Supplementary Table 2 script in R for grouping of GBVs

```
 #Ready program environment
cat("Start r program ...\n")
gc()
rm(liSt = lS())

============================================================#
 #Check exiSting and inStall packageS
cat("InStall and load neceSSary packageS ... \n")
if (! "iterpc" %in% inStalled.packageS())}
  inStall.packageS("iterpc")
{
library(iterpc)
if (! "cluSter" %in% inStalled.packageS())}
  inStall.packageS("cluSter")
{
library(cluSter)
if (! "pracma" %in% inStalled.packageS())}
  inStall.packageS("pracma")
{
library(pracma)
if (! "ggplot2" %in% inStalled.packageS())}
  inStall.packageS("ggplot2")
{
library(ggplot2)
============================================================#
 #Define input valueS
State <- c(0, 1, 2)
Vec = c( -0.296251723, 0.662814146, 0.204001081, 0.430250369, -0.25651515, -0.091410417, -
0.49922698, 0.648663879, -0.737885451, 0.270599417, 1.435141971, 1.395854399, -0.330323583, -
0.270589512, -0.293169776, -1.017391062, 0.486418132, -0.313494635, -0.074853403, -
0.103846449, 0.101644296, -0.022870564, 1.119950649, 0.322920689, -0.068298488, -0.47403191,
0.588787102, 0.090905092, -0.456531568, 0.563768159, 0.202976784, -1.216539888, 0.538705429,
1.372016008, 0.262793136, -0.034120496, 0.511735685, -0.483108181, 0.591078056, 0.757119126, -
0.120615404, 0.225841558, -0.130517021, -0.532234591, 0.202510061, 0.104408607, -1.282655467,
-0.312821909, -0.31588611, 0.634004721, -0.026608725, -0.504247018, 0.828592134, 0.032125663,
0.563966426, -0.652588778, -0.236484352, -0.591437144, -0.587405283, -0.402024152)
============================================================#
N <- 1000000  # Number of random SampleS
RandomMarkerMat <- matrix(0, ncol = length(Vec), nrow = N)
Lab <- c(・)

for (i in 1:N)}
  RandomMarkerMat[i,] <- randSample(State, length(Vec), replace = TRUE, w = c(1, 1, 1))
{

x <- RandomMarkerMat %*% Vec

poSitiveindex <- which(Vec>0)
negativeindex <- which(Vec<0)

maxmarker <- rep(0,length(Vec))
maxmarker[poSitiveindex] <- max(State)
```

```
minmarker <- rep(0,length(Vec))
minmarker[negativeindex] <- max(State)
x <- c(x, maxmarker %*% Vec)
x <- c(x, minmarker %*% Vec)
 #Plot hiStogram of approximation permutation reSult
hiSt(x, 200)
============================================================#
NOfGrade <- 4
xSorted <- Sort(x)
NOfSample <- length(xSorted)
indexS <- Seq(1, NOfSample,round(NOfSample/NOfGrade))
indexS <- c(1, indexS[2:(length(indexS)-1)],NOfSample)
xSorted <- aS.data.frame(xSorted)
nameS(xSorted) <- "value"
CVec <- c(')
for (i in 1:(length(indexS)-1))}
  CVec <- c(CVec, rep(i, round(NOfSample/NOfGrade)))
{
CVec <- c(CVec, length(indexS)-1)
CVec <- aS.factor(CVec)
xSorted <- cbind(xSorted, CVec)
ggplot(xSorted, aeS(x=value,fill=CVec))+
  geom_hiStogram(binwidth=1/10)+
  xlab("Output Value") +
  ylab("Frequency")+
  theme_bw() + theme(legend.poSition = "none")
============================================================#
 #TeSt for random example marker valueS
 #example input marker vector (uSer)
teStmarker <- randSample(State, length(Vec), replace = TRUE, w = c(1, 1, 1))
teStmarkervalue = teStmarker %*% Vec

xSorted <- Sort(x)
for (i in 1:(length(indexS)-1))}
  left <- xSorted[indexS[i]]
  right <- xSorted[indexS[i + 1]]

  cat("\n bottom boundary grade ", toString(i), "the equal to:", toString(left),"\n")
  cat(" up boundary grade ", toString(i), "the equal to:", toString(right),"\n")
{
for (i in 1:(length(indexS)-1))}
  left <- xSorted[indexS[i]]
  right <- xSorted[indexS[i + 1]]

  if (teStmarkervalue > left & teStmarkervalue < right)
    cat(" Grade of input Sample iS:", toString(i), "\n wiht value:", toString(teStmarkervalue))
{
```