Version 2: 2 clusters; Fix cluster label switching a,b; ensure w in [0,1]; 5 chains

Simulate data

```r
I <- 50
K <- 2
S <- 10

# choose diffuse priors for gamma
a_gamma <- 2
b_gamma <- 10

avrg <- a_gamma * b_gamma
std.dv <- sqrt(a_gamma*b_gamma^2)
g_range = seq(0, avrg + 5*std.dv, 0.01)
g_y = dgamma(g_range, a_gamma, rate = 1/b_gamma)
#plot(g_range, g_y, type = "l", ylim=c(0, max(g_y) + 0.01))

set.seed(123)

a <- matrix(NA, nrow=K, ncol=S)
b <- matrix(NA, nrow=K, ncol=S)
for (s in 1:S) {
  a[, s] <- rgamma(K, a_gamma, rate = 1/b_gamma)
  b[, s] <- rgamma(K, a_gamma, rate = 1/b_gamma)
}

pi <- as.vector(rdirichlet(1, rep(1, K)))
z <- sample(1:K, size = I, replace = T, prob = pi)

w <- matrix(NA, nrow=I, ncol=S)
for (s in 1:S) {
  w[, s] <- rbeta(I, a[,s][z], b[,s][z])
}

tcn <- matrix(2, nrow=I, ncol=S)
m <- matrix(rep(sample(1:2, size = I, replace = T), S), nrow=I, ncol=S)

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])
  }
}
```

JAGS

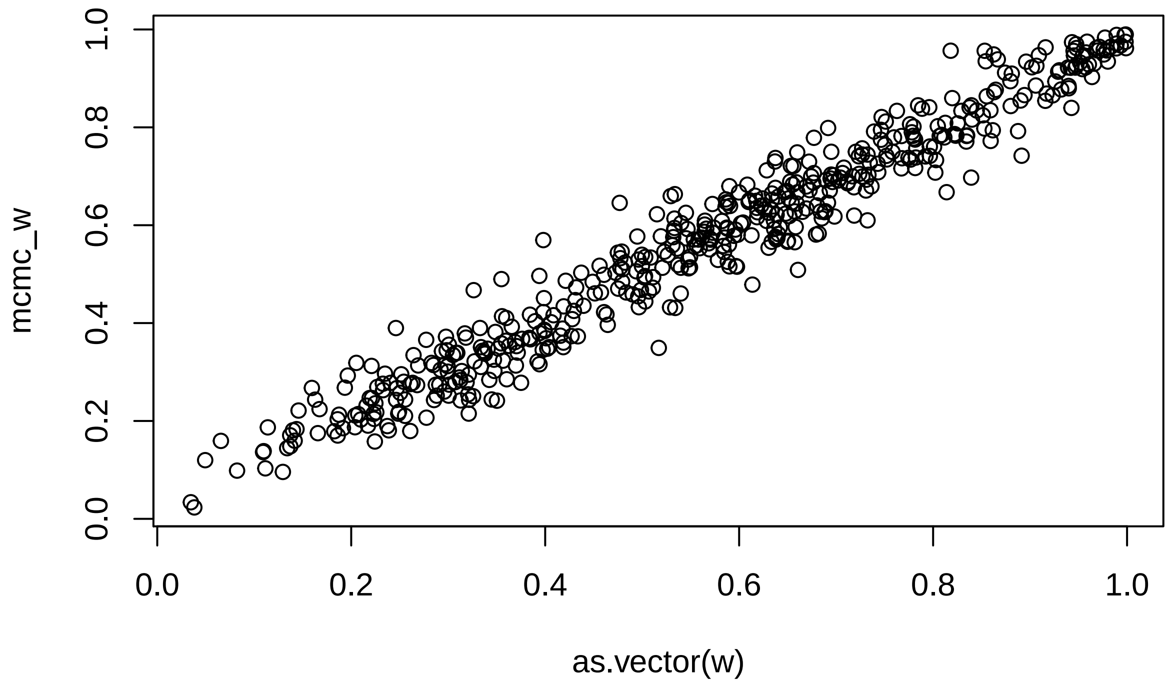
```
jags.file <- file.path(models.dir, "v2_ab.jags")

test.data <- list("I" = I, "S" = S, "K" = K,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
jags.m <- jags.model(jags.file, test.data,
                       n.chains = numChains)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 500
##    Unobserved stochastic nodes: 591
##    Total graph size: 8391
##
## Initializing model
```

