**Supplement 2. R code for the network analysis**

R code

#Title" Network analysis for online learning readiness of Vietnamese medical students during COVID-19 pandemic "

#Author" Dinh Duong Le"

#Date"June 10, 2022"

library(readxl)

data <- read\_excel("C:/Users/Admin/ ")

##Required libraries

library("readxl")

library(qgraph)

library(ggplot2)

library(JGR)

library(rJava)

library(huge)

library("psychTools")

library(reshape2)

library(NetworkComparisonTest)

library(data.table)

#library("Deducer")

library(bootnet)

library(data.table)

library(tidyverse)

library(stringi)

library("NetworkToolbox")

library(lattice)

library(survival)

library(Formula)

library(Hmisc)

library(igraph)

# Set the random seed:

set.seed(123)

#data

male<-subset(data, sex=="1")

female<-subset(data, sex=="2")

lowgrade<-subset (data, year\_code=="1")

highgrade<-subset(data, year\_code=="2")

#Reset names

setnames (data, old = c("computerskill", "internet", "communication", "motivation","selfcontrol", "selflearning"), new = c("CS","IA","OC","M", "SC","SL"))

setnames (male, old = c("computerskill", "internet", "communication", "motivation","selfcontrol", "selflearning"), new = c("CS","IA","OC","M", "SC","SL"))

setnames (female, old = c("computerskill", "internet", "communication", "motivation","selfcontrol", "selflearning"), new = c("CS","IA","OC","M", "SC","SL"))

setnames (lowgrade, old = c("computerskill", "internet", "communication", "motivation","selfcontrol", "selflearning"), new = c("CS","IA","OC","M", "SC","SL"))

setnames (highgrade, old = c("computerskill", "internet", "communication", "motivation","selfcontrol", "selflearning"), new = c("CS","IA","OC","M", "SC","SL"))

#define catogorical variable.names

data$sex=as.factor(data$sex)

data$year=as.factor(data$year)

data$year\_code=as.factor(data$year\_code)

data$major=as.factor(data$major)

data$resident=as.factor(data$resident)

data$livingcost=as.factor(data$livingcost)

#install.packages(table1)

#install.packages(compareGroups)

library(table1)

library(compareGroups)

#Tabel 1. General characteristics and readiness score

table1(sex~sex+age+year+major+resident+livingcost+CS+IA+OC+M+SC+SL+total, data=data)

t1=compareGroups(sex~sex+age+year+major+resident+livingcost+CS+IA+OC+M+SC+SL+total, data=data)

t1

createTable(t1)

#grade level

table1(year\_code~sex+age+year+major+resident+livingcost+CS+IA+OC+M+SC+SL+total, data=data)

t2=compareGroups(year\_code~sex+age+year+major+resident+livingcost+CS+IA+OC+M+SC+SL+total, data=data)

t2

createTable(t2)

#nomility test

shapiro.test(data$IA)

shapiro.test(data$IA)

shapiro.test(data$CS)

shapiro.test(data$M)

shapiro.test(data$SL)

#compare sex and OLRS

CS<-wilcox.test(CS~sex,data=data)

CS

IA<-wilcox.test(IA~sex,data=data)

IA

OC<-wilcox.test(OC~sex,data=data)

OC

M<-wilcox.test(M~sex,data=data)

M

SC<-wilcox.test(SC~sex,data=data)

SC

SL<-wilcox.test(SL~sex,data=data)

SL

total<-wilcox.test(total~sex,data=data)

total

## compare grade level and OLRS

CS<-wilcox.test(CS~year\_code,data=data)

CS

IA<-wilcox.test(IA~year\_code,data=data)

IA

OC<-wilcox.test(OC~year\_code,data=data)

OC

M<-wilcox.test(M~year\_code,data=data)

M

SC<-wilcox.test(SC~year\_code,data=data)

SC

SL<-wilcox.test(SL~year\_code,data=data)

SL

total<-wilcox.test(total~year\_code,data=data)

total

#Network structure correlation

#create data

data1 <- data[, c('CS','IA', 'OC', 'M', 'SC', 'SL' )]

male1<-male[,c('CS','IA', 'OC', 'M', 'SC', 'SL' )]

female1<-female[,c('CS','IA', 'OC', 'M', 'SC', 'SL' )]

lowgrade1<-lowgrade[,c('CS','IA', 'OC', 'M', 'SC', 'SL' )]

highgrade1<-highgrade[,c('CS','IA', 'OC', 'M', 'SC', 'SL' )]

