Version 3: 3 clusters

Simulate data

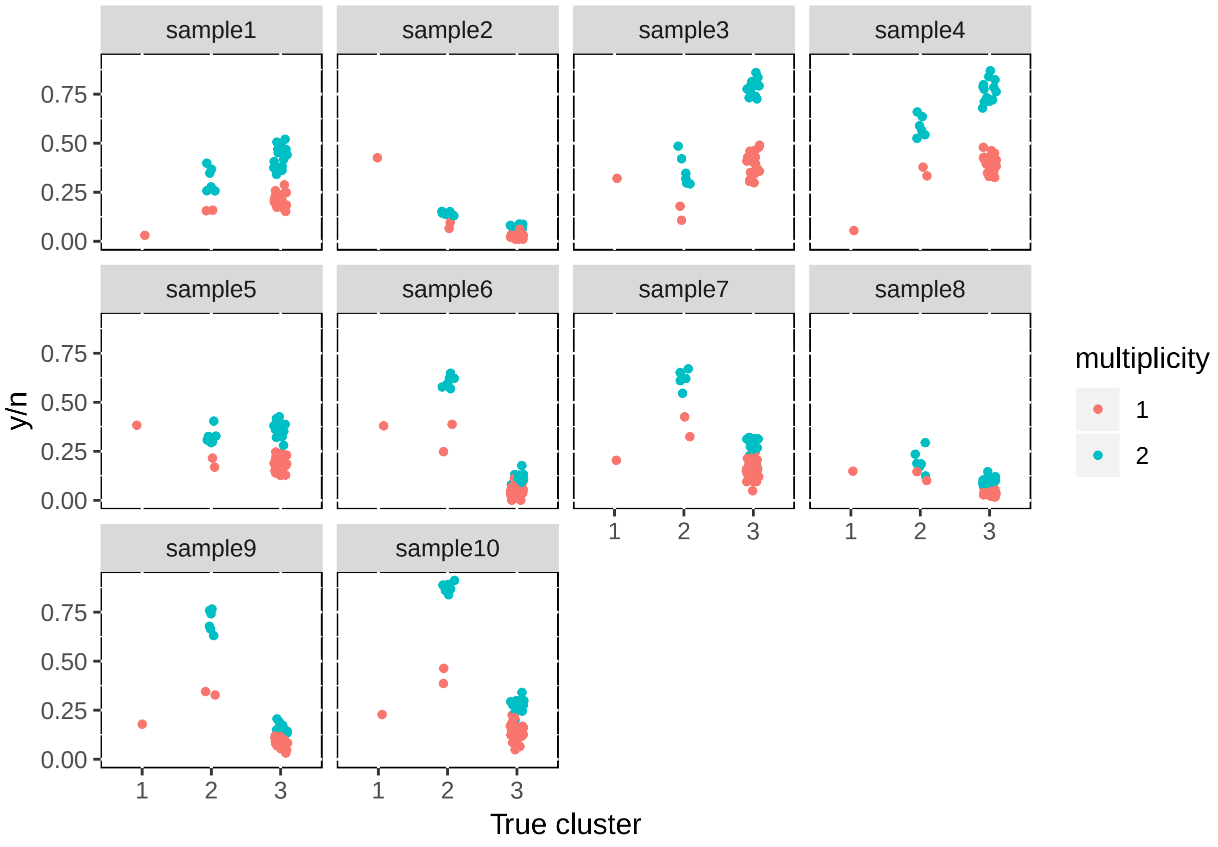
```
I <- 50
K <- 3
S <- 10

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

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)
}

# reorder a,b matrices to match ordering of means (U) in S1
U <- a/(a+b)
V <- a+b
ix <- order(U[, 1])
U.ordered <- U[ix, ]
a.ordered <- a[ix, ]
b.ordered <- b[ix, ]
V.ordered <- V[ix, ]
##
## very small values of pi will make this a difficult problem
## - might be better to specify pi manually
##
pi <- as.vector(rdirichlet(1, rep(1, K)))
z <- sample(1:K, size = I, replace = T, prob = pi)
##
## omega (w) is the cancer cell fraction
## - shouldn't omega have dimension K x S
## - I'd like to be able to recover the true values of a and b
## ->  a single random ordinate drawn from this distribution
##     might not be representative
##     a an average over 1000 random ordinates guarantees that
##     we have a representative ordinate of beta(a, b)
w <- matrix(NA, nrow=K, ncol=S)
for(k in seq_len(K)){
    tmp <- rowMeans(replicate(1000,
                              rbeta(S, a.ordered[k, ],
                                    b.ordered[k, ])))
    w[k, ] <- tmp
}
##for (s in 1:S) {
```

```r
##  w[, s] <- rbeta(I, a.ordered[,s][z], b.ordered[,s][z])
##}
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])
  }
}
```