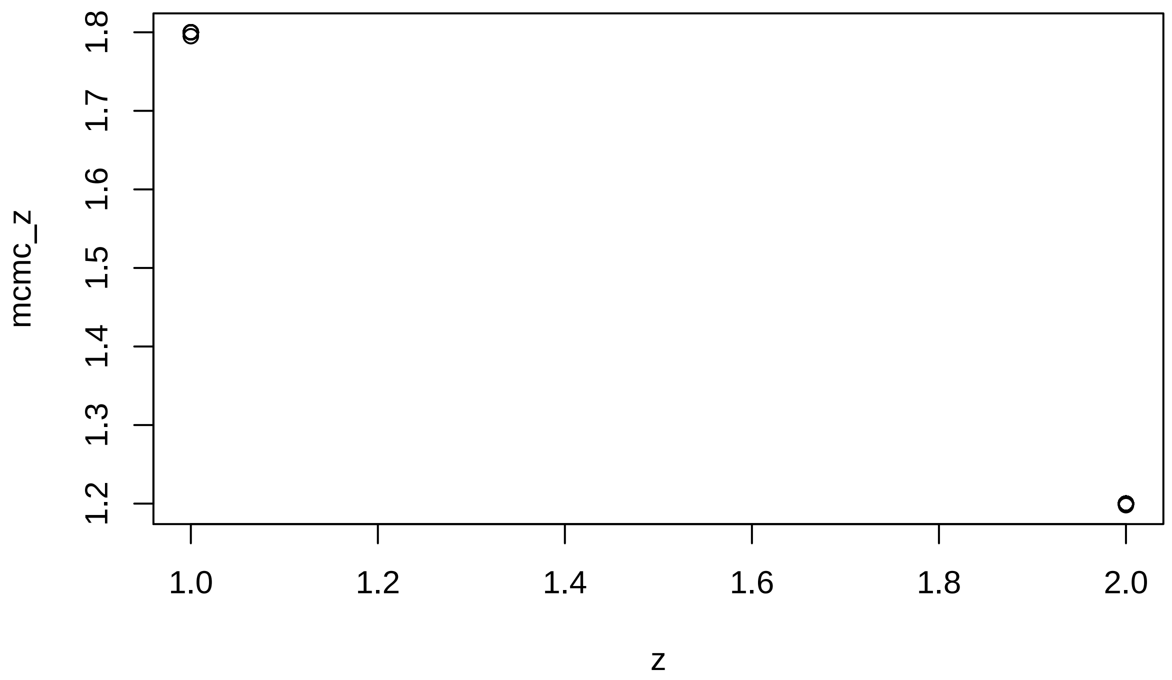
```
params <- c("z", "w", "a", "b")
update(jags.m, n.iter=1000)
samps <- coda.samples(jags.m, params, n.iter=1000)
s <- summary(samps)
dir.create(file.path(working.dir, "trace"), showWarnings = F)
pdf(file.path(working.dir, "trace", paste0(runName, "_trace.pdf")))
plot(samps)
dev.off()
```

```
## pdf
##   2
```

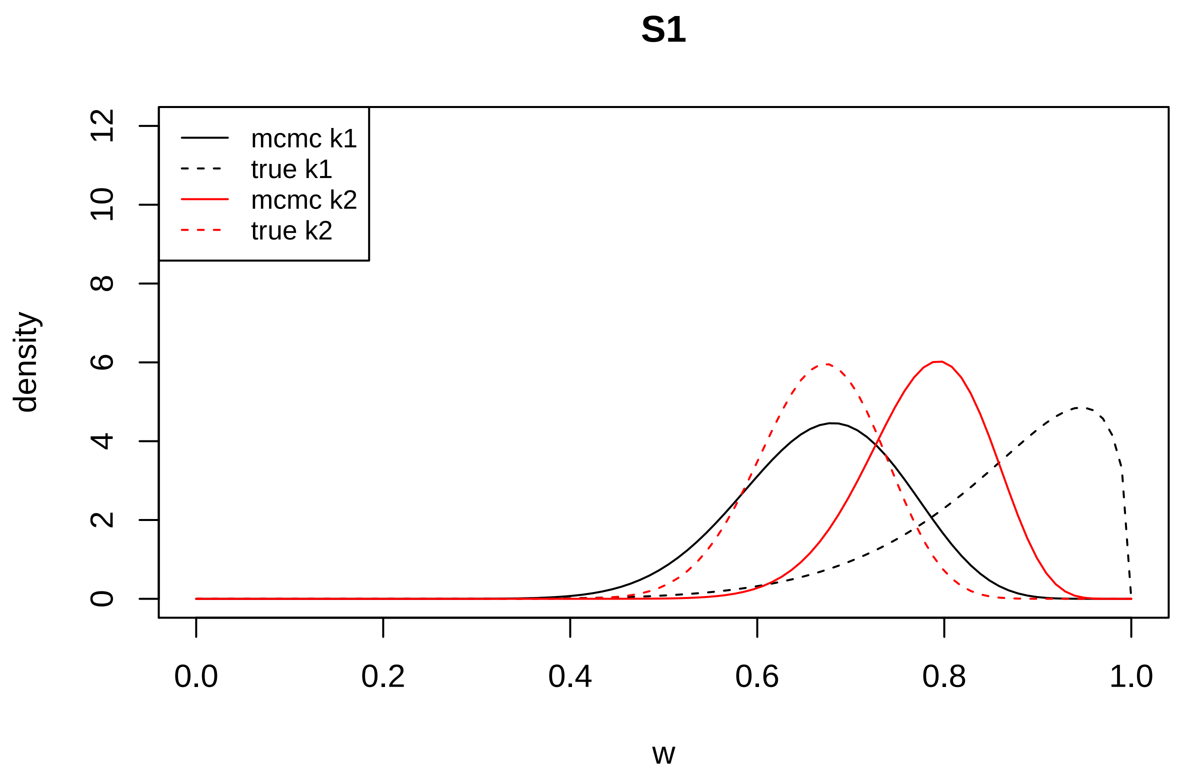
```
mcmc_vals <- s$statistics
mcmc_w <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "w", "Mean"]
plot(as.vector(w), mcmc_w, type = "p")
```