#transforme data

tdata1<-huge.npn(data1)

tmale1 <- huge.npn(male1)

tfemale1 <- huge.npn(female1)

tlowgrade1<-huge.npn(lowgrade1)

thighgrade1<-huge.npn(highgrade1)

#correlation

cortdata1<-cor\_auto(tdata1)

cortmale1<-cor\_auto(tmale1)

cortfemale1<-cor\_auto(tfemale1)

cortlowgrade1<-cor\_auto(tlowgrade1)

corthighegrade1<-cor\_auto(thighegrade1)

#graph

graph1 <- qgraph(EBICglasso(cortdata1, nrow(tdata1), lambda.min.ratio=0.01, gamma=0.5, threshold = TRUE),color='#FFCC66', label.scale=FALSE,edge.labels=TRUE,

 edge.color='#993300', title="a")

graph2 <- qgraph(EBICglasso(cortmale1, nrow(tmale1),

 lambda.min.ratio=0.01, gamma=0.5, threshold = TRUE), color='#FFCC66', label.scale=FALSE,edge.labels=TRUE,

 edge.color='#993300')

graph3 <- qgraph(EBICglasso(cortfemale1, nrow(tfemale1),

 lambda.min.ratio=0.01, gamma=0.5, threshold = TRUE), color='#FFCC66', label.scale=FALSE,edge.labels=TRUE,

 edge.color='#993300')

graph4 <- qgraph(EBICglasso(cortlowgrade1, nrow(tlowgrade1),

 lambda.min.ratio=0.01, gamma=0.5, threshold = TRUE), color='#FFCC66', label.scale=FALSE,edge.labels=TRUE,

 edge.color='#993300')

graph5 <- qgraph(EBICglasso(corthighegrade1, nrow(thighegrade1),

 lambda.min.ratio=0.01, gamma=0.5, threshold = TRUE), color='#FFCC66', label.scale=FALSE,edge.labels=TRUE,

 edge.color='#993300')

#estimate network

net1 <- estimateNetwork(tdata1,default = "EBICglasso" , corMethod = "cor\_auto",tuning = 0.5)

net2 <- estimateNetwork(tmale1,default = "EBICglasso",corMethod = "cor\_auto",tuning = 0.5)

net3 <- estimateNetwork(tfemale1,default = "EBICglasso",corMethod = "cor\_auto",tuning = 0.5)

net4 <-estimateNetwork(tlowgrade1, default="EBICglasso", corMethod="cor\_auto", tuning=0.5)

net5 <-estimateNetwork(thighgrade1, default="EBICglasso", corMethod="cor\_auto", tuning=0.5)

#changing the hyperparameter tuning

#setting tungning = 0.0

netmale0.0 <- estimateNetwork(tmale1,default = "EBICglasso",corMethod = "cor\_auto",tuning = 0.0)

netmale0.25 <- estimateNetwork(tmale1,default = "EBICglasso",corMethod = "cor\_auto",tuning = 0.25)

netmale0.5 <- estimateNetwork(tmale1,default = "EBICglasso",corMethod = "cor\_auto",tuning = 0. 5)

#get matrices

net1$graph

net2$graph

net3$graph

net4$graph

net5$graph

#Plot

L <- averageLayout(net2, net3, net3, net4)

n1=plot(net1, title="a)", cut = 0.03,negDashed=TRUE,layout = L)

n2=plot(net2, title="b)", cut = 0.03,negDashed=TRUE,layout = L)

n3=plot(net3, title="c)", cut = 0.03,negDashed=TRUE,layout = L)

n4=plot(net4, title="d)", cut = 0.03,negDashed=TRUE,layout = L)

n5=plot(net5, title="e)", cut = 0.03,negDashed=TRUE,layout = L)

#attach graph

attach(networkplot)

par(mfrow=c(1,1))

plot(graph1)

plot(graph2)

plot(graph3)

plot(graph4)

plot(graph5)

