Codigo

Manuel Merino

5/31/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# load required packages
library(MASS)
library(boot)
# number of respondents
J <-1000
# number of items
I <-20
# number of skills
K <-4
# Q- matrix
Q <- t( matrix (c(1 ,0 ,0 ,0 ,1 ,0 ,0 ,0 ,1 ,1 ,0 ,0 ,1 ,1 ,0 ,0 ,1 ,1 ,0 ,0 ,1 ,1 ,0 ,0 ,
                        1 ,1 ,0 ,0 ,1 ,1 ,0 ,0 ,0 ,1 ,0 ,0 ,0 ,1 ,0 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1 ,1
                        1 ,1 ,1 ,1 ,1 ,1 ,1 ,0 ,0 ,1 ,0 ,1 ,1 ,1 ,1 ,0 ,1 ,0 ,1 ,1 ,0 ,0 ,1 ,1 ,0 ,0 ,0
rownames(Q) <- paste0 (" Item ", 1:I)
colnames(Q) <- paste0 ("A", 1:K)
# skill profile patterns
alpha_patt <- as.matrix ( expand.grid (c(0 ,1) ,c(0 ,1) ,c(0 ,1) ,c (0 ,1)))
colnames(alpha_patt) <- paste0 ("A", 1:4)
alpha_patt
# slip and guess
slip <- c(0.192 ,0.260 ,0.119 ,0.291 ,0.143 ,0.182 ,0.237 ,0.209 ,0.134 ,
                0.241 ,0.238 ,0.206 ,0.279 ,0.164 , 0.266 ,0.256 ,0.118 ,0.291 ,0.210 ,0.264)
guess <- c(0.201 ,0.242 ,0.263 ,0.122 ,0.230 ,0.186 ,0.119 ,0.117 ,0.174 ,
                 0.205 ,0.274 ,0.123 ,0.265 ,0.278 ,0.293 ,0.233 ,0.133 ,0.165 ,0.150 ,0.283)

# generate higher - order latent traits at two time points
set.seed (1234)
theta <- mvrnorm (n=J,mu=c(0 ,0.3) , Sigma = matrix (c(1 ,.8 ,.8 ,1) ,2 ,2))
# structural model parameters
lambda0 <- c(1.51 , -1.42 , -0.66 , 0.5)
# generate true skill mastery profiles and sample responses
resp <-array (NA , dim =c(J,I ,2))
A_all <-array (NA , dim=c(J,K ,2))
for (t in 1:2) {
    # find the prob of respondent j having skill k
```

```
      eta.jk <- matrix (0,J,K)

    for (j in 1:J) {
        for (k in 1:K){
            eta.jk[j, k]<-exp(theta[j,t] + lambda0[k])/(1 + exp(theta [j,t] + lambda0 [k]))}  }
            A <- matrix (0,J,K)
        for (j in 1:J) {
          for (k in 1:K) {
            A[j,k]=rbinom(1,1,eta.jk[j,k])
          }
        }

        # calculate if respondents have all skills needed for each item
        xi_ind <- matrix (0, J, I)
        for (j in 1:J) {
            for (i in 1:I) {
            xi_ind[j,i]<-prod(A[j, ]^Q[i, ])
            }
      }
       # generate prob correct and sample responses
       prob.correct <- matrix(0, J, I)
       y <- matrix (0, J, I)
       for (j in 1:J) {
          for (i in 1:I) {
              prob.correct [j,i] <- ((1 - slip [i])^xi_ind [j,i])*( guess [i ]^(1 - xi_ind[j,i]))
              y[j, i] <- rbinom (1, 1, prob.correct [j,i])}
      }

       A_all[,,t]<-A
       resp [,,t]<-y
}
        skill_data<-cbind(A_all [,,1],A_all [ , ,2])
        resp_data<-cbind( resp [,,1], resp [ , ,2])
```