import libraries

```
suppressMessages(library(spatialEco))
suppressMessages(library(readxl))
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
library(parallel)
```

```
mymerge135 <- read_excel("/Users/amogh/Documents/UTD/Spring 2023/STAT Research/mymerge135.xlsx")
vars <- c("CVL", "CVR", "NPiL", "NPiR")
mymerge135 %>% select(vars) %>% summary()
```

```
#example code to view shape of real data
mymerge135 %>% select(c(SID, CVL, CVR)) %>% group_by(SID) %>% summarise(CVL = max(CVL, na.rm = TRUE), C
```

skeleton doExp code

```
cp = 1
dat <- simDat(100, cp)
cp.vec <- seq(0, 6, .1)[-1]
cons <- matrix(NA, nrow = length(cp.vec), ncol = length(cp.vec))

    # rows are X1(i), columns at X2(j)
rownames(cons) <- cp.vec
colnames(cons) <- cp.vec

for (i in 1:length(cp.vec)) {
  for (j in 1:length(cp.vec)) {
    fit <- glm(Y ~ I(X1 < cp.vec[i]) + I(X2 < cp.vec[j]) + X3 + X4 + X5, binomial, dat)
    cons[i, j] <- concordance(dat$Y, predict(fit, type = "response"))$con
  }
}
which.max(cons)
cons[which.max(cons)]
```

my implementation

```
# set random seed, find extreme case and test
set.seed(0)

#create dataset with exp with mean 2
simDatExp <- function(n, cp = 1) {
  dat <- data.frame(matrix(rexp(n * 5, rate = 0.5), n)) #set distribution X vars come from
  dat$X1 <- round(dat$X1, digits = 1)
  dat$X2 <- round(dat$X2, digits = 1)
  xb <- (dat$X1 > cp) + (dat$X2 > cp) + dat$X3 - dat$X4
```

```
  dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
  return(dat)
}

#replicate several times
doExp <- function(n) {
    cp = 1
    dat <- simDatExp(200, cp) # increased to 200 from 100
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- matrix(NA, nrow = length(cp.vec), ncol = length(cp.vec))

    # rows are X1(i), columns at X2(j)
    rownames(cons) <- cp.vec
    colnames(cons) <- cp.vec

    for (i in 1:length(cp.vec)) {
      for (j in 1:length(cp.vec)) {
        fit <- glm(Y ~ I(X1 < cp.vec[i]) + I(X2 < cp.vec[j]) + X3 + X4 + X5, binomial, dat)
        cons[i, j] <- concordance(dat$Y, predict(fit, type = "response"))$con

      }
    }
    find <- which.max(cons)
    column <- ceiling(find / length(cp.vec))
    column
    row <- find - ((column - 1) * length(cp.vec))
    row
    #print(find)
    #print(length(cp.vec))
    #print(column)
    #print(row)
    #cons[which.max(cons)]
    #coord <- c(which.max(cons), cp.vec[column], cp.vec[row])
    coord <- c(cp.vec[column], cp.vec[row])
    return (coord)
}

cp = 1
propExp <- simDatExp(1e5, cp) %>% filter(X1>cp, X2>cp) %>% count() %>% as.numeric() / 1e5
propYExp <- simDatExp(1e5, cp) %>% filter(Y==1) %>% count() %>% as.numeric() / 1e5

propExp
```

```
## [1] 0.34853
```

```
propYExp
```

```
## [1] 0.65692
```

```
#fooExp <- replicate(1, doExp())
#fooExp <- t(fooExp)
#fooExp
#summary(fooExp)
```

```
c1 <- makePSOCKcluster(detectCores())
setDefaultCluster(c1)
clusterExport(NULL, "simDatExp")
clusterExport(NULL, "doExp")
clusterExport(NULL, "concordance")
test <- parSapply(NULL, 1:100, FUN = doExp)
test
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]  2.4  3.6  1.2  1.2  3.8  1.1  1.1  1.0  1.6   1.1   1.2   1.0   0.6   0.7
## [2,]  1.0  1.1  1.6  1.6  1.1  2.1  0.1  5.8  1.0   1.6   1.7   1.1   1.2   1.1
##      [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25] [,26]
## [1,]   1.2   1.7   1.1   1.1   3.2   1.1   0.2   5.6   1.0   1.1   2.2   1.1
## [2,]   0.9   1.0   4.1   1.0   0.6   1.1   1.1   2.1   1.1   4.6   0.9   0.6
##      [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38]
## [1,]   1.2   5.0   1.7   1.0   5.5   1.7   1.2   0.9   3.8   1.7   1.1   0.3
## [2,]   1.0   1.4   1.1   1.1   1.1   2.4   1.1   1.1   0.9   1.0   1.0   0.7
##      [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]
## [1,]   0.7   5.1   1.0   0.9   3.2   0.9   2.4   5.8   1.1   1.1   2.9   0.1
## [2,]   2.3   2.2   1.1   0.2   0.3   5.5   2.7   1.6   1.2   3.4   1.0   2.6
##      [,51] [,52] [,53] [,54] [,55] [,56] [,57] [,58] [,59] [,60] [,61] [,62]
## [1,]   1.1   0.9   2.9   0.9   1.1   1.2   1.1   0.9   1.4   1.2   1.4   1.2
## [2,]   1.1   1.1   0.5   1.1   1.1   0.7   3.8   0.2   1.1   1.1   0.7   0.3
##      [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72] [,73] [,74]
## [1,]   1.1   1.2   1.2   1.0   0.9   2.1   1.7   1.2   1.2   0.6   1.1   0.2
## [2,]   0.7   1.1   1.2   0.9   0.5   1.0   5.8   0.3   0.1   2.7   1.1   2.6
##      [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85] [,86]
## [1,]   1.1   1.1   5.1   0.1   1.7   1.1   1.1   1.1   1.1   1.0   1.2   1.6
## [2,]   2.8   1.1   1.1   1.1   1.1   3.2   1.1   4.5   2.6   1.1   0.5   2.3
##      [,87] [,88] [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97] [,98]
## [1,]   1.0   1.6   1.2   1.1     1   3.3   1.1   4.3   1.6   1.1   2.7   1.1
## [2,]   1.6   3.7   0.2   0.5     1   0.7   4.8   0.6   5.5   2.8   1.2   1.2
##      [,99] [,100]
## [1,]   1.6    4.6
## [2,]   0.7    1.9
```

