}
p(\sigma_2^2 \mid \mathbf{x}, \mathbf{y}, p, \theta_1, \theta_2, \sigma_1^2) &\propto p(\sigma_2^2 \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_1^2) \times p(\mathbf{y} \mid \mathbf{x}, p, \theta_1, \theta_2, \sigma_1^2, \sigma_2^2) \\
&\propto p(\sigma_2^2) \times \prod_{i=1}^n (\text{dnorm}(y_i, \theta_1, \sigma_1^2)^{x_i} \text{dnorm}(y_i, \theta_2, \sigma_2^2)^{1-x_i}) \\
&\propto \text{inverse} - \text{gamma}(\sigma_2^2, \nu_0, \sigma_0^2 \nu_0 / 2) \times \prod_{y \in \mathbf{y}_1} \text{dnorm}(y, \theta_2, \sigma_2^2) \\
&\propto \exp\left((\sigma_2^2)^{-(\nu_0/2)-1} \exp\left(-\frac{1}{\sigma_2^2} \sigma_0^2 \nu_0 / 2\right)\right) \times (\sigma_2^2)^{-n/2} \exp\left(-\frac{1}{2\sigma_2^2} \sum_{y \in \mathbf{y}_1} (y - \theta_2)^2\right) \\
&\propto \text{inverse} - \text{gamma}\left(\frac{\nu_{n,2}}{2}, \frac{\sigma_{n,2}^2(\theta_2) \nu_{n,2}}{2}\right) \\
\nu_{n,2} &= \nu_0 + n_2 \\
\sigma_{n,2}^2(\theta_2) &= \frac{1}{\nu_{n,2}} [\nu_0 \sigma_0^2 + n_2 s_{n,2}^2(\theta_2)]
\end{aligned}$$

Part c: Compute and plot the autocorrelation functions of $\theta(s)$ (1) and $\theta(s)$ (2), as well as their effective sample sizes.

```

Y = glucose_data$V1
n = length(Y)
# Priors given in the question
a = b = 1
mu0 = 120
t20 = 200

```

```

s20 = 1000
nu0 = 10

# We need to run the Gibb's sampler for 10000 iterations
S = 10000

# We need to store the values of Theta1 and Theta2 for the computation of autocorrelation between them.

THETA1 = numeric(S)
THETA2 = numeric(S)

# Posterior predictive
YPRED = numeric(S)
# Assuming the initial values in order to perform Gibb's sampling

Y = glucose_data$V1 (from the data)

n = length(Y)
p = 0.5
theta1 = theta2 = mean(Y)
s21 = s22 = var(Y)
# Gibbs sampling
for (s in 1:S) {

  # dnorm for
  p1 = p * dnorm(Y, theta1, sqrt(s21))
  p2 = (1 - p) * dnorm(Y, theta2, sqrt(s22))
  bernoulli_p = p1 / (p1 + p2)
  X = rbinom(n, 1, bernoulli_p)

  # With X sample, calculate group-specific summary statistics
  n1 = sum(X)
  n2 = n - n1
  y1 = Y[X == 1]
  y2 = Y[X == 2]
  ybar1 = mean(y1)
  ybar2 = mean(y2)
  yvar1 = var(y1)
  yvar2 = var(y2)

  # Sample p from the beta distribution
  p = rbeta(1, a + n1, b + n2)

  # generate a new theta value from its full conditional
  t2n1 = 1 / (1 / t20 + n1 / s21)
  mun1 = (mu0 / t20 + n1 * ybar1 / s21) / (1 / t20 + n1 / s21)
  theta1 = rnorm(1, mun1, sqrt(t2n1))