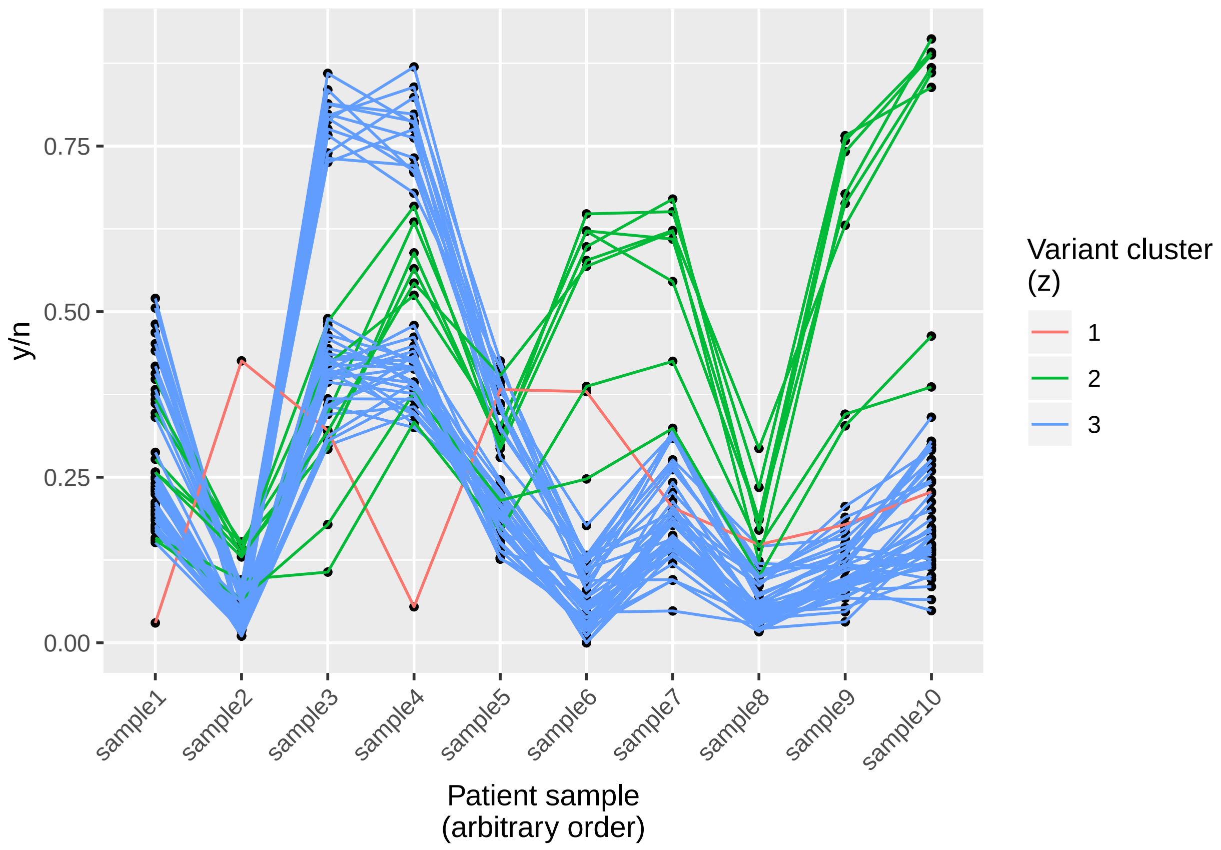
Visualize densities of simulated data

Clustering is by ω

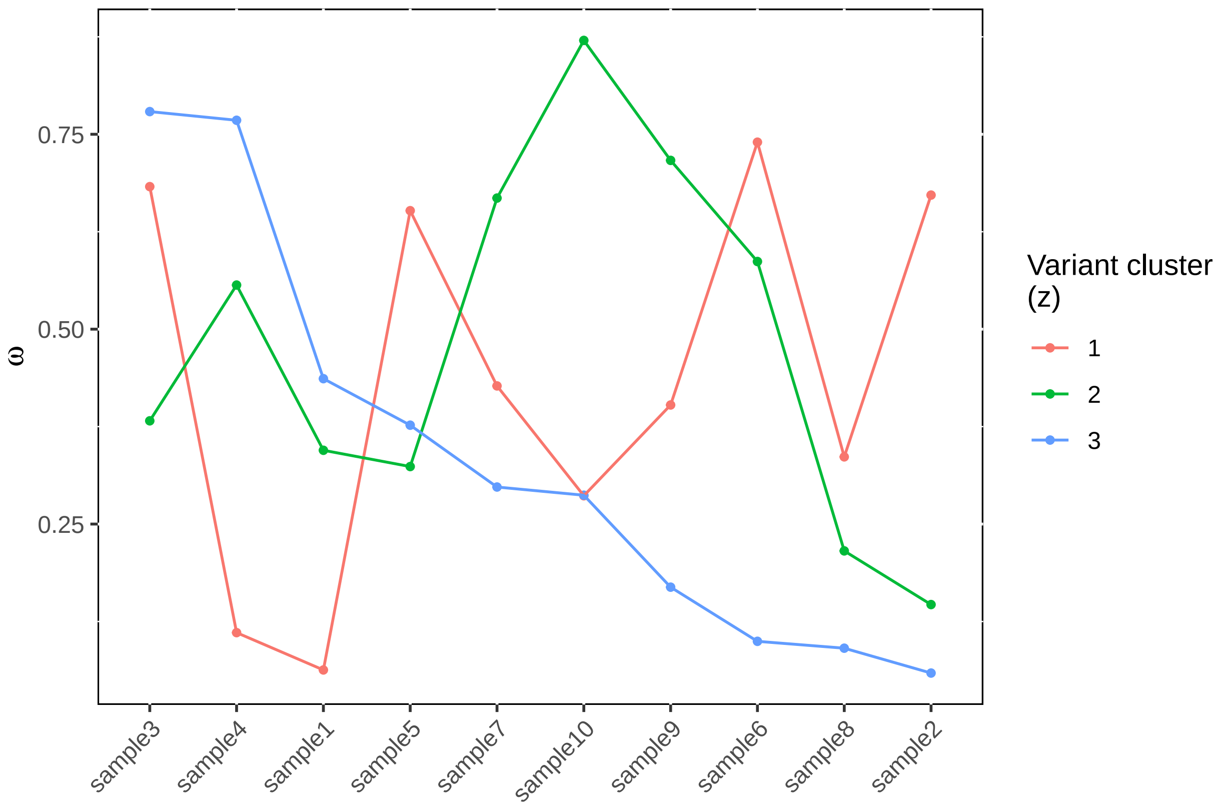
```r
test.data <- list("I" = I, "S" = S, "K" = K,
                  "y" = y, "n" = n,
                  "m" = m, "tcn" = tcn)
p <- test.data$y/test.data$n
colnames(w) <- colnames(p) <- paste0("sample", seq_len(ncol(p)))
colnames(m) <- colnames(p)
m2 <- m %>%
    as_tibble() %>%
    mutate(z=factor(z),
           variant_index=seq_len(nrow(.))) %>%
    gather("sample", "multiplicity", -c(z, variant_index)) %>%
    mutate(sample=factor(sample, levels=colnames(p)))
p2 <- p %>%
    as_tibble() %>%
    mutate(z=factor(z),
           variant_index=seq_len(nrow(.))) %>%
    gather("sample", "fraction", -c(z, variant_index)) %>%
    mutate(sample=factor(sample, levels=colnames(p))) %>%
    left_join(m2, by=c("z", "variant_index", "sample")) %>%
    mutate(multiplicity=factor(multiplicity))
## what the data looks like by sample
ggplot(p2, aes(z, fraction)) +
    geom_jitter(width=0.1, size=1, aes(color=multiplicity)) +
    facet_wrap(~sample) +
    ylab("y/n") +
    xlab("True cluster") +
    theme(panel.background=element_rect(fill="white", color="black"))
```



```
## by variant
ggplot(p2, aes(sample, fraction, group=variant_index)) +
    geom_point(size=1) +
    geom_line(aes(color=z)) +
    ylab("y/n") +
    xlab("Patient sample\n(arbitrary order)") +
    theme(axis.text.x=element_text(angle=45, hjust=1)) +
    guides(color=guide_legend(title="Variant cluster\n(z)"))
```



```
##
## Cluster means and variances
##
meds   <- apply(W, 2, median)
slevels <- colnames(w)[order(meds, decreasing=TRUE)]
w2 <- w %>%
    as_tibble() %>%
    mutate(z=factor(1:3)) %>%
    ##       variant_index=seq_len(nrow(.))) %>%
    gather("sample", "omega", -z) %>%
    mutate(sample=factor(sample, levels=slevels))
##fig1 <- ggplot(w2, aes(z, omega)) +
##    geom_jitter(width=0.1, size=1) +
##    facet_wrap(~sample) +
##    ylab(expression(omega)) +
##    xlab("True cluster") +
##    theme(panel.background=element_rect(fill="white", color="black"))
fig2 <- ggplot(w2, aes(sample, omega, group=z)) +
    geom_point(size=1,  aes(color=z)) +
    geom_line(aes(color=z)) +
    ylab(expression(omega)) +
    xlab("") +
    theme(axis.text.x=element_text(angle=45, hjust=1),
          panel.background=element_rect(fill="white", color="black"),
          legend.key=element_rect(fill="white", color="white")) +
    guides(color=guide_legend(title="Variant cluster\n(z)"))
fig2
```



