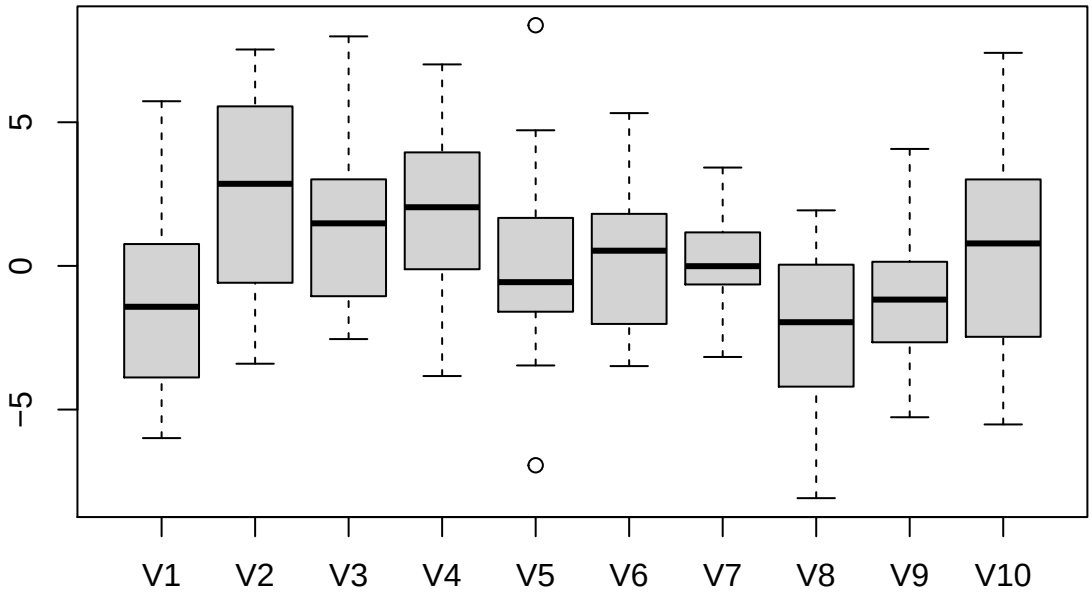
lecture_8

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

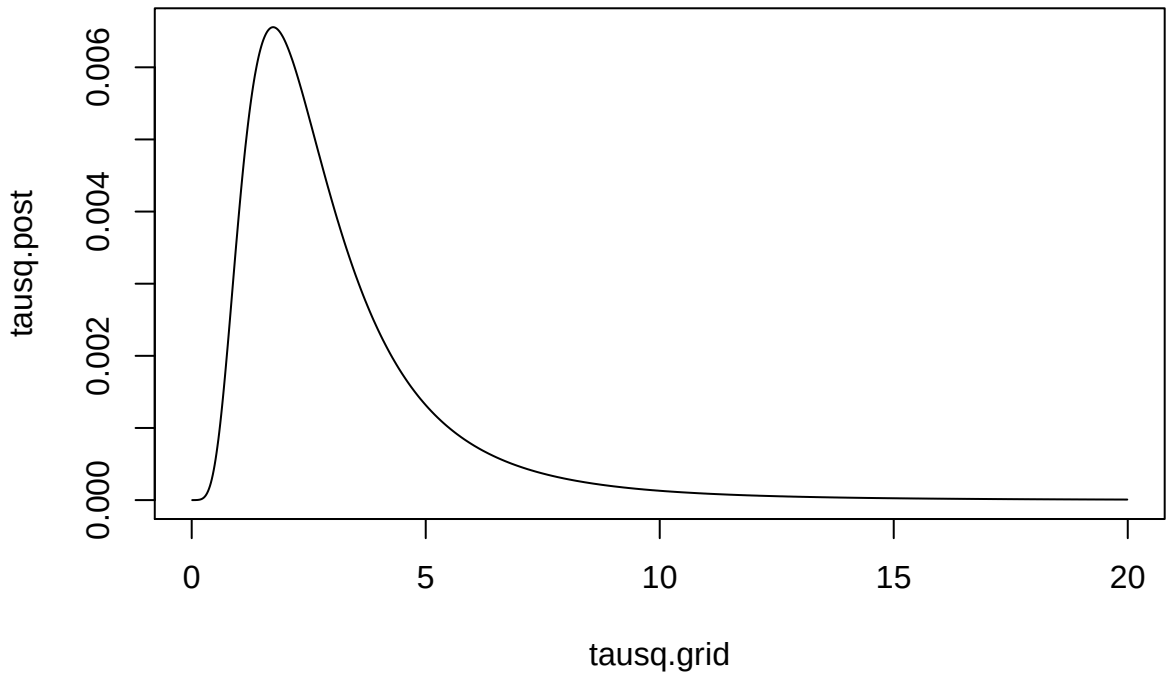
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#######################################################
####### Data for Normal Hierarchical Model ##########
#######################################################