```

```

t2n2 = 1 / (1 / t20 + n2 / s22)
mun2 = (mu0 / t20 + n2 * ybar2 / s22) / (1 / t20 + n2 / s22)
theta2 = rnorm(1, mun2, sqrt(t2n2))

# # generate a new sigma^2 value from its full conditional

#Sampling s21 and s22 variance from the inverse gamma distribution
nun1 = nu0 + n1
s2n1 = (nu0 * s20 + (n1 - 1) * yvar1 + n1 * (ybar1 - theta1)^2) / nun1
s21 = 1 / rgamma(1, nun1 / 2, s2n1 * nun1 / 2)
nun2 = nu0 + n2
s2n2 = (nu0 * s20 + (n2 - 1) * yvar2 + n2 * (ybar2 - theta2)^2) / nun2
s22 = 1 / rgamma(1, nun2 / 2, s2n2 * nun2 / 2)

# Sample posterior predictive
xpred = runif(1) < p
ypred = ifelse(xpred, rnorm(1, theta1, sqrt(s21)), rnorm(1, theta2, sqrt(s22)))

# Storing values of theta1, theta2 and posterior predicted samples
THETA1[s] = theta1
THETA2[s] = theta2
YPRED[s] = ypred
}

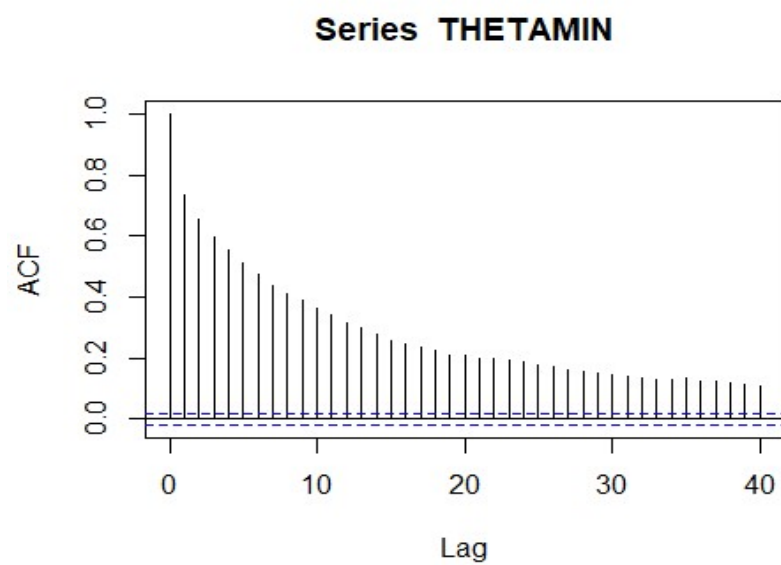
# Calculation of effective sample size for  $\theta(s)$  (1) =  $\min\{\theta(s)_1, \theta(s)_2\}$  and  $\theta(s)$  (2) =  $\max\{\theta(s)_1, \theta(s)_2\}$ .

THETAMIN = pmin(THETA1, THETA2)
THETAMAX = pmax(THETA1, THETA2)
library(coda)

## Warning: package 'coda' was built under R version 3.5.2

acf(THETAMIN, lag.max=40)

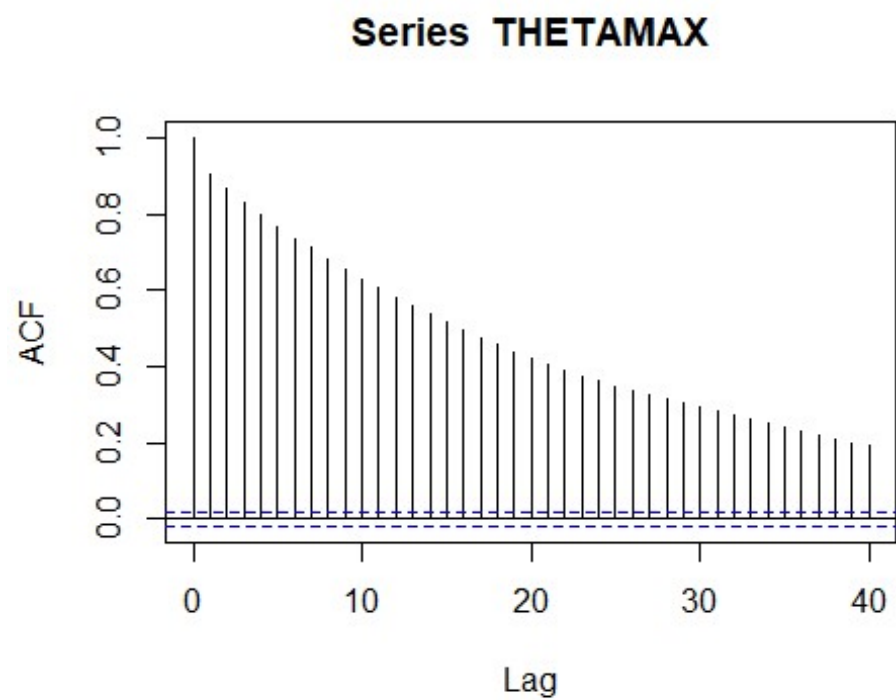
```



```
effectiveSize(THETAMIN)
```

```
## var1  
## 474.7763
```

```
acf(THETAMAX,lag.max=40)
```



```
effectiveSize(THETAMAX)
```



```
## var1  
## 205.6278
```

These samples are highly autocorrelated, hence the minimum effective sample sizes.

Part D: $Y(s) \sim \text{normal}(\theta(s) \times, \sigma^2(s) \times)$. Plot a histogram or kernel density estimate for the empirical distribution of $\sim Y(1), \dots, \sim Y(S)$, and compare to the distribution in part a). Discuss the adequacy of this two-component mixture model for the glucose data.

```
Y_sam = rbind(data.frame(y = YPRED, dataset = 'predictive'), data.frame(y = Y, dataset =  
'original'))  
ggplot(Y_sam, aes(x = y, fill = dataset)) +  
  geom_density(alpha = 0.5)
```