#centrality

centralityPlot(net1, include = c("Strength", "Closeness","Betweenness"), theme\_bw = TRUE, print = TRUE,

 verbose = TRUE, weighted = TRUE, signed = TRUE,

 orderBy = "default", decreasing = FALSE)

centralityPlot(net2, include = c("Strength", "Closeness","Betweenness"), theme\_bw = TRUE, print = TRUE,

 verbose = TRUE, weighted = TRUE, signed = TRUE,

 orderBy = "default", decreasing = FALSE)

centralityPlot(net3, include = c("Strength", "Closeness","Betweenness"), theme\_bw = TRUE, print = TRUE,

 verbose = TRUE, weighted = TRUE, signed = TRUE,

 orderBy = "default", decreasing = FALSE)

centralityPlot(net4, include = c("Strength", "Closeness","Betweenness"), theme\_bw = TRUE, print = TRUE,

 verbose = TRUE, weighted = TRUE, signed = TRUE,

 orderBy = "default", decreasing = FALSE)

centralityPlot(net5, include = c("Strength", "Closeness","Betweenness"), theme\_bw = TRUE, print = TRUE,

 verbose = TRUE, weighted = TRUE, signed = TRUE,

 orderBy = "default", decreasing = FALSE)

Cent1 <- centralityTable(net1, standardized = TRUE)

View(Cent1)

table.data1 <- data.frame("Item" = Cent1[Cent1$measure == "Strength",3],

 "Strength" = Cent1[Cent1$measure == "Strength",5])

View(table.data1)

Cent2 <- centralityTable(net2, standardized = TRUE)

table.data2 <- data.frame("Item" = Cent2[Cent2$measure == "Strength",3],

 "Strength" = Cent2[Cent2$measure == "Strength",5])

Cent3 <- centralityTable(net3, standardized = TRUE)

table.data3 <- data.frame("Item" = Cent3[Cent3$measure == "Strength",3],

 "Strength" = Cent3[Cent3$measure == "Strength",5])

centralityPlot(Centrality = list("total" = Cent1,"male" = Cent2, "female"=Cent3),

 include=c("Strength","Closeness", "Betweenness"),

 labels = names(student),decreasing=T)+ theme(legend.title = element\_blank())

df=cbind(table.data1 ,table.data2 ,table.data3)

df <- df[ -c(3,5,7) ]

write.csv(df,file="cent1.csv")

#Stability

b1 <- bootnet(net1, nCores = 8,nBoots = 1000, type = 'nonparametric')

b2 <- bootnet(net1, nCores = 8,nBoots = 1000, type = 'case',

 statistics=c('strength','closeness','betweenness'))

b3 <- bootnet(net2, nCores = 8,nBoots = 1000, type = 'nonparametric')

b4 <- bootnet(net2, nCores = 8,nBoots = 1000, type = 'case',

 statistics=c('strength','closeness','betweenness'))

b5 <- bootnet(net3, nCores = 8,nBoots = 1000, type = 'nonparametric')

b6 <- bootnet(net3, nCores = 8,nBoots = 1000, type = 'case',

 statistics=c('strength','closeness','betweenness'))

b7 <- bootnet(net4, nCores = 8,nBoots = 1000, type = 'nonparametric')

b8 <- bootnet(net4, nCores = 8,nBoots = 1000, type = 'case',

 statistics=c('strength','closeness','betweenness'))

b9 <- bootnet(net5, nCores = 8,nBoots = 1000, type = 'nonparametric')

b10 <- bootnet(net5, nCores = 8,nBoots = 1000, type = 'case',

 statistics=c('strength','closeness','betweenness'))

#total

plot(b1, order="sample", plot="area", prop0=F)# confidence intervals

plot(b1, "edge", plot = "difference", onlyNonZero = TRUE, order = "sample") # difference of edges

plot(b1, "strength", plot = "difference") # node strength

plot(b2 ,statistics=c('strength','closeness','betweenness'))+ theme(legend.title = element\_blank())

#male

plot(b3, order="sample", plot="area", prop0=F)# confidence intervals

plot(b3, "edge", plot = "difference", onlyNonZero = TRUE, order = "sample") # difference of edges

plot(b3, "strength", plot = "difference") # node strength

plot(b4 ,statistics=c('strength','closeness','betweenness'))+ theme(legend.title = element\_blank())

#female

plot(b5, order="sample", plot="area", prop0=F)# confidence intervals

plot(b5, "edge", plot = "difference", onlyNonZero = TRUE, order = "sample") # difference of edges

plot(b5, "strength", plot = "difference") # node strength

plot(b6 ,statistics=c('strength','closeness','betweenness'))+ theme(legend.title = element\_blank())

#lowgrade

plot(b7, order="sample", plot="area", prop0=F)# confidence intervals

plot(b7, "edge", plot = "difference", onlyNonZero = TRUE, order = "sample") # difference of edges

plot(b7, "strength", plot = "difference") # node strength

plot(b8 ,statistics=c('strength','closeness','betweenness'))+ theme(legend.title = element\_blank())