## simulated data with truesigsq = 8, true tausq = 1.5, true mu0 = 0
data <- read.table("data/normhier.txt")
y<-data
par(mfrow=c(1,1))
boxplot(y)
```




```
##Calculating necessary statistics:
m <- length(y[1,])
n <- rep(NA,m)
means <- rep(NA,m)
for (i in 1:m){
  n[i] <- length(y[,i])
  means[i] <- mean(y[,i])
}
ntot <- sum(n)

## true sigmasq
```

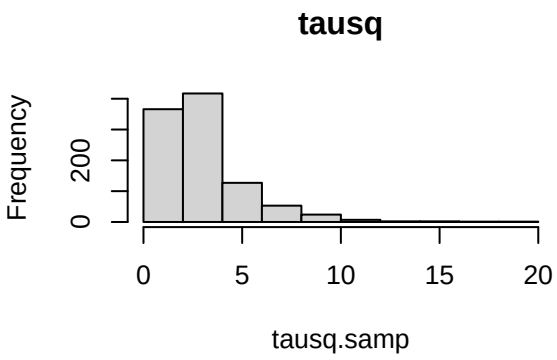
```r
truesigsq <- 8

#####################################################
# Sampling Parameters for Normal Hierarchical Model #
#####################################################

## finding right grid for tausq
tausq.grid <- ppoints(1000)*20

tausq.logpostfunc <- function(tausq){
    Vmu0 <- 1/sum(1/(tausq + truesigsq/n))
    mu0hat <- sum(means/(tausq + truesigsq/n))*Vmu0
    out <- -0.5*log(tausq)+0.5*log(Vmu0)
    for (group in 1:m){
        out <- out - 0.5*log(tausq + truesigsq/n[group])
    }
    for (group in 1:m){
        out <- out - 0.5*((means[group]-mu0hat)^2)/(tausq + truesigsq/n[group])
    }
    out
}
tausq.logpost <- rep(NA,1000)
for (i in 1:1000){
    tausq.logpost[i] <- tausq.logpostfunc(tausq.grid[i])
}
tausq.post <- exp(tausq.logpost-max(tausq.logpost))
tausq.post <- tausq.post/sum(tausq.post)

par(mfrow=c(1,1))
plot(tausq.grid,tausq.post,type="l")
```

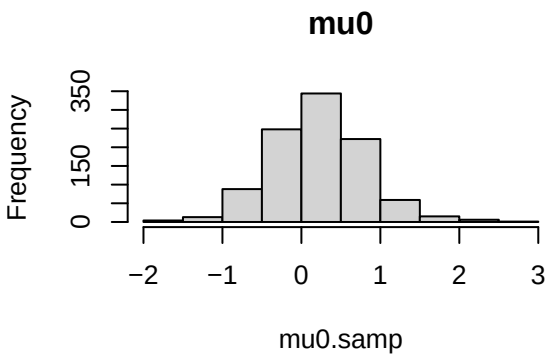


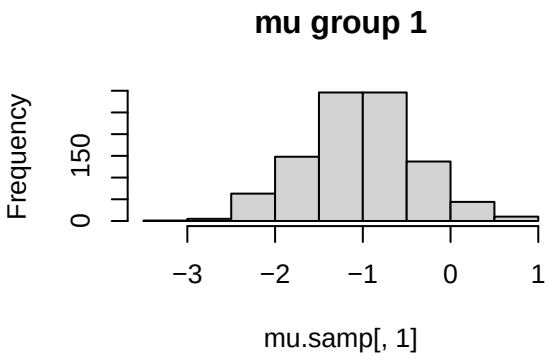
```r
numsamp <- 1000
tausq.samp <- rep(NA,numsamp)
mu0.samp <- rep(NA,numsamp)
mu.samp <- matrix(NA,nrow=numsamp,ncol=m)
```

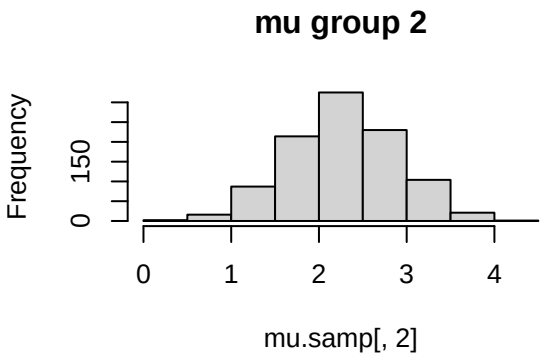
```r
for (i in 1:numsamp){
    # sampling tausq from grid of values
    curtausq <- sample(tausq.grid,size=1,prob=tausq.post)
    # sampling mu0 given curtausq
    Vmu0 <- 1/sum(1/(curtausq + truesigsq/n))
    mu0hat <- sum(means/(curtausq + truesigsq/n))*Vmu0
    curmu0 <- rnorm(1,mean=mu0hat,sd=sqrt(Vmu0))
    # sampling group means given curtausq and curmu0
    curmu <- rep(NA,m)
    for (j in 1:m){
        curvar <- 1/(n[j]/truesigsq + 1/curtausq)
        curmean <- (means[j]*n[j]/truesigsq + curmu0/curtausq)*curvar
        curmu[j] <- rnorm(1,mean=curmean,sd=sqrt(curvar))
    }
    tausq.samp[i] <- curtausq
    mu0.samp[i] <- curmu0
    mu.samp[i,] <- curmu
    print (i)
}
```

