Classroom Simulation: Understanding One-Way Random-Effect ANOVA
Eric A. Suess, Bruce E. Trumbo, and Yun Jiang, CSU, East Bay
Poster: Section on Statistics Education, JSM 2005, Minneapolis, MN

 # Fig. 7. R code for our Gibbs Sampler.
 # Fig. 8. Graphics for Gibbs Sampler: Dataset 1.
To do the Gibbs Sampler for Dataset2, just change the value of batch sd. and error sd.

```
#simulate dataset 1
set.seed(1237) # R
g <- 30      # number of batches
r <- 5       # number of replications per batch
mu.grand <- 1000    # the grand mean
sd.Bat <- 25        # the batch stand deviation
sd.Err <- 15        # the error stand deviation
Y <- matrix(0, nrow=r, ncol=g)
mu <- numeric(g)
for (i in 1:g)
 {mu[i] <- rnorm(1, mu.grand, sd.Bat)
   for (j in 1:r)
      {Y[j,i] <- rnorm(1, mu[i], sd.Err)
      }
  }
x <- round(as.vector(Y))
Bat <- as.factor(rep(1:g, each = r))
FRAM <- data.frame(x, Bat)
X <- round(t(Y))
X.bar <- apply(X, 1, mean)#batch sample means

# Assume g x r matrix X of observations
m <- 100000 # iterations
b <- m/4     # burn-in
va <- numeric(m); ve <- numeric(m)
mu <- numeric(m); a <- rowMeans(X)
## Prior parameters, G/S Scenario I
a1 <- 0 # mean of prior on mu
b1 <- 10^(10) # var of prior on mu
a2 <- .001 # shape of prior on batch var
b2 <- .001 # rate of prior on batch var
a3 <- .001 # shape of prior on error var
b3 <- .001 # rate of prior on error var
## Initial values of model parameters
mu[1] <- 1500
va[1] <- 1
```

```
ve[1] <- 1
for (n in 2:m){
va[n] <- 1/rgamma(1, a2 + g/2,b2 + sum((a - mu[n-1])^2)/2)
ve[n] <- 1/rgamma(1, a3 + r*g/2,b3 + sum((X - matrix(a, g, r))^2)/2)
mu[n] <- rnorm(1, (va[n]*a1 +b1*sum(a))/(va[n] +
   g*b1),sqrt(b1*va[n]/(va[n] + g*b1)))
a <- rnorm(g, (r*va[n]*X.bar +ve[n]*mu[n])/(r*va[n] +
   ve[n]),sqrt((va[n]*ve[n])/(r*va[n]+ve[n])))
}

pm <- c(-1,1)
par(mfrow=c(4,3))

l.va <- "Est Batch Var"
l.sa <- "Est Batch SD"
l.cva <- "Cum Batch Var"
l.csa <- "Cum Batch SD"
l.dn <- "Density"
l.sq <- "Iteration"
plot(sqrt(va),type="l", xlab=l.sq, ylab=l.sa, main="")
plot(cumsum(sqrt(va))/(1:m),type="l",
  ylim=mean(sqrt(va))+pm*sd(sqrt(va)), xlab=l.sq, ylab=l.csa, main="")
hist(sqrt(va[(b+1):m]), prob=T, xlab=l.sa, ylab=l.dn, main="")

l.ve <- "Est Error Var"
l.se <- "Est Error SD"
l.cve <- "Cum Error Var"
l.cse <- "Cum Error SD"
plot(sqrt(ve),type="l",xlab=l.sq,ylab=l.se, main="")
plot(cumsum(sqrt(ve))/(1:m),type="l",
  ylim=mean(sqrt(ve))+pm*sd(sqrt(ve)), xlab=l.sq, ylab=l.cse, main="")
hist(sqrt(ve[(b+1):m]), prob=T,xlab=l.se,ylab=l.dn, main="")

l.mu <- "Est Grand Mean"
l.cmu <- "Cum Grand Mean"
plot(mu,type="l", xlab=l.sq, ylab=l.mu, main="")
plot(cumsum(mu)/(1:m),type="l",
  ylim=mean(mu)+pm*sd(mu), xlab=l.sq, ylab=l.cmu, main="")
hist(mu[(b+1):m], prob=T, xlab=l.mu, ylab=l.dn, main="")

l.rat <- "ICC"
l.crat <- "Cum ICC"
rat <- va/(va+ve)
plot(rat,type="l", xlab=l.sq, ylab=l.rat, main="")
```

