StatMod2 - Multinomial Dirichlet - Mixture of Normals

Maurice Diesendruck

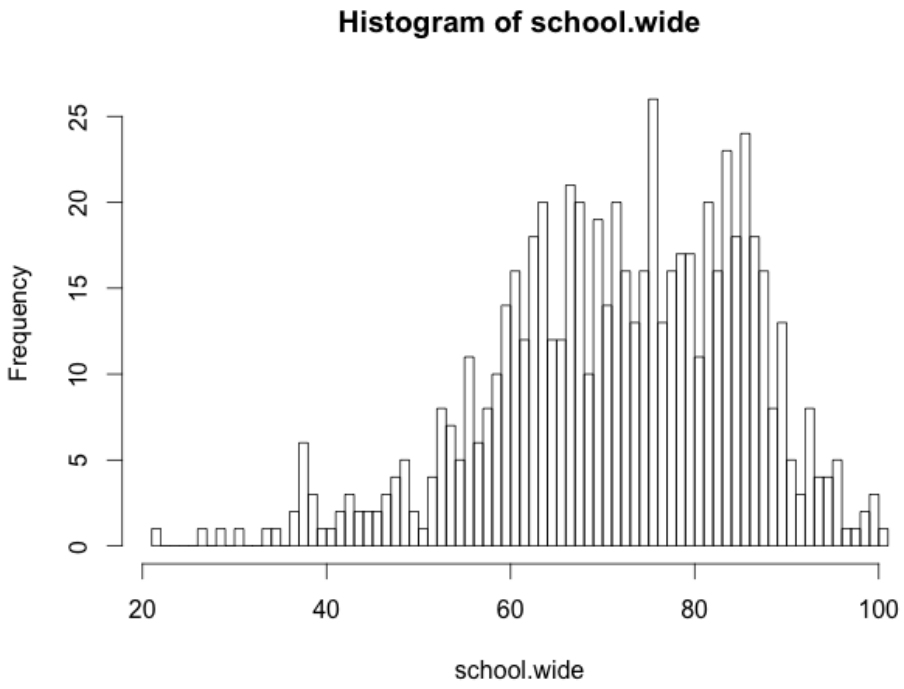
April 23, 2015

1 Simulated Test Scores

1.1 Generate Data

The school-wide results appear unimodal and slightly skewed left.

```
simulate.scores <- function() {
  y.rem <- rnorm(100, 55, 15)
  y.avg <- rnorm(400, 70, 10)
  y.hon <- rnorm(150, 85, 5)
  school.wide <- c(y.rem, y.avg, y.hon)
  hist(school.wide, breaks=100)
  return (school.wide)
}
school.wide <- simulate.scores()
```



1.2 Write Full Conditionals

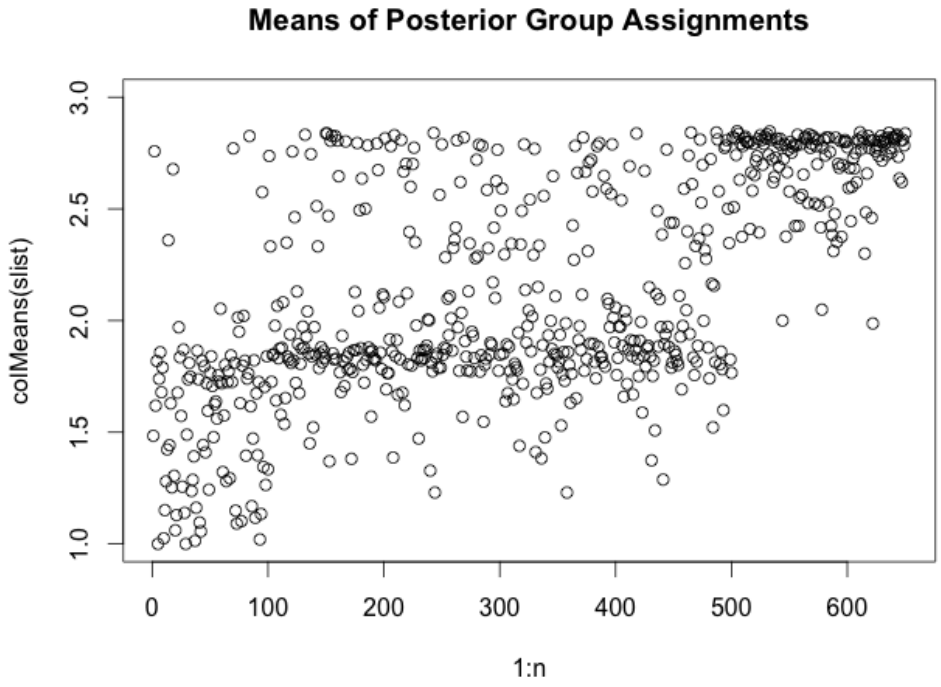
Write functions to draw from full conditional posterior distributions of gamma.i and w.

