

Normal model with conjugate priors

Case study: Nissan Maxima data-set

Georgios P. Karagiannis @ MATH3341/4031 Bayesian statistics III/IV (practical implementation)

[Back to README](#)

```
rm(list=ls())
```

Aim

Students will become able to:

- produce Monte Carlo approximations of posterior quantities required for Bayesian analysis with the RJAGS R package
- implement Bayesian posterior analysis in R with RJAGS package

Students are not required to learn by heart any of the concepts discussed

Reading material

The material about RJAGS package is not examinable material, but it is provided for the interested student. It contains references that students can follow if they want to further explore the concepts introduced.

- Lecture notes:
 - the examples related to the Bernoulli model with conjugate prior
- References for *RJAGS*:
 - JAGS homepage
 - JAGS R CRAN Repository
 - JAGS Reference Manual
 - JAGS user manual
- Reference for *R*:
 - Cheat sheet with basic commands
- Reference of *rmarkdown* (optional):
 - R Markdown cheatsheet
 - R Markdown Reference Guide
 - knitr options
- Reference for *Latex* (optional):
 - Latex Cheat Sheet

New software

- R package `rjags` functions:

```
- jags.model{rjags}
- jags.samples{rjags}
- update{rjags}
```

Application:

The applications below is ‘reproduced’ from DASL, and

- De Veaux, R. D., Velleman, P. F., Bock, D. E., Vukov, A. M., Augustine, C. W., & Burkett, C. (2005). Stats: data and models. Boston: Pearson/Addison Wesley.

Richard DeVeaux owned a Nissan Maxima for 8 years. Being a statistician, He recorded the car’s fuel efficiency (in mpg) each time he filled the tank. He wanted to know what fuel efficiency to expect as “ordinary” for his car. Knowing this, he was able to predict when he’d need to fill the tank again and to notice if the fuel efficiency suddenly got worse, which could be a sign of trouble.

General scientific question:

```
# load the data
#mydata <- read.csv("./nissan.csv")
mydata <- read.csv("https://raw.githubusercontent.com/georgios-stats/Bayesian_Statistics/master/Computer%20lab%201/nissan.csv")
#mydata$mpg <- rnorm(100, 20, 5)
# print data
cat(mydata$mpg)
```

```
## 21.964 23.694 18.824 20.851 26.37 22.81 25.785 24.353 23.385 23.381 28.175 21.232 25.603 21.064 22.064 21.064 22.064 22.064 22.064 22.064
```

Preliminary analysis

Task

Examine graphically whether there is any substantial evidence that the above data-set is not generated from the Normal distribution.

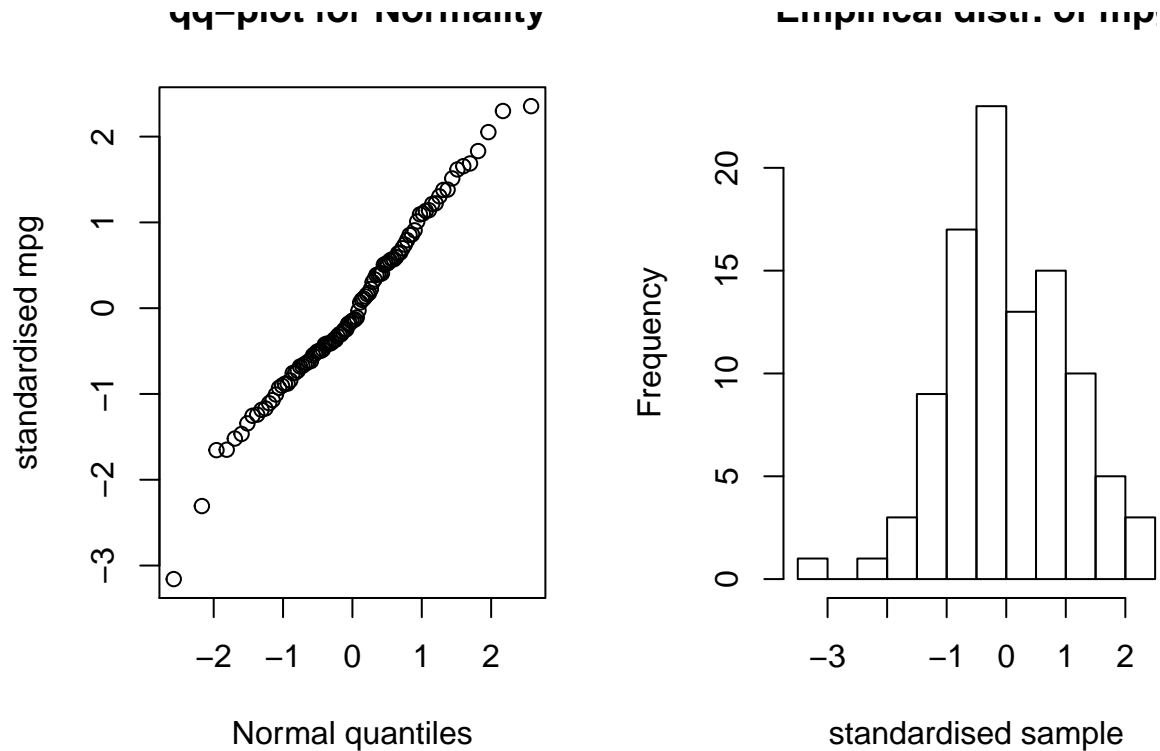
Are there any substantiaintial evivence against Normality?

... your answer

We can assume that the data-set generating prodedure can be represented by a Normal distribution family. This is based on the preliminary examination below.

```
# Preliminary examination
## standardise me
z_obs <- (mydata$mpg - mean(mydata$mpg)) / sd(mydata$mpg)
## plot me
par(mfrow=c(1,2))
qqnorm(z_obs,
  main = "qq-plot for Normality",
  ylab = "standardised mpg",
  xlab = "Normal quantiles"
)
```

```
hist(z_obs,
     main = "Empirical distr. of mpg",
     xlab = "standardised sample"
)
```



Model

We specify the Normal distribution with uncertain mean and variance parameters μ and σ^2 as a parametric model.

We assign a Normal-Inverse Gamma conjugate prior with fixed hyper-parameters $\mu_0 = 0.0$, $\lambda_0 = 1.0$, $a_0 = 1.0$, and $b_0 = 1.0$.

The Bayesian hierarchical model under consideration is:

$$\begin{cases} x_i | \mu, \sigma^2 & \stackrel{\text{iid}}{\sim} N(\mu, \sigma^2), \forall i = 1, \dots, n \\ \mu | \sigma^2 & \sim N(\mu_0, \frac{\sigma^2}{\lambda_0}) \\ \sigma^2 & \sim G(a, b) \end{cases}$$

for some fixed values of the hyper-parameters λ_0 , a , and b .