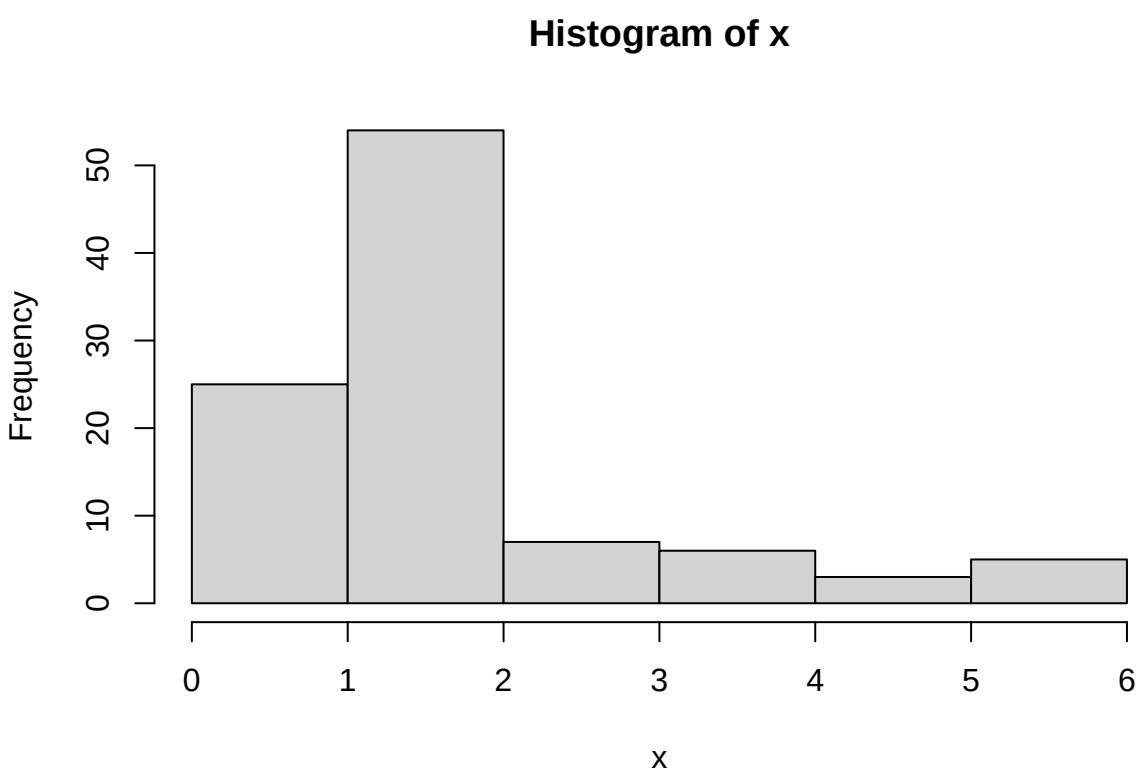
```
test <- t(test)
summary(test)
```

```
##        V1              V2
##  Min.   :0.100   Min.   :0.100
##  1st Qu.:1.075   1st Qu.:0.975
##  Median :1.100   Median :1.100
##  Mean   :1.659   Mean   :1.589
##  3rd Qu.:1.700   3rd Qu.:1.950
##  Max.   :5.800   Max.   :5.800
```

