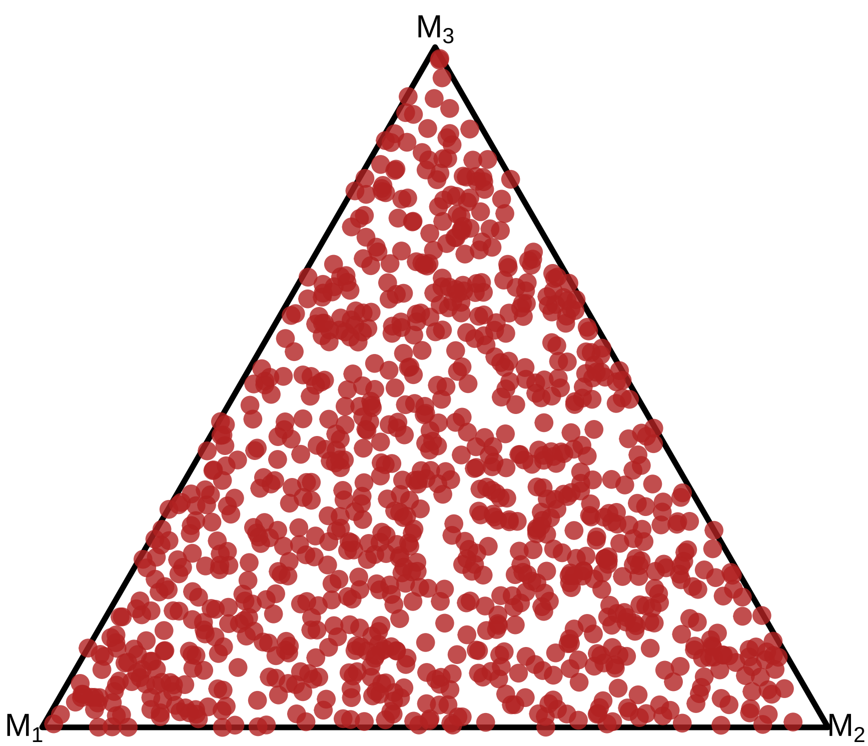
Finite mixtures, Dirichlet processes and (dependent) Dirichlet process mixtures

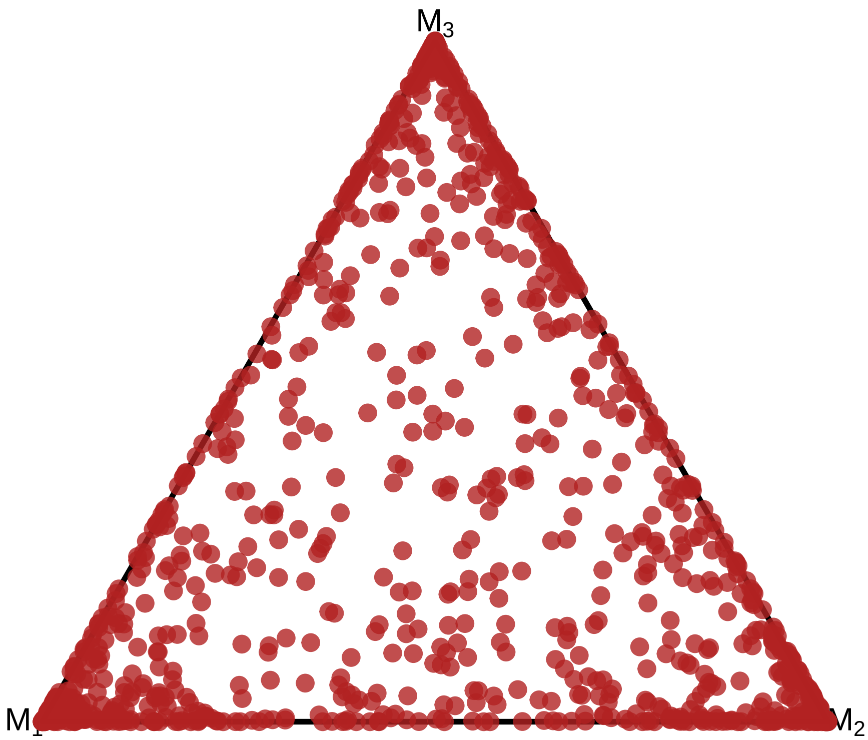
2023-12-12

Reproduce plots from the Dirichlet distribution

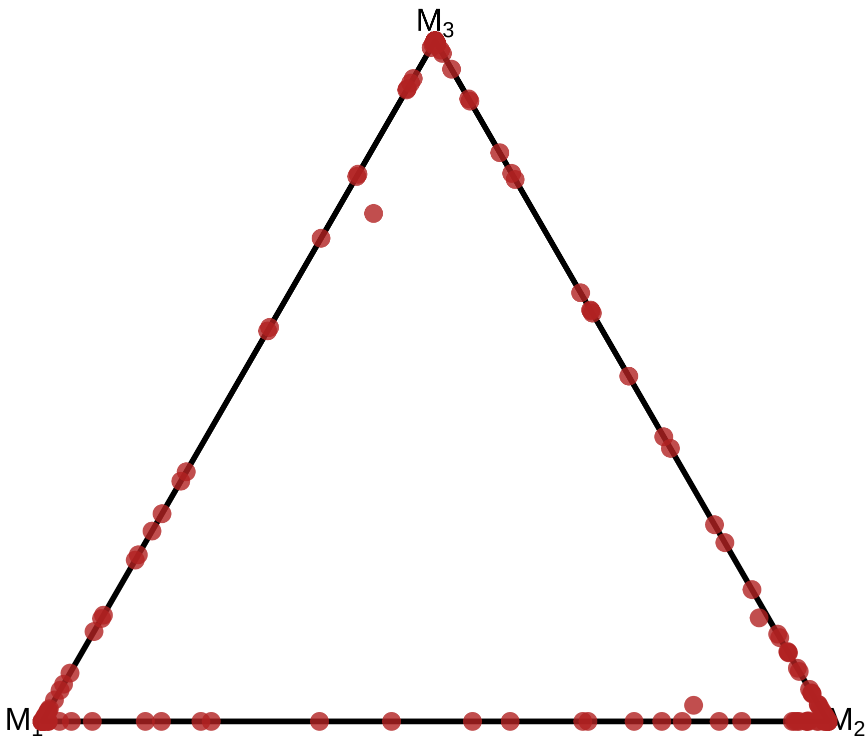
```
require(MCMCpack)  #to simulate random numbers from a Dirichlet distribution
require(ggsimplex)  #plots on the simplex
require(ggplot2)

data <- rdirichlet(n = 1000, alpha = c(1, 1, 1))
data <- as.data.frame(data)
colnames(data) <- c("pmp_1", "pmp_2", "pmp_3")

data$pmp <- with(data, make_list_column(pmp_1, pmp_2, pmp_3))

ggplot() + coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0,
    1)) + theme_void() + geom_simplex_canvas() + geom_simplex_point(data = data,
    aes(pmp = pmp), size = 0.7, color = "firebrick", alpha = 0.8)
```



```r
data <- rdirichlet(n = 1000, alpha = c(0.3, 0.3, 0.3))
data <- as.data.frame(data)
colnames(data) <- c("pmp_1", "pmp_2", "pmp_3")

data$pmp <- with(data, make_list_column(pmp_1, pmp_2, pmp_3))

ggplot() + coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0,
    1)) + theme_void() + geom_simplex_canvas() + geom_simplex_point(data = data,
    aes(pmp = pmp), size = 0.7, color = "firebrick", alpha = 0.8)
```




```r
data <- rdirichlet(n = 1000, alpha = c(0.01, 0.01, 0.01))
data <- as.data.frame(data)
colnames(data) <- c("pmp_1", "pmp_2", "pmp_3")

data$pmp <- with(data, make_list_column(pmp_1, pmp_2, pmp_3))

ggplot() + coord_fixed(ratio = 1, xlim = c(0, 1), ylim = c(0,
    1)) + theme_void() + geom_simplex_canvas() + geom_simplex_point(data = data,
    aes(pmp = pmp), size = 0.7, color = "firebrick", alpha = 0.8)
```



Finite mixture model

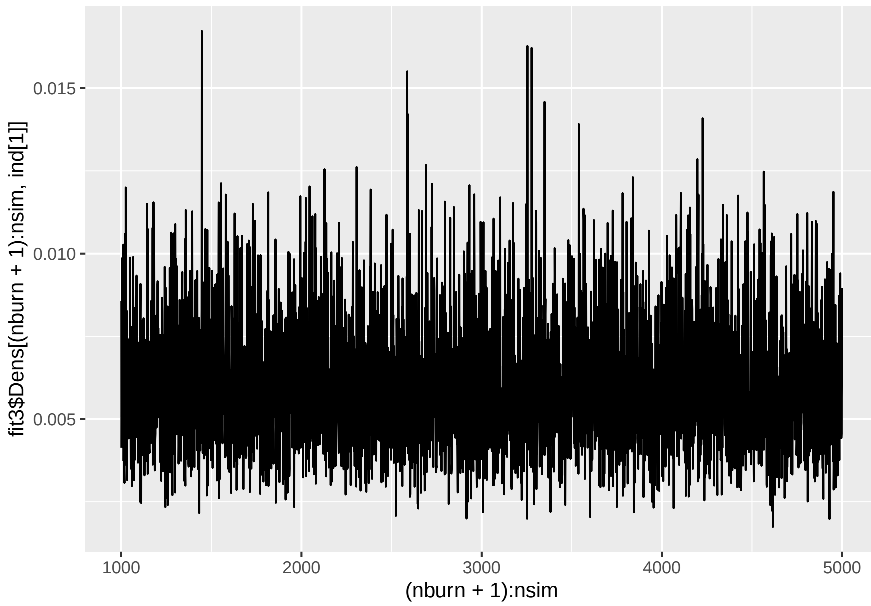
Simulated data example

```
require(mixtools)  #to simulate data from mixture of normals
require(MCMCpack)  #to simulate random numbers from a Dirichlet distribution
require(coda)

# function implementing a location scale mixture of K
# (K>=2) normal dists
fmm <- function(y, grid, K, amu, b2mu, asigma2, bsigma2, alpha,
    nsim) {
    n <- length(y)
    ngrid <- length(grid)

    prop <- prob <- matrix(0, nrow = n, ncol = K)
    P <- Mu <- Sigma2 <- matrix(0, nrow = nsim, ncol = K)
    Dens <- array(0, c(nsim, ngrid, K))
    Densm <- matrix(0, nrow = nsim, ncol = ngrid)

    z <- rep(1, n)
    ns <- rep(0, K)

    P[1, ] <- rdirichlet(1, alpha)
    Mu[1, ] <- rep(mean(y), K)
    Sigma2[1, ] <- rep(var(y), K)
```