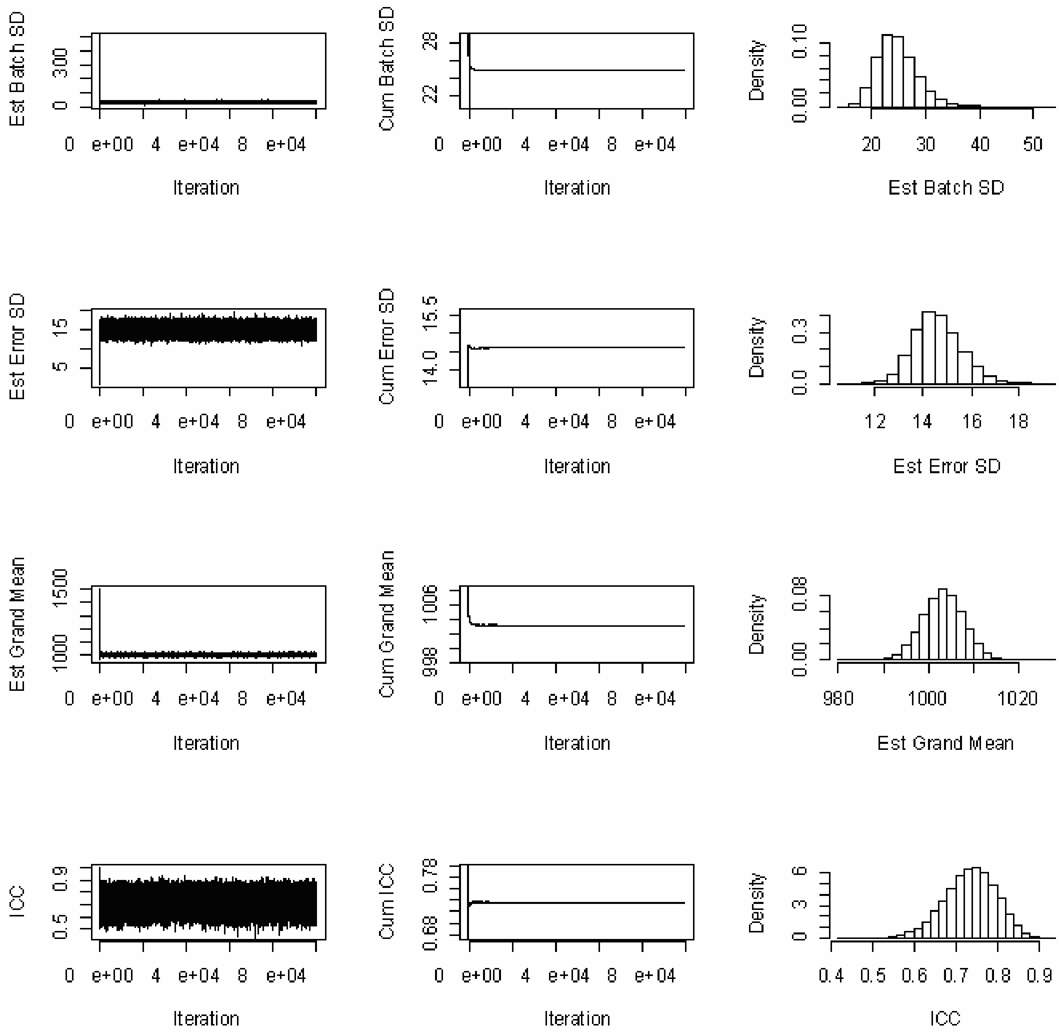
```
plot(cumsum(rat)/(1:m),type="l",ylim=mean(rat)+pm*sd(rat), xlab=l.sq,
  ylab=l.crat, main="")
hist(rat[(b+1):m], prob=T, xlab=l.rat, ylab=l.dn, main="")

#Bayes estimate(R,Gibbs) for among batches standard deviation
quantile(sqrt(va[(b+1):m]), c(.025, .975))
summary(sqrt(va[(b+1):m]))
#Bayes estimate(R,Gibbs) for within batches standard deviation
quantile(sqrt(ve[(b+1):m]), c(.025, .975))
summary(sqrt(ve[(b+1):m]))
#Bayes estimate(R,Gibbs) for the grand mean
quantile(mu[(b+1):m], c(.025, .975))
summary(mu[(b+1):m])
#Bayes estimate(R,Gibbs) for ICC
quantile(rat[(b+1):m], c(.025, .975))
summary(rat[(b+1):m])

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

☆ Output

Fig. 8. Graphics for Gibbs Sampler: Dataset 1.



Bayes Estimates and Probability Intervals for dataset 1.

