

```

# devtools::install_github("celehs/PanelCurrentStatus")
library(PanelCurrentStatus)

data <- readRDS("data.rds")
head(data)

##   d1 d2 d3 d4 d5      c1      c2      c3      c4      c5      z1
## 1  0  0  0  1  1 1.466907 1.763902 2.078570 2.352064 2.633078 -2.6990811
## 2  0  1  1  1  1 1.541113 1.708809 2.081996 2.366938 2.586009  0.6203506
## 3  0  0  0  0  1 1.581326 1.803390 2.054550 2.233843 2.656011  2.4248842
## 4  0  0  1  1  1 1.667167 1.792692 2.179182 2.293049 2.626627 -5.2451395
## 5  0  0  1  1  1 1.460037 1.779462 1.982407 2.415980 2.548397  0.9595520
## 6  0  0  1  1  1 1.489140 1.856881 2.175501 2.363438 2.631624  1.1315754
##          z2      z3
## 1 -2.6952075 0.73815782
## 2  0.6741001 -0.03317348
## 3 -3.4416333 -2.97365274
## 4  1.4207321 -0.41903780
## 5  1.5718480 -2.36532196
## 6 -4.2616836 -3.41857940

n <- nrow(data)
delta <- as.matrix(data[, 1:5])
ctime <- as.matrix(data[, 5 + 1:5])
predictors <- as.matrix(data[, 10 + 1:3])

```

Calcuate the scaled coefficients from the conditional censoring logistic (CCL) estimator

```

fit <- ccl.fit(delta, ctime, predictors, n.ptb = 500, seed = 1)
data.frame(est = fit[[2]], est.se = fit[[3]])

```

```

##           est     est.se
## 1 0.6418898 0.04868211
## 2 0.5514986 0.05546564
## 3 0.5327539 0.05103419

```

Prepare the data for evaluating the model metrics from ROC curve via kernel smoothing

```

alldata <- NULL
for (k in 1:ncol(delta)) {
  DF <- data.frame(delta = delta[, k], ctime = ctime[, k], predictors)
  alldata <- rbind(alldata, DF)
}
t0 <- median(alldata$ctime)
h <- sd(alldata$ctime) / n^0.3
ans <- ccl.roc(alldata, fit, t0, h)
data.frame(roc = ans[[1]], roc.se = ans[[2]])

##           roc     roc.se
## auc.tilde.lower 0.6049827 0.02274278
## auc.tilde.upper 0.6049842 0.02274271
## tpr.05.tilde    0.1307085 0.02560484
## tpr.10.tilde    0.1799483 0.02859876

```

```
proc.time()
```

```

##    user  system elapsed
## 15.624   1.662  17.313

```