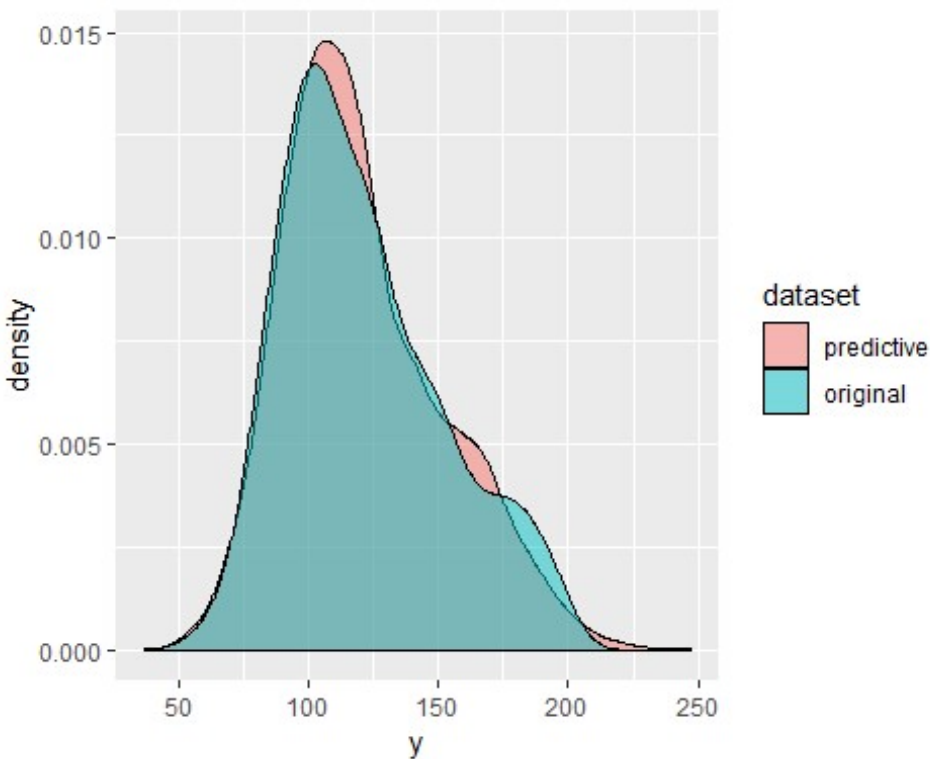


Fig: Kernel Density estimation plot for the empirical distribution of $\sim Y(1), \dots, \sim Y(S)$

Based on the kernel density estimation plot for the empirical distribution of data. We can observe very close resemblance between the densities of the original and posterior predictive dataset, we can infer that two-component mixture model for the glucose data fits very well.

7.3

```
bluecrab =
as.matrix(read.table('http://www.stat.washington.edu/people/pdhoff/Book/Data/hwdata/bluec
rab.dat'))
orangecrab =
as.matrix(read.table('http://www.stat.washington.edu/people/pdhoff/Book/Data/hwdata/oran
gecrab.dat'))
```

Part a: Obtain 10,000 posterior samples of θ and Σ

```
# Gibbs sampling
library(MASS)

# Created a function to obtain 10,000 posterior samples which takes input a matrix, this function is
applied to both the input matrices to get the posterior samples.
crab.list = lapply(list('bluecrab' = bluecrab, 'orangecrab' = orangecrab), function(Input_matrix) {
  p = ncol(Input_matrix)
  n = nrow(Input_matrix)
  ybar = colMeans(Input_matrix)

  # Using the semiconjugate prior distributions for  $\theta$  and  $\Sigma$ , set  $\mu_0$  equal to the sample mean of the data
# theta0 and S0 equal to the sample covariance matrix and nu0 = 4.
# Prior parameters
  mu0 = ybar
  L0 = S0 = cov(Input_matrix)
  nu0 = 4

  # Obtained 10,000 posterior samples of  $\theta$  and  $\Sigma$ 
  S = 10000
  THETA = matrix(nrow = S, ncol = p)
  SIGMA = array(dim = c(p, p, S))

  # Initializing the value of sigma
  sigma = S0

  for (i in 1:S) {
    # Update theta
    Ln = solve(solve(L0) + n * solve(S0))
    mun = Ln %*% (solve(L0) %*% mu0 + n * solve(S0) %*% ybar)
    theta = mvnrm(n = 1, mun, Ln)

    # Update sigma
    sn = S0 + (t(Input_matrix) - c(theta)) %*% t(t(Input_matrix) - c(theta))
    Sigma = solve(rWishart(1, nu0 + n, solve(sn))[, , 1])

    THETA[i, ] = theta
    SIGMA[, , i] = sigma
  }
}
```

```
list(theta = THETA, sigma = SIGMA)
})
```

Plot values of $\theta = (\theta_1, \theta_2)$ for each group and compare. Describe any size differences between the two groups

```
## #Here I used the above created function to create a data frame for the theta values of the bluecrab as well as for the orange crab
```

```
# Included the third column named 'species' for both the categories. Column species will help us to plot values of theta1 and theta2 of the respective species and compare between their sizes.
```

```
df_bluecrab = data.frame(crab.list$bluecrab$theta, species = 'blue')
```

```
df_orangecrab = data.frame(crab.mcmc$orangecrab$theta, species = 'orange')
```

```
colnames(df_bluecrab) = colnames(df_orangecrab) = c('theta1', 'theta2', 'species')
```

```
combined_df = rbind(df_bluecrab, df_orangecrab)
```

```
# Mean calculation is done for both the species in order to compare the size differences between the two groups.
```

```
bluecrab.means = as.data.frame(t(as.matrix(colMeans(df_bluecrab[, c('theta1', 'theta2')]))))
```

```
orangecrab.means = as.data.frame(t(as.matrix(colMeans(df_orangecrab[, c('theta1', 'theta2')]))))
```

```
bluecrab.means$species = 'blue'
```

```
orangecrab.means$species = 'orange'
```

```
crab.means = rbind(bluecrab.means, orangecrab.means)
```