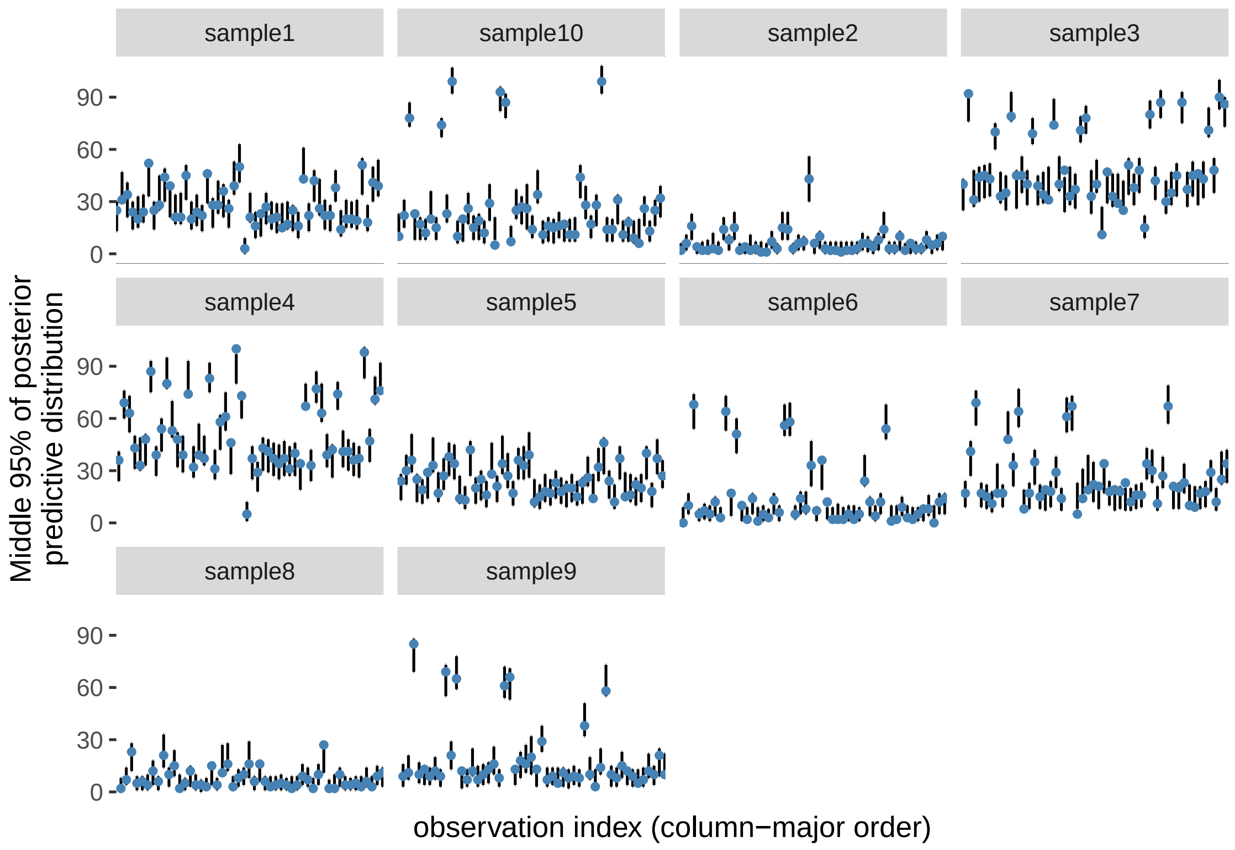
JAGS

```
##jags.file <- file.path(models.dir, "v3_no_constraints.jags")
jags.file <- file.path(models.dir, "rs_no_constraints.jags")
inits <- list(".RNG.name" = "base::Wichmann-Hill",
              ".RNG.seed" = 123)
jags.m <- jags.model(jags.file, test.data,
                     n.chains = 3,
                     inits = inits,
                     n.adapt=1000)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 500
##    Unobserved stochastic nodes: 641
##    Total graph size: 8050
##
## Initializing model
```

```
params <- c("z", "w", "U", "V", "ystar")
## z and ystar are big
samps <- coda.samples(jags.m, params,
                      n.iter=1000,
                      thin=2)
jags_df <- ggs(samps)
```

```r
##ggs_traceplot(jags_df, family="U")
s <- summary(samps)
##effectiveSize(samps)
##pdf(file.path(trace.dir, paste0(runName, "_trace.pdf")))
##plot(samps)
##dev.off()

##
## 50 mutations x 10 samples
chains <- do.call(rbind, samps)
ystar <- chains[, grep("ystar", colnames(chains))]
## each row of MCMC is in column-major order
orig.order <- tibble(statistic=colnames(ystar))
ppd.summaries <- ystar %>%
    as_tibble() %>%
    gather("statistic", "value") %>%
    group_by(statistic) %>%
    summarize(mean=mean(value),
              q1=quantile(value, 0.025),
              q3=quantile(value, 0.975))
ppd.summaries2 <- left_join(orig.order,
                            ppd.summaries, by="statistic") %>%
    mutate(observed=as.numeric(test.data$y)) %>%
    mutate(sample=paste0("sample", rep(1:10, each=50)),
           variant=rep(1:50, 10))
ggplot(ppd.summaries2, aes(x=statistic, y=mean,
                           ymin=q1,
                           ymax=q3)) +
    geom_errorbar() +
    geom_point(aes(x=statistic, y=observed),
               size=1, color="steelblue") +
    theme(axis.text.x=element_blank(),
          axis.ticks.x=element_blank(),
          panel.background=element_rect(fill="white",
                                        color="black")) +
    ylab("Middle 95% of posterior\npredictive distribution") +
    xlab("observation index (column-major order)") +
    facet_wrap(~sample)
```