Here, we just set:

- $\mu_0 = 0.0$,
- $\lambda_0 = 1.0$,
- $a_0 = 0.01$,

- $b_0 = 0.01$

Task

1. Write the rjags program to draw a sample from the posterior distribution of the parameters μ , and σ^2
2. Generate a sample of size $N = 1000$ from the posterior distribution of the parameters.
3. Draw trace plots of the samples desired

... your answer

Load the library

```
# Load rjags
library("rjags")
```

Create an input script, for rjags, containing the Bayesian hierarchical model

1. Write the Stan program to draw a sample from the posterior distribution of the parameters μ , and σ^2
2. Generate a sample of size $N = 1000$ from the posterior distribution of the parameters.

```
hierarhicalmodel <- "

model {

  # this is related tot he sampling distribution

  tau_0 = lam_0 / sigma2

  for ( i in 1 : n ) {
    y[ i ] ~ dnorm( mu, tau )
  }

  # this is related to the prior distributions

  mu ~ dnorm( mu_0 , tau*lam_0 )
  sigma2 = 1 / tau
  tau ~ dgamma(a_0 , b_0)

}

"
```

Create and list with inputs for rstan

```
a_0 <- 0.01
b_0 <- 0.01
lam_0 <- 1.0
mu_0 <- 0.0
y_obs <- mydata$mpg ;
n_obs <- length( y_obs ) ;
data.bayes <- list(n = n_obs ,      # the number of the observations
                  y = y_obs ,      # the observations
                  a_0 = a_0,
```

```

b_0 = b_0,
lam_0 = lam_0,
mu_0 = mu_0
)

```

Create an input list, for jags, containing the data and fixed parameters of the model

```

model.smpl <- jags.model( file = textConnection(hierarhicalmodel),
                        data = data.bayes)

```

Initialize the sampler with $N_{\text{adapt}} = 1000$ iterations.

- This is a warming-up procedure (used as a black-box), where the sampler is automatically tuned and calibrated before it starts generation your samples.
- Regarding $N_{\text{adapt}} = 1000$, the larger the better.

```

adapt( object = model.smpl,
       n.iter = 1000 )

```

```
## [1] TRUE
```

Generate a posterior sample of size $N = 10000$.

Use

- `jags.samples{rjags}`

We need to pay attention on two flags:

- the `n.iter`: the total size of the total sample sequence.
- the `variable.names`: it specifies the names of the random variables correspondign to the posterior samples I am interested in generating

```

N = 10000      # the size of the sample we ll gonna get
output = jags.samples( model= model.smpl,          # the model
                      variable.names= c("mu","sigma2"), # names of variables to be sampled
                      n.iter = N                # size of sample
                      )

```

Check the names of the variables sampled

- use `names {base}`

```
names(output)
```

```
## [1] "mu"      "sigma2"
```

Check the dimensions of each of the variables sampled

- use `dim {base}`

```
dim( output$mu )
```

```
##          iteration      chain
##          1      10000         1
```

```
dim( output$sigma2 )
```

```
##          iteration      chain
##          1      10000         1
```

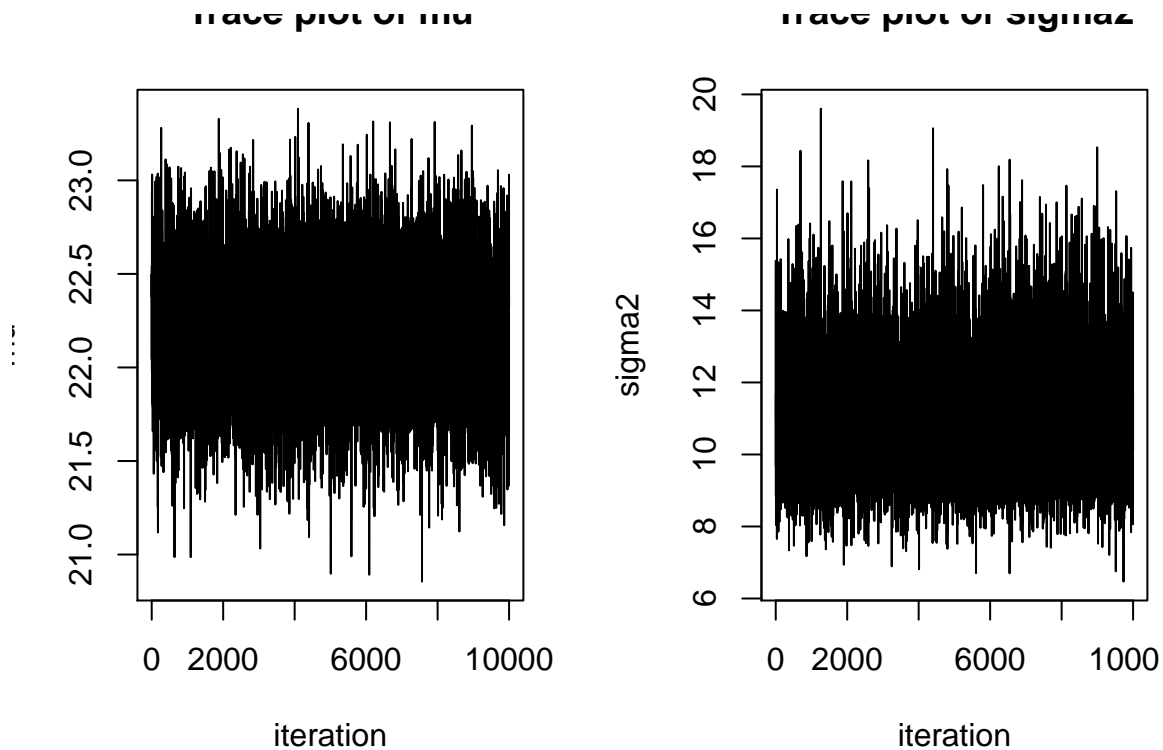
```
# the first dimension is the numbers of columns of the variable  
# the second dimension is the size of the sample drawn  
# the thirs dimention is the number of the sub-samples drawn (in our case it is just 1)
```

Copy the sample of each variable in a vector with a more friendly name...

```
mu.smpl <-output$mu  
sigma2.smpl <-output$sigma2
```

We plot the trace plots of the generated samples

```
# draw the trace plots  
  
par(mfrow=c(1,2))  
  
mu.smpl <-output$mu  
  
plot(mu.smpl,  
     type = "l",  
     main = "Trace plot of mu",  
     xlab = "iteration",  
     ylab = "mu"  
)  
  
sigma2.smpl <-output$sigma2  
  
plot(sigma2.smpl,  
     type = "l",  
     main = "Trace plot of sigma2",  
     xlab = "iteration",  
     ylab = "sigma2"  
)
```



Task

Produce and plot the exact joint PDF of (μ, σ^2) as a 2D plot, and as a 2D contour plot

To draw 3D plots like those required to plot the exact PDF you can use the functions `persp{graphics}`, or `contour{graphics}` as `persp(x, y, z)` or `contour(x, y, z)`, where x, y are locations of grid lines at which the values in z are measured; and z is a matrix containing the values to be plotted ;

Usage

% To plot $z=f(x,y)$ for a specific function $f(.,.)$ at vectors x and y , then
 % for known x, y vectors of length 100

```
z <- matrix(rep(NaN,100*100),100,100)
for (i in 1:100)
  for (j in 1:100)
    z[i,j] <-f( x[i],z[j] )
```

```
persp(x, y, z)
```

