

```

generate.t <- function(h0, beta0, u, z, alpha, n) {
  if(beta0 == 1) {
    beta <- c(1, 1, 1) / 3
  } else if (beta0 == 2) {
    beta <- c(2, -1, 0) / 3
  }
  beta <- beta / sqrt(sum(beta^2))
  mix.frac <- 0.5
  mu1 <- -2
  mu2 <- 3
  sigma2.1 <- 1
  sigma2.2 <- 1
  mus <- c(mu1, mu2)
  sds <- sqrt(c(sigma2.1, sigma2.2))
  components <- sample(1:2, prob = c(mix.frac, 1 - mix.frac), size = n, replace = TRUE)
  e <- rnorm(n, mean = mus[components], sd = sds[components])
  h0t <- 2 * e - z %*% beta
  if(h0 == 1) {
    t <- exp((h0t + 10) / 20)
  } else if(h0 == 2) {
    t <- apply(cbind((h0t + 40) / 20, 0), 1, max)
  }
}

sim.data.proj.inter <- function(h0, beta0, samp.size) {
  alpha <- 2
  data <- matrix(NA, nrow = samp.size, ncol = 9)
  data[, 1:2] <- rnorm(samp.size * 2, mean = 0, sd = sqrt(5))
  data[, 3] <- rnorm(samp.size, mean = 0, sd = sqrt(3)) + (data[, 1] * data[, 2]) / 3
  u.vec <- runif(samp.size)
  data[, 4] <- generate.t(h0 = h0, beta0 = beta0, u = u.vec,
                         z = data[, 1:3], alpha = alpha, n = samp.size)
  cut <- seq(min(data[, 4]), max(data[, 4]), (max(data[, 4]) - min(data[, 4])) / 6)
  data[, 5] <- runif(samp.size, min = cut[2], max = cut[3])
  data[, 6] <- runif(samp.size, min = cut[3], max = cut[4])
  data[, 7] <- runif(samp.size, min = cut[4], max = cut[5])
  data[, 8] <- runif(samp.size, min = cut[5], max = cut[6])
  data[, 9] <- runif(samp.size, min = cut[6], max = cut[7])
  cen <- runif(samp.size, min = 0, max = 500)
  delta <- cen > data[, 4]
  delta5 <- data[, 5] > data[, 4]
  delta6 <- data[, 6] > data[, 4]
  delta7 <- data[, 7] > data[, 4]
  delta8 <- data[, 8] > data[, 4]
  delta9 <- data[, 9] > data[, 4]
  final.data <- data.frame(
    cen, data[, 5], data[, 6], data[, 7], data[, 8], data[, 9],
    delta, delta5, delta6, delta7, delta8, delta9, data[, 1:3], data[, 4])
  colnames(final.data) <- c(
    "cen.time", "cen1", "cen2", "cen3", "cen4", "cen5", "delta",
    "delta1", "delta2", "delta3", "delta4", "delta5", "z1", "z2", "z3", "t")
  return(final.data)
}

```

```

set.seed(1234)
sim <- sim.data.proj.inter(h0 = 2, beta0 = 1, samp.size = 1e3)
DF <- data.frame(
  d1 = as.integer(sim$delta1),
  d2 = as.integer(sim$delta2),
  d3 = as.integer(sim$delta3),
  d4 = as.integer(sim$delta4),
  d5 = as.integer(sim$delta5),
  c1 = sim$cen1,
  c2 = sim$cen2,
  c3 = sim$cen3,
  c4 = sim$cen4,
  c5 = sim$cen5,
  z1 = sim$z1,
  z2 = sim$z2,
  z3 = sim$z3)
data.table::data.table(DF)

##      d1 d2 d3 d4 d5      c1      c2      c3      c4      c5
## 1:  0  0  0  1  1 1.466907 1.763902 2.078570 2.352064 2.633078
## 2:  0  1  1  1  1 1.541113 1.708809 2.081996 2.366938 2.586009
## 3:  0  0  0  0  1 1.581326 1.803390 2.054550 2.233843 2.656011
## 4:  0  0  1  1  1 1.667167 1.792692 2.179182 2.293049 2.626627
## 5:  0  0  1  1  1 1.460037 1.779462 1.982407 2.415980 2.548397
##    ---
## 996:  0  0  1  1  1 1.507676 1.764975 2.152182 2.263013 2.681040
## 997:  0  0  0  1  1 1.571493 1.772272 2.007120 2.371708 2.599965
## 998:  0  1  1  1  1 1.579383 1.732190 2.103496 2.238966 2.519050
## 999:  0  0  0  0  1 1.561269 1.786735 2.074223 2.283763 2.665592
## 1000: 0  0  0  1  1 1.445756 1.712838 2.193099 2.351227 2.706236
##          z1      z2      z3
## 1: -2.6990811 -2.6952075  0.73815782
## 2:  0.6203506  0.6741001 -0.03317348
## 3:  2.4248842 -3.4416333 -2.97365274
## 4: -5.2451395  1.4207321 -0.41903780
## 5:  0.9595520  1.5718480 -2.36532196
##    ---
## 996: -0.4837055  0.5852277  2.86133578
## 997:  1.2295155  2.9844677 -1.41172090
## 998:  1.0794310  0.5539783  2.18584544
## 999:  1.7013095 -3.0763485 -1.33854561
## 1000: -0.9965216 -2.6621279  1.82517192

saveRDS(DF, "simdata.rds")

```