```
#BRIGANTI GIOVANNI 2019 TAS Network Analysis
library(stats)
library(qgraph)
library(readr)
library(bootnet)
library(mgm)
library(igraph)
library(glasso)
library(lavaan)
library(dplyr)
require(pcalg)
require(ggm)
require(corpcor)
require(Rgraphviz)
require(RBGL)
library(reshape2)
library(data.table)
library(psych)
library (ggplot2)
library(Hmisc)
library(Matrix)


#reading TAS database
data <-
as.data.frame(read_delim("/Users/giovannibriganti/Desktop/Network_Analysis_Mac/
TAS/TAS_data.csv",
                                    ";", escape_double = FALSE, trim_ws = TRUE))

options(max.print = .Machine$integer.max)

#creating groups corresponding to the three clusters of TAS
gr <- list (c(1, 3, 6, 7, 9, 13, 14), c(2, 4, 11, 12, 17),
            c(5, 8, 10, 15, 16, 18, 19, 20 ))

#correlations
datacor <- cor(data)
datacor #visualize correlation matrix


#factor modeling and subscores (lavaan) for the original dataset
cmodel <- ' Identify =~ TAS1 + TAS3 + TAS6 + TAS7 + TAS9 + TAS13 +   TAS14
Describe =~ TAS2 + TAS4 + TAS11 + TAS12 + TAS17
ThinkingExt =~ TAS5 + TAS8 + TAS10 + TAS15 + TAS16 +   TAS18 +   TAS19 +
TAS20'

#confirmatory factor analysis for the 3-factor model > fit the model
fit <- cfa(cmodel, data=data)

#get the factorscore dataset
```

```r
lavPredict(fit)
factorscoredata <- lavPredict(fit)
factorscoredata <- as.data.frame(factorscoredata)
write_excel_csv(factorscoredata, "factorscoredata.csv")

#estimating network
network1 <- estimateNetwork(data, default="EBICglasso", corMethod = "cor",
                                    corArgs =list(use = "pairwise.complete.obs"),
                                    threshold=TRUE, lambda.min.ratio=0.001)

#node predictability
type=rep('g', 20) #g=gaussian, 20 = number of nodes in the network
fit1<-mgm(data,
            type=type,
            level=rep(1,20))

pred1<- predict(fit1, data)
pred1$error$R2
mean(pred1$error$R2)

#get descriptive statistics for dataset
describe(data)
means <- colMeans(data, na.rm=T)
plot(means)
sds <- as.vector(sapply(data, sd, na.rm=T))
plot(sds)

#glasso item network - Figure 1
graph1 <- plot(network1, pie = pred1$error$R2, groups=gr,
                  layout="spring", legend.cex=.45,vsize=7,
                  border.width=2, border.color='#555555',
                  minimum = 0.05,
                  color=c("#66c2a5", "#fc8d62", "#8da0cb"))
#to see maximum and minimum, set details=TRUE
pdf("Figure1.pdf", width = 10, height=6)
plot(graph1)
dev.off()

#graph1 weight matrix
graph1mat <- getWmat(graph1)
graph1mat #visualize weight matrix
write.csv(graph1mat, "itemnetworkweightmatrix.csv")

#mean graph1 weight matrix
mean(graph1mat) #visualize the mean edge weight of the network = 0.026

#walktrap algorithm
#1/build a graph object for the algorithm
glasso.ebic <-EBICglasso(S=datacor, n = nrow(data), threshold=TRUE)
graph.glasso <-as.igraph(qgraph(glasso.ebic, layout = "spring", vsize = 3))
```

```
#2/show the membership of an item to a community
wc<- walktrap.community(graph.glasso)
wc$membership
#3/number of communities
n.dim <- max(wc$membership)
#the walktrap algorithm detects 4 communities -> visualize the network
gr2 <- list(c(1, 3, 6, 7, 9, 13, 14), c(15, 16, 20),
            c(5, 8, 10, 18, 19), c(2, 4, 11, 12, 17))
pdf("Figure2.pdf", width=10, height=6)
graph1a <- plot(network1, pie = pred1$error$R2, groups=gr2,
                layout="spring", legend.cex=.45,vsize=7,