% or

```
contour(x, y, z)
```

Hint-1

The joint posterior distribution (μ, σ^2) , is:

$$\begin{cases} \mu | x_{1:n}, \sigma^2 & \sim N(\mu_n, \frac{\sigma^2}{\lambda_n}) \\ \sigma^2 | x_{1:n} & \sim \text{IG}(a_n, b_n) \end{cases}$$

where

$$\begin{aligned}\mu_n &= \frac{\lambda_0 \mu_0 + n \bar{x}}{\lambda_0 + n} \\ \lambda_n &= \lambda_0 + n, \\ a_n &= a_0 + \frac{n}{2} \\ b_n &= b_0 + \frac{1}{2} n s^2 + \frac{1}{2} (\lambda_0 + n)^{-1} \lambda_0 n (\mu_0 - \bar{x})^2\end{aligned}$$

Hence the PDF is

$$\pi(\mu, \sigma^2 | x_{1:n}) = N(\mu | \mu_n, \frac{\sigma^2}{\lambda_n}) \text{IG}(\sigma^2 | a_n, b_n)$$

Hint:

Additional functions for PDFs and CDFs are given

```
# Inverse Gamma PDF
invgamma_pdf <- function (x,a,b) {
  return(exp(a*log(b)-lgamma(a)-(a+1)*log(x)-b/x))
}
# Inverse Gamma CDF
invgamma_cdf <- function (x,a,b) {
  return(1.0-pgamma(1.0/x,shape=a,scale=1.0/b) )
}
```

... your answer

```
library(gplots)

par(mfrow=c(1,2))

# EXACT

## compute the posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)

## plot
xmin = 21
xmax = 23
x_plot <- seq( from = xmin,
               to = xmax,
               length.out = 100)

ymin = 7
ymax = 19
y_plot <- seq( from = ymin,
               to = ymax,
               length.out = 100)
```

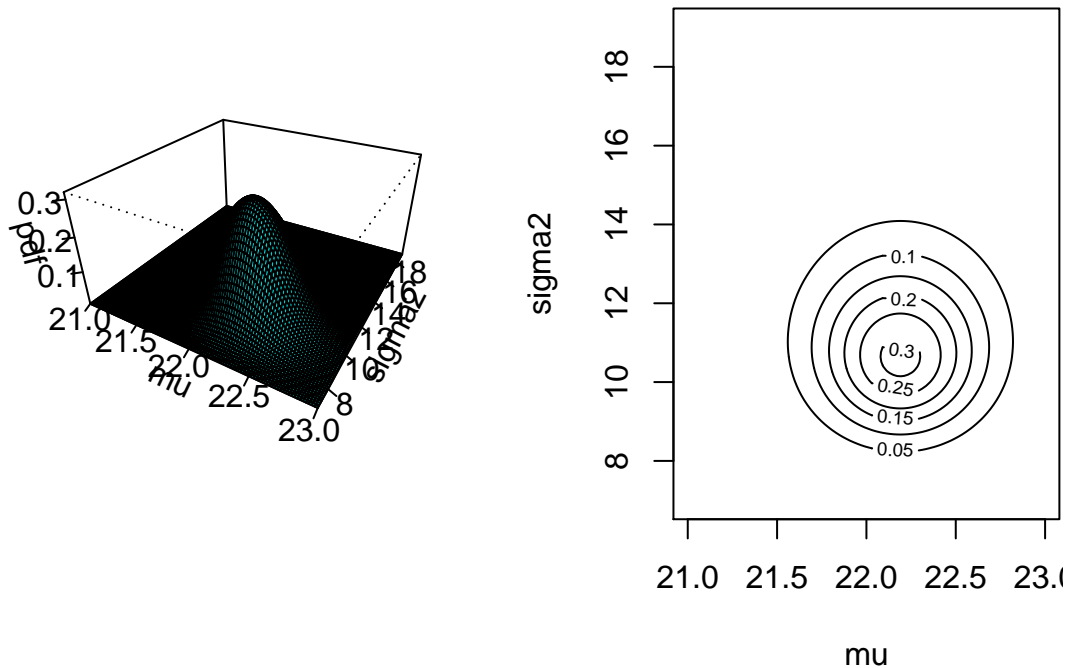


```

z_plot <- matrix(rep(NA,100*100),100,100)
for (i in 1:100)
  for (j in 1:100)
    z_plot[i,j] <- dnorm( x_plot[i], mu_n, sqrt(y_plot[j]/lam_n) ) * invgamma_pdf(y_plot[j], a_n, b_n)

## plot
persp(x_plot, y_plot, z_plot,
      ticktype="detailed", theta=30, phi=30,
      expand=0.5, shade=0.5, col="cyan", ltheta=-30,
      xlab = 'mu',
      ylab='sigma2',
      zlab='pdf')
## plot
contour(x_plot, y_plot, z_plot,
        xlab = 'mu',
        ylab='sigma2')

```



Task

Produce and plot the MC approximation of the joint PDF of (μ, σ^2) , as a 2D histogram, and as a 2D contour plot.

To draw histograms of 2D samples like required by the MC approximation of the PDF you can use the functions `hist2d{gplots}` and `persp{graphics}`, or `contour{graphics}` as:

% known vectors x and y of length 100 having the 2D samples:

```
h2d <- hist2d(x,y,nbins=c(20,30),show=FALSE)
```

```
persp( h2d$x, h2d$y, h2d$counts, ticktype="detailed", theta=30, phi=30,expand=0.5, shade=0.5, col="cyan"
```

```
contour( h2d$x, h2d$y, h2d$counts, ticktype="detailed", theta=30, phi=30,expand=0.5, shade=0.5, col="cy
```

... your answer

```
library(gplots)
```

```
mu.smpl <-output$mu  
sigma2.smpl <-output$sigma2
```

```
# MC APPROXIMATION
```

```
par(mfrow=c(1,2))
```

```
## extract the sample
```

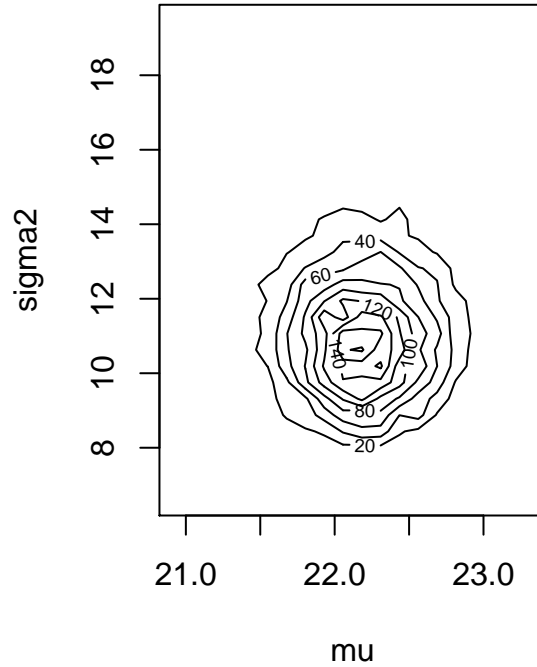
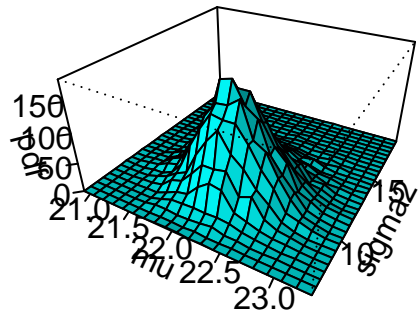
```
h2d <- hist2d(mu.smpl,  
             sigma2.smpl,  
             nbins=c(20,30),  
             show=FALSE)
```

```
## plot
```

```
persp(h2d$x, h2d$y, h2d$counts,  
      ticktype="detailed", theta=30, phi=30,  
      expand=0.5, shade=0.5, col="cyan", ltheta=-30,  
      xlab = 'mu',  
      ylab='sigma2',  
      zlab='pdf')
```

```
## plot
```

```
contour(h2d$x, h2d$y, h2d$counts,  
       xlab = 'mu',  
       ylab='sigma2')
```



Task

Produce and plot the Exact marginal PDF of μ .

