sim

Courtney Schaller

6/20/2021

```
library(spatialEco)
library(ggplot2)

#' Simulated data:
#' Assume there are 5 time-independent (exponential) covariates; X1, X2, X3, X4, X5
#' Assume X1 is the important covariate that we want to find a cutpoint of.
#' Think of this as the constriction velocity in the dataset.
#' Assume X2 and X3 have significant effect while X4 and X5 are nuisance variables.
#'
#' @param n is the sample size
#' @param cp is the true cutpoint (we assume we know this in the simulation,
#' but is unknown in real life data)


#create dataset
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$X3
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}
```

```
dat <- simDat(100)

cp.vec <- seq(0, 6, .1)[-1]
cons <- rep(NA, length(cp.vec))

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

```
## [1] 0.9
```

```
#replicate several times
do <- function() {
    dat <- simDat(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I(X1 < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat)
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
```

```
    cp.vec[which.max(cons)]
}

foo <- replicate(1e3, do())
foo100 <- replicate(1e2, do())
summary(foo)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   0.775   1.000   1.222   1.500   6.000
```

```
## Tree method?
library(partykit)
```

```
## Loading required package: grid
```

```
##
## Attaching package: 'grid'
```

```
## The following object is masked from 'package:spatialEco':
##
##     explode
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
ctree(Y ~ X1 + X2 + X3 + X4 + X5, dat)
```

```
##
## Model formula:
## Y ~ X1 + X2 + X3 + X4 + X5
##
## Fitted party:
## [1] root
## |   [2] X3 <= 0.77155: 0.745 (n = 47, err = 8.9)
## |   [3] X3 > 0.77155
## |   |   [4] X2 <= 1.36348
## |   |   |   [5] X1 <= 0.90451: 0.000 (n = 22, err = 0.0)
## |   |   |   [6] X1 > 0.90451: 0.267 (n = 15, err = 2.9)
## |   |   [7] X2 > 1.36348: 0.750 (n = 16, err = 3.0)
##
## Number of inner nodes:    3
## Number of terminal nodes: 4
```

```
ctree(Y ~ X1 + X2 + X3 + X4 + X5, dat, control = ctree_control(stump = TRUE))
```

```
##
## Model formula:
## Y ~ X1 + X2 + X3 + X4 + X5
##
## Fitted party:
## [1] root
## |   [2] X3 <= 0.77155: 0.745 (n = 47, err = 8.9)
## |   [3] X3 > 0.77155: 0.302 (n = 53, err = 11.2)
##
## Number of inner nodes:    1
## Number of terminal nodes: 2
```

Familiarize self with example data for general ranges of variables, e.g. CV ranges from 0 to 4

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     do
```

```
## The following object is masked from 'package:spatialEco':
##
##     combine
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
mymerge135 <- read_excel("../data/mymerge135.xlsx")

vars <- c("CVL", "CVR", "NPiL", "NPiR")
mymerge135 %>% select(vars) %>% summary()
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(vars)` instead of `vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
##       CVL             CVR             NPiL            NPiR
##  Min.   :0.220   Min.   :0.420   Min.   :0.000   Min.   :0.000
##  1st Qu.:1.220   1st Qu.:1.140   1st Qu.:3.075   1st Qu.:3.400
##  Median :2.450   Median :2.310   Median :4.300   Median :4.400
##  Mean   :2.373   Mean   :2.294   Mean   :3.305   Mean   :3.856
##  3rd Qu.:3.220   3rd Qu.:3.005   3rd Qu.:4.400   3rd Qu.:4.600
##  Max.   :4.040   Max.   :3.660   Max.   :4.600   Max.   :4.900
##  NA's   :25      NA's   :11      NA's   :3       NA's   :2
```

Test rounding X1 to nearest tenth

```
#replicate several times
doRound <- function() {
    dat <- simDat(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I(round(X1, digits = 1) < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat) #ad
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
    cp.vec[which.max(cons)]
}

roundedX1 <- replicate(1e2, doRound())
summary(roundedX1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   0.900   1.100   1.248   1.600   3.800
```