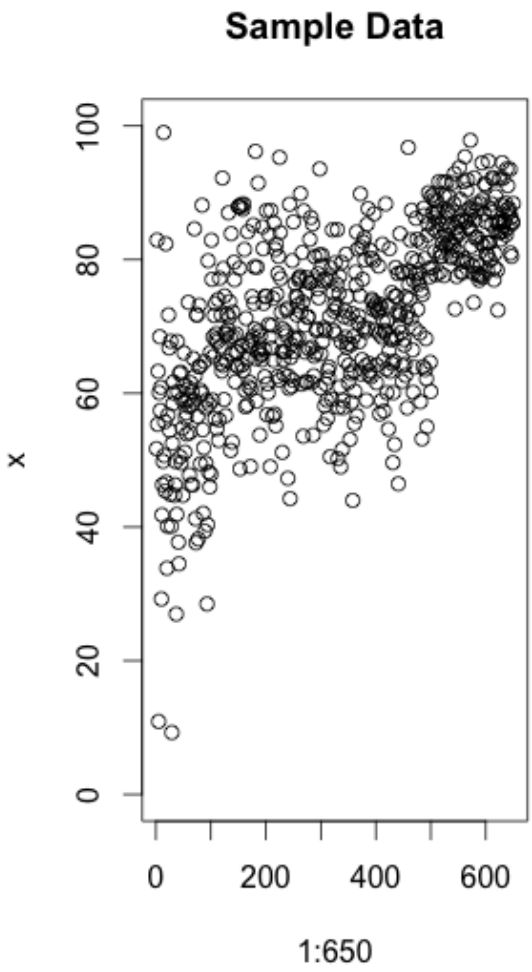
```
sample.gamma <- function(y.i, w) {
  if (w==w1) {
    p <- w*dnorm(y.i, 55, 15)
  } else if (w==w2) {
    p <- w*dnorm(y.i, 70, 10)
  } else if (w==w3) {
    p <- w*dnorm(y.i, 85, 5)
  }
  return (p)
}
sample.w <- function(group.counts=c(n1, n2, n3), dir.params=c(a1, a2, a3)) {
  rdirichlet(1, c(group.counts[1]+dir.params[1],
                  group.counts[2]+dir.params[2],
                  group.counts[3]+dir.params[3]))
}
```

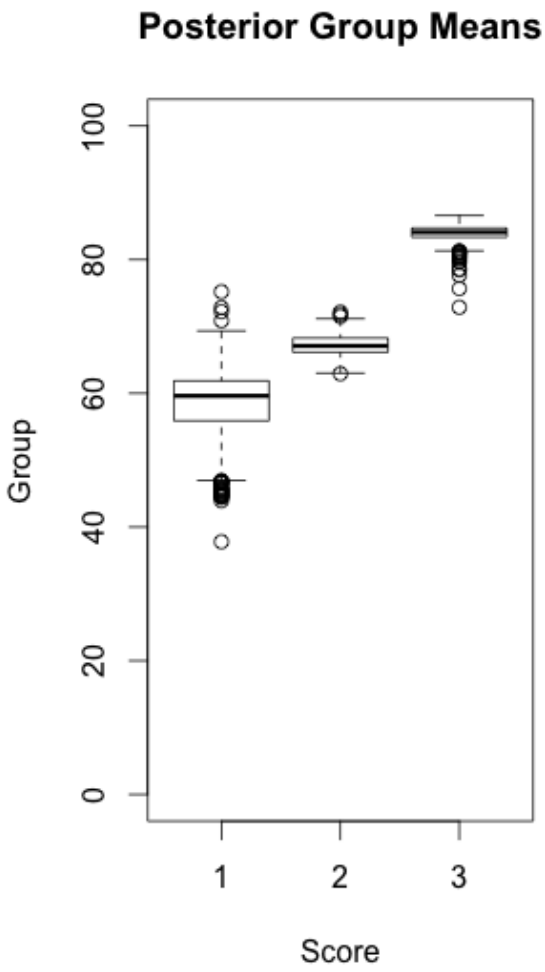
1.3 Posterior Given Unknown Mean and Variance

