

hw7.R

mwilde

Tue Nov 21 18:40:47 2017

```
### 1
### (e)
### posterior mean
print(mean(rgamma(10000, shape=36, rate=11)))

## [1] 3.278267

### (f)
quantile(rgamma(10000, shape=36, rate=11),c(0.025,0.5,0.975))

##      2.5%      50%      97.5%
## 2.293547 3.245206 4.410697

print(c("lambda left = ", qgamma(0.025, shape = 36, rate=11)))

## [1] "lambda left = "      "2.29217794702866"

print(c("lambda r = ", qgamma(0.975, shape = 36, rate=11)))

## [1] "lambda r = "      "4.42513885173482"

### 2
### (a)
### Use discrete approximation method to get posterior

n = 10
y = c(9.95, 10.47, 10.68, 10.90, 11.11, 11.22, 11.53, 11.86, 12.31, 13.51)

mu0 = 10 #mean of prior distribution
tau20 <- 0.01 # precision (or 1/variance) of prior distribution
nu0 <- 1
phi20 <- 1

# compute sample statistics
mean.y <- mean(y)
var.y <- var(y)
n <- length(y)
print(c("mean"=mean.y, "var"=var.y, "n"=n))

##      mean      var      n
## 11.35400  1.03776 10.00000

G <- 100
H <- 100
mu.grid <- seq(10,13,length=G)
phi2.grid <- seq(0.01,3,length=H)
# Note: trial and error may be needed to find the
# right values for these sequences, so that the
# posterior is visible.
post.grid <- matrix(nrow=G,ncol=H) #to store posterior
for(g in 1:G){
```

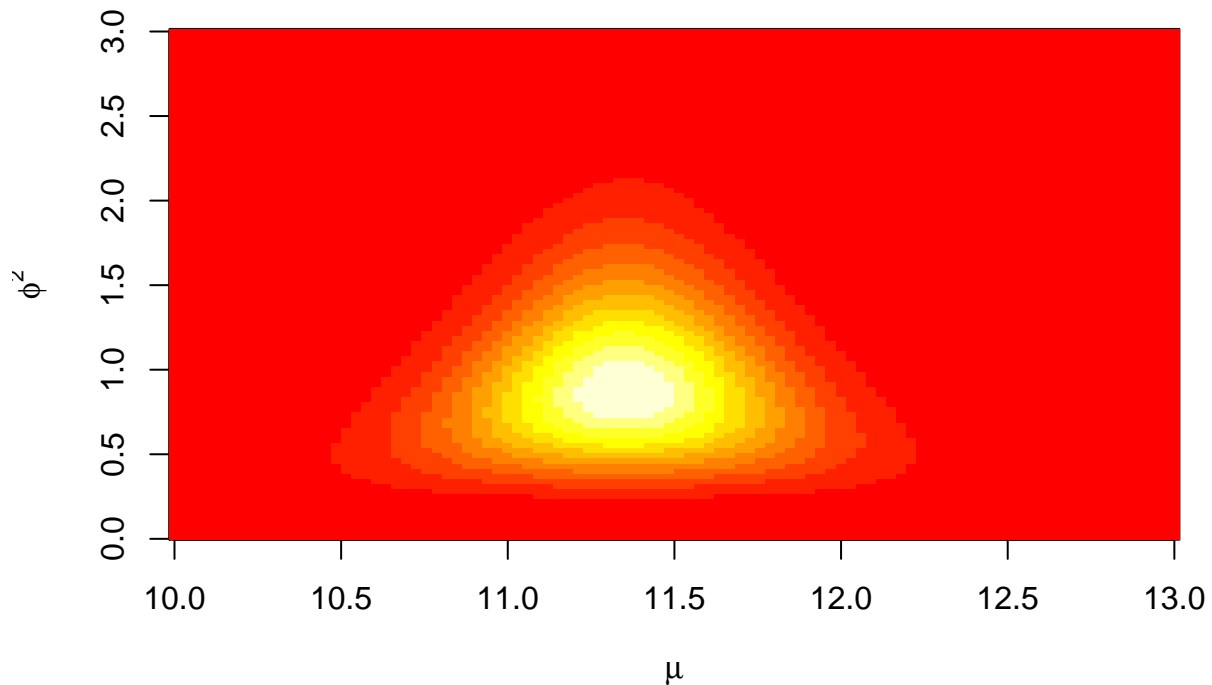
```

for(h in 1:H){
  post.grid[g,h] =
    dnorm(mu.grid[g],mu0,1/sqrt(tau20))*
    dgamma(phi2.grid[h],nu0/2,nu0/(2*phi20))*
    prod(dnorm(y,mu.grid[g],1/sqrt(phi2.grid[h])))
}
}
post.grid <- post.grid / sum(post.grid) # normalizing

### (i)
### heat map of joint posterior
image(mu.grid,phi2.grid,post.grid,
      main=expression(paste("Heat map for Posterior on (", mu, ",", phi^2, ")")),
      xlab=expression(mu),ylab=expression(phi^2))