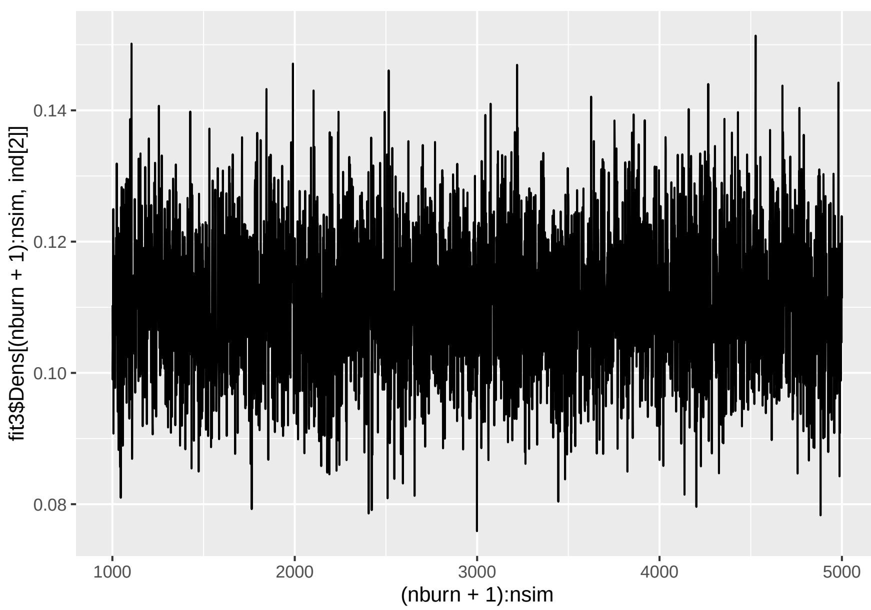
```r
    for (i in 2:nsim) {

        for (k in 1:K) {
            prop[, k] <- P[i - 1, k] * dnorm(y, mean = Mu[i -
                1, k], sd = sqrt(Sigma2[i - 1, k]))
        }

        prob <- prop/apply(prop, 1, sum)

        for (l in 1:n) {
            z[l] = sample(1:K, size = 1, prob = prob[l, ])
        }

        P[i, ] <- rdirichlet(1, alpha + tabulate(z, nbins = K))

        for (k in 1:K) {
            ns[k] <- length(which(z == k))
        }

        for (k in 1:K) {
            varmu <- 1/((1/b2mu) + (ns[k]/Sigma2[i - 1, k]))
            meanmu <- ((sum(y[z == k])/Sigma2[i - 1, k]) + (amu/b2mu))/((1/b2mu) +
                (ns[k]/Sigma2[i - 1, k]))
            Mu[i, k] <- rnorm(1, mean = meanmu, sd = sqrt(varmu))

            Sigma2[i, k] <- 1/rgamma(1, asigma2 + (ns[k]/2),
                bsigma2 + 0.5 * sum((y[z == k] - Mu[i, k])^2))

            Dens[i, , k] <- P[i, k] * dnorm(grid, Mu[i, k], sqrt(Sigma2[i,
                k]))
        }

        for (j in 1:ngrid) {
            Densm[i, j] <- sum(Dens[i, j, ])
        }
    }
    return(list(P = P, Mu = Mu, Sigma2 = Sigma2, Dens = Densm))
}

# simulating data and defining grid where to evaluate the
# density
n <- 500
set.seed(123)
y <- rnormmix(n, c(0.3, 0.3, 0.4), c(-6, 0, 6), c(1, 1, 1))
grid <- seq(min(y) - 1, max(y) + 1, len = 200)
ngrid <- length(grid)
nsim <- 5000

# fitting the model for K=2, 3, 4, 5, 20, and 50
set.seed(123)
fit2 <- fmm(y = y, grid = grid, K = 2, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 2), nsim = nsim)
```

```r
set.seed(123)
fit3 <- fmm(y = y, grid = grid, K = 3, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 3), nsim = nsim)

set.seed(123)
fit4 <- fmm(y = y, grid = grid, K = 4, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 4), nsim = nsim)

set.seed(123)
fit5 <- fmm(y = y, grid = grid, K = 5, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 5), nsim = nsim)

set.seed(123)
fit20 <- fmm(y = y, grid = grid, K = 20, amu = 0, b2mu = 100,
    asigma2 = 0.1, bsigma2 = 0.1, alpha = rep(1, 20), nsim = nsim)

set.seed(123)
fit50 <- fmm(y = y, grid = grid, K = 50, amu = 0, b2mu = 100,
    asigma2 = 0.1, bsigma2 = 0.1, alpha = rep(1, 50), nsim = nsim)

set.seed(123)
fit_overfitted <- fmm(y = y, grid = grid, K = 50, amu = 0, b2mu = 100,
    asigma2 = 0.1, bsigma2 = 0.1, alpha = rep(0.1, 50), nsim = nsim)

# looking at the traceplots of the estimated densities (the
# quantities that we care about) for sake of space only for
# K=3
nburn <- 1000
ind <- sample(1:ngrid, 5, replace = TRUE)
qplot((nburn + 1):nsim, fit3$Dens[(nburn + 1):nsim, ind[1]],
    geom = "line")
```

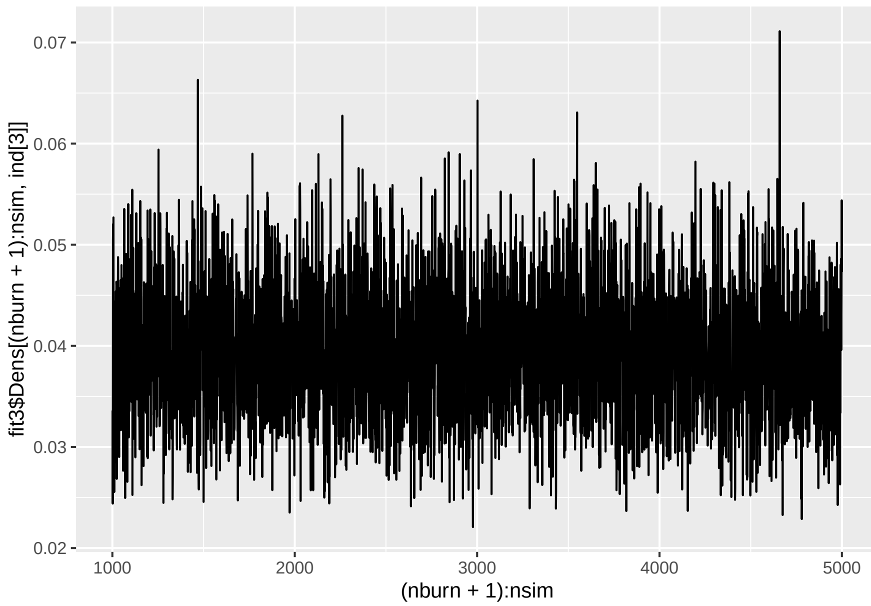
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



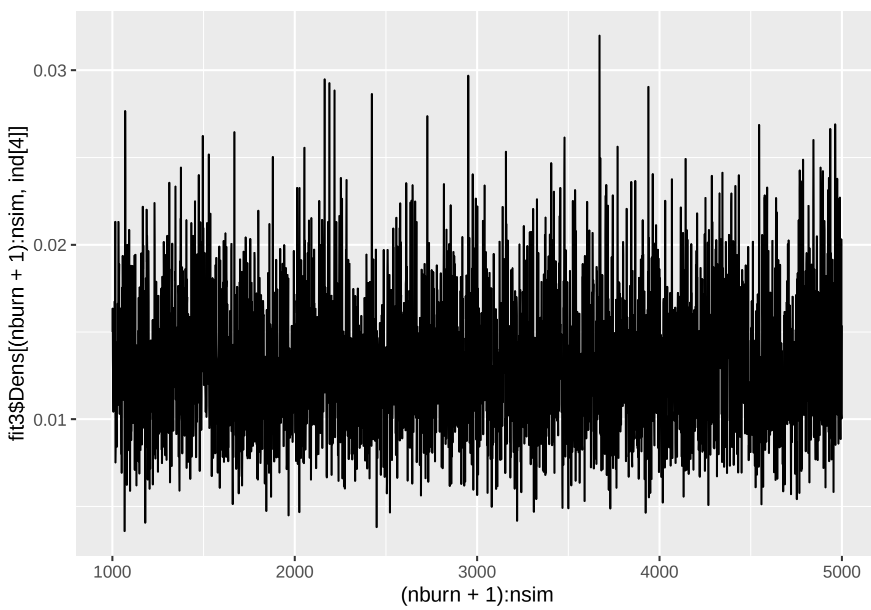

```
qplot((nburn + 1):nsim, fit3$Dens[(nburn + 1):nsim, ind[2]],
    geom = "line")
```