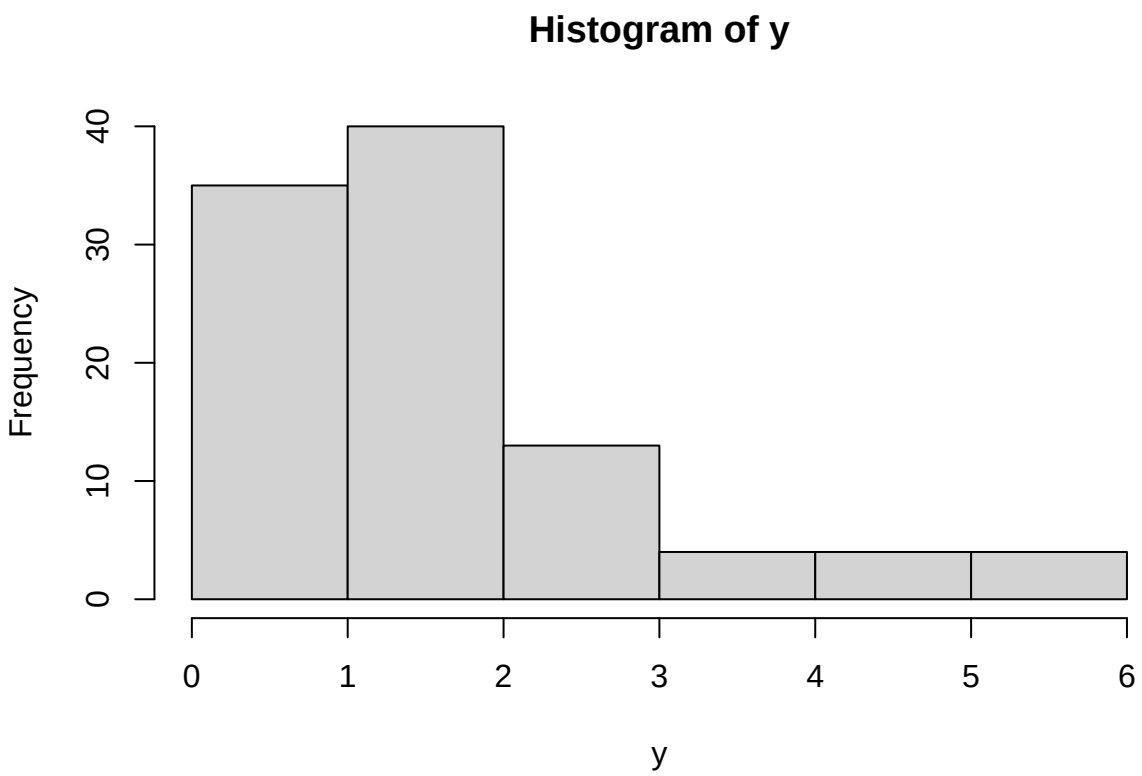
```
stopCluster(c1)
```

```
x <- test[, 1]
y <- test[, 2]

hist(x)
```




```
hist(y)
```


Histogram of y
Frequency
0
10
20
30
40
0
1
2
3
4
5
6
y