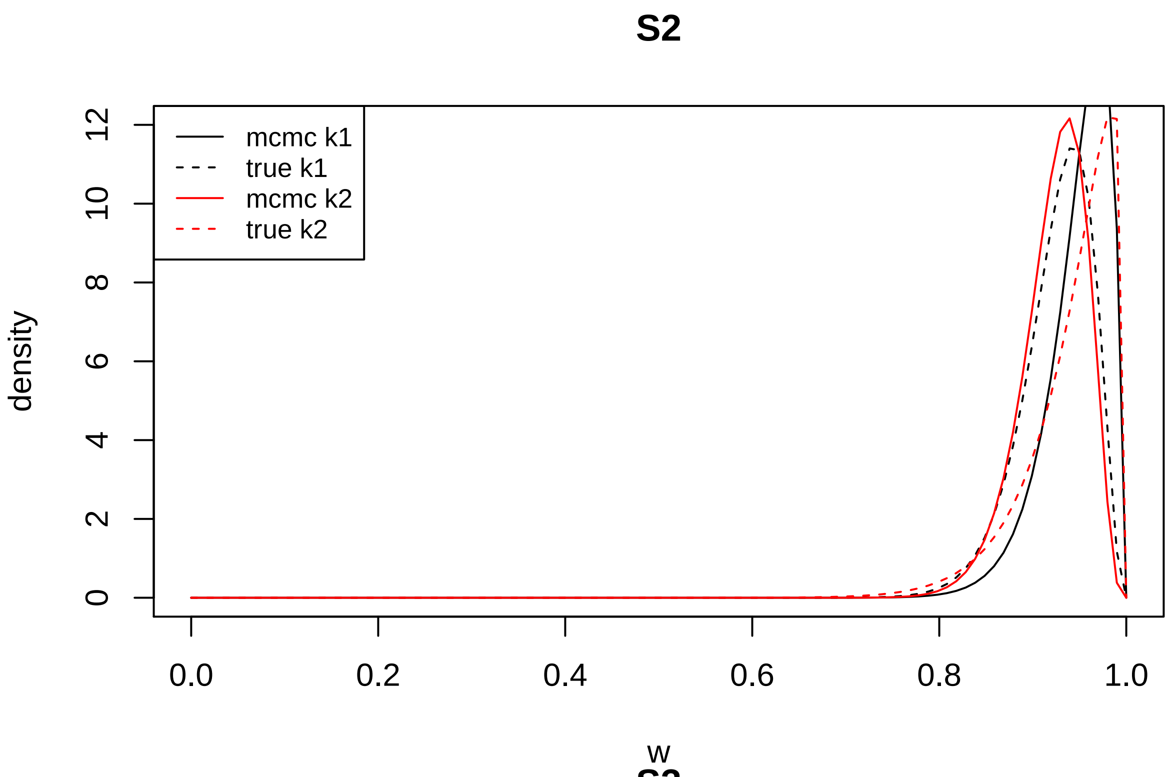


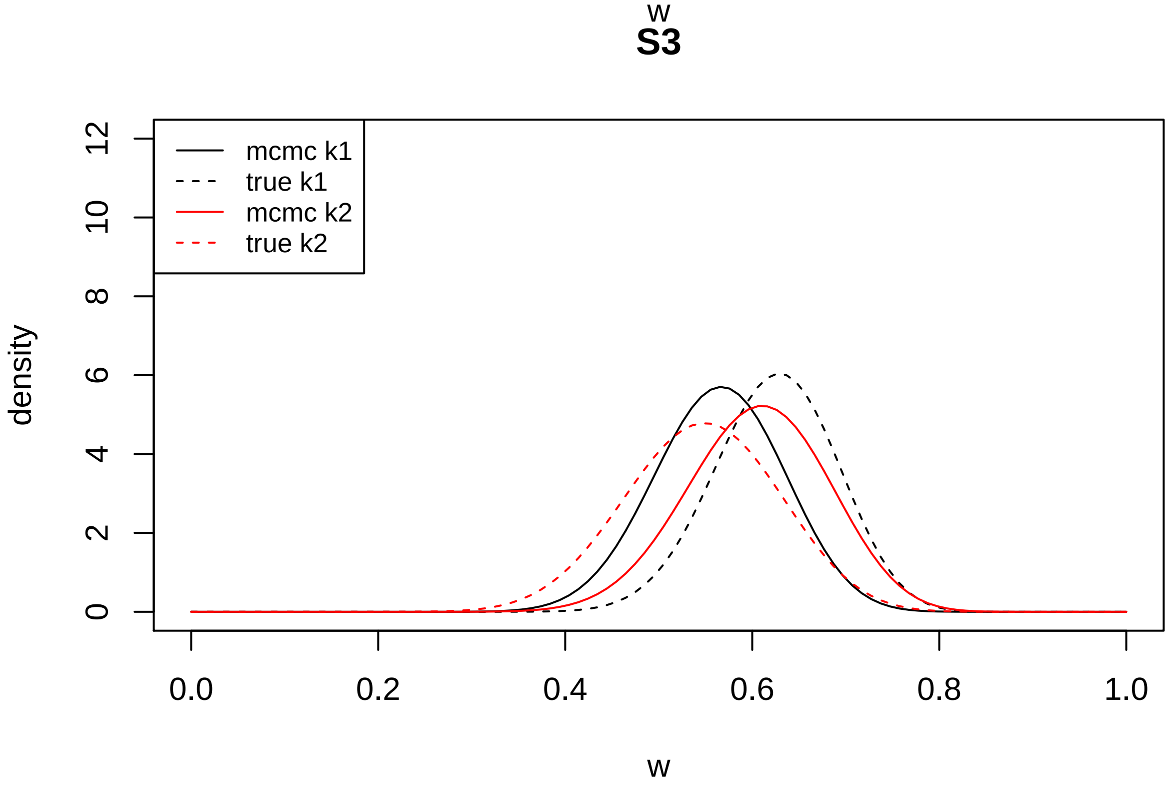
```r
mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
#mcmc_z <- round(mcmc_z, 0)
plot(z, mcmc_z, type = "p")
```