```

Heat map for Posterior on (μ, ϕ^2)

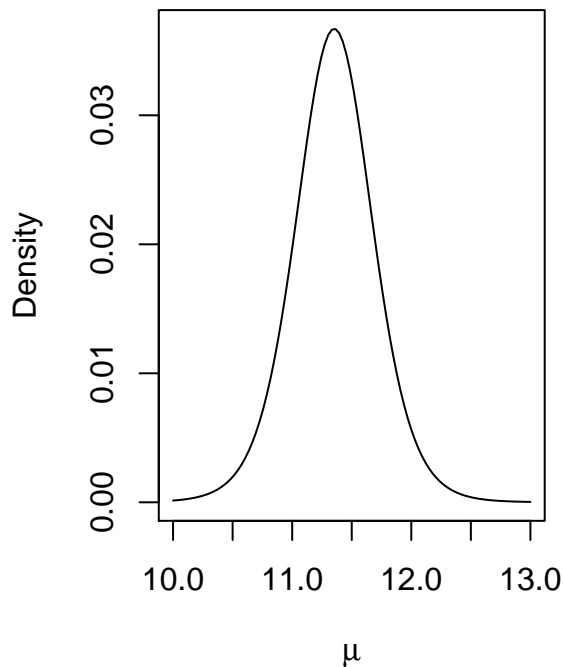


```

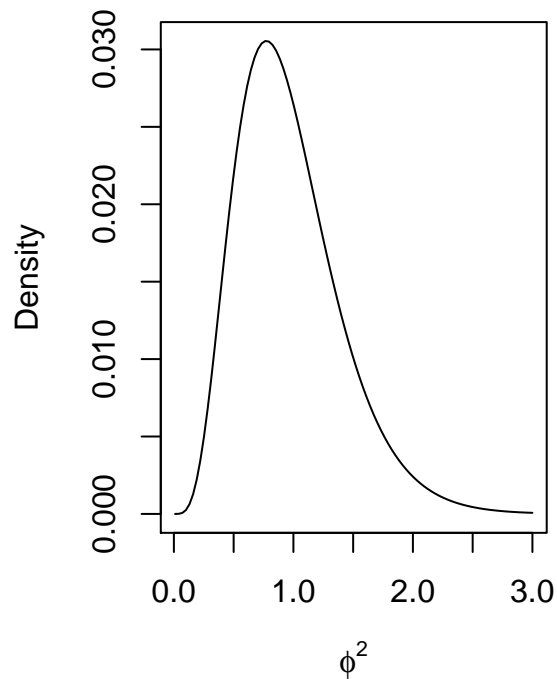
### (ii)
par(mfrow=c(1,2))
# Compute MARGINAL posteriors for mu and phi2
post.mu <- rowSums(post.grid)
post.phi2 <- colSums(post.grid)
plot(mu.grid,post.mu,type="l",
     main=expression(paste("Marginal posterior for ", mu)),
     xlab=expression(mu),ylab="Density")
plot(phi2.grid,post.phi2,type="l",
     main=expression(paste("Marginal posterior for ", phi^2)),
     xlab=expression(phi^2),ylab="Density")