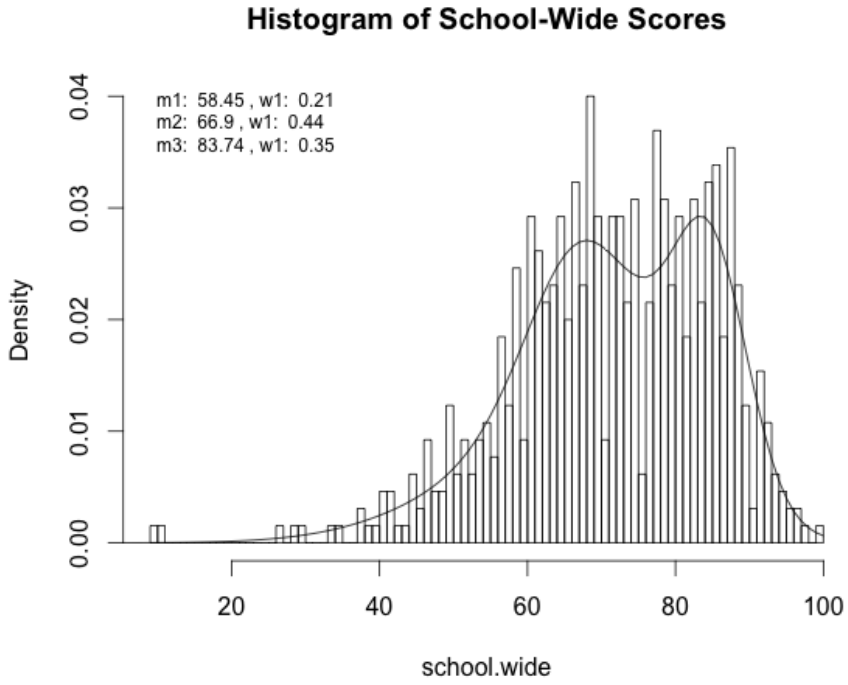
For a known group k, but unknown mean and variance (use sensible priors), write posterior distribution.
See page 4 of handwritten notes.

1.4 Gibbs Sampler Results




Sample Data
x
1:650
0
20
40
60
80
100
0
200
400
600

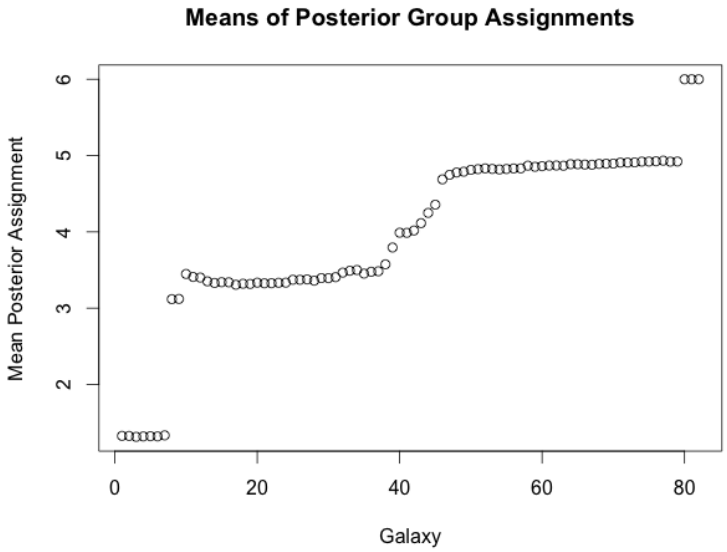

Posterior Group Means
Group
Score
0
20
40
60
80
100
1
2
3

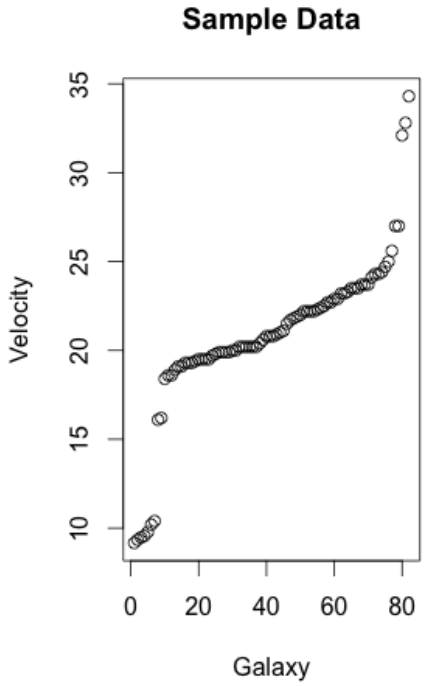

Histogram of School-Wide Scores
m1: 58.45 , w1: 0.21
m2: 66.9 , w1: 0.44
m3: 83.74 , w1: 0.35
Density
school.wide
0.00
0.01
0.02
0.03
0.04
20
40
60
80
100

