Supplementary material for paper van der Heijden, Whittaker, Cruyff, Bakker and van der Vliet.

Estimation in R

This document provides an example of the estimation procedure and the corresponding R-code to fit a log-linear model and obtain a confidence interval for the estimate of the population size N. The example involves the log-linear model [AX2][BX1] to the data presented in Table 5 in the paper. In this example where A and B are the two incomplete registers, with covariate X1 measured in A but not in B and covariate X2 measured in B but not in A. The data are supposed to be in an SPSS file called "data.sav", with the missing values on the covariates coded as system missing.

The estimation procedure:

Step 1: reading the data into R
Step 2: fitting the log-linear model to the data with the missing covariates
Step 3: obtain data with imputation of the missing values of the covariates
Step 4: fitting the log-linear to the imputed data to obtain an estimate of N
Step 5: calculation of the fit statistics
Step 6: parametric bootstrap to obtain the confidence interval for the estimate of N

R-code:

```
# loading the necessary libraries

library(cat)
library(foreign)



# step 1: reading the data into R

datafile            <- read.spss("data.sav",use.value.labels=FALSE,to.data.frame=TRUE)
colnames(datafile)  <- c("A","B","X1","X2")

a  <- 1
b  <- 2
x1 <- 3
x2 <- 4

# specification of the log-linear model [AX2][BX1],

MODEL       <- Freq~A*X2+B*X1
MARGINS     <- c(a,x2,0,b,x1)
MAXMODEL    <- Freq~A*X2+B*X1+X1*X2

# step 2: fitting the log-linear model to the data with the missing covariates

table.miss  <- as.data.frame(table(datafile))
observed    <- ifelse(table.miss[,1] == 2 & table.miss[,2] == 2, 0, 1)
table.miss  <- cbind(table.miss,observed)
dcat        <- prelim.cat(as.matrix(datafile))
```

```
struc.zero        <- array(observed, dim = dcat$d)/sum(observed)
dcat.saturated  <- em.cat(dcat,start=struc.zero,showits=FALSE)
dcat.model       <- ecm.cat(dcat,start=struc.zero,margins=MARGINS,showits=FALSE,eps= 1e-7)

# Step 3: obtain data with imputation of the  missing values covariates

table.complete          <- table.miss
table.complete$Freq   <- as.numeric(dcat.model*dcat$n)

# Step 4: fitting the log-linear to the imputed data to obtain an estimate of N

glm.complete            <- glm(MODEL,family=poisson,subset=(observed ==1), data = table.complete)
model.frame.complete <- model.frame(MODEL, data = table.complete)
fitted.complete         <- predict(glm.complete, model.frame.complete, type = "response")
hatN                        <- sum(fitted.complete)

# Step 5: calculation of the fit statistics

deviance.model          <- -2*(logpost.cat(dcat,dcat.model)-logpost.cat(dcat,dcat.saturated))
glm.maxmodel            <- glm(MAXMODEL, family = poisson, subset = (observed == 1),
                              data = table.complete)
df.adjusted             <- glm.complete$df.residual-glm.maxmodel$df.residual
aic                         <- deviance.model+2*(glm.complete$df.null-glm.complete$df.residual+1)

# printing the results

results                     <- cbind(deviance.model,df.adjusted,aic,hatN)
colnames(results)       <- c("Deviance","df","AIC","Pop.size")
rownames(results)       <- c(MODEL)
print(results)

# Step 6: parametric bootstrap to obtain the confidence interval for the estimate of N

nreps           <- 1000
boot.result     <- matrix(0,nreps,1)

for(i in 1:nreps){
        boot.data       <- table.complete[,1:(length(table.complete)-2)]
        Freq            <- rmultinom(n=1,size=round(hatN,0),
                           prob=fitted.complete/sum(fitted.complete))
        boot.data       <- cbind(boot.data,Freq)
        obs             <- ifelse(boot.data$A== 2 & boot.data$B == 2, 0, 1)
        boot.fit        <- glm(MODEL,family=poisson,subset=(obs==1),data=boot.data)
        boot.matrix     <- model.frame(MODEL, data = boot.data)
        boot.pred       <- predict(boot.fit, boot.matrix, type = "response")
        boot.result[i]  <- sum(boot.pred)
         }

CI_Nhat <- quantile(boot.result[,1], probs = c(0.025,0.975))
```