```

Marginal posterior for μ



Marginal posterior for ϕ^2



```
### b) Gibbs sampler:

# Specify chain length
S <- 4000 # generate 4000 dependent samples
samples <- matrix(nrow=S,ncol=2) # to store samples
# col 1 is mu samples
# col 2 is phi^2 samples

### Starting values
samples[1,] <- c(mean.y, 1/var.y)

set.seed(1) #to ensure random numbers are always the same
for(s in 2:S){
  ### Generate a new value of mu from f(mu | prev value of phi2, data)
  prev.phi2 <- samples[s-1,2]
  mustar <- (mu0*tau20+n*mean.y*prev.phi2)/(tau20 +n*prev.phi2)
  phi2star <- tau20 +n*prev.phi2
  mu.new <- rnorm(1,mustar,1/sqrt(phi2star))
  ### Generate a new value of phi2 from f(phi2 | new value of mu, data)
  astar <- (nu0 + n)/2
  bstar <- (nu0/phi20 + (n-1)*var.y + n*(mean.y-mu.new)^2)/2

  phi2.new <- rgamma(1,astar,bstar)
  samples[s,] <- c(mu.new,phi2.new)
}

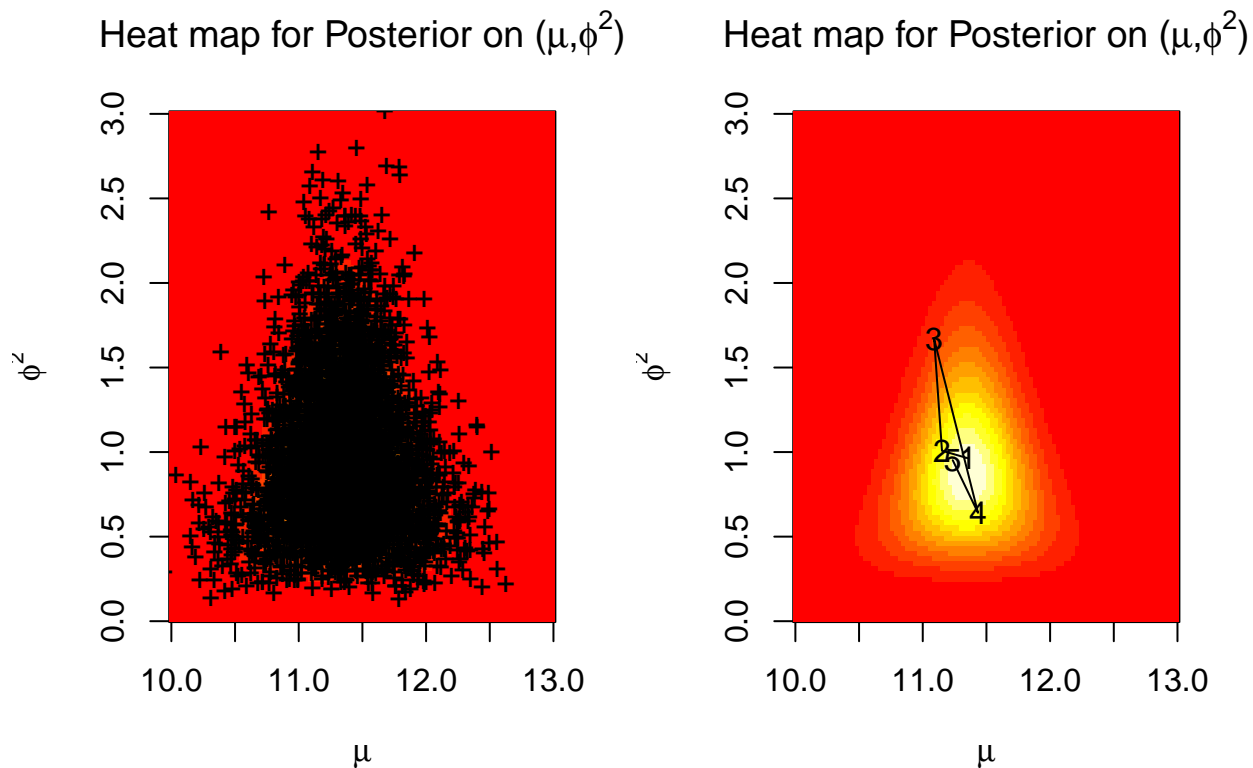
### scatter plot showing these 4000 draws
image(mu.grid,phi2.grid,post.grid,
```

```

    main=expression(paste("Heat map for Posterior on (", mu, ",", phi^2, ")")),
    xlab=expression(mu),ylab=expression(phi^2))
points(samples[,1],samples[,2],pch="+")

### To show burn-in and dependence
image(mu.grid,phi2.grid,post.grid,
      main=expression(paste("Heat map for Posterior on (", mu, ",", phi^2, ")")),
      xlab=expression(mu),ylab=expression(phi^2))
### Just the first 5 points
points(samples[1:5,1],samples[1:5,2],type="l")
text(samples[1:5,1],samples[1:5,2])

```



```

### (ii)
### Compute quantiles from marginal posterior distributions:
quantile(samples[,1],c(0.025,0.5,0.975)) # For mu

```

```

##      2.5%      50%      97.5%
## 10.63261 11.35859 12.06882

```

```

### (iii)
### Compute quantiles from marginal posterior distributions:
quantile(samples[,2],c(0.025,0.5,0.975)) # For phi^2

```

```

##      2.5%      50%      97.5%
## 0.3078316 0.8912881 1.9898418

```

```

### (iv)
### Quantiles for the population sd
quantile(1/sqrt(samples[,2]),c(0.025,0.5,0.975)) # For sd

```

##	2.5%	50%	97.5%
##	0.7089094	1.0592317	1.8023682