```
> #Bayes estimate(R,Gibbs) for among batches standard deviation
> quantile(sqrt(va[(b+1):m]), c(.025, .975))
    2.5%    97.5%
18.87721 33.06647
> summary(sqrt(va[(b+1):m]))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.08   22.29   24.46   24.85   26.98   53.15
> #Bayes estimate(R,Gibbs) for within batches standard deviation
> quantile(sqrt(ve[(b+1):m]), c(.025, .975))
    2.5%    97.5%
12.87604 16.60274
> summary(sqrt(ve[(b+1):m]))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.89   13.93   14.54   14.59   15.20   19.41
> #Bayes estimate(R,Gibbs) for the grand mean
> quantile(mu[(b+1):m], c(.025, .975))
     2.5%     97.5%
 993.9318 1012.5510
> summary(mu[(b+1):m])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    981    1000    1003    1003    1006    1026
> #Bayes estimate(R,Gibbs) for ICC
> quantile(rat[(b+1):m], c(.025, .975))
     2.5%     97.5%
0.6085973 0.8446031
> summary(rat[(b+1):m])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4320  0.6965  0.7389  0.7358  0.7784  0.9255
>
```

Classroom Simulation: Understanding One-Way Random-Effect ANOVA
Eric A. Suess, Bruce E. Trumbo, and Yun Jiang, CSU, East Bay
Poster: Section on Statistics Education, JSM 2005, Minneapolis, MN

 # R code for our Gibbs Sampler, dealing with dataset2.
not shown in the paper

```
#simulate dataset 2
set.seed(12) # R
g <- 30     # number of batches
r <- 5      # number of replications per batch
mu.grand <- 1000   # the grand mean
sd.Bat <- 1        # the batch stand deviation
sd.Err <- 25        # the error stand deviation
Y <- matrix(0, nrow=r, ncol=g)
mu <- numeric(g)
for (i in 1:g)
 {mu[i] <- rnorm(1, mu.grand, sd.Bat)
   for (j in 1:r)
      {Y[j,i] <- rnorm(1, mu[i], sd.Err)
      }
  }
x <- round(as.vector(Y))
Bat <- as.factor(rep(1:g, each = r))
FRAM <- data.frame(x, Bat)
X <- round(t(Y))
X.bar <- apply(X, 1, mean)#batch sample means

# Assume g x r matrix X of observations
m <- 100000 # iterations
b <- m/4    # burn-in
va <- numeric(m); ve <- numeric(m)
mu <- numeric(m); a <- rowMeans(X)
## Prior parameters, G/S Scenario I
a1 <- 0 # mean of prior on mu
b1 <- 10^(10) # var of prior on mu
a2 <- .001 # shape of prior on batch var
b2 <- .001 # rate of prior on batch var
a3 <- .001 # shape of prior on error var
b3 <- .001 # rate of prior on error var
## Initial values of model parameters
mu[1] <- 1500
va[1] <- 1
ve[1] <- 1
```