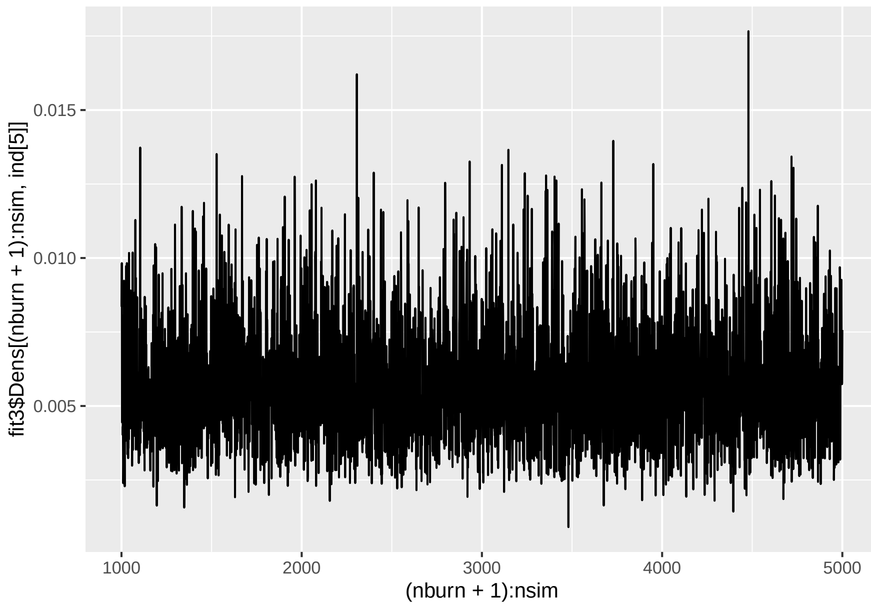

```
qplot((nburn + 1):nsim, fit3$Dens[(nburn + 1):nsim, ind[3]],
    geom = "line")
```




```
qplot((nburn + 1):nsim, fit3$Dens[(nburn + 1):nsim, ind[4]],
    geom = "line")
```




```
qplot((nburn + 1):nsim, fit3$Dens[(nburn + 1):nsim, ind[5]],
    geom = "line")
```




```
effectiveSize(fit3$Dens[(nburn + 1):nsim, ind[1]])
```

```
##     var1
## 3576.282
```

```
geweke.diag(fit3$Dens[(nburn + 1):nsim, ind[1]])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.2552
```

```
effectiveSize(fit3$Dens[(nburn + 1):nsim, ind[2]])
```

```
##     var1
## 3126.878
```

```
geweke.diag(fit3$Dens[(nburn + 1):nsim, ind[2]])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## -0.4439
```

```
effectiveSize(fit3$Dens[(nburn + 1):nsim, ind[3]])
```

```
##     var1
## 2970.933
```

```r
geweke.diag(fit3$Dens[(nburn + 1):nsim, ind[3]])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## -0.0566
```

```r
effectiveSize(fit3$Dens[(nburn + 1):nsim, ind[4]])
```

```
##     var1
## 2847.535
```

```r
geweke.diag(fit3$Dens[(nburn + 1):nsim, ind[4]])
```

```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.7302
```

```r
effectiveSize(fit3$Dens[(nburn + 1):nsim, ind[5]])
```

```
##     var1
## 3712.453
```

```r
geweke.diag(fit3$Dens[(nburn + 1):nsim, ind[5]])
```

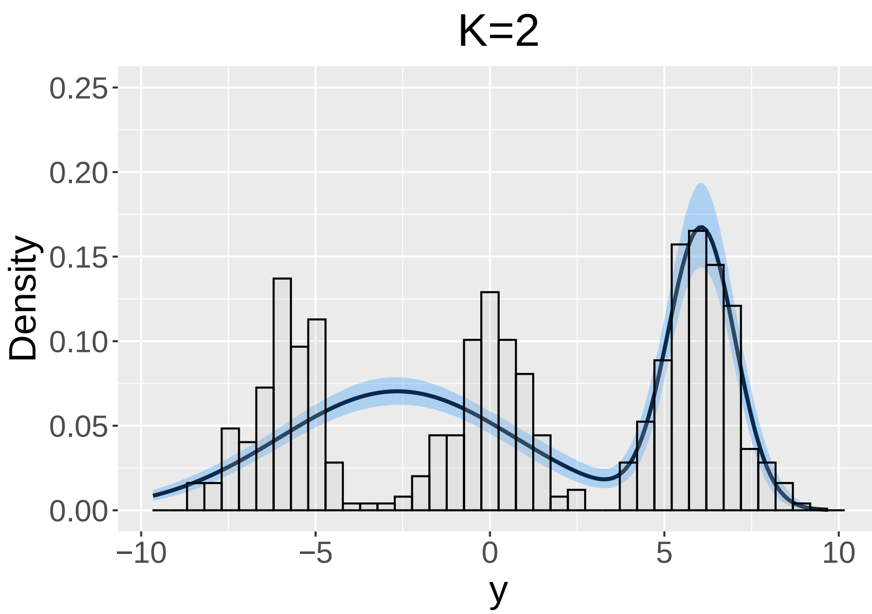
```
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##  var1
## -1.95
```

```r
# estimated densities
dens2m <- apply(fit2$Dens[(nburn + 1):nsim, ], 2, mean)
dens2l <- apply(fit2$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens2h <- apply(fit2$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens3m <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, mean)
dens3l <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens3h <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens4m <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, mean)
dens4l <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens4h <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)
```

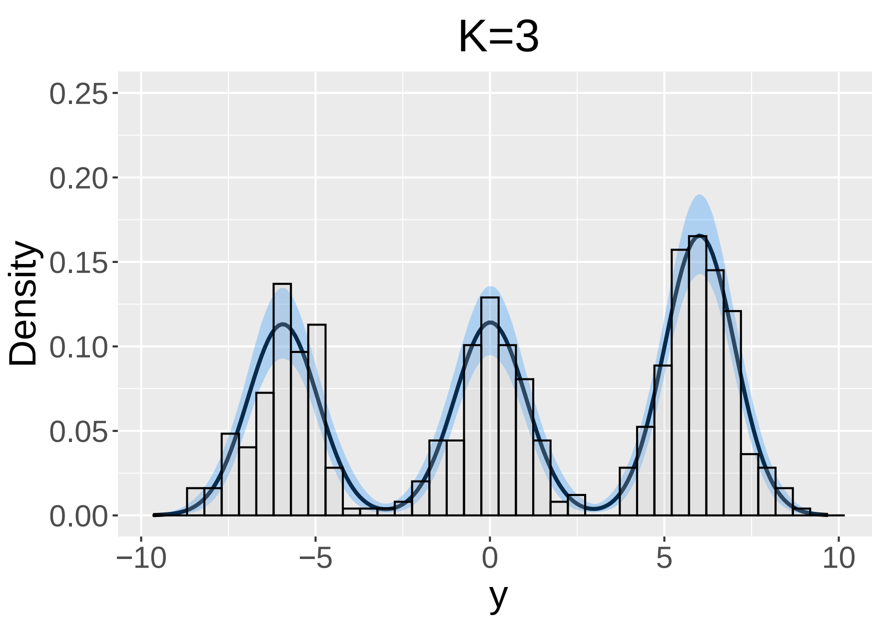
```r
dens5m <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, mean)
dens5l <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens5h <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens20m <- apply(fit20$Dens[(nburn + 1):nsim, ], 2, mean)
dens20l <- apply(fit20$Dens[(nburn + 1):nsim, ], 2, quantile,
    prob = 0.025)
dens20h <- apply(fit20$Dens[(nburn + 1):nsim, ], 2, quantile,
    prob = 0.975)

dens50m <- apply(fit50$Dens[(nburn + 1):nsim, ], 2, mean)
dens50l <- apply(fit50$Dens[(nburn + 1):nsim, ], 2, quantile,
    prob = 0.025)
dens50h <- apply(fit50$Dens[(nburn + 1):nsim, ], 2, quantile,
    prob = 0.975)

densoverfittedm <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, mean)
densoverfittedl <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, quantile, prob = 0.025)
densoverfittedh <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, quantile, prob = 0.975)


dfhist <- data.frame(y = y)

dfdens2 <- data.frame(dm = dens2m, dl = dens2l, dh = dens2h,
    seqgrid = grid)
dfdens3 <- data.frame(dm = dens3m, dl = dens3l, dh = dens3h,
    seqgrid = grid)
dfdens4 <- data.frame(dm = dens4m, dl = dens4l, dh = dens4h,
    seqgrid = grid)
dfdens5 <- data.frame(dm = dens5m, dl = dens5l, dh = dens5h,
    seqgrid = grid)
dfdens20 <- data.frame(dm = dens20m, dl = dens20l, dh = dens20h,
    seqgrid = grid)
dfdens50 <- data.frame(dm = dens50m, dl = dens50l, dh = dens50h,
    seqgrid = grid)
dfdensoverfitted <- data.frame(dm = densoverfittedm, dl = densoverfittedl,
    dh = densoverfittedh, seqgrid = grid)

ggplot(dfdens2, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens2, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("y") + ylab("Density") +
    ylim(0, 0.25) + geom_histogram(data = dfhist, aes(x = y,
    y = after_stat(density)), alpha = 0.2, bins = 40, inherit.aes = FALSE,
    fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=2") + theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.
```

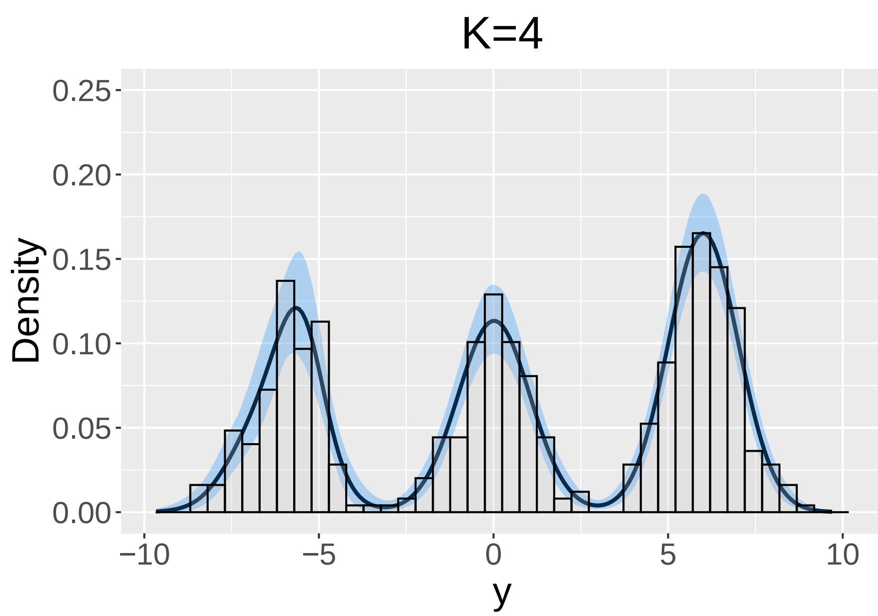



```
ggplot(dfdens3, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens3, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("y") + ylab("Density") +
    ylim(0, 0.25) + geom_histogram(data = dfhist, aes(x = y,
    y = after_stat(density)), alpha = 0.2, bins = 40, inherit.aes = FALSE,
    fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=3") + theme(plot.title = element_text(hjust = 0.5))
```

