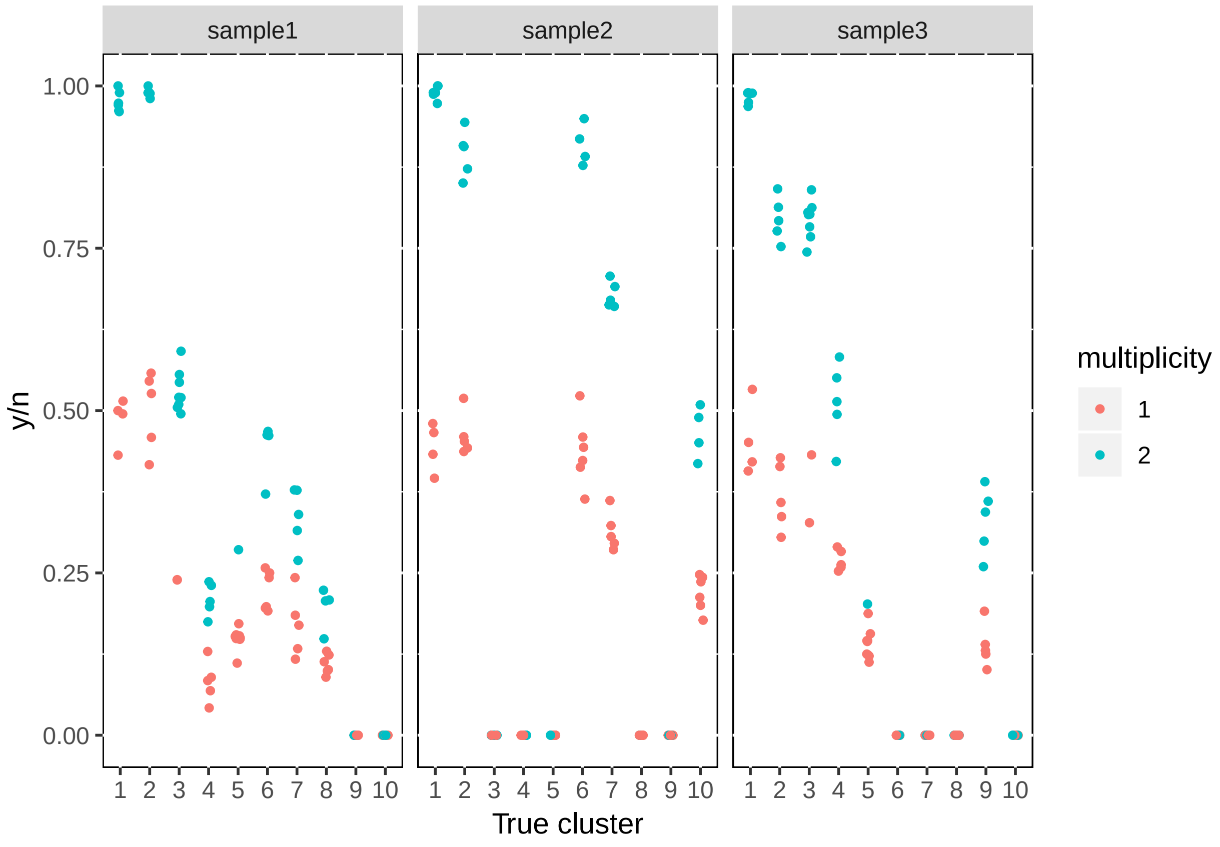
Version 6: Cluster first

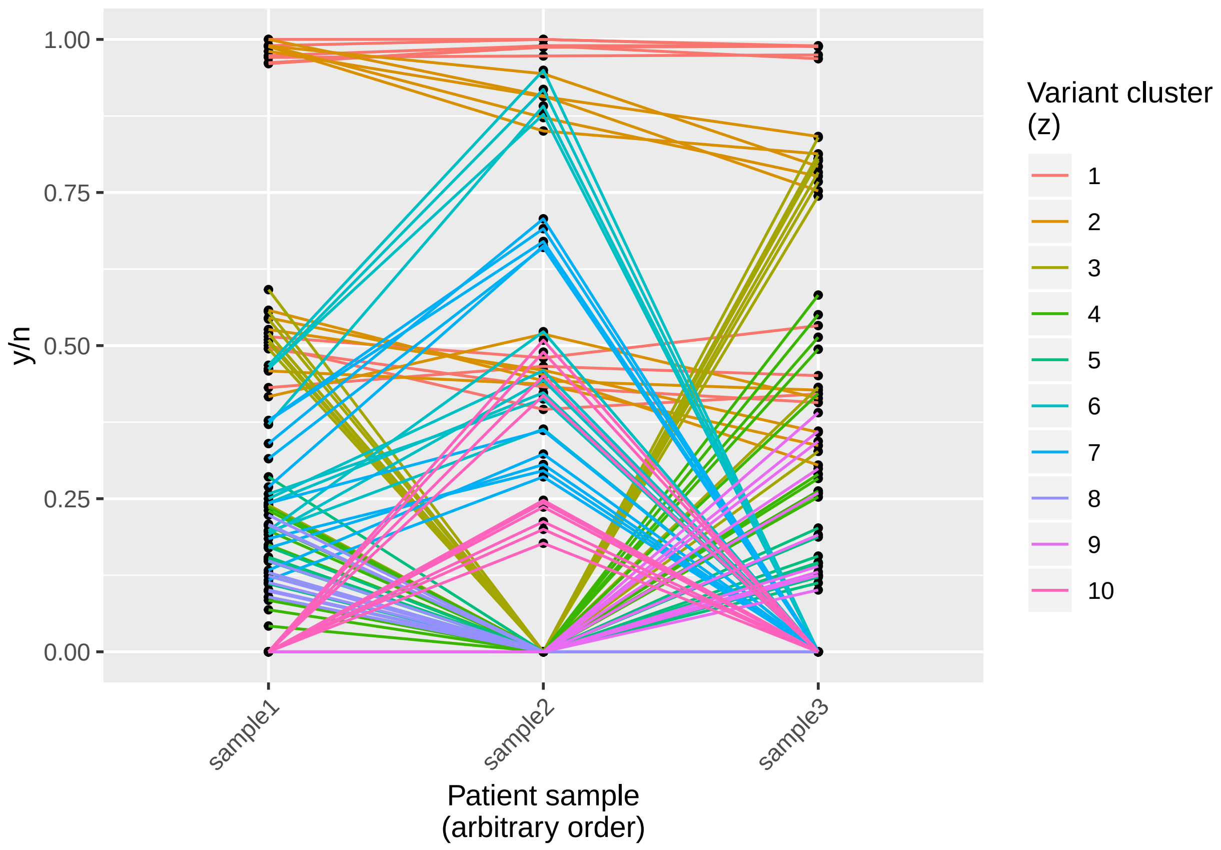
Simulate data

```r
I <- 100
K <- 10
S <- 3

set.seed(123)

pi <- rep(0.1, 10)
#z <- sample(1:K, size = I, replace = T, prob = pi)
z <- rep(1:10, each=10)
w <- matrix(c(0.98, 0.99, 0.97,
              0.98, 0.90, 0.82,
              0.55, 0.00, 0.80,
              0.20, 0.00, 0.50,
              0.30, 0.00, 0.30,
              0.43, 0.90, 0.00,
              0.30, 0.70, 0.00,
              0.20, 0.00, 0.00,
              0.00, 0.00, 0.30,
              0.00, 0.50, 0.00),
            byrow=T,
            nrow=K, ncol=S)

colnames(w) <-  paste0("sample", 1:S)

tcn <- matrix(2, nrow=I, ncol=S)
m <- matrix(rep(sample(1:2, size = I, replace = T), S),
            nrow=I, ncol=S)
W <- w[z, ]
calcTheta <- function(m, tcn, w) {
  (m * w) / (tcn * w + 2*(1-w))
