https://doi.org/10.1027/1015-5759/a000631

ESM 3. R-codes

```
# installing R-packages and preparing data
# we assume that dataframe named „data“ is downloaded, with the first column named „country“ which identifies membership in a group (country). Further, we assume that all items are coded with the first value (category) set at 0.

install.packages("MultiLCIRT")
library(MultiLCIRT)

# data preparation
n = nrow(data)
dat=subset(data,select=c(2:23))
dat = as.matrix(dat)
clust = as.vector(data$country)
clust_unique = unique(clust)
for(i in 1:length(clust)) clust[i] =
+ which(clust_unique==clust[i])

# assigning the items to the corresponding latent trait:
dim = c(1,2,4,6,7,14,17,18,20,3,5,8,11,13,16,19,21,rep(0,1),9,10,12,15,22,rep(0,4))
dim = t(matrix(dim,9,3))

# fitting the standard LCA model with 1 latent class, deterministic input:
out1 = est_multi_poly(dat,k=1,start=0, link=0)
# fitting LCA model with increasing number of latent classes, deterministic input:
out2 = est_multi_poly(dat,k=2,start=0, link=0)
out3 = est_multi_poly(dat,k=3,start=0, link=0)
out4 = est_multi_poly(dat,k=4,start=0, link=0)
out5 = est_multi_poly(dat,k=5,start=0, link=0)
out6 = est_multi_poly(dat,k=6,start=0, link=0)
out7 = est_multi_poly(dat,k=7,start=0, link=0)
out8 = est_multi_poly(dat,k=8,start=0, link=0)

# displaying values of log-likelihood, the number of parameters, and BIC for models with the deterministic input (see Table 1 for the output):
rbind(cbind(out1$lk, out1$np, out1$bic), cbind(out2$lk, out2$np, out2$bic), cbind(out3$lk, out3$np, out3$bic), cbind(out4$lk, out4$np, out4$bic), cbind(out5$lk, out5$np, out5$bic), cbind(out6$lk, out6$np, out6$bic), cbind(out7$lk, out7$np, out7$bic), cbind(out8$lk, out8$np, out8$bic))

# fitting the standard LCA model with 1 latent class, random input:
out11 = est_multi_poly(dat,k=1,start=1, link=0)
# fitting the standard LCA model with increasing number of latent classes, random input:
out22 = est_multi_poly(dat,k=2,start=1, link=0)
out33 = est_multi_poly(dat,k=3,start=1, link=0)
out44 = est_multi_poly(dat,k=4,start=1, link=0)
out55 = est_multi_poly(dat,k=5,start=1, link=0)
out66 = est_multi_poly(dat,k=6,start=1, link=0)
out77 = est_multi_poly(dat,k=7,start=1, link=0)
```

```
out88 = est_multi_poly(dat,k=8,start=1, link=0)

# displaying values of log-likelihood, the number of parameters, and BIC for models with the random input (see Table 1 for the output):
rbind(cbind(out11$lk, out11$np, out11$bic), cbind(out22$lk, out22$np, out22$bic),
cbind(out33$lk, out33$np, out33$bic), cbind(out44$lk, out44$np, out44$bic),
cbind(out55$lk, out55$np, out55$bic), cbind(out66$lk, out66$np, out66$bic),
cbind(out77$lk, out77$np, out77$bic), cbind(out88$lk, out88$np, out88$bic))

# fitting the three-dimensional graded-response LCA models with 7 latent classes, global logits link, different IRT parameterizations:
# GRM
outA=est_multi_poly(dat,k=7,multi=dim,start=1,link=1,disc=1,difl=0)
# RS-GRM
outB=est_multi_poly(dat,k=7,multi=dim,start=1,link=1, disc=1,difl=1)
# 1P-GRM
outC=est_multi_poly(dat,k=7,multi=dim,start=1,link=1,disc=0,difl=0)
# 1P-RS-GRM
outD=est_multi_poly(dat,k=7,multi=dim,start=1,link=1,disc=0, difl=1)

# displaying values of log-likelihood, the number of parameters, and BIC for models with different IRT parameterizations (see Table 2 for the output):
rbind(cbind(outA$lk, outA$np, outA$bic), cbind(outB$lk, outB$np, outB$bic),
cbind(outC$lk, outC$np, outC$bic), cbind(outD$lk, outD$np, outD$bic))

# fitting the three-dimensional multilevel graded-response LCA models with 7 latent classes
out1=est_multi_poly_clust(dat,kU=1,kV=7,multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)
out2=est_multi_poly_clust(dat,kU=2,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)
out3=est_multi_poly_clust(dat,kU=3,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)
out4=est_multi_poly_clust(dat,kU=4,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)
out5=est_multi_poly_clust(dat,kU=5,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)
out6=est_multi_poly_clust(dat,kU=6,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust)

# displaying values of log-likelihood, the number of parameters, and BIC (see Table 3 for the output):
rbind(cbind(out1$lk, out1$np, out1$bic), cbind(out2$lk, out2$np, out2$bic),cbind(out3$lk,
out3$np, out3$bic), cbind(out4$lk, out4$np, out4$bic), cbind(out5$lk, out5$np, out5$bic),
cbind(out6$lk, out6$np, out6$bic))

# re-fit the final model to obtain additional output
out5=est_multi_poly_clust(dat,kU=5,kV=7, multi=dim,start=0,link=1,
disc=1,difl=0,clust=clust,output=TRUE)

# Abilities at respondents' level for each latent class and each dimension
```

```
ind = order(out5$Th[1,])
Ths = out5$Th[,ind]
Ths

# Latent class average weights at respondents' level
Pivs = out5$Piv[,ind,]
Pivms = matrix(0,n,7)
for(h in unique(clust)){
+ ind1 = which(clust==h)
+ for(u in 1:5) Pivms[ind1,] =
+ Pivms[ind1,]+Pivs[ind1,,u]*out5$La[h,u]
+ }
pivs = colMeans(Pivms)
pivs

# Standardized respondents' abilities (see Table 4 for the output)
mThs = as.vector(Ths%*%pivs)
vThs = as.vector((Ths-mThs)^2%*%pivs)
sThs = (Ths-mThs)*(1/sqrt(vThs))
sThs

# Standardized difficulties for each item (see Table E1d for the output)
outST = standard.matrix(t(out5$Th),pivs)
inad = as.vector(t(out5$Bec[c(1,2,4,6,7,14,17,18,20),]))
reas = as.vector(t(out5$Bec[c(3,5,8,11,13,16,19,21),]))
hate = as.vector(t(out5$Bec[c(9,10,12,15,22),]))
inadSTdif = (inad-outST$mu[1])/outST$si[1]
reasSTdif = (reas-outST$mu[2])/outST$si[2]
hateSTdif = (hate-outST$mu[3])/outST$si[3]

# Standardized discriminations for each item (see Table E1d for the output)
inadSTdisc = out5$gac[c(1,2,4,6,7,14,17,18,20)]*outST$si[1]
reasSTdisc = out5$gac[c(3,5,8,11,13,16,19,21)]*outST$si[2]
hateSTdisc = out5$gac[c(9,10,12,15,22)]*outST$si[3]

# Mean values of respondents' abilities for each type of countries (see Table 5 for the output)
ThU = rep(0,times=5)
for(u in 1:5) ThU[u] = sum(Pivs[,,u]%*%t(sThs))/(n*7)
ThU
```