```
for (n in 2:m){
va[n] <- 1/rgamma(1, a2 + g/2,b2 + sum((a - mu[n-1])^2)/2)
ve[n] <- 1/rgamma(1, a3 + r*g/2,b3 + sum((X - matrix(a, g, r))^2)/2)
mu[n] <- rnorm(1, (va[n]*a1 +b1*sum(a))/(va[n] +
   g*b1),sqrt(b1*va[n]/(va[n] + g*b1)))
a <- rnorm(g, (r*va[n]*X.bar +ve[n]*mu[n])/(r*va[n] +
   ve[n]),sqrt((va[n]*ve[n])/(r*va[n]+ve[n])))
}

pm <- c(-1,1)
par(mfrow=c(4,3))

l.va <- "Est Batch Var"
l.sa <- "Est Batch SD"
l.cva <- "Cum Batch Var"
l.csa <- "Cum Batch SD"
l.dn <- "Density"
l.sq <- "Iteration"
plot(sqrt(va),type="l", xlab=l.sq, ylab=l.sa, main="")
plot(cumsum(sqrt(va))/(1:m),type="l",
  ylim=mean(sqrt(va))+pm*sd(sqrt(va)), xlab=l.sq, ylab=l.csa, main="")
hist(sqrt(va[(b+1):m]), prob=T, xlab=l.sa, ylab=l.dn, main="")

l.ve <- "Est Error Var"
l.se <- "Est Error SD"
l.cve <- "Cum Error Var"
l.cse <- "Cum Error SD"
plot(sqrt(ve),type="l",xlab=l.sq,ylab=l.se, main="")
plot(cumsum(sqrt(ve))/(1:m),type="l",
  ylim=mean(sqrt(ve))+pm*sd(sqrt(ve)), xlab=l.sq, ylab=l.cse, main="")
hist(sqrt(ve[(b+1):m]), prob=T,xlab=l.se,ylab=l.dn, main="")

l.mu <- "Est Grand Mean"
l.cmu <- "Cum Grand Mean"
plot(mu,type="l", xlab=l.sq, ylab=l.mu, main="")
plot(cumsum(mu)/(1:m),type="l",
  ylim=mean(mu)+pm*sd(mu), xlab=l.sq, ylab=l.cmu, main="")
hist(mu[(b+1):m], prob=T, xlab=l.mu, ylab=l.dn, main="")

l.rat <- "ICC"
l.crat <- "Cum ICC"
rat <- va/(va+ve)
plot(rat,type="l", xlab=l.sq, ylab=l.rat, main="")
```

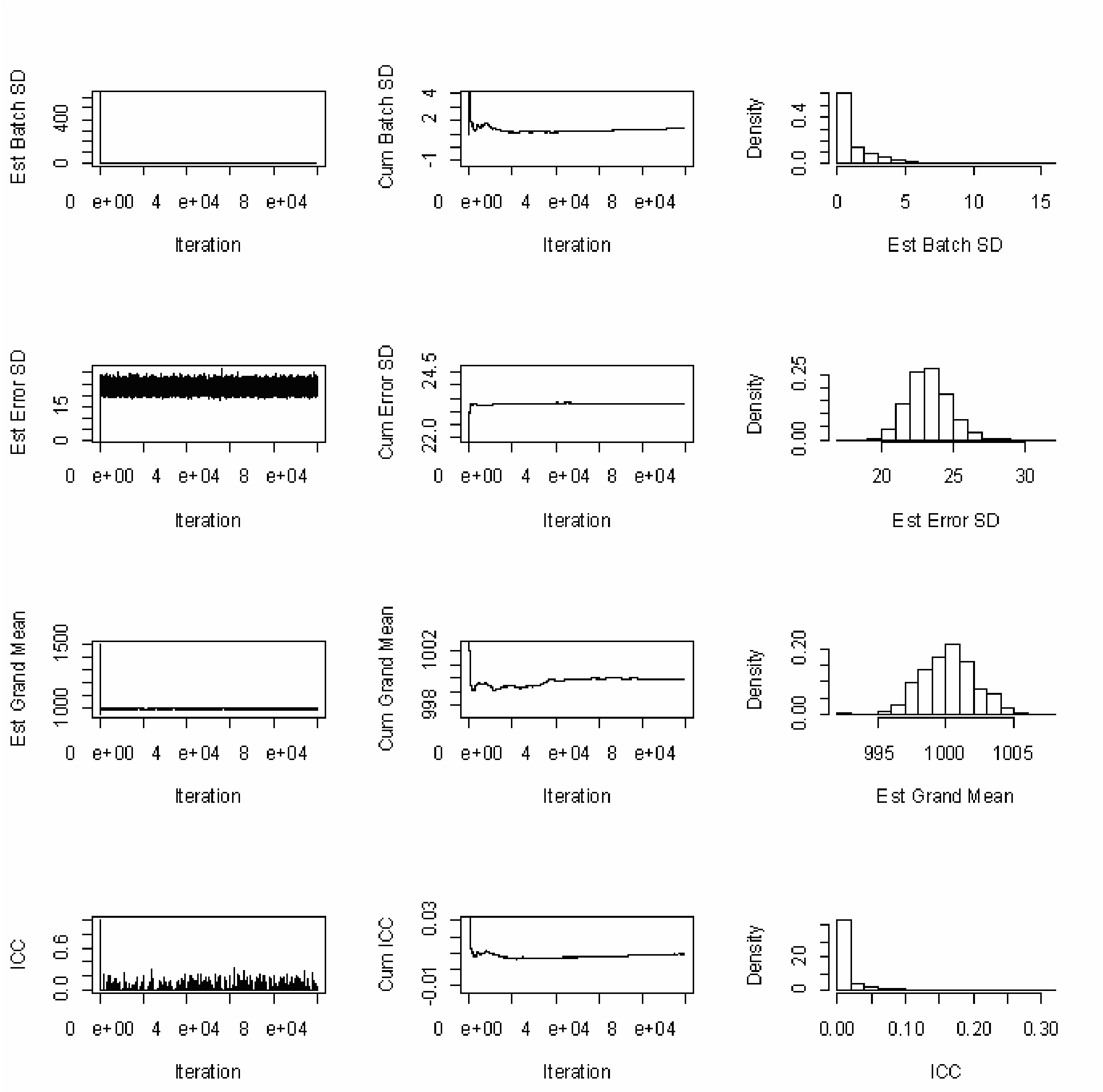
```
plot(cumsum(rat)/(1:m),type="l",ylim=mean(rat)+pm*sd(rat), xlab=l.sq,
  ylab=l.crat, main="")
hist(rat[(b+1):m], prob=T, xlab=l.rat, ylab=l.dn, main="")