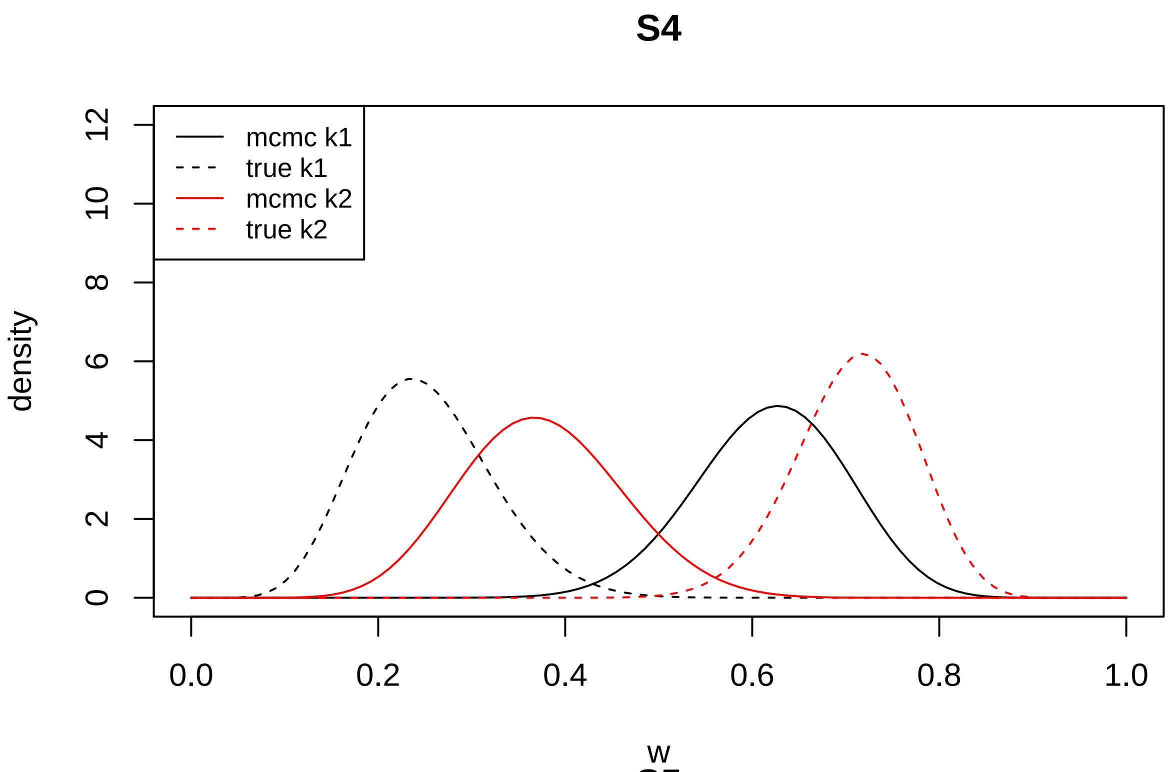


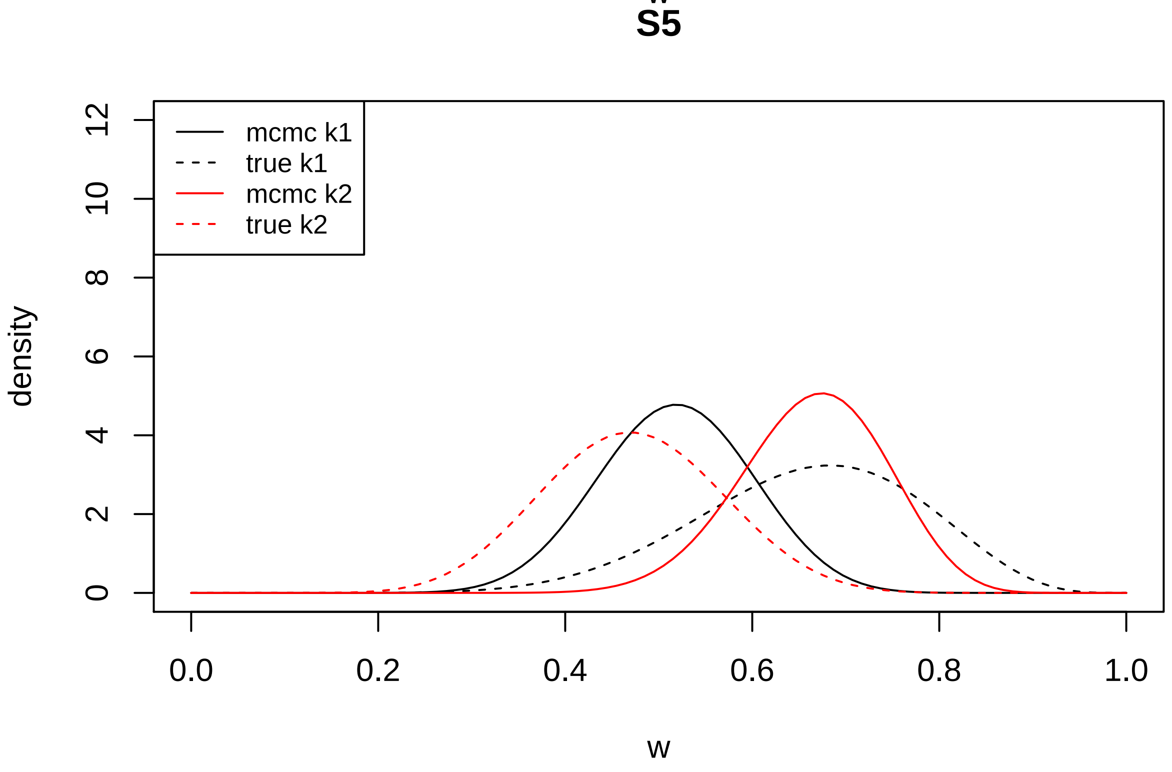
```r
mcmc_a <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "a", "Mean"]
mcmc_a <- matrix(mcmc_a, nrow=K)
mcmc_b <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "b", "Mean"]
mcmc_b <- matrix(mcmc_b, nrow=K)

p <- seq(0, 1, length = 100)
for (s in 1:S) {
  for (k in 1:K) {
    if (k == 1) {
      # plot mcmc mean U,V
      plot(p, dbeta(p, mcmc_a[k,s], mcmc_b[k,s]),
           main = paste0("S", s),
           ylab = "density", xlab = "w", type = "l", col = k,
           ylim = c(0, 12))
      # plot truth
      lines(p, dbeta(p, a[k,s], b[k,s]), type = "l", col = k, lty=2)
      # add legend
      legend(x = "topleft",
             legend = paste0(c("mcmc k", "true k"), rep(1:K, each=2)),
             col = rep(1:K, each=2),
             lty = rep(1:2, K),
             cex=0.8)
    } else {
      # plot mcmc mean U,V
      lines(p, dbeta(p, mcmc_a[k,s], mcmc_b[k,s]),
            type = "l", col = k)
      # plot truth
      lines(p, dbeta(p, a[k,s], b[k,s]), type = "l", col = k, lty=2)
    }
  }
}
```