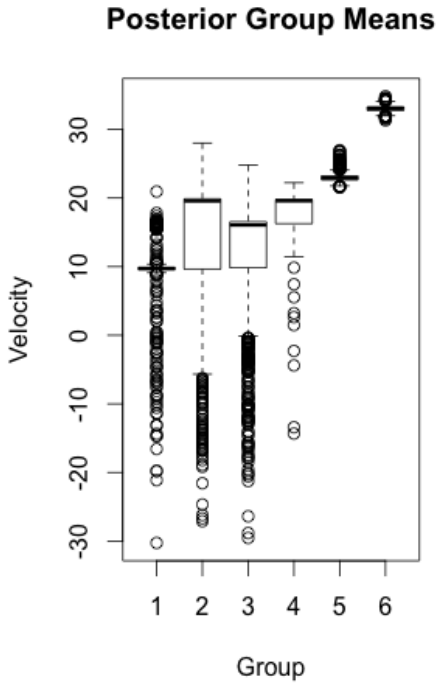
2 Galaxy Data Set

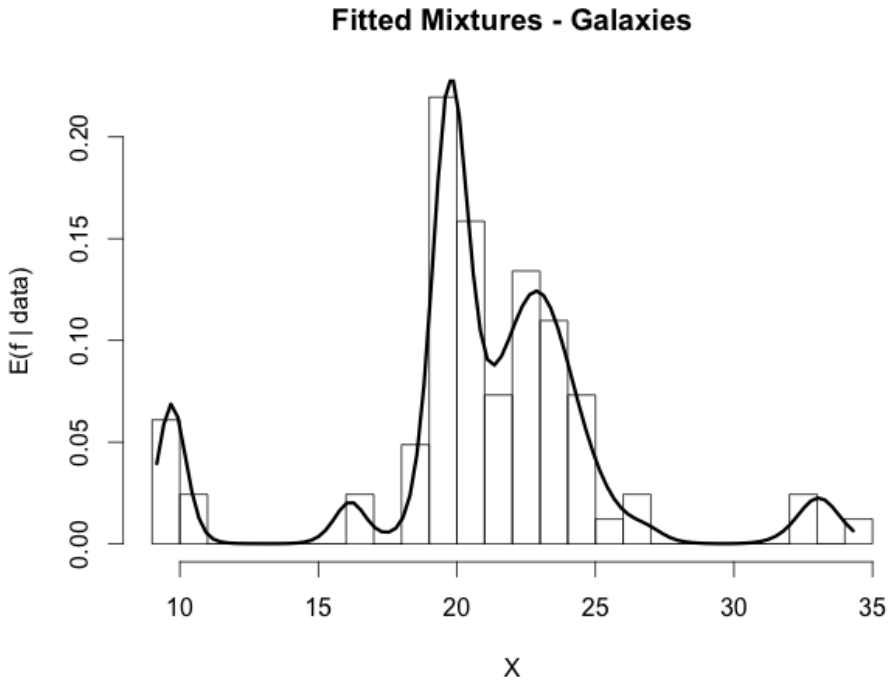
Run similar analysis on galaxy data set, with $J = 6$.