```r
#########################################################
########### Examining Model Parameters ##############
#########################################################

par(mfrow=c(2,2))
hist(tausq.samp,main="tausq")
hist(mu0.samp,main="mu0")
hist(mu.samp[,1],main="mu group 1")
hist(mu.samp[,2],main="mu group 2")
```

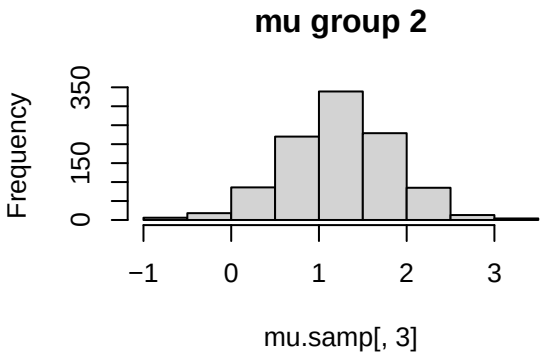


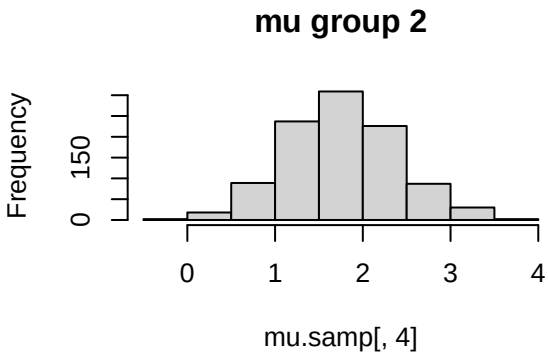


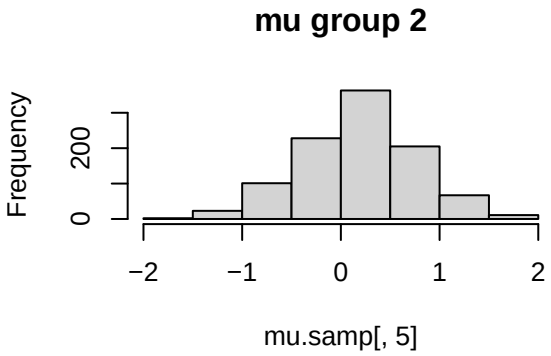




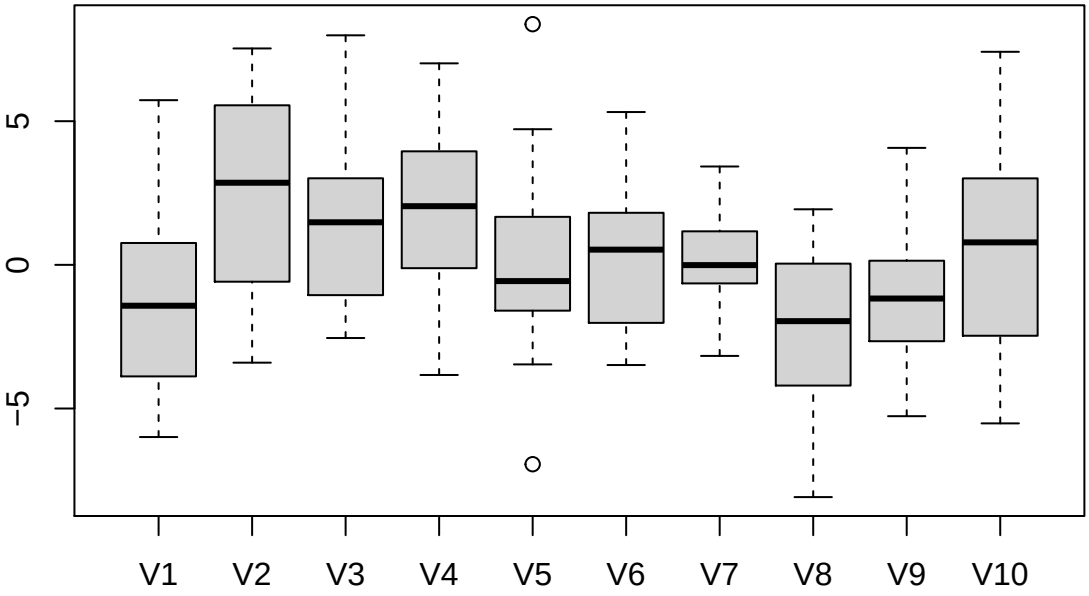