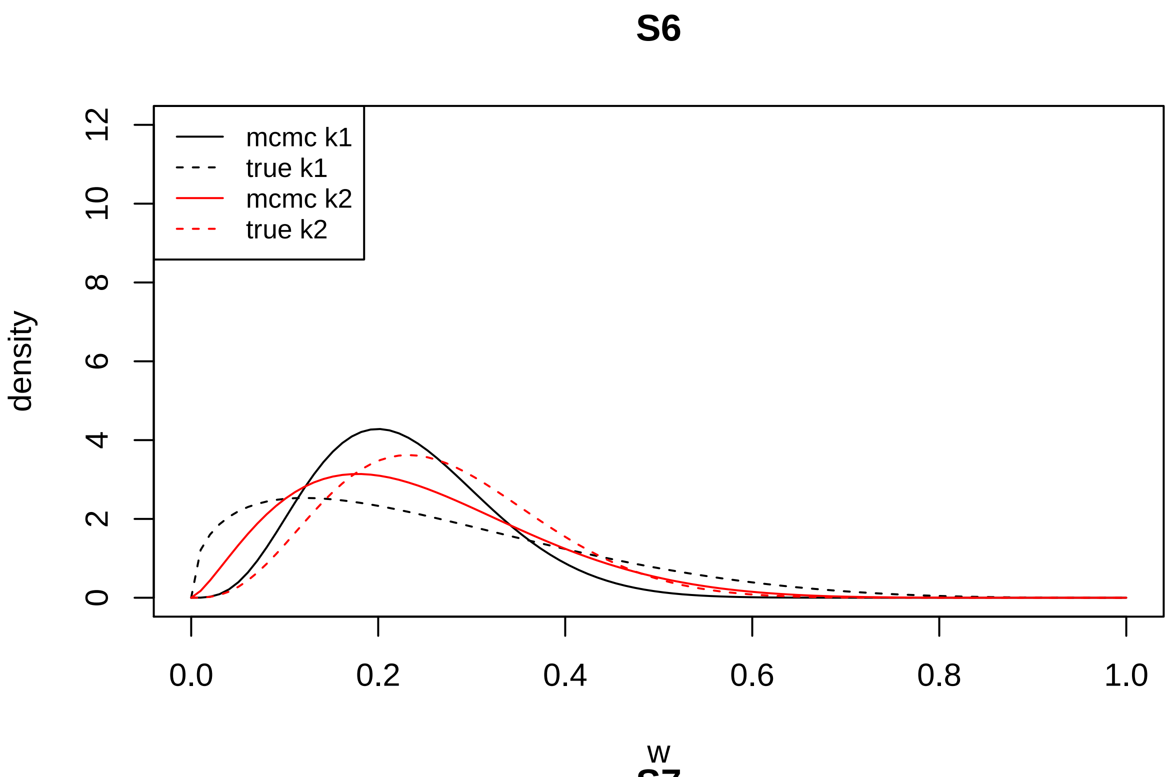


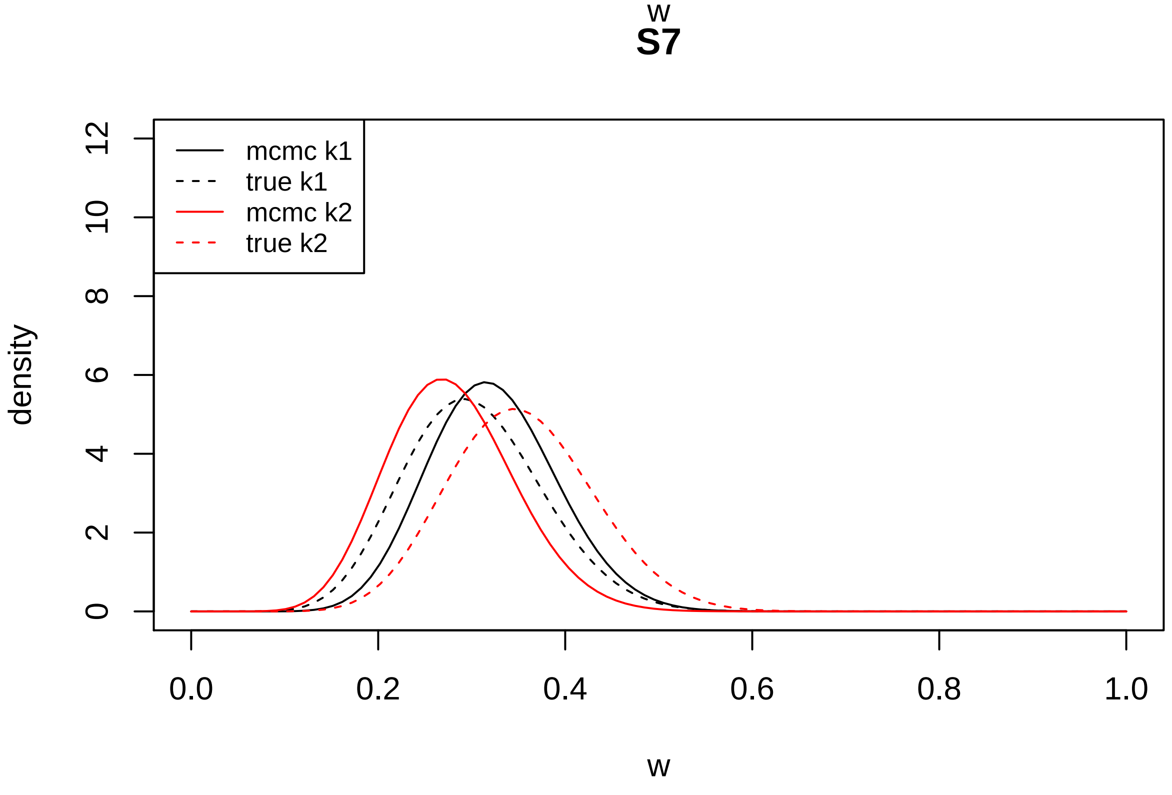

S2
density
w
mcmc k1
true k1
mcmc k2
true k2
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0