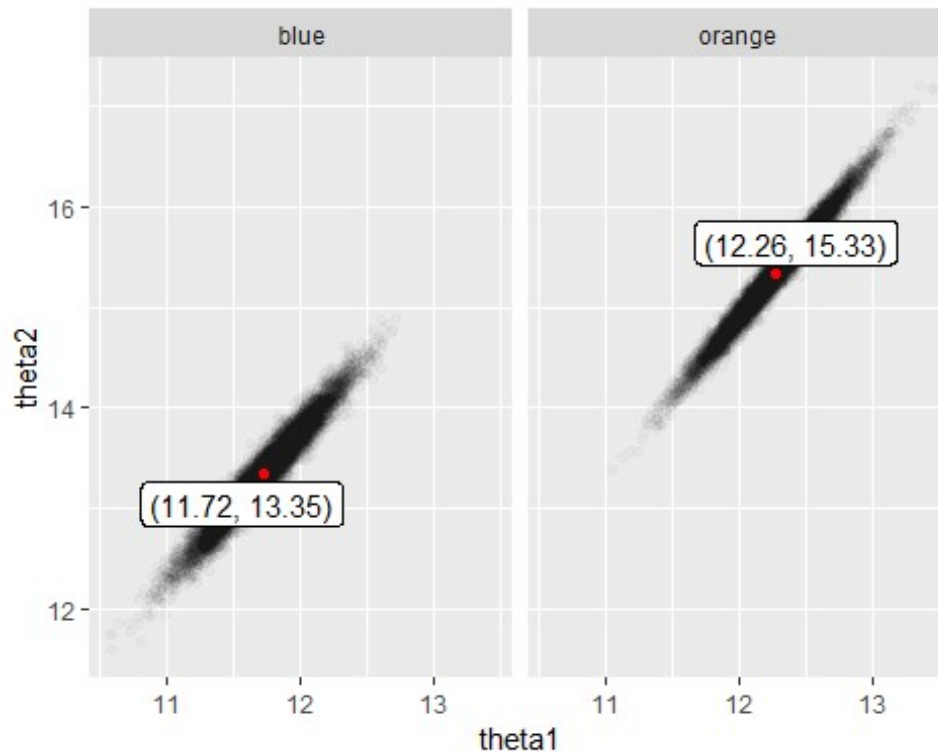
```
# Plotted the values of theta1 and theta2
```

```
library(ggplot2)
```

```
ggplot(combined_df, aes(x = theta1, y = theta2)) +geom_point(alpha = 0.02) +geom_point(data = crab.means, color = 'red') +
```

```
  geom_label_repel(data = crab.means, aes(label = paste0("(", round(theta1, 2), ", ", round(theta2, 2), ")"))) +facet_wrap(~ species)
```

```
## Loading required package: ggplot2
```



Calculation of probability of both the thetas being greater in case of orange crab compared to the bluecrab.

```
mean(df_orangecrab$theta1 > df_bluecrab$theta1)
```

```
## [1] 0.8983
```

```
mean(df_orangecrab$theta2 > df_bluecrab$theta2)
```