#Bayes estimate(R,Gibbs) for among batches standard deviation
quantile(sqrt(va[(b+1):m]), c(.025, .975))
summary(sqrt(va[(b+1):m]))
#Bayes estimate(R,Gibbs) for within batches standard deviation
quantile(sqrt(ve[(b+1):m]), c(.025, .975))
summary(sqrt(ve[(b+1):m]))
#Bayes estimate(R,Gibbs) for the grand mean
quantile(mu[(b+1):m], c(.025, .975))
summary(mu[(b+1):m])
#Bayes estimate(R,Gibbs) for ICC
quantile(rat[(b+1):m], c(.025, .975))
summary(rat[(b+1):m])

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

☆ Output

Graphics for Gibbs Sampler: Dataset 2.



Bayes Estimates and Probability Intervals for dataset 2.

```
> #Bayes estimate(R,Gibbs) for among batches standard deviation
> quantile(sqrt(va[(b+1):m]), c(.025, .975))
      2.5%      97.5%
0.03319708 6.63352395
> summary(sqrt(va[(b+1):m]))
    Min.  1st Qu.  Median    Mean  3rd Qu.     Max.
 0.01274  0.17100  0.60550  1.41200  2.00600 15.55000
> #Bayes estimate(R,Gibbs) for within batches standard deviation
> quantile(sqrt(ve[(b+1):m]), c(.025, .975))
    2.5%    97.5%
20.77900 26.13708
> summary(sqrt(ve[(b+1):m]))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  17.55   22.32   23.21   23.27   24.16   31.55
> #Bayes estimate(R,Gibbs) for the grand mean
> quantile(mu[(b+1):m], c(.025, .975))
     2.5%     97.5%
 996.2833 1004.0721
> summary(mu[(b+1):m])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  992.0   998.8  1000.0  1000.0  1001.0  1007.0
> #Bayes estimate(R,Gibbs) for ICC
> quantile(rat[(b+1):m], c(.025, .975))
        2.5%        97.5%
2.013907e-06 7.830048e-02
> summary(rat[(b+1):m])
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
2.774e-07 5.384e-05 6.772e-04 9.496e-03 7.403e-03 3.161e-01
>
```