```
updatemu <- function(data,sigma2,mumu,sigma2mu){

  #Get the mean of the data
  xbar <- mean(data)

  #Get the length of the data
  n <- length(data)

  #compute parameters to make mean and variance equations simpler
  prec <- 1/sigma2
  precmu <- 1/sigma2mu

  #Get the mean of the posterior m
  meanpost <- (n*prec*xbar + precmu*mumu)/(n*prec+precmu)

  #Get the variance of the posterior mu
  varpost <- 1/(n*prec+precmu)

  #Output values
  out <- c(meanpost,varpost)
    return(out)
}



updatesig2 <- function(data,mu,prshape,prscale){

  #Get the length of the data as 'n'
  n <- length(data)

  #Get the posterior shape
  poshape <- n/2 + prshape

  #Get the sum of the differences squared of the data
  ssx <- sum((data-mu)^2)

  #Get the posterior scale parameter using the prior scale and sum of the squared x's
```

```
  poscale <- (2*prscale)/(prscale*ssx+2)

  #Output shape and scale
  out <- c(poshape,poscale)
  return(out)
}



gibbs <- function(data,mumu,s2mu,prsh,prsc,loops){

  #Create a matrix to store data with 2 columns and rows = loops
  out <- matrix(0,nrow=loops,ncol=2)

  #Make first value in the 'out' matrix to be the mean of the data
  out[1,1] <- mean(data)

  #Start for loop
  for (i in 2:loops){

    #Assign 'post' to the output of the updatesig2 function. This gives us shape and scale
parameters based on the first value in out and the data
    post <- updatesig2(data,out[i-1,1],prsh,prsc)

    #Generates a random number from the gamma distribution using shape and scale parameters we
just found in the previous step
    out[i,2] <- 1/rgamma(1,shape=post[1],scale=post[2])

    #Now independent of what just happened, assign 'post' to the mean and variance parameters
found using the updatemu funciton
    post <- updatemu(data,out[i,2],mumu,s2mu)

    #Generate a random number from the normal distribution using the mean and variance found the
in previous step
    out[i,1] <- rnorm(1,post[1],sqrt(post[2]))
  }

  #Return everythingbut the first row
  out <- out[-1,]
```

```
  return(out)
}
```