```
hist(mu.samp[,3],main="mu group 2")
hist(mu.samp[,4],main="mu group 2")
hist(mu.samp[,5],main="mu group 2")

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








```
boxplot(y)
```



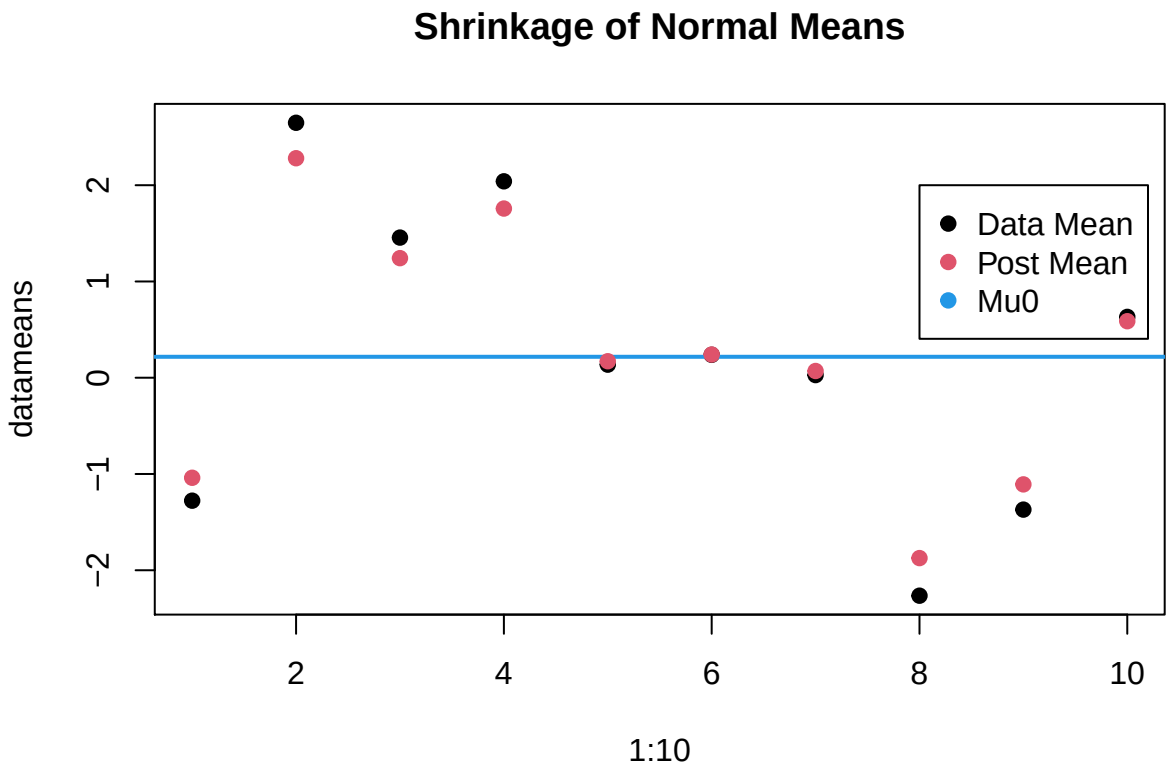

```
# posterior probability group 5 has greater mean than group 6
postprob <- sum(mu.samp[,5] > mu.samp[,6])/numsamp
postprob
```

```
## [1] 0.441
```

```
# posterior probability group 2 has greater mean than group 1
postprob <- sum(mu.samp[,2] > mu.samp[,1])/numsamp
postprob
```