Produce and plot the MC approximation of the marginal PDF of μ .

You can use the functions `hist` {graphics} and `lines` {graphics} or `plots` {graphics}

Hint

The marginal distribution of μ , is:

$$\mu|x_{1:n} \sim \text{St}_k(\mu_n, \frac{b_n}{\lambda_n a_n}, 2a_n)$$

where

$$\mu_n = \frac{\lambda_0 \mu_0 + n \bar{x}}{\lambda_0 + n}$$

$$\lambda_n = \lambda_0 + n,$$

$$a_n = a_0 + \frac{n}{2}$$

$$b_n = b_0 + \frac{1}{2} n s^2 + \frac{1}{2} (\lambda_0 + n)^{-1} \lambda_0 n (\mu_0 - \bar{x})^2$$

Hint:

Additional PDFs and CDFs are given

```
# Student T PDF
studentT_pdf <- function(x,m,s,v) {
```

```

    return(dt((x-m)/sqrt(s),df=v)/sqrt(s))
}
# Student T CDF
studentT_cdf <- function(x,m,s,v) {
  return(pt((x-m)/sqrt(s),df=v))
}

```

... your answer

For the marginal posterior PDF of μ

```

# Draw the histogram as the MC approximate of the PDF

mu.smpl <- output$mu

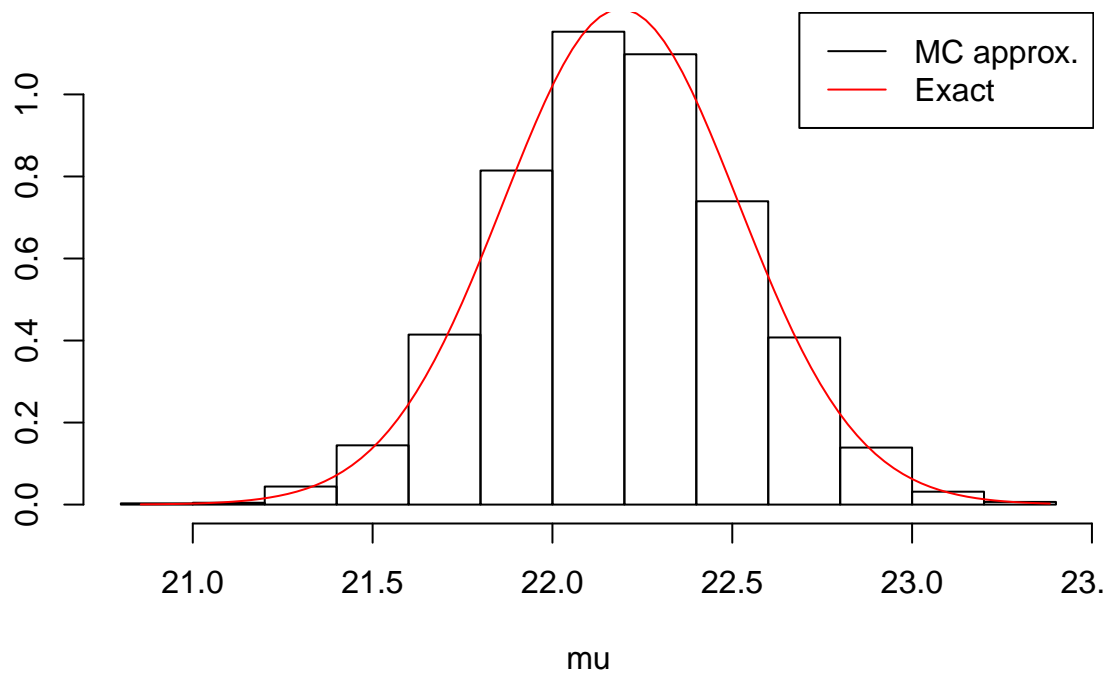
z <- mu.smpl
hist(z,
     probability = TRUE,
     main = "Post PDF of mu",
     xlab = "mu",
     ylab = "PDF")

# Draw the exact PDF

## Posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0 - mean(y_obs))^2/(lam_0 + n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0 + n_obs*mean(y_obs)) / (lam_0 + n_obs)

x_plot <- seq( from = min(z), to = max(z), length.out = 100)
y_plot <- studentT_pdf(x_plot, mu_n, b_n/(a_n*lam_n), 2*a_n)
lines(x_plot,
      y_plot,
      type = 'l',
      col = 'red')
legend("topright",
      legend=c("MC approx.", "Exact"),
      lty = c(1,1),
      col=c("black", "red"))