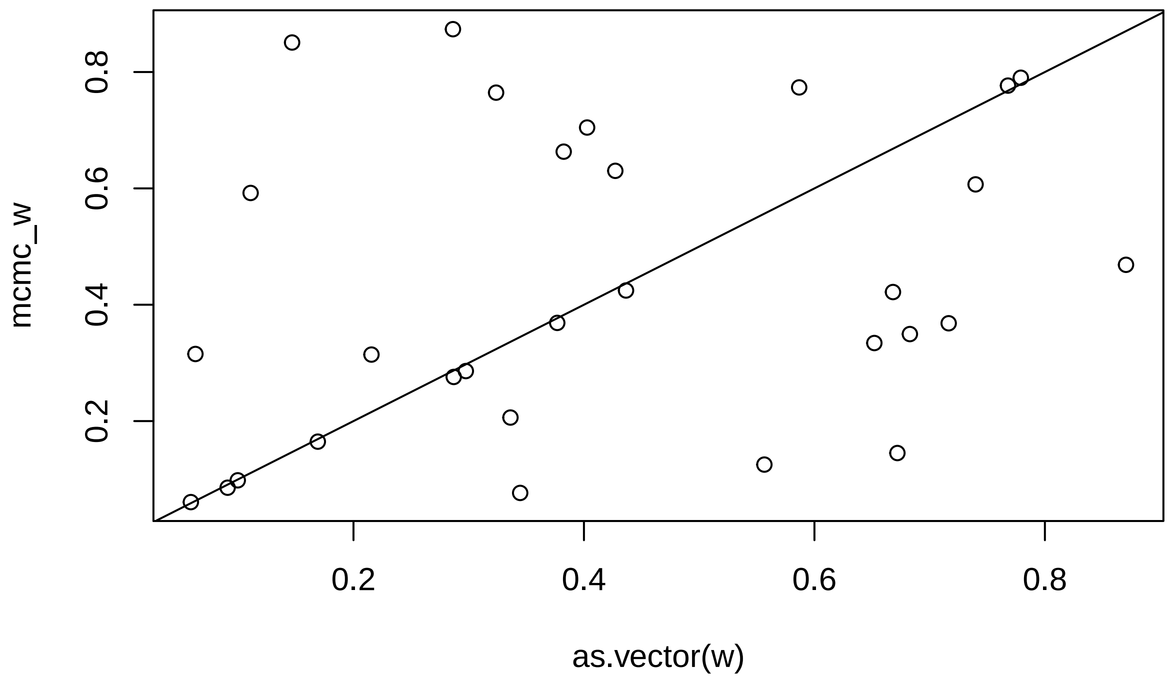


```r
## Figure out how jags stores the chains
y <- chains[1, grep("y", colnames(chains))]
y <- matrix(y, 50, 10)
all.equal(test.data$y, y) ##  each row (MCMC iteration) is stored in column-major order
```

Old code

Plot ω:

```r
mcmc_vals <- s$statistics
mcmc_w <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "w", "Mean"]
plot(as.vector(w), mcmc_w, type = "p")
abline(a=0, b=1)
```



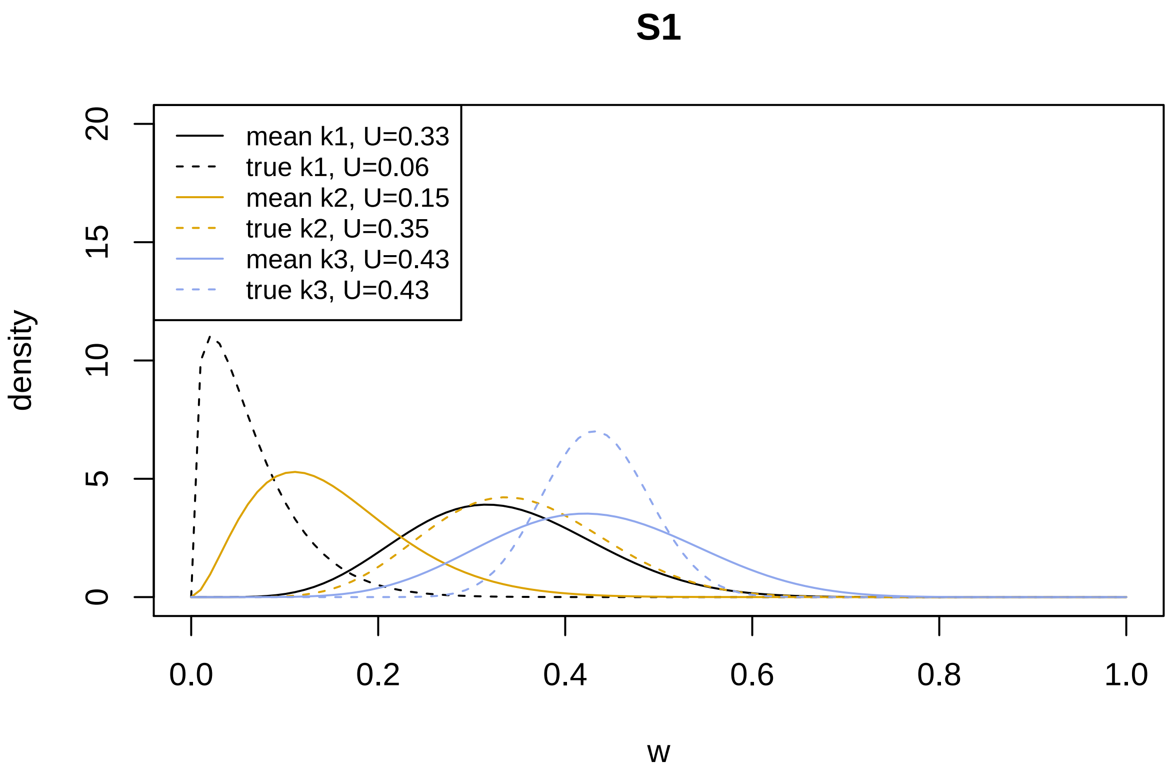
```
mcmc_z <- as.vector(mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "z", "Mean"])
#mcmc_z <- round(mcmc_z, 0)
##plot(z, mcmc_z, type = "p")
table(z, mcmc_z)
```

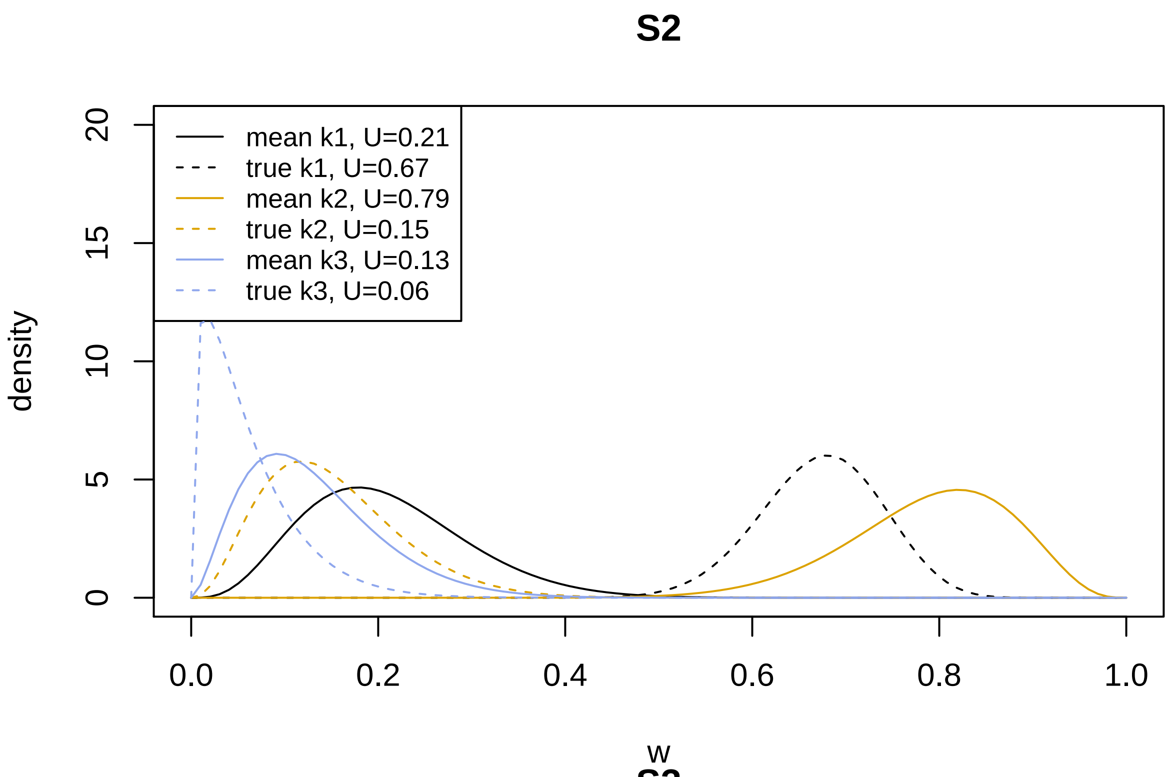
```
##    mcmc_z
## z    1  2  3
##   1  0  1  0
##   2  8  0  0
##   3  0  0 41
```