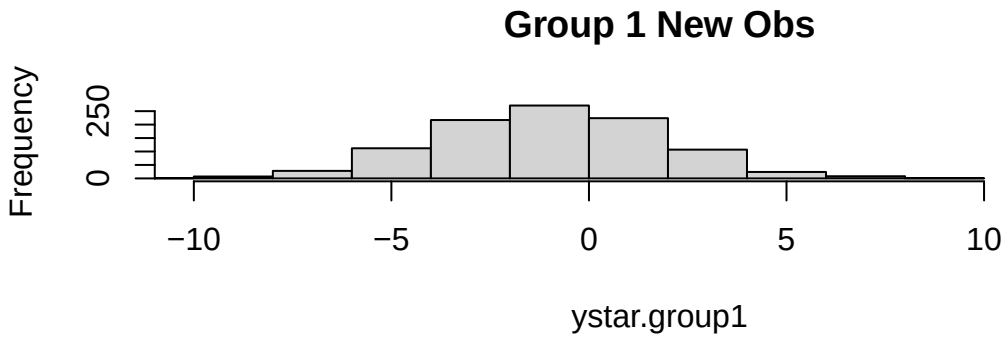
```
## [1] 1
```

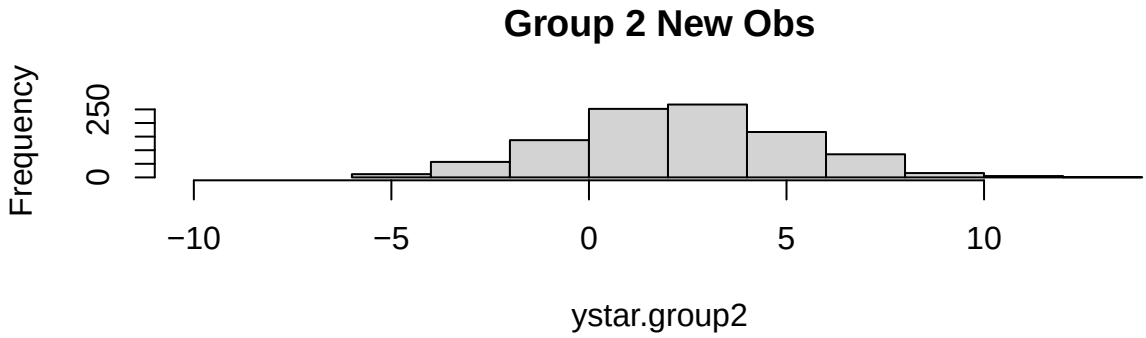
```
#########################################################
######### Examining Shrinkage Graphically ###########
#########################################################

datameans <- apply(y,2,mean)
postmeans <- apply(mu.samp,2,mean)
mu0.mean <- mean(mu0.samp)

par(mfrow=c(1,1))
plot(1:10,datameans,main="Shrinkage of Normal Means",pch=19)
abline(h=mu0.mean,col=4,lwd=2)
points(1:10,postmeans,pch=19,col=2)
legend(8,2,c("Data Mean","Post Mean","Mu0"),pch=19,col=c(1,2,4))
```