```



For the marginal posterior CDF of μ

```
# Draw the histogram as the MC approximate of the CDF

## extract the sample, and copy it to smple
mu.smpl <- output$mu

z <- mu.smpl
x_plot <- seq( from = mean(z)-4*sd(z), to = mean(z)+4*sd(z), length.out = 100)
y_plot <- rep(NaN, 100)
for (i in 1:100) y_plot[i] <- mean(z<=x_plot[i])
plot(x_plot,
     y_plot,
     type = "l",
     main = "CDF of mu",
     xlab = "mu",
     ylab = "CDF")

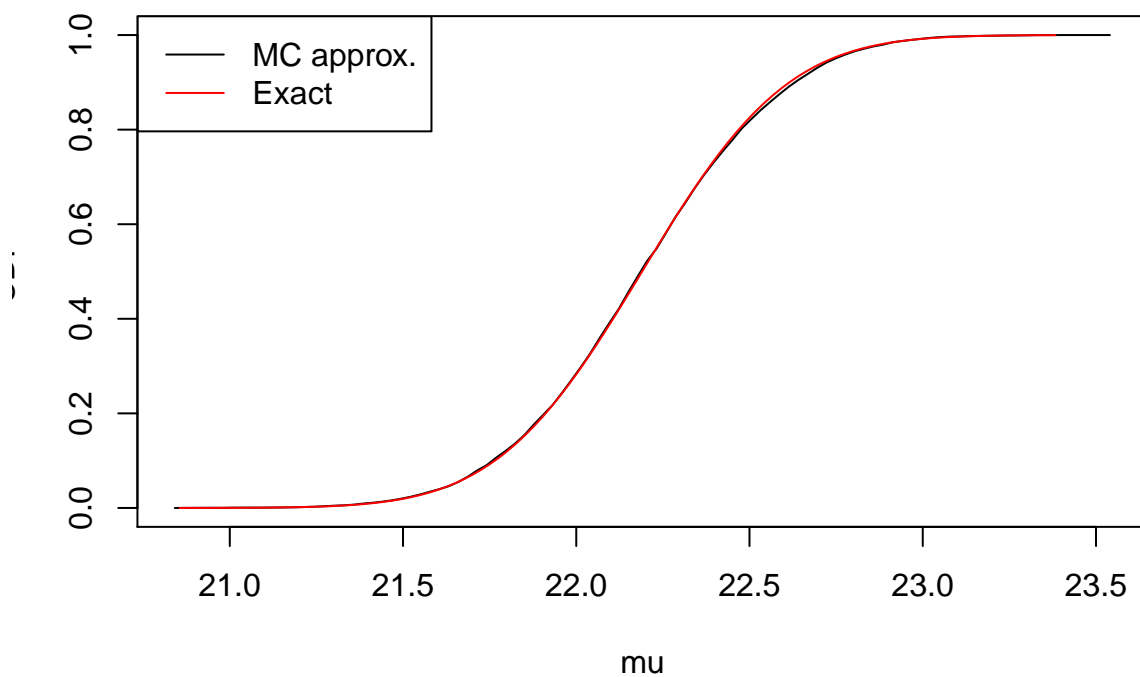
# Draw the exact CDF

## Posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)
x_plot <- seq( from = min(z), to = max(z), length.out = 100)
y_plot <- studentT_cdf(x_plot , mu_n , b_n/(a_n*lam_n), 2*a_n)
lines(x_plot,
     y_plot,
     type = 'l',
     col = 'red')
```

```

legend("topleft",
      legend=c("MC approx.", "Exact"),
      lty = c(1,1),
      col=c("black", "red"))

```



Task

Produce and plot the Exact marginal PDF of σ^2 .

Produce and plot the MC approximation of the marginal PDF of σ^2 .

You can use the functions `hist {graphics}` and `lines {graphics}` or `plot {graphics}`

Hint

The marginal distributions of σ^2 , as:

$$\sigma^2|x_{1:n} \sim \text{IG}(\sigma^2|a_n, b_n)$$

where

$$\mu_n = \frac{\lambda_0 \mu_0 + n \bar{x}}{\lambda_0 + n}$$

$$\lambda_n = \lambda_0 + n,$$

$$a_n = a_0 + \frac{n}{2}$$

$$b_n = b_0 + \frac{1}{2}ns^2 + \frac{1}{2}(\lambda_0 + n)^{-1}\lambda_0 n(\mu_0 - \bar{x})^2$$

Hint:

Additional PDFs and CDFs are given

```
# Inverse Gamma PDF
invgamma_pdf <- function (x,a,b) {
  return(exp(a*log(b)-lgamma(a)-(a+1)*log(x)-b/x))
}

# Inverse Gamma CDF
invgamma_cdf <- function (x,a,b) {
  return(1.0-pgamma(1.0/x,shape=a,scale=1.0/b) )
}
```

... your answer

For the marginal posterior PDF of σ^2

```
# Draw the histogram as the MC approximate of the PDF

## extract the sample, and copy it to smple
sigma2.smpl <-output$sigma2

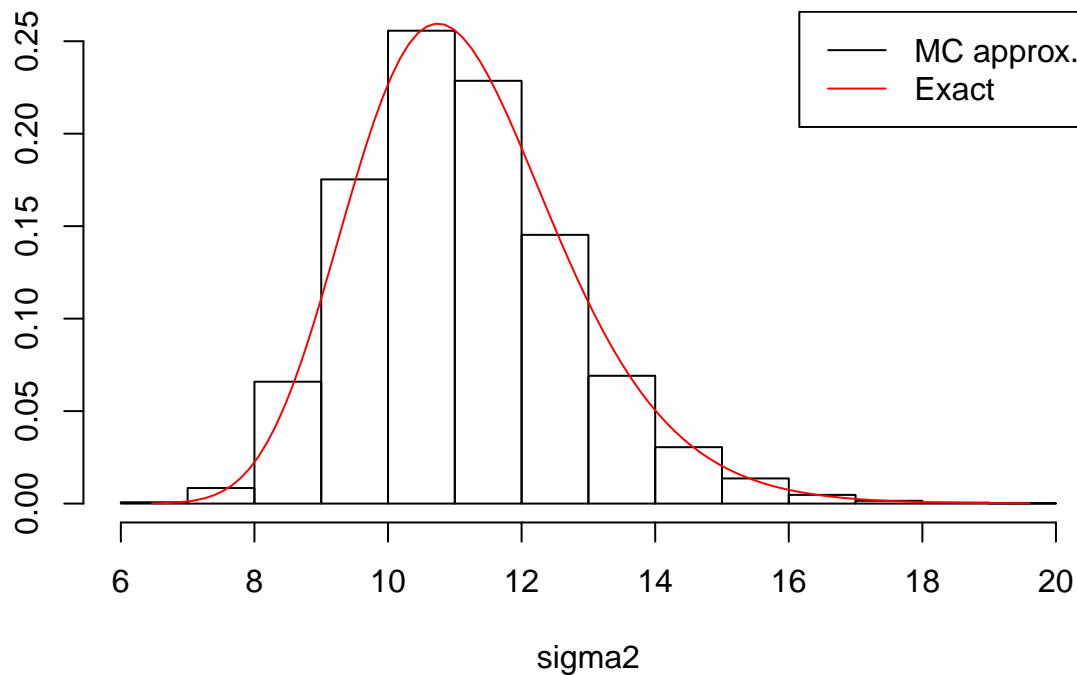
z <- sigma2.smpl
hist(z,
     probability = TRUE,
     main = "Post PDF of sigma2",
     xlab = "sigma2",
     ylab = "PDF")

# Draw the exact PDF

## Posterior parameters
a_n <- a_0 +0.5*n_obs
b_n <- b_0 +0.5*n_obs*var(y_obs) +0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)

x_plot <- seq( from = min(z), to = max(z), length.out = 100)
y_plot <- invgamma_pdf(x_plot , a_n, b_n)
lines(x_plot,
     y_plot,
     type = 'l',
     col = 'red')
legend("topright",
     legend=c("MC approx.", "Exact"),
     lty = c(1,1),
     col=c("black", "red"))
```

Posterior of sigma2



For the marginal posterior CDF of σ^2

```
# Draw the histogram as the MC approximate of the CDF
```

```
## extract the sample, and copy it to smple  
sigma2.smpl <- output$sigma2
```

```
z <- sigma2.smpl  
x_plot <- seq( from = mean(z)-4*sd(z), to = mean(z)+4*sd(z), length.out = 100)  
y_plot <- rep(NaN, 100)  
for (i in 1:100) y_plot[i] <- mean(z<=x_plot[i])  
plot(x_plot,  
     y_plot,  
     type = "l",  
     main = "CDF of sigma2",  
     xlab = "sigma2",  
     ylab = "CDF")
```

```
# Draw the exact CDF
```

```
## Posterior parameters
```

```
a_n <- a_0 + 0.5*n_obs  
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)  
lam_n <- lam_0 + n_obs  
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)
```