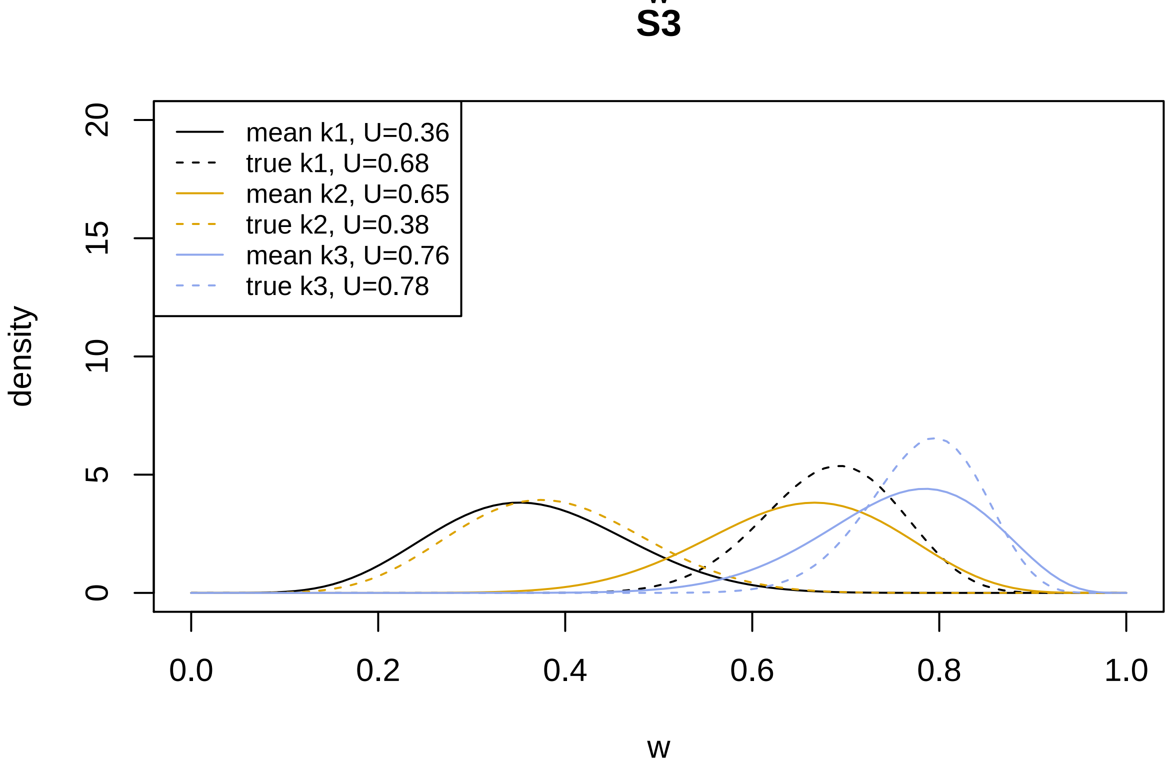
Overlay posterior predictive density on the true (simulated) density.

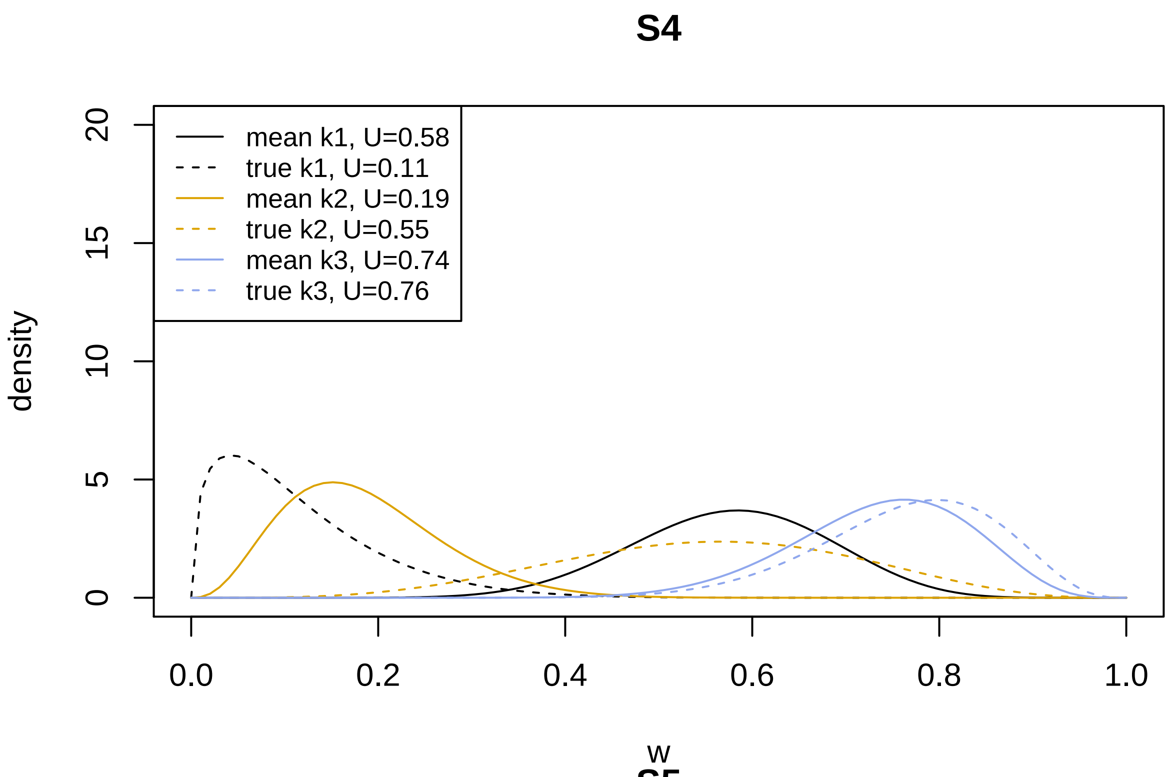
This should be a single plot with 10 facets (rows). Legend is very clunky. Need to make this simpler.