2.1 Gibbs Sampler Results









3 Full R Code

```
# Gibbs Sampler
require("gtools") ## for generating from a Dirichlet distribution
require("coda") ## for convergence diagonstics
#x <- scan("galaxies.dta")
x <- as.matrix(read.csv("galaxies.csv" ,header=F))
n <- length(x)
J <- 6
xgrid <- seq(from=min(x),to=max(x),length=100)
## hyperparameters for Mu.
m0 <- 20
v0 <- 100
## hyperparameters for (1/sig2).
# 1/sig2 ~ Ga(4, 1) s.t. E(1/sig2) = 4... By guessing that sig=0.5; sig2=0.25;
# a/b = 1/0.25 = 4/1.
a0 <- 4
b0 <- 1
## hyperparameters for dirichlet weights.
alpha <- 1

sample.s <- function(w, mu, sig2) {
  ## sample s[i] from p(s[i] | ...)
  n <- length(x)
```

```
  sd <- sqrt(sig2)
  s <- rep(0,n) # initialize
  for(i in 1:n){
    pr <- w*dnorm(x[i], m=mu, sd=sd)
    s[i] <- sample(1:J, 1, replace=T, prob=pr)
  }
  return(s)
}

sample.mu <- function(s, w, sig2) {
  ## sample mu[j]
  mu <- rep(0, J) # initialize
  for (j in 1:J) {
    Aj <- which(s==j) # makes vector of indices
    nj <- length(Aj)
    sig2.j <- sig2[j]
    if (nj==0) {
      m <- 0; V <- v0;
    } else {
      xbar <- mean(x[Aj])
      V <- 1/(1/v0 + nj/sig2.j)
      m <- V*(m0/v0 + xbar*nj/sig2.j)
    }
    mu[j] <- rnorm(1, m=m, sd=sqrt(V))
  }
  return(mu)
}

sample.w <- function(s) {
  ## sample w
  a <- rep(0,J) # initialize
  for(j in 1:J) {
    Aj <- which(s==j)
    nj <- length(Aj)
    a[j] <- alpha+nj
  }
  w <- rdirichlet(1, a)
  w <- c(w)
  return(w)
}

sample.sig <- function(s, mu) {
  ## sample sig2
  sig2 <- rep(0, J)
  for (j in 1:J) {
    Aj <- which(s==j) # makes vector of indices
    nj <- length(Aj)
```

```
    mu.j <- mu[j]
    if (nj==0) {
      a1 <- a0; b1 <- b0;
    } else {
      S2 <- sum((x[Aj]-mu.j)^2)
      a1 <- a0 + nj/2
      b1 <- b0 + S2/2
    }
    sig2[j] <- 1.0/rgamma(1, shape=a1, rate=b1)
  }
  return(sig2)
}

f <- function(xi, w, mu, sig2) {
  y <- 0
  for (j in 1:J){
    y <- y+w[j]*dnorm(xi, m=mu[j], sd=sqrt(sig2[j]))
  }
  return(y)
}

init <- function() {
  # use some exploratory data analysis for initial values of the parameters
  # initialize s with a hierarchical tree, cut for J clusters
  hc <- hclust(dist(x), "ave")
  s <- cutree(hc,k=J)
  mu <- rnorm(J, m=m0, sd=sqrt(v0))
  sig2 <- 1/rgamma(J, a0, b0)
  w <- rdirichlet(1, rep(alpha, J))

  return(th=list(mu=mu, sig2=sig2, w=w, s=s))
}

gibbs <- function(n.iter=1000) {
  ## initialize the parameters
  th <- init()
  s <- th$s; mu <- th$mu; sig2 <- th$sig2; w <- th$w
  ## set up lists to save simulations
  slist <- NULL
  mlist <- NULL
  flist <- NULL
  siglist <- NULL
  wlist <- NULL

  for(iter in 1:n.iter){
    s <- sample.s(w, mu, sig2)
    mu <- sample.mu(s, w, sig2)
```

```
    w <- sample.w(s)
    sig2 <- sample.sig(s, mu)
    ## save summaries of current simulation
    slist <- rbind(slist, s)
    mlist <- rbind(mlist, mu)
    siglist <- c(siglist, sig2)
    wlist <- rbind(wlist,w)
    flist <- rbind(flist, f(xgrid, w, mu, sig2))
  }
  return(list(s=slist, m=mlist, f=flist, sig=siglist, w=wlist))
}

# GET RESULTS
dev.off()
results <- gibbs(n.iter=2000)
slist <- results$s
mlist <- results$m
flist <- results$f
siglist <- results$sig
wlist <- results$w

# Fitted Mixtures Plot
plt.f <- function(flist, col=1, lty=1, add=F) {
  fbar <- apply(flist,2,mean)
  if (!add) {
    hist(x, bty="l", xlab="X", ylab="E(f | data)",
         main="Fitted Mixtures - Galaxies", prob=T,
         breaks=30)
  }
  lines(xgrid, fbar, type="l", lwd=3, col=col, lty=lty)
}
plt.f(flist)

# PLOT RESULTS
plot(1:n, colMeans(slist), main="Means of Posterior Group Assignments",
     xlab="Galaxy", ylab="Mean Posterior Assignment")
par(mfrow=c(1,2))
plot(1:length(x), x, main="Sample Data",
     xlab="Galaxy", ylab="Velocity")
o <- order(colMeans(mlist))
boxplot(tail(mlist[,o],1000), main="Posterior Group Means",
        xlab="Group", ylab="Velocity")
```