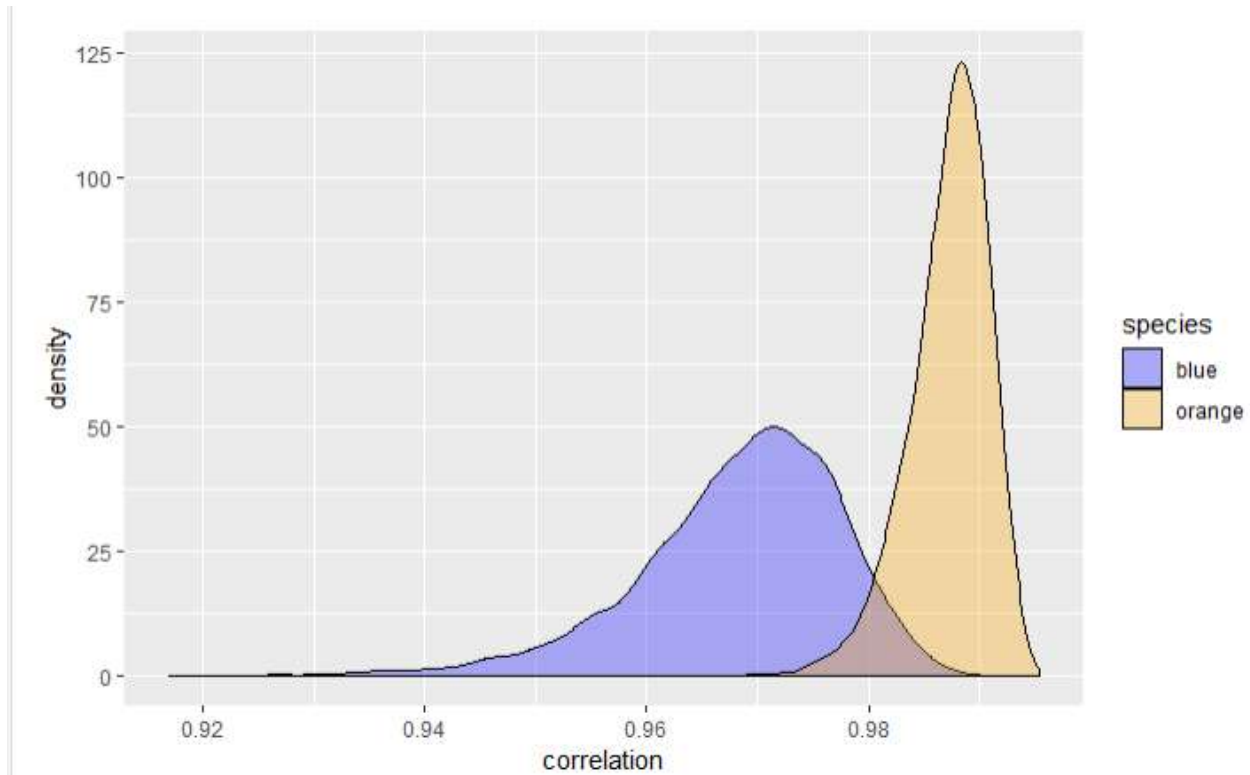
```
## [1] 0.998
```

From the above plot, we can observe that the orange crab is larger in both the measurements, i.e body depth(theta1) and rear width(theta2). From the plot, the values are (theta1 and theta2 for orange crab is 12.26,15.33 respectively and that of bluecrab is 11.72 and 13.35.) Moreover to confirm my opinion, I performed the calculation of probability of both the thetas being greater in case of orange crab compared to the bluecrab and found that the probabilities are higher and the results can be interpreted as I am about 90 percent sure that, theta1 for orange crab is greater than that of blue crab, similarly I am 99.8 percent sure that the theta2 of orange crab is greater than theta2 of bluecrab.

Part C: plot posterior densities of the correlations for the two groups.

```
bluecrab.cor = apply(crab.list$bluecrab$sigma, MARGIN = 3, FUN = function(covmat) {
  covmat[1, 2] / (sqrt(covmat[1, 1] * covmat[2, 2]))
})
orangecrab.cor = apply(crab.list$orangecrab$sigma, MARGIN = 3, FUN = function(covmat) {
  covmat[1, 2] / (sqrt(covmat[1, 1] * covmat[2, 2]))
})
cor.df = data.frame(species = c(rep('blue', length(bluecrab.cor)), rep('orange',
```

```
length(orangecrab.cor))),
      cor.relation= c(bluecrab.cor, orangecrab.cor))
ggplot(cor.df, aes(x = correlation, fill = species)) +
  geom_density(alpha = 0.3) +
  scale_fill_manual(values = c('blue', 'orange'))
```



```
mean(bluecrab.cor < orangecrab.cor)
```

```
## [1] 0.9897
```

The orange crab species appears to have a much higher correlation between its two measurements than the blue crab species. This statement is well supported by the probability of correlation of orange crab being higher than that of bluecrab.

7.4

```
agehw=as.matrix(read.table(url('http://www.stat.washington.edu/people/pdhoff/Book/Data/hwdata/agehw.dat')))
```

Part a # formulate a semi conjugate prior distribution for $\theta = (\theta_h, \theta_w)^T$ and Σ , where θ_h, θ_w are mean husband and wife ages, and Σ is the covariance matrix

With my prior knowledge, studies from some article and references, set $\mu_0 = ((25 + 85)/2, (25 + 85)/2)^T = (55, 55)^T$

From my class teachings, I think that it is normal distribution, it seems obvious that there are less married couples outside the range of ages 25 and 85 than there are married couples around age

50-55, thus justifying a normal curve around 55 with variance $15^2 \approx 225$ such that approximately 95% of the prior falls within the range (25,85).

It is obvious that the ages of the couples are quite tightly correlated, so knowing the above variance, I am assuming a prior correlation of 0.75, Solving the correlation equation gives

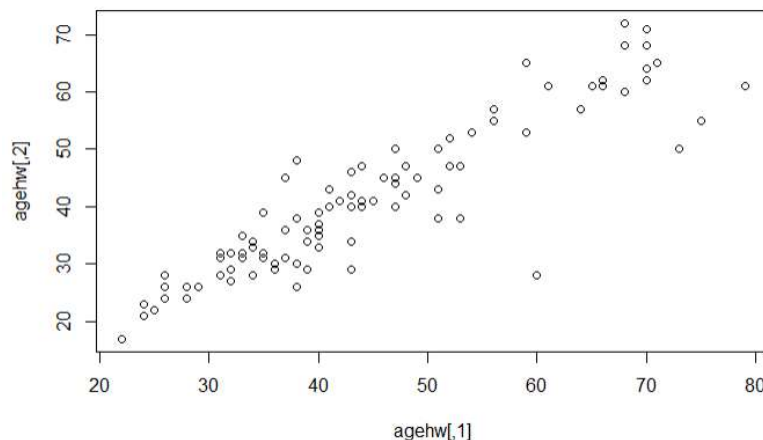
$$\Lambda_0 = \begin{bmatrix} 225 & 168.75 \\ 168.75 & 225 \end{bmatrix}$$

Like previous problems, for the variance, I will set $S_0^{-1} = \Lambda_0$ and $\nu_0 = p + 2 = 4$. Assuming, a vague prior belief that Σ is centered around Σ_0

```
agehw = matrix(as.numeric(agehw), nrow = 100)
```

```
Y = agehw
p = ncol(agehw)
n = nrow(agehw)
ybar = colMeans(agehw)
mu0 = rep(55, p)
L0 = s0 = rbind(c(225, 168.75), c(168.75, 225))
# nu0 = p + 2sssss
nu0 = p + 2 + 10
```

```
Plot(agehw)
```



Part b

Choosing larger ν_0 denotes stronger prior beliefs.