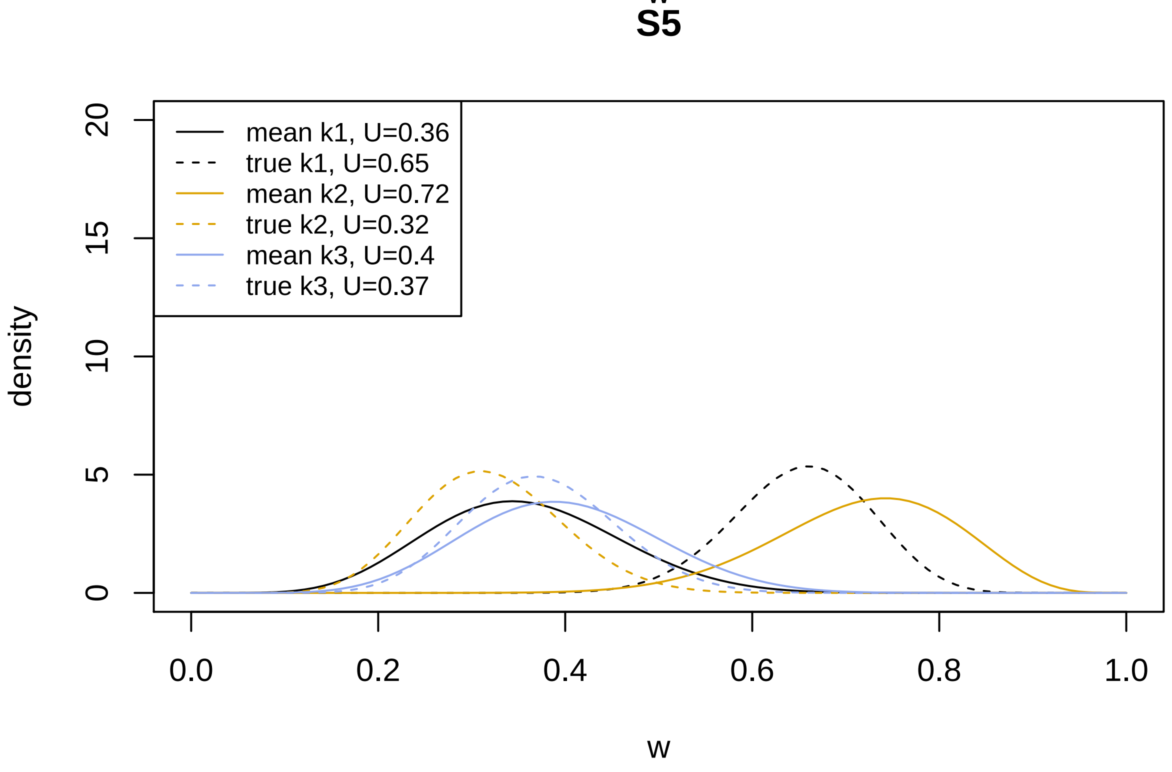
```
## summarizing U and V by posterior mean
mcmc_U <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "U", "Mean"]
mcmc_U <- matrix(mcmc_U, nrow=K)
mcmc_V <- mcmc_vals[substr(rownames(mcmc_vals), 1, 1) == "V", "Mean"]
mcmc_V <- matrix(mcmc_V, nrow=K)
samples_matrix <- as.matrix(samps)
getMaxDens <- function(vals) {
  d <- density(vals)
  round(d$x[which.max(d$y)], 2)
}
m <- apply(samples_matrix, 2, getMaxDens)
mode_U <- m[substr(names(m), 1, 1) == "U"]
mode_U <- matrix(mode_U, nrow=K)
mode_V <- m[substr(names(m), 1, 1) == "V"]
mode_V <- matrix(mode_V, nrow=K)
p <- seq(0, 1, length = 100)
colors <- c("#000000", "#DCA200", "#8FA7ED", "#9D847A", "#A47901")

for (s in 1:S) {
  for (k in 1:K) {
    if (k == 1) {
      # plot mcmc mean U,V
      plot(p, dbeta(p, mcmc_U[k,s] * mcmc_V[k,s], (1-mcmc_U[k,s])*mcmc_V[k,s]),
           main = paste0("S", s),
           ylab = "density", xlab = "w", type = "l", col = colors[k],
           ylim = c(0, 20))
      # plot truth
      lines(p, dbeta(p, a.ordered[k,s], b.ordered[k,s]), type = "l", col = colors[k], lty=2)
      # add legend
      allU <- round(as.vector(rbind(mcmc_U[,s], U.ordered[,s])), digits = 2)
      legend(x = "topleft",
             legend = paste0(c("mean k", "true k"), rep(1:K, each=2), ", U=", allU),
             col = colors[rep(1:K, each=2)],
             lty = rep(1:2, K),
             cex=0.8)
    } else {
      # plot mcmc mean U,V
      lines(p, dbeta(p, mcmc_U[k,s] * mcmc_V[k,s], (1-mcmc_U[k,s])*mcmc_V[k,s]),
            type = "l", col = colors[k])
      # plot truth
      lines(p, dbeta(p, a.ordered[k,s], b.ordered[k,s]), type = "l", col = colors[k], lty=2)
    }
  }
}
```