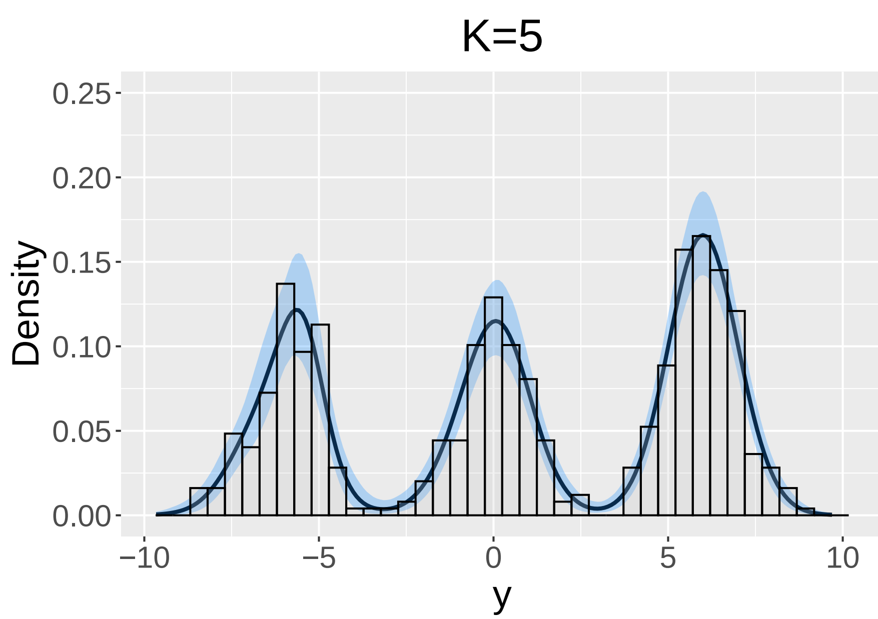



```
ggplot(dfdens4, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens4, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("y") + ylab("Density") +
```

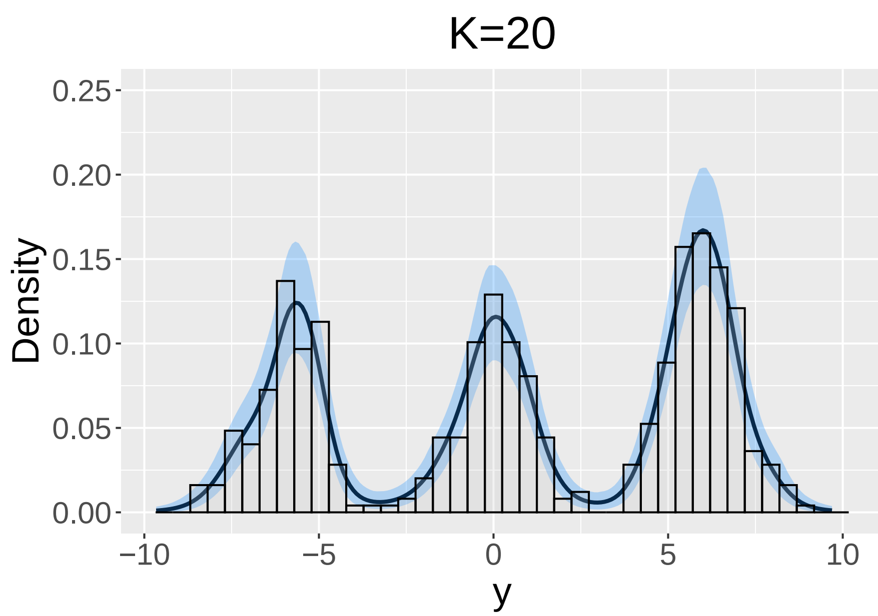
```r
    ylim(0, 0.25) + geom_histogram(data = dfhist, aes(x = y,
    y = after_stat(density)), alpha = 0.2, bins = 40, inherit.aes = FALSE,
    fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=4") + theme(plot.title = element_text(hjust = 0.5))
```



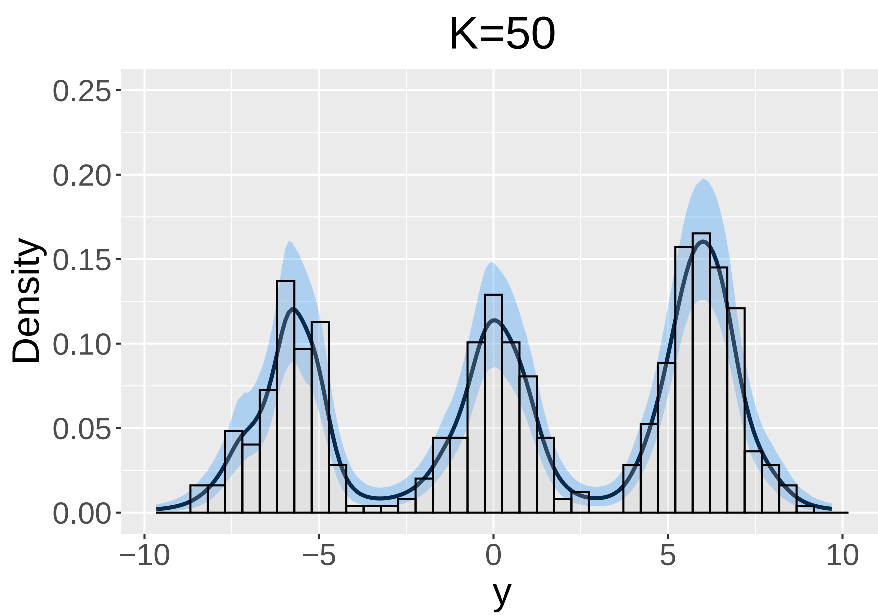

```r
ggplot(dfdens5, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens5, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("y") + ylab("Density") +
    ylim(0, 0.25) + geom_histogram(data = dfhist, aes(x = y,
    y = after_stat(density)), alpha = 0.2, bins = 40, inherit.aes = FALSE,
    fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=5") + theme(plot.title = element_text(hjust = 0.5))
```



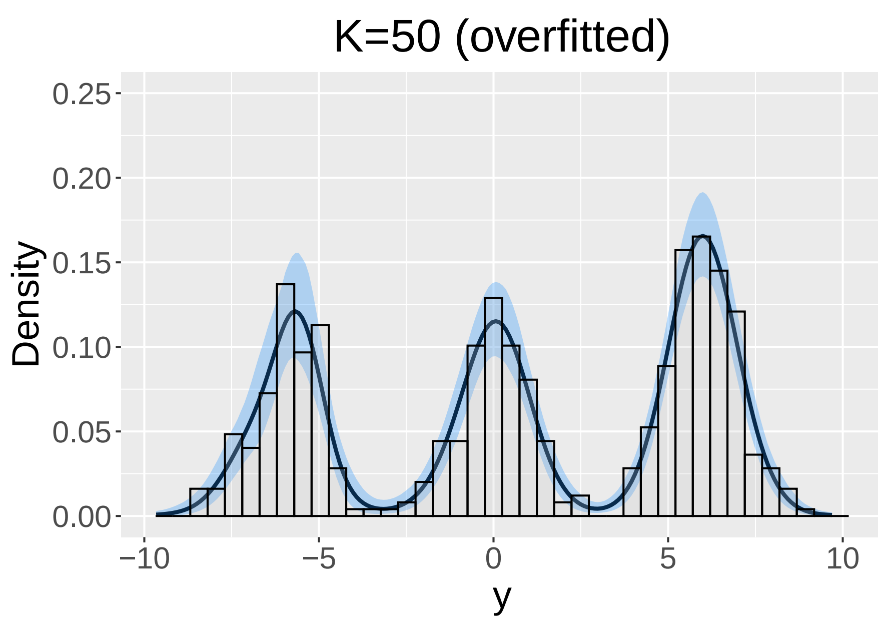
```r
ggplot(dfdens20, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens20, aes(x = seqgrid, ymin = dl,
        ymax = dh), alpha = 0.3, fill = "dodgerblue1") + xlab("y") +
    ylab("Density") + ylim(0, 0.25) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=20") + theme(plot.title = element_text(hjust = 0.5))
```




```r
ggplot(dfdens50, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens50, aes(x = seqgrid, ymin = dl,
        ymax = dh), alpha = 0.3, fill = "dodgerblue1") + xlab("y") +
    ylab("Density") + ylim(0, 0.25) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=50") + theme(plot.title = element_text(hjust = 0.5))
```




```r
ggplot(dfdensoverfitted, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdensoverfitted, aes(x = seqgrid, ymin = dl,
        ymax = dh), alpha = 0.3, fill = "dodgerblue1") + xlab("y") +
    ylab("Density") + ylim(0, 0.25) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=50 (overfitted)") + theme(plot.title = element_text(hjust = 0.5))
```




```r
# function to calculate the CPO/LPML and WAIC
good_fit_criteria <- function(y, P, Mu, Sigma2, nburn) {
    nsim <- nrow(P)
    K <- ncol(Mu)
    n <- length(y)
```

```r
    Densy <- array(0, c((nsim - nburn), n, K))
    Densym <- matrix(0, nrow = (nsim - nburn), ncol = n)

    for (i in (nburn + 1):nsim) {
        for (k in 1:K) {
            Densy[i - nburn, , k] <- P[i, k] * dnorm(y, Mu[i,
                k], sqrt(Sigma2[i, k]))
        }
        for (j in 1:n) {
            Densym[i - nburn, j] <- sum(Densy[i - nburn, j, ])
        }
    }

    cpoinv <- apply(1/Densym, 2, mean)
    cpo <- 1/cpoinv
    lpml <- sum(log(cpo))

    lpd <- sum(log(apply(exp(log(Densym)), 2, mean)))
    p2 <- sum(apply(log(Densym), 2, var))
    waic <- -2 * (lpd - p2)

    return(list(CPO = cpo, LPML = lpml, WAIC = waic))
}

ghc2 <- good_fit_criteria(y = y, P = fit2$P, Mu = fit2$Mu, Sigma2 = fit2$Sigma2,
    nburn = nburn)
ghc2$LPML
```

```
## [1] -1378.841
```

```r
ghc2$WAIC
```

```
## [1] 2757.683
```

```r
ghc3 <- good_fit_criteria(y = y, P = fit3$P, Mu = fit3$Mu, Sigma2 = fit3$Sigma2,
    nburn = nburn)
ghc3$LPML
```

```
## [1] -1260.408
```

```r
ghc3$WAIC
```

```
## [1] 2520.809
```

```r
ghc4 <- good_fit_criteria(y = y, P = fit4$P, Mu = fit4$Mu, Sigma2 = fit4$Sigma2,
    nburn = nburn)
ghc4$LPML
```

```
## [1] -1259.599
```

```r
ghc4$WAIC
```

```
## [1] 2519.197
```

```r
ghc5 <- good_fit_criteria(y = y, P = fit5$P, Mu = fit5$Mu, Sigma2 = fit5$Sigma2,
    nburn = nburn)
ghc5$LPML
```

```
## [1] -1259.899
```

```r
ghc5$WAIC
```

```
## [1] 2519.768
```

```r
ghc20 <- good_fit_criteria(y = y, P = fit20$P, Mu = fit20$Mu,
    Sigma2 = fit20$Sigma2, nburn = nburn)
ghc20$LPML
```

```
## [1] -1266.232
```

```r
ghc20$WAIC
```

```
## [1] 2532.418
```

```r
ghc50 <- good_fit_criteria(y = y, P = fit50$P, Mu = fit50$Mu,
    Sigma2 = fit50$Sigma2, nburn = nburn)
ghc50$LPML
```

```
## [1] -1278.796
```

```r
ghc50$WAIC
```

```
## [1] 2557.563
```

```r
ghcoverfitted <- good_fit_criteria(y = y, P = fit_overfitted$P,
    Mu = fit_overfitted$Mu, Sigma2 = fit_overfitted$Sigma2, nburn = nburn)
ghcoverfitted$LPML
```

```
## [1] -1263.77
```

