**Section1 The code for constructing the Item Pool**

library("mirt")

#Loading the mirt package.

library("mokken")

# Loading the mokken package.

library("lordif")

library("catR")

**#model fit**

data<-as.matrix (read.csv("F:/Study/IRT /SHQ/ Allitems.csv",header=T))

# Importing the data, the data only contains the items that need to be analyzed. If there are demographic variables in the data, delete them first.

modelfit\_graded<- mirt(data, model=1,itemtype="graded")

# Calculation model fitting index.

# "data" is the imported data that needs to be analyzed.

# "model" represents the model. If the analyzed data is unidimensionality, model=1 is used. If the analyzed data is multidimensional, model can be defined as the required model.

# 3" itemtype " represents the model type used to fit the data.

**# Local Independence**

data<-as.matrix (read.csv("F:/Study/IRT /SHQ/items\_MF.csv",header=T))

# The remaining data after deleting the items with *Q3* value was larger than 0.36

modelfit\_graded<- mirt(data, model=1,itemtype="graded")

Q3\_value<-residuals (modelfit\_graded, type = 'Q3')

write.csv(Q3\_value," "F:/Study/IRT/SHQ/items\_MLI.csv ")

#**Estimation of Item Discrimination Parameter**

data<-as.matrix (read.csv("F:/Study/IRT /SHQ/ items\_MLI.csv ",header=T))

# The remaining data with the items met IRT assumption of local independence.

modelfit\_graded<- mirt(data, model=1,itemtype="graded")

itempaem\_a<- coef(modelfit\_graded)

# Calculating the item discrimination parameter.

write.csv(itempaem\_a, "F:/Study/IRT /SHQ/items\_D.csv ")

**#item fit**

data<-as.matrix (read.csv("F:/Study/IRT /SHQ/items\_D.csv ",header=T))

# The remaining data with the item discrimination parameter was not bad.

modelfit\_graded<- mirt(data, model=1,itemtype="graded")

SX2\_value<-itemfit (modelfit\_graded, fit\_stats = "S\_X2")

write.csv(SX2\_value," F:/Study/IRT/ SHQ/items\_IF.csv ")

# **Differential Item Functioning**

data<-as.matrix (read.csv("F:/Study/IRT /SHQ/items\_IF.csv ",header=T))

# The remaining data with the items fitted the data well.

modelfit\_graded<- mirt(data, model=1,itemtype="graded")

gender<-as.matrix(read.csv("F:/Study/IRT /SHQ/gender.csv",header=T))

DIF<-lordif(data,gender,model="grm",pseudo.R2="McFadden",criterion="R2",R2.change=0.02)

region1<-as.matrix(read.csv("F:/Study/IRT/SHQ/region1.csv",header=T)) DIF<-lordif(data,region1,model="grm",pseudo.R2="McFadden",criterion="R2",R2.change=0.02)

region2<-as.matrix(read.csv("F:/Study/IRT/SHQ/region2.csv",header=T))

DIF<-lordif(data,region2,model="grm",pseudo.R2="McFadden",criterion="R2",R2.change=0.02)

**Section 2 The code for the Simulation of the CAT-SWB**

**(Simulated data)**

library("catR")

theta\_true<-as.matrix(rep(seq(-3.5,3.5,0.1),100))

# Simulating the true theta value.

m.GRM<-as.matrix(read.csv("F:/Study /CAT/SHQ /item\_parameter.csv",header=T))

# Import item parameters

theta\_end<-matrix(NA,nrow(theta\_true),1)

# Matrix for storing the final ability values of the subjects.

MI<-matrix(NA,nrow(theta\_true),1)

# Used to store the total amount of information that the subject answered.

response\_item<-matrix(NA,nrow(theta\_true),nrow(m.GRM))

# Used to store the items that have been answered by the participants.

for(i in 1:nrow(theta\_true)){

 x<-vector(mode="numeric",length=0)

# Generate an empty vector to store the items that the participant has already answered.

U<-vector(mode="numeric",length=0)

# Generate an empty vector to store the score of the item that the subject has already answered.

rownames(m.GRM)<-c(1:nrow(m.GRM))

row\_name<-as.numeric(rownames(m.GRM))

# Extracting the row name of the matrix.

sample\_row\_name<-sample(row\_name,1)

# Randomly selecting a row from the matrix and randomly selecting an item from the item bank as the initial item.

 x[1]<-sample\_row\_name

# The first item answered by the participant.

response\_item[i,1]<-x[1]

# Putting the first item into the response matrix of the subject.

 U[1]<-genPattern(theta\_true[i,], m.GRM[x[1],],model="GRM")

# Simulating the response according to the true value of the participant’s ability.

theta\_est<-eapEst(m.GRM[x[1],],U[1],model="GRM",nqp=41)

# Estimating the current ability value of the subject according to the item parameters of the selected initial item and the simulation item response.

 for(j in 1:nrow(m.GRM)){

next\_item<-nextItem(m.GRM, model = "GRM", theta = theta\_est, criterion = "MFI", out = x)

# Selecting the next item based on the current estimated ability of the subject.

x[j+1]<-next\_item$item

# Putting the newly selected item into the completed item as a new scalar.

 response\_item[i,j+1]<-x[j+1]

# Putting the selected items that will be answered into the established empty matrix.

U[j+1]<-genPattern(theta\_true[i,], m.GRM[x[j+1],],model="GRM")

# According to the true value of the subject’s ability, simulate the response to the newly selected item which the subject answered. theta\_est\_new<-eapEst(m.GRM[x,],U,model="GRM",nqp=41)

# Estimating their ability based on all the items answered by the participants.

 information<-Ii(theta\_est\_new,m.GRM[x,],model="GRM")$Ii

Minformation<-sum(information)

# Calculating the total amount of information for items that has been answered.

 if(Minformation>=4|length(x)==nrow(m.GRM)){

 theta\_end[i,]<-print(theta\_est\_new)

# Saving the estimated ability value of each subject.

 MI[i,]<-print(Minformation)

 print(x)

# Saving the items answered by each participant.

break

 } else {

 theta\_est<-theta\_est\_new

 }

 }

}

write.csv(theta\_end,file= " F:/Study /CAT/SHQ /theta\_end\_sim4.csv")

write.csv(response\_item, " F:/Study/CAT/SHQ/response\_item\_sim4.csv")

write.csv(theta\_true, " F:/Study /CAT/SHQ /theta\_true\_sim4.csv")

write.csv(MI, " F:/Study /CAT/SHQ /MI\_sim4.csv")