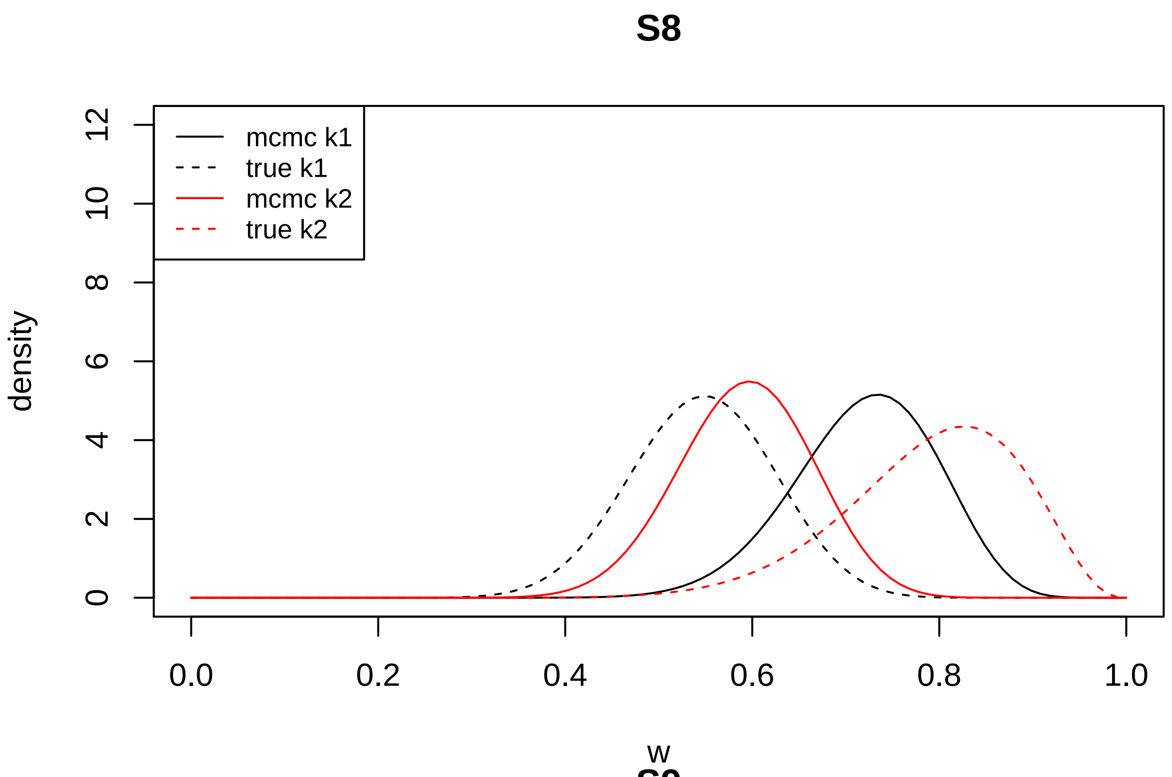

S3
density
w
mcmc k1
true k1
mcmc k2
true k2
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0