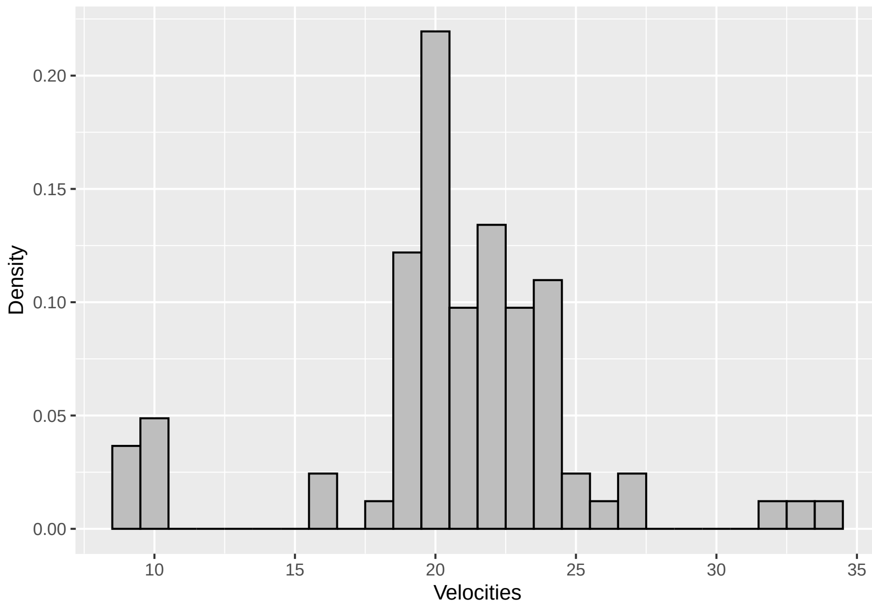
```r
ghcoverfitted$WAIC
```

```
## [1] 2527.51
```

Galaxy data example

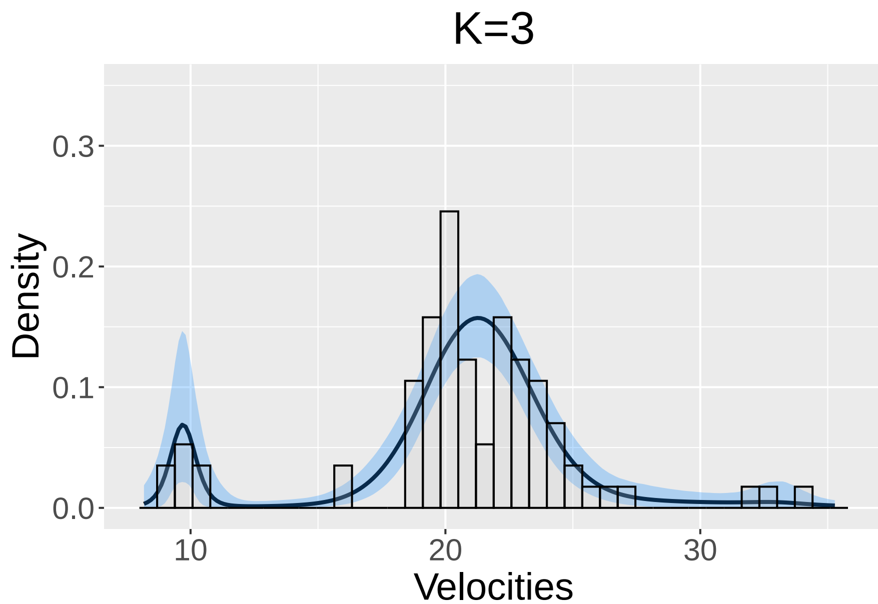
```r
require(MASS)

y <- galaxies/1000

ggplot(data.frame(y = y), aes(x = y)) + geom_histogram(aes(y = ..density..),
    binwidth = 1, fill = "gray", colour = "black") + xlab("Velocities") +
    ylab("Density")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

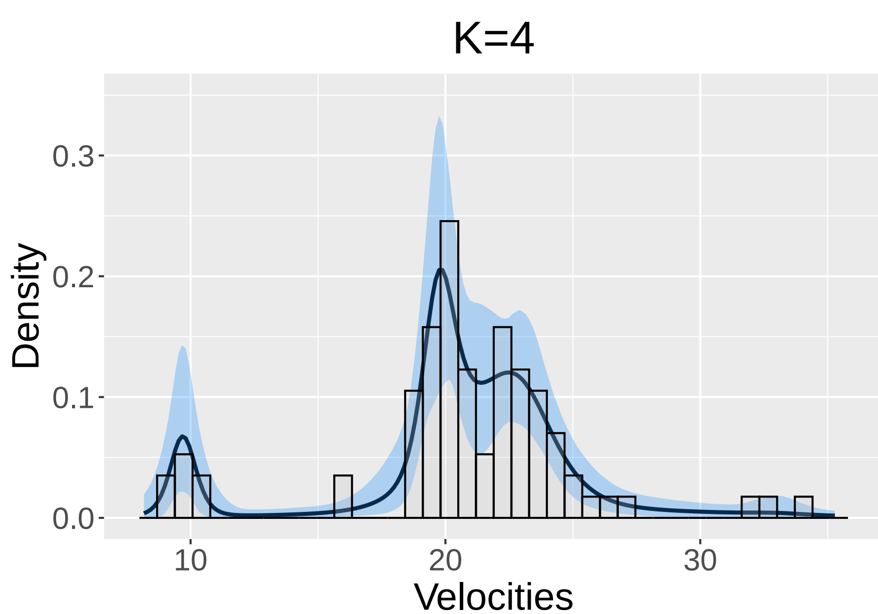



```r
grid <- seq(min(y) - 1, max(y) + 1, len = 200)
ngrid <- length(grid)

nsim <- 10000
nburn <- 1000

set.seed(123)
fit3 <- fmm(y = y, grid = grid, K = 3, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 3), nsim = nsim)

set.seed(123)
fit4 <- fmm(y = y, grid = grid, K = 4, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 4), nsim = nsim)
set.seed(123)
fit5 <- fmm(y = y, grid = grid, K = 5, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 5), nsim = nsim)

set.seed(123)
```

```
fit6 <- fmm(y = y, grid = grid, K = 6, amu = 0, b2mu = 100, asigma2 = 0.1,
    bsigma2 = 0.1, alpha = rep(1, 6), nsim = nsim)

set.seed(123)
fit_overfitted <- fmm(y = y, grid = grid, K = 50, amu = 0, b2mu = 100,
    asigma2 = 0.1, bsigma2 = 0.1, alpha = rep(0.1, 50), nsim = nsim)

dens3m <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, mean)
dens3l <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens3h <- apply(fit3$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens4m <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, mean)
dens4l <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens4h <- apply(fit4$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens5m <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, mean)
dens5l <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens5h <- apply(fit5$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

dens6m <- apply(fit6$Dens[(nburn + 1):nsim, ], 2, mean)
dens6l <- apply(fit6$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.025)
dens6h <- apply(fit6$Dens[(nburn + 1):nsim, ], 2, quantile, prob = 0.975)

densoverfittedm <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, mean)
densoverfittedl <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, quantile, prob = 0.025)
densoverfittedh <- apply(fit_overfitted$Dens[(nburn + 1):nsim,
    ], 2, quantile, prob = 0.975)

dfhist <- data.frame(y = y)