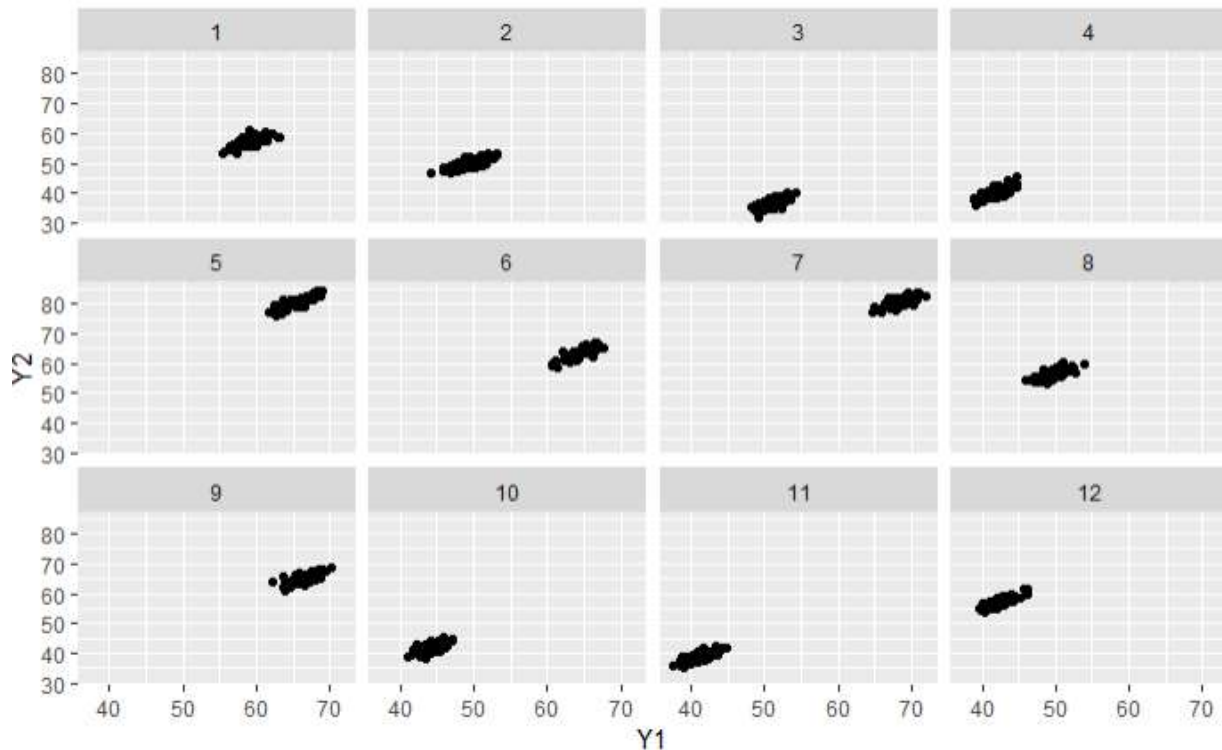
```
N = 100
S = 12
Y_preds = lapply(1:S, function(s) {
  # Sample THETA according to prior
  theta = mvrnorm(n = 1, mu0, L0)
  sigma = solve(rWishart(1, nu0 + n, solve(s0))[, , 1])
```

```

Y_s = mvrnorm(n = 100, theta, sigma)
data.frame(Y1 = Y_s[, 1], Y2 = Y_s[, 2], dataset = s)
})

# Referred from: https://www.stat.berkeley.edu/~s133/Docall.html
Y_comb = do.call(rbind, Y_preds)
ggplot(Y_comb, aes(x = Y1, y = Y2)) + geom_point() + facet_wrap(~ dataset)

```



Part c

```

S = 10000
# Creating a function same as in previous question 7.2
mcmc_approximation = function(Y, mu0, L0, s0, nu0) {
  ybar = colMeans(Y)
  p = ncol(Y)
  n = nrow(Y)
  THETA = matrix(nrow = S, ncol = p)
  SIGMA = array(dim = c(p, p, S))

  # Start with sigma sample
  sigma = cov(Y)

  # Gibbs sampling
  for (s in 1:S) {
    # Update theta
    Ln = solve(solve(L0) + n * solve(sigma))
    mun = Ln %*% (solve(L0) %*% mu0 + n * solve(sigma) %*% ybar)
    theta = mvrnorm(n = 1, mun, Ln)
  }
}