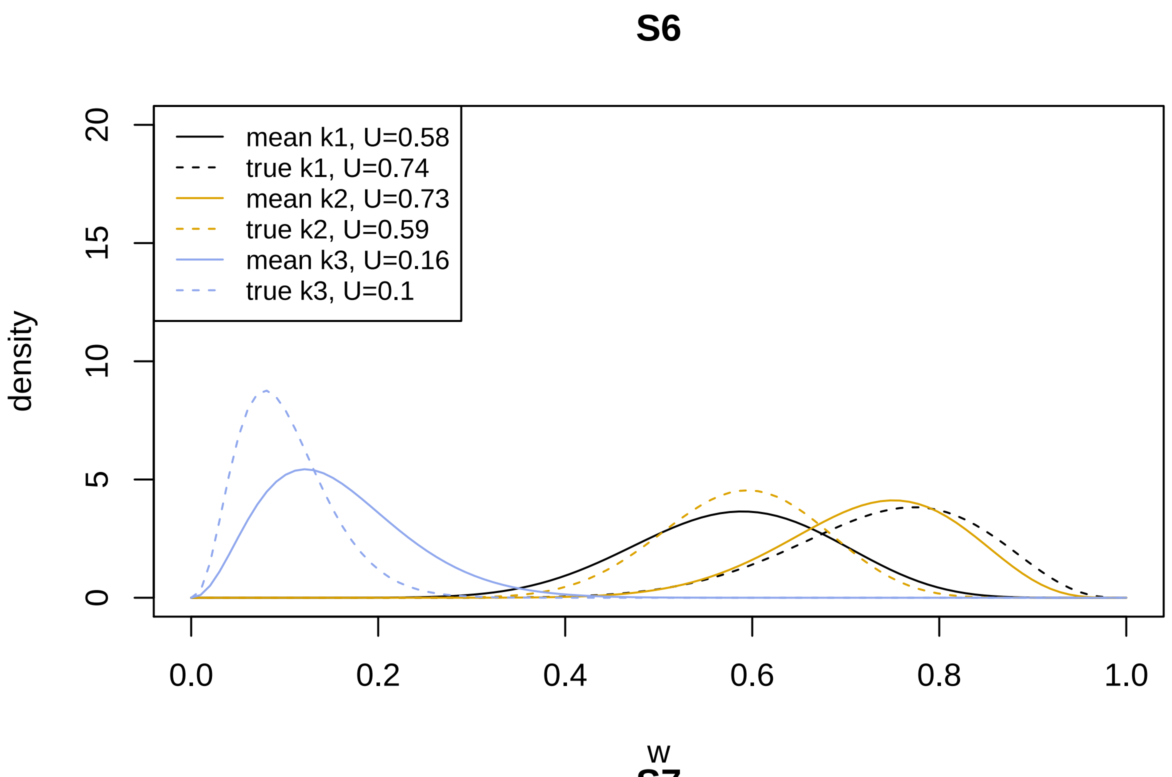


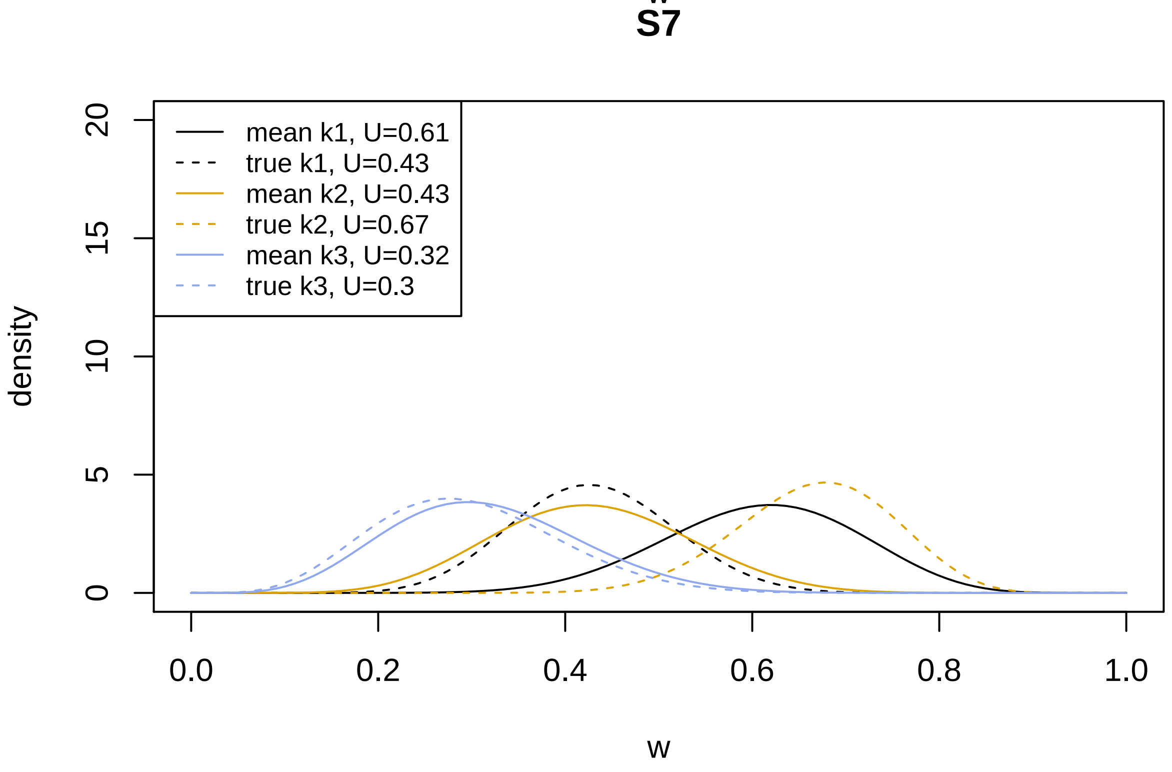

S2
mean k1, U=0.21
true k1, U=0.67
mean k2, U=0.79
true k2, U=0.15
mean k3, U=0.13
true k3, U=0.06
density
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
w

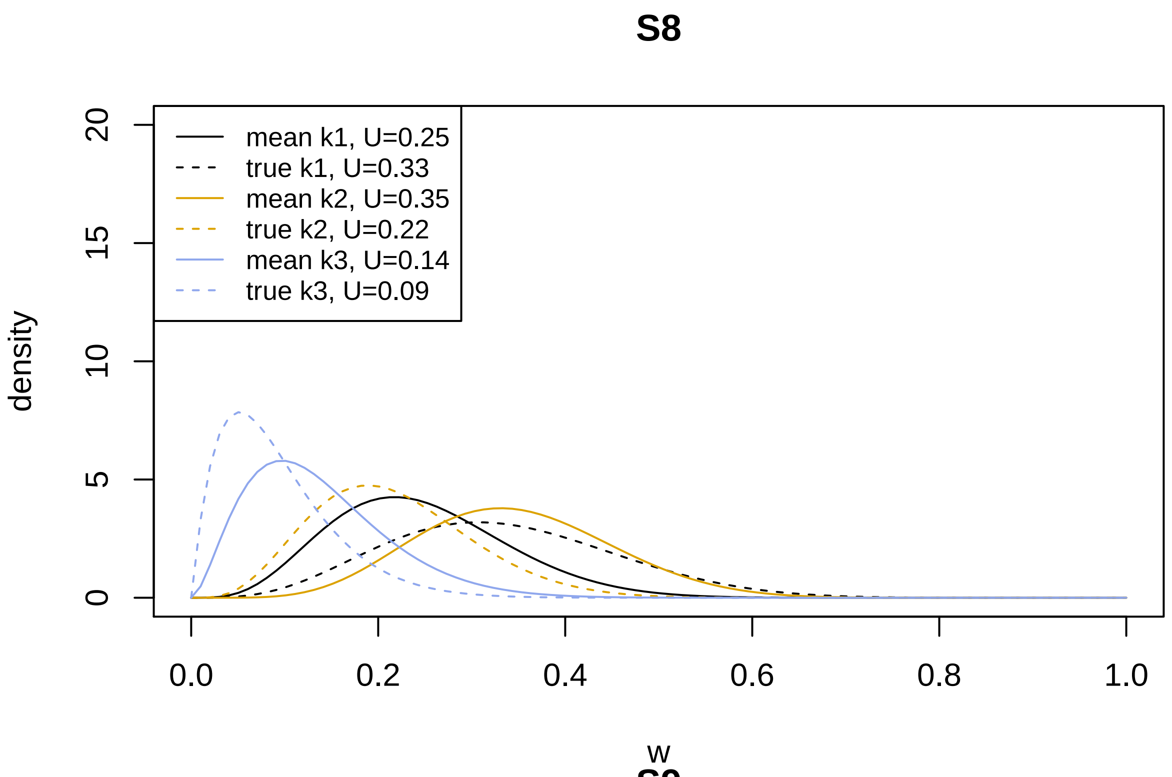

S3
mean k1, U=0.36
true k1, U=0.68
mean k2, U=0.65
true k2, U=0.38
mean k3, U=0.76
true k3, U=0.78
density
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
w