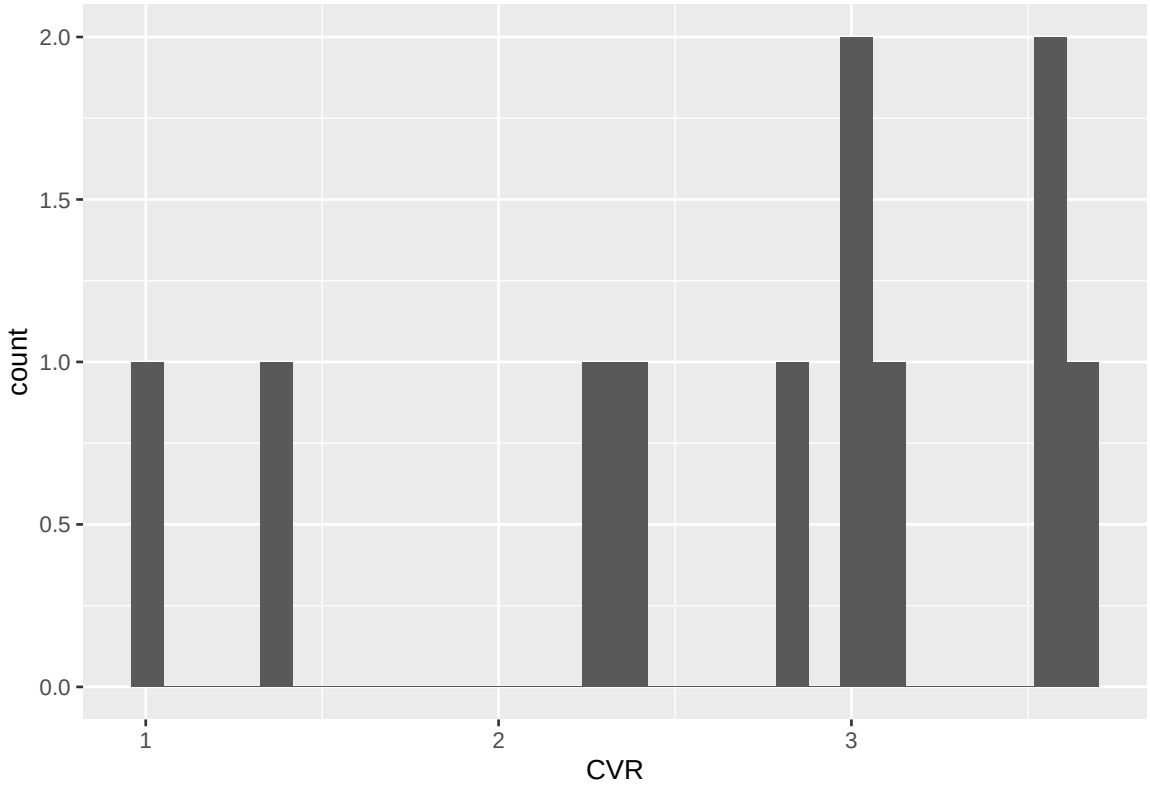
try different positive distribution for X1 (exponential with different mean, gamma, uniform, beta, half-normal, etc)

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

```
## Warning in max(CVL, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```

```
## Warning in max(CVL, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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

#Create other simDat functions

#gamma with params 2 & 0.5
simDatGamma <- function(n, cp = 1, shape = 2, scale = 0.5) {
    dat <- data.frame(matrix(rgamma(n*5, shape, scale), n)) #set what distribution X vars come from
    xb <- (dat$X1 > cp) + dat$X2 - dat$X3
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}
```

```
#uniform from 0 to 4
simDatUnif <- function(n, cp = 1, min = 0, max = 4) {
    dat <- data.frame(matrix(runif(n*5, min, max), n)) #set what distribution X vars come from
    xb <- (dat$X1 > cp) + dat$X2 - dat$X3
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}
#beta with params 2 and 2 (multiplied by four to range 0 to 4 instead of 0 to 1)
simDatBeta <- function(n, cp = 1, shape1 = 2, shape2 = 2) {
    dat <- data.frame(matrix(4*rbeta(n*5, shape1, shape2), n)) #set what distribution X vars come fr
    xb <- (dat$X1 > cp) + dat$X2 - dat$X3
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}

#half-normal (standard)
simDatNorm <- function(n, cp = 1, mean = 1, sd = 1) {
    dat <- data.frame(matrix(abs(rnorm(n*5, mean, sd)), n)) #set what distribution X vars come from
    xb <- (dat$X1 > cp) + dat$X2 - dat$X3
    dat$Y <- rbinom(n, 1, 1 / (1 + exp(-xb)))
    return(dat)
}
```

test new simDat funcs

```
#replicate several times
doExp <- function() {
    dat <- simDatExp(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I((X1) < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat)
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
    cp.vec[which.max(cons)]
}

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

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fooExp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   0.700   1.200   1.862   2.550   6.000
```

```
#why am I getting separation?
doGamma <- function() {
    dat <- simDatGamma(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
```

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

fooGamma <- replicate(1e2, doGamma())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fooGamma)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.200   0.900   2.050   2.378   3.625   5.900
```

```
doUnif <- function() {
    dat <- simDatUnif(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I((X1) < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat)
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
    cp.vec[which.max(cons)]
}

fooUnif <- replicate(1e2, do())
summary(fooUnif)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   0.800   1.000   1.229   1.400   4.100
```

```
doBeta <- function() {
    dat <- simDatBeta(100)
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I((X1) < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat)
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
    cp.vec[which.max(cons)]
}

fooBeta <- replicate(1e2, doBeta())
summary(fooBeta)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.200   0.875   1.200   1.641   2.600   3.800
```

```
doNorm <- function() {
    dat <- simDatNorm(100)
```

```
    cp.vec <- seq(0, 6, .1)[-1]
    cons <- rep(NA, length(cp.vec))
    for (i in 1:length(cp.vec)) {
        fit <- glm(Y ~ I((X1) < cp.vec[i]) + X2 + X3 + X4 + X5, binomial, dat)
        cons[i] <- concordance(dat$Y, predict(fit, type = "response"))$con
    }
    cp.vec[which.max(cons)]
}

fooNorm <- replicate(1e2, doNorm())
summary(fooNorm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   0.800   1.000   1.195   1.300   3.200
```

results

```
library(reshape2)
df <- data.frame(foo100, fooBeta, fooExp, fooGamma, fooNorm, fooUnif, roundedX1)
dfmelt <- melt(df)
```

```
## No id variables; using all as measure variables
```

```
ggplot(dfmelt, aes(x = value, color = variable)) + geom_density()
```