```

```

# Update sigma
sn = s0 + (t(Y) - c(theta)) %*% t((t(Y) - c(theta)))
sigma = solve(rWishart(1, nu0 + n, solve(sn)), , 1])

THETA[s, ] = theta
SIGMA[, s] = sigma
}

list(theta = THETA, sigma = SIGMA)
}
prior_mcmc = mcmc_approximation (agehw, mu0, L0, s0, nu0)
THETA = prior_mcmc$theta
SIGMA = prior_mcmc$sigma
# Husband-theta[,1], # Wife-theta[,2]
print("Husband")
print(quantile(THETA[, 1], probs = c(0.025, 0.5, 0.95, 0.975)))
print("Wife")
print(quantile(THETA[, 2], probs = c(0.025, 0.5, 0.95, 0.975)))
correlation= apply(SIGMA, 3, FUN = function(covmat) {
  covmat[1, 2] / (sqrt(covmat[1, 1] * covmat[2, 2]))

print("Correlation")

print(quantile(correlation, probs = c(0.95)))
print(quantile(correlation, probs = c(0.025, 0.5, 0.95, 0.975)))
}
print_quantiles(THETA, SIGMA)

plot(SIGMA[1,1,], xlab = 'Iteration',
ylab = expression(sigma[11]^2), type = 'l')

plot(SIGMA[2,2,], xlab = 'Iteration',
ylab = expression(sigma[22]^2), type = 'l')

plot(SIGMA[1,2,], xlab = 'Iteration',
ylab = expression(sigma[12]^2), type = 'l')

```

```

[1] "Husband"
      2.5%      50%      95%      97.5%
41.68113 44.44434 46.77111 47.21598
[1] "wife"
      2.5%      50%      95%      97.5%
38.29604 40.89117 43.08752 43.51033
[1] "Correlation"
      95%
0.8933223
      2.5%      50%      95%      97.5%
0.7933318 0.8558266 0.8933223 0.9000985