}

theta <- calcTheta(m, tcn, W)

n <- replicate(S, rpois(I, 100))
y <- matrix(NA, nrow=I, ncol=S)
for (i in 1:I) {
  for (s in 1:S) {
    y[i, s] <- rbinom(1, n[i, s], theta[i,s])
  }
}

test.data <- list("I" = I, "S" = S, "K" = K,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
```

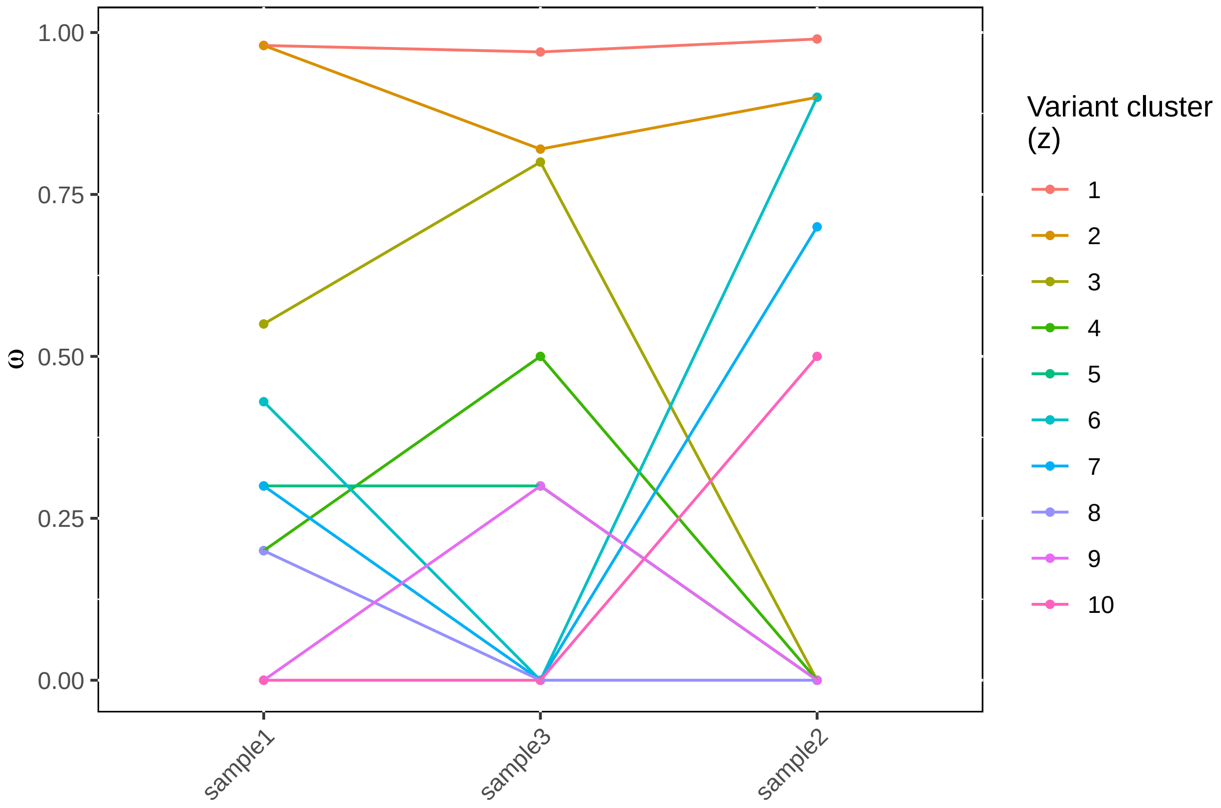
Visualize densities of simulated data

Clustering is by ω





1.00
0.75
0.50
0.25
0.00
y/n
sample1
sample2
sample3
Patient sample
(arbitrary order)
Variant cluster
(z)
1
2
3
4
5
6
7
8
9
10



1.00
0.75
0.50
0.25
0.00
ω
sample1
sample3
sample2
Variant cluster
(z)
1
2
3
4
5
6
7
8
9
10

functions

```r
runMCMC <- function(data, K, jags.file, inits, params, n.iter, thin) {
  data$K <- K
  jags.m <- jags.model(jags.file, data,
                       n.chains = 1,
                       inits = inits,