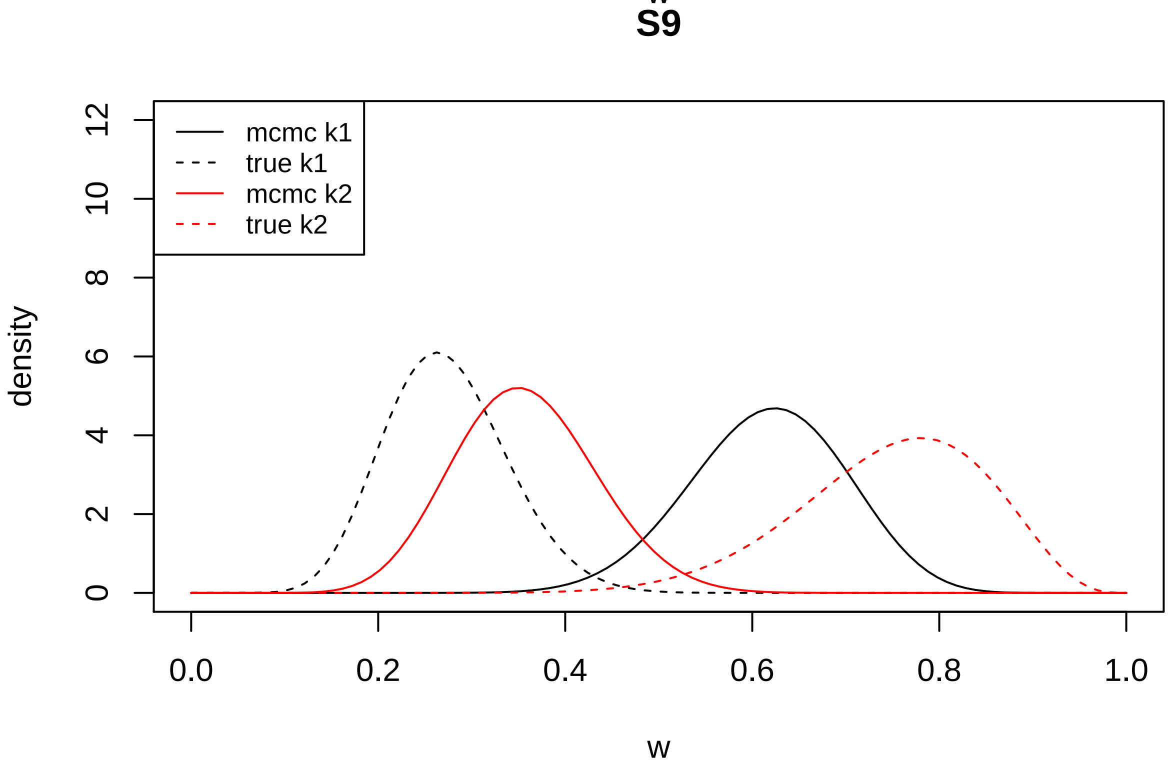

S4
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12

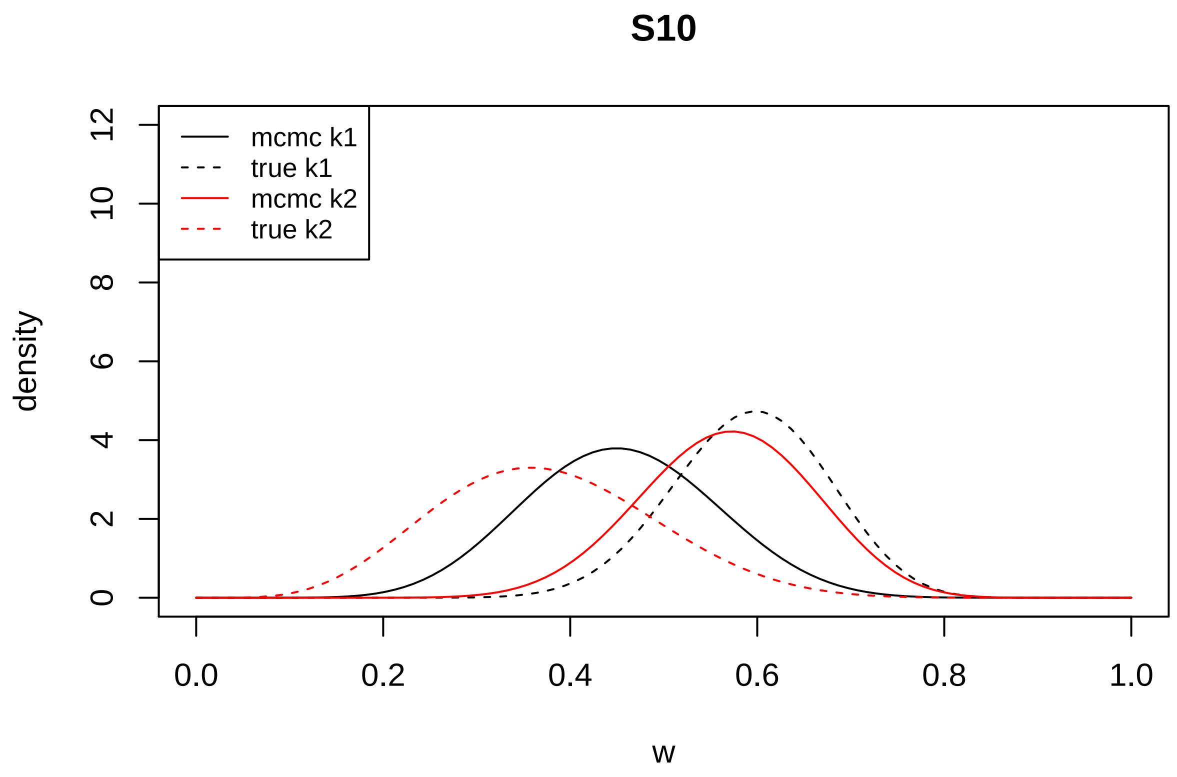

S5
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


S6
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


S7
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


S8
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


S9
mcmc k1
true k1
mcmc k2
true k2
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10
12


S10
mcmc k1
true k1
mcmc k2
true k2
density
w
0
2
4
6
8
10
12
0.0
0.2
0.4
0.6
0.8
1.0