                border.width=2, border.color='#555555',
                minimum = 0.05,
                color=c("#66c2a5", "#d7191c", "#8da0cb","#fc8d62"))
dev.off()

#stability for the item network
boot1 <- bootnet(network1, ncores=7, nboots=2000)
boot2 <- bootnet(network1, ncores=7, nboots=2000, type="case")
save(boot1, file = "boot1.Rdata")
save(boot2, file = "boot2.Rdata")

#Figure1a Edge weight bootstrap for the item network
plot(boot1, labels = FALSE, order = "sample")
fig2c <- plot(boot1, labels = FALSE, order = "sample")
pdf("Figure1a.pdf", width=10, height=7)
plot(boot1, labels = TRUE, order = "sample")
dev.off()

#Figure1b - Item network, Edge weight difference: is edge X significantly larger than
edge Y? Black=Y Gray=N
boot3 <- plot(boot1, "edge", plot = "difference", onlyNonZero = TRUE, order =
"sample")
plot(boot3)
pdf("Figure1b.pdf", width=10, height=7)
plot(boot3)
dev.off()

###################### Part 2 : 3 domain network
##########################
#Estimate a 3-domain network
data2 <- as.data.frame(read_delim("/Users/giovannibriganti/Google
Drive/Network_Analysis_Mac/Briganti2019_TAS/factorscoredata.csv",
                          ",", escape_double = FALSE, trim_ws = TRUE))

factornames <- c("ID", "DES", "Think")

longnames <- c("Difficulty identifying feelings", "Difficulty describing feelings",
               "Externally-oriented thinking")
```

```
#get descriptive statistics for dataset
describe(data2)
means2 <- colMeans(data2, na.rm=T)
pdf("meansplot2.pdf", width=6, height=4)
plot(means2)
dev.off()
sds <- as.vector(sapply(data2, sd, na.rm=T))
pdf("sdplot2.pdf", width=6, height=4)
plot(sds2)
dev.off()

#node predictability
type=rep('g', 3) #g=gaussian, 3 = number of nodes in the network
fit2<-mgm(data2,
          type=type,
          level=rep(1,3))

pred2<- predict(fit2, data2)
pred2$error$R2
mean(pred2$error$R2)

#estimating network
network2 <- estimateNetwork(data2, default="EBICglasso", corMethod = "cor",
                            corArgs =list(use = "pairwise.complete.obs"),
                            threshold=TRUE, lambda.min.ratio=0.001)

#glasso item network - Figure 3
graph2 <- plot(network2, pie = pred2$error$R2,
               layout="spring", legend.cex=.45,vsize=7,
               border.width=2, border.color='#555555',
               labels=factornames, nodeNames=longnames,
               color=c("#66c2a5", "#fc8d62", "#8da0cb"))
#to see maximum and minimum, set details=TRUE
pdf("Figure3.pdf", width = 10, height=6)
plot(graph2)
dev.off()

#graph2 weight matrix
graph2mat <- getWmat(graph2)
graph2mat #visualize weight matrix
write.csv(graph2mat, "3domainnetworkweightmatrix.csv")

#mean graph1 weight matrix
mean(graph2mat) #visualize the mean edge weight of the network = 0.23

#stability for the 3 domain network
boot12 <- bootnet(network2, ncores=7, nboots=2000)
boot22 <- bootnet(network2, ncores=7, nboots=2000, type="case")
save(boot12, file = "boot12.Rdata")
save(boot22, file = "boot22.Rdata")
```

```
#Figure3a Edge weight bootstrap for the 3 domain network
plot(boot12, labels = FALSE, order = "sample")
fig22c <- plot(boot12, labels = FALSE, order = "sample")
pdf("Figure3a.pdf", width=10, height=7)
plot(boot12, labels = TRUE, order = "sample")
dev.off()

#Figure3b - 3 domain network, Edge weight difference: is edge X significantly larger
than edge Y? Black=Y Gray=N