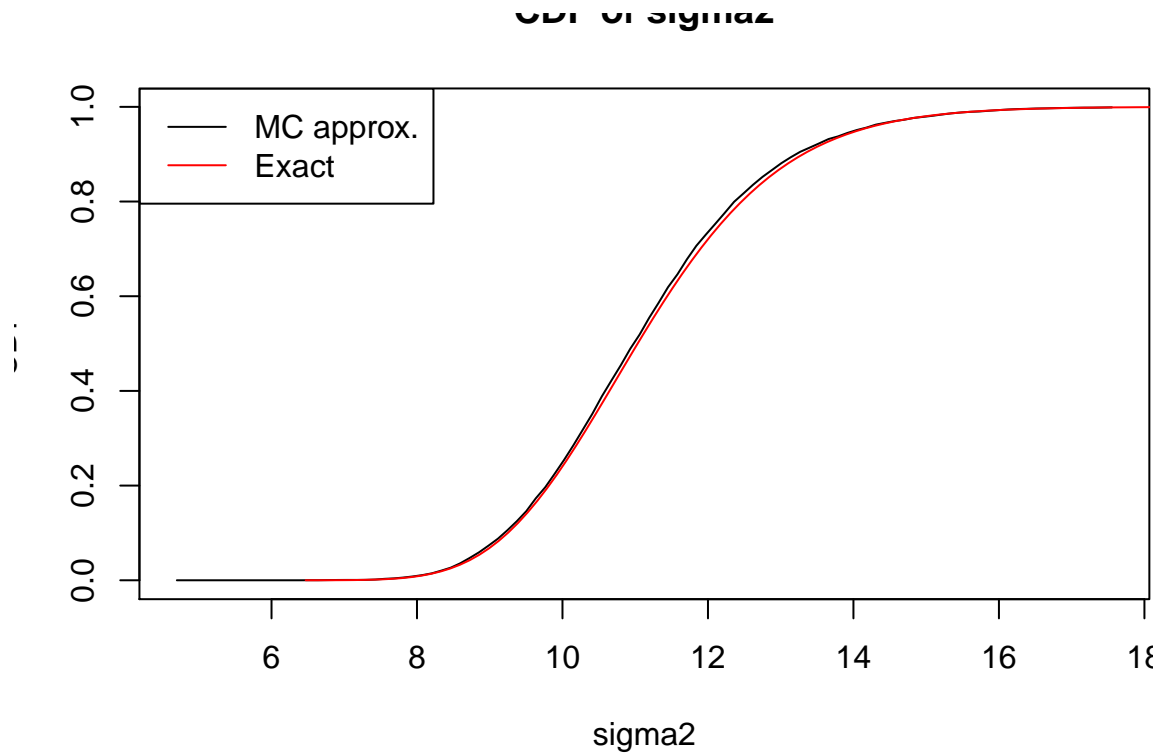
```
x_plot <- seq( from = min(z), to = max(z), length.out = 100)  
y_plot <- invgamma_cdf(x_plot , a_n, b_n)  
lines(x_plot,  
      y_plot,
```



```

type = 'l',
col = 'red')
legend("topleft",
      legend=c("MC approx.", "Exact"),
      lty = c(1,1),
      col=c("black", "red"))

```



Task

- Compute the MC approximate of the posterior mean of μ , and σ^2 :

$$E_{\pi}(\mu|y_{1:n}) \approx \frac{1}{N} \sum_{j=1}^N \mu^{(j)}$$

and

$$E_{\pi}(\sigma^2|y_{1:n}) \approx \frac{1}{N} \sum_{j=1}^N \left(\sigma^{(j)}\right)^2$$

- Compute their exact values which are

$$E_{\pi}(\mu|y_{1:n}) = \mu_n$$

and

$$E_{\pi}(\sigma^2|y_{1:n}) = \frac{b_n}{a_n - 1}$$

... your answer

The Monte Carlo approximation for $E_\pi(\mu|y_{1:n})$ is

```
mu.smpl <-output$mu
mc.approx <- mean(mu.smpl)
mc.approx
```

```
## [1] 22.19088
```

The exact value for $E_\pi(\mu|y_{1:n}) = \mu_n$ is

```
# Posterior parameters
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)
mu_n
```

```
## [1] 22.19009
```

The Monte Carlo approximation for $E_\pi(\sigma^2|y_{1:n})$ is

```
sigma2.smpl <-output$sigma2
mc.approx <- mean(sigma2.smpl)
mc.approx
```

```
## [1] 11.12615
```

The exact value for $E_\pi(\sigma^2|y_{1:n}) = \frac{b_n}{a_n-1}$ is

```
# Posterior parameters
a_n <- a_0 +0.5*n_obs
b_n <- b_0 +0.5*n_obs*var(y_obs) +0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
exct <- b_n/(a_n-1)
exct
```

```
## [1] 11.18335
```

Task

- Compute the MC approximate of the posterior probability that the mean fuel efficiency (in mpg) of car is greater or equal to 22.5 mpg.

$$\Pr_\pi(\mu \geq 22.5|y_{1:n}) = 1 - \Pr_\pi(\mu < 22.5|y_{1:n}) \quad (1)$$

$$= 1 - E_\pi(1(\mu < 22.5)|y_{1:n}) \quad (2)$$

$$\approx 1 - \frac{1}{N} \sum_{j=1}^N 1(\mu^{(j)} < 22.5) \quad (3)$$

- Compute the exact value which is

$$\Pr_\pi(\mu \geq 22.5|y_{1:n}) = 1 - \Pr_{\text{St}_k(\mu_n, \frac{b_n}{\lambda_n a_n}, 2a_n)}(\mu \leq 22.5|y_{1:n}) \quad (4)$$

$$= 1 - \int_{-\infty}^{22.5} \text{St}_k(\mu|\mu_n, \frac{b_n}{\lambda_n a_n}, 2a_n) d\mu \quad (5)$$

Hint

Functions for the PDFs and CDF are given

```

# Student T PDF
studentT_pdf <- function(x,m,s,v) {
  return(dt((x-m)/sqrt(s),df=v)/sqrt(s))
}
# Student T CDF
studentT_cdf <- function(x,m,s,v) {
  return(pt((x-m)/sqrt(s),df=v))
}

```

... your answer

The MC approximate is

```

## extract the sample, and copy it to smpl
mu.smpl <- output$mu
Pr.mu.mc <- mean( mu.smpl >= 22.5 )
Pr.mu.mc

```

```
## [1] 0.1814
```

and its exact value is ...

```

## Posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0 - mean(y_obs))^2/(lam_0 + n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0 + n_obs*mean(y_obs)) / (lam_0 + n_obs)
## The Exact value is
Pr.mu.ex <- 1 - studentT_cdf(22.5, mu_n, b_n/(lam_n*a_n), 2*a_n)
Pr.mu.ex

```

```
## [1] 0.1745368
```

The MC approximation is good!