dfdens3 <- data.frame(dm = dens3m, dl = dens3l, dh = dens3h,
    seqgrid = grid)
dfdens4 <- data.frame(dm = dens4m, dl = dens4l, dh = dens4h,
    seqgrid = grid)
dfdens5 <- data.frame(dm = dens5m, dl = dens5l, dh = dens5h,
    seqgrid = grid)
dfdens6 <- data.frame(dm = dens6m, dl = dens6l, dh = dens6h,
    seqgrid = grid)
dfdensoverfitted <- data.frame(dm = densoverfittedm, dl = densoverfittedl,
    dh = densoverfittedh, seqgrid = grid)

ggplot(dfdens3, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens3, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=3") + theme(plot.title = element_text(hjust = 0.5))
```

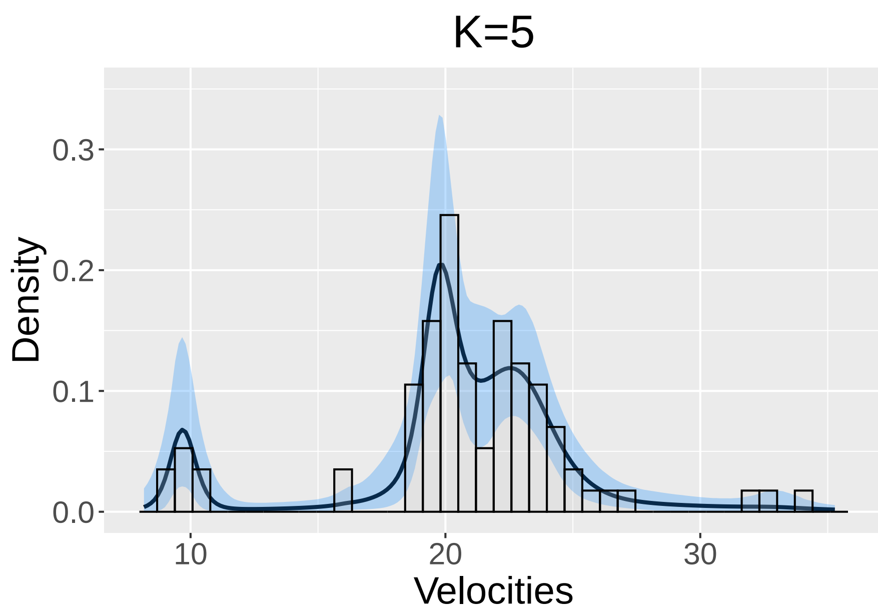



```r
ggplot(dfdens4, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens4, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=4") + theme(plot.title = element_text(hjust = 0.5))
```

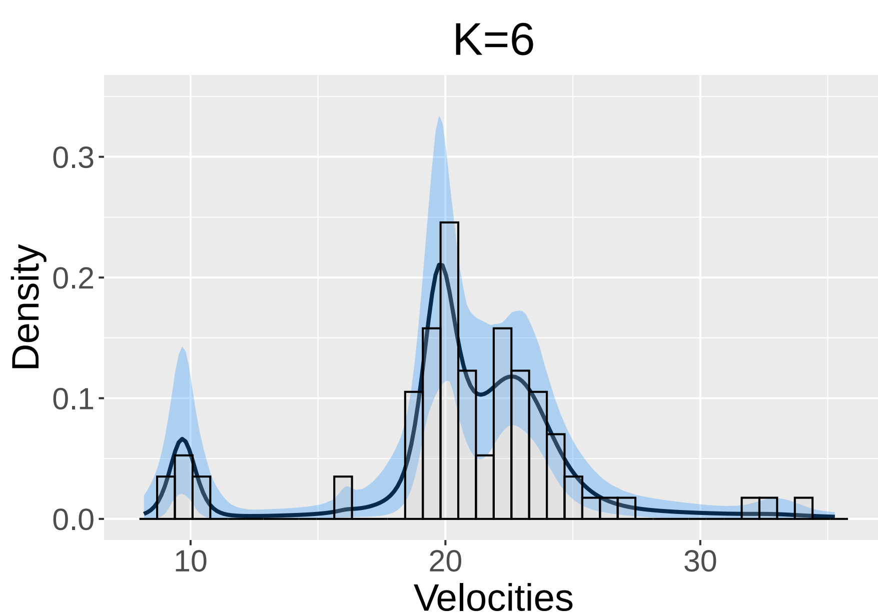



```r
ggplot(dfdens5, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens5, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
```

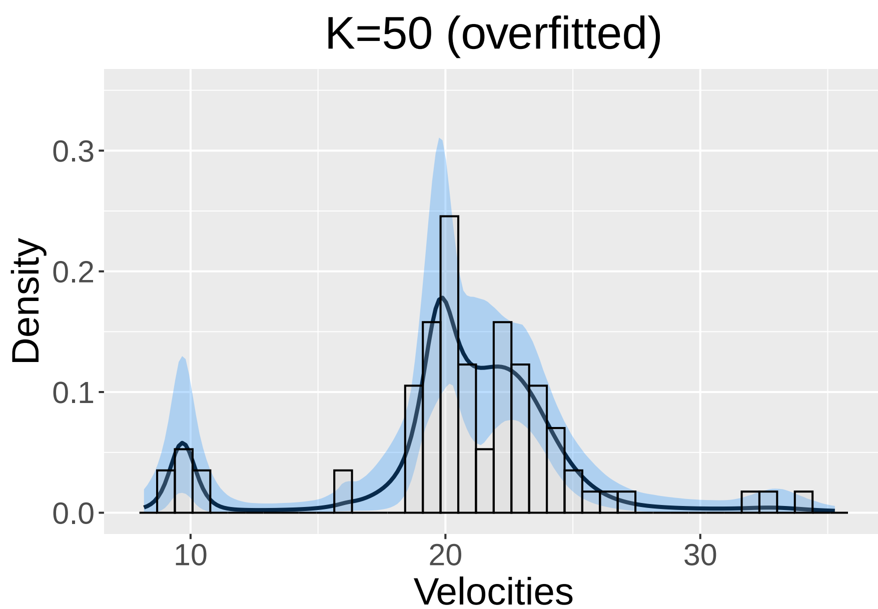
```
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=5") + theme(plot.title = element_text(hjust = 0.5))
```



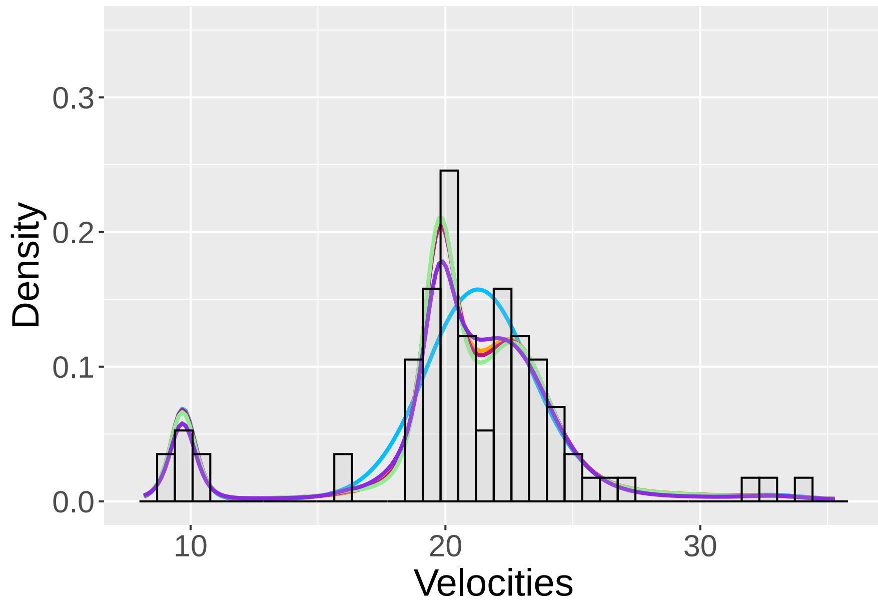

```
ggplot(dfdens6, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens6, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=6") + theme(plot.title = element_text(hjust = 0.5))
```



```r
ggplot(dfdensoverfitted, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdensoverfitted, aes(x = seqgrid, ymin = dl,
        ymax = dh), alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=50 (overfitted)") + theme(plot.title = element_text(hjust = 0.5))
```




```r
dfj <- data.frame(dm3 = dens3m, dm4 = dens4m, dm5 = dens5m, dm6 = dens6m,
    dmover = densoverfittedm, seqgrid = grid)

ggplot(dfj, aes(x = seqgrid, y = dm3)) + geom_line(size = 1,
    color = "deepskyblue1") + geom_line(aes(y = dm4), size = 1,
    color = "orange") + geom_line(aes(y = dm5), size = 1, color = "deeppink3") +
    geom_line(aes(y = dm6), size = 1, color = "lightgreen") +
    geom_line(aes(y = dmover), size = 1, color = "blueviolet") +
    xlab("Velocities") + ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20))
```




```r
ghc3 <- good_fit_criteria(y = y, P = fit3$P, Mu = fit3$Mu, Sigma2 = fit3$Sigma2,
    nburn = nburn)
ghc3$LPML
```

```
## [1] -214.3474
```

```r
ghc3$WAIC
```

```
## [1] 428.1348
```

```r
ghc4 <- good_fit_criteria(y = y, P = fit4$P, Mu = fit4$Mu, Sigma2 = fit4$Sigma2,
    nburn = nburn)
ghc4$LPML
```

```
## [1] -212.1646
```

```r
ghc4$WAIC
```

```
## [1] 423.9259
```

```r
ghc5 <- good_fit_criteria(y = y, P = fit5$P, Mu = fit5$Mu, Sigma2 = fit5$Sigma2,
    nburn = nburn)
ghc5$LPML
```

```
## [1] -212.8018
```

```r
ghc5$WAIC
```

```
## [1] 425.0935
```

```r
ghc6 <- good_fit_criteria(y = y, P = fit6$P, Mu = fit6$Mu, Sigma2 = fit6$Sigma2,
    nburn = nburn)
ghc6$LPML
```

```
## [1] -213.0878
```

```r
ghc6$WAIC
```

```
## [1] 425.71
```

```r
ghcoverfitted <- good_fit_criteria(y = y, P = fit_overfitted$P,
    Mu = fit_overfitted$Mu, Sigma2 = fit_overfitted$Sigma2, nburn = nburn)
ghcoverfitted$LPML
```