boot32 <- plot(boot12, "edge", plot = "difference", onlyNonZero = TRUE,
               order = "sample")
plot(boot32)
pdf("Figure3b.pdf", width=10, height=7)
plot(boot32)
dev.off()

################################## Part 3 - 4 domain
network##############

#estimate a new 4 domain structure
cmodel4 <- ' Identify =~ TAS1 + TAS3 + TAS6 + TAS7 + TAS9 + TAS13 +   TAS14
Describe =~ TAS2 + TAS4 + TAS11 + TAS12 + TAS17
ThinkingExt =~ TAS5 + TAS8 + TAS10 +   TAS18 +   TAS19
Distraction =~ TAS15 + TAS16 + TAS20 '

#confirmatory factor analysis for the 4-factor model > fit the model
fit4 <- cfa(cmodel4, data=data)

#get the factorscore dataset
lavPredict(fit4)
factorscoredata4 <- lavPredict(fit4)
factorscoredata4 <- as.data.frame(factorscoredata4)
write_excel_csv(factorscoredata4, "4domainfactorscoredata.csv")

#Estimate a 4-domain network
data3 <- as.data.frame(read_delim("/Users/giovannibriganti/Google
Drive/Network_Analysis_Mac/Briganti2019_TAS/4domainfactorscoredata.csv",
                                  ",", escape_double = FALSE, trim_ws = TRUE))

factornames2 <- c("ID", "DES", "Think", "DST")

longnames2 <- c("Difficulty identifying feelings", "Difficulty describing feelings",
               "Externally-oriented thinking", "Distraction")

#get descriptive statistics for dataset
describe(data3)
means2 <- colMeans(data3, na.rm=T)
pdf("meansplot3.pdf", width=6, height=4)
plot(means3)
```

```
dev.off()
sds <- as.vector(sapply(data3, sd, na.rm=T))
pdf("sdplot3.pdf", width=6, height=4)
plot(sds3)
dev.off()

#node predictability
type=rep('g', 4) #g=gaussian, 4 = number of nodes in the network
fit3<-mgm(data3,
          type=type,
          level=rep(1,4))

pred3<- predict(fit3, data3)
pred3$error$R2
mean(pred3$error$R2)

#estimating network
network3 <- estimateNetwork(data3, default="EBICglasso", corMethod = "cor",
                            corArgs =list(use = "pairwise.complete.obs"),
                            threshold=TRUE, lambda.min.ratio=0.001)

#glasso item network - Figure 4
graph3 <- plot(network3, pie = pred3$error$R2,
               layout="spring", legend.cex=.45,vsize=7,
               border.width=2, border.color='#555555',
               labels=factornames, nodeNames=longnames,
               color=c("#66c2a5", "#d7191c", "#8da0cb","#fc8d62"))
#to see maximum and minimum, set details=TRUE
pdf("Figure4.pdf", width = 10, height=6)
plot(graph3)
dev.off()

#graph2 weight matrix
graph3mat <- getWmat(graph3)
graph3mat #visualize weight matrix
write.csv(graph3mat, "4domainnetworkweightmatrix.csv")

#mean graph1 weight matrix
mean(graph3mat) #visualize the mean edge weight of the network = 0.21

#stability for the 4 domain network
boot13 <- bootnet(network3, ncores=7, nboots=2000)
boot23 <- bootnet(network3, ncores=7, nboots=2000, type="case")
save(boot13, file = "boot13.Rdata")
save(boot23, file = "boot23.Rdata")

#Figure4a Edge weight bootstrap for the 4 domain network
plot(boot13, labels = FALSE, order = "sample")
fig23c <- plot(boot13, labels = FALSE, order = "sample")
pdf("Figure4a.pdf", width=10, height=7)
```

```
plot(boot13, labels = TRUE, order = "sample")
dev.off()

#Figure4b - 4 domain network, Edge weight difference: is edge X significantly larger
than edge Y? Black=Y Gray=N
boot33 <- plot(boot13, "edge", plot = "difference", onlyNonZero = TRUE,
                      order = "sample")
plot(boot33)
pdf("Figure4b.pdf", width=10, height=7)
plot(boot3)
dev.off()
```