Task

Compute the exact 95% equal tail posterior credible interval of μ .

Hint

The marginal distribution of μ , is:

$$\mu|x_{1:n} \sim \text{St}_k\left(\mu_n, \frac{b_n}{\lambda_n a_n}, 2a_n\right)$$

where

$$\begin{aligned}\mu_n &= \frac{\lambda_0 \mu_n + n \bar{x}}{\lambda_0 + n} \\ \lambda_n &= \lambda_0 + n, \\ a_n &= a_0 + \frac{n}{2} \\ b_n &= b_0 + \frac{1}{2} n s^2 + \frac{1}{2} (\lambda_0 + n)^{-1} \lambda_0 n (\mu_0 - \bar{x})^2\end{aligned}$$

Compute the MC approximation of the 95% equal tail posterior credible interval of μ .

Hint:

Additional PDFs and CDFs are given

```
# Student T INVERSE CDF
studentT_inv <- function(prob,m,s,v) {
  q = qt( prob, df=v )
  return( m+sqrt(s*v/(v-2))*q )
}
```

... your answer

The exact CI is

```
# Posterior parameters
a_n <- a_0 +0.5*n_obs
b_n <- b_0 +0.5*n_obs*var(y_obs) +0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)
# CI
studentT_inv(c(0.025,0.975), mu_n , b_n/lam_n/a_n, 2*a_n )
```

```
## [1] 21.52991 22.85026
```

The MC approximation of the CI is

```
# extract the sample, and copy it to smple
mu.smpl <-output$mu
quantile(mu.smpl, probs =c(0.025,0.975))
```

```
##      2.5%      97.5%
## 21.53196 22.84984
```

Task

Compute the exact 95% equal tail posterior credible interval of μ .

Hint

The marginal distribution of μ , is:

$$\mu|x_{1:n} \sim \text{St}_k(\mu_n, \frac{b_n}{\lambda_n a_n}, 2a_n)$$

where

$$\begin{aligned}\mu_n &= \frac{\lambda_0 \mu_n + n \bar{x}}{\lambda_0 + n} \\ \lambda_n &= \lambda_0 + n, \\ a_n &= a_0 + \frac{n}{2} \\ b_n &= b_0 + \frac{1}{2} n s^2 + \frac{1}{2} (\lambda_0 + n)^{-1} \lambda_0 n (\mu_0 - \bar{x})^2\end{aligned}$$

Compute the MC approximation of the 95% equal tail posterior credible interval of μ .

Hint:

Additional PDFs and CDFs are given

```
# Student T INVERSE CDF
invgamma_inv <- function (prob, a, b){
  return(qgamma(1 - prob, a, b)^(-1))
}
```

... your answer

The exact CI is

```
# Posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0 - mean(y_obs))^2/(lam_0 + n_obs)
# CI
invgamma_inv(c(0.025, 0.975), a_n, b_n)
```

```
## [1] 8.459318 14.765687
```

The MC approximation of the CI is

```
# extract the sample, and copy it to smple
sigma2.smpl <- output$sigma2
quantile(sigma2.smpl, probs = c(0.025, 0.975))
```

```
##      2.5%      97.5%
```

```
## 8.436661 14.764231
```

Task

- Compute and plot the MC approximate of the predictive PDF of the next outcome y_{n+1}

$$\begin{aligned} f(y_{n+1}|y_{1:n}) &= \int f(y_{n+1}|\mu, \sigma^2) \pi(\mu, \sigma^2|y_{1:n}) d\mu d\sigma^2 \\ &= E_{\pi}(f(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\ &= E_{\pi}(N(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\ &\approx \frac{1}{N} \sum_{j=1}^N N\left(y_{n+1}|\mu^{(j)}, (\sigma^{(j)})^2\right) \end{aligned}$$

for $y_{n+1} \in (5, 40)$ where $f(y_{n+1}|\mu, \sigma^2)$ is a Normal PDF with mean μ and variance σ^2 .

- Compute and plot the exact predictive PDF of the next outcome y_{n+1} , which is the PDF of

$$y_{n+1}|y_{1:n} \sim \text{St}(\mu_n, \frac{\lambda_n b_n}{(\lambda_n + 1)a_n}, 2a_n)$$

where

$$\begin{aligned} \mu_n &= \frac{\lambda_0 \mu_0 + n\bar{x}}{\lambda_0 + n} \\ \lambda_n &= \lambda_0 + n, \\ a_n &= a_0 + \frac{n}{2} \\ b_n &= b_0 + \frac{1}{2}ns^2 + \frac{1}{2}(\lambda_0 + n)^{-1}\lambda_0 n(\mu_0 - \bar{x})^2 \end{aligned}$$

for $y_{n+1} \in (5, 40)$.

Hint

Additional PDF and CDF functions are given

```
# Student T PDF
studentT_pdf <- function(x,m,s,v) {
  return(dt((x-m)/sqrt(s),df=v)/sqrt(s))
}
# Student T CDF
studentT_cdf <- function(x,m,s,v) {
  return(pt((x-m)/sqrt(s),df=v))
}
```

