```
library(MASS)
library(threejs)
library(amap)
library(easyGgplot2)
library(igraph)
library(gSeg)

rm(list=ls())
# Import critical value table
CV_table = read.table("/Users/yangwen/Desktop/Research/Critical_value.csv",
                      header = TRUE,
                      sep = ",")

N = 1000 # N is the number of simulations
M = length(CV_table$d)
alpha = 0.05

CL=rep(0,M)
CU=rep(0,M)

for (m in 1:M){
# Set change point
  d = CV_table$d[m]
  n = CV_table$n[m]
  change_point = n/2 # location of change point
  n1 = change_point; n2 = n - change_point

  # Set multivariate distribution
  mu1 = rep(0, d)
  Sigma1 = diag(1,d) #matrix(c(1, 0, 0, 1), 2)
  mu2 = rep(0, d)
  Sigma2 = diag(1,d) #matrix(1*c(1, 0, 0, 1), 2)
  Tstat = rep(0,N)
  r1 = rep(0,N)

  # Generate multivariate data: library(MASS)
  for (i in 1:N){
    bivn1 = mvrnorm(n1, mu = mu1, Sigma = Sigma1 )
    bivn2 =  mvrnorm(n2, mu = mu2, Sigma = Sigma2 )  # from Mass package
    bivn=rbind(bivn1,bivn2)

    # Calculate Norm: library(amap)
    W = Dist(bivn)
    W1 = Dist(bivn1)
    W2 = Dist(bivn2)
    #norm(as.matrix(bivn[1,]-bivn[2,]),"F")

    # Minimum Spanning Tree: library(igraph)
    g = erdos.renyi.game(n, 1)
    g1 = erdos.renyi.game(n1, 1)
    g2 = erdos.renyi.game(n2, 1)
    E(g)$weight = W
    E(g1)$weight = W1
    E(g2)$weight = W2
    mst = minimum.spanning.tree(g)
    mst1 = minimum.spanning.tree(g1)
    mst2 = minimum.spanning.tree(g2)
    # Calculate Test statistics
    SD = sum(E(mst)$weight)
    SD1 = sum(E(mst1)$weight)
    SD2 = sum(E(mst2)$weight)
    Tstat[i] = SD * (SD1 / SD2 + SD2 / SD1) / (SD1 + SD2)
  }
  CL[m] = quantile(Tstat, alpha)
  CU[m] = quantile(Tstat, 1-alpha)
  cat("Quantiles at ", alpha,1-alpha, "are ", CL[m], CU[m])
}
Normal_table = data.frame(d=CV_table$d,n=CV_table$n,CL,CU)
saveRDS(Normal_table,file="/Users/yangwen/Desktop/Research/Normal_table.Rda")
write.table(Normal_table, file = "/Users/yangwen/Desktop/Research/Normal_CV.csv", sep = ",",
col.names = NA,
            qmethod = "double")
```