```

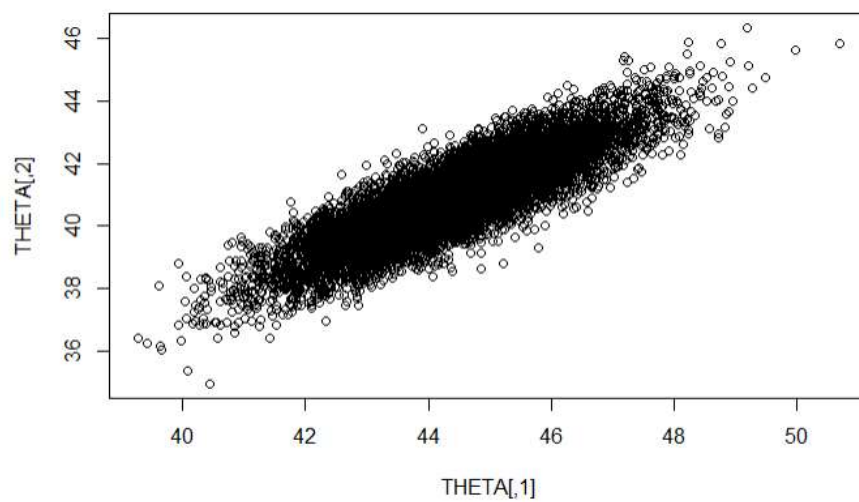
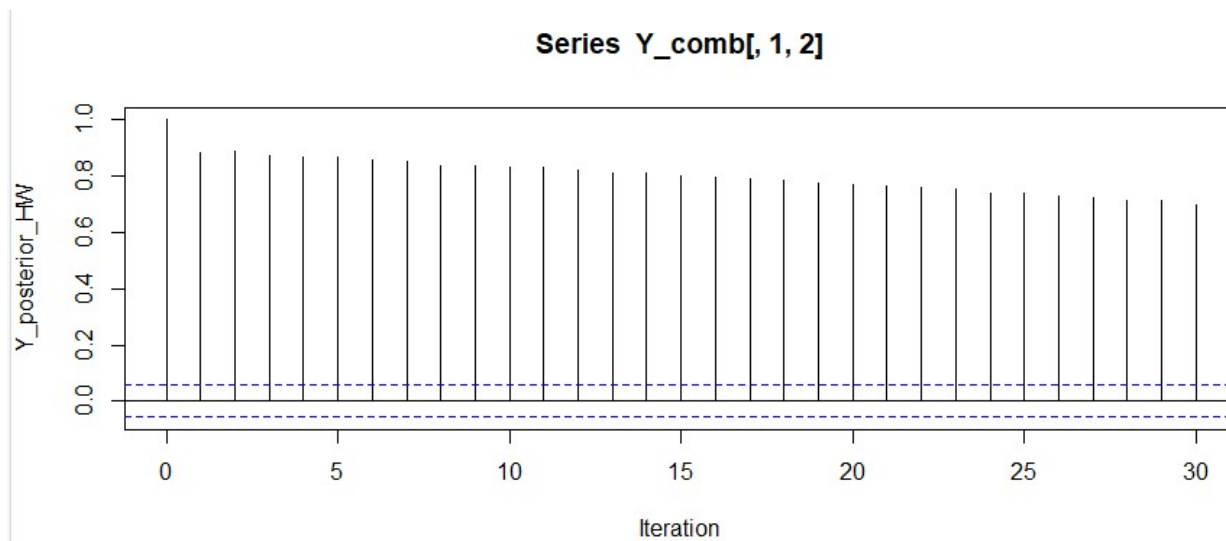
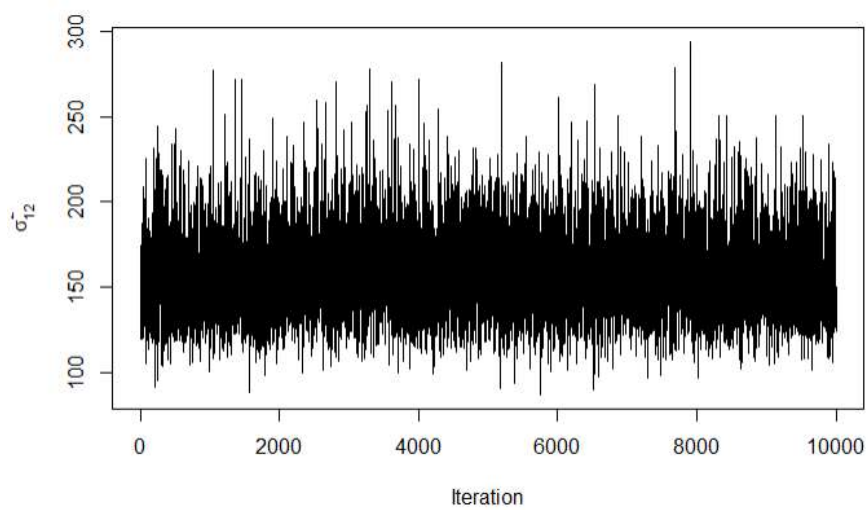
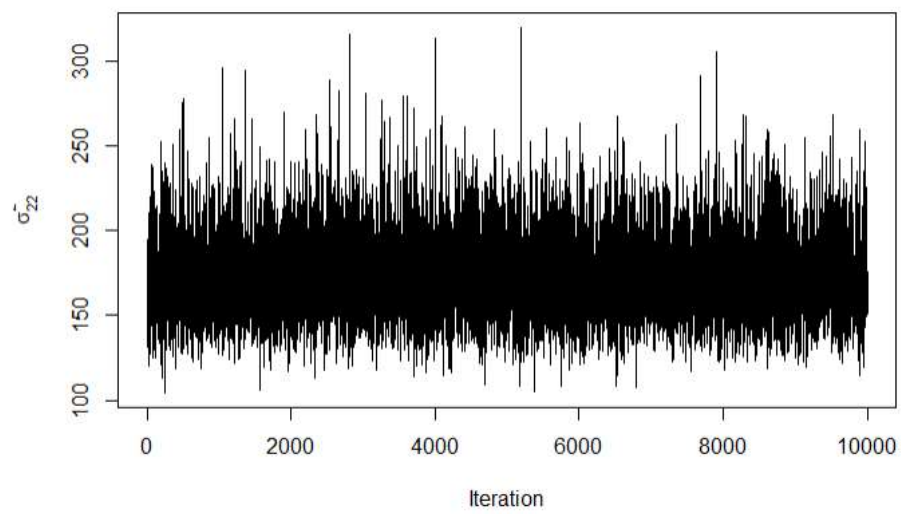
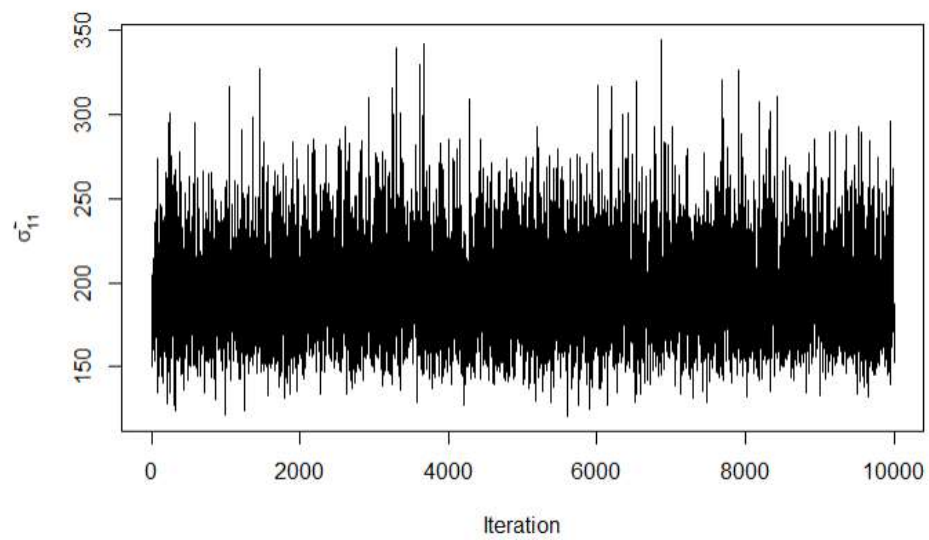



Fig: Plot of posterior distribution of θ_h and θ_w

R-Code: `acf(Y_comb[,1,2], xlab = 'Iteration', ylab = expression(Y_posterior_HW))`





\

Reference was taken from :

Class Notes- R code

Textbook-A First Course in Bayesian Statistical Methods

<https://www.tutorialspoint.com/r/index.htm>,

<https://www.tutorialspoint.com/r>

<http://www.r-tutor.com/> ,

http://rstudio-pubsstatic.s3.amazonaws.com/4952_7af63ededf804cbc9dbc85dda50d07e3.html

<https://github.com/jayelm/hoff-bayesian-statistics>

<https://stackoverflow.com/questions/2871763/>