

```

## Code for Spanish ICU data analysis

## Load Required Packages

library(Epi)
library(gam)
library(multipleNCC)
library(peperr)
library(plyr)
library(stats)
library(survival)
library(lattice)
library(etm)
library(geepack)

CreateSamplestat <- function(nestinfo, data, samplestat, id, set) {

  ## Adjusts samplestat vector for use in multipleNCC package.
  ## Sampled controls are given "1."
  ##
  ##
  ## Args:
  ##   nestinfo:    data frame of information for reduced cohort after
  ##                IDS sampling
  ##   joininfo:    full cohort data frame before sampling
  ##   samplestat:  vector of sampling and status (case or control) info
  ##                before adjustment
  ##   id:          id for joining data frames
  ##   set:         time-matched case-control set
  ##
  ## Returns:
  ##   data frame for full cohort with adjusted samplestat vector

  ## marks patient as sampled
  nestinfo[, samplestat] <- replace (nestinfo[, samplestat],
                                     nestinfo[[samplestat]] == 0, 1)

  # gives columns consistent names for join function
  nccl <- data.frame( nestinfo[[id]], nestinfo[[samplestat]], nestinfo[[set]])
  colnames(nccl) <- c(id, samplestat, set)

  # removes duplicate event times
  nccl <- unique(nccl[!duplicated(nccl[-3]),,] )

  # merges new samplestat vector to original data frame
  nccl <- join(nccl, data, by = id, type = "full")

  # returns data frame
  nccl
}

## Read in data
real.data1 <- read.csv("cc_data_test.csv")

## real.data1$apache_q is the vector for APACHE quartiles

## Create covariate for Antibiotic Treatment within 48 hours. Yes = '1', No = '0'
real.data1$ATB48H <- ifelse(real.data1$ATB48H == 1, 1, 0)

## Delete patient with NA apache_ values.
real.data1 <- real.data1[-c(6090),]

```

```

## Delete patients with NA ATB48H values.
real.data1 <- real.data1[complete.cases(real.data1[, "ATB48H"]),]

## Add noise to "stop" times to break ties
set.seed(1)
real.data1$stop <- jitter(real.data1$stop,
                          .000001)

## Combine Death and Discharge into one competing event
real.data1$fail_2Events <- replace(real.data1[, "fail"], real.data1[, "fail"]== 3, 2)

## Creates samplestat vector
## Assign '2' for event 1 individuals
##(these will be given inverse probability weight = 1 in analysis)
temp <- replace(real.data1$fail_2Events, real.data1$fail_2Events== 2, 0)
temp <- replace(temp, temp == 1, 2)
real.data1$samplestat1 <- temp
rm(temp)

## FULL COHORT ANALYSIS

## NOSOCOMIAL INFECTION
real.data1_ncc_IDS_Full1 <- coxph(Surv(stop, fail_2Events == 1) ~ factor apache_q
                                + ATB48H, data = real.data1, ties = 'breslow')

## DEATH/DISCHARGE WITHOUT INFECTION
real.data1_ncc_IDS_Full2 <- coxph(Surv(stop, fail_2Events == 2) ~ factor apache_q
                                + ATB48H, data = real.data1, ties = 'breslow')

## NOSOCOMIAL INFECTION RISK RATIO

# Censor competing events
real.data1$NI <- ifelse(real.data1[, "fail_2Events"]==1, 1, 0)

# Log Binomial Model
real.data1_Sublite_NI_Full <- glm(NI ~ factor apache_q + ATB48H,
                                family = binomial(link = "log"), data = real.data1)

## SAMPLING

## Performs Incidence Density Sampling. 1 control per 1 event
set.seed(2)
ncc <- ccwc(entry = 0,
            exit = stop,
            fail = fail_2Events == 1,
            origin=0,
            controls = 1,
            data = real.data1,
            include = list(adm_id, stop, ATB48H, apache_q, samplestat1, fail_2Events))

# Adjusts samplestat vector. Sampled controls are given value '1'
real.data1 <- CreateSamplestat(ncc, real.data1, "samplestat1", "adm_id", "Set")

## TRADITIONAL ANALYSIS NCC DESIGN (CONDITIONAL LOGISTIC REGRESSION: NOSOCOMIAL INFECTION)
real.data1_ncc_IDS_Clog <- clogit(Fail ~ factor apache_q + ATB48H + strata(Set),
                                data = ncc)

## INVERSE PROBABILITY WEIGHTING (IPW) ANALYSIS: NOSOCOMIAL INFECTION

## Step 1
## Calculate Inclusion Probabilities

# Kaplan-Meier Weights

```



```
data= real.data1_subcohort,  
weights = (1/(KM_weights1) ), id= Set)  
  
# Logistic Regression Weights  
real.data1_Sublite_NI_Subcohort_GLM <- geeglm(NI ~ factor apache_q + ATB48H,  
family = binomial(link = "log"),  
data= real.data1_subcohort,  
weights = (1/(GLM_weights1) ), id= Set)
```