```
## [1] -218.3053
```

```r
ghcoverfitted$WAIC
```

```
## [1] 436.2959
```

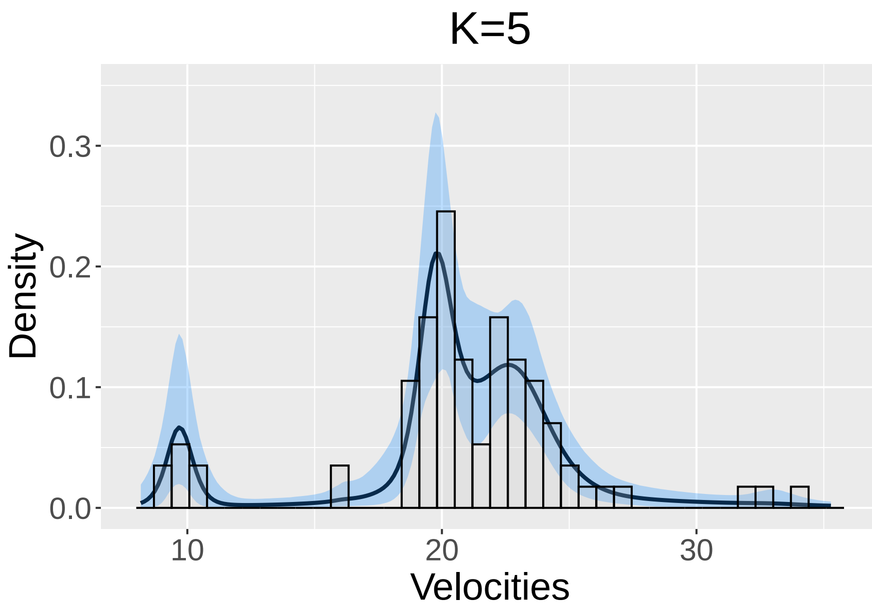
Finite mixture model in JAGS

The below code shows how to implement exactly the same mixture model but using JAGS. it has the advantage that one can be much more flexible in the prior distributions (or the mixture kernel), without having to recode everything. Once we extract the parameters, everything proceeds as before.

```r
require(rjags)

model_string <- "model{
for(i in 1:n){
y[i]~dnorm(mu[z[i]],tau[z[i]])
z[i]~dcat(p[])
}

p[1:K]~ddirch(alpha)

for(k in 1:K){
mu[k]~dnorm(0,0.01)
tau[k]~dgamma(0.1,0.1)
sigma2[k] <- 1/tau[k]
}
}"

K <- 5
alpha <- rep(1, K)
data <- list(n = length(y), y = y, K = K, alpha = alpha)
model <- jags.model(textConnection(model_string), n.chains = 1,
    data = data, n.adapt = 2000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 82
##    Unobserved stochastic nodes: 93
##    Total graph size: 356
##
## Initializing model
```

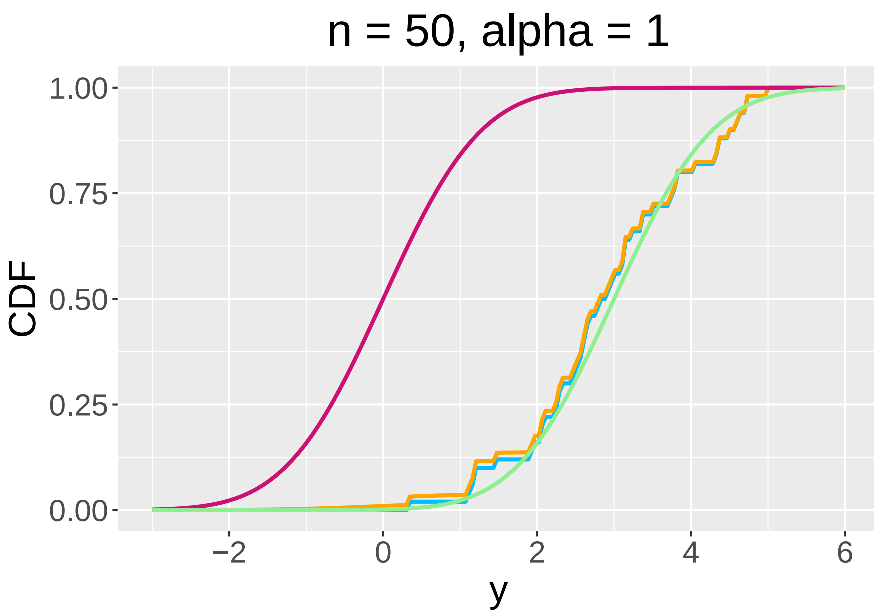
```
update(model, nburn)
res <- jags.samples(model, variable.names = c("mu", "sigma2",
    "p"), n.iter = nsim)

# extracting the parameters
P <- Mu <- Sigma2 <- matrix(0, nrow = nsim, ncol = K)
for (k in 1:K) {
    Mu[, k] <- res$mu[k, , 1]
    Sigma2[, k] <- res$sigma2[k, , 1]
    P[, k] <- res$p[k, , 1]
}

# computing the density (as we did before inside the fmm
# function)
dens <- array(0, c(nsim, ngrid, K))
dens1 <- matrix(0, nrow = nsim, ncol = ngrid)

for (i in 1:nsim) {
    for (k in 1:K) {
        dens[i, , k] <- P[i, k] * dnorm(grid, Mu[i, k], sqrt(Sigma2[i,
            k]))
    }
    for (j in 1:ngrid) {
        dens1[i, j] <- sum(dens[i, j, ])
    }
}

densm <- apply(dens1, 2, mean)
densl <- apply(dens1, 2, quantile, prob = 0.025)
densh <- apply(dens1, 2, quantile, prob = 0.975)

dfdens <- data.frame(dm = densm, dl = densl, dh = densh, seqgrid = grid)

ggplot(dfdens, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + xlab("Velocities") +
    ylab("Density") + ylim(0, 0.35) + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + theme(text = element_text(size = 20)) +
    ggtitle("K=5") + theme(plot.title = element_text(hjust = 0.5))
```



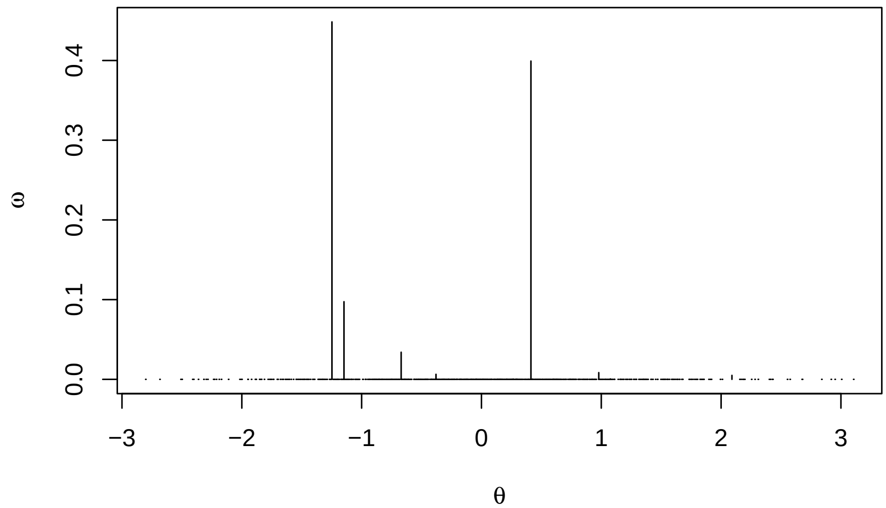
Dirichlet process prior

DP conjugacy – posterior mean

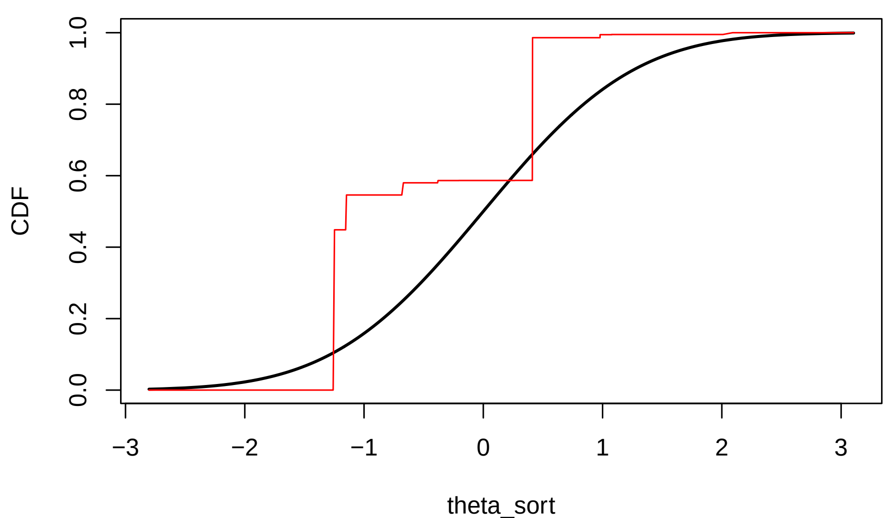
```
n <- 50
y <- rnorm(n, 3, 1)
alpha <- 1
Femp <- ecdf(y)
grid <- seq(-3, 6, len = 200)
ngrid <- length(grid)
post_mean <- numeric(ngrid)

for (j in 1:ngrid) {
    post_mean[j] <- (alpha/(alpha + n)) * pnorm(grid[j], 0, 1) +
        (n/(alpha + n)) * Femp(grid[j])
}

df <- data.frame(demp = Femp(grid), pm = post_mean, seqgrid = grid)

ggplot(df, aes(x = seqgrid, y = demp)) + geom_line(size = 1,
    color = "deepskyblue1") + geom_line(aes(y = pm), size = 1,
    color = "orange") + stat_function(fun = pnorm, args = list(mean = 0,
    sd = 1), size = 1, color = "deeppink3") + stat_function(fun = pnorm,
    args = list(mean = 3, sd = 1), size = 1, color = "lightgreen") +
    ylab("CDF") + xlab("y") + theme(text = element_text(size = 20)) +
    ggtitle("n = 50, alpha = 1") + theme(plot.title = element_text(hjust = 0.5))
```