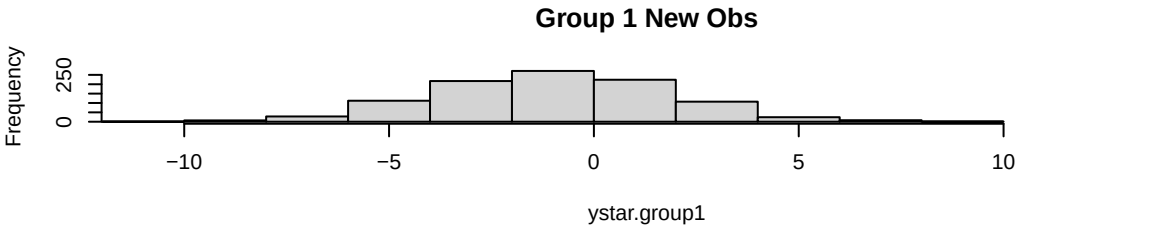

```
#########################################################
######### Posterior Predictive Sampling #############
#########################################################

## sampling distribution of new observation
## from a currently existing group

ystar.group1 <- rep(NA,numsamp)
ystar.group2 <- rep(NA,numsamp)
for (i in 1:numsamp){
    ystar.group1[i] <- rnorm(1,mean=mu.samp[i,1],sd=sqrt(truesigsq))
    ystar.group2[i] <- rnorm(1,mean=mu.samp[i,2],sd=sqrt(truesigsq))
}

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

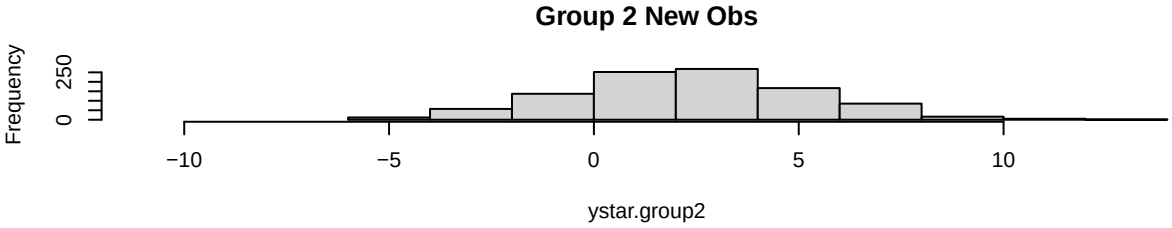
```
xmin <- min(c(ystar.group1,ystar.group2))
xmax <- max(c(ystar.group1,ystar.group2))
hist(ystar.group1,main="Group 1 New Obs",xlim=c(xmin,xmax))
hist(ystar.group2,main="Group 2 New Obs",xlim=c(xmin,xmax))
```

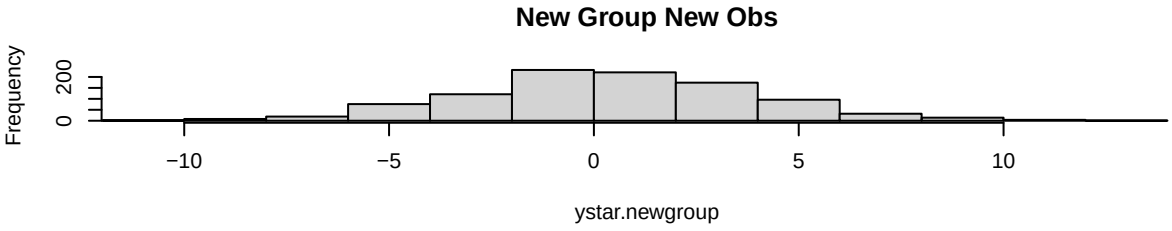





```
## sampling distribution of new observation
## from an entirely new group

ystar.newgroup <- rep(NA,numsamp)
for (i in 1:numsamp){
    mu.newgroup <- rnorm(1,mean=mu0.samp[i],sd=sqrt(tausq.samp[i]))
    ystar.newgroup[i] <- rnorm(1,mean=mu.newgroup,sd=sqrt(truesigsq))
}

par(mfrow=c(3,1))
xmin <- min(c(ystar.group1,ystar.group2,ystar.newgroup))
xmax <- max(c(ystar.group1,ystar.group2,ystar.newgroup))
hist(ystar.group1,main="Group 1 New Obs",xlim=c(xmin,xmax))
hist(ystar.group2,main="Group 2 New Obs",xlim=c(xmin,xmax))
hist(ystar.newgroup,main="New Group New Obs",xlim=c(xmin,xmax))
```


Group 1 New Obs
Frequency
0
250
−10
−5
0
5
10
ystar.group1


Group 2 New Obs
Frequency
0
250
−10
−5
0
5
10
ystar.group2


New Group New Obs
Frequency
0
200
−10
−5
0
5
10
ystar.newgroup