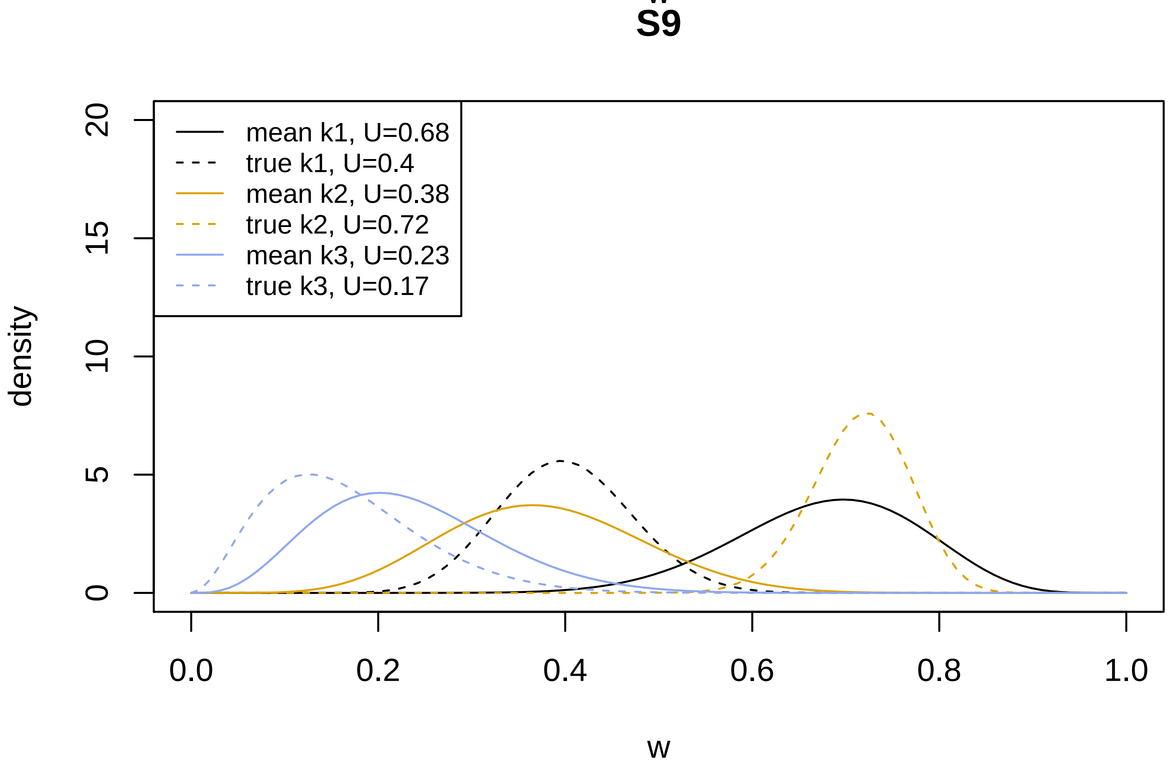

S4
density
w
mean k1, U=0.58
true k1, U=0.11
mean k2, U=0.19
true k2, U=0.55
mean k3, U=0.74
true k3, U=0.76

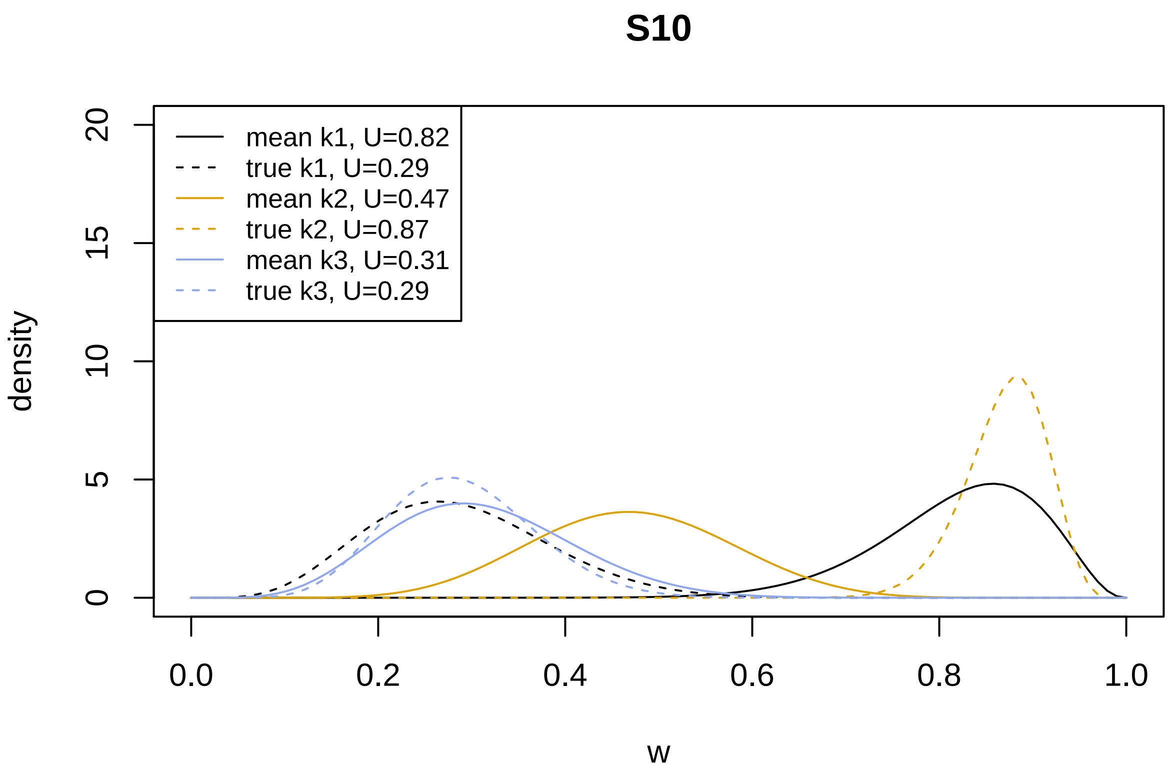

S5
density
w
mean k1, U=0.36
true k1, U=0.65
mean k2, U=0.72
true k2, U=0.32
mean k3, U=0.4
true k3, U=0.37


S6
mean k1, U=0.58
true k1, U=0.74
mean k2, U=0.73
true k2, U=0.59
mean k3, U=0.16
true k3, U=0.1
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20


S7
mean k1, U=0.61
true k1, U=0.43
mean k2, U=0.43
true k2, U=0.67
mean k3, U=0.32
true k3, U=0.3
density
w
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20


S8
mean k1, U=0.25
true k1, U=0.33
mean k2, U=0.35
true k2, U=0.22
mean k3, U=0.14
true k3, U=0.09
density
w


S9
mean k1, U=0.68
true k1, U=0.4
mean k2, U=0.38
true k2, U=0.72
mean k3, U=0.23
true k3, U=0.17
density
w


S10
mean k1, U=0.82
true k1, U=0.29
mean k2, U=0.47
true k2, U=0.87
mean k3, U=0.31
true k3, U=0.29
density
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0
w