                       n.adapt = 1000)
  samps <- coda.samples(jags.m, params, n.iter=n.iter, thin=thin)
  samps
}

getParamChain <- function(samps, param) {
  chains <- do.call(rbind, samps)
  chain <- chains[, grep(param, colnames(chains))]
}

reshapeW <- function(w, S, K) {
  w.mat <- matrix(w, nrow = K)
  colnames(w.mat) <- paste0("sample", 1:S)
  w.mat
}

calcLogLik <- function(z.iter, w.iter, data) {
  W <- w.iter[z.iter, ]
  theta <- calcTheta(data$m, data$tcn, W)
  sum(dbinom(data$y, data$n, theta, log=T))
}

calcChainLogLik <- function(samps, data, K) {
  z.chain <- getParamChain(samps, "z")
  w.chain <- getParamChain(samps, "w")
  lik <- c()
  for(iter in 1:nrow(z.chain)) {
    z.iter <- z.chain[iter,]
    w.iter <- reshapeW(w.chain[iter,], data$S, K)
    lik <- c(lik, calcLogLik(z.iter, w.iter, data))
  }
  mean(lik)
}

calcBIC <- function(n, k, ll) log(n)*k - 2*ll
```

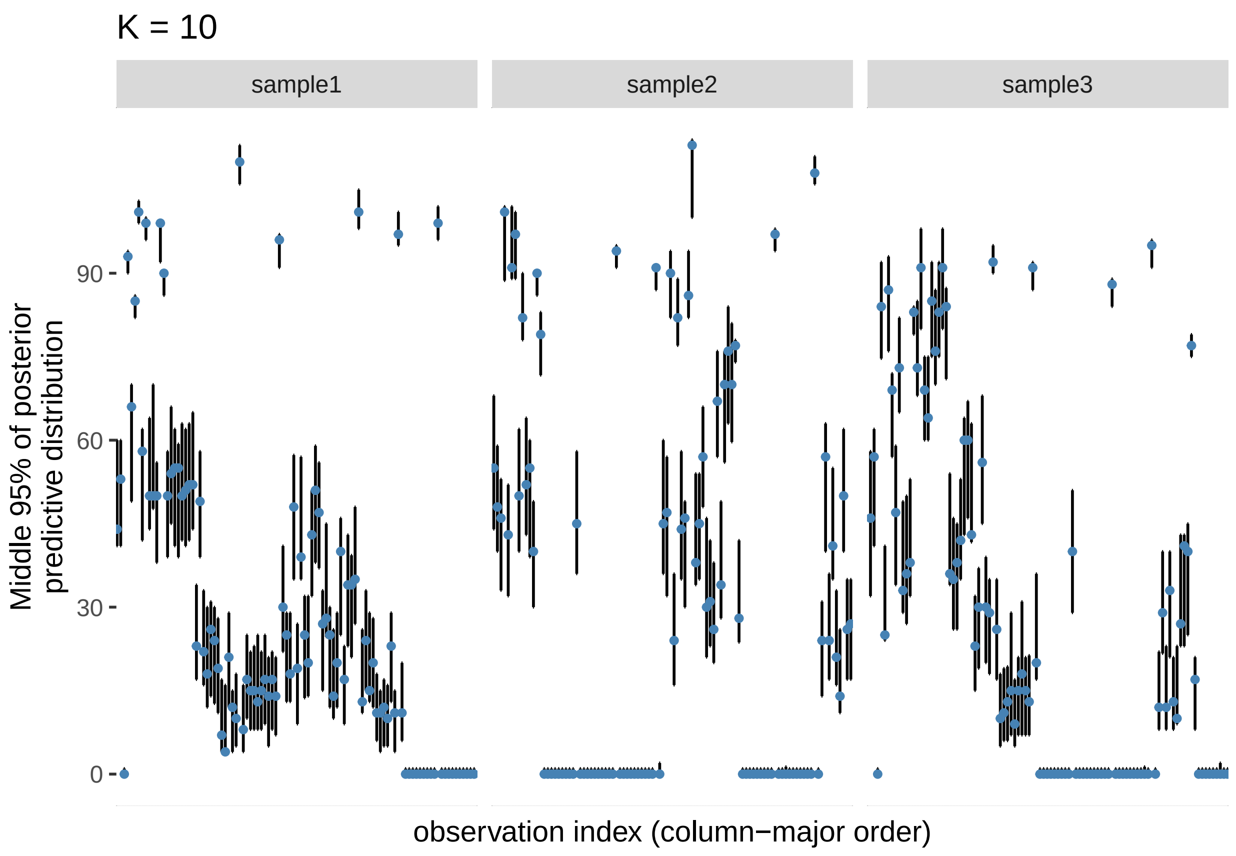
Cluster – JAGS

```r
jags.file <- file.path(models.dir, "w.jags")
inits <- list(".RNG.name" = "base::Wichmann-Hill",
              ".RNG.seed" = 123)
test.data <- list("I" = I, "S" = S,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
params <- c("z", "w", "ystar")
```

```
n.iter = 10000
thin = 7
K <- 10

samps <- runMCMC(test.data, K, jags.file, inits, params, n.iter, thin)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 300
##    Unobserved stochastic nodes: 431
##    Total graph size: 4797
##
## Initializing model
```

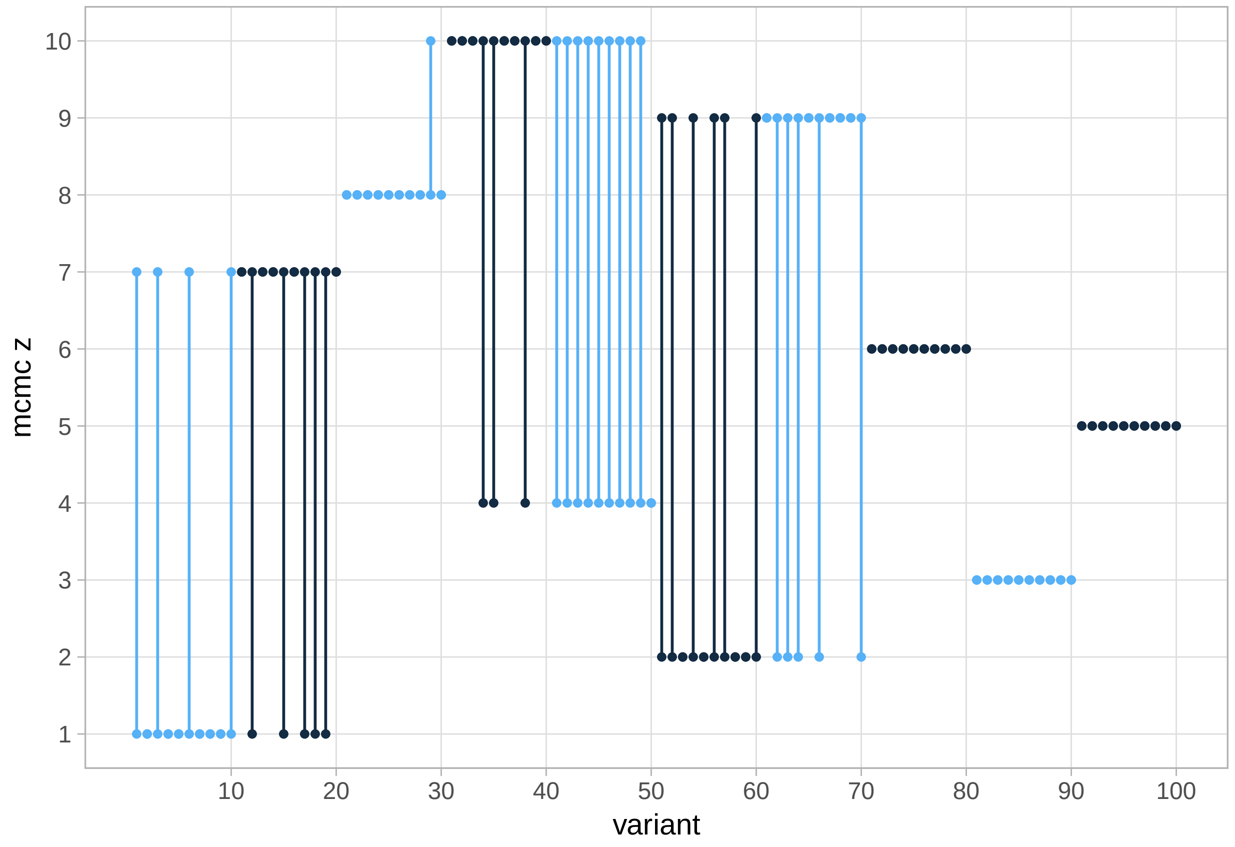
```
z.chain <- getParamChain(samps, "z")
w.chain <- getParamChain(samps, "w")
```



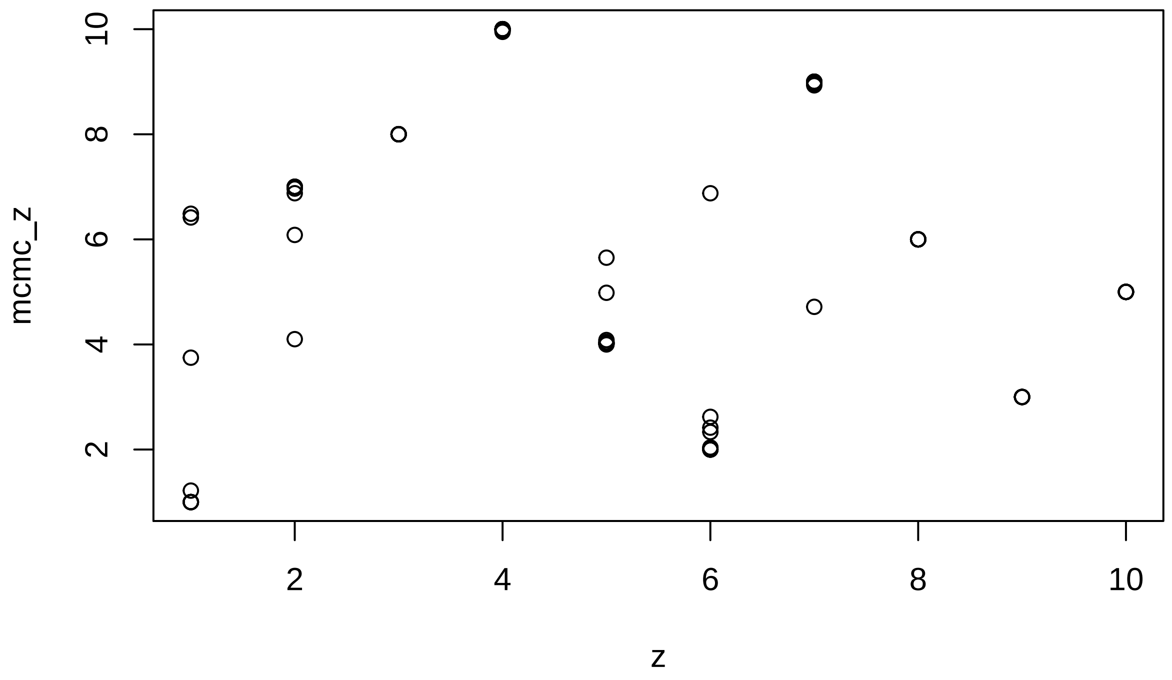
```
plot.z <- function(samps, z) {
  mcmc_vals <- summary(samps)$statistics
  mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
  plot(z, mcmc_z, type = "p")
  z_comp <- data.frame(z, mcmc_z)
  #plot.new()
  #grid.table(z_comp, rows=NULL)
  #z_mapping <- distinct(round(z_comp, 0))
  #z_mapping <- z_mapping[order(z_mapping$mcmc_z), ]
  #plot.new()
  #grid.table(z_mapping, rows=NULL)
}

z.chain.to.tb <- function(z.chain) {
  z.chain.tb <- z.chain %>%
    as_tibble() %>%
    mutate(iter=1:nrow(z.chain)) %>%
    gather(variant, mcmc_z, -c(iter))
  z.chain.tb %>%
    mutate(variant = as.integer(gsub(".*\\[(.*)\\].*", "\\1", z.chain.tb$variant)))
}

z.chain.tb <- z.chain.to.tb(z.chain)
z.chain.tb <- z.chain.tb %>%
  mutate(true_z = rep(1:10, each=nrow(z.chain)*10))
z.chain.tb_simp <- distinct(select(z.chain.tb, -c(iter)))

z.chain.plot <- ggplot(z.chain.tb, aes(variant, mcmc_z)) +
  geom_point(size=1, aes(color=true_z)) +
  ylab("mcmc z") +
  xlab("variant") +
  theme_light() +
  scale_y_continuous(breaks = 1:K)

z.simp.tb <- tibble(variant = integer(),
                    mcmc_z_1 = integer(),
                    mcmc_z_2 = integer(),
                    true_z = integer())
for (i in 1:ncol(z.chain)) {
  z.vals <- as.integer(names(table(z.chain[,i])))
  if (length(z.vals) > 1) {
    z.simp.tb[i, ] <- c(i, z.vals[1], z.vals[2], z[i])
  } else {
    z.simp.tb[i, ] <- c(i, z.vals, z.vals, z[i])
  }
}
z.simp.tb
```

```
## # A tibble: 100 x 4
##    variant mcmc_z_1 mcmc_z_2 true_z
##  *   <int>    <int>    <int>  <int>
##  1       1        1        7      1
##  2       2        1        1      1
##  3       3        1        7      1
##  4       4        1        1      1
##  5       5        1        1      1
##  6       6        1        7      1
##  7       7        1        1      1
##  8       8        1        1      1
##  9       9        1        1      1
## 10      10        1        7      1
## # ... with 90 more rows
```

```
z.simp.plot <- ggplot(z.simp.tb, aes(variant, mcmc_z_1)) +
  ylab("mcmc z") +
  xlab("variant") +
  theme_light() +
  scale_y_continuous(breaks = 1:K, minor_breaks=NULL) +
  scale_x_continuous(breaks = seq(10,100,10), minor_breaks=NULL) +
  geom_segment(aes(x=variant, xend=variant,
                   y=mcmc_z_1, yend=mcmc_z_2,
                   color=true_z%%2)) +
  geom_point(size=1, aes(color=true_z%%2)) +
  geom_point(aes(y=mcmc_z_2, color=true_z%%2), size=1) +
  theme(legend.position = "none")

z.simp.plot
```



```
mcmc_vals <- summary(samps)$statistics
mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
plot.z(samps, z)
```


mcmc_z
z
2
4
6
8
10
2
4
6
8
10