... your answer

The MC approximation is such as

$$\begin{aligned} f(y_{n+1}|y_{1:n}) &= \int f(y_{n+1}|\mu, \sigma^2) \pi(\mu, \sigma^2|y_{1:n}) d\mu d\sigma^2 \\ &= E_{\pi}(f(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\ &= E_{\pi}(N(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\ &\approx \frac{1}{N} \sum_{i=1}^N N(y_{n+1}|\mu^{(i)}, (\sigma^{(i)})^2) \end{aligned}$$

```

# Draw the MC approximation plot

## extract the sample, and copy it to smple
mu.smpl <- output$mu
sigma2.smpl <- output$sigma2

x_plot <- seq( from = min(
                    mean(mu.smpl)-5*mean(sqrt(sigma2.smpl))
                ),
              to = max(
                    mean(mu.smpl)+5*mean(sqrt(sigma2.smpl))
                ),
              length.out = 100)
y_plot <- rep(NA,100)
for (i in 1:100)
  y_plot[i] <- mean( dnorm( x_plot[i],
                           mu.smpl,
                           sqrt(sigma2.smpl)
                         ) )

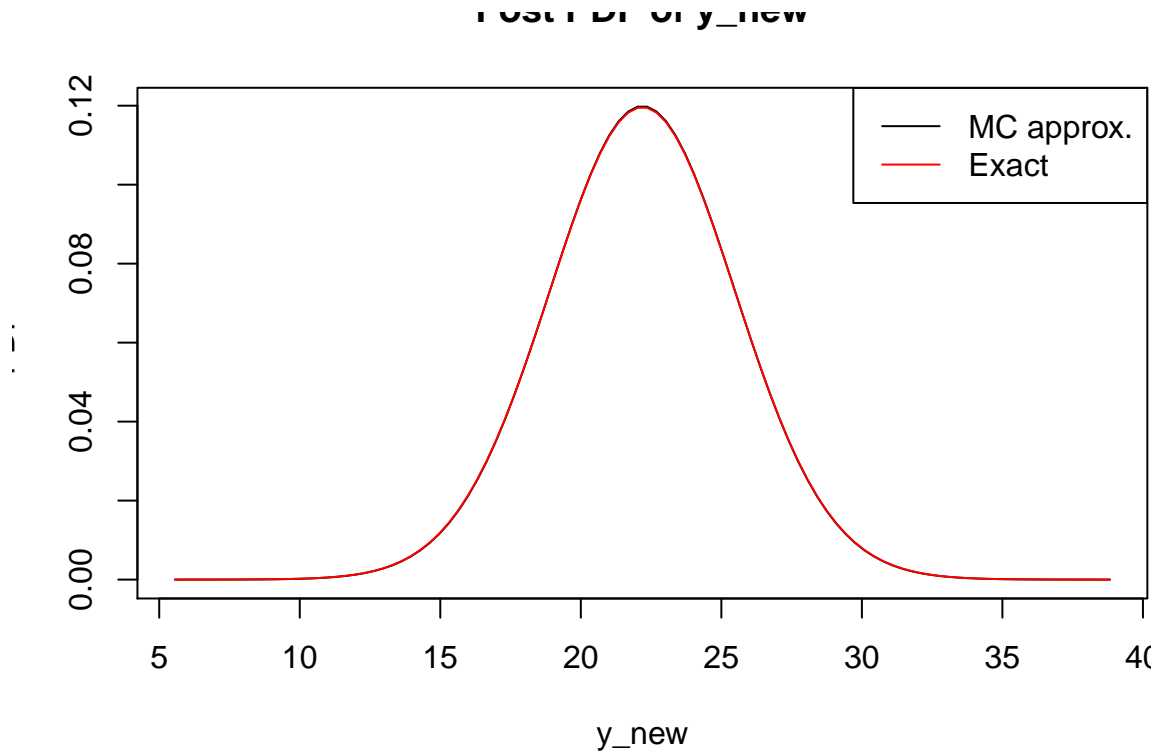
plot(x_plot,
     y_plot,
     type = 'l',
     main = "Post PDF of y_new",
     xlab = "y_new",
     ylab = "PDF")

# Draw the exact now

## Posterior parameters
a_n <- a_0 +0.5*n_obs
b_n <- b_0 +0.5*n_obs*var(y_obs) +0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0+n_obs*mean(y_obs)) / (lam_0+n_obs)
## draw the plot
### I m using the x_plot fromt he previous plot

y_plot <- studentT_pdf(x_plot , mu_n , (lam_n+1)*b_n/((lam_n)*a_n), 2*a_n)
lines(x_plot,
     y_plot,
     type = 'l',
     col = 'red')
legend("topright",
     legend=c("MC approx.", "Exact"),
     lty = c(1,1),
     col=c("black", "red"))

```



Task

- Compute the MC approximate of the predictive expected value of the next outcome y_{n+1}

$$\begin{aligned}
 E_f(y_{n+1}|y_{1:n}) &= \int y_{n+1} f(y_{n+1}|y_{1:n}) dy_{n+1} \\
 &= \int y_{n+1} \left(\int f(y_{n+1}|\mu, \sigma^2) \pi(\mu, \sigma^2|y_{1:n}) d\mu d\sigma^2 \right) dy_{n+1} \\
 &= \int \left(\int y_{n+1} f(y_{n+1}|\mu, \sigma^2) dy_{n+1} \right) \pi(\mu, \sigma^2|y_{1:n}) d\mu d\sigma^2 \\
 &= E_\pi(E_f(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\
 &= E_\pi(\mu|y_{1:n}) \\
 &\approx \frac{1}{N} \sum_{i=1}^N \mu^{(i)}
 \end{aligned}$$

- Compute the exact predictive expected value of the next outcome y_{n+1} which is

$$\begin{aligned}
 E_f(y_{n+1}|y_{1:n}) &= E_\pi(E_f(y_{n+1}|\mu, \sigma^2)|y_{1:n}) \\
 &= E_\pi(\mu|y_{1:n}) \\
 &= \mu_n
 \end{aligned}$$

... your answer

Regarding the MC approximate value:

```
mu.smpl <- output$mu
mu.mc <- mean(mu.smpl)
mu.mc
```

```
## [1] 22.19088
```

Regarding the exact value:

```
# Posterior parameters
a_n <- a_0 + 0.5*n_obs
b_n <- b_0 + 0.5*n_obs*var(y_obs) + 0.5*lam_0*n_obs*(mu_0 - mean(y_obs))^2/(lam_0 + n_obs)
lam_n <- lam_0 + n_obs
mu_n <- (lam_0*mu_0 + n_obs*mean(y_obs)) / (lam_0 + n_obs)
mu_n
```

```
## [1] 22.19009
```

Task

- Compute the MC approximate of the predictive variance of the next outcome y_{n+1}

$$\begin{aligned}\text{Var}_f(y_{n+1}|y_{1:n}) &= \text{E}_\pi(\text{Var}_f(y_{n+1}|\mu, \sigma^2)) + \text{Var}_\pi(\text{E}_f(y_{n+1}|\mu, \sigma^2)) \\ &= \text{E}_\pi(\sigma^2|y_{1:n}) + \text{Var}_\pi(\mu|y_{1:n}) \\ &\approx \frac{1}{N} \sum_{j=1}^N (\sigma^{(j)})^2 + \frac{1}{N} \sum_{j=1}^N \left((\mu^{(j)})^2 \right) - \left(\frac{1}{N} \sum_{j=1}^N \mu^{(j)} \right)^2\end{aligned}$$

- Compute the exact predictive expected value of the next outcome y_{n+1} which is

$$\text{Var}_f(y_{n+1}|y_{1:n}) = \frac{\lambda_n b_n}{(\lambda_n + 1)(a_n - 1)}$$

where

$$\begin{aligned}\mu_n &= \frac{\lambda_0 \mu_n + n \bar{x}}{\lambda_0 + n} \\ \lambda_n &= \lambda_0 + n, \\ a_n &= a_0 + \frac{n}{2} \\ b_n &= b_0 + \frac{1}{2} n s^2 + \frac{1}{2} (\lambda_0 + n)^{-1} \lambda_0 n (\mu_0 - \bar{x})^2\end{aligned}$$

because

$$y_{n+1}|y_{1:n} \sim \text{St}(\mu_n, \frac{\lambda_n b_n}{(\lambda_n + 1)a_n}, 2a_n)$$

... your answer

Regarding the MC approximate value:

```

## extract the sample, and copy it to smple
mu.smpl <-output$mu
sigma2.smpl <-output$sigma2
# compute the MC approximation
y.var.mc <- mean(sigma2.smpl) + mean(mu.smpl^2) - mean(mu.smpl)^2
y.var.mc

```

```
## [1] 11.23995
```

Regarding the exact value:

```

# Posterior parameters
a_n <- a_0 +0.5*n_obs
b_n <- b_0 +0.5*n_obs*var(y_obs) +0.5*lam_0*n_obs*(mu_0-mean(y_obs))^2/(lam_0+n_obs)
lam_n <- lam_0 + n_obs
y.var <- lam_n/(lam_n+1)*b_n/(a_n)
y.var

```

```
## [1] 10.85228
```