#highgrade

plot(b9, order="sample", plot="area", prop0=F)# confidence intervals

plot(b9, "edge", plot = "difference", onlyNonZero = TRUE, order = "sample") # difference of edges

plot(b9, "strength", plot = "difference") # node strength

plot(b10 ,statistics=c('strength','closeness','betweenness'))+ theme(legend.title = element\_blank())

#corstability

corStability(b2)

corStability(b4)

corStability(b6)

corStability(b8)

corStability(b10)

##Network comparsion test

nct1<- NCT(net2, net3, it=1000,paired = FALSE,test.edges = TRUE,edges="all",

 test.centrality=TRUE,centrality=c("strength","closeness","betweenness"))

nct2 <-NCT (net4,net5,it=1000,paired = FALSE,test.edges = TRUE,edges="all", test.centrality=TRUE,centrality=c("strength","closeness","betweenness"))

nct1

nct2

## Perform permutation test

#compare male and female

perm.str <- network.permutation(tmale1, tfemale1, iter = 1000, network = "glasso", measure = "strength", alternative = "two.tailed", ncores = 4)

#Average shortest path length

ASP <- network.permutation(prev.perm = perm.str, measure = "ASPL", ncores = 4)

ASP$result

#Average clustering coefficient

AVC <- network.permutation(prev.perm = perm.str, measure = "CC", ncores = 4)

# Check results

AVC$result

# Modularity quality index

MQI <- network.permutation(prev.perm = perm.str, measure = "Q", ncores = 4)

MQI$result

#smallworldness

SS <- network.permutation(prev.perm = perm.str, measure = "S", ncores = 4)

# Check results

perm.aspl$result

#density

library(igraph)

g=as.igraph(n2, attributes=TRUE)

plot(n2)

a<-(degree(g))/(vcount(g)-1)

g <- as.igraph(n3, attributes=TRUE)

b<-(degree(g))/(vcount(g)-1)

t.test(a,b,statistic=c("t","mean"),alternative=c("two.sided"), midp=TRUE, B=1000)

#density

g1=as.igraph(n4, attributes=TRUE)

plot(n4)

a1<-(degree(g1))/(vcount(g1)-1)

g2 <- as.igraph(n5, attributes=TRUE)

b1<-(degree(g2))/(vcount(g2)-1)

t.test(a1,b1,statistic=c("t","mean"),alternative=c("two.sided"), midp=TRUE, B=1000)

##Represent Node sizes represent the difference of centrality

#strength

Cent2 <- centralityTable(net2, standardized = TRUE)

Cent3 <- centralityTable(net3, standardized = TRUE)

Cent4 <- centralityTable(net4, standardized = TRUE)

Cent5 <- centralityTable(net5, standardized = TRUE)

table.s1 <- data.frame("Item" = Cent2[Cent2$measure == "Strength",3],

 "Strength" = Cent2[Cent2$measure == "Strength",5])

table.s2 <- data.frame("Item" = Cent3[Cent3$measure == "Strength",3],

 "Strength" = Cent3[Cent3$measure == "Strength",5])

table.s3 <- data.frame("Item" = Cent4[Cent4$measure == "Strength",3],

 "Strength" = Cent4[Cent4$measure == "Strength",5])

table.s4 <- data.frame("Item" = Cent5[Cent5$measure == "Strength",3],

 "Strength" = Cent5[Cent5$measure == "Strength",5])

s1 <- table.s1[,2]

s2 <- table.s2[,2]

s12=s2-s1

s3 <- table.s3[,2]

s4 <- table.s4[,2]

s34=s4-s3

tablesa=cbind(s3 ,s4)

a <- tablesa[c(3,5,7) ]

write.csv(atablesa,file="gg.csv")

# Plot centrality index

Cent2 <- centralityTable(net2, standardized = TRUE)

Cent3 <- centralityTable(net3, standardized = TRUE)

#male and female

plot (s1,type = "o",col = "black", lwd=3, xlab="nodes", ylab = "Strength" ,ylim = c(min(s1), max(s2)), main = "Centrality index")

lines(s2, type = "o",lwd=3, col = "brown")

legend(1,2,c("male","female"), lwd=c(3,3), col=c("blue","red"), y.intersp=1.5)

# lowgrade and highgrade

plot (s3,type = "o",col = "black", lwd=4, xlab="nodes", ylab = "Strength" ,ylim = c(min(s3), max(s4)))

lines(s4, type = "o",lwd=4, col = "red")