import libraries

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

existing simuation code

```
set.seed(1)

simDat <- function(n, cp = 1) {
    dat <- data.frame(matrix(rexp(n * 5), n)) #set what distribution X vars come from
    xb <- (dat$X1 > cp) + dat$X2 - dat$X3 # i + X2 - X3, if true: 1 else: 0
    #dat$xb <- xb
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}
```

test simulation

```
output <- simDat(10)
output
```

```
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)
```

```
## [1] 430
```

```
cons[which.max(cons)]
```

```
## [1] 0.7955936
```

my implementation

```
cp = 1
#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 # need to review
  dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb))) # why exp(-xb)
  return(dat)
}
#replicate several times
doExp <- function() {
    dat <- simDatExp(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
      }
    }
    cp.vec[which.max(cons)]
}

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] 0.34924
```

```
propYExp
```

```
## [1] 0.65799
```

```
#fooExp <- replicate(1e2, doExp())
#summary(fooExp)
```

Things to work on: - Need to verify correct formula for dat$Y (line 78)
- Identify correct equation (line 77)
- refine exponential model, expand to other distributions