```
## Sethuraman representation
alpha <- 0.5
K <- 1000

# baseline (G_0) is standard normal distribution
theta <- rnorm(K, 0, 1)

v <- rbeta(K, 1, alpha)
omega <- numeric(K)
omega[1] <- v[1]
cumv = cumprod(1 - v)
for (k in 2:(K - 1)) {
    omega[k] <- v[k] * cumv[k - 1]
}
omega[K] <- 1 - sum(omega[1:(K - 1)])

plot(theta, omega, type = "h", main = "", xlab = expression(theta),
    ylab = expression(omega))
```




```
theta_sort <- sort(theta)
omega_ordered <- omega[order(theta)]
fdist <- numeric(K)
for (k in 1:K) {
    fdist[k] <- cumsum(omega_ordered)[k]
}

plot(theta_sort, pnorm(theta_sort, 0, 1), type = "l", lwd = 2,
    ylab = "CDF")
lines(theta_sort, fdist, col = "red")
```



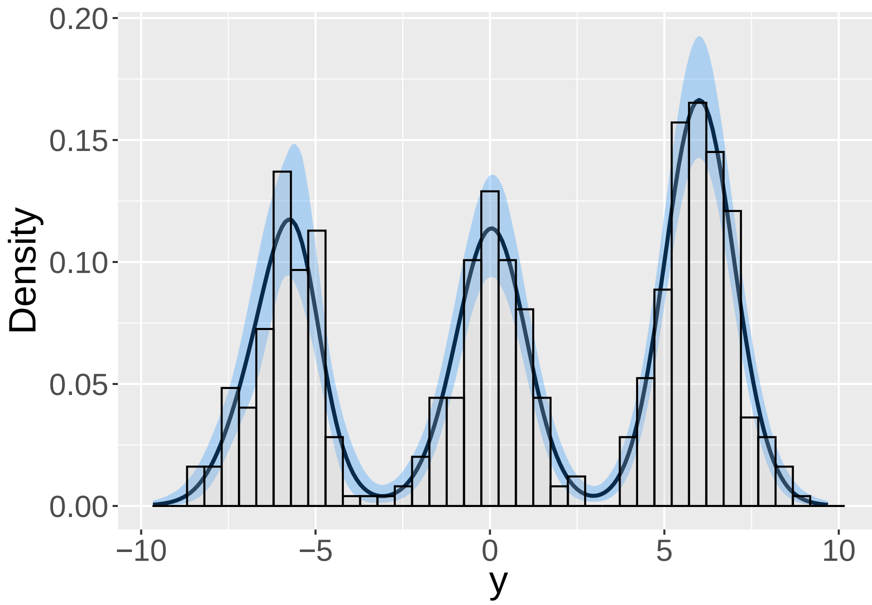
Dirichlet process mixtures

R code

```
dpm <- function(y, amu, b2mu, asigma2, bsigma2, alpha, L, nsim,
    nburn) {
```

```r
n <- length(y)
p <- ns <- rep(0, L)
v <- rep(1/L, L)
v[L] <- 1
prop <- prob <- matrix(0, nrow = n, ncol = L)

z <- matrix(0, nrow = nsim, ncol = n)
z[1, ] <- rep(1, n)

P <- Mu <- Sigma2 <- matrix(0, nrow = nsim, ncol = L)
Mu[1, ] <- rep(mean(y), L)
Sigma2[1, ] <- rep(var(y), L)

for (i in 2:nsim) {
    cumv <- cumprod(1 - v)
    p[1] <- v[1]
    for (l in 2:L) {
        p[l] <- v[l] * cumv[l - 1]
    }

    for (l in 1:L) {
        prop[, l] <- p[l] * dnorm(y, mean = Mu[i - 1, l],
            sd = sqrt(Sigma2[i - 1, l]))
    }
    prob <- prop/apply(prop, 1, sum)
    for (j in 1:n) {
        z[i, j] <- sample(1:L, size = 1, prob = prob[j, ])
    }
    P[i, ] <- p

    for (l in 1:L) {
        ns[l] <- length(which(z[i, ] == l))
    }

    for (l in 1:(L - 1)) {
        v[l] <- rbeta(1, 1 + ns[l], alpha + sum(ns[(l + 1):L]))
    }

    for (l in 1:L) {
        varmu <- 1/((1/b2mu) + (ns[l]/Sigma2[i - 1, l]))
        meanmu <- ((sum(y[z[i, ] == l])/Sigma2[i - 1, l]) +
            (amu/b2mu))/((1/b2mu) + (ns[l]/Sigma2[i - 1,
            l]))
        Mu[i, l] <- rnorm(1, mean = meanmu, sd = sqrt(varmu))

        Sigma2[i, l] <- 1/rgamma(1, asigma2 + ns[l]/2, bsigma2 +
            0.5 * sum((y[z[i, ] == l] - Mu[i, l])^2))
    }
}

res = list()
res$P = P[(nburn + 1):nsim, ]
res$Mu = Mu[(nburn + 1):nsim, ]
```

```r
    res$Sigma2 = Sigma2[(nburn + 1):nsim, ]
    res$z = z[(nburn + 1):nsim, ]
    return(res)
}

n <- 500
set.seed(123)
y <- rnormmix(n, c(0.3, 0.3, 0.4), c(-6, 0, 6), c(1, 1, 1))
grid <- seq(min(y) - 1, max(y) + 1, len = 200)
ngrid <- length(grid)
nsim <- 5000
nburn = 500

set.seed(123)
fitdpm <- dpm(y = y, amu = 0, b2mu = 100, asigma2 = 1, bsigma2 = 1,
    alpha = 1, L = 50, nsim = nsim, nburn = nburn)

L <- 50
p <- fitdpm$P
mu <- fitdpm$Mu
sigma2 <- fitdpm$Sigma2
niter <- nrow(p)

dens <- array(0, c(niter, ngrid, L))
dens1 <- matrix(0, nrow = niter, ncol = ngrid)

for (l in 1:niter) {
    for (k in 1:L) {
        dens[l, , k] <- p[l, k] * dnorm(grid, mu[l, k], sqrt(sigma2[l,
            k]))
    }
    for (j in 1:ngrid) {
        dens1[l, j] <- sum(dens[l, j, ])
    }
}

densm <- apply(dens1, 2, mean)
densl <- apply(dens1, 2, quantile, prob = 0.025)
densh <- apply(dens1, 2, quantile, prob = 0.975)

dfhist <- data.frame(y = y)
dfdens <- data.frame(dm = densm, dl = densl, dh = densh, seqgrid = grid)

ggplot(dfdens, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + xlab(expression(y)) +
    ylab("Density") + theme(text = element_text(size = 20))
```




```
ghcdpm <- good_fit_criteria(y = y, P = fitdpm$P, Mu = fitdpm$Mu,
    Sigma2 = fitdpm$Sigma2, nburn = nburn)
ghcdpm$LPML
```

```
## [1] -1260.859
```

```
ghcdpm$WAIC
```

```
## [1] 2521.71
```

JAGS code

```
model_string = "
model {
  for (i in 1:n) {
    y[i] ~ dnorm(mu[z[i]], tau[z[i]])
    z[i] ~ dcat(pi[])
  }

  for (l in 1:L) {
    mu[l] ~ dnorm(0, 0.01)
    tau[l] ~ dgamma(1, 1)
    sigma2[l] <- 1/tau[l]
  }

  # Stick breaking
  for (l in 1:(L-1)) {v[l] ~ dbeta(1,alpha)}
  v[L] <- 1
  pi[1] <- v[1]
  for (l in 2:L) {
    pi[l] <- v[l] * (1-v[l-1]) * pi[l-1] / v[l-1]
```

```
  }
alpha ~ dgamma(2,2)
}"

require(mixtools)
n <- 500
set.seed(123)
y <- rnormmix(n, c(0.3, 0.3, 0.4), c(-6, 0, 6), c(1, 1, 1))
grid <- seq(min(y) - 1, max(y) + 1, len = 200)
ngrid <- length(grid)
L <- 50

data <- list(n = length(y), y = y, L = L)
model <- jags.model(textConnection(model_string), n.chains = 1,
    data = data)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 500
##    Unobserved stochastic nodes: 650
##    Total graph size: 2354
##
## Initializing model
```

```
update(model, nburn)
res <- jags.samples(model, variable.names = c("mu", "sigma2",
    "pi", "z"), n.iter = nsim)

P <- Mu <- Sigma2 <- matrix(0, nrow = nsim, ncol = L)
for (l in 1:L) {
    Mu[, l] <- res$mu[l, , 1]
    Sigma2[, l] <- res$sigma2[l, , 1]
    P[, l] <- res$pi[l, , 1]
}

grid <- seq(min(y) - 1, max(y) + 1, len = 200)
ngrid <- length(grid)
dens <- array(0, c(nsim, ngrid, L))
dens1 <- matrix(0, nrow = nsim, ncol = ngrid)

for (i in 1:nsim) {
    for (l in 1:L) {
        dens[i, , l] <- P[i, l] * dnorm(grid, Mu[i, l], sqrt(Sigma2[i,
            l]))
    }
    for (j in 1:ngrid) {
        dens1[i, j] <- sum(dens[i, j, ])
    }
}

densm <- apply(dens1, 2, mean)
```

```r
densl <- apply(dens1, 2, quantile, prob = 0.025)
densh <- apply(dens1, 2, quantile, prob = 0.975)

dfhist <- data.frame(y = y)
dfdens <- data.frame(dm = densm, dl = densl, dh = densh, seqgrid = grid)

ggplot(dfdens, aes(x = seqgrid, y = dm)) + geom_line(size = 1) +
    geom_ribbon(data = dfdens, aes(x = seqgrid, ymin = dl, ymax = dh),
        alpha = 0.3, fill = "dodgerblue1") + geom_histogram(data = dfhist,
    aes(x = y, y = after_stat(density)), alpha = 0.2, bins = 40,
    inherit.aes = FALSE, fill = "gray", colour = "black") + xlab(expression(y)) +
    ylab("Density